ОПТИМИЗАЦИЯ ПАРАМЕТРОВ НЕЧЕТКИХ СИСТЕМ НА ОСНОВЕ ГРАВИТАЦИОННОГО АЛГОРИТМА

А.В. Цой

Научный руководитель: профессор кафедры КИБЭВС И.А. Ходашинский

Томский государственный университет систем управления и радиоэлектроники

Кафедра комплексной информационной безопасности электронно-вычислительных систем

Аннотация: в статье представлено описание алгоритма «GSA: A Gravitational Search Algorithm», выбранного для идентификации нечетких систем.

Ключевые слова: *нечеткие системы, оптимизация, эвристические алгоритмы ~~поиска~~, гравитационный алгоритм поиска.*

Введение

Методы аппроксимации используются в компьютерном моделировании и идентификации параметров исследуемых систем в тех случаях, когда описание объекта задано в виде таблицы наблюдений и отсутствует математическая модель данного объекта. Среди большого разнообразия методов и средств решения проблем аппроксимации особое место занимают нечеткие аппроксиматоры, в силу их способности работать в условиях неопределенности данных, а также интерпретировать эти данные. Однако остается открытым вопрос определения параметров нечетких аппроксиматоров по конкретной таблице наблюдений.

Цель работы состоит в описании гравитационного алгоритма для оптимизации нечеткого аппроксиматора.

Нечеткая система

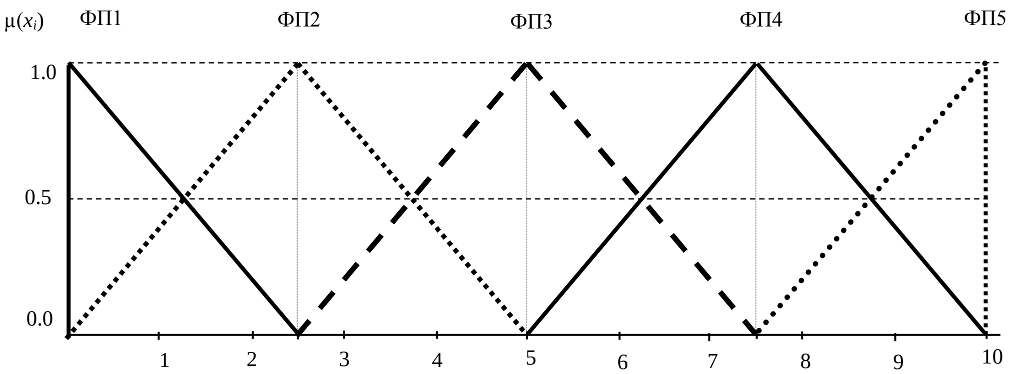
На начальном этапе у нас имеется таблица наблюдений, вида:

$$(x_{1i}, x_{2i} \dots x_{ni}, O(x_{1i}, x_{2i} \dots x_{ni})), \text{ где } i = \overline{1, n},$$

где $x_{1i}, x_{2i} \dots x_{ni}$ – значения входных переменных, $O(x_{1i}, x_{2i} \dots x_{ni})$ – значение выходной переменной. Необходимо для каждой входной переменной задать термы. Имея минимальное и максимальное значения для каждой переменной, нужно разделить каждый интервал на необходимое число термов так, чтобы область определения входных данных была покрыта равномерно.

В работе использованы треугольные функции принадлежности для определения термов, т.к. они задаются всего 3 переменными (a_i, b_i, c_i), требуют меньшего количества машинных вычислений при определении степени принадлежности, что сокращает общее время оптимизации [1]. Формула треугольной функции принадлежности приведена ниже, вид распределения представлен на рисунке.

$$\mu(x) = \begin{cases} \frac{x-a}{b-a}, & a \le x \le b \\ \frac{x-c}{b-c}, & b \le x \le c \\ 0, & \text{в других случаях} \end{cases}$$,где $\mu(x)$ – функция принадлежности;



После создания термов, необходимо задать базу правил. В системе типа *«Singleton»* i-ое правило имеет вид:

$$\text{ЕСЛИ } (x_1 = A_{1i} \text{ и } x_2 = A_{2i} \text{ и} \ldots\ x_n = A_{ni}) \text{ ТО } y = O_i,$$

где A_{ij} – терм, O_i – значение консеквента -го правила [2].

