ИЗВЕСТИЯ РАН. ТЕОРИЯ И СИСТЕМЫ УПРАВЛЕНИЯ, 2023, № 4, с. 17–24

ТЕОРИЯ СИСТЕМ И ОБЩАЯ ТЕОРИЯ УПРАВЛЕНИЯ

УДК 004.93'14,004.021

НОВЫЙ ДВУХУРОВНЕВЫЙ МЕТОД МАШИННОГО ОБУЧЕНИЯ ДЛЯ ОЦЕНИВАНИЯ ВЕЩЕСТВЕННЫХ ХАРАКТЕРИСТИК ОБЪЕКТОВ[1]

© 2023 г. А. А. Докукин[a,*], О. В. Сенько[a,]**

[a]*Федеральный исследовательский центр "Информатика и управление" РАН, Москва, Россия*

**e-mail: dalex@ccas.ru*

***e-mail: senkoov@mail.ru*

Поступила в редакцию 14.02.2023 г.
После доработки 01.03.2023 г.
Принята к публикации 03.04.2023 г.

Рассматривается новый двухуровневый ансамблевый регрессионный метод, его модификации и применение в прикладных задачах. Ключевой особенностью метода является нацеленность его на построение ансамбля предикторов, хорошо аппроксимирующих целевую переменную, и при этом состоящего из алгоритмов, по возможности отличающихся друг от друга по вычисляемым прогнозам. Построение ансамбля, обладающего указанными свойствами, на первом этапе производится через оптимизацию специального функционала, выбор которого теоретически обосновывается в работе. На втором этапе по сформированным этим ансамблем прогнозам вычисляется коллективное решение. Кроме того, описываются некоторые эвристические модификации, положительно сказывающиеся на качестве прогноза в прикладных задачах. Эффективность метода подтверждается результатами, полученными для конкретных прикладных задач.

DOI: 10.31857/S0002338823040029, **EDN:** OCEZGW

Введение. Ансамблевые методы имеют весьма продолжительную историю и являются существенной частью технологий машинного обучения, используемых при решении задач обучения по прецедентам: классификации или предсказания числовых переменных [2, 3]. Под ансамблевым методом обычно понимается метод, вычисляющий решения в два этапа: отдельными алгоритмами ансамбля и коллективным алгоритмом.

Среди ансамблевых технологий наибольшее распространение получили метод случайных лесов [4] и метод градиентного бустинга [5]. Данные технологии успешно использовались ранее, в том числе для задач оценивания вещественных характеристик по признаковым описаниям объектов.

Задача обучения по прецедентам в приведенном смысле определяется следующим образом. Рассмотрим совокупность объектов, позволяющих измерить или вычислить n своих числовых характеристик (признаков) $X_1,\ldots,X_n$. Пусть для некоторых из этих объектов также измерен целевой признак Y, их будем называть прецедентами. Обучающей выборкой назовем множество прецедентов $S = \{(y_1,x_1),\ldots,(y_m,x_m)\}$, где y_i, $i = \overline{1,m}$ – значения целевой переменной Y для i-го объекта, а x_i – вектор признакового описания i-го объекта, $x_i = (X_1^i,\ldots,X_n^i)$. Требуется построить алгоритм A для определения (предсказания) значения Y для остальных объектов: $A(x) = y$. В случае, когда Y принимает категориальные значения, задача может называться задачей классификации, а в случае, когда Y является непрерывной числовой характеристикой, – регрессией.

В методе случайных лесов [4] деревья генерируются независимо с использованием комбинации бэггинга и метода случайных подпространств [6, 7]. Иными словами, каждое новое дерево T_k, добавляемое в ансамбль на шаге k, строится по выборке S_k^b, являющейся выборкой с возвра-

[1] Работа проведена в рамках государственного задания (проект 0063-2016-0003) с помощью инфраструктуры ЦКП "Информатика" ФИЦ ИУ РФН [1].

