Bag Dissimilarities for Multiple Instance Learning

David M.J. Tax, Marco Loog, Robert P.W. Duin,
Veronika Cheplygina, and Wan-Jui Lee

Pattern Recognition Laboratory, Delft University of Technology,
Mekelweg 4, 2628 CD Delft, The Netherlands
D.M.J.Tax@tudelft.nl

Abstract. When objects cannot be represented well by single feature vectors, a collection of feature vectors can be used. This is what is done in Multiple Instance learning, where it is called a bag of instances. By using a bag of instances, an object gains more internal structure than when a single feature vector is used. This improves the expressiveness of the representation, but also adds complexity to the classification of the object. This paper shows that for the situation that not a single instance determines the class label of a bag, simple bag dissimilarity measures can significantly outperform standard multiple instance classifiers. In particular a measure that computes just the average minimum distance between instances, or a measure that uses the Earth Mover's distance, perform very well.

Keywords: pattern recognition, multiple instance learning, dissimilarity representation.

1 Introduction

Standard pattern recognition assumes that objects are represented by a feature vector, containing measurements on the objects that are informative for the class separability [7]. Unfortunately, for complex real world objects this is often insufficient. By using a single feature vector, much of the internal structure of the object is lost. Take for instance an image, that can contain several regions with very different characteristics: a person, a face, a tree in the background, a blue sky. It is a priori not clear how important each region is for the classification problem at hand. Only when a very clear classification task is requested, suitable features may be selected and extracted. Otherwise, the representation should be flexible enough to encode all information in the image, and let the classifier optimize its model to get a good performance.

When the representation requires more flexibility, the single feature representation may be replaced by a collection of feature vectors. For instance in the case of image classification or image retrieval, it is customary to segment the image in more-or-less homogeneous subparts, and to represent the full image by a collection of feature vectors. This is what is called Multiple Instance Learning (MIL)[5]. Objects are represented by a set (called *bag*) of feature vectors

M. Pelillo and E.R. Hancock (Eds.): SIMBAD 2011, LNCS 7005, pp. 222–234, 2011.

(called *instances*), and each object can belong to the positive or negative class. Typically, it is assumed that objects from the positive class contain at least one instance from a so-called *concept*. The task of a classifier is then to identify if one of the instances belong to the concept, and label the object then to the positive class. Many MIL algorithms therefore contain an optimization strategy to search for the most informative instance per bag, and create a model of the concept [20,13,22,1].

For the situation that no clear concept can be defined, or the situation that most instances in a bag actually contribute to the class discrimination, a more global approach in comparing bags can be defined. Instead of focusing on the single most informative instance in a bag, a similarity measure between sets of feature vectors is defined [9,15,2,12]. In most cases the goal is to define a Mercer kernel between the bags, such that a standard support vector classifier can be trained. By this one tries to implicitly reduce the complexity of a bag of instances back to a simple vector representation. The advantage is that the well understood procedures of pattern recognition can be applied, but the drawback is that a part of the representational power is lost.

When the demand for Mercer kernels is relaxed, more powerful dissimilarity measures can be defined. Actually, any (dis)similarity can be constructed, as long it may be informative for the class separability [17]. This is at the expense that it cannot be directly plugged into the support vector classifier. The alternative is then to apply a classifier that can operate on distances, like the k-nearest neighbor classifier or a nearest mean classifier, or to use a dissimilarity space approach [8,14]. In a dissimilarity space approach the dissimilarities are treated as new features, such that *any* classifier can be trained on these features. The distance character of the dissimilarities is then not used, but as features they can still contribute to a good class separation.

In this paper we propose a few simple dissimilarity measures between bags, based on pairwise dissimilarities between instances. These dissimilarities capture a more global differences between instance distributions of bags. This is done in section 2. We show in section 4 that for quite some multiple instance problems, the more global dissimilarity measures are very informative in that the classifiers trained on top of them give very good classification performance. In section 5 we conclude and have a bit more discussion on the results.

2 Bag Dissimilarities

Assume an object i is represented by a bag $B_i = \{\mathbf{x}_{ik}, k = 1, ..., n_i\}$ containing n_i instances, where each instance is represented by a vector $\mathbf{x} \in \mathbb{R}^d$. In the training set $\{(B_i, y_i),\ i = 1, ..., N\}$ each bag is labeled positive $y_i = +1$ or negative $y_i = -1$. Given the bag of instances, a classifier has to predict its class label $\hat{y}_i = f(B_i)$. First define the pairwise dissimilarities of instances in the bags B_i and B_j:

