SUS 2020
Lecture 2: KNN and Decision Trees

Hung Son Nguyen

MIM
University of Warsaw

Based on the slides published by Benjamin M. Marlin (marlin@cs.umass.edu).
Created with support from National Science Foundation Award# IIS-1350522.

Outline

Views on Machine Learning

Definition:
computational methods using experience to improve performance.

Experience:
data-driven task, thus statistics, probability, and optimization.

Computer science:
learning algorithms, analysis of complexity, theoretical guarantees.

Example:
use document word counts to predict its topic.

Machine Learning Tasks

Classification: assign a category to each item (e.g., document classification).

Regression: predict a real value for each item (prediction of stock values, economic variables).

Ranking: order items according to some criterion (relevant web pages returned by a search engine).

Clustering: partition data into 'homogenous' regions (analysis of very large data sets).

Dimensionality reduction: find lower-dimensional manifold preserving some properties of the data.

Example of Learning Tasks

- **Text:** document classification, spam detection.
- **Language:** NLP tasks (e.g., morphological analysis, POS tagging, context-free parsing, dependency parsing).
- **Speech:** recognition, synthesis, verification.
- **Image:** annotation, face recognition, OCR, handwriting recognition.
- Games (e.g., chess, backgammon).
- Unassisted control of vehicles (robots, car).
- Medical diagnosis, fraud detection, network intrusion.

General Objective of Machine Learning

Theoretical questions:

- what can be learned, under what conditions?
- are there learning guarantees?
- analysis of learning algorithms.

Algorithms:

- more efficient and more accurate algorithms.
- deal with large-scale problems.
- handle a variety of different learning problems.

Learning problem

- **Spaces:** input space $\mathcal{X}$, output space $\mathcal{Y}$.
- **Loss function:** $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.
  - $\mathcal{L}(\hat{y}, y)$: cost of predicting $\hat{y}$ instead of y.
  - binary classification: 0-1 loss, $\mathcal{L}(\hat{y}, y) = 1_{\hat{y} \neq y}$.
  - regression ($\mathcal{Y} \subseteq \mathbb{R}$) : $\mathcal{L}(\hat{y}, y) = (\hat{y} - y)^2$
- **Hypothesis set:** $H \subseteq \mathcal{Y}^{\mathcal{X}}$, subset of functions out of which the learner selects his hypothesis.

Supervised Learning Set-Up

- Training data: sample of size m drawn i.i.d. from $\mathcal{X} \times \mathcal{Y}$ according to distribution D:

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$$

- Problem: find hypothesis $h \in H$ with small generalization error.
  - deterministic case: output label deterministic function of input, $y = f(\mathbf{x})$.
  - stochastic case: output probabilistic function of input.

