

**86**

## **НЕЧЕТКАЯ ЛОГИКА**

*Методические указания  
к выполнению лабораторных работ  
для студентов дневного обучения специальностей  
222000 «Инноватика»,  
080500 «Бизнес-информатика»*

Воронеж 2014

Министерство образования и науки РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Воронежский государственный архитектурно-строительный университет»

Кафедра инноватики и строительной физики

**НЕЧЕТКАЯ ЛОГИКА**

Методические указания  
к выполнению лабораторных работ  
для студентов дневного обучения специальностей  
222000 «Инноватика»,  
080500 «Бизнес-информатика»

Воронеж 2014

УДК 004.94

*Составители  
П.А. Головинский, А.А. Дробышев*

Нечеткая логика: метод. указания к выполнению лаб. работ для студ. д/о спец. 222000 «Инноватика», 080500 «Бизнес-информатика»/ Воронежский ГАСУ; сост.: П.А. Головинский, А.А. Дробышев. - Воронеж, 2014. - 40 с.

Излагается краткий теоретический материал по нечеткой логике, приводятся примеры по ее применению. Содержатся контрольные задания и вопросы.

Предназначены для выполнения лабораторных работ по дисциплинам: «Методы и программные системы анализа данных», «Исследование операций», «Нечёткая логика и нейронные сети», для студентов дневного обучения специальностей 222000 «Инноватика», 080500 «Бизнес-информатика»

Ил. 17. Табл. 1. Библиогр.: 3 назв.

УДК 004.94

*Печатается по решению научно-методического совета  
Воронежского ГАСУ*

*Рецензент – С.Н.Дьяконова, к. т. н., доц. кафедры инноватики и  
строительной физики Воронежского ГАСУ*

## ВВЕДЕНИЕ

С точки зрения фундаментальной проблемы искусственного интеллекта, завершен сорокалетний этап развития математического аппарата, и происходит возвращение к проблеме искусственного интеллекта как семантике знаков. Идеи Л. Заде о нечетком разделении объектов и нечетком вычислении объединяют нечеткие множества и логический вывод в терминах естественного языка. Существенным достижением является описание сложных систем, создание средств, сочетающих число и слово, сигнал и понятие, восприятие и абстракцию, непрерывное и дискретное. У данного научного направления многообещающее будущее.

Началом практического применения теории нечетких множеств можно считать 1975г., когда Мамдани и Ассилиан построили первый нечеткий контроллер для управления простым паровым двигателем. Впоследствии нечеткой логике были посвящены работы таких ученых как Холмблад и Остергад, Бартоломей Коско и др.

В настоящее время системы, основанные на нечетких множествах, разработаны и успешно внедрены во многих областях, в том числе: управление технологическими процессами, распознавание образов, техническая диагностика, финансовый менеджмент, управление транспортом, биржевое прогнозирование. Опыт разработки систем нечеткого логического вывода свидетельствует, что при сравнимом уровне робастности и прозрачности моделей, сроки и стоимость разработки таких систем значительно меньше, чем при использовании традиционного математического аппарата.

## Лабораторная работа № 1

### Формирование нечетких множеств и операции над ними

#### 1.1. Цель работы

Изучить методы построения нечетких множеств с использованием различных типов функций принадлежности. Ознакомиться с наиболее распространенными логическими операциями над нечеткими множествами.

#### 1.2. Краткие теоретические сведения

**Функции принадлежности.** Инструменты нечеткой логики в составе пакета MATLAB содержат 11 встроенных типов функций принадлежности, формируемых на основе кусочно-линейных функций, распределения Гаусса, сигмоидной кривой, квадратических и кубических полиномиальных кривых. К наиболее простым функциям принадлежности можно отнести треугольную и трапециевидную.

Треугольная функция принадлежности – *trimf* (*triangle membership function*) в параметрическом виде представляет собой набор трех точек, образующих треугольник.

Описание функции:

$$y = \text{trimf}(x, [a b c]),$$

где вектор  $x$  – базовое множество, на котором определяется функция принадлежности. Величины  $a$  и  $c$  задают основание треугольника,  $b$  – его вершину.

В аналитическом виде треугольная функция принадлежности может быть задана следующим образом (рис.1а):

$$f(x, a, b, c) = \begin{cases} 0, & x < a, \\ \frac{x-a}{b-a}, & a \leq x \leq b, \\ \frac{c-x}{c-b}, & b \leq x \leq c, \\ 0, & x > c. \end{cases}$$

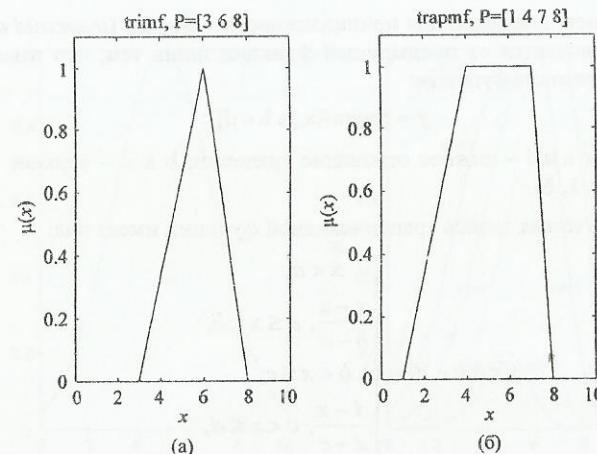


Рис. 1. Треугольная (а) и трапециевидная (б) функции принадлежности

Рассмотрим примеры использования различных функций принадлежности в системе MATLAB.

**Пример №1. Программа использования функций принадлежности trimf.**

```
x = 0:0.1:10; - задается базовое множество
y = trimf(x,[3 6 8]); - определяется треугольная функция
принадлежности
plot(x,y); - выводится график функции
title('trimf(x,P),P =[3 6 8]'); - название графика
xlabel('x'); - подписывается график под осью абсцисс
ylabel('μ(x)'); - подписывается график под осью абсцисс (выводится
надпись «μ(x)»)
```

Для вывода математических формул, содержащих индексы, дроби, корни, буквы греческого алфавита и т.п. в MATLAB используются интерпретаторы *TeX* и *LaTeX*. Соответствующую информацию можно найти в справочной системе MATLAB, для этого в строке поиска нужно ввести «*Mathematical Symbols, Greek Letters, and TEX Characters*» или «*Text Properties*».

Трапециевидная функция принадлежности – trapmf (*trapezoid membership function*) – отличается от предыдущей функции лишь тем, что имеет верхнее основание. Описание функции:

$y = \text{trapmf}(x, [a \ b \ c \ d]);$

где параметры  $a$  и  $d$  – нижнее основание трапеции;  $b$  и  $c$  – верхнее основание трапеции (рис. 1, б).

Аналитическая запись трапециевидной функции имеет вид:

$$f(x, a, b, c, d) = \begin{cases} 0, & x < a, \\ \frac{x-a}{b-a}, & a \leq x < b, \\ 1, & b < x \leq c, \\ \frac{d-x}{d-c}, & c < x \leq d, \\ 0, & x > d. \end{cases}$$

Одно из основных достоинств треугольных и трапециевидных функций принадлежности – их простота. На основе функции распределения Гаусса можно построить функции принадлежности двух видов: простую функцию принадлежности Гаусса и двухстороннюю, образованную с помощью различных функций распределения Гаусса. Первая из них обозначается gaussmf а вторая – gauss2mf.

Простая функция принадлежности Гаусса. Описание функции:

$y = \text{gaussmf}(x, [a, c]).$

Симметричная функция Гаусса зависит от двух параметров  $a$  и  $c$  (рис. 2а):

$$f(x, \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}}.$$

**Пример 2.** Программа использования функции принадлежности gaussmf.

```
x = 0 : 0.1 : 10;
y = gaussmf(x, [2 5]);
plot(x, y)
title('gaussmf');
xlabel('x');
ylabel('mu(x)');
```

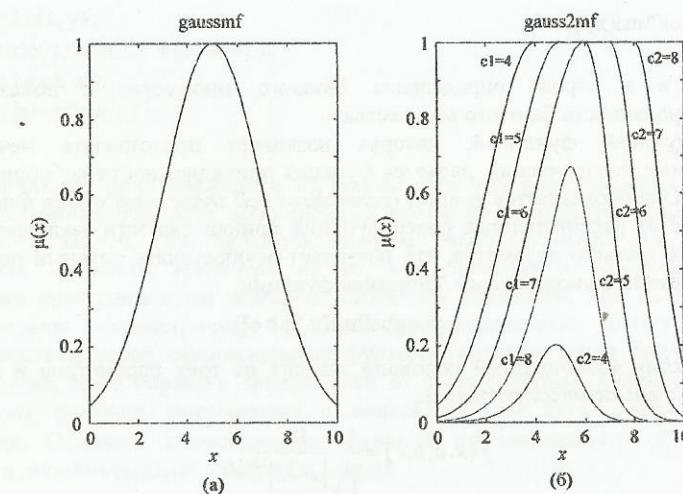


Рис. 2. Простая (а) и двухсторонняя (б) функции принадлежности Гаусса

Двухсторонняя функция принадлежности Гаусса. Описание функции:

$y = \text{gauss2mf}(x, [a1 \ c1 \ a2 \ c2]).$

Выражение является комбинацией двух различных функций распределения Гаусса. Первая определяется параметрами  $a1$  и  $c1$  и задает форму левой стороны, а вторая (параметры  $a2$ ,  $c2$ ) – правой стороны функции принадлежности. Если  $c1 < c2$ , то функция gauss2mf достигает своего максимального значения на уровне 1. Иначе – максимальное значение функции меньше 1 (рис. 2б).

**Пример 3.** Программа использования функции принадлежности gauss2mf.

```
x = (0 : 0.1 : 10)';
y1 = gauss2mf(x, [2 4 1 8]);
y2 = gauss2mf(x, [2 5 1 7]);
y3 = gauss2mf(x, [2 6 1 6]);
y4 = gauss2mf(x, [2 7 1 5]);
y5 = gauss2mf(x, [2 8 1 4]);
plot(x, y1, x, y2, x, y3, x, y4, x, y5);
title('gauss2mf');
xlabel('x');
```

```
ylabel('mu(x)');
```

Символ «'» в строке определения базового множества  $x$  показывает транспонированность базового множества.

