

СЕКЦИЯ 4.
Нейро-сетевые технологии

Председатель:
д. ф.-м. н., профессор М. Н. Рычагов

Оглавление

Агамалов О. Н., Костерев Н. В., Лукаш Н. П. ЭКСПЕРТНАЯ СИСТЕМА ОЦЕНКИ ТЕХНИЧЕСКОГО СОСТОЯНИЯ СИСТЕМЫ ВОЗБУЖДЕНИЯ СИНХРОННОГО ГЕНЕРАТОРА В РЕАЛЬНОМ МАСШТАБЕ ВРЕМЕНИ.....	1234
Аманов К. А., Вощанкин С. В., Смирнов А. Б., Черняк Б. Я. ПРИМЕНЕНИЕ МАТЛАВ ПРИ РАЗРАБОТКЕ СИСТЕМ УПРАВЛЕНИЯ БЕНЗИНОВЫМИ ДВИГАТЕЛЯМИ ВНУТРЕННЕГО СГОРАНИЯ	1255
Артюхин В. В., Соломаха А. А., Горбаченко В. И. ИСПОЛЬЗОВАНИЕ ПАКЕТА NEURAL NETWORK TOOLBOX СРЕДЫ МАТЛАВ ДЛЯ ДИАГНОСТИКИ ГЕПАТИТА У ХИРУРГИЧЕСКИХ БОЛЬНЫХ.....	1262
Злобин В. В. РЕШЕНИЕ РЕСУРСОЕМКИХ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ С ПОМОЩЬЮ НЕЙРОСЕТЕЙ	1266
Зюзев А. М., Костылев А. В. СИСТЕМА ДИАГНОСТИКИ ШТАНГОВОЙ ГЛУБИННО-НАСОСНОЙ УСТАНОВКИ НА ОСНОВЕ НЕЙРОННОЙ СЕТИ	1273
Локтионов А. А., Аргынова А. Х., Лось В. Л., Токарский Э. А. РАССТОЯНИЕ МЕЖДУ ОБЪЕКТАМИ ПРИ ПРОГНОЗЕ ПОЛЕЗНЫХ ИСКОПАЕМЫХ.....	1288
Мещеряков В. А. ИДЕНТИФИКАЦИЯ СТРОИТЕЛЬНЫХ МАШИН КАК НЕЛИНЕЙНЫХ ДИНАМИЧЕСКИХ СИСТЕМ НА ОСНОВЕ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ.....	1300
Хрящёв В. В., Соколенко Е. А., Приоров А. Л. СРАВНЕНИЕ ЭФФЕКТИВНОСТИ РЕАЛИЗАЦИИ АЛГОРИТМОВ ОБУЧЕНИЯ НЕЙРОСЕТИ В ЗАДАЧЕ ВОССТАНОВЛЕНИЯ ИЗОБРАЖЕНИЙ	1309
Штовба С. Д., Панкевич О. Д. ПРОЕКТИРОВАНИЕ НЕЧЕТКИХ КЛАССИФИКАТОРОВ В СИСТЕМЕ МАТЛАВ	1320
Юзбашев Д. А. НЕЙРОННЫЕ СЕТИ ТРЕТЬЕГО ПОКОЛЕНИЯ. РАСПОЗНАВАНИЕ ОБРАЗОВ И СКРЫТЫХ ЗАВИСИМОСТЕЙ В ПРОИЗВОЛЬНОМ СИГНАЛЕ С ПОМОЩЬЮ ИМПУЛЬСНЫХ НЕЙРОСЕТЕЙ С ИСПОЛЬЗОВАНИЕМ СРЕДЫ МАТЛАВ.....	1336

УДК 658.012

ПРОЕКТИРОВАНИЕ НЕЧЕТКИХ КЛАССИФИКАТОРОВ В СИСТЕМЕ MATLAB

Штовба С. Д., Панкевич О. Д.

Винницкий национальный технический университет, Винница, Украина,

e-mail: shtovba@ksu.vstu.vinnica.ua

Введение

Классификация на основе нечеткого логического вывода используется при принятии решений в технике, экономике, политике, медицине, биологии и в других областях [1-3]. Системы нечеткого вывода базируются на лингвистических правилах «Если —То». Они обеспечивают хороший баланс между безошибочностью классификации и прозрачностью модели принятия решения.

Пакет **Fuzzy Logic Toolbox** вычислительной системы MATLAB предоставляет широкий набор инструментов для проектирования и исследования систем нечеткого логического вывода с непрерывным выходом. В настоящей статье показано, как расширить **Fuzzy Logic Toolbox** для проектирования нечетких классификаторов, т. е. систем нечеткого логического вывода с дискретным выходом.

Статья организована следующим образом: в разделе 1 излагаются математические модели нечеткого вывода для задач классификации; в разделе 2 ставятся задачи настройки нечеткого классификатора по различным критериям обучения; в разделе 3 приводится программа нечеткой классификации, использующая функции пакета **Fuzzy Logic Toolbox**; в разделе 4 предлагаются программы настройки нечеткого классификатора, использующие **Optimization Toolbox**; в разделе 5 приводятся примеры проектирования нечетких классификаторов с применением предложенных программных средств.

1. Нечеткий классификатор

Будем рассматривать классификатор с n входами (x_1, x_2, \dots, x_n) и одним выходом y (рис. 1), что соответствует отображению вида:

$$X = (x_1, x_2, \dots, x_n) \rightarrow y \in \{d_1, d_2, \dots, d_m\},$$

где d_1, d_2, \dots, d_m - классы (типы решений).

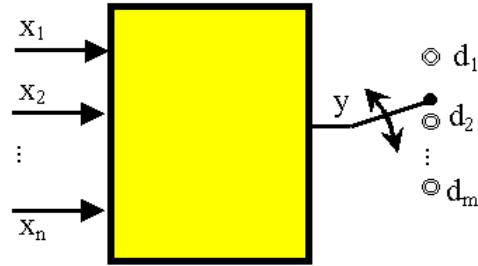


Рис. 1. Классификатор.

