A KERNEL-BASED SUPPORT TENSOR DATA DESCRIPTION FOR ONE-CLASS CLASSIFICATION

Xue WANG[1], Minghui WANG[2], Kuaini WANG[3], Yanyan CHEN[4]*

The issue of one-class classification has got a great deal of studies. However, the classical algorithms represented by Support Vector Data Description (SVDD) have restrictions when the input is not vector. Therefore, we present a nonlinear tensor-based data description that is named as Kernel-based Support Tensor Data Description (KSTDD). The basic thought of KSTDD is to seek for an enclosing hypersphere of smallest volume that contains most of target objects. KSTDD uses tensor as input, and it has the ability to keep more data topology. Meanwhile, the number of parameters that need to be estimated by KSTDD is reduced considerably, which makes KSTDD more fit for small-sample learning. KSTDD is iteratively solved, and the computation complexity and the convergence of the corresponding iterative algorithm are provided respectively. We prove that KSTDD is equivalent to One-Class Support Tensor Machine (OCSTM) for Gaussian-based kernel matrix. However, the two algorithms cannot be completely equivalent to each other since they are quite different for other kernel matrices. Therefore, we evaluate KSTDD with different kernel matrices including Gaussian-based kernel matrices and polynomial-based kernel matrices. Experiments have verified the efficiency of the KSTDD.

Keywords: Support Tensor Data Description; Support Vector Domain Description; Kernel matrix; One-class classification

1. Introduction

Inspired by support vector machine (SVM) [1], Tax et al. proposed a support vector domain description which seeks for an enclosing ball with the minimum volume to contain most target objects [2]. In recent years, SVDD has been used for many real-world applications [3,4], such as anomaly detection [5], face recognition [6], medical imaging [7] and other studies on SVDD [8,9].

In many fields, a lot of objects are described as tensors. When the input is tensor, the classical vector-based data description represented by SVDD can not work directly. Despite tensor can be converted to vector, it has been shown that

[1] College of Applied Science and Technology, Beijing Union University, Beijing 100101, China
[2] College of Applied Science and Technology, Beijing Union University, Beijing 100101, China
[3] College of Science, Xi'an Shiyou University, Xi'an 710065, China
[4]* Corresponding author: Yanyan Chen, Institute of Mathematics and Physics, Beijing Union University, Beijing 100101, China, e-mail: yykjtyanyan@buu.edu.cn

structural information of data may be lost, and at the same time, small sample size problem and curse of dimensionality may be occurred [10,11].

To solve the above problem, researches on data representation have attracted widespread interest in the past few years Wu et al. gave Supervised Tensor Learning (STL) framework [12]. Cai et al. extended SVM to fit for tensor representation, and the tensor model was used for text categorization and named as Support Tensor Machine (STM) [13]. By combining the advantages of C-SVM and rank one decomposition of tensor, a Support Higher-order Tensor Machine was established in [14]. As a matter of course, researchers were interested in extending the SVM algorithms to tensor space by establishing multilinear algorithms [15-19]. As for kernel methods for tensor, one can refer to references [20-22].

For handling one-class classification with tensor input, one of naturel ideas is to search a separating hyperplane which pursues a maximal margin between target class and the origin. This thought has been proposed as One-Class Support Tensor Machines (OCSTM) [23, 24]. Another thought is to find an enclosing hypersphere that contains most of target objects, we also have proposed a linear support tensor domain description [25] for higher order tensor.

Utilizing the advantages of support vector data domain and the STL framework, we derive an effective nonlinear tensor-based algorithm in this paper, named Kernel-based Support Tensor Data Description (KSTDD). We present KSTDD models with two clear and convenient kernel matrices. Compared with vector-based algorithms, KSTDD more fits for the small sample size problem due to the considerable reduction of parameters that need to be estimated. Therefore, we test KSTDD on several datasets that are truly small size and high dimensionality. We further prove that KSTDD proposed in this paper and OCSTM are equivalent for Gaussian-based kernel matrix. The later experiments also verified the fact. But that doesn't mean there is no necessary to study KSTDD, since it has different performance along with other kernel matrices.

2. Relevant Research

2.1 Support Vector Data Description

Denote R the radius of the hypersphere, and c the center, respectively. The aim of SVDD is to minimize the volume of hypersphere by minimizing R^2. Denote training data, $\mathbf{x}_i \in \mathbb{R}^n, i = 1,...,l$. SVDD gets the classifier by solving:

$$\begin{aligned} \min_{R,c,\xi} \quad & R^2 + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i \\ s.t. \quad & \left\|\phi(\mathbf{x}_i) - c\right\|^2 \le R^2 + \xi_i \\ & \xi_i \ge 0, i = 1,...,l \end{aligned} \tag{1}$$

where ν is a parameter which controls balance of the errors and the volume. The corresponding dual problem of (1) is:

$$\begin{aligned} \min_{\alpha} \quad & \sum_{i,j=1}^{l} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{l} \alpha_i k(\mathbf{x}_i, \mathbf{x}_i) \\ s.t. \quad & \sum_{i=1}^{l} \alpha_i = 1, \\ & 0 \le \alpha_i \le \frac{1}{\nu l}, \ i = 1, \ldots, l \end{aligned} \tag{2}$$

where $k(\cdot,\cdot)$ is the kernel function which needs to satisfy the Mercer's theorem.

