УДК 004.85

Г.А. Мельников, В.В. Губарев

Метод построения деревьев регрессии на основе муравьиных алгоритмов

Описан новый метод построения деревьев регрессии на основе моделирования поведения колонии муравьев при поиске пищи, совмещающий идеи традиционных алгоритмов построения деревьев регрессии и муравьиных алгоритмов. Результаты численных экспериментов показывают, что разработанный метод превосходит традиционные алгоритмы данной группы по среднеквадратичной адекватности идентификации и приводит к менее сложным моделям.

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

Постановка задачи. Деревья регрессии являются одним из важных классов регрессионных моделей, позволяющих 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 регрессионном анализе данных была успешно продемонстрирована ещё в [1] (1984 г.), алгоритмам данной группы было уделено сравнительно мало внимания.

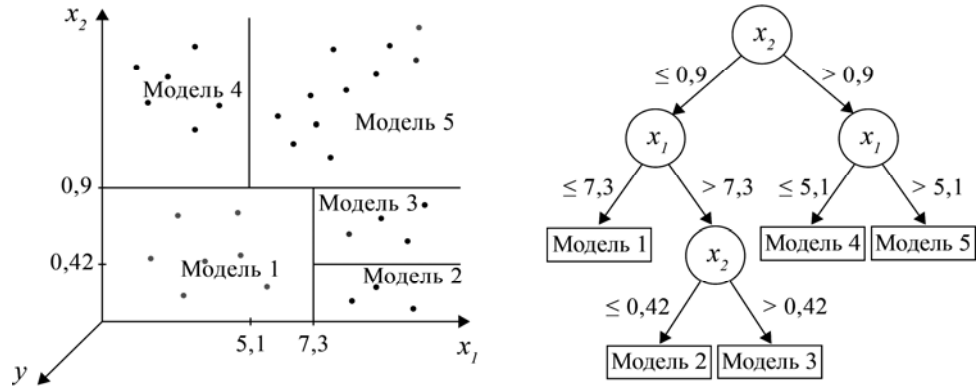

Рис. 1. Пример разбиения данных на сегменты и соответствующего ему дерева регрессии

Большинство современных алгоритмов построения деревьев регрессии являются жадными. Такие алгоритмы осуществляют построение деревьев сверху вниз путем рекурсивного разделения обучающих данных и кратко могут быть описаны следующим образом:

1. выбрать лучшее разделение (обычно выбор разделения, которое обеспечивает экстремум некоторого критерия);
2. разделить исходные данные на подмножества;
3. рекурсивно применить данную процедуру для каждого из выделенных подмножеств.

Жадные алгоритмы обладают низкой трудоемкостью, хорошо масштабируемы, но имеют ряд недостатков: *а*) дерево регрессии строится постепенно без возврата к ранее принятым решениям; *б*) на каждом шаге работы алгоритма принимается локально оптимальное решение, т.е. решение, дающее максимальный эффект на текущем шаге, без учета его влияния на всё решение в целом. Поэтому они приводят, как правило, к неоптимальному разделению данных.

Целью данной работы является дополнение сути и более подробное описание нового стохастического метода построения деревьев регрессии на основе муравьиных алгоритмов, предложенного и кратко изложенного авторами в [2]. На каждой итерации разработанный метод строит не одно, а множество деревьев регрессии, действуя методом проб и ошибок. Накапливая информацию о принятых решениях и результатах, к которым они привели, он использует её для построения в дальнейшем более качественных моделей.

Предыдущие работы. Как было отмечено выше, большинство современных алгоритмов построения деревьев регрессии являются жадными алгоритмами. Отличаются они главным образом правилом выбора лучшего разделения данных. Ряд альтернативных правил выбора разделений был предложен для построения деревьев регрессии.

Одним из первых и наиболее известных алгоритмов, который можно отнести к рассматриваемой теме, является алгоритм CART [1]. CART выбирает разделения, минимизируя взвешенную сумму дисперсий целевой переменной после разделения данных [см. далее (2)], и использует константные локальные модели. Алгоритм M5 [3] стал следующим шагом в развитии алгоритмов построения деревьев регрессии. Этот алгоритм использует правило выбора разделений, схожее с CART, но в листьях дерева строит линейные регрессионные модели.

