One-class Text Classification with Multi-modal Deep Support Vector Data Description

Chenlong Hu†, Yukun Feng†, Hidetaka Kamigaito†, Hiroya Takamura†† and Manabu Okumura†

This work presents multi-modal deep SVDD (mSVDD) for one-class text classification. By extending the uni-modal SVDD to a multiple modal one, we build mSVDD with multiple hyperspheres, that enable us to build a much better description for target one-class data. Additionally, the end-to-end architecture of mSVDD can jointly handle neural feature learning and one-class text learning. We also introduce a mechanism for incorporating negative supervision in the absence of real negative data, which can be beneficial to one-class text models including mSVDD model. We conduct experiments on Reuters, 20 Newsgroup, and TREC datasets, and the experimental results demonstrate that mSVDD outperforms uni-modal SVDD and mSVDD can get further improvements when negative supervision is incorporated.

Key Words: One-class Classification, Text Classification, Support Vector Data Description, Negative Supervision

1 Introduction

One-class classification (OCC), a special classification problem, aims to learn a model on the basis of training samples only from one class. The learned model is expected to make an accurate description of the class (so called target or normal) and then to distinguish the target from samples for negative classes during testing (Moya et al. 1993; Tax 2001). The one-class classification problem has arisen in many real-world applications, including anomaly or novelty detection (Roberts 1999; Chandola et al. 2010; Gupta et al. 2013), bioinformatics (Alashwal et al. 2006), and especially computer vision (Rodner et al. 2011; Ruff et al. 2018).

One-class text classification can be applied to the scenario where a large number of labelled negative samples (e.g., web pages, spam emails) are hard or time-consuming to obtain (Yu et al. 2004; Heron 2009), while positive training data samples (e.g., ham emails) are available. One of the early work on one-class text classification is Manevitz and Yousef (2001), who implemented versions of one-class support vector machines (OC-SVM) (Schölkopf et al. 2001) and showed good

† Tokyo Institute of Technology
†† National Institute of Advanced Industrial Science and Technology (AIST)
performances over the Reuters dataset (Dumais et al. 1998). OC-SVM and support vector data description (SVDD) (Tax and Duin 2004) are boundary-based one-class methods (Tax 2001). Both try to describe the target data using a boundary. SVDD learns an optimal hypersphere with the minimum radius to include the most target data, while OC-SVM builds a hyperplane to maximally separate the data points from the origin where outlier examples lie around.

Reconstruction-based one-class approaches, including AutoEncoder and its variants (Oja 1982; Rumelhart et al. 1985; Manevitz and Yousef 2007) and principal component analysis (PCA) (Pearson 1901; Hotelling 1933; Hoffmann 2007), which aim to learn a more compact representation for the description of target data. The compact representation is a set of prototypes or subspaces obtained by optimizing a reconstruction error on the target training data.

Regarding the features for representing text in OCC, document-to-word co-occurrence matrices or hand-crafted features have been commonly used in most of the previous work (Manevitz and Yousef 2001, 2007; Kumaraswamy et al. 2015). Pre-trained vectors have been popular for many NLP tasks (Mikolov et al. 2013; Bengio et al. 2003). The recent context vector data description (CVDD), proposed by Ruff et al. (2019), fully uses word embedding knowledge and a neural network structure to process one-class text classification problems.

Ruff et al. (2018) introduced deep support vector data description (deep SVDD), a fully unsupervised method for deep one-class classification for image data. Deep SVDD learns to extract the common factors of target training samples with a neural network to minimize the radius of a hypersphere that encloses the network representations of the data. The learned hypersphere, with a center $c$ and a neural feature transformer $\phi(x)$, can be an end-to-end feature learning and one-class classification model. Another piece of work would be one-class neural networks (OC-NN) (Chalapathy et al. 2018), which extends OC-SVM to an end-to-end neural architecture. Instead of a hypersphere in deep SVDD, OC-NN builds a hyperplane on features learned by a feed-forward neural network.

Target data samples may have distinctive distributions that are located in different regions. Therefore, uni-modal deep SVDD with one hypersphere may not be enough to describe the target samples. In this work, we extend deep SVDD to multiple modes, where each mode describes the target samples from a distinctive aspect. Given our multi-modal deep SVDD, $m$SVDD in short, we can create an ensemble set of hyperspheres with different centers to build a better one-class model. Ghafoori and Leckie (2020) proposed deep multi-sphere SVDD (DMSVDD), a similar but different work from ours. We will also discuss the relationship between the two and compare them in the experiments.

In one-class classification, only samples from the target class are available for training, while
the model needs to discriminate between the target class and other classes in testing. Due to the unavailability of training samples from negative classes, it is hard for the one-class models to learn effective discrimination information, especially for mSVDD with a multi-layer neural structure. In this study, we also propose an architecture for improving the discrimination ability of mSVDD by incorporating negative supervision. Specifically, we define two kinds of losses, contrastive and triplet, for joint training with the objective function of mSVDD, which is expected to enhance the discriminative power of mSVDD.

In summary, the main contributions of this work are as follows. 1) We propose a general one-class neural learning framework, called mSVDD, to extend the uni-modal deep SVDD to end-to-end multi-modal. 2) We also prove that four one-class models, deep SVDD, CVDD, DMSVDD, and OC-NN, are all special cases of the mSVDD model. 3) We propose two approaches for effectively incorporating negative supervision information to improve the performance of the proposed mSVDD and other related one-class text classification methods.

The remainder of this paper is structured as follows: Section 2 introduces notations and baseline models. Section 3 describes the proposed mSVDD. Section 4 describes the combinatorial use of mSVDD and negative supervision. We then analyze evaluation results by comparing mSVDD and the baseline models in Section 5. Section 6 summarizes related work for this paper. Finally, we conclude this paper in Section 7.

2 Preliminary

Before describing our mSVDD, we first introduce SVDD (Tax and Duin 2004) and OC-SVM (Schölkopf et al. 2001) and their extensions, deep SVDD (Ruff et al. 2018) and OC-NN (Chalapathy et al. 2018).

2.1 SVDD and One-class SVM

2.1.1 SVDD

Support vector data descriptions (SVDD) is a support vector learning method for one-class classification. It aims at constructing an optimal boundary in a feature space that includes almost all normal target data, given only the target training samples, \( T = \{x_1, ..., x_n\}, x_i \in \mathcal{X} \), where \( n \in \mathbb{N} \) is the size of the training data, and \( \mathcal{X} \) is a compact subset of \( \mathbb{R}^d \). The main idea of SVDD is to optimize a hypersphere with a center \( c \) and radius \( R \), that encloses the majority of the data.
SVDD solves the following quadratic problem:

$$\min_{R,c,\xi} R^2 + C \sum_i \xi_i$$

s.t. \( \|x_i - c\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0, \quad \forall i = 1, \ldots, n, \)

where \( \xi_i \) is a slack variable for allowing a flexible boundary. \( C \) is a regularization parameter, that is usually represented by \( \frac{1}{\nu n} \), where \( \nu \in (0, 1] \) is a parameter that controls the tradeoff between the radius of the hypersphere and the penalties \( \xi_i \).

### 2.1.2 OC-SVM

One-class Support Vector Machines (OC-SVM) (Schölkopf et al. 2001) is another commonly used approach for solving one-class classification problems. Unlike SVDD, OC-SVM intends to seek the best hyperplane by maximizing the margin between the data points and the origin. Formally, given the training data \( T = \{x_1, \ldots, x_n\}, \) \( x_i \in \mathcal{X} \) like SVDD, OC-SVM solves the following optimization problem:

$$\min_{\rho,w,\xi} \frac{1}{2}\|w\|^2 - \rho + \frac{1}{\nu n} \sum_i \xi_i$$

s.t. \( w^T x_i \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad \forall i = 1, \ldots, n, \)

where \( \rho \) is the margin between the origin and the hyperplane \( w \), and \( \xi_i \) is a slack variable for allowing data to cross the hyperplane boundary. As with SVDD, \( \nu \in (0, 1] \) is a hyperparameter that controls the trade-off between the distance from the origin to the hyperplane and the number of data points that are allowed to cross the hyperplane.

The kernel trick (Schölkopf and Smola 2002) can be incorporated into OC-SVM and SVDD to obtain more flexible descriptions. Assuming \( \mathcal{H} \) is reproducing kernel Hilbert space (RKHS) (Saitoh 2003) and \( K(x, x') = \Phi(x) \cdot \Phi(x') \) is a kernel function, \( \Phi(x) : \mathcal{X} \to \mathcal{H} \) becomes an implicit feature mapping function from the input space \( \mathcal{X} \) to the feature space \( \mathcal{H} \). In this case, the objectives of the kernel-based OC-SVM and SVDD (Eq. (2) and Eq. (1)) are to find respectively a hyperplane and a hypersphere in feature space \( \mathcal{H} \). Then, both can be solved in the corresponding dual optimization problems, which are quadratic functions involving the dot products of the transformed vectors. And every dot product can be replaced with a nonlinear kernel function, without knowing the explicit form of \( \Phi(x) \).
2.1.3 Relationship between SVDD and OC-SVM

The identity of SVDD and OC-SVM has been mentioned in several papers (Tsybakov et al. 1997; Tax 2001; Vert et al. 2006). Tax (2001)\(^1\) proved that although the hyperplane obtained by OC-SVM is not a closed boundary around the data that SVDD optimized, both can be represented by an equivalent formulation when data samples are normalized to unit norm vectors. Also for the standard Gaussian kernel, where the data is implicitly normalized, these two one-class approaches are also equivalent to each other.

Several efforts have been proposed to extend kernel-induced OC-SVM and SVDD with multi-modal form. Hao and Lin (2007) was early work to use multi-sphere SVDD, which was used for multi-class tasks. For one-class tasks, (Xiao et al. 2009) used multi-sphere SVDD to encode multi-distribution target data. Multiple kernel learning was proposed for OC-SVM (Das et al. 2010). Two more efforts have been proposed by (Le et al. 2010, 2013), which found the optimal solution by an iterative algorithm consisting of the following two steps: 1) calculate radii and centers, and 2) calculate the assignments of data to centers. While one limitation of SVDD and OC-SVM, along with their extensions, would be that it has to perform hand-crafted feature engineering (Pal and Foody 2010), the limitation can be solved by incorporating neural models into SVDD.

2.2 Deep SVDD and One-class Neural Networks

2.2.1 Deep SVDD

Deep support vector data description (deep SVDD) (Ruff et al. 2018) is an end-to-end deep neural model that not only optimizes the SVDD objective loss but also learns a neural feature transformation. Given target training samples \( T = \{x_1, ..., x_n\} \), deep SVDD first transforms instance \( x \) into a data point of the output feature space with \( \phi \), which is a multi-layer neural network of \( L \in \mathbb{N} \) layers with parameters \( \mathbf{W} = \{ \mathbf{W}^1, ..., \mathbf{W}^L \} \). Deep SVDD defines two kinds of loss functions:

**Soft-boundary deep SVDD:**

\[
\frac{1}{\nu n} \sum_i \max(0, \| \phi(x_i; \mathbf{W}) - c \|_2^2 - R^2) + R^2 + \frac{\lambda}{2} \sum_l \| \mathbf{W}^l \|_F^2
\] (3)

The first penalty term is for samples lying outside the sphere, i.e., when the distance of \( x_i \) to the center, \( \| \phi(x_i; \mathbf{W}) - c \|_2^2 \), is greater than radius \( R \) after the transformation by network \( \phi \). The above loss also regularizes the radius and neural weight parameters in the second term. As with

---

\(^1\) Section 2.5.
SVDD, parameter $\nu \in (0, 1]$ adjusts the tradeoff between the radius of the hypersphere and the points outside the hypersphere.

