Marcin PLUCIŃSKI, Marcin PIETRZYKOWSKI

West Pomeranian University of Technology

doi:10.15199/48.2017.01.19

Application of the *k* nearest neighbors method to fuzzy data processing

Abstract. *The paper presents that with the application of fuzzy numbers arithmetic, the k nearest neighbors method can be adapted to various types of data. Both, the learning data and the input data may be in the form of the crisp number, interval or fuzzy number. Experiments proved that the method works correctly and gives credible results. There is also shown that the kNN method can be used for the determination of the fuzzy model output.*

Streszczenie. *W artykule pokazano, że z wykorzystaniem arytmetyki rozmytej, metoda k najbliższych sąsiadów może być zastosowana do danych różnego typu. Zarówno dane uczące, jak i dane wejściowe modelu mogą być liczbami, interwałami lub liczbami rozmytymi. Eksperymenty wykazały, że metoda działa prwidłowo i daje wiarygodne wyniki. Zaprezentowano również możliwość użycia metody k najbliższych sąsiadów do wyznaczania wyjścia modelu rozmytego.* ***(Zastosowanie metody k najbliższych sąsiadów do przetwarzania danych rozmytych)***

Keywords: k nearest neighbors method, fuzzy numbers, fuzzy arithmetic, fuzzy model
Słowa kluczowe: metoda k najbliższych sąsiadów, liczby rozmyte, arytmetyka rozmyta, model rozmyty

Introduction

Learning of function approximators with an application of memory-based learning methods can be very often an attractive approach in comparison with creating of global models based on a parametric representation. In some situations like: small number of samples, lack of some attributes or data uncertainty, building of global models can be difficult and then memory-based methods become one of possible solutions for the approximation task.

The k-nearest neighbors (kNN) method belongs to the memory based approximation methods. It is one of the most important between them and probably one of the best described in many versions [1, 2, 6], but what is significant it is still the subject of new researches [10, 11, 15]. Other popular memory based techniques are methods based on locally weighted learning [1, 2] which use various ways of samples weighting. Methods widely applied in this category are probabilistic neural networks and generalized regression networks [14, 17].

Thanks to the fuzzy number arithmetic described further on, the kNN method can be applied to various types of data. Both, the learning data and the input data may be in the form of the crisp number, interval or fuzzy number. Results of experiments are presented in subsequent section.

The k-nearest neighbors method

The kNN method realizes a local regression. It means that the output for the considered input point $\mathbf{x}^*$ is calculated on the base of a local model created only for k samples nearest (in a meaning of an applied metric) to $\mathbf{x}^*$.

In the classic kNN method, the model output is calculated as a mean value of target values of k neighbor samples. It can be also calculated as the weighted mean value and in such case, weight values usually depend on a distance $d(\mathbf{x}^*, \mathbf{x})$ between the input point $\mathbf{x}^*$ and analyzed neighbors $\mathbf{x}$, for example:

$$w_{\mathbf{x}^*,\mathbf{x}} = \frac{1}{1 + m \cdot d(\mathbf{x}^*, \mathbf{x})/k^2}\,, \tag{1}$$

where: the m parameter is determined empirically.

The main parameter of the kNN method is the number of neighbors k that are used in calculations. It can be constant for entire data set, but it can be also dynamically varied – according to the input point location in the input space. The most popular techniques of k evaluation are applying cross-validation or applying two distinct data sets: training data – that are memorized by the model, and testing data – to evaluate the real model error. The best k value is the value that gives the lowest test or crossvalidation error and in this way it guarantees the lowest real error of the model and the best generalization.

Fuzzy numbers

The main concepts connected with fuzzy numbers (FN) are well described in many literature positions, e.g. in [4, 8, 13]. Let's recall some basic definitions.

The fuzzy subset of the real numbers set $\mathbb{R}$, with the membership function $\mu : \mathbb{R} \to [0, 1]$, is a fuzzy number if:

a) A is normal, i.e. there exists an element $x_0 \in \mathbb{R}$ such that $\mu(x_0) = 1$;
b) A is convex, i.e. $\mu(\lambda x + (1 - \lambda)y) \geq \mu(x) \wedge \mu(y)$, $\forall\, x, y \in \mathbb{R}$ and $\forall\, 0 \leq \lambda \leq 1$;
c) μ is upper semicontinuous;
d) supp(μ) is bounded.

