Exercise Sheet 3: Clustering

Exercise 3.1 K-means clustering:

By using K-means clustering the dataset can be sparcified, and timing and accuracy performance could be obtained.

3.1.1 Try to improve the performance of two person training data (person independent). Perform K-means clustering of each cipher individually for the training set, in order to represent the training data as a number of cluster centroids. Now perform the training of the KNN using the centroids of these clusters. You can try with different cluster sizes and see the resulting performance.

If “id” is the dataset for the first person, whom the training is performed on.

```
 cipher_cluster <- c()
 label_cluster <- c()

for( i in 0:9) {
        clusterData <- kmeans(id[ id[,1] == i, -1 ], 200)
        cipher_cluster[[i + 1]] <- clusterData$centers
        label_cluster[[i + 1]] <- c(1:200)*0 + i
}

train_lab <- factor(unlist(label_cluster))
train_dat <- cipher_cluster[[1]]
for( i in 2:10) {
        train_dat <- rbind(train_dat,cipher_cluster[[i]])
}
```

Then train_lab and train_dat can be used as input for KNN.

3.1.2 Compare your KNN performance based on the raw training data and based on the cluster centroids of the training data. During the comparison you should also consider the run times of the algorithm.

3.1.3 Perform K-means clustering on each cipher individually for the training data from the entire class (person independent), in order to represent the training data as a number of cluster centroids. (Depending on the size of the dataset good values may vary, but for two person training data 50, 100, 200 and 400 Clusters showed interesting results. An alternative might be sparsify the data to match 1/2, 1/4, 1/8 and 1/16 of the full datasize)

Exercise 3.2: Hierarchical clustering

3.2.1 Show a low level dendrogram containing 5 instances of each digit (one or two person).

3.2.2 Use K-Means clustering to compress each digit into 5 clusters and show a low level dendrogram of this (two person).

3.2.3 Discuss the results and relate them to the cross validation tables from K-NN classification.

Exercise 3.3: Evaluation methods of KNN:

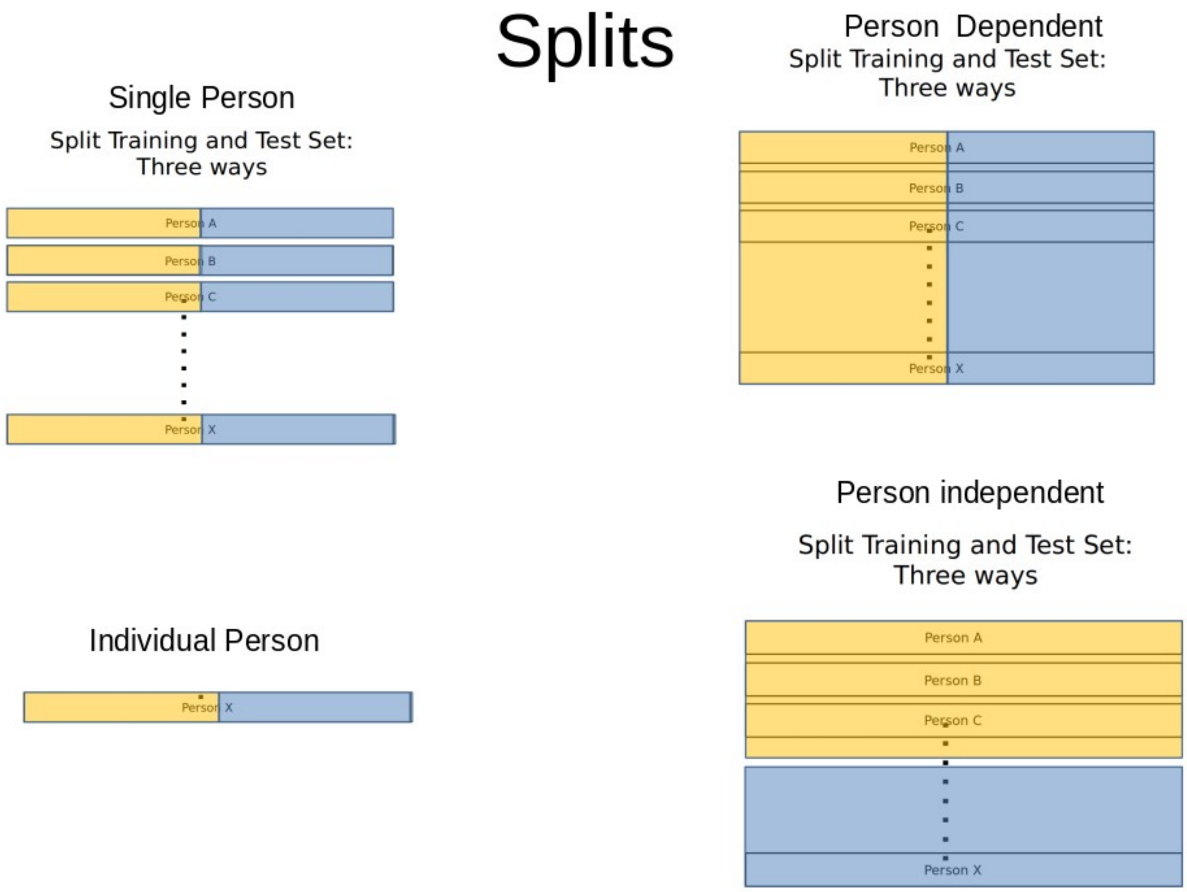
As seen in the hierarchical plot we often get different labels when finding the nearest neighbors of different ciphers. This could indicate that we are not completely sure about our estimate. Until now, in KNN we have simply used the one with most votes. But we can also exclude predictions which does not have enough of the same labels.

In knn we can set the “l” to the minimum number of “k” nearest neighbours of the strongest label to accept a match.

3.3.1 Plot the the precision-recall curves for 1 to 13 “k” with “l” values up to the “k” value. Here the results should be one plot containing “k” lines, who each have “k” datapoints.

3.3.2 Plot the maximum F1 values for each of the k in a plot together. With F1 score on the y-axis and “k”-value on the x-axis.

3.3.3 Discuss the results from 3.3.1 and 3.3.2. What do you think would be the most important part of a digit recognition system. Precision or recall, in what situations would the different things be important ?


Splits
Single Person
Split Training and Test Set:
Three ways
Person A
Person B
Person C
Person X
Person Dependent
Split Training and Test Set:
Three ways
Person A
Person B
Person C
Person X
Individual Person
Person X
Person independent
Split Training and Test Set:
Three ways
Person A
Person B
Person C
Person X