

A heuristic algorithm for generating the numerical terms of a linguistic variable

Elena N. Chujkova

*Associate Professor, Department of Computer Systems and Information Security
Don State Technical University
Address: 1, Gagarin Square, Rostov-on-Don, 344000, Russia
E-mail: elenchu@mail.ru*

Vasilij V. Galushka

*Associate Professor, Department of Computer Systems and Information Security
Don State Technical University
Address: 1, Gagarin Square, Rostov-on-Don, 344000, Russia
E-mail: galushkavv@yandex.ru*

Abstract

In this paper we describe an easy-to-implement algorithm for automated generation of the linguistic variable term membership functions to allow for information search in a relational database based on qualitative criteria by means of the SQL query language.

The proposed algorithm makes it possible to calculate the parameters of the triangular and trapezoid membership functions taking into account the distribution of the variable of interest stored in the database. The algorithm defines the intervals covered by the term bases, so that each interval contains about the same number of values. Upper bounds of the defined intervals are used to calculate the parameters of membership functions. The parameters of the membership functions generated with this algorithm can be easily calculated with the limited computational means of the SQL language.

We review the algorithm realizations for the generation of 3 and 5 terms of a linguistic variable based on a sample from a database containing 100 or 500 different values.

The membership functions obtained through the algorithm have the required properties of orderliness, completeness, consistency and normality. They do not require further approximation. Unlike the known methods, the algorithm does not require significant computing resources, the use of specialized software, settings configuring, or a training set formation.

The algorithm implementation creates opportunities to support fuzzy search queries in relational databases using the means of the SQL language, as limited as they are. Thus, the system's level of intelligence would be increased, and the user would be provided with the means of search query formulation in a natural language. The linguistic variable terms generated using our algorithm can be used within the framework of a fuzzy rule-based knowledge base of an information system, as well as to perform fuzzy inference.

Key words: relational database; SQL language; fuzzy logic; linguistic variable; fuzzy set; membership function.

Citation: Chujkova E.N., Galushka V.V. (2018) A heuristic algorithm for generating the numerical terms of a linguistic variable. *Business Informatics*, no. 3 (45), pp. 29–38.
DOI: 10.17323/1998-0663.2018.3.29.38.

Introduction

At the core of any information system lies a database that stores the information processed by the system. One of the functions of the system is searching the database for any requested data. Currently, relational databases that support the search for information using SQL language are the most widely used. Composing a search request in SQL requires the user to set definite ranges for the values of the data being requested, something which is often impossible due the lack of such information. Moreover, the user, especially a novice, is better accustomed to using qualitative search criteria that could be expressed verbally. This kind of user-system interaction is also more convenient for the user.

In an implementation of information search in a database based on qualitative criteria, the tools of fuzzy logic can be used. They involve the use of linguistic variables and the generation of corresponding base term sets (fuzzy sets) that provide a quantitative interpretation for the qualitative search criteria [1–5].

Performing a fuzzy search query in a relational database requires converting it first into a command of the standard SQL language. Paper [6] describes such a conversion procedure that is implemented with the help of the graphical interface of the information system. Within this procedure, the values of the membership functions of the fuzzy sets used are calculated on-the-fly in the course of performing an SQL query. Linguistic variable terms are represented by parametric fuzzy numbers with trapezoid and triangular membership functions. Their parameters can be easily calculated using the commands of the SQL language, which is not intended for complex calculations.

This paper [6] uses the assumption about a uniform distribution of the data sampled from the database to find the parameters of the terms. This assumption limits the applicability of the proposed method and might lead to unaccep-

table results when it is not valid. In particular, if the distribution of the experimental data is nonuniform, it might happen that no values at all from the database would be in the base of some terms. To avoid such issues, a calculation of the terms' parameters must take the experimental data distribution into account. The term bases must contain approximately equal numbers of values available in the database.

