Алгоритм «мозговой штурм» для настройки нечеткого аппроксиматора

Ю.А. Сорокина

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

г. Томск, ТУСУР, 2016

Проект ГПО КИБЭВС-1211 – Нечёткие системы

Введение

За последние несколько десятилетий алгоритмы роевого интеллекта, вдохновленные принципами природных явлений, привлекают все больше внимания.

В основе алгоритмов роевого интеллекта лежит коллективное поведение индивидов (частиц), используемое для эффективной оптимизации задачи. Все частицы постепенно двигаются в сторону лучшего решения в области решений. Эти частицы олицетворяют простые объекты, такие как птицы, муравьи, бактерии и др. Люди – «социальные животные», самые разумные существа на планете. Следовательно, алгоритм оптимизации, вдохновленный человеческим процессом поиска наилучшего решения – в данной работе мозговым штурмом (т.е. генерации идей для решения поставленной задачи), – должен давать хорошие результаты при грамотной реализации.

Целью данной работы является исследование алгоритма «мозговой штурм» с использованием оператора дифференциальной эволюции, а также применение алгоритма для идентификации параметров нечетких аппроксиматоров.

Постановка задачи

Нечеткий аппроксиматор

Нечеткий аппроксиматор задается правилами следующего вида:

$$\text{ЕСЛИ } s_1 = A_{i1}\ AND\ s_2 = A_{i2}\ AND\ \dots\ AND\ s_n = A_{in}\ \text{ТО}\ y = r_{i0} + r_{i1}s_1 + \dots + r_{in}s_n$$

где A_{ji} – лингвистический терм, которым оценивается входная переменная s_j; выход y задается линейной функцией от входных переменных.

Выход нечеткого аппроксиматора определяет отображение:

$$f(\mathbf{s};\mathbf{X};\mathbf{r}) = \frac{\sum_{i=1}^{R}\prod_{j=1}^{n}\mu_{ij}(s_j) * (r_{i0} + r_{i1}s_1 + \dots + r_{in}s_n)}{\sum_{i=1}^{R}\prod_{j=1}^{n}\mu_{ij}(s_j)}$$

где $\mathbf{s}$ – входной вектор; R – число правил; n – количество входных переменных; μ_{ij} – функция принадлежности j-й входной переменной i-го нечеткого правила, характеризующая нечеткий терм A_{ij}; $\mathbf{X} = ||X_1, \dots, X_N||$ – вектор параметров нечеткого аппроксиматора; $\mathbf{r} = ||r_1, \dots, r_N||$ – вектор параметров консеквентов.

Пусть имеется таблица наблюдений $\{(\mathbf{s}_p\ ;\ t_p), p = \overline{1,m}\}$, тогда критерий качества аппроксимации может быть выражен среднеквадратической функцией ошибки, вычисленной по формуле:

$$MSE(\mathbf{s}_p, \mathbf{X}, \mathbf{r}) = \frac{\sum_{p=1}^{m}(t_p - f(\mathbf{s}_p, \mathbf{X}, \mathbf{r}))^2}{m} \quad (1)$$

Для оптимизации параметров $\mathbf{X}$ предлагается применить алгоритм «мозговой штурм».

Вход: N – число идей; m – число кластеров; максимальное число итераций; вероятности выбора $p\_one, p\_one\_center, p\_two\_center$.

Выход: $\mathbf{X}^*$– вектор оптимальных параметров нечеткой системы для аппроксиматора минимизирующий среднеквадратическую ошибку (1).

Описание алгоритма

Алгоритм оптимизации «мозговой штурм» основан на коллективном решении поставленной задачи.

Рассмотрим алгоритм, имитирующий поиск лучшего решения, используя «мозговой штурм» для идентификации параметров нечеткой системы.

На первом шаге производим инициализацию идей N и оценку их фитнес-функции. Самую лучшую идею в популяции обозначим *GlobalIdea*. Также задаем количество кластеров m, в которые в последствии объединим идеи ($m<N$).

На втором шаге N идей объединяются в m кластеров по предпочитаемому методу кластеризации. Авторы алгоритма предлагают метод К-средних, возможно использование горной кластеризации.

