Final Project – Basic Probability, Computing and Statistics 2016

Fall 2016, Master of Logic, University of Amsterdam

Instructors: Christian Schaffner and Bas Cornellisen

Submission deadline: Friday, December 23rd, 8 p.m.

1 Project Description

In this project you will perform your own data analysis. You will program an algorithm to analyse a data set and then summarise your results in a short report. Each group (of 2 or 3 people) will be assigned a project supervisor. The role of the project supervisor is to help you assess whether the chosen combination of algorithm and data set is appropriate. He can also help you with conceptual questions about the algorithm. Finally, your supervisor will proof-read your report before submission and make suggestions for improvements. The supervisor is **not** there to help you program!

The project is divided into stages. These should serve as a guideline for your own organisation. How you proceed through each stage and how much time you invest for each stage is up to you (but feel free to consult with your supervisor).

- **Stage 1:** Pick a data set and algorithm that you want to work with. Clearly lay out what you want to achieve and what your evaluation measure is. For example, in a classification task, you goal might be to achieve good classification results and your evaluation measure is the percentage of correctly predicted items (this measure is also known as *accuracy*). Approximate Time Frame: Week 6.

- **Stage 2:** Implement your algorithm and assess its performance. Once you have the algorithm running, try to improve it. For example, you can try to construct new features when using Naïve Bayes. Your final report should include a comparison between the baseline version of your algorithm and all improvements that you made. Approximate Time Frame: Week 7 + beginning of Week 8

- **Stage 3:** Write your report. Check the style and structure with your supervisor. Approximate Time Frame: Week 8

2 The Algorithms

In general, there are two broad classes of algorithms: regression and classification algorithms. The boundary is not always clear, but regression algorithms predict continuous values (such as the price of a house) while classification algorithms predict discrete values (such as whether or not a e-mail is spam). For the purpose of this project you are restricted to using 4 algorithms.

- **EM** EM is actually a general strategy for learning latent variable models. We have already used it for mixture modelling. You can again implement an EM algorithm for a mixture model. If you are daring you can also make the mixture components depend on one another. There are many ways to do this. The one we allow for here is known as a Hidden Markov Model.

- **Naïve Bayes** We have already experimented with NB for classification. You can again implement NB for this project. Extensions to NB include conditioning on two or more instead of only one variable.

- **Linear Regression** As the name suggests, this is a classical regression model. It assumes that each data point x_i is drawn from a Gaussian Distribution with mean parameter μ_i and variance parameter σ^2. Since the data point and its mean share the same index, the data points are independent but **not** identically distributed (each data point comes from a Gaussian with a different mean). The mean parameter is assumed to be a linear combination of the data point's features $f(x_i)$ and a global weight vector w. This means that $\mu_i = w^\top f(x_i)$. While the theory may seem a bit involved, linear regression models can be learned with relative ease using the method of least squares.

- **Feed-forward Neural Network** Neural networks have recently enjoyed a revival in the machine learning community. They are layered learning algorithms where the value of each layer is a (possibly transformed) linear combination of the previous layer. Using the chain rule for derivatives, neural networks can be trained with an algorithm know as backpropagation. While this algorithm may be a bit harder to understand than the other ones, implementing it is actually rather straightforward.

If you want to use linear regression or a feed-forward neural net, we highly recommend that you have a look at the relevant videos of Andrew Ng's famous machine learning course. While he does not give you all the mathematical details, he explains these algorithms and their implementation with great care.

3 The Data Sets

You can pick any data set that you fancy from the UCI Machine Learning Repository. Notice that your supervisor has to approve of your choice. You are only allowed to chose data sets for either a regression or classification task. Your data set may not include any missing values.

4 The Report

You will write a report of at most 4 pages + references. All reports must be created using the EACL-2017 LaTeX template and follow the instructions provided therein. Your report must contain the following sections:

- **Introduction:** Motivate your choice of data set and algorithm and lay out the goals of your analysis.
- **Background:** Describe the prediction and learning algorithms that you used theoretically. Also describe the assumptions that you make about the (in)dependencies and distributions of the variables in your model.
- **Experiments:** Shortly describe the specifics of the data set (size, number of classes etc.) and then report your results for the baseline model and possible extensions. Your results need to be supported by tables and/or graphs.
- **Conclusion:** Describe what you learned from your analysis, either about the algorithm or the data set. Also explain whether or not your expectations from the Introduction were met.

5 Deliverables

In order to complete this project you need to submit a .zip file containing:

- **Report:** The report described above
- **Code:** Any code that you have written. This code should be executable either from the command line or from Pycharm (or both)
- **Readme:** A text file describing the contents of the folder. This file should also contain a link to the data set which you used. Most importantly it should contain very detailed instructions on how to run your code!