Supervised Learning II
K-Nearest Neighbors and Naïve Bayes

Jeff Tang

Outline

- Introduction
- How K-Nearest Neighbors works
- Choice of size of K
- Worked example of KNN
- Strength and weakness of KNN
- How Naïve Bayes works
- Worked example of Naïve Bayes
- Strength and weakness of Naïve Bayes
- Practical applications

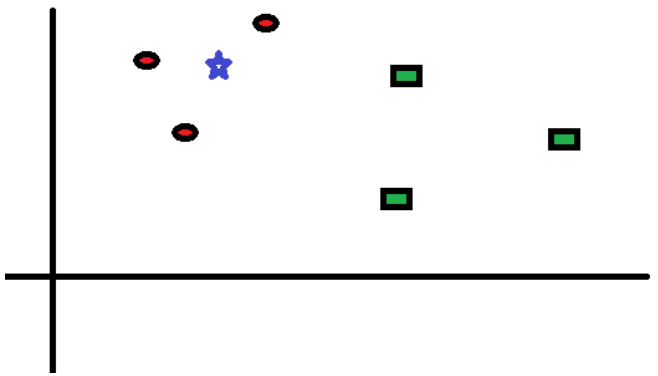
Introduction

- Both **K-Nearest Neighbors** and **Naïve Bayes** are non-parametric and non-linear supervised learning algorithms.
- **K-Nearest Neighbors (KNN):**
  - It is one of the simplest ML algorithms and is an example of *instance-based* learning, where new data are classified based on stored and labeled instances.
  - There is no training step because there is no model to build. The model is just the whole dataset.
  - KNN can be used for both classification and regression.
- **Naïve Bayes:**
  - *Naive Bayes classifiers* are a family of simple probabilistic classifiers based on applying *Baynes' therorm* with strong independence assumptions (i.e. being naive) between the features.

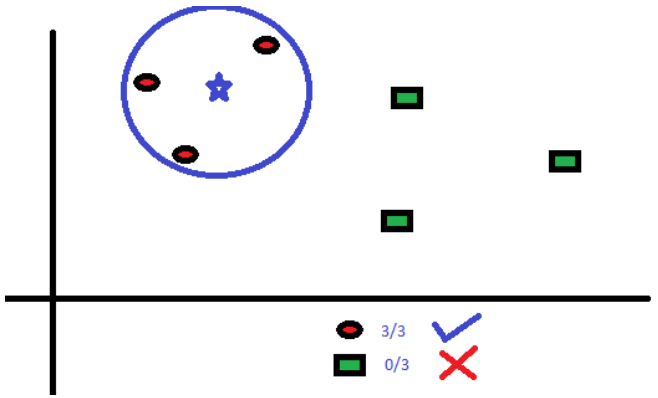
How K-Nearest Neighbors works

Let’s take a simple case to understand this algorithm:

- The figure shows a spread of red circles (RC) and green squares (GS). You intend to find out the unknown class of the blue star (BS), which can be either RC or GS.



- The “K” in KNN algorithm is the number of the nearest neighbors we wish to take vote from. Let’s say K = 3.
- Hence, we will now make a circle with BS as centre just as big as to enclose only three datapoints on the plane. The metric often used for distance measurement is the *Euclidian distance*.



