Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 19, No. 4, pp. 675–688, 2023

AN UPDATED OT-PCF ALGORITHM FOR ONE-CLASS CLASSIFICATION OF TENSOR DATASETS*

Min Wang, Qiuling Hou, Haibin Chen†, Yi Xu and Guanglu Zhou

Abstract: One-class classification is an important problem encountered in many applications. In this paper, we present a new polyhedral conic function algorithm based on tensor data called OT-PCF algorithm. Since the level set of PCF is polyhedron, only convex decision boundary can be obtained by PCF. However, the target class may have a non-convex structure. To overcome this drawback, the proposed algorithm divides the target class into k clusters first and then a PCF for each cluster can be obtained. The final classifier of the OT-PCF algorithm is given as the minimum of k PCFs to generate non-convex separating surfaces. By testing on real-world datasets, experiments show that the proposed algorithm is a promising method for handling one-class classification problems with tensor inputs.

Key words: *one-class classification, polyhedral conic functions, tensor datasets, support tensor machine*

Mathematics Subject Classification: *65H17, 15A18, 90C30*

1 Introduction

Machine learning is one of the most popular topics in recent years, in which the so called one-class classification problem is one of important and interesting problems [15, 26, 22, 29]. The purpose of supervised data classification is to label test data by a classification algorithm based on training data. Different from the traditional classification problem, one-class classification studies a special classification problem that only one class of training samples is available or reliable, while others are either expensive to acquire or difficult to characterize. To the best of our knowledge, the one-class classification problem has many applications such as fault diagnosis, face recognition, the network anomaly detection and text classification etc.

One-class classifier was first proposed by Moya *et al.* in [22]. In 2014, Khan and Madden comprehensively elaborated the algorithm, technology and significance of one-class classification [15]. Based on the pioneer work of [22, 15], several kinds of problems are presented such as outlier detection [25, 16], novelty detection [30, 31, 11], concept learning [36] and single class classification [32, 19]. Very recently, Cimen and Ozturk [10] developed a novel one-class polyhedral conic function(O-PCF) algorithm for the one-class classification problem. The level set of a PCF is a convex polyhedron, thus, only convex decision boundaries

*This work is supported by National Natural Science Foundation of China (Grant No. 12071249) and Shandong Provincial Natural Science Foundation of Distinguished Young Scholars (Grant No. ZR2021JQ01).
†Corresponding author.

can be obtained with one PCF. However, the target class may have a non-convex structure. So, the O-PCF algorithm divides the target class into k clusters and obtains a PCF for each cluster. Compared to other one-class classifier in the literature, the test results tell us that the O-PCF algorithm outperforms the other methods in many cases [10].

Noted that, most one-class classifier such as one-class support vector machines(SVM) are encoded in vector and matrix spaces. The main idea is to construct a decision boundary around positive data in order to distinguish it from outliers data [20, 26, 27, 10]. However, in many real world applications data are represented more naturally as higher order tensors. For example, sensor data are often organised into the three modes of location, type, and time, while videos are represented as 3D objects corresponding to concatenated frames over time. Thus, with the development of tensor theory, more and more classifiers have been proposed to extend SVM to tensor space, i.e., so-called support tensor machines(STM) that take a tensor as input [3, 13, 21, 28, 34, 35]. In [8], Chen *et al.* presented a linear support higher order tensor domain description(LSTDD) for one-class classification problem. It aims to find a closed hypersphere with the minimal volume in the tensor space which can contain almost entirely of the target samples. The superiority is that the LSTDD algorithm can keep data topology and make the parameters need to be estimated less, and it is more suitable for learning the high dimensional and small sample size problem. Furthermore, there are many other algorithms based on tensor data, such as, the hyperplane-based one-class STMs [7, 9, 6] and Transductive STM [33, 18] are extended to high-order tensor case.

In this paper, based on the efficient O-PCF algorithm in [10], we give a new updated version named OT-PCF for tensor datasets as input. We first recall the O-PCF algorithm in detail for one-class classification with vectors. Then, it's extended to OT-PCF algorithm for the high-order tensor case. Finally, we test the OT-PCF algorithm by experiments on both the matrices datasets and the higher order tensor datasets, and compare it with the LSTDD algorithm.

The rest of this paper is organized as follows. In Section 2, we first recall several useful definitions and symbols. Then, we give an introduction for the existed O-PCF algorithm. In Section 3, we extend the O-PCF algorithm to the high-order case, and a new OT-PCF algorithm is established. Several numerical results are given in Section 4. Finally, a conclusion is given in Section 5.

2 Notation and Preliminaries

In this section, we first recall some preliminaries and some useful symbols. To continue, it should be noted that vectors (scalars) are denoted by boldface lowercase letters like $\mathbf{x}$(lowercase letters like x). Matrices are represented by capital letters like A and tensors are represented by calligraphic capitals like $\mathcal{T}$. Furthermore, the l_1-norm of vector $\mathbf{x}$ is denoted by $\|\mathbf{x}\|_1$. Similarly, the l_1-norm of matrix $M \in R^{m\times n}$ and tensor $\mathcal{Z} \in R^{n_1\times n_2\times\cdots n_m}$ are defined such that

$$\|M\|_1 = \sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|,\ \|\mathcal{Z}\|_1 = \sum_{i_1,i_2,\cdots,i_m=1}^{n}|a_{i_1 i_2\cdots i_m}|.$$

The tensor product of vectors $\mathbf{u} \in R^{n_1}$ and $\mathbf{v} \in R^{n_2}$ is denoted by $W = \mathbf{u}\otimes\mathbf{v} \in R^{n_1\times n_2}$. The Euclidean inner product of $\mathbf{u}$ and $\mathbf{v}$ is denoted by $\langle\mathbf{u},\mathbf{v}\rangle$. An m-th order tensor $\mathcal{Z}$ is called a rank-1 tensor if there are vectors $\mathbf{z}_i \in R^{n_i}(1 \le i \le m)$ such that

$$\mathcal{Z} = \mathbf{z}_1 \otimes \mathbf{z}_2 \cdots \otimes \mathbf{z}_m = \prod_{i=1}^{m} \otimes \mathbf{z}_i.$$

