Pattern Recognition 47 (2014) 2542–2557

Contents lists available at ScienceDirect

Pattern Recognition

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

Multi-label core vector machine with a zero label

Jianhua Xu

School of Computer Science and Technology, Nanjing Normal University, Nanjing, Jiangsu 210023, China

ARTICLE INFO

Article history:
Received 13 April 2013
Received in revised form
9 January 2014
Accepted 22 January 2014
Available online 5 February 2014

Keywords:
Multi-label classification
Support vector machine
Core vector machine
Frank–Wolfe method
Quadratic programming
Linear programming

ABSTRACT

Multi-label core vector machine (Rank-CVM) is an efficient and effective algorithm for multi-label classification. But there still exist two aspects to be improved: reducing training and testing computational costs further, and detecting relevant labels effectively. In this paper, we extend Rank-CVM via adding a zero label to construct its variant with a zero label, i.e., Rank-CVMz, which is formulated as the same quadratic programming form with a unit simplex constraint and non-negative ones as Rank-CVM, and then is solved by Frank–Wolfe method efficiently. Attractively, our Rank-CVMz has fewer variables to be solved than Rank-CVM, which speeds up training procedure dramatically. Further, the relevant labels are effectively detected by the zero label. Experimental results on 12 benchmark data sets demonstrate that our method achieves a competitive performance, compared with six existing multi-label algorithms according to six indicative instance-based measures. Moreover, on the average, our Rank-CVMz runs 83 times faster and has slightly fewer support vectors than its origin Rank-CVM.

1. Introduction

Traditional supervised classification deals with problems in which one instance is only associated with a single class label and thus the classes are mutually exclusive [1]. However, in many real-world applications, e.g., text categorization [2–4], scene and video annotation [5–7], bioinformatics [8–10], and music emotion classification [11], one instance possibly belongs to several labels simultaneously. One typical example is that a sunrise image could be labeled by sun, sky and sea at the same time [5,6]. Such a classification setting is referred to as multi-label classification, which has attracted a lot of attention in the past 10 years [12–15]. Nowadays, there mainly exist four ways to construct various discriminative multi-label classification algorithms: data decomposition, algorithm extension, hybrid and ensemble strategies.

Data decomposition strategy splits a multi-label data set into either one or more single-label (binary or multi-class) subsets, trains a sub-classifier for each subset using an existing classifier, and then combines all sub-classifiers into an entire multi-label classifier. There are two widely used decomposition tricks: one-versus-rest (OVR) or binary relevance (BR), and label powerset (LP) [12–15]. It is convenient to build a data decomposition multi-label method since many popular single-label classifiers (e.g., support vector machine (SVM) and *k*-nearest neighbor method (kNN)) and their free software are available. The main criticism is that label correlations are not depicted explicitly in OVR methods and lots of new classes with a few instances are created in LP methods.

Algorithm extension strategy extends a specific multi-class algorithm to consider all training instances and classes (or labels) of a multi-label training data set all together. But such a strategy could induce some complicated optimization problems, e.g., two large-scale quadratic programming (QP) ones in multi-label support and core vector machines (Rank-SVM and Rank-CVM) [9,10], and a large-scale unconstrained optimization one in multi-label back-propagation neural networks (BP-MLL) [8]. Attractively, these methods explicitly describe as many label correlations of individual instance using pairwise constraints between relevant labels and irrelevant ones as possible.

Hybrid strategy not only generalizes an existing single-label method but also divides a multi-label data set into a series of subsets implicitly or explicitly. After the OVR trick is embedded, the kNN is cascaded with discrete Bayesian rule and logistic model respectively in ML-kNN and IBLR-ML [6,16], and the label correlation is characterized using different upper bounds in OVR-ESVM [17]. Such a strategy weakly depicts label correlations either explicitly or implicitly with a relatively lower computational cost.

