Знания-Онтологии-Теории (ЗОНТ-09)

Алгоритмы обучения нечетких систем на основе методов муравьиной колонии и роящихся частиц

И.А. Ходашинский, П.А. Дудин, Д.С. Синьков

Томский государственный университет систем управления и радиоэлектроники, пр. Ленина, д. 40, г. Томск, 634050, Россия.
hodashn@rambler.ru

Аннотация. *Обсуждаются вопросы построения нечетких систем на основе таблиц наблюдений. В качестве оптимизационных процедур предложены дискретный, непрерывный и прямой алгоритмы муравьиной колонии, а также алгоритм роящихся частиц. Представлены результаты имитационных экспериментов.*

Ключевые слова: нечеткие системы, обучение, роевой интеллект, алгоритмы муравьиной колонии, алгоритм роящихся частиц

1 Введение

Основная задача нечеткого моделирования заключается в нахождении конечного множества локальных отношений вход-выход, которые описывают систему или процесс в виде нечетких «ЕСЛИ-ТО» правил. Обучение нечетких систем включает два основных этапа: определение структуры и оценка параметров. На первом этапе определяются такие характеристики нечеткой системы, как число нечетких правил, количество лингвистических термов, на которое разбиты входные и выходные переменные. Этот этап может быть выполнен с использованием субъективного разделения данных, алгоритма диффузии или нечеткого кластерного анализа [1]. К задачам оценки параметров относятся нахождение оптимальных значений параметров антецедентов и консеквентов правил. Важно отметить, что указанные выше этапы взаимосвязаны и должны выполняться итеративно. Для оценки параметров используются классические методы оптимизации, основанные на производных: метод наименьших квадратов, градиентный метод, фильтр Калмана. Эти методы дают точные результаты, но они имеют тенденцию сходиться к локальным оптимумам. Вторая группа методов – метаэвристические, такие как алгоритмы муравьиной колонии, роящихся частиц, имитации отжига, генетические алгоритмы. Достоинство метаэвристических методов заключается в большей устойчивости. Но это методы грубой настройки, требующие больших временных ресурсов. Кроме того, применение метаэвристик не гарантирует нахождения оптимального решения. Третья группа – гибридные алгоритмы, позволяющие объединить преимущества метаэвристических методов с преимуществами классических методов, основанных на производных. Такое объединение повысит качество решений при умеренном количестве ресурсов и за приемлемое время [2].

2 Постановка задачи

Пусть имеется объект исследования, заданный своей таблицей наблюдений. Проблема исследования обусловлена невозможностью построения аналитической модели изучаемого объекта, либо слишком большой сложностью такой модели, либо отсутствием достаточного опыта для построения экспертных систем, либо недостаточностью экспериментальных данных для статистического моделирования. Решением проблемы может быть переход от аналитических или статистических моделей к нечетким.

Рассматривается нечеткая система типа синглтон, i-ое правило в которой имеет следующий вид:

$$\text{IF } x_1 = A_{1i} \text{ AND } x_2 = A_{2i} \text{ AND } \dots \text{ AND } x_n = A_{ni} \text{ THEN } y = r_i,$$

где A_{ij} – лингвистический терм, которым оценивается переменная x_i; r_i – действительное число, которым оценивается выход y.

Нечеткая система осуществляет отображение $f:\Re^n \to \Re$:

$$f(x) = \sum_{i=1}^{R} \mu_{A_{1i}}(x_1) \cdot \mu_{A_{2i}}(x_2) \cdot \ldots \cdot \mu_{A_{ni}}(x_N) \cdot r_i / \sum_{i=1}^{R} \mu_{A_{1i}}(x_1) \cdot \mu_{A_{2i}}(x_2) \cdot \ldots \cdot \mu_{A_{ni}}(x_n)$$

где $\boldsymbol{x}$ – входной вектор, R – число правил; n – количество входных переменных; $\mu_{A_{ij}}$ – функция принадлежности (ФП), определяемая набором своих параметров, например, треугольная – тремя параметрами, трапециевидная – четырьмя, гауссова и параболическая – двумя.

Нечеткая система может быть представлена как

$$y = f(\boldsymbol{x}, \boldsymbol{\theta}),$$

где $\boldsymbol{\theta} = ||\theta_1, \ldots, \theta_N||$ – вектор параметров, $N = n \cdot$ (число параметров, описывающих одну функцию принадлежности) $\cdot$ (число термов, описывающих одну входную переменную), y – скалярный выход системы.

Пусть дано множество обучающих данных $\{(\boldsymbol{x}_p;\ t_p),\ p = 1, \ldots, m\}$, тогда среднеквадратическая функция ошибки, которую необходимо минимизировать, будет иметь вид

$$E(\theta) = \sum_{p=1}^{m} (t_p - f(x_p, \theta))^2.$$

Для решения проблемы минимизации предлагается использовать алгоритмы муравьиной колонии и роящихся частиц.

