УДК 681.3

С.В. Хмелевой, Ю.А. Скобцов, А.М. Фонотов

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ДЛЯ СОЗДАНИЯ БАЗЫ ЗНАНИЙ НА ОСНОВЕ НЕЧЕТКИХ ПРОДУКЦИЙ, НАСТРАИВАЕМЫХ С ПОМОЩЬЮ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

В статье показана разработка инструментальных средств для создания *базы знаний на основе нечетких продукций, создаваемых и настраиваемых генетическими алгоритмами. Разработана иерархия объектов с использованием языка UML, реализованная с использованием интегрированной среды разработки Borland Delphi.*

Введение. Для прогнозирования, решения задач классификации и аппроксимации одним из самых естественным подходов является создание набора правил (продукций). Одним из наиболее эффективных механизмов для автоматического создания такого набора является аппарат генетических алгоритмов (ГА). При создании базы знаний (БЗ) наибольшей гибкостью обладают нечеткие правила. Системы, использующие ГА для построения и настройки набора нечетких продукций, дают достаточно точные и интерпретируемые результаты, получаемые автоматически, без помощи человека.

Целью данной статьи является разработка инструментальных средств для создания базы знаний на основе нечетких продукций.

Анализ исследований и публикаций в данной области. Основой подход для реализации системы по созданию набора правил для классификации/прогнозирования с использованием генетических алгоритмов изложен в обзоре [1]. Он расширен и дополнен как в последующих работах автора [2], так и в других работах [3]. Математический аппарат нечетких множеств создан и получил развитие в работах Л. Задэ, Т. Тэрано, М.Сугэно. Подробно с ними можно ознакомиться, например, в [4]. Применение генетических алгоритмов для автоматического создания и настройки правил на основе нечетких логических контроллеров показано в [5]. Этот подход к формированию БЗ был успешно применен в предыдущих работах авторов [6].

Описание системы для получения базы знаний. В данной работе для автоматического создания и настройки множества правил, представленных с помощью нечетких продукций используется ГА (в качестве алгоритма нечеткого вывода использован алгоритм Mamdani). Полученные правила могут применяться, в том числе, и для прогнозирования временных рядов. В простейшем случае (при использовании лингвистической семантики) используются правила следующего вида:

если x_1 *есть* α_1 и…и x_n *есть* α_n то y *есть* α', где x_1, x_2, …, x_n – входные лингвистические переменные, y – выходная лигвистическая переменная, α_1, α_2, …, α_n, α' - термы из терм-множеств соответствующих лингвистических переменных. Термы всех подзаключений нечетких продукций (отражающие ограничения на входные и выходные переменные) имеют треугольные функции принадлежности.

Однако такой подход имеет ограниченную точность. Для улучшения точности форма функций принадлежности подзаключений задается тремя вещественными числами и может произвольно меняться (в случае использования свободной семантики правил). Это можно реализовать с помощью ГА. При этом потенциальное решение – набор правил в ГА представляется хромосомой, состоящей из двух фрагментов. Первая часть (фрагмент) включает в себя номера термов из терм-

множеств лингвистических переменных. На основе функций принадлежности этих термов строится изначально набор правил. Вторая часть содержит более точный вид правила в виде набора закодированных функций принадлежности (каждая кодируется тремя числами, всего n входных и 1 выходная переменная).

Таким образом, хромосома имеет вид $C=C_1C_2$, где $C_1= C_{11},C_{21},\ldots,C_{n1},C_y$ – выбор конкретного терма лингвистической переменной для каждого входа и выхода, $C_2=C_{a12}\ C_{b12}\ C_{c12}\ C_{a22}\ C_{b22}\ C_{c22}\ldots\ C_{an2}\ C_{bn2}\ C_{cn2}\ C_{ay2}\ C_{by2}\ C_{cy2}$ – точная настройка нечетких функций правила (абсциссы левого края треугольника, его центра и его правого края). С помощью фрагмента хромосомы C_1 выполняется грубая настройка правила, в то время как фрагмент C_2 используется для точной настройки.

Еще один подход для улучшения точности создаваемых правил является создание двухуровневого генетического алгоритма [**Ошибка! Источник ссылки не найден.**], функцией нижнего уровня является конструирование нечетких правил, функцией верхнего – сбор созданных правил и выдача набора данных, по которому нижний уровень будет создавать следующее правило.

