Pattern Recognition 148 (2024) 110191

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Bounded exponential loss function based AdaBoost ensemble of OCSVMs

Hong-Jie Xing [a,*], Wei-Tao Liu [a], Xi-Zhao Wang [b]

[a] Hebei Key Laboratory of Machine Learning and Computational Intelligence, College of Mathematics and Information Science, Hebei University, Baoding 071002, China

[b] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

ARTICLE INFO

Keywords:
One-class classification
AdaBoost
Exponential loss function
One-class support vector machine
Outliers

ABSTRACT

As a commonly used ensemble method, AdaBoost has drawn much consideration in the field of machine learning. However, AdaBoost is highly sensitive to outliers. The performance of AdaBoost may be greatly deteriorated when the training samples are polluted by outliers. For binary and multi-class classifications, there have emerged many approaches to improving the robustness of AdaBoost against outliers. Unfortunately, there are too few researches on enhancing the robustness of AdaBoost against outliers in the case of one-class classification. In this study, the exponential loss function of AdaBoost is replaced by a more robust one to improve the anti-outlier ability of the conventional AdaBoost based ensemble of one-class support vector machines (OCSVMs). Furthermore, based on the redesigned loss function, the update formulae for the weights of base classifiers and the probability distribution of training samples are reformulated towards the AdaBoost ensemble of OCSVMs. The empirical error upper bound is derived from the theoretical viewpoint. Experimental outcomes upon the artificial and benchmark data sets show that the presented ensemble approach is more robust against outliers than its related methods.

1. Introduction

Different from binary or multi-class classification, one-class classification utilizes the samples taken from only the target class to learn a decision boundary in the training phase. In the testing phase, testing samples can be identified as target or non-target. Therefore, the binary or multi-class classifier cannot be used to solve the problem of one-class classification. So far, there are many one-class classification approaches, among which one-class support vector machine (OCSVM) [1] and support vector data description (SVDD) [2] are the most popular. When certain conditions are satisfied, OCSVM and SVDD are equivalent [2]. In our work, we only consider OCSVM.

To enhance the classification ability of one-class classifiers, Tax and Duin [3] proposed to combine one-class classifiers. They experimentally compared seven one-class classifiers integrated by five one-class combination rules and declared that combining Parzen density estimators constructed on different feature subsets by the product rule may get the best outcomes for tackling a handwritten digit recognition problem. Seguí et al. [4] combined minimum spanning tree class descriptors by two bagging based ensemble approaches. In comparison with the single descriptor, both ensemble methods obtained higher and similar performances upon the low dimensional and high dimensional data sets, respectively. Casale et al. [5] constructed approximate polytope ensemble of one-class classifiers. They built the boundary of the target class by convex hull. For the non-convex structures, they designed a tilling strategy. Krawczyk and Woźniak [6] combined weighted OCSVMs by a weighted bagging strategy that allocates weights for all the training samples. Therefore, the degree of importance for each training sample can be considered to construct the classification boundary. Liu et al. [7] proposed a two-round clustering based structural ensemble of one-class classifiers. In comparison with pertinent structural or clustering based one-class classifiers, their proposed ensemble method demonstrates better performance and faster training speed. Through the divide-and-conquer strategy, Krawczyk et al. [8] decomposed a difficult problem of multi-class classification into several subproblems of one-class classification. They introduced an ensemble approach based on dynamic ensemble selection to prevent non-competent classifiers and proposed a threshold-based pruning method to get rid of the redundant classifiers in the final ensemble. Although the above-mentioned ensemble approaches can improve the performances of their corresponding one-class classifiers, the adverse effect of outliers on the generalization performance of these ensemble methods has not been considered.

As is well known, bagging and boosting are two most commonly used approaches to construct an ensemble. Bagging is regarded as a variance reduction method [9]. However, combining one-class classifiers by bagging may marginally enhance the generalization ability

* Corresponding author.
E-mail address: hjxing@hbu.edu.cn (H.-J. Xing).

https://doi.org/10.1016/j.patcog.2023.110191
Received 29 December 2022; Received in revised form 22 March 2023; Accepted 5 December 2023
Available online 8 December 2023

of these one-class classifiers. Moreover, the performance improvement produced by bagging on one-class classifiers is statistically insignificant [4]. By contrast, boosting may concurrently decrease bias and variance [10]. Nevertheless, the traditional boosting methods are unfit for tackling outliers because these methods put more emphasis upon outliers. As a representative of boosting algorithm, AdaBoost [11] is prone to be adversely affected by class-label noise and outliers [12, 13]. So far, there are many methods to enhance the performance of AdaBoost against class-label noise. Among these methods, replacing the exponential loss function of AdaBoost with a more robust one is regarded as a commonly used method to reduce the negative effect of class-label noise. Towards binary classification, Cao et al. [14] proposed a revised exponential loss function for AdaBoost by considering different types of samples with respect to noise and class label decision. They utilized k-NN and EM based noise identification functions to construct the noise-detection criterion. Moreover, a new regeneration condition to control the generalization error bound of the proposed method was developed. Thereafter, Sun et al. [15] generalized noise-detection based AdaBoost from the binary classification scenario to the multi-class classification scenario. They designed a noise-detection based multi-class loss function and proposed a new weight updating scheme to alleviate the negative effect of noise. To improve the anti-noise ability of AdaBoost, Miao et al. [16] designed two algorithms named as RBoost1 and RBoost2, which optimize a non-convex loss function of the classification margin and demonstrate higher anti-noise performance in comparison with the conventional AdaBoost. Sabzevari et al. [17] determined the weights of the training samples for vote-boosting by considering the disagreement rate among the base classifiers in the ensemble. Moreover, they validated their method by utilizing the beta distribution as the emphasis function of vote-boosting on the benchmark data sets. Gu and Angelov [18] combined AdaBoost and zero-order fuzzy inference systems (FIS) together, then proposed a multi-class fuzzily weighted AdaBoost (FWAdaBoost)-based ensemble system with a self-organizing FIS (SOFIS). Moreover, FWAdaBoost uses the confidence scores generated by the SOFIS in both sample weight updating and ensemble output generation.

For binary and multi-class classification, there are many approaches to alleviate the negative impact of outliers upon the performance of AdaBoost. Takenouchi and Eguchi [13] utilized the linear combination of exponential loss function and naive error loss function to substitute the exponential loss function of AdaBoost. Hence, the impact of forgetfulness is introduced into AdaBoost. To avoid the overfitting problem caused by outliers, Sun et al. [19] devised a regularized AdaBoost algorithm with its corresponding optimization problem is transformed to a linear programming problem by the stabilized column generation technique. To improve the anti-outlier performance of AdaBoost, Kanamori et al. [20] presented a transformation formula of loss functions. Moreover, they designed a robust eta-boost algorithm which is robust against outliers. To make AdaBoost possess adaptability for the changing network environment, Hu et al. [21] designed two online AdaBoost-based intrusion detection algorithms. One uses decision stumps as weak classifiers, while the other utilizes online Gaussian mixture models as weak classifiers. Experimental outcomes on the outliers polluted network data demonstrate that both algorithms are superior to their related methods. Wang [22] proposed several robust boosting algorithms based on the majorization–minimization framework and the truncated loss functions. Furthermore, robust AdaBoost based on the truncated exponential loss function was proposed to alleviate the impact of outliers. Wang et al. [23] introduced the idea of self-paced learning into AdaBoost and designed a new robust AdaBoost algorithm. Moreover, they validate the robustness of their proposed algorithm against outliers through theoretical analysis and experimental investigation.

Although AdaBoost and its robust versions are extensively used in the scenarios of binary classification and multi-class classification, they are rarely utilized to tackle one-class classification problems. For the existing AdaBoost based one-class classifiers, OCSVM is usually used as their base classifier. In order to make AdaBoost suitable for integrating OCSVMs, Chen et al. [24] redesigned the update formulae of combination weights for base classifiers and probability distribution of training samples. After discussed the relation between support vector machine and boosting, Rätsch et al. [25] proposed a novel leveraging algorithm for one-class classification. The proposed algorithm benefits from both ν-Arc and column-generation algorithms. Moreover, its corresponding optimization problem can be solved by the barrier-optimization technique. Tao et al. [26] derived a linear programming problem for linear one-class classifier. To obtain nonlinear one-class classifier, they utilized boosting to combine linear base classifiers. Moreover, they analyzed the generalization error of their proposed linear one-class classifier in terms of margin and covering number. Recently, Xing and Liu [27] used the mixture of the modified exponential loss and the squared loss functions to replace the exponential loss function of AdaBoost, then constructed the robust AdaBoost based ensemble of OCSVMs.

In this paper, we present a bounded exponential loss function and apply it to construct the AdaBoost ensemble of OCSVMs. The contributions of our study can be summarized as follows:

- The bounded exponential loss function is used to replace the conventional exponential loss function of AdaBoost to decrease the negative impact of outliers. Furthermore, the properties of the bounded exponential loss function are introduced.
- The Newton–Raphson method is designed to update the combination weights of OCSVMs in the bounded exponential loss function based AdaBoost ensemble. Moreover, the update formula for the probability distribution of training samples is deduced for our proposed ensemble method.
- The empirical error upper bound of the bounded exponential loss function based AdaBoost ensemble is established.

The remainder of this paper is structured as follows. Section 2 briefly introduces OCSVM and AdaBoost. Section 3 describes in detail the proposed bounded exponential loss function based AdaBoost ensemble of OCSVMs. In Section 4, empirical studies on the artificial and benchmark data sets validate the presented ensemble. Finally, Section 5 concludes this paper.

2. Preliminaries

In the following, the original and dual optimization problems of OCSVM are surveyed, while the algorithmic description of AdaBoost is provided.

2.1. OCSVM

The aim of OCSVM is to find a separating hyperplane with the maximum margin between the images of training samples and the origin in the high-dimensional feature space. It should be mentioned here that the training samples are all target (or positive-class) samples. Moreover, the origin is utilized as the representative of non-target (or negative-class) samples. The hyperplane of OCSVM is given by

$$\mathbf{w}^T \boldsymbol{\Phi}(\mathbf{x}) - \rho = 0, \tag{1}$$

where $\mathbf{w}$ denotes the weight vector, the superscript T is the transpose of a vector or a matrix, $\boldsymbol{\Phi}(\cdot)$ is a nonlinear transformation, and ρ denotes the bias term.

The quadratic programming problem of OCSVM is as follows.

$$\begin{aligned} \min_{\mathbf{w},\rho,\xi} \ & \frac{1}{2}\|\mathbf{w}\|^2 - \rho + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i \\ s.t. \ & \mathbf{w}^T\boldsymbol{\Phi}(\mathbf{x}_i) \geq \rho - \xi_i \\ & \xi_i \geq 0, \ i = 1, 2, \ldots, N, \end{aligned} \tag{2}$$

where $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_N)^T$ with its elements are slack variables, $\|\cdot\|$ represents the ℓ_2-norm, and $0 < \nu \leq 1$ controls the rate of outliers among the training samples.

According to the Lagrange multiplier approach, the following dual optimization problem for OCSVM can be obtained.

$$\begin{aligned} &\min_{\boldsymbol{\beta}} \ \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\beta_i\beta_j K(\mathbf{x}_i,\mathbf{x}_j) \\ &s.t. \ \sum_{i=1}^{N}\beta_i = 1 \\ &\quad 0 \leq \beta_i \leq \frac{1}{\nu N}, \ i = 1,2,\ldots,N, \end{aligned} \tag{3}$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_N)^T$ with its elements are Lagrange multipliers, while $K(\cdot,\cdot)$ indicates the kernel function with $K(\mathbf{x},\mathbf{y}) = \boldsymbol{\Phi}(\mathbf{x})^T\boldsymbol{\Phi}(\mathbf{y})$. The weight vector $\mathbf{w}$ can be represented as

$$\mathbf{w} = \sum_{i=1}^{N}\beta_i\boldsymbol{\Phi}(\mathbf{x}_i). \tag{4}$$

The bias term ρ can be calculated by utilizing any $\mathbf{x}_i$ whose Lagrange multiplier meets $0 < \beta_i < \frac{1}{\nu N}$, i.e.,

$$\rho = \sum_{j=1}^{N}\beta_j K(\mathbf{x}_j, \mathbf{x}_i). \tag{5}$$

In addition, the decision function is given by

$$f(\mathbf{x}) = \sum_{i=1}^{N}\beta_i K(\mathbf{x}_i,\mathbf{x}) - \rho. \tag{6}$$

Furthermore, the class label of $\mathbf{x}$ is given by

$$\hat{y} = \text{sign}(f(\mathbf{x})), \tag{7}$$

where $\text{sign}(\cdot)$ indicates the sign function.

2.2. AdaBoost

AdaBoost is regarded as an outstanding representative of all boosting algorithms. It can construct a strong classifier from weak classifiers. In the training phase of AdaBoost, the weights of base classifiers and the probability distribution of training samples are calculated on the basis of training error rates. In the beginning, the probabilities of training samples are assigned with the same values, i.e., $\frac{1}{N}$. Thereafter, the first base classifier is trained with the samples chosen according to their probability distribution while its weight is determined according to its training error rate. The probability distribution of training samples is thus updated. In the subsequent iterations, the weights of base classifiers are determined and the probability distributions of training samples are adjusted.

Towards binary classification, we are given N training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ with $y_i \in \{-1, 1\}$. The linear combination of T base classifiers is given by

$$f_T(\mathbf{x}) = \sum_{t=1}^{T}\alpha_t h_t(\mathbf{x}), \tag{8}$$

where α_t is the weight of the tth base classifier h_t. The training process of AdaBoost is shown in Algorithm 1.

3. Bounded exponential loss function based AdaBoost ensemble of OCSVMs

In this section, the bounded exponential loss function is first introduced and its four properties are explored. Then, the update formulae for the weights of base classifiers and the probability distribution of training samples are reformulated towards the bounded exponential loss function based AdaBoost ensemble of OCSVMs. Finally, the empirical error upper bound of the proposed ensemble method is formulated and proved.

Algorithm 1 AdaBoost

Input: Training set $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, weak classifier $\mathfrak{L}$, number of base classifiers T.

Output: Boosted classifier $H(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{T}\alpha_t h_t(\mathbf{x})\right)$.

1: **Initialization:** Probability distribution of training samples $\mathcal{D}_1(\mathbf{x}_i) = \frac{1}{N}$ $(i = 1, 2, \ldots, N)$.
2: **for** $t = 1 \rightarrow T$ **do**
3: $\quad h_t \leftarrow \mathfrak{L}(D, \mathcal{D}_t)$.
4: $\quad \epsilon_t \leftarrow \sum_{i, y_i \neq h_t(\mathbf{x}_i)} \mathcal{D}_t(\mathbf{x}_i)$.
5: $\quad$ **if** $\epsilon_t > 0.5$ **then**
6: $\quad\quad T \leftarrow t - 1$.
7: $\quad\quad$ **break.**
8: $\quad$ **end if**
9: $\quad \alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
10: $\quad \mathcal{D}_{t+1}(\mathbf{x}_i) = \frac{\mathcal{D}_t(\mathbf{x}_i)\exp\{-\alpha_t h_t(\mathbf{x}_i) y_i\}}{Z_t}$ $(i = 1, 2, \ldots, N)$, where Z_t is a normalization constant to ensure that $\sum_{i=1}^{N}\mathcal{D}_{t+1}(\mathbf{x}_i) = 1$.
11: **end for**

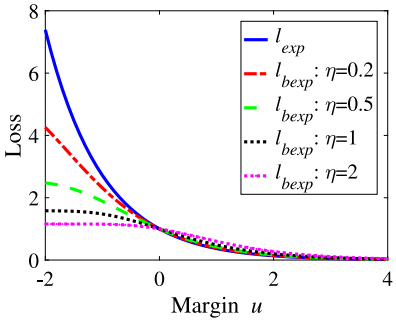


Fig. 1. Curves of the bounded exponential loss function for different values of the scale factor η.

3.1. Bounded exponential loss function

To make AdaBoost suitable for combining OCSVMs and more robust against outliers, the bounded exponential loss function is used to replace its exponential loss function. The bounded exponential loss function of margin u is defined as

$$\ell_{bexp}(u) = \xi\left[1 - \exp\left(-\eta\ell_{exp}(u)\right)\right], \tag{9}$$

where $\eta > 0$ is a multiplicative scale factor, while $\xi = \frac{1}{1-\exp(-\eta)}$ is a normalizing constant to ensure that $\ell_{bexp}(0) = 1$. Fig. 1 illustrates the bounded exponential loss functions with different η values. One can observe from Fig. 1 that the bounded exponential loss function is bounded, smooth, and nonconvex. Moreover, the shape of the bounded exponential loss function gets more gentle as the value of η grows larger. The left tails of the bounded exponential loss are comparatively lower than that of the traditional exponential loss. Therefore, using the bounded exponential loss to replace the conventional exponential loss in AdaBoost can alleviate the adverse impact of outliers.

In the following, the four properties of the bounded exponential loss function, including generalized form of the exponential loss function, insensitivity to outliers, equivalence with ℓ_0-norm and Fisher consistency, are stated in their corresponding propositions. Moreover, the correctness proof of these propositions is given in detail.

First, the relation between the exponential and the bounded exponential loss functions is included in Proposition 1.

Proposition 1. *For arbitrary margin u, $\lim_{\eta\to 0}\ell_{bexp}(u) = \ell_{exp}(u)$ holds. That is to say, the traditional exponential loss function can be considered as a special form of the proposed bounded exponential loss function.*

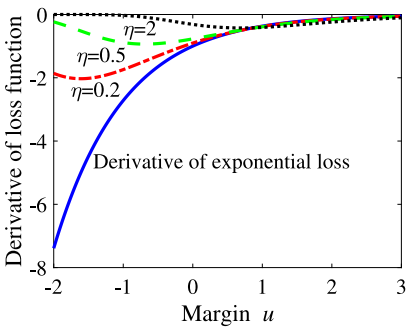


Fig. 2. The derivatives of the bounded exponential loss function (dotted lines) for different values of the scale factor η. The derivative of the exponential loss function (solid blue line) is also included.

Proof. The Taylor expansion of the bounded exponential loss function is given by

$$\ell_{bexp}(u) = \sum_{i=1}^{\infty} \frac{\xi \eta^i (-1)^{i+1} \ell_{exp}^i(u)}{i!}. \tag{10}$$

According to L'Hospital rule, we obtain

$$\lim_{\eta\to 0} \xi\eta^i = \lim_{\eta\to 0} \frac{\eta^i}{1-\exp(-\eta)} = \begin{cases} 1, & \text{if } i = 1, \\ 0, & \text{if } i \geq 2. \end{cases} \tag{11}$$

Utilizing the above two results, one can easily infer that $\lim_{\eta\to 0} \ell_{bexp}(u) = \ell_{exp}(u)$. □

Second, the insensitivity of the bounded exponential loss function to outliers is presented in Proposition 2.

Proposition 2. *For appropriately selected values of the scale factor η, the bounded exponential loss function is less sensitive to outliers in comparison with the conventional exponential loss function.*

Proof. The sensitivity of the bounded exponential loss function with respect to margin can be evaluated by the following derivative:

$$\frac{\partial \ell_{bexp}(u)}{\partial u} = \frac{\partial}{\partial u}\xi\left[1-\exp\left(-\eta\exp(-u)\right)\right] = -\frac{\eta\exp\left(-\eta\exp(-u)\right)}{1-\exp(-\eta)}\exp(-u). \tag{12}$$

Fig. 2 demonstrates the sensitivity (12) with respect to u for different values of the scale factor η. As is well known, outliers usually produce large negative values of margin. Hence, one can observe from Fig. 2 that the sensitivities produced by the bounded exponential loss function with different values of η are relatively smaller than the sensitivity generated by the conventional exponential loss function upon the outliers. □

Third, the behavior of the empirical risk achieved by utilizing the bounded exponential loss function is included in Proposition 3.

Proposition 3. *For $\eta \to \infty$, the empirical risk utilizing the bounded exponential loss function, i.e., $\hat{R}_{bexp}(f) = E[\ell_{bexp}(u)]$ behaves like the ℓ_0-norm.*

Proof.

$$\begin{aligned}\hat{R}_{bexp}(f) &= \xi\left[1-\frac{1}{n}\sum_{i=1}^{n}\exp\left(-\eta\exp(-u_i)\right)\right] \\ &= \frac{1-\frac{1}{n}\sum_{i=1}^{n}\exp\left(-\eta\exp(-u_i)\right)}{1-\exp(-\eta)}.\end{aligned} \tag{13}$$

Therefore,

$$\lim_{\eta\to\infty}\hat{R}_{bexp}(f) = \lim_{\eta\to\infty}\frac{1-\frac{1}{n}\sum_{i=1}^{n}\exp\left(-\eta\exp(-u_i)\right)}{1-\exp(-\eta)}$$

$$= 1-\frac{1}{n}\sum_{i=1}^{n}\lim_{\eta\to\infty}\exp\left(-\eta\exp(-u_i)\right) = \begin{cases} 0, & \text{if } u_i \to \infty, \\ 1, & \text{if } |u_i| < \infty. \end{cases} \tag{14}$$

Hence, the limit in (14) is fundamentally a count of the number of non-zero margins, or the ℓ_0-norm of the margin vector $\mathbf{u} = (u_1, u_2, \ldots, u_n)^T$ with $u_i = y_i f(\mathbf{x}_i)$, i.e.,

$$\lim_{\eta\to\infty}\hat{R}_{bexp}(f) = \|\mathbf{u}\|_0. \quad \square \tag{15}$$

As is well known, the Bayes decision is regarded as the optimal decision for classification problems. Moreover, Fisher consistency [28] is considered as an important concept to check whether the minimizer of a margin-based loss function leads to the Bayes optimal. To verify Fisher consistency of margin-based loss functions, Lin [28] established a theorem containing two assumptions. We restate the theorem as the following lemma.

Lemma 1 ([28]). *Let $g(\cdot)$ be a margin-based loss function. If $g(\cdot)$ satisfies the following two assumptions:*

1. *For $\forall u > 0$, $g(u) < g(-u)$;*
2. *$g'(0) \neq 0$ exists;*

where $u = yf(\mathbf{x})$ denotes the margin. Furthermore, if $E[g(yf(\mathbf{x}))|\mathbf{x}]$ has a global minimizer $f^*(\mathbf{x})$, then the loss function $g(\cdot)$ is Fisher consistent, which leads to $sign[f^*(\mathbf{x})] = sign[p(\mathbf{x}) - \frac{1}{2}]$ with the condition $p(\mathbf{x}) \neq \frac{1}{2}$, where $p(\mathbf{x}) = P(y = +1|\mathbf{x})$.

In Lemma 1, the first assumption guarantees that the sign of the minimizer $f^*(\mathbf{x})$ for $E[g(yf(\mathbf{x}))|\mathbf{x}]$ is the same with $sign[p(\mathbf{x}) - \frac{1}{2}]$. Moreover, the second assumption is used to ensure that $f^*(\mathbf{x}) \neq 0$.