Schölkopf et al. (2001) proved that, in single-class classification, $\nu$ is the upper bound of the fraction of anomalies, and the lower bound of the fraction of training samples being anomalies or on the optimal boundary. Ruff et al. (2018) proved that this $\nu$-property still holds for uni-modal soft-boundary deep SVDD.

Another simplified objective that minimizes the mean distance of all positive training samples to the center, the one-class form, can be defined as follows:

**One-class deep SVDD (simplified form):**

$$\frac{1}{n} \sum_i \|\phi(x_i; W) - c\|_2^2 + \frac{\lambda}{2} \sum_l \|W^l\|_2^2$$

(4)

Here, we can rewrite both the above in a unified form:

$$\mathcal{L}_{\text{DSVDD}} = C \sum_i [\|\phi(x_i; W) - c\|_2^2 - \beta]_+ + \beta + \frac{\lambda}{2} \sum_l \|W^l\|_2^2,$$

(5)

where $[\cdot]_+ = \max\{0, \cdot\}$, $\beta \in \{0, R^2\}$ and regularization parameter $C \in \{\frac{1}{n}, \frac{1}{\nu n}\}$ correspond to the two types of forms.

### 2.2.2 One-class Neural Networks

Chalapathy et al. (2018) proposed one-class neural networks (OC-NN), which extends OC-SVM to a neural architecture. As with deep SVDD, OC-NN also uses an end-to-end neural network to perform feature learning and to optimize a OC-SVM equivalent loss objective function. Given a training set $T = \{x_1, ..., x_n\}$ and an $L$-layer neural network $\phi$ with parameter $W$, the objective of OC-NN is:

$$\frac{1}{2}\|w\|_2^2 + \frac{1}{\nu n} \sum_i \max(0, \rho - w^T \phi(x_i; W)) - \rho + \frac{\lambda}{2} \sum_l \|W^l\|_2^2,$$

(6)

where the key idea is to use $\phi$ to learn the feature transformation for each instance $x$. Similar to soft-boundary deep SVDD (Eq. (3)), OC-NN also uses $\nu$ to regularize the loss.

We will discuss the relationship between deep SVDD and OC-NN in Section 3.3.4.

### 3 Multi-modal Deep SVDD

In this section, we present our mSVDD, a method for deep one-class classification. Unlike a uni-modal model with a hypersphere, mSVDD uses a set of hyperspheres to describe target
class data and to reject samples from negative classes. Fig. 1 shows the general idea of mSVDD with two modes. Consider that we have $m$ modes, each of which is described by a hypersphere $M_j$ with center $c_j$ and radius $R_j$; mSVDD uses each $M_j$ to describe a distinctive aspect of the target class and then ensemble them. This ensembled deep mSVDD model can provide better descriptions for the target data.

As with deep SVDD, given target training samples $T = \{x_1, ..., x_n\}$, mSVDD first transforms instance $x$ into a data point of the output feature space with $\phi$, where $\phi$ is a deep neural network of $L \in \mathbb{N}$ layers with parameters $\mathcal{W} = \{W^1, ..., W^L\}$. In contrast to deep SVDD, mSVDD uses $m$ hyperspheres to include almost all of the target data with the minimum radii, i.e., $\frac{1}{m} \sum_{j=1}^{m} R_j^2$. The objective of kernel SVDD and deep SVDD is to minimize the volume of a data-enclosing hypersphere in feature space that is represented by center $c$ and radius $R$. Therefore, to minimize the volume, deep SVDD punishes points lying outside the sphere, i.e., if the distance of $x$ to the center $c$, $\|\phi(x; \mathcal{W}) - c\|^2$, is greater than radius $R$. Since we have a set of hyperspheres $M = \{M_1, ..., M_m\}$, one choice would be to punish $x$ with respect to each hypersphere by adding $\sum_j \max(0, \|\phi(x; \mathcal{W}) - c_j\|^2 - R_j^2)$ to the loss function. However, the above penalty term does not take into account the different influences from multiple hyperspheres together. Therefore, we would like to take another ensembled constraint by incorporating attention mechanism. Attention mechanism has been adapted to various tasks, allowing models to learn the weights of focusing on different contexts (Luong et al. 2015). Thus, it is naturally to incorporate this mechanism to assign a weight for each sample to each center. Given non-negative attention weight $\alpha_{ij}$ for $x_i$ to

![Diagram](image_url)
each $M_j$, the penalty term can be computed as the weighted average over $m$ constraints. Now, only one ensembled constraint is required, i.e., the sum of radii is greater than the sum distance to the center. Formally, we can define our mSVDD objective as follows:

$$L_{\text{soft-mSVDD}} = \frac{1}{\nu n} \sum_i \max(0, \sum_j \alpha_{ij}(\|\phi(x_i; W) - c_j\|_2^2 - R_j^2)) + \frac{1}{m} \sum_j R_j^2 + \frac{\lambda}{2} \sum_l \|W^l\|_F^2, \quad (7)$$

where $\frac{1}{m} \sum_{j=1}^m R_j^2$ is the regularization term for radii from all $m$ hyperspheres to get a closer boundary around the target data. This form can be seen as mSVDD with weighted soft-boundary constraints, which we call soft-boundary mSVDD.

Although the $\nu$-property, mentioned in Section 2.2.1, does not hold true for our multi-modal case as it is in general, it is still true when the attention weight $\alpha_{ij}$ is constant for different hyperspheres. This will give us an intuition on the role of $\nu$.

Proposition 1. The $\nu$-property holds if we set equal attention weight to each hypersphere: i.) $\nu$ is an upper bound for the fraction of outlier samples and ii.) $\nu$ is a lower bound for the fraction of training samples being rejected or on the optimal boundary.

3.1 One-class mSVDD (Simplified Form)

As in deep SVDD, we also have the simplified form and called: one-class mSVDD. If we assume that the majority of the training data is not anomalous, then the radius can be ignored and we can define the simplified mSVDD as follows:

$$L_{\text{oc-mSVDD}} = \frac{1}{n} \sum_i \sum_j \alpha_{ij} \|\phi(x_i; W) - c_j\|_2^2 + \frac{\lambda}{2} \sum_l \|W^l\|_F^2, \quad (8)$$

where the attention weight $\alpha_{ij}$ will be kept, while the penalty of radius $R$ is deleted.

3.2 Unified Form of mSVDD

We can write the two variants of mSVDD (i.e., soft-boundary mSVDD and simplified one-class mSVDD) in a unified form:

$$L_{\text{mSVDD}} = C \sum_i \sum_j \alpha_{ij} (\|\phi(x_i; W) - c_j\|_2^2 - \beta_j)_+ + \frac{1}{m} \sum_j \beta_j + \frac{\lambda}{2} \sum_l \|W^l\|_F^2, \quad (9)$$

where $\beta_j \in \{0, R_j^2\}$. $\beta_j = 0$ corresponds to simplified one-class mSVDD, and $\beta_j = R_j^2$ corresponds to soft-boundary mSVDD. For $x_i$ to the $j$-th hypersphere, attention weight $\alpha_{ij}$ should be inversely

---

2 The proofs of propositions can be found in the appendix.
3 The term “one-class” is used following (Ruff et al. 2018). Note that most of the models discussed in this paper are one-class models.
proportional to its distance to center $c_j$. Thus, we define:

$$
\alpha_{ij} = \exp\left(\frac{d(x_i, c_j)}{\delta}\right) = \frac{\exp(\|\phi(x_i; W) - c_j\|^2/\delta)}{\sum_{k=1}^m \exp(\|\phi(x_i; W) - c_k\|^2/\delta)}
$$

(10)

where $\delta < 0$ is a temperature hyperparameter.

### 3.3 Discussions on Relationships between mSVDD and Other Models

In this subsection, we would like to discuss the relationships between mSVDD and other related models. First, it will help us to understand the differences, and more importantly the common characteristics of deep SVDD-related approaches. Second, these discussions would give us an unified view on the development of deep SVDD-related approaches in a limited range. Further, it also provides possible future directions. For example, if one model obtained improvements from certain method, e.g., negative supervision introduced later, this may motivate other deep SVDD-related models.

#### 3.3.1 Relationship between mSVDD and Uni-modal Deep SVDD

Their relationship is obvious and can be summarized by the following proposition.

**Proposition 2.** Deep SVDD is a special case of the unified form of mSVDD with one hypersphere used.

**Proof.** Obviously, mSVDD becomes uni-modal (Ruff et al. 2018) if we use only one hypersphere, i.e., $m = 1$.

#### 3.3.2 Relationship between mSVDD and CVDD

CVDD (Ruff et al. 2019) is a one-class model for text data. In CVDD, each training sample $x_i$ (i.e., a text) is represented by $r$ self-attention feature vectors $S_i = (s_{i1}, ..., s_{ir})$ (Lin et al. 2017). CVDD uses a group of $r$ context vectors $C = (c_1, ..., c_r)$ to describe the target one-class data, where $c_k \in \mathbb{R}^p$. CVDD tries to reduce the one-to-one reconstruction distance between feature vectors $S_i$ and context vectors $C$. The loss can be defined as:

$$
L_{CVDD} = \frac{1}{n} \sum_i \sum_k \sigma_{ik} d(c_k, s_{ik}),
$$

(11)

where $d(c_k, s_{ik})$ computes the distance, and $\sigma_{ik}$ denotes the attention weight. The following proposition implies the close connection between the two models for one-class text classification.

**Proposition 3.** CVDD is a very special case of one-class mSVDD when mSVDD is applied to text-based tasks under certain conditions.
Proof. W.L.O.G., rewrite the loss function of one-class mSVDD in a simplified form for each sample as follows:

\[ L_{mSVDD} = \frac{1}{n} \sum_i \sum_j \alpha_{ij} \left\| \phi_j (x_i; W) - c_j \right\|^2 \]

\[ = \frac{1}{n} \sum_i \sum_j \sigma_{ij} d(x_i, c_j) \approx L_{CVDD}, \]

where we drop the regularization terms for weights of \( \phi \) and radii, and set \( m = r, \sigma_{ij} = \alpha_{ij}, d(x_i, c_j) = \left\| \phi_j (x_i; W) - c_j \right\|^2 \). \( \phi (x_i; W) \) has to be a self-attention neural model, \( \phi_j (x_i; W) \) is the \( j \)-th feature vector of sample \( x_i \), and \( c_j \) is the \( j \)-th context vector of target samples. Now the loss functions of CVDD and one-class mSVDD are almost the same. \[ \square \]

[A-(d)] As stated before, CVDD uses a different multi-head structure for text feature learning, while the common point would be both adopt the simplified form of mSVDD loss. In the experiments, we will show that the proposed negative supervision for mSVDD (Section 4) can also be used for CVDD due to the similarity of both.

### 3.3.3 Relationship between mSVDD and DMSVDD

DMSVDD (Ghafoori and Leckie 2020) also uses multi-hyperspheres to extend SVDD. The loss function of DMSVDD is as follows:

\[ L_{DMSVDD} = \frac{1}{\nu n} \sum_i \left[ \left\| \phi(x_i; W) - c_i^* \right\|^2 - R_i^2 \right]_+ + \frac{1}{K} \sum_k R_k^2 + \frac{\lambda}{2} \sum_l \left\| W_l \right\| F^2, \] (12)

where \( K \) is the number of hyperspheres\(^4\), \( c_i^* \) is the nearest center of sample \( x_i \) and \( R_i^* \) is its radius.

**Proposition 4.** DMSVDD can be seen as a hard-version of soft-boundary mSVDD if we set the attention weight in some way.

