ReliefF-MI: An extension of ReliefF to Multiple Instance Learning

Amelia Zafra[*,a], Mykola Pechenizkiy[b], Sebastián Ventura[a]

[a]*Department of Computer Science and Numerical Analysis. University of Cordoba*
[b]*Department of Computer Science. Eindhoven University of Technology*

Abstract

In machine learning the so-called curse of dimensionality, pertinent to many classification algorithms, denotes the drastic increase in computational complexity and classification error with data having a great number of dimensions. In this context, feature selection techniques try to reduce dimensionality finding a new more compact representation of instances selecting the most informative features and removing redundant, irrelevant, and/or noisy features. In this paper, we propose a filter-based feature selection method for working in the multiple-instance learning scenario called ReliefF-MI; it is based on the principles of the well-known ReliefF algorithm. Different extensions are designed and implemented and their performance checked in multiple instance learning. ReliefF-MI is applied as a pre-processing step that is completely independent from the multi-instance classifier learning process and therefore is more efficient and generic than wrapper approaches proposed in this area. Experimental results on five benchmark real-world data sets and seventeen classification algorithms confirm the utility and efficiency of this method, both statistically and from the point of view of execution time.

Key words: Multiple Instance Learning, Feature Selection, Relief Algorithm

1. Introduction

Multiple Instance Learning (MIL) is a generalization of traditional supervised learning which deals with the uncertainty of instance labels. Thus, instead of receiving a set of instances which are labeled positive or negative, the learner receives a set of bags that are labeled positive or negative. Each bag can contain a different number of instances, whose labels remain unknown. The goal of this learning paradigm is to learn a concept that correctly classifies training data and generalizes unseen data. The difficulty of MIL lies in the fact that there is no information as to which, or how many, of the instances are actually positive, and this lack of information complicates the learning process.

MIL has been actively studied in the last few years. This is because it has been found that in many applications MIL is a more natural and appropriate form of problem formulation and representation that allows achieving an improvement over results obtained by traditional supervised

[*] Department of Computer Science and Numerical Analysis. University of Cordoba. Campus Universitario Rabanales, Edificio Einstein, Tercera Planta. 14071 Cordoba. Spain. Tel: +34957212031; Fax: +34957218360

Email addresses: azafra@uco.es (Amelia Zafra), m.pechenizkiy@tue.nl (Mykola Pechenizkiy), sventura@uco.es (Sebastián Ventura)

learning. Particular examples of such applications include text categorization (1), content-based image retrieval (2; 3), image annotation (4; 5), drug activity prediction (6), web index page recommendation (7; 8), semantic video retrieval (9), video concept detection (10; 11), pedestrian detection (12), and the prediction of student performance (13). In all of these applications an instance space is typically represented by hundreds or thousands of variables or extracted features. In pattern recognition, machine learning, data mining and related fields a high dimensionality of the data is known to be one of the main challenges to address.

High dimensionality of the data as such brings serious complications to the learning process and often referred to the so-called curse of dimensionality (14), the problem caused by the exponential increase in volume associated with adding extra dimensions to a representation space and a consequence the notions of distance, proximity and density become less meaningful. Besides, the representation space may contain a number of irrelevant, redundant or noisy features that can mislead a machine learning algorithm and adversely affect its performance. MIL algorithms are not an exception, they also suffer from these problems.

Feature selection is one way for reducing dimensionality by choosing a subset of features from among original ones. Feature selection has been shown to have a positive effect on efficiency, effectiveness and comprehensibility of machine learning (15). However, while there exists a large body of literature devoted to feature selection for traditional supervised learning tasks (16; 17; 18) where various filter-based, wrapper-based, embedded and more recently hybrid (19; 20) models and strategies for feature selection have been designed, in MIL far less research has been done. Here, due to the fact the labels are known for each bag as a whole, but not for the individual instances a bag consists of, utilizing class labels in feature selection becomes less straightforward. Specifically, the uncertainties that surround this type of inductive learning do that the problem appears to be more difficult.

Major existing works on feature selection for MIL include the following. Zhang and Zhou (21) improved a previous method for MIL, BP-MIP, through adopting two different feature selection techniques (feature scaling with Diverse Density and feature reduction with principal component analysis). Yuan et al. (22) proposed an algorithm, named MI-AdaBoost, that firstly maps each bag into a new bag feature space using a certain set of instance prototypes, and then adopts AdaBoost to select the bag features and build classifiers simultaneously. Raykar et al. (23) proposed a Bayesian MIL algorithm that uses feature selection.

Although these works provide an important contribution to MIL in dealing with high dimensional data, they have two main limitations. First, all these approaches have a flavor of wrapper methods, i.e. they utilize the learning machine of interest as a black box to score subsets of features according to their predictive power. Therefore, they require the use of a multi-instance algorithm to evaluate each feature subset considered in the search space explored with a certain feature subset selection strategy. As a consequence they are hardly applicably for domains containing thousands of features due to high computational time. Second, published works on feature selection in MIL are difficult to assess and compare altogether due to the lack of a comprehensive empirical evaluation. Indeed, when a wrapper or embedded approach for feature selection in MIL is proposed, typically it is compared against the performance of the same and possibly few other MIL algorithms that do not employ any feature selection at all. Moreover, in many cases the results reported in these studies are scattered and not directly comparable to each other due to different or incompatible experimental settings. Surprisingly, no work published on this topic considered earlier proposed approaches in the experimental evaluation.

This paper presents a first attempt to fill this gap by proposing a filter-based feature selection method, called ReliefF-MI, to deal with multiple-instance problems. This method can be used as a preprocessing step before applying the multi-instance classifier learning process, i.e. it is completely independent of the MIL classification algorithm to be used. This ensures that it is efficient not only from a statistical point of view but also from a computational point of view. ReliefF-MI is based on the principles of the well-known ReliefF algorithm (24), extended to select features in this learning paradigm by modifying the distance, the difference function, and the computation of the weight of the features. Some properties of this method are that it can be applied to both continuous and discrete problems; it includes interaction among features, and may capture local dependencies that other methods miss. To evaluate the performance of the proposed filter approach, we consider four different versions of ReliefF-MI modifying the similarity function to calculate the distance between patterns and the weights of the different features.