щением из проекции исходной обучающей выборки $S_k^b \subset S^b = \{(y_1, x_1^b), \ldots, (y_m, x_m^b)\}$, где x_i^b – проекция вектора x_i на признаковое подпространство $X^b \subset \{X_1, \ldots, X_n\}$. Процесс генерации деревьев прекращается, когда общее число деревьев в ансамбле достигает заранее заданного числа. В методе случайных лесов применяются простые коллективные решения: при решении задач классификации объект относится в тот класс, к которому его отнесло большинство деревьев ансамбля; при решении регресионных задач коллективный прогноз вычисляется как средний прогноз по ансамблю.

В методах, основанных на градиентном бустинге, каждое новое дерево T_k, добавляемое в ансамбль на шаге k, осуществляет один шаг градиентного спуска. Пусть функция $l(y_j, A(x_j))$ описывает потери, возникающие при использовании в качестве прогноза Y в точке x_j величины $A(x_j)$, например квадратичная ошибка $l(y_j, A(x_j)]) = (y_j - A(x_j))^2$. Для обучения вычисляется градиент l по $(A(x_1), \ldots, A(x_m))$, $\nabla l = \{r_1, \ldots, r_m\}$, где $r_j = \partial l(y_j, A(x_j))/\partial A(x_j)$, $j = \overline{1,m}$. Очвидно, для квадратичной ошибки $r_j = 2(A(x_k) - y_j)$. Пусть A_k – ансамбль, построенный к k-му шагу:

$$A_1 = T_0,$$

$$A_k = T_0 - \sum_{i=1}^{k-1} \varepsilon_i T_i, \quad k > 1,$$

где T_0 – некоторое начальное приближение, например тождественный ноль. Если теперь T_k обучается по выборке $\{r_1^k, \ldots, r_m^k\}$, тем самым аппроксимируя градиент функции потерь при $A = A_k$, а ε_k – шаг градиентного спуска на шаге k, то ансамбль, построенный таким образом, будет с каждым шагом все точнее оптимизировать заданный функционал, а значит, приближать целевую переменную.

Метод случайных регрессионных лесов и метод градиентного бустинга широко применяются при решении разнообразных прикладных задач, демонстрируя во многих случаях высокую эффективность. Причем ни один из них не является бесспорным лидером – на практике встречаются задачи, где то один, то другой из этих методов демонстрируют превосходство, иногда со значительным отрывом. Вместе с тем можно предположить, что два указанных метода не исчерпывают все возможности достижения высокой эффективности ансамблевых решений. Возникает даже желание объединить эти подходы, что в какой-то мере реализуется в предлагаемом методе.

1. Оптимизируемый функционал. В методе случайных лесов ансамбли генерируются случайно, что, очевидно, не обеспечивает их оптимальности. В то же время градиентный бустинг не является в полной мере ансамблевым методом – это, несомненно, сумма алгоритмов, но каждый в отдельности аппроксимирует не целевую переменную, а производные функции потерь. В нашем методе предлагается совместить лучшее из двух подходов. С одной стороны, будем строить ансамбль из разнообразных элементов, каждый из которых в определенной степени случаен, и аппроксимирует целевую переменную. При этом сами элементы будут строиться путем оптимизации специального функционала, к выводу которого мы переходим.

Рассмотрим итерационную процедуру построения ансамбля. Обозначим через $B_i(x)$ произвольный алгоритм: как дерево, так, например, и линейную комбинацию деревьев, получаемый на i-м шаге. К шагу k ансамбль будет состоять из всех алгоритмов, построенных на предыдущих шагах, а ансамблевым решением $A_k(x)$ считаем среднее этих алгоритмов:

$$A_1(x) = O(x),$$

$$A_k(x) = \frac{1}{k-1} \sum_{i=1}^{k-1} B_i(x), \quad k > 1,$$

где $O(x)$ – тождественный ноль.

