ELL784/AIP701: Assignment 1

Last updated: 01-09-2022, 11:52am

Deadlines

1. Deadline for final submission of assignment report and all code: **21 September 2022, 11:59 PM**.

Instructions

1. While you are free to discuss all aspects of the assignment with your classmates or others, your code (for those registered for AIP701) and your results and report must be entirely your own. We will be checking for any copying, and this will be treated as plagiarism and dealt with accordingly. In case of any doubts in this regard, please ask us.
2. **For AIP701: You are free to use any platform to write and test your code. However, if you are using Python, please make sure you submit .py files for evaluation during the viva**.
3. All students should prepare a report, compiling all your results and your interpretation of them, along with your overall conclusions. In particular, you should attempt to answer all of the questions posed in the respective parts of the assignment below. Any graphs or other visualisations should also be included therein.
4. A single tar/zip file containing the source code (if applicable), report, and any other relevant files has to be submitted via Moodle. The zip/tar file should be structured as per the below guidelines:
    - Upon deflating all submission files should be under a directory with the student's entry number. E.g., if a student's entry number is `20XXCSXX999` then the tarball/zip submission should be named 20XXCSXX999.zip or equivalent, and upon deflating **all contained files** should be under a directory named ./20XXCSXX999 only.
5. The schedule for demos/vivas will be announced by your respective TAs, in advance. If for any reason you cannot attend in your scheduled slot, you must arrange for an alternative slot with your TA well in advance. Last-minute requests for rescheduling will normally not be accepted.

Weightage

For ELL784: The mandatory part of this assignment is worth 6 marks or 6% of your overall course grade; a part-wise break-up is provided below. In addition, there is an optional extra-credit portion worth 2 marks.
For AIP701: The coding part of this assignment will additionally be worth 5 marks, which will amount to 25% of your grade for that course.

1 Part 1A (ELL784: 3 marks; AIP701: 5 marks)

The objective here is to implement and/or utilise the concepts of regression learnt in class via polynomial curve fitting. To recap, polynomial curve fitting is an example of regression. In regression, the objective is to learn a function that maps an input variable x to a continuous target variable t.

For the first part of this assignment, we provide two personalised input files (for training and testing) that contain data of the form:

$$x_1, t_1$$
$$x_2, t_2$$
$$.$$
$$.$$
$$.$$
$$x_N, t_N$$

The relation between x and t is of the form

$$t = w_0 + w_1 x + ... + w_M x^M + \epsilon$$

where the noise ϵ is drawn from a normal distribution with mean 0 and unknown (but fixed, for a given file) variance. M is also unknown. You should download your files from `https://web.iitd.ac.in/~sumeet/A1/<EntryID>/gaussian/train.csv` and `https://web.iitd.ac.in/~sumeet/A1/<EntryID>/gaussian/test.csv` (for example, `https://web.iitd.ac.in/~sumeet/A1/2017CRZ8638/gaussian/train.csv`). The end goal is to identify the underlying polynomial (both the degree and the coefficients), as well as to obtain an estimate of the noise variance.

The tasks to be accomplished are:

- **(AIP701 only)** *Code implementation:* You may use a programming language of your choice, though Python is recommended. Naturally, your code should not make use of any machine learning libraries but should implement the method from scratch. By default, the implementation should carry out least-squares regression. The minimisation of the error function should be attempted in two ways: by directly computing the Moore-Penrose pseduoinverse (pinv), and via gradient descent (gd). For the latter, you should have a parameter controlling the *batch size*: ranging from 1 (stochastic gradient descent) to N (full batch gradient descent).
- **(non-AIP701)**: Identify a standard machine learning library (for example, scikit-learn in Python) and familiarise yourself with how it can be used for least-squares regression.
- To begin with, use only the first 20 data points in your training file. (Note that you will need to generate the design matrix by creating feature vectors containing the powers of x from 1 to M.)
- Solve the curve fitting regression problem using error function minimisation – try both the analytic and gradient descent approaches, and comment on any variation in the results (for both training and testing sets), especially with change in batch size for gradient descent. Also explore how the convergence of gradient descent varies with the stopping criterion (you could plot the loss as a function of the number of iterations).
- **(AIP701 only)** You can define your own error function other than sum-of-squares error (note that the error function need not be convex). Try different error formulations and report the results.
- Try and use a validation approach (within your training set; the test set should never be touched during validation) to characterise the goodness of fit for polynomials of different order. Can you distinguish overfitting, underfitting, and the best fit? In addition to this, obtain an estimate for the noise variance.
- Introduce regularisation and observe the changes. For quadratic regularisation, can you obtain an estimate of the optimal value for the regularisation parameter λ? What is your corresponding best guess for the underlying polynomial? And the noise variance?