Правило выбора разделений в M5 основано на дисперсии целевой переменной и никак не учитывает тип локальных моделей. Поэтому 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.

Альтернативу работам [4–7] составили алгоритмы GUIDE [8] и SECRET [9]. Вопросам построения деревьев классификации уделено значительно больше внимания, чем вопросам построения деревьев регрессии. Поэтому они преобразуют исходную задачу регрессии в задачу классификации и затем используют методы, применяемые для построения деревьев классификации.

На сегодняшний день есть лишь два алгоритма, в которых авторы попытались выйти за рамки жадных алгоритмов. Это M5Opt [10] и GMT [11]. M5Opt представляет собой частично жадный алгоритм построения деревьев регрессии, в котором верхние уровни дерева строятся полным перебором, а остальная часть дерева строится с помощью быстрых жадных алгоритмов. Это обеспечивает баланс между исследованием пространства поиска и временем выполнения алгоритма. Так, в [10] в качестве жадного алгоритма был использован алгоритм M5. Алгоритм GMT относится к алгоритмам генетического программирования и наследует основные черты алгоритмов данной группы. Алгоритм работает с совокупностью деревьев регрессии (популяцией особей) и для улучшения качества моделей использует аналоги механизмов генетического наследования, генетической изменчивости и естественного отбора. Все операции выполняются непосредственно над деревьями, например, в качестве оператора скрещивания может служить обмен поддеревьями, начиная с выбранных узлов.

Предлагаемый метод. В основе предлагаемого метода лежит идея моделирования непрямого обмена информацией через наблюдение в окружающей среде особого вещества, оставляемого муравьями на своём пути при возвращении в муравейник, – феромона [12]. При поиске пищи муравьи, почувствовав такие следы, инстинктивно устремляются к ним. А так как со временем феромон испаряется, то на более коротких путях к источнику пищи его концентрация окажется выше, и муравьи будут предпочитать эти пути другим. В случае построения деревьев регрессии виртуальный феромон позволяет выделить наиболее удачные комбинации вариантов разделения данных и как можно чаще использовать их в дальнейшем.

Кратко предлагаемый метод можно представить следующим образом:

1. Инициализация.
2. Для каждого муравья выполнить шаги 2.1–2.2.
2.1. Выбрать следующее разделение данных на основе вероятностного правила.
2.2. Если решение построено не полностью, то перейти к шагу 2.1.
3. Обновить архив решений.
4. Обновить распределение феромона.
5. Если условия остановки не выполнены, то перейти к шагу 2.

Для пошагового построения решений (деревьев регрессии) пространство поиска представляется в виде дерева прототипа (рис. 2). Узлы такого дерева содержат множество меток возможных значений узла в дереве регрессии – метки переменных, по которым осуществляется разделение x_1, …, x_n, и метку локальной модели *leaf*. Меткам соответствуют определенные значения концентрации феромона τ и эвристической функции оценки разделения данных η, в соответствии с которыми муравей при обходе дерева (например, в глубину или ширину) осуществляет выбор метки в каждом посещенном узле.

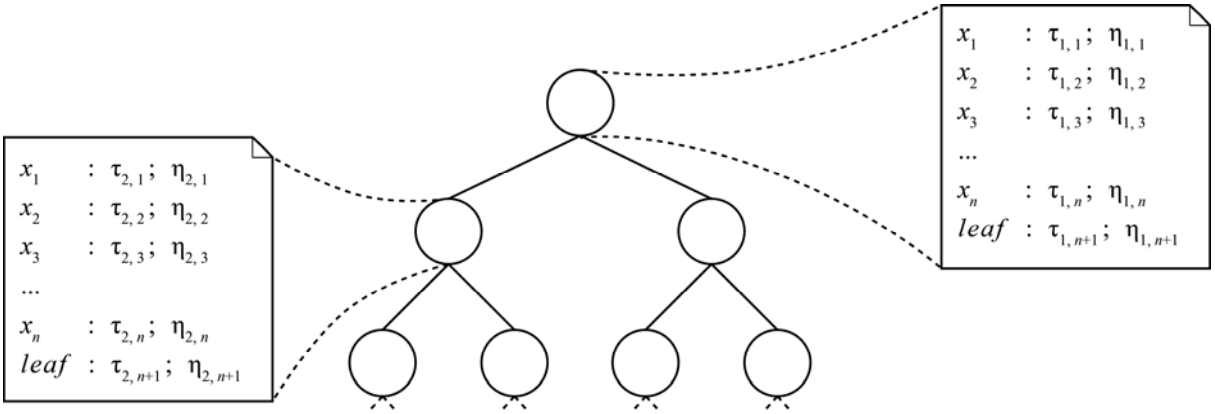

Рис. 2. Представление пространства поиска решений в виде дерева прототипа

Вероятность выбора метки j в узле i определяется как [13]

