Neural Comput & Applic (2017) 28 (Suppl 1):S337–S347
DOI 10.1007/s00521-016-2343-3

ORIGINAL ARTICLE

Ellipsoidal support vector data description

Kasemsit Teeyapan[1] · **Nipon Theera-Umpon**[1,2] · **Sansanee Auephanwiriyakul**[2,3]

Received: 28 July 2015 / Accepted: 9 May 2016 / Published online: 24 May 2016

Abstract This paper presents a data domain description formed by the minimum volume covering ellipsoid around a dataset, called "ellipsoidal support vector data description (eSVDD)." The method is analogous to support vector data description (SVDD), but instead, with an ellipsoidal domain description allowing tighter space around the data. In eSVDD, a hyperellipsoid extends its ability to describe more complex data patterns by kernel methods. This is explicitly achieved by defining an "empirical feature map" to project the images of given samples to a higher-dimensional space. We compare the performance of the kernelized ellipsoid in one-class classification with SVDD using standard datasets.

Keywords Kernel minimum volume covering ellipsoid · Ellipsoidal support vector data description · Domain data description · Empirical feature space

1 Introduction

Data description is a problem of how to represent a group of data. In general, a description is constructed from a given set of target objects and is later used to predict whether incoming unknown objects belong to the same target group or not. The simplest form of the problem is known as one-class classification where only one description is concerned. Such a problem can be used for outlier or novelty detection. One-class classification can be viewed as a two-class problem where only data in one class called targets are easily obtained. However, the availability of the samples from the other class could be very rare because of either the difficulty or high cost in data collection. These samples may be considered outliers in the case of undesirable, or novelties in the case of desirable ones.

Support vector data description (SVDD), inspired by the rise of the support vector machine (SVM) [6], was proposed by Tax and Duin [24]. This method solves data description problems by fitting a spherical shape around the targets in a higher-dimensional space defined by a kernel function. Like SVM, the method has an ability to apply kernel tricks and does not depend on estimating a probability density function of the target data like in some existing literature [4].

The success of SVDD is obviously witnessed by a number of derived works; for example, Tax and Duin themselves later extended SVDD so that outliers are also included in estimating the descriptive domain. The method is called SVDD with negative samples, or nSVDD [25], which is the more complete form of SVDD, making it comparable to SVM. In other words, SVDD possesses a set of hyperspheres as a predefined hypothesis set instead of hyperplanes as in SVM. Some other extensions of SVDD include the two-norm nSVDD by Mu and Nandi [16] where they also proposed a scheme for multiclass classification. Their multiclass paradigm is to find a hypersphere containing the data in one class but excluding the others. The optimization is done for all classes so each class has its

✉ Nipon Theera-Umpon
nipon@ieee.org; nipon.t@cmu.ac.th

[1] Department of Electrical Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai 50200, Thailand

[2] Biomedical Engineering Center, Chiang Mai University, Chiang Mai 50200, Thailand

[3] Department of Computer Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai 50200, Thailand

own covering sphere. The classification rule for a new test sample is determined by the shortest proximity from the sample to the closest ball. Huang [11] also proposed the two-class SVDD (TC-SVDD) simultaneously trying to find two hyperspheres covering two classes of samples.

The idea of SVDD is built around the minimum enclosing ball problem (MEB), proposed by Sylvester [23] for more than 150 years, to find the smallest ball that contains all training samples. SVDD may be considered as a soft-margin case of MEB with kernel tricks. As a further step, it is obviously tempting to develop an ellipsoid version of the SVDD which will be called the "ellipsoidal support vector data description" (eSVDD). Intuitively, ellipsoids are considered the next simple, convex, smooth geometrical shape to a plane and a sphere. The popular and well-known Gaussian distribution function is also represented by ellipsoidal shapes.

A natural choice for selecting a covering ellipsoid is the one with minimum volume containing the target data. Such an ellipsoid is known as "minimum volume covering ellipsoid" (MVCE) which is the ellipsoid with the smallest volume containing a set of points. In a larger picture, MVCE belongs to one class of ellipsoid inclusion problems [1]. Its potential applications span over various fields; for example, it was used to approximate a stability region in control theory [14], represent a group of data in statistics, or even play a part in obstacle collision detection [18]. This is because its shape is unarguably less conservative than a sphere.

One of the earliest interests in MVCE can be traced back to the famous Löwner-John's ellipsoid [10] where the optimal ellipsoid with minimum volume containing a convex body is unique and only characterized by the contacted points between the convex body and the optimal ellipsoid. Since then, numerous studies have been devoted to particularly solving the MVCE problem such as the works from Khachiyan [12], Sun and Freund [22], Kuma and Yıldırım [13], Todd and Yıldırım [29], and Ahipaşaoğlu [2]. In addition, the problem can also be cast as a semidefinite programming problem. Toh [30] and Vandenberghe [32] proposed a generic logarithm-determinant maximization solver which also can be used to solve MVCE problems.

MVCE was also applied to pattern classification problems. The earliest work was by Rosen [19] where the size of an ellipsoid was measured by the trace of a matrix, equivalently equal to the sum of squares of the ellipsoid semiaxes. The application at the time was to perform a binary classification on vectorcardiograms with 33 samples in $\mathbb{R}^3$. Other more recent works include [8, 21, 34].

However, perhaps due to its higher degrees of nonlinearity and more computational cost than using a plane or sphere, MVCE has not been widely applied to pattern classification problems. Nevertheless, by considering the progress of SVM, it can be observed that SVM became more and more efficiently solved than when it was initially introduced. Therefore, working with ellipsoids is still interesting and it may provide fruitful results over SVDD as we will later show in this paper. In addition, MVCE problems are largely close to the D-optimal design problem which is an active area of research for decades. The research results from the field of experimental designs may help support the uses of MVCE in pattern classification problems.

In this paper, the main objective is to design a learning machine whose predefined hypothesis set consists of ellipsoids. The proposed method will be called eSVDD since it is similar to SVDD but with more degrees of flexibility allowed by the use of ellipsoids. The proposed eSVDD is not merely the MVCE, but equipped with more functionalities including soft margins, negative samples, as well as kernel methods. A soft margin is added to eSVDD to control the *capacity* of the hypothesis set. Incorporating negative samples may help better reject outliers. More complex descriptive boundary is also enabled by kernel methods.

Nevertheless, the problem of solving eSVDD in feature space cannot be done by usual kernel tricks as in SVDD since its dual formulation is expressed in terms of outer products. There are some studies by Dolia et al. [7, 8] trying to estimate MVCE in the kernel-defined feature space where they used spectral decomposition and singular value decomposition (SVD) to reformulate the problem in the form of inner products in order to apply kernel tricks. Wei et al. [33, 34] also applied the similar approach and proposed the concept of "enclosing machine learning" for data description. Unfortunately, their method in applying kernel tricks is too specific to the structure of the problem. As a result, it cannot promptly utilize various existing MVCE solvers. In order to overcome such a problem, we propose that the problem should be tackled from the perspective of "empirical feature map" [20, 35]. In other words, it is not necessary to rewrite the problem in terms of inner products. Instead, it is better to apply kernel methods by means of approximately estimating an image of samples in the feature space.

In the next section, we describe how MVCE problems are formed as eSVDD. Then, the previous work on estimating MVCE in feature space with kernel tricks is explained in Sect. 3 as well as the proposed method based on empirical feature mapping. In Sect. 4, the performance of the proposed method will be evaluated against SVDD on one-class classification with some standard benchmark datasets, followed by the conclusion in Sect. 5.

