One-class SVM Probabilistic Outputs

Zhongyi Que and Chih-Jen Lin, *Fellow, IEEE*

***Abstract*—One-class SVM is an extension of SVM to handle unlabeled data. As a mature technique for outlier detection, one-class SVM has been widely used in many applications. However, similar to standard two-class SVM, the design of one-class SVM does not give probabilistic outputs. For two-class SVM, some methods have been proposed to effectively obtain probabilistic outputs, but due to the difficulty of no label information, less attention has been paid on one-class SVM. Our aim in this work is to propose some practically viable techniques to generate probabilistic outputs for one-class SVM. We investigate existing methods for two-class SVM and explain why they may not be suitable for one-class SVM. Due to the lack of label information, we think a feasible setting is to have probabilities mimic to the decision values of training data. Based on this principle, we propose several new methods. Detailed experiments on both artificial and real-world data demonstrate the effectiveness of the proposed methods.**

***Index Terms*—Probability estimation, One-class SVM, Platt scaling, Outlier detection.**

I. Introduction

One-class SVM [1] is an extension of SVM to handle unlabeled data. As a mature technique for outlier detection, one-class SVM has been widely used in many applications. However, similar to standard two-class SVM, the design of one-class SVM does not give probabilistic outputs. For two-class SVM, some methods have been proposed to effectively obtain probabilistic outputs, but less attention has been paid on one-class SVM. The reason is apparently due to the lack of label information. Our aim in this work is to propose some practically viable techniques to generate probabilistic outputs for one-class SVM.

We consider training vectors $\boldsymbol{x}_i \in R^n$, $i = 1, \ldots, l$, where l is the number of observations and n is the number of features. The optimization problem of one-class SVM is as follows.

$$\begin{aligned} \min_{\boldsymbol{w},\xi,\rho} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \rho + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i \qquad (1) \\ \text{subject to} \quad & \boldsymbol{w}^T\phi(\boldsymbol{x}_i) \geq \rho - \xi_i, \\ & \xi_i \geq 0, i = 1, \ldots, l, \end{aligned}$$

where $\nu \in (0, 1)$ is a given parameter and $\phi(\boldsymbol{x}_i)$ is a function to map $\boldsymbol{x}_i$ into a vector of higher dimensionality. Similar to the standard SVM, to address the possible high dimensionality of $\phi(\boldsymbol{x}_i)$, we can prove that the solution $\boldsymbol{w}$ of (1) is a linear combination of all $\phi(\boldsymbol{x}_i)$ with coefficients $\boldsymbol{\alpha}$:

$$\boldsymbol{w} = \sum_{i=1}^{l}\alpha_i\phi(\boldsymbol{x}_i)$$

and solve a dual problem with the variable $\boldsymbol{\alpha}$. Then all the calculation can rely on using kernel operations.

$$K(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^T\phi(\boldsymbol{x}').$$

After (1) is solved, the decision value of one-class SVM is

$$\begin{aligned} f(\boldsymbol{x}) &= \boldsymbol{w}^T\phi(\boldsymbol{x}) - \rho \qquad (2) \\ &= \sum_{i=1}^{l}\alpha_i\phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}) - \rho, \end{aligned}$$

and a negative decision value leads to the prediction of an outlier. The parameter ν is very essential in one-class SVM because it is shown to be an upper bound on the number of training instances considered as outliers [2].

The above setting leads to only a decision value of an instance but not the probability as an outlier. This situation is similar to two-class SVM, which does not give probabilistic outputs. By using decision values, some past works have successfully developed methods to predict an instance's probability to be in each class (e.g., [3]–[6]). Among them, Platt scaling [3] is an effective technique by assuming the following probability model.

$$P(y = 1|\boldsymbol{x}) \approx \frac{1}{1 + \exp(Af + B)}, \qquad (3)$$

where $y = \pm 1$ is the class label, f is the decision value $f(\boldsymbol{x})$ of the instance $\boldsymbol{x}$, and A, B are the parameters to be decided from the training data.

In contrast to the situation for two-class SVM, the probabilistic outputs of one-class SVM are less studied. The very few existing works often assume the availability of some label information. For example, [7] assumes that normal data instances are known, while the set of outliers is either empty or under-represented. Some works (e.g., [8]–[10]) apply one-class SVM to handle unbalanced data in two classes. In such a situation, methods for two-class probabilistic outputs may be applicable on the decision values obtained by one-class SVM. On the other hand, works such as [9], [11]–[13] proposed combining several one-class SVM problems as a more sophisticated optimization problem. Then through the several obtained decision boundaries, we can predict probabilistic outputs. Different from existing works, this study focuses on the fully unsupervised setting, where no or very little (e.g. ratio of outliers) label information is available. Further, we aim at a simple setting for obtaining probabilities after solving

Zhongyi Que and Chih-Jen Lin are with Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 106 e-mail: r08922152@ntu.edu.tw and cjlin@csie.ntu.edu.tw. Chih-Jen Lin is also with Department of Machine Learning, Mohamed Bin Zayed University of Artificial Intelligence. This work was supported in part by MOST of Taiwan grant 110-2221-E-002-115-MY3.

(1), so sophisticated modifications of (1) are not within our consideration.

In this work, we broadly check possible approaches and develop feasible methods for generating probabilistic outputs. To begin, although Platt scaling is an outstanding probability estimation method for two-class classification, through theoretical analysis and experiments, we explain that it is not appropriate for one-class SVM. The main reasons are first the lack of label information, and second different scenarios of data distributions for two-class and one-class SVM. We then check other methods such as isotonic regression, kNN and EM-based algorithms, which have been applied to two-class SVM. Unfortunately, they are not suitable for one-class SVM for the reason of lacking labels. We finally think the best we can do is to have probabilities mimic to the decision values of training data. To this end, we propose two methods, where one is non-parametric and the other is parametric. All the methods are evaluated on artificial data and real-world data. Experiments demonstrate the effectiveness of the proposed methods.

This paper is organized as follows. In Section II, we review how Platt scaling gives the probability of the two-class SVM and discuss why it is not suitable for one-class SVM's probability estimation. In Section III, we investigate several parametric or non-parametric methods and discuss if they are applicable to one-class SVM. We propose new methods in Section IV. In Section V, we present detailed experiments and analysis. Section VI is the conclusion. Supplementary materials and programs for experiments are available at https://www.csie.ntu.edu.tw/~cjlin/papers/oneclass_prob/. This paper is an extension of the first author's master thesis [14].

One of the proposed methods has been incorporated into the popular package LIBSVM [15] for support vector machines (version 3.3 and after).

II. Issues in Extending Platt's Two-class SVM Probabilistic Outputs to One-class SVM

We begin with illustrating why the popular approach by Platt [3] is suitable for two-class SVM. This discussion is then extended in Section II-C and Section II-D to explain why the same approach may not be suitable for one-class SVM.

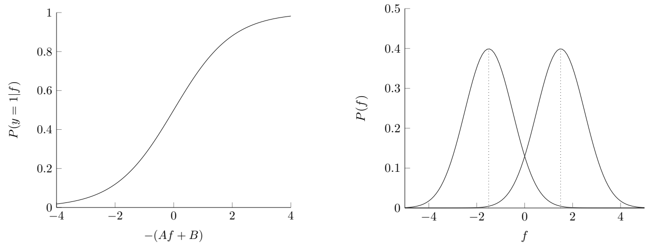
A. A Review of the Approach by Platt

The idea of Platt scaling is that the model in (3) as a function of $-(Af+B)$ is a sigmoid function illustrated in Figure 1a. It satisfies the following property

$$P(y=1|\boldsymbol{x}) \begin{cases} \to 1 & \text{if } -(Af+B) \to \infty, \\ = 0.5 & \text{if } -(Af+B) = 0, \\ \to 0 & \text{if } -(Af+B) \to -\infty. \end{cases}$$

To identify the parameters A and B, the standard way is to minimize the negative log-likelihood

$$\min_{A,B} \quad -\sum_{i=1}^{l} \log(P(y_i|\boldsymbol{x}_i)), \tag{4}$$



(a) Platt scaling probability model.

(b) An assumed distribution of decision values.

Fig. 1: The left figure is an example of the probability model in (3). The right figure is an example assuming that decision values of two classes of data follow (10). If $P(f|y=-1)$ and $P(f|y=1)$ are like the right figure, then the probabilistic output is in a shape shown in the left figure.

where $y_i \in \{-1,1\}$ is the class label of $\boldsymbol{x}_i$. To avoid overfitting the training data, in Platt scaling [3] a revised problem is solved:

$$\min_{A,B} \quad -\sum_{i=1}^{l}(t_i \log(P(y_i=1|\boldsymbol{x}_i)) + (1-t_i)\log(1-P(y_i=1|\boldsymbol{x}_i))), \tag{5}$$

where

$$t_i = \begin{cases} \frac{N_+ + 1}{N_+ + 2} & \text{if } y_i = +1 \\ \frac{1}{N_- + 2} & \text{if } y_i = -1 \end{cases}, i = 1, ..., l, \tag{6}$$

and N_+ and N_- are the numbers of positive data and negative data respectively.