В качестве функционала ошибки нас будет интересовать среднее квадратичное отклонение:

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

Заметим, что этот функционал можно выразить через средний квадрат отклонения по всем объектам и алгоритмам:

$$\frac{1}{km}\sum_{j=1}^{m}\sum_{i=1}^{k}(y_j - B_i(x_j))^2,$$

для чего рассмотрим последний:

$$\frac{1}{km}\sum_{j=1}^{m}\sum_{i=1}^{k}(y_j - B_i(x_j))^2 = \frac{1}{km}\sum_{j=1}^{m}\sum_{i=1}^{k}(y_j - A_k(x_j) + A_k(x_j) - B_i(x_j))^2 =$$

$$= \frac{k}{km}\sum_{j=1}^{m}(y_j - A_k(x_j))^2 + \frac{1}{km}\sum_{j=1}^{m}\sum_{i=1}^{k}(A_k(x_j) - B_i(x_j))^2 -$$

$$-\frac{1}{km}\sum_{j=1}^{m}\sum_{i=1}^{k}(y_j - A_k(x_j))(A_k(x_j) - B_i(x_j)) = \frac{1}{m}\sum_{j=1}^{m}(y_j - A_k(x_j))^2 +$$

$$+\frac{1}{km}\sum_{j=1}^{m}\sum_{i=1}^{k}(A_k(x_j) - B_i(x_j))^2 - \frac{1}{m}\sum_{j=1}^{m}(y_j - A_k(x_j))\frac{(A_k(x_j) - B_k(x_j))}{k}.$$

Откуда следует, что средний квадрат ошибки для алгоритма A_k вычисляется по формуле

$$\frac{1}{m}\sum_{j=1}^{m}(y_j - A_k(x_j))^2 = \frac{1}{km}\sum_{j=1}^{m}\sum_{i=1}^{k}(y_j - B_i(x_j))^2 - \frac{1}{km}\sum_{j=1}^{m}\sum_{i=1}^{k}(A_k(x_j) - B_i(x_j))^2 +$$

$$+\frac{1}{m}\sum_{j=1}^{m}(y_j - A_k(x_j))\frac{(A_k(x_j) - B_k(x_j))}{k}.$$

Два слагаемых $L(S, A_k)$ в полученной формуле имеют простую интерпретацию. Первое описывает отклонение прогнозов от истинных значений Y, а второе представляет собой среднюю дисперсию прогнозов для объектов из S. Последнее слагаемое интерпретируется сложнее, но можно сделать несколько наблюдений. Во-первых, оно стремится к нулю при росте размера ансамбля k. Во-вторых, само его появление вызвано желанием упростить алгоритм и определить A_k как сумму слагаемых предыдущих шагов без включения искомого алгоритма, иначе это слагаемое строго равно нулю. Наконец, в дальнейшем метод потребует еще несколько нестрогих переходов и дополнительных эвристик, поэтому не будем включать последнее слагаемое в функционал:

$$L(S, A_k) = \frac{1}{km}\sum_{j=1}^{m}\sum_{i=1}^{k}(y_j - B_i(x_j))^2 - \frac{1}{km}\sum_{j=1}^{m}\sum_{i=1}^{k}(A_k(x_j) - B_i(x_j))^2. \tag{1.1}$$

Таким образом, оптимальность ансамбля может достигаться за счет двух факторов: хорошей аппроксимации связи Y с переменными $X_1, \ldots, X_n$ на обучающей выборке и высокой дисперсии прогнозов обучающих объектов.

Эти рассуждения позволяют выбрать функционал для оптимизации нового слагаемого ансамбля B_k, $B_k = \arg\min_B Q(B, A_k)$. Строгое вычисление прироста качества $L(S, A_k) - L(S, A_{k-1})$ согласно (1.1) приводит к значительному усложнению алгоритма:

$$L(S, A_k) - L(S, A_{k-1}) = \frac{1}{km}\sum_{j=1}^{m}\sum_{i=1}^{k}(y_j - B_i(x_j))^2 - \frac{1}{(k-1)m}\sum_{j=1}^{m}\sum_{i=1}^{k-1}(y_j - B_i(x_j))^2 -$$

$$-\frac{1}{km}\sum_{j=1}^{m}\sum_{i=1}^{k}(A_k(x_j) - B_i(x_j))^2 + \frac{1}{(k-1)m}\sum_{j=1}^{m}\sum_{i=1}^{k-1}(A_{k-1}(x_j) - B_i(x_j))^2 =$$

