Engineering Applications of Artificial Intelligence 45 (2015) 453–463

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Dependence maximization based label space dimension reduction for multi-label classification

Ju-Jie Zhang [a], Min Fang [a,*], Hongchun Wang [a,b], Xiao Li [a]

[a] School of Computer Science and Technology, Xidian University, No. 2, South Taibai Road, Xi'an, Shaanxi 710071, PR China
[b] AVIC Computing Technique Research Institute, Xi'an, China

ARTICLE INFO

Article history:
Received 8 February 2015
Received in revised form
27 July 2015
Accepted 30 July 2015
Available online 27 August 2015

Keywords:
Multi-label classification
Dimension reduction
Label space
Dependence maximization
Hilbert–Schmidt independence criterion

ABSTRACT

High dimensionality of label space poses crucial challenge to efficient multi-label classification. Therefore, it is needed to reduce the dimensionality of label space. In this paper, we propose a new algorithm, called *dependence maximization based label space reduction* (DMLR), which maximizes the dependence between feature vectors and code vectors via Hilbert–Schmidt independence criterion while minimizing the encoding loss of labels. Two different kinds of instance kernel are discussed. The global kernel for $DMLR_G$ and the local kernel for DM_LR_L take global information and locality information into consideration respectively. Experimental results over six categorization problems validate the superiority of the proposed algorithm to state-of-art label space dimension reduction methods in improving performance at the cost of a very short time.

1. Introduction

During the last decade, multi-label classification has aroused the interest of researchers from engineering and academic areas because of its wide applications in real world. In multi-label setting, a document may be associated with multiple categories (Ji et al., 2010; Ueda and Saito, 2003); an image may be annotated with several concepts (Boutell et al., 2004). It is rather different from the traditional single-label (binary or multi-class) classification where each document is only allowed to be associated with one category.

A lot of algorithms have been proposed for multi-label classification (Zhang and Zhou, 2014). Currently a consensus on multi-label classification is that label correlations play an important role and should be utilized for performance improvement (Dembczyński et al., 2010; Zhang and Zhang, 2010; Zhang and Zhou, 2014). Most algorithms usually build classification model based on some label correlation assumption, such as *ensemble of classifier chains* (ECC) (Read et al., 2011) and *calibrated label ranking* (CLR) (Fürankranz et al., 2008).

Although these algorithms achieve satisfactory results, they suffer from computational inefficiency in both training and testing even for the most intuitive approach *binary relevance* (BR) (Boutell et al., 2004), which decomposes a multi-label classification problem into several independent binary classification problems, one for each label, based on one-versus-all (OVA) strategy (Hastie et al., 2009). This problem poses a rather crucial challenge to classification especially when there are a large number of possible labels. Therefore, it is necessary to explore ways that balances the classification performance and computational efforts.

Several algorithms on label space dimension reduction (LSDR) have been proposed along this avenue, which can be categorized into two groups: *learning methods* and *reduction methods*. The former group reduces the label space while jointly learning a classifier from the instances to the code vectors as well, for example *multi-label prediction via compressed sensing* (CS) (Hsu et al., 2009). We can obtain a classifier finally and use it for predicting directly. However, in order to get a promising classifier, these methods often employ complicated algorithms in the learning part, which again costs too much time. Therefore, the latter group is the mainstream in this avenue.

The latter group focuses on how to efficiently compress the label space and does not consider what learning algorithm to apply after compression. An exemplar is *principal label space transformation* (PLST) (Tai and Lin, 2012), which only reduces the dimensionality of label space by analyzing the principal components. A key problem of this group is on how to utilize the instances, which is still an open question. Since the ultimate objective is to make classification, some methods only use a simple model from instances to code vectors, for instance, *conditional PLST* (CPLST) (Chen and Lin, 2012). Nevertheless, this strategy might be suboptimal as it may over-fit the learnt model, which has a negative impact on the learning process later.

