Машинное обучение

Консультация

Ковалев Евгений

НИУ ВШЭ, 2020

Вопросы

https://vk.cc/aAvU23

Вопрос №1

Какие изначальные характеристики задачи говорят, что в ней лучше использовать случайный лес, а какие - что бустинг? Если можно, с примерами задач

Ответ на вопрос №1

В целом бустинг – очень популярная сейчас модель

Есть много разных техник и даже специальных библиотек: XGBoost, LightGBM, CatBoost

Вы можете тестировать разные методы (и случайный лес, и бустинг) вне зависимости от данных

По умолчанию можете использовать бустинг, но есть нюансы

Ответ на вопрос №1

1. Случайный лес уменьшает разброс, а бустинг – смещение

- Значит, в задаче, где много шума, может лучше сработать случайный лес
- Примеры: предсказание поведения пользователей, определение мошеннических операций
- Оказаться шумной может даже валидация
- Градиентный бустинг может переобучиться из-за функций потерь, плохо работающих с выбросами (пример: MSE)

Ответ на вопрос №1

2. Случайный лес имеет тенденцию не переобучаться с ростом числа деревьев

- Конечно, случайный лес тоже может переобучиться, но ему гораздо сложнее сделать это, нежели градиентному бустингу
- Соответственно, в задачах, где очень высок риск переобучения, случайный лес может быть лучше (например, когда мало данных)

Ответ на вопрос №1

3. Случайный лес имеет меньше важных гиперпараметров

- Поэтому чтобы получить оптимальное качество для градиентного бустинга, нужно будет кропотливо подбирать гиперпараметры, в отличие от случайного леса

- Зачастую это дает положительные результаты, и бустинг оказывается лучше (но не всегда)

Ответ на вопрос №1

4. Случайный лес легче параллелизовать

- Потому что деревья обучаются независимо друг от друга

Ответ на вопрос №1

Ссылки:

- https://www.quora.com/When-would-one-use-Random-Forests-over-Gradient-Boosted-Machines-GBMs
- https://mlcourse.ai/articles/topic5-part2-rf/
- https://mlcourse.ai/articles/topic10-boosting/

Вопрос №2

Бустинг, улучшенное обучение: подбор оптимального значения в листьях с точки зрения исходной функции потерь - не очень понятна идея

Ответ на вопрос №2

«Когда дерево построено, можно подобрать оптимальные значения в листьях с точки зрения исходной функции потерь»

Ответ на вопрос №2

Решающее дерево разбивает пространство на непересекающиеся области, и в каждой из областей ответ равен константе.

Пусть у нас на шаге n построено дерево b_n с J_n листьев, есть области $R_1, \dots, R_{J_n}$ с ответами $b_{n1}, \dots, b_{nJ_n}$ соответственно. Тогда прогноз решающего дерева можно представить следующим образом:

$$b_n(x) = \sum_{j=1}^{J_n} b_{nj} \times I[x \in R_j]$$

Ответ на вопрос №2

$$b_n(x) = \sum_{j=1}^{J_n} b_{nj} \times I[x \in R_j]$$

Запишем обновление композиции:

$$a_N(x) = a_{N-1}(x) + \gamma_N \sum_{j=1}^{J_N} b_{Nj} \times I[x \in R_j]$$

Ответ на вопрос №2

$$a_N(x) = a_{N-1}(x) + \gamma_N \sum_{j=1}^{J_N} b_{Nj} \times I[x \in R_j]$$

Шаг обучения γ_N - это константа, так что мы можем перенести ее в сумму:

$$a_N(x) = a_{N-1}(x) + \sum_{j=1}^{J_N} \gamma_N b_{Nj} \times I[x \in R_j]$$

Ответ на вопрос №2

$$a_N(x) = a_{N-1}(x) + \sum_{j=1}^{J_N} \gamma_N b_{Nj} \times I[x \in R_j]$$

Заметим, что это выражение равносильно тому, как если бы у нас было не J_N листьев, а J_N базовых алгоритмов – предикатов $I[x \in R_j]$. Вспомним, что b_{Nj} - это предсказание в j-ом листе. Давайте вместо него подберем ответ j-ого константного базового алгоритма - γ_{Nj}.

