Project descriptions and tasks in classification.

Fagansvarlig: Magne H. Johnsen

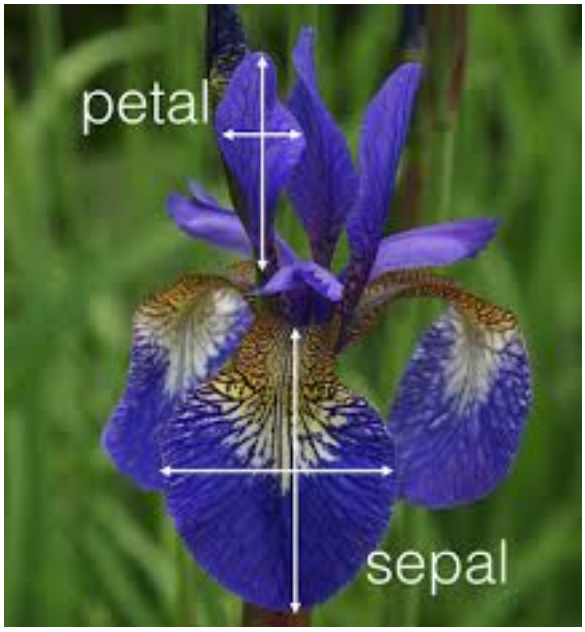
27. februar 2018

All students will be organized into groups of two. The Iris task is to be done by all groups which choose a classification task. The groups shall in addition choose one of the tasks "vowels" or "handwritten numbers". All experiments shall be implemented in Matlab or other preferred languages (as Python).

1 The Iris task

The Iris flower has several variants including three called respectively Setosa, Versicolor and Virginica, see figure 1. The flower has both large (Sepal) and små ll (Petal) leaves, see figure 2. The three mentioned variants can be discriminated by the different lenghts and widths of the petal and sepal; i.e. these four measurements are a logical choice for input features.
A database (often called "Fisher Iris data") is produced, consisting of 50 examples of each of the three variants/classes . We refer to https://en.wikipedia.org/wiki/Iris_flower_data_set for a more detailed description of the database.
The Iris task is one of a few practical problems which are close to linearly separable. Thus an error free linear classifier can be designed for the database. This first part of the project therefore has focus on the design and evaluation of a linear classifier. Further one should analyze the relative importance of each of the four feature with respect to linear separability.

Figur 1: The three Iris variants



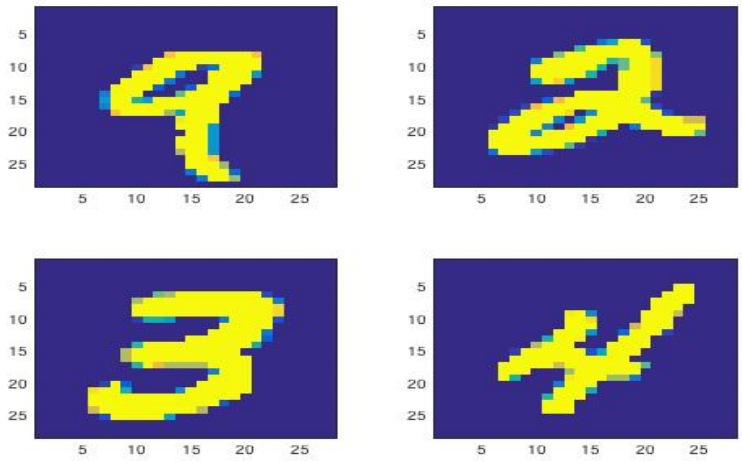
Figur 2: Length and width for repectively petal og sepal

