Classification Techniques (1)

Overview

Today

- Classification Problem
- Classification based on Regression
- Distance-based Classification (KNN)

Next Lecture

- Decision Trees
- Classification using Rules
- Quality of Classifiers

Classification Problem

- Given a database $D = \{t_1, t_2, \ldots, t_n\}$ and a finite set of classes $C = \{C_1, \ldots, C_m\}$, the ***Classification Problem*** is to define a mapping $f: D \rightarrow C$ where each t_i is assigned to one class.
- Actually, f divides D into *equivalence classes*.
- ***Prediction*** is a similar process, but may be viewed as having an infinite number of classes.

Classification: Definition

- Given a collection of records (training set)
  - Each record contains a set of attributes, one of the attributes is the class.
- Find a model for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
  - A test set is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

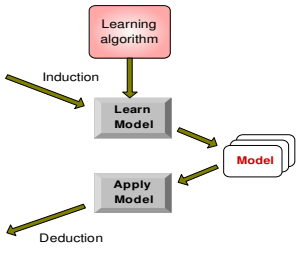
Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

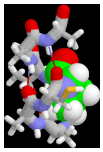
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Examples of Classification Tasks

- Predicting tumor cells as benign or malignant
- Classifying credit card transactions as legitimate or fraudulent
- Classifying secondary structures of protein as alpha-helix, beta-sheet, or random coil
- Categorizing news stories as finance, weather, entertainment, sports, etc

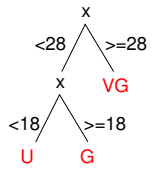


More Classification Examples

- Classify students' grades as VG, G, or U.
- Identify mushrooms as poisonous or edible.
- Classify stocks as buy, keep, or sell.
- Identify individuals with credit risks.
- Perform speech recognition.
- Perform pattern recognition.

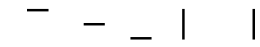
Classification Example: Grading Exams

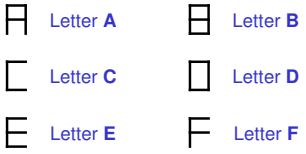
- If x >= 28 then grade = VG
- If 18 <= x < 28 then grade = G
- If x < 18 then grade = U



Classification Example: Letter Recognition

View letters as constructed from 5 components:





Classification Techniques

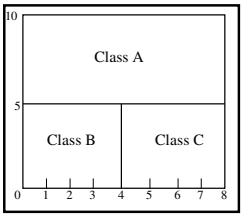
Approach:

- Create specific model by evaluating training data (or by using knowledge from domain experts).
- Apply model developed to new data.

Classes must be predefined.

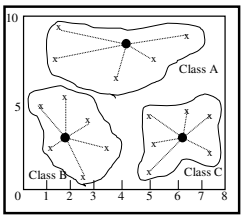
Most common techniques use decision trees (DTs), neural networks (NNs), or are based on distances or statistical methods.

Defining Classes



Partitioning Based

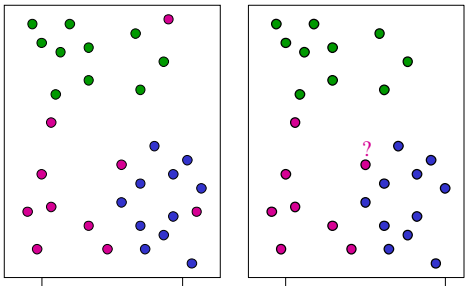
Distance Based



Issues in Classification

- Missing Data
  - Ignore
  - Replace with assumed value
- Handling of Outliers in Training Data
- Measuring Performance
  - Classification accuracy on test data
  - Confusion matrix

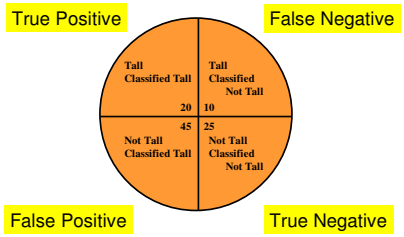
Handling of Outliers in Training Data



Height Example Data

Name	Gender	Height	Output1	Output2
Kristina	F	1.60	Short	Medium
Jim	M	2.02	Tall	Medium
Maggie	F	1.90	Medium	Tall
Martha	F	1.88	Medium	Tall
Stephanie	F	1.71	Short	Medium
Bob	M	1.85	Medium	Medium
Kathy	F	1.60	Short	Medium
Dave	M	1.72	Short	Medium
Worth	M	2.12	Tall	Tall
Steven	M	2.10	Tall	Tall
Debbie	F	1.78	Medium	Medium
Todd	M	1.95	Medium	Medium
Kim	F	1.89	Medium	Tall
Amy	F	1.81	Medium	Medium
Wynette	F	1.75	Medium	Medium

Classification Performance



Confusion Matrix Example

Using height data example with Output1 correct and Output2 actual assignment

Actual Membership	Assignment Short	Medium	Tall
Short	0	4	0
Medium	0	5	3
Tall	0	1	2

