HW9

Guannan Shen

November 29, 2018

Contents


```
## set up workspace
library(class)
library(knitr)
library(tidyverse)
library(magrittr)
library(stats)
options(stringsAsFactors = F)
options(dplyr.width = Inf)
getwd()
```

```
## [1] "/home/guanshim/Documents/Stats/CIDA_OMICs/7659Stats_Genetics/HW9"
```

```
## not in function
"%nin%" <- Negate("%in%")

# ######## clean memory ####################### rm(list =
# ls()) gc() slotNames(x) getSlots(x)
```

1 Classification

• Download the data provided on Canvas (dataHW9-breastcancer.Rdata). This file contains the "breastcancer" object, which is a dataset for 46 breast tumor samples where 23 are positive for an estrogen receptor (ER+) and 23 were negative (ER-) (West et al., PNAS 2001 98:11462-11467). There are expression levels for 7129 genes for each sample in this list (x) and class labels for each sample (y).

• Install the class package from CRAN

1.1 (a) Describe the k-nearest neighbor algorithm and how it classifies observations. Using the function knn() from the class package, run k-nearest neighbors with k = 3 to train and test on the same training data set (train). What percentage of subjects were correctly classified? It is not good practice to train and test on the same data set, why not?

The KNN here is a non-parametric method used for classification. The output is a class label. An object is classified by a majority vote of its k nearest neighbors, and the classes of neighbors are known. Thus, this is a type of supervised learning.

This method using training examples with features and class labels. When a test point (have features but does not have a class label) is provided, the distance metric (such as Euclidean distance, correlation coefficients) is calculated to define its k nearest neighbors. Eventually, the test point is assigned to the majority labels of its neighbors.

```
load("dataHW9_breastcancer.Rdata")
## import train test data
train <- breastcancer[[1]]  #training expression data
trainclass <- breastcancer[[2]]  #training classes
test <- newpatients  #new expression data
testclass <- trueclasses  #new classes

## run knn()
k1 <- knn(t(train), t(train), trainclass, k = 3)
k1_err <- sum(!(k1 == trainclass))/length(k1)
k1_err
```

```
## [1] 0.06521739
```

In this case, 0.9347826 %