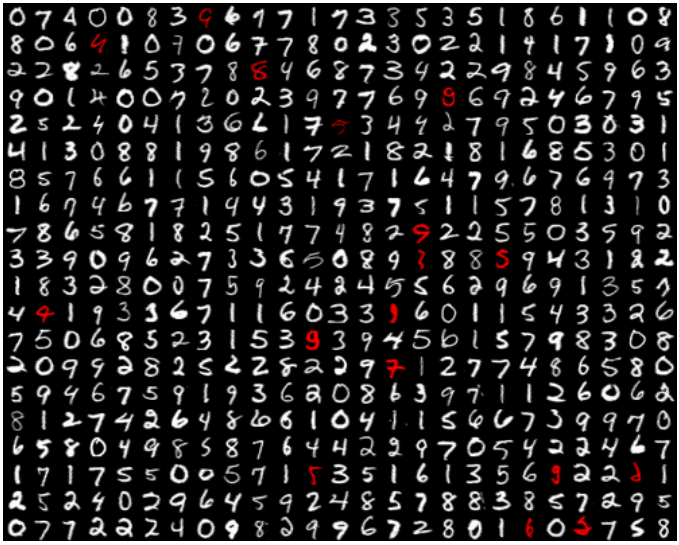
Outline

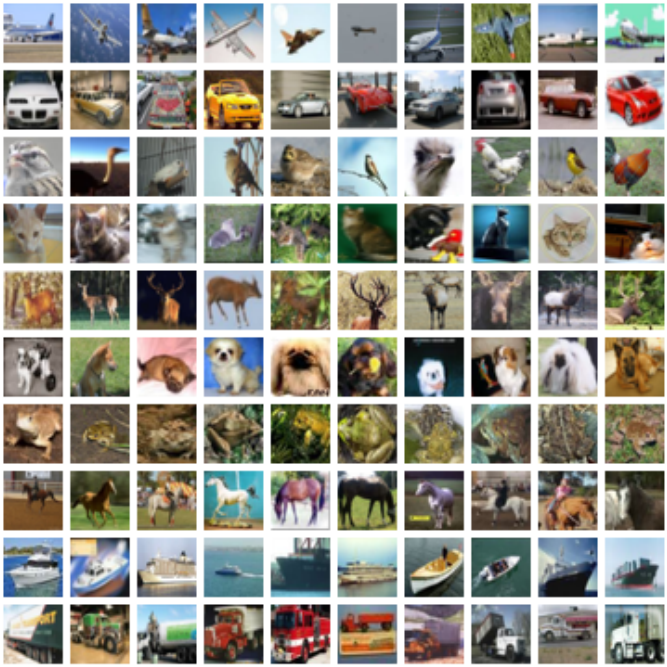
The Classification Task

Definition: The Classification Task

Given a feature vector $\mathbf{x} \in \mathbb{R}^D$ that describes an object that belongs to one of C classes from the set $\mathcal{Y}$, predict which class the object belongs to.

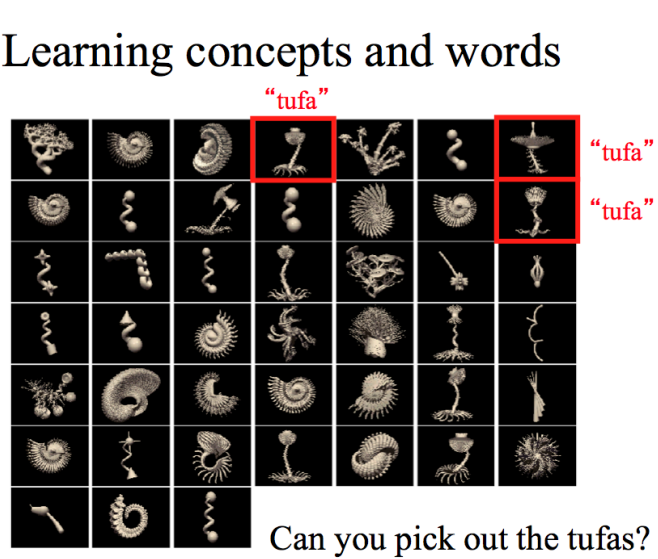
Example: Digits





Example: Natural Images

Example: Synthetic Images



The Classifier Learning Problem

Definition: Classifier Learning

Given a data set of example pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1 : N\}$ where $\mathbf{x}_i \in \mathbb{R}^D$ is a feature vector and $y_i \in \mathcal{Y}$ is a class label, learn a function $f : \mathbb{R}^D \to \mathcal{Y}$ that accurately predicts the class label y for any feature vector $\mathbf{x}$.

Classification Error and Accuracy

Definition: Classification Error Rate

Given a data set of example pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1 : N\}$ and a function $f : \mathbb{R}^D \to \mathcal{Y}$, the classification error rate of f on $\mathcal{D}$ is:

$$Err(f, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(y_i \neq f(\mathbf{x}_i))$$

Definition: Classification Accuracy Rate

Given a data set of example pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1 : N\}$ and a function $f : \mathbb{R}^D \to \mathcal{Y}$, the classification accuracy rate of f on $\mathcal{D}$ is:

$$Acc(f, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(y_i = f(\mathbf{x}_i))$$

Terminology

Example: item, object, case, instance of the data used.

