

Моделирование оценки состояния объектов на основе запросов с нечёткими условиями

Е.В. Нурматова, Е.И. Зайцев

МИРЭА – Российский технологический университет, Москва

Аннотация: В работе предлагается один из вариантов построения и использования механизма нечетких запросов для моделирования оценки состояния телекоммуникационной сети. Детально рассмотрена модульная архитектура приложения, описана логика вычисления степеней принадлежности к каждому терму выбранных лингвистических переменных, приведены этапы алгоритма работы с механизмом нечетких запросов.

Ключевые слова: нечеткий запрос, поиск, параметры функций принадлежности, моделирование, оценка состояния, база данных, знания, нечёткая классификация.

Неполнота и неопределённость данных в аналитических системах позволяют моделировать причинные взаимосвязи, выявляемые между концептами некоторой области, увеличить диапазон выборки по запросу. Но оперирование стандартными запросами в таких ситуациях неудобно, так как в качестве параметров требуется задавать численные, а не лингвистические значения [1]. В отличие от поддерживаемых видов select-запросов, результат нечёткого запроса менее восприимчив к ошибкам, сокращениям и учитывает контекст условия поиска, в данном случае поисковой фразы [2].

Архитектура приложения обработки запросов с нечёткими условиями

Архитектура приложения приведена на рис.1 и включает модули:

- добавления данных в таблицу лингвистических переменных(ЛП), их терм-множеств;
- формирования параметров нечеткого запроса: блоки выбора ЛП, терм-множеств, нечетких модификаторов, ослабляющих/усиливающих диапазон значений терм-множеств;

- настройки параметров нечёткого запроса: блок добавления сформированного нечеткого запроса в текстовое поле со всеми выбранными параметрами;
- модуль обработки и отображения результатов запроса с нечетким условием

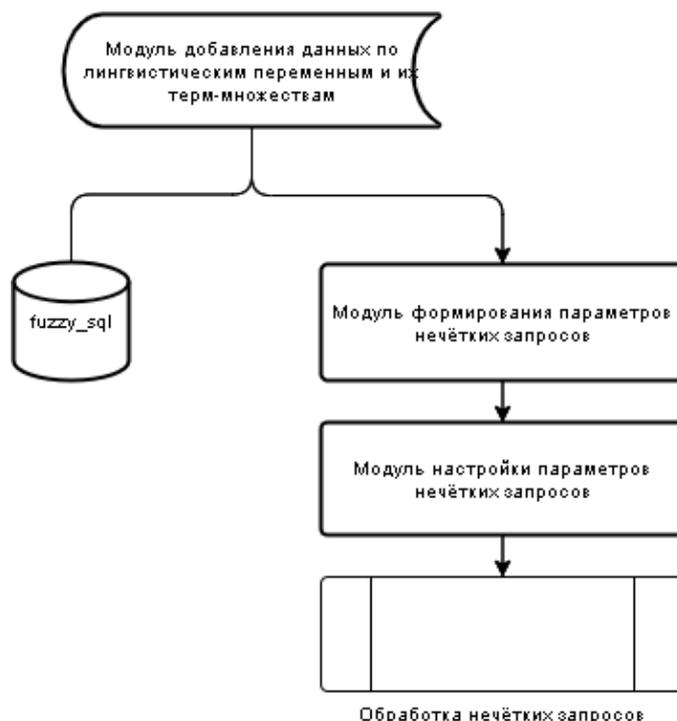


Рис. 1. – Модульная архитектура приложения

Принцип работы реализованного механизма обработки запросов с условиями, в которых присутствует нечеткость, заключается в приеме на вход переменных лингвистических переменных, и затем построенный запрос обрабатывается, после чего формируется выходное значение переменных.