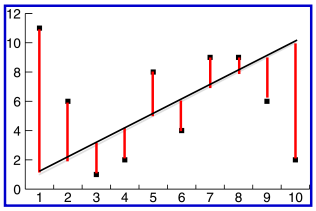
Regression

- Assume data fits a predefined function
- Determine best values for ***regression coefficients*** $c_0,c_1,\ldots,c_n$.
- Assume an error: $y = c_0+c_1x_1+\ldots+c_nx_n+\varepsilon$
- Estimate error using mean squared error for training set:

$$y_i = c_0 + c_1 x_{1i} + \epsilon_i, i = 1, \ldots, k$$

$$L = \sum_{i=1}^{k} \epsilon_i^2 = \sum_{i=1}^{k} (y_i - c_0 - c_1 x_{1i})^2$$

Linear Regression Poor Fit

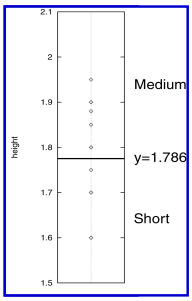


Classification Using Regression

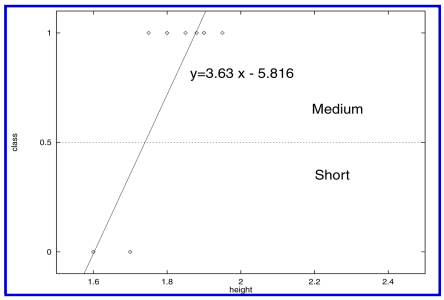
Division: Use regression function to divide area into regions.

Prediction: Use regression function to predict a class membership function. Input includes desired class.

Division



Prediction



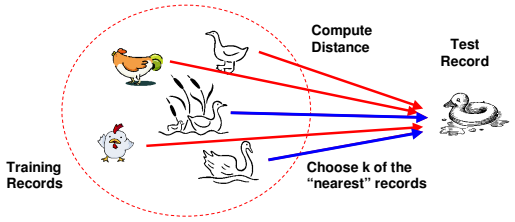
Instance Based Classifiers

Examples:

- Rote-learner
  - Memorizes entire training data and performs classification only if attributes of record match one of the training examples exactly
- Nearest neighbor
  - Uses k "closest" points (nearest neighbors) for performing classification

Nearest Neighbor Classifiers

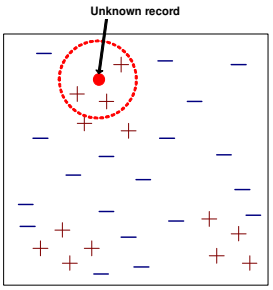
- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



Classification Using Distance Measures

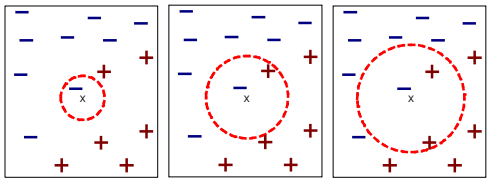
- Place items in the class to which they are "closest".
- Must determine distance between an item and a class.
- Classes represented by
  - **Centroid**: Central value.
  - **Medoid**: Representative point.
  - A set of individual points
- Algorithm: **K-Nearest Neighbors (KNN)**

Nearest-Neighbor Classifiers



- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve
- To classify an unknown record:
  - Compute distance to other training records
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Definition of Nearest Neighbor

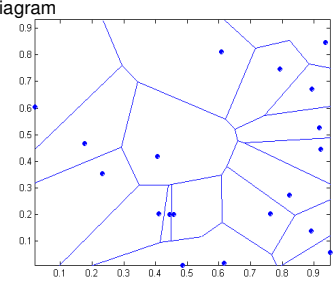


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

1 Nearest Neighbor

Voronoi Diagram



Nearest Neighbor Classification

- Compute distance between two points:
  - Euclidean distance

$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

- Determine the class from nearest neighbor list
  - take the majority vote of class labels among the k-nearest neighbors
  - Weigh the vote according to distance
    - weight factor, $w = 1/d^2$

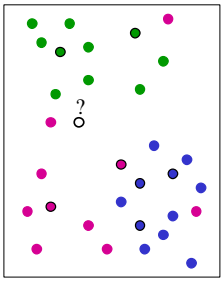
K Nearest Neighbors (KNN):

- Training set includes classes.
- Examine K items near item to be classified.
- New item placed in class with the most number of close items.
- O(q) for each tuple to be classified. (Here q is the size of the training set.)

KNN Algorithm

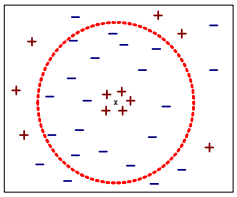
```
Input
  D  // training data
  N  // neighbors
  t  // tuple to classify
Output
  c  // class to which t gets classified
KNN Algorithm
  N = ∅ ;
  for each d ∈ D do
    if |N| < K then N = N ∪ {d} ;
    else
      u = the item in N with max distance (dissimilarity) from t ;
      if sim(t,u) < sim(t,d) then N = (N - {u}) ∪ {d} ;
  c = the class to which most n ∈ N are classified ;
```

KNN Algorithm



Nearest Neighbor Classification: Issues

- Choosing the value of k:
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other classes



Nearest Neighbor Classification: More issues

- Scaling issues
  - Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
  - Example:
    - height of a person may vary from 1.5m to 1.8m
    - weight of a person may vary from 90lb to 300lb
    - income of a person may vary from $10K to $1M

Nearest Neighbor Classification: Wrap-up

- k-NN classifiers are lazy learners
  - They do not build models explicitly
  - Unlike eager learners such as decision tree induction and rule-based systems
  - Classifying unknown records is relatively expensive