Features: attributes associated to an item, often represented as a vector (e.g., word counts).

Labels: category (classification) or real value (regression) associated to an item.

Data:
- training data (typically labeled).
- test data (labeled but labels not seen).
- validation data (labeled, for tuning parameters).

batch: learner receives full (training) sample, which he uses to make predictions for unseen points.

on-line: learner receives one sample at a time and makes a prediction for that sample.

Outline

K Nearest Neighbors Classification

The KNN classifier is a non-parametric classifier that simply stores the training data $\mathcal{D}$ and classifies each new instance $\mathbf{x}$ using a majority vote over its' set of K nearest neighbors $\mathcal{N}_K(\mathbf{x})$ computed using any distance function $d : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$.

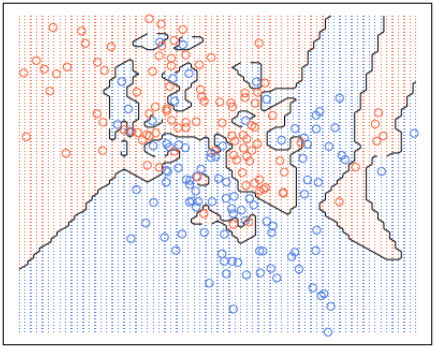
KNN Classification Function

$$f_{KNN}(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} \sum_{i \in \mathcal{N}_K(\mathbf{x})} \mathbb{I}[y_i = y]$$

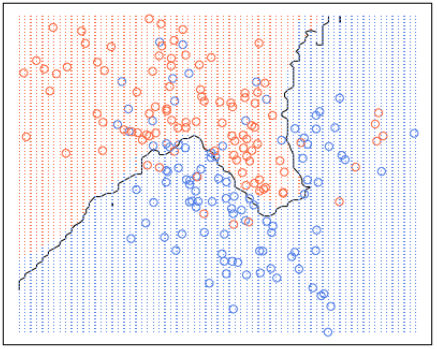
Use of KNN requires choosing the distance function d and the number of neighbors K.

Illustration

nearest neighbour (k = 1)



20-nearest neighbour



Distance and Similarity

- In general, KNN can work with any distance function d satisfying non-negativity $d(\mathbf{x}, \mathbf{x}') \geq 0$ and identity of indiscernibles $d(\mathbf{x}, \mathbf{x}) = 0$.
- Alternatively, KNN can work with any similarity function s satisfying non-negativity $s(\mathbf{x}, \mathbf{y}) \geq 0$ that attains it's maximum on indiscernibles $s(\mathbf{x}, \mathbf{x}) = \max_{\mathbf{x}'} s(\mathbf{x}, \mathbf{x}')$.
- However, the more structure the distance or similarity function has (symmetry, triangle inequality), the more structure you can exploit when designing algorithms.

Distance Metrics

Definition: Minkowski Distance (ℓ_p norms)

Given two data vectors $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^D$, the Minkowski Distance with parameter p (the ℓ_p norm) is a proper metric defined as follows:

$$\begin{aligned} d_p(\mathbf{x}, \mathbf{x}') &= ||\mathbf{x} - \mathbf{x}'||_p \\ &= \left(\sum_{i=1}^{D} |x_d - x'_d|^p \right)^{1/p} \end{aligned}$$

Special cases include Euclidean distance ($p = 2$), Manhattan distance ($p = 1$) and Chebyshev distance ($p = \infty$).

Brute Force KNN

- Given any distance function d, brute force KNN works by computing the distance $d_i = d(\mathbf{x}_i, \mathbf{x}_*)$ from a target point $\mathbf{x}_*$ to all of the training points $\mathbf{x}_i$.
- You then simply sort the distances $\{d_i, i = 1 : N\}$ and choose the data cases with the K smallest distances to form the neighbor set $\mathcal{N}_K(\mathbf{x}_*)$. Using a similarity function is identical, but you select the K most similar data cases.
- Once the K neighbors are selected, applying the classification rule is easy.