Each fuzzy number can be described as:

$$\mu(x) = \begin{cases} 0 & \text{for} \quad x < a_1 \\ f(x) & \text{for} \quad a_1 \leq x < a_2 \\ 1 & \text{for} \quad a_2 \leq x < a_3 \\ g(x) & \text{for} \quad a_3 \leq x < a_4 \\ 0 & \text{for} \quad x \geq a_4 \end{cases} \tag{2}$$

where: $a_1, a_2, a_3, a_4 \in \mathbb{R}$. f is a nondecreasing function and is called the left side of the fuzzy number. g is a nonincreasing function and is called the right side of the fuzzy number.

The next important concept are α-levels of the fuzzy set. The α-level set A_α of the fuzzy number A is a nonfuzzy set defined by:

$$A_\alpha = \{x \in \mathbb{R} :\ \mu(x) \geq \alpha\}\,. \tag{3}$$

The family $\{A_\alpha :\ \alpha \in (0, 1]\}$ can be a representation of the fuzzy number.

From the definition of the fuzzy number results that α-level set is compact for each $\alpha > 0$. As a consequence, each A_α can be represented by an interval:

$$A_\alpha = [f^{-1}(\alpha), g^{-1}(\alpha)]\,, \tag{4}$$

where: $f^{-1} = \inf\{x :\ \mu(x) \geq \alpha\}$ and $g^{-1} = \sup\{x :\ \mu(x) \geq \alpha\}$.

Distance between fuzzy numbers

Fuzzy numbers do not form a natural linear order, like e.g. real numbers, so different approaches are necessary for

calculating the distance between them. Many methods have been described in the literature [3, 8, 16]. Each one has its own advantages and disadvantages, so it is hard to decide which one is the best. In this paper, methods proposed in [8] will be applied.

The distance, indexed by parameters $p \in [1, \infty)$, $q \in [0, 1]$, between fuzzy numbers A and B can be calculated as:

(5)
$$\delta_{p,q}(A,B) = \begin{cases} \sqrt[p]{(1-q)\int_0^1 |f_B^{-1}(\alpha) - f_A^{-1}(\alpha)|^p \,\mathrm{d}\alpha \\ \qquad +q\int_0^1 |g_B^{-1}(\alpha) - g_A^{-1}(\alpha)|^p \,\mathrm{d}\alpha} \\ \qquad \text{for } 1 \le p < \infty \\ (1-q)\sup\limits_{0<\alpha\le 1}(|f_B^{-1}(\alpha) - f_A^{-1}(\alpha)|) \\ \qquad +q\sup\limits_{0<\alpha\le 1}(|g_B^{-1}(\alpha) - g_A^{-1}(\alpha)|) \\ \qquad \text{for } p = \infty \end{cases}$$

or the distance indexed by the parameter $p \in [1, \infty)$, can be calculated as:

(6)
$$\rho_p(A,B) = \begin{cases} \max\left\{\sqrt[p]{\int_0^1 |f_B^{-1}(\alpha) - f_A^{-1}(\alpha)|^p \,\mathrm{d}\alpha}, \right. \\ \qquad \left. \sqrt[p]{\int_0^1 |g_B^{-1}(\alpha) - g_A^{-1}(\alpha)|^p \,\mathrm{d}\alpha}\right\} \\ \qquad \text{for } 1 \le p < \infty \\ \max\left\{\sup\limits_{0<\alpha\le 1}(|f_B^{-1}(\alpha) - f_A^{-1}(\alpha)|), \right. \\ \qquad \left. \sup\limits_{0<\alpha\le 1}(|g_B^{-1}(\alpha) - g_A^{-1}(\alpha)|)\right\} \\ \qquad \text{for } p = \infty \end{cases}$$

where: $A_\alpha = [f_A^{-1}(\alpha), g_A^{-1}(\alpha)]$ and $B_\alpha = [f_B^{-1}(\alpha), g_B^{-1}(\alpha)]$. The second parameter q of $\delta_{p,q}$ characterizes the weights connected with sides of fuzzy numbers. If there is no reason to distinguish any side, $q = 0.5$ is recommended.

If we assume that all fuzzy numbers are elements of the space $F(\mathbb{R})$ then it can be proved that $(F(\mathbb{R}), \delta_{p,q})$ and $(F(\mathbb{R}), \rho_p)$ are metric spaces [8].