With the solution $c = \sum_{i=1}^{l} \alpha_i \phi(\mathbf{x}_i)$, the decision function can be described as:

$$f(\mathbf{x}) = \mathrm{sgn}(R^2 - \sum_{i,j=1}^{l} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) + 2\sum_{i=1}^{l} \alpha_i k(\mathbf{x}, \mathbf{x}_i) - k(\mathbf{x}, \mathbf{x})) \tag{3}$$

where R^2 is computed from any support vector $\mathbf{x}_i$ with $0 < \alpha_i < \frac{1}{\nu l}$.

2.2 Kernel Matrix for Tensor

For nonlinear cases, kernel method is an essential way to deal with it. For 2nd-order tensor $\mathbf{X}_i \in \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2}$, let a nonlinear function $\Phi(\cdot)$ maps $\mathbf{X}_i$ into a tensor feature space: $\Phi(\mathbf{X}_i) = [\varphi(x_{i1}), \varphi(x_{i2}), \cdots \varphi(x_{in_1})]^T$, where x_{ir} is the r-th row of $\mathbf{X}_i$. By introducing vector kernel function to the element $\varphi(x_{ir_1})\varphi(x_{ir_2})^T$ of kernel matrix[24], one can obtain the Gaussian-based kernel matrix and Polynomial-based kernel matrix as follows:

Gaussian-based kernel matrix

$$K(\mathbf{X}_i, \mathbf{X}_j) = \begin{bmatrix} \mathbf{e}^{-\|x_{i1} - x_{j1}\|^2 / 2\sigma^2} & \cdots & \mathbf{e}^{-\|x_{i1} - x_{jn_1}\|^2 / 2\sigma^2} \\ \vdots & \ddots & \vdots \\ \mathbf{e}^{-\|x_{in_1} - x_{j1}\|^2 / 2\sigma^2} & \cdots & \mathbf{e}^{-\|x_{in_1} - x_{jn_1}\|^2 / 2\sigma^2} \end{bmatrix} \tag{4}$$

Polynomial-based kernel matrix

$$K(\mathbf{X}_i, \mathbf{X}_j) = \begin{bmatrix} \left(x_{i1} x_{j1}{}^T + 1\right)^d & \cdots & \left(x_{i1} x_{jn_1}{}^T + 1\right)^d \\ \vdots & \ddots & \vdots \\ \left(x_{in_1} x_{j1}{}^T + 1\right)^d & \cdots & \left(x_{in_1} x_{jn_1}{}^T + 1\right)^d \end{bmatrix} \tag{5}$$

3. Kernel-based Support Tensor Data Description

3.1 Kernel-based Support Tensor Data Description

Given that training samples $\mathbf{X}_1, \mathbf{X}_2, \cdots \mathbf{X}_l \in \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2}$, $\mathbb{R}^{n_1}$ and $\mathbb{R}^{n_2}$ are two vector spaces. For tensor data description, the hypersphere is characterized by the radius $R > 0$ and the center $C \in \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2}$, and the center C is replaced by the rank one tensor $\mathbf{u}\mathbf{v}^T$ ($\mathbf{u} \in \mathbb{R}^{n_1}$, $\mathbf{v} \in \mathbb{R}^{n_2}$). Therefore, the corresponding decision function is

$$f(\mathbf{X}) = \operatorname{sgn}\left(R^2 - \left\|\Phi(\mathbf{X}) - \mathbf{u}\mathbf{v}^T\right\|^2\right) \tag{6}$$

By minimizing R^2 the volume of the hypersphere can be minimized. Then, Kernel-based Support Tensor Data Description (KSTDD) solves the following OP:

$$\begin{aligned} \min_{\mathbf{u}\in\mathbb{R}^{n_1}, \mathbf{v}\in\mathbb{R}^{n_2}, R\in\mathbb{R}, \xi\in\mathbb{R}^l} \quad & R^2 + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i \\ s.t. \quad & \left\|\Phi(\mathbf{X}_i) - \mathbf{u}\mathbf{v}^T\right\|^2 \le R^2 + \xi_i \\ & \xi_i \ge 0, i = 1, \ldots, l \end{aligned} \tag{7}$$

