Бустинг

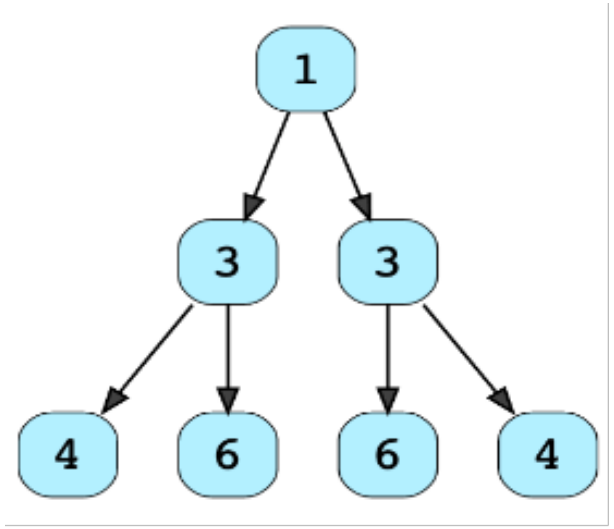
МАКСИМОВСКАЯ
АНАСТАСИЯ

План

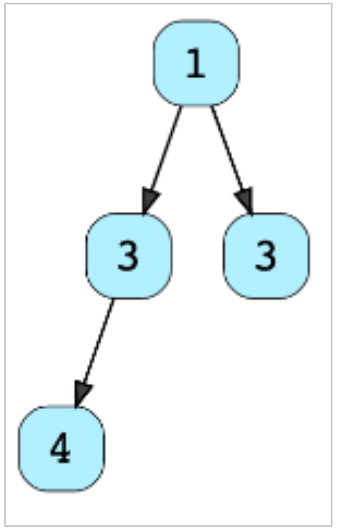
1. Градиентный бустинг
2. Экстремальный градиентный бустинг
3. CatBoost
4. LightGBM

CatBoost – **симметричные деревья**

- Делает алгоритм более устойчивым к переобучению и к изменению параметров
- Быстрее работает



A symmetric binary tree



An asymmetric binary tree

CatBoost – работа с данными

Численные данные разбиваются как обычно (рост > 170: да или нет)

Категориальные данные:

- Подсчитывается число вхождений (+ нормализация)
- Идея: добавить вероятность встретить этот лейбл среди всех
- Но такой подход может привести к переобучению

CatBoost – работа с данными

- Еще идея: разделить выборку на 2 части, на одной считать эту статистику, на другой обучаемся.
- Но тогда мы используем не все данные
- Решение: для подсчета статистики используем все данные, которые идут до этого наблюдения (лейбл этого наблюдения не будет включен в расчеты)
- Какие беды могут быть с таким подходом?

CatBoost – работа с данными

- Необходимо перемешать данные + добавить priors (https://catboost.ai/docs/concepts/python-reference_parameters-list.html)
- Перестановки (permutation) происходит 4 раза: первый – для выбора структуры деревьев
- Используются комбинации категориальных признаков
- Задачка: есть 4 цвета у 5 пород единорогов. Сколько комбинаций разноцветных единорогов может быть?

CatBoost – работа с данными

- Ответ: 4*5 = 20
- С ростом числа категорий число комбинаций растет экспоненциально
- Поэтому хорошие комбинации отбираются жадным способом: когда выбирается структура деревья, необходимо выбрать первую ноду, для нее будет выбран один категориальный признак
- На следующей ноде пробуем включить все комбинации с выбранным в предыдущем пункте категориальным признаков

CatBoost – работа с данными

- One-Hot кодирование происходит внутри алгоритма. Хорошо работает для небольшого числа категорий
- Поэтому не надо делать это самому! (алгоритм будет думать, что это еще один признак, время сильно увеличится)