Fuzzy numbers arithmetic

As it was said before, if A_α is the α-level set of the fuzzy number A, then it can be represented in the form:

(7)
$$A = \bigcup_{\alpha\in[0,1]} \alpha, A_\alpha \,.$$

Each α-level set is an interval, so rules of interval arithmetic [12] can be applied in formulation of basic arithmetic operations of fuzzy numbers. If we have two interval numbers $[a_1, a_2]$ and $[b_1, b_2]$ then:

(8)
$$[a_1, a_2] \oplus [b_1, b_2] = [a_1 \oplus b_1, a_2 \oplus b_2] \,,$$

(9)
$$\begin{aligned} &[a_1, a_2] \otimes [b_1, b_2] \\ &= [\min(a_1 \otimes b_1, a_1 \otimes b_2, a_2 \otimes b_1, a_2 \otimes b_2), \\ &\quad \max(a_1 \otimes b_1, a_1 \otimes b_2, a_2 \otimes b_1, a_2 \otimes b_2)] \,, \end{aligned}$$

where: $\oplus \in \{+, -\}$, $\otimes \in \{\times, \div\}$ and $0 \notin [b_1, b_2]$ if $\otimes = \div$.

Above interval operations can be extended to fuzzy numbers [4, 5, 7, 9]. Let:

$$A = \bigcup_{\alpha\in[0,1]} \alpha, [a_1^\alpha, a_2^\alpha] \quad \text{and} \quad B = \bigcup_{\alpha\in[0,1]} \alpha, [b_1^\alpha, b_2^\alpha] \,,$$

be two fuzzy numbers, then:

(10)
$$A \circ B = \bigcup_{\alpha\in[0,1]} \alpha, ([a_1^\alpha, a_2^\alpha] \circ [b_1^\alpha, b_2^\alpha]) \,,$$

where: $\circ = \{+, -, \times, \div\}$.

The kNN method for crisp, interval and fuzzy numbers

Thanks to the fuzzy numbers arithmetic, the kNN method can be adapted to various types of data. Both, the learning data and the input data may be in the form of the crisp number, interval or fuzzy number. However, to be able to calculate the distance between the input vector and the samples, and to calculate the output in the way described in previous section, it will be necessary to unify all data. Each attribute in the training data set and in the input vector must be represented by the fuzzy number. Crisp numbers can be replaced by fuzzy numbers with a singleton membership function and interval numbers – by fuzzy numbers with a rectangular membership function, Fig. 1.

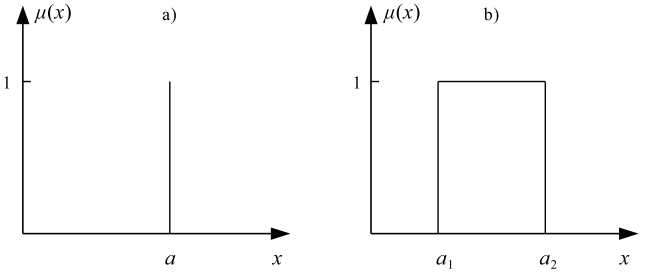

Fig. 1. Singleton fuzzy number (a) and rectangular fuzzy number (b) representing the crisp number and interval number respectively

Results of experiments

All experiments were realized on the basis of a specially prepared Python library that allows calculations on interval and fuzzy numbers.

Experiment 1 – SISO model

The objective of first experiments was to determine whether the method works correctly and its results are credible. So, the research was carried out on training data with only one input attribute and one output attribute. Data were prepared to be diverse as much as possible, thus both attributes took values in a form of crisp numbers, interval numbers and fuzzy numbers. Data are presented in Table 1.

In experiment, a distance measure $\delta_{p,q}$ described by formula (5) was applied with parameters $p = 2$ and $q = 0.5$. The measure ρ_p described by formula (6) turned out to be less effective. The $\max$ operator, occurring in formula, caused a less sensitivity to changes of the input vector $\mathbf{x}^*$ value.

Calculations were carried out for input vectors in the form of crisp numbers, interval numbers and fuzzy numbers.

