Decision trees

Martha Lewis
(based on slides from Raúl Santos Rodriguez)

Have a look at ...

... Russell and Norvig (Ch. 18.3)

... Hastie, Tibshirani, Friedman. The elements of statistical learning, (Ch. 9.2)

... Criminisi, Shotton, Konukoglu. Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning. Microsoft research.

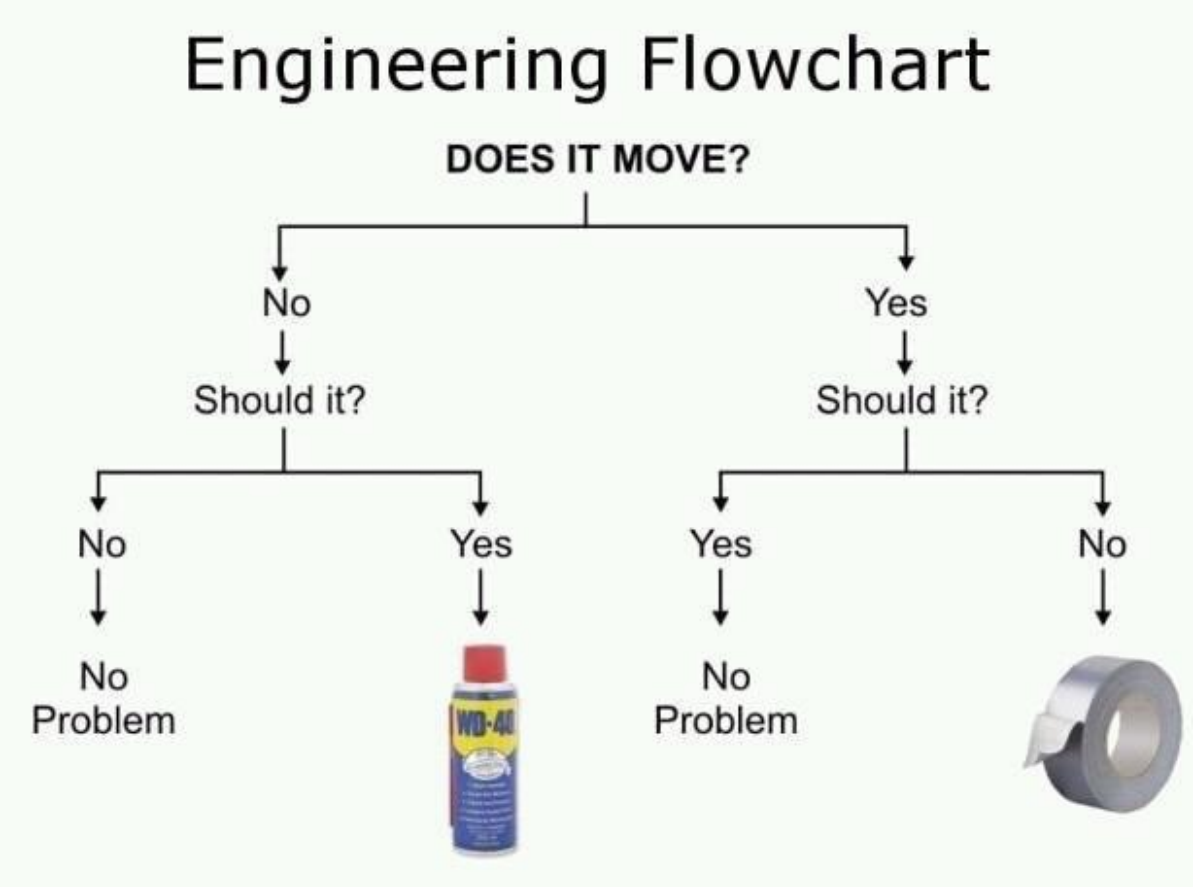
... Python: `http://scikit-learn.org/`

Outline

We will discuss:

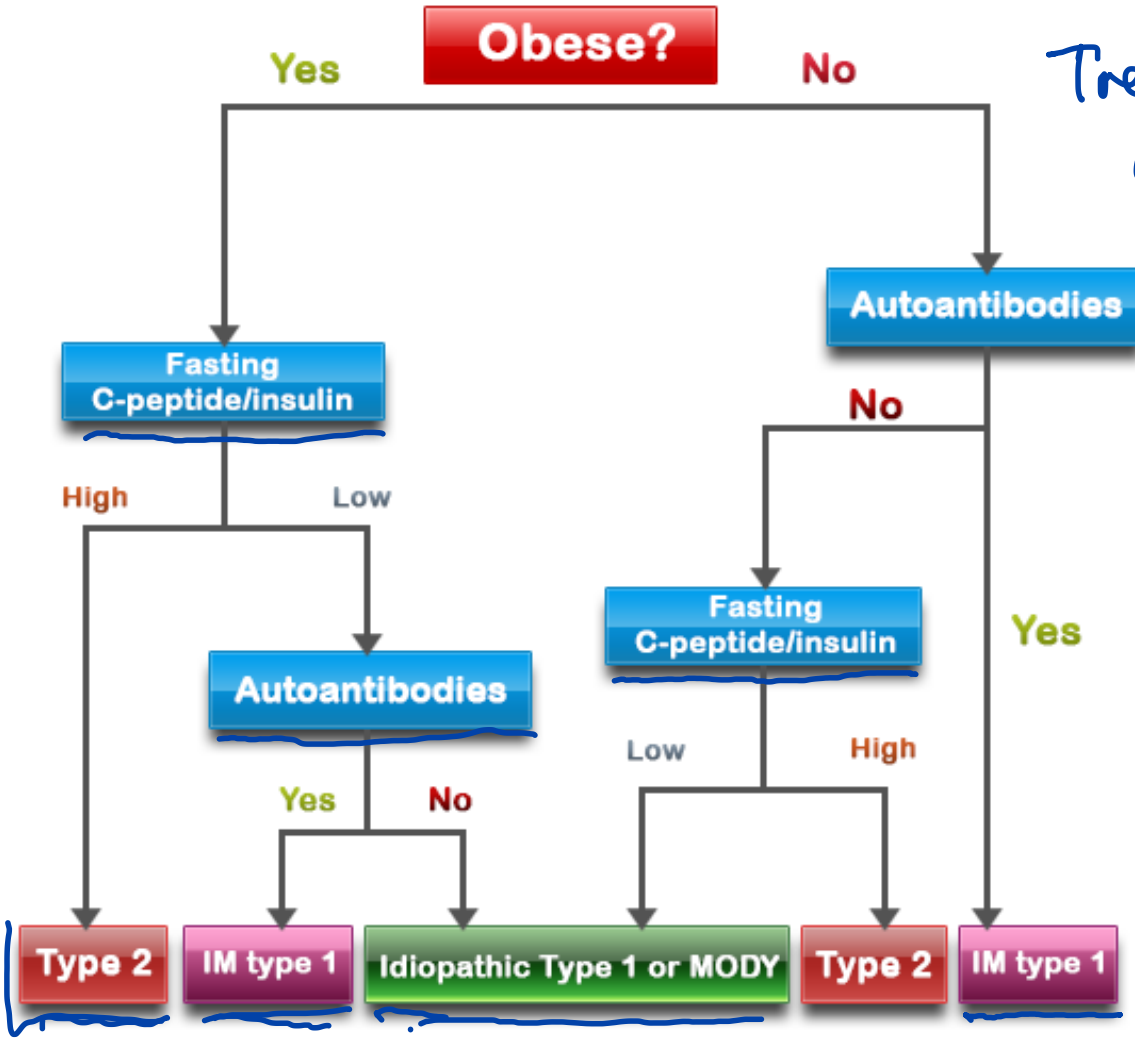
- Decision tree structure
- Learning with decision trees
- Random forests

Motivation: trees and decisions



outcomes
do nothing
WD40
duct tape.

Motivation: trees and decisions



Treatment for diabetes?

Motivation: object recognition

?

?

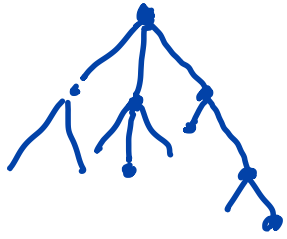
?