Консеквенты O_i инициализируются по ближайшему значению из таблицы наблюдений (ТН). Алгоритм следующий:

Шаг 1. Взять по одной вершине терма каждой переменной – таким образом, образуется точка $\tilde{x} = \{b^1_{j_{1l}}, b^2_{j_{2l}}, \ldots, b^n_{j_{nl}}\}$, где j_{il}- номер функции принадлежности (ФП) -ой переменной в l-ой строке базы правил, $l = \overline{1..L}$, где L – количество правил в базе.

Шаг 2. Для точки $\tilde{x}$ находится ближайшее значение из ТН по формуле

$$min_k[\sum_{i=1}^{m}\left(\frac{b_i^i - x_{ij}}{x_{\max i} - x_{\min i}}\right)^2]$$

В качестве консеквента O_l берется значение из ТН в ближайшей точке $O(x_{1i}, x_{2i} \ldots x_{ni})$

Повторяя шаги 1 и 2 L раз, будет сформирована база правил системы [3].

Нечеткая система типа «*Singleton*» осуществляет отображение $f: R^n \to R$:

$$F(\mathbf{x}) = \frac{\sum_{i=1}^{L} \mu_{A_{1i}}(x_1) \cdot \mu_{A_{2i}}(x_2) \cdot \ldots \cdot \mu_{A_{ni}}(x_n) \cdot O_i}{\sum_{i=1}^{L} \mu_{A_{1i}}(x_1) \cdot \mu_{A_{2i}}(x_2) \cdot \ldots \cdot \mu_{A_{ni}}(x_n)},$$

где L – число правил в базе.

Таким образом, ошибка вычисляется по следующей формуле:

$$MSE = \frac{1}{R}\sqrt{\sum_i^R (O(\mathbf{x_i}) - F(\mathbf{x_i}))^2},$$

где R – число наблюдений [3].

Результатом выполнения вышеуказанных действий является вектор $\mathbf{X_i}$, который можно представить в виде:

MSE	a_1^1	b_1^1	c_1^1	a_2^1	b_2^1	c_2^1	…	a_n^j	b_n^j	c_n^j	O_1	O_2	…	O_L

Изменяя параметры и повторяя действия, будет получено несколько векторов – популяция. У каждого вектора – своя ошибка MSE. Суть оптимизации нечеткой системы сводится к минимизации ошибки, путем подбора соответствующих антецедентов и консеквентов. В решении задач оптимизации в высокомерном пространстве поиска алгоритмы классической оптимизации не обеспечивают приемлемое решение, потому что пространство поиска с размером проблемы возрастает экспоненциально, поэтому решение этой проблемы при использовании явных методов (например, перебор), не является приемлемым.

В течение последних десятилетий наблюдается растущий интерес к алгоритмам, основанных на природных явлениях. Многими исследователями было показано, что эти алгоритмы хорошо подходят для решения сложных вычислительных задач, в частности таких, как оптимизация целевых функций. В данной работе будет рассмотрен гравитационный алгоритм поиска [4].

Об алгоритме

Гравитационный алгоритм является эвристическим, т.е. это алгоритм, правильность которого для всех возможных случаев не доказана, однако он предлагает достаточно хорошие решения в большинстве случаев. Поиск ведется в параллельно в нескольких точках. В каждой точке производится ряд определенных операций и обмен информацией с другими. Эти операции просты, однако их общий эффект производит неожиданный результат.

В гравитационном алгоритме агенты представляют собой набор масс, которые взаимодействуют друг с другом на основе ньютоновских законов движения и гравитации: «Каждая частица во вселенной притягивает другую частиц с силой, прямо пропорциональной произведению их масс и обратно пропорциональной квадрату расстояния между ними» [4].

В предложенном алгоритме агентами являются объекты, чья производительность измеряется их массой. Все эти объекты притягиваются друг к другу с силой тяжести, эта сила влияет на глобальное движение всех объектов к объектам с более тяжелой массой. Следовательно, массы соединяются, используя направленную форму общения, через гравитационные силы. Более тяжелые массы, соответствующие хорошим решениям, двигаются медленнее легких масс, это гарантирует использование шага алгоритма.

Каждая масса (вектор) имеет 4 характеристики: положение, инерционная масса, активная гравитационная масса, пассивная гравитационная масса. Положение массы соответствует решению проблемы, а ее гравитационная и инерционная массы определяются функцией пригодности (ошибкой MSE). Другими словами, каждая масса представляет решение, алгоритм управляется правильной регулировкой инерционных и гравитационных масс. По истечению времени массы будут притягиваться к более тяжелым массам. Эти массы предоставят оптимальное решение.

Алгоритм может рассматриваться как отдельная система масс. Это как искусственный мир масс, повинующийся законам гравитации и движения Ньютона. Точнее массы соблюдают законы.