Fourth and finally, the Fisher consistency of the proposed bounded exponential loss function is introduced in Proposition 4. The proof is deferred to A.

Proposition 4. *If the conditions $p(\mathbf{x}) \neq 1$ and $\exp(-f(\mathbf{x})) + \exp(f(\mathbf{x})) < \frac{2}{\eta}$ are satisfied, the bounded exponential loss function $\ell_{bexp}(u)$ is Fisher consistent.*

3.2. Update formulae

Similar to the traditional AdaBoost, an additive model $f_T(\mathbf{x})$ can also be expressed as a linear combination of T base classifiers for the ensemble of OCSVMs, i.e.,

$$f_T(\mathbf{x}) = \sum_{t=1}^{T}\alpha_t h_t(\mathbf{x}). \tag{16}$$

The first base classifier h_1 is generated by learning an OCSVM upon the training samples satisfying the probability distribution $\mathcal{D}_1$. In the tth iteration, the tth base classifier h_t is constructed by learning an OCSVM upon the training samples satisfying the probability distribution $\mathcal{D}_t$. The weight of h_t is also indicated by α_t. Moreover, the weight α_t is obtained by minimizing the bounded exponential loss function below.

$$\begin{aligned}&\ell_{bexp}(\alpha_t h_t|\mathcal{D}_t) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t}\left\{\xi\left[1-e^{-\eta e^{-\alpha_t h_t(\mathbf{x})}}\right]\right\} \\ &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t}\left\{\xi\left[1-e^{-\eta e^{-\alpha_t}}\right]\mathbb{I}(h_t(\mathbf{x})=1) + \xi\left[1-e^{-\eta e^{\alpha_t}}\right]\mathbb{I}(h_t(\mathbf{x})=-1)\right\} \\ &= \xi\left[1-e^{-\eta e^{-\alpha_t}}\right]P_{\mathbf{x}\sim\mathcal{D}_t}(h_t(\mathbf{x})=1) + \xi\left[1-e^{-\eta e^{\alpha_t}}\right]P_{\mathbf{x}\sim\mathcal{D}_t}(h_t(\mathbf{x})=-1) \\ &= \xi\left\{(1-\epsilon_t)\left[1-e^{-\eta e^{-\alpha_t}}\right] + \epsilon_t\left[1-e^{-\eta e^{\alpha_t}}\right]\right\}.\end{aligned} \tag{17}$$

where $\epsilon_t = P_{\mathbf{x}\sim\mathcal{D}_t}\left(h_t(\mathbf{x}) = -1\right)$. Note that the term y is not included in (17). The reason lies in that the class labels of training samples in the scenario of one-class classification are all +1.

The partial derivative of (17) with respect to α_t is given by

$$\frac{\partial \ell_{bexp}(\alpha_t h_t|\mathcal{D}_t)}{\partial \alpha_t} = \xi\eta\left[-(1-\epsilon_t)e^{[-\alpha_t-\eta e^{-\alpha_t}]} + \epsilon_t e^{[\alpha_t-\eta e^{\alpha_t}]}\right]. \tag{18}$$

Let $s(\alpha_t h_t|D_t) = \frac{\partial \ell_{bexp}(\alpha_t h_t|D_t)}{\partial \alpha_t}$. However, the analytic solution α_t of the equation $s(\alpha_t h_t|D_t) = 0$ cannot be deduced directly. Here, $s(\alpha_t h_t|D_t) = 0$ is solved by the Newton–Raphson approach [29]. Therefore, for the τth iteration, α_t^τ can be iteratively derived by

$$\alpha_t^\tau = \alpha_t^{\tau-1} - \frac{s(\alpha_t^{\tau-1} h_t|D_t)}{H(\alpha_t^{\tau-1} h_t|D_t)}, \tag{19}$$

where

$$\begin{aligned} H(\alpha_t^{\tau-1} h_t|D_t) &= \frac{\partial s(\alpha_t h_t|D_t)}{\alpha_t}|_{\alpha_t=\alpha_t^{\tau-1}} \\ &= \xi\eta\left[(1-\epsilon_t)e^{\left(-\alpha_t^{\tau-1}-\eta e^{-\alpha_t^{\tau-1}}\right)}\left(1-\eta e^{-\alpha_t^{\tau-1}}\right)\right. \\ &\quad \left. + \epsilon_t e^{\left(\alpha_t^{\tau-1}-\eta e^{\alpha_t^{\tau-1}}\right)}\left(1-\eta e^{\alpha_t^{\tau-1}}\right)\right]. \end{aligned} \tag{20}$$

After f_{t-1} has been obtained, the probability distribution of training samples needs to be updated to make the base classifier h_t rectify some mistakes of f_{t-1} in the next iteration. The perfect base classifier h_t can correct all the mistakes of f_{t-1} by minimizing the loss function below.

$$\begin{aligned} \ell_{bexp}(f_{t-1}+h_t|D) &= \mathbb{E}_{\mathbf{x}\sim D}\left[\xi\left(1-e^{-\eta e^{[-f_{t-1}(\mathbf{x})-h_t(\mathbf{x})]}}\right)\right] \\ &= \mathbb{E}_{\mathbf{x}\sim D}\left[\xi\left(1-e^{-\eta e^{-f_{t-1}(\mathbf{x})}e^{-h_t(\mathbf{x})}}\right)\right], \end{aligned} \tag{21}$$

where D denotes the distribution over the original training samples. Based on the Taylor expansion, (21) can be approximately expressed as

$$\begin{aligned} &\ell_{bexp}(f_{t-1}+h_t|D) \simeq \mathbb{E}_{\mathbf{x}\sim D}\left[\xi\left(1-e^{-\eta e^{-f_{t-1}(\mathbf{x})}\left[1-h_t(\mathbf{x})+\frac{h_t^2(\mathbf{x})}{2}\right]}\right)\right] \\ &= \mathbb{E}_{\mathbf{x}\sim D}\left[\xi\left(1-e^{-\eta e^{-f_{t-1}(\mathbf{x})}\left[\frac{3}{2}-h_t(\mathbf{x})\right]}\right)\right] \\ &= \mathbb{E}_{\mathbf{x}\sim D}\left[\xi\left(1-e^{-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x})}e^{\eta h_t(\mathbf{x})e^{-f_{t-1}(\mathbf{x})}}}\right)\right] \\ &\simeq \mathbb{E}_{\mathbf{x}\sim D}\left[\xi\left(1-e^{-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x})}\left[1+\eta h_t e^{-f_{t-1}(\mathbf{x})}+\frac{\eta^2 e^{-2f_{t-1}(\mathbf{x})}}{2}\right]}\right)\right]. \end{aligned} \tag{22}$$

Therefore, the ideal base classifier satisfies

$$\begin{aligned} h_t(\mathbf{x}) &= \arg\min_h \ell_{bexp}(f_{t-1}+h|D) \\ &= \arg\min_h \mathbb{E}_{\mathbf{x}\sim D}\left\{\xi\left[1-e^{-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x})}}\left(1+\eta h(\mathbf{x})e^{-f_{t-1}(\mathbf{x})}+\frac{\eta^2 e^{-2f_{t-1}(\mathbf{x})}}{2}\right)\right]\right\} \\ &= \arg\min_h \mathbb{E}_{\mathbf{x}\sim D}\left\{\xi\left[1-e^{-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x})}}\left(\eta h(\mathbf{x})e^{-f_{t-1}(\mathbf{x})}\right)\right]\right\} \\ &= \arg\min_h \mathbb{E}_{\mathbf{x}\sim D}\left[e^{\left[-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x})}-f_{t-1}(\mathbf{x})\right]}h(\mathbf{x})\right] \\ &= \arg\min_h \mathbb{E}_{\mathbf{x}\sim D}\left\{\frac{e^{\left[-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x})}-f_{t-1}(\mathbf{x})\right]}h(\mathbf{x})}{\mathbb{E}_{\mathbf{x}\sim D}e^{\left[-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x})}-f_{t-1}(\mathbf{x})\right]}}\right\}. \end{aligned} \tag{23}$$

Let $\mathcal{D}_t$ denote a probability distribution satisfying

$$\mathcal{D}_t(\mathbf{x}) = \frac{\mathcal{D}(\mathbf{x})\exp\left(-\frac{3}{2}\eta\exp(-f_{t-1}(\mathbf{x}))-f_{t-1}(\mathbf{x})\right)}{\mathbb{E}_{\mathbf{x}\sim D}\left[\exp\left(-\frac{3}{2}\eta\exp(-f_{t-1}(\mathbf{x}))-f_{t-1}(\mathbf{x})\right)\right]}. \tag{24}$$

According to the definition of mathematical expectation, one can easily get that

$$\begin{aligned} h_t(\mathbf{x}) &= \arg\max_h \mathbb{E}_{\mathbf{x}\sim D}\left\{\frac{e^{\left[-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x})}-f_{t-1}(\mathbf{x})\right]}h(\mathbf{x})}{\mathbb{E}_{\mathbf{x}\sim D}\left[e^{\left[-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x})}-f_{t-1}(\mathbf{x})\right]}\right]}\right\} \\ &= \arg\max_h \mathbb{E}_{\mathbf{x}\sim D_t}\left[h(\mathbf{x})\right]. \end{aligned} \tag{25}$$

Because $h(\mathbf{x}) \in \{-1,+1\}$, so we have

$$h(\mathbf{x}) = 1 - 2\mathbb{I}(h(\mathbf{x}) \neq +1). \tag{26}$$

Hence, the ideal base classifier satisfies

$$h_t(\mathbf{x}) = \arg\min_h \mathbb{E}_{\mathbf{x}\sim D_t}\left[\mathbb{I}(h(\mathbf{x}) \neq +1)\right]. \tag{27}$$

It can be deduced from (27) that the ideal base classifier h_t over the probability distribution $\mathcal{D}_t$ should minimize the classification error.

