Workshop 8: Filtering
(Version 1.2)

1 Overview

This workshop relates to Lecture 8. In that lecture, we discussed Dynamic Bayesian Networks (DBNs). Here you will see how DBNs work using a mixture of Excel, and the Python Pomegranate package. In particular, you will carry out a mixture of filtering and prediction tasks on the umbrella network that we studied in the lecture.

2 Excel for filtering

The spreadsheet `umbrella-filtering.xls` that can be found on Blackboard models the umbrella example over over the first 2 days.

On the top line, the probability of rain for Day 0 is the prior probability (see Slides 26 and 49 in Lecture 8).

At the bottom of the sheet are the conditional probability tables for the transition model and the sensor model. The predicted probability for rain on Day 1 (top) is computed from the probability for Day 0 and the transition model. This is exactly as on Slide 49 (and 53).

To get the filtered probability, we have to bring in information about whether we saw an umbrella or not. The filtered probability of rain for Day 1 (middle of the sheet) is computed by combining the predicted probability for Day 1, the sensor model, and what we know about umbrellas. This gives the results you see on Slide 50 (and 53).

Note: There are two versions of the filtered probability. The raw values which we get directly from the calculation, and the normalized values (raw values scaled so they add to 1).

Look at what happens if you change the probability of umbrella/not umbrella. Currently the values say you see an umbrella (probability of umbrella is 1 and that of not umbrella is 0). What happens if you don't see an umbrella (probability of umbrella is 0 and that of not umbrella is 1)? What about if you have no information (probability of umbrella is 0.5 and that of not umbrella is 0.5)?

3 More filtering with Excel

The column for Day 2 just repeats the calculations for Day 1, but starting from the results from Day 1.

Thus the predicted probability for Day 2 is calculated by applying the transition model to the (normalized) filtered probability for Day 1. The results are just like those on Slide 51 (and 53).

The filtered probability of Day 2 is calculated from the predicted probability for Day 2, the sensor model, and what we know about umbrellas. The results are just like those on Slide 52 (and 53).

In other words, the probabilities for Day 2 are computed just like those for Day 1. The calculation is modular.

Look at what happens when the probabilities of umbrella/not umbrella on Days 1 and 2 vary.

4 Using pomegranate for Predicting and Filtering

For this example we will use a Python package called pomegranate, which provides support for probabilistic reasoning. If you have not installed it before, you will need to do so with:

```
pip install pomegranate==v0.14.9
```

Then you can run the version of the umbrella model in umbrella.py. pomegranate can only solve Bayesian neworks (not Dynamic Bayesian Networks), so we have to unroll the whole example to the depth that we want.

The file umbrella.py, which you can download from Blackboard, has the network unrolled to a depth of 2 days. Read through the code.

pomegranate makes it possible to specify the following elements:

- Variables

  Rain0, Rain1, Rain2, Umbrella1 and Umbrella2 are the variables here.

- Probability distributions.

  Variables can have probability distributions associated with them. The distribution associated with Rain0 is an example of a prior distribution, whereas those connecting Rain0 and Rain1, and then Rain1, Rain2 are conditional.

  Similarly, there are conditional distributions connecting Umbrella1 and Rain1, and Umbrella2, Rain2

- Nodes in a network

  s1 to s5 are nodes, associated with the variables Rain0, Rain1, Umbrella1, Rain2 and Umbrella2 respectively.

- Models

  model is defined as a Bayesian Network that includes all the nodes, and then edges that connect them.

- Evidence

  scenario specifies values for the variables in the model. So the setting in umbrella.py:

  ```
  scenario = [[None, None, None, None, None]]
  ```

  leaves all variables unspecified, while:

  ```
  scenario = [[None, None, 'y', None, None]]
  ```

  specifies that an umbrella was seen on day 1 (i.e. the variable Umbrella1 associated with node s3 has value 'y'.

If we run the model with:

```
scenario = [[None, None, None, None, None]]
```

we will compute the values of rain on Days 1 and 2 when we have no evidence of umbrellas. This is predicting rain on those days with no evidence. Do it. What results do you get?

Now add evidence of seeing an umbrella on Day 1. How does the probability of rain on Day 1 change? This is now the *filtered* probability of rain on Day 1.

Note that, in both these cases, pomegranate is giving us values for the probabilities of all the variables in the model. When we give it evidence about umbrella1, it not only gives us the filtered probability of rain on Day 1, but it also *predicts* the probability of rain on Day 2, and provides a *smoothed* estimate of the probability of rain on Day 0.

5 Day 2 using `pomegranate`

Use `pomegranate` to calculate the filtered probability of rain on Day 2 when we see an umbrella on Day 1 and Day 2. What is the filtered probability of rain on Day 2 when we don't see an umbrella on Day 1 (that is, the evidence is "no umbrella" on Day 1)? How about if we just have no information about Umbrellas on Day 1?

How do these results compare with what you get using Excel?

6 More prediction using Excel

Now go back to the Excel spreadsheet and predict the probability for Day 3. And Day 4. And so on until the predicted probability of rain has converged.

What value does it converge to?

7 Filtering Day 3 using Excel

Take your spreadsheet from the previous example, and add the filtered probability calculation for Day 3.

8 Day 3 using `pomegranate`

Extend the `pomegranate` model to handle Day 3.

Use your model to predict the probability of rain on Day 3 when the only evidence that you have is that you see an Umbrella on Day 1.

How does this value change when you also see an umbrella on Day 3?

9 Version list

- Version 1.0, 3rd December 2022.

- Version 1.1, 6th December 2022.

    - Corrected the name of the sample Python file.

- Version 1.2, 13th November 2023.

    - Corrected the number of referenced lecture's slide.