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

Due date: **23.11.2017 (A), 30.11.2017 (B), 30.11.2017 (C)**

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 database 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).

Here you can ask several interesting questions: What if I select the class with the highest number of votes (without requiring it to be 9)? What if more than one class gets the maximum number of votes?

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 weak point in preparing our canonical solution is the fact that each of the individual classifiers was selected on its own recognition quality. It can be that slightly inferior two-digit classifier will work better in ensemble with other 44 classifiers.

   It is however not clear which classifiers in the canonical classifier should be "optimized" in the first place. I would try to identify them by analyzing confusion matrix of the canonical classifier in the following manner.

   In the first step I would find a few highest confusion "spots". Although source of such confusion is not clear the chances are that replacing the individual classifier for this pair of digits can lead to better overall classification. So I would train 10 (it's just an example) classifiers for this pair of digits and check recognition quality of 10 non-canonical ensembles (using each of my 10 candidates as replacement for specific classifier in canonical solution).

   If any attempt will have given better recognition quality I would replace individual classifier in the canonical solution.

   I can repeat this procedure for other weak spots.

   Very important is to use only the training set in this search for best ensemble.

   The last step is to test the new ensemble on the test set. Are there differences in classification quality on the training and testing set? Are these differences similar to canonical solution? ….

Your report should include:

1. Description of the basic (canonical) voting method.
2. Classification quality (as in the table above) and confusion matrix data.
3. Description of non-standard classification and comparison of its results with respect to the canonical solution.
4. You should send your report together with the code but **you should not send the data sets**.