DECISION TREES

Consider a classification problem with feature vector

$$x = (x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)})$$

where

$x^{(1)}$ = age (years)

$x^{(2)}$ = weight (pounds)

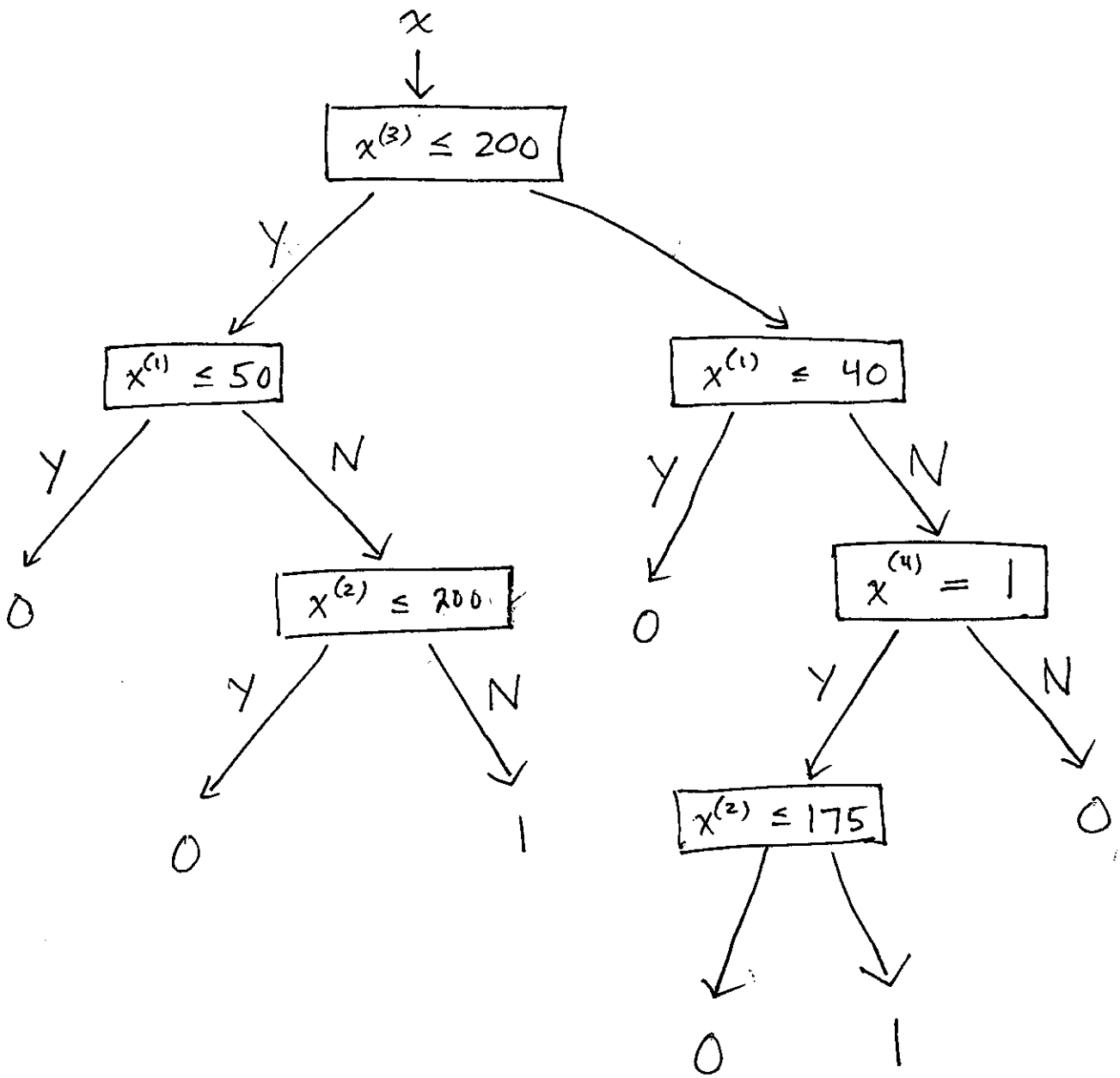
$x^{(3)}$ = cholesterol level

$x^{(4)}$ = $1_{\{\text{father had heart disease}\}}$

and label

$$y = \begin{cases} 1 \rightarrow \text{at risk for heart disease} \\ 0 \rightarrow \text{not at risk.} \end{cases}$$

Further suppose the data in x is collected from a male patient in a certain population



A <u>decision tree</u> is a tree-structured sequence of questions about x, used to make predictions about x.

