

УДК 004.93.14

ОПТИМИЗАЦИЯ НЕЧЕТКИХ НЕЙРОННЫХ КЛАССИФИКАТОРОВ С ПОМОЩЬЮ МОДЕЛЬНО-ОРИЕНТИРОВАННОГО КЛАСТЕРНОГО АНАЛИЗА

А.Г. Натекин

Департамент Корпоративных Технологий, ООО "Сименс"
Россия, 191186, Санкт-Петербург, Волынский пер., 3
E-mail: alexey.natekin.ext@siemens.com

М.А. Калинин

Департамент Корпоративных Технологий, ООО "Сименс"
Россия, 191186, Санкт-Петербург, Волынский пер., 3
E-mail: mikhail.kalinkin@siemens.com

Ключевые слова: нечеткие производственные модели, нейронные сети, кластеризация, EM-алгоритм, машинное обучение, классификация, прокат стали.

Нейросетевое представление нечетких производственных моделей успешно используются в различных задачах машинного обучения. Основная трудность применения данных моделей заключается в процедуре инициализации баз нечетких производственных правил. Рассматривается способ инициализации баз правил нечетких производственных моделей с помощью EM-алгоритма кластеризации. Полученные модели используются для задания нечетких производственных моделей с многомерными функциями принадлежности. Предложен способ нейросетевого представления полученных моделей, эффективность которого проверена как на модельных данных, так и на реальных данных из области проката стали.

OPTIMIZATION OF FUZZY NEURAL CLASSIFIERS BY USE OF THE MODEL-ORIENTED CLUSTER ANALYSIS

A.G. Natekin

Department of Corporative Technologies, "Siemens" Ltd
Russia, 191186, Saint Petersburg, Volynsky lane, 3
E-mail: alexey.natekin.ext@siemens.com

M.A. Kalinkin

Department of Corporative Technologies, "Siemens" Ltd
Russia, 191186, Saint Petersburg, Volynsky lane, 3
E-mail: mikhail.kalinkin@siemens.com

Key words: fuzzy production models, neural networks, clustering, EM-algorithm, machine learning, classification, steel rolling.

Neural network representation of fuzzy production models is successfully used in various problems of the machine learning. The main difficulty of applying these models is the procedure of initializing the bases of fuzzy production rules. A way of initializing the rule bases of fuzzy production models by use of the EM-algorithm of clustering is considered. The models obtained are used for setting fuzzy production models with multivariate membership functions. A way of neural network representation of the obtained models is proposed, whose efficiency has been verified by use of both modeled and real-world (from the steel rolling branch) data.

1. Введение

Системы нечеткого вывода и модели, построенные с использованием теории нечеткой логики, применяются во многих прикладных задачах, таких как машинное обучение [1, 2], моделирование сложных систем [3], распознавание образов [4, 5]. Полученные в результате моделирования нечеткие производственные модели являются моделями "серого ящика": информация представляется в виде правил, доступных для трактования экспертами.

При синтезе данных систем, важнейшей задачей является корректная идентификация базы нечетких правил, используемых в системе. После того, как система инициализирована, ее параметры требуется оптимизировать. Наиболее эффективным способом оптимизации параметров систем нечеткого вывода является представление данных систем в виде искусственных нейронных сетей прямого распространения сигнала [6, 3].

Ранее были предложены различные подходы к определению числа правил с помощью кластерного анализа [7, 8, 9, 10, 11]. В данной работе рассматривается метод инициализации базы правил с помощью EM-алгоритма с использованием Байесовского информационного критерия для определения необходимого числа кластеров. Приводится обобщение нечетких правил на случай многомерных функций принадлежности, а также предложено нейросетевое представление данных моделей. Метод тестируется как на модельных данных, так и на реальных данных из области проката стали.

Статья написана по следующему плану. В разделе 2 дается описание используемых нечетких производственных моделей. В разделе 3 рассматривается задача инициализации базы правил. В разделе 4 описывается представление нечетких производственных моделей в виде нейронных сетей и предлагается способ оптимизации их параметров. В разделе 5 предложенный метод тестируется на модельных и на реальных данных.

2. Нечеткие продукционные модели

2.1. Нечеткие продукционные модели

Нечеткая продукционная модель представляет из себя набор согласованных друг с другом правил, истинность каждого из которых определяется с помощью методов нечеткого вывода. Нечеткие продукционные правила имеют вид:

ЕСЛИ A , ТО B , где

A - предпосылка (антецедент) правила;

B - заключение (консеквент) правила.

Данное правило предназначено для определения истинности заключения на основе предпосылки A . Выбрав способ (схему) нечеткого вывода, для практического применения нечетких продукционных моделей необходимо задать базу нечетких продукционных правил.

2.2. Задание базы нечетких правил

Существуют различные способы задания структуры базы нечетких продукционных правил [3, 12, 10, 6, 9]. Простейший способ задания и инициализации базы правил [3] состоит в разбиении пространств входных переменных на решетку гиперкубов, задающих нечеткие множества. Для каждой входной переменной x_j , соответствующая ей область определения X_j , $j = \overline{1..m}$, делится на несколько одинаковых по размеру интервалов. При этом, правила формируются таким образом, чтобы получить все возможные комбинации нечетких множеств с соответствующими им функциями принадлежности всех входных переменных. Таким образом, число нечетких продукционных правил, необходимых для покрытия пространства входных переменных, определяется следующим образом:

$$N = \prod_{j=1}^m K_j$$

где m - число входных переменных;

K_j - число интервалов, на которые делится область определения j -ой входной переменной X_j , $j = \overline{1..m}$.

Очевидным недостатком данного подхода являются показательный рост числа правил при росте числа входных переменных. Также, границы такого разбиения ограничены прямоугольной сеткой, заданной нечеткими множествами отдельных входных переменных.

Наиболее распространенный на практике способ решения данной проблемы заключается в группировке одномерных функций принадлежности, соответствующих нечетким множествам на множествах значений каждой входной переменной. Это позволяет существенно сократить число правил по сравнению с предыдущим методом. Но так как входные переменные не всегда являются взаимно независимыми, то границы между задаваемыми областями могут быть произвольно изогнуты.

Чтобы избавиться от данного недостатка, можно использовать разбиение на многомерные нечеткие множества с представлением входных переменных в виде вектора

$x = [x_1, \dots, x_m]$, а предпосылок в виде соответствующих многомерных функций принадлежности $\mu_{A_i}(x)$, заданных на множестве определения x :

Π_i : **ЕСЛИ** x есть A_i **ТО** ... , где

$x = [x_1, \dots, x_m]^T$ - вектор входных переменных, $x \in \mathbb{R}^m$;

A_i - нечеткое множество с многомерной функцией принадлежности $\mu_{A_i}(x) \in [0, 1]$, $i = \overline{1..G}$;

G - число нечетких продукционных правил в базе.

Данный подход позволяет еще значительно уменьшить число используемых правил, одновременно повысив точность нечеткой продукционной модели и обобщающие способности правил. В данной работе будет рассматриваться разбиение на многомерные нечеткие множества с многомерными функциями принадлежности. Примеры описанных способов разбиений пространств входных переменных представлены на Рис.1. Красным и зеленым цветами на рисунке выделены две области в данных, а черным цветом показаны значения принадлежности соответствующим нечетким множествам.

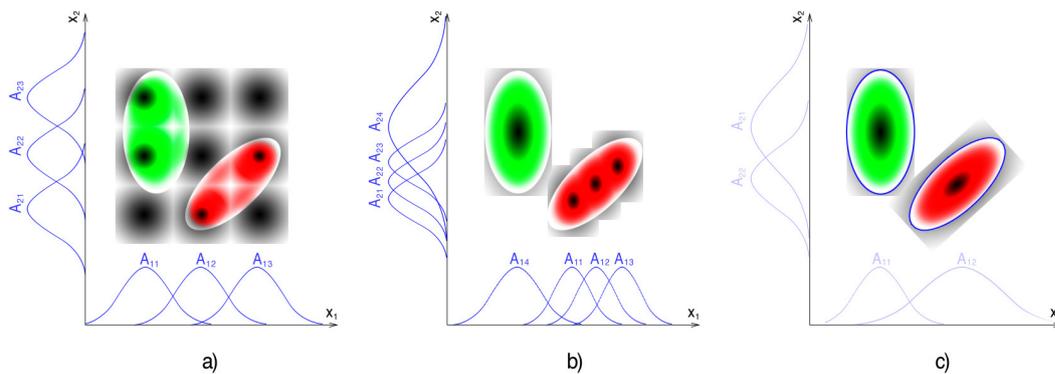


Рис. 1. Способы разбиения пространств входных переменных для инициализации базы нечетких правил

- a) Разбиение каждого из пространств входных переменных на одинаковое число равноотстоящих нечетких множеств. Данная база правил будет состоять из 9 правил вида:

Π_{ij} : **ЕСЛИ** x_1 есть A_{1i} **И** x_2 есть A_{2j} **ТО** ... , $i = \overline{1..3}$, $j = \overline{1..3}$.

- b) Разбиение каждого из пространств входных переменных путем группировки одномерных функций принадлежности. Данная база правил будет состоять из 4 правил вида:

Π_i : **ЕСЛИ** x_1 есть A_{1i} **И** x_2 есть A_{2i} **ТО** ... , $i = \overline{1..4}$.

- c) Разбиение пространства входных переменных на нечеткие множества с многомерными функциями принадлежности. Данная база правил будет состоять из 2 правил вида:

Π_i : **ЕСЛИ** x есть A_i **ТО** ... , $x = [x_1, x_2]^T$, $i = \overline{1..2}$.

2.3. Этапы нечеткого вывода

После определения базы нечетких продукционных правил, необходимо задать процедуру нечеткого вывода заключений правил. Нечеткий вывод состоит из ряда последовательно выполняющихся этапов, каждый из которых может быть определен различными способами, тем самым предоставляя исследователю богатый перечень структур моделей. Рассмотрим основные этапы нечеткого вывода, используемые для задачи классификации.

