CS 4824/ECE 4424: Function Approximation

Acknowledgement:
Many of these slides are derived from Tom Mitchell, Pascal Poupart, Pieter Abbeel, Eric Eaton, Carlos Guestrin, William Cohen, and Andrew Moore.

Supervised function approximation

- Problem setting
  - Set of possible instances $\boldsymbol{X}$
  - Unknown target function $\boldsymbol{f}$
  - Set of function hypotheses: $\boldsymbol{H} = \{h \mid h: \boldsymbol{X} \rightarrow \boldsymbol{Y}\}$

- Input
  - Training examples $\{<X^{(i)}, Y^{(i)}>\}$ of unknown function $\boldsymbol{f}$

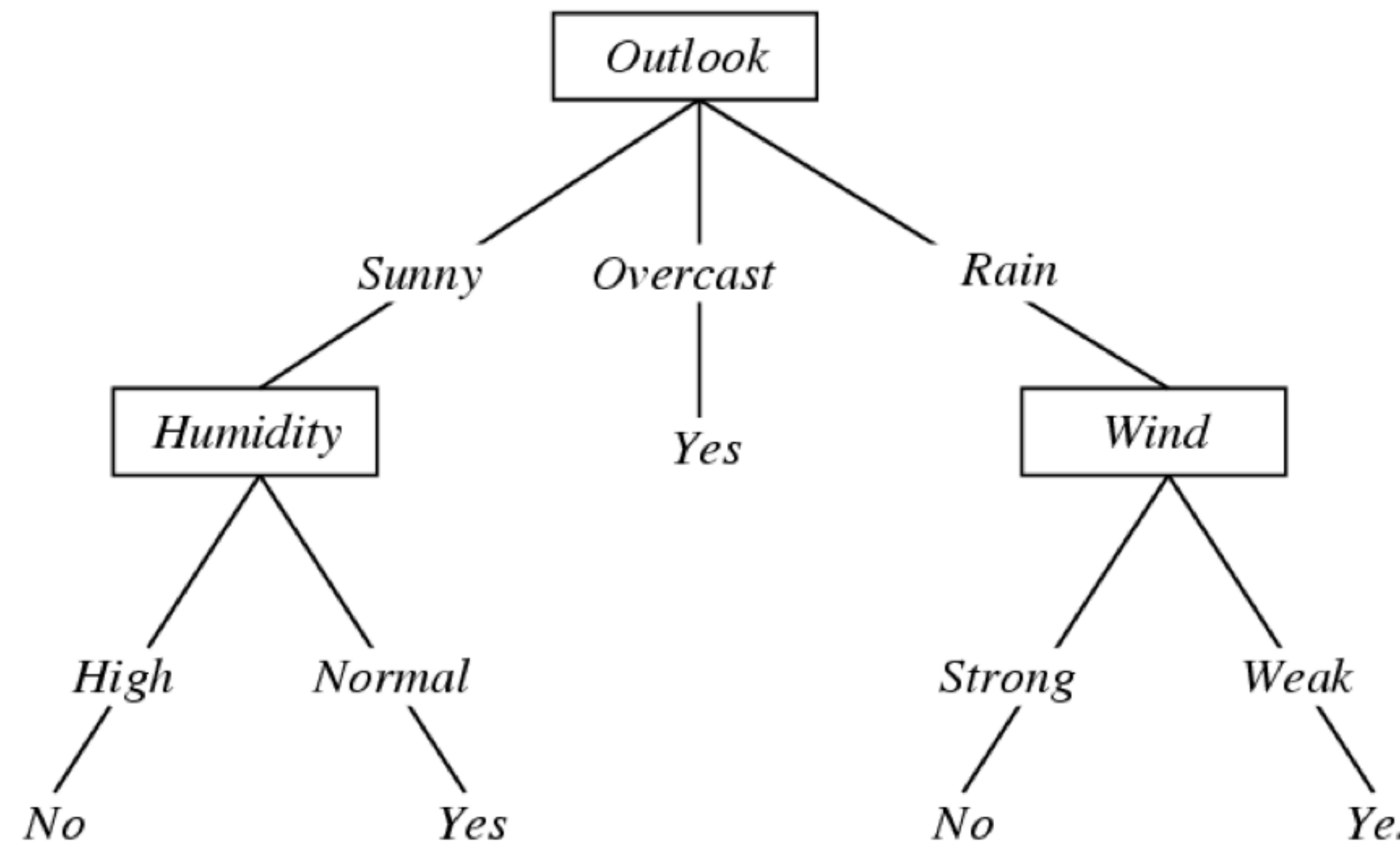
- Output
  - Hypothesis $h \in \boldsymbol{H}$ that best approximates $\boldsymbol{f}$