Decision trees are

- interpretable
- conceptually simple

For this reason, decision trees are favored by many practitioners, such as medical professionals, who want more than a

(A) _____ _____ predictor.

Classification vs. Regression Trees

Decision trees can also be applied to regression problems.

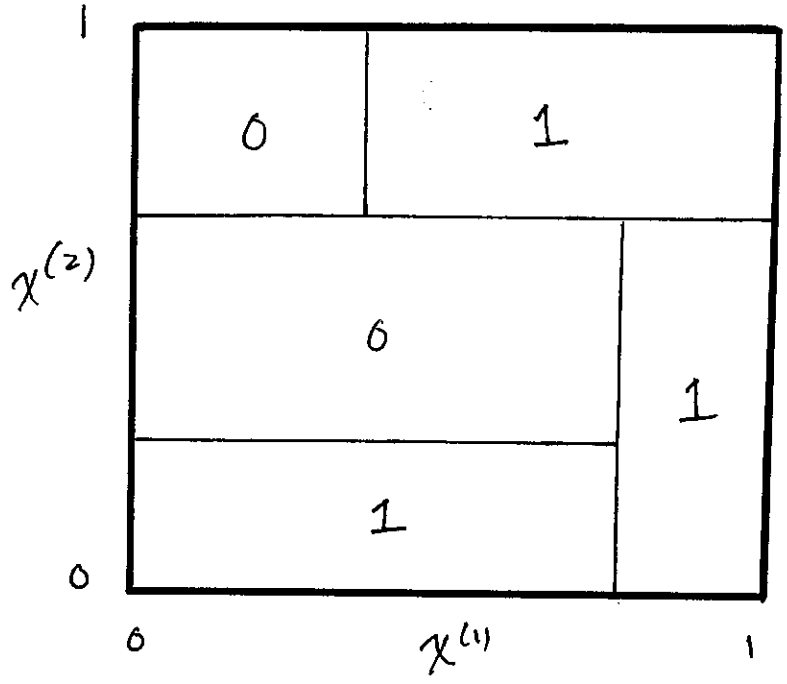
In the previous example, we could try to predict

(B) $y =$

In these notes we'll focus on classification.

Partitions

Every decision tree is associated with a partition of feature space



Generalizations

- splits involving more than one feature

(P) Ex

- splits with more than two outcomes

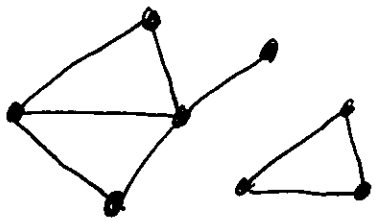
Ex

In practice, the generalizations are typically not adopted for the following reasons:

(E)

Formal Definition

A graph is a collection of vertices certains pairs of which are joined by edges.

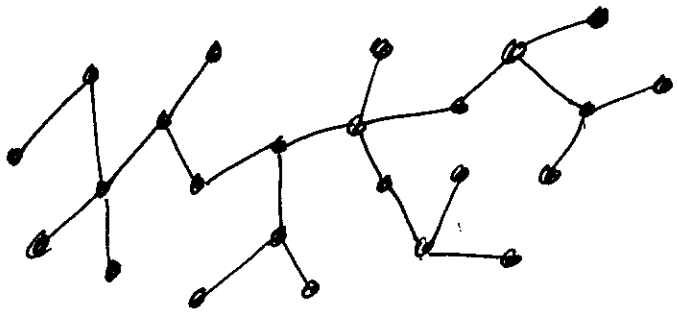


Two vertices joined by an edge are neighbors

The degree of a vertex is the number of neighbors it has.