3 Алгоритмы муравьиной колонии

3.1 Дискретный алгоритм

Классический алгоритм муравьиной колонии (АМК) [3] — процедура дискретной оптимизации, в то время как параметры ФП меняются непрерывно. Переход от непрерывной оптимизации к дискретной осуществляется посредством построения полного ориентированного графа поиска решения, количество вершин в котором определяется точностью нахождения значений параметров. Из каждой вершины выходят дуги с равномерно распределенными значениями нормированных параметров ФП входных переменных. Во все вершины графа равномерно распределяются муравьи. Цель муравья в задаче идентификации параметров нечеткой системы — посетить столько вершин, сколькими параметрами задается ФП. Пометки дуг, по которым прошел муравей, будут являться найденным им решением. Каждая лингвистическая переменная описана несколькими ФП. Муравьи в алгоритме делятся на колонии. Каждая колония муравьев отвечает за нахождение параметров своей ФП. Количество фермента, наносимого на дуги пропорционально качеству решения, чем меньше ошибка вывода нечеткой системы, выполненного на выбранных параметрах, тем больше фермента наносится на дуги [4].

3.2 Непрерывный алгоритм муравьиной колонии

В АМК для дискретной оптимизации при выборе очередной дуги, муравей «руководствуется» дискретным распределением вероятности. В случае непрерывного алгоритма (НАМК) выбор, который делает муравей, не ограничен конечным множеством, для этого дискретное распределение заменяется на непрерывное, то есть на функцию плотности вероятности (ФПВ).

В НАМК используется ФПВ с Гауссовым ядром [5]. Каждому i-му параметру соответствует свое Гауссово ядро, $i = 1, \ldots, n$, n — число настраиваемых параметров нечеткой модели. Каждая функция $G^i(x)$ описывается тремя векторами: ω — вектор весов, связанных с индивидуальными Гауссовыми функциями, μ^i — вектор математических ожиданий, и σ^i — вектор среднеквадратичных отклонений. Количество элементов всех этих векторов равно числу функций Гаусса, составляющих Гауссово ядро.

В НАМК вводится понятие архива решений. Архив решений представлен таблицей, в которой k строк. Каждая строка представляет собой найденное муравьем решение $\mathbf{s}_l=\{s_l^1, s_l^2,\dots, s_l^n\}$, ошибку НС $f(\mathbf{s}_l)$ и вес решения ω_l.

Следует отметить, что как и в АМК, муравьи в НАМК делятся на колонии, каждая из которых отвечает за нахождение параметров своей ФП. У каждой колонии свой независимый архив решений.

3.3 Прямой алгоритм муравьиной колонии

В прямом алгоритме муравей «отвечает» за вычисление значений закрепленного за ним параметра, поэтому муравьев в алгоритме столько, сколько параметров нечеткой модели [6]. Каждый i-ый муравей создает свое решение, генерируя нормально распределенное действительное число $N(\mu_i, \sigma_i)$. В алгоритме используются два вида феромонов: первый связан с центрами нормальных распределений $\boldsymbol{\mu} = \|\mu_1, \dots, \mu_N\|$, второй с разбросом $\boldsymbol{\sigma} = \|\sigma_1 \dots, \sigma_N\|$.

После того как муравьи нашли решения, определяется испарение феромона. Далее происходит нанесение феромона.

Особенностью прямого алгоритма муравьиной колонии является включение в него простейшего локального поиска, состоящего из двух этапов: на первом значение параметра нечеткой системы θ_j увеличивается с заданным шагом до значения $\theta_j + d_j$, на втором этапе значение параметра уменьшается до значения $\theta_j - d_j$.

Значения этих параметров передаются в нечеткую систему в качестве новых значений параметров ФП. Вычисляется ошибка и лучшее решение текущего шага. Глобальное лучшее решение запоминаются.

Для преодоления локальных минимумов в алгоритме введен параметр конвергенции. Если коэффициент конвергенции становится меньше критического значения cf_r, то вектор $\boldsymbol{\sigma}$ возвращается в начальное состояние.

4 Алгоритм роящихся частиц

Алгоритм роящихся частиц (АРЧ) – это стохастический метод поиска, основанный на итеративном взаимодействии частиц, образующих рой [7]. Каждая частица — это решение, заданное как координаты частицы в многомерном пространстве. Перемещение частицы в пространстве поиска определяют следующие три фактора: инерция, память, сотрудничество. Инерция подразумевает, что частица не может мгновенно изменить свое направление движения. Каждая частица имеет память и хранит свою лучшую позицию в пространстве поиска. Известна частице и лучшая позиция роя. Зная эти две позиции, частица динамически изменяет скорость согласно ее собственному опыту и опыту полета других частиц. Таким образом, движение каждой частицы задается ее лучшей позицией, ее текущей скоростью, ускорением, заданным предыдущей позицией, и ускорением, заданным лучшей частицей в рое. Рой прекращает движение при выполнении хотя бы одного из следующих условий: рой достиг состояния равновесия; найдено оптимальное решение (ошибка меньше заданной); выполнено определенное количество итераций.

