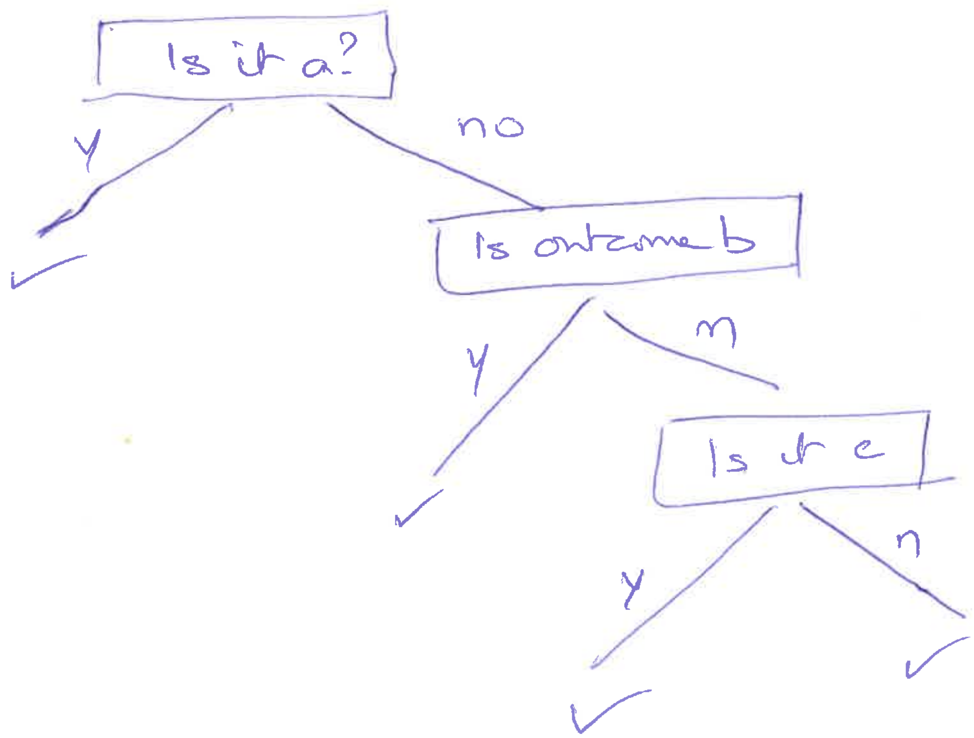
ENTROPY etc.