$$= \frac{1}{km}\sum_{j=1}^{m}(y_j - B_k(x_j))^2 + \frac{1}{k(k-1)m}\sum_{j=1}^{m}\sum_{i=1}^{k-1}(y_j - B_i(x_j))^2 -$$

$$-\frac{1}{km}\sum_{j=1}^{m}(A_k(x_j) - B_k(x_j))^2 - \frac{1}{k(k-1)m}\sum_{j=1}^{m}\sum_{i=1}^{k-1}(A_{k-1}(x_j) - B_i(x_j))^2.$$

Вклад одного эдемента в этот функционал убывает с ростом размера ансамбля, поэтому логичнее для оптимизации отдельного элемента использовать $k(L(S,A_k)-L(S,A_{k-1}))$, а эта разность, в свою очередь, с ростом k приближается к следующему виду:

$$Q(B,A_k)=\frac{1}{m}\sum_{j=1}^{m}\big(y_j-B(x_j)\big)^2-\frac{1}{km}\sum_{j=1}^{m}\sum_{i=1}^{k}\big(A_k(x_j)-B(x_j)\big)^2,$$

что уже значительно удобнее. Тем не менее, $Q(B,A_k)$ не достигает минимума по прогнозам $B(x_1)$, ..., $B(x_m)$ и требует введения параметра μ, где $\mu\in[0,1)$. Дальнейшие построения основываются на предположении, что оптимальность ансамбля в смысле обеспечения с его помощью максимальной обобщающей способности будет достигаться для корректирующих деревьев, при которых функционал $Q(B,A_k,\mu)$ минимален:

$$Q(B,A_k,\mu)=\frac{1}{m}\sum_{j=1}^{m}\big(y_j-B(x_j)\big)^2-\mu\frac{1}{km}\sum_{j=1}^{m}\sum_{i=1}^{k}\big(A_k(x_j)-B(x_j)\big)^2, \tag{1.2}$$

т.е. $B_k=\arg\min_B Q(B,A_k,\mu)$.

Далее функционал (1.2) позволяет применить процедуру градиентного бустинга напрямую для построения элементов ансамбля. Такой подход был реализован в [8] и показал хорошие результаты в практических задачах. Однако он не лишен недостатков. В первую очередь это возрастание сложности обучения. Кроме того, возникает необходимость подбирать шаг градиентного бустинга.

Этих недостатков лишено непосредственное построение дерева, оптимизирующего функционал (1.2). Аналогия с градиентным бустингом сохраняется, но B попадает в минимум функционала за один шаг, что снижает сложность и не требует подбора параметра. Однако такой способ связан с некоторыми техническими сложностями и будет целью последующих исследований. Для проверки самой концепции рассмотрим другой подход, реализуемый с помощью стандартных методов scikit-learn [9]. Предположим, что минимум Q достигается в точке $B^*(x_1),\ldots,B^*(x_m)$. В качестве B_k может быть использовано регрессионное дерево T_k, обучаемое по выборке $\{(B^*(x_1),x_1),\ldots,(B^*(x_m),x_m)\}$. По сложности и количеству параметров этот подход аналогичен предыдущему, но в данной работе мы пожертвуем простотой для достижения дополнительного разнообразия в ансамбле с помощью дополнительных техник, таких, как бэггинг и метод случайных подпространств. Опишем данный подход подробнее.

Предположим, что за первые $k-1$ шагов получен ансамбль $B_1,\ldots,B_{k-1}$. Сгенерируем бутстрэп репликацию S_k^b выборки S в проекции на случайное подпространство, как это описано во Введении. По репликации S_k^b построим регрессионное дерево T_k. Алгоритм B_k ищется в виде суммы $B_k=T_k+t_k$, где t_k – корректирующее дерево, которое строится исходя из условия минимизации $Q(T_k+t_k)$.