Создание популяции нижним уровнем генетического алгоритма.

Правила в начальную популяцию генерируются частично случайно, частично используя точки из ОВ как базу для создания. Треть правил создаются, только подбирая значения части С1 хромосомы (используются термы стандартных ЛП, лучше соответствующие точкам, часть С2 заполняется из ФП терма). Вторая треть правил создается, подбирая терм лингвистической переменной частью С1 хромосомы, создавая часть С2 случайно в разрешенных для неё пределах. Для остальных правил выбор индекса С1 случаен, часть С2 заполняется случайно в разрешенных пределах.

Затем начинается работа нижнего уровня ГА. Результатом работы нижнего ГА является лучшее правило, которое копируется в массив правил БЗ.

Функции верхнего уровня эволюционного алгоритма.

После этого верхний уровень алгоритма начинает новый нижний ГА, создающий новую особь. Остановка процесса генерации правил (генерации знаний) производится после достаточного описания правилами БЗ всех точек ОВ. Для этого ОВ делится на две части: уже описанную созданными правилами и еще ими не описанную. При создании правила, описывающего (покрывающего) данную точку переносят точку из второй подвыборки в первую при однократном [3] или множественном [5] вхождении точки. Тогда для точки e_l считается количество вхождений в различные правила - пересчитывается величина покрытия CV каждой точки по выражению:

$$CV_R(e_l) = \sum_{i=1}^{T} R_i(e_l),\ R_i(e_l) = *(A_i^1(ex_1^l),\ldots A_i^n(ex_n^l)B_i(ey^l))$$

где R_i - i-е правило,

T – общее число правил;

A_i, B_i – подусловия правила.

При достижении значением $CV_R(e_l)$ определенного порога (например, 1.5) она удаляется из множества обучающих точек. Т.о., вновь создаваемые правила не учитывают достаточно покрытые точки, и будут описывать более редкие ситуации.

Результатом работы алгоритма является набор правил, по которым можно получить прогнозное/аппроксимированное значение на основании входных факторов. Набор правил записывается в БЗ и может быть интерпретирован словесно [4].

Разработка инструментальных средств, реализующих данный алгоритм.

Наиболее рационально составить программу решения задачи прогнозирования из следующих программных модулей: модуля ввода информации, модуля получения базы знаний, а также модуля управления и визуализации.

Все модули программного комплекса реализованы с использованием средств Borland Delphi 7 [8]. Проектирование всех программных модулей выполнялись с использованием объектно-ориентированного подхода, предусматривающего создание классов для каждой решаемой задачи.

В простейшем случае для *ввода информации* используют стандартные текстовые файлы – benchmark-и. Достоинством их использования является то, что полученные результаты можно сравнить с результатами, полученными другими исследователями на этих наборах.

Для *создания базы знаний* на основе нечетких логических контроллеров необходима создание иерархии объектов. Разработанная диаграмма классов в терминах UML [7] представлена на рис. 1.

