K-nearest Neighbor Algorithm and Its Application

DR. WEIHUA LIU, CAMILA MURILLO, NATALIA ZAYTSEVA
What is K-Nearest Neighbor (KNN)

- One of the simplest machine language algorithms
- It stores all available cases and classifies new cases by a majority vote of its $k$ neighbors.
- It separates unlabeled data points into well defined groups.
When do we use KNN Algorithm

- Can be used for both classification and regression predictive problems, although it is most commonly applied to classification models.
- 3 important aspects to look at to evaluate any technique
  - The simplicity of interpreting the output
  - The Calculation time
  - The Predictive Power
Steps On How To Calculate KNN
(1) Determine k

- $k$ is the number of neighbors considered by the algorithm that the designer must pick in order to get the best possible fit for the data set.

- A small value for $k$ provides the most flexible fit, which will have low bias but high variance.

- A larger value for $k$ averages more voters in each prediction and hence is more resilient to outliers.
There are many distance functions but Euclidean is the most commonly used measure.

- Euclidean distance formula:

\[ d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \]

- The distance calculates the rank in terms of distance.
  - The smallest distance value will be ranked 1 and considered as the current nearest neighbor.
(3) Find Smallest Distance Values and Take The Average

- Find k smallest distance values
- Take k smallest and average the predictions
  - Add k smallest and divide by k
    - If k=1 then only closest value is considered
    - If k is infinity then entire dataset is considered.
Example: Suppose we have height, weight and T-shirt size of some customers and we need to predict the T-shirt size of a new customer given only the height and weight information.

New customer named 'Monica' has height 161cm and weight 61kg.

Let k=5
library(data.table)
mydat <- fread('http://archive.ics.uci.edu/ml/machine-learning-
databases/iris/bezdekIris.data')
head(mydat)
View(mydat)
data_norm <- function(x) {{(x - min(x))/(max(x) - min(x))}}
iris_norm <- as.data.frame(lapply(mydat[,-5], data_norm))
View(iris_norm)
summary(mydat[,1:4])
summary(iris_norm)
iris_train <- iris_norm[1:100,]
iris_test <- iris_norm[101:150,]
library(class)
iris_pred <- knn(iris_train, iris_test, mydat[1:100,5], k=12)
table(iris_pred, mydat[101:150,1])

<table>
<thead>
<tr>
<th>Iris_pred</th>
<th>setosa</th>
<th>versi</th>
</tr>
</thead>
<tbody>
<tr>
<td>setosa</td>
<td>24</td>
<td>3</td>
</tr>
<tr>
<td>versi</td>
<td>0</td>
<td>23</td>
</tr>
</tbody>
</table>