Ensemble Methods And Random Forests

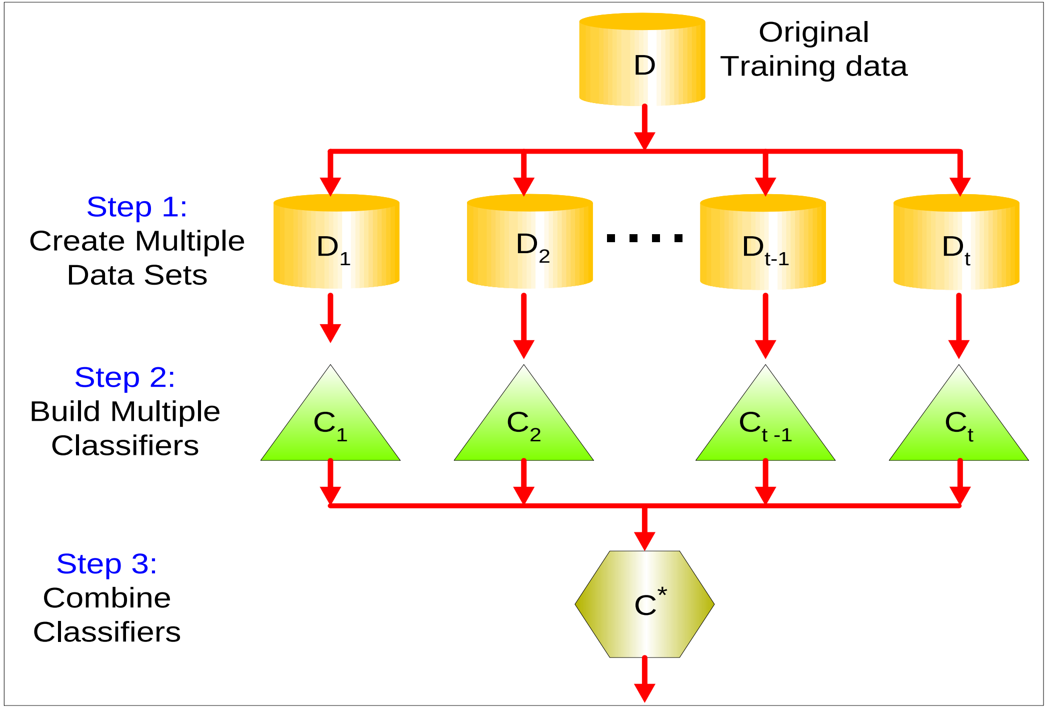
Ensemble Methods

- Overview and rationale
- Binomial Distribution
- Bagging
- Random Forests
- Boosting
- AdaBoost

Ensemble Methods

- Improve classification accuracy by combining classifiers
- You can also have an ensemble of regression models

