Chapter 9: Searching & sorting 8 May 2023 MAT 2250

Searching & sorting are 2 important tools in real life.

Example: (e.g. Asthma, COPD, ...) A doctor decides the possible disease of a patient from test data. (This is a classification problem $\Rightarrow$ a searching problem.

§9.1: Search problem & Decision tree

Example: 2 people play a game.

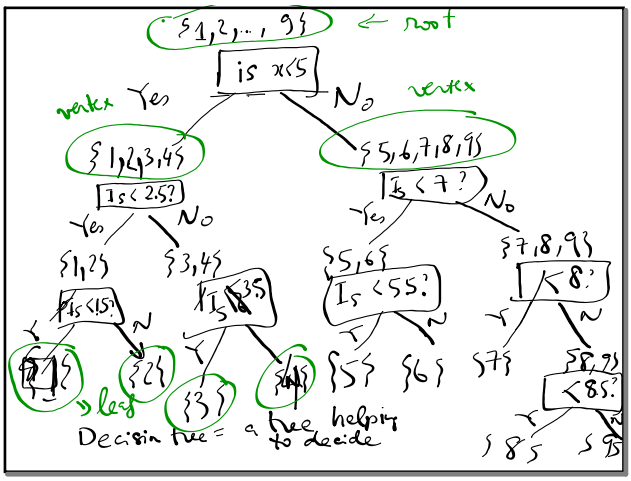
Person A thinks about an integer between 1 & 9. Person B can ask Yes/No questions, & after such questions must provide the correct answer to win.

In this case, one way to proceed is to at each step try to divide the concerned data into 2 roughly same size subsets.

Original Data $D_0 = \{1,2,3,\dots,9\}$

One way is use median

Median $(D_0) = 5$



Decision tree = a tree helping to decide

For $\{1,2,3,4,5,6,7\}$ $\Rightarrow$ need 3 questions.

For $\{8,9\}$: need 4 questions.

So the worst case is 4 questions.

The average of the questions to ask is:

$$\frac{7\times 3 + 2\times 4}{9} = \frac{29}{9} \text{ questions.}$$

(next section learn more about this average with more general probability distribution)

Rooted tree = A usual tree + a distinguished vertex called the root of the tree.

Predecessor / Immediate predecessor:
x is a predecessor of y if x belongs to the unique path from r to y.
x is an immediate predecessor of y if x is predecessor of y & $x \to y$ is an edge in the tree.
Successor / Immediate successor: similarly.
Leaf: x is a leaf if it has no successors.
(Terminal vertices)

Internal vertices: x is an internal vertex if it is not a leaf.
(That is, it has a successor.)
(Root is also an internal vertex).

Length of a vertex: x a vertex.
$l(x)$ = length of the unique path from r to x.

Length of the tree:
$L(T) = \max_{x \in T} l(x)$
(= the "worst case")

(n,q) tree: T is an (n,q) tree if:
1. T has n leaves
2. Any vertex has at most q immediate successors.

Example: The decision tree for beginning of today is a $(9,2)$ tree. $(9 = 2^3)$
binary tree.

Complete (n,q)-tree: If T is an (n,q) tree & every internal vertex has exactly q immediate successors.

Thm 9.1: If T is an (n,q) tree $(q \geq 2)$ then:
$$L(T) \geq \lceil \log_q n \rceil .$$
($\lceil x \rceil$ = the smallest integer which is bigger than or equal to x.
$\lceil 2 \rceil = 2$ $\lceil -3.9 \rceil = -3$
ceiling function $\lceil 2.1 \rceil = 3$)
Proof: ($\Rightarrow$) to showing that $q^L \geq n$.

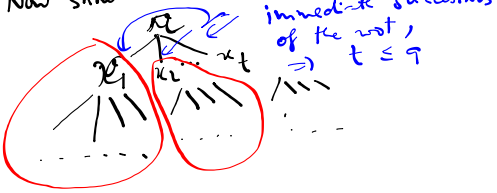
Induction on the height.

$L=0 \Rightarrow T =$ root (also leaf)

$\Rightarrow n = 1 \Rightarrow q^L = q^0 \geq 1$. (true)

Assume that it is true for all trees with height $\leq L_0$.

Now show also true for L_0+1.



Let $T(x_i)$ = the rooted tree with root x_i & all successors of x_i.

For every $y \in T$, $\exists!$ x_i such that x_i is a predecessor of y. That is $y \in T(x_i)$.

We have n leaves, & we have t subtrees $T(x_1), \ldots, T(x_t)$ containing these leaves $\Rightarrow$ $\exists$ at least one i for which the number of leaves of $T(x_i)$ is at least $\frac{n}{t} \geq \frac{n}{q}$.

Also $T(x_i)$ is an $(\frac{n}{q}, q)$ tree & $L(T(x_i))$ is $\leq L_0$.

By induction:

$q^{L(T(x_i))} \geq \frac{n}{q}$

$\Rightarrow q^{L(T)} = q^{L_0+1} \geq q^{L(T(x_i))+1}$

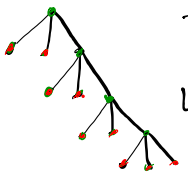
$= q \cdot q^{L(T(x_i))} = q \cdot \frac{n}{q}$

$= n$.

(Q.E.D.)

Example: The tree from beginning is a $(9,2)$ tree with height 4, & $2^4 = 16 \geq 9$.

Example:



This is a complete $(9,3)$ tree

Height = 4

$3^4 = 81 > 9$

So the Claim in Example 9.2 of the textbook seems to be incorrect.

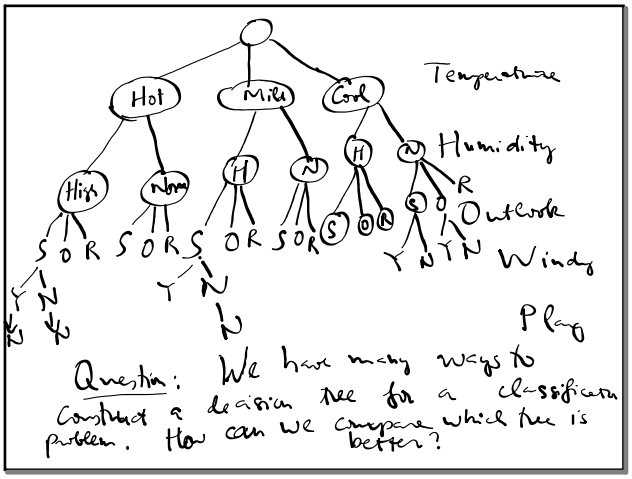
Wikipedia page on Decision tree learning.

One practical example: The weather attributes (Taken from the website Data Mining: Practical Machine Learning Tools & Techniques, + WEKA (New Zealand)

Outlook	Temp	Humidity	Windy	Play
S(sunny)	H(hot)	H(high)	F(false)	No
S	H	H	T	No
Overcast(O)	H	H	F	Yes
R(rainy)	M(mild)	H	F	Yes
R	C(cool)	N(normal)	F	Yes
R	C	N	T	No
O	C	N	T	Yes
S	M	H	F	No
S	C	N	F	Yes
R	M	N	F	Yes
S	M	N	T	Yes
O	M	H	T	Yes
O	H	N	F	Yes
R	M	H	T	No

We can construct a decision tree by going through each features (Outlook, Temp, Humidity, Windy) to get separate tree outputs (Play or not Play).

Example, if we will gradually do like: Temp, Humidity, Outlook, Windy:



Question: We have many ways to construct a decision tree for a classification problem. How can we compare which tree is better?

One idea is to use the so-called entropy. (Used by so-called ID3, C4.5 & C5.0 tree-generation algorithms)

Say we have an input data set & each with output Yes / No.

Example: In our Weather data then an input is {S, H, N, F} & output is N.

~~Entropy for data~~ is

Let p_1 = probability that the output is Yes & p_2 = probability that the output is No. (From given data).

Example : our Weather data has 14 inputs & 5 outputs N, 9 outputs Y.

$p_1 = \frac{9}{14}$, $p_2 = \frac{5}{14}$.

Entropy (Shannon): =

$- p_1 \log_2 p_1 - p_2 \log_2 p_2$

$= - \frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14}$.

The idea is to use entropy to decide which feature to go next when constructing the decision tree. (Need a more general notion of entropy for trees.)

More detail can be seen on Wikipedia page for Decision tree learning, § Information gain where this Weather data is analysed in some detail.