* Corresponding author.
E-mail address: fanglabtg@163.com (M. Fang).

http://dx.doi.org/10.1016/j.engappai.2015.07.023
0952-1976/

In this paper, we propose a new LSDR method, called *dependence maximization based label space reduction* (DMLR), which can be categorized into the latter group as a reduction method. Different from previous reduction methods, it assumes that the objective function should consist of two components: encoding loss and dependence loss. The former one measures the loss of label compression while the latter one measures the dependence between instances and code vectors. Specifically, it measures the encoding loss using least square loss function as used in PLST and measures the dependence loss based on *Hilbert–Schmidt independence criterion* (HSIC) (Gretton et al., 2005). Two different instance kernels are applied and we obtain two methods: DMLR_G with the instance kernel exploiting global information and DMLR_L with the instance kernel exploiting local information. Experimental results across six data sets from various application domains validate the superiority of two proposed algorithms to two state-of-art LSDR methods, PLST and CPLST, in performance and save a lot of training and testing time compared with a simple representative multi-label classification method – BR. Moreover, DMLR_L outperforms DMLR_G in performance in most cases and costs similar or less training time due to the sparsity of instance kernel used in DMLR_L.

The rest of this paper is organized as follows. Section 2 presents a brief literature review on multi-label classification algorithms and pays more attention on LSDR methods and the HSIC. We describe the two proposed algorithms, DMLR_G and DMLR_L, in detail in Section 3 and. Experimental results and discussion are given in Section 4 Finally, Section 5 concludes this paper and presents some clues for future work.

2. Related works

Since this paper focuses on LSDR methods, we present a brief literature review on multi-label classification in Section 2.1 and existing LSDR methods in Section 2.2. Section 2.3 describes the dependence measurement criterion HSIC on which our proposed methods relies. But for convenience of presentation, we first give the formulation of multi-label classification.

Let $\mathbf{D}=\left\{(\mathbf{X}_i,\mathbf{Y}_i)_{i=1}^N\right\}$ be the training set with N examples, where $\mathbf{X}_i\in R^d$ is the ith instance (or feature vector) and $\mathbf{Y}_i\in\{-1,+1\}^L$ is the corresponding label vector of $\mathbf{X}_i$. If $\mathbf{X}_i$ is associated with the lth label, $\mathbf{Y}_{il}=+1$; otherwise, $\mathbf{Y}_{il}=-1$. We denote by $\mathbf{X}\in R^{N\times d}$ the instance matrix with each row being $\mathbf{X}_i^T$(the superscript T stands for transpose); likewise, we denote by $\mathbf{Y}\in\{-1,+1\}^{N\times L}$ the label matrix with each row being $\mathbf{Y}_i^T$. The task of multi-label classification is to learn a mapping: $h: R^d\to\{-1,+1\}^L$ so that for a new instance $\mathbf{x}\in R^d$, h is able to make prediction $h(x)\in\{-1,+1\}^L$. In this paper, $\mathbf{e}$ is a vector of appropriate length with all elements being one and $\mathbf{I}$ is an identity matrix of appropriate size.

2.1. Works on multi-label classification

Existing algorithms for multi-label classification can be grouped into two classes: *algorithm adaptation* and *problem transformation* (Tsoumakas et al., 2010). Algorithm adaptation methods handle multi-label classification problems by adapting existing single-label methods to multi-label cases by enforcing some assumptions on label correlations, such as preference ranking (Bucak et al., 2009; Xu, 2012; Elisseeff and Weston, 2001; Schapire and Singer, 2000), shared subspace (Ji et al., 2010), local relationship (Cheng and Hüllermeier, 2009; Huang and Zhou, 2012; Zhang and Zhang, 2010), hyper-graph connection (Sun et al., 2008), and etc. These algorithms usually need to solve a complicated optimization problem, which is at least linear in the number of possible labels L as in the case of BR.

