Neurocomputing 182 (2016) 187–196

Contents lists available at ScienceDirect

Neurocomputing

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

Neighbor selection for multilabel classification

Huawen Liu [a], Xindong Wu [b], Shichao Zhang [c,*]

[a] Department of Computer Science, Zhejiang Normal University, China
[b] Department of Computer Science, University of Vermont, USA
[c] Department of Computer Science, Guangxi Normal University, China

ARTICLE INFO

Article history:
Received 16 July 2015
Received in revised form
6 December 2015
Accepted 11 December 2015
Communicated by Shiliang Sun
Available online 21 December 2015

Keywords:
Multilabel learning
*k*NN
Shelly neighbor
Certainty factor

ABSTRACT

*k*NN is extensively studied for multilabel classification in the literature. Several *k*NN-based multilabel learning algorithms have been witnessed during the past years. They usually take *k*NN as their base classifiers to construct classification models, and then predict the class labels by virtue of Bayesian or majority rules. In this paper, a nearest neighbor selection for multilabel classification is proposed. Specifically, the target labels of new data are predicted with the help of those relevant and reliable data, which explored by the concept of shelly nearest neighbor. For effectiveness, the certainty factor is further adopted to well address the problem of unbalanced and uncertain data. The comparison experiments with eleven popular multilabel classifiers are conducted on ten benchmark data sets. The experimental results show that the performance of the proposed method is competitive and outperforms the popular multilabel classifiers in most cases.

1. Introduction

Pattern classification is a hot research topic in the fields of data mining, machine learning and pattern recognition. The learning process of a typical classification algorithm mainly consists of two stages: building a model from given training data and making a prediction for unknown data according to the generated model [9]. A common assumption of traditional learning algorithms is that the prediction result for each unknown or new data is only a single label (or value) of pre-specified class labels. This means that each data or instance can be tagged with only one class label out of two or more disjoint class labels, if the predicted result is generated by the traditional learning methods.

In reality, data instances are often associated with two or more class labels simultaneously. For example, the '*Avatar*' movie can be classified as different types, such as *action*, *horror* and *science fiction*; a document about financial storm can be labeled as *report*, *market*, *economic* and *politic*; a patient can be suffered from *colon cancer* and *flu* at the same time. Such kind of instances are called multilabel data [37]. Multilabel data are ubiquitous in real-world applications. Typical domains include text categorization, information retrieval and image processing [11].

In the multilabel situations, the traditional single-label learning techniques cannot work well. Multilabel learning derived from text categorization has been introduced to handle multilabel data [11]. Comparing to the traditional single-label learning, multilabel learning is more complicated and challenging. Given an unknown instance, a multilabel model outputs a set of class labels, hitherto called *labelset*, rather than a single label of the traditional learning at one time. From this view, the traditional learning is a special case of the multilabel learning. Since it has a great number of potential applications, in the last decades multilabel learning has attracted increasing attention from a board range of disciplines, including information retrieval, pattern recognition, data mining and machine learning [37], and has been successfully applied in many domains, such as text categorization, images retrieval and annotation, video and content annotation, music processing, and bioinformatics [11].

Generally speaking, the multilabel learning algorithms can be divided into two major categories: problem transformation and classifier extension [28]. The former transforms the multilabel data into the corresponding single-label ones with different strategies, while the later extends the traditional learning methods with some constraints, so that they can handle the multilabel data appropriately. Among the extension learning methods, the instance-based (i.e., lazy) multilabel learning has been extensively investigated. A representative example of this kind is BR*k*NN [29], which extends the classical lazy classifier, *k*NN (*k* nearest neighbors), for the multilabel data. Despite the lazy multilabel learning

* Corresponding author.
E-mail addresses: hwliu@zjnu.edu.cn (H. Liu), xwu@uvm.edu (X. Wu), zhangsc@gxnu.edu.cn (S. Zhang).

http://dx.doi.org/10.1016/j.neucom.2015.12.035
0925-2312/© 2015 Elsevier B.V. All rights reserved.

has achieved a considerable progress in recent years, several challenging issues are still left to be explored [25]. For instance, the optimal value of k, i.e., the number of nearest neighbors, in kNN is different for each data set and hard to be assigned. Moreover, kNN is sensitive to noisy data, resulting in poor performance of prediction [38].

In this paper, we propose a new lazy learning algorithm for the multilabel data by using two strategies. The first strategy used in our method is the concept of shelly nearest neighbor (SNN) [38], instead of kNN within the other lazy multilabel learning methods. SNN is a neighbor-instance selection method. Given an instance, its shelly nearest neighbors refer to those nearest neighbors that form a shell to encapsulate the instance [38]. From this perspective, SNN can get more reliable and true neighbor information when building a learning model. Besides, SNN can exempt from the cumbersome problem of choosing an optimal value of k in kNN for each data set. The second strategy of our method is the certainty factor (CF) [23]. The motivation is that in reality uncertain and unbalanced situations often occur [31]. Furthermore, the unbalanced property of data makes the situations worse, where the majority class certainly wins the minority class during the prediction stage in general [5]. To alleviate this problem, we exploit the certainty factor rule to determine the prediction results, after the shelly nearest neighbor information available.

The rest of this paper is organized as follows. Section 2 briefly reviews the recent work about the multilabel learning. In Section 3, the concepts of k nearest neighbors and shelly neighbors are given. Our lazy multilabel learning algorithm with two strategies is presented in Section 4, followed by the performance evaluation with other classifiers on public data sets in Section 5. Section 6 concludes this paper finally.