In our experimental study we compare the performance of the most popular MIL approaches based on diverse density, logistic regression, support vector machines, nearest neighbours, decision trees, rules, and probabilistic methods; Concretely, seventeen algorithms are used to guarantee the effectiveness of the feature reduction provided by ReliefF-MI. The experimental results on five different datasets show, on the one hand, that the new similarity function designed arranges for ReliefF-MI to achieve the best results. On the other hand, it is proven that multi-instance learning algorithms for classification achieve theoretical superiority when they use the subset of features provided by ReliefF-MI as a preprocessing step compared to when they work with the whole feature set. At the same time, feature selection is also shown to reduce training and inference time and it leads to better data measurement reduction and storage requirements. Finally, to facilitate future comparisons in this area, all the datasets and algorithms used are available. The experimental section contains all necessary information to reproduce these results at any time. This point is very relevant for real progress in feature selection in MIL because new proposals can then be compared under the same conditions.

This paper is structured as follows. In Section 2 we describe the proposed ReliefF-MI approach and its different modifications. In Section 3 we provides the results of the extensive experimental study comparing the performance of the proposed and other state-of-the-art approaches. Finally, in Section 4 we draw conclusions and raise several issues for future work.

2. The ReliefF-MI Algorithm

This section describes the algorithm, together with its different extensions. First, the main steps and the general philosophy of the algorithm are given and then, different extensions and the adaptations necessary to work correctly with multiple instance data are commented on.

2.1. General Description of the Method

The procedure is based on the principles of the ReliefF algorithm (24) and its main steps can be seen in Listing 1. This method works by randomly sampling examples (most commonly called bags in MIL) from the training data. For each sampled bag R, its k nearest neighbours from the same class (called nearest hits) are found as well as the opposite class of each sampled instance (called nearest misses). Multi-class datasets are handled by finding the nearest neighbours from each class that are different from the currently sampled instance, and weighting their contributions by the prior probability of each class estimated from the training data. The weight updating $W[A]$ of attribute A uses the equation in line 11 in Listing 1. Its value is the average of all examples of the magnitude of the difference in distance to the k nearest hits and the distance to the k nearest misses, projecting on the A attribute. Each weight reflects its ability to

Listing 1 – ReliefF-MI Method

```
Set selectedSubset = $\oslash$.
Set W = 0.
Set m = value specified by user. It represents the number of times that the process is repeated.
Set k = value specified by user. It represents the number of nearest examples of the same and different class considered in the process.
for i=1 to numberExamples do
    Get one example/bag $R_i$ from the training data set D.
    Get $k$ nearest hits: $H_1, \ldots, H_k$ ($H_j$ example/bag in D where dist($R_i$,$H_j$) is $j^{th}$ closest & $R_i$.class=$H_j$.class)
    for each class C <> Class ($R_i$) do
        Get $k$ nearest misses $M_1, \ldots, M_k$ ($M_j$ example/bag in D where dist($R_i$,$M_j$) is $j^{th}$ closest & $R_i$.class=$M_j$.class )
    end for
    for A=1 to numberFeatures do
        Set $W[A] = W[A] - \frac{\sum_{j=1}^{k} diff_{bag}(A, R_i, H_j)}{m \cdot k} + \sum_{C \neq Class(R_i)} \left[ \frac{\frac{P(C)}{1-P(class(R_i))} \sum_{j=1}^{k} diff_{bag}(A, R_i, M_j(C))}{m \cdot k} \right]$
    end for
end for
for i=1 to numberFeatures do
    if W[i] > threshold then
        Add feature $i$ to selectedSubset.
    end if
end for
```

distinguish class labels, thus a high weight indicates that there is differentiation to this attribute among instances from different classes and it has the same value for instances in the same class. Features are ranked by weight and those that exceed a user-specified threshold are selected to form the final subset. The following section studies the calculation of the nearest neighbour and the definition of the $diff_{bag}$ function applied to the bags.

Fig. 1 shows one of the iterations of this algorithm using three nearest neighbours. The example or bag selected in this iteration is called R, and its three nearest neighbours of the same class (H_1, H_2, H_3) are selected as well as three of a different class (M_1, M_2, M_3). The information about these neighbours are used to update the weights. Although the figure shows the class label for each particular instance, this information is unknown in the learning process and it is indicated only for didactic objectives. The only information available is that if the example, pattern, or bag, is positive, then at least one instance is positive. However, there is no information about which set of them is the positive one.

2.2. Different versions of ReliefF-MI

The original ReliefF algorithm estimates the quality of attributes by how well their values distinguish between patterns that are near to each other. In traditional supervised learning each pattern is assumed to contain only one instance. In MIL, the distance between two patterns has to be calculated taking into account that each pattern contains one or more instances. So, the similarity function needs to be upgraded because the Manhattan distance measure schema is not applicable. The difference between the two cases can be seen in Fig. 2. This figure shows the calculation of the distance between two patterns or examples of the data set, considering both traditional supervised learning and multiple instance learning. In this example each instance is composed of an n feature vector. Fig. 2(a) shows the calculation in a traditional supervised learning scenario. In this case, the correspondence between pattern and instance is one to one and a simple Manhattan distance could be employed. Fig. 2(b) shows the case in MIL where the correspondence between pattern and instance is one to many, therefore the distance between the two patterns is different. In this case, the pattern has, in the case of bag A, nine different instances and for B, six different instances (here, each bag may contain a different number of instances). Therefore, Manhattan distance is not possible in this scenario and other suggestions have to be proposed.

The literature proposes different distance-based approaches to solve MI problems (25; 26; 27). The most extensively used metric is Hausdorff distance (28), which measures the distance between two sets. Several adaptations of this measurement have been implemented: maximal Hausdorff, minimal Hausdorff, and average Hausdorff. Moreover, we propose a new metric, based on previous ones, which we call the *adapted Hausdorff*. These metrics are used to design four different versions of ReliefF-MI where each version implements a different metric.

Next, the metric and differential function in each version are shown. For all cases we consider the following concrete data:

- R_i is the bag selected in the current iteration which contains three instances (R_i^1, R_i^2, R_i^3),
- H_j is the j^{th} bag of the k nearest hit selected in the current iteration which contains four instances, $(H_j^1, H_j^2, H_j^3, H_j^4)$ and
- M_j is the j^{th} bag of the k nearest misses selected in the current iteration which contains six instances, $(M_j^1, M_j^2, M_j^3, M_j^4, M_j^5, M_j^6)$.

