COMPSCI 270: Artificial Intelligence

Homework 5: Probabilistic Reasoning

(Due April 16 before the beginning of class)

Please read the rules for assignments on the course web page. Of course, **do not share code**. Acknowledge any help you received and any sources you used. If you are not sure whether you are allowed to use something, ask. If you want to use LaTeX that is great. If you write by hand and scan, please make sure that your handwriting is clear; illegible/ambiguous handwriting will receive no points. Please use Piazza for questions and Gradescope (Homework 5: Probabilistic Reasoning) to turn in the assignment.

1 Bayes Nets (50 points)

We are going to take the perspective of an instructor who wants to determine whether a student has understood the material, based on the exam score. Figure 1 gives a Bayes net for this.

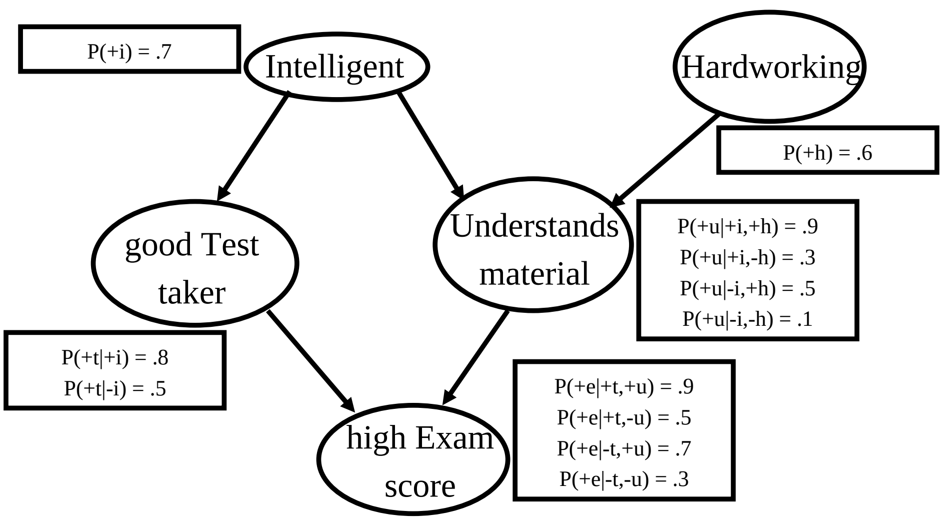


Figure 1: A Bayesian network representing what influences an exam score.

As you can see, whether the student scores high on the exam is influenced both by whether she is a good test taker, and whether she understood the material. Both of those, in turn, are influenced by whether she is intelligent; whether she understood the material is also influenced by whether she is a hard worker. (Disclaimer: the numbers and even the structure are completely made up and should not be taken to accurately reflect reality.)

1 (25 points).

Using variable elimination (by hand!), compute the probability that a student who did well on the test, and who is known to be a good test taker, actually understood the material. That is, compute $P(+u|+t,+e)$. Show all your work, including any intermediate factors that you produce.

2 (25 points.)

For the above Bayesian network, label the following statements about conditional independence as true or false. For this question, you should consider only the structure of the Bayesian network, not the specific probabilities. Explain each of your answers as follows: if you think the two variables are not conditionally independent, give *one* path between them that is not blocked. If you think the two variables are conditionally independent, point out, for *every* path between them, *where* that path is blocked.

1. T and U are independent.

2. T and U are conditionally independent given I, E, and H.

3. E and H are conditionally independent given U.

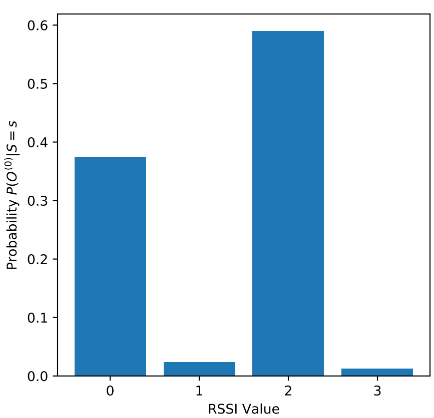
4. E and H are conditionally independent given U, I, and T.

