УДК 004.85

Алгоритм построения деревьев моделей с внутренними регрессионными узлами на основе моделирования поведения колонии пчел при поиске нектара

Мельников Г.А., Гаврилов А.В.

Новосибирский государственный технический университет

Описан новый метод построения деревьев моделей с внутренними регрессионными узлами на основе моделирования поведения колонии пчел при поиске нектара. Результаты экспериментов показывают, что предложенный алгоритм превосходит традиционные алгоритмы по среднеквадратичной адекватности идентификации и позволяет значительно уменьшить сложность получаемых моделям.

Ключевые слова: кусочно-заданная регрессия, деревья регрессии, деревья моделей, пчелиные алгоритмы.

Деревья моделей являются одним из важных классов регрессионных моделей, позволяющим осуществить разделение пространства объясняющих переменных на сегменты с последующим построением для каждого из них собственной модели и представить кусочно-заданную функцию регрессии в интуитивно понятной и наглядной форме. В таком дереве внутренние узлы содержат правила разделения пространства объясняющих переменных; дуги – условия перехода по ним; а листья – локальные регрессионные модели.

Интересным развитием идеи классических деревьев моделей является введение внутренних регрессионных узлов (рис. 1). Такие узлы содержат факторы, одинаково влияющие на целевую переменную для всех нижележащих сегментов, что может положительно сказаться на интерпретируемости деревьев моделей.

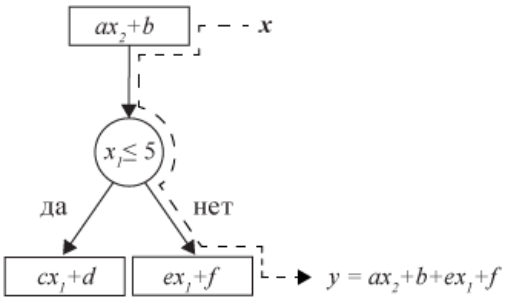

Рисунок 1 – Пример дерева моделей с внутренними регрессионными узлами

Однако строить такие модели значительно сложнее. Хорошей альтернативой жадным алгоритмам здесь может быть группа эвристических методов поиска и оптимизации. Поэтому мы разработали стохастический многоагентный алгоритм построения деревьев моделей с внутренними регрессионными узлами на основе моделирования поведения колонии пчел при поиске нектара.

В предлагаемом алгоритме можно условно разделить агентов на две группы: рабочие пчелы и пчелы наблюдатели. За каждой рабочей пчелой закреплен некоторый источник нектара (решение в пространстве поиска), который она разрабатывает. Пчелы наблюдатели на основе полученной информации о найденных решениях от рабочих пчел выбирают одно из решений и пытаются улучшить его.

Выделим основные шаги предлагаемого алгоритма:

1. Сгенерировать начальное множество решений **S**.
2. Для каждой *i*-ой рабочей пчелы:

2.1. сгенерировать новое решение *Vi* в окрестности решения *Si* и оценить его;

2.2. если новое решение лучше *Si*, то заменить *Si* на *Vi*.

3. Для каждой *i*-ой пчелы наблюдателя:

3.1. Вероятностно выбрать решение *Sj* прямо пропорционально качеству решений;

3.2. сгенерировать новое решение *Vi* в окрестности решения *Sj* и оценить его;

3.3. если новое решение лучше *Sj*, то заменить *Sj* на *Vi*.

4. Определить «исчерпанные» (не удалось улучшить *limit* раз) источники пищи и заменить их на случайные решения.

5. Повторять шаги 2 – 4 до тех пор, пока не будут достигнуты условия останова алгоритма.

Все операции непосредственно осуществляются над деревьями моделей, изначально сгенерированными случайным образом. Для оценивания решений был использован расширенный байесовский информационный критерий [1]:

$$EBIC = n \cdot \ln(SSE / n) + J \cdot (\ln(n) + 2\ln(p)), \quad (1)$$

где *SSE* – сумма квадратов остатков на обучающих данных; *J* – количество настраиваемых параметров; *n* – размер обучающей выборки; *p* – сложность пространства моделей (произведение размера дерева на количество объясняющих переменных).

Поиск новых решений происходит в два этапа. На первом этапе в выбранном решении осуществляется замена поддерева на поддерево, выбранное случайным образом из совокупности всех решений ***S***. На втором этапе к случайно выбранному узлу в новом поддереве применяется одна из следующих операций: удаление из регрессионного узла наименее значимого фактора; добавление в регрессионный узел нового фактора; расщепление регрессионного узла; замена правила разделения; замена поддерева на лист с регрессионной моделью.

В качестве критерия останова используется достижение максимального числа итераций или достижение заданного числа итераций без изменения лучшего найденного решения.

Разработанный алгоритм (далее ABCRT) был протестирован на 6 наборах данных из UC Irvine Machine Learning Repository. Также было

выполнено его сравнение с двумя классическими жадными алгоритмами построения деревьев моделей: M5 [2] и RETIS [3]. Все результаты приведены в Таблице 1. Они получены с помощью 10-слойной перекрёстной проверки и усреднены по 30 запускам.

