ASI assessed exercise 2017/2018

2nd May 2018

Introduction and Instructions

In this work you will analyze the Fashion MNIST and CIFAR10 datasets available to download from:

https://www.kaggle.com/zalando-research/fashionmnist/data

https://www.cs.toronto.edu/~kriz/cifar.html

Listed below are various exercises to undertake. Note that in each case you should implement the algorithms yourselves - you may not use existing implementations (specifically, for this exercise you are not allowed to use any off-the-shelf implementation of the Naïve Bayes classifier) - and should submit all of your code.

Note that you are not allowed to work in groups for this assessed exercise - each student is required to submit her/his own work having worked on the exercise individually

Submission

You are free to use any programming language of your choosing but it is your responsibility to ensure that we can run your code. We recommend you use either Matlab or Python. Please submit either:

- Your code (including instructions for running - there should be one script that answers all the questions) and a .pdf report documenting your answers to the exercises.

- Or (preferably) a single iPython notebook that we can run. If you take this route, please *also* submit a .pdf output of the script (print the html to pdf). Your notebook should include any text descriptions required in the answers. (iPythons markdown cells allow you to add text)

Please submit your work through the submission system at: http://bigfoot-m1.eurecom.fr/teachingsub

The deadline is Wednesday 30th May 2018 at 4:00pm.

Exercises

Note (code) and (text) before each task indicate whether the corresponding part involves coding or writing.

1. (code) Download the Fashion MNIST and CIFAR10 datasets and import them. [3]

2. (text) Comment on the distribution of class labels and the dimensionality of the input and how these may affect the analysis. [7]

3. Classification

    a) (code) Implement the Naïve Bayes classifier. [10]

    b) (text) Describe a positive and a negative feature of the classifier for these tasks [5]

    c) (text) Describe any data pre-processing that you suggest for this data and your classifier [5]

    d) (code) Apply your classifier to the two given datasets. Make sure your optimization is clearly commented. Use classification accuracy and test log-likelihood as your figures of merit [15]

    e) (code) Display the confusion matrix on the test data [5]

    f) (text) Discuss the performance, compare them against a classifier that outputs random class labels, and suggest ways in which performance could be improved [5]

4. Linear regression

    a) (code) Implement Bayesian linear regression (you should already have an implementation from the lab sessions) [10]

    b) (code) Treat class labels as continuous and apply regression to the training data. [15]

    c) (code) Produce a scatter plot showing the predictions versus the true targets for the test set and compute the mean squared error on the test set [5]

    d) (text) Suggest a way to discretize predictions and display the confusion matrix on the test data and report accuracy [5]

    e) (text) Discuss regression performance with respect to classification performance [5]

    f) (text) Describe one limitation of using regression for this particular task. [5]

5. Bonus question

    a) (text / code) The state-of-the-art in these image classification problems suggests that convolutional layers in convolutional neural networks yield most of the improvements compared to standard neural networks. The reason is that they are capable of modeling spatial patterns through the hierarchical analysis of patches of images. Propose and implement ways to exploit patch information in the Naïve Bayes classifier or linear regression. A couple of suggestions are: (i) apply Naïve Bayes classification to the output of convolutional layer in the LeNet architecture (ii) construct

the Naïve Bayes classifier by calculating patch-specific statistics and extend this by stacking multiple of these

Numbers at the end of each section are the number of marks available.

Be concise - a complete solution should be around 10 pages (including figures) and no more than 20.