Considering the relation between $\mathcal{D}_t$ and $\mathcal{D}_{t+1}$, we can obtain

$$\begin{aligned} \mathcal{D}_{t+1}(\mathbf{x}) &= \frac{\mathcal{D}(\mathbf{x})e^{\left[-\frac{3}{2}\eta e^{-f_t(\mathbf{x})}-f_t(\mathbf{x})\right]}}{\mathbb{E}_{\mathbf{x}\sim D}\left\{e^{\left[-\frac{3}{2}\eta e^{-f_t(\mathbf{x})}-f_t(\mathbf{x})\right]}\right\}} = \frac{\mathcal{D}(\mathbf{x})e^{\left\{-\frac{3}{2}\eta e^{\left[-f_{t-1}(\mathbf{x})-\alpha_t h_t(\mathbf{x})\right]}-f_t(\mathbf{x})\right\}}}{\mathbb{E}_{\mathbf{x}\sim D}\left\{e^{\left[-\frac{3}{2}\eta e^{-f_t(\mathbf{x})}-f_t(\mathbf{x})\right]}\right\}} \\ &= \mathcal{D}(\mathbf{x})\left\{e^{\left[-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x})}-f_{t-1}(\mathbf{x})\right]}\right\}^{\left[e^{-\alpha_t h_t(\mathbf{x})}\right]}\frac{\left[e^{f_{t-1}(\mathbf{x})}\right]^{\left[e^{-\alpha_t h_t(\mathbf{x})}-1\right]}e^{-\alpha_t h_t(\mathbf{x})}}{\mathbb{E}_{\mathbf{x}\sim D}\left\{e^{\left[-\frac{3}{2}\eta e^{-f_t(\mathbf{x})}-f_t(\mathbf{x})\right]}\right\}} \\ &= \mathcal{D}(\mathbf{x})e^{\left[-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x})}-f_{t-1}(\mathbf{x})\right]}\frac{\left[e^{-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x})}}\right]^{\left[e^{-\alpha_t h_t(\mathbf{x})}-1\right]}e^{-\alpha_t h_t(\mathbf{x})}}{\mathbb{E}_{\mathbf{x}\sim D}\left\{e^{\left[-\frac{3}{2}\eta e^{-f_t(\mathbf{x})}-f_t(\mathbf{x})\right]}\right\}} \\ &= \mathcal{D}_t(\mathbf{x})\left[e^{-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x})}}\right]^{\left[e^{-\alpha_t h_t(\mathbf{x})}-1\right]}e^{-\alpha_t h_t(\mathbf{x})}\frac{\mathbb{E}_{\mathbf{x}\sim D}\left\{e^{\left[-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x})}-f_{t-1}(\mathbf{x})\right]}\right\}}{\mathbb{E}_{\mathbf{x}\sim D}\left\{e^{\left[-\frac{3}{2}\eta e^{-f_t(\mathbf{x})}-f_t(\mathbf{x})\right]}\right\}} \\ &= \frac{\mathcal{D}_t(\mathbf{x})\left\{e^{\left[-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x})}\right]}\right\}^{\left[e^{-\alpha_t h_t(\mathbf{x})}-1\right]}e^{-\alpha_t h_t(\mathbf{x})}}{Z_t}, \end{aligned} \tag{28}$$

where $Z_t = \frac{\mathbb{E}_{\mathbf{x}\sim D}\left[\exp\left(-\frac{3}{2}\eta\exp(-f_t(\mathbf{x}))-f_t(\mathbf{x})\right)\right]}{\mathbb{E}_{\mathbf{x}\sim D}\left[\exp\left(-\frac{3}{2}\eta\exp(-f_{t-1}(\mathbf{x}))\right)-f_{t-1}(\mathbf{x})\right]}$ is a normalization factor to make sure that $\mathcal{D}_{t+1}$ is a probability distribution.

3.3. Algorithm description

The training process of the bounded exponential loss function based AdaBoost ensemble of OCSVMs is summarized in Algorithm 2. The computational cost for constructing an OCSVM is $\mathcal{O}(N^3)$ [30] with N denotes the number of training samples. In each loop, the computational complexities of error rate ϵ_t, weight of base classifier α_t, and probability distribution $\mathcal{D}_t$ are respectively $\mathcal{O}(N)$, $\mathcal{O}(I)$, and $\mathcal{O}(N^2)$. Hence, the computational complexity of Algorithm 2 is $\mathcal{O}(T(N^3+N^2+N+I))$. Neglecting lower order items, one can eventually know that the computational cost of the whole algorithm is $\mathcal{O}(TN^3)$.

Algorithm 2 Bounded exponential loss function based AdaBoost ensemble of OCSVMs

Input: Training set $D = \{\mathbf{x}_i\}_{i=1}^N$, number of base classifiers T, number of iterations I.

Output: Boosted classifier $H(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(\mathbf{x})\right)$.

1: **Initialization:** Probability distribution of training samples $\mathcal{D}_1(\mathbf{x}_i) = \frac{1}{N}$ $(i = 1, 2, \ldots, N)$.
2: **for** $t = 1 \to T$ **do**
3: $\quad h_t \leftarrow \text{OCSVM}(D, \mathcal{D}_t)$.
4: $\quad \epsilon_t \leftarrow P_{\mathbf{x}\sim\mathcal{D}_t}(h_t(\mathbf{x}) = -1)$.
5: $\quad$ **for** $\tau = 1 \to I$ **do**
6: $\quad\quad \alpha_t^\tau \leftarrow \alpha_t^{\tau-1} - \frac{s(\alpha_t^{\tau-1} h_t|D_t)}{H(\alpha_t^{\tau-1} h_t|D_t)}$.
7: $\quad$ **end for**
8: $\quad \mathcal{D}_{t+1}(\mathbf{x}) = \frac{\mathcal{D}_t(\mathbf{x})\left[e^{-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x})}}\right]^{\left[e^{-\alpha_t h_t(\mathbf{x})}-1\right]}e^{-\alpha_t h_t(\mathbf{x})}}{z_t}$.
9: **end for**

3.4. Empirical error upper bound of BELF-AEOCSVMs

For convenience, the bounded exponential loss function based AdaBoost ensemble of OCSVMs is shorten as BELF-AEOCSVMs. There are several approaches to evaluate the performance of the traditional AdaBoost, such as error bound estimation [11] and training error boundary analysis [31]. Inspired by the latter, the empirical error upper bound of BELF-AEOCSVMs is estimated, which is summarized in Proposition 5.

Proposition 5. *All the base classifiers in the proposed ensemble are OCSVM. The weighted strategy is utilized to combine the trained base classifiers in the ensemble. Suppose the error rates of the T base classifiers are $\epsilon_1, \epsilon_2, \ldots, \epsilon_T$. Moreover, $\mathcal{D}_1$ is the initial probability distribution of training samples. The empirical error upper bound of the ensemble classifier H with respect to $\mathcal{D}_1$ is given by*

$$P_{\mathbf{x}_i\sim\mathcal{D}_1}[H(\mathbf{x}_i)=-1] < \frac{\prod_{t=1}^{T}[(e^2-1)\epsilon_t+1]}{e^T}. \tag{29}$$

Proof. According to Algorithm 2, $\mathcal{D}_{T+1}$ is defined as

$$\begin{aligned}\mathcal{D}_{T+1}(\mathbf{x}_i) &= \mathcal{D}_1(\mathbf{x}_i)\frac{\left[e^{-\frac{3}{2}\eta e^{-f_0(\mathbf{x}_i)}}\right]^{\left[e^{-\alpha_1 h_1(\mathbf{x}_i)}-1\right]}e^{-\alpha_1 h_1(\mathbf{x}_i)}}{Z_1}\\ &\cdots\frac{\left[e^{-\frac{3}{2}\eta e^{-f_{T-1}(\mathbf{x}_i)}}\right]^{\left[e^{-\alpha_T h_T(\mathbf{x}_i)}-1\right]}e^{-\alpha_T h_T(\mathbf{x}_i)}}{Z_T}\\ &= \mathcal{D}_1(\mathbf{x}_i)e^{-f_T(\mathbf{x}_i)}\frac{\prod_{t=1}^{T}\left[e^{-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x}_i)}}\right]^{\left[e^{-\alpha_t h_t(\mathbf{x}_i)}-1\right]}e^{-\alpha_t h_t(\mathbf{x}_i)}}{\prod_{t=1}^{T}Z_t}.\end{aligned} \tag{30}$$

It is known that $\eta > 0$ and $e^{-f_{t-1}(\mathbf{x}_i)} > 0$, which leads to $0 < e^{-\frac{3}{2}\eta\exp(f_{t-1}(\mathbf{x}_i))} < 1$. Here we assume that $0 \le \alpha_t \le 1$. This assumption can be easily obtained through dividing all the weights $\{\alpha_1, \alpha_2, \ldots, \alpha_T\}$ by their sum $\sum_{t=1}^{T}\alpha_t$. Therefore,

$$\begin{aligned}&\mathcal{D}_{T+1}(\mathbf{x}_i)\prod_{t=1}^{T}Z_t = \mathcal{D}_1(\mathbf{x}_i)e^{-f_T(\mathbf{x}_i)}\prod_{t=1}^{T}\left[e^{-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x}_i)}}\right]^{\left[e^{-\alpha_t h_t(\mathbf{x}_i)}-1\right]}\\ &\ge \mathcal{D}_1(\mathbf{x}_i)e^{-f_T(\mathbf{x}_i)}\prod_{t=1}^{T}\left[e^{-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x}_i)}}\right]^{\left(\frac{1}{e}-1\right)} > \mathcal{D}_1(\mathbf{x}_i)\exp(-f_T(\mathbf{x}_i)).\end{aligned} \tag{31}$$

Since $H(\mathbf{x}) = sign(f_T(\mathbf{x}))$, if $H(\mathbf{x}) = -1$, then $f_T(\mathbf{x}) \le 0$, which implies that $\exp\left(-f_T(\mathbf{x})\right) \ge 1$. That is, $\mathbb{I}(H(\mathbf{x})=-1) \le \exp\left(-f_T(\mathbf{x})\right)$. Hence, the empirical error is

$$\begin{aligned}P_{\mathbf{x}_i\sim\mathcal{D}_1}[H(\mathbf{x}_i)=-1] &= \sum_{i=1}^{N}\mathcal{D}_1(\mathbf{x}_i)\mathbb{I}(H(\mathbf{x}_i)=-1) \le \sum_{i=1}^{N}\mathcal{D}_1(\mathbf{x}_i)\exp\left(-f_T(\mathbf{x}_i)\right)\\ &< \sum_{i=1}^{N}\mathcal{D}_{T+1}(\mathbf{x}_i)\prod_{t=1}^{T}Z_t = \prod_{t=1}^{T}Z_t.\end{aligned} \tag{32}$$

The last equality in (32) uses the fact that $\mathcal{D}_{T+1}$ is a distribution (which sums to 1). Furthermore,

$$\begin{aligned}Z_t &= \frac{\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\exp\left(-\frac{3}{2}\eta\exp(-f_t(\mathbf{x}))-f_t(\mathbf{x})\right)\right]}{\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\exp\left(-\frac{3}{2}\eta\exp(-f_{t-1}(\mathbf{x}))\right)-f_{t-1}(\mathbf{x})\right]}\\ &= \sum_{i=1}^{N}\mathcal{D}_t(\mathbf{x}_i)\left[e^{-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x}_i)}}\right]^{\left[e^{-\alpha_t h_t(\mathbf{x}_i)}-1\right]}e^{-\alpha_t h_t(\mathbf{x}_i)}\\ &\le \sum_{i=1}^{N}\mathcal{D}_t(\mathbf{x}_i)\left[e^{-\frac{3}{2}\eta e^{-f_{t-1}(\mathbf{x}_i)}}\right]^{(e-1)}e^{-\alpha_t h_t(\mathbf{x}_i)} < \sum_{i=1}^{N}\mathcal{D}_t(\mathbf{x}_i)e^{-\alpha_t h_t(\mathbf{x}_i)}\\ &= \sum_{i:h_t(\mathbf{x}_i)=1}\mathcal{D}_t(\mathbf{x}_i)e^{-\alpha_t} + \sum_{i:h_t(\mathbf{x}_i)=-1}\mathcal{D}_t(\mathbf{x}_i)e^{\alpha_t}\\ &= e^{-\alpha_t}(1-\epsilon_t)+e^{\alpha_t}\epsilon_t \le e^{-1}(1-\epsilon_t)+e\epsilon_t = \frac{\left(e^2-1\right)\epsilon_t+1}{e}.\end{aligned} \tag{33}$$

Plugging (33) into (32), we can finally get (29). □

4. Experimental results

In this section, BELF-AEOCSVMs is compared with its relevant approaches on one artificial data set, sixteen UCI benchmark data sets and one handwritten digit data set. The base classifiers in the following ensemble methods are all OCSVM. The Gaussian kernel function $K(\mathbf{x},\mathbf{y}) = \exp\{-\gamma\|\mathbf{x}-\mathbf{y}\|^2\}$ is chosen. The geometric mean (g-mean) is adopted to evaluate the performances of the following approaches.

