Отчет о практическом задании «Ансамбли алгоритмов. Веб-сервер. Композиции алгоритмов для решения задачи регрессии».

Практикум 317 группы, ММП ВМК МГУ.

Алексеев Илья Алексеевич.

Декабрь 2022.

Содержание

1 Введение.

В рамках данного практического задания были реализованы методы линейного ансамблирования деревьев: Random Forest и Gradient Boosting. Изучена зависимость точности предсказаний и время обучения в зависимости от размера подпространства в методе случайных подпространств, ограничений на глубину деревьев, числа деревьев и величины шага спуска. Реализованные алгоритмы обёрнуты в web-приложение, которое загружено на dockerhub.

1.1 Random Forest.

Алгоритм *Random Forest* [1] в рамках задачи регрессии он состоит в том, чтобы обучить N деревьев и посчитать среднее арифметическое их предсказаний.

Пусть $b_n : \mathbb{R}^d \to \mathbb{R},\ n = \overline{1, N}$ – решающие деревья для задачи регрессии. Тогда Random Forest делает предсказание как

$$a_N(x) = \frac{1}{N} \sum_{i=1}^{N} b_n(x).$$

Для алгоритма $a_N(x)$ variance-компонента в разложении на смещение и разброс записывается следующим образом:

$$\mathrm{Var}[a_N] = \frac{1}{N} \mathrm{Var}[b_i] + \frac{N(N-1)}{N^2} \mathrm{Cov}(b_i, b_j). \tag{1}$$

То есть разброс Random Forest состоит из разброса одного дерева, поделённого на их число, плюс ковариация всех деревьев. Это значит, что если одновременно снижать коррелированность (зависимость, похожесть) деревьев и увеличивать их число, разброс ансамбля будет минимизироваться.

Корреляцию между деревьями можно уменьшить, например, двумя способами: 1) обучать каждое дерево не на всём датасете, а на подвыборке, сформированной с помощью взятия с возвращением (*bootstrap*); 2) обучать каждое дерево не по всем признакам, а по случайно выбранному подмножеству (*метод случайных подпространств*).

1.2 Gradient Boosting.

Алгоритм *Gradient Boosting* следующий [2]. Первый базовый алгоритм обучается предсказывать целевую переменную $y \in \mathbb{R}^\ell$, где ℓ – размер выборки. Второй базовый алгоритм обучается на ошибку первого алгоритма, т.е. вектор $s^{(1)} \in \mathbb{R}^\ell$ с компонентами $s_i^{(1)} = y_i - b_1(x_i)$. $(n+1)$-ый базовый алгоритм обучается предсказывать $s_i^{(n)} \in \mathbb{R}^\ell$ с компонентами $s_i^{(n)} = y_i - a_n(x_i)$, где a_n – ансамбль из n алгоритмов. Его предсказание строится следующим образом:

$$a_n(x) = a_{n-1} + \alpha w_n b_n(x),$$

где α — гиперпараметр метода, w_n – вес. Веса подбираются как решение оптимизационной задачи:

$$w_n = \arg\min_w \sum_{i=1}^{\ell} L(y_i, a_{n-1} + w b_n(x_i))^2.$$

Величина $s^{(n)}$ имеет смысл направления убывания ошибки предсказания, поэтому когда мы обучаем b_n, мы ищем аппроксимацию антиградиента, а когда добавляем обученный b_n в ансамбль, мы делаем шаг градиентного спуска в пространстве предсказаний $\mathbb{R}^\ell$.

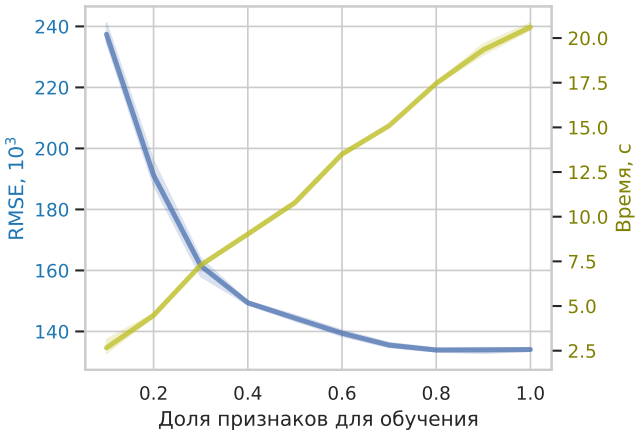
2 Эксперименты.

