Data Science: Principles and Practice

Final assignment B, 2019/2020, take-home assessment

Description

Final Assignment B is based primarily on the material in lectures 7–9, and in the accompanying walkthroughs, but it also draws on material covered in the earlier part of the course.

You should write a practical report of not more than 1500 words excluding tables, graphs and images. The mark breakdown for the course is: practicals (20%), Final Assignment A (50%), Final Assignment B (30%). The deadline for submitting work, on Moodle, is 5pm on 7 February 2020. Please submit three files: your report (in pdf, with your name and crsid on the front page), a cover sheet (available on Moodle), and a zip file with your code. Assessors may run your code, but you will not be assessed on code quality. Nor will you be assessed on how your quantitative results compare to others'. The assessment will be based on the report itself and on clarity of description of the work you did, the evaluations you chose, and the insights you explain.

Dataset

You will work with the the full DIABETES 130-US HOSPITALS FOR YEARS 1999–2008 dataset, diabetic_data_original.csv, which was provided with Final Assignment A.

For the purposes of this assignment, we wish to predict the time_in_hospital value. It doesn't make sense to predict the duration of a stay in hospital by using attributes recorded *during that stay*. Instead, we will restrict attention to follow-on visits, and seek to predict the length of a stay for follow-on visit on the basis of attributes recorded in the previous visit.

Your task

Your task is to implement a neural network for predicting time_in_hospital, and to investigate the impact on training time of variations in the neural network architecture. Your implementation and report should include these considerations:

- *Data preparation.* Describe briefly how you prepared the data. Don't explain the code you used, but do explain what filtering or processing you applied, and report how many items there are in your dataset.
- *Loss function.* The time_in_hospital value is an integer. You should choose an appropriate probability model, e.g. the Poisson distribution, and derive the corresponding loss function. *[Hint. As a sanity check, you may like to check your answer against a simple mean-square-error loss function. This is just to help debugging; you don't need to include it in your report.]*
- *Machine learning algorithms implementation.* There are many variations one might investigate. For the purposes of this project, you should report on one of the following:
  - Does it speed training if we pre-process the data using PCA, and use PCA features as inputs to the neural network rather than raw data? The hope is that it saves time if the neural network doesn't have to work as hard to learn a useful representation.
  - Does it speed training to use multi-task learning? In multi-task learning, we train the neural network to predict several values at once, e.g. let the network have two outputs, predictions for both time_in_hospital and readmitted, and train using a loss function that measures the accuracy of both of these predictions. Evaluation should be based only on the prediction of time_in_hospital. The hope is that the extra tasks will give the network more feedback information at each training step, which will help it learn faster.

  You are welcome to explore other variations, but you are only required to report on one of the two described above.

- *Measurement.* Your task is to investigate training time for a neural network. Note that if we want to compare neural networks that have different numbers of parameters, it doesn't make sense to measure training time in epochs, since a neural network with more parameters will take more computational effort per epoch. We might measure training time in clock time (seconds), or in number of parameters times epochs.

- *Evaluation.* We wish to learn *generalizable* results, i.e. to find a predictor which works well in novel scenarios. It's easy to get unduly optimistic results when one shuffles a single dataset then splits it into training and holdout parts; ideally we would split the dataset into training and holdout parts along some meaningful division, e.g. use data from 100 hospitals for training and the remaining 30 hospitals for evaluation.

  This dataset does not include any attributes that suggest themselves for this purpose. Instead, use t-SNE to split the data into qualititively different training and holdout parts. In this way, we'll get a reasonable indication of what prediction accuracy might be in novel scenarios.