Оптимизация, системный анализ и исследование операций

© 2025 г. О.В. СЕНЬКО, д-р физ-мат. наук (senkoov@mail.ru),
А.А. ДОКУКИН, канд. физ.-мат. наук (dalex@ccas.ru)
(Федеральный исследовательский центр
«Информатика и управление» РАН, Москва),
Ф.А. МЕЛЬНИК (melnik.tedor@gmail.com)
(Московский государственный университет им. М.В. Ломоносова)

ИСПОЛЬЗОВАНИЕ АНСАМБЛЕЙ С УВЕЛИЧЕННОЙ ДИВЕРГЕНЦИЕЙ В ПРОСТРАНСТВЕ ПРОГНОЗОВ В РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМАХ

Рассматривается метод дивергентного решающего леса, основанный на достижении более высокой дивергенции в пространстве прогнозов по сравнению со стандартным случайным решающим лесом за счет включения на каждом шаге в ансамбль нового дерева T_x, которое строится исходя из условий минимизации специального функционала, являющегося разностью квадратичной ошибки T_x и квадрата расхождения прогнозов T_x и текущего ансамбля. Метод является развитием аналогичных ранее разработанных методов, которые предназначены для прогнозирования числовых переменных. В работе представлены результаты применения метода дивергентного решающего леса для решения задач классификации, возникающих при создании рекомендательных систем. Исследована зависимость эффективности прогноза от глубины деревьев и одного из ключевых параметров алгоритма, регулирующего вклад двух составляющих в минимизируемый функционал. Исследования показали, что точность предлагаемой технологии заметно превышает точность случайного решающего леса и близка к точности метода CatBoost.

Ключевые слова: ансамблевый метод, машинное обучение, рекомендательные системы.

DOI: 10.31857/S0005231025040061, **EDN:** CAPHAM

1. Введение

Методы, основанные на использовании ансамблей регрессионных или решающих деревьев, являются одной из основных ветвей современного машинного обучения и активно используются при решении разнообразных прикладных задач [1]. Можно выделить два популярных направления ансамблевых алгоритмов: случайные леса [2] и градиентный бустинг [3]. Для обеих технологий разработаны методы, предназначенные для решения как задач

автоматической классификации, так и задач прогнозирования числовых переменных. При прогнозировании целевой числовой переменной Y по признакам $X_1, \ldots, X_n$ обучение производится по выборке $S = \{s_1 = (y_1, x_1), \ldots, s_m = (y_m, x_m)\}$, где y_j и x_j – значения целевой переменной Y и вектора значений признаков $X_1, \ldots, X_n$ объекта s_j соответственно. Задачи классификации с двумя непересекающимися классами также могут рассматриваться как задачи прогнозирования бинарной числовой переменной с обучением по выборке аналогичного вида. Сформулируем основные различия между методами, основанными на градиентном бустинге, и случайными лесами. В методе случайных лесов отдельные деревья строятся независимо. При этом дерево на шаге k нацелено на прогнозирование целевой переменной Y и строится по выборке S_k, получаемой с использованием процедур бэггинга [4] и метода случайных подпространств [5]. Процедура бэггинга заключается в построении новой выборки как случайной выборки с возвращениями из S. Метод случайных подпространств предполагает использование случайного подмножества исходного множества признаков $X_1, \ldots, X_n$ фиксированного размера при обучении каждого нового дерева случайного леса. Выходной прогноз ансамбля $\hat{A}_k$, состоящего из k деревьев, вычисляется как средний прогноз по всем деревьям, вошедшим в него, т.е. $\hat{A}_k = \frac{1}{k}\sum_{j=1}^{k} A_j$. В методе градиентного бустинга оптимальный алгоритм ищется как линейная комбинация регрессионных деревьев $\hat{A}_N = T_0 + \alpha_1 \times T_1 + \ldots + \alpha_N \times T_N$, где исходный алгоритм T_0 обычно вычисляет прогноз, тождественно равный среднему значению Y. На каждом шаге в текущую линейную комбинацию $\hat{A}_{k-1} = A_0 + \alpha_1 \times T_1 + \ldots + \alpha_{k-1} \times T_{k-1}$ добавляется новое слагаемое $\alpha_k \times T_k$. При этом выбор последнего производится исходя из цели минимизации функции потерь для линейной комбинации $\hat{A}_k = T_0 + \alpha_1 \times T_1 + \ldots + \alpha_k \times T_k$. Последнее достигается за счет использования процедуры градиентного спуска.

Пусть $L(\hat{A}_{k-1}, S)$ – некоторая функция потерь, зависящая от прогнозов, вычисляемых $\hat{A}_{k-1}$ и истинных значениях Y на объектах S. Сопоставим прогнозам, вычисляемым для объектов S, переменные $z_1, \ldots, z_m$. Согласно методу градиентного спуска предполагаемый минимум L будет достигаться для прогноза на k-м шаге для объекте s_j при условии

$$\hat{A}_k(s_j) = \hat{A}_{k-1}(s_j) - \eta \frac{\partial L}{\partial z_j}\bigg|_{z_j = \hat{A}_{k-1}(s_j)}.$$

Однако прогноз в указанном виде невозможно вычислить для объектов, для которых истинное значение y_j неизвестно, поскольку функция потерь и ее частные производные определены только при известных значениях Y. Поэтому авторами метода было предложено использовать вместо величин g_j их прогнозы, вычисленные с помощью регрессионного дерева, обученного по выборке $\{(g_1, x_1), \ldots, (g_m, x_j)\}$. Именно это дерево и является деревом T_k, добавляемым в линейную комбинацию. При обучении дерева T_k использовались бэггинг и метод случайных подпространств.

В настоящее время существует несколько модификаций градиентного бустинга, включающих XGBoost [6], LightGBM [7], CatBoost [8]. При этом метод случайного леса остается практически неизменным с момента своего создания. Вместе с тем отсутствуют убедительные доказательства того, что используемая в случайных лесах схема генерации ансамбля действительно является оптимальной при использовании простого усреднения в качестве агрегирующей процедуры.

2. Построение лесов с увеличенной дивергенцией

В [9–11] был предложен новый подход, нацеленный на построение ансамбля исходя из условия минимизации потерь прогнозов, вычисляемых как средние прогнозы по ансамблю. Иными словами, была поставлена задача подбора деревьев в ансамбль таким образом, чтобы потери были по возможности минимальными при использовании в качестве прогноза $\hat{A}_k$. При этом в качестве функционала потерь использовалась обычная среднеквадратичная ошибка:

$$L(S, A) = \frac{1}{m}\sum_{j=1}^{m}\left(y_j - \hat{A}_k(x_j)\right)^2.$$

Было показано, что средний квадрат ошибки для алгоритма A_k вычисляется по формуле

$$L(S, \hat{A}_k) = \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{k}(y_j - T_i(x_j))^2 - \frac{1}{km}\sum_{j=1}^{m}\sum_{i=1}^{k}\left(\hat{A}_k(x_j) - T_i(x_j)\right)^2. \tag{1}$$

Можно поставить задачу поиска такого набора деревьев $T_1, \ldots, T_k$, при котором достигается минимум потерь (1). Данная задача является чрезвычайно трудоемкой. Однако может быть использован простой эвристический подход, когда на каждом шаге в ансамбль добавляется дерево, которое одновременно хорошо аппроксимирует зависимость и максимально удалено по вычисляемым прогнозам от текущего ансамбля. Для реализации такого подхода в [9–11] было предложено на шаге k добавлять дерево T_k, при котором по возможности минимален был бы функционал

$$Q(T_k, \hat{A}_k, \mu) = \frac{1}{m}\sum_{j=1}^{m}(y_j - T_k(x_j))^2 - \mu\frac{1}{m}\sum_{j=1}^{m}\left(\hat{A}_k(x_j) - T_k(x_j)\right)^2, \tag{2}$$

где μ – параметр из интервала $(0, 1)$. Функционал $Q(T_k, \hat{A}_k, \mu)$ может быть переписан в виде

$$\begin{aligned} Q(T_k, \hat{A}_k, \mu) &= \frac{1}{m}\sum_{j=1}^{m}(y_j - T_k(x_j))^2 - \\ &- \theta\frac{1}{m}\sum_{j=1}^{m}\left(\hat{A}_k(x_j) - T_k(x_j)\right)^2, \ \theta = \frac{k^2\mu}{(k+1)^2}. \end{aligned} \tag{3}$$

