Tree-based models

Jesús Calderón

Learning Objectives

By the end of this lecture, students should be able to:

- Describe the basic methods for constructing decision trees.
- Explain and implement three methods of tree regularization or pruning.
- Explain and implement three ensemble methods: bagging, boosting, and random forests.

Decision Trees for Classification

Recursion: A Definition

- Recursion is a technique by which a problem is solved by solving the same problem at a smaller scale. For example, calculating n!:

$$n! = n \times (n-1) \times (n-2) \times \ldots \times 2 \times 1$$

- Notice that
  - 3! = 6
  - 4! = 24
  - $4! = 4 \times 3!$

- In general, $n! = n \times (n-1)!$

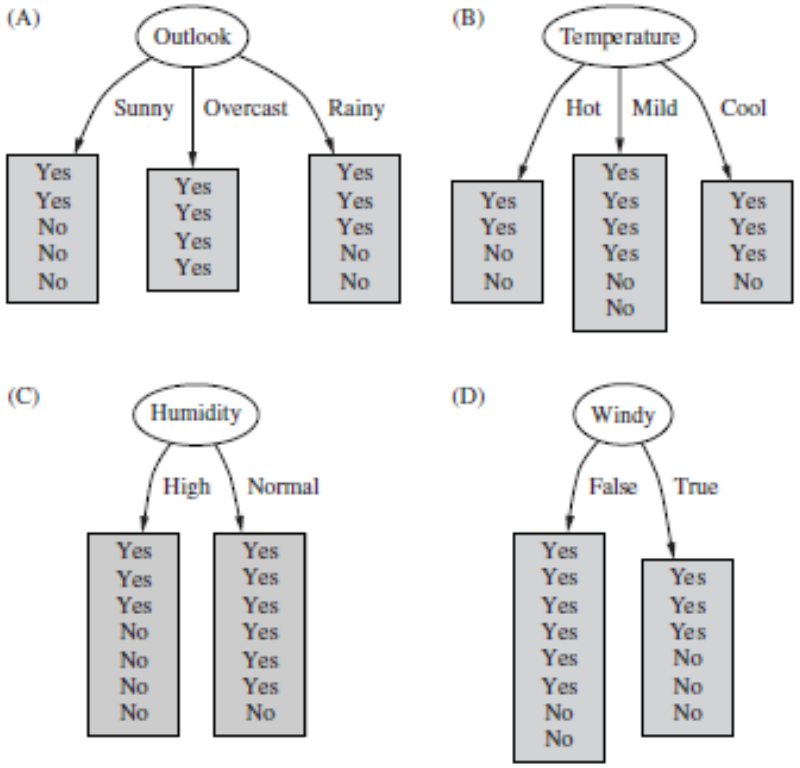
Decision Trees

- Decision Trees are recursive methods that can be applied to classification and regression problems.
- To build a decision tree classifier:
  - Select an attribute to place at the root of the tree.
  - Make one branch for each possible value. This splits up the example set into subsets.
  - Repeat the process recursively for each branch, using only those instances that only reach the branch.
  - If at any time all instances at a node have the same classification, stop.

A Simple Example: Weather Data (Witten et al, 2017)

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

Which Dimension Should We Use to Partition the Data?



Partitions using different dimensions on the Weather Data. The objective is to predict the variable 'Play' (Witten, et al., 2017)

Information

- We would like to reach a useful classification as fast as possible. Therefore we would like to find a shallow or short tree.
- From a classification perspective, we want to increase the *purity* of each subset at each partition.
- Therefore, we should choose the partition that increases the *purity* of the resulting subsets as much as possible.
- The measure of *purity* that we will use is called the *information*, and it is measured in *bits* (similar to the ones used in a computer's representation of numbers.)

Information Gain

- Assume that we can measure *Information*.
- We start with the total data set, where we have nine Yes and five No values. In this case,