Классификация на основе нечеткого логического вывода осуществляется по следующей базе знаний [2, 3]:

ЕСЛИ $(x_1 = a_{1,j1})$ И $(x_2 = a_{2,j1})$ И ... И $(x_n = a_{n,j1})$ с весом w_{j1} ,
ИЛИ $(x_1 = a_{1,j2})$ И $(x_2 = a_{2,j2})$ И ... И $(x_n = a_{n,j2})$ с весом w_{j2} ,
 ...
ИЛИ $(x_1 = a_{1,jk_j})$ И $(x_2 = a_{2,jk_j})$ И ... И $(x_n = a_{n,jk_j})$ с весом w_{jk_j} ,
ТО $y = d_j$, $j = \overline{1, m}$, (1)

где $a_{i,jp}$ — нечеткий терм, которым оценивается переменная x_i в строчке с номером jp ($p = \overline{1, k_j}$), т. е. $a_{i,jp} = \int_{[x_i, \bar{x}_i]} \mu_{jp}(x_i)/x_i$; k_j — количество строчек-конъюнкций, в которых выход y оценивается значением d_j ; $w_{jp} \in [0, 1]$ — весовой коэффициент правила с номером jp .

Степени принадлежности объекта $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ классам d_j рассчитываются так [2, 3]:

$$\mu_{d_j}(X^*) = \bigvee_{p=1, k_j} w_{jp} \cdot \bigwedge_{i=1, n} (\mu_{jp}(x_i^*)), \quad j = \overline{1, m}, \quad (2)$$

где $\mu_{jp}(x_i^*)$ — степень принадлежности входа x_i^* нечеткому терму $a_{i,jp}$; \bigvee (\bigwedge) — s-норма (t-норма), которой в задачах классификации обычно соответствует максимум (минимум).

В качестве решения выбирают класс с максимальной степенью принадлежности:

$$y^* = \arg \max_{\{d_1, d_2, \dots, d_m\}} (m_{d_1}(X^*), m_{d_2}(X^*), \dots, m_{d_m}(X^*)). \quad (3)$$

Пример 1.1. Известна нечеткая база знаний:

Если $x_1 = \text{низкий}$	И	$x_2 = \text{низкий}$,	то $y = \text{класс 1}$;
Если $x_1 = \text{средний}$	И	$x_2 = \text{высокий}$,	то $y = \text{класс 2}$;
Если $x_1 = \text{высокий}$	И	$x_2 = \text{высокий}$,	то $y = \text{класс 3}$;
Если $x_1 = \text{высокий}$	И	$x_2 = \text{низкий}$,	то $y = \text{класс 2}$.

На рис. 2 приведены результаты классификации 600 объектов при реализации t-нормы операцией минимума и s-нормы операцией максимума. Области, соответствующие первому, второму, третьему и четвертому правилам базы знаний обозначены на рисунке символами #1, #2, #3 и #4.

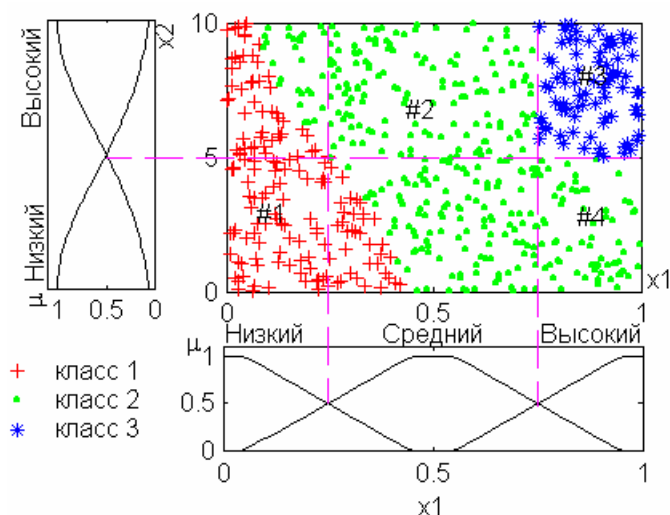


Рис. 2. Классификации по нечеткой базе знаний из примера 1.1

2. Постановки задач настройки нечеткого классификатора

Будем предполагать, что существует обучающая выборка из M пар экспериментальных данных, связывающих входы $X = (x_1, x_2, \dots, x_n)$ с выходом y исследуемой зависимости:

$$(X_r, y_r), \quad (r = \overline{1, M}), \quad (4)$$

где $X_r = (x_{r,1}, x_{r,2}, \dots, x_{r,n})$ — входной вектор в r -ой паре и y_r — соответствующий выход.

Настройка представляет собой нахождение таких параметров функций принадлежности термов входных переменных и весовых коэффициентов правил, которые минимизируют отклонение между желаемым и действительным поведением нечеткого классификатора на обучающей выборке. Критерий близости можно определить различными способами.

Первый способ заключается в выборе в качестве критерия настройки процента ошибок классификации на обучающей выборке. Введем следующие обозначения:

P — вектор параметров функций принадлежности термов входных и выходной переменных;

W — вектор весовых коэффициентов правил базы знаний;

$F(X_r, P, W)$ — результат вывода по нечеткой базе (1) с параметрами (P, W) при значении входов X_r . Нечеткий логический вывод осуществляется по формулам (2)–(3).

Тогда настройка нечеткого классификатора сводится к следующей задаче оптимизации: *найти такой вектор (P, W) , чтобы:*

$$\frac{1}{M} \sum_{r=1, M} \Delta_r \rightarrow \min, \quad (5)$$

где Δ_r — ошибка классификации объекта X_r :

$$\Delta_r = \begin{cases} 1, & \text{если } y_r \neq F(X_r, P, W) \\ 0, & \text{если } y_r = F(X_r, P, W) \end{cases}.$$

Преимущества критерия настройки (5) заключаются в его простоте и ясной содержательной интерпретации. Процент ошибок широко используется как критерий обучения различных систем распознавания образом. Целевая функция задачи оптимизации (5) принимает дискретные значения. Это затрудняет использование градиентных методов оптимизации, т. к. на протяженных плато целевой функции алгоритмы оптимизации «застревают». Особенно трудно подобрать подходящие параметры градиентных алгоритмов (например, приращения аргументов для расчета частных производных) при настройке нечеткого классификатора на небольшой выборке данных.

