CS4200/CS5200 On-line Machine Learning Coursework 2

Yuri Kalnishkan

February 27, 2022

This assignment must be submitted by the **9th of March, 10am**.

Feedback will be provided on the 25th of March.

The total number of point on the paper is 50. This coursework contributes 15% to the final mark so the coursework result will be scaled accordingly.

The **CS4200 students** do not need to do questions marked as (*) and their total is 40 marks. (You are still welcome to do these questions if you want to get feedback, but they will not be counted for the grade.)

Learning outcomes assessed

- understand on-line learning set-up;
- demonstrate advanced knowledge of methods of on-line learning;
- analyse the properties of on-line learning algorithms; develop and apply performance bounds;
- apply on-line algorithms to real-world data;
- implement machine learning algorithms in MATLAB or Python or R.

Instructions

The coursework consists of two parts:

1. practical part: implementing algorithms and applying them to data;
2. theoretical part: some theoretical problems.

Submission should be done electronically on the CS4200/CS5200 moodle page.

The programming part can be done using any of the following languages: MATLAB or Python or R. You programs should work with the version of MATLAB (or R or Python) presently installed on the `linux.cim.rhul.ac.uk` server.

If you want to use a different language, get a permission from the course lecturer.

You should submit files with the following names:

- `AA_Brier.m` (or `AA_Brier.py` or `AA_Brier.R`) should contain the source code you used for the Aggregating Algorithm;
- `AA_test.m` (or `AA_test.py` or `AA_test.R`) should contain the code loading the data from file, applying AA and calculating what is required in Section 1.2; in MATLAB is does not have to be a function and can be a script;
- if you use Python, you can submit a Jupyter notebook `AA.ipynb` with a function named `AA_Brier` and code for applying AA to the data;
- `results.txt` should contain all the values which you are asked to find and all the comments you would like to add; use the template from the course web site and fill in the answers where question marks are (do not change the layout of the file because it will be processed by a script);
- `report.pdf` (or one of the formats `doc`, `docx`, `rtf`) should contain the pictures you need to produce for the AA example question;
- `theory.pdf` (or one of the formats `doc`, `docx`, `rtf`, `txt`, `ipynb`; `jpg` scans or photos of handwritten notes are also acceptable) should contain the answers to the theoretical problems. It is the student's responsibility to check that the file to be submitted can be opened. The mark will be set to 0 if the file is either in a wrong format or not readable. It is your responsibility to ensure that the scan or photo is legible.

NOTE: The coursework assignment is strictly individual. **Plagiarism** and **collusion** will be prosecuted. Coursework submissions are routinely checked for plagiarism. Note that using someone else's code with altered variable names is still a case of plagiarism and is not acceptable.

Marking Criteria

In the practical part, full marks are awarded only when the code is correct and applied to the input data in a correct way. The results should be close to the correct values.

In the theoretical part, full marks are awarded only when it is clear how you have arrived at the final answer. Simply giving a result without any justification is unlikely to get you full marks.

1 Practical Part

1.1 Aggregating Algorithm

Implement a function calculating predictions output by the Aggregating Algorithms with equal weights in the binary square loss (Brier) game.

```
function predictions = AA_Brier(expertsPredictions,outcomes)

% AA(expertsPredictions,outcomes) calculates predictions
% produced by the AA given expertsPredictions (this is a matrix
% whose lines are sequences of predictions) and outcomes.
```

or

```
def AA_Brier(expertsPredictions,outcomes):

    # AA(expertsPredictions,outcomes) calculates predictions
    # produced by the AA given expertsPredictions (this is a
    # matrix whose lines are sequences of predictions) and
    # outcomes.
```

If the index count starts from 1, then the element `expertsPredictions(n,t)` is the prediction produced by expert E_n on step t. The element `predictions(t)` should be the prediction output by the AA on step t.

If the index count starts from 0, then the the prediction produced by expert E_n on step t is `expertsPredictions(n-1,t-1)`. The element `predictions(t-1)` should be the prediction output by the AA on step t.

In your program use the optimal $\eta = 2$ and equal initial weights. Your program should work with the square loss; you do not need to make provisions to expand it to other loss functions (and doing so may be rather difficult).

[10 marks]

1.2 AA Example

This real-data example about bookmakers' odds is taken from V. Vovk and F. Zhdanov, Prediction with expert advice for the Brier game, in: *Proceedings of the Twenty Fifth International Conference on Machine Learning*, pages 1104-1111. New York: ACM Press, 2008. See http://vovk.net/ICML2008/ for more details.

Bookmakers quote betting odds for a sports event. One may interpret those as probabilities that one side will win. (In fact the situation is a bit more complicated: bookmakers are interested in making money rather than accurate predictions so they always include a margin to guarantee themselves a profit. However we ignore that and interpret their odds as a prediction.) The file `tennis1.txt` contains predictions output by four bookmakers. You can refer to the paper to see how the predictions were calculated from the odds quoted by bookmakers.

The format of the file is as follows. Each line describes a game. The first number is an ID with the technical information about the game; you may ignore it. The second number shows whether player 1 won; it is always 1 because players were re-ordered this way retrospectively. The third column shows whether player 2 won; it is always 0 for the same reason. In your calculations assume that the outcome is always 1 (of course this only makes sense because we apply the AA retrospectively).