KNN Variants

- Instead of giving all of the K neighbors equal weight in the majority vote, a distance-weighted majority can be used:

$$f_{KNN}(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} \frac{\sum_{i \in \mathcal{N}_K(\mathbf{x})} w_i \mathbb{I}[y_i = y]}{\sum_{i \in \mathcal{N}_K(\mathbf{x})} w_i}$$

$$w_i = \exp(-\alpha d_i)$$

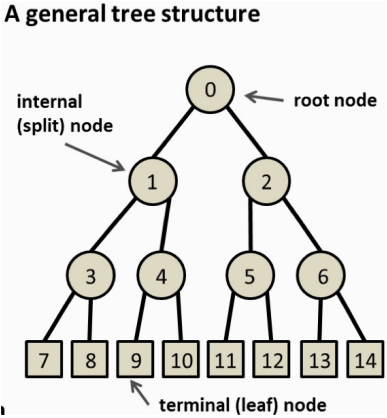
- Instead of a brute force nearest neighbor search, data structures like ball trees can be constructed over the training data that support nearest neighbor search with lower amortized computational complexity.

KNN Trade-Offs

- Low bias: Converges to the correct decision surface as data goes to infinity.
- High variance: Lots of variability in the decision surface when amount of data is low.
- Curse of dimensionality: Everything is far from everything else in high dimensions.
- Space and Time Complexity: Need to store all training data and perform neighbor search can make the method use a lot of memory and take a lot of time.

Outline

Decision Tree

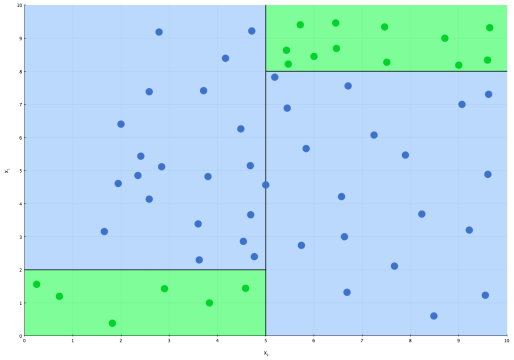


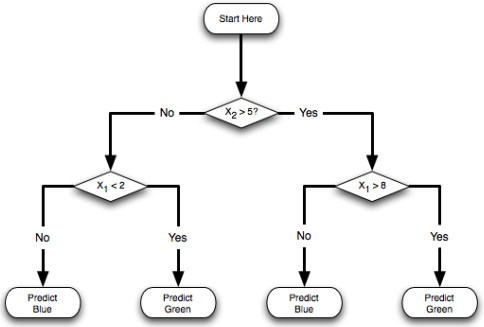
- A classical decision tree classifies data cases using a conjunction of rules organized into a binary tree structure.
- Each internal node in a classical decision tree contains a rule of the form $(x_d < t)$ or $(x_d = t)$ that tests a single data dimension d against a single value t and assigns the data case to it's left or right sub-tree according to the result.
- A data item is routed through the tree from the root to a leaf. Each leaf node is associated with a class, and a data item is assigned the class of the leaf node it is routed to.

Type of trees

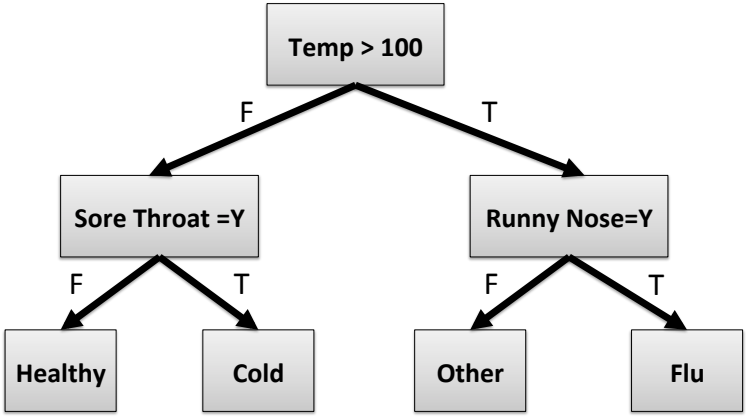
- We'll only consider
  - binary trees (vs multiway trees where nodes can have more than 2 children)
  - decisions at each node involve only a single feature (i.e. input coordinate)
  - for continuous variables, splits always of the form $x_i \leq t$
  - for discrete variables, partitions values into two groups
- Other types of splitting rules
  - oblique decision trees or binary space partition trees (BSP trees) have a linear split at each node
  - sphere trees - space is partitioned by a sphere of a certain radius around a fixed point

