COMP9418 Assignment 2

Advanced Topics in Statistical Machine Learning, 17s2, UNSW Sydney

Last Update: Wednesday 4th October, 2017 at 15:45

Submission deadline: Tuesday October 24th, 2017 at 23:59:59

Late Submission Policy: 20% marks will be deducted from the total for each day late, up to a total of four days. If five or more days late, a zero mark will be given.

Form of Submission: You should submit your solution with the following files:

1. `solution.pdf`: Technical report;
2. `solution.py`: Python code; and
3. `predictions.txt`: Model's prediction on the test data.

No other formats will be accepted. There is a maximum file size cap of 5MB so make sure your submission does not exceed this size.

Submit your files using give. On a CSE Linux machine, type the following on the command-line:
`$ give cs9418 ass2 solution.pdf solution.py predictions.txt`
Alternative, you can submit your solution via the course website
`https://webcms3.cse.unsw.edu.au/COMP9418/17s2/resources/12704`

Please note that this is a group assignment. See §6 below for details.

Recall the guidance regarding plagiarism in the course introduction: this applies to this homework and if evidence of plagiarism is detected it may result in penalties ranging from loss of marks to suspension.

[100 Marks] Probabilistic Dynamical Models

In this assignment you will make use of the `Character Trajectories` dataset. This dataset was created for a PhD study regarding primitive extraction with sequential models. It consists of labelled samples of pen pin trajectories recorded while writing individual characters. It can be downloaded from: `https://archive.ics.uci.edu/ml/datasets/Character+Trajectories`. You can also read more about this application in Ben Williams' PhD thesis: `https://www.era.lib.ed.ac.uk/handle/1842/3221`. However, you will be provided with all the necessary information to carry out this assignment.

1 Data

You are given the Matlab data file `trajectories_train.mat`, which contains the variables `xtrain`, `ytrain`, and `key`, as well as the file `trajectories_xtest.mat`, which contains the variable `xtest`. The `x` variables contain 3-dimensional representations of trajectories for 20 characters, where the first two dimensions are the velocities on the horizontal and vertical axes and the third dimension is the pen tip force. These variables are cell arrays where each component is a $D \times T$ matrix with $D = 3$ and T being the length of the sequence. `xtrain` contains 1429 training examples and `xtest` contains `1429` test examples. The `y` variables contain the class numbers which refer to the corresponding labels given by the variable `key`.

2 Main Task: Classifying Characters Using Their Trajectories

Your task is to build a *probabilistic* classifier for predicting the class probabilities of new characters described by their trajectories. You are required to submit (i) a technical report describing your solution; (ii) the code used for your solution; and (iii) the predicted class log probabilities for each of the test datapoints in `xtest`.

3 Technical Report: `solution.pdf`

In this report you will describe your solution to the problem above. The maximum length of the report is 4 pages excluding references and appendix. Keep in mind that your assessor reserves the right to read your appendix. The report must contain only the following sections:

Abstract [$\approx$ 1 paragraph] A short summary of your approach and the results of your method.

Introduction [$\approx$ 1/2 page] An introduction to the problem, the basic approach you have taken and your contributions.

Model [$\approx$ 1/2 – 1 page] A mathematical and conceptual description of your model.

Inference [$\approx$ 1 page] A description of how your inference method works. For example, posterior inference (if applicable) and how predictions are done.

Parameter Estimation [$\approx$ 1/2 page] A description of how parameter estimation is carried out (if applicable). Examples of this can be cross-validation, MAP, MLE or full Bayesian inference.

