Knowledge-Based Systems 110 (2016) 210–223

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A novel approach to pre-extracting support vectors based on the theory of belief functions

Deqiang Han[a,*], Weibing Liu[a], Jean Dezert[b], Yi Yang[c]

[a] MOE KLINNS Lab, Institute of Integrated Automation, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, 710049, China
[b] ONERA, The French Aerospace Lab, Chemin de la Hunière, F-91761 Palaiseau, France
[c] SKLSVMS, School of Aerospace, Xi'an Jiaotong University, Xi'an, 710049, China

ARTICLE INFO

Article history:
Received 3 February 2016
Revised 11 June 2016
Accepted 21 July 2016
Available online 22 July 2016

Keywords:
SVM
Belief functions
Pre-extraction
Pattern recognition

ABSTRACT

Applications of the support vector machine (SVM) in the large scale datasets are seriously hampered by its high computational cost for training. In SVM training, the classification hyperplane is determined by support vectors (SVs). If those samples likely to be SVs can be pre-extracted and used for training, the computational cost can be reduced without the loss of classification accuracy. An approach to pre-extracting SVs is proposed where the training samples' uncertainty in terms of classification is modeled using belief functions. Those samples with a higher degree of uncertainty are more likely to be SVs. Our approach can also detect outliers and noisy samples. Experimental results based on benchmark datasets show that the proposed approach performs better compared with traditional approaches, where the training time is significantly reduced (approximate to one or two orders of magnitude), meanwhile it can obtain good classification accuracies.

1. Introduction

Support vector machine (SVM) [1,2] proposed by Vapnik and Chervonenkis, is a kind of powerful statistical learning method based on the principle of structural risk minimization (SRM) [1]. SVM has shown excellent learning and generalization performance in the classification and regression [3]. It has been widely used in applications such as disease diagnosis [4,5], electrical load forecasting [6], image recognition [7], etc.

The training of the SVM is actually to solve an optimization problem of convex quadratic programming (QP) with constraints [8]. Therefore, high computational cost are generated when the training set is large. This restricts the applications of SVM in large scale data sets. To deal with the problem of high computational complexity, one strategy is to propose some new optimization algorithms to alleviate the computational cost, e.g., decomposition methods [9], incremental SVM [10,11], least square support vector machine (LSSVM) [12], sequential minimal optimization (SMO) [13], SVM using approximate extreme points (AESVM) [14], etc. The other strategy is to pre-extract or select training samples before training. It is rational because the classification hyperplane obtained via the optimization is only determined by the support vectors (SVs) [1]. Thus, it is not necessary to use total training samples in the training procedure. One can pre-extract the samples that most likely to be SVs to reduce the size of data set without losing SVM's classification accuracy.

For the strategy of pre-extracting SVs, Almeida [15] proposed an approach using the clustering analysis and delete clusters composed of samples with the same class label. Jiao [16] proposed an approach to pre-extracting the SVs according to the distances between the sample and the centers of all the classes. There are other similar methods like the adaptive projective algorithm [17]. Some methods [18–20] pre-extract SVs according to the neighborhood information. However, the tuning of parameters is a problem for these methods, and when there exist some outliers and noisy samples in the training data set, the above approaches will have poor classification performances after pre-extracting SVs. In this paper, we focus on the second strategy, i.e., the pre-extracting of training samples that are likely to be support vectors.

As aforementioned, SVs are the samples that determine the classification hyperplane and are located near the boundary between the two classes of samples. That is, the class membership of SVs is ambiguous. Therefore, the ambiguity of class membership can be considered as the criterion of pre-extracting SVs. The

* Corresponding author.
E-mail addresses: deqhan@gmail.com (D. Han), 872756596@qq.com (W. Liu), jean.dezert@onera.fr (J. Dezert), jiafeiyy@mail.xjtu.edu.cn (Y. Yang).

http://dx.doi.org/10.1016/j.knosys.2016.07.029
0950-7051/

bigger the ambiguity is, the bigger the likelihood of being SV is. The ambiguity includes two parts [21]: the discord and the non-specificity [22]. The discord of a sample's class membership represents the disagreement between the sample's likelihood of belonging to different classes. The non-specificity [21] of a sample's class membership represents the sample's likelihood belonging to both two classes left unspecified. The aforementioned neighborhood based approaches actually use the probability to model the uncertainty, which actually can only model the discord part, but cannot describe the non-specificity part. Hence, the traditional approaches cannot well describe the ambiguity encountered here.