- Now repeat all of the above using the full training set of 70 points. How are your results affected by adding more data? Comment on the differences.
- At the end: what is your final estimate of the true underlying polynomial? Why?

1.1 Part 1B (ELL784 *extra credit*: 2 marks)

You are provided with a second personalised data set, available at https://web.iitd.ac.in/~sumeet/A1/<EntryID>/non_gaussian/train.csv and https://web.iitd.ac.in/~sumeet/A1/<EntryID>/non_gaussian/test.csv. It is generated from a different polynomial, and the noise is now non-Gaussian. Can you repeat the analysis for this data set, focusing in particular on characterising the noise – can you figure out what kind of noise it is? Please justify your answer using appropriate analysis.

1.2 Evaluation criteria

- **(AIP701 only)** Correctness of your code (see the example specifications at the end of the document for a more precise idea of the functionality that is expected).
- Visualise the data and results in meaningful ways. [Gnuplot is a good tool for this.]
- How close did you get to the actual answer? (Which we know, and will tell you during your viva!) If there is a big discrepancy, why might this have happened?
- How well you understood what you are demonstrating and the concepts involved.
- How extensively you have played around with various parameters, and your analysis of the variations in the consequent results.
- The insights that you have gained from your experiments.

2 Part 2 (ELL784: 3 marks)

For this part, you will be trying to model a real-world time-series data set. This contains measurements of a certain quantity at different points in time. (Details of what that quantity is will be revealed later.) The provided data sets should be downloaded from https://web.iitd.ac.in/~sumeet/A1/train.csv and https://web.iitd.ac.in/~sumeet/A1/test.csv. The train set contains 110 time points; and the test set contains another 10 points for which the measured value has been removed. In each row, the first value is the date of the measurement (in US format, so month comes before day), and the second value is the actual measurement. Your task is to train a linear regression model (using appropriate basis functions) which can predict the missing values on the test set as accurately as possible.

You are allowed to use only linear regression models for this task. Cross-validation, hyperparameter tuning and regularisation are encouraged to produce better results.

2.1 Evaluation criteria

Evaluation on this task will be based on conceptual clarity, interpretation of results and reproducibility. Your understanding of the algorithm, and the extent of validation and tuning will also be evaluated.

Additionally, your predictions for the 10 test points are to be submitted to a real-time leaderboard (link to be shared later), where you will be ranked on your performance on that test set (in terms of minimising mean-squared error).

Example code specifications (AIP701 only)

The below gives an indication of the kind of options or command-line flags your code should allow for. While you don't have to follow exactly this format, your code is expected to have similar functionality or flexibility.

```
sh run.sh --method gd --batch_size 10 --lamb 0.02 --X file.csv --polynomial
2
```

OUTPUT

```
weights=[0.95666953 0.79969864 -0.48338126]
```

```
sh run.sh --method pinv --X file.csv --polynomial 2
```

OUTPUT

```
weights=[0.95666953 0.79969864 -0.48338126]
```

`--method`	Method to minimise error [pinv | gd].
`--batch_size`	Batch size to use.
`--lamb`	Regularisation strength (λ).
`--X`	Complete file location.
`--polynomial`	Degree of polynomial to fit.