Нечёткий запрос в рамках приложения будет выглядеть как select-запрос с наличием нечетких модификаторов и терм-множеств ЛП [3]. Для лучшего понимания стоит разобрать код, принадлежащий к компонентам фрагмента. К примеру, на рис. 2 представлен класс ResultG, содержащий все необходимые данные для создания нечеткого запроса, выполнения, а также формирования итоговой оценки.

```
Ссылка: 78
static class resultG
{
    public static string y1;
    public static string y2;
    public static string y3;
    public static string y4;
    public static string resultg;
    public static int currentX1 = 0;
    public static int currentX2 = 0;
    public static int currentX3 = 0;
    public static double MFy1;
    public static double MFy2;
    public static double MFy3;
    public static double MFy4;
}
```

Рис. 2. – Класс ResultG

Разберём элементы класса: переменные *y1-y4* нужны для хранения промежуточного результата каждого из состояний, которые получаются после моделирования; переменная *resultg* для хранения итоговой оценки; переменные *currentX1-currentX3* – для хранения идентификатора термножеств ЛП, для последующего извлечения их из базы данных (БД); переменные *MF1-MF4* – для хранения степени принадлежности полученных результатов по отношению к среднему идеальному.

В построении запроса участвует функция *first_object_check*, краткий ее вариант представлен на рис. 3.

```
public void first_object_check()
{
    openConnection();

    string select = "select * from LPTEST";
    MySqlCommand cmd = new MySqlCommand(select, connection);
    MySqlDataReader reader = cmd.ExecuteReader();
    int index_of_table = 0;

    if (reader.HasRows) //если есть данные в БД
    {
        while (reader.Read())
        {
            object runameLP = reader["runameLP"];
            object idLP = reader["KeyLP"];
            object nameLP = reader["nameLP"];
            if ((string)runameLP == comboBox1.SelectedItem.ToString())
            {
                index_of_table = (int)idLP;
                name_of_object1 = (string)nameLP;
            }
        }
    }
    reader.Close();
}
```

Рис. 3. – Функция *first_object_check*

Основная её цель заключается в считывании элементов ЛП, которые были выбраны в выпадающем списке, для последующей работы с ними. Таким образом, осуществляется подключение к БД, затем выполняется

запрос, после чего выполняется сравнение имеющихся элементов в базе с выбранным. При совпадении показателей, происходит запись в ЛП и её индекса в БД, после чего выполняется считывание данных терм-множеств, принадлежащих к ЛП. Это необходимо для записи конкретных характеристик, а также для определения минимального и максимального значений, по которым будут строиться нечеткие модификаторы [4]. Также стоит заметить, что существует вторая функция для второго объекта, которая аналогична функции *first_object_check*.

Основной шаг в построении запроса заключается в использовании нечётких логических функций, их пример обработки приведен на рис. 4:

```
if (radioButton9.Checked)
{
    status = "AND";
    textBox1.Text = "select * from Y3 where " + name_of_object1 + " = " + modifier1 + " "
        + currentname + " " + status + " " + name_of_object2 + " = " + modifier2 + " " + currentname2;
}
if(radioButton8.Checked)
{
    status = "OR";
    textBox1.Text = "select * from Y3 where " + name_of_object1 + " = " + modifier1 + " "
        + currentname + " " + status + " " + name_of_object2 + " = " + modifier2 + " " + currentname2;
}
```

Рис. 4. – Построение нечеткого запроса с использованием логических функций

После того, как разобранные выше функции будут запущены, приложение считывает выбранный элемент нечеткого модификатора и строит нечеткий запрос, состоящий из ЛП, нечеткого модификатора, текущего имени переменной, логическим оператором между объектами. Пример запроса с нечётким условием представлен на рис. 5.

select * from Y3 where x7 = Около Отсутствует AND x8 = Более-меее Средняя

Рис. 5. – Пример нечеткого запроса

Реализованный инструмент использует следующие типы реализации нечетких модификаторов: «более-меее» (выборка 20% от среднего значения, результирующий диапазон значений увеличивается); «около» (аналогично более-меее, но выборка составляет 5%); «выше» (добавляются значения в

большую часть); «ниже» (добавляются значения в меньшую часть); «очень» (диапазон выборки уменьшается на 5%); «равно» (выборка из имеющихся данных).

