

Dissimilarity-based Ensembles for Multiple Instance Learning

Veronika Cheplygina, David M.J. Tax, and Marco Loog

Abstract—In multiple instance learning, objects are sets (bags) of feature vectors (instances) rather than individual feature vectors. In this paper we address the problem of how these bags can best be represented. Two standard approaches are to use (dis)similarities between bags and prototype bags, or between bags and prototype instances. The first approach results in a relatively low-dimensional representation determined by the number of training bags, while the second approach results in a relatively high-dimensional representation, determined by the total number of instances in the training set. However, an advantage of the latter representation is that the informativeness of the prototype instances can be inferred. In this paper a third, intermediate approach is proposed, which links the two approaches and combines their strengths. Our classifier is inspired by a random subspace ensemble, and considers subspaces of the dissimilarity space, defined by subsets of instances, as prototypes. We provide insight into the structure of some popular multiple instance problems and show state-of-the-art performances on these datasets.

Index Terms—Dissimilarity representation, multiple instance learning, combining classifiers, random subspace method

I. INTRODUCTION

Nowadays, many applications face the problem of using weakly labeled data for training a classifier. For example, in image classification, we may only have an overall label for an image (such as a “tiger”), not where the tiger is actually located in the image. Such problems are often formulated as Multiple Instance Learning (MIL) [1] problems. MIL is an extension of supervised learning, and occurs in cases when class labels are associated with sets (*bags*) of feature vectors (*instances*) rather than with individual feature vectors. The bag labels provide weak information about the instance labels. For example, the label “tiger” could apply to only some of the image patches, because patches of sand, sky or other surroundings could be present as well. This is a natural representation for many real-world problems, therefore MIL has been successfully used to address molecule [1] or drug [2] activity prediction, image classification [3], [4], document categorization [5], computer aided diagnosis [6] and many other problems.

We can group methods that learn in this weakly supervised setting in two categories. The first category, which we call instance-based methods, relies on different assumptions [7] about the relationship of the bag labels and the instance labels to build instance classifiers. A bag is classified a bag by first classifying that bag’s instances, and then fusing these outputs into a bag label. For example, the standard assumption is that

a bag is positive if and only if at least one of its instances is positive or inside a concept [1]. The second category, which we call bag-based methods, often use the collective assumption: they assume all instances contribute to the bag label, and that bags with the same label are somehow similar to each other. Therefore, bags can be classified directly using distances [8] or kernels [9], or by converting bags to a single-instance representation and using supervised learners [4], [10], [11]. The bag-based methods have frequently demonstrated the best performances on a wide range of datasets.

One of the ways to represent structured objects, such as bags, in a feature space, is to describe them relative to a set of reference objects or prototypes. This approach is called the dissimilarity representation [12] and is in contrast to traditional pattern recognition because the dimensions of this space are defined in a relative way: the dissimilarity to the j -th prototype can therefore be seen as the j -th feature in the transformed space. A successful approach we studied, uses *training bags* as prototypes [10], [13], while an alternative approach called MILES [4] uses all the *training instances* as prototypes. Both alternatives have demonstrated the best performances on a wide range of MIL problems, and, as we show in this paper, are in fact strongly related. However, both approaches are extremes with respect to the dimensionality and the information content of the resulting dissimilarity space. The bag representation reduces the dimensionality from the number of instances to the number of bags, but risks losing information contained in the individual instances. The instance representation preserves more information, but increases the dimensionality dramatically, possibly including many redundant features.

We propose a third alternative, which combines the advantages of using bags and instances as prototypes. We train classifiers on different subspaces of the instance dissimilarity space, where each subspace is formed by the instances of a particular bag, and combining the decisions of these classifiers in the test phase. This way, the information provided by different dissimilarities is preserved, but the dimensionality of each subspace is lower. Therefore the ensemble has the potential to be more robust than a single classifier, but still has the ability to provide interpretation for which instances are important in classification. Note that the bag, instance and subspace representations are analogous to different ways – averaging, concatenating and ensembles – of combining information from different sources [14], [15], [16]. The informativeness of these sources is central to this paper, and we focus on investigating which prototypes – bags, instances, or subspaces – are more informative in MIL problems.

The authors are with the Pattern Recognition Laboratory, Delft University of Technology, The Netherlands. E-mail: v.cheplygina@tudelft.nl

We first introduced dissimilarity-based subspace ensembles in [17]. We now present an extended analysis of our approach, as well as significantly improved results on many MIL datasets. With respect to [17], this paper has several differences. In Section 2, we explain the preliminary tools used in our method, and relate our method to other dissimilarity-based approaches in MIL. In Section 3, we more formally define the proposed approach and explain the parameter choices that need to be made, that were treated as defaults in [17]. In Section 4, we provide understanding of the relationship between these parameters and the ensemble performance, which also explains our earlier results. We then demonstrate the effectiveness of the proposed approach with competitive results on several benchmark datasets¹. In addition to these results, we provide insight into the structure of some popular MIL problems, and why the dissimilarity space is a successful approach for such problems in general.

II. DISSIMILARITY-BASED MULTIPLE INSTANCE LEARNING

A. Data Representation

In multiple instance learning, an object is represented by a bag $B_i = \{\mathbf{x}_{ik} | k = 1, \dots, n_i\} \subset \mathbb{R}^d$ of n_i feature vectors or instances. The training set $\mathcal{T} = \{(B_i, y_i) | i = 1, \dots, N\}$ consists of positive ($y_i = +1$) and negative ($y_i = -1$) bags, although multi-class extensions are also possible [18]. The standard assumption for MIL is that there are instance labels y_{ik} which relate to the bag labels as follows: a bag is positive if and only if it contains at least one positive, or *concept* instance: $y_i = \max_k y_{ik}$. In this case, it might be worthwhile to search for only these informative instances. Alternative formulations, where a fraction or even all instances are considered informative, have also been proposed [7].

We can represent an object, and therefore also a MIL bag B_i , by its dissimilarities to prototype objects in a representation set \mathcal{R} [12]. Often \mathcal{R} is taken to be a subset of size M of the training set \mathcal{T} of size N ($M \leq N$). If we apply this to MIL, each bag is represented as $\mathbf{d}(B_i, \mathcal{T}) = [d(B_i, B_1), \dots, d(B_i, B_M)]$: a vector of M dissimilarities. Therefore, each bag is represented by a single feature vector \mathbf{d} and the MIL problem can be viewed as a standard supervised learning problem.

The bag dissimilarity $d(B_i, B_j)$ is defined as a function of the pairwise instance dissimilarities $[d(\mathbf{x}_{ik}, \mathbf{x}_{jl})]_{n_i \times n_j}$. There are many alternative definitions (see [10], [19]) but in this work we focus on the average minimum instance distance, which tends to perform well in practice. Suppose that we are only given one prototype B_j . With the proposed bag dissimilarity, the bag representation of B_i using prototype B_j is:

$$d^{bag}(B_i, B_j) = \frac{1}{n_i} \sum_{k=1}^{n_i} \min_l d(\mathbf{x}_{ik}, \mathbf{x}_{jl}). \quad (1)$$

Note that the dissimilarity between bag B_i and B_j is now reduced to a scalar, and $\mathbf{d}(B_i, \mathcal{T})$ becomes an M -dimensional vector.

¹Datasets available online at <http://www.mipproblems.org>

A related method, MILES [4], considers a different definition of prototype, by using the training instances rather than the training bags. The motivation is that, when we assume just a few concept instances per bag, it is better to consider just these informative instances rather than the bag as a whole. MILES is originally a similarity-based approach, where the similarity is defined as $s(B_i, \mathbf{x}) = \max_k \exp(-\frac{d(\mathbf{x}_{ik}, \mathbf{x})}{\sigma^2})$ and σ is the parameter of the radial basis function kernel. However, by leaving out kernel and the need to choose σ , we get a dissimilarity-based counterpart. The instance representation of B_i using the instances of B_j is then defined as:

$$\mathbf{d}^{inst}(B_i, B_j) = [\min_l d(\mathbf{x}_{i1}, \mathbf{x}_{jl}), \min_l d(\mathbf{x}_{i2}, \mathbf{x}_{jl}), \dots, \min_l d(\mathbf{x}_{in_i}, \mathbf{x}_{jl})]. \quad (2)$$

Now the dissimilarity between B_i and B_j is summarized in a n_i -dimensional vector, resulting in a representation $\mathbf{d}(B_i, \mathcal{T})$ that has a dimensionality of $\sum_{k=1}^M n_k$.