Ответ на вопрос №2

Исходная задача оптимизации на шаге N:

$$\sum_{i=1}^{\ell} L\big(y_i, a_{N-1}(x_i) + \gamma_N b_N(x_i)\big) \to \min_{b_N, \gamma_N}$$

Новая задача оптимизации на шаге N:

$$\sum_{i=1}^{\ell} L\left(y_i, a_{N-1}(x_i) + \sum_{j=1}^{J_N} \gamma_{Nj} \times I\big[x \in R_j\big]\right) \to \min_{\{\gamma_{Nj}\}_{j=1}^{J_N}}$$

Ответ на вопрос №2

Области R_j не пересекаются, так что в итоге мы имеем J_N независимых подзадач:

$$\gamma_{Nj} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_j} L(y_i, a_{N-1}(x_i) + \gamma), \qquad j = 1, \dots, J_N$$

Ответ на вопрос №2

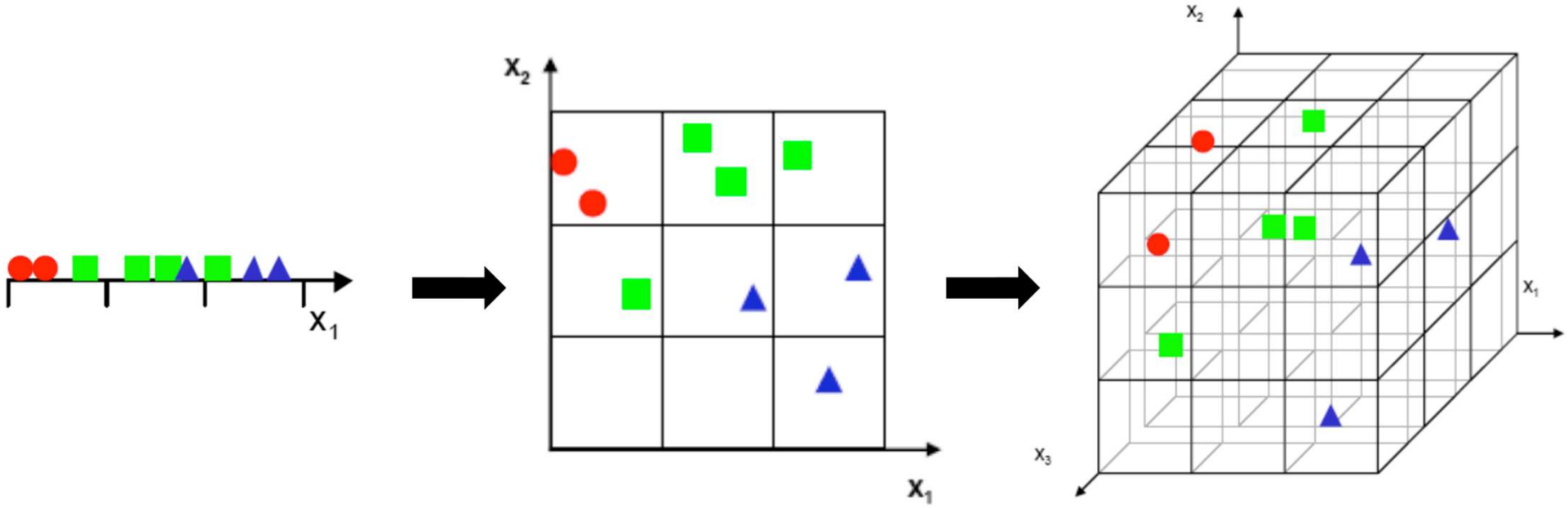
Ссылка:

- https://github.com/esokolov/ml-course-hse/blob/master/2019-fall/lecture-notes/lecture09-ensembles.pdf

Вопрос №3

В чем конкретно проблема того, что объекты мало похожи в проклятии размерности? Есть ли математическое обоснование (на уровне понятий), из которого следует, что мы в этом случае получим плохие результаты? И по какому параметру они будут плохи?

Ответ на вопрос №3



Ответ на вопрос №3

Рассмотрим единичный интервал [0,1]. 100 равномерно разбросанных точек будет достаточно, чтобы покрыть этот интервал с частотой не менее 0,01.

