

Оптимизация нечетких систем типа Такаги-Сугено алгоритмом водного цикла.

Д. А. Чакелев

Научный руководитель И. А. Ходашинский, профессор каф. КИБЭВС

Томский государственный университет систем управления и радиоэлектроники

Кафедра комплексной информационной безопасности электронно-вычислительных систем

Проект ГПО КИБЭВС-1211 – Нечеткие системы

Введение

Нечеткие системы давно и успешно применяются в таких проблемных областях, как классификация, аппроксимация, интеллектуальный анализ данных, распознавание образов, прогнозирование и управление, в задачах принятия решений. Они встроены в огромное количество промышленных изделий, начиная с роботов и систем управления электропоездами, и заканчивая такими потребительскими товарами, как фото и видеокамеры, кондиционеры, стиральные машины и др. Достоинствами нечетких систем является невысокая стоимость разработки, гибкость, интуитивно понятная логика функционирования.

Целью работы является реализация метаэвристического алгоритма водного цикла (Water cycle algorithm) [1] и сравнение результатов с результатами других алгоритмов [2] на наборах данных KEEL (<http://keel.es/>)

Постановка задачи

В данной работе рассматривается нечеткий аппроксиматор типа Такаги-Сугено.

Построение нечеткой системы осуществляется на основе таблицы наблюдений:

$$T = \{t_1, t_2, \dots, t_M\},$$

где $t_k = (\mathbf{x}_k, y_k)$ — строка таблицы наблюдений, $\mathbf{x}_k = (x_1, x_2, \dots, x_n)$ — вектор значений входных переменных, y_k — выходное значение, M — число наблюдений, n — число входных переменных.

На основе таблицы наблюдений на области определения каждой входной переменной формируются нечеткие термы.

Исходя из разбиения переменных на нечеткие термы формируется база правил нечеткой системы. Нечеткое правило в нечеткой системе типа Такаги-Сугено имеет вид:

$$\text{IF } x_1 = A_{1j} \text{ AND } x_2 = A_{2j} \text{ AND } \dots \text{ AND } x_n = A_{nj} \text{ THEN } y = d_{0j} + d_{1j} \cdot x_1 + \dots + d_{mj} \cdot x_m,$$

где $d_{0j} + d_{1j} \cdot x_1 + \dots + d_{mj} \cdot x_m$ — линейная функция, которой оценивается выход y , $j = \overline{1 \dots m}$, m — количество правил, A_{ij} — терм j -го правила i -ой переменной.

На основе сформированной базы правил нечеткая система осуществляет отображение $f: \mathfrak{R}^n \rightarrow \mathfrak{R}$, определяемое формулой:

$$f(\mathbf{x}; \boldsymbol{\theta}; \mathbf{D}) = \frac{\sum_{i=1}^R \prod_{j=1}^n \mu_{A_{ji}}(x_j) (d_{0i} + d_{1i} \cdot x_1 + \dots + d_{mi} \cdot x_m)}{\sum_{i=1}^R \prod_{j=1}^n \mu_{A_{ji}}(x_j)},$$

где $f(\mathbf{x}; \boldsymbol{\theta}; \mathbf{D}) \in \mathfrak{R}$, $\mathbf{x}$ — входной вектор, n — количество входных переменных, $\mu_{A_{ji}}$ — ФП j -ой входной переменной, $\boldsymbol{\theta}$ — вектор параметров ФП, $\mathbf{D}$ — вектор параметров линейных функций консеквентов правил $\mathbf{D} = [d_{01}, \dots, d_{0R}, d_{11}, \dots, d_{1R}, \dots, d_{n1}, \dots, d_{nR}]^T$.

Нечеткий аппроксиматор может быть представлен как:

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

где $\mathbf{x}$ — входной вектор $\boldsymbol{\theta}$ — вектор параметров антецедентов, N — число параметров, описывающих одну функцию принадлежности, y — скалярный выход системы. В качестве

численного критерия адекватности модели используется среднеквадратическая функция ошибки, вычисляющаяся по формуле:

$$MSE(\theta; \mathbf{D}) = \frac{1}{2n} \cdot \sum_{i=1}^n (y_i - f(\mathbf{x}_i; \theta; \mathbf{D}))^2.$$

Описание алгоритма

Алгоритм оптимизации основан на идее движения капель воды от самой верхней точки, к самой нижней. Все капли разделены на три группы:

- поток;
- река;
- море.

Потоки текут к ближайшим рекам, реки текут к морю, море является лучшим решением.

Принадлежность капель к группам определяется их стоимостью. Считается, что чем меньше стоимость, тем ниже положение капли. В качестве стоимости используется значение $MSE(\theta; \mathbf{D})$.

Для моря и рек количество текущих к ним потоков определяется по формуле:

$$NS_n = \text{round} \left\{ \frac{C_n}{\sum_{i=1}^{N_{sr}} C_i} * N_{Raindrops} \right\}, n = 1, 2, 3, \dots, N_{sr}$$

где N_{sr} - количество рек плюс одно море, C_n - текущая стоимость реки или моря, $\sum_{i=1}^{N_{sr}} C_i$ - суммарная стоимость рек и моря, $N_{Raindrops}$ - количество потоков.