From this point onwards, we will discuss the dissimilarity matrices D^{bag} and D^{inst} , which look as follows:

$$D^{bag} = [d^{bag}(B_*, B_*)]_{N \times M} \quad (3)$$

and

$$D^{inst} = [\mathbf{d}^{inst}(B_*, B_*)]_{N \times \sum_{k=1}^M n_k}. \quad (4)$$

D^{bag} and D^{inst} are two extremes with respect to the amount of information that is preserved. In cases where only a few instances per bag are informative, D^{bag} could suffer from averaging out these dissimilarities. D^{inst} would preserve these dissimilarities, but it could be difficult for the classifier to select only these relevant dissimilarities due to the high dimensionality of the representation. As an example, consider an image categorization problem, where an image is a bag, and an image region or patch is an instance. If many images in the training set contain regions that include the sky, the dissimilarities to the sky instances in D^{inst} will provide heavily correlated information about the bags. Therefore, D^{inst} could contain many redundant (but not necessarily uninformative) dissimilarities.

On the other hand, when most instances in a bag are informative, we would expect D^{bag} to perform well. D^{inst} would still have access to all the informative dissimilarities, however, selecting a few relevant dissimilarities, as in [11], where a single instance per bag is selected, might not be the best strategy if most instances are, in fact, relevant for the classification problem. The problem of being unable to specify how many dissimilarities are informative, still holds in this case.

B. Classifier and Informative prototypes

In this work we consider linear classifiers (\mathbf{w}, w_0) such that $f(\mathbf{d}) = \mathbf{w}^T \mathbf{d} + w_0$ and \mathbf{w} is an M -dimensional vector. The

entries of \mathbf{w} correspond to the weights assigned to each of the prototypes, either bags or instances. These weights are found by minimizing an objective function of the form:

$$\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}, \mathcal{T}) + \lambda \Omega(\mathbf{w}), \quad (5)$$

where \mathcal{L} is the loss function evaluated on the training set, such as the logistic (for a logistic classifier) or hinge loss (for a support vector classifier or SVM). Ω is a regularization function of the weight vector and is often the l_2 or the l_1 norm. The l_2 norm typically results in most coefficients of \mathbf{w} being non-zero, while the l_1 norm promotes sparsity, i.e., only some of the coefficients have non-zero values. λ is a parameter that trades off the loss with the constraints on the weight vector, and therefore influences the final values in \mathbf{w} .

A larger coefficient in \mathbf{w} means the dissimilarity was found to be discriminative by the classifier, we therefore can examine the coefficients to discover which prototypes are more informative. However, the low sample size and high dimensionality/redundancy of feature space can make it difficult to find the \mathbf{w} that leads to the best performance on a held-out test set.

There are several alternatives for dealing with the redundancy of the prototypes, which are similar to the filter, wrapper and embedded methods in feature selection, which we describe in this section.

In the filter approach, (subsets of) prototypes are evaluated prior to training a classifier, therefore reducing the dimensionality of the training set. Such methods include clustering the prototypes and then selecting the cluster centers as prototypes. In [20] and [21], *clustering of instances* is performed. In [20], the authors first rely on the standard assumption to cluster and prune the instances of negative bags, creating negative prototypes. The positive prototypes are created by selecting from each bag the instance that is furthest from the negative prototypes. In [21], the prototypes are initialized as the k -means cluster centers of the instances. The prototypes are then updated by letting the prototypes take any value in the instance space \mathbb{R}^d . However, note that in both cases, clustering instances has the risk of excluding informative instances in sparsely populated parts of the feature space, because these would not have any neighbors to form a cluster with. On the other hand, [22] uses *clustering of bags*, by performing k -centers with a bag dissimilarity measure, such as (1), and selecting cluster centers as prototypes. Note that the dissimilarity in (1) is non-metric in which case clustering may lead to unpredictable results [23]. A more general disadvantage of filter approaches is that the informativeness of the filtered prototypes is not available.

In a wrapper approach, the classifier is used to determine which prototypes are more informative. The less informative prototypes are then removed, and the classifier is trained again on a smaller subset of prototypes. This procedure is repeated several times. This approach is used in MILIS [11] and is in fact an application of the popular SVM with recursive feature elimination (SVM-RFE) [24] on the D^{inst} representation. Again, a disadvantage is that in the final model, the informativeness of the removed prototypes is not available.

An example of an embedded approach is a sparse classifier that performs feature selection and classification simultaneously, such as the l_1 -norm SVM [25] or Liknon classifier [26], used in MILES. This way, each prototype is associated with a weight, representing its informativeness. However, such sparse classifiers require cross-validation to set λ , which is ill advised in case the training set is small already. A common consequence of such problems is that a poor set of features might be chosen for the testing phase.

We would like to stress that it is not our purpose to focus on *selection* of prototypes, whether instances or bags. Next to the methods described above, there are many works on prototype selection in dissimilarity spaces [27], [28], that could be applied on either D^{bag} or D^{inst} representations. However, from the perspective of MIL, we are instead interested in the more general question of which prototypes – bags, instances or subspaces – are more informative, and which particular (for example positive or negative) prototypes this might be.

III. PROPOSED APPROACH

A. Random Subspace Ensembles

We can see the bag and instance representations as two alternative ways of combining dissimilarities of different instances: by averaging or by concatenating. If we view these approaches as ways of combining different sources of information, a third alternative, ensembles, springs to mind.

The random subspace method (RSM) [29] is one way to create an ensemble that is particularly geared at small sample size, high-dimensional data. Each classifier is built on a lower-dimensional subspace of the original, high-dimensional feature space. This strategy addresses both aspects of a successful ensemble: accurate and diverse classifiers [30], [31]. Subsampling the feature space reduces the dimensionality for the individual base classifiers, therefore allowing for more accurate classifiers. Resampling of features introduces diversity [30], i.e. decorrelates the classifier decisions, which improves the performance of the overall ensemble.

More formally, the RSM ensemble consists of the following components:

- Number of subspaces L to be sampled
- Numbers of features $\{s_1 \dots s_L\}$ (or just s if $s_i = s_j \forall i, j$) to be selected for each subspace.
- Base classifier f , which is applied to each subspace. We denote the trained classifiers by $\{f_1, \dots, f_L\}$.
- Combining function g , which for a test feature vector \mathbf{d} , combines the outputs into a final classifier $F(\mathbf{d}) = g(f_1(\mathbf{d}), \dots, f_L(\mathbf{d}))$.

RSM is interesting in high-dimensional problems with high feature redundancy [32]. For example, the expression levels of co-regulated (both influenced by another process) genes will provide correlated information about whether a subject has diabetes or not. Other genes may be irrelevant to diabetes, only adding noise. We typically do not have prior knowledge about the number of underlying processes that are responsible to diabetes, i.e., the amount of redundancy is unknown. This increases the number of possible relevant feature sets, and makes selecting only the relevant features

more difficult. RSM decreases this risk, simplifying the feature selection problem for each individual classifier, and by still allowing access to all the (possibly relevant) features, thus letting the classifiers correct each other. Other examples where RSM is a successful approach include functional magnetic resonance imaging (fMRI) data [33], microarray data [34] and hyperspectral data [35].

The different prototypes in MIL may also provide redundant information, but we do not know in advance how many such redundant prototypes there might be. Furthermore, many MIL problems are small sample size problems in the number of bags, so additional classifier evaluations during training are undesirable. Therefore we believe that RSM can be an attractive method to address dissimilarity-based MIL, and to combine the strengths of the dissimilarity space with those of ensemble classifiers.

B. Choice of Subspaces

There are two alternatives for how the subspace classifiers can be defined:

- By choosing each prototype bag as a subspace, i.e. the subspace is formed by the dissimilarities to the instances of a prototype bag. This representation immediately follows from our intuition about bags and instances being analogous to averaging and concatenating different information sources. We denote this representation by D^{BS} , where BS stands for Bag Subspaces. The RSM parameters are straightforward here: $L = M$ and the subspace dimensionalities s_i correspond to the bag sizes n_i .
- By choosing each subspace randomly. We denote this representation by D^{RS} , where RS stands for Random Subspaces. D^{RS} offers more flexibility with regard to the RSM parameters. In [17], we used default parameters $L = M$ and $s = \frac{1}{N} \sum_i n_i$. However, alternative settings are possible as well, and we will demonstrate further on in this paper that other choices (which can be set by rules of thumb rather than cross-validation), can in fact improve the results significantly.

Equations 6 and 7 show the choices in matrix format:

$$D^{BS} = \{ [d^{inst}(B_*, B_1)]_{N \times n_1}, \dots, [d^{bag}(B_*, B_M)]_{N \times n_M} \} \quad (6)$$

and

$$D^{RS} = \{ [d^{inst}(B_*, R_1)]_{N \times s}, \dots, [d^{bag}(B_*, R_M)]_{N \times s} \} \quad (7)$$

where R_i are prototype bags, generated randomly from all available instances in the training set.

Note that these alternatives are both slightly different from RSM because the dissimilarity representation depends on the training set. In traditional RSM, all features are available to the classifier at any split of the training and test data, whereas with D^{BS} and D^{RS} , the features are defined through dissimilarities to the training set, which obviously changes with every training-test split. However, we still expect there to be a relationship between how RSM parameters, and the choices in D^{BS} and D^{RS} , affect ensemble performance.