Illustration

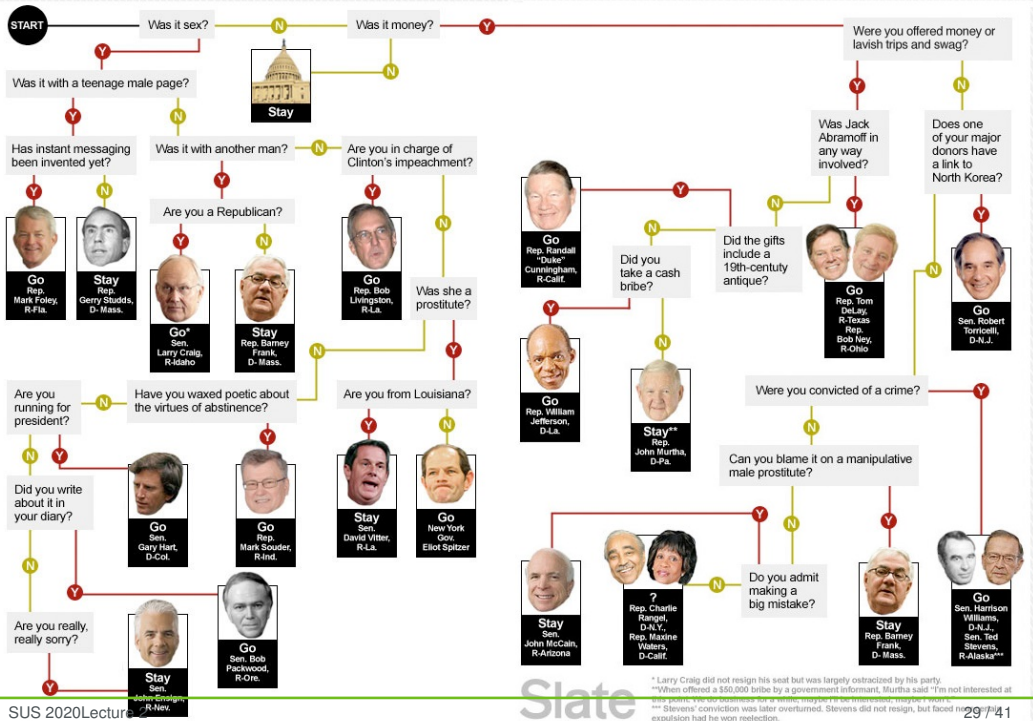




Example: Decision Tree for Flu



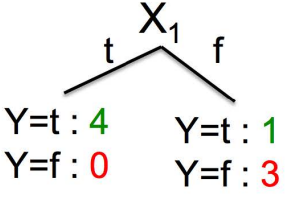
https://alliance.seas.upenn.edu/~cis520/wiki/index.php?n=Lectures.DecisionTrees