$$p_{i,j} = \begin{cases} \dfrac{[\tau_{i,j}]^{\alpha}[\eta_{i,j}]^{\beta}}{\sum\limits_{k \in A}[\tau_{i,k}]^{\alpha}[\eta_{i,k}]^{\beta}} & \text{если } j \in A, \\ 0 & \text{в противном случае,} \end{cases} \quad (1)$$

где α и β – регулируемые параметры, численные значения которых определяют важность феромона и эвристики при выборе метки; A – множество допустимых для посещения меток.

Значение эвристической функции для метки *leaf* обратно пропорционально стандартному отклонению целевой переменной, для меток переменных – взвешенной сумме стандартных отклонений целевой переменной после разделения данных:

$$\eta_{i,j} = \left[\left(\frac{N_{x_j \le x^*}}{N} \mathrm{std}(T_i^{x_j \le x^*}) + \frac{N_{x_j > x^*}}{N} \mathrm{std}(T_i^{x_j > x^*}) \right) + 1 \right]^{-1}, \quad (2)$$

где $T_i^{x_j \le x^*}$, $T_i^{x_j > x^*}$ – множества, полученные путем разбиения данных в узле i по переменной x_j в точке x^* по правилу $x_j \le x^*$; N, $N_{x_j \le x^*}$, $N_{x_j > x^*}$ – количество элементов в каждом из множеств соответственно; std(·) – стандартное отклонение значений целевой переменной. Точка разделения данных x^* определяется как решение задачи максимизации выражения (2). Эвристическая функция позволяет оценить качество выбора лишь на текущем шаге, она не учитывает дальнейшие шаги алгоритма.

При выборе метки j-й переменной в соответствующем узле дерева регрессии осуществляется разделение данных по x_j, а при выборе метки *leaf* – построение локальной модели. В качестве локальных моделей будем использовать множественную линейную регрессию. Ее построение выполняется с помощью алгоритма пошаговой регрессии [14].

После построения решений всеми муравьями они оцениваются с точки зрения выбранных критериев (например, минимизации корня из среднего квадрата ошибки дерева регрессии и минимизации его размера), и все недоминируемые решения поочередно заносятся в архив решений Q. Если архив решений переполнен, то осуществляется попытка заменить одно из решений архива. Замещаемое решение ищется только среди решений (D), доминируемых новым решением. Если таковые отсутствуют, то размер архива увеличивается на единицу и решение добавляется без замещения. Каждому решению из D присваивается ранг, равный количеству доминируемых над ним решений. Замещается решение с максимальным рангом. Если их несколько, то выбирается решение, наиболее похожее на новое. Такая стратегия удаляет слабые решения и одновременно поддерживает разнообразие решений в архиве.

В конце каждой итерации происходит обновление распределения феромона на основе выборки $P \subseteq Q$, состоящей из k решений, наиболее близких (в пространстве критериев оценки качества) к решению π, выбранному из Q случайно [15]:

$$\tau_{i,j} = \tau_{\min} + \Delta \cdot \left| \{ \pi \mid \pi \in P \text{ и } (i,j) \in \pi \} \right|, \quad (3)$$

где Δ определяется как $(\tau_{max} - \tau_{min})/k$; τ_{min} и τ_{max} – регулируемые параметры, представляющие собой минимальное и максимальное значение концентрации феромона. Распределение феромона в отличие от эвристической функции учитывает взаимодействие различных атрибутов, в том числе ещё не выбранных, и отражает приобретенный колонией муравьев опыт. Удаление решения из Q можно рассматривать как процесс испарения, а добавление – как увеличение концентрации феромона на соответствующих участках пути.

В качестве критерия остановки можно использовать достижение максимального числа итераций или достижение заданного числа итераций без изменения архива решений. Итоговое дерево моделей выбирается из недоминируемых решений архива, исходя из ошибки на независимой проверочной (валидационной) выборке данных.