Следующей функцией, которая позволяет представлять нечеткие субъективные предпочтения, является функция принадлежности «обобщенный колокол». Она обозначается *gbellmf* (*generalized bell shape membership function*). Ее отличие от рассмотренных ранее функций принадлежности заключается в добавлении третьего параметра, что позволяет осуществлять плавный переход между нечеткими множествами. Описание функции:

```
y = gbellmf(x,[a b c]).
```

Функция «обобщенный колокол» зависит от трех параметров и имеет следующую аналитическую запись:

$$f(x, a, b, c) = \frac{1}{1 + \left| \frac{x - c}{a} \right|^{2b}},$$

где параметр  $c$  определяет расположение центра функции принадлежности;  $a$  и  $b$  влияют на форму кривой (рис. 3).

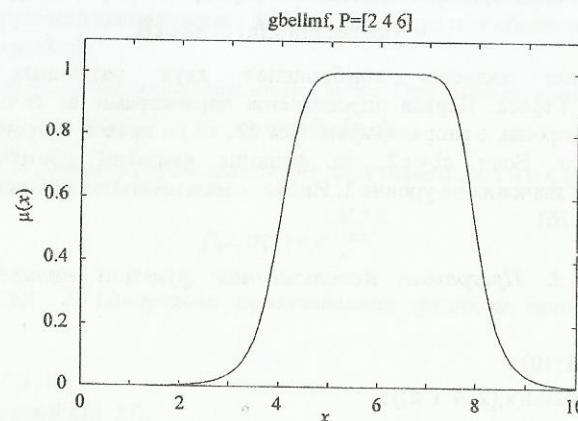


Рис. 3. Функция принадлежности «обобщенный колокол» *gbellmf*

*Пример 4. Программа использования gbellmf.*

```
x = 0:0.1:10;  
y = gbellmf(x,[2 4 6]);
```

```
plot(x,y);  
title('gbellmf,P =[2 4 6]');  
xlabel('x');  
ylabel("mu(x)");
```

Функции принадлежности на основе функции распределения Гаусса и функции принадлежности «обобщенный колокол» отличаются гладкостью и простотой записи. Они являются наиболее используемыми при описании нечетких множеств. Несмотря на то, что гауссова и колоколообразные функции принадлежности обладают свойством гладкости, они не позволяют формировать асимметричные функции принадлежности. Для этих целей предусмотрен набор сигмоидальных функций, которые могут быть открыты либо слева, либо справа в зависимости от типа функции. Симметричные и закрытые функции синтезируют с использованием двух дополнительных сигмоид. Основная сигмоидальная функция принадлежности обозначается *sigmf*, а дополнительные – *dsigmf* и *psigmf*.

Описание основной сигмоидальной функции:

```
y = sigmf(x,[a c]).
```

В аналитической форме сигмоидальная функция *sigmf* записывается следующим образом:

$$f(x, a, c) = \frac{1}{1 + e^{-a(x-c)}}.$$

В зависимости от знака параметра  $a$  рассматриваемая функция принадлежности будет открыта или справа или слева (рис. 4а), что позволит применять ее при описании таких нечетких понятий, как «очень большой», «крайне отрицательно» и др.

Описание дополнительной сигмоидальной функции:

```
y = dsigmf(x,[a1 c1 a2 c2]).
```

Функция принадлежности *dsigmf* зависит от четырех параметров  $a_1$ ,  $c_1$ ,  $a_2$  и  $c_2$  и определяется как разность двух сигмоидальных функций:  $f_1(x, a_1, c_1)$  –  $f_2(x, a_2, c_2)$  (рис. 4б).

Описание дополнительной сигмоидальной функции:

```
y = psigmf(x,[a1 c1 a2 c2]).
```

Функция принадлежности *psigmf*, также, как и предыдущая функция, зависит от четырех параметров  $a_1$ ,  $c_1$ ,  $a_2$ ,  $c_2$  и определяется как произведение двух сигмоидальных функций:  $f_1(x, a_1, c_1)f_2(x, a_2, c_2)$  (рис. 4в).

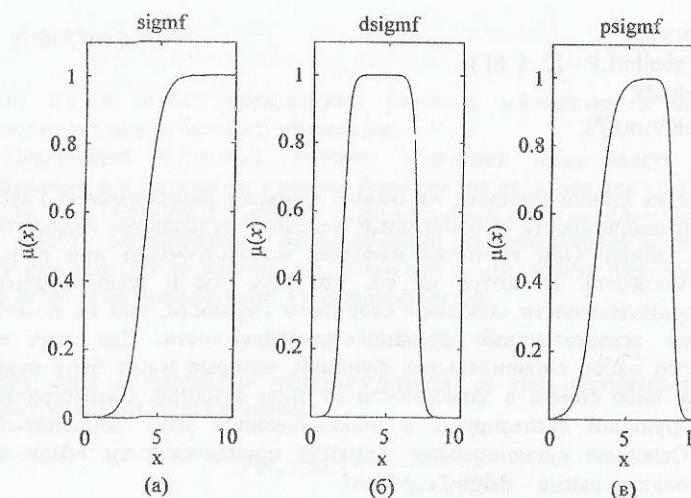


Рис. 4. Сигмоидальные функции принадлежности: (а) – основная односторонняя; (б) – дополнительная двухсторонняя; (в) – дополнительная несимметричная

#### Пример 5. Программа использования сигмоидных функций.

```

x = 0:0.1:10; – определяется базовое множество
subplot(1,3,1); – формируется матрица графиков (3×1) – первый
элемент (текущий)
y = sigmf(x,[2 4]);
plot(x,y); – выводится график в первый элемент матрицы
title('sigmf');
xlabel('x');
ylabel('\mu(x)');
subplot(1,3,2); – выбирается второй текущий элемент
y = dsigmf(x,[5 2 5 7]);
plot(x,y); – выводится график во второй элемент матрицы
title('dsigmf');
xlabel('x');
ylabel('\mu(x)');
subplot(1,3,3); – выбирается третий текущий элемент
y = psigmf(x,[2 3 -5 8]);
plot(x,y); – выводится график в третий элемент матрицы

```

```

title('psigmf');
xlabel('x');
ylabel('\mu(x)');

```

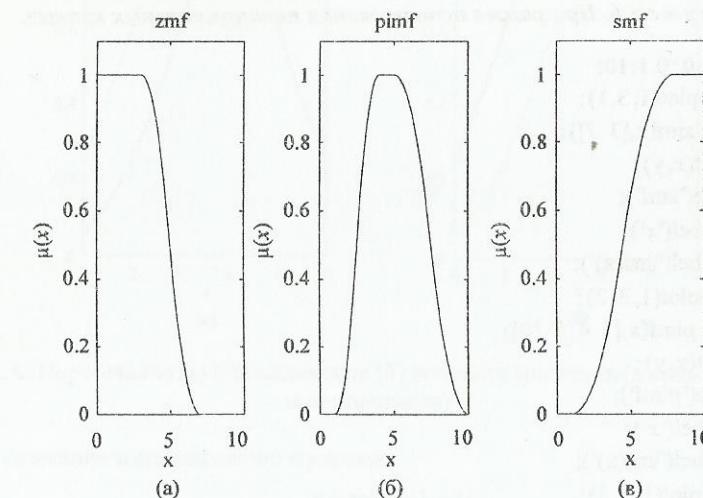


Рис. 5. Полиномиальные функции принадлежности: (а) – Z-функция; (б) – PI-функция; (в) – S-функция

Инструментарий нечеткой логики (*fuzzy logic toolbox*) в составе *Matlab* предоставляет возможность формирования функции принадлежности на основе полиномиальных кривых. Соответствующие функции называются Z – функциями (*zmf*), PI – функции (*pimf*) и S – функции (*smf*). Функция *zmf* представляет собой асимметричную полиномиальную кривую, открытую слева (рис. 5, а), функция *smf* – зеркальное отображение функции *zmf* (рис. 5, б). Соответственно функция *pimf* равна нулю в правом и левом пределах и принимает значение, равное единице, в середине некоторого отрезка (рис. 5, в).

Описание функции:

$$y = zmf(x,[a \ b]).$$

Параметры *a* и *b* определяют экстремальные значения кривой (рис. 5а).

Описание функции:

$$y = pimf(x,[a \ b \ c \ d]).$$

Параметры *a* и *d* задают переход функции в нулевое значение, а параметры *b* и *c* – в единичное (рис. 5б).

Описание функции:

$$y = \text{smf}(x, [a \ b]).$$

Параметры  $a$  и  $b$  определяют экстремальные значения кривой (рис. 5в).

*Пример 6. Программа использования полиномиальных криевых.*

```
x = 0:0.1:10;
subplot(1,3,1);
y = zmf(x,[3 7]);
plot(x,y);
title('zmf');
xlabel('x');
ylabel('μ(x)');
subplot(1,3,2);
y = pimf(x,[1 4 5 10]);
plot(x,y);
title('pimf');
xlabel('x');
ylabel('μ(x)');
subplot(1,3,3);
y = smf(x,[1 8]);
plot(x,y);
title('smf');
xlabel('x');
ylabel('μ(x)').
```

Помимо рассмотренных выше функций, позволяющих представлять нечеткие множества, в MATLAB имеется возможность формировать собственные функции принадлежности или модифицировать встроенные.

**Операции с нечеткими множествами.** Выделяют три основные логические операции с нечеткими множествами: конъюнкцию, дизъюнкцию и логическое отрицание. В среде MATLAB существует возможность определять конъюнктивные и дизъюнктивные операторы с точки зрения минимаксной и вероятностной интерпретаций.

Рассмотрим минимаксную интерпретацию логических операторов, в которой конъюнктивный оператор представляет нахождение минимума –  $\min$  (рис. 6а), а дизъюнктивный – максимума –  $\max$  (рис. 6б).

