Pattern Recognition 45 (2012) 1019–1034

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Two stage architecture for multi-label learning

Gjorgji Madjarov [a,b,*], Dejan Gjorgjevikj [a], Sašo Džeroski [b]

[a] Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, Rugjer Boshkovikj 16, 1000 Skopje, Macedonia
[b] Department of Knowledge Technologies, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 11 January 2011
Received in revised form
1 July 2011
Accepted 8 August 2011
Available online 16 August 2011

Keywords:
Multi-label learning
Multi-label ranking
Multi-label classification
Two stage architecture
Classifier chain

ABSTRACT

A common approach to solving multi-label learning problems is to use problem transformation methods and dichotomizing classifiers as in the pair-wise decomposition strategy. One of the problems with this strategy is the need for querying a quadratic number of binary classifiers for making a prediction that can be quite time consuming, especially in learning problems with a large number of labels. To tackle this problem, we propose a Two Stage Architecture (TSA) for efficient multi-label learning. We analyze three implementations of this architecture the Two Stage Voting Method (TSVM), the Two Stage Classifier Chain Method (TSCCM) and the Two Stage Pruned Classifier Chain Method (TSPCCM). Eight different real-world datasets are used to evaluate the performance of the proposed methods. The performance of our approaches is compared with the performance of two algorithm adaptation methods (Multi-Label k-NN and Multi-Label C4.5) and five problem transformation methods (Binary Relevance, Classifier Chain, Calibrated Label Ranking with majority voting, the Quick Weighted method for pair-wise multi-label learning and the Label Powerset method). The results suggest that TSCCM and TSPCCM outperform the competing algorithms in terms of predictive accuracy, while TSVM has comparable predictive performance. In terms of testing speed, all three methods show better performance as compared to the pair-wise methods for multi-label learning.

1. Introduction

The traditional problem of single-label classification is concerned with learning from examples, each associated with a single label λ_i from a finite set of disjoint labels $L=\{\lambda_1,\lambda_2,\ldots,\lambda_Q\}, Q>1$. For $Q>2$, the learning problem is referred to as a *multi-class classification*. On the other hand, the task of learning a mapping from an example $x \in X$ (X denotes the domain of examples) to a set of labels $Y \subseteq L$ is referred to as a *multi-label classification*. Thus, in contrast to multi-class classification, alternatives are not assumed to be mutually exclusive such that multiple labels may be associated with a single example, i.e., each example can be a member of more than one class. The set of labels Y are called relevant, while the set $L \backslash Y$ represents irrelevant labels for a given example.

Label ranking studies the problem of learning a mapping from a set of examples to rankings over a finite number of predefined labels. It can be considered a natural generalization of conventional (multi-class) classification, where instead of requesting only a single label (a top label), a ranking of all the labels is performed.

Besides the concept of multi-label classification, the multi-label learning introduces the concept of *multi-label ranking* [1], which is understood as learning a model that the query example x associates both with a (label) ranking of the complete label set $\{\lambda_1,\lambda_2,\ldots,\lambda_Q\}$ and a bipartite partition of this set into relevant and irrelevant labels.

The issue of learning from multi-label data has recently attracted significant attention from many researchers. They are motivated from an increasing number of new applications, such as semantic annotation of images and video (news clips, movies clips), functional genomics (gene and protein function), music categorization into emotions, text classification (news articles, web pages, patents, emails, bookmarks,…), directed marketing and others.

In recent years, many different approaches have been developed to solve the multi-label learning problems. Tsoumakas and Katakis [2] summarize them into two main categories: (a) algorithm adaptation methods, and (b) problem transformation methods. Algorithm adaptation methods extend specific learning algorithms to handle multi-label data directly. Examples include lazy learning [3–5], neural networks [6,7], boosting [8,9], classification rules [10], etc. Problem transformation methods, on the other hand, transform the multi-label learning problem into one or more single-label

* Corresponding author at: Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, Rugjer Boshkovikj 16, 1000 Skopje, Macedonia.
E-mail address: gjorgji.madjarov@finki.ukim.mk (Gj. Madjarov).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.08.011

classification problems. The single-label classification problems are solved with a commonly used single-label classification approach and the output is transformed back into a multi-label representation via some reverse process. A common approach for problem transformation is to use class binarization methods, i.e., decomposition of the problem into several binary sub-problems that can then be solved using a binary base classifier. The simplest strategies in the multi-label setting are the one-against-all and one-against-one strategies, also referred to as the binary relevance method [2] and pair-wise method [11,12], respectively. The computational complexity of the pair-wise learning approach to the multi-label scenario is large, especially in learning problems with a large number of labels.

