

УДК 681.3.068

В.А. Силич, М.П. Силич, С.В. Аксёнов

Алгоритм построения нечеткой системы логического вывода Мамдани, основанный на анализе плотности обучающих примеров

Предложен алгоритм автоматического нахождения конфигурации нечеткой системы логического вывода типа Мамдани для решения задач аппроксимации, с помощью выделения оптимального числа нечетких кластеров гиперэллипсоидной формы в пространстве обучающих примеров и определения на их основе параметров функций принадлежности для входных переменных и результата вывода. Предложен вариант автоматического нахождения типов функций принадлежности для описания нечетких величин. Оценка эффективности алгоритма произведена при решении задачи восстановления сложной поверхности.

Ключевые слова: нечеткая система логического вывода Мамдани, нечеткая кластеризация, генетический алгоритм.

В настоящий момент вычисления, использующие достоинства нечетких систем, показали себя достаточно мощным и эффективным средством для решения многих задач контроля и управления в сложных программных и аппаратных комплексах [1]. Однако недостаток создания таких систем заключается в том, что набор правил вывода постулируется человеком-экспертом. Построение окончательной модели требует зачастую агрегации знаний специалистов многих областей и многократного тестирования системы. Экспертам, помимо правил, необходимо определиться с типами и параметрами функций принадлежности (ФП) нечетких множеств, заданных на совокупности входных и выходных параметров, а это требует достаточно длительного и утомительного процесса подбора указанных величин.

Существуют подходы по автоматическому и автоматизированному построению нечетких моделей. При их использовании либо пользователь, либо система задает количество термов для каждого из факторов, влияющих на результат вывода, и программная система равномерно разбивает отрезок оси каждого параметра между имеющимися в выборке максимальным и минимальным значениями. Одной из возможностей для автоматического нахождения параметров функций принадлежности нечеткой системы является использование алгоритмов как локальной, так и глобальной оптимизации. В обоих случаях, если не используется равномерное разбиение факторов, генерируется некоторое количество случайных нечетких множеств, описываемых заданными пользователем ФП, и настройке подвергаются аргументы последних. Недостатком подхода является то, что, несмотря на низкую погрешность при выполнении нечеткого вывода при условии успешного обучения модели, функции принадлежности и система правил не всегда будут отражать те правила вывода и те термножества, которые человек считает интуитивно понятными для создаваемой системы [2]. В этом случае нечеткая система работает как многослойный персептрон для решения задачи аппроксимации данных, а способность к логическому объяснению полученного результата вывода будет в значительной степени утрачена. Для устранения этого недостатка необходимо разработать алгоритм, позволяющий с помощью предобработки данных выделить оптимальное количество правил вывода для решения задачи аппроксимации, определить типы ФП для каждого из термножеств, использующихся в системе вывода, и произвести настройку параметров ФП системы.

Постановка задачи. Пусть имеется выборка данных, состоящая из входных векторов $\mathbf{D} = \mathbf{d}_i = \{d_{ij}\}$, $i = 1 \dots M$, $j = 1 \dots N$, и массива выходных значений $\mathbf{O} = \mathbf{o}_i$, где M – количество обучающих векторов, N – размерность входного вектора. Необходимо получить безызбыточную структуру нечеткой системы типа Мамдани, работающую как аппроксимационная модель, позволяющая получить выход \mathbf{r}_i при активации тестовым вектором \mathbf{d}_i , такой что $|\mathbf{r}_i - \mathbf{o}_i| < \varepsilon$.