The task consists of two parts

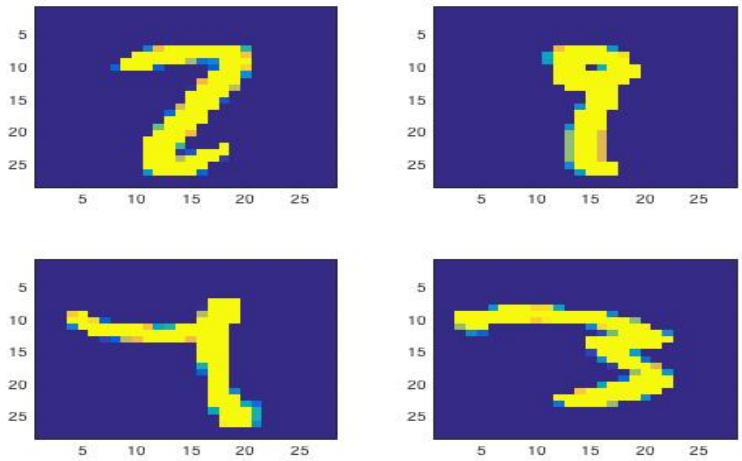
1. The first part has focus on design/training and generalization.
   (a) Choose the first 30 samples for training and the last 20 samples for testing.
   (b) Train a linear classifier as described in subchapter 2.4 and 3.2. Tune the step factor α in equation 19 until the training converge.
   (c) Find the confusion matrix and the error rate for both the training and the test set.
   (d) Now use the last 30 samples for training and the first 20 samples for test. Repeat the training and test phases for this case.
   (e) Compare the results for the two cases and comment
2. The second part has focus on features and linear separability. In this part the first 30 samples are used for training and the last 20 samples for test.
   (a) Produce histograms for each feature and class. Take away the feature which shows most overlap between the classes. Train and test a classifier with the remaining three features.
   (b) Repeat the experiment above with respectively two and one features.
   (c) Compare the confusion matrixes and the error rates for the four experiments. Comment on the property of the features with respect to linear separability both as a whole and for the three separate classes.

2 Classification of handwritten numbers 0-9

NIST is the USA variant of the Norges Forskningsråd (NFR). NIST has designed a database, called MNIST (see yann.lecun.com/exdb/mnist), with pixtures of handwritten numbers 0-9. The pixtures have dimension 28x28 pixels and are in 8-bit greyscale; i.e. pixel values between 0-255. For practical purpose one should note that the the pixtures have been "preprocessed"; i.e. centred and scaled to prepare them for classification. Figure 3 shows four "easy" examples, while figure 4 shows examples which are harder to classify correctly. A large amount of classifiers have been designed for this case, resulting in error rates between $1 - 10$ %. The state-of-the-art is (of course) a deep neural network (DNN).
The database consists of 60000 training examples written by 250 different persons and 10000 test examples written by 250 other persons.



Figur 3: "Easy" examples of the numbers 9, 2, 3 og 4



Figur 4: "Dubious" examples of the numbers 2, 9, 7 og 3. Or the numbers 7, 1, 4 og 7 ?

The task consists of two parts both using variants of a nearest neighbourhood classifier.

1. In the first part part the **whole** training set shall be used as templates.

   (a) Design a NN-based classifiser using the Euclidian distance. Find the confusion matrix and the error rate for the test set. The data sets should preferably be split up into chunks of images (for example 1000) in order to a) avoid too big distance matrixes b) avoid using excessive time (as when classifying a single image at a time)

   (b) Plot some of the misclassifed pixtures. Some useful Matlab commands for this are :

   - **x = zeros(28,28); x(:)= testv(i,:);** will convert the pixture vector (number i) to a 28x28 matrix
   - **image(x)** will plot the matrix x
   - **dist(template,test)** will calculate the Euclidian distance between a set of templates and a set of testvectors, both in matrix form.

   (c) Also plot some correctly classified pixtures. Do you as a human disagree with the classifier for some of the correct/incorrect plots?

2. In the second part you shall use clustering to produce a small(er) set of templates for each class. The Matlab function $[\mathbf{idx_i}, \mathbf{C_i}] = \mathbf{kmeans}(\mathbf{trainv_i}, \mathbf{M})$; will cluster training vectors from class ω_i into M templates given by the matrix C_i.

   (a) Perform clustering of the 6000 training vectors for each class into $M = 64$ clusters.