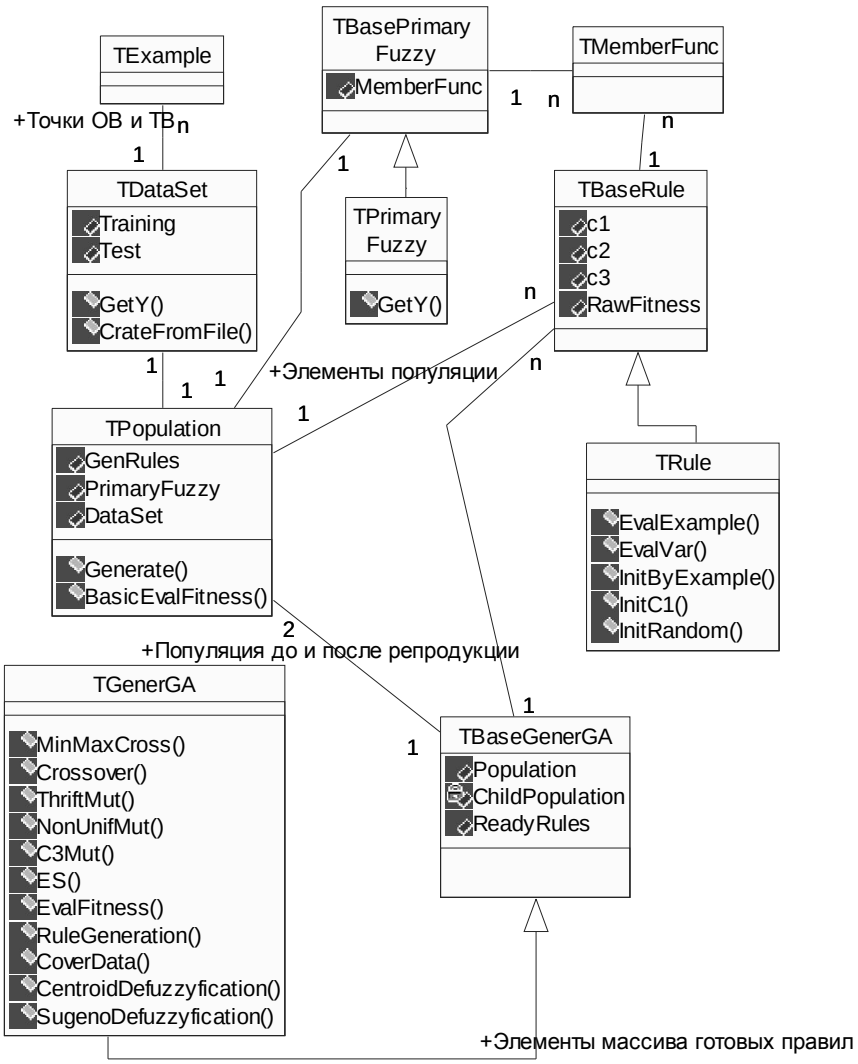

Рис.1. Разработанная диаграмма классов объектов

Для создания базы правил предусмотрены следующие типы и классы:

- TExample: одна точка из ОВ или ТВ;
- TMemberFunc: одно условие правила (или терм);
- TRule: одно правило;
- TPrimaryFuzzy: объект для терм-множества лингвистических переменных;
- TDataSet: класс для множества обучающих и тестовых данных;
- TPopulation: класс для нижнего уровня генетического алгоритма (содержащий популяции правил);
- TGenerGa: верхний уровень генетического алгоритма создания базы знаний.

Объекты TRule, TPrimaryFuzzy, TGenerGa расчитаны на треугольную функцию принадлежности. Для возможности её модификации предусмотрено создание абстрактных классов TBaseRule, TBasePrimaryFuzzy, TBaseGenerGa, на которых могут основываться и классы-потомки с другими функциями принадлежности.

Подсистема визуализации. Из всех видов применяемой для данной системы видов визуализации можно выделить оценку процесса получения результата по набору входных точек. Она весьма полезна на этапе разработки инструментальных средств для проверки правильности результата, а также может быть применена для демонстрации механизма действия системы нечеткого вывода посторонним лицам. Экранная форма визуализации процесса получения результата по набору правил представлена на рис.2.