2.1 Polyhedral conic functions(PCF)

We now recall the concept of polyhedral conic function(PCF) defined in [12]. The function $g : R^n \to R$ is a called PCF if it satisfies the following equation:

$$g_{(\mathbf{w},\xi,\gamma,\mathbf{c})}(\mathbf{x}) = \langle \mathbf{w}, (\mathbf{x} - \mathbf{c}) \rangle + \xi \|\mathbf{x} - \mathbf{c}\|_{\mathbf{1}} - \gamma, \tag{2.1}$$

where $\mathbf{w}, \mathbf{c} \in R^n$ are given vectors and $\xi, \gamma \in R$. It has been showed in [12] that the graph of (2.1) is a cone with vertex at $(\mathbf{c}, -\gamma) \in R^n \times R$ and each sublevel set is a polyhedron. Such a function can separate two sets by dividing the entire space into two parts, an inside region and an outside region. Here $\mathbf{x}$ is the object to be classified, $\mathbf{c}$ is the centre of the PCF, and $\mathbf{w}, \xi$, γ are additional parameters of the PCF. Let A and B be two given sets in R^n:

$$A = \{\mathbf{a}^1, \mathbf{a}^2, ..., \mathbf{a}^m\},\ \mathbf{a}^i \in R^n,\ i \in I = \{1, 2, ..., m\},$$

$$B = \{\mathbf{b}^1, \mathbf{b}^2, ..., \mathbf{b}^q\},\ \mathbf{b}^j \in R^n,\ j \in J = \{1, 2, ..., q\}.$$

To separate two sets A and B by a PCF, the parameter $\mathbf{c}$ is defined in advance, and the parameters $\mathbf{w}, \xi$,γ are then optimized by minimizing the classification errors, where A is the target class and B is the outlier class. These parameters are used to define the piecewise setting of a polyhedral conic function that divides the entire space into two parts, so that all points of B remain outside, and as many points of A as possible remain within the piecewise set. The test points $\mathbf{x}$ can be categorized, if $g(\mathbf{x}) \leq 0$, then $\mathbf{x} \in A$; if $g(\mathbf{x}) > 0$, then $\mathbf{x} \in B$.

2.2 One-class polyhedral conic functions algorithm(O-PCF algorithm)

Now, we recall the O-PCF algorithm for one-class classification problem. In one-class classification, there are no available data information from the outlier class. Therefore, one can only get the data information from the target class. When a classifier is proposed, only the target class can be used for training. Very recently, the one-class polyhedral conic functions(O-PCF) algorithm was proposed in [10], where the O-PCF algorithm can successfully separates the target class A from the outliers. The corresponding linear programming(LP) model is as follows:

$$\begin{aligned} \min_{\mathbf{w},\xi,\gamma} \quad & \sum_{i\in I} -(\langle \mathbf{w}, (\mathbf{a}^i - \mathbf{c}) \rangle + \xi \|\mathbf{a}^i - \mathbf{c}\|_1 - \gamma) + \lambda \sum_{i \in I} z_i \\ \text{s.t.} \quad & \langle \mathbf{w}, (\mathbf{a}^i - \mathbf{c}) \rangle + \xi \|\mathbf{a}^i - \mathbf{c}\|_1 - \gamma \leq z_i,\ \forall i \in I \\ & \xi, \gamma \geq 1,\ z_i \geq 0,\ \forall i \in I, \end{aligned}$$

where $\mathbf{w}, \mathbf{a}, \mathbf{c} \in R^n$ and $\xi, \gamma, \lambda, z \in R$. Here, $\mathbf{c}$ is the centroid of the target set A and is calculated in advance before the solution of the LP problem. While that $\mathbf{w}, \xi, \gamma$ are obtained from the solution of the LP problem. The lower bounds on ξ and γ are defined to be 1 to ensure closed convex polyhedral level sets. The bigger λ is, the less tolerance it has for mistakes. Let $g(\mathbf{a}^i) = \langle \mathbf{w}, (\mathbf{a}^i - \mathbf{c}) \rangle + \xi \|\mathbf{a}^i - \mathbf{c}\|_1 - \gamma$. Then it aims to minimize the size of the decision boundaries. The parameter λ controls the tradeoff between the size of the decision boundaries and the size of the classification error. The variable z_i represents the classification error of data object $\mathbf{a}^i \in A$. When $g(\mathbf{a}^i) \leq 0$, that means the classification is correct, and $z_i = 0$; When $g(\mathbf{a}^i) > 0$, that means there is a misclassification, and z_i is positive i.e.

$$z_i = \begin{cases} g(\mathbf{a}^i), & \text{if } g(\mathbf{a}^i) > 0,\ \forall i \in I, \\ 0, & \text{if } g(\mathbf{a}^i) \leq 0,\ \forall i \in I. \end{cases}$$

Since the level set of PCF is convex [29], the O-PCF algorithm can only be used for convex decision boundaries. However, in many cases, the target class is non-convex. For this reason, the k-means algorithm can be used to divide target class A into k clusters in advance. Then, one can obtain the clusters A_r and their centers $\mathbf{c}_r$, $r = 1, 2, \cdots, k$, where

$$A_r = \{\mathbf{a}^1, \cdots, \ \mathbf{a}^{l_r}\}, \ \mathbf{a}^i \in R^n, \ i \in I_r = \{1, \cdots, l_r\}.$$

By the results above, the following model is presented for each cluster to obtain parameters $\mathbf{w}_r, \xi_r, \gamma_r$:

$$\begin{aligned} \min_{w_r, \xi_r, \gamma_r} \quad & \sum_{i \in I_r} -(\langle \mathbf{w}_r, (a^i - c_r) \rangle + \xi_r \|a^i - c_r\|_1 - \gamma_r) + \lambda \sum_{i \in I_r} z_i \\ \text{s.t.} \quad & \langle \mathbf{w}_r, (a^i - c_r) \rangle + \xi_r \|a^i - c_r\|_1 - \gamma_r \le z_i, \ \forall i \in I_r \\ & \xi, \gamma \ge 1, \ z_i \ge 0, \ \forall i \in I_r. \end{aligned} \tag{2.1}$$

Eventually, the final classifier is obtained as the point-wise minimum of these PCFs, as shown in Equation (2.2).

$$G(\mathbf{x}) = \min_{r=1,\cdots,k} \langle \mathbf{w}_r, (\mathbf{x} - \mathbf{c}_r) \rangle + \xi_r \|\mathbf{x} - \mathbf{c}_r\|_1 - \gamma_r. \tag{2.2}$$

Equation (2.2) provides to label any test data point $\mathbf{x}$ as target class if the value of the $\mathbf{x}$ is negative in at least one PCF among k PCFs. In other words, in order to label the $\mathbf{x}$ as outlier, the value of the $\mathbf{x}$ should be positive in all k PCFs.