2.2.1. ReliefF-MI with maximal Hausdorff distance

This extension of ReliefF for MIL uses the *maximal Hausdorff distance* (28). This distance is the classical Hausdorff distance:

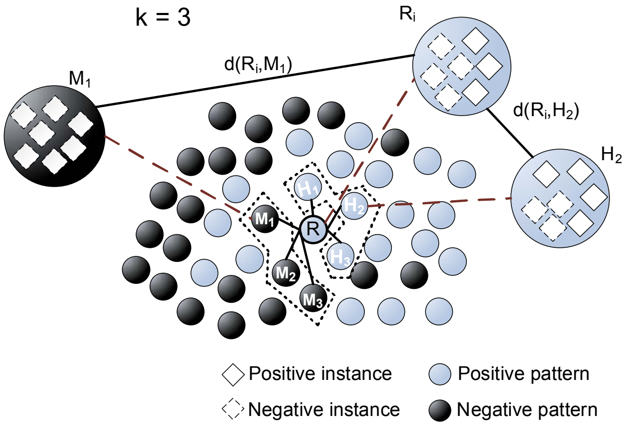
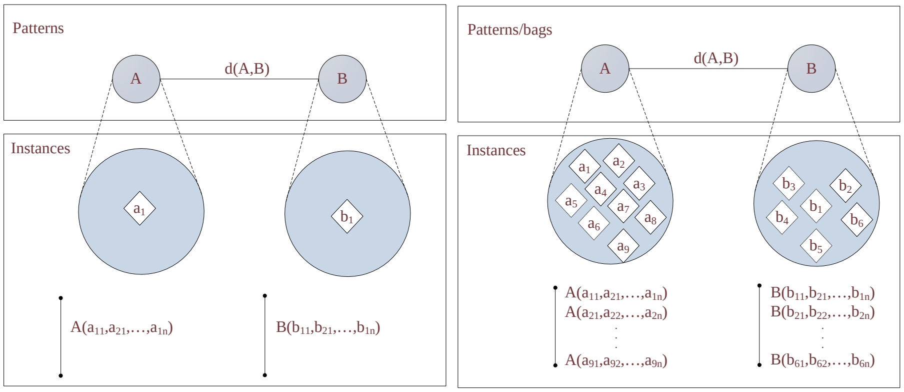


Figure 1: A step in ReliefF-MI algorithm



(a) Distance between patterns in single instance

(b) Distance between patterns in multiple instance

Figure 2: Calculating distances in single and multiple instance

$$H_{max}(R_i, H_j) = max\{h_{max}(R_i, H_j), h_{max}(H_j, R_i)\} \text{ where}$$

$$h_{max}(R_i, H_j) = max_{r \in R_i} min_{h \in H_j} ||r - h||$$

To calculate the difference between the attributes of two patterns, the instance of each bag selected to calculate the distance is that which maximizes the minimal distance between the different instances of one bag and another:

$$diff_{bag-max}(A, R_i, H_j) = diff_{instance}(A, R_i^3, H_j^4)$$

where R_i^3 and H_j^4 are instances which satisfy this condition. Similarly, $diff_{bag-max}(A, R_i, M_j)$ is computed but with the instances of R_i and M_j.

2.2.2. *ReliefF-MI with minimal Hausdorff distance*

This extension of ReliefF for MIL uses the *minimal Hausdorff distance* (25). Instead of choosing the maximum, first the distance is ranked and the lowest distance value is selected. Formally, it modifies the Hausdorff distance definition as follows:

$$H_{min}(A, B) = min_{a \in A} min_{b \in B} ||a - b||$$

To calculate the difference between the attributes of two patterns, the instance of each bag selected to calculate the distances is that which minimizes the distance between instances of each bag:

$$diff_{bag-min}(A, R_i, H_j) = diff_{instance}(A, R_i^1, H_j^3)$$

R_i^1 and H_j^3 being instances that satisfy this condition. Similarly, $diff_{bag-min}$ (A, R_i, M_j) would be computed with the instances of R_i and M_j.

2.2.3. ReliefF-MI with average Hausdorff distance

This extension of ReliefF for MIL uses the *average Hausdorff distance* (27) proposed by Zhang and Zhou to measure the distance between two bags. It is defined as follows:

$$H_{avg}(R_i, H_j) = (\sum_{r \in R_i} min_{h \in H_j} ||r - h|| + \sum_{h \in H_j} min_{r \in R_i} ||h - r||)/(|R_i| + |H_j|)$$

where |.| denotes the cardinality of a set. $H_{avg}(A, B)$ averages the distances between each instance in one bag and its nearest instance in the other bag. Conceptually speaking, the average Hausdorff distance takes into consideration more geometric relationships between two bags of instances than do the maximal and minimal Hausdorff.

To calculate the difference between the attributes of two patterns, there are several instances involved in updating the weights of the features. Supposing

- $d(R_i^1, H_j^2)$, $d(R_i^2, H_j^1)$ and $d(R_i^3, H_j^4)$ are the minimal distances between each instance $r \in R_i$ with respect to instances $h \in H_j$.
- $d(H_j^1, R_i^1)$, $d(H_j^2, R_i^1)$, $d(H_j^3, R_i^2)$ and $d(H_j^4, R_i^3)$, are the minimal distances between each instance $h \in H_j$ with respect to the instances $r \in R_i$.

The function $diff$ would be:

$$diff_{bag-avg}(A, R_i, H_j) = \frac{1}{r+h} * [diff_{instance}(A, R_i^1, H_j^2) + diff_{instance}(A, R_i^2, h_j^1) + diff_{instance}(A, R_i^3, h_j^4) + \\ diff_{instance}(A, H_j^1, R_i^1) + diff_{instance}(A, H_j^2, R_i^1) + diff_{instance}(A, H_j^3, R_i^2) + diff_{instance}(A, H_j^4, R_i^3)]$$

The same process is used to calculate $diff_{bag-avg}(A, R_i, M_j)$, but considering the pattern M_j.

2.2.4. ReliefF-MI with adapted Hausdorff distance

This extension of ReliefF for MIL uses the adapted Hausdorff distance proposed. Due to the particularities of this learning, this new proposed distance combines the previous ones and represents a different calculation depending on the class of the pattern. In this learning, the information about instances in each pattern depends on the class that it belongs to. Thus, the metric is different when evaluating the distance between two positive or negative patterns rather than that between one positive and one negative pattern.

