Fuzzy Multi-Instance Classifiers

Sarah Vluymans, Dánel Sánchez Tarragó, Yvan Saeys, Chris Cornelis, and Francisco Herrera, *Senior Member, IEEE*

***Abstract*—Multi-instance learning is a setting in supervised learning where the data consist of bags of instances. Samples in the dataset are groups of individual instances. In classification problems, a decision value is assigned to the entire bag, and the classification of an unseen bag involves the prediction of the decision value based on the instances it contains. In this paper, we develop a framework for multi-instance classifiers based on fuzzy set theory. Fuzzy sets have been used in many machine learning applications, but so far not in the classification of multi-instance data. We explore its untapped potential here. We interpret the classes as fuzzy sets and determine membership degrees of unseen bags to these sets based on the available training data. In doing so, we develop a framework of classifiers that extract the required membership degrees either at the level of instances (instance-based) or at the level of bags (bag-based). We offer an extensive analysis of the different settings within the proposed framework. We experimentally compare our proposal to state-of-the-art multi-instance classifiers, and based on two evaluation measures, our methods are shown to perform very well.**

***Index Terms*—Fuzzy classifiers, fuzzy set theory, multi-instance classification (MIC).**

I. Introduction

In traditional single-instance classification, elements of a dataset can be described by vectors containing their attribute values and associated decision value. In multi-instance classification (MIC) [1] on the other hand, data samples correspond to *bags* of instances. Every bag contains a number of traditional instances, and this number can vary between bags. Intuitively, these instances correspond to alternative representations of the same object or concept. They remain described by a number of attributes, but have no associated decision value. Instead, decision values are assigned to bags. The complex nature of such datasets requires custom learning algorithms, and recently, some useful observations were made about which types of problems benefit from a multi-instance approach rather than from single-instance approaches [2]. In MIC, the aim is to assign a class label to an unlabeled bag, based on the instances it contains and/or its similarity with the training bags. Several classification schemes can be set up, depending on whether the discerning information between bags is found at a local (instances) or global (bags) level. Many classifiers have been proposed in the literature, implementing different approaches to optimally predict the decision value of new bags. A review on MIC can be found in, e.g., [3].

Fuzzy set theory [4] has been used successfully in many single-instance machine learning applications, such as classification (e.g., [5]–[7]), regression (e.g., [8]–[10]), clustering (e.g., [11]–[13]), and so on. For classification data, it conveniently allows us to model a degree of membership of elements to the decision classes and to express a degree of similarity between pairs of instances.

In this paper, we propose a new set of classifiers for multi-instance problems using fuzzy set theory. To the best of our knowledge, this paradigm has not been used in MIC so far. Its unexplored use in this domain and previous successes in others motivates us to analyze its benefits for MIC in this work. We propose a framework consisting of two families.

1) *Instance-based fuzzy multi-instance classifiers (IFMIC family):* To classify an unlabeled bag, the classifiers determine a membership degree of that bag to each class and assign it to the class for which this value is largest. The membership estimation is based on instance-level information: The classifier first calculates membership degrees of instances to the classes and aggregates these values to derive the corresponding value for the bag.
2) *Bag-based fuzzy multi-instance classifiers (BFMIC family):* Like the IFMIC family, the label of an unseen bag is predicted by determining its class membership degrees. However, for BFMIC methods, these values are not based on instance-wise calculations, but are directly derived at the level of bags, using global rather than local information.

Within the two families, members differ from each other by their internal setup, for instance, by using different ways to aggregate instance-based class predictions to the level of bags. In our experimental study, we evaluate many classifiers in order to compare the different settings within our framework. We further compare our proposed methods to state-of-the-art MIC methods.

Manuscript received April 21, 2015; revised October 7, 2015 and November 26, 2015; accepted December 5, 2015. Date of publication January 11, 2016; date of current version December 22, 2016. This work was supported in part by the Spanish Ministry of Economy and Competitiveness under Project TIN2014-57251-P and the Andalusian Research Plans P11-TIC-7765 and P10-TIC-6858, and by Project PYR-2014-8 of the Genil Program of CEI BioTic GRANADA. The work of S. Vluymans was supported by the Special Research Fund (BOF) of Ghent University.

S. Vluymans is with the Department of Applied Mathematics, Computer Science and Statistics, Ghent University, 9000 Gent, Belgium, and also with the VIB Inflammation Research Center, 9052 Zwijnaarde, Belgium (e-mail: sarah.vluymans@irc.vib-ugent.be).

D. Sánchez Tarragó is with the Department of Computer Science, Central University "Marta Abreu" of Las Villas, 54830 Santa Clara, Cuba (e-mail: danels@uclv.edu.cu).

Y. Saeys is with the VIB Inflammation Research Center, 9052 Zwijnaarde, Belgium, and also with the Department of Respiratory Medicine, Ghent University, 9000 Gent, Belgium (e-mail: yvan.saeys@irc.vib-ugent.be).

C. Cornelis is with the Department of Applied Mathematics, Computer Science and Statistics, Ghent University, 9000 Gent, Belgium, and also with the Department of Computer Science and Artificial Intelligence, Research Center on Information and Communications Technology (CITIC-UGR), University of Granada, 18071 Granada, Spain (e-mail: chris.cornelis@UGent.be).

F. Herrera is with the Department of Computer Science and Artificial Intelligence, Research Center on Information and Communications Technology (CITIC-UGR), University of Granada, 18071 Granada, Spain, and also with the Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia (e-mail: herrera@decsai.ugr.es).

Digital Object Identifier 10.1109/TFUZZ.2016.2516582

The remainder of this paper is organized as follows. In Section II, we provide the necessary background on MIC. Section III introduces our classifier framework, describing the two proposed families and their members in detail. The experimental study is conducted in Section IV. Our conclusion and future research directions are outlined in Section V.

II. Preliminaries and Related Work

In this section, we formally describe MIC (see Section II-A) and recall ordered weighted average (OWA) aggregation (see Section II-B), which is used further on in this work.

A. Multi-Instance Classification

Multi-instance problems were introduced in [1] and have received a considerable amount of attention over the past two decades. Example applications include the drug activity problem [1], the web index recommendation problem [14], [15], image recognition [16], and performance prediction [17]. We consider a multi-instance training dataset $T = \{(X_1, l_1), (X_2, l_2), \ldots, (X_N, l_N)\}$, where $X_i \in 2^{\mathcal{X}}$ ($i = 1, \ldots N$) are labeled bags of nonlabeled instances, and $\mathcal{X}$ is the instance universe, which corresponds to the feature set describing the individual instances. The value l_i is the class label assigned to bag X_i and is assumed to be drawn from a finite set L. The task of a multi-instance classifier is to predict the label l of an unlabeled bag, that is, the classifier $\mathcal{F} : 2^{\mathcal{X}} \to L$ is a map from the powerset of $\mathcal{X}$ to L.

In [3], a taxonomy for MIC algorithms was proposed. The classifiers are divided into three families, based on whether they rely on instance-level information (instance space paradigm) or bag-level information to classify new bags. The latter is further divided into two groups: the bag space paradigm and the embedded space paradigm. A method embodying the instance space paradigm constructs its classification model to optimally separate instances belonging to bags from different classes. The bag-level decision is directly derived from the instance predictions. Examples are the Axis-Parallel Rectangle [1], Diverse Density [18], miSVM [19], and MIWrapper [20] methods. In the bag space paradigm, a classifier is trained to discern bags from different classes. Predictions for new bags are made using the global bag-wise information instead of relying on instance-based decisions. This paradigm includes CitationKNN [21] and MI-Graph [22]. Finally, the embedded space paradigm maps every bag to a single feature vector and trains a traditional single-instance classifier in this new space. We encounter this setup in, e.g., the SimpleMI [23], YARDS [24], DD-SVM [25], MILES [26], and BARTMIP [27] algorithms. In this paper, we present two classifier families, one incorporating the instance space paradigm and the other the bag space paradigm. Fig. 1 displays the general difference between these two approaches. To predict the label l of a new bag X, instance-based classifiers [see Fig. 1(a)] first predict class labels for all instances $x \in X$ and aggregate these values to estimate l in a second step. Bag-based classifiers [see Fig. 1(b)], on the other hand, consider X as a whole and immediately predict l based on the similarity of X with the training bags.

B. Ordered Weighted Average Aggregation

OWA operators [28] can be used to aggregate a set of real values to a single scalar. OWA aggregation of a set of values $V = \{v_1, \ldots, v_p\}$ consists of 1) ordering them in a decreasing sequence, 2) assigning them weights according to their position in this sequence, and 3) taking their weighted average. In particular, let a weight vector $W = \langle w_1, \ldots, w_p \rangle$ be provided for which $\sum_{i=1}^{p} w_i = 1$ and $w_i \in [0, 1]$ for all $i \in \{1, \ldots, p\}$ holds. If for all $i \in \{1, \ldots, p\}$, c_i is the ith largest value in V, then the OWA_W aggregation of the values in V is given by $\text{OWA}_\text{W}(V) = \sum_{i=1}^{p}(w_i c_i)$. In the remainder of this paper, we often aggregate function values calculated over the training set. To fix notation, we set

$$\text{OWA}_\text{W} \underbrace{f(x)}_{x \in \{x_1, \ldots, x_n\}} = \text{OWA}_\text{W}(\{f(x_1), \ldots, f(x_n)\}). \quad (1)$$

As a result of the ordering of the instances, weight vectors can easily be constructed as generalizations of the traditional maximum (minimum) by not only assigning a nonzero weight to the extreme value, but also to other relatively high (low) ones. The minimum and maximum operators can themselves be modeled by the weight vectors $\langle 0, 0, \ldots, 0, 1 \rangle$ and $\langle 1, 0, \ldots, 0, 0 \rangle$, respectively. The $andness$ and $orness$ degrees of a weight vector W express to which extent the corresponding OWA aggregation is similar to taking the minimum and maximum, respectively (e.g., [29]). For a vector of length p, these values are calculated as

$$orness(W) = \frac{1}{p-1} \sum_{i=1}^{p} [(p - i) \cdot w_i]$$

and $andness(W) = 1 - orness(W)$, where we assume $p > 1$. When $orness(W) > \frac{1}{2}$, the OWA aggregation is said to soften the maximum. When $andness(W) > \frac{1}{2}$, it softens the minimum.

