Informatics 2D: Reasoning and Agents

Alex Lascarides

School of informatics
The University of Edinburgh

Lecture 27b: Time and Uncertainty: Inference III

Where are we?

So far...

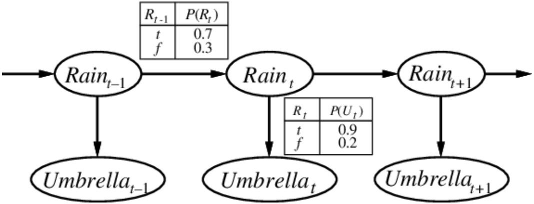
- Dynamic Bayesian Networks
- Inference:

  Filtering: $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$
  Likelihood: $\mathbf{P}(e_{1:t})$
  Prediction: $\mathbf{P}(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$
  Hindsight: $\mathbf{P}(\mathbf{X}_{t-k}|\mathbf{e}_{1:t})$

- Today: **Inference III: Estimating the most probable explanation for your observed evidence**

Reminder of example DBN

- Transition model $P(Rain_t|Rain_{t-1})$, sensor model $P(Umbrella_t|Rain_t)$

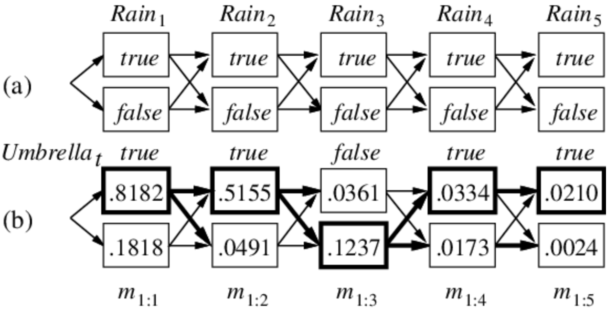


Finding the most likely sequence

- Suppose [*true*, *true*, *false*, *true*, *true*] is the umbrella sequence for first five days, what is the most likely weather sequence that caused it?
- Could use smoothing procedure to find posterior distribution for weather at each step and then use most likely weather at each step to construct sequence
- NO! Smoothing considers distributions over individual time steps, but we must consider **joint** probabilities over all time steps
- Actual algorithm is based on viewing each sequence as path through a graph (nodes=states at each time step)

Finding the most likely sequence

- In umbrella example:



- Look at states with $Rain_5 = true$ (part (a)), Markov property
  - most likely path to this state consists of most likely path to state at time 4 followed by transition to $Rain_5 = true$
  - state at time 4 that will become part of the path is whichever maximises likelihood of the path

Finding the most likely sequence

- There is a recursive relationship between most likely paths to $\mathbf{x}_{t+1}$ and most likely paths to each state $\mathbf{x}_t$

$$\max_{\mathbf{x}_1 \ldots \mathbf{x}_t} \mathbf{P}(\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha \mathbf{P}(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \max_{\mathbf{x}_t} (\mathbf{P}(\mathbf{X}_{t+1} | \mathbf{x}_t) \max_{\mathbf{x}_1 \ldots \mathbf{x}_{t-1}} P(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t | \mathbf{e}_{1:t}))$$

- This is like filtering only that the forward message is replaced by

$$\mathbf{m}_{1:t} = \max_{\mathbf{x}_1 \ldots \mathbf{x}_{t-1}} \mathbf{P}(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{X}_t | \mathbf{e}_{1:t})$$

- And summing (marginalisation) is now replaced by maximisation

Finding the most likely sequence

- This algorithm (**Viterbi algorithm**) is similar to filtering
- Runs forward along sequence computing **m** message in each step
- Progress in example shown in part (b) of diagram above
- In the end it has probability for most likely sequence for reaching each final state
  Easy to determine overall most likely sequence
- Has to keep pointers from each state back to the best state that leads to it

Hidden Markov Models

- So far, we have seen a general model for temporal probabilistic reasoning (independent of transition/sensor models)
- In this and the following lecture we are going to look at more concrete models and applications
- **Hidden Markov Models (HMMs)**: temporal probabilistic model in which state of the process is described by a single variable
- Like our umbrella example (single variable $Rain_t$)
- More than one variable can be accommodated, but only by combining them into a single "mega-variable"
- Structure of HMMs allows for a very simple and elegant matrix implementation of basic algorithms

Summary

- Finding the most likely sequence (Viterbi algorithm)
- Talked about HMMs
- HMMs: single state variable, simplifies algorithms (see other courses for these)
- Huge significance, for example in speech recognition:

$$P(words|signal) = \alpha P(signal|words)P(words)$$

- Vast array of applications, but also limits.
- Next time: **Dynamic Bayesian Networks: Model design**