Classification by Estimating the Cumulative Distribution Function for Small Data

MENG-XIAN ZHU¹ AND YUAN-HAI SHAO²

¹School of Economics, Hainan University, Haikou 570228, China
²School of Management, Hainan University, Haikou 570228, China

Corresponding author: Yuan-Hai Shao (shaoyuanhai21@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 12271131, Grant 61866010, and Grant 11871183; in part by the Natural Science Foundation of Hainan Province under Grant 120RC449; and in part by the Key Laboratory of Engineering Modeling and Statistical Computation of Hainan Province.

ABSTRACT In this paper, we study the classification problem by estimating the conditional probability function of the given data. Different from the traditional expected risk estimation theory on empirical data, we calculate the probability via Fredholm equation, this leads to estimate the distribution of the data. Based on the Fredholm equation, a new expected risk estimation theory by estimating the cumulative distribution function is presented. The main characteristic of the new expected risk estimation is to measure the risk on the distribution of the input space. The corresponding empirical risk estimation is also presented, and an ε-insensitive L₁ cumulative support vector machines (ε-L₁ VSVM) is proposed by introducing an insensitive loss. It is worth mentioning that the classification model and the classification evaluation indicator based on the new mechanism are different from the traditional one. Experimental results show the effectiveness of the proposed ε-L₁ VSVM and the corresponding cumulative distribution function indicator on validity and interpretability of small data classification.

INDEX TERMS Classification, cumulative distribution function, Fredholm equation, expected risk estimation, support vector machines.

I. INTRODUCTION

Classification [1], [2] is also known as pattern recognition. It is one of the main research problems of machine learning. For binary classification, generally, we are given a set of training samples \( S = \{(x_i, y_i), \ i = 1, 2, \ldots, m\} \), where \( x_i \in \mathbb{R}^d \) is the input and \( y_i \in \{0, 1\} \) is the corresponding output with \( i = 1, \ldots, m \). Suppose the given data is generated from an independent and identically distributed (i.i.d.) distribution, our task is to deduce the output \( y \) of any input \( x \) from the given data set. In fact, the classification problem corresponds to the problem of conditional probability \( P(y|x) \) estimation in statistical inference [3], [4], and it has been studied extensively in both machine learning and statistical inference [5], [6], [7].

A common way to estimate the conditional probability in classification is to suppose a function \( f(x) \approx P(y|x) \). For the binary classification problem, since \( P(y = 1|x) + P(y = 0|x) = 1 \), without loss of generality, we describe the problem of estimating the conditional probability as estimating \( P(y = 1|x) \) from now on. One of the most widely circulated theories on classification is the expected risk minimization theory [8], and the core idea of expected risk minimization theory is to minimize the error of estimation function \( f(x) \) and real output in the expected space. The advantage of expected risk minimization theory is that it does not need to have too many assumptions about the estimation function. Most discriminant classifiers are based on the expected risk minimization theory, such as least squares classifiers (LSC) [9], neural networks (NN) [10], support vector machines (SVM) [11], [12], ensemble learning (EL) [13] et al. In fact, when estimating the expected risk, it is necessary to estimate the distribution structure \( F(x,y) \) of data. For simplicity of calculation, most of the models assume that the data is uniformly distributed, thus there is no operation to estimate the data distribution.
Recently, Vapnik and Izmailov [14] further studied the classification problem by estimating the cumulative distribution function and conditional probability. And proposed a new way to estimate the conditional probability $P(y = 1|x)$ by solving the Fredholm integral equation based on the definition of conditional probability and the density function directly. To solve the Fredholm integral equation, [14], [15] presented a form of least squares implementation called VSVM and compares it with the traditional least squares and support vector machine classifiers. The advantages of using the integral equation are that $f(x)$ can be solved by estimating the distribution function in input space, and the integral equation has wider convergence properties. Therefore, this method has attracted a lot of attention from other machine learning fields [16], [17], [18], [19]. However, the general paradigm of using Fredholm integral equation to infer $f(x)$ is not discussed, and the corresponding expected risk minimization theory and empirical estimation theory are not clear yet.

In this paper, different from the traditional expected risk estimation theory on empirical data, we propose the original theoretical model of probability estimation from the framework of expected risk via Fredholm equation, and present the corresponding empirical model based on the cumulative distribution function estimation. One $\varepsilon$-insensitive empirical risk estimation is established and an $\varepsilon$-insensitive $L_1$ loss cumulative support vector machines ($\varepsilon$-$L_1$ VSVM) is presented. It is worth mentioning that the classification evaluation indicator via estimating Fredholm equation is different from the traditional one. Experimental results show the effectiveness of the proposed $\varepsilon$-$L_1$ VSVM and the corresponding cumulative distribution function indicator on validity and interpretability of classification, especially when the training data is insufficient. The main contributions are summarized as follows:

i) For estimating $P(y = 1|x)$ by using the cumulative distribution function, we perform an objective transformation of the Fredholm integral equation to estimated conditional probabilities from a probabilistic perspective. The main characteristic of using Fredholm integral equation is the function to be evaluated and the real output is equal under the cumulative distribution function.

ii) For estimating $P(y = 1|x)$ by using statistical estimation, the expected risk minimization and empirical risk minimization theory based on the Fredholm integral equation are presented. Due to the cumulative distribution function needs to be estimate, the loss function and the expected risk defined by Fredholm integral equation is different from the traditional one.

iii) Different from traditional 0-1 loss, an insensitive way of estimating the empirical risk using $\varepsilon$-insensitive loss is proposed, it is more efficient than least squares loss and has sparsity. And an $\varepsilon$-$L_1$ VSVM is proposed based on $\varepsilon$-insensitive loss. A new evaluation criterion with distribution information is proposed, an out-of-sample estimation of the $\varepsilon$ value of testing sample.

iv) Experimental results show the effectiveness of the proposed $\varepsilon$-$L_1$ VSVM and the corresponding cumulative distribution function indicator on validity and interpretability of classification, especially when the training data is insufficient.

II. PRELIMINARIES
A. RISK MINIMIZATION IN CLASSIFICATION
In the following, we consider binary classification of learning from training samples through three components [3]:

1) A generator of random vectors $x \in \mathcal{X}$, drawn independently from a fixed but unknown probability distribution function $F(x)$.

2) A supervisor who returns a binary output value $y_i \in \mathcal{Y} := \{0, 1\}$ to every input vector $x_i$, according to a conditional probability function $P(y = 1|x)$, where $P(y = 0|x) = 1 - P(y = 1|x)$, also fixed but unknown.

3) A learning machine capable of implementing a set of indicator functions (functions which take only two values: zero and one) $\mathcal{F}$.

The problem of learning is that of choosing from the given set of indicator functions $\mathcal{F}$, the one that best approximates the supervisor’s response. The selection of the desired function is based on a training set of $m$ i.i.d. observations

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}, x_i \in \mathcal{X}, y_i \in \mathcal{Y}$$

(1)
drawn according to $F(x, y) = F(x)P(y|x)$. To find the function $y = f(x)$ (the mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$) based on a training set that satisfies the above basic assumptions, we need to have some measures of how good a function as a decision function. To this end, we introduce the concept of loss function. This is a function $L(y, f(x))$ which tells us the cost of classifying instance $x \in \mathcal{X}$ as $y \in \mathcal{Y}$. For example, the natural loss function is the 0-1 loss

$$L_{0/1}(y, f(x)) = I(y, f(x)), \tag{2}$$

where

$$I(u, v) = \begin{cases} 0, & \text{if } u = v, \\ 1, & \text{if } u \neq v. \end{cases}$$

The function (2) determines the probability of different answers given by the indicator function $f(x)$ and label $y$. We call the case of different answers a classification error. While the loss function measures the error of a function on some individual data sample, the risk of a function is the average loss over data samples generated according to the underlying distribution $F(x, y)$. In a given set of functions $\{f(x)\}$, find the one that minimizes the expected risk functional

$$R(f) = \mathbb{E}_{X,Y} [L(y, f(x))] = \int_{\mathcal{X} \times \mathcal{Y}} L(y, f(x))dF(x, y). \tag{3}$$

Therefore, the goal of learning is to find the function which minimizes the risk function (3) in the given set of indicator function $\mathcal{F}$ when the probability measure $F(x, y)$ is unknown but training data (1) are given.
Since the training data (1) sampled from the underlying probability distribution $F(x, y)$, we can infer a function $f(x)$ from (1) whose risk is close to expect risk. In practical, we approximate the unknown cumulative distribution function $F(x, y)$ by its joint empirical cumulative distribution function

$$F_m(x, y) = \sum_{i=1}^{m} \theta(x - x_i)\theta(y - y_i),$$

where the one-dimensional step function is defined as

$$\theta(x - x_i) = \begin{cases} 1, & \text{if } x \geq x_i, \\ 0, & \text{otherwise}. \end{cases}$$

and the multi-dimensional step function for $x = (x^1, \ldots, x^d)$ is defined as

$$\theta(x - x_i) = \prod_{k=1}^{d} \theta(x_k - x_{i_k}).$$

Using approximation (4) in the expected loss function (3), we obtain the following empirical loss function

$$R_m(f) = \frac{1}{m} \sum_{i=1}^{m} L(y_i, f(x_i)).$$

That is, given some training data (1), a loss function $L$ and a function space $F$ to work with, we define the classifier $f_m$ as

$$f_m := \arg \min_{f \in F} R_{\text{emp}}(f).$$

This approach is called the empirical risk minimization (ERM) induction principle. The ERM principle is conducted with the law of large numbers well. However, when the sample size $m$ is relatively small, ERM may be inconsistent with the expected risk minimization. Vapnik-Chervonenkis theory [20] introduces a generalization of the empirical risk minimization method which called structural risk minimization (SRM) induction principle [21], [22]. The principle of SRM for solving classification problem is to choose a set of decision functions $F(t)$ that depends on parameter $t$, which satisfies the following nested structure that increases as $t$ increases

$$F(t_1) \subset F(t_2), \forall t_1 < t_2.$$ 

For each $t$, a function $f_t$ is found in the set $F(t)$ that minimizes its empirical risk, and at the same time, there is a value of structural value for $f_t$. We want to choose the best $\hat{t}$ that minimizes the structural risk and use the corresponding function $f_{\hat{t}}$ as the decision function.