Problem transformation methods transform multi-label classification problems into the framework of single-label classification problems, which can then be solved by existing single-label algorithms. Various transformation techniques have been applied in order to exploit correlations, such as binary relevance (Boutell et al., 2004; Zhang and Zhou, 2007; Zhang et al., 2015), combination method (Tsoumakas et al., 2010), pruned set (Read et al., 2008), classifier chains (Dembczyński et al., 2010; Read et al., 2011), preference ranking (Fürankranz et al., 2008), random label selection (Tsoumaks and Vlahavas, 2007; Tsoumakas et al., 2011a), and etc. In order to obtain satisfactory performance, these algorithms often adopt ensemble technique, which is quite time consuming as well. What's more, some algorithms treat unique label vector as a new label, which increases the number of labels exponentially. This will cost more computational effort for the learning task.

2.2. Existing LSDR methods

In order to alleviate the computation burden of existing multi-label classification algorithms, some researchers have made exploration on label space dimensionality reduction in multi-label classification problems. Algorithm 1 illustrates the general procedure of LSDR methods, in which $\mathbf{z}=\phi(\mathbf{y})\in R^{t\times 1}$, $\tilde{\mathbf{y}}=\varphi(\mathbf{z})\in\{-1,+1\}^{L\times 1}$ and $f: R^d\to R^{t\times 1}$ are the compression function, reconstruction function and learning function respectively for $\mathbf{y}\in\{-1,+1\}^L$. Here $t\le d$ is the dimensionality of reduced label space.

Algorithm 1. The general procedure of LSDR methods.

Training
1. Compression: use compression function $\phi(\cdot)$ to compress the label matrix $\mathbf{Y}$ and obtain $\{\phi(\mathbf{Y}_i)\}_{i=1}^N$;
2. Learning: use a certain algorithm to learn a model $f(\mathbf{x})$ mapping from $\mathbf{X}$ to $\{\phi(\mathbf{Y}_i)\}_{i=1}^N$

Testing
3. Prediction: make prediction $f(\mathbf{x})$ for a new instance $\mathbf{x}$;
4. Reconstruction: use reconstruction function $\varphi(\cdot)$ to obtain the final prediction $\varphi(f(\mathbf{x}))$.

A general objective function of LSDR is:

$$E(\boldsymbol{\theta})=E_e(\varphi(\phi(\mathbf{Y})),\mathbf{Y})+E_d(\phi(\mathbf{Y}),\mu(\mathbf{X})) \tag{1}$$

where $\boldsymbol{\theta}$ represents all parameters to be determined. For simplicity of presentation, we use $\phi(\mathbf{Y})$, $\varphi(\mathbf{Z})$ and $\mu(\mathbf{X})$ to represent the matrix stacked by $\phi(\mathbf{Y}_i)$, $\varphi(\mathbf{Z}_i)$ and $\mu(\mathbf{X}_i)$. (1) consists of two components: encoding loss $E_e(\cdot)$ and dependence loss $E_d(\cdot)$.$E_e(\cdot)$ measures the encoding (or compression) loss between the true label matrix $\mathbf{Y}$ and the reconstruction matrix $\tilde{\mathbf{Y}}$ whose ith row is the transpose of $\varphi(\phi(\mathbf{Y}_i))$ and $E_d(\cdot)$ measures the dependence loss between $\mathbf{X}$ and the code vectors $\{\phi(\mathbf{Y}_i)\}_{i=1}^N$. Here the dependence loss means a loss measuring the dependence between two variables and various loss functions that consider different kinds of dependence can be applied, such as no loss in Tai and Lin (2012), the prediction loss used in Hsu et al. (2009) and the regression loss in Chen and Lin (2012) and the HSIC loss used in this paper. According to difference of the ultimate goal, LSDR methods fall into two categories: learning methods and reduction methods.

Learning methods try to obtain a classifier after dimension reduction of label space. They place more emphasis on the dependence loss, i.e. E_d, and pay less attention on the encoding loss E_e. The CS method proposed in Hsu et al. (2009) exploits