2 Minimum volume covering ellipsoid

An ellipsoid can be represented in various ways [5]. Let $\mathcal{E}(\mathbf{E}, \mathbf{d})$ denote an ellipsoid $\mathcal{E}$ whose center is at $\mathbf{d} \in \mathbb{R}^n$ with its shape described by $\mathbf{E} \in \mathbb{S}^n_{++}$. An ellipsoid is a closed convex set defined by

$$\mathcal{E}(\mathbf{E}, \mathbf{d}) = \{\mathbf{x} \in \mathbb{R}^n : (\mathbf{x} - \mathbf{d})^{\mathrm{T}} \mathbf{E} (\mathbf{x} - \mathbf{d}) \leq 1\}. \tag{1}$$

Alternatively, by the change of variables with $\mathbf{E} = \mathbf{M}^2$ and $\mathbf{d} = \mathbf{M}^{-1}\mathbf{z}$, another form of the same covering ellipsoid can be expressed as

$$\mathcal{E}(\mathbf{M}^2, \mathbf{M}^{-1}\mathbf{z}) = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{M}\mathbf{x} - \mathbf{z}\| \leq 1\} \tag{2}$$

with its volume equal to

$$\frac{(n\pi)^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)} \frac{1}{\det \mathbf{M}} \tag{3}$$

where Γ is the gamma function.

Given a set of samples $\mathbf{x}_i \in \mathbb{R}^n$ where $i \in \{1, \ldots, m\}$, the main interest is to find the minimum volume ellipsoid covering all the given points. However, in order to avoid a degenerate case where an ellipsoid has zero volume in a particular dimension, we first make the following assumption similar to [22]:

Assumption 1 The affine hull of the m given samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$ span $\mathbb{R}^n$.

2.1 MVCE centered at the origin

According to Titterington [28], estimating the MVCE covering $\{\mathbf{x}_i\}_{i=1}^m \subset \mathbb{R}^n$ is equivalent to finding the MVCE centered at the origin covering the set of augmented samples $\tilde{\mathbf{x}}_i = [\mathbf{x}_i^{\mathrm{T}}, \ 1]^{\mathrm{T}}$ for $i = 1, 2, \ldots, m$. The mapping of the MVCE from the augmented space $\mathbb{R}^{n+1}$ back to $\mathbb{R}^n$ is trivial as described in [9, 34]. Therefore, it is possible to merely concern about solving for the MVCE among the family of ellipsoids whose centers are fixed at the origin in $\mathbb{R}^{n+1}$. Let the ellipsoid described as $\tilde{\mathcal{E}}(\tilde{\mathbf{E}}, \mathbf{0})$ where $\tilde{\mathbf{E}} \in \mathbb{S}^{n+1}_{++}$ and $\mathbf{0}$ is the zero vector with the appropriate dimension. The minimum volume covering ellipsoid at the origin is the solution to the following problem:

$$\begin{aligned} \min_{\tilde{\mathbf{E}}} \quad & \log\det(\tilde{\mathbf{E}}^{-1}) \\ \text{s.t.} \quad & \tilde{\mathbf{x}}_i^{\mathrm{T}} \tilde{\mathbf{E}} \tilde{\mathbf{x}}_i \leq 1, \quad i = 1, \ldots, m. \end{aligned} \tag{P1}$$

Let $\mathbf{a} = [a_1, a_2, \ldots, a_m]^{\mathrm{T}} \in \mathbb{R}^m$ be a vector of Lagrange multipliers. The Lagrangian is formed as

$$L(\tilde{\mathbf{E}}, \mathbf{a}) = \log\det(\tilde{\mathbf{E}}^{-1}) + \sum_{i=1}^m a_i (\tilde{\mathbf{x}}_i^{\mathrm{T}} \tilde{\mathbf{E}} \tilde{\mathbf{x}}_i - 1). \tag{4}$$

Under the first-order necessary conditions for optimality, we obtain

$$\tilde{\mathbf{E}}^{-1} = \sum_{i=1}^m a_i \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^{\mathrm{T}} = \tilde{\mathbf{X}} \mathbf{A} \tilde{\mathbf{X}}^{\mathrm{T}} \tag{5}$$

where $\mathbf{A} = \mathrm{diag}(\mathbf{a})$ and $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_m]$. The Lagrangian (4) with the optimal $\tilde{\mathbf{E}}^*$ results in

$$L(\tilde{\mathbf{E}}^*, \mathbf{a}) = -\log\det(\tilde{\mathbf{E}}^*) - \mathbf{e}^{\mathrm{T}}\mathbf{a} + n \tag{6}$$

where $\mathbf{e}$ is the vector of ones with the appropriate dimension. To make the objective value of the dual problem the same as the primal one, we set $\mathbf{e}^{\mathrm{T}}\mathbf{a} = n$, leading to the dual problem:

$$\begin{aligned} \max_{\mathbf{a}} \quad & \log\det(\tilde{\mathbf{X}} \mathbf{A} \tilde{\mathbf{X}}^{\mathrm{T}}) \\ \text{s.t.} \quad & \mathbf{e}^{\mathrm{T}}\mathbf{a} = n \\ & \mathbf{a} \geq \mathbf{0}. \end{aligned} \tag{D1}$$

(D1) can be solved by standard interior point softwares with logarithm-determinant maximization support, such as SDPT3 version 4 [31]. Then, $\tilde{\mathbf{E}}$ is computed according to (5). The method of projecting the ellipsoid $\tilde{\mathcal{E}}(\tilde{\mathbf{E}}, \mathbf{0})$ to $\mathcal{E}(\mathbf{E}, \mathbf{d})$ is described in [9, 34]. In addition, for a given sample $\mathbf{x} \in \mathbb{R}^n$, it is easy to determine whether the point is covered by the ellipsoid by computing the distance from the sample to the origin in the augmented space as $d^2(\tilde{\mathbf{x}}, \mathbf{0}) = \tilde{\mathbf{x}}^{\mathrm{T}} \tilde{\mathbf{E}} \tilde{\mathbf{x}}$. If $0 \leq d \leq 1$, then $\mathbf{x}$ is inside or on the boundary of the ellipsoid. Otherwise, $\mathbf{x}$ is not covered by the ellipsoid.

2.2 MVCE with optimally selected center

MVCE with optimally selected center can be extended from the one centered at the origin. Some previous works such as [9] and [34] solved the problem in two steps by, first, solving for MVCE in the augmented space and then projecting the resulted ellipsoid back into the original space. However, the process can be more concise. In this section, we discuss the following formulation adapted from [13] whose result is similar to a combination of the two steps.

Lemma 1 (MVCE) Given the ellipsoid equation of the form (2), the minimum volume $\mathcal{E}(\mathbf{M}^2, \mathbf{M}^{-1}\mathbf{z})$ containing the given samples $\{\mathbf{x}_i\}_{i=1}^m$ is the solution to the following optimization problem in the dual form:

$$\begin{aligned} \max_{\mathbf{a}} \quad & \log\det \tilde{\mathbf{X}} \mathbf{A} \tilde{\mathbf{X}}^{\mathrm{T}} \\ \text{s.t.} \quad & \mathbf{e}^{T}\mathbf{a} = n \\ & \mathbf{a} \geq \mathbf{0}. \end{aligned} \tag{D2}$$

where $\tilde{\mathbf{X}} = [\mathbf{X}^{\mathrm{T}}, \mathbf{e}]^{\mathrm{T}}$ and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m]$. $\mathbf{a} \in \mathbb{R}^m$ is the vector of Lagrange multipliers and $\mathbf{A} = \mathrm{diag}(\mathbf{a})$. The optimal $\mathcal{E}(\mathbf{M}^2, \mathbf{M}^{-1}\mathbf{z})$ is obtained from the first-order optimality conditions:

$$\begin{aligned}\mathbf{M} &= \left(\mathbf{X}\left[\mathbf{A} - \frac{\mathbf{a}\mathbf{a}^{\mathrm{T}}}{\mathbf{e}^{\mathrm{T}}\mathbf{a}}\right]\mathbf{X}^{\mathrm{T}}\right)^{-1/2} \\ \mathbf{z} &= \frac{\mathbf{M}\mathbf{X}\mathbf{a}}{\mathbf{e}^{\mathrm{T}}\mathbf{a}}.\end{aligned} \tag{7}$$

Proof Since the MVCE problem formulated with the constraint in the form of (1) is not a convex program as a counterexample was given in [9], we instead consider the following MVCE problem:

$$\begin{aligned}\min_{\mathbf{M},\mathbf{z}} \quad & 2\log\det(\mathbf{M}^{-1}) \\ \text{s.t.} \quad & \|\mathbf{M}\mathbf{x}_i - \mathbf{z}\| \le 1, \quad i = 1,\ldots,m.\end{aligned} \tag{P2}$$

