DE G DE GRUYTER OPEN

Int. J. Appl. Math. Comput. Sci., 2016, Vol. 26, No. 2, 407–421
DOI: 10.1515/amcs-2016-0029

amcs

AN n-ARY λ-AVERAGING BASED SIMILARITY CLASSIFIER

ONESFOLE KURAMA [a,b,*], PASI LUUKKA [a,c], MIKAEL COLLAN [c]

[a] Laboratory of Applied Mathematics
Lappeenranta University of Technology, P.O. Box 20, FIN-53851 Lappeenranta, Finland
e-mail: onesfole.kurama@lut.fi

[b] Department of Mathematics
Makerere University, P.O. Box 7062, Kampala, Uganda

[c] School of Business and Management
Lappeenranta University of Technology, P.O. Box 20, FIN-53851 Lappeenranta, Finland
e-mail: {pasi.luukka,mikael.collan}@lut.fi

We introduce a new n-ary λ similarity classifier that is based on a new n-ary λ-averaging operator in the aggregation of similarities. This work is a natural extension of earlier research on similarity based classification in which aggregation is commonly performed by using the OWA-operator. So far λ-averaging has been used only in binary aggregation. Here the λ-averaging operator is extended to the n-ary aggregation case by using t-norms and t-conorms. We examine four different n-ary norms and test the new similarity classifier with five medical data sets. The new method seems to perform well when compared with the similarity classifier.

Keywords: similarity classifier with λ-averaging, n-ary λ-averaging operator, n-ary t-norm, n-ary t-conorm, classification.

1. Introduction

In this paper we present a new extension of the similarity based classifier, presented by Luukka *et al.* (2001) and Luukka (2005). The core idea of the similarity based classifier is to build ideal vectors of class representatives and use similarity in making the classification decision for the class of the sample. Similarity based classification was previously studied in several papers: different similarity measures in similarity classifiers were examined by Luukka (2007; 2008), while aggregation with OWA operators within the similarity classifier was studied by Luukka and Kurama (2013). Similarity based classification was also found to be useful in combination with using various principal component analysis (PCA) methods (Luukka, 2009; Luukka and Leppalampi, 2006) and with feature selection (Luukka, 2011). Similarity based classification was applied in a variety of classification problems, e.g., in classifying chromosomes (Sivaramakrishnan and Arun, 2014), in 3D face recognition (Ezghari *et al.*, 2015), and in freeway incident duration modeling (Vlahogianni and Karlaftis, 2013).

In this paper we examine how λ-averaging (Klir and Yuan, 1995) can be applied in place of other aggregation methods, such as the ordered weighted averaging operator (OWA), in aggregation of similarities in the similarity classifier. Note that λ-averaging (Klir and Yuan, 1995) has earlier been introduced only for binary aggregation due to the fact that it uses t-norms and t-conorms. Since we are dealing with associative functions, these can be extended to the n-ary case and in this way also λ-averaging can be used to aggregate vectors of n elements.

Originally, t-norms first appeared in Menger's work to generalize the classical triangle inequality in metric spaces (Menger, 1942). However, the current axioms of the t-norm and its dual t-conorm were modified in the context of probabilistic metric spaces for a binary case by Schweizer and Sklar (1960; 1983). Later, Hohle (1978), Alsina *et al.* (1983) and others introduced the t-norm and the t-conorm into fuzzy set theory and suggested that they could be used for the intersection and union of fuzzy sets, respectively. Due to their

*Corresponding author

associativity, t-norms and t-conorms were easily extended to the n-ary case, as suggested by Klement *et al.* (2003a; 2003b; 2000). These extensions were applied in several cases including the design and construction of kernels (Fengqiu and Xiaoping, 2012a; 2012b), and in neuro-fuzzy systems (Gabryel *et al*., 2010; Korytkowski and Scherer, 2010). Other application areas are found in the framework of aggregation operators and in the resolution and optimization of fuzzy relational equations (Saminger *et al*., 2007; Li and Fang, 2008).

Aggregation of information is very useful in classification, and it is often one of the required steps before reaching the decision making stage in data analysis. The concept of aggregation existed in the literature for some time (Dubois and Prade, 1985; 2004), with a variety of applications in knowledge based systems, such as decision making, pattern recognition, and machine learning, among others (Detyniecki, 2000). Basically, when faced with several values from different sources, an aggregation function fuses the separate values into a single outcome that can be used in the system or in supporting decision making. The simplest and most common way to aggregate (numerical) information is to use the arithmetic mean. However, there are also several other operations that have been used in aggregation, such as geometric, quadratic, or harmonic means. Experts have developed more specialized operators that guide aggregation, such as the use of the minimum t-norm, the maximum t-conorm, Łukasiewicz, product t-norms and t-conorms, averaging operators, and others (Calvo *et al*., 2002; Yager, 1988; O'Hagan, 1988; Xu, 2008; Schweizer and Sklar, 1960; 1983). For all these methods, aggregation is easier for the binary case, but higher dimensional cases have also been considered. Generally, we make the observation that aggregation operators have recently gained interest (Calvo *et al*., 2002).

Classification of objects is a well studied issue within artificial intelligence and it has many application opportunities in other fields, too. In classification, one is interested in partitioning the feature space into regions. Ideally, partitioning is done so that none of the decisions are ever wrong (Duda *et al*., 1973). The aggregation method that is used in the classification stage affects the accuracy of the classifier in one way or another. It is therefore important to design classifiers that utilize aggregation methods that are as accurate as possible, since even a small change in the classification accuracy may produce very meaningful effects in the application space (Klir and Folger, 1988). In this paper, aggregation is done using the n-ary lambda averaging operator that was previously defined for the binary case by Klir and Yuan (1995). As a new contribution, here we extend it to the n-ary case and use it in aggregation within the similarity classifier, thus presenting and proposing a new classification method.

This paper is organized as follows. In Section 2, we provide the mathematical background with the notions and definitions used in the paper. We start with presenting aggregation operators and then we introduce the new n-ary extension to λ-averaging. The mathematical background of similarity measures is also introduced. In Section 3, we start with a short introduction to the similarity classifier and present the proposed new λ-averaging based similarity classifier. In Section 4, the new similarity classifier is benchmarked with five different medical data sets and the results compared with those obtained with a standard similarity classifier. Finally, the paper is complemented with a discussion.

2. Mathematical background

In this section we first start with a short introduction of aggregation operators in general. Then we move into presenting t-norms and t-conorms and introduce also some new n-ary extensions to them. After this we present averaging operators, within which we focus especially on λ-averaging and start from the binary case. Then we introduce how the λ-average can be extended to the n-ary case with n-ary t-norms and t-conorms. We close this section with the introduction of the similarity measures that are used in the similarity classifier presented here.

2.1. Aggregation operators. Aggregation of fuzzy inputs is the process of combining several numerical values into a single value that is a representation of all the included values. In this case, an aggregation function, or an operator, is used to perform this operation. Definitions are presented, following the works of Klir and Yuan (1995), Detyniecki (2000), as well as Dubois and Prade (1985).

An n-ary aggregation operator is defined by Klir and Yuan (1995) as follows: An n-ary aggregation operator $(n \geq 2)$ is a mapping $f : [0,1]^n \to [0,1]$.

For $n = 2$, we obtain the usual binary case $f : [0,1]^2 \to [0,1]$. For example, if the aggregation operator f is applied to two fuzzy sets, say A_1, A_2, via their membership grades $A_1(x), A_2(x)$ to produce a single aggregated fuzzy set A, with a membership grade $A(x)$, where

$$A(x) = f(A_1(x), A_2(x)) \tag{1}$$

for all $x \in X$, we get the universe where all fuzzy sets are defined. We can extended this to an n-ary case. Suppose n fuzzy sets $A_1, A_2, \ldots, A_n$ defined on X are to be aggregated so that we get $A = f(A_1, A_2, \ldots, A_n)$. This can be done, since $f(A_1, A_2, \ldots, A_n)(x) = f(A_1(x), A_2(x), \ldots, A_n(x))$ means that a single membership grade $A(x)$ can be obtained from

$$A(x) = f(A_1(x), A_2(x), \ldots, A_n(x)). \tag{2}$$

The concept of aggregation requires the operator used to satisfy a number of properties. The “strength” of an aggregation operator may depend on which properties it satisfies but, basically, any aggregation operator should satisfy the following three properties (Klir and Yuan, 1995):

1. *Boundary conditions*
   An aggregation operator f on an interval $[0, 1]$ satisfies $f(0, 0, \ldots, 0) = 0$ and $f(1, 1, \ldots, 1) = 1$. This means that aggregation of small values returns a small value, and aggregation of large values returns a large value (Detyniecki, 2000).