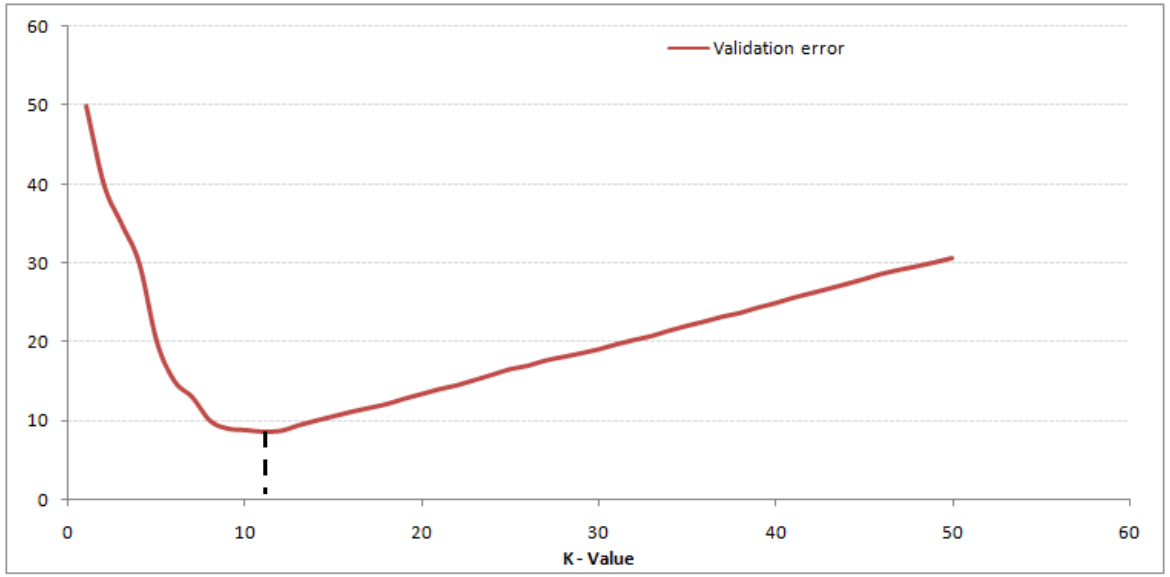
- The three closest points to BS are all RC. By voting with a result 3/3 votes (i.e. 100%) for RC, we can say that the BS should belong to the class RC.

Choice of size of K

- To get the optimal value of K, you can segregate the training and validation sets from the initial dataset, e.g. in a ratio of 2 : 1.
- Use the classification inaccuracy as a measure of *validation error*:

  Validation error % = (No. of examples wrongly predicted/Total no. of examples) x 100%

- Now plot the validation error curve and locate its minimum to get the optimal value of K. This value of K should be used for all predictions.



Worked example of KNN

Problem:

- We have data from a survey on a special paper tissue and objective testing with its two attributes (acid durability and strength) to classify whether it is good or not. Below are 4 samples:

X1 = Acid durability (seconds)	X2 = Strength (kg/square metre)	Y = Classification
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

- Now the factory produces a new paper tissue that passes the laboratory test with X1 = 3 and X2 = 7. Without another survey at a high cost, take a guess of the classification of this new paper tissue.

Worked example (cont'd)

- Suppose K = 3.
- Now calculate the Euclidean distances between the query instance and all the sample instances. We can save time just by calculating the square distances instead.

X1 = Acid durability (seconds)	X2 = Strength (kg/square metre)	Square distance to query instance (3,7)
7	7	$(7-3)^2 + (7-7)^2 = 16$
7	4	$(7-3)^2 + (4-7)^2 = 25$
3	4	$(3-3)^2 + (4-7)^2 = 9$
1	4	$(1-3)^2 + (4-7)^2 = 13$

Worked example (cont'd)

- Then sort the distance and determine the 3 nearest neighbors based on the K-th minimum distance to check their classifications.

X1 = Acid durability (seconds)	X2 = Strength (kg/square metre)	Square distance to query-instance (3,7)	Rank by minimum Distance	Included in 3 nearest neighbors?	Y = Classification
7	7	$(7-3)^2 + (7-7)^2 = 16$	3	Yes	Bad
7	4	$(7-3)^2 + (4-7)^2 = 25$	4	No	/
3	4	$(3-3)^2 + (4-7)^2 = 9$	1	Yes	Good
1	4	$(1-3)^2 + (4-7)^2 = 13$	2	Yes	Good