3 OT-PCF Algorithm for One-Class Classification with Tensors

In this section, we extend O-PCF to the tensor based model, and present an OT-PCF algorithm to find a minimal volume hypersphere in the tensor space to contain target samples. We firstly present OT-PCF algorithm with matrices, then extend it to the higher order tensor space.

3.1 OT-PCF algorithm with matrices

Suppose the training set A is defined as follows:

$$A = \{A^1, ..., A^l\}, \ A^i \in R^{n \times m}, \ i \in I = \{1, ..., l\}.$$

Then, we divide the set A into several clusters using the k-means algorithm. Obtain the clusters A_r and the cluster centres are obtained as a result of this step as $C_r, r = 1, \cdots, k$, where:

$$A_r = \{A^{r_1}, ..., A^{r_{t_r}}\}, \ A^{r_i} \in R^{n \times m}, \ r_i \in I_r = \{r_1, ..., r_{t_r}\}, t_r \in I,$$

and $t_1 + t_2 + \cdots + t_k = l$. Once the clusters A_r and the centres C_r are obtained. Then, OT-PCF algorithm aims to solve the following LP problem:

$$\begin{aligned} \min_{W_r, \xi_r, \gamma_r} \quad & \sum_{i \in I_r} -(\langle W_r, (A^i - C_r) \rangle + \xi_r \|A^i - C_r\|_1 - \gamma_r) + \lambda \sum_{i \in I_r} z_i \\ \text{s.t.} \quad & \langle W_r, (A^i - C_r) \rangle + \xi_r \|A^i - C_r\|_1 - \gamma_r \le z_i, \ \forall i \in I_r, \\ & \xi_r, \gamma_r \ge 1, \ z_i \ge 0, \ \forall i \in I_r, \end{aligned}$$

where parameters $W_r, C_r \in R^{n\times m}$, $\xi_r, \gamma_r, \lambda, z \in R$. What's more, parameter W_r can be expressed as a rank-1 tensor $\mu_r \otimes \upsilon_r$, $\mu_r \in R^n$ and $\upsilon_r \in R^m$. Then the LP model can be written as follows:

$$\begin{aligned} \min_{\mu_r,\upsilon_r,\xi_r,\gamma_r} \quad & \sum_{i\in I_r} -(\langle \mu_r \otimes \upsilon_r, (A^i - C_r)\rangle + \xi_r\|A^i - C_r\|_1 - \gamma_r) + \lambda \sum_{i\in I_r} z_i \\ \text{s.t.} \quad & \langle \mu_r \otimes \upsilon_r, (A^i - C_r)\rangle + \xi_r\|A^i - C_r\|_1 - \gamma_r \le z_i,\ \forall i \in I_r \\ & \xi_r, \gamma_r \ge 1,\ z_i \ge 0,\ \forall i \in I_r. \end{aligned} \tag{3.1}$$

Next, we use an alternative minimization method to solve (3.1). Firstly, fix υ_r and solve μ_r. In other words, we solve the problem (3.2) and obtain μ_r, ξ_r and γ_r :

$$\begin{aligned} \min_{\mu_r,\xi_r,\gamma_r} \quad & \sum_{i\in I_r} -(\langle \mu_r, (A^i - C_r)\upsilon_r\rangle + \xi_r\|A^i - C_r\|_1 - \gamma_r) + \lambda \sum_{i\in I_r} z_i \\ \text{s.t.} \quad & \langle \mu_r, (A^i - C_r)\upsilon_r\rangle + \xi_r\|A^i - C_r\|_1 - \gamma_r \le z_i,\ \forall i \in I_r \\ & \xi_r, \gamma_r \ge 1,\ z_i \ge 0,\ \forall i \in I_r. \end{aligned} \tag{3.2}$$

Then, fix μ_r to solve υ_r, it follows that

$$\begin{aligned} \min_{\upsilon_r,\xi_r,\gamma_r} \quad & \sum_{i\in I_r} -(\langle \upsilon_r, (A^i - C_r)\mu_r\rangle + \xi_r\|A^i - C_r\|_1 - \gamma_r) + \lambda \sum_{i\in I_r} z_i \\ \text{s.t.} \quad & \langle \upsilon_r, (A^i - C_r)\mu_r\rangle + \xi_r\|A^i - C_r\|_1 - \gamma_r \le z_i,\ \forall i \in I_r \\ & \xi_r, \gamma_r \ge 1,\ z_i \ge 0,\ \forall i \in I_r. \end{aligned} \tag{3.3}$$

For any $r \in \{1, \cdots, k\}$, similar way can be used to solve the corresponding optimization problems. And then, we can get some PCFs and the number of PCFs obtained is equal to the number of clusters. Eventually, the final classifier is obtained as the point-wise minimum of these PCFs, and the corresponding decision function is given as follows:

$$G(X) = \min_{r=1,\cdots,k} \langle W_r, (X - C_r)\rangle + \xi_r\|X - C_r\|_1 - \gamma_r, \tag{3.4}$$

where $W_r = \mu_r \otimes \upsilon_r$.