**Proof.** In the calculation of the attention weight for mSVDD with \( \alpha_{ij} = \frac{\exp(d(x_i, c_j/\delta))}{\sum_{k=1}^{K} \exp(d(x_i, c_k)/\delta)} \), the temperature parameter \( \delta \) can influence the assignment of center \( c_k \). If we set \( \delta \to 0^- \), the above formula acts as the argmin operation.\(^5\) In this case, \( \alpha_{ii*} = 1 \) if \( i* = \arg\min_{k=1,...,K} d(x_i, c_k) \), and 0 otherwise. Now, we can get the form of DMSVDD from soft-boundary mSVDD (Eq. (7)) through the adjustment of the attention weight. Therefore, we can prove that DMSVDD is also a special case of mSVDD. \[ \square \]

---

\(^4\) In DMSVDD, \( K \) changes dynamically. However, we ignore this difference and focus on the comparison of the models.

\(^5\) \( \delta \) approaches 0 from the negative side.
The above relation illustrates key difference between them: DMSVDD puts value on one hypersphere with the largest weight.

3.3.4 Relationship between mSVDD and OC-NN

As mentioned, OC-NN and deep SVDD (Chalapathy et al. 2018; Ruff et al. 2018) extend one-class SVM and SVDD (Schölkopf et al. 2001; Tax and Duin 2004) to their neural network structures, respectively. Tax (2001) has discussed and proved the equivalence of OC-SVM and SVDD. Therefore, in this subsection, we discuss the relationships among the three neural models: OC-NN, deep SVDD and mSVDD. Since deep SVDD is the uni-modal version of mSVDD (See Proposition 2), we focus on the discussion of the relationship between mSVDD and OC-NN. We provide their relationship in the following proposition.

**Proposition 5.** i.) OC-NN can be extended to the multi-modal version. ii.) Soft-boundary mSVDD can be seen as multi-modal OC-NN under certain conditions.

**Proof.** Ad (i). First, we show that OC-NN can be extended to its multi-modal version. OC-NN uses one hyperplane to separate the target data from the origin. Actually we can also use a set of hyperplanes to separate target class data in the feature space, which is a similar idea to mSVDD. We give the objective function of multi-modal OC-NN as follows:

\[
L_{mOCNN} = \frac{1}{νn} \sum_i \max(0, \sum_j α_{ij}(ρ_j - w_j^T ϕ(x_i; W))) + \frac{1}{m} \sum_j ||w_j||^2 - \frac{1}{m} \sum_j ρ_j + \frac{λ}{2} \sum_l ||W^l||_F^2,
\]

where \(w_j\) and \(ρ_j\) denote the norm perpendicular and the margin of the \(j\)-th hyperplane in the neural feature space, respectively. For one data sample \(x_i\), the ensembled constraint term, \(\max(0, \sum_j α_{ij}(ρ_j - w_j^T ϕ(x_i; W)))\), denotes that the sum of margins should be less than the sum of \(x_i\)'s distances to the origin. The definition of weight \(α\) is the same as in mSVDD (Eq. (10)).

Ad (ii). Then, we prove the equivalence of soft-boundary mSVDD and multi-modal OC-NN. For clarity, we use \(f_i = ϕ(x_i; W)\) to denote the feature vector of \(x_i\). And we also assume the feature vector \(f\) is normalized \((\hat{f} = f/\sqrt{\sum ||f_i||^2})\), so does \(c\). As shown in Fig. 2, in this case, all the data representations and centers are mapped onto an unit hypersphere. In this case, a simplified formulation for the loss of mSVDD (Eq. (7)) with normalized \(\hat{f}\) and \(\hat{c}\) would be:

\[
L_{mSVDD} = \frac{1}{νn} \sum_i \max(0, \sum_j α_{ij}(||\hat{f}_i - \hat{c}_j||^2_2 - R_j^2)) + \frac{1}{m} \sum_j R_j^2,
\]

We further transform Eq. (14) by using the relation \(||\hat{f}_i - \hat{c}_j||^2_2 = ||\hat{f}_i||^2_2 - 2\hat{f}_i^T \hat{c}_j + ||\hat{c}_j||^2_2 = 2 - 2\hat{f}_i^T \hat{c}_j\),

1063
Fig. 2 Multi-modal Deep SVDD and OC-NN with two modes, where the data representations are mapped to unit norm hypersphere. Black points denote normalized data samples.

(a) Multi-modal Deep SVDD with two hyperspheres represented by \((c_1, R_1)\) and \((c_2, R_2)\).

(b) Multi-modal OC-NN with two hyperplanes represented by \((w_1, \rho_1)\) and \((w_2, \rho_2)\).

\[
\mathcal{L}_{mSVDD} = \frac{1}{\nu n} \sum_i \max(0, \sum_j \alpha_{ij} (2 - 2 \hat{f}_i^T \hat{c}_j - R_j^2)) + \frac{1}{m} \sum_j R_j^2
\]

\[
= \frac{1}{\nu n} \sum_i \max(0, \sum_j \alpha_{ij} (1 - \frac{1}{2} R_j^2 - \hat{c}_j^T \hat{f}_i)) + \frac{1}{m} \sum_j R_j^2
\]

\[
= \frac{2}{\nu n} \sum_i \max(0, \sum_j \alpha_{ij} (\rho_j^{'-} - \hat{c}_j^T \hat{f}_i)) + \frac{2}{m} \sum_j (1 - \rho_j^{'})
\]

\[
= 2 \left( \frac{1}{\nu n} \sum_i \max(0, \sum_j \alpha_{ij} (\rho_j^{'-} - \hat{w}_j^T \hat{f}_i)) + \frac{1}{m} \sum_j \|\hat{w}_j\|^2_2 - \frac{1}{m} \sum_j \rho_j^{'}) \right)
\]

\[
= 2 \mathcal{L}_{mOCNN}, \tag{15}
\]

where we redefine \(\rho_j^{'-} = 1 - \frac{1}{2} R_j^2, \|\hat{w}_j\|^2_2 = 1, \forall j \in \{1, ..., m\}\). The above formulation indicates that the loss \(mSVDD\) is equal to the loss of multi-modal OC-NN, \(\mathcal{L}_{mOCNN}\) (Eq. (13)), up to a factor 2. Therefore, we can prove the equivalence of soft-boundary \(mSVDD\) and multi-modal OC-NN under the condition of normalization. Certainly, the identity can hold when we use uni-modal deep SVDD and OC-NN.

As for the geometric interpretation, Fig. 2 illustrates how the two models show when data representations are normalized and mapped onto unit hypersphere. The two subfigures in Fig. 2 show multi-modal OC-NN and SVDD with two modes, respectively. As shown in Fig. 2, each data sample can be regarded as a point on the unit hypersphere, and maximizing the margins.
of hyperplanes from the origin in OC-NN is equal to minimizing the radii of hyperspheres in SVDD. We can also get the relationship from \( \rho'_j = 1 - \frac{1}{2} R_j^2 \) which shows that the margin \( \rho'_j \) is inversely proportional to the radius \( R_j \). During the (maximizing/minimizing) optimization process in OC-NN and mSVDD, these two neural models can also perform the feature learning such that data representations are rotated to their “center” representations (i.e., \( w \) and \( c \)) on the unit hypersphere (Liu et al. 2017; Deng et al. 2019; Yokoi et al. 2020).

### 3.4 Summarizing mSVDD

We summarize the proposed mSVDD in accordance with the discussions presented above. The proposed multi-modal deep SVDD (mSVDD) learns a compact description of one-class data with multiple hyperspheres. mSVDD is also a generic framework that includes deep SVDD, CVDD, DMSVDD, and OC-NN if the corresponding conditions are met.

### 4 Multi-modal Deep SVDD with Negative Supervision

In this section, we incorporate negative supervision into the training of mSVDD. The SVDD-related models are usually trained with only positive samples from the target one-class, while, if negative samples are available, the models can be extended to train with them to improve the description (Tax 2001).\(^6\) Note that these samples are not necessarily required to be from “real” negative class. In our experiment, external data can be seen as one of the choices for the incorporated pseudo-negative samples. Note that, the proposed negative supervision can be compatible with other SVDD-related one-class text classification methods, e.g., CVDD (see Section 5.2.4), not just mSVDD.

Given a set of extended training samples \( T' = \{(x_1, y_1), \ldots, (x_n', y_n')\} \), where the first \( n \) samples are labeled \( y_i = 1 \), denoting positive, whereas the others are labeled \( y_i = 0 \), which denotes negative samples that should be rejected by mSVDD. Our mSVDD is represented with \( m \) hyperspheres and is formulated as \( M = \{M_1(c_1, R_1), \ldots, M_m(c_m, R_m)\} \). It is required that the positive samples should be inside the \( m \) hyperspheres, while the negative samples should lie outside. Given training samples composed of positive and a negative samples, we can first get their corresponding distances to each center \( c_j \). The goal of optimization should be to pull the positive samples closer the center and to push the negative ones away. Formally, we define the distance between one sample \( x_i \) and one center \( c_j \) as \( d_{ij} = d(x_i, c_j) = \|\phi(x_i; W) - c_j\|_2^2 \). There

---

\(^6\) Section 2.2.
are usually two types of losses to obtain the discriminative loss.

**Contrastive type:** The contrastive-type loss directly optimizes the distance by encouraging the distance between a positive sample and a center to be smaller, while it forces the larger distance to a negative sample:

\[
L_{\text{Con}}^{(ij)} = y_i [d_{ij} - R_j^2]_+ + (1 - y_i) [R_j^2 - d_{ij}]_+ ,
\]  
(16)

where \( R_j^2 \) can be seen as a margin (or threshold) with a function that prevents too much effort from being wasted in enlarging/reducing distances (Hadsell et al. 2006).

**Triplet type:** The triplet-type loss is defined for a pair of positive sample \( x_i \) and negative sample \( x_{i'} \). If we consider center \( c_j \) as an anchor representative of target data, the triplet loss punishes only when \( d_{ij} \), the distance from \( x_i \) to \( c_j \), is greater than \( d_{i'j} \), the distance from \( x_{i'} \) to \( c_j \), with a margin \( \tau > 0 \):

\[
L_{\text{Tri}}^{(ij)} = [d_{ij} - d_{i'j} + \tau]_+ .
\]  
(17)

For clarity, Eqs. (16) and (17) show only the two types of losses for one hypersphere. Multi-modal version can be obtained by sum operation over \( j \in \{1, ..., m\} \).

The triplet loss forces only positive samples to be closer to the center than negative samples, and the contrastive loss requires only keeping the distances for negative samples above the radius. These two types of objectives are easy to achieve, especially when we assume that negative samples are “not real.” This may result in failing to make a full use of negative supervision. Therefore, we will reformulate both \( L_{\text{Tri}} \) and \( L_{\text{Con}} \).

### 4.1 Reformulating Contrastive and Triplet Losses

**Normalization layer** In neural models with the contrastive or triplet loss, it is a common strategy to normalize the feature representations of samples for training stability (Schroff et al. 2015; Wang et al. 2017). Therefore, we apply the normalization to the input vectors: \( \hat{x} = x / \sqrt{\sum |x_i|^2 + \epsilon} \), where \( \epsilon > 0 \) is a value avoiding division by zero.

**Reformulation** Given a center \( c_j \) and positive and negative samples, we can use the probability form in the optimization objective, rather than the two non-probabilistic ones: \( L_{\text{Con}} \) and \( L_{\text{Tri}} \). We introduce \( p(y_i = 1|x_i, c_j) \), which is the probability that a hypersphere with center \( c_j \) accepts the sample \( x_i \), and define it as follows:

\[
p(y_i = 1|x_i, c_j) = \sigma(s \hat{f}_i^T \hat{c}_j),
\]  
(18)

where \( \sigma(x) = \frac{1}{1 + \exp(-x)} \), \( \hat{f}_i = \phi(x_i; W) \) denotes the feature output vector of \( x_i \), and \( s \) is a scale
hyper-parameter for preventing failed convergence (Wu et al. 2018) after the normalization. For each sample $\mathbf{x}_i$, $\mathbf{c}_j$ acts as a pseudo-weight vector for the classification of the $j$-th hypersphere of mSVDD. Thus, given $p(y_i = 1|\mathbf{x}_i, \mathbf{c}_j)$, the probability of a sample being accepted by hypersphere $M_j$, we can reformulate the two discriminative losses with the probability.

