Classification

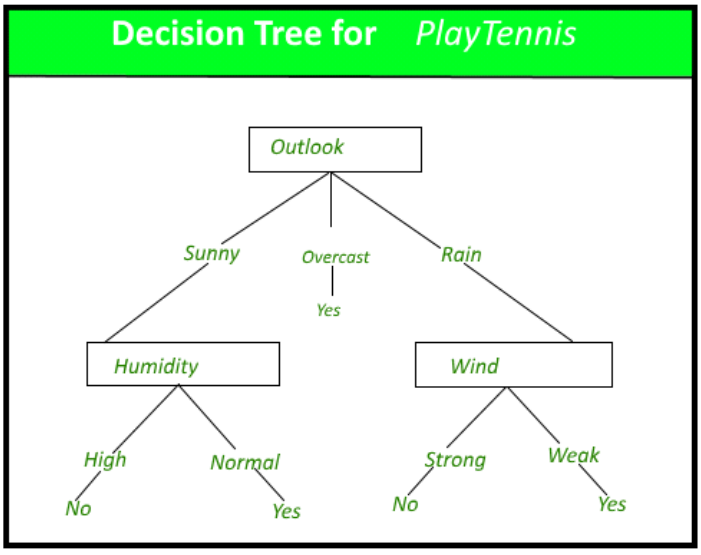
Tuesday, October 12, 2021 4:32 PM

Given a training set where data is labeled with a special attribute called a class (discrete value)
We want to find a model for the class attribute as a function of the values of the other attributes
    Seems similar to arrhythmia dataset with class in **y**

Classification Techniques

- Instance based classifiers
  - Keep a record of attributes which we can use to predict the class label of unseen classes
  - If we have an old record that matches the new record, we can label it the same
  - Rote-learners:
    - Perform classification only if the attributes of the unseen record exactly match a record in our training set
  - Nearest Neighbor:
    - Use the **k** closest records to perform classification
    - Requires a large training set, distance function, and a value for **k**
    - Classifying an unseen record:
      1. Compute distance of unseen record to all training records
      2. Identify the k nearest neighbors
      3. Aggregate the labels of these k neighbors to predict the label of the unseen record
    - Aggregation methods:
      - Majority rule
      - Weighted majority based on distance (w=1/d^2)
    - Scaling issues:
      - Attributes should be scaled to prevent distance measures from being dominated by one attribute
      - Example
        - Height: 1m ---> 2m
        - Income: 10k ---> 1million
    - Choosing the value of k:
      - If k is too small -> sensitive to noise points + overfitting (doesn't generalize well)
      - If k is too big -> neighborhood may include points from other classes
    - Pros:
      - Sime to understand why a given unseen record was given a particular class
      - Adapts to new attributes
    - Cons:
      - Expensive to classify new points
      - KNN can be problematic in high dimensions

- Decision trees
  - Classify records by traversing the decision tree until we reach a class attribute



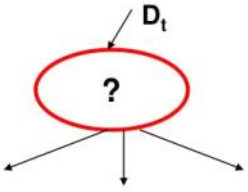
Algorithms:

Hunt's Algorithm

General Structure of Hunt's Algorithm

- Let D_t be the set of training records that reach a node t
- General Procedure:
  - If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t
  - If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset.

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



17

If D_t is an empty set, then t is a leaf node labeled by the default class, y_d

Attribute types:

- Nominal
- Ordinal
- Continuous

Nominal Attribute Splitting:

- Multiway split: Use as many partitions as distinct values
- Binary split: Divide values into two subsets, requires optimal partitioning

Continuous Attribute Splitting:
- Discretization: form an ordinal categorical attribute
  - Static - discretize once at the beginning
  - Dynamic - ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering
- Binary Decision: (A < v) or (A >= v)
  - Find best possible value for an optimal cut

Determining the best split:
- Greedy Approach:
  - Nodes with homogeneous class distribution are preferred
  - Need a measure of node impurity

CO: 5	C1: 5

Non-homogeneous, high impurity

C0: 9	C1: 1

Homegeneous, low impurity

  - Compare loss of impurity between two different splits
    - Methods of measuring node impurity:
      - Gini Index
      - Entropy

Impurity Criterion

Gini Index

$$I_G = 1 - \sum_{j=1}^{c} p_j^2$$

p_j: proportion of the samples that belongs to class c for a particular node

Entropy

$$I_H = -\sum_{j=1}^{c} p_j log_2(p_j)$$

p_j: proportion of the samples that belongs to class c for a particular node.

*This is the the definition of entropy for all non-empty classes ($p \neq 0$). The entropy is 0 if all samples at a node belong to the same class.

Misclassification
- Similar to Gini but with linear growth/decay

CART
ID3, C4.5
SLIQ, SPRINT

Stopping Criteria for tree induction:
- Stop expanding a node when all the records belong to the same class
- Stop expanding when all the records have similar attribute values
- Early termination

Creating too large of a tree will result in overfitting and higher error percentage
- Might happen because of noise points or insufficient examples

Naïve Bayes

Support Vector Machines
Neural Networks