General Idea of Ensemble Methods

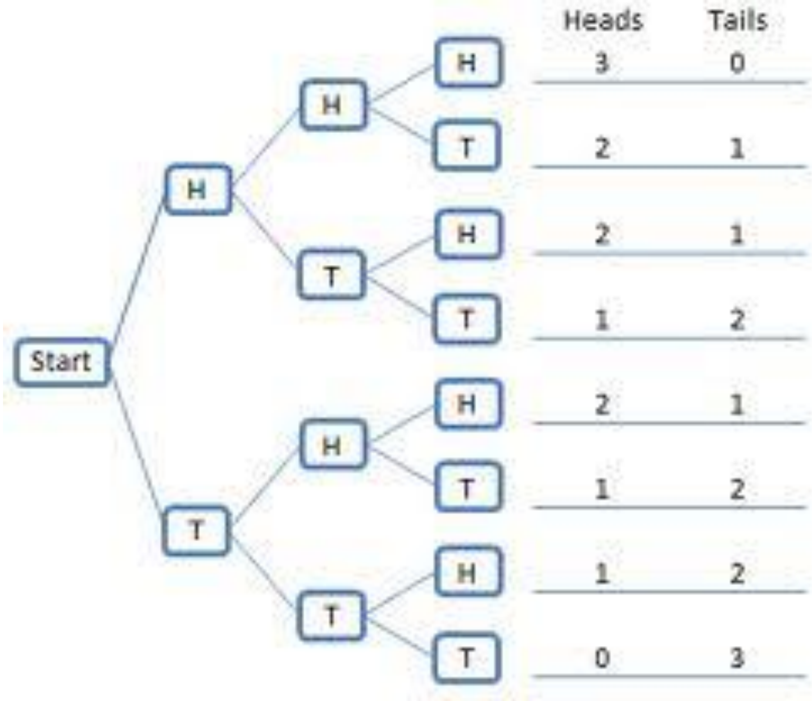


datasciencedojo
unleash the data scientist in you

Ensemble Methods

- Overview and rationale
- **Binomial Distribution**
- Bagging
- Random Forests
- Boosting
- AdaBoost

Binomial Distribution



Binomial Distribution

$$f(k;n,p) = \Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Tossing A Biased Coin

What is the probability of 0,1,2,3,4,5,6 heads in six tosses of a biased coin?

n = 6
k = 0,1,2,3,4,5,6
P=0.3 (prob of head)

$$\Pr(0\text{ heads}) = f(0) = \Pr(X=0) = \binom{6}{0}0.3^0(1-0.3)^{6-0} \approx 0.1176$$

$$\Pr(1\text{ head }) = f(1) = \Pr(X=1) = \binom{6}{1}0.3^1(1-0.3)^{6-1} \approx 0.3025$$

$$\Pr(2\text{ heads}) = f(2) = \Pr(X=2) = \binom{6}{2}0.3^2(1-0.3)^{6-2} \approx 0.3241$$

$$\Pr(3\text{ heads}) = f(3) = \Pr(X=3) = \binom{6}{3}0.3^3(1-0.3)^{6-3} \approx 0.1852$$

$$\Pr(4\text{ heads}) = f(4) = \Pr(X=4) = \binom{6}{4}0.3^4(1-0.3)^{6-4} \approx 0.0595$$

$$\Pr(5\text{ heads}) = f(5) = \Pr(X=5) = \binom{6}{5}0.3^5(1-0.3)^{6-5} \approx 0.0102$$

$$\Pr(6\text{ heads}) = f(6) = \Pr(X=6) = \binom{6}{6}0.3^6(1-0.3)^{6-6} \approx 0.0007$$

Rolling a Die

- When rolling a die 100 times, what is the probability of rolling a "4" exactly 25 times?
- Number of trials n = 100
- Number of successes k = 25
- Probability of a ‘4’ in a single roll = 1/6

$$\binom{100}{25}\left(\frac{1}{6}\right)^{25}\left(1-\left(\frac{1}{6}\right)\right)^{100-25}$$

R functions:
dbinom(25,size=100,prob=1/6) = 0.009825882
Pbinom is for calculating cdfs

datasciencedojo
unleash the data scientist in you

Multiple Choice Test

- A test consists of 10 multiple choice questions with five choices for each question. As an experiment, you **guess** on each and every answer without even reading the questions.
  - What is the probability of getting **exactly 6** questions correct on this test?
  - What is the probability of getting **at least 6** questions correct on this test?
  - What is the probability of getting **less than 6** questions correct on this test?
  - What is the probability of getting **all 10** questions correct on this test?

Applications

- Number of life insurance holders who will claim in a given period
- Number of loan holders who will default in a certain period
- Number of false starts of a car in n attempts
- Number of faulty items in n samples from a production line
- **AND** Ensemble Methods

Why Does It Work?

