Previous Lecture (/courses/78878/lectures/1148762)
Complete and continue (/courses/78878/lectures/1437360)

Parameters cheat-sheet

The following list presents the full set of parameters used to tune XGBoost algorithm. Use it as a cheat-sheet while experimenting.

General

- booster=[**gbtree**|gblinear] - which booster to use.
- silent=[**0**|1] - 0 prints running messages, 1 means silent mode
- nthread - number of threads to use. Maximum by default

Tree Booster

- eta=[0 .. **0.3** .. 1] - step shrinkage used in update to prevents overfitting. After each boosting step eta shrinks instance weights. Lower value makes learning process more conservative (slower learning),
- gamma=[**0** .. ∞] - minimum loss reduction required to make a further partition on a leaf node of the tree. Higher value makes the algorithm more conservative,
- max_depth=[1 .. **6** .. ∞] - maximum depth of each tree,
- min_child_weight=[0 .. **1** .. ∞] - minimum sum of instance weight needed in a tree node. Further partitioning from that node is abandoned when a sum is not obtained. Higher value makes the algorithm more conservative.
- max_delta_step=[**0** .. ∞] - maximum delta step we allow each tree's weight estimation to be. If the value is set to 0, it means there is no constraint. If it is set to a positive value it can help making the update step more conservative.
- subsample=[0,**1**] - subsample ratio of randomly selected training instances used to grow trees,
- colsample_bytree=[0,**1**] - subsample ratio of columns when construction each tree,
- lambda=[**1**] - L2 regularization term on weights
- alpha=[**0**] - L1 regularization term on weights

Linear Booster

- alpha=[**0**] - L1 regularization term on weights
- lambda=[**0**] - L2 regularization term on weights
- lambda_bias=[**0**] - L2 regularization term on bias

Learning task parameters

- objective=[
  - **reg:linear** # linear regression,
  - reg:logistic # logistic regression,
  - binary:logistic # logistic regression for binary classification, outputs probability,
  - binary:logitraw #logistic regression for binary classification, outputs score before logistic transformation,
  - count:poisson # poisson regression for count data, outputs mean of poisson distribution,
  - multi:softmax # do multiclass classification using softmax objective,
  - multi:softprob # same as above but outputs probability for each class,

Previous Lecture (/courses/78878/lectures/1148762)

Complete and continue (/courses/78878/lectures/1437360)

- rank:pairwise # execute ranking task by minimizing the pairwise loss
]

- base_score=[0.5] - global bias. Initial prediction score for all instances
- eval_metric=[rmse|logloss|error|merror|mlogloss|auc|...] - evaluation metric. Default value will be assigned based on the objective. There is possibility of having custom metric.
- seed=[0] - seed used for reproducibility