**Contrastive type loss:**

$$
L^{(ij)}_{\text{Con}} = -y_i \log p(y_i = 1|\mathbf{x}_i, \mathbf{c}_j) - (1 - y_i) \log p(y_i = 0|\mathbf{x}_i, \mathbf{c}_j)
$$

$$
= -y_i \log \sigma(\hat{s}f_i^T \hat{c}_j) - (1 - y_i) \log \sigma(-\hat{s}f_i^T \hat{c}_j),
$$

This loss maximizes the likelihood of training positive samples being accepted or negative rejected.

**Triplet type loss:**

$$
L^{(i'i')}_{\text{Tri}} = \left[ \log p(y_{i'} = 1|\mathbf{x}_{i'}, \mathbf{c}_j) - \log p(y_i = 1|\mathbf{x}_i, \mathbf{c}_j) + \tau \right]_+
$$

$$
= \left[ \log \sigma(\hat{s}\hat{f}_{i'}^T \hat{c}_j) - \log \sigma(\hat{s}\hat{f}_i^T \hat{c}_j) + \tau \right]_+
$$

(20)

The loss will punish when the log probability of a negative sample is greater than a positive sample with a margin $\tau$.

### 4.2 Reformulating Contrastive and Triplet Losses for Multiple Modes

While Eqs. (18), (19), and (20) show the uni-modal case, for the multi-modal one, we have to consider $m$ different centers $\{\mathbf{c}_1, \ldots, \mathbf{c}_m\}$ in the calculation of the two reformulated discriminative losses. Therefore, we propose two strategies as follows:

$$
p(y_i = 1|\mathbf{x}_i) = \begin{cases} 
\max_{1 \leq j \leq m} p(y_i = 1|\mathbf{x}_i, \mathbf{c}_j) & \text{Max} \\
\frac{1}{m} \sum_j p(y_i = 1|\mathbf{x}_i, \mathbf{c}_j) & \text{Mean},
\end{cases}
$$

(21)

where Max references only $M_j$ with the max logit output, while Mean takes account of all hyperspheres equally. Then, we can obtain the corresponding Contrastive and Triplet losses by substituting Eqs. (19) and (20) with the probability term (Eq. (21)).

### 4.3 Training Loss

The final training loss for the mSVDD with negative supervision can be formulated as:

$$
\mathcal{L} = \mathcal{L}_{\text{mSVDD}} + \gamma \mathcal{L}_{\text{Con}|\text{Tri}},
$$

(22)

where $\gamma$ adjusts between the mSVDD loss and the discrimination with negative supervision. In the training process, $\mathcal{L}_{\text{Con}|\text{Tri}}$ will sum the loss from one batch samples with Eqs. (19) and (20).
4.4 Relationship between mSVDD and the Use of Negative Supervision

mSVDD and negative supervision are not two independent sub-architectures. Negative supervision, including contrastive and triplet losses, are specially equipped to mSVDD. Specifically, these two components are closely connected by the center of the hypersphere, $c_j$. Both mSVDD (Eq. (9)) and negative supervision (Eq. (19) or (20)) contain $c_j$. Since there is no real negative data, external data are used as pseudo negative samples to complete negative supervision. The use of negative supervision can improve the discrimination ability of mSVDD. In training, negative supervision loss forces mSVDD to reject unseen samples since real negative data in testing are also unseen in training. This improves inter-class discrepancy, compared with intra-class loss mSVDD optimized. However, in testing, the decision function will be the same as mSVDD trained only with positive samples.

5 Experiments

5.1 Datasets and Implementation Details

Datasets Experiments were conducted on two datasets: 20 Newsgroups\(^7\) and Reuters\(^8\) which have been commonly used in other one-class text classification work (Manevitz and Yousef 2001; Ruff et al. 2019). We also conducted experiments on another dataset TREC\(^9\) (Li and Roth 2002). We follow the same one-class classification setting in (Ruff et al. 2019), i.e., one of the classes in each dataset is considered as the target class, we call it as target or positive, while the remaining classes are considered negative. Note that we labelled positive samples as $y = 1$, while negative ones $y = 0$. We used the same pre-processing steps as the ones used in earlier work (Ruff et al. 2019), including lowercasing, removing stopwords, and tokenization. We used the external data for negative supervision in the absence of “real” labeled negative instances. We followed the similar logic for choosing our external data as the one in the field of pretrained word vectors, in which one general corpus, such as Wikipedia articles, is often adopted as the training dataset (Mikolov et al. 2013). So we also chose one publicly available corpus WikiText-2 (Merity et al. 2016), extracted from Wikipedia articles, as our external data. In the training, data loader loads one batch of negative samples, i.e., sentences from WikiText-2, which are labeled with 0.

Encoder For encoding the text input, i.e., $\phi(x,W)$, we used a Bidirectional LSTM with atten-

---

\(^7\) http://qwone.com/\texttt{json/20Newsgroups}

\(^8\) http://daviddilevi\texttt{s.com/resources/testcollections/reuters21578/}

\(^9\) https://cogcomp.seas.upenn.edu/Data/QA/QC/
tion (Hochreiter and Schmidhuber 1997; Xu et al. 2015), with the number of hidden units being 150. For the pre-trained word embeddings, we experimented with GloVe Vectors (Pennington et al. 2014) and set the dimension to 300. In our experiments, we did not adopt the widely used BERT model (Devlin et al. 2019), as Ruff et al. (2019) showed that BERT model did not improve the performance.

**Settings**  
As for the optimization of parameters, Adam (Kingma and Ba 2014) with a base learning rate of 0.001 was used for 50 epochs. The batch sizes were set to 32, 32, and 64 for Reuters, TREC, and Newsgroups, respectively. For the initialization of mSVDD model, we employed two operation steps. In the absence of negative samples, mSVDD was first pre-trained on target samples by using an AutoEncoder with two objectives: 1) warm-up and 2) reducing the reconstruction error for the target samples, such that the model can be more robust to noise or anomalous inputs (Jacobs 1995; Hinton and Salakhutdinov 2006). An AutoEncoder feed-forward network with a 0.5 compression rate, which consists of an encoder and a decoder, was put on the back of the BiLSTM feature network. Then, the weights of the $m$ hyperspheres in mSVDD were initialized by running $k$-means clustering on the features learned before (Lloyd 1982). As for the regularization term of mSVDD, $c_j$ was regularized (Ng 2004), and a weight decay with 0.95 was applied for the parameters. As for the number of hyperspheres, different settings, 1, 3, 5, 10, were tested. For the hyperparameters, we set parameter $s = 1.2$ for scale, $\nu = 0.1$, $\delta = -0.9$ for the attention weight, $\tau = 0.1$ for the triplet loss, $\epsilon = 1e^{-6}$ for norm, and $\gamma = 1$ for the training loss. The results were averaged over 10 runs with different random seeds.

**Evaluation metrics**  
The performance was measured by the area under the receiver operating characteristics (ROC) curve (AUCs), a commonly used metric for one-class text classification (Manevitz and Yousef 2001; Ruff et al. 2019).

### 5.2 Results

#### 5.2.1 Comparison between multi-modal and unit-modal

Table 1 shows the performance of mSVDD with different choices of $m$, i.e., the number of hyperspheres. Here, mSVDD(1) represents uni-modal deep SVDD (DSVDD) (Ruff et al. 2018). The results show that: 1) mSVDD outperformed the uni-modal DSVDD under most target classes over three datasets. Generally, multi-modal mSVDD provided better performances by obtaining best scores in more times than the uni-modal one. As for the one-class version, compared to mSVDD(3), mSVDD with more hyperspheres (5) setting performed better in Reuters and TREC, while there was no improvement found on Newsgroup. mSVDD(3) also shows improvements on four target classes on TREC datset. Similar results can also be observed for soft-boundary
Table 1 Results of mSVDD with different settings of $m$. Numbers in brackets denote the number of hyperspheres in mSVDD. One-v and Soft-v denote the two versions of mSVDD, One-class and Soft-boundary, respectively. AUCs in % on the Reuters (upper part), 20 Newsgroup (middle part) and TREC (lower part) datasets. Best scores in each row are presented in bold within each One-v and Soft-v group. Significant improvements over uni-modal have been marked with a dagger ($p < 0.05$).

| Target Class | One-v mSVDD($m$) | Soft-v mSVDD($m$) |
|--------------|------------------|------------------|
|              | 1    | 3    | 5    | 10   | 1    | 3    | 5    | 10   |
| **Reuters**  |      |      |      |      |      |      |      |      |
| earn         | 95.6 | 95.5 | **96.0** | 95.9 | 95.9 | 95.9 | **96.2** | 96.1 |
| acq          | 89.4 | 90.0† | 89.3 | **90.1** | 89.0 | 89.1 | 89.1 | **89.2** |
| crude        | **92.7** | 92.5 | 92.5 | 92.4 | **92.8** | 91.5 | 92.5 | 92.4 |
| trade        | 98.4 | 98.3 | **98.8**† | 98.6† | 98.3 | **98.9**† | 98.7 | 98.8 |
| money        | 86.3 | 85.1 | 86.4 | **87.1** | 86.2 | 86.2 | 86.4 | **86.8** |
| interest     | 97.2 | 97.2 | 97.2 | **97.3** | 97.3 | 96.9 | 96.6 | 96.8 |
| ship         | 92.5 | **93.8**† | **93.8**† | 92.6 | 91.7 | 91.6 | **92.3**† | 91.7 |
| **20 News**  |      |      |      |      |      |      |      |      |
| comp         | 85.3 | 86.2 | 86.1 | **86.7** | 84.9 | 86.0 | 85.9 | **86.5** |
| rec          | 77.1 | **77.7**† | 77.6 | 77.6 | 76.2 | **77.0**† | 76.9 | 76.8 |
| sci          | 66.5 | **67.3**† | 67.1 | 66.9 | 66.3 | **67.3**† | 66.7 | 67.0 |
| misc         | 75.2 | **76.0**† | 75.5 | 75.5 | 75.0 | **76.2** | **76.2** | **76.2** |
| pol          | **79.2** | 78.5 | 78.7 | 78.5 | **79.1** | 78.7 | 78.4 | 78.4 |
| rel          | **83.6** | 83.1 | 83.1 | 83.1 | **82.5** | **82.5** | 82.3 | 82.2 |
| **TREC**     |      |      |      |      |      |      |      |      |
| ENTY         | 71.8 | 72.4 | **72.5**† | 72.3 | 70.9 | 71.8 | **72.0** | 71.8 |
| HUM          | 75.3 | **76.4**† | **76.4** | **76.4** | 75.7 | 76.6 | **76.8**† | 76.7 |
| DESC         | **52.9** | **52.9** | 52.6 | 52.4 | 54.5 | **55.3**† | **55.3**† | 54.9 |
| NUM          | 73.7 | 74.5† | **74.7** | 74.6 | 73.4 | 74.4 | **74.5** | 74.4 |
| LOC          | 79.0 | **79.7** | **79.7** | 79.6 | 78.7 | 79.5 | **79.6**† | 79.4 |
| ABBR         | **97.6** | 96.6 | 96.3 | 94.7 | **97.3** | 96.9 | 96.4 | 96.7 |

2) The performance of mSVDD did not improve linearly along with $m$. For examples, except for comp in Newsgroup dataset, mSVDD(10) with one-class form didn’t achieve best scores. While mSVDD(3 or 5) perform the best in most target classes over three datasets in the one-class version. On the three target classes, rec, sci and misc, soft-boundary mSVDD(3) shows consistent performances. It shows also that the soft-boundary mSVDD(5) achieved the best scores on five out of six cases in TREC dataset. The results show the effects of incorporating more hyperspheres to better describe the target data.
loss setting. This indicates that the incorporation of more descriptions is not necessary sometimes. Fig. 3 shows an example with t-SNE (Van der Maaten and Hinton 2008) visualization of feature representations learned by DSVDD, mSVDD, and CVDD. For the data inputs for target rec, mSVDD maps data to multiple clusters in the feature space which matches the assumption that there are multiple sub-groups on rec. While DSVDD maps them into one cluster, and CVDD outputs feature representations separated by five heads. As for the performance, both forms of mSVDD obtained better scores than CVDD or DSVDD on rec.