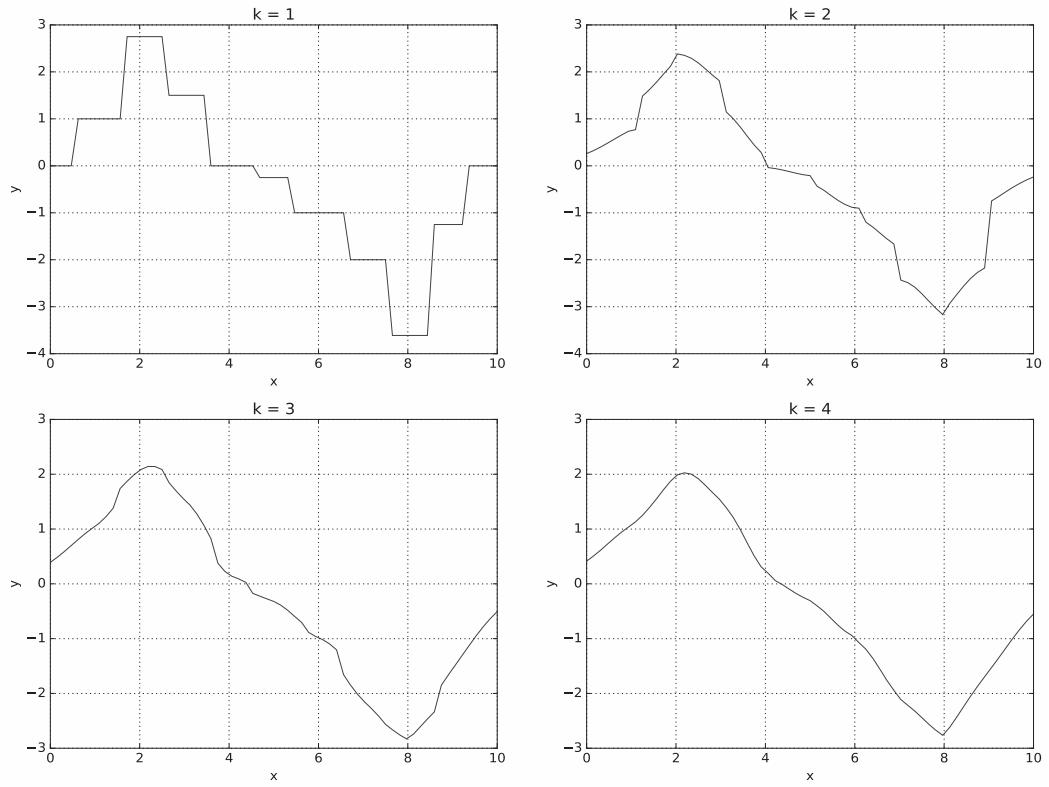

Fig. 2. Characteristics of weighted kNN model created for data from Table 1 for $k = 1 \ldots 4$

Table 1. Learning data used in experiment 1

input x	output y
0	0
1	1
$[2, 2.5]$	$[2.5, 3]$
3	triangle FN $[1, 1.5, 2]$
4	0
triangle FN $[4.5, 5, 5.5]$	$[-0.5, 0]$
6	-1
$[7, 7.5]$	-2
8	trapezoid FN $[-4, -4, -3.5, -3]$
trapezoid FN $[8, 8.5, 9, 9.5]$	$[-1.5, -1]$
10	0

Some exemplary results are presented in Table 2. All calculations were performed for $k = 4$. Results obtained for the kNN method and its weighted version are similar, but the weighted kNN method is more sensitive to the input vector $\mathbf{x}^*$ value.

Table 2. Results of calculations for learning data from Table 1

input vector $\mathbf{x}^*$	result for kNN method
3.5	trapezoid FN $[0.75, 0.875, 1.125, 1.25]$
$[6, 10]$	trapezoid FN $[-2.13, -2.13, -1.88, -1.75]$
triangle FN $[3, 3.5, 4]$	trapezoid FN $[0.75, 0.875, 1.125, 1.25]$

input vector $\mathbf{x}^*$	result for weighted kNN method
3.5	trapezoid FN $[0.64, 0.85, 0.93, 1.14]$
$[6, 10]$	trapezoid FN $[-2.23, -2.23, -1.91, -1.78]$
triangle FN $[3, 3.5, 4]$	trapezoid FN $[0.66, 0.86, 0.96, 1.16]$

Fig. 2 presents characteristics of weighted kNN model created for data from Table 1 for $k = 1 \ldots 4$. As the model output is usually the fuzzy number, the abscissa of the center of gravity of results is presented in the plot.

Experiment 2 – fuzzy model

The next experiment will show that the kNN method can be used for the determination of the fuzzy model output.

