Chemometrics and Intelligent Laboratory Systems 151 (2016) 15–25

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemolab

Robust one-class SVM for fault detection

Yingchao Xiao [a], Huangang Wang [a,*], Wenli Xu [a], Junwu Zhou [b]

[a] Institute of Control Theory and Technology Department of Automation, Tsinghua University, Beijing 100084, China
[b] State Key Laboratory of Automation of Mining & Metallurgy Process, Beijing 100160, China

ARTICLE INFO

Article history:
Received 11 June 2015
Received in revised form 20 October 2015
Accepted 14 November 2015
Available online 9 December 2015

Keywords:
Robust to outliers
One-class SVM
One-class classification
Fault detection
Tennessee Eastman Process

ABSTRACT

One-class SVM (OCSVM) has been widely adopted in many one-class classification (OCC) application fields. However, when there are outliers in OCC training samples, the OCSVM performance will degrade. In order to solve this problem, a new method is proposed in this paper. This method first identifies some "suspected outliers" and removes them so as to obtain the decision boundary enclosing the "cluster core". Then outliers are identified by this boundary and are removed from OCSVM training. The effectiveness of this proposed method is verified by experiments on UCI benchmark data sets and Tennessee Eastman Process data sets.

1. Introduction

In recent years, data-driven fault detection has received increasing attention [1–3]. In the process of chemical industry, the samples under the normal operation condition are easy to obtain and are in large quantities, but the fault samples are expensive to obtain and are very rare. In this situation, the conventional binary or multiclass classification methods are not suitable for fault detection, and the one-class classification methods can be used. One-class classification (OCC) tries to describe the distribution area of the target class in the condition that only target samples are available for training, so as to predict whether a new sample belongs to the target class or not [4,5]. If the OCC methods, one-class support vector machine (OCSVM) uses the kernel trick to deal with nonlinearity, and its decision function is sparse in the number of support vectors. Therefore, it is widely adopted [6–8].

Due to its potential application, OCSVM has received considerable studies recently. For example, Liu et al. [9] assumed that the training samples are uncertain, so they proposed to alternately train the OCSVM models and optimize the location of each sample, so as to improve the ability of OCSVM to deal with uncertain data. Khan et al. [10] pointed out that the low variance directions in the training data carry crucial information for OCSVM. So they introduced into the OCSVM model a term on covariance matrix of the training data, moving the resultant hyperplane toward the low variance directions. In order to overcome the limits caused by a single OCSVM classifier, Krawczyk et al. [11] proposed to assemble OCSVM classifiers. They first split the target class into several clusters, then built the OCSVM models based on each cluster, and fused their decisions at last. To further prune the classifier ensemble, they proposed several criteria to measure diversity of classifiers [12] and proposed to apply a metaheuristic optimization procedure to prune and weight OCC ensembles [13].

To better describe the target class area, it is expected that the target samples are sufficient and representative. However, in practice, target samples often contain a few outliers, which distribute differently from the majority of the target class but they are labeled as target [14,15]. They can be viewed as the counterpart of label noise in binary classification [16]. Outliers in the target samples will negatively affect the performance of OCC methods. Liu et al. [17] proposed an unsupervised method to divide the corrupted training samples into two classes (inliers and outliers) and then to clean up the outliers. As for the OCSVM method, it determines its decision function only by some support vectors; therefore, its decision boundary is apt to be affected by outliers [18,19].

To reduce the negative influence of outliers on OCSVM decision boundary, Yin et al. [18] proposed to weight training samples according to their distances to the center in the feature space. They first calculate the total square loss center [20] of the training samples and the distance d_i from this center to each sample $\mathbf{x}_i$. Then the weight $\hat{d}_i$ related to d_i is obtained by $\hat{d}_i = d_{i,\max}/d_i$, where $d_{i,\max}$ is the maximal distance. In this way, smaller weights are expected to be given to outliers, so that less punishment is triggered by excluded outliers, making it more likely to exclude outliers. This method will be referred to as wOCSVM hereinafter. However, the outlier distribution area cannot be available in advance, so it is difficult to give samples proper weights. Amer et al.

