CMSC 726
Lecture 4:Decision Trees

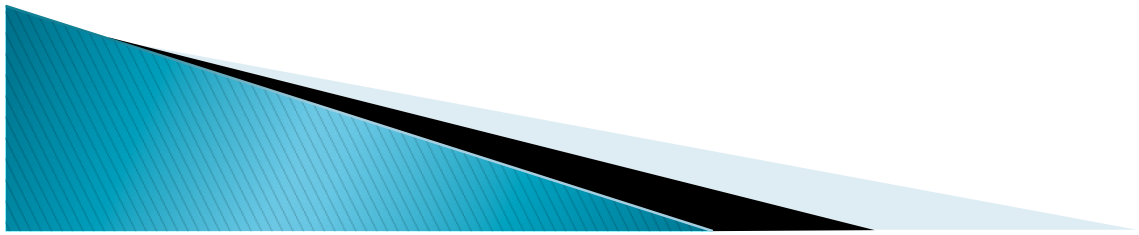
Lise Getoor

September 9, 2010

ACKNOWLEDGEMENTS: The material in this course is a synthesis of materials from many sources, including: Hal Daume III, **Mark Drezde**, Carlos Guestrin, Andrew Ng, Ben Taskar, Eric Xing, and others. I am very grateful for their generous sharing of insights and materials.

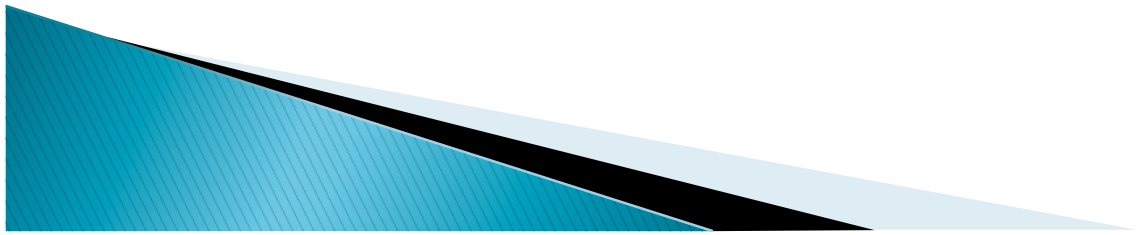
Today's Topics

- Decision Trees



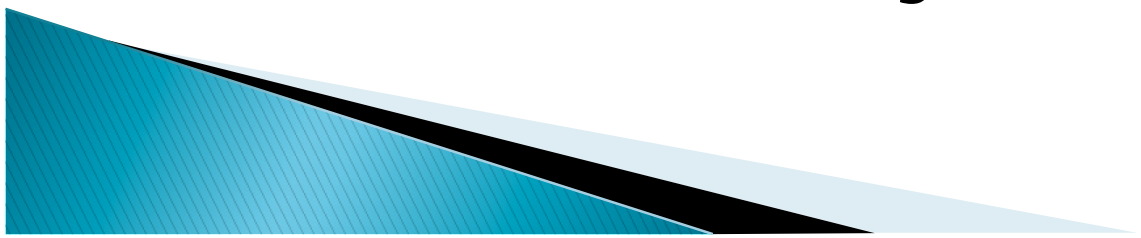
Decision Trees

- Why start with decision trees?
  - Decision trees have a long history in ML
    - First popular algorithms 1979
  - Very popular in many real world settings
  - Intuitive to understand
  - Easy to build



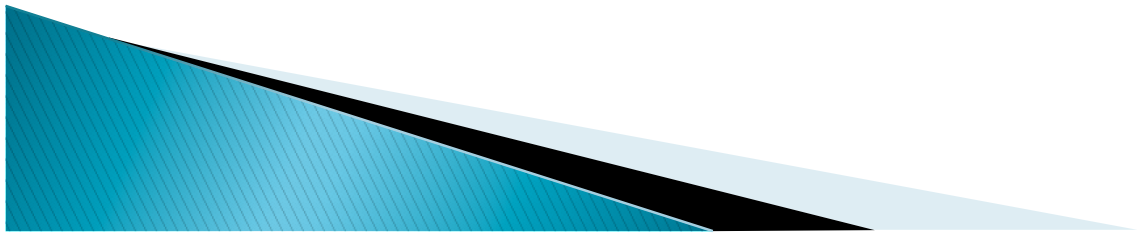
History

- EPAM- Elementary Perceiver and Memorizer
  - Feigenbaum 1961
  - Cognitive simulation model of human concept learning
- CLS- early algorithm for decision tree construction
  - Hunt 1966
- ID3 based on information theory
  - Quinlan 1979
- C4.5 improved over ID3
  - Quinlan 1993
- Also has long, rich history in statistics as CART (Classification And Regression Trees)



