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.

The other possibility is to use a single vectorial representation, v^t, formed for the bag using all the instances in the bag, for example, through calculation of some statistics over the bag. This is the *metadata* [1] or *embedded-space* [2] approach. Once the bag is represented by such a vector, it is fed to the bag-level classifier $g(v^t)$.

Previously researchers have come up with taxonomies for MI problems. Weidmann's hierarchy [12], which we use below, is extended by Foulds and Frank [1]. Amores's taxonomy [2] is similar except that the names are slightly different.

3. Classifiers

We use the support vector machine (SVM) with a linear kernel as the base classifier and adapt it to various MI scenarios. We use the same classification algorithm in all cases to make sure that any difference is due to the representation and not due to the classification algorithm:

(1) *Instance-level classifier SIL-SVM*: In this case, instance t takes the label of its bag and an instance-level SVM classifier $f(x^t)$ is trained with these instances. The assessment during test is also done at the instance level, again by assigning bag labels to the test instances. This corresponds to discarding totally the bag information during both training and testing. If such a classifier works fine, we understand that the bag information does not bring much, that is, it is not actually necessary to define such a problem as a multiple-instance problem.

(2) *Combined instance-level classifier NOR-SVM*: Instances take the labels of their bags and an instance-level SVM classifier is trained as above, but during test, the decision is made at the bag level by combining the instance-level SVM $f(x_i^t)$ decisions in the bag using noisy-or. This is the so-called *standard MI* classifier [13].

(3) *Bag-level classifier MIL-SVM*: For each bag t, a vectorial representation v^t is formed summarizing the information in the instances in the bag, and a bag-level classifier $g(v^t)$ is trained. During test, a similar representation is formed for the bag query and is given as input to the classifier.
MIL-SVM is an SVM classifier that uses the bag-based similarity representation classifier where we represent each bag by a vector of dissimilarities to all the other N bags in the training set [14,15]. As such, it is a combination of the bag-space and embedded-space approaches discussed above:

$$v^t = [d(b^1, b^t), d(b^2, b^t), \ldots, d(b^N, b^t)]^T \quad (1)$$

The dissimilarity between bags b^r and b^s is defined as

$$d(b^r, b^s) = \frac{1}{n^r}\sum_i \min_j \|x_i^r - x_j^s\|^2 \quad (2)$$

That is, for each instance i in bag r, we find the most similar instance from bag s in terms of squared Euclidean distance, and then we average this over all such i in bag r. This gives us a distance, or a dissimilarity score, between bags r and s. The N-dimensional representation for bag t, which we denote by v^t (Eq. (1)), is formed as a vector of such dissimilarities between bag t and all the bags in the training set. This new v^t is then as the input to a support vector machine which now works at the bag-level, $g(v^t)$. We name this classifier MIL-SVM.

(4) *Bag-level classifier* MILES: This also uses a bag-level similarity representation and a linear classifier with L1 regularization [13]. All instances t in the training set are considered to be prototypes and every bag, both at training and at test time, is represented by its distances to all the training instances. Using the definition in Eq. (2), every feature vector for a bag b^s consists of the distances $d(t, b^s)$ to all t. The subsequently used L1-regularized linear classifier makes sure that one can deal with the high dimensionality of this representation.

There are various MI algorithms proposed in the literature, acting somewhere in the range between (2) and (3, 4) above; see [1,2] for two very detailed reviews. We do not compare those different MI algorithms here, because our aim is not to compare the different MI approaches among themselves, but rather to compare MI with SI, and determine when casting a problem in the MI framework is preferable to SI.

4. Synthetic MI tasks

We use the synthetic example from [13] and convert it to a setting where MI tasks of increasing complexity can be defined. This is a two-dimensional problem with five Gaussian components centered at $(5,5)$, $(5,-5)$, $(-5,5)$, $(-5,-5)$, $(0,0)$, all having unit diagonal covariance matrices (see Fig. 1).

The increasing complexities of the tasks we define roughly correspond to Weidmann's hierarchy [12]. These can be considered as canonical problems of increasing complexity and a more complex task implies a more complex dependency between instances in a bag, which we call intra-bag dependency. The level that can be handled by an MI algorithm is an indicator of the power of that MI algorithm.

We generate the data as follows: For each bag, we first sample the bag size m from a uniform distribution between 1 and 8 and then we generate m instances by randomly drawing from one of the Gaussians mentioned above with equal prior probability. For each task, we define the rule for a bag to be positive; so if the generated instances satisfy the rule, the bag is labelled positive, otherwise the bag is labelled negative. We generate the data so that each data set contains equal number of positive and negative bags. For each task, to test for the effect of training set size, we do four different experiments where the number of bags used in training is 10, 20, 50, and 100; testing is done on separate sets containing 100 bags sampled in the same way. For each task and for each bag size, we do 10 independent runs using a different randomly generated sample of training and test sets. The performance measure is the area under the ROC curve. In the graphs, we plot averages of those 10 runs with one standard deviation error.

