

# Эвристический алгоритм формирования термов числовой лингвистической переменной

## Е.Н. Чуйкова

кандидат технических наук  
доцент кафедры вычислительных систем и информационной безопасности  
Донской государственной технической университет  
Адрес: 344000, г. Ростов-на-Дону, пл. Гагарина, д. 1  
E-mail: elenchu@mail.ru

## В.В. Галушка

кандидат технических наук  
доцент кафедры вычислительных систем и информационной безопасности  
Донской государственной технической университет  
Адрес: 344000, г. Ростов-на-Дону, пл. Гагарина, д. 1  
E-mail: galushkavv@yandex.ru

### Аннотация

Описывается простой в реализации метод автоматизированного формирования функций принадлежности термов лингвистической переменной для обеспечения возможности поиска информации в реляционной базе данных на основе качественных критериев средствами языка запросов SQL.

Предлагаемый алгоритм позволяет вычислить параметры треугольных и трапециевидальных функций принадлежности термов с учетом распределения содержащихся в базе данных значений рассматриваемой характеристики. Алгоритм определяет охватываемые основаниями термов интервалы таким образом, чтобы в каждый интервал попадало приблизительно одинаковое количество значений. Верхние границы определяемых интервалов используются для вычисления параметров функций принадлежности. Определяемые по алгоритму функции принадлежности допускают возможность оперативного расчета их значений ограниченными вычислительными средствами языка SQL.

Рассмотрены реализации алгоритма для формирования трех и пяти термов лингвистической переменной, основанные на выборках из базы данных, содержащих 100 и 500 различных значений.

Получаемые в результате работы алгоритма функции принадлежности обладают требуемыми свойствами упорядоченности, полноты, согласованности, нормальности и не нуждаются в дополнительной аппроксимации. В отличие от известных методов, алгоритм не требует значительных вычислительных ресурсов, привлечения специализированного программного обеспечения, настройки параметров или формирования обучающей выборки.

Реализация алгоритма создает условия для поддержки нечеткого поиска в базе данных средствами языка SQL, повышая тем самым уровень интеллектуальности системы и обеспечивая пользователю средства общения на естественном языке. Определенные в результате работы алгоритма термы лингвистических переменных могут быть использованы в нечетких правилах базы знаний информационной системы и при выполнении нечеткого вывода.

**Ключевые слова:** реляционная база данных; язык SQL; нечеткая логика; лингвистическая переменная; нечеткое множество; функция принадлежности.

**Цитирование:** Чуйкова Е.Н., Галушка В.В. Эвристический алгоритм формирования термов числовой лингвистической переменной // Бизнес-информатика. 2018. № 3 (45). С. 29–38.  
DOI: 10.17323/1998-0663.2018.3.29.38.

## Введение

Ядром любой информационной системы является база данных, хранящая обрабатываемую системой информацию, и одной из функций системы является поиск в базе данных требуемой информации. Сегодня наиболее широко распространены реляционные базы данных, поддерживающие поиск информации средствами языка SQL. Формулировка поискового запроса на SQL требует задания конкретных диапазонов значений запрашиваемых данных, что зачастую невозможно из-за отсутствия подобной информации. Кроме того, пользователю, особенно неопытному, более привычно оперировать качественными критериями поиска, определяемыми вербально. При этом также повышается комфортность взаимодействия пользователя с системой.

Для реализации информационного поиска в базе данных на основе качественных характеристик применим аппарат нечеткой логики, который предполагает использование лингвистических переменных и решение задачи формирования соответствующих им наборов базовых термов (нечетких множеств), задающих количественную интерпретацию применяемых качественных характеристик [1–5].

Выполнение нечеткого поискового запроса в реляционной базе данных предполагает его преобразование в инструкцию стандартного языка SQL. В работе [6] описана такая процедура преобразования, реализуемая средствами графического интерфейса информационной системы. При этом значения функций принадлежности используемых нечетких множеств рассчитываются «на лету» в процессе выполнения SQL-запроса. Термы лингвистических переменных представлены параметрическими нечеткими числами с трапецеидальными и треугольными функциями принадлежности, расчет значений которых легко выполним с помощью инструкций языка SQL, не предназначенного для сложных вычислений.

