STATISTICAL METHODS IN DATA MINING

DR. ALPER VAHAPLAR

Classification in R

- ✓ Classification Examples in R
- ✓ Apply knn (k-nearest neighborhood algorithm) (manually for now ☺)

```
rm(list=ls()) # clear all variables
data(iris) # Loading iris data...
summary(iris[,1:4]) # let's remember the data set
› boxplot(iris, las=2)
› plot(iris[, 1:4], col=iris$Species)
```

Classification in R

- ✓ Apply knn (k-nearest neighborhood algorithm) (manually for now ☺)
- ✓ We need some functions like…

```
minmax = function (x) # to normalize min-max
{
  (x-min(x))/(max(x)-min(x))
}
zscore = function(x) # for standardization
{
  (x - mean(x))/sd(x)
}

euclid = function(x,y) # distance between x and y (for iris)
{
  y$uza=dist(rbind(x[,1:4],y[1:4]), method = "euclidean")[1:nrow(y)]
  return(y[order(y$uza),])
  # returns the ordered set bu "uza" in descending order
}
```

Classification in R

✓ Apply knn (k-nearest neighborhood algorithm) (manually for now ☺)

```
# min-max normalize iris data (Sepal/Petal – Length/Width)
iris_n = as.data.frame(lapply(iris[,c(1,2,3,4)], minmax))
iris_n$Species = iris$Species
# Let the 120th data be the unknown class.
x = iris_n[120,]
# calculate distances to all the others in the set
yeni = euclid(x, iris_n)
# now we have the normalized iris data with distances to x
# closer at the top, furthest at to bottom
# the fist data is x itself (with 0 distance)
```

Classification in R

- ✓ Apply knn (k-nearest neighborhood algorithm) (manually for now ☺)
- ✓ Let's look at the nearest k neighbors of x.

```
# Let k = 5

k=5

table(yeni[1:k,]$Species) # yeni[1,] is the "x" itself

# so k neighbors are...

yeni[2:(k+1),]

# We will decide according to the greatest class in neighbors.

table(yeni[2:(k+1),]$Species)
     setosa versicolor  virginica
        0          4          1

# Which means that 4 (of 5) of the neighbors of x is in
# "versicolor" class. So we should assign "x" to "versicolor"
# class.
```

Classification in R

- ✓ Apply knn (k-nearest neighborhood algorithm) (Now with a library)
- ✓ We need a training set and a test set. Let 120th data be our test set

```
library(class)
test = 120
iris_train = iris_n[-test,1:4] # the neighbors
iris_test  = iris_n[test,1:4] # unknown flower
iris_target_cat = iris_n[-test,]$Species # classes of neighbors
iris_test_cat = iris_n[test,]$Species # class(es) of x
# Apply knn for k=5
tahmin = knn(iris_train, iris_test, cl = iris_target_cat, k = 5)
table(iris_test_cat, tahmin)
```

Classification in R

- ✓ Apply knn (k-nearest neighborhood algorithm) (Now with a library)
- ✓ To test more than 1 flower, change the line "test = 120" with the test items

```
library(class)
test = sample(1:nrow(iris_n), 30) # reserve 30 data for test
iris_train = iris_n[-test,1:4] # the neighbors
iris_test  = iris_n[test,1:4] # unknown flowers
iris_target_cat = iris_n[-test,]$Species # classes of neighbors
iris_test_cat = iris_n[test,]$Species # classes of test data
# Apply knn for k=5
tahmin = knn(iris_train, iris_test, cl = iris_target_cat, k = 5)
table(iris_test_cat, tahmin)
```

Classification in R

- ✓ Apply knn (k-nearest neighborhood algorithm) (Now with a library)
- ✓ To test more than 1 flower, change the line "test = 120" with the test items

```
# Apply knn for k=5
tahmin = knn(iris_train, iris_test, cl = iris_target_cat, k = 5)
table(iris_test_cat, tahmin)
              tahmin
iris_test_cat setosa versicolor virginica
   setosa          9          0         0
   versicolor      0         14         0
   virginica       0          0         7
# All class predictions are correct.
```

Classification in R

- ✓ Apply knn (k-nearest neighborhood algorithm) (Now with a library)
- ✓ Try different k values

```
# Apply knn for k=3
tahmin = knn(iris_train, iris_test, cl = iris_target_cat, k = 5)
table(iris_test_cat, tahmin)
                tahmin
iris_test_cat setosa versicolor virginica
   setosa          9          0         0
   versicolor      0         13         1
   virginica       0          1         6
# 2 misclassifications this time.
```