Ensemble methods
(Decision by Committee)

Ensemble Methods

- Use multiple models to obtain better predictive performance.
- Includes much more computation, since you are training multiple learners
- Typically combine multiple fast learners (like decision trees)
- Tend to overfit
- Tend to get better results since there is deliberately introduced significant diversity among models

Motivation for Ensemble Learning

- No Free Lunch theorem: There is no algorithm that is always the most accurate
- Generate a group of base-learners which when combined have higher accuracy
- Different learners use different
  - Algorithms
  - Parameters
  - Representations
  - Training sets
  - Etc.

Bagging (Bootstrap aggregating)

- Take M bootstrap samples (with replacement)
- Train M different classifiers on these bootstrap samples
- For a new query, let all classifiers predict and take an average (or majority vote)

  If the classifiers make independent errors, then their ensemble can improve performance.

 Stated differently: the variance in the prediction is reduced (we don’t suffer from the random errors that a single classifier is bound to make).

Boosting

- Train classifiers (e.g. decision trees) in a sequence.
- A new classifier should focus on those cases which were incorrectly classified in the last round.
- Combine the classifiers by letting them vote on the final prediction (like bagging).
- Each classifier is “weak” but the ensemble is “strong.”
- AdaBoost is a specific boosting method.

Boosting

- We adaptively weigh each data case.
- Data cases which are wrongly classified get high weight (the algorithm will focus on them)
- Each boosting round learns a new (simple) classifier on the weighed dataset.
- These classifiers are weighed to combine them into a single powerful classifier.
- Classifiers that obtain low training error rate have high weight.

Bagging: Bootstrap aggregating

- Each model in the ensemble votes with equal weight
- Train each model with a random training set

Boosting

- Incremental
- Build new models that try to do better on previous model's mis-classifications
  - Can get better accuracy
  - Tends to overfit
- Adaboost is canonical boosting algorithm

Random Forest

- Ensemble consisting of a bagging of decision tree learners with a randomized selection of features at each split.

- Grow many trees on datasets sampled from the original dataset with replacement (a bootstrap sample).
  - Draw K bootstrap samples of a fixed size
  - Grow a DT, randomly sampling a few attributes/dimensions to split on at each internal node

- Average the predictions of the trees for a new query (or take majority vote)

- Random Forests are state of the art classifiers!

Random forest

- **Random forest** (or **random forests**) is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees.
- The term came from **random decision forests** that was first proposed by Tin Kam Ho of Bell Labs in 1995.
- The method combines Breiman's "bagging" idea and the random selection of features.

Features and Advantages

The advantages of random forest are:

- It is one of the most accurate learning algorithms available. For many data sets, it produces a highly accurate classifier.
- It runs efficiently on large databases.
- It can handle thousands of input variables without variable deletion.
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.
- It has methods for balancing error in class population unbalanced data sets.
- Generated forests can be saved for future use on other data.

```
Boston House Prices dataset

Notes
------
Data Set Characteristics:

    :Number of Instances: 506

    :Number of Attributes: 13 numeric/categorical predictive

    :Median Value (attribute 14) is usually the target

    :Attribute Information (in order):
        - CRIM     per capita crime rate by town
        - ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
        - INDUS    proportion of non-retail business acres per town
        - CHAS     Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
        - NOX      nitric oxides concentration (parts per 10 million)
        - RM       average number of rooms per dwelling
        - AGE      proportion of owner-occupied units built prior to 1940
        - DIS      weighted distances to five Boston employment centres
        - RAD      index of accessibility to radial highways
        - TAX      full-value property-tax rate per $10,000
        - PTRATIO  pupil-teacher ratio by town
        - B        1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
        - LSTAT    % lower status of the population
        - MEDV     Median value of owner-occupied homes in $1000's

    :Missing Attribute Values: None

    :Creator: Harrison, D. and Rubinfeld, D.L.
```