Let's consider 2-input fuzzy model. Membership functions of all concept used in rules are presented in Fig. 3 and rules of the model can be found in Table 3.

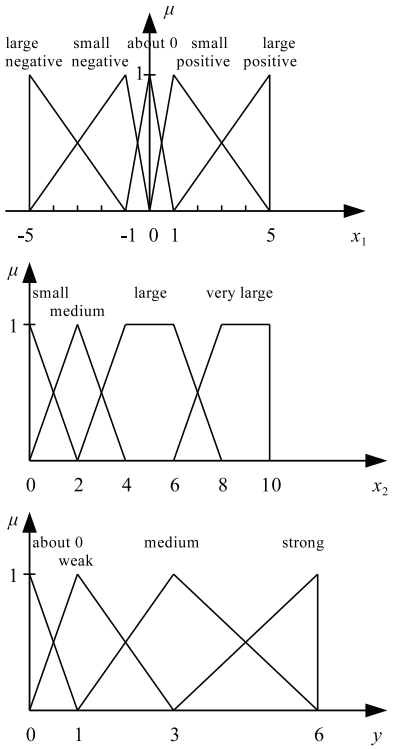

Fig. 3. Membership functions of all concept used in rules

Table 3 can be treated as a set of training data values in the form of fuzzy numbers and in this way it can be used in the kNN method. Let's check results of its work.

In the beginning, let's calculate the model output for the input vector $\mathbf{x}^* = (-2, 2.5)$. The kNN method gives as the result (for $k = 4$) the triangle FN $[0.25, 1.25, 3.75]$ and

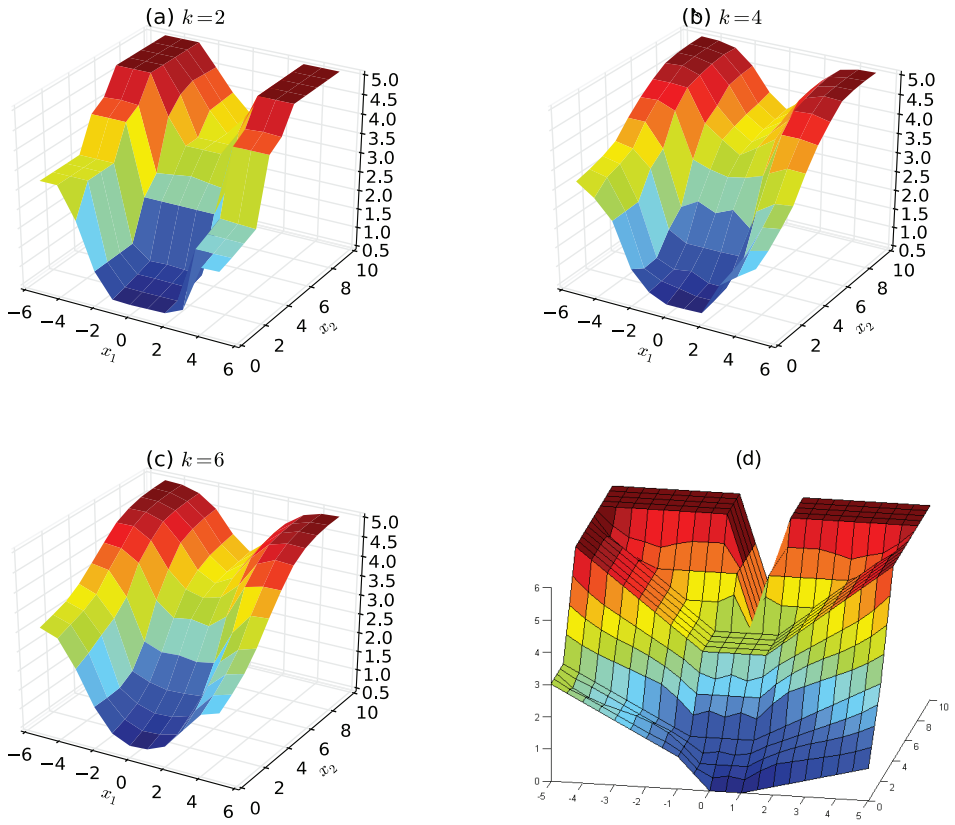

Fig. 4. Characteristics of the weighted kNN model created for fuzzy data for various values of k parameter (a) - (c). Subplot (d) presents the characteristic of the fuzzy model created with use of the min-max inference and the singleton defuzzification

