УДК 004.85

Итеративный метод построения деревьев регрессии

Г. А. Мельников, В. В. Губарев

Подавляющее большинство современных алгоритмов построения деревьев регрессии являются жадными. Все они основаны на рекурсивном разделении данных. Предлагается пересмотреть эту «традицию». В работе представлены оригинальный итеративный метод и алгоритм построения деревьев регрессии с ранней остановкой. Результаты численных экспериментов показывают, что предложенный алгоритм не уступает рекурсивным алгоритмам по среднеквадратичной адекватности идентификации, приводит к менее сложным моделям и обладает значительно меньшей трудоёмкостью.

Ключевые слова: машинное обучение, нелинейная регрессия, кусочно-заданная линейная регрессия, деревья моделей, деревья регрессии, упрощение деревьев регрессии.

1. Введение

Во многих задачах регрессионного анализа изучаемые объекты (явления, процессы и т.п.) имеют сложную неоднородную структуру и поэтому не могут быть адекватно описаны компактной простой моделью, построенной по всему диапазону значений данных. Одним из важных классов регрессионных моделей, призванных решить данную проблему, являются деревья регрессии. Они позволяют осуществить разделение входного пространства на сегменты с последующим построением для каждого из них собственной (локальной) модели и представить кусочно-заданную функцию регрессии в интуитивно понятной и наглядной форме. В таком дереве внутренние узлы содержат правила разделения пространства объясняющих переменных $\boldsymbol{X}$; дуги – условия перехода по ним; а листья – локальные регрессионные модели (рис. 1). Несмотря на то что возможность применения деревьев регрессии в анализе данных была успешно продемонстрирована ещё в 1984 году [1], алгоритмам данной группы было уделено сравнительно мало внимания.

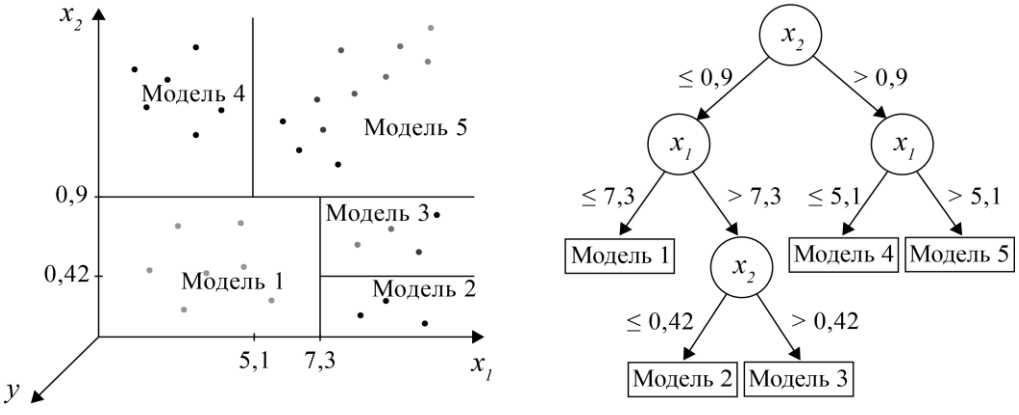

Рис. 1. Пример разбиения данных на сегменты и соответствующее ему дерево регрессии

Задача построения деревьев регрессии является NP-сложной [2]. При её 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?

Подавляющее большинство современных алгоритмов построения деревьев регрессии являются жадными. Такие алгоритмы осуществляют построение деревьев сверху вниз путем рекурсивного разделения обучающих данных. Целью данной работы является изложение сути нового итеративного (пошагового) метода построения деревьев регрессии. На каждой итерации согласно некоторому выбранному критерию в предлагаемом методе осуществляется расщепление одного листа строящегося дерева регрессии. Это обеспечивает более гибкий, по сравнению с рекурсивным подходом, контроль процесса построения дерева за счёт:

- определения произвольного порядка расщепления узлов;
- внедрения алгоритмов ранней остановки, анализирующих как отдельные узлы, так и всё дерево регрессии в целом;
- остановки процесса построения дерева регрессии в любой момент.

2. Предшествующие результаты

Как было отмечено ранее, большинство современных алгоритмов построения деревьев регрессии являются жадными и осуществляют построение деревьев сверху вниз путем рекурсивного разделения обучающих данных. Кратко их можно описать следующим образом:

1. Выбор «лучшего» разделения данных C, как правило, такого, которое обеспечивает экстремум некоторого показателя R:
   a. выбор объясняющей переменной;
   b. выбор точки разделения a или разделяющего множества A.
2. Разделение данных на подмножества.
3. Рекурсивное применение шагов 1–3 к каждому из выделенных подмножеств.