- If both patterns are negative, we can be sure that there is no instance in the pattern that represents the concept we want to learn. Therefore, an average distance will be used to measure the distance between these bags because all instances are guaranteed to be negative: $H_{adapted}(R_i, H_j) = H_{avg}(R_i, H_j)$.
- If both patterns are positive. The correct information is that at least one instance in each of them represents the concept that we want to learn, but there is no information about which particular instance or set represents the concept. Therefore, the minimal distance is used to measure their distance because the positive instance has more probability of being near: $H_{adapted}(R_i, H_j) = H_{min}(R_i, H_j)$
- Finally, if we evaluate the distance between patterns where one of them is a positive bag and the other is a negative one, the measurement considered will be the maximal Hausdorff distance because the instances in the different classes are probably outliers between the two patterns: $H_{adapted}(R_i, M_j) = H_{max}(R_i, M_j)$

In this case, the calculation of the *diff* function also depends on the pattern class. Therefore, the pattern label will determine how the function *diff* will be evaluated.

- If R_i is positive and H_j is positive,

$$diff_{bag-adapted}(A, R_i, H_j) = diff_{bag-min}(A, R_i, H_j)$$

- If R_i is negative and H_j is negative,

$$diff_{bag-adapted}(A, R_i, H_j) = diff_{bag-avg}(A, R_i, H_j)$$

- Finally, if R_i is positive and M_j is negative or viceversa,

$$diff_{bag-adapted}(A, R_i, M_j) = diff_{bag-max}(A, R_i, M_j)$$

Finally, the function $diff_{instance}$ used in the previous calculations for all ReliefF-MI extensions is the difference between two particular instances for a given attribute and the total distance is the sum of distances throughout all attributes (Manhattan distance, (29)). When dealing with nominal attributes, the function $diff_{instance}(A, I_x, I_y)$ is defined as

$$diff_{instance}(A, I_x, I_y) = \begin{cases} 0; & \text{value(A, } I_x\text{) = value(A, } I_y\text{)} \\ 1; & \text{otherwise} \end{cases}$$

and for numerical attributes by:

$$diff_{instance}(A, I_x, I_y) = \frac{|value(A, I_x) - value(A, I_y)|}{max(A) - min(A)}$$

I_x being an instance x that belongs to the data set while I_y is another instance in the data set. Moreover, it is also used to calculate the distance between instances to find their nearest neighbours.

3. Empirical Study

This section considers, firstly, a description of application domains, algorithms and other settings used in this study. Then, the experimental results are discussed to check if the method designed, Relief-MI, is relevant to MIL algorithms.

3.1. Experimental Setting

With respect to the applications, three data sets used in our experiments are classified as content-based image classification while two are classified as drug activity prediction. These problems have been widely used in MIL (1; 2; 4; 30) due to the advantage of using multi-instance representation. On the one hand, the image classification data sets represent three categories of animals: elephants, foxes and tigers. Each data set consists of 100 images which contain the specific animal and another 100 images which contain different animals using 230 features. The final goal consists of distinguishing images of the animal in question from others that do not contain it. On the other hand, the drug activity prediction datasets study a property of molecules.

The first data set considered, Musk1, contains 92 molecules where 47 of them present the property and 45 do not. With respect to the second data set, Musk2, there are 102 molecules where 39 present the property and 63 do not (an imbalanced data set).

With respect to the algorithms, this study has considered some of the most representative paradigms used in MIL to date. The main paradigms considered are methods based on diverse density considering MDD (6), MIDD (31) and MIEMDD (32); logistic regression algorithms considering MILR (33); distance-based approaches considering CitationKNN (34) and MIOptimalBall (35); decision trees considering DecisionStump (33) and RepTree (33); rule-based methods considering different classic algorithms of traditional supervised learning adapted to MIL. There are two possible ways to adapt them. The first is MIWrapper (33), a method that assigns the class label of a bag to all its instances and then trains a single instance algorithm on the resulting data and the other, MISimple (33), which is a method that computes summary statistics for a bag to form a single instance from it. The algorithms included in this paradigm are: PART(MIWrapper), PART(MISimple) and combinations of the different proposals using rule based systems: AdaBoost&PART(MISimple), Bagging&PART(MIWrapper) and AdaBoost&PART(MIWrapper); another paradigm taken into account is that of support vector machines considering MISMO (1) and SMO (MIWrapper) (36; 37); and finally probabilistic methods considering the Naive Bayes (MIWrapper) algorithm (33). More in-depth information for each method can be consulted on the WEKA workbench (33).

With respect to the configuration parameters of ReliefF-MI, the values of k and m vary according to the data set. Thus, for image categorization data sets, the values are *k = 80* and *m = 180*. In the case of drug activity prediction, they are *k = 35* and *m = 82* for musk1 data set and *k = 29* and *m = 90* for musk2 data set. The threshold used by ReliefF-MI to select the final feature number, independently of the metric used, is the 23 most relevant features for image categorization and the 116 most relevant features for the drug activity prediction problem. These configuration parameters have been set by means of a study testing different values and selecting the most appropriate.

The validation of the final performance of the model is obtained using the 10-fold cross validation method to evaluate the generalization performance of the classifiers trained using a combination of selected features. Each fold contains roughly the same proportion of different classes while the validation method is adapted to MIL to conserve the bag structure composed of different instances. Reductions in the data set and the partitions by means of the 10-fold cross validation used in this work will be available to facilitate future comparison to new methods. [1]

3.2. Experimental Results

In this section, experimental results with different algorithms, data sets and numbers of features are compared to show the efficiency of ReliefF-MI for dimensionality reduction in MIL. This evaluation has been broken down into two parts: a comparative study between the results obtained by the different designed versions and a comparison between algorithm performance that does or does not use dimensionality reduction to show the relevance of this filter method in the MIL framework.

3.2.1. Comparison between the different metrics of ReliefF

A study of the different versions developed of ReliefF-MI will be discussed in this section. The different methods modify the relationships considered between the instances which belong

[1] http://www.uco.es/grupos/kdis/mil/fs

to each bag. In this inductive learning, the information about the class label of each instance was unknown. Thus, each version allows us to calculate the distance between bags in a different way, setting different relationships between the instances in a bag. This study is aimed at discovering the best possible option.

To carry out an evaluation of these different proposals, the results obtained by the different algorithms are compared using the feature sets provided by each technique. The comparison carried out allows us to know which metric generally achieves better accuracy for the algorithms in multi-instance classification tasks. In the experimentation, seventeen of the most popular proposals in MIL are considered with five applications. The average results, reported in Table 1 for accuracy in tiger, fox, elephant, musk1 and musk2 data sets, are obtained by the algorithms using different proposals for ReliefF-MI.

