Алгоритм креативного обучения для настройки нечеткого классификатора

П.А. Ковчунов

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

г. Томск, ТУСУР

Проект ГПО КИБЭВС-1211 – Нечёткие системы

Введение

Огромное количество задач, встречающихся в различных отраслях точных наук, связаны с таким вычислительным аппаратом, как непрерывная глобальная оптимизация. Эти задачи отличает крайняя вычислительная сложность при попытке решить их аналитическим методом. [1] Начиная с 80-х годов XX века разрабатываются стохастические алгоритмы оптимизации, предназначенные для решения именно таких задач. [2] В этой статье рассмотрен метаэвристический алгоритм, вдохновленный механизмом человеческого мышления.

Цель работы

Целью данной работы является настройка параметров нечеткого классификатора. Основная идея нечеткого классификатора состоит в описании предполагаемого кластера нечетким прототипом, размерность которого определена размерностью пространства исследуемых данных. Таким образом, i-й кластер определяется нечетким правилом следующего вида:

R_{ij}: ЕСЛИ $x_1 = A_{1i}$ И $x_2 = A_{2i}$ И $x_3 = A_{3i}$ И … И $x_n = A_{ni}$ ТО $class = c_j$,

где $\mathbf{x} = (x_1, x_2, x_3, \dots, x_n)$ – вектор признаков классифицируемого объекта;

A_{ki} – нечеткий терм, характеризующий k-ый признак в i-том правиле ($i \epsilon [1, R]$), R – число правил;

c_j – идентификатор j-того уровня, $j \in [1, m]$.

Задача традиционной классификации может быть описана функцией

$f: \Re^n \to \{0,1\}^m$,

где $f(\mathbf{x}) = (c_1, c_2, \dots, c_m)$ причем $c_i = 1$, а $c_j = 0$ ($j \in [1, m]$, $i \neq j$), когда объект, заданный вектором **x**, принадлежит к классу c_i.

Нечеткая классификация описывается функцией

$f: \Re^n \to [0,1]^m$,

которая относит классифицируемый объект к каждому классу с определённой степенью принадлежности, вычисленной следующим образом:

$\beta_j(\mathbf{x}) = \sum_{R_{ij}} \prod_{k=1}^{n} A_{ki}(x_k), j = \overline{1, m}$.

Выходом классификатора является класс, определяемый следующим образом:

$class = c_{j*}, j* = \arg \max\limits_{1<j<m} \beta_j$.

Нечеткий классификатор может быть представлен как функция

$$c = f(\mathbf{x}; \boldsymbol{\omega})$$

где $\boldsymbol{\omega}$ – вектор, описывающий базу правил.

Пусть дано множество обучающих данных (таблица наблюдений) $\{(\mathbf{x}_p : c_p), p = \overline{1, z}\}$, определим следующую функцию

$$delta(p, \boldsymbol{\omega}) = \begin{cases} 1, \text{если } c_p = f(\mathbf{x}_p, \boldsymbol{\omega}) \\ 0, \text{иначе} \end{cases}, p = \overline{1, z},$$

тогда численный критерий качества классификации выражается следующим образом:

$$E(\boldsymbol{\omega}) = \frac{\sum_{p=1}^{z} delta(p, \boldsymbol{\omega})}{z}$$

На основе таблицы наблюдений формируются нечеткие термы на области определения каждой входной переменной. В данной работе термы представляются в виде треугольников и каждая переменная описывается тремя термами – A_{p1}, A_{p2}, A_{p3}.

