Algorithms

Decision trees and random forests

Emanuele Rodolà
rodola@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

2nd semester a.y. 2020/2021 · April 28, 2021

Learning from data

We are looking at a family of techniques known as supervised learning.

Learning from data

We are looking at a family of techniques known as supervised learning.

- Classification

  The output is categorical (yes/no, animal/human/plant, etc.).

  Decision trees and random forests.

Learning from data

We are looking at a family of techniques known as supervised learning.

- Classification

  The output is categorical (yes/no, animal/human/plant, etc.).

  Decision trees and random forests.

- Regression

  The output is a continuous value (quantities, temperature, etc.).

  Regression forests.

Learning from data

We are looking at a family of techniques known as supervised learning.

- Classification

  The output is categorical (yes/no, animal/human/plant, etc.).

  Decision trees and random forests.

- Regression

  The output is a continuous value (quantities, temperature, etc.).

  Regression forests.

Overall idea: Given a large dataset where each data point also has a label, construct a machine that can predict the label of a new data point.

Learning from data

We are looking at a family of techniques known as supervised learning.

- Classification

  The output is categorical (yes/no, animal/human/plant, etc.).

  Decision trees and random forests.

- Regression

  The output is a continuous value (quantities, temperature, etc.).

  Regression forests.

Overall idea: Given a large dataset where each data point also has a label, construct a machine that can predict the label of a new data point.

- Training set: data points with “ground-truth” labels.

Learning from data

We are looking at a family of techniques known as supervised learning.

- Classification

  The output is categorical (yes/no, animal/human/plant, etc.).

  Decision trees and random forests.

- Regression

  The output is a continuous value (quantities, temperature, etc.).

  Regression forests.

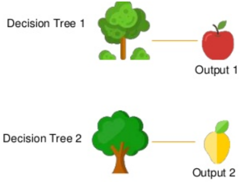
Overall idea: Given a large dataset where each data point also has a label, construct a machine that can predict the label of a new data point.

- Training set: data points with "ground-truth" labels.
- Test set: data points with unknown labels.

Random forest: Intuition

Figures by Richard Kershner

Random forest: Intuition

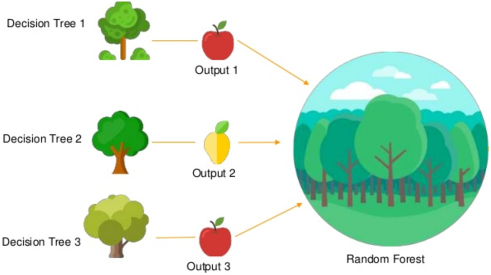


Construct multiple decision trees.

Each decision tree outputs a prediction for a given input.

Figures by Richard Kershner

Random forest: Intuition

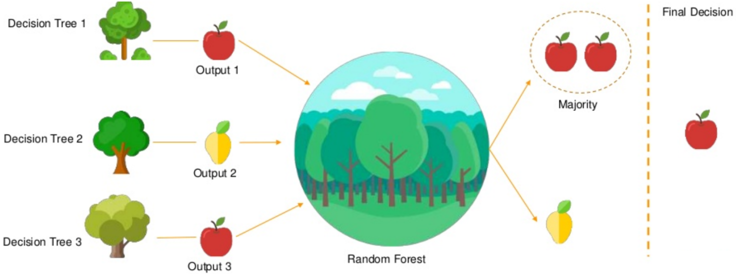


Construct multiple decision trees.

Each decision tree outputs a prediction for a given input.

Figures by Richard Kershner

Random forest: Intuition



Construct multiple decision trees.

Each decision tree outputs a prediction for a given input.

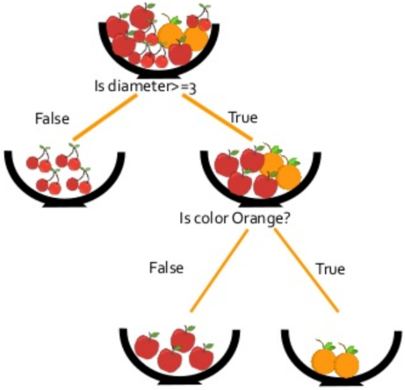
The predictions from each tree are combined into one final prediction.