Спроецируем все входные вектора на каждую ось. Предположим, что области оси для каждого из входных параметров и выхода с наибольшей концентрацией примеров соответствуют нечетким множествам, заданным для каждого из анализируемых параметров. Области пространства, на кото-

рых наблюдается концентрация векторов обучающих примеров $\{d_i, o_i\}$, позволяют судить о взаимосвязи входных аргументов и выходного, их корреляции, т.е. о существовании правила вывода, связывающего в единое терм-множество входных аргументов, попадающих в границы кластера, и терм-множества результата. Таким образом, задача построения системы нечеткого вывода заключается в поиске регионов концентрации обучающих примеров и выявлении взаимосвязи между терм-множествами входных и выходной переменных. Поиск этих областей концентрации векторов $\{d_i, o_i\}$ производится с помощью алгоритмов кластеризации. Для каждого кластера, обнаруженного в результате процедуры кластеризации, его центр будет соответствовать центрам (L-R)-ФП терм-множеств переменных, описывающих эту группу точек. Для нахождения степени принадлежности переменной к терм-множеству необходимо использовать меру, определяющую близость вектора к центру кластера или близость этого вектора к нескольким группам точек, а также «насыщенность» кластера обучающими векторами. Другим шагом алгоритма является нахождение оптимального типа ФП для каждого обнаруженного терм-множества и корректировка параметров ФП, позволяющая функционировать всем нечетким правилам как единое целое.

Нечеткая кластеризация данных. Обилие алгоритмов кластеризации естественным образом ставит вопрос, какой алгоритм выделения кластеров лучше всего использовать. Методы четкой кластеризации, например самоорганизующиеся карты Кохонена, метод K-средних или DBSCAN [3], разделяют пространство на непересекающиеся области и их форма (в случае первых двух – это гиперсферы, последний алгоритм позволяет получить кластер произвольной формы) не очень хорошо подходит для выделения нечетких множеств и связанных с ними правил вывода. Для решения задачи лучше подходят алгоритмы, основанные на выделении областей гиперэллипсоидной формы, использующие метрику Махаланобиса [2] для оценки расстояния между векторами кластера и его центром:

$$\text{Dist}_k = (\mathbf{x}_i - \mathbf{c}_k) \cdot (\det(\mathbf{A}_k))^{\frac{1}{K}} \cdot \mathbf{A}_k^{-1} \cdot (\mathbf{x}_i - \mathbf{c}_k)^T,$$

где \mathbf{A}_k – матрица ковариации для k -го кластера; \mathbf{c}_k – вектор значений центра кластера; \mathbf{x}_i – вектор обучающей выборки, получающийся добавлением к строке матрицы \mathbf{D} соответствующего ей значения выхода из \mathbf{O} , т.е. \mathbf{x}_i – вектор входов модели и её желаемый выход; K – количество кластеров; $k = 1 \dots K, i = 1 \dots M$.

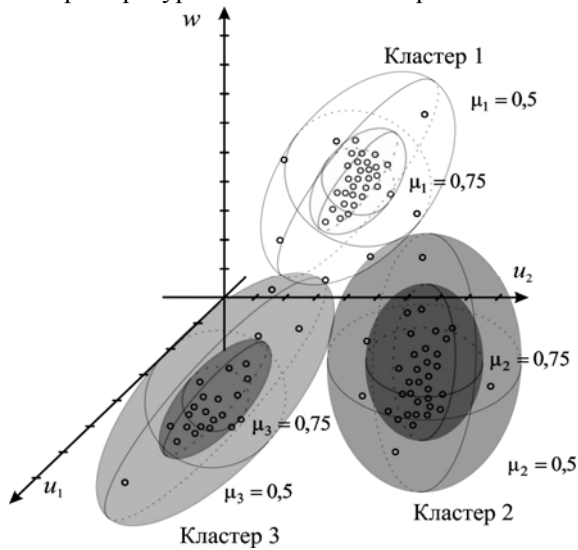
Для получения множества пересекающихся кластеров необходимо использовать методы, основанные на оценивании степени принадлежности каждого анализируемого вектора к каждому кластеру. Этим требованиям наиболее полно удовлетворяет алгоритм нечеткой кластеризации Густафсона–Кесселя [4]. Использование нечетких гиперэллипсов по сравнению с гиперсферами выглядит более естественно, т.к. в большей степени отражает взаимозависимость между компонентами входа d_i и выходным значением o_i . Одним из входных параметров алгоритма нечеткой кластеризации является выбор пользователем желаемого числа кластеров K . Практически достаточно сложно определить количество связанных областей визуально по выборке. Поскольку каждая найденная область отвечает за применение одного из правил вывода, необходимо точно определить количество кластеров. К сожалению, алгоритм нечеткой кластеризации очень чувствителен как к числу кластеров, так и к начальным установкам центров кластеров и инициализации матрицы нечетких значений, отражающей степени принадлежности каждой точки к каждому кластеру [3]. Зачастую требуется выполнить достаточно большое число рестартов алгоритма, чтобы получить удовлетворительное решение. Другой проблемой, связанной с принадлежностью векторов к разным кластерам (но с разной степенью), т.е. взаимным перекрытием кластеров, являются достаточно невысокие показатели индексов компактности и отделимости кластеров для полученных решений.