Motivation: object recognition

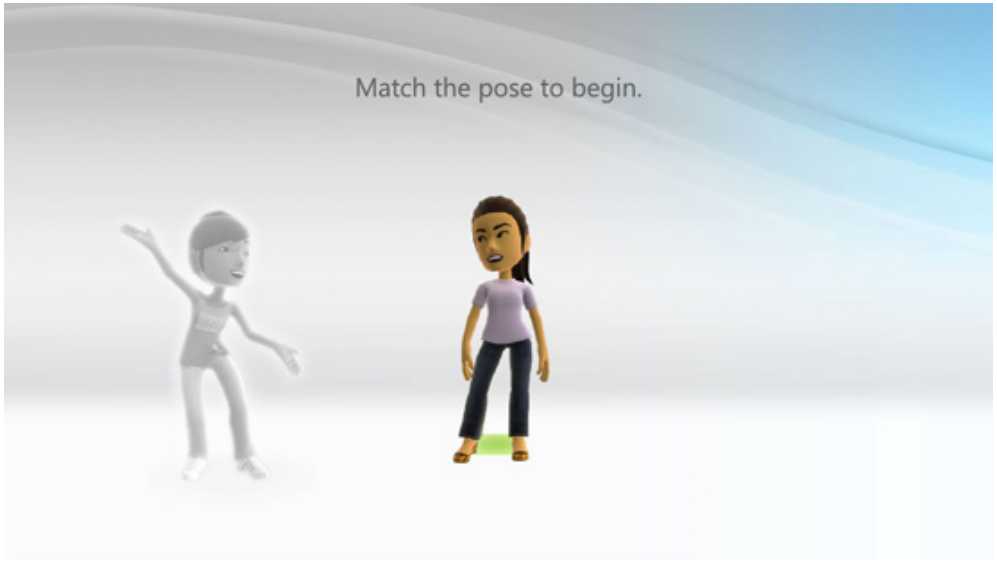
CAR

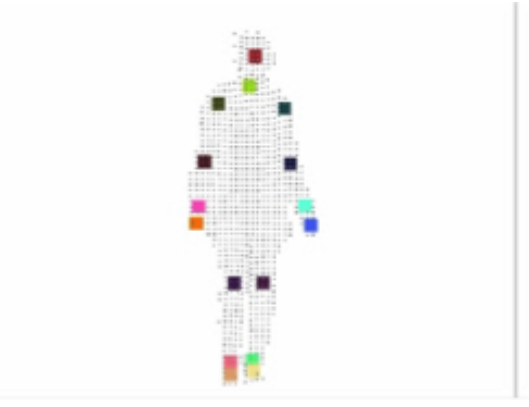
SCOOTER

PEDESTRIAN

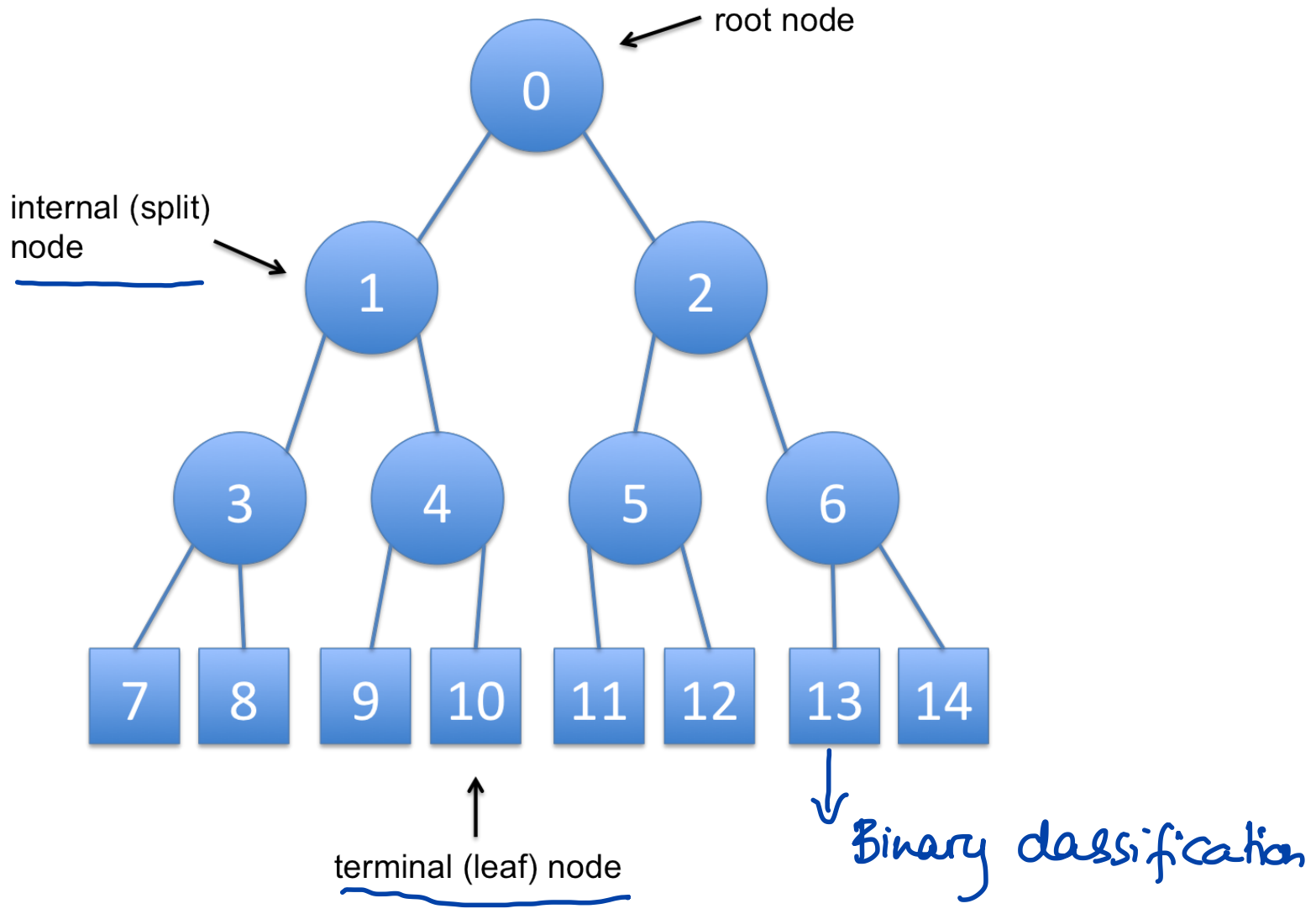


Motivation: pose recognition

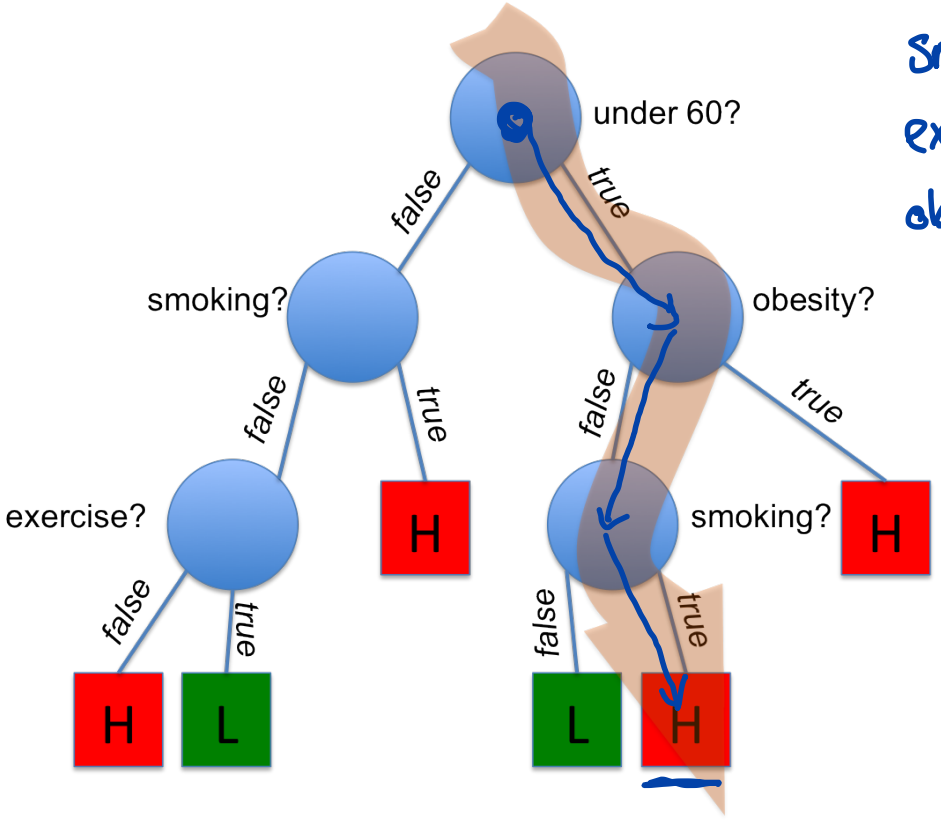




What is a decision tree?



Insurance risk assessment example



under 60 T F
Smoking
exercise
obesity

From data to trees

Examples described by attribute values (Boolean, categorical, continuous, etc.)
E.g., situations where I will assess the insurance risk as high/low:

Example	Attributes					Target
	Age < 60	*Smoking*	*Exercise*	*Obesity*	*City*	*Risk*
X_1	*F*	*F*	*F*	*T*	*Manchester*	*H*
X_2	*F*	*F*	*F*	*T*	*Liverpool*	*H*
X_3	*F*	*T*	*F*	*F*	*Bristol*	*H*
X_4	*F*	*F*	*T*	*F*	*Liverpool*	*L*
X_5	*T*	*F*	*T*	*F*	*Manchester*	*L*
X_6	*T*	*T*	*F*	*T*	*London*	*H*
X_7	*T*	*F*	*F*	*F*	*London*	*L*
X_8	*F*	*F*	*T*	*T*	*Bristol*	*L*

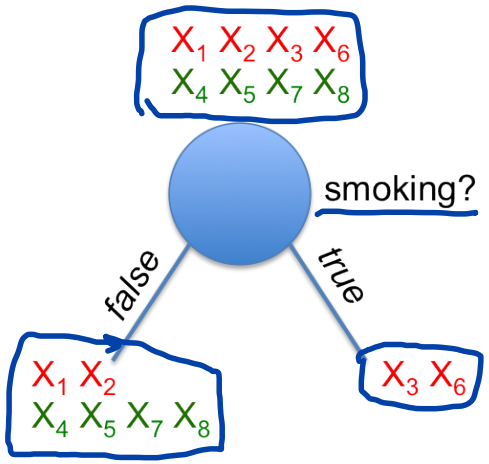
Classification of examples is low (L) or high (H)