Перепишем функционал (1.2) для этой процедуры:

$$Q(t,\mu)=\frac{1}{m}\sum_{j=1}^{m}\big(y_j-T_k(x_j)-t(x_j)\big)^2-\mu\frac{1}{km}\sum_{j=1}^{m}\sum_{i=1}^{k}\big(A_k(x_j)-T_k(x_j)-t(x_j)\big)^2. \tag{1.3}$$

На первом шаге ищется действительный вектор $t^*=t_1^*,\ldots,t_m^*$, компонентами которого являются оптимальные смещения прогнозов, вычисляемых деревом T_k, т.е. $t^*=\arg\min Q(t,\mu)$. Минимальное значение функционала $Q(t,\mu)$ по t_j достигается при $\partial Q(t,\mu)/\partial t(x_j)=0$ или при

$$t(x_j)=t_j^*=\frac{k}{k-\mu}y_j-T_k(x_j)-\frac{\mu}{k-\mu}A_k(x_j),\quad j=\overline{1,m}. \tag{1.4}$$

Вместе с тем вычисление оптимальных смещений по формуле (1.4) может приводить к снижению точности формируемого алгоритма B_k по отношению к T_k при соблюдении наборов неравенств:

$$B_k(x_k)=T_k(x_j)+t_k(x_j)<T_k(x_j)<y_j \tag{1.5}$$

или

$$B_k(x_k) = T_k(x_j) + t_k(x_j) > T_k(x_j) > y_j. \quad (1.6)$$

Такого снижения точности можно избежать, если при выполнении одного из наборов неравенств (1.5), (1.6) приравнивать t_j^* нулю.

Таким образом, описана рекурсивная процедура построения ансамбля алгоритмов. Начиная с пустого ансамбля и тождественного нуля в качестве коллективного решения, затем пополним его суммами пар деревьев, где первое аппроксимирует целевую переменную напрямую, а второе аппроксимирует поправку, минимизирующую функционал $Q(t,\mu)$ (1.3). Построение ансамбля завершается, если k достигает задаваемого пользователем порогового значения N. Ансамбли, которые строятся с помощью представленной выше процедуры, далее будем называть декоррелированными.

2. Коллективное решение и дополнительные эвристики. Наряду со способом генерации ансамбля важную роль для достижения высокой эффективности ансамблевого алгоритма играет процедура, вычисляющая коллективное решение. Хотя ансамбль строится, исходя из соображений минимизации ошибки усредненного ответа алгоритмов, использование такой схемы коллективного решения оказалось недостаточно эффективно. В работе [8] рассматривались несколько других способов, кроме среднего по декоррелированному ансамблю, в частности стекинг [10], т.е. применение прогнозов, вычисляемых отдельными деревьями ансамбля, в качестве признаков для алгоритмов второго уровня, рассчитывающих выходное коллективное решение. Эксперименты, представленные в [8], показывают, что более высокая точность достигается с помощью стекинга со случайным регрессионным лесом, чем с простым усреднением. По этой причине в методе реализованы несколько вариантов коллективного решения: усреднение и стекинг с методом градиентного бустинга и случайным лесом, а также их выпуклой комбинацией.

Наибольшая точность предсказания обычно достигается при больших размерах ансамбля. Однако чрезмерно большое число признаков приводит к увеличению неустойчивости и снижению точности прогноза. Для снижения признакового пространства могут быть использованы различные подходы. В предлагаемом методе применяются две техники: прореживание и переоптимизация. В первом случае после вычисления полного декоррелированного ансамбля все вошедшие в него регрессионные деревья ранжируются по величине функционала $Q(B,\mu)$, где для произвольного элемента B_j, ансамбль полагается равным $\{B_i \in A_k \,|\, i \neq j\}$. Далее из ансамбля исключаются все недостаточно эффективные слагаемые так, чтобы оставить заданные заранее N' элементов. Эксперименты на реальных задачах не выявили заметного повышения эффективности от использоваия этой процедуры. Возможно, это не так при большем количестве исходных слагаемых, но в таком случае значительно увеличивается вычислительная сложность процедуры. Переоптимизация же оставляет количество слагаемых, но пытается дополнительно “раздвинуть” их. Рассмотрим ансамбль $\{B_1,\ldots,B_N\}\backslash B_j$ и применим процедуру построения нового алгоритма B к этому ансамблю, а затем заменим B_j на полученный алгоритм, и так для всех слагаемых по очереди. Эффект от такой процедуры также не стал грандиозным, однако эти две процедуры оставлены в методе для дальнейшего исследования.