Ensemble strategy [18] either extends an existing multi-class ensemble classifier or realizes a new ensemble of the aforementioned three kinds of multi-label techniques. The famous AdaBoost is generalized to implement two different multi-label versions: AdaBoost.MH and AdaBoost.MR [3]. The former is further integrated with alternative decision tree to construct a tree-type ensemble classifier ADTree [19]. Random *k*-labelsets (RAkEL) method splits an entire label set into several subsets of the size *k*, learns LP classifiers and then integrates an ensemble multi-label algorithm [20]. Ensemble of classifier chains (ECC) [21] is an ensemble technique which uses classifier chains (CC) as a base classifier, where CC indicates to construct an OVR classifier in

E-mail addresses: xujianhua99@tsinghua.org.cn, xujianhua@njnu.edu.cn

http://dx.doi.org/10.1016/j.patcog.2014.01.012
0031-3203

a cascade way rather than a parallel one. In [18], random forest of predictive clustering trees (RF-PCT) is strongly recommended because of its good performance from an extensive experimental comparison, including ECC and RAkEL. Usually, these ensemble methods spend more training and testing costs to achieve their classification performance improvement.

As mentioned above, algorithm extension strategy considers as many label correlations as possible, which is one of the optimal ways to improve multi-label classification performance further [22]. But, its corresponding methods have a relatively high computational complexity which limits their usability for lots of real world applications. Therefore, it is still necessary to construct some novel efficient multi-label algorithms. In this paper, our focus is on SVM-type multi-label classifiers.

The famous Rank-SVM [9] is formulated as a QP problem with equality constraints and box ones. When Frank–Wolfe method (FWM) [23,24] is applied, Rank-SVM needs to deal with a large-scale linear programming (LP) at its each iteration. The Rank-CVM [10] is depicted as a QP problem with a unit simplex constraint and non-negative ones. When Rank-CVM is solved by FWM, at its each iteration, there exist a closed solution, a closed step size and several efficient recursive formulae. The theoretical analysis and experimental study show that Rank-CVM has a lower time complexity than Rank-SVM, although both of them have the same number of variables to be solved. This implies that a special QP form possibly can result in an efficient SVM-type multi-label classifier.

On the other hand, both Rank-SVM and Rank-CVM need an additional linear threshold function to detect relevant labels. Through adding a zero label for isolating relevant labels from irrelevant ones, a variant of Rank-SVM, i.e., Rank-SVMz [25], is proposed, whose QP form includes disjoint equality constraints for different classes, and then is solved by FWM with the OVR trick. When the label cardinality is slightly large, the number of variables to be solved in Rank-SVMz is much fewer than that in Rank-SVM, e.g., 21,000/58,248 for Yeast [9,25]. Therefore embedding a zero label both reduces the computational cost and learns a threshold function simultaneously.

In this paper, we generalize Rank-CVM to build its variant with a zero label, i.e., Rank-CVMz, which is formulated as the same QP form as Rank-CVM and thus is solved by FWM efficiently. Particularly, the number of variables to be solved in Rank-CVMz is the same as that in Rank-SVMz and fewer than that in Rank-CVM. Hence, our Rank-CVMz has a lower time complexity than Rank-CVM. Additionally, the relevant labels are detected effectively via the zero label. Experiments on 12 benchmark data sets illustrate that our method is a competitive candidate for multi-label classification according to six instance-based measures, compared with six existing techniques including Rank-CVM [10], Rank-SVMz [25], Rank-SVM [9], BP-MLL [8], ML-kNN [6] and RF-PCT [18]. Furthermore, our Rank-CVMz runs 83 times faster and has slightly fewer support vectors than Rank-CVM on the average.

The rest of this paper is organized as follows. Multi-label classification setting is introduced in Section 2 and three related SVM-type methods are summarized in Section 3. In Sections 4 and 5, our Rank-CVMz is proposed and then an efficient training algorithm is constructed and analyzed. Section 6 is devoted to experiments with 12 benchmark data sets. This paper ends with some conclusions in Section 7.