2.3.1. Введение нечеткости. Под введением нечеткости (фаззификацией) понимается процедура получения значений функций принадлежности нечетких множеств A_{ij} по значениям всех соответствующих входных переменных x_j , $j = \overline{1..m}$ для каждой из предпосылок каждого из нечетких правил Π_i , $i = \overline{1..G}$, в базе.

2.3.2. Агрегирование степеней истинности предпосылок правил. Если нечеткое правило содержит несколько предпосылок, то после введения нечеткости, вычисляются значения функций принадлежности по каждой из этих предпосылок:

$$\mu_{A_{i1}}(x'_1), \dots, \mu_{A_{ij}}(x'_j), \dots, \mu_{A_{im}}(x'_m), i = \overline{1..G}, j = \overline{1..m}.$$

Для того чтобы определить общую истинность всей предпосылки правила, значения $\mu_{A_{ij}}$ агрегируются, в зависимости от логических связей, используемых в каждом правиле.

Так, при использовании связки "И" между значениями функций принадлежности предпосылок правил $\mu_{A_{ij}}(x'_j)$ и $\mu_{A_{ik}}(x'_k)$, в нечетких продукционных моделях применяется оператор треугольной T-нормы [3, 12]. Мы будем в качестве T-нормы использовать алгебраическое произведение степеней истинности предпосылок правил:

$$(1) \quad \mu_{A_{ij} \cdot A_{ik}}(x) = \mu_{A_{ij}}(x'_j) \cdot \mu_{A_{ik}}(x'_k)$$

В результате данной процедуры определяется агрегированная степень истинности по всем предпосылкам для каждого правила: α_i , $i = \overline{1..G}$.

2.4. Модель нечетких продукционных правил нечеткого классификатора

Мы будем рассматривать правила, в которых предпосылки являются нечеткими высказываниями, а заключения - меткой класса. Рассмотрим нечеткие правила следующего вида:

$$(2) \quad \begin{array}{l} \Pi_i : \text{ЕСЛИ } x_1 \text{ есть } A_{i1} \text{ И } \dots \text{ И } x_m \text{ есть } A_{im} \\ \text{ТО } y \text{ есть } \mathbf{Class}_{c_i}, i = \overline{1..G}, j = \overline{1..m}, c_i \in [1..C], \text{ где} \end{array}$$

x_j - входные переменные, $x_j \in \mathbb{R}$, $j = \overline{1..m}$;

y - выходная переменная, задающая метку класса \mathbf{Class}_{c_i} , $c_i \in [1..C]$;

G - число правил в базе;

C - число классов;

A_{ij} - нечеткие множества, определенные для переменной x_j с функциями принадлежности $\mu_{A_{ij}}(x_j) \in [0, 1]$, $i = \overline{1..G}$, $j = \overline{1..m}$;

Для данной модели правил (2) используются следующие положения:

- функции принадлежности всех нечетких множеств являются Гауссовскими функциями

$$(3) \quad \mu_{A_{ij}}(x_j) = e^{-\frac{(x_j - a_{ij})^2}{2b_{ij}^2}}, \quad i = \overline{1..G}, j = \overline{1..m};$$

- T -норма - произведение функций принадлежности;
- декартово произведение нечетких множеств задано произведением функций принадлежности;
- приведение к четкости - $\arg \max$ класса, агрегированное значение принадлежности правила которого максимально:

$$\hat{y} = \text{Class}_{result}, \quad result = \arg \max_{i \in \overline{1..G}} \alpha_i.$$

Исходя из этого описания, функциональная модель нечеткого классификатора будет иметь следующий вид:

$$(4) \quad \hat{y} = \arg \max_{i \in \overline{1..G}} \left(\prod_{j=1}^m \mu_{A_{ij}}(x_j) \right).$$

2.4.1. Модель правил нечеткого классификатора с многомерными функциями принадлежности. Заметим, что для i -го правила модели нечеткого классификатора (2) выполняется:

$$(5) \quad \prod_{j=1}^m \mu_{A_{ij}}(x_j) = \prod_{j=1}^m e^{-\frac{(x_j - a_{ij})^2}{2b_{ij}^2}} = e^{-\frac{1}{2}(x-a_i)^T \begin{pmatrix} \frac{1}{b_{i1}} & 0 & \dots & 0 \\ 0 & \frac{1}{b_{i2}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{b_{im}} \end{pmatrix} (x-a_i)} = e^{-\frac{1}{2}(x-a_i)^T \Sigma_i^{-1} (x-a_i)} = \mu_{\hat{A}_i}(x), \quad \text{где}$$

- $\mu_{\hat{A}_i}(x)$ - многомерная функция принадлежности i -го правила, соответствующая ненормированной плотности многомерного нормального распределения $\mathcal{N}(a_i, \Sigma_i)$, $i = \overline{1..G}$. $\mu_{\hat{A}_i}(x) \in [0, 1]$;

- $x = [x_1, \dots, x_m]^T$ - вектор входных переменных;

- $a_i = [a_{i1}, \dots, a_{im}]^T$ - вектор средних значений многомерной функции принадлежности i -го правила, $i = \overline{1..G}$;

- $\Sigma_i = \begin{pmatrix} b_{i1} & 0 & \dots & 0 \\ 0 & b_{i2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & b_{im} \end{pmatrix}$ - ковариационная матрица многомерной функции принадлежности i -го правила, $i = \overline{1..G}$.

Таким образом, выражение (4) примет вид:

$$(6) \quad \hat{y} = \arg \max_{i \in 1..G} (\mu_{\hat{A}_i}(x)).$$

Заметим, что многомерная функция принадлежности имеет диагональную ковариационную матрицу. Таким образом, делается предположение о некоррелированности входных переменных, что эквивалентно заданию данным правилом в пространстве входных переменных эллиптической области без учета поворотов.

С одной стороны, такое упрощение модели обусловлено тем, что при росте размерности решаемой задачи, число параметров ковариационной матрицы будет расти квадратично. С другой стороны, в случае, когда истинная область во входных переменных повернута, для того чтобы корректно представить ее, потребуется более одной области без поворотов. На Рис.1 изображена ситуация, когда одну и ту же область можно представить 3 многомерными функциями принадлежности без поворотов, или одной с учетом поворотов.

В данной работе, исследуются многомерные функции принадлежности с полными ковариационными матрицами. Под полной ковариационной матрицей мы будем подразумевать произвольную ковариационную матрицу без каких-либо структурных ограничений, то есть, прежде всего без ограничений на некоррелированность входных переменных.

Помимо введения многомерных функций принадлежности с произвольными ковариационными матрицами, внесем в модель нечеткого классификатора еще одно дополнение. В отличие от моделей гауссовских смесей [13, 14, 8], ранее в моделях (4),(6) не учитывались объемы нормальных распределений, задающих многомерные функции принадлежности (5). Нормировка на объем нужна для контрастирования эллиптических областей меньшего объема, находящихся внутри больших областей. Таким образом, модель с учетом объемов выглядит следующим образом:

$$(7) \quad \hat{y} = \arg \max_{i \in 1..G} \left(\frac{\mu_{\hat{A}_i}(x)}{V_i} \right),$$

$$(8) \quad V_i = (2\pi)^{\frac{m}{2}} |\Sigma_i|^{\frac{1}{2}}, i = \overline{1..G}.$$

3. Кластерный анализ

Одним из наиболее эффективных способов инициализации базы правил нечетких продукционных моделей является кластерный анализ [9, 8]. Под кластерным анализом мы подразумеваем процедуру разбиения данных на структурированные подгруппы, называемые кластерами, когда мы не располагаем какой-либо информацией, кроме самих данных.

Алгоритмы кластеризации, применяемые для решения данной задачи, могут основываться как на эвристиках, так и на формальных статистических подходах [13, 15]. Наибольшее распространение и практическое применение получили алгоритмы кластерного анализа семейства k-средних [16], и их нечеткая адаптации, алгоритм c-средних [17]. Данные алгоритмы имеют хорошую скорость сходимости, но они не гарантируют выделение каких-либо структурных особенностей в случае когда данные не являются хорошо разделимыми.

На Рис.2 показана кластеризация тестовых данных, представляющих из себя смесь трех двумерных нормальных распределений различными методами. Цветами выделены точки, соответствующие полученным кластерным разбиениям. Также на графиках показаны эллипсы рассеивания двумерных нормальных распределений, параметры каждого из которых были оценены по точкам соответствующих кластерных разбиений.

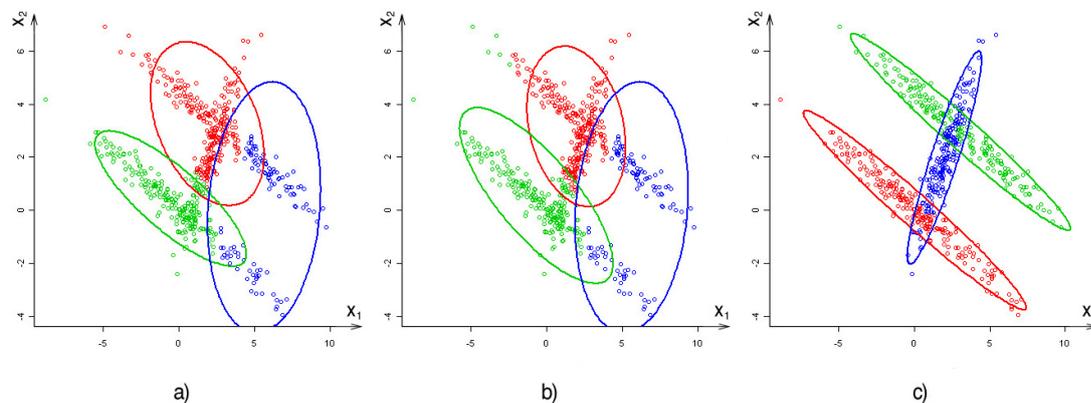


Рис. 2. Различные алгоритмы кластеризации, примененные к смеси из 3 нормальных распределений

- a) кластеризация по алгоритму k -средних, число кластеров $G = 3$;
- b) кластеризация по алгоритму нечетких c -средних, число кластеров $G = 3$;
- c) кластеризация с помощью EM алгоритма, число кластеров $G = 3$.

