УДК 004.85

Г. А. Мельников, аспирант, **Т. А. Мельников,** магистрант, **В. В. Губарев,** д-р техн. наук, проф.,
Новосибирский государственный технический университет, г. Новосибирск

Алгоритмы упрощения деревьев регрессии: обзор и эмпирическое сравнение

Дан обзор и выполнена систематизация существующих алгоритмов упрощения деревьев регрессии. Также проведено эмпирическое сравнение пяти ключевых алгоритмов упрощения по трем показателям: время работы, адекватность полученных моделей и их сложность. Результаты экспериментов показывают, что в отличие от деревьев классификации, где себя хорошо зарекомендовали алгоритмы упрощения на основе отсечения ветвей, для деревьев регрессии более предпочтительны алгоритмы ранней остановки. Последние значительно менее трудоемки и строят модели, в большинстве случаев обладающие лучшей адекватностью при сопоставимой сложности.

Ключевые слова: *интеллектуальный анализ данных, машинное обучение, нелинейная регрессия, кусочно-заданные модели, деревья моделей, деревья регрессии, упрощение деревьев регрессии*

Введение

Деревья регрессии являются одним из важных классов регрессионных моделей, позволяющим представить кусочно-заданную функцию регрессии в интуитивно понятной и наглядной форме в виде дерева принятия решений. В таком дереве внутренние узлы содержат правила разделения пространства объясняющих переменных X, дуги — условия перехода по ним, а листья — локальные регрессионные модели. Впервые этот подход был реализован в алгоритме AID [1] в 1963 г. Однако популярны деревья регрессии стали лишь после публикации в 1984 г. работы [2] и появления алгоритма CART, который и по сей день входит практически в любой пакет статистического анализа данных.

Задача построения деревьев регрессии является *NP*-сложной [3]. При ее решении необходимо ответить на три основных вопроса:

- Каким образом разделить данные на сегменты?
- Как и какого типа локальные регрессионные модели строить в листьях дерева?
- Каким должен быть размер дерева регрессии?

Исследования в рассматриваемой области главным образом направлены на поиск новых правил разделения данных. Большинство алгоритмов в листьях дерева строят линейные локальные модели (видимо вследствие их простоты и наглядности) и лишь некоторые — константы или полиномы. Третий вопрос хоть и затрагивается во всех работах, но специально ему внимания практически не уделяется.

Разделять пространство объясняющих переменных X можно до тех пор, пока в узле останется лишь один обучающий пример (элемент обучающей выборки). Однако с уменьшением числа примеров в узлах результат становится менее статистически значим, модель начинает "вбирать" в себя шумы, присутствующие в данных. Такая чрезмерно точная настройка модели на обучающую выборку, как правило, приводит к переобучению (рис. 1). Помимо переобучения, с увеличением размера деревьев регрессии теряется одно из главных их достоинств — простота интерпретации. Вследствие изложенных выше причин задача упрощения дерева регрессии, т.е. *задача выбора "правильного" размера дерева*, встает особенно остро.

Многие исследователи в данной области используют алгоритмы упрощения деревьев классификации. Некоторые предлагают свои, но их эффективность оценить сложно независимо от введенных ими правил разделения данных. Здесь можно выделить лишь одну работу [4], которая полностью посвящена эмпирическому сравнению алгоритмов упрощения деревьев регрессии. Однако в ней рассматриваются только три алгоритма, два из которых предложены непосредственно авторами статьи, а такой популярный алгоритм, как отсечение ветвей на основе оценки цена—сложность проигнорирован. Это, по нашему мнению, является существенным упущением. Кроме того, в последние годы появились новые алгоритмы ранней остановки [5, 6], которые гипотетически могут существенно снизить трудоемкость построения деревьев регрессии.

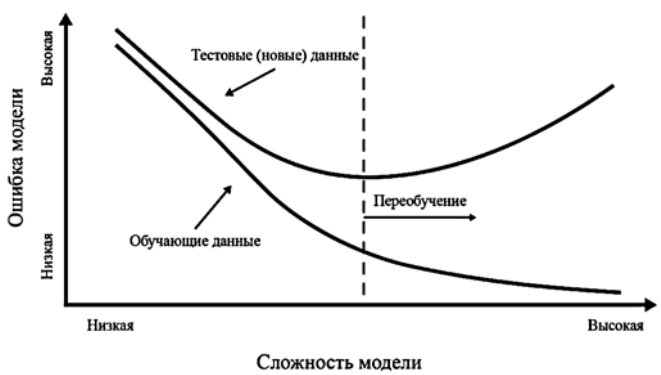

Рис. 1. Иллюстрация явления переобучения

Цель данной работы — обзор, систематизация и сравнение современных алгоритмов упрощения деревьев регрессии, а также определение их роли и значимости при построении деревьев регрессии.