В работе [6] определение параметров термов основано на предположении о равномерном распределении данных, выбираемых из базы данных, что ограничивает область применения предложенной методики и может привести к неприемлемым результатам в тех случаях, когда данное предположение неверно. В частности, при неравномерном распределении экспериментальных данных может возникнуть ситуация, когда в основание каких-ли-

бо термов вообще не попадет ни одного значения из базы данных. Для преодоления указанных недостатков параметры термов следует вычислять с учетом распределения экспериментальных данных, причем таким образом, чтобы основания термов включали приблизительно равное количество имеющихся в базе данных значений.

Функции принадлежности термов лингвистической переменной традиционно формируют на основе экспертной информации, что препятствует автоматизации процедуры их построения, поэтому многие исследователи направили свои усилия на разработку алгоритмических методов их формирования. В работах [7; 8] приведен обзор известных на сегодняшний день методов автоматизированного построения функций принадлежности. Среди них можно выделить методы индуктивного логического вывода [9], нечеткой кластеризации  $s$ -средних [10–13], нейронных сетей [14; 15], гистограмм [16], методы на основе нечеткой энтропии и других специальных мер [17; 18], генетические [19–21], муравьиные [22], эвристические алгоритмы [23], методы роя частиц [24]. Характеристики указанных методов представлены в таблице 1 [8].

Все перечисленные методы разработаны для определенной области применения и предназначены для классификации, распределяя экспериментальные данные по нечетким множествам. При этом они не гарантируют равномерное распределение с приблизительно одинаковым числом элементов в каждом множестве.

В работе [25] описан способ разбиения исходной выборки значений на интервалы, содержащие одинаковое число значений, но при таком разбиении одинаковые значения могут попасть в разные интервалы, что противоречит цели классификации.

Функции принадлежности, вычисляемые с помощью указанных выше методов, не всегда обладают требуемыми свойствами упорядоченности, полноты, согласованности, нормальности [2; 4] и нуждаются в дополнительной аппроксимации. Например, получаемые по методу роя частиц Гауссовы функции принадлежности являются субнормальными [8].

Для существующих алгоритмических методов построения функций принадлежности характерна достаточно высокая вычислительная сложность. При использовании нейронных сетей необходимо формирование обучающей выборки. Генетические, муравьиные алгоритмы, методы роя частиц требуют

Таблица 1.

**Характеристики методов автоматизированного  
построения функций принадлежности**

Методы	Характеристики			
	Источник информации	Область применения	Форма функции принадлежности	Число нечетких множеств
Нейронные сети	набор данных	классификация	произвольная	фиксированное
Гистограммы	набор данных	распознавание образов, классификация	Гауссова	произвольное
Нечеткая кластеризация <i>c</i> -средних	набор данных	классификация	треугольная, трапецидальная	фиксированное
Генетические алгоритмы	набор данных и экспертные оценки	контроллеры	треугольная, трапецидальная, Гауссова	фиксированное
Муравьиные алгоритмы	набор данных	анализ данных, контроллеры	произвольная	произвольное
Методы роя частиц	набор данных и экспертные оценки	контроллеры, обработка изображений	Гауссова, треугольная, S-подобная	фиксированное
Другие методы	набор данных	классификация, контроллеры	треугольная, Гауссова	2–9

задания целевой функции для поиска решения, а также характеризуются большим временем сходимости и наличием опасности попадания в локальный оптимум. Реализация многих из указанных методов основывается на использовании специализированного программного обеспечения.

В данной статье предложен простой в реализации метод автоматизированного формирования функций принадлежности термов лингвистической переменной, учитывающий распределение хранимых в базе данных значений рассматриваемого параметра, обеспечивающий включение в основание термов приблизительно одинакового количества значений и позволяющий получить произвольное количество функций принадлежности, обладающих требуемыми свойствами.

### 1. Постановка задачи

В таблицах реляционной базы данных хранятся численные значения используемых информационной системой характеристик объектов, оценивать и выбирать которые необходимо на основании качественных нечетких критериев (например, «низкий», «средний», «высокий»). Для выполнения нечеткого поиска следует задать лингвистические переменные, соответствующие оцениваемым характеристикам. Для автоматизированного формирования лингвистической переменной необходим алгоритм, позволяющий определить число и параметры функций принадлежности термов лингви-

