



OSTIS-2011

(Open Semantic Technologies for Intelligent Systems)

УДК 004.8

ЭВОЛЮЦИОННЫЕ МЕТОДЫ ФОРМИРОВАНИЯ НЕЧЕТКИХ БАЗ ПРАВИЛ

Л.Г. Комарцова (*lkomartsova@yandex.ru*)

*Московский государственный технический университет им. Н.Э. Баумана
(Калужский филиал), г.Калуга, Россия*

В докладе рассматриваются проблемы, связанные с обработкой неполной, неточной и недостоверной информации в системах искусственного интеллекта. Применение новых подходов на основе эволюционных методов для оптимизации нечетких баз правил позволит значительно снизить время формирования решения и повысить достоверность его принятия в интеллектуальных системах.

Ключевые слова: интеллектуальные системы, нечеткие базы правил, эволюционные методы.

Введение

Развитие систем искусственного интеллекта (СИИ) в настоящее время связано с созданием новых методов и технологий, направленных на решение сложных проблем, таких, как предсказание, планирование, распознавание образов и т.д. в различных прикладных областях. Наиболее перспективным направлением исследований в этой области является создание гибридных СИИ, позволяющих наиболее эффективно использовать преимущества каждой из интеллектуальных технологий для решения сложных проблем науки и техники. Из практики известно, что целенаправленное, устойчивое к флуктуациям поведение интеллектуальных систем определяется двумя основными факторами [1, 4]:

- цельный интеллект имеет распределенную структуру и возникает из взаимодействия множества простых, отдельных, агентских (модульных) интеллектов;
- развитие интеллектуальных моделей опирается на дарвиновский принцип естественного отбора как на основной механизм, формирующий поведение отдельных агентов и обеспечивающий их адаптацию к внешней среде.

В соответствии с этим в целях повышения эффективности использования СИИ для решения сложных интеллектуальных задач необходимо использовать модели, которые могут развиваться во времени с целью адаптации к среде функционирования. В докладе рассматриваются проблемы создания таких моделей на основе технологий нечетких систем и эволюционных методов [2, 3].

1. Проблема минимизации нечетких баз правил

Для построения нечетких систем необходимо решить важную задачу нахождения подходящего ряда нечетких правил, отображающих специфику предметной области. Эта проблема может быть решена двумя способами. Первый заключается в извлечении необходимых знаний непосредственно из экспертов определенной предметной области и трансляции этих знаний в нечеткие правила. Однако, это трудный путь, поскольку эксперты, как правило, не знают нечеткой теории, а инженерные знания не всегда адекватны действительности.

Второй способ связан с автоматическим извлечением знаний из данных, которые представляют собой примеры решения конкретных задач. В настоящее время существует

довольно большое число методов, позволяющих автоматически извлекать знания из данных: нейронные сети, алгоритмы классификации и кластеризации, эволюционные методы и генетические алгоритмы [2, 3].

В докладе рассматриваются проблемы минимизации нечеткой базы правил на основе генетического алгоритма, что имеет большое значение для практической реализации и дальнейшего развития гибридных технологий, используемых в интеллектуальных системах.

Нечеткие системы являются фундаментальной технологией представления и обработки нечеткой информации, которая может быть неполной, неточной и недостоверной. В настоящее время этот тип систем широко применяется в системах контроля, для решения задач моделирования, классификации и т.д. [1]. Одним из центральных моментов при формировании нечетких баз знаний (БЗ) в интеллектуальных системах является проблема устранения их избыточности для повышения достоверности вывода и сокращения времени просмотра правил при выводе.

Минимизация БЗ может быть поставлена как комбинаторная оптимизационная задача для одновременного достижения двух целей: минимизации ошибки распознавания и количества нечетких правил "IF ... THEN" в БЗ. Для решения поставленной задачи в работе используются генетические алгоритмы. Генетические методы оптимизации БЗ можно разбить на три типа [1,3]: Мичиганский, Питтсбургский и итеративный. В первом подходе каждая хромосома соответствует единственному правилу, во втором - в каждой хромосоме содержится ряд правил, в третьем – каждая хромосома кодирует одно правило (как в первом случае), но только лучшее правило из популяции возвращается как решение, остальные удаляются.