Motivation

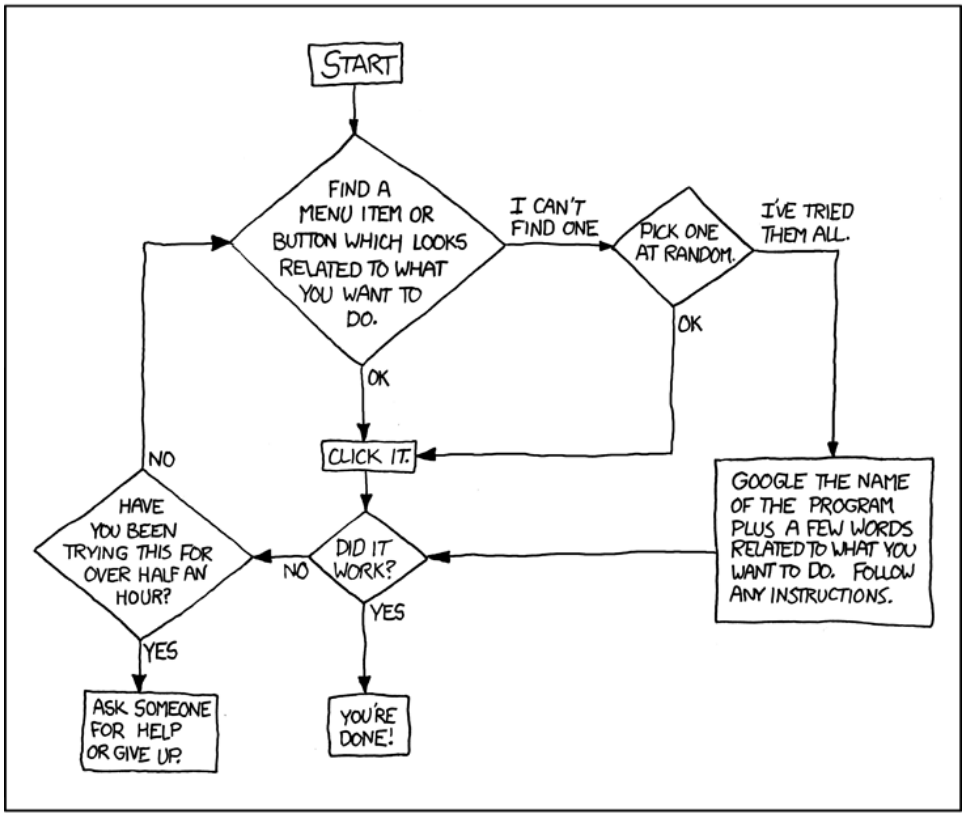
- How do people make decisions?
  - Consider a variety of factors
  - Follow a logical path of checks
- Should I eat at this restaurant?
  - If there is no wait
    - Yes
  - If there is short wait and I am hungry
    - Yes
  - Else
    - No



Decision Graph Example

DEAR VARIOUS PARENTS, GRANDPARENTS, CO-WORKERS, AND OTHER "NOT COMPUTER PEOPLE."

WE DON'T MAGICALLY KNOW HOW TO DO EVERYTHING IN EVERY PROGRAM. WHEN WE HELP YOU, WE'RE USUALLY JUST DOING THIS:



PLEASE PRINT THIS FLOWCHART OUT AND TAPE IT NEAR YOUR SCREEN.
CONGRATULATIONS; YOU'RE NOW THE LOCAL COMPUTER EXPERT!

http://www.xkcd.com/627/

Example: Should We Play Tennis?

Play Tennis	Outlook	Temperature	Humidity	Windy
No	Sunny	Hot	High	No
No	Sunny	Hot	High	Yes
Yes	Overcast	Hot	High	No
Yes	Rainy	Mild	High	No
Yes	Rainy	Cold	Normal	No