$$D_{ij} = D(B_i, B_j) = \begin{pmatrix} D(\mathbf{x}_{i1}, \mathbf{x}_{j1}) & ... & D(\mathbf{x}_{i1}, \mathbf{x}_{jn_j}) \\ D(\mathbf{x}_{i2}, \mathbf{x}_{j1}) & ... & D(\mathbf{x}_{i2}, \mathbf{x}_{jn_j}) \\ \vdots & & \vdots \\ D(\mathbf{x}_{in_i}, \mathbf{x}_{j1}) & ... & D(\mathbf{x}_{in_i}, \mathbf{x}_{jn_j}) \end{pmatrix}, \tag{1}$$

where $D(\mathbf{x}_{ik}, \mathbf{x}_{jl})$ defines the distance between instance k from bag B_i and instance l from bag B_j. In principle, any distance $D(\mathbf{x}_i, \mathbf{x}_j)$ can be used, but in this paper the squared Euclidean distance is used.

The classic approach for the classification of a bag B is to first identify a concept $C \in \mathbb{R}^d$, and to check for each instance if it is member of this concept.

$$f(B_i) = \begin{cases} +1, & \text{if} \quad \exists \mathbf{x}_{ik} \in C \\ -1, & \text{otherwise} \end{cases} \tag{2}$$

In section 3 a few approaches using concepts are explained in more depth.

Instead of focussing on the single most informative instance from a bag, a bag can be described by its full distribution of its instances. This assumes that all instances in a bag are informative about the bag label and not a single instance can determine the class label. It is then possible to define a dissimilarity matrix $d_{ij} = d(B_i, B_j)$ between bags, that is measuring the difference between (or overlap in) the distributions of B_i and B_j.

A drawback may be that the distances obtained in such manner may not be euclidean, or even metric. Therefore only methods that directly operate on distances can be applied, for instance a k-nearest neighbor (k-nearest bag) classifier would be suitable. The alternative approach is to interpret the distances to the other bags as new features, and to train classifiers on this new dissimilarity space [14]:

$$f(B_i) = f((d_{i1}, d_{i2}, ..., d_{iR})) \tag{3}$$

Typically, the distances to all training bags can be used so $R = N$, but reductions in complexity and computational requirements can be obtained when a smaller representation set is chosen $R << N$.

We did not specify the dissimilarity d_{ij} between bags yet. In this paper we consider two approaches, the first using bag distribution dissimilarities (section 2.1) and the second using the pairwise instance dissimilarities (section 2.2).

2.1 Bag Distribution Dissimilarities

To characterize bag differences in terms of differences between distributions of the instances would mean that for each bag a probability density has to be estimated, and next the difference between the distributions of two bags. It is not only very hard to estimate a high dimensional probability density function in a high dimensional feature space, it is also very computational demanding to estimate the difference, or overlap, of two distributions. Therefore approximations are made, and the following approximate distribution comparisons are considered:

Mahalanobis Distance. The distribution of each bag is approximated by a single Gaussian distribution with mean μ and covariance matrix Σ. The difference between two Gaussian distributions is computed using the Mahalanobis distance:

$$d_{ij} = (\mu_i - \mu_j)^T \left(\frac{1}{2}\Sigma_i + \frac{1}{2}\Sigma_j \right)^{-1} (\mu_i - \mu_j) \tag{4}$$

Note that the *averaged* covariance matrix is used of the covariance matrices Σ_i and Σ_j of the two bags. That means that when the number of instances per bag is low, and the feature dimensionality is high, it can become hard (or, in fact, impossible) to invert the averaged covariance matrix.

Earth Mover's Distance. The Earth Mover's distance measures the dissimilarity between two distributions p_i and p_j by measuring the effort to turn one distibution p_i, one 'pile of earth', into another one p_j. [16] In case of a discrete probability mass, the probability has to be moved over distances $D_{ij}(k, l)$ as defined in (1). For the MIL bag similarity that we consider, we assume that each instance in bag B_i contains $1/n_i$ of the total probability mass. The Earth Mover's distance is defined by the minimum amount of work that is needed to transform distribution p_i into p_j:

$$d_{ij} = \min_{f_{kl}} \sum_{k,l} f_{kl} D_{ij}(k, l) \tag{5}$$

where f_{kl} defines the flow between instance k and instance l, and with the additional constraints that $f_{kl} \geq 0$, $\forall k, l$, $\sum_l f_{kl} \leq 1/n_i$, $\sum_k f_{kl} \leq 1/n_j$ and $\sum_{kl} f_{kl} = 1$.

2.2 Pairwise Instance Dissimilarities

Instead of modeling full probability densities, the empirical distances between instances can be used.