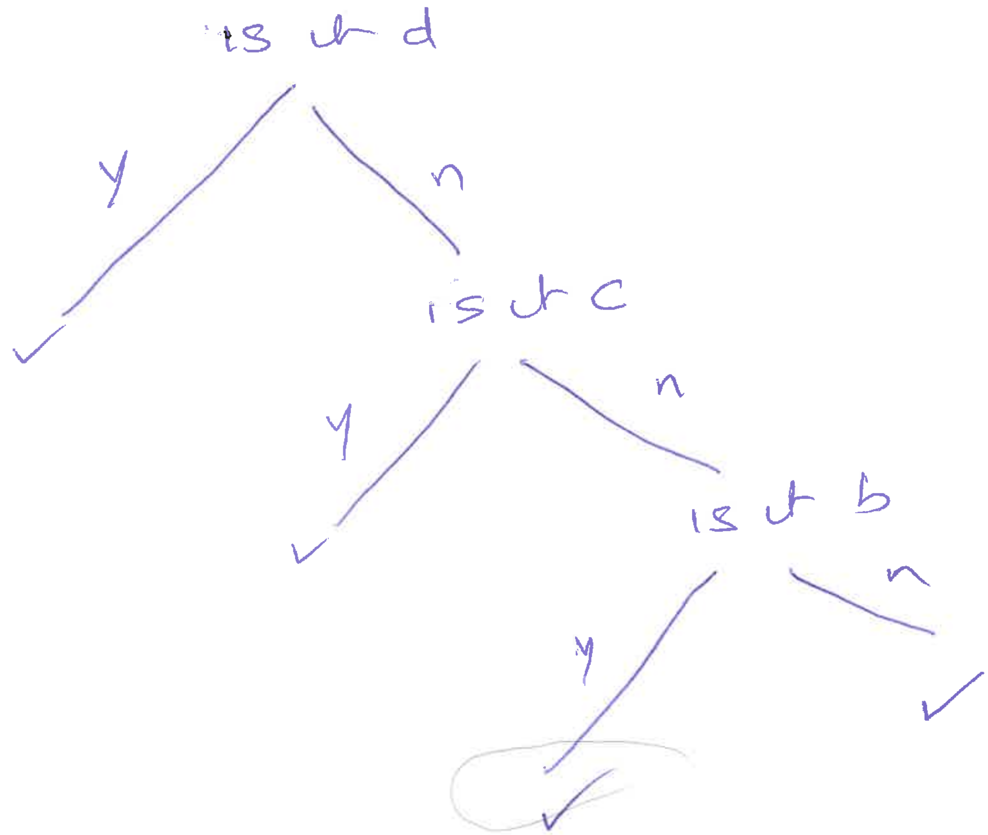
Random variable : X

Outcomes : {a, b, c, d}.

- How many yes-no questions ~~can~~ do you need to ask to know the outcome?
- What if outcomes differed in how frequently they happen?
- How many yes-no questions on an average?
- When will this be high?
- What does a low number (such as 0) mean?

$$p(x=a) = \frac{1}{2} \quad p(x=b) = \frac{1}{4} \quad p(x=c) = p(x=d) = \frac{1}{8}.$$



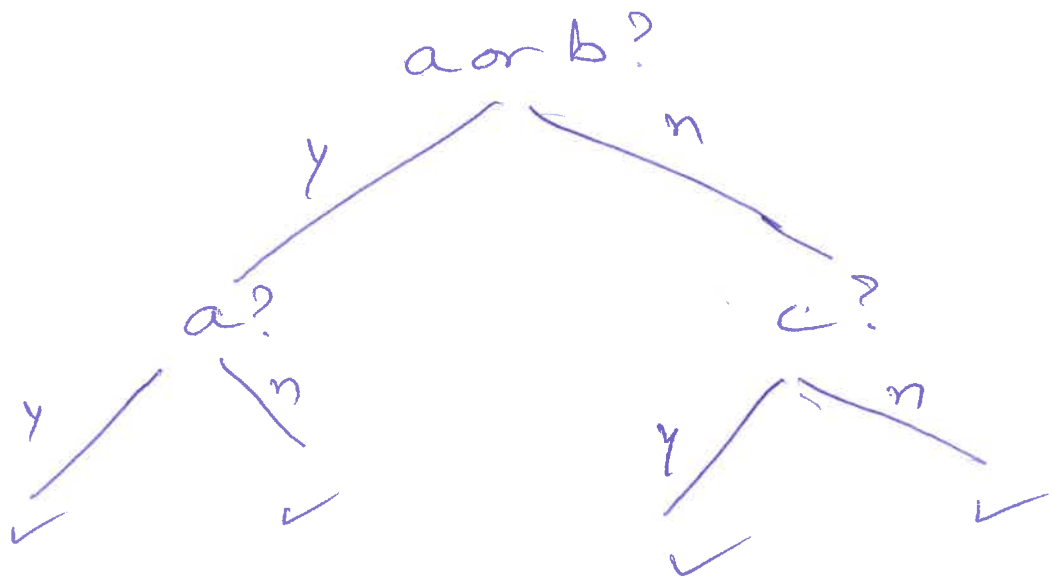


questions on an average?

Expected number of questions?

general strategy — with each question cover about half the remaining probability mass with any branch?

tree for equiprobable.



$$H(x) = \sum_{o} p(\underset{=o}{\text{outcome}})\ \#(\text{questions}) \text{ for outcome} = o$$

$$= \sum_{x} p(x) \log_2 \frac{1}{p(x)}$$

$$= -\sum_{x} p(x) \log_2 p(x)$$

Properties. 1. $H(x) \geq 0$

2. $H(x)$ is maximum when all outcomes are equiprobable.