Для получения удовлетворительного решения необходимо приблизить начальные значения центров нечетких кластеров и значения нечеткой матрицы к желаемым значениям. Такое приближение можно получить с помощью самоорганизующейся карты Кохонена, выполняющей кластеризацию данных, качество результатов которой оценивается одним из методов анализа компактности/отделимости.

В качестве одной из мер оценки качества кластеризации алгоритмом Кохонена предложено использовать индекс Силуэта [5]. В этом случае для определения первоначального приближения пользователь устанавливает три значения для алгоритма Кохонена: минимальное количество кластеров NCMIN, максимальное количество кластеров NCMAX и число рестартов NRESTARTS. Для каждого

числа кластеров NC в диапазоне от $NCMIN$ до $NCMAX$ генерируется $NRESTARTS$ решений, т.е. всего анализируются $(NCMAX - NCMIN + 1) * NRESTARTS$ решений, для каждого из которых вычисляется индекс Силуэта. Решение из этого набора, обладающее наибольшим значением индекса, принимается за начальное приближение нечеткой кластеризации.

Теперь значения центров нечетких кластеров алгоритма Густаффсона–Кесселя устанавливаются равными центрам кластеров, найденным с помощью карты Кохонена, т.е. центры помещены в регионы наибольшей концентрации примеров, и значения в нечеткой матрице устанавливаются исходя из принадлежности вектора к тому или иному четкому кластеру по следующему правилу: значение элемента нечеткой матрицы является случайной величиной в диапазоне $(0,75; 1)$, если вектор принадлежит соответствующему четкому кластеру или равно случайному значению на интервале $(0; 0,25)$ в противном случае. При запуске алгоритма нечеткой кластеризации происходит коррекция значений в нечеткой матрице и значений центров. Достоинство предлагаемого решения заключается также и в его производительности, т.к. классический алгоритм Густаффсона–Кесселя значительно отбирает ресурсы из-за ёмких матричных вычислений.



На рис. 1 показан пример кластеризации нечеткими гиперэллипсами, заметим, что даже точки, не попавшие в регионы максимальной концентрации векторов, обладают вектором принадлежности к каждому из кластеров и окончательная форма каждого кластера учитывает распределение разрозненных векторов. Здесь u_1, u_2 – являются компонентами вектора \mathbf{d}_i , а w – компонента \mathbf{o}_i (выходное значение).

Рис. 1. Пример кластеризации Густаффсона–Кесселя с указанием областей со степенями принадлежности $\mu = 0,75$ – «внутренние» гиперэллипсы и $\mu = 0,5$ – «внешние» гиперэллипсы

Более темными гиперэллипсами изображены области пространства, включающие векторы со степенью принадлежности к выбранному кластеру $\mu > 0,75$, более светлые «внешние» гиперэллипсы – ограничивают векторы с $\mu > 0,5$. Заметим, что для построения ФП для переменных в каждом кластере используются также и точки, принадлежащие в большей степени к другому кластеру, однако обладающие положительным значением (пусть даже очень небольшим) степени принадлежности к кластеру, для которого определяются ФП.

Генерация правил вывода по полученным нечетким кластерам. Спроецировав все точки кластера k на ось одного из входных параметров или выходного значения, получится набор значений $\{y_i, \alpha_i\}$, $i = 1 \dots R$, где y_i – значение входного или выходного параметра, в зависимости от того, для какой переменной строится ФП, т.е. $y_i = u_j$ или w , α_i – степень принадлежности к кластеру k ; R – количество точек, попавших в кластер K . Эти значения $\{y_i, \alpha_i\}$ подходят к описанию дискретного нечеткого множества и теперь по этим значениям необходимо восстановить вид ФП. На рис. 2 показана идея формирования ФП для переменных, попавших в кластер 1 из рис. 1.