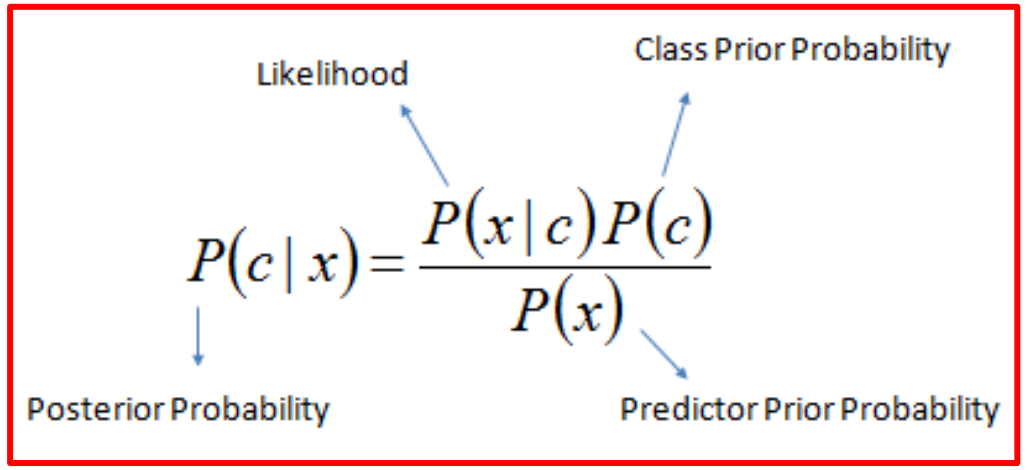
- Finally use simple majority of the classification of nearest neighbors to find the predicted class of the query instance. We have 2 good's and 1 bad. Therefore we can predict the new paper tissue to be in the 'good' class.
- KNN algorithm can also be used for regression other than classification problems. The only difference will be using averages of nearest neighbors rather than voting from nearest neighbors.

Strength and weakness of KNN

- **Strength**
  - Robust to noisy training data.
  - Effective and accurate if the training data size is large.
- **Weakness**
  - It is always difficult to determine which is the most optimal metric for measuring the distance and which attributes to use to produce the best results. Therefore, KNN is suitable for applications with which sufficient domain knowledge is available. This knowledge supports the selection of an appropriate measure.
  - The computation cost is high because we need to compute the distance of the query instance from all the given sample instances.

How Naïve Bayes works

- **Bayes theorem** provides a way of calculating the posterior probability, $P(c|x)$, from $P(c)$, $P(x)$, and $P(x|c)$. Naive Bayes classifier assume that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called *class conditional independence*.