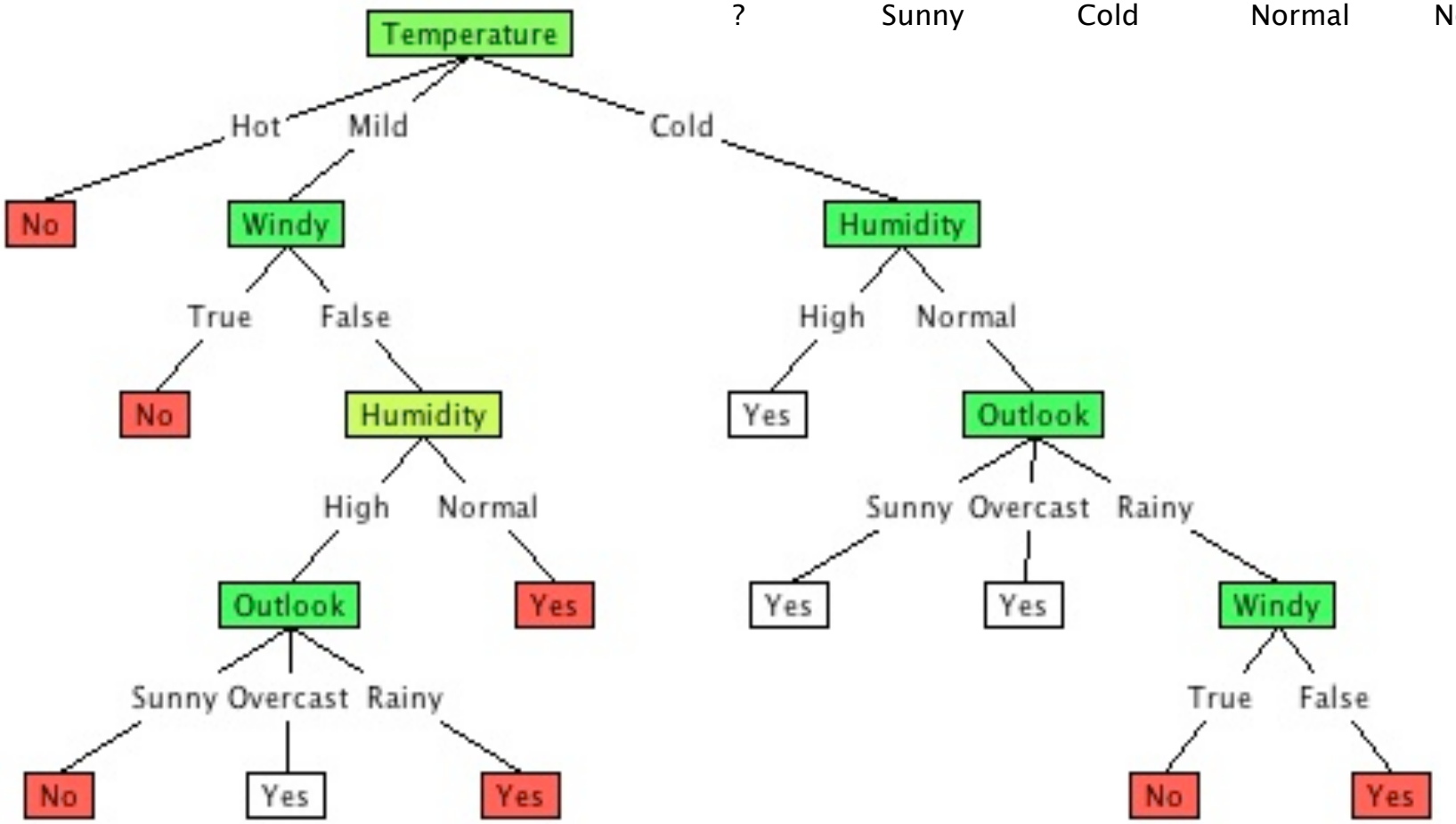
y

x

- If temperature is not hot
  - Play tennis
- If outlook is overcast
  - Play tennis
- Otherwise
  - Don't play tennis

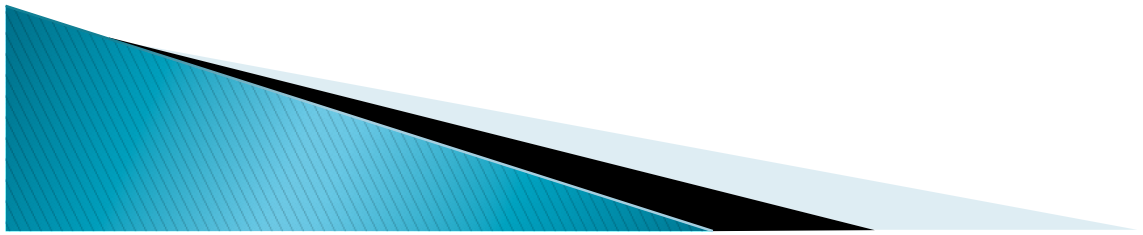
Decision Tree

Play Tennis	Outlook	Temperature	Humidity	Windy
?	Sunny	Cold	Normal	No



Decision Trees

- A decision tree is formed of
  - Nodes
    - Attribute tests
  - Branches
    - Results of attribute tests
  - Leaves
    - Classifications

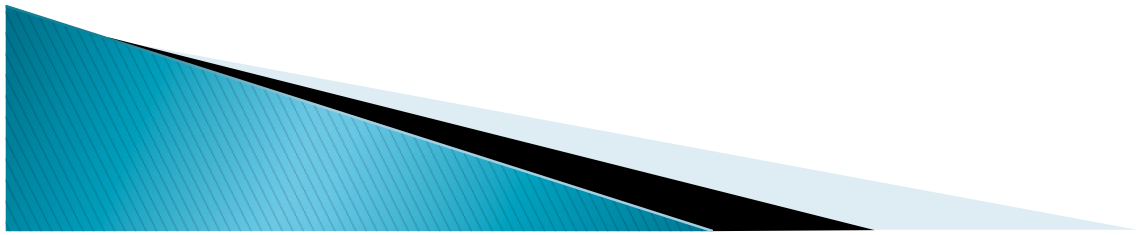


Hypothesis Class

- What functions can decision trees model?
  - Non-linear: very powerful hypothesis class
  - A decision tree can encode *any Boolean function*
  - Proof
    - Given a truth table for a function
    - Construct a path in the tree for each row of the table
    - Given a row as input, follow that path to the desired leaf (output)

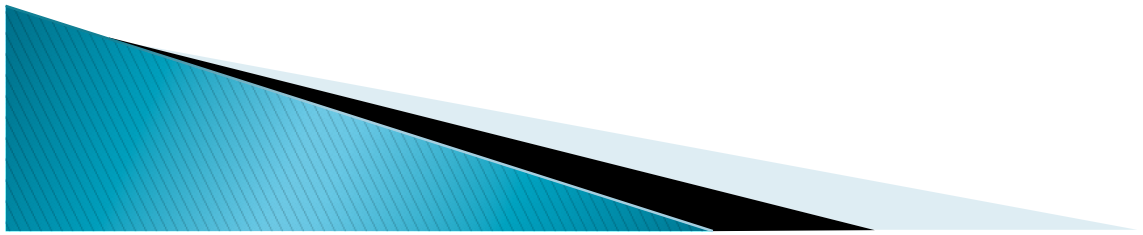
Y	X_1	X_2	X_3
1	0	0	0
0	0	0	1
1	0	1	0

- Problem:
  - exponentially large trees!



