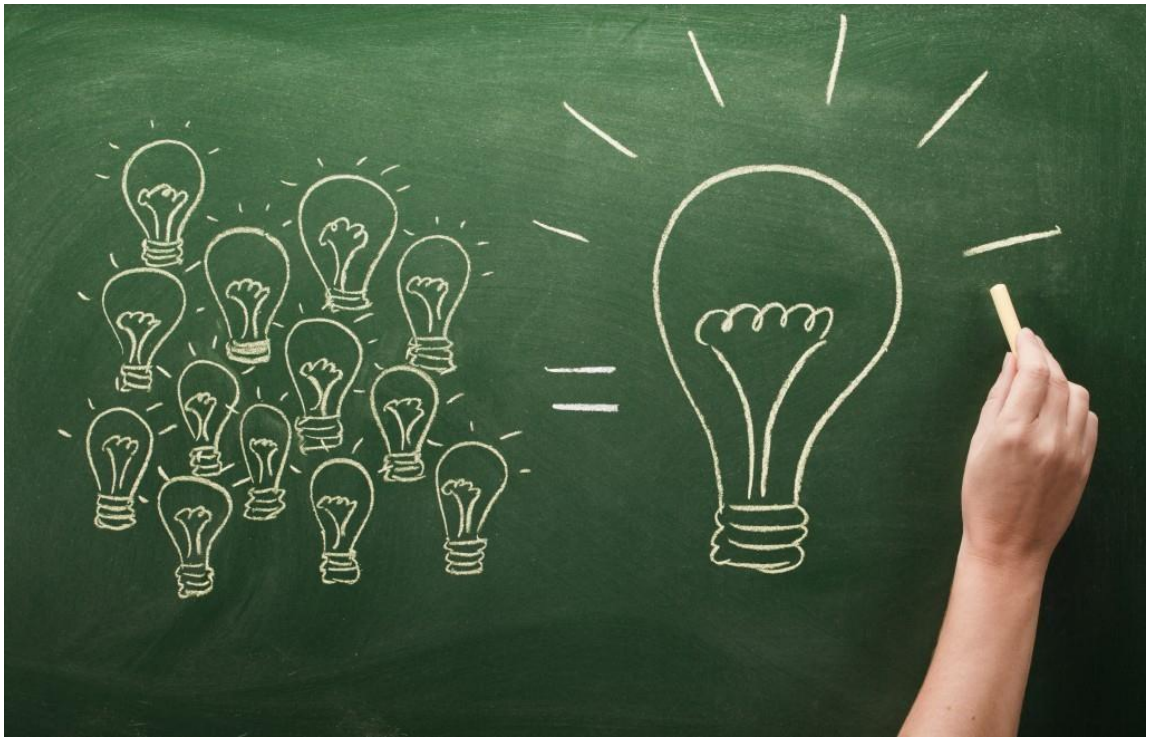
Lecture 6 Random Forest

1. From decision tree to random forest
2. Algorithm
3. Pros & Cons
4. More information

1. From Decision tree to Random Forrest

- To handle decision tree's generalization capability, we may use a new approach
  - Power of the crowds



Definition

- A single decision tree does not perform well
- But it is super fast, what if we learn multiple trees?
- 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.

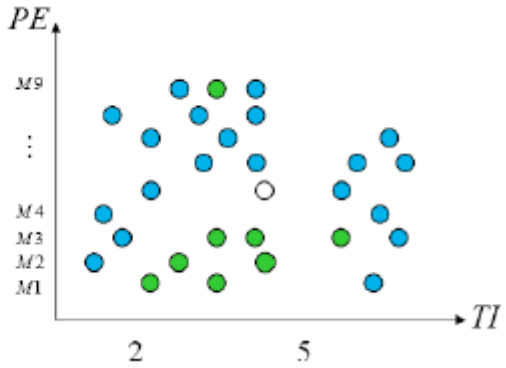
Decision trees

- Decision trees are individual learners that are combined.
  - They are one of the most popular learning methods commonly used for data exploration.
  - One type of decision tree is called **CART** - classification and regression tree.
- CART - greedy, top-down binary, recursive partitioning, that divides feature space into sets of disjoint rectangular regions.
  - Regions should be pure wrt response variable.
  - Simple model is fit in each region – majority vote for classification, constant value for regression.