In this paper, we use two types of weight vectors that both represent softenings of the maximum and minimum operators. We again set p to the length of the vector. The first pair is constructed using linear increasing or decreasing weights and is defined as

$$\text{Wmin-L} = \left\langle \frac{2}{p(p+1)}, \frac{4}{p(p+1)}, \ldots, \frac{2(p-1)}{p(p+1)}, \frac{2}{p+1} \right\rangle \quad (2)$$

and

$$\text{Wmax-L} = \left\langle \frac{2}{p+1}, \frac{2(p-1)}{p(p+1)}, \ldots, \frac{2}{p(p+1)} \right\rangle. \quad (3)$$

Second, we consider the use of inverse additive weight vectors, which are defined as

$$\text{Wmin-IA} = \left\langle \frac{1}{p \sum_{i=1}^{p} \frac{1}{i}}, \frac{1}{(p-1) \sum_{i=1}^{p} \frac{1}{i}}, \ldots, \frac{1}{2 \sum_{i=1}^{p} \frac{1}{i}}, \frac{1}{\sum_{i=1}^{p} \frac{1}{i}} \right\rangle \quad (4)$$

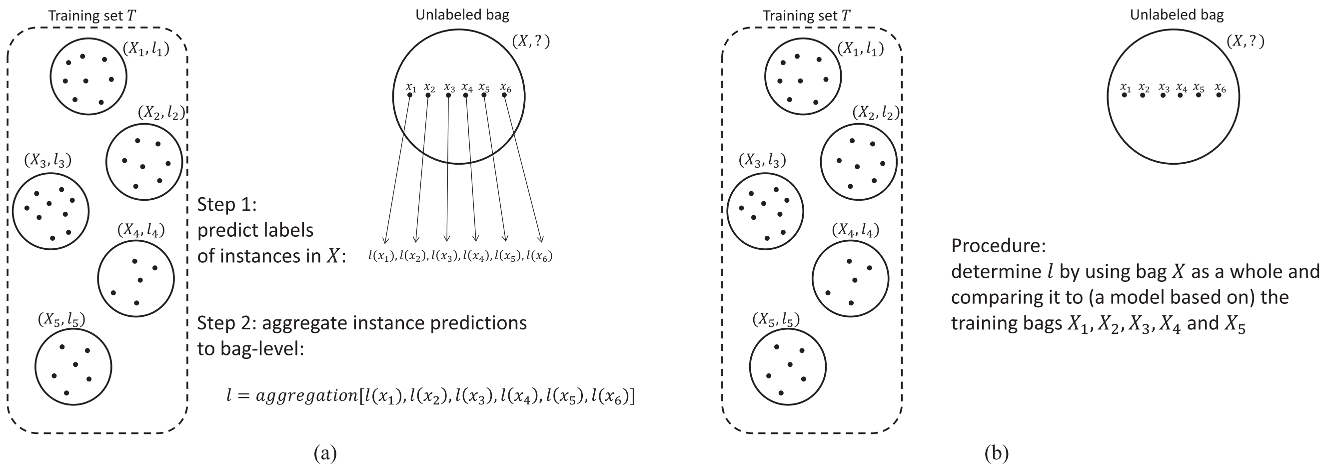

Fig. 1. (a) Instance space and (b) bag space paradigms in MIC.

and

$$\text{Wmax-IA} = \left\langle \frac{1}{\sum_{i=1}^{p}\frac{1}{i}}, \frac{1}{2\sum_{i=1}^{p}\frac{1}{i}}, \ldots, \frac{1}{p\sum_{i=1}^{p}\frac{1}{i}} \right\rangle. \tag{5}$$

The name of this second pair refers to the presence of the summation in the denominator. We can easily show that both (3) and (5) soften the maximum (see the Appendix). In Section III-A3, following the full description of our proposals, we provide a detailed motivation for why we use OWA aggregation and of our choice of weight vectors.

III. Proposed Classifiers

In our proposed fuzzy multi-instance classifiers, each class is regarded as a fuzzy set to which every bag has a degree of membership. As indicated in Section I, when classifying an unseen bag, the classifiers calculate its membership degree to each class and assign it to the class for which this value is largest. In Section III-A, we provide a wide range of possible ways to determine these values. We discuss the IFMIC family in Section III-A1 and the BFMIC family in Section III-A2. We present a compact overview of the framework in Section III-B and provide some worked examples as illustrations in Section III-C. Alternatives to all its components can be easily plugged in and evaluated as well. We end this section with a discussion of the theoretical complexity of our proposals in Section III-D.

A. Fuzzy Multi-Instance Classifiers

We define a fuzzy multi-instance classifier as

$$\mathcal{F}_{FM} : 2^{\mathcal{X}} \rightarrow L : X \mapsto \arg\max_{C \in L}[C(X)] \tag{6}$$

where $C(X)$ is the membership degree of the bag X to the class C. When the maximum value is reached for multiple classes, one is randomly selected. The crucial step when classifying a bag X is, therefore, the calculation of the values $C(X)$. To do so, we consider two approaches: 1) instance-based classifiers, where $C(X)$ relies on the class membership degree of the individuals instances in the bag; and 2) bag-based classifiers, where the membership degrees $C(X)$ are derived directly from bag information. Both cases require the definition of membership degrees to the classes, either for the training instances or training bags. These values are determined based on the training data. For the two groups of classifiers, we introduce ways to do so in the sections below.

1) Instance-Based Fuzzy Multi-Instance Classifiers (IFMIC Family): For our instance-based classifiers, the membership functions $C(X)$ in expression (6) are derived from the membership degrees of individual instances to classes. We, therefore, need to specify how to calculate the membership function $C(x)$ of an instance x to a class C. First, however, we introduce different ways to determine a membership degree $B(x)$ of an instance x to a training bag B, an important concept on which the calculation of $C(x)$ relies. We stress that this differs from merely determining whether an instance belongs to a bag or not. Interpreting B as a fuzzy set allows us to express how typical x is for bag B.

a) Membership $B(x)$ of instances to bags: The definitions of $B(x)$ themselves rely on the relation $R_I(\cdot,\cdot)$, measuring similarity between instances. We decided to use one of two similarity measures, depending on the number of features describing the instances. As a cutoff value, we use 20 features. This value is based on the experimental study of [30], which showed that when a dataset contains more than this number of features, traditional Euclidean-like distance or similarity measures are not appropriate. When at most 20 features are present, we use the Gaussian similarity, which is generally given as $R_I(x,y) = \exp(-\frac{\|x-y\|^2}{2\sigma^2})$, where $\|\cdot\|$ represents the vectorial norm. In this work, we set the Gaussian similarity to

$$R_I(x,y) = \exp\left(-\|x-y\|^2\right). \tag{7}$$

This choice was made to avoid further tuning of the parameter σ, but the user may select appropriate values for σ according to the problem at hand. In case of more than 20 features, we use the cosine similarity, defined as

$$R_I(x,y) = \frac{x \cdot y}{\|x\|\|y\|}. \tag{8}$$

TABLE I
SETTINGS FOR THE IFMIC METHODS EVALUATED WITHIN THE PROPOSED FRAMEWORK

Code	$B(x)$	Code	$C(x)$	Code	$C(X)$
Max	$\max_{y \in B} R_I(x,y)$	Avg	$\frac{1}{\lvert T_C \rvert} \sum_{B \in T_C} B(x)$	Max	$\max_{x \in X} C(x)$
MaxL	$\text{OWA}_{\text{Wmax-L}} \underbrace{R_I(x,y)}_{y \in B}$	OWAL	$\text{OWA}_{\text{Wmax-L}} \underbrace{B(x)}_{B \in T_C}$	MaxL	$\text{OWA}_{\text{Wmax-L}} \underbrace{C(x)}_{x \in X}$
MaxIA	$\text{OWA}_{\text{Wmax-IA}} \underbrace{R_I(x,y)}_{y \in B}$	OWAIA	$\text{OWA}_{\text{Wmax-IA}} \underbrace{B(x)}_{B \in T_C}$	MaxIA	$\text{OWA}_{\text{Wmax-IA}} \underbrace{C(x)}_{x \in X}$
Avg	$\frac{1}{\lvert B \rvert} \sum_{y \in B} R_I(x,y)$	Max	$\max_{B \in T_C} B(x)$	Avg	$\frac{1}{\lvert X \rvert} \sum_{x \in X} C(x)$
		CAvg	$1 - \frac{1}{\lvert co(T_C) \rvert} \sum_{B \in co(T_C)} B(x)$		
		CAvgL	$1 - \text{OWA}_{\text{Wmin-L}} \underbrace{B(x)}_{B \in co(T_C)}$		
		CAvgIA	$1 - \text{OWA}_{\text{Wmin-IA}} \underbrace{B(x)}_{B \in co(T_C)}$		

The cosine similarity has been shown to be the most appropriate similarity measure to use in high-dimensional multi-instance datasets related to text processing and image retrieval (e.g., [31]), examples of which are included in our experiments.

Given the similarity relation $R_I(\cdot,\cdot)$, we can define $B(x)$ in several ways. Table I lists the possibilities that we have included in this paper. They represent the most natural choices that one can make. First, we can set $B(x)$ to the maximum similarity of x with any instance in the bag B (Max). To allow for a more flexible approximation, we can replace the strict maximum operator by an OWA aggregation. We consider using a version using linearly decreasing weights (MaxL) and one where we apply inverse additive weights (MaxIA). As a fourth alternative, we can assign an equal weight to the similarity with all instances in B and set $B(x)$ to the average value (Avg). Apart from being very intuitive, we briefly note that these proposals are also related to fuzzy rough set theory [32]. The Max alternative coincides with the membership degree of x to the traditional fuzzy rough upper approximation of B, as shown in, e.g., [33]. MaxL and MaxIA correspond to the membership degree of x to the fuzzy rough upper approximation of B within the OWA-based fuzzy rough set model of [29].

b) Membership $C(x)$ of instances to classes: Having determined the values $B(x)$, we consider seven different definitions for $C(x)$. A first approach (Avg) is to average the membership degrees of x to all training bags labeled with class C. In this definition, T_C is the set of training bags of class C. Second, we consider a softened maximum membership degree of x to bags in T_C, calculated by means of an OWA aggregation. We consider linearly decreasing OWA weights (OWAL) or inverse additive weights (OWAIA). For completeness, we also include the strict maximum operator (Max), which allows for only one bag to take part in the definition of $C(x)$. An alternative approach is to consider the membership of x to bags not in T_C, that is, bags labeled with a different class. We can set $C(x)$ to the complement of the average membership of x to such bags (CAvg), where $co(T_C)$ is the complement of the set T_C. As above, we also include versions using the OWA operators. In this case, we replace the average in CAvg by a softened minimum operator. Once more, we can use linearly increasing weights (CAvgL) or inverse additive weights (CAvgIA).

c) *Membership $C(X)$ of bags to classes:* To finally determine the membership degree of the bag X to the class C, we can aggregate the instance-based values $C(x)$ in four different ways. First, we can set this value to the maximum membership degree of one of the instances in X to C (Max). This maximum can be softened by using an OWA operator with linear weights (MaxL) or inverse additive weights (MaxIA). As before, we also consider taking the average over the class membership degree of the instances (Avg).

2) Bag-Based Fuzzy Multi-Instance Classifiers (BFMIC Family): The bag-based definition of the fuzzy multi-instance classifiers relies on a similarity measure $R(\cdot,\cdot)$ between bags. The class membership degrees $C(X)$ are based on the similarity between X and the training bags. All settings for $R(\cdot,\cdot)$ and $C(X)$ evaluated in this paper are listed in Table II.

a) Bag-wise similarity $R(A,B)$: The similarity between bags A and B is based on the distance $\delta(\cdot,\cdot)$ between their instances, which we define based on the similarity measure (7), (8):

$$(\forall a \in A)(\forall b \in B)(\delta(a,b) = 1 - R_I(a,b)). \quad (9)$$

In this work, we consider six similarity functions, which can be divided into two groups of three. A first group is based on the Hausdorff distance [34] between two bags. First, we set the bag-wise similarity to the complement of this distance measure (Haus). Two other versions replace the maximum and minimum operators in this definition by OWA operators. The HausL version uses linear weights (2), (3), while HausIA uses the inverse additive weights (4), (5). The second group is based on the average Hausdorff distance [27]. The similarity based on this measure is given by AvgH. As above, we replace the minimum and maximum operators by OWA aggregations in the two modified versions: AvgHL and AvgHIA.