A tree is a graph that is

- acyclic
- connected



abstract tree

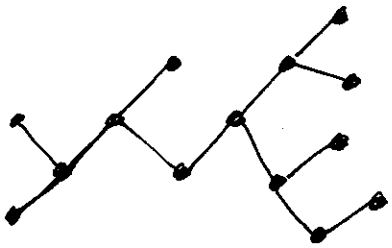
In a tree, we use the following terms:

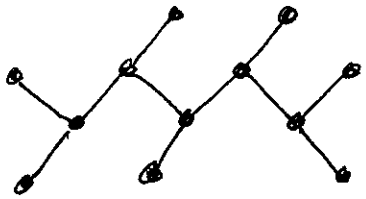
- node = vertex
- internal node: degree ≥ 2
- terminal/leaf node: degree $= 1$
- branch = edge

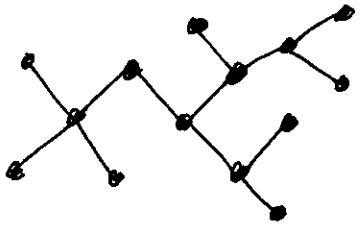
A <u>rooted binary tree</u> is a tree such that:

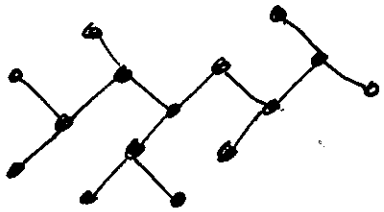
- there is a unique internal node of degree 2 called the <u>root</u>.
- every other internal node has degree 3.

Which of the following is a rooted binary tree:









It is customary to draw a rooted binary tree with the root at the left or top.

(F) The tree above:

For a node in a rooted binary tree, how would you define its

(G)

- depth
- parent
- children
- sibling

Definition | A binary decision tree is a rooted binary tree such that

1) every internal node asks a question of the feature vector with a binary answer
2) every leaf node predicts a label (or a response variable, in the case of regression trees)

Constructing Decision Trees

Suppose we have a labelled training data set

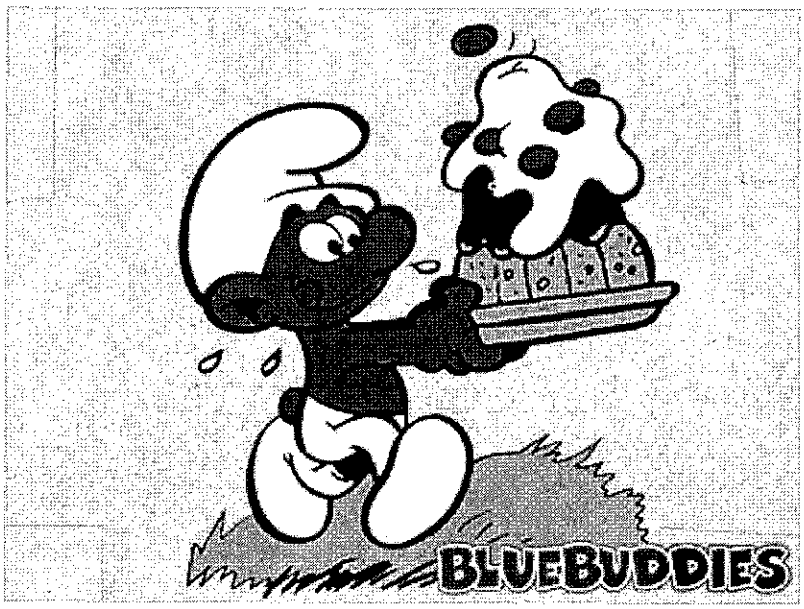
$$(x_1, y_1), \ldots, (x_n, y_n), \quad y_i = 0, 1$$

How should we construct a decision tree to correctly classify future patterns?

Unfortunately, the space of all possible decision trees is huge, and efficient global search strategies do not exist.

Therefore, we will resort to a

(H) _____ algorithm, which chooses the next best split in a recursive fashion.