Одним из первых и наиболее известных алгоритмов, который можно отнести к рассматриваемому типу, является алгоритм CART [1]. Он делает разделения, минимизируя взвешенную сумму дисперсий целевой переменной после разделения данных (см. далее (1)), и использует константные локальные модели. Алгоритм M5 [3] стал следующим шагом в развитии алгоритмов построения деревьев регрессии. Он использует правило выбора разделений схожее с CART, но в листьях дерева строит линейные регрессионные модели. Правило выбора разделений в CART и M5 основано на дисперсии целевой переменной и никак не учитывает тип локальных моделей. В отличие от них алгоритм RETIS [4] выбирает разделение путем минимизации взвешенной суммы квадратов остатков (отклонений от действительных значений целевой переменной) локальных моделей. Однако вычислительная сложность такого правила очень высока. Самый большой набор данных, на котором был протестирован алгоритм, содержал лишь 300 наблюдений [4]. С тех пор многие [5–7] пытались уменьшить его вычислительную сложность.

Разделять входное пространство можно до тех пор, пока в узле останется лишь один обучающий пример. Вследствие чего при построении деревьев регрессии, как и для деревьев классификации, необходима процедура их упрощения. Здесь можно выделить два основных подхода: *отсечение ветвей* и *ранняя остановка*.

Отсечение ветвей предполагает построение дерева регрессии максимального размера, которое затем упрощается снизу вверх путем преобразования узлов в листья. В [7] отсечение ветвей происходит в два этапа. На первом этапе из полного (максимального размера) дерева

регрессии строится последовательность деревьев уменьшающегося размера. На втором этапе ошибка моделей оценивается на дополнительной независимой выборке, и модель с наименьшей ошибкой выбирается как итоговая. В [3] и [4] для каждого узла на обучающем множестве вычисляется ошибка локальной модели, но чтобы учесть сложность модели и количество обучающих примеров, по которым она была построена, ошибка умножается на поправочный коэффициент. Затем при движении снизу вверх узлы преобразовываются в листья до тех пор, пока скорректированная ошибка уменьшается.

Ранняя остановка предполагает ограничение роста дерева регрессии на этапе его построения путем прекращения разделения данных при достижении некоторого условия. Самым распространенным [3–9] правилом ранней остановки является ограничение на минимальное число примеров в узле. Практика показала, что данное правило не робастно [10]. И хотя его используют практически все алгоритмы, обычно оно используется совместно с другими правилами упрощения деревьев. В [6] при очередном расщеплении узла ошибка после разделения данных оценивается с помощью перекрестной проверки. Вычисленное значение сравнивается с ошибкой до разделения, если ошибка увеличилась, то разделение данных прекращается. В [5] с помощью теста Чоу проверяется статистическая гипотеза о том, что все данные в узле порождены некоторым скрытым линейным процессом. Разделение данных осуществляется, только если сумма квадратов остатков после разделения данных значительно меньше суммы квадратов остатков до разделения.

Рекурсивное разделение данных очень удобно с концептуальной и алгоритмической точек зрения: исходная задача декомпозируется, подмножества разделенных данных рассматриваются независимо друг от друга и от всего решения в целом. Это значительно упрощает процесс построения деревьев регрессии. Однако в этом случае возникает ряд проблем:

- Значимые улучшения при разделении данных на конкретном подмножестве могут быть ничтожными в рамках всей задачи. Поэтому использование алгоритмов ранней остановки для упрощения деревьев регрессии приводит к неоправданному росту дерева и, как следствие, к ухудшению интерпретируемости модели, а иногда и её адекватности.
- Алгоритмы упрощения деревьев решений на основе отсечения ветвей хорошо зарекомендовали себя. Некоторые из них были успешно применены в деревьях регрессии. Но здесь необходимо понимать, что трудоёмкость поиска разделений данных для деревьев регрессии значительно выше, чем для деревьев классификации, т.к. требуется построение огромного числа локальных моделей. И поэтому использование для упрощения деревьев регрессии алгоритмов, требующих построения дерева максимально возможного размера, ведет к значительным увеличениям временных затрат.

