CS5805 : Machine Learning I
Lecture #12

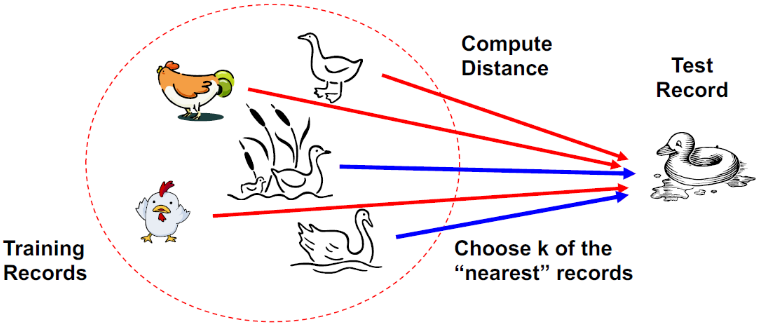
Reza Jafari, Ph.D

Collegiate Associate Professor
rjafari@vt.edu

COLLEGE OF ENGINEERING
COMPUTER SCIENCE
VIRGINIA TECH.

K-nearest Neighbors

- If it walks like duck, quacks like a duck, then it is probably a duck.
- KNN is typical example of a lazy learner.
- It is called lazy not because of its apparent simplicity, but because it does not learn a discriminative function from the training data but memorizes the dataset instead.



K-nearest Neighbors Classifiers

- Requires three things:
    - The set of labeled.
    - Distance Metric to compute distance between records.
    - The value k, the number of nearest neighbors to retrieve.

K-nearest Neighbors Classifiers

- Requires three things:
  - The set of labeled.
  - Distance Metric to compute distance between records.
  - The value k, the number of nearest neighbors to retrieve.

K-nearest Neighbors Classifiers

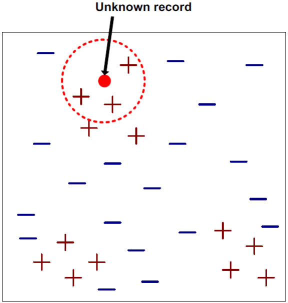
- Requires three things:
  - The set of labeled.
  - Distance Metric to compute distance between records.
  - The value k, the number of nearest neighbors to retrieve.

K-nearest Neighbors Classifiers

- Requires three things:
  - The set of labeled.
  - Distance Metric to compute distance between records.
  - The value k, the number of nearest neighbors to retrieve.

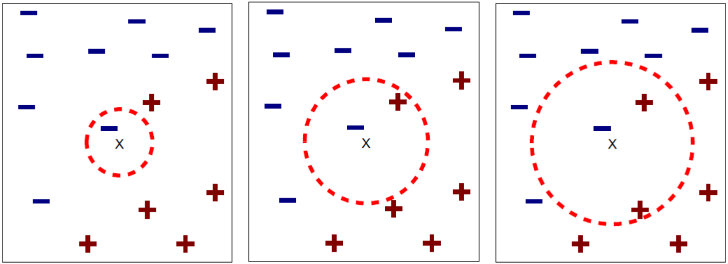
K-nearest Neighbors Classifiers

- Requires three things:
  - The set of labeled.
  - Distance Metric to compute distance between records.
  - The value k, the number of nearest neighbors to retrieve.
- To classify an unknown record:
  - Compute distance to other training records.
  - Identify k nearest neighbors.
  - Classifies e.g. by taking majority vote



Definition of Nearest Neighbor

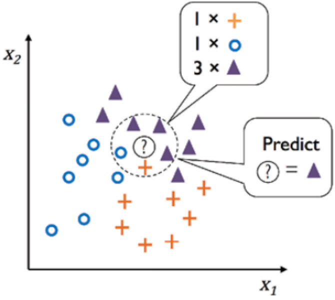
- K-nearest neighbors of a record x are data points that have the smallest distance to x.



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

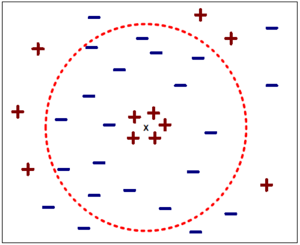
k-Nearest Neighbor Algorithm

- The kNN algorithm itself is straightforward and can be summerized by the following steps:
  1. Choose the number k and a distance metric.
  2. Find the k-nearest neighbors of the data record that we want to classify.
  3. Assign the class label by majority vote.



k-Nearest Neighbor Algorithm

- If k is too small, then the nearest neighbor classifier may be susceptible to overfitting due to noise, i.e., mislabeled examples in the training data.
- If k is too large, then the nearest neighbor classifier may misclassify the test instance because its list of nearest neighbors includes training examples that are located far away from its neighborhood.



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, weigh factor $w = \frac{1}{d^2}$
- But how does the weighted k-NN Algorithm works?