As for uni-modal has good results in some cases such as ABBR, pol, and rel. We can explain this from the following aspects. As for the model, mSVDD with more centers means that it has more parameters and a complex model structure, which is hard to be optimized, especially on the data with a small training size (e.g., pol or rel.) As for the data, some data might have simple data distributions without the need for more modes. Another aspect would be the attention weights of multiple hyperspheres. Ghafoori and Leckie (2020) showed that focusing on some “good” hyperspheres would be beneficial rather than over all hyperspheres. In the calculation of attentions, we did not adjust $\delta$ so as to have a large weight for one specific hypersphere. This may cause limited improvements. We will compare mSVDD with DMSVDD later.

As the choice of $m$, the number of hyperspheres of mSVDD for different situations, it would be beneficial to use multiple hyperspheres if we have prior knowledge about the target class such as the existence of subtopics as shown in Fig. 3. Besides the prior information, we can also use the information in training process, e.g., the number of close points of each hypersphere, to determine whether some ones should be combined or be discarded (Ghafoori and Leckie 2020). Another method is cross validation, especially leave-one-out method, which is commonly used to automatically optimize the values for parameters (Stone 1974; Tax and Duin 2001). Considerably more work will need to be done to determine the right number of $m$.

![Fig. 3](image)

Fig. 3 t-SNE visualization of feature representations learned by DSVDD, mSVDD and CVDD over rec on NewsGroup dataset
5.2.2 Comparison between mSVDD and other models

Table 2 shows the results of the comparison between mSVDD and other models. From the discussion in the last subsection, we used \( m = 3 \) for mSVDD in this subsection. As for the three baseline methods, kernel OC-SVM, SVDD, and CVDD, we adopted their best scores on Reuters and 20 Newsgroup from (Ruff et al. 2019). As for TREC, we conducted experiments with the same setting with (Ruff et al. 2019) and reported the best results for the above three approaches. For the method of DMSVDD, we reported the results in the setting of the initial number of spheres \( K_{\text{init}} = 10 \). We also ran experiments with one-class neural networks (OC-NN) (Chalapathy et al. 2018). Since OC-NN\(^{10} \) was originally designed and applied for computer vision tasks, we re-implemented it to match our experiments with PyTorch (Paszke et al. 2019).

From the results in Table 2, we have the following observations. In general, CVDD performed better than other neural models on Reuters dataset, while mSVDD achieved better performances than other models on other two datasets, Newsgroup and TREC. As for the two traditional kernel methods, they performed well on Reuters, especially for earn and acq. As regard the comparison between DMSVDD and mSVDD, DMSVDD puts value on one hypersphere and performed slightly better over mSVDD(3) in some cases (e.g., earn, acq and comp). This indicates one inspiration that discarding “bad” hyperspheres is sometimes necessary. Regarding OC-NN, another unimodal method, mSVDD performed better than it in most target classes, while OC-NN obtained high scores in the cases of ABBR and pol. Therefore, OC-NN and its multi-modal version deserve to be explored and discussed in-depth in the future. Even mSVDD did not obtain the best AUC scores on Reuters, it still attained the second best in three cases. And the performances on the most cases for other two datasets showed the advantage of the proposed mSVDD over other related baseline models.

| Model            | Reuters target class |          | 20 Newsgroup target class |          | TREC target class |          |
|------------------|----------------------|----------|---------------------------|----------|------------------|----------|
|                  | earn acq             | crude    | trade money interest      | ship     | comp             | rev      | src      | make pol | rel      | ENTY    | HUM     | DESC    | NUM     | LOC     | ABBR    |
| OC-SVM/SVDD      | 91.1 93.1 92.4 92.0 88.6 87.4 98.4 92.0 82.0 75.6 64.1 63.1 75.5 79.2 | 66.6 71.5 52.0 67.7 74.9 93.1 | 91.1 93.1 92.4 92.0 88.6 87.4 98.4 92.0 82.0 75.6 64.1 63.1 75.5 79.2 | 66.6 71.5 52.0 67.7 74.9 93.1 | 91.1 93.1 92.4 92.0 88.6 87.4 98.4 92.0 82.0 75.6 64.1 63.1 75.5 79.2 | 66.6 71.5 52.0 67.7 74.9 93.1 |
| OC-NN            | 94.5 90.0 91.6 95.2 99.2 92.8 82.8 97.7 97.6 | 70.9 73.3 75.6 75.1 75.3 82.1 | 71.1 73.9 74.0 73.8 79.0 97.2 | 60.4 67.1 65.4 64.8 82.0 | 94.5 90.0 91.6 95.2 99.2 92.8 82.8 97.7 97.6 | 70.9 73.3 75.6 75.1 75.3 82.1 |
| CVDD             | 94.8 91.5 95.5 99.2 92.8 82.8 97.7 97.6 | 70.9 73.3 75.6 75.1 75.3 82.1 | 71.1 73.9 74.0 73.8 79.0 97.2 | 60.4 67.1 65.4 64.8 82.0 | 94.8 91.5 95.5 99.2 92.8 82.8 97.7 97.6 | 70.9 73.3 75.6 75.1 75.3 82.1 |
| DMSVDD           | 96.0 89.8 92.1 98.8 87.1 97.2 93.0 86.3 77.1 66.8 75.3 78.5 82.0 | 70.6 73.2 53.6 73.0 78.2 96.0 | 96.0 89.8 92.1 98.8 87.1 97.2 93.0 86.3 77.1 66.8 75.3 78.5 82.0 | 70.6 73.2 53.6 73.0 78.2 96.0 | 96.0 89.8 92.1 98.8 87.1 97.2 93.0 86.3 77.1 66.8 75.3 78.5 82.0 | 70.6 73.2 53.6 73.0 78.2 96.0 |
| mSVDD\(_{\text{One}}\) | 95.5 90.0 97.5 98.3 85.1 97.2 91.5 94.2 77.7 67.3 76.0 78.5 83.1 | 72.4 73.1 52.9 74.5 79.7 96.6 | 95.5 90.0 97.5 98.3 85.1 97.2 91.5 94.2 77.7 67.3 76.0 78.5 83.1 | 72.4 73.1 52.9 74.5 79.7 96.6 | 95.5 90.0 97.5 98.3 85.1 97.2 91.5 94.2 77.7 67.3 76.0 78.5 83.1 | 72.4 73.1 52.9 74.5 79.7 96.6 |
| mSVDD\(_{\text{Soft}}\) | 95.9 89.1 91.5 98.9 86.2 96.9 91.6 96.8 77.0 67.3 76.2 78.7 52.5 | 74.4 73.1 52.9 74.5 79.7 96.6 | 95.9 89.1 91.5 98.9 86.2 96.9 91.6 96.8 77.0 67.3 76.2 78.7 52.5 | 74.4 73.1 52.9 74.5 79.7 96.6 | 95.9 89.1 91.5 98.9 86.2 96.9 91.6 96.8 77.0 67.3 76.2 78.7 52.5 | 74.4 73.1 52.9 74.5 79.7 96.6 |

Table 2 Results of mSVDD and other models. Average AUCs in % on the Reuters (left part), 20 Newsgroup (middle part), and TREC (right part). One and Soft mean One-class and Soft-boundary forms, respectively. Best scores in each column are presented in bold, while the second best are underlined.

\(^{10}\) https://github.com/raghavchalapathy/oc-nn

1072
5.2.3 Results of mSVDD with negative supervision

Table 3 shows the performance of mSVDD trained with negative supervision. As mentioned, we still used m = 3 in this subsection. To perform negative supervision for mSVDD, we evaluated four approaches where different losses and their reformulated probability forms were selected.

For Reuters, the results indicate that mSVDD benefited from the joint training of the discrimination losses, except for acq and ship where negative supervision did not show consistent improvements (three of four methods). We can see more obvious improvements for 20 Newsgroup. All four negative supervision methods improved mSVDD markedly for all target classes of 20 Newsgroup. For example, mSVDD with negative supervision gained 2–3 points for comp.

For different losses for negative supervision, the contrastive type loss, which has larger punishment over negative data, performs better than the triplet type loss, which uses a relatively small margin. In addition, the performance of Con+Max was greater than the Con+Mean strategy to reformulate the probability. We hypothesize that focusing on one of the hyperspheres is effective when we used mSVDD with the contrastive loss.

As for the third TREC dataset, mSVDD with negative supervision achieved the best results on all target classes. The greatest gain came from DESC when the one-class form mSVDD trained with contrastive type loss with mean probability strategy. As for the probability strategy on TREC, max strategy would be beneficial to the one-class form, while max for soft-boundary mSVDD. In summary, Table 3 validated that mSVDD can be further improved by incorporating the proposed negative supervision methods.

| Model       | mSVDD.One | Triple+Max | Triple+Mean | Con+Max | Con+Mean | mSVDD.Soft | Triple+Max | Triple+Mean | Con+Max | Con+Mean |
|-------------|-----------|------------|-------------|---------|----------|------------|------------|-------------|---------|----------|
|             | +         | +          |             |         |          | +          | +          |             |         |          |
| mSVDD_One   | 95.5+     | 90.0       | 92.3+       | 99.5+   | 98.4+    | 93.8+      | 96.2+      | 77.7+       | 75.3+   | 76.4+    |
|             | 90.0       | 94.5+      | 98.4+       | 99.5+   | 98.4+    | 93.8+      | 96.2+      | 77.7+       | 75.3+   | 76.4+    |
|             | +Triplet+Max | 96.9       | 89.4       | 93.8     | 99.6     | 89.0       | 98.4       | 92.7        |         |          |
|             | +Triplet+Mean | 97.1     | 89.9       | 93.9     | 99.5     | 89.5       | 98.3       | 92.3        |         |          |
|             | +Con+Max    | 97.2       | 90.8       | 92.9     | 98.8     | 91.3       | 97.8       | 92.3        |         |          |
|             | +Con+Mean   | 97.2       | 90.8       | 92.9     | 98.8     | 91.3       | 97.8       | 92.3        |         |          |
| mSVDD_Soft  | 95.3+     | 89.1       | 91.5+      | 99.4+   | 86.1+    | 96.8+      | 86.1+      | 77.0+       | 67.3+   | 76.2+    |
|             | 97.1       | 89.6       | 92.9       | 99.4     | 89.3     | 97.4       | 92.6        |             |         |          |
|             | +Triplet+Max | 97.1     | 89.6       | 92.9     | 99.4     | 89.3       | 97.4       | 92.6        |         |          |
|             | +Triplet+Mean | 97.2     | 90.1       | 92.4     | 99.3     | 91.0       | 98.0       | 92.7        |         |          |
|             | +Con+Max    | 97.2       | 88.2       | 93.1     | 99.2     | 91.2       | 98.4       | 91.6        |         |          |
|             | +Con+Mean   | 97.2       | 88.1       | 90.8     | 91.3     | 98.4       | 90.4        |             |         |          |

Table 3 Results of mSVDD with negative supervision. Average AUCs in % on the Reuters (left part), 20 Newsgroup (middle part), and TREC (right part). +Triple+Max, which denotes mSVDD with Triplet loss with Max probability strategy, followed by three other negative supervision methods. In the rows of mSVDD_Soft and mSVDD_One, ‘+’ following numbers means that there were improvements with negative supervision (three of four methods.) Best scores in each column are presented in bold within each One and Soft group.
5.2.4 Results of CVDD with negative supervision

Table 4 shows the results of CVDD with the proposed negative supervision for mSVDD. As mentioned in Section 3.3, CVDD can be seen as a special case of mSVDD. Therefore, the proposed negative supervision approaches to mSVDD can be also applied to CVDD theoretically. To highlight the usefulness of the negative supervision, we conducted the experiments to use the triplet loss with \textit{Max} probability for CVDD. As for the implementation, since CVDD uses a different multi-head structure, we also used a different form to incorporate \textit{Triplet+Max} to CVDD (See Section B in the appendix for the details of the implementation.).