Пусть имеются две входные переменные u_1 и u_2 и выход w , $\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1$ – терм-множества, полученные для кластера 1, переменных u_1, u_2, w соответственно. Тогда правило вывода, получаемое из данного кластера, записывается:

1) Если $u_1 = \mathbf{A}_1$ и $u_2 = \mathbf{B}_1$ то $w = \mathbf{C}_1$.

То есть если определить терм-множества нечетких переменных, попадающих во второй и третий кластер из рис. 1, то получается дополнительно два правила

2) Если $u_1 = \mathbf{A}_2$ и $u_2 = \mathbf{B}_2$, то $w = \mathbf{C}_2$.

3) Если $u_1 = \mathbf{A}_3$ и $u_2 = \mathbf{B}_3$, то $w = \mathbf{C}_3$.

Итоговая система вывода будет иметь вид, показанный на рис. 3.

Рис. 2. Пример получения ФП входных и выходной переменной путем анализа степени принадлежности вектора к кластеру 1 из рис. 1

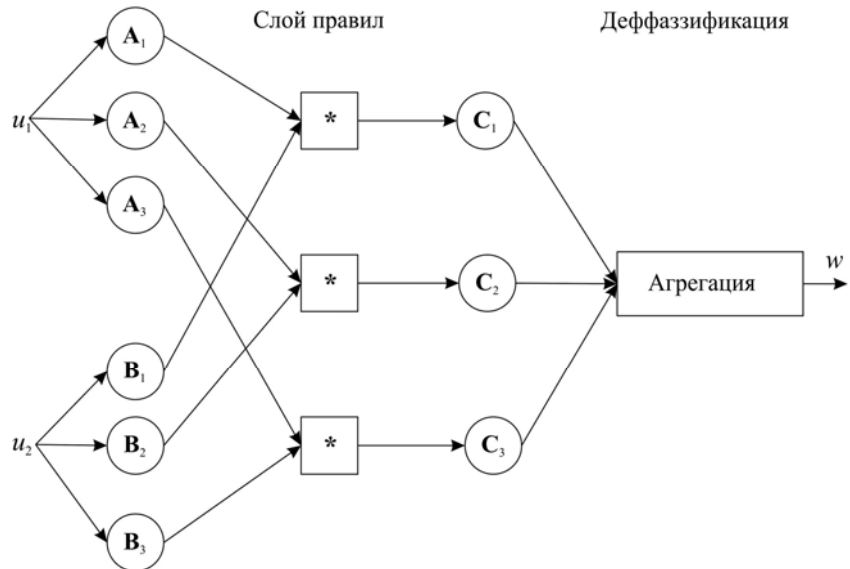
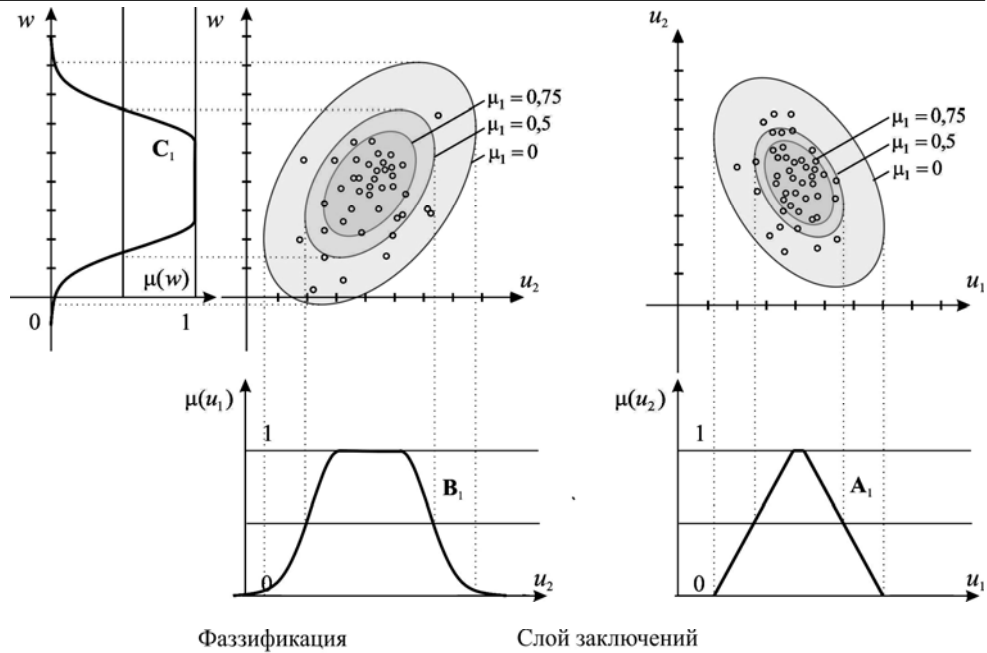