Второй способ использует в качестве критерия настройки расстояние между результатом вывода в виде нечеткого множества $\left(\frac{\mu_{d_1}(X)}{d_1}, \frac{\mu_{d_2}(X)}{d_2}, \dots, \frac{\mu_{d_m}(X)}{d_m} \right)$ и значением выходной переменной в обучающей выборке. Для этого выходную переменную y в обучающей выборке (4) фаззифицируют следующим образом [2, 3]:

$$\left. \begin{aligned} \tilde{y} &= (1/d_1, 0/d_2, \dots, 0/d_m), & \text{если } y = d_1 \\ \tilde{y} &= (0/d_1, 1/d_2, \dots, 0/d_m), & \text{если } y = d_2 \\ &\dots \\ \tilde{y} &= (0/d_1, 0/d_2, \dots, 1/d_m), & \text{если } y = d_m \end{aligned} \right\} \quad (6)$$

В этом случае настройка нечеткого классификатора сводится к следующей задаче оптимизации [2, 3]: *найти такой вектор (P, W) , чтобы:*

$$\frac{1}{M} \cdot \sum_{r=1}^M \sum_{j=1}^m \left(m_{d_j}(y_r) - m_{d_j}(X_r, P, W) \right)^2 \rightarrow \min, \quad (7)$$

где $m_{d_j}(y^r)$ — степень принадлежности значения выходной переменной y в r -ой пары обучающей выборке к решению d_j в соответствии с (6); $m_{d_j}(X_r, P, W)$ — степень принадлежности выхода нечеткой модели с параметрами (P, W) к решению d_j , определяемая по формуле (2) при значениях входов из r -ой пары обучающей выборке (X_r) .

Целевая функция в задаче (7) не имеет протяженных плато, поэтому

она пригодна к оптимизации градиентными методами. Однако, результаты оптимизации не всегда удовлетворительные: нечеткая база знаний, обеспечивающая минимум критерия (7), не всегда обеспечивает также и минимум ошибок классификации. Это объясняется тем, что точки, близкие к границам раздела классов, вносят почти одинаковый вклад в критерий настройки, как при правильной, так и при ошибочной классификации.

Третий способ наследует достоинства предыдущих способов. Идея заключается в том, чтобы вклад ошибочно классифицированных объектов в критерий настройки увеличивать, посредством умножением расстояния

$\sum_{j=1}^m (m_{d_j}(y_r) - m_{d_j}(X_r, P, W))^2$ на штрафной коэффициент. В результате задача оптимизации принимает следующий вид:

$$\frac{1}{M} \cdot \sum_{r=1}^M (\Delta_r \cdot \text{penalty} + 1) \cdot \sum_{j=1}^m (m_{d_j}(y_r) - m_{d_j}(X_r, P, W))^2 \rightarrow \min, \quad (8)$$

где $\text{penalty} > 0$ — штрафной коэффициент.

Задачи (5), (7) и (8) могут быть решены различными технологиями оптимизации, среди которых часто применяется метод наискорейшего спуска, квазиньютоновские методы и генетические алгоритмы. На управляемые переменные P обычно накладывают ограничения, обеспечивающие линейную упорядоченность элементов терм-множеств. Такие ограничения не позволяют алгоритмам оптимизации сделать, например, нечеткое множество «Низкий» больше «Высокого». Кроме того, ядра нечетких множеств не должны выходить за пределы диапазонов изменения соответствующих переменных. Это обеспечивает прозрачность, т. е. содержательную интерпретабельность нечеткой базы знаний после настройки. Что касается вектора W , то его координаты должны находиться в диапазоне $[0, 1]$. Если к уровню интерпретабельности базы знаний предъявляются высокие требования, то веса правил не настраивают, оставляя их равными 1. Возможен и промежуточный вариант, когда весовые коэффициенты могут принимать значения 0 и 1. В этом случае нулевое значения весового коэффициента эквивалентно исключению правила из нечеткой базы знаний.

Параметры функций принадлежности и веса правил можно настраивать одновременно или по отдельности. При настройке только весов правил объем вычислений можно значительно сократить, т. к. степени принадлежности $\mu_{jp}(x_i^*)$, входящие в (2), не зависят от W . Для этого в начале оптимизации надо рассчитать степени выполнения правил при единичных весовых коэффициентах ($w_{jp} = 1$) для каждого объекта из обучающей выборки:

$$g_{jp}(X_r) = \bigwedge_{i=1, n} \mu_{jp}(x_{r,i}), \quad j = \overline{1, m}, \quad p = \overline{1, k_j}, \quad r = \overline{1, M}.$$

Для новых весовых коэффициентов степени принадлежности объекта X_r классам d_j рассчитываются так:

$$\mu_{d_j}(X_r) = \bigvee_{p=1, k_j} w_{jp} \cdot g_{jp}(X_r), \quad j = \overline{1, m}. \quad (9)$$

3. Классификация на основе нечеткого вывода в системе MATLAB

Пакет **Fuzzy Logic Toolbox** [4] вычислительной системы MATLAB обеспечивает проектирование систем нечеткого логического вывода для объектов с непрерывным выходом, т. е. для случая, когда выходная переменная y может принимать значения из диапазона $[\underline{y}, \overline{y}]$. В настоящей разделе показывается, как расширить **Fuzzy Logic Toolbox** для выполнения классификации на основе нечеткого вывода.

В качестве нечеткого классификатора предлагается использовать систему нечеткого логического вывода типа Сугено. Классам решений $\{d_1, d_2, \dots, d_m\}$ ставятся в соответствие термы выходной переменной; наименование класса решений зададим как элемент терм-множества выходной переменной. Параметры заключений правил (параметры «функций принадлежности» выходной переменной) могут быть произвольными, т. к. они не влияют на результат классификации. Проектирование системы нечеткого вывода типа Сугено можно удобно осуществлять в редакторе **fuzzy** из **Fuzzy Logic Toolbox**.

Для выполнения нечеткой классификации нами разработана функция **fuzzy_classifier**, листинг которой приведен в Приложении 1. Функция нечеткой классификации вызывается в таком формате:

decision=fuzzy_classifier(X, fis, type),

где **X** — вектор информативных признаков объекта классификации; **fis** — система нечеткого логического вывода; **type** — тип возвращаемого функцией результата (допустимые значения: **'number'** — порядковый номер класса и **'name'** — имя класса. Значение по умолчанию — **'number'**); **decision** — результат классификации для объекта **X**.