3.2 OT-PCF algorithm with high order tensors

Suppose the training set $\mathcal{T}$ is given as follows:

$$\mathcal{T} = \{\mathcal{T}^1, \ldots, \mathcal{T}^l\},\ \mathcal{T}^i \in R^{n_1\times\cdots\times n_m},\ i \in I = \{1, \ldots, l\},$$

and we divide the set $\mathcal{T}$ into clusters using the k-means algorithm. Then, the clusters $\mathcal{T}_r$ and the cluster centres are obtained as a result of this step as $\mathcal{C}_r, r = 1, \cdots, k$, where:

$$\mathcal{T}_r = \{\mathcal{T}^{r_1}, \ldots, \mathcal{T}^{r_{t_r}}\},\ \mathcal{T}^{r_i} \in R^{n_1\times\cdots\times n_m},\ r_i \in I_r = \{r_1, \ldots, r_{t_r}\}.$$

Then, the corresponding optimization problem is as follows:

$$\begin{aligned} \min_{\mathcal{W}_r,\xi_r,\gamma_r} \quad & \sum_{i\in I_r} -(\langle \mathcal{W}_r, (\mathcal{T}^i - \mathcal{C}_r)\rangle + \xi_r\|\mathcal{T}^i - \mathcal{C}_r\|_1 - \gamma_r) + \lambda \sum_{i\in I_r} z_i \\ \text{s.t.} \quad & \langle \mathcal{W}_r, (\mathcal{T}^i - \mathcal{C}_r)\rangle + \xi_r\|\mathcal{T}^i - \mathcal{C}_r\|_1 - \gamma_r \le z_i,\ \forall i \in I_r \\ & \xi_r, \gamma_r \ge 1,\ z_i \ge 0,\ \forall i \in I_r, \end{aligned} \tag{3.5}$$

where parameters $\mathcal{W}_r, \mathcal{C}_r \in R^{n_1 \times \cdots \times n_m}$; $\xi_r, \gamma_r, \lambda_r, z \in R$. Noted that the parameter $\mathcal{W}_r$ can be expressed as a rank-1 tensor $\mu_{1_r} \otimes \mu_{2_r} \otimes \cdots \otimes \mu_{m_r}$. The problem (3.5) can be formulated as follows:

$$\begin{aligned} \min_{\mu_{j_r}, \xi_r, \gamma_r} \quad & \sum_{i \in I_r} -(\langle \mu_{1_r} \otimes \mu_{2_r} \otimes \cdots \otimes \mu_{m_r}, (\mathcal{T}^i - \mathcal{C}_r) \rangle + \xi_r \|\mathcal{T}^i - \mathcal{C}_r\|_1 - \gamma_r) + \lambda_r \sum_{i \in I_r} z_i \\ \text{s.t.} \quad & \langle \mu_{1_r} \otimes \cdots \otimes \mu_{m_r}, (\mathcal{T}^i - \mathcal{C}_r) \rangle + \xi_r \|\mathcal{T}^i - \mathcal{C}_r\|_1 - \gamma_r \le z_i, \ \forall i \in I_r \\ & \xi_r, \gamma_r \ge 1, \ z_i \ge 0, \ \forall i \in I_r. \end{aligned} \tag{3.6}$$

To solve (3.6) , we also use alternative minimization method to do that. First of all, fix $\mu_{2_r}, \cdots, \mu_{m_r}$ to solve μ_{1_r}, then the optimization model (3.6) becomes the following model:

$$\begin{aligned} \min_{\mu_{j_r}, \xi_r, \gamma_r} \quad & \sum_{i \in I_r} -(\langle \mu_{1_r}, (\mathcal{T}^i - \mathcal{C}_r) \mu_{2_r} \otimes \cdots \otimes \mu_{m_r} \rangle + \xi_r \|\mathcal{T}^i - \mathcal{C}_r\|_1 - \gamma_r) + \lambda \sum_{i \in I_r} z_i \\ \text{s.t.} \quad & \langle \mu_{1_r}, (\mathcal{T}^i - \mathcal{C}_r) \mu_{2_r} \otimes \cdots \otimes \mu_{m_r} \rangle + \xi_r \|\mathcal{T}^i - \mathcal{C}_r\|_1 - \gamma_r \le z_i, \ \forall i \in I_r \\ & \xi_r, \gamma_r \ge 1, \ z_i \ge 0, \ \forall i \in I_r. \end{aligned}$$

Then, fix $\mu_{1_r}, \mu_{3_r}, \cdots, \mu_{m_r}$ to solve μ_{2_r}, it follows that:

$$\begin{aligned} \min_{\mu_{j_r}, \xi_r, \gamma_r} \quad & \sum_{i \in I_r} -(\langle \mu_{2_r}, (\mathcal{T}^i - \mathcal{C}_r) \mu_{1_r} \otimes \mu_{3_r} \otimes \cdots \otimes \mu_{m_r} \rangle + \xi_r \|\mathcal{T}^i - \mathcal{C}_r\|_1 - \gamma_r) + \lambda \sum_{i \in I_r} z_i \\ \text{s.t.} \quad & \langle \mu_{2_r}, (\mathcal{T}^i - \mathcal{C}_r) \mu_{1_r} \otimes \mu_{3_r} \otimes \cdots \otimes \mu_{m_r} \rangle + \xi_r \|\mathcal{T}^i - \mathcal{C}_r\|_1 - \gamma \le z_i, \ \forall i \in I_r \\ & \xi_r, \gamma_r \ge 1, \ z_i \ge 0, \ \forall i \in I_r. \end{aligned}$$

Repeat the procedure step by step, until we obtain all the μ_{j_r}, ξ_r and γ_r, $j = 1, 2, \ldots, m$. Then, we get some PCFs and the number of PCFs obtained is equal to the number of clusters. Finally, the final classifier is obtained as the point-wise minimum of these PCFs. And the corresponding decision function is given in (3.7) :

$$G(\mathcal{X}) = \min_{r=1,\cdots,k} \langle \mathcal{W}_r, (\mathcal{X} - \mathcal{C}_r) \rangle + \xi_r \|\mathcal{X} - \mathcal{C}_r\|_1 - \gamma_r, \tag{3.7}$$

with $\mathcal{W}_r = \mu_{1_r} \otimes \mu_{2_r} \otimes \cdots \otimes \mu_{m_r}$.

