Information Sciences 486 (2019) 434–449

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Application of interval-valued aggregation to optimization problem of $k - NN$ classifiers for missing values case

Urszula Bentkowska*, Jan G. Bazan, Wojciech Rząsa, Lech Zaręba

Faculty of Mathematics and Natural Sciences, University of Rzeszów, Al. Rejtana 16C, Rzeszów 35–959, Poland

ARTICLE INFO

Article history:
Received 4 April 2018
Revised 20 December 2018
Accepted 20 February 2019
Available online 23 February 2019

MSC:
03E72

Keywords:
Interval-valued aggregation function
Decision making
Classifier
Missing values
$k - NN$ Classifier

ABSTRACT

In this contribution we consider methods for improving the quality of classification by the k nearest neighborhood classifiers in the case of large number of missing values. The classical version of such classifier is applied, moreover the classifier with the aggregation of certainty coefficients of the individual classifiers with the use of the arithmetic mean, as well as new versions of the classifier with the use of diverse interval-valued aggregation functions are considered. After adding missing values to the data, the quality of the classical method decreases. By contrast, the interval modeling of missing values and application of interval-valued aggregation functions in the classifiers entail a much slower decrease of their quality. The paper includes the results of experiments performed on the data sets from the machine learning repository UCI.

1. Introduction

Classifiers, known also in literature as *decision algorithms, classifying algorithms* or *learning algorithms*, may be treated as constructive, approximate descriptions of concepts (decision classes). These algorithms constitute the kernel of *decision systems* that are widely applied in solving many problems occurring in such domains as *pattern recognition, machine learning, expert systems, data mining* and *knowledge discovery* (cf. [38]).

Data sets used for classifiers generation may be represented using data tables. In this representation individual objects correspond to rows of a given data table and attributes to columns of a given data table. In this paper, we consider *decision tables* of the form $\mathbf{T} = (U, A, d)$ in Pawlak's sense (cf. [44]) for representation of data tables, where A is a set of attributes or columns in the data table, U is a set of objects (rows) in the data table, and d is a distinguished attribute from the set A called a *decision attribute*.

When classifiers are used, there is a problem of lowering their performance on test data due to the missing values in data test tables. The more missing values the lower quality of the classification can be significant compared to the situation when we classify test objects without missing values. Classifiers are usually created on training data that does not have many missing values or even does not have them at all. As a result, they are able to learn well to recognize test cases based on the values of conditional attributes. Therefore, when classifying test objects with missing values, difficulties arise due to the fact that the classifier cannot recognize the test object.

* Corresponding author.
E-mail addresses: ududziak@ur.edu.pl (U. Bentkowska), bazan@ur.edu.pl (J.G. Bazan), wrzasa@ur.edu.pl (W. Rząsa), lzareba@ur.edu.pl (L. Zaręba).

https://doi.org/10.1016/j.ins.2019.02.053
0020-0255/

In literature there may be found descriptions of numerous approaches to constructing classifiers, which are based on such paradigms of machine learning theory as *classical and modern statistical methods, neural networks, decision trees, decision rules*, and *inductive logic programming* (cf. [2–5,38]).

In this paper we propose a new approach of creating uncertainty intervals for the k nearest neighborhood classifier ($k-NN$) and then aggregating the obtained intervals with the use of interval-valued aggregation functions. Our aim is to compare the new interval methods with classical and generally recognized implementation of $k-NN$ classifiers. The results show that the newly proposed classifier, based on interval modeling of missing values and interval-valued aggregation methods applied, performs much better in the situation of increase of missing data than the classical versions of $k-NN$ classifiers do. As for the $k-NN$ method, there are many variants and several parameters that can be used (cf. [1,21]). One of the most important parameters is the distance used, which can be chosen in many ways. We have chosen the Euclidean distance since it is the best known one and the most frequently used. In addition, it matches data with numeric attributes that were analyzed. Moreover, there are many implementations of the $k-NN$ method in various software libraries and programming languages. We applied Java programming language, where one of the most important implementations of the $k-NN$ algorithm is the implementation in the WEKA API library (cf. [27,29]) used in the mentioned experiments.

Aggregation functions proved to be an effective tool in many application areas [6]. For example, in information retrieval diverse aggregation strategies are used [37]. Let us add that there are currently many approaches to use domain knowledge and improve the quality of data mining models [2–5,41–43]. In this context, the use of aggregation functions can be treated as a way to use domain knowledge to improve the quality of classifiers. Continually, new classes of aggregation functions with good performance in applications are introduced, for example in fuzzy rule-based classification systems [18,36]. In this contribution we will use extensions of aggregation functions, namely interval-valued aggregation functions which are recently developed by several authors and successfully applied (cf. [8,23]). In our new version of the classifier we propose to consider interval modeling in the case of incomplete data. In this way we obtain the so called *uncertainty intervals* (cf. [20]) used in interval-valued fuzzy settings (cf. [54]). Moreover, aggregation of uncertainty intervals is proposed. Here we present the results on aggregation methods connected with recently introduced possible and necessary aggregation functions [7]. These notions are connected with possible and necessary comparability relations which follow from the epistemic setting of interval-valued fuzzy calculus [19,20].

To assess the quality of the analyzed methods, we used the AUC parameter, which is the area under the ROC curve (Receiver Operating Characteristic curve, cf. [24,49]). This parameter is much more reliable than giving specific values of sensitivity and specificity, because each such pair of parameters corresponds to one point on the ROC curve, it means that the AUC method gives much more information and enables the adjustment of sensitivity and specificity. Certainly, the use of this method requires classifiers with the ability to adjust sensitivity and specificity, but classifiers that were used in our experiments have such an option.

An evaluation of our approach was conducted on data sets from UCI machine learning repository [51]. The results confirmed that methods based on interval modeling and interval-valued aggregation make it possible to reduce the negative impact of lack of data.

The paper is structured as follows. In Section 2, notions connected with aggregation operators are recalled. In Section 3, the motivation to consider new versions of classifier are provided as well as a description of them is given. In Section 4, details concerning the performed experiments are presented.

2. Aggregation operators for interval-valued setting

Firstly, we recall definition of an interval-valued fuzzy set and the classically applied order for this setting.

Definition 2.1 (cf. [54])**.** An interval-valued fuzzy set F on X is a mapping $F\colon X\to L^I$ such that $F(x)=[\underline{F}(x),\overline{F}(x)]\in L^I$ for $x\in X$, where

$$L^I=\{[\underline{x},\overline{x}]:\ \underline{x},\overline{x}\in[0,1],\ \underline{x}\leqslant\overline{x}\}.$$

The well-known classical monotonicity (partial order) for interval-valued fuzzy settings is of the form (cf. [35])

$$[\underline{x},\overline{x}]\preceq[\underline{y},\overline{y}]\Leftrightarrow\underline{x}\leqslant\underline{y},\ \overline{x}\leqslant\overline{y}. \tag{1}$$

We may also consider the following comparability relations on L^I (cf. [45]):

$$[\underline{x},\overline{x}]\preceq_\pi[\underline{y},\overline{y}]\Leftrightarrow\underline{x}\leqslant\overline{y}, \tag{2}$$

$$[\underline{x},\overline{x}]\preceq_\nu[\underline{y},\overline{y}]\Leftrightarrow\overline{x}\leqslant\underline{y}. \tag{3}$$

These relations, including classical order, follow from the epistemic settings of interval-valued fuzzy sets (representing the uncertainty of membership value of a given object in an interval-valued fuzzy set). Relation $\preceq_\pi$ between intervals may be interpreted as follows: at least one element in the first interval is smaller or equal to at least one element in the second interval. Relation $\preceq_\nu$ between intervals may be interpreted as follows: each element in the first interval is smaller or equal to each element in the second interval. Relation (1) (traditionally used in interval-valued setting) may be interpreted as a conjunction of the condition: there exists an element in the first interval that is smaller or equal to each element in the

second interval and the condition: for each element in the first interval there exists a greater or equal element in the second interval. Regarding this type of interpretation (all combinations of universal and existential quantifiers) by considering these three relations we have the full possible set of interpretations of comparability relations on intervals. Relations $\preceq_\pi$ and $\preceq_\nu$ are not partial orders. Relation $\preceq_\pi$ is an interval order (complete and fulfils the Ferrers property, cf. [26]) and the relation $\preceq_\nu$ is antisymmetric and transitive. Other notions of orders for intervals were introduced for example in the paper [11] where the general method to build different linear orders for the family of intervals L^I, covering some of the known linear orders for intervals, was presented. Other examples of orders for the family of intervals were considered for example in [14,32,46]. In this paper we will focus only on the comparability relations (1) - (3) and the notions of aggregation functions on L^I which follow from these comparability relations, namely aggregation functions, possible aggregation functions and necessary aggregation functions. The latter two notions were recently introduced in [7]. We will present their usefulness in classification problems. Before giving adequate definitions we recall the notion of an aggregation function on the unit interval [0,1]. Certainly, we may define aggregation function on any other closed interval but for application reasons in fuzzy sets theory (or interval-valued fuzzy sets theory) they are usually considered on [0,1]. If needed, the results may be transformed to the domain of other intervals by rescaling.

Definition 2.2 (cf. [12]. p. 6)**.** A function A: $[0, 1]^n \to [0, 1]$, $n \in \mathbb{N}$, $n \geqslant 2$, which is increasing in each variable, i.e.

$$\underset{s_1,\ldots,s_n,t_1,\ldots,t_n\in[0,1]}{\forall} \Big(\underset{1\leqslant i\leqslant n}{\forall} s_i \leqslant t_i\Big) \Rightarrow A(s_1,\ldots,s_n) \leqslant A(t_1,\ldots,t_n), \tag{4}$$

is called an aggregation function if $A(0,\ldots,0) = 0$, $A(1,\ldots,1) = 1$.

Condition (4) means that if one input increases while the others are kept constant, the overall degree of A must not decrease. The notion of an aggregation function may be naturally extended to the domain L^I.

Definition 2.3 (cf. [34])**.** An operation $\mathcal{A} : (L^I)^n \to L^I$ is called an aggregation function on L^I if it is increasing, i.e.

$$\underset{\mathbf{x}_i,\mathbf{y}_i\in L^I}{\forall} \mathbf{x}_i \preceq \mathbf{y}_i \Rightarrow \mathcal{A}(\mathbf{x}_1,\ldots,\mathbf{x}_n) \preceq \mathcal{A}(\mathbf{y}_1,\ldots,\mathbf{y}_n) \tag{5}$$

and $\mathcal{A}(\underbrace{\mathbf{0},\ldots,\mathbf{0}}_{n\times}) = \mathbf{0}$, $\mathcal{A}(\underbrace{\mathbf{1},\ldots,\mathbf{1}}_{n\times}) = \mathbf{1}$, where $\mathbf{0} = [0, 0]$, $\mathbf{1} = [1, 1]$.

Below we give some construction methods of interval-valued aggregation functions on L^I. These methods involve the notion of an aggregation function on [0,1]. One of the most common methods is the *representability* also called *decomposability*.

Definition 2.4 (cf. [15])**.** $\mathcal{A} : (L^I)^n \to L^I$ is said to be a representable aggregation function on L^I if there exist two aggregation functions A_1, A_2: $[0, 1]^n \to [0, 1]$, $A_1 \leqslant A_2$ such that, for every $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1]$, $\mathbf{x}_2 = [\underline{x}_2, \overline{x}_2]$, ..., $\mathbf{x}_n = [\underline{x}_n, \overline{x}_n] \in L^I$ it holds that

$$\mathcal{A}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = [A_1(\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n), A_2(\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)].$$

Representability is not the only possible way to build interval-valued aggregation functions. Let $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1]$, $\mathbf{x}_2 = [\underline{x}_2, \overline{x}_2]$, ..., $\mathbf{x}_n = [\underline{x}_n, \overline{x}_n] \in L^I$ and A_1, A_2: $[0, 1]^n \to [0, 1]$, $A_1 \leqslant A_2$ be aggregation functions. The following is a non-representable aggregation function on L^I (cf. [16])

$$\begin{aligned}\mathcal{A}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = [&A_1(\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n),\\ &\max(A_2(\underline{x}_1, \overline{x}_2, \ldots, \overline{x}_n), A_2(\overline{x}_1, \underline{x}_2, \overline{x}_3, \ldots, \overline{x}_n), \ldots, A_2(\overline{x}_1, \ldots, \overline{x}_{n-1}, \underline{x}_n))].\end{aligned} \tag{6}$$

In this paper we would like to pay special attention to the usefulness of interval modeling while dealing with the missing values in classification process and application of interval-valued aggregation operators in improving the classification results. Especially, we will consider aggregation operators from the recently introduced classes of possible and necessary aggregation functions. By replacing in the monotonicity condition (5) the natural order $\preceq$ with the relations $\preceq_\pi$ and $\preceq_\nu$, new types of aggregation functions are obtained [7].

