XGBoost (eXtreme Gradient Boosting) (regularized boosting)

Литература

XGBoost Parameters (official guide)

http://xgboost.readthedocs.org/en/latest/parameter.html#general-parameters

XGBoost Demo Codes (xgboost GitHub repository)

https://github.com/dmlc/xgboost/tree/master/demo/guide-python

Python API Reference (official guide)

http://xgboost.readthedocs.org/en/latest/python/python_api.html

Постоить модель с помощью XGBoost легко.

Подобрать оптимальные значения параметров XGBoost очень сложно.

Какие параметры настраивать, какие значения оптимальны?

Однозначного ответа нет.

Есть даже «стакинг XGBoost’ов»

1. XGBoost лучше GBM

1. Есть регуляризация
2. Лучше реализована параллелизация, Hadoop
3. Эмпирические приемы работы с пропусками
4. Обрезание деревьев.

Сначала строится дерево с максимальным числом слоев. Допускается даже увеличение загрязнения при расщеплении. Затем отбрасываются неэффективные узлы.

Параметры XGBoost

Авторы и разработчики XGBoost’а делят параметры на 3 группы.

General Parameters

Booster Parameters

Learning Task Parameters

General Parameters

booster [default=gbtree]

Выбор слабой модели. Выбираем один из 2-х вариантов:

gbtree: деревья

gblinear: линейные модели (либо хуже, либо очень долго)

silent [default=0]:

Вывод промежуточных результатов в ходе обучения модели

1 : промежуточные результаты не выдаются

0 : промежуточные результаты выдаются.

nthread [default to maximum number of threads available if not set]

Число ядер, используемых при вычислениях.

… есть еще параметры ...

Booster Parameters (для варианта booster =gbtree)

num_boosting_rounds

Число деревьев

eta [default=0.3]

Скорость обучения. Обычно используют значения в интервале 0.01-0.2

min_child_weight [default=1]

Минимальное значение для суммы весов в узле потомке.

Аналогично минимально возможному числу наблюдений в узле потомке **min_child_leaf** в GBM, но есть отличие. суммы весов не то же самое, что число наблюдений. Предназначено для предотвращения переподгонки. Слишком большое значение ухудшит модель. Интуиция не работает, надо подбирать, используя кросс-валидацию.

max_depth [default=6]

Максимальное число слоев дерева.

Предотвращает переподгонку. Значение может быть разным в разных задачах. Надо подбирать, используя кросс-валидацию. Обычно используют значения в интервале 3-10.

max_leaf_nodes

Максимальное количество конечных узлов в дереве. Если используется, заменяет **max_depth**.

gamma [default=0]

Запрещает расщепление узла, если загрязнение потомков уменьшилось менее, чем на **gamma**. Значение параметра зависит от используемого критерия качества. Надо подбирать, используя кросс-валидацию.

max_delta_step [default=0]

Не знаю. Мало, кто знает. Используется значение по умолчанию.

subsample [default=1]

Доля наблюдений, попадающих в случайную подвыборку при построении очередного дерева. Маленькие значения препятствуют переподгонке, очень маленькие ухудшают качество модели. Обычно используют значения в интервале 0.5-1.

colsample_bytree [default=1]

Доля переменных, попадающих в случайную подвыборку при построении очередного дерева. Маленькие значения препятствуют переподгонке, очень маленькие ухудшают качество модели. Обычно используют значения в интервале 0.5-1.

colsample_bylevel [default=1]

Доля переменных, попадающих в случайную подвыборку при расщеплении очередного узла дерева. Маленькие значения препятствуют переподгонке, очень маленькие ухудшают качество модели. Забава для параноиков. Замедляет обучение.

lambda [default=1]

Коэффициент при L2 регуляризационном слагаемом (как в Ridge регрессии). Надо подбирать, используя кросс-валидацию.

alpha [default=0]

Коэффициент при L1 регуляризационном слагаемом (как в Lasso регрессии). Надо подбирать, используя кросс-валидацию. Рекомендуется использовать при большом числе переменных.

scale_pos_weight [default=1]

Не знаю. Но параметр важный. Используется при анализе несбалансированных выборок.

Learning Task Parameters

Описывают критерий качества, используемый при обучении.

objective [default=reg:linear]

Задается критерий качества, используемый при обучении. Чаще всего используются:

binary:logistic – когда имеется два класса, выходные значения — вероятности принадлежать классу, не код класса

multi:softmax – когда имеется больше двух классов, выходные значения — код класса, не вероятности принадлежать классу.

Надо еще дополнительно задать **num_class** — число классов в задаче.

multi:softprob – когда имеется больше двух классов, выходные значения — вероятности принадлежать классу, не код класса.

eval_metric [default according to objective]

Метрика, используемая при валидации. По умолчанию используется rmse в задаче регрессии и error в задаче классификации.

Популярные варианты:

rmse – root mean square error

mae – mean absolute error

logloss – negative log-likelihood

error –Binary classification error rate (0.5 threshold)

merror – Multiclass classification error rate

mlogloss – Multiclass logloss

auc: Area under the curve

seed [default=0]

Зерно датчика случайных чисел. Полезен для воспроизводимости результатов, в частности при подборе параметров.

Scikit-Learn содержит XGBClassifier обертку для модуля(?) xgboost

При этом некоторые параметры приобретают другое имя:

1. eta –> learning_rate
2. lambda –> reg_lambda
3. alpha –> reg_alpha