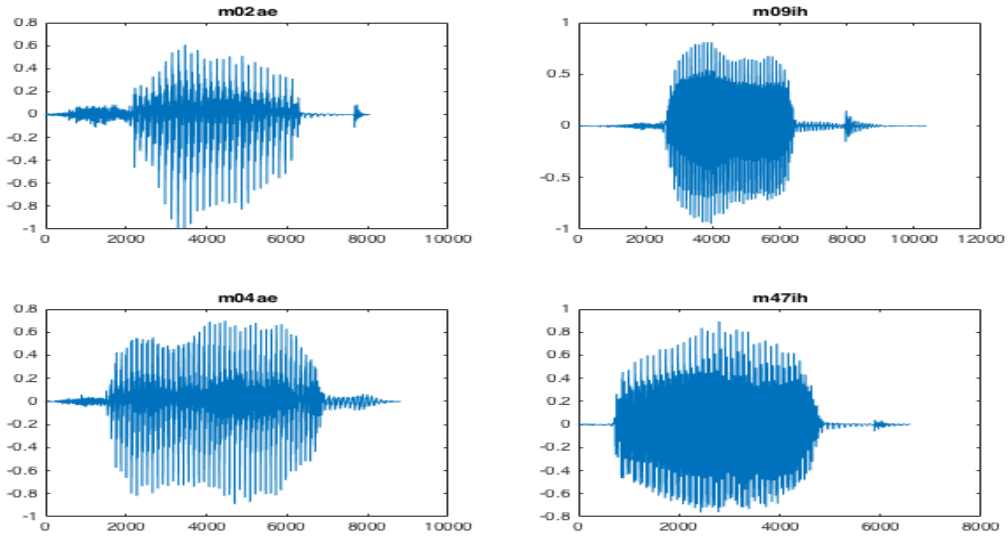
   (b) Find the confusion matrix and the error rate for the NN classifier using these $M = 64$ templates pr class. Comment on the processing time and the performance relatively to using all training vectors as templates.

   (c) Now design a KNN classifier with K=7. Find the confusion matrix and the error rate and compare to the two other systems.

3 Classification of pronounced vowels

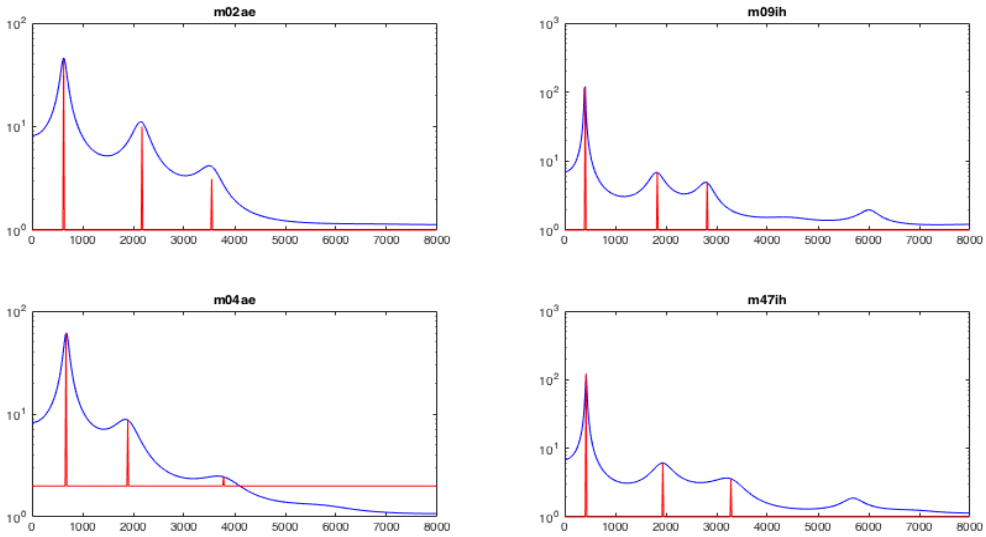
In this task we will try to classify a full set of twelve vowel classes. The data set consists of 139 recordings of each vowel class. The first 70 samples shall be used for training while the remaining 69 samples shall be used for testing. From each recording three features, so called formants, are extracted. The formants represent three frequencies where the sound have peaks in the frequency spectrum. Typical formant frequences differ from vowel to vowel, however they also show a significant variation within the classes. Thus, we have a classical nonseparable problem. The waveform files has five letter names with extension wav. The first letter indicates man/female/boy/girl, the next two person number and the two last the vowel name. In exercise 2 (dealing with classification) a subset of three classes were analyzed.