Definition 2.5. An operation $\mathcal{A} : (L^I)^n \to L^I$ is called a *possible aggregation function* (for short pos-aggregation function) if

$$\underset{\mathbf{x}_i,\mathbf{y}_i\in L^I}{\forall} \mathbf{x}_i \preceq_\pi \mathbf{y}_i \Rightarrow \mathcal{A}(\mathbf{x}_1,\ldots,\mathbf{x}_n) \preceq_\pi \mathcal{A}(\mathbf{y}_1,\ldots,\mathbf{y}_n) \tag{7}$$

and $\mathcal{A}(\underbrace{\mathbf{0},\ldots,\mathbf{0}}_{n\times}) = \mathbf{0}$, $\mathcal{A}(\underbrace{\mathbf{1},\ldots,\mathbf{1}}_{n\times}) = \mathbf{1}$.

Definition 2.6. An operation $\mathcal{A} : (L^I)^n \to L^I$ is called *a necessary aggregation function* (for short nec-aggregation function) if

$$\underset{\mathbf{x}_i,\mathbf{y}_i\in L^I}{\forall} \mathbf{x}_i \preceq_\nu \mathbf{y}_i \Rightarrow \mathcal{A}(\mathbf{x}_1,\ldots,\mathbf{x}_n) \preceq_\nu \mathcal{A}(\mathbf{y}_1,\ldots,\mathbf{y}_n) \tag{8}$$

and $\mathcal{A}(\underbrace{\mathbf{0},\ldots,\mathbf{0}}_{n\times}) = \mathbf{0}$, $\mathcal{A}(\underbrace{\mathbf{1},\ldots,\mathbf{1}}_{n\times}) = \mathbf{1}$.

Some representatives of aggregation operators, belonging to the three classes of aggregation operators on L^I considered in this paper, will be presented. The performance in classification process of these aggregation operators will be provided. Below there are given six aggregation operators applied in our experiments. Namely,

$$\mathcal{A}_1(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n)=\left[\frac{\underline{x}_1+\underline{x}_2+\cdots+\underline{x}_n}{n},\frac{\overline{x}_1+\overline{x}_2+\cdots+\overline{x}_n}{n}\right], \tag{9}$$

where $\mathcal{A}_1$ is a representable aggregation function on L^I constructed with the use of the arithmetic mean. In [7] (cf. Theorem 2, Theorem 3) it was shown that this is a possible and necessary aggregation function.

$$\mathcal{A}_2(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n)=\left[\frac{\underline{x}_1+\underline{x}_2+\cdots+\underline{x}_n}{n},\right.$$
$$\left.\max\left(\frac{\underline{x}_1+\overline{x}_2+\cdots+\overline{x}_n}{n},\frac{\overline{x}_1+\underline{x}_2+\overline{x}_3+\cdots+\overline{x}_n}{n},\ldots,\frac{\overline{x}_1+\cdots+\overline{x}_{n-1}+\underline{x}_n}{n}\right)\right], \tag{10}$$

where $\mathcal{A}_2$ is an aggregation function of the form (6). In [7] it was proved that this is a necessary aggregation function (cf. [7]. Proposition 6) and it was shown that this is not a pos-aggregation function (cf. [7]. Example 16).

$$\mathcal{A}_3(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n)=\left[\frac{\underline{x}_1+\underline{x}_2+\cdots+\underline{x}_n}{n},\frac{\overline{x}_1^2+\overline{x}_2^2+\cdots+\overline{x}_n^2}{\overline{x}_1+\overline{x}_2+\cdots+\overline{x}_n}\right] \tag{11}$$

and its more general version $\mathcal{A}_4$, where $p\in\mathbb{N}$, $p\geqslant 2$,

$$\mathcal{A}_4(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n)=\left[\frac{\underline{x}_1+\underline{x}_2+\cdots+\underline{x}_n}{n},\frac{\overline{x}_1^p+\overline{x}_2^p+\cdots+\overline{x}_n^p}{\overline{x}_1^{p-1}+\overline{x}_2^{p-1}+\cdots+\overline{x}_n^{p-1}}\right]. \tag{12}$$

$\mathcal{A}_3$ and $\mathcal{A}_4$ are possible aggregation functions (cf. [7]. Theorem 2) but they are neither aggregation functions (in the sense of Definition 2.3) nor nec-aggregation functions (cf. [7]. Theorem 3), where in the case of $\mathcal{A}_4$ the greater the p value, the greater the value of the end of interval (which depends on the value p). $\mathcal{A}_3$, $\mathcal{A}_4$ are given with the convention $\frac{0}{0}=0$, where $\frac{0}{0}$ occurs in the case $\mathbf{x}_1=\mathbf{x}_2=\cdots=\mathbf{x}_n=[0,0]$, so if $\overline{x}_1=\overline{x}_2=\cdots=\overline{x}_n=0$. Operators $\mathcal{A}_3$ and $\mathcal{A}_4$ are not aggregation functions (in the sense of Definition 2.3) since the upper bounds of the intervals are not monotonic in the classical way (namely according to (4)). The upper bounds of these aggregation operators were built with the use of *Lehmer means* (cf. [10]. p. 185), where

$$L_{(p)}(t_1,\ldots,t_n)=\frac{\sum_{k=1}^n t_k^p}{\sum_{k=1}^n t_k^{p-1}}$$

where $p\in\mathbb{R}$, and there is used the convention $\frac{0}{0}=0$. However, as it was mentioned $\mathcal{A}_3$ and $\mathcal{A}_4$ (for $p\geqslant 2$) fulfill the monotonicity condition (7) and as a result they are pos-aggregation functions.

$$\mathcal{A}_5(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n)=\left[\sqrt{\frac{\underline{x}_1^2+\underline{x}_2^2+\cdots+\underline{x}_n^2}{n}},\sqrt[3]{\frac{\overline{x}_1^3+\overline{x}_2^3+\cdots+\overline{x}_n^3}{n}}\right], \tag{13}$$

$$\mathcal{A}_6(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n)=\left[\sqrt[3]{\frac{\underline{x}_1^3+\underline{x}_2^3+\cdots+\underline{x}_n^3}{n}},\sqrt[4]{\frac{\overline{x}_1^4+\overline{x}_2^4+\cdots+\overline{x}_n^4}{n}}\right], \tag{14}$$

where $\mathcal{A}_5$ and $\mathcal{A}_6$ are representable aggregation functions (cf. Definition 2.4). In [7] it was shown that they are also possible aggregation functions (cf. [7]. Theorem 2 and Corollary 1) but they are not necessary aggregation functions (cf. [7]. Theorem 3 which characterizes decomposable nec-aggregation functions). Aggregation operators $\mathcal{A}_5$ and $\mathcal{A}_6$ are constructed with the use of the adequate root-power means (cf. [30]. p. 26), where by a root-power mean we understand an aggregation function on [0,1] (cf. Definition 2.2) of the form

$$P_{(r)}(t_1,\ldots,t_n)=\begin{cases}0, & r<0,\ \exists\limits_{1\leqslant k\leqslant n} t_k=0\\ \left(\frac{1}{n}\sum\limits_{k=1}^n t_k^r\right)^{\frac{1}{r}}, & \text{otherwise}\end{cases}$$

where $r\in\mathbb{R}_0$. For these types of means we have the following dependency (cf. [30]. p. 26)

$$\forall_{r<q}\ \forall_{t_1,\ldots,t_n\in(0,1]}\quad P_{(r)}(t_1,\ldots,t_n)\leqslant P_{(q)}(t_1,\ldots,t_n). \tag{15}$$

According to the inequality (15), the lower bound and the upper bound of the operator $\mathcal{A}_5$ were obtained for $r=2$ and $q=3$, respectively. Similarly, the lower bound and the upper bound of the operator $\mathcal{A}_6$ were obtained for $r=3$ and $q=4$, respectively.

Possible and necessary aggregation functions were considered in detail in [7]. Here we presented some promising examples of the families of aggregation functions to be applied in classification problems. Since this is the first attempt to apply

the new classes of aggregation functions in classification problems, we have chosen some typical representatives of both the classically used aggregation functions for interval-valued fuzzy settings (cf. Definition 2.3) and the new classes of aggregation operators such as possible and necessary aggregation functions (cf. Definition 2.5 and Definition 2.6). The aggregation functions that were chosen are intuitively suitable for the construction of classifiers for the considered here type of data.

3. Construction of the classifiers

Now, we will present the motivation to consider intervals when missing values appear in classification procedure. This will also justify the application of aggregation operators for interval calculus.

3.1. Motivation

The problem of missing values in classification problems is usually solved in the following three ways.

The first method is that the missing value is treated as a normal value that carries certain information. For example, for medical data, the missing value for a certain attribute representing the result of a specialized medical examination often means that this test was not needed because of the patient's condition. Therefore, the missing value may be treated as information about the patient's condition. However, this approach requires teaching the classifier on data containing these kind of missing values. The classifier so learned can classify test objects containing missing values as well. This approach does not really consider missing values, because they are treated as normal values.

In the second approach such classifiers are created that can classify objects only on the basis of those attributes on which there are no empty spaces. This approach requires the creation of specific classifiers that can classify test objects based only on some attributes. A typical example of such classifier is a classifier based on decision rules. If there are many rules in a set of rules, then usually one will be able to recognize the test object based on non-missing values. It is worth noting, that such classifier will work much worse if it classifies test objects with missing values compared to the case when there are no such values in the test object.

The third approach, which seems to be the most commonly used in practice (cf. [17,28]), is that when classifying a test object having empty spaces, before classifying the object, empty spaces are filled with a specific value determined based on training data. It is only after this filling that the classifier is used to classify the test object thus completed. The method of value replenishment (data imputation) usually consists in searching for the most frequently occurring value of a symbolic attribute or the average value of a numerical attribute. Although this approach seems uncertain, in practice it often allows good classification results to be obtained on test data with empty spaces. It is worth noting, however, that this method in a sense introduces artificial values into data, which in reality could not occur at this point. These are frequent values in the training data and therefore the method works well for typical objects. However, for untypical test objects, the classifier used in this way may be mistaken. Another method was applied in OvaExpert, one of the real-life diagnosis support systems for ovarian tumor where the missing values were replaced with the whole intervals representing the given feature (cf. [22,23,40,47,50,52,53,55,56]). This approach proved to be effective to make high-quality decisions under incomplete information and uncertainty.

If we assume that the phenomenon of appearance of missing values in the test data is random, it means that the biggest problems with the correct classification of test data will have classifiers by definition using all conditional attributes. This is due to the fact that classifiers that do not use all attributes for classification at the same time (e.g. decision rules, decision trees, etc.) when classifying a test object can often use attribute values that do not require the above-mentioned fill and therefore classify objects more confidently. However, classifiers that use all attributes do not have this chance and must also use artificially filled values. A typical example of such classifier is the k-nearest neighbors classifier ($k - NN$) [38]. The test object data can be classified using the $k - NN$ method with diverse parameters k. Therefore, a conflict appears between classifiers that operate on the basis of different k values, which must be resolved in order to finally classify the test object. In this work, to resolve this conflict, we suggest aggregation of uncertainty intervals by individual classifiers. As a result, we build a new compound classifier. There are applied classical versions of the classifier as well as the aggregated versions for diverse interval-valued aggregation functions. We compare the obtained results for the classifier using interval modeling of missing data with the classical version of the classifier and the classifier with the aggregation of certainty coefficients of individual classifiers by means of the arithmetic mean (without the use of uncertainty intervals). We will show that appearance of the missing values causes the decrease of the quality of the classical methods. However, adding missing values to the aggregated versions of classifiers (with interval modeling of missing values) causes a much slower decrease of their quality and outperforms the other two presented methods which is also confirmed by the statistical measures.