B. Why Fitting a Sigmoid Function is Appropriate for Two-class Problems

We explain why a sigmoid function is a reasonable choice in the two-class scenario, where part of our derivation follows [10]. A binary classifier is often designed in a way of assuming that the distributions of the two classes are two clusters on both sides of the decision value $f = 0$; see an illustration in Figure 1b. For example, in two-class SVM, positive and negative data are assumed to have their corresponding decision values satisfying $f \geq 0$ and $f < 0$, respectively.

By Bayes' Theorem, the probability of a sample $\boldsymbol{x}$ with decision value f in the class $y=1$ is

$$\begin{aligned} & P(y=1|f) \qquad (7) \\ = & \frac{P(f|y=1)P(y=1)}{P(f)} \\ = & \frac{P(f|y=1)P(y=1)}{P(f|y=1)P(y=1)+P(f|y=-1)P(y=-1)} \\ = & \frac{1}{1+e^{-z}}, \qquad (8) \end{aligned}$$

where

$$\begin{aligned} z &= \log \frac{P(f|y=1)P(y=1)}{P(f|y=-1)P(y=-1)} \\ &= \log \frac{P(f|y=1)}{P(f|y=-1)} + \log \frac{P(y=1)}{P(y=-1)} \\ &= \log \frac{P(f|y=1)}{P(f|y=-1)} + \text{a constant}. \end{aligned} \tag{9}$$

Here the logarithmic ratio between $P(y = 1)$ and $P(y = -1)$ is a constant.

Consider the example in Figure 1b, where the decision values of the two classes of data follow Gaussian distributions with the following probability density functions.

$$\begin{aligned} P(f|y=1) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(f-1.5)^2}, \\ P(f|y=-1) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(f+1.5)^2}. \end{aligned} \tag{10}$$

Then (7) and (9) imply that

$$P(y=1|f) = \frac{1}{1+e^{-3f-(\text{a constant})}},$$

which is a special case of (3). This example roughly explains the validity of assuming the probability model in (3).

Note that our derivation is not formal because for a continuous random variable, we have $P(f) = P(f|y = 1) = 0$. However, the result is the same if we consider a formal setting $\lim_{\Delta f \to +0} P(f \le F \le f + \Delta f | y = 1)$, where F is the random variable of decision values.

C. Lack of Labels in Maximizing the Likelihood

For one-class SVM, a serious problem is that no true labels are available for constructing the optimization problem in (4). What we can do is to solve a one-class problem first and then treat the predicted labels as the true labels. However, we will show that this setting cannot effectively produce probabilistic outputs.

Specifically, if predicted labels are used in the optimization problem (4), the following perfect situation holds.

$$y_i = 1 \quad \text{if and only if} \quad f_i \ge 0. \tag{11}$$

This property and the maximization of the likelihood cause that the resulting model satisfies $P(y = 1|f) \to 1$ as soon as f is changed from zero to positive, especially when the size of training data is large. Even with the technique in (6) to avoid overfitting, the resulting sigmoid curve is still very close to a 0-1 function shown in Figure 2.

Thus, Platt scaling cannot give a good probability estimation without true labels. This issue has been mentioned in other works such as [10]. In fact, many other methods of probabilistic outputs suffer the same issue if the situation in (11) holds. We will show more examples in Section IV.

D. Fitting a Sigmoid Function may not be Suitable for One-class SVM

Besides the issue discussed in Section II-C, another issue is that, under certain data scenarios, the probability model of one-class SVM may not be resemble to a sigmoid function.

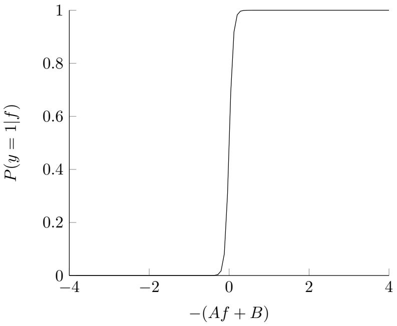
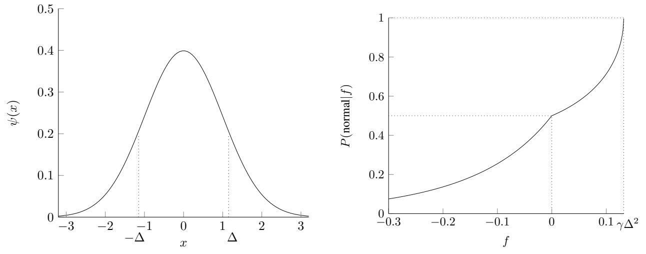


Fig. 2: An illustration of the probability model if (11) holds.



(a) An assumed distribution of data.

(b) $P(\text{normal}|f)$.

Fig. 3: A conceptual illustration of one-class SVM probabilistic outputs. Assume data are generated by a standard normal distribution as in the left figure. Data not in $[-\Delta, \Delta]$ are considered as outliers. Then the curve of $P(\text{normal}|f)$ shapes like the right figure, which uses (22) with $\Delta \approx 1.15$ leading to $P[-\Delta \le X \le \Delta] = 0.75$.

The goal of one-class SVM is to separate the majority of data from others, where these two groups of data my occur in many different ways. For example, the work in [10] empirically finds that, for certain problems, $P(f|\boldsymbol{x}$ is a normal point) follows an exponential distribution, while $P(f|\boldsymbol{x}$ is an outlier) still follows a normal distribution. They then construct a likelihood function to obtain parameters of the two probability models. In this subsection we analyze one-class SVM by considering another scenario of the data.

From the viewpoint of outlier detection, we may think that data are generated from the same distribution and those occurred in the tail are considered as outliers [2]. By considering the RBF kernel, we derive the analytic form of the model $P(\text{normal}|\boldsymbol{x})$ to directly show that it is not a sigmoid function of the decision value f.

Now consider a simplified scenario so that each instance is a one-dimensional point generated by a Gaussian distribution with mean 0 and variance 1. Thus the probability density function is

$$\psi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}, \tag{12}$$

which is shown in Figure 3a. Naturally, values too large or too small can be considered as outliers. We further assume that our one-class SVM has successfully identified $(-\infty, -\Delta) \cup$

(Δ, ∞) as the range of outliers. Then the decision function of one-class SVM should satisfy

$$f(x) = \begin{cases} \geq 0 & \text{if } x \in [-\Delta, \Delta], \\ < 0 & \text{otherwise.} \end{cases} \tag{13}$$

Because the model $P(\text{normal}|x)$ should satisfy

$$P(\text{normal}|x) = \begin{cases} 1 & \text{if } x = 0, \\ 0 & \text{if } x \to \pm\infty, \end{cases}$$

we can consider

$$P(\text{normal}|x) \equiv P(X \leq -|x| \text{ or } X \geq |x|), \tag{14}$$

where X is the random variable in generating x. That is, in our example X has the density function in (12). Then

$$P(X \leq -|x| \text{ or } X \geq |x|) = 1 + \text{erf}(\frac{-|x|}{\sqrt{2}}), \tag{15}$$

where erf$(\cdot)$ is the so called error function in mathematics and (15) serves as its definition. From (13), the probabilistic output should satisfy

$$P(\text{normal}|x = \Delta) = 0.5. \tag{16}$$

To this end, if

$$\bar{\Delta} \equiv 1 + \text{erf}(\frac{-\Delta}{\sqrt{2}}), \tag{17}$$

the value in (15) should be scaled so the probabilistic output $P(\text{normal}|x)$ is

$$\begin{cases} (1 + \text{erf}(\frac{-|x|}{\sqrt{2}}) - \bar{\Delta}) \times \frac{0.5}{1-\bar{\Delta}} + 0.5 & \text{if } x \in [-\Delta, \Delta], \\ (1 + \text{erf}(\frac{-|x|}{\sqrt{2}})) \times \frac{0.5}{\bar{\Delta}} & \text{otherwise.} \end{cases} \tag{18}$$

From (17), $\bar{\Delta}$ is the ratio of data as outliers. Therefore, after training one-class SVM, we can set $\bar{\Delta}$ to be the ratio of training data considered as outliers. Alternatively, because ν is an upper bound of this ratio of training data, we can set

$$\bar{\Delta} = \nu. \tag{19}$$

By considering each instance to be one-dimensional, we have discussed how to get the ideal probabilistic outputs in (18) from an assumed data distribution. However, an instance may have multiple features so in practice decision values instead of the original data are often used to generate the probabilistic output. Thus we need to investigate the relation between the decision values in one-class SVM and the original data.