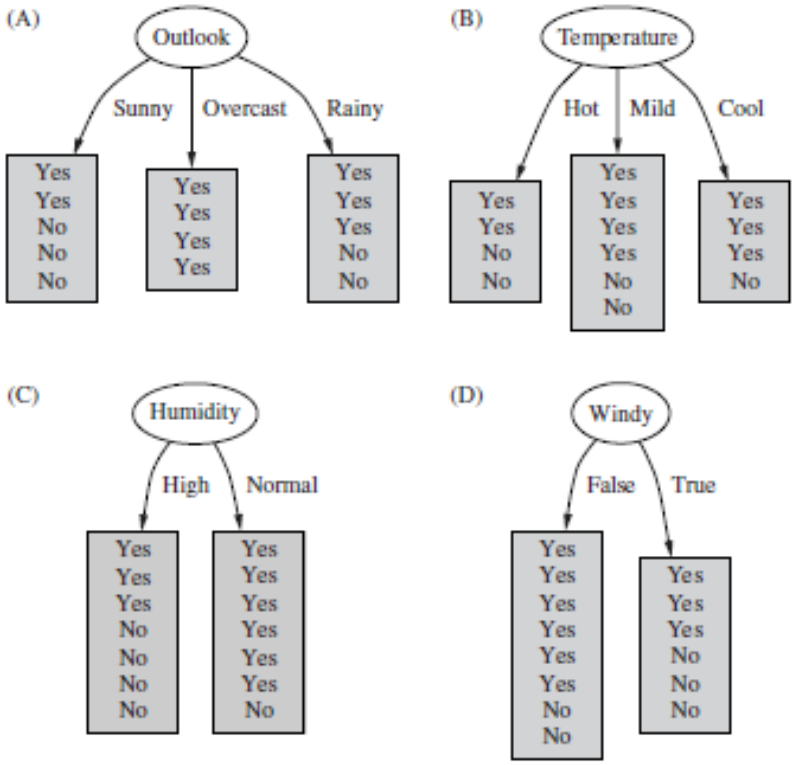
$$Info(9,5) = 0.94bits.$$

- For option A, partitioning on Outlook, we obtain:

$$Info_A = \frac{5}{14} Info(2,3) + \frac{4}{14} Info(4,0) + \frac{5}{14} Info(3,2)$$

- We can calculate:
  - $Info(2,3) = 0.971$
  - $Info(4,0) = 0$
  - $Info(3,2) = 0.971$
- Therefore, the information gain of Option A is: 0.94 - 0.693 = 0.247

Maximizing Information Gain

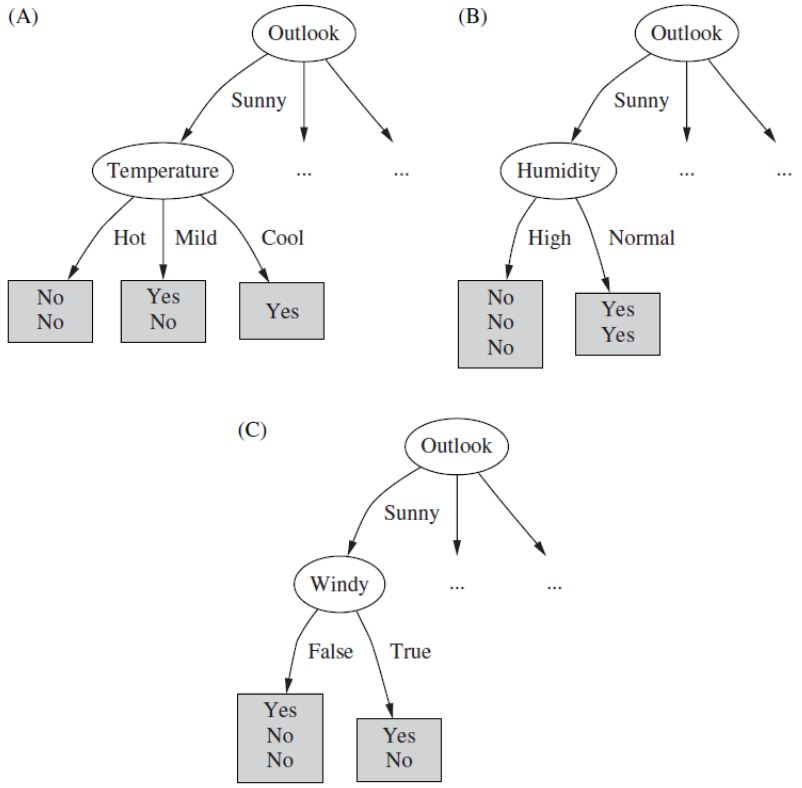


(Witten et al, 2017)

- $Gain(outlook) = 0.247$ bits
- $Gain(temperature) = 0.029$ bits
- $Gain(humidty) = 0.152$ bits
- $Gain(windy) = 0.048$ bits