Results [$\approx$ 1 page] Here you need to describe an evaluation methodology that convinces the reader that your approach is sound. For this, you need to split your training set into training and validation and show performance metrics with respect to a sensible baseline that uses a softmax classifier. The performance metrics that you need to report are the error rate (ER) and the mean negative log probability (MNLP) defined as:

$$\text{ER} = 1 - \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(y^{(i)}_{\text{pred}} = y^{(i)}), \tag{1}$$

$$\text{MNLP} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} \mathbb{I}(y^{(i)} = j) \log p_{\text{pred}}(y^{(i)}|\mathbf{x}^{(i)}), \tag{2}$$

where $\mathbb{I}(\cdot)$ is an indicator function that is 1 if the condition inside the brackets is satisfied and 0 otherwise; $y^{(i)}_{\text{pred}}$ is your model's prediction and $y^{(i)}$ is the true label for datapoint $\mathbf{x}^{(i)}$ on the validation set; $\log p_{\text{pred}}(y^{(i)}|\mathbf{x}^{(i)})$ is your model's predicted log probability on class $y^{(i)}$ for datapoint $\mathbf{x}^{(i)}$; and N is the number of datapoints on the validation set. In addition to the performance metrics above, you can report other analysis/insights of the data or results in this section.

References A full list of previous work that you used or is relevant to your report.

Appendix Additional material such as derivations or extra analysis of results.

The length of each section is provided as a guide only and you may deviate from this as long as you do not exceed the page limit of the report.

4 Code: `solution.py`

This is a Python file that implements your solution. It must be well-documented and self-contained and able to generate your predictions in the next section. The program should load the file `trajectories_xtest.mat` from the current directory and its output should be the file `predictions.txt`.

5 Predictions: `predictions.txt`

This file should contain the log probabilities $\log p(y^{(i)} = j|\mathbf{x}^{(i)})$ for all the datapoints in `xtest` in the same order. On each line it must contain comma-separated log probabilities for all classes $j = 1, \ldots, 20$.

6 Group Submission

This is a group assignment with the minimum group size of 2 people and a maximum of 3 people. It can be submitted from one of the group members' account. Authorship should be stated in the technical report `solution.pdf` and the code `solution.py`.

7 Hard Constraints

While you have freedom on the method, inference machinery and coding techniques you use, these are the constraints your submission must satisfy. Failure to meet these requirements will yield an overall mark of zero.

(i) The maximum length of your technical report `solution.pdf` is 4 pages excluding references and appendix (for which there is no limit).

(ii) The minimum font size of your technical report `solution.pdf` is 11pt.

(iii) Your solution must use at least one of the techniques explained in the course. It can be an extension of one of the methods described in the lectures. However, methods that are completely unrelated to the course material are not acceptable.

- An example of a method that will not be accepted is a standard neural network trained using back-propagation.

- A variation of this method that will be accepted is a Bayesian neural network trained using variational inference.

If in doubt, please contact the course lecturer.

(iv) Your code `solution.py` must be executable on a standard architecture (Linux and Mac OS) and if non-standard Python packages (i.e. packages that are not hosted on PyPI) are required it must advise the user to install them.

(v) The prediction file `predictions.txt` must be in the format specified in §5.

(vi) Although you are given the test features `xtest` in the file `trajectories_xtest.mat` for making predictions, under absolutely no circumstances they may be used during training.

(vii) Only group submissions of 2 or 3 people are accepted.

8 Assessment

Your submission will be assessed based on the quality of the technical report and the performance of your predictions. Note that, although the code does not have a specific weight in the assessment, penalties will be applied for unsuitable documentation, unreproducible results or failure to execute (with the latter yielding an overall mark of zero). This is a breakdown of the marks:

- [**50 Marks, technical report**] The techical report must satisfy the constraints above and the marks will take into account the following criteria:
  - [10 Marks] Overall clarity of presentation. This includes clarity, formatting, organisation, language use, correct spelling and grammar. Note that rambling or waffling to fill space unnecessarily will be penalised. Your report may well be under 4 pages if it is sufficiently clear and descriptive.
  - [30 Marks] Technical description of your solution (sections Model, Inference, and Parameter Estimation of your report). This includes clarity, technical difficulty and innovation.
  - [10 Marks] Sound evaluation of your technique (section Results of your report). This includes presentation and analysis of the results.
  - A well-written appendix that expands on the technical description of your solution or on the analysis of the results may increase your overall mark. However, as stated above, your asessor reserves the right to read the appendix in detail.
- [**50 Marks, predictive performance**] Predictive performance on the test data will be evaluated using the error rate (ER) and the mean negative log probability (MNLP) as defined in Equations (1) and (2) respectively.