- Suppose there are 25 base classifiers
  - Each classifier has error rate, $\varepsilon = 0.35$
  - Assume classifiers are independent
  - Probability that the ensemble classifier makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^{i} (1-\varepsilon)^{25-i} = 0.06$$

Examples of Ensemble Methods

Bagging

- All classifiers are created equal

Boosting

- **Not** all classifiers are created equal

Ensemble Methods

- Overview and rationale
- Binomial Distribution
- **Bagging**
- Random Forests
- Boosting
- AdaBoost

Bagging

- Sampling with replacement

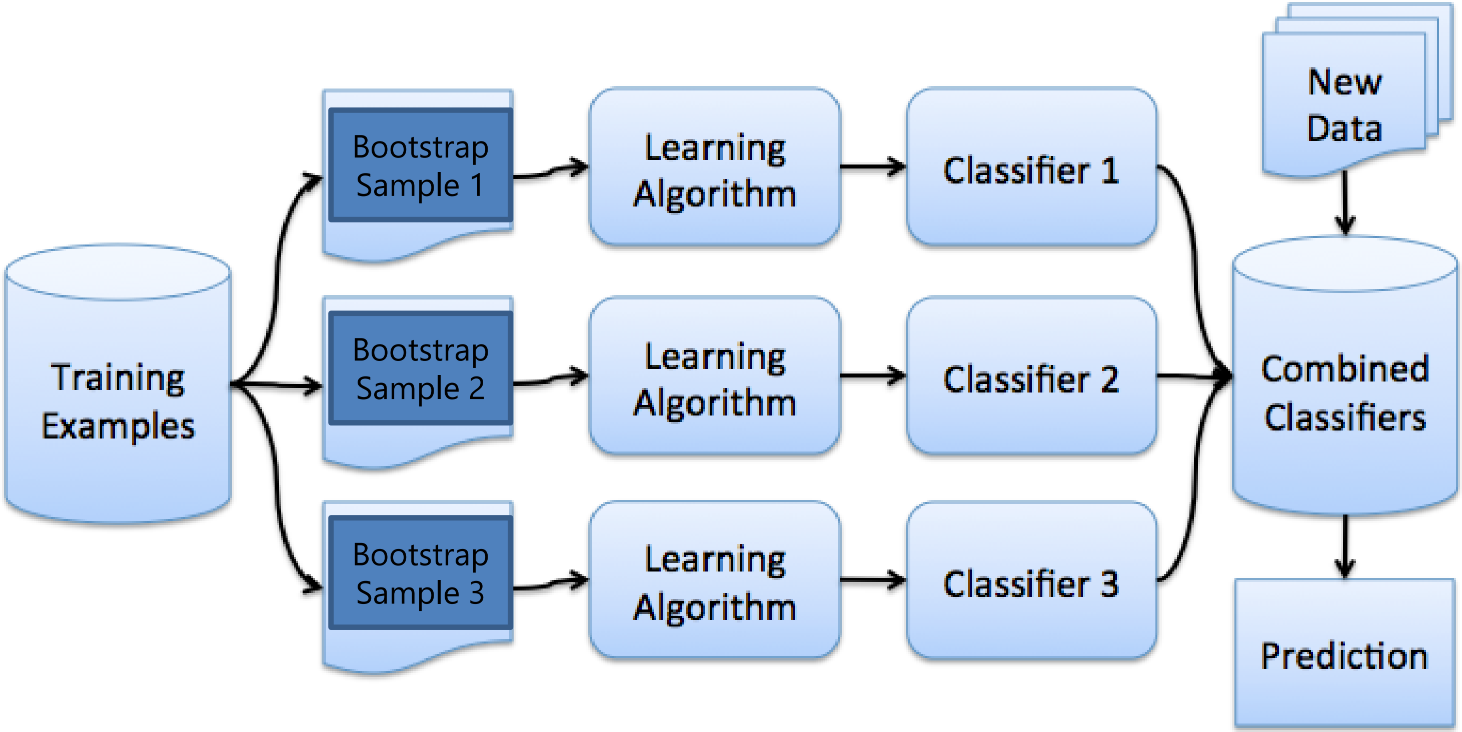
Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