3.2. $k - NN$ Algorithm

In an unpublished US Air Force School of Aviation Medicine report, Fix and Hodges in 1951 introduced a non-parametric method for pattern classification that has since become known as the k-nearest neighbor rule [25]. In 1967, some of the formal properties of the k-nearest neighbor rule were obtained [13]. The $k - NN$ (cf. [38]) is a method for classifying objects based on closest training examples in a feature space. It is a type of instance-based learning. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors

(k is a positive integer, usually small). If $k = 1$, then the object is simply assigned to the class of its nearest neighbor. There were considered also distance weighted approaches ([1,21]), soft computing methods [9] and fuzzy ones ([31,33]).

In this paper we will use three methods involving k-nearest neighbors classifier. The first, is the classical one and will be denoted Algorithm C, for short C. The experiments were performed with the use of WEKA API library (cf. [27,29]), where before calculating the Euclidean distance, values on all numeric attributes are normalized to the interval [0,1]. The difference between numeric values in the distance formula is calculated as the absolute value between them and for nominal values the result equals 0 if they are the same and it equals 1 if they are different. The case of missing values is solved at the stage of calculating the distance between the values on a given attribute. For nominal attributes, assume that a missing feature is maximally different from any other feature value. Thus if either or both values are missing, or if the values are different, the difference between them is taken as one; the difference is zero only if they are not missing and both are the same. For numeric attributes, the difference between two missing values is also taken as one. However, if just one value is missing, the difference is often taken as either the (normalized) size of the other value or one minus that size, whichever is larger. This means that if values are missing, the difference is as large as it can possibly be.

The test object data can be classified using the $k - NN$ method with diverse parameters k. For example, it can be $k = 1, 3, 5, 10, 20$ etc. In each of these cases a different set of neighbors is used, which can classify the test object with different levels of confidence for different decision classes. This is due, among other things, to the fact that various objects appear in particular environments, including objects with artificially filled values. Therefore, a conflict appears between classifiers that operate on the basis of different k values, which must be resolved in order to finally classify the test object. In this work, to resolve this conflict, we suggest aggregation of the certainty coefficients obtained by the individual classifiers. This is connected with the second method, Algorithm M (for short M, cf. Algorithm 1) which is based on the fact that we have

Algorithm 1: Classification of a test object by the M classifier.

Input:
1. training data set represented by decision table $\mathbf{T} = (U, A, d)$, where $n = card(U)$ and $l = card(A)$,
2. collection $C_1, \ldots, C_m$ of $k - NN$ classifiers for different k, where, e.g., $k \in \{5, 10, 20, 30\}$,
3. test object u.

Output: The membership of the object u to the "main class" or to the "subordinate class"

```
begin
    for i := 1 to m do
        Compute certainty coefficient ("main class" membership probability) for the given test object u using the
        classifier C_i and assign it to p_i
    end
    Determine the final certainty coefficient p for the object u by aggregating (with a use of the arithmetic mean) the
    certainty coefficients p_1,...,p_m.
    if p > 0.5 then
        return u belongs to the "main class"
    else
        return u belongs to the "subordinate class"
    end
end
```

a number of the classical $k - NN$ classifiers and we make a simple aggregation (with the use of the arithmetic mean) of the degree of classification from all the individual classifiers.

We suggest also another method - Algorithm F involving interval-valued aggregation of uncertainty intervals by individual classifiers. Details of this method will be given in the next section.

3.3. New version of classifier

The proposed new method of building a classifier, Algorithm F (for short F, cf. Algorithm 2) consists in using a special procedure that is in fact a complex classifier.

At the beginning, we set a certain decision class, which we call the main class. We assume that there is a second decisional class in the data set, which we call a subordinate class. The main class can be, for example, the class of patients suffering from a given disease, and the subordinate class may be the class of healthy patients. For a given family of k parameters (e.g. $k = 1, 3, 5, 10$), each $k - NN$ classifier determines the uncertainty interval of belonging of a test object to the main class (it will be explained later). Then, the method of aggregating intervals is used to obtain the final uncertainty interval. The final uncertainty interval is used to classify the object at the level corresponding to the center point of the final uncertainty interval.

Now, we will explain the method of determining the uncertainty interval by a single $k - NN$ classifier. The method proposed for determining the uncertainty interval by a single classifier consists in the fact that a test object having missing

Algorithm 2: Classification of a test object by the F classifier.

Input:
1. data set represented by decision table $\mathbf{T} = (U, A, d)$, where $n = card(U)$ and $l = card(A)$,
2. collection $C_1, \ldots, C_m$ of $k - NN$ classifiers for different k, where, e.g., $k \in \{5, 10, 20, 30\}$,
3. fixed parameter r, e.g., $r = 10$,
4. aggregation function $\mathcal{A}$,
5. test object u

Output: The membership of the object u to the "main class" or to the "subordinate class"

begin
  if *exists at least one missing value in the object* u **then**
    for $i := 1$ **to** m **do**
      Choose randomly with the Monte Carlo method r objects $u_1, \ldots, u_r$ on the basis of object u, where any object u_j ($j \in \{1, \ldots, r\}$) is constructed in the following way:
      begin
        Copy values of attributes from u to u_j;
        For each attribute whose value in u_j is missing, replace it with a randomly selected value from the range of possible values for this attribute (the range designated from the training data);
      end
      Compute certainty coefficient for objects $u_1, \ldots, u_r$ using the classifier C_i and assign these values to $p_1, \ldots, p_r$;
      Compute $min\{p_1, \ldots, p_r\}$ and assign it to min_i;
      Compute $max\{p_1, \ldots, p_r\}$ and assign it to max_i;
    end
    Determine the uncertainty interval $[down(u), up(u)]$ for the object u by aggregating (with the use of interval-valued aggregation function $\mathcal{A}$) the intervals $[min_1, max_1]$, ..., $[min_m, max_m]$;
    Determine the final certainty coefficient $\mathbf{p} = \frac{down(u)+up(u)}{2}$ for the object u;
  else
    for $i := 1$ **to** m **do**
      Compute certainty coefficient ("main class" membership probability) for the given test object u using the classifier C_i and assign it to p_i ;
    end
    Determine the uncertainty interval $[down(u), up(u)]$ for the object u by aggregating (with the use of $\mathcal{A}$) the intervals $[p_1, p_1]$, ..., $[p_m, p_m]$;
    Determine the final certainty coefficient $\mathbf{p} = \frac{down(u)+up(u)}{2}$ for the object u;
  end
  if $\mathbf{p} > 0.5$ **then**
    return u *belongs to the "main class"*;
  else
    return u *belongs to the "subordinate class"*;
  end
end

values is classified by a given $k - NN$ classifier in a specific way. Namely, during this classification, many classifications of different objects are actually made, which are constructed based on the test object. The construction of these objects is based on the fact that missing values in the object are filled in various ways based on the values from the training data. The ideal situation here would be that during the classification procedure, all possible test objects that can be generated from the given test object are classified by inserting empty values in all possible ways of the attribute values from the training data for the given attribute. The result of each such classification is the certainty value of belonging to the main class. Thanks to this value, the uncertainty interval may be computed by determining the minimum of these values (lower end of the interval) and the maximum (upper end of the interval) (cf. [39,53]). Unfortunately, due to the possibility of the appearance of a very large number of objects generated from the given test object, in practice, the above method cannot be used in its pure form. If, for example, there are 20 attributes in the training table and each of them can have only 10 values, then in the case of the classification of a test object having 10 empty and 10 filled places, $10^{10} = 10$ billion objects would have to be classified. From the point of view of computational complexity, this is definitely too much. Therefore, we propose the Monte Carlo method of choosing the above objects. This method consists in the fact that in the space of all possible objects that can be generated for the given test object with missing values, we select a random sample (the draw being made in accordance with the distribution of variable). Then, we classify only objects from this sample and on the basis of the obtained classification results, we estimate the lower and upper end of the uncertainty interval.

Table 1
Experimental data set details.

UCI data	Objects	Attributes	Classes
Banknote	1372	5	2
Biodeg	1055	43	2
Breast cancer	699	11	2
Diabetes	768	9	2
German	1000	25	2
Ozone	2536	74	2
Parkinson	1040	29	2
Red wine	1599	12	2
Rethinopathy	1151	20	2
Spam	4601	59	2

Now, we present a brief analysis of the computational time complexity of the Algorithm 2. If we assume that the classical $k-NN$ algorithm works in time of order $O(n \cdot l)$, where n is the number of objects in the set U, and l is the number of attributes in the set A, then it is easy to see that the time complexity of the Algorithm 2 is of order $O(m \cdot r \cdot n \cdot l)$, where m is the number of classifiers from the collection $C_1, \ldots, C_m$ and r is the parameter of Monte Carlo method. This complexity is not large and the algorithm can be used in practical applications wherever the $k-NN$ algorithm can be used.

4. Experiments

Now, we will present detailed information about the experiments that were carried out.

4.1. Conditions of experiments

The experiments have been performed on the 10 data sets obtained from UC Irvine (UCI) Machine Learning repository [51]. Table 1, shows the summary of the characteristics of the data sets. The considered attributes have numerical values only. All algorithms were implemented and tested in Java programming language (Weka API library, cf. [27,29]). To assess the quality of the analyzed methods the AUC parameter was used. The use of AUC method requires classifiers with the ability to adjust sensitivity (SN) called also true positive rate (TPR) and specificity (SP) called also true negative rate, where

$$SN = TPR = \frac{TP}{P} = \frac{TP}{TP+FN},$$

$$SP = TNR = \frac{TN}{N} = \frac{TN}{TN+FP},$$

$$FPR = \frac{FP}{N} = \frac{FP}{FP+TN},$$

$$FNR = \frac{FN}{P} = \frac{FN}{FN+TP},$$

$$FPR = 1 - SP, \qquad FNR = 1 - SN.$$

FPR is called an error of the first type (or α error) and FNR is called an error of the second type (or β error). AUC is the indicator of the quality of a classifier which is the area under the ROC curve (cf. [24,49]). ROC shows the dependence of sensitivity SN on error of the first type FPR during calibration of the classifier (at various threshold settings). These two coefficients are determined on the basis of the cost matrix and as a result each singular binary classifier may be presented as a point $(1-SP, SN)$ in the coordinate system. The greater is the AUC value the better is the classifier. For the ideal classifier the value AUC is equal to 1. Other details of the performed experiments are presented below.

- The Euclidean metric is applied for measuring distances.
- Each data set is divided into two training and test parts, in the proportion of 50% to 50%.
- Training data does not have missing values.
- A random missing value with different probability is entered into the test data (this is the experiment parameter). In each experiment, missing values are randomly entered into all conditional attributes in the same proportion. For example, if the input parameter is 0.2, then 20% of the missing value is randomly entered into every attribute (column).
- In the experiments, in the method F, six interval-valued aggregation functions $\mathcal{A}_1$ - $\mathcal{A}_6$ were applied ($\mathcal{A}_4$ is used with $p = 3$).
- Experiments are carried out for the following classifiers:

- classifier with the **C** code - this is the classical classifier with the $k - NN$ method (classifiers for $k = 1, 3, 5, 10, 15, 20$ and 30 were used);
- classifier with the **M** code - uses the aggregation of certainty coefficients of individual classifiers by means of the arithmetic mean;
- classifier with the **F** code - this is the algorithm of aggregation of uncertainty intervals described in Subsection 3.3.

• Each experiment is repeated 10 times and the average AUC and standard deviation are reported.
• In each experiment, the main class has always been determined, which is given as an experiment parameter. The data was so selected that only two decision classes were considered.

4.2. Discussion on the results

Generally, the increase of missing values resulted in decrease of classification possibility. However, applying the classification method *F*, with the proposed here method of treating the case of missing values and aggregation of uncertainty intervals, always gave the best classification results. The level of missing values was: 0, 0.01, 0.03, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5. In Tables 2, 3, and 4 we present the results of experiments for all considered data sets but with the selected levels of missing values: 0, 0.01, 0.05, 0.1, 0.3, 0.5.

