Pattern Recognition Laboratory – Exercises #3 & #4
Recognition of handwritten digits

Due date: **19.04.2018**

In this exercise, your task will be to recognize handwritten digits . We'll use one of the most widely used data sets in the pattern recognition MNIST handwritten digits (http://yann.lecun.com/exdb/mnist). Learning and test sets can be downloaded at the address given above. In addition they are available on the Galera server.

You should know that the images are normalized after scanning as follows:

1. The rectangle containing the black-and- white image of a character scanned at a resolution of 300 dpi is scaled proportionally to fit into square 20 by 20 pixels. During scaling an image is converted to grayscale.
2. The center of gravity of the scaled character is determined, and character is placed in a 28x28 pixel image so that the center of gravity lie in the middle of the bigger image.

Link to the dataset on the Galera server is following:

http://galera.ii.pw.edu.pl/~rkz/epart/mnist.zip (original MNIST data)

Your task is to produce a classifier that uses **linear classifiers** distinguishing individual digits. In addition to the quality of classification on the test set you should produce **confusion matrix**. Additional task is to propose a different (let's hope - better) method for determining the classifier decision than simply voting of elementary classifiers.

The point of reference is the classic voting (45 linear ovo, i.e. *one versus one*, classifiers) classifying pairs of digits if it collected maximum possible number of votes = 9. In other cases, voting classifier makes reject decision. The tests used the first 40 principal components. Classification results are summarized in the table below (although an interesting insight into the classification may give confusion matrix analysis).

	MNIST Training Set			MNIST Testing Set		
	OK.	Error	Rejection	OK.	Error	Rejection
Classification coefficients	91.34%	5.72%	2.94%	91.55%	5.49%	2.96%

The task can be divided into a few parts:

1. Preparation of the procedure to compute separation plane parameters given a training set containing just two classes. The easiest way to accomplish it is to use two-dimensional data sets, which can be visualized together with the separating plane.
2. Checking the algorithm for multidimensional digits data. You should store individual one versus one classifiers quality to put them in your report. Although you can use directly pixel data I suggest you to reduce dimensionality with PCA (40-80 primary components).
3. "Canonical " solution is 45 voting classifiers - one for each pair of digits - and making the final decision with unanimity voting (only digits with 9 votes are classified; if the number of votes is smaller classifier produces reject decision). You

should report individual classifier error rates as well as quality of the ensemble. Quite interesting insight into ensemble operation can give you confusion matrix.

4. The final step - which in my opinion you should implement after your canonical classifier is ready - is the enhancement of the canonical solution.

   The general idea is following: our base classifiers have unequal error rate. For weak classifier it seems probable the separating plane between two classes is not linear. What's more, we can suspect that there are several clusters within just one digit. Let's perform clustering of samples one digit to find finer structure in our data. Perform clustering for the second problematic digit data.

   Now you have the possibility to train more than one linear classifier. You should think about which classifiers you should train and at the same time you should develop voting schema for the ensemble. After implementing your solution you should check error ratio of your two-digit ensemble (and of course compare it with original base classifier). As the final step you should "turn-off" base classifier in the canonical solution and add in this place votes of your two-digit ensemble.

Your report should include:

1. Description of the basic (canonical) voting method.
2. Error rates of individual basic classifiers.
3. Classification quality (as in the table above) and confusion matrix data.
4. Description of non-standard two-digit classification ensemble and comparison of its results with respect to the base classifier.
5. Final classification quality and confusion matrix data and comparison with the canonical classifier results.
6. You should send your report together with the code but **you should not send the data sets**.