Обзор алгоритмов упрощения деревьев регрессии

Можно выделить два основных подхода к упрощению деревьев регрессии:

- отсечение ветвей, которое предполагает построение дерева регрессии максимального размера, затем упрощается снизу вверх путем преобразования узлов в листья;
- ранняя остановка, которая предполагает ограничение роста дерева регрессии на этапе его построения путем прекращения разделения данных при достижении некоторого условия.

Отсечение ветвей. Подход на основе отсечения ветвей хорошо зарекомендовал себя при построении деревьев классификации [7] и был использован в большинстве ранних алгоритмов построения деревьев регрессии [2, 8—10]. Здесь можно выделить три основных алгоритма: упрощение на основе показателя цена—сложность (Cost-Complexity Pruning) [8], алгоритмы упрощения из M5 [9] и RETIS [10]. Как было указано ранее, во всех этих алгоритмах сначала строится дерево регрессии максимального размера, а затем уже осуществляется его упрощение.

Упрощение на основе показателя цена—сложность происходит в два этапа. На первом из полного (максимального размера) дерева регрессии строится последовательность деревьев регрессии уменьшающегося размера, минимизирующая при различных значениях параметра γ значение показателя цена—сложность*:

$$R(T, \gamma) = R(T) + \gamma|T|, \qquad (1)$$

где $R(T)$ — средняя квадратическая ошибка дерева регрессии T; $|T|$ — число листьев дерева или штраф за сложность дерева; γ — параметр регуляризации, контролирующий вклад штрафа в общую оценку. На втором этапе адекватность деревьев регрессии оценивается на дополнительной независимой (валидационной) выборке, и дерево регрессии с наименьшей средней квадратической ошибкой и соответствующее ему значение параметра $\gamma = \gamma^*$ выбираются как итоговые.

Оптимальное значение параметра $\gamma = \gamma^*$ может быть оценено с помощью перекрестной проверки. После чего полное дерево регрессии строится уже на всех обучающих данных и из него выбирается поддерево, минимизирующее (1) при $\gamma = \gamma^*$. В этом случае необходимость в дополнительной независимой выборке данных пропадает, однако возрастает трудоемкость.

В *алгоритме M5* для каждого узла на обучающем множестве вычисляется средняя квадратическая ошибка локальной модели. Чтобы учесть сложность модели и число обучающих примеров, по которым она была построена, значение ошибки умножается на поправочный коэффициент:

$$k = \frac{n + mv}{n - v}, \qquad (2)$$

где n — число обучающих примеров в узле; v — число параметров модели; m — свободный параметр, контролируемый пользователем. Затем при движении снизу вверх узлы преобразовываются в листья до тех пор, пока скорректированная ошибка уменьшается.

В *RETIS* алгоритм упрощения деревьев регрессии похож на предыдущий, но скорректированная ошибка для каждого узла вычисляется *на основе байесовского подхода*, который комбинирует априорные и апостериорные знания. В этом случае показатель адекватности модели φ определяется как

$$adjR(\varphi) = \frac{m}{n + m} R_{\alpha}(\varphi) + \frac{n}{n + m} R_{\varphi}, \qquad (3)$$

где n — число обучающих примеров в узле; $R_{\alpha}(\varphi)$ — средняя квадратическая ошибка модели φ, вычисленная на всех обучающих примерах (а не только в текущем узле); R_{φ} — средняя квадратическая ошибка модели φ, вычисленная на обучающих примерах в текущем узле; m — контролируемый пользователем свободный параметр.

Ранняя остановка. Отсечение ветвей значительно более трудоемко, чем ранняя остановка, так как требует построения полного дерева регрессии. Поэтому работы последних лет сосредоточены в основном на использовании второго подхода — ранней остановки [5, 6, 11, 12].

Самым распространенным [2, 5, 6, 8—12] правилом ранней остановки является ограничение на минимальное число примеров в узле. Практика использования деревьев классификации показала, что данное правило не робастно [7]. И хотя его используют практически все алгоритмы, обычно его применяют совместно с другими правилами упрощения деревьев.