To get a single dissimilarity measure between bags B_i and B_j, the matrix in (1) has to be reduced to a single scalar. A collection of operations O_1, .., O_5 is defined that first reduce the rows and columns of the matrix to (two) vectors, and then reduces the vectors to a scalar. In figure 1 a graphical representation of the general family of operations on the dissimilarity D_ij is shown. The first two operations perform a row and column wise reduction:

$$\tilde{\mathbf{d}}_i = O_1(D(\mathbf{x}_{i1}, \mathbf{x}_{j1}), ..., D(\mathbf{x}_{in_i}, \mathbf{x}_{j1})) \tag{6}$$

$$\tilde{\mathbf{d}}_j = O_2(D(\mathbf{x}_{i1}, \mathbf{x}_{j1}), ..., D(\mathbf{x}_{i1}, \mathbf{x}_{jn_j})) \tag{7}$$

where the individual operators reduce a vector to a scalar: $O_i : \mathbb{R}^n \rightarrow \mathbb{R}$. On these reduced vectors, the final bag dissimilarity is defined:

$$d_{ij} = O_5(O_3(\tilde{\mathbf{d}}_i), O_4(\tilde{\mathbf{d}}_j)). \tag{8}$$

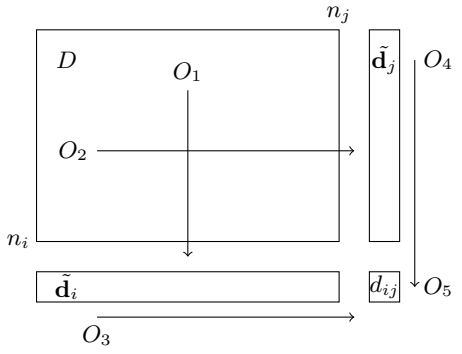

Fig. 1. The operations that can be performed on a general dissimilarity matrix D between bags B_i and B_j

(Note that d_{ij} contains a single scalar dissimilarity, while D_{ij} contains the full instance dissimilarity matrix.) Often a symmetric dissimilarity matrix is preferred, $d_{ij} = d_{ji}$, and therefore the operations are defined in a symmetric way: $O_1 = O_2$ and $O_3 = O_4$.

This reduction of the full dissimilarity matrix using these operations generalizes many approaches, depending on the choices for O_i. This results in well-known and new bag similarity measures:

Overall Minimum. $O_1 = O_2 = \min$, $O_3 = O_4 = \min$, $O_5 = \min$: Use the overall minimum pairwise distance between instances. This is expected to be quite noisy because a single instance determines the final distance between bags. When the number of instances per bag is low, and there is a very dense concept C, i.e. it is covering a small area in the feature space, this measure may actually work.

Mean Minimum Distance. $O_1 = O_2 = \min$, $O_3 = O_4 = \text{mean}$, $O_5 = \text{mean}$ The mean minimum distance between bags, where for each instance the closest instance in the other bag is found, and where the minimum distances are averaged over all the instances. This is certainly not as noise sensitive as the overall minimum, and captures more of the general similarity between the distributions of the two bags. This does not work if there is a single instance that determines the class label.

Standard Haussdorf Distance. $O_1 = O_2 = \min$, $O_3 = O_4 = \max$, $O_5 = \max$: The standard Haussdorf distance between bags, where for each instance the closest instance in the other bag is found, and from all the closest matches, the lastest distance is used to define the bag distance. The advantage is that the Haussdorf distance defines a metric, but it is sensitive to a single outlier instance, that can dominate the full bag distance.

Modified Haussdorf. $O_1 = O_2 = \min$, $O_3 = O_4 = \max$, $O_5 = \min$: The modified Haussdorf distance between bags [6] that is less sensitive to single outliers.

2.3 Linear Assignment Dissimilarity

The operations that are defined in (8) matches instances independently of each other; each element in (6) or (7) are computed individually. By performing a linear assignment [11], instances in bag B_i are matched to bag B_j. When one bag is larger than the other, instances of the largest bag are not matched, and will not contribute to the distance between the two bags. Define $I_{kl} = 1$ when instances k and l are matched, and $I_{kl} = 0$ otherwise, then the bag dissimilarity is defined as:

$$d_{ij} = \sum_{k,l} I_{kl} D_{ij}(k, l). \tag{9}$$

3 Standard MIL Classifiers

The original model proposed by [5] was an axis-parallel rectangle that was grown and shrunk to best cover the area of the concept. Several parameters determine the optimization of the rectangle, and one of them (τ) defines a slight extrapolation around to box to become a bit resistant against noise. It is applied to a drug discovery problem where molecules have to be distinguished based on their shape into active and inactive molecules. It appears that this rectangular model fits well with the molecule shape classification, but it is less successful in other applications.

A probabilistic description of the MIL problem was given by [13]. The concept is modeled by a general probabilistic model, where typically an axis-parallel Gaussian is used. Unfortunately, the optimization of the parameters requires a computationally expensive maximization of an likelihood that is adapted to include the constraint that at least one of the instances in a positive bag has a high concept probability. Because the error landscape is very wild, several random initialisations are tried, and the solution with the highest likelihood is used.

