Machine Learning for Applications in Computer Vision: Week 1

Date: Friday, 17. April 2015

Please work in groups of 2–3 people. We will check your solutions next week after the lecture. Please be prepared to present your solution and explain the code.

Download the MNIST Dataset

Please download MNIST Digit Recognition Dataset from: `http://yann.lecun.com/exdb/mnist/`
Download all 4 .gz files.

Coding

Please code in Python.
You can use `scikit-learn` Python package for machine learning algorithms.
The documentation is on `http://scikit-learn.org`

Exercise 1: Dataset Preparation

1. Once you have downloaded the datasets extract the images and labels.
2. Use the image pixels as features. Reshape the 28×28 images to 1D vectors (with length 784).

You can ask the tutors how to read database in case you could not do it yourself.

Exercise 2: Support Vector Machines

1. Make yourself familiar with `scikit-learn` and the SVM tool
2. Load the training data (images and labels) into your program. You might have to unpack the data set and convert it to a reasonable format.
3. Train the default SVM on the first 1,000 samples.
4. Check the model you trained against the test set (NEVER do training with the test set!).
5. Tune the parameters (particularly `kernel`) to find a good set of hyperparameters.
6. Now run your training on the entire dataset and check it against the test set.
7. Report the methods used and your results (cf. lecture slide).

Exercise 3: Decision Trees and Random Forests

1. **Decision Trees**

   (a) Train a decision tree with default parameters and output the classification score for the test images.

   (b) Tune the parameters of the decision tree and find the best hyperparameters:

   * criterion (gini and entropy)
   * max_depth
   * max_features

   (c) Visualize the constructed tree using pydot.

   (d) Visualize the pixel importances on a 28×28 image.

2. **Random Forests**

   (a) Construct a Random Forests classifier with default parameters and output the classification score for the test images.

   (b) Tune the parameters of the decision tree and find the best hyperparameters:

   * n_estimators
   * criterion (gini and entropy)
   * max_depth
   * max_features

   (c) Visualize the pixel importances on a 28×28 image.

3. **Cross-Validation**
   Apply cross-validation on a decision tree and random forests using the training images and output the mean scores. You can use the best hyperparameters.