- Build classifier on each bootstrap sample
- For a training data of size n, each sample has probability $[1 - (1 – 1/n)^n]$ of being selected
- If n is large, this approximates to 1-1/e ~ 0.632

Bagging

- Reduces variance in estimate
- Prevents overfitting
- Robust to outliers

Bagging


Training
Examples
Bootstrap
Sample 1
Bootstrap
Sample 2
Bootstrap
Sample 3
Learning
Algorithm
Learning
Algorithm
Learning
Algorithm
Classifier 1
Classifier 2
Classifier 3
New
Data
Combined
Classifiers
Prediction

Ensemble Methods

- Overview and rationale
- Binomial Distribution
- Bagging
- **Random Forests**
- Boosting
- AdaBoost

Random Forests Classifier

- *Random forests are a **combination of tree predicators** such that each tree depends on the values of a random vector sampled independently and with same distribution for all trees in the forest.*

- Leo Breiman

What Is A Random Forest?

- An ensemble classifier using many decision tree models
- Can be used for classification or regression
- Accuracy and variable importance information is built-in

How Do Random Forests Work?

- A different subset of the training data are selected (~2/3), with replacement, to train each tree
- Remaining training data (aka out-of-bag data or simply OOB) is used to estimate error and variable importance
- Class assignment is made by the number of votes from all of the trees, and for regression, the average of the results is used

Which Features Are Used For Learning?

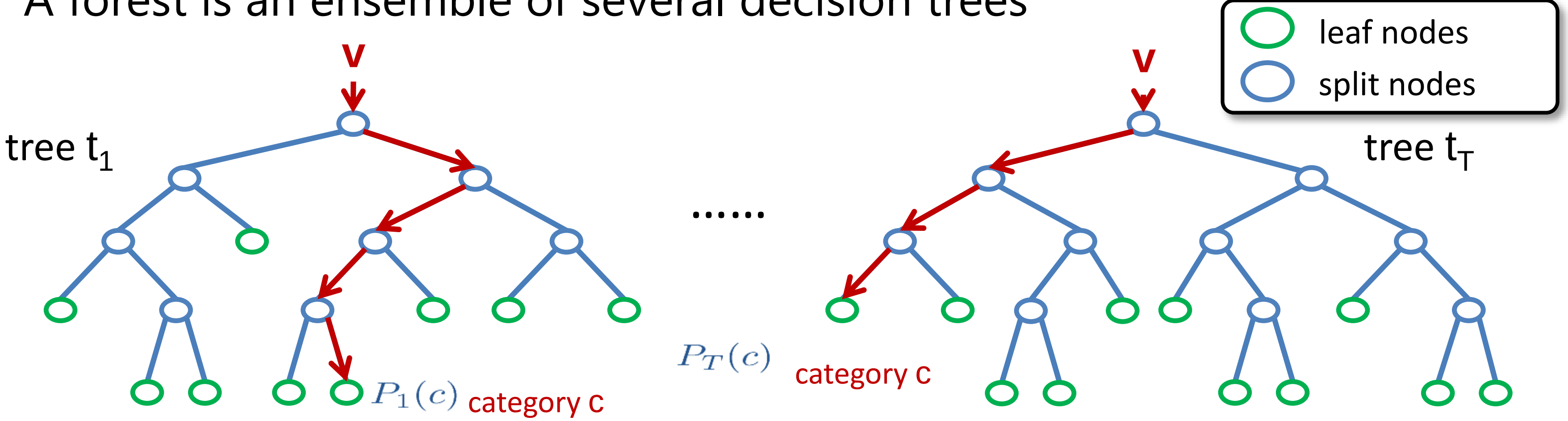
- A randomly selected subset of variables is used to split each node
- The number of variables used is decided by the user (mtry parameter in R)
- A smaller subset produces less correlation (lower error rate)