Let us look at the results for the Parkinson data set. Starting from the level of missing values 0.05 the method *F* was the winning one. If it comes to the performance of aggregation operators it depended on the level of missing values. However, the operator $\mathcal{A}_5$ obtained the worst results (although in most of the cases still better than the methods *C* or *M*). We will now provide an analysis of the tested data sets and performance of the considered here methods. Banknote data set - starting from the level of missing values 0.01 and ending with 0.3 the method *F* was the wining one. Aggregations $\mathcal{A}_1$ - $\mathcal{A}_4$ or $\mathcal{A}_6$ had always the 5 top results (their position depended on the level of missing values). The aggregation $\mathcal{A}_5$ was the loosing one with respect to the methods *M* or *C* (it seems that $\mathcal{A}_5$ was not appropriate for the aggregation of this set of data). For the level of missing values 0.5 the best was the method *M* but the method *F* with aggregation operator $\mathcal{A}_6$ was at the second place. For no missing values case, the classification method *C* was the winner (for $k = 15, 20, 30$), the next was the method *M*, and then the method *F* with $\mathcal{A}_1$.

Biodeg data set - independently of the level of missing values the method *F* was the best one (with one of the aggregation operators applied). In the most cases of the levels of missing values, five of the aggregation operators $\mathcal{A}_1$ - $\mathcal{A}_4$ or $\mathcal{A}_6$ had the top 5 positions. Again the performance of the method *F* with $\mathcal{A}_5$ was worse comparing to other aggregation operators. Similar situation was with the data sets Red wine, Rethinopathy, Breast cancer and Diabetes, however in the last three data sets sometimes in the sequence of the top methods (for diverse levels of missing values), the methods *C* or *M* appeared among the winning ones. For the data sets German, Ozone and Spam, again among the winning methods were the *F* ones with the most frequent appearance but in these cases performance of $\mathcal{A}_5$ was much better and for the Ozone data set $\mathcal{A}_5$ was one of the best ones.

We see that the obtained results justify our approach of using the method *F* with the proposed idea of filling missing values together with aggregation of the obtained in this way intervals. Moreover, we see that the choice of the applied interval-valued aggregation function depends on the data set and also on the level of missing values in it. Furthermore, we may notice that it is worth to use diverse types of aggregation functions, not only the representable aggregation function $\mathcal{A}_1$ which is a natural extension of the arithmetic mean from [0,1] to L^I. In most of the cases (with respect to the data sets and missing value levels in them) other than $\mathcal{A}_1$ aggregation operators obtained the top results. Finally, the high values of AUC, regarding the increase of missing values, confirm very good performance of the applied method *F* in comparison to the methods *C* and *M*.

The observed results were justified with the use of statistical tests in the Statistica program [48]. We proved, in the case of each method, that the values of AUC are decreasing while the levels of missing values are increasing. It was verified with the Pearson correlation coefficient which measures the linear dependence between variables (cf. Table 5).

We considered the linear regression of changes in the average AUC values depending on the levels of missing values. We applied the test to check if the gradients of the lines are equal. The test has shown that on the level of $p < 0.05$ such equality holds. In Fig. 1 we present the linear regressions of changes for the average AUC levels depending on the levels of missing values.

The gradients of lines are the same but the line representing the *F* method is significantly higher than the lines representing the *C* and *M* methods and the line representing the *M* method is significantly higher than the line representing the *C* method (it was verified with the test of equality for *y*-intercepts of the lines which gave negative results). The highest and significantly different values are obtained for the *F* method (the *F* method gives on average a higher AUC value than the *C* method approximately by 0.05). Moreover, the statistically significant changes of the values AUC (regarding the levels of missing values) between pairs of methods were confirmed with the Wilcoxon test (cf. Table 6).

The applied tests proved that there are statistically significant changes between the *F* method and the *M* and *C* methods and analogously, between the *M* method and the *C* method (the analysis for the *F* method was performed with adequate aggregation operators and for the *C* method with adequate parameters *k*).

To sum up, we see that in the considered case of missing values application of interval modeling enabled much better results of classification than simple aggregation with the use of the numerical arithmetic mean. The superiority of the method *F* may be justified in the following way. It is connected with the mechanism of coping with missing values in the

Table 2
The average AUC for classification algorithms, part I.

Missing	Banknote		Biodeg		Breast cancer		Diabetes	
	Method	AUC	Method	AUC	Method	AUC	Method	AUC
0.0	C, k=15	1.0 ± 0.001	F, $\mathcal{A}_6$	0.903 ± 0.014	C, k=10	0.991 ± 0.006	M	0.805 ± 0.013
0.0	C, k=20	1.0 ± 0.0	F, $\mathcal{A}_1$	0.902 ± 0.007	C, k=15	0.991 ± 0.004	C, k=30	0.804 ± 0.018
0.0	C, k=30	1.0 ± 0.0	F, $\mathcal{A}_2$	0.902 ± 0.01	M	0.991 ± 0.005	F, $\mathcal{A}_6$	0.804 ± 0.015
0.0	M	1.0 ± 0.0	F, $\mathcal{A}_3$	0.902 ± 0.014	F, $\mathcal{A}_1$	0.991 ± 0.004	F, $\mathcal{A}_1$	0.8 ± 0.011
0.0	F, $\mathcal{A}_1$	1.0 ± 0.0	F, $\mathcal{A}_5$	0.901 ± 0.012	F, $\mathcal{A}_3$	0.99 ± 0.003	F, $\mathcal{A}_2$	0.8 ± 0.02
0.0	F, $\mathcal{A}_2$	1.0 ± 0.0	M	0.899 ± 0.008	C, k=5	0.989 ± 0.006	F, $\mathcal{A}_5$	0.8 ± 0.018
0.0	F, $\mathcal{A}_3$	1.0 ± 0.0	F, $\mathcal{A}_4$	0.895 ± 0.011	C, k=20	0.989 ± 0.005	F, $\mathcal{A}_3$	0.798 ± 0.011
0.0	F, $\mathcal{A}_4$	1.0 ± 0.0	C, k=10	0.891 ± 0.009	C, k=3	0.988 ± 0.004	F, $\mathcal{A}_4$	0.796 ± 0.017
0.0	F, $\mathcal{A}_5$	1.0 ± 0.0	C, k=15	0.891 ± 0.022	F, $\mathcal{A}_2$	0.988 ± 0.008	C, k=20	0.795 ± 0.017
0.0	F, $\mathcal{A}_6$	1.0 ± 0.0	C, k=30	0.89 ± 0.009	F, $\mathcal{A}_5$	0.988 ± 0.005	C, k=15	0.793 ± 0.021
0.0	C, k=3	0.999 ± 0.001	C, k=20	0.887 ± 0.01	F, $\mathcal{A}_6$	0.987 ± 0.006	C, k=10	0.785 ± 0.009
0.0	C, k=5	0.999 ± 0.001	C, k=5	0.886 ± 0.013	C, k=30	0.986 ± 0.009	C, k=5	0.762 ± 0.024
0.0	C, k=10	0.999 ± 0.001	C, k=3	0.87 ± 0.015	F, $\mathcal{A}_4$	0.986 ± 0.008	C, k=3	0.744 ± 0.019
0.0	C, k=1	0.998 ± 0.001	C, k=1	0.814 ± 0.013	C, k=1	0.946 ± 0.013	C, k=1	0.663 ± 0.018
0.05	F, $\mathcal{A}_1$	0.974 ± 0.015	F, $\mathcal{A}_4$	0.818 ± 0.049	F, $\mathcal{A}_2$	0.987 ± 0.005	F, $\mathcal{A}_1$	0.762 ± 0.013
0.05	F, $\mathcal{A}_3$	0.971 ± 0.02	F, $\mathcal{A}_1$	0.806 ± 0.058	C, k=15	0.985 ± 0.004	F, $\mathcal{A}_4$	0.752 ± 0.032
0.05	F, $\mathcal{A}_4$	0.971 ± 0.022	F, $\mathcal{A}_2$	0.79 ± 0.054	F, $\mathcal{A}_1$	0.985 ± 0.009	F, $\mathcal{A}_3$	0.749 ± 0.029
0.05	F, $\mathcal{A}_6$	0.97 ± 0.02	F, $\mathcal{A}_3$	0.79 ± 0.056	F, $\mathcal{A}_4$	0.985 ± 0.006	F, $\mathcal{A}_6$	0.748 ± 0.036
0.05	F, $\mathcal{A}_2$	0.969 ± 0.016	F, $\mathcal{A}_6$	0.785 ± 0.066	F, $\mathcal{A}_6$	0.985 ± 0.006	F, $\mathcal{A}_2$	0.746 ± 0.054
0.05	C, k=15	0.933 ± 0.037	F, $\mathcal{A}_5$	0.766 ± 0.035	M	0.984 ± 0.007	C, k=20	0.705 ± 0.031
0.05	C, k=30	0.93 ± 0.044	M	0.738 ± 0.058	F, $\mathcal{A}_3$	0.984 ± 0.006	M	0.703 ± 0.042
0.05	M	0.928 ± 0.041	C, k=10	0.736 ± 0.035	C, k=30	0.983 ± 0.007	C, k=15	0.693 ± 0.04
0.05	C, k=20	0.925 ± 0.044	C, k=15	0.725 ± 0.048	C, k=20	0.981 ± 0.005	C, k=10	0.689 ± 0.047
0.05	F, $\mathcal{A}_5$	0.92 ± 0.031	C, k=30	0.72 ± 0.07	C, k=10	0.979 ± 0.008	C, k=30	0.686 ± 0.045
0.05	C, k=10	0.905 ± 0.042	C, k=20	0.708 ± 0.059	F, $\mathcal{A}_5$	0.972 ± 0.01	F, $\mathcal{A}_5$	0.677 ± 0.038
0.05	C, k=5	0.903 ± 0.042	C, k=5	0.696 ± 0.055	C, k=5	0.969 ± 0.016	C, k=5	0.651 ± 0.039
0.05	C, k=3	0.887 ± 0.051	C, k=3	0.682 ± 0.052	C, k=3	0.949 ± 0.022	C, k=3	0.648 ± 0.042
0.05	C, k=1	0.868 ± 0.051	C, k=1	0.573 ± 0.055	C, k=1	0.89 ± 0.03	C, k=1	0.609 ± 0.035
0.1	F, $\mathcal{A}_6$	0.939 ± 0.037	F, $\mathcal{A}_3$	0.761 ± 0.058	F, $\mathcal{A}_2$	0.984 ± 0.005	F, $\mathcal{A}_2$	0.738 ± 0.052
0.1	F, $\mathcal{A}_3$	0.932 ± 0.043	F, $\mathcal{A}_4$	0.759 ± 0.06	F, $\mathcal{A}_1$	0.98 ± 0.01	F, $\mathcal{A}_3$	0.726 ± 0.04
0.1	F, $\mathcal{A}_1$	0.929 ± 0.041	F, $\mathcal{A}_2$	0.753 ± 0.079	F, $\mathcal{A}_3$	0.979 ± 0.007	F, $\mathcal{A}_1$	0.721 ± 0.039
0.1	F, $\mathcal{A}_4$	0.929 ± 0.039	F, $\mathcal{A}_1$	0.745 ± 0.065	F, $\mathcal{A}_6$	0.978 ± 0.009	F, $\mathcal{A}_6$	0.71 ± 0.054
0.1	F, $\mathcal{A}_2$	0.911 ± 0.048	F, $\mathcal{A}_6$	0.73 ± 0.083	C, k=1	0.976 ± 0.013	F, $\mathcal{A}_4$	0.704 ± 0.053
0.1	C, k=15	0.863 ± 0.064	C, k=15	0.698 ±0.045	F, $\mathcal{A}_4$	0.974 ± 0.013	C, k=30	0.657 ± 0.043
0.1	F, $\mathcal{A}_5$	0.853 ± 0.05	F, $\mathcal{A}_5$	0.692 ±0.043	C, k=20	0.973 ± 0.014	F, $\mathcal{A}_5$	0.652 ± 0.055
0.1	C, k=20	0.846 ± 0.062	C, k=5	0.679 ± 0.055	M	0.972 ± 0.014	C, k=10	0.643 ± 0.051
0.1	M	0.846 ± 0.076	M	0.674 ± 0.059	C, k=15	0.97 ± 0.012	C, k=20	0.638 ± 0.057
0.1	C, k=30	0.832 ± 0.081	C, k=20	0.667 ± 0.063	F, $\mathcal{A}_5$	0.958 ± 0.02	C, k=15	0.628 ± 0.048
0.1	C, k=5	0.823 ± 0.068	C, k=30	0.658 ± 0.083	C, k=10	0.957 ± 0.018	M	0.626 ± 0.043
0.1	C, k=3	0.817 ± 0.078	C, k=10	0.65 ± 0.08	C, k=5	0.956 ± 0.026	C, k=5	0.602 ± 0.076
0.1	C, k=10	0.815 ± 0.089	C, k=3	0.611 ± 0.068	C, k=3	0.892 ± 0.047	C, k=3	0.587 ± 0.047
0.1	C, k=1	0.776 ± 0.089	C, k=1	0.591 ± 0.033	C, k=1	0.785 ± 0.089	C, k=1	0.578 ± 0.03
0.3	F, $\mathcal{A}_6$	0.756 ± 0.107	F, $\mathcal{A}_2$	0.644 ± 0.082	F, $\mathcal{A}_3$	0.91 ± 0.058	F, $\mathcal{A}_4$	0.635 ± 0.056
0.3	F, $\mathcal{A}_3$	0.754 ± 0.101	F, $\mathcal{A}_3$	0.629 ± 0.097	F, $\mathcal{A}_6$	0.903 ± 0.06	F, $\mathcal{A}_3$	0.625 ± 0.078
0.3	F, $\mathcal{A}_4$	0.752 ± 0.107	F, $\mathcal{A}_4$	0.624 ± 0.099	F, $\mathcal{A}_1$	0.901 ± 0.061	F, $\mathcal{A}_1$	0.62 ± 0.08
0.3	F, $\mathcal{A}_2$	0.748 ± 0.11	F, $\mathcal{A}_6$	0.623 ± 0.093	F, $\mathcal{A}_4$	0.892 ± 0.072	F, $\mathcal{A}_6$	0.62 ± 0.066
0.3	F, $\mathcal{A}_1$	0.74 ± 0.118	F, $\mathcal{A}_1$	0.622 ± 0.087	F, $\mathcal{A}_2$	0.891 ± 0.074	F, $\mathcal{A}_2$	0.615 ± 0.083
0.3	C, k=30	0.681 ± 0.082	F, $\mathcal{A}_5$	0.595 ± 0.047	C, k=30	0.879 ± 0.063	C, k=15	0.606 ± 0.041
0.3	F, $\mathcal{A}_5$	0.681 ± 0.089	C, k=30	0.591 ± 0.061	M	0.853 ± 0.072	F, $\mathcal{A}_5$	0.591 ± 0.05
0.3	C, k=20	0.674 ± 0.065	C, k=20	0.59 ± 0.073	C, k=15	0.85 ± 0.066	C, k=10	0.574 ± 0.065
0.3	C, k=10	0.654 ± 0.083	C, k=10	0.587 ± 0.063	C, k=20	0.842 ± 0.078	C, k=30	0.568 ± 0.052
0.3	C, k=15	0.642 ± 0.091	M	0.58 ± 0.06	F, $\mathcal{A}_5$	0.832 ± 0.065	C, k=3	0.559 ± 0.026
0.3	M	0.637 ± 0.104	C, k=5	0.568 ± 0.047	C, k=10	0.818 ± 0.075	C, k=20	0.549 ± 0.044
0.3	C, k=3	0.629 ± 0.071	C, k=15	0.553 ± 0.061	C, k=5	0.78 ± 0.07	M	0.545 ± 0.05
0.3	C, k=5	0.614 ± 0.073	C, k=3	0.526 ± 0.02	C, k=3	0.711 ± 0.106	C, k=5	0.526 ± 0.027
0.3	C, k=1	0.587 ± 0.058	C, k=1	0.511 ± 0.031	C, k=1	0.629 ± 0.069	C, k=1	0.525 ± 0.031
0.5	M	0.655 ± 0.152	F, $\mathcal{A}_4$	0.581 ± 0.064	F, $\mathcal{A}_6$	0.766 ± 0.128	F, $\mathcal{A}_1$	0.572 ± 0.072
0.5	F, $\mathcal{A}_6$	0.641 ± 0.11	M	0.577 ± 0.073	F, $\mathcal{A}_4$	0.764 ± 0.141	F, $\mathcal{A}_3$	0.568 ± 0.055
0.5	C, k=30	0.64 ± 0.074	F, $\mathcal{A}_1$	0.575 ± 0.078	F, $\mathcal{A}_3$	0.761 ± 0.137	C, k=15	0.561 ± 0.041
0.5	F, $\mathcal{A}_3$	0.634 ± 0.092	F, $\mathcal{A}_3$	0.565 ± 0.083	C, k=30	0.753 ± 0.141	F, $\mathcal{A}_2$	0.56 ± 0.061
0.5	F, $\mathcal{A}_4$	0.631 ± 0.103	F, $\mathcal{A}_2$	0.561 ± 0.055	M	0.752 ± 0.103	F, $\mathcal{A}_4$	0.559 ± 0.066
0.5	F, $\mathcal{A}_1$	0.624 ± 0.098	F, $\mathcal{A}_6$	0.561 ± 0.058	F, $\mathcal{A}_1$	0.751 ± 0.135	F, $\mathcal{A}_6$	0.557 ± 0.056
0.5	F, $\mathcal{A}_2$	0.618 ± 0.102	C, k=30	0.55 ± 0.066	F, $\mathcal{A}_2$	0.75 ± 0.141	M	0.555 ± 0.045
0.5	C, k=15	0.604 ± 0.083	C, k=20	0.544 ± 0.043	C, k=5	0.724 ± 0.095	F, $\mathcal{A}_5$	0.554 ± 0.052
0.5	C, k=20	0.604 ± 0.085	C, k=15	0.535 ± 0.053	C, k=15	0.719 ± 0.092	C, k=20	0.53 ± 0.029
0.5	C, k=10	0.592 ± 0.051	F, $\mathcal{A}_5$	0.53 ± 0.038	F, $\mathcal{A}_5$	0.705 ± 0.085	C, k=3	0.525 ± 0.033