An implicit way of dealing with nested function spaces is the principle of regularization. Instead of minimizing the empirical risk $R_{\text{emp}}(f)$ and then expressing the generalization ability of the resulting classifier $f_t$ using some capacity measure of the underlying set of functions $F$, a more direct approach is preferred to minimize the regularization risk

$$R_{\text{reg}}(f) = R_{\text{emp}}(f) + \lambda \Omega(f),$$

where $\Omega(f)$ is the so-called structure term or regularizer, and this regularizer is supposed to punish overly complex functions.

### B. CONDITIONAL PROBABILITY ESTIMATION VIA CDF

In the classical approach to the learning machines (analyzed by the above VC theory), we ignored the fact that the desired decision rule is related to the conditional probability function $P(y = 1|x)$ used by supervisor and the different distribution structure of the data $F(x)$, which will have an impact on the learned mapping $f: X \rightarrow \{0, 1\}$. Reference [14] proposes to consider the problem of directly estimating the conditional probability function $P(y = 1|x)$ using training data (1) as the main problem of learning. And further [14], [15] gives a direct setting of conditional probability estimation problem based on the standard definitions of a density and a conditional probability.

From the perspective of probability, to predict the label $y$, we consider the special (but common) case $P(y = 1|x) = 1 - P(y = 0|x)$. Thus, we describe the classification problem as the estimation problem of conditional probability function, i.e., $f_0(x) = P(y = 1|x)$, where $f_0(x)$ is the function to be estimated and is a single valued deterministic function that at every $x \in \mathbb{R}^d$ specifies the probability $y$.

Consider the definition of cumulative distribution function (CDF), the probability $F(x) = P(\omega < x)$ is CDF, where $\omega \in \mathbb{R}^d$ is a random vector. The density $p(x)$ is defined from CDF if there exists a nonnegative function $p(x)$ such that for all $x$ the following equality is valid

$$\int_{-\infty}^{x} p(\omega) d\omega = F(x).$$

The density $p(y = 1, \omega)$ is defined as the solution of the integral equation

$$\int_{-\infty}^{x} p(y = 1, \omega) d\omega = F(y = 1, x).$$

The definitions of conditional probability function $f_0(x) = P(y = 1|x)$ for continuous $x$ is as follows

$$f_0(x) p(x) = p(y = 1, x)$$

or $f_0(x) = \frac{p(y = 1, x)}{p(x)}, \ p(x) > 0,$

where $p(y = 1, x)$ is the joint density function of $x$ and $y = 1$, $p(x)$ is the probability density function (PDF) of $x$, $f_0(x)$ is the ratio of the two density functions. Different from using (8) and Bayes theory [23] to calculate $f_0(x)$, we consider another direct definition of the conditional probability function based on (7) and (8), which is defined as the solution of the Fredholm equation

$$\int G(x - x') f_0(x') dF(x') = \int G(x - x') dF(y = 1, x'),$$

where $G(x - x')$ is any kernel from $L_2$ space, $F(x)$ and $F(y = 1, x)$ are the CDF of $x$ and joint distribution function of
and \( y = 1 \), respectively. The solution of the Fredholm equation (9) directly defines the conditional probability function \( f_0(x) = P(y = 1|x) \) when the kernel \( G(x - x') \) is given and \( F(x) \) and \( F(x, y = 1) \) are known.

Compared to (8), there is no density function in the integral equation, but only distribution functions in (9). There are at least two main advantages of using (9) to calculate \( f_0(x) \).

i) The distribution function is monotonically non-decreasing function with a value range of \([0, 1]\), which is easier to be estimated than the ratio of two probability density functions, for example, it can be approximated by the empirical cumulative distribution function (ECDF).

ii) The conditional probability function directly controls many different statistical properties that are expressed in the training data, such as the distribution of the data. Keeping these invariants in the rule is equivalent to incorporating some training data, such as the distribution of the data. Keeping these invariants in the rule is equivalent to incorporating some training data, such as the distribution of the data. Keeping these invariants in the rule is equivalent to incorporating some training data, such as the distribution of the data.

In order to solve (9), [14] estimates \( f(x) \) using the empirical distribution approximations of \( F(x) \) and \( F(y = 1, x) \) which are given by \( F_m(y = 1, x) = \frac{1}{m} \sum_{i=1}^{m} y_i \theta(x - x_i) \) and \( F_m(x) = \frac{1}{m} \sum_{i=1}^{m} \theta(x - x_i) \). At this time, the strict equality relation (9) is hard to establish. It solves (9) in the set of function \( \mathcal{F} \) by minimizing the distance of the \( L_2 \) metric

\[
\rho^2 = \frac{1}{m^2} \int \left( \sum_{i=1}^{m} G(x - x_i) f_0(x_i) - \sum_{i=1}^{m} y_i G(x - x_i) \right)^2 d\mu(x),
\]

where \( \mu(x) \) is a probability measure defined on domain \( D \) consisting of \( x \in \mathbb{R}^d \). Under the given training samples, it simplifies to

\[
\rho^2 = \frac{1}{m^2} \sum_{i,j=1}^{m} (f_0(x_i) - y_i)(f_0(x_j) - y_j) v(x_i, x_j),
\]

where \( v \) is an \( m \times m \)-dimensional symmetric non-negative matrix and its specific form is as follow

\[
v(x_i, x_j) = \int G(x - x_i) G(x - x_j) d\mu(x).
\]

Obviously, considering the characteristics of the data, we can choose the corresponding appropriate kernel functions \( G(\cdot) \) and \( \mu(x) \), so as to get a more accurate location information. And the value of \( v \) does not depend on labels \( y_i \) but shows the location of each sample in the distribution. It’s an objective property of the input while it’s ignored in most researches, such as the classical least square method only consider the residual error \( \Delta_i = f_0(x_i) - y_i \), \( i = 1, \ldots, m \) of each \( x_i \) and take less account of the distribution of data.

As the approximation (10) of Fredholm equation may be an ill-posed problem, [14] gives the solution of Tikhonov’s regularization method as

\[
R(f) = \frac{1}{m^2} \sum_{i,j=1}^{m} (f_0(x_i) - y_i)(f_0(x_j) - y_j) v(x_i, x_j) + \gamma \| f(x) \|_2,
\]

where \( \gamma > 0 \) is the regularization constant.

By defining the following notations: \( A = (a_{ij}, \ldots, a_{im})^T, K(x_i) = (K(x_1, x_i), K(x_2, x_i), \ldots, K(x_m, x_i))^T, m \times m \)-dimensional matrix \( K \) with \( K_{ij} = K(x_i, x_j) \) being the \( ij \)-th entry, \( Y = (y_1, y_2, \ldots, y_m)^T, 1_m = (1, 1, \ldots, 1)^T, I \) is the \( m \times m \)-dimensional identity matrix. Then \( f(x) = A^T K(x_i) + c \) and \( R(x) \) can be rewritten as

\[
R(A) = (KA + c1_m)^T V(KA + c1_m) - 2(KA + c1_m)^TYV + Y^T Y\gamma^T K A
\]

and the closed form solution for \( A \) and \( c \) are given by \( A = A_b - cA_c, A_b = (VK + \gamma I)^{-1} YV, A_c = (VK + \gamma I)^{-1} Y1_m \) and \( c = \frac{i^T V(Ka_b Y)}{1^T V(Ka_b Y - 1_m)} \).

Although this model looks very similar to the weighted support vector machines [24], [25], [26] and the weighted least squares support vector machines [27], [28], their starting points and functions are very different. In the following, we will explain their differences by further analyzing the estimated distribution function.

III. CLASSIFICATION BY CUMULATIVE DISTRIBUTION ESTIMATION THEORY

A. PROBABILITY CALCULATION VIA FREDHOLM EQUATION

In this subsection, we consider the conditional probability \( P(y = 1|x) \) calculation from the Fredholm equation theory. According to the definition of the density function \( p(x) \) and \( p(y = 1, x) \) in (7), we have

\[
P(y = 1|x) = \frac{p(x, y = 1)}{p(x)} = \frac{\partial (F(x, y = 1))}{\partial (F(x))}, \tag{12}
\]

when the derivatives of \( F(x) \) and \( F(x, y = 1) \) exist and non-zero. Therefore, (9) can be simplified to

\[
\int G(x - x') f_0(x') dF(x') = \int G(x - x') \left( \frac{\partial (F(x', y = 1))}{\partial (F(x'))} \right) dF(x'), \tag{13}
\]

where \( \partial (v) \) represents the gradient of \( v \).

Instead of treating each random variable \( X \) as an independent individual and aiming at let \( f_0(x) = P(y = 1|x) \), where \( P(y = 1|x) \) denotes the true and known conditional probability of \( x \). We see that there are different ways to compute the conditional probability \( f_0(x) \) from (9), (12) and (13) by using the cumulative distribution function. Here each random variable \( X \) is in the entire distribution, and the method of directly estimating conditional probability using (9) considers the distribution. For each sample \( x \), a relationship with the relative position of other samples in the distribution is established, which contains the prior information of the data.

In real-world problem, the classical setting of output is hard output in classification, that is the label of the sample is known instead of the conditional probability of the sample. At this time, we take into account the fact that the transformation rule for classification. For binary classification problem
with equal class prior probabilities, we have

\[ r(x) = \theta \left( \frac{P(y = 1|x) - 0.5}{p(x)} - 0.5 \right), \]

where \( \theta(z) \) is the step function. Thus, the rule \( r(x) \) classifies vector \( x \) to the first class \( (y = 1) \) if the conditional probability \( p_0(x) \) exceeds 0.5. It is worth noting that when the category prior probabilities are not equal, \( r(x) = \theta \left( p_0(x) - p(y = 1) \right) \).