We solve the OP (7) by constructing the Lagrangian function with multipliers $\alpha_i, \beta_i, i = 1, \ldots, l$:

$$L(\mathbf{u}, \mathbf{v}, R, \xi, \alpha, \beta) = R^2 + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i + \sum_{i=1}^{l}\alpha_i\left(\left\|\Phi(\mathbf{X}_i) - \mathbf{u}\mathbf{v}^T\right\|^2 - R^2 - \xi_i\right) - \sum_{i=1}^{l}\beta_i\xi_i \tag{8}$$

Since

$$\begin{aligned} \left\|\Phi(\mathbf{X}_i) - \mathbf{u}\mathbf{v}^T\right\|^2 &= trace[\left(\Phi(\mathbf{X}_i) - \mathbf{u}\mathbf{v}^T\right)^T\left(\Phi(\mathbf{X}_i) - \mathbf{u}\mathbf{v}^T\right)] \\ &= trace[\Phi(\mathbf{X}_i)\Phi(\mathbf{X}_i)^T] - 2trace[\mathbf{v}\mathbf{u}^T\Phi(\mathbf{X}_i)] + \left(\mathbf{v}^T\mathbf{v}\right)\left(\mathbf{u}^T\mathbf{u}\right) \end{aligned} \tag{9}$$

substituting (9) in (8), we get

$$\begin{aligned} L(\mathbf{u}, \mathbf{v}, R, \xi, \alpha, \beta) = {} & R^2 + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i + \sum_{i=1}^{l}\alpha_i\left(\mathbf{v}^T\mathbf{v}\right)\left(\mathbf{u}^T\mathbf{u}\right) + \sum_{i=1}^{l}\alpha_i trace[\Phi(\mathbf{X}_i)\Phi(\mathbf{X}_i)^T] \\ & - 2\sum_{i=1}^{l}\alpha_i trace[\mathbf{v}\mathbf{u}^T\Phi(\mathbf{X}_i)] - \sum_{i=1}^{l}\alpha_i R^2 - \sum_{i=1}^{l}\alpha_i\xi_i - \sum_{i=1}^{l}\beta_i\xi_i \end{aligned} \tag{10}$$

From KKT conditions, one can obtain:

$$\frac{\partial L}{\partial \xi_i} = 0 \;\Rightarrow\; \frac{1}{\nu l} + \alpha_i - \beta_i = 0 \tag{11}$$

$$\frac{\partial L}{\partial R} = 0 \;\Rightarrow\; \sum_{i=1}^{l}\alpha_i = 1 \tag{12}$$

$$\frac{\partial L}{\partial \mathbf{u}} = 0 \;\Rightarrow\; \mathbf{u} = \frac{1}{\|\mathbf{v}\|^2}\sum_{i=1}^{l}\alpha_i\Phi(\mathbf{X}_i)\mathbf{v} \tag{13}$$

$$\frac{\partial L}{\partial \mathbf{v}} = 0 \;\Rightarrow\; \mathbf{v} = \frac{1}{\|\mathbf{u}\|^2}\sum_{i=1}^{l}\alpha_i \Phi(\mathbf{X}_i)^T \mathbf{u} \tag{14}$$

We notice that $\mathbf{u}$ and $\mathbf{v}$ in (13) and (14) are dependent on each other. The alternating projection method introduced by Cai [13] can be employed to solve.

We firstly fix $\mathbf{u}$, and let $\mu_1 = \|\mathbf{u}\|^2$. According to (11)- (14), (10) can be rewritten as:

$$L = \sum_{i=1}^{l}\alpha_i trace[\Phi(\mathbf{X}_i)\Phi(\mathbf{X}_i)^T] - \frac{1}{\mu_1}\sum_{i,j=1}^{l}\alpha_i\alpha_j \mathbf{u}^T\Phi(\mathbf{X}_i)\Phi(\mathbf{X}_j)^T\mathbf{u} \tag{15}$$

We can derive the following dual OP:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{\mu_1}\sum_{i,j=1}^{l}\alpha_i\alpha_j \mathbf{u}^T K(\mathbf{X}_i, \mathbf{X}_j)\mathbf{u} - \sum_{i=1}^{l}\alpha_i trace[K(\mathbf{X}_i, \mathbf{X}_i)] \\ s.t. \quad & \sum_{i=1}^{l}\alpha_i = 1, \\ & 0 \le \alpha_i \le \frac{1}{\nu l},\; i = 1,\ldots,l \end{aligned} \tag{16}$$

where $K(\cdot,\cdot)$ is defined by (4) or (5). When (16) is solved, we can obtain:

$$\|\mathbf{v}\|^2 = \frac{1}{\mu_1^2}\sum_{i,j=1}^{l}\alpha_i^*\alpha_j^*\mathbf{u}^T K(\mathbf{X}_i, \mathbf{X}_j)\mathbf{u} \tag{17}$$