How does the weighted k-NN Algorithm works

- Compute all distances from the item-to-classify all the labeled data.
- For example, if a labeled data item is $(0.4, 0.7)$ and the item to classify is $(0.55, 0.6)$ then the Euclidean distance

$$\begin{aligned} dist =& \sqrt{(0.4 - 0.55)^2 + (0.7 - 0.6)^2} \\ =& 0.1803 \end{aligned}$$

- Then use the voting algorithm to determine the predicted class, i.e. inverse weights technique.
- Find the inverse of distances and the associated labeled classes.
- Find the sum of the inverses, then divide each inverse by the sum.
- Find the sum per each class. The predicted class is the one with the greatest vote.

Example of inverse weights technique

- Let suppose the six distances and the associated labels are :

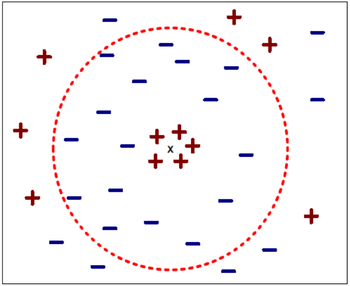
Tabel 1: **Distances and Labels**

	1	2	3	4	5	6
dist	0.070	0.076	0.092	0.111	0.133	0.158
class	0	0	1	2	2	2
dist inverse	14.14	13.12	10.84	8.94	7.49	6.32
$\sum$ dist inverse	60.8796					
$\frac{dist}{sum}$	0.232	0.215	0.178	0.146	0.123	0.103

- $\boxed{\text{Class 0}}$: $0.2323 + 0.2157 = 0.448$
- Class 1 : 0.1782
- Class 2 : $0.1469 + 0.1039 + 0.1231 = 0.3739$

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

- Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
- Examples:
  - 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...

- k-NN classifiers are lazy learners since they do not build models explicitly.
- Classifying unknown records is relatively expensive.
- Can produce arbitrarily shaped decision boundaries
- Easy to handle variable interactions since the decisions are based on local information.
- Selection of right proximity measure is essential.
- Superfluous or redundant attributes can create problems.
- Missing attributes are hard to handle.

Parametric versus non-parametric models

- Machine learning algorithms can be grouped into parametric and non-parametric models:

Parametric

1. Estimate parameters from training dataset to learn a function that can classify new data points without requiring the original training dataset.
2. Examples: Perceptron, logistic regression, linear SVM.

Non-parametric

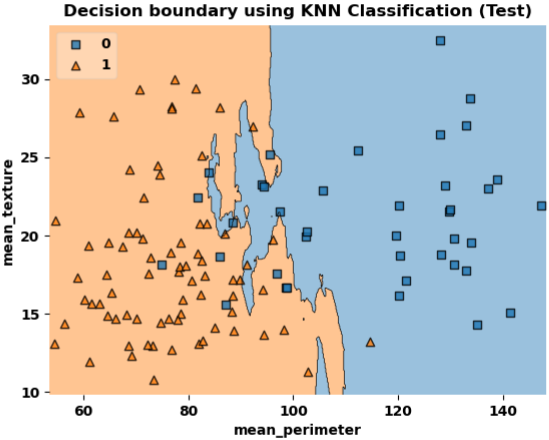
1. Can not be characterised by fixed set of parameters..
2. Number of parameters grows with the training data.
3. Examples: decision tree, random forest and kernal SVM.

- kNN belongs to subcategory of non-parametric models, described as instance-based learning.

kNN Python Implementation- Breast cancer

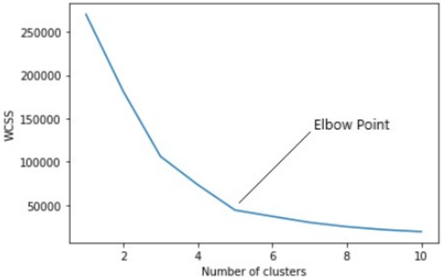
- Importing necessary python libraries
- Importing the dataset Breast cancer dataset from scikit-learn. The dataset consists of data related to breast cancer patients and their diagnosis **malignant** or **benign**.
- Separating the features and target variables
- Splitting the dataset into training and test sets
- Fitting the model to the training set. Using KNeighboursClassifier() class from scikit-learn.
- Predicting the test results
- Evaluating the model.
- Plotting the decision boundary
- **Develop the python code for above procedures**

- The decision boundary of the model on test data for the breast cancer dataset



K Means Clustering Using the Elbow Method

- In the Elbow method, we are actually varying the number of clusters (K) from 1-10.
- For each value of K, we are calculating WCSS (Within-Cluster Sum of Square).
- WCSS is the sum of the squared distance between each point and the centroid in a cluster.
- When we plot the WCSS with the K value, the plot looks like an Elbow.



K Means Clustering Using the Elbow Method

- WCSS value is largest when K = 1.
- From the elbow point, the graph moves almost parallel to the X-axis.
- The K value corresponding to this point is the optimal value of K or an optimal number of clusters.