Таблица 1 – Результаты сравнения разработанного алгоритма с аналогами.

Набор данных	Алгоритм	RMSE	Размер дерева	Количество параметров	Время, с
autompg	M5`	3.08 ± 0.08	32.0 ± 2.1	38.0 ± 1.9	**0.22 ± 0.03**
	RETTIS	3.52 ± 0.20	37.1 ± 2.3	99.5 ± 6.3	4.01 ± 0.09
	ABCRT	**3.03 ± 0.05**	**3.8 ± 0.2**	**7.3 ± 0.3**	75.90 ± 4.95
housing	M5`	4.10 ± 0.13	36.5 ± 1.8	41.7 ± 1.4	**0.74 ± 0.02**
	RETTIS	4.74 ± 1.18	29.0 ± 2.2	115.8 ± 9.3	22.56 ± 0.31
	ABCRT	**3.95 ± 0.20**	**6.3 ± 0.5**	**14.7 ± 0.6**	245.38 ± 7.17
stock	M5`	1.47 ± 0.04	35.5 ± 1.7	47.6 ± 1.4	**0.61 ± 0.00**
	RETTIS	**0.89 ± 0.05**	35.3 ± 1.7	122.6 ± 5.0	11.60 ± 0.12
	ABCRT	1.13 ± 0.06	**8.4 ± 0.5**	**25.4 ± 1.1**	223.94 ± 11.16
machine	M5`	52.86 ± 3.73	21.3 ± 2.7	26.0 ± 2.7	**0.11 ± 0.00**
	RETTIS	94.25 ± 97.89	21.4 ± 2.7	53.9 ± 6.6	2.48 ± 0.04
	ABCRT	**46.13 ± 2.61**	**3.9 ± 0.3**	**8.8 ± 0.4**	54.54 ± 4.04
abalone	M5`	2.20 ± 0.03	21.4 ± 1.4	31.5 ± 1.2	**3.29 ± 0.02**
	RETTIS	2.25 ± 0.1	52.3 ± 1.9	129.7 ± 4.0	8.02 ± 0.11
	ABCRT	**2.16 ± 0.01**	**4.7 ± 0.3**	**14.4 ± 0.3**	432.34 ± 24.74
ailerons	M5`	0.000169 ± 0.0	**4.3 ± 0.3**	**16.8 ± 0.8**	**7.09 ± 0.04**
	RETTIS	**0.000163 ± 0.0**	57.3 ± 0.8	239.9 ± 3.2	105.33 ± 6.21
	ABCRT	0.000164 ± 0.0	5.6 ± 0.3	24.1 ± 0.5	4313.25 ± 97.88

Если говорить о точностных характеристиках, то разработанный алгоритм оказался лучшим на 4 из 6 наборах данных, уменьшение RMSE составило от 2% до 12%. На наборе «ailerons» он и алгоритм RETTIS показали очень близкие результаты. И лишь на наборе «stock» разработанный алгоритм уступил алгоритму RETTIS.

Если говорить о сложности полученных моделей, то на 5 из 6 наборах данных разработанный алгоритм позволил уменьшить сложность деревьев моделей в 4 – 10 раз. Лишь на наборе «ailerons» он уступил алгоритму M5, построив модели в среднем в 1,4 раза сложнее. Однако стоит отметить, что на этом наборе данных M5 уступает разработанному алгоритму по точностным характеристикам.

Таким образом результаты численных экспериментов показывают,

что разработанный алгоритм построения деревьев моделей с внутренними регрессионными узлами на основе моделирования поведения колонии пчел превосходит традиционные алгоритмы в терминах адекватности моделей, а также позволяет получать более простые деревья регрессии. Однако ценой за это служит значительное увеличение времени исполнения алгоритма.

Литература:

1. Chen J., Chen Z. Extended Bayesian information criteria for model selection with large model spaces // Biometrika. 2008. V. 95, № 3. P. 759–771.
2. Quinlan J.R. Learning with continuous classes / J.R. Quinlan // Proc. AI'92, 5th Australian Joint Conference on Artificial Intelligence, Singapore. – 1992. – P. 343-348.
3. Karalic A. Employing linear regression in regression tree leaves / A. Karalic // Proceedings of the 10th European Conference on Artificial Intelligence / B. Neumann. – Vienna: Wiley, 1992. – P. 440-441.

Algorithm for induction of model tree with internal regression nodes based on bee colony foraging behavior

Grigoriy A. Melnikov, Andrey V. Gavrilov

Novosibirsk State Technical University

This paper describes a novel method for induction of model tree with internal regression nodes based on bee colony foraging behavior. The results of experiments on publicly available data sets show that the proposed algorithm outperforms conventional algorithms for regression tree induction in accuracy and results in significantly less complex solutions.

Keywords: piecewise regression, regression tree, regression model, artificial bee colony algorithm.