The performance of bag-level classifiers, NOR-SVM, MIL-SVM, and MILES, are measured on the bag data, whereas the performance of the instance-level classifier SIL-SVM is measured on the instance data created from the bag data by labeling the instances with their bag labels. Strictly speaking, the bag-level and instance-level data are different and hence it is not fair to compare the AUC of SIL-SVM with the AUC of others. However note that our aim is not to look at the absolute AUC value of SIL-SVM but its relative position with respect to those of the bag-level classifiers: if the

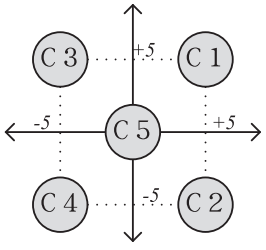

Fig. 1. The synthetic MI tasks are generated by different combinations of these five Gaussian components.

AUC value of SIL-SVM is almost as high as the AUC values of the MI variants, we can say that the bag-level approach does not bring much, but where there is a large difference is where the MI approach hits the mark.

Let us say we have a bag with four instances and let us say it is a positive bag. So if a bag-level classifier classifies it as a positive bag, we have one correct (bag-level) decision. The four instances in the positive bag translate to four positively labelled instances in the instance-level data. Let us say that our instance-level classifier classifies two of them as positive and two as negative, so we have two correct and two erroneous (instance-level) decisions. Hence here, the bag-level approach makes sense.

4.1. Task 1

Task definition: The instances of a positive bag are drawn from any of components 1, 2, or 5, and the instances of a negative bag are drawn randomly from any of components 3, 4, or 5.

An example data set is shown in Fig. 2(a). This is a relatively simple problem and all classifiers have quite high AUC values. The instances in the positive bags are relatively easily distinguishable from instances in the negative bags and in Fig. 2(b), we see that the instance-level classifier SIL-SVM is already very accurate; from an instance-level point of view, the two classes only overlap in component 5 and this causes some error. NOR-SVM that does noisy-or combination can correct for these by using the other instances in the bag, but still makes some errors. The bag-level classifiers, MIL-SVM and MILES, work the best because they make better use of all the instances and do not rely on the instance-level classifier. Note that when there are few bags (here, 10), there is not enough data to train the bag-level classifiers and we see that the MI approaches are not any better than SI. The fact that a bag-level data is smaller than its instance-level version is an important point and we will discuss it further.

Overall, this can be considered as an example where the instance-level approach works well and the MI approaches add some, but not too much.

4.2. Task 2

Task definition: A bag is positive if it contains at least one instance from component 1.

An example data set is shown in Fig. 3(a). This is a more complex task where bag-level information is needed. A negative bag does not contain any instance from component 1 but a positive bag can also contain instances from other components. This corresponds to the *standard MI level* in Weidmann's hierarchy.

As we see in Fig. 3(b), the instance-level SIL-SVM does not do well, because there are both positive and negative instances from components 2 to 5; so the only way is to check if the bag contains an instance from component 1. Noisy-or can check for this and NOR-SVM does much better, but it does not do perfectly because the underlying instance-level SVM classifier is not a very good one (due to the mixed labeling of instances). The bag-level classifiers, MIL-SVM and MILES, have no problem of classification.

4.3. Task 3

Task definition: A bag is positive if it contains at least one instance from component 1 and one instance from component 4.

An example data set is shown in Fig. 4(a). This is a more complex task corresponding to the *presence-based level* in Weidmann's hierarchy. An example is given in [1]: to be able to say that we have the image of a beach, we need to have patches both from sand and from sea; if we see only sand patches, it can be a desert image; if we see only sea patches, it can be a seascape.

A negative bag cannot contain instances from both components 1 and 4, but can contain instances from either one; positive bags can contain instances from all components.

As we see in Fig. 4(b), the instance-level SIL-SVM does not do well, NOR-SVM does much better, but the best are the bag-level MIL-SVM and MILES. Note that the accuracies of these bag-level classes reach the maximum possible value of 1.0 with enough (bag-level) data.

4.4. Task 4

Task definition: A bag is positive if it contains at least two instances from component 1 and at least two instances from component 4.