In this paper, we propose a novel architecture for efficient pair-wise multi-label learning, named Two Stage Architecture (TSA). We analyze three different methods/implementations of this architecture the Two Stage Voting Method (TSVM), the Two Stage Classifier Chain Method (TSCCM) and the Two Stage Pruned Classifier Chain Method (TSPCCM). The two stage architecture and its three methods belong to the group of the problem transformation methods. Their main idea is to reduce the computational complexity of pair-wise methods and increase their predictive accuracy. We evaluate the performance of these methods on a selection of multi-label datasets that vary in terms of problem domain, number of labels and label cardinality. The obtained results demonstrate that our approaches outperform the competing methods (five problem transformation and two algorithm adaptation methods) in terms of predictive accuracy. Also, in terms of testing speed our architecture shows better performance as compared to the pair-wise methods for multi-label learning.

For the readers' convenience, Section 2 surveys some previous work in multi-label learning. The Two Stage Architecture and its computational complexity are presented in Section 3. Section 4 presents the experimental results, that compare the performance of the proposed approaches (TSVM, TSCCM and TSPCCM) with other competing methods. The conclusion and directions for further work are given in Section 5.

2. Related work

In this section, we will give an overview of different methods for solving multi-label learning problems. These methods can be summarized in two main categories: Algorithm adaptation methods and problem transformation methods. Additionally, the problem transformation methods can be grouped in three sub-categories: Binary relevance methods, label power-set methods and pair-wise methods.

2.1. Algorithm adaptation methods

AdaBoost.MH and AdaBoost.MR [8] are two extensions of AdaBoost for multi-label data. While AdaBoost.MH is designed to minimize Hamming loss, AdaBoost.MR is designed to find a hypothesis which places the correct labels at the top of the ranking. A combination of AdaBoost.MH with an algorithm for producing alternating decision trees [9] has been proposed, with the motivation of producing multi-label models that can be understood by humans.

Clare et al. [13] adapted the C4.5 algorithm for multi-label data (ML-C4.5). They modified the formula of entropy calculation (Eq. (1)) in order to solve the multi-label problem. They also allowed multiple labels in the leaves of the tree. The modified entropy sums the entropies for each individual class label.

$$entropy(S) = -\sum_{i=1}^{N}(p(c_i)\log p(c_i)+q(c_i)\log q(c_i)) \quad (1)$$

where S is the set of examples, $p(c_i)$ is the relative frequency of class c_i and $q(c_i)=1-p(c_i)$.

ML-kNN [3] is based on the popular k nearest neighbors (kNN) lazy learning algorithm. The first step in this approach is the same as in kNN, i.e., retrieving the k nearest examples. It uses the maximum a posteriori principle in order to determine the label set of the test example, based on prior and posterior probabilities, i.e., the frequency of each label within the k nearest neighbors. Other kNN based approaches for multi-label learning also exist [4,5].

Neural networks have also been adapted for multi-label classification [6,7]. BP-MLL [7] is an adaptation of the popular back-propagation algorithm for multi-label learning. The main modification to the algorithm is the introduction of a new error function that takes multiple labels into account.

2.2. Problem transformation methods

An extensive bibliography of learning algorithms for problem transformation methods is given by Tsoumakas and Katakis [2]. The simplest strategy in the multi-label setting is the one-against-all strategy also referred to as the binary relevance method (BR) [2]. It addresses the multi-label learning problem by learning one classifier for each class, using all the examples labeled with that class as positive examples and all remaining examples as negative examples. At query time, each binary classifier predicts whether its class is relevant for the query example or not, resulting in a set of relevant labels. In the ranking scenario, the labels are ordered according to the probability association of each label from each binary classifier. A method closely related to the BR method is the Classifier Chain (CC) method proposed by Read et al. [14]. This method involves Q binary classifiers as in BR. Classifiers are linked along a chain where each classifier deals with the binary relevance problem associated with label $\lambda_i \in L$, $(1 \le i \le Q)$. The feature space of each link in the chain is extended with the 0/1 label associations of all previous links. The ranking and the prediction of the relevant labels in the CC method are the same as in the BR method.

Second problem transformation method is the label combination method, or label power-set (LP) method, which has been the focus of several recent studies [15,16,2]. The basis of this method is to combine entire label sets into atomic (single) labels to form a single-label problem for which the set of possible single labels represents all distinct label subsets in the original multi-label representation. Each (x,Y) is transformed into (x,l) where l is the atomic label representing a distinct label subset. In this way, LP based methods directly take into account label correlations. To solve the problem of the large number of label combinations, Read [17] developed a pruned problem transformation method (PPT), that selects only the transformed labels that occur more than predefined number of times. A disadvantage of these methods, however, is their worst-case time complexity.

Third problem transformation approach to solving the multi-label learning problem by using binary classifiers is pair-wise classification or round robin classification [11,12]. Its basic idea is to use $Q*(Q-1)/2$ classifiers covering all pairs of labels. Each classifier is trained using the samples of the first label as positive examples and the samples of the second label as negative examples. To combine these classifiers, the pair-wise classification method naturally adopts the majority voting algorithm. Given a test example, each classifier delivers a prediction for one of the two labels. This prediction is decoded into a vote for