Smaller Trees

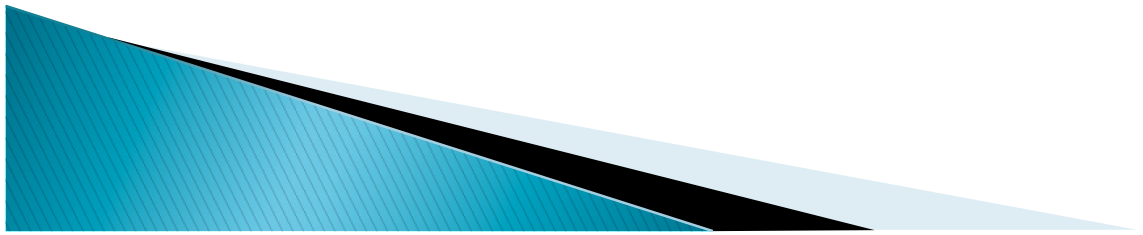
- Can we produce smaller decision trees for functions?
  - Yes (most of the time)
  - Counter examples
    - Parity function
      - Return 1 on even inputs, 0 on odd inputs
- Decision trees are good for some functions but bad for others
- Tradeoff between hypothesis class expressiveness and learnability



Building Decision Trees

What Makes a Good Tree?

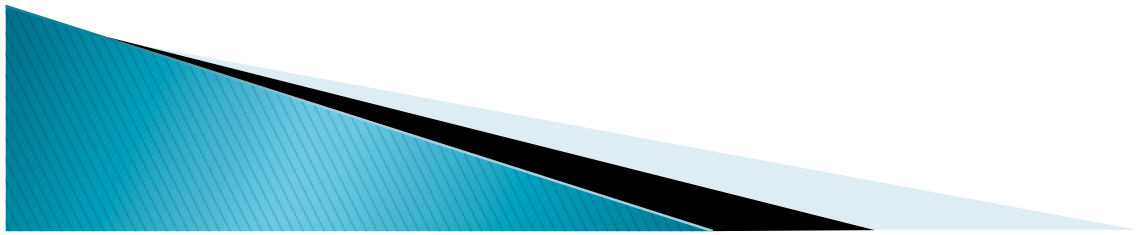
- Small
  - Ockham's razor
    - Simpler is better
  - Avoids over-fitting
- A decision tree may be human readable, but not use human logic
  - The decision tree you would write for a problem may differ from what is generated by a decision tree learning algorithm



Small Trees

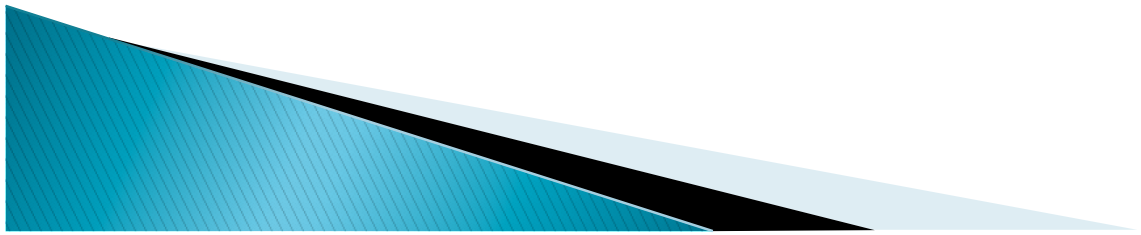
- How do we build small trees that accurately capture data?
- Optimal decision tree learning is NP-complete

- Constructing Optimal Binary Decision Trees is NP-complete. Laurent Hyafil, RL Rivest. Information Processing Letters, Vol. 5, No. 1. (1976), pp. 15-17.



Greedy Algorithms

- Like many NP-complete problems we can get pretty good solutions
- Most decision tree learning is by greedy algorithms
  - Adjustments are usually to fix greedy selection problems
- Top down decision tree learning
  - Recursive algorithms



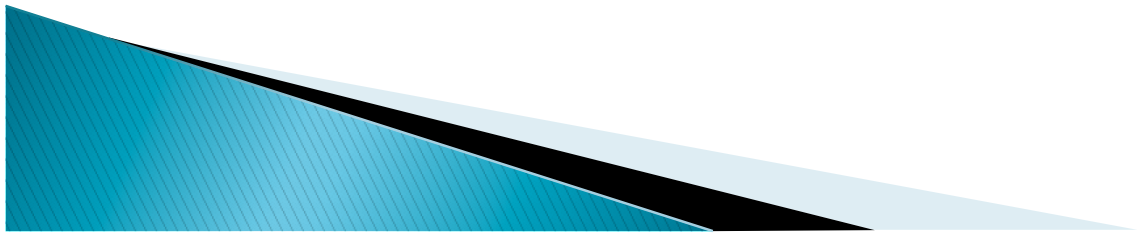
ID3

- function BuildDecisionTree(data, labels):
  - if all labels are the same
    - return leaf node for that label
  - else
    - let f be the best feature for splitting
    - left = BuildDecisionTree(data with f=0, labels with f=0)
    - right= BuildDecisionTree(data with f=1, labels with f=1)
    - return Tree(f, left, right)

Does this always terminate?