An example data set is shown in Fig. 5(a). This is an even more complex task. A negative bag can contain instances from components 1 and 4, and even two from one of the components and one

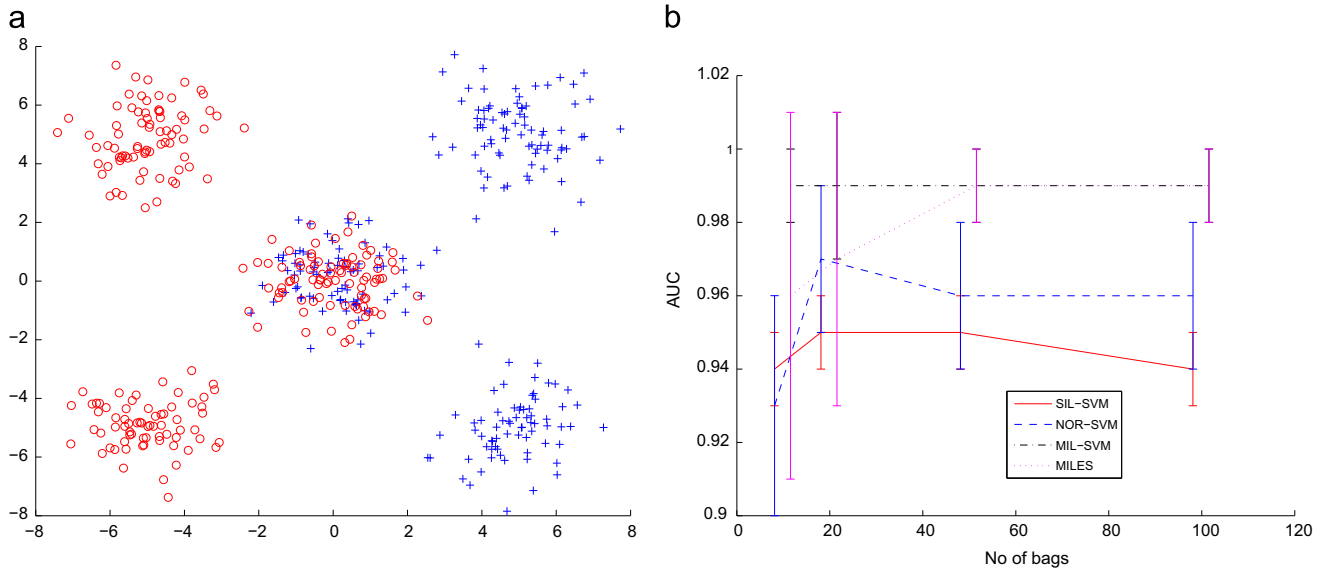

Fig. 2. On task 1, comparison of SIL-SVM, NOR-SVM, MIL-SVM and MILES, for bag sizes of 10, 20, 50, and 100. (a) An example data set where instances are marked by the bag labels: '+' positive, 'o' negative. (b) Area under the ROC curve values on the test sets; these are average and one standard deviation error bars of 10 independent runs.

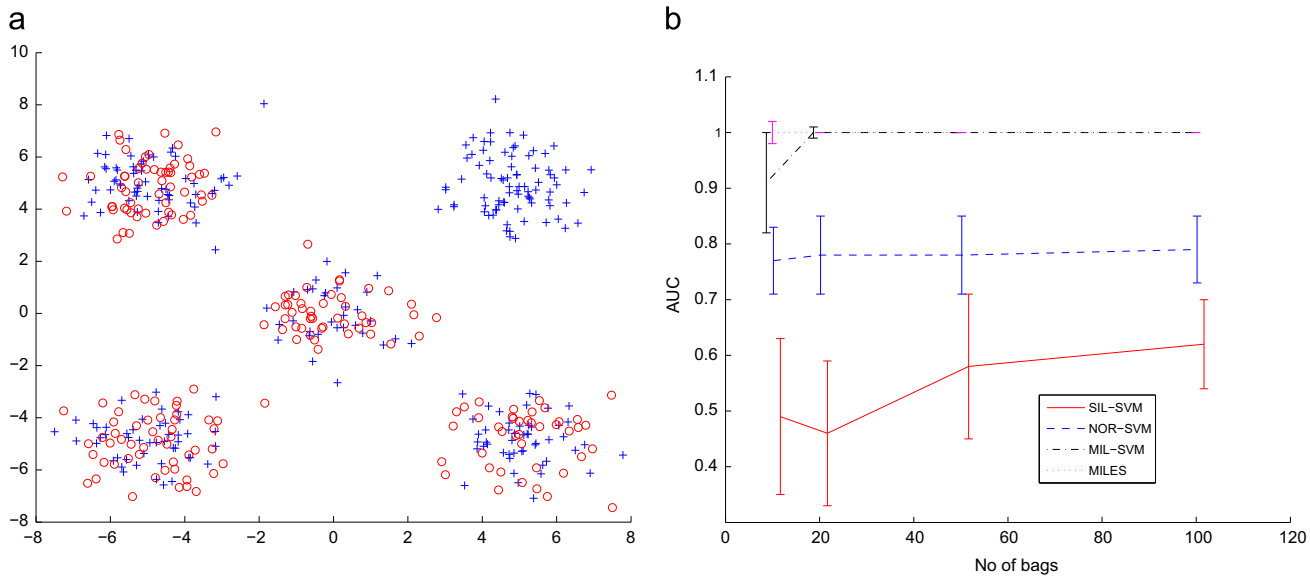

Fig. 3. Task 2: (a) Example data and (b) performances.