From data to trees

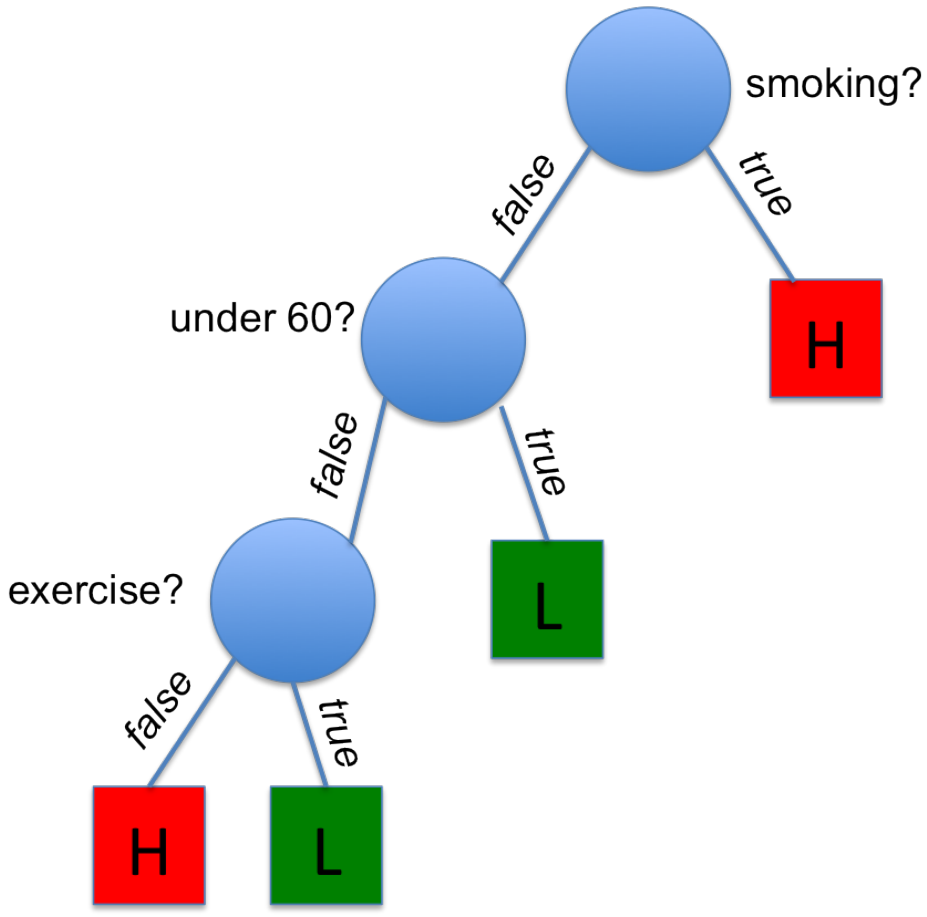
Examples described by attribute values (Boolean, categorical, continuous, etc.)
E.g., situations where I will assess the insurance risk as high/low:

Example	Attributes					Target
	Age < 60	*Smoking*	*Exercise*	*Obesity*	*City*	*Risk*
X_1	*F*	*F*	*F*	*T*	*Manchester*	*H*
X_2	*F*	*F*	*F*	*T*	*Liverpool*	*H*
X_3	*F*	*T*	*F*	*F*	*Bristol*	*H*
X_4	*F*	*F*	*T*	*F*	*Liverpool*	*L*
X_5	*T*	*F*	*T*	*F*	*Manchester*	*L*
X_6	*T*	*T*	*F*	*T*	*London*	*H*
X_7	*T*	*F*	*F*	*F*	*London*	*L*
X_8	*F*	*F*	*T*	*T*	*Bristol*	*L*

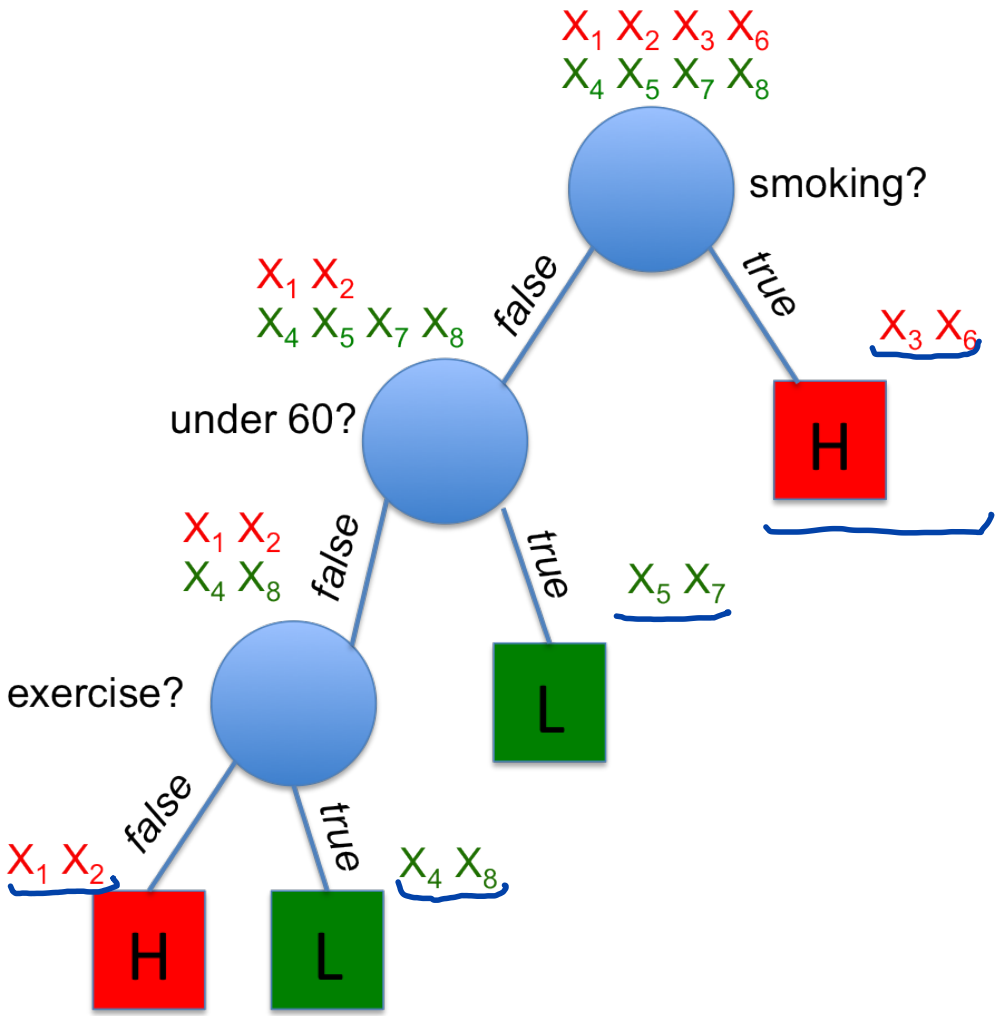
Classification of examples is low (L) or high (H)



A learned decision tree



A learned decision tree



Question

To achieve good generalisation, how to construct the tree?

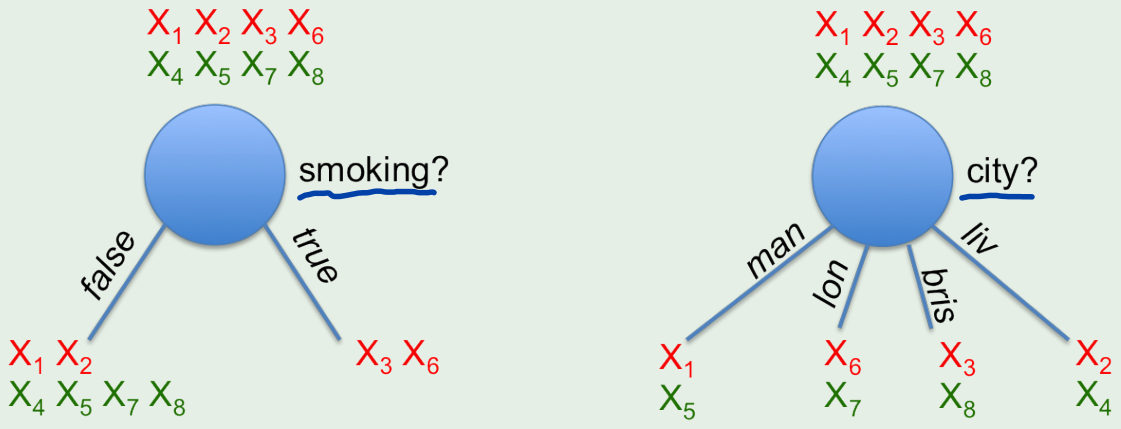
- Few nodes
- As many nodes as possible
- A random number of nodes

How to construct a decision tree?

Aim: find a small tree consistent with the training examples
Idea: (recursively) choose "most significant" attribute as root of (sub)tree

A good attribute splits the examples into subsets that are (ideally) "all red" or "all green"

How to pick a question?