Большинство правил ранней остановки связаны с оценкой адекватности модели на ранее неизвестных данных. Согласно [4—6, 11, 12] здесь можно выделить следующие два основных подхода.

1. Использование *валидационного множества* [4, 12]. При очередном расщеплении узла на независимой выборке сравниваются средние квадратические ошибки моделей до и после разделения данных. Если средняя квадратическая ошибка увеличилась, то разделение отменяется и алгоритм индукции останавливается.

* Детали процесса нахождения последовательности значений параметра γ и соответствующих им деревьев регрессии приведены в [2].

2. Оценка адекватности модели непосредственно из ее характеристик лишь на обучающем множестве [5, 6, 11]. Главным образом, это использование статистических тестов. Потенциально здесь возможно использование информационных критериев выбора моделей (Колмогоровская сложность).

Рассмотрим второй подход более детально. В работе [5] проверяется статистическая гипотеза о том, что все данные в узле порождены некоторым скрытым линейным процессом. Пусть есть три линейных модели: одна построена на всех N обучающих примерах, вторая — на N_L примерах, лежащих слева от точки разделения, и третья — на N_R примерах, лежащих справа. Тогда, если альтернативная гипотеза верна, то сумма квадратов остатков после разделения данных должна быть значительно меньше суммы квадратов остатков до разделения. Это может быть проверено с помощью *теста Чоу*:

$$F = \frac{(RSS - RSS_L - RSS_R)(N - 2d)}{(RSS_L + RSS_R)d}, \qquad (4)$$

где F — это F-статистика с d и $N - 2d$ степенями свободы; d — число объясняющих переменных; RSS, RSS_L и RSS_R — сумма квадратов остатков модели до разделения данных и после соответственно. Разделение данных должно быть осуществлено только если p-значение меньше некоторого порогового значения α (т.е. нулевая гипотеза отвергается на уровне значимости α), заданного пользователем.

Эмпирическое сравнение алгоритмов упрощения деревьев регрессии

Рассматриваемые алгоритмы упрощения деревьев регрессии были протестированы на двух синтетических наборах данных из работы [13] и восьми наборах данных из UC Irvine Machine Learning Repository [14] и KEEL-dataset repository [15]. Их краткая характеристика представлена в табл. 1.

Чтобы поставить все тестируемые алгоритмы упрощения в равные условия, в каждом случае использовался один и тот же жадный метод построения деревьев регрессии на основе рекурсивного разделения данных. На текущий момент он является самым распространенным методом построения деревьев регрессии. Кратко его можно описать следующим образом.

Таблица 1

Краткая характеристика рассматриваемых наборов данных

Название набора данных	Объем выборки	Число объясняющих переменных
Armchair	1000	2
Split plane	1000	1
Abalone	4177	8
Ailerons	13 750	40
Auto-mpg	392	7
CPU	209	6
Housing	506	14
Stock	950	10
Breast Cancer Wisconsin	198	34
Triazines	186	61

1. Выбор "лучшего" разделения данных C (объясняющей переменной и точки разделения a или разделяющего множества A), как правило, такого, которое обеспечивает экстремум некоторого критерия R.

2. Разделение данных на подмножества.

3. Рекурсивное применение шагов 1—3 к каждому из подмножеств.

В качестве критерия выбора модельного разделения данных была использована минимизация взвешенной суммы квадратов отклонений локальных моделей:

$$R(T, C) = \frac{N_L}{N} \sum_{i \in I_L} (y_i - g_L(x_i))^2 + \frac{N_R}{N} \sum_{i \in I_R} (y_i - g_R(x_i))^2, \qquad (5)$$

где T — исходный набор данных; T_L и T_R — наборы, образованные путем разделения T по C (т. е. по a или A); N, N_L и N_R — число элементов в каждом из наборов; I_L и I_R — индексы принадлежности элементов к T_L и T_R соответственно; g_L и g_R — локальные модели для T_L и T_R соответственно. Данный критерий является довольно трудоемким, поэтому для каждой переменной мы проверяли лишь 20 разделений равномерно распределенных на области ее значений. В качестве локальных моделей была использована множественная линейная регрессия. Ее построение осуществлялось с помощью алгоритма пошаговой регрессии с использованием Байесовского информационного критерия для отбора переменных. Для всех алгоритмов упрощения минимальное число примеров в листе было установлено равным 5 % от числа обучающих данных. Значение свободного параметра m алгоритмов упрощения из M5 и RETIS в обоих случаях было установлено равным 2, согласно рекомендациям [9] и [10], соответственно. Для алгоритмов упрощения на основе оценки цена—сложность и ранней остановки по ошибке на валидационном множестве в качестве валидационного множества использовалось 30 % от обучающей выборки.

