Expert Systems with Applications 41 (2014) 3418–3428

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Random block coordinate descent method for multi-label support vector machine with a zero label

Jianhua Xu

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

ARTICLE INFO

Keywords:
Multi-label classification
Support vector machine
Zero label
Frank–Wolfe method
Block coordinate descent method
Quadratic programming
Linear programming

ABSTRACT

Multi-label support vector machine with a zero label (Rank-SVMz) is an effective SVM-type technique for multi-label classification, which is formulated as a quadratic programming (QP) problem with several disjoint equality constraints and lots of box ones, and then is solved by Frank–Wolfe method (FWM) embedded one-versus-rest (OVR) decomposition trick. However, it is still highly desirable to speed up the training and testing procedures of Rank-SVMz for many real world applications. Due to the special disjoint equality constraints, all variables to be solved in Rank-SVMz are naturally divided into several blocks via OVR technique. Therefore we propose a random block coordinate descent method (RBCDM) for Rank-SVMz in this paper. At each iteration, an entire QP problem is divided into a series of small-scale QP sub-problems, and then each QP sub-problem with a single equality constraint and many box ones is solved by sequential minimization optimization (SMO) used in binary SVM. The theoretical analysis shows that RBCDM has a much lower time complexity than FWM for Rank-SVMz. Our experimental results on six benchmark data sets demonstrate that, on the average, RBCDM runs 11 times faster, produces 12% fewer support vectors, and achieves a better classification performance than FWM for Rank-SVMz. Therefore Rank-SVMz with RBCDM is a powerful candidate for multi-label classification.

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 (Duda, Hart, & Stork, 2001). However, in many real world applications, one instance possibly belongs to several labels simultaneously, e.g., a sunrise image could be annotated by sun, sky and sea at the same time (Boutell, Luo, Shen, & Brown, 2004). Such a classification issue is referred to as multi-label classification and has been attracted a lot of attention in the recent decade. So far, a variety of multi-label methods have been proposed and validated (Madjarov, Kocev, Gjorgjevikj, & Dzeroski, 2012; Tsoumakas, Katakis, & Vlahavas, 2010; Zhang & Zhou, 2013), most of which can be categorized into three groups: problem transformation, algorithm adaptation and ensemble methods.

Problem transformation methods convert a multi-label problem into either one or more single-label (binary or multi-class) sub-problems, construct a sub-classifier for each sub-problem using an existing classification technique, and then aggregate all sub-classifiers into an entire multi-label classifier. It is convenient and fast to implement a problem transformation method due to lots of existing techniques and their free software, e.g., support vector machine (SVM), *k*-nearest neighbor method (kNN), naive Bayes (NB), and so on. There are mainly two widely-used transformation tricks: one-versus-rest (OVR) or binary relevance (BR), and label powerset (LP) (Tsoumakas et al., 2010; Zhang & Zhou, 2013). 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. Through incorporating label correlations into OVR methods specially, the classification performance can be enhanced further (Alvares-Cherman, Metz, & Monard, 2012; Montanes et al., 2013).

Algorithm adaptation methods extend their original multi-class classification algorithms to handle an entire multi-label training data set directly. It is worth noting that this kind of methods could be further divided into two sub-groups. One considers all class labels and all instances simultaneously in order to explicitly characterize as many label correlations as possible, and usually induces some complicated optimization problems, such as, large-scale quadratic programming (QP) problems in multi-label SVM-type methods (e.g., Rank-SVM (Elisseeff & Weston, 2001), Rank-SVMz (Xu, 2012), and Rank-CVM (Xu, 2013)), and large-scale unconstrained problem in multi-label BP neural networks (BP-MLL) (Zhang & Zhou, 2006). The other still deals with each class independently after using some problem transformation tricks (e.g., OVR), such as, four multi-label kNN-type methods (ML-kNN (Zhang & Zhou, 2007), IBLR-ML (Cheng & Hullermeier, 2009), FSKNN (Jiang, Tsai, & Lee, 2012), and MlCBR (Nicolas, Sancho-Asensio, Golobardes, Fornells, & Orriols-Puig, 2013)), multi-label classification based