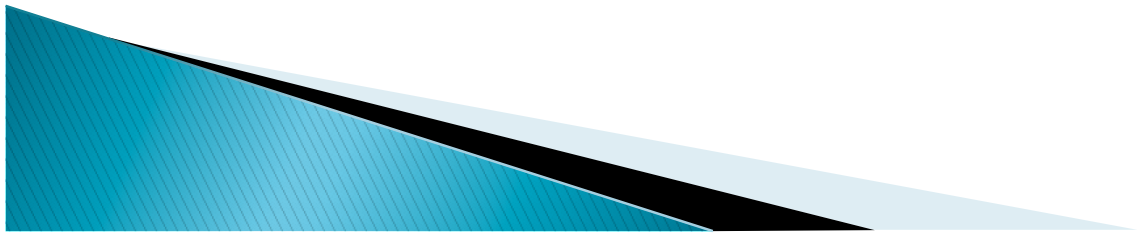
Base Cases

- All data have same label
  - Return that label
- No examples
  - Return majority label of all data
- No further splits possible
  - Return majority label of passed data



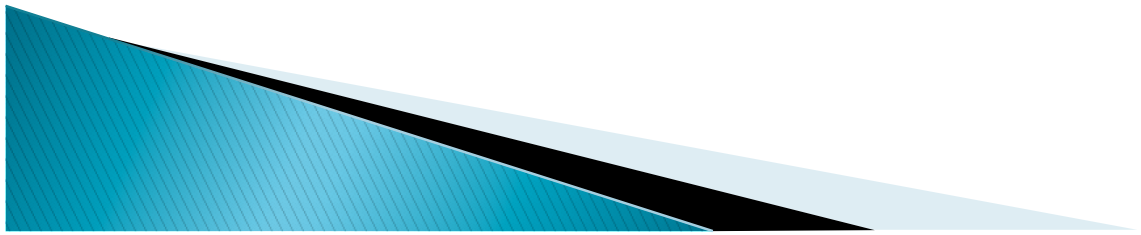
ID3

- function BuildDecisionTree(data, labels):
  - if all labels are the same
    - return leaf node for that label
  - else
    - let f be the best feature for splitting
    - left = BuildDecisionTree(data with f=0, labels with f=0)
    - right= BuildDecisionTree(data with f=1, labels with f=1)
    - return Tree(f, left, right)



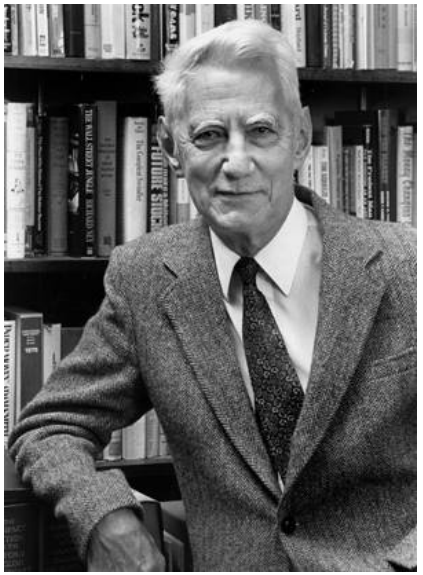
Selecting Features

- The best feature for splitting
  - The most *informative feature*
  - Select the feature that is most informative about the labels
- Information theory



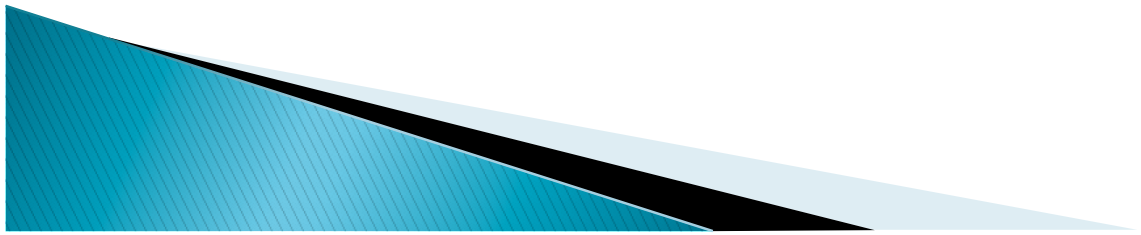
Information Theory

- The quantification of information
- Founded by Claude Shannon
  - Landmark paper in 1948
  - Noisy channel theorem



Information Theory

- A brief introduction…

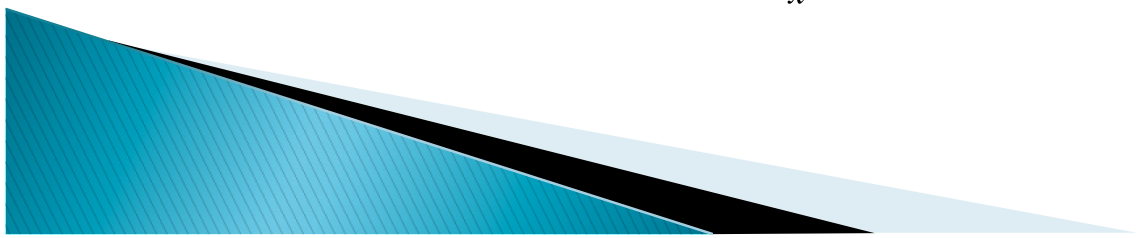


Information Theory (PRML 1.6)

- In information theory, **entropy** is a measure of the uncertainty associated with a random variable. It quantifies, using expected value, the information contained in a message in bits. Equivalently, it is a measure of the average information content one is missing when one does not know the value of the random variable (Wikipedia)

