Decision Trees

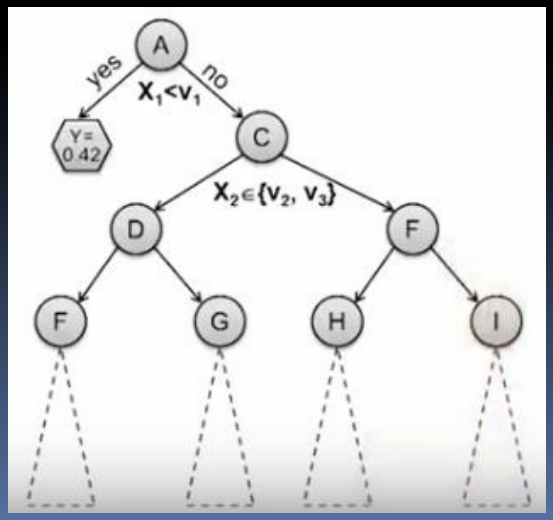
CSCI 447/547 MACHINE LEARNING

Outline

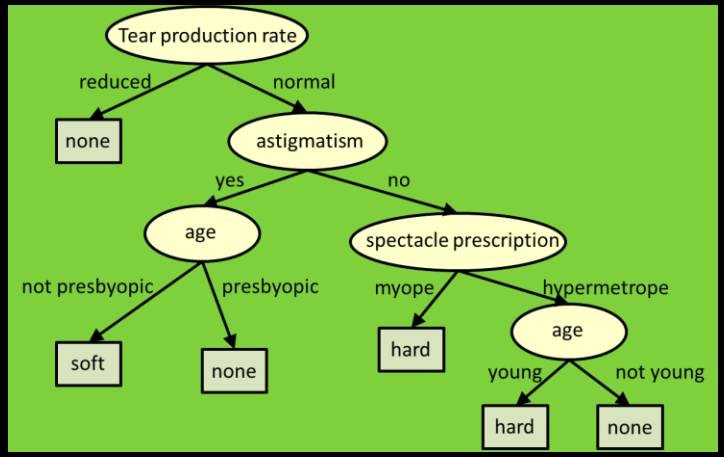
- Overview
- Decision Boundaries
- Construction of a Tree
- Information Gain

Overview

- Goal is to predict the value of the dependent variable based on values of other variables
  - Sound familiar?
- We still have d attributes, each with their own domain (can be categorical or numeric)
- Output variable also has a domain
- We have n (labeled) examples from which to learn



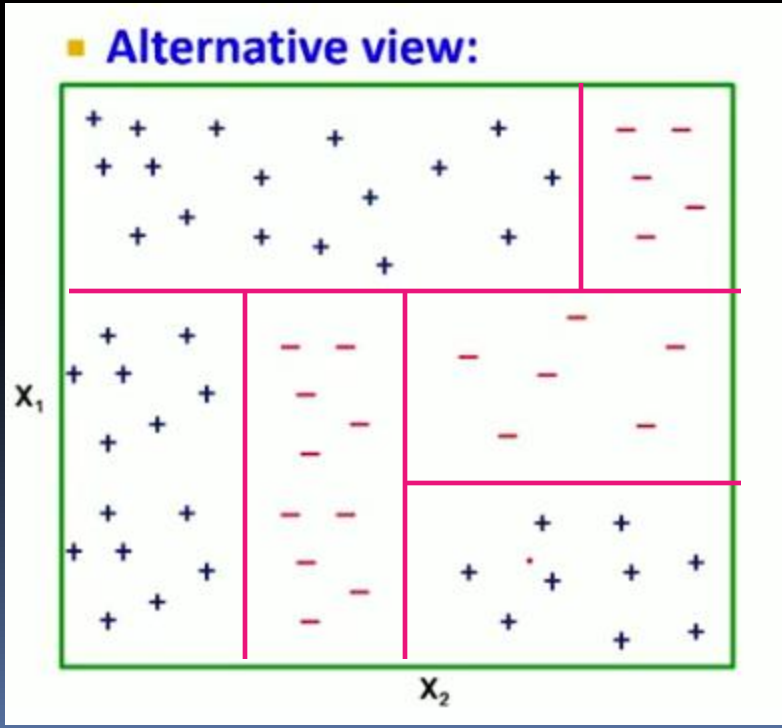
Overview



- Making a prediction is easy
  - Take an example,
    - Test the first variable in the tree against the appropriate attribute
      - Follow the direction the value indicates
    - Repeat for each variable in the tree encountered until you reach a leaf node
  - Use the leaf value (or some function of the leaf values) as the prediction or classification