Функция **fuzzy_classifier** вызывает функцию **evalfis** в формате

[a, b, c, d] = evalfis(x, fis),

что позволяет получить промежуточные результаты нечеткого логического вывода. Затем находятся правила с максимальной степенью выполнения. Если таких правил несколько, тогда среди конкурирующих классов выбирает тот, сумма степеней принадлежности которому максимальна.

Функция нечеткой классификации может вызываться с двумя выходными аргументами:

[decision, mf_grades]=fuzzy_classifier(X, fis, type),

где **mf_grades** — вектор степеней принадлежности объекта **X** классам решений.

4. Настройка нечеткого классификатора в системе MATLAB

Настройка нечеткого классификатора сводится к задачам оптимизации (5), (7) или (8). Для решения этих задач в системе MATLAB можно воспользоваться пакетом **Optimization Toolbox**. На основе программ обучения нечетких моделей типа Мамдани [5] нами разработаны типовые сценарии настройки нечетких классификаторов, а также типовые целевые функции. М-файлы этих программ приведены на сайте www.MATLAB.ru в разделе Fuzzy Logic Toolbox.

Для быстрой настройки весов правил нечеткого классификатора на основе соотношения (9) авторами написан пакет FALEFC (**FA**st **LE**arning the **F**uzzy **C**lassifier). Время настройки весов правил с помощью пакета FALEFC на порядок меньше, чем при настройке функций принадлежности. Пакет FALEFC использует функции **Fuzzy Logic Toolbox** и **Optimization Toolbox**. Пакет включает такие программы:

- fuzzy_classifier** - выполнение нечеткого вывода для задач классификации;
- fast_w_learning** - сценарий быстрой настройки весов правил нечеткого классификатора;
- dp_for_fuzzy_cl_learning** - считывание выборки данных и преобразование ее к формату, требуемого программами настройки нечеткого классификатора;
- rmg_fuzzy_cl** - расчет RMG - матрицы степеней выполнения правил при единичных весовых коэффициентах, необходимой для быстрой классификации;
- fuzzy_classifier_rmg** - быстрая классификация на основе нечеткого вывода при новых значениях весов правил с использованием RMG (матрицы степеней выполнения правил при единичных весовых коэффициентах);
- ob_fun_fast_w** - целевая функция для быстрой настройки весов нечетких правил, использующая RMG;
- fast_cl_testing_with_rmg** - быстрое тестирование нечеткого классификатора при новых весах правил с использованием RMG;
- rule_order_fuzzy_cl** - определение количества правил в базе знаний для каждого класса (типа решения).

Пакет FALEFC также доступен в разделе Fuzzy Logic Toolbox сайта www.MATLAB.ru.

5. Примеры проектирования нечетких классификаторов в системе MATLAB

Пример 5.1. Рассматривается задача классификации ирисов, предложенная Фишером в 1936 г. Задача состоит в отнесении ириса к одному из трех классов: 1 - *Iris Setosa*, 2 - *Iris Versicolor*; 3 - *Iris Virginica*. При классификации используются следующие признаки цветков: x_1 - длина чашелистика; x_2 - ширина чашелистика; x_3 - длина лепестка; x_4 - ширина лепестка. Исходные данные для классификации ирисов записаны в файле `iris.dat`, входящего в **Fuzzy Logic Toolbox**. Файл содержит 150 строк, каждая из которых описывает один ирис. Информация о цветке представлена пятеркой чисел - первые четыре числа соответствуют значениям признаков, а пятое — классу ириса. Двумерные распределения ирисов показаны на рис. 3.

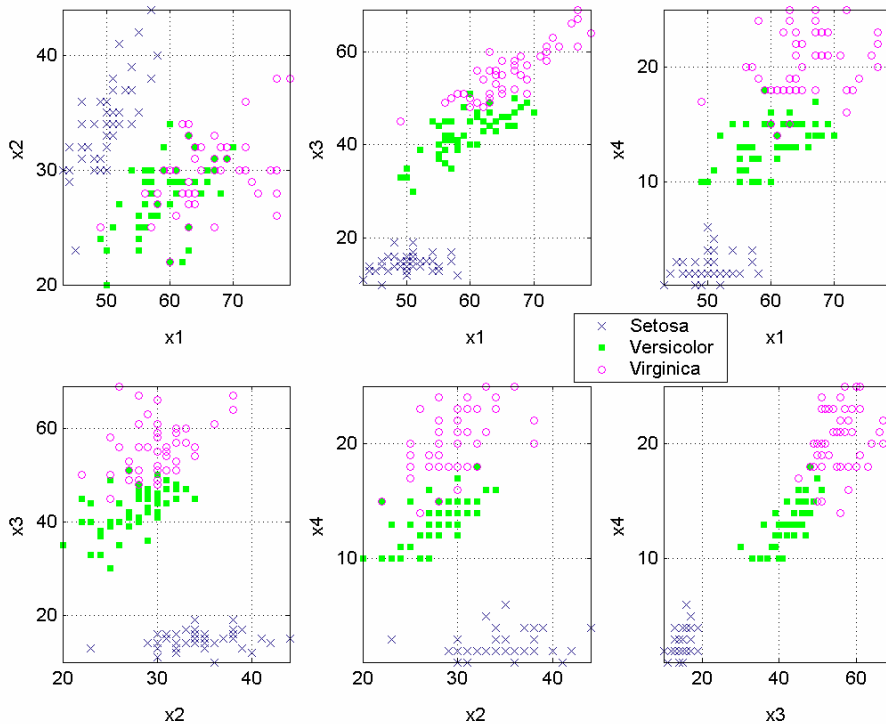


Рис. 3. 2D-распределения фишеровских ирисов.