By considering the most commonly used RBF kernel:

$$K(\boldsymbol{x}, \boldsymbol{x}') = e^{-\gamma\|\boldsymbol{x}-\boldsymbol{x}'\|^2}, \tag{20}$$

where $\gamma > 0$ is the kernel parameter, in supplementary materials we show that if γ is small, the decision boundary of the one-class SVM is nearly a hyper-sphere and the data points outside the sphere are outliers, a situation similar to the Support Vector Data Description (SVDD) [16]. Specifically, the decision value can be approximated by

$$f(\boldsymbol{x}) \approx -\gamma\|\boldsymbol{x} - \bar{\boldsymbol{x}}\|^2 + C, \tag{21}$$

where C is a positive constant and $\bar{\boldsymbol{x}}$ is the mean of training data. We are able to derive the probabilistic output and the decision value $P(\text{normal}|f)$ as follows.

$$\begin{cases} (1 + \text{erf}(\frac{-\sqrt{\Delta^2 - f/\gamma}}{\sqrt{2}}) - \nu) \times \frac{0.5}{1-\nu} + 0.5 & \text{if } x \in [-\Delta, \Delta], \\ (1 + \text{erf}(\frac{-\sqrt{\Delta^2 - f/\gamma}}{\sqrt{2}})) \times \frac{0.5}{\nu} & \text{otherwise.} \end{cases} \tag{22}$$

By considering $\gamma = 0.1$ and $\nu = 0.25$ the curve is like that in Figure 3b.

From (22) and Figure 3b, clearly the curve of $P(\text{normal}|f)$ is very different from a sigmoid function in Figure 1a. First, the range of valid f is no longer $(-\infty, \infty)$. Instead, the valid range is now $(-\infty, \gamma\Delta^2]$. Second, for the curve in Figure 1a, as $-(Af + B) \to \infty$, a negative curvature leads to a flattened curve. In contrast, the function in (22) always has a positive curvature even as $f \to \gamma\Delta^2$. Therefore, fitting the sigmoid model by minimizing the problem in (4) cannot give an accurate probability estimate.

Recall we consider $P(\text{normal}|f)$ instead of $P(\text{normal}|\boldsymbol{x})$ because $\boldsymbol{x}$ may be multi-dimensional. All the above analysis on one-dimensional data can be extended though we leave details in supplementary materials.

In Appendix, we give a similar illustration by assuming that data are generated under a distribution other than (12).

In summary, because one-class SVM may deal with data from the same distribution, the class probability may not follow the form in (3) by Platt's method for two-class classification. We will conduct experiments in Section V and supplementary materials to confirm the points made in this section.

III. Other Existing Probability Estimation Methods

In Section II, we discuss why Platt scaling, an appropriate probability estimation method for two-class classification problems, is not a good choice for one-class SVM. In this section, we check other existing parametric or non-parametric settings for generating probabilistic outputs. Most require the availability of true labels and are not applicable for one-class SVM probabilistic outputs. We then discuss some works that attempt to address this issue.

A. Isotonic Regression

Isotonic regression is a non-parametric method for probability estimation [17]. Assume that instances have been sorted according to the decision values such that

$$f_i \leq f_{i+1}, \quad i = 1, ..., l-1.$$

The idea is that if the ranking by decision values is correct, then a non-decreasing mapping of probabilities should be obtained. To this end, this method obtains the probabilistic output p_i by solving a least square problem.

$$\begin{aligned} \min_{\boldsymbol{p}} \quad & \sum_{i=1}^{l} (t_i - p_i)^2, \\ \text{subject to} \quad & p_{i+1} \geq p_i, i = 1, \ldots, l-1, \end{aligned} \tag{23}$$

where

$$t_i = \begin{cases} 1 & \text{if } y_i = 1, \\ 0 & \text{if } y_i = -1. \end{cases}$$

A common method to solve the optimization problem (23) is through the pair-adjacent violators (PAV) algorithm [18], though here we do not get into details.

Unfortunately, this method is not applicable to one-class SVM, which is an unsupervised learning model with no true labels. When the isotonic regression is applied, only predicted labels are available to generate the target values $t_i, \forall i$. From (11), all of the data with non-negative decision values have the corresponding target value 1, while others have 0. Since there is no interleaving of 0 and 1 in the sequence of scores, the result of isotonic regression is a 0-1 function which does not give meaningful probabilistic outputs.

B. The Method of k Nearest Neighbors

The method of k nearest neighbors (kNN) is a widely used non-parametric approach for classification. It can be extended to give probability estimation.

For a coming test point, the algorithm firstly finds the k nearest neighbors in the training data. Here k is a user-defined positive integer. Then it determines which class the test point belongs to by the majority voting among those k nearest neighbors. For the probability estimation problem, the test point can directly take the ratio of the target class samples in the k nearest neighbors as the probability. Therefore, the probabilistic output strongly relies on the availability of true labels.

With the absence of true labels, we use predicted labels as the replacement in one-class SVM. For any coming instance, unless the decision value is very close to 0, this instance and almost all its neighbors have the same label. Therefore, if the parameter k is not too large, we tend to obtain a 0-1 function.

Another issue is the selection of the parameter k. In two-class classification, k is often chosen by a cross-validation procedure. This is not possible for one-class SVM because true labels are not available.

C. Modeling Sigmoid Probability Using an EM-based Algorithm

To address the issue of lacking labels, [10] proposed an expectation-maximization (EM) algorithm to fit a sigmoid function like Platt did, but also learn the labels simultaneously.

In this EM-based algorithm, the core target is still to minimize the negative log-likelihood shown in (5). However, we treat the missing labels t_i as hidden variables. Thus the missing labels and the unknown parameters (A, B) in the probability model (3) are simultaneously solved by an iterative procedure in the EM algorithm. At the sth iteration the procedure involves an E-step and an M-step. First, with current parameter $\theta^s = (A^s, B^s)$, we assign the expected values to the missing labels $T^s = \{t_i^s \mid i = 1, ..., l\}$. Second, a new parameter θ^{s+1} is computed by minimizing the negative log-likelihood given the values of T^s. A pseudocode of the EM algorithm is shown in Algorithm 1.

Algorithm 1 EM algorithm to model the sigmoid probability [10]

```
Require: The set of decision values $F$ in (24).
Ensure: Model parameters, $\theta = (A, B)$.
  $s \leftarrow 0$.
  Initialize the parameters to $\theta^0$.
  repeat
    E-step: Set $t_i^s$ = $\mathrm{E}(t_i|F, \theta^s)$.
    M-step: Compute $\theta^{s+1}$ = $\arg\min_\theta L(T^s|F)$.
    Set $s \leftarrow s + 1$.
  until convergence
```

In the E-step, under the condition of the current parameter θ^s and the decision values

$$F \;=\; \{f_i \mid i = 1, \ldots, l\}, \tag{24}$$

the expect labels are decided by (3). Therefore,

$$t_i^s = \mathrm{E}(t_i|F, \theta^s) = \begin{cases} 1 & \text{if } A^s f_i + B^s \leq 0, \\ 0 & \text{if } A^s f_i + B^s > 0. \end{cases} \tag{25}$$

Let $L(T^s|F)$ represent the negative log-likelihood function in (5). Parameters $\theta^{s+1} = (A^{s+1}, B^{s+1})$ are obtained in the M-step by minimizing the following function.

$$\begin{aligned} L(T^s|F) = -\sum_{i=1}^{l} (t_i^s \log(P(y_i = 1|\boldsymbol{x}_i)) \\ + (1 - t_i^s) \log(1 - P(y_i = 1|\boldsymbol{x}_i))). \end{aligned} \tag{26}$$

Once $t_i, \forall i$ are fixed, (26) is the same as (5), which is the optimization problem solved in Platt scaling. To avoid the overfitting situation, techniques in (6) can be incorporated in the label prediction by (25). Unfortunately, our experiments indicate that even though t_i is now iteratively updated, the final curve is still close to the 0-1 curve. Therefore, it seems this algorithm has not fully addressed the issue of Platt scaling for one-class probabilistic outputs.

D. Converting a Sequence of Scores into Probabilities by Regularization and Normalization

For outlier detection problems, a widely used method to get probabilistic outputs is by regularizing and normalizing a sequence of scores [19]. Decision values in one-class SVM can also be treated as a sequence of scores.

In this method, a score S is called *regular* if $S(\boldsymbol{x}) \geq 0$ for any instance $\boldsymbol{x}$. Since usually a small portion of data instances are outliers, the authors of [19] consider

$$S(\boldsymbol{x}) \begin{cases} \approx 0 & \text{if } \boldsymbol{x} \text{ is a normal instance,} \\ \gg 0 & \text{if } \boldsymbol{x} \text{ is an outlier.} \end{cases} \tag{27}$$

Next, to obtain probabilistic outputs, we must convert $S(\boldsymbol{x})$ to the interval $[0, 1]$. In [19], a converted score S is called *normal* if S is *regular* and the values are restricted by $S(\boldsymbol{x}) \in [0, 1]$.[1]

[1] This definition is about a score S. It is not related to whether an instance is normal or not.

In one-class SVM, normal data always have larger decision values than outliers. To make outliers have larger values, the simplest way is to take the difference between each score $f(\boldsymbol{x})$ and the maximal observed score $f_{\max}$. By the following definition our scores satisfy $S(\boldsymbol{x}) \geq 0$, $\forall \boldsymbol{x}$.

$$S(\boldsymbol{x}) = f_{\max} - f(\boldsymbol{x}). \tag{28}$$

We can then linearly scale $S(\boldsymbol{x})$ to $[0, 1]$ for obtaining probabilistic values. However, because the scores are usually not distributed uniformly, a nonlinear way to convert scores to probabilities is needed. Some possibilities are Gaussian scaling and Gamma scaling [19].