Теперь рассмотрим 10-мерный куб. Для достижения той же степени покрытия потребуется уже 10^{20} точек. То есть, по сравнению с одномерным пространством, требуется в 10^{18} раз больше точек.

http://www.machinelearning.ru/wiki/index.php?title=Проклятие_размерности

Ответ на вопрос №3

Проблема мультиколлинеарности

Линейная регрессия:

$$y = w_1 x_1 + w_2 x_2 + w_3 x_3$$

Пусть $x_1 = x_2 + x_3$

Тогда заменим $w_1 = w_1 + z$, $w_2 = w_2 - z$, $w_3 = w_3 - z$ получим:

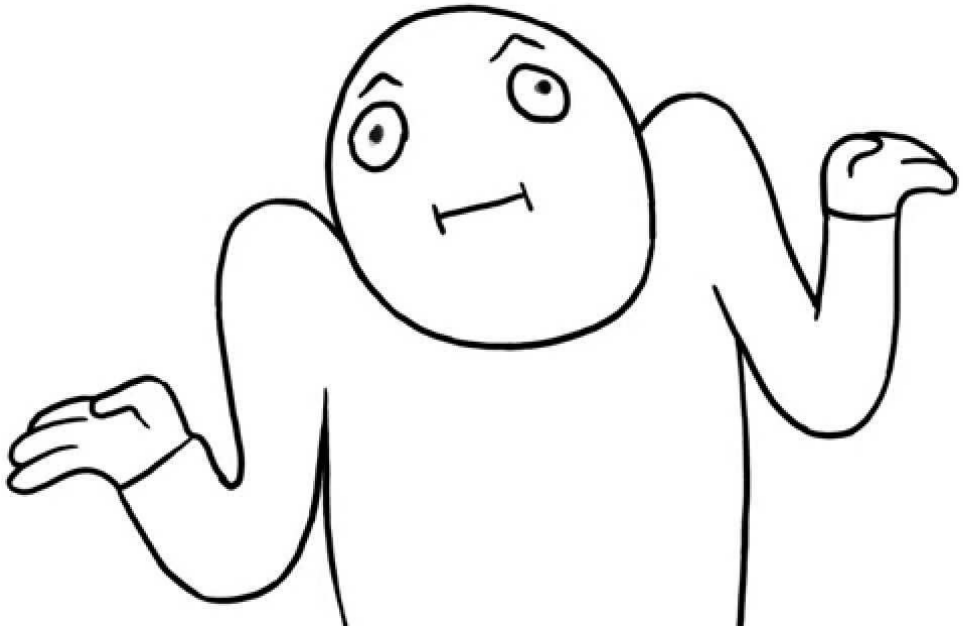
$$y = (w_1 + z)x_1 + (w_2 - z)x_2 + (w_3 - z)x_3 = w_1 x_1 + w_2 x_2 + w_3 x_3$$

То есть значения оптимальных весов на самом деле не определены – их бесконечно много

Вопрос №4

Если leaf-wise построение решающих деревьев имеет преимущества, используют ли его в случайном лесе?

Ответ на вопрос №4



Вопрос №5

Можно / нужно ли использовать random state при разбиении на test и train при настройке модели? Если да, нужно ли впоследствии делать проверку на разбиениях с другим random state, и как это соотносится с кросс-валидацией?

Ответ на вопрос №5

Random state использовать нужно, чтобы потом не возникло путаницы с тем, какую обучающую и валидационную выборки вы брали

train_test_split(…, random_state=None) (по умолчанию) будет выдавать разные выборки

Ответ на вопрос №5

Обучение/валидация могут быть смещены относительно того, как именно вы разобьете выборку (особенно если у вас мало данных)

Решение: сделать несколько случайных разбиений

Ответ на вопрос №5

Ссылки:

- https://www.researchgate.net/post/Repeated_N-fold_cross_validation
- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RepeatedKFold.html#sklearn.model_selection.RepeatedKFold
- https://stats.stackexchange.com/questions/82546/how-many-times-should-we-repeat-a-k-fold-cv

Вопрос №6

Есть ли способ “заставить” модель (например, случайный лес) обращать особое внимание на какой-то признак (например, чтобы использовать знания специалиста в предметной области, который рекомендует особо выделить этот признак)?

Ответ на вопрос №6

Не факт, что вам стоит это делать – потому что машинное обучение сделает все за вас.

Вся суть в том, чтобы найти сложные закономерности – и если эксперт считает, что какой-то признак очень важен, а очень хорошая модель считает наоборот, то, возможно, эксперт ошибается

С другой стороны, признаки, предложенные экспертом, могут сильно улучшить качество модели

Ответ на вопрос №6

Если очень хотите – в линейных моделях, которые подвержены влиянию масштаба, можете увеличить масштаб конкретного признака

В деревьях можно при построении дерева часто брать конкретный признак при разбиении вершин

Ответ на вопрос №6

Ссылка:

- https://stackoverflow.com/questions/38034702/how-to-put-more-weight-on-certain-features-in-machine-learning