Exercise 1 – Data preparation and KNN

This week the exercise will consist of four main tasks. Downloading the SVN repository, create and prepare data, setup R and perform K-nearest neighbor classification. There will be included both a guide to do it in Windows and Ubuntu.

1.1 Connecting to SVN

The first task is to connect to the SVN server, both to receive the exercises, but also to share cipher sheets.

Windows, Ubuntu & Mac:

1.1.1 Install svn - http://www.smartsvn.com/

1.1.2 Make a folder to place rep in, e.g "trunk".

1.1.3 Start SmartSVN and select "Check out project from repository".
Repository: "https://aero.mmmi.sdu.dk/svn/SML-database/trunk"
username: "SMLstudent" and password: "SMLstudent".

1.1.4 Select "trunk"

1.1.5 Checkout into your created directory

1.2 Data collection and preparation

The goal of the second task is to create and share the data used for the course. Multiple DPI are needed to test the result of different resolutions.

1.2.1 Fill in the forms from class (also available on svn) with digits as indicated on the sheets.

1.2.2 Using the scanners on the school scan the images as a jpeg. Remember to set DPI, (100, 200, 300). 15 images should be produced.

1.2.3 Name them as following Ciphers"DPI"-"INDEX".jpeg (i.e. the first 100 DPI sheet becomes Ciphers100-0.jpeg)

1.2.5 You now need to create the file containing the corners of the large boxes each containing 400 ciphers, this should be named Corners.txt. Using e.g. Paint or Gimp, open the Ciprhers300 png's and the following corners should be found: UL, UR, DL, DR. (U = up, D = down, L = left, R = right)(see image). The rows represent the cipherboxes from 0-9.

1.2.6 Inspiration for the complete data folder can be seen in "/trunk/2017/group0/member1"

1.2.7 Lastly you will commit the images to the svn repository. Create for your group in the folder "2018", a folder with the group number, e.g. "group1", and in that create a folder named with your group number, e.g. resulting in "trunk/2017/group1/member2". Now you will commit this to the svn.

Press "Commit" to upload your own changes and "Update" to receive the newest version of the repository. Please don't change any other files in the repository than your own ciphers.

1.3 Installing R and libraries

This tasks concerns installation of R and Rstudio, respectively the programming language and the graphical frontend. Additional software packages used in the course are also installed.

1.3.1 Install R and Rstudio.

Windows & Mac:

http://cran.r-project.org/bin/windows/base/,
get Installer (not tarball) from http://www.rstudio.com/products/rstudio/download/

Ubuntu:

```
“sudo apt-get install r-base r-base-dev”
```

get Installer (not tarball) from http://www.rstudio.com/products/rstudio/download/ and use software manager

1.3.2 When R and r-studio is running you need to install the package, EBImage, by typing:

```
source("http://bioconductor.org/biocLite.R")

biocLite("EBImage")
```

1.3.3 You also need to get some packages, missing packages can be installed by the command install.packages(), e.g. to install gmodels, type:

```
install.packages("gmodels", dependencies=TRUE)
```

1.3.4 By sourcing the file “loadImage.R” in “Basefolder” the function “loadSinglePersonsData(DPI,groupNr,groupMemberNr,folder)”, can be used to get the data from the images.

The data is represented as a matrix, using data.frame it is converted giving 4000 obs. of (325 variables, 1445 variables or 3365 variables, depending on the DPI). The first element of the variables is the correct cipher).

1.3.5 Documentation on R can be found on http://www.r-project.org/

1.4 Performing K-nearest neighbor (Report 1)

The following exercise will concern the topic of today and it is required that the exercises are made into a report. The method will be tested on your created cipher data.

The dataset should be split into 50/50 for training and testing. Before splitting the dataset can be shuffled using: (remember to set seed for reproducable results , e.g. “set.seed(423)”):

```
dataset_shuffle <- dataset[sample(nrow(dataset)),]
```

1.4.1 K-Nearest Neighbour: Using the methods learned in the Chapter 3 in “Machine Learning With R”, KNN can now be performed on your own generated dataset. Remember to split between training and test set (split in two equally sized parts). Document the results. Can you explain the performance on the training and test-set?

1.4.2 Performance of varying K: Show how performance (speed and test recognition) changes with changing K.

1.4.3 Cross validation: Perform a cross validation with a 90% / 10% split with 10 runs. Report mean and standard deviation of the performance.

1.4.4 Preprocessing: Apply one of the two smoothing functions to the to the images, instead of the one implemented:

- Average over four neighboring fields (apply this procedure hierarchically)
- Gaussian smoothing with various sigmas

Perform again the steps in task 1.4.3 (**Cross validation**). Describe and analyze the results for one of the smoothing methods depending on the amount of smoothing.

1.4.5 Person independent KNN: Now try to apply k-nearest neighbor classification to the complete data set from all students attending the course. Distinguish two cases: Having data from all individuals in the training set and splitting the data according to individuals. Generate and explain the results.

1.4.6 Performance of sample size: Lastly report on timing of the prediction step for varying k and using a small and large dataset. You don’t have to test every K simply give an overview on the performance’s dependency on smaller and larger K and dataset. Discuss how the performance changes with different sizes of the dataset, is K dependent on the dataset size?