На графике хорошо видно, что ни алгоритм k -средних (a), ни алгоритм нечетких c -средних (b), не смогли корректно определить группировки данных. Для сравнения с ними, на правом графике (c) показан результат кластеризации с помощью EM алгоритма [13, 15, 8], который корректно разделил данные на 3 кластера.

Ранее мы заметили, что для нечеткого классификатора представление нечетких правил в виде многомерных функций принадлежности (6),(7), образует в пространстве входных переменных эллиптические области. При инициализации базы правил с помощью алгоритмов k - и c -средних, оценки параметров эллиптических областей по полученным разбиениям не гарантируют соответствия истинной структуре данных. Будет потеряно много информации, так как алгоритмы k - и c -средних в процессе кластеризации не учитывают повороты и корреляции в данных.

Для того, чтобы избежать этих проблем и наиболее эффективно инициализировать базу правил нечеткой продукционной модели, эллиптические области, соответствующие многомерным функциям принадлежности, следует искать с помощью модельно-ориентированного кластерного анализа.

3.1. Модельно-ориентированный кластерный анализ

В модельно-ориентированном кластерном анализе делается предположение, что данные порождаются смесью вероятностных распределений, в которой каждая компонента смеси представляет некоторую подгруппу в данных, то есть кластер. Инте-

ресующая нас плотность распределения имеет вид:

$$(9) \quad p(x) = \sum_{j=1}^G \phi_j f_j(x|\theta_k), \text{ где}$$

- $x = [x_1, \dots, x_m]^T$, $x \in \mathbb{R}^m$ - вектор входных переменных;
- G - число компонент смеси плотностей;
- $f_j(x|\theta_j)$ - функция плотности распределения j -ой компоненты смеси распределений, $j = \overline{1..G}$;
- θ_j - параметры плотности j -ой компоненты смеси распределений;
- ϕ_j - нормализующий вес j -ой компоненты смеси распределений,

$$\sum_{j=1}^G \phi_j = 1.$$

Мы будем рассматривать случай, когда $f_j(x|\theta_j)$ задают многомерные нормальные распределения: $f_j(x|\theta_j) \sim \mathcal{N}(a_j, \Sigma_j)$. Таким образом, параметры θ_j будут состоять из параметра средних многомерного нормального распределения a_j и параметров ковариационной матрицы Σ_j : $\theta_j = \{a_j, \Sigma_j\}$.

Главным образом использование данных методов обусловлено тем, что в задаче инициализации базы правил требуется получить именно многомерные нормальные функции принадлежности.

3.2. EM-алгоритм

Одним из наиболее популярных и эффективных методов оценки параметров θ_k смеси многомерных нормальных распределений является EM алгоритм [13, 15]. Суть алгоритма заключается в том, чтобы последовательно находить оценки максимального правдоподобия параметров вероятностных распределений, зависящих как от параметров θ_k , так и от скрытых параметров w_{kj} , $j = \overline{1..G}$, $k = \overline{1..N}$. При этом, скрытые параметры рассматриваются и оцениваются, как наблюдаемые.

Для оценки того, насколько успешно найдено разбиение данных на кластера, в EM алгоритме используется логарифм функции правдоподобия:

$$(10) \quad L(\theta_j, \phi_j, w_{kj}|x) = \prod_{j=1}^N \log(p(x_j)).$$

EM алгоритм максимизирует логарифм функции правдоподобия (10), последовательно выполняя два шага:

- На E-шаге по текущему приближению параметров θ_j, ϕ_j вычисляются ожидаемые значения скрытых переменных w_{kj} .
- На M-шаге по функции правдоподобия максимизируются параметры θ_j, ϕ_j , тем самым увеличивая ожидаемое правдоподобие, полученное на E-шаге.

EM алгоритм:

Выбор числа кластеров G .

Инициализация параметров μ_j, Σ_j, ϕ_j с помощью алгоритма k -средних.

Repeat**Е-шаг.**

$$w_{ij} \leftarrow p(f_j | x_i) = \frac{\phi_j}{(2\pi)^{\frac{m}{2}} |\Sigma_j|^{\frac{1}{2}}} e^{-\frac{1}{2}(x_i - \mu_j)^\top \Sigma_j^{-1} (x_i - \mu_j)}$$

М-шаг.

$$\phi_j \leftarrow \frac{1}{N} \sum_{i=1}^N w_{ij}$$

$$\mu_j \leftarrow \frac{\frac{1}{N} \sum_{i=1}^N w_{ij} x_i}{\frac{1}{N} \sum_{i=1}^N w_{ij}}$$

$$\Sigma_j \leftarrow \frac{\frac{1}{N} \sum_{i=1}^N w_{ij} (x_i - \mu_j)(x_i - \mu_j)^\top}{\frac{1}{N} \sum_{i=1}^N w_{ij}}$$

Until алгоритм сошелся, или выполнены условия его остановки.

Единственным параметром, необходимым для работы алгоритма, является число кластеров G , которое вводится пользователем. Также, несмотря на то, что алгоритм k -средних не дает достаточный результат при кластеризации только им, с его помощью можно эффективно найти начальное приближение параметров смеси нормальных распределений (9).

3.3. Байесовский информационный критерий

Как правило, число кластеров в данных заведомо неизвестно. В отличие от алгоритмов k - и s -средних, где для определения успешности разбиения используются различные индексы валидности кластеризации, такие как индекс Xie-Beni [18, 19, 9], в модельно-ориентированном кластерном анализе мы можем воспользоваться более эффективным способом выбора оптимальной модели для разного числа кластеров.

С увеличением числа компонент смеси распределений, одновременно растут как число параметров модели, так и ее правдоподобие. Если переписать задачу определения числа кластеров, как задачу оценки качества аппроксимации плотности распределения моделью смеси распределений, мы можем воспользоваться функцией правдоподобия параметров данной модели. Для того, чтобы в поисках модели с максимальным правдоподобием, мы не сошлись к модели, в которой число кластеров соответствует числу точек, необходим критерий для определения оптимального числа компонент смеси плотностей.

Воспользуемся Байесовским информационным критерием (BIC), также известным как информационный критерий Шварца [20]. Данный критерий ищет баланс

между приростом правдоподобия и штрафом на добавление параметров модели. Критерий задается формулой:

$$(11) \quad BIC = 2 \log(L(x, \hat{\theta}_j)) - n_{par} \log(N), \text{ где}$$

- $\log(L(x, \hat{\theta}_j))$ - логарифм максимума функции правдоподобия для j -ой модели;
- n_{par} - число независимых параметров j -ой оцениваемой модели:
 - $G - 1$ параметров $\phi_j, j = \overline{1..G}$;
 - Gm параметров m -мерных векторов средних значений кластеров $\mu_j, j = \overline{1..G}$;
 - $G \frac{m(m+1)}{2}$ параметров ковариационных матриц $\Sigma_j, j = \overline{1..G}$;

$$n_{par} = (G + 1)m + G \frac{m(m+1)}{2} - 1.$$

- N - число точек, по которым оценивается модель.

Байесовский информационный критерий успешно используется во многих задачах оценки числа параметров статистических моделей [13, 14]. Стоит отметить, что сами по себе значения информационного критерия ВИС имеют наглядный смысл только при сравнении между собой для одних и тех же данных, и не несут иной смысловой нагрузки о качестве модели.

В качестве искомой модели выбирается модель с числом компонент G , для которой достигается максимум информационного критерия.

3.4. Инициализация базы правил

Применим описанный метод кластеризации для инициализации базы правил нечеткой продукционной модели. Для задачи классификации, требуется отдельно для каждого класса провести кластеризацию соответствующих ему точек.

Для этого, каждому из классов необходимо указать искомые числа кластеров G_{init} , среди которых мы будем искать наилучшим образом описывающую наши данные модель смеси распределений.

Далее, для каждого $G \in G_{init}$ и соответствующего ему набору данных, проводится кластеризация по EM алгоритму, описанному выше. После завершения работы алгоритма, для каждой полученной модели $p(x, \theta_k, \phi_k)_g, g \in G_{init}$ по (11) вычисляется значение Байесовского информационного критерия BIC_g .

В качестве результата инициализации нечетких продукционных правил соответствующего класса выбирается та модель смеси распределений, для которой достигается максимум информационного критерия:

$$(12) \quad \begin{aligned} p(x, \theta)_{result} &= p(x, \theta_{k_{result}}, \phi_{k_{result}}), \\ k_{result} &= \arg \max_{k \in G_{init}} BIC_k. \end{aligned}$$

4. Нечеткие продукционные нейронные сети

С помощью кластеризации можно успешно инициализировать базу правил. Для того, чтобы достичь наилучшей точности работы нечеткой продукционной модели, структурные параметры правил требуется оптимизировать. Наиболее эффективным и хорошо изученным способом оптимизации параметров нечетких продукционных моделей является их представление в виде искусственных нейронных сетей [6, 3, 21].

4.1. Искусственные нейронные сети

Искусственные нейронные сети - популярный метод машинного обучения, применяющийся во многих прикладных задачах: в распознавании образов [22, 5], классификации [11], управлении динамическими системами [23].

Нейронная сеть состоит из ряда слоев, в каждом из которых расположены простые обрабатывающие элементы - нейроны. Некоторые из нейронов содержат параметрическую функцию активации, преобразующую поступающие ей входные сигналы, остальные нейроны выполняют промежуточные вычисления. Данные для сети подаются на входной слой, после чего они распространяются на последующие слои по заданным в нейронной сети правилам.