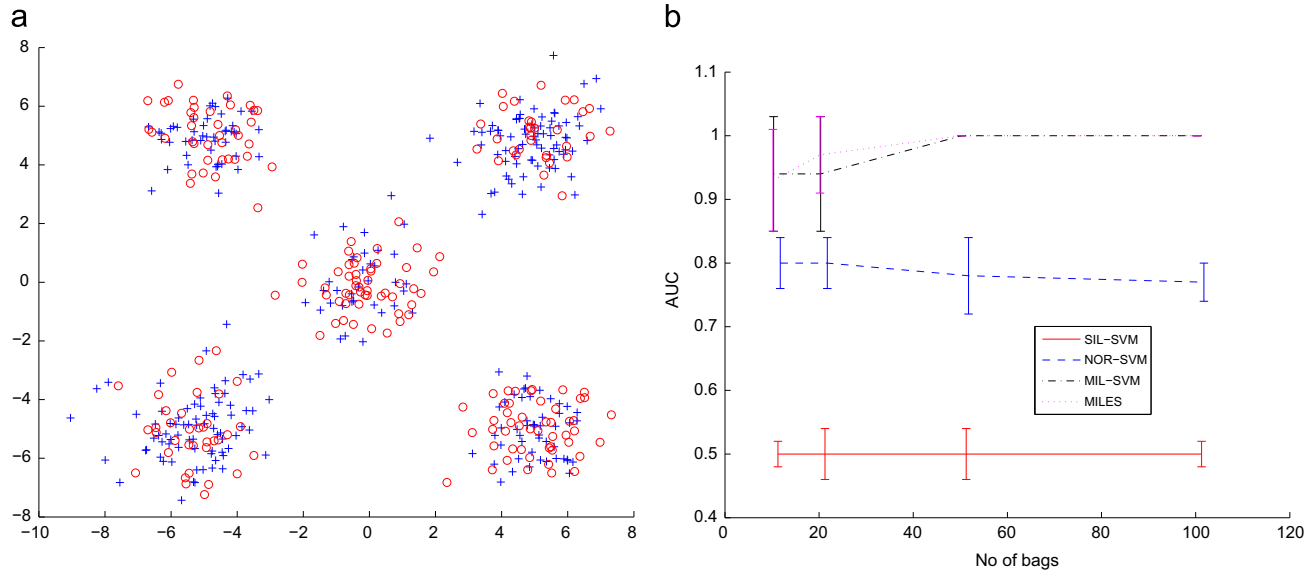

Fig. 4. Task 3: (a) Example data and (b) performances.

from the other; positive bags contain instances from all components. This corresponds to the *threshold-based level* in Weidmann's hierarchy.

As we see in Fig. 5(b), the instance-level SIL-SVM does badly, NOR-SVM does better because now there may be more instances in positive bags. In terms of the bag-level classifiers, MIL-SVM and MILES using bag similarities work better. Note that this is quite a difficult task—we are not just checking for the presence, but the occurrence of a number of times. We see that even bag-level MIL-SVM or MILES cannot learn perfectly, even when they are trained with 100 bags per class.

We believe that these four tasks can be interpreted as canonical tasks at different levels of intra-bag dependency, measuring the discriminative power of MI algorithms. Looking at the accuracies of the methods we discussed, we can say that the instance-level SIL-SVM can handle Task 1 but not the others. NOR-SVM can handle Tasks 1 and 2, but not 3 or 4. MIL-SVM and MILES can handle up to Task 3 but not Task 4. How MIL-SVM or MILES can be extended to be able to handle Task 4 is an open future research topic.

5. Experiments on bioinformatics applications

5.1. Gene expression data

In our first batch of experiments on real-world data, we focus on four databases for gene expression. This original study focused on the identification of genomic features that influence the binding of transcription factors responsible for gene expression. This is interesting because it allows for the identification of the mechanisms that are responsible for cell differentiation in tissues [16].