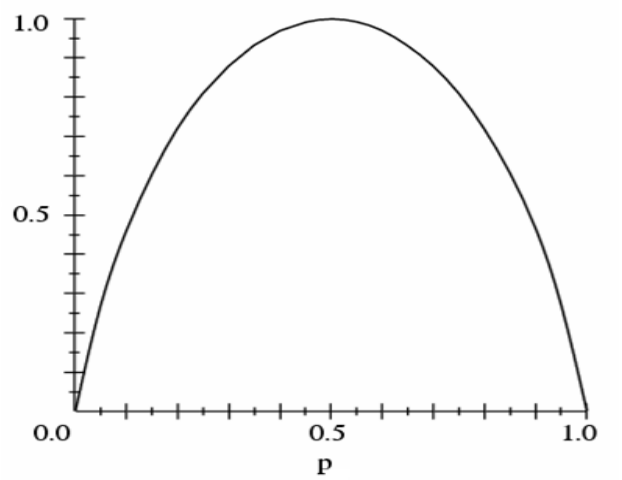
- Definition:

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



Binary Entropy Example

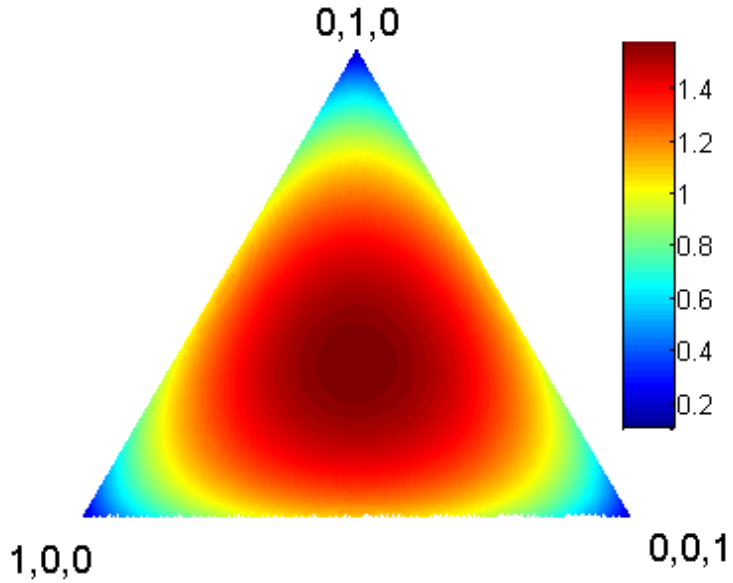
- For binary Y, the entropy is a function of $p=P(Y=1)$, $H(Y)=-p\log_2 p-(1-p)\log_2(1-p)$ so we can plot it:



- Note that entropy is zero when $p = 0$ or $p = 1$, and 1 when $p = 0.5$. One interpretation of entropy is the expected number of bits needed to encode Y or questions needed to guess Y.

Ternary Entropy Example

- Entropy of Y w/ 3 outcomes w/ prob. p1 + p2 + p3 = 1:



- The corners correspond to the distributions that put all the weight on one of 3 outcomes (the entropy is zero there) and the center corresponds to the uniform distribution (1/3,1/3,1/3) which achieves the entropy of 1.585.

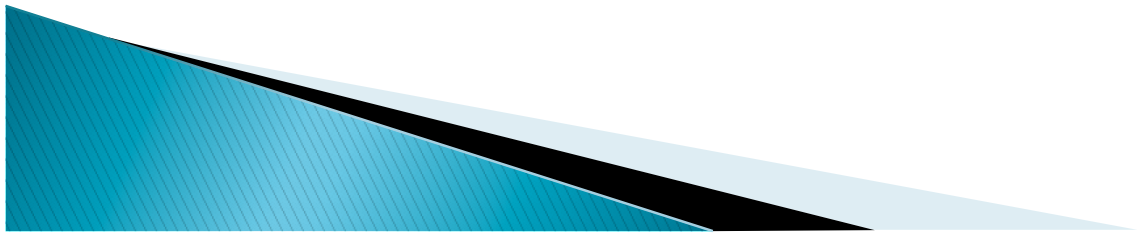
Conditional Entropy & IG

- To quantify predictiveness of a feature X for Y, we consider the *conditional entropy*, or the expected number of bits needed to encode Y or questions needed to guess Y, **knowing** X.

$$H(Y \mid X) = \sum_{x} p(X = x) H(Y \mid x = x)$$

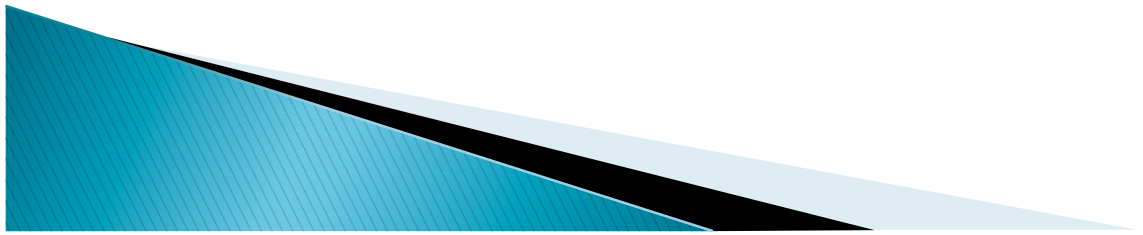
- So to measure the reduction in entropy of Y from knowing X, we use *information gain*:

$$IG(Y \mid X) = H(Y) - H(Y \mid X)$$



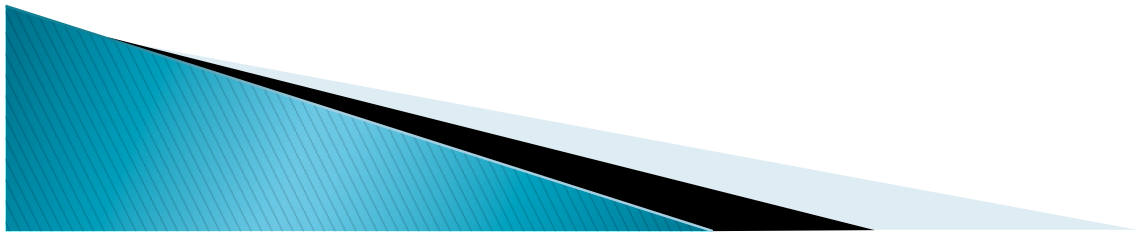
Selecting Features

- The best feature for splitting
  - The most *informative feature*
  - The feature with the highest *information gain*