3. Предлагаемый метод

Чтобы преодолеть вышеописанные недостатки рекурсивного подхода, мы предлагаем осуществлять построение дерева пошагово путём постепенного итеративного расщепления (разделения данных) его листьев. Кратко предлагаемый метод можно представить следующим образом:

1. Начать с дерева регрессии, состоящего только из корня (он же единственный листовой узел).
2. Для каждого нерассмотренного листового узла найти «лучшее» расщепление C, как правило, такое, которое обеспечивает экстремум некоторого показателя R:
   a. выбрать объясняющую переменную;
   b. выбрать точку разделения a или разделяющее множество A.

3. Из возможных на текущей итерации расщеплений выбрать «лучшее» в соответствии с R.
4. Если выбранное расщепление действительно (в соответствии с некоторым критерием Q) улучшает модель, то выполнить его и перейти к шагу 2; в противном случае завершить выполнение алгоритма.

Отказ от рекурсии и переход к итерационной версии позволяют более гибко контролировать процесс построения дерева. Во-первых, на любой итерации мы имеем доступ к текущему состоянию строящегося дерева регрессии и можем проанализировать вклад каждого расщепления в общую модель ещё в процессе построения дерева регрессии. Во-вторых, мы можем расщеплять листья на любой глубине и в произвольном порядке. Так, например, сначала можно осуществлять расщепление, максимально улучшающее нашу модель на всех данных. Всё это даёт возможность строить дерево регрессии пошагово сверху вниз таким образом, что дерево регрессии на текущей итерации будет лучшим среди всех возможных деревьев регрессии такого же размера, которые мог бы построить жадный рекурсивный алгоритм. И, как следствие, в-третьих, становится возможным завершить процесс построения дерева регрессии в любой момент и легко внедрить алгоритмы упрощения на основе ранней остановки, анализирующие всё дерево регрессии целиком. Это способствует предотвращению чрезмерного роста дерева и уменьшению времени его построения.

Метод не определяет, какие критерии (показатели) должны использоваться в качестве R и Q. Однако, как можно заметить, поиск разделения данных в конкретном узле ничем не отличается от процесса поиска в рекурсивном подходе. Поэтому в качестве R может быть использован любой из ранее предложенных [1, 3–7] для построения деревьев регрессии показателей. Критерии для ранней остановки (Q), анализирующие всю модель целиком, в теории построения деревьев регрессии не разрабатывались. В качестве перспективных направлений здесь стоит выделить:

- вычисление обобщенной ошибки модели по ошибке на проверочном множестве;
- статистические тесты;
- информационные критерии выбора моделей.

Для демонстрации возможностей предложенного метода определим R и Q, после чего сравним полученные результаты его работы с рекурсивными методами.

В качестве критерия выбора «лучшего» расщепления листовых узлов будем использовать минимум статистики $R(T,C)$, равной взвешенной сумме квадратов ошибок локальных моделей:

$$R(T,C) = \frac{N_L}{N}\sum_{i\in I_L}(y_i - g_L(x_i))^2 + \frac{N_R}{N}\sum_{i\in I_R}(y_i - g_R(x_i))^2, \quad (1)$$

где T – исходный набор данных; T_L и T_R – наборы, образованные путем разделения T по C; N, N_L и N_R – количество элементов в каждом из наборов; I_L и I_R – множества индексов 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 трудоёмким. Поэтому для каждой переменной будем проверять лишь 20 разделений, границы которых равномерно распределены на области её значений.

Для критерия ранней остановки Q адаптируем расширенный байесовский информационный критерий из [11]. В нём осуществляется минимизация статистики:

$$EBIC = n\cdot\ln(\frac{SSE}{n}) + J\cdot(\ln(n) + 2\ln(p)), \quad (2)$$

где SSE – сумма квадратов остатков модели на обучающих данных; J – количество настраиваемых параметров модели; n – количество примеров в обучающей выборке; p – величина, характеризующая сложность пространства моделей (в нашем случае она равна произведению размера дерева на количество объясняющих переменных). В выражении (2) первое слагаемое – это максимальное значение логарифмической функции правдоподобия модели, а второе слагаемое представляет собой штраф за сложность модели. Расщепление узлов продолжается до тех пор, пока величина (2) уменьшается. Отметим, что использование данного критерия в рекурсивном подходе невозможно, т.к. в рекурсивных методах на этапе построения дерева рассматривается лишь часть модели безотносительно ко всей модели в целом.