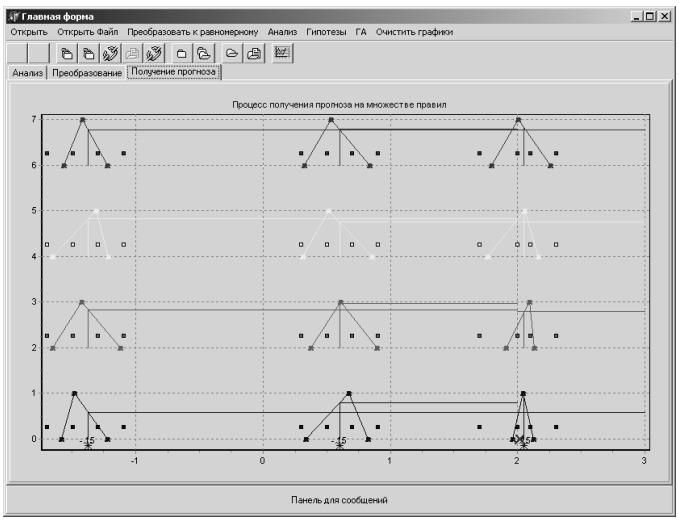

рис. 2. Экранная форма получения прогноза для 2 входных факторов

Для тестирования использовались наборы из Proben1 [9] и UCI Repository [10]. Для Glass1 (Proben1) сравнение выполнялось с результатами, полученными нейронными сетями на несокращенном множестве факторов (НС) и сокращенном с помощью генетических алгоритмов (ГА+НС), показанными в документации [9], а также с [3, 11, 12]. Для данных Ionosphere (UCI), результаты сравнивались с [3, 11, 12, 13, 15, 16] В [11, 12] использовалась комбинация ГА+НС, в [3] – C4.5 и EDRL, в [13] - система нечеткого вывода (FRBS, fuzzy rule-based system) на основе гауссовских функций принадлежности свободной семантики, в [14] – FRBS, подобная тестируемой, в [15] – алгоритмы SLAVE, C4.5, BP, в [16] – алгоритм SLAVE для генерации нечетких правил. Производилось сравнение и с результатами, полученными НС, и комбинацией «Анализ главных компонент-нейронная сеть» («АГК-НС»), реализация в пакете Matlab.

Таблица 1. Исследование на наборе Glass.

9+6, 214зап 0,0,1,1,0,0,0,1,0	Прав	ОВ	СКО	ТВ	СКО	Ош. класс ОВ	СКО	Ош. класс ТВ	Точн. класс. ТВ	СКО
ГА+НС		5.7		10.7		8.16		9.68	**90.32**	
Лингв. генер.	63.25/4	6.51	0.20	8.90	2.05	6.54	7.20	10.89	**89.11**	8.71
АГК+НС		0.77	0.65	17.5	10.12	0.34	0.61	15.22	84.78	10.8
Proben1 [9] linear		8.83	0.01	9.98	0.1			46.04	53.96	2.21
Proben1 [9] mult 8+0+l						9.184		32.08	67.92	
Proben1 [9] pivot		7.68	0.79			9.75	0.41	39.03	61.97	8.14
Yang [12]									70.5	8.5
Yang [12] сокр.									80.8	5
Oh [11]									100	
Kwedlo [3] C4.5									67.5	0.8
Kwedlo [3] EDRL									66.7	1

Таблица 2. Исследование на наборе Ionosphere

Факторы: 1,0,0,0,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

Ionosphere 34+1, 351зап	Прав.	СКО	ОВ	СКО	ТВ	СКО	Ош. класс ОВ	СКО	Ош. класс ТВ	Точн. класс. ТВ	СКО
НС+ГА			0.30		1.40		0.35		0.61	**99.39**	
Своб. генер.	111.03/4	17.68	0.16	0.02	0.48	0.21	1.00	1.73	0.10	99.90	0.08
Лингв. генер.	105.3/4	30.24	0.04	0.03	1.15	0.25	0.00	0.00	1.71	**98.29**	1.21
Surmann [13]	2/35						99.34		2.00	98	
Castillo [16]	4/?								7.12	92.88	

Таблица 3. Исследование на наборе Iris.

Iris 4+3, 150зап	Прав.	СКО	ОВ	СКО	ТВ	СКО	Ош. класс ОВ	СКО	Ош. класс ТВ	Точн класс ТВ	СКО
НС			2.29	2.60	3.79	4.13	1.86	2.64	3.38	**96.62**	4.79
Лингв. генер.	88.67/4	6.81	0.98	1.34	1.65	0.32	0.87	1.53	2.16	97.84	1.15
Лингв. мульт.	68.5/4	3.21	0.50	0.02	1.06	0.35	0.33	0.55	1.08	**99.20**	1.31
Kwedlo [3], C4.5										95.2	0.2
Kwedlo [3], EDRL										96	0
Cordon [14], FRBS					2.69				5.68	94.32	
Surmann [13], FRBS										97.33	
Cordon [15], SLAVE										95.43	
Cordon [15], C4.5										91.13	
Cordon [15], BP										91.56	

На этапе генерации знаний система нечеткого вывода обучалась 10 раз для каждого набора, отдельно для каждого класса (например, для Glass1 – 10*6=60 обучений). Сначала делалась попытка использовать лингвистические средства алгоритма и стандартные функции принадлежности («лингв.»),а в случае

недостаточной точности использовались функции принадлежности произвольной формы (свободная семантика, «своб.»).

Как показали опыты, с помощью созданных инструментальных средств возможно получить результаты, зачастую превосходящие результаты других исследований.