Все алгоритмы были реализованы на языке программирования Matlab и протестированы в среде Matlab R2007b на ПК со следующей конфигурацией: Intel Core 2 Duo E6600 и 4 GB RAM.

Результаты эксперимента приведены в табл. 2. Они получены с помощью 10-слойной перекрестной проверки и усреднены по 30 запускам. В качестве показателя адекватности полученных моделей был использован квадратный корень из средней квадратической ошибки (RMSE). Для оценки сложности моделей использовали число вершин в дереве регрессии. В табл. 2 также приведено время построения (в секундах) дерева регрессии.

Для всех значений в табл. 2 приведено среднее квадратическое отклонение. Однако интервальные

Таблица 2

Эмпирическое сравнение алгоритмов упрощения деревьев регрессии

Данные	Показатели	Алгоритмы				
		M5	RETIS	Цена—сложность	Валидационное множество	Тест Чоу
ArmChar	RMSE Сложность Время, с	0,47 ± 0,06 19,6 ± 1,7 0,39 ± 0,01	0,25 ± 0,059 **13,1 ± 1,1** 0,29 ± 0,02	0,18 ± 0,028 16,5 ± 0,9 0,44 ± 0,06	0,19 ± 0,042 17,5 ± 0,8 **0,25 ± 0,01**	**0,15 ± 0,016** 15,9 ± 0,6 0,27 ± 0,02
SplitPlane	RMSE Сложность Время, с	**0,002 ± 0,003** 4,5 ± 0,4 0,24 ± 0,02	0,003 ± 0,003 4,29 ± 0,56 0,05 ± 0,01	**0,002 ± 0,002** 4,49 ± 0,51 0,06 ± 0,01	**0,002 ± 0,002** 4,4 ± 0,59 **0,05 ± 0,01**	0,003 ± 0,003 **4,27 ± 0,56** **0,05 ± 0,01**
Abalone	RMSE Сложность Время, с	2,17 ± 0,04 28,1 ± 0,7 7,6 ± 0,1	2,16 ± 0,02 27,6 ± 0,5 9,0 ± 0,2	2,22 ± 0,02 **6,5 ± 2,8** 10,4 ± 0,2	2,16 ± 0,01 7,4 ± 1,3 **5,3 ± 0,5**	**2,15 ± 0,02** 19,4 ± 0,7 9,0 ± 0,3
Ailerons	RMSE Сложность Время, с	**0,000162 ± 0,0** 27,4 ± 0,6 104,1 ± 4,4	**0,000162 ± 0,0** 28,4 ± 0,6 129,6 ± 14,7	0,000165 ± 0,0 **10,0 ± 2,5** 129,3 ± 6,9	0,000163 ± 0,0 14,7 ± 1,3 **90,3 ± 4,7**	0,000163 ± 0,0 18,4 ± 0,9 124,5 ± 12,8
Auto-mpg	RMSE Сложность Время, с	3,26 ± 0,16 30,5 ± 0,6 4,15 ± 0,1	3,23 ± 0,16 25,6 ± 1,0 3,9 ± 0,1	3,35 ± 0,09 **4,4 ± 2,8** 5,1 ± 0,3	**3,11 ± 0,1** 5,9 ± 1,1 **1,93 ± 0,21**	3,22 ± 0,15 19,5 ± 1,1 4,0 ± 0,4
Housing	RMSE Сложность Время, с	4,82 ± 1,64 30,5± 0,5 27,5 ± 2,0	4,90 ± 1,53 20,9 ± 1,3 26,3 ± 0,9	5,08 ± 1,76 **5,87 ± 2,2** 30,3 ± 0,9	**4,78 ± 1,06** 7,3 ± 1,2 **13,6 ± 1,6**	4,93 ± 1,66 20,3 ± 0,8 28,5 ± 0,8
Machine	RMSE Сложность Время, с	68,84 ± 38,54 30,4± 0,5 2,9 ± 0,1	**55,11 ± 6,56** 17,4 ± 1,3 3,0 ± 0,1	61,17 ± 9,33 **2,65 ± 0,8** 3,6 ± 0,1	56,68 ± 7,61 4,0 ± 0,9 **1,1 ± 0,1**	56,48 ± 6,67 16,7 ± 1,1 2,9 ± 0,1
Stock	RMSE Сложность Время, с	0,89 ± 0,05 29,9 ± 0,4 13,4 ± 0,3	0,90 ± 0,04 22,8 ± 0,9 14,5 ± 0,5	1,06 ± 0,09 **16,9 ± 2,3** 17,1 ± 0,2	1,05 ± 0,08 19,9 ± 1,4 **12,7 ± 0,4**	**0,88 ± 0,037** 29,8 ± 0,6 15,8 ± 0,4
Breast	RMSE Сложность Время, с	77,21 ± 7,92 29,5 ± 0,6 320,0 ± 30,2	65,96 ± 3,69 21,1 ± 0,9 436,4 ± 90,6	30,59 ± 0,8 **1,0 ± 0,0** 370,4 ± 47,7	31,1 ± 1,21 1,1 ± 0,1 **23,3 ± 2,1**	**30,33 ± 0,44** **1,0 ± 0,0** 37,0 ± 3,5
Triazine	RMSE Сложность Время, с	0,54 ± 0,76 28,9 ± 0,7 24,5 ± 0,6	0,17 ± 0,02 19,8 ± 1,1 23,2 ± 1,0	0,16 ± 0,03 2,0 ± 1,0 24,3 ± 0,9	0,16 ± 0,03 3,2 ± 0,7 8,9 ± 1,3	**0,14 ± 0,0** **1,0 ± 0,0** **4,9 ± 0,2**