Representation	Dimensionality	Classifiers
D^{bag}	M	1
D^{inst}	$\sum_i n_i$	1
D^{BS}	$\{n_1, \dots, n_M\}$	M
D^{RS}	any	any

TABLE I: Different ways for constructing dissimilarity representations. D^{bag} consists of dissimilarities to bags in the training set (one for each bag), whereas D^{inst} consists of dissimilarities to instances in the training set. In D^{BS} , a separate classifier is built on each prototype's instance dissimilarities. In D^{RS} , classifiers are built on random selections of all available instances.

We provide a summary of the ensembles, as well as the single classifier dissimilarity representations in Table I.

C. Illustrative example

The basic intuition about the benefit of the proposed ensembles is illustrated by the artificial problem in the top of Fig. 1. This is the classical MIL problem from [36]. This dataset contains bags with 50 two-dimensional instances. The instances from the bags are uniformly distributed in a square, and the positive bags contain at least one feature vector from a concept region that is located in the middle of the square. Only the dissimilarities of the concept instances are informative. Averaging over the dissimilarities as in D^{bag} dilutes these informative features, and indeed, the learning curves in the bottom of Fig. 1 show that D^{bag} performs poorly here. D^{inst} has trouble selecting only the informative dissimilarities because many dissimilarities are uninformative, and because dissimilarities of the informative instances are correlated. The ensemble methods are more robust against these problems and achieve the best performances (Fig. 1, bottom).

IV. EXPERIMENTS

A. Data and Setup

We provide a list of datasets we used in Table II. The Musk problems [1] are traditional benchmark MIL problems about molecule activity, Mutagenesis [37] is a drug activity prediction problem. Fox, Tiger and Elephant [3] are benchmark image datasets. African and Beach are also image datasets originating from a multi-class scene recognition problem [4], but here formulated as one-against-all problems. The dataset alt.atheism originates from the Newsgroups data [5], and are concerned with text categorization. Brown Creeper and Winter Wren are both bird song [38] datasets, where the goal is to classify whether a particular bird species can be heard in an audio recording.

We preprocess the data by scaling each feature to zero mean and unit variance. The scaled data is used to compute the dissimilarity representations. The instance dissimilarity function is defined as the squared Euclidean distance: $d(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)$.

For the base classifier f , we consider linear classifiers as described in Section 2. We used several linear classifiers: logistic, 1-norm SVM and a linear SVM, where the primal formulation is optimized [39]. The trade-off parameter λ is set

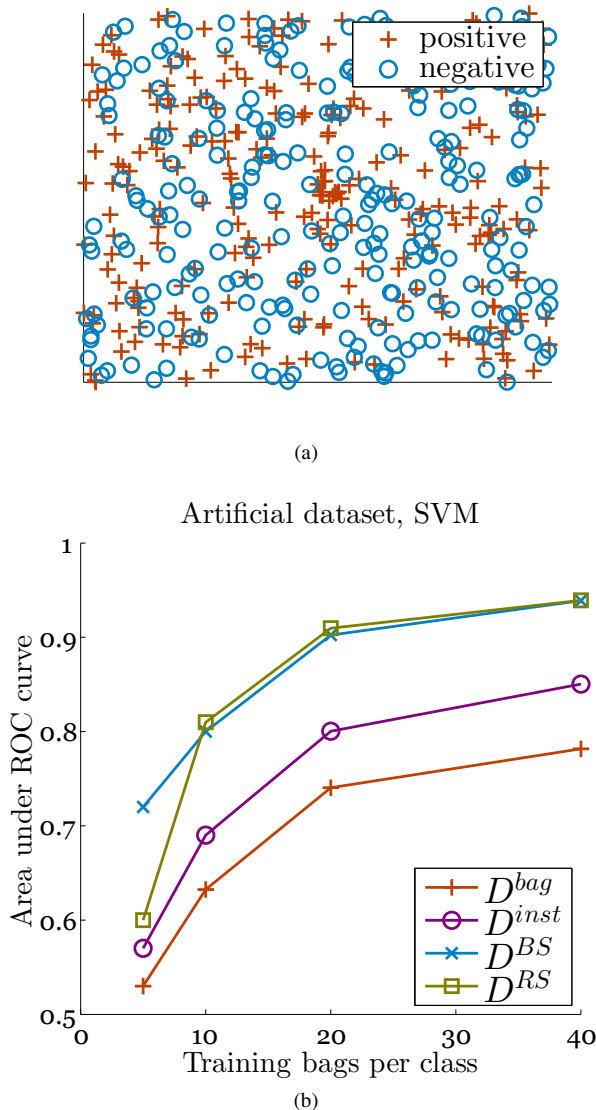


Fig. 1: Top: Artificial 2D MIL problem with informative instances in the center. Bottom: Learning curves for dissimilarity-based classifiers on this dataset. The amount of uninformative, and redundant instances deteriorates performances of D^{bag} and D^{inst} , but the ensemble methods D^{BS} and D^{RS} are more robust against these problems.

to 1 by default. The common characteristic of these classifiers is that we can inspect the weight vector \mathbf{w} to determine which dissimilarities are deemed to be more important by the classifier. Although the individual performances of the classifiers differ, we observe similar trends (such as relative performances of two different ensembles) for these choices. We therefore only show results for the linear SVM.

For the combining function g , we average the posterior probabilities, which are obtained after normalizing the outputs of each classifier to the $[0, 1]$ range. We also considered majority voting, and using the product and the maximum of the posterior probabilities as combining schemes. With the product and maximum rules, the performances were lower, especially deteriorating towards larger ensemble sizes (thus

being more sensitive to outlier, inaccurate base classifiers). With majority voting, the results were slightly lower than with averaging, but the standard deviations across different folds were larger. Therefore we chose averaging as a more robust combining strategy. Furthermore, averaging posterior probabilities performs well in practice in other problems as well [40], [41]. It could be argued that the fixed g can be replaced by a more flexible function, such as a linear classifier, trained on the outputs of the base classifiers. However, as discussed in [42], this would ideally require splitting the training data into two parts: one for training the base classifiers, and one for training the combiner. This is undesirable because we are already faced with a low object-to-feature (i.e., bag-to-instance) ratio. Indeed, our earlier experiments [17] with a nearest mean classifier as a combiner did not lead to improved results.

The metric used for comparisons is area under the receiver-operating characteristic (AUC) [43]. This measure has been shown to be more discriminative than accuracy in classifier comparisons [44], and more suitable for MIL problems [45].

Dataset	+bags	-bags	total	average
Musk 1	47	45	476	5
Musk 2	39	63	6598	65
Fox	100	100	1302	7
Tiger	100	100	1220	6
Elephant	100	100	1391	7
Mutagenesis 1	125	63	10486	56
African	100	1900	7947	8
Beach	100	1900	7947	8
Alt.atheism	50	50	5443	54
Brown Creeper	197	351	10232	19
Winter Wren	109	439	10232	19

TABLE II: MIL datasets, number of bags, instances, the average number of instances per bag. The datasets are available online at <http://www.miprobblems.org>

B. Subspace Experiments

We start by comparing the two alternatives of creating the subspaces, D^{BS} and D^{RS} . For simplicity, we base the parameters of D^{RS} on those for D^{BS} , as in [17]: M subspaces, each subspace with dimensionality $\frac{1}{N} \sum_i n_i$. We use a linear SVM as the classifier (C parameter is set to 1 by default) and perform 10-fold cross-validation. Fig. 2 shows the distributions of the individual classifier performances and the ensemble performance for both representations.

The results show that overall,

- 1) the random subspaces are more informative than bag subspaces, and
- 2) the random subspace ensemble is better at improving upon the base classifiers.

Why do the random subspaces perform better than the bag subspaces? One difference might be the subspace size: the bag subspaces have variable dimensionalities, whereas the random subspaces are equally large. For D^{BS} , we plot the bag size against the performance of that subspace. A few of the results are shown in Fig. 3. In all datasets except alt.atheism, we find medium to high correlations between these quantities. Therefore, as prototypes, small bags are not very informative.

This might seem counterintuitive in a MIL problem, because small bags are less ambiguous with respect to the instance labels under the standard assumption. The fact that large, ambiguous bags are better at explaining the class differences suggests that most instances are informative as prototypes.

The informativeness of most instances as prototypes is supported by the relationship between the bag label and the subspace performance in the plots. Although for a fixed bag size, positive bag subspaces perform better on average, negative bags can also be very good prototypes. This is also true for random bags, for which we do not have labels, but for which we can examine the percentage of instances, which were sampled from positive bags. We found no correlations between the positiveness of the random subspaces and their performance. This provides opportunities for using unlabeled data in a semi-supervised way: unlabeled bags can be used to extend the dissimilarity representation to improve performance, similar to the approach in [46].