It is complicated to draw conclusions at this point because there is too much information. General information can be considered by evaluating the average accuracy values in the three data sets of categorization of these images obtained by different versions. This information is

Table 1: Results with the different versions of ReliefF-MI

Algorithms	Maximal					Minimal				
	Eleph	Tiger	Fox	Musk1	Musk2	Eleph	Tiger	Fox	Musk1	Musk2
citationKNN	0.750	0.830	0.615	0.889	0.790	0.745	0.850	0.630	0.944	0.810
MDD	0.725	0.810	0.620	0.856	0.750	0.710	0.800	0.600	0.856	0.810
RepTree [1]	0.825	0.870	0.655	0.833	0.830	0.840	0.845	0.665	0.856	0.830
DecisionStump [1]	0.825	0.800	0.655	0.711	0.810	0.820	0.785	0.695	0.689	0.840
MIDD	0.755	0.780	0.600	0.878	0.730	0.750	0.780	0.645	0.844	0.800
MIEMDD	0.725	0.775	0.530	0.856	0.780	0.685	0.720	0.605	0.922	0.840
MILR	0.815	0.855	0.600	0.822	0.790	0.840	0.825	0.630	0.744	0.850
MIOptimalBall	0.795	0.740	0.575	0.711	0.740	0.765	0.715	0.495	0.711	0.740
RBF Kernel[2]	0.765	0.835	0.615	0.711	0.870	0.800	0.865	0.655	0.700	0.860
Polynomial Kernel[2]	0.765	0.825	0.620	0.900	0.890	0.780	0.825	0.685	0.856	0.880
AdaBoost&PART[3]	0.830	0.840	0.615	0.867	0.860	0.830	0.825	0.745	0.867	0.860
Bagging&PART[3]	0.830	0.850	0.585	0.889	0.860	0.810	0.865	0.595	0.911	0.860
PART[3]	0.830	0.815	0.580	0.856	0.800	0.815	0.830	0.615	0.800	0.870
SMO[3]	0.705	0.815	0.660	0.844	0.730	0.715	0.835	0.675	0.844	0.790
Naive Bayes[3]	0.655	0.820	0.590	0.767	0.740	0.675	0.815	0.650	0.800	0.700
AdaBoost&PART[4]	0.800	0.855	0.570	0.833	0.870	0.840	0.830	0.600	0.856	0.840
PART[4]	0.770	0.795	0.620	0.722	0.830	0.765	0.730	0.670	0.811	0.770
Algorithms	Average					Adapted				
	Eleph	Tiger	Fox	Musk1	Musk2	Eleph	Tiger	Fox	Musk1	Musk2
citationKNN	0.745	0.840	0.610	0.922	0.800	0.745	0.815	0.615	0.900	0.790
MDD	0.710	0.790	0.605	0.867	0.790	0.705	0.805	0.660	0.844	0.790
RepTree [1]	0.840	0.865	0.700	0.867	0.790	0.840	0.855	0.710	0.867	0.840
DecisionStump [1]	0.820	0.800	0.660	0.722	0.800	0.830	0.805	0.700	0.722	0.850
MIDD	0.750	0.780	0.595	0.856	0.790	0.755	0.770	0.695	0.900	0.780
MIEMDD	0.685	0.745	0.530	0.900	0.810	0.715	0.770	0.615	0.900	0.780
MILR	0.840	0.840	0.615	0.744	0.820	0.835	0.875	0.635	0.756	0.830
MIOptimalBall	0.765	0.735	0.525	0.744	0.760	0.775	0.740	0.535	0.789	0.700
RBF Kernel[2]	0.800	0.830	0.655	0.733	0.870	0.785	0.855	0.650	0.811	0.860
Polynomial Kernel[2]	0.780	0.830	0.665	0.878	0.890	0.770	0.820	0.655	0.878	0.860
AdaBoost&PART[3]	0.830	0.820	0.620	0.856	0.880	0.840	0.860	0.665	0.900	0.910
Bagging&PART[3]	0.810	0.860	0.610	0.900	0.820	0.830	0.865	0.605	0.911	0.870
PART[3]	0.815	0.810	0.570	0.878	0.850	0.835	0.840	0.620	0.822	0.900
SMO[3]	0.715	0.830	0.655	0.844	0.730	0.705	0.820	0.690	0.833	0.740
Naive Bayes[3]	0.675	0.825	0.585	0.856	0.740	0.660	0.820	0.680	0.789	0.770
AdaBoost&PART[4]	0.840	0.840	0.560	0.856	0.810	0.830	0.845	0.650	0.867	0.850
PART[4]	0.765	0.740	0.660	0.778	0.800	0.775	0.780	0.665	0.767	0.810

