Assignment 2[1]

Daniel Woolnough z5116128

The Problem

In the problem, we have several entities and relationships to deal with:

- A number of rooms and a number of sensors
- Spatial Relationships: between adjacent rooms, between a sensor and its room location, etc.
- Time Relationships: how a room evolves over time.

In the problem, we need to use these entities and their relationships to learn some parameters over our training data. We then use these parameters to make inferences on some test data, and to decide whether or not to turn a light on in a room. We can think of this as making a judgement about whether a room is occupied, or empty – we assume that the number of people in a room is irrelevant for this judgement[2].

Model Description

The data that we are given to make inferences is a set of observations from each of the sensors (which may or may not be reliable). We cannot directly observe whether a room is occupied or empty. We also need to be able to propagate our decisions over time. Thus, the best fit for this problem is a **Hidden Markov Model**, which we need to modify from the simple univariate case.

Define:

- $\boldsymbol{X}_t$**:** hidden variables representing the rooms at timestep t. Each variable $R_i^t \in \boldsymbol{X}_t$ represents room i at time t, and can take the value 0 ("empty") or 1 ("occupied").
- $\boldsymbol{E}_t$: emission variables representing the output of sensors. To simplify calculations, we will transform all emissions to be binary variables:
    - For reliable/unreliable sensors, 0 indicates "no motion" and 1 indicates "motion".
    - For doorway sensors, 0 correlates to 0, whilst 1 correlates to any non-zero number.
    - For robots, 0 indicates "incorrect", and 1 indicates "correct" (only with respect to occupancy, not the number of people present). We describe later how to deal with the fact that robots are room dependent.

To construct our graphical model:

---

[1] Since I was marked down in Assignment 1 for going over the word limit – when I didn't – I am including the final word count here: 1962 words.

[2] Throughout this report, all of our assumptions are written in red, for the convenience of the reader.

- At a given timestep t, we create an outgoing edge from each room to each sensor that can detect motion in that room. We assume that sensors can possibly see just beyond the room that they are fixed in, apart from doorways and robots.
- Between timesteps, we create an outgoing edge from each room at time t to each adjacent room (including itself) at time $t + 1$.

As a small example, consider the following floorplan for a simple building:

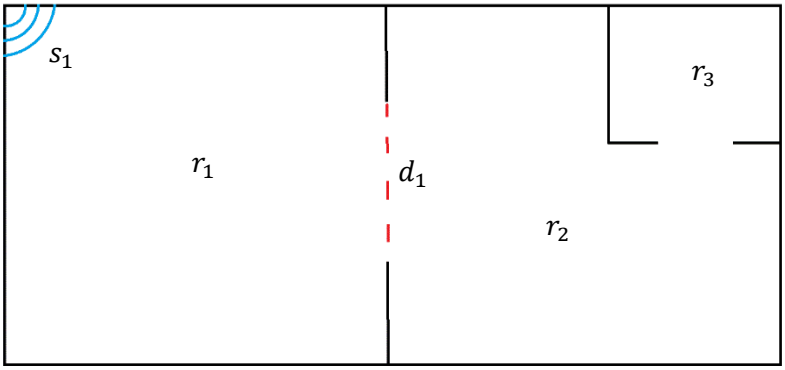


Figure 1: A basic building.

For a given timestep, our HMM would look like this:

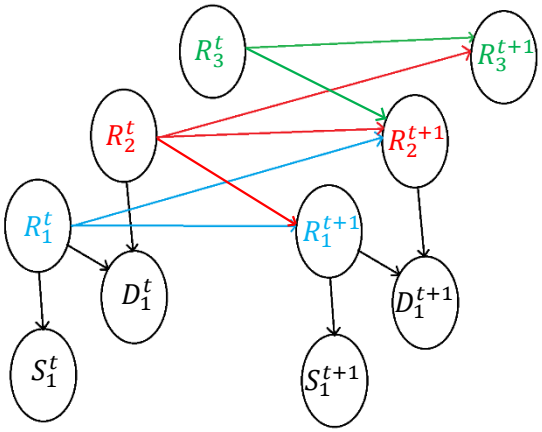


Figure 2: Time Evolution of the Building Network

This is a generalisation of the Hidden Markov Chain to a multivariate problem.

On the next page (Figure 3), we show our full HMM model for the building in the problem.