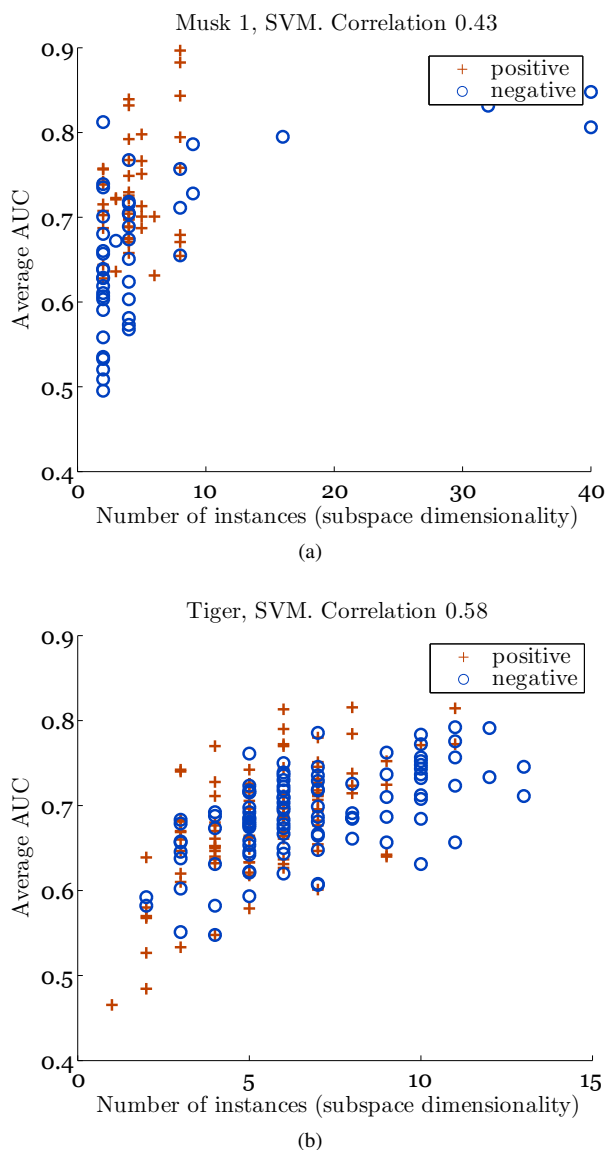


Fig. 3: Relationship of bag size, bag label, and AUC performance of the dissimilarity subspace formed by that bag

The results with respect to the bag size suggest that it is advantageous to combine classifiers built on higher-dimensional subspaces. We therefore investigate the effects of subspace dimensionality in D^{RS} , where this parameter can be varied. We vary the subspace size for each classifier from 5 to 100 features, which for most datasets, would be larger than the default dimensionalities used previously, as shown in Table II. We generate 100 subspaces of each dimensionality, and train a linear SVM on each subspace. The classifiers are then evaluated individually. Some examples of performance distributions at different subspace dimensionalities are shown in Fig. 4. For most datasets (except Newsgroups, as will be explained later), larger subspaces lead to more accurate classifiers. However, increasing the dimensionality too much, eventually using all the features, decreases the performance. This can be more clearly seen in the results of Musk2, where the total number of instances (and thus dimensionality) larger than in Musk1. The results of other datasets show similar trends depending on the bag/instance ratio, for example, the results of Mutagenesis are quite similar to Musk2.

Why is D^{RS} better than D^{BS} at improving upon the individual classifiers? A possible explanation is that the classifiers created by D^{RS} are more diverse than the classifiers of D^{BS} . For each set of classifiers, we examine their $L \times L$ disagreement matrix C , where each entry $C_{i,j}$ corresponds to the number of test bags for which the i -th and j -th classifier provide different labels. $C_{i,j}$ can be seen as the distance of the two classifiers. We perform multi-dimensional scaling with each of these distance matrices and map the classifiers into a 2-dimensional classifier projection space [47].

The classifier projection spaces for two datasets are shown in Fig. 5. These results are surprising, because the classifiers in D^{BS} actually cover a larger part of the space, and are therefore more diverse. This diversity alone, however, is not able to improve the overall performance of the ensemble. A possible explanation is that here we are dealing with bad diversity [48]. For example, a classifier that is 100% wrong is very diverse with respect to a classifier that is 100% correct, but not beneficial when added to an ensemble. We showed in Fig. 3 that in D^{BS} , small bags often result in inaccurate classifiers, which are indeed responsible for higher diversity, but worsen the ensemble performance.

In the experiments, the Newsgroups data shows very different behavior from the other datasets. Most of the subspaces (both bag and random) have nearly random performance, and the performances are not correlated with the bag label or subspace dimensionality. A possible explanation is that in this dataset, many of the dissimilarities only contain noise, and the informativeness is distributed only across a few dissimilarities. RSM is particularly suitable for problems where the informativeness is spread out over many (redundant) features [32], which could explain the worse than random performance. Indeed, examining the individual informativeness (as measured by the nearest neighbor error) of each individual dissimilarity, it turns out that more than half of the dissimilarities in alt.atheism have worse than random performance, as opposed to only around 10% of dissimilarities for the other datasets.

We have noticed previously [13] that positive bags in the

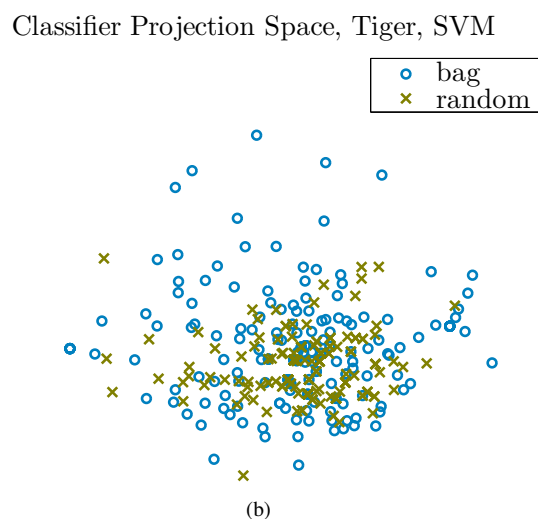
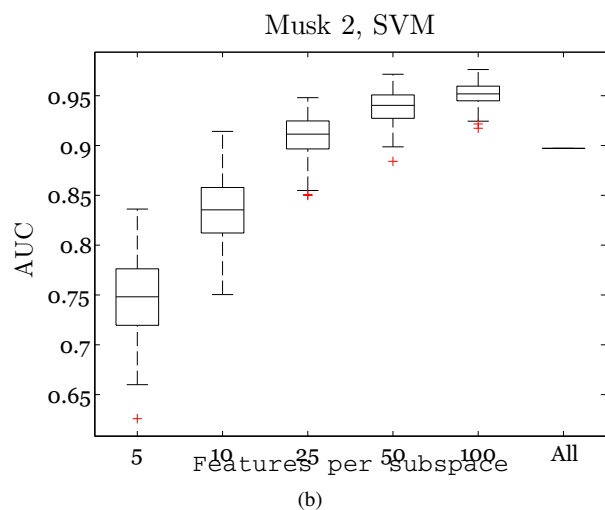
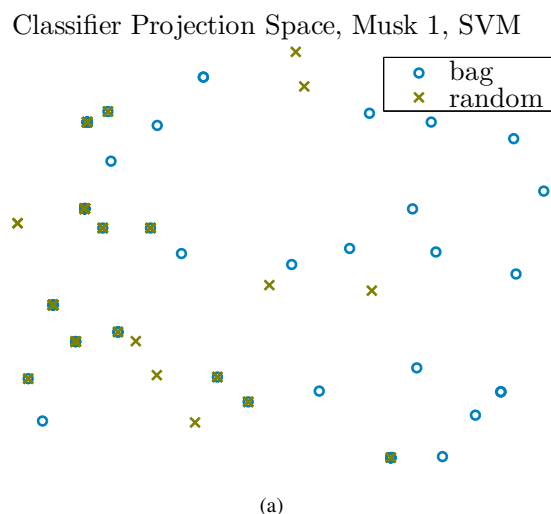
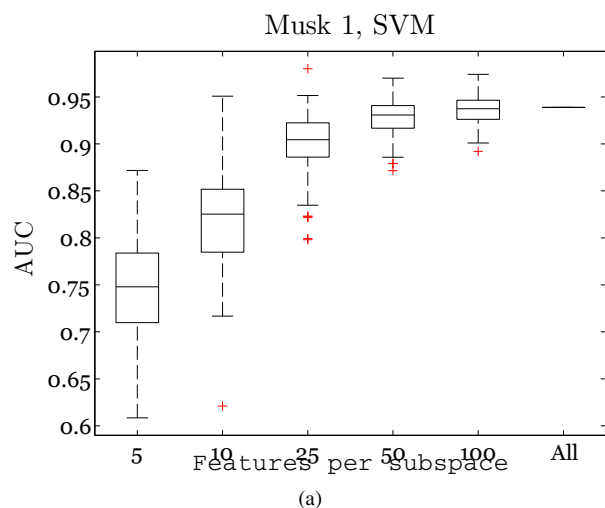


Fig. 4: Distributions of AUC performances of individual subspace classifiers, for different dimensionalities of the subspaces. The degradation of performance using all features is larger in Musk2 because instance / bag (and thus feature / object) ratio is larger than in Musk1. Similar trends can be observed in other datasets.