TABLE II
SETTINGS FOR THE BFMIC METHODS EVALUATED WITHIN THE PROPOSED FRAMEWORK

Code	$R(A,B)$	Code	$C(X)$
Haus	$1-\max(\max_{a\in A}\min_{b\in B}\delta(a,b), \max_{b\in B}\min_{a\in A}\delta(a,b))$	Avg	$\frac{1}{\lvert T_C\rvert}\sum_{B\in T_C} R(X,B)$
HausL	$1-\max(\underbrace{\mathrm{OWA}_{\text{Wmax-L}}\,\mathrm{OWA}_{\text{Wmin-L}}\underbrace{\delta(a,b)}_{b\in B}}_{a\in A},$	OWAL	$\mathrm{OWA}_{\text{Wmax-L}}\underbrace{R(X,B)}_{B\in T_C}$
	$\underbrace{\mathrm{OWA}_{\text{Wmax-L}}\,\mathrm{OWA}_{\text{Wmin-L}}\underbrace{\delta(a,b)}_{a\in A}}_{b\in B})$	OWAIA	$\mathrm{OWA}_{\text{Wmax-IA}}\underbrace{R(X,B)}_{B\in T_C}$
HausIA	$1-\max(\underbrace{\mathrm{OWA}_{\text{Wmax-IA}}\,\mathrm{OWA}_{\text{Wmin-IA}}\underbrace{\delta(a,b)}_{b\in B}}_{a\in A},$	Max	$\max_{B\in T_C} R(X,B)$
	$\underbrace{\mathrm{OWA}_{\text{Wmax-IA}}\,\mathrm{OWA}_{\text{Wmin-IA}}\underbrace{\delta(a,b)}_{a\in A}}_{b\in B})$	CAvg	$1-\frac{1}{\lvert co(T_C)\rvert}\sum_{B\in co(T_C)} R(X,B)$
AvgH	$1-\frac{\sum_{a\in A}\min_{b\in B}\delta(a,b)+\sum_{b\in B}\min_{a\in A}\delta(a,b)}{\lvert A\rvert+\lvert B\rvert}$	CAvgL	$1-\mathrm{OWA}_{\text{Wmin-L}}\underbrace{R(X,B)}_{B\in co(T_C)}$
AvgHL	$1-\frac{\sum_{a\in A}\mathrm{OWA}_{\text{Wmin-L}}\underbrace{\delta(a,b)}_{b\in B}+\sum_{b\in B}\mathrm{OWA}_{\text{Wmin-L}}\underbrace{\delta(a,b)}_{a\in A}}{\lvert A\rvert+\lvert B\rvert}$	CAvgIA	$1-\mathrm{OWA}_{\text{Wmin-IA}}\underbrace{R(X,B)}_{B\in co(T_C)}$
AvgHIA	$1-\frac{\sum_{a\in A}\mathrm{OWA}_{\text{Wmin-IA}}\underbrace{\delta(a,b)}_{b\in B}+\sum_{b\in B}\mathrm{OWA}_{\text{Wmin-IA}}\underbrace{\delta(a,b)}_{a\in A}}{\lvert A\rvert+\lvert B\rvert}$		

b) Membership $C(X)$ of bags to classes: Using a method to measure the similarity between bags, we consider seven ways to determine the membership degree of bag X to class C. These proposals coincide with the ways to compute $C(x)$ for the instance-based classifiers presented in Section III-A1. We consider the use of the average similarity (Avg) as well as softened maximum similarity degrees based on one of the two OWA operators (3) or (5) in OWAL and OWAIA, respectively. Again, for completeness, we also include the strict maximum itself (Max). Second, we can base the values $C(X)$ on bags not in T_C (CAvg). Replacing the average in this definition by softened minimum operators, we obtain versions CAvgL and CAvgIA.

We note that, when using one of the alternatives Avg, OWAL, OWAIA, or Max to calculate $C(X)$, our proposed bag-based classifiers exhibit a link with a nearest neighbor approach to MIC. In case of version Max, the classifier uses the most similar (nearest) training bag of each class to determine the membership of the unseen bag to that class. Since the final decision is made by expression (6), the class label of the overall nearest bag is automatically used as the prediction for the new bag. We can, therefore, conclude that the use of Max reduces our proposal to a 1-nearest neighbor multi-instance classifier, where the distance measure is taken as the complement of the similarity $R(\cdot,\cdot)$. When applying OWAL or OWAIA, however, we take a step away from the nearest neighbor paradigm. For each class, every training bag of that class contributes to the estimation of the membership degree to that class. Its contribution not only depends on its similarity with the bag to classify, but also on the number of bags in the class, since the class size determines the length of the OWA weight vector. In a nearest neighbor classifier, this would require the number of neighbors to be set to the number of samples, which is commonly avoided, combined with a complex voting procedure to replace the majority vote in the class prediction step. As part of our experimental study, we show that our proposal outperforms the most prominent nearest neighbor multi-instance classifier: CitationKNN [21].

3) Discussion: From the descriptions in the two preceding sections and Tables I and II, it should be clear that in various cases, an aggregation step can be set to the maximum of a group of values or to their average. These correspond to the most intuitive approaches that one can use. As stated above, the strict maximum assigns weight 1 to a single instance and weight 0 to all others. The average, on the other hand, involves all values in its calculation, assigning them equal weight. Although intuitive, we also want to consider more intermediate options. Due to their sorting procedure, OWA aggregations are ideally suited to model the tradeoff between the maximum and average. This motivates our use of the OWA operators in this paper. The same remark holds for the comparison of the minimum and average operators.

For the OWA weights, we use the vectors described by expressions (2)–(5). The selection of the weight vector in an OWA aggregation is an important question. Several procedures for this purpose have been proposed (see, e.g., [35]–[37]). Many of them are optimization methods, optimizing a certain objective function (e.g., entropy) for a user-specified value of the $orness$

of the weight vector. Within the scope of this paper, fixing the $orness$ of the weight vector beforehand feels arbitrary, and we, therefore, opt not to use such an optimization algorithm. We also avoid them in the interest of the computational cost of our proposal. For a given $orness$ value, existing optimization methods do not provide the user with a closed formula to determine the weights, but rather with a particular weight set of the specified length. As should be clear from the description of our framework above, the vector lengths are not fixed throughout the course of our algorithms. For instance, in the MaxL variant of $B(x)$ in Table I, the length of the weight vector is the size of the bag B. Since all bags in a multi-instance dataset can have a different size, all weight vectors can be different. The variable length implies that multiple runs of the optimization procedure would be necessary, imposing a significant cost. Furthermore, these methods become practically intractable when the lengths of the vectors increase. The procedure proposed in [38], for instance, relies on calculating the roots of a polynomial equation with degree equal to the length of the weight vector. As shown in the experimental section, we can expect the vector lengths, and therefore the degree of these equations, to be high (100 and above).

We keep in mind that we use the OWA aggregations in our goal to verify whether a tradeoff between the average and a strict maximum or minimum operator can improve the performance of a method within our proposed framework. We included the two types of weight vectors (2), (3) and (4), (5) and have shown that both represent softenings of the maximum and minimum operators (see the Appendix). As pointed out in [39], weights (2) and (3) correspond to the Borda count or law of Borda–Kendall in decision making. They are normalized versions of the vectors $\langle 1, 2, \ldots, p-1, p\rangle$ and $\langle p, p-1, \ldots, 2, 1\rangle$, respectively. The second set (4), (5) was put forward as an alternative to the Wmin-L and Wmax-L vectors in [40]. They are part of the class of harmonic OWA operators given in [41]. We also note that both (3) and (5) are buoyancy measures [42], which means that they contain decreasing weights. This is ideal for our effort to evaluate intermediate versions between the maximum and average.

In the $orness$ calculations included in the Appendix, we show that $orness$(Wmax-IA) is dependent on p, the length of the vector, while $orness$(Wmax-L) is independent of this value. This constitutes an important difference between the two types of OWA aggregations. We come back to this point in the experimental analysis, where the effects of this difference are illustrated. Furthermore, we also observe that the weights in vectors (2) and (3) are more evenly distributed among the positions compared with those in (4) and (5). This implies that the latter are more closely related to the strict minimum and maximum operators. As an example, consider the set of values $V = \{0.3, 0.5, 0.1, 1.0, 0.7\}$ to be aggregated. The weight vectors (2) and (4) of length 5 are

$$\text{Wmin-L}_5 = \left\langle \frac{2}{30}, \frac{4}{30}, \frac{6}{30}, \frac{8}{30}, \frac{10}{30} \right\rangle \approx \langle 0.07, 0.13, 0.20, 0.27, 0.33 \rangle$$

and

$$\text{Wmin-IA}_5 = \left\langle \frac{12}{137}, \frac{15}{137}, \frac{20}{137}, \frac{30}{137}, \frac{60}{137} \right\rangle \approx \langle 0.09, 0.11, 0.15, 0.22, 0.44 \rangle .$$

Clearly, the weights in Wmin-L_5 are more evenly distributed, while Wmin-IA_5 has a heavier tail to the end. The aggregations of V are $\text{OWA}_{\text{Wmin-L}_5}(V) \approx 0.3733$ and $\text{OWA}_{\text{Wmin-IA}_5}(V) \approx 0.3467$.

B. Overview of the Proposed Framework

As discussed above, the group of fuzzy classifiers consists of two families, which both require different settings to be specified. The general formulation of the BFMIC family is to compute $C(X) = \mathcal{A}^T[R(X,\cdot)]$, where $\mathcal{A}^T$ is an aggregation over training bags. For the IFMIC family, the general formulation would be $C(X) = \mathcal{A}^X_{x\in X}[\mathcal{A}^T[\mathcal{A}^B[R_I(x,\cdot)]]]$, with $\mathcal{A}^X$ an aggregation over instances inside the test bag X, $\mathcal{A}^T$ an aggregation over training bags, and $\mathcal{A}^B$ an aggregation over instances inside those training bags. In naming our methods, we use the following conventions, based on the abbreviations introduced in Tables I and II:

Instance-based fuzzy multi-instance classifiers: These methods are listed as IFMIC-$B(x)$-$C(x)$-$C(X)$. As an example, our instance-based classifier IFMIC-Max-Avg-MaxL uses Max to calculate the values $B(x)$, Avg to determine $C(x)$ and MaxL to finally find $C(X)$.

Bag-based fuzzy multi-instance classifiers: These methods are listed as BFMIC-$R(A,B)$-$C(X)$. As an example, our bag-based classifier BFMIC-Haus-Avg uses the Hausdorff similarity to determine the similarity between bags and Avg to calculate $C(X)$.

C. Worked Examples

In this section, we provide an illustration of the workflow of our proposed methods. We consider the classification of an unseen bag $X = \{x_1, x_2, x_3\}$. The training dataset consists of five bags B_1, B_2, B_3, B_4, and B_5. The first three belong to class C_1, and the other two to class C_2. In Table III, we provide the necessary values to compute the class membership degrees of X to the two classes. The table lists the similarity values $R_I(x,y)$ between instances in the training bags and those in X. We select two methods from our framework: the instance-based classifier IFMIC-MaxIA-Avg-Max and the bag-based classifier BFMIC-AvgH-OWAL. Below, we describe the calculations that these methods perform when classifying X.