Рис. 3. Нечеткая система вывода Мамдани для трех кластеров

Определение типов и параметров функций принадлежности. Настройка нечеткой системы. Пусть есть набор $(L-R)$ -функций F , определенных пользователем для данной задачи. К стандартному набору можно отнести треугольную, трапециевидную, гауссову и колоколообразную ФП, но предлагаемый алгоритм не накладывает ограничений на вид ФП. Один из аргументов ФП-центр – известен (это значение компоненты центра нечеткого кластера, для которой строится ФП), необходимо найти остальные параметры, выражающие степень группировки точек у центра и пологость функции. Задачи подобного типа, связанные с анализом групп разнородных объектов и определением их характеристик, достаточно эффективно решаются с использованием генетического алгоритма [5]. В качестве фитнес-функции берется функция минимизации среднеквадратичной ошибки между значениями полученной ФП и нечетким значением принадлежности вектора, содержащего анализируемую компоненту к нечеткому кластеру α_i .

Хромосома представляет собой набор значений всех параметров ФП за исключением центра. На рис. 4 показан вид хромосомы для ФП $f(c, p_1, p_2, \dots, p_G)$, где c – центр ФП; p_i – оставшиеся аргументы ФП (пологость, ширина центра и т.д.); G – количество параметров ФП за исключением центра.

p_1	p_2	\dots	\dots	p_G
-------	-------	---------	---------	-------

Рис. 4. Вид хромосомы ФП $f(c, p_1, p_2, \dots, p_G)$, использующейся для детектирования типа ФП

Для нахождения оптимального решения генерируются несколько популяций для каждой из заданных пользователем типов ФП (колоколообразная, гауссова и т.д.) и запускается параллельно не-

сколько генетических алгоритмов для каждого типа ФП. Среди всех полученных решений (для всех типов ФП) выбирается наилучшее решение, обладающее минимальным фитнесом, – эта ФП будет использоваться в искомой нечеткой системе.

Однако полученные таким образом ФП представляют собой только начальное приближение (хотя и довольно близкое к оптимальному), т.к. они были рассчитаны независимо друг от друга. Требуется дополнительная настройка параметров p_j на первоначальной выборке $= \{d_{ij}, o_i\}$. Коллективная настройка параметров выполняется генетическим алгоритмом и для поиска решения было предложено взять хромосому, кодирующую все параметры всех ФП, использующихся в нечеткой системе за исключением центров ФП. Вид хромосомы для настройки всей системы представлен на рис. 5, где $\mu_1(d_1) = f(c_1, p_{11}, p_{21}, \dots, p_{G1})$, $\mu_2(d_1) = f(c_1, p_{11}, p_{21}, \dots, p_{G1}), \dots, \mu_T(o) = f(c_{1H}, p_{1H}, p_{2H}, \dots, p_{GH})$; T – количество правил вывода; H – число ФП для всей системы вывода.