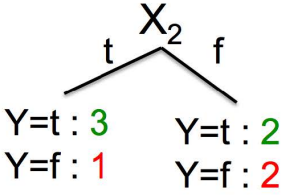

the scandal survivor
START
Was it sex?
Was it money?
Stay
Was it with a teenage male page?
Has instant messaging been invented yet?
Was it with another man?
Are you in charge of Clinton's impeachment?
Are you a Republican?
Was she a prostitute?
Go Rep. Mark Foley, R-Fla.
Stay Rep. Gerry Studds, D- Mass.
Go* Sen. Larry Craig, R-Idaho
Stay Rep. Barney Frank, D- Mass.
Go Rep. Bob Livingston, R-La.
Are you running for president?
Have you waxed poetic about the virtues of abstinence?
Are you from Louisiana?
Did you write about it in your diary?
Go Sen. Gary Hart, D-Col.
Go Rep. Mark Souder, R-Ind.
Stay Sen. David Vitter, R-La.
Go New York Gov. Eliot Spitzer
Are you really, really sorry?
Go Sen. Bob Packwood, R-Ore.
Stay Sen. John Ensign, R-Nev.
the scandal survivor
Were you offered money or lavish trips and swag?
Was Jack Abramoff in any way involved?
Does one of your major donors have a link to North Korea?
Go Rep. Randall "Duke" Cunningham, R-Calif.
Did you take a cash bribe?
Did the gifts include a 19th-centuty antique?
Go Rep. Tom DeLay, R-Texas Rep. Bob Ney, R-Ohio
Go Sen. Robert Torricelli, D-N.J.
Go Rep. William Jefferson, D-La.
Stay** Rep. John Murtha, D-Pa.
Were you convicted of a crime?
Can you blame it on a manipulative male prostitute?
Stay Sen. John McCain, R-Arizona
? Rep. Charlie Rangel, D-N.Y., Rep. Maxine Waters, D-Calif.
Do you admit making a big mistake?
Stay Rep. Barney Frank, D- Mass.
Go Sen. Harrison Williams, D-N.J., Sen. Ted Stevens, R-Alaska***
* Larry Craig did not resign his seat but was largely ostracized by his party.
**When offered a $50,000 bribe by a government informant, Murtha said "I'm not interested at this point. We do business for a while, maybe I'm interested, maybe I won't."
*** Stevens' conviction was later overturned. Stevens did not resign, but faced re-election.
Slate

Decision Tree Learning Algorithm

- Decision trees are learned using recursive greedy algorithms that select the variable and threshold at each node from top to bottom.
- The learning algorithm begins with all data cases at the root of the tree.
- The algorithm selects a variable and a threshold to split on according to a heuristic.
- The algorithm applies the chosen rule and assigns each data case to the left or right sub-tree.

Which split is better?





X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F

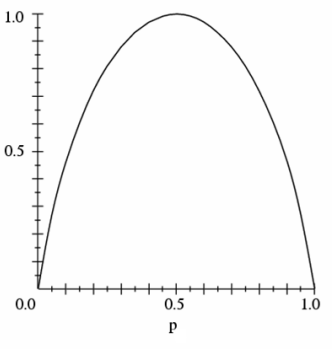
Decision Tree Learning

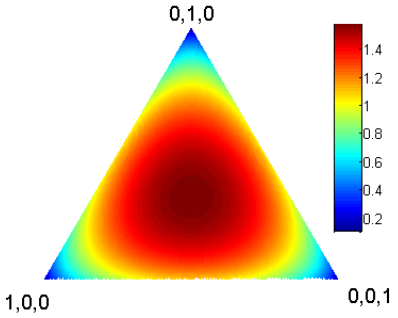
- The two main criteria used to evaluate the the split that results from a potential variable d and threshold t are Gini Impurity (C_{GI}) and information gain (C_{IG}).
- Suppose a given variable d and threshold t result in a distribution of class labels in the proportions $p_1, p_2, ..., p_C$ for the C classes.

$$C_{GI} = \sum_{c=1}^{C} p_i(1 - p_i) \qquad C_{IG} = -\sum_{c=1}^{C} p_i \log(p_i)$$

- The decision tree construction algorithm recursively searches for optimal (variable, threshold) pairs according to a given criteria down to a specified depth.