To end this section, we present the computing detail of the OT-PCF algorithm.

Algorithm 3.1: OT-PCF Algorithm

Input: Data points $\mathcal{T}^i \in \mathcal{T},\ \lambda \in [0,\infty),\ k \in [1,\infty)$.

Output: The set of PCFs separating the set $\mathcal{T}$ from the outliers.
Step 1: (Clustering). Divide the set $\mathcal{T}$ into clusters using the k-means algorithm. Obtain the clusters $\mathcal{T}_r$ and their centres $\mathcal{C}_r,\ r = 1,\cdots,k$, where:

$$\mathcal{T}_r = \{\mathcal{T}^1,...,\mathcal{T}^{l_r}\},\ \mathcal{T}^i \in R^{n_1\times\cdots\times n_m},\ i \in I = \{1,...,l_r\}$$

Step 2:(Computation of the PCFs).
for $r = 1$ **to** k **do:**
Solve the following LP model to find g_r for cluster $\mathcal{T}_r$:

$$\begin{aligned}\min_{\mathcal{W}_r,\xi_r,\gamma_r} \quad & \sum_{i\in I_r} -(\langle \mathcal{W}_r, (\mathcal{T}^i - \mathcal{C}_r)\rangle + \xi_r\|\mathcal{T}^i - \mathcal{C}_r\|_1 - \gamma_r) + \lambda_r \sum_{i\in I_r} z_i \\ \text{s.t.} \quad & \langle \mathcal{W}_r, (\mathcal{T}^i - \mathcal{C})\rangle + \xi_r\|\mathcal{T}^i - \mathcal{C}_r\|_1 - \gamma_r \le z_i,\ \forall i \in I_r \\ & \xi_r, \gamma_r \ge 1,\ z_i \ge 0,\ \forall i \in I_r,\end{aligned}$$

where $\mathcal{W}_r = \mu_{1_r} \otimes \mu_{2_r} \otimes \cdots \otimes \mu_{m_r}$.
end for
Step 3:(Obtaining the final classifier).

$$G(\mathcal{X}) = \min_{r=1,\cdots,k} \langle \mathcal{W}_r, (\mathcal{X} - \mathcal{C}_r)\rangle + \xi_r\|\mathcal{X} - \mathcal{C}_r\|_1 - \gamma_r.$$

Step 4: end.

4 Numerical Experiments

In this section, we evaluate the performance of the OT-PCF algorithm in comparison with LSTDD method in the literature. All experiments are finished in Matlab2014b on a Lenovo computer with Intel(R) Core(TM)i5-3230M CPU @ 2.60GHz 2.20 GHz and 4 GB of RAM.

4.1 Experiments on vector datasets

The experiments are performed on 7 publicly available datasets from UCI repository [17]. Table 1 gives the description of these datasets. The approach of transforming a vector $\mathbf{x} \in R^n$ to a matrix $Z \in R^{n_1\times n_2}$ is that the first column of Z is filled by the first n_1 elements of $\mathbf{x}$, and the second column of Z is filled by the next n_1 elements of $\mathbf{x}$ and so on. If $n < n_1 n_2$, then the rest is filled with 0. Following the idea in [7, 4], the proper values of n_1 and n_2 are also listed in Table 1. What's more, the Abalone dataset is reorganized into three one-class classification problems. In each problem, one class is considered as the target class and the two others are treated as outliers.

Table 1: Description of the datasets

Dataset	Sample	Dimensions	$n_1 \times n_2$	Targ.Class	Targ.Sample
AbaloneF	4177	8	2×4	1	1307
AbaloneI	4177	8	2×4	2	1342
AbaloneM	4177	8	2×4	3	1528
Breast-cancer	683	9	3×3	2	443
Hepatitis	155	19	4×5	2	123
Iris	150	4	2×2	1	50
Spect	267	44	4×11	2	212

In one class classification problem, the training set is formed by target samples, the **true positive rate(tpr)** is used as the performance evaluation criterion. We employ 5-fold cross validation to optimize the parameter. Here, the parameter λ is chosen from $\{1, 10, 100, 1000, 10000\}$ and the parameter k is chosen from $\{1, 3, 5, 10\}$. Tpr is defined in Equation (4.1).

$$tpr = \frac{tp}{tp + fn}, \tag{4.1}$$

where tp, fp and fn correspond to the true positive, false positive and false negative respectively. Noted that larger tpr achieves better classification in the tests.

Table 2: Averaged tprs on various datasets

Dataset	Targ.class	OT-PCF	LSTDD
AbaloneF	1	**0.9288**	0.5989
AbaloneI	2	**0.9106**	0.4363
AbaloneM	3	**0.9771**	0.6137
Breast-cancer	2	**0.9521**	0.6892
Hepatitis	2	**0.9036**	0.6187
Iris	1	**0.9400**	0.7771
Spect	2	**0.7228**	0.6935

The testing results on the 7 datasets are concluded in Table 2. As we can see, the tpr of OT-PCF algorithm is significantly promoted in comparison with LSTDD algorithm. By a direct computing, the average value of tpr through 7 datasets are 0.9050 of OT-PCF algorithm and 0.6312 of LSTDD algorithm.

What's more, the effects on tpr with different k and λ are shown in Table 3 and Table 4 respectively. From Table 3, it shows that the tpr will increase with the increasing of λ for any fixed k, and it is extremely clear when λ is from 1 to 100. However, Table 4 that, if λ is fixed, the value k has almost no impact on tpr.

Table 3: Averaged tprs with different values of λ on some datasets when k=3

λ	Breast-cancer	AbaloneF	Iris
1	0.1667	0.0285	0.0040
10	0.5473	0.8873	0.8860
100	0.9049	0.9338	0.8880
1000	0.9245	0.9419	0.9000
10000	0.9352	0.9426	0.9160

Table 4: Averaged tprs with different values of k on some datasets when $\lambda = 1000$

k	Breast-cancer	AbaloneF	Iris
1	0.9887	0.9992	0.9400
3	0.9775	0.8871	0.8800
5	0.9910	0.8279	0.8600
10	0.9977	0.9623	0.8800

4.2 Experiments on matrix datasets

In this section, we focus on testing the proposed algorithm based on human face datasets that are included in ORL [1] and the FERET [14]. The ORL dataset includes 40 individuals face images, and each face has 10 different images. Each image is 112×92 with 256 grayscale levels per pixel. The FERET dataset includes 200 persons images, and each one has 7 different images. Each image is 80×80 with 256 grayscale levels per pixel. The 5-fold cross validation is still adopted for optimizing parameters, and report the averaged results in Table 5 and Table 6.