В редакторе **fuzzy** создадим систему нечеткого логического вывода типа Сугено с четырьмя входными и одной выходной переменными. Диапазоны изменения входных переменных установим такими же, как и для исходных данных: $x_1 \in [47, 79]$; $x_2 \in [20, 44]$; $x_3 \in [10, 69]$; $x_4 \in [1, 25]$. Для лингвистической оценки признаков цветков будем использовать термы «низкий», «средний» и «высокий» с установленными по умолчанию треугольными функциями принадлежности. Взаимосвязь «входы - выход» опишем тремя нечеткими правилами:

если x_4 = 'низкий',

то y = '*Iris Setosa*' ;

если $x_3 = \text{'средний'}$ и $x_4 = \text{'средний'}$, то $y = \text{'Iris Versicolor'}$;
 если $x_3 = \text{'высокий'}$ и $x_4 = \text{'высокий'}$, то $y = \text{'Iris Virginica'}$.

Логической операций «И» в посылках правил поставим в соответствие операцию минимума над функциями принадлежности. Для этого в редакторе **fuzzy** выберем опцию **min** в меню **And method**.

Применяя функцию **fuzzy_classifier** обнаруживаем, что созданная нечеткая модель правильно классифицирует 130 из 150 ирисов. Для настройки весов правил сформируем обучающую и тестирующую выборки. В обучающую выборку включим 120 ирисов с порядковыми номерами не кратными 5. В связи с небольшим количеством данных тестировать нечеткий классификатор будем на всей выборке. Для быстрой настройки весов правил нечеткого классификатора в типовом сценарии **fast_w_learning** укажем имена файлов системы нечеткого вывода **iris_cl.fis** и экспериментальных данных **iris.dat**. Сценарий настройки нечеткого классификатора ирисов приведен в Приложении 2.

В результате настройки получены такие весовые коэффициенты: 0.9 — для первого правила; 0.1 — для второго правила; 0.8 — для третьего правила. Настроенная система нечеткого логического вывода правильно классифицирует 116 ирисов (96.67%) из обучающей выборки и 144 (96%) ирисов из всей выборки цветков.

Пример 5.2. Рассматривается объект с двумя входами $x_1, x_2 \in [0, 10]$ и одним выходом y , который может принимать одно из трех дискретных значений $\{d_1, d_2, d_3\}$ в соответствии с решающими правилами:

$$y = \begin{cases} d_1, & \text{если } x_2 < \frac{14.6}{2.25 + (x_1 - 6.5)^2} \\ d_2, & \text{если } \frac{14.6}{2.25 + (x_1 - 6.5)^2} < x_2 < 2.2 \cdot \sqrt{x_1} + 3 \\ d_3, & \text{если } x_2 > 2.2 \cdot \sqrt{x_1} + 3 \end{cases} \quad (9)$$

$$(10)$$

Обучающие и тестирующие выборки показаны на рис. 4. Они содержат 80 и 5000 пар «входы — выход», соответственно. В выборках значения входов выбирались случайно, а значения выхода рассчитывались по (10). Файлы данных доступны в разделе **Fuzzy Logic Toolbox** сайта www.MATLAB.ru.

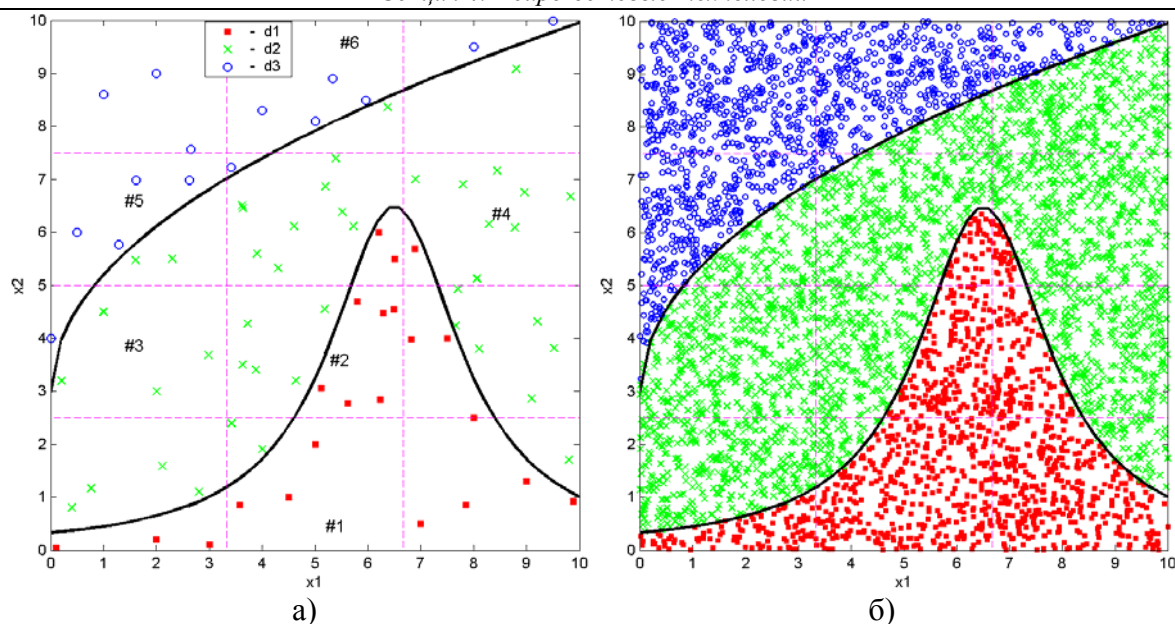


Рис. 4. Обучающая (а) и тестирующая (б) выборки.

На основе этих выборок спроектируем нечеткий классификатор. Входы нечеткого классификатора будем рассматривать как лингвистические переменные, значения которых определяются из следующих термножеств: {«low», «average», «high»} для x_1 , и {«low», «lower average», «higher average», «high»} для x_2 . Формализацию термов осуществим симметричной гауссовской функции принадлежности:

$$\mu(x) = e^{-\frac{(x-h)^2}{2c^2}},$$

где x — элемент универсального множества; h и c — параметры функции принадлежности - координата максимума коэффициент концентрации.

До настройки коэффициенты концентраций всех функций принадлежности равны 2. Координаты максимумов выбирались так, чтобы разбить интервал $[0, 10]$ на три (для x_1) и на четыре (для x_2) равные части (рис. 5а). По рис. 4а эксперт сгенерировал шесть нечетких правил классификации (табл. 1).

Таблица 1.
К примеру 5.2: нечеткая база знаний.

x_1	x_2	y	w (до настройки)	w (классификатор I)	w (классификатор II)	w (классификатор III)
average	low	d_1	1	0.62	0.75	0.71
average	below average	d_1	1	0.41	0.39	0.49
low	below average	d_2	1	0.81	1	0.90
high	higher average	d_2	1	0.46	1	0.71
low	higher average	d_3	1	0.66	0.49	0.65
average	high	d_3	1	0.02	0.02	0.91

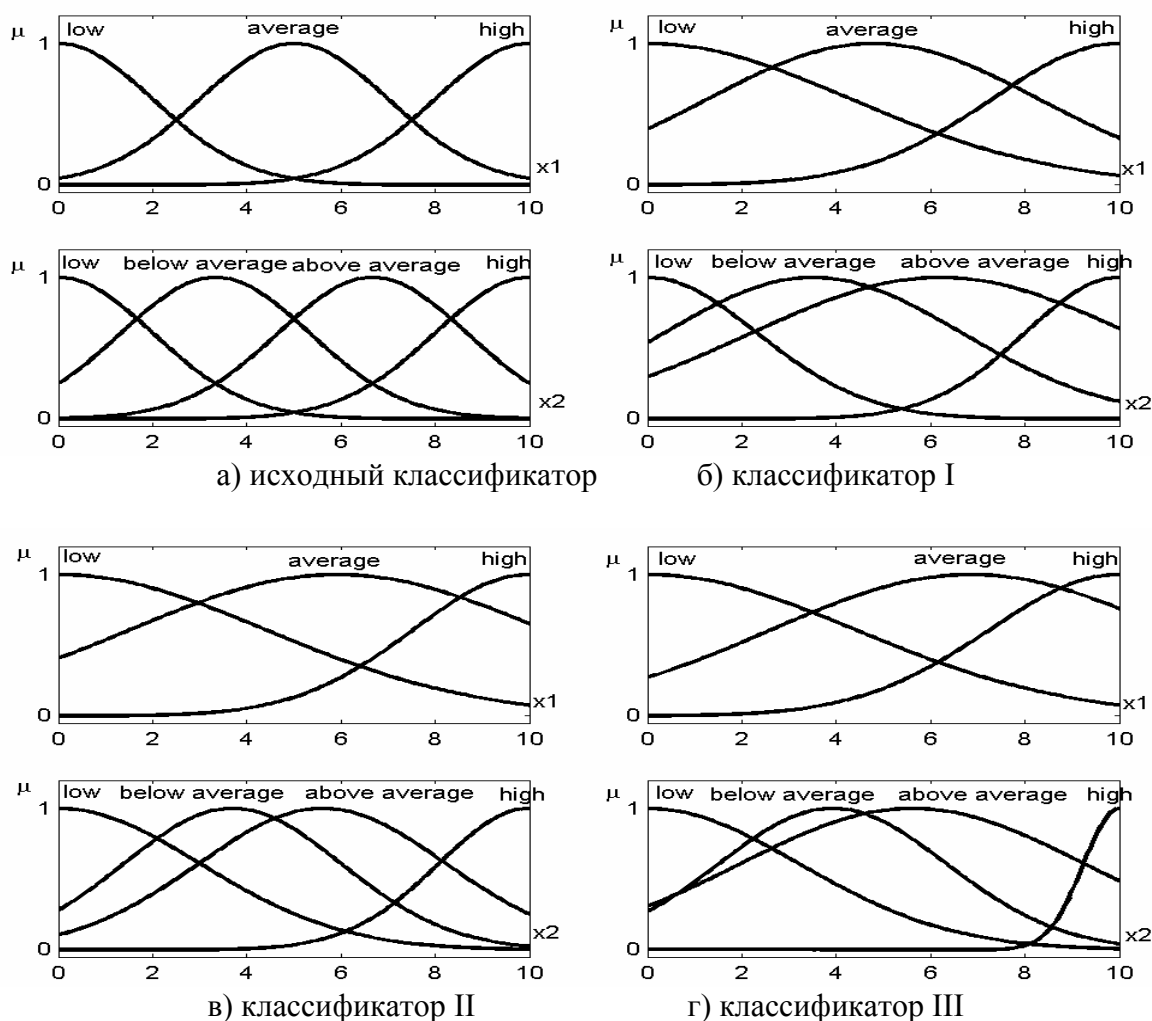


Рис. 5. Функции принадлежности.

Исходная нечеткая база знаний с шестью правилами грубо отражает нелинейные разделяющие кривые - на тестирующей выборке ошибочно классифицировано 26.6% объектов (рис. 6а). После настройки весов правил количество ошибок уменьшилось до 15%, однако безошибочность нечеткого классификатора остается на уровне следующего простого дерева решений:

Если $((x_1 > 1.2929) \& (x_2 \leq 1))$	то $y = d_1$,
Если $((x_1 > 4.6335) \& (x_2 > 1) \& (x_1 \leq 7.5) \& (x_2 \leq 6))$	то $y = d_1$,
Если $((x_2 > 8.3607) \& (x_1 > 5.3301))$	то $y = d_3$,
Если $((x_1 \leq 5.3301) \& (x_2 > 6.9107))$	то $y = d_3$,
Если $((x_1 \leq 1.2929) \& (x_2 > 3.4988) \& (x_2 \leq 6.9107))$	то $y = d_3$,
Иначе,	$y = d_2$.

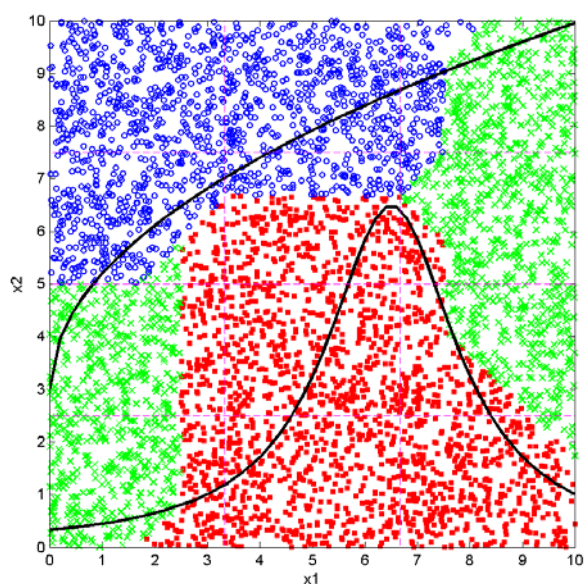
Низкая безошибочность нечеткого классификатора после настройки весов правил объясняется «плохими» функциями принадлежности. Поэтому, необходимо модифицировать не только веса правил, но и функции принадлежности. Будем настраивать следующие 16 параметров нечеткого классификатора:

- 3 координаты максимумов функций принадлежности термов «average», «lower average» и «higher average»;
- 7 коэффициентов концентраций функций принадлежности термов входных переменных;
- 6 весовых коэффициентов правил базы знаний.

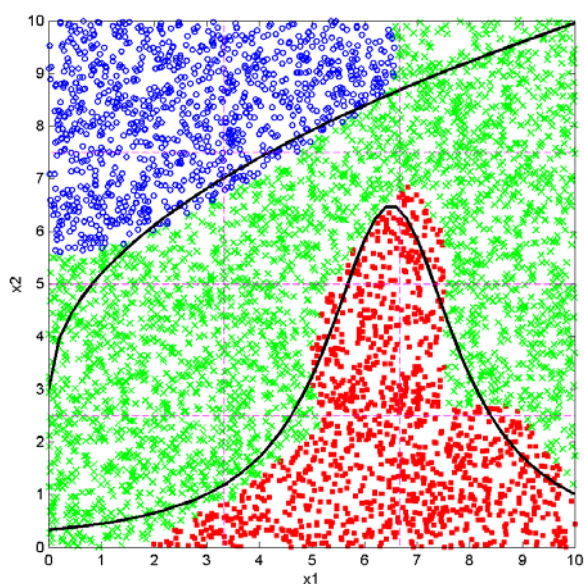
Результаты настройки с использованием различных критериев приведены на рис. 5 и в табл. 1. Использовались такие критерии настройки: критерий I — формула (5); критерий II — формула (7); критерий III - формула (8). Результаты тестирования классификаторов сведены в табл. 2. Классификатор I настроен по критерию I, классификатор II — по критерию II и классификатор III - по критерию III.

Таблица 2.
Результаты тестирования классификаторов.

Классификатор	Критерий I	Критерий II	Критерий III	Безошибочность на тестирующей выборке
Исходный	33.75%	0.433	3.99	26.6%
Классификатор I	6.25%	0.481	1.08	9.78%
Классификатор II	18.75%	0.417	1.49	16.42%
Классификатор III	6.25%	0.416	0.98	9.28%
Дерево решений	7.5%	-	-	15.24%



а) исходный классификатор



б) классификатор I

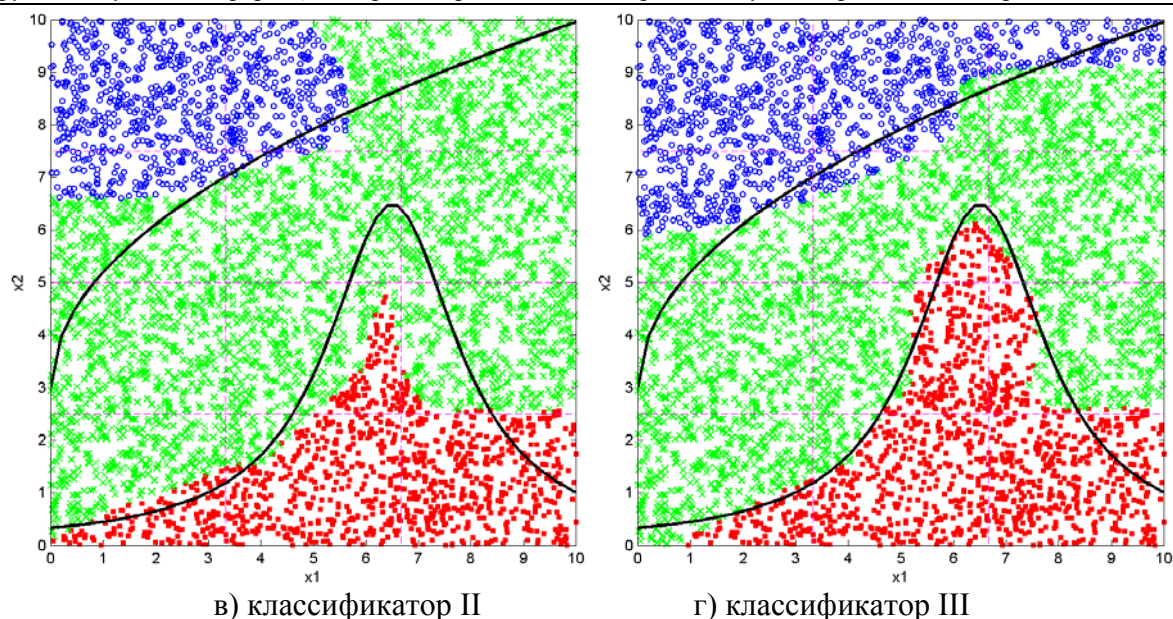


Рис. 6. Классификация на тестирующей выборке.

Выводы

В статье показано как расширить **Fuzzy Logic Toolbox** для проектирования нечетких классификаторов, т. е. систем нечеткого логического вывода с дискретным выходом. Проанализированы критерии настройки нечеткого классификатора. Предложен пакет программ FALEFC для быстрой настройки весов правил нечеткого классификатора. Приведенные примеры проектирования нечетких классификаторов подтверждают эффективность предложенных программ расширения пакета **Fuzzy Logic Toolbox**.

Литература

1. *Kasabov N.* Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering. The MIT Press.— 1996.— 550 p.
2. *Rotshtein A.* Desig and Tuning of Fuzzy Rule-Based System for Medical Diagnosis. In Fuzzy and Neuro-Fuzzy Systems in Medicine (Teodorescu N.H. (ed.)). CRC-Press, 1998.— P.243–289.
3. *Ротштейн А. П.* Интеллектуальные технологии идентификации: нечеткая логика, генетические алгоритмы, нейронные сети.— Винница: УНІВЕРСУМ-Вінниця, 1999.— 320 с.
4. Fuzzy Logic Toolbox. User's Guide, Version 2.— The MathWorks, Inc., 2001.
5. *Штовба С. Д.* Идентификация нелинейных зависимостей с помощью нечеткого логического вывода в системе MATLAB // Exponenta Pro. Математика в приложениях.— 2003.— №2.— С.9–15.