(*continued on next page*)

Table 2 (*continued*)

Missing	Banknote		Biodeg		Breast cancer		Diabetes	
	Method	AUC	Method	AUC	Method	AUC	Method	AUC
0.5	F, $\mathcal{A}_5$	0.576 ± 0.084	C, k=10	0.507 ± 0.078	C, k=20	0.699 ± 0.12	C, k=5	0.524 ± 0.037
0.5	C, k=5	0.568 ± 0.068	C, k=5	0.503 ± 0.026	C, k=10	0.692 ± 0.085	C, k=10	0.52 ± 0.042
0.5	C, k=1	0.541 ± 0.04	C, k=1	0.501 ± 0.006	C, k=3	0.601 ± 0.095	C, k=1	0.515 ± 0.02
0.5	C, k=3	0.539 ± 0.038	C, k=3	0.497 ± 0.024	C, k=1	0.578 ± 0.061	C, k=30	0.514 ± 0.04

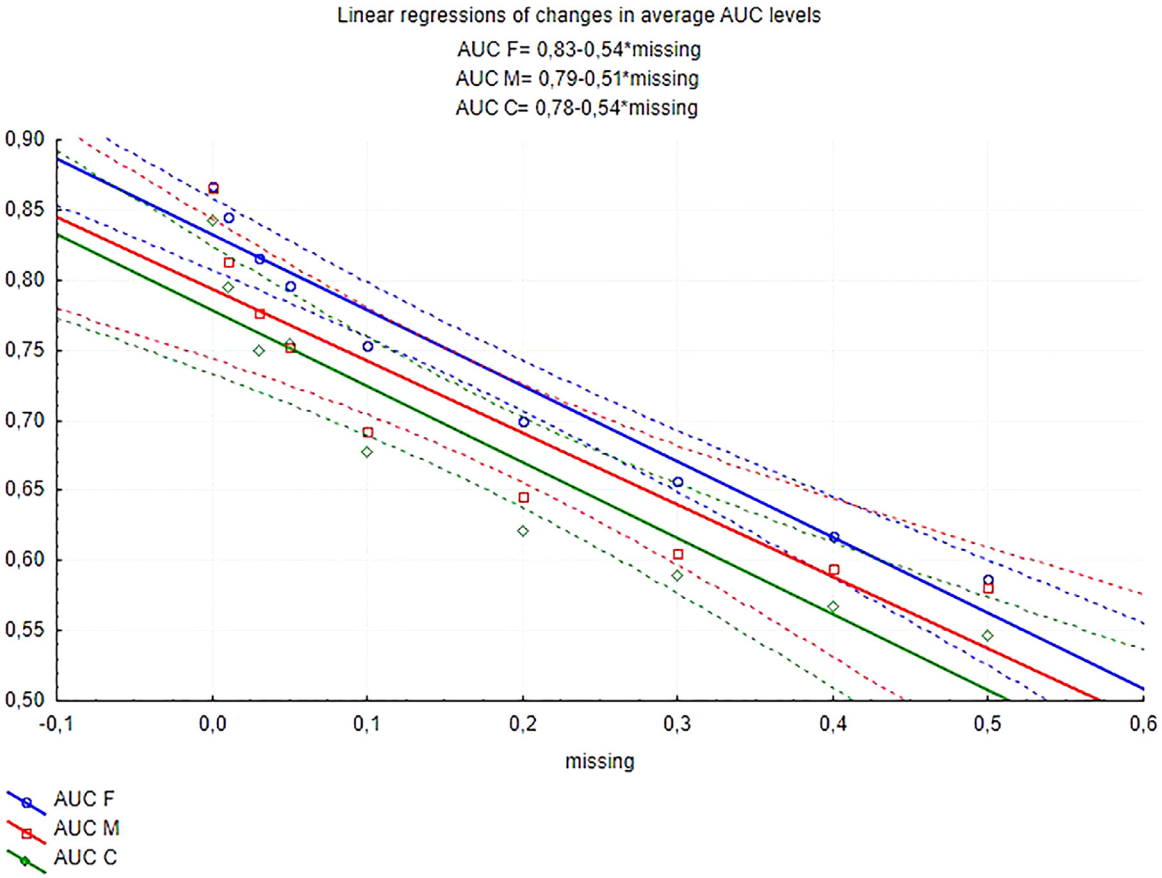

Fig. 1. Changes in average AUC depending on the levels of missing values.

presented methods. As it was mentioned in Section 3.2, in the WEKA API library (cf. [27,29]), this mechanism works when calculating the distance between the values on a given attribute. The difference between the numerical values of the two objects (only numerical values occurred in the analyzed data sets) is calculated as the absolute value of the difference of these values. If both values of the attributes are given, then the absolute value of the difference is determined for these values. If both values of the attributes are missing, then their difference is equal to 1. If one of the values of attributes is missing and the other one is equal to v, then the difference is taken as the value $\max(v, 1-v)$, as a result this difference is as high as possible. Therefore, we see that when classifying objects by the classical $k-NN$ method, if the test objects have missing values of attributes, then it results in a sense in "moving" these objects from the training objects (without missing values). As a result, the test object with a large number of missing values may be too strongly moved away from the training objects and it can get closer to the training objects from the wrong class. It makes this object wrong classified in the method C. While in the method F, by randomly filling missing values, we obtain objects that are less distanced from the proper training objects. Therefore, the F method achieves better results than both the C method and the M method (which is based on aggregating with the use of the arithmetic mean the values obtained in the C method).

Table 3
The average AUC for classification algorithms, part II.