[1] MIBoost
[3] MIWrapper
[2] MISMO
[4] MISimple

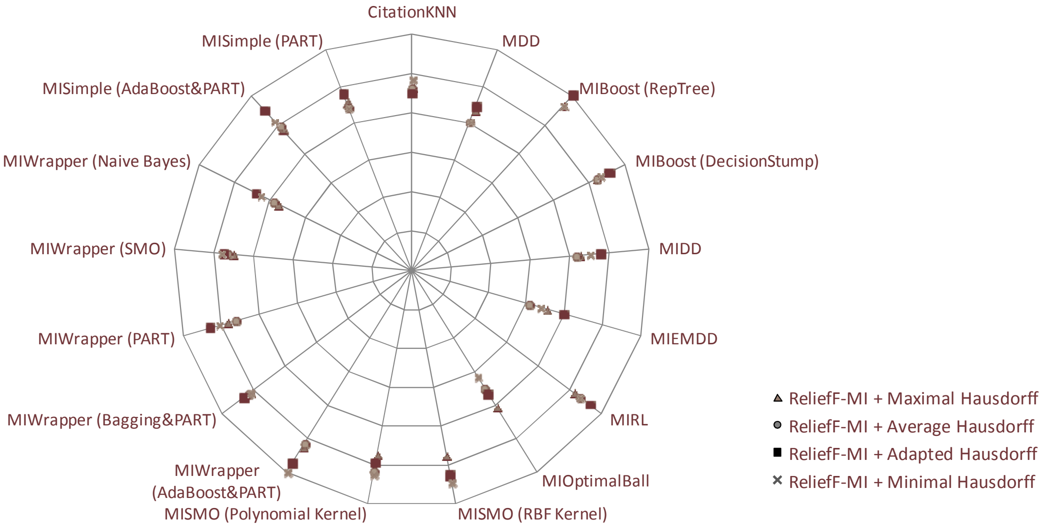


Figure 3: Comparison of different ReliefF-MI versions

shown in Fig. 3. Thus, for each algorithm, the average results are indicated using different dimensionality-reduction methods. According to this figure, the closer a value is to the extreme of the circumference, the better the accuracy obtained by this method, using that version of ReliefF-MI. We can see that more algorithms get their best accuracy results when they employ the feature set given by ReliefF-MI using the adapted Hausdorff distance. According to this data, some cases where this method is out-performed occur for CitationKNN or MISMO algorithms which obtain their best results when they use the feature set given by the minimal Hausdorff distance. Also, it is interesting to note that the algorithms usually obtain their worst results when using the dimensionality reduction given by the version of ReliefF-MI that uses maximum and average Hausdorff distance.

To determine if there really are significant differences when using one or the other of the proposals, a non parametric statistical test called the Friedman test is used. To carry out this test, the results considered are those obtained by the algorithms using different versions of ReliefF-MI and the five applications. The Friedman test (38) compares the average ranks of the proposals considered, where the lowest value of the rank means that that version out-performs the results

Table 2: Algorithm Ranking

Method Used	Accuracy Ranking
Average hausdorff distance	2.641
Maximal hausdorff distance	2.765
Minimal hausdorff distance	2.512
Adapted hausdorff distance	2.082

obtained using the dimensional reduction provided by the other algorithms. The ranking of each proposal is shown in Table 2. A first glance reveals that the adapted Hausdorff distance obtains the lowest value. However, if we want to obtain a statistically valid conclusion, the Friedman test results have to be evaluated (Table 3 shows these results). This test determines (at a confidence level of 90%) that there are significant differences in the results when different metrics are used. As its value is 13.493 and the $\chi^2(n = 3)$ is 6.251, therefore the null hypothesis is rejected and it is obvious that there are differences between them.

The Bonferroni test (39) is carried out to determine which metric selects the most relevant features and therefore optimizes the accuracy of most algorithms. The results of this test determine a *Threshold* and those methods with a ranking value that exceeds this value are considered proposals with significantly worse results than the control proposal (that is, the method with the lowest ranking value). According to this specification, this test reveals that the methods with a threshold over 2.504 (confidence at 90%) are considered worse proposals than the control algorithm. In this case, the control algorithm is ReliefF-MI with the adapted hausdorff metric because it gets the lowest ranking value and therefore is the best option. Statistically, the rest of the proposals are worse than the adapted hausdorff because they have a higher ranking than the threshold set by this test. Thus, the use of the new metric is shown to obtain a better reduction in dimensionality. Its advantage lies in using different ways of measuring the distance depending on the specific information available in each bag.

3.2.2. *The Effectiveness of ReliefF-MI*

In this section, the aim is to study the effectiveness of ReliefF-MI. Thus, in this case, the study is focused on the use of the version of ReliefF-MI with adapted Hausdorff distance because it is the option which achieves the best results.

The results obtained by the algorithms when they use the reduced data set provided by ReliefF-MI using adapted Hausdorff distance are compared to results using the full data set. Table 4 shows the results of the seventeen algorithms for different data sets in two cases. The results shown are the average results per test set of the 10-fold operation carried out in the cross-validation process. These results show that the algorithms differ in the amount of emphasis they place on feature selection. At first glance, the accuracy with different algorithms is always outperformed when the algorithms use the feature set provided by RelelieffF-MI. However, the improvement yielded for different algorithms is not the same. Thus, at one extreme are algorithms such as the simple nearest-neighbour learner, which classifies novel examples by retrieving the nearest stored training example, using all the available features in its distance computations (such as CitationKNN). Towards the other extreme lie algorithms that explicitly try to focus on relevant features and ignore irrelevant ones. Decision tree inducers are examples of this approach (such as, the MIBoost algorithms with repetition tree and decision stump). Nevertheless, in both cases feature selection prior to learning is beneficial. Reducing the dimensionality of the data reduces

Table 3: Friedman Tests (comparison between metrics of ReliefF-MI)

Friedman Test		
χ^2 ($\alpha = 0.90$)	Value Test	Conclusion
6.251	13.493	Reject null hypothesis

Table 4: Results with Full Feature Data Set

ALGORITHMS	Reduced Set					Full Set				
	Eleph	Tiger	Fox	Musk1	Musk2	Eleph	Tiger	Fox	Musk1	Musk2
citationKNN	0.745	0.815	0.615	0.9000	0.7900	0.500	0.500	0.500	0.944	0.850
MDD	0.705	0.805	0.660	0.8444	0.7900	0.800	0.755	0.700	0.789	0.760
MIBoost(RepTree)	0.840	0.855	0.710	0.8667	0.8400	0.815	0.825	0.670	0.867	0.840
MIBoost (DecisionStump)	0.830	0.805	0.700	0.7222	0.8500	0.815	0.780	0.650	0.678	0.830
MIDD	0.755	0.770	0.695	0.9000	0.7800	0.825	0.740	0.655	0.922	0.730
MIEMDD	0.715	0.770	0.615	0.9000	0.7800	0.730	0.745	0.600	0.889	0.900
MILR	0.835	0.875	0.635	0.7556	0.8300	0.780	0.840	0.510	0.733	0.840
MIOptimalBall	0.775	0.740	0.535	0.7889	0.7000	0.730	0.625	0.530	0.767	0.780
MISMO (RBF Kernel)	0.785	0.855	0.650	0.8111	0.8600	0.800	0.795	0.590	0.744	0.840
MISMO (Polynomial Kernel)	0.770	0.820	0.655	0.8778	0.8600	0.790	0.785	0.580	0.878	0.840
MIWrapper (AdaBoost&PART)	0.840	0.860	0.665	0.9000	0.9100	0.840	0.790	0.685	0.867	0.890
MIWrapper (Bagging&PART)	0.830	0.865	0.605	0.9111	0.8700	0.845	0.810	0.600	0.900	0.870
MIWrapper (PART)	0.835	0.840	0.620	0.8222	0.9000	0.790	0.780	0.550	0.800	0.820
MIWrapper (SMO)	0.705	0.820	0.690	0.8333	0.7400	0.715	0.800	0.635	0.844	0.740
MIWrapper (Naive Bayes)	0.660	0.820	0.680	0.7889	0.7700	0.680	0.760	0.590	0.789	0.760
MISimple(AdaBoost&PART)	0.830	0.845	0.650	0.8667	0.8500	0.840	0.795	0.625	0.756	0.820
MISimple (PART)	0.775	0.780	0.665	0.7667	0.8100	0.765	0.765	0.635	0.744	0.800
Number of Features	23	23	23	116	116	230	230	230	166	166

the size of the hypothesis space and allows algorithms to operate faster and more effectively.