The multiple-instance framework is suitable here because there are multiple transcription factors that need to bind in order for a gene to express. Therefore, each gene is considered to be a bag that

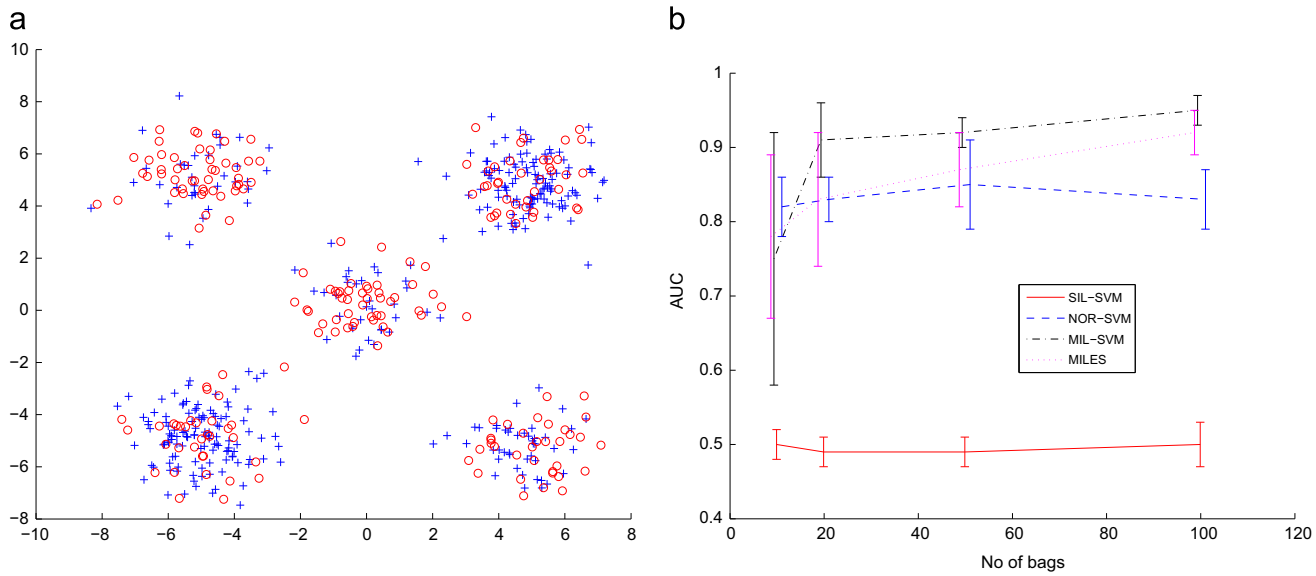

Fig. 5. Task 4: (a) Example data and (b) performances.

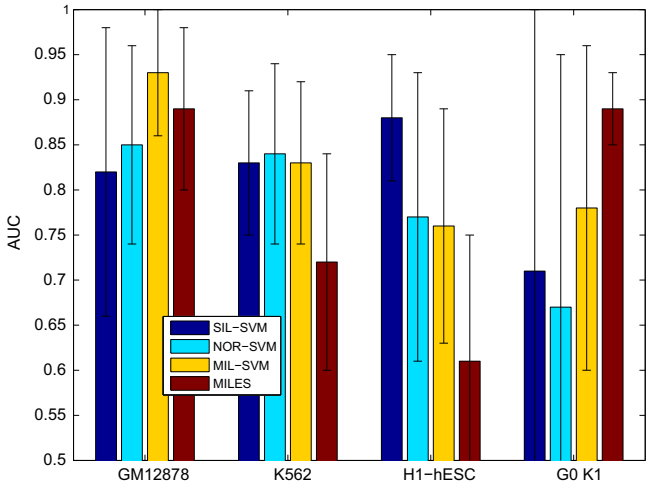

Fig. 6. Comparison on four different gene expression data sets.

contains a collection of instances. The instances are the transcription factor binding sites that are characterized by 100 genomic features, that describe, among other things, the methylation status of the binding sites, the histone modifications, the openness of the chromatin. Four different cell types were investigated in which gene expressions were measured, and these make up the four different MIL data sets, which are named GM12878, K562, H1-hESC, and G0 K1. Further details on the origin of the data, the pre-processing steps, and the features can be found in [16].

We use five-fold cross-validation and our results on the four data sets are shown in Fig. 6. This seems to be a problem similar to Task 1 above; we see that there is not a significant difference between SI and MI approaches. Neither noisy-or combination nor bag-level SVM nor MILES work better—on one of the four data sets, SIL-SVM is even the most accurate. We note hence that using the MI framework here is not necessary for higher classification accuracy; other instances in the bag do not add extra information.