It should be noted that the multiples of two in the objective of (P2) are maintained merely for the sake of mathematical convenience in order to later avoid multiplication and division by two in the dual formulation. It follows that the Lagrangian of (P2) is

$$\begin{aligned}L(\mathbf{M},\mathbf{z},\mathbf{a}) &= 2\log\det(\mathbf{M}^{-1}) \\ &\quad + \sum_{i=1}^{m} a_i((\mathbf{M}\mathbf{x}_i - \mathbf{z})^{\mathrm{T}}(\mathbf{M}\mathbf{x}_i - \mathbf{z}) - 1).\end{aligned}$$

Under the first-order necessary conditions for optimality, we have

$$0 = \sum_{i=1}^{m} a_i(\mathbf{z} - \mathbf{M}\mathbf{x}_i)$$

$$0 = -2\mathbf{M}^{-1} + \sum_{i=1}^{m} a_i[(\mathbf{M}\mathbf{x}_i - \mathbf{z})\mathbf{x}_i^{\mathrm{T}} + \mathbf{x}_i(\mathbf{M}\mathbf{x}_i - \mathbf{z})^{\mathrm{T}}]$$

which lead to

$$\mathbf{z} = \frac{\mathbf{M}\mathbf{X}\mathbf{a}}{\mathbf{e}^{\mathrm{T}}\mathbf{a}} \tag{8}$$

$$\mathbf{M}^{-1} = \frac{1}{2}(\mathbf{S}\mathbf{M} + \mathbf{M}\mathbf{S}) \tag{9}$$

where

$$\mathbf{S} = \mathbf{X}\left[\mathbf{A} - \frac{\mathbf{a}\mathbf{a}^{\mathrm{T}}}{\mathbf{e}^{\mathrm{T}}\mathbf{a}}\right]\mathbf{X}^{\mathrm{T}}. \tag{10}$$

According to the proof by Sun and Freund [22], and Zhang and Gao [36], (9) has the unique positive definite solution as

$$\mathbf{M} = \mathbf{S}^{-1/2}. \tag{11}$$

As a result, the dual problem associated with (P2) can be derived as

$$\begin{aligned}\max_{\mathbf{a}} \quad & \log\det\left(\mathbf{X}\left[\mathbf{A} - \frac{\mathbf{a}\mathbf{a}^{\mathrm{T}}}{\mathbf{e}^{\mathrm{T}}\mathbf{a}}\right]\mathbf{X}^{\mathrm{T}}\right) - \mathbf{e}^{\mathrm{T}}\mathbf{a} + n \\ \text{s.t.} \quad & \mathbf{a} \ge \mathbf{0}.\end{aligned} \tag{12}$$

Similar to (6), we set $\mathbf{e}^{\mathrm{T}}\mathbf{a} = n$ so that the primal and dual problems have the same objective. In addition, the quadratic term as a function of $\mathbf{a}$ in (12) can be rewritten in a linear form using Schur complement with the fact that $\mathbf{a} = \mathbf{A}\mathbf{e}$. That is

$$\mathbf{X}\left[\mathbf{A} - \frac{\mathbf{a}\mathbf{a}^{\mathrm{T}}}{\mathbf{e}^{\mathrm{T}}\mathbf{a}}\right]\mathbf{X}^{\mathrm{T}} = \begin{bmatrix}\mathbf{X}\mathbf{A}\mathbf{X}^{\mathrm{T}} & \mathbf{X}\mathbf{A}\mathbf{e} \\ \mathbf{e}^{\mathrm{T}}\mathbf{A}\mathbf{X}^{\mathrm{T}} & \mathbf{e}^{\mathrm{T}}\mathbf{A}\mathbf{e}\end{bmatrix} = \tilde{\mathbf{X}}\mathbf{A}\tilde{\mathbf{X}}^{\mathrm{T}}.$$

Theresult follows. □

From Lemma 1, it is clearly seen that solving MVCE with optimally selected center in $\mathbb{R}^n$ (D2) is equivalent to solving MVCE centered at the origin (D1) in the augmented space $\mathbb{R}^{n+1}$ with the augmented samples. The lemma also provides a direct interpretation of the solution as an ellipsoid in $\mathbb{R}^n$, eliminating the unnecessary step required to project the ellipsoid from the augmented space to the original one. It is also worth noting from (7) that if all the elements of $\mathbf{a}$ are set to $1/m$, then $\mathbf{M}^{-2}$ and $\mathbf{M}^{-1}\mathbf{z}$ become the sample covariance and sample mean of the training samples. Lastly, the distance from the center of the ellipsoid to a given sample $\mathbf{x}$ is defined by $d(\mathbf{x},\mathbf{z}) = \|\mathbf{M}\mathbf{x} - \mathbf{z}\|$. The sample is covered by the ellipsoid when $0 \le d \le 1$, otherwise it is considered an outlier.

2.3 Ellipsoidal support vector data description

The idea of creating a soft-margin MVCE is inspired by other popular soft-margin learning machines such as SVM and SVDD. We found that Dolia et al. [7] and Wei et al. [34] also incorporated this idea in constructing their MVCEs. An MVCE problem with ℓ_1-relaxation is formulated by introducing a slack variable ξ_i, for $i = 1,\ldots,m$, to allow possible misclassification as

$$\begin{aligned}\min_{\mathbf{M},\mathbf{z},\boldsymbol{\xi}} \quad & 2\log\det(\mathbf{M}^{-1}) + \mathbf{c}^{\mathrm{T}}\boldsymbol{\xi} \\ \text{s.t.} \quad & \|\mathbf{M}\mathbf{x}_i - \mathbf{z}\|^2 \le 1 + \xi_i \\ & \xi_i \ge 0, \quad i = 1,\ldots,m\end{aligned} \tag{P3}$$

where $\boldsymbol{\xi}$ and $\mathbf{c}$ are the vectors of ξ_i and c_i, respectively, where $i = \{1,2,\ldots,m\}$ and $c_i > 0$ is given.

The introduced slack variables simply represent empirical errors which are subjected to minimization. Since the formulation follows the same concept as SVDD, it is called "ellipsoidal support vector data description" (eSVDD).

It is also possible to incorporate known outliers, if they exist, into the formulation. Such a problem is still called a one-class classification even though a training sample is labeled as either a target or an outlier. The knowledge of the presence of outliers adds a possibility to enhance data descriptive boundaries. It is natural to exclude outliers from being inside the ellipsoid. The MVCE formulation to explicitly exclude known outliers was briefly presented

in [34] for two-class classification. However, our following formulation is slightly different.

Let each sample be labeled as either $y_i = 1$ or -1 for targets and outliers, respectively, for all training samples. The MVCE formulated with both soft margins and outliers is called "eSVDD with negative samples" (neSVDD), analogous to SVDD with negative samples (nSVDD). When there is no outlier, neSVDD becomes an eSVDD problem. The neSVDD problem is stated as

$$\begin{aligned} \min_{\mathbf{M},\mathbf{z},\boldsymbol{\xi}} \quad & 2\log\det(\mathbf{M}^{-1}) + \mathbf{c}^{\mathrm{T}}\boldsymbol{\xi} \\ \text{s.t.} \quad & y_i\|\mathbf{M}\mathbf{x}_i - \mathbf{z}\|^2 \le y_i + \xi_i \\ & \xi_i \ge 0, \quad i = 1,\ldots,m \end{aligned} \tag{P4}$$

where the corresponding dual problem is

$$\begin{aligned} \max_{\mathbf{a}} \quad & \log\det \tilde{\mathbf{X}}\mathbf{A}\mathbf{Y}\tilde{\mathbf{X}}^{\mathrm{T}} \\ \text{s.t.} \quad & \mathbf{y}^{\mathrm{T}}\mathbf{a} = n \\ & \mathbf{0} \le \mathbf{a} \le \mathbf{c} \end{aligned} \tag{D4}$$

where $\mathbf{y} = [y_1, y_2, \ldots, y_m]$ and $\mathbf{Y} = \mathrm{diag}(\mathbf{y})$ with $y_i \in \{1, -1\}$ for $i = 1, 2, \ldots, m$. The optimal $\mathcal{E}(\mathbf{M}^2, \mathbf{M}^{-1}\mathbf{z})$ is computed from

$$\begin{aligned} \mathbf{M} &= \left(\mathbf{X}\mathbf{Y}\left[\mathbf{A}\mathbf{Y} - \frac{\mathbf{a}\mathbf{a}^{\mathrm{T}}}{\mathbf{y}^{\mathrm{T}}\mathbf{a}}\right]\mathbf{Y}\mathbf{X}^{\mathrm{T}}\right)^{-1/2} \\ \mathbf{z} &= \frac{\mathbf{M}\mathbf{X}\mathbf{Y}\mathbf{a}}{\mathbf{y}^{\mathrm{T}}\mathbf{a}}. \end{aligned} \tag{13}$$