На третьем шаге лучшие идеи в каждом кластере объявляются центрами кластера x_{center}.

Основу алгоритма также составляет произвольный выбор кластера/кластеров. Соответственно на четвертом шаге выбирается кластер: $rand < p\_one$ или два кластера: $rand \geq p\_one$.

На пятом шаге выполняется оригинальный оператор алгоритма: $rand \geq p\_one\ center$ или оператор дифференциальной эволюции: $rand < p\_one\ center$.

1. Оригинальный оператор алгоритма «мозговой штурм»

Выбираем идею и изменяем ее на шаг $\xi * normrnd(0,1)$:

$$\mathbf{X}_{select} = \begin{cases} \mathbf{X}_i + \xi * normrnd(0,1) - \text{для одного кластера;} \\ rand * \mathbf{X}_{i1} + (1 - rand)\mathbf{X}_{i2} + \xi * normrnd(0,1) - \text{для двух.} \end{cases}$$

где $\mathbf{X}_i$ – выбранная произвольная идея в кластере;

$\mathbf{X}_{i1}$, $\mathbf{X}_{i2}$ – выбранные произвольные идеи в первом кластере и втором соответственно;

$\xi = rand * logsig\left(\frac{0.5 * max\_iteration - current\_iteration}{p}\right)$;

$max\_iteration$ – заданное число итераций;

$current\_iteration$ – номер итерации в момент выполнения;

$logsig(x) = \frac{1}{1+e^{-x}}$ – сигмоида*;*

$p -$ изгиб сигмоиды;

$normrnd(0,1)$ – нормально распределенная величина.

2. Оператор дифференциальной эволюции

Операция мутации (увеличивает зону поиска)*:*

Для каждого целевого вектора $\mathbf{X_i^k}$ генерируется вектор мутации:

$$\mathbf{V_{ij}^k} = \begin{cases} X_{center}^k + F * \left(X_{r1}^k - X_{r2}^k\right) \\ GlobalIdea + F * \left(X_{r1}^k - X_{r2}^k\right) \end{cases}$$

где $F \in [0,2]; i = 1..N; j = 1..D; D$ – число параметров; k – поколение.

Рекомбинация (имитирует естественный отбор):

$$\mathbf{U_{ij}^k} = \begin{cases} \mathbf{V_{ij}^k}, \text{ если } rand \leq CR \\ \mathbf{X_{ij}^k} \end{cases}$$

где $CR \in [0,1]$

Отбор (использует успешные идеи из предыдущих итераций):

$$\mathbf{X}_{select} = \begin{cases} \mathbf{U_i^k}, \text{ если } f(\mathbf{U_i^k}) < f(\mathbf{X_i^k}) \\ \mathbf{X_i^k} \end{cases}$$

На шестом шаге происходит обновление идеи с помощью «нового шага»:

$$\mathbf{X}_{new} = \mathbf{X}_{select} + \xi * normrnd(0,1)$$

$$\xi = rand * exp\left(1 - \frac{max\_iteration}{max\_iteration - current\_iterration + 1}\right),$$

Принятие новой идеи, если она лучше существующей:

$$\mathbf{X}_i = \begin{cases} \mathbf{X}_{new}, \text{ если } f(\mathbf{X}_{new}) < f(\mathbf{X}_i) \\ \mathbf{X}_i, \text{иначе} \end{cases}$$

Заключение

Алгоритм мозгового штурма – популяционный организм, вдохновленный групповой мозговой деятельностью людей.

Так же, как и другие метаэвристические алгоритмы, данный алгоритм относится к числу эволюционных и способен решать ряд оптимизационных проблем, в числе которых нелинейность, не дифференцируемость, высокая размерность пространства поиска с большой скоростью схождения.

Список литературы

1. An Improved Brain Storm Optimization with Differential Evolution Strategy for Applications of ANNs // Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2015, Article ID 923698, 18 pages;

2. Brain Storm Optimization Algorithm // Yuhui Shi, Xi'an Jiaotong-Liverpool University, Suzhou, China 215123, P.2 – 5;

3. Сарин К.С. Гибридные алгоритмы анализа данных на основе компактных и точных нечетких систем типа Такаги-Сугено // ТУСУР. – 2016. – С. 30-31.