Выводы

По результатам, описанным в данной статье, можно сделать выводы:

1. Спроектирована иерархия объектов для объектно-ориентированной программной реализации системы, реализующей автоматическое построения базы знаний на основе нечетких продукций с помощью генетических алгоритмов.
2. По разработанным алгоритмам и спроектированной иерархии объектов созданы инструментальные средства для решения задачи прогнозирования ВР с помощью набора нечетких продукций. Их реализация была успешно произведена с использованием интегрированной среды разработки Borland Delphi 7.
3. Выполнено тестирование созданных аппаратных средств. Результаты тестирования показали успешность реализации созданных инструментальных средств.

ЛИТЕРАТУРА:

1. Freitas Alex, 2002. A survey of evolutionary algorithms for data mining and knowledge discovery. Springer-Verlag. In A. Ghosh and S. Tsutsui editors, Advances in Evolutionary Computation, chapter 33, pages 819-845.
2. Dieferson Luis Alves de Araujo, Heitor S. Lopes, Alex A. Freitas. A parallel genetic algorithm for Rule Discovery in Large Databases. Proc 1999 IEEE Systems, Man and Cybernetics Conf., v.3, 940-945. Tokyo, 1999.
3. Wojciech Kwedlo and Marek Kretowski, 199. Discowery of Decision Rules from Databases: An Evolutionary Approach. Proc. 2nd European Symp. on principles of Data Myning and Knowledge Discovery (PKDD-98). Lecture Notes in Artificial Intelligence 1510, 371-378. Springer-Verlag, 1998.
4. А.Леоненков. Нечеткое моделирование в среде Matlab и fuzzyTECH. СПб.: БХВ-Петербург, 2005. – 736с.: ил.
5. O.Cordon, F.Herrera. A Three-Stage Evolutionary Process for Learning Descriptive and Approximative Fuzzy Logic Controller Knowledge Bases from Examples.// International Journal of Approximate Reasoning Vo. 17-4 (1997) 369-407.
6. Хмелевой С.В. Создание и применение базы знаний на основе аппроксимативных нечетких логических контроллеров для прогнозирования internet траффика. // Наукові праці ДонНТУ Серія: “Обчислювальна техніка та автоматизація” Випуск 13 (121). -Донецьк : ДонНТУ. – 2007.- 226с. с.132-139.
7. Фаулер М. Скотт К. UML. Основы. – Пер. с англ. – СПб: Символ-Плюс, 2002. – 192 с., ил.
8. Архангельский А. Я.. Программирование в Delphi 7. – СПб: «Бином-Пресс», 2003. - 1152 с.
9. PROBEN1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules [Electronic resource] : Сборник тестовых наборов данных для нейронных сетей / Lutz Prechelt, Fakultaet fuer Informatic, Universitaet Karlsruhe. –

Электрон. дан. – Karlsruhe, Germany, 1994. – Режим доступа: ftp.ira.uka.de/pub/neuron. - Загл. с экрана.

10. UCI Machine Learning: Сборник тестовых наборов данных. – Электрон. дан. - University of California, Irvine . – Режим доступа: http://mlearn.ics.uci.edu/databases . - Загл. с экрана.

11. *Oh II-Seek, Lee Jin-Seon, And Moon Byung-Ro,* 2004. Hybrid Genetic Algorithms for Feature Selection. IEEE Transactions on pattern analysis and machine intelligence, vol 26, no.11.

12. *Yang J.H., Honawar V.,* 1998. Feature Subset Selection Using a Genetic Algorithms. IEEE Intelligent Systems, vol. 13, no. 2, pp. 44-49.

13. *H.Surmann, A.Selenschtschikow.* Automatic Generation of fuzzy logic rule bases: Examples I. – Proc. of the NF2002: first international ICSC conference on neuro-fuzzy technologies. Pp 75-81, CUBA 16-19 jan, 2002.

14. *Oscar Cordon, Francisco Herrera et al.*. Different proposals to improve the accuracy of fuzzy linguistic modeling. //http://decsai.ugr.es/~ocordon/

15. *O. Cordon, M.J. del Jesus and F. Herrera.* Evolutionary approaches to the learning of fuzzy rule based classification systems. //http://decsai.ugr.es/~herrera/public

16. *Luis Castillo, Antonio Gonzalez, and Raul Perez,* "Includinga simplicity criterion in the selection of the best rule in a genetic fuzzy learning algorithm," Fuzzy Sets and Systems, vol. 120, pp. 309 – 321, 2001.