Newer methods often avoid the modeling of the concept by a density model, and try to separate concept instances from background instances using a discriminative approach. Two of them include the MISVM [1] and the MiBoost [19]. The first uses a support vector classifier, in which one instance from each positive bag is selected as being the 'witness', i.e. each bag is reduced to its most positive member. The second is a variant of boosting, where in each boosting step a weight per instance is updated. The weight indicates how informative this instance seems to be in the prediction of the class label of the bag.

The above mentioned methods assume the presence of a concept. Other methods avoid this assumption, and try to apply standard pattern recognition techniques directly to the MIL problem. The first approach is to extract features from the bag of instances, like the average instance, or the minimum and maximum feature values that appear in the bag, and train a standard classifier on this feature vector [9]. A second approach is to ignore the MIL problem and to label all instances according to their bag label. [21] Then a standard classifier can be fitted to the fully labeled instance dataset. To classify a new bag of instances, first all instances are classified, and then a simple combining rule like taking the maximum, or majority

voting is applied. Finally, an idea similar to the bag of words in the natural language processing can be applied. In particular, in MILES [4] all instances in the training set are considered words (or potential concepts), and new bags are represented by their similarity to each of the words. On these long similarity vectors a sparse classifier is fitted to select the most informative words.

4 Experiments

To show the benefits and limitations of the bag similarities, classification experiments are performed on some standard real world MIL datasets. The datasets often deal with image classification, where with different procedures segments are extracted, different features per segment are computed and different classes are defined. [3,1,4]. Two non-image problems are the classical drug discovery problems Musk1 and Musk2, in which molecules are described by 166 shape features [5], and the webpage classification, in which webpages are described by a collection of pages that have links to the original page. In table 1 some characteristics are shown of the datasets that are considered in this paper. The datasets are chosen to show some variability in the number of features, the number of bags, and the average number of instances per bag.

Table 1. Some characteristics of the standard MIL datasets used in this paper

dataset	nr.inst.	dim.	pos. bags	neg. bags	min. inst/bag	median inst/bag	max. inst/bag
MUSK 1 [5]	476	166	47	45	2	4	40
MUSK 2 [5]	6598	166	39	63	1	12	1044
Corel African [4]	7947	9	100	1900	2	3	13
Corel Historical [4]	7947	9	100	1900	2	3	13
SIVAL AjaxOrange [10]	47414	30	60	1440	31	32	32
Web atheism [23]	5443	200	50	50	22	58	76
Web motorcycles [23]	4730	200	50	50	22	49	73
Web mideast [23]	3373	200	50	50	15	34	55
Corel Fox [1]	1320	230	100	100	2	6	13
Corel Tiger [1]	1220	230	100	100	1	6	13
Corel Elephant [1]	1391	230	100	100	2	7	13

In tables 2, 3 and 4 the results of the classifiers mentioned in Section 2 are shown. Three different types of classifiers are used: the standard MIL classifiers in the top block, the k-nearest neighbor that is directly operating on the distances defined in Section 2 given in the middle block, and finally classifiers that use the distances as features in the last block.

For the Axis-parallel Rectangle classifier (APR) the τ parameter is varied, because that appears to have the most significant influence on the performance. The other parameters are fixed. For the Diverse Density 100 random restarts of the optimization is chosen. In the miBoost the number of boosting runs was set to $M = 100$. For the MI-SVM and MILES the kernel was chosen to be an RBF

kernel, where the width parameter σ was roughly optimized (using 5 candidates). For the MI-SVM the linear kernel was also applied for comparison.

The more simple MIL classifiers includes first the Linear Discriminant Analysis (LDA) trained on all instances, with a maximum combination rule to get from instance to bag labels. The next two classifiers represent a bag of instances by the mean instance (where the feature values are averaged) or the minimum and maximum feature value, respectively. On this new feature vector a LDA is trained. The last simple MIL classifier applies a bag of words approach, where first k cluster centers are obtained by applying k-means clustering on all instances, next the bags are represented by the number of instances that are assigned to each cluster, and finally a (linear) support vector classifier is trained on the histograms.

The standard MIL classifier are compared to the classifiers that work with the bag dissimilarities. Five different dissimilarities are considered here, the 'Overall Minimum' (minmin.) dissimilarity, the 'Mean Minimum' (mindist) distance, the 'Hausdorff' (haussd.) distance, the Mahalanobis (mahal.) distance, the Earth Mover's distance (emd) and, finally, the linear assignment (lin.ass.) distance. The classifier that is used for classifying distance data is the k-nearest neighbor. The k is optimized on the training set using leave-one-out crossvalidation.