1) Classification of IFMIC-MaxIA-Avg-Max: For each instance in the bag X, this method calculates its membership degree to the classes C_1 and C_2. The membership degree of X to the classes is then determined as the maximum value obtained by one of its instances. As an example, to compute the value $C_1(x_1)$, we select the bags belonging to class C_1, i.e., bags B_1, B_2, and B_3, and first estimate the membership degree of x_1 to

TABLE III
EXAMPLE INSTANCE SIMILARITY VALUES

Bag of class C_1		$R_I(\cdot, x_1)$	$R_I(\cdot, x_2)$	$R_I(\cdot, x_3)$
B_1	x_1^1	0.1	0.2	0.2
	x_2^1	0.7	0.6	0.3
	x_3^1	0.5	0.0	0.1
B_2	x_1^2	0.4	0.4	0.5
	x_2^2	0.3	0.6	0.4
B_3	x_1^3	0.7	0.8	0.7
	x_2^3	0.5	0.3	0.9
	x_3^3	0.7	0.2	0.2
Bag of class C_2		$R_I(\cdot, x_1)$	$R_I(\cdot, x_2)$	$R_I(\cdot, x_3)$
B_4	x_1^4	1.0	0.8	0.9
	x_2^4	0.7	0.6	0.3
	x_3^4	0.5	0.5	0.9
	x_4^4	0.2	0.9	0.8
B_5	x_1^5	0.6	0.3	0.7
	x_2^5	0.9	0.8	0.8
	x_3^5	0.7	1.0	0.6

these bags. We use version MaxIA to find these values:

$$B_1(x_1) = \text{OWA}_{W_{\text{max-IA}}} \underbrace{R_I(x_1, y)}_{y \in B_1} = \frac{6}{11} \cdot 0.7 + \frac{3}{11} \cdot 0.5 + \frac{2}{11} \cdot 0.1 = \frac{59}{110}.$$

In the same way, we find $B_2(x_1) = \frac{11}{30}$ and $B_3(x_1) = \frac{83}{110}$. In order to arrive at the class memberships of the instances, that is, the values $C(x)$, we use the Avg alternative. We find

$$C_1(x_1) = \frac{1}{3}\left(\frac{59}{110} + \frac{11}{30} + \frac{83}{110}\right) = \frac{547}{990}.$$

Analogously, we find $C_1(x_2) = \frac{485}{990}$ and $C_1(x_3) = \frac{469}{990}$. As a result, using Max, we determine the membership of X to class C_1 as

$$C_1(X) = \max\{C_1(x_1), C_1(x_2), C_1(x_3)\} = \frac{547}{990}.$$

For the second class, we find $C_2(X) = \frac{879}{1100} > C_1(X)$; therefore, the method assigns X to class C_2.

2) Classification of BFMIC-AvgH-OWAL: The prediction of this method is based on the similarity of X with the training bags. This similarity is measured by the average Hausdorff distance. Note that this definition makes use of the instance-wise distance rather than similarity. As stated in (9), these distance values can be obtained by taking the complement of the values in Table III. The similarity of X with bag B_1 is calculated as

$$\begin{aligned} R(X, B_1) &= 1 - \frac{\sum_{x \in X} \min_{b \in B_1} \delta(x, b) + \sum_{b \in B_1} \min_{x \in X} \delta(b, x)}{|X| + |B_1|} \\ &= 1 - \frac{(0.3 + 0.4 + 0.7) + (0.8 + 0.3 + 0.5)}{3 + 3} \\ &= \frac{1}{2}. \end{aligned}$$

Similarly, we find the values $R(X, B_2) = \frac{26}{50}$ and $R(X, B_3) = \frac{4}{5}$. Based on these values and the definition of OWAL, we find the class memberships of X as

$$\begin{aligned} C_1(X) &= \text{OWA}_{W_{\text{max-L}}} \underbrace{R(X, B)}_{B \in T_{C_1}} \\ &= \frac{6}{12} \cdot \frac{4}{5} + \frac{4}{12} \cdot \frac{26}{50} + \frac{2}{12} \cdot \frac{1}{2} = \frac{197}{300} \end{aligned}$$

and $C_2(X) = \frac{161}{180}$. As for the instance-based classifier, we find $C_2(X) > C_1(X)$; therefore, the method assigns bag X to class C_2.

D. Theoretical Complexity Analysis

We now analyze the theoretical complexity of our proposals. Below, we use the symbols n_B and n_X to denote the sizes of bags B and X and b and n to denote the number of training bags and instances, respectively. The value c is the number of classes in the dataset. Based on expressions (7) and (8), we derive that, when the instances in the bags are described with d features, the computation of the distance or similarity between them has complexity $\mathcal{O}(d)$.

1) Instance-Based Fuzzy Multi-Instance Classifier Methods: The membership of an instance to a bag B is determined by one of the alternatives listed in Table I. The Max and Avg versions are linear in the number of instances in B, and their complexity is, therefore, given as $\mathcal{O}(n_B \cdot d)$, not forgetting the cost of the instance-wise similarity computations. The use of the OWA aggregation in MaxL and MaxIA implies the additional cost of the sorting operation. We, therefore, derive that their complexity is $\mathcal{O}(n_B \cdot (\log(n_B) + d))$. Since the size of the training bags is limited by n, we find that Max and Avg have complexity $\mathcal{O}(n \cdot d)$ and MaxL and MaxIA complexity $\mathcal{O}(n \cdot (\log(n) + d))$.

Second, when the values $B(\cdot)$ have been computed, an IFMIC method determines the class membership of an instance. Avg, CAvg, and Max are obviously linear in the number of training bags and have complexity $\mathcal{O}(b)$. For the OWA-based versions (OWAL, OWAIA, CAvgL, CAvgIA), we have to account for the sorting step. Their complexity is, therefore, $\mathcal{O}(b \log(b))$. Note that these values ignore the cost of computing the membership degrees of instances to bags.

Finally, we determine the cost of calculating membership degrees of bags to classes, ignoring the cost of computing the instance-wise values. As for the first step, computing the membership of instances to bags, it is easily seen that Max and Avg have complexity $\mathcal{O}(n)$, and that for MaxL and MaxIa, we find $\mathcal{O}(n \log(n))$.

As an example for the overall complexity of an instance-based fuzzy classifier, we consider the method IFMIC-MaxL-OWAIA-MaxL. We select this classifier, as it has, for each step, the option with the highest computational complexity. As a result, its complexity constitutes an upper bound on the complexity of all proposed IFMIC methods. To classify a new bag X, for each class C, this method computes

$$\begin{aligned} C(X) &= \mathrm{OWA}_{\text{Wmax-L}} \underbrace{C(x)}_{x \in X} \\ &= \mathrm{OWA}_{\text{Wmax-L}} \underbrace{\mathrm{OWA}_{\text{Wmax-IA}} \underbrace{B(x)}_{B \in T_C}}_{x \in X} \\ &= \mathrm{OWA}_{\text{Wmax-L}} \underbrace{\mathrm{OWA}_{\text{Wmax-IA}} \underbrace{\mathrm{OWA}_{\text{Wmax-L}} \underbrace{R_I(x,y)}_{y \in B}}_{B \in T_C}}_{x \in X}. \end{aligned}$$

For each class, this calculation has complexity $\mathcal{O}(n_X \cdot (b \cdot (n \cdot (\log(n) + d)) + b\log(b)) + n_X \log(n_X)) = \mathcal{O}(n_X \cdot (b \cdot (n \cdot (\log(n) + d) + \log(b)) + \log(n_X)))$. If we take into account that this computation needs to be repeated for each class, the total complexity of this method and, therefore, the upper bound for the IFMIC family is given as

$$\mathcal{O}(c \cdot n_X \cdot (b \cdot (n \cdot (\log(n) + d) + \log(b)) + \log(n_X))). \tag{10}$$

This upper bound on the classification of an unseen bag by an instance-based classifier is log-linear in both the number of training instances and the number of training bags. At first sight, this implies that an increase in the number of bags or one in the number of instances is expected to have a similar effect on the runtime of the methods. Nevertheless, one should keep in mind that when the training set is increased by the addition of a new bag, the number of total instances automatically increases as well. The overall effect is, therefore, larger compared with a situation where additional instances are added to existing bags.

2) Bag-Based Fuzzy Multi-Instance Classifier Methods: In its first step, a bag-based method computes the similarity between bags. The Hausdorff and average Hausdorff distances (Haus and AvgH) are based on the pairwise distances between instances in the bags and, consequently, have complexity $\mathcal{O}(n^2 \cdot d)$. Their OWA modified versions (HausL, HausIA, AvgHL, and AvgHIA) require the pairwise distances to be sorted and have complexity $\mathcal{O}(n^2 \cdot (\log(n^2) + d)) = \mathcal{O}(n^2 \cdot (\log(n) + d))$.

Second, the method estimates the membership of a bag to the classes. As for the analogous instance membership degrees to the classes for the IFMIC family, we find that the Avg, CAvg, and Max settings have complexity $\mathcal{O}(b)$ and that OWAL, OWAIA, CAvgL, and CAvgIA have complexity $\mathcal{O}(b\log(b))$.

As an example, we determine the overall complexity of BFMIC-AvgHIA-OWAIA. As for the instance-based classifier used above, this method is one attaining the highest complexity within the BFMIC family, and its complexity, therefore, forms an upper bound on that of these methods. When classifying an unseen bag X, for each class C, this method calculates the value

$$C(X) = \mathrm{OWA}_{\text{Wmax-IA}} \underbrace{R(X,B)}_{B \in T_C},$$

with $R(X, \cdot)$ calculated using the AvgHIA alternative in Table II. For each class, we find that this calculation has a cost of $\mathcal{O}(b \cdot n_X \cdot n \cdot (\log(n_X \cdot n) + d) + b\log(b)) = \mathcal{O}(b \cdot (n_X \cdot n \cdot (\log(n_X \cdot n) + d) + \log(b)))$. Again considering that this computation is repeated for all classes, the total complexity of BFMIC-AvgH-OWAIA to classify a bag X is

$$\mathcal{O}(c \cdot b \cdot (n_X \cdot n \cdot (\log(n_X \cdot n) + d) + \log(b))). \tag{11}$$

As for the IFMIC family, we find that this upper bound is log-linear in the number of training bags as well as in the number of training instances.

3) Summary: Comparing the complexity bounds (10) and (11) for the instance-based and bag-based methods, respectively, we already noted that they both have a log-linear dependence on the number of training bags b and the number of training instances n. When we remove the terms that (10) and (11) have in common, we can derive that the former contains the additional $\mathcal{O}(c \cdot n_X \cdot b \cdot \log(b) + c \cdot n_X \cdot \log(n_X))$ terms and the latter the $\mathcal{O}(c \cdot b \cdot \log(b) + c \cdot b \cdot n_X \cdot n \cdot \log(n_X))$ terms. The first term, i.e., log-linear term in b, is n_X times larger in the IFMIC family than it is in the BFMIC family. The second term, i.e., log-linear term in n_X, is $n \cdot b$ times larger for the BFMIC methods compared with the IFMIC methods, a value linear in the number of both the training bags and the training instances. Overall, we can expect the second term to imply a larger difference between the costs of the two families. The IFMIC methods can, therefore, intuitively be expected to be less complex than their BFMIC relatives.

The cost of state-of-the-art methods in MIC is reviewed in the work of [3]. For methods from the bag space paradigm, the authors note that the cost can be expected to be at least quadratic in the number of training bags and in the average number of instances in them. For the instance space paradigm, the overall cost depends on the selected instance-level classifier, and a downside is that such a classifier is trained with a high number of instances, namely all those present in the training set. Finally, for the embedded space paradigm, the cost is divided into two parts. The first consists of the complexity to determine the new feature space; the second to perform the actual mapping from the original to the new space. The second part is estimated to be at least linear in the number of bags, the average number of instances in the bags, and the size of the original feature space and of the new feature space. For the definition of the new space, the cost is dependent on the algorithm. In certain methods, like BARTMIP [27], a clustering procedure is applied in this construction process. The cost thereof is the same as for the second part, i.e., linear in the number of bags and the average number of instances in them, but also depends on the number of iterations performed in the clustering. The conclusion of [3] is that the embedded space paradigm is the most appealing with respect to complexity and scalability.

IV. Experimental Study

In this section, we experimentally evaluate our classifiers. In Section IV-A, we describe our experimental setup. The experiments themselves are divided into two parts. First, in Section IV-B, we evaluate the different settings within our framework and consider their strengths and weaknesses. In the

TABLE IV
DATASETS USED IN THE EXPERIMENTAL STUDY

Name	# Bags	# Inst.	# Att.	Name	# Bags	# Inst.	# Att.
Musk1	92	476	166	TREC7	400	3367	300
Musk2	101	12 179	166	TREC9	400	3300	299
Atoms	188	1618	10	TREC10	400	3453	303
Bonds	188	3995	16	WIR7	113	3423	303
Chains	188	5349	24	WIR8	113	3423	303
Elephant	200	1391	230	WIR9	113	3423	301
Fox	200	1320	230	Corel1vs2	200	838	9
Tiger	200	1220	230	Corel1vs3	200	794	9
EastWest	20	213	24	Corel1vs4	200	1243	9
WestEast	20	213	24	Corel1vs5	200	684	9
AntDrugs5	400	3728	5	Corel2vs3	200	664	9
AntDrugs10	400	3787	10	Corel2vs4	200	1113	9
AntDrugs20	400	3736	20	Corel2vs5	200	554	9
TREC1	400	3224	320	Corel3vs4	200	1069	9
TREC2	400	3344	303	Corel3vs5	200	510	9
TREC3	400	3246	324	Corel4vs5	200	959	9
TREC4	400	3391	306				

We list the number of bags, number of instances (Inst.) and number of attributes (Att.).

second part, reported in Sections IV-C and IV-D, we conduct the comparison with state-of-the-art multi-instance classifiers.

A. Setup

We perform the experiments on the datasets listed in Table IV. Although our methods can be applied to multiclass dataset without modifications, all datasets in this study consist of only two classes, which more or less contain an equal amount of bags. Multiclass multi-instance datasets are not abundantly available in dataset repositories. In the recent review study of [3], the presence of only two classes, i.e., one positive and one negative, was also assumed. All reported results are obtained via fivefold cross validation.

To test for statistical significance in the observed differences in performance, we conduct a nonparametric analysis as advised in [43]. When comparing two methods, we use the Wilcoxon test [44]. The null hypothesis of this test is the equivalent performance of the two methods. It uses a ranking method to evaluate whether this hypothesis can be rejected. In particular, it ranks the absolute values of the differences in performance in increasing order and sums the ranks to R^+ and R^-. The former is the sum of the ranks of the positive differences, the latter the sum of the ranks of the negative differences. The smaller of the two values is used as test statistic. When the associated p-value is smaller than the predefined significance level α, the null hypothesis is rejected, and it is concluded that the two methods perform statistically significantly different. Second, to test for significant differences within a group of methods, we use the Friedman test [45]. This nonparametric test allows us to decide whether any statistically significant differences are present between M methods. The null hypothesis states that no such differences exist or, more precisely, that the behavior of all methods under consideration is equivalent. To detect a deviation from the null hypothesis, the test uses a ranking method. When the Friedman test rejects its null hypothesis, which it does when its p-value is lower than α, we conclude that statistically significant differences exist among the methods. However, there is no exact indication where these significant differences can be found, as the comparison is executed group-wise and not pair-wise. Therefore, we apply a post-hoc procedure. We use the Holm post-hoc procedure [46], using the method with the lowest Friedman ranking as control method, meaning that it is compared with the $M - 1$ remaining methods. The Holm procedure is a step-down approach and yields for each method m an adjusted p-value p_{Holm}, which represents the smallest global significance level at which a particular null hypothesis within a group of hypotheses would still be rejected. When the value is smaller than α, we conclude that the control method performs significantly better than method m. In this paper, we use $\alpha = 0.05$.

When comparing our proposals to the state of the art in Section IV-C, we follow [3] in our choice of a representative group of previously proposed multi-instance classifiers, available in Weka [47]. We opt not to include any ensemble schemes like MIBoost [48], as they do not make for a fair comparison with our current proposal. Our classifiers could themselves be set up in a boosting scheme, at which point they could be compared with other ensemble approaches. In this paper, our methods are compared with the following.

1) *MIWrapper* [20]: This is a wrapper approach to apply traditional single-instance learners (here, C4.5 [49]) to multi-instance data. To classify an unseen bag, all its instances are classified by the single-instance learner. The bag-wise prediction is obtained by taking the arithmetic mean of the predictions on the instances.
2) *miSVM* [19]: This algorithm is a support vector machine [50] for multi-instance problems, using the so-called maximum pattern margin formulation. We have set the complexity constant C to 1 and use a linear kernel.
3) *SimpleMI* [23]: This method converts every bag to a single instance. Afterward, it applies a traditional single-instance learner (here, C4.5) to classify these induced elements. The representative instance of a bag is constructed by taking the geometric mean of each attribute over all its instances.
4) *MILR* [48]: This algorithm is an extension of the traditional logistic regression classifier to the multi-instance setting. The model requires the specification of the procedure to aggregate instance-level class conditional probabilities to the level of bags. Here, we use the arithmetic mean, as recommended by the authors.
5) *MILES* [26]: This method converts the dataset to a traditional single-instance problem, such that the output of a single-instance classifier can be used to make predictions for unseen bags. In the original proposal, a support vector machine was used as base classifier. A more general version of this method was developed in [51], using alternative learners as base classifiers. They put forward AdaBoost as a strong choice as base learner. We use this method in our experimental study as well. Although AdaBoost is a boosting, and therefore ensemble, algorithm, this does not conflict with our previous statement that we will not be including ensemble multi-instance classifiers, because the boosting scheme is used as a single-instance classifier after application of the preprocessing method

TABLE V
TOP FIVE PERFORMING VERSIONS FOR BOTH FAMILIES BASED ON THE ACCURACY (ACC.)

Instance-based	Acc.	Time (s)	Bag-based	Acc.	Time (s)
IFMIC-Max-OWAIA-MaxL	80.9555	0.4739 ± 0.0079	BFMIC-AvgH-OWAIA	82.0125	0.8367 ± 0.2360
IFMIC-Max-OWAIA-Avg	80.9109	0.4723 ± 0.0092	BFMIC-HausIA-OWAIA	80.5781	1.0176 ± 0.2751
IFMIC-Max-OWAIA-MaxIA	80.3774	0.4786 ± 0.0036	BFMIC-AvgIA-OWAIA	80.1670	1.0473 ± 0.3397
IFMIC-MaxIA-OWAIA-Avg	79.7222	0.6564 ± 0.1896	BFMIC-HausL-OWAIA	79.4274	0.9584 ± 0.2381
IFMIC-Max-Max-MaxL	79.6222	0.4723 ± 0.0092	BFMIC-AvgH-OWAL	79.4122	0.7665 ± 0.1926

We also list the average runtime of the methods, taken over five runs.

MILES. The MILES method is a powerful representative of the embedded space paradigm. In earlier studies [24], [26], it has been shown to outperform other embedded alternatives like YARDS and DDSVM. This motivates our choice for its inclusion here.

6) *CitationKNN* [21]: This method uses the traditional k-nearest neighbor classifier (kNN) on the level of bags. The distance between bags is measured by the Hausdorff distance. It extends kNN by using both *references* and *citers*. The former are the nearest neighbors of a bag, while the latter are samples of which the bag under consideration is a nearest neighbor. When classifying the bag, the labels of both references and citers are considered, and a bag is assigned to the most prominent class among them. In our experiments, we use two references and four citers, as in [23].
7) *BARTMIP* [27]: This embedded algorithm consists of two stages. First, it applies a multi-instance clustering algorithm to cluster the training bags in k clusters. Based on this procedure, every bag is mapped to a new feature space. In this space, a bag is represented by a single vector containing k features, each feature being the distance of the corresponding bag to the center of a constructed cluster. Single-instance support vector machines are applied as classifier in the induced space. We have used the parameter settings of the original proposal [27].

Note that we have included representatives of the three MIC groups of the taxonomy of [3]: MIWrapper, miSVM, and MILR belong to the instance space paradigm, CitationKNN to the bag space paradigm, and SimpleMI, MILES, and BARTMIP to the embedded space paradigm. The review of [3] states that the best algorithms belong to the embedded space paradigm, of which we have included three prominent representatives. However, we do not want to limit our comparison to this group, as a limitation of the experimental study of [3] is the small number of datasets included in their experiments, too small to make a generalization in our opinion.

B. Setting Evaluation

For every family, we evaluate all members on the 33 datasets listed in Table IV and determine their accuracy. As can be derived from Tables I and II, this implies the evaluation of 112 instance-based classifiers and 42 bag-based classifiers. For both families, we rank the possible settings of each step according to their average obtained accuracy. This is achieved by fixing a particular option and averaging over all members of the family which take on this value. For example, to evaluate the use of the Hausdorff bag-similarity in the bag-based classifiers, we report the average value obtained by all BFMIC-Haus-* methods. A general overview of the performance of our methods can be found in Table V. For each family, the table lists the five methods with the highest classification accuracy within their family. The highest overall accuracy is attained by the bag-based classifier BFMIC-AvgH-OWAIA. The table also lists the execution times of the methods, taken as averages over all datasets and five runs. It is evident that all methods are very fast. A further observation is that using the maximum to determine the membership degrees $B(x)$ of instances to bags is the most prominent choice for this step that is encountered in the top five of instance-based classifiers. It seems to be best paired with the OWA-based approach using inverse additive weights (OWAIA) to calculate the membership degrees $C(x)$ of instances to classes. For the bag-based classifiers, the OWA-based approaches OWAL and OWAIA are the only ones found in the top five among the alternative ways to determine $C(X)$ for the bag-based methods. Version OWAIA in particular exhibits a good performance, an observation which we discuss in more detail below. The combination with the average Hausdorff similarity, without OWA modifications, to measure the bag-wise similarity $R(A, B)$ yields our most powerful classifier BFMIC-AvgH-OWAIA, outperforming all other combinations by quite a margin. In the following sections, we examine the settings of the two families in more detail.

1) Instance-Based Fuzzy Multi-Instance Classifiers: The IFMIC methods require the following three specifications: the ways to determine 1) the membership degrees $B(x)$ of instances to bags, 2) the membership degrees $C(x)$ of instances to classes, and 3) the membership degrees $C(X)$ of bags to classes. We have evaluated all combinations proposed in Section III-A1 and report their average accuracy results, obtained as described above, in Table VI.