оценки очень трудны для анализа и не наглядны. Поэтому для сравнения алгоритмов по показателю RMSE, значения которого сравнительно близки у различных алгоритмов, применяли статистические тесты и все выводы осуществляли на их основе. Сложность же получаемых моделей и время исполнения сильно различаются между сравниваемыми группами алгоритмов. При этом среднее квадратическое отклонение для этих показателей обычно составляет менее 10 % от их значения. Поэтому для их сравнения достаточно проанализировать лишь средние значения.

Чтобы выделить лишь статистически значимые различия по показателю RMSE, попарно сравним исследуемые алгоритмы с помощью непараметрического статистического теста Уилкоксона [16] (табл. 3). Согласно табл. 3 алгоритм упрощения на основе теста Чоу строит модели, обладающие наименьшей ошибкой аппроксимации данных (в 22 случаях из 40 наблюдаются значимо меньшие значения показателя RMSE). Далее практически с одинаковыми результатами идут алгоритм ранней остановки по ошибке на валидационном множестве и

Таблица 3

Сравнение RMSE моделей с помощью теста Уилкоксона ($\alpha = 0{,}05$)

Алгоритм	M5	RETIS	Цена—сложность	Валидационное множество	Тест Чоу	Всего побед
M5	—	0	4	2	1	7
RETIS	4	—	5	2	1	12
Цена—сложность	3	3	—	0	0	6
Валидационное множество	5	4	4	—	1	14
Тест Чоу	5	5	7	5	—	22
Всего поражений	17	12	20	9	3	

Примечание. Под победой здесь понимается статистически значимое меньшее среднее значение показателя RMSE, а под поражением, соответственно статистически значимое большее

алгоритм упрощения на основе Байесовского подхода (14 и 12 побед, соответственно). Аутсайдерами здесь являются алгоритм упрощения на основе оценки цена—сложность и алгоритм M5 (7 и 6 побед, соответственно). Таким образом, когда интерес представляют исключительно прогностические свойства моделей, то безусловным фаворитом здесь является алгоритм ранней остановки на основе теста Чоу.

Вместе с тем с помощью алгоритма упрощения на основе оценки цена—сложность создаются наиболее простые модели (рис. 2). Они в среднем в 3,5 раза проще, чем модели, полученные с помощью алгоритма упрощения M5; в 3 раза проще, чем при упрощении на основе Байесовского подхода; в 2 раза проще, чем при использовании теста Чоу; и в среднем имеют размер на 1,5 узла меньше, чем при использовании ранней остановки по ошибке на валидационном множестве. Поэтому здесь необходимо понять: значимо ли уменьшение RMSE при наблюдаемом увеличении сложности модели? Для этого можно воспользоваться F-тестом [17], применив его ко всему дереву регрессии на тестовых данных (табл. 4). Из табл. 4 видно, что при использовании алгоритма упрощения на основе теста Чоу рост сложности моделей по сравнению с алгоритмами упрощения на основе оценки цена—сложность и валидационного множества оправдан лишь в трех из восьми случаев и на двух наборах данных были получены более компактные модели, обладающие меньшей ошибкой. Поэтому последние два алгоритма можно рекомендовать, когда необходимо получить простые и интерпретируемые модели.