стической переменной. Число значений (термов) лингвистической переменной может выбрать пользователь системы. Базовая шкала для лингвистической переменной задается на основе выборки всех значений рассматриваемой характеристики из базы данных и определяется интервалом  $U = [u_{\min}, u_{\max}]$ , где  $u_{\min}$  – минимальное значение рассматриваемой характеристики,  $u_{\max}$  – максимальное значение указанной характеристики. Основания термов должны включать приблизительно одинаковое количество содержащихся в базе данных значений оцениваемой характеристики. Набор значений и их частотность могут быть получены из базы данных с помощью SQL-запроса в виде отсортированной в порядке возрастания матрицы  $H_{k \times 2}$ , где число строк  $k$  – количество различных измерений (значений интересующего параметра в таблице базы данных), элементы матрицы  $h_{i1}$  – значения  $i$ -го измерения,  $h_{i2}$  – частоты  $i$ -го измерения,  $i = 1, \dots, k$ . Определяемые функции принадлежности следует задать таким образом, чтобы обеспечивалась возможность оперативного расчета их значений средствами языка SQL, не обладающего вычислительной мощностью универсального языка программирования. Функции принадлежности должны удовлетворять предъявляемым к ним требованиям [2; 4].

Термы лингвистической переменной представим параметрическими нечеткими числами с наиболее распространенными трапецидальными и треугольными функциями принадлежности. При этом минимальный и максимальный термы – тра-

пецеидальные, остальные – треугольные. Выбор указанных видов функций принадлежности также объясняется стремлением обеспечить возможность определения их значений ограниченными средствами языка SQL в процессе выполнения SQL-запроса. Вычисление значений функций принадлежности осуществляется по формуле:

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

где  $a \leq b \leq c \leq d$  – параметры функции принадлежности  $\mu(x)$ .

Основание каждого терма разделим на два интервала. Число таких интервалов  $m = l + 1$ , где  $l$  – число термов. На рисунке 1 представлен вид функций принадлежности термов лингвистической переменной с указанием их параметров и верхних границ значений упомянутых интервалов. При этом используются следующие обозначения:

$T_i$  –  $i$ -й терм;

$a_i, b_i, c_i, d_i$  – параметры  $i$ -го терма,  $i = 1, \dots, l$ ;

$p_j$  – верхняя граница значений  $j$ -го интервала,  $j = 1, \dots, m$ .

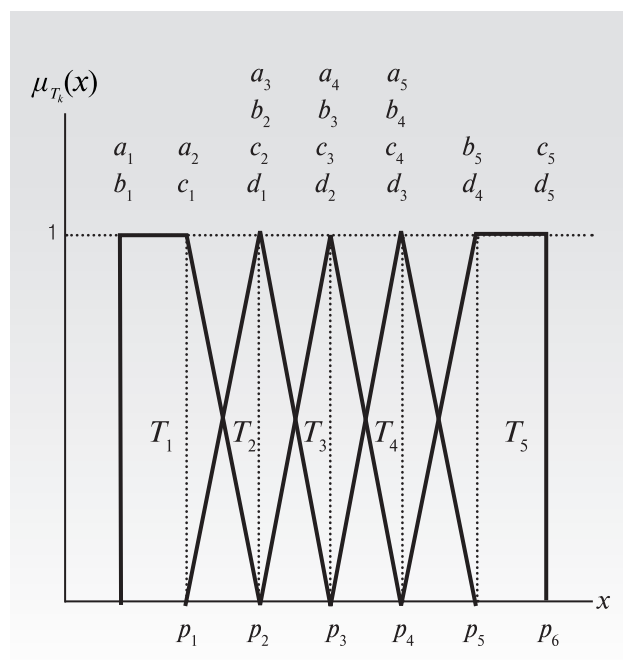


Рис. 1. Функции принадлежности термов лингвистической переменной

Необходимо определить охватываемые основаниями термов интервалы таким образом, чтобы в каждый интервал попадало приблизительно одинаковое число значений рассматриваемой характеристики, причем одинаковые значения должны оказаться в одном и том же интервале. Верхние границы  $p_j$  определяемых интервалов используются для вычисления параметров  $a_j, b_j, c_j, d_j$  функций принадлежности термов и представляют собой квантили равномерной дискретизации исходной выборки.

На основании упомянутой выше матрицы  $H$  число  $n$  значений, которые должны содержаться в каждом интервале при равномерном распределении, определяется соотношением:

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

где  $\sum_{i=1}^k h_{i2}$  – общее количество измерений (размер выборки).