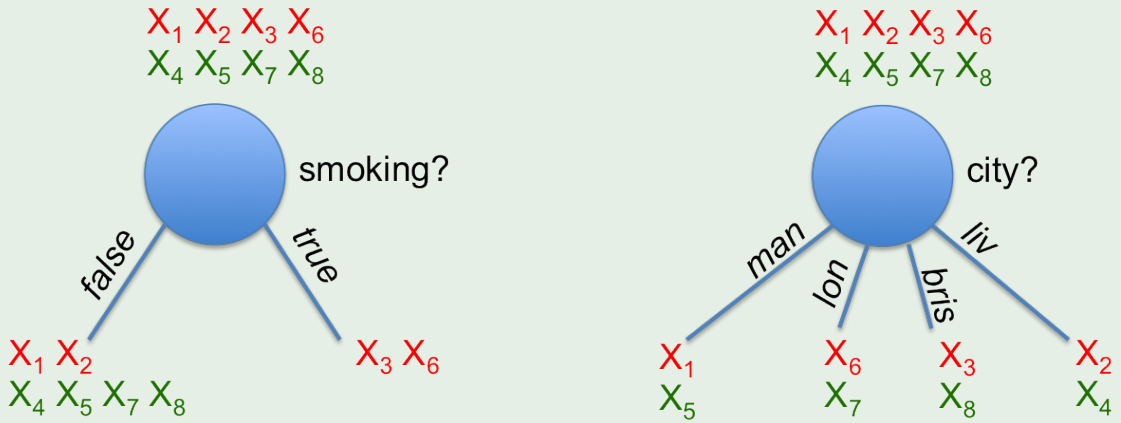


How to construct a decision tree?

Aim: find a small tree consistent with the training examples
Idea: (recursively) choose "most significant" attribute as root of (sub)tree

A good attribute splits the examples into subsets that are (ideally) "all positive" or "all negative"

How to pick a question?



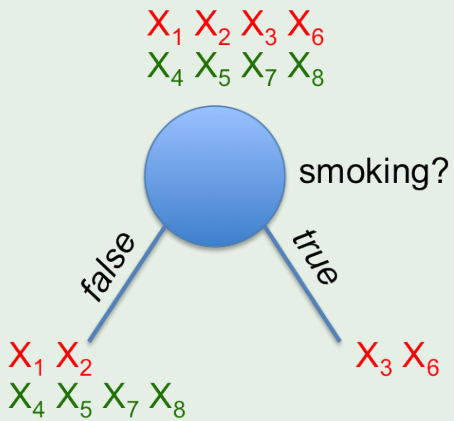
smoking? is a better choice $\rightarrow$ gives *information* about the classification

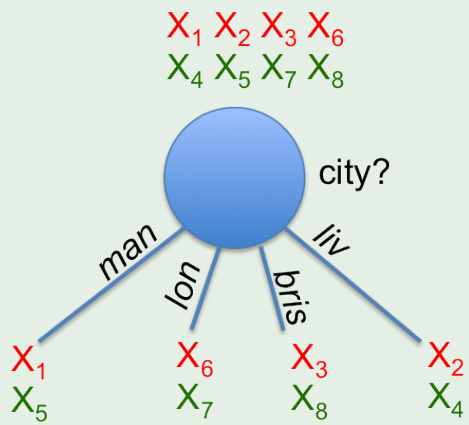
How to pick a question: entropy

Aim: find a small tree consistent with the training examples
Idea: (recursively) choose "most significant" attribute as root of (sub)tree

A good attribute splits the examples into subsets that are (ideally) "all positive" or "all negative"

How to pick a question?





Entropy

Shannon

$$H(X) = -\sum_{x} p(x) \log(p(x))$$

How to pick a question: example

Define $B(q)$ as the entropy of a Boolean random variable that is true with probability q:

$$B(q) = -(q \log_2 q + (1-q) \log_2 (1-q))$$

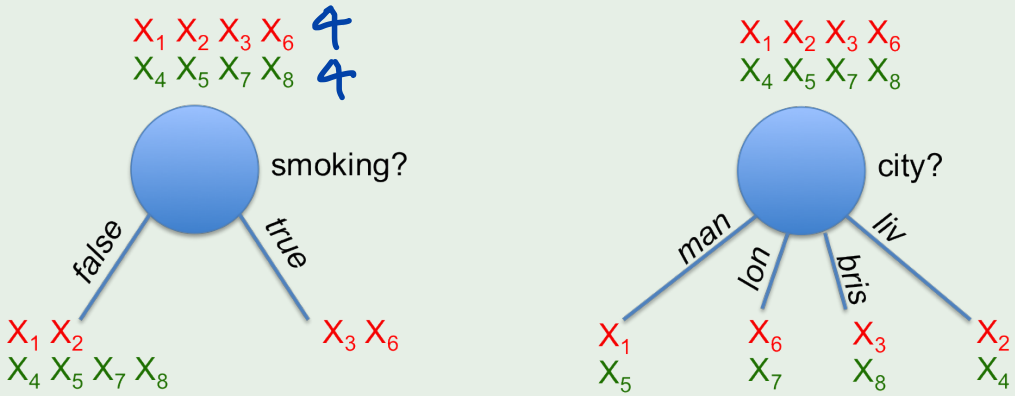
If a training set contains p positive examples and n negative examples, then the entropy of the goal attribute on the whole set is

$$H(Goal) = B\left(\frac{p}{p+n}\right)$$

$B(\frac{1}{2}) = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right)$
$= -\log_2\frac{1}{2} = 1$

In our example $p = n = 4 \rightarrow B(0.5) = 1$ bit.

How to pick a question?



A test on a single attribute might give us only part of the 1 bit: entropy remaining after the attribute test.

How to pick a question: information gain

An attribute A with d values splits the training set E into subsets $E_1, \ldots, E_d$. Each subset E_k has p_k positive examples and n_k negative examples.

Remainder

The expected entropy remaining after testing attribute A is

$$Remainder(A) = \sum_{k=1}^{d} \frac{p_k + n_k}{p + n} B\left(\frac{p_k}{p_k + n_k}\right)$$

weighted sum of binary entropy of each subset E_k

Information gain

The information gain from the attribute test on A is the expected reduction in entropy:

$$Gain(A) = B\left(\frac{p}{p + n}\right) - Remainder(A)$$

Choose the attribute with the largest Gain!

[Russell and Norvig]

How to pick the nodes: example

A test on a single attribute might give us only part of the 1 bit: entropy remaining after the attribute test.

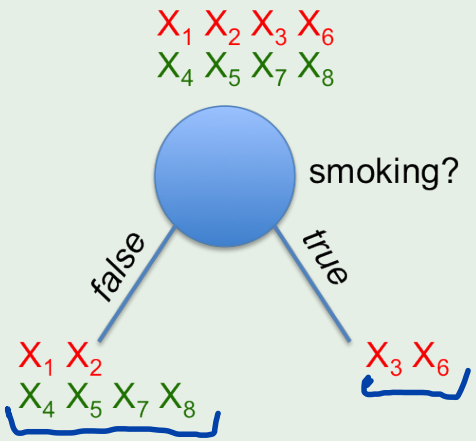
$$Remainder(Smoking) = \frac{2}{8}B(1) + \frac{6}{8}B\left(\frac{2}{6}\right) \approx 0.689 bits$$

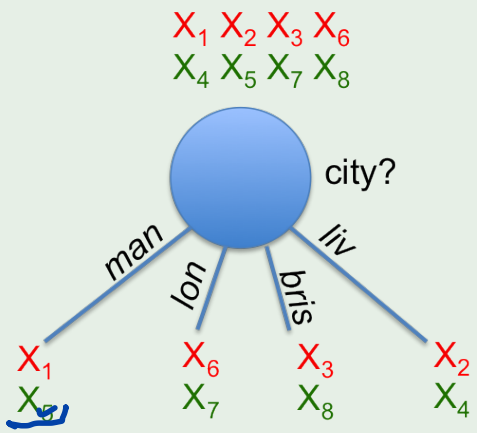
$$Remainder(City) = \frac{2}{8}B(0.5) + \frac{2}{8}B(0.5) + \frac{2}{8}B(0.5) + \frac{2}{8}B(0.5) = 1 bits$$

$$Gain(Smoking) = 1 - 0.689 = 0.311 bits$$

$$Gain(City) = 1 - 1 = 0 bits$$

How to pick a question?