Missing	German		Ozone		Parkinson	
	Method	AUC	Method	AUC	Method	AUC
0.0	C, k=30	0.752 ± 0.028	C, k=15	0.82 ± 0.042	C, k=30	1.0 ± 0.0
0.0	M	0.742 ± 0.01	F, $\mathcal{A}_4$	0.816 ± 0.036	M	1.0 ± 0.0
0.0	F, $\mathcal{A}_4$	0.736 ± 0.01	C, k=20	0.811 ± 0.049	F, $\mathcal{A}_1$	1.0 ± 0.0
0.0	F, $\mathcal{A}_3$	0.735 ± 0.015	F, $\mathcal{A}_2$	0.809 ± 0.048	F, $\mathcal{A}_2$	1.0 ± 0.001
0.0	F, $\mathcal{A}_5$	0.735 ± 0.017	F, $\mathcal{A}_3$	0.805 ± 0.023	F, $\mathcal{A}_3$	1.0 ± 0.0
0.0	F, $\mathcal{A}_1$	0.734 ± 0.017	F, $\mathcal{A}_6$	0.805 ± 0.031	F, $\mathcal{A}_4$	1.0 ± 0.0
0.0	C, k=15	0.733 ± 0.014	F, $\mathcal{A}_5$	0.788 ± 0.024	F, $\mathcal{A}_5$	1.0 ± 0.0
0.0	C, k=20	0.733 ± 0.012	F, $\mathcal{A}_1$	0.785 ± 0.041	F, $\mathcal{A}_6$	1.0 ± 0.0
0.0	F, $\mathcal{A}_2$	0.731 ± 0.018	C, k=30	0.784 ± 0.038	C, k=5	0.999 ± 0.001
0.0	F, $\mathcal{A}_6$	0.728 ± 0.023	M	0.777 ± 0.034	C, k=10	0.999 ± 0.001
0.0	C, k=10	0.722 ± 0.016	C, k=10	0.763 ± 0.034	C, k=15	0.999 ± 0.0
0.0	C, k=5	0.69 ± 0.024	C, k=5	0.692 ± 0.028	C, k=20	0.999 ± 0.0
0.0	C, k=3	0.676 ± 0.021	C, k=3	0.662 ± 0.022	C, k=3	0.997 ± 0.002
0.0	C, k=1	0.619 ± 0.018	C, k=1	0.579 ± 0.02	C, k=1	0.985 ± 0.004
0.05	F, $\mathcal{A}_1$	0.729 ± 0.023	F, $\mathcal{A}_6$	0.755 ± 0.104	F, $\mathcal{A}_2$	0.98 ± 0.017
0.05	C, k=20	0.712 ± 0.011	F, $\mathcal{A}_1$	0.75 ± 0.118	F, $\mathcal{A}_1$	0.979 ± 0.014
0.05	F, $\mathcal{A}_5$	0.712 ± 0.03	F, $\mathcal{A}_2$	0.735 ± 0.059	F, $\mathcal{A}_4$	0.974 ± 0.022
0.05	F, $\mathcal{A}_3$	0.708 ± 0.02	F, $\mathcal{A}_3$	0.715 ± 0.126	F, $\mathcal{A}_3$	0.972 ± 0.02
0.05	F, $\mathcal{A}_6$	0.704 ± 0.028	F, $\mathcal{A}_5$	0.698 ± 0.129	F, $\mathcal{A}_6$	0.972 ± 0.015
0.05	M	0.703 ± 0.025	F, $\mathcal{A}_4$	0.661 ± 0.124	C, k=30	0.939 ± 0.031
0.05	F, $\mathcal{A}_4$	0.701 ± 0.047	C, k=30	0.653 ± 0.092	M	0.934 ± 0.022
0.05	C, k=30	0.699 ± 0.04	M	0.645 ± 0.049	C, k=20	0.929 ± 0.04
0.05	C, k=15	0.693 ± 0.028	C, k=15	0.636 ± 0.075	C, k=15	0.926 ± 0.039
0.05	F, $\mathcal{A}_2$	0.686 ± 0.039	C, k=20	0.629 ± 0.046	F, $\mathcal{A}_5$	0.922 ± 0.021
0.05	C, k=10	0.674 ± 0.038	C, k=5	0.588 ± 0.067	C, k=5	0.917 ± 0.033
0.05	C, k=5	0.643 ± 0.029	C, k=10	0.553 ± 0.07	C, k=10	0.911 ± 0.036
0.05	C, k=3	0.622 ± 0.026	C, k=3	0.519 ± 0.032	C, k=3	0.883 ± 0.044
0.05	C, k=1	0.602 ± 0.031	C, k=1	0.503 ± 0.006	C, k=1	0.816 ± 0.083
0.1	F, $\mathcal{A}_1$	0.698 ± 0.04	F, $\mathcal{A}_5$	0.764 ± 0.096	F, $\mathcal{A}_4$	0.937 ± 0.042
0.1	F, $\mathcal{A}_4$	0.69 ± 0.033	F, $\mathcal{A}_3$	0.668 ± 0.109	F, $\mathcal{A}_2$	0.935 ± 0.049
0.1	F, $\mathcal{A}_6$	0.687 ± 0.034	F, $\mathcal{A}_4$	0.656 ± 0.113	F, $\mathcal{A}_1$	0.934 ± 0.04
0.1	F, $\mathcal{A}_3$	0.683 ± 0.044	C, k=15	0.634 ± 0.087	F, $\mathcal{A}_3$	0.93 ± 0.046
0.1	C, k=30	0.669 ± 0.059	C, k=20	0.626 ± 0.078	F, $\mathcal{A}_6$	0.927 ± 0.053
0.1	F, $\mathcal{A}_5$	0.669 ± 0.035	F, $\mathcal{A}_1$	0.62 ± 0.118	C, k=15	0.877 ± 0.053
0.1	M	0.667 ± 0.045	F, $\mathcal{A}_5$	0.601 ± 0.12	F, $\mathcal{A}_5$	0.877 ± 0.037
0.1	C, k=15	0.662 ± 0.047	C, k=5	0.58 ± 0.057	C, k=30	0.869 ± 0.061
0.1	C, k=20	0.661 ± 0.041	F, $\mathcal{A}_6$	0.563 ± 0.099	C, k=20	0.865 ± 0.063
0.1	F, $\mathcal{A}_2$	0.658 ± 0.04	C, k=10	0.548 ± 0.05	C, k=10	0.86 ± 0.06
0.1	C, k=10	0.644 ± 0.042	C, k=30	0.539 ± 0.048	M	0.859 ± 0.048
0.1	C, k=5	0.631 ± 0.043	M	0.519 ± 0.038	C, k=5	0.841 ± 0.055
0.1	C, k=3	0.587 ± 0.034	C, k=3	0.518 ± 0.012	C, k=3	0.782 ± 0.088
0.1	C, k=1	0.552 ± 0.04	C, k=1	0.505 ± 0.01	C, k=1	0.707 ± 0.134
0.3	C, k=20	0.61 ± 0.061	F, $\mathcal{A}_6$	0.653 ± 0.114	F, $\mathcal{A}_1$	0.783 ± 0.113
0.3	C, k=30	0.61 ± 0.057	F, $\mathcal{A}_5$	0.613 ± 0.095	F, $\mathcal{A}_6$	0.767 ± 0.129
0.3	F, $\mathcal{A}_1$	0.604 ± 0.067	F, $\mathcal{A}_3$	0.606 ± 0.119	F, $\mathcal{A}_3$	0.764 ± 0.111
0.3	F, $\mathcal{A}_2$	0.603 ± 0.058	F, $\mathcal{A}_2$	0.579 ± 0.1	F, $\mathcal{A}_4$	0.755 ± 0.133
0.3	F, $\mathcal{A}_4$	0.602 ± 0.066	F, $\mathcal{A}_4$	0.57 ± 0.068	F, $\mathcal{A}_2$	0.74 ± 0.16
0.3	F, $\mathcal{A}_3$	0.599 ± 0.062	F, $\mathcal{A}_1$	0.554 ± 0.067	C, k=30	0.72 ± 0.094
0.3	C, k=15	0.596 ± 0.051	C, k=20	0.518 ± 0.024	F, $\mathcal{A}_5$	0.704 ± 0.075
0.3	M	0.594 ± 0.065	C, k=3	0.513 ± 0.043	C, k=15	0.697 ± 0.104
0.3	F, $\mathcal{A}_6$	0.594 ± 0.068	C, k=30	0.513 ± 0.025	C, k=10	0.686 ± 0.106
0.3	F, $\mathcal{A}_5$	0.587 ± 0.064	C, k=10	0.507 ± 0.02	C, k=20	0.68 ± 0.113
0.3	C, k=10	0.579 ± 0.056	C, k=15	0.507 ± 0.016	M	0.676 ± 0.12
0.3	C, k=3	0.564 ± 0.045	M	0.507 ± 0.015	C, k=3	0.657 ± 0.097
0.3	C, k=5	0.551 ± 0.068	C, k=1	0.5 ± 0.0	C, k=5	0.655 ± 0.11
0.3	C, k=1	0.529 ± 0.025	C, k=5	0.5 ± 0.001	C, k=1	0.592 ± 0.071
0.5	F, $\mathcal{A}_5$	0.57 ± 0.059	F, $\mathcal{A}_2$	0.532 ± 0.041	F, $\mathcal{A}_1$	0.667 ± 0.116
0.5	F, $\mathcal{A}_6$	0.564 ± 0.069	F, $\mathcal{A}_3$	0.532 ± 0.075	F, $\mathcal{A}_2$	0.665 ± 0.133
0.5	F, $\mathcal{A}_2$	0.562 ± 0.074	F, $\mathcal{A}_1$	0.523 ± 0.025	F, $\mathcal{A}_3$	0.658 ± 0.151
0.5	F, $\mathcal{A}_3$	0.561 ± 0.064	F, $\mathcal{A}_4$	0.52 ± 0.027	F, $\mathcal{A}_4$	0.656 ± 0.124
0.5	M	0.56 ± 0.07	F, $\mathcal{A}_6$	0.515 0.029	F, $\mathcal{A}_6$	0.636 ± 0.136
0.5	F, $\mathcal{A}_1$	0.558 ± 0.067	F, $\mathcal{A}_5$	0.508 ± 0.036	M	0.621 ± 0.117
0.5	C, k=30	0.557 ± 0.06	C, k=20	0.502 ± 0.007	C, k=5	0.617 ± 0.088
0.5	F, $\mathcal{A}_4$	0.551 ± 0.057	C, k=3	0.501 ± 0.004	F, $\mathcal{A}_5$	0.594 ± 0.075
0.5	C, k=10	0.546 ± 0.053	C, k=30	0.501 ± 0.003	C, k=15	0.588 ± 0.092
0.5	C, k=20	0.541 ± 0.064	M	0.501 ± 0.004	C, k=10	0.587 ± 0.104
0.5	C, k=15	0.537 ± 0.05	C, k=1	0.5 ± 0.0	C, k=20	0.587 ± 0.102

(*continued on next page*)

Table 3 (*continued*)

Missing	German		Ozone		Parkinson	
	Method	AUC	Method	AUC	Method	AUC
0.5	C, k=3	0.534 ± 0.037	C, k=5	0.5 ± 0.001	C, k=30	0.581 ± 0.151
0.5	C, k=5	0.53 ± 0.045	C, k=10	0.5 ± 0.001	C, k=1	0.526 ± 0.035
0.5	C, k=1	0.518 ± 0.021	C, k=15	0.5 ± 0.0	C, k=3	0.518 ± 0.117

Table 4
The average AUC for classification algorithms, part III.

