K-Nearest Neighbors (KNN)

Overview

- Core idea of KNN
- Distance Scores
- KNN for Classification and Prediction
- Advantages and Disadvantages of KNN
- Summary

SUPERVISED LEARNING

Categorical Outcome

Classification

Pass / Fail
Purchase / No purchase,
Fraudulent / Genuine
Creditworthy / Defaulter
Owner / Non-owner

Prediction

(referred to as regression also)

Numerical Outcome

Total Marks scored
Total Amount spent on purchase
Amount of Fraud
Amount of Default
Net worth

Characteristics of KNN Algorithm

- KNN can used for classification(categorical outcome) as well as prediction (numerical outcome).

- Data Driven
  - Little or no prior knowledge about the distribution of the data, i.e. there are no assumptions about the data such as normality

- Non-parametric
  - Unlike parametric models which have fixed number of parameters, in a non-parametric model, the complexity of the model grows with the number of training data

- Instance-based learning/Case-based learning/Memory-based learning/Lazy learning
  - instead of performing explicit generalization, new instances are compared with instances stored in the memory during training, wherein the computation is delayed until classification

Core Idea of KNN

- For a new datapoint to be classified, identify the nearby records
- “Nearby” means the labeled datapoints with similar predictor values $x_1, x_2, \ldots x_p$
- Classify the datapoint as belonging to the majority class among the nearby datapoints (the “neighbors”)

Distance Scores

(x_1, y_1) (x_2, y_2) $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

- Euclidean Distance

$$ED(x,y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$$

- Mahalanobis Distance

$$d(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})^T S^{-1} (\vec{x}-\vec{y})}.$$

- Manhattan Distance

$$MD(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

- Chebychev Distance

$$CD(x,y) = \max_i |x_i - y_i|$$

- Minkowski

$$D_{Mink}(x,y) = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p},$$

- Hamming

$$HamD(x,y) = \sum_{i=1}^{n} 1_{x_i \neq y_i}$$

- …

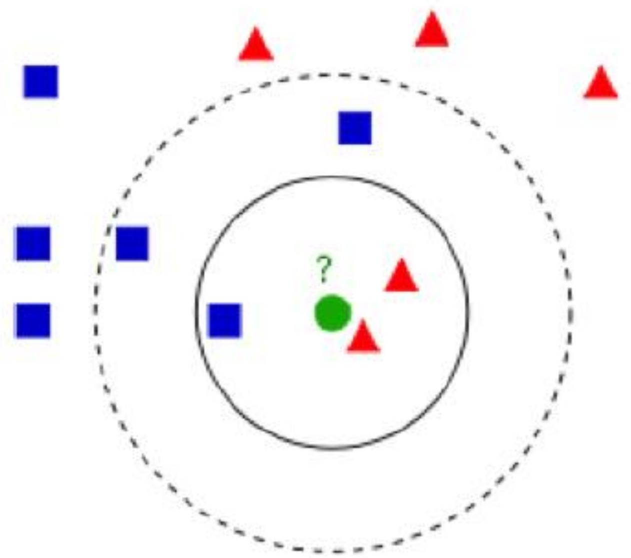
Ref: V. B. S. Prasath et al, Effects of Distance Measure Choice on KNN Classifier Performance - A Review, https://arxiv.org/pdf/1708.04321.pdf

Basic KNN Algorithm - Classification

Input : Training Set D, Test datapoint d, Parameter k

Output : Class label of Test datapoint d

1. Compute the distance between test datapoint d and every datapoint in the training set D
2. Choose the k datapoints in training set D that are nearest to test sample d; denote this set by P (∈ D)
3. Assign the test datapoint d to the most frequent class (majority class)



Ref: V. B. S. Prasath et al, Effects of Distance Measure Choice on KNN Classifier Performance - A Review, https://arxiv.org/pdf/1708.04321.pdf

How to choose k

- *k* is the number of nearby neighbors to be used to classify the new record
  - *K*=1 means use the single nearest record
  - *K*=3 means use the 3 nearest records

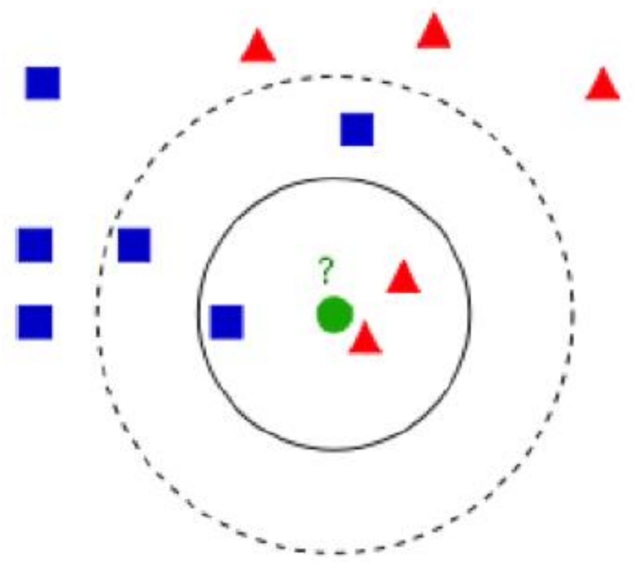
- Low values of *k* (1, 3, …) capture local structure in data (but also noise)

- High values of *k* provide more smoothing, less noise, but may miss local structure
  - extreme case of k = n (i.e., the entire data set) is the same as the “naïve rule” (classify all records according to majority class. Here the relevance of predictors vanish and we can avoid the computation of distance scores altogether. Enough to just look at the proportion of outcome labels and choose the majority)

- Odd values of *k* preferred, to avoid tie caused by even values of k during majority voting

KNN for Prediction/Regression (for Numerical Outcome)

- Use average of outcome values, instead of “majority vote”
- Weighted average, weight decreasing with distance, etc



Advantages of KNN

- Simple approach and robust to noise in training data
- Learns complex structures in data easily (relatively) without having to define any statistical model
- Little or no prior knowledge about the distribution of the data, i.e. there are no assumptions about the data
- Can be used for ‘imputation’ of missing data

Disadvantages of KNN

- Lazy learning where computation is delayed until classification and hence need to compute distance scores each time
- Time consuming when dealing with large datasets
- Computational costs when applied to high dimensional data (‘curse-of-dimensionality’)

Summary

- KNN is a data driven, non-parametric approach
- Simple but powerful technique that selects k-nearest neighbors based on distance scores
- Can be used for both Classification and Prediction(Regression)
- Performance impacted by high dimensional data as well as volume of data.

Readings

- https://www.analyticsvidhya.com/blog/2018/08/k-nearest-neighbor-introduction-regression-python/
- https://machinelearningmastery.com/k-nearest-neighbors-for-machine-learning/
- https://arxiv.org/pdf/1708.04321.pdf