Rules of Thumb

- Given:
  - **N:** Total number of training data points
  - **M:** Number of features in training data
  - **m:** Number of features randomly selected for training each node
- Sample the data with replacement N times for building the training data for each tree.
- m<<M
- Classification: m = sqrt(M)
- Regression: m = M/3

A Forest of Trees

A forest is an ensemble of several decision trees



Classification is $$P(c|\mathbf{v}) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|\mathbf{v})$$

[Amit & Geman 97]
[Breiman 01]
[Lepetit *et al.* 06]

Learning a Forest

- Dividing training examples into T subsets improves generalization
  - Reduces memory requirements & training time
- Train each decision tree t on subset I_t
  - Same decision tree learning as before
- Multi-core friendly (GPU implementation)

Implementation Details

- How many features and thresholds to try?
  - Just one = "extremely randomized" [Geurts *et al.* 06]
  - Few → fast training, may under-fit, may be too deep
  - Many → slower training, may over-fit
- When to stop growing the tree?
  - Maximum depth
  - Minimum entropy gain
  - Delta class distribution
  - Pruning

Ensemble Methods

- Overview and rationale
- Binomial Distribution
- Bagging
- Random Forests
- **Boosting**
- AdaBoost

Boosting

- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records
  - Initially, all N records are assigned equal weights
  - Unlike bagging, weights may change at the end of boosting round

Boosting

- Records that are wrongly classified will have their weights increased
- Records that are classified correctly will have their weights decreased

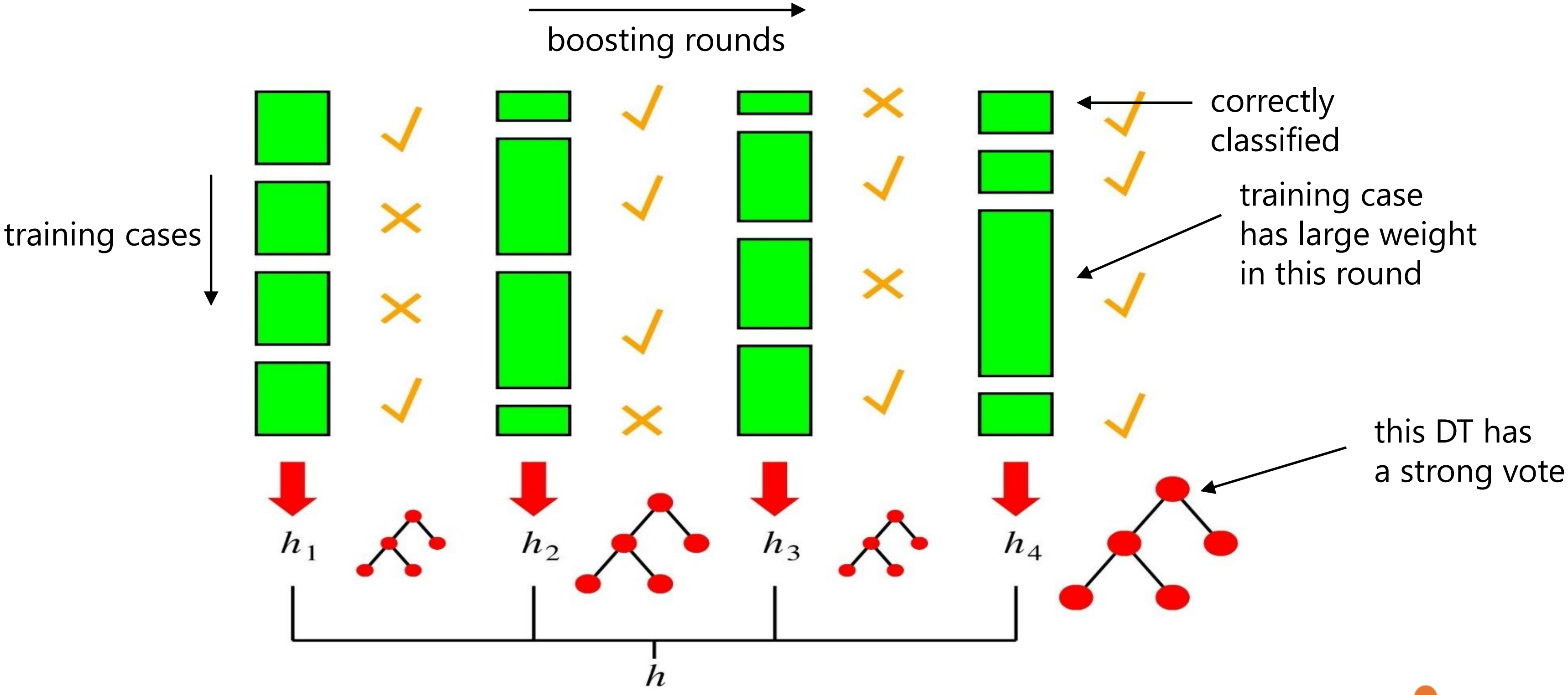
Original Data	1	2	3	4	5	6	7	8	9	10
Boosting (Round 1)	7	3	2	8	7	9	4	10	6	3
Boosting (Round 2)	5	4	9	4	2	5	1	7	4	2
Boosting (Round 3)	4	4	8	10	4	5	4	6	3	4