Несколько более высокую эффективность показало применение для снижения размерности признакового пространства варианта метода экстремальной группировки (ЭГ) параметров [11]. На начальном этапе признаки случайно разбиваются на некоторое заранее заданное число групп, для каждой из которых вычисляется соответствующий групповой фактор, как среднее по всем признакам, вошедшим в группу. При этом часть из признаков умножается на -1 для обеспечения одинаковой направленности связей с целевой переменной. Используется процедура, сходная со стандартным методом кластеризации “k-средних” и заключающаяся в переносе каждого из признаков в группу, для которой модуль коэффициента корреляции между признаком и соответствующим групповым фактором максимален. Далее производится пересчет групповых факторов. Процесс завершается в случае отсутствия необходимости переноса признаков на каком-то из шагов. В реализованном методе ЭГ может быть использован как для исходных признаков, что особенно актуально, например, для химических задач с высокой коррелированностью признаков, так и для генерируемого пространства.

В работе не используются стекинг и предварительная кластеризация признаков, поскольку это некие общие процедуры, применимые к любому методу.

Таким образом, свойства итогового алгоритма определяются следующим набором параметров: N – число элементов ансамбля (в экспериментах фиксировалось $N = 200$); N' – число эле-

ментов ансамбля после отсеивания (в экспериментах фиксировалось $N' = N$); μ – влияние декоррелирующей компоненты функционала (в экспериментах фиксировалось $\mu = 0.5$); ν – доля признаков, задействованных для построения T_k; κ – параметр вклада t_k: $B_k = T_k + \kappa t_k$ (в экспериментах фиксировалось $\kappa = 1$); G – число групп ЭГ в ансамбле; corrector=average|forest|boosting – тип корректирующей процедуры. Отдельно задаются параметры итоговой корректирующей процедуры. Для бустинга и бэггинга берутся параметры по умолчанию, для усреднения параметры не нужны.

Как это обычно бывает, не существует единого набора гиперпараметров, эффективного для любых прикладных задач. Пример комбинаций, позволяющих достичь преимущества по сравнению с чистыми методами градиентного бустинга и случайного леса, для предсказания свойств химических элементов представлен в [12]. В следующей главе описано применение разработанного метода для решения нескольких других задач.

3. Прикладные задачи. Для оценки качества исследованного алгоритма и диапазона оптимальных гиперпараметров были использованы следующие задачи.

Houses – выборка 'House Prices: Advanced Regression Techniques'[2], содержащая данные для оценки стоимости недвижимости по различным признакам, от рельефа участка и длины примыкающей улицы до типа и качества отделки постройки и близости железной дороги. В задаче имеется 1460 объектов, описываемых 79 признаками, большинство из которых категориальные. Для применения метода к таким признакам применялась следующая процедура кодирования: по обучающей подвыборке значения признаков заменялись на среднее целевого показателя для всех объектов с таким значением признака; в тестовой подвыборке использовалось значение из обучающей подвыборки; если в тестовую подвыборку попадало значение, неизвестное в обучающей, оно заменялось на среднее значение целевого признака по всей выборке; пропуски в количественных признаках заменялись нулями, поскольку их природа – площадь отсутствующего элемента.

Prices, Costs – еще одна выборка неджвижимости[3] из репозитория UCI [13]. Выборка содержит 372 объекта, описанных 107 числовыми признаками, по которым предсказывается цена недвижимости и стоимость ее постройки. Эти две задачи рассматривались отдельно.

Systolic – задача оценивания систолического давления по сигналу электрокардиограммы (ЭКГ) [14]. Выборка содержит 836 объектов, описанных 160 спектральными показателями.