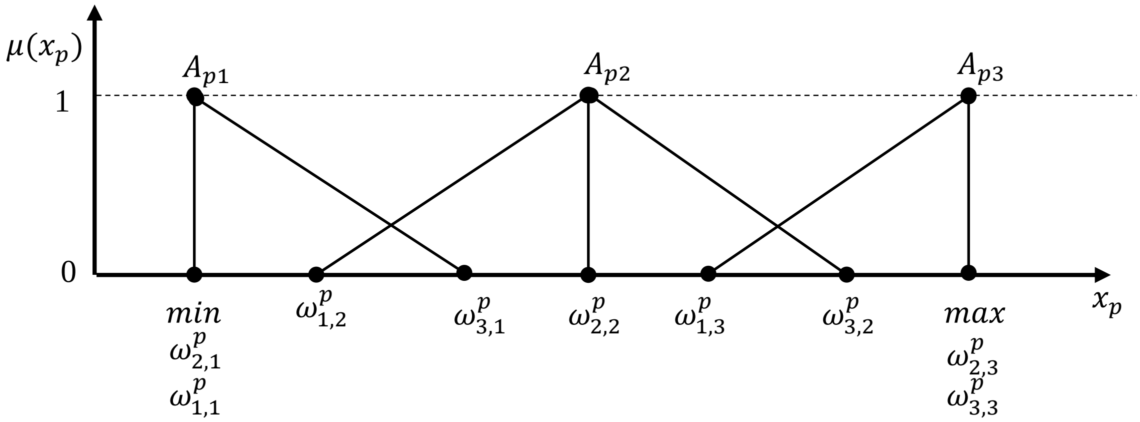

Рисунок 1 – Нечёткий терм для k-го признака

Проблема идентификации сводится к проблеме поиска максимума заданной функции в многомерном пространстве, координаты которого соответствуют параметрам нечеткого классификатора.

Базу правил нечеткого классификатора можно представить в виде вектора **ω**, обозначенном в таблице 1, который и будет обрабатываться нашим алгоритмом. В где $\omega_{j,i}^p$ – i-й параметр j-го терма p-й признака

Таблица 1 – Вектор, описывающий базу правил

E(**ω**)	$\omega_{1,1}^1$	$\omega_{2,1}^1$	$\omega_{3,1}^1$	$\omega_{1,2}^1$	$\omega_{2,2}^1$	$\omega_{3,2}^1$	…	$\omega_{i,j}^p$	…	$\omega_{1,j}^k$	$\omega_{2,j}^k$	$\omega_{2,j}^k$

Описание алгоритма

Алгоритм креативного обучения – это метаэвристический алгоритм поиска. В данной работе будут описаны основные составляющие алгоритма, представление решения, процесс дивергентного, конвергентного и креативного мышления.

Данный алгоритм предложили Xiang Feng, Ru Zou и Huiqun Yu [3] в 2014 г. Алгоритм моделирует мышление человека в процессе решения им прикладной задачи. Предполагается, что мы найдем такую идею-вектор, фитнес-функция от которой будет всегда меньше либо равна фитнес-функции от любого другого вектора. (рассматриваем задачу глобальной минимизации).

Рассмотрим алгоритм подробнее:

Шаг 1. Для каждой идеи генерируется вектор $\omega_i = (\omega_{i1}, \omega_{i2}, \ldots, \omega_{in})$.

Чтобы начать процесс дивергентного мышления, случайным образом генерируется вектор

$$\Delta\boldsymbol{\omega}_i = (\Delta\boldsymbol{\omega}_{i1}, \Delta\boldsymbol{\omega}_{i2}, \ldots, \Delta\boldsymbol{\omega}_{in})$$

$$\omega'_i = \omega_i + \Delta\boldsymbol{\omega}_i,$$

где $\Delta\boldsymbol{\omega}_{ij}$ – величина, распределенная по нормальному закону, с математическим ожиданием, равным нулю и дисперсией δ^2, которая называется новизной идеи и выбирается в зависимости от исследуемой области.

Шаг 2. Сравниваем $E(\omega)$ и $E(\boldsymbol{\omega}')$

Если $E(\boldsymbol{\omega}')$ меньше, то δ^2=$\delta^2 * Sfactor$, здесь $Sfactor \in (0..1)$ – заранее заданный коэффициент масштабирования, предназначенный для того, чтобы уменьшить дисперсию и искать оптимум в более тесной окружности вокруг точки.

Иначе δ^2=$\delta^2 / Sfactor$, таким образом переходы становятся дальше.

