Scientific Research Publishing

Journal of Computer and Communications, 2024, 12, 186-200
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Least Squares One-Class Support Tensor Machine

Kaiwen Zhao, Yali Fan

College of Science, University of Shanghai for Science and Technology, Shanghai, China
Email: yalifan@usst.edu.cn

How to cite this paper: Zhao, K.W. and Fan, Y.L. (2024) Least Squares One-Class Support Tensor Machine. *Journal of Computer and Communications*, **12**, 186-200.
https://doi.org/10.4236/jcc.2024.124014

Received: March 12, 2024
Accepted: April 23, 2024
Published: April 26, 2024

Abstract

One-class classification problem has become a popular problem in many fields, with a wide range of applications in anomaly detection, fault diagnosis, and face recognition. We investigate the one-class classification problem for second-order tensor data. Traditional vector-based one-class classification methods such as one-class support vector machine (OCSVM) and least squares one-class support vector machine (LSOCSVM) have limitations when tensor is used as input data, so we propose a new tensor one-class classification method, LSOCSTM, which directly uses tensor as input data. On one hand, using tensor as input data not only enables to classify tensor data, but also for vector data, classifying it after high dimensionalizing it into tensor still improves the classification accuracy and overcomes the over-fitting problem. On the other hand, different from one-class support tensor machine (OCSTM), we use squared loss instead of the original loss function so that we solve a series of linear equations instead of quadratic programming problems. Therefore, we use the distance to the hyperplane as a metric for classification, and the proposed method is more accurate and faster compared to existing methods. The experimental results show the high efficiency of the proposed method compared with several state-of-the-art methods.

Keywords

Least Square One-Class Support Tensor Machine, One-Class Classification, Upscale, Least Square One-Class Support Vector Machine, One-Class Support Tensor Machine

1. Introduction

The classification problem is an important component of data mining and has been discussed very extensively in fields such as machine learning and pattern

recognition. As a result, many kinds of classification methods have emerged. Although deep learning has the advantages of high accuracy and adaptivity, classification models based on deep learning are usually large-sample models and have a long training time, thus, this paper focuses on small-sample machine learning classification methods. In machine learning, fully supervised classification has been one of the main focuses of research. For example, support vector machine (SVM) [1], which is a statistical learning method with an important role in machine learning, statistical analysis, classification and regression. Like SVM, most of the existing methods are for multi-class classification. However, in many real cases, there may be only one type of data available for collection. This is due to the high cost of acquiring data information from other categories than the target category, or the difficulty of portraying data information. In this paper, we focus on the one-class classification algorithm, which differs most from the multi-class classification algorithm in that it uses information from only one class.

One-class classification problems [2] [3] [4] are common in real-life areas, such as face recognition, text classification, novelty detection, and anomaly detection, among others. One-class support vector machine (OCSVM) [5] is an extension of SVM, which finds a hyperplane in the sense of maximum margins that separate most samples from the origin. Another method derived from SVM is support vector domain description (SVDD) [6]. Unlike OCSVM, SVDD maps the data to a high-dimensional feature space and then finds a hypersphere to surround most of the samples. Similar to OCSVM, a one-class mini-max probability machine (OCMPM) [7] tries to maximize the distance between the origin and learned hyper-place with the objective of arriving at a tighter lower bound to the data. Generalized one-class discriminative Sub-spaces (GODS) [8] extends OCSVM formulation into two separating hyper-planes. The least square one-class support vector machine (LSOCSVM) [9] is also a variant of OCSVM that replaces the original loss with a squared loss. LSOCSVM finds the hyperplane by solving a system of linear equations instead of using quadratic programming (QP) solutions like OCSVM.

Each of the above vector methods has its own advantages in dealing with the vector one-class classification problem. However, in many fields such as image processing, machine learning and pattern recognition, raw data are presented as multidimensional arrays. If we downscale the tensor data into vectors for classification, not only it will cause overfitting problems due to increased data dimensionality, but also a large amount of structural information in the data will be lost, resulting in a decrease in classification accuracy. In addition, vector data upscaling into tensor data is a more popular scheme. When one-dimensional vector data is elevated to multi-dimensional, the original linking relationships are preserved while the probability is high that the linking of other dimensions in the data is also added. During the upscaling process, connections between vector elements may be identified, thus enhancing the classification accuracy [10] [11] [12]. Therefore, it is necessary to explore the tensor classification methods.

Based on the support vector machine(SVM) learning framework and combining the ideas of alternate projection and multiple linear algebra operations, Tao *et al.* proposed a supervised tensor learning (STL) framework [13] with tensors as input data. Under this framework, linear models based on tensor classification were first proposed [14] [15]. Since data is usually not linearly separable in real life, kernel methods were represented in the tensor datasets [16] [17]. Kernel support tensor regression [18] and kernelized support tensor machines [19] [20] were suitable for nonlinear regression and nonlinear separable classification problems. Cai *et al.* proposed a support tensor machine (STM) [21] based on the STL framework and the alternating projection algorithm. Similar to OCSVM, which was the one-class extension model of SVM, Chen *et al.* proposed the One-Class Support Tensor Machine (OCSTM) [22]. Maboudou-Tchao [23] proposed support tensor data description (STDD). These tensor one-class classification methods were based on the hinge loss function, thus requiring the solution of a quadratic programming problem, so these methods were time consuming.