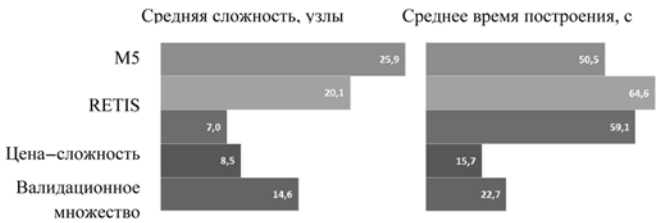

Рис. 2. Усредненные по всем наборам данных значения показателей

Таблица 4

Сравнение моделей с помощью F-теста ($\alpha = 0{,}05$)

Алгоритм	M5	RETIS	Цена—сложность	Валидационное множество	Тест Чоу	Всего побед
M5	—	2	4	2	1	9
RETIS	8	—	3	2	1	14
Цена—сложность	6	7	—	5	5	23
Валидационное множество	8	8	5	—	5	26
Тест Чоу	9	9	5	5	—	28
Всего поражений	31	26	17	14	12	

Примечание. Под победой здесь понимается, что модель согласно F-тесту статистически значимо лучше; в обратном случае имеет место поражение.

Если же сравнивать скорость работы алгоритмов упрощения деревьев регрессии, то самым быстрым вариантом является ранняя остановка по ошибке на валидационном множестве. Данный алгоритм в среднем в 1,4—4 раза быстрее остальных алгоритмов. На втором месте идет алгоритм ранней остановки на основе теста Чоу, в среднем проигрывающий первому лишь 7 с. Остальные три алгоритма упрощения по времени работают примерно одинаково. В среднем они приблизительно в 3 раза медленнее алгоритмов ранней остановки.

Анализируя рис. 2 и табл. 3, заключаем, что алгоритмы упрощения деревьев регрессии из M5 и RETIS строят неоправданно сложные модели. При этом они не дают никаких конкурентных преимуществ по сравнению с другими алгоритмами ни по точностным характеристикам, ни по скорости работы. Но здесь необходимо отметить, что данные алгоритмы имеют настраиваемые параметры, регулирующие их работу, влияние которых мы никак не рассматривали. И при значительных дополнительных временных затратах эти алгоритмы можно настроить под каждый из рассматриваемых наборов данных.

Заключение

В работе, во-первых, выполнены обзор и систематизация существующих алгоритмов упрощения деревьев регрессии. Во-вторых, проведено эмпирическое сравнение пяти ключевых алгоритмов: M5, RETIS, упрощения на основе показателя цена—сложность, ранней остановки по ошибке на валидационном множестве и на основе теста Чоу. По результатам экспериментов сделаны следующие выводы.

1. Упрощение деревьев регрессии является одним из ключевых шагов как для получения простых и интерпретируемых моделей, так и для достижения высоких точностных характеристик.

2. В отличие от деревьев классификации, где себя хорошо зарекомендовали алгоритмы упрощения на основе отсечения ветвей, для деревьев регрессии более предпочтительны алгоритмы ранней остановки. На рассмотренных наборах данных:

а) последние в среднем в 3 раза быстрее;

б) ранняя остановка по тесту Чоу чаще остальных алгоритмов приводит к получению моделей с наименьшей ошибкой аппроксимации данных;

в) ранняя остановка по ошибке на валидационном множестве строит деревья регрессии в среднем лишь на 1,5 узла больше самых компактных моделей, а по ошибке аппроксимации данных идет сразу за упрощением по тесту Чоу.

3. Алгоритмы упрощения на основе отсечения ветвей M5 и RETIS строят неоправданно сложные

модели и при этом не дают никаких конкурентных преимуществ.

4. При отсечении ветвей на основе оценки цена—сложность и ранняя остановка по ошибке на валидационном множестве строятся самые компактные деревья регрессии. При этом почти на половине исследуемых наборов данных рост сложности, получаемый при использовании теста Чоу, не оправдан. Поэтому эти два алгоритма можно рекомендовать, когда необходимо получить простые интерпретируемые модели. А так как ранняя остановка по ошибке на валидационном множестве почти в 4 раза быстрее, то ее стоит предпочесть отсечению ветвей на основе оценки цена—сложность.

