

Reasoning Over Time

- Chapter 15.1-3

Time and Uncertainty

The world changes; we need to track and predict it

Diabetes management vs vehicle diagnosis

Basic idea: copy state and evidence variables for each time step

X_t = set of unobservable state variables at time t

e.g., BloodSugar_t, StomachContents_t, etc.

E_t = set of observable evidence variables at time t

e.g., MeasuredBloodSugar_t, PulseRate_t, FoodEaten_t

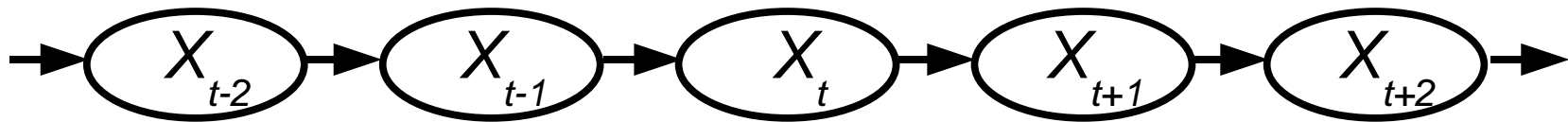
This assumes discrete time; step size depends on problem

Notation: $X_{a:b} = X_a, X_{a+1}, \dots, X_{b-1}, X_b$

Markov processes (Markov chains)

- Construct a Bayes net from the unobservable variables X_t
- Markov assumption:
 - X_t depends on bounded subset of $X_{0:t-1}$
- First-order Markov process (no memory):

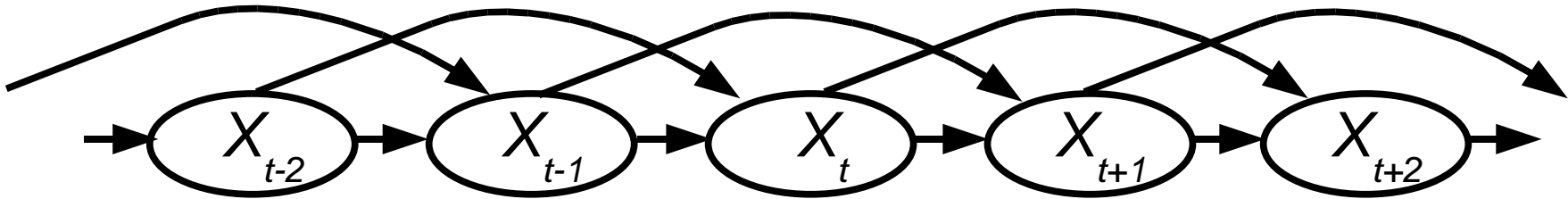
$$P(X_t | X_{0:t-1}) = P(X_t | X_{t-1})$$



Markov processes (cont.)

- Second-order Markov process (a little memory):

$$P(X_t | X_{0:t-1}) = P(X_t | X_{t-2}, X_{t-1})$$



States and Observations

- The variables X_t represent the *state* of the underlying system at each time step t .
- The *observable* variables E_t are *sensors*, these represent what we can sense about the state of the system.
 - There is some set of possible values for each E_t
- $P(E_t | X_{0:t}, E_{0:t-1}) = P(E_t | X_t)$
 - Observable variables depend only on the current state.
 - We can add to our Bayes net, the only parent of E_t is X_t