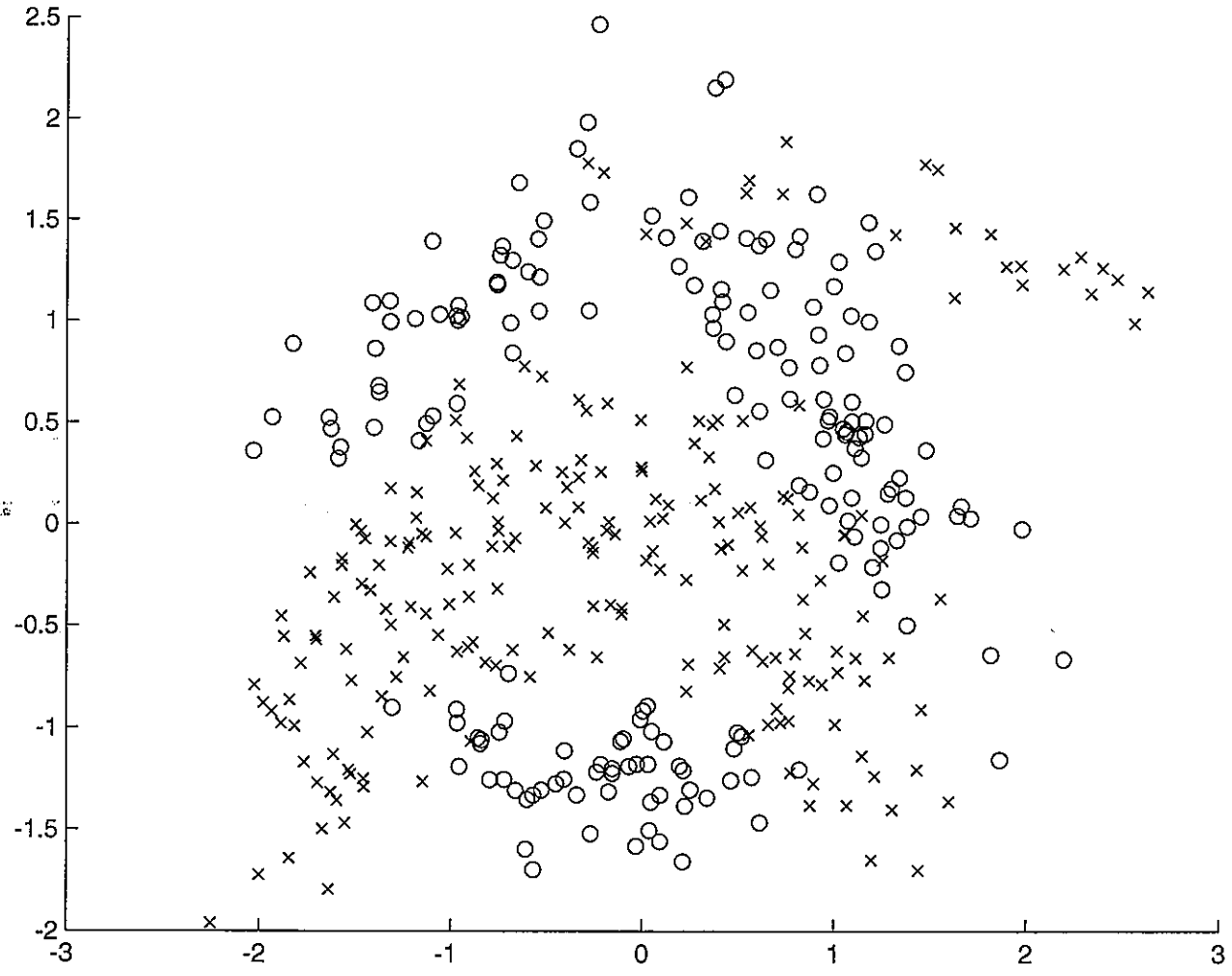
Greedy Growing

1. Start at root node (= entire feature space)
2. Decide whether to stop growing tree.
   If yes, assign label to node. Stop.
3. If no, consider all possible questions that might be asked of data reaching that node. Determine best split.
4. For each outcome of the split, create a new node and go to 2.

Thus, tree growing requires four ingredients:

1.
2.
3.
4.

Example 1



Possible splits

For unordered / categorical / discrete features:

"What is $x^{(j)}$?"

For ordered / numerical features:

"Is $x^{(j)} \leq t$?"

These are univariate splits. Multivariate splits are also possible.

In practice a finite list of possible splits is considered.