In this paper, we propose a novel one-class tensor classification method, called the least squares one-class support tensor machine (LSOCSTM), which combines the advantages of LSOCSVM and OCSTM. On one hand, the proposed LSOCSTM is based on tensor space, which takes tensors directly as inputs. Obtaining a classifier in tensor space not only preserves the data structure information, but also helps to overcome the over-fitting problem caused by vectorization. On the other hand, compared to OCSTM, LSOCSTM solves a system of learning equations in each iteration instead of QP problems. These iterations eventually converge to the optimal solution after several iterations, greatly saving computational time and complexity. The key contributions of this work are summarized as follows:

- We propose a new one-class classification method called LSOCSTM for second order tensors.
- Upscaling of the vector data to tensor can effectively solve the data over-fitting problem and improve the classification accuracy.
- The proposed method solves a system of learning equations in each iteration instead of QP problems, which has a time advantage in comparison with existing tensor methods.
- We develop the corresponding algorithm and prove its convergence by theoretical analysis and experiments.
- We have done a lot of comparative experiments to prove the effectiveness of the method in this paper by comparing it with the existing tensor-based and vector-based methods.

The rest of this paper is organized as follows. In section 2, we give a brief overview of least squares one-class support vector machine, one-class support tensor machine and the kernel function for tensors. The proposed least squares one-class support tensor machine is described in section 3. The experimental

results are presented in section 4. Finally, some concluding remarks and suggestions for future work are given in section 5.

2. Preliminaries

2.1. Tensor Kernel Function

In practical problems, the datasets are usually nonlinearly separable, so a nonlinear mapping function $\Phi(\cdot)$ is needed to map the original dataset to a high-dimensional Hilbert space, i.e., $x \to \Phi(x)$. Thus, the nonlinear problem in the original space can be transformed into a linear classification problem in a high-dimensional Hilbert space, and then go on to find the optimal hyperplane. The kernel function is defined as follows:

$$K(\mathbf{X},\mathbf{Y}) = \left(\Phi(\mathbf{X}), \Phi(\mathbf{Y})\right), \tag{1}$$

where $(,)$ denotes the inner product of Hilbert spaces, each sample $\mathbf{X},\mathbf{Y} \in \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2}$.

Gao *et al.* [18] proposed a kernel function in which the input data are tensors. This function $\Phi(\mathbf{X})$ is a nonlinear mapping that maps $\mathbf{X}$ to a high-dimensional feature space. Similar to Gao *et al.* [18], we define $\Phi(\mathbf{X})$ as:

$$\Phi(\mathbf{X}) = \begin{bmatrix} \phi(x_1) \\ \phi(x_2) \\ \vdots \\ \phi(x_{n_1}) \end{bmatrix}, \tag{2}$$

where x_k is the kth row of $\mathbf{X}$. Correspondingly, the following y_k is the kth row of $\mathbf{Y}$. Therefore, this kernel function can be defined as

$$\begin{aligned} K(\mathbf{X},\mathbf{Y}) = \Phi(\mathbf{X})\Phi(\mathbf{Y})^{\mathrm{T}} &= \begin{bmatrix} \phi(x_1) \\ \phi(x_2) \\ \vdots \\ \phi(x_{n_1}) \end{bmatrix} \begin{bmatrix} \phi(y_1) \\ \phi(y_2) \\ \vdots \\ \phi(y_{n_1}) \end{bmatrix}^{\mathrm{T}} \\ &= \begin{bmatrix} \phi(x_1)\phi(y_1)^{\mathrm{T}} & \cdots & \phi(x_1)\phi(y_{n_1})^{\mathrm{T}} \\ \vdots & \ddots & \vdots \\ \phi(x_{n_1})\phi(y_1)^{\mathrm{T}} & \cdots & \phi(x_{n_1})\phi(y_{n_1})^{\mathrm{T}} \end{bmatrix}. \end{aligned} \tag{3}$$

2.2. Least Square One-Class Support Vector Machine

Choi derived least square one-class support vector machine (LSOCSVM) [9] which is a least squares version of standard One-Class Support Vector Machine (OCSVM) [5]. Suppose the training samples are $\{\mathbf{x}_i : i = 1,2,\cdots,l\}$, and $\mathbf{x}_i \in \mathbb{R}^n$. LSOCSVM can be formulated as the following optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{\omega},\rho,\xi} \frac{1}{2}\|\boldsymbol{\omega}\|^2 - \rho + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2, \\ &\text{s.t. } \boldsymbol{\omega}\cdot\phi(\mathbf{x}_i) = \rho - \xi_i, \end{aligned} \tag{4}$$

where $\boldsymbol{\omega}$ is the normal vector of the hyperplane and ρ is the intercept of the hyperplane. C is the penalty parameter. Unlike OCSVM [5], the restriction on slack variable $\xi_i \geq 0$ no longer exists. Optimization problem (4) tries to find a hyperplane which has the maximal distance $\frac{\rho}{\boldsymbol{\omega}^2}$ from the origin, and with respect to this distance, ξ_i^2 are minimized. To solve the optimization problem, Lagrangian multipliers $\alpha_i, i = 1, 2, \cdots, l$ are introduced for these equality constrains. The Lagrangian function can be written as follows:

$$L(\boldsymbol{\omega}, \rho, \xi, \alpha) = \frac{1}{2}\|\boldsymbol{\omega}\|^2 - \rho + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 - \sum_{i=1}^{l}\alpha_i\left(\boldsymbol{\omega}\cdot\phi(\mathbf{x}_i) - \rho + \xi_i\right). \tag{5}$$

Setting the derivatives with respect to the primal variables $\boldsymbol{\omega}, \rho, \xi_i, \alpha$ equal to zero, and then simplifying, which gives

$$\sum_{j=1}^{l}\alpha_j\phi(\mathbf{x}_i)\phi(\mathbf{x}_j) - \rho + \frac{\alpha_i}{C} = 0. \tag{6}$$

Unlike OCSVM, LSOCSVM does not have a decision function. Instead, the hyperplane in (6) itself represents the optimal hyperplane in a regularized least squares sense, where most of training objects may reside. Least squares minimizes the sum of squared errors and establishes an equilibrium between the errors of the individual equations, thus preventing one extreme error from gaining dominance. The computational process is clear and convenient as only the partial derivatives are required to solve the system of linear equations. However, the LSOCSVM method is not directly adaptive to tensor data.