Traditionally, the membership functions of the linguistic variable terms are generated based on expert information, which makes the procedure difficult to automate. For this reason, many researchers have made considerable effort towards the development of algorithmic methods of membership function generation. The papers [7; 8] provide an overview of the existing methods of automatic generation of membership functions. Among them are the methods of inductive logical inference [9], fuzzy c-means clustering methods [10–13], neural networks [14; 15], histograms [16], methods based on fuzzy entropy and other special measures [17; 18], genetic algorithms [19–21], ant colony system algorithms [22], heuristic algorithms [23], and particle swarm optimization [24]. The characteristics of the aforementioned methods are presented in *Table 1* [8].

All the methods listed above have been developed for particular applications of classification and are used to assign experimental data to fuzzy sets. However, they are not guaranteed to produce uniform distribution with an approximately equal number of elements in each set.

Paper [25] proposed a method of partitioning a source sample of values into intervals that contain equal numbers of values. However, using this method one might see identical values being assigned to different intervals, which defeats the purpose of classification.

The membership functions generated using the above methods do not always meet the necessary conditions of being ordered, complete, consistent, or normalized [2; 4], and still

Table 1.

**Characteristics of the methods
of automatic generation of membership functions**

Methods	Characteristics			
	Source of information	Area of applicability	Shape of the membership functions	Number of fuzzy sets
Neural networks	dataset	classification	arbitrary	fixed
Histograms	dataset	image recognition, classification	Gaussian	arbitrary
Fuzzy <i>c</i> -means clustering	dataset	classification	triangular, trapezoid	fixed
Genetic algorithms	dataset and expert estimates	controllers	triangular, trapezoid, Gaussian	fixed
Ant colony system algorithms	dataset	data analysis, controllers	arbitrary	arbitrary
Particle swarm methods	dataset and expert estimates	controllers, image processing	Gaussian, triangular S-shaped	fixed
Other methods	dataset	classification, controllers	triangular, Gaussian	2–9

require further approximation. For example, the Gaussian membership functions generated by the particle swarm optimization are subnormal [8].

The existing algorithmic methods of generating membership functions typically have a high computational complexity. Using neural networks makes it necessary to provide a training set. The genetic algorithms, ant colony system algorithms, and particle swarm optimization methods require an objective function to be specified. They are also notorious for their large time to convergence and the possibility of their convergence to a local optimum. The implementation of most methods listed above requires some specialized software.

In this paper, we propose a simple-to-implement method for automatic generation of the membership functions for linguistic variable terms. The method takes into account the distribution of values of the parameter at hand stored in a database, which makes it possible to assign an approximately equal number of val-

ues to each term base. In addition, this method can be used to generate an arbitrary number of membership functions that possess the required properties.

1. Problem formulation

Tables of a relational database store numerical values of objects' characteristics used by an information system that must be evaluated and selected based on some qualitative fuzzy criteria (for example, "low," "medium," "high"). To perform a fuzzy search query, one has to provide linguistic variables corresponding to the properties being evaluated. To enable the automatic generation of a linguistic variable, we need an algorithm that would allow us to calculate the number and parameters of the membership functions of the linguistic variable terms. The choice of the number of values (terms) of the linguistic variable is up to the user. The basic scale of a linguistic variable is specified based on a set of all values of the property under investigation obtained

from the database and is given by the interval $U=[u_{\min}, u_{\max}]$, where u_{\min} is the minimum value of that property and u_{\max} is the maximum value of that property. Each term base must contain an approximately equal number of values of the property being evaluated that are available in the database. The set of values and their frequencies can be obtained from the database using an SQL query as a sorted (ascending) matrix $H_{k \times 2}$, where the number of rows k is the number of different measurements (values of the parameter at hand in the database), the matrix elements h_{i1} are the values of the i -th measurement, and h_{i2} are the frequencies of the i -th measurement, $i = 1, \dots, k$. The membership functions we are seeking to generate should be defined in such a way that their values could be easily calculated by means of the SQL language, which falls short of the computational efficiency of a universal programming language. The membership functions must meet certain requirements [2; 4].