Different from estimating the conditional probability function \( f(x) \), we compute the hard output classification rule \( r(x) \). Similarly, the standard definitions of \( r(x) \) for continuous \( x \) is as follows

\[ r(x) = \theta \left( \frac{P(y = 1,x) - 0.5}{p(x)} - 0.5 \right) = y, \ y \in \{0,1\}. \quad (14) \]

Then corresponding to \((9)\), we have the direct definition of decision rule \( r(x) \).

\[ \int G(x-x')r(x')dF(x',y') = \int G(x-x')y'dF(x',y'). \quad (15) \]

Compare the two pairs of expressions \((8)\) and \((9)\) or \((14)\) and \((15)\), we have

i), when the kernel function is chosen as \( (5) \) (this kernel function was mentioned and used in \[15\]), which covers the setting of cumulative distribution function), \((15)\) or \((9)\) is an equation about the distribution function, while \((14)\) or \((8)\) is an equation about the density function. \((15)\) or \((9)\) is easier to compute with respect to \((14)\) or \((8)\) because in the probability theory, the distribution function is determined firstly while the density function is defined by the distribution function. And the distribution function is unique, the density function is not necessarily unique.

ii), when the choice of kernel function is extended to any kernel function from the \( L_2 \) space, this involves two different modes of convergence in Hilbert space \[15\], which describes the relationship between a sequence of functions \( r_0(x) \in L_2 \) and the desired function \( r_0(x) \). The strong mode of convergence (convergence in metrics) is expressed as

\[ \lim_{n \to \infty} \|f_n(x) - f_0(x)\| = 0, \forall x. \]

The weak mode of convergence (convergence in inner products) is

\[ \lim_{n \to \infty} \langle G(x), f_n(x) - f_0(x) \rangle = 0, \forall G(x) \in L_2. \]

Obviously, \((15)\) or \((9)\) is an implementation of weak convergence, which is more common in real life than strong convergence mode. From this point of view, probability calculation via Fredholm equation is more general than the traditional one due to the weak convergence is also strong convergence.

The above discussion is a description of the outcomes of random experiments for a theoretical model from the perspective of probability theory, without considering the specific data and hypothesis space of functions. The next part will discuss the classification from a statistical perspective.

### B. Expected Risk of Fredholm Equation

Consider the statistical estimation problem with an input space \( \mathcal{X} \) and the output set \( \mathcal{Y} = \{0,1\} \) and a set of training samples \((1)\). If we estimate \((9)\) under the above assumptions, the integral equation might not hold strictly. In order to find the optimal decision function satisfied the above assumptions, we firstly introduce the concept of loss function to evaluate the pros and cons of decision function \( f(x) \).

**Definition 1 (Loss Function Based on Fredholm Equation):** For quadruple \((y, r(x), F(x, y), G(\cdot)) \in \mathcal{Y} \times \mathcal{Y} \times \mathcal{F} \times \mathcal{P}\), where \( \mathcal{Y} = \{0,1\}, \mathcal{P} = \{0,1\} \), \( y \) is an observation corresponding to the input \( x \), and \( r(x) \) is the conditional probability value of the decision function \( r \) at \( x \). For a given kernel function \( G(\cdot) \in L_2 \) and any \( x \in \mathcal{X}, y \in \mathcal{Y} \), if the mapping \( L_{Fred}: \mathcal{Y} \times \mathcal{Y} \times \mathcal{F} \times \mathcal{P} \rightarrow [0, \infty) \), there are \( L_{Fred}(y, x, F(x, y), G(\cdot)) = 0 \), then it is called \( L_{Fred} \) a loss function based on the Fredholm equation.

Now, the classification problem becomes to estimate Eq.\((9)\) with minimized losses. For a given kernel function \( G(x-x') \), the loss function based on the Fredholm equation corresponding to the decision function \( f(x) \) is expressed as

\[ L_{Fred}(y, f(x), F(x, y), G(\cdot)) = L \left( \int G(x-x')r(x')dF(x',y'), \int G(x-x')y'dF(x',y') \right). \quad (16) \]

The value of loss function \((16)\) is just a quantitative measure of how well a decision function \( f(x) \) performs at any input \( x \). For the population, the expected risk is introduced below to assess the overall effect. The expected loss of \((16)\) is written as

\[ R_{Fred}(f) = \mathbb{E}_x \left[ L \left( \int_{\mathcal{X} \times \mathcal{Y}} G(x-x')r(x')dF(x',y'), \right. \right. \]

\[ \left. \left. \int_{\mathcal{X} \times \mathcal{Y}} G(x-x')y'dF(x',y') \right) \right] \]

\[ = \int_X L \left( \int_{\mathcal{X} \times \mathcal{Y}} G(x-x')r(x')dF(x',y'), \right. \]

\[ \left. \int_{\mathcal{X} \times \mathcal{Y}} G(x-x')y'dF(x',y') \right) dF(x), \quad (17) \]

where \( F(x) \) is a distribution function with respect to random variables \( X \).

Compared to the loss in \((3)\), the loss function \((16)\) measures the difference between the two sides of the integral equation \((9)\), it is the difference between a series of \( y \) and \( f(x) \) under the concept of distribution. While for \((3)\) and \((17)\), we have the following theorems.

**Theorem 3.1:** Given \( R(x) \) in \((3)\) and \( R_{Fred}(x) \) in \((17)\), assuming that the distribution structure of the data is not considered, i.e., \( F(x, y) = \theta(x-x_0)\theta(y-y_0) \), \( R_{Fred}(x) \) degenerates to classical expect risk \( R(x) \), where

\[ F(x, y) = \theta(x-x_0)\theta(y-y_0) = \begin{cases} 1, & \text{if } x \geq x_0, y \geq y_0, \\ 0, & \text{otherwise}. \end{cases} \]

(proof is in the appendix)
In short, the statistical formulation of the original classification problem can be expressed as: in a given set of functions \( \mathcal{F} \), estimate the one that minimizes the expected risk for functional (17), under the assumption that probability measure \( F(x) \) and \( F(x, y) \) are unknown but \( m \) i.i.d. pairs \( (x_i, y_i) \), where \( i = 1, \ldots, m \), satisfied this distribution are given.

Theoretically, the proposed expected risk is an implementation of weak convergence approximating strong convergence, while the traditional one only considers strong convergence mode. Further, from the structure of the above expect risk, the inner layer in equation (17) over the \( X \times Y \) space describes the difference between the estimated and true values of a series of pairs \( (x_i', y_i') \), and the outer layer over the \( X \) space and the kernel function \( G(x - x') \) jointly construct information about the structure of the distribution of \( x' \). Thus, from the perspective of empirical estimation, the outer layer structure captures information about the distribution of the data, which in turn improves the performance of the estimated classifier. The details of implementation will be elaborated in Section IV.

**C. EMPIRICAL RISK OF FREDDHOLM EQUATION**

Now, we express the integral equation (15) by the form

\[
Af = F, \quad (18)
\]

where \( A \) is a linear operator which maps the elements \( f \) of a metric space \( E_1 \) into the elements \( F \) of a metric space \( E_2 \). Then the empirical risk of the given sample \( \{(x_i, y_i)\}_{i=1}^m \) is expressed as the distance on both sides of the equation (15) in \( E_2 \) space. We mark it as \( \rho_{E_2}(A_{\text{emp}}, F_m) \).

We say that the problem of solving operator equation (18) in the set \( \mathcal{F} \) is well-posed if the solution exits, is unique, and is continuous. The problem is called ill-posed if at least one of the three above conditions is violated. The solution of ill-posed problems is based on the following lemma.

**Lemma 1 (Lemma About Inverse Operator):** If \( A \) is a continuous one-to-one operator defined on compact set \( \mathcal{M} \) of functions \( f \), then the inverse operator \( A^{-1} \) is continuous on the set \( \mathcal{N} = AM \). Consider a continuous non-negative functional \( W(f) \) and the set of functions

\[
\mathcal{M}_C = \{ f : W(f) \leq C \},
\]

defined by a constant \( C > 0 \). Let the set of functions \( \mathcal{M}_C \) be convex and compact for any \( C \). Suppose that the solution of operator equation belongs to compact sets \( \mathcal{M}_C \).

The idea of solving ill-posed problems is to choose an appropriate compact set (i.e., to choose a constant \( C^* \)) and then solve (18) on the compact set of functions defined by \( C^* \). In other words, to minimize the distance on both sides of the equation (15) in \( E_2 \) space

\[
\rho = \min \rho_{E_2}(A_{\text{emp}}, F_m)
\]

over functions \( f \) subject to the constraint

\[
W(f) \leq C^*.
\]

This is similar to that of Tikhonov’s regularization method. In this way, the function

\[
R(f) = \rho_{E_2}(A_{\text{emp}}, F_m) + \gamma W(f)
\]

(19)
is minimized, where \( \gamma > 0 \) is a regularization parameter. Obviously, this idea of solving ill-posed problem is the same as structural risk minimization in VC theory. In both cases, a structure is defined on the set of functions. When solving well-posed problems, elements of structure should have finite VC-dimension. When solving ill-posed problems, elements of structure should be compact sets.

In practical, \( F(x) \) in expected risk (17) can be estimated with given i.i.d. data. A common method is to approximate the unknown cumulative distribution functions by their empirical cumulative distribution functions (4). Using approximation (4) in the expected loss function (17), we obtain the empirical loss function and the expression of the specific empirical risk will be described in detail in the next section.

**IV. CLASSIFICATION BY CUMULATIVE DISTRIBUTION ESTIMATION**

**A. MODEL ESTIMATION WITH LOSS FUNCTION**

Now, we consider the empirical estimation of the expected risk (17) with the 0-1 loss function [11], [29]. As is discussed above, we approximate the unknown cumulative distribution function \( F(x', y') \) by using empirical cumulative distribution function (4). Putting (4) into (17) instead of unknown elements \( F(x', y') \), we obtain the following approximation

\[
\hat{R}(f) = \int I \left( \sum_{i=1}^m G(\hat{x} - x_i)r(x_i), \sum_{i=1}^m G(\hat{x} - x_i)y_i \right) d \mu(\hat{x}).
\]

(20)

Note that \( \mu(\hat{x}) \) is a cumulative distribution function which define the concept of closeness of \( \hat{x} \).