5. Ранняя остановка по тесту Чоу чаще остальных алгоритмов приводит к получению моделей с наименьшей ошибкой аппроксимации данных, но по сравнению с отсечением ветвей на основе оценки цена—сложность и ранней остановкой по ошибке на валидационном множестве, наблюдается двукратное усложнение моделей, что не всегда оправдано. Поэтому данный алгоритм можно рекомендовать, в первую очередь, когда интересны прогностические свойства получаемых моделей.

Список литературы

1. **Morgan J. N., Sonquist J. A.** Problems in the analysis of survey data, and a proposal // J. Amer. Statist. Assoc., 1963. N. 58. P. 415—434.
2. **Breiman L., Friedman J. H., Olshen R. A., Stone C. J.** Classification and Regression Trees. Belmont: Wadsworth International Group, 1984. 259 p.
3. **Hyafil L., Rivest R. L.** Constructing optimal binary decision trees is NP-complete // Information Processing Letters. 1976. N. 5 (1). P. 15—17.
4. **Torgo L.** A Comparative Study of Reliable Error Estimators for Pruning Regression Trees // In Proceeding of the Iberoamerican Conference on Artificial Intelligence. Springer-Verlag, Coelho, 1998.
5. **Potts D., Sammut C.** Incremental Learning of Linear Model Trees // Machine Learning. 2005. Vol. 61. P. 5—48.
6. **Vogel D., Asparouhov O., Scheffer T.** Scalable look-ahead linear regression trees // Proc. of 13th ACM SIGKDD. New York: ACM Press, 2007. P. 757—764.
7. **Sreerama K. Murthy.** Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey // Data Mining and Knowledge Discovery. 2005. Vol. 2, N. 4. P. 345—389.
8. **Loh W.-Y.** Regression trees with unbiased variable selection and interaction detection // Statistica Sinica. 2002. Vol. 12. P. 361—386.
9. **Quinlan J. R.** Learning with continuous classes // Proc. AI'92, 5th Australian Joint Conference on Artificial Intelligence. Singapore: World Scientific, 1992. P. 343—348.
10. **Karalic A.** Employing linear regression in regression tree leaves // Proc. of the 10th European Conference on Artificial Intelligence / Ed. B. Neumann. Vienna: Wiley, 1992. P. 440—441.
11. **Malerba D., Esposito F., Ceci M., Appice A.** Top-down induction of model trees with regression and splitting nodes // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004. N. 26. P. 612—625.
12. **Melnikov G. A., Gubarev V. V.** Ant Colony Based Semi-Greedy Algorithm for Regression Tree Induction // Proc. of the 8-th International forum on strategic technology 2013, (IFOST 2013), Mongolia, Ulaanbaatar, 28 June — 1 July 2013. Ulaanbaatar, 2013. Vol. II. P. 238—240.
13. **Мельников Г. А.** Применение методов искусственного интеллекта для исследования инфекционных заболеваний: Магистерская дис. ... "Магистр техники и технологии": 230100. Новосибирск, 2012. 141 с.
14. **Frank A., Asuncion A.** UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science, 2010. [http://archive.ics.uci.edu/ml]
15. **Alcalá-Fdez J., Fernandez A., Luengo J., Derrac J., García S., Sánchez L., Herrera F.** KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework // Journal of Multiple-Valued Logic and Soft Computing. 2011. Vol. 17. P. 255.
16. **Wilcoxon F.** Individual comparisons by ranking methods // Biometrics. International Biometric Society. 1945. Vol. 1, N. 6. P. 80—83.
17. **Green W. H.** Econometric analysis. 5th ed. Bearson education, 2003. 1056 p.

G. A. Melnikov, Postgraduate student, grmel89@gmail.com,
T. A. Melnikov, Master student, temmelnik@gmail.com,
V. V. Gubarev, PhD, Professor of Department of Computer Engineering
Novosibirsk State Technical University

Regression Tree Pruning Algorithms: an Overview and Empirical Comparison

Regression trees belong to a very important class of regression models which allows to split feature space into segments with building specialized local model for each of them and to achieve visualizable, easy interpretable and accurate piecewise models. As for the classification tree, choosing the right size of the tree is one of the key issues of regression tree induction. Unfortunately, this issue is given very little attention. The majority of works focused on the development of data splitting algorithms.

The first part of the paper gives an overview and systematization of existing regression tree pruning algorithms. These algorithms can be divided into two standard groups: pre-pruning and post-pruning. Most authors follow the best practices of classification trees induction algorithms and use cost-complexity pruning or design their own post-pruning algorithms. There are only few works where used pre-pruning. Is worth noting here only two algorithms from this group: the first uses validation dataset to estimate generalization error and the second are based on the Chow test. In the second part of the paper, we conducted an empirical comparison of five key pruning algorithms in three indicators: running time of the algorithms, adequacy and complexity of the obtained models. The results of experiments show that, unlike the case of the classification trees, pre-pruning algorithms are more preferred to post-pruning algorithms in regression tree induction. The first are much less time-consuming, induction time decreased on average by three times. In addition, the pre-pruning algorithms induct models that in most cases has a better adequacy and complexity comparable with that of the best post-pruning models.

Future work should be focused on the development pre-pruning algorithms of regression tree induction. Of particular interest, in our opinion, should be given to adaptation of statistical tests and information criteria of model selection.

Keywords: *data mining, machine learning, non-linear regression, piecewise models, model trees, regression trees, regression tree pruning*

References

1. **Morgan J. N., Sonquist J. A.** Problems in the analysis of survey data, and a proposal, *J. Amer. Statist. Assoc.*, 1963, no. 58, pp. 415—434.
2. **Breiman L., Friedman J. H., Olshen R. A., Stone C. J.** *Classification and Regression Trees,* Wadsworth International Group, Belmont, 1984, 259 p.
3. **Hyafil L., Rivest R. L.** Constructing optimal binary decision trees is NP-complete, *Information Processing Letters*, 1976, no. 5 (1), pp. 15—17.
4. **Torgo L.** A Comparative Study of Reliable Error Estimators for Pruning Regression Trees, *Proceeding of the Iberoamerican Conference on Artificial Intelligence*, Springer-Verlag, Coelho, 1998.
5. **Potts D., Sammut C.** Incremental Learning of Linear Model Trees, *Machine Learning*, 2005, vol. 61, pp. 5—48.
6. **Vogel D., Asparouhov O., Scheffer T.** Scalable look-ahead linear regression trees, *Proc. of 13th ACM SIGKDD*, New York, ACM Press, 2007, pp. 757—764.
7. **Murthy Sreerama K.** Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey, *Data Mining and Knowledge Discovery*, 2005, vol. 2, no. 4, Kluwer Academic Publishers, 2005, pp. 345—389.
8. **Loh W.-Y.** *Regression trees with unbiased variable selection and interaction detection*, Statistica Sinica, vol. 12, 2002, pp. 361—386.
9. **Quinlan J. R.** Learning with continuous classes, *Proc. AI'92, 5th Australian Joint Conference on Artificial Intelligence, Singapore, World Scientific, 1992*, pp. 343—348.
10. **Karalic A.** Employing linear regression in regression tree leaves, *Proc. of the 10th European Conference on Artificial Intelligence*, ed. B. Neumann, Vienna, Wiley, 1992, pp. 440—441.
11. **Malerba D., Esposito F., Ceci M., Appice A.** Top-down induction of model trees with regression and splitting nodes. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004*, no. 26, pp. 612—625.
12. **Melnikov G. A., Gubarev V. V.** Ant Colony Based Semi-Greedy Algorithm for Regression Tree Induction, *Proc. of the 8-th International forum on strategic technology 2013, (IFOST 2013), Mongolia, Ulaanbaatar, 28 June — 1 July 2013, — Ulaanbaatar, 2013*, vol. II, pp. 238—240.
13. **Melnikov G. A.** *Primenenie metodov iskusstvennogo intellekta dlya issledovaniya infektsionnykh zabolevaniy: magisterskaya dis. ...* "Magistr tekhniki i tekhnologii": 230100. Novosibirsk, 2012, 141 p. (Melnikov G. A. Application of artificial intelligence methods for the study of infectious diseases: Master's thesis... "Master of engineering and technology": 230100 / Melnikov Grigoriy Andreevich, Novosibirsk, 2012. 141 p.)
14. **Frank A., Asuncion A.** *UCI Machine Learning Repository* [http://archive.ics.uci.edu/ml] / Irvine, CA: University of California, School of Information and Computer Science, 2010.
15. **Alcalá-Fdez J., Fernandez A., Luengo J., Derrac J., García S., Sánchez L., Herrera F.** KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework, *Journal of Multiple-Valued Logic and Soft Computing*, 2011, vol. 17, pp. 255.
16. **Wilcoxon F.** Individual comparisons by ranking methods, *Biometrics*, 1945, vol. 1, no. 6, International Biometric Society, pp. 80—83.
17. **Green W. H.** Econometric analysis, 5th ed., Bearson education, 2003, 1056 p.