Gaussian scaling assumes the scores follow a Gaussian distribution when the sample size is large enough. Since Gaussian distribution has just two parameters, mean μ and variance σ^2, this method is not susceptible to overfitting. Two estimators can be easily got by

$$\hat{\mu} = E(S), \tag{29}$$

and

$$\hat{\sigma}^2 = E(S^2) - E(S)^2. \tag{30}$$

Because of (27), values on the right end of the distribution have higher possibility to be outliers. Further, if $S(\boldsymbol{x}) \leq \mu$, then very unlikely the point is an outlier and we can impose the probability to be zero. To this end, a score can be transformed to a probability value in $[0, 1]$ by the way in (31).

$$P(\text{outlier}|\boldsymbol{x}) \approx \max\left\{0, \text{erf}(\tfrac{S(\boldsymbol{x})-\mu}{\sigma\cdot\sqrt{2}})\right\}. \tag{31}$$

However, scores are not always distributed in Gaussian. For example, [19] have pointed out that the scores of kNN-based methods on low-dimensional data instead resemble a Gamma distribution. By assuming the Gamma distribution, [19] introduced Gamma scaling, which is conceptually similar to Gaussian scaling. From $\hat{\mu}$ and $\hat{\sigma}^2$ obtained by (29) and (30), the parameters k and θ of the Gamma distribution $\Gamma(k, \theta)$ can be respectively approximated by

$$\hat{k} = \frac{\hat{\mu}^2}{\hat{\sigma}^2}, \quad \hat{\theta} = \frac{\hat{\sigma}^2}{\hat{\mu}}.$$

The CDF of Gamma distribution $\Gamma(k, \theta)$ is given by

$$\text{CDF}^{\text{gamma}}(S(\boldsymbol{x})) = \frac{\gamma(k, S(\boldsymbol{x})/\theta)}{\Gamma(k)}, \tag{32}$$

where $\gamma(\cdot)$ is the lower incomplete gamma function and $\Gamma(\cdot)$ is the gamma function. Then similar to (31), the probability can be obtained by the following way.

$$P(\text{outlier}|\boldsymbol{x}) \approx \max\left\{0, \tfrac{\text{CDF}^{\text{gamma}}(S(\boldsymbol{x}))-\mu_{\text{CDF}}}{1-\mu_{\text{CDF}}}\right\}, \tag{33}$$

where $\mu_{\text{CDF}} = \text{CDF}^{\text{gamma}}(\mu)$.

We can see that (31) and (33) are proposed to enhance the contrast between outlier and normal data. An instance's probability to be an outlier is directly assigned to 0 if its *regular* score is less than the mean.

E. Advanced Methods by Modifying Data or Optimization Problem

Since the lack of labels is one of the biggest challenges in one-class SVM probability estimation, some existing studies tried to address this issue by modifying the data set used or the optimization problem.

The work [7] considers the situation where normal data instances are known, but the set of outliers is empty or under-represented. While labels of training data are known, due to the high class imbalance, they apply one-class SVM. To apply Platt's probabilistic outputs, the authors of [7] point out the over-fitting issue when the set of outliers is empty or too small. Then they artificially generate outliers to address the issue. Thus for this approach the training data set is expanded from the original one.

In the work [11], the idea is to solve several one-class SVM problems to obtain more information of the data distribution. By considering

$$0 < \nu_q < \cdots < \nu_1 < 1,$$

they devise an extension of one-class SVM to obtain q parallel separating hyper-planes in the mapped space. Then the decision boundaries in the input space become nested level sets like a contour. This setting can be used for obtaining probabilistic outputs. By the property that ν is an upper bound of support vectors, $\alpha_i = 1-\nu_i$ can be treated as the probability $P(\text{normal}|\boldsymbol{x})$. Then for any test point $\boldsymbol{x}$, we simply check which level set it falls into. Other works that have applied similar ideas include, for example, [9], [12], [13].

However, we do not consider the above works for the following reasons.

- We aim to have a simple way for providing probabilistic outputs for one-class SVM. Therefore, data modification or augmentation is beyond our consideration.
- For approaches such as q-OCSVM in [11], an optimization problem much larger than that of standard one-class SVM is solved. The number of variables is q times.[2] The training process may be time consuming, but here we want a simple and efficient setting. Moreover, even though their design has utilized properties of the parameter ν, they still need to decide other parameters such as the kernel parameters.

Nevertheless, in Section V-C for experiments, we include a simplified setting of q-OCSVM (called I-OCSVM) by solving q independent optimization problems.

F. An Extension of SVDD for Probabilistic Outputs

Support Vector Data Description (SVDD) is a variant of one-class SVM that can also be used for outlier detection [16]. The work [20] modifies the loss function of SVDD to be the logistic loss so that probabilistic outputs can be produced. This extension, called Import Vector Domain Description (IVDD) is essentially kernel SVDD logistic regression. The relation between SVDD and IVDD is the same as that between kernel two-class SVM and kernel logistic regression. By similar

[2] See Eq. (5) in [11].

concepts we can extend kernel one-class SVM to kernel one-class logistic regression for directly getting the probabilities. Details of our derivations are in supplementary materials.

However, we note a concern about using kernel one-class logistic regression. In the two-class scenario, the main differences between SVM and logistic regression can be summarized as follows.

- For kernel logistic regression, every instance is a support vector and therefore, the model size is generally much larger than that of kernel SVM.
- Kernel logistic regression can directly give probabilities, but kernel SVM cannot. This situation is exactly why some methods for probabilistic outputs (e.g., Platt's method) are needed for SVM.

The same property holds for one-class SVM and SVDD; see, for example, [20] mentioned that every instance is a support vector by saying "at the optimum, all alphas are positive and that they sum to 1". To reduce the model size, the authors of [20] then rely on an early stopping criterion of the optimization process to gain some sparsity. Therefore, if we prefer kernel one-class SVM over logistic regression for a smaller model involving only a subset of data, then we need techniques proposed in this work for probabilistic outputs.

IV. Methods to Generate Probabilistic Outputs for One-class SVM

In previous sections we discuss some mature approaches widely used in probability estimation, but many are not suitable for one-class SVM. In this section, we propose a new non-parametric method and a new parametric method to estimate the probabilities. To address the difficulty of not having true labels of training data, our methods are designed based on the principle to have probabilities mimic to the decision values of training data.

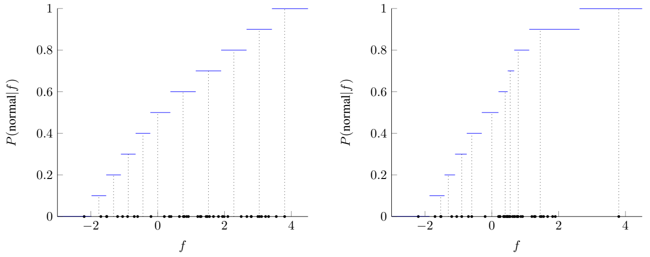
A. Binning by Decision Values

Binning by decision values has been an existing approach to generate probabilistic outputs [21]. The idea is to cut the range of the decision values into several intervals, and let the samples with decision values in the same interval have the same class probability. For two-class classification, the probability value of each interval is decided by the ratio of positive instances. Now for one-class SVM, true labels are not available, so our idea is to obtain the probabilistic output solely by using decision values of training data. Because a larger decision value indicates a higher probability of being positive, we generate several intervals to cover all training data's decision values and assign probability values to these intervals in an increasing manner. While this idea is simple, some details must be specified and we discuss them below.

To begin, it is reasonable to assume

$$P(\text{normal}|f=0)=0.5.$$

Take $f=0$ as a mark, and we should pick some other marks from $[f_{\min}, f_{\max}]$, where $f_{\min}$ and $f_{\max}$ are respectively the minimal and the maximal decision values among training data, and each mark (except the one at $f=0$) is designated as the center of an interval. Assume that we select five marks with $f>0$ and another five with $f<0$. Then the easiest way to pick the marks is to respectively select 5 points from $[f_{\min}, 0]$ and $[0, f_{\max}]$ equidistantly. Specifically, we obtain the following marks.



(a) Binning equidistantly. (b) Binning according to density.

Fig. 4: Illustrations for the two binning methods. Black dots on the x-axis are the decision values of training set. The vertical dotted lines show the marks we selected, and the blue line segments represent the bins with different probabilistic outputs.

$$f_{\min}, 4\times f_{\min}/5, \ldots, f_{\min}/5, 0, f_{\max}/5, \ldots, 4\times f_{\max}/5, f_{\max}.$$

The 11 marks correspond to 11 probability values 0, 0.1, . . ., 1. For a coming decision value, its probability of being a normal data instance is decided by its nearest mark. Figure 4a illustrates the idea.

For the first and the last marks, assigning values 0 and 1 respectively may not be appropriate as some uncertainty should remain. Thus we may instead consider values such as 0.001 and 0.999.