* Corresponding author. Tel.: +86 10 62781993; fax: +86 10 62786911.
E-mail addresses: xiaoyc10@mails.tsinghua.edu.cn (Y. Xiao), hgwang@tsinghua.edu.cn (H. Wang), xuwl@tsinghua.edu.cn (W. Xu), zhou_jw@bgrimm.com (J. Zhou).

http://dx.doi.org/10.1016/j.chemolab.2015.11.010
0169-7439/

[19] also tried to reduce outliers' negative influence by introducing weights into OCSVM. They used the weights to modify the constraints that samples should be located above the OCSVM hyperplane. However, this modification is not proper. It makes the hyperplane located inappropriately, labeling many target training samples as negative. Therefore, this method is not as reasonable as wOCSVM. Different from the above method, which first weights samples and then trains OCSVM, Amer et al. [19] proposed to find out the outliers during optimizing the OCSVM hyperplane. They modified the optimization objective of OCSVM by introducing a 0-1 variable η_i for each training sample, using this variable to indicate whether $\mathbf{x}_i$ is an outlier or not. The introduction of this discrete variable makes it very difficult to solve the optimization. So they relaxed the optimization problem and presented an iterative algorithm. In each iteration, the sample with $\eta_i = 1$ is constrained to be located above the OCSVM hyperplane, and the sample with $\eta_i = 0$ is not used for training this time. This method will be referred to as etaOCSVM hereinafter. If η_i is not able to indicate outliers correctly in some iteration, then the unidentified outliers would negatively affect the result of this iteration and subsequent iterations. Therefore, this method does not effectively reduce outliers' influence as expected.

In this paper, a new method is proposed to reduce the negative influence of outliers on OCSVM decision boundary. In this method, the OCSVM parameter υ is adjusted systematically to select samples for next round training so that the cluster core of training samples can be enclosed by the OCSVM decision boundary. Then outliers are identified by this decision boundary and excluded from the final OCSVM training; therefore, their influence is reduced.

The remainder of this paper is arranged as follows: the second section states the motivation; the third section proposes the new method; the fourth section compares the proposed method with other relevant methods on outlier contaminated data sets and on fault detection; the last section concludes this paper.

2. Motivation

OCSVM tries to separate the training samples from the origin in the feature space using a hyperplane $<\mathbf{w}, \varphi(\mathbf{x})> - \rho = 0$ with maximum margin, where $\mathbf{x}$ denotes a sample and $\varphi(\mathbf{x})$ denotes its image in the feature space. $\mathbf{w}$ and ρ are the normal vector and offset of the hyperplane, respectively. The OCSVM optimization problem is written as follows [21]:

$$\begin{aligned} \min_{\mathbf{w},\xi,\rho} \quad & \frac{1}{2}\|\mathbf{w}\|^2 - \rho + \frac{1}{\upsilon n}\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & \langle \mathbf{w}, \varphi(\mathbf{x}_i)\rangle \ge \rho - \xi_i, \\ & \xi_i \ge 0, i = 1, \cdots, n \end{aligned} \tag{1}$$

where $\mathbf{x}_i$ denotes training samples, n is the total number of training samples, υ is a trade-off parameter, and ξ_i is the slack variable. This is a convex optimization problem and can be solved via its dual problem. The Lagrangian of (1s) is as follows:

$$\begin{aligned} L(\mathbf{w}, \xi, \rho, \alpha, \beta) = & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\upsilon n}\sum_{i=1}^{n}\xi_i - \rho - \sum_{i=1}^{n}\beta_i\xi_i \\ & - \sum_{i=1}^{n}\alpha_i(\langle \mathbf{w}, \varphi(\mathbf{x}_i)\rangle - \rho + \xi_i) \end{aligned} \tag{2}$$

where $\alpha_i, \beta_i \ge 0$ are the multipliers. By setting the derivatives of the Lagrangian with respect to the primal variables to zero, we obtain the following equations:

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{n}\alpha_i\varphi(\mathbf{x}_i) \tag{3}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow \alpha_i + \beta_i = \frac{1}{\upsilon n} \tag{4}$$

$$\frac{\partial L}{\partial \rho} = 0 \Rightarrow \sum_{i=1}^{n}\alpha_i = 1 \tag{5}$$

After substituting Eqs. (3)–(5) into Eq. (2), we have the dual problem as shown in Eq. (6):

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & 0 \le \alpha_i \le \frac{1}{\upsilon n} \\ & \sum_{i=1}^{n}\alpha_i = 1 \end{aligned} \tag{6}$$

where $k(\mathbf{x}_i, \mathbf{x}_j) = <\varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j)>$ is a kernel function. After the optimal solution α is obtained, the constant ρ can be calculated by $\rho = <\mathbf{w}, \varphi(\mathbf{x}_i)>$, where $\mathbf{x}_i$ is some sample whose corresponding $\alpha_i \in (0, 1/\upsilon n)$. Then the OCSVM decision function $f(\mathbf{x}) = \sum_{i=1}^{n}\alpha_i k(\mathbf{x}_i, \mathbf{x}) - \rho$ is obtained, and the decision boundary is $f(\mathbf{x}) = 0$. If a sample $\mathbf{x}_B$ lies outside of the decision boundary, i.e., $f(\mathbf{x}_B) < 0$, its $\xi_B > 0$. According to the KKT conditions, $\beta_B = 0$, so $\alpha_B = 1/n\upsilon$. That is, the sample lying outside of the OCSVM decision boundary becomes a support vector (SV), and its $\alpha_B = 1/n\upsilon$.

As for outliers in the training samples, they are defined in the literature as follows: "an outlier is an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data" [14]; "outliers are data that do not obey rules considered normal for the majority of the data elements" [15]. Outliers prevent the OCSVM model from properly describing the distribution area of training samples. In order to obtain a more general model and tolerate a certain fraction of outliers, OCSVM allows some training samples to be located outside of its decision boundary, and uses the parameter υ to adjust the number of these outside samples [22].

However, it is shown in Eq. (3) that the normal vector of OCSVM hyperplane $\mathbf{w}$ is determined by the linear combination of the mappings of SVs. If an outlier is located outside the OCSVM decision boundary, it becomes an SV and thus affects the normal vector $\mathbf{w}$. Moreover, the value of α_i corresponding to this outlier is $1/n\upsilon$, the maximal value of the feasible region for α_i. This means the outliers do affect the hyperplane.

An example is used here to illustrate this shortcoming. First, 100 samples distributed in an "Ellipse" area are given. An OCSVM model is trained on these samples, where the widely used Gaussian kernel is adopted and its width parameter is set to $s = 4.2$ properly. No outliers exist in the training samples now, so υ is set to $\upsilon = 0.01$ to exclude no sample. The OCSVM decision boundary is shown as the solid curve in Fig. 1(a), and it describes the sample area properly. In Fig. 1, the solid points denote the training samples, and the circles and the squares denote the support vectors right on the boundary (boundary support vectors) and those outside the boundary (non-boundary support vectors), respectively. Next, 5 sparse-distributed samples on the right of this "Ellipse" are added into the training samples, and they are outliers to those original samples. If υ is still set to $\upsilon = 0.01$ in this situation, then the decision boundary in Fig. 1(b) is obtained. It is obvious that this boundary is negatively affected by the outliers, unable to properly describe the "Ellipse". To avoid this situation, υ should be increased to exclude more training samples. υ is increased to larger values, $\upsilon = 0.1$ and $\upsilon = 0.2$, and the corresponding decision boundaries are shown in Fig. 1(c) and (d), respectively. In Fig. 1(c), 8 training samples are excluded out of the boundary as non-boundary support vectors, but they do not contain all the 5 outliers, so the outliers still affect the boundary. In Fig. 1(d), all the 5 outliers become non-boundary support vectors, located outside the boundary, but they still affect the boundary, drawing