Пусть имеется n-мерное пространство поиска $S \subset \Re^n$, рой состоит из N частиц. Позиция i-ой частицы определяется вектором

$$x_i = (x_{i1}, x_{i2}, \dots, x_{in}) \in S\ .$$

Лучшая позиция, которую занимала i-ая частица, определяется вектором

$$p_i = (p_{i1}, p_{i2}, \dots, p_{in}) \in S\ .$$

Скорость частицы определяется также n-местным вектором

$$v_i = (v_{i1}, v_{i2}, \dots, v_{in})\ .$$

Движение частиц определяют простые математические уравнения:

$$v_i(k+1) = w \cdot v_i(k) + c_1 \cdot rand \cdot (p_i(k) - x_i(k)) + c_2 \cdot Rand \cdot (p_g(k) - x_i(k))\ ,$$

$$x_i(k+1) = x_i(k) + v_i(k+1)\ ,$$

где $i = 1, 2, \ldots, N$; $\boldsymbol{v}_i(k)$ — вектор скорости частицы i на итерации k; $\boldsymbol{x}_i(k)$ — координаты частицы i на итерации k; c_1, c_2 —положительные коэффициенты ускорения; $\boldsymbol{p}_i(k)$ — лучшая позиция частицы i на первых k итерациях; $\boldsymbol{p}_g(k)$ — лучшая позиция частицы в рое (задается индексом g) на первых k итерациях; w — эмпирический коэффициент инерции; *rand, Rand* — случайные числа из интервала [0, 1].

Коэффициент c_1 является когнитивным (познавательным) параметром, отражающим доверие частицы к ее собственному прошлому опыту, этот коэффициент ответственен за обнаружение новых областей в пространстве поиска. Коэффициент c_2 является социальным параметром, показывающим насколько частица доверяет рою, этот коэффициент ответственен за исследование окрестностей ранее найденной перспективной области.

Коэффициент инерции ответственен за изменение скорости и управляет обнаружением новых областей и поиском в окрестностях перспективной области.

Процесс модификации решения повторяется многократно, пока не будет выполнено условие останова.

5 Эксперименты

Суть эксперимента заключалась в аппроксимации нечеткой системой тестовой функции $f(x_1, x_2) = x_1{*}\sin(x_2)$. Входные переменные описаны пятью нечеткими термами. База содержит 25 нечетких правил. Термы заданы треугольными ФП.

Были исследованы все алгоритмы муравьиной колонии, для каждого из них были произведены 30 опытов при неизменных параметрах алгоритма. На рисунке 1 представлены средняя динамика изменения ошибки и лучшие результаты оптимизации.

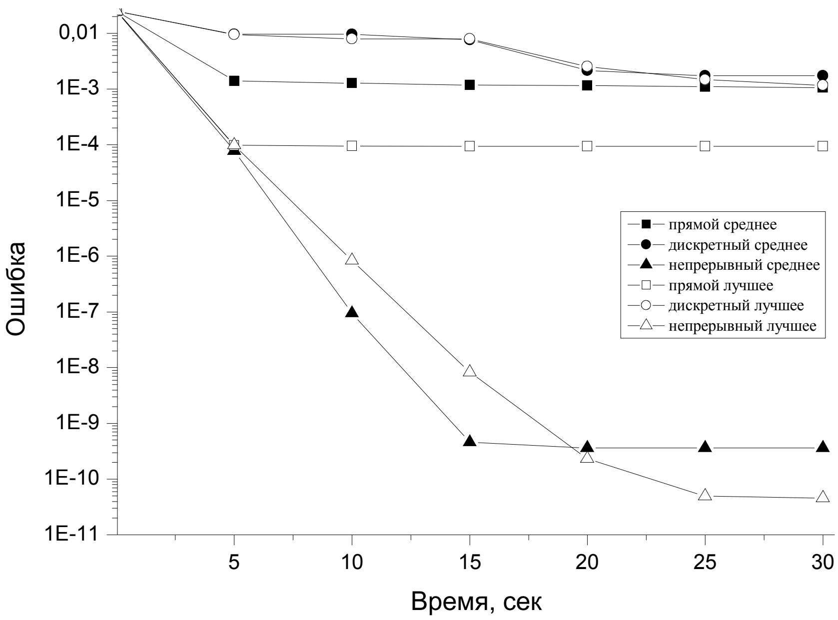

Рис. 1. Результаты работы алгоритмов муравьиной колонии.