Table 3. Rules of the fuzzy model

x_1	x_2	y
large negative	small	medium
large negative	medium	medium
large negative	large	strong
large negative	very large	strong
small negative	small	weak
small negative	medium	weak
small negative	large	medium
small negative	very large	strong
about 0	small	about 0
about 0	medium	weak
about 0	large	medium
about 0	very large	medium
small positive	small	about 0
small positive	medium	weak
small positive	large	medium
small positive	very large	strong
large positive	small	weak
large positive	medium	medium
large positive	large	strong
large positive	very large	strong

the weighted kNN – the triangle FN $[0.135, 1.27, 3.404]$. For comparison, calculations of the fuzzy model output were also performed in a classic way (with use of the $\min$-$\max$ inference and the singleton defuzzification). They gave a crisp result equal to 2.4.

With the kNN method, it is also easy to perform calculations for uncertain input data. For example, let's assume that $\mathbf{x}^*$ = ('bigger than 2' , 'about 3'). Such input vector can be described as $\mathbf{x}^*$ = ($[2, 5]$, triangle FN $[2.5, 3, 3.5]$). As the result we get: triangle FN $[1, 2.75, 4.5]$ for the kNN method and triangle FN $[1.08, 2.94, 4.9]$ for the weighted kNN method.

Fig. 4 presents characteristics of the weighted kNN model created for fuzzy data for various values of k parameter. As before, the abscissa of the center of gravity of results is presented in the plot. For comparison, there is also shown the characteristic of the fuzzy model determined in a classic way.

The main advantages of the model based on the kNN method are as follows.

- The possibility of choosing any number of samples involved in calculations (in the classical 2-input fuzzy system, usually 4 rules take part in calculations).
- The possibility of easy calculations for the input of any form (crisp, interval or fuzzy number).
- Insensitivity for the incompleteness of the rule base – this is particularly important in the case of multi-input data (as illustrated in the next experiment).

Experiment 3 – weather data

In the last experiment, let's apply the popular benchmark – 'weather' data, Table 4. Let's assume that attribute values are described by fuzzy numbers which membership functions are presented in Fig. 5

Fuzzy system with a complete rule base should have 36 rules. There is 14 samples in the data set and each one of them can be interpreted as one decision rule – so the rule base is incomplete. As before, the output of the model can be determined for various types of input data.

For example:

- for $\mathbf{x}^* = (0.9, 70, 76, 0.5)$
  - output of the kNN model: triangle FN [0, 0.25, 1],
  - output of the weighted kNN model: triangle FN [0, 0.13, 1],
- for $\mathbf{x}^*$ = ('below 0.2', 'over 80', 'between 70 and 80',

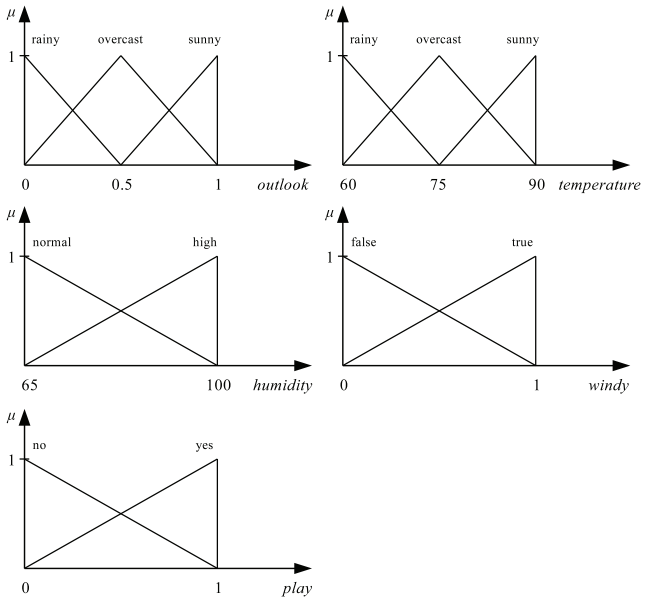

Fig. 5. Membership functions of 'weather' data concepts

Table 4. 'Weather' data

outlook	temperature	humidity	windy	play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

'over 0.7') or more formally $\mathbf{x}^*$ = ($[0, 0.2]$, $[80, 90]$, $[70, 80]$, $[0.7, 1]$)

- output of the kNN model: triangle FN [0, 0.5, 1],
- output of the weighted kNN model: triangle FN [0, 0.28, 1],

- for $\mathbf{x}^*$ = ('about 0.2', 'about 80', 'about 80', 'about 0.8') or more formally $\mathbf{x}^*$ = (triangle FN [0.1, 0.2, 0.3], triangle FN [70, 80, 90], triangle FN [70, 80, 90], triangle FN [0.6, 0.8, 1])
  - output of the kNN model: triangle FN [0, 0.75, 1],
  - output of the weighted kNN model: triangle FN [0, 0.34, 1].