Splitting Rules

Suppose N is a node. We want to find the best split of $\{x_i : x_i \in N\}$.

Intuitively, a good split leads to children

(J) that are _____.

We say a node has higher _____ if the distribution of classes is closer to uniform. We want children to be _____ than their parents.

Suppose $i(N)$ is an impurity function, with $i(N) = 0$ indicating only one class present. Let N_1, N_2 be children of N and define

$$p(N_1) =$$

$$p(N_2) =$$

Then we seek the split into (N_1, N_2) for which

(K)

is maximal.

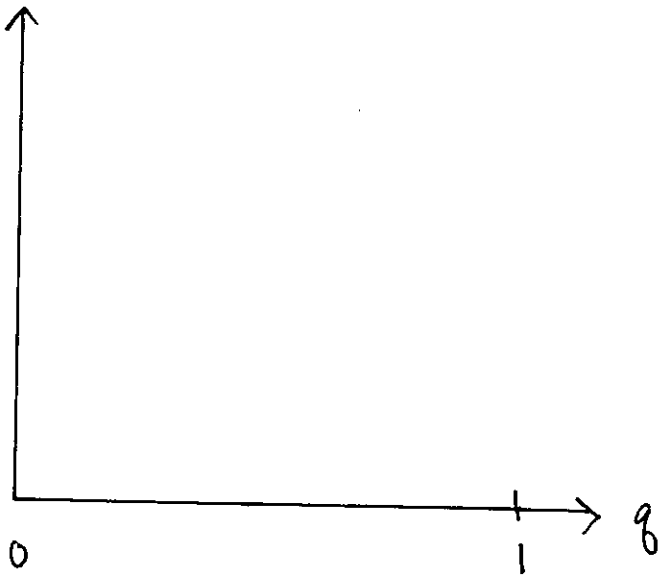
Impurity functions

Consider a fixed node N. Assume $y_i \in \{0,1\}$.

Define

$$q := \frac{|\{i : x_i \in N,\ y_i = 0\}|}{|\{i : x_i \in N\}|}$$

What should an impurity function look like as a function of q?



Entropy

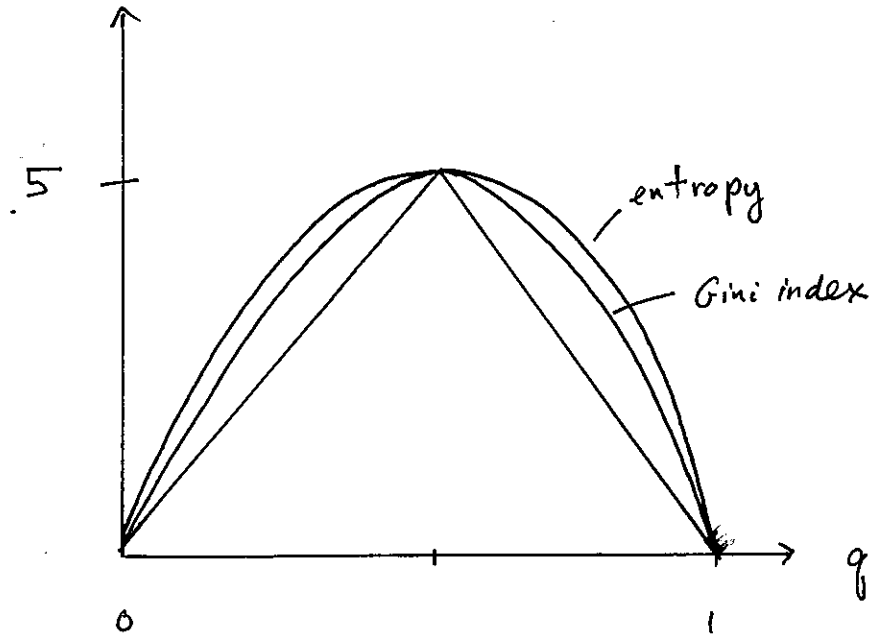
(L) $i(N) =$

Gini index

$i(N) =$

Misclassification rate

$i(N) =$



In practice, the rounded functions are preferred. Why do you think that is? (M)

Stop Splitting Rule

- When decrease in impurity falls below some tolerance.
- When 90% of samples belong to same class.
- When tree reaches certain depth.