В работе исследован первый подход для обучения и оптимизации базы правил, поскольку в этом случае отбор необходимых правил осуществляется между популяциями, а не внутри одной популяции, что позволяет увеличить пространство поиска лучших решений.

Реализация этого подхода выполняется в два этапа:

- на первом этапе генетический алгоритм (ГА) используется для нахождения кандидатов нечетких правил в БЗ;
- на втором этапе с помощью ГА выполняется процедура оптимизации БЗ с целью исключения из рассмотрения худших правил, мало влияющих на получение окончательного решения.

2. Процедура обучения нечеткой базы правил на основе генетического алгоритма

В этой процедуре ГА используется для генерации подходящего множества нечетких правил, корректно описывающих предметную область, при этом применяется базовый ГА, в котором последовательно выполняются следующие действия: кодирование, формирование начальной популяции, оценка хромосом на основе функции фитнеса (Fit), применение генетических операторов, проверка условия останова.

Кодирование. Поскольку обучение основывается на Питтсбургском подходе, то каждая хромосома полностью кодирует нечеткую базу знаний. Правила кодируются целыми числами, которые представляют индекс нечетких множеств, присутствующих в антецеденте и консеквенте правила. Число 0 идентифицирует отсутствие «условия» или «заклЮчения» и представляется как «неизвестно». Рассмотрим пример кодирования БЗ. Предположим, что нечеткая база правил состоит из k правил:

$$R_1 : IF X_1 \text{ is } A_{21} \text{ and } X_2 \text{ is } A_{32} \text{ and } X_3 \text{ is "don't know"} THEN Class = C_1$$

$$R_2 : IF X_1 \text{ is } A_{31} \text{ and } X_2 \text{ is } A_{12} \text{ and } X_3 \text{ is } A_{13} THEN Class = C_2$$

...

$$R_k : IF X_1 \text{ is "don't know"} \text{ and } X_2 \text{ is } A_{k2} \text{ and } X_3 \text{ is } A_{k1} THEN Class = C_3$$

В этом случае хромосома H_i (вариант нечеткой БЗ) кодируется следующим образом:

$$H_i: \underline{2\ 3\ 0\ 1\ (R_1)}\ \underline{3\ 1\ 1\ 2\ (R_2)}\dots\ \underline{0\ 2\ 1\ 3\ (R_k)}$$

Начальная популяция генерируется на основе предварительного исследования предметной области, применения случайной стратегии и формируется из чисел, которые кодируют принадлежность входных данных нечетким множествам.

Функция фитнеса (Fit) определяется как число правильно распознанных образов или ситуаций нечеткой базой знаний и вычисляется по формуле:

$$Fit(H_i) = NCS(H_i), \quad (1)$$

где NCS (*Number Correctly Samples*) – число правильно распознанных базой правил H_i примеров. Необходимо максимизировать формулу (1) для получения наибольшего значения функции Fit .

На первом этапе формирования оптимальной БЗ в силу недостаточной структурированности информации используются стандартные параметры ГА: кроссинговер – одноточечный, мутация – генная, селекция – случайный выбор хромосом для скрещивания, отбор – элитная стратегия.

3. Решение задачи оптимизации нечеткой базы правил

На этом этапе с помощью генетического алгоритма осуществляется редукция сформированной БЗ. Для этого выполняются следующие действия.

Кодирование. Хромосомы кодируются двоичными числами длины m , где m – число правил, сгенерированных на предыдущем этапе. Каждый ген h_{ij} – бит, связанный с одним правилом. Если $h_{ij}=1$, то правило активно. Пример кодирования нечеткой БЗ:

$$\begin{array}{cccccc} H_i & 1 & 0 & 0 & 1 & 1 \\ & R_1 & R_2 & R_3 & R_4 & R_5 \end{array}$$

Начальная популяция генерируется на основе хромосомы, которая включает все предварительно полученные правила.

Функция фитнеса оценивает каждую хромосому, т.е. каждое выбираемое подмножество правил-кандидатов, на основе двух критериев: количестве правильно классифицируемых примеров $NCS(H_i)$ и числе активных правил, содержащихся в этом подмножестве примеров $NR(H_i) - (Number\ Rule)$. Фитнесс - функция вычисляется следующим образом:

$$Maximize(Fit(H_i) = NCS(H_i) * (R - NR(H_i))), \quad (2)$$

где R – общее число правил.