Программную реализацию нечетких модификаторов можно увидеть на примере выражения «более-менее», представленного на рис.6.

```
while (reader.Read())
{
    object KeyTZ = reader["KeyTZ"];
    object nameTZ = reader["nameTZ"];
    object FKKeyLP = reader["FKKeyLP"];
    object rangemin = reader["rangemin"];
    object rangemax = reader["rangemax"];
    if (currentKeyTZ2 == (int)KeyTZ)
    {
        min_range_before_changes2 = (int)rangemin;
        max_range_before_changes2 = (int)rangemax;
        double test = (1 - differenceML) * (int)rangemin;
        double test1 = (1 + differenceML) * (int)rangemax;
        currentRangeMin2 = (int)test;
        currentRangeMax2 = (int)test1;
    }
}
```

Рис. 6. – Реализация модификатора «более-менее»

В данном случае главными являются переменные, отвечающие за диапазон необходимых значений: *min_range_before_changes2* и *max_range_before_changes2* принимают минимальные и максимальные значения диапазона, загруженные в базу нечетких правил, а также *currentRangeMin* и *currentRangeMax2*, которые содержат измененные данные в зависимости от нечеткого модификатора. Дополнительно следует учесть наличие булевой переменной *resuatl_second_object*, благодаря которой происходит вход в цикл обработки нечеткого запроса. Аналогично реализованы методы представления выражений «около», «выше», «ниже», «очень» и «равно».

Следующим шагом является вычисление степени принадлежности функции $MF(x)$ для значения, полученного из БД. Если нечеткий запрос осуществляет вызов к базе, содержащей структурированные данные, значение функции принадлежности $MF(x)$ будет приниматься единичным. Но существуют промежуточные значения между термами, которые в зависимости от условия могут принадлежать как одной характеристике, так и

другой, поэтому в таблице представлены идеальные условия, при которых степень принадлежности всегда будет равна $MF(x) = 1$.

Расчет значений функции принадлежности зависит от множества условий, в том числе от выбранного нечеткого модификатора, который усиливает или ослабляет пул значений подходящих терм-множеству [5].

Отметим основные критерии:

- $MF(x) = 1$, при x в $[b, c]$, где b и c минимальные и максимальные значения, хранящиеся в таблице терм-множеств;
- $MF(x) = 1$ при «равно», «очень», так как данные нечеткие модификаторы либо усиливают, либо не изменяют данные;
- $MF(x) = 1 - (b - x) / (b - a)$, при условии что x находится в $[a, b]$, где b – минимальное значение из таблицы, a – значение посчитанное в зависимости от условия;
- $MF(x) = 1 - (x - c) / (d - c)$, при условии что x находится в $[c, d]$, где c – максимальное значение из таблицы, d – значение посчитанное в зависимости от условия.

Точно также соотносятся значения функции принадлежности по отношению ко второй характеристике, и затем берется среднее между двумя значениями, чтобы понять, насколько полученный результат имеет вес к идеальному [6].

Информационные структуры приложения

Основная БД, необходимая для хранения параметров нечетких запросов, состоит из таблицы ЛП и их терм-множеств, что можно увидеть на рис. 7.

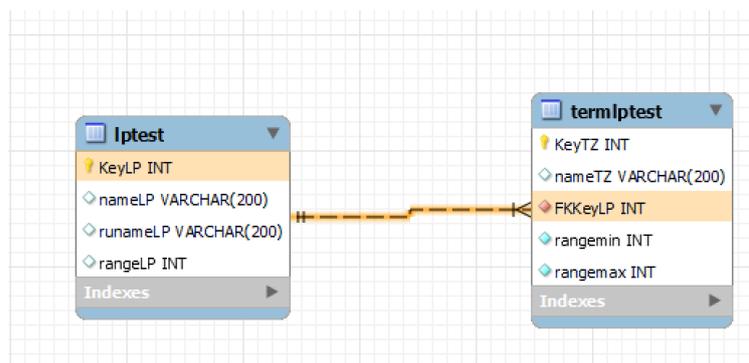


Рис. 7. – Схема данных для определения параметров нечеткого запроса

База данных для обработки результатов нечеткого запроса будет состоять из таблиц, относящихся к моделированию оценки состояния телекоммуникационной сети и включающим: состояние кабельной системы (y_1); работа прикладного программного обеспечения (ППО) (y_2); состояние активного оборудования (y_3); состояние сервера (y_4).