Определим вектор  $\vec{z}$ , компонентами которого являются значения числа измерений, включаемых в каждый интервал. Тогда  $j$ -я компонента вектора  $\vec{z}$  – число  $z_j$  измерений, включаемых в  $j$ -й интервал ( $j = 1, \dots, m$ ):

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

где  $G$  – множество номеров  $i$  строк матрицы  $H$ , содержащих значения частот  $h_{i2}$ , включаемые в  $z_j$ .

Обозначим через  $i^*$  максимальный элемент множества  $G: i^* = \max_{i \in G} i$ . Тогда вычисление квантилей  $p_j$  ( $j = 1, \dots, l$ ) сводится к выбору в качестве  $p_j$  таких значений  $h_{i^*}$  ( $i^* = 1, \dots, k-1$ ) из матрицы  $H$ , при которых достигается минимум функции  $E(\vec{z}) = \max_{j=1, \dots, l+1} |z_j - n|$ . Может также использоваться другой равнозначный критерий оптимальности:  $E(\vec{z}) = \sum_{j=1}^{l+1} |z_j - n| \rightarrow \min$ .

Значение  $p_{l+1}$  совпадает с максимальным значением в выборке:  $p_{l+1} = h_{k1}$ .

Оптимизация выполняется при следующих ограничениях:

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

для  $\forall j$  ( $j = 1, \dots, l+1$ )  $z_j = \sum_{i \in G} h_{i2}, i_{g+1} = i_g + 1$  ( $g = i_{\min}, \dots, i_{\max} - 1$ ), где  $i_{\min} = \min_{i \in G} i, i_{\max} = \max_{i \in G} i$ ; для  $\forall j_1, j_2, j_1 \neq j_2$ , для которых определены  $z_{j_1} = \sum_{i \in G_1} h_{i2}, z_{j_2} = \sum_{i \in G_2} h_{i2}, G_1 \cap G_2 = \emptyset$ .

Для получения нетривиальных и невырожденных решений число  $l$  определяемых термов должно

удовлетворять условию:  $2 \leq l \leq k$ . Количество содержащихся в каждом из  $m$  интервалов измерений не может быть меньше максимальной частоты измерений, поэтому  $m \leq \sum_{i=1}^k h_{i2} / \max_{i=1, \dots, k} h_{i2}$ . Следовательно, на число  $l$  термов накладывается ограничение:

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

Параметры  $a_i, b_i, c_i, d_i$  функций принадлежности термов вычисляются следующим образом:

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

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

## 2. Алгоритм формирования функций принадлежности

Рассчитаем фактическое число  $z_j$  измерений, попадающих в  $j$ -й интервал ( $j = 1, \dots, m$ ) и параметры  $a_j, b_j, c_j, d_j$  термов лингвистической переменной по следующему алгоритму:

**Шаг 0.** Проверить выполнение условий (2) получения нетривиальных и невырожденных решений. Если условия (2) удовлетворяются, то перейти к шагу 1.

**Шаг 1.** Инициализировать  $i$  ( $i := 0$ ).

**1.1.** Для каждого  $j$  ( $j = 1, \dots, m - 1$ ), вычислить  $z1_j$  следующим образом:

**1.2.1.** Инициализировать  $z1_j$  ( $z1_j := 0$ ).

**1.2.2.** Пока  $z1_j < n$  накапливать последовательно в  $z1_j$  значения  $h_{i2}$ :

$$i := i + 1 \\ z1_j := z1_j + h_{i2}.$$

**1.2.3.** Если  $z1_j \geq n$ , то вычислить значения  $R$  и  $Q$ :

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

Здесь  $R$  – разность между требуемым числом  $n$  измерений в интервале и фактическим числом  $z1_j$  измерений в  $j$ -м интервале без учета  $i$ -го измерения;  $Q$  – разность между фактическим числом  $z1_j$  измерений в  $j$ -м интервале (с учетом  $i$ -го измерения) и требуемым числом  $n$  измерений. Значение  $R$  показывает, насколько число измерений в  $j$ -м интервале меньше требуемого значения (недостачу), величина  $Q$  определяет превышение числа измерений в  $j$ -м интервале над требуемым  $n$ .