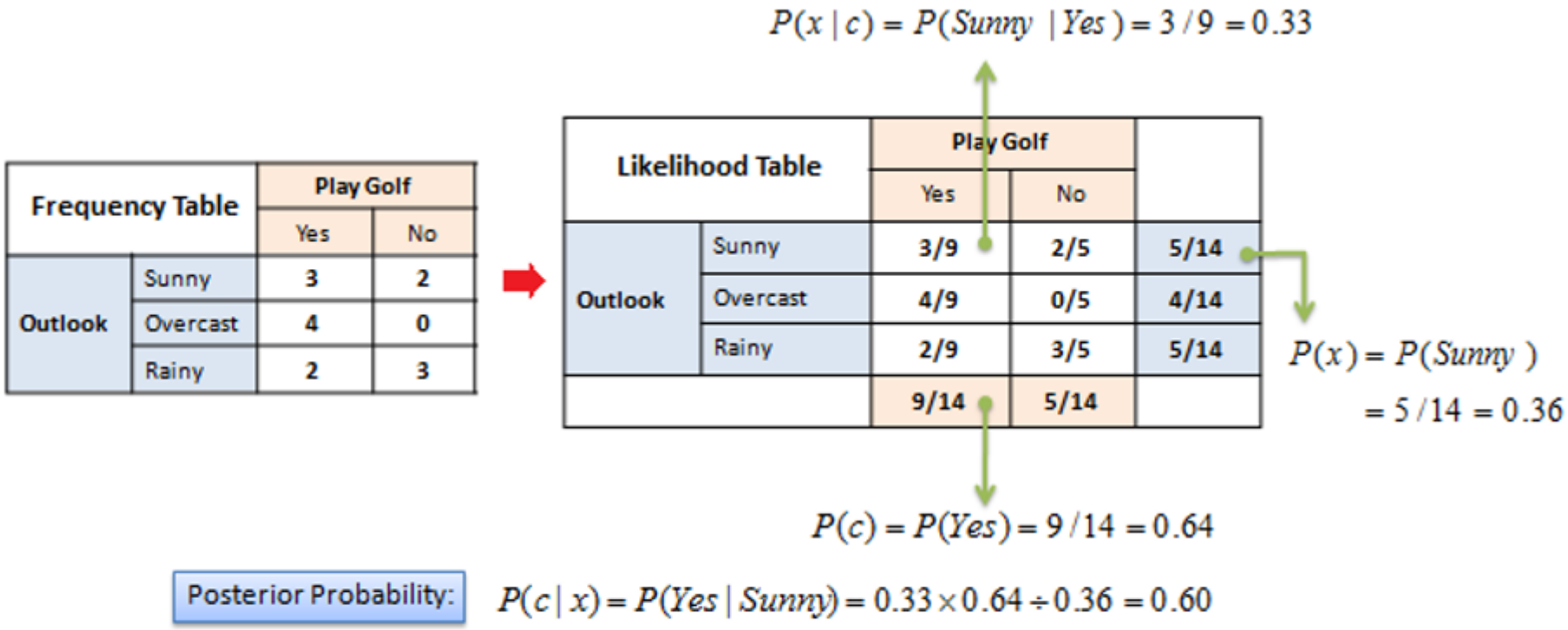


$P(c|x)$ is the posterior probability of *class* (*target*) given *predictor* (*attribute*)
$P(c)$ is the prior probability of *class*.
$P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
$P(x)$ is the prior probability of *predictor*.

- Steps to be taken:
  1. Construct a frequency table for each attribute against the target.
  2. Transform the frequency tables to likelihood tables.
  3. Use the *Naive Bayesian equation* to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Worked example

- Statement: “Players will play if outlook is sunny.” Is it correct?



- Since *P*(*Yes*|*Sunny*) is higher than 0.5 (as there are 2 classes only), the answer is ‘yes’. You can also check that *P*(*No*|*Sunny*) = 0.4 by applying the equation.

Strength and weakness of Naïve Bayes

- **Strength:**
  - It is easy and fast.
  - It performs well in multiclass predictions.
  - When assumption of independence holds, a Naive Bayes classifier performs better compared to other models like logistic regression and you need less training data.
  - It performs well in case of categorical input variables compared to numerical variable(s). For numerical variables, normal distribution is assumed.
- **Weakness:**
  - If the categorical variable has a category (in test data set) which was not observed in training data set, then the model will assign a zero probability and will be unable to make a prediction. This is often known as "*Zero Frequency*". To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called *Laplace estimation*.
  - Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

Practical applications

- **KNN:**
  - **Content retrieval:** Basically we can use it in Computer Vision for many cases, e.g. video search.
  - **Medical diagnosis:** The algorithm can be used to enhance the automated diagnoses, which include diagnosis of multiple diseases showing similar symptoms.
- **Naïve Bayes:**
  - **Real time prediction:** Naive Bayes is an eager learning classifier and it is fast. Thus, it could be used for making predictions in real time.
  - **Multiclass prediction:** This algorithm is also well known for multiclass prediction feature.
  - **Text classification/Spam filtering/Sentiment analysis:** Due to better results in multiclass problems and independence rule, Naive Bayes classifiers are often used in *text classification* and have higher success rate as compared to other algorithms. As a result, it is widely used in *spam filtering* (to identify spam e-mails) and *sentiment analysis* (in social media analysis, to identify positive and negative customer sentiments).
  - **Recommendation systems:** Naive Bayes Classifiers are used to build *recommendation systems* which use machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not.

URL references

- https://www.analyticsvidhya.com/blog/2014/10/introduction-k-neighbours-algorithm-clustering/
- http://people.revoledu.com/kardi/tutorial/KNN/HowTo_KNN.html
- https://www.youtube.com/watch?v=j0cjUkgfacI
- https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/
- http://www.saedsayad.com/naive_bayesian.htm
- https://machinelearningmastery.com/naive-bayes-tutorial-for-machine-learning/
- http://blog.aylien.com/naive-bayes-for-dummies-a-simple-explanation/
- https://www.youtube.com/watch?v=CPqOCI0ahss