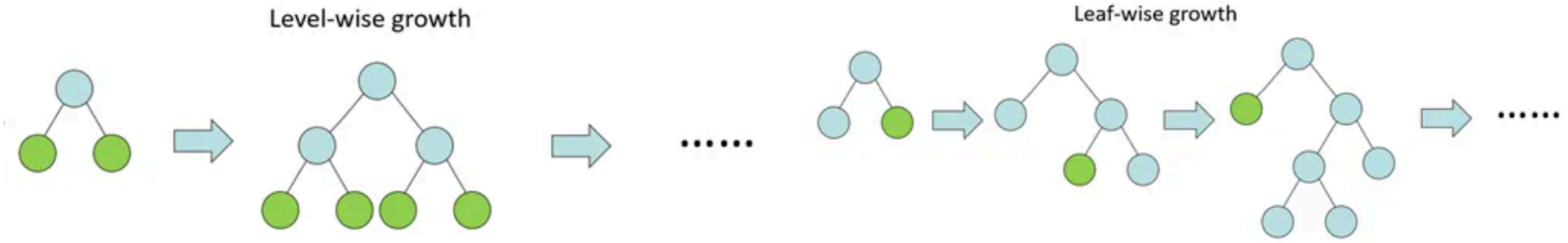
Упорядоченный (ordered) бустинг

- В обычном бустинге значение в листе = среднее по всем градиентам
- Этот градиент – оценка градиентам по всем возможным наблюдениям, которые содержатся в этом листе. Но это оценка будет смещенной (biased), т.к. используется модель, которая была построена на этих наблюдениях
- В CatBoost оценка градиента делается для каждого объекта отдельно, при обучении модели используется только те наблюдения, которые были до данного. По сути для каждого объекта – своя модель
- Таким образом, при оценке градиента модель еще не видела лейбл данного объекта

Упорядоченный (ordered) бустинг

- Опять беда: $O(n^2)$ время и память
- Упрощение: вместо n моделей обучаем logn моделей (сначала на 1 объекте, потом на 1 и 2, потом на 1, 2, 3, 4 и т.д.)
- Для наборов данных менее 100 000 наблюдений помогает улучшить качество
- Но уже прошло много времени и улучшений с момента релиза, и алгоритм работает в 4 раза быстрее, чем XGBoost на больших датасетах и +- одинаково с LightGBM
- На маленьких датасетах примерно как XGBoost и в 2 раза дольше, чем LightGBM

LightGBM



LightGBM

- И XGBoost, и LightGBM строят деревья leaf-wise
- Но при обучении деревьев LightGBM делает это вертикально, по листьям: выбирается разбиение, которое наиболее сильно уменьшает потери (loss)
- Level-wise обучение можно рассматривать как форму упорядоченного обучения, так как при leaf-wise обучении можно построить любое дерево, которое может пройти level-wise обучение, тогда как обратное не имеет места
- Leaf-wise обучение более склонно к переобучению, но более гибкое. Хорошо подходит для больших наборов данных

Источники

1. Лекции по машинному обучению на ФКН: градиентный бустинг – https://github.com/esokolov/ml-course-hse/blob/master/2019-fall/lecture-notes/lecture09-ensembles.pdf
2. Лекции по машинному обучению на ФКН: экстремальный градиентный бустинг – https://github.com/esokolov/ml-course-hse/blob/master/2019-fall/lecture-notes/lecture10-ensembles.pdf
3. Лекция по CatBoost (их несколько на ютубе от Яндекса, все хорошие, но на английском): https://www.youtube.com/watch?v=8o0e-r0B5xQ

Дополнительное изучение

Дополнительное самостоятельное изучение (маленькая ДЗ ☺):

1. Feature Importance используя SHAP: https://github.com/slundberg/shap
2. Хороший туториал по LightGBM: https://ru.raw3h.net/page/what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-5295f7/
3. Сранение XGBoost и LightGBM: https://mlexplained.com/2018/01/05/lightgbm-and-xgboost-explained/
4. И еще всякие тулы: https://catboost.ai/docs/concepts/model-analysis.html

По вопросам с вебинара

- Target encoding / One Hot encoding: что, зачем, как – https://medium.com/analytics-vidhya/target-encoding-vs-one-hot-encoding-with-simple-examples-276a7e7b3e64