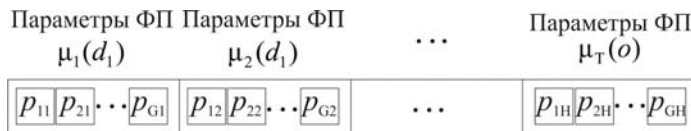


Рис. 5. Вид хромосомы, использующейся для настройки нечеткой системы

Создается значительный объем популяции, в которой представлены различные ФП с разными параметрами, и запускается процедура естественного отбора. Значение фитнес-функции представляет собой среднеквадратичную ошибку реального выхода системы от значений o_i на тестовом наборе из $\{d_{ij}, o_i\}$.

Алгоритм

Шаг 1. Начальное приближение. Кластеризация входного набора данных алгоритмом Кохонена. Выполняется несколько раз, число рестартов задается пользователем.

Шаг 2. Выделение лучшей четкой кластеризации. Оценка качества полученных кластеризаций с помощью индекса Силуэта. Нахождение кластеризации, обладающей наибольшим значением индекса Силуэта, эта кластеризация объявляется лучшей.

Шаг 3. Установка начальных значений для алгоритма нечеткой кластеризации. Инициализация алгоритма Густафсона–Кесселя центрами, обнаруженными лучшей кластеризацией алгоритмом Кохонена. Инициализация матрицы нечетких значений по следующему правилу: значение элемента матрицы равно случайному значению в диапазоне $\{0,75; 1\}$, если вектор попадает в четкий кластер или берется случайно из диапазона $\{0; 0,25\}$, если вектор не попадает в четкий кластер.

Шаг 4. Нечеткая кластеризация Густафсона–Кесселя. Корректировка значений нечеткой матрицы, матриц ковариации и центров нечетких кластеров.

Шаг 5. Формирование базы правил системы логического вывода. Каждый кластер описывает одно правило вывода. ФП терм-множеств, используемых в правиле для каждой входной и выходной переменных, получают следующим образом. Из векторов, попавших в кластер со степенью принадлежности больше 0, берутся значения компоненты, для которой строится ФП. Формируется матрица, состоящая из элементов $(x_k, \mu(x_k))$, где x_k – значение компоненты вектора, а $\mu(x_k)$ – значение степени принадлежности вектора к данному кластеру, k – индекс компоненты вектора. Матрица используется в качестве обучающей выборки, используемой для поиска оптимального типа ФП для данного терм-множества. Формируется набор хромосом для разных типов ФП и запускается алгоритм поиска оптимального типа по критерию минимизации среднеквадратичной ошибки. Для значения центра ФП берется значение компоненты k для центра нечеткого кластера, для которого определяется эта ФП. Значения остальных параметров ФП устанавливаются случайным образом. Степень варьирования этих параметров зависит от влияния последних на вид ФП. ФП, обладающая минимальной ошибкой на обучающем наборе, берется за исходную.

Шаг 6. Настройка функций принадлежности для входных и выходных переменных системы нечеткого вывода. Совместная работа правил вывод требует дополнительной корректировки, т.е. такие параметры, как отклонение, ширина центра и т.д. в окончательном варианте могут незначительно измениться. На предыдущем шаге был детектирован тип ФП для каждого терм-множества. Центры ФП рассчитаны алгоритмом нечеткой кластеризации. Теперь для настройки всей нечеткой системы генерируется популяция хромосом с закодированными параметрами всех обнаруженных ФП. Генетический алгоритм работает в целях минимизации среднеквадратичной ошибки модели: значение реального выхода модели сравнивается с идеальным в первоначальной выборке. Алгоритм

останавливается при достижении предельного числа итераций либо при получении удовлетворительной ошибки.

