Pattern Recognition 47 (2014) 2517–2531

Contents lists available at ScienceDirect

Pattern Recognition

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

Multiple-instance discriminant analysis

Jing Chai [a,*], Xinghao Ding [b], Hongtao Chen [a], Tingyu Li [a]

[a] College of Information Engineering, Taiyuan University of Technology, Taiyuan 030024, PR China
[b] School of Information Science and Engineering, Xiamen University, Xiamen 361005, PR China

ARTICLE INFO

Article history:
Received 28 June 2013
Received in revised form
12 December 2013
Accepted 4 February 2014
Available online 13 February 2014

Keywords:
Multiple-instance learning
Feature extraction
Dimensionality reduction
Block coordinate ascent

ABSTRACT

Multiple-instance discriminant analysis (MIDA) is proposed to cope with the feature extraction problem in multiple-instance learning. Similar to MidLABS, MIDA is also derived from linear discriminant analysis (LDA), and both algorithms can be treated as multiple-instance extensions of LDA. Different from MidLABS which learns from the bag level, MIDA is designed from the instance level. MIDA consists of two versions, i.e., binary-class MIDA (B-MIDA) and multi-class MIDA (M-MIDA), which are utilized to cope with binary-class (standard) and multi-class multiple-instance learning tasks, respectively. The block coordinate ascent approach, by which we seek positive prototypes (the most positive instance in a positive bag is termed as the positive prototype of this bag) and projection vectors alternatively and iteratively, is proposed to optimize B-MIDA and M-MIDA to obtain lower dimensional transformation subspaces. Extensive experiments empirically demonstrate the effectiveness of B-MIDA and M-MIDA in extracting discriminative components and weakening class-label ambiguities for instances in positive bags.

1. Introduction

Different from traditional supervised learning where class labels are attached to instances and the goal is to predict the class labels of unseen instances, in multiple-instance learning only class labels of bags (a set of instances is termed as a bag) are known and the goal is to predict the class labels of unseen bags. A bag is classified as positive iff it contains at least one positive instance, and otherwise it is classified as negative. Fig. 1 depicts the comparison between supervised (mono-instance) learning and multiple-instance learning, where blue circles and red stars denote positive and negative instances, respectively. The collection of several instances with a rectangular contour represents a positive bag, while that with an ellipsoidal contour represents a negative bag. In subfigure (b), the number around each bag denotes the index of the bag, the prefixes "+" and "−" denote the positive and the negative classes, respectively. E.g., "+1" denotes this is the first positive bag, "−2" denotes this is the second negative bag. It is obvious that each object in supervised learning is an instance and that in multiple-instance learning is a collection of instances, i.e., a bag. Moreover, through Fig. 1, it is easy to see that whether containing at least one positive instance or not determines the class label of a bag.

The terminology "multiple-instance learning" was originally proposed by Dietterich et al. [1] when they were investigating the drug activity prediction problem. In their seminal paper, Dietterich et al. considered the problem of predicting whether a candidate drug molecule binds to the target protein or not. In particular, a molecule may take on many different shapes, and if any of these shapes conforms closely to the structure of the binding site, the candidate molecule binds to the target protein. By treating each shape of a molecule as an instance and each molecule as a bag, it is easy to see that drug activity prediction is a typical multiple-instance learning problem.

Besides drug activity prediction, multiple-instance learning appears in many other areas, such as image categorization [2–5], image retrieval [6–9], text classification [10,11], stock selection [10,12], protein sequence classification [8,13], computer aided diagnosis [14,15], and security application [16]. Zhou [17] gave a survey on the topic of multiple-instance learning and reviewed some important issues of this topic, such as the learnability, application domains, typical algorithms, and potential research scopes in the future.

In the past 15 years, multiple-instance learning has become very popular in the machine learning community, and researchers have proposed many representative algorithms to cope with various multiple-instance learning tasks. Maron and Ratan [2] studied the natural scene classification problem under the multiple-instance learning framework. They utilized diverse density (DD) to measure the closeness of a point to at least one instance in each positive bag and the remoteness of this point from all instances in negative bags, and then utilized the point with maximum DD as the "target concept" to operate classifications. Zhang and Goldman [18] combined DD and expectation maximization (EM) into a unified framework, and

* Corresponding author. Tel.: +86 15035101386; fax: +86 351 6010029.
E-mail address: jingchai@aliyun.com (J. Chai).

http://dx.doi.org/10.1016/j.patcog.2014.02.002
0031-3203

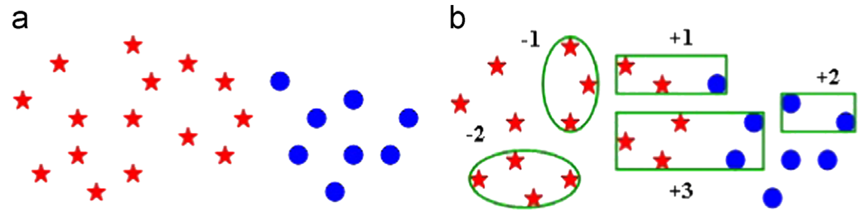

Fig. 1. Illustration of supervised learning and multiple-instance learning: (a) for supervised learning and (b) for multiple-instance learning. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