**1.2.4.** Если недостача без учета последнего измерения меньше превышения ( $R < Q$ ), то исключить из  $z1_j$  последнее добавленное измерение  $h_{i2}$ :

$$z1_j := z1_j - h_{i2} \\ i := i - 1.$$

**1.2.5.** Определить верхнюю границу  $p1_j$   $j$ -го интервала:  $p1_j := h_{i1}$ .

**1.3.** Вычислить число  $z1_m$  измерений, попадающих в последний  $m$ -й интервал:

**1.3.1.** Инициализировать  $z1_m$  ( $z1_m := 0$ ).

**1.3.2.** Добавить в  $z1_m$  все оставшиеся измерения: для всех  $i$  ( $i = i + 1, \dots, k$ )

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

**1.3.3.** Определить верхнюю границу  $p1_m$   $m$ -го интервала:  $p1_m := h_{k1}$ .

**Шаг 2.** Рассчитать суммарное отклонение  $\delta_1$  числа измерений от требуемого  $n$ :

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

**Шаг 3.** Выполнить повторный расчет  $z_j$ , просматривая матрицу  $H$  в обратном порядке, начиная с последней строки:

**3.1.** Инициализировать  $i$  ( $i := k$ ).

**3.2.** Определить верхнюю границу  $p2_m$   $m$ -го интервала:  $p2_m := h_{k1}$ .

**3.3.** Для каждого  $j$  ( $j = m, \dots, 2$ ), вычислить  $z2_j$  следующим образом:

**3.3.1.** Инициализировать  $z2_j$  ( $z2_j := 0$ ).

**3.3.2.** Пока  $z2_j < n$  накапливать последовательно в  $z2_j$  значения  $h_{i2}$ :

$$i := i - 1 \\ z2_j := z2_j + h_{i2}.$$

**3.3.3.** Если  $z2_j \geq n$ , то вычислить значения  $R$  и  $Q$ :

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

**3.3.4.** Если  $R < Q$ , то исключить из  $z2_j$  значение  $h_{i2}$ :

$$z2_j := z2_j - h_{i2} \\ i := i + 1.$$

**3.3.5.** Определить верхнюю границу  $p2_{j-1}$  предыдущего ( $j - 1$ )-го интервала:

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

**3.4.** Вычислить число  $z2_1$  измерений, попадающих в первый интервал:

**3.4.1.** Инициализировать  $z_2$  ( $z_{2_i} := 0$ ).

**3.4.2.** Добавить в  $z_2$  все оставшиеся измерения: для всех  $i$  ( $i = i - 1, \dots, l$ )

$$z_{2_i} := z_{2_i} + h_{i2}.$$

**Шаг 4.** Рассчитать суммарное отклонение  $\delta_2$  числа измерений от требуемого  $n$ :

$$\delta_2 = \sum_{j=1}^m |z_{2_j} - n|.$$

**Шаг 5.** Вычислить значения параметров  $a_j, b_j, c_j, d_j$  термов лингвистической переменной на основе значений вектора  $p$ , соответствующего распределению  $z$  измерений, для которого достигается меньшее суммарное отклонение  $\delta$  (если  $\delta_1 < \delta_2$ , то использовать вектор  $p1$ , соответствующий распределению  $z1$  измерений, в противном случае –  $p2$ , соответствующий  $z2$ ), следующим образом:

$$a_1 := h_{11}$$

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

Пусть  $\delta_1 < \delta_2$ .

Тогда для каждого  $j$  ( $j = 1, \dots, m - 1$ ) выполнить:

если  $j = 1 < m - 1$ , то  $c_j := p1_j, a_{j+1} := p1_j$ ;

если  $j > 1$ , то  $d_{j-1} := p1_j, b_j := p1_j$ ;

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

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

### 3. Анализ результатов работы алгоритма

Рассмотрены реализации алгоритма для формирования трех и пяти термов лингвистической переменной, основанные на выборках из базы данных, содержащих 100 и 500 различных значений.

Для выборки из 100 значений и трех термов имеем: число интервалов  $m = 4$ ; число значений, попадающих в один интервал  $n = 100/4 = 25$ ; количество различных измерений (число строк матрицы  $H$ , содержащей набор значений и их частотность)  $k = 20$ . Исходные данные (содержимое матрицы  $H$ ) и результаты выполнения этапов 1–4 алгоритма представлены в *таблице 2*.

Таблица 2.