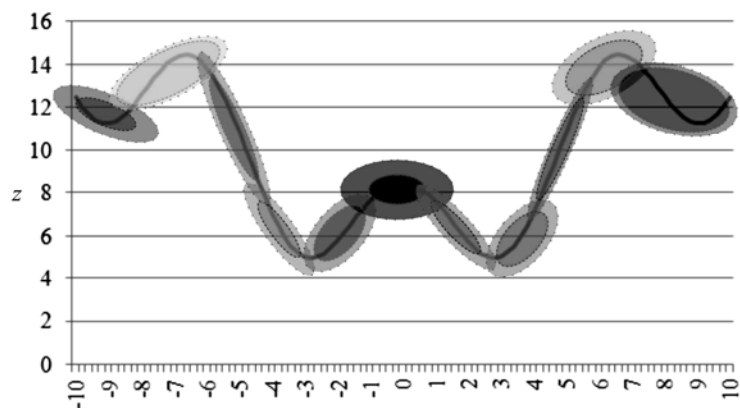
Эксперимент. Для оценки эффективности предлагаемого алгоритма была использована задача по восстановлению поверхности, заданной уравнением $z = f(x, y) = \sqrt{x^2 + y^2} + 3\cos(\sqrt{x^2 + y^2}) + 5$ в диапазонах $x = \{-10; 10\}$ и $y = \{-10; 10\}$. Входными значениями для алгоритма являлись значения 1000 точек, расположенных на поверхности $f(x, y)$, требовалось построить систему нечеткого вида типа Мамдани, позволяющую производить аппроксимацию функции по предложенному алгоритму и оценить ошибку аппроксимации модели.

Для эксперимента были заданы следующие условия: число кластеров NC для определения их оптимального числа в алгоритме Кохонена изменялось в диапазоне от 5 до 50. Поиск кластеризации при каждом заданном числе кластеров выполнялся 20 раз, т.е. производился анализ $(50-5+1) \times 20$ конкурирующих кластеризаций, для каждой из которых вычислялся индекс Силуэта [5]. Решение, обладающее максимальным значением индекса, выбиралось в качестве итогового результата. В качестве ФП для терм-множеств переменных x, y, z использовались гауссова, трапециевидная и колоколообразная ФП. Генетический алгоритм для настройки всей модели запускался со следующими параметрами:

- начальная популяция – 1000 хромосом,
- количество потомков – 3000 хромосом,
- кроссовер – одноточечный,
- процент мутации – 0,05.

Тестирование производилось на аппаратной базе CPU Intel Core 2DUO T8100 2x2.1 GHz, среднее время моделирования (тестировалось пять раз) алгоритма составило около 17 мин. В результате алгоритм детектировал 34 нечетких кластера. На рис. 6 показаны проекция поверхности при значении $x = 0$ и $y = \{10; 10\}$ и распределение нечетких кластеров вдоль заданной функции.

Рис. 6. Проекция поверхности при $x = 0$ и $y = \{10; 10\}$ и нечеткие кластеры, обнаруженные в результате нечеткой кластеризации. Более темные со степенями принадлежности – «внутренние» эллипсы указывают области точек с границей $\mu = 0,75$ и более светлые – «внешние» эллипсы с $\mu = 0,5$



На рис. 7 показаны идеальная поверхность и полученная в результате восстановления по сгенерированной базе правил.