Missing	Rethinopathy		Red wine		Spam	
	Method	AUC	Method	AUC	Method	AUC
0.0	F, $\mathcal{A}_1$	0.705 ± 0.012	F, $\mathcal{A}_3$	0.806 ± 0.016	F, $\mathcal{A}_3$	0.951 ± 0.005
0.0	F, $\mathcal{A}_4$	0.703 ± 0.013	M	0.805 ± 0.009	F, $\mathcal{A}_4$	0.951 ± 0.004
0.0	F, $\mathcal{A}_3$	0.699 ± 0.017	F, $\mathcal{A}_4$	0.804 ± 0.012	M	0.95 ± 0.003
0.0	F, $\mathcal{A}_5$	0.698 ± 0.018	C, k=30	0.802 ± 0.01	F, $\mathcal{A}_2$	0.95 ± 0.004
0.0	F, $\mathcal{A}_6$	0.696 ± 0.014	F, $\mathcal{A}_1$	0.802 ± 0.012	F, $\mathcal{A}_1$	0.949 ± 0.003
0.0	C, k=15	0.695 ± 0.019	F, $\mathcal{A}_2$	0.801 ± 0.015	F, $\mathcal{A}_5$	0.948 ± 0.004
0.0	F, $\mathcal{A}_2$	0.693 ± 0.02	F, $\mathcal{A}_6$	0.798 ± 0.016	F, $\mathcal{A}_6$	0.948 ± 0.005
0.0	M	0.692 ± 0.014	F, $\mathcal{A}_5$	0.796 ± 0.01	C, k=10	0.945 ± 0.003
0.0	C, k=30	0.686 ± 0.013	C, k=20	0.795 ± 0.01	C, k=15	0.942 ± 0.004
0.0	C, k=10	0.679 ± 0.02	C, k=15	0.793 ± 0.008	C, k=5	0.941 ± 0.005
0.0	C, k=20	0.676 ± 0.025	C, k=10	0.789 ± 0.013	C, k=30	0.94 ± 0.004
0.0	C, k=5	0.67 ± 0.019	C, k=5	0.766 ± 0.016	C, k=20	0.939 ± 0.005
0.0	C, k=3	0.656 ± 0.016	C, k=3	0.754 ± 0.008	C, k=3	0.932 ± 0.006
0.0	C, k=1	0.608 ± 0.013	C, k=1	0.721 ± 0.014	C, k=1	0.883 ± 0.008
0.05	F, $\mathcal{A}_6$	0.661 ± 0.035	F, $\mathcal{A}_2$	0.757 ± 0.029	F, $\mathcal{A}_3$	0.72 ± 0.045
0.05	F, $\mathcal{A}_2$	0.644 ± 0.063	F, $\mathcal{A}_6$	0.756 ± 0.017	F, $\mathcal{A}_4$	0.712 ± 0.05
0.05	F, $\mathcal{A}_3$	0.642 ± 0.03	F, $\mathcal{A}_1$	0.753 ± 0.024	F, $\mathcal{A}_6$	0.711 ± 0.067
0.05	F, $\mathcal{A}_1$	0.633 ± 0.036	F, $\mathcal{A}_3$	0.75 ± 0.023	F, $\mathcal{A}_2$	0.706 ± 0.058
0.05	F, $\mathcal{A}_4$	0.62 ± 0.052	F, $\mathcal{A}_4$	0.745 ± 0.022	F, $\mathcal{A}_1$	0.702 ± 0.054
0.05	M	0.609 ± 0.026	F, $\mathcal{A}_5$	0.725 ± 0.033	F, $\mathcal{A}_5$	0.702 ± 0.052
0.05	C, k=10	0.599 ± 0.044	C, k=30	0.687 ± 0.042	M	0.619 ± 0.015
0.05	C, k=5	0.591 ± 0.031	C, k=20	0.678 ± 0.042	C, k=15	0.614 ± 0.026
0.05	F, $\mathcal{A}_5$	0.585 ± 0.052	C, k=10	0.67 ± 0.029	C, k=10	0.609 ± 0.023
0.05	C, k=15	0.582 ± 0.043	C, k=15	0.657 ± 0.041	C, k=20	0.608 ± 0.014
0.05	C, k=20	0.581 ± 0.052	M	0.656 ± 0.051	C, k=30	0.602 ± 0.033
0.05	C, k=30	0.575 ± 0.056	C, k=5	0.647 ± 0.052	C, k=5	0.577 ± 0.026
0.05	C, k=3	0.568 ± 0.03	C, k=3	0.631 ± 0.049	C, k=3	0.575 ± 0.025
0.05	C, k=1	0.546 ± 0.032	C, k=1	0.601 ± 0.041	C, k=1	0.567 ± 0.013
0.1	F, $\mathcal{A}_5$	0.615 ± 0.039	F, $\mathcal{A}_6$	0.72 ± 0.026	F, $\mathcal{A}_2$	0.664 ± 0.051
0.1	F, $\mathcal{A}_2$	0.611 ± 0.042	F, $\mathcal{A}_3$	0.714 ± 0.051	F, $\mathcal{A}_1$	0.661 ± 0.055
0.1	F, $\mathcal{A}_1$	0.609 ± 0.049	F, $\mathcal{A}_2$	0.71 ± 0.042	F, $\mathcal{A}_5$	0.652 ± 0.045
0.1	F, $\mathcal{A}_3$	0.597 ± 0.066	F, $\mathcal{A}_4$	0.702 ± 0.05	F, $\mathcal{A}_6$	0.639 ± 0.054
0.1	F, $\mathcal{A}_4$	0.583 ± 0.052	F, $\mathcal{A}_1$	0.697 ± 0.066	F, $\mathcal{A}_4$	0.629 ± 0.048
0.1	C, k=20	0.576 ± 0.038	F, $\mathcal{A}_5$	0.68 ± 0.047	F, $\mathcal{A}_3$	0.628 ± 0.091
0.1	F, $\mathcal{A}_6$	0.57 ± 0.065	C, k=20	0.648 ± 0.052	M	0.584 ± 0.024
0.1	C, k=15	0.557 ± 0.048	M	0.634 ± 0.027	C, k=15	0.581 ± 0.031
0.1	C, k=10	0.556 ± 0.03	C, k=30	0.631 ± 0.04	C, k=5	0.576 ± 0.019
0.1	C, k=30	0.554 ± 0.038	C, k=15	0.623 ± 0.045	C, k=30	0.57 ± 0.031
0.1	C, k=3	0.547 ± 0.028	C, k=10	0.62 ± 0.053	C, k=10	0.568 ± 0.039
0.1	M	0.543 ± 0.048	C, k=3	0.613 ± 0.03	C, k=20	0.565 ± 0.022
0.1	C, k=5	0.535 ± 0.03	C, k=5	0.606 ± 0.046	C, k=3	0.554 ± 0.017
0.1	C, k=1	0.526 ± 0.018	C, k=1	0.568 ± 0.055	C, k=1	0.538 ± 0.02
0.3	F, $\mathcal{A}_1$	0.561 ± 0.049	F, $\mathcal{A}_3$	0.613 ± 0.061	F, $\mathcal{A}_5$	0.627 ± 0.049
0.3	F, $\mathcal{A}_6$	0.561 ± 0.054	F, $\mathcal{A}_1$	0.608 ± 0.075	F, $\mathcal{A}_1$	0.6 ± 0.038
0.3	F, $\mathcal{A}_3$	0.558 ± 0.034	F, $\mathcal{A}_4$	0.607 ± 0.06	F, $\mathcal{A}_3$	0.599 ± 0.048
0.3	F, $\mathcal{A}_4$	0.55 ± 0.036	F, $\mathcal{A}_2$	0.604 ± 0.045	F, $\mathcal{A}_6$	0.591 ± 0.061
0.3	F, $\mathcal{A}_2$	0.546 ± 0.035	F, $\mathcal{A}_6$	0.601 ± 0.052	F, $\mathcal{A}_4$	0.589 ± 0.045
0.3	F, $\mathcal{A}_5$	0.544 ± 0.039	F, $\mathcal{A}_5$	0.584 ± 0.074	F, $\mathcal{A}_2$	0.58 ± 0.046
0.3	C, k=30	0.524 ± 0.022	M	0.558 ± 0.03	M	0.575 ± 0.05
0.3	M	0.524 ± 0.031	C, k=3	0.552 ± 0.05	C, k=5	0.565 ± 0.043
0.3	C, k=10	0.521 ± 0.019	C, k=5	0.549 ± 0.033	C, k=10	0.546 ± 0.031
0.3	C, k=15	0.521 ± 0.024	C, k=30	0.545 ± 0.036	C, k=20	0.543 ± 0.042
0.3	C, k=20	0.52 ± 0.02	C, k=15	0.543 ± 0.05	C, k=15	0.542 ± 0.033
0.3	C, k=3	0.513 ± 0.011	C, k=10	0.54 ± 0.035	C, k=30	0.537 ± 0.019
0.3	C, k=5	0.5 ± 0.024	C, k=1	0.535 ± 0.029	C, k=3	0.531 ± 0.029
0.3	C, k=1	0.496 ± 0.01	C, k=20	0.518 ± 0.044	C, k=1	0.528 ± 0.024

(*continued on next page*)

Table 4 (*continued*)

Missing	Rethinopathy		Red wine		Spam	
	Method	AUC	Method	AUC	Method	AUC
0.5	F, $\mathcal{A}_2$	0.548 ± 0.042	F, $\mathcal{A}_2$	0.582 ± 0.063	F, $\mathcal{A}_6$	0.589 ± 0.108
0.5	F, $\mathcal{A}_1$	0.524 ± 0.029	F, $\mathcal{A}_1$	0.571 ± 0.061	F, $\mathcal{A}_2$	0.582 ± 0.036
0.5	C, k=10	0.522 ± 0.017	F, $\mathcal{A}_3$	0.57 ± 0.045	F, $\mathcal{A}_4$	0.566 ± 0.06
0.5	F, $\mathcal{A}_3$	0.521 ± 0.023	F, $\mathcal{A}_5$	0.554 ± 0.057	F, $\mathcal{A}_5$	0.566 ± 0.075
0.5	C, k=5	0.516 ± 0.013	F, $\mathcal{A}_6$	0.545 ± 0.053	M	0.558 ± 0.062
0.5	C, k=30	0.514 ± 0.023	F, $\mathcal{A}_4$	0.544 ± 0.061	C, k=20	0.547 ± 0.029
0.5	C, k=3	0.511 ± 0.013	C, k=5	0.529 ± 0.022	C, k=5	0.542 ± 0.044
0.5	C, k=15	0.511 ± 0.021	M	0.528 ± 0.036	F, $\mathcal{A}_3$	0.542 ± 0.075
0.5	F, $\mathcal{A}_4$	0.51 ± 0.033	C, k=15	0.527 ± 0.038	C, k=15	0.536 ± 0.05
0.5	F, $\mathcal{A}_5$	0.507 ± 0.028	C, k=20	0.527 ± 0.039	F, $\mathcal{A}_1$	0.533 ± 0.075
0.5	M	0.504 ± 0.027	C, k=30	0.525 ± 0.032	C, k=10	0.521 ± 0.021
0.5	C, k=1	0.501 ± 0.01	C, k=3	0.515 ± 0.021	C, k=30	0.501 ± 0.035
0.5	C, k=20	0.499 ± 0.024	C, k=1	0.512 ± 0.016	C, k=1	0.498 ± 0.028
0.5	F, $\mathcal{A}_6$	0.496 ± 0.032	C, k=10	0.509 ± 0.025	C, k=3	0.474 ± 0.068

Table 5
Pearson correlation coefficient.

Method	Coefficient
F	-0.6493
M	-0.5890
C	-0.6000

Table 6
The p value in the Wilcoxon test.

Pair of methods	The p value
F, M	0.0076
F, C	0.0076
M, C	0.0108

5. Conclusions

In this contribution it was shown that, despite appearance of missing values, application of interval modeling and aggregation methods causes much slower decrease of the classification quality comparing to the classical version of the $k - NN$ classifier. It was shown that diverse interval-valued aggregation operators enable to choose the one which is the most suitable for a given data set and a level of missing values. Moreover, a new method of dealing with missing values was proposed with satisfactory results.

The results of the experiments are an empirical proof of better quality of the new methods. From theoretical point of view, two reasons can be given for which the proposed method F may be better than others. Firstly, the proposed method has more knowledge of objects at the classification than the classical $k - NN$ method does. This is due to the fact that classification results are used for a number of classifiers at different k. It is generally known that there is no way to determine optimal value of k for all data. For each data, optimal k must be determined through many experiments. In this method, different k are examined simultaneously, which immediately reveals the knowledge about how the test object is evaluated from the point of view of different k. A separate issue is how to use this knowledge, i.e. how to aggregate it, to extract the most correct (in line with reality) conclusions regarding the decision class to which the test object belongs. The work proposes two aggregation approaches, namely with and without using interval-valued fuzzy calculus. Both approaches give good results, although the method with interval-valued fuzzy calculus involved is a better one.

Secondly, the proposed method uses interval-valued fuzzy methods to aggregate uncertainty intervals, not the values of membership functions (single numerical values). Therefore, there is more knowledge in this method at the beginning. As the next step, this knowledge is aggregated to make the final decision about the belonging of the test object to a decision class. As a result, the classifier F, with the interval-valued methods involved, has a better chance of making a good decision because it has more knowledge. However, the aggregation must be done in such a way that the quality improvement in the classification is obtained. The methods of aggregation of intervals that are proposed in the work are based on human intuitions how to aggregate information from various sources (this information is represented in the form of uncertainty intervals). Practical experiments confirmed the usefulness of these intuitions, because the quality of classification has increased.

To sum up, the presented method F has a chance to be better than the previous one, because it uses more knowledge discovered from data than previous methods did and uses aggregation mechanisms that have injected domain knowledge corresponding to the intuitions used by people in such situations.