In figure 5 we see two waveform examples from each of the vowels ”AE” and ”IH”. Note that the vowels (large energy) are surrounded by consonants (i.e. CVC). It is not easy to decide upon which examples belong to which class just by inspecting the waveforms.



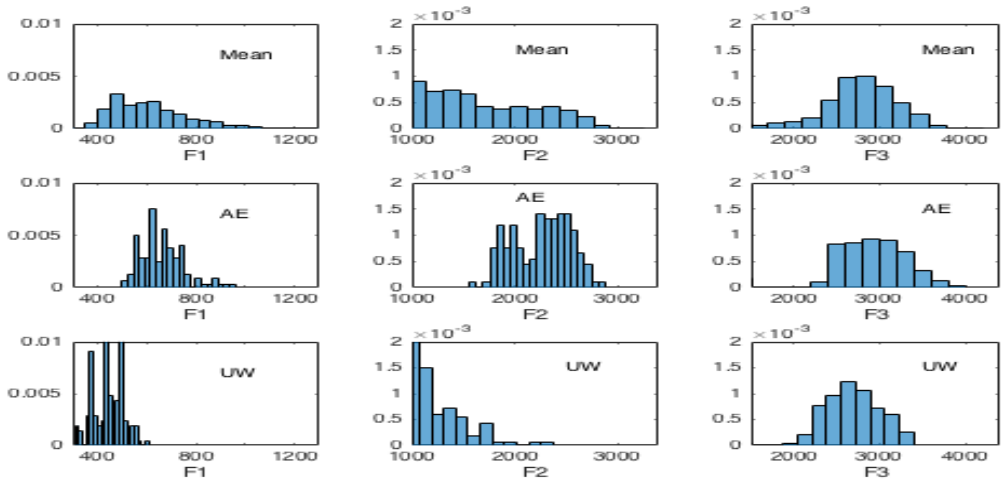
Figur 5: Examples of CVC waveforms for the two vowels "AE"and "IH"

Figure 6 shows (estimates of) the frequency spectra of the vowel examples in figure 5. We see that the spectra have clear peaks (formants). The three first/dominant formants are indicated by red. We further see that the formants are quite similar for the same class while the first and third formant differ significantly between the classes. Thus it should be possible to use the formants todiscriminate between the classes.



Figur 6: Examples of spectrograms and formants for the two vowels "AE"and "IH"

In figure 7 formant histograms are plotted for the vowels "AE"and "UW"and the mean over all twelve classes. For the two shown vowels especially the first two formants differ.



Figur 7: Formant histograms for the two vowels "AE"and "UW"

The task consists of two parts which both use Gaussians as models.

1. In the first part part a single Gaussian class model with respectively a full and a diagonal covariance matrix is to be analyzed.

   (a) Find the sample mean and the sample covariance matrix for each class set.

   (b) Design a classifier assuming full covariance pdfs and find the confusion matrix and the error rate for the test set.

   (c) Repeat the above for diagonal covariance matrixes (i.e. set off-diagonal matrix values to zero). Compare the confusion matrixes and the error rates.

2. In the second part you shall use a mixture of 2-3 Gaussians for each class. The Matlab function $\mathbf{GMM_i} = \mathbf{fitgmdist}(\mathbf{trainv_i}, \mathbf{M})$; will put M mixtures into GMM_i using the training vectors $trainv_i$ from class ω_i. The function **mvnpdf** will calculate the likelihoods for a test set. Another option is to use a class of function **gmmdistribution.fit** to do the training and the function **pdf** in the test phase

   (a) Find GMM models with diagonal covariance matrixes for both 2 and 3 mixtures for each class.

   (b) Modify the classifier in task 1b to deal with mixture of gaussians

   (c) Find the confusion matrixes and the error rates for the GMM classifier using respectively $M = 2$ and $M = 3$ Gaussians pr class.

   (d) Compare the performances for all four model types (tasks 1b, 1c and 2c). For which classe(s) is the difference largest?