Fig. 5: Classifier projection spaces. The plots show the relative disagreement of trained subspace classifiers on a test set. The higher the disagreement of two classifiers on the labels of the test set, the larger their distance in the plot.

Newsgroups data consist of dense cluster of instances and a few outliers, while negative bags of the dense cluster of instances, and a few outliers. This distribution is caused by the bag of words representation of the data – while the outlier instances are in fact all positive for the alt.atheism topic, they do not consist of the same words, and are far from each other in the feature space. This situation is shown in Fig.6. The presence or absence of such outliers is very informative for the bag class. The definition of dissimilarity in (2), however, does not succeed in extracting this information: for any training bag, the instance closest to the prototype outlier instance, is in the dense cluster. In this case the minimum function in the dissimilarity is not suitable for the problem at hand, and much better performances can be obtained by, for instance, using bags and prototypes, and considering the asymmetry of D^{bag} [13].

C. Ensemble Experiments

With several interesting results for the individual performances of the subspace classifiers, we now investigate how the ensemble behaves when both the parameters of interest are varied. The ensembles are built in such a way that classifiers are only added (not replaced) as the ensemble size increases, i.e. an ensemble with 100 classifiers has the same classifiers as the ensemble with 50 classifiers, and 50 additional ones.

The results are shown in Fig. 7. Here different plots show the results for different subspace dimensionalities (5, 10, 25, 50 or 100 features), and the x-axis indicates the number of classifiers used in the ensemble. The black line shows the baseline performance of D^{inst} . The performance metric is again the area under the ROC curve (AUC), and the results are averaged over 10-fold cross-validation.

Ensembles created with higher-dimensional subspaces tend to perform better. An unsurprising result is that the fixed dimensionality values in this experiment are not suitable for all

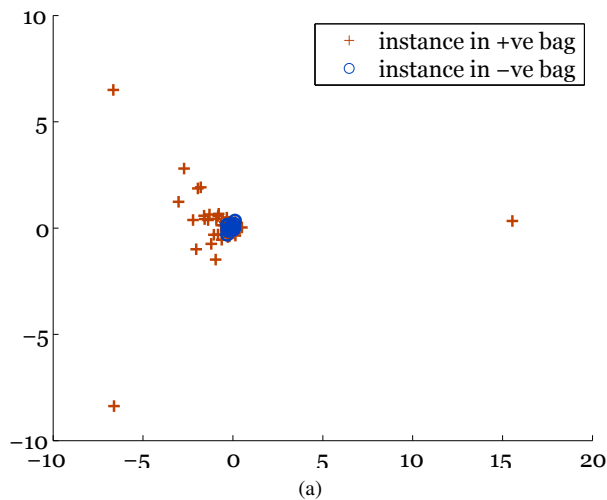


Fig. 6: Multi-dimensional scaling of the instances in the alt.atheism dataset

datasets. For example, in Musk none of the ensembles outperform the single classifier. Clearly, the subspace dimensionality should depend on the dimensionality (and possibly redundancy of the dissimilarities) of the original problem.

Another interesting observation in Fig. 7 is that it is often sufficient to build the ensemble from a few base classifiers, and adding further classifiers probably would not improve performance significantly. This is in line with our earlier results for ensembles of one-class classifiers [41], even though both the data and the classifiers in question are very different. The recommendation in [33] is that when there is no prior knowledge about how redundant the features are, L should be chosen to be relatively small, while s should be relatively large. Based on these observations, we settle on the following choices for D^{RS} : $L = 100$ and $s = \frac{1}{5} \sum_i n_i$. We emphasize that these choices are good rules of thumb and do not have to be set to these exact values, which is supported by our results from the previous section: the performance is quite robust to changes in L and s provided s is large enough.

The performances of the proposed ensemble against those of the single classifier representations D^{bag} and D^{inst} are shown in Table III. Contrary to our earlier results in [17], D^{RS} is now a clear competitor for the single classifiers, and has especially surprising performances on the Musk 1, Fox and Mutagenesis datasets. The advantages of D^{RS} over the high-dimensional D^{inst} are more visible, however, there are no significant differences with D^{bag} . This suggests that many of the dissimilarities are, in fact, informative, and averaging the dissimilarities over each bag preserves sufficient information. Despite the similar performances of D^{bag} and D^{RS} , D^{RS} has an additional advantage: it is possible to recover the importance of different instances, as we will show in Section IV-E.

D. Comparison with other MIL Classifiers

We compare our method to other popular MIL classifiers, which cover a range of instance-based and bag-based methods, and are often being compared to in recent papers. EM-

Dataset	Representation		
	D^{bag}	D^{inst}	D^{RS}
Musk1	93.7 (3.5)	93.9 (3.6)	95.4 (2.4)
Musk2	95.7 (1.4)	89.7 (4.7)	93.2 (3.2)
Fox	67.9 (2.5)	66.0 (3.9)	70.2 (1.8)
Tiger	86.8 (5.2)	83.7 (3.9)	87.8 (4.2)
Elephant	90.9 (2.5)	87.6 (3.3)	92.3 (2.7)
Mutagenesis	84.3 (2.9)	83.2 (3.2)	87.4 (3.5)
Brown Creeper	94.7 (1.0)	93.9 (0.8)	94.2 (0.8)
Winter Wren	99.5 (0.2)	98.4 (0.5)	99.0 (0.3)
African	92.5 (1.1)	91.6 (1.4)	92.8 (1.2)
Beach	88.3 (1.2)	85.6 (1.2)	87.9 (1.2)
alt.atheism	62.4 (8.3)	46.4 (5.9)	46.4 (5.2)

TABLE III: AUC ($\times 100$) mean and standard error of 10-fold cross-validation of the single dissimilarity-based representations, and the proposed ensemble representation. **Bold** = best, *italics* = second best result per dataset

DD [49], mi-SVM [50] and MILBoost [51] are instance-based methods. EM-DD is an expectation-maximization algorithm which uses diverse density (DD), which, for a given point t in the feature space, measures the ratio between the number of positive bags which have instances near t , and the distance of the negative instances to t . The expectation step selects the most positive instance from each bag according to t , the maximization step then finds a new concept t' by maximizing DD on the selected, most positive instances. mi-SVM is an extension of support vector machines which attempts to find hidden labels of the instances under constraints posed by the bag labels. Likewise, MILBoost is an extension of boosting and MIForest is an extension of random forests, where the standard assumption is used to reweigh or relabel the instances in each iteration of the algorithm.

MILES [4] and the Minimax kernel are bag-based methods which convert bags to a single-instance representation. MILES is similar to the 1-norm SVM applied to the D^{inst} , except that in MILES, a Gaussian kernel is used for defining similarities, and an appropriate σ parameter is necessary. The Minimax kernel [9] is obtained by representing each bag by the minimum and maximum feature values of its instances, this representation can then be used with a supervised classifier. All classifiers are implemented in PRTools [52] and the MIL toolbox [53] and default parameters are used unless stated otherwise.

Next to the standard MIL classifiers, we use D^{RS} with the guidelines from the previous section: the dimensionality of each subspace is 1/5th of the total dimensionality, and 100 classifiers are used in the ensemble. The base classifier is the linear SVM. For both MI-SVM and MILES, the radial basis kernel with width 10 was used. The results are shown in Tables IV and V. Some results could not be reported; in particular, for EM-DD when one cross-validation fold lasts longer than five days, and MILBoost when features with the same value for all instances are present, as in alt.atheism. Needless to say, there is no method that performs the best at all times. Some of the default parameter choices may be unsuitable for a particular dataset, or the assumptions that the method is based on do not hold. However, overall the method we present is always able to provide competitive performance.

Dataset	Classifier					
	EM-DD	MI-SVM	MILBoost	MILES	minimax+SVM	D^{RS} +SVM
Musk1	85.0 (5.1)	91.5 (3.7)	74.8 (6.7)	93.2 (2.9)	87.8 (5.0)	95.4 (2.4)
Musk2	88.1 (2.7)	93.9 (2.8)	76.4 (3.5)	97.1 (1.6)	91.3 (1.8)	93.2 (3.2)
Fox	67.6 (3.2)	68.7 (2.6)	61.3 (3.2)	66.8 (3.5)	55.8 (2.9)	70.2 (1.8)
Tiger	-	87.2 (3.5)	87.0 (3.0)	84.6 (4.5)	76.0 (4.1)	87.8 (4.2)
Elephant	88.5 (2.1)	90.7 (2.3)	88.8 (2.2)	88.4 (2.5)	88.4 (2.1)	92.3 (2.7)
Mutagenesis	67.4 (5.3)	60.3 (4.5)	88.1 (3.1)	72.1 (4.3)	63.7 (4.4)	87.4 (3.5)
Brown Creeper	94.5 (0.9)	92.8 (1.2)	94.9 (0.9)	96.1 (0.6)	94.2 (0.9)	94.2 (0.8)
Winter Wren	98.3 (0.5)	99.2 (0.4)	93.8 (5.6)	99.1 (0.5)	98.2 (0.3)	99.0 (0.3)
African	91.2 (1.8)	88.6 (1.7)	89.4 (1.7)	48.7 (2.3)	87.6 (1.4)	92.8 (1.2)
Beach	84.6 (2.0)	78.2 (2.5)	85.2 (2.9)	72.8 (5.0)	83.0 (2.3)	87.9 (1.2)
Alt.atheism	52.0 (8.0)	38.8 (5.2)	-	50.0 (5.5)	80.0 (3.6)	46.4 (5.2)

TABLE IV: AUC $\times 100$, mean and standard error of 10-fold cross-validation of different MIL classifiers. **Bold** = best, *italics* = second best per dataset.