Since the theory of belief functions [23] provides a powerful mathematical framework to describe and handle the uncertainty, especially the ambiguity including the discord and non-specificity, we propose a new approach to pre-extracting SVs based on the belief functions. First, the basic belief assignment (BBA) of each sample is obtained using the BBA generation approach in the evidential *c*-means (ECM) [24] algorithm. Then, the ambiguity measures of the BBAs representing the ambiguity of the corresponding samples in terms of the class membership are calculated. Finally, the samples likely to be SVs are pre-extracted according to the values of ambiguity measures. If a sample's corresponding ambiguity measure is greater than a given threshold, it is highly possible a SV and pre-extracted as a training sample. Furthermore, the outliers can be detected according to the mass assignment of empty set in the generated BBA, which represents the degree of a sample being away from the region where most of the samples are located. Meanwhile, a mechanism to detect the noisy sample is designed using the degree of inconsistency between the true class label and the label obtained based on the BBA. Experimental results and related analyses show the rationality and efficiency of our proposed approach. It can reduce the training samples' size, and thus, accelerate the training time without degrading the classification accuracy. It also performs well when some outliers and noisy samples exist.

This paper is organized as follows:in Section 2, the problem statement is provided. Then, we briefly review related works in Section 3. We present our approach to pre-extracting SVs in Section 4. In Section 5, three traditional approaches are compared with our approach using an illustrative example. Section 6 provides the experimental results of our proposed method with respect to some traditional methods based on some benchmark data sets. The conclusion is drawn in Section 7.

2. Problem statement

2.1. Basics of support vector machine

Suppose that there is a group of training samples $\{(x_i, y_i), x_i \in R^d, y_i \in \{+1, -1\}, i = 1, \ldots, N\}$, where x_i represents the ith sample and y_i represents its corresponding class label. SVM aims to find a hyperplane separating the positive training samples from those negative ones and maximize the margin. The primary quadratic programming of SVM is [1,2]

$$\begin{aligned} \min \quad & \tfrac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i \\ s.t. \quad & y_i (w^T x_i + b) \geq 1; \\ & \xi_i \geq 0; i = 1, \ldots, N \end{aligned} \tag{1}$$

where ξ_i is the error term and $C > 0$ is the regularization parameter. By introducing Lagrange multiples α_i and β_i for the constrains in Eq. (1), its dual form is formulated as

$$\begin{aligned} \max \quad & \sum_{i=1}^{N} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_i x_i^T x_j \\ s.t. \quad & \sum_{i=1}^{N} \alpha_i y_i = 0; \\ & \alpha_i \geq 0; i = 1, \ldots N \end{aligned} \tag{2}$$

By solving the dual problem, the classification hyperplane is obtained as

$$\sum_{i=1}^{N} \alpha_i y_i x_i^T x + b = 0 \tag{3}$$

The corresponding classifier is

$$f(x) = \mathrm{sgn}(w^T x + b) = \mathrm{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i x_i^T x + b \right) \tag{4}$$

where sgn(.) is the sign function defined as: $\mathrm{sgn}(x) = -1, \forall x < 0$; $\mathrm{sgn}(x) = 0, \forall x = 0$; $\mathrm{sgn}(x) = 1, \forall x > 0$. For the classifier in Eq. (4), each coefficient α_i corresponds to a training sample. Those samples with zero coefficients, i.e., $\alpha_i = 0$, have no contribution to the classifier. Only the samples with non-zero coefficients $\alpha_i > 0$ called SVs contribute to the classifier [1].

For non-linear problems, the function $\varphi(x_i)$ maps the training samples from the original input space to the kernel space. The kernel function $K(x_i, x_j) = \varphi^T(x_i)\varphi(x_j)$ is used to represent the inner product in kernel space instead of $x_i^T x_j$ in the original space. Solving the QP problem to establish the corresponding classifier for the non-linear problem is similar to solving the linear problem. Some common nonlinear kernels [2] are introduced as follows. The polynomial kernel is expressed as

$$K(x_i, x_j) = ((x_i \cdot x_j) + 1)^d \tag{5}$$

where d is a positive integer specifying the order of it. The radial basis function (RBF) kernel is expressed as

$$K(x_i, x_j) = \exp\left(-\frac{\left\| x_i - x_j \right\|^2}{\sigma^2} \right) \tag{6}$$

where σ is a positive number specifying the scaling factor.

Note that the two-class (binary) SVM can be generalized to handle multi-class classification problems [25]. Here we focus on the two-class SVM.

2.2. Computational complexity analysis

As aforementioned, one needs to solve a QP problem to train the SVM. Suppose that the number of training samples is *N*. Solving this QP problem takes up to $O(N^3)$ [9] computational complexity. A simple example is provided to show the training time against the number of samples when active set algorithm is used to solve the QP problem. As shown in Fig. 1, the number of training samples starts from 200 and increases 25% at each time until 3418. The relationship between the training time and the number of training samples (feature dimension is 50) is approximately linear in the dual-logarithm coordinate. The training time increases about 7 times from 8.30 s to 58.15 s while the number of samples increase 1.95 times from 550 to 1014. The rate of time increase is approximately three times as large as that of the samples number increase. It conforms to the aforementioned rule that the computational complexity is about $O(N^3)$. Obviously, the training time increases quickly with the enlargement of the size of the training data set.

Therefore, high computational cost of SVM training makes it difficult to use SVM on large scale data sets. In addition, the computational complexity is also influenced by the feature dimension.