Исходные данные и результаты выполнения этапов 1–4 алгоритма

Номер строки матрицы $H(i)$	Элементы матрицы $H$		Номер интервала ( $j$ )	Число измерений в $j$ -м интервале ( $z1_j$ )	Верхняя граница $j$ -го интервала ( $p1_j$ )	Отклонение ( $\delta_1$ )	Число измерений в $j$ -м интервале ( $z2_j$ )	Верхняя граница $j$ -го интервала ( $p2_j$ )	Отклонение ( $\delta_2$ )
	Значение $i$ -го измерения ( $h_{i1}$ )	Частота $i$ -го измерения ( $h_{i2}$ )							
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	2	26	784		27	770	
7	741	2							
8	745	5							
9	758	2							
10	764	2							
11	770	9	3	27	800	24	791		
12	784	6							
13	785	2							
14	790	14							
15	791	2							
16	800	9	4	19	855	28	855		
17	810	11							
18	813	5							
19	814	2							
20	855	1							

Из *таблицы 2* видно, что суммарное отклонение числа измерений от требуемого  $n$  при прямом просмотре матрицы  $H$  больше, чем при обратном ( $\delta_1 > \delta_2$ ), поэтому параметры функций принадлежности определяются на основе вектора  $p_2$ . Вычисленные значения параметров функций принадлежности (шаг 5 алгоритма) представлены в *таблице 3*.

Таблица 3.

**Вычисленные по алгоритму параметры функций принадлежности**

Номер функции принадлежности ( $j$ )	Параметры $j$ -й функции принадлежности			
	$a_j$	$b_j$	$c_j$	$d_j$
1	354	354	718	770
2	718	770	770	791
3	770	791	855	855

Графики полученных в результате выполнения алгоритма функций принадлежности для трех и пяти термов показаны на *рисунках 2 и 3* соответственно. Экспериментальные данные, выбранные из базы данных, представлены на рисунках гистограммой.

Вид функций принадлежности, построенных без учета распределения экспериментальных данных для той же выборки значений, изображен на *рисунке 4*.

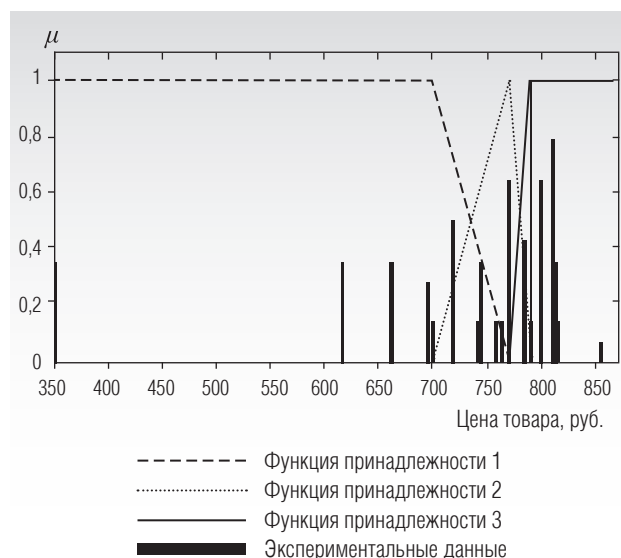


Рис. 2. Функции принадлежности трех термов лингвистической переменной «Цена товара» для выборки из 100 значений цены

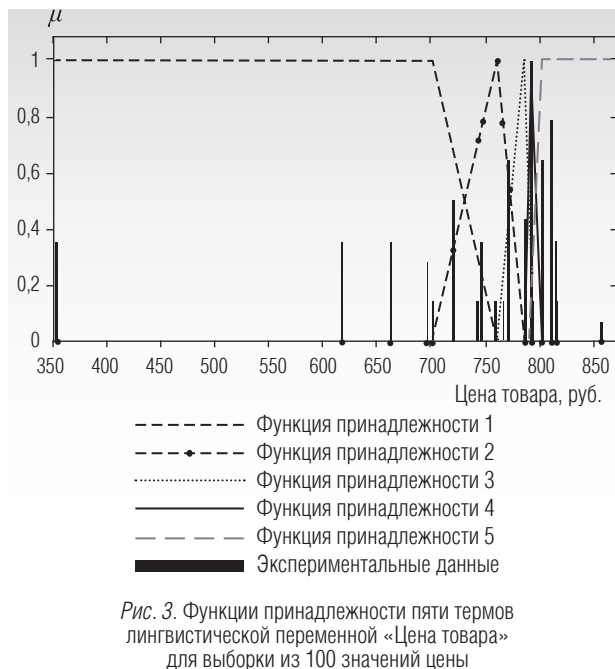


Рис. 3. Функции принадлежности пяти термов лингвистической переменной «Цена товара» для выборки из 100 значений цены