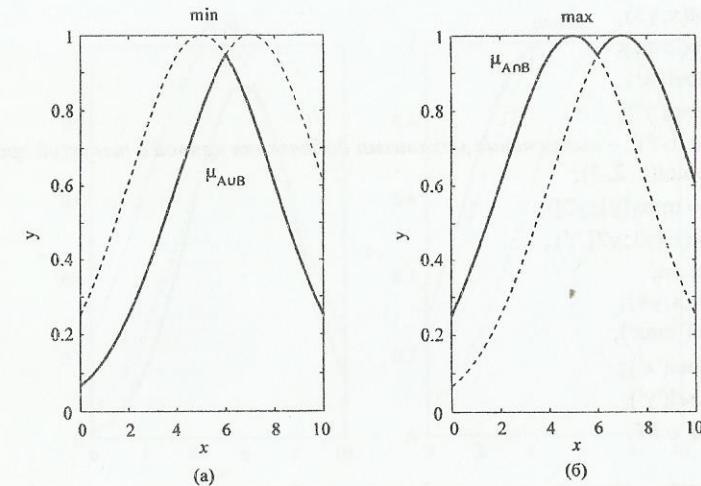


Рис. 6. Пересечение (а) и объединение (б) нечетких множеств (минимаксная интерпретация)

Описание конъюнктивной функции:

$$y = \min([y_1; y_2]).$$

Описание дизъюнктивной функции:

$$y = \max([y_1; y_2]).$$

Параметры  $y_1$  и  $y_2$  представляют собой исходные функции принадлежности. Функция  $\min$  работает со списком функций принадлежности. В MATLAB список оформляется квадратными скобками, а элементы списка разделяются точкой с запятой.

*Пример 7. Программа использования операций  $\min$  и  $\max$ .*

```
x = 0:0.1:10;
subplot(1,2,1);
y1 = gaussmf(x,[3 5]);
y2 = gaussmf(x,[3 7]);
y3 = min([y1;y2]);
plot(x,[y1;y2],':') – построение исходных функций принадлежности пунктирной линией
hold on; – включение механизма добавления кривой в текущий график
```

```

plot(x,y3);
title('min');
xlabel('x');
ylabel('y');
hold off; — выключение механизма добавления кривой в текущий график
subplot(1,2,2);
y4=max([y1;y2]);
plot(x,[y1;y2],':');
hold on;
plot(x,y4);
title('max');
xlabel('x');
ylabel('y');
hold off.

```

Пунктирной линией на графиках изображены исходные функции принадлежности, а сплошной линией – результат действия логических операторов.

Минимаксная интерпретация является наиболее распространенной при построении нечетких систем. Тем не менее, на практике довольно часто используется альтернативная вероятностная интерпретация конъюнктивных и дизъюнктивных операторов. MATLAB содержит соответствующие функции.

В рамках данной интерпретации конъюнктивный оператор представляет собой оператор вычисления алгебраического произведения – *prod* (рис. 7а), а дизъюнктивный оператор – оператор вычисления алгебраической суммы – *prob0r* (рис. 7б).

Описание функции:

$$y = \text{prod}([y_1; y_2]).$$

Описание функции:

$$y = \text{prob0r}([y_1; y_2]).$$

Параметры  $y_1$  и  $y_2$  представляют собой исходные функции принадлежности.

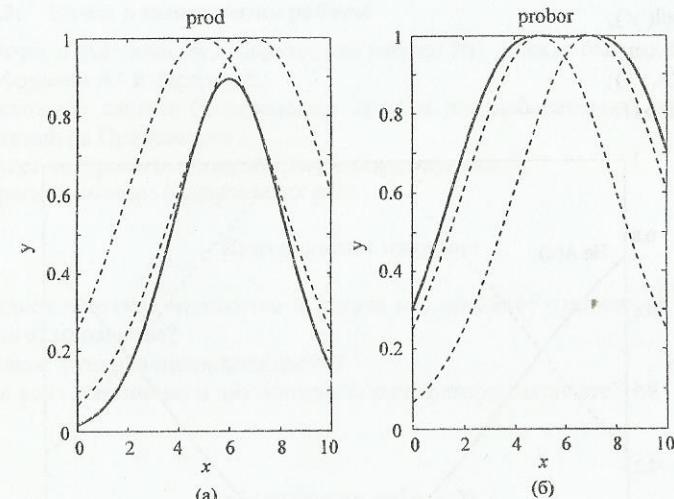


Рис. 7. Пересечение (а) и объединение (б) нечетких множеств (вероятностная интерпретация)

*Пример 8. Программа использования вероятностных операторов конъюнкции и дизъюнкции.*

```

x = 0 : 0.1 : 10;
subplot(1, 2, 1);
y1 = gaussmf(x, [3 5]);
y2 = gaussmf(x, [3 7]);
y3 = prod([y1; y2]);
plot(x, [y1; y2], ':');
title('prod');
xlabel('x');
ylabel('y');
hold on;
plot(x, y3);
hold off;
subplot(1, 2, 2);
y4 = probor([y1; y2]);
plot(x, [y1; y2], ':');
title('prob0r');
xlabel('x');

```

```

ylabel('y');
hold on;
plot(x,y4);
hold off.

```

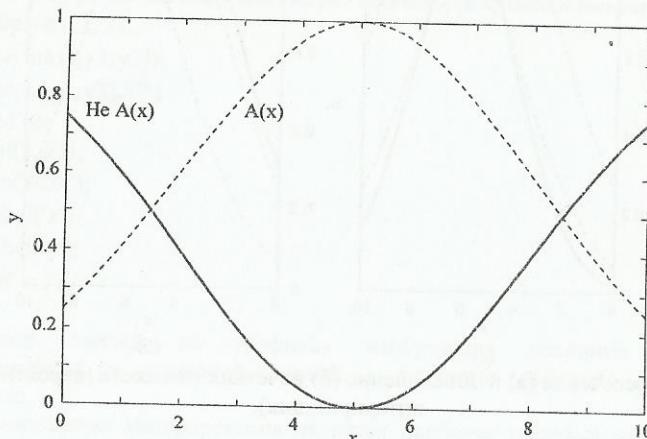


Рис. 8. Дополнение нечеткого множества

Дополнение нечеткого множества есть не что иное, как математическое представление верbalного выражения «НЕ  $A$ » (рис. 8), где  $A$  – нечеткое множество, описывающее некоторое размытое суждение.

Описание функции дополнения:

$$y = 1 - y^*,$$

где  $y^*$  – исходная функция принадлежности.

#### *Пример 9. Программа использования операции дополнения.*

```

x = 0:0.1:10;
y1=gaussmf(x,[3 5]);
y = 1-y1;
plot(x,y1,:);
xlabel('x');
ylabel('y');
hold on;
plot(x,y);
hold off;

```

### 1.3. Отчет о выполнении работы

Отчет о выполнении лабораторной работы №1 должен быть выполнен на листах формата А4 и содержать:

- 1) исходные данные (контрольные задания для лабораторной работы №1 указаны в Приложении);
- 2) текст программы с подробными комментариями;
- 3) краткие выводы о результатах работы.

### Контрольные вопросы

1. Что такое нечеткое множество и каково его основное отличие от обычного (четкого) множества?
2. Что такое функция принадлежности?
3. Какие конъюнктивные и дизъюнктивные операторы вы знаете?

## Лабораторная работа № 2

### Моделирование нечеткой системы инструментами нечеткой логики

#### 2.1. Цель работы

Изучить метод построения нечеткой системы инструментами нечеткой логики.

#### 2.2. Краткие теоретические сведения

В составе MATLAB присутствуют пять основных средств графического интерфейса пользователя (ГИП), которые обеспечивают доступ к инструментам нечеткой логики: редакторы системы нечеткого вывода (СНВ), функции принадлежности, правил вывода, а также средства просмотра правил и поверхности вывода. Эти средства связаны между собой динамически и производимые изменения в одном из них влекут изменения в других.

*Редактор СНВ* предоставляет возможность формирования проектируемой системы на высоком уровне абстракции: количество входных и выходных переменных, наименование переменных.

*Редактор* функций принадлежности используется для определения формы функций принадлежности, ассоциированных с каждой переменной.

*Редактор правил вывода* применяется для редактирования списка правил, которые определяют поведение проектируемой системы.

*Средство просмотра правил вывода* используется в целях диагностики и может показывать, например, активность правил или форму влияния отдельных функций принадлежности на результат нечеткого вывода.

Средство просмотра поверхности вывода используется для отображения зависимости выхода системы от одного или двух входов. Другими словами, оно генерирует и выводит карту поверхности вывода разработанной СНВ.

**Редактор СНВ.** Построение нечетких систем по Мамдани. Для построения создаваемой системы в командной строке основного окна MATLAB необходимо набрать команду *fuzzy*. Окно редактора новой СНВ содержит входную, обозначенную «*input 1*» и выходную — «*output 1*» переменные. По умолчанию инструмент нечеткой логики предлагает создавать СНВ типа Мамдани.

Для того чтобы добавить новую переменную, необходимо выбрать в меню *Edit* соответствующий пункт (для входной переменной — *Add input*, для выходной — *Add output*). Изменение наименования переменной происходит по шагам.

Шаг 1. Отмечается переменная, которую необходимо переименовать.

Шаг 2. В поле редактирования изменяется наименование переменной по умолчанию на имя, предложенное пользователем.

Сохранение проектируемой системы в рабочее пространство среды MATLAB (в переменную) производится с помощью пункта меню *File — Save to workspace as...*. В этом случае данные сохраняются до окончания сеанса работы с MATLAB. Для сохранения данных на диске после окончания сеанса работы применяется соответствующий пункт того же меню — *Save to disk as...*.

**Редактор функций принадлежности.** Следующим шагом в построении нечеткой модели средствами инструментов нечеткой логики является ассоциирование набора функций принадлежности с каждой входной и выходной переменной. Данная операция производится в редакторе функций принадлежности тремя способами, активизировать которые можно:

- выбором в меню *View* пункта *Edit Membership Functions...*;
- двойным щелчком мыши на изображении соответствующей переменной (входной или выходной);
- набором в командной строке оператора *mfedit*.

С помощью редактора функций принадлежности можно отображать и редактировать любые функции принадлежности, ассоциированные (связанные) со всеми входными и выходными переменными разрабатываемой СНВ.

Связывание функции принадлежности с именем переменной происходит следующим образом:

- выбирается переменная по имени из набора графических объектов окна редактора функции принадлежности;
- указывается диапазон изменения значений для базовой и видимый диапазон для текущей переменных;
- в меню *Edit* выбирается пункт *Add MFs...*. В появившемся окне выбирают вид функций принадлежности и их количество.