4.1. Artificial data set

The description of the artificial data set is given below.

Square-Outlier: 200 target samples are randomly generated in the square $\{(x,y)|x\in[0.4,2.6], y\in[0.4,0.6]\cup[2.4,2.6]\}\cup\{(x,y)|x\in[0.4,0.6]\cup[2.4,2.6], y\in[0.4,2.6]\}$, while 50 outliers are randomly chosen from the area $\{(x,y)|x,y\in[0,3]\}$.

Towards *Square-Outlier*, the width parameter of the Gaussian kernel function and the trade-off parameter for the single OCSVM are assigned with $\gamma = 2^5$ and $\nu = 0.2$, respectively. The values of γ and σ for the base classifiers in the traditional AdaBoost based ensemble of OCSVMs (AEOCSVMs), weighted bagging based ensemble of OCSVMs (WBEOCSVMs) [32], and BELF-AEOCSVMs are all the same with their counterparts of the single OCSVM. The number of OCSVMs in AEOCSVMs, WBEOCSVMs and BELF-AEOCSVMs are all 20. For WBEOCSVMs, the width parameter σ of the weighted kernel density estimator and the number of iterations for updating the probability weights of all samples are assigned with 1 and 5, respectively. Moreover, for BELF-AEOCSVMs, the scale factor η and the number of iterations for updating the weights of base classifiers are taken as 0.1 and 20, respectively. The outcomes of the four methods are illustrated in Fig. 3.

One can observed from Fig. 3 that BELF-AEOCSVMs achieves the best anti-outlier ability in comparison with the other three approaches on *Square-Outlier*.

4.2. UCI benchmark data sets

Besides OCSVM, AEOCSVMs and WBEOCSVMs, the proposed BELF-AEOCSVMs is also compared with random subspace method based ensemble of OCSVMs (RSMEOCSVMs) [33] and clustering based ensemble of OCSVMs (CEOCSVMs) [34] on the sixteen benchmark data sets selected from the UCI machine learning repository [35]. However, all the sixteen data sets are designed for binary classification. To make them suitable for one-class classification, the samples in one class are utilized as target samples and the samples in the other class as non-target samples. Furthermore, 70% of the target samples and 5% of the non-target samples are randomly selected to generate the training set. Note that the labels of non-target samples in the training set are altered from negative to positive to make these non-target samples play the role of outliers. The rest 30% target samples and 95% non-target samples are used as the test set. The information of the sixteen data sets is included in Table 1.

The parameters γ and ν for OCSVM are exhaustively searched within the domains $\{2^{-6}, 2^{-5}, \ldots, 2^6\}$ and $\{0.1, 0.2, \ldots, 1\}$, respectively. The optimal values of γ and ν for OCSVM on the sixteen UCI benchmark data sets are tabulated in Table 2. Towards the five ensemble methods, γ and ν for their base classifiers are all designated with the same values as those of OCSVM. The width parameter of the weighted kernel density estimator σ for WBEOCSVMs is selected in the domain $\{1, 2, 4, 8, 16, 32\}$. The percentage of the remained features for RSMEOCSVMs is fixed at 75% and the majority voting rule is adopted. For CEOCSVMs, the fuzzy c-means clustering algorithm is utilized to partition the input space. The domain of the scale factor η for BELF-AEOCSVMs is $\{0.01, 0.05, 0.1, 0.2, 0.5, 1, 1.5, 2\}$. The number of

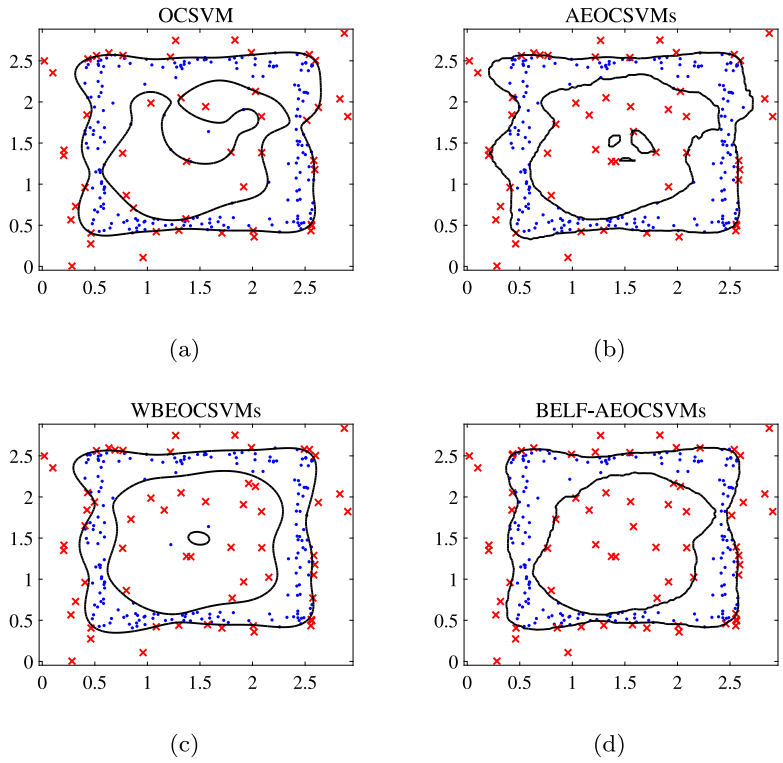


Fig. 3. The outcomes of the four methods upon *Square-Outlier*. (a) OCSVM with g-mean 0.7060. (b) AEOCSVMs with g-mean 0.7547. (c) WBEOCSVMs with g-mean 0.7669. (d) BELF-AEOCSVMs with g-mean 0.7931.

Table 1
The information of the sixteen UCI benchmark data sets.

Data sets	N_{tar}	$N_{non-tar}$	N_{fea}	N_{tr}	N_{ts}
Banana	2376	2924	2	1809	3491
Blood Transfusion	178	570	4	154	594
Cancer	239	444	9	189	494
Diabetis	268	500	8	213	555
Flare Solar	94	50	9	71	73
Hepatitis	123	32	19	91	64
Hill Valley	301	305	100	226	380
Housing	245	261	13	185	321
Pima	268	500	8	213	555
Ringnorm	3664	3736	20	2752	4648
Splice	1344	1647	60	1023	1968
Thyorid	65	150	5	54	161
Twonorm	3703	3697	20	2777	4623
Waveform	1647	3353	21	1321	3679
Wdbc	212	357	9	166	403
Wholesale Customers	298	142	7	216	224

Note: N_{tar}–Number of target samples; $N_{non-tar}$–Number of non-target samples; N_{fea}–Number of features; N_{tr}–Number of training samples; N_{ts}–Number of testing samples.

Table 2
The parameter settings of the six approaches upon the sixteen UCI benchmark data sets.

Data sets	γ	ν	σ	η
Banana	2^4	0.3	2	0.05
Blood Transfusion	2^3	0.3	4	0.05
Cancer	2^{-5}	0.4	16	0.01
Diabetis	2^{-5}	0.5	1	0.2
Flare Solar	2^{-2}	0.3	4	1.5
Hepatitis	2^{-6}	0.4	4	0.01
Hill Valley	2^5	0.4	1	0.01
Housing	2^{-5}	0.2	2	0.1
Pima	2^{-1}	0.5	16	0.5
Ringnorm	2^2	0.2	16	0.2
Splice	2^{-5}	0.4	16	0.01
Thyorid	2^2	0.4	4	0.05
Twonorm	2^{-6}	0.3	32	0.01
Waveform	2^{-1}	0.4	4	0.5
Wdbc	2^{-6}	0.4	8	0.01
Wholesale Customers	2^2	0.2	1	0.2

base classifiers for the ensemble approaches is 20. The max-epochs for updating the probability weights of training samples for WBOCSVMs and the weights of base classifiers for BELF-AEOCSVMs are 5 and 20, respectively. Finally, the settings of σ for WBEOCSVMs and η for BELF-AEOCSVMs on the sixteen data sets are also included in Table 2.

It should be mentioned here that there are no non-target data in the training set for the scenario of one-class classification. Hence, the traditional strategy of parameter selection for binary or multi-class classification, e.g., k-fold cross validation can only perform on the target data but not on the non-target data. If one chooses the parameter value to minimize the validation error, the obtained one-class classifier may label all the testing samples as the target data. Therefore, for the scenario of one-class classification, the researchers usually split the whole data set into the training and test sets without the validation set, then exhaustively search for the best parameter combination values [30,36].

The average testing results of 20 trails for the six methods upon the sixteen benchmark data sets are tabulated in Table 3. The standard deviations for the 20 trials upon each data set are included in the table. The training costs of the six methods in one trial on each data set are also shown in Table 3. In addition, the paired T-test and Wilcoxon rank-sum test are utilized to examine whether the performance enhancement obtained by BELF-AEOCSVMs over the other methods is statistically significant.

The outcomes in Table 3 show that BELF-AEOCSVMs is statistically different from its related methods on all the sixteen data sets except *Hill Valley*. The generalization ability of BELF-AEOCSVMs is better than the other five approaches on twelve out of sixteen data sets. Considering

Table 3
The testing results of the six approaches upon the sixteen UCI benchmark data sets. The best outcomes are emphasized in bold.