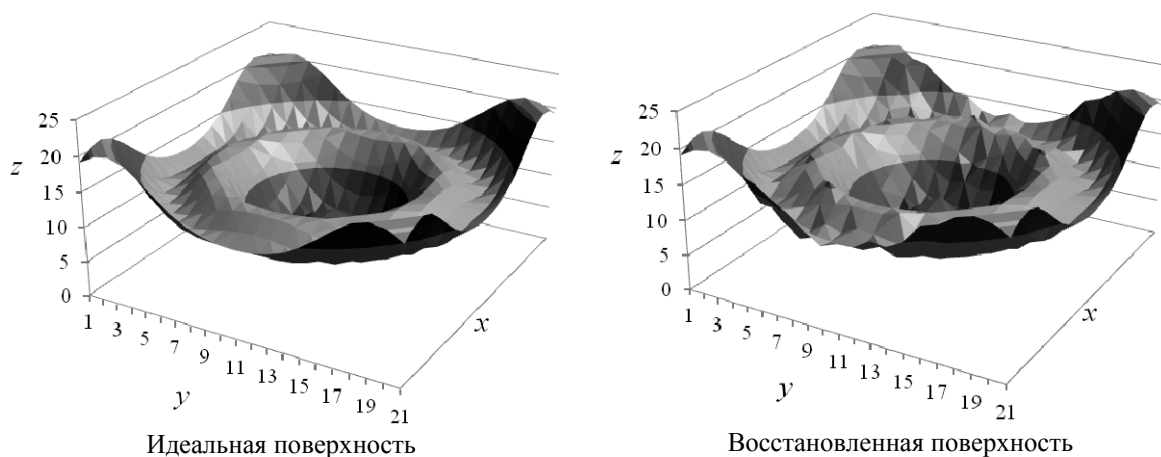


Рис. 7. Исходная и восстановленная поверхности

Средняя квадратичная ошибка модели при тесте из 1000 точек, случайным образом взятых с идеальной поверхности, составила 7,4%.

Заключение. В работе был предложен метод автоматического поиска структуры и параметров нечеткой системы логического вывода Мамдани с помощью анализа плотности распределения примеров обучающей выборки. Алгоритм позволяет выявлять нечеткие правила, которые выводятся из примеров с помощью нечеткой кластеризации Густафсона–Кесселя, позволяющей детектировать кластеры гиперэллипсоидной формы. Оценка качества кластеризации с помощью индекса Ситуэта позволяет найти необходимое число сконцентрированных групп точек. Найденные кластеры представляют собой самые общие случаи для данных, что позволяет их использовать для поиска ФП, а взаимозависимость компонент векторов внутри кластеров формирует само правило вывода. Использование генетического алгоритма для исследования пространства возможных ФП входных параметров и выхода заканчивает формирование нечеткой системы. Результаты экспериментов подтверждают возможность использования предлагаемого алгоритма при анализе реальных наборов данных.

Работа выполнена при финансовой поддержке гранта Министерства образования и науки в рамках Госзадания «Наука» № 8.8113.2013 и гранта РФФИ № 13-07-00397а.

Литература

1. Борисов В.В. Нечеткие модели и сети/ В.В. Борисов, В.В. Круглов, А.С. Федулов. – М.: Горячая линия – Телеком, 2007. – 284 с.
2. Fuzzy Cluster Analysis / F. Hoeffner, F. Klawonn, R. Kruse, T. Runkler. – New York: John Wiley&Sons, 2000. – 288 p.
3. Han J. Data mining: Concepts and Techniques/ J. Han, M. Kamber, J. Pei. – Waltham, MA, USA: Morgan Kaufmann, 2012. – 703 p.
4. Larose D.T. Discovering Knowledge in Data. – New York: John Wiley&Sons, 2005. – 217 p.
5. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. – М.: Горячая линия – Телеком, 2006. – 383 с.

Силич Виктор Алексеевич

Д-р техн. наук, профессор каф. оптимизации систем управления НИ ТПУ

Тел.: +7-913-825-7862

Эл. почта: vas@tpu.ru

Силич Мария Петровна

Д-р техн. наук, профессор каф. автоматизации обработки информации ТУСУРа

Тел.: 8 (382-2) 70-15-91

Эл. почта: smp@muma.tusur.ru

Аксёнов Сергей Владимирович

Канд. техн. наук, доцент каф. оптимизации систем управления НИ ТПУ

Тел.: 8 (382-2) 42-04-59

Эл. почта: axonovsw@tpu.ru

Silich V.A., Silich M.P., Axonov S.V.

A Mamdani-type fuzzy system construction algorithm based on training vectors density analysis

The article presents an algorithm for automatic Mamdani-type fuzzy inference system configuration for approximation task. The constructed inference systems uses the optimal fuzzy hyperellipse-shaped clusters search and allows to clarify the inference steps. The construction is based on fuzzy Gusstaffson-Kessel clustering procedure and detects a set of clusters for all training vectors. The clustering results und later genetic algorithm processing allow to find the set of inference rules, types of fuzzy sets and all membership functions factors. The analysis of efficiency of proposed algorithm was performed with the complex surface recovery task.

Keywords: Mamdani-type fuzzy inference system, fuzzy clustering, genetic algorithm.