Furthermore, all classifiers are implemented, trained and evaluated using a Matlab toolbox [18]. In quite some cases the performance as mentioned in the literature could not be reproduced. This might be caused by the fact that the optimization of the free parameters in the methods was not so extensive as in the original papers. In this paper a reasonable range of parameters was chosen and an internal crossvalidation was used to find the final optimal value. In some cases (in particular the Diverse Density) the optimization was so slow, that just a fixed parameter setting was chosen. Furthermore, all features have been rescaled to zero mean and unit variance on the training set. The reported performance is the area under the ROC curve ($\times 100$). A performance of 50.0 means that the two classes are not separated at all, a performance of 100.0 is perfect.

From the results in Tables 2, 3 and 4 several things can be concluded:

Datasets that contain a clear concept often do not gain much by the use of bag similarities. That is visible in datasets Musk 1, Musk 2, AjaxOrange, Corel Tiger and Corel Elephant. For datasets in which many instances contain some information about the class label, like in the webpage classification, but also a bit in Corel African, Corel Historical and Corel Fox, the bag dissimilarity measures are informative.

It is not always the case that using a nearest neighbor classifier on the distances gives the highest performance. In particular on the webpage classification problems significant improvements can be made by using a k-nearest neighbor classifier (or a Parzen classifier) in the dissimilarity space. On the other hand, on the Corel African and Corel Historical datasets, training a classifier in the dissimilarity space slightly deteriorates the results. This is probably caused by the fact that the dissimilarity space is quite large here because the number of training bags is high: 90% of 2000 = 1800D.

Table 2. AUC performances (100×) of the classifiers on datasets Musk1, Musk2, Corel African and Corel Historical. Results are obtained using five times 10-fold stratified crossvalidation. Results [(1)] cannot be obtained because some bags in Musk2 are too large to compute the Earth Mover's distance between bags.

classifier	Musk 1	Musk 2	Corel African	Corel Historical
Standard MIL classifiers				
APR $\tau = 0.999$	81.8 (1.3)	82.5 (1.2)	50.5 (0.0)	50.5 (0.1)
APR $\tau = 0.995$	78.9 (1.7)	80.8 (2.3)	57.4 (0.8)	61.4 (0.4)
Diverse Density (100 restarts)	89.4 (1.3)	**93.2 (0.0)**	85.6 (0.1)	83.4 (0.7)
MiBoost ($M = 100$ rounds)	80.3 (3.1)	49.3 (3.7)	68.0 (0.0)	80.4 (1.6)
MI-SVM (linear kernel)	70.3 (3.0)	81.5 (2.1)	63.4 (2.0)	78.9 (0.6)
MI-SVM (RBG kernel)	**92.9 (1.3)**	NaN (0.0)	NaN (0.0)	**90.8 (1.0)**
MILES (RBF kernel)	**92.8 (1.4)**	**95.3 (1.5)**	58.9 (9.2)	60.8 (12.8)
Simple MIL with LDA, max-comb.	72.9 (3.4)	76.7 (3.4)	68.8 (0.2)	74.4 (0.2)
LDA on mean-inst	85.7 (1.4)	87.6 (2.8)	77.2 (0.3)	86.2 (0.1)
LDA on extremes	**92.4 (1.9)**	88.9 (4.0)	88.5 (0.1)	85.3 (0.2)
BagOfWords (k=10)+linear SVM	72.7 (4.7)	63.7 (6.1)	75.1 (3.2)	78.4 (3.9)
BagOfWords (k=100)+linear SVM	78.7 (5.5)	71.2 (3.1)	83.4 (1.8)	85.6 (2.6)
Distance-based classifiers on bag dissimilarities				
minmin+k-NND	90.1 (1.4)	84.0 (1.9)	86.6 (0.4)	84.1 (1.2)
mindist+k-NND	86.3 (2.0)	83.2 (1.6)	**92.7 (0.7)**	**90.7 (1.1)**
haussd.+k-NND	89.0 (1.6)	84.2 (0.8)	86.7 (0.9)	88.5 (1.0)
mahal.+k-NND	61.8 (2.8)	65.7 (5.7)	67.3 (0.7)	63.2 (1.2)
emd+k-NND	90.1 (2.7)	[(1)]	**92.0 (0.7)**	88.8 (1.7)
lin.ass.+kNND	84.7 (1.6)	76.5 (2.7)	69.9 (0.6)	87.8 (0.4)
Standard classifiers on bag dissimilarity space				
minmin.+Parzen Classifier	**94.7 (3.0)**	92.3 (2.7)	90.4 (0.6)	84.0 (0.6)
mindist.+Parzen Classifier	61.2 (6.0)	50.0 (0.0)	83.4 (0.9)	86.0 (0.5)
haussd.+Parzen Classifier	86.9 (0.7)	92.1 (2.5)	79.1 (0.6)	84.3 (0.5)
mahal.+Parzen Classifier	52.1 (0.9)	65.8 (2.4)	46.3 (2.4)	52.4 (1.3)
emd+Parzen Classifier	87.4 (3.4)	[(1)]	89.4 (0.4)	85.4 (0.7)
lin.ass.+Parzen Classifier	83.3 (2.7)	72.2 (2.9)	83.5 (0.7)	86.2 (0.5)
minmin.+k-NN	**93.3 (1.5)**	90.7 (3.9)	88.7 (0.8)	83.5 (1.3)
mindist.+k-NN	88.8 (3.0)	83.8 (1.4)	81.7 (1.1)	85.5 (1.0)
haussd.+k-NN	89.2 (2.7)	91.6 (1.0)	77.0 (0.7)	80.0 (1.3)
mahal.+k-NN	72.0 (3.1)	61.6 (2.7)	53.3 (1.6)	57.0 (0.8)
emd+k-NN	**92.4 (1.4)**	[(1)]	86.9 (1.1)	79.6 (1.5)
lin.ass.+k-NN	88.6 (2.1)	72.6 (3.7)	81.5 (1.4)	84.7 (1.4)