2.3. One-Class Support Tensor Machine

Chen developed the one-class support tensor machine (OCSTM) [22], which separates most samples of interested class from the origin in the tensor space, with maximal margin. Suppose the training samples $\{\mathbf{X}_i : i = 1, 2, \cdots, l\}$, every sample $\mathbf{X}_i \in \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2}$ which is the data point in 2nd-order tensor space. The decision function of an one-class classifier in the tensor space can be represented as follows:

$$f(\mathbf{X}) = sgn\left(\boldsymbol{u}^{\mathrm{T}}\Phi(\mathbf{X})\boldsymbol{v} - \rho\right), \boldsymbol{u} \in \mathbb{R}^{n_1}, \boldsymbol{v} \in \mathbb{R}^{n_2}, \tag{7}$$

where Φ is a nonlinear mapping function noted in Section 2.1. OCSTM can be denoted as the following optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{u},\boldsymbol{v},\rho,\xi} \frac{1}{2}\left\|\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}\right\|^2 - \rho + \frac{C}{2}\sum_{i=1}^{l}\xi_i, \\ &\text{s.t. } \boldsymbol{u}^{\mathrm{T}}\Phi(\mathbf{X}_i)\boldsymbol{v} \geq \rho - \xi_i, \\ &\quad \xi_i \geq 0, i = 1, 2, \cdots, l, \end{aligned} \tag{8}$$

where $\boldsymbol{u}$ and $\boldsymbol{v}$ can be obtained by iteratively solving the optimization problems.

Although OCSTM is a one-class classification method for tensor, solving optimization problems requires solving quadratic programming problems itera-

tively, so it is very time-consuming.

3. Least Square One-Class Support Tensor Machine

3.1. Overview

Based on the two existing methods in Section 2, we utilize the idea of LSOCSVM to improve OCSTM and propose a new tensor one-class classification method called LSOCSTM. Specifically, we change the loss term in the objective function to a least squares loss, which allows us to replace the QP problem in the solution process with solving a system of linear equations. We use an alternating iteration algorithm, which greatly reduces the number of decision variables. These features make LSOCSTM particularly suitable when it comes to small sample size problems.

3.2. Methodology

Suppose $\{\mathbf{X}_i : i = 1,2,\cdots,l\}$ is the training samples, where $\mathbf{X}_i \in \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2}$ which is the data point in 2nd-order tensor space. Based on this, LSOCSTM can be formulated as the following optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{u},\boldsymbol{v},\rho,\xi} \frac{1}{2}\left\|\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}\right\|^2 - \rho + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2, \\ &\text{s.t. } \boldsymbol{u}^{\mathrm{T}}\Phi\left(\mathbf{X}_i\right)\boldsymbol{v} = \rho - \xi_i, \end{aligned} \tag{9}$$

where C is a trade-off parameter between the margin maximization and empirical error minimization, ξ_i is a slack variable. Unlike the hinge loss function of the OCSTM, the loss in our optimization problem is the least squares loss, which is easier to solve. At the same time, like LSOCSVM, the restriction on slack variable $\xi_i \geq 0$ no longer exists. Optimization problem (9) tries to find a hyperplane which has the maximal distance $\frac{\rho}{\left\|\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}\right\|^2}$ from the origin, and with respect to this distance, ξ_i^2 are minimized.

To solve the optimization problem (9), we introduce Lagrangian multipliers $\alpha_i, i = 1,2,\cdots,l$ for these equality constrains. The Lagrangian function can be written as follows:

$$L\left(\boldsymbol{u},\boldsymbol{v},\rho,\xi,\alpha\right) = \frac{1}{2}\left\|\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}\right\|^2 - \rho + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 - \sum_{i=1}^{l}\alpha_i\left(\boldsymbol{u}^{\mathrm{T}}\Phi\left(\mathbf{X}_i\right)\boldsymbol{v} - \rho + \xi_i\right). \tag{10}$$

Setting the derivatives with respect to the primal variables $\boldsymbol{u},\boldsymbol{v},\rho,\xi$ equal to zero, we have

$$\frac{\partial L}{\partial \boldsymbol{u}} = 0 \Rightarrow \boldsymbol{u} = \frac{1}{\left\|\boldsymbol{v}\right\|^2}\sum_{i=1}^{l}\alpha_i\Phi\left(\mathbf{X}_i\right)\boldsymbol{v}, \tag{11}$$

$$\frac{\partial L}{\partial \boldsymbol{v}} = 0 \Rightarrow \boldsymbol{v} = \frac{1}{\left\|\boldsymbol{u}\right\|^2}\sum_{i=1}^{l}\alpha_i\Phi\left(\mathbf{X}_i\right)^{\mathrm{T}}\boldsymbol{u}, \tag{12}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow \xi_i = \frac{\alpha_i}{C}, \tag{13}$$

$$\frac{\partial L}{\partial \rho}=0\Rightarrow\sum_{i=1}^{l}\alpha_i=1. \tag{14}$$

It can be seen from (11) and (12), $\boldsymbol{u}$ and $\boldsymbol{v}$ are interdependent and cannot be solved independently of each other. We develop alternating projection algorithm to solve our problem, and the specific solution procedure is as follows:

First, we fix $\boldsymbol{u}$ and then let $\mu_1=\|\boldsymbol{u}\|^2$. After that, we denote that $x_i=\Phi\left(\boldsymbol{X}_i\right)^{\mathrm{T}}\boldsymbol{u}$. Optimization problem (9) can be rewritten in the following form:

$$\begin{aligned}&\min_{\boldsymbol{v},\rho,\xi}\frac{1}{2}\mu_1\|\boldsymbol{v}\|^2-\rho+\frac{C}{2}\sum_{i=1}^{l}\xi_i^2\\&\text{s.t. }\boldsymbol{v}^{\mathrm{T}}x_i=\rho-\xi_i\end{aligned} \tag{15}$$

The Lagrangian function can be rewritten as follows:

$$L\left(\boldsymbol{v},\rho,\xi,\alpha\right)=\frac{1}{2}\boldsymbol{\mu}_1\left(\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}\right)-\rho+\frac{C}{2}\sum_{i=1}^{l}\xi_i^2-\sum_{i=1}^{l}\alpha_i\left(\boldsymbol{v}^{\mathrm{T}}x_i-\rho+\xi_i\right). \tag{16}$$

By setting the derivatives with respect to the primal variables $\boldsymbol{v},\rho,\xi,\alpha$ equal to zero, we have

$$\frac{\partial L}{\partial \boldsymbol{v}}=0\Rightarrow\boldsymbol{v}=\frac{1}{\|\boldsymbol{\mu}_1\|^2}\sum_{i=1}^{l}\alpha_i\Phi\left(\mathbf{X}_i\right)^{\mathrm{T}}\boldsymbol{u}, \tag{17}$$

$$\frac{\partial L}{\partial \xi_i}=0\Rightarrow\xi_i=\frac{\alpha_i}{C}, \tag{18}$$

$$\frac{\partial L}{\partial \rho}=0\Rightarrow\sum_{i=1}^{l}\alpha_i=1, \tag{19}$$

$$\frac{\partial L}{\partial \alpha_i}=0\Rightarrow\boldsymbol{v}^{\mathrm{T}}x_i-\rho+\xi_i=0. \tag{20}$$

Substituting (17) and (18) into (20), we get

$$\frac{1}{\mu_1}\sum_{j=1}^{l}\alpha_j\boldsymbol{u}^{\mathrm{T}}K\left(\mathbf{X}_j,\mathbf{X}_i\right)\boldsymbol{u}-\rho+\frac{\alpha_i}{C}=0. \tag{21}$$

Coupled with the constraint in (19), equations in (21) can be reduced to the following system of linear equations to solve:

$$\begin{bmatrix}0 & \boldsymbol{e}^{\mathrm{T}}\\ \boldsymbol{e} & \dfrac{\boldsymbol{M}}{\mu_1}+\dfrac{\boldsymbol{I}}{C}\end{bmatrix}\begin{bmatrix}-\rho\\ \boldsymbol{a}\end{bmatrix}=\begin{bmatrix}1\\ \mathbf{0}\end{bmatrix}, \tag{22}$$

where vectors $\boldsymbol{e}$ and $\mathbf{0}$ represent all one and all zero column vectors of l dimension, $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{a}$ is the column vector of Lagrangian multipliers $\left[\alpha_1,\cdots,\alpha_l\right]$ and $\boldsymbol{M}$ denotes a new matrix which $\boldsymbol{M}_{ij}=\boldsymbol{u}^{\mathrm{T}}K\left(\mathbf{X}_j,\mathbf{X}_i\right)\boldsymbol{u}$.

After simple algebraic operations, we can get

$$\boldsymbol{a}^*=\frac{\left(\dfrac{\boldsymbol{I}}{C}+\dfrac{\boldsymbol{M}}{\mu_1}\right)^{-1}\boldsymbol{e}}{\boldsymbol{e}^{\mathrm{T}}\left(\dfrac{\boldsymbol{I}}{C}+\dfrac{\boldsymbol{M}}{\mu_1}\right)^{-1}\boldsymbol{e}}, \tag{23}$$

$$\boldsymbol{v} = \frac{1}{\boldsymbol{\mu}_1}\sum_{i=1}^{I}\alpha_i^*\Phi\left(\mathbf{X}_i\right)^{\mathrm{T}}\boldsymbol{u}, \tag{24}$$

$$\left\|\boldsymbol{v}\right\|^2 = \frac{1}{\boldsymbol{\mu}_1^2}\sum_{i,j=1}^{I}\alpha_i^*\alpha_j^*\boldsymbol{u}^{\mathrm{T}}K\left(\mathbf{X}_i,\mathbf{X}_j\right)\boldsymbol{u}. \tag{25}$$

Secondly, to calculate $\boldsymbol{u}$, let $\mu_2 = \left\|\boldsymbol{v}\right\|^2$ and $x_i' = \Phi\left(\mathbf{X}_i\right)\boldsymbol{v}$. The optimal problem for $\boldsymbol{u}$ can be constructed as follow:

$$\begin{aligned}&\min_{\boldsymbol{u},\rho,\xi}\frac{1}{2}\mu_2\left\|\boldsymbol{u}\right\|^2 - \rho + \frac{C}{2}\sum_{i=1}^{I}\xi_i^2,\\ &\text{s.t. } \boldsymbol{u}^{\mathrm{T}}x_i' = \rho - \xi_i.\end{aligned} \tag{26}$$

As with the arithmetic procedure from (15) to (21), we can get

$$\frac{1}{\mu_2}\sum_{j=1}^{I}\alpha_j x_j'^{\mathrm{T}}x_i' - \rho + \frac{\alpha_i}{C} = 0. \tag{27}$$

Similar to the treatment of (21), equations in (27) can be reduced as follow linear equation:

$$\begin{bmatrix}0 & \boldsymbol{e}^{\mathrm{T}}\\ \boldsymbol{e} & \dfrac{\boldsymbol{Q}}{\mu_2}+\dfrac{\boldsymbol{I}}{C}\end{bmatrix}\begin{bmatrix}-\rho\\ \boldsymbol{a}\end{bmatrix} = \begin{bmatrix}1\\ \boldsymbol{0}\end{bmatrix}, \tag{28}$$

where vectors $\boldsymbol{e}$ and $\boldsymbol{0}$ represent all one and all zero column vectors of I dimension, and $\boldsymbol{Q}$ denotes a new matrix which $\boldsymbol{Q}_{ij} = \boldsymbol{u}^{\mathrm{T}}\sum_{p=1}^{I}K\left(\mathbf{X}_p,\mathbf{X}_j\right)\sum_{q=1}^{I}K\left(\mathbf{X}_i,\mathbf{X}_q\right)\boldsymbol{u}$.