Example data

Day	Outlook	Temperature	Humidity	Wind	PlayTen
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

Example function approximator

- Each internal node
  - Tests one attribute X_i
- Each branch from a node
  - Selects on value for X_i
- Each leaf node:
  - Predicts Y or P(Y | X ∈ leaf)



A decision tree for

F: <Outlook, Humidity, Wind, Temp→PlayTennis?>

Dynamics of the function approximator

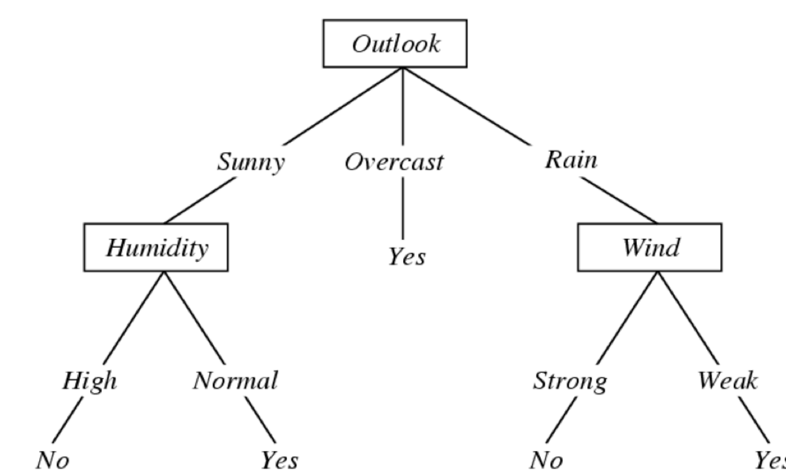
- Set of possible instances $\boldsymbol{X}$
  - Each instance x is a feature vector

- Unknown target function $\boldsymbol{f}$
  - Y is discrete valued

- Set of function hypotheses: $\boldsymbol{H} = \{h \mid h: \boldsymbol{X} \rightarrow \boldsymbol{Y}\}$
  - Each hypothesis h is a decision tree
  - Tree sorts x to leaf, which assigns y

Dynamics of the function approximator

- Set of possible instances ***X***
  - Each instance *x* is a feature vector
- Unknown target function ***f***
  - Y is discrete valued
- Set of function hypotheses: ***H*** = {*h* | *h*: ***X*** → ***Y***}
  - Each hypothesis *h* is a decision tree
  - Tree sorts *x* to leaf, which assigns *y*



Q. How many decision trees are possible?

Function approximation using decision trees

- Suppose $X = <X_1, \ldots X_n>$, $X_i \in \{0, 1\}$
- How would you represent $Y = X_2 X_5$? $Y = X_2 \vee X_5$?
- Or a more complicated one $X_2 X_5 \vee X_3 X_4 (\neg X_1)$?

Function approximation using decision trees

Q. Can we represent arbitrary boolean (or discrete-valued) functions using decision trees?

Decision tree as function approximator

- Decision trees are expressive
  - Can represent any Boolean (or discrete-valued) functions
  - This makes decision trees **universal function approximator**

Top-down induction of decision trees

node = Root

Main loop:

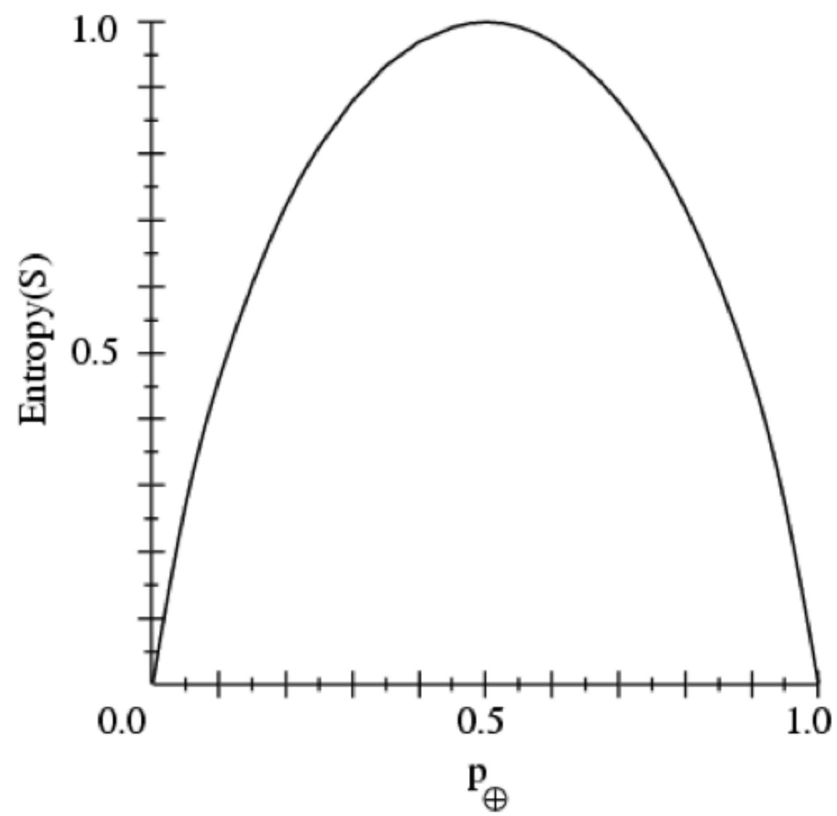
1. $A \leftarrow$ the "best" decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A, create new descendant of *node*
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

ID3, C4.5

Intuition: top-down greedy growth of decision tree using "best" attribute until all examples are perfectly classified.

Q. How to pick "best" attribute?

Sample Entropy



- S is a sample of training examples
- $p_\oplus$ is the proportion of positive examples in S
- $p_\ominus$ is the proportion of negative examples in S
- Entropy measures the impurity of S

$$H(S) \equiv -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$

Entropy

- Entropy $H(X)$ of a random variable X is defined as:
  - $H(X) = -\sum_i P(X=i) \log_2 P(X=i)$

- Specific conditional entropy $H(X|Y=v)$ is
  - $H(X|Y=v) = -\sum_i P(X=i \mid Y=v) \log_2 P(X=i \mid Y=v)$

- Conditional entropy $H(X|Y)$ is
  - $H(X|Y) = \sum_{v \in values(Y)} P(Y=v)\, H(X|Y=v)$

- Mutual information (a.k.a. information gain) of X and Y
  - $I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$

Information gain

- Mutual information (a.k.a. information gain) of X and Y
  - $I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$

- Information Gain is the expected reduction in entropy of target variable Y for data sample S, due to sorting on variable A
  - $Gain\ (S,A) = I_s(A, Y) = H_S(Y) - H_S(Y|A)$

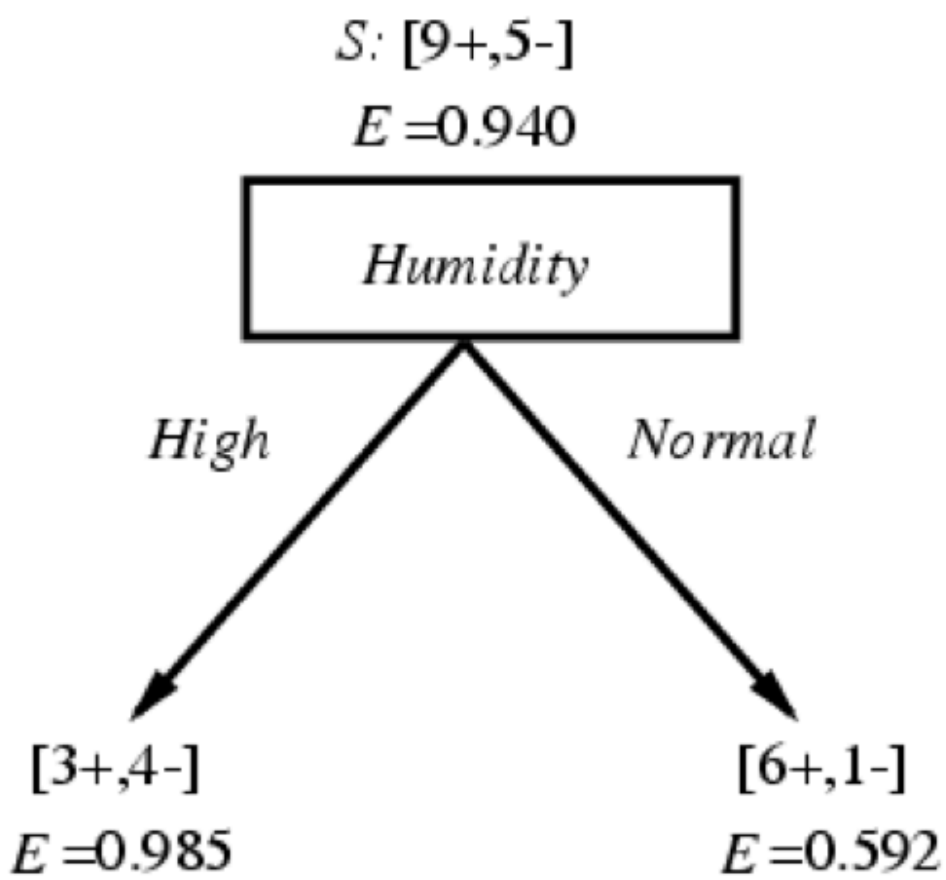
Q. How to pick "best" attribute?