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

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.12.004

on neighborhood rough set using local correlation (MLRS-LC) (Yu, Pedrycz, & Miao, 2013), multi-label RBF neural networks (ML-RBF) (Zhang, 2009), and extended OVR support vector machine (OVR-ESVM) (Xu, 2011). The latter sub-group of methods has much lower computational complexity than the former one, which inspires us to introduce problem transformation tricks into algorithm adaptation methods for reducing as many computational costs as possible.

Ensemble methods (Madjarov et al., 2012) either generalize an existing multi-class ensemble classifier, or realizes a new ensemble of the aforementioned two kinds of multi-label techniques. The famous AdaBoost is extended to construct two slightly different multi-label versions: AdaBoost.MH and AdaBoost.MR (Schapire & Singer, 2000), where "H" and "R" indicate that Hamming and ranking losses are minimized respectively. Random k-labelsets (RAkEL) method divides an entire label set into several subsets of the size k, trains LP classifiers and then constructs an ensemble multi-label algorithm (Tsoumakas, Vlahavas, & Katakis, 2011). Ensemble of classifier chains (ECC) (Read, Pfahringer, Holmes, & Frank, 2011) is an ensemble technique which uses classifier chains (CC) as a base classifier, where CC implies to build an OVR classifier in a cascade way rather than a parallel one. In Madjarov et al. (2012), random forest of predictive clustering trees (RF-PCT) is strongly recommended due to its good performance from an extensive experimental comparison, including ECC and RAkEL. Usually, these ensemble methods spend more training and testing time to achieve their classification performance improvement.

Now it is universally recognized that characterizing as many label correlations as possible could effectively improve the multi-label classification performance (Alvares-Cherman et al., 2012; Dembczynski, Waegeman, Cheng, & Hullermeier, 2012; Montanes et al., 2013; Yu et al., 2013). As mentioned above, considering all training instances and all labels simultaneously is an explicit and effective way to depict label correlations. One representative algorithm is multi-label support vector machine with a zero label (Rank-SVMz, originally SVM-ML) (Xu, 2012). The original form of Rank-SVMz depicts the label correlations using all possible pairwise constraints between the relevant (and irrelevant) labels and a zero label. For a q-class multi-label classification data set of the size l, the dual version of Rank-SVMz is formulated as a QP problem with q disjoint equality constraints and ql box constraints, whose number of total variables to be solved is ql.

Frank–Wolfe method (FWM) is a simple first order feasible direction optimization technique (Frank & Wolfe, 1956), which converts a original problem with linear constraints and box constraints into a series of linear programming (LP) problems. When FWM is applied to Rank-SVMz, at each iteration or epoch, the entire LP problem can be divided into q LP sub-problems of the size l via OVR decomposition trick according to the disjoint equality constraints of different classes. The time complexity of each epoch is $O(q^2l^2)$. On the other hand, FWM has a sub-linear convergence rate (Frank & Wolfe, 1956; Guelat & Marcotte, 1986; Xu, 2013). To achieve an ϵ accuracy solution, it is needed to execute $O(2\sqrt{q2+3q}\|\boldsymbol{K}\|_F\Delta^2/\epsilon)$ epochs, where $\|\boldsymbol{K}\|_F$ represnts the Frobenius norm of kernel matrix and Δ indicates the diameter of the polyhedron satisfying all constraints. It has been experimentally demonstrated that Rank-SVMz based on FWM can produces a dense solution vector, which costs more computational time in the testing procedure. Therefore it is still imperative to speed up the training and testing procedures of Rank-SVMz further for many real world applications.