Entropy





Entropy: $H(Y) = -\sum_{y} P(Y = y) \log_2 P(Y = y)$

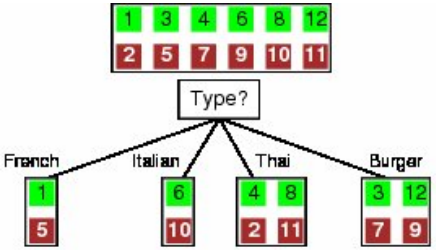
Example	Input Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 = Yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 = No$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 = Yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 = Yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = No$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 = Yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 = No$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 = Yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = No$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$y_{10} = No$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$y_{11} = No$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$y_{12} = Yes$

Attributes:

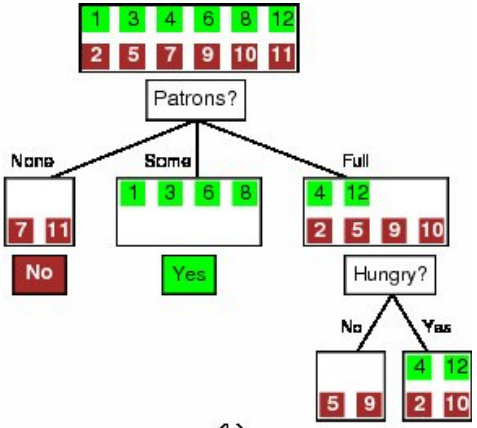
1. Alternate: whether there is a suitable alternative restaurant nearby.
2. Bar: whether the restaurant has a comfortable bar area to wait in.
3. Fri/Sat: true on Fridays and Saturdays.
4. Hungry: whether we are hungry.
5. Patrons: how many people are in the restaurant (values are None, Some, and Full).
6. Price: the restaurant's price range ($, $$, $$$).
7. Raining: whether it is raining outside.
8. Reservation: whether we made a reservation.
9. Type: the kind of restaurant (French, Italian, Thai or Burger).
10. WaitEstimate: the wait estimated by the host (0-10 minutes, 10-30, 30-60, >60).

[from: Russell & Norvig]

Which split is better



(a)



(b)

Split evaluation

- Discernibility:

$$disc(t, X) = conflict(X) - \sum conflict(X_i)$$

- Information gain.

$$Gain(t, X) = Entropy(X) - \sum_i p_i \cdot Entropy(X_i)$$

- gain ratio

$$Gain\_ratio = \frac{Gain(t, X)}{-\sum_{i=1}^{r} \frac{|X_i|}{|X|} \cdot \log \frac{|X_i|}{|X|}}$$

- Other (e.g. Gini's index, test χ^2, ...)