An issue of picking the marks in an equidistant setting is that the model may be affected by some outliers. For example, if $f_{\max}$ is extremely large and all other positive training decision values are in $[0, f_{\max}/2]$, the model may end up with predicting most data to have $P(\text{normal}|f) \leq 0.75$. A possible remedy is to generate intervals according to the density of decision values. That is, all positive decision values are sorted and split into equal-sized groups. An illustration is in Figure 4b.

For some implementation details, see Section 8.3 in LIBSVM implementation document [15].

B. A New Gamma Scaling

In Section III-D, we described the method in [19] to convert scores (decision values) to probabilities. However, the setting in (31) implies that

$$P(\text{normal}|\boldsymbol{x})=1 \quad \text{if} \quad S(\boldsymbol{x}) \leq \hat{\mu}. \tag{34}$$

This property may not be desired because we hope to have that $P(\text{normal}|\boldsymbol{x})$ gradually increases to 1 as $S(\boldsymbol{x})$ goes to zero. The setting in (33) of assuming the Gamma distribution possesses the same problem. In this section, by taking properties of one-class SVM into account, we propose a method to address the issue. In particular, we need to check if the decision values are under certain distributions.

1) Kernel Selection: To get decision values by kernel one-class SVM, we must consider a suitable kernel. Past works such as [22] have argued that for one-class SVM, kernels that can reflect the similarity between two input instances are preferable. That is, a kernel function should show a high similarity if the two vectors are close to each other and a low similarity if the two vectors are distant. Therefore, the maximal similarity occurs if the two input vectors are identical. However, kernels like linear kernel or polynomial kernel do not have this property.

In this way, good candidates for one-class SVM should be distance based kernels like RBF kernel and Laplacian kernel. Here we focus on considering the RBF kernel.

2) Distribution of the Decision Values: In (21), we introduced an approximation form of decision values. Here we still assume that the training set has n independent features, and for a sample $\boldsymbol{x}_i$, its jth feature $x_i^{(j)}$ is under a Gaussian distribution of $N(\mu^{(j)}, (\sigma^{(j)})^2)$. From (21), the domain of decision values is in the range of $(-\infty, C]$. Then the maximum decision value that appears in the training data is approximately

$$f_{\max} \approx f(\bar{\boldsymbol{x}}) = C. \tag{35}$$

To know the distribution clearly, we regularize the decision values into a non-negative domain by the following setting.

$$\begin{aligned} f(\boldsymbol{x})_{\text{Reg}} &= f_{\max} - f(\boldsymbol{x}) \in [0, \infty) && (36)\\ &\approx \gamma \|\boldsymbol{x} - \bar{\boldsymbol{x}}\|^2, && (37)\end{aligned}$$

where (37) is from (21) and (35).

If the training data x is one-dimensional, from the assumption that the feature is under a Gaussian distribution, we know that $(x - \bar{x})/\sigma$ follows a standard normal distribution. Thus the regularized decision value

$$f(x)_{\text{Reg}} \approx \gamma\sigma^2 (\frac{x - \bar{x}}{\sigma})^2$$

can be considered as a weighted Chi-square random variable with one freedom degree and weight $\gamma\sigma^2$.

Multi-dimensional training data can be considered in the same way. The formulation of the regularized decision value is a weighted sum of Chi-square random variables.

$$\begin{aligned} f(\boldsymbol{x})_{\text{Reg}} &\approx \gamma \|\boldsymbol{x} - \bar{\boldsymbol{x}}\|^2 \\ &= \gamma \sum_{j=1}^{n} (x^{(j)} - \mu^{(j)})^2 \\ &= \sum_{j=1}^{n} \gamma (\sigma^{(j)})^2 (\frac{x^{(j)} - \mu^{(j)}}{\sigma^{(j)}})^2. \end{aligned}$$

3) Transformation to Probabilities: Although the weighted sum of Chi-square random variables has no specific probability density function, there exist various approaches to approximate it by other distributions. Satterthwaite-Welch approximation is an efficient method which involves matching the moments of the weighted sum of Chi-square random variables to a Gamma distribution [23]. To be specific, it is a kind of method of moment, and only needs the mean and the variance. Define S to be the value $f(\boldsymbol{x})_{\text{Reg}}$ in (36). We can firstly get the estimated mean and variance by

$$\hat{\mu} = E(S),$$

and

$$\hat{\sigma}^2 = E(S^2) - E(S)^2.$$

These two estimators can then help to estimate the shape parameter k and the scale parameter θ of the target Gamma distribution by

$$\hat{k} = \frac{\hat{\mu}^2}{\hat{\sigma}^2}, \quad \hat{\theta} = \frac{\hat{\sigma}^2}{\hat{\mu}}.$$

Since the regularized decision values have the characteristic that the greater the more likely to be an outlier, we can model $P(\text{outlier}|\boldsymbol{x})$ as

$$P(\text{outlier}|\boldsymbol{x}) \equiv P(X \leq f(\boldsymbol{x})_{\text{Reg}}),$$

where X stands for a random variable having the same distribution with f_{Reg}. Similar to (32), the probability can also be represented by a CDF of the Gamma distribution with shape $\hat{k}$ and scale $\hat{\theta}$

$$\begin{aligned} P(\text{outlier}|\boldsymbol{x}) &= \text{CDF}^{\text{gamma}}(f(\boldsymbol{x})_{\text{Reg}}) \\ &= \frac{\gamma(k, f(\boldsymbol{x})_{\text{Reg}}/\theta)}{\Gamma(k)}. \end{aligned} \tag{38}$$

Notice that the γ function in (38) represents a lower incomplete gamma function rather than the parameter in RBF kernel. For one-class SVM, samples located at the hyper-plane should have the same probabilities belonging to the normal and the outlier classes. That is, samples with decision value 0 have the probability 0.5 to be an outlier. From (36),

$$f(\boldsymbol{x})_{\text{Reg}} = f_{\max} \quad \text{if} \quad f(\boldsymbol{x}) = 0, \tag{39}$$

so similar to (18), the scaled probabilistic output $P(\text{normal}|\boldsymbol{x})$ is

$$\begin{cases} \frac{\text{CDF}^{\text{gamma}}(f_{\max}) - \text{CDF}^{\text{gamma}}(f(\boldsymbol{x})_{\text{Reg}})}{\text{CDF}^{\text{gamma}}(f_{\max})} \times 0.5 + 0.5 & \text{if } f(\boldsymbol{x}) \geq 0, \\ \frac{(1 - \text{CDF}^{\text{gamma}}(f(\boldsymbol{x})_{\text{Reg}})}{1 - \text{CDF}^{\text{gamma}}(f_{\max})} \times 0.5 & \text{otherwise.} \end{cases} \tag{40}$$

4) Advantages of the New Gamma Scaling: Compared with the methods discussed before, we think the new Gamma scaling is competitive for the following reasons.

- The new Gamma scaling is a parametric probability estimation method. The three parameters $\hat{\mu}$, $\hat{\sigma}^2$ and $f_{\max}$ can be obtained in linear time $O(l)$.
- The probability function in (40) is continuous, so its values may more accurately reflect the distribution than the binning methods.
- Although we only consider the RBF kernel here, this method may be applied to other distance-based kernels like the Laplacian kernel.

V. Experiments

In this section, we first give our performance measure, describe the setting of the experiments, and then show our evaluation result of various methods. We use the one-class SVM implementation in LIBSVM [15] for experiments. The RBF kernel in (20) is considered because it is the most commonly used kernel.

A. Performance Measure

If ground-truth probabilities can be obtained, we take the mean squared error (MSE) as a criterion

$$\text{MSE} = \frac{1}{l}\sum_{i=1}^{l}(p_i - \hat{p}_i)^2,$$

where p_i means the ground truth probability of data $\boldsymbol{x}_i$, and $\hat{p}_i$ means the probabilistic output obtained by the method to be evaluated.

For artificial data used in our experiments, they are created according to certain distribution patterns (e.g., Gaussian distribution, uniform distribution), so we can get their ideal probabilities as the ground truth. On the other hand, we have no clear knowledge about the distribution of most real-world data sets, so we do not have their ground truth in the evaluation stage. In this way, MSE is only applicable on the data sets generated under known distributions.

If ground truth is not available, we consider the quantile-quantile plot (Q-Q plot) to check if the probability estimation results can reflect the distribution of the decision values accurately. The reason is that an important aspect of estimating probabilities is to maintain some semblance to the original score distribution [24]. Statistical methods like drawing a Q-Q plot provide an idea to compare samples from any two distributions.

Here is a brief introduction about how the Q-Q plot works. Consider an example to check if n samples follow any target distribution. Firstly we sort the data from the smallest to the largest. Then we consider the density function of the target distribution and find the n points that make the area under the density function split to $n+1$ equally spaced areas. We draw a scatter plot with the given n values and the n values from the target distribution. The resulting figure is called the Q-Q plot. If the data set to be checked closely follow the target distribution, the dots in the resulting plot should almost locate on the line of $y = x$. In this way, the Q-Q plot can helps to verify if the distributions of the two sets of data are the same.