We will represent the terms of a linguistic variable by parametric fuzzy numbers with the most common trapezoid and triangular membership functions. We will use the trapezoid membership functions for the minimum and maximum terms, and the triangular membership functions for the rest of the terms. The above choice of the membership function types was influenced by our goal to ensure the possibility of calculation on-the-fly while running an SQL query, considering the limited capabilities of SQL language. The calculation of the values of the membership functions is performed according to the following formula:

$$\mu(x) = \begin{cases} 0, (x < a) \vee (x > d) \\ 1, b \leq x \leq c \\ \frac{x-a}{b-a}, (a \leq x < b) \wedge (a < b), \\ \frac{d-x}{d-c}, (c < x \leq d) \wedge (c < d) \end{cases} \quad (1)$$

where $a \leq b \leq c \leq d$ are the parameters of the membership function $\mu(x)$.

Let us split the base of the term into two intervals. The total number of such intervals is $m = l + 1$, where l is the number of terms. Figure 1 shows the shape of the membership functions of linguistic variable terms. Their parameters and the upper bounds of the values of the intervals are provided. The following notation is used:

- T_i – the i -th term;
- a_i, b_i, c_i, d_i – the parameters of the i -th term, $i = 1, \dots, l$;
- p_j – the upper bound of the values of the j -th interval, $j = 1, \dots, m$.

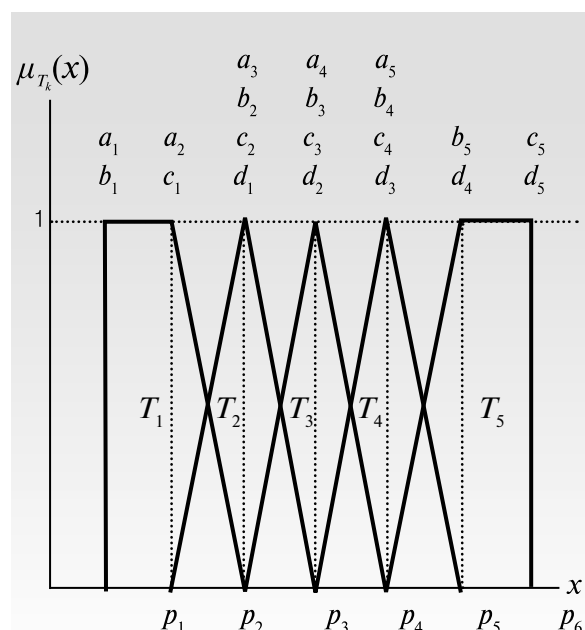


Fig. 1. Membership functions of linguistic variable terms

We need to define the intervals covered by the term bases so that each interval would contain an approximately equal number of values of the property at hand, and moreover, so that identical values would lie within the same interval. The upper bounds p_j of the intervals being defined are used to calculate the parameters a_j, b_j, c_j, d_j of the terms membership functions and are given by the quantiles of the uniform discretization of the source sample.

Based on the aforementioned matrix H , the number n of values that must lie within each interval, assuming a uniform distribution, is given by the equation

$$n = \sum_{i=1}^k h_{i2} / m,$$

where $\sum_{i=1}^k h_{i2}$ is the total number of measurements (the sample size).

Let us define the vector \vec{z} , its components being the numbers of measurements within each interval. Then the j -th component of the vector \vec{z} is the number z_j of measurements within the j -th interval ($j = 1, \dots, m$):

$$z_j = \sum_{i \in G} h_{i2},$$

where G is the set of numbers i of the rows of matrix H that contain the values of the frequencies h_{i2} included in z_j .

Let us denote by i^* the largest element of G : $i^* = \max_{i \in G} i$. Then, to calculate the quantiles p_j ($j = 1, \dots, l$), we only need to choose p_j to be such elements $h_{i^*}(i^* = 1, \dots, k-1)$ of matrix H that the function $E(\vec{z}) = \max_{j=1, \dots, l+1} |z_j - n|$ is minimized. Alternatively, another equivalent criterion of optimality could be used:

$$E(\vec{z}) = \sum_{j=1}^{l+1} |z_j - n| \rightarrow \min.$$

The value p_{l+1} coincides with the largest value in the sample: $p_{l+1} = h_{k1}$.

