Machine Learning for Applications in Computer Vision: Week 3

Liu, Nan
Wiedemann, Andreas

GitHub: `https://github.com/DJLoco/TUMMachineLearning.git`

Exercise 1: Boosting

The objective is to use the Boosting framework in MATLAB to train a AdaBoost classifier with the MNIST hand-written digits dataset. Here is our work:

- Load the MNIST data into MATLAB program, unpack the dataset and convert it into arrays:
  - Training images: 60000*784 uint8
  - Training labels: 60000*1 uint8
  - Testing images: 10000*784 uint8
  - Testing labels: 10000*1 uint8
- Train the default AdaBoost on the first 1000 samples using fitensemble:
  ```
  ClassTreeEns = fitensemble(TrainImages(1:1000,:),
                    TrainLabels(1:1000),'AdaBoostM2',1000,'Tree');
  ```
- Test against the test set using misclassification error:
  ```
  L = loss(ClassTreeEns,TestImages,TestLabels);
  ```
  The classification error is 0.3765.
- According the figure, when learning rounds is 800, the resubstitution loss is the least, which is 0.3.
- Run the training on the entire dataset. The classification error becomes 0.3300.

- Train the LPBoost on the first 1000 samples:

  ```
  ClassTreeEns = fitensemble(TrainImages(1:1000,:),
                     TrainLabels(1:1000),'LPBoost',1000,'Tree');
  ```

  The classification error is high, which is 0.6478.

Exercise 2: Gaussian Process Classification

The objective is to get familiar with the Gaussian Process Classification tool and compare with the Boosting classification method.

To get to know the Gaussian process we changed the parameters. The inference method specifies how the (approximate) posterior for a Gaussian process is calculated. Several methods are offered. We tried `'infEP'` and `'infLaplace'`, however there was no great impact on the result.

For the covariance function we utilised `'covSEard'` and `'covSEiso'`.

Finally we tried different likelihood functions. There was a great change not only in the prediction but also the performance of the code. We obtained the most accurate results with `'likErf'` and `'likLogistic'`.

You can find our code `ex2.m` and `histo.m` within the doc folder of gpml. Unfortunately the prediction of new test data seems to be incorrect. We tried to follow `demoClassification` by finding all the indices where $\exp(lp) > .5$. This is true for almost every test data. Thus, we can merely guess what the histogramms should show. False predictions will often happen when $p(x|X,y) \approx \frac{1}{2}$, which will lead to a maximal uncertainty.

Reference

http://de.mathworks.com/help/stats/fitensemble.html

http://yann.lecun.com/exdb/mnist/

http://gaussianprocess.org/