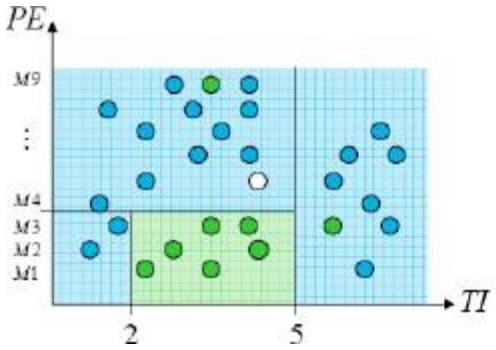
Decision tress involve greedy, recursive partitioning.

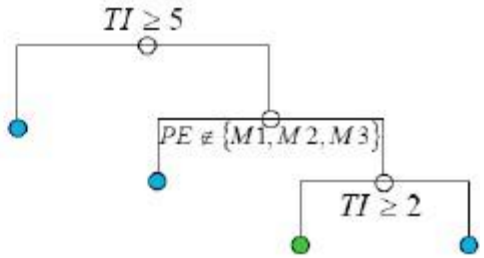
Simple dataset with two predictors

TI	PE	Response
1.0	M2	good
2.0	M1	bad
...	...	...
4.5	M5	?



Greedy, recursive partitioning along TI and PE





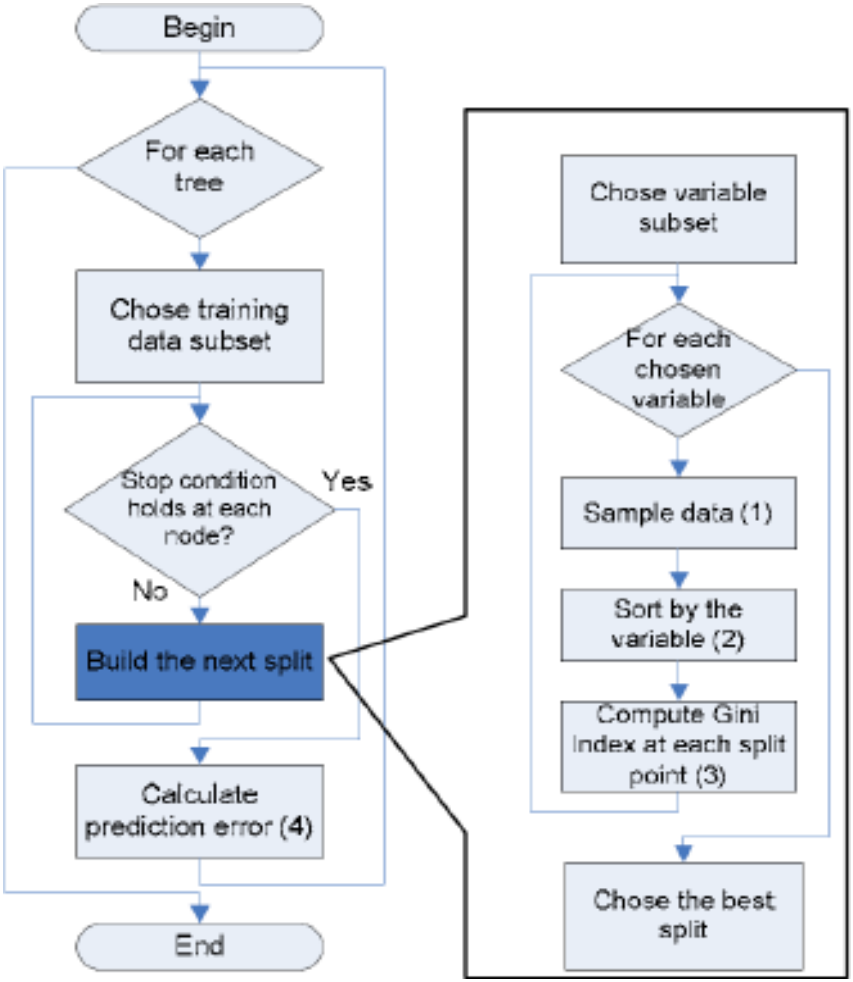
2. Algorithm

Each tree is constructed using the following algorithm:

1. Let the number of training cases be *N*, and the number of variables in the classifier be *M*.
2. We are told the number *m* of input variables to be used to determine the decision at a node of the tree; *m* should be much less than *M*.
3. Choose a training set for this tree by choosing *n* times with replacement from all *N* available training cases (i.e. take a bootstrap sample). Use the **rest of the cases** to estimate the error of the tree, by predicting their classes.
4. For each node of the tree, randomly choose *m* variables on which to base the decision at that node. Calculate the best split based on these *m* variables in the training set.
5. Each tree is fully grown and not pruned (as may be done in constructing a normal tree classifier).

For prediction a new sample is pushed down the tree. It is assigned the label of the training sample in the terminal node it ends up in. This procedure is iterated over all trees in the ensemble, and the average vote of all trees is reported as random forest prediction.

Algorithm flow chart



Practical consideration

- Splits are chosen according to a purity measure:
  - E.g., squared error (regression), Gini index or deviance (classification)
- How to select *n*?
  - Build trees until the error no longer decreases
- How to select *m*?
  - Try to recommend defaults, half of them and twice of them and pick the best.

3. Pros and Cons

The advantages of random forest:

- 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 gives estimates of what variables are important in the classification.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- 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.
- Prototypes are computed that give information about the relation between the variables and the classification.
- It computes proximities between pairs of cases that can be used in clustering, locating outliers, or (by scaling) give interesting views of the data.