2. *Monotonicity*
   Assume that $(x_1, x_2, x_3, \ldots, x_n) \in [0, 1]$ and let $(y_1, y_2, y_3, \ldots, y_n) \in [0, 1]$ be any pair of n-tuples, such that $x_i \leq y_i \ \forall i \in \mathbb{N}$; then we are guaranteed that $f(x_1, x_2, \ldots, x_n) \leq f(y_1, y_2, \ldots, y_n)$. This property ensures that the aggregated values are always increasing for any increasing set of objects.

3. *Continuity*
   The aggregation operator f is continuous on $[0, 1]$.

Continuity ensures that certain operations which would rather be complex are made possible.

Certain aggregation operators also satisfy symmetry and idempotency conditions. Notice that the symmetry property implies that interchanging arguments does not affect the aggregated value, and thus the aggregated fuzzy sets are treated with equal importance (Klir and Yuan, 1995).

In view of the above properties, it can be noted that there are several operators that satisfy the main conditions required from “true” aggregation operators. It needs to be said that for some operators that can generally be called aggregation operators the above properties are not fulfilled, especially when an extension of the operators is made from the binary case. Here, we focus on intersection (t-norm), union (t-conorm), and averaging operators. In the next subsections we review the basic idea behind the three above mentioned aggregation operators.

2.2. Intersection operator (t-norm). In crisp set theory, the intersection operation $\cap$ between two sets A and B is understood as the set represented by the region shared by both sets. Its fuzzy counterpart is a binary operation that takes in two membership grades $A(x), B(x) \in [0, 1]$ of A and B, respectively, yielding a single membership grade in $[0, 1]$. Thus, for intersection i on the two sets, $i : [0, 1]^2 \to [0, 1]$, we have

$$(A \cap B)(x) = i[A(x), B(x)] \tag{3}$$

for all $x \in X$ (Klir and Yuan, 1995). This can be generalized to any number of fuzzy sets without any loss of general properties. Following the work of Klir and Yuan (1995), we can define the t-norm as follows.

Definition 1. An aggregation operator $T : [0, 1]^2 \to [0, 1]$ is called a *t-norm* if it is commutative, associative, monotonic, and satisfies the boundary conditions. That is, for all $x, y, z \in [0, 1]$ we have that
$A1 :\ T(x, y) = T(y, x)$ (commutativity),
$A2 :\ T(x, T(y, z)) = T(T(x, y), z)$ (associativity),
$A3 :\ T(x, y) \leq T(x, z)$, whenever $y \leq z$ (monotonicity),
$A4 :\ T(x, 1) = x$ (boundary condition).

This works as a skeleton for a norm operator. We can also introduce further axioms to have even stricter forms. For example, for the Archimedean t-norm we can also require sub-idempotency and continuity (Klement *et al.*, 2003a).

Definition 2. A triangular norm, T, is said to be an *Archimedean norm* if it is continuous and $T(x, x) < x, \ \forall x \in [0, 1]$.

There are several examples of t-norms used in applications; take, for example, any $x, y \in [0, 1]$. The most commonly used t-norms include the following.

1. Standard intersection, T_M: $T_M(x, y) = \min(x, y)$. This takes in arguments x and y and returns $\min(x, y)$ as an output. This is the largest of the t-norm family that is considered in this article.

2. Algebraic product, T_P: $T_P(x, y) = xy$. Clearly, for all $x, y \in [0, 1]$, the algebraic product xy is in $[0, 1]$.

3. Łukasiewicz t-norm, T_L: $T_L(x, y) = \max(0, x + y - 1)$. This is also called the bounded difference.

4. Drastic intersection, T_D:

$$T_D(x, y) = \begin{cases} x, & \text{if } y = 1, \\ y, & \text{if } x = 1, \\ 0, & \text{otherwise.} \end{cases}$$

The drastic t-norm is the smallest or is at the extreme end in the family of four t-norms mentioned here. Clearly, the value obtained by using an intersection aggregation operator depends on which type of t-norm is used.

Due to the associativity of t-norms, it is possible to extend the operation to the n-ary case, $n \geq 2$. For $n = 3$, the t-norm T can be computed from $T(x_1, x_2, x_3) = T(T(x_1, x_2), x_3)$. This was shown by Klement *et al.* (2003b, p. 415), who gave the following definition.

Definition 3. Let T be a t-norm and $(x_1, x_2, \ldots, x_n) \in [0, 1]^n$ be any n-ary tuple. We define $T(x_1, x_2, \ldots, x_n)$ as

$$T(x_1, x_2, \ldots, x_n) = T(T(x_1, x_2, \ldots, x_{n-1}), x_n). \tag{4}$$

Equation (4) provides a recursive procedure that is very useful for any n-ary case. We can use this to derive a direct formula in some cases (see, e.g., Definitions 4–6), but in some cases these need to be computed using recursion, as given in Definition 3. Next we go through some n-ary t-norms that we arrive at in this way and that are applied later in the aggregation.

Definition 4. The standard intersection, T_M, can be obtained for all $(x_1, x_2, \ldots, x_n) \in [0, 1]^n$ by

$$T_M(x_1, x_2, x_3, \ldots, x_n) = \min(x_1, x_2, x_3, \ldots, x_n) \quad (5)$$

Note. In our application, input vectors are the ideal vectors obtained from a data matrix. Each ideal vector is taken one at a time by the algorithm and aggregated to obtain a value that is compared with the λ value. Also, the following definitions, which we are applying later in λ-averaging are given by Klement *et al.* (2003b).

Definition 5. Let T_P be the algebraic product, also called the probabilistic t-norm. For any n-ary input vector $(x_1, x_2, x_3, \ldots, x_n)$ we define T_P as

$$T_P(x_1, x_2, x_3, \ldots, x_n) = \prod_{k=1}^{n} (x_k). \quad (6)$$

Definition 6. Let T_L be the Łukasiewicz t-norm. For any n-ary vector $(x_1, x_2, x_3, \ldots, x_n)$, T_L is extended by

$$T_L(x_1, x_2, x_3, \ldots, x_n) = \max[0, (1 - \sum_{k=1}^{n} (1 - x_k))]. \quad (7)$$

Definition 7. Let T_D be the drastic product t-norm. The n-ary extension for all $x_i, x_{i+1},\ i = 1, 2, 3, \ldots, n - 1$, is computed from

$$T_D(x_1, x_2, x_3, \ldots, x_n) = \begin{cases} x_i, & \text{if } x_{i+1} = 1, \\ x_{i+1}, & \text{if } x_i = 1, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

The above process of implementing $T_D(x_1, x_2, x_3, \ldots, x_n)$ goes forward so that first two arguments are considered; if one of them is 1, then the other one is picked. If none of them is 1, a zero is selected instead and compared with the next argument. The process is repeated until all the x_i's are considered.

Next, we present a simple example that demonstrates the implementation of the 8-ary argument using the four different t-norm aggregation operators.

Example 1. Consider an 8-ary vector,

$$h = (0.4, 0.2, 0.5, 0.1, 0.3, 1.0, 0.4, 0.8).$$

We compute the standard intersection T_M, the algebraic product T_P, Łukasiewicz t-norm T_L and the drastic product T_D as

$$\begin{aligned}
T_M(h) &= \min(0.4, 0.2, 0.5, 0.1, 0.3, 1.0, 0.4, 0.8) = 0.1, \\
T_P(h) &= \prod (0.4, 0.2, 0.5, 0.1, 0.3, 1.0, 0.4, 0.8) \\
&= 0.000384, \\
T_L(h) &= \max[1 - \sum((1, 1, 1, 1, 1, 1, 1, 1) \\
&\quad - (0.4, 0.2, 0.5, 0.1, 0.3, 1.0, 0.4, 0.8)), 0] \\
&= \max[1 - \sum(0.6, 0.8, 0.5, 0.9, \\
&\quad 0.7, 0.0, 0.6, 0.2), 0] \\
&= \max[1 - (4.3), 0] = 0, \\
T_D(h) &= 0.
\end{aligned}$$

This is because

$$T_D(0.4, 0.2) = 0, T_D(T_D(0.4, 0.2), 0.5) = 0,$$

$$T_D(T_D(0.4, 0.2, 0.5), 0.1) = 0,$$

$$T_D(T_D(0.4, 0.2, 0.5, 0.1), 0.3) = 0,$$

$$T_D(T_D(0.4, 0.2, 0.5, 0.1, 0.3), 1.0) = 1,$$

$$T_D(T_D(0.4, 0.2, 0.5, 0.1, 0.3, 1.0), 0.4) = 0,$$

$$T_D(T_D(0.4, 0.2, 0.5, 0.1, 0.3, 1.0, 0.4), 0.8) = 0.$$

♦

2.3. Union operator (t-conorm).

Another key part of the λ-averaging operator is a union operator, which is referred to as the t-conorm. The t-co norm is an aggregation operator that can be considered to be dual to the t-norm. Given two sets A and B, the fuzzy union u is a function $u : [0, 1]^2 \to [0, 1]$ such that

$$(A \cup B)(x) = u[A(x), B(x)]. \quad (9)$$

Inputs are the two membership grades, one from A and another from B, which gives one output from $A \cup B$ (Klir and Yuan, 1995). Given any t-norm, T, a t-co norm, S, can be obtained using the fact that (Klement *et al.*, 2003b)

$$S(x, y) = 1 - T(1 - x, 1 - y) \quad (10)$$

for all $x, y \in [0, 1]$.

Definition 8. An aggregation operator $S : [0, 1]^2 \to [0, 1]$ is called a triangular conorm (t-conorm) if it is commutative, associative, monotone, and has 0 as its neutral element (Klir and Yuan, 1995). That is, for all $x, y, z \in [0, 1]$, we have the following axioms satisfied:
$A1 :\ S(x, y) = S(y, x)$ (commutativity),
$A2 :\ S(x, S(y, z)) = S(S(x, y), z)$ (associativity),
$A3 :\ S(x, y) \leq S(x, z)$, whenever $y \leq z$ (monotonicity),
$A4 :\ S(x, 0) = x$ (boundary condition).

These axioms are general and quite basic, more strict axioms have been defined for triangular conorms (Klir and Yuan, 1995), making them applicable in a number of areas.

The following are common examples of t-conorms implemented in this paper. For all $x, y \in [0, 1]$, we have

1. standard union, S_M: $S_M(x, y) = \max(x, y)$,
2. algebraic sum, S_P: $S_P(x, y) = x + y - xy$,
3. Łukasiewicz t-conorm, S_L: $S_L(x, y) = \min(1, x + y)$,
4. drastic union, S_D:

$$S_D(x, y) = \begin{cases} x, & \text{if } y = 0, \\ y, & \text{if } x = 0, \\ 1, & \text{otherwise.} \end{cases}$$

Triangular conorms can be extended to n-ary arguments (Klement *et al.*, 2003b; 2003a) due to their associativity. In the work of Klement *et al.* (2003b), general constructions of n-ary t-norms and t-conorms were presented. The following definitions are given, based on extensions proposed by Klement *et al.* (2003b; 2000).

Definition 9. Let S be a t-conorm and $(x_1, x_2, \dots, x_n) \in [0, 1]^n$ be any n-ary tuple. Then $S(x_1, x_2, \dots, x_n)$ is given by

$$S(x_1, x_2, \dots, x_n) = S(S(x_1, x_2, \dots, x_{n-1}), x_n) \quad (11)$$

Definition 10. Let $(x_1, x_2, \dots, x_n) \in [0, 1]^n$ be an n-ary vector. Then the standard union can be extended using

$$S_M(x_1, x_2, \dots, x_n) = \max(x_1, x_2, \dots, x_n). \quad (12)$$

The standard union aggregation operator is the smallest of the t-conorm family mentioned above. Thus it is the only one that is idempotent (Klir and Yuan, 1995). Due to the commutative property, any order of pairwise groupings can be computed for an n-ary vector. Next we shortly go through three other n-ary t-conorms.

Definition 11. The *extended probabilistic t-conorm* is given by

$$S_P(x_1, x_2, x_3, \dots, x_n) = 1 - \prod_{k=1}^{n} (1 - x_k). \quad (13)$$

This operation is clearly giving higher values than the standard union operation, so we have

$$S_M(x_1, x_2, x_3, \dots, x_n) \le S_P(x_1, x_2, x_3, \dots, x_n)$$

for all $x_1, x_2, x_3, \dots, x_n \in [0, 1]$.

Definition 12. The n-ary Łukasiewicz t-conorm is given by

$$S_L(x_1, x_2, \dots, x_n) = \min[1, \sum_{i=1}^{n} x_i]. \quad (14)$$

This operation is again giving higher values than previously and we have $S_P(x_1, x_2, x_3, \dots, x_n) \le S_L(x_1, x_2, x_3, \dots, x_n)$.

Lastly, the largest of the t-conorms is the drastic union.

Definition 13. Let S_D be the drastic sum. For all $x_i, x_{i+1} \in [0, 1],\ i = 1, 2, 3, \dots, n-1$, we have

$$S_D(x_1, x_2, x_3, \dots, x_n) = \begin{cases} x_i, & \text{if } x_{i+1} = 0, \\ x_{i+1}, & \text{if } x_i = 0, \\ 1, & \text{otherwise.} \end{cases} \quad (15)$$

It can be shown that for any vector $t = (x_1, x_2, x_3, \dots, x_n) \in [0, 1]^n$ we have $S_M(t) \le S_P(t) \le S_L(t) \le S_D(t)$.

Next, a simple example is used to show the implementation of the union aggregation operators for an n-ary vector.

Example 2. Consider a 5-ary vector

$$t = (0.6, 0.1, 0.5, 0.0, 0.8) \in [0, 1]^5.$$

To aggregate the vector using the union operator, we obtain standard union,

$$S_M(t) = \max(t) = 0.8,$$

algebraic sum,

$$\begin{aligned} S_P(t) = 1 - \Big(\prod [(1, 1, 1, 1, 1) \\ - (0.6, 0.1, 0.5, 0.0, 0.8)]\Big) \\ = 0.964, \end{aligned}$$

Łukasiewicz t-conorm,

$$S_L(t) = \min[1, \sum (0.6, 0.1, 0.5, 0.0, 0.8)] = 1,$$

and the drastic union,

$$S_D(t) = 1$$

♦

2.4. Averaging operators. Apart from the intersection and the union operators, another class of aggregating operators that are monotonic and idempotent are what is called averaging operators. The following axiomatic properties are required for an operator to be an *averaging* one.

Definition 14. Let $a, b \in [0,1]^n$ be column vectors such that $a = [a_1, \ldots, a_n]^T$, $b = [b_1, \ldots, b_n]^T$ and $\overline{a} = [a, \ldots, a]^T$. An aggregation operator $h : [0,1]^n \to [0,1]$ is an *averaging* one if

1. $h(\overline{a}) = a, \forall\, a \in [0,1]$ (idempotency);
2. $h(a) = h(a_\pi)$, where π means any permutation on $\{1, \ldots, n\}$ and $a_\pi = [a_{\pi(1)}, \ldots, a_{\pi(n)}]^T$ (symmetry on all its arguments);
3. $h(0) = 0,\ h(1) = 1$ (boundary conditions);
4. for any pair $a, b \in [0,1]^n$, if $a_i \leq b_i$ for all $i \in \{1, \ldots, n\}$, $h(a) \leq h(b)$ (monotonic in all its arguments);
5. h is continuous.

Averaging operators usually "occupy" the interval between the intersection and the union. Due to the monotonicity condition, an averaging operator h, normally satisfies

$$\begin{aligned}\min(x_1, x_2, \ldots, x_n) &\leq h(x_1, x_2, \ldots, x_n) \\ &\leq \max(x_1, x_2, \ldots, x_n) \qquad (16)\end{aligned}$$

for all $x_i \in [0,1],\ i = 1, \ldots, n$.

In general, there are several kinds of averaging operators that can be used in aggregation. One of the most commonly employed averaging operators is the generalized mean. The generalized mean operator "covers" the whole interval between the minimum (intersection) and the maximum (union).

Definition 15. Let $x_1, x_2, \ldots, x_n$ be an n-ary vector. The generalized mean aggregation operator h is given by

$$h_p(x_1, x_2, \ldots, x_n) = \left(\frac{1}{n} \sum_{i=1}^{n} (x_i)^p \right)^{1/p}, \qquad (17)$$

where $p \neq 0 \in \mathbb{R}$ is a parameter by which several means are differentiated.

For example, if $p = 1$, we obtain the arithmetic mean given by

$$h_1(x_1, x_2, \ldots, x_n) = \frac{1}{n} \sum_{i=1}^{n} x_i, \qquad (18)$$

and if $p = -1$, we obtain the harmonic mean given by

$$h_{-1}(x_1, x_2, \ldots, x_n) = \frac{n}{\sum\limits_{i=1}^{n} \dfrac{1}{x_i}}. \qquad (19)$$

Another averaging operator that is often used in the literature is the ordered weighted averaging (OWA) one, introduced by Yager (1988). This averaging operator was used for classification purposes with(in) the similarity classifier by Luukka and Kurama (2013). The OWA operator is characterized by an adjustable weighting vector. The adjustment of the weights in the vector allows the averaging operator to "move" between the minimum and the maximum.

Definition 16. A mapping $g : \mathbb{R}^n \to \mathbb{R}$, with an associated vector $w = (w_1, w_2, w_3, \ldots, w_n)^T$, $w_i \in [0,1], i = 1, 2, \ldots, n$ is an ordered weighted averaging operator if for $\sum_{i=1}^{n} w_i = 1$ and $(a_1, a_2, \ldots, a_n) \in \mathbb{R}^n$ we have

$$g(a_1, a_2, a_3, \ldots, a_n) = \sum_{i=1}^{n} w_i b_i, \qquad (20)$$

where b_i is the i-th largest of the elements $a_1, a_2, \ldots, a_n$ arranged in descending order, (Yager, 1988).

2.5. λ-averaging operator and its extension to the n-ary case. Here we first introduce the λ-averaging operator as given by Klir and Yuan (1995, p. 93), and after this we are going to present the new n-ary generalization of the operator.

Definition 17. A *lambda averaging operator* (λ-average) is a parameterized class of norm operations defined for a binary case by

$$h_\lambda(x, y) = \begin{cases} \min(\lambda, S(x, y)), & \text{if } x, y \in [0, \lambda], \\ \max(\lambda, T(x, y)), & \text{if } x, y \in [\lambda, 1], \\ \lambda, & \text{otherwise}, \end{cases} \qquad (21)$$

for all $x, y \in [0,1]$ and $\lambda \in (0,1)$, where T is a t-norm and S is a t-conorm.

The value of λ is essential in the averaging process, since the intervals $[0, \lambda]$ and $[\lambda, 1]$ are central to the resulting aggregated value. $T(x, y)$ and $S(x, y)$ can basically be any t-norm or t-conorm. The λ-averaging operator satisfies all of the above discussed properties of aggregation operators, but the boundary conditions are "weaker". The usual boundary conditions are replaced by $h(0,0) = 0$ and $h(1,1) = 1$. The properties of continuity and idempotency are satisfied. Accordingly, this class of operators can reduce to t-norms if $h(x, 1) = x$ and to t-conorm if $h(x, 0) = x$; thus the whole range from drastic intersection T_D to drastic union S_D is covered by the λ averaging operator.

Since the λ-averaging operator is an associative operator, it can be extended to the n-ary case in the same way as can be done for general t-norms and t-conorms.

Definition 18. For any n-tuple, $t = (x_1, x_2, \ldots, x_n) \in [0,1]^n$, we define the n-ary lambda averaging operator,

$h_\lambda(t)$, by

$$h_\lambda(t) = \begin{cases} \min(\lambda, S(t)), & \text{if } t \in [0,\lambda]^n, \\ \max(\lambda, T(t)), & \text{if } t \in [\lambda,1]^n, \\ \lambda, & \text{otherwise,} \end{cases} \tag{22}$$

where the t-norm $T(t)$ and t-conorm $S(t)$ can be recursively computed from any n-ary t-norm/t-conorm.

Implementation of the n-ary λ- averaging operator is done via Eqn. (22). The next example briefly illustrates how the new extension can be applied when generalized versions of the standard t-norm and t-conorm are chosen for the n-ary t-norms and t-conorms.

Example 3. Suppose that a 4-ary vector

$$t = (0.2, 0.5, 0.4, 0.3) \in [0,1]^4$$

is to be aggregated using the lambda averaging operator for $\lambda = 0.6 \in [0,1]$. Since

$$t = (0.2, 0.5, 0.4, 0.3) \in [0,\lambda]^4,$$

we obtain

$$h_\lambda(t) = \min(\lambda, S(t)).$$

For

$$S(t) = S_M(t) = \max(t) = 0.5$$

we get

$$h_\lambda(t) = \min(0.6, 0.5) = 0.5.$$

♦

Different values of $h_\lambda(t)$ are obtained if we use the t-conorms $S_P(t), S_L(t)$ and $S_D(t)$ in the lambda averaging operator.

2.6. Similarity measures. The similarity measure based on the Łukasiewicz structure (Łukasiewicz, 1970) is the one that we use in our similarity classifier. One reason for this selection is that it is the most used similarity measure in examining similarity classifiers, and hence it is well studied in the literature (Luukka, 2007; 2005). One of the advantages of using a similarity measure in the Łukasiewicz structure is that the mean of many similarities is still a similarity, as shown by Turunen (2002). This similarity measure also works well in comparing objects. Next, we shortly go through the most important definitions of similarity and finally present the similarity measure which we are using. In the work of Mattila (2002), the following definition was given.

Definition 19. Let $\mu_S(x,y)$ be the degree of membership of the ordered pair (x,y). A fuzzy relation S on a set X is called a *similarity relation* if it is reflexive, symmetric, and transitive, i.e.,

1. for all $x \in X,\ \mu_S(x,x) = 1$ (reflexive),
2. for all $x, y \in X,\ \mu_S(x,y) = \mu_S(y,x)$ (symmetric),
3. for all $x, y, z \in X,\ \mu_S(x,z) \geq \mu_S(x,y) * \mu_S(y,z)$, where $*$ is a binary operation (transitive).

Definition 20. A fuzzy binary relation that is reflexive, symmetric, and transitive is known as a *fuzzy equivalence relation*, or as a similarity relation.

Definition 21. In the Łukasiewicz structure, we define the *Łukasiewicz norm* as

$$x \circledast y = \max[x + y - 1, 0] \tag{23}$$

with an implication $x \to y = \min[1, 1 - x + y]$.

This norm, together with the implication, provides a basis for the definition of the Łukasiewicz structure.

Definition 22. In the *Łukasiewicz structure*, we define the similarity relation $x \Leftrightarrow y$ as

$$x \Leftrightarrow y = 1 - |x - y|. \tag{24}$$

Definition 23. The *generalized Łukasiewicz structure* takes the form

$$x \circledast y = \sqrt[p]{\max\{x^p + y^p - 1, 0\}}, \quad p \in [1, \infty], \tag{25}$$

with the implication $x \to y = \min\{1, \sqrt[p]{1 - x^p + y^p}\}$, where p, a fixed integer, is a parameter in the Łukasiewicz structure.

3. New similarity based n-ary lambda classifier

The new similarity measure based classifier that uses the n-ary lambda extension of the lambda averaging operator is introduced here. To be precise, the new classification method is based on the extension of the lambda averaging operator presented by Klir and Yuan (1995) and on the similarity classifier proposed by Luukka and Leppalampi (2006). The new n-ary lambda operator is used in the aggregation stage of the classification, after a vector of similarities has been calculated. First we give a brief description of how the similarity classifier based on Łukasiewicz structure works.

3.1. Similarity based classifier. In this context, a similarity is viewed as a numerical measure of how similar data sets, or vectors, are in a matrix. Thus, the higher the similarity value, the closer (the more similar) the objects in terms of characteristics. The major task of classification is to partition attributes (features) into regions that categorize the date with the best accuracy. Ideally, one would like to arrange the partitions so that none of the decisions is ever wrong (Duda *et al.*, 1973).

Suppose that we want to classify a set Y of objects into M different classes $C_1, C_2, \ldots, C_M$ by their attributes. Let n be the number of different features $f_1, \dots, f_n$ that can be measured for the given objects. To preserve the fuzzy domain $[0, 1]$, the values of each attribute are normalized, so that the objects to be classified are vectors in the interval $[0, 1]^n$. Each of the classes $C_1, C_2, \dots, C_M$ is represented by an ideal vector $\mathbf{v}_i = (v_i(f_1), \dots, v_i(f_n))$.

First, we must determine an ideal vector for each class $\mathbf{v}_i = (v_i(f_1), \dots, v_i(f_n))$ that acts as a representation of the class i. This vector can be user defined, or calculated from a sample set X_i of vectors $\mathbf{x} = (x(f_1), \dots, x(f_n))$ which are known to belong to a given class C_i. The method requires the user to have some knowledge about what kind of classes exist; the better one knows the better the results will be. We can, e.g., use the generalized mean as aggregation operator for calculating $\mathbf{v}_i$, which is

$$v_i(r) = \left(\frac{1}{\sharp X_i} \sum_{\mathbf{x} \in X_i} x(f_r)^m \right)^{\frac{1}{m}}, \quad \forall r = 1, \dots, n, \tag{26}$$

where power m (coming from the generalized mean) is fixed for all i, r and $\sharp X_i$ simply denotes the number of samples in the class i.

After the ideal vectors have been determined, the decision to which class an arbitrarily chosen $\mathbf{x} \in X$ belongs is made by comparing it with each ideal vector. The comparison can be done, e.g., by using a similarity measure in the generalized Łukasiewicz structure:

$$S\langle \mathbf{x}, \mathbf{v} \rangle = \left(\frac{1}{n} \sum_{r=1}^{n} w_r \left(1 - |x(f_r)^p - v(f_r)^p|\right)^{m/p} \right)^{1/m}, \tag{27}$$

for $\mathbf{x}, \mathbf{v} \in [0, 1]^n$. Here p is a parameter coming from the generalized Łukasiewicz structure (Luukka *et al.*, 2001).

3.2. λ-averaging based similarity classifier. The idea with the λ-averaging based similarity classifier is to "replace" any previously used aggregation operator with the λ-averaging operator. If we assume $w_r = 1, \quad \forall r$ and we remove the averaging operator

$$h_m(s_1, s_2, \dots, s_n) = \left(\frac{1}{n} \sum_{i=1}^{n} (s_i)^m \right)^{1/m}$$

from (27), we are left with a vector of similarities $(s_1, s_2, \dots, s_n)$, where

$$s_i = \left(1 - |x(f_i)^p - v(f_i)^p|\right)^{1/p}.$$

This similarity vector is now applied to our λ-averaging presented in Eqn. (22). This leads to gaining overall similarity as, for $i = 1, 2, 3, \dots, n$, we have

$$S_\lambda(s_i, \lambda) = \begin{cases} \min(\lambda, S(s_i)), & \text{if } s_i \in [0, \lambda]^n, \\ \max(\lambda, T(s_i)), & \text{if } s_i \in [\lambda, 1]^n, \\ \lambda, & \text{otherwise}, \end{cases} \tag{28}$$

where $S(s_i)$ and $T(s_i)$ are a t-conorm and a t-norm, respectively. From this operation, a single similarity value is obtained for each class.

The decision to which class an arbitrarily chosen object $\mathbf{y} \in Y$ belongs is made by comparing it to the aggregated value $S_\lambda(s_1, s_2, s_3, \dots, s_n, \lambda)$. The object belongs to class with the highest similarity value as computed from $\max_{i=1,2,\dots,M} S_\lambda(\mathbf{x}, \mathbf{v_i})$

The λ-averaging based similarity classifier is described using the pseudo-code in Algorithm 1 below. In a (t–norm, t-conorm) pair one can apply any of the four different t–norms, T_M, T_L, T_P, T_D, in aggregation, and their corresponding t-conorms, S_M, S_L, S_P, S_D, each at a time depending on the choice.

Algorithm 1. Pseudo code for the similarity classifier with an n-ary lambda averaging operator.

Require: $data[1, \dots, N], ideals, \lambda, p$
for $i = 1$ to N **do**
  for $j = 1$ to M **do**
    for $k = 1$ to L **do**
      $S(i, j, k) = [1 - |(data(i, j))^p - (ideal(k, j))^p|]^{\frac{1}{p}}$
    end for
  end for
end for
for $i = 1$ to N **do**
  for $k = 1$ to L **do**
    if $\max(S(i, :, k)) \leq \lambda$ **then**
      $S_\lambda(i, k) = \min(tnorm(S(i, :, k)), \lambda)$
    else if $\min(S(i, :, k)) \geq \lambda$ **then**
      $S_\lambda(i, k) = \max(tconorm(S(i, :, k)), \lambda)$
    else
      $S_\lambda(i, k) = \lambda$
    end if
  end for
end for
for $i = 1$ to n **do**
  $class(:, i) = find(S_\lambda(i, :) == \max(S_\lambda(i, :)))$
end for

4. Testing the new method: Data sets used and classification results

To test the new classification method, we run tests on five data sets of medical data. First, we shortly present the data sets that are used for the tests, and then we move on to present the classification results.

Table 1. Data sets used and their properties.

Data set	Number of classes	Number of attributes	Number of instances
Fertility	2	10	100
Liver disorder	2	7	345
Sick	2	29	3772
Hypothyroid1	2	25	3772
Pima Indians	2	8	768

4.1. Data sets used. Five different data sets were used to check the performance of our new classifier. The chosen data sets have different numbers of attributes and instances as shown in Table 1. The data sets used were obtained from the UCI machine learning data repository (Newman *et al.*, 2012), which contains free databases meant for research.

4.1.1. Data set 1: The fertility data set. The set is a newly donated database by David Gil, who has previously utilized it (Gil *et al.*, 2012). The main focus of study in this data set is on sperm concentration, which consequently influences fertility. In its collection, 100 volunteers provided semen samples that were analyzed according to the WHO 2010 criteria. It is believed that sperm concentrations are related to socio-demographic data, environmental factors, health status, and life habits. These factors were studied based on (i) the season in which the analysis was performed, (ii) the age of the donor at the time of analysis, (iii) Child-age diseases experienced–chicken pox, measles, mumps, and polio, (iv) accident or serious traumas incurred, (v) the surgical intervention done, (vi) the high fevers experience, (vii) the frequency of alcohol consumption, (viii) smoking and (ix) the number of hours spent sitting per day. The data set has 2 classes, 10 attributes including the class attribute, and 100 observations.

4.1.2. Data set 2: The liver disorder data set. The data set is provided by R.S. Forsyth (Newman *et al.*, 2012). The problem implied by the set is to predict whether a male patient has a liver disorder or not, based on blood test results and information about alcohol consumption. The attributes included are (i) mean corpuscular volume (ii) alkaline phosphotase, (iii) alamine aminotransferase, (iv) aspartate aminotransferase, (v) gamma-glutamyl transpeptidase, and (vi) the number of half-pint equivalents of alcoholic beverages drunk per day. The first five variables are results from blood tests, which are thought to be sensitive to liver disorders that might arise from excessive alcohol consumption. Each line in the liver disorder data file constitutes a record of a single male individual. The data set has 2 classes, 7 attributes including the class attribute, and 345 observations.

4.1.3. Data sets 3 and 4: The thyroid data sets. Two of the 6 thyroid data sets in the UCI (Newman *et al.*, 2012) were used, both including 3772 observations: (1) the sick data set that consists of 29 attributes and two classes and (2) hypothyroid1 that also has 29 attributes of which 25, including the class attribute, were used in the classification. The problem to be solved based on these two data sets is to discover, whether or not the patient has a thyroid related disease.

4.1.4. Data set 5: Pima Indians diabetes. The aim of the data set is to test the prevalence of diabetes among Indians of the Pima heritage. In particular, all patients were females of at least 21 years of age. The data set was donated by Vincent Sigillito from Johns Hopkins University, and it has 8 attributes with 768 observations. The attributes considered are (I) the number of times pregnant, (II) plasma glucose concentration—a 2 hour in an oral glucose tolerance test, (III) diastolic blood pressure (mm Hg), (IV) triceps skin fold thickness (mm), (V) 2-hour serum insulin (mu U/ml), (VI) body mass index (weight in kg/(height in m)^2), (VII) the diabetes pedigree function, (VIII) the age (years), and the class variable.

4.2. Classification results. We computed the results using the new n-ary λ similarity classifier. Since we are using four different types of n-ary norm in our λ-averaging, we refer to them simply by using the names of the norms. We benchmark the obtained results by comparing them not only to one another, but also to a similarity classifier that uses the generalized mean for aggregation (we call this the "standard similarity classifier"). Besides classification accuracy and classification variance, we also computed the area under receiving operator characteristic (AUROC) values. Also the receiving operator characteristic (ROC) curve is computed. In all experiments, the data sets were split into two parts; one half was used for training and the other for testing. This procedure was repeated by using random cut-points for 30 times, and the mean classification accuracies, variances, and AUROC values were computed. Corresponding figures are made to allow graphical inspection of how changing parameter values affects the classification results. In Figs. 1–4, the parameters which are varied are p, the value from the generalized Łukasiewicz structure, and λ from the λ-averaging operator.

4.2.1. Classification results with the fertility data set. Results obtained with the fertility data set for four different norms are recorded in Table 2. The best classification accuracy of 88.07% was obtained with the standard t-norm and the t-conorm. This represents

Table 2. Classification results with the fertility data set.

Method	Mean accuracy (%)	Variance	AUROC
λ-average classifier with:			
standard norms	**88.07**	1.3333×10^{-5}	0.5000
Łukasiewicz norms	87.71	1.5314×10^{-4}	0.5000
probabilistic norms	87.63	2.0709×10^{-4}	0.5000
drastic norms	88.00	5.1004×10^{-32}	0.5000
standard similarity classifier	68.20	0.0094	**0.7088**

Table 3. Classification results with the sick data set.

Method	Mean accuracy (%)	Variance	AUROC
λ-average classifier with:			
standard norms	94.04	7.0887×10^{-7}	**0.8523**
Łukasiewicz norms	**94.16**	1.2473×10^{-6}	0.5065
probabilistic norms	94.06	4.3401×10^{-6}	0.6997
drastic norms	93.85	5.1004×10^{-32}	0.5000
standard similarity classifier	72.61	0.0025	0.8212

an improvement of 19.87% over the standard similarity classifier (68.20% accuracy). Generally, using the other studied norms produced results that are "close to each other", but better than the results obtained with the standard similarity classifier. On the other hand, it can be noted that the standard similarity classifier produced the highest AUROC value of 70.88%. Variances obtained are very small, which indicates that true classification accuracies are close to the mean classification accuracy obtained.

Figure 1 shows mean classification accuracies and their corresponding variances. A combined plot of receiving operator characteristics for all the five data sets is presented in Fig. 6.

4.2.2. Classification results with the sick data set. With this data set the performance of Łukasiewicz norms was the best, with a mean classification accuracy of 94.16%. This can bee seen in Table 3. The accuracy obtained is very close to that produced using probabilistic norms, and is not very different from the mean accuracies obtained with standard n-ary norms and with drastic n-ary norms. The standard similarity measure has a lower mean classification accuracy of 72.61%. The improvement in performance is 21.55% for this particular data set. We also observe that the standard norms have the highest AUROC value.

In Fig. 2 the best mean classification accuracy (one

Table 4. Classification results with the hypothyroid data set.

Method	Mean accuracy (%)	Variance	AUROC
λ-average classifier with:			
standard norms	99.51	2.1880×10^{-6}	0.7088
Łukasiewicz norms	86.13	2.3502×10^{-4}	0.5000
probabilistic norms	99.27	2.0281×10^{-6}	0.5000
drastic norms	08.22	1.7931×10^{-33}	0.5000
standard similarity classifier	**99.61**	1.3613×10^{-6}	**0.9747**

obtained with Łukasiewicz norms) for the sick data set and its corresponding variance is shown. A plot that corresponds with the highest AUROC value is also presented in Fig. 6. It can be seen that the largest AUROC value was 85.23%, obtained by using standard norms.

4.2.3. Classification results with the hypothyroid data set. The standard similarity classifier achieved the highest mean classification accuracy of 99.61%, but the proposed methods also produced good results with this data set. In Table 4, mean classification accuracies of 99.51% and 99.27% were achieved by using the standard and probabilistic norms, respectively. There is no large difference in the general mean performance of the three methods, apart from the drastic norms, which performed poorly with this data set.

In Fig. 3 we also present plots with respect to parameter changes for the best mean classification accuracy and the variances.

4.2.4. Classification results with the liver disorder data set. In Table 5, results obtained with the liver disorder data set are presented. The standard n-ary norms produced the highest mean classification accuracy value of 60.29%, and the drastic norms produced the lowest accuracy of 42.20%. The standard similarity measure this time outperformed the Łukasiewicz, probabilistic and drastic n-ary norms for this data set. The standard similarity classifier performance was close to the best n-ary lambda classifier results with a classification accuracy of 59.92%.

In Fig. 4 the best mean classification accuracies and corresponding variances are presented with varying parameter values.

4.2.5. Classification results with the Pima Indians data set. With this data set, the probabilistic and Łukasiewicz norms have the mean classification accuracies of 74.18% and 74.13%, and clearly outperform the standard and the drastic norms. The standard

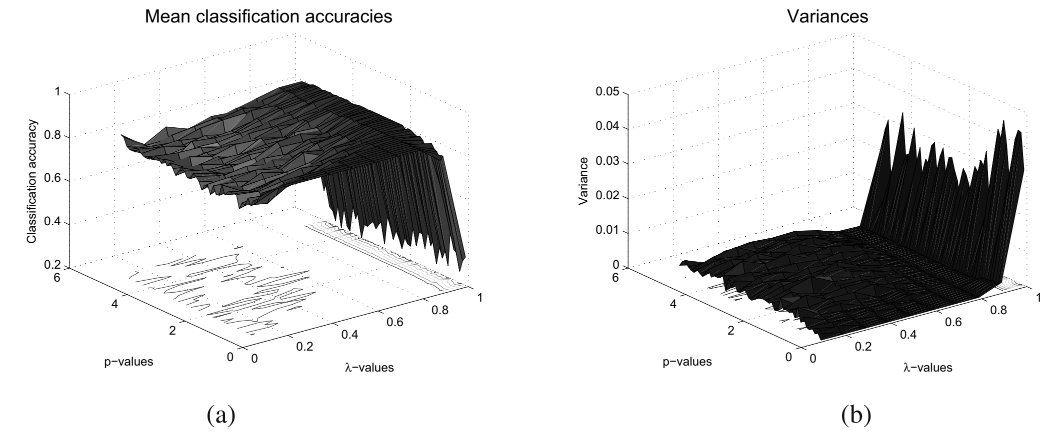


Fig. 1. Mean classification accuracies (a) and variances (b) obtained from the fertility data set with the use of the standard t-norm and t-conorm.

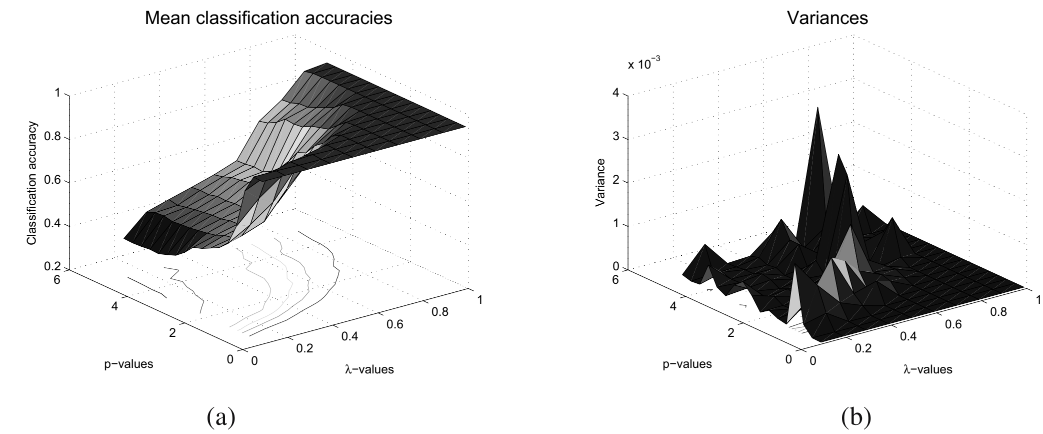


Fig. 2. Mean classification accuracies (a) and variances (b) obtained from the sick data set with the use of the Łukasiewicz t-norm and t-conorm.