4. Сравнение предлагаемого метода с рекурсивными алгоритмами построения деревьев регрессии

Алгоритм на основе предложенного метода был протестирован на 2 синтетических наборах данных из [9] и 8 наборах данных из UC Irvine Machine Learning Repository [12] и KEEL-Dataset Repository [13]. Их краткая характеристика представлена в табл. 1.

Таблица 1. Краткая характеристика рассматриваемых наборов данных

Название	Объём выборки	Количество объясняющих переменных
Armchair	1000	2
Split plane	1000	1
Abalone	4177	8
Ailerons	13750	40
Auto-mpg	392	7
CPU	209	6
Housing	506	14
Stock	950	10
Breast Cancer Wisconsin	198	34
Triazines	186	61

Результаты работы алгоритма приведены в табл. 2. Они получены с помощью 10-слойной перекрёстной проверки, усреднены по 30 запускам и приведены в формате: среднее значение показателя по всем запускам ± одно среднеквадратическое отклонение. В качестве показателя адекватности полученных моделей было использовано нормализованное по среднеквадратическому отклонению значение квадратного корня из среднего квадрата ошибки (nRMSE). Для оценки сложности моделей использовалось количество вершин в дереве регрессии (количество внутренних узлов и листьев). В таблице также приведено время построения (в секундах) дерева регрессии.

Выполнено сравнение предложенного алгоритма построения деревьев регрессии с несколькими алгоритмами на основе рекурсивного разделения данных. Чтобы поставить их в равные условия, в каждом случае использовалось одно и то же правило выбора «лучшего» разделения данных на основе минимизации взвешенной суммы квадратов остатков локаль-

ных моделей (1) и один и тот же алгоритм построения линейных локальных моделей [14]. Однако в случае рекурсивного разделения данных варьировались алгоритмы упрощения деревьев регрессии. Были рассмотрены алгоритм отсечения ветвей по оценке «цена – сложность» и два алгоритма ранней остановки – остановка по ошибке на валидационном множестве и по тесту Чоу.

Таблица 2. Эмпирическое сравнение предложенного итерационного алгоритма построения деревьев регрессии с алгоритмами на основе рекурсивного разделения данных

Набор данных	Показатели	Предлагаемый метод	«Цена – сложность»	Валидационное множество	Тест Чоу
ArmChar	nRMSE	0.25 ± 0.02	0.1 ± 0.02	0.11 ± 0.02	**0.09 ± 0.01**
	Сложность	17.2 ± 0.9	16.5 ± 0.9	17.5 ± 0.8	**15.9 ± 0.6**
	Время	**0.24 ± 0.01**	0.44 ± 0.06	0.25 ± 0.01	0.27 ± 0.02
Split-Plane	nRMSE	**0.01 ± 0.01**	**0.01 ± 0.01**	**0.01 ± 0.01**	**0.01 ± 0.01**
	Сложность	4.8 ± 0.2	4.49 ± 0.51	4.4 ± 0.59	**4.27 ± 0.56**
	Время	**0.04 ± 0.01**	0.06 ± 0.01	0.05 ± 0.01	0.05 ± 0.01
Abalone	nRMSE	**0.67 ± 0.0**	0.69 ± 0.01	**0.67 ± 0.0**	**0.67 ± 0.02**
	Сложность	**4.1 ± 0.2**	6.5 ± 2.8	7.4 ± 1.3	19.4 ± 0.7
	Время	**3.0 ± 0.3**	10.4 ± 0.2	5.3 ± 0.5	9.0 ± 0.3
Ailerons	nRMSE	**0.4 ± 0.0**	**0.4 ± 0.0**	**0.4 ± 0.0**	**0.4 ± 0.0**
	Сложность	**6.9 ± 0.1**	10.0 ± 2.5	14.7 ± 1.3	18.4 ± 0.9
	Время	**59.5 ± 6.5**	129.3 ± 6.9	90.3 ± 4.7	124.5 ± 12.8
Autompg	nRMSE	**0.39 ± 0.01**	0.43 ± 0.01	0.40 ± 0.01	0.41 ± 0.02
	Сложность	**3.0 ± 0.0**	4.4 ± 2.8	5.9 ± 1.1	19.5 ± 1.1
	Время	**1.2 ± 0.1**	5.1 ± 0.3	1.93 ± 0.21	4.0 ± 0.4
Housing	nRMSE	**0.51 ± 0.1**	0.55 ± 0.19	0.52 ± 0.12	0.54 ± 0.18
	Сложность	**4.8 ± 0.2**	5.87 ± 2.2	7.3 ± 1.2	20.3 ± 0.8
	Время	**9.3 ± 0.6**	30.3 ± 0.9	13.6 ± 1.6	28.5 ± 0.8
Machine	nRMSE	**0.28 ± 0.02**	0.38 ± 0.06	0.35 ± 0.05	0.35 ± 0.04
	Сложность	3.1 ± 0.1	**2.65 ± 0.8**	4.0 ± 0.9	16.7 ± 1.1
	Время	**0.8 ± 0.1**	3.6 ± 0.1	1.1 ± 0.1	2.9 ± 0.1
Stock	nRMSE	0.15 ± 0.01	0.16 ± 0.01	0.16 ± 0.01	**0.13 ± 0.01**
	Сложность	**9.5 ± 0.5**	16.9 ± 2.3	19.9 ± 1.4	29.8 ± 0.6
	Время	**7.5 ± 0.4**	17.1 ± 0.2	12.7 ± 0.4	15.8 ± 0.4
Breast	nRMSE	**0.88 ± 0.01**	0.89 ± 0.02	0.9 ± 0.04	**0.88 ± 0.01**
	Сложность	**1.0 ± 0.0**	**1.0 ± 0.0**	1.1 ± 0.1	**1.0 ± 0.0**
	Время	**14.8 ± 1.1**	370.4 ± 47.7	**23.3 ± 2.1**	37.0 ± 3.5
Triazine	nRMSE	**0.89 ± 0.0**	1.01 ± 0.19	1.01 ± 0.19	**0.89 ± 0.0**
	Сложность	**1.0 ± 0.0**	2.0 ± 1.0	3.2 ± 0.7	**1.0 ± 0.0**
	Время	**4.6 ± 0.2**	24.3 ± 0.9	8.9 ± 1.3	4.9 ± 0.2