Then we can calculate $\mathbf{u}$ in the next passage. Let $\mu_2 = \|\mathbf{v}\|^2$, then

$$\begin{aligned} \mathbf{x}_j &= \Phi(\mathbf{X}_j)\mathbf{v} = \frac{1}{\mu_1}\sum_{i=1}^{l}\alpha_i^*\Phi(\mathbf{X}_j)\Phi(\mathbf{X}_i)^T\mathbf{u} \\ &= \frac{1}{\mu_1}\sum_{i=1}^{l}\alpha_i^* K(\mathbf{X}_j, \mathbf{X}_i)\mathbf{u} \end{aligned} \tag{18}$$

Note that $trace[\mathbf{v}\mathbf{u}^T\Phi(\mathbf{X}_i)] = \mathbf{u}^T\Phi(\mathbf{X}_i)\mathbf{v}$, the optimization problem is derived in the dual in the similar lines as:

$$\begin{aligned} \min_{\hat{\alpha}} \quad & \frac{1}{\mu_2}\sum_{i,j=1}^{l}\hat{\alpha}_i\hat{\alpha}_j \mathbf{x}_i^T\mathbf{x}_j - \sum_{i=1}^{l}\hat{\alpha}_i trace[K(\mathbf{X}_i, \mathbf{X}_i)] \\ s.t. \quad & \sum_{i=1}^{l}\hat{\alpha}_i = 1, \\ & 0 \le \hat{\alpha}_i \le \frac{1}{\nu l},\; i = 1,\ldots,l \end{aligned} \tag{19}$$

From (19), we can get the new $\mathbf{u} = \frac{1}{\mu_2}\sum_{i=1}^{l}\hat{\alpha}_i^*\mathbf{x}_i$. Thus, $\mathbf{u}$ can be achieved by iteratively solving (16) and (19).

The optimal boundary (6) is determined by:

$$f(\mathbf{X}) = \mathrm{sgn}(R^2 - trace[K(\mathbf{X},\mathbf{X})] - \frac{1}{\mu_1}\sum_{i,j=1}^{l}\alpha_i^*\alpha_j^*\mathbf{u}^T K\left(\mathbf{X}_i,\mathbf{X}_j\right)\mathbf{u} + \frac{2}{\mu_1}\sum_{i=1}^{l}\alpha_i^*\mathbf{u}^T K\left(\mathbf{X},\mathbf{X}_i\right)\mathbf{u}) \tag{20}$$

and α_i^* is the solution of (16), the training samples $\mathbf{X}_i$ with $\alpha_i^* \neq 0$ are support tensors. In addition, the radius R^2 can be calculated from:

$$R^2 = mean\left(\left\|\Phi(\mathbf{X}_{i\_sv}) - \mathbf{u}\mathbf{v}^T\right\|^2\right) \tag{21}$$

where $\mathbf{X}_{i\_sv}$ is the support tensor.

3.2 Relationship with OCSTM

For the two vector-based one-class classifiers, Scholkopf had shown that for kernels $k(\mathbf{x},\mathbf{y})$ depending on $\mathbf{x}-\mathbf{y}$, the optimization problems of SVDD and One-Class Support Vector Machine (OCSVM) turn out to be equivalent.

We can obtain the similar conclusion for KSTDD and OCSTM. The optimization problem of OCSTM can be iteratively solved with following dual problem:

$$\textbf{OCSTM-v:}\quad \begin{aligned} &\min_{\alpha} \quad \frac{1}{2\mu_1}\sum_{i,j=1}^{l}\alpha_i\alpha_j\mathbf{u}^T K\left(\mathbf{X}_i,\mathbf{X}_j\right)\mathbf{u} \\ &s.t. \quad \sum_{i=1}^{l}\alpha_i = 1, \\ &\qquad 0 \le \alpha_i \le \frac{1}{\nu l},\ i = 1,\dots,l \end{aligned} \tag{22}$$

$$\textbf{OCSTM-u:}\quad \begin{aligned} &\min_{\hat{\alpha}} \quad \frac{1}{2\mu_2}\sum_{i,j=1}^{l}\hat{\alpha}_i\hat{\alpha}_j\mathbf{x'}_i^T\mathbf{x'}_j \\ &s.t. \quad \sum_{i=1}^{l}\hat{\alpha}_i = 1, \\ &\qquad 0 \le \hat{\alpha}_i \le \frac{1}{\nu l},\ i = 1,\dots,l \end{aligned} \tag{23}$$

For Gaussian kernel matrix (4), the elements in the diagonal of the $K\left(\mathbf{X}_i,\mathbf{X}_i\right)$ are: $\varphi(x_{ir_1})\varphi(x_{ir_2})^T = \mathbf{e}^{-\left\|x_{ir_1}-x_{ir_2}\right\|^2/2\sigma^2} = 1$, thus $traceK\left(\mathbf{X}_i,\mathbf{X}_i\right)$ in the dual objective function (16) and (19) is a constant. In this case, the problem (16) and (19) turn out to be equivalent to (22) and (23), respectively. Hence, the two tensor-based one-class classification algorithms coincide in this case.