The optimization is performed under the following conditions:

$$\sum_{j=1}^{l+1} z_j = \sum_{i=1}^k h_{i2};$$

for $\forall j$ ($j = 1, \dots, l+1$) $z_j = \sum_{i \in G} h_{i2}$, $i_{g+1} = i_g + 1$

($g = i_{\min}, \dots, i_{\max} - 1$), where $i_{\min} = \min_{i \in G} i$, $i_{\max} = \max_{i \in G} i$;

for $\forall j_1, j_2, j_1 \neq j_2$, for which $z_{j_1} = \sum_{i \in G_1} h_{i2}$, $z_{j_2} = \sum_{i \in G_2} h_{i2}$, $G_1 \cap G_2 = \emptyset$ are defined.

To obtain non-trivial and nondegenerate solutions, the number l of the terms must obey the following condition: $2 \leq l \leq k$. The number of measurements contained in each of m

intervals cannot be smaller than the maximum frequency of the measurements, therefore

$m \leq \sum_{i=1}^k h_{i2} / \max_{i=1, \dots, k} h_{i2}$. This means that the number

l of the terms must meet the following condition:

$$2 \leq l \leq \sum_{i=1}^k h_{i2} / \max_{i=1, \dots, k} h_{i2} - 1 \quad (2)$$

The parameters a_j, b_j, c_j, d_j of the terms' membership functions are found as follows:

$$a_j = \begin{cases} h_{11}, j=1 \\ p_{j-1}, 1 < j \leq l \end{cases}; \quad b_j = \begin{cases} h_{11}, j=1 \\ p_j, 1 < j \leq l \end{cases};$$

$$c_j = \begin{cases} p_j, 1 \leq j < l \\ p_{j+1}, j=l \end{cases}; \quad d_j = p_{j+1}, 1 \leq j \leq l.$$

2. Algorithm for generation of membership functions

Let us find the actual number z_j of the measurements that get assigned to the j -th interval ($j = 1, \dots, m$) and the parameters a_j, b_j, c_j, d_j of the linguistic variable terms according to the following algorithm:

Step 0. Verify that the conditions (2) of the existence of non-trivial and nondegenerate solutions are satisfied. If the conditions (2) are satisfied, then go to Step 1.

Step 1. Initialize i ($i := 0$).

1.1. For each j ($j = 1, \dots, m-1$), find $z1_j$ as follows

1.2.1. Initialize $z1_j$ ($z1_j := 0$).

1.2.2. While $z1_j < n$, consequentially accumulate the values h_{i2} in $z1_j$:

$$\begin{aligned} i &:= i + 1 \\ z1_j &:= z1_j + h_{i2}. \end{aligned}$$

1.2.3. If $z1_j \geq n$, then find R and Q :

$$\begin{aligned} R &:= n - (z1_j - h_{i2}) \\ Q &:= z1_j - n. \end{aligned}$$

Here R is the difference between the required number n of measurements within the interval and the actual number $z1_j$ of measurements

within the j -th interval, not counting the i -th measurement; Q is the difference between the actual number $z1_j$ of measurements within the j -th interval (counting the i -th measurement) and the required number n of measurements. The value of R shows how much the number of measurements within the j -th interval is smaller than the required number (the shortage), while the value of Q shows how much the number of measurements within the j -th interval is greater than the required number n (the surplus).

1.2.4. If the shortage without the account for the last measurement is smaller than the surplus ($R < Q$), then remove from $z1_j$ the last measurement added to it h_{i2} :

$$\begin{aligned} z1_j &:= z1_j - h_{i2} \\ i &:= i - 1. \end{aligned}$$

1.2.5. Define the upper bound $p1_j$ of the j -th interval as follows: $p1_j := h_{i1}$.

1.3. Find the number $z1_m$ of measurements that are assigned to the last m -th interval:

1.3.1. Initialize $z1_m$ ($z1_m := 0$).

1.3.2. Add to $z1_m$ all the remaining measurement: for all i ($i = i + 1, \dots, k$)

$$z1_m := z1_m + h_{i2}.$$

1.3.3. Define the upper bound $p1_m$ of the m -th interval as follows: $p1_m := h_{k1}$.

Step 2. Find the total deflection δ_1 of the number of measurements from the required n :

$$\delta_1 = \sum_{j=1}^m |z1_j - n|.$$

Step 3. Repeat the calculations z_j by processing the rows of matrix H in reverse order beginning with the last row:

3.1. Initialize i ($i := k$).

3.2. Define the upper bound $p2_m$ of the m -th interval as follows: $p2_m := h_{k1}$.

3.3. For each j ($j = m, \dots, 2$), find $z2_j$ as follows:

3.3.1. Initialize $z2_j$ ($z2_j := 0$).

3.3.2. While $z2_j < n$, consequentially accumulate the values h_{i2} in $z2_j$:

$$i := i - 1.$$

$$z2_j := z2_j + h_{i2}.$$

3.3.3. If $z2_j \geq n$, then find R and Q :

$$R := n - (z2_j - h_{i2})$$

$$Q := z2_j - n.$$

3.3.4. If $R < Q$, then remove from $z2_j$ the value h_{i2} :

$$z2_j := z2_j - h_{i2}$$

$$i := i + 1.$$

3.3.5. Define the upper bound $p2_{j-1}$ of the previous ($j - 1$)-th interval as follows:

$$p2_{j-1} := h_{(i-1)1} - h_{i2}.$$

3.4. Find the number $z2_l$ of measurements assigned to the first interval:

3.4.1. Initialize $z2_l$ ($z2_l := 0$).

3.4.2. Add to $z2_l$ all the remaining measurement: for all i ($i = i - 1, \dots, l$)

$$z2_l := z2_l + h_{i2}.$$

Step 4. Find the total deflection δ_2 of the number of measurements from the required n :

$$\delta_2 = \sum_{j=1}^m |z2_j - n|.$$

Step 5. Find the values of the parameters a_j, b_j, c_j, d_j of the linguistic variable terms using the elements of vector p corresponding to the distribution z of measurements for which the least total deflection δ is achieved (if $\delta_1 < \delta_2$, then use vector $p1$ corresponding to the distribution $z1$ of measurements, otherwise use $p2$ corresponding to $z2$) as follows:

$$a_1 := h_{11}$$

$$b_1 := h_{11}.$$

Let $\delta_1 < \delta_2$.

Then for each j ($j = 1, \dots, m - 1$) do:

if $j = m - 1$, then $c_j := p1_j, a_{j+1} := p1_j$;

if $j > 1$, then $d_{j-1} := p1_j, b_j := p1_j$;

$$c_{m-1} := p1_m;$$

$$d_{m-1} := p1_m.$$

3. Analysis of the results produced by the algorithm

We consider the realizations of the above algorithm used to generate 3 and 5 terms of a linguistic variable based on samples from a database that contained 100 or 500 different values.

For a sample of 100 values and 3 terms, the number of intervals $m = 4$; the number of values per interval $n = 100/4 = 25$; the number of distinct measurements (the number of rows of matrix H of the values and their frequencies)

$k = 20$. The source data (the elements of matrix H) along with the results of Steps 1-4 of the algorithm are presented in *Table 2*.

Table 2 demonstrates that the total deflection of the number of measurements from the required n for the downward processing of matrix H is greater than that for the reverse processing ($\delta_1 > \delta_2$), therefore the parameters of the membership functions are defined using vector $p2$. The values of the parameters of membership functions calculated at Step 5 of the algorithm are presented in *Table 3*.

Table 2.