- Example 4 is hard to classify
- Its weight is increased, therefore it is more likely to be chosen again in subsequent rounds

Boosting Intuition

- We adaptively weight 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.
- We stop by monitoring a hold out set.

Boosting in a Picture



Ensemble Methods

- Overview and rationale
- Binomial Distribution
- Bagging
- Random Forests
- Boosting
- **AdaBoost**

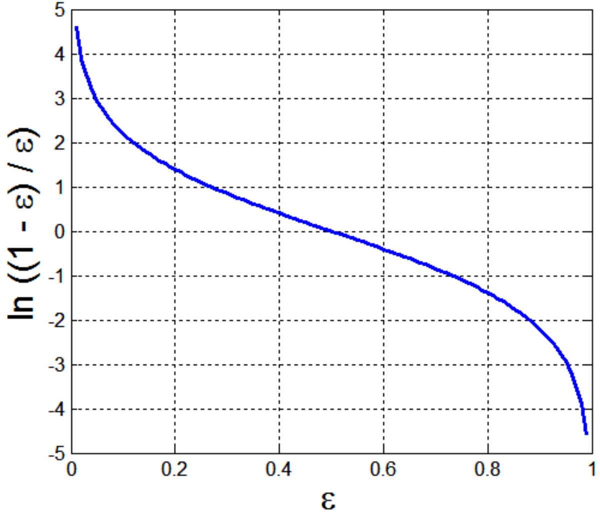
AdaBoost (Adaptive Boosting)

- Base classifiers: $C_1, C_2, \ldots, C_T$
- Error rate [Weighted loss function]:

$$\varepsilon_i = \frac{1}{N} \sum_{j=1}^{N} w_j \delta\left(C_i(x_j) \neq y_j\right)$$

- Importance of a classifier:

$$\alpha_i = \frac{1}{2} \ln\left(\frac{1-\varepsilon_i}{\varepsilon_i}\right)$$



AdaBoost

- Weight update: $$w_i^{(j+1)} = \frac{w_i^{(j)}}{Z_j} \begin{cases} \exp^{-\alpha_j} & \text{if } C_j(x_i) = y_i \\ \exp^{\alpha_j} & \text{if } C_j(x_i) \neq y_i \end{cases}$$

  where Z_j is the normalization factor

- If any intermediate rounds produce error rate higher than 50%, the weights are reverted back to 1/n and the resampling procedure is repeated.

- Classification: $$C^*(x) = \arg\max_y \sum_{j=1}^{T} \alpha_j \delta\left(C_j(x) = y\right)$$

Common Pitfall

A Random Forest and a Boosted Decision Tree are **not** the same

QUESTIONS