Разработанный метод реализован с помощью стандартных методов из библиотеки scikit-learn [9], новизна сводится к использованию модифицированной целевой переменной. Дерево T_k ищется с помощью BaggingRegressor с числом элементов, равным одному, и $max\_features = \nu$. Поправка t_k реализована с помощью DecisionTreeRegressor. Для экспериментов часть гиперпараметров фиксировалась и делалось несколько запусков с различными значениями μ, ν, G и корректора. В этом порядке они и приводятся при описании результатов.

Также из scikit-learn были взяты референсные методы: BoostingRegressor в качестве реализации градиентного бустинга и RandomForestRegressor – как случайный лес. Все методы запускались с размером ансамбля, равным 200, и остальными параметрами по умолчанию. Мы сознательно отказались от попыток сравнить качество с другими исследователями и провели собственные тесты по нескольким соображениям, главное из которых – поставить методы в относительно одинаковые условия и выявить эффект, оказываемый техникой декорреляции. Так, задача Houses требует предобработки признаков, и улучшение результатов достигается лидерами соревнования Kaggle, в том числе за счет улучшения этой обработки. Это же во многом касается и задач Costs и Prices – наиболее свежее доступное исследование этих данных также во многом сосредоточено на специфике выборки [15], в то время как мы изучаем общий подход. Задача Systolic в таком виде никем не исследовалась.

Полученные результаты и параметры запуска представлены в табл. 1. Для оценки качества предсказания использовались две характеристики: коэффициент детерминации (R-квадрат, R^2) и среднее абсолютное отклонение (MAE). Решение задач производилось в режиме скользящего контроля с разбиением на 10 частей. Кроме того, результаты усреднялись по 10 запускам, чтобы отразить их устойчивость, поэтому в таблице приводится среднее значение (mean) и стандартное отклонение (stdev) показателей.

[2] https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data.
[3] https://archive.ics.uci.edu/ml/datasets/Residential+Building+Data+Set.

Таблица 1. Результаты экспериментов

Задача	Модель	R^2		MAE	
		mean	stdev	mean	stdev
Houses	Boosting	0.882	0.006	17642	352.37
	Forest	0.861	0.0051	17923	112.72
	0.5, 0.4, 40, forest	0.881	0.0134	16645	211.1
	0.5, 0.4, 40, boosting	0.879	0.01	17232	239.12
	0.5, 0.4, 40, average	**0.888**	**0.0035**	**16577**	**94.9**
Costs	Voosting	0.96	0.0027	**16.6**	0.342
	Forest	0.95	0.0035	20.19	0.372
	0.7, 40, boosting	0.946	0.0056	20.59	0.659
	0.7, 40, forest	0.932	0.0042	19.41	0.662
	0.7, 40, average	0.933	0.0057	19.35	0.494
	1, 40, forest	0.96	0.0036	18.21	0.558
	1, 80, forest	**0.961**	**0.0015**	17.09	**0.26**
	1, 120, forest	0.96	0.0032	18.07	0.409
Prices	Boosting	**0.977**	**0.0017**	**81.76**	**3.014**
	Forest	0.961	0.003	121.21	3.073
	0.7, 40, boosting	0.929	0.0063	202.99	7.278
	0.7, 40, forest	0.941	0.0056	184.44	9.078
	0.7, 40, average	0.938	0.0064	191.83	8.133
	1, 40, forest	0.971	0.0027	103.53	3.15
	1, 80, forest	0.972	0.0021	102.17	3.41
	1, 120, forest	0.97	0.0033	104.39	3.5
Systolic	Boosting	0.432	0.0134	6.02	0.055
	Forest	0.464	0.0073	5.81	0.043
	0.3, 80, boosting	0.475	0.0077	5.75	0.044
	0.3, 80, forest	0.482	0.0084	5.79	0.05
	0.3, 80, average	**0.486**	**0.0064**	**5.75**	**0.033**

Первое, на что стоит обратить внимание, это разные соотношения качества самих референсных методов. Видно, что в задачах с недвижимостью бустинг показывает значительно лучшее качество, чем случайный лес. В то же время в задаче оценивания давления бустинг значительно отстает. В свою очередь предлагаемый метод всегда демонстрирует превосходство над аутсайдером. В задачах Prices и Costs при этом удалось только сравняться с лидером. В задаче Houses лидерство уже более явное – предлагаемый метод лидирует как по R^2, так и по средней абсолютной ошибке. Наилучший же результат метод продемонстрировал в задаче оценивания систолического давления, где лидера пары референсных методов – на этот раз случайный лес – удалось значительно обойти по обоим критериям.