Decision Boundaries

- Each choice in the tree separates the space into two (with a binary tree)



Construction of a Tree

- Decide on what variable to use as decision nodes recursively from top to bottom
  - Evaluate which attribute is "best"
  - Evaluate what value to split on
- Decide on when to stop
  - Not necessarily when everything has been categorized perfectly…

Construction of a Tree

- Finding the best split – categorical data
  - Finding the attribute and value that optimizes some criterion
  - Use Information Gain
    - Measures how much an attribute tells us about a class Y
    - IG(Y|X)
      - Transmit Y over a binary link. How many bits, on average, would it save us if both ends of the line knew X?

Construction of a Tree

- Finding the best split – numeric data
  - Find split that maximizes:
    $|D|*Var(D) – (|D_L|*Var(D_L) + |D_R|*Var(D_R))$
    where D is the data in the parent, and D_L and D_R is the data in the left and right child and Var is the variance
  - Do this by sorting on value and considering splits between each adjacent values
    - For categorical attributes, find best split based on subsets

Construction of a Tree

- When to Stop?
  - When the leaf is "pure"
    - It contains only one type of classification, or if numeric, if the variance is small
  - When the number of examples in the leaf is too small
- In practice, probably use both measures

Construction of a Tree

- How to Make a Prediction?
  - Regression
    - Predict the average value of the examples in the leaf
    - Build a linear regression model on the leaf examples
  - Classification
    - Predict the majority class of the examples in the leaf

Information Gain

- Entropy
  - The smallest possible number of bits, on average, per symbol, needed to transmit a stream of symbols drawn from a variable's distribution
  - $H(X) = -\sum_{j=1}^{m} p_j log_2 p_j$
  - High entropy means X is from a uniform distribution
  - Low entropy means X has a varied distribution

Information Gain

- An example
  - From this data we can estimate:
    - P(Y=Yes) = 0.5
    - P(X=Math & Y=No) = 0.25
    - P(X=Math) = 0.5
    - P(Y=Yes|X=History) = 0

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
Math	Yes
History	No

    - $H(Y) = -1/2*\log_2(1/2) + 1/2*\log_2(1/2) = 1$
    - H(X) = 1.5

Information Gain

- Specific Conditional Entropy
  - H(Y | X=v) = The entropy of Y among only those records in which X has value v
  - H(Y|X=Math) = 1
  - H(Y|X=History) = 0
  - H(Y|X=CS) = 0

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
Math	Yes
History	No

Information Gain

- Conditional Entropy
  - H(Y | X) = The average specific conditional entropy of Y
    - If you choose a record at random, what will be the conditional entropy of Y, conditioned on that row's value of X
    - Expected number of bits to transmit Y, if both sides will know the value of X

$$\sum_j P(X = v) H(Y|X = v)$$

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
Math	Yes
History	No

Information Gain

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
Math	Yes
History	No

V_j	$P(X=V_j)$	$H(Y\|X=v_j)$
Math	0.5	1
History	0.25	0
CS	0.25	0

So $H(Y|X) = 0.5*1 + 0.25*0 + 0.25*0 = 0.5$

Information Gain

- Information Gain
  - How many bits would we save on average if both ends knew X?
  - The bigger the difference, the more information we gain from X
  - $IG(Y|X) = H(Y) - H(Y|X)$
  - Compute this for each attribute and choose the one with the highest information gain to split on

Information Gain

- Suppose you are trying to predict whether someone is going to live past 80 years
- From historical data you might find:
  - IG(LongLife | HairColor) = 0.01
  - IG(LongLife | Smoker) = 0.2
  - IG(LongLife | Gender) = 0.25
  - IG(LongLife | LastDigitOfSSN) = 0.00001
- IG tells us how much information about Y is contained in X
  - So attribute X that has high IG(Y|X) is a good split

Tying Up Decision Trees

- Decision trees are a very popular tool
  - Easy to understand
  - Easy to implement
  - Easy to use
  - Computationally cheap
  - Can do classification as well as regression
  - BUT, can easily overfit the data

Tying Up Decision Trees

- Learn multiple trees and combine their predictions
  - Gives better performance in practice
- Bagging
  - Learn multiple trees over independent samples of the training data
  - Predictions from each tree are averaged to compute final prediction

Summary

- Overview
- Decision Boundaries
- Construction of a Tree
- Information Gain