Table 5: Averaged tprs on 40 target classes in ORL dataset

Targ.cls	OT-PCF	LSTDD	Targ.cls	OT-PCF	LSTDD	Targ.cls	OT-PCF	LSTDD
Face01	**0.6**	0.5	Face15	0.6	**0.7**	Face29	**0.9**	0.8
Face02	0.7	0.7	Face16	**0.5**	0.4	Face30	0.6	**0.8**
Face03	**0.8**	0.7	Face17	**0.7**	0.6	Face31	**0.9**	0.8
Face04	**0.6**	0.5	Face18	**0.9**	0.6	Face32	**0.7**	0.6
Face05	0.7	**0.8**	Face19	**0.8**	0.7	Face33	**0.9**	0.8
Face06	0.6	0.6	Face20	**0.7**	0.6	Face34	0.7	**0.8**
Face07	**0.6**	0.5	Face21	**0.9**	0.7	Face35	**0.9**	0.5
Face08	0.6	**0.8**	Face22	0.7	**0.8**	Face36	**0.9**	0.6
Face09	**0.7**	0.6	Face23	**0.7**	0.6	Face37	**0.8**	0.6
Face10	0.6	**0.8**	Face24	0.7	**0.8**	Face38	0.5	**0.6**
Face11	0.7	0.7	Face25	**0.7**	0.6	Face39	**0.6**	0.5
Face12	0.8	0.8	Face26	0.6	**0.7**	Face40	**0.6**	0.5
Face13	**0.8**	0.6	Face27	0.8	0.8			
Face14	0.5	0.5	Face28	**0.7**	0.6			

In Table 5, the averaged *tpr* for OT-PCF algorithm and LSTDD algorithm are 0.6900 and 0.655 respectively. Specifically, the *tpr*s of OT-PCF algorithm are better than LSTDD algorithm in 24 out of 40 comparisons, and equal in the other 6 comparisons.

For the test on FERET datasets, the first 100 samples are selected for the experiment. For each sample, the target contains 7 sample points, and the outliers are 693 sample points. So it is clear that each experiment is an unbalanced classification problem. Therefore, we introduce two commonly used evaluation indicators for unbalanced classification problems: *tnr* and *Gmeans* [5]. The *Gmeans* represents the geometric mean of *tpr* and *t*nr. Larger *tnr* and *Gmeans* achieve better classification in the tests. To continue, we list the definition of tnr and Gmeans below:

$$tnr = \frac{tn}{fp + tn}, \qquad Gmeans = \sqrt{tpr \times tnr}.$$

As can be seen in Table 6, the test results of OT-PCF algorithm is quite good, and the averaged value of *tpr*s, *tnr*s and *Gmeans* of all 100 datasets are 0.6860, 0.8763, 0.7667 respectively.

4.3 Experiments on higer-order tensor datasets

In this section, we focus on testing the proposed algorithm based on color human face datasets that are included in ABERDEEN and the Caltech Color Face datasets. The ABERDEEN dataset includes 629 images, and each person has 4-15 different images. Each image is $32 \times 32 \times 3$. In the experiments, face images of one person are considered as a target class, we randomly choose 15 target classes, and each target class has 10 samples.