Table 3. AUC performances (100×) of the classifiers on datasets SIVAL AjaxOrange, webpage Atheism, webpage Motorcycles and webpage Mideast. Results are obtained using five times 10-fold stratified crossvalidation. Results [(2)] cannot be obtained because the linear programming optimizer required more than 128GB of memory, which was not available.

classifier	AjaxOrange	alt.atheism	rec.motorcycles	politics.mideast
Standard MIL classifiers				
APR $\tau = 0.995$	48.4 (0.8)	50.0 (0.0)	50.0 (0.0)	49.8 (0.4)
Diverse Density (100 restarts)	55.5 (2.9)	52.2 (2.4)	46.4 (2.9)	40.2 (2.5)
MiBoost ($M = 100$ rounds)	56.5 (2.4)	50.0 (0.0)	NaN (0.0)	50.3 (1.5)
MI-SVM (linear kernel)	**93.6 (2.6)**	69.8 (2.8)	76.4 (4.0)	79.8 (2.3)
MI-SVM (RBG kernel)	NaN (0.0)	45.5 (7.1)	49.7 (5.4)	46.1 (2.4)
MILES (RBF kernel)	[(2)]	47.1 (4.5)	44.7 (4.8)	54.1 (1.8)
Simple MIL with LDA, max-comb.	89.3 (0.3)	81.6 (1.2)	80.4 (2.1)	75.0 (3.1)
LDA on mean-inst	82.3 (0.9)	83.7 (2.1)	84.4 (1.8)	78.1 (1.7)
LDA on extremes	90.3 (0.3)	50.0 (0.0)	51.2 (0.4)	65.0 (1.8)
BagOfWords (k=100)+linear SVM	81.2 (2.5)	54.0 (0.0)	65.2 (9.3)	58.6 (6.8)
Distance-based classifiers on bag dissimilarities				
minmin+k-NND	53.6 (1.2)	50.0 (0.0)	50.0 (0.0)	52.8 (2.2)
mindist+k-NND	62.9 (1.3)	59.2 (1.9)	58.4 (0.5)	53.4 (1.1)
haussd.+k-NND	72.4 (1.3)	72.8 (3.0)	68.7 (3.2)	67.1 (1.8)
mahal.+k-NND	64.0 (1.6)	47.7 (4.4)	45.0 (3.4)	58.5 (6.0)
emd+k-NND	77.6 (2.6)	56.0 (1.2)	60.8 (0.4)	57.2 (1.3)
lin.ass.+kNND	71.6 (1.4)	69.2 (1.7)	53.7 (2.9)	58.5 (3.2)
Standard classifiers on bag dissimilarity space				
minmin.+Parzen Classifier	55.7 (1.6)	49.8 (0.4)	50.0 (0.0)	50.4 (2.3)
mindist.+Parzen Classifier	78.0 (1.3)	78.9 (2.8)	78.4 (0.5)	75.2 (1.9)
haussd.+Parzen Classifier	71.8 (0.9)	73.8 (2.0)	82.0 (2.2)	73.8 (0.9)
mahal.+Parzen Classifier	75.3 (0.9)	54.2 (3.3)	43.7 (3.5)	61.9 (1.8)
emd+Parzen Classifier	78.7 (1.1)	**89.7 (1.3)**	77.6 (1.5)	**87.8 (1.1)**
lin.ass.+Parzen Classifier	78.9 (0.6)	80.1 (2.4)	84.2 (2.8)	84.3 (3.1)
minmin.+k-NN	56.0 (1.6)	50.0 (0.0)	50.0 (0.0)	47.8 (2.7)
mindist.+k-NN	70.6 (2.6)	84.9 (1.6)	**86.6 (2.0)**	82.2 (1.5)
haussd.+k-NN	68.9 (1.9)	85.6 (2.1)	**89.2 (3.5)**	77.2 (3.2)
mahal.+k-NN	70.8 (1.5)	51.2 (3.6)	56.3 (3.8)	55.8 (4.6)
emd+k-NN	72.0 (2.4)	**90.0 (1.4)**	**86.7 (0.7)**	82.6 (1.7)
lin.ass.+k-NN	70.1 (0.8)	82.1 (2.3)	82.9 (2.4)	80.8 (3.8)