Notes for Decision Trees

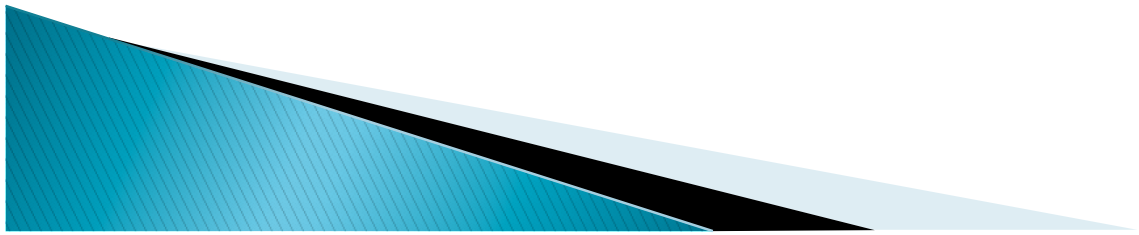
- Since we compare H(Y|X) across all features, H(Y) is a constant
  - We can omit it for comparisons
- The base of the log doesn't matter as long as it is consistent



Example: Should We Play Tennis?

Play Tennis	Outlook	Temperature	Humidity	Windy
No	Sunny	Hot	High	No
No	Sunny	Hot	High	Yes
Yes	Overcast	Hot	High	No
Yes	Rainy	Mild	High	No
Yes	Rainy	Cold	Normal	No

- $H(\text{Tennis}) = -3/5 \log_2 3/5 - 2/5 \log_2 2/5 = 0.97$



Example: Should We Play Tennis?

Play Tennis	Outlook	Temperature	Humidity	Windy
No	Sunny	Hot	High	No
No	Sunny	Hot	High	Yes
Yes	Overcast	Hot	High	No
Yes	Rainy	Mild	High	No
Yes	Rainy	Cold	Normal	No

- $H(\text{Tennis}|\text{Outlook}=\text{Sunny}) = -2/2 \log_2 2/2 - 0/2 \log_2 0/2 = 0$
- $H(\text{Tennis}|\text{Outlook}=\text{Overcast}) = -0/1 \log_2 0/1 - 1/1 \log_2 1/1 = 0$
- $H(\text{Tennis}|\text{Outlook}=\text{Rainy}) = -0/2 \log_2 0/2 - 2/2 \log_2 2/2 = 0$
- $H(\text{Tennis}|\text{Outlook}) = 2/5 * 0 + 1/5 * 0 + 2/5 * 0 = 0$

Example: Should We Play Tennis?

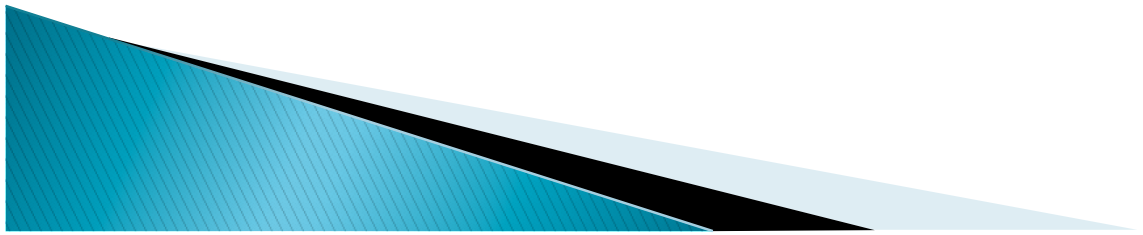
Play Tennis	Outlook	Temperature	Humidity	Windy
No	Sunny	Hot	High	No
No	Sunny	Hot	High	Yes
Yes	Overcast	Hot	High	No
Yes	Rainy	Mild	High	No
Yes	Rainy	Cold	Normal	No

- IG(Tennis|Outlook) = 0.97 – 0 = 0.97
- If we knew the Outlook we'd be able to perfectly predict Tennis!

- Outlook is a great feature to pick for our decision tree

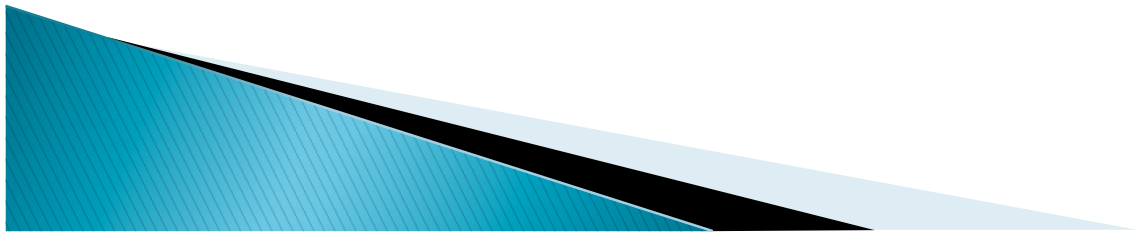
ID3

- function BuildDecisionTree(data, labels):
  - if basecase
    - return appropriate leaf node
  - Else
    - f = arg max IG(label|f)
    - left = BuildDecisionTree(data with f=0, labels with f=0)
    - right= BuildDecisionTree(data with f=1, labels with f=1)
    - return Tree(f, left, right)



Base Cases

- All data have same label
  - Return that label
- No examples
  - Return majority label of all data
- No further splits possible
  - Return majority label of passed data
- If max IG = 0?