However, since different kernel matrix leads to different result, it means that the two algorithms cannot be completely equivalent to each other. Therefore, we concentrate on the performance of KSTDD with different kernel matrices in the numerical experiments.

3.3 Proof of convergence

Theorem 1 The objective function values of (7) are monotone decreasing as optimization problems (16) and (19) are solved iteratively. Hence KSTDD algorithm converges.

Proof Define:

$$f(\mathbf{u},\mathbf{v}) = R^2 + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i = mean\left(\left\|\Phi(\mathbf{X}_{i\_sv}) - \mathbf{u}\mathbf{v}^T\right\|^2\right) + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i \qquad (25)$$

Given that $\mathbf{u}_1$ the initial value. We can get an expression of $\mathbf{v}_1$ by (16). Then, we yield $\mathbf{u}_2$ through solving (19) by fixing $\mathbf{v}_1$. Note that (16) and (19) are convex, and they have global solutions. Thus, we can obtain:

$$f(\mathbf{u}_1,\mathbf{v}_1) \geq f(\mathbf{u}_2,\mathbf{v}_1) \geq f(\mathbf{u}_2,\mathbf{v}_2) \geq f(\mathbf{u}_3,\mathbf{v}_2)\cdots$$

Since f is bounded from below by 0, it converges.

4. Experiments on Vector-Based Datasets

4.1 Preparations

There are 9 considered publicly available vector-based datasets, all of them from the UCI repository on LIBSVM webpage [24]. All features are scaled to [-1,1]. Table 1 also gives the considered target classes which we focus on. Normally, we can convert a vector sample $\mathbf{x} \in \mathbb{R}^n$ to a 2nd-order tensor $\mathbf{X} \in \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2}$, based on the idea of [13].

In order to test whether tensor-based algorithms fit for small sample learning with large dimensionality, we use small sizes of training data as in [10,21]. The considered training sizes include 4, 6, 8, and 10 samples, respectively. For statistical significance, we report the average results of fifty random splits. To test the KSTDD, we employ two kinds of kernel matrices (4), (5). To distinguish the KSTDD with these two kernel matrices, we call them gSTDD and pSTDD, respectively. We use $k(\mathbf{x},\mathbf{y}) = \mathbf{e}^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2}$ and $k(\mathbf{x},\mathbf{y}) = (\mathbf{x}^T\mathbf{y}+1)^d$ for the standard SVDD. Similarly, they are shorted as gSVDD and pSVDD in the following passage, respectively. We employ both TPR and AUC as evaluation criteria to compare the performance of algorithms.

In parameters selection, we set k equals to the sizes of training sets for k-fold cross-validation, since the training sets are too small. In our experiments, there exist 3 parameters need to be selected: ν, σ and *d*, which are from the sets ν ={0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}, σ ={2^{-5}, 2^{-4}, …, 2^4, 2^5} and *d*={1,2,3,4,5,6,7,8,9}.

4.2 Experimental Results and Analysis

In this part, we evaluate gSTDD, pSTDD, gSVDD, and pSVDD on all vector-based datasets given in Table 1. To avoid repetition, we only demonstrate the results when there are 10 training samples. The classification results with respect to the other training set sizes will be shown in Fig.1.

Table 1 has summarized the results of both averaged TPRs and AUCs on various datasets. The best ones are shown in boldfaces. Table 2 also includes the experimental results of OCSTM with Gaussian-based kernel matrix. As can be seen, generally, TPRs have been improved greatly by STDD and AUCs between STDD and SVDD have no remarkable difference. Moreover, paired comparisons of different kernel methods are also given in Table 2. We can see that the TPRs of gSTDD and pSTDD are better than those of gSVDD and pSVDD in all comparisons, respectively. For AUC, we see that pSTDD is better than or similar to pSVDD in 9 out of 10 comparisons. The AUCs of gSTDD are similar to those of gSVDD in 5 comparisons and not much worse in 5 comparisons. Furthermore, the two kernel systems make very little difference to STDD rather than to SVDD in all the 10 datasets. In addition, it was also verified by the experiments that OCSTM and STDD turn to be equivalent for Gaussian-based kernel matrix.

Table 1

Averaged TPRs and AUCs of 9 datasets, 10 samples.