Для них введем переменные x_i , которые будут влияющими факторами на поведение состояния, а также из-за которых их результирующее поведение будет меняться.

Состояние кабельной системы (y_1) определяется переменными: уровень ошибок передачи данных (x_1); уровень шума от внешних источников (x_2); пропускная способность канала связи (x_3).

Для определения статуса *работы прикладного ПО* (y_2) введем следующие переменные: ошибки ПО (x_4); время реакции прикладного ПО (x_5); объем данных (x_6).

Для определения *состояния активного оборудования* (y_3) введем следующие переменные: дефекты в системах электропитания оборудования (x_7); устойчивость работы оборудования (x_8).

Для определения *состояния сервера* (y_4) введем следующие переменные: время реакции сервера (x_9); доступ к серверным ресурсам со стороны пользователя (x_{10}) [7].

Итоговая таблица будет состоять из результирующих y_i состояний.

На рис. 8 представлено содержание таблицы лингвистических переменных и их терм-множеств.

KeyLP	nameLP	ruNameLP	rangeLP	KeyTZ	nameTZ	FKKeyLP	rangemin	rangemax
1	x1	Уровень ошибок передачи данных	100	1	Низкий	1	0	30
2	x2	Уровень шума от внешних источников	100	2	Средний	1	40	60
3	x3	Скорость передачи данных	100	3	Высокий	1	70	100
4	x4	Ошибки ПО	100	4	Отсутствует	2	0	35
5	x5	Время реакции прикладного ПО	100	5	Присутствует	2	60	100
6	x6	Объем данных трафика	100	6	Низкая	3	0	30
7	x7	Дефекты в системах электропитания	100	7	Средняя	3	40	70
8	x8	Устойчивость работы оборудования	100	8	Мгновенная	3	80	100
9	x9	Время реакции сервера	100	9	Отсутствует	4	0	40
10	x10	Доступ к серверным ресурсам со стороны п...	100	10	Присутствует	4	60	100
				11	Минимальная	5	0	30
				12	Средняя	5	40	70
				13	Максимальная	5	80	100
				14	Малый	6	0	30
				15	Средний	6	40	70
				16	Значительный	6	80	100
				17	Отсутствует	7	0	40
				18	Присутствует	7	60	100
				19	Низкая	8	0	30
				20	Средняя	8	40	70
				21	Высокая	8	80	100
				22	Минимальная	9	0	30

Рис. 8. – Значения лингвистических переменных и их терм-множеств

id	x1	x2	x3	resultY1	id	x4	x5	x6	resultY2
1	11	25	95	Устойчивое	1	30	50	93	Оптимальная
2	11	58	95	Устойчивое	2	30	21	60	Оптимальная
3	46	20	60	Удовлетворительное	3	30	22	21	Оптимальная
4	46	59	64	Удовлетворительное	4	70	63	91	Неоптимальная
5	89	20	63	Удовлетворительное	5	70	99	51	Неоптимальная
6	90	20	16	Неустойчивое	6	70	99	11	Неоптимальная
7	93	66	10	Неустойчивое	7	11	21	87	Оптимальная
					8	11	50	67	Оптимальная

id	x7	x8	resultY3	id	x9	x10	resultY4
1	31	91	Отличное	1	11	70	Нормальное
2	20	55	Приемлемое	2	55	73	Нормальное
3	26	21	Удовлетворительное	3	91	80	Удовлетворительное
4	70	90	Приемлемое	4	20	30	Удовлетворительное
5	80	41	Удовлетворительное	5	60	21	Неудовлетворительное
6	90	11	Удовлетворительное	6	96	26	Неудовлетворительное

Рис. 9. – Таблицы состояния кабельной системы, статуса работы прикладного ПО, состояния активного оборудования, состояния сервера

Как осуществляется построение нечёткого запроса можно показать на примере моделирования состояния кабельной системы (см. рисунок 10).

