Neurocomputing 189 (2016) 1–10

Contents lists available at ScienceDirect

Neurocomputing

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

A weighted one-class support vector machine

Fa Zhu [a,*], Jian Yang [a], Cong Gao [b], Sheng Xu [c], Ning Ye [d], Tongming Yin [e]

[a] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, PR China
[b] Department of Computer Science, University of Regina, Canada
[c] Department of Geomatics Engineering, University of Calgary, Canada
[d] College of Information Science and Technology, Nanjing Forestry University, Nanjing, PR China
[e] College of Forest Resources and Environment, Nanjing Forestry University, PR China

ARTICLE INFO

Article history:
Received 20 May 2015
Received in revised form
28 September 2015
Accepted 2 October 2015
Communicated by Dr. Haiqin Yang
Available online 21 February 2016

Keywords:
Weighted one-class support vector machine
One-class classification
Neighbors' distribution knowledge
Instance weights

ABSTRACT

The standard one-class support vector machine (OC-SVM) is sensitive to noises, since every instance is equally treated. To address this problem, the weighted one-class support vector machine (WOC-SVM) was presented. WOC-SVM weakens the impact of noises by assigning lower weights. In this paper, a novel instance-weighted strategy is proposed for WOC-SVM. The weight is only relevant to the neighbors' distribution knowledge, which is only decided by k-nearest neighbors. The closer to the boundary of the data distribution the instance is, the lower the corresponding weight is. The experimental results demonstrate that WOC-SVM outperforms the standard OC-SVM when using the proposed instance-weighted strategy. The proposed instance-weighted method performs better than previous ones.

1. Introduction

Support vector machines (SVMs) were first proposed by Vapnik and his colleagues in the mid-1990s [1,2]. The primal SVMs only focuses on two-class classification problem. When the problem is imbalance, the performance deteriorates. One of the strategies to handle unbalanced problems is to adopt one-class classifier instead. The one-class version of SVMs was proposed in [3] and had succeeded in handwritten signature verification, information retrieval, disease diagnosis, remote-sensing, image retrieval, document classification and so on [4–10]. Inspired by one-class support vector machine (OC-SVM), Tax et al. proposed a similar algorithm, named support vector data description (SVDD), whose task was to find a hyper-sphere to include all targets [11]. In this paper, we only focus on OC-SVM.

OC-SVM inherits the merits of the SVMs: implementing the Structural Risk Minimization (SRM), unique global solution, sparsity of the solution etc. [1,2,12]. But it also retains the disadvantages of the SVMs: not considering the prior knowledge of the training set, needing a good kernel function for mapping data into a linear separable space, and requiring solving quadratic programming (QP), which is time consuming. In order to avoid solving QP problem, least-squares support vector machines (LS-SVMs) related algorithms were proposed [13–15]. In LS-SVMs, we only need to solve a linear programming rather than QP problem. However, LS-SVMs also lose the sparsity of the solution. The solution is decided by all instances rather than minor support vectors. In this paper, we only focus on how to utilize the prior knowledge. One way to utilize prior knowledge is to assign training instances with different weights, termed as weighted one-class support vector machine (WOC-SVM). We propose a novel instance-weighted strategy for WOC-SVM. The weight is only relevant to the neighbors' distribution knowledge, which is firstly proposed in [16,17]. The neighbors' distribution knowledge is only decided by the instance's k-nearest neighbors. Therefore, it is very simple to calculate. The rest of this paper is organized as follows. The related work is reviewed in Section 2. The basic review of the WOC-SVM is summarized in Section 3. A new instance-weighted strategy is introduced in Section 4. The experiments on artificial synthetic problems and benchmarks datasets are reported in Section 5. The discussion and conclusions are provided in the last section.

2. Related work

How to improve the performance of SVMs via introducing prior knowledge is a hot topic in the community of machine learning and pattern recognition. One way to incorporate prior knowledge into SVMs is to assign the training data with different weights. The side effect of the noises can be weakened by assigning them with lower weights. Thus, SVMs could become more robust to noises.