By solving the linear equation in (28), we can get

$$\hat{\boldsymbol{a}} = \frac{\left(\dfrac{\boldsymbol{I}}{C}+\dfrac{\boldsymbol{Q}}{\mu_2}\right)^{-1}\boldsymbol{e}}{\boldsymbol{e}^{\mathrm{T}}\left(\dfrac{\boldsymbol{I}}{C}+\dfrac{\boldsymbol{Q}}{\mu_2}\right)^{-1}\boldsymbol{e}}, \tag{29}$$

and then we can update $\boldsymbol{u}_{new}$ by

$$\boldsymbol{u}_{new} = \frac{1}{\mu_2}\sum_{i=1}^{I}\hat{\alpha}_i\Phi\left(\mathbf{X}_i\right)\boldsymbol{v} = \frac{1}{\mu_1\mu_2}\sum_{i,j=1}^{I}\hat{\alpha}_i\alpha_j^*K\left(\mathbf{X}_i,\mathbf{X}_j\right)\boldsymbol{u}. \tag{30}$$

Thus, $\boldsymbol{u}$ and $\boldsymbol{v}$ can be obtained iteratively. We mention training samples with non-zero α_i^* are support tensors, labeled as $\mathbf{X}_{i\_s}$. Therefore, ρ and $f\left(\mathbf{X}\right)$ can be calculated as follow:

$$\rho = mean_{i\_s}\left(\frac{1}{\boldsymbol{\mu}_1}\sum_{i=1}^{I}\alpha_i^*\boldsymbol{u}^{\mathrm{T}}K\left(\mathbf{X}_i,\mathbf{X}_{i\_s}\right)\boldsymbol{u}\right) \tag{31}$$

$$f\left(\mathbf{X}\right) = \frac{1}{\boldsymbol{\mu}_1}\sum_{i=1}^{I}\alpha_i^*\boldsymbol{u}^{\mathrm{T}}K\left(\mathbf{X},\mathbf{X}_i\right)\boldsymbol{u} - \rho \tag{32}$$

The maximum value of $f\left(\mathbf{X}_{i\_s}\right)$ is the threshold for distinguishing whether it is a target class or not. Substitute the test sample $\mathbf{X}$ into f. If it is less than this maximum value, the test sample belongs to the target class.

Substitute f for $\mathbf{X}$ in the training set and compare the minimum values of

these f with f in the test set.

LSOCSTM is a new tensor based one-class classification method that has two advantages. One is that it can directly use tensor data as input to avoid structural information loss and overfitting problems caused by vectorization. Secondly, it replaces the loss function with the least squares loss, thereby transforming the optimization problem into solving a system of linear equations, greatly reducing the complexity and time consumption of operations.

3.3. Convergence

Lemma 1. The iterative process of the optimization problems (15) and (26) will make $\frac{1}{2}\left\|\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}\right\|^2-\rho+\frac{C}{2}\sum_{i=1}^{l}\xi_i^2$ in the optimization problem (9) monotonically decreasing, and thus the LSOCSTM algorithm converges.

Proof. Define the function as follows:

$$f\left(\boldsymbol{u},\boldsymbol{v}\right)=\left\|\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}\right\|^2-\rho+\frac{C}{2}\sum_{i=1}^{l}\xi_i^2 \tag{33}$$

Since LSOCSVM is convex optimization, the optimization problems (15) and (26) are also convex optimization problems. Therefore, their solutions are globally optimal. Assume that the initial value of $\boldsymbol{u}$ is $\boldsymbol{u}_0$, and solve the optimization problem (15) to obtain the initial value of $\boldsymbol{v}$ as $\boldsymbol{v}_0$, and then solve the optimization problem (26) to obtain the solution $\boldsymbol{u}_1$ for one iteration. Thus we can obtain:

$$f\left(\boldsymbol{u}_0,\boldsymbol{v}_0\right)\geq f\left(\boldsymbol{u}_1,\boldsymbol{v}_0\right) \tag{34}$$

Repeating the above procedure gives:

$$f\left(\boldsymbol{u}_0,\boldsymbol{v}_0\right)\geq f\left(\boldsymbol{u}_1,\boldsymbol{v}_0\right)\geq f\left(\boldsymbol{u}_1,\boldsymbol{v}_1\right)\geq f\left(\boldsymbol{u}_2,\boldsymbol{v}_1\right)\geq\cdots \tag{35}$$

And since f is constantly greater than or equal to 0, f converges.

4. Experimental Results

4.1. Datasets and Parameters

In this section, we present our experimental results and compare the performance of least squares one-class support tensor machine (LSOCSTM) with three methods: OCSTM, LSOCSVM and OCSVM. All of our datasets are downloaded from UCI repository. Parameter m presents the total sample size, n presents the dimensionality of the vector sample and $n_1\times n_2$ shows the tensor size of our datasets. For each dateset, features are scaled to [−1, 1]. Since we consider one-class classification problem, our focus is only on the one target class. **Table 1** summarizes the information of the datasets.