Dataset	Classifier					
	EM-DD	MI-SVM	MILBoost	MILES	minimax+SVM	D^{RS} +SVM
Musk1	85.1 (4.1)	83.0 (4.6)	69.8 (5.6)	82.8 (4.7)	86.3 (3.8)	89.3 (3.4)
Musk2	81.5 (2.9)	76.3 (5.4)	66.0 (3.7)	86.3 (3.4)	82.3 (2.5)	85.5 (4.7)
Fox	62.0 (2.7)	63.5 (2.2)	63.0 (2.6)	62.5 (4.2)	58.0 (2.5)	64.5 (2.2)
Tiger	-	73.0 (2.9)	78.5 (2.8)	81.0 (3.4)	72.5 (3.9)	81.0 (4.6)
Elephant	82.5 (1.5)	76.0 (2.4)	79.5 (2.8)	79.0 (2.3)	82.5 (2.5)	84.5 (2.8)
Mutagenesis	61.1 (5.1)	66.5 (0.4)	75.7 (2.7)	71.1 (3.4)	68.9 (3.0)	83.2 (3.4)
Brown Creeper	85.6 (1.5)	50.5 (1.5)	88.7 (1.2)	89.8 (1.4)	87.3 (1.6)	88.3 (1.0)
Winter Wren	94.5 (1.2)	81.0 (1.5)	89.0 (7.8)	96.7 (0.8)	93.8 (0.8)	96.0 (0.8)
African	82.7 (0.9)	94.7 (0.1)	95.2 (0.2)	95.0 (0.0)	95.0 (0.0)	96.0 (0.5)
Beach	73.4 (1.1)	95.0 (0.1)	95.2 (0.2)	95.0 (0.0)	94.9 (0.1)	94.8 (0.4)
Alt.atheism	49.0 (5.7)	48.0 (2.0)	-	50.0 (4.5)	76.0 (4.0)	44.0 (4.5)

TABLE V: Accuracy $\times 100$, mean and standard error of 10-fold cross-validation of different MIL classifiers. **Bold** = best, *italics* = second best per dataset.

E. Instance weights

An advantage of linear classifiers is the interpretability of the result – from the weights w of the dissimilarities, we can derive which dissimilarities, and therefore which instances are more important (i.e., have a larger weight) to the classifier. This property can also be used in ensembles of linear classifiers, with a procedure described in [54]. For each dissimilarity, we calculate the average absolute value of its weight over all L subspaces in which the dissimilarity was selected. We then sort the dissimilarities by this average weight, and view the position of each dissimilarity in this list as its rank. The distributions of the dissimilarities with ranks 1 to 100 are shown in Fig.8 shows the distributions of top 100 dissimilarities. These most informative dissimilarities originate from both positive and negative bags, supporting the idea that not only concept, positive instances are important for these MIL problems.

V. DISCUSSION

We proposed a dissimilarity-based ensemble as a novel classification method for MIL problems. When bags are represented by their dissimilarities to instances from the training set, such instances can provide redundant information about the problem. A random subspace inspired ensemble, where classifiers are trained on different subspaces of the dissimilarity space, is a way of dealing with this redundancy. We show that our method achieves competitive performances with other MIL algorithms, and has intuitive parameters that do not need to be set by cross-validation to achieve these good results.

We investigated two choices for generating the subspaces: by using each training bag as a subspace, or by using a random selection of instances (with replacement) as a subspace. The random method achieved better results, especially when the dimensionality of the subspaces was increased. In fact, we found that the subspace dimensionality is the most important factor affecting the performance of the ensemble. Overall, larger subspaces lead to more accurate classifiers. Combining subspaces of similar size and therefore accuracy as in D^{RS} therefore leads to better results than combining subspaces with high variance in size and accuracy as in D^{BS} , even though the subspaces in D^{BS} are more diverse. On the other hand, the number of subspaces does not play a very important role and just a few classifiers are sufficient for good performance. These conclusions are in line with conclusions from other applications of the random subspace method, where the amount of redundancy of the features is unknown.

In general, the informativeness of a prototype is more related to the dimensionality of the subspace, then to the label of instances forming that subspaces. Negative bags, and unlabeled random sets of instances were often good prototypes, suggesting that most instances, and not only a few concept ones, are informative for these MIL problems. These results are more in line with the collective assumption for MIL, where all instances are considered to contribute to the bag label, rather than with the standard assumption, where only a few positive instances are considered important.

Based on the encouraging results concerning the effectiveness of random subspaces as prototypes, we also considered

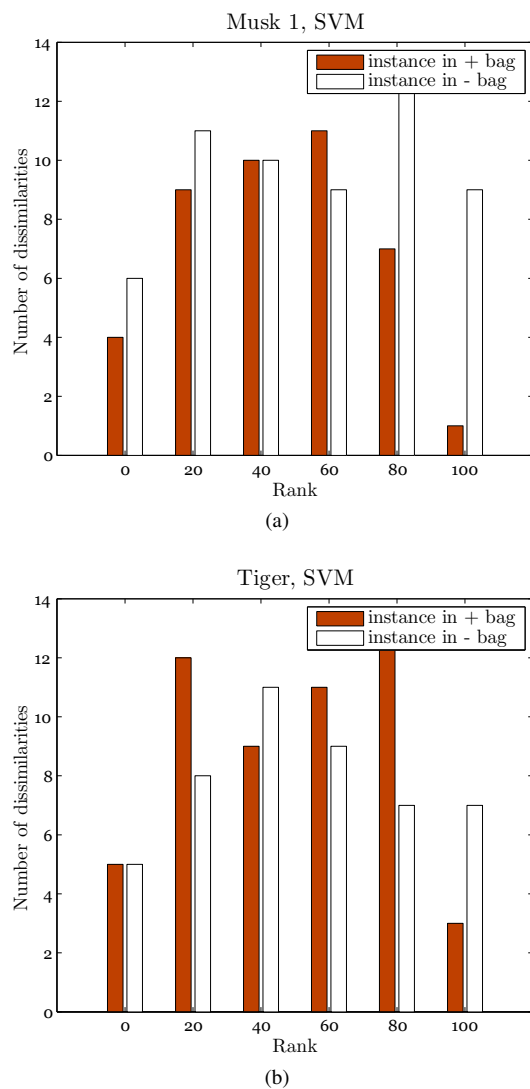


Fig. 8: Top 100 ranked dissimilarities, where the rank is determined by average weight of dissimilarities across $L = 100$ subspace classifiers

randomly sampling the instance space (rather than randomly selecting existing instances) to generate artificial prototype bags that are not in the training set. Although the results with artificial prototypes were slightly worse than with real prototypes, this does seem to provide opportunities for using unlabeled, or artificial bags in a semi-supervised way.

We would like to conclude by emphasizing that a dissimilarity-based representation combined with a linear classifier (or an ensemble thereof) is a powerful way of classifying MIL bags. A question that still remains is the use of structured norms in such linear classifiers, which would enable selection of groups of dissimilarities, therefore revealing more about the relationships of the instances.

REFERENCES

[1] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997.

[2] G. Fu, X. Nan, H. Liu, R. Patel, P. Daga, Y. Chen, D. Wilkins, and R. Doerksen, "Implementation of multiple-instance learning in drug activity prediction," *BMC Bioinformatics*, vol. 13, no. Suppl 15, p. S3, 2012.

[3] S. Andrews, T. Hofmann, and I. Tsochantaris, "Multiple instance learning with generalized support vector machines," in *Proc. National Conference on Artificial Intelligence*, Edmonton, Alberta, Canada, Jul. 2002, pp. 943–944.

[4] Y. Chen, J. Bi, and J. Z. Wang, "MILES: multiple-instance learning via embedded instance selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1931–1947, Dec. 2006.

[5] Z. H. Zhou, Y. Y. Sun, and Y. F. Li, "Multi-instance learning by treating instances as non-IID samples," in *Proc. International Conference on Machine Learning*, Montreal, Canada, Jun. 2009, pp. 1249–1256.

[6] G. Fung, M. Dundar, B. Krishnapuram, and R. B. Rao, "Multiple instance learning for computer aided diagnosis," in *Proc. Advances in Neural Information Processing Systems*, vol. 19, Vancouver, BC, Canada, Dec. 2007, pp. 425–432.