Data sets	OCSVM G-mean/Time(s) P_T, P_W	WBEOCSVMs G-mean/Time P_T, P_W	RSMEOCSVMs G-mean/Time P_T, P_W	CEOCSVMs G-mean/Time P_T, P_W	AEOCSVMs G-mean/Time P_T, P_W	BELF-AEOCSVMs G-mean/Time
Banana	0.7300 ± 0.0057/**0.06** 5.76E−034,3.65E−008	0.7412 ± 0.0044/23.10 4.76E−023,3.65E−008	0.7300 ± 0.0057/0.79 5.76E−034,3.65E−008	0.7162 ± 0.0030/0.73 1.01E−024,1.15E−009	0.7357 ± 0.0065/1.27 2.24E−023,3.65E−008	**0.7534** ± **0.0053**/1.28
Blood Transfusion	0.5911 ± 0.0107/**0.002** 6.66E−013,4.34E−008	0.5847 ± 0.0011/0.30 7.91E−006,4.34E−008	0.5373 ± 0.0022/0.01 5.67E−007,3.65E−008	0.5088 ± 0.0092/0.08 1.28E−024,1.15E−009	0.5958 ± 0.0041/0.03 0.0148,4.34E−008	**0.6013** ± **0.0134**/0.03
Cancer	0.8242 ± 0.0051/**0.002** 2.82E−011,4.34E−008	0.8102 ± 0.0049/0.37 1.65E−018,3.65E−008	0.7575 ± 0.0074/0.02 9.21E−008,4.34E−008	0.7751 ± 0.0016/0.15 1.51E−031,1.15E−009	0.8218 ± 0.0039/0.03 3.17E−010,4.34E−008	**0.8319** ± **0.0077**/0.03
Diabetes	0.4656 ± 0.0202/**0.002** 0.0001,9.75E−007	0.4771 ± 0.0100/0.43 0.0002,4.34E−008	**0.5338** ± **0.0088**/0.02 3.63E−014,3.65E−008	0.5043 ± 0.0213/0.17 1.57E−005,9.75E−007	0.5094 ± 0.0202/0.04 0.0005,4.34E−008	0.5185 ± 0.0299/0.04
Flare Solar	0.3858 ± 0.0025/**0.002** 9.70E−052,1.15E−009	0.4151 ± 0.0046/0.08 2.55E−033,1.15E−009	0.3496 ± 0.0003/0.01 3.89E−037,1.15E−009	0.3940 ± 0.0029/0.04 8.08E−045,1.15E−009	0.3939 ± 0.0020/0.01 4.08E−057,1.15E−009	**0.5275** ± **0.0022**/0.01
Hepatitis	0.5934 ± 0.0196/**0.002** 1.92E−022,3.65E−008	0.5810 ± 0.0046/0.12 4.99E−005,4.34E−008	0.5958 ± 0.0202/0.01 9.58E−018,3.65E−008	0.4860 ± 0.0096/1.21 2.42E−023,4.34E−008	0.5940 ± 0.0168/0.02 3.63E−019,3.65E−008	**0.6077** ± **0.0184**/0.02
Hill Valley	0.4735 ± 0.0080/**0.01** 3.13E−012,4.34E−008	0.4861 ± 0.0003/0.78 0.1038,4.34E−008	0.4820 ± 0.0065/0.09 3.15E−006,4.34E−008	**0.5189** ± **0.0092**/0.22 1.07E−009,4.34E−008	0.4799 ± 0.0059/0.19 8.07E−014,3.98E−008	0.4844 ± 0.0048/0.20
Housing	0.6187 ± 0.0127/**0.002** 4.87E−027,3.65E−008	**0.6928** ± **0.0015**/0.48 3.38E−014,3.65E−008	0.6262 ± 0.0046/0.01 2.38E−011,4.34E−008	0.5447 ± 0.0090/0.10 8.15E−022,1.15E−009	0.6211 ± 0.0108/0.03 7.18E−017,3.65E−008	0.6387 ± 0.0136/0.03
Pima	0.4860 ± 0.0083/**0.002** 4.15E−020,1.15E−009	0.4876 ± 0.0027/0.55 5.40E−026,1.15E−009	0.5001 ± 0.0024/0.02 2.59E−018,1.15E−009	0.4945 ± 0.0053/0.25 9.14E−027,1.15E−009	0.5170 ± 0.0009/0.04 2.78E−005,4.34E−008	**0.5213** ± **0.0045**/0.04
Ringnorm	0.9331 ± 0.0007/**0.36** 1.38E−048,1.15E−009	0.9299 ± 0.0001/82.53 1.27E−046,1.15E−009	0.9288 ± 0.0022/5.53 2.48E−009,4.34E−008	0.9104 ± 0.0074/0.33 7.71E−045,1.15E−009	0.9468 ± 0.0037/9.98 1.07E−015,3.65E−008	**0.9475** ± **0.0038**/10.02
Splice	0.6682 ± 0.0027/**0.06** 6.74E−015,3.65E−008	0.6670 ± 0.0011/11.75 2.70E−008,4.34E−008	0.6595 ± 0.0008/0.95 6.83E−027,3.65E−008	0.6603 ± 0.0019/0.19 1.72E−024,4.34E−008	**0.6710** ± **0.0027**/2.04 9.91E−014,4.34E−008	0.6696 ± 0.0024/1.98
Thyroid	0.3995 ± 0.0154/**0.001** 5.48E−019,1.15E−009	0.3644 ± 0.0220/0.04 2.41E−018,1.15E−009	0.4461 ± 0.0039/0.005 2.20E−023,1.15E−009	0.4702 ± 0.0340/0.05 4.72E−019,1.15E−009	0.4988 ± 0.0355/0.01 1.43E−011,4.34E−008	**0.5870** ± **0.0078**/0.01
Twonorm	0.7683 ± 0.0022/**0.15** 1.59E−010,4.34E−008	0.7694 ± 0.0030/79.53 1.19E−019,3.65E−008	0.7700 ± 0.0020/2.59 0.0022,4.34E−008	0.6025 ± 0.0001/0.35 3.27E−034,1.15E−009	0.7686 ± 0.0026/5.25 4.15E−013,4.34E−008	**0.7709** ± **0.0032**/5.26
Waveform	0.7516 ± 0.0028/**0.05** 7.15E−006,4.34E−008	0.7503 ± 0.0026/17.64 3.34E−009,4.34E−008	0.7366 ± 0.0013/0.81 3.36E−038,1.15E−009	0.6448 ± 0.0178/1.10 4.43E−015,1.15E−009	0.7476 ± 0.0018/1.61 3.20E−010,4.34E−008	**0.7528** ± **0.0037**/1.55
Wdbc	0.7744 ± 0.0075/**0.001** 8.07E−014,3.97E−008	0.7539 ± 0.0012/0.28 5.76E−012,4.34E−008	0.7486 ± 0.0089/0.01 1.88E−027,3.65E−008	0.5673 ± 0.0085/0.19 8.90E−048,1.15E−009	0.7734 ± 0.0061/0.03 1.02E−005,4.34E−008	**0.7757** ± **0.0077**/0.03
Wholesale Customers	0.7512 ± 0.0148/**0.002** 7.96E−015,3.65E−008	0.7758 ± 0.0156/0.42 2.20E−010,4.34E−008	0.7064 ± 0.0113/0.02 2.51E−009,3.65E−008	0.6990 ± 0.0152/0.08 5.06E−018,1.15E−009	0.7660 ± 0.0138/0.03 2.20E−010,4.34E−008	**0.7910** ± **0.0230**/0.03

Note: s–Second; P_T–P-value for paired T-test; P_W–P-value for Wilcoxon rank-sum test.

the average g-mean values, the values of standard deviation in Table 3 exhibit that BELF-AEOCSVMs is more stable than its related methods on the foresaid twelve data sets. One can further obtain the following two observations from the outcomes in Table 3.

- Compared to AEOCSVMs, BELF-AEOCSVMs demonstrates better generalization performance upon all the sixteen data sets except *Splice*. Hence, replacing the conventional exponential loss function in AEOCSVMs by the proposed bounded exponential loss function can improve its robustness against outliers.
- In comparison with OCSVM, WBEOCSVMs obtains better performance upon eight out of sixteen data sets. AEOCSVMs gets better performance on thirteen out of sixteen data sets, the average of all improvement g-mean values upon these thirteen data sets is 0.0180. RSMEOCSVMs produces better performance on seven out of sixteen data sets, while CEOCSVMs generates better performance on five out of sixteen data sets. BELF-AEOCSVMs gets better performance on all the sixteen data sets and the average of all improvement g-mean values upon all the data sets is 0.0353. Therefore, WBEOCSVMs, RSMEOCSVMs and CEOCSVMs produce unsatisfying outcomes mainly because that the training samples are polluted by outliers. In comparison with AEOCSVMs, BELF-AEOCSVMs achieves better results. It is thus again verified that replacing exponential loss function with the proposed bounded exponential loss function can improve the anti-outlier ability of AEOCSVMs.

As for the training costs of the six methods, the following observations can be drawn from Table 3. First, OCSVM is the fastest on all the sixteen data sets, while WBEOCSVMs is the slowest on fourteen data sets. Second, RSMEOCSVMs is the fastest among the five ensemble methods. BELF-AEOCSVMs achieves faster training speed than CEOCSVMs on eleven data sets, while gets the same training time as AEOCSVMs on ten data sets. Third, the training costs of BELF-AEOCSVMs on *Ringnorm* and *Twonorm* are much higher than those on the other data sets. Hence, the training cost of BELF-AEOCSVMs increases greatly as the number of training samples increases.

Moreover, the relations between the generalization performance of the six approaches and the ratios of outliers occupying all the training samples on the four data sets are shown in Fig. 4. The value of ratio increases from 5% to 30% with step size 5%. From Fig. 4, we can deduce the following two outcomes.

- The generalization performances of the six methods decrease as the value of ratio increases. Compared to the other five approaches, BELF-AEOCSVMs gets the better outcomes upon the four data sets.
- In comparison with OCSVM, WBEOCSVMs demonstrates better performance only on *Housing*, RSMEOCSVMs and AEOCSVMs obtain approximate performances on all the four data sets, and CEOCSVMs achieves worse performances on *Banana*, *Housing*, and *Wdbc*. In contrast, BELF-AEOCSVMs gets better outcomes upon all the four benchmark data sets.

Furthermore, the impact of different parameter settings on the performance of BELF-AEOCSVMs is examined on the four data sets. The width parameter γ, the trade-off parameter ν, and the scale constant η of BELF-AEOCSVMs are investigated. The ranges for γ, ν and η are respectively $\{2^{-6}, 2^{-5}, \ldots, 2^{6}\}$, $\{0.1, 0.2, \ldots, 0.9\}$ and $\{0.01, 0.05, 0.1, 0.2, 0.5, 1, 1.5, 2\}$. Each parameter varies within the above domain whilst the rest two parameters remain unchanged with their values directly derived from Table 2. 20 trials are repeated for each setting of γ, ν, and η. Fig. 5(a) illustrates the effect of γ on the testing performance of BELF-AEOCSVMs upon the four data sets when ν and η keep unchanged as their values derived from Table 2. Fig. 5(b) demonstrates the impact

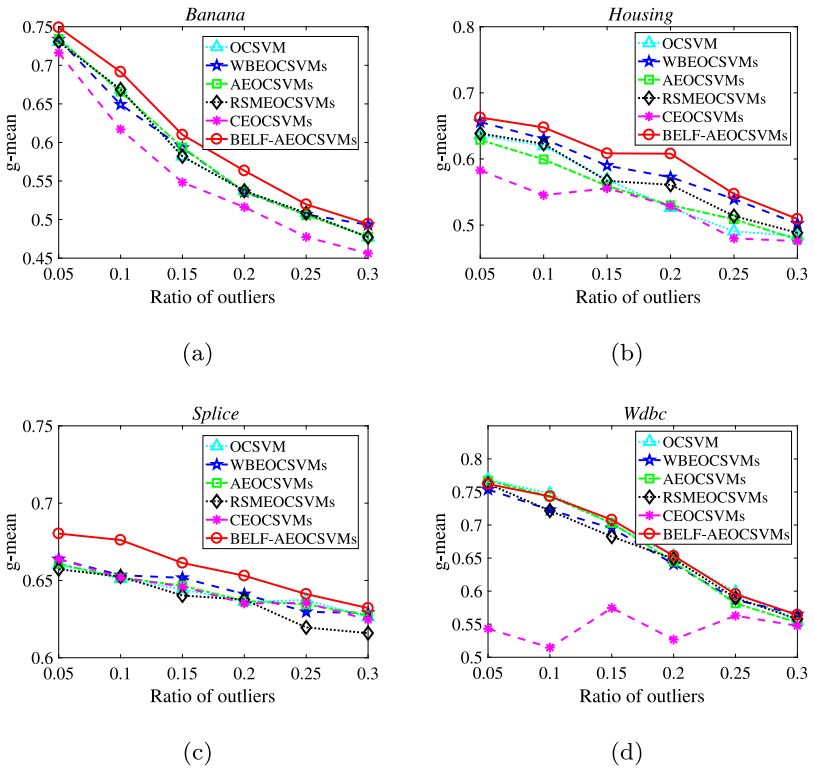


Fig. 4. The outcomes of the six methods for the different ratio of outliers occupying all the training samples on the four benchmark data sets. (a) *Banana*. (b) *Housing*. (c) *Splice*. (d) *Wdbc*.