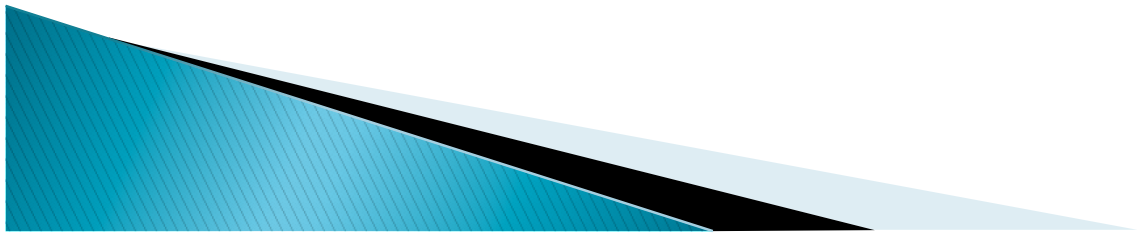


IG=0 As a Base Case

- Consider the following

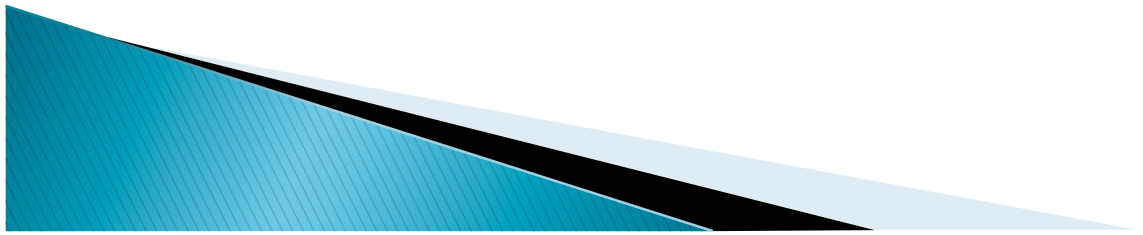
Y	X_1	X_2
0	0	0
1	0	1
1	1	0
0	1	1

- Both features give 0 IG
- Once we divide the data, perfect classification!



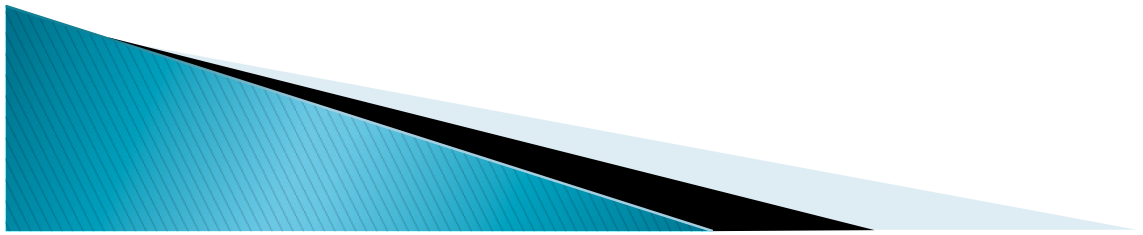
Bias/Variance Tradeoff

- Complete trees have no bias
  - But can over-fit badly
  - Lots of variance
- 0 depth trees (return most likely label) have no variance
  - Totally biased towards majority label
- A good tree balances between these two
  - How do we learn balanced trees?



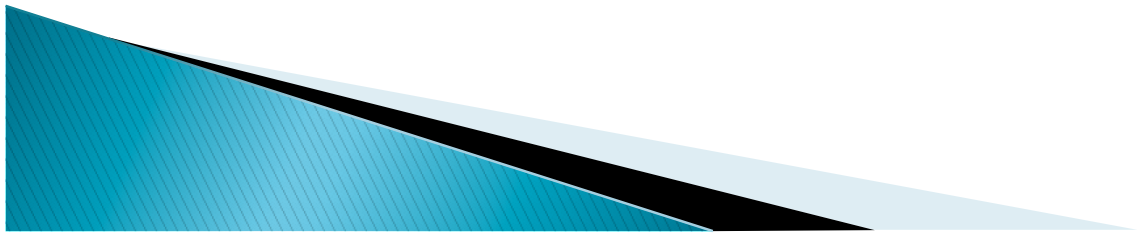
Pruning: New Base Cases

- Stop when too few examples
- Stop when max depth reached
- Stop when my classification error is not much more than average of my children
- X^2 pruning- stop when remainder is no more likely than chance



Extensions

- Non-binary attributes
  - Categorical
  - Continuous (real valued)
    - Handle by thresholding- find the best single threshold to split the range
    - Regression trees
- Missing attributes
- Alternatives to information gain: Gini index, miss-classification rate
- Non-greedy algorithms?



Next Time….

- Reading: PRML 3-3.1.4, 3.2, 1.2.5