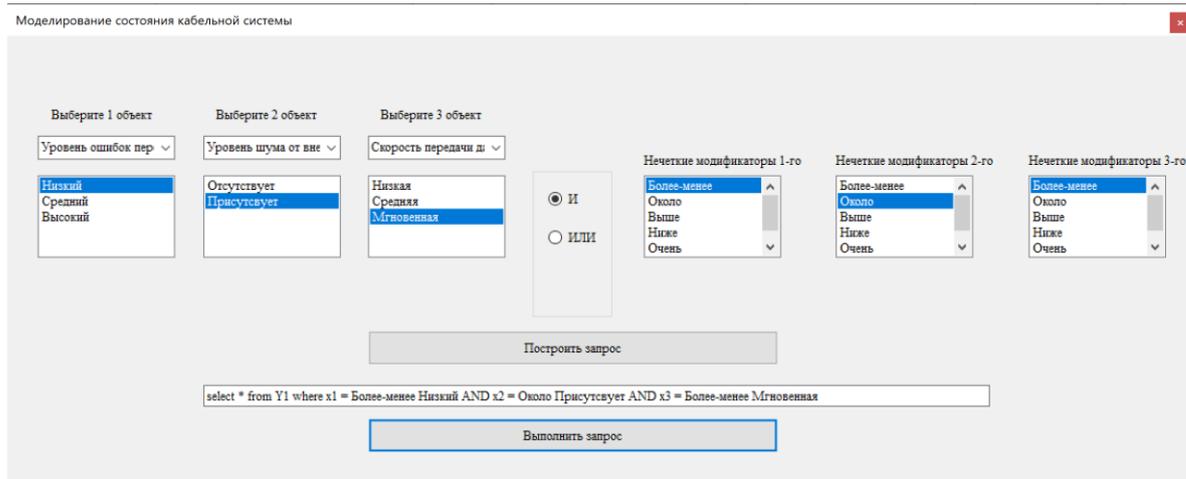


Рис. 10. – Моделирование состояния кабельной системы

После выполнения нечеткого запроса, с соответствующими критериями, результатом оказалось состояние кабельной системы – «Устойчивое», а степень истинности равна 0,78.

Осуществив нечёткие запросы по отношению к каждому объекту исследования и получив их результаты совместно со значениями функции принадлежности, приложение предоставит решение по классу итоговой оценки состояния сети, как показано на рисунке 11.

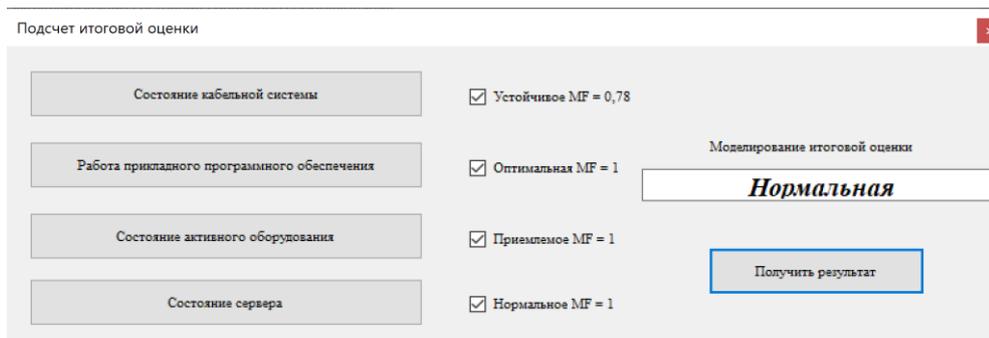


Рис. 11. – Окно вывода итоговой оценки по результату обработки нечетких запросов

Нечеткие запросы перспективно использовать в областях, где осуществляется выбор информации из баз данных с использованием качественных критериев и нечетко сформулированных условий [8,9]. Инструмент нечетких запросов позволяет согласовать формальные критерии и неформальные требования к объектам баз данных и задавать их интервалы

выбора как нечеткие множества. В таком случае данные, не удовлетворяющие какому-то одному критерию, могут быть выбраны, если они имеют хорошие показатели по другим критериям.

Систему, созданную на основе описанной модели, будет удобно использовать при работе с разнообразными ресурсами, содержащими структурированные данные. В этом случае запрос позволит выполнить формирование различных иерархий, учитывающих нечеткие критерии, а механизм нечетких запросов позволяет согласовать формальные критерии и неформальные требования к объектам БД [10], опираясь на подходящие функции принадлежности.