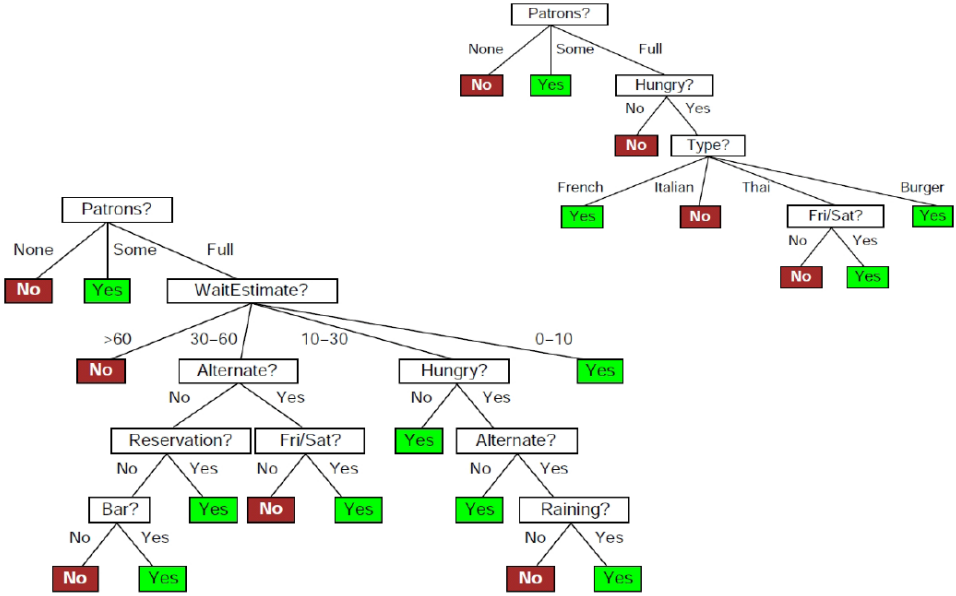
Decision Tree Learning

- The algorithm then recurses on the child nodes until a given stopping condition is satisfied.
- When the stopping condition is satisfied, the current node is a leaf in the tree. It is typically assigned a label that corresponds to the most common label of the data cases it contains.
- The main stopping criteria used are all data cases assigned to a node have the same label, the number of data cases assigned to the node falls below a threshold, and the node is at the maximum allowable depth.
- Given sufficient depth, a decision tree can approximate any classification function to arbitrary accuracy.
- There are a number of heuristics for selecting variables and thresholds to split on. In general, these heuristics are aimed at producing splits of the training data that are as homogeneous as possible in terms of the labels.

Decision Tree Trade-Offs

- Interpretability: The learned model is easy to understand as a collection of rules.
- Test Complexity: Shallow trees can be extremely fast classifiers at test time.
- Train Complexity: Finding optimal trees is NP-complete, thus need for greedy heuristics.
- Representation: Splitting on single variables can require very large trees to accurately model non-axis aligned decision boundaries.

Pruning


Patrons?
None
Some
Full
No
Yes
Hungry?
No
Yes
No
Type?
French
Italian
Thai
Burger
Yes
No
Fri/Sat?
Yes
No
Yes
No
Yes
Patrons?
None
Some
Full
No
Yes
WaitEstimate?
>60
30–60
10–30
0–10
No
Alternate?
Hungry?
Yes
No
Yes
No
Yes
Reservation?
Fri/Sat?
Yes
Alternate?
No
Yes
No
Yes
No
Yes
Bar?
Yes
No
Yes
Yes
Raining?
No
Yes
No
Yes
No
Yes
No
Yes

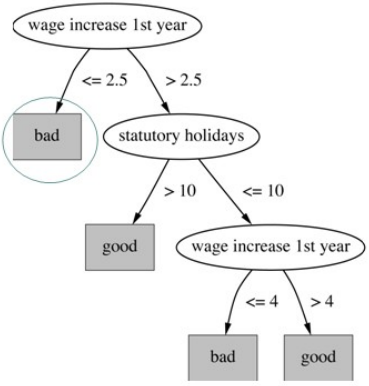
Pruning Criteria

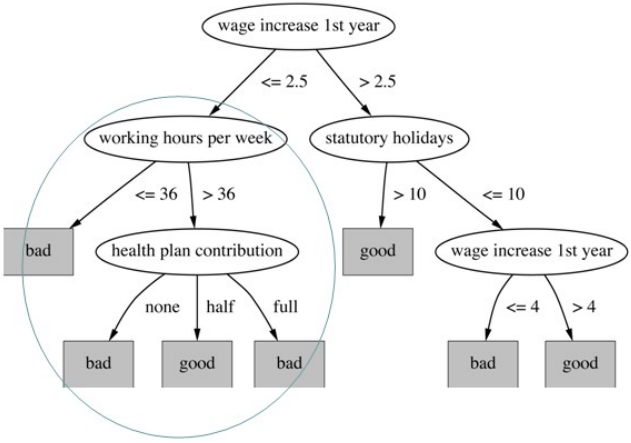
- Niech
  $e_T(l)$ - error of the tree when using leave l,
  $e_T(n)$ - error of the tree when using subtree n.
- n is replaced by l if

$$e_T(l) \leq e_T(n) + \mu \sqrt{\frac{e_T(n)(1 - e_T(n))}{|P_{T,n}|}}$$

  Usually $\mu = 1$.

Example


wage increase 1st year
<= 2.5
> 2.5
bad
statutory holidays
> 10
<= 10
good
wage increase 1st year
<= 4
> 4
bad
good


wage increase 1st year
<= 2.5
> 2.5
working hours per week
statutory holidays
<= 36
> 36
> 10
<= 10
bad
health plan contribution
good
wage increase 1st year
none
half
full
<= 4
> 4
bad
good
bad
bad
good