Редактируют функции принадлежности текущей переменной двумя способами: используя графическое окно функций принадлежности или изменения

характеристики функции принадлежности (наименование, тип и числовые параметры). При выборе необходимой функции принадлежности в графическом окне допускается плавное изменение кривой с помощью мыши.

Таким образом, при построении СНВ необходимо с помощью редактора функций принадлежности определить соответствующие функции для каждой из входных и выходных переменных.

**Редактор правил вывода.** После того как указано количество входных и выходных переменных, определены их наименования и построены соответствующие функции принадлежности, в СНВ необходимо включить правила вывода. Для этого в меню *View* выбирается пункт *Edit Rules...* или в командной строке MATLAB набирается команда *ruleedit*.

Основываясь на описаниях входных и выходных переменных, определенных в редакторе функций принадлежности, редактор правил вывода формирует структуру правила автоматически. От пользователя требуется лишь связать значения входных и выходных переменных, выбирая из списка заданных ранее функций принадлежности и определить логические связи между ними. Также допускается использование логического отрицания (*НЕ*) и изменение весов правил в диапазоне от 0 до 1.

Правила вывода могут отображаться в окне в различных форматах, которые определяются путем выбора соответствующего пункта подменю *Format* меню *Options*. По умолчанию используется *расширенный* формат отображения правил вывода (*verbose form*):

```
If (input_1 is[not] mf_lj1) <and, or>...
  (input_i is[not] mf_jji)...<and, or>
  (input_n is[not] mf_njn) then
    (output_l is[not] mf_n + 1 jn + 1)<and,or>...
    (output_k is[not] mf_k + njk+n) <and,or>...
    (output_m is[not] mf_m + njm + n) (w),
```

где *i* — номер входной переменной;

*ji* — номер функции принадлежности *i*-й переменной;

*k* — номер выходной переменной;

*n* — количество входных переменных;

*m* — количество выходных переменных;

*w* — вес правила.

Круглые скобки заключают в себе обязательные параметры, квадратные — необязательные, а угловые — альтернативные параметры (один на выбор).

Кроме формата по умолчанию, существуют еще два вида форматов отображения правил: *символьный* (*symbolic form*) и *индексный* (*indexed form*). Символьный формат имеет следующий вид:

```

(input_l<~,==>mf_lj1) <&,| >...
(input_i<~,==>mf_iji )...<&, |>
(input_n<~,==>mf_njn) =>
(output_l<~,==>mf_n + ljn+1) ...<&, |>
(output_k<~,==>mf_k + njk+n)<&, |>...
(output_m<~,==>mf_m + njm+n) (w)

```

Отличие символьного формата от расширенного состоит в том, что вместо словесной интерпретации связок используется символьная (символы «&» и «|» — соответственно определяют логическое «И» и логическое «ИЛИ», символ «~» — логическое отрицание, а символ «=>» является разделителем условной и заключительной частей правила (антецедента и консеквента). Общее описание правила вывода в индексном формате может быть представлено в следующем виде:

[-]1j<sub>1</sub>...[-]ij<sub>i</sub>...[-]nj<sub>n</sub>[-]n+1j<sub>n+1</sub>.[-]k+nj<sub>k+1</sub>...[-]m+nj<sub>m+n</sub>(w):<1,2>.

Здесь порядок следования чисел соответствует очередности вводимых переменных, причем символ «::» разделяет правило на условную и заключительную части. До двоеточия записывается порядковый номер соответствующей функции принадлежности, после двоеточия — вид логической связки («1» — логическое И, «2» — логическое ИЛИ). Логическое отрицание задается символом «~».

После определения правил вывода в одноименном редакторе можно утверждать, что СНВ полностью создана.

#### *Пример 10. Создание СНВ.*

Рассмотрим следующую ситуацию. Необходимо оценить степень инвестиционной привлекательности конкретного бизнес-проекта на основании данных о ставке дисконтирования и периоде окупаемости.

**Шаг 1.** Вызываем редактор для создания СНВ, набирая в командной строке *fuzzy*. Добавляем входную переменную посредством выбора в меню *Edit* пункта *Add input*. В результате получаем следующую структуру СНВ: два входа, механизм нечеткого вывода по Мамдани, один выход. Объявляем первую переменную как *discont*, а вторую — *period*, которые соответственно будут представлять ставку дисконтирования и период окупаемости бизнес-проекта. Наименование выходной переменной, на основании которой принимается решение о степени инвестиционной привлекательности бизнес-проекта, задается как *rate*. Сохраним создаваемую модель под именем *Invest*. На рис. 9 представлено текущее состояние окна редактора СНВ.

**Шаг 2.** Каждой входной и выходной переменной поставим в соответствие набор функций принадлежности. Данная процедура реализуется в редакторе функций принадлежности. Для *discont* определяем диапазон базовой переменной от 5 до 50 (единица измерения — проценты). Такой же диапазон выбираем для ее отображения. Добавим три функции принадлежности, тип которых — *trimf*. Последовательно выделяя мышью отдельные функции принадлежности, присвоим наименования — *small*, *middle*, *big* соответственно небольшой, средней и большой ставке дисконтирования. Окно редактора функций принадлежности в текущем состоянии показано на рис. 10. Переменной *period* диапазон базовой переменной определен равным [3, 36] (единица измерения — месяцы), поставлены в соответствие три функции принадлежности типа *gaussmf* с наименованиями — *short*, *normal*, *long*. Таким образом, переменная срока окупаемости бизнес-проекта будет принимать следующие значения: короткий, обычный и длительный срок окупаемости. Наконец, для переменной *rate* определяем: базовая переменная изменяет значение в диапазоне [0, 1], семантика описывается тремя функций принадлежности типа *trimf* с наименованиями: *bad*, *normal*, *good* (рис. 10).

**Шаг 3.** Заключительным этапом построения СНВ является определение набора правил, которые задают связь входных переменных с выходными. Для этого в редакторе правил вывода определим (выбираем соответствующие значения в окне редактора правил):

```

ЕСЛИ discont = small И period = short TO rate = good
ЕСЛИ discont — HE small И period = long TO rate = bad
ЕСЛИ discont = middle И period = normal TO rate = normal
ЕСЛИ discont — big И period = short TO rate = normal

```

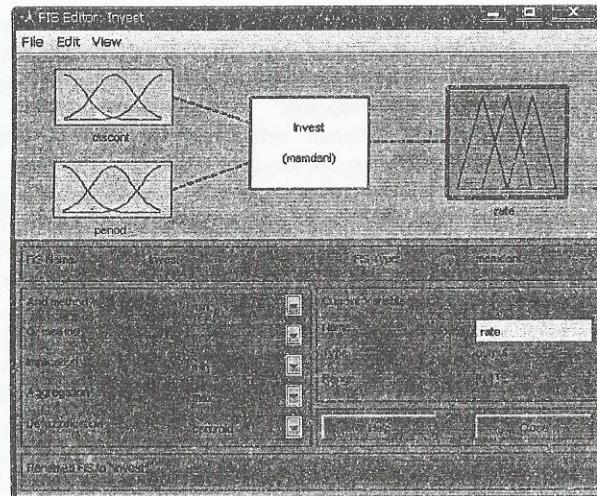


Рис. 9. Окно редактора системы нечеткого вывода

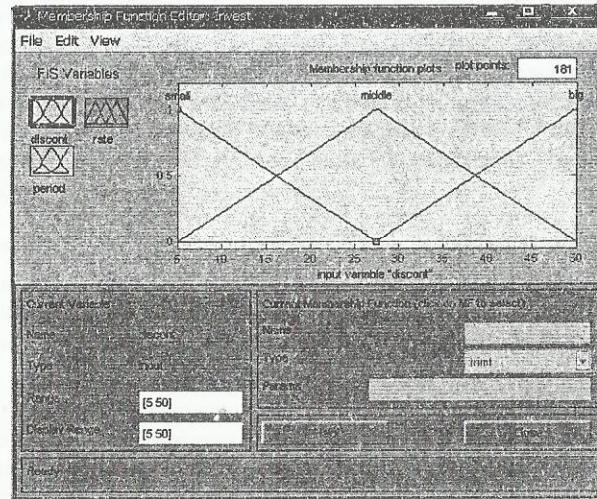


Рис. 10. Окно редактора функций принадлежности

Текущее состояние окна редактора правил вывода показано на рис. 11. В расширенном формате отображения указанные правила вывода представляются следующим образом:

```
if(discont is small) and (period is short) then (rate is good) (1)
if (discont is not small) and (period is long) then (rate is bad) (1)
if(discont is middle) and (period is normal) then (rate is normal) (1)
if(discont is big) and (period is short) then (rate is normal) (1)
```

При изменении формата на символьный правила вывода будут иметь вид:

```
(discont == small) & (period == short) => (rate = good) (1)
(discont ~= small) & (period == long) => (rate = bad) (1)
(discont == middle) & (period == normal) => (rate = normal) (1)
(discont == big) & (period == short) => (rate = normal) (1)
```

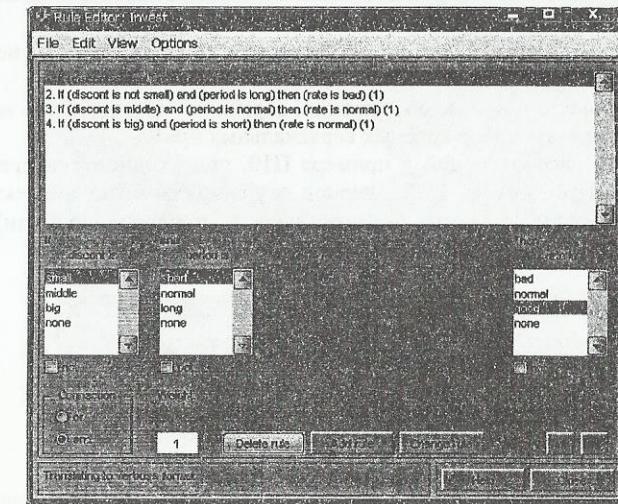


Рис. 11. Окно редактора правил вывода

Наконец, то же самое, но в индексном формате:

```
1 1,3 (1) : 1
-1 3,1 (1) : 1
2 2,2 (1) : 1
3 1,2 (1) : 1.
```

*Средство просмотра правил вывода.* Данное средство позволяет отобразить процесс нечеткого вывода и получить результат. Главное окно средства просмотра состоит из нескольких графических окон, располагаемых

по строкам и столбцам. Количество строк соответствует числу правил нечеткого вывода, а количество столбцов — числу входных и выходных переменных, заданных в разрабатываемой СНВ. Дополнительное графическое окно служит для отображения результата нечеткого вывода и операции дефазификации. В каждом окне отображается соответствующая функция принадлежности, уровень ее среза (для входных переменных) и вклад отдельной функции принадлежности в общий результат (для выходных переменных).