Из датасета *House Sales in King County, USA* убран признак `id`, принимающий значения типа `7129300520` или `2487200875`. Признак `date` формата `%Y%m%dT000000` заменён на три признака `year`, `month`, `day`. Затем признак `year` заменён на `year_2014` и `year_2015` с помощью OneHot кодирования. Признаки `month` и `day` преобразованы в `month_sin`, `month_cos`, `day_sin`, `day_cos` как циклические признаки. Весь датасет случайным образом поделён на три выборки: обучающую, валидационную и тестовую (в соотношении 3:1:1).

2.1 Размер случайного подпространства.

В данном эксперименте были рассмотрены значения параметра `subspace_size` (доля признаков, используемых для обучения отдельного дерева) в отрезке $[0.1, 1]$ с шагом 0.1. Для каждого значения были обучены модели Random Forest и Gradient Boosting и замерена ошибка RMSE на валидации. Эксперимент повторён пять раз, значения ошибки усреднены (рис. 1). Использовались параметры из табл. 1.

(a) Random Forest.



(b) Gradient Boosting.

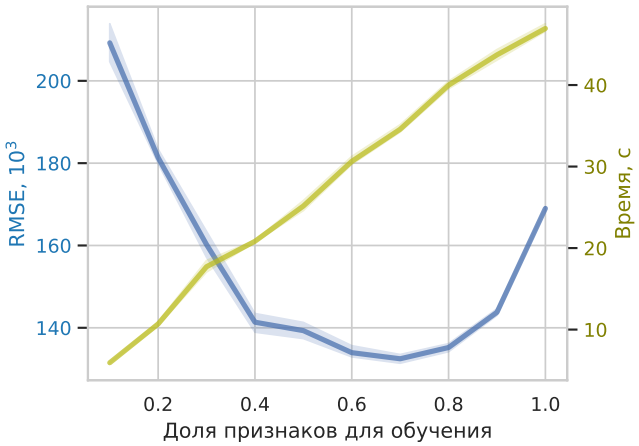


Рис. 1: Значения RMSE и время обучения в зависимости от доли признаков, выбираемых для обучения деревьев в ансамблях деревьев.

На рис. 1a ошибка предсказания уменьшается с ростом параметра вплоть до значения параметра 0.8, затем выходит на плато. Это можно интерпретировать следующим образом: 1) малое число признаков не дает достаточное количество информации для точного предсказания; 2) полный набор признаков не улучшает качество предсказания, так как делает деревья коррелированными и вносит вклад во второе слагаемое в формуле (1).

Параметр	RF	GB
`n_estimators`	200	200
`max_depth`	50	50
`learning_rate`	—	0.05

Таблица 1: Параметры эксперимента 2.1.

На рис. 1b ошибка предсказания уменьшается с ростом параметра вплоть до значения 0.7, а затем в отличие от Random Forest резко растет. Значит, для аппроксимации антиградиента излишняя точность приводит к ухудшению бустинга.

Время обучения для обеих моделей имеет одинаковую зависимость: оно прямо пропорционально параметру, что объясняется тем, что увеличение признакового пространства увеличивает время поиска оптимального признака, по которому производится деление в узлах дерева.

2.2 Максимальная глубина дерева.

В данном эксперименте были рассмотрены значения параметра `max_depth` (ограничение на максимальную глубину дерева в ансамбле). Для модели Random Forest рассмотрены значения в отрезке $[10, 190]$ с шагом 20. Для модели Gradient Boosting – значения $[1, 10]$ и $20, 30$. Для каждого значения замерена ошибка RMSE на валидации. Эксперимент повторён пять раз, значения ошибки усреднены (рис. 2).