Table 6: Averaged results on 100 target classes in FERET dataset

Targ.cls	tprs	tnrs	Gmeans	Targ.cls	tprs	tnrs	Gmeans
Face01	0.7143	0.9606	0.8283	Face51	0.5714	0.9971	0.7548
Face02	0.5714	0.9409	0.7332	Face52	0.5714	0.9913	0.7526
Face03	0.7143	0.6552	0.6844	Face53	0.7143	0.9942	0.8427
Face04	0.5714	0.8768	0.7079	Face54	1.0000	0.8455	0.9195
Face05	1.0000	0.6995	0.8364	Face55	1.0000	0.9242	0.9614
Face06	0.6000	0.8050	0.6950	Face56	0.5714	0.9592	0.7403
Face07	0.7143	0.6650	0.6892	Face57	0.7143	1.0000	0.8452
Face08	0.7143	0.9901	0.8410	Face58	0.5714	1.0000	0.7559
Face09	0.5714	0.7734	0.6684	Face59	0.5714	0.9883	0.7515
Face10	0.7143	0.6897	0.7019	Face60	1.0000	0.9359	0.9674
Face11	0.5714	0.9015	0.7177	Face61	0.5714	0.9913	0.7526
Face12	0.7143	0.5419	0.6221	Face62	0.8571	0.5862	0.7052
Face13	0.8571	0.5271	0.6722	Face63	0.5714	0.9942	0.7537
Face14	1.0000	0.9163	0.9572	Face64	0.7143	0.8805	0.7930
Face15	0.4286	0.9901	0.6514	Face65	0.5714	0.8688	0.7046
Face16	0.8571	0.9901	0.9212	Face66	0.5714	0.9446	0.7343
Face17	0.7143	0.9360	0.8176	Face67	0.7143	0.9592	0.8277
Face18	1.0000	0.9754	0.9876	Face68	0.5714	0.6647	0.6163
Face19	0.4286	0.8818	0.6147	Face69	0.7143	0.9767	0.8352
Face20	0.8571	1.0000	0.9258	Face70	0.5714	0.9883	0.7515
Face21	0.8571	0.8325	0.8447	Face71	0.5714	0.9971	0.7548
Face22	0.5714	0.9261	0.7275	Face72	0.5714	0.9300	0.7290
Face23	0.5714	0.7241	0.6433	Face73	0.5714	0.8630	0.7022
Face24	0.4286	0.8325	0.5973	Face74	0.5714	0.9184	0.7244
Face25	0.7143	0.9951	0.8431	Face75	0.5714	0.9125	0.7221
Face26	0.4286	0.8473	0.6026	Face76	0.7143	0.8717	0.7891
Face27	1.0000	0.9360	0.9675	Face77	0.5714	0.9038	0.7186
Face28	0.8571	0.8473	0.8522	Face78	0.5714	0.7026	0.6336
Face29	0.7143	0.8719	0.7892	Face79	0.5714	0.9446	0.7347
Face30	1.0000	0.7635	0.8738	Face80	0.5714	0.9883	0.7515
Face31	0.5714	0.7594	0.6587	Face81	0.5714	0.9621	0.7415
Face32	0.7143	0.9850	0.8388	Face82	0.4286	0.9125	0.6254
Face33	0.7143	0.8195	0.7651	Face83	0.5714	0.7744	0.6652
Face34	1.0000	0.7293	0.8540	Face84	0.5714	0.6536	0.6738
Face35	0.8571	0.8271	0.8420	Face85	0.5714	0.8047	0.6781
Face36	0.7143	0.9248	0.8128	Face86	0.5714	0.9213	0.7526
Face37	0.8571	0.8647	0.8609	Face87	0.7143	0.9323	0.8161
Face38	0.7143	0.8346	0.7721	Face88	0.5714	0.8271	0.6875
Face39	0.8571	1.0000	0.9258	Face89	0.5714	0.8947	0.7150
Face40	0.7143	0.9925	0.8420	Face90	0.8571	0.8872	0.8721
Face41	1.0000	0.7669	0.8757	Face91	0.7143	0.8496	0.6968
Face42	1.0000	0.9774	0.9887	Face92	0.5714	0.7068	0.6355
Face43	0.5714	0.9173	0.7240	Face93	0.7143	0.8947	0.7994
Face44	0.8571	0.5714	0.6999	Face94	0.7143	0.8947	0.7150
Face45	0.8571	0.9774	0.9153	Face95	0.7143	0.6767	0.6952
Face46	0.5714	0.9624	0.7416	Face96	0.4286	0.9398	0.6347
Face47	0.5714	0.9323	0.7299	Face97	0.4286	1.0000	0.6547
Face48	0.8571	0.9248	0.8903	Face98	0.5714	0.7970	0.6749
Face49	0.7143	0.9173	0.8095	Face99	0.5714	0.9398	0.7328
Face50	0.8571	0.9023	0.8794	Face100	0.5714	0.9624	0.7416

For the Caltech Color Face dataset, it has 18 target classes, and each target class has 20 samples. Each image is 50×50×3. We adopt the 5-fold cross validation for optimizing parameters, and report the averaged results. Table 7 and Table 8 have summarized the averaged results of tprs, tnrs and Gmeans on target classes of these two datasets, respectively. As we can see, in ABERDEEN dataset, the average value of tprs; tnrs; and Gmeans through 15 datasets are 0.8200; 0.7677 and 0.7871. What's more, in Caltech Color Face dataset, the average value of tprs; tnrs; and Gmeans through 18 datasets are 0.7222; 0.8186 and 0.7591.

Table 7: Averaged results on 15 target classes in ABERDEEN dataset

Targ.cls	tprs	tnrs	Gmeans	Targ.cls	tprs	tnrs	Gmeans
Face01	0.7000	0.8071	0.7517	Face09	0.7000	0.7214	0.7106
Face02	0.7000	0.7643	0.7314	Face10	0.9000	0.8000	0.8485
Face03	0.7000	0.8143	0.7550	Face11	0.8000	0.6214	0.7051
Face04	0.8000	0.5000	0.6325	Face12	1.0000	0.8429	0.9181
Face05	0.7000	0.8286	0.7616	Face13	1.0000	0.7143	0.8452
Face06	0.6000	0.9214	0.7435	Face14	1.0000	0.9786	0.9892
Face07	1.0000	0.7857	0.8864	Face15	0.7000	0.7929	0.7450
Face08	1.0000	0.6143	0.7838				

Table 8: Averaged results on 18 target classes in Caltech Color Face dataset

Targ.cls	tprs	tnrs	Gmeans	Targ.cls	tprs	tnrs	Gmeans
Face01	1.0000	0.7912	0.8894	Face10	0.5000	0.9353	0.6838
Face02	0.7500	0.6941	0.7215	Face11	0.5500	0.8235	0.6730
Face03	0.7000	0.9882	0.8317	Face12	0.6000	0.8824	0.7276
Face04	0.6500	0.9324	0.7785	Face13	0.7000	0.8735	0.7820
Face05	0.6000	0.8500	0.7141	Face14	1.0000	0.5676	0.7534
Face06	0.6000	0.8088	0.6966	Face15	0.6500	0.8441	0.7407
Face07	0.9500	0.7941	0.8686	Face16	0.7500	0.8706	0.8080
Face08	0.6000	0.8676	0.7215	Face17	0.7500	0.8882	0.8162
Face09	0.9000	0.5029	0.6728	Face18	0.7500	0.8206	0.7845

5 Conclusion

In this paper, a new one-class classification algorithm with the tensor input was proposed. The classifier was trained by the target class only. We first divided the target class into clusters by the k-means algorithm in order to change the non-convex decision boundary into a convex set. Next, a PCF was found for each cluster by solving an LP model and the minimum of these PCFs yielded the final classifier. Finally, to show the performance of the proposed algorithm, several numerical experiments were given based on different datasets. The test results lead us to conclude that the OT-PCF algorithm outperforms the LSTDD algorithm in many cases.

Acknowledgments

We are very grateful to the editor and the anonymous reviewers for their constructive comments and meaningful suggestions on our manuscript.

References

[1] ATT Labs Cambridge, The Olivetti and Oracle Research Laboratory database of faces (1994), http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[2] A. Bagirov, J. Ugon, D. Webb, G. Ozturk and R.Kasimbeyli, A novel piecewise linear classifier based on polyhedral conic and max-min separabilities, *Top* 21 (2013) 3–24.