2. Related work

This section briefly reviews the state-of-the-art of multilabel learning methods. For more details about the multilabel learning, please refer to good surveys and references therein (see, e.g., [37,11,28]).

Formally, a multilabel data set is $D = \{(X_i, Y_i) | i = 1, \ldots, n\}$, where X_i denotes the i-th instance represented as a vector of p attributes and Y_i is the corresponding vector of class labels. It is noticeable that if only a single label is involved within each labelset Y_i, i.e., $|Y_i| = 1$, D is degenerated into a conventional data set. Naturally, an intuitive solution of the multilabel learning is to transform the multilabel data into the corresponding single-label ones. Generally, three strategies, i.e., *copy*, *selection* and *ignore*, are often used to transform the multilabel data [28]. For each instance $(X_i, Y_i) \in D$, the *copy* technique simply replaces it q times with (X_i, y_{ij}), each time with different y_{ij}, where $q = |Y_i|$ and $y_{ij} \in Y_i$. The *selection* strategy picks only one label y_{ij} out from Y_i and takes its place in (X_i, Y_i), while the *ignore* one does not take the instances with multiple labels into account when building learning models.

Binary relevance (BR) [28] is another kind of transformation technique, where each label is treated individually. Specifically, for each different label y_i, BR firstly generates a training data set D_i, in which the class label of each instance $X_k \in D_i$ is positive if $y_i \in Y_k$, and negative otherwise. Later these training data sets $\{D_1, \ldots, D_q\}$ are used to construct q binary classifiers. The final prediction results for an unknown instance can be determined according to these q classifiers, which will be combined into an overall one. From this point, the binary relevance belongs to ensemble learning [13,18].

It should be pointed out that the transformation techniques above have not taken the interrelations of the class labels into account. In real-world applications, the class labels are often relevant to each other. Taking this aspect into consideration, several multilabel learning methods exploit the relevancy of the labels to construct classification models. The label powerset (LP) [1] is such kind of learning method. It takes each subset of Y_i occurring within D as a new label, when building models. Then the labelset with the highest probable or a probability distribution over all labelsets will be outputted as the final prediction result for an instance. Note that the computational complexity of LP is relatively high, especially when the data set has a large number of labels or instances. To alleviate this problem, several variations of LP have been proposed. For instance, RAkEL [29] trains several classifiers with different k, i.e., the size of labelset, and then combines them together.

Pairwise correlation has also received attractions during the past years. For example, for each pair (y_i, y_j) of the class labels, the ranking pairwise comparison (RPC) [12] firstly generates a new data set D_{ij}, where for each instance $(X_i, Y_i) \in D$, if $y_i \in Y_i$ or $y_j \in Y_i$ exclusively, it will be added into D_{ij}. After the generating stage, RPC trains a binary classifier on each data set D_{ij}. The calibrated label ranking (CLRanking) [10] goes further by adding a virtual label to determine a natural breaking point between relevant and irrelevant labels. On the other hand, MLStacking [27] prunes the stacking models of BR by introducing correlation coefficient, which is used to estimate the correlation of each label pair.

The structures or high-order dependencies of the class labels have also been used to explore the multilabel data. For example, Brucker et al. [2] extracted hierarchical relations of labels via a neural network with an association rule learner. Wang et al. [30] adopted a Bayesian network structure to describe the conditional dependencies, which explored by maximum likelihood estimation, of the class labels. dependencies of the class labels. Charte et al. [4] took use of association rules to discover label dependencies. Ma et al. [17] adopted a generative probabilistic model to capture the correlations of labels. For the BR model, Montanes et al. [19] go further by developing the dependent binary relevance (DBR) method, which exploits the conditional dependencies of the class labels.

Several works resort the techniques of dimension reduction, including feature extracting and feature selection, to capture the correlations [8]. As a typical example, Li et al. [14] extended traditional pairwise constraints to project the multilabel data into a lower-dimensional space, while Liu et al. [15] performed the $\ell_{1,2}$ penalty on logistic regression to achieve the purpose of multilabel classification. Recently, the technique of partial least squares along with the ℓ_1 regularization have been utilized to explore the dependencies of the labels [16]. Zhao et al. [39] exploited group lasso to analyze facial expressions at one time instead of modeling as a binary learning problem. Shu et al. [24] did a similar work. Reyes et al. [21] extended the ReliefF algorithm for weighting and selecting features for multilabel data, while Zhang et al. [34] picked discriminative features for each label when constructing learning models.

The traditional learning methods, such as C4.5, SVM, ANN and AdaBoost, have been extended for the multilabel data by imposing some constrain conditions on them. AdaBoost.MH and AdaBoost. MR [22], which represents two different types of AdaBoost, are such kind of multilabel classifiers. Clare and King [7] employed C4.5 to deal with the multilabel data by altering the discriminative formula of information entropy, while Zhang and Zhou adopted anneal neural networks to handle the multilabel data [35]. Besides, SVM and core machine are also used to train classification models on the multilabel data by being assigned different parameters [33].

The lazy learning technique, kNN, is extensively studied in multilabel learning because of its simplicity, robustness and easy interpretation. The kNN-based multilabel learning algorithms obtain the final prediction results on the basis of the nearest