5. I and H are conditionally independent given E.

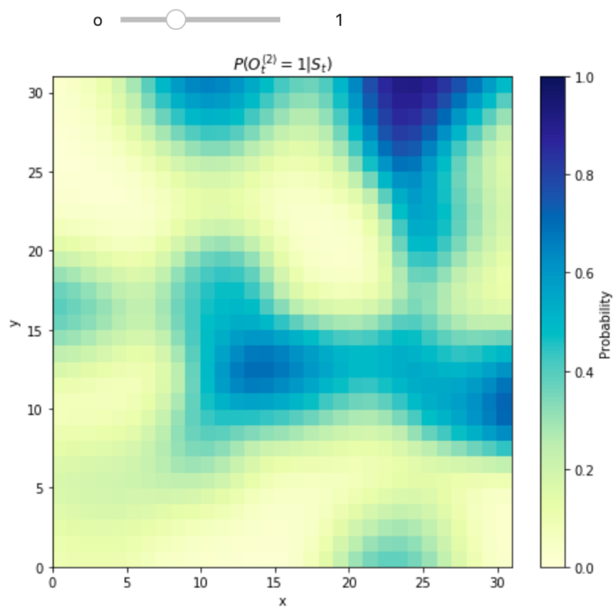
2 Hidden Markov Models (50 points)

2.1 Introduction

In this part, we will consider an indoor positioning system using hidden Markov models (HMMs). The aim here is to locate a mobile sensor receiving signals from beacons placed in a building or a room. In an open area, the signal strength of a message decreases as the distance between the transmitter and the receiver increases; therefore, the strength of the signal received can be directly used to estimate the position of the sensor. However, this is not the case in buildings. The same message can be received with different signal strengths due to reflection and scattering. One way to overcome this problem is to collect the signal strengths for each position and construct a probability distribution. This procedure is called *fingerprinting*. An example distribution for a particular position is shown in Figure 2a. We can estimate the position of the sensor using these probability distributions as our observation model. We also need to specify the transition probabilities to obtain a complete HMM. If there is no prior knowledge about the motion of the sensor, we can model the transitions as a random walk.



(a) The probability of observing a RSSI value for a particular position



(b) The probability of observing the RSSI value 1

Figure 2: An example observation model

2.2 Helper Code

In this assignment, you are given the transition and observation models, and you need to code up the algorithms that calculate the monitoring probabilities. The helper code uses the following Python libraries: `numpy`, `matplotlib` and `ipywidgets`. You can install them using `pip`:

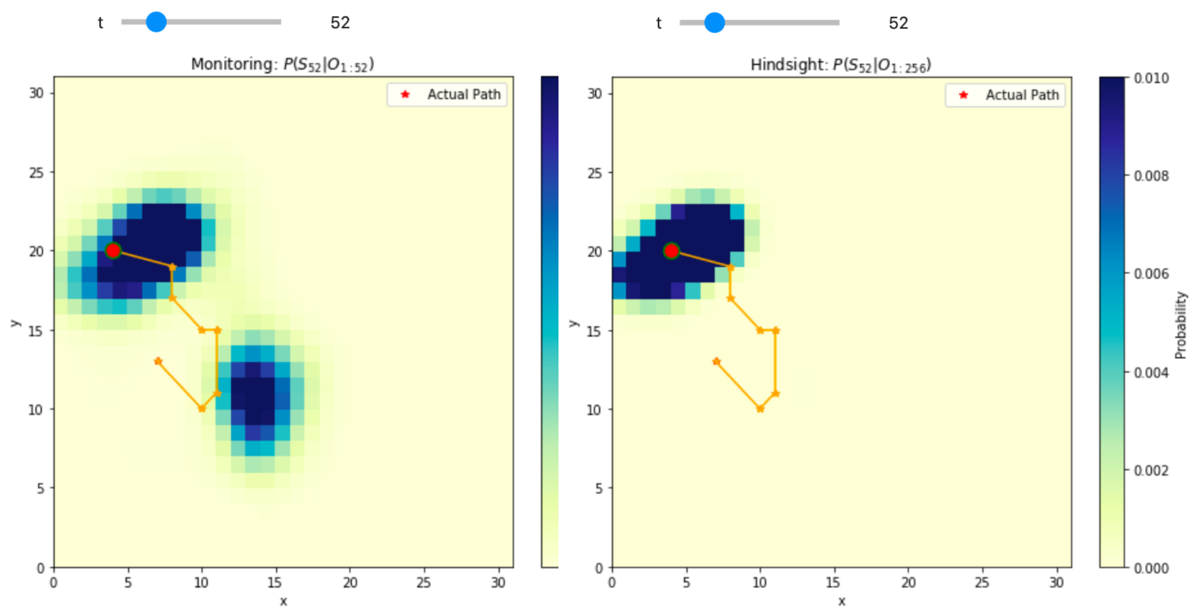
```
$ pip install numpy matplotlib ipywidgets
```