3. $0 \leq H(x) \leq \log_2 n.$

Joint Entropy $H(X,Y)$

$$= -\sum_{x,y} p(x,y) \log p(x,y)$$

Conditional Entropy $H(X|Y)$

$$= \sum_{y} p(y)\, H(X|Y=y)$$

$$= H(X,Y) - H(Y)$$

$$p(x|y) = \frac{P(x,y)}{P(x)}$$

$$H(X|Y) = H(X,Y) - H(Y) \cancel{= H(X,Y}$$

$$H(X,Y) = H(X|Y) + H(Y)$$
$$= H(Y|X) + H(X)$$

$$H(Y) + H(X|Y) = H(Y|X) + H(X)$$

$$H(Y) - H(Y|X) = H(X) - H(X|Y)$$

Mutual Information: $I(X;Y)$

- $I(X;Y) = H(Y) - H(Y|X) = H(X) - H(X|Y)$
- reduction in uncertainty in value of Y when X is known = reduction in uncertainty in X when Y is known.
- note if Y is independent of X
  $I(X;Y) = 0$.
- note if Y completely determined by X
  $I(X;Y) = H(Y)$
- $I(X;Y) \geq 0$
  $I(X;Y) = I(Y;X)$

Assume that $p()$ & q are two probability ~~differ~~ distributions on same X.

Can we measure how similar or different are they?

Start assuming the real probability is given by $P()$ & we guess that it is given by Q.

What penalty to we pay for this incorrect assumption.

- If p & q are identical there should be no penalty.

ie $D(p \| p) = 0$.

$D(p \| q)$ — Kullback-Leibler divergence.

— KL divergence.

Since we think the probability is given by $q(\,)$, our questioning is based on $q(\,)$.

That is ~~the~~ number of questions to determine if ~~it is~~ outcome is x is

$$\log_2 \frac{1}{q(X=x)} = -\log_2 q(x)$$

Expected number of questions is

$$-\sum p(x) \log_2 q(x)$$

But if we know what the correct probability distribution was, then the ~~average~~ expected # of questions is

$$-\sum_x p(x) \log p(x).$$

To get the penalty, let's subtract:

$$D(p\|q) = -\sum_x p(x) \log q(x) - \left(-\sum_x p(x) \log p(x)\right)$$

$$= -\sum_x p(x) \log \frac{q(x)}{p(x)}$$

$$= +\sum_x p(x) \log \frac{p(x)}{q(x)}.$$

Decision Trees

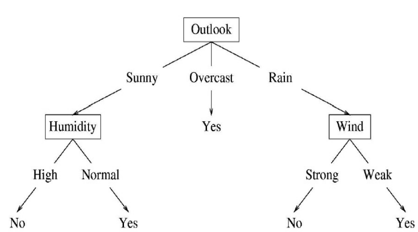
1

An Example Training Set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

2

An Example Decision Tree



3

Characteristics of Decision Trees

- Internal nodes are labeled with attribute names
- Branches (edges) from an internal node labeled by attribute A are labeled by values of attribute A.
- Leaf Nodes are labeled by target values

4

An Example Dataset

	A	B	Target
Instance 1	a1	b1	0
Instance 2	a2	b2	1
Instance 3	a1	b2	1
Instance 4	a2	b1	0

Attribute to be checked at the root? What does the tree look like?
Which tree will we build?

5

A Different Example

- Current Situation: 8 + and 8 -
- Attribute A1 has 3 values:
  - for c1: 5+, 5-; for c2: 2+, 1-; and for c3: 1+, 2-.
- For attribute A2:
  - for d1: 2+, 2-; for d2: 4+, 2-; and for d3: 2+, 4-.
- Which attribute should we choose?
  - Expected value of entropy and gain in information

6

Entropy as a measure of uncertainity

7

Decision Trees

- An n-tuple: values for n attributes. E.g., <sunny, hot, …>
- Attribute values can be categorial.
  - In perceptrons, Log Reg, MLP etc. attribute values had to be numerical.
- Unlike SVM, logistic Regression etc, we are not constrained to binary classification.
  - Regression with CART

8

PlayTennis (from Mitchell's Book)

- A running example – binary classification with 14 training instances.
- Four attributes plus *target* attribute
  - Outlook
    - Sunny, Overcast, Rain
  - Temperature
    - Hot, Mild, Cool
  - Humidity
    - High, Normal
  - Wind
    - Strong, Weak

9

Running Example

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

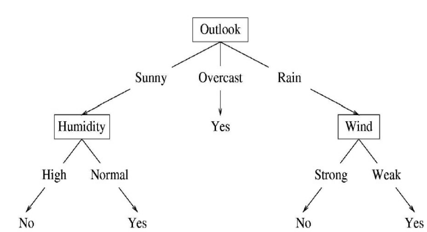
10

Decision Tree Nodes

- A decision tree classifies an instance by testing the attributes sequentially.
- Each internal (non-leaf) node will be labeled by an attribute.
- Branches of this node correspond to the possible values of the attribute.
- Leaf nodes are labeled by target values.

