Pattern Recognition 48 (2015) 2831–2838

Contents lists available at ScienceDirect

Pattern Recognition

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

Single- vs. multiple-instance classification

Ethem Alpaydın [a,*], Veronika Cheplygina [b], Marco Loog [b,c], David M.J. Tax [b]

[a] Department of Computer Engineering, Boğaziçi University, 34342 Istanbul, Turkey
[b] Pattern Recognition Laboratory, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
[c] The Image Group, University of Copenhagen, Universitetsparken 5, 2100 Copenhagen, Denmark

ARTICLE INFO

Article history:
Received 1 September 2014
Received in revised form
16 February 2015
Accepted 3 April 2015
Available online 16 April 2015

Keywords:
Classification
Multiple-instance learning
Similarity-based representation
Bioinformatics

ABSTRACT

In multiple-instance (MI) classification, each input object or event is represented by a set of instances, named a bag, and it is the bag that carries a label. MI learning is used in different applications where data is formed in terms of such bags and where individual instances in a bag do not have a label. We review MI classification from the point of view of label information carried in the instances in a bag, that is, their sufficiency for classification. Our aim is to contrast MI with the standard approach of single-instance (SI) classification to determine when casting a problem in the MI framework is preferable. We compare instance-level classification, combination by noisy-or, and bag-level classification, using the support vector machine as the base classifier. We define a set of synthetic MI tasks at different complexities to benchmark different MI approaches. Our experiments on these and two real-world bioinformatics applications on gene expression and text categorization indicate that depending on the situation, a different decision mechanism, at the instance- or bag-level, may be appropriate. If the instances in a bag provide complementary information, a bag-level MI approach is useful; but sometimes the bag information carries no useful information at all and an instance-level SI classifier works equally well, or better.

1. Introduction

In pattern recognition, the object or event to be classified is denoted by an instance x represented as a d-dimensional vector of features. The training set is composed of N such instances and their labels, $\mathcal{X} = \{x^t, r^t\}_{t=1}^N$, where r^t is the class label of x^t. Here (without loss of generality), we focus on two-class classification where instances are negative, i.e., $r^t = -1$, or positive, $r^t = +1$. The aim is to learn a classifier $f(x^t)$ using this training set of instances.

In the framework of *multiple-instance* (MI) learning [1,2], each object or event is represented by a *bag* b^t. A bag is an unordered set of instances and different bags may contain different number of instances:

$$b^t = \{x_1^t, x_2^t, \ldots, x_{n^t}^t\}$$

where n^t is the number of instances in bag t. The training set is now denoted as $\mathcal{X} = \{b^t, r^t\}_{t=1}^N$ where $r^t \in \{-1, +1\}$ is the class label of bag b^t. Single-instance (SI) classification is a special case where each bag contains only one instance: $b^t = \{x_1^t\}$. In the multiple-instance case, the classifier works at the bag level and takes a bag as its input, $g(b^t)$, and generates a decision for the bag.

MI learning is applicable when the data is generated as a bag of instances all somehow related (for example because all are due to the same hidden cause or factor)—there is a label for the whole but not for the individual instances. Since its original definition [3], MI learning has been used in different applications where the only common characteristic is that inputs are bags of instances, but different MI learning methods assume different types of relationships between instances, bags, and hence class labels [1,2].

For example in the original *Musk* drug activity prediction, a molecule (bag) has the desired drug effect (positive label) if and only if one or more of its conformations (instances) bind to the target site; we do not know a priori which one, so we cannot label the instances individually, and we have an overall label for the whole molecule.

As opposed to this, a relatively recent application of MI is in image classification where we want to label a scene, e.g., beach, sea, and desert. The image (bag) is segmented into small patches (instances) and for example we have a beach image (positive label for the beach class), if we have a “sand segment” *and* a “sea segment” (Desert class is defined as a “sand segment” and no “sea segment”). Here the problem, though is still MI, is quite different from Musk; instances are subparts and are not at the same level of abstraction as bags and

* Corresponding author.
E-mail address: alpaydin@boun.edu.tr (E. Alpaydın).

http://dx.doi.org/10.1016/j.patcog.2015.04.006
0031-3203/

therefore, labels at the level of bags, e.g., beach, are not applicable for instances.

Because of these reasons, though we see numerous applications of MI in the literature and various learning methods having been proposed, the MI approach does not always lead to improved performance [1,2]. It seems that MI learning is sometimes being used without a meticulous investigation of its assumptions and concomitant restrictions.

We believe that because of such significant differences in the underlying characteristics of the MI problems, it may be futile to look for a single MI learning algorithm that can work successfully on all, just because they can all be defined in terms of bags. We propose that a more fruitful approach may be to categorize the different MI problems in terms of their characteristics and then for each category, define the requirements for an MI learning algorithm. Such a categorization also better differentiates MI learning from SI learning.