[7] J. Foulds and E. Frank, "A review of multi-instance learning assumptions," *Knowledge Engineering Review*, vol. 25, no. 1, pp. 1–25, 2010.

[8] J. Wang, "Solving the multiple-instance problem: A lazy learning approach," in *Proc. International Conference on Machine Learning*, Stanford, CA, USA, Jun. 2000, pp. 1119–1126.

[9] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola, "Multi-instance kernels," in *Proc. International Conference on Machine Learning*, Sydney, Australia, Jul. 2002, pp. 179–186.

[10] D. M. J. Tax, M. Loog, R. P. W. Duin, V. Cheplygina, and W.-J. Lee, "Bag dissimilarities for multiple instance learning," in *Proc. Similarity-Based Pattern Recognition*, Venice, Italy, Sep. 2011, pp. 222–234.

[11] Z. Fu, A. Robles-Kelly, and J. Zhou, "MILIS: Multiple instance learning with instance selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 958–977, May 2010.

[12] E. Pekalska and R. P. W. Duin, *The dissimilarity representation for pattern recognition: foundations and applications*. Singapore: World Scientific Pub Co Inc, 2005.

[13] V. Cheplygina, D. M. J. Tax, and M. Loog, "Class-dependent dissimilarity measures for multiple instance learning," in *Proc. Structural, Syntactic, and Statistical Pattern Recognition*, Hiroshima, Japan, Nov. 2012, pp. 602–610.

[14] E. Pekalska and R. P. W. Duin, "On combining dissimilarity representations," in *Proc. Multiple Classifier Systems*, Cambridge, United Kingdom, Jul. 2001, pp. 359–368.

[15] J. Kittler, "Combining classifiers: A theoretical framework," *Pattern Analysis & Applications*, vol. 1, no. 1, pp. 18–27, 1998.

[16] R. P. W. Duin and D. M. J. Tax, "Experiments with classifier combining rules," in *Proc. Multiple Classifier Systems*. Springer, 2000, pp. 16–29.

[17] V. Cheplygina, D. M. J. Tax, and M. Loog, "Combining instance information to classify bags," in *Proc. Multiple Classifier Systems*, Nanjing, China, May 2013, pp. 13–24.

[18] Z.-H. Zhou and M.-L. Zhang, "Multi-instance multi-label learning with application to scene classification," in *Proc. Advances in Neural Information Processing Systems*, Vancouver, BC, Canada, Dec. 2006, pp. 1609–1616.

[19] V. Cheplygina, D. M. J. Tax, and M. Loog, "Does one rotten apple spoil the whole barrel?" in *Proc. International Conference on Pattern Recognition*, Tsukuba, Japan, Nov. 2012, pp. 1156–1159.

[20] A. Erdem and E. Erdem, "Multiple-instance learning with instance selection via dominant sets," in *Proc. Similarity-Based Pattern Recognition*, Venice, Italy, Sep. 2011, pp. 177–191.

[21] E. Akbas, B. Ghanem, and N. Ahuja, "MIS-Boost: Multiple instance selection boosting," *arXiv preprint arXiv:1109.2388*, 2011.

[22] M.-L. Zhang and Z.-H. Zhou, "Multi-instance clustering with applications to multi-instance prediction," *Applied Intelligence*, vol. 31, no. 1, pp. 47–68, 2009.

[23] D. W. Jacobs, D. Weinshall, and Y. Gdalyahu, "Classification with nonmetric distances: Image retrieval and class representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 583–600, Jun. 2000.

[24] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 389–422, 2002.

[25] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," in *Proc. Advances in Neural Information Processing Systems*, vol. 16, no. 1, Vancouver, BC, Canada, Dec. 2004, pp. 49–56.

[26] C. Bhattacharyya, L. Grate, A. Rizki, D. Radisky, F. Molina, M. I. Jordan, M. J. Bissell, and I. S. Mian, "Simultaneous classification and relevant feature identification in high-dimensional spaces: application to

molecular profiling data," *Signal Processing*, vol. 83, no. 4, pp. 729–743, 2003.

[27] E. Pekalska, R. P. W. Duin, and P. Paclík, "Prototype selection for dissimilarity-based classifiers," *Pattern Recognition*, vol. 39, no. 2, pp. 189–208, 2006.

[28] Y. P. Calana, E. G. Reyes, M. O. Alzate, and R. P. W. Duin, "Prototype selection for dissimilarity representation by a genetic algorithm," in *Proc. International Conference on Pattern Recognition*, Istanbul, Turkey, Aug. 2010, pp. 177–180.

[29] T. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 832–844, Aug. 1998.

[30] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181–207, 2003.

[31] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: a survey and categorisation," *Information Fusion*, vol. 6, no. 1, pp. 5–20, 2005.

[32] M. Skurichina and R. P. W. Duin, "Bagging and the random subspace method for redundant feature spaces," in *Proc. Multiple Classifier Systems*, Cambridge, United Kingdom, Jul. 2001, pp. 1–10.

[33] L. I. Kuncheva, J. J. Rodríguez, C. O. Plumptre, D. E. J. Linden, and S. J. Johnston, "Random subspace ensembles for fMRI classification," *IEEE Transactions on Medical Imaging*, vol. 29, pp. 531–542, Feb. 2010.

[34] A. Bertoni, R. Folgieri, and G. Valentini, "Bio-molecular cancer prediction with random subspace ensembles of support vector machines," *Neurocomputing*, vol. 63, pp. 535–539, 2005.

[35] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, pp. 492–501, Mar. 2005.

[36] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *Proc. Advances in Neural Information Processing Systems*, Denver, CO, USA, Nov. 1998, pp. 570–576.

[37] A. Srinivasan, S. Muggleton, and R. D. King, "Comparing the use of background knowledge by inductive logic programming systems," in *Proc. International Workshop on Inductive Logic Programming*, Leuven, Belgium, Sep. 1995, pp. 199–230.

[38] F. Briggs, B. Lakshminarayanan, L. Neal, X. Fern, R. Raich, S. Hadley, A. Hadley, and M. Betts, "Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach," *Journal of the Acoustical Society of America*, vol. 131, p. 4640, 2012.

[39] O. Chapelle, "Training a support vector machine in the primal," *Neural Computation*, vol. 19, no. 5, pp. 1155–1178, 2007.

[40] D. M. J. Tax, M. Van Breukelen, R. P. W. Duin, and J. Kittler, "Combining multiple classifiers by averaging or by multiplying?" *Pattern Recognition*, vol. 33, no. 9, pp. 1475–1485, 2000.

[41] V. Cheplygina and D. M. J. Tax, "Pruned random subspace method for one-class classifiers," in *Proc. Multiple Classifier Systems*, Naples, Italy, Jun. 2011, pp. 96–105.

[42] R. P. W. Duin, "The combining classifier: to train or not to train?" in *Proc. International Conference on Pattern Recognition*, vol. 2, Quebec, Canada, Aug. 2002, pp. 765–770.

[43] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[44] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 299–310, Mar. 2005.

[45] D. M. J. Tax and R. P. W. Duin, "Learning curves for the analysis of multiple instance classifiers," in *Proc. Structural, Syntactic, and Statistical Pattern Recognition*, Tampa, FL, USA, Dec. 2008, pp. 724–733.

[46] C. V. Dinh, R. P. W. Duin, and M. Loog, "A study on semi-supervised dissimilarity representation," in *Proc. International Conference on Pattern Recognition*, Tsukuba, Japan, Nov. 2012, pp. 2861–2864.

[47] E. Pekalska, R. P. W. Duin, and M. Skurichina, "A discussion on the classifier projection space for classifier combining," in *Proc. Multiple Classifier Systems*, Cagliari, Italy, Jun. 2002, pp. 137–148.

[48] G. Brown and L. I. Kuncheva, "Good and bad diversity in majority vote ensembles," in *Proc. Multiple Classifier Systems*, Cairo, Egypt, Apr. 2010, pp. 124–133.

[49] Q. Zhang, S. Goldman *et al.*, "EM-DD: an improved multiple-instance learning technique," in *Proc. Advances in Neural Information Processing Systems*, vol. 14, Vancouver, BC, Canada, Dec. 2001, pp. 1073–1080.

[50] S. Andrews, I. Tsochantaris, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proc. Advances in Neural Information Processing Systems*, vol. 15, Vancouver, BC, Canada, Dec. 2002, pp. 561–568.

[51] P. Viola, J. Platt, and C. Zhang, "Multiple instance boosting for object detection," in *Proc. Advances in Neural Information Processing Systems*, vol. 18, Vancouver, BC, Canada, Dec. 2006, pp. 1417–1426.