Source data and the results of steps 1-4 of the algorithm

Row number of matrix $H(i)$	Elements of matrix H		Interval number (j)	Number of measurements in the j -th interval ($z1_j$)	Upper bound of the j -th interval ($p1_j$)	Variance (δ_1)	Number of measurements in the j -th interval ($z2_j$)	Upper bound of the j -th interval ($p2_j$)	Variance (δ_2)
	Value of the i -th measurement (h_{i1})	Frequency of the i -th measurement (h_{i2})							
1	354	5	1	28	718	12	21	718	10
2	616	5							
3	662	5							
4	695	4							
5	699	2							
6	718	7							
7	741	2	2	26	784				
8	745	5							
9	758	2							
10	764	2							
11	770	9							
12	784	6							
13	785	2	3	27	800				
14	790	14							
15	791	2							
16	800	9							
17	810	11	4	19	855				
18	813	5							
19	814	2							
20	855	1							

Table 3.

Parameters of membership functions calculated by the algorithm

Membership function number (j)	Parameters of the j -th membership function			
	a_j	b_j	c_j	d_j
1	354	354	718	770
2	718	770	770	791
3	770	791	855	855

Plots of the membership functions for 3 and 5 terms obtained through the algorithm are shown in *Figures 2 and 3*, respectively. The experimental data sampled from the database are presented by histograms in the figures.

The shape of the membership functions calculated without the account for the distribution of experimental data for the same data samples are shown in *Figure 4*.

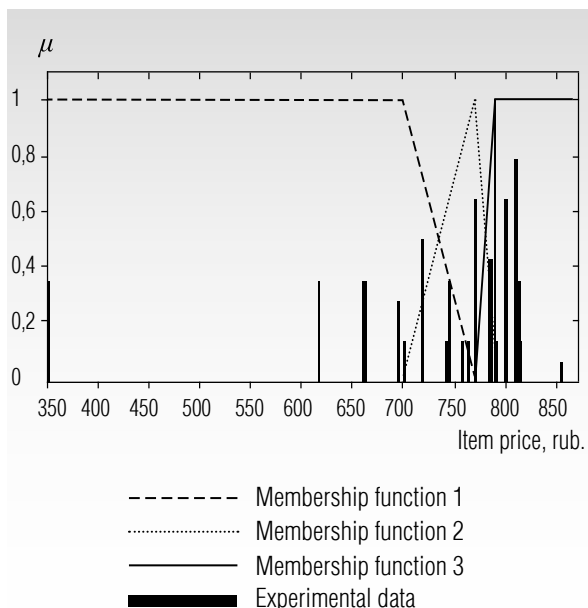


Fig. 2. Membership functions for 3 terms of the linguistic variable "Item price" for a sample of 100 price values

As seen from *Figures 2 and 3*, the proposed algorithm generates the membership functions for the linguistic variable terms that meet all the essential requirements [2; 4]. In particular, it is ensured that each term base contains an approximately equal number of values. By contrast, the membership functions generated without the account for the experimental data distribution (see *Figure 4*) do not possess the required properties and require further approximation.

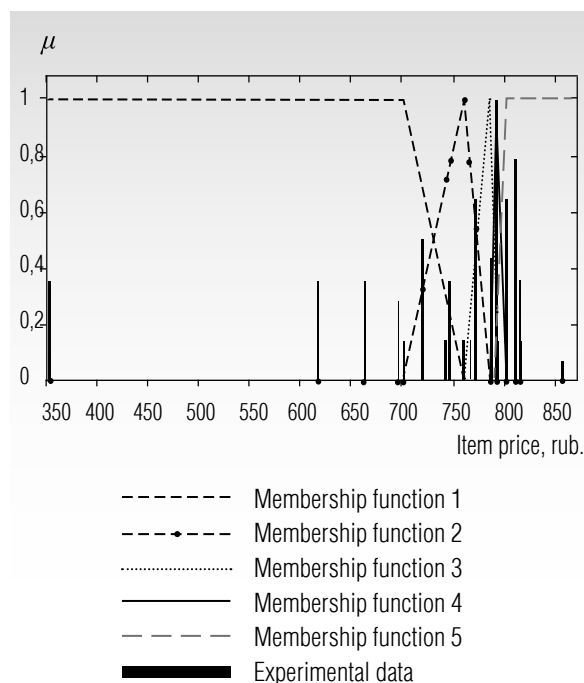


Fig. 3. Membership functions for 5 terms of the linguistic variable "Item price" for a sample of 100 price values