In the evaluation by using a Q-Q plot, we use the generated probabilities of the test set to get the sample quantiles, and scale the percentile of training data's decision values to get the theoretical quantiles (i.e., values from the target distribution). A scaling step is necessary because data with zero decision value should have their probabilities to be normal around 0.5. Here we denote the percentile at $f(\boldsymbol{x}) = 0$ as $\hat{\text{Prc}}(0)$.[3] Then, by settings similar to (22) and (50), we calculate each sample's percentile $\text{Prc}(f(\boldsymbol{x}))$ from the decision value $f(\boldsymbol{x})$ and have

$$\begin{cases} \frac{\text{Prc}(f(\boldsymbol{x}))-\hat{\text{Prc}}(0)}{1-\hat{\text{Prc}}(0)} \times 0.5 + 0.5 & \text{if } \text{Prc}(f(\boldsymbol{x})) \geq \hat{\text{Prc}}(0), \\ \frac{\text{Prc}(f(\boldsymbol{x}))}{\hat{\text{Prc}}(0)} \times 0.5 & \text{otherwise.} \end{cases} \tag{41}$$

B. Artificial Data: Data Generation

The purpose of considering artificial data is twofold: first, because the theoretical probabilities are known, the performance comparison of different probabilistic output methods is possible. Second, we would like to verify the analysis in Section II, which shows that fitting a sigmoid function may not be suitable for one-class SVM. Note that for each set, to apply the theoretical probability in the case of imbalanced normal and outlier data, we need to scale it by the same setting in (18) according to the proportion of outlier data.

We begin with generating two kinds of artificial data sets by following the discussion in Section II-D. Each set has 10,000 instances and all the instances are one-dimensional. We assume 25% of the instances are outliers. Test sets are generated in the same way as training sets.

In the first artificial data set (ART1), we generate the data randomly in Gaussian distribution with mean 0 and variance 1 in (12). By setting $\nu = 0.25$ in (18), Δ is approximately 1.15 and we have

$$P(\text{normal}|x) = \begin{cases} 1 + \text{erf}(\frac{-|x|}{\sqrt{2}}) \times \frac{2}{3} & \text{if } x \in [-\Delta, \Delta], \\ (1 + \text{erf}(\frac{-|x|}{\sqrt{2}})) \times 2 & \text{otherwise.} \end{cases} \tag{42}$$

The second artificial data set (ART2) is similar to the first one, but we replace the 75% instances closest to the mean 0 with the uniform distribution as we described in (49). The probabilistic output is in the form of (50), and Δ is approximately 1.15. By setting $\nu = 0.25$ in (50), we have

$$P(\text{normal}|x) = \begin{cases} 1 + \frac{|x| \times \text{erf}(-\frac{\Delta}{\sqrt{2}})}{\Delta} \times \frac{2}{3} & \text{if } x \in [-\Delta, \Delta], \\ (1 + \text{erf}(\frac{-|x|}{\sqrt{2}})) \times 2 & \text{otherwise.} \end{cases} \tag{43}$$

The two sets ART1 and ART2 are one-dimensional. To check the performance on multi-dimensional data, we artificially generate some sets by assuming that each feature of the n-dimensional data follows an independent Gaussian distribution with mean 0 and standard deviation 1.

In this way, we generate a 5-dimensional set called ART_5d and a 10-dimensional set called ART_10d. Both sets include 10,000 instances and we assume 25% of them are outliers. From the derivation in supplementary materials, by setting $\nu = 0.25$, then $P(\text{normal}|\boldsymbol{x})$ is

$$P(\text{normal}|\boldsymbol{x}) = \begin{cases} 1 - \frac{\int_0^{\frac{\|\boldsymbol{x}\|}{\sqrt{2}}} e^{-u^2} u^{n-1} du}{\int_0^{\infty} e^{-u^2} u^{n-1} du} \times \frac{2}{3} & \text{if } \|\boldsymbol{x}\| \leq \Delta, \\ (1 - \frac{\int_0^{\frac{\|\boldsymbol{x}\|}{\sqrt{2}}} e^{-u^2} u^{n-1} du}{\int_0^{\infty} e^{-u^2} u^{n-1} du}) \times 2 & \text{otherwise.} \end{cases} \tag{44}$$

Next, we generate a set ART3 that has different characteristics from the earlier ones. Some of the methods we proposed (e.g., the new Gamma scaling) assume that γ in the RBF kernel is small. From (21), a small γ implies that the decision function is like a hyper-sphere[4] no matter how many clusters there exist in the data set. Then the obtained model may not be capable of predicting the sparse samples that locate between clusters as outliers. To check how our proposed methods perform in such a situation, we generate a 2-dimensional artificial data set called ART3 by the following ways.

[3] In general no sample's decision value is right at 0, so we take the one which has the minimum absolute decision value as the replacement.

[4] Note that we mean a hyper-sphere in the input space rather than the feature space after kernel mappings.

- First, two clusters are respectively generated by multivariate normal distributions $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ with

$$\begin{aligned}\boldsymbol{\mu}_1 &= \begin{bmatrix}6\\5\end{bmatrix}, \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix},\\ \boldsymbol{\mu}_2 &= \begin{bmatrix}-6\\-5\end{bmatrix}, \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix},\end{aligned} \tag{45}$$

  and each cluster has 4,950 samples.
- Second, 100 outliers are generated by a uniform distribution in the range $[-10, 10]$ for both dimensions.

Since the ratio of outliers in ART3 is very low, we can calculate the nearly ideal probability by ignoring the distribution of the outliers. The probability density function of the mixture distribution in ART3 can be approximated to

$$\frac{1}{2}\psi_1(\boldsymbol{x}) + \frac{1}{2}\psi_2(\boldsymbol{x}), \tag{46}$$

where $\boldsymbol{x}$ is a vector with two features, and $\psi_1(\boldsymbol{x})$ and $\psi_2(\boldsymbol{x})$ are the probability density functions of distributions $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, respectively. Similar to (13), we can consider the instances that are far from the dense places as outliers. Without loss of generality, let $\boldsymbol{x}$ be an instance closer to $\boldsymbol{\mu}_1$. We can derive the following ideal probability $P(\text{normal}|\boldsymbol{x})$ by assuming that the ratio of predicted outliers is ν.

$$\begin{cases} \frac{(1-P(\text{outlier}|\boldsymbol{x})-\nu)}{1-\nu} \times 0.5 + 0.5 & \text{if } P(\text{outlier}|\boldsymbol{x}) \leq 1-\nu,\\ \frac{(1-P(\text{outlier}|\boldsymbol{x}))}{\nu} \times 0.5 & \text{otherwise,} \end{cases} \tag{47}$$

where

$$P(\text{outlier}|\boldsymbol{x}) = \iint\limits_{\|\boldsymbol{X}-\boldsymbol{\mu}_1\| \leq \|\boldsymbol{x}-\boldsymbol{\mu}_1\|} \frac{1}{2\pi} e^{-\frac{X_1^2+X_2^2}{2}} dX_1 dX_2,$$

and $\boldsymbol{X}$ is a random variable in generating $\boldsymbol{x}$. Details of the derivation are in supplementary materials.

C. Artificial Data: Results

For ART1 and ART2, we set parameter $\nu = 0.25$ since we have that 25% of the artificial data are outliers. We set $\gamma = 0.0001$ because a small γ is needed in the discussion in Section II-D and supplementary materials, and we have the following relationship between an instance x and its decision value f.

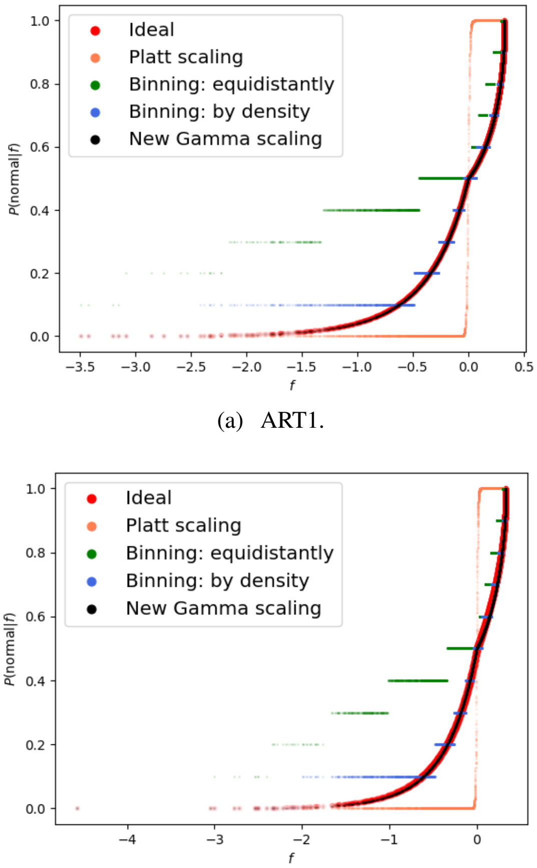
$$f \approx -\gamma(x-\bar{x})^2 + \gamma\Delta^2, \tag{48}$$

where $\bar{x}$ is the mean of training data. With (48), in Figure 5a and Figure 5b respectively for ART1 and ART2 we present the relation between decision values and probabilistic outputs by the following methods.