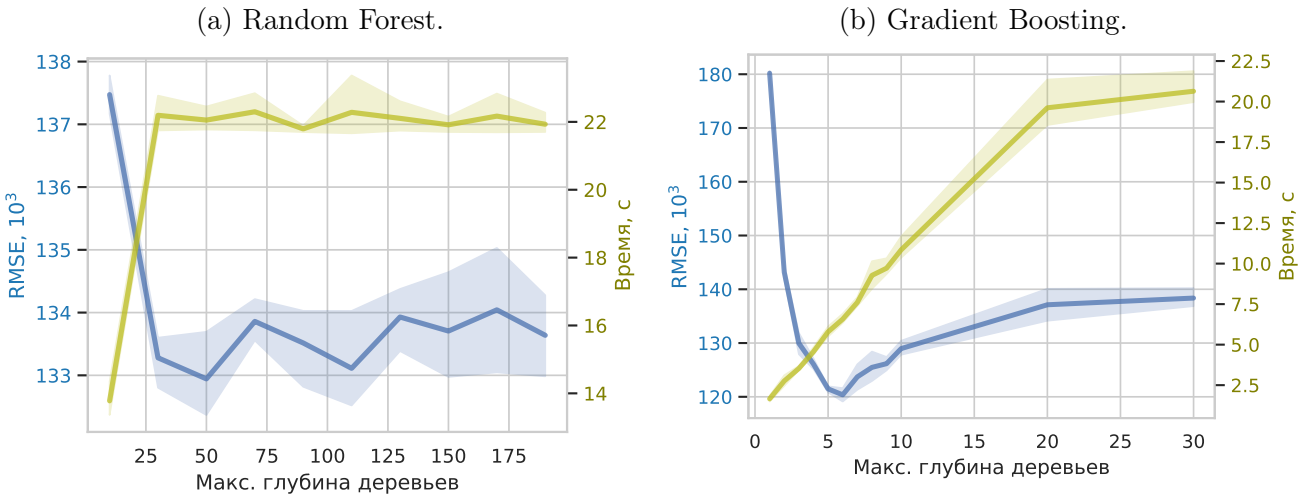


Рис. 2: Значения RMSE и время обучения в зависимости от ограничения на глубину дерева в ансамблях деревьев.

На рис. 2a ошибка предсказания уменьшается при изменении глубины с 10 до 30, затем выходит на плато. Заметим, что отдельные деревья в большинстве не используют всю предоставленную глубину, т.к. время обучения не меняется с ростом параметра.

Параметр	RF	GB
`n_estimators`	200	200
`subspace_size`	0.8	0.7
`learning_rate`	—	0.05

Таблица 2: Параметры эксперимента 2.2.

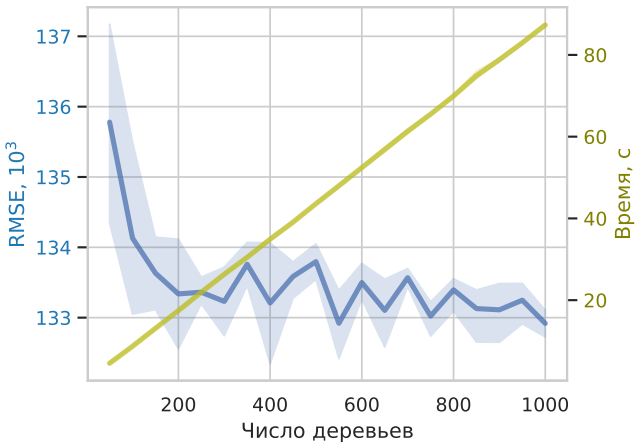
Полупрозрачная область около линии – это 95-доверительный интервал, он характеризует разброс результатов пяти экспериментов. Визуально может показаться, что разброс велик, но на самом деле он составляет не более $\pm 10^3$, что в сравнении с ошибкой RMSE порядка 10^5 незначительно. Это подтверждает, что отдельные деревья не строятся во всю глубину.

На рис. 2b ошибка предсказания уменьшается при изменении глубины с 1 до 6, затем растёт и выходит на плато. Увеличение времени обучения говорит о том, что отдельные деревья строятся в полную глубину. Значит, 1) существует оптимальная глубина; 2) выводы, аналогичные Random Forest.

2.3 Число деревьев.

В данном эксперименте были рассмотрены значения параметра `n_estimators` (число деревьев в ансамбле) в отрезке $[50, 1000]$ с шагом 50. Для каждого значения были обучены модели Random Forest и Gradient Boosting и замерена ошибка RMSE на валидации. Эксперимент повторён четыре раза, значения ошибки усреднены (рис. 3). Использовались параметры из табл. 3.

(a) Random Forest.



(b) Gradient Boosting.

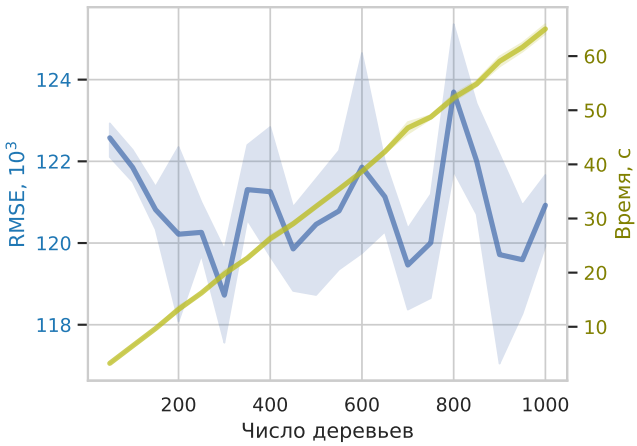


Рис. 3: Значения RMSE и время обучения в зависимости от числа деревьев в ансамбле.