We strongly recommend that you use a Jupyter notebook (or JupyterLab) to plot the probabilities you compute. A couple of examples are shown in Figure 2b and 3. You also need to enable the widget extension to have interactive controls:

```
$ pip install jupyter
$ jupyter nbextension enable --py widgetsnbextension
```

Once you install it, you can run it by:

```
$ jupyter notebook /path/to/hw/directory
```



(a) Monitoring

(b) Hindsight

Figure 3: Example Monitoring and Hindsight Plots

The indoor environment is represented by a grid where each cell corresponds to a state in the HMM, which makes the total number of states `n_state = (width * length)`. Thus, a probability distribution of the location of the target sensor is a vector of length `n_states`. The transition probabilities $P(S_{t+1}|S_t)$ are stored in a 2-dimensional NumPy array, which can be accessed through `self.trans_probs`. For example, the transition probability from State 1 to State 3 can be obtained by `self.trans_prob[3,1]`. In each time step, the sensor receives an observation vector `o` of length `n_beacons`. The Received Signal Strength Indicator (RSSI) value of the message received from the beacon `b` is stored in `o[b]`. The probability of observing `o[b]` in a particular state `s`, $P(O^{(b)} = o_b|S = s)$, can be obtained by `self.obs_probs[b,o[b],s]`. Here, we assume the RSSI values of different messages are conditionally independent. Therefore, you can decompose the probability of observing the vector `o`, $P(O = o|S = s)$ into a product of individual likelihood functions:

$$P(O = o|S = s) = \prod_{i=0}^{\texttt{n\_beacons}-1} P(O^{(i)} = o_i|S = s).$$

2.3 Submission Instructions and Testing

You need to implement three methods of the `HMM` class, namely `predict`, `update` and `monitoring` in `hmm.py` file. The backward reasoning has already been implemented. You can access and use the attributes listed right above the constructor. You can also use NumPy (you should!) and the standard library, but you **cannot** use/import anyone elses code or any other package. You should only submit `hmm.py` to the Gradescope assignment "Homework 5: Probabilistic Reasoning - CODE", and you should upload it directly without zipping.

We have given you a Jupyter notebook `hmm.ipynb` where you can find several examples about how to use the methods and how to plot the computed probabilities. You can also test your code using `hmm_test.py`:

```
$ python -m unittest hmm_test.py
```

Here are some questions for you to consider, but you don't need to turn in your answers. Do you think the HMM filter (monitor) is robust? What if we guess the initial position of the sensor wrong? Is the filter still able to track the mobile sensor? Can we correct our guess for the initial position using hindsight reasoning? Do you think we need more than one beacon to track the sensor, or one is enough? How about more beacons? Is it worth it to buy a lot of beacons?