proposed the EM-DD algorithm to seek the point with maximum DD in an alternative and iterative way. Wang and Zucker [19] tried to utilize neighborhood information in multiple-instance learning and designed two *k*-nearest-neighbor (KNN) based multiple-instance classifiers: Bayesian-KNN and Citation-KNN. Andrews et al. [10] extended the support vector machines' (SVM) classifier to multiple-instance case and got MI-SVM and mi-SVM to cope with multiple-instance learning tasks. Gartner et al. [20] focused on the kernel design for multiple-instance data and proposed Multiple-Instance Kernel (MI-Kernel) to distinguish positive and negative bags. Zhou et al. [21] treated instances in each bag as independent samples and proposed two graph based algorithms (MI-Graph and mi-Graph) to mine the underlying structural information among within-bag instances. Zafra et al. [22] extended the traditional ReliefF algorithm [23] to multiple-instance case and proposed the ReliefF-MI algorithm to cope with multiple-instance feature selection tasks. Li et al. [24] studied the multiple-instance learning problem by assuming that instances are modeled as a mixture of concept and non-concept distributions, and thus classified a bag as positive if the fraction of concept instances in it was larger than a particular threshold. Zhang et al. [25] treated automatically grouping motion patterns in traffic scenes as a multiple-instance learning problem, and then proposed the Maximum Margin Multi-instance Multi-cluster Learning (M^4L) algorithm to cope with this problem.

Standard multiple-instance learning consists of two classes, i.e., a positive class and a negative class. However, with the rapid development of multiple-instance learning, its application domain has been extended from the binary-class case to the multi-class case [4,7,10]. In multi-class multiple-instance learning, for each given class, if any instance in a bag represents the class label of the given class, we say this instance is positive for the class, and hence, this bag is positive for the class as well; otherwise, this bag is negative for the class. Note that in multi-class multiple-instance learning, usually we do not define the specific negative bags for each class, because positive bags for some class can be simultaneously treated as negative bags for other classes, e.g., positive bags for class c are also negative bags for classes except c.

Similar to other machine learning branches such as supervised, unsupervised and semi-supervised learning, the feature extraction problem exists in multiple-instance learning as well, e.g., the multiple-instance data may also contain noisy and redundant components, the curse-of-dimensionality problem may also occur in high dimensional applications. Through feature extraction, we may reduce data's dimensionality and save memory space, remove useless and noisy components, reduce the time complexity in testing phase, weaken the disadvantage caused by the curse-of-dimensionality problem, and improve classification accuracies. In the past few years, several researchers have studied the multiple-instance feature extraction problem and proposed several dimensionality reduction algorithms. E.g., Sun et al. [26] designed a probabilistic multiple-instance dimensionality reduction algorithm, namely Multi-Instance Dimensionality Reduction (MIDR), and proposed to solve it by gradient descent along the tangent space of the orthonormal projection matrix; Ping et al. [27] utilized the structural information conveyed by instances in a bag to learn lower dimensional representations of original data, and designed an algorithm named as Multi-Instance Dimensionality reduction by Learning a mAximum Bag margin Subspace (MidLABS); Kim and Choi [28] proposed the Citation Local Fisher Discriminant Analysis (CLFDA) algorithm to utilize the citation and reference information in detecting false positive instances and extracting local discriminative information for multiple-instance learning.

Linear Discriminant Analysis (LDA) [29], which utilized the class-label information to maximize the ratio of between-class scattering to within-class scattering, was a classical supervised feature extraction algorithm and had been successfully applied in many supervised learning tasks [30–35]. In this paper, we propose Multiple-Instance Discriminant Analysis (MIDA), an extension of LDA, to cope with the multiple-instance feature extraction and dimensionality reduction problems. Since there are two kinds of multiple-instance learning problems, i.e., the binary-class one and the multi-class one, the proposed MIDA algorithm has two versions as well, which can be abbreviated as B-MIDA (Binary-class MIDA) and M-MIDA (Multi-class MIDA), respectively. Note that the above mentioned MidLABS algorithm can be treated as multiple-instance extension of LDA as well, and hence our MIDA work is very similar to MidLABS. Both MIDA and MidLABS try to maximize the trace of the between-class scattering matrix and minimizes that of the within-class one simultaneously, but their design principles are very different, because they construct the scattering matrices from different levels. MIDA constructs the scattering matrices from the instance level, i.e., it selects a prototype for each bag and utilizes this prototype as the representative of the bag to construct scattering matrices. In contrast, MidLABS constructs the scattering matrices from the bag level, i.e., it directly evaluates the similarity and scattering among bags. Moreover, since MIDA is derived from LDA, some limitations of LDA such as the unavailability for multi-modal data and the independently and identically distributed (i.i.d.) assumption for instances in the same class also exist in MIDA. Hence, although the experimental results shown in Section 5 demonstrate that MIDA performs well in extracting discriminative components, it may still be improved in the future.

Note that the main difference of multiple-instance learning from supervised learning is that there are class-label ambiguities for instances derived from positive bags. If we can find out the most positive instance in each positive bag, the disadvantage caused by the class-label ambiguities may be weakened and we may utilize supervised techniques to design multiple-instance feature extraction algorithms. Therefore, both B-MIDA and M-MIDA contain two types of unknown variables, of which the first type are positive prototypes (the most positive instance in each positive bag is termed as the positive prototype of this bag), the second type are projection vectors. It is difficult to optimize the two types of unknown variables simultaneously, because they are neither jointly convex w.r.t. (with respect to) the objective function nor can be optimized with analytical solutions. Instead, we utilize the block coordinate ascent approach [36] to update the above two types of unknown variables alternatively and iteratively. In each iteration, first we fix one type of unknown variables and update the other type of ones, then alternate the order of the above two types of unknown variables and update the fixed type of ones in last step. We repeat the above two steps iteratively, until