How to pick the nodes: example

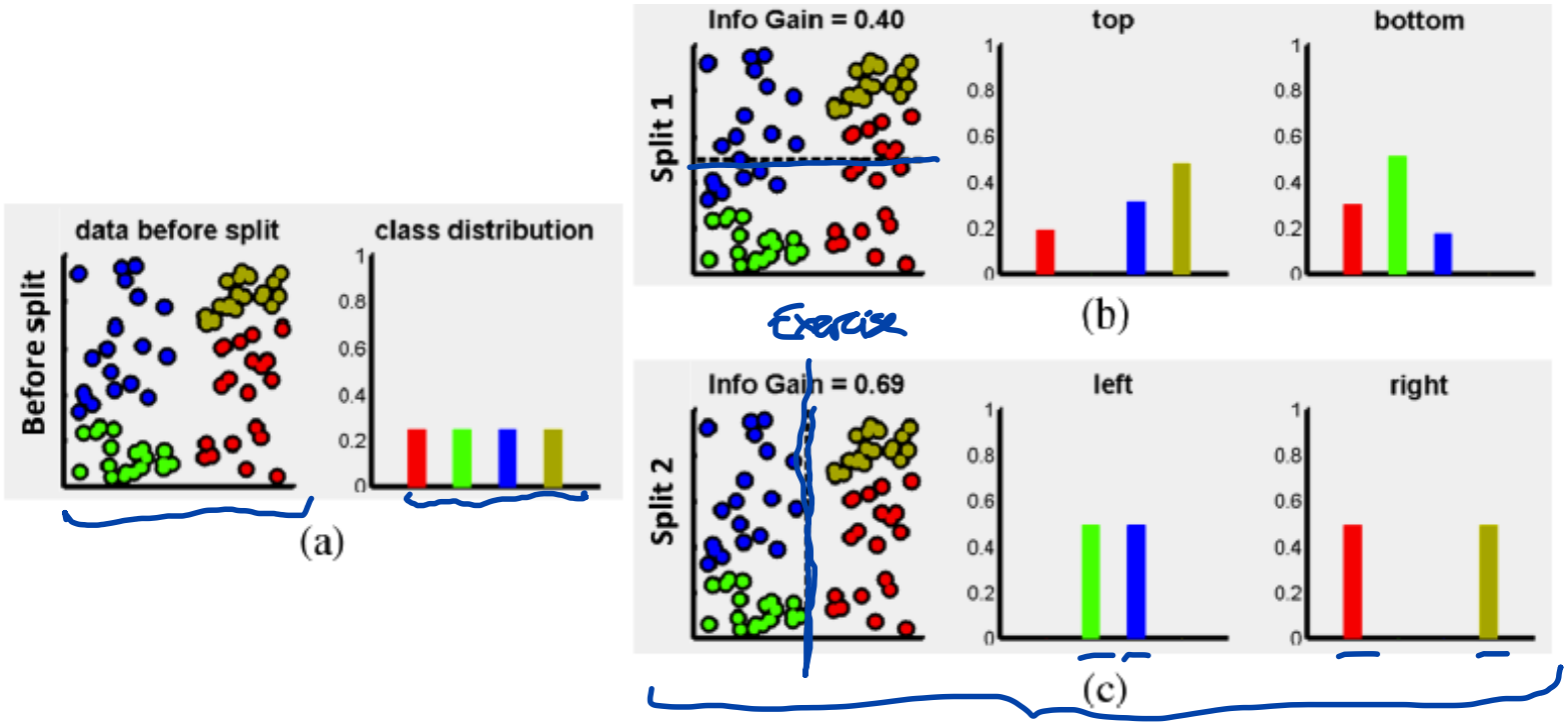


Fig. 2.5 Information gain for discrete, non-parametric distributions. (a) Dataset $\mathcal{S}$ before a split. (b) After a horizontal split. (c) After a vertical split. In this example the vertical split produces purer class distributions in the child nodes. Classes are colour coded.

Decision tree algorithm

Algorithm

function DECISION-TREE-LEARNING(*examples*, *attributes*, *parent examples*) **returns** *a tree*

if *examples* is empty **then return** PLURALITY-VALUE(*parent examples*)
else if all *examples* have the same classification **then return** the classification
else if *attributes* is empty **then return** PLURALITY-VALUE(examples)
else
  $A \leftarrow \arg\max_{a \in attributes}$ IMPORTANCE(a, *examples*)
  tree $\leftarrow$ a new decision tree with root test A
  for each value v_k of A do
    $exs \leftarrow \{e : e \in examples \quad \textbf{and} \quad e.A = v_k\}$
    subtree $\leftarrow$DECISION-TREE-LEARNING(exs, $attributes - A$, *examples*)
    add a branch to *tree* with label $(A = v_k)$ and subtree *subtree*
  return *tree*

recursive

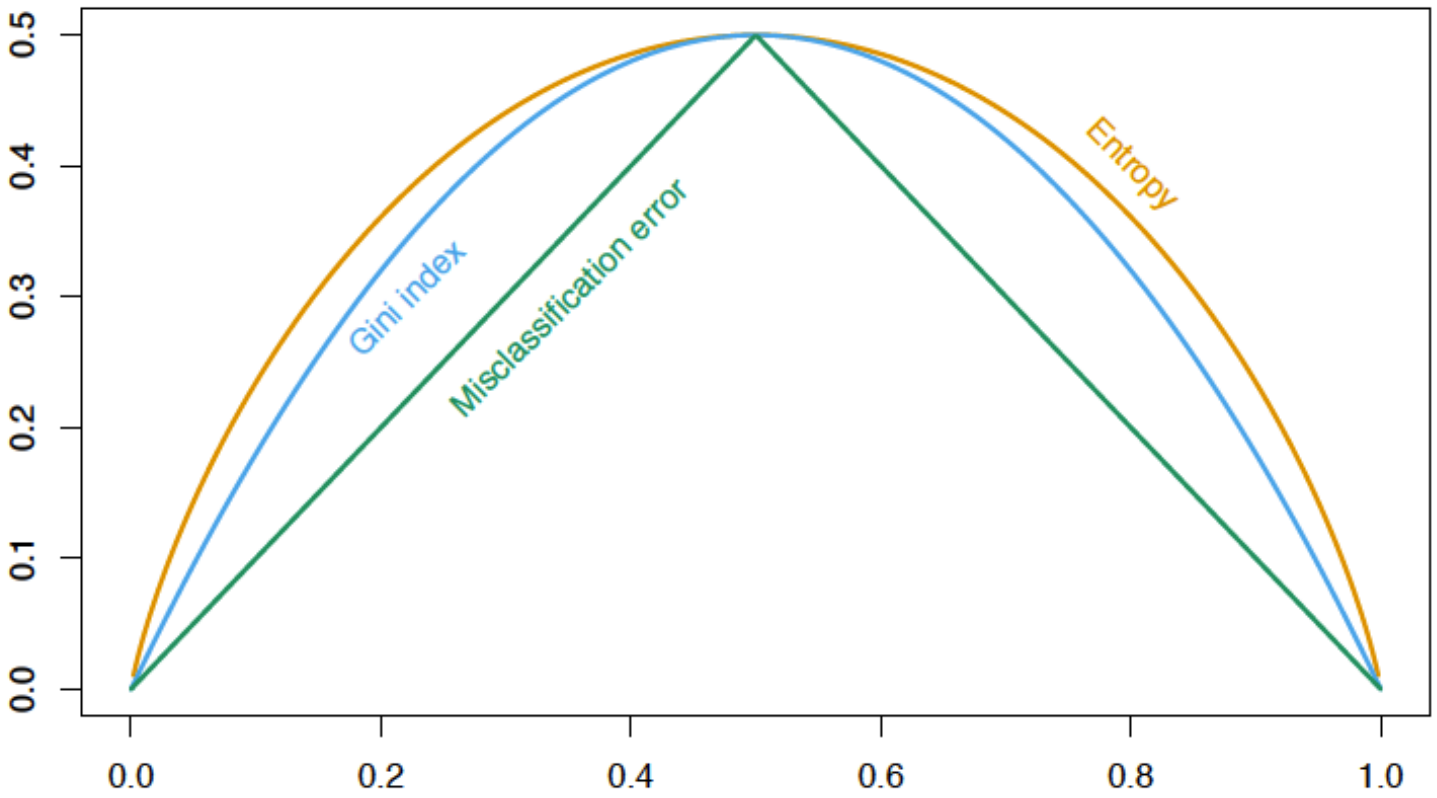
IMPORTANCE

The function IMPORTANCE(*a*, *examples*) returns the information gain of *atrribute a* for the examples in *examples*.

PLURALITY-VALUE

The function PLURALITY-VALUE(*examples*) selects the most common output value among a set of *examples*, breaking ties randomly.

Decision trees in practice: alternative quality measures



Entropy	$-q \log q - (1-q) \log(1-q)$
Gini index	$2q(1-q)$
Misclassification error	$1 - \max(q, 1-q)$

Decision trees in practice: properties

Properties

- Interpretability
- Missing data
- Binary partitions
- Tree size
- Instability of trees
- Continuous and integer-valued input attributes
- Continuous-valued output attributes

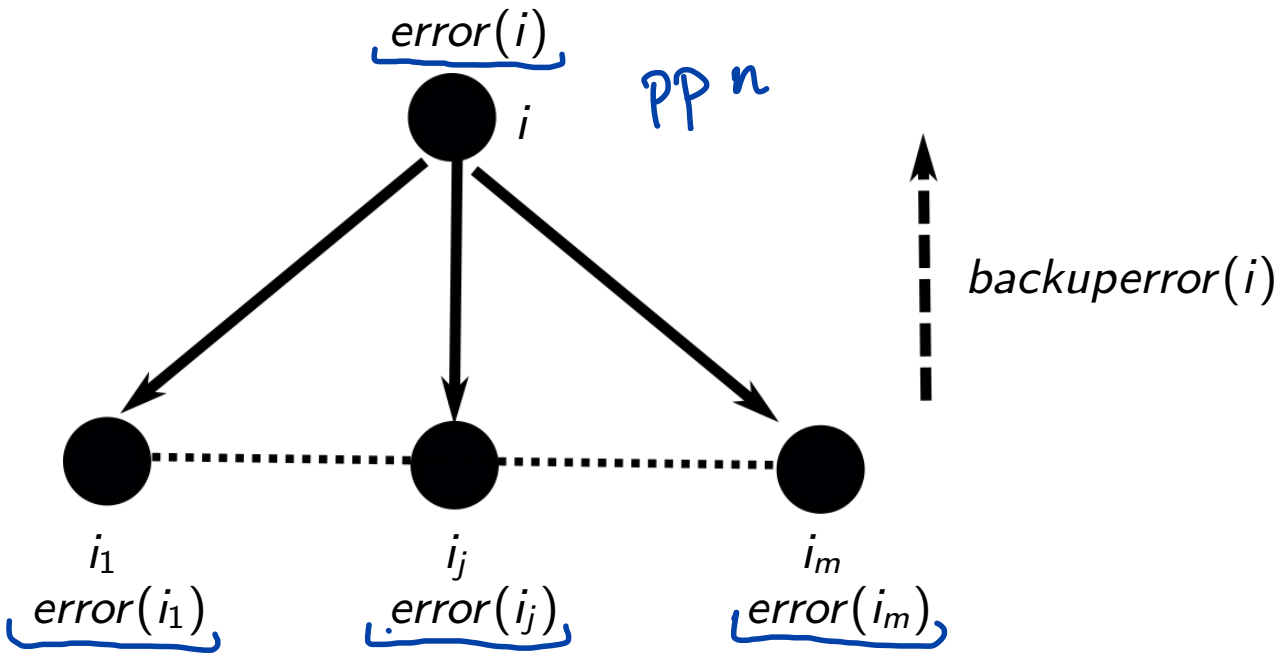
[Hastie, Tibshirani, Friedman and Russell and Norvig]

Pruning Decision Trees

- Pruning decision trees can help to improve generalisation.
- Suppose for a node i there are k patterns and $c \geqslant \frac{k}{2}$ belong to the majority class.
- Suppose that we truncate the tree a node i and label that node with the majority class.
- Then there is an associated error (denoted $error(i)$) corresponding to the probability that a pattern belonging to i will be wrongly classified i.e. that it will belong to a class different from the majority class.
- If we have two classes then we can work out this misclassification probability by using Laplace's Law.