Отметим, что минимум функционала (2) достигается, если каждой точке x_j сопоставить прогноз t_j^o, равный значению вспомогательной переменной t_j, при котором достигается минимум функции

$$(y_j - t_j)^2 - \theta \left(\hat{A}_{k-1}(x_j) - t_j \right)^2 .$$

Нетрудно показать, что необходимые условия экстремума выполняются в случае, когда

$$t_j^o = \frac{y_j - \theta \hat{A}_{k-1}(x_j)}{1 - \theta} . \tag{4}$$

Очевидно, что $\theta \to 1$ при $k \to \infty$ и $\mu = 1$, что приводит к неустойчивости оценок по формуле (4). С этим и связано использование дополнительного множителя $\mu \in (0, 1)$. Для поиска оптимального ансамбля в [10, 11] был предложен подход, в котором на шаге k в ансамбль добавляется новое дерево T_k, прогнозирующее величины t_j^o, рассчитываемые по (4) в зависимости от векторных описаний. Дерево T_k обучается по выборке $\{(t_1^o, x_1), \ldots, (t_m^o, x_m)\}$. Использование гиперпараметра μ, оказывающего влияние на эффективность построенных ансамблей, является эвристическим приемом. Теоретически обоснованный способ расчета оптимального значения μ отсутствует. Предполагается, что оптимальное значение μ может быть подобрано в результате экспериментов, что и является одной из целей настоящего исследования.

3. Применение для автоматической классификации

Изложенный выше подход основан на использовании квадратичных функций потерь и предназначен для решения задач прогнозирования числовых переменных. Однако метод может быть использован также для решения задач автоматической классификации. При решении задач автоматической классификации для оценки точности аппроксимации зависимости принято использовать кросс-энтропию, т.е. величину, пропорциональную взятому со знаком минус логарифму функции правдоподобия, рассчитанной исходя из распределения Бернулли. В случае задачи бинарной классификации целевая величина Y принимает значения из множества $\{0, 1\}$, где равенство $Y = 1$ указывает на принадлежность целевому классу K. Предположим, что некоторое дерево T вычисляет вероятности принадлежности классу K объектов из $S = \{s_1 = (y_1, x_1), \ldots, s_m = (y_m, x_m)\}$. Пусть p_j – вероятность принадлежности объекта s_j целевому классу, вычисляемая деревом T от его описания x_j. Потери в смысле кросс-энтропии для дерева T на выборке S оцениваются по формуле

$$L(T, S) = -\ln \prod_{j=1}^{m} p_j^{y_j} (1 - p_j)^{1 - y_j} = -\sum_{j=1}^{m} [y_j \ln p_j + (1 - y_j) \ln(1 - p_j)]. \tag{5}$$

Потери по формуле (5) являются аналогом квадратичных потерь для дерева T на выборке S и соответствуют левому слагаемому в (2). Для оценки с использованием отклонения добавляемого в ансамбль дерева T_k от ансамбля A_k может быть предложен следующий подход. Пусть p_j^k – вероятность принадлежности объекта s_j целевому классу, вычисляемая деревом T_k от его описания x_j, $\hat{p}_j^k$ – вероятность принадлежности объекта s_j целевому классу, вычисляемая ансамблем A_k. Отклонение добавляемого дерева от ансамбля оценим по формуле

$$D(T_k, A_k, S) = -\ln \prod_j^m (p_j^k)^{\hat{p}_j^k} (1-p_j^k)^{1-\hat{p}_j^k} = \sum_{j=1}^m \Big[-\hat{p}_j^k \ln p_j^k - (1-\hat{p}_j^k) \ln(1-p_j^k) \Big].$$

Для поиска оптимальных значений вероятностей p_j^k может быть использован тот же самый подход, что использовался в случае квадратичных потерь. На первом этапе ищутся величины $p_1^o, \ldots, p_m^o$, для которых достигается минимум функционала

$$Q(T_k, A_k, S) = L(T, S) - \mu D(T_k, A_k, S).$$

Далее новое, добавляемое в ансамбль дерево T_k обучается по выборке

$$\{(p_1^o, x_1), \ldots, (p_m^o, x_m)\}.$$

Ансамбль A_k на шаге k является неизвестным. Поэтому вместо $Q(T_k, A_k, S)$ может быть использован близкий ему функционал

$$Q(T_k, A_{k-1}, S) = L(T, S) - \mu D(T_k, A_{k-1}, S). \tag{6}$$

Нетрудно показать, что минимум функционала (6) достигается при

$$p_j^o = \frac{y_j - \mu \hat{A}_{k-1}(x_j)}{1-\mu}. \tag{7}$$

К сожалению, использование (7) приводит к нарушению условия $p_j^o \in [0,1]$. Данное условие может быть сохранено при переходе к задаче условной оптимизации. Однако последнее приводит к существенному усложнению алгоритма. Более простым является решение, основанное на использовании (7) с последующим обучением нового дерева по выборке

$$\{(p_1^o, x_1), \ldots, (p_m^o, x_j)\}. \tag{8}$$

Для разделения классов оценка величины p_j сравнивается с порогом, для поиска которого используются средства ROC (Receiver Operating Characteristic) анализа. Указанный метод далее будем называть дивергентным решающим лесом (Divergent Decision Forest, DDF).