The rest of the line contains "predictions" made by four bookmakers. The fourth column is the estimate done by the first bookmaker whether player 1 would win; the fifth is the estimate by the first bookmaker that player 2 would win. They sum up to 1. Ignore the fifth number and treat the fourth as prediction in the square loss game. The sixth and seventh columns contain predictions by the second bookmaker, the eighth and ninth by the third, and the tenth and eleventh by the fourth. (Clearly you do not need odd columns and the second is not very informative either).

Process this file and run your function on the predictions by four bookmakers to get AA predictions.

Calculate the total loss of the AA and quote it in your report and in the `results.txt` file.

Make three graphs and include them into `report.pdf`.

- The first should show five cumulative losses vs. time, $\mathrm{Loss}_{E_1}(t)$, $\mathrm{Loss}_{E_2}(t)$, $\mathrm{Loss}_{E_3}(t)$, $\mathrm{Loss}_{E_4}(t)$ of the experts and $\mathrm{Loss}_L(t)$ of the AA.
- The second should show the following four differences vs. time: $\mathrm{Loss}_{E_1}(t) - \mathrm{Loss}_L(t)$, $\mathrm{Loss}_{E_2}(t) - \mathrm{Loss}_L(t)$, $\mathrm{Loss}_{E_3}(t) - \mathrm{Loss}_L(t)$, $\mathrm{Loss}_{E_4}(t) - \mathrm{Loss}_L(t)$.
- The third should show the following four differences vs. time: $\mathrm{Loss}_{E_1}(t) - \mathrm{Loss}_{\mathrm{ave}}(t)$, $\mathrm{Loss}_{E_2}(t) - \mathrm{Loss}_{\mathrm{ave}}(t)$, $\mathrm{Loss}_{E_3}(t) - \mathrm{Loss}_{\mathrm{ave}}(t)$, $\mathrm{Loss}_{E_4}(t) - \mathrm{Loss}_{\mathrm{ave}}(t)$, where $\mathrm{Loss}_{\mathrm{ave}}(t)$ is the loss of the following simple strategy. On every steps it takes the predictions of experts E_1, E_2, E_3, and E_4 and averages them with equal weights $1/4$:

$$\gamma_t = \frac{1}{4}(\gamma_t^1 + \gamma_t^2 + \gamma_t^3 + \gamma_t^4) \ .$$

Include these graphs in your report.

[10 marks]

2 Theoretical Problems

You may use MATLAB or Python to get a hint or to verify your result, but you should submit calculations worked out on paper.

1. Consider the simple prediction game with the outcome and prediction spaces $\{0, 1\}$. The following table shows the predictions produced by four experts and outcomes over five steps:

Step	1	2	3	4	5
E_1	0	0	1	0	1
E_2	1	1	0	0	1
E_3	0	0	0	0	0
E_4	0	0	0	1	0
Outcome	0	0	1	0	1

Work out the predictions made by the halving algorithm with access to these experts observing these outcomes. [5 marks]

2. (*) Consider the simple prediction game with the outcome space $\{0, 1\}$. The following table shows the weights experts have before trial t and predictions they output on trial t:

	weight	prediction
E_1	0.2	0
E_2	0.4	0
E_3	0.3	1
E_4	0.1	1

   (a) Calculate the prediction made by the Weighted Majority algorithm on trial t. [2 marks]

   (b) Suppose that the outcome 1 occurs on trial t. Update the weights assuming $\beta = 0.7$ and normalise them. [3 marks]

3. If the game is η-mixable, the loss of a learner using the Fixed Share algorithm is bounded by

$$\mathrm{Loss}_L(T) \leq$$
$$\mathrm{Loss}_{\mathrm{superexpert}}(T) + \frac{1}{\eta}\left(\ln N + k \ln \frac{N-1}{\alpha} + (T-1-k)\ln\frac{1}{1-\alpha}\right) \ ,$$

   where k is the number of switches the superexpert has made.

   Alice says that in practice Fixed Share should always be used instead of the Aggregating Algorithm. The reason is that a superexpert with 0 switches is the same as a base expert and therefore Fixed Share can compete with base experts too. Is this argument correct? Justify your answer. [6 marks]

4. (*) Calculate all predictions made by the exponential smoothing on the series $2, 4, 5$; calculate the prediction for the fourth element of the series. In your calculation assume the smoothing parameter $\alpha = 0.7$ and use $\ell_0 = 3$ to predict the first element. [5 marks]

5. For each of the following ARMA models write down its order (p, q) and find out whether it specifies a weakly stationary series and whether it specifies a causal weakly stationary series. Justify your answers. (Here Z_t denotes a white noise process (i.i.d. with zero mean and constant variance).)

(a) $X_t = Z_t - 2Z_{t-1}$; [3 marks]

(b) $X_t = -0.5X_{t-1} + Z_t + 3Z_{t-1}$; [3 marks]

(c) $X_t = 6X_{t-1} - 9X_{t-2} + Z_t - Z_{t-1} + Z_{t-2}$. [3 marks]

Total marks: 50.