Adding slack variables in the primal problem results in the box constraints on Lagrange multipliers in the dual form. Recall that the center ($\mathbf{d}$) of an ellipsoid is a linear combination of training samples weighted by the corresponding Lagrange multipliers. The ellipsoid's shape ($\mathbf{E}$) is also a linear combination of outer products of training samples weighted by the same Lagrange multipliers. When a sample possesses $a_i = 0$, its presence does not affect the ellipsoid. The samples whose $a_i > 0$ are the so-called support vectors. The shape and center of the optimal ellipsoid depend only on a linear combination of support vectors.

Suppose $\mathbf{c} = C\mathbf{e}$. An effect of regularization parameter $\mathbf{c}$ on eSVDD and neSVDD is shown in Figs. 1 and 2, respectively. Since the sum of all α_i must be n, the possible minimum value of C is $\frac{n}{m}$ for the case of eSVDD. It can be seen that when C increases, the size of the ellipsoid also increases. In the case of eSVDD, the size of the ellipsoid is limited by the maximum volume containing the samples as in Fig. 1. Increasing C beyond a certain value does not affect the shape of the ellipsoid. This differs from the case with negative samples shown in Fig. 2. The reason is that, for neSVDD, the constraint $\mathbf{y}^{\mathrm{T}}\mathbf{a} = n$ is still satisfied even for a very large C due to the subtraction between Lagrange multipliers. Therefore, a Lagrange multiplier can be as large as the value of C. However, in eSVDD, when C is set too large, it is impossible for a Lagrange multiplier to be too large because of the constraint $\mathbf{e}^{\mathrm{T}}\mathbf{a} = n$.

3 Flexible ellipsoidal support vector data description

Although ellipsoidal data descriptions are more flexible than spherical ones, they are still insufficient to describe complex data patterns. A general approach in kernel learning machines introduces kernel methods to enable more complicated descriptive boundary by replacing an inner product $\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$ with a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ for all samples i and j. Intuitively, an inner product measures a similarity between two samples. Replacing it with a kernel function provides an alternative similarity measurement. With an appropriate choice of kernel functions, the samples are mapped into space with better class separability [35].

In general, a kernel is a positive definite function $k(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$ satisfying Mercer's conditions. These properties allow the ability to explicitly factorize a kernel in the

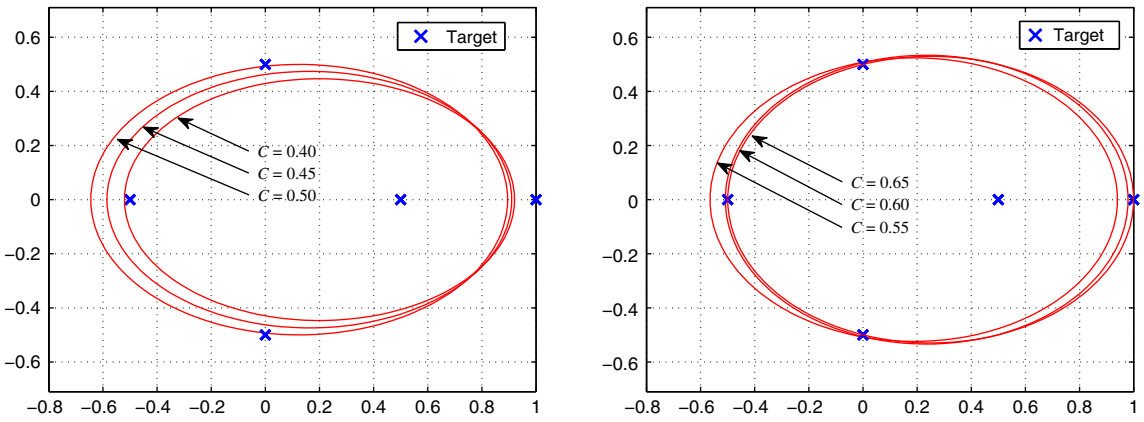

Fig. 1 Examples of eSVDD by varying the parameter $c_i = C$ for all samples

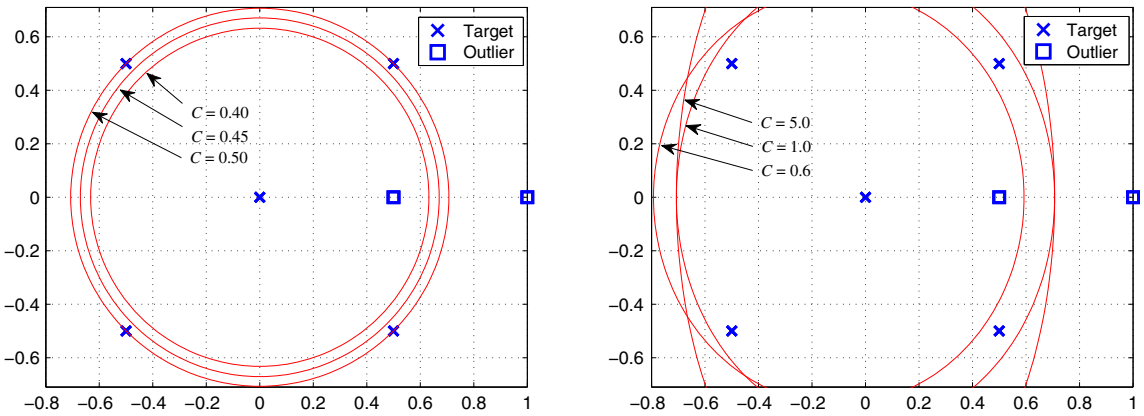

Fig. 2 Examples of neSVDD by varying the parameter $c_i = C$ for all samples

form of an inner product between two vectors, i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^{\mathrm{T}}\Phi(\mathbf{x}_j)$, where $\Phi(\mathbf{x}_i) : \mathbb{R}^n \mapsto \mathbb{H}$, and $\mathbb{H}$ is the feature space. In other words, a kernel also defines an inner product in $\mathbb{H}$. In general, the mapping $\Phi(\mathbf{x})$ is unknown and it usually maps samples into a higher-dimensional space, possibly the infinite one.

Since it is more concise to represent variables in a matrix form, let $\mathbf{K} \in \mathbb{S}_+^m$ denote a positive semidefinite matrix, the (i, j)th element of which has the value $k(\mathbf{x}_i, \mathbf{x}_j)$. By denoting $\Phi = [\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2), \ldots, \Phi(\mathbf{x}_m)]$, we also have $\mathbf{K} = \Phi^{\mathrm{T}}\Phi$.

MVCE constructions in a higher-dimensional space using kernel methods are presented in the following two subsections. We first review the existing method [7, 8, 33, 34] which tries to explicitly rewrite the MVCE formulation in the form of inner products. Then, the proposed method based on "empirical feature maps" [20] is presented. The ellipsoid formulation (D2) will be extensively used throughout the section with $\mathbf{X}$ replaced by Φ. Note that $\tilde{\Phi}$ and $\tilde{\Phi}$ represent the augmented versions of Φ and Φ, respectively.