Overall, we can see that the proposed negative supervision enhanced CVDD in most cases on

| Model (r) | \textit{Reuters} target class | \textit{20 Newsgroup} target class | TREC target class |
|-----------|-------------------------------|-----------------------------------|------------------|
|           | \textit{earn} \textit{acq} \textit{crude} \textit{trade} \textit{money} \textit{interest} \textit{ship} \textit{comp} \textit{rec} \textit{sci} \textit{misc} \textit{pol} \textit{rel} \textit{ENTY} \textit{HUM} \textit{DESC} \textit{NUM} \textit{LOC} \textit{ABBR} |
| CVDD(3)   | 94.0 90.2 89.6 98.3 82.5 92.3 97.6 | 70.9 50.8 56.7 75.1 62.9 76.3 | 58.6 55.4 76.1 60.5 62.1 74.2 |
| +Triplet+Max | 96.1 90.2 97.3 98.3 84.2 92.4 91.8 | 74.5 64.2 61.0 75.1 62.2 72.5 | 68.3 70.7 81.1 74.1 71.6 86.7 |
| CVDD(5)   | 92.8 88.7 92.5 98.2 76.7 91.7 96.9 | 66.4 52.8 56.8 70.2 65.3 72.9 | 60.4 62.0 54.2 65.4 63.8 82.0 |
| +Triplet+Max | 94.0 94.4 96.7 98.7 84.0 97.3 92.5 | 73.2 64.5 58.4 76.2 63.6 76.1 | 65.6 59.7 82.4 73.4 67.7 88.7 |
| CVDD(10)  | 91.8 91.5 95.5 99.2 82.8 97.7 95.6 | 63.3 53.3 55.7 68.6 65.1 70.7 | 60.0 56.1 69.8 65.2 59.7 80.7 |
| +Triplet+Max | 93.0 91.2 97.4 99.6 85.7 98.7 94.2 | 78.3 69.7 60.5 73.3 67.5 79.1 | 72.0 69.8 80.4 69.9 81.0 88.1 |

Table 4 Results of CVDD with the proposed negative supervision. Average AUCs in \% on the \textit{Reuters} (upper part), \textit{20 Newsgroup} (middle part), and \textit{TREC} (lower part) datasets. Number \textit{r} in brackets denotes the number of heads in CVDD. \textbf{Bold} means the better AUCs score.
the three datasets. The overall performance mainly shows the following: 1) The improvement by
the negative supervision to CVDD is consistent with mSVDD due to the similarity between the
two. 2) The generality of the negative supervision can be shown, as Triple+Max was successfully
applied to the different multi-head structure.

Regarding different target-classes, ship with the smaller training data size may cause worse
performance, so does real with CVDD(3), which are similar phenomena with mSVDD. While for
TREC, almost all the target classes can be improved by using negative supervision. In addition,
the negative supervision also prevented over-fitting for CVDD. For example, CVDD(3) with the
minimal parameters achieved the best score for comp when varying “r” among 3, 5 and 10.
In contrast, when the negative supervision was used, CVDD(10) with the maximal parameters
attained the best and also performed better for all six target classes of the 20 Newsgroup dataset.

5.2.5 Negative supervision using word replacement with TF-IDF

In the former parts, we have introduced that mSVDD can be further improved by incorporat-
ing negative supervision. And negative supervision was trained on pseudo-negative samples that
were randomly sampled sentences from a certain publicly available corpus, WikiText-2 (Merity
et al. 2016). While this sentence-level sampling strategy can provide a wealth of out-of-domain
samples with diverse semantics, it is not easy to control the generation of samples and the adapt-
ability of diverse target classes settings over different datasets.

Thus, we provide another method for pseudo-negative construction without the use of out-of-
domain corpus. This idea was motivated by the recent remarkable data augmentation techniques,
especially by unsupervised data augmentation method (Xie et al. 2019). In general, we propose a
word-level data augmentation method to construct pseudo-negative samples from the only given
the positive target training dataset.

Our method is very simple. First, we assume that: 1) The pseudo-negative samples should
have more common words to target samples than samples from external corpus in the one-class
classification setting. 2) While pseudo-negative samples constructed should not contain words
that are more informative to the instances from the target class. Therefore, we proposed a method
that replaces informative words with high TF-IDF scores while keeping those with low values.
Now, for a given target training dataset, this approach could construct pseudo-negative samples
on the basis of itself, rather than any other external corpus.

Formally, for a word w in one target training text sequence, we compute its TF-IDF value
as \( \text{TFIDF}(w) = \text{TF}(w) \times \text{IDF}(w) \), where TF(w) is the term frequency score for word w in this
sequence, and IDF(w) denotes the inverse document frequency calculated on the only target
training dataset. Specifically, we sample words to be replaced with weights according to the TF-IDF value, i.e., sampling with weights $p(w) \propto \text{TFIDF}(w)/Z$, where $Z$ is the sum of TF-IDF scores in one sequence. When a word is replaced, we randomly sample one word from the whole vocabulary for the replacement. Once completed the replacement for one input sequence, we can get one corresponding pseudo-negative sample. In the training, for one batch of positive training samples, we randomly sampled its pseudo-negative sample from the generated augmented set. Then, same as before, the constructed word-level pseudo-negative samples were used for negative supervision.

Table 5 shows the results of mSVDD with negative supervision using word replacement with TF-IDF. In general, it shows that the proposed negative supervision still works with the pseudo-negative samples generated by the method of TFIDF replacement. We can see that mSVDD obtained improvements in most cases over the three datasets, as the same with the use of external dataset shown in Table 3. Besides, for acq and ship on Reuters, mSVDD still did not benefit from negative supervision. And ABBR also did not show consistent improvements for four approaches. While differently, there were more obvious variations for four negative supervision approaches, especially on the the former two datasets. Con+Mean strategy performed better than other three methods on ten target classes, except for ship where this strategy degraded the performance of mSVDD. As for TREC dataset, max strategy would be the best choice for the triplet loss type which beats other combinations on HUM and LOC. While mean strategy would be still beneficial to constrastive type loss, as with other two datasets.

5.2.6 Affects of parameter $\gamma$

Parameter $\gamma$ in Eq. (22) controls the trade-off between the intra-class loss mSVDD optimized and the discrimination loss negative supervision learned. In the former experiments on negative supervision using TFIDF word-level replacement method, we fixed $\gamma = 1$. Here, we would like to conduct experiments with varying $\gamma$ settings to verify how the trade-off affects the overall performance. There are four negative supervision approaches that can be jointly trained with mSVDD. As shown in Table 5, other three choices outperformed Triple+Max in many cases. Thus we chose Triple+Max for our sensitivity analysis experiments here. We expected Triple+Max can obtain improvements through the adjustment of $\gamma$.

Fig. 4 illustrates the relationship of $\gamma$ and the evaluation of mSVDD+Triple+Max on different target classes over three datasets. As for the two versions of mSVDD on Reuters, Figs. 4a and 4d show that a larger $\gamma$ (10 or 1) would be better setting on the former three target classes, especially on acq. While on trade or money-fx, mSVDD achieved better performances with
Table 5 Results of mSVDD with negative supervision using TFIDF replacement method. Settings are the same with Table 3.

Fig. 4 Results of mSVDD using the method of Triple+Max with parameter $\gamma$ varying from 0.1, 1 to 10.

smaller value of $\gamma = 0.1$. Figs. 4b and 4e indicate that when $\gamma$ is set to 1 or 10, mSVDD can become slightly better for most target classes of Newsgroup dataset. While for rel with smaller data size, mSVDD became worse by focusing more on negative supervision. We can observe a clear relationship between the increases of $\gamma$ and the improvements of mSVDD from Figs. 4c and 4f. These two subfigures show that mSVDD gained steady improvements over most target classes on TREC for both mSVDD forms. Besides, the verification performance of the negative supervision remains stable across different settings of $\gamma$ over a wide range of target class settings.
6 Related Work

One-class learning The first related area would be one-class learning or classification. Generally, one-class classification approaches can be divided into the following groups: the density estimation-based methods, the boundary-based methods, and the reconstruction-based methods (Tax 2001; Khan and Madden 2009, 2014). Density estimation-based methods, including early work uses different probability models (Parzen 1962; Härdle 1990; Kim and Scott 2012), estimate the density of the training data and sets the threshold for the target of high probabilities in the distribution. Reconstruction-based methods, including AutoEncoder (Rumelhart et al. 1985) and principal component analysis (PCA) (Pearson 1901), attempt to learn a more compact representation for the description of target data. Most of these methods learn a model that is optimized to reduce the reconstruction error of normal data samples. CVDD (Ruff et al. 2019), that optimizes the one-to-one reconstruction distance between context vectors and self-attention feature vectors, can be viewed as a kind of Reconstruction-based approach. Boundary-based methods learn only a boundary around the target set. As introduced in Section 2, OC-SVM seeks a hyperplane to separate the data from the origin, while SVDD finds a hypersphere to enclose the data. Their neural extensions, DSVDD (Ruff et al. 2018) and OC-NN (Chalapathy et al. 2018), can also optimize the related boundaries in the neural feature space.

Metric learning The incorporation of negative supervision in Section 4 is related to metric learning (Guillaumin et al. 2009). Specifically, in our proposed mSVDD with the negative supervision framework, pseudo-negative samples are used to calculate the contrastive and triplet type losses. Through adding these losses to mSVDD, mSVDD can learn discriminative features that keep target data points close, while pushing pseudo-negative data apart in the feature space. This shares a similar concept with metric learning, such as ε-SVM, which minimizes the within-class distances by adding constraints to SVM (Do et al. 2012). In terms of using neural networks to learn features, our methods are more relevant to deep metric learning (Hoffer and Ailon 2015; Kaya and Bilge 2019). Different from Mahalanobis metric learning (Weinberger and Saul 2009), that finds only a linear transform, recent work on deep metric learning can learn non-linear neural feature mapping, such that the distance of a positive pair is smaller than a negative one (Hu et al. 2014; Wu et al. 2017; Sun et al. 2020). Formally, deep metric learning methods usually define a distance between data points as \( d(x_i, x_j) = \|f(x_i) - f(x_j)\| \), where \( f(x) \) is an embedding of a sample \( x \) learned by a neural network. Given three data points \( a, p, n \), where \( a \) is an anchor, \( p \) is a positive of the same class as the anchor, and \( n \) is a negative of a different class, the triplet loss that the deep metric learning optimized is: \( \mathcal{L} = \max\{0, d(a, p) - d(a, n) + \text{margin}\} \). If we
assume that the *centers* in mSVDD are anchors, data from the target class are positive, and external samples are negative, the loss of mSVDD with negative supervision can be written in similar formulation.