В нижней части главного окна можно отобразить номера правил вывода в различных форматах вывода, отмечая их мышью. Для изменения формата в меню *Options* выбирается пункт *Rule display format*.

Изменение значений входных переменных допустимо двумя способами:

- путем ввода в поле *Input* записи входного вектора, размерность которого равна количеству входных переменных;
- щелчком мыши в любом графическом окне, которое относится к входной переменной.

Входной вектор в каждом из этих вариантов определения исходных данных будет задавать набор красных вертикальных прямых.

Для СНВ, рассмотренной в примере П10, при входном векторе [15 10] (ставка дисконтирования — 15 %, период окупаемости бизнес-проекта — 10 месяцев) результат (степень инвестиционной привлекательности) будет составлять 0,639 (рис. 12).

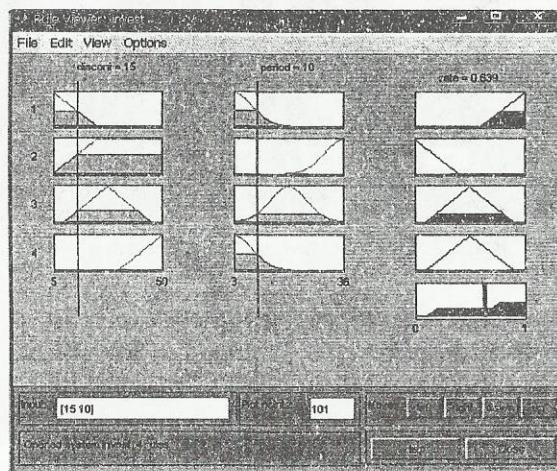


Рис. 12. Окно средства просмотра правил вывода

**Средство просмотра поверхности вывода.** Средство просмотра поверхности вывода позволяет строить трехмерную поверхность как зависимость одной из выходных переменных от двух входных.

Выбор входных и выходных переменных осуществляется посредством ниспадающих меню главного окна рассматриваемого программного средства. Количество выводимых линий по осям X и Y определяется в полях ввода *Xgrids*, *Ygrids*. Поверхность вывода, соответствующая правилам вывода примера П10 показана на рис. 13.

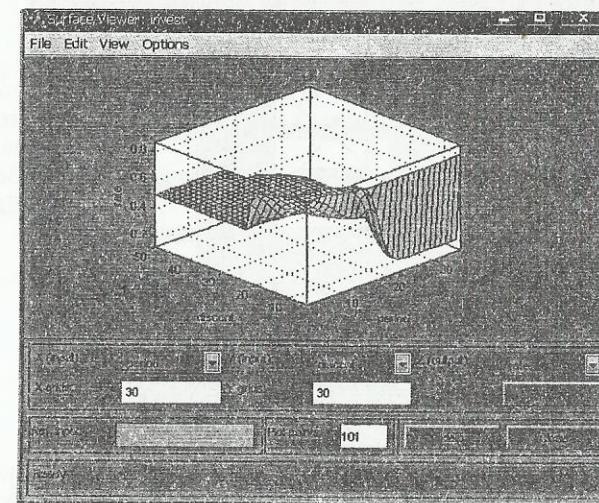


Рис. 13. Окно просмотра поверхности решений

**Построение нечетких систем типа Судженко.** Рассмотрим построение СНВ двумя редакторами — СНВ и функций принадлежности. Для построения СНВ типа Судженко необходимо в меню *File* выбрать пункт *New Sugeno FIS*. Количество входных и выходных переменных определяется так же, как и при построении СНВ типа Мамдани.

**Редактор функций принадлежности.** Для СНВ типа Судженко изменения касаются только схемы определения функций принадлежности для выходных переменных. Инструмент нечеткой логики в среде *Matlab* позволяет разрабатывать два вида нечетких моделей. Первая модель — это нечеткая модель Судженко нулевого порядка. Нечеткое правило вывода имеет следующий вид:

if  $x$  is A and  $y$  is B then  $z = k$ ,

где A и B — нечеткие множества антецедента; k — четко заданная константа консеквента.

Для построения такой модели при добавлении функции принадлежности необходимо выбрать тип — константа (*constant*) и задать в качестве параметра функции принадлежности численное значение соответствующей константы. Вторая модель — нечеткая модель Суджено первого порядка.

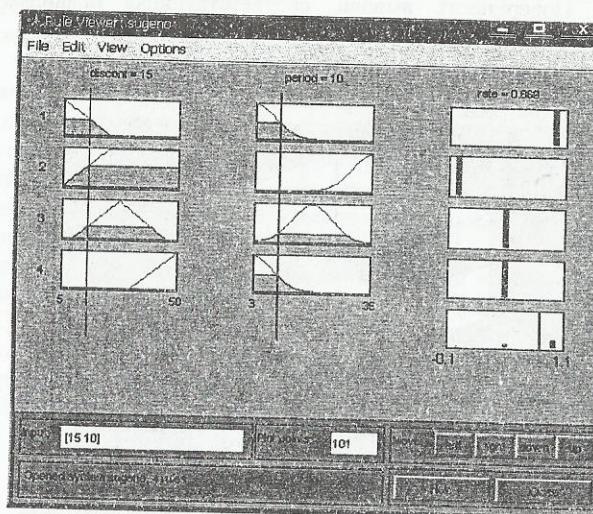


Рис. 14. Окно просмотра правил вывода (вывод по Суджено)

Для нечеткого правила вывода записывается следующим образом:

$$\text{if } x \text{ as } A \text{ and } y \text{ is } B \text{ then } z = p \cdot x + q \cdot y + r,$$

где p, q и r — константы.

В данном случае тип функции принадлежности — линейная зависимость (*linear*). Для определения параметров функции принадлежности необходимо ввести вектор, элементы которого соответствуют численным значениям констант консеквента.

Работа с редактором правил вывода, а также со средствами просмотра правил и поверхности вывода выполняется аналогично случаю построения СНВ по Мамдани.

Пример нечеткого вывода по Суджено с использованием нечеткой модели нулевого порядка и правил вывода, определенных выше, представлен на рис. 14 (выходная переменная имеет три значения: *bad*, *normal*, *good*, которые задаются соответственно тремя константами — 0, 0.5, 1).

### 2.3. Отчет о выполнении работы

Отчет о выполнении лабораторной работы №2 должен быть выполнен на листах формата А4 и содержать:

- 1) исходные данные (контрольные задания для лабораторной работы №2 указаны в Приложении);
- 2) текст программы с подробными комментариями;
- 3) результаты моделирования, краткие выводы о результатах работы.

### Контрольные вопросы

1. Какова структура типовой системы нечеткого вывода?
2. В чем отличие метода нечеткого вывода по Суджено от метода нечеткого вывода по Мамдани?
3. Каким образом формируются антецеденты и консеквенты нечетких правил в MATLAB?

## Лабораторная работа № 3

### Исследование алгоритма нечеткой кластеризации

#### 3.1. Цель работы

Изучить алгоритм нечеткой кластеризации, получить практические навыки решения задач кластеризации методами нечеткой логики.

#### 3.2. Краткие теоретические сведения

**FCM-алгоритм кластеризации.** Алгоритм нечеткой кластеризации называют FCM-алгоритмом (*Fuzzy Classifier Means*, *Fuzzy C-Means*). Целью FCM-алгоритма кластеризации является автоматическая классификация множества объектов, которые задаются векторами признаков в пространстве признаков. Другими словами, такой алгоритм определяет кластеры и соответственно классифицирует объекты. Кластеры представляются нечеткими множествами, и, кроме того, границы между кластерами также являются нечеткими.

FCM-алгоритм кластеризации предполагает, что объекты принадлежат всем кластерам с определенной функцией принадлежности. Степень принадлежности определяется расстоянием от объекта до соответствующих кластерных центров. Данный алгоритм итерационно вычисляет центры кластеров и новые степени принадлежности объектов.

Для заданного множества  $K$  входных векторов  $x_k$  и  $N$  выделяемых кластеров  $c_j$  предполагается, что любой  $x_k$  принадлежит любому  $c_j$  с принадлежностью  $\mu_{jk} \in [0, 1]$ , где  $j$  — номер кластера, а  $k$  — входного вектора. Принимаются во внимание следующие условия нормирования для  $\mu_{jk}$ :

$$\sum_{j=1}^N \mu_{jk} = 1, \forall k = 1, \dots, K;$$

$$0 < \sum_{k=1}^K \mu_{jk} \leq K, \forall j = 1, \dots, N.$$

Цель алгоритма — минимизация суммы всех взвешенных расстояний  $\|x_k - c_j\|$ :

$$\sum_{j=1}^N \sum_{k=1}^K (\mu_{jk})^q \|x_k - c_j\| \rightarrow \min,$$

где  $q$  — фиксированный параметр, задаваемый перед итерациями.

Для достижения вышеуказанной цели необходимо решить следующую систему уравнений:

$$\begin{aligned} \frac{\partial}{\partial \mu_{jk}} \left( \sum_{j=1}^N \sum_{k=1}^K (\mu_{jk})^q \|x_k - c_j\| \right) &= 0, \\ \frac{\partial}{\partial c_j} \left( \sum_{j=1}^N \sum_{k=1}^K (\mu_{jk})^q \|x_k - c_j\| \right) &= 0. \end{aligned}$$

Совместно с условиями нормирования  $\mu_{jk}$  данная система дифференциальных уравнений имеет следующее решение:

$$c_j = \frac{\sum_{k=1}^N (\mu_{jk})^q \cdot x_k}{\sum_{k=1}^K (\mu_{jk})^q},$$

(взвешенный центр гравитации) и

$$\mu_{jk} = \frac{\|x_k - c_j\|^{-1/(q-1)}}{\sum_{j=1}^N \|x_k - c_j\|^{-1/(q-1)}}.$$

Алгоритм нечеткой кластеризации выполняется по шагам.