Сравнение предложенного метода с аналогами. Предложенный метод был протестирован на 6 наборах данных из UC Irvine Machine Learning Repository [16] и KEEL-dataset repository [17]:

- Abalone – задача прогнозирования возраста морских ракушек в зависимости от физических характеристик. Количество примеров – 4177; количество переменных – 8.
- Ailerons – задача управления самолетом. Атрибуты описывают состояние самолета, в то время как целью является прогнозирование управляющего воздействия на элероны. Количество примеров – 13750; количество переменных – 40.
- Auto-mpg – задача прогнозирования расхода топлива (мили на галлон, mpg) в зависимости от характеристик автомобиля. Количество примеров – 392; количество переменных – 7.
- CPU – задача прогнозирования производительности процессоров. Количество примеров – 209; количество переменных – 6.
- Housing – задача прогнозирования средней цены на дома в Бостоне. Количество примеров – 506; количество переменных – 14.
- Stock – данные представляют ежедневные цены на акции с января 1988 по октябрь 1991 года десяти компаний аэрокосмической отрасли, требуется прогнозировать цену на акции десятой компании по остальным девяти. Количество примеров – 950; количество переменных – 10.

Для предложенного метода были выбраны следующие настройки: размер популяции – 50, количество итераций – 25, параметры $\alpha = 1$ и $\beta = 3$, размер архива решений Q – 50, размер подвыборки из Q для обновления феромона $k = 10$, минимальное и максимальное количество феромона $\tau_{min} = 0{,}01$, $\tau_{max} = 2$ и два критериальных показателя – корень из среднего квадрата ошибки (RMSE) и размер деревьев регрессии (количество всех узлов дерева). Обучающая выборка данных разбивалась на две подвыборки: 70% непосредственно использовалось для построения деревьев регрессии, 30% – для выбора итогового дерева из архива решений.

Результаты работы предложенного метода (далее AntMT) приведены в таблице. Было также выполнено его сравнение с классическим жадным алгоритмом построения деревьев регрессии M5, эволюционным алгоритмом GMT и собственной реализацией алгоритма RETIS, в которой 30% обучающих данных используются в качестве валидационной выборки для ранней остановки. Все результаты получены с помощью 10-слойной перекрестной проверки и усреднены по 20 запускам, в таблице они приведены в формате: среднее значение показателя по всем запускам ± одно среднеквадратическое отклонение[*].

Практически на всех рассматриваемых наборах данных предлагаемый метод имеет адекватность идентификации (по критериальному показателю RMSE) лучше, чем M5, RETIS или GMT. При этом улучшение составляет от 1 до 12% по отношению к лучшей из альтернатив. Лишь на наборе данных Stock адекватность моделей AntMT и RETIS оказалась сопоставимой. Результаты сравнения сложности моделей не так однозначны. Деревья регрессии, построенные методом AntMT, обычно имеют меньшие размеры, чем построенные с помощью M5 или RETIS, но большие, чем построенные алгоритмом GMT. Однако если сравнить деревья, построенные AntMT и GMT, имеющие одинаковый размер (таблица, колонка $\text{AntMT}_{\text{GMT}}$), то можно увидеть, что адекватность идентификации метода AntMT всё ещё лучше GMT.

[*] Результаты работы алгоритма GMT взяты из публикации [11], в которой отсутствуют сведения о среднеквадратическом отклонении исследуемых показателей производительности.