- We have not included time indices, for convenience.
- An undirected edge between two rooms Q and R indicates that there is an edge from Q^t to R^{t+1} and from R^t to Q^{t+1}. This relation is always symmetric by room adjacency.
- A directed edge is given from rooms to sensors.
- We use R for rooms, S for reliable sensors, U for unreliable sensors, D for door sensors, and O_1, O_2 for room o_1 and outside respectively. All variables in $\boldsymbol{X}$ are indicated in blue, and all variables in $\boldsymbol{E}$ are indicated in red.

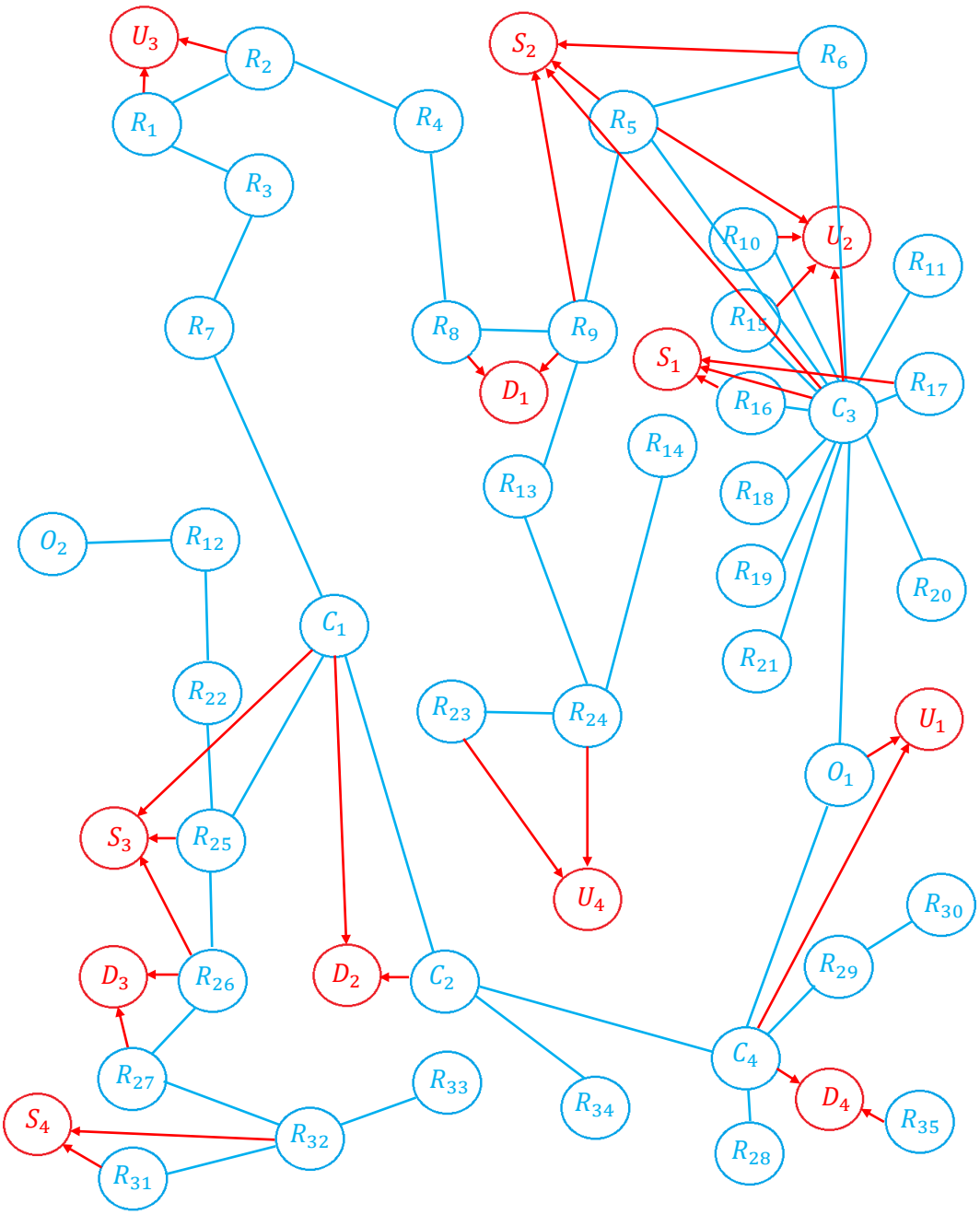


Figure 3: HMM for the complete building.