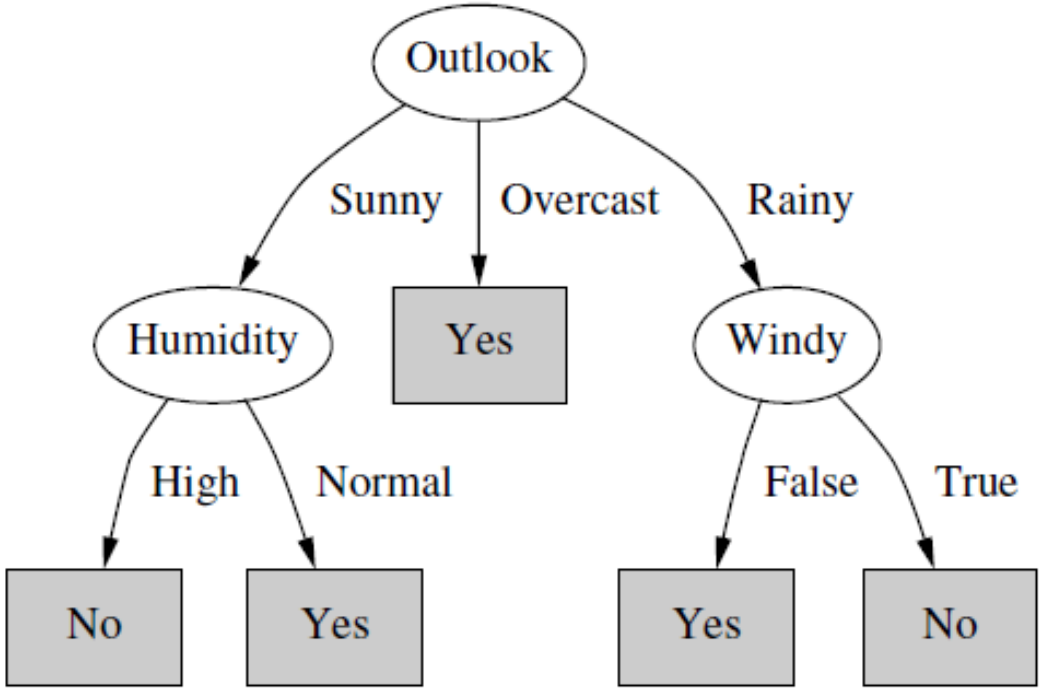
Decision Tree

- We select *Outlook* as the root of the tree. Intuitively, it is the only choice for which a child node is entirely pure.
- We continue *recursively* (splitting on *Outlook* produces nothing new):
  - $Gain(temperature) = 0.571$ bits
  - $Gain(humidity) = 0.971$ bits
  - $Gain(windy) = 0.020$ bits
- A continued application of this process terminates when all leaf nodes are *pure*.



(Witten et al. , 2017)

Decision Tree for Weather Data



(Witten et al., 2017)

Calculating Information

Desirable properties

- When the number of Yes or No is zero, then information is zero.
- When the number of Yes and No is equal, then information is maximum.
- Multistage calculation:

$$Info(2,3,4) = Info(2,7) + \frac{7}{9}Info(3,4)$$

- There is one function that satisfies these properties. It is called entropy:

$$Entropy(p_1, p_2, \ldots, p_n) = -p_1 log p_1 - p_2 log p_2 \ldots - p_n log p_n$$

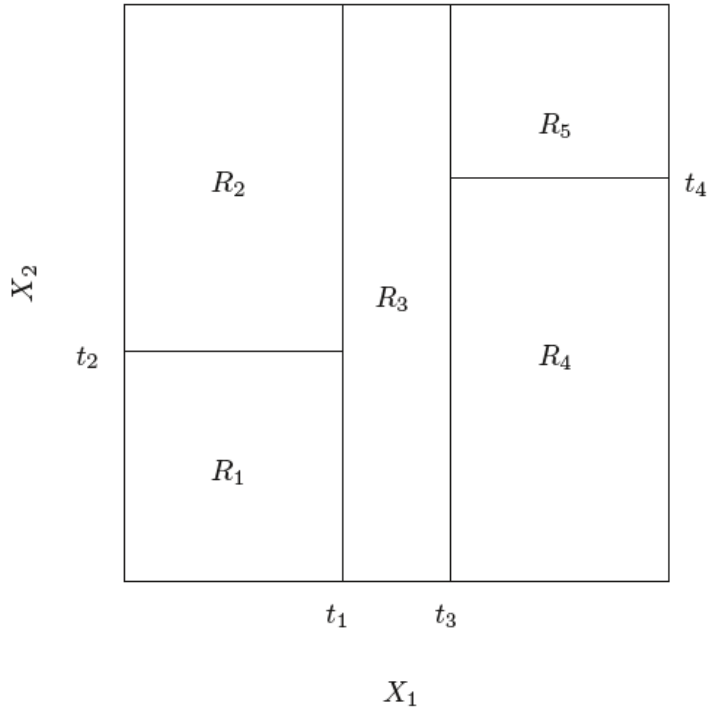
- Usually, the logarithm is base 2, and the resulting bits are similar to the ones used by computers.