Литература

1. Кривенко М.П Мягкие вычисления в задачах медицинской диагностики // Информатика и её применения. 2021. №15(2). С. 52-59. DOI: 10.14357/19922264210208
2. Паклин Н. Нечеткие запросы к реляционным базам данных // BaseGroup Labs Технологии анализа данных. URL: basegroup.ru/community/articles/fuzzylogic-queries
3. Косоуров Д. А. Применение нечетких запросов в задачах планирования технического обслуживания и ремонта судов // Программные продукты и системы. 2013. №2. URL: swsys.ru/index.php?page=article&id=3454
4. Zadeh L. A., "Fuzzy logic = computing with words," in IEEE Transactions on Fuzzy Systems, vol. 4, no. 2, pp. 103-111, May 1996.
5. Rajabi M., Bohloli B. and Ahangar E., "Intelligent approaches for prediction of compressional, shear and Stoneley wave velocities from conventional well log data: A case study from the Sarvak carbonate reservoir in the Abadan Plain (Southwestern Iran)," in Computers & Geosciences 36, no. 5, 2010, P. 647-664

6. Хтет Сое Паинг, Е Тет Линн, Хан Мьо Хтун Моделирование нечетко-логического управления квадрокоптером // Инженерный вестник Дона. 2020. №7. URL: ivdon.ru/ru/magazine/archive/N7y2020/6533

7. Нурматов А.Т., Нурматова Е.В. Решение задачи диагностики локальной сети в условиях неопределенности // Известия Института инженерной физики. 2011. № 2 (20). С. 40-48.

8. Венцов Н.Н., Долгов В.В., Подколзина Л.А. Об одном способе построения запросов к базе данных на основе аппарата нечеткой логики // Инженерный вестник Дона. 2015. №3 URL: ivdon.ru/ru/magazine/archive/n3y2015/3172

9. Атанов С.К. Программные средства реализации адаптивных моделей с нечеткой логикой // Вестник науки КазАТУ им. С. Сейфуллина, № 2, 2009. С. 27—31. URL: do.gendocs.ru/docs/index-82232.html

10. Коновалов Д. П. К вопросу нечётких запросов к реляционным базам данных // Перспективы развития информационных технологий. 2010. №2.

References

1. Krivenko M.P Informatika i ejo primenenija. 2021. №15(2). pp. 52-59. DOI: 10.14357/19922264210208

2. Paklin N. Nechetkie zaprosy k reljacionnym bazam dannyh [Fuzzy queries to relational databases]. BaseGroup Labs Tehnologii analiza dannyh. URL: basegroup.ru/community/articles/fuzzylogic-queries

3. Kosourov D. AProgrammnye produkty i sistemy. 2013. №2. URL: swsys.ru/index.php?page=article&id=3454

4. Zadeh L.A., IEEE Transactions on Fuzzy Systems, vol. 4, no. 2, pp. 103-111, May 1996.

5. Rajabi M., Bohloli B. and Ahangar E. Computers & Geosciences 36, no. 5, 2010, pp. 647-664.



6. Htet Soe Paing, E Tet Linn, Han M'o Htun. Inzhenernyj vestnik Dona. 2020. №7. URL: ivdon.ru/ru/magazine/archive/N7y2020/6533
7. Nurmatov A.T., Nurmatova E.V. Izvestija Instituta inzhenernoj fiziki. 2011. № 2 (20). pp. 40-48
8. Vencov N.N., Dolgov V.V., Podkolzina L.A. Inzhenernyj vestnik Dona. 2015. №3. URL: ivdon.ru/ru/magazine/archive/n3y2015/3172
9. Atanov S.K. Vestnik nauki KazATU im. S. Sejfullina, № 2, 2009. pp. 27—31. URL: do.gendocs.ru/docs/index-82232.html
10. Konovalov D. P. Perspektivy razvitija informacionnyh tehnologij. 2010. №2.

Дата поступления: 4.12.2023

Дата публикации: 4.01.2024