2. Multi-label classification setting

Let $X \in R^d$ be a d-dimensional real input space, $Q = \{1, 2, \ldots, q\}$ a finite set of q class labels, and 2^Q all possible subsets of Q. We denote a training data set of size l drawn identically and independently from an unknown probability distribution on $X \times 2^Q$ by

$$\{(\boldsymbol{x}_1, L_1), \ldots, (\boldsymbol{x}_i, L_i), \ldots, (\boldsymbol{x}_l, L_l)\}, \tag{1}$$

where $\boldsymbol{x}_i \in X$ and $L_i \in 2^Q$ represent the ith instance and its relevant label set. Additionally, the complement of L_i, i.e., $\overline{L}_i = Q \backslash L_i$, is referred to as the irrelevant label set. For the convenience of formula representation, we also adopt a binary vector $\boldsymbol{y}^i = [y_{i1}, y_{i2}, \ldots, y_{iq}]$ to label the instance $\boldsymbol{x}_i$, where $y_{ik} = 1$ if the kth label is in L_i, and -1 otherwise.

The goal of multi-label classification is to learn a classifier $f(\boldsymbol{x}) : X \to 2^Q$ which can predict the relevant labels for unseen instances in the sense of optimizing some expected risk functional with respect to a specific empirical loss function [8–10,16].

In classical q-class single-label classification, a widely used trick is to learn q discriminant functions $f_k(\boldsymbol{x}) : X \to R, k = 1, \ldots, q$ such that $f_k(\boldsymbol{x}) > f_{k'}(\boldsymbol{x}), k \neq k'$ if $\boldsymbol{x} \in$ class k [1]. For multi-label classification, as a natural extension of multi-class one, this trick is adapted as $f_k(\boldsymbol{x}) > f_{k'}(\boldsymbol{x}), k \in L$ and $k' \in \overline{L}$, which means that any relevant label should be ranked higher than any irrelevant one [8–10,25]. In case such an ideal case does not happen, the ranking loss over the training set (1) can measure the average fraction of label pairs (any relevant label versus any irrelevant one) that are not correctly ordered:

$$\text{Ranking loss} = \frac{1}{l}\sum_{i=1}^{l}\left(\frac{1}{|L_i||\overline{L}_i|}\left|(k,k') \in (L_i \times \overline{L}_i) | f_k(\boldsymbol{x}_i) \leq f_{k'}(\boldsymbol{x}_i)\right|\right). \tag{2}$$

It is worth noting that this measure is non-differentiable. But, we could find out an approximate proxy of (2) as an empirical loss to be minimized in SVM-type multi-label classifiers. Finally, the multi-label prediction is executed through a proper threshold function $t(\boldsymbol{x})$:

$$f(\boldsymbol{x}) = \{k | f_k(\boldsymbol{x}) \geq t(\boldsymbol{x}), k = 1, \ldots, q\}. \tag{3}$$

Now there are mainly three kinds of thresholds: a constant (e.g., 0.0 for $-1/+1$ setting and 0.5 for 0/1 one) [2,5], a linear regression model associated with q discriminant function values [8–10], and an additional discriminant function for a virtual, calibrated or zero label [25–27]. In the last two cases, $t(\boldsymbol{x})$ is dependent on $\boldsymbol{x}$ either directly or implicitly. Several more complicated threshold methods can improve classification performance further [28,29].

3. Related work

In the recent years, since multi-label classification has received a lot of attention in machine learning, pattern recognition and statistics, a variety of methods have been presented, which have been summarized and reviewed in four exhaustive overviews [12–15] and our previous work [10,17,25]. In this section, we mainly review three multi-label SVM-type methods: Rank-SVM [9], Rank-CVM [10] and Rank-SVMz [25].

For a q-class multi-label training set (1), we define the following linear discriminant functions in the original input space:

$$f_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + b_k, \quad k = 0, 1, \ldots, q, \tag{4}$$

where $\boldsymbol{w}_k$ and b_k denote respectively the weight vector and the bias term for the kth label or class, and $k = 0$ indicates a zero label specially.

3.1. Multi-label support vector machine and core vector machine

Multi-label support vector machine (i.e., Rank-SVM) [9] extends multi-class SVM [30] to deal with multi-label classification. It is desirable that any relevant label should be ranked one higher than any irrelevant. In case such an ideal situation does not occur, a slack variable is introduced. Therefore, the relative relationship between any relevant label and any irrelevant one for some training instance $\boldsymbol{x}_i$, i.e., $(m, n) \in (L_i \times \overline{L}_i)$, is described using the following pairwise