* Corresponding author.
E-mail address: zhufag@gmail.com (F. Zhu).

http://dx.doi.org/10.1016/j.neucom.2015.10.097
0925-2312/

The previous related work mainly focuses on two-class classification problem [18–21].

The weighted version of OC-SVM and SVDD were firstly proposed by Bicego and Figueiredo in [22]. They utilized WOC-SVM to do soft clustering. In their algorithm, the WOC-SVM was regarded as the similarity between the sample and cluster center. Zhang et al. proposed to utilize kernel possibilistic c-means clustering (KPCM) to assign weights for weighted support vector data description (WSVDD) [23]. Since SVDD could contain negative instances, they also assigned negative instances weights. In [10], Cyganek proposed to utilize fuzzy c-means algorithm to assign weights for the ensemble one-class support vector machines. The weights were obtained from the cluster membership values, calculated through fuzzy c-means algorithm. It could be regarded as an extension of ensemble one-class support vector machines [9]. Furthermore, Cyganek also extended ensemble OC-SVM to solve multi-class classification problem [24,25]. The above work is all related to cluster algorithms. Thus, the problems in cluster algorithms also exist. The number of clusters is difficult to choose. The results are affected by initialization. In this paper, a simple instance-weighted strategy is proposed. It is only relevant to the neighbors' distribution knowledge, which is only decided by the instance's k-nearest neighbors. Therefore, it can avoid the above problems.

Additionally, there also exists some other work which does not utilize cluster algorithms. Lee et al. provided a density-induced distance and used it to replace Euclidean distance in the SVDD [26]. Li et al. applied relative density degree as the instance weights for WSVDD [27].

3. Basic review of weighted one-class support vector machine

3.1. Problem description

In some real applications, the training samples are too few except only one class. There are plenty normal persons but few patients in disease diagnostics tasks, for instance. A good solution to deal with these problems is to utilize one-class classifier instead. In one-class classification, only massive one class data (usually named as targets) and fewer other classes data (usually named as outliers) are available. The aim of one-class classification problem is to find a data description to ensemble all or most of the targets and exclude all outliers. A graphical illustration is shown in Fig. 1. The pluses represent the targets and the solid closed line is the data description, which ensembles all targets.

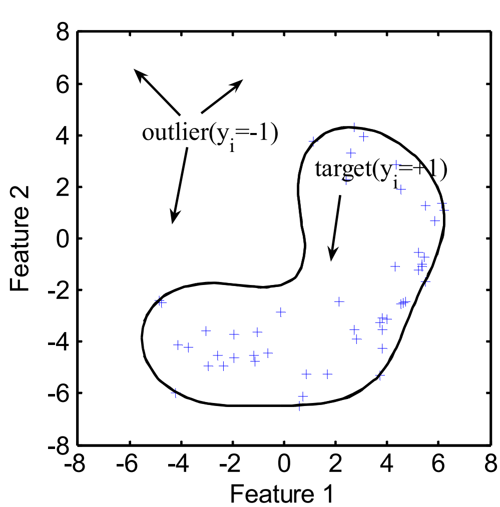

Fig. 1. A graphical illustration of the one-class classification problem. The pluses are targets and solid line is the data description.

Let $\mathbf{X}$ represent training set consisting of l targets, $\{\mathbf{x}_i, i=1,\dots,l\}$, $\mathbf{x}_i \in R^n$ (n is the dimension of $\mathbf{x_i}$). $\mathbf{x}$ is a new instance and $\mathbf{x} \in R^n$. The aim of one-class classification is to find a function $f(\mathbf{x})$. When $\mathbf{x}$ is a target, $f(\mathbf{x})$ returns 1; otherwise, $f(\mathbf{x})$ returns -1.

