П.Н. Дружков, Н.Ю. Золотых, И.Б. Мееров, А.Н. Половинкин

Нижегородский государственный университет им. Н.И. Лобачевского

РЕАЛИЗАЦИЯ АЛГОРИТМА ГРАДИЕНТНОГО БУСТИНГА ДЕРЕВЬЕВ РЕШЕНИЙ

Машинное обучение является подразделом весьма обширной области науки, изучающей искусственный интеллект. Алгоритмы машинного обучения приходят на помощь при решении задач, для которых сложно или невозможно придумать явный алгоритм решения, и, следовательно, практические сферы их применения огромны: от прогнозирования погоды, экономических и социальных процессов, медицинской диагностики до детектирования объектов на фото или видео, распознавания текста, речи, создания антивирусных программ и алгоритмов фильтрации рекламы и спама.

Авторами данной работы была выполнена программная реализация одного из наиболее перспективных алгоритмов обучения с учителем – алгоритма градиентного бустинга деревьев решений (GBT – gradient boosting trees) [1, 2]. Насколько известно авторам, это первая открытая C/C++ реализация данного метода. Мы приводим результаты экспериментов, проведенных с использованием этой реализации на наборах реальных данных, взятых из репозитория UCI (UCI Machine Learning Repository. URL: http://archive.ics.uci.edu/ml). Результаты вычислительного эксперимента свидетельствуют о конкурентоспособности предлагаемой реализации по сравнению с реализациями других алгоритмов. Разработка ведется в рамках популярной библиотеки компьютерного зрения с открытым исходным кодом OpenCV (OpenCV. URL: http://opencv.willowgarage.com.).

Постановка задачи. Одной из задач, изучаемых в машинном обучении, является *задача обучения с учителем*. В рамках этой задачи дано некоторое множество *объектов X*. Каждому объекту $x \in X$ поставлена в соответствие величина *y*, называемая *выходом*, или *ответом*, и принадлежащая множеству

допустимых ответов *Y*. Упорядоченная пара «объект–ответ» (x, y), где $x \in X, Y \in y$, называется *прецедентом*. Требуется восстановить зависимость между входом и выходом, основываясь на данных о конечном наборе прецедентов, называемом *обучающей выборкой*: $\left\{(x_i, y_i) \middle| x_i \in X, y_i \in Y, i = \overline{1, N}\right\}$. Другими словами, задача состоит в построении функции *f* из некоторого множества *K*, которая, получив на вход *x*, предсказала бы значение ответа *y* как можно точнее. Процесс нахождения *f* называется *обучением* или *настройкой* модели. В случае конечного *Y* говорят о *задаче классификации*, если $Y = \mathbf{R}$ – *о задаче восстановления регрессии* [3].

Метод решения. Один из общих подходов решения задач обучения заключается в комбинировании моделей. Две основные конкурирующие идеи данного подхода – бэггинг (*bagging* от *Bootstrap Aggregating*) [4] и бустинг (*boosting*) [5]. Первая из них состоит в построении множества независимых (между собой) моделей с дальнейшим принятием решения путем голосования в случае задачи классификации и усреднения в случае регрессии. Данный подход реализован в алгоритме случайных деревьев [6] (*random trees* или *random forest*). Бустинг, в противоположность бэггингу, обучает каждую следующую модель с использованием данных об ошибках предыдущих моделей.

Алгоритм градиентного бустинга деревьев решений является развитием бустинг-идеи. Он позволяет строить аддитивную функцию в виде суммы деревьев решений итерационно по аналогии с методом градиентного спуска. Данный подход позволяет расширить круг решаемых этим алгоритмом задач, а также зачастую получить выигрыш в точности предсказания.

Экспериментальные результаты. В данном разделе приведены некоторые экспериментальные результаты, показывающие достоинства и недостатки метода градиентного бустинга. Наряду с подходом, которому посвящена данная работа, были рассмотрены и конкурирующие алгоритмы: одиночные дере-

вья решений (алгоритм CART [7]), случайные деревья (случайные леса) [6], машина опорных векторов [8]. Программной основой проведенных экспериментов является открытая библиотека компьютерного зрения OpenCV: все результаты, относящиеся к конкурирующим алгоритмам, были получены непосредственно с помощью ее компонентов: CvDTree, CvRTrees, CvERTrees и CvSVM.

Эксперименты проводились на наборах реальных данных, взятых из репозитория UCI. Их краткие характеристики приведены в табл. 1.

Таблица 1

Тестовые наборы данных

Название	Общее количество прецедентов	Число переменных (количественные/ номинальные)	Количество классов
Восстановление регрессии			
auto-mpg	398	7 (4/3)	—
Computer hardware	209	8 (7/1)	—
Concrete slump	103	9 (9/0)	—
Forestfires	517	12 (10/2)	—
Boston housing	506	13 (13/0)	—
imports-85	201	25 (14/11)	—
Servo	167	4 (0/4)	—
Abalone	4177	8 (7/1)	—
Классификация			
Agaricus lepiota	8124	22 (0/22)	2
Liver disorders	345	6 (6/0)	2
Car evaluation	1728	6 (0/6)	4

Сравнение различных алгоритмов машинного обучения производилось по результатам 10-кратного скользящего контроля, с помощью которого осуществлялся выбор наилучших значений параметров для каждой конкретной задачи и каждого алгоритма. Тестовая ошибка считалась при помощи нескольких критериев:

1) средняя абсолютная ошибка (*average-absolute-error*):

$$\mathrm{Err}_{\mathrm{rms}} = \frac{1}{10}\sum_{k-1}^{10}\frac{\sum_{i=1}^{T_k}\left|y_{jk,i} - f(x_{jk,i})\right|}{T_k},$$

где T_k – объем k-й тестовой выборки, а $(x_{jk,i}, y_{jk,i})$ – i-й прецедент k-й тестовой выборки;

2) корень среднеквадратичной ошибки (*root-mean-squared error*):

$$\mathrm{Err}_{\mathrm{abs}} = \frac{1}{10}\sum_{k-1}^{10}\sqrt{\frac{\sum_{i=1}^{T_k}\left(y_{jk,i} - f(x_{jk,i})\right)^2}{T_k}};$$

3) для задачи классификации велся подсчет частоты неправильной классификации прецедентов тестовой выборки:

$$\mathrm{Err}_{\mathrm{misclass}} = \frac{1}{10}\sum_{k-1}^{10}\frac{\sum_{i=1}^{T_k}\left(y_{jk,i} \neq f(x_{jk,i})\right)}{T_k}.$$

Прецеденты с пропущенными значениями удалялись из обучающей и тестовой выборок при использовании CvSVM.

Наименьшие из полученных описанным способом ошибок приведены в табл. 2 и 3. Из этих данных видно, что алгоритм градиентного бустинга, как правило, дает результат, близкий к наилучшему, для конкретной задачи, что подтверждает его универсальность и способность подстраиваться под специфику решаемой задачи. В то же время для некоторых из рассматриваемых задач существуют алгоритмы, дающие меньшую тестовую ошибку.

Для некоторых из перечисленных в табл. 1 наборов данных были проведены и другие эксперименты. Данные случайным образом разбивались на обучающую и тестовую выборки в соотношении 9:1 для наборов Boston housing, Computer hardware и Servo, и 8:2 для Abalone, после чего выполнялось обучение и предсказание. Процесс повторялся 100 раз для каждой задачи. Данный подход также позволяет проследить влияние значений некоторых параметров на обобщающую способность метода и служит методом для сравнения различных алгоритмов.

Таблица 2

Корни среднеквадратических ошибок (RMS) и средние абсолютные ошибки (ABS), полученные различными алгоритмами при 10-кратном перекрестном контроле

Название	Градиентный бустинг (GBT)		Дерево решений (CvDTree)		Случайные деревья (CvRTrees)		Случайные деревья (CvERTrees)		Машина опорных векторов (CvSVM)	
	RMS	ABS	RMS	ABS	RMS	ABS	RMS	ABS	RMS	ABS
auto-mpg	2,682	2	3,133	2,238	2,653	1,879	2,955	2,147	4,042	2,981
Computer hardware	23,55	12,62	30,13	15,62	26,02	11,62	19,12	9,631	50,51	37
Concrete slump	2,524	2,257	3,727	2,923	3,193	2,6	2,945	2,359	2,164	1,767
Forestfires	35,15	18,74	38,09	17,26	35,22	17,79	34	16,64	45,51	12,9
Boston housing	2,914	2,033	3,653	2,602	3,042	2,135	3,127	2,196	5,71	4,049
imports-85	1827	1306	2317	1649	1821	1290	2146	1487	2583	1787
Servo	0,385	0,238	0,455	0,258	0,418	0,247	0,686	0,42	0,884	0,655
Abalone	2,144	1,47	2,281	1,604	2,115	1,492	2,124	1,498	2,644	2,091

Таблица 3

Средние частоты неправильной классификации прецедентов, полученные различными алгоритмами при 10-кратном перекрестном контроле

Название	Градиентный бустинг (GBT)	Дерево решений (CvDTree)	Случайные деревья (CvRTrees)	Случайные деревья (CvERTrees)	Машина опорных векторов (CvSVM)
Agaricus lepiota	0	0,000123	0	0	0
Liver disorders	0,251357	0,30543	0,227828	0,254299	0,278582
Car evaluation	0	0,0513824	0,0364987	0,0394574	0,0509819

Средние значения ошибок, полученных в результате этих экспериментов, приведены в табл. 4. На рассмотренных нами

наборах данных градиентный бустинг в большинстве случаев превосходит метод случайных деревьев или показывает сравнимый результат.

Таблица 4

Сравнение средних ошибок алгоритмов градиентного бустинга и случайных деревьев

Набор данных	Тестовая ошибка алгоритма градиентного бустинга		Тестовая ошибка алгоритма случайных деревьев	
	Средняя абсолютная ошибка	Средняя квадратичная ошибка	Средняя абсолютная ошибка	Средняя квадратичная ошибка
Boston housing	1,995	8,234	2,13	9,689
Computer hardware	10,29	1096,1	13,996	2047,8
Servo	0,198	0,237	0,24	0,257
Abalone	1,517	4,732	1,514	4,653

На рисунке изображены бокс-диаграммы, которые показывают разброс результатов эксперимента на наборе данных Servo. Слева находятся диаграммы, соответствующие алгоритму градиентного бустинга с использованием деревьев решений различных размеров: глубина от 1 до 4. Самая правая бокс-диаграмма соответствует методу случайных деревьев. Таким образом, можно наблюдать снижение тестовой ошибки бустинга при увеличении глубины используемых деревьев. Также эта диаграмма иллюстрирует небольшое превосходство алгоритма градиентного бустинга над случайными деревьями на рассматриваемом наборе данных.

Данные экспериментов свидетельствуют о конкурентоспособности программной реализации, выполненной авторами данной работы. Текущая реализация интегрирована в библиотеку с открытыми исходными кодами OpenCV, что, во-первых, дает возможность легко проводить сравнение качества работы различных алгоритмов машинного обучения для конкретных прикладных задач, во-вторых, предоставляет удобные средства для применения алгоритма градиентного бустинга деревьев решений в задачах компьютерного зрения.

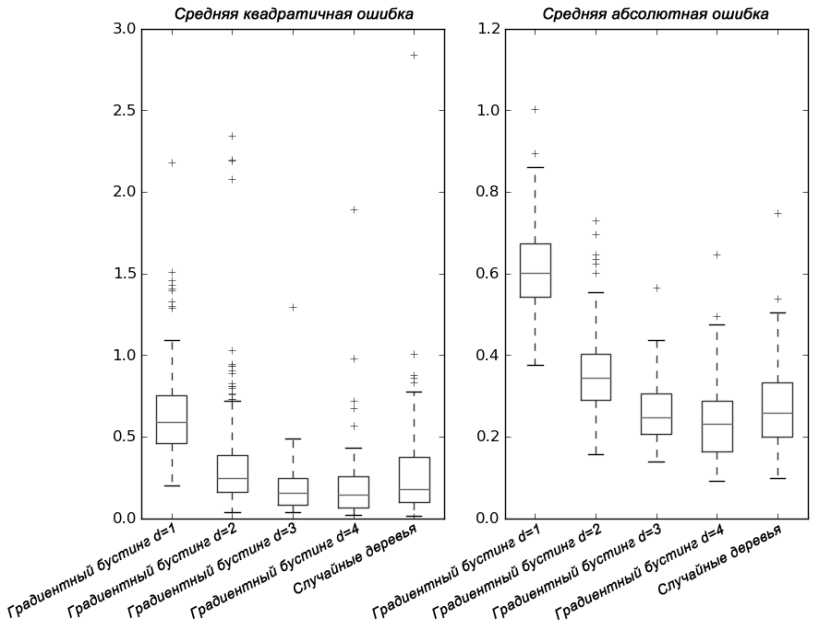

Рис. Бокс-диаграммы для результатов тестирования алгоритма градиентного бустинга с использованием деревьев решений различной глубины и метода случайных деревьев на наборе данных Servo

Мы благодарим В.Л. Ерухимова (компания ITSeez) за ценные замечания и постоянное внимание к работе.

Работа выполнена при поддержке федеральной целевой программы «Научные и научно-педагогические кадры инновационной России», госконтракт 02.740.11.5131.

Список литературы

1. Friedman J.H. Greedy Function Approximation: a Gradient Boosting Machine. Technical report. Dept. of Statistics. – Stanford University, 1999.

2. Friedman J.H. Stochastic Gradient Boosting. Technical report. Dept. of Statistics. – Stanford University, 1999.

3. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. Springer, 2008.

4. Breiman L. Bagging predictors // Machine Learning. – 1996. – Vol. 26, No. 2. – P. 123–140.

5. Freund Y., Schapire R. Experiments with a New Boosting Algorithm // Machine Learning: Proceedings of the Thirteenth International Conference, 1996.

6. Breiman L. Random Forests // Mach. Learn. – 2001. – Vol. 45, No. 1. – P. 5–32.

7. Breiman L., Friedman J., Olshen R., Stone C. Classification and Regression Trees. – Wadsworth, 1983.

8. Вапник В.Н. Восстановление зависимостей по эмпирическим данным. – М.: Наука, 1979. – 448 с.

О.В. Джосан

Московский государственный университет имени М.В. Ломоносова

О ПЕРСПЕКТИВАХ И ПРОБЛЕМАХ ЭКЗАФЛОПСНЫХ ВЫЧИСЛЕНИЙ В РОССИИ

Осенью 2009 года, во время проведения конференции «Научный сервис в сети Интернет–2009» [1] группой энтузиастов было зарегистрировано доменное имя exascale.ru – российский аналог домена exascale.org международного сообщества International Exascale Software Project [2]. Сделано это было в рамках вечной проблемы «научных отцов и детей» – молодежь требует революции, а старшее поколение – соблюдения традиций. Молодым хотелось доказать, что те проблемы, которые обсуждают «отцы», устарели и потеряли актуальность, необходимо выходить на новый уровень и думать о препятствиях, которые нам обещают к 2018 году. Иначе шансов быть на достойном месте в мировом суперкомпьютерном сообществе у России немного. Однако широкой поддержки тогда эта тема не получила и была в целом признана несвоевременной.