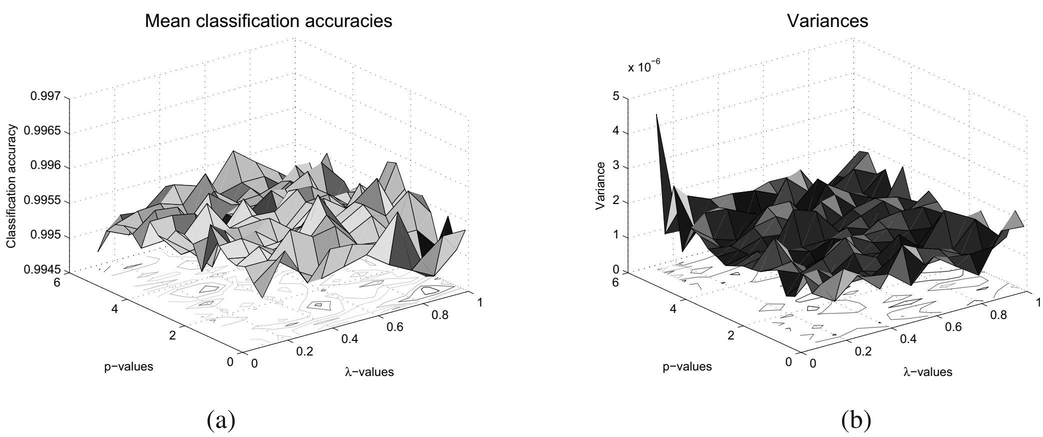
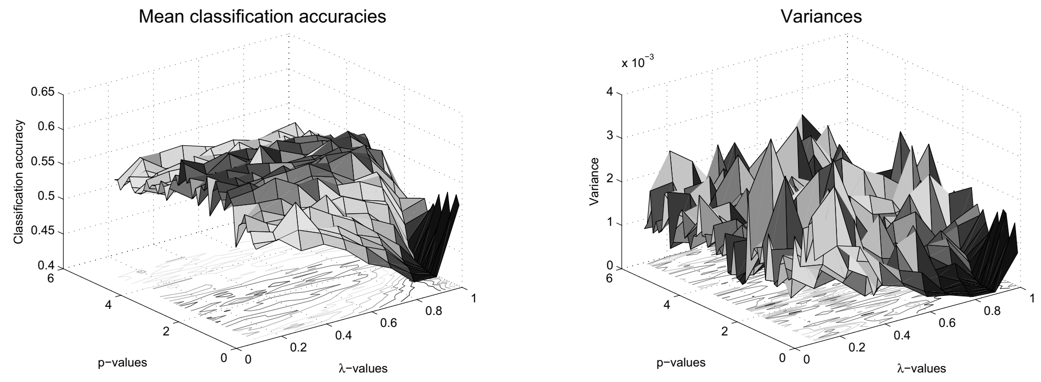


Fig. 3. Mean classification accuracies (a) and variances (b) obtained from the hypothyroid data set using the standard t-norm and t-conorm.

similarity classifier returns the best mean classification accuracy of 74.70%. In Table 6 the results are presented in detail.

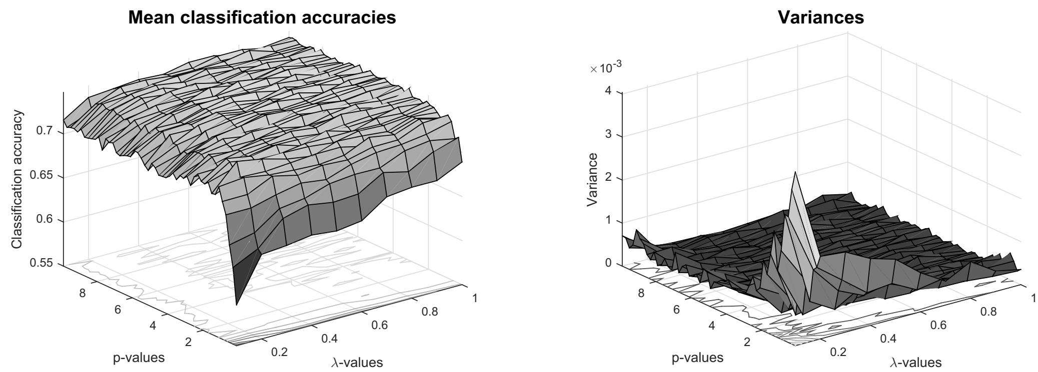
In Fig. 5 one can again see the mean accuracies and variance changes with respect to the parameter values.

The highest AUROC values for all the five data sets are plotted in Fig. 6.



(a) (b)

Fig. 4. Mean classification accuracies (a) and variances (b) obtained from the liver disorder data set using the standard t-norm and t-conorm.



(a) (b)

Fig. 5. Mean classification accuracies (a) and variances (b) obtained from the Pima data set with using the standard t-norm and t-conorm.

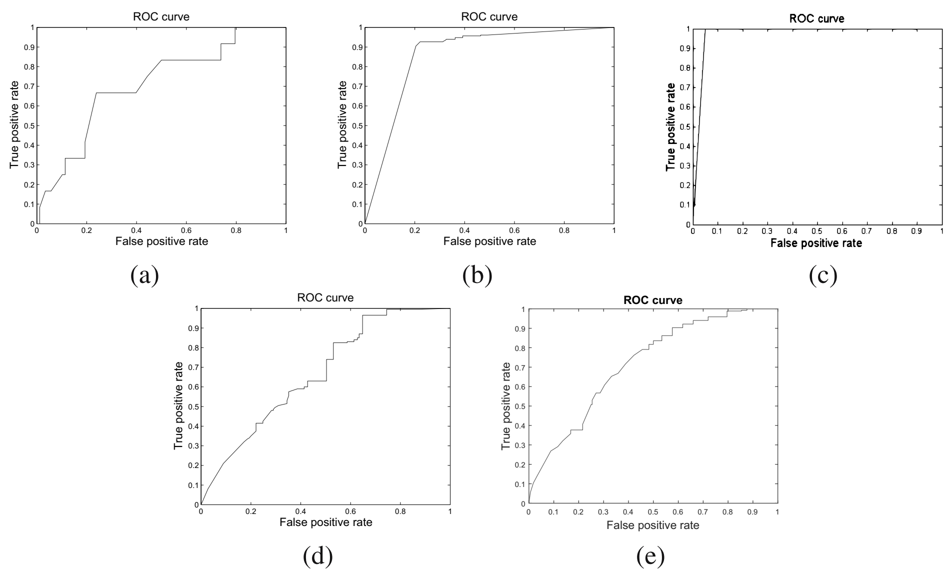


Fig. 6. AUROC curves for fertility with the standard similarity classifier (a), sick with the standard norms (b), hypothyroid with the standard similarity classifier (c), liver disorder with the standard similarity classifier (d), and the Pima Indians data set with the standard similarity classifier (e).

Table 5. Classification results with the liver disorder data set.

Method	Mean accuracy (%)	Variance	AUROC
λ-average classifier with:			
standard norms	**60.29**	6.7320×10^{-4}	0.6270
Łukasiewicz norms	57.36	0.0011	0.5399
probabilistic norms	58.69	0.0019	0.6482
drastic norms	42.20	7.9694×10^{-32}	0.5000
standard similarity classifier	59.92	7.0358×10^{-4}	**0.6688**

5. Discussion

In this paper we presented a new similarity based classification method that uses the λ-averaging operator in aggregation. To the best of our knowledge, λ-averaging has not been previously extended (generalized) to the n-ary case, and/or applied in real world applications. The new n-ary λ-averaging similarity classifier was tested with four n-ary norms namely standard norm (maximum and minimum), the Łukasiewicz norm, and with probabilistic and drastic t-norms and t-conorms. In numerical classification tests the obtained results were compared with those gained with a standard similarity classifier that uses the generalized mean for aggregation.

Five different real world data sets were used in testing the new method. The obtained results compared with the benchmark standard similarity classifier were mixed, but overall the new method's performance was in line with the standard similarity classifier or better. Furthermore, there is no evidence to suggest that using a specific norm would outperform using the other norms; however, weaker results were obtained when the drastic norms were used. The difference in results from our benchmark data sets is attributed to the varying properties of the respective aggregators used; that is, in the intersection case the standard n-ary t-norm is the weakest, or the most allowing norm, whereas the drastic n-ary t-norm creates the strongest requirement. The other two n-ary norms fall within this interval. Similarly, on the union side, the standard n-ary t-conorm is the strongest union and the drastic n-ary t-conorm the weakest.

By changing n-ary norms in the λ-averaging operator, we are "moving" towards the union side from the intersection side, or vice versa. This causes the differences in classification accuracies between the four different cases. Basically, what we have is a "hybrid" aggregation operator that can vary between intersection/averaging/union operators by selecting from between the four n-ary norms. Since changing the n-ary norm used affects the classification accuracy, there is a reason to believe that some data sets have a better fit with union like aggregation operators, while others have a better fit with intersection type operators.

Further research on this topic will include studying the optimization of parameter values for classification as well as determinants of optimal selection of the norms used.

Table 6. Classification results with the Pima Indians data set.

Method	Mean accuracy (%)	Variance	AUROC
λ-average classifier with:			
standard norms	69.97	5.1595×10^{-4}	0.5000
Łukasiewicz norms	74.13	1.6806×10^{-4}	0.5000
probabilistic norms	74.18	1.4456×10^{-4}	0.5000
drastic norms	65.10	0.0000	0.5000
standard similarity classifier	**74.70**	2.0379×10^{-4}	**0.7192**

Acknowledgment

The work of Onesfole Kurama was supported by the Makerere University staff development fund, ref. 0231.11.1346 under the Directorate of Human Resources, and the Research Foundation at the Lappeenranta University of Technology.