Основное преимущество нейросетевого подхода заключается в возможности обучения и адаптации параметров нейронных сетей. Для этого не требуются полные знания об исследуемом объекте, такие как, например, его четкая математическая модель. На основе обучающей выборки из пар входных-выходных значений, параметры нейронной сети можно оптимизировать таким образом, чтобы нейронная сеть аппроксимировала имеющиеся в данных скрытые функциональные зависимости. Данный вид оптимизации параметров нейронных сетей называется обучением с учителем [1].

Однако, у нейронных сетей есть ряд недостатков. Во-первых, в нейронных сетях невозможно в явном виде наглядно интерпретировать функциональную зависимость между входными и выходными данными [22, 24]. Во-вторых, нет общего способа построения нейронных сетей. Неизвестен алгоритм расчета количества слоев сети и количества нейронов в каждом слое: их число обычно определяется экспериментально, методом проб и ошибок.

Этих недостатков лишены нечеткие продукционные модели. Таким образом, подходы к моделированию на основе нечетких и нейросетевых моделей взаимно дополняют друг друга. В качестве наглядного примера сравнения нейронных сетей различных архитектур, на Рис.3 представлено их применение к известной задаче классификации двух спиралей.

Нечеткие продукционные модели удобно представимы в виде нечетких продукционных сетей - сетевых структур, элементы, или совокупности элементов которых реализуют различные компоненты нечетких моделей и этапы нечеткого вывода. Данные сети идентичны по структуре многослойным нейронным сетям с прямым распространением сигнала. Элементы каждого слоя данных сетей реализуют отдельный этап нечеткого вывода в нечеткой продукционной модели.

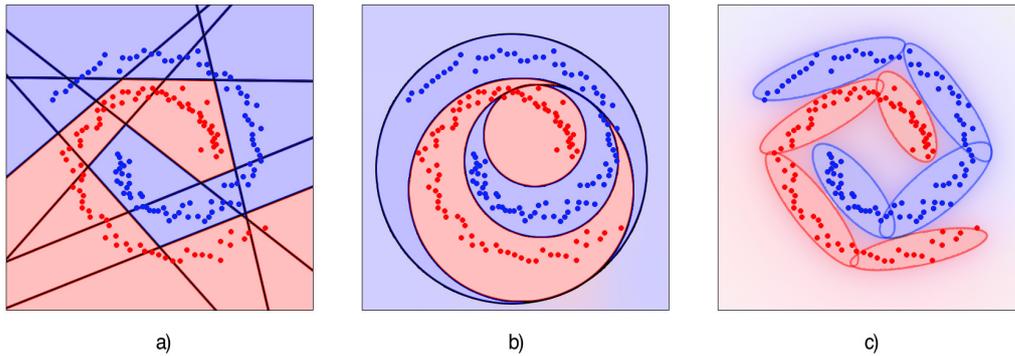


Рис. 3. Задача классификации двух спиралей, решенная нейронными сетями различных архитектур

- a) Разделение двух спиралей с помощью нейросети с сигмоидными функциями активации нейронов. Это margin-based [1, 22] нейронная сеть, в которой каждый нейрон задает разделительную гиперплоскость. Оптимальное число разделительных плоскостей находится с помощью метода проб и ошибок; трактовка результата затруднительна из-за взаимодействия с другими плоскостями.
- b) Разделение двух спиралей с помощью нейросети с радиальными базисными функциями активации нейронов. Это density-based [1, 24] нейронная сеть, в которой каждый нейрон задает сферическую область во входных данных. Оптимальное число нейронов находится с помощью метода проб и ошибок; инициализация сети производится эвристически, трактовка результата затруднительна из-за того, что конечный вид сферических областей после обучения сети может быть произвольным и не репрезентативным. Данный результат соответствует исследованиям геометрической интерпретации нейронных сетей [24].
- c) Разделение двух спиралей с помощью нечеткой продукционной нейронной сети, полученной из системы нечеткого вывода с инициализацией базы правил с помощью EM алгоритма. Это также density-based нейронная сеть, только при инициализации классов с помощью EM алгоритма была найдена наилучшая аппроксимация плотности распределений каждого из классов. Оптимальное число компонент плотностей находится с помощью информационного критерия для оценки плотности распределения.

4.2. Нейро-сетевое представление нечетких продукционных моделей

При представлении правил нечеткой продукционной модели в виде многомерных функций принадлежности с диагональной ковариационной матрицей (6), (7), предложенная в [6] структура нечеткой продукционной нейронной сети, соответствующая данной модели нечеткого вывода, не изменится. Нашей целью стоит построение нечеткой продукционной нейронной сети с многомерными функциями принадлежности в случае полных ковариационных матриц (6).

4.2.1. Представление ковариационных матриц в нейронных сетях.

С геометрической точки зрения, параметры стандартных отклонений b_{ij} , $i = \overline{1..G}$,

$j = \overline{1..m}$, одномерных функций принадлежности (3) обычной нечеткой продукционной модели (4) соответствуют размахам главных осей рассеивания эллипсоида многомерной гауссовской функции принадлежности (5) $\mu_{\hat{A}_i}(x)$, $i = \overline{1..G}$ (6). Для многомерной гауссовской функции принадлежности с полной ковариационной матрицей Σ_i , $i = \overline{1..G}$, размахи главных осей рассеивания эллипса соответствуют собственным числам ковариационной матрицы.

Таким образом, исследуемая модель (4) может быть представлена таким образом, чтобы параметры размахов одномерных функций принадлежности соответствовали собственным числам ковариационной матрицы многомерной функции принадлежности. Для представления такой модели в виде нейронной сети, для каждого правила следует сделать разложение ковариационной матрицы многомерной функции принадлежности по собственным векторам:

$$(13) \quad \Sigma_i = U_i \Lambda_i U_i^T = \sum_{j=1}^m \lambda_{ij} u_{ij} u_{ij}^T,$$

$$U_i = \begin{pmatrix} | & & | \\ u_{i1} & \cdots & u_{im} \\ | & & | \end{pmatrix}, u_{ij} = \begin{pmatrix} u_{ij1} \\ \vdots \\ u_{ijm} \end{pmatrix}, \text{ где}$$

- Σ_i - ковариационная матрица i -го правила, $i = \overline{1..G}$;
- λ_{ij} - j -ое собственное число ковариационной матрицы i -го правила, $i = \overline{1..G}$, $j = \overline{1..m}$;
- u_{ij} - собственный вектор, соответствующий j -му собственному числу ковариационной матрицы i -го правила, $i = \overline{1..G}$, $j = \overline{1..m}$;
- Λ_i - диагональная матрица, составленная из собственных чисел ковариационной матрицы i -го правила, $i = \overline{1..G}$, $j = \overline{1..m}$;
- U_i - матрица, составленная из собственных векторов ковариационной матрицы i -го правила, $i = \overline{1..G}$.

С помощью данного разложения, для i -го правила, $i = \overline{1..G}$, мы можем представить ковариационную матрицу Σ_i в виде двух групп параметров:

- матрицы поворота U_i , составленной из собственных векторов u_{ij} ковариационной матрицы Σ_i , $j = \overline{1..m}$;
- вектора, состоящего из собственных чисел λ_{ij} ковариационной матрицы Σ_i , $j = \overline{1..m}$.

Другим преимуществом использования данного разложения является простой способ вычисления обратной матрицы, необходимой для вычисления значений функций принадлежности (5). Так как ковариационная матрица Σ_i является симметричной и положительно-определенной, то с помощью (13) она представляется следующим образом:

$$(14) \quad \Sigma_i^{-1} = \sum_{j=1}^m \frac{1}{\lambda_{ij}} u_{ij} u_{ij}^T.$$

Также, для модели (7), учитывающей объем правил (8), разложение (13) предоставляет простой способ вычисления объема многомерных функций принадлежности:

$$(15) \quad \det(\Sigma_i) = |\Sigma_i| = \prod_{j=1}^m \lambda_{ij}.$$

Таким образом целесообразно вместо параметров ковариационной матрицы Σ_i хранить в нейронной сети разложение этой матрицы на собственные вектора u_{ij} и собственные числа λ_{ij} , $j = \overline{1..m}$.

4.2.2. Нейросетевое представление нечеткого классификатора. Построим нейронную сеть, представляющую нечеткую продукционную модель для задачи классификации с полными ковариационными матрицами многомерных функций принадлежности. Исходя из исследований, проведенных в работах [3, 6], искомая нейронная сеть будет иметь структуру с прямым распространением сигнала. На Рис.4 представлено нейросетевое представление правил модели(7).

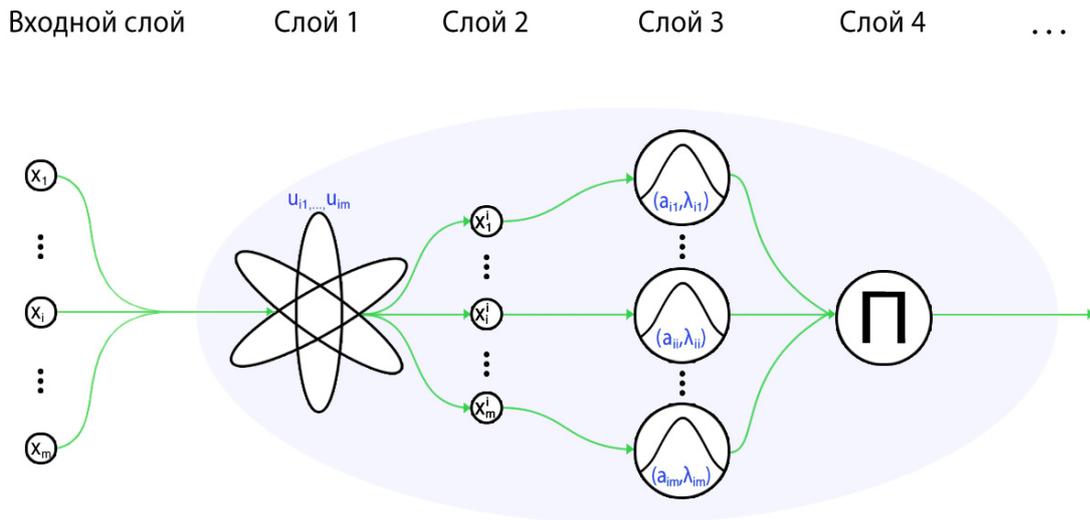


Рис. 4. Нейросетевое представление правил нечеткого классификатора

Опишем функции каждого из слоев данного представления:

Слой 1: элементы данного слоя содержат матрицы поворота U_i , полученные из собственных векторов ковариационных матриц Σ_i , $i = \overline{1..G}$. Для сохранения наглядного представления параметров нечеткой нейронной продукционной сети, входной вектор $x = [x_1, \dots, x_m]^T$, поступаая на вход правила, переводится в новые координаты, в которых ковариационная матрица Σ_i имеет диагональный вид. Поворот производится по формуле

$$(16) \quad \check{x} = U_i x.$$

Так как матрица U_i является ортогональной (ввиду свойств разложения по собственным векторам), преобразование (16) сохраняет расстояние между векторами x и a_i : $\|x - a_i\| = \|U_i(x - a_i)\|$.