3.2. Basic review of OC-SVM

Let $\Phi(\mathbf{x}_i)$ denote the image of $\mathbf{x}_i$ in the feature space. $\Phi(\cdot)$ is unknown, however the inner product of the image $\Phi(\mathbf{x}_i)$ can be computed easily via kernel function, such as the Gaussian kernel $K(\mathbf{x}_i,\mathbf{x}_j)=\exp(-\|\mathbf{x}_i-\mathbf{x}_j\|^2/(2\sigma^2))$, where indices i and j range over $1,\dots l$.

In OC-SVM, the training instances are mapped into the feature space via kernel trick and separated from the origin by the hyperplane with the maximum margin [3]. Then the following convex programming needs to be solved.

$$\begin{aligned}\min_{\mathbf{w},\rho}\quad & \frac{1}{2}\mathbf{w}^T\mathbf{w}-\rho+\frac{1}{vl}\sum_{i=1}^{l}\xi_i \\ s.t\quad & \mathbf{w}^T\Phi(\mathbf{x}_i)\ge\rho-\xi_i,\quad \xi_i\ge 0\ \forall i=1,2,\dots,l\end{aligned} \tag{1}$$

Here ξ_i is the non-negative slack variable of $\mathbf{x}_i$; $v\in(0,1)$ controls the fraction of outliers (the training data outside the estimated region) and that of support vectors. v is the upper bound on the fraction of the outliers as well as a lower bound on the fraction of the support vectors. The superscript T denotes the transpose of a matrix or vector.

Introducing multipliers $\alpha_i,\beta_i\ge 0$ for the constraints $\mathbf{w}^T\Phi(\mathbf{x}_i)-\rho+\xi_i\ge 0$ and $\xi_i\ge 0$, the Lagrangian function of (1) can be written as following

$$\begin{aligned}L(\mathbf{w},\boldsymbol{\xi},\rho,\boldsymbol{\alpha},\boldsymbol{\beta}) &= \frac{1}{2}\mathbf{w}^T\mathbf{w}-\rho+\frac{1}{vl}\sum_{i=1}^{l}\xi_i-\sum_{i=1}^{l}\alpha_i(\mathbf{w}^T\Phi(\mathbf{x}_i)-\rho+\xi_i) \\ &\quad -\sum_{i=1}^{l}\beta_i\xi_i\end{aligned} \tag{2}$$

Setting the derivatives with respect to $\mathbf{w}$,$\boldsymbol{\xi}$ and ρ to zero, then we can obtain

$$\mathbf{w}=\sum_{i=1}^{l}\alpha_i\Phi(\mathbf{x}_i) \tag{3}$$

$$\alpha_i=\tfrac{1}{vl}-\beta_i,\quad \sum_{i=1}^{l}\alpha_i=1 \tag{4}$$

Substituting (3) and (4) into (2), the dual form of (1) can be written as

$$\begin{aligned}\min_{\boldsymbol{\alpha}}\quad & \boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha} \\ s.t.\quad & 0\le\alpha_i\le\tfrac{1}{vl},\quad \sum_{i=1}^{l}\alpha_i=1\end{aligned} \tag{5}$$

Here $\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\dots,\alpha_l)$ is the vector form of Lagrange multipliers of the constraints; $\mathbf{Q}$ is the kernel matrix for the training set, $Q(i,j)=K(\mathbf{x}_i,\mathbf{x}_j)$.

The instances $\{\mathbf{x}_i|\alpha_i>0, i=1,\dots,l\}$ are called support vectors. The function $f(\mathbf{x})$ can be written as

$$f(\mathbf{x})=\mathrm{sgn}\left(\sum_{i\in SVs}\alpha_iK(\mathbf{x}_i,\mathbf{x})-\rho\right) \tag{6}$$

Here SVs represents the indices of support vectors and the variable ρ in (6) can be recovered by

$$\rho=\mathbf{w}^T\Phi(\mathbf{x}_i)=\sum_{j\in\{j|\alpha_j\ne 0\}}\alpha_jK(\mathbf{x}_i,\mathbf{x}_j) \tag{7}$$