3.1 Existing works

There exist few research studies concerning ellipsoid formulations with kernel tricks. To the best of our knowledge, such works include Dolia et al. [7, 8] and Wei et al. [33, 34]. However, their methods are basically the same and can be summarized as follows:

First, rewrite the objective (D1) by utilizing the fact that, for any matrices $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{AB}$ and $\mathbf{BA}$ have the same nonzero eigenvalues. Together with Cholesky decomposition of the kernel matrix in the augmented space, $\tilde{\mathbf{K}} = \mathbf{K} + \mathbf{e}\mathbf{e}^{\mathrm{T}} = \mathbf{C}^{\mathrm{T}}\mathbf{C}$, ones arrive at the conclusion, $\log\det(\tilde{\Phi}\mathbf{A}\tilde{\Phi}^{\mathrm{T}}) = \log\det(\mathbf{CAC}^{\mathrm{T}})$.

Second, find the definition of the distance in the augmented feature space, $\tilde{\Phi}^{\mathrm{T}}(\mathbf{x})\tilde{\mathbf{E}}\tilde{\Phi}(\mathbf{x})$ for a given sample $\mathbf{x} \in \mathbb{R}^n$. This is accomplished by performing a truncated eigenvalue decomposition on the matrix $\mathbf{A}\tilde{\mathbf{K}}\mathbf{A} = \mathbf{V}\Sigma^{\mathrm{T}}\Sigma\mathbf{V}^{\mathrm{T}}$, where $\mathbf{V}$ and Σ are the matrices of corresponding eigenvectors and singular values, respectively. The reason that the matrix $\mathbf{A}\tilde{\mathbf{K}}\mathbf{A}$ is important here arises from the reduced singular value decomposition (SVD) of $\Phi\mathbf{A}^{\frac{1}{2}} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$ which provides both $\tilde{\mathbf{E}}^{-1} = \tilde{\Phi}\mathbf{A}\tilde{\Phi}^{\mathrm{T}} = \mathbf{U}\Sigma\Sigma^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}$ and $\mathbf{AKA} = \mathbf{V}\Sigma^{\mathrm{T}}\Sigma\mathbf{V}^{\mathrm{T}}$. Obtaining $\mathbf{V}$ and Σ helps estimate $\mathbf{U} = \Phi\mathbf{A}^{\frac{1}{2}}(\Sigma\mathbf{V}^{\mathrm{T}})^{+}$ where $(\cdot)^{+}$ is a pseudoinverse operator. As a result, the Mahalanobis distance can be roughly computed by $\tilde{\Phi}^{\mathrm{T}}\tilde{\mathbf{E}}\tilde{\Phi} = \Phi^{\mathrm{T}}\mathbf{U}\left(\Sigma\Sigma^{\mathrm{T}}\right)^{-1}\mathbf{U}^{\mathrm{T}}\Phi$, where the term $\tilde{\Phi}^{\mathrm{T}}\tilde{\Phi}$ is the vector of inner products defined by a kernel.

3.2 Proposed method

Although the aforementioned approach could attain the objective to construct MVCE in the feature space, its formulation is rather specific to the factorization of the constituent of the Lagrange multipliers and kernel matrices, making it depend on the structure of the problem. In other words, the factorization of the matrix $\mathbf{A}\tilde{\mathbf{K}}\mathbf{A}$ is required. Therefore, it will not always readily be usable if the formulation of the MVCE problem is altered. In addition, it is inconvenient to apply the approach to kernelize existing MVCE algorithms such as the DRN algorithm [22] and the WA-TY algorithm [29]. From our perspective, the versatility issue must be addressed. An MVCE algorithm should be able to apply kernel methods without a burdensome step in trying to refactorize the problem in the form of inner products.

As a result, we propose a method, called "empirical feature mapping," which explicitly defines a map from the input space to an empirical feature space $\mathbb{H}_E$ which is a

finite-dimensional Euclidean space. The term "empirical" is added to indicate that the mapping is constructed from a given set of empirical measures and also to distinguish it from the feature space $\mathbb{H}$. In general, $\mathbb{H}$ and $\mathbb{H}_E$ are two different spaces, and $\mathbb{H}$ possibly even has infinite dimensions as in the case of RBF kernels. Therefore, in order to create an empirical feature mapping to a finite space in a meaningful way, for a given set of training samples, we define the mapping in Definition 1 such that the inner product in $\mathbb{H}_E$ is equal to the one in $\mathbb{H}$. This specific version of empirical feature map is called "kernel PCA map" [20] since it includes kernel whitening.

Definition 1 (Kernel PCA map) Given the training samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m \in \mathbb{R}^n$, the kernel PCA map from $\mathbb{R}^n$ to $\mathbb{H}_E$ is defined as

$$\Phi_E : \mathbf{x} \mapsto (\Omega^+)^{\mathrm{T}} \mathbf{k}(\mathbf{x}) \tag{14}$$

where $\mathbf{k}(\mathbf{x}) = [k(\mathbf{x}, \mathbf{x}_1), k(\mathbf{x}, \mathbf{x}_2), \ldots, k(\mathbf{x}, \mathbf{x}_m)]^{\mathrm{T}}$ and $\mathbf{K} = \Omega^{\mathrm{T}}\Omega$ is a matrix factorization of the kernel matrix.

Corollary 1 The empirical feature space $\mathbb{H}_E$ and the feature space $\mathbb{H}$ possess the same inner product and Euclidean distance.

Proof Let $\mathbf{k}_i = \mathbf{k}(\mathbf{x}_i)$, $\Phi_i = \Phi(\mathbf{x}_i)$, and $\Phi_{E_i} = \Phi_E(\mathbf{x}_i)$. It follows that $\Phi_{E_i}^{\mathrm{T}} \Phi_{E_j} = \mathbf{k}_i^{\mathrm{T}} \Omega^+ (\Omega^+)^{\mathrm{T}} \mathbf{k}_j = \mathbf{k}_i^{\mathrm{T}} \mathbf{K}^+ \mathbf{k}_j = \Phi_i^{\mathrm{T}} \Phi (\Phi^{\mathrm{T}}\Phi)^+ \Phi^{\mathrm{T}} \Phi_j = \Phi_i^{\mathrm{T}} \Phi_j = k(\mathbf{x}_i, \mathbf{x}_j)$. The Euclidean distance in both space is also the same according to $\|\Phi_i - \Phi_j\|^2 = k(\mathbf{x}_i, \mathbf{x}_i) - 2k(\mathbf{x}_i, \mathbf{x}_j) + k(\mathbf{x}_j, \mathbf{x}_j)$. □

Although $\mathbb{H}$ can have infinite dimensions, kernel learning machines perform only in a subspace of $\mathbb{H}$ spanned by the images of given samples $\{\Phi(\mathbf{x}_i)\}$ for $i = 1, 2, \ldots, m$. Since the inner product and the Euclidean distance in $\mathbb{H}_E$ are the same as in $\mathbb{H}$ as shown in Corollary 1, data separability is also the same in both spaces. In fact, $\mathbb{H}$ is isomorphic with $\mathbb{H}_E$ [35]. Therefore, it is tempting to work with $\mathbb{H}_E$, instead of $\mathbb{H}$, since the former is easier to access by the mapping defined in Definition 1.

Even though a kernel matrix satisfying Mercer's condition can always be factorized as $\mathbf{K} = \Omega^{\mathrm{T}}\Omega$, the decomposition may not be unique; for example, the factorization can be obtained from eigendecomposition or LDL decomposition. For eigendecomposition, we have $\Omega = (\mathbf{V}\Lambda^{\frac{1}{2}})^{\mathrm{T}}$ from $\mathbf{K} = \mathbf{V}\Lambda\mathbf{V}^{\mathrm{T}}$, where Λ is the diagonal matrix of eigenvalues corresponding to the eigenvector matrix $\mathbf{V}$. Alternatively, for LDL decomposition $\mathbf{K} = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}}$, we have $\Omega = (\mathbf{L}\mathbf{D}^{\frac{1}{2}})^{\mathrm{T}}$ where $\mathbf{L}$ is the lower triangular matrix whose diagonal elements are all ones and $\mathbf{D}$ is a diagonal matrix. Both Λ and $\mathbf{D}$ generally are not full rank since $\mathbf{K}$ is positive semidefinite. Therefore, in this paper, it is assumed to work with the reduced version of eigendecomposition and LDL decomposition.