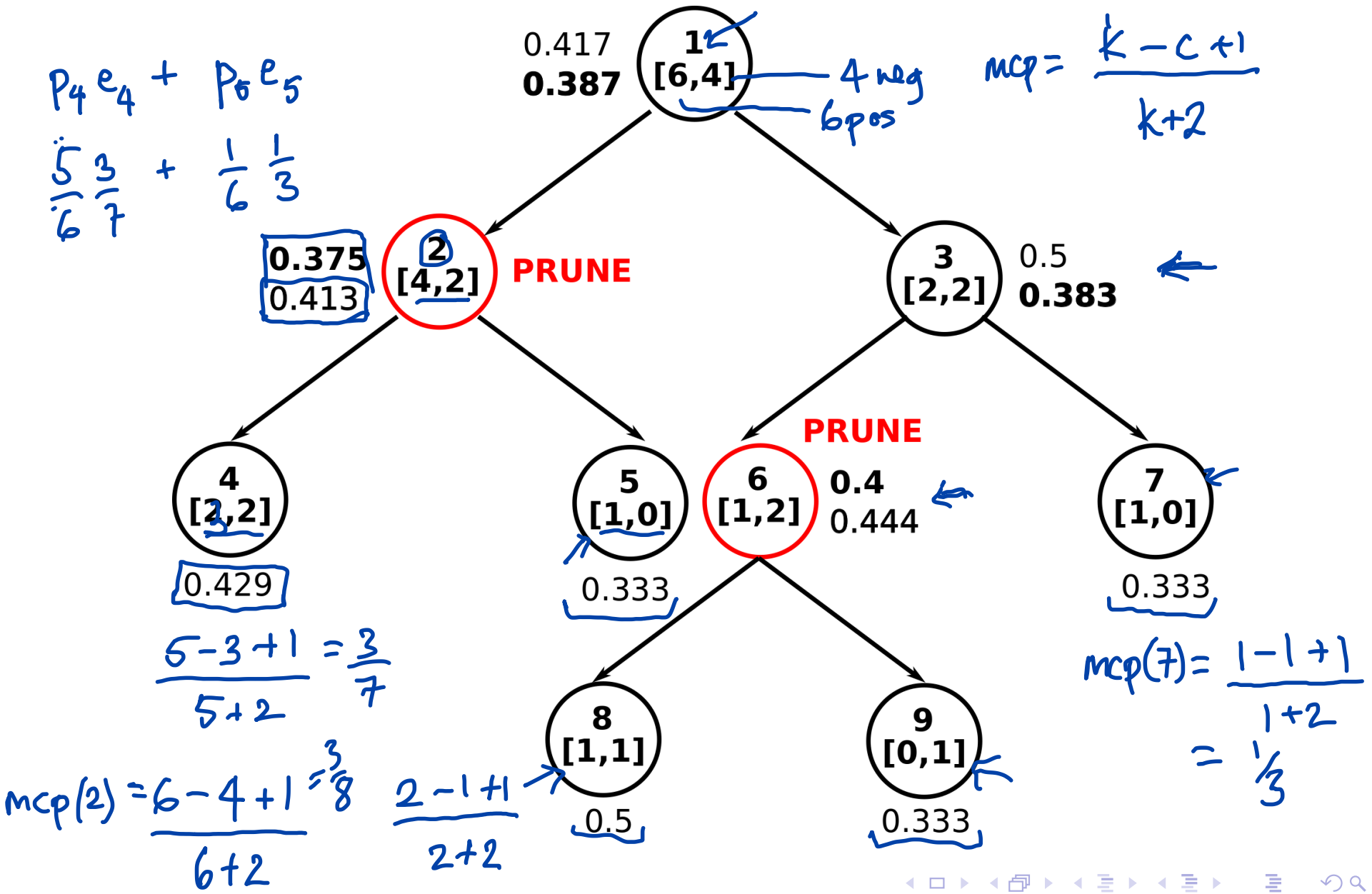
$$mcp(i) = \frac{k - c + 1}{k + 2}$$

Pruning: The Binary Case

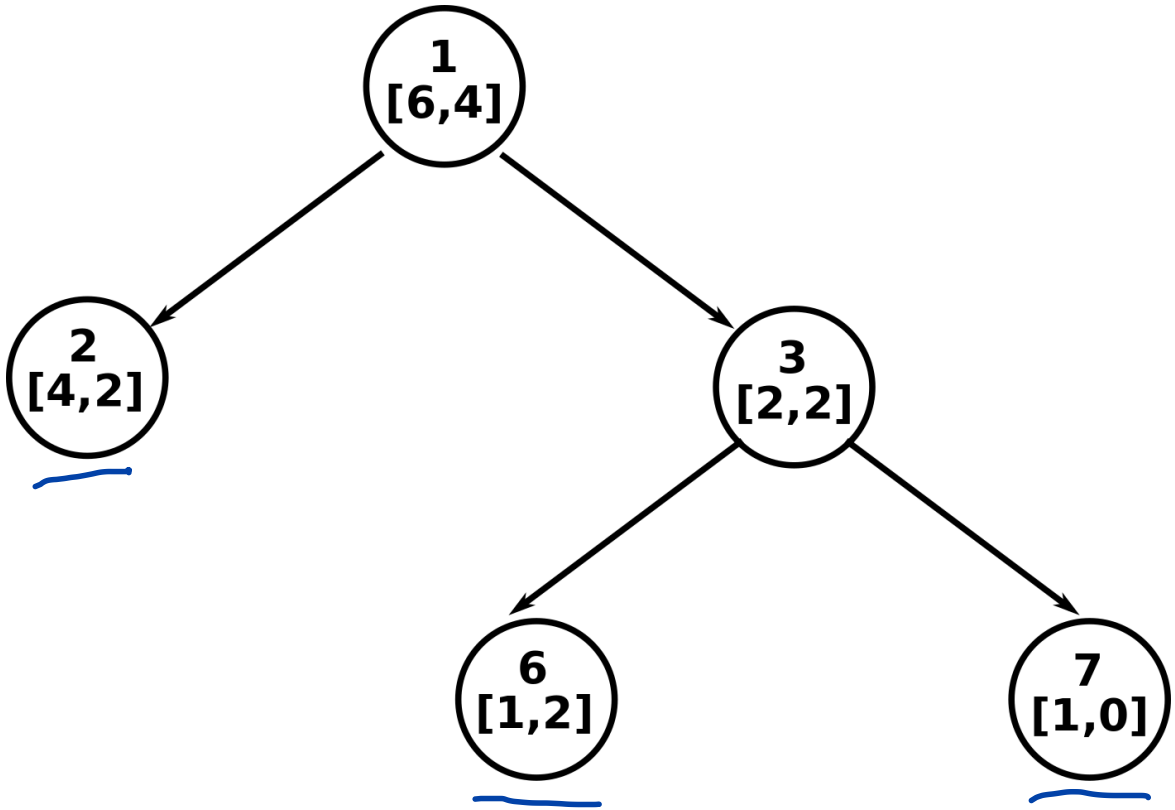


- Where $backuperror(i) = \sum_{j=1}^{m} error(i_j)$
- And where $error(i) = \min(mcp(i), backuperror(i))$
- The node i is pruned if $backuperror(i) \geq error(i)$