- Ideal probability: see (42), (43) and we also use (48).
- Platt scaling: see Section II-A.
- Binning methods: see Section IV-A. We have two settings: binning by density and binning equidistantly.
- New Gamma scaling: see Section IV-B.

From Figure 5, we have the following observations.

- The curves of Platt scaling are extremely steep around 0 as we discussed before. Its probabilistic outputs are very close to either 0 or 1. Some additional experiments to analyze Platt scaling for one-class SVM are in supplementary materials.
- Results of the two binning approaches and the new Gamma scaling are not clustered around 0 or 1. More specifically, the estimation results of binning by density and the new Gamma scaling are very close to the ideal probabilities.



(a) ART1.

(b) ART2.

Fig. 5: Relationship between probabilistic outputs and decision values for artificial data sets ART1 and ART2.

Table I shows the MSE of probabilistic outputs by various methods. We include one more method for the comparison.

- I-OCSVM: this stands for independent one-class SVMs. Recall that in Section III-E we discussed the approach q-OCSVM [7]. A simplified setting experimented in the same paper is to consider q independent one-class SVM problems with $\nu = \nu_1, \ldots, \nu_q$. Details of our implementation are in supplementary materials.

From the result in Table I, the performance of the new Gamma scaling on ART1 and ART2 is outstanding with the smallest MSE, and binning by density also performs well. For the two binning methods, the equidistance setting is clearly inferior. The performance of I-OCSVM is close to binning by density though the former must solve q instead of one SVM problems.

In Section IV-B we developed a new Gamma scaling method

	ART1	ART2	ART_5d	ART_10d	ART3 ($\gamma = 0.1$)	ART3 ($\gamma = 0.0001$)
Platt scaling	0.077723	0.077020	0.078548	0.077663	0.081106	0.105299
Binning: equidistantly	0.026122	0.020175	0.012999	0.013073	0.024938	**0.047060**
Binning: by density	0.001056	0.001135	0.001087	0.001132	**0.002000**	0.051590
New Gamma scaling	**0.000003**	**0.000212**	**0.000041**	**0.000079**	0.007661	0.058946
I-OCSVM	0.001044	0.001161	0.001088	0.001119	0.002458	0.051775

TABLE I: Mean squared error (MSE) of the probabilistic outputs by various methods. For each set, the method with the smallest MSE is bold-faced.

because from the discussion in Section III-D, the method of [19] satisfies the undesired property in (34). To confirm the need of our proposed method, we conduct a comparison in supplementary materials. Results indicate that (34) really causes inferior performance.

Next, we consider the two multi-dimensional data sets ART_5d and ART_10d. Parameters are set in the same way as those for ART1 and ART2. From MSE shown in Table I, both the new Gamma scaling and the binning by density work well.

Finally, we check the set ART3. Recall ART3 contains two clusters of normal data, while outliers (1% of total data) are those that are away from the two clusters. We apply the following parameter settings.

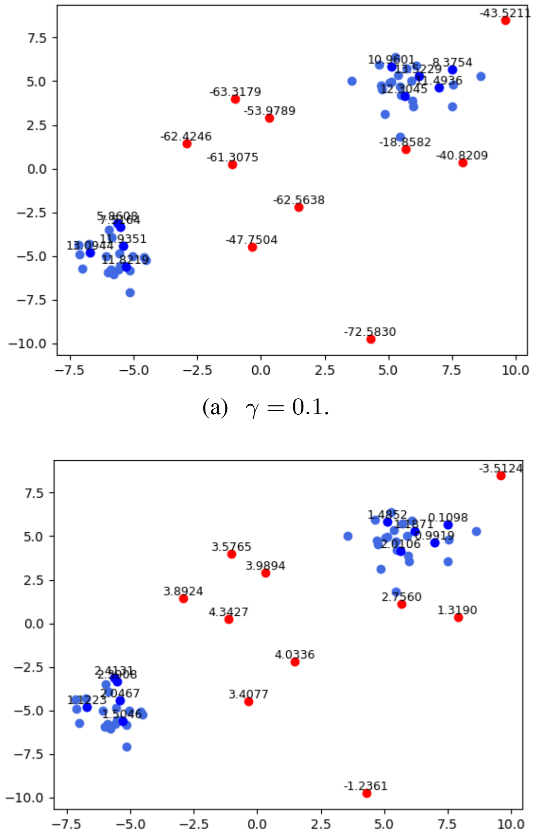
- For ν, we consider $\nu = 0.05$, instead of the ratio of outliers (0.01). Because 0.01 is small, we impose a larger value as ν for hoping that the model can predict enough outliers.
- For the kernel parameter γ, we check two values.

  $$\gamma = 0.1 \text{ and } \gamma = 0.0001.$$

  It is known that a larger γ implies a higher nonlinearty of the decision boundary.

To see how the model performs, in Figure 6 we plot some randomly selected points. Next to each point, we show the decision values after the training procedure. We observed that if $\gamma = 0.1$ is used, the one-class SVM model leads to two non-linear curves that roughly circle the two clusters. In contrast, if $\gamma = 0.0001$ is used, one-class SVM leads to a larger sphere that covers almost all the points. In this situation, outliers cannot be identified. Next, we separately generate corresponding probabilistic outputs and report MSE in Table I. As expected, the MSE under $\gamma = 0.0001$ is much worse than that under $\gamma = 0.1$. This result fully demonstrates that a bad one-class SVM model leads to inappropriate decision values and then poor probabilistic outputs. Further, in contrast to the MSE results for ART1 and ART2, now the binning by density is better than the new Gamma scaling. The reason may be that the latter assumes each feature follows a Gaussian distribution, but the data generation of ART3 violates this assumption.

In summary, binning by density is a reliable choice because it solely relies on the quality of decision values rather than other assumptions. Of course obtaining a good one-class SVM model with suitable decision values can still be challenging, but the same issue has occurred in conducting classification instead of finding probabilistic outputs. On the other hand, although the performance of I-OCSVM is close to binning by density, training several one-class SVM models is not necessary.

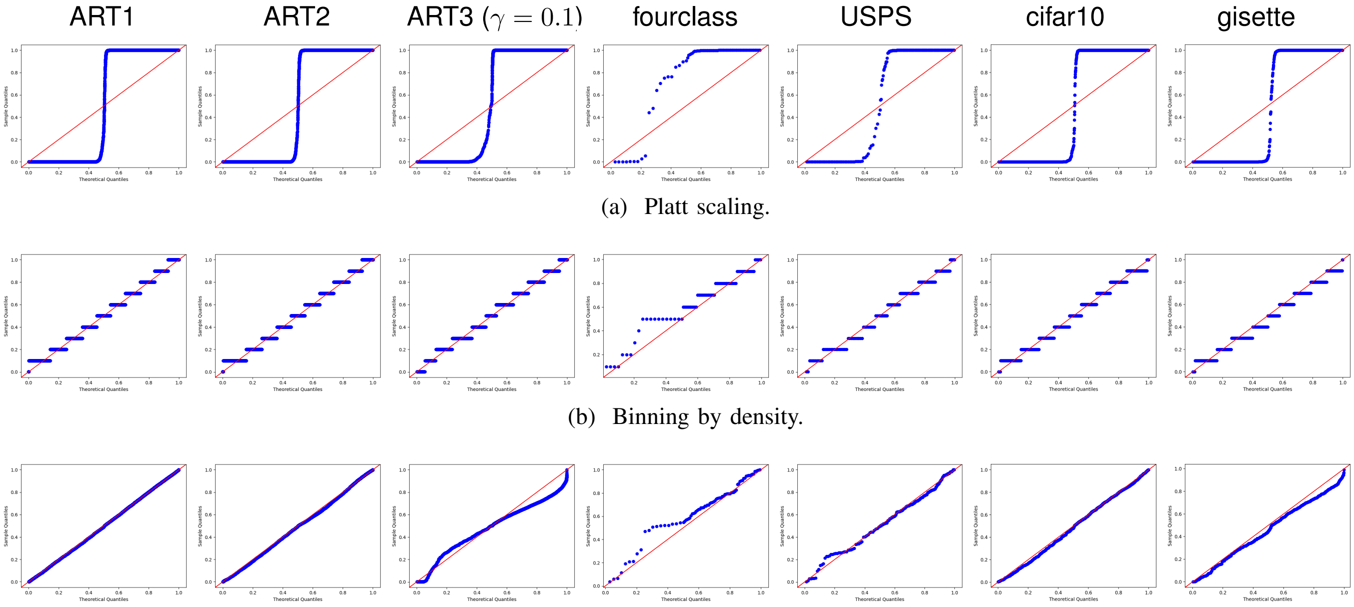


(a) $\gamma = 0.1$.

(b) $\gamma = 0.0001$.

Fig. 6: An illustration of ART3. Blue points indicate normal data and red points are outliers. The value next to each point is the decision value.