Datasets	Target Class		gOCSTM	gSTDD (a)	gSVDD (b)	(a) VS (b)	pSTDD (c)	pSVDD (d)	(c) VS (d)
BREAST-CANCER	2	TPR	0.7525	0.7525	0.6757	>	**0.7601**	0.6943	>
		AUC	0.9669	0.9669	**0.9740**	~	0.9681	0.9704	~
ABALONE	1	TPR	0.7477	**0.7477**	0.6370	>	0.7409	0.6099	>
		AUC	0.7703	0.7703	0.7585	~	**0.7789**	0.7667	~
HEART	1	TPR	0.7116	0.7116	0.5125	>	**0.7172**	0.4979	>
		AUC	0.7133	0.7133	**0.7681**	<	0.6754	0.6777	~
HEPATITIS	2	TPR	0.6102	**0.6102**	0.4392	>	0.6064	0.4665	>
		AUC	0.6376	0.6376	**0.7027**	<	0.6601	0.6992	~
IMPORTS	1	TPR	0.7588	0.7588	0.5528	>	**0.7608**	0.6633	>
		AUC	0.6916	0.6916	**0.7051**	~	0.6829	0.6018	>
IONOSPHERES	1	TPR	0.7552	0.7552	0.6160	>	**0.7620**	0.7152	>
		AUC	0.7196	0.7196	**0.7979**	<	0.7340	0.7524	~
SPECTF	2	TPR	0.7088	**0.7088**	0.4807	>	0.7055	0.5281	>

		AUC	0.7537	0.7537	**0.8108**	<	0.7363	0.7787	<
DELFTPUMP	2	TPR	0.7844	0.7844	0.3730	>	**0.8030**	0.7252	>
		AUC	0.5781	0.5781	**0.6721**	<	0.5750	0.5285	>
USPS	2	TPR	0.8722	**0.8722**	0.6965	>	0.7949	0.7241	>
		AUC	0.9612	0.9612	**0.9783**	~	0.9747	0.9777	~

To show the performances of the four algorithms on the different training sizes, we tested gSTDD, gSVDD, pSTDD and pSVDD over 4, 6, 8, and 10 samples for all datasets shown in Table 1. To give a clear display, we present the experimental results in Fig. 1.

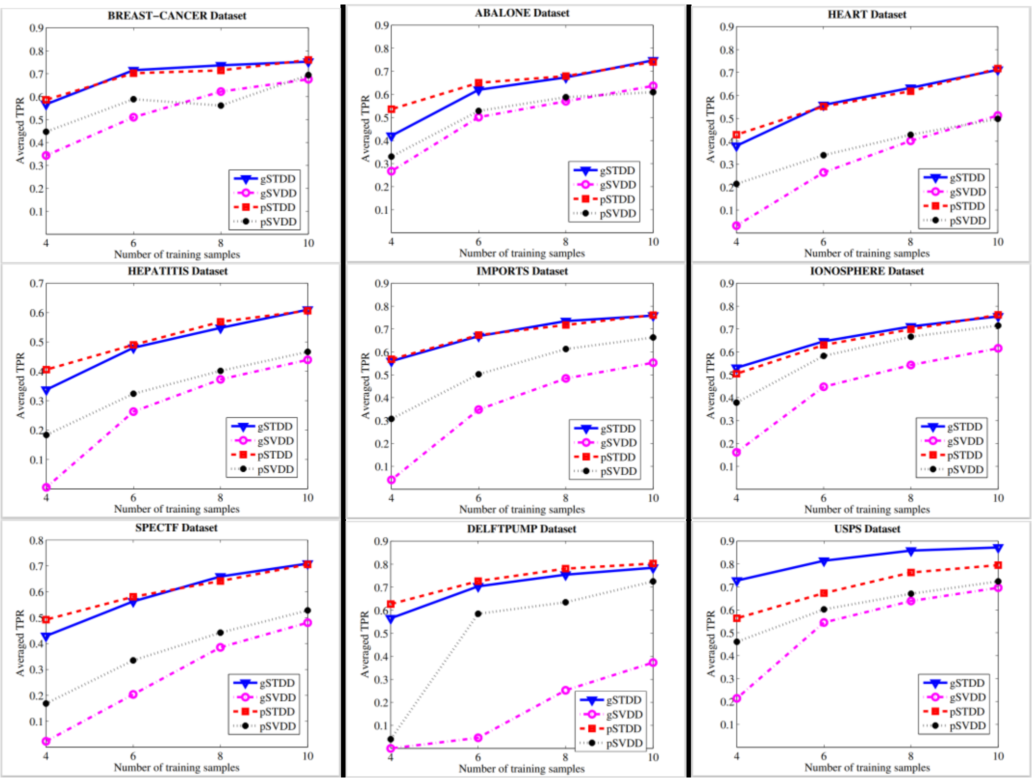

Fig. 1. Averaged TPRs: gSTDD, gSVDD, pSTDD and pSVDD on various datasets with the different training sizes.