11

An Induced Decision Tree



- D5: Outlook=Rain, Temp=cool, humidity=normal and Wind=Weak. Prediction?
- D1: Outlook=Sunny, Temp= Hot, Humidity=High and wind=weak. ???

12

Inducing a Decision Tree

- We will discuss the classical Decision Tree Training algorithm called ID3.
- It builds the tree top-down
- The inductive bias – Make the tree as short as possible.

13

How to Train: Case 1

- Consider the node corresponding to Outlook= overcast (one level down from the root)
- Look at the instances in the training data that match this constraint. (Instances D3, D7, D12 and D13)

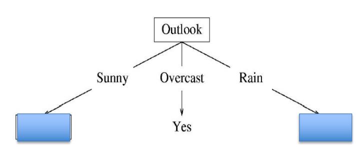
14

Running Example

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

15

Decision Tree



16

ID3 Algorithm

- ID3(S,$\mathcal{A}$)
- Create a root node
- Terminate this branch? (*overcast vs rain – next slide*)
- Otherwise
  - Choose the “best” A from $\mathcal{A}$
  - For each value v of A
    - Add new branch appropriate subset S_v of S
    - ID3(S_v, $\mathcal{A}$ - {A})
      - Sv is subset of instances of S which have v as value of A.

17

How to Train: Case 2

- Now consider outlook = Rain
- The training instances match this constraint
- The instances with Outlook = Rain are D4, D5, D6, D10, D14.
- Now we can restart decision tree induction from here but now the only instances we need to consider are given by D4, D5, D6, D10, D14.

18

Running Example

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

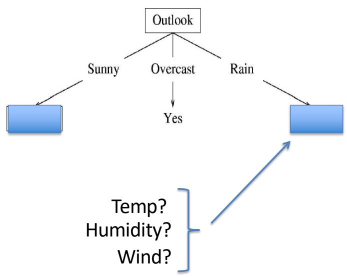
19

ID3 Algorithm

- ID3(S,$\mathcal{A}$)
- Create a root node
- Terminate this branch? (*overcast vs rain – next slide*)
- Otherwise
  - Choose the “best” A from $\mathcal{A}$
  - For each value v of A
    - Add new branch appropriate subset S_v of S
    - ID3(S_v, $\mathcal{A}$ - {A})
      - Sv is subset of instances of S which have v as value of A.

20

Decision Tree



21

Outlook = Rain

- The training instances (without Outlook) are
  - D4: Temp=mild,Humidity=high,wind=weak: YES
  - D5: Temp=cool,Humidity=normal,wind=weak: YES
  - D6: Temp=cool,Humidity=normal,wind=strong: NO
  - D10: Temp=mild,Humidity=normal,wind=weak:YES
  - D14: Temp=mild,Humidity=high,wind=strong: NO
- Temp?
  - Mild – D4, D10, D14 – 2y and 1n
  - Cool – D5, D6 – 1y and 1n

22

Outlook = Rain

- The training instances (without Outlook) are
  - D4: Temp=mild,Humi=high,wind=weak: YES
  - D5: Temp=cool,Humi=normal,wind=weak: YES
  - D6: Temp=cool,Humi=normal,wind=strong: NO
  - D10: Temp=mild,Humi=normal,wind=weak:YES
  - D14: Temp=mild,Humi=high,wind=strong: NO
- Humidity
  - High: D4: Y and D14:N
  - Normal: D5, D10: Y and D6: N

23

Outlook = Rain

- The training instances (without Outlook) are
  - D4: Temp=mild,Humi=high,wind=weak: YES
  - D5: Temp=cool,Humi=normal,wind=weak: YES
  - D6: Temp=cool,Humi=normal,wind=strong: NO
  - D10: Temp=mild,Humi=normal,wind=weak:YES
  - D14: Temp=mild,Humi=high,wind=strong: NO
- Wind
  - Weak: D4,D5, D10: Y
  - Strong: D6, D14: N

24

Choosing an Attribute

- Before looking at *wind* attribute, we were unsure (uncertain) about what the outcome (i.e., prediction) should be. Uncertainty can be measured by entropy of outcome (P(yes)= 3/5 and P(NO)=2/5).
- After knowing the value of *wind* attribute on these 5 instances, we will have no uncertainty on either branches.
- Information gain of *wind* is maximum possible since the reduction in uncertainty is highest possible.