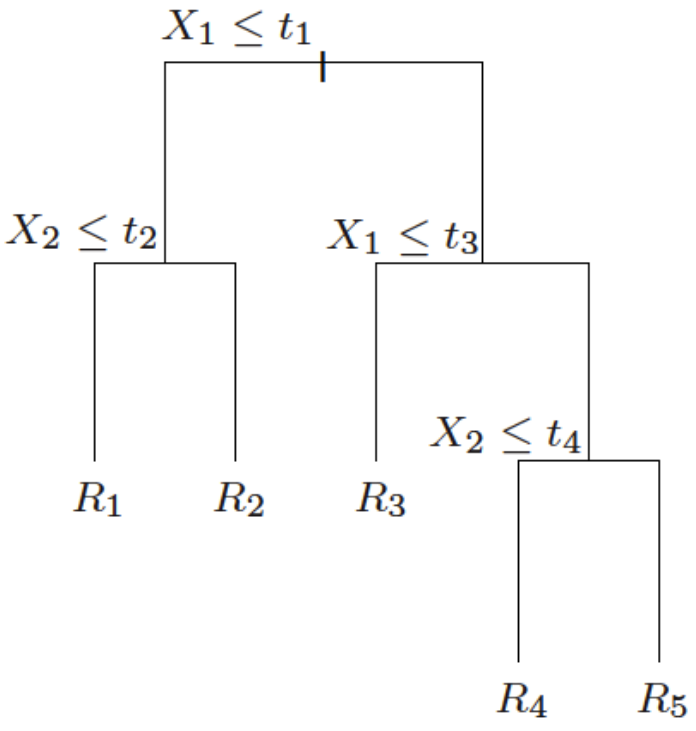
Example of Pruning

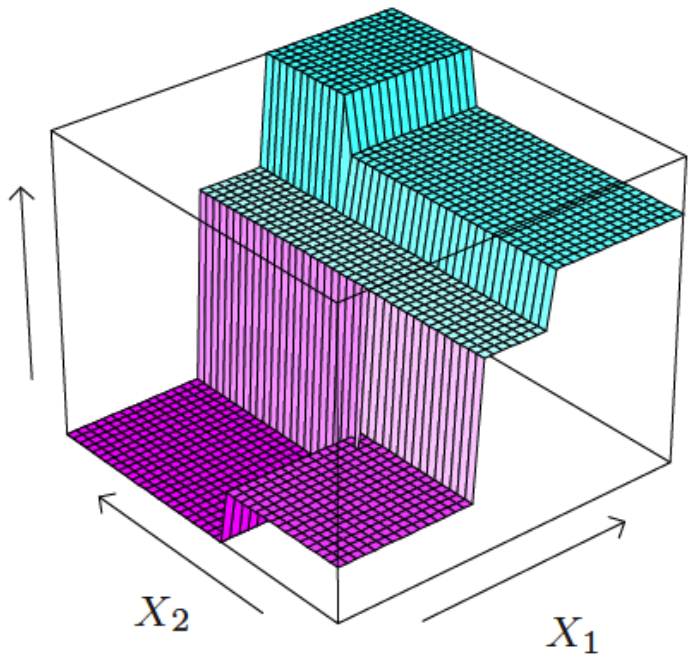


After Pruning



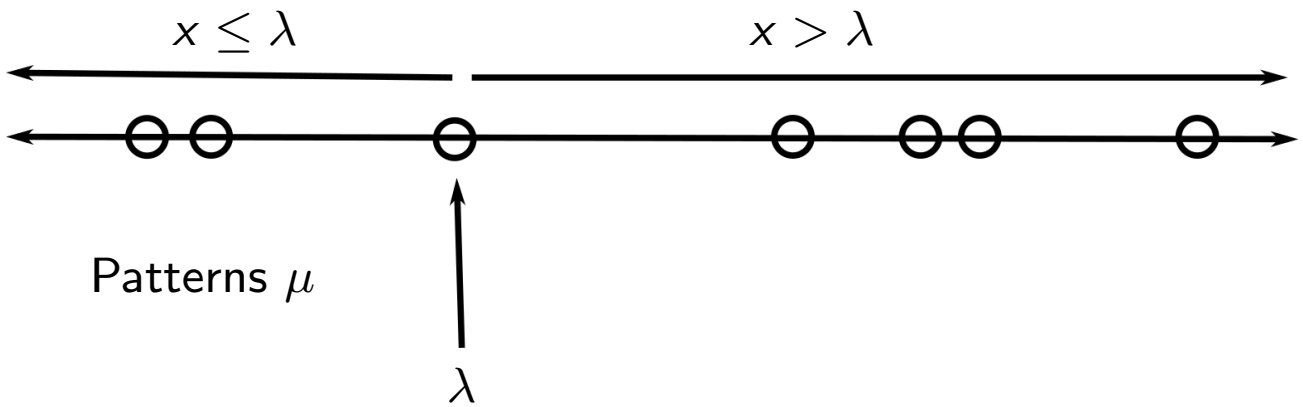
Decision trees for regression





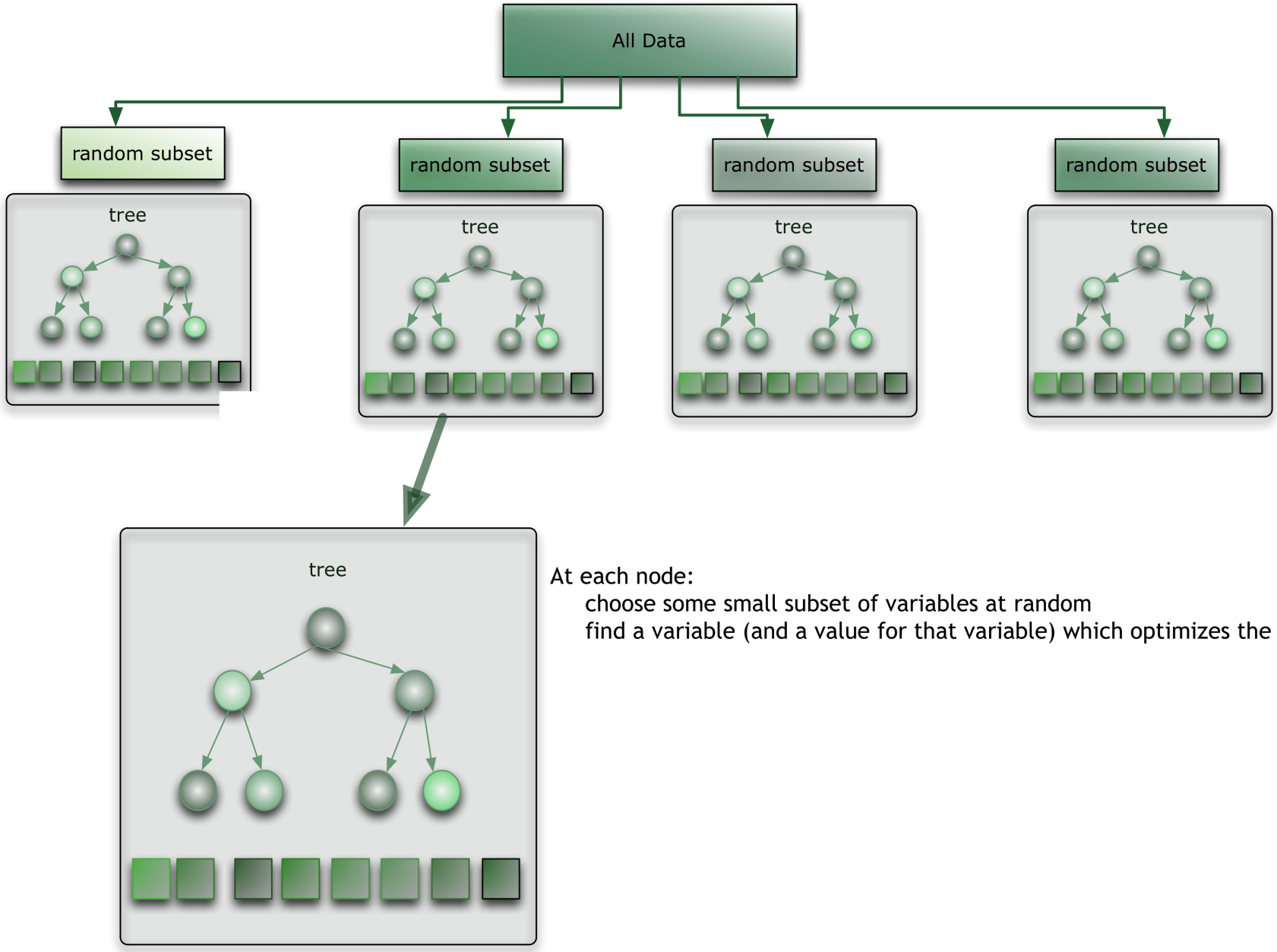
Decision Trees with Continuous Variables

- So far we have only considered discrete variables so that the number of leaves at a node is always finite.
- If a variable is continuous then a threshold λ is determined at a given node, so that evaluation the variable forms two leaves; $x \leq \lambda$ and $x > \lambda$.



- $H_{x \leq \lambda} = -P(yes|x \leq \lambda) \log_2(P(yes|x \leq \lambda)) - P(no|x \leq \lambda) \log_2(P(no|x \leq \lambda))$
- $H_{x > \lambda} = -P(yes|x > \lambda) \log_2(P(yes|x > \lambda)) - P(no|x > \lambda) \log_2(1 - P(no|x > \lambda))$
- Pick λ to minimize $P(x \leq \lambda)H_{x \leq \lambda} + P(x > \lambda)H_{x > \lambda}$

Random forests



At each node:

- choose some small subset of variables at random
- find a variable (and a value for that variable) which optimizes the split

Random forests algorithm

Algorithm

For some number of trees T:

- Sample n examples at random with replacement to create a subset of the data (e.g. about 66% of the total set).
- **At each** tree:

  At each node:

  For some number m (see below), m predictor variables are selected at random from all the predictor variables.
  The predictor variable that provides the best split, according to some objective function, is used to do a binary split on that node.

Depending upon the value of m, there are three slightly different systems:

- Random splitter selection: $m = 1$
- Breiman's bagger: $m =$ total number of predictor variables.
- Random forest: $m <<$total number of predictor variables.

Prediction: when a new input is entered into the system, it is run down all of the trees. The result may either be an average or weighted average of all of the terminal nodes that are reached, or, in the case of categorical variables, a voting majority.

Random forests in practice

Strengths: Random forest runtimes are quite fast, and they are able to deal with unbalanced and missing data.

Weaknesses: When used for regression they cannot predict beyond the range in the training data, and that they may over-fit datasets that are particularly noisy.