Note however that MIL formulation may have its other benefits. For example, the MIL formulation allows the use of the MILES classifier which provides the possibility of inspecting the informativeness of each instance, that is, the transcription factor. When knowledge extraction and the interpretability of the solution are also important, the MIL approach with MILES is still interesting and useful, though it may not be better in terms of classification accuracy.

5.2. Text categorization data

Our second batch of real-world data sets concerns biomedical text categorization [17,18]. The goal is to annotate article–protein pairs with codes from the Gene Ontology (GO). An article–protein pair should be annotated with a GO code if the article contains text that links the protein to the GO code.

Each article is described by a bag of paragraphs, and each paragraph is described by a 200-dimensional feature vector with word counts, as well as statistics about the co-occurrences of the protein and the GO code in that paragraph. The assumption is that an article should be annotated with a GO code if and only if there exists a paragraph in it that supports this annotation, which gives rise to a multiple-instance problem.

There are three types of GO codes, which refer to cellular components, biological processes, or molecular functions. There are therefore three distinct MIL problems to consider. Each problem has a predefined training set with equal class priors (359, 385 and 620 bags per class), and a highly imbalanced test set (64, 58 and 137 positive bags vs. 2K, 4K and 10K negative bags).[1]

We use five-fold cross-validation and report AUC both on the cross-validation data and the results on a separate held-out test set; see Fig. 7. This is a data set where the predefined training and test sets differ considerably in terms of prior probabilities, and that is why it is instructive to look at accuracies on both cross-validation and test results. We see in Fig. 7(a) that on the cross-validation data, NOR-SVM and MIL-SVM are more accurate than SIL-SVM (these are large data sets and we could not run MILES on them). There does not seem to be a significant difference between NOR-SVM and MIL-SVM here.

On the test data, as we see in Fig. 7(b), again NOR-SVM and MIL-SVM are more accurate than SIL-SVM, but here, NOR-SVM seems better than MIL-SVM. We believe that this is an indication that MIL-SVM trained on the training set does not generalize as

[1] These data sets are available for download in MATLAB format from http://www.miproblems.org.

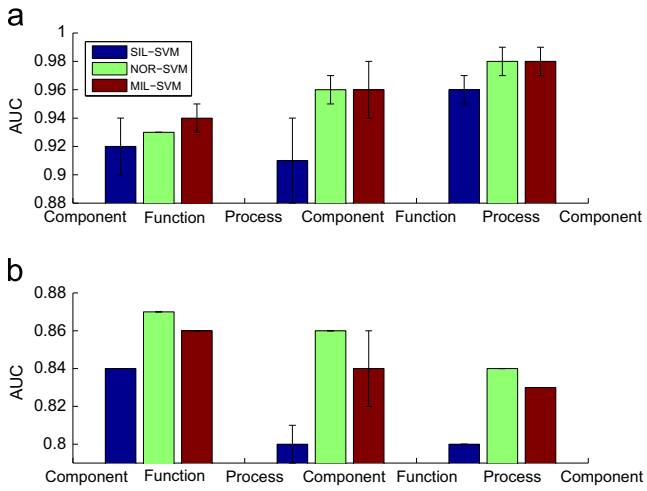

Fig. 7. Comparison on three biocreative data sets: (a) Cross-validation and (b) test performances.

well to the test set because it has fewer training data. Component data set for example has 620 positive bags and 620 negative bags, but 14,070 positive instances and 7,697 negative instances. The instance-level classifier used by NOR-SVM is trained with an order of magnitude more data and makes more robust decisions than the trained bag-level classifier MIL-SVM. Note that we see such a behavior also in Fig. 5(b); with 10 bags, NOR-SVM is more accurate than MIL-SVM (and MILES).

6. Discussion and conclusions

The multiple-instance framework is used in a variety of applications but without stringently checking its underlying assumptions. As has previously been noted (see [6,19] for example), the original MI assumption that "every positive bag contains at least one positive instance" is rather restrictive and there exist numerous scenarios where this assumption is not tenable. Our aim is to contrast multiple-instance learning with the single-instance case, to see when casting the problem in the MI framework is useful, and when it is not necessary.

In their comparison of various MIL classifiers, Ray and Craven [17] also studied the performance of standard instance-level classifiers on MIL problems. They cite results from PAC learning theory [20] for showing when instance-level algorithms can cope with certain type of MIL problems successfully. They raise the question of how good instance-level classifiers would generally perform on such data sets, concluding that they do rather well, even being the best on several of the problems considered. Though they state that the success of the instance-level classifiers may be due to the difference in nature between the instances in positive and negative bags, they do not provide any further investigation into this issue.