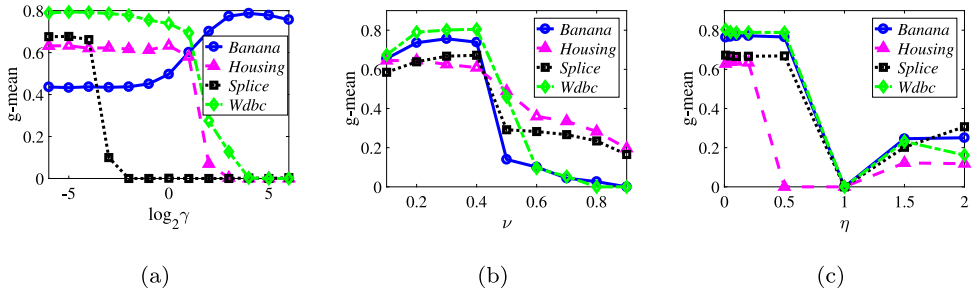


Fig. 5. Performances of BELF-AEOCSVMs with different values of γ, ν, and η on the four data sets. (a) The effect of different values of γ, (b) The impact of different values of ν, (c) The influence of different values of η.

of ν when γ and η remain unchanged. Furthermore, Fig. 5(c) illustrates the impact of η when γ and ν keep unchanged.

It is shown in Fig. 5 that when the values of the three parameters get close to their corresponding values in Table 2, the average g-mean values of BELF-AEOCSVMs change slightly. However, when the values of the three parameters are far away from their corresponding values in Table 2, the average g-mean values of BELF-AEOCSVMs drop quickly. Hence, the performance of BELF-AEOCSVMs definitely depends on the values of γ, ν, and η. The suitable parameter setting makes BELF-AEOCSVMs obtain higher performance, while the unsuitable parameter setting certainly make BELF-AEOCSVMs achieve comparatively poor performance.

4.3. Handwritten digit data set

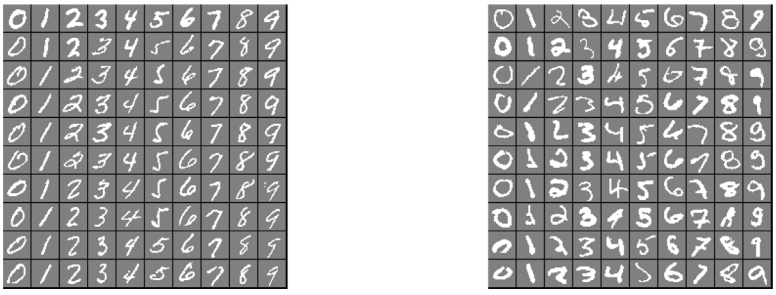
The efficiency of BELF-AEOCSVMs is further verified on MNIST [37]. The information of this handwritten data set is briefly introduced below.

MNIST: It consists of 60000 training images and 10000 testing images of the handwritten digits 0~9 in gray scale with 28 × 28 pixels. These images are blurred and sub-sampled down to 8 × 8 pixels. The images of one certain digit are used as target samples, whilst the images of the rest nine digits are used as non-target samples. 2000 samples are randomly selected from the target samples in the training images to compose the unpolluted training set, while 100 samples are randomly selected from the non-target samples in the training images to generate the outliers. The remain target samples and the first 109 non-target samples from each digit in the testing images are utilized for testing.

The parameter settings of the six methods contain $\gamma = 2^{-3}$, $\nu = 0.1$, $\sigma = 1$, and $\eta = 0.1$. The testing results of the 20 trials are summarized in Table 4. One can observe from Table 4 that BELF-AEOCSVMs is statistically different from its related approaches. Furthermore, the generalization performance of BELF-AEOCSVMs is better than the other five methods upon all the data sets except MNIST(2), MNIST(3), and MNIST(5). Hence, BELF-AEOCSVMs has better anti-outlier performance

Table 4
The testing results of the six different methods on the handwritten digit data set. The best outcomes are emphasized in bold.

Data sets	OCSVM G-mean P_T, P_W	WBEOCSVMs G-mean P_T, P_W	RSMEOCSVMs G-mean P_T, P_W	CEOCSVMs G-mean P_T, P_W	AEOCSVMs G-mean P_T, P_W	BELF-AEOCSVMs G-mean
MNIST(0)	0.7269 ± 0.0003 1.33E−041,1.15E−009	0.8182 ± 0.0004 1.79E−024,1.15E−009	0.7362 ± 0.0010 1.43E−046,1.15E−009	0.6343 ± 0.0020 1.12E−037,1.15E−009	0.7444 ± 0.0021 3.46E−035,1.15E−009	**0.8331 ± 0.0006**
MNIST(1)	0.8893 ± 0.0007 3.66E−032,1.15E−009	0.8754 ± 0.0018 4.34E−027,1.15E−009	0.8755 ± 0.0002 3.47E−041,1.15–009	0.8893 ± 0.0007 3.66E−032,1.1eE-009	0.8955 ± 0.0022 2.10E−018,1.15E−009	**0.9109 ± 0.0001**
MNIST(2)	0.5713 ± 0.0017 3.42E−059,1.15E−009	**0.7131 ± 0.0063** 2.10E−017,1.15E−009	0.5556 ± 0.0026 1.85E−041,1.15E−009	0.5713 ± 0.0017 3.42E−059,1.15E−009	0.5893 ± 0.0016 9.28E−055,1.15E−009	0.6595 ± 0.0017
MNIST(3)	0.6377 ± 0.0037 1.15E−052,1.15E−009	**0.7234 ± 0.0004** 2.32E−010,1.15E−009	0.6258 ± 0.0072 9.60E−029,1.15E−009	0.6412 ± 0.0120 1.96E−014,1.15E−009	0.6448 ± 0.0039 1.17E−058,1.15E−009	0.7141 ± 0.0039
MNIST(4)	0.7474 ± 0.0045 4.45E−025,1.15E−009	0.7933 ± 0.0008 3.76E−039,1.15E−009	0.7655 ± 0.0010 6.99E−031,1.15E−009	0.7474 ± 0.0045 4.45E−025,1.15E−009	0.7690 ± 0.0013 3.08E−065,1.15E−009	**0.8471 ± 0.0013**
MNIST(5)	0.5908 ± 0.0053 1.00E−028,1.15E−009	**0.6757 ± 0.0015** 1.33E−023,1.15E−009	0.5701 ± 0.0056 4.01E−030,1.15E−009	0.5908 ± 0.0053 1.00E−028,1.15E−009	0.5979 ± 0.0064 6.44E−025,1.15E−009	0.6566 ± 0.0028
MNIST(6)	0.7162 ± 0.0031 2.04E−039,1.15E−009	0.7608 ± 0.0004 1.69E−027,1.15E−009	0.7163 ± 0.0027 3.16E−043,1.15E−009	0.7162 ± 0.0031 2.04E−039,1.15E−009	0.7272 ± 0.0019 3.78E−053,1.15E−009	**0.8174 ± 0.0021**
MNIST(7)	0.7065 ± 0.0022 1.68E−034,1.15E−009	0.7773 ± 0.0015 5.59E−022,1.15E−009	0.6868 ± 0.0008 4.86E−061,1.15E−009	0.7087 ± 0.0119 1.00E−017,1.15E−009	0.7177 ± 0.0015 8.93E−044,1.15E−009	**0.7865 ± 0.0007**
MNIST(8)	0.6851 ± 0.0026 2.14E−034,1.15E−009	0.7624 ± 0.0029 5.16E−012,1.15E−009	0.6809 ± 0.0054 6.79E−027,1.15E−009	0.6851 ± 0.0026 2.14E−034,1.15E−009	0.7032 ± 0.0013 1.06E−045,1.15E−009	**0.7751 ± 0.0009**
MNIST(9)	0.7464 ± 0.0023 1.38E−034,1.15E−009	0.8071 ± 0.0030 1.73E−021,1.15E−009	0.7475 ± 0.0012 2.29E−035,1.15E−009	0.7464 ± 0.0023 1.38E−034,1.15E−009	0.7623 ± 0.0007 4.96E−037,1.15E−009	**0.8446 ± 0.0004**



(a) (b)

Fig. 6. The 100 best and 100 worst target samples in the testing images identified by BELF-AEOCSVMs. (a) The 100 best target samples recognized by BELF-AEOCSVMs. (b) The 100 worst target samples recognized by BELF-AEOCSVMs.

than its related methods on MNIST. The reasons for BELF-AEOCSVMs obtaining inferior performances on the above three data sets may be as follows.

- In comparison with the other seven digits, the digits 2, 3 and 5 are more easily effected by outliers, which can be deduced from Table 4 that the average g-mean values of almost all the six methods on the three data sets are smaller than those upon the rest seven data sets.
- The parameter settings of BELF-AEOCSVMs on all the ten data sets are fixed to avoid the tedious search process of the optimal parameter combination. Assigning different values to the parameters of BELF-AEOCSVMs on different data sets surely can enhance its generalization performance. However, the fixed parameter values are utilized for BELF-AEOCSVMs on all the data sets to save time.
- Compared to the other seven digits, the handwritten digits 2, 3 and 5 are more likely to be incorrectly recognized as the other digits.

In addition, Fig. 6 illustrates the 100 best and 100 worst target samples in the testing images identified by BELF-AEOCSVMs. One can find from Fig. 6 that BELF-AEOCSVMs can assign bigger decision function values to the regular samples and smaller values to the irregular samples.

5. Conclusion

To enhance the anti-outlier performance of the traditional AdaBoost based ensemble of OCSVMs, the bounded exponential loss function is proposed to replace the exponential loss function in AdaBoost. Several properties of the proposed bounded exponential loss function are investigated. Furthermore, the update formulae for weights of base classifiers and probability distribution of training samples within the proposed ensemble method, i.e., BELF-AEOCSVMs are designed. In addition, the empirical error upper bound of BELF-AEOCSVMs is deduced from the theoretical point of view. In comparison with OCSVM and its pertinent ensemble methods, BELF-AEOCSVMs exhibits better anti-outlier performance upon the artificial and benchmark data sets.

The anti-outlier and generalization abilities of BELF-AEOCSVMs may be further enhanced by choosing different parameter values for different OCSVMs in the ensemble. Nevertheless, choosing the suitable parameter values is time-consuming. In the further, we will attempt to devise a heuristic approach for selecting suitable parameter values for each OCSVMs in the ensemble. Moreover, the training complexity of BELF-AEOCSVMs is high on large-scale data set mainly due to the iterative update of weights for its base classifiers. We will consider other optimization methods rather than the Newton–Raphson approach. In addition, BELF-AEOCSVMs is designed for one-class classification. The

bounded exponential loss function based AdaBoost ensemble of binary or multi-class classifiers will be investigated.

Declaration of competing interest

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

Data availability

The data that has been used is confidential.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Nos. 61672205 and 62376161), the Natural Science Foundation of Hebei province (No. F2017201020) and the High-Level Talents Research Start-Up Project of Hebei University (No. 521100222002).

Appendix A. Proof of Proposition 4

One can easily obtain that

1. For $\forall u$, $\ell_{bexp}(u) < \ell_{bexp}(-u)$ holds, and,
2. $\ell'_{bexp}(u) = -\xi\eta\exp(-(\eta\exp(-u)+u))$, which leads to $\ell'_{bexp}(0) \neq 0$ exists.