Сравнение алгоритмов построения деревьев регрессии

		AntMT	$AntMT_{GMT}$	M5	RETIS	GMT
Abalone	RMSE	**2,14 ± 0,01**	2,16 ± 0,01	2,24 ± 0,01	2,16 ± 0,01	2,24
	Размер	10,8 ± 1,9	**6,7 ± 0,1**	34,9 ± 1,3	18,4 ± 0,1	**6,7**
Ailerons	RMSE	**0,000164 ± 0,0**	0,000165 ± 0,0	0,000192 ± 0,000001	0,000186 ± 0,0	0,000200
	Размер	**17,2 ± 1,1**	24 ± 0,0	126,1 ± 9,2	38,6 ± 0,1	24
Auto-mpg	RMSE	3,06 ± 0,09	**3,02 ± 0,08**	3,34 ± 0,11	3,18 ± 0,1	3,23
	Размер	11,4 ± 2,9	**4,7 ± 0,3**	11,0 ± 2,5	12,8 ± 3,2	**4,7**
CPU	RMSE	**52,14 ± 8,36**	53,11 ± 10,98	74,2 ± 6,1	57,08 ± 9,53	63,4
	Размер	**5,5 ± 1,3**	6,1 ± **0,2**	7,6 ± 1,0	7,1 ± 1,3	6,1
Housing	RMSE	3,91 ± 0,28	**3,77 ± 0,19**	4,35 ± 0,19	4,31 ± 0,42	4,21
	Размер	12,1 ± 0,9	**6,6 ± 0,2**	23,8 ± 1,1	10,2 ± 1,1	**6,6**
Stock	RMSE	**1,02 ± 0,07**	1,04 ± 0,08	1,08 ± 0,05	1,03 ± 0,08	1,22
	Размер	22,4 ± 1,5	**18 ± 0,0**	65,2 ± 2,2	32,2 ± 0,6	**18**

Интересно, что с уменьшением размера деревьев регрессии, полученных с помощью AntMT, их адекватность улучшилась на двух наборах данных – Auto-mpg и Housing. Можно сказать, что упрощение деревьев регрессии на основе информационных критериев выбора моделей, которое использует GMT, в некоторых случаях может вести себя лучше упрощения на основе валидационного множества. Вероятно, подход, используемый в алгоритме GMT, может быть эффективно использован и в AntMT.

Одним из недостатков предлагаемого метода является большое число настраиваемых параметров. Однако на всех протестированных наборах данных AntMT показал хорошие результаты. Заметим, в исследовании не делалось никакой настройки параметров метода под исходные данные. Их влияние на производительность будет исследовано в последующих работах.

В конце работы в качестве примера приведем часть дерева регрессии, построенного предложенным методом, для набора данных Auto-mpg (рис. 3). Отметим здесь лишь одну интересную деталь. Помимо очевидного разделения автомобилей по количеству цилиндров (cylinders), в модели присутствует разделение по году выпуска («model-year ≤ 77»). Вероятно, это связано с тем, что в 1975 г. в США был принят новый стандарт по экономии расхода топлива, который начал действовать на новые модели автомобилей начиная с 1978 г. [18].

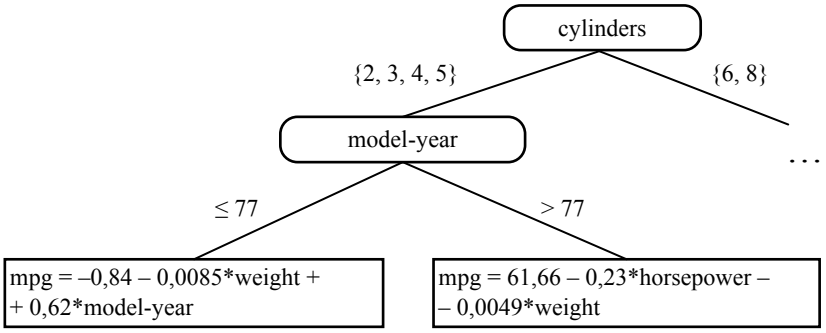

Рис. 3. Пример части дерева регрессии для набора данных Auto-mpg

Заключение. В работе представлен новый метод построения деревьев регрессии на основе моделирования поведения колонии муравьев при поиске пищи. Эксперименты показывают, что алгоритм по предложенному методу превосходит традиционные алгоритмы по адекватности моделей, а также позволяет получать более простые (компактные) деревья регрессии.

В качестве дальнейших направлений исследований выделим следующие:

- разработка и исследование отдельных компонент муравьиных алгоритмов;
- исследование применимости существующих и разработка новых масштабируемых алгоритмов поиска разделений данных;
- разработка и исследование алгоритмов самонастройки свободных параметров метода;
- разработка и исследование новых подходов упрощения деревьев регрессии, ориентированных на применение в стохастических итеративных методах построения деревьев регрессии.

Литература

1. Breiman L. Classification and Regression Trees / L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone. – Belmont: Wadsworth International Group, 1984. – 259 p.