[3] D. Cai, X. He, J. Wen, J. Han and W. Ma, Support tensor machines for text categorization, Technical report, Department of Computer Science, University of Illinois at Urbana-Champaign, (2006) UIUCDCS-R-2006-2714.

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

[5] Y. Chen, One class classification model and algorithm based on tensor theory, China Agricultural University, 2016 (in Chinese).

[6] Y. Chen, L. Lu and P. Zhong, One class support higher order tensor machine classifier, *Applied Intelligence* 7 (2017) 1–9.

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

[8] Y. Chen, K. Wang and P. Zhong, A linear support higher order tensor domain description for onc-class classification, *Journal of Intelligent & Fuzzy Systems* 34 (2018) 4237–4247.

[9] Y. Chen and P. Zhong, Linear one-class support tensor machine, *International Journal of Signal Processing, Image Processing and Pattern Recognition* 9 (2016) 379–388.

[10] E. Cimen and G. Ozturk, O-PCF algorithm for one-class classification, *Optimization Methods and Software* 35 (2020) 1065–1079.

[11] G. Dohnal and I. Bukovsk, Novelty detection based on learning entropy, *Applied Stochastic Models in Business and Industry* 36 (2020) 178–183.

[12] R. Gasimov and G. Ozturk, Separation via polyhedral conic functions, *Optimization Methods and Software* 21 (2006) 527–540.

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

[14] Information Technology Laboratory (2023), Available at http://www.nist.gov/itl/iad/ig/colorferet.cfmhttp://www.nist.gov/itl/iad/ig/colorferet.cfm.

[15] S. Khan and M. Madden, One-class classification: taxonomy of study and review of techniques, *Knowledge Engineering Review* 29 (2014) 345–374.

[16] G. Lee and S. Park, Multiple outlier detection in logistic regression by using influence matrix, *Journal of the Korean Statistical Society* 36 (2007) 457–469.

[17] M. Lichman, UCI machine learning repository (2013), Available at http://archive.ics.uci.edu/ml.

[18] X. Liu, T. Guo, L. He and X. Yang, A low-aank approximation-based transductive support tensor machine for semisupervised classification, *IEEE Transactions on Image Processing* 24 (2015) 1825–1838.

[19] D. Lorch and M. Kirschmer, Single-class genera of positive integral lattices, *LMS Journal of Computation and Mathematics* 16 (2013) 172–186.

[20] L. Manevitz and M. Yousef, One–class svms for document classification, *Journal of Machine Learning Research* 2 (2002) 139–154.

[21] M. Mavroforakis and S. Theodoridis, A geometric approach to Support Vector Machine (SVM) classification, *IEEE Transactions on Neural Networks* 17 (2006) 671–682.

[22] M. Moya, M. Koch and L. Hosteler, One-class classifier networks for target recognition applications, *NASA STI/Recon Technical Report N* (1993): 24043.

[23] G.Ozturk, A.Bagirov and R.Kasimbeyli, An incremental piecewise linear classifier based on polyhedral conic separation, *Machine Learning* 101 (2015) 397–413.

[24] G. Ozturk and M. Ciftci, Clustering based polyhedral conic functions algorithm in classification, *Journal of Industrial Management Optimization* 11 (2014) 921–932.

[25] K. Ro, C. Zou, Z. Wang and G.Yin, Outlier detection for high-dimensional data, *BIOMETRIKA* 102 (2015) 589–599.

[26] B. Scholkopf, J. Platt, J. Taylor, A. Smola and R. Williamson, Estimating the support of a high-dimensional distribution, *Neural Comput* 13 (2001) 1443–1471.

[27] D. Shin and S. Kim, Nearest mean classification via one-Class svm, *IEEE Computer Society* 1 (2009) 593–596.

[28] D. Tao, X. Li, X. Wu and S. Maybank, Supervised tensor learning, *Knowledge and Information Systems* 13 (2005) 450–457.

[29] D. Tax, *One-Class Classification*, Applied Sciences, 2001.

[30] F. Tsai, Y. Zhang, A. Kwee and W. Tang, Multilingual novelty detection, *Expert Systems with Applications* 38 (2011) 652–658.

[31] B. Tsang and W. Schultz, Deep neural network classifier for variable stars with novelty detection capability, *The Astrophysical Journal Letters* 877 (2019): L14.

[32] S. Wang and X. Yao, Relationships between diversity of classification ensembles and single-class performance measures, *IEEE Transactions on Knowledge and Data Engineering* 25 (2011) 206–219.

[33] F. Wu, Y. Liu and Y. Zhuang, Tensor-based transductive learning for multimodality video semantic concept detection, *IEEE Transactions on Multimedia* 11 (2009) 868–878.

[34] L. Zhang, W. Zhou and L. Jiao, Wavelet support vector machine, *IEEE Transactions on Systems, Man, and Cybernetics*, Part B (Cybernetics) 34 (2004) 34–39.

[35] X. Zhang, X. Gao and Y. Wang, Twin support tensor machines for MCs detection, *Journal of Electronics (China)* 26 (2009) 318–325.

[36] Y. Zhao, J. Li, W. Liu and W. Xu, Cognitive concept learning from incomplete information, *International Journal of Machine Learning and Cybernetics* 8 (2017) 159–170.

Manuscript received 2 June 2021
revised 1 July 2022
accepted for publication 28 July 2022

Min Wang
School of Management Science, Qufu Normal University
No. 80, Yantai Road, Donggang District, Rizhao Shandong
276800, P.R. China
E-mail address: 782091135@qq.com

Qiuling Hou
School of Management Science, Qufu Normal University
No. 80, Yantai Road, Donggang District, Rizhao Shandong
276800, P.R. China
E-mail address: qiulinghouhou@163.com

Haibin Chen
Department of Information Science and Technology
East China University of Political Science and Law
Beijing 102249, China
E-mail address: chenhaibin508@163.com

Yi Xu
School of Mathematics, Southeast University
Nanjing Jiangsu, 211189, P.R. China
E-mail address: yi.xu1983@hotmail.com

Guanglu Zhou
Department of Mathematics and Statistics, Curtin University
Perth, WA, 6000, Australia
E-mail address: g.zhou@curtin.edu.au