Considering the choice for $B(x)$, the use of the maximum (Max) is clearly the best option, as was already observed from the general overview given in Table V. This step determines the membership of an instance x to a bag B, and the results show that the most similar instance $y \in B$ contains the most information. Involving all instances in B in this calculation, either by assigning them all equal weights (Avg) or not (MaxL, MaxIA), deteriorates the performance. This phenomenon is explained as follows. In a multi-instance dataset, the variety between instances in a bag can be quite large. Indeed, considering the standard two-class multi-instance hypothesis [52] that states

TABLE VI
SETTING RANKINGS FOR THE INSTANCE-BASED FUZZY MULTI-INSTANCE CLASSIFIERS, SORTED ON THE AVERAGE ACCURACY (ACC.)

$B(x)$	Acc.	Time (s)	$C(x)$	Acc.	Time (s)	$C(X)$	Acc.	Time (s)
Max	74.9221	0.4721 ± 0.0200	OWAIA	76.0551	0.5632 ± 0.1032	Avg	74.9641	0.5529 ± 0.1005
MaxIA	73.4730	0.6341 ± 0.1632	CAvg	74.4068	0.5527 ± 0.1073	MaxL	74.6617	0.5546 ± 0.0979
MaxL	73.0577	0.5681 ± 0.0369	OWAL	74.1138	0.5678 ± 0.1155	MaxIA	73.8590	0.5548 ± 0.0984
Avg	70.9472	0.5497 ± 0.1822	Avg	72.8451	0.5457 ± 0.0932	Max	68.9152	0.5616 ± 0.1054
			Max	72.1007	0.5140 ± 0.0391			
			CAvgL	71.5612	0.5723 ± 0.1206			
			CAvgIA	70.6173	0.5762 ± 0.1252			

We also list the average runtime and standard deviation in seconds, taken over five runs.

that a bag is positive when at least one of its instances belongs to the positive class, a positive bag can contain both instances affiliated with the positive concept as instances affiliated with the negative concept. Assume that we draw two instances x and y from bag B, which are affiliated with the positive and negative class, respectively, and that, based on their feature values, their similarity is (unsurprisingly) low. If we involved the value $R_I(x, y)$ in the calculation of $B(x)$ and $B(y)$, it would unjustifiably lower the results, even though x and y both belong to B; therefore, their membership degree should be high. We also observe that the OWA approaches, as intermediate options between the average and maximum, do not provide an advantage over the strict maximum, and we conclude that they are not useful for the estimation of $B(x)$.

The second step of an instance-based algorithm is its calculation of the instance-to-class membership degrees $C(x)$. First and foremost, it is important to observe the considerable difference between the two OWA approaches OWAL and OWAIA. For the calculations in the other two steps ($B(x)$ and $C(X)$), the differences between them are a lot smaller, up to the point that one cannot clearly be preferred over the other. For $C(x)$ however, the dominance of OWAIA is remarkable. The explanation lies in the length of the vectors to be aggregated in the OWA procedures. For the $B(x)$ and $C(X)$ steps, the length of these vectors are the bag sizes. In our experiments, the average bag size in the datasets is about 13. For $C(x)$, the length is equal to the class sizes, which is 117 on average. The larger the length of the inverse additive vectors (5), the larger their $orness$ value. When the length is 13, we find an $orness$ of 0.7427. For length 117, this value increases to 0.8199. As shown in the Appendix, the $orness$ of (3) is fixed at 0.6667. We conclude that the two OWA types are more different from each other in the $C(x)$ step compared with the $B(x)$ and $C(X)$ calculations, and we can therefore expect the overall results to differ more as well. This coincides with our observations. It remains to explain why OWAIA is the clear winner in the $C(x)$ step. First, we observe that the complement approaches (CAvg, CAvgL, and CAvgIA) are generally inferior to the regular versions (Avg, OWAL, OWAIA, and Max). Comparing the latter among themselves, we note that the average operator slightly outperforms the strict maximum. This implies that, in order to predict the membership degree of an instance x to a class C, we should consider all bags of that class, which coincides with our intuition. Nevertheless, the average is clearly outperformed by OWAL and OWAIA, showing that not all bags of a class should equally contribute to this prediction. Based on the average orness values, we can order the four alternatives as Avg (0.5), OWAL (0.6667), OWAIA (0.8199), and Max (1). They are more or less equally spaced in the interval [0.5, 1] and OWAIA is closer to the maximum than OWAL is. Evidently, the best performance is achieved by an OWA aggregation with $orness$ value around that of OWAIA.

TABLE VII
SETTING RANKINGS FOR THE BAG-BASED FUZZY MULTI-INSTANCE CLASSIFIERS, SORTED ON THE AVERAGE ACCURACY (ACC.)

$R(A, B)$	Acc.	Time (s)	$C(X)$	Acc.	Time (s)
AvgH	77.4463	0.7740 ± 0.1697	OWAIA	79.2166	0.9439 ± 0.2620
AvgHIA	76.2459	0.9588 ± 0.2294	OWAL	77.2857	0.8990 ± 0.2273
HausIA	75.8933	0.9367 ± 0.2107	CAvg	76.4118	1.0080 ± 0.2819
AvgHL	75.4407	0.9296 ± 0.2150	Max	75.6982	0.7500 ± 0.0072
HausL	75.2752	0.9272 ± 0.2171	Avg	74.8966	0.8549 ± 0.2140
Haus	71.4643	0.7741 ± 0.1759	CAvgL	73.0829	0.8695 ± 0.2010
			CAvgIA	70.4683	0.8588 ± 0.2273

We also list the average runtime and standard deviation in seconds, taken over five runs.

Finally, we consider the settings of the $C(X)$ calculations. It is interesting to observe that the ranking of the options is exactly the reverse of the one for $B(x)$. This makes sense for the opposite reasons listed above. The $C(X)$ degrees are aggregations of the values $C(x)$ values, for all instances $x \in X$. It is reasonable to expect that all instances in the bag should contribute equally to this calculation. The experimental results confirm this. The OWA versions, which still involve all instances although assigning them different weights, also yield decent, albeit somewhat inferior, results. The strict maximum does not perform well, and we observe a large gap with its competitors.

2) Bag-Based Fuzzy Multi-Instance Classifiers: The bag-based classifiers were introduced in Section III-A2 and require the specification of the ways to determine 1) the bag-wise similarity measure $R(A, B)$ and 2) the membership degree $C(X)$ of bags to classes. We present the rankings in Table VII.

The experiments show that, for the bag-wise similarity, the average Hausdorff similarity is the best choice. The preference of the average over the regular Hausdorff distance was already expressed in [27]. Modified versions using OWA aggregations do not further improve its performance. This is entirely in line with the observations with regard to the $B(x)$ step in the instance-based classifiers. There, the strict maximum proved superior to the OWA-softened versions, and we could attribute this to the variety between instances within bags. The same occurs here. The OWA modifications AvgHL and

TABLE VIII
ACCURACY AND κ RESULTS FOR THE CLASSIFIERS, TOGETHER WITH THE FRIEDMAN RANKS OF THE METHODS AND THE CORRESPONDING P-VALUES OF THE HOLM POST-HOC PROCEDURE

Method	Acc.	Rank	p_{Holm}	Method	κ	Rank	p_{Holm}
BFMIC	82.013 ± 10.132	3.8030	-	BFMIC	0.628 ± 0.216	3.8788	≥ 0.999999
IFMIC	80.911 ± 10.311	4.3030	≥ 0.999999	IFMIC	0.611 ± 0.213	4.2424	≥ 0.999999
MILES	80.738 ± 12.655	3.9394	≥ 0.999999	MILES	0.610 ± 0.254	3.8788	-
MIWrapper	80.139 ± 11.870	4.2424	≥ 0.999999	MIWrapper	0.595 ± 0.241	4.3485	≥ 0.999999
SimpleMI	79.492 ± 10.992	4.8636	0.578440	SimpleMI	0.586 ± 0.221	4.7849	0.887649
MILR	78.452 ± 9.311	5.6667	**0.034235**	MILR	0.554 ± 0.199	5.6970	**0.042006**
BARTMIP	77.616 ± 15.337	4.3636	≥ 0.999999	BARTMIP	0.545 ± 0.307	4.3939	≥ 0.999999
miSVM	73.686 ± 14.351	7.1818	**0.000004**	miSVM	0.443 ± 0.316	7.1667	**0.000009**
CitKNN	70.465 ± 18.901	6.6364	**0.000185**	CitKNN	0.398 ± 0.378	6.6061	**0.000366**

P-values implying significant differences with the control method are printed in bold.

AvgHIA of the average Hausdorff instance lie at the level of the instance-wise comparisons. Once again, we find that the least distant (most similar) instance carries the most information.

Second, we consider the settings of the $C(X)$ step. We can make a similar observation as in our discussion of $C(x)$ for the instance-based classifiers above. For the calculations of $C(X)$, a notable difference is observed between the two types of OWA aggregations, while this is less prominent for those in $R(A, B)$. The reason is the same as before: The lengths of the vectors to be aggregated to find $C(X)$ are the average class sizes, while those for $R(A, B)$ are the average bag sizes. For the larger class sizes, the $orness$ of an inverse additive weight vector is considerably larger than that of a linear weight vector and achieves a more suitable tradeoff between taking the average and the maximum.

C. Comparison With the State of the Art

We continue our experimental study with a comparison of our proposal to state-of-the-art MIC methods. We select the best-performing member of both families, as presented in Table V, in order to evaluate where each family roughly places compared with previous proposals. We use the family-wise abbreviations to denote our best-performing methods, i.e., BFMIC stands for BFMIC-AvgH-OWAIA and IFMIC for IFMIC-Max-OWAIA-Avg. Note that the latter is listed at second place in Table V, but its settings coincide with all top choices for the instance-based classifiers, which makes it more appropriate to include here.

We evaluate all classifiers by their obtained accuracy and value for Cohen's kappa (κ, [53]). The latter was introduced as a coefficient of agreement between two judges and can be used to measure the agreement between actual and predicted classes in a classification process. By definition, κ accounts for random hits or, using the terminology of the original proposal, agreement by chance. When the dataset contains Ω classes, this measure is defined as $\kappa = \frac{n \sum_{i=1}^{\Omega} c_{ii} - \sum_{i=1}^{\Omega} c_{i.} c_{.i}}{n^2 - \sum_{i=1}^{\Omega} c_{i.} c_{.i}}$, where n is the total number of instances, c_{ii} is the number of correctly classified instances of the ith class, $c_{i.}$ is the cardinality of this class, and $c_{.i}$ represents the number of elements that was classified as belonging to it. The values of κ are contained in the interval $[-1, 1]$. We interpret $\kappa = 1$ as total agreement between reality and prediction and $\kappa = -1$ as total disagreement. The value $\kappa = 0$ corresponds to a random level of agreement, implying that the classifier is equivalent to random guessing.