Furthermore, analyzing the properties of specific data sets the following should be mentioned. Firstly, both the M and F methods will work better than the C method does for such data sets that have more empty spaces. In other words, the more missing values in data, the better the superiority of M and F over C and F over M. Secondly, the M and F methods will work better for data where there is a spatial diversity of the distribution of decision classes depending on the size of the environment. This is manifested in the fact that the same test object will often be classified into different decision classes depending on the number of nearest neighbors used in the $k - NN$.

For the future work, we plan to consider in the experiments more sophisticated examples of interval-valued aggregation operators and as for the $k - NN$ algorithm other distances than the Euclidean one. Moreover, we would also like to consider multi-valued classifiers.

Acknowledgements

This work was partially supported by the following grant of the Polish National Science Centre DEC-2013/09/B/NZ5/00758 and also by the Centre for Innovation and Transfer of Natural Sciences and Engineering Knowledge of University of Rzeszów, Poland.

References

[1] T. Bailey, A. Jain, A note on distance-weighted k-nearest neighbor rules, IEEE Trans. Systems, Man, Cybernetics 8 (1978) 311–313.
[2] J.G. Bazan, Hierarchical Classifiers for Complex Spatio-temporal Concepts, in: Transactions on Rough Sets IX, Springer, 2008, pp. 474–750.
[3] J.G. Bazan, S. Buregwa-Czuma, A. Jankowski, A domain knowledge as a tool for improving classifiers, Fundam Inform 127 (1–4) (2013) 495–511.
[4] J.G. Bazan, S. Bazan-Socha, S. Buregwa-Czuma, L. Dydo, W. Rzasa, A. Skowron, A classifier based on a decision tree with verifying cuts, Fundam Inform 143 (1–2) (2016) 1–18.
[5] S. Buregwa-Czuma, J.G. Bazan, S. Bazan-Socha, W. Rzasa, L. Dydo, A. Skowron, Resolving the Conflicts between Cuts in a Decision Tree with Verifying Cuts, in: Proceedings of IJCRS 2017, Olsztyn 3–7 July, Lecture Notes in Computer Science (LNCS), 10314, Springer, 2017, pp. 403–422. (The Best Application Paper Award)
[6] G. Beliakov, H. Bustince, T. Calvo, A practical guide to averaging functions, Studies in Fuzziness and Soft Computing, Springer International Publishing Switzerland, 2016.
[7] U. Bentkowska, New types of aggregation functions for interval-valued fuzzy setting and preservation of pos-b and nec-b-transitivity in decision making problems, Inform. Sci. 424 (2018) 385–399.
[8] U. Bentkowska, B. Pękala, Diverse Classes of Interval-valued Aggregation Functions in Medical Diagnosis Support, in: J. Medina, et al. (Eds.), IPMU 2018, CCIS 853, pp. 661–672, Springer International Publishing AG, part of Springer Nature, 2018.
[9] S. Bermejo, J. Cabestany, Adaptive soft k-nearest-neighbour classifiers, Pattern Recognit 33 (2000) 1999–2005.
[10] P.S. Bullen, D.S. Mitrinović, P.M. Vasić, Means and their inequalities, Reidel, Dordrecht, 1988.
[11] H. Bustince, J. Fernandez, A. Kolesárová, R. Mesiar, Generation of linear orders for intervals by means of aggregation functions, Fuzzy Sets Syst. 220 (2013) 69–77.
[12] T. Calvo, A. Kolesárová, M. Komorniková, R. Mesiar, Aggregation Operators: Properties, Classes and Construction Methods, in: T. Calvo, et al. (Eds.), Aggregation Operators, Physica-Verlag, Heidelberg, 2002, pp. 3–104.
[13] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE trans, Inform. Theory, IT 13 (1) (1967) 21–27.
[14] K. Dembczyński, S. Greco, R. Słowiński, Rough set approach to multiple criteria classification with imprecise evaluations and assignments, Eur J Oper Res 198 (2009) 626–636.
[15] G. Deschrijver, Arithmetic operators in interval-valued fuzzy set theory, Inform. Sci. 177 (2007) 2906–2924.
[16] G. Deschrijver, Quasi-arithmetic Means and OWA Functions in Interval-valued and Atanassov's Intuitionistic Fuzzy Set Theory, in: S. Galichet, et al. (Eds.), Proceedings of EUSFLAT-LFA 2011, 18–22.07.2011, Aix-les-Bains, France, 2011, pp. 506–513.
[17] T. De Waal, J. Pannekoek, S. Scholtus, Handbook of statistical data editing and imputation, John Wiley and Sons vol. 563 (2011).
[18] G.P. Dimuro, G. Lucca, J.A. Sanz, H. Bustince, B. Bedregal, CMin-integral: A Choquet-like Aggregation Function Based on the Minimum T-norm for Applications to Fuzzy Rule-based Classification Systems, in: V. Torra, et al. (Eds.), Aggregation Functions in Theory and in Practice, Advances in Intelligent Systems and Computing 581, Springer AG Cham, 2018, pp. 83–95.
[19] D. Dubois, H. Prade, Possibility theory, Plenum Press, New York, 1988.
[20] D. Dubois, H. Prade, Gradualness, uncertainty and bipolarity: making sense of fuzzy sets, Fuzzy Sets Syst. 192 (2012) 3–24.
[21] S.A. Dudani, The distance-weighted k-nearest-neighbor rule, IEEE Trans. Syst. Man Cybern., SMC 6 (1976) 325–327.
[22] K. Dyczkowski, A. Wójtowicz, P. Żywica, A. Stachowiak, R. Moszyński, S. Szubert, An Intelligent System for Computer-aided Ovarian Tumor Diagnosis, in: Intelligent Systems'2014, Springer International Publishing, 2015, pp. 335–344.
[23] K. Dyczkowski, Intelligent Medical Decision Support System Based on Imperfect Information, The Case of Ovarian Tumor Diagnosis, Studies in Computational Intelligence, Springer, 2018.
[24] T. Fawcett, An introduction to ROC analysis, Pattern Recognit Lett 27 (8) (2006) 861–874.
[25] E. Fix, J.L. Hodges, Discriminatory analysis, nonparametric discrimination: Consistency properties, USAF School of Aviation Medicine, Randolph Field, Texas, 1951 Technical report 4.
[26] J. Fodor, M. Roubens, Fuzzy preference modelling and multicriteria decision support, Kluwer Acad. Publ., Dordrecht, 1994.
[27] E. Frank, M.A. Hall, I.H. Witten, The WEKA workbench, online appendix for "data mining: Practical machine learning tools and techniques", 4th, Morgan Kaufmann, 2016.
[28] J.W. Grzymala-Busse, Three approaches to missing attribute values: a rough set perspective, Studies in Computational Intelligence (SCI) 118 (2008) 139–152.
[29] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: An update, SIGKDD explorations, 2009. Volume 11, Issue 1
[30] G.H. Hardy, J. Littlewood, G. Pólya, Inequalities, Cambridge University Press, Cambridge, 1955.
[31] A. Jozwik, A learning scheme for a fuzzy k-nn rule, Pattern Recognit Lett 1 (1983) 287–289.
[32] S. Karmakar, A.K. Bhunia, A comparative study of different order relations of intervals, Reliab. Comput. 16 (2012) 38–72.
[33] J.M. Keller, M.R. Gray, J.A. Givens, A fuzzy k-nn neighbor algorithm, IEEE Trans. Syst. Man Cybern., SMC 15 (4) (1985) 580–585.
[34] M. Komorníková, R. Mesiar, Aggregation functions on bounded partially ordered sets and their classification, Fuzzy Sets Syst. 175 (2011) 48–56.
[35] U.W. Kulish, W.L. Miranker, Computer arithemtic in theory and practice, Academic Press, 1981.

[36] G. Lucca, J.A. Sanz, G.P. Dimuro, B. Bedregal, M.J. Asiain, M. Elkano, H. Bustince, CC-Integrals: Choquet-like copula-based aggregation functions and its application in fuzzy rule-based classification systems, Knowl Based Syst 119 (2017) 32–43.
[37] S. Marrara, G. Pasi, M. Viviani, Aggregation operators in information retrieval, Fuzzy Sets Syst. 324 (2017) 3–19.
[38] D. Michie, D.J. Spiegelhalter, D.J. Taylor, Machine learning, neural and statistical classification, Ellis Horwood Limited, England, 1994.
[39] R.E. Moore, Interval analysis, Prentice-Hall, Englewood Cliffs, 1966. vol. 4
[40] R. Moszyński, P. Żywica, A. Wójtowicz, S. Szubert, S. Sajdak, A. Stachowiak, K. Dyczkowski, M. Wygralak, D. Szpurek, Menopausal status strongly influences the utility of predictive models in differential diagnosis of ovarian tumors: an external validation of selected diagnostic tools, Ginekol. Pol. 85 (12) (2014) 892–899.
[41] K. Pancerz, A. Lewicki, Encoding Symbolic Features in Simple Decision Systems over Ontological Graphs for PSO and Neural Network Based Classifiers, in: Neurocomputing, volume Vol. 144, Elsevier, 2014, pp. 338–345.
[42] K. Pancerz, A. Lewicki, R. Tadeusiewicz, Ant-based extraction of rules in simple decision systems over ontological graphs, Int. J. Appl. Math. Comput. Sci. 25 (2) (2015) 377–387.
[43] K. Pancerz, Paradigmatic and Syntagmatic Relations in Information Systems over Ontological Graphs, in: Fundamenta Informaticae, Vol. 148 (1–2), IOS Press, Amsterdam, 2016, pp. 229–242.
[44] Z. Pawlak, A. Skowron, Rudiments of rough sets, Inf Sci 177 (2007) 3–27.
[45] B. Pękala, U. Bentkowska, B. De Baets, On comparability relations in the class of interval-valued fuzzy relations, Tatra Mountains Math. Publ. 66 (2016) 91–101.
[46] D.S. Scot, Outline of a Mathematical Theory of Computation, in: 4th Annual Princeton Conference on Information Sciences and Systems, 1970, pp. 169–176.
[47] A. Stachowiak, K. Dyczkowski, A. Wójtowicz, P. Żywica, M. Wygralak, A Bipolar View on Medical Diagnosis in Ovaexpert System, in: T. Andreasen, H. Christiansen, J. Kacprzyk, et al. (Eds.), Flexible Query Answering Systems 2015: Proc. of FQAS 2015, Cracow, Poland, October 26–28, 2015, Advances in Intelligent Systems and Computing, volume Vol. 400, Switzerland: Springer International Publishing, Cham, 2016, pp. 483–492.
[48] http://www.statsoft.com.
[49] J.A. Swets, Measuring the accuracy of diagnostic systems, Science 240 (1988) 1285–1293.
[50] S. Szubert, A. Wójtowicz, R. Moszyński, P. Żywica, K. Dyczkowski, A. Stachowiak, S. Sajdak, D. Szpurek, J.L. Alcázar, External validation of the IOTA ADNEX model performed by two independent gynecologic centers, Gynecol. Oncol. 142 (3) (2016) 490–495.
[51] UC Irvine Machine Learning Repository, http://archive.ics.uci.edu/ml/.
[52] A. Wójtowicz, P. Żywica, K. Szarzyński, R. Moszyński, S. Szubert, K. Dyczkowski, A. Stachowiak, D. Szpurek, M. Wygralak, Dealing with Uncertinity in Ovarian Tumor Diagnosis, Modern Approaches in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics. Vol. II: Applications, pp. 151–158, IBS PAN-SRI PAS, 2014.
[53] A. Wójtowicz, P. Żywica, A. Stachowiak, K. Dyczkowski, Solving the problem of incomplete data in medical diagnosis via interval modeling, Appl Soft Comput 47 (2016) 424–437.
[54] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning-i, Inf Sci 8 (1975) 199–249.
[55] P. Żywica, A. Wójtowicz, A. Stachowiak, K. Dyczkowski, Improving Medical Decisions under Incomplete Data Using Interval-valued Fuzzy Aggregation, in: Proc IFSA-EUSFLAT 2015, Atlantis Press, 2015, pp. 577–584.
[56] P. Żywica, K. Dyczkowski, A. Wójtowicz, A. Stachowiak, S. Szubert, R. Moszyński, Development of a fuzzy-driven system for ovarian tumor diagnosis, Biocybernetics and Biomedical Engineering 36 (4) (2016) 632–643.