Table 4. AUC performances (100×) of the classifiers on datasets Corel Fox, Corel Tiger, and Corel Elephant. Results are obtained using five times 10-fold stratified crossvalidation.

classifier	Corel Fox	Corel Tiger	Corel Elephant
Standard MIL classifiers			
APR $\tau = 0.995$	55.2 (1.2)	57.9 (1.6)	74.6 (3.2)
Diverse Density (100 restarts)	66.5 (1.6)	79.3 (0.2)	90.8 (0.0)
MiBoost ($M = 100$ rounds)	53.5 (1.4)	74.2 (1.3)	88.9 (1.3)
MI-SVM (linear kernel)	54.4 (1.5)	80.1 (1.1)	84.1 (1.3)
MI-SVM (RBF kernel)	69.6 (1.4)	**86.5 (1.4)**	91.1 (1.2)
MILES (RBF kernel)	69.8 (1.7)	**87.2 (1.7)**	88.3 (1.1)
Simple MIL with LDA, max-comb.	57.9 (1.4)	83.4 (1.3)	**90.8 (1.6)**
LDA on mean-inst	58.5 (2.8)	**86.5 (1.2)**	89.7 (1.3)
LDA on extremes	62.9 (3.0)	84.8 (1.0)	**91.3 (1.3)**
BagOfWords (k=10)+linear SVM	51.8 (4.6)	71.2 (4.0)	73.0 (1.9)
Distance-based classifiers on bag dissimilarities			
minmin+k-NND	65.7 (1.3)	83.4 (1.2)	83.4 (1.1)
mindist+k-NND	63.9 (1.5)	76.4 (1.3)	87.9 (1.7)
haussd.+k-NND	63.5 (3.0)	80.9 (1.2)	80.9 (2.2)
mahal.+k-NND	58.8 (2.9)	58.8 (2.9)	66.3 (4.1)
emd+k-NND	61.3 (2.1)	85.5 (0.9)	86.8 (2.3)
lin.ass.k+NND	57.5 (4.1)	78.9 (2.4)	72.8 (2.5)
Standard classifiers on bag dissimilarity space			
minmin.+Parzen Classifier	61.2 (4.0)	74.3 (2.8)	86.7 (0.7)
mindist.+Parzen Classifier	62.3 (1.9)	70.7 (1.2)	74.9 (4.2)
haussd.+Parzen Classifier	59.8 (1.7)	66.9 (2.2)	73.2 (1.1)
mahal.+Parzen Classifier	68.9 (3.4)	68.9 (3.4)	64.9 (1.7)
emd+Parzen Classifier	54.3 (2.1)	67.8 (1.5)	76.5 (2.2)
lin.ass.+Parzen Classifier	64.4 (1.9)	64.6 (1.4)	69.6 (2.1)
minmin.+k-NN	67.0 (1.4)	78.6 (1.4)	87.8 (1.1)
mindist.+k-NN	59.6 (3.1)	73.7 (1.3)	76.0 (1.8)
haussd.+k-NN	56.7 (3.8)	70.6 (1.9)	77.8 (0.9)
mahal.+k-NN	**75.0 (3.8)**	75.0 (3.8)	65.6 (1.1)
emd+k-NN	61.4 (0.9)	76.5 (0.6)	76.3 (1.0)
lin.ass.+k-NN	65.0 (3.1)	68.7 (2.9)	71.8 (2.3)

5 Conclusions

In some MIL problems not a single instance may be decisive, but the full distribution of all the instances in a bag. For these situations bag dissimilarities are defined that characterize the difference in distribution between bags. For the webpage classification problem this resulted in very good performances, while for other problems, where a single concept can be expected, the bag dissimilarity is far less successful. It seems that most webpages that link to another webpage, contain information about the linked-to webpage, and therefore selecting just one single most informative webpage is not optimal. For other problems, like the image classification problem, the different segments appear to be more independent, in that detecting the single most informative segment is often best. This effect is also enhanced by the fact that in the image classification problems images often do not have many segments (around 3-6), so it is hard to treat these few instances as a distribution.

When the given the bag dissimilarities are interpreted as new features to represent the bag, a classifier can be trained on these distance features. In this paper only the k-nearest neighbor and the Parzen classifier are considered. Although the choice of the classifier has some influence on the final performance, the choice of the bag dissimilarity is more important. One well-performing dissimilarity is using the Earth Mover's Distance.