We also note that, in only adding an edge between adjacent rooms, we are assuming that only adjacent rooms affect each other; i.e. a person does not move more than one room away within any timestep.

Inference

We now proceed to generalise the mathematics of belief update in Hidden Markov Chains to the multivariate case.

For now, suppose that we have access to the following information (we describe later how these parameters were learnt):

- $B(R_i^0)$, that is, $P(R_i^0)$, which is the initial probability of the room being occupied before 8am.
- $P\left(R_i^t \middle| \text{Adj}(R_i, t)\right)$, where $\text{Adj}(R_i, t)$ is the set of variables representing the adjacent rooms to R_i, at time t (e.g. for R_1, we have $\text{Adj}(R_1, t) = \{R_1^t, R_2^t, R_3^t\}$).
- $P(\boldsymbol{e}_t | R_i^t)$, where $\boldsymbol{e}_t$ is an instantiation of sensor outputs (emission variables) at time t. We assume this is independent of the time t.

We also assume that:

1. The same assumptions as for a Hidden Markov Model hold; see slide 27 of lectures.
2. At time t, $R_i^t \perp R_j^t, j \neq i$. That is, no one changes rooms within a timestep. While this is not realistic, it is a reasonable assumption for our problem where we are using discrete timesteps, and do not care what happens in-between (any such activity can be modelled by an instantaneous change at the beginning of the next time step).

For a particular hidden variable R_i^t, we define our belief as

$$B(R_i^t) = P(R_i^t | \boldsymbol{e}_{1..t})$$

where $\boldsymbol{e}_{1..t}$ is over times 1 to t. Then, in the same manner as in the lectures, we have *Passage of Time*:

$$\begin{aligned} P(R_i^{t+1} \middle| \boldsymbol{e}_{1..t}) &= \sum_{\boldsymbol{r}^t \in \text{Adj}(R_i, t)} P(R_i^{t+1}, \boldsymbol{r}^t \middle| \boldsymbol{e}_{1..t}) \\ &= \sum_{\boldsymbol{r}^t \in \text{Adj}(R_i, t)} P(R_i^{t+1} \middle| \boldsymbol{r}^t, \boldsymbol{e}_{1..t}) P(\boldsymbol{r}^t | \boldsymbol{e}_{1..t}) \\ &= \sum_{\boldsymbol{r}^t \in \text{Adj}(R_i, t)} P(R_i^{t+1} \middle| \boldsymbol{r}^t) \prod_{Q^t \in \boldsymbol{r}^t} B(Q^t) \end{aligned}$$

where the last equality follows by assumption 2. We can then update our belief by *Observation*:

$$B(R_i^{t+1}) = P(R_i^{t+1} \middle| \boldsymbol{e}_{1..t+1}) \propto P(\boldsymbol{e}_{t+1} \middle| R_i^{t+1}) P(R_i^{t+1} \middle| \boldsymbol{e}_{1..t})$$

Some evidence will be independent of R_i^{t+1} - those that cannot observe that room. Denote by $\text{Sig}(E)$ the set of all room variables that have outgoing edges to E; then $E \perp \boldsymbol{X} - \text{Sig}(E)$, and by assumption 1 $E \perp \boldsymbol{E} - E$, and so

$$P(\boldsymbol{e}_{t+1} \middle| R_i^{t+1}) = \prod_{e \in \boldsymbol{e}_{t+1}} P(e \middle| R_i^{t+1}) = \left[\prod_{\substack{e \in \boldsymbol{e}_{t+1}: \\ R_i^{t+1} \in \text{Sig}(E)}} P(e \middle| R_i^{t+1}) \right] \left[\prod_{\substack{e \in \boldsymbol{e}_{t+1}: \\ R_i^{t+1} \notin \text{Sig}(E)}} P(e) \right]$$

which gives

$$B(R_i^{t+1}) \propto P(R_i^{t+1} \middle| \boldsymbol{e}_{1..t}) \prod_{\substack{e \in \boldsymbol{e}_{t+1}: \\ R_i^{t+1} \in \text{Sig}(E)}} P(e \middle| R_i^{t+1})$$

This is our belief in the occupancy of a room, expressed as a factor.

Cost Minimisation

To make a decision about which lights are turned on, we choose a threshold T; if our belief that a room is occupied is greater than T, we turn the light on; else, we turn it off. To define T, we use the following heuristic:

$$Light(R_i^t) = \begin{cases} on, & B(R_i^t) > \dfrac{c}{c+4} \\ off, & \text{otherwise} \end{cases}$$

where c is the price of electricity at timestep t.