Как показывают *рисунки 2 и 3*, предложенный алгоритм позволяет получить функции принадлежности термов лингвистической переменной, полностью отвечающие всем предъявляемым к ним требованиям [2; 4]. При этом в основание каждого терма попадает приблизительно равное количество значений. Напротив, функции принадлежности, построенные без учета распределения экспериментальных данных (*рисунок 4*), требуемыми свойствами не обладают и нуждаются в дополнительной аппроксимации.

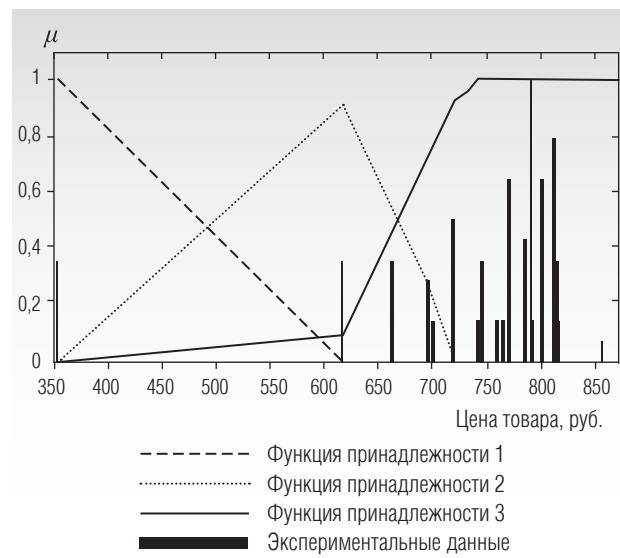


Рис. 4. Функции принадлежности трех термов лингвистической переменной «Цена товара», построенные без учета распределения экспериментальных данных для выборки из 100 значений цены

### Заключение

Предложенный алгоритм позволяет автоматизировать процесс построения функций принадлежности термов числовой лингвистической переменной, параметры которых могут быть сохранены в базе данных и затем использованы при формировании нечетких запросов на языке SQL. В отличие от известных методов автоматизированного построения функций принадлежности данный алгоритм прост в реализации, не требует существенных вычислительных ресурсов и привлечения специализированного программного обеспечения, не нуждается в настройке параметров или формировании обучающей выборки, обеспечивает формирование произвольного, заданного пользователем количества функций принадлежности треугольной и трапециoidalной формы. Построенные по данному алгоритму функции принадлежности учитывают распределение экспериментальных данных, при-

чем основания соответствующих термов лингвистической переменной содержат приблизительно одинаковое количество выбранных из базы данных значений. Получаемые по алгоритму функции принадлежности обладают всеми требуемыми свойствами [2; 4] и не нуждаются в дополнительной аппроксимации. В то же время алгоритм разработан для решения описанной задачи и не может претендовать на универсальность.

Реализация алгоритма создает условия для поддержки нечеткого поиска в базах данных с помощью ограниченных вычислительных средств языка SQL, повышая тем самым уровень интеллектуальности системы и обеспечивая пользователю средства общения на естественном языке. Определенные по алгоритму функции принадлежности термов лингвистических переменных могут быть использованы в нечетких правилах базы знаний информационной системы и при выполнении нечеткого вывода. ■

### Литература

1. Штовба С.Д. Введение в теорию нечетких множеств и нечеткую логику. [Электронный ресурс]: <http://matlab.exponenta.ru/fuzzylogic/book1> (дата обращения: 14.10.2013).
2. Базы данных. Интеллектуальная обработка информации / В.В. Корнеев и [др.]. М.: Нолидж, 2000.
3. Zadeh L. Fuzzy sets // *Information and Control*. 1965. Vol. 8. No. 3. P. 338–353.
4. Zadeh L. The concept of a linguistic variable and its application to approximate reasoning – II // *Information Sciences*. 1975. Vol. 8. No. 4. P. 301–357.
5. Андрейчиков А.В. Интеллектуальные информационные системы. М.: Финансы и статистика, 2006.
6. Чуйкова Е.Н. Реализация нечеткого выбора оборудования в системе проектирования информационной сети // *Вестник ДГТУ*. 2014. № 3. С. 164–171.
7. Medasani S., Kim J., Krishnapuram R. An overview of membership function generation techniques for pattern recognition // *International Journal of Approximate Reasoning*. 1998. No. 19. P. 391–417.
8. Schwaab A., Nassar S., Filho P. Automatic methods for generation of type-1 and interval type-2 fuzzy membership functions // *Journal of Computer Sciences*. 2015. No. 11. P. 976–987.
9. Kim C., Russell B. Automatic generation of membership function and fuzzy rule using inductive reasoning // *3rd International Conference on Industrial Fuzzy Control and Intelligent Systems (IFIS '93)*. Houston, Texas, 1–3 December 1993. P. 93–96.
10. Chen M., Wang S. Fuzzy clustering analysis for optimizing membership functions // *Fuzzy Sets and Systems*. 1999. No. 103. P. 239–254.
11. Liao T., Celmins A., Hammell R. A fuzzy c-means variant for the generation of fuzzy term sets // *Fuzzy Sets and Systems*. 2001. Vol. 135. No. 2. P. 241–257.
12. Lopes P., Camargo H. Automatic labeling by means of semi-supervised fuzzy clustering as a boosting mechanism in the generation of fuzzy rules // *13th IEEE International Conference on Information Reuse and Integration (IRI)*. Las Vegas, USA, 8–10 August 2012. P. 279–286.
13. Jamsandekar S., Mudholkar R. Fuzzy classification system by self generated membership function using clustering technique // *BVICAM's International Journal of Information Technology*. 2014. Vol. 6. No. 1. P. 697–704.
14. Wu S., Er M., Gao Y. A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks // *IEEE Transactions on Fuzzy Systems*. 2001. Vol. 9. No. 4. P. 578–594.
15. Castellano G., Castiello C., Fanelli A., Mencar C. Knowledge discovery by a neuro-fuzzy modeling framework // *Fuzzy Sets and Systems*. 2005. No. 149. P. 187–207.
16. Refaey M. Automatic generation of membership functions and rules in a fuzzy logic system // *5th International Conference on Informatics and Applications (ICIA 2016)*. Takamatsu, Japan, 14–16 November 2016. P. 117–122.
17. Cheng H., Chen J. Automatically determine the membership function based on the maximum entropy principle // *Information Sciences*. 1997. Vol. 96. No. 3–4. P. 163–182.
18. Nieradka G., Butkiewicz B. A method for automatic membership function estimation based on fuzzy measures // *International Fuzzy Systems Association World Congress (IFSA)*. Cancun, Mexico, 18–21 June 2007. P. 451–460.
19. Homaifar A., McCormick E. Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms // *IEEE Transactions on Fuzzy Systems*. 1995. Vol. 3. No. 2. P. 129–139.
20. Shimojima K., Fukuda T., Hasegawa Y. Self-tuning fuzzy modeling with adaptive membership function, rules, and hierarchical structure based on genetic algorithm // *Fuzzy Sets and Systems*. 1995. Vol. 71. No. 3. P. 295–309.



21. Kaya M., Alhaji R. Integrating multi-objective genetic algorithms into clustering for fuzzy association rules mining // 4th IEEE International Conference on Data Mining (ICDM '04). Brighton, UK, 1–4 November 2004. P. 431–434.
22. An ACS-based framework for fuzzy data mining / T. Hong [et al.] // Expert Systems with Applications. 2009. No. 36. P. 11844–11852.
23. Ishibuchi H., Nozaki K., Tanaka H. Efficient fuzzy partition of pattern space for classification problems // Fuzzy Sets and Systems. 1993. Vol. 59. No. 3. P. 295–304.
24. Permana K., Zaiton S. Fuzzy membership function generation using particle swarm optimization // International Journal of Open Problems in Computer Science and Mathematics. 2010. Vol. 3. No. 1. P. 27–41.
25. Piatetsky-Shapiro G., Connell C. Accurate estimation of the number of tuples satisfying a condition // ACM SIGMOD Record. 1984. Vol. 19. No. 2. P. 256–276.

---

## 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.

## 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, pp. 697–704.
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.