We present the results and the statistical analysis in Table VIII. The detailed results for each dataset can be found on the webpage http://www.cwi.ugent.be/sarah.php. We observe that our proposed bag-based classifier BFMIC exhibits the best performance for both evaluation measures. Our instance-based classifier obtains very competitive results as well, being second for both measures. We conducted a Friedman test for the two evaluation measures. Both p-values were smaller than 0.000001, from which we conclude that significant differences are observed. For the accuracy, the lowest Friedman rank is assigned to our BFMIC method. Based on the post-hoc procedure, we conclude that our method performs significantly better than MILR, miSVM, and CitationKNN. It does not significantly outperform our IFMIC classifier, nor the other included state-of-the-art methods. However, the assignment of the lowest Friedman rank does form an indication to the superiority of our proposal. For the κ measure, BFMIC and MILES are assigned the same rank. The test internally considers MILES to have the lowest rank, and the post-hoc procedure shows that it significantly outperforms the MILR, miSVM, and CitationKNN algorithms. It does not outperform our classifiers. Second, we take a closer look at the differences between BFMIC and MILES by means of a Wilcoxon test. In the "BFMIC versus MILES" comparison for the accuracy, we find $R^+ = 284.0$, $R^- = 244.0$ and the p-value 0.694687. For the κ values, we find $R^+ = 279.0$, $R^- = 249.0$, and p-value 0.770902. In both situations, we cannot reject the null hypothesis. BFMIC has the advantage in the comparison however, and Table V also shows that its average performance is better. We also note that a disadvantage of the MILES method is that it only performs preprocessing and needs to be combined with a classifier. Our methods are standalone classifiers and can immediately be applied to the problem, eliminating the choice of a posterior classification algorithm that MILES requires. Additionally, based on the conclusions drawn in Section IV-B, we can put forward the settings of BFMIC and IFMIC as optimal within the evaluated set, avoiding any further tuning required of the user, if not possible or desirable. We conclude that our proposal is clearly competitive with the state of the art in MIC.

D. Within-Domain Comparison

The datasets in Table IV originate from different application domains. We consider textual data applications in the three

TABLE IX
ACCURACY AND κ RESULTS FOR THE CLASSIFIERS WITHIN EACH DOMAIN

	Method	Acc.	Method	κ
Bio-IT (8 datasets)	MIWrapper	80.355 ± 4.025	MILES	0.579 ± 0.095
	MILES	79.859 ± 4.640	MIWrapper	0.575 ± 0.109
	BARTMIP	79.686 ± 4.021	BARTMIP	0.562 ± 0.103
	MILR	77.742 ± 4.735	SimpleMI	0.515 ± 0.077
	SimpleMI	76.610 ± 3.964	MILR	0.493 ± 0.158
	CitKNN	76.093 ± 5.795	**IFMIC**	**0.487 ± 0.101**
	IFMIC	**75.901 ± 3.611**	CitKNN	0.477 ± 0.144
	BFMIC	**75.216 ± 3.527**	**BFMIC**	**0.455 ± 0.117**
	miSVM	70.783 ± 4.127	miSVM	0.284 ± 0.227
Text (10 datasets)	SimpleMI	77.585 ± 9.895	SimpleMI	0.552 ± 0.198
	MIWrapper	76.976 ± 7.641	MIWrapper	0.539 ± 0.153
	BFMIC	**76.568 ± 5.907**	**BFMIC**	**0.532 ± 0.118**
	IFMIC	**76.135 ± 7.349**	**IFMIC**	**0.524 ± 0.147**
	MILR	73.799 ± 6.388	MILR	0.476 ± 0.128
	MILES	71.599 ± 12.898	miSVM	0.434 ± 0.181
	miSVM	71.536 ± 9.209	MILES	0.431 ± 0.259
	BARTMIP	61.733 ± 7.416	BARTMIP	0.235 ± 0.148
	CitKNN	53.560 ± 9.100	CitKNN	0.070 ± 0.181
Logic programming (2 datasets)	**BFMIC**	**80.000 ± 0.000**	**BFMIC**	**0.600 ± 0.000**
	MILES	77.500 ± 2.500	MILES	0.550 ± 0.050
	IFMIC	**72.500 ± 2.500**	**IFMIC**	**0.450 ± 0.050**
	MILR	65.000 ± 0.000	MILR	0.300 ± 0.000
	BARTMIP	62.500 ± 7.500	BARTMIP	0.250 ± 0.150
	SimpleMI	60.000 ± 5.000	SimpleMI	0.200 ± 0.100
	MIWrapper	50.000 ± 0.000	MIWrapper	0.000 ± 0.000
	CitKNN	50.000 ± 0.000	CitKNN	0.000 ± 0.000
	miSVM	42.500 ± 7.500	miSVM	−0.150 ± 0.150
Image (13 datasets)	BARTMIP	90.885 ± 11.328	BARTMIP	0.818 ± 0.227
	BFMIC	**90.692 ± 9.926**	**BFMIC**	**0.814 ± 0.199**
	IFMIC	**88.962 ± 10.538**	**IFMIC**	**0.779 ± 0.211**
	MILES	88.808 ± 11.528	MILES	0.776 ± 0.231
	MIWrapper	87.500 ± 10.353	MIWrapper	0.742 ± 0.207
	SimpleMI	85.731 ± 10.611	SimpleMI	0.715 ± 0.212
	MILR	84.539 ± 9.645	MILR	0.691 ± 0.193
	CitKNN	83.154 ± 18.913	CitKNN	0.663 ± 0.378
	miSVM	81.923 ± 14.567	miSVM	0.639 ± 0.291

Our proposals are printed in bold.

WIR datasets (web mining) and the seven TREC datasets (general medical documentation mining). A second group of eight datasets come from the bioinformatics domain, particularly considering the prediction of chemical activity. This group contains Atoms, Bonds, and Chains (molecular mutagenicity), Musk1 and Musk2 (musky odor), and the three AntDrugs datasets (antagonist drugs). The third group represents image categorization problems. It includes 13 datasets in total: Elephant, Fox, Tiger, and the ten Corel datasets. Finally, datasets EastWest and WestEast originate from an artificial inductive logic programming problem.

Table IX presents the domain-specific rankings of the included methods. It shows that our proposals, BFMIC in particular, perform very well in the image categorization domain and for the artificial inductive logic programming problem. In text processing, the results of BFMIC and IFMIC are not the highest among the group, but they do perform notably better than MILES. The reason that we could not conclude that BFMIC is statistically significantly better than MILES can be found in the results for the bioinformatics datasets. The performance of both BFMIC and IFMIC for this group of eight datasets is quite a bit worse than that of MILES. We suspect the defining characteristic of these datasets leading to the inferior performance of our methods, to be their higher degree of imbalance between the two classes. Indeed, the classes in the image, text, and logic programming problems are almost all perfectly balanced, with the highest observed degree of imbalance, which represents how much larger the majority class is compared with the minority class, being 1.05. On the other hand, the average imbalance in the bioinformatics group reaches 1.49. In research on class imbalance, this value constitutes only a mild degree of imbalance, but we suspect it to be the defining characteristic causing the observed lesser performance. To counteract this phenomenon, we propose to integrate some imbalance-resistant heuristics within our classifier framework as future work.

Finally, we perform a statistical comparison between BFMIC and MILES by means of the Wilcoxon test for the Bio-IT, Text, and Image data groups. We exclude the logic programming datasets from this analysis, as a group consisting of two elements does not allow us to draw any strong conclusions. We observe that MILES only outperforms BFMIC with statistical significance for the Bio-IT datasets, indeed implying that there is still room for improvement for our proposal for this type of problems. The p-values for the accuracy measure was 0.019533 and that for κ was 0.007812. For the remaining two groups, BFMIC attains better results, but cannot yet be concluded to perform significantly better than MILES, although the p-value for the image datasets is close to significance, being 0.08728 for the accuracy and 0.08032 for κ.

V. CONCLUSION AND FUTURE WORK

MIC is the task of assigning a class label to a bag of instances based on an available training set of labeled bags. The decision can be made using discriminative information based on instances (instance-based) or bags (bag-based). In this paper, we have proposed not merely a single classifier, but an entire framework of multi-instance classifiers using fuzzy set theory. The framework consists of two families: bag-based and instance-based fuzzy multi-instance classifiers. We have laid out a nonexhaustive list of settings for all families and experimentally evaluated them within the framework. We have offered intuitive interpretations why certain choices perform better than others, supported by experimental evidence. Furthermore, we have conducted an experimental comparison of our methods with previously proposed multi-instance classifiers and found them to be performing very well.

The list of proposed settings is not complete, and the search for optimal fuzzy classifiers can be continued easily by plugging new proposals into our framework. In this paper, our focus has been on setting up the framework, and we limited ourselves to the most natural settings so far. As other future work, we would like to extend and evaluate our proposal for specific challenges in MIC, like class imbalance in multi-instance datasets.

APPENDIX
ORNESS OF ORDERED WEIGHTED AVERAGE WEIGHT VECTORS

We show that the $orness$ of the OWA weight vectors (3) and (5) always exceeds 1/2, proving that they soften the maximum operator. Due to the symmetry in the definitions, (2) and (4) can be shown to soften the minimum in an analogous way. First, for vector (3), it was shown in, e.g., [40] that $orness(\text{Wmax-L}) = \frac{2}{3} > \frac{1}{2}$. For weight vector (5), where we set $D = \sum_{i=1}^{p} \frac{1}{i}$ for the sake of readability, we easily find $orness(\text{Wmax-IA}) = \frac{p(D-1)}{(p-1)D}$. We need to verify whether this value is always larger than 1/2:

$$\begin{aligned} & orness(\text{Wmax-IA}) > \frac{1}{2} \\ \Leftrightarrow\ & \frac{p(D-1)}{(p-1)D} > \frac{1}{2} \\ \Leftrightarrow\ & 2p(D-1) > (p-1)D \\ \Leftrightarrow\ & p(D-2) + D > 0. \end{aligned}$$

When $p = 4$, we have $D = \frac{25}{12} > 2$. This implies that there are only positive terms in the sum on the left-hand side, which is, therefore, larger than 0. The inequality also holds for all $p > 4$, as D clearly increases with p. It remains to be verified whether the $orness$ degree of Wmax-IA also exceeds $\frac{1}{2}$ when $p = 2$ or $p = 3$. We immediately find $orness(\text{Wmax-IA}_2) = \frac{2}{3}$ and $orness(\text{Wmax-IA}_3) = \frac{15}{22}$, both of which are higher than $\frac{1}{2}$, and conclude that (5) indeed softens the maximum operator.

REFERENCES

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

[2] E. Alpaydın, V. Cheplygina, M. Loog, and D. M. Tax, "Single-vs. multiple-instance classification," *Pattern Recog.*, vol. 48, no. 9, pp. 2831–2838, 2015.

[3] J. Amores, "Multiple instance classification: Review, taxonomy and comparative study," *Artif. Intell.*, vol. 201, pp. 81–105, 2013.

[4] L. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965.

[5] J. Keller, M. Gray, and J. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-15, no. 4, pp. 580–585, Jul./Aug. 1985.

[6] M. Elkano, M. Galar, J. Sanz, A. Fernandez, E. Barrenechea, F. Herrera, and H. Bustince Sola, "Enhancing multi-class classification in FARC-HD fuzzy classifier: On the synergy between n-dimensional overlap functions and decomposition strategies," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1562–1580, Oct. 2015.

[7] J. Hühn and E. Hüllermeier, "FURIA: An algorithm for unordered fuzzy rule induction," *Data Mining Knowl. Discovery*, vol. 19, no. 3, pp. 293–319, 2009.

[8] D. Savic and W. Pedrycz, "Evaluation of fuzzy linear regression models," *Fuzzy Sets Syst.*, vol. 39, no. 1, pp. 51–63, 1991.

[9] H. Wang and R. Tsaur, "Insight of a fuzzy regression model," *Fuzzy Sets Syst.*, vol. 112, no. 3, pp. 355–369, 2000.