In this paper, we devise a synthetic setting where MI problems of different complexities can be defined and we define four tasks to compare instance-level and bag-level classifiers. We believe that these tasks, which are simple to define and easy to visualize, can be considered as canonical MI problems at different levels of intra-bag dependency, corresponding to Weidmann's hierarchy, and they can be used to measure the level of the discriminative power of MI algorithms. Once such MI categories are defined, it may be a more fruitful approach to propose new MI algorithms for each category, rather than trying to come up with a MI algorithm that can handle all the different MI categories.

We see that the noisy-or combination of instance-level decisions can handle MI problems that a pure instance-level classifier cannot learn, and also that a bag-level classifier using a bag-level representation can handle one more level than the noisy-or combiner. There is also a task that cannot be learned by the bag-level classifier using a bag-level representation, which indicates that there is a need for more powerful multiple-instance learners.

Still, as has already been noticed empirically in the literature (see [8] for example) and as we also see on our two bioinformatics applications, a pure instance-level classifier, or an instance-level classifier combined with noisy-or, may sometimes be as accurate or even more accurate than bag-level classifiers. Based on our experimental results we believe that this may be due to a number of reasons:

- Most MI problems actually are not that difficult. In terms of the synthetic tasks we define, we believe that most are at the level of Task 1 or 2. So in most MI problems, a relatively easy approach, using a simple combination, e.g., using noisy-or, of instance-level classifier works quite well. We see this for example in the two bioinformatics applications we use.
- In a data set, there are many more instances than bags. So when the data set is small, there may not be enough data to train a bag-level classifier but an instance-level classifier may learn better. Hence even though the underlying problem is really an MI one (maybe at the level of Task 3 or higher), an instance-level classifier (by itself or combined through noisy-or) may work better, because it can be trained with more data.
- The MI assumption that "every positive bag contains at least one positive instance" does not always hold. In a positive bag, there may be many positive instances, or the nonpositive instances in the positive bags may be quite different from the negative instances in the negative bags. In such a case, the instance-level classifier, either by itself or combined through noisy-or, actually turns out to be quite accurate.

If these conditions hold, there is no need to cast the problem in the MI framework or look for a bag-level representation or classification. We believe that the MI approach is necessary when instances provide only partial information and when no label can be attached to any instance, that is, when one cannot say anything about the positivity or negativity of individual instances. Only in such a case, and only when we have enough bag-level data, one should use a bag-level representation and a bag-level classifier.

Conflict of interest

None declared.

Acknowledgments

This work was done while Ethem Alpaydın was a visiting scholar at Delft University of Technology, supported by a BİDEB fellowship from the Turkish Scientific Technical Research Council (TÜBİTAK). We gratefully acknowledge Dimitrios Palachanis for sharing the bioinformatics data sets he originally created for his master thesis research at Delft University of Technology.

References

[1] J. Foulds, E. Frank, A review of multi-instance learning assumptions, Knowl. Eng. Rev. 25 (2010) 1–25.
[2] J. Amores, Multiple instance classification: review, taxonomy and comparative study, Artif. Intell. 201 (2013) 81–105.