We can prove that this value is optimal with respect to minimising the worst-case cost. For example, for $c = 1$, we plot the worst-case cost for different thresholds.

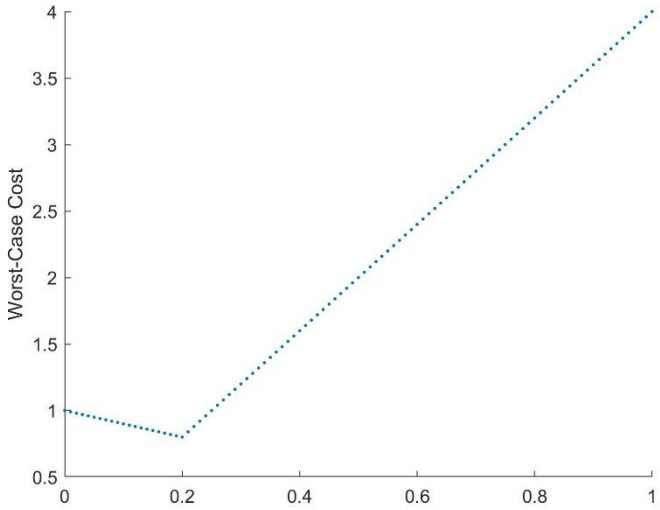


Figure 4: worst-case cost, for different thresholds.

We see that a minimum is achieved at $0.2 = \frac{1}{1+4}$.

Incorporating Robot Data

To handle robots, we only consider its observation after applying our *Passage of Time* and *Observation* for the other sensors first. Since a robot can only be in one room, we assume that its observation can only affect the room in which it is in. We can use the same principle as for *Observation*:

$$P\left(R_i^{t+1} \middle| \boldsymbol{e}_{1\ldots t+1}, robot_j\right) \propto P\left(robot_j \middle| R_i^{t+1}\right) B(R_i^{t+1})$$

Where R_i^{t+1} is the room that the robot is in at time $t+1$. We only apply this updating rule to that room at this timestep. We are assuming that the robot can accurately know where it is.

In order to estimate the required $P\left(robot_j \middle| R_i^{t+1}\right)$, we will make another assumption: the probability of a robot returning the correct measurement of occupied or empty is independent of both the room it is in, and the time at which it gives the output. That is,

$$P\left(robot_j \middle| R_i^{t+1}\right) = P\left(robot_j\right), \qquad j = 1,2$$

Estimating the Parameters from the Data

To use our inference system, we need to have learn some prior knowledge from the training data. Please note all of these are *factors*, not just single probabilities.

- $B(R_i^0), \forall i$: For this, we will take from the initial state of the data (a single row), the proportion A of rooms occupied at the initial state. Each room will then be given $P(R_i^t = 1) = A$.
- $P\left(R_i^{t+1} \middle| \text{Adj}(R_i, t)\right)$ is estimated from empirical counts, with smoothing ($\alpha = 1$).
- $P(E^t|R_i^t)$: likewise. We will need to assume that $P(E^t) = P(E)$ is independent of time (due to lack of training data).
- $P(robot_j)$ is also estimated from empirical counts, with smoothing, but only over the rooms in which the robot is present. In this manner, our assumptions about what it does and does not know is encoded into the probability of it being correct.

Algorithms and Complexity

Our inference method relies only on the algorithms *Passage of Time* and *Observation*.

For *Passage of Time*, we need to compute, for each room,

$$\sum_{\boldsymbol{r}^t \in \text{Adj}(R_i,t)} P(R_i^{t+1}|\boldsymbol{r}^t) \prod_{Q^t \in \boldsymbol{r}^t} B(Q^t)$$

To compute this efficiently, we begin with $P(R_i^{t+1}|\boldsymbol{r}^t)$, and iteratively:

- Join with $B(Q^t)$
- Sum out Q^t

This can be done since Q^t does not appear in any other belief factor. Since our variables are all binary, we see that the complexity of this algorithm is

$$O(2^{w_i})$$

where w_i is the width of the elimination order; that is, $w_i = |\text{Adj}(R_i, t)|$. In our model, the worst-case value for this number is for c_3, with has $w_{c3} = 13$. So at each timestep *Passage of Time* takes time

$$O(n \cdot 2^{w_{c3}})$$

where n is the number of rooms. We also note that our Join function has been implemented very efficiently, taking into account the fact that all variables are binary, and that the order of rooms in the factor $P(R_i^{t+1}|\boldsymbol{r}^t)$ is fixed.