Regression Trees

Building a Regression Tree

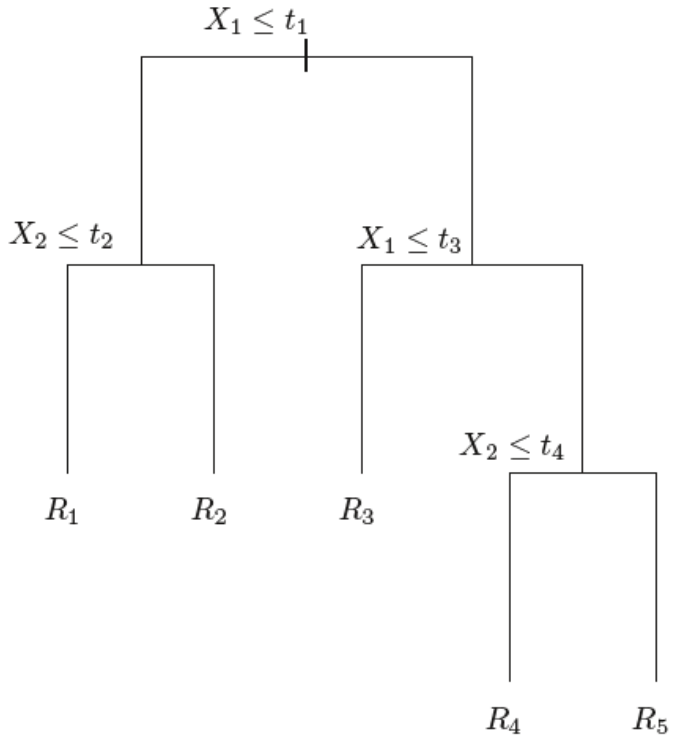
There are two steps:

- Divide the predictor space (the possible values for the predictors) into non-overlapping regions, $R_1, R_2, \ldots R_J$). The regions are determined by *recursive binary splitting*, a top-down and greedy approach:
  - Each region is an n-dimensional dimensional *box*.
  - Consider all possible predictors and all possible cut-off points: find the predictor and cut-off point that produce a split that minimizes error.
- For each observation in a region R_j, predict the response as the average value of all responses for observations in R_j.

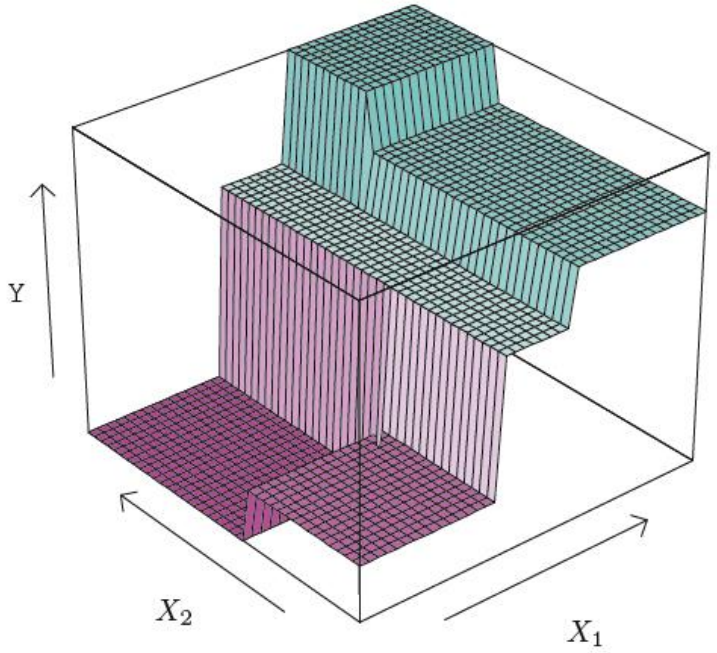


An example of recursive binary splitting (James et al., 2017)