[52] R. P. W. Duin, P. Juszczak, P. Paclík, E. Pekalska, D. De Ridder, D. M. J. Tax, and S. Verzakov, "PRTools, a Matlab toolbox for pattern recognition," 2013. [Online]. Available: <http://www.prtools.org>

[53] D. M. J. Tax, "MIL, a Matlab toolbox for multiple instance learning," 2011. [Online]. Available: <http://prlab.tudelft.nl/david-tax/mil.html>

[54] C. Lai, M. J. T. Reinders, and L. Wessels, "Random subspace method for multivariate feature selection," *Pattern Recognition Letters*, vol. 27, no. 10, pp. 1067–1076, 2006.



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, learning in (non)-metric spaces, and structured data.



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. with the thesis "One-class Classification" from the Delft University of Technology, the Netherlands, under the supervision of Dr. Robert P.W. Duin. 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.



Marco Loog received an M.Sc. degree in mathematics from Utrecht University and a Ph.D. degree from the Image Sciences Institute in 2004. After the latter, joyful event, he moved to Copenhagen where he acted as assistant and, eventually, associate professor, next to which he worked as a research scientist at Nordic Bioscience. In 2008, after 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. He currently is also chair of Technical

Committee 1 of the IAPR and honorary full professor in pattern recognition at the University of Copenhagen. Marco's principal research interest is with supervised pattern recognition in all sorts of shapes and sizes.

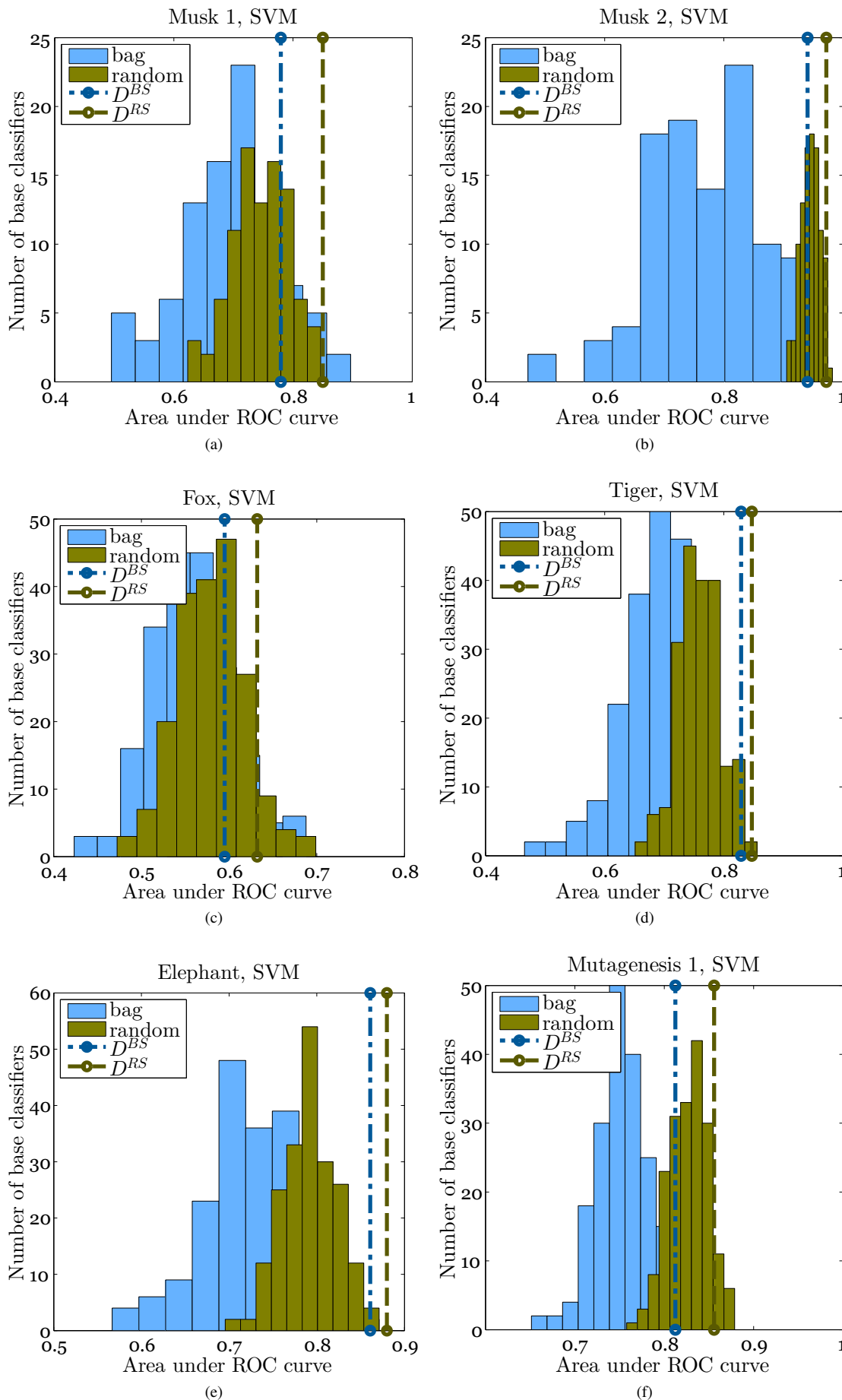


Fig. 2: Distributions of AUC performances of individual bag subspace classifiers

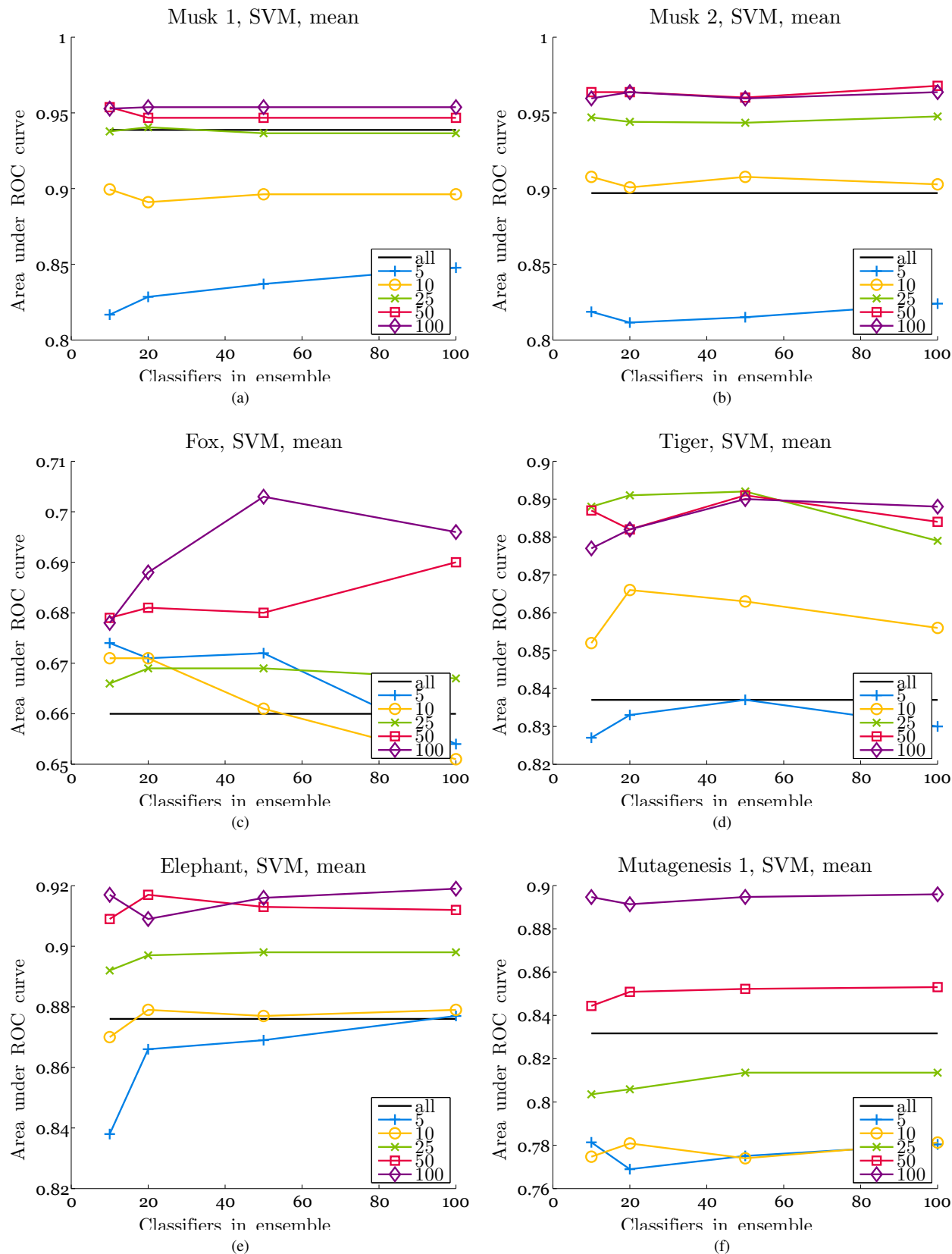


Fig. 7: AUC performances of the instance representation (black line) and the ensemble classifiers. Different lines per plot indicate different dimensionalities of subspace classifiers.