**Remark:** Random variables \( x \) and \( x' \) in (17) are both from the fixed unknown distribution \( P \) on \( X \). By using the empirical cumulative distribution function in (20), \( x' \) is corresponding to the labeled sample \( x_i \) from training set, and \( x \) is corresponding to \( \hat{x} \) (here \( \hat{x} \) is an arbitrary sample from the distribution, whether it’s labeled or not, continuous or not). We point out that \( \mu(\hat{x}) \) could be estimated by the empirical data or other different ways.

For each \( \hat{x} \), we substitute training data \( (x_i, y_i) \) instead of a continuous variable \( x' \) in (17). Meanwhile, (15) is hard to satisfied because the empirical estimation is hard to be exactly equal and we have

\[
\sum_{i=1}^m G(\hat{x} - x_i)r(x_i) \approx \sum_{i=1}^m G(\hat{x} - x_i)y_i.
\]

(21)

Therefore, the indicator function \( I(\cdot, \cdot) \) in the empirical approximation expression (20) obtained from the expected risk takes a high probability of 1. There will be a large bias in evaluating the performance of \( f(x) \) by using 0-1 loss function above. It means that the traditional 0-1 loss is not suitable for
this type of problem. In this case, for each \( \hat{x} \), we give a small tolerance to empirical estimation
\[
\| \sum_{i=1}^{m} G(\hat{x} - x_i)r(x_i) - \sum_{i=1}^{m} G(\hat{x} - x_i)y_i \| \leq \varepsilon,
\]
where \( \varepsilon > 0 \) is a small scalar. Then, the 0-1 loss converts to the 0-1 loss with \( \varepsilon \)-insensitive zone, and (20) is transformed to the following unconstrained optimization problem
\[
\min_{f} \int I_x \left( \sum_{i=1}^{m} G(\hat{x} - x_i) r(x_i) - \sum_{i=1}^{m} G(\hat{x} - x_i)y_i \right) d\mu(\hat{x})
+ \lambda W(f),
\]
where 0-1 loss function with \( \varepsilon \)-insensitive zone is
\[
I_x(u, v) = \begin{cases} 0, & \text{if } |u - v| \leq \varepsilon, \\ 1, & \text{otherwise.} \end{cases}
\]
\( W(f) \) is the regularization term to avoid the ill-posed problem and \( \lambda \) is a positive regular parameter.

Noticing that both \( I_x(u, v) \) and \( r(\cdot) \) are nonconvex discontinuous, arises from the fact that the gradient of the objective function is either not defined or is equal to zero, and it is difficult to optimize. In order to solve this problem, we use common alternative to utilize a convex surrogate to substitute the non-convex 0-1 type function as
\[
\min_{f} \int I_x \left( \sum_{i=1}^{m} G(\hat{x} - x_i) (f(x_i) - y_i) \right) d\mu(\hat{x}) + \lambda W(f),
\]
where \( \varepsilon \)-insensitive loss function is
\[
\ell_x(z) = \begin{cases} 0, & \text{if } |z| \leq \varepsilon, \\ |z| - \varepsilon, & \text{otherwise.} \end{cases}
\]

Now, we have known all other statistical variables except for \( f(x) \). In the following, we are looking for solutions of our inference problem in the set of functions \( F \) that belongs to Reproducing Kernel Hilbert Space (RKHS) associated with kernel \( K(x, x') \), where \( K(x, x') \) is a continuous positive semi-definite function of variables \( x, x' \in X \subset \mathbb{R}^n \). For the solution in RKHS with bias, we use the representation
\[
f(x) = \sum_{i=1}^{m} a_i K(x_i, x) + b,
\]
where \( x_i, i = 1, \ldots, m \) are vectors from the training set. For the regularization term, we have \( W(f) = \| f(x) \|^2 = \sum_{i=1}^{m} a_i a_j K(x_i, x_j) \). Therefore, finding the solution of (17) in a given set of functions \( F \) requires the minimization of the functional
\[
\min_{f} \int I_x \left( \sum_{i=1}^{m} G(\hat{x} - x_i) (f(x_i) - y_i) \right) d\mu(\hat{x}) + \lambda \| f(x) \|^2.
\]

In fact, VSVM [14] is also could be considered as an approximation to (17) by using the \( L_2 \) metrics
\[
\min_{f} \int \left( \sum_{i=1}^{m} G(\hat{x} - x_i)f(x_i) - y_i \right)^2 d\mu(\hat{x}) + \lambda \| f(x) \|^2.
\]

B. \( \varepsilon-L_1 \) VSVM
In this subsection, we discuss the solution of the optimization objective function (23). Since \( G(\cdot) \) in (23) is a non-negative function. The following inequality hold
\[
\int \ell_x \left( \sum_{i=1}^{m} G(\hat{x} - x_i) (f(x_i) - y_i) \right) d\mu(\hat{x})
= \int \ell_x \left( \int \left( \sum_{i=1}^{m} G(\hat{x} - x_i) (f(x_i) - y_i) \right) d\mu(\hat{x}) \right)
\leq \int \ell_x \left( \sum_{i=1}^{m} G(\hat{x} - x_i) (f(x_i) - y_i) \right) d\mu(\hat{x})
\leq \int \sum_{i=1}^{m} G(\hat{x} - x_i) \ell_x (|f(x_i) - y_i|) d\mu(\hat{x})
= \sum_{i=1}^{m} \ell_x (f(x_i) - y_i) \int G(\hat{x} - x_i) d\mu(\hat{x})
\]
\( \leq \sum_{i=1}^{m} \ell_x (f(x_i) - y_i) \int G(\hat{x} - x_i) d\mu(\hat{x}) + \lambda \| f(x) \|^2. \)

Therefore, under this inequality, minimize (23) turns into
\[
\min_{f} \sum_{i=1}^{m} \ell_x (f(x_i) - y_i) \int G(\hat{x} - x_i) d\mu(\hat{x}) + \lambda \| f(x) \|^2.
\]

The advantage of minimizing (25) is that the variables \( \hat{x} \) and \( x_i \) are separable. Thus, in practice, minimization (25) can be performed in two steps:

**Step 1:** In the domain of \( \hat{x} \), for each training sample \( x_i \), compute the following equation and call them \( v \)-vector
\[
v_i = \int G(\hat{x} - x_i) d\mu(\hat{x}).
\]

**Step 2:** In the set of \( \{ f(x) \} \), minimize
\[
\min_{f} \sum_{i=1}^{m} \ell_x (f(x_i) - y_i) v_i + \lambda \| f(x) \|^2.
\]

Note that \( v_i \) is the integral over \( \hat{x} \) for the training sample \( x_i \) in the distribution, so \( v_i \) can be obtained by the parameter or non-parameter estimation above. In (26), \( x_i \) is the training sample which has the corresponding label \( y_i \), whereas \( \hat{x} \) can be any \( \hat{x} \) from the unknown distribution \( P(x) \). From this point of view, use the prior or unlabeled samples to estimate a more accurate location information \( v_i \) about the sample \( x_i \) than only using training set is promising, and a relationship between the training set (labeled samples) and the test set (unlabeled samples) through \( v \) method could be established.

The model (27) is called \( \varepsilon-L_1 \) VSVM as it is similar to a weighted SVR model, where the weights are \( v \)-values describing the location information of the training samples. Unlike traditional weighted models, the weight in (27) is a kind of knowledge from distribution, adding \( v \) to the model is similar to adding the geometric properties attributes of the data to the model. As discussed above, \( v \) can construct a link
between the training samples and the overall distribution to improve the model performance. In the experiments, we first use all the samples (including the test set) to construct \( v \)-values to manifest the location information of training set, and then use it for modeling.

To solve \( \varepsilon \)-L1 VSVM effectively, we rewrite (27) as

\[
\min_{f} \frac{1}{2} \| f(x) \|^2 + \gamma \sum_{i=1}^{m} \ell_i(f(x_i) - y_i)v_i, \tag{28}
\]

where the specific form of \( v_i \) can be chosen according to the data characteristics. By introducing the slack variables, (28) can be equivalently written as

\[
\min_{f, \xi, \xi^*} \gamma \sum_{i=1}^{m} (\xi_i^* + \xi_i) + \frac{1}{2} \sum_{i,j=1}^{m} a_i a_j K(x_i, x_j),
\]

s.t. \( -\varepsilon - \xi_i^* \leq f(x_i) - y_i \leq \varepsilon + \xi_i \),

\[\xi_i^* \geq 0, \xi_i \geq 0.\]  \tag{29}

By defining the following notations: \( a = (a_1, \ldots, a_m)^T, \alpha = (\alpha_1, \ldots, m)^T, \alpha^* = (\alpha_1^*, \ldots, m^*)^T, \xi = (\xi_1, \ldots, \xi_m)^T, \xi^* = (\xi_1^*, \ldots, \xi_m^*)^T, v = (v_1, \ldots, v_m)^T, 1 \times l \)-dimensional matrix \( K \) with \( K_{ij} = K(x_i, x_j) \) being the \( ij^{th} \) entry, \( y = (y_1, y_2, \ldots, y_m)^T, l = (1, 1, \ldots, 1)^T \), we consider the dual formulation of (29). The Lagrangian with \( \mu_i \geq 0, \mu_i^* \geq 0, \alpha_i \geq 0, \alpha_i^* \geq 0 \) of (29) is

\[
L(a, b, \xi, \xi^*, \alpha, \mu, \mu^*) = \frac{1}{2} a^T Ka + \gamma (\xi^* + \xi)^T v - \mu^T \xi - \mu^* v + \alpha^T (K + b_1 m - Y - 1 m) - \alpha^* (Y - K - b_1 m - 1 m - \xi^*). \tag{30}
\]

According to

\[
\frac{\partial L}{\partial a} = Ka + K\alpha - \alpha^* = 0, \quad \frac{\partial L}{\partial b} = a^T 1_m - a^* 1_m = 0, \quad \frac{\partial L}{\partial \xi} = \mu + \gamma v - \alpha = 0, \quad \frac{\partial L}{\partial \xi^*} = -\mu^* + \gamma v - \alpha^* = 0, \tag{31}
\]

we obtain the dual problem of (29) as follows

\[
\max_{\alpha, \alpha^*} (\alpha^* - \alpha)^T Y - \varepsilon (\alpha^* + \alpha)^T 1_m - \frac{1}{2} (\alpha^* - \alpha)^T K (\alpha^* - \alpha),
\]

s.t. \( (\alpha^* - \alpha)^T 1_m = 0, \)

\[0 \leq \alpha, \alpha^* \leq \gamma v. \]  \tag{32}

Since the \( \varepsilon \)-insensitive loss is sparse and the dual problem (32) has sparse solutions, we have the following lemma.