Stationary Processes

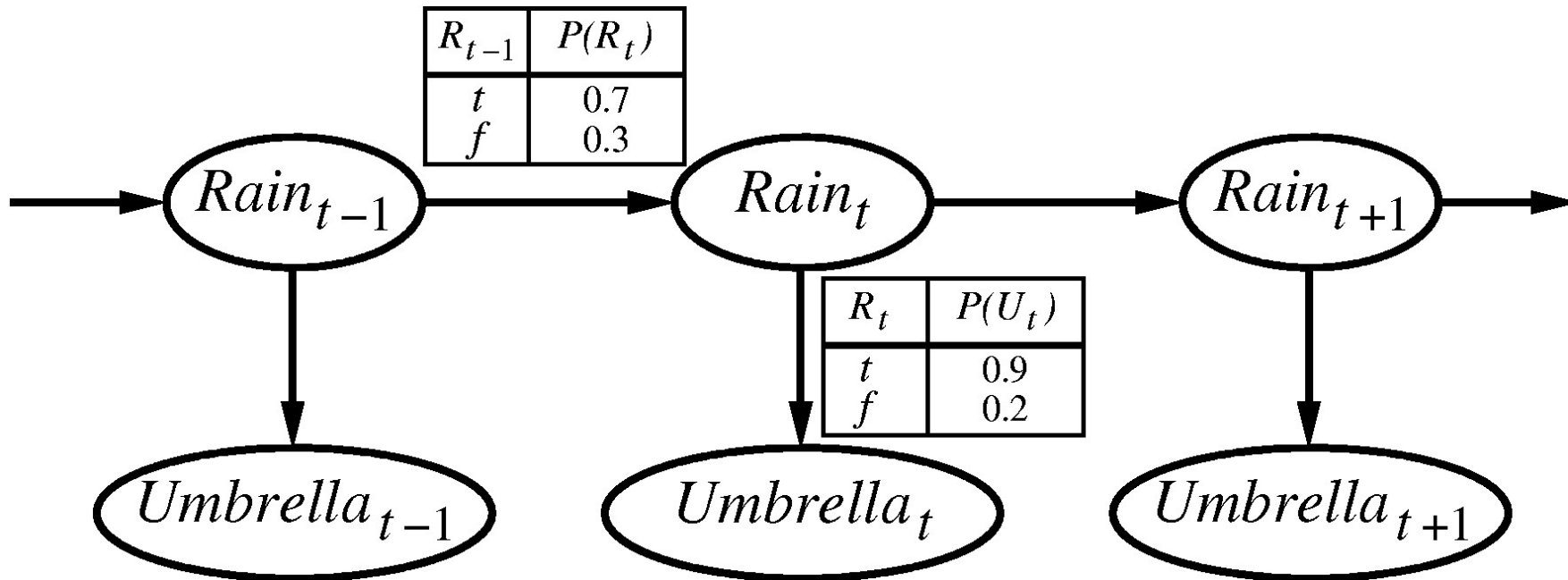
- We assume that the underlying process (system) is *stationary*:
 - The rules that determine the next state (and observation) are fixed.
- Without this assumption we can't do much!
- In general we want to figure out what the rules are, or build a set of rules to mimic some observable behavior.
 - This would be hard/impossible if we didn't assume the rules can't change every time step.

Umbrella Example

- You are a guard in a top-secret facility
 - There are no windows in the building.
 - You are never allowed outside.
- You want to know if it is raining outside.
- Each morning the boss comes to work
 - Sometimes she has an umbrella.
- State Variable R_t : it is raining on day t
- Observable Variable U_t boss has an umbrella on day t

Umbrella (cont.)

First Order Markov Process



Admittedly weak, we would probably want to use a higher-order and add additional states variables (temperature, pressure, ...)

Another Example: Robot Position

- You want to track the position of a robot that wanders randomly.
- You know the position and velocity at time $t-1$
- When the battery gets low, the robot motion changes in a predictable manner, how could we account for this?
 - The battery level could depend on how far has the robot traveled. Could we take advantage of this?
 - This violates the Markov assumption (we can only look back a fixed amount of time).
 - We could add a “battery charge” variable and possibly sensor.

Inference Tasks

Filtering (monitoring): $P(X_t | e_{1:t})$

belief state - input to the decision process of a rational agent

Prediction: $P(X_{t+k} | e_{1:t})$ for $k > 0$

evaluation of possible action sequences

Smoothing: $P(X_k | e_{1:t})$ for $0 \leq k < t$

better estimate of past states, essential for learning

Most likely explanation: $\arg \max_{X_{1:t}} P(x_{1:t} | e_{1:t})$

speech recognition, decoding with a noisy channel

Filtering/Prediction Math

- General idea is to compute $P(X_{t+1}|e_{1:t+1}) = P(X_{t+1}|e_{1:t}, e_{t+1})$ in online fashion (each time step).
- We can use the result from computation at the previous time step.

$$\begin{aligned} P(X_{t+1}|e_{1:t+1}) &= P(X_{t+1}|e_{1:t}, e_{t+1}) \\ &= \alpha P(e_{t+1}|X_{t+1}, e_{1:t}) P(X_{t+1}|e_{1:t}) \\ &= \alpha P(e_{t+1}|X_{t+1}) P(X_{t+1}|e_{1:t}) \end{aligned}$$

Bayes Rule

Markov Property

α is a normalizing constant : $1/P(e_{1:t+1})$

Prediction Math (cont.)

$$P(X_{t+1} | e_{1:t+1}) = \alpha P(e_{t+1} | X_{t+1}) P(X_{t+1} | e_{1:t})$$

This represents the new observable at time t, and comes directly from the CPT for the observable variable E