Закон гравитации (всемирного тяготения) каждая частица притягивает другую частицу и сила тяжести между двумя частицами прямо пропорциональна произведению их масс и обратно пропорциональна расстоянию между ними.

Закон движения: текущая скорость массы эквивалентна сумме доли ее предыдущей скорости и изменений в скорости. Изменение скорости или ускорение любой массы равно силе разделенной массы инерции [4].

Информация, необходимая для понимания алгоритма

$$G(t) = G(t_0) \times (\frac{t_0}{t})^{\beta},$$

где $\beta < 1$; $G(t)$ – значение гравитационной постоянной в момент времени t, $G(t_0)$ – значение в начальный момент времени,.

Активная гравитационная масса M_a показывает, какое гравитационное поле создает само тело. Гравитационное поле объекта с небольшой активной гравитационной массой слабее, чем объект с большей активной гравитационной массой.

Пассивная гравитационная масса M_p показывает, с какой силой тело взаимодействует с внешними гравитационными полями. В одном и том же гравитационном поле объект с меньшей пассивной гравитационной массой испытывает меньшую силу, чем объект с большей пассивной гравитационной массой.

Инертная масса M_i – это мера объекта сопротивляться изменению его состояния движения, когда будет применена сила. Объект с большой инерционной массой, изменяет состояние своего движения более медленно, в то время как объект с небольшой инерционной массой меняет быстрее.

$$F_{ij} = G\,\frac{M_{aj} \times M_{pi}}{R^2},$$

где F_{ij} – величина гравитационной силы, которая действует на массу i по массе j пропорционально произведению активной гравитационной массы j и пассивной гравитационной массы i и обратно пропорционально квадрату расстояния между ними.

$$a_i = \frac{F_{ij}}{M_{ii}},$$

где a_i – второй закон Ньютона [4].

Гравитационный алгоритм оптимизации для нечеткой системы

Шаг 1. Инициализация. Задается максимальное количество итераций T, максимальное количество векторов S. Задается β – коэффициент, уменьшающий G, что позволяет контролировать точность поиска. Задается популяция из n векторов:

$$\mathbf{X_i} = \left(x_i^1 \dots x_i^d \dots x_i^n\right)$$

где x_i^d – позиция вектора в пространстве размерности d, $i = \overline{1,n}$.

Шаг 2. Рассчитывается ошибка, т.е. $MSE(\mathbf{X_i}, t)$, где $i = \overline{1,n}$.

Шаг 3. Находится G. Находится лучшее и худшее решение, $best(t)$ и $worst(t)$, соответственно.

Для задачи минимизации:

$$worst(t) = \min\ MSE_i(t),$$

$$best(t) = \max\ MSE_i(t),$$

где $i = \overline{1,n}$.

Шаг 4. Рассчитывается масса M, при помощи $m_i(t)$, и ускорение a.

$$m_i(t) = \frac{MSE_i(\mathbf{X_i}, t) - worst(t)}{best(t) - worst(t)},$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{T} m_j(t)}.$$

Для простоты реализации предполагаем, что $M_{ai} = M_{pi} = M_{ii} = M_{i,}$ где $i = \overline{1,n}$.

$$a_i^d(t) = \frac{\sum_{j=1, j\neq i}^{M} rand_j F_{ij}^d(t)}{M_{ii}(t)},$$

где $rand_j$– случайное число в интервале [0,1].

Шаг 5. Рассчитывается скорость и местоположение агентов.

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t),$$

где $v_i^d(t+1)$ – скорость агента следующий момент времени, рассматривается как доля его текущей скорости, добавленная к его ускорению.

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1)$$

где $x_i^d(t+1)$ – положение агента в следующий момент времени.

Шаг 6. Все вектора упорядочиваются по убыванию $best(t)$ и M_i. Если количество векторов превышает S, популяция уменьшается до S. Если текущая итерация меньше T, то переходим к шагу 2 [4].

Список литературы

1. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И.Д. Рудинского. – М.:Горячая линия – Телеком, 2006. – 452 с.: ил.
2. Ходашинский И.А. Идентификация нечетких систем: методы и алгоритмы // Проблемы управления. 2009. № 4. С. 15–23.
3. Ходашинский И.А., Дудин П.А. Идентификация нечетких систем на основе непрерывного алгоритма муравьиной колонии // Автометрия. – 2012. – Т. 48, № 1. – С. 45–71.
4. Esmat Rashedi GSA: A Gravitational Search Algorithm // Department of Electrical Engineering, Shahid Bahonar University of Kerman. – 2009. – 18 с.