25

What about Humidity

- Uncertainity when humidity is
  - high: entropy of P(yes) = ½ and P(no)= ½.
  - normal: entropy of P(yes) = 2/3 and P(no)= 1/3
- How do we combine these entropies?

26

Combining Entropies

- Attribute A1 has 3 choices:
  - for c1: 5+, 5-; for c2: 1+, 0-; and for c3: 0+, 1-.
- For attribute A2:
  - for d1: 3+, 3-; for d2: 3+, 0-; and for d3: 0+, 3-.
- Expected level of uncertainity is lower with A2.

27

Expected Value of Entropy

- Recall: humidity: High (D4, D14); normal (D5,D6,D10)
  - D4: Temp=mild,Humi=high,wind=weak: YES
  - D5: Temp=cool,Humi=normal,wind=weak: YES
  - D6: Temp=cool,Humi=normal,wind=strong: NO
  - D10: Temp=mild,Humi=normal,wind=weak:YES
  - D14: Temp=mild,Humi=high,wind=strong: NO
- Expected entropy, knowing value of humidity for this set is

  2/5(1/2 log2 + ½ log2) + 3/5 (2/3 log 3/2 + 1/3 log 3)
- Recall $H(X) = \Sigma_i (p_i \log 1/p_i)$

28

Information Gain Formula

- IG(S,A)= reduction in Entropy of S because of knowledge of values of A (therefore partitioning S according to this attribute).

- IG(S,A) = Entropy (S) – Σ_v ($|S_v|/|S|$) Entropy (S_v)

29

Idea behind IG(S,A) formula

- Let v be one of the possible values for A.
- S is the set of instances being considered
- Let S_v be the subset of S where instances have value v for A.
- Then we can compute the entropy of S_v based on the outcome of the distribution.
- Also we can estimate the probability an instance of S will belong to S_v. This can be computed as $|S_v|/|S|$.

30

Information Gain Formula

- IG(S,A)= reduction in Entropy of S because of knowledge of values of A (therefore partitioning S according to this attribute).

- IG(S,A) = Entropy (S) – Σ_v ($|S_v|/|S|$) Entropy (S_v)

31

Information Gain – Example 1

- S = {D4,D5,D6,D10,D14}
  - D4: Temp=mild,Humi=high,wind=weak: YES
  - D5: Temp=cool,Humi=normal,wind=weak: YES
  - D6: Temp=cool,Humi=normal,wind=strong: NO
  - D10: Temp=mild,Humi=normal,wind=weak:YES
  - D14: Temp=mild,Humi=high,wind=strong: NO

- IG(S,Humidity) = (3/5log 5/3 + 2/5 log 5/2) –
(2/5(1/2 log2 + ½ log2) + 3/5 (2/3 log 3/2 + 1/3 log 3)

32

Information Gain – Example 2

- S = {D4,D5,D6,D10,D14}
  - D4: Temp=mild,Humi=high,wind=weak: YES
  - D5: Temp=cool,Humi=normal,wind=weak: YES
  - D6: Temp=cool,Humi=normal,wind=strong: NO
  - D10: Temp=mild,Humi=normal,wind=weak:YES
  - D14: Temp=mild,Humi=high,wind=strong: NO

- IG(S,Wind) = (3/5log 5/3 + 2/5 log 5/2) – (3/5(3/3 log1 + 0 log 0) + 2/5 (0 log 0 +2/2 log 1)

33

ID3 Algorithm

- ID3(S,$\mathcal{A}$)
- Create a root node
- Time to end this branch? (next slide)
- Otherwise
  - Choose A from $\mathcal{A}$ with highest IG(S,A)
  - For each value v of A
    - Add new branch appropriate subset S_v of S
    - ID3(S_v, $\mathcal{A}$ - {A})
      - Sv is subset of instances of S which have v as value of A.