Шаг 3. Вычисляем для каждой пары векторов из популяции:

$$O(\overline{\boldsymbol{\omega}}_i) = \begin{cases} \left|\left|\overline{\boldsymbol{\omega}}_i, \overline{\boldsymbol{\omega}}_t\right|\right|, & \text{если } E(\overline{\boldsymbol{\omega}}_i) < E(\overline{\boldsymbol{\omega}}_t) \\ 0, & \text{иначе} \end{cases}$$

$$i = 1, 2, \ldots, DNum$$

$$\overline{\boldsymbol{\omega}}_{t+1} = \begin{cases} \overline{\boldsymbol{\omega}}_k & \text{если } O(\overline{\boldsymbol{\omega}}_k) > 0 \\ \overline{\boldsymbol{\omega}}_t, & \text{иначе} \end{cases}$$

$$k = \arg\max O(\overline{\boldsymbol{\omega}}_i)$$

Шаг 4. Строим элементы матрицы $\boldsymbol{\alpha}$ следующим образом, используя вспомогательную матрицу:

$$M_{i,j} = \begin{cases} 0, & \text{если } i = j \\ \dfrac{E(\overline{\boldsymbol{\omega}}_j) - E(\overline{\boldsymbol{\omega}}_i)}{|E(\overline{\boldsymbol{\omega}}_j)|}, & \text{если } i \neq j \end{cases}$$

$$\alpha_{i,j} = \frac{M_{i,j}}{\sum_{i=1}^{n} |M_{i,j}|}$$

Шаг 5. Вычислим новую идею U_i для каждого вектора на основе матрицы влияния $\boldsymbol{\alpha}$ и двух случайно взятых векторов $\boldsymbol{\omega}_a$ и $\boldsymbol{\omega}_b$:

$$U_i = \begin{cases} \overline{\boldsymbol{\omega}}_a + \alpha_{a,i} * (\overline{\boldsymbol{\omega}}_a - \overline{\boldsymbol{\omega}}_i) + (\alpha_{b,i} - \alpha_{r,i}) * (\overline{\boldsymbol{\omega}}_b - \overline{\boldsymbol{\omega}}_r), & \text{если } \alpha_{a,i} > 0 \\ \overline{\boldsymbol{\omega}}_i + \alpha_{a,i} * (\overline{\boldsymbol{\omega}}_a - \overline{\boldsymbol{\omega}}_i) + (\alpha_{b,i} - \alpha_{r,i}) * (\overline{\boldsymbol{\omega}}_b - \overline{\boldsymbol{\omega}}_r), & \text{иначе} \end{cases}$$

Шаг 6. $\boldsymbol{\omega}_i = \begin{cases} U_i, \text{если } E(U_i) < E(\boldsymbol{\omega}_i) \\ \boldsymbol{\omega}_i \text{ иначе} \end{cases}$

Шаг 7. Отсортировать идеи по возрастанию ошибки и вывести лучшее решение (максимальное значение фитнес-функции $E(\omega)$).

Заключение

Алгоритм креативного обучения – это популяционный алгоритм, берущий свои корни из жизни. Также, как и другие метаэвристические алгоритмы, данный алгоритм относится к числу эволюционных и способен решать ряд оптимизационных проблем, в числе которых нелинейность, недифференцируемость, высокая размерность пространства поиска с большой скоростью схождения. Ещё одно преимущество описываемого алгоритма заключается в том, что данный алгоритм регулируется малым числом параметров, делая его достаточно лёгким для реализации.

Литература

1. Карпенко А. П. Популяционные алгоритмы глобальной поисковой оптимизации. Обзор новых и малоизвестных алгоритмов // «Информационные технологии», — 2012 — №7 — С. 1-3.
2. И.А. Ходашинский, И.В. Горбунов – Построения нечетких классификаторов на основе алгоритма пчелиной колонии // Автометрия. – 2012. – Т. 48, № 1. – С. 45–71.
3. Xiang Feng, Ru Zou, Huiqun Yu. A novel optimization algorithm inspired by the creative thinking process // Springer-Verlag Berlin Heidelberg. – 2014 — С. 1-12.