#### Шаг 1. Инициализация.

Выбираются следующие параметры:

- необходимое количество кластеров  $N$ ,  $2 \leq N \leq K$ ;
- мера расстояний, как Евклидово расстояние;
- фиксированный параметр  $q$  (обычно  $\sim 1.5$ );
- начальная (на нулевой итерации) матрица принадлежности  $U^{(0)} = (\mu_{jk})^{(0)}$  объектов  $x_k$  с учетом заданных начальных центров кластеров  $c_j$ .

#### Шаг 2. Регулирование позиций центров кластеров.

На  $i - i$  итерационном шаге при известной матрице вычисляется  $c_j^{(t)}$  в соответствии с вышеупомянутым решением системы дифференциальных уравнений.

#### Шаг 3. Корректировка значений принадлежности $\mu_{jk}$ .

Учитывая известные  $c_j^{(t)}$ , вычисляются  $\mu_{jk}^{(t)}$ , если  $x_k \neq c_j$ , в противном случае:

$$\mu_{jk}^{(t+1)} = \begin{cases} 1, & l = i, \\ 0, & l \neq i. \end{cases}$$

#### Шаг 4. Остановка алгоритма.

Алгоритм нечеткой кластеризации останавливается при выполнении следующего условия:

$$\|U^{(t+1)} - U^{(t)}\| \leq \varepsilon,$$

где  $\|\cdot\|$  — матричная норма (например, Евклидова норма);  $\varepsilon$  — заранее задаваемый уровень точности.

**Решение задач кластеризации.** Существуют два способа решения задач кластеризации в Matlab: с использованием командной строки или графического интерфейса пользователя.

Рассмотрим первый из указанных способов.

Для нахождения центров кластеров в Matlab имеется встроенная функция fcm, описание которой представлено ниже.

Описание функции:

[center, U, obj\_fcn] = fcm(data, cluster\_n).

Аргументами данной функции являются:

- 1) data — множество данных, подлежащих кластеризации, каждая строка описывает точку в многомерном пространстве характеристик;
- 2) cluster\_n — количество кластеров (более одного).

Функцией возвращаются следующие параметры:

- 1) center — матрица центров кластеров, каждая строка которой содержит координаты центра отдельного кластера;

- 2)  $U$  — результирующая матрица функций принадлежности;  
 3)  $\text{obj\_fcn}$  — значение целевой функции на каждой итерации.

**Пример 11. Программа нечеткой кластеризации.**

```
// загрузка данных, подлежащих кластеризации, из файла
load fcldata.dat;
// определение центров кластеризации (два кластера)
[center,U,obj_fcn]=fcm(fcldata, 2);
// определение максимальной степени принадлежности
// отдельного элемента данных кластеру
maxU = max(U);
// распределение строк матрицы данных между соответствующими
// кластерами
index1=find(U(1,:)==maxU);
index2=find(U(2,:)==maxU);
// построение данных, соответствующих первому кластеру
plot(fcldata(index1,1),fcldata(index1,2),'ko','markerSize',5,'LineWidth',1);
hold on
// построение данных, соответствующих второму кластеру
plot(fcldata(index2,1),fcldata(index2,2),'kx','markerSize',5,'LineWidth',1);
// построение кластерных центров
plot(center(1,1),center(1,2),'ko','markerSize',15,'LineWidth',2)
plot(center(2,1),center(2,2),'kx','markerSize',15,'LineWidth',2)
```

На рис. 15 представлено множество данных, подлежащих кластеризации и найденные центры кластеров для примера 11.

Функция `fcm` выполняется итерационно до тех пор, пока изменения целевой функции превышают некоторый заданный порог. На каждом шаге в командном окне Matlab выводятся порядковый номер итерации и соответствующее текущее значение целевой функции.

| Номер итерации | Значения целевой функции | Номер итерации | Значения целевой функции |
|----------------|--------------------------|----------------|--------------------------|
| 1              | 8,94                     | 7              | 3,81                     |
| 2              | 7,31                     | 8              | 3,80                     |
| 3              | 6,90                     | 9              | 3,79                     |
| 4              | 5,41                     | 10             | 3,79                     |
| 5              | 4,08                     | 11             | 3,79                     |
| 6              | 3,83                     | 12             | 3,78                     |

Табл. 1. Данные в командном окне

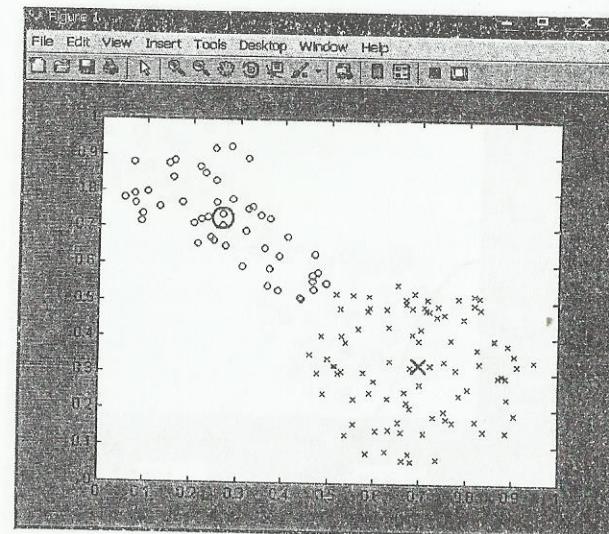


Рис. 15. Множество анализируемых данных и центры кластеров

Для оценки динамики изменения значений целевой функции используется команда построения графика `plot(obj_fcn)`. Результаты примера 11 показаны на рис. 16.

Функцию кластеризации можно вызвать с дополнительным набором параметров: `fcm(data, cluster_n, options)`. Дополнительные аргументы используются для управления процессом кластеризации:

- 1) `options(1)` — показатель степени для матрицы  $U$  (по умолчанию — 2.0);
- 2) `options(2)` — максимальное количество итераций (по умолчанию — 100);
- 3) `options(3)` — предельное изменение значений целевой функции (по умолчанию —  $1e-5$ );
- 4) `options(4)` — отображение информации на каждом шаге (по умолчанию — 1).

Пример определения функции `fcm` с дополнительными параметрами:

```
[center,U,obj_fcn]=fcm(fcldata, 2,[2,100,1e-5,1]).
```

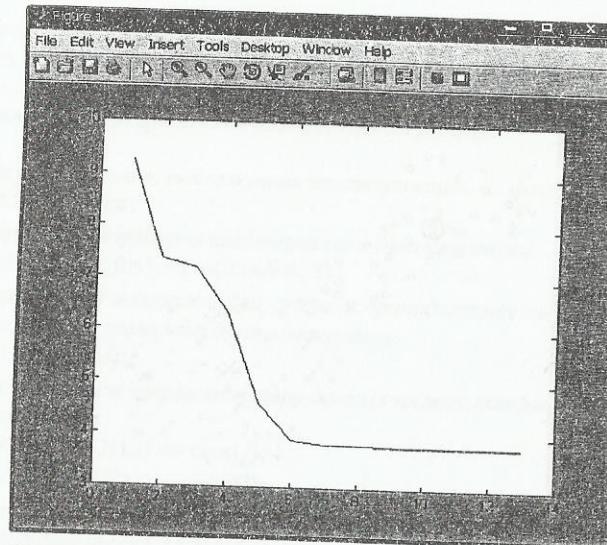


Рис. 16. График изменения значений целевой функции

*Второй способ решения задач кластеризации* в MATLAB основан на использовании ГИП, который вызывается командой `findcluster`. Главное окно инструмента кластеризации показано на рис. 17. Кнопка «Load Data» используется для загрузки подлежащих кластеризации исходных данных следующего формата: каждая строка представляет собой точку в многомерном пространстве характеристик, количество строк соответствует количеству точек (элементов данных). Графическую интерпретацию исходных данных можно наблюдать в одноименном окне главного окна инструмента.

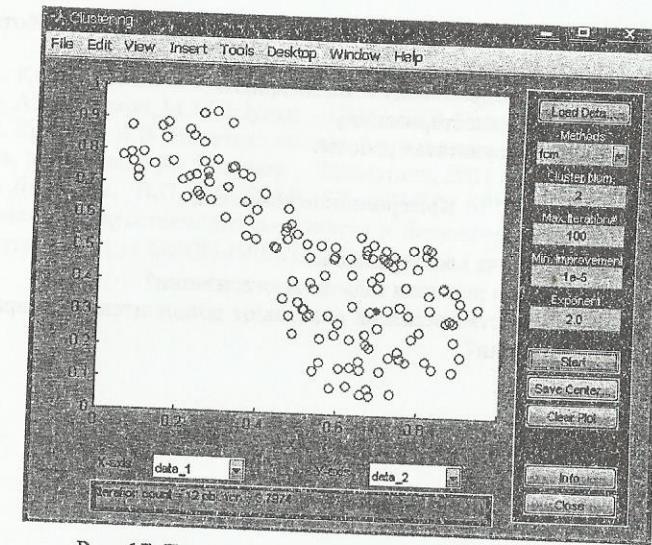


Рис. 17. Главное окно кластеризации в MATLAB

Выбор типа алгоритма кластеризации осуществляется с использованием ниспадающего меню «*Methods*» (пункт меню — `fcm`). Далее определяются параметры алгоритма кластеризации:

- 1) количество кластеров (строка ввода — *Cluster Num.*);
- 2) максимальное количество итераций (строка ввода — *Max Iteration#*);
- 3) минимальное значение улучшения целевой функции (строка ввода — *Min. Improvement*);
- 4) показатель степени при матрице функций принадлежности (строка ввода — *Exponent*).

После определения необходимых значений указанных параметров осуществляется запуск алгоритма кластеризации с помощью кнопки «*Start*». Количество произведенных итераций и значение целевой функции можно просмотреть в нижней части главного окна инструмента кластеризации.

Координаты найденных центров кластеров можно сохранить, щелкнув мышью по кнопке «*Save Center...*». Каждая строка матрицы в файле представляет собой набор координат отдельного кластера. Количество строк соответствует количеству кластеров.