Regression Trees



An example of a Decision/Regression Tree (James et al., 2017)



Predictions from a Regression Tree (James et al., 2017)

Regularizing Decision Trees

Regularization or Tree Pruning

- As described, a decision tree model will produce good predictions on the training set but will most likely overfit the data.
- A tree with fewer splits (fewer regions) may lead to lower variance and better interpretation, at the cost of some bias.
- Regularization in decision trees can be performed by three methods:
- Set a maximum tree depth and limit the maximum number of levels of the tree.
- Set a minimum number of observations per terminal or leaf nodes.
- Tree Pruning:
  - A better strategy may be to learn a large tree and then prune it by identifying branches that do not add much predictive ability and "pruning" them.
  - Each time that a branch is added, the cost of adding that branch is calculated as $\alpha|T|$, where α is a regularization parameter and |T| is the total number of terminal nodes.

R Implementation

- In R, the implementation in library `parsnip` is given by decision_tree(). It has three arguments that control regularization:
  - `cost_complexity`: The cost/complexity parameter (a.k.a. Cp) used by CART models (rpart only).
  - `tree_depth`: The maximum depth of a tree (rpart and spark only).
  - `min_n`: The minimum number of data points in a node that is required for the node to be split further.

Decision Tree Advantages

Advantages (James et al, 2017)

- Easy to explain: easier to explain than linear regression.
- Intuitiveness: some authors believe that decision trees more closely mirror human decision-making than do the regression and classification approaches seen before (linear regression and k-nn).
- Trees can be displayed graphically and are easily interpreted even by a non-expert (especially if they are small).
- Trees can easily handle qualitative predictors without the need to create dummy variables.

Disadvantages

- Unfortunately, trees generally do not have great predictive accuracy as some other regression and classification models.
- Additionally, trees can be very brittle or non-robust. In other words, a small change in the data can cause a large change in the final estimated tree.

Ensemble Methods

Ensemble Methods

"Aggregating the judgment of many consistently beats the accuracy of the average member of the group and is often as startlingly accurate [...] In fact, in any group there are likely to be individuals who beat the group. But those bull's-eye guesses typically say more about the power of luck [...] than about the skill of the guesser. That becomes clear when the exercise is repeated many times."

(Tetlock and Gardner, 2015)

Bagging: Bootstrap Aggregation

- Ensemble methods combine predictions of different models.
- To combine predictions means amalgamating the various outputs into a single prediction.
- The simplest ways:
  - Regression: get the mean or median.
  - Classification: vote.

Bagging Algorithm (N observations in data set)

- Model Training (Bootstrap): For each of t iterations
  - Sample N instances with replacement from training data.
  - Apply the learning method (Decision Tree, DT) to the sample.
  - Store the resulting model.
- Prediction (Aggregation): For each of the t models
  - Predict class of the instance using model.
- Return class that has been predicted more often (classification) or the average prediction (regression).

Bagging Algorithm: Pros and Cons

Advantages

- Bagging helps most when algorithms are unstable.
- Combining multiple models only helps when these models are significantly different from one another, and each one treats a reasonable percentage of the data correctly.
- Sometimes switching pruning off helps in enhancing bagging algorithms.
- Strong choice for cost-sensitive predictions.

Disadvantages

- Bagging models are challenging to explain.