We choose Radial Basis Function (RBF) kernel function for the proposed method since it has been proven that RBF kernel function performs better than the other kernels [24]. For each classification problem, ten independent runs are performed and the average classification accuracy (ACC) and AUC [25] on the

Table 1. Information of datasets.

Dataset	m	n	$n_1 \times n_2$	Class	Sample size
BREAST-CANCER	286	9	3 × 3	1	201
				2	85
IRIS	150	4	2 × 2	1	50
				2	50
SONAR	208	60	8 × 8	1	97
				2	111
IONOSPHERE	351	34	6 × 6	1	225
				2	126
USPS	300	256	16 × 16	1	100
				2	100
Letter Recognition	155	16	4 × 4	1	79

sets can be calculated and then averaged. It is well known that tensor methods have obvious advantages when dealing with small sample problems [26], thus, our training sample size l is set to two cases: $l=3$, $l=6$. All the algorithms have been implemented in MATLAB R2019b on Windows 11 running on a PC with system configuration AMD Ryzen 7 5800H (3.2 GHz) and 16 GB of RAM.

4.2. Sensitivity Analysis

4.2.1. Hyperparametric Sensitivity Analysis

There is only one hyper-parameter in our work, so for each test, we train every machine by choosing C from {0.001, 0.01, 0.1, 1, 10, 100} for 3 times, and then the best C for every test could be found. In **Table 2**, for example, in Iris Dataset Class 1, 0.01 is the optimal value.

4.2.2. Tensor Size Sensitivity Analysis

In this subsection, we focus on the size selection of tensors. For example, for each sample $x \in \mathbb{R}^n$, a suitable tensor size needs to be found. In our work, we transform the vectors into second-order tensors $X \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ ($n_1 \times n_2 \approx n$). Cai [21] provided a method to establish n_1 and n_2, which was simplified as minimizing $n_1 + n_2$ while $(n_1 - 1) \times n_2 \le n \le n_1 \times n_2$. Under such conditions, there are still many choices of n_1 and n_2. For example, in Ionosphere Dataset, there are 5 choices of sizes available. Generally all these types can be used. Therefore, it is worth finding out which one is the best. **Table 3** below shows the performance of these 5 choices. These experiments suggest that, n_1 and n_2 should be as close as possible. In the meanwhile, $n_1 \le n_2$ can be a good choice and it is particularly suitable for small sample size problem.

4.2.3. Convergence Test

Figure 1 is the 2-norm of the differences of $\boldsymbol{u}$ between two adjacent iterations from Sonar Dataset Class 1. It can be seen that after 20 iterations, the value converges to 0. This also verifies that Lemma 1 is valid.

Table 2. Accuracy and ACC in different parameter C from iris dataset class 1 when $I = 3$.

C	ACC	AUC
0.001	0.918	0.9316
0.01	0.923	0.9387
0.1	0.911	0.9323
1	0.897	0.9236
10	0.895	0.9191

Table 3. Accuracy and ACC in different tensor size from ionosphere dataset class 1.

Size	ACC	AUC
2×18	0.6618	0.7815
4×9	0.6681	0.7902
6×6	0.7025	0.8199
9×4	0.6427	0.6777
18×2	0.6356	0.6309

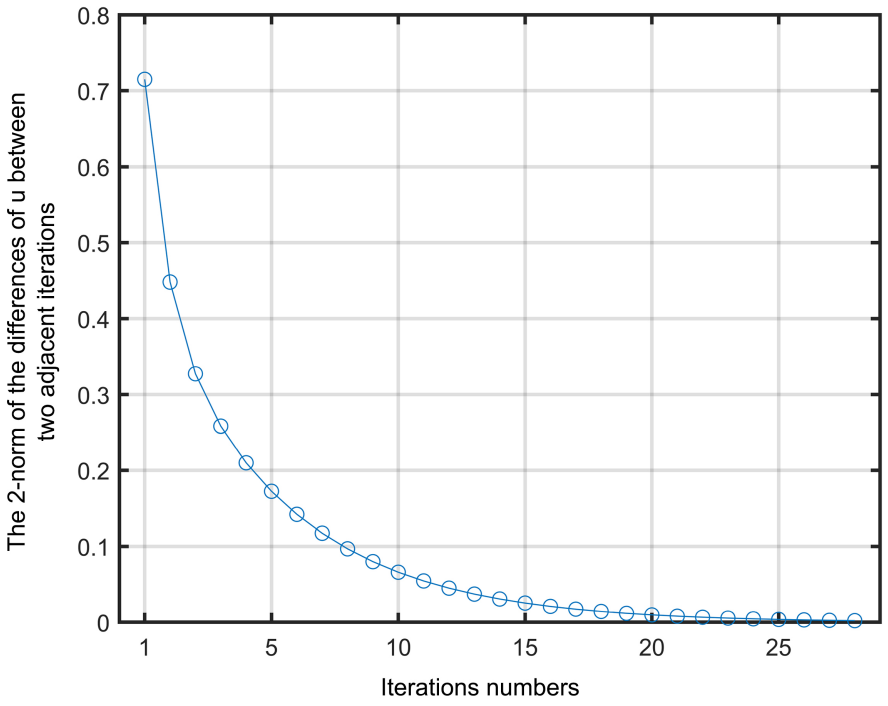

Figure 1. The 2-norm of the differences of u between two adjacent iterations from sonar dataset class 1.

4.3. Comparison Experiments

In this subsection, we compared our method with the three methods mentioned in the first subsection. The performance of the four methods with training set sample sizes of 3 and 6 is shown in the following Table 4 and Table 5.

Table 4. Accuracy and ACC in different methods when $l = 3$.

2* Datasets	2* Target Class	LSOCSTM		OCSTM		LSOCSVM		OCSVM	
		ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
IRIS	1	**0.923**	**0.9387**	0.881	0.9078	0.9104	0.9165	0.9022	0.9028
	2	**0.8361**	**0.8813**	0.822	0.8661	0.8331	0.8762	0.8327	0.8751
BREAST-CANCER	1	**0.6552**	**0.6011**	0.6291	0.5833	0.6489	0.5918	0.6042	0.5746
	2	**0.6972**	**0.7167**	0.6811	0.7012	0.6901	0.7118	0.6844	0.7086
SONAR	1	**0.6298**	**0.6557**	0.5865	0.6423	0.5947	0.6441	0.5812	0.6397
	2	0.5337	0.7668	**0.5596**	**0.7812**	0.5218	0.7421	0.5315	0.7512
IONOSPHERE	1	**0.7025**	**0.8132**	0.6615	0.7961	0.6487	0.7772	0.6348	0.7661
	2	**0.6387**	**0.6789**	0.5256	0.5667	0.5987	0.6214	0.6061	0.6217
USPS	1	**0.9223**	**0.9467**	0.9128	0.9378	0.8992	0.9216	0.9026	0.9274
	2	0.9358	0.9552	**0.9366**	**0.9598**	0.9085	0.9377	0.9102	0.9401
Letter Recognition	1	**0.7390**	**0.7761**	0.7168	0.7561	0.7069	0.7372	0.7143	0.7278

Table 5. Accuracy and ACC in different methods when $l = 6$.

2* Datasets	2* Target Class	LSOCSTM		OCSTM		LSOCSVM		OCSVM	
		ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
IRIS	1	**0.97**	**0.9723**	0.944	0.9483	0.956	0.9608	0.953	0.9512
	2	0.8711	0.9013	0.8590	0.8827	**0.8733**	**0.9036**	0.8691	0.8801
BREAST-CANCER	1	0.6941	0.6064	0.6796	0.6246	0.6881	0.5971	**0.6957**	**0.6090**
	2	0.7227	0.7389	**0.7286**	**0.7442**	0.7201	0.7298	0.7115	0.7213
SONAR	1	**0.6298**	**0.6557**	0.5865	0.6109	0.6123	0.6513	0.6112	0.6291
	2	0.5368	0.7701	**0.5571**	**0.7793**	0.5316	0.7613	0.5328	0.7591
IONOSPHERE	1	**0.6998**	**0.7984**	0.6716	0.7893	0.6553	0.7712	0.6524	0.7698
	2	**0.6437**	**0.6869**	0.5443	0.5874	0.6073	0.6354	0.5551	0.5982
USPS	1	**0.9386**	**0.9558**	0.9354	0.9493	0.9176	0.9325	0.9092	0.9301
	2	**0.9578**	**0.9677**	0.9411	0.9712	0.9228	0.9496	0.9242	0.9502
LETTER RECOGNITION	1	**0.7688**	**0.7879**	0.7598	0.7764	0.7284	0.7453	0.7159	0.7332

Table 6. Time (in second) of different methods when $I = 3$.

Datasets	Target Class	LSOCSTM	OCSTM	LSOCSVM	OCSVM
IRIS	1	11	196	2	8
	2	12	189	2	7
BREAST-CANCER	1	36	120	4	10
	2	37	128	4	10
SONAR	1	139	280	8	14
	2	144	281	11	15
IONOSPHERE	1	186	393	10	16
	2	178	366	11	16
USPS	1	774	2412	21	37
	2	789	2433	22	39
LETTER RECOGNITION	1	25	108	4	9

From the two tables, we can find that our method has the best results in most cases. Meanwhile, the two tensor classification methods are significantly better than the vector methods. From the tendency of the averaged test accuracy (ACC) and AUC, we can conclude that tensor based classifiers LSOCSTM and OCSTM have significant advantages over all vector based classifiers in all datasets, especially for small training sample size cases. It also supports the conclusion that tensor-based methods are suitable for dealing with small sample problems.

Table 6 summarized the time required for each method. From Table 6, it can be found that the proposed method has a significant advantage over the tensor-based method OCSTM in terms of time duration. Although the proposed method takes more time than the vector-based method due to iteration and computation involved kernel functions, the accuracy of classification is also significantly higher than theirs.

5. Conclusion

This article combines the two one-class classification methods LSOCSVM and OCSTM, and then proposes a new one-class classification method LSOCSTM for tensors. It is verified that tensor methods perform better than vector methods in general, and tensor methods are more suitable for small sample size one-class classification problems. Compared with the existing tensor methods, the proposed method extracts the hyperplane where most of the training objects may lie nearby, while OCSTM extracts the hyperplane where most of the training objects may cross. By using least squares loss, the optimization process for the proposed method no longer requires solving quadratic programming problems as in OCSTM, but rather a system of linear equations. Therefore, the pro-

posed method is faster and more efficient compared to OCSTM. A possible future research direction is to extend the proposed method to deal with one-class classification for third-order or higher order tensor datasets. In addition, when datasets contain outliers, the improvement of robustness is also another future research direction.

Declarations

This work was supported by the National Natural Science Foundation of China (Grant No.12371308).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Vapnik, V. (1999) The Nature of Statistical Learning Theory. Springer Science & Business Media, Berlin.

[2] Zhu, W. and Zhong, P. (2014) A New One-Class SVM Based on Hidden Information. *Knowledge-Based Systems*, **60**, 35-43. https://doi.org/10.1016/j.knosys.2014.01.002

[3] Manevitz, L.M. and Yousef, M. (2001) One-Class SVMs for Document Classification. *Journal of Machine Learning Research*, **2**, 139-154.