For *Observation*, we need to compute, for each room,

$$P(R_i^{t+1}|\boldsymbol{e}_{1..t}) \prod_{\substack{e \in \boldsymbol{e}_{t+1}: \\ R_i^{t+1} \in \text{Sig}(E)}} P(e|R_i^{t+1})$$

and then normalize the resulting factor. Since we are dealing with instantiations, the "joins" here are simply scalar products (not factor products) and are constant time. The normalization process is then done in constant time, since the final product $P(R_i^{t+1}|\boldsymbol{e}_{1..t})$ always has only two rows (binary). In total this is $O(mn)$, where m is the number of sensors.

For the final inference (checking if the belief threshold has been exceeded), we need only perform a simple if/else for each room. This is $O(n)$.

Therefore, the total time complexity of our inference at one timestep is

$$O(n \cdot 2^{w_{c3}}) + O(mn) + O(n) = O(n \cdot 2^{w_{c3}})$$

since $m \ll n$.

Results

Our model has performed very well on all accounts. With further code optimizations and some changes to the base model (see below), our model is second on the leaderboard, and performing a day's inference in approximately 17 seconds. We have done this with relatively few assumptions and, when we have made those assumptions, they have been very general.

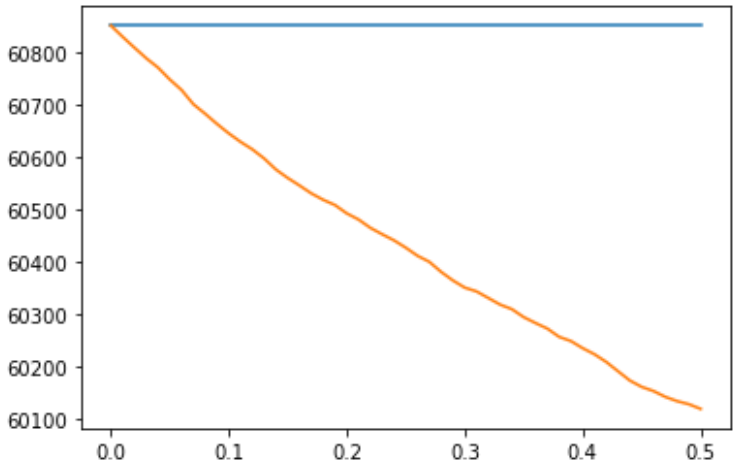
Changelog

Here, we give a list of changes that have been made since this original model was formulated and submitted to the first leaderboard.

- Friday 6/11: All possible optimizations in the code have been performed, and the model takes just over 2 minutes to run. For now, I am going to only update belief in the rooms, and ignore corridors, o_1 and outside. This will reduce largest treewidth down to 4. Model now runs in just under 10 seconds.
- Monday 9/11: To overcome previous problem, I have decided to split corridor c_3 in two, down the middle: c_{3a} and c_{3b}. This way, the largest treewidth is now 7, and I can re-incorporate all other rooms into the model. One day's inference now takes about 17 seconds. Final inference complexity: $O(n \cdot 2^{w_{c3a}})$.
- Monday 16/11: In an effort to reach the top of the leader-board, I have taken into consideration a comment on the Assignment spec page, and adjusted my Sig such that a sensor cannot see further than the room in which they are positioned. Cost has decreased by approximately 600.
- Thursday 19/11: I have been playing around with adjusting the threshold. I have found that, for very large electricity prices, if I increase the threshold then those large costs can decrease in the thousands! I have made the following adjustment:

$$Light(R_i^t) = \begin{cases} on, & B(R_i^t) > \dfrac{(1+a)c}{c+4} - \dfrac{1.2(1+a)}{1.2+4} + \dfrac{1.2}{1.2+4} \text{ and } c > 1.2 \\ on, & B(R_i^t) > \dfrac{c}{c+4} \text{ and } c \leq 1.2 \\ off, & otherwise \end{cases}$$

A plot of coefficient a vs mean cost (over 100 simulations) is given below. Note the original threshold corresponds to $a = 0$.



I have also noticed that, as a increases, solutions for smaller c destabilise – sometimes better, sometimes worse (though on average, about the same). To avoid any potential problems, I am clipping a at 0.5.