2. Melnikov G.A. Ant Colony Based Semi-Greedy Algorithm for Regression Tree Induction / G.A. Melnikov, V.V. Gubarev // Proceedings 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

3. Quinlan J.R. Learning with continuous classes / J.R. Quinlan // Proc. AI'92, 5th Australian Joint Conference on Artificial Intelligence. – Singapore: World Scientific, 1992. – P. 343–348.

4. Karalic A. Employing linear regression in regression tree leaves / A. Karalic // Technical Report IJS DP-6450. – Ljubljana, Slovenia: Jozef Stefan Institute, 1992. – 11 p.

5. Alexander W.P. Treed regression / W.P. Alexander, S.D. Grimshaw // Journal of Computational and Graphical Statistics. – 1996. – Vol. 5. – P. 156–175.

6. Top-down induction of model trees with regression and splitting nodes / D. Malerba, F. Esposito, M. Ceci, A. Appice // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2004. – Vol. 26. – P. 612–625.

7. Vogel D. Scalable look-ahead linear regression trees / D. Vogel, O. Asparouhov, T. Scheffer // Proc. of 13th ACM SIGKDD. – New York: ACM Press, 2007. – P. 757–764.

8. Loh W.-Y. Regression trees with unbiased variable selection and interaction detection // Statistica Sinica. – 2002. – Vol. 12. – P. 361–386.

9. Dobra A. SECRET: A scalable linear regression tree algorithm / A. Dobra, J. Gehrke // Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. – New York: ACM Press, 2002. – P. 481–487.

10. Solomatine D.P. Semi-optimal Hierarchical Regression Models and ANNs / D.P. Solomatine, L.A. Siek // Proc. Intern. Joint Conference on Neural Networks, Budapest. – New York: IEEE, 2004. – P. 1173–1177.

11. Czajkowski M. An Evolutionary Algorithm for Global Induction of Regression and Model Trees / M. Czajkowski, M. Kretowski // International Journal of Data Mining, Modelling and Management [Accepted for publication].

12. Bonabeau E. Swarm Intelligence: From Natural to Artificial Systems / E. Bonabeau, M. Dorigo, G. Theraulaz. – New York: Oxford University Press, 1999. – 307 p.

13. Dorigo M. The Ant System: Optimization by a Colony of Cooperating Agents / M. Dorigo, V. Maniezzo, A. Colorni // IEEE Transactions on Systems, Man, and Cybernetics, Part B. – 1996. – Vol. 26, № 1. – P. 29–41.

14. Gramacy R.B. Model Choice and Data Mining [Электронный ресурс]. – Режим доступа: http://faculty.chicagobooth.edu/robert.gramacy/teaching/ara/lect7.pdf, свободный (дата обращения: 14.09.2014).

15. Guntsch M. Solving Multi-criteria Optimization Problems with Population-Based ACO / M. Guntsch, M. Middendorf // Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization, Faro, Portugal, April 8–11, 2003. – Berlin: Springer, 2003. – P. 464–478.

16. Frank A. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml] / A. Frank, A. Asuncion. – Irvine, CA: University of California, School of Information and Computer Science, 2010.

17. 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. – Vol. 17. – P. 255–287.

18. Fuel economy standards and automobile prices / R.E. Falvey, J. Frank, H.O. Fried, M. Babunovic // Journal of Transport Economics and Policy. – 1986. – Vol. 20. – P. 31–45.

Мельников Григорий Андреевич
Аспирант каф. вычислительной техники
Новосибирского государственного технического университета (НГТУ)
Тел.: 8-961-225-22-96
Эл. почта: grmel89@gmail.com

Губарев Василий Васильевич
Д-р техн. наук, профессор каф. вычислительной техники НГТУ
Тел.: 8 (383-3) 46-11-33
Эл. почта: gubarev@vt.cs.nstu.ru

Melnikov G.A., Gubarev V.V.
Method for regression tree induction based on the ant algorithms

We propose a novel method for regression tree induction based on modeling ant foraging behavior, combining techniques from both traditional regression tree induction algorithms and ant algorithms. The results of experiments on publicly available data sets show that the proposed method outperforms conventional algorithms for regression tree induction in accuracy and results in less complex solutions.
Keywords: machine learning, non-linear regression, piecewise linear regression, regression trees, model trees, ant algorithms, ant colony optimization.