Для наглядности на рис. 2 приведены усредненные по всем наборам данных значения показателя nRMSE, сложности полученных моделей, а также времени их построения.

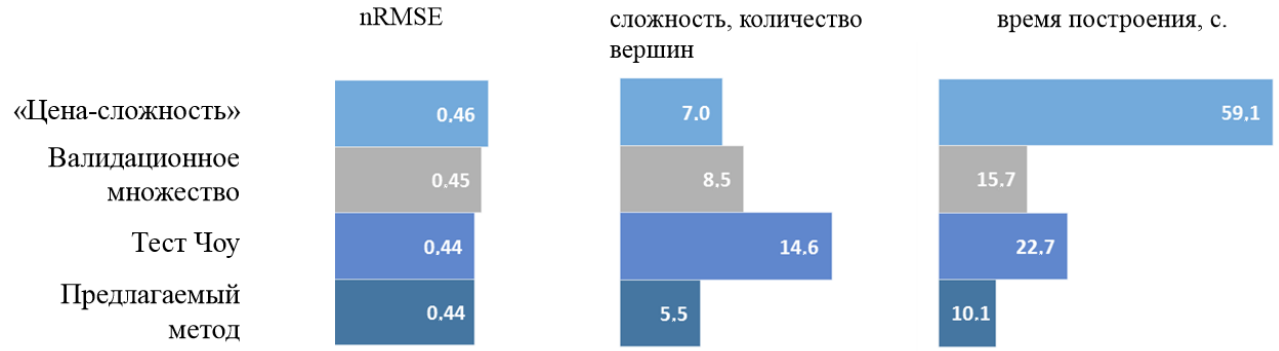

Рис. 2. Усредненные по всем наборам данных значения исследуемых показателей

Анализируя табл. 2 и рис. 2, можно заметить, что ошибка аппроксимации предложенного алгоритма сравнима с рассмотренными альтернативами. При этом наблюдается малозначительное улучшение значений показателей алгоритма предложенного метода по сравнению с двумя из трёх рассмотренных алгоритмов рекурсивного метода.

Предложенный алгоритм позволяет строить самые простые модели. Они в среднем практически в 3 раза проще, чем модели, полученные с помощью рекурсивного алгоритма с ранней остановкой по тесту Чоу; в среднем имеют размер на 3 узла меньше, чем у алгоритма ранней остановки по ошибке на валидационном множестве, и на 1.5 узла, чем у алгоритма отсечения ветвей на основе оценки «цена – сложность». Если сравнивать время построения деревьев регрессии, то здесь самым быстрым также оказывается предложенный алгоритм. Он в среднем в 6 раз быстрее рекурсивного алгоритма с отсечением ветвей на основе оценки «цена – сложность», в 1.5 раза быстрее ранней остановки по ошибке на валидационном множестве и в 2 раза быстрее ранней остановки по тесту Чоу.

5. Заключение

Предложены итеративный шаговый метод и алгоритм построения деревьев регрессии, обеспечивающие более гибкий, по сравнению с рекурсивными алгоритмами, контроль процесса построения дерева. Это позволяет использовать и разрабатывать более эффективные методы упрощения деревьев регрессии, анализирующие на каждом шаге все дерево регрессии, а не только его отдельные части.

Разработанный на основе предложенного метода алгоритм построения деревьев регрессии с использованием для упрощения расширенного байесовского информационного критерия показывает перспективность работ в данном направлении. Он не уступает по точностным показателям алгоритмам на основе рекурсивного разделения данных и при этом строит более компактные модели. К тому же он значительно менее трудоёмок: приблизительно в 3 раза быстрее рекурсивных алгоритмов с упрощением на основе отсечения ветвей и в 1.5 – 2 раза быстрее рекурсивных алгоритмов с ранней остановкой.

Литература

1. Classification and regression trees / L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone. Belmont: Wadsworth International Group, 1984. 259 p.
2. *Hyafil L., Rivest R. L.* Constructing optimal binary decision trees is NP-complete // Information Processing Letters. 1976. V. 5, № 1. P. 15–17.
3. *Quinlan J. R.* Learning with continuous classes // Proc. AI'92, 5th Australian Joint Conference on Artificial Intelligence. 1992. P. 343–348.
4. *Karalic A.* Employing linear regression in regression tree leaves // Proceedings of the 10th European Conference on Artificial Intelligence. 1992. P. 440–441.
5. *Potts D., Sammut C.* Incremental learning of linear model trees // Machine Learning. 2005. V. 61. P. 5–48.
6. *Vogel D., Asparouhov O., Scheffer T.* Scalable look-ahead linear regression trees // In: Proc. of 13th ACM SIGKDD. 2007. P. 757–764.
7. Loh W.-Y. Regression trees with unbiased variable selection and interaction detection // Statistica Sinica. 2002. V. 12. P. 361–386.
8. *Torgo L. A* Comparative study of reliable error estimators for pruning regression trees // In Proceeding of the Iberoamerican Conference on Artificial Intelligence. 1998. P. 1–12.
9. Мельников Г. А. Применение методов искусственного интеллекта для исследования инфекционных заболеваний: магистерская дис. ... «Магистр техники и технологии»: 230100. г. Новосибирск, 2012. 141 с.
10. *Murthy Sreerama K.* Automatic construction of decision trees from data: A multi-disciplinary survey // Data Mining and Knowledge Discovery. 2005. V. 2, № 4. P. 345–389.
11. *Chen J., Chen Z.* Extended Bayesian information criteria for model selection with large model spaces // Biometrika. 2008. V. 95, № 3. P. 759–771.
12. *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.
13. KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework / J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera // Journal of Multiple-Valued Logic and Soft Computing. 2011. V. 17. P. 255–287.
14. *Gramacy R. B.* Model choice and data mining [Электронный ресурс]. URL: `http://faculty.chicagobooth.edu/robert.gramacy/teaching/ara/lect7.pdf` (дата обращения: 14.09.2014).

Статья поступила в редакцию 26.02.2016

Мельников Григорий Андреевич

аспирант кафедры вычислительной техники Новосибирского государственного технического университета (630087, Новосибирск, ул. Немировича-Данченко, 136), тел. +7-961-225-22-96, e-mail: `grmel89@gmail.com`.

Губарев Василий Васильевич

д.т.н., профессор кафедры вычислительной техники Новосибирского государственного технического университета (630087, Новосибирск, ул. Немировича-Данченко, 136), тел. (383) 346-11-33, e-mail: `gubarev@vt.cs.nstu.ru`.

The iterative method of regression trees induction

G.A. Melnikov, V.V. Gubarev

The majority of modern algorithms for regression tree induction are greedy. They are all based on a recursive division of the data. We propose to revise this «tradition». The paper describes the novel iterative method and algorithm with pre-pruning for regression tree induction. The results of the experiments based on publicly available data sets show that the proposed algorithm is comparable with recursive algorithms for regression tree induction in accuracy. However, it results in less complex solutions and has a much lower time complexity.

Keywords: machine learning, nonlinear regression, piecewise linear regression, models trees, regression trees, regression trees pruning.