References

Alsina, C., Trillas, E. and Valverde, L. (1983). On some logical connectives for fuzzy set theory, *Mathematical Analysis and Applications* **93**(1): 15–26.

Calvo, T., Mayer, G. and Mesiar, R. (2002). *Aggregation Operators: New Trends and Applications*, Physica-Verlag, Heidelberg/New York, NY.

Detyniecki, M. (2000). *Mathematical Aggregation Operators and Their Application to Video Querying*, Ph.D. thesis, University of Paris, Paris.

Dubois, D. and Prade, H. (1985). A review of fuzzy sets aggregation connectives, *Information Sciences* **36**(1): 85–121.

Dubois, D. and Prade, H. (2004). On the use of aggregation operations in information fusion processes, *Fuzzy Sets and Systems* **142**(1): 143–161.

Duda, R., Hart, P. and Stork, G.D. (1973). *Pattern Classification and Scene Analysis*, John Wiley and Sons, New York, NY.

Ezghari, S., Belghini, N., Zahi, A. and Zarghili, A. (2015). A gender classification approach based on 3D depth-radial curves and fuzzy similarity based classification, *Intelligent Systems and Computer Vision Conference, Fez, Morocco*, pp. 1–6.

Fengqiu, L. and Xiaoping, X. (2012a). Constructing kernels by fuzzy rules for support vector regressions, *International Journal of Innovative Computing, Information and Control* **8**(7): 4811–4822.

Fengqiu, L. and Xiaoping, X. (2012b). Design of natural classification kernels using prior knowledge, *IEEE Transactions on Fuzzy Systems* **20**(1): 135–152.

Gabryel, M., Korytkowski, M., Pokropinska, A., Scherer, R. and Drozda, S. (2010). *Evolutionary Learning for Neuro-Fuzzy Ensembles with Generalized Parametric Triangular Norms*, Springer-Verlag, Berlin/Heidelberg.

Gil, G., Girela, L.J., De Juan, J., Gomez-Torres, J.M. and Johnsson, M. (2012). Predicting seminal quality with artificial intelligence methods, *Expert Systems with Applications* **39**(16): 12564–12573.

Hohle, U. (1978). Probabilistic uniformization of fuzzy topologies, *Fuzzy Sets and Systems* **1**(4): 311–332.

Klement, E.P., Mesiar, R. and Pap, E. (2000). *Triangular Norms*, Kluwer Academic Publishers, Dordrecht.

Klement, E.P., Mesiar, R. and Pap, E. (2003a). Triangular norms, Position paper I: Basic analytical and algebraic properties, *Fuzzy Sets and Systems* **143**(1): 5–26.

Klement, E.P., Mesiar, R. and Pap, E. (2003b). Triangular norms, Position paper II: General constructions and parametrized families, *Fuzzy Sets and Systems* **145**(3): 411–438.

Klir, G.J. and Folger, T.A. (1988). *Fuzzy Sets, Uncertainty and Information*, Prentice Hall, Englewood Cliffs, NJ.

Klir, G.J. and Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, Englewood Cliffs, NJ.

Korytkowski, M. and Scherer, R. (2010). *Modular Neuro-fuzzy Systems Based on Generalized Parametric Triangular Norms*, Springer-Verlag, Berlin/Heidelberg.

Łukasiewicz, J. (1970). *Selected Works*, North Holland, Amsterdam/London.

Li, P. and Fang, S.C. (2008). On the resolution and optimization of a system of fuzzy relational equations with sup-t composition, *Fuzzy Optimization Decision Making* **7**(2): 169–214.

Luukka, P. (2005). *Similarity Measure Based Classification*, Ph.D. thesis, Lappeenranta University of Technology, Lappeenranta.

Luukka, P. (2007). Similarity classifier using similarity measure derived from Yu's norms in classification of medical data sets, *Computers in Biology and Medicine* **37**(7): 1133–1140.

Luukka, P. (2008). Similarity classifier using similarities based on modified probabilistic equivalence relations, *Knowledge Based Systems* **22**(1): 57–62.

Luukka, P. (2009). Classification based on fuzzy robust PCA algorithms and similarity classifier, *Expert Systems with Applications* **36**(4): 7463–7468.

Luukka, P. (2011). Feature selection using fuzzy entropy measures with similarity classifier, *Expert Systems with Applications* **38**(4): 4600–4607.

Luukka, P. and Kurama, O. (2013). Similarity classifier with ordered weighted averaging operators, *Expert Systems with Applications* **40**(4): 995–1002.

Luukka, P. and Leppalampi, T. (2006). Similarity classifier with generalized mean applied to medical data, *Computers in Biology and Medicine* **36**(9): 1026–1040.

Luukka, P., Saastamoinen, K. and Kononen, V. (2001). A classifier based on the maximal fuzzy similarity in the generalized Łukasiewicz structure, *Proceedings of the FUZZ-IEEE 2001 Conference, Melbourne, Australia*, pp. 195–198.

Mattila, J.K. (2002). *Fuzzy Logic Course Book,* Art House, Helsinki, (in Finnish).

Menger, K. (1942). Statistical metrics, *Proceedings of the National Academy of Sciences of the United States of America* **28**(12): 535–537.

Newman, D.J., Hettich, S., Blake, C.L. and Merz, C.J. (2012). UCI Repository of Machine Learning Databases, `www.ics.uci.edu/~{}mlearn/MLRepository.html`.

O'Hagan, M. (1988). Aggregating template or rule antecedents in real time expert systems with fuzzy set logic, *Proceedings of the 22nd Annual IEEE Asilomar Conference on Signals, Systems, Computers, Pacific Grove, CA, USA*, pp. 681–689.

Saminger, P.S., Mesier, R. and Dubois, D. (2007). Aggregation operators and commuting, *IEEE Transactions on Fuzzy Systems* **15**(6): 1032–1045.

Schweizer, B. and Sklar, A. (1960). Statistical metric spaces, *Pacific Journal of Mathematics* **10**(1): 313–334.

Schweizer, B. and Sklar, A. (1983). *Probabilistic Metric Spaces*, North-Holland, New York, NY.

Sivaramakrishnan, R. and Arun, C. (2014). Classification of Denver systems of chromosomes using similarity classifier guided by OWA operators, *Current Bioinformatics* **9**(5): 449–508.

Turunen, E. (2002). *Mathematics Behind Fuzzy Logic*, Physic-Verlag, Heidelberg.

Vlahogianni, E. and Karlaftis, M.G. (2013). Fuzzy-entropy neural network freeway incident duration modeling with single and competing uncertainties, *Computer-aided Civil and Infrastructure Engineering* **28**(6): 420–433.

Xu, Z.S. (2008). Linguistic aggregation operators: An overview, *Fuzzy Sets and Their Extension, Representation, Aggregation and Models* **220**(1): 163–181.

Yager, R.R. (1988). On ordered weighted averaging aggregation operators in multi-criteria decision making, *IEEE Transactions on Systems, Man and Cybernetics,* **18**(1): 183–190.

Onesfole Kurama received his M.Sc. degree in mathematics in 2010 from Makerere University, Uganda. He also received another M.Sc. in technomathematics in 2012 from the Lappeenranta University of Technology, Finland. He is currently a doctoral researcher at the Lappeenranta University of Technology and an assistant lecturer at Makerere University. His research is focused on soft computing in data analysis with applications in industrial, medical and economic problems.

Pasi Luukka received his M.Sc. degree in information technology in 1999 and the D.Sc. degree in applied mathematics in 2005 from the Lappeenranta University of Technology, Finland. He is currently an associate professor of applied mathematics and strategic finance with that university. His research is focused on soft computing methods in data analysis and decision making. The main research interests include classification, feature selection and projection methods, similarity and distance measures, fuzzy multi-criteria decision making methods, decision support systems, and university consensus.

Mikael Collan received his M.Sc. degree in social sciences in 2000 and the D.Sc. degree in economics and business administration in 2004 from Abo Akademi University, Turku, Finland. He is currently a professor of strategic finance with the Lappeenranta University of Technology Business School in Finland. His research is focused on profitability analysis, asset valuation methods, and the application of fuzzy logic and real options in management decision support.

Received: 1 September 2015
Revised: 2 February 2016
Accepted: 19 February 2016