### 3.3. Отчет о выполнении работы

Отчет о выполнении лабораторной работы №3 должен быть выполнен на листах формата А4 и содержать:

- 1) исходные данные (контрольные задания для лабораторной работы №3 указаны в Приложении);
- 2) текст программы с подробными комментариями;
- 3) координаты центров кластеризации;
- 4) краткие выводы о результатах работы.

#### Контрольные вопросы.

1. В чем заключается задача кластеризации?
2. Каковы основные этапы решения задачи кластеризации?
3. Какое влияние на качество решения оказывают дополнительные параметры алгоритма кластеризации?

#### Библиографический список

1. Кетков, Ю.Л. Matlab 7: программирование, численные методы / Ю.Л. Кетков, А.Ю. Кетков, М.М. Шульц. – СПб.: БХВ-Петербург, 2005. – 752 с.
2. Круглов, В.В. Нечеткая логика и искусственные нейронные сети / В.В. Круглов, М.И. Дли, Р.Ю. Голунов. – Физматлит, 2001. – 224 с.
3. Дьяконов, В.П. MATLAB 6.5 SP1/7/7 SP1/7 SP2 + Simulink 5/6. Инструменты искусственного интеллекта и биоинформатики / В.П. Дьяконов, В.В. Круглов. – М.: СОЛОН-ПРЕСС, 2006. – 456 с.

## ПРИЛОЖЕНИЕ

### Контрольные задания для лабораторной работы №1

Исследовать функции `trimf`, `trapmf`, `gaussmf`, `gauss2mf`, `gbellmf`, `sigmf`, `dsmf`, `psigmf`, `zmf`, `pimf`, `smf`, `min`, `max`, `prod`, `prober`, а также операцию дополнения при следующих значениях переменных:

- 1)  $x = 0:0,1:10$ ;  $a = 2$ ;  $b = 5$ ;  $c = 9$ ;  $d = 11$ ;  
 $a1 = 2$ ;  $c1 = 5$ ;  $a2 = 9$ ;  $c2 = 11$ ;  
 $y1 = \text{gbellmf}(x, [a b c])$ ;  
 $y2 = \text{gaussmf}(x, [a c])$
- 2)  $x = 0:0,1:10$ ;  $a = 1$ ;  $b = 2$ ;  $c = 8$ ;  $d = 9$ ;  
 $a1 = 1$ ;  $c1 = 2$ ;  $a2 = 8$ ;  $c2 = 9$ ;  
 $y1 = \text{gauss2mf}(x, [a1, c1, a2, c2])$ ;  $y2 = \text{gaussmf}(x, [a c])$
- 3)  $x = 0:0,1:10$ ;  $a = 0,5$ ;  $b = 4$ ;  $c = 9$ ;  $d = 10$ ;  
 $a1 = 0,5$ ;  $c1 = 4$ ;  $a2 = 9$ ;  $c2 = 10$ ;  
 $y1 = \text{sigmf}(x, [a c])$ ;  $y2 = \text{gaussmf}(x, [a c])$
- 4)  $x = 0:0,1:10$ ;  $a = 0,7$ ;  $b = 3,5$ ;  $c = 9$ ;  $d = 9,5$ ;  
 $a1 = 0,7$ ;  $c1 = 3,5$ ;  $a2 = 9$ ;  $c2 = 9,5$ ;  
 $y1 = \text{trimf}(x, [a b c])$ ;  $y2 = \text{gaussmf}(x, [a c])$
- 5)  $x = 0:0,1:10$ ;  $a = 0$ ;  $b = 2$ ;  $c = 6$ ;  $d = 11$ ;  
 $a1 = 0$ ;  $c1 = 2$ ;  $a2 = 6$ ;  $c2 = 11$ ;  
 $y1 = \text{gbellmf}(x, [a b c])$ ;  $y2 = \text{trimf}(x, [a b c])$
- 6)  $x = 0:0,1:10$ ;  $a = 2$ ;  $b = 3$ ;  $c = 7$ ;  $d = 9$ ;  
 $a1 = 2$ ;  $c1 = 3$ ;  $a2 = 7$ ;  $c2 = 9$ ;  
 $y1 = \text{trimf}(x, [a b c])$ ;  $y2 = \text{trapmf}(x, [a b c d])$
- 7)  $x = 0:0,1:10$ ;  $a = 0$ ;  $b = 1,5$ ;  $c = 6$ ;  $d = 15$ ;  
 $a1 = 0$ ;  $c1 = 1,5$ ;  $a2 = 6$ ;  $c2 = 15$ ;  
 $y1 = \text{trapmf}(x, [a b c d])$ ;  $y2 = \text{gaussmf}(x, [a c])$
- 8)  $x = 0:0,1:10$ ;  $a = 2$ ;  $b = 2$ ;  $c = 7$ ;  $d = 9$ ;  
 $a1 = 2$ ;  $c1 = 2$ ;  $a2 = 7$ ;  $c2 = 9$ ;  
 $y1 = \text{trimf}(x, [a b c])$ ;  $y2 = \text{gauss2mf}(x, [a1 c1 a2 c2])$
- 9)  $x = 0:0,1:10$ ;  $a = 0$ ;  $b = 5$ ;  $c = 7$ ;  $d = 11$ ;  
 $a1 = 0$ ;  $c1 = 5$ ;  $a2 = 7$ ;  $c2 = 11$ ;  
 $y1 = \text{zmf}(x, [a b])$ ;  $y2 = \text{pimf}(x, [a b c d])$
- 10)  $x = 0:0,1:10$ ;  $a = 2$ ;  $b = 3$ ;  $c = 7$ ;  $d = 8$ ;  
 $a1 = 2$ ;  $c1 = 3$ ;  $a2 = 7$ ;  $c2 = 8$ ;  
 $y1 = \text{gbellmf}(x, [a b c])$ ;  $y2 = \text{pimf}(x, [a b c d])$
- 11)  $x = 0:0,1:10$ ;  $a = 1$ ;  $b = 3$ ;  $c = 5$ ;  $d = 8$ ;  
 $a1 = 2$ ;  $c1 = 3$ ;  $a2 = 6$ ;  $c2 = 7$ ;  
 $y1 = \text{gbellmf}(x, [a b c])$ ;  $y2 = \text{pimf}(x, [a b c d])$

- 12)  $x = 0:0,1:10$ ;  $a = 0$ ;  $b = 1$ ;  $c = 3$ ;  $d = 8$ ;  
 $a1 = 2$ ;  $c1 = 4$ ;  $a2 = 5$ ;  $c2 = 9$ ;  
 $y1 = \text{zmf}(x, [a b c])$ ;  $y2 = \text{trimf}(x, [a b c d])$
- 13)  $x = 0:0,1:10$ ;  $a = 1$ ;  $b = 3$ ;  $c = 7$ ;  $d = 8$ ;  
 $a1 = 2$ ;  $c1 = 5$ ;  $a2 = 7$ ;  $c2 = 8$ ;  
 $y1 = \text{trimf}(x, [a b c])$ ;  $y2 = \text{pimf}(x, [a b c d])$
- 14)  $x = 0:0,1:10$ ;  $a = 0$ ;  $b = 2$ ;  $c = 7$ ;  $d = 9$ ;  
 $a1 = 2$ ;  $c1 = 3$ ;  $a2 = 7$ ;  $c2 = 8$ ;  
 $y1 = \text{gbellmf}(x, [a b c])$ ;  $y2 = \text{trimf}(x, [a b c d])$
- 15)  $x = 0:0,1:10$ ;  $a = 2$ ;  $b = 4$ ;  $c = 8$ ;  $d = 9$ ;  
 $a1 = 1$ ;  $c1 = 4$ ;  $a2 = 7$ ;  $c2 = 9$ ;  
 $y1 = \text{gbellmf}(x, [a b c])$ ;  $y2 = \text{pimf}(x, [a b c d])$

### Контрольные задания для лабораторной работы №2

1. Оценить степень инвестиционной привлекательности предоставления кредита на основании данных о проценте по кредиту, срока кредита и суммы кредита.
2. Оценить степень привлекательности получения высшего образования на основании данных о сроке обучения, стоимости обучения и ожидаемом уровне заработной платы по выбранной специальности.
3. Оценить степень удовлетворенности клиентов работой отделения банка на основании данных о среднем количестве клиентов в день и количестве заключаемых договоров.
4. Оценить степень привлекательности приобретения недвижимого имущества в зависимости от удаленности от центра, площади, стоимости.
5. Оценить степень привлекательности приобретения автомобиля в зависимости от стоимости, объема потребляемого топлива, пробеге автомобиля.
6. Оценить автомобиль исходя из состояния кузова, данных о состоянии двигателя и лакокрасочного покрытия.
7. Оценить фильм по данным о сюжете, актерах, съемке.
8. Оценить блюдо исходя из информации о калорийности, полезности и предполагаемом времени потребления.
9. В зависимости от калорийности, содержания белков и углеводов определить будет ли диета являться нормальным питанием, способствовать снижению веса или являться спортивной.
10. Дайте оценку ПК в зависимости от частоты процессора, объема оперативной памяти и объема жесткого диска.
11. Определить спелость ананаса по значениям таких показателей как твердость корки, интенсивности аромата и цвету мякоти.

12. Оценить велосипед по значениям таких показателей как количество скоростей, ширина колес и стоимость велосипеда.
13. Подобрать одеяло в соответствии с такими характеристиками как размер одеяла, степень теплоты одеяла и состав наполнителя.
14. Подобрать пылесос в зависимости от мощности всасывания, наличия аксессуаров и стоимости.
15. Подобрать породу собаки по размеру, агрессивности и обучаемости.