Conclusion

The proposed algorithm makes it possible to automate the procedure of generating the membership functions for the linguistic variable terms. The parameters of such membership functions can be stored in a database and later used to submit fuzzy queries in SQL language. In contrast to the existing methods of automatic generation of the membership functions, our algorithm is simple to implement,

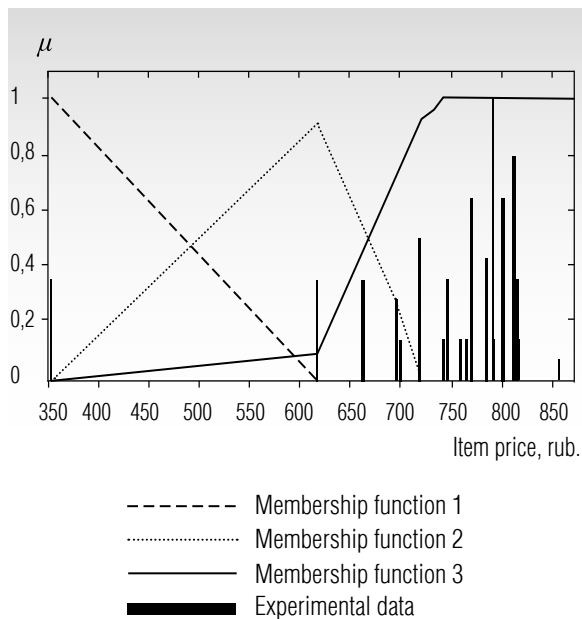


Fig. 4. Membership functions for 3 terms of the linguistic variable "Item price" generated without the account for the experimental data distribution for a sample of 100 price values

does not require considerable computational resources or specialized software, is free of adjustable parameters, does not require a training set, and enables the generation of any user-

specified number of triangular and trapezoid membership functions. The membership functions generated with the help of this algorithm take into account the experimental data distribution. Additionally, the respective term bases of the linguistic variable contain approximately equal numbers of values selected from the database. The membership functions obtained through this algorithm possess all the required properties [2; 4] and do not necessitate further approximation. At the same time, the algorithm was developed for the solution of the particular problem stated above, and cannot be claimed to be universal.

The algorithm implementation creates opportunities to support fuzzy search queries in relational databases using the means of SQL language, as limited as they are. Thus, the system's level of intelligence would be increased, and the user would be provided with the means of search query formulation in a natural language. The membership functions of the linguistic variable terms generated using our algorithm can be used within the framework of a fuzzy rule-based knowledge base of an information system, as well as to perform fuzzy inference. ■

References

1. Shtovba S. (2001) *Vvedenie v teoriyu nechetkih mnozhestv i nechetkuyu logiku* [Introduction to the theory of fuzzy sets and fuzzy logic]. Available at: <http://matlab.exponenta.ru/fuzzylogic/book1> (accessed 14 October 2013) (in Russian).
2. Korneev V.V., Gareev A.F., Vasjutin S.V., Rajh V.V. (2000) *Bazy dannyh. Intellektual'naya obrabotka informatsii* [Databases. Intelligent processing of information]. Moscow: Knowledge (in Russian).
3. Zadeh L. (1965) Fuzzy sets. *Information and Control*, vol. 8, no. 3, pp. 338–353.
4. Zadeh L. (1975) The concept of a linguistic variable and its application to approximate reasoning – II. *Information Sciences*, vol. 8, no. 4, pp. 301–357.
5. Andrejchikov A.V. (2006) *Intellektual'nye informatsionnye sistemy* [Intelligent information systems]. Moscow: Finance and Statistics (in Russian).
6. Chujkova E.N. (2014) Realizatsiya nechetkogo vybora oborudovaniya v sisteme proektirovaniya informatsionnoy seti [Realization of the fuzzy selection of equipment in the information network design system]. *Vestnik of Don State Technical University*, no. 3, pp. 164–171 (in Russian).
7. Medasani S., Kim J., Krishnapuram R. (1998) An overview of membership function generation techniques for pattern recognition. *International Journal of Approximate Reasoning*, no. 19, pp. 391–417.
8. Schwaab A., Nassar S., Filho P. (2015) Automatic methods for generation of type-1 and interval type-2 fuzzy membership functions. *Journal of Computer Sciences*, no. 11, pp. 976–987.