4. Исследование эффективности ансамблей с увеличенной дивергенцией при построении рекомендательных систем

Предложенный в предыдущих разделах метод зависит от набора гиперпараметров, в которые входят как множитель μ, так и гиперпараметры, используемые при построении отдельных деревьев. Успешное применение большинства методов машинного обучения зависит от выбранных значений гиперпараметров. Вместе с тем точные теоретические оценки для выбора гиперпараметров обычно отсутствуют. Поэтому их оптимальные значения приходится искать через эксперименты с данными. Целью настоящей работы являлся экспериментальный поиск оптимальных значений гиперпараметров для дивергентного решающего леса. Исследования проводились в рамках решения задач прогнозирования пользовательских предпочтений, возникающих при создании рекомендательных систем. Выбор данного класса задач связан как с широким распространением рекомендательных систем в различных отраслях экономики, так и с интенсивным использованием в этой области методов машинного обучения [12]. Рассматривались три задачи оценивания предпочтений интернет-пользователей при выборе компьютерной игры и две задачи оценивания предпочтений интернет-пользователей при выборе стикера в социальных сетях. Оценивание проводилось на основе информации о взаимодействии пользователей с соответствующими каталогами. Для оценки использовался показатель HR5 (hitrate at 5) – доля пользователей, для которых среди первых пяти рекомендаций оказался хотя бы один релевантный объект. Используются следующие обозначения: k – число пользователей, p – число объектов, n – число признаков, M – число строк в обучающей выборке. Характеристики рассмотренных задач представлены в табл. 1.

Таблица 1. Характеристики рассмотренных задач

Задача	k	p	n	M
game0	414	187	41	4132
game1	1706	190	41	16 327
game2	3888	398	55	29 065
sticker1	2685	118	40	27 032
sticker2	5942	197	41	44 613

Целью работы было изучение зависимости точности алгоритма в смысле показателя HR5 от глубины деревьев и параметра μ. Результаты для задач game0 и sticker1 представлены в табл. 2 и 3 соответственно. Значение $\mu = 0$ соответствует обычной модели случайного решающего леса (RF).

Из таблиц видно, что при глубине деревьев, не превышающей 7, HR(5) возрастает с ростом μ. Вместе с тем при больших значениях μ при глубине деревьев более 7 показатель HR(5) снижается.

В табл. 4 представлено сравнение предлагаемой технологии со стандартными случайными решающими лесами, а также с методом CatBoost. Стандарт-

Таблица 2. Связь показателя HR(5) с глубиной деревьев и параметром μ для задачи game0

	$\mu = 0{,}0$	$\mu = 0{,}25$	$\mu = 0{,}5$	$\mu = 0{,}75$	$\mu = 0{,}9$
$depth = 3$	0,584	0,594	0,599	0,613	0,640
$depth = 5$	0,596	0,604	**0,623**	**0,630**	**0,657**
$depth = 7$	0,591	0,606	0,621	0,621	0,623
$depth = 9$	0,592	**0,618**	0,606	0,606	0,570
$depth = 11$	**0,606**	0,601	0,606	0,611	0,493

Таблица 3. Связь показателя HR(5) с глубиной деревьев и параметром μ для задачи sticker1

	$\mu = 0{,}0$	$\mu = 0{,}25$	$\mu = 0{,}5$	$\mu = 0{,}75$	$\mu = 0{,}9$
$depth = 3$	0,447	0,449	0,448	0,458	0,514
$depth = 5$	0,482	0,488	0,498	**0,516**	**0,527**
$depth = 7$	0,475	0,491	**0,501**	0,508	0,513
$depth = 9$	0,479	0,492	0,500	0,507	0,477
$depth = 11$	**0,482**	**0,494**	0,499	0,494	0,467

Таблица 4. Лучшее значение HR(5) для каждой из моделей

	RF	DDF ($\mu \neq 0$)	catboost
games0	0,606	0,657	**0,664**
games1	0,642	**0,654**	0,637
games2	0,513	0,550	**0,562**
stickers1	0,482	**0,527**	0,513
stickers2	0,337	0,372	**0,385**
среднее	0,516	0,552	0,5522

ным решающим лесам соответствует $\mu = 0$. Из таблицы видно, что точность в смысле метрики HR5 для предлагаемой технологии заметно превышает точность случайного решающего леса и близка к точности метода CatBoost.

