CS4200/CS5200 On-line Machine Learning Coursework 1

Yuri Kalnishkan

February 4, 2022

This assignment must be submitted by the **14th of February, 4pm**.

Feedback will be provided on the 1st of March.

The total number of point on the paper is 50. This coursework contributes 10% 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 MAT-LAB (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:

- `beta_dynamic.m` (or `beta_dynamic.py` if you use Python or `beta_dynamic.r` if you use R) should contain the source code you used for calculating betas;

- `gamma_dynamic.m` (or `.py` or `.R`) should contain the source code you used for calculating gammas;

- if you use Python, you can submit a Jupyter notebook `dynamic.ipynb` with the functions named `beta_dynamic` and `gamma_dynamic` instead;

- `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 moodle page 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);

- `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. It is your responsibility to ensure that the scan or photo is legible. The mark will be set to 0 if the file is either in a wrong format or not readable.

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.

You must not use existing library functions for calculating alphas, betas, gammas, or Viterbi maximum posterior probability path.

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 Beta

Implement a dynamic programming function calculating coefficients beta for a hidden Markov model. See Slide 28 of Class 2 for a definition.

Use the following header (or a close equivalent if you are using R).

```
function b = beta_dynamic(M,p,B,v)

% BETA_DYNAMIC(M,p,B,v) calculates the matrix of betas for the hmm with
% transition matrix M, emission matrix B, and initial probabilities
% p, given the observations v
```

or

```
def beta_dynamic(M,p,B,v):

    # BETA_DYNAMIC(M,p,B,v) calculates the matrix of betas for
    # the hmm with transition matrix M, emission matrix B, and
    # initial probabilities p, given the observations v
```

This function will output the $N \times T$ matrix of betas.

[10 marks]

1.2 Gamma

Implement a function calculating coefficients gamma for a hidden Markov model. See Slide 29 of Class 2 for a definition.

Use the following header (or a close equivalent if you are using R).

```
function g = gamma_dynamic(alpha,beta)

% GAMMA_DYNAMIC(alpha,beta) calculate gamma for hmm given alpha and beta
```

or

```
def gamma_dynamic(alpha,beta):

    # GAMMA_DYNAMIC(alpha,beta) calculate gamma for hmm given
    # alpha and beta
```

The header is written in this way so that one could run the function as follows: `g = gamma_dynamic(alpha_dynamic(M,p,B,v),beta_dynamic(M,p,B,v))`.

This function will output the $N \times T$ matrix of gammas.

[10 marks]

1.3 HMM Example

The files `M.txt`, `B.txt`, and `p.txt` contain the transition matrix, the emission matrix, and the initial probabilities for a hmm with 15 hidden states and 10

3

observable states. The file `v.txt` contains a sequence of 20 observed states (it is assumed that the states are numbered from 1, not from 0). You can read the files using `M = dlmread('M.txt')` in Matlab or `M = np.loadtxt("M.txt")` in Python. For `v` in Python you will need to explicitly convert the numbers to integers: `v = np.loadtxt("v.txt").astype(int)`.

Run your functions on this data and calculate the matrices of betas and gammas. Get the values $\beta_{18}(10)$ and $\gamma_{10}(11)$. (Note that in this notation indices start from 1.) Enter the values on the template `results.txt` where question marks are and include the file in your submission.

[4 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. The following weather observations were made over twelve consecutive days: sunny, sunny, cloudy, rain, rain, sunny, sunny, cloudy, rain, rain, cloudy, rain.

   (a) Assuming that this sequence was generated by a first-order Markov chain with three states, "sunny", "cloudy", and "rain", find the maximum likelihood estimate of the transition probabilities.   [4 marks]

   (b) Use the transition probabilities to work out the probabilities of the thirteenth day being sunny, cloudy, and rainy (you need to calculate three probabilities).   [2 marks]

2. (*) Consider a hidden Markov model with two hidden states s_1 and s_2 and three visible states x_1, x_2 and x_3. Let the transition matrix be

$$M = \begin{pmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{pmatrix} \ ,$$

the emission matrix be

$$B = \begin{pmatrix} 0.3 & 0.4 & 0.3 \\ 0.2 & 0.3 & 0.5 \end{pmatrix} \ ,$$

and the matrix of initial probabilities be

$$p = \begin{pmatrix} 0.4 \\ 0.6 \end{pmatrix} \ .$$

Suppose that a sequence of visible states x_2, x_3, x_1 has been observed. Calculate:

   (a) the coefficients α for time 3;   [4 marks]

4

(b) the probabilities of the hidden state at time 3 being s_1 and s_2 conditional on the visible states observed; [3 marks]

(c) the probabilities of the hidden state at time 4 being s_1 or s_2 conditional on the visible states observed. [3 marks]

3. Consider a hidden Markov model with two hidden states s_1 and s_2 and three visible states x_1, x_2, and x_3. Let the transition matrix be

$$\begin{pmatrix} 0.3 & 0.7 \\ 0.8 & 0.2 \end{pmatrix},$$

emission matrix be

$$\begin{pmatrix} 0.3 & 0.4 & 0.3 \\ 0.5 & 0.3 & 0.2 \end{pmatrix},$$

and initial probabilities be $p = (0.4, 0.6)$.

(a) Calculate the steady state of the hidden Markov chain. Show your calculations. [5 marks]

(b) Calculate the limit of the distribution of the emission variable X_t in this hidden Markov model as $t \to +\infty$. Justify your answer. [5 marks]

Total marks: 50.