9. Kim C., Russell B. (1993) Automatic generation of membership function and fuzzy rule using inductive reasoning. Proceedings of the *3rd International Conference on Industrial Fuzzy Control and Intelligent Systems (IFIS '93)*, Houston, Texas, 1–3 December 1993, pp. 93–96.
10. Chen M., Wang S. (1999) Fuzzy clustering analysis for optimizing membership functions. *Fuzzy Sets and Systems*, no. 103, pp. 239–254.
11. Liao T., Celmins A., Hammell R. (2001) A fuzzy c-means variant for the generation of fuzzy term sets. *Fuzzy Sets and Systems*, vol. 135, no. 2, pp. 241–257.
12. Lopes P., Camargo H. (2012) Automatic labeling by means of semi-supervised fuzzy clustering as a boosting mechanism in the generation of fuzzy rules. Proceedings of the *13th IEEE International Conference on Information Reuse and Integration (IRI)*. Las Vegas, USA, 8–10 August 2012, pp. 279–286.
13. Jamsandekar S., Mudholkar R. (2014) Fuzzy classification system by self generated membership function using clustering technique. *BVICAM's International Journal of Information Technology*, vol. 6, no. 1,
14. Wu S., Er M., Gao Y. (2001) A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks. *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, pp. 578–594.
15. Castellano G., Castiello C., Fanelli A., Mencar C. (2005) Knowledge discovery by a neuro-fuzzy modeling framework. *Fuzzy Sets and Systems*, no. 149, pp. 187–207.
16. Refaey M. (2016) Automatic generation of membership functions and rules in a fuzzy logic system. Proceedings of the *5th International Conference on Informatics and Applications (ICIA 2016)*. Takamatsu, Japan, 14–16 November 2016, pp. 117–122.
17. Cheng H., Chen J. (1997) Automatically determine the membership function based on the maximum entropy principle. *Information Sciences*, vol. 96, no. 3–4, pp. 163–182.
18. Nieradka G., Butkiewicz B. (2007) A method for automatic membership function estimation based on fuzzy measures. Proceedings of the *International Fuzzy Systems Association World Congress (IFSA)*. Cancun, Mexico, 18–21 June 2007, pp. 451–460.
19. Homaifar A., McCormick E. (1995) Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms. *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 2, pp. 129–139.
20. Shimojima K., Fukuda T., Hasegawa Y. (1995) Self-tuning fuzzy modeling with adaptive membership function, rules, and hierarchical structure based on genetic algorithm. *Fuzzy Sets and Systems*, vol. 71, no. 3, pp. 295–309.
21. Kaya M., Alhaji R. (2004) Integrating multi-objective genetic algorithms into clustering for fuzzy association rules mining. Proceedings of the *4th IEEE International Conference on Data Mining (ICDM '04)*. Brighton, UK, 1–4 November 2004, pp. 431–434.
22. Hong T., Tung Y., Wang S., Wu M., Wu Y. (2009) An ACS-based framework for fuzzy data mining. *Expert Systems with Applications*, no. 36, pp. 11844–11852.
23. Ishibuchi H., Nozaki K., Tanaka H. (1993) Efficient fuzzy partition of pattern space for classification problems. *Fuzzy Sets and Systems*, vol. 59, no. 3, pp. 295–304.
24. Permana K., Zaiton S. (2010) Fuzzy membership function generation using particle swarm optimization. *International Journal of Open Problems in Computer Science and Mathematics*, vol. 3, no. 1, pp. 27–41.
25. Piatetsky-Shapiro G., Connell C. (1984) Accurate estimation of the number of tuples satisfying a condition. *ACM SIGMOD Record*, vol. 19, no. 2, pp. 256–276.