Block coordinate descent method (BCDM) is also one of the oldest first-order optimization techniques, which has been paid more attention to recently due to its simplicity and efficiency (Necoara & Patrascu, 2013; Wen, Goldfarb, & Scheinberg, 2012). BCDM partitions all variables into some manageable blocks and updates a single block only at each iteration while the remaining blocks are fixed. Additionally, there are three possible ways to select a single block: greedy using gradient information, cyclic and random methods. As we known, binary SVM is widely optimized by sequential minimization optimization (SMO) (Chang & Lin, 2011; Fan, Chen, & Lin, 2005), in which two variables are chosen according to gradient information at each iteration. Therefore SMO essentially is an efficient greedy BCDM. In order to avoid calculating or maintaining a large-scale gradient vector, the random way is widely accepted now (Necoara & Patrascu, 2013; Wen et al., 2012).

Since Rank-SVMz involves q disjoint equality constraints, all ql variables are naturally split into q blocks of the size l via OVR decomposition trick. Therefore we introduce a random block coordinate descent method (RBCDM) for Rank-SVMz in this paper. At each iteration, we select a block randomly and optimize its corresponding QP sub-problem with a single equality constraint and l box ones using SMO in binary SVM (Chang & Lin, 2011; Fan et al., 2005). At each epoch, we update all q blocks, whose time complexity is $O(ql^{2.3})$, which is slightly higher than that of FWM. Our RBCDM for Rank-SVMz still has a sub-linear convergence rate in terms of the work in Necoara and Patrascu (2013). To achieve an ϵ accuracy solution, the number of epochs is $O(2\|\boldsymbol{K}\|_F R_0^2/\epsilon)$, where R_0 stands for the size of the level set. Since $R_0 \leqslant \Delta$, RBCDM theoretically needs much fewer epochs than FWM for Rank-SVMz. Summarily, RBCDM has a lower overall time complexity than FWM for Rank-SVMz.

The experimental results on six data sets illustrate that on the average RBCDM runs 11 times faster, has 12% fewer support vectors, and obtains a better performance than FWM for Rank-SVMz, and Rank-SVMz with RCBDM is a powerful multi-label classifier, compared with the other state-of-the-art multi-label techniques including Rank-SVMz with FWM (Xu, 2012), Rank-SVM (Elisseeff & Weston, 2001), BP-MLL (Zhang & Zhou, 2006), ML-kNN (Zhang & Zhou, 2007) and RF-PCT (Madjarov et al., 2012).

The rest of this paper is organized as follows. Rank-SVMz and its FWM are reviewed in Sections 2 and 3. RBCDM for Rank-SVM is proposed and analyzed in Section 4. Section 5 is devoted to experiments with six benchmark data sets. This paper ends with some conclusions in Section 6.

2. Multi-label support vector machine with a zero label

In this section, we review multi-label support vector machine with a zero label: Rank-SVMz, i.e., SVM-ML originally in Xu (2012). Let a finite set of q class labels be $L = \{1, 2, \ldots, q\}$ and its all possible subsets be 2^L. We denote a training set of the size l drawn identically and independently from an unknown probability distribution on $R^d \times 2^L$ by,

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

where $\boldsymbol{x}_i \in R^d$ and $L_i \in 2^L$ represent the ith d-dimensional instance and its relevant label subset. Additionally, the complement of L_i, i.e., $\overline{L}_i = L - L_i$, indicates the irrelevant label subset. 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.

In the original input space, $q+1$ linear discriminant functions are defined as,

$$f_k(\boldsymbol{x}) = \boldsymbol{w}_k^T\boldsymbol{x} + b_k, \quad k = 0, 1, 2, \ldots, q, \tag{2}$$

where $k = 0$ represents a zero label, and $\boldsymbol{w_k}$ and b_k stand for the weight vector and bias term of the kth function. In Rank-SVMz, the zero label is used as a natural zero point to separate the relevant labels from the irrelevant labels. Now it is desirable that any relevant label should be ranked 1 higher than such a zero label, and any irrelevant label 1 lower than this zero label, at the same time.