34

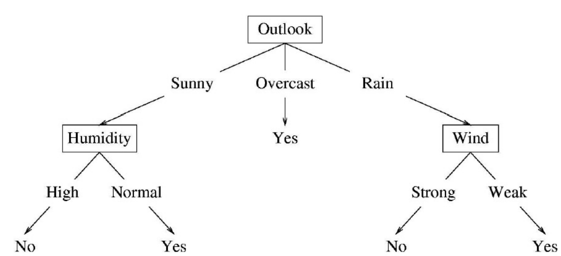
Finishing a branch

- In ID3(S, $\mathcal{A}$), we have created a node.
  - $\mathcal{A}$ Is empty (label root with most common target value)
  - All instances have same target value

    (sufficiently pure: proportion of instances in S having a value is higher than a threshold)
    - Label root with this target value

35

Resulting Tree



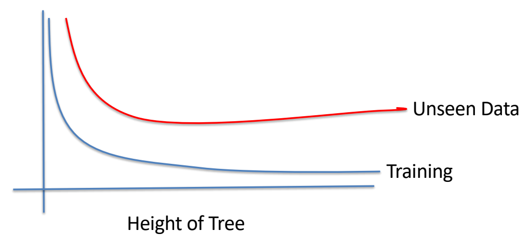
36

Tree as Set of Decision Rules

- Outlook=sunny & Humidity=high → playTennis=no
- Outlook=sunny&Humidity=normal →playTennis=yes
- Outlook = Overcast → playTennis=yes
- Outlook = Rain & Wind = strong → playTennis=no
- Outlook = Rain & Wind = weak → playTennis=yes

37

Overfitting



Number of instances at a node as tree depth increases?
Impact of Noise

38

Post-Pruning (Reduced Error Pruning)

- Pruning a (internal) node – removing subtree rooted at the node and replacing with leaf node with most common classification as label
- Use development set
- Start from leaf nodes
- Prune a node only if the resulting tree performs as good or better than original.

39

Post-Pruning Rules

- Remember each decision tree can be considered as a set of rules of the form
- If (condition 1) & … & (condition n) → class
- Pruning involves eliminating any condition
- Notice, in tree pruning we will consider the "lowest" condition first. Here any condition can be dropped.

40

Numerical Values

- Suppose Temperature value was numerical and not just hot, mild or cool.
- Suppose the 14 instances had temperature values of 90, 94, 90, 70, 54, 54, 52, 70, 56, 66, 70, 74, 94 and 66.
- Sort them: 52, 54, 56, 66,70,74,90 and 94
- Create 7 binary valued attributes based on average between consecutive values:
- temp>53, temp>55, temp>61, temp>68, temp>72, temp>82 and temp>92

41

Missing Values

Suppose there is a training instance x with a missing value, how do we compute Info Gain?

1. If S is current set of instances at node n, use the most common value for this attribute among the instances in S and use it as the value for the attribute in instance x.

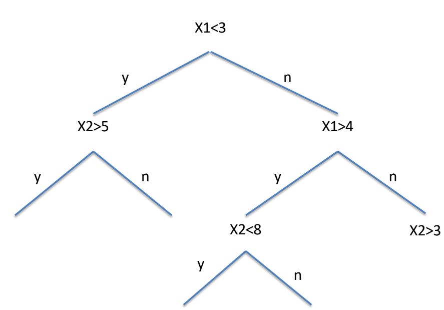
42

Missing values

- Method 2: Let |S|=s. Let the number of instances in S with value v_i of attribute is A is s_i. Then we assume that s_i/s fraction of the instance x has value v_i for the attribute for each v_i.
- For example, let a be an attribute with possible values of v_1 and v_2. Let us assume we are

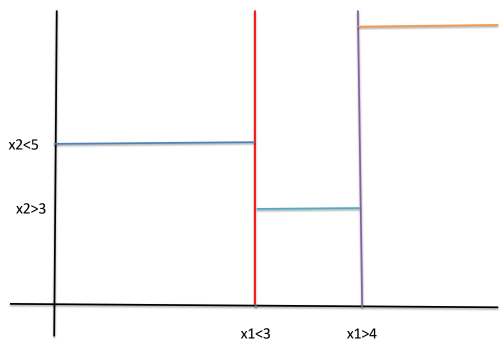
43

Decision Boundaries



44

Decision Boundaries



45

Summary

- How does a decision tree work?
  - Examine attributes sequentially.
- How to build a decision tree?
  - Select the next attribute to test.
  - ID3 algorithm: Information gain.
  - Many practical methods.

46

Summary

- Inductive bias
  - Short tree/information gain
- Overfitting
  - Pruning
- Real-valued Attributes
  - Use threshold
- Missing attribute values
  - Several common methods

47

Perceptrons

1

Modeling a Neuron

- Brain is an interconnection of nerve cells (neurons)
- Neurons have many inputs (from other neurons).
- If a neuron gets inputs such that the neuron's state value exceed a threshold, then it "fires".
- We assume that neuron's state is determined as a weighted sum of its inputs.