Despite being discussed in [20, 35] as an approach to apply kernel methods to an algorithm that cannot explicitly formulate the problem in the form of inner products, empirical feature mapping is still new to MVCE problems. It will allow the problems to seamlessly utilize kernel methods. Furthermore, one benefit of using empirical feature maps is that a sample in the original space can be visualized in the feature space as one sample. The matrix $\mathbf{E}$ and the vector $\mathbf{d}$ describing the shape of an ellipsoid can also be computed, although they may not be unique depending on how $\mathbf{K}$ is factorized. As a result, empirical feature mapping opens a possibility to perform kernel optimizations to determine the most suitable kernel for a given dataset [35].

In order to show how a kernelized eSVDD looks like in data description, Fig. 3 demonstrates the effect of RBF kernels on neSVDD where 5 samples belong to the target class and one sample belongs to the outlier class. By varying only kernel parameter σ, it can be seen that the data descriptive contours fit tighter to the samples when σ

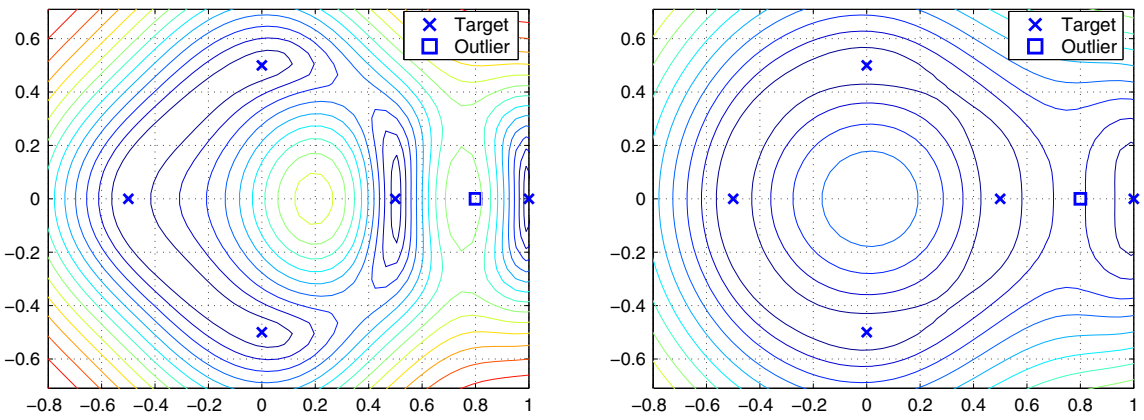

Fig. 3 Contour plots of neSVDD with the RBF kernel $k(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma}}$ varying $\sigma = 0.50$ (*left*) and 0.75 (*right*)

decreases. This is in contrast to the number of support vectors (not shown in the figure).

4 Experimental results

In this section, we are finally ready to perform some experiments on one-class classification using eSVDD with some standard datasets to demonstrate its performance. The experiments compare the minimum ellipsoids, eSVDD and neSVDD, with their minimum spherical counterparts, i.e., SVDD and nSVDD. We deployed MATLAB with YALMIP [15] as an interface to the SDPT3 solver [31] to solve logarithm-determinant minimization. The DDtools toolbox [27] which was particularly designed for one-class classification was also used to help the implementation. Since the goal is to compare eSVDD directly with SVDD, the datasets we used for the experiments were also obtained from Tax [26], the author of DDtools, who provides standard datasets in PRTools format [17] with some minor data cleanup such as filling missing values to the datasets. The list of 27 datasets used in the experiments is shown in Table 1 where m_t, m_o, and n denote the number of target samples, outliers, and features, respectively. These datasets are derived from multiclass datasets by assigning the interested group of data as the target class and the rest as outliers.

In the experiments, the domains of the training data were scaled to one. Only target classes were used for training SVDD and eSVDD. Since the RBF kernel is a popular choice among kernel functions and as it is also encouraged to use with SVDD [25], we compare these ellipsoidal and spherical data descriptive algorithms through the uses of the RBF kernel. The two hyperparameters in SVDD, nSVDD, eSVDD, and neSVDD, namely $\mathbf{c}$ and σ, were selected by grid search over 120 pairs of parameters. The predefined values of σ were 0.5, 1, 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50. The box-constraint parameter $\mathbf{c}$ was assumed to be the same for all training samples, i.e., $\mathbf{c} = C\mathbf{e}$. In the case of SVDD, the search values of C were $\frac{1}{mr}$ for $r \in [0.1, 0.2, \ldots, 1.0]$.

For eSVDD, C was set to $\frac{N}{mr}$ for $r \in [0.1, 0.2, \ldots, 1.0]$ where N is the approximate dimension of the empirical feature space. Here, kernel matrices were factorized by LDL decomposition to obtain $\Omega = (\mathbf{L}\mathbf{D}^{\frac{1}{2}})^{\mathrm{T}}$ in order to obtain the kernel PCA map. However, since $\mathbf{D}$ does not always has full rank, its diagonal whose elements are less than 10^{-5} were truncated. The dimension of the reduced $\mathbf{D}$ is simply the empirical feature space's dimension N. In addition, in order to avoid a degenerate case where Assumption 1 does not hold, a weighted identity matrix $\gamma\mathbf{I}$ with the appropriate dimension, for a small $\gamma = 10^{-5}$, was added inside the log det function.

Table 1 Datasets

Dataset (target)	m_t	m_o	n
Iris (setosa)	50	100	4
Iris (versicolor)	50	100	4
Iris (virginica)	50	100	4
Sonar (mines)	111	97	60
Sonar (rocks)	97	111	60
Imports (low risk)	71	88	25
Hepatitis (live)	123	32	19
Ecoli (periplasm)	52	284	7
Cancer wpbc (non-ret)	151	47	33
Cancer wpbc (ret)	47	151	33
Spectf (0)	95	254	44
Spectf (1)	254	95	44
Balance-scale (left)	288	337	4
Balance-scale (middle)	49	576	4
Balance-scale (right)	288	337	4
Glass (building float)	70	144	9
Glass (building nonfloat)	76	138	9
Glass (vehicle float)	17	197	9
Glass (containers)	13	201	9
Glass (headlamps)	29	185	9
Liver (1)	145	200	6
Liver (2)	200	145	6
Thyroid (normal)	93	3679	21
Wine (1)	59	119	13
Wine (2)	71	107	13
Wine (3)	48	130	13
Housing (MEDV > 35)	48	458	13

For each pair of $(\mathbf{c}, \sigma)$, tenfold cross-validation was performed. In order to avoid any bias on random reshuffles of the data, all of the four ellipsoidal and spherical SVDDs did all training and testing on the same partitioned data. However, for 9 out of 10 folds used for training nSVDD and neSVDD, negative samples were limited to only 20 samples, instead of $\frac{9m_o}{10}$. This was done to reduce the training time. The best pair was the one that yielded the maximum mean of the area under the receiver operating characteristic (ROC) curve. The ROC curve used here is defined as the plot between false-positive rates and true-positive rates with the maximum area under the curve equal to one. The data descriptive boundaries were set up to output membership probabilities so that the ROC curve could be constructed. For one sample, the membership value was defined as $e^{-\|d\|}$, where d is the distance to the center of the descriptive domain. After the best parameters were selected, the areas under the ROC curves were reported from tenfold cross-validation for 5 independent runs. The results are summarized in Table 2 where, for

Table 2 Comparison between the ellipsoidal and spherical SVDDs by the mean and standard deviation of the area under the ROC curve