Приложение 1. Листинг функции *fuzzy_classifier*

```

function [decision, mf_grades]=fuzzy_classifier(x, fis, type)
%FUZZY_CLASSIFIER выполняет нечеткий логический вывод для задач
%      классификации.
% x - входной вектор (признаки классифицируемого объекта);
% fis - система нечеткого логического вывода;
% type - формат решения:
%'number' - порядковый номер класса (значение по умолчанию);
%'name' - наименование класса;
% decision - результат классификации объекта x;
% mf_grades - вектор степеней принадлежности каждому классу.
%Требуемые программные средства: Fuzzy Logic Toolbox v.2.X

%Serhiy D. Shtovba shtovba@ksu.vstu.vinnica.ua
%Vinnitsa State Technical University
%$Revision: 1.6 $ $Date: 2004/03/12

%Проверка входных аргументов:
if nargin<2
    error('Необходимо задать 2 или 3 входных аргумента');
end
[tmp1 tmp2]=size(x);
if tmp1~=1
    error('Классификация выполняется только одного объекта');
end
if isfis(fis)==0
    error('Второй аргумент должен быть системой нечеткого вывода');
end
if fis.type(1:6)~='sugeno'
    error('Система нечеткого вывода должна быть типа Сугено');
end
if nargin==2
    type='number';      % <---- Установка значения по умолчанию
end

number_of_decisions=length(fis.output(1).mf);
number_of_rules=length(fis.rule);
[a,b,c,d]=evalfis(x,fis); % <---- Нечеткий логический вывод
%Номера правил с максимальной степенью выполнения:
rule_num_with_max_fulfilment=find(d==max(d));
%количество таких правил:
number_rules_with_max=length(rule_num_with_max_fulfilment);
if number_rules_with_max>1
    %Если таких правил несколько, то возможно объект попал на границу классов:
    colision_flag=zeros(1,number_of_decisions);
    for i=1:number_rules_with_max
        index=rule_num_with_max_fulfilment(i);
        %Помечаем решения с максимальной степенью принадлежности:
        colision_flag (fis.rule(index).consequent )=1;
    end
end

```



```
end
%Суммируем степени принадлежности проблемного объекта по разным прави-
лам:
sum_mf=zeros(1,number_of_decisions);
for i=1:number_of_rules
    index=fis.rule(i).consequent;
    sum_mf(index)=sum_mf(index)+d(i);
end
%оставляем классы с максимальной степенью принадлежности (через *)
%и выбираем класс по максимальной сумме степеней принадлежности:
[tmp1 number_of_the_class]=max(sum_mf.*colision_flag);
else number_of_the_class=fis.rule(rule_num_with_max_fulfilment).consequent;
end
%Возврат результата классификации в требуемом формате:
switch type
    case 'number', decision=number_of_the_class;
    case 'name', decision=fis.output.mf(number_of_the_class).name;
    otherwise, error('Допустимые значения для 3-го аргумента: number или name')
end
if nargin==2
    mf_grades(1:number_of_decisions)=0;
    for i=1:number_of_rules
        index=fis.rule(i).consequent; %номер класса в i-м правиле
        %Объединим через операцию максимума степени принадлежности одному и
        %тому же классу по различным правилам:
        mf_grades(index)=max(mf_grades(index), d(i));
    end
end
end
```

Приложение 2. Листинг сценария настройки нечеткого классификатора ирисов

```
%Настройки нечеткого классификатора ирисов
%Требуемые программные средства: Fuzzy Logic Toolbox v.2.X
% Optimization Toolbox v.2.X
% FALEFC v.1.0
%Serhiy D. Shtovba shtovba@ksu.vstu.vinnica.ua
%Vinnitsa State Technical University
%$Revision: 1.2 $ $Date: 2004/03/14

%Загрузка исходной нечеткой модели:
fis=readfis('iris_сд.fis');
RULE_order=rule_order_fuz_cl(fis);
%Загрузка обучающей выборки:
[INP OUT_c OUT_mu]=dp_for_fuzzy_cl_learning(fis, 'iris.dat');
%Расчет RMG:
num_rule=length(fis.rule);
for i=1:num_rule
    fis.rule(i).weight=1;
end
```

```

RMG=rmg_fuzzy_cl(INP, fis);
%НАСТРАИВАЕМЫЕ
ПАРАМЕТРЫ
vlb_w(1:num_rule)=0;           %нижняя граница
w0(1:num_rule)=rand(1, num_rule); %начальная точка
vub_w(1:num_rule)=1;           %верхняя граница
%-----
%ПАРАМЕТРЫ ОПТИМИЗАЦИИ:
options=[];
options=optimset('Display', 'iter');
options.DiffMinChange=0.1;
options.DiffMaxChange=0.25;
options.LargeScale='off';
options.MaxIter=20;
options.MaxFunEvals=150;
%Формирование обучающей выборки:
ii=1;
for i=1:150
    if rem(i,5)~=0
        tr_index(ii)=i; ii=ii+1;
    end
end
%Оптимизация:
[wopt, delta]=fmincon(@ob_fun_fast_w, w0, [], [], [], [], vlb_w, ...
    vub_w, [], options, RULE_order, RMG(tr_index,:), OUT_c(tr_index,:), ...
    3, OUT_mu(tr_index,:))
%Тестирование по трем критериям на обучающей выборке:
[decision, delta_m, delta_s, delta_p] =fast_cl_testing_with_rmg(wopt, ...
    RULE_order, RMG(tr_index,:), OUT_c(tr_index,:), OUT_mu(tr_index,:));
%Тестирование по трем критериям на всей выборке:
[decision, delta_m, delta_s, delta_p] =fast_cl_testing_with_rmg(wopt, ...
    RULE_order, RMG, OUT_c, OUT_mu);
%Запись оптимальных весов в систему нечеткого вывода:
for i=1:num_rule
    fis.rule(i).weight=wopt(i);
end

```