### Контрольные задания для лабораторной работы №3

- 1) fcndata = [0 1, 1.1 2.5, 0.4 2.2, 4.5 2.7, 3.4 1.2, 2.2 1.8, 0.5 0.4, 3 2.2, 1 1.6, 2.3 4, 1 4.7, 0.2 0.7, 0.9 4.4, 0.3 5, 0.1 2.1, 3.1 4.7, 3.2 5, 1 1, 0 1.6, 4 2.2, 1.6 2.2, 5 2.3, 4.1 4.6, 3.3 1, 2.8 2.9, 1.8 0.7, 1.9 3.4, 4 1.5, 3.8 1] cluster\_n = 4
- 2) fcndata = [0.5 1.9, 1.1 2.4, 0.8 2.8, 4.5 2.7, 3.7 1.2, 2.2 1.8, 0.1 0.4, 3 2.2, 1.1 1.6, 2.3 4.2, 1 4.7, 0.2 2.7, 0.9 4.4, 0.8 5, 0 2.1, 3.1 4, 3.2 5, 3.1 1, 0 1.6, 4.5 2, 1.9 2.1, 3.5 2.3, 4.1 4.6, 3.3 1, 2.6 2.9, 1 0.7, 1.9 3.4, 4 1.5, 3.8 1] cluster\_n = 5
- 3) fcndata = [0 5, 1.5 2.1, 3.4 2.1, 4.5 2.4, 3.4 1.2, 2.2 1.8, 0.9 0.2, 3.5 2.1, 1 1.6, 2.3 4, 1 4.7, 0.9 0.7, 0.1 4.9, 0.3 5, 0.1 2.1, 3.7 4.2, 3.2 5, 1 1, 0 1.6, 4 4.2, 1.9 2.2, 2.5 2.3, 2.1 4.6, 3.3 1, 3.8 2.9, 4.8 0.7, 2.9 3.4, 4 1.5, 1.8 1] cluster\_n = 3
- 4) fcndata = [0.7 1.9, 1.8 2.2, 0.4 2.2, 4.5 2.7, 3.4 1.2, 2.2 1.8, 0.5 0.4, 3 2.2, 1 1.6, 2.3 4, 1 4.7, 0.2 0.7, 0.3 2.4, 0.3 5, 0.9 2.8, 3.1 4.7, 1.2 3.5, 1 1, 0 1.7 1.2, 4 2.2, 1.2 2.2, 5 2.3, 4.1 4.6, 1.3 1.7, 2.8 2.9, 1.8 0.2, 1.9 3.4, 4.2 1.9, 3.8 1] cluster\_n = 4
- 5) fcndata = [0 1, 1.9 2.1, 3.4 4.2, 2.5 2.2, 1.4 1.2, 2.1 1.8, 0.5 0.4, 3 2.9, 1 1.6, 2.3 4, 1 4.7, 0.2 0.5, 0.9 4.4, 0.9 5, 4.1 2.9, 3.1 4.2, 3.2 1.5, 1 1, 0 1.6, 4 3.2, 1.6 2.2, 5 2.3, 4.1 2.6, 1.3 1, 1.8 2.1, 1.8 0.7, 1.9 3.4, 4 1.5, 3.3 1] cluster\_n = 5
- 6) fcndata = [0 4, 1.7 3.5, 0.4 4.2, 4.5 2.7, 3.8 1.1, 2.2 1.8, 0.5 0.9, 1.3 2.2, 1 1.6, 2.3 1, 1 4.7, 4.2 2.7, 0.9 4.4, 0.3 5, 4.1 2.6, 3.1 4.7, 3.9 5, 1.7 1, 0.5 1.9, 4 2.2, 2.6 2.8, 5 2.3, 2.1 4.6, 3.2 1, 2.8 2.9, 1.8 0.7, 1.9 3.4, 4 1.5, 3.8 1] cluster\_n = 4
- 7) fcndata = [4 1, 1.2 2, 0 2.2, 4.5 2.7, 3.8 2.2, 2.9 1.8, 0.5 0.9, 3 2.2, 1 1.6, 2.3 4, 1 4.7, 0.2 0.7, 0.9 4.4, 0.3 5, 0.1 2.1, 3.1 4.7, 1.2 5, 1 1, 0 3.6, 1 2.2, 1.7 2.2, 4.2 2.3, 2.1 4.6, 1.3 1, 2.8 2.9, 1.8 2.7, 1.9 3.4, 4 3.5, 3.5 4] cluster\_n = 4
- 8) fcndata = [0.7 1.5, 3.1 2.5, 4.4 1.2, 4.5 1.6, 3.4 1.2, 2.1 1.8, 3.5 0.4, 3 1.2, 1 1.6, 2.3 4, 1 4.7, 0.2 0.7, 0.9 1.4, 0.3 2, 0.1 2.1, 3.7 1.7, 3.9 1, 1 1.8, 0.7 2.6, 4 2.2, 1.6 2.2, 5 2.3, 4.1 2.6, 1.3 1, 2.8 2.1, 1.8 0.7, 1.9 3.4, 4.3 1.5, 2.8 1] cluster\_n = 3

- 9) fcndata = [1.7 2.5, 2.1 2.5, 2.4 1.2, 2.5 1.6, 3.4 2.2, 2.1 2.8, 2.5 0.4, 3 1.2, 1 1.6, 3.3 4, 1 3.7, 3.2 0.7, 0.9 2.4, 0.3 2, 2.1 2.1, 3.7 2.7, 3.9 1, 4 1.8, 3.7 2.6, 4 2.2, 1.6 4.2, 3 2.3, 4.1 3.6, 3.3 1, 2.8 2.1, 3.8 0.7, 1.9 3.4, 2.3 1.5, 3.8 1] cluster\_n = 2
- 10) fcndata = [4.7 2.5, 4.1 2.2, 2.4 1.2, 2.5 1.1, 2.4 2.2, 2.1 2.8, 2.9 0.4, 3 1.2, 1 1.6, 3.1 4.6, 1 3.7, 0.2 3.7, 0.9 1.4, 0.3 2, 2.1 0.1, 3.7 2.7, 3.9 1, 0 1.8, 4.7 2.6, 3 2.2, 2.6 4.2, 3 0.3, 4.1 3.5, 0.3 1, 2.8 2.1, 0.8 0.7, 1.9 3.4, 2.3 4.5, 3.8 1] cluster\_n = 3
- 11) fcndata = [4.7 1.5, 4.1 2.2, 2.4 1.2, 2.5 1.1, 2.4 2.2, 2.1 2.8, 2.9 0.4, 3 1.2, 1 1.6, 3.1 4.6, 1 3.7, 0.2 4.7, 0.9 1.4, 0.3 2, 2.1 4.1, 3.7 2.7, 3.9 1, 0 1.8, 4.7 2.6, 3 2.2, 2.6 4.2, 3 0.3, 4.1 3.5, 0.3 1, 5.8 2.1, 0.8 0.7, 1.9 3.4, 2.3 4.5, 4.5, 3.8 1] cluster\_n = 5
- 12) fcndata = [4.7 3.5, 4.1 2.2, 2.4 1.2, 2.5 1.1, 2.4 2.2, 2.1 2.8, 2.9 0.4, 3 1.2, 1 1.6, 3.1 4.6, 1 3.7, 0.2 4.7, 0.9 1.4, 0.3 2, 2.1 0.1, 3 2.7, 1.9 1, 0 1.8, 0.7 2.6, 3 2.2, 3.6 1.2, 3 0.3, 2.1 3.5, 0.3 1, 2.8 2.1, 0.8 0.7, 1.9 3.4, 2.3 4.5, 3.8 1] cluster\_n = 4
- 13) fcndata = [4.7 2.5, 4.1 2.2, 2.4 1.2, 2.5 2.1, 2.4 2.2, 2.1 2.8, 2.9 0.4, 3 1.2, 1 1.6, 3.1 4.6, 1 2.7, 1 2.3 3.7, 0.9 1.4, 0.3 2, 2.1 3.1, 3.7 2.7, 3.9 1, 0 1.8, 2.7 4.6, 3 2.2, 2.6 4.2, 3 0.3, 5.1 3.5, 0.3 1, 0.8 2.1, 0.8 1.2, 1.9 3.9, 2.3 1.5, 0.8 0.1] cluster\_n = 4
- 14) fcndata = [1.7 2.5, 4.1 2.2, 2.4 1.2, 2.5 1.1, 2.4 2.2, 2.1 2.8, 2.9 0.4, 3 1.2, 1 1.6, 3.1 4.6, 1 4.7, 2.2 3.6, 1.9 1.4, 0.3 2, 2.1 0.1, 1.7 2.7, 2.9 1, 0 1.8, 1.8, 4.7 2.6, 3 2.2, 2.6 4.2, 3 0.3, 4.1 3.5, 0.3 1, 2.8 2.1, 0.8 0.7, 1.9 3.4, 2.4 4.2, 1.7 4] cluster\_n = 5
- 15) fcndata = [4.9 1.9, 2.1 2.2, 2.4 1.2, 2.5 1.1, 3.4 2.2, 2.1 4.8, 2.9 0.4, 3 1.2, 1 1.6, 3.1 4.6, 1 3.5, 0.2 3.7, 0.9 1.4, 0.3 2, 1.1 0.1, 3.7 2.7, 3.2 1, 0 1.8, 4.7 2.6, 3 2.2, 2.6 4.2, 3 1.3, 4.6 1.5, 0.3 1, 2.8 2.1, 0.8 0.5, 4.9 1.4, 2.1 3.5, 3.2 1.6] cluster\_n = 4

## Оглавление

|  |    |
|--|----|
| ВВЕДЕНИЕ .....   | 3  |
| Лабораторная работа № 1  |    |
| Формирование нечетких множеств и операции над ними .....           | 4  |
| Лабораторная работа № 2  |    |
| Моделирование нечеткой системы инструментами нечеткой логики ..... | 17 |
| Лабораторная работа № 3  |    |
| Исследование алгоритма нечеткой кластеризации .....                | 27 |
| Библиографический список рекомендуемой литературы .....            | 35 |
| ПРИЛОЖЕНИЕ .....   | 36 |
| Контрольные задания для лабораторной работы №1 .....               | 36 |
| Контрольные задания для лабораторной работы №2 .....               | 37 |
| Контрольные задания для лабораторной работы №3 .....               | 38 |

## НЕЧЕТКАЯ ЛОГИКА

*Методические указания  
к выполнению лабораторных работ  
для студентов дневного обучения  
специальностей  
222000 «Инноватика»,  
080500 «Бизнес-информатика»*

Составители: д. ф.-м.н., проф. Головинский Павел Абрамович  
к. ф.-м.н. Дробышев Алексей Александрович

Подписано в печать 27.10.14. Формат 60x84 1/16. Уч.-изд. л. 2,0. Усл.-печ. л. 2,1.  
Бумага писчая. Тираж 50 экз. Заказ № 428.

Отпечатано: отдел оперативной полиграфии издательства учебной литературы и  
учебно-методических пособий Воронежского ГАСУ  
394006 Воронеж, ул. 20-летия Октября, 84