Слой 2: элементы данного слоя представляют повернутые значения входных переменных каждого из правил. Данный слой служит для наглядности нейросетевого представления.

Слой 3: данный слой состоит из элементов, вычисляющих функции принадлежности $\mu_{\hat{A}_i}(\tilde{x})$ для данных после поворота (16) \tilde{x} , $i = \overline{1..G}$. Заметим, что мы можем упростить вычисления:

$$\begin{aligned} (x - a_i)^\top \Sigma_i^{-1} (x - a_i) &= (x - a_i)^\top \sum_{j=1}^m \frac{1}{\lambda_{ij}} u_{ij} u_{ij}^\top (x - a_i) = [z = U_i(x - a_i)] = \\ &= \sum_{j=1}^m \frac{z_j^2}{\lambda_{ij}}. \end{aligned}$$

Таким образом, в этом слое мы храним параметры центров многомерных функций принадлежности a_i , а также собственные числа λ_{ij} ковариационной матрицы Σ_i . Для удобства вычислений легче вычислять поворот, как $\tilde{x} = U_i(x - a_i)$. Для наглядности представления нейронной сети можно считать, что в этом слое повернутые значения $\tilde{x} = U_i x$ вычисляются для функций принадлежности с повернутыми центрами $\tilde{a}_i = U_i a_i$.

Так как для входного вектора ранее был произведен поворот (16), то для вектора в новых повернутых координатах вычисляется значение многомерной функции принадлежности с диагональной ковариационной матрицей. Ввиду эквивалентности между данным представлением и моделью с одномерными функциями принадлежности данный слой можно представить в виде ряда одномерных функций принадлежности, подобно тому, как это сделано в [3, 6].

Слой 4: состоит из элементов, выполняющих агрегирование одномерных функций принадлежности в новых координатах. В отличие от агрегирования в [3, 6], в элементах данного слоя вычисляются не только произведения всех одномерных функций принадлежности, но и объем каждого из правил. Таким образом, нейроны этого слоя вычисляют итоговые значения многомерных функций принадлежности нечетких продукционных правил $\frac{\mu_{\hat{A}_i}(x)}{V_i}$, $i = \overline{1..G}$.

На Рис.5 представлена общая структура нейронной сети, построенной для нечеткого классификатора с многомерными функциями принадлежности. В отдельные группы, по цветам, вынесены соответствующие правилам нейроны, задающие метки конкретного класса. Опишем слои предложенной нейронной сети, завершающие вычисления, получаемые после вычисления значений принадлежности нечетких правил:

Слои 1-4: в данном слое для каждого класса содержатся нейросетевые представления правил, представленные на Рис.4.

Слой 5: состоит из элементов, определяющих максимальное значение принадлежности среди всех правил, задающих метку соответствующего класса.

Слой 6: состоит из одного элемента, находящего индекс класса с максимальным значением принадлежности.

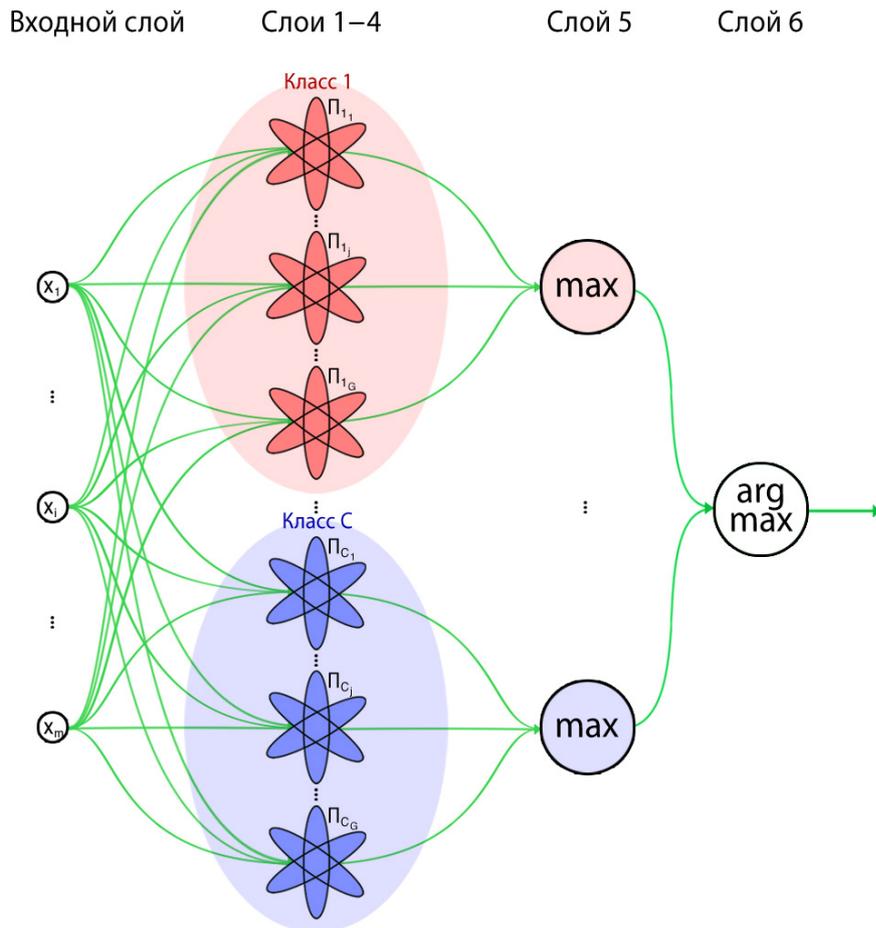


Рис. 5. Нейросетевое представление нечеткого классификатора

4.3. Оптимизация параметров нечеткой нейронной продукционной сети

4.3.1. Представление моделей с полными ковариационными матрицами. Из приведенного описания структуры нейронной сети следует, что данная сеть имеет два параметрических, то есть содержащих параметры, слоя: 1, и 3. Параметрами нечеткой нейронной продукционной модели являются:

- в первом слое, собственные вектора ковариационных матриц многомерных функций принадлежности правил $i = \overline{1..G}$, $j = \overline{1..m}$;
- в третьем слое, структурные параметры функций принадлежности нечетких множеств a_{ij} и λ_{ij} , $i = \overline{1..G}$, $j = \overline{1..m}$;

Для данной нейронной сети необходим эффективный алгоритм, с помощью которого нейросеть могла бы успешно работать на задачах большой размерности, так как число параметров первого слоя растет квадратично с ростом размерности. В отличие от [25] мы используем всю информацию о корреляциях между переменными. Нас интересует создание такого алгоритма, при использовании которого не требуется сокращать число параметров модели, жертвуя ее аппроксимационными и обобщающими способностями.

Так как исследуемая нами модель имеет представление в виде многослойной нейронной сети, то для оптимизации ее параметров могут быть применены алгоритмы машинного обучения, такие как алгоритм обратного распространения ошибки [1], случайный поиск и алгоритмы имитации отжига [2], или другие известные алгоритмы, такие как алгоритм Нелдера-Мида [26].

Нам потребуется обучающая выборка в виде N пар данных $(x, y)_k$, $k = \overline{1..N}$, где $x_k = [x_1^k, \dots, x_m^k]^T$ - k -ый входной вектор, y_k - k -ое эталонное значение метки класса.

Задача заключается в такой модификации параметров нечеткой продукционной модели, чтобы минимизировать ошибку, имеющую вид:

$$(17) \quad E = \sum_{k=1}^N \frac{1}{2} E^k, \\ E^k = (\hat{y}_k - y_k)^2,$$

где \hat{y}_k - значение, полученное нечеткой продукционной моделью для k -го входного вектора.

Для задачи классификации ошибка определяется следующим образом:

$$(18) \quad \hat{y}_k - y_k = \begin{cases} 0 & \hat{y}_k = y_k \\ 1 & \text{иначе} \end{cases}.$$

С другой стороны, оценка параметров ковариационной матрицы Σ_i , $i = \overline{1..G}$, а следовательно, оценка параметров ее собственных чисел λ_{ij} и собственных векторов u_{ij} , $i = \overline{1..G}$, $j = \overline{1..m}$, проводится с помощью EM алгоритма по методу максимального правдоподобия. Данный метод дает состоятельную оценку ковариационной матрицы многомерного нормального распределения, при этом требуя куда меньших вычислительных затрат, нежели алгоритмы машинного обучения, примененные к квадратичному по размерности входных данных числу параметров ковариационной матрицы.

4.3.2. Алгоритм обучения нейронной сети. При инициализации базы правил нечеткой продукционной модели с помощью EM алгоритма по (12), для всех параметров многомерных функций принадлежности вычисляются оценки максимального правдоподобия. Тем не менее, содержательный смысл оценок максимального правдоподобия отличается от оценок параметров, используемых в методах машинного обучения, а именно, в используемой мере ошибки E .