To summarize, in this paper, we compare SI and MI learning to be able to clarify what the MI framework brings over SI; our aim is not to compare the already numerous MI algorithms or propose a new MI algorithm, but rather to determine when casting a problem in the MI framework is preferable to SI, and also define the different MI categories.

More specifically, we make a distinction between MI problems based on the amount of label information carried by the instances in a bag, that is their self-sufficiency for classification, or inversely, the amount of complementary information carried by the instances in a bag, which we name intra-bag dependency. Towards this aim, we create a sequence of synthetic classification problems of increasing complexity, which corresponds to increasing the intra-bag dependency, and we use these to assess and compare the discriminative power of SI and MI learning.

This paper is organized as follows: In Section 2, we discuss the spectrum of MI problems. In Section 3, we discuss the instance- and bag-level classifiers we use in this study and in Section 4, we define the synthetic tasks we use to assess SI and MI approaches; we also use them as canonical tasks to quantify the power of different MI learning algorithms. We give our experimental results on two sets of real-world bioinformatics data for gene expression and text categorization in Section 5. We discuss our findings and conclude in Section 6.

2. The spectrum of multiple-instance problems

We categorize MI problems by the amount of information each instance in a bag carries about the label:

(1) On one extreme lies the pure instance-level approach. Each instance can be assigned a label and carries enough information for classification so that its vectorial representation is sufficient for it to be classified correctly. In this case, there is no need to take into account the other instances in the bag and hence no need for the MI approach. The instances in a bag are labelled with the bag label and we can train an instance-level classifier $f(x^t)$. The instances in a bag are assumed independent: the bag information, namely, whether two instances are in the same bag or in different bags, is assumed to be useless and can be disregarded.
For example, if each bag contains a number of face images of the same person, e.g., from different poses or lighting conditions, and if each image in a bag is detailed and informative enough for recognition, then there is no need to define bags for people. In such a case, the whole operation, including both training and testing, can work at the instance level. We can just train and use an instance-level classifier $f(x^t)$ that takes a single image x^t and makes a decision. As the individual face images deteriorate, for example due to bad lighting or occlusion, and become less informative, making use of other instances for complementary information, that is, the MI approach starts making sense.

(2) In the earliest work on MI learning [3], the assumption made was that *a positive bag contains at least one positive instance*. Here, it is assumed that instances carry labels, that is actually they can be classified as instances, but it is not known which one(s) carry the label, and because we lack label information at the instance level, we use the MI approach.
Let us say we have face images of people in a meeting and that we know one of the faces belongs to the person we want to identify but we do not know which. Then we have a multiple-instance problem where the faces in the meeting define a bag. In the bag, there is one instance which is the "real" positive instance; the other instances actually are uninformative but we cannot get rid of them because we do not have label information at the level of instances.
The approach in such a case is to train an instance-level classifier, and combine its decisions on the instances in the bag to get a bag-level decision:

$$g(b^t) = \phi(f(x_1^t), f(x_2^t), \ldots, f(x_{n^t}^t))$$

The assumption that the positive decision of at least one instance classifier is sufficient for the bag decision implies the noisy-or as the combination function [4], but note that the best $\phi()$ depends on the application; for example, noisy-or may lead to a high rate of false positives and when positive bags contain a higher percentage of positive instances, named the "witness" of the bag [5], majority vote may be better.
This approach where the bag-level decision is formed by combining instance-level decisions is named the *collective* approach, and various methods have been used for training the instance-level classifiers and for their combination [6–8]. When we have bags where some of the instances are positive and the rest have indeterminate labels, we can also view this as a semi-supervised learning problem and can handle it as such [9]. Fusing the decisions for instances to arrive at a decision for the bag can also be viewed as an ensemble method, where learners each with a different instance as its input make a decision and a combiner calculates the overall output [10], e.g., by majority voting.

(3) On the other extreme, an instance in a bag has no label because an instance by itself carries only a portion of the information necessary for classification. In such a case, a bag-level classifier should be used.
As an example, let us say that from a single face image, we take small patches, e.g., part of an eye and chin as instances, and all these patches together make up the bag that represent the face. In such a case, each patch by itself is not informative enough and no label can be attached, and hence no instance-level classifier $f(x)$ can be trained. We need a bag-level representation corresponding to the complete image and a bag-level classifier that uses the collective information from all the patches, x_i^t.
There are two possibilities: In the *bag-space* approach [2], we use a distance function $d(x_i^r, x_j^s)$ for the distance between instances i and j, respectively, from bags r and s, and we use these to calculate the distance between bags r and s (typically by taking average, minimum, or maximum between all possible pairs). Once we define such a distance between bags, we can use k-nearest neighbor or any variant, or support vector machines with a kernel defined through such a distance function. Another possibility is to directly define a kernel over bags measuring the similarity of two bags in terms of the underlying data structure used to represent the bags; for example, in [11], each bag is represented by a graph and graph kernels are used with support vector machines.