На рис. 3a ошибка предсказания уменьшается при изменении числа деревьев с 50 до 200, затем выходит на плато (отклоняется от тренда максимум на $\pm 10^3$). Учитывая, что порядок изменяется на порядок (до тысячи), а ошибка предсказаний не увеличивается, можем сделать вывод, что число деревьев не приводит к переобучению.

Параметр	RF	GB
`max_depth`	50	6
`subspace_size`	0.8	0.7
`learning_rate`	—	0.05

Таблица 3: Параметры эксперимента 2.3.

На рис. 3b ошибка предсказания осциллирует около одного и того же значения, изменяясь незначительно. В сравнении с 3a значение ошибки меньше примерно на 10%, но разброс в два раза больше. Время обучения для обеих моделей имеет одинаковую зависимость: оно прямо пропорционально параметру, что объясняется тем, что увеличение числа деревьев увеличивает суммарное время обучения модели.

2.4 Размер шага в Gradient Boosting.

В данном эксперименте были рассмотрены значения параметра `learning_rate` (размер шага градиентного бустинга) на отрезке $[10^{-3}, 10^0]$, восемь значений из логарифмической сетки. Параметры брались из табл. 1. Для каждого значения была обучена модель Gradient Boosting и замерена ошибка RMSE на валидации. Эксперимент повторён пять раз, значения ошибки усреднены (рис. 4). Ошибка предсказания уменьшается при изменении параметра с 10^{-3} до $\approx 5 \cdot 10^{-}2$, затем выходит на плато и растет.

Поскольку Gradient Boosting является градиентными спуском в пространстве предсказаний, такое поведение ошибки объяснимо тем, что 1) малый шаг не позволяет спуску

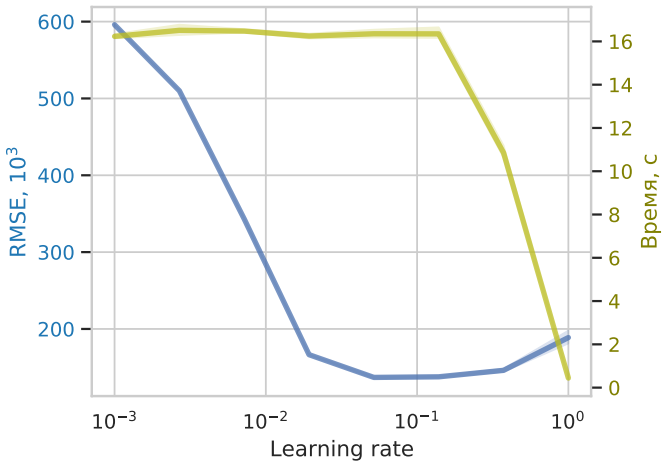


Рис. 4: Значения RMSE и время обучения модели Gradient Boosting в зависимости от шага.

добраться до окрестности оптимума за отведённое число итераций; 2) большой шаг не позволяет спуску сойтись.

При больших значениях шага также резко снижается время обучения. Это объясняется тем, что окрестность оптимума успешно достигается самыми первыми деревьями и под большинство объектов ансамбль уже «настроился». Остаётся часть объектов, предсказания для которых отличаются значительно. Их достаточно много, чтобы ошибка RMSE была высокой, и одновременно их достаточно мало, чтобы деревья могли делить их и уменьшать энтропию.

3 Выводы.

Линейные методы ансамблирования Random Forest и Gradient Boosting чувствительны к выбору параметров базовых алгоритмов. Чтобы избежать их коррелированности и излишней точности нужно подбирать размер случайного подпространства (раздел 2.1). Метод Gradient Boosting требует строгое ограничение на высоту деревьев, в то время как Random Forest просто не использует доступную высоту (раздел 2.2). Оба метода не деградируют при увеличении числа базовых алгоритмов (раздел 2.3). Размер шага существенно влияет на сходимость Gradient Boosting (раздел 2.4).

Параметр	RF	GB
`n_estimators`	200	200
`max_depth`	50	6
`subspace_size`	0.8	0.7
`learning_rate`	—	0.05

Таблица 4: Оптимальные параметры.

Оптимальные параметры к датасету *House Sales in King County, USA* приведены в табл. 4.

Список литературы

1. *Воронцов К. В.* Линейные ансамбли. — URL: http://www.machinelearning.ru/wiki/images/3/3a/Voron-ML-Compositions1-slides.pdf. — (Дата обращения: 15.12.2022).
2. *Воронцов К. В.* Продвинутые методы ансамблирования. — URL: http://www.machinelearning.ru/wiki/images/2/21/Voron-ML-Compositions-slides2.pdf. — (Дата обращения: 15.12.2022).