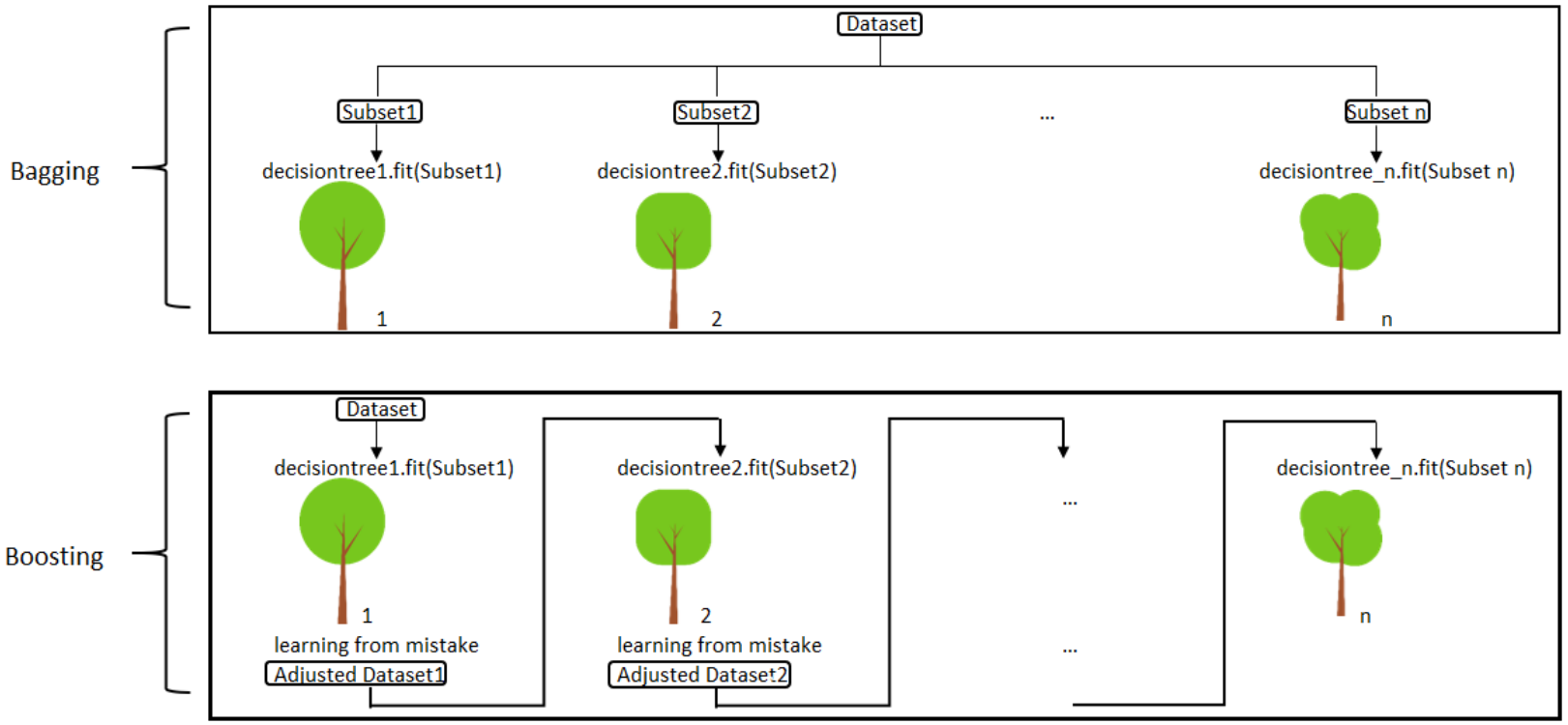
Boosting

- Boosting seeks models that specialize in specific regions, such that models complement each others' strengths or specializations.
- Like bagging, boosting combines models of the same type (e.g., DT).
- Boosting is iterative: it encourages new models to specialize in instances handled incorrectly by earlier ones.
- Boosting weights a model's contribution by its performance, rather than giving equal weights to all models.

Boosting Algorithm

- Model training
  - Assign an equal weight to each training instance.
  - For each of t iterations:
    - Apply learning algorithm to weighted data set and store resulting model.
    - Compute error e of model on weighted dataset and store error.
    - If e equal to zero, or e greater or equal to 0.5:
      - Terminate model generation.
  - For each instance in dataset:
    - If instance classified correctly by model:
      - Multiply weight of instance by e/(1-e).
  - Normalize weight of all instances.
- Classification:
  - Assign weight of zero to all classes.
  - For each of the t (or less) models:
    - Add -log(error/(1-error)) to weight of model class prediction.
  - Return class with highest weight or weighted average predictions.