После оценки ковариационной матрицы по EM алгоритму, можно зафиксировать собственные вектора, а методами машинного обучения оптимизировать все остальные параметры по мере ошибки, а именно параметры собственных чисел и центров функций принадлежности, то есть структурные параметры центров a_{ij} и размахов λ_{ij} , $i = \overline{1..G}$, $j = \overline{1..m}$, многомерных функций принадлежности (5), содержащихся в третьем слое.

Одним из наиболее известных и часто применяемых на практике методов оптимизации нелинейных параметров в нейронных сетях является алгоритм обратного распространения ошибки [1]. На каждой итерации алгоритма на вход сети подаются все значения из обучающей выборки.

Сначала требуется вычислить выходное значение нейронной сети \hat{y}_k наблюдения x_k , $k = \overline{1..N}$, из обучающей выборки. Затем вычисляется ошибка нейронной сети на каждом наблюдении E^k . Это требуется для градиентной оптимизации нелинейных параметров, модифицируя их значения в обратном направлении градиента ошибки. Производится расчет градиента целевой функции относительно параметров функции принадлежности. Вычисления проводятся по следующим формулам:

$$(19) \quad a_{ij}^k(t+1) := a_{ij}^k(t) - \eta \frac{\partial E^k(t)}{\partial a_{ij}^k},$$

$$\lambda_{ij}^k(t+1) := \lambda_{ij}^k(t) - \eta \frac{\partial E^k(t)}{\partial \lambda_{ij}^k}, \text{ где}$$

- k - k -ый пример из обучающей выборки, $k = \overline{1..N}$;
- t - номер итерации оптимизационного процесса;
- η - положительная константа, регулирующая скорость градиентного спуска.

После каждой эпохи обучения производится проверка условий останковки оптимизационного процесса. В качестве таких критериев обычно выбирается целевая точность E_{max} , чувствительность к приращению целевой функции ε , и используются один, или одновременно, несколько из ниже перечисленных критериев:

- **Критерий 1:** $E^k < E_{max}$ для каждого элемента из обучающей выборки, $k = \overline{1..N}$;
- **Критерий 2:** $E = \sum_{k=1}^N \frac{1}{2} E^k < E_{max}$;
- **Критерий 3:** Выполнение правила останковки итерационного процесса: $|E^k(t) - E^k(t-1)| \leq \varepsilon$;
- **Критерий 4:** Ограничение числа итераций: $t > t_{max}$.

4.3.3. Общая схема инициализации и обучения нечеткого классификатора. Предложим общую схему алгоритма инициализации и обучения предложенной нечеткой продукционной модели для задачи классификации, представленной в виде нечеткой нейронной продукционной сети:

Шаг 0.

Выбирается число кластеров для инициализации базы правил: $G \in G_{init}$.

Задаются параметры сходимости оптимизационного процесса: E_{max} , t_{max} , ε .

Шаг 1.

С помощью EM-алгоритма кластеризуются входные данные всех классов для всех $G \in G_{init}$.

Шаг 2.

С помощью Байесовского информационного критерия, для каждого класса находится оптимальное число компонент G : $G = \arg \max_{i \in G_{init}} BIC_i$.

Шаг 3.

Для полученных компонент многомерных нормальных распределений $\mathcal{N}_i(\mu_i, \Sigma_i)$, инициализируем базу правил с многомерными функциями принадлежности $\mu_{A_i}(a_i, \Sigma_i)$, $i = \overline{1..G}$.

Шаг 4.

Для каждого из инициализированных правил проводим разложение по собственным векторам (13) ковариационных матриц Σ_i многомерных функций принадлежности. Тем самым инициализируем параметры λ_{ij}, u_{ij} , $i = \overline{1..G}$, $j = \overline{1..m}$.

while(Критерий 1 И Критерий 2 И Критерий 3 И Критерий 4)

Шаг t.

С помощью градиентного спуска (19), или другого алгоритма [26], приспособленного для данной задачи, оптимизируются параметры λ_{ij}, a_{ij} , $i = \overline{1..G}$, $j = \overline{1..m}$.

5. Практические примеры

Математический аппарат нечетких продукционных нейронных сетей с инициализацией базы правил с помощью кластерного анализа, был реализован в виде набора функций с помощью языка программирования R. Для кластеризации с помощью EM-алгоритма используется пакет **mclust**. Для гибридной оптимизации нелинейных параметров многомерных функций принадлежности методом Нелдера-Мида используется функция **optim** из базового пакета **stats**.

Данная система функций может быть использована как готовый инструмент для решения задач регрессии и классификации, аналогично тому, как используются обычные нейронные сети в пакете **nnet**, или машины опорных векторов в пакете **e1071**. Все далее представленные результаты получены с помощью данной программы.

5.1. Применение метода к искусственным данным

Рассмотрим задачу классификации на искусственных данных. Имеются два класса, распределения которых структурно представлены на Рис. 6(a).

Эти данные являются хорошим примером, на которых можно сравнить различные методы классификации. Очевидно, что эти данные не являются линейно-разделимыми. Более того, класс, выделенный красным цветом, разбивается на две подобласти. Одна из этих подобластей нелинейно отделима от другого класса, в то время как другая равномерно перемешана с другим классом, тем самым являясь от него неотделимой.

На Рис. 6(b) представлены используемые в анализе данные, в которые был внесен нормально-распределенный шум. Не трудно видеть, что построение точного классификатора с хорошими обобщающими способностями затруднительно по причине взаимо-проникновения классов.

В начале требуется провести кластеризацию каждого из классов. Мы не знаем, сколько кластеров соответствуют каждому из классов, поэтому число кластеров G , необходимое для EM-алгоритма, зададим отрезком $G_{init} = \overline{1..10}$. Значения Байесовского информационного критерия для кластеризации обоих классов на G_{init} компонент представлены на Рис. 7: по горизонтальной оси отложены количества кластеров; по вертикальной оси отложены значения информационного критерия.

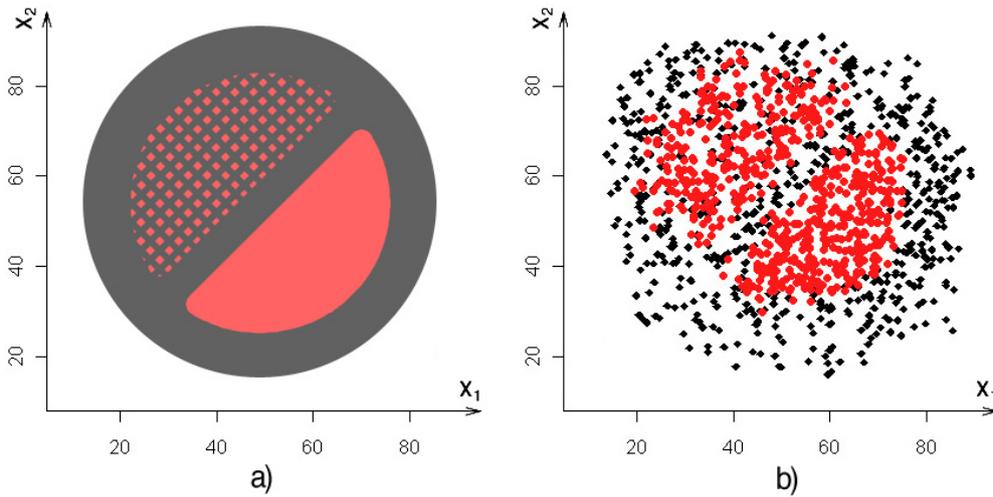


Рис. 6. Искусственные данные

- a) Структура искусственных данных.
 б) Выборка зашумленных данных, используемая в анализе.

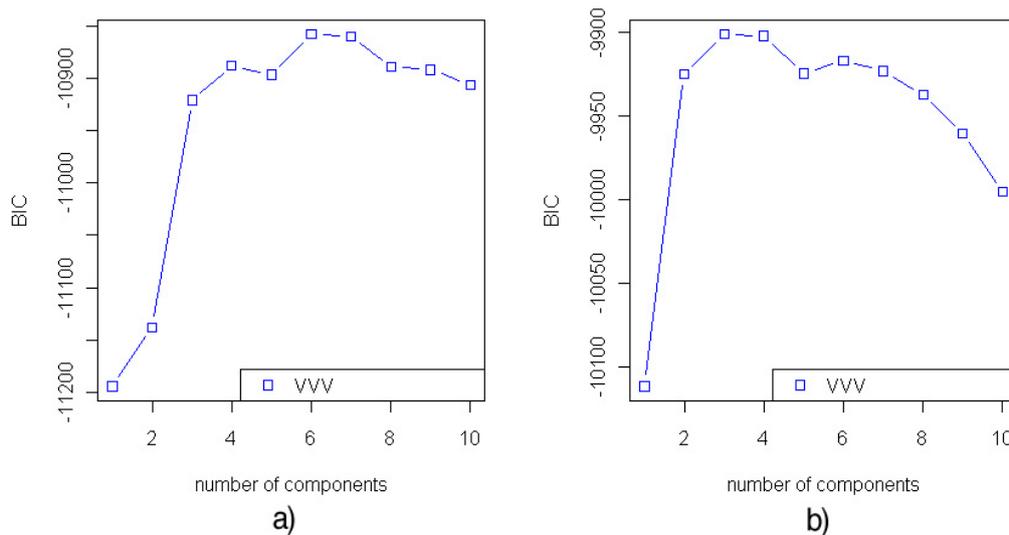


Рис. 7. Графики значений Байесовского информационного критерия

- a) Значения $BIC(11)$ первого класса для всех $G \in G_{init}$.
 б) Значения $BIC(11)$ второго класса для всех $G \in G_{init}$.