A statistical study is carried out to check whether the use of ReliefF-MI improves the results of algorithms when no dimensionality reduction is done. The Wilcoxon rank sum test is used to find whether there are differences between the accuracy values obtained by different algorithms using the feature set provided by ReliefF-MI. This test is a non-parametric test recommended in Demsar's study (38). The procedure is based on calculating the average ranges of each sample and then computing the number of times that the ranks are better than the other samples. The null hypothesis of this test maintains that there are no significant differences between the accuracy values obtained by algorithms using different feature sets, while the alternative hypothesis ensures that there are. Table 5 shows the mean ranks and the sum of ranks for each of the two options. The scores are ranked from lowest to highest. Therefore, we can see that algorithms not using feature selection have a lower mean rank than algorithms using the ReliefF-MI method. This information can be used to ascertain a priori that ReliefF-MI is a better proposal.

The results of the Wilcoxon statistical test are 6725.5 and the corresponding z-score is -1.690. We reject the null hypothesis, then, at a 90% confidence level (p-value = 0.091 < 0.1). Consequently, ReliefF-MI yields significantly higher accuracy values than the option that does not use feature reduction. Note that, for example, ReliefF-MI scores, with a mean rank 91.88, are higher in the algorithms using selection feature as a pre-processing step than those of the other

Table 5: Sum of Ranks and Mean Rank of the two proposals

	Mean Rank	Sum of Ranks
ReliefF-MI Method (Adapted Hausdorff Distance)	91.88	7809.50
Not Reducting features	79.12	6725.50

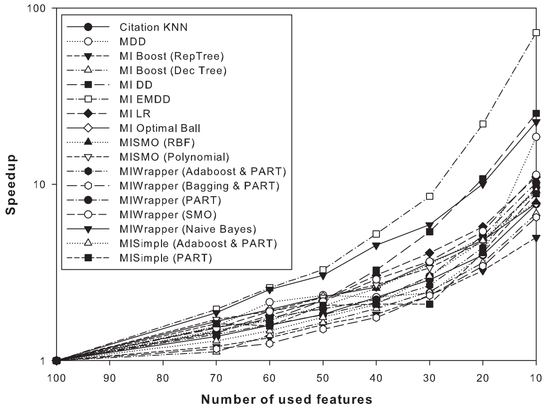


Figure 4: Computation time according to the number of features used

option (with a mean rank of 79.12).

In general, we can conclude that the use of this method improves the results achieved by the algorithms. This study considers seventeen different algorithms and five different data sets, so this is representative enough to ascertain that a lower number of features minimize the classification error in the framework of MIL.

Finally, another advantage of using a reduced number of features is that the computational time decreases significantly. Fig. 4 shows the improvement in algorithm efficiency with respect to different feature-number uses. Concretely, what is shown is the acceleration experienced when the feature number decreases. The acceleration (speedup) is represented in the logarithmic scale in order to better understand the results. That can be seen, independently of the algorithm. A reduction in feature number has a positive affects on the efficiency of the algorithms.

4. Conclusion

Feature selection methods are methods that reduce the number of characteristics. These methods eliminate those features that are not relevant for solving the problem, making the algorithms improve their accuracy and computation time. Dimensionality reduction has shown great efficiency in the results achieved in traditional supervised learning. However, the attention given to the study of these techniques in MIL has been negligible. This paper, designs a filter method of feature selection called ReliefF-MI to deal with multiple-instance problems. This is an efficient method that acts as a pre-processing step that is completely independent of the choice of a particular multi-instance algorithm: it can be applied to continuous and discrete problems, and is faster than wrapper methods.

The experimental results show the effectiveness of our approach using five different applications and seventeen algorithms with reduced data. First, four versions of ReliefF-MI were designed using different metrics for calculating the closeness of the patterns. A comparison determined that, statistically, the best performance is achieved by the new metric proposed, called adapted Hausdorff distance. This metric is designed to adapt the variable information in each pattern according to the class of that pattern. Then, the evaluation was carried out by comparing the results obtained by different algorithms when they use ReliefF-MI and when, instead, they work with full data sets. In this case, the Wilcoxon test shows the benefits of applying data reduction in MIL. This test determined that there are significant differences between the results obtained by the algorithms when they use feature selection compared to when they do not. The results confirm that algorithms obtain better results when they work with only the most relevant features. Thus, the relevance of using feature selection in this scenario is established for improving the performance of algorithms with high-dimensional data.

These results are significant, but much more research can be done in this area, such as designing other metrics to measure the distance between bags which would optimize performance, as well as designing other feature selection methods based on filtering in the MIL scenario to study which methods work best in this learning context.

Acknowledgment

The authors gratefully acknowledge the subsidy provided by the Spanish Department of Research under TIN2008- 06681-C06-03, P08-TIC-3720 Projects and FEDER funds.

References

[1] S. Andrews, I. Tsochantaridis, T. Hofmann, Support vector machines for multiple-instance learning, in: NIPS'02: Proceedings of Neural Information Processing System, Vancouver, Canada, 2002, pp. 561–568.

[2] A. Zafra, S. Ventura, G3P-MI: A genetic programming algorithm for multiple instance learning, Information Sciences 180 (23) (2010) 4496–4513.

[3] G. Herman, G. Ye, J. Xu, B. Zhang, Region-based image categorization with reduced feature set, in: Proceedings of the 10th IEEE Workshop on Multimedia Signal Processing, Cairns, Qld, Australia, 2008, pp. 586–591.

[4] C. Yang, M. Dong, F. Fotouhi, Region based image annotation through multiple-instance learning, in: Multimedia'05: Proceedings of the 13th Annual ACM International Conference on Multimedia, New York, USA, 2005, pp. 435–438.

[5] X. Qi, Y. Han, Incorporating multiple svms for automatic image annotation, Pattern Recognition 40 (2) (2007) 728–741.

