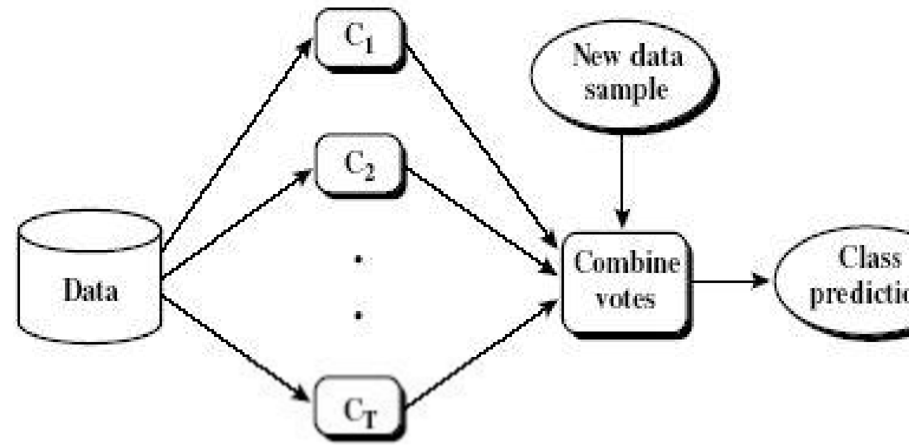
DATA MINING

RANDOM FOREST

Ensemble Methods: Increasing the Accura



- Ensemble methods
  - Use a combination of models to increase accuracy
  - Combine a series of k learned models, M_1, M_2, …, M_k, with the aim of creating an improved model M*
- Popular ensemble methods
  - Bagging: averaging the prediction over a collection of classifiers
  - Boosting: weighted vote with a collection of classifiers
  - Ensemble: combining a set of heterogeneous classifiers

Bagging: Boostrap Aggregation

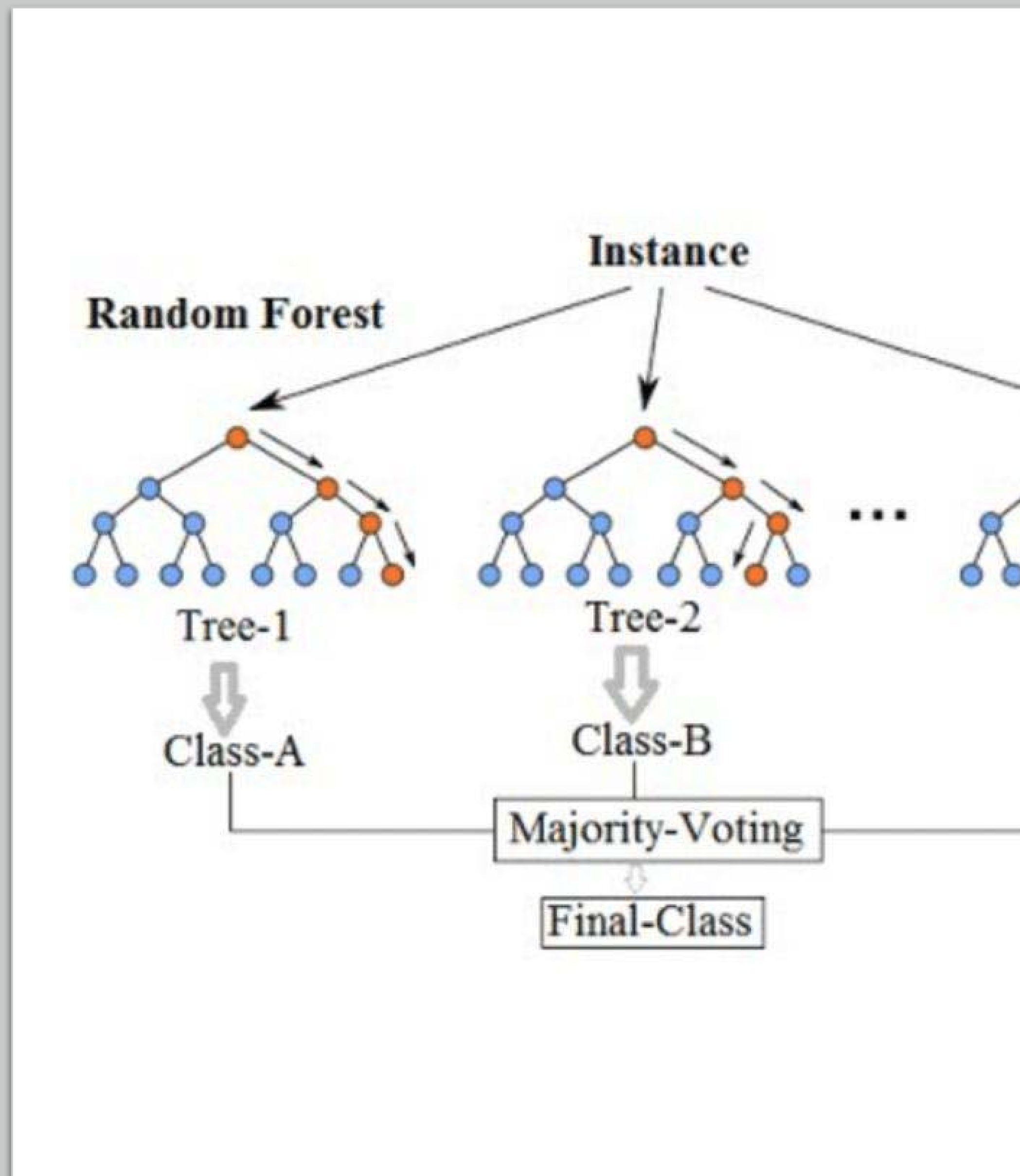
- Analogy: Diagnosis based on multiple doctors' majority vote
- Training
  - Given a set D of *d* tuples, at each iteration *i*, a training set D_i of *d* tuples is sampled with replacement from D (i.e., bootstrap)
  - A classifier model M_i is learned for each training set D_i
- Classification: classify an unknown sample **X**
  - Each classifier M_i returns its class prediction
  - The bagged classifier M* counts the votes and assigns the class with the most votes to **X**
- Prediction: can be applied to the prediction of continuous values by taking th average value of each prediction for a given test tuple
- Accuracy
  - Often significantly better than a single classifier derived from D
  - For noise data: not considerably worse, more robust
  - Proved improved accuracy in prediction

