Practical work 04 – 11/10/2018
Model Selection

Objectives

The main objectives of this Practical Work for Week 4 are the following :

a) Play through an example of overfitting and determine the optimal model complexity by using hyper-parameter tuning based on 5-fold cross validation.

b) Compose the confusion matrix from the scores obtained from a classification model, compute the different performance measures and learn what the characteristics these are capable of highlighting.

Submission

- **Deadline** : Wednesday 24 October, 12am
- **Format** :
  - Exercise 1 (Hyper-Parameter Tuning)
    - iPython notebook.
    - Small pdf-report with a learning curve plot.
  - Exercise 2 (Performance Measures) :
    - iPython notebook.
    - Small pdf report with the answers to the questions.

Exercice 1 Hyper-Parameter Tuning

The objective of this exercise is to build a classification systems to predict whether a student gets admitted into a University or not based on their results on two exams [1].

You have historical data from previous applicants that you can use as a training set (see `ex1-data-train.csv`). For each training example i, you have the applicant's scores on two exams $(x_1^{(i)}, x_2^{(i)})$ and the admissions decision $y^{(i)}$. Your task is to build a classification model that estimates an applicant's probability of admission based on the scores from those two exams. As test set you should use see `ex1-data-test.csv` .

In the notebook see `student_data_set.ipynb`, you'll find the code to load the data and further instructions.

a) Construct a dummy predictor that does random predictions. Show numerically that it produces a performance equals to 1/#classes - i.e. here equal to 50%.

b) Construct different models of different complexity (different number of parameters). Train these models with the training set and use the trained parameters for doing predictions and measure the error rate on the training set and the test set. Use 5-fold cross validation and remember the validation error for each of the folds.

   Remark : DO the programming so that you can easily change the input dataset.

c) Determine the model best suited for the problem at hand and justify why it is the best model.

Exercice 2 Confusion Matrix and Performance Measures

Let's assume we have trained a digit classification system able to categorise images of digits from 0 to 9.

After training, the system has been run against a test set (independent of the training set) including $N_t = 10'000$ samples. The system's output are Soft-Max like, i.e. estimations of a posteriori probabilities $P(C_k|\mathbf{x})$ for $k = 0, 1, 2 \ldots, 9$.

In file `ex1-system-a.csv`, you find the output of a first system A with the a posteriori probabilities $P(C_k|\mathbf{x})$ in the first 10 columns and with the ground truth y in the last column.

a) Write a function to take classification decisions on such outputs.

b) What is the overall error rate of the system ?

c) Compute and report the confusion matrix of the system A.

d) What are the worst and best classes in terms of precision and sensitivity (recall) ?

e) In file `ex1-system-b.csv` you find the output of a second system B. What is the best system between A and B in terms of error rate and F1.

1. Data source : Andrew Ng - Machine Learning class Stanford

Exercice 3 Optional : Review Questions

a) Explain the terms bias and variance. What are the factors that make the bias larger or how can it be made smaller ? What factors lead to a large variance or how can it be reduced ?

b) Why is the training error an increasing function of the split ratio (fraction of samples used for training) ? Why is the validation error a decreasing function of the split ratio ?

c) Describe in words how you construct a confusion matrix. How can you compute the accuracy from it ? How does this relate to the error rate ?

d) Describe the terms class accuracy, recall and precision. Describe typical situations where you would like to obtain a high recall or a high precision, respectively.

e) Explain what is understood as 'Curse of Dimensionality' and why deep models are believed to (at least partly) overcome this problem ?