In the example, the final decision is "apple with probability 66%".

Figures by Richard Kershner

Decision tree

A decision tree is a binary tree in which each branch represents a possible decision. A path through the tree is therefore a course of action.



The data is iteratively partitioned into subsets from the root to the leaves.

Information gain

Each intermediate node is called a decision node.

Each leaf node stores the final decision (i.e. the classification result).

Information gain

Each intermediate node is called a decision node.

Each leaf node stores the final decision (i.e. the classification result).

We use entropy as a measure to choose what is done at each node.

Entropy measures the uncertainty for a given data distribution.

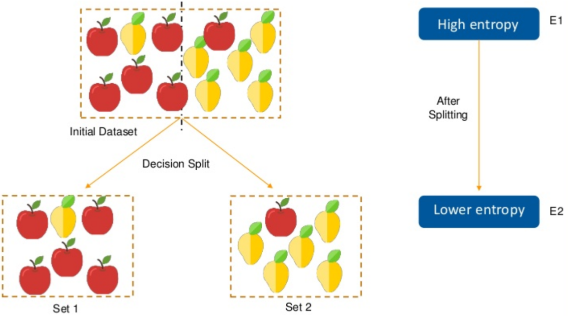
Information gain

Each intermediate node is called a decision node.

Each leaf node stores the final decision (i.e. the classification result).

We use entropy as a measure to choose what is done at each node.

Entropy measures the uncertainty for a given data distribution.



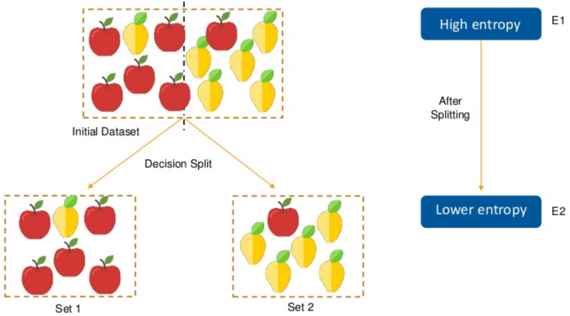
Information gain

Each intermediate node is called a decision node.

Each leaf node stores the final decision (i.e. the classification result).

We use entropy as a measure to choose what is done at each node.

Entropy measures the uncertainty for a given data distribution.



Ideally we want each split to be as discriminative as possible, and thus maximize the information gain $E2 - E1$.

Training

Problem: Classify the types of fruit based on their features.

We start from a set of data points that have high entropy.

Training

Problem: Classify the types of fruit based on their features.

We start from a set of data points that have high entropy.

Define the features (diameter, color, etc.) on which to base our decisions.

Training

Problem: Classify the types of fruit based on their features.

We start from a set of data points that have high entropy.

Define the features (diameter, color, etc.) on which to base our decisions.

At each node, choose the question that maximizes the information gain.

Training

Problem: Classify the types of fruit based on their features.

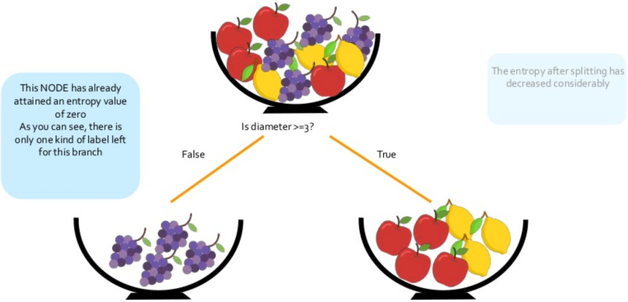
We start from a set of data points that have high entropy.

Define the features (diameter, color, etc.) on which to base our decisions.

At each node, choose the question that maximizes the information gain.