It can be seen that the averaged TPRs of gSTDD and pSTDD are much better than those of gSVDD and pSVDD, especially for small training set. When the number of training samples increases, the TPRs of the four algorithms tend to get close.

5. Experiments on Tensor-Based Datasets

This section focuses on human face images on the YALE dataset [19]. They are all grayscale images which can be represented as second order tensors. It has fifteen people's faces with eleven images for each one, we show some examples for each dataset in Figure 2. The images in YALE are size of 100×100, and all the features are scaled to [0,1].

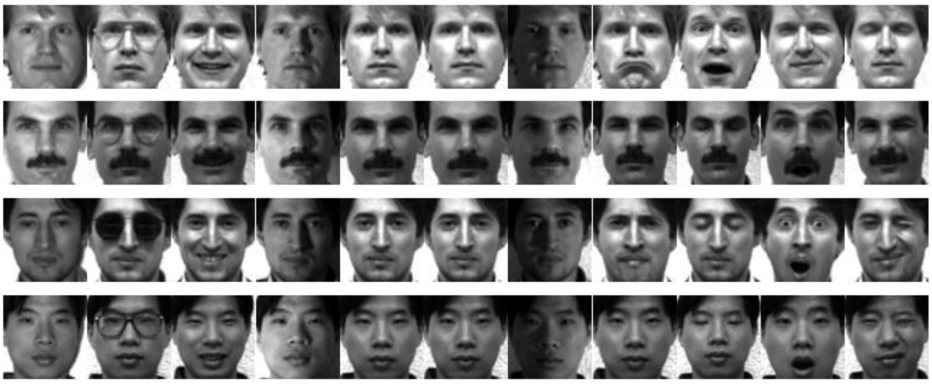

Fig. 2. YALE human faces dataset.

We utilize the 5-fold cross validation to optimize the parameters. Since there are 3 testing samples in each fold of cross validation, it is more meaningful to concentrate on the true positive rate (TPR), which are reported in Table 2.

Table 2

Averaged TPRs of YALE dataset.

Target Class	gSTDD	gSVDD	pSTDD	pSVDD
1	**0.9333**	0.7333	**0.9333**	0.8667
2	**0.9333**	0.4667	0.7333	**0.9333**
3	**0.9333**	0.7333	0.7333	0.8
4	**0.9333**	0.6667	**0.9333**	0.8667
5	**0.9333**	0.8667	0.8667	0.8667
6	**0.9333**	0.8	0.8667	**0.9333**
7	**0.8667**	0.4	0.8	0.4667
8	**0.9333**	0.7333	**0.9333**	0.8
9	**0.9333**	0.6	**0.9333**	0.8
10	0.9333	0.4	**1**	0.6667
11	**0.8667**	0.7333	**0.8667**	**0.8667**
12	**0.8667**	0.7333	**0.8667**	**0.8667**
13	**0.8667**	**0.8667**	**0.8667**	**0.8667**
14	**0.9333**	0.8	0.8	0.8667
15	0.5333	0.5333	0.5333	**0.6**

Table 2 summarize the averaged TPRs of each target class. The best classification results are shown in boldfaces. We can see that the result of the STDD is also well on tensor dataset. But it has some difference from ORL. gSTDD has achieved almost all the best TPRs, except in the experiments on face 10 and 15. The averaged TPR of 15 experiments of gSTDD is 0.8889, which is much better than 0.6711 of gSVDD. On the contrary, the averaged TPR of 15 experiments of pSTDD is 0.8444, which is a little better than 0.8044 of pSVDD. These indicate that the different kernel matrices may lead to different classification performance.

6. Conclusions

In the paper, we present a tensor-based data description named Support Tensor Data Description. KSTDD finds a hypersphere with smallest volume in tensor feature space with kernel trick, that can involve most data of the target class. Since the input of KSTDD is tensor, KSTDD has the advantages of efficiently keeping data topology and considerably reducing the number of parameters. We evaluate KSTDD on both Gaussian and polynomia kernel matrices. As to be expected, KSTDD shows better generalization capability.

However, the new algorithm has some drawbacks. KSTDD costs longer training time than SVDD since it is iterative to solve parameters involved in the alternative projection algorithm. It is worth investigating more faster methods to solve KSTDD's optimization problems. In addition, further study includes the kernel technology on KSTDD for high order tensors.

Acknowledgement

The work is supported by the National Natural Science Foundation of China under No. 61907033.