Acknowledgments. We acknowledge the financial support from the FET programme within the EU FP7, under the project "Similarity-based Pattern Analysis and Recognition - SIMBAD" (contract 213250).

References

1. Andrews, S., Hofmann, T., Tsochantaridis, I.: Multiple instance learning with generalized support vector machines. In: Proceedings of the AAAI National Conference on Artificial Intelligence (2002)
2. Blaschko, M.B., Hofmann, T.: Conformal multi-instance kernels. In: NIPS 2006 Workshop on Learning to Compare Examples, pp. 1–6 (2006)
3. Carson, C., Thomas, M., Belongie, S., Hellerstein, J.M., Malik, J.: Blobworld: A system for region-based image indexing and retrieval. In: Huijsmans, D.P., Smeulders, A.W.M. (eds.) VISUAL 1999. LNCS, vol. 1614, pp. 509–517. Springer, Heidelberg (1999)
4. Chen, Y., Bi, J., Wang, J.: MILES: Multiple-instance learning via embedded instance selection. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(12), 1931–1947 (2006)
5. Dietterich, T., Lathrop, R., Lozano-Perez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence 89(1-2), 31–71 (1997)
6. Dubuisson, M., Jain, A.: Modified Hausdorff distance for object matching. In: 12th Internat. Conference on Pattern Recognition, vol. 1, pp. 566–568 (1994)
7. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. John Wiley & Sons, Chichester (2001)

8. Duin, R.P., Pekalska, E.: The dissimilarity space: bridging structural and statistical pattern recognition. Pattern Recognition Letters (in press, accepted manuscript 2011)
9. Gärtner, T., Flach, P., Kowwalczyk, A., Smola, A.: Multi-instance kernels. In: Sammut, C., Hoffmann, A. (eds.) Proceedings of the 19th International Conference on Machine Learning, pp. 179–186. Morgan Kaufmann, San Francisco (2002)
10. Goldman, S.: SIVAL (spatially independent, variable area, and lighting) benchmark (1998), http://www.cs.wustl.edu/~sg/accio/SIVAL.html
11. Kuhn, H.: The hungarian method for the assignment problem. Naval Research Logistics Quarterly 2, 83–97 (1955)
12. Kwok, J.T., Cheung, P.M.: Marginalized multi-instance kernels. In: Proceedings of International Joint Conference on Artificial Intelligence, pp. 901–906 (2007)
13. Maron, O., Lozano-Pérez, T.: A framework for multiple-instance learning. In: Advances in Neural Information Processing Systems, vol. 10, pp. 570–576. MIT Press, Cambridge (1998)
14. Pekalska, E.: The Dissimilarity representations in pattern recognition. Concepts, theory and applications. Ph.D. thesis, Delft University of Technology (January 2005)
15. Ray, S., Craven, M.: Supervised versus multiple instance learning: an empirical comparison. In: ICML 2005: Proceedings of the 22nd International Conference on Machine Learning, pp. 697–704. ACM, New York (2005)
16. Rubner, Y., Tomasi, C., Guibas, L.: A metric for distributions with applications to image databases. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 59–66 (1998)
17. Sörensen, L., Loog, M., Tax, D.M.J., Lee, W.J., de Bruijne, M., Duin, R.P.W.: Dissimilarity-based multiple instance learning. In: Hancock, E.R., Wilson, R.C., Windeatt, T., Ulusoy, I., Escolano, F. (eds.) SSPR&SPR 2010. LNCS, vol. 6218, pp. 129–138. Springer, Heidelberg (2010)
18. Tax, D.: MIL, a Matlab toolbox for multiple instance learning, version 0.7.9 (May 2011), http://prlab.tudelft.nl/david-tax/mil.html
19. Viola, P., Platt, J., Zhang, C.: Multiple instance boosting for object dection. In: Advances in Neural Inf. Proc. Systems (NIPS 2005), pp. 1419–1426 (2005)
20. Weidmann, N., Frank, E., Pfahringer, B.: A two-level learning method for generalized multi-instance problems. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) ECML 2003. LNCS (LNAI), vol. 2837, pp. 468–479. Springer, Heidelberg (2003)
21. Xu, X., Frank, E.: Logistic regression and boosting for labeled bags of instances. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 272–281. Springer, Heidelberg (2004)
22. Zhang, Q., Goldman, S.: EM-DD: An improved multiple-instance learning technique. In: Advances in Neural Information Processing Systems, vol. 14, MIT Press, Cambridge (2002)
23. Zhou, Z.H., Jiang, K., Li, M.: Multi-instance learning based web mining. Applied Intelligence 22(2), 135–147 (2005)