Parameters

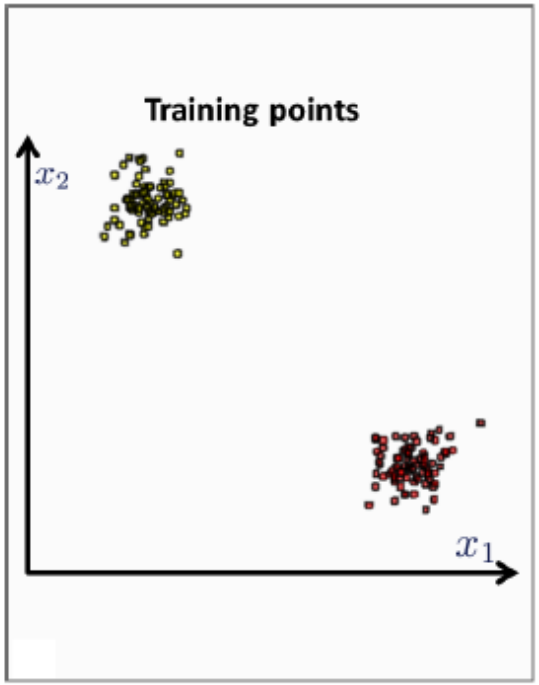
The parameters that most influence the behavior of a decision forest are:

- The maximum allowed tree depth D;
- The amount of randomness and its type;
- The forest size (number of trees) T;
- The choice of weak learner model;
- The training objective function;
- The choice of features in practical applications.

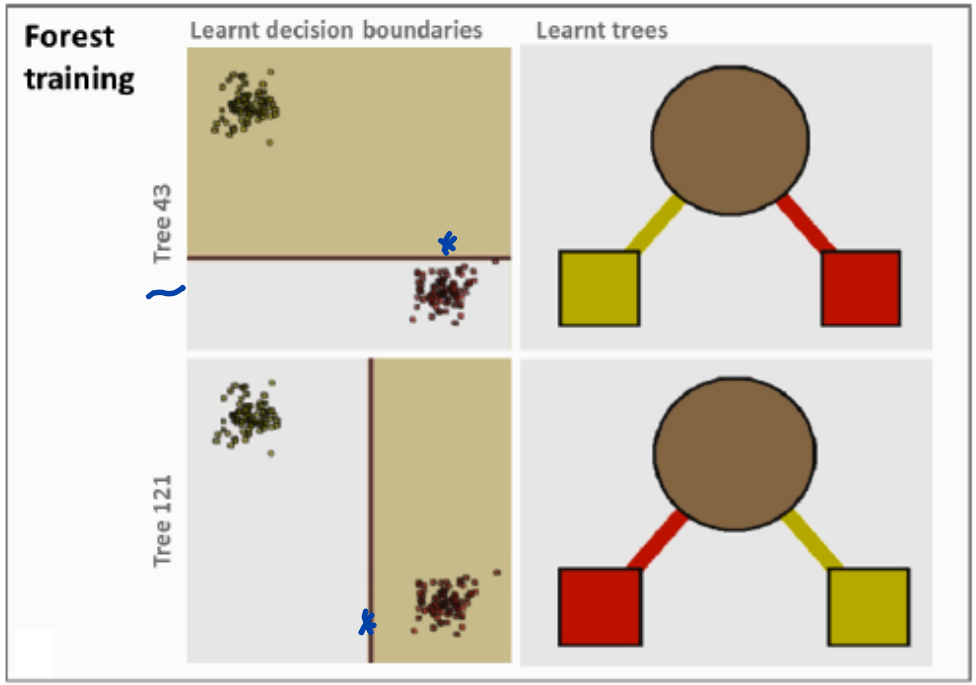
[Criminisi, Shotton, Konukoglu]

https://www.theguardian.com/science/the-lay-scientist/2016/feb/18/has-a-rampaging-ai-algorithm-really-killed-thousands-in-pakistan

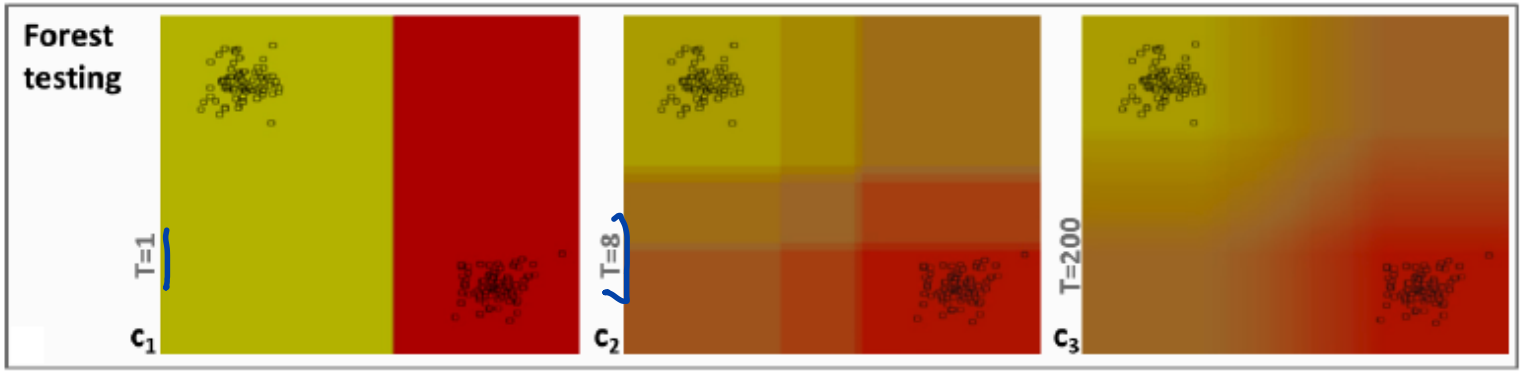
Random forests in practice: effect of T



(a)



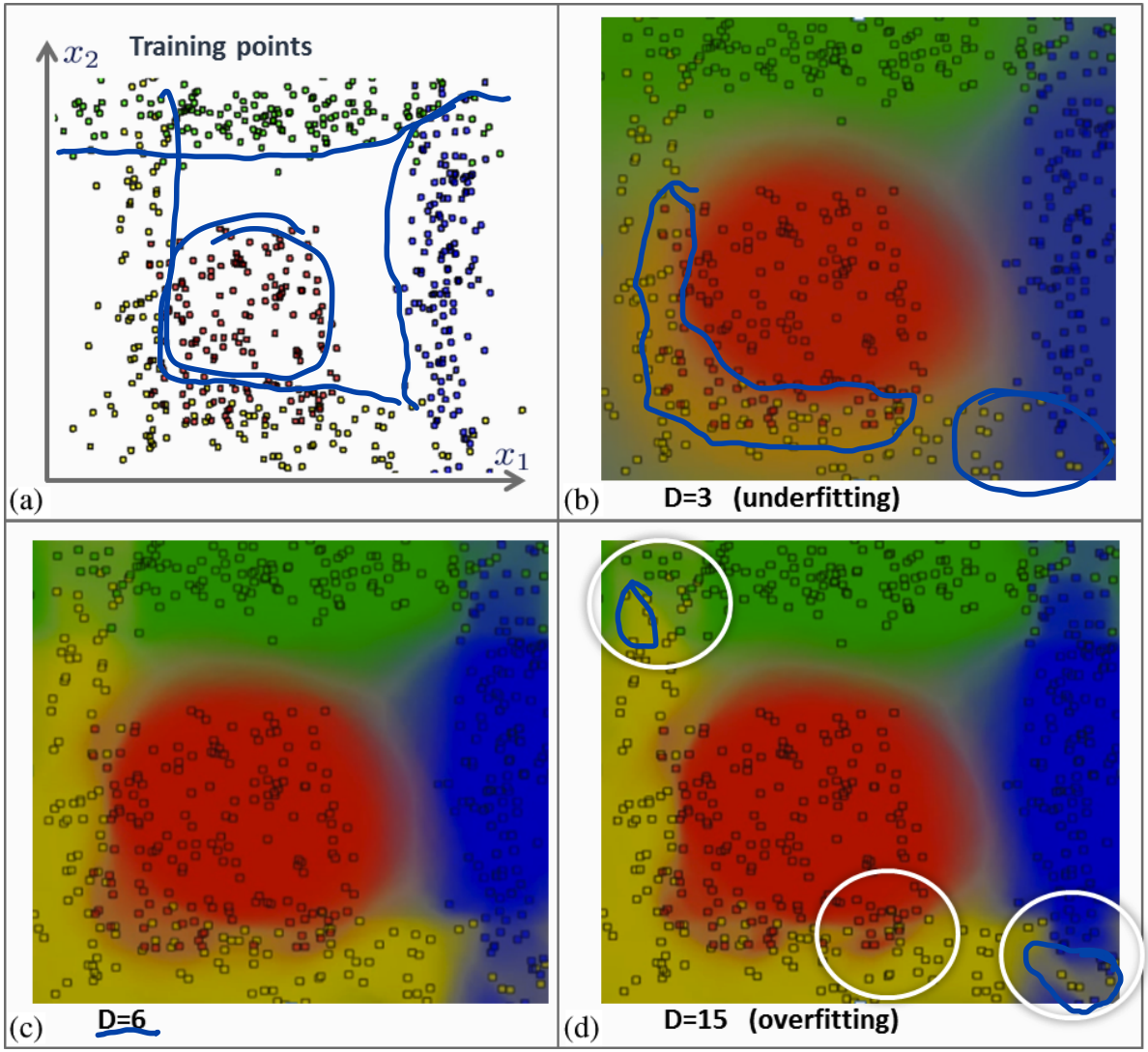
(b)



(c)

[Criminisi, Shotton, Konukoglu]

Random forests in practice: effect of D



[Criminisi, Shotton, Konukoglu]