**Other related work** The *pseudo*-negative samples used in our negative supervision are from publicly available corpus, WikiText-2. We then use these external data to complete auxiliary tasks with contrastive and triplet losses. This can be seen as one kind of *weakly supervision* method, which learns classification models from data that are incompletely or scarcely labeled (Lee 2013; Zhou 2018). Since the discriminative information is learned from external data that is not directly related to the original training data in our negative supervision, this also follows the idea from *transfer learning* (Pan and Yang 2010). Perera and Patel (2019) proposed a much more related work, where they also incorporated the external image dataset to learn discriminative features for a one-class problem in the computer vision area. As for the difference between our work (Perera and Patel 2019), the primary purpose of (Perera and Patel 2019) is to learn features for one-class classification with two stages setting: feature learning and *k*-nearest-neighbor classifier usage. In contrast, our purpose is to design a specific structure for mSVDD such that it can be trained in an end-to-end neural structure. Another related area is *self-supervised learning*, that constructs kinds of *contrastive learning* tasks for unsupervised representation learning (Khosla et al. 2020; Ohashi et al. 2020; Chen et al. 2020). For example, in the contrastive learning framework used in Momentum Contrast (MoCo) (He et al. 2020), they learned similar/dissimilar representations from data that are grouped into pairs by optimizing a so-called InfoNCE loss function (Oord et al. 2018). The contrastive loss (Eq. (19)) used in our work can be viewed as a simplified InfoNCE loss function.

7 Conclusion

In this work, we proposed mSVDD, a new generic one-class text classification framework that uses multi-modal deep SVDD. Rather than the uni-modal deep SVDD, mSVDD can enhance the description ability to the target one-class data with multiple hyperspheres. We also proved that this generic framework can include four variants, deep SVDD, DMSVDD, OC-NN, and CVDD under certain conditions. In addition, in the absence of “real” negative training data, we also proposed approaches for effectively adding negative supervision to further improve the performance of one-class text classification methods. The experiments validated that the proposed mSVDD provides better performance compared to uni-modal SVDD. The experiments also showed the further improvements in most cases when negative supervision was used for mSVDD and CVDD.
Acknowledgement

The authors would like to gratefully acknowledge the reviewers for their time and their valuable comments. Hu thanks the support from China Scholarship Council. This paper is an extended version of the paper (Hu et al. 2021) published at the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021). The extended version introduces more baseline models in Section 2, discusses their relationships with other models in Section 3, and shows the evaluation results in Section 5. More experiments and discussions on the incorporation of negative supervision are also included. Section 6 is newly added for the summary of related work.

References

Alashwal, H., Deris, S., and Othman, R. M. (2006). “One-class Support Vector Machines for Protein-protein Interactions Prediction.” *International Journal of Biological and Medical Sciences*, 1 (2), pp. 120–127.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). “A Neural Probabilistic Language Model.” *Journal of Machine Learning Research*, 3 (Feb), pp. 1137–1155.

Chalapathy, R., Menon, A. K., and Chawla, S. (2018). “Anomaly Detection using One-Class Neural Networks.” *CoRR*, abs/1802.06360.

Chandola, V., Banerjee, A., and Kumar, V. (2010). “Anomaly Detection for Discrete Sequences: A Survey.” *IEEE Transactions on Knowledge and Data Engineering*, 24 (5), pp. 823–839.

Chen, P.-H., Lin, C.-J., and Schölkopf, B. (2005). “A Tutorial on ν-Support Vector Machines.” *Applied Stochastic Models in Business and Industry*, 21 (2), pp. 111–136.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). “A Simple Framework for Contrastive Learning of Visual Representations.” In *Proceeding of the 37th International Conference on Machine Learning*, pp. 1597–1607.

Das, S., Matthews, B. L., Srivastava, A. N., and Oza, N. C. (2010). “Multiple Kernel Learning for Heterogeneous Anomaly Detection: Algorithm and Aviation Safety Case Study.” In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 47–56.

Deng, J., Guo, J., Xue, N., and Zafeiriou, S. (2019). “Arcface: Additive Angular Margin Loss for Deep Face Recognition.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699.
Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186.

Do, H., Kalousis, A., Wang, J., and Woznica, A. (2012). “A Metric Learning Perspective of SVM: on the Relation of LMNN and SVM.” In Artificial Intelligence and Statistics, pp. 308–317.

Dumais, S., Platt, J., Heckerman, D., and Sahami, M. (1998). “Inductive Learning Algorithms and Representations for Text Categorization.” In Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management (Bethesda, MD, 1998), pp. 148–155.

Ghafoori, Z. and Leckie, C. (2020). “Deep Multi-sphere Support Vector Data Description.” In Proceedings of the 2020 SIAM International Conference on Data Mining, pp. 109–117. SIAM.

Guillaumin, M., Verbeek, J., and Schmid, C. (2009). “Is That You? Metric Learning Approaches for Face Identification.” In 2009 IEEE 12th International Conference on Computer Vision, pp. 498–505. IEEE.

Gupta, M., Gao, J., Aggarwal, C. C., and Han, J. (2013). “Outlier Detection for Temporal Data: A Survey.” IEEE Transactions on Knowledge and Data Engineering, 26 (9), pp. 2250–2267.

Hadsell, R., Chopra, S., and LeCun, Y. (2006). “Dimensionality Reduction by Learning an Invariant Mapping.” In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), Vol. 2, pp. 1735–1742. IEEE.

Hao, P.-Y. and Lin, Y.-H. (2007). “A New Multi-class Support Vector Machine with Multi-sphere in the Feature Space.” In Okuno, H. G. and Ali, M. (Eds.), New Trends in Applied Artificial Intelligence, pp. 756–765, Berlin, Heidelberg. Springer Berlin Heidelberg.

Härdle, W. (1990). Applied Nonparametric Regression. No. 19. Cambridge University Press.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). “Momentum Contrast for Unsupervised Visual Representation Learning.” In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738.

Heron, S. (2009). “Technologies for Spam Detection.” Network Security, 2009 (1), pp. 11–15.

Hinton, G. E. and Salakhutdinov, R. R. (2006). “Reducing the Dimensionality of Data with Neural Networks.” Science, 313 (5786), pp. 504–507.

Hochreiter, S. and Schmidhuber, J. (1997). “Long Short-term Memory.” Neural Computation, 9 (8), pp. 1735–1780.

Hoffer, E. and Ailon, N. (2015). “Deep Metric Learning using Triplet Network.” In International
Workshop on Similarity-based Pattern Recognition, pp. 84–92. Springer.

Hoffmann, H. (2007). “Kernel PCA for Novelty Detection.” Pattern Recognition, 40 (3), pp. 863–874.

Hotelling, H. (1933). “Analysis of a Complex of Statistical Variables into Principal Components.” Journal of Educational Psychology, 24 (6), p. 417.

Hu, C., Feng, Y., Kamigaito, H., Takamura, H., and Okumura, M. (2021). “One-class Text Classification with Multi-modal Deep Support Vector Data Description.” In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 3378–3390, Online. Association for Computational Linguistics.

Hu, J., Lu, J., Yuan, J., and Tan, Y.-P. (2014). “Large Margin Multi-metric Learning for Face and Kinship Verification in the Wild.” In Asian Conference on Computer Vision, pp. 252–267. Springer.

Jacobs, R. A. (1995). “Methods for Combining Experts’ Probability Assessments.” Neural Computation, 7 (5), pp. 867–888.

Kaya, M. and Bilge, H. Ş. (2019). “Deep Metric Learning: A Survey.” Symmetry, 11 (9), p. 1066.

Khan, S. S. and Madden, M. G. (2009). “A Survey of Recent Trends in One Class Classification.” In Irish Conference on Artificial Intelligence and Cognitive Science, pp. 188–197. Springer.

Khan, S. S. and Madden, M. G. (2014). “One-class Classification: Taxonomy of Study and Review of Techniques.” The Knowledge Engineering Review, 29 (3), pp. 345–374.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. (2020). “Supervised Contrastive Learning.” arXiv preprint arXiv:2004.11362.

Kim, J. and Scott, C. D. (2012). “Robust Kernel Density Estimation.” The Journal of Machine Learning Research, 13 (1), pp. 2529–2565.

Kingma, D. P. and Ba, J. (2014). “Adam: A Method for Stochastic Optimization.” arXiv preprint arXiv:1412.6980.

Kumaraswamy, R., Wazalwar, A., Khot, T., Shavlik, J., and Natarajan, S. (2015). “Anomaly Detection in Text: The Value of Domain Knowledge.” In the 28th International Flairs Conference, pp. 225–228.

Le, T., Tran, D., and Ma, W. (2013). “Fuzzy Multi-sphere Support Vector Data Description.” In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 570–581. Springer.

Le, T., Tran, D., Ma, W., and Sharma, D. (2010). “A Theoretical Framework for Multi-sphere Support Vector Data Description.” In International Conference on Neural Information Processing, pp. 132–142. Springer.

Lee, D.-H. (2013). “Pseudo-label: The Simple and Efficient Semi-supervised Learning Method
for Deep Neural Networks.” In Workshop on Challenges in Representation Learning, ICML, Vol. 3, p. 896.

Li, X. and Roth, D. (2002). “Learning Question Classifiers.” In COLING 2002: The 19th International Conference on Computational Linguistics, pp. 1–7.

Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B., and Bengio, Y. (2017). “A Structured Self-attentive Sentence Embedding.” arXiv preprint arXiv:1703.03130.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. (2017). “Sphereface: Deep Hypersphere Embedding for Face Recognition.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 212–220.

Lloyd, S. (1982). “Least Squares Quantization in PCM.” IEEE Transactions on Information Theory, 28 (2), pp. 129–137.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). “Effective Approaches to Attention-based Neural Machine Translation.” arXiv preprint arXiv:1508.04025.

Manevitz, L. and Yousef, M. (2007). “One-class Document Classification via Neural Networks.” Neurocomputing, 70 (7-9), pp. 1466–1481.

Manevitz, L. M. and Yousef, M. (2001). “One-class SVMs for Document Classification.” Journal of Machine Learning Research, 2 (Dec), pp. 139–154.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. (2016). “Pointer Sentinel Mixture Models.” arXiv preprint arXiv:1609.07843.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). “Distributed Representations of Words and Phrases and Their Compositionality.” In Advances in Neural Information Processing Systems, pp. 3111–3119.

Moya, M. M., Koch, M. W., and Hostetler, L. D. (1993). “One-class Classifier Networks for Target Recognition Applications.” NASA STI/Recon Technical Report N, 93, pp. 797–801.

Ng, A. Y. (2004). “Feature selection, L 1 vs. L 2 Regularization, and Rotational Invariance.” In Proceedings of the 21st International Conference on Machine Learning, p. 78.

Ohashi, S., Takayama, J., Kajiwara, T., Chu, C., and Arase, Y. (2020). “Text Classification with Negative Supervision.” In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 351–357.

Oja, E. (1982). “Simplified Neuron Model as a Principal Component Analyzer.” Journal of Mathematical Biology, 15 (3), pp. 267–273.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). “Representation Learning with Contrastive Predictive Coding.” arXiv preprint arXiv:1807.03748.

Pal, M. and Foody, G. M. (2010). “Feature Selection for Classification of Hyperspectral Data by
SVM.” *IEEE Transactions on Geoscience and Remote Sensing*, **48** (5), pp. 2297–2307.

Pan, S. J. and Yang, Q. (2010). “A Survey on Transfer Learning.” *IEEE Transactions on Knowledge and Data Engineering*, **22** (10), pp. 1345–1359.