Во время движения на каждой итерации потоки двигаются к рекам по формуле:

$$\theta_{Stream}^{i+1} = \theta_{Stream}^i + rand * k * (\theta_{River}^i - \theta_{Stream}^i)$$

где k принимает значения от 1 до 2, $rand$ - случайное число между 0 и 1, θ_{Stream}^i - текущее положение потока, θ_{Stream}^{i+1} - новое положение потока, θ_{River}^i - положение реки для данного потока.

А реки двигаются к морю по формуле:

$$\theta_{River}^{i+1} = \theta_{River}^i + rand * k * (\theta_{Sea}^i - \theta_{River}^i)$$

где k принимает значения от 1 до 2, $rand$ - случайное число между 0 и 1, θ_{River}^i - текущее положение реки, θ_{River}^{i+1} - новое положение реки, θ_{Sea}^i - положение моря.

После каждой итерации если стоимость потока становится меньше чем стоимость соответствующей реки, данный поток становится рекой, а река потоком. Так же если стоимость реки становится меньше чем стоимость моря, данная река становится морем, а море рекой.

Для замедления сходимости алгоритма используется испарение, если выполняется условие:

$$|\theta_{Sea} - \theta_{River}^i| < d_{max}, \quad i = 1, 2, 3, \dots, N_{sr} - 1$$

где d_{max} - максимальное расстояние приближения рек к морю.

То река перемещается по формуле (1) или (2):

$$\theta_{River}^{new} = \mathbf{LB} + rand * (\mathbf{UB} - \mathbf{LB}) \quad (1)$$

где $\mathbf{LB}$ и $\mathbf{UB}$ - нижняя и верхняя граница, $rand$ - случайное число от 0 до 1, θ_{River}^{new} - новое положение реки.

$$\theta_{River}^{new} = \theta_{Sea} + \sqrt{\mu} * randn(1, N_{var}) \quad (2)$$

где $randn(1, N_{var})$ - нормальное распределение со сдвигом на 1 и разбросом равным количеству параметров системы, μ - значение примерно равное 0.1, θ_{Sea} - текущее положение моря, θ_{River}^{new} - новое положение реки.

После того как прошла процедура испарения значение d_{max} пересчитывается по

формуле:

$$d_{max}^{i+1} = d_{max}^i - \frac{d_{max}^i}{I}$$

где I – максимальное число итераций алгоритма.

Алгоритм продолжается до тех пор, пока не будет достигнуто необходимое количество итераций или выполнено другое условие выхода.

Возникшие при реализации проблемы и их решения

В ходе программной реализации основной проблемой было использование в расчетах формулы (2), т.к. иногда положение рек выходило за пределы исследуемого пространства. Также испарение не давало прироста в точности системы и только замедляло работу алгоритма.

По этим причинам в качестве основной формулы для испарения была выбрана (1) и в программной реализации появилась возможность отключать испарение при необходимости.

Полученные результаты

В данный момент алгоритм программно реализован для оптимизации нечетких аппроксиматоров типа сингтон и Такаги-Сугено. На данный момент проведено тестирование на следующих наборах данных plastic, ele-2, delta_ail, friedman, dee и проведено сравнение с результатами полученными другими алгоритмами.

Таблица 1 – Сравнение с алгоритмами идентификации

Data	ANFIS-SUB		TSK-IRL		Linear-LMS		LEL-TSK		METSK-HD ^e		WCA	
	<i>trn</i>	<i>Tst</i>	<i>trn</i>	<i>tst</i>	<i>trn</i>	<i>Tst</i>	<i>Trn</i>	<i>tst</i>	<i>trn</i>	<i>tst</i>	<i>trn</i>	<i>tst</i>
pla	1,011	1,504	1,090	1,146	1,166	1,172	1,032	1,188	1,057	1,136	1,062	1,141
ele-2	8208	8525	17024	19786	13361	13541	2928	3752	2270	3192	6643	5922
delail	0,973	1,484	1,321	1,419	1,478	1,478	1,193	1,760	1,190	1,402	1,716	1,717
fried	0,085	3,158	0,433	1,419	3,612	3,653	0,322	1,070	1,075	1,888	0,43	0,822
dee	3087	2083	0,545	882,016	0,081	0,085	0,662	0,682	0,030	0,103	0,054	0,083

Заключение

Из таблицы 1 видно, что на данный момент алгоритм уже показывает неплохие результаты, но выборки с большим количеством параметров обрабатываются медленно из-за высокой сложности системы.

Сейчас основной целью является нахождение или создание метода по уменьшению количества правил системы за счет отключение некоторых параметров и подбора термов отдельно для каждого параметра, а также сокращение обучающей выборки для достижения более быстрой скорости обучения системы и удаления из выборки аномальных экземпляров.

Список литературы

1. Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems / H. Eskandar et al. // Computers and Structures. – 2012. – V. 110–111. – P. 151–166.
2. METSK-HDe: A multiobjective evolutionary algorithm to learn accurate TSK-fuzzy systems in high-dimensional and large-scale regression problems / M.J. Gacto et al. // Information Sciences. – 2014. – V. 276. – P. 63–79.