Для каждого из классов выберем то кластерное разбиение и число кластеров, для которого Байесовский информационный критерий максимален. В итоге получим разбиение входных данных на 9 кластеров. После нахождения оптимального разбиения, инициализируем базу правил, сопоставив каждому из кластеров метку соответствующего ему класса. Далее проводится алгоритм обучения нечеткой продукционной модели. Полученная в результате обучения модель нечеткого классификатора избоб-

ражена на Рис. 8(a). На Рис. 8(b,c) показаны ошибки классификации, выделенные красным цветом.

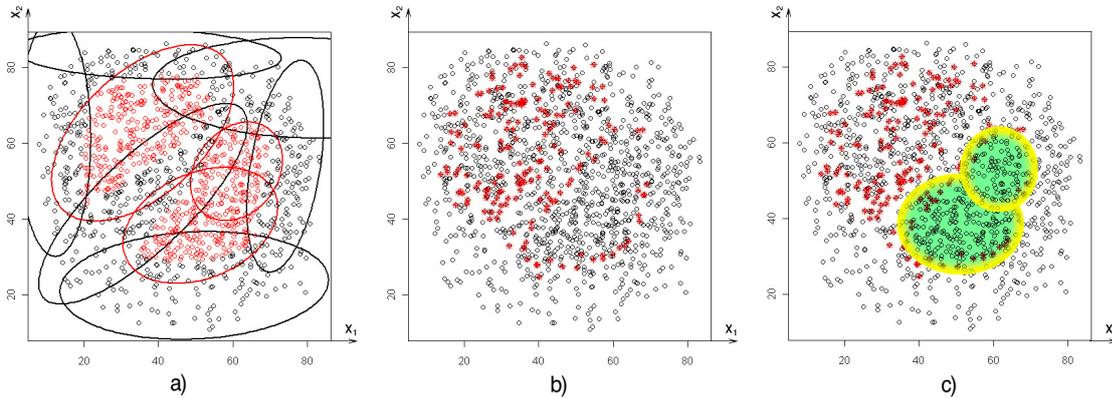


Рис. 8. Нечеткая нейронная продукционная сеть после гибридной оптимизации

- a) Классификация и эллипсы рассеивания двумерных функций принадлежности.
- b) Ошибки классификации нечеткой продукционной модели.
- c) Пример найденных областей безопасных параметров.

Для того, чтобы оценить полученные результаты, сравним предложенную нами модель с другими методами машинного обучения, используемыми в задачах классификации: машиной опорных векторов(SVM) [2], обычными нейронными сетями прямого распространения сигнала(NNet) [1], случайными лесами(Random Forest) [2] и деревьями принятия решений(CART) [2]. Для того, чтобы сравнение было репрезентативным, для каждого из этих методов была проведена оптимизация их параметров с помощью кросс-валидации [2]. Результат работы каждого из этих методов представлен в таблице ниже:

Таблица 1. Сравнение эффективности методов машинного обучения для задачи классификации.

Метод	Ошибка, %
Дерево классификации	16.90
Случайный лес	15.71
Нейронная сеть	15.30
Машина опорных векторов	14.72
Предложенная модель	14.26

Из таблицы мы видим, что предложенная модель показала наилучший результат по сравнению со всеми описанными выше методами. Это означает то, что с помощью информационного критерия мы нашли оптимальную инициализацию нечеткой продукционной модели классификатора.

Важной задачей является дальнейшее применения классификатора по полученным нечетким продукционным правилам. Так как они имеют вид (5) с многомерными функциями принадлежности, няглядная трактовка ковариационных матриц может

быть затруднительна. Правила можно представить в форме (2), если использовать вместо входных переменных линейные комбинации, соответствующие собственным векторам EVD-разложения ковариационных матриц. Для того, чтобы получить правила вида (2) в обычной форме для входных переменных без подобных преобразований, ковариационные матрицы правил можно диагонализировать.

Помимо способов, предложенных в [7, 9], одним из возможных способов провести диагонализацию является нормирование дисперсии соответствующей переменной на $1 - \rho$, где ρ - максимальная для ее столбца ковариационной матрицы корреляция с другой входной переменной. Пример данной диагонализации представлен на Рис.8(с), где в роли искомым правил выступают области, задаваемые той частью внутреннего(красного) класса, который отделим от внешнего(черного) класса.

5.2. Определение дефектов проката стали

5.2.1. Постановка задачи. Традиционно в области промышленности много усилий уделяется оптимизации производства. Важной является задача минимизации количества дефектов выпускаемой продукции. Один из примеров области, где возникает такая задача, это прокат стали. Прокат стали является сложным технологическим процессом, состоящим из несколько этапов, представленных на Рис.9 - отливка стали, прокат в клетях черновой обработки, прокат в клетях чистовой обработки, охлаждение, травление и т.д. На каждом из этапов процесс контролируется заданием определенных технологических параметров, количество которых может превышать сотню. Примерами параметров является химический состав примесей, геометрические параметры сляба, скорости вращения валков клетей, температура полосы при смотке, время охлаждения на складе. В зависимости от марки стали, требований заказчика и других условий технологические параметры постоянно меняются. При этом иногда выходная продукция, листы стали, содержит дефекты поверхности, что является недопустимым. Это приводит к необходимости обрезания участков полос с дефектами, т.е. к уменьшению эффективности производства.

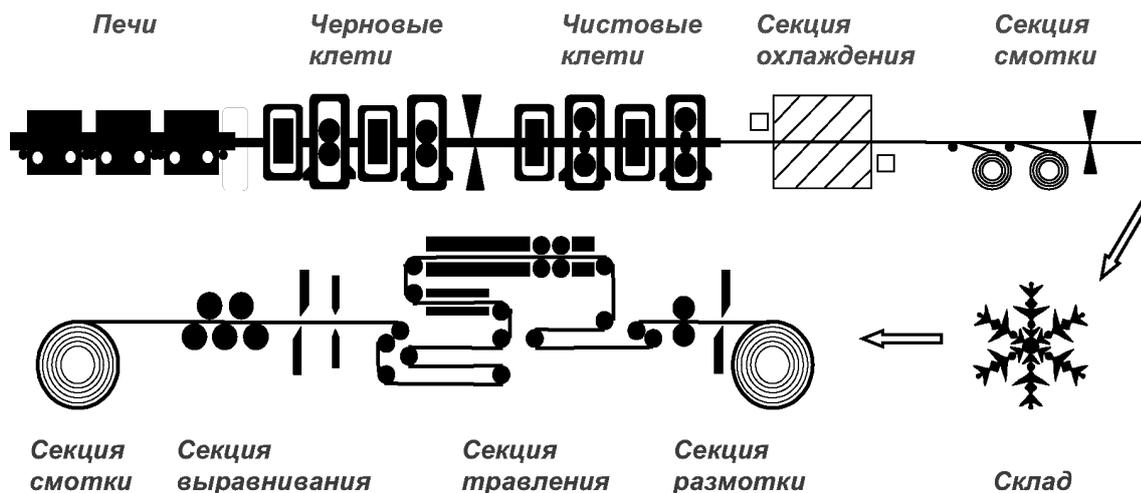


Рис. 9. Этапы процесса проката стали

Факторы, вызывающие дефекты, неизвестны, т.к. любой технологический параметр теоретически может влиять на их образование. Однако даже если не удастся

найти причину дефектов, то прогнозирование их появления с помощью методов машинного обучения может существенно снизить издержки производства. При наличии такой модели, перед прокатом каждой полосы делается прогноз возникновения дефекта. Если прогнозируется отсутствие дефекта, то никаких мер не предпринимается. В обратном же случае инженера, отвечающие за прокат, получают сигнал о возможном образовании дефекта и могут изменить параметры проката.

Прогнозирующая модель представляет собой полезный инструмент для задачи предотвращения дефектов. Однако, еще более полезным является определение областей безопасных параметров проката, т.е. таких, при которых вероятность появления дефектов минимальна. Наличие безопасных параметров дает инженерам прямую и четкую инструкцию предотвращения дефектов.

Уникальность предлагаемого нами подхода состоит в том, что он позволяет решить обе задачи - как прогноза дефектов, так и определение безопасных параметров. Ниже приводятся результаты тестирования метода на реальных данных по прокату стали.

5.2.2. Определение дефектов проката стали. Имеются 27-мерные данные, в которых 26 входных переменных задают параметры проката стали, а 27ая переменная - метка, показывающая, является ли данный лист стали дефектным, или нет.

Таблица 2. Данные проката стали.

$T_{КП}$	$T_{СМ}$	C	Mn	...	Скорость клетки 12	Наличие дефекта
837	653	0.065	0.41	...	683	0
833	656	0.065	0.41	...	720	0
841	655	0.065	0.41	...	675	0
⋮	⋮	⋮	⋮	...	⋮	⋮
831	660	0.068	0.38	...	699	0
843	662	0.068	0.38	...	680	1
835	662	0.068	0.38	...	657	0

Основной трудностью является то, что перегибы, как правило, появляются в незначительной части от общего числа листов стали. В наших данных мы имеем 581 точку без дефекта и всего 42 точки с дефектами. Изза недостаточного числа точек, в 26-мерном пространстве мы можем задать $G_{init_{good}} = \overline{1..10}$, $G_{init_{defect}} = 1$.

Для класса без дефектов максимум информационного критерия достигается при $G = 5$, а для класса с дефектами $G = 1$. По полученным результатам кластеризации инициализируем нечеткую продукционную модель и проведем по ней алгоритм обучения. В результате, после обучения, мы получим классификатор с целевой ошибкой в 1.76%, и матрицу классификации:

Таблица 3. Результаты классификации данных проката стали.

	Нет дефекта (истина)	Дефект (истина)
Нет дефекта (прогноз)	580	1
Дефект (прогноз)	15	27

Для оценки полученных результатов, сравним наш метод с другими методами

машинного обучения, использованными при сравнении эффективности предложенного метода на синтетических данных. Как и ранее, оптимизация параметров данных методов была проведена с помощью кросс-валидации. Результат работы каждого из этих методов представлен в таблице ниже:

Таблица 4. Сравнение эффективности методов машинного обучения для задачи сталепроката

Метод	Количество ошибок, %
Дерево классификации	11.87
Случайный лес	10.11
Нейронная сеть	4.49
Машина опорных векторов	4.33
Предложенная модель	1.76

Если для всех областей, соответствующих отсутствию дефектов, посчитать, сколько неверно классифицированных точек было отнесено к ним, то окажется, что в один из 5 кластеров не попадет ни одной точки с дефектами. Этот кластер, в который не попало ни одной неверно классифицированной дефектной точки и является решением задачи определения зон "безопасных" параметров.

Чтобы определить параметры, при которых не возникают дефекты, достаточно выставить все параметры проката стали такими же, как средние значения любого из найденных кластеров. Также для этих кластеров можно задать доверительные эллипсоиды, чтобы автоматически определять несоответствие введенным параметрам.

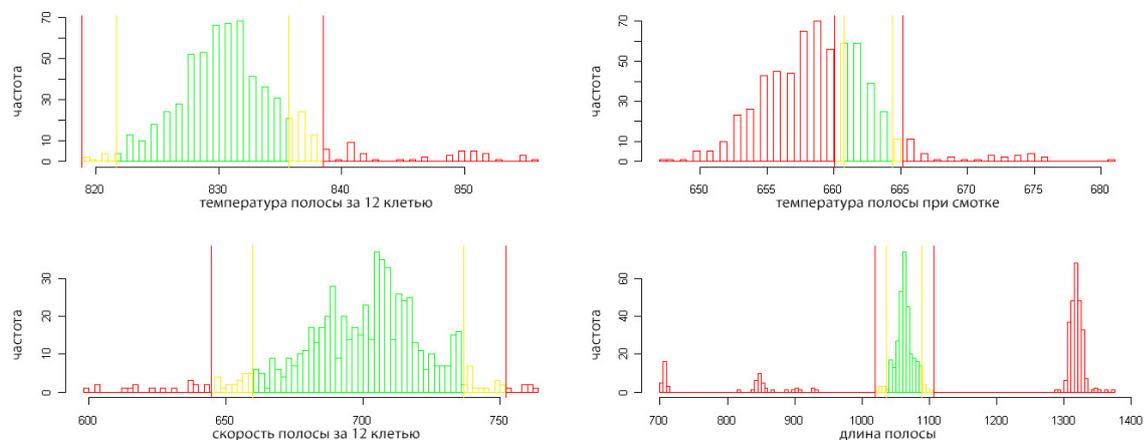


Рис. 10. Области безопасных параметров проката стали

На Рис.10 представлены полученные из найденного правила области безопасных параметров, спроецированные на 4 наиболее значимых для экспертов характеристики проката стали. Для обеспечения отсутствия появления дефектов стали, необходимо обеспечить безопасный интервал по каждой из переменных. Цветами указаны доверительные интервалы проекций многомерных нормальных распределений: красным обозначены области, не укладывающиеся в 3 стандартных отклонения; желтым показаны области, не укладывающиеся в 2 стандартных отклонения; зеленым указан наиболее предпочтительный безопасный интервал параметров в пределах двух стандартных отклонений по соответствующей переменной многомерного кластера.

Таким образом, решены обе поставленные задачи, а алгоритм обучения нечеткой нейронной продукционной модели был успешно применен на 26-мерных данных.

6. Заключение

В ходе решения поставленных задачи построения и оптимизации предложенных нечетких продукционных моделей с помощью нечетких нейронных продукционных сетей, были получены следующие результаты:

- Построено обобщение нечеткого классификатора для случая многомерных функций принадлежности.
- Исследован эффективный способ инициализации баз правил нечетких продукционных моделей с помощью EM-алгоритма кластеризации и автоматического определения оптимального числа кластеров с помощью Байесовского информационного критерия.
- Представлен алгоритм обучения нечетких продукционных моделей с многомерными функциями принадлежности.
- Предложенная модель была успешно протестирована на модельных и реальных данных. Был построен классификатор, показавший наилучший результат по сравнению с другими сравниваемыми методами машинного обучения. Были определены зоны безопасных параметров проката стали, при которых вероятность возникновения дефектов минимальна.

Рассмотренные нечеткие продукционные модели с многомерными функциями принадлежности показали успешные практические результаты, что делает их привлекательными методами решения задач машинного обучения. Наиболее сложным этапом построения данных моделей являются эффективные методы обхода проклятья размерности. Помимо рассмотренного в данной работе способа борьбы с этой проблемой, необходимы дополнительные исследования эффективных многомерных алгоритмов обучения.

С точки зрения теории нечетких множеств, многомерные функции принадлежности с полными ковариационными правилами имеют более сложную лингвистическую трактовку. Отдельного внимания заслуживают исследования в области удобного представления информации многомерных функций принадлежности. Для многомерных функций принадлежности является значимой информация об их пересечении. Для лучшей трактовки разделения многомерных функций принадлежности требуется исследовать нечеткие модели с правилами вида: **ЕСЛИ А И (НЕ В)**. Если дать экспертам информацию о пересечении функций принадлежности, то это сможет повысить точность воспроизведения результатов работы нечетких продукционных правил на входных данных.

Список литературы

1. Bishop C.M. Neural Networks for Pattern Recognition. New York: Oxford University Press Inc., 1995.
2. Bishop C.M. Pattern Recognition and Machine Learning. Springer, 2006.

3. Борисов В.В., Круглов В.В., Федулов А.С. Нечеткие модели и сети. М.: Горячая линия-Телеком, 2007.
4. Nahla Ibraheem Jabbar and Monica Mehrotra. Application of fuzzy neural network for image tumor description // World Academy of Science, Engineering and Technology. 2008. Vol. 44. P. 575-577.
5. Razvan Andonie, Levente Fabry-Asztalos, Nicholas Salim. Neuro-fuzzy prediction of biological activity and rule extraction for hiv-1 protease inhibitors // In: Proceedings of the 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology CIBCB '05. November 14-15. 2005. P. 1-8.
6. Jang J.-S.R. ANFIS: adaptive-network-based fuzzy inference systems // IEEE Transactions on Systems, Man and Cybernetics. 1993. Vol. SMC-23, No. 3. P. 665-685.
7. Babuska A.J., Szeifert R.F. Modified gath-geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models // IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics. 2002. Vol. SMC-32, No. 5. P. 612-621.
8. Verma N.K., Hanmandlu M. Adaptability in additive fuzzy system via EM algorithm // The International Journal of Artificial Intelligence and Machine Learning. , 6:35-42, 2006.
9. Tsekouras G.E., Sarimveis H. A hierarchical fuzzy-clustering approach to fuzzy modeling // Fuzzy Sets and Systems. 2005. Vol. 150, No. 2. P. 245-266.
10. Wen Yu, Xiaou Li. Fuzzy identification using fuzzy neural networks with stable learning algorithms // IEEE Transactions on Fuzzy Systems. 2004. Vol. FS-12, No. 3. P. 411-420.
11. Eftekhari M., Katebi S.D. Extracting compact fuzzy rules for nonlinear system modeling using subtractive clustering, GA and unscented filter // Applied Mathematical Modelling. 2008. Vo. 32, No. 12. P. 2634-2651.
12. Леоненков А. Нечеткое моделирование в среде MATLAB и fuzzyTECH. Санкт-Петербург: BHV, 2005.
13. Das Gupta U., Menon V., Babbar U. Parameter selection for EM clustering using information criterion and PDDP // IACSIT International Journal of Engineering and Technology. 2010. Vol. 2, No. 4. P. 340-344.
14. Penalver A., Escolano F., Saez J.-M. Two entropy-based methods for learning unsupervised gaussian mixture models // Lecture Notes in Computer Science. 2006. Vol. 4109. P. 649-657.
15. Baudry J.P., Raftery A.E., Celeux G.K., Gottardo R. Combining mixture components for clustering // Journal of Computational and Graphical Statistics. 2010. Vol. 19. P. 332-353.
16. Netanyahu N.S., Piatko C.D. An efficient k-means clustering algorithm: Analysis and implementation // IEEE Transactions Pattern Analysis and Machine Intelligence. 2002. Vol. PAMI-24. P. 881-892.
17. Miyamoto S., Ichihashi H., Honda K. Algorithms for Fuzzy Clustering Methods in c-Means Clustering with Applications. Berlin, Heidelberg: Springer-Verlag, 2008.
18. Kuo-Lung Wu, Miin-Shen Yang. A cluster validity index for fuzzy clustering // Pattern Recognition Letters. 2005. Vol. 26. P. 1275-1291.
19. Dae-Won Kim, Kwang H. Lee, Doheon Lee. On cluster validity index for estimation of the optimal number of fuzzy clusters // Pattern Recognition. 2004. Vol. 37. P. 2009-2025.
20. Schwarz G. Estimating the dimension of a model // Annals of Statistics. 1978. Vol. 6. P. 461-464.
21. Kosko B. Fuzzy systems as universal approximators // IEEE Transactions on Computers. 1994. Vol. 43. P. 1329-1333.
22. Draghici S. On the capabilities of neural networks using limited precision weights // Journal Neural Networks. 2002. Vol. 15, No. 3. P. 395-414.
23. Iyatomi H., Hagiwara M. Adaptive fuzzy inference neural networ // Pattern Recognition. 2004. Vol. 37. P. 2049-2057.
24. Ling Zhang, Bo Zhang. A geometrical representation of mcculloch-pitts neural model and its applications // IEEE Transactions on Neural Networks. 1999. Vol. NN-10, No. 4. P. 925-929.
25. Dickerson J.A., Kosko B. Fuzzy function approximation with ellipsoidal rules // IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics. 1996. Vol. SMC-26. P. 542-560.
26. Nelder J.A., Mead R. A simplex method for function minimization // Computer Journal. 1965, Vol. 7. P. 308-313.