Boosting

- Analogy: Consult several doctors, based on a combination of weighted diagnoses—weight assigned based on the previous diagnosis accuracy
- How boosting works?
  - **Weights** are assigned to each training tuple
  - A series of k classifiers is iteratively learned
  - After a classifier M_i is learned, the weights are updated to allow the subsequent classifier, M_{i+1}, to **pay more attention the training tuples that were misclassified** by M_i
  - The final **M* combines the votes** of each individual classifie where the weight of each classifier's vote is a function of its accuracy
- Boosting algorithm can be extended for numeric prediction
- Comparing with bagging: Boosting tends to have greater accura but it also risks overfitting the model to misclassified data

RANDOM FOREST

- The basic idea behind Random Forest is to create a large number of decision trees, each trained on a random subset of the data and using a random subset of the features. The final prediction is then made by combining the predictions of all the trees.
- Random Forest is an ensemble method, which means it combines the predictions of multiple models to improve accuracy and reduce overfitting.



- To prevent overfitting:
  - bagging and
  - feature selection.

- ✔ Bagging, short for bootstrap aggregating, involves randomly selecting subsets of the training data with replacement and training a decision tree on each subset.
- ✔ By using different subsets of the data for each tree, Random Forest reduces the variance of the predictions and makes them more robust.

- ✔ Feature selection involves randomly selecting a subse
the features for each tree.
- ✔ By using different subsets o
features for each tree, Ranc
Forest reduces the correlati
between the trees and mak
them more diverse.

Out-of-bag Data

- For each tree in the forest, we select a bootstrap sample from data.
- The bootstrap sample is used to grow the tree.
- The remaining data are said to be “out-of-bag” (about one-thi the cases).
- The out-of-bag data can serve as a test set for the tree grown bootstrap sample.

The out-of-bag Error Estimate

- Think of a single case in the training set. It will be out-of-bag ir one-third of the trees.
- Each time it is out of bag, pass it down the tree and get a pred class.
- The RF prediction is the class that is chosen the most often.
- For each case, the RF prediction is either correct or incorrect.
  - Average over the cases within each class to get a classwise out-of-bag error rate.
  - Average over all cases to get an overall out-of-bag error rat

Using out-of-bag Data to Choose m

- The out-of-bag error rate is used to select m.
- Here's how:

1. Start with $m = \sqrt{M}$.
2. Run a few trees, recording the out-of-bag error rate.
3. Increase m, decrease m, until you are reasonably confident y
   found a value with minimum out-of-bag error rate.

Out-of-bag Data and Variable Importanc

- Consider a single tree (fit to a bootstrap sample).
  - 1. Use the tree to predict the class of each outof-bag case.
  - 2. Randomly permute the values of the variable of interest in all the out-of-bag cases, and use the tree to predict the class for these pertu out-of-bag cases.
- The variable importance is the increase in the misclassificatior between steps 1 and 2,

Random Forest (Breiman 2001)

- Two Methods to construct Random Forest:
    - Forest-RI (*random input selection*): Randomly select, at each node, F attributes as candidates for the split at the node. The
    - Forest-RC (*random linear combinations*): Creates new attributes (or features) that are a linear combination of the existing attributes (reduces the correlation between individual classifiers)
- Comparable in accuracy to Adaboost, but more robust to errors and outliers
- Insensitive to the number of attributes selected for consideration at each split, and faster than bagging or boosting