[4] Maboudou-Tchao, E.M. (2020) Change Detection Using Least Squares One-Class Classification Control Chart. *Quality Technology & Quantitative Management*, **17**, 609-626. https://doi.org/10.1080/16843703.2019.1711302

[5] Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C., *et al.* (1999) Estimating the Support of a High-Dimensional Distribution. Technical Report MSR-T R-99-87, Microsoft Research.

[6] Tax, D.M. and Duin, R.P. (1999) Support Vector Domain Description. *Pattern Recognition Letters*, **20**, 1191-1199. https://doi.org/10.1016/S0167-8655(99)00087-2

[7] Ghaoui, L.E., Jordan, M.I. and Lanckriet, G.R. (2003) Robust Novelty Detection with Single-Class MPM. *Advances in Neural Information Processing Systems*, **15**, 929-936.

[8] Wang, J. and Cherian, A. (2019) Gods: Generalized One-Class Discriminative Subspaces for Anomaly Detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, 27 October-2 November 2019, 8200-8210. https://doi.org/10.1109/ICCV.2019.00829

[9] Choi, Y.S. (2009) Least Squares One-Class Support Vector Machine. *Pattern Recognition Letters*, **30**, 1236-1240. https://doi.org/10.1016/j.patrec.2009.05.007

[10] Deivendran, P., Kumar, N.J., Yashwanth, R., Raghul, R. and Naresh, D. (2023) Predict the Price in Stock Market Based on Heuristic Search Techniques Using Tensor Representation. 2023 *International Conference on Networking and Communications* (*ICNWC*), Chennai, 5-6 April 2023, 1-6. https://doi.org/10.1109/ICNWC57852.2023.10127251

[11] Renard, P. and Ababou, R. (2022) Equivalent Permeability Tensor of Heterogeneous Media: Upscaling Methods and Criteria (Review and Analyses). *Geosciences*, 12,

Article 269. https://doi.org/10.3390/geosciences12070269

[12] Cover, T.M. (1965) Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *IEEE Transactions on Electronic Computers*, **EC-14**, 326-334. https://doi.org/10.1109/PGEC.1965.264137

[13] Tao, D., Li, X., Hu, W., Maybank, S. and Wu, X. (2005) Supervised Tensor Learning. *Fifth IEEE International Conference on Data Mining* (*ICDM'05*), Houston, 27-30 November 2005, 8.

[14] Zhou, H., Li, L. and Zhu, H. (2013) Tensor Regression with Applications in Neuroimaging Data Analysis. *Journal of the American Statistical Association*, **108**, 540-552. https://doi.org/10.1080/01621459.2013.776499

[15] Hao, Z., He, L., Chen, B. and Yang, X. (2013) A Linear Support Higher-Order Tensor Machine for Classification. *IEEE Transactions on Image Processing*, **22**, 2911-2920. https://doi.org/10.1109/TIP.2013.2253485

[16] Signoretto, M., De Lathauwer, L. and Suykens, J.A. (2011) A Kernel-Based Framework to Tensorial Data Analysis. *Neural Networks*, **24**, 861-874. https://doi.org/10.1016/j.neunet.2011.05.011

[17] Zhao, Q., Zhou, G., Adal, T., Zhang, L. and Cichocki, A. (2013) Kernel-Based Tensor Partial Least Squares for Reconstruction of Limb Movements. 2013 *IEEE International Conference on Acoustics*, *Speech and Signal Processing*, Vancouver, 26-31 May 2013, 3577-3581. https://doi.org/10.1109/ICASSP.2013.6638324

[18] Gao, C. and Wu, X.J. (2012) Kernel Support Tensor Regression. *Procedia Engineering*, **29**, 3986-3990. https://doi.org/10.1016/j.proeng.2012.01.606

[19] He, L., Lu, C.T., Ma, G., Wang, S., Shen, L., Philip, S.Y. and Ragin, A.B. (2017) Kernelized Support Tensor Machines. *Proceedings of the* 34*th International Conference on Machine Learning*, Sydney, 6-11 August 2017, 1442-1451.

[20] Chen, C., Batselier, K., Yu, W. and Wong, N. (2022) Kernelized Support Tensor Train Machines. *Pattern Recognition*, **122**, Article ID: 108337. https://doi.org/10.1016/j.patcog.2021.108337

[21] Cai, D., He, X. and Han, J. (2006) Learning with Tensor Representation. Technical Report, Department of Computer Science, University of Illinois, 2006. UIUCDCSR-2006-2716.

[22] Chen, Y., Wang, K. and Zhong, P. (2016) One-Class Support Tensor Machine. *Knowledge-Based Systems*, **96**, 14-28. https://doi.org/10.1016/j.knosys.2016.01.007

[23] Maboudou-Tchao, E.M. (2021) Support Tensor Data Description. *Journal of Quality Technology*, **53**, 109-134. https://doi.org/10.1080/00224065.2019.1642815

[24] Maji, S., Berg, A.C. and Malik, J. (2012) Efficient Classification for Additive Kernel SVMs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**, 66-77. https://doi.org/10.1109/TPAMI.2012.62

[25] Airola, A., Pahikkala, T., Waegeman, W., De Baets, B. and Salakoski, T. (2011) An Experimental Comparison of Cross-Validation Techniques for Estimating the Area under the Roc Curve. *Computational Statistics & Data Analysis*, **55**, 1828-1844. https://doi.org/10.1016/j.csda.2010.11.018

[26] Guo, W., Kotsia, I. and Patras, I. (2011) Tensor Learning for Regression. *IEEE Transactions on Image Processing*, **21**, 816-827. https://doi.org/10.1109/TIP.2011.2165291