R E F E R E N C E S

[1]. V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, USA, (1995).
[2]. D. M. J. Tax, R. P. W. Duin, Support vector domain description, Mach. Learn. 54 (1)(2004): 45-66.
[3]. X. Chen, Z. Cui, J. Shen, L. Ma, Fault analysis and identification of motor bearing based on esmd and svm, UPB Scientific Bulletin, Series C: Electrical Engineering and Computer Science, Volume 83(4)(2021):239-250.
[4]. Y. Yu, L. Kong, Y. Liu, J. Song, PSO optimized SVM parameters for far infrared pedestrian detection, UPB Scientific Bulletin, Series C: Electrical Engineering and Computer Science, Volume 81(4)(2019):15-26.
[5]. Turkoz M, Kim S, Son Y, et al, Generalized support vector data description for anomaly detection. Pattern Recognition, 100(2020): 107119.
[6]. Mygdalis V, Iosifidis A, Tefas A, et al, Semi-supervised subclass support vector data description for image and video classification, Neurocomputing, 278(2018): 51-61.
[7]. Y. Zou, H. Wu, X. Guo, et al, MK-FSVM-SVDD: a multiple kernel-based fuzzy SVM model for predicting DNA-binding proteins via support vector data description, Current Bioinformatics, 16(2)(2021): 274-283.

[8]. G. Zhong, Y, Xiao,B. Liu, et al, Pinball loss support vector data description for outlier detection, Applied Intelligence, 2022: 1-22.

[9]. Sadeghi R, Hamidzadeh J, Automatic support vector data description, Soft Computing, 22(1)(2018): 147-158.

[10]. H. Liu, Y. Yang, N. Hu, et al, Soft-margin hyperdisk tensor machine and its application to the fault intelligent diagnosis of rotating machinery, Measurement Science and Technology, 2022.

[11]. T. Sun, X. Sun, New Results on Classification Modeling of Noisy Tensor Datasets: A Fuzzy Support Tensor Machine Dual Model, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021.

[12]. X. Wu, S. J. Maybank, W. Hu, X. Li, D. Tao, Supervised tensor learning, IEEE International Conference on Data Mining, IEEE Computer Society, 13(2005):450-457.

[13]. D. Cai, X. Hei, J. Han, Learning with tensor representation, Department of Computer Science, University of Illinois at Urbana-Champaign, UIUCDCS-R-2006-2716 (2006)

[14]. Z. Hao, L. He, B. Chen, X. Yang, A linear support higher-order tensor machine for classification, IEEE Transactions on Image Processing 22(7)(2013): 2911-2920.

[15]. C. Yang, M. Jia, Hierarchical multiscale permutation entropy-based feature extraction and fuzzy support tensor machine with pinball loss for bearing fault identification, Mechanical Systems and Signal Processing, 149(2021): 107182.

[16]. Z. He, H. Shao, J. Cheng, et al, Support tensor machine with dynamic penalty factors and its application to the fault diagnosis of rotating machinery with unbalanced data, Mechanical systems and signal processing, 141(2020): 106441.

[17]. C. Hu, S. He, Y. Wang, A classification method to detect faults in a rotating machinery based on kernelled support tensor machine and multilinear principal component analysis, Applied Intelligence, 51(4)(2021): 2609-2621.

[18]. S. Li, N. Yang, B. Li, Z. Zhou, H. Hao, M. R. Folkert, J. Wang, A pilot study using kernelled support tensor machine for distant failure prediction in lung SBRT, Medical image analysis, 50(2018): 106-116.

[19]. C. Yang, M. Jia, Hierarchical multiscale permutation entropy-based feature extraction and fuzzy support tensor machine with pinball loss for bearing fault identification, Mechanical Systems and Signal Processing, 149(2021): 107182.

[20]. Z. He, H. Shao, J. Cheng, et al, Kernel flexible and displaceable convex hull based tensor machine for gearbox fault intelligent diagnosis with multi-source signals, Measurement, 163(2020): 107965.

[21]. C. Chen, K. Batselier, W. Yu, N. Wong, Kernelized support tensor train machines, Pattern Recognition, 122(2022), 108337.

[22]. Razzak I, Moustafa N, Mumtaz S, et al. One-class tensor machine with randomized projection for large-scale anomaly detection in high-dimensional and noisy data, International Journal of Intelligent Systems, 37(8)(2022): 4515-4536.

[23]. Maboudou-Tchao E M, Support tensor data description, Journal of Quality Technology, 53(2)(2021): 109-134.

[24]. Y. Chen, K. Wang, P. Zhong, One-class support tensor machine, Knowledge-Based Systems, 96(2016): 14-28.

[25]. Y. Chen, K. Wang, P. Zhong, A linear support higher order tensor domain description for one-class classification, Journal of Intelligent & Fuzzy Systems, 34(6)(2018): 4237-4247.