Parzen, E. (1962). “On Estimation of a Probability Density Function and Mode.” *The Annals of Mathematical Statistics*, **33** (3), pp. 1065–1076.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). “PyTorch: An Imperative Style, High-Performance Deep Learning Library.” In Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., and Garnett, R. (Eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc.

Pearson, K. (1901). “LIII. On Lines and Planes of Closest Fit to Systems of Points in Space.” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, **2** (11), pp. 559–572.

Pennington, J., Socher, R., and Manning, C. (2014). “Glove: Global Vectors for Word Representation.” In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Perera, P. and Patel, V. M. (2019). “Learning Deep Features for One-class Classification.” *IEEE Transactions on Image Processing*, **28** (11), pp. 5450–5463.

Roberts, S. J. (1999). “Novelty Detection using Extreme Value Statistics.” *IEE Proceedings-Vision, Image and Signal Processing*, **146** (3), pp. 124–129.

Rodner, E., Wacker, E.-S., Kemmler, M., and Denzler, J. (2011). “One-class Classification for Anomaly Detection in Wire Ropes with Gaussian Processes in a Few Lines of Code.” *Training*, **1**, pp. 1–5.

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., and Kloft, M. (2018). “Deep one-class Classification.” In *International Conference on Machine Learning*, pp. 4393–4402.

Ruff, L., Zemlyanskiy, Y., Vandermeulen, R., Schnake, T., and Kloft, M. (2019). “Self-Attentive, Multi-Context One-Class Classification for Unsupervised Anomaly Detection on Text.” In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4061–4071.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). “Learning Internal Representations by Error Propagation.” Tech. rep., California Univ., San Diego, La Jolla, Inst. for Cognitive
Hu et al. One-class Text Classification with Multi-modal Deep Support Vector Data Description

Science.
Saitoh, S. (2003). Theory of Reproducing Kernels, pp. 135–150. Springer US, Boston, MA.
Schölkopf, B. and Smola, A. (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, USA.
Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). “Estimating the Support of a High-dimensional Distribution.” Neural Computation, 13 (7), pp. 1443–1471.
Schölkopf, B., Smola, A. J., Williamson, R. C., and Bartlett, P. L. (2000). “New Support Vector Algorithms.” Neural Computation, 12 (5), pp. 1207–1245.
Schroff, F., Kalenichenko, D., and Philbin, J. (2015). “Facenet: A Unified Embedding for Face Recognition and Clustering.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823.
Stone, M. (1974). “Cross-validatory Choice and Assessment of Statistical Predictions.” Journal of the Royal Statistical Society: Series B (Methodological), 36 (2), pp. 111–133.
Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., and Wei, Y. (2020). “Circle Loss: A Unified Perspective of Pair Similarity Optimization.” In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
Tax, D. M. J. (2001). One-class Classification: Concept Learning in the Absence of Counterexamples. Ph.D. thesis, Technische Universiteit Delft.
Tax, D. M. and Duin, R. P. (2001). “Uniform Object Generation for Optimizing One-class Classifiers.” Journal of Machine Learning Research, 2 (Dec), pp. 155–173.
Tax, D. M. and Duin, R. P. (2004). “Support Vector Data Description.” Machine Learning, 54 (1), pp. 45–66.
Tsybakov, A. B. et al. (1997). “On Nonparametric Estimation of Density Level Sets.” The Annals of Statistics, 25 (3), pp. 948–969.
Van der Maaten, L. and Hinton, G. (2008). “Visualizing Data using t-SNE.” Journal of Machine Learning Research, 9 (11), pp. 2597–2605.
Vert, R., Vert, J.-P., and Schölkopf, B. (2006). “Consistency and Convergence Rates of One-Class SVMs and Related Algorithms.” Journal of Machine Learning Research, 7 (5), pp. 817–854.
Wang, F., Xiang, X., Cheng, J., and Yuille, A. L. (2017). “Normface: L2 Hypersphere Embedding for Face Verification.” In Proceedings of the 25th ACM International Conference on Multimedia, pp. 1041–1049.
Weinberger, K. Q. and Saul, L. K. (2009). “Distance Metric Learning for Large Margin Nearest
Neighbor Classification.” *Journal of Machine Learning Research*, **10** (2), pp. 207–244.

Wu, C.-Y., Manmatha, R., Smola, A. J., and Krahenbuhl, P. (2017). “Sampling Matters in Deep Embedding Learning.” In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2840–2848.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. (2018). “Unsupervised Feature Learning via Non-parametric Instance Discrimination.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742.

Xiao, Y., Liu, B., Cao, L., Wu, X., Zhang, C., Hao, Z., Yang, F., and Cao, J. (2009). “Multisphere Support Vector Data Description for Outliers Detection on Multi-distribution Data.” In *2009 IEEE international conference on data mining workshops*, pp. 82–87. IEEE.

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., and Le, Q. V. (2019). “Unsupervised Data Augmentation for Consistency Training.” *arXiv preprint arXiv:1904.12848*.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention.” In *International Conference on Machine Learning*, pp. 2048–2057.

Yokoi, S., Takahashi, R., Akama, R., Suzuki, J., and Inui, K. (2020). “Word Rotator’s Distance.” In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2944–2960.

Yu, H., Han, J., and Chang, K.-C. (2004). “PEBL: Web Page Classification without Negative Examples.” *IEEE Transactions on Knowledge and Data Engineering*, **16** (1), pp. 70–81.

Zhou, Z.-H. (2018). “A Brief Introduction to Weakly Supervised Learning.” *National Science Review*, **5** (1), pp. 44–53.

### Appendix

**A Proofs of Proposition 1**

**Proposition 1.** (\(\nu\)-property) The hyperparameter \(\nu \in (0, 1]\) in soft-boundary deep mSVDD holds if we set an equal attention weight to each hypersphere:

i. \(\nu\) is an upper bound on the fraction of outlier samples.

ii. \(\nu\) is a lower bound on the fraction of training samples being rejected or on the optimal boundary.

**Proof.** Ad (i). For each training instance \(x_i\), its loss function is defined as hinge-loss: \(l(f(x_i)) = \)

1086
max\{0, \sum_j \alpha_{ij}(\|f(x_i; W) - c_j\|^2 - R_j^2)\}, \text{ where } f \text{ is the model with parameters. Let us define } d_i = \sum_j \alpha_{ij}d(x_i, c_j). \text{ Assume } \alpha_{ij} = 1/m, \text{ we have } d_i = \frac{1}{m} \sum_j d(x_i, c_j). \text{ We also define } R_s = \frac{1}{m} \sum_j R_j^2. \text{ And W.L.O.G, we also assume } d_1 \leq \ldots \leq d_n \text{ which means } d_n \text{ is } n\text{-th farthest sum distance. The number of outliers is given by } n_{out} = |\{i|d_i > R_s\}|. \text{ Rewrite the objective of soft-boundary deep mSVDD (Eq. (7)) as: }

\[ L_{\text{soft-mSVDD}} = R_s - \frac{n_{out}}{\nu n} R_s = (1 - \frac{n_{out}}{\nu n}) R_s \tag{23} \]

Since the objective of mSVDD is to get a minimum } R_s, \text{ therefore } 1 - \frac{n_{out}}{\nu n} \text{ should be positive, Thus, } n_{out} \leq \nu n \text{ must hold in the training. It implies that at most } \nu n \text{ outliers should be rejected.}

**Ad (ii).** The optimal } R_s^* \text{ has to hold the inequality } n_{out} \leq \nu n. \text{ If } R_s^* \geq d_n, \text{ then } n_{out} \text{ takes the minimum value of 0 which means the boundary includes all the samples. Since } n_{out} \text{ is increased as long as } R_s \text{ decreased. If } n_{out} \text{ take the maximum value of } \nu n \text{ under condition (i), we can have the minimal } R_s^* = d_{i*}, \text{ where } i* = n - n_{out} \text{ means } d_{i*} \text{ is } (n - n_{out})\text{-th farthest distance. We define } \{x_i|d_i \geq R_s^*\} \text{ is the set of training samples being rejected } (d_i > R_s^*) \text{ or on the optimal boundary } (d_i = R_s^*). \text{ Then we have inequality: } |\{x_i|d_i \geq R_s^*\}| + |\{x_i|d_i = R_s^*\}| \geq n_{out} + 1 \geq \nu n. \text{ This implies that at least } \nu n \text{ samples being rejected or just on the optimal boundary.} \quad \square

**Proposition 1** and its proof refer to works (Ruff et al. 2018; Chen et al. 2005; Schölkopf et al. 2000).

### B Implementation of CVDD with Negative Supervision

CVDD uses a group of \( r \) context vectors \( C = (c_1, \ldots, c_r) \) to describe the target one-class data, where \( c_k \in \mathbb{R}^p \). Given one context vector \( c_k \), and a pair of training positive and negative samples, \((x_i, x_{i'})\), we can get the reformulated probability form. First, CVDD maps a training sample \( x_i \) to \( r \) heads of feature vectors \( S_i = (s_{i1}, \ldots, s_{ir}) \). Then, we denote \( p(y_i = 1|s_{ik}, c_k) \) as the probability that \( k\)-th \( s_{ik} \) reconstructs \( k\)-th context vector \( c_k \) well.

\[ p(y_i = 1|s_{ik}, c_k) = \sigma(s_{ik}^T c_k) \tag{24} \]

And with **triplet** and **Max** probability strategy, we can define the negative supervision loss as:

\[ L_{\text{Tri}}^{(i'i')} = [\log p(y_{i'} = 1|x_{i'}) - \log p(y_i = 1|x_i) + \tau]_+ \]

\[ = [\log \max_{k=1,\ldots,r} p(y_{i'} = 1|s_{ik}, c_k) - \log \max_{k=1,\ldots,r} p(y_i = 1|s_{ik}, c_k) + \tau]_+ \tag{25} \]
where \( \tau \) is a margin. Then, \( L^{(ii')}_{Tr1} \) can then be added to Eq. (11) to obtain the training loss.

**Chenlong Hu:** He received his Ph.D. from Tokyo Institute of Technology. Before that, he received M.E. from Xidian University, Xi’an. His research interests include natural language processing and machine learning.

**Yukun Feng:** He is currently a Ph.D. candidate at Department of Information and Communications Engineering, Tokyo Institute of Technology. Before that, he received M.E. from Tokyo Institute of Technology in 2020 and B.E. from Beijing Language and Culture University in 2016. His research interests include natural language processing and machine learning, particularly word representation learning and language modeling.

**Hidetaka Kamigaito:** He received his Ph.D. from Tokyo Institute of Technology, and is currently an assistant professor at Institute of Innovative Research, Tokyo Institute of Technology. His current research interests are in natural language processing with a specific focus on document-level text processing.

**Hiroya Takamura:** He received his Ph.D. from Nara Institute of Science and Technology. He worked as a professor at Tokyo Institute of Technology, and currently is a research team leader at AI Research Center of Advanced Industrial Science and Technology. His current research interests include natural language processing.

**Manabu Okumura:** Manabu Okumura was born in 1962. He received B.E., M.E. and Dr. Eng. from Tokyo Institute of Technology in 1984, 1986 and 1989 respectively. He was an assistant at the Department of Computer Science, Tokyo Institute of Technology from 1989 to 1992, and an associate professor at the School of Information Science, Japan Advanced Institute of Science and Technology from 1992 to 2000. He is currently a professor at Institute of Innovative Research, Tokyo Institute of Technology. His current research interests include natural language processing, especially text summarization, computer assisted language learning, sentiment analysis, and text data mining.

(Received March 1, 2021)
(Revised June 1, 2021)
(Accepted July 10, 2021)