CSM6420/SEM6420 – Machine Learning for Intelligent Systems

Second Assignment, March 2020

This is the second of two assignments for CSM6420/SEM6420, and comprises 60% *of the total marks for the module. It will be assessed according to the Department's assessment criteria for essays (see online Appendix AC of Student Handbook), as well as the practical workshop sign-off. In particular, the marks for the report will take account of understanding of the problem, challenge of the chosen task, completion of the task, accuracy of the prediction on the test set and quality of the presentation. Other marks will cover knowledge of the literature, justification of the approaches taken, quality of analysis and amount of work involved.*

Please submit your work through Blackboard before 5pm, Thursday, 21th May 2020

---

1 Dataset

This assignment consists of two components. One component is the practical workshops worth of 8% of the marks (1% for each signed-off worksheet). The main component of this assignment, carrying 92% of the marks, is based on the ECG dataset derived from the PhysioNet Computing in Cardiology Challenge 2017 [1], for detecting cardiac arrhythmia (irregular heartbeat) using short single lead ECG (electrocardiogram) recordings.

There are two type of data files for training and testing. The files ending with 'signal.csv' store the ECG recording in fixed length of 20 seconds (corresponding to 6000 time points). The files ending with 'feat.csv' store the features extracted from the ECG waveform using various signal processing algorithms. Variable "Type" in the training data indicates the type of ECG segment:

"N" - normal sinus rhythm, "A" - atrial fibrillation (AF),

"O" - other cardiac rhythms, "∼" - noise segment.

Note that there exist some missing values both in the training and test data because: (1) some ECG recording segments are shorter than 20 sec (less than 6000 time points); (2) the feature extraction algorithms failed in generating valid features in some cases.

Your task is to develop machine learning models using the training data to classify the type of the ECG segments based on the given ECG recordings and the extracted features.

More information regarding the original datasets and the feature extraction process can be found in the papers below:

Gari Clifford, Chengyu Liu, et al.. AF Classification from a Short Single Lead ECG Recording: the Physionet Computing in Cardiology Challenge 2017. DOI: 10.22489/CinC.2017.065-469.

Shreyasi Datta, Chetanya Puri, et al.. Identifying Normal, AF and other Abnormal ECG Rhythms using a Cascaded Binary Classifier. DOI: 10.22489/CinC.2017.173-154.

2 Assignment specification

For this assignment, you are to investigate the performance of at least two types of classifiers: one with a feature-based approach using the given features as input only, and one with an end-to-end (deep) learning approach using the ECG recordings as input only. You should then write a report on this, taking into account the guidelines given below. The datasets described above are available on Blackboard and the Kaggle inclass competition.

You can use Python, R, or WEKA for this task, although Python is preferable. Various free cloud-based computing resources with access to GPU are available, for example Kaggle kernels, Google Colaboratory, and virtual machines in Google Cloud Platform (with free educational credit and guidance provided on Blackboard).

[1] https://physionet.org/challenge/2017/

- The following list describes the different elements of your report and how they contribute to the total marks.

  (1) Describe the data exploration you have performed, discuss any observations you have about the data and what data preprocessing and/or dimensionality reduction you have performed and why. (10%)

  (2) Describe the classifiers you have selected, explaining how they work and discussing the reasons behind why you chose them. Note that you have to develop at least one model using a feature-based approach and another using an end-to-end (deep) learning approach. (10%)

  (3) Discuss and investigate the options you chose for configuration, i.e. the hyper-parameter settings and choice of model architectures. The default settings may not be the optimal ones for many classifiers. (12%)

  (4) Describe the experiments performed and discuss the results. Issues you might like to consider: the impact of data preprocessing, dimensionality reduction or model architecture, how the classifier performances compare, etc. (30%)

  (5) Build one classifier with the learning methods and possible hyper-parameters that you select based on your experiments and analysis, then use this classifier to predict the class labels for the given 4000 test cases, and submit the prediction results in a separate csv file to the kaggle inclass competition (see www.kaggle.com/c/ml2020-assignment-2/) for development validation and final evaluation. The submission csv file should consist of two columns with the header of "ID" and "Predicted" (coded in 'N', 'A', 'O', '∼'). You receive marks according to the average F1 score of the three classes 'N', 'A' and 'O' for a test sample of 2800 cases, e.g. marks of 5.5 for a score of 0.55, and 10 for a score of 0.95 or above. (10%).

     Note: You should submit your predictions on all the 4000 test cases by participating the kaggle inclass competition , where you can check your performance (the average F1 score for three classes) of a test sample of 1200 cases on the public leaderboard, although the noisy recordings of class ' ' will be ignored in evaluation for final ranking of this competition. The private leaderboard results will be based on the remaining test cases and will not be available till the end of the competition.

  (6) To make your work reproducible, please provide in appendix the relevant Python/R code/notebook or Weka configurations, whichever appropriate for building the final prediction model. This should cover important steps such as data preprocessing, cross-validation for associated hyper-parameter tuning and model training. Also please indicate the version of python, R or Weka and relevant libraries/packages that you've used for this work. (10%)

  (7) The quality of the report will be marked as 10%, assessing the structure and presentation, including writing style, citation and formatting issues.

- You should aim to keep your answers concise, while conveying the important information. Graphs and/or tables should be included where appropriate to present the results of your comparisons and experiments. Between 3000-4000 words (excluding references and appendices) might be appropriate for this report. A report which is not in .pdf format or whose length is larger than 4500 words may lead to reduced marks in the "quality of report".

Plagiarism: One of the dangers of this assignment is the temptation to use paragraphs from web documents or papers that you have read. Please resist this temptation and do not do it. Otherwise, you will be heavily penalised. The report should be in your own words. If it is appropriate and absolutely necessary to include sentences and materials from elsewhere, then they should be clearly indicated as quotes, and references should be cited.

Please do not show your report to any other students.

Chuan Lu (cul), 30th March 2019