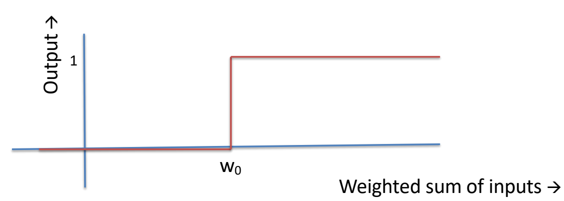
2

A Perceptron

- Has three parts:
  – Inputs
  – (Weighted) Summation
  – Threshold Function

3

Threshold Function

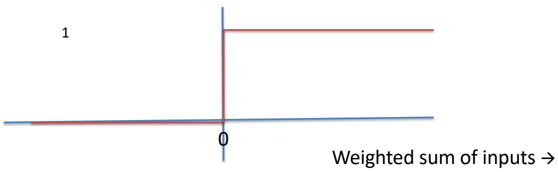


$$y = \begin{cases} 1 \text{ if } w_1x_1 + \ldots + w_kx_k > w_0 \\ 0 \text{ otherwise} \end{cases}$$

4

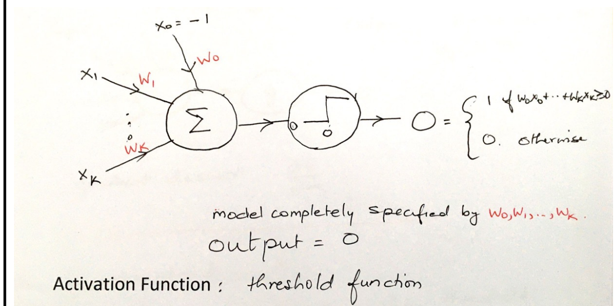
Alternatively

$$y = \begin{cases} 1 \text{ if } w_1x_1 + \ldots + w_kx_k - w_0 > 0 \\ 0 \text{ otherwise} \end{cases}$$



5

Pictorially



6

New Terms: Net and Sign

- Given an instance **x**, we define

net(**x**) = $w_0x_0 + w_1x_1 + \ldots + w_kx_k$ = **w.x**.

where $w_0 = -1$

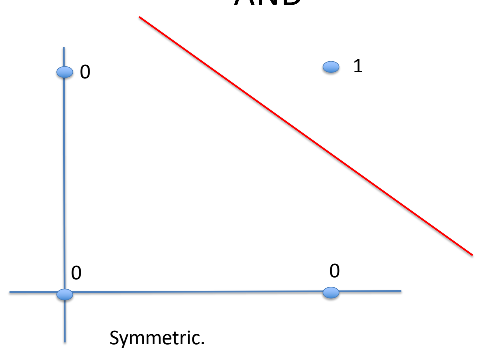
- Then output for x is

O=sign(net(x)) where

$$\text{sign}(z) = \begin{cases} 1 \text{ if } Z>0 \\ 0 \text{ otherwise} \end{cases}$$

7

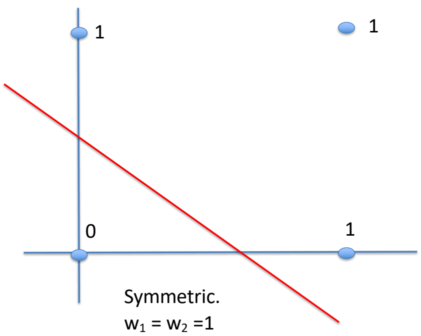
AND



Symmetric.
$w_1 = w_2 = 1$
$w_0 = 1.5$ (not - 1.5)

8

OR (Inclusive)



Symmetric.
$w_1 = w_2 = 1$
$w_0 = 0.5$ (not -0.5)

9

Negation (not)

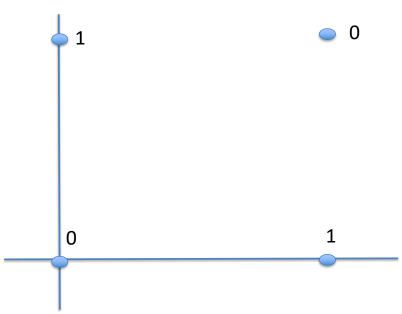
- not(0)=1 and not(1)=0
- Flip it by multiplying by -1 and use threshold of -0.5.
- W_0=-0.5 and w_1 = -1.