[6] O. Maron, T. Lozano-Pérez, A framework for multiple-instance learning, in: NIPS'97: Proceedings of Neural Information Processing System 10, Denver, CO, USA, 1997, pp. 570–576.

[7] Z.-H. Zhou, K. Jiang, M. Li, Multi-instance learning based web mining, Applied Intelligence 22 (2) (2005) 135–147.

[8] A. Zafra, S. Ventura, C. Romero, E. Herrera-Viedma, Multi-instance genetic programming for web index recommendation, Expert System with Applications 36 (2009) 11470–11479.

[9] X. Chen, C. Zhang, S. . Chen, S. Rubin, A human-centered multiple instance learning framework for semantic video retrieval, IEEE Transactions on Systems, Man and Cybernetics Part C, 39 (2) (2009) 228–233.

[10] Z. Gu, T. Mei, J. Tang, X. Wu, X. Hua, MILC2: A multi-layer multi-instance learning approach to video concept detection, in: MMM'08: Proceedings of the 14th International Conference of Multimedia Modeling, Kyoto, Japan, 2008, pp. 24–34.

[11] S. Gao, Q. Suna, Exploiting generalized discriminative multiple instance learning for multimedia semantic concept detection, Pattern Recognition 41 (10) (2008) 3214–3223.

[12] J. Pang, Q. Huang, S. Jiang, Multiple instance boost using graph embedding based decision stump for pedestrian detection, in: ECCV'08: Proceedings of the 10th European Conference on Computer Vision, LNCS 5305, Springer-Verlag, Berlin, 2008, pp. 541–552.

[13] A. Zafra, C. Romero, S. Ventura, Multiple instance learning for classifying students in learning management systems, Expert Systems with Applications. Accepted.

[14] R. Bellman, Adaptive control processes: a guided tour, Rand Corporation Research studies, Princeton University Press, 1961.

[15] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.

[16] M. Hall, Correlation-based feature selection for discrete and numeric class machine learning, in: ICML'00: Proceedings of the 17th International Conference on Machine Learning, 2000, pp. 359–366.

[17] H. Liu, R. Setiono, A probabilistic approach to feature selection - a filter solution, in: ICML'00: Proceedings of the 13th International Conference on Machine Learning, 1996, pp. 319–327.

[18] L. Yu, H. Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, in: ICML'00: Proceedings of the 20th International Conference on Machine Learning, 2003, pp. 856–863.

[19] A. Abraham, E. Corchado, J. Corchado, Hybrid learning machines, Neurocomputing 72 (2009) 2729–2730.

[20] E. Corchado, A. Abraham, A. P. Leon, Hybrid intelligent algorithms and applications, Information Science 180 (2010) 2633–2634.

[21] M.-L. Zhang, Z.-H. Zhou, Improve multi-instance neural networks through feature selection, Neural Processing Letter 19 (1) (2004) 1–10.

[22] X. Yuan, X.-S. Hua, M. Wang, G.-J. Qi, X.-Q. Wu, A novel multiple instance learning approach for image retrieval based on adaboost feature selection, in: ICME'07: Proceedings of the IEEE International Conference on Multimedia and Expo, IEEE, Beijing, China, 2007, pp. 1491–1494.

[23] V. C. Raykar, B. Krishnapuram, J. Bi, M. Dundar, R. B. Rao, Bayesian multiple instance learning: automatic feature selection and inductive transfer, in: ICML '08: Proceedings of the 25th international conference on Machine learning, ACM, New York, 2008, pp. 808–815.

[24] I. Kononenko, Estimating attributes: analysis and extension of relief, in: ECML'94: Proceedings of the 7th European Conference in Machine Learning, Springer-Verlag, Berlin, 1994, pp. 171–182.

[25] Y.-Z. Chevaleyre, J.-D. Zucker, Solving multiple-instance and multiple-part learning problems with decision trees and decision rules. Application to the mutagenesis problem, in: AI'01: Proceedings of the 14th of the Canadian Society for Computational Studies of Intelligence, LNCS 2056, Ottawa, Canada, 2001, pp. 204–214.

[26] D. Zhang, F. Wang, L. Si, T. Li, M3IC: Maximum margin multiple instance clustering, 2009, pp. 1339–1344.

[27] M.-L. Zhang, Z.-H. Zhou, Multi-instance clustering with applications to multi-instance prediction, Applied Intelligence 31 (2009) 47–68.

[28] G. Edgar, Measure, topology, and fractal geometry. Third Edition, Springer-Verlag, Berlin, 1995.

[29] H. Cohen, Image restoration via n-nearest neighbour classification, in: ICIP'96: Proceedings of the International Conference on Image Processing, 1996, pp. 1005–1007.

[30] T. G. Dietterich, R. H. Lathrop, T. Lozano-Perez, Solving the multiple instance problem with axis-parallel rectangles, Artifical Intelligence 89 (1-2) (1997) 31–71.

[31] X. Xu, Statistical learning in multiple instance problems, Ph.D. thesis, Department of Computer Science. University of Waikato (2003).

[32] Q. Zhang, S. Goldman, EM-DD: An improved multiple-instance learning technique, in: NIPS'01: Proceedings of Neural Information Processing System 14, Vancouver, Canada, 2001, pp. 1073–1080.

[33] I. H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques. Second Edition, Morgan Kaufmann, 2005.

[34] J. Wang, J.-D. Zucker, Solving the multiple-instance problem: A lazy learning approach, in: ICML'00: Proceedings of the 17th International Conference on Machine Learning, Stanford, CA, USA, 2000, pp. 1119–1126.

[35] P. Auer, R. Ortner, A boosting approach to multiple instance learning, in: ECML'04: Proceedings of the 5th European Conference on Machine Learning, LNCS 3201, Pisa, Italy, 2004, pp. 63–74.

[36] S. Keerthi, S. Shevade, C. Bhattacharyya, K. Murthy, Improvements to Platt's SMO algorithm for svm classifier design, Neural Computation 13 (3) (2001) 637–649.

[37] T. Gärtner, P. A. Flach, A. Kowalczyk, A. J. Smola, Multi-instance kernels, in: ICML'02: Proceedings of the 19th International Conference on Machine Learning, Morgan Kaufmann, Sydney, Australia, 2002, pp. 179–186.

[38] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[39] O. J. Dunn, Multiple comparisons among means, Journal of the American Statistical Association 56 (293) (1961) 52–64.