Now we use Q-Q plots in Figure 7 to check the relation between the obtained probabilistic outputs and the decision values. As mentioned earlier, if points in the Q-Q plot are close to the $y = x$ line, then the probabilistic outputs can represent the distribution of decision values well. From the figures we have the following observations. Platt scaling cannot reflect the distribution of decision values at all. Binning by density is a conservative approach so that their probability results always have nearly a fixed error pattern. Finally, the new Gamma scaling works extremely well for ART1 and ART2, but is not as good for ART3. This result is reasonable because, as we mentioned earlier, decision values of ART3 may not satisfy the assumption made by the new Gamma scaling.



(a) Platt scaling.

(b) Binning by density.

(c) The new Gamma scaling.

Fig. 7: Q-Q plots for comparing decision values and probabilistic outputs obtained by different methods.

	fourclass	USPS	cifar10	gisette
Classes (original)	2	10	10	2
Features	2	256	3,072	5,000
Training set	245/81	584/194	5,000/1,666	3,000/1,000
Test set	61/20	146/48	1,000/333	500/166

TABLE II: Details of the real-world data sets. The sizes of training sets and test sets are in the form of normal data/outliers.

D. Real-world Data

We use four real-world data sets, which are widely used classification benchmarks. We pick one class as the normal data and randomly select a small number of data from other classes as outliers.

The first set is the "fourclass" problem.[5] We choose the positive class as the normal samples and randomly select instances from the negative class as outliers. The second set is from the "USPS" problem,[6] which includes 10 handwritten digits. We choose digit 3 as the normal data, and randomly pick the outliers from the other nine classes uniformly. The third set is from the "cifar10" problem.[7] which includes image information. We choose the class of automobile as the normal data and randomly pick the outliers from the other nine classes uniformly. The forth set is from the "gisette" problem.[8] We choose the positive class as the normal samples and randomly select instances from the negative class as outliers.

Both training sets and test sets for the four real-world data sets includes about 25% of outliers. The details of the data sets are in Table II.

Figure 7 shows the evaluation results on real-world data sets. As in the case of artificial data, Platt scaling does not reflect the distribution of decision values. Both the new Gamma scaling and binning by density perform well on USPS, cifar10 and gisette. However, they are not as good on the fourclass set. One reason might be that the size of the data set is small so the distributions of training and test sets are not similar. Another possible reason is that, since the proportion of outliers is less than 0.5, the scaling in (40) stretches the probabilities on the outlier side. Thus, the error is magnified.

VI. Conclusions

This paper analyzes the reasons why one traditional probability estimation method (Platt scaling) for two-class classification problems may not be appropriate for one-class SVM. The possibility of applying other mature probability prediction methods is also discussed. To let probabilities reflect the distribution of decision values, we propose two methods to obtain the probability. One is binning, a non-parametric method, and another is a parametric method called the new Gamma scaling. Both methods can obtain probabilistic outputs in linear time. We show their stability and effectiveness in compared with existing methods through detailed experiments. Based on this study, the simple binning method by density is chosen as the way of generating one-class SVM probabilistic outputs in the popular SVM package LIBSVM.

Appendix
An Additional Illustration for Section II-D

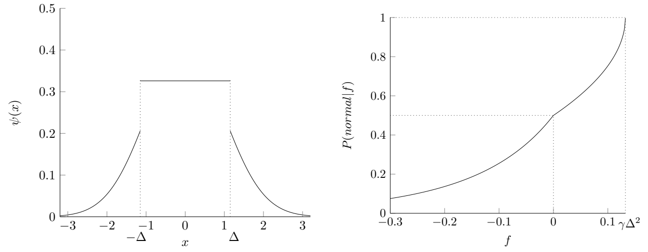
We give an illustration similar to that in Section II-D by assuming that data are generated in a different way. The

[5] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#fourclass. We use the scaled version, where each feature is linearly scaled to [-1,1].

[6] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#usps

[7] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#cifar10. We use the scaled version, where each feature is linearly scaled to [0,1].

[8] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#gisette



(a) An assumed distribution of data.

(b) $P(\text{normal}|f)$.

Fig. 8: A conceptual illustration of one-class SVM probabilistic outputs. Assume normal data are distributed uniformly as in the left figure. Data out of $[-\Delta, \Delta]$ are considered as outliers and they are generated by a Gaussian distribution as in Figure 3a. The curve of $P(\text{normal}|f)$ shapes like the right figure with $\Delta \approx 1.15$.

example in Section II-D assumes all the normal data (i.e., $x \in [-\Delta, \Delta]$) follow a Gaussian distribution. Now let us assume that these points follow a uniform distribution, so the density function becomes

$$\psi(x) = \begin{cases} -\frac{\text{erf}(-\frac{\Delta}{\sqrt{2}})}{2\Delta} & \text{if } x \in [-\Delta, \Delta], \\ \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} & \text{otherwise.} \end{cases} \tag{49}$$

See the curve shown in Figure 8a. The probabilistic output for an instance x can be formed in the same way as in (14). With (19), $P(\text{normal}|x)$ can be modified from (18) to the following form.

$$\begin{cases} (1 + \frac{|x| \times \text{erf}(-\frac{\Delta}{\sqrt{2}})}{\Delta} - \nu) \times \frac{0.5}{1-\nu} + 0.5 & \text{if } x \in [-\Delta, \Delta], \\ (1 + \text{erf}(\frac{-|x|}{\sqrt{2}})) \times \frac{0.5}{\nu} & \text{otherwise.} \end{cases} \tag{50}$$

The curve of $P(\text{normal}|f)$ is shown in Figure 8b. Clearly, the curve is again very different from a sigmoid function.

References

[1] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[2] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," in *Advances in Neural Information Processing Systems*, 2000.

[3] J. C. Platt, "Probabilistic outputs for support vector machines and comparison to regularized likelihood methods," in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press, 2000.

[4] H.-T. Lin, C.-J. Lin, and R. C. Weng, "A note on Platt's probabilistic outputs for support vector machines," *Machine Learning*, vol. 68, pp. 267–276, 2007. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/plattprob.pdf

[5] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.

[6] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, 2005, pp. 625–632.

[7] L. Clifton, D. A. Clifton, Y. Zhang, P. Watkinson, L. Tarassenko, and H. Yin, "Probabilistic novelty detection with support vector machines," *IEEE Transactions on Reliability*, vol. 63, no. 2, pp. 455–467, 2014.

[8] V. Leclère, E. Grave, and L. El Ghaoui, "Probabilistic approach to one-class support vector machine," 2016.

[9] M. El Azami, C. Lartizien, and S. Canu, "Converting SVDD scores into probability estimates: Application to outlier detection," *Neurocomputing*, vol. 268, pp. 64–75, 2017.

[10] J. Gao and P.-N. Tan, "Converting output scores from outlier detection algorithms into probability estimates," in *Proceedings of the Sixth International Conference on Data Mining (ICDM)*, 2006, pp. 212–221.

[11] A. Glazer, M. Lindenbaum, and S. Markovitch, "q-OCSVM: A q-quantile estimator for high-dimensional distributions," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[12] G. Lee and C. Scott, "Nested support vector machines," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1648–1660, 2010.

[13] A. Glazer, M. Lindenbaum, and S. Markovitch, "Learning high-density regions for a generalized Kolmogorov-Smirnov test in high-dimensional data," in *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[14] Z. Que, "One-class svm probabilistic outputs," Master's thesis, National Taiwan University, 2022.

[15] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[16] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, 2004.

[17] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *Proceedings of the Eighth International Conference on Knowledge Discovery and Data Mining*, 2002.

[18] M. Ayer, H. D. Brunk, G. M. Ewing, W. T. Reid, and E. Silverman, "An empirical distribution function for sampling with incomplete information," *The Annals of Mathematical Statistics*, vol. 26, no. 4, pp. 641–647, 1955.

[19] H.-P. Kriegel, P. Kroger, E. Schubert, and A. Zimek, "Interpreting and unifying outlier scores," in *Proceedings of SIAM International Conference on Data Mining (SDM)*, 2011, pp. 13–24.

[20] S. Decherchi and W. Rocchia, "Import vector domain description: A kernel logistic one-class learning algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 7, pp. 1–8, 04 2016.

[21] M. Roederer, A. Treister, W. Moore, and L. A. Herzenberg, "Probability binning comparison: a metric for quantitating univariate distribution differences," *Cytometry: The Journal of the International Society for Analytical Cytology*, vol. 45, no. 1, pp. 37–46, 2001.

[22] A. Bounsiar and M. G. Madden, "Kernels for one-class support vector machines," in *Proceedings of International Conference on Information Science Applications (ICISA)*, 2014, pp. 1–4.

[23] D. A. Bodenham and N. M. Adams, "A comparison of efficient approximations for a weighted sum of chi-squared random variables," *Statistics and Computing*, vol. 26, no. 4, pp. 917–928, 2016.

[24] R. A. Bauder and T. M. Khoshgoftaar, "Estimating outlier score probabilities," in *Proceedings of IEEE International Conference on Information Reuse and Integration*, 2017, pp. 559–568.