MASTER OF SCIENCE IN ENGINEERING

Teachers: J. Hennebert, A. Perez-Uribe
Assistants: L. Rychener, C. Gisler

HES-SO

Machine Learning

Practical work 05 – 17th of October 2017 Classification Systems - Logistic Regression - Systems Evaluation

---

Summary for the organisation :

- Submit the solutions of the practical work before Monday 12h00 next week in Moodle.
- Preferred modality : iPython notebook.
- Alternative modality : pdf report.
- The file name must contain the number of the practical work, followed by the names of the team members by alphabetical order, for example `02_dupont_muller_smith.pdf`.
- Put also the name of the team members in the body of the notebook (or report).
- Only one submission per team.

Exercice 1 Confusion Matrix

Let's assume we have trained a digit classification system able to categorise images of digits from 0 to 9, as illustrated on Figure 1.

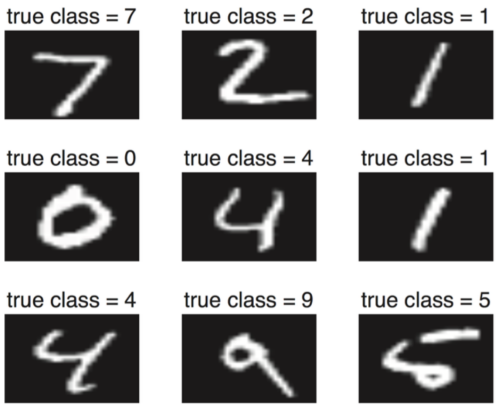

FIGURE 1 – Digit classification system

After training, the system has been run against a test set (independent of the training set) including $N_t = 10'000$ samples. The system is able to compute 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 according to Bayes'rule.

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

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

d) What are the worst and best classes in terms of 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.

Exercice 2 Classification system

This is a continuation of the exercise of previous week, where the objective is to build a classification system to predict whether a student gets admitted into a university or not based on their results on two exams [1]. For each training example n, you have the applicant's scores on two exams $(x_{n,1}, x_{n,2})$ and the admissions decision y_n that is illustrated on Figure 2.

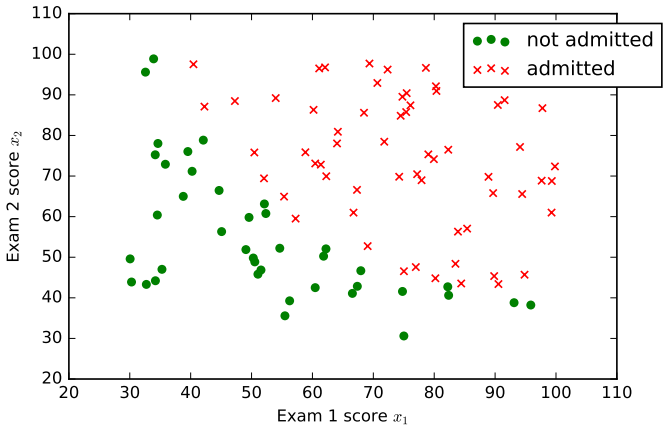

FIGURE 2 – Training data

a. Getting the training data

In a similar way as for the exercise of the previous week, read the training data from file `ex2-data-train.csv`. The first two columns are x_1 and x_2. The last column holds the class label y.

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

b. Logistic regression classifier with linear decision boundary

Implement a classifier based on a logistic regression approach and with a linear decision boundary :

$$h_\theta(\mathbf{x}) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

a) Implement a sigmoid function $g(z) = \frac{1}{1+e^{-z}}$. Use numpy to compute the exp so that your function can take numpy arrays as input. Plot the sigmoid function.

b) Implement the hypothesis function $h_\theta(\mathbf{x})$

c) Implement the objective function $J(\theta)$:

$$J(\theta) = \frac{1}{N}\sum_{n=1}^{N} y_n \log h_\theta(\mathbf{x_n}) + (1 - y_n)\log(1 - h_\theta(\mathbf{x_n}))$$

d) In a similar way as in PW02 and PW03, implement the gradient ascent with the update rule :

$$\theta_i \leftarrow \theta_i + \alpha \frac{1}{N}\sum_{n=1}^{N}(y_n - h_\theta(\mathbf{x}_n))x_{n,i}$$

e) Test your implementation by observing the evolutions of the objective function $J(\theta)$ during the gradient ascent.

f) Compute the correct classification rate on `ex2-data-test.csv` after convergence assuming you have an estimator of the posterior probabilities with

$$P(y_n = 1|\mathbf{x_n}; \theta) = h_\theta(\mathbf{x_n})$$
$$P(y_n = 0|\mathbf{x_n}; \theta) = 1 - h_\theta(\mathbf{x_n})$$

g) Draw the decision boundary of your system on top of the scatter plot of the testing data.

h) Compare the performance of the logistic regression system with the ones of previous's week.

c. Logistic regression classifier with non-linear decision boundary

Redo the experiments of 2.b by increasing the complexity of the model in order to have a non-linear decision boundary :

$$h_\theta(\mathbf{x}) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 + \ldots)$$