- The capabilities of the above can be extended to unlabeled data, leading to unsupervised clustering, data views and outlier detection.
- It offers an experimental method for detecting variable interactions.

Disadvantages

- Random forests have been observed to overfit for some datasets with noisy classification/regression tasks.
- For data including categorical variables with different number of levels, random forests are biased in favor of those attributes with more levels. Therefore, the variable importance scores from random forest are not reliable for this type of data.

4. Additional information

Estimating the test error:

- While growing forest, estimate test error from training samples
  - For each tree grown, 33-36% of samples are not selected in bootstrap, called **out of bootstrap** (OOB) samples
  - Using OOB samples as input to the corresponding tree, predictions are made as if they were novel test samples
  - Through book-keeping, majority vote (classification), average (regression) is computed for all OOB samples from all trees.
  - Such estimated test error is very accurate in practice, with reasonable *n*

Estimating the importance of each input:

- Denote by $\hat{e}$ the OOB estimate of the loss when using original training set, D.
- For each input x_p where $p \in \{1,..,k\}$
  - Randomly permute p^{th} input to generate a new set of samples $D' = \{(y1,x'1),\ldots,(yN,x'N)\}$
  - Compute OOB estimate $\hat{e}_k$ of prediction error with the new samples
- A measure of importance of predictor x_p is $\hat{e}_k - \hat{e}$, the increase in error due to random perturbation of p^{th} predictor

Summary:

- Fast fast fast!
  - RF is fast to build. Even faster to predict!
  - Practically speaking, not requiring cross-validation alone for model selection significantly speeds training by 10x-100x or more.
  - Fully parallelizable … to go even faster!
- Automatic predictor (inputs) selection from large number of candidates
- Resistance to over training
- Ability to handle data without preprocessing
  - data does not need to be rescaled, transformed, or modified
  - resistant to outliers
  - automatic handling of missing values

Want to know more?

https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm