Pattern Recognition Laboratory – Assignment #2
Optimal Bayes Classification

Due date: 25.03.2024 (LAB/101)

In this assignment, your task will be to prepare Bayesian classifiers with different methods of calculating the conditional probability density distributions. That density should be computed for each class. You'll be comparing three methods of determining the density :

1. Assuming that the features are independent and each attribute distribution is normal (in this case the probability density for more than one feature can be calculated as the product of the density for each feature).
2. Assuming that we are dealing with a multi-dimensional normal distribution of the features.
3. Using Parzen window to compute the probability density approximation based on the training set.

The first two classifiers can be named parametric classifiers in that we compute only parameters of normal distribution based on the samples in the training set. In the last case we'll use the training set to compute approximation of the unknown distribution (i.e. we'll compute value of probability density for the sample to be classified). Quite common method for approximation of unknown probability distribution is to use Parzen window. It is based on the fact that an unknown distribution density is "built" on the samples in the training set. Each sample brings a small partial density share, located in the vicinity of the sample.

For this assignment, we assume the window function to be multinomial normal distribution (and we'll compute it as a product of one-dimensional distributions for each feature). The only parameter we'll have to supply is the window's width. You'll have to check different width values in interval <0.0001, 0.01>.

Training set for this assignment uses as features Hu moment invariants (http://en.wikipedia.org/wiki/Image_moment) of the scanned images of cards suits. The first column contains class identifier (4 – spades, 3 – hearts, 2 – diamonds, 1 – clubs). It's worth noting, that suits were printed with different methods: on half of images there is printing raster and on half there is no such raster. Before starting classification we change labelling to take into account these observation.

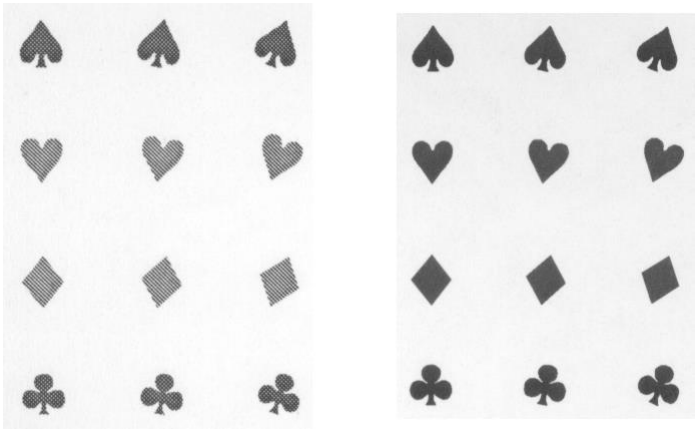

Fig. 1 Images of card suits printed on black and white (left) and colour (right) printer.

The first step to be performed is to implement the pdf functions (probability density function): `pdf_indep.m`, `pdf_multi.m`, `pdf_parzen.m`.
Note that functions computing parameters for these pdf functions are already implemented (`para_indep.m`, `para_multi.m`, `para_parzen.m`).

To check pdf functions, use the data from the `pdf_test.txt` file (2 classes and only 20 samples in two dimensions).

Parzen window approximation of a pdf requires a comment. The value of density is "assembled" here, counting the shares of samples from the training set at the point where we should calculate the probability density. There are no shortcuts here: for each sample x you have to compute one-dimensional pdf for each feature (here we have `number_of_samples_in_class * number_of_features` values), and then properly aggregate them:

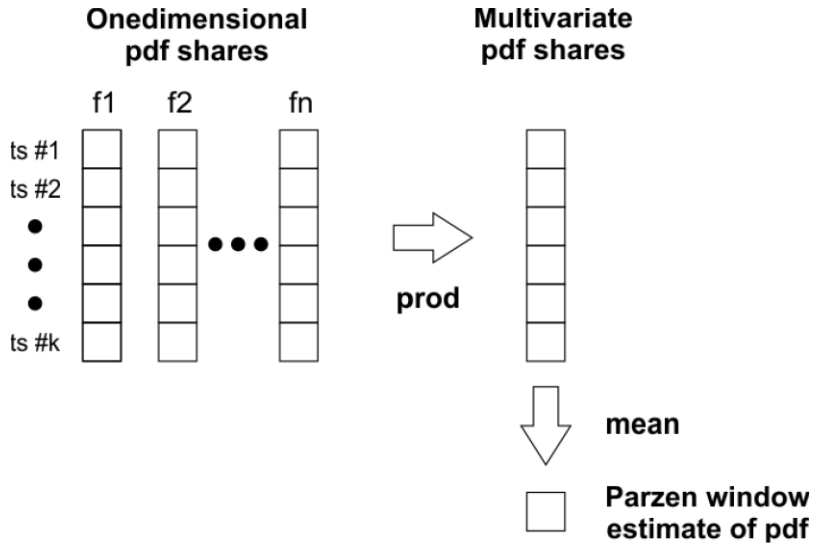

Fig. 2 Illustration of computing Parzen window estimate of pdf of given class at point x.

Quite rational selection for the window function $\varphi(u)$ is normal distribution (we have in Octave `normpdf` defined). The value of a specific feature of sample x should be supplied as the mean argument and the window width h_1, adjusted to the number of samples for given class in the training set: $h_n = \frac{h_1}{\sqrt{n}}$, should be supplied as the standard deviation argument.

After implementation and verification of pdf functions, you can deal with the recognition of card suits data:

1. Check the data, esp. the training set. Outliers can change significantly computed distribution parameters, which can dramatically reduce recognition quality. You can try here to compare `mean` and `median` values, plot histogram of individual features (`hist` function) …
   To remove a sample with known index `idx` use expression:
   ```
   train(idx, :) = [] ;
   ```
2. Select two features (note that you have `plot2features` function supplied) and build three Bayes classifiers with different probability density computations (according to points 1-3 above). You should use equal *a priori* probabilities of 0.125.
3. Check how the number of samples in the training set influences the classification quality (you can take for example 10%, 25%, 50% of the whole training set).
   Note: an appropriate part of the samples from the training set should be drawn independently from each class; because we introduce a random element, the experiment must be repeated (minimum 5 times) and report should contain averaged results (good practice is to include not only mean value but also a standard deviation).
   Here you should implement `reduce` function, which leaves the appropriate part of each class. At this point, the reduction applies only to the training set.
4. Check how width of the Parzen window h_1 influences the classification quality (note that this point has sense for Parzen classifier only).
5. How will the classification results change if the *a priori* probability will be two times higher for black suits, i.e. (0.165, 0.085, 0.085, 0.165, 0.165, 0.085, 0.085, 0.165)?

Note that in this case you should reduce number of red suits in the **testing set** only!

6. What is the classification quality of the 1-NN classifier (`cls1nn.m`) for these data?
   Don't use in this case leave-one-out method, you have large enough testing set at your disposal. Think about data normalization. If there is big difference in standard deviations between features you should normalize data before classification.

Please look carefully at the results of your experiments: Do they make sense? Do they agree with your expectations? Is there something unexpected there?

I expect written report – concise, but containing the important information and of course experiment results (preferably in pdf format). You should also submit your Octave code used in this assignment (from time to time I want to reproduce your results). Pack all these files into zip archive and submit in Leon (task Bayes classifier) until our next lab meeting.

I definitely **do not want to receive original datasets**: neither training nor testing. For sending your solution with the data I will subtract 1 point.

List of files provided:

bayescls.m	- Bayes classifier; pdf function and its parameters are passed as parameters
cls1nn.m	- good old 1NN classifier
epart_l2.pdf	- this instruction
load_cardsuits_data.m	- card suits loading with label mapping (1-4 -> 1-8)
mainscript.m	- main experiment notebook ☺
mvnpdf.m	- multivariate normal distribution (in case you don't have statistics package)
normpdf.m	- univariate normal distribution (in case you don't have statistics package)
para_indep.m	- computes parameters for `pdf_indep` function
para_multi.m	- computes parameters for `pdf_multi` function
para_parzen.m	- computes parameters for `para_parzen` function
pdf_indep.m	- computes pdf value assuming independence of features
pdf_multi.m	- computes pdf using multivariate normal distribution
pdf_parzen.m	- computes pdf using Parzen window approximation
pdf_test.txt	- small data set used to verify `pdf_*` functions
plot2features.m	- plots scatter graph of two features
reduce.m	- reduces number of samples in data set according to class reduction coefficients
test.txt	- card suits testing set
train.txt	- card suits training set

Files printed in red require your special attention ☺