CSE472 (Machine Learning Sessional)
Assignment 2: Hidden Markov Models (HMM)

Introduction

In this assignment, you will learn modeling with a Hidden Markov Model (HMM), and implement the Viterbi and the Baum-Welch algorithm. HMMs are applied in many fields including speech and image processing, bioinformatics and finance.

Climate pattern modeling

El Niño–Southern Oscillation (ENSO) is an irregular periodic variation in winds and sea surface temperatures over the tropical eastern Pacific Ocean, affecting the climate of much of the tropics and subtropics. The warming phase of the sea temperature is known as El Niño and the cooling phase as La Niña both of which have long-term persistence. El Niño years in a particular basin tend to be wetter than La Niña years.

We can observe whether or not it is an El Niño year based on rainfall in the tropical Pacific for the present. But we are interested in understanding past climate variation using tree ring widths. We can infer from the tree ring data (with some error) what the total precipitation might have been in each year of the tree's life.

So, we have two hidden states representing El Niño and La Niña. The observed quantities are rainfall estimates (from tree ring width) for the past T years. Let's assume for simplicity that our observations Y_t can be modeled by Gaussian distributions i.e. $f(Y_t|X_t = 1) \sim N(\mu_1, \sigma_1^2)$ and $f(Y_t|X_t = 2) \sim N(\mu_2, \sigma_2^2)$.

For more details, please see `https://waterprogramming.wordpress.com/2018/07/03/fitting-hidden-markov-models-part-i-background-and-methods/`

Dataset

You will be given two files titled "data.txt" and "parameters.txt" containing the rainfall data for the past T years and parameters for the HMM respectively.

Input

- The “data.txt” file contains T rows each containing a number indicating the rainfall for that year.

- The “parameters.txt” file contains the number of states n (2 in this case) in the first line. The next n lines provides the transition matrix P. The next line gives the means of the n Gaussian distributions and the last line lists the standard deviations of the n Gaussian distributions.

Note: Your implementations must be easy to extend to arbitrary number of states ~~of~~ and emission probability distributions.

Output

- A file containing estimated states using the parameters provided in “parameters.txt”. This will contain T rows each containing the estimated state for that year.

- A file containing parameters learned using the Baum-Welch algorithm. The format will be the same as “parameters.txt”. Add the stationary distribution in the last line.

- A file containing estimated states using the learned parameters. This will contain T rows each containing the estimated state for that year.

Viterbi algorithm implementation

In this case, you will be given

1. The parameters of the HMM i.e. the transition matrix P, the initial probabilities of the states π, and the parameters for the Gaussian distributions $\mu_1, \sigma_1, \mu_2, \sigma_2$ in the “parameters.txt” file. (Use the stationary distribution of P as the initial probabilities. `https://www.stat.berkeley.edu/~mgoldman/Section0220.pdf`)

2. The rainfall estimates $y_1, y_2, \dots y_T$ in “data.txt”

Your task is to

- Implement the Viterbi algorithm to estimate the most likely hidden state sequence $x_1, x_2, \dots x_T$ (El Niño or La Niña) for the past T years.

Baum-Welch implementation

Now we will also estimate the parameters of the HMM. Given,

- The rainfall estimates $y_1, y_2, \dots y_T$ in "data.txt"

You will implement the Baum-Welch algorithm to estimate

1. The most likely values of parameters of the HMM i.e. the transition matrix P, and the parameters for the Gaussian distributions $\mu_1, \sigma_1, \mu_2, \sigma_2$. You can assume the initial probabilities π to be the stationary distribution of P.

2. The most likely hidden state sequence $x_1, x_2, \dots x_T$ (El Niño or La Niña) for the past T years for the finally estimated parameters.

The Baum–Welch algorithm is a special case of the Expectation-Maximization (EM) algorithm used to find the unknown parameters of a hidden Markov model (HMM). Expectation-Maximization is a two-step process for maximum likelihood estimation when the likelihood function cannot be computed directly, for example, because its observations are hidden as in an HMM.

For this, initialize the parameters $P, \mu_1, \sigma_1, \mu_2, \sigma_2$ with random values (save the seed). Then iterate the following two until convergence:

1. **E-step:** Use the forward-backward equations to find the expected hidden states given the observed data and the set of current parameter values

2. **M-step:** This is the update phase. In this step, find the parameter values that best fit the expected hidden states given the observed data.

Note: You can use the values provided in "parameters.txt" for initialization if there are convergence problems.

Outputs

You will output the estimated hidden state sequence and parameter values to files. The details will be provided later. Also, compare the solution results with the sci-kit hmmlearn.

Submission

1. Upload the codes in Moodle within **10:00 P.M. of 8th January, 2022 (Saturday)**. (Strict deadline)

2. Write code in a single *.py file, then rename it with your student id. For example, if your student id is 1605123, then your code file name should be "1605123.py" and the report name should be "1605123.pdf".

3. Finally make a main folder, put the code and report in it, and rename the main folder as your student id. Then zip it and upload it.

Evaluation

1. You have to reproduce your experiments during in-lab evaluation. Keep everything ready to minimize delay.

2. You are likely to give online tasks during evaluation which will require you to modify your code.

3. You will be tested on your understanding through viva-voce.

4. If evaluators like performance, efficiency or modularity of a particular code, they can give bonus marks. This will be completely at the discretion of evaluators.

5. You are encouraged to bring your computer in the sessional to avoid any hassle. But in that case, ensure an internet connection as you have to instantly download your code from the Moodle and show it.

Warning

1. Don't copy! We regularly use copy checkers.

2. First time copier and copyee will receive negative marking because of dishonesty. Their default is bigger than those who will not submit.

3. Repeated occurrence will lead severe departmental action and jeopardize your academic career. We expect fairness and honesty from you. Don't disappoint us!