Supplementary material for Bagging and Boosting in Sklearn

February 26, 2018

This supplementary material briefly illustrates the functions used in the Lab 5. For more details, please see "http://scikit-learn.org". Note that in this material, we assume every student knows Python's class mechanism.

- Bagging ("from sklearn.ensemble import BaggingClassifier as bagging")
  - bagging(base_estimator=None, n_estimators=10, max_samples=1.0, max_features=1.0, bootstrap=True, bootstrap_features=False, oob_score=False, warm_start=False, n_jobs=1, random_state=None, verbose=0)
    - Common parameters
      - base_estimator: The base estimator to fit on random subsets of the dataset. The default estimator is a decision tree.
      - n_estimators: The number of base estimators in the ensemble. The default value is 10.
      - max_samples: The number of samples to draw from X to train each base estimator.
      - max_features: The number of features to draw from X to train each base estimator.
    - Common methods
      - fit(X, y): fit linear model.
      - score(X, y): return the mean accuracy on the given data X and labels y.
- Boosting ("from sklearn.ensemble import GradientBoostingClassifier as boosting")
  - boosting(loss='deviance', learning_rate=0.1, n_estimators=100, subsample=1.0, criterion='friedman_mse', min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_depth=3, min_impurity_decrease=0.0,

min_impurity_split=None, init=None, random_state=None, max_features=None, verbose=0, max_leaf_nodes=None, warm_start=False, presort='auto')

* Common parameters
  - loss: loss function to be optimized. Two options: deviance and exponential. Deviance refers to deviance (= logistic regression) for classification with probabilistic outputs. Exponential recovers the AdaBoost algorithm.
  - n_estimators: the number of boosting stages to perform.
* Common attributes
  - feature_importances_: the feature importances (the higher, the more important the feature).
  - oob_improvement_: the improvement in loss on the out-of-bag samples relative to the previous iteration.
* Common methods
  - fit(X, y): fit the gradient boosting model.
  - score(X, y): return the mean accuracy on the given data X and labels y.
  - staged_predict(X): predict class at each stage for X.