Moreover,

$$E[\ell_{bexp}(yf(\mathbf{x}))|\mathbf{x}] = E\left\{\xi\left[1-e^{-\eta e^{-yf(\mathbf{x})}}\right]|\mathbf{x}\right\}$$
$$= \xi\left[1-e^{-\eta e^{-f(\mathbf{x})}}\right]p(\mathbf{x}) + \xi\left[1-e^{-\eta e^{f(\mathbf{x})}}\right][1-p(\mathbf{x})]. \tag{A.1}$$

Hence,

$$\frac{\partial E[\ell_{bexp}(yf(\mathbf{x}))|\mathbf{x}]}{\partial f(\mathbf{x})} = -\xi p(\mathbf{x})\eta e^{\left[-\eta e^{-f(\mathbf{x})}-f(\mathbf{x})\right]} + \xi[1-p(\mathbf{x})]\eta e^{\left[-\eta e^{f(\mathbf{x})}+f(\mathbf{x})\right]} \tag{A.2}$$

and

$$\frac{\partial^2 E[\ell_{bexp}(yf(\mathbf{x}))|\mathbf{x}]}{\partial f^2(\mathbf{x})} = \xi p(\mathbf{x})\eta e^{\left[-\eta e^{-f(\mathbf{x})}-f(\mathbf{x})\right]}\left[1-\eta e^{-f(\mathbf{x})}\right] + \xi[1-p(\mathbf{x})]\eta e^{\left[-\eta e^{f(\mathbf{x})}+f(\mathbf{x})\right]}\left[1-\eta e^{f(\mathbf{x})}\right]. \tag{A.3}$$

Unfortunately, setting the partial derivative in (A.2) to zero, we cannot get the explicit expression of the global extremum $f^*(\mathbf{x})$. Instead, the following relation can be obtained.

$$e^{\left[-\eta e^{-f(\mathbf{x})}-f(\mathbf{x})\right]} = \frac{1-p(\mathbf{x})}{p(\mathbf{x})}e^{\left[-\eta e^{f(\mathbf{x})}+f(\mathbf{x})\right]}. \tag{A.4}$$

Substituting (A.4) into (A.3), we get

$$\frac{\partial^2 E[\ell_{bexp}(yf(\mathbf{x}))|\mathbf{x}]}{\partial f^2(\mathbf{x})} = \xi[1-p(\mathbf{x})]\eta e^{\left[-\eta e^{f(\mathbf{x})}+f(\mathbf{x})\right]}\left[-\eta e^{-f(\mathbf{x})}-\eta e^{f(\mathbf{x})}+2\right]. \tag{A.5}$$

It is easy to examine that $\xi[1-p(\mathbf{x})]\eta e^{\left[-\eta e^{f(\mathbf{x})}+f(\mathbf{x})\right]} > 0$. Moreover, one can find that $-\eta e^{-f(\mathbf{x})}-\eta e^{f(\mathbf{x})}+2 > 0$ if the condition $e^{-f(\mathbf{x})}+e^{f(\mathbf{x})} < \frac{2}{\eta}$ holds.

Therefore, if the conditions $p(\mathbf{x}) \neq 1$ and $e^{-f(\mathbf{x})}+e^{f(\mathbf{x})} < \frac{2}{\eta}$ meet, we can obtain $\frac{\partial^2 E[\ell_{bexp}(yf(\mathbf{x}))|\mathbf{x}]}{\partial f^2(\mathbf{x})} > 0$. Let $f^*(\mathbf{x})$ be the global extremum of $E[\ell_{bexp}(yf(\mathbf{x}))|\mathbf{x}]$. Since $\frac{\partial^2 E[\ell_{bexp}(yf(\mathbf{x}))|\mathbf{x}]}{\partial f^2(\mathbf{x})} > 0$, $f^*(\mathbf{x})$ must be the global minimizer of $E[\ell_{bexp}(yf(\mathbf{x}))|\mathbf{x}]$. Thus, according to the above outcomes and Lemma 1, we know that the bounded exponential loss function $\ell_{bexp}(u)$ is Fisher consistent.

References

[1] B. Schölkopf, R.C. Williamson, A.J. Smola, J. Shawe-Taylor, J.C. Platt, Support vector method for novelty detection, Adv. Neural Inf. Process. Syst. 12 (2000) 582–588.
[2] D.M.J. Tax, R.P.W. Duin, Support vector data description, Mach. Learing 54 (1) (2004) 45–66.
[3] D.M.J. Tax, R.P.W. Duin, Combining one-class classifiers, in: Proceedings of the 2nd International Workshop on Multiple Classifier Systems, 2001, pp. 299–308.
[4] S. Seguí, L. Igual, J. Vitrià, Weighted bagging for graph based one-class classifiers, in: N. ElGayar, J. Kittler, F. Roli (Eds.), MCS2010, in: LNCS, vol. 5997, 2010, pp. 1–10.
[5] P. Casale, O. Pujol, P. Radeva, Approximate polytope ensemble for one-class classification, Pattern Recognit. 47 (2014) 854–864.
[6] B. Krawczyk, M. Woźniak, Wagging for combining weighted one-class support vector machines, Procedia Comput. Sci. 51 (2015) 1565–1573.
[7] J. Liu, Q. Miao, Y. Sun, J. Song, Y. Quan, Fast structural ensemble for one-class classification, Pattern Recognit. Lett. 80 (2016) 179–187.
[8] B. Krawczyk, M. Galar, M. Woźniak, H. Bustince, F. Herrera, Dynamic ensemble selection for multi-class classification with one-class classifiers, Pattern Recognit. 83 (2018) 34–51.
[9] M. Sabzevari, G. Martínez-Muñoz, A. Suárez, Small margin ensembles can be robust to class-label noise, Neurocomputing 160 (2015) 18–33.
[10] G.I. Webb, Multiboosting: a technique for combining boosting and wagging, Mach. Learn. 40 (2) (2000) 159–196.
[11] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. System Sci. 55 (1) (1997) 119–139.
[12] G. Rätsch, T. Onoda, K.R. Müller, Soft margins for AdaBoost, Mach. Learn. 42 (3) (2001) 287–320.
[13] T. Takenouchi, S. Eguchi, Robustifying AdaBoost by adding the naive error rate, Neural Comput. 16 (2004) 767–787.
[14] J. Cao, S. Kwong, R. Wang, A noise-detection based AdaBoost algorithm for mislabeled data, Pattern Recognit. 45 (2012) 4451–4465.
[15] B. Sun, S. Chen, J. Wang, H. Chen, A robust multi-class AdaBoost algorithm for mislabeled noisy data, Knowl.-Based Syst. 102 (2016) 87–102.
[16] Q. Miao, Y. Cao, G. Xia, M. Gong, J. Liu, J. Song, Rboost: label noise-robust boosting algorithm based on a nonconvex loss function and the numerically stable base learners, IEEE Trans. Neural Netw. Learn. Syst. 27 (11) (2016) 2216–2228.
[17] M. Sabzevari, G. Martńez-Muñoz, A. Suárez, Voteboosting ensembles, Pattern Recognit. 83 (2018) 119–133.
[18] X. Gu, P.P. Angelov, Multiclass fuzzily weighted adaptive-boosting-based self-organizing fuzzy inference ensemble systems for classification, IEEE Trans. Fuzzy Syst. 30 (9) (2022) 3722–3735.
[19] Y. Sun, S. Todorovic, J. Li, D.O. Wu, A robust linear programming based boosting algorithm, in: 2005 IEEE Workshop on Machine Learning for Signal Processing, 2005, pp. 49–54.
[20] T. Kanamori, T. Takenouchi, S. Eguchi, N. Murata, Robust loss functions for boosting, Neural Comput. 19 (2007) 2183–2244.
[21] W. Hu, J. Gao, Y. Wang, O. Wu, S. Maybank, Online Adaboost-based parameterized methods for dynamic distributed network intrusion detection, IEEE Trans. Cybern. 44 (1) (2014) 66–82.
[22] Z. Wang, Robust boosting with truncated loss functions, Electron. J. Stat. 12 (2018) 599–650.
[23] K. Wang, Y. Wang, Q. Zhao, D. Meng, X. Liao, Z. Xu, SPLBoost: an improved robust boosting algorithm based on self-paced learning, IEEE Trans. Cybern. 51 (3) (2021) 1556–1570.
[24] X.F. Chen, H.J. Xing, X.Z. Wang, A modified AdaBoost method for one-class SVM and its application to novelty detection, in: 2011 IEEE International Conference on Systems, Man, and Cybernetics, 2011, pp. 3506–3511.
[25] G. Rätsch, S. Mika, B. Schölkopf, K.R. Müller, Constructing boosting algorithms from SVMs: an application to one-class classification, IEEE Trans. Pattern Anal. Mach. Intell. 24 (9) (2002) 1184–1199.
[26] Q. Tao, G.W. Wu, J. Wang, A new maximum margin algorithm for one-class problems and its boosting implementation, Pattern Recognit. 38 (2005) 1071–1077.
[27] H.J. Xing, W.T. Liu, Robust AdaBoost based ensemble of one-class support vector machines, Inf. Fusion 55 (2020) 45–58.
[28] Y. Lin, A note on margin-based loss functions in classification, Statist. Probab. Lett. 68 (1) (2004) 73–82.
[29] P. Henrici, Applied and Computational Complex Analysis, Wiley, New York, 1974.
[30] N.M. Khan, R. Ksantini, I.S. Ahmad, L. Guan, Covariance-guided one-class support vector machine, Pattern Recognit. 47 (2014) 2165–2177.
[31] R.E. Schapire, Y. Freund, P. Bartlett, W.S. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, in: Fourteenth International Conference on Machine Learning, Vol. 26, No. 5, 1997, pp. 322–330.
[32] A.D. Shieh, D.F. Kamm, Ensembles of one class support vector machines, in: J.A. Benediktsson, J. Kittler, F. Roli (Eds.), Multiple Classifier Systems, in: Lecture Notes in Computer Science, vol. 5519, 2009, pp. 181–190.

[33] V. Cheplygina, D.M.J. Tax, Pruned random subspace method for one-class classifiers, in: The 10th International Workshop on Multiple Classifier Systems, 2011, pp. 96–105.
[34] B. Krawczyk, M. Woźniak, B. Cyganek, Clustering-based ensembles for one-class classification, Inform. Sci. 264 (2014) 182–195.
[35] M. Lichman, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2013, http://archive.ics.uci.edu/ml.
[36] W. Zhang, L. Du, L. Li, X. Zhang, H. Liu, Infinite Bayesian one-class support vector machine based on Dirichlet process mixture clustering, Pattern Recognit. 78 (2018) 56–78.
[37] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

Hong-Jie Xing received his M.Sc. degree from Hebei University, Baoding, China in 2003 and his Ph.D. degree in Pattern Recognition and Intelligent Systems from Institute of Automation, Chinese Academy of Sciences, Beijing, China in 2007. His research interests include neural networks, supervised and unsupervised learning, and support vector machines. Now, he works as a professor in Hebei University and serves as a member of IEEE.

Wei-Tao Liu received her B. Sc. Degree from Cangzhou Normal University, Cangzhou, China in 2016 and her M.Sc. degree in applied mathematics from Hebei University, Baoding, China in 2019. Her research interests include kernel method and novelty detection.

Xi-Zhao Wang received his Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1998. He is currently a Professor with the College of Computer Science and Software Engineering, Shenzhen University, Guangdong, China. From September 1998 to September 2001, he was a Research Fellow with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. He has edited over 10 special issues and published three monographs, two textbooks, and over 200 peer-reviewed research papers. His current research interests include uncertainty modeling and machine learning for big data. Dr. Wang was a recipient of the IEEE SMCS Outstanding Contribution Award in 2004 and the IEEE SMCS Best Associate Editor Award in 2006. He is the previous BoG Member of IEEE SMC Society, the Chair of IEEE SMC Technical Committee on Computational Intelligence, the Chief Editor of the International Journal of Machine Learning and Cybernetics, and an associate editors for a couple of journals in the related areas. He is the General Co-Chair of the 2002–2017 International Conferences on Machine Learning and Cybernetics, cosponsored by IEEE SMCS. He was a Distinguished Lecturer of the IEEE SMCS.