Classifying Leaf Nodes

- Majority vote (most common)
- Any other classification rule

Pruning

- Growing right-sized trees is important
  - ▷ too small ⟹ can't adequately model Bayes classifier
  - ▷ too large ⟹ overfitting
- Stopping rules don't work well in practice.

Why?

(N)

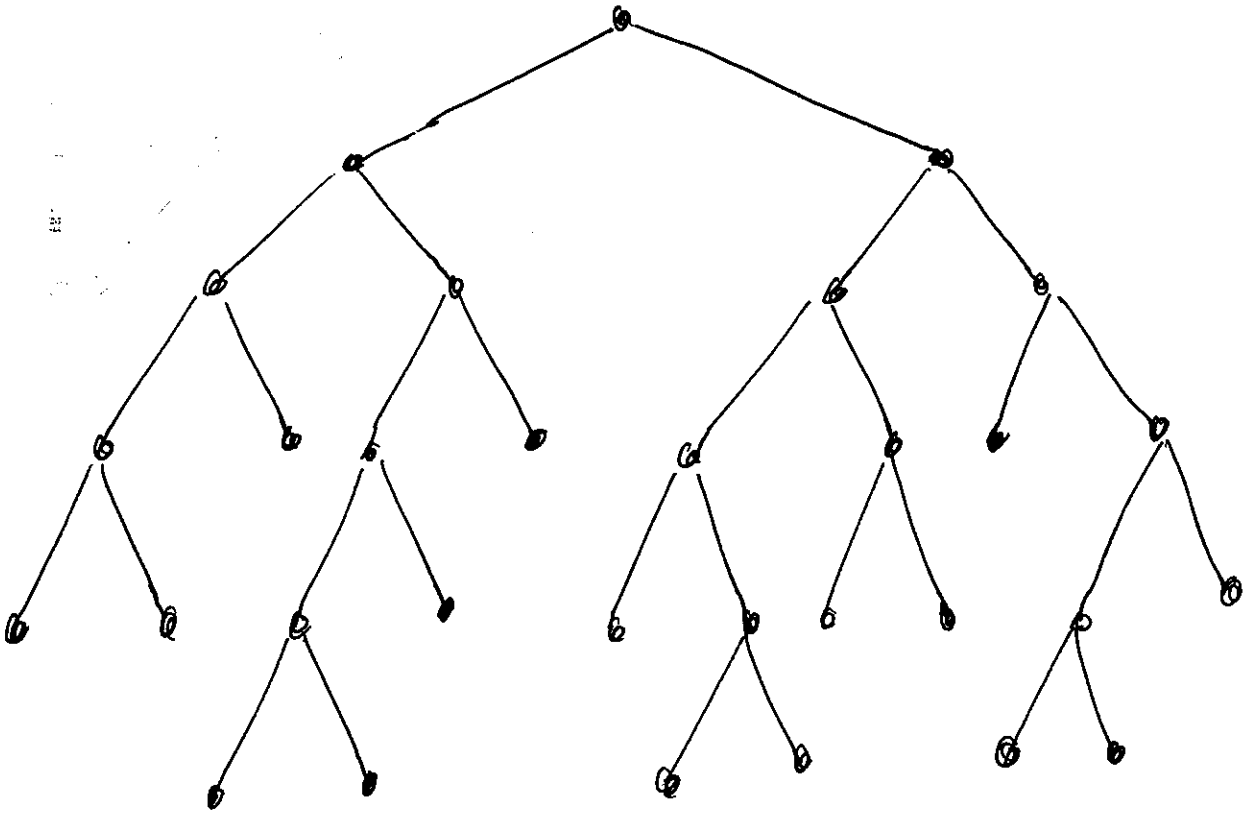
- The alternative: don't stop splitting until training data are perfectly classified. Then _____ back

Pruned Subtrees

Suppose T is a rooted binary tree.

A <u>subtree</u> is a subgraph of T which is also a tree.

A <u>pruned subtree</u> is a subtree that includes the root node.



If T' is a pruned subtree of T we'll write $T' \precsim T$

If T is a binary decision tree and $T' \preceq T$, then T' is also a binary decision tree. If we assign a label/response to each internal node of T (in addition to the leaves), then T' inherits those labels.