Conclusions

With the application of fuzzy numbers arithmetic, the kNN method can be used for diverse data. Both, the learning data and the input data can be crisp or imprecise (interval or fuzzy number). Experiments described in the paper proved that the method gives correct and credible results.

The kNN method can be also applied for the determination of the fuzzy model result. Such approach has some very important advantages. First of all, the rule base can be incomplete and it does not have to be consistent. Moreover, calculations can be realized for crisp or uncertain input data. We can also assume in advance, how many rules will be used in determining the model output. All these advantages cause that the kNN method can be an attractive alternative to the classic way of data processing by the fuzzy model.

REFERENCES

[1] Atkeson C.G., Moore A.W., Schaal S.A.: Locally weighted learning. Artificial Intelligence Review, 11, pp. 11–73, 1997.
[2] Cichosz P.: Learning systems. WNT Publishing House, Warsaw, 2000. [in Polish]
[3] Diamond P., Rosenfeld A.: Metric spaces of fuzzy sets. Fuzzy Sets and Systems, 35, pp. 241–249, 1990.
[4] Dubois D., Prade H.: Operations on fuzzy numbers. International Journal of Systems Science, 9(6), pp. 613–626, 1978.
[5] Dutta P., Boruah H., Ali T.: Fuzzy Arithmetic with and without using α-cut method: A Comparative Study. International Journal of Latest Trends in Computing, 2(1), pp. 99–107, 2011.
[6] Hand D., Mannila H., Smyth P.: Principles of data mining. The MIT Press, 2001.
[7] Hanss M.: Applied fuzzy arithmetic. Springer Verlag, Berlin, Heidelberg, 2005
[8] Grzegorzewski P: Metrics and orders in space of fuzzy numbers. Fuzzy Sets and Systems, 97, pp. 83–94, 1998.
[9] Kaufmann A., Gupta M.M.: Introduction to fuzzy arithmetic. Van Nostrand Reinhold, New York, 1991.
[10] Kordos M., Blachnik M., Strzempa D.: Do We Need Whatever More Than k-NN? In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010. LNCS, vol. 6113, pp. 414-421. Springer, Heidelberg, 2010.
[11] Korzeń M., Klęsk P.: Sets of approximating functions with finite Vapnik-Czervonenkis dimension for nearest-neighbours algorithm. Pattern Recognition Letters, 32, pp. 1882–1893, 2011.
[12] Moore R.E., Kearfott R.B., Cloud M.J.: Introduction to interval analysis. Society for Industrial and Applied Mathematics, Philadelphia, 2009.
[13] Piegat A.: Fuzzy modeling and control. Physica Verlag, Heidelberg-New York, 2001.
[14] Pluciński M.: Application of data with missing attributes in the probability RBF neural network learning and classification. Artificial Intelligence and Security in Computing Systems: Proceedings of the 9th International Conference ACS'2002, Eds.: J. Sołdek, L. Drobiazgiewicz, Boston/Dordrecht/London: Kluwer Academic Publishers, pp. 63–72, 2003.
[15] Pluciński M.: Application of the information-gap theory for evaluation of nearest neighbours method robustness to data uncertainty. Przegląd Elektrotechniczny, 88(10b), pp. 272–275, 2012.
[16] Tang W., Li X., Zhao R.: Metric spaces of fuzzy variables. Computers & Industrial Engineering, 57, pp. 1268–1273, 2009.
[17] Wasserman P.D.: Advanced methods in neural computing. New York, Van Nostrand Reinhold, 1993.

***Authors**: Ph.D. Marcin Pluciński, M. Sc. Marcin Pietrzykowski, Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, Żołnierska 49, 71-210 Szczecin, Poland, email: mplucinski@wi.zut.edu.pl, mpietrzykowski@wi.zut.edu.pl*