В ходе эксперимента с АРЧ изучались индивидуальные траектории оптимизации при неизменном размере популяции, равном 20 особям, в зависимости от числа итераций. Рассмотрим поведение алгоритма в зависимости от коэффициента инерции.

Большие значения w направляют алгоритм на поиск новых перспективных областей, малые — на исследование окрестностей ранее найденных областей.

При $0 < w < 1$ частицы замедляют движение, а скорость сходимости алгоритма определяется значениями коэффициентов c_1 и c_2. При малых значениях c_1 и c_2 наблюдается преждевременная сходимость алгоритма, являющаяся следствием раннего обнуления скоростей частиц. При $w = 0{,}3$; $c_1 = 1{,}4$; $c_2 = 1{,}4$ и популяции размером в 10 особей алгоритм сходится при 200-300 итерациях. При этом найденные решения далеки от оптимальных (ошибка решения равна 0,007-0,07).

При $c_1 > 0$ и $c_2 = 0$ каждая частица ведет независимый локальный поиск, никак не взаимодействуя с остальными частицами роя. Эффективность такого поиска очень низка, алгоритм сходится после нескольких десятков итераций, незначительно улучшив начальное решение. Указанное сочетание параметров непригодно для решения задачи обучения нечетких систем.

При $c_1 = 0$ и $c_2 > 0$ все частицы исследуют окрестности найденного на предыдущих итерациях лучшего решения. В отдельных случаях при таком сочетании параметров ошибка может быть уменьшена на два порядка.

При $c_1 > 0$ и $c_2 > 0$ на направление движения частиц влияют лучшая локальная позиция $\boldsymbol{p}_i(k)$ и лучшая позиция в рое $\boldsymbol{p}_g(k)$. Проведенные имитационные эксперименты позволили определить лучшие параметры для аппроксимации данной функции: $w = 0{,}7$; $c_1 = 1{,}4$; $c_2 = 1{,}4$.

При $w > 1{,}1$ для поведения роя характерны большие скорости и, как следствие, большой разлет частиц и отсутствие сходимости алгоритма. После пятисот итераций отдельные частицы задавали худшее решение с ошибкой превышающей три единицы (лучшее найденное решение на этой итерации имело ошибку 0,0287366), и значения ошибок худших решений не уменьшалась к тысячной итерации. Таким образом, при коэффициенте инерции больше единицы рой не способен ни обнаружить, ни исследовать перспективные области решений.

6 Заключение

Результаты эксперимента показали, что эффективность непрерывного алгоритма муравьиной колонии выше по сравнению с остальными алгоритмами. Эффективность и скорость прямого алгоритма немного выше дискретного алгоритма муравьиной колонии и сопоставима с алгоритмом роящихся частиц.

Работа выполнена при финансовой поддержке РФФИ (грант 09-07-99008).

Литература

[1] Guillaume S.: Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review // *IEEE Transactions on Fuzzy Systems*, 2001. V. 9, N. 3. P. 426-443.

[2] Ходашинский И.А., Гнездилова В.Ю., Дудин П.А., Лавыгина А.В.: Основанные на производных и метаэвристические методы идентификации параметров нечетких моделей // *Труды VIII международной конференции* "Идентификация систем и задачи управления" SICPRO '08. Москва, 26-30 января 2009 г. Институт проблем управления им. В.А. Трапезникова РАН. М: Институт проблем управления им. В.А. Трапезникова РАН, 2009. С. 501-529.

[3] Dorigo M., Maniezzo V., Colorni A.: Ant System: Optimization by Colony of Cooperating Agents // *IEEE Transaction Systems, Man and Cybernetics*. Part B. 1996. V. 26. P. 29-41.

[4] Ходашинский И.А., Дудин П.А.: Параметрическая идентификация нечетких моделей на основе гибридного алгоритма муравьиной колонии // *Автометрия*. 2008. Том 44, № 5. С. 24-35.
Khodashinsky I. A., Dudin P. A.: Parametric Fuzzy Model Identification Based on a Hybrid Ant Colony Algorithm // *Optoelectronics, Instrumentation and Data Processing*, 2008, Vol. 44, No. 5, pp. 402–411.

[5] Socha K., Dorigo M.: Ant Colony Optimization for Continuous Domains / *Technical Report* TR/IRIDIA/2005-037, Universite Libre de Bruxelles. Bruxelles, 2005. 34 p.

[6] Kong M., Tian P.: Application of ACO in Continuous Domain / L. Jiao et al. (Eds.): ICNC 2006, *LNCS* 4222, Part II. Berlin, Springer-Verlag, 2006. P. 126-135.

[7] Kennedy J., Ebenhart R.: Particle Swarm Optimization / *Proceedings* of the 1995 IEEE International Conference on Neural Networks. Perth: IEEE Service Center, 1995. P. 1942-1948.