Cost-Complexity Pruning

In cost-complexity pruning, we select

$$T_\lambda^* = \underset{T' \preceq T}{\arg\min} \; \hat{R}(T') + \lambda \cdot |T'|$$

where

- $\lambda > 0$, fixed
- $|T'|$ = # of leaf nodes
- $\hat{R}(T') =$

(P)

CCP implements the principle of _____

_____ .

Issues

- λ is typically chosen by cross-validation
- There is an efficient algorithm for computing T_λ^* based on _____

Major Tree Programs

- CART: Classification and Regression Trees
  - Breiman, Friedman, Olshen & Stone
  - entropy impurity, CCP
- C4.5, C5.0
  - Quinlan
  - Gini index, other pruning methods

Pros & Cons of decision trees

Advantages

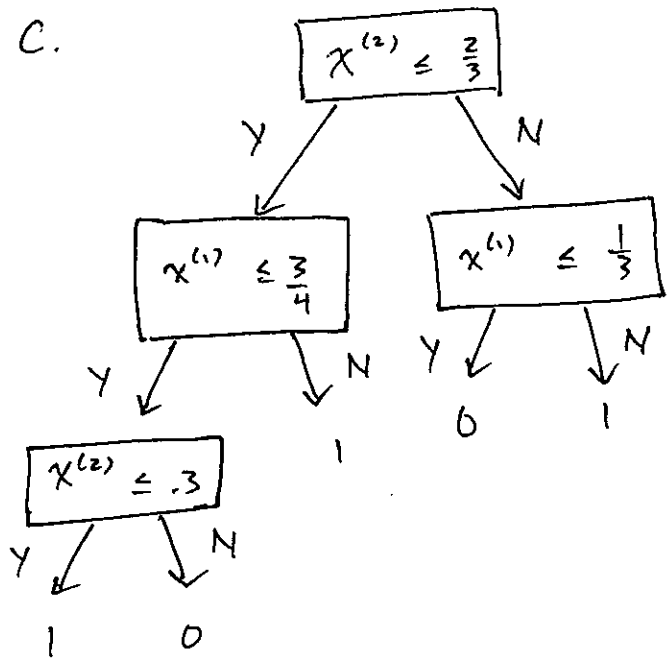
- categorical data
- multiple classes
- fast evaluation
- interpretable
- missing features

Disadvantages

- Greedy growing is suboptimal
- Design based on heuristics
- Unstable

Key A. "black box" B. y = survival time (for example)

C.



D. $x^{(2)} - .3x^{(1)} \le 4$?

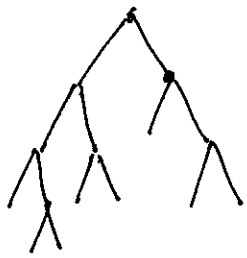
$x^{(1)} \le 0$ or

$0 \le x^{(1)} < 1$ or

$x^{(1)} \ge 1$?

E. computation,
overfitting,
lose interpretability

F. (lower right graph)



G. depth = length of path to root
parent = neighbor whose depth is one less
children = neighbors whose depth is one more
sibling = node with same parent

H. greedy I. list of possible questions / splits,
split selection criterion, stop splitting rule, labeling rule

J. homogeneous, impurity, purer

$$p(N_1) = \frac{|\{x_i \in N : x_i \in N_1\}|}{|\{x_i \in N\}|}, \quad p(N_2) = 1 - p(N_1)$$

K. $0 \le i(N) - [p(N_1) \cdot i(N_1) + p(N_2)\, i(N_2)]$

L. entropy: $i(N) = -[q \log q + (1-q) \log (1-q)]$

Gini: $i(N) = 2q(1-q)$

MC rate: $i(N) = \min(q, 1-q)$

M. Since they are <u>strictly</u> convex, the average impurity will go down

N. They don't find "ancillary" splits, i.e. splits that don't immediately lead to an improved classifier, but that are useful when combined with other splits.

O. prune

P. $\hat{R}(T') = \frac{1}{n} \sum_{i=1}^{n} 1_{\{T'(x_i) \neq y_i\}}$

Q. complexity regularization, recursion