A. One that reduces entropy the most. i.e. highest information gain

Example data

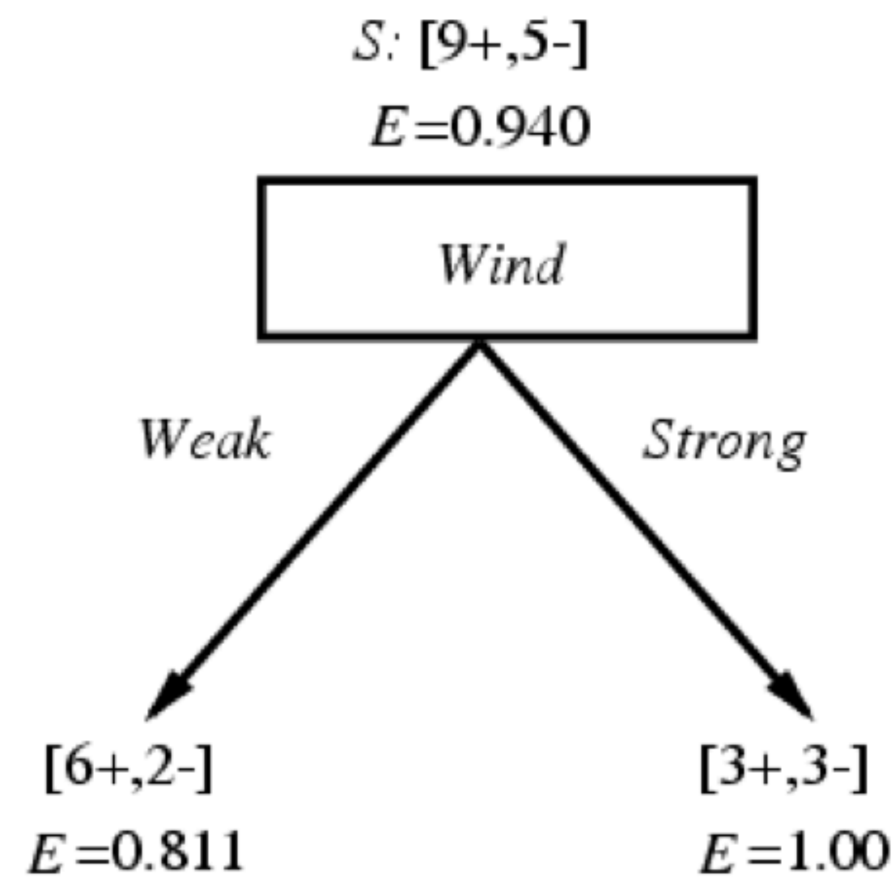
Day	Outlook	Temperature	Humidity	Wind	PlayTen
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	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

Selecting the "best" attribute



Gain (S, Humidity)

= .940 - (7/14).985 - (7/14).592
= .151



Gain (S, Wind)

= .940 - (8/14).811 - (6/14)1.0
= .048

Questions to think about…

Is there more than one decision tree that will perfectly sort the data?

If so, which one do you choose and why?