Что касается выбора гиперпараметров, то в очередной раз подтвердилась их неуниверсальность. Где-то преимущество дали высокие ν, где-то – наоборот. В задачах с недвижимостью хорошим корректором оказался случайный лес, а в прогнозе давления – простое усреднение.

Заключение. Представлен новый ансамблевый регрессионный метод. Дается дополнительное обоснование подхода, уточняющее вывод, который приведен в предыдущих работах. Также рассмотрены результаты тестирования метода, оценивающие возможность его использования при решении прикладных задач в бизнесе и медицине. Результаты экспериментов подтверждают перспективность предлагаемого подхода.

Дальнейшие исследования предполагается направить на повышение быстродействия метода, что возможно достигнуть, во-первых, за счет непосредственного построения элементов B_k как

деревьев со специальным функционалом качества, а во-вторых, за счет оптимизации построения этих деревьев, в том числе с помощью библиотеки cython.

СПИСОК ЛИТЕРАТУРЫ

1. Положение о ЦКП “Информатика” // [Электронный ресурс]. Режим доступа http://www.frccsc.ru/ckp (дата обращения 14.02.2023).
2. *Zhou Z.H.* Ensemble Methods: Foundations and Algorithms. N.Y.: Chapman and Hall/CRC, 2012.
3. *Hastie T., Tibshirani R., Friedman J.* The Elements of Statistical Learning Data Mining, Inference, and Prediction. Springer Series in Statistics. N.Y.: Springer, 2009.
4. *Breiman L.* Random forests // Machine Learning. 2001. V. 45. № 1. P. 5–32.
5. *Schapire R.E., Freund Y.* Foundations and Algorithms. Cambridge, Massachusetts, London: MIT Press, 2012.
6. *Ho T.K.* The Random Subspace Method for Constructing Decision Forests // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1998. V. 20. № 8. P. 832–844.
7. *Garcia-Pedrajas N., Ortiz-Boyer D.* Boosting Random Subspace Method // Neural Networks. 2008. V. 21. № 9. P. 1344–1362.
8. *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 // Doklady Mathematics. 2021. V. 103. P. 1–4.
9. *Pedregosa F., Varoquaux G., Gramfort A. et al.* Scikit-learn: Machine Learning in Python // Machine Learning Research. 2011. V. 12. P. 2825–2830.
10. *Wolpert D.H.* Stacked Generalization // Neural Networks. 1992. V. 5. № 2. P. 241–259.
11. *Braverman E.M., Muchnik I.B.* Structural Methods for Processing Empirical Data. M.: Nauka, 1983.
12. *Senko O.V., Dokukin A.A., Kiselyova N.N., Dudarev V.A., Kuznetsova Yu.O.* New Two-Level Ensemble Method and Its Application to Chemical Compounds Properties Prediction // Lobachevskii Journal of Mathematics. 2023. V. 44. № 1. P. 188–197.
13. *Rafiei M.H., Adeli H.* A Novel Machine Learning Model for Estimation of Sale Prices of Real Estate Units // J. Construction Engineering & Management. 2015. V. 142. № 2.
14. *Сенько О.В., Чучупал В.Я., Докукин А.А.* Неинвазивное оценивание уровня артериального давления с помощью кардиомонитора CardioQvark // Математическая биология и биоинформатика. 2017. Т. 2. № 12. С. 536–546.
15. *Mostofi F., Toğan V., Başağa H.B.* Real-estate Price Prediction with Deep Neuralnetwork and Principal Component Analysis // Organization, Technology and Management in Construction. 2022. V. 14. № 1. P. 2741–2759.