COMP9418 Assignment2 report

Student Name：Xuetong Wang   Student zID: z5431400
Student Name：Dixuan Liu   Student zID: z5232340

Algorithms used:

Hidden Markov model.

HMM is an effective tool for dealing with time series data. It assumes that the states of the system are invisible, but that each state produces some observable output. The HMM models this process through state transfer probabilities, observation probabilities, and initial state probabilities.

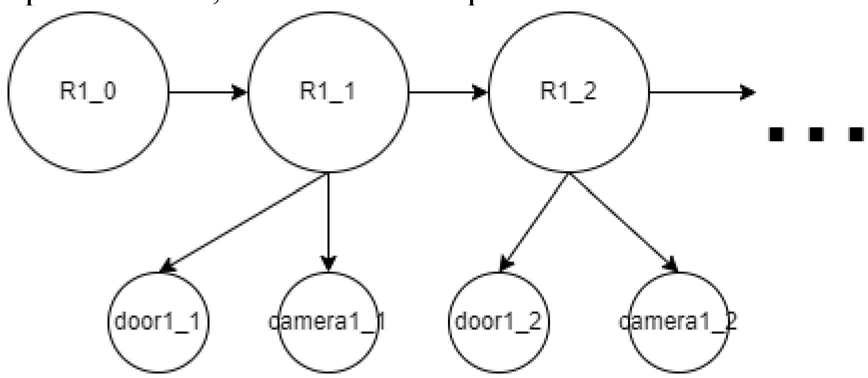

Figure 1: HMM model of r1

We consider whether the light is on or not in each room (i.e., whether someone is present or not) as the hidden state, and infer the state transfer probability by approximating the time before and after the historical data, and the data from various detection chips as the observation probability. We also use the historical data to derive.

In order to simplify the model and make it more efficient, we partition the rooms into separate HMMs. For the initial P(X0), we set the probability of all rooms except r3 to 0.5 for lights on, 0.5 for lights off, and the probability of r3 to 1 for lights on, 1 for lights off. and store the probability values in the dictionary distribution dict. This is done by looking at the datasets data1.csv and data2.csv, where the probability of r3 being occupied is very high at the initial time. And in conjunction with the fact that everyone entering the building for work at 8am needs to have been in r3.

For P(Xt) the calculation is.

P(Xt = 'on') = P(Xt-1 = 'on') * P(Xt = 'on' | Xt-1 = 'on ') * P(et | Xt = 'on') + P(Xt-1 = 'off') * P(Xt = 'on' | Xt-1 = 'off') * P(et | Xt = 'on')
P(Xt = 'off') = P(Xt-1 = 'on') * P(Xt = 'on' | Xt-1 = 'off ') * P(et | Xt = 'off') + P(Xt-1 = 'off') * P(Xt = 'off' | Xt- 1 = 'off') * P(et | Xt = 'off')
Normalize P(Xt = 'on') and P(Xt = 'off') and store them in the dictionary distribution_dict to be used as P(Xt-1) in the next round.
Compare P(Xt = 'on') and P(Xt = 'off'), if P(Xt = 'on') is greater than P(Xt = 'off'), the room lighting condition is recorded as on and vice versa.

For the algorithm implementation we used the classical forward algorithm.

Time complexity:
O(N^2T), where N is the number of states(the number of rooms) and T (2400 time steps) is the length of the observation sequence. This is because for each point in time in the sequence, the algorithm needs to consider a transfer from each state to another state.

Methodological justification:

The main reason for choosing HMM is its efficient processing capability for time-dependent data. In smart building simulations, the lighting state of a room is not only dependent on the current environmental conditions, but is also influenced by the past state. the HMM is able to capture this time-series dependency.

We transform the reliability of a monitoring device into a probability, and try to achieve a realistic prediction through the interaction of different monitoring sensors and temporal transfer probabilities.

We try to use Bayesian network to model the room state without considering the time-varying permutations, and we only care about the state of each monitoring sensor at the current time step to evaluate the state of the room.
After verification, the results are not satisfactory. The effect of the previous time step on the current time step cannot be ignored. At the same time, due to the reliability of the monitoring devices, the monitoring sensors alone cannot accurately represent the state of the rooms at the current time.

We also tried to use deep learning to backpropagate by the number of people in the room, using gradient descent to learn the transfer probabilities. But we could not converge to a good result.

Assumptions:

Since we are only interested in the room with or without lights on, we don't care about the number of people in the room. So we can make the following simplifications to the training data:

1. change the number of people in r1-r10 to 'on', 'off', 'on' corresponds to the case where the number of people is greater than 0, 'off ' corresponds to cases where the number of people is 0.

2. change the number of people in camera1-camera4 to 'found' and 'not found', 'found' corresponds to cases where the number of people is greater than 0, 'not found' corresponds to cases where the number of people is 0. 'found' corresponds to cases where the number of people is greater than 0, and 'not found' corresponds to cases where the number of people is less than 0. 3.

3. change the number of people entering and leaving the door from door_sensor1 to door_sensor11 to 'not move', 'even' and 'odd'. '. 'not move' corresponds to the case where the door did not move, 'even' corresponds to the case where the number of people is even, and 'odd' corresponds to the case where the number of people is odd.

Here we assume that an odd number of counts occurring in the room door sensor means that the position of a person in both rooms has shifted, which causes a change in the number of people in the room, and there is a certain probability that the room will change from occupied to unoccupied, or from unoccupied to occupied.

The following table shows the OUTCOMES table for each factor:

room	('on', 'off')
door_sensor	('not move', 'even', 'odd')
camera	('found', 'not found')

These assumptions are justified by the fact that the two rooms are independent when the state of the door is observed. In the real world, most people have their own stable areas of activity during the course of the day. It is rare that there is a large movement of people in the space of 15s in such a house. Also, since we are only interested in the switching of the lights, we can see whether the room is occupied or not. So by simplifying the model in this way, the model is more efficient and easier to implement.

We used the code to determine the accuracy of the sensors in the data, with robot being the most accurate, representing the number of people in the room it's in.

Robot1 Accuracy:	1.0
Robot1 Mean Error:	0
Robot2 Accuracy:	1.0
Robot2 Mean Error:	0

Conclusion:

In this project, we use Hidden Markov Models (HMM) as the core algorithm to optimize the lighting system of rooms in smart buildings. By analyzing historical data and sensor readings, we construct a model that accurately predicts the occupancy status of a room and adjusts the lighting accordingly to maximize energy efficiency. Our methodology focuses on the ability to process time-series data, utilizing the state transfer probabilities and observation probabilities of the HMM to model changes in room lighting states.

To ensure efficiency our modelling assumptions simplify the facts of what really could happen and may fail to capture all the key dynamics in the actual environment. In future work, the introduction of more environmental variables and more complex state transfer logic can be considered. Since the total number of people is consistent with a normal distribution. The energy of the whole network can be guaranteed to flow within a certain range. The combination of Markov network and HMM can be considered to predict the flow of people in space and time.