[10] H. Jiang, C. Kwong, W. Ip, and Z. Chen, "Chaos-based fuzzy regression approach to modeling customer satisfaction for product design," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 5, pp. 926–936, Oct. 2013.

[11] J. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, no. 2, pp. 191–203, 1984.

[12] T. Glenn, A. Zare, and P. Gader, "Bayesian fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1063–1561, Oct. 2015.

[13] G. Beliakov, L. Gang, V. Huy Quan, and T. Wilkin, "Characterizing compactness of geometrical clusters using fuzzy measures," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 4, pp. 1030–1043, Aug. 2015.

[14] Z. Zhou, K. Jiang, and M. Li, "Multi-instance learning based web mining," *Appl. Intell.*, vol. 22, no. 2, pp. 135–147, 2005.

[15] A. Zafra, C. Romero, S. Ventura, and E. Herrera-Viedma, "Multi-instance genetic programming for web index recommendation," *Exp. Syst. Appl.*, vol. 36, no. 9, pp. 11470–11479, 2009.

[16] B. Babenko, M. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2009, pp. 983–990.

[17] A. Zafra, C. Romero, and S. Ventura, "Multiple instance learning for classifying students in learning management systems," *Exp. Syst. Appl.*, vol. 38, no. 12, pp. 15020–15031, 2011.

[18] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *Proc. Conf. Adv. Neural Inf. Process. Syst.*, 1998, pp. 570–576.

[19] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proc. Conf. Adv. Neural Inf. Process. Syst.*, 2002, pp. 561–568.

[20] E. Frank and X. Xu, "Applying propositional learning algorithms to multi-instance data," Dept. Comput. Sci., Univ. Waikato, Hamilton, New Zealand, Tech. Rep. 06/03, 2003.

[21] J. Wang and J. Zucker, "Solving multiple-instance problem: A lazy learning approach," in *Proc. 17th Int. Conf. Mach. Learning*, 2000, pp. 1119–1125.

[22] Z. Zhou, Y. Sun, and Y. Li, "Multi-instance learning by treating instances as non-iid samples," in *Proc. 26th Ann. Int. Conf. Mach. Learning*, 2009, pp. 1249–1256.

[23] L. Dong, "A comparison of multi-instance learning algorithms," Ph.D. dissertation, Univ. Waikato, Hamilton, New Zealand, 2006.

[24] J. Foulds, "Learning instance weights in multi-instance learning," Ph.D. dissertation, Univ. Waikato, Hamilton, New Zealand, 2008.

[25] Y. Chen and J. Wang, "Image categorization by learning and reasoning with regions," *J. Mach. Learning Res.*, vol. 5, pp. 913–939, 2004.

[26] Y. Chen, J. Bi, and J. Wang, "MILES: Multiple-instance learning via embedded instance selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1931–1947, Dec. 2006.

[27] M. Zhang and Z. Zhou, "Multi-instance clustering with applications to multi-instance prediction," *Appl. Intell.*, vol. 31, no. 1, pp. 47–68, 2009.

[28] R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decisionmaking," *IEEE Trans. Syst., Man Cybern.*, vol. 18, no. 1, pp. 183–190, Jan./Feb. 1988.

[29] C. Cornelis, N. Verbiest, and R. Jensen, "Ordered weighted average based fuzzy rough sets," in *Rough Set and Knowledge Technology*. New York, NY, USA: Springer, 2010, pp. 78–85.

[30] C. Aggarwal, A. Hinneburg, and D. Keim, *On the Surprising Behavior of Distance Metrics in High Dimensional Space*. New York, NY, USA: Springer, 2001.

[31] D. Sánchez Tarragó, C. Cornelis, R. Bello, and F. Herrera, "A multi-instance learning wrapper based on the Rocchio classifier for web index recommendation," *Knowl.-Based Syst.*, vol. 59, pp. 173–181, 2014.

[32] D. Dubois and H. Prade, "Rough fuzzy sets and fuzzy rough sets," *Int. J. Gen. Syst.*, vol. 17, nos. 2/3, pp. 191–209, 1990.

[33] N. Verbiest, C. Cornelis, and R. Jensen, "Fuzzy rough positive region based nearest neighbour classification," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2012, pp. 1961–1967.

[34] G. Edgar, *Measure, Topology, and Fractal Geometry*. New York, NY, USA: Springer Science & Business Media, 2007.

[35] Z. Xu, "An overview of methods for determining OWA weights," *Int. J. Intell. Syst.*, vol. 20, no. 8, pp. 843–865, 2005.

[36] X. Liu, "A review of the OWA determination methods: Classification and some extensions," in *Recent Developments in the Ordered Weighted Averaging Operators: Theory and Practice*. New York, NY, USA: Springer, 2011, pp. 49–90.

[37] A. Kishor, A. Singh, and N. Pal, "Orness measure of OWA operators: A new approach," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 4, pp. 1039–1045, Aug. 2014.

[38] R. Fullér and P. Majlender, "An analytic approach for obtaining maximal entropy OWA operator weights," *Fuzzy Sets Syst.*, vol. 124, no. 1, pp. 53–57, 2001.

[39] M. Lamata and E. Pérez, "Obtaining OWA operators starting from a linear order and preference quantifiers," *Int. J. Intell. Syst.*, vol. 27, no. 3, pp. 242–258, 2012.

[40] N. Verbiest, "Fuzzy rough and evolutionary approaches to instance selection," Ph.D. dissertation, Ghent Univ., Ghent, Belgium, 2014.

[41] J. Montero and V. Cutello, "Aggregation rules in committee procedures," in *The Ordered Weighted Averaging Operators*. New York, NY, USA: Springer, 1997, pp. 219–237.

[42] R. Yager, "Families of OWA operators," *Fuzzy Sets Syst.*, vol. 59, no. 2, pp. 125–148, 1993.

[43] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, no. 10, pp. 2044–2064, 2010.

[44] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.

[45] M. Friedman, “The use of ranks to avoid the assumption of normality implicit in the analysis of variance,” *J. Amer. Statist. Assoc.*, vol. 32, no. 200, pp. 675–701, 1937.
[46] S. Holm, “A simple sequentially rejective multiple test procedure,” *Scandinavian J. Statist.*, vol. 6, pp. 65–70, 1979.
[47] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, “The WEKA data mining software: an update,” *ACM SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.
[48] X. Xu and E. Frank, “Logistic regression and boosting for labeled bags of instances,” in *Advances in Knowledge Discovery and Data Mining*. New York, NY, USA: Springer, 2004, pp. 272–281.
[49] J. Quinlan, *C4.5: Programs for Machine Learning*, vol. 1. San Mateo, CA, USA: Morgan Kaufmann, 1993.
[50] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learning*, vol. 20, no. 3, pp. 273–297, 1995.
[51] J. Foulds and E. Frank, “Revisiting multiple-instance learning via embedded instance selection,” in *Proc. 21st Australasian Joint Conf. Artif. Intell.*, 2008, pp. 300–310.
[52] N. Weidmann, E. Frank, and B. Pfahringer, “A two-level learning method for generalized multi-instance problems,” in *Proc. 14th Eur. Conf. Mach. Learning*, 2003, pp. 468–479.
[53] J. Cohen, “A coefficient of agreement for nominal scales,” *Edu. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, 1960.

Sarah Vluymans received the Master’s degree in mathematical informatics from Ghent University, Ghent, Belgium, in 2014. She is currently working toward the Ph.D. degree with Ghent University and the Inflammation Research Center, part of the Flemish Institute of Biotechnology, as a Member of the DAMBI research group (Data Mining and Modeling for Biomedicine).

Her research is funded by the Special Research Fund (BOF) by Ghent University. Her research interests include the integration of concepts from fuzzy rough theory in a wide variety machine learning techniques.

Dánel Sánchez Tarragó received the Ph.D. degree from the University of Granada, Granada, Spain, in 2014.

He is currently a Postdoctoral Researcher with the Center of Information Studies, Central University “Marta Abreu” of Las Villas, Santa Clara, Cuba. His current research interests include soft computing, machine learning, data mining, and complexity theory.

Yvan Saeys received the M.Sc. and Ph.D. degrees in computer science from Ghent University, Ghent, Belgium.

He is currently leading the DAMBI group, Ghent University, where his research focuses on the development and application of data mining and machine learning techniques for biological and medical applications. He has published more than 60 papers in international journals and high-profile conferences and has been involved in the organization of many international workshops and conferences on machine learning and bioinformatics. His current research interests include instance and feature selection, large-scale machine learning, and machine learning for structured input and output spaces.

Chris Cornelis received the M.Sc. and Ph.D. degrees in computer science from Ghent University, Ghent, Belgium.

He is currently a Postdoctoral Fellow with the University of Granada, Granada, Spain, supported by the Ramón y Cajal programme of the Spanish government. He is also a Guest Professor with Ghent University. He has cosupervised seven Ph.D. theses and has authored more than 130 papers in international journals, edited volumes, and conference proceedings. His current research interests include fuzzy sets, rough sets, and machine learning.

Dr. Cornelis serves as an Executive Board Member of the International Rough Set Society.

Francisco Herrera (SM’15) received the M.Sc. and Ph.D. degrees in mathematics from the University of Granada, Granada, Spain, in 1988 and 1991, respectively.

He is currently a Professor with the Department of Computer Science and Artificial Intelligence, University of Granada. He has been the supervisor of 38 Ph.D. students. He has published more than 300 papers in international journals. He is coauthor of the books: *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases* (Singapore: World Scientific, 2001), *Data Preprocessing in Data Mining* (Springer, 2015), *The 2-tuple Linguistic Model. Computing with Words in Decision Making* (Springer, 2016). His current research interests include bibliometrics, computing with words in decision making, information fusion, evolutionary algorithms, evolutionary fuzzy systems, biometrics, data preprocessing, data science, and big data. He became a senior member of the IEEE in 2015.

Dr. Herrera is an Editor-in-Chief of *Information Fusion* (Elsevier) and *Progress in Artificial Intelligence* (Springer). He acts as Editorial Board Member or Associate Editor of a number of journals, among others: IEEE TRANSACTIONS ON FUZZY SYSTEMS, IEEE TRANSACTIONS ON CYBERNETICS, *Information Sciences*, *Knowledge and Information Systems*, *Fuzzy Sets and Systems*, *International Journal of Computational Intelligence Systems*, *Applied Intelligence*, *Knowledge-Based Systems*, *Memetic Computation*, and *Swarm and Evolutionary Computation*. He received the following honors and awards: ECCAI Fellow 2009, IFSA Fellow 2013, 2010 Spanish National Award on Computer Science ARITMEL to the “Spanish Engineer on Computer Science,” International Cajastur “Mamdani” Prize for Soft Computing (Fourth Edition, 2010), IEEE TRANSACTIONS ON FUZZY SYSTEM Outstanding 2008 and 2012 Paper Award (bestowed in 2011 and 2015 respectively), 2011 Lotfi A. Zadeh Prize Best paper Award of the International Fuzzy Systems Association, 2013 AEPIA Award to a scientific career in Artificial Intelligence (September 2013), and 2014 XV Andaluca Research Prize Maimnides (By the regional government of Andaluca). He belongs to the list of the Highly Cited Researchers in the areas of Engineering and Computer Sciences: http://highlycited.com/.