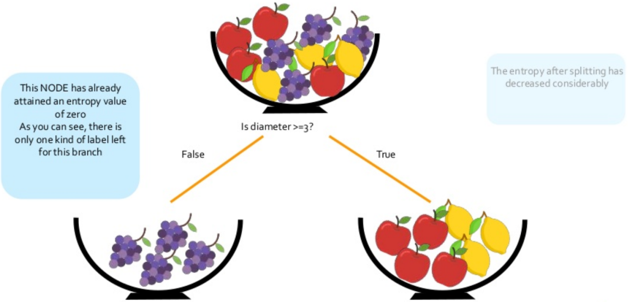
Keep splitting until the dataset is empty or accuracy is high enough.

Accuracy



A leaf is where no more splitting is required or possible (zero entropy).

Accuracy

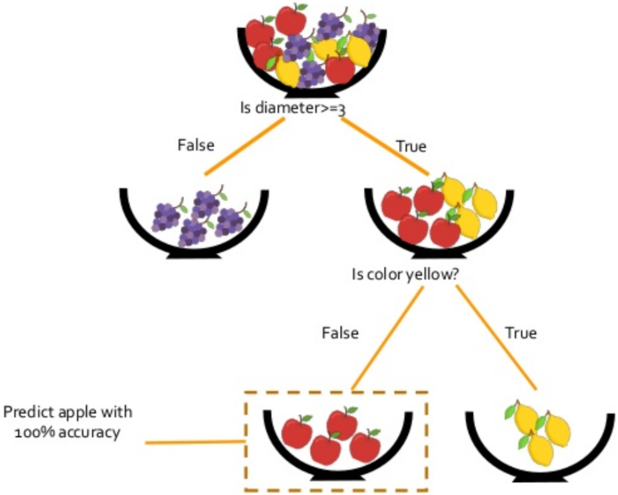


A leaf is where no more splitting is required or possible (zero entropy).

At each leaf, accuracy with respect to label ℓ is measured as:

$$\frac{\#\text{ data points with label }\ell}{\#\text{ data points}}$$

Accuracy



Is diameter>=3
False
True
Is color yellow?
False
True
Predict apple with
100% accuracy

Random forest

The pipeline is as follows:

- Split the given training data into random partitions (bagging).

Breiman, “Random forests”, 2001

Random forest

The pipeline is as follows:

- Split the given training data into random partitions (bagging).
- With each partition, construct a decision tree.

Breiman, "Random forests", 2001

Random forest

The pipeline is as follows:

- Split the given training data into random partitions (bagging).
- With each partition, construct a decision tree.

  Each tree asks different questions, has different depth, and different accuracy at the leaves.

Breiman, "Random forests", 2001

Random forest

The pipeline is as follows:

- Split the given training data into random partitions (bagging).
- With each partition, construct a decision tree.

  Each tree asks different questions, has different depth, and different accuracy at the leaves.

  However, all the trees try to solve the same classification problem.

Breiman, "Random forests", 2001

Random forest

The pipeline is as follows:

- Split the given training data into random partitions (bagging).
- With each partition, construct a decision tree.

  Each tree asks different questions, has different depth, and different accuracy at the leaves.

  However, all the trees try to solve the same classification problem.
- Given a new data point (test set), route it through each tree.

Breiman, "Random forests", 2001

Random forest

The pipeline is as follows:

- Split the given training data into random partitions (bagging).
- With each partition, construct a decision tree.

  Each tree asks different questions, has different depth, and different accuracy at the leaves.

  However, all the trees try to solve the same classification problem.
- Given a new data point (test set), route it through each tree.

  Each tree gives a prediction (a probability distribution over all the classes).

Breiman, "Random forests", 2001

Random forest

The pipeline is as follows:

- Split the given training data into random partitions (bagging).
- With each partition, construct a decision tree.

  Each tree asks different questions, has different depth, and different accuracy at the leaves.

  However, all the trees try to solve the same classification problem.
- Given a new data point (test set), route it through each tree.

  Each tree gives a prediction (a probability distribution over all the classes).

  Average the tree predictions to get the forest prediction.

Breiman, "Random forests", 2001

Example

In scikit-learn:

`https://scikit-learn.org/stable/modules/ensemble.html`

Exercise: Use the iris dataset to predict flower species.

Reading: "Understanding random forests: from theory to practice"

`https://arxiv.org/pdf/1407.7502`