5. Заключение

Разработан метод дивергентного решающего леса, предназначенный для решения задач бинарной классификации. Метод основан на развитии подходов, ранее разработанных для решения задач регрессии [9–11]. На примере задач, связанных с созданием рекомендательных систем, проведено исследование метода дивергентного решающего леса (DDF), основанного на достижении более высокой дивергенции в пространстве прогнозов по сравнению со стандартным случайным решающим лесом. Показана существенная зависимость эффективности от глубины деревьев и от коэффициента μ, регулирующего относительные вклады составляющих, отвечающих за аппроксимацию целевой переменной и дивергенции ансамблей. При этом при небольшой

глубине деревьев точность алгоритма возрастает с ростом μ. Такого эффекта не наблюдается при большой глубине деревьев. Однако в целом DDF выигрывает у стандартного случайного решающего леса, соответствующего $\mu = 0$. Эксперименты показали, что эффективность DDF приближается к широко известной бустинговой модели CatBoost. Метод DDF основан на последовательной генерации наборов деревьев. Поэтому его быстродействие близко к быстродействию обычных случайных лесов и вариантов градиентного бустинга. Одной из проблем метода является выбор оптимального значения гиперпараметра μ. Данная проблема может быть решена с помощью известных средств выбора гиперпараметров. Однако проведенные эксперименты указывают, что более высокая эффективность достигается при высоких значениях гиперпараметра, т.е. при μ, равном 0,8 или 0,9. В отличие от методов решающих и регрессионых деревьев, обладающих высокой прозрачностью и интерпретируемостью, методы, использующие большие ансамбли деревьев, эти свойства, к сожалению, утрачивают. Последнее касается и метода DDF. Данный недостаток может быть компенсирован с помощью различных известных методов достижения интерпретируемости. Для повышения прозрачности могут быть использованы методы интеллектуального анализа данных, а также статистический анализ.

СПИСОК ЛИТЕРАТУРЫ

1. *Hastie T., Tibshirani R., Friedman J.* The Elements of Statistical Learning Data Mining, Inference, and Prediction. Springer Series in Statistics. New York: Springer, 2009.
2. *Breiman L.* Random Forests // Machine Learning. 2001. V. 45. No. 1. P. 5–32.
3. *Friedman J.* Stochastic gradient boosting // Comput. Statist. Data Anal. 2002. V. 38. No. 4. P. 367–378.
4. *Breiman L.* Bagging predictors // Machine Learning. 1996. No. 24. P. 123–140.
5. *Tin Kam Ho.* The random subspace method for constructing decision forests // IEEE Transact. Patt. Machine Intelligen. 1998. V. 20. No. 8. P. 832–844.
6. *Chen T., Guestrin C.* XGBoost: A Scalable Tree Boosting System // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13–17 / eds. Krishnapuram B., Shah M., Smola A.J., Aggarwal C.C., Shen D., Rastogi R. 2016. P. 785–794.
7. *Ke G., Meng Q., Finley T. et al.* LightGBM: A Highly Efficient Gradient Boosting Decision Tree // NIPS'17: Proc. 31st Int. Conf. Neural Inform. Proc. Syst. 2017. P. 3149–3157.
8. *Hancock J.T., Khoshgoftaar T.M.* CatBoost for big data: an interdisciplinary review // J. Big Data. 2020. V. 7. No. 94.
9. *Zhuravlev Yu.I., Senko O.V., Dokukin A.A., Kiselyova N.N., Saenko I.A.* Two-Level Regression Method Using Ensembles of Trees with Optimal Divergence // Dokl. Math. 2021. V. 103. P. 1–4.
10. *Докукин А.А., Сенько О.В.* Новый двухуровневый метод машинного обучения для оценивания вещественных характеристик объектов // Изв. РАН. Теория и системы управления. 2023. No. 4. P. 17–24.

11. *Senko O.V., Dokukin A.A., Kiselyova N.N., et al.* New Two-Level Ensemble Method and Its Application to Chemical Compounds Properties Prediction // Lobachev. J. Math. 2023. V. 44. No. 1. P. 188–197.
12. *Roy D., Dutta M.* A systematic review and research perspective on recommender systems // J. Big Data. 2022. V. 9. No. 59.

Статья представлена к публикации членом редколлегии А.А. Галяевым.

Поступила в редакцию 29.11.2024

После доработки 09.01.2025

Принята к публикации 14.01.2025