

**School of Computer Science  
The University of Adelaide  
Artificial Intelligence  
Assignment 3  
Semester 1 2018  
due 11:55pm 6th June 2018**

May 21, 2018

---

## 1 Introduction

In this assignment, you will implement an approximate inference engine for the Hidden Markov Model shown in Figure 1 (which is similar to the one introduced in Lecture 21).

The HMM depicts the scenario shown in Lecture 21. In this model, there are a sequence of latent(unobserved) random variables  $\{R_0, R_1, \dots, R_T\}$  which represent if it is raining outside and a sequence of observed random variables (evidence)  $\{U_0, U_1, \dots, U_T\}$  indicating whether the director arriving with an

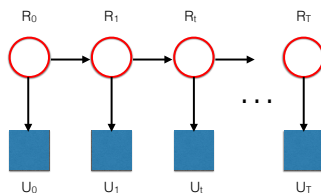


Figure 1: The hidden Markov model.

umbrella. Here, we use  $R_t = \text{True}$  denotes raining and  $U_t = \text{True}$  represents that an umbrella is observed. We define the following probabilities:

$$\begin{aligned} P(R_0 = \text{True}) &= 0.2 \\ P(R_t = \text{True} | R_{t-1} = \text{True}) &= 0.7 \\ P(R_t = \text{True} | R_{t-1} = \text{False}) &= 0.3 \\ P(U_t = \text{True} | R_t = \text{True}) &= 0.9 \\ P(U_t = \text{True} | R_t = \text{False}) &= 0.2. \end{aligned}$$

We are interested in estimating the distribution of  $P(R_T | U_0, U_1, \dots, U_T)$ . To calculate this distribution, you are required to use **both** likelihood weighted sampling method and Gibbs sampling method to perform the approximate inference.

## 2 Deliverables

You can write your program in C/C++ or Python. Your program takes the a txt file (observation.txt) as the input. The text file contains a sequence of binary numbers (separating by white spaces), e.g. 0 1 1 0. This sequence corresponds to the observed variable sequence  $\{U_0, U_1, \dots, U_T\}$  with 0 indicating False and 1 indicating True. You can infer the sequence length  $T$  by the sequence of binary numbers.

If you write your code in C/C++, you must supply a Makefile with a rule called inference to compile your program into a Linux executable binary named inference.bin. Your program must be able to be compiled and run as follows:

```
$ make inference
$ ./inference.bin observation.txt
```

In the case of python, your program must be run as follows:

```
$ python inference.py observation.txt
```

The result of an inference method returned by your program must be written as two decimal values (corresponding to the values of  $P(R_t = \text{True} | \dots)$  and  $P(R_t = \text{False} | \dots)$ ) onto the standard output stream, separated by white space and followed by the name of inference method, For example:

```
0.873 0.127 Likelihood
0.872 0.128 Gibbs
```

## 3 Submission and Assessment

Two example input files, observation1.txt and observation2.txt will be supplied for the test of your code.

You must submit, by the due date, two files:

- files containing your code with necessary dependencies.

- file with a short written report detailing your implementation in no more than 1 page. A figure showing the relationship between the estimated posterior probability and the number of iterations is needed for the two methods. For example, you can vary the iteration number from 100 to 20000 and plot the estimated posterior probabilities (for the HMM in `observation2.txt`) for each iteration number.

The implementation of likelihood weighted sampling methods takes up 40 marks and the implementation of Gibbs sampling takes up 60 marks. For each part, your work will be assessed on the basis of quality of code(40%), quality of write-up (30%) and accuracy of results (30%).