	Dataset (target)	SVDD	eSVDD	nSVDD	neSVDD
1	Iris (setosa)	**1.0000** ± 0.0000	**1.0000** ± 0.0000	**1.0000** ± 0.0000	**1.0000** ± 0.0000
2	Iris (versicolor)	0.9708 ± 0.0353	**0.9920** ± 0.0194	0.9796 ± 0.0322	**0.9888** ± 0.0233
3	Iris (virginica)	0.9688 ± 0.0456	**0.9788** ± 0.0356	0.9692 ± 0.0459	**0.9708** ± 0.0394
4	Sonar (mines)	0.7394 ± 0.0987	**0.7879** ± 0.0919	0.7429 ± 0.0992	**0.8015** ± 0.0918
5	Sonar (rocks)	**0.7179** ± 0.1109	0.6273 ± 0.1296	**0.7209** ± 0.1103	0.7063 ± 0.1281
6	Imports (low risk)	**0.8338** ± 0.0968	0.7678 ± 0.1368	0.8351 ± 0.0964	**0.8606** ± 0.0948
7	Hepatitis (live)	**0.8183** ± 0.1252	0.8182 ± 0.1369	0.8183 ± 0.1252	**0.8258** ± 0.1273
8	Ecoli (periplasm)	**0.9580** ± 0.0608	0.9414 ± 0.0592	**0.9599** ± 0.0586	0.9472 ± 0.0585
9	Cancer wpbc (non-ret)	**0.5433** ± 0.1233	0.5254 ± 0.1373	**0.5362** ± 0.1305	0.5177 ± 0.1456
10	Cancer wpbc (ret)	0.6128 ± 0.1509	**0.6449** ± 0.1291	**0.6283** ± 0.1578	0.6100 ± 0.1508
11	Spectf (0)	0.8978 ± 0.0570	**0.9435** ± 0.0585	0.9008 ± 0.0564	**0.9420** ± 0.0532
12	Spectf (1)	**0.7153** ± 0.0845	0.6453 ± 0.0744	**0.7327** ± 0.0834	0.6474 ± 0.0471
13	Balance-scale (left)	0.9665 ± 0.0204	**0.9858** ± 0.0108	0.9671 ± 0.0201	**0.9780** ± 0.0158
14	Balance-scale (middle)	0.8009 ± 0.1099	**0.9224** ± 0.0475	0.8014 ± 0.1082	**0.9884** ± 0.0371
15	Balance-scale (right)	0.9665 ± 0.0188	**0.9853** ± 0.0108	0.9676 ± 0.0184	**0.9770** ± 0.0152
16	Glass (building float)	0.8000 ± 0.0943	**0.8317** ± 0.0824	0.8008 ± 0.0893	**0.8021** ± 0.1125
17	Glass (building nonfloat)	0.6541 ± 0.1244	**0.7509** ± 0.1125	0.6841 ± 0.1281	**0.7097** ± 0.1269
18	Glass (vehicle float)	0.7116 ± 0.1230	**0.8600** ± 0.1430	0.7324 ± 0.1258	**0.8853** ± 0.1540
19	Glass (containers)	0.8269 ± 0.3286	**0.9802** ± 0.0371	0.8269 ± 0.3286	**0.9665** ± 0.0872
20	Glass (headlamps)	**0.9425** ± 0.0808	0.8925 ± 0.1220	**0.9425** ± 0.0808	0.9108 ± 0.1247
21	Liver (1)	0.5614 ± 0.0770	**0.6155** ± 0.0782	0.5670 ± 0.0927	**0.5928** ± 0.0847
22	Liver (2)	0.5485 ± 0.1068	**0.5533** ± 0.0862	0.6057 ± 0.0972	**0.6216** ± 0.0942
23	Thyroid (normal)	0.7928 ± 0.0735	**0.9454** ± 0.0450	0.8453 ± 0.0676	**0.9127** ± 0.0794
24	Wine (1)	**0.9989** ± 0.0047	0.9983 ± 0.0054	0.9989 ± 0.0047	**0.9991** ± 0.0037
25	Wine (2)	0.9011 ± 0.0703	**0.9491** ± 0.0504	0.9011 ± 0.0703	**0.9687** ± 0.0412
26	Wine (3)	0.9949 ± 0.0181	**0.9986** ± 0.0062	0.9949 ± 0.0181	**0.9986** ± 0.0062
27	Housing (MEDV > 35)	0.8523 ± 0.0936	**0.8905** ± 0.0766	0.8604 ± 0.0935	**0.8862** ± 0.0853

each dataset, the best value among all the four algorithms is underlined. In addition, the table can be split in half to compare the performance between SVDD and eSVDD, or nSVDD and neSVDD. The best value between each pair is highlighted in bold.

From the table, we observe that the change from a spherical descriptive boundary to an ellipsoidal one does help improve the results as seen in the cases of SVDD vs. eSVDD and nSVDD vs. neSVDD. In the experiment, eSVDD performed better than SVDD on 18 out of 26 datasets, and neSVDD also provided better results than nSVDD on 20 out of 26 datasets. However, since some of the results only slightly differ, we further illustrate graphically how the gaps between the results exist as shown in Fig. 4. From the figure, by neglecting absolute area differences less than 0.05 (or 5 %) due to their less significance, the uses of an ellipsoid instead of a sphere only degraded the performance in 4 and 1 datasets for the cases of SVDD vs. eSVDD ($\times$) and nSVDD vs. neSVDD ($+$), respectively.

In addition, the introduction of negative samples to SVDD and eSVDD also helped improve the results in many datasets, though we observed performances dropped in some datasets. For the spherical case, negative samples improved the area under the ROC curve for 19 datasets with only one dataset (No. 9) whose value very slightly decreased. For the ellipsoidal case, the improvement was not obvious in terms of the number of datasets. We observed that including negative samples improved the results in 12 datasets, while it also degraded the results in 13 datasets. Although it may be seen that the inclusion of negative samples in eSVDD worsens the results, those results are, in fact, average values from multiple runs of cross-validation which could also be biased if they are compared directly digit by digit. Therefore, it is also interesting to look at the results from another perspective. By neglecting the absolute changes in the area under the ROC curve which are less than 5 %, we observe from Fig. 4 that the performance of eSVDD ($\square$) did not significantly degrade as negative samples were introduced.

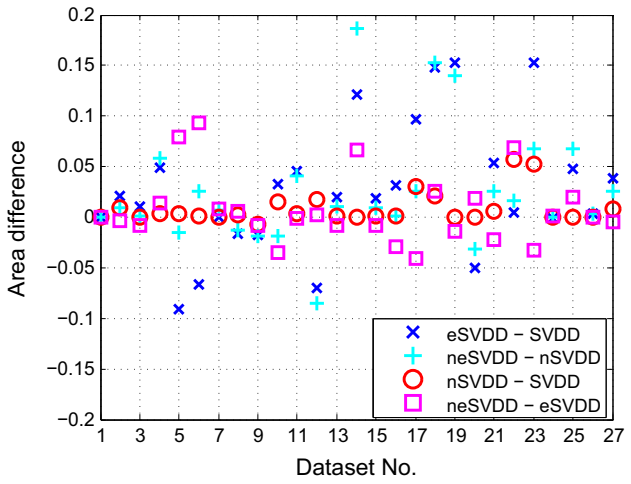

Fig. 4 Differences of the area under the ROC curve between two algorithms summarized from Table 2

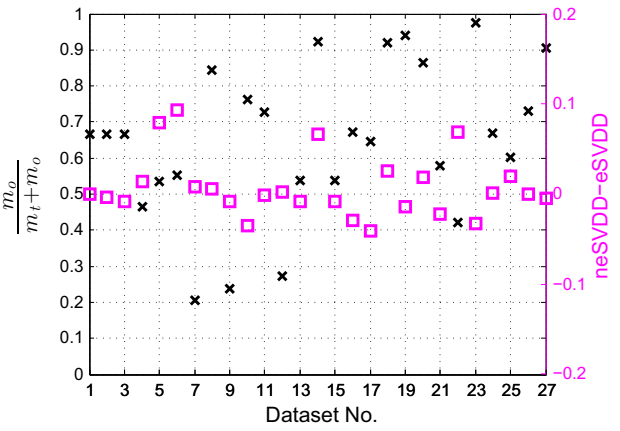

Fig. 5 Ratio of negative samples in each dataset compared against the performance changes in eSVDD after negative samples were included

It is also worth noting that the decreases in performances after negative samples were included are possibly because the presence of some negative samples may not be useful in constructing a better descriptive boundary since only 20 samples of outliers were used. In addition, in the case of neSVDD, the decreases may contribute to occasional numerical errors that occurred during the simulation. In some situations, the matrix **M** in neSVDD is indefinite so the optimization (D4) cannot be solved. In order to deal with such a problem, any pair of cross-validation parameters was discarded whenever any particular training folds could not be trained. Although this seems to be a limitation, neSVDD is still in favor in terms of the average performance.