When $x_1 = 0$: (-0.5)(-1) + (-1).0 = 0.5-0 >0 Hence out=1

When $x_1 = 1$: (-0.5)(-1) + (-1).1 = 0.5-1 <0 Hence out=0

10

Exclusive OR (XOR)



11

Training a Perceptron

- Considering a training instance $\mathbf{x}^t$
- Based on current model ($\mathbf{w}$) let output = $o^{t.}$
- Let target value be y^t.
- If $y^t = o^t$ i.e., $(y^t – o^t)$
- No need to change the model (weights).
- Change only when $y^t – o^t > 0$ or $y^t – o^t < 0$

12

When $y^t - o^t > 0$

- y^t is fixed. Changing model can change o^t
- $o^t = \text{sign}(w_o x_0^t + w_1 x_1^t + \ldots + w_i x_i^t + \ldots w_k x_k^t)$
- Is current net^t too small or too big?
- We need to increase net^t. We will increase each of the (k+1) terms.
- We can only change the weights.
- So consider updating w_i by $w_i + \Delta w_i$.

13

Updating Weights

- Current net^t is $w_o x_0^t + w_1 x_1^t + \ldots + w_i x_i^t + \ldots w_k x_k^t$
- New $net^t = \ldots + (w_i + \Delta w_i)\, x_i^t + \ldots$
- New $net^t = \ldots + (w_i x_i^t + \Delta w_i x_i^t) + \ldots$

$(y^t - o^t)$	$w_i x_i^t + \Delta w_i x_i^t$	x_i^t	Δw_i	$(y^t - o^t)\, x_i^t$
positive	increase	positive	positive	positive
positive	increase	negative	negative	negative
negative	decrease	positive	negative	negative
negative	decrease	negative	positive	positive

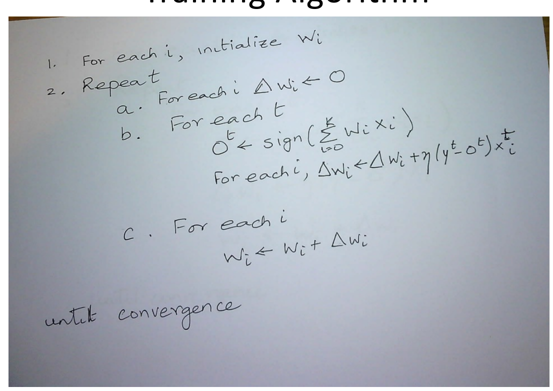
14

Perceptron Update Rule

- $\Delta w_i = \eta\, (y^t - o^t)\, x_i^t$
- Perceptron Update Rule:
  - update w_i by $w_i + \Delta w$
  - $w_i \leftarrow w_i + \eta\, (y^t - o^t)\, x_i^t$
- We are not able to use gradient descent and yet we have something like linear regression updates.
- Difference is that both y^t and o^t are 0/1.

15

Training Algorithm



16

Training Algorithm (Stochastic)

1. For each i initialize w_i
2. Repeat

   For each t

   $o^t \leftarrow \text{sign}\left(\sum_{i=0}^{K} w_i x_i\right)$

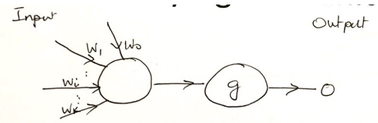
   $\Delta w_i \leftarrow \eta (y^t - o^t) x_i^t$

   $w_i \leftarrow w_i + \Delta w_i$

until convergence.

17

Why sigmoid unit



Error determined by $(Y-O)$

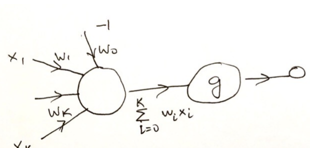
gradient descent $\frac{\partial E}{\partial w_i}$

$g =$ threshold ×

$g =$ sigmoid function. looks like threshold.

18

Activation Function and Types of Units



$O = g\left(\sum_{i=0}^{K} w_i x_i\right)$

① linear unit $g =$ identity $O = \sum_{i=0}^{K} w_i x_i$ (regression)

② thresholded unit $O = I\left(\sum_{i=0}^{K} w_i x_i \geq 0\right)$

③ sigmoid unit $O = \sigma\left(\sum_{i=0}^{K} w_i x_i\right)$

19