При вычислении H_i определяется, все ли обучающие примеры классифицируются подмножеством правил, закодированных в H_i . Если нечеткое правило в H_i не классифицирует какой-либо образ, то это правило удаляется из подмножества правил путем присваивания соответствующему гену нулевого значения. Таким образом, происходит редукция БЗ, а число активных правил запоминается в $NR(H_i)$.

3. Разработка самоадаптивного генетического алгоритма

Использование ГА для решения прикладных задач требует выбора ряда параметров (размера популяции, типов операторов селекции, отбора и т.д.), которые оказывают большое влияние на возможности алгоритма эффективно исследовать все поисковое пространство [2, 3]. Для определения оптимального значения этих параметров необходимо провести большое число экспериментов. Однако некоторые значения параметров будут являться оптимальными для одной предметной области и неоптимальными для другой. Неадекватные значения параметров ГА могут привести к преждевременной сходимости алгоритма, например, из-за недостатка генетического разнообразия популяции. Важную роль в этом процессе играют операторы кроссинговера и мутации: мутация вводит новые хромосомы, а кроссинговер изменяет генетический материал на основе обмена генами между хромосомами.

В данной работе применяется динамическое изменение процента хромосом, подвергающихся мутации и скрещивания, базирующееся на текущей информации о поисковом пространстве с тем, чтобы поддерживать генетическое разнообразие в популяции, поскольку алгоритм с фиксированными параметрами быстро сходится в точку локального оптимума. Введем коэффициент k , оценивающий генетическое разнообразие в популяции, который вычисляется на основе использования отношения среднего $Fit_{cp.}$ и Fit_{max} в каждом поколении, изменяясь в диапазоне $[0, 1]$.

Когда k близко к 1, это означает, что генетическое разнообразие мало, поэтому для предотвращения преждевременной сходимости должен увеличиваться процент хромосом, подвергающийся мутации, а кроссинговера снижаться. С другой стороны, если k близко к нулю, то генетическое разнообразие увеличилось благодаря использованию оператора мутации, поэтому процент мутации должен снижаться, а кроссинговера увеличиваться.

Рассмотренная процедура может быть представлена следующим образом на языке псевдокода

```

If  $k \geq V_{max}$ 
{
    проц_мут =  $\min(\text{проц\_мут}) * v, \text{max\_мут}$ ;
    проц_крос =  $\max(\text{проц\_крос} / v, \text{min\_крос})$ 
}
else if  $k \leq V_{min}$ 
{
    проц_мут =  $\max(\text{проц\_мут} / v, \text{min\_мут})$ ;
    проц_крос =  $\min(\text{проц\_крос} * v, \text{max\_крос})$ ;
}

```

Параметры V_{max} и V_{min} определяют границы изменения коэффициента k , а $v \geq 1$ – константа, min_мут и max_мут – границы изменения процента мутации, min_крос и max_крос – границы изменения процента кроссинговера. В конкретной реализации параметры алгоритма подбираются для каждой конкретной задачи.

Заключение

В работе предложена методика автоматического обучения и редукции правил нечетких баз знаний на основе генетического алгоритма, которая позволяет оптимизировать базы нечетких правил, что, в свою очередь приведет к повышению эффективности работы систем интеллектуальных систем.

Библиографический список

1. Kasabov N. Song Q. DENFIS: Dynamic evolving neuro-fuzzy inference system and its application for time-series prediction. –IEEE. Trans. Fuzzy system, 10(2).-pp. 144-154.
2. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. - М. –Физматлит. -2006.
3. Комарцова Л.Г., Особенности построения эволюционных систем на основе генетических алгоритмов.1 Межд. Конф. САИТ-2005–М.-2005. С.201-204.
4. Люгер Дж. Искусственный интеллект. Стратегии и методы решения сложных проблем.- Пер. с англ. – М.: Изд дом «Вильямс».-2003.
5. Cordon O., Herrera F. A proposal on reasoning methods in fuzzy rule-based classification systems. International Journal of Approximate Reasoning.- vol.20, pp21-45-2001.