Bagging and Boosting

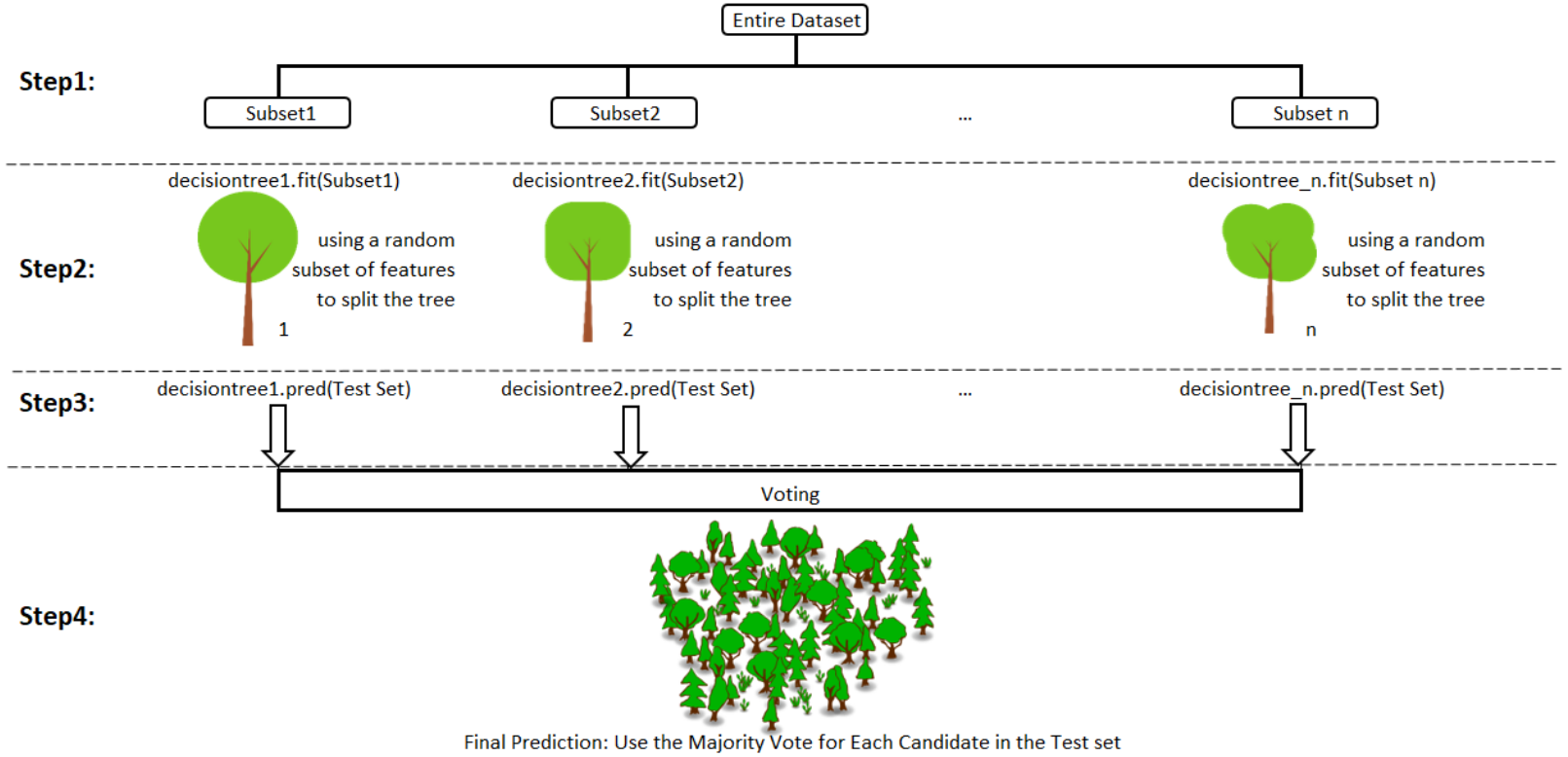


Bagging and Boosting Algorithms (Chen, 2019)

Random Forests

- Inducing randomization to the learning process can increase generalization.
- Randomization can enhance stable learners (knn) and unstable learners (DT): the trick is to randomize in a way that makes the classifiers diverse without sacrificing too much performance.
- In random forests, randomization is performed on the sample selection (bootstrapping) and in the selection of attributes.
- Random forests do not require any modification to the learning algorithm.

Random Forest



Random Forest Learning model (Chen, 2019)

References

References

- Chen, L. *Basic Ensemble Methods*. 2019: URL
- James, G., D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning with Applications in R*. United States: Springer, 2017.
- Tetlock, P. and D. Gardner. *Superforcasting: The Art and Science of Prediction*. United States: Crown Publishers, 2015.
- Witten, F., E. Frank, M. Hall, C. Pal. *Data Mining: Practical Machine Learning Tools and Techniques*. United States: Morgan Kaufmann, 2017. 4th Edition.