**Lemma 2 (Lemma About Inverse Operator):** \( \alpha_i \) takes a non-zero value only if \( f(x_i) - y_i - \varepsilon - \xi_i = 0 \) and \( \alpha_i^* \) takes a non-zero value only if \( f(x_i) - y_i - \varepsilon - \xi_i^* = 0 \). In addition, constraints \( f(x_i) - y_i - \varepsilon - \xi_i = 0 \) and \( f(x_i) - y_i - \varepsilon - \xi_i^* = 0 \) cannot be held at the same time, so at least one of \( \alpha_i \) and \( \alpha_i^* \) is zero.

From the Karush-Kuhn-Tucker [30] conditions

\[
\begin{align*}
\alpha_i (f(x_i) - y_i - \varepsilon - \xi_i) &= 0, \\
\alpha_i^* (f(x_i) - y_i - \varepsilon - \xi_i^*) &= 0, \\
\alpha_i \alpha_i^* &= 0, \xi_i \xi_i^* &= 0, \\
(y_i v_i - \alpha_i) \xi_i &= 0, (y_i v_i - \alpha_i) \xi_i^* &= 0,
\end{align*}
\]

we have the conclusion that \( \alpha_i \) and \( \alpha_i^* \) takes a non-zero value only if the sample \( (x_i, y_i) \) does not fall into the \( \varepsilon \)-insensitive zone, and at least one of \( \alpha_i \) and \( \alpha_i^* \) is zero. That is to say, the solutions of \( \varepsilon \)-L1 VSVM is sparse.

**C. EVALUATION**

In this subsection, we discuss the indicator and the evaluation of the classification. Typically, the generalisation error of a classifier cannot be evaluated directly but estimated by some testing error [5]. The ‘testing error’ on the test set is then used as an approximation to the generalisation error. Assuming that the probability of test samples is equal or unknown, we usually use the error rate or accuracy rate to evaluate the classifier. Given a testing set \( \{x_t, y_t\}_{t=1}^{T} \), the accuracy rate is defined as follows

\[
Acc_{test} = \frac{1}{T} \sum_{t=1}^{T} 1(y_t = r(x_t)), \tag{33}
\]

where

\[
1(y_t = r(x_t)) = \begin{cases} 
1, & \text{if } y_t = r(x_t), \\
0, & \text{otherwise}.
\end{cases}
\]

As for the expected risk (17), it estimates the relative position and distribution of samples based on the Fredholm equation, which requires that for each sample \( x_t \), satisfying \( r(x_t) = y_t \) in the sense of distribution (That is, the integral equation is satisfied). Therefore, it is reasonable to construct the corresponding evaluation indicator considering the distribution than to use the classical one (the probability of test samples is equal or unknown).

Based on the above proposed \( v \)-vector, for testing samples, we define the testing accuracy by considering location information of testing samples as

\[
Vac_{test} = \frac{1}{T} \sum_{t=1}^{T} 1(y_t = r(x_t)) \int_{\tilde{X}} G(\hat{x} - x_t) d\mu(\hat{x})
\]

\[
= \frac{1}{T} \sum_{t=1}^{T} 1(y_t = r(x_t)) v_t. \tag{34}
\]

Clearly, when the data distribution is uniform or not considered, we have

\[
\mu(\hat{x}) = \theta(\hat{x} - x_t), \tag{35}
\]

and

\[
v_t = \int_{\tilde{X}} G(\hat{x} - x_t) d\mu(\hat{x}) = G(x_t - x_t) = 1.
\]

At this time, (34) degenerates into classical accuracy (33).
Traditionally, the data distribution is non-uniform and we need to estimate the $v$ vector out-of-sample. As discussed above, $v_t$ describes the position of each sample in the distribution, corresponding to the samples one by one. Now, we show that $v_t$ also could be estimate out-of-sample. Similarly, two types of $v_t$ are given, depending on whether the expression contains parameters.

1) ($v_t$ without parameter) Let $G(\hat{x} - x_t) = \theta(\hat{x} - x_t) = \begin{cases} 1, & \text{if } \hat{x} \geq x_t \\ 0, & \text{otherwise} \end{cases}$. Consider one-dimensional case of $\hat{x} \in (-\infty, \infty)$, where $\hat{x}$ follows a normal distribution $(a, b^2)(i.e., \mu(\hat{x}) = 1/\sqrt{2\pi b^2})$. We have

$$v_t = -\frac{1}{2a} \int^{a}_{-a} \exp\left(-\frac{(\hat{x} - x_t)^2}{2\sigma^2}\right) d\hat{x}.$$  

After integration, we obtain

$$v_t = C \left( \text{erf} \left( \frac{a + x_t}{\sigma} \right) + \text{erf} \left( \frac{a - x_t}{\sigma} \right) \right),$$

where $C$ is a constant. For multidimensional case of $\hat{x}$, the elements $v_t$ of $v$-vector have the form

$$v_t = \prod_{k=1}^{d} \left( \int^{x_t^k}_{-\infty} \frac{1}{\sqrt{2\pi b^2}} \exp\left(-\frac{(t - a)^2}{2b^2}\right) dt \right),$$

where $C$ is a constant. For multidimensional case of $\hat{x}$, the elements $v_t$ of $v$-vector have the form

$$v_t = \prod_{k=1}^{d} \left( \int^{x_t^k}_{-\infty} \frac{1}{\sqrt{2\pi b^2}} \exp\left(-\frac{(t - a)^2}{2b^2}\right) dt \right).$$

Algorithmic Remark: In high dimensions of the $v$-vector stated in (26) and $v$-matrix stated in (11) may be ill-conditioned [14], [31]. So, it suggests using an additive version of the $v$-vector. This version derives from considering each dimension of the empirical distribution function separately and then minimizing the sum of the $L_1$ or $L_2$ errors. Thus, $v$-vector becomes

$$\hat{v}_t = \int \left( \sum_{k=1}^{d} G(\hat{x}^k - x_t^k) \right) d\mu(\hat{x}).$$

It is worth noting that $x_t$ in the expression of $v$ represents the test samples, and $\hat{x}$ can be broader in scope to include all samples (e.g., training set and testing set). This leads the position of the samples in the whole dataset, including the training samples and test samples. Obviously, for a given dataset, if $v$-vector without parameter is used, then the calculated $v$ is fixed, and the results of evaluation indicator can be compared directly between different models. However, if a parameterized $v$-vector is used, the value of $v$ varies with the selection of parameter $\sigma$, and the values of evaluation indexes of different models are not comparable. Therefore, we only use the above evaluation indicator in the cross-validation selection process in the experiments. In addition, from the perspective of knowledge, for other models that do not consider data distribution information, using the above proposed evaluation indicator is another way to adding the priori knowledge of data distribution information to the modeling (model selection), thus improving the performance of the model.

Compared to the expression (33) and (34), in (33), the weight given to each test sample in the testing set is 1, i.e., each test sample is treated equally, without considering the location information of the samples. Whereas, in (34), it computes not only the residuals of the true and estimated values, but also information about the distribution of each sample.

V. NUMERICAL EXPERIMENTS

In this section, we carry out all our numerical experiments to compare the proposed method and improved evaluation indicator with other five related SVM classifiers. Firstly, we show the details of the experimental setup. Then, we present an empirical analysis based on two synthetic datasets, where the optimal Bayes classifier is known. At last, we compare all classifiers over various widely investigated real-life small datasets. All experiments are implemented by using Matlab R2020a on a desktop with 8 Intel® Core™ i7-7700K processors (4.20 GHz) and 32GB RAM.

A. EXPERIMENTAL SETUP

Five related SVM-type classifiers are used for experimental comparison with the proposed $\varepsilon$-L1 VSVM.\(^1\)

- **CSVM [32]:** CSVM is the classical SVM with hinge loss which maximizes the margin between the nearest points of different classes.
- **LSSVM [33]:** LSSVM is a least squares version of the support vector machine, which derives its solution from solving a set of linear equations.
- **IDLSSVM [34]:** IDLSSVM (improved density weighted LSSVM) is an improved 2-norm-based density-weighted least squares SVM for binary classification which considers the density of each sample in its class.
- **VSVM [14]:** VSVM is a model constructed based on the Fredholm integral equation and the least squares loss.
- **$\varepsilon$-L1 SVM:** When $v$-vector is an all-1 vector, the proposed $\varepsilon$-L1 VSVM recover to this model, which can be used for classification problems and is denoted as

\(^1\)Codes available at: http://www.optimal-group.org/Resources/Code/L1VSM.html
To evaluate the classification performance, two commonly employed metrics are applied: accuracy and G-means.

- Accuracy represents the proportion of correctly classified instances, i.e., \( \text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \).
- Sensitivity (also referred as recall) explains the prediction accuracy among positive instances, i.e., \( \text{Sensitivity} = \frac{TP}{TP+FN} \).
- Specificity represents the prediction accuracy among negative instances, i.e., \( \text{Specificity} = \frac{TN}{FP+TN} \).
- G-mean is defined as the geometric mean of sensitivity and specificity, i.e., \( \text{G-mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}} \).

Since we need to compare Acc and Vac, Bayes decision is the main indicator of artificial data experiments and G-mean is the main indicator of benchmark data experiments. Since all methods compared in the paper solve the quadratic programming problems and linear system of equations, they are fast in small size problem, and the CPU time is affected by other factors of the computer operation, so the running time was not compared in the experiments.

**B. Experiments on Synthetic Datasets**

1) **Experiment 1: Validity of Model and Indicator**

For visualization, we first consider a two-dimensional example, for which the features are simulated based on a Gaussian bivariate model and labels are generated with equal probability. We generate \( N = \{100, 200, 300\} \) random variate from Gaussian distribution, where \( N \) is the total number of the samples from the two classes \( y_i \in \{0, 1\} \). Therefore, the training data \( \{(x_i, y_i), i = 1, \ldots, N\} \) are simulated as follows:

\[
x_i | y = 1 \sim N(\mu, \Sigma) \quad \text{and} \quad x_i | y = -1 \sim N(-\mu, \Sigma),
\]

where \( \mu = [1, -2]^T \) and \( \Sigma = \text{diag}(0.5, 2) \). The data generated in this way has centralized features of each class, and the theoretical optimal linear classification boundary (Bayes classifier) of this example is known as

\[
x_2 = k_0 x_1 + q_0, \quad \text{where} \quad k_0 = 2 \quad \text{and} \quad q_0 = 0.
\]

Our goal is to estimate \( k \) and \( q \) for each classifier and compare them with the Bayes classifier (38). The six classifiers are compared via repeating the sampling and training 100 times of sample size \( N \) for which the linear decision rule is computed, i.e., \( (k_i, q_i) \) for all \( 1 \leq i \leq 100 \).

Each classifier is fairly compared against the Bayes classifier via the mean and variance of the slope \( \bar{k} \) and the intercept \( \bar{q} \). Further, to show the differences in the models more clearly, we additionally calculated the distance

\[
Dist_j = |\bar{k}_j - k_0|\bar{\sigma}_j + |\bar{q}_j - k_0|\bar{\sigma}_j
\]

for \( j \in \{\text{C-SVM, LSSVM, VSVM, IDLSSVM, } \varepsilon-L_1\text{SVM, } \varepsilon-L_1\text{VSVVM, } \varepsilon-L_1\text{VSVVM}\} \) to Bayes classifier, where \( \bar{k}_j(\bar{q}_j) \) and \( \bar{\sigma}_j(\bar{\sigma}_j) \) are the mean and standard deviation estimates of \( k(q) \) based on the 100 estimates of \( k(q) \). We set v-vector in Table 1 to version (36) and in Table 2 to version (37) and the V-matrix is set the same way, the purpose of which is to show the characteristics and advantages of different versions of v-vector and V-matrix.

The results are presented in Tables 1 and 2, where Bayesian classifier is the baseline and is noted on the caption of the table.

From the experimental results in Table 1 and 2, we yield the following observations:

i) We compare the performance of these six classifiers, using Bayesian decision as a baseline, and it is clear that with 12 results in these two tables, \( \varepsilon-L_1\text{VSVVM} \) has the best results in 10 out of 12 and is shown in bold. When comparing LSSVM with VSVM, VSVM is almost always better than LSSVM, and for \( \varepsilon-L_1\text{VSVVM} \) and \( \varepsilon-L_1\text{VSVVM} \) is also closer to the optimal decision.

ii) The amount of artificial data in the two tables are increasing in the order of \( \{100, 200, 300\} \), and it can be found that the performance of each model is improving as the amount of data increases. It is worth noting that when \( N=100 \) (small datasets), the performance of \( \varepsilon-L_1\text{VSVVM} \) and VSVM with v-value is more prominent than others.

iii) Taking Table 1 as an example to evaluate the performance difference between Vac and Acc, we find that the results of columns 2, 4, 6 are better than those of columns 1, 3, 5, which verifies the validity of Vac. Especially, the performance improvement for CSV, LSSVM, IDLSSVM and \( \varepsilon-L_1\text{SVM} \) are particularly significant compared to VSVVM and \( \varepsilon-L_1\text{VSVVM} \).

iv) The results in Table 1 are better compared to Table 2, especially the VSVVM and our model \( \varepsilon-L_1\text{VSVVM} \) have outstanding performance. Since the difference between the two tables is that the kernel functions \( G(\cdot) \) and \( \mu(x) \) are not the same and therefore capture the distribution information in different ways. It is obvious that (36) is more suitable for the artificial dataset of Gaussian distribution. The VSVVM is better for \( N=300 \) in Table 1 attributes to the fact that the estimation of v value is more accurate at this time and the V-matrix is a matrix compared to the v-vector in our model to get more comprehensive information.
| N   | 100         | 200         | 300         |
|-----|-------------|-------------|-------------|
|     | Indicator    | Acc         | Vac         | Acc         | Vac         | Acc         |
|     |             | 0.4560      | 0.2677      | 0.1960      | 0.1759      | 0.1806      | 0.1191      |
|     |             | 1.43±0.80   | 1.70±0.85   | 1.66±0.57   | 1.76±0.69   | 1.65±0.51   | 1.78±0.52   |
|     |             | -0.0001±0.32| -0.03±0.35  | 0.004±0.25  | -0.04±0.26  | 0.002±0.21  | -0.21±0.12  |
|     | LSSVM       | 0.3548      | 0.1839      | 0.1958      | 0.1253      | 0.1562      | 0.0946      |
|     |             | 1.35±0.54   | 1.73±0.65   | 1.53±0.41   | 1.70±0.41   | 1.63±0.42   | 1.76±0.40   |
|     |             | 0.02±0.17   | 0.03±0.19   | 0.026±0.13  | 0.005±0.13  | -0.01±0.11  | 0.009±0.12  |
|     | VSVN        | 0.0892      | 0.0836      | 0.0759      | 0.0433      | 0.0089      | 0.0057      |
|     |             | 1.86±0.64   | 1.87±0.58   | 1.82±0.40   | 1.90±0.42   | 2.03±0.30   | 2.02±0.32   |
|     |             | 0.01±0.21   | -0.05±0.20  | -0.02±0.13  | 0.02±0.15   | -0.007±0.12 | 0.008±0.12  |
|     | IDLSSVM     | 0.2741      | 0.1398      | 0.1302      | 0.0982      | 0.0890      | 0.0479      |
|     |             | 1.60±0.68   | 1.73±0.48   | 1.73±0.48   | 1.78±0.44   | 1.7510±0.35 | 1.85±0.307  |
|     |             | 0.01±0.21   | 0.04±0.21   | -0.009±0.22 | 0.02±0.17   | -0.01±0.12  | 0.002±0.12  |
|     | ε-L₁ SVSN   | 0.2812      | 0.0944      | 0.1140      | 0.0429      | 0.0735      | 0.0312      |
|     |             | 1.57±0.65   | 1.90±0.89   | 1.84±0.69   | 2.04±0.93   | 1.87±0.55   | 1.94±0.46   |
|     |             | 0.01±0.20   | 0.03±0.22   | 0.01±0.18   | -0.04±0.16  | -0.01±0.14  | -0.02±0.13  |
|     | ε-L₁ VSVM   | 0.0872      | 0.0645      | 0.0509      | 0.0294      | 0.0480      | 0.0263      |
|     |             | 1.88±0.72   | 1.93±0.81   | 1.93±0.70   | 1.95±0.59   | 1.90±0.48   | 1.96±0.68   |
|     |             | 0.02±0.20   | -0.02±0.21  | 0.06±0.15   | 0.01±0.18   | -0.01±0.12  | -0.014±0.12 |

In order to demonstrate the benefits of Vac more visually, Figure 1 shows the linear decisions based on Acc and Vac for the above six models as well as the Bayesian optimal decisions for Table 2. The green and red dots in the figure are the positive and negative class samples, the black line is the Bayesian optimal decision, the blue solid line is the decision using Vac as the evaluation indicator, and the blue dashed line is for Acc. Clearly, the model with Vac as the indicator performs better and our proposed model is the closest to the Bayesian decision among the six models in both cases.

2) EXPERIMENT 2: ROBUSTNESS TEST OF THE CLASSIFIERS
Since VSVM and ε-L₁VSVM are based on the Fredholm integral equation and approximate the conditional probability by estimating the distribution function. In the following, we show the stability of the solutions of the different classifiers as mentioned in [15] and [31]. Similar to the way...
FIGURE 1. Classification boundaries based on Acc (33) and Vac (34) as evaluation indicators for six classifiers with the v-vector of (37).

TABLE 3. Descriptions of 17 real datasets.

| Datasets        | ID | #of samples | #of features |
|-----------------|----|-------------|--------------|
| AustralianCAS   | D1 | 690         | 14           |
| Cleveland_HeartS| D2 | 270         | 13           |
| Cleveland10v2   | D3 | 201         | 13           |
| Cleveland10v3   | D4 | 200         | 13           |
| Cleveland10v4   | D5 | 178         | 13           |
| Cleveland10vr   | D6 | 303         | 13           |
| Echo            | D7 | 131         | 10           |
| Ecoli           | D8 | 336         | 7            |
| Ecoli10vr       | D9 | 336         | 7            |
| German          | D10| 1000        | 20           |
| Led7digit       | D11| 443         | 7            |
| Monks3          | D12| 121         | 6            |
| Monkst          | D13| 432         | 6            |
| WOBC            | D14| 699         | 9            |
| Yeast4vr        | D15| 1484        | 8            |
| Yeast456vr      | D16| 1484        | 8            |
| Abalone3vr      | D17| 4177        | 8            |

$\{(x_i, y_i), i = 1, \ldots, N\}$ are simulated as follows:

$$X_i|y = 1 \sim N (-3, 3) \quad \text{and} \quad X_i|y = -1 \sim N (3, 3). \quad (40)$$

In order to reflect the stability of the output results of each classifier, we use the rbf kernel function for nonlinear experiments. The experimental results are shown in Figure 2, where the black lines are $P(y = 1|x) = \frac{1}{1 + e^{-0.5x}}$. The horizontal coordinates are the values of the one-dimensional data and the data distribution is shown at the bottom of the figure, where red is the positive class and blue is the negative class; the vertical coordinates represent the probability values estimated by the classifier, and the blue image is a two-dimensional curve of the estimated results versus input variable for each nonlinear classifier. In addition, the v-values are used in version (37).

From Figure 2, we see that the classifiers based on Fredholm integral equation, like VSVM and $\varepsilon$-$L_1$ VSVM, they are more smooth and stable in their prediction. In contrast, the predictions by C-SVM, LSSVM, IDLSSVM, $\varepsilon$-$L_1$ SVM are less smooth. That is to say, the estimation of the $v$-value is more robust the traditional ones.

C. EXPERIMENTS ON BENCHMARK DATASETS

In this subsection, we conduct numerical experiments to show the effectiveness of the proposed $\varepsilon$-$L_1$ VSVM for some small benchmark datasets. We apply 6 classifiers to 17 UCI datasets summarized in Table 3, which could retrieve from UCI repository of Machine Learning Dataset.

In all the experiments, we normalize all feature vectors to be in [0,1] and the standard 10-fold cross validation is performed to select the optimal parameters according to selected evaluation indicator. To reflect the stability of the results, the following tables of experimental results record the mean and standard deviation of ten times 10-fold cross-validation results. Moreover, considering that the real data distribution

2https://archive.ics.uci.edu/ml/index.php
Table 4. Testing results on benchmark datasets for linear classifiers.

| ID  | CSVM G-means(Acc) (%) | LSSVM G-means(Acc) (%) | VSSVM G-means(Acc) (%) | e-L1SVM G-means(Acc) (%) | e-L1VSVM G-means(Acc) (%) |
|-----|-----------------------|------------------------|------------------------|--------------------------|---------------------------|
| D1  | 85.97±0.00(85.51)     | 86.33±0.16(85.90)      | 86.86±0.31(86.74)      | 86.50±0.46(86.12)        | 86.16±0.43(86.25)         |
| D2  | 82.84±0.42(83.33)     | 83.04±0.61(83.67)      | 83.39±0.85(84.00)      | 82.81±0.70(83.11)        | 82.63±0.66(83.03)         |
| D3  | 83.47±1.70(90.19)     | 82.24±1.81(91.34)      | 81.95±1.15(91.84)      | 81.20±1.06(91.74)        | 86.61±1.87(92.44)         |
| D4  | 88.53±1.22(93.45)     | 84.52±2.06(92.90)      | 86.14±2.08(93.40)      | 85.43±1.56(92.55)        | 88.48±1.79(93.60)         |
| D5  | 73.68±5.51(95.39)     | 68.62±7.89(95.00)      | 64.04±5.19(95.11)      | 60.50±2.74(94.78)        | 71.63±3.27(95.11)         |
| D6  | 83.65±0.56(84.12)     | 83.98±0.50(84.55)      | 84.11±0.53(84.65)      | 83.17±0.92(83.33)        | 82.72±0.42(82.90)         |
| D7  | 85.17±0.78(89.54)     | 82.51±2.34(88.77)      | 82.82±1.53(89.69)      | 85.70±0.68(90.61)        | 86.81±0.89(90.45)         |
| D8  | 75.39±2.28(92.32)     | 42.12±2.79(90.54)      | 57.24±2.20(91.36)      | 43.79±2.04(90.63)        | 73.35±2.01(91.78)         |
| D9  | 96.51±0.46(96.61)     | 96.65±0.20(96.67)      | 96.65±0.20(96.08)      | 96.37±0.29(96.49)        | 96.36±0.27(96.83)         |
| D10 | 63.07±0.60(75.74)     | 62.11±0.54(75.28)      | 62.10±0.52(75.61)      | 63.54±0.72(75.52)        | 69.60±0.30(76.40)         |
| D11 | 89.81±1.03(93.30)     | 0.00±0.00(91.65)       | 0.00±0.00(91.65)       | 43.31±5.03(92.33)        | 86.96±1.65(94.45)         |
| D12 | 76.87±1.70(77.02)     | 75.94±1.83(76.12)      | 76.45±1.98(76.45)      | 76.26±2.00(76.36)        | 77.39±1.76(77.44)         |
| D13 | 82.73±0.57(82.78)     | 78.09±0.54(79.88)      | 81.40±0.91(82.22)      | 81.69±0.66(81.69)        | 87.04±0.93(87.57)         |
| D14 | 96.38±0.33(96.67)     | 94.71±0.19(95.77)      | 94.90±0.23(95.91)      | 94.41±0.16(95.57)        | 97.10±0.17(96.97)         |
| D15 | 82.86±0.48(94.59)     | 58.38±0.33(92.38)      | 75.53±0.62(93.97)      | 64.00±0.85(92.97)        | 75.61±3.35(94.06)         |
| D16 | 84.10±0.30(92.86)     | 77.61±0.58(91.37)      | 81.24±0.24(92.06)      | 80.61±0.44(92.08)        | 84.36±0.71(92.82)         |
| D17 | 70.95±0.17(78.01)     | 69.06±0.17(77.31)      | 69.05±0.17(77.31)      | 70.56±0.17(77.55)        | 73.26±0.18(77.67)         |

Avg. 82.47 71.73 74.36 75.33 82.66 84.16
W/T/L 2/0/15 0/0/17 1/0/16 0/0/17 0/0/17 –

is unknown, the v-vector or v-matrix in the version of equation (37) is used for all the following experiments.

1) LINEAR RESULTS
Table 4 summarizes the linear classification results for all 17 datasets and the best G-means is shown in boldface. For each measure, the last two rows show the average G-means and the number of wins, draws and losses between our \( \varepsilon \)-L1 VSVM and other classifiers. Specially, we also give the corresponding Acc values to ensure the reasonableness of the G-means. It can be clearly seen that \( \varepsilon \)-L1 VSVM achieves the best performance in 14 out of 17. Grouping the results in the table for comparison, it is worth noting that \( \varepsilon \)-L1 VSVM has higher G-means compared to \( \varepsilon \)-L1 SVM in all seventeen datasets. Similarly, VSSVM performs better than LSSVM on most of the datasets. Overall, \( \varepsilon \)-L1 VSVM exhibit competitive advantage towards their competitor. Further, VSSVM and \( \varepsilon \)-L1 VSVM are the same type of model containing data location information. But \( \varepsilon \)-L1 VSVM using \( \varepsilon \)-insensitive loss, which is sparse and robust, and it perform better than VSSVM.

2) NONLINEAR RESULTS
The nonlinear classification results on the benchmark datasets are recorded in Table 5. Clearly, the overall performances of nonlinear \( \varepsilon \)-L1 VSVM are superior to all that of other nonlinear classifiers. Moreover, \( \varepsilon \)-L1 VSVM gains higher G-means than \( \varepsilon \)-L1 SVM on 16 datasets and VSSVM performs better than LSSVM on 12 datasets, which confirms the conclusion above further. From another point of view, our model is similar to a weighted model, which has a v-value weight and represents the location information of the data, while the contrasted IDLSSVM is a density-weighted model that also tries to reflect the distribution of the data. However, IDLSSVM uses KNN to calculate the samples in the domain and thus converts the corresponding density, which focuses on the situation within the local area, while \( \varepsilon \)-L1 VSVM considers a relative position of each sample in the whole data and highlights a global concept.

Furthermore, in the synthetic datasets experiments, we show the effectiveness of Vac as an evaluation metric. Now, we conduct experiments using Vac on real data in the following. Considering the space of the paper, we select nine smallest datasets as examples for validation since the artificial data experiments reflects the advantage of \( \varepsilon \)-L1 VSVM and Vac on small data. Figure 3 and Figure 4 show the classification results for each model on the selected nine small datasets, respectively, where the red line is a line graph of the results with Vac as the evaluation indicator and the blue is for Acc.

3) LINEAR RESULTS AND NON-LINEAR RESULTS OF Vac AS AN INDICATOR
From the numerical results labeled in the figure, it is clear that our proposed model is still more advantageous compared to the other models. Whether Acc or Vac is used as the indicator for selecting the parameters (red or blue line), the results of our model are in most cases the peak of the line graph. For the red line, for both linear and nonlinear results, our model is the best in 6 out of 9.

From another perspective, for these nine small datasets, the red line is above the blue one in most cases, which indicates that Vac, as a more rigorous evaluation metric of the same
TABLE 5. Testing results on benchmark datasets for nonlinear(rbf) classifiers.

| ID | CSVM | LSSVM | VSVM | IDLSSVM | ε-L1 SVM | ε-L1 VSVM |
|----|------|-------|------|---------|--------|--------|
|    | G-means(Acc)(%) | G-means(Acc)(%) | G-means(Acc)(%) | G-means(Acc)(%) | G-means(Acc)(%) | G-means(Acc)(%) |
| D1 | 85.67±0.28(85.67) | 85.61±0.60(85.70) | 85.68±0.38(85.52) | 86.54±0.41(86.39) | 84.17±2.43(84.74) |
| D2 | 82.45±0.74(82.93) | 82.27±0.58(83.11) | 83.40±0.66(83.89) | 82.81±0.70(83.11) | 82.63±0.60(82.96) |
| D3 | 82.55±3.22(91.49) | 72.75±1.75(88.56) | 83.36±0.90(92.19) | 81.58±1.15(91.84) | 81.93±1.12(91.79) |
| D4 | 87.60±1.83(93.40) | 79.82±1.45(91.50) | 85.73±1.62(93.35) | 85.43±1.56(92.55) | 85.31±2.19(92.35) |
| D5 | 75.12±3.61(95.05) | 60.29±3.99(94.33) | 78.39±1.52(96.01) | 64.83±5.06(95.67) | 65.18±5.01(95.06) |
| D6 | 83.71±0.54(84.22) | 82.33±0.69(82.94) | 84.22±0.56(84.68) | 82.85±0.70(82.97) | 82.55±0.78(83.17) |
| D7 | 86.46±1.36(89.47) | 83.63±1.42(88.86) | 85.88±0.77(90.00) | 84.46±3.11(90.15) | 84.27±1.23(89.85) |
| D8 | 78.47±0.73(93.57) | 75.91±1.70(93.09) | 73.53±1.34(93.18) | 77.29±1.51(93.54) | 73.59±1.34(92.92) |
| D9 | 96.73±0.30(96.75) | 96.83±0.18(96.84) | 97.26±0.24(97.20) | 96.47±0.51(96.55) | 97.10±0.25(97.14) |
| D10| 62.23±0.58(76.46) | 66.38±0.81(76.98) | 64.31±0.82(76.39) | 64.02±0.98(76.42) | 64.85±1.00(76.43) |
| D11| 88.52±0.79(96.73) | 87.90±1.67(96.64) | 71.04±0.00(94.36) | 90.65±0.09(96.88) | 88.35±1.27(96.68) |
| D12| 87.31±1.82(87.36) | 88.70±1.32(88.76) | 84.62±0.59(85.18) | 88.69±1.04(88.68) | 90.24±1.72(90.33) |
| D13| 99.10±0.82(99.12) | 96.57±0.51(96.50) | 97.40±0.46(97.53) | 97.12±0.26(97.01) | 97.31±0.07(97.20) |
| D14| 96.94±0.16(96.82) | 96.63±0.16(96.70) | 96.88±0.36(96.87) | 96.33±0.27(96.55) | 96.79±0.19(96.82) |
| D15| 85.09±0.27(95.51) | 82.31±0.46(95.10) | 82.20±0.34(95.24) | 84.09±0.52(92.92) | 86.82±0.47(95.80) |
| D16| 86.77±0.51(93.07) | 84.19±0.54(92.67) | 84.62±0.29(92.83) | 82.32±0.57(95.05) | 85.10±0.26(93.09) |
| D17| 74.59±0.14(79.37) | 74.90±0.18(79.35) | 73.27±0.14(78.71) | 73.62±0.10(79.13) | 75.63±0.17(78.96) |

Avg 84.67 82.18 83.05 83.48 83.64 86.69

W/D/L 1/0/16 0/0/17 0/0/17 1/0/16 1/0/16 -

FIGURE 3. Linear results based on Acc (33) and Vac (34) as evaluation indicators for six classifiers on nine small datasets.

FIGURE 4. Nonlinear results based on Acc (33) and Vac (34) as evaluation indicators for six classifiers on nine small datasets.

type as Acc but with additional consideration of distribution, can improve the performance of the trained models. Further, for models like CSVM, LSSVM, IDLSSVM and ε-L1 SVM, the effects shown in Figure 3 and Figure 4 are more significant. However, for ε-L1 VSVM and VSVM, the performance improvement is not obvious, and even decreases on some datasets.

The following conclusions can be drawn for the results of the above synthetic datasets and benchmark datasets:

i) Both for synthetic datasets and benchmark datasets, our model shows a superiority on small datasets. For example, ε-L1 VSVM significantly outperforms other classifiers for artificial data of N=100, while the advantage is relatively smaller at N=300. This is because when the sample size is not sufficient, our model can capture the distribution structure of the data and obtain more location information which cannot be directly extracted by classical classifiers, thus classifying more accurately.
ii) Different versions of v-vectors have a large impact on the performance of the model and evaluation metrics. Two kinds of v-vector have been proposed, and their expressions are (36) and (37). When the data distribution is known to be Gaussian, (36) is more recommended because its \( \mu(x) \) is estimated based on the Gaussian distribution, but when the data distribution is unknown, (37) may be better. The follow-up on the calculation of v-vector needs to be studied in depth.

iii) For models that do not contain data distribution information in the objective function, such as CSVM, LSSVM, IDLSSVM, \( \varepsilon-L_1 \) SVM, they are significantly improved the performance of their models when Vac is used as an evaluation metric. Therefore, using Vac as an evaluation metric is another way to add a priori information such as data distribution to the model. For both VSVM and \( \varepsilon-L_1 \) V SVM models that inherently have distribution information, Vac gives a relatively much smaller improvement in model performance.

VI. CONCLUSION

In this paper, we study the expected risk estimation principle and corresponding empirical risk estimation by estimating the conditional probability function of the data via Fredholm equation. The corresponding classification models and the classification evaluation indicators under this paradigm are different from the traditional one. Experimental results show the effectiveness of the proposed \( \varepsilon-L_1 \) SVM and the proposed indicator on validity and interpretability of classification, especially in case of insufficient training data. It is worth mentioning that the risk measurement space in proposed paradigm is different from the traditional one, so further study the more appropriate loss measurement is our main research directions in the future. In addition, extend the paradigm by estimating the conditional probability function of the data via Fredholm equation to different statistical inference problems are also promising.

APPENDIX A

PROOF of THEOREM 3.1

Proof: We first consider the distributions defined by the Dirac function and the commonly used distributions (with the normal distribution as an example). The Dirac function [35], [36] has a value at only one point and the integral over the domain of definition is 1. Its definition is as follow

\[
\delta(x - x_0) = \begin{cases} 
+\infty, & \text{if } x = x_0, \\
0, & \text{if } x \neq x_0.
\end{cases}
\]

and its corresponding cumulative distribution function is

\[
F(x) = \theta(x - x_0) = \begin{cases} 
1, & \text{if } x \geq x_0, \\
0, & \text{otherwise}.
\end{cases}
\]

Denote \( L_{\text{left}} = \int_{\mathcal{X} \times \mathcal{Y}} G(x - x') \delta(x') dF(x', y') \). Since the integrated function is independent of \( y \), \( F(x, \infty) = F_y(x) \) and we have

\[
L_{\text{left}} = \int_{\mathcal{X}} G(x - x') \delta(x') dF(x').
\]  

Put \( F(x) = \theta(x - x_0) \) into (41), then

\[
L_{\text{left}} = \int_{\mathcal{X}} G(x - x') \delta(x') dF(x') = \int_{\mathcal{X}} (x - x') \delta(x' - x) r(x') dx' = (x - x) r(x)
\]

\[
= r(x).
\]

Similarly, for \( F(x, y) = \theta(x - x_0) \theta(y - y_0) \), \( L_{\text{right}} = \int_{\mathcal{X} \times \mathcal{Y}} G(x - x') y' dF(x', y') = y \). Therefore, \( R(x) \) in (3) is a special case of (17) and \( K_{\text{FVw}}(x) \) is a more general expression of expected risk. □

REFERENCES

[1] A. K. Jain, R. P. W. Duin, and J. C. Mao, “Statistical pattern recognition: A review,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 1, pp. 4–37, Jan. 2000.

[2] V. Vapnik, “SVM method of estimating density, conditional probability, and conditional density,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), vol. 2, May 2000, pp. 749–752.

[3] V. Vapnik, The Nature of Statistical Learning Theory, Springer, 1999.

[4] V. Vapnik and R. Izmailov, “V-matrix method of solving statistical inference problems,” J. Mach. Learn. Res., vol. 16, no. 2015, pp. 1683–1730, 2015.

[5] S. Theodoridis and K. Koutroumbas, Pattern Recognition, Amsterdam, The Netherlands: Elsevier, 2006.

[6] Y. H. Shao, C. H. Zhang, X. B. Wang, and N. Y. Deng, “Improvements on twin support vector machines,” IEEE Trans. Neural Netw., vol. 22, no. 6, pp. 962–968, May 2011.

[7] K. Fukunaga, Introduction to Statistical Pattern Recognition, Amsterdam, The Netherlands: Elsevier, 2013.

[8] V. Vapnik, “Principles of risk minimization for learning theory,” in Proc. Adv. Neural Inf. Process. Syst., vol. 4, 1991, pp. 1–8.

[9] J. A. K. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” Neural Process. Lett., vol. 9, no. 3, pp. 293–300, Jun. 1999.

[10] C. M. Bishop, Neural Networks for Pattern Recognition, New York, NY, USA: Oxford Univ. Press, 1995.

[11] C. Cortes and V. Vapnik, “Support-vector networks,” Mach. Learn., vol. 20, no. 3, pp. 273–297, Jul. 1995.

[12] N. Y. Deng, Y. J. Tian, and C. H. Zhang, Support Vector Machines: Optimization Based Theory, Algorithms, and Extensions. Boca Raton, FL, USA: CRC Press, 2012.

[13] Z. H. Zhou, Ensemble Methods: Foundations and Algorithms. Boca Raton, FL, USA: CRC Press, 2012.

[14] V. Vapnik and R. Izmailov, “Rethinking statistical learning theory: Learning using statistical invariants,” Mach. Learn., vol. 108, no. 3, pp. 381–423, Mar. 2019.

[15] V. Vapnik and R. Izmailov, “Complete statistical theory of learning: Learning using statistical invariants,” in Proc. 9th Symp. Conformal Probabilistic Predict. Appl., 2020, pp. 4–40.

[16] V. Cherkassky, H.-H. Chen, and H.-T. Shiao, “Group learning for high-dimensional sparse data,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2019, pp. 1–10.

[17] A. Mottaghi, M. A. Khanjari, R. Alizadeh, and H. Maghsoudi, “Prediction of affinity coefficient for estimation of VOC adsorption on activated carbon using V-matrix regression method,” Adsorption, vol. 27, no. 6, pp. 963–978, Aug. 2021.

[18] C. Lu, Y. Wu, J. Miguel Hernández-Lobato, and B. Schölkopf, “Nonlinear invariant risk minimization: A causal approach,” 2021, arXiv:2102.12353.

[19] H. Zhang, A. U. Zaman Robin, J. Zhu, C. Lu, C. Fang, and I. Shyha, “Approach to breech face impression comparison based on the robust estimation of a correspondence function,” Forensic Sci. Int., vol. 333, Apr. 2022, Art. no. 111229.
MENG-XIAN ZHU is currently pursuing the master’s degree with the School of Economics, Hainan University, Haikou, China. Her research interests include machine learning and statistical inference.

YUAN-HAI SHAO received the bachelor’s degree from the College of Mathematics, Jilin University, in 2006, and the Ph.D. degree from the College of Science, China Agricultural University, China, in 2011. He is currently a Professor with the School of Management, Hainan University. He has published more than 100 refereed papers. His research interests include optimization methods, machine learning, and data mining.

***