Among all of the algorithms, the number of datasets that each algorithm performed best was 14, 10, 5, and 3 out of 27 datasets, for eSVDD, neSVDD, nSVDD, and SVDD, respectively, according to Table 2 highlighted with underlines. In overall, the spherical SVDDs performed best in 6 datasets (both the first and third columns), while the ellipsoidal SVDDs did best in 22 datasets (both the second and forth columns). Therefore, it may be concluded that using ellipsoids could provide better results than using spheres.

Furthermore, it may be interesting to see how negative samples influenced the results. We provided a plot in Fig. 5 comparing the ratio of negative samples in each dataset against how the results changed from eSVDD to neSVDD when negative samples were added. However, from the plot, no obvious relationship can be observed. This is because the number of negative samples has no direct effect on the results. In fact, what does matter is how the locations of samples provide any meaningful information to the construction of descriptive boundaries.

Despite being pointed out in [3] that the use of ℓ_1-relaxation to formulate a soft-margin ellipsoid could result in unexpected outcomes, our experimental results practically showed promising performance. We believe that this is because the claim in [3] concerned only an appropriate selection of the parameter **c** with no kernel method, which is not the case for our experiments. In fact, the ability to adjust the kernel parameter σ is also so important that it helps improve the solution. It is worthwhile noting that in this research we chose the best pair of **c** and σ using grid search. From the experiments, we found that the values of **c** and σ are not very sensitive to the results. However, from our observations, the value of σ possesses more sensitivity than that of **c**. The best method to determine the best parameters is still an ongoing research area, and a basic approach to choose the parameters is generally by brute force. A better method in determining the values of training parameters or how those parameters affect descriptive boundaries is beyond the scope of this paper and is the subjects of further studies.

5 Conclusions

In this paper, the proposed ellipsoidal SVDD is formed by using the same concept as in SVDD. That is the size of ellipsoids is limited by the minimum volume to cover a given set of data. The soft-margin ellipsoid is also achieved by minimizing one-norm empirical risk. The method also embraces kernel methods to construct an ellipsoid in a higher-dimensional space via empirical feature mapping. In the empirical feature space, which is a finite Euclidean space, the two-norm and inner products are also the same as in the feature space. The experimental results on one-class classification with standard benchmark datasets

showed that eSVDD and neSVDD provided better results than the others for most of the datasets. Therefore, the ellipsoidal SVDDs can be good alternatives to SVDD. For future work, a better approach in incorporating negative samples into eSVDD is required in order to improve numerical stability. A faster MVCE solver is also needed so that the proposed method can perform on larger datasets. It is also interesting to see if the ellipsoidal SVDDs also perform well in multiclass classification.

Acknowledgments The authors would like to thank the anonymous reviewers for their valuable comments to improve this manuscript.

References

1. Ahipaşaoğlu S (2015) A first-order algorithm for the a-optimal experimental design problem: a mathematical programming approach. Stat Comput 25(6):1113–1127
2. Ahipaşaoğlu S (2015) Fast algorithms for the minimum volume estimator. J Global Optim 62(2):351–370
3. Ahmadi A, Dmitry Malioutov RL (2014) Robust minimum volume ellipsoids and higher-order polynomial level sets. In: 7th NIPS workshop on optimization for machine learning, Montreal, Quebec, Canada
4. Barnett V, Lewis T (1994) Outliers in statistical data, 3rd edn. Wiley series in probability and mathematical statistics. Wiley, Chichester
5. Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge
6. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
7. Dolia A, Bie T, Harris C, Shawe-Taylor J, Titterington DM (2006) The minimum volume covering ellipsoid estimation in Kernel-defined feature spaces, Lecture Notes in Computer Science, vol 4212. Springer, Berlin, Heidelberg, pp 630–637
8. Dolia A, Harris C, Shawe-Taylor J, Titterington D (2007) Kernel ellipsoidal trimming. Comput Stat Data Anal 52(1):309–324
9. Glineur F (1998) Pattern separation via ellipsoids and conic programming. Master's thesis, Faculté Polytechnique de Mons, Mons, Belgium
10. Henk M (2012) Löwner–John Ellipsoids. Doc Math (extra volume: optimization stories) pp 95–106
11. Huang G, Chen H, Zhou Z, Yin F, Guo K (2011) Two-class support vector data description. Pattern Recognit 44(2):320–329
12. Khachiyan LG (1996) Rounding of polytopes in the real number model of computation. Math Oper Res 21(2):307–320
13. Kumar P, Yildirim E (2005) Minimum-volume enclosing ellipsoids and core sets. J Optim Theory Appl 126(1):1–21
14. Lasserre JB (2013) A generalization of the Löwner-John's ellipsoid theorem. In: 52nd IEEE Conference on Decision and Control (CDC). Florence, Italy, pp 415–420
15. Löfberg J (2004) YALMIP : a toolbox for modeling and optimization in MATLAB. In: Computer aided control systems design, 2004 IEEE international symposium on, pp 284–289. http://users.isy.liu.se/johanl/yalmip
16. Mu T, Nandi AK (2009) Multiclass classification based on extended support vector data description. IEEE Trans Syst Man Cybern B 39(5):1206–1216
17. PW R, Juszczak P, Paclik P, Pekalska E, de Ridder D, Tax D, Verzakov S (2015) PRTools 5.0. A Matlab toolbox for pattern recognition, software and documentation downloaded March
18. Rimon E, Boyd SP (1997) Obstacle collision detection using best ellipsoid fit. J Intell Robot Syst 18(2):105–126
19. Rosen J (1965) Pattern separation by convex programming. J Math Anal Appl 10(1):123–134
20. Schölkopf B, Mika S, Burges CJC, Knirsch P, Müller KR, Rätsch G, Smola AJ (1999) Input space versus feature space in kernel-based methods. IEEE Trans Neural Netw 10:1000–1017
21. Shioda R, Tunçel L (2007) Clustering via minimum volume ellipsoids. Comput Optim Appl 37(3):247–295
22. Sun P, Freund RM (2004) Computation of minimum-volume covering ellipsoids. Oper Res 52(5):690–706
23. Sylvester JJ (1857) A question in the geometry of situation. Q J Pure Appl Math 1:79
24. Tax DM, Duin RP (1999) Support vector domain description. Pattern Recogn Lett 20(11–13):1191–1199
25. Tax DM, Duin RP (2004) Support vector data description. Mach Learn 54(1):45–66
26. Tax DMJ (2015a) Data sets for one-class classification. http://homepage.tudelft.nl/n9d04/occ/
27. Tax DMJ (2015b) DDtools, the data description toolbox for Matlab. Version 2.1.2
28. Titterington DM (1975) Optimal design: some geometrical aspects of D-optimality. Biometrika 62(2):313–320
29. Todd MJ, Yıldırım EA (2007) On Khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids. Discret Appl Math 155(13):1731–1744
30. Toh KC (1999) Primal-dual path-following algorithms for determinant maximization problems with linear matrix inequalities. Comput Optim Appl 14(3):309–330
31. Tütüncü RH, Toh KC, Todd MJ (2003) Solving semidefinite-quadratic-linear programs using SDPT3. Math Progr 95(2):189–217
32. Vandenberghe L, Boyd S, Wu SP (1998) Determinant maximization with linear matrix inequality constraints. SIAM J Matrix Anal Appl 19:499–533
33. Wei X, Löfberg J, Feng Y, Li Y (2007) Enclosing machine learning for class description. Lect Notes Comput Sci LNCS 4491(Part 1):424–433
34. Wei XK, Li YH, Li YF, Zhang DF (2007b) Enclosing machine learning: concepts and algorithms. Neural Comput Appl 17(3):237–243
35. Xiong H, Swamy MNS, Ahmad MO (2005) Optimizing the kernel in the empirical feature space. IEEE Trans Neural Netw 16(2):460–474
36. Zhang Y, Gao L (2003) On numerical solution of the maximum volume ellipsoid problem. SIAM J Optim 14(1):53–76