[3] T.G. Dietterich, R. Lathrop, T. Lozano-Perez, Solving the multiple-instance problem with axis-parallel rectangles, Artif. Intell. 89 (1997) 31–71.
[4] O. Maron, T. Lozano-Perez, A framework for multiple-instance learning, in: Advances in Neural Information Processing Systems, The MIT Press, Cambridge, MA, 1998, pp. 570–576.
[5] P. Gehler, O. Chapelle, Deterministic annealing for multiple-instance learning, in: M. Meila, X. Shen (Eds.), Proceedings of the 11th International Conference on Artificial Intelligence and Statistics, 2007, pp. 123–130.
[6] X. Xu, Statistical learning in multiple instance problems (M.Sc. thesis), University of Waikato (2003).
[7] X. Xu, E. Frank, Logistic regression and boosting for labeled bags of instances, in: Proceedings of the Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, Berlin, Germany, 2004, pp. 272–281.
[8] V.C. Raykar, B. Krishnapuram, J. Bi, M. Dundar, R.B. Rao, Bayesian multiple instance learning: Automatic feature selection and inductive transfer, in: Proceedings of the 25th International Conference on Machine Learning, ACM, New York, NY, 2008, pp. 808–815.
[9] Z.H. Zhou, J.M. Xu, On the relation between multi-instance learning and semi-supervised learning, in: Proceedings of the 24th International Conference on Machine Learning, ACM, New York, NY, 2007, pp. 1167–1174.
[10] Y. Li, D.M.J. Tax, R.P.W. Duin, M. Loog, Multiple-instance learning as a classifier combining problem, Pattern Recognit. 46 (2013) 865–874.
[11] Z.H. Zhou, Y.Y. Sun, Y.F. Li, Multi-instance learning by treating instances as non-iid samples, in: Proceedings of the 26th International Conference on Machine Learning, ACM, New York, NY, 2009, pp. 1249–1256.
[12] N. Weidmann, E. Frank, B. Pfahringer, A two-level learning method for generalized multi-instance problems, in: Proceedings of the 14th European Conference on Machine Learning, Springer, Berlin, Germany, 2003, pp. 468–479.
[13] Y. Chen, J. Bi, J.Z. Wang, Miles: multiple-instance learning via embedded instance selection, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 1931–1946.
[14] V. Cheplygina, D.M.J. Tax, M. Loog, Multiple instance learning with bag dissimilarities, Pattern Recognit. 48 (2015) 264–275.
[15] D.M.J. Tax, M. Loog, R.P.W. Duin, V. Cheplygina, W.-J. Lee, Bag dissimilarities for multiple instance learning, in: Similarity-Based Pattern Recognition, Springer, Berlin, Germany, 2011, pp. 222–234.
[16] D. Palachanis, Using the multiple instance learning framework to address differential regulation (M.Sc. thesis), Pattern Recognition and Bioinformatics Group, Delft University of Technology, 2014.
[17] S. Ray, M. Craven, Supervised versus multiple instance learning: An empirical comparison, in: Proceedings of the 22nd International Conference on Machine Learning, ACM, New York, NY, 2005, pp. 697–704.
[18] S. Ray, M. Craven, Learning statistical models for annotating proteins with function information using biomedical text, BMC Bioinform. 6 (Supplement 1) (2005) S18.
[19] V. Cheplygina, D.M.J. Tax, M. Loog, Does one rotten apple spoil the whole barrel? in: Proceedings of the 21st International Conference on Pattern Recognition, IEEE, Piscataway, New Jersey, 2012, pp. 1156–1159.
[20] A. Blum, A. Kalai, A note on learning from multiple-instance examples, Mach. Learn. 30 (1) (1998) 23–29.

Ethem Alpaydın received his Ph.D. degree from Ecole Polytechnique Fédérale de Lausanne, Switzerland, in 1990, and was a postdoc at the International Computer Science Institute, Berkeley in 1991. He is a Professor in the Department of Computer Engineering, Boğaziçi University, Istanbul and a Member of the Science Academy, Istanbul. He was a Visiting Researcher at MIT in 1994, IDIAP, in 1998, and TU Delft, in 2014. He was a Fulbright scholar, in 1997. The third edition of his book Introduction to Machine Learning was published by The MIT Press, in 2014.

Veronika Cheplygina received her M.Sc. degree in Media and Knowledge Engineering from the Delft University of Technology, the Netherlands, in 2010. Her thesis project Random Subspace Method for One-class Classifiers about detecting outliers during automatic parcel sorting was performed in collaboration with Prime Vision (http://www.primevision.com/). She is currently working towards her Ph.D. at the Pattern Recognition Laboratory at the Delft University of Technology. Her research interests include multiple instance learning, dissimilarity representation and combining classifiers.

Marco Loog received an M.Sc. degree in mathematics from Utrecht University and a Ph.D. degree from the Image Sciences Institute, the Netherlands. After this latter, joyful event, he moved to Copenhagen where he acted as an Assistant and, eventually, an Associate Professor, next to which he worked as a Research Scientist at Nordic Bioscience. Following several splendid years in Denmark, Marco moved to Delft University of Technology where he now works as an Assistant Professor in the Pattern Recognition Laboratory. Currently, he is also Honorary Professor in pattern recognition at the University of Copenhagen and Chairman of Technical Committee 1 of the IAPR. Marco's principal research interest is with supervised pattern recognition in all sorts of shapes and sizes.

David M.J. Tax studied Physics at the University of Nijmegen, the Netherlands, in 1996, and received his Masters degree with the thesis Learning of Structure by Many-take-all Neural Networks. After that he received his Ph.D. from the Delft University of Technology, the Netherlands, in the Pattern Recognition group, under the supervision of Dr. Robert P.W. Duin. In 2001 he was promoted with the thesis One-class Classification. After working for two years as a MarieCurie Fellow in the Intelligent Data Analysis group in Berlin, he is currently an Assistant Professor in the Pattern Recognition Laboratory at the Delft University of Technology. His main research interest is in the learning and development of detection algorithms and (one-class) classifiers that optimize alternative performance criteria like ordering criteria using the Area under the ROC curve or a Precision-Recall graph. Furthermore, the problems concerning the representation of data, multiple instance learning, simple and elegant classifiers and the fair evaluation of methods have focus.