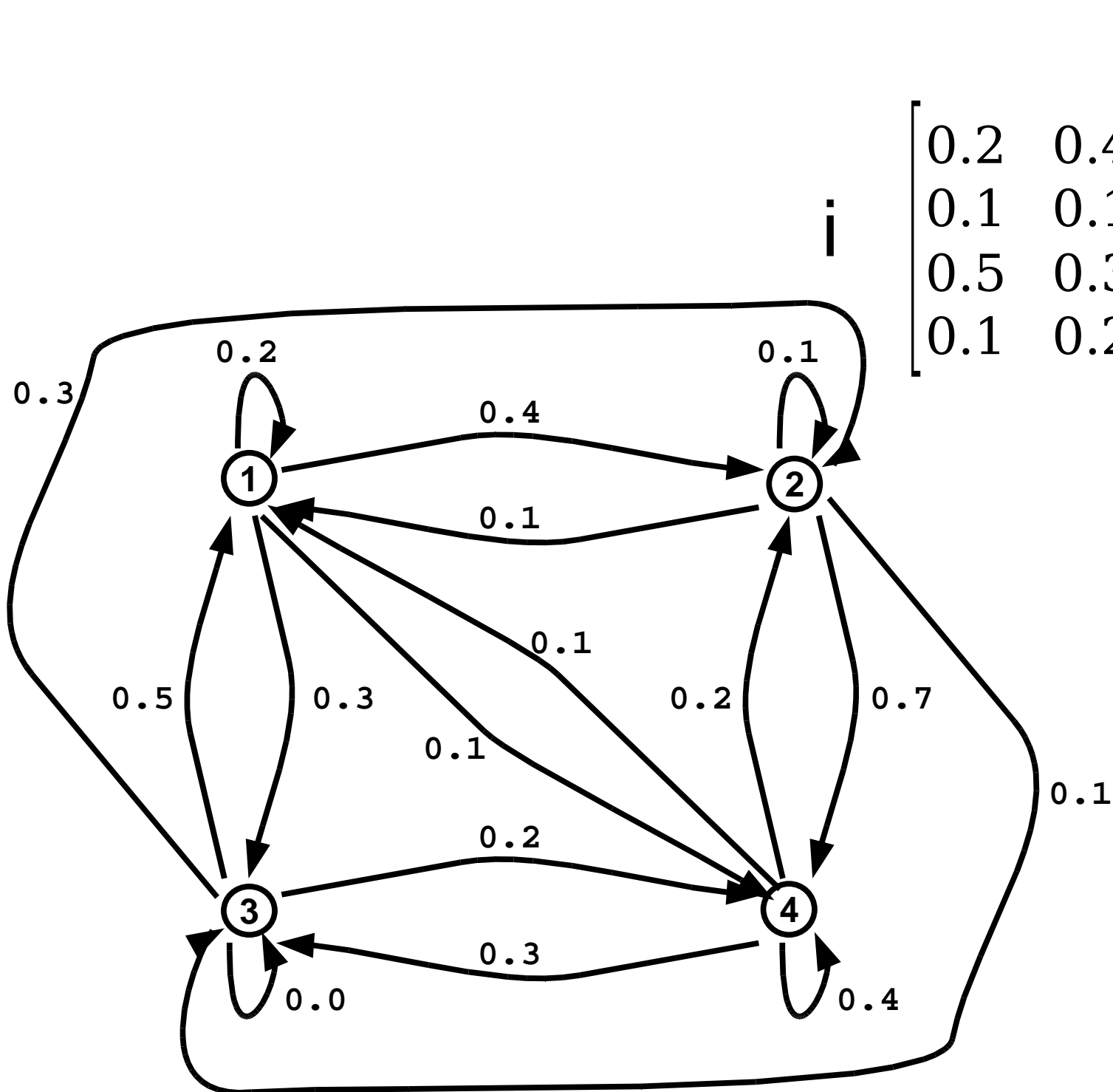
Prediction of the next state, this can be computed by summing over the probabilities of all possible next states.

$$\begin{aligned} P(X_{t+1} | e_{1:t}) &= \sum P(X_{t+1} | x_t, e_{1:t}) P(x_t | e_{1:t}) \\ &= \sum P(X_{t+1} | x_t) P(x_t | e_{1:t}) \end{aligned}$$

Markov Property

Hidden Markov Models

- Sometimes we know nothing about the underlying states.
 - We make an assumption that there is a single state variable X that can have some fixed number of values N .
 - Values of X_t can be from $\{1, \dots, N\}$
 - We are really guessing that there are N distinct states.
- All we have are observable variables (often just one).
- We can still make predictions.



j

	0.2	0.4	0.3	0.1
0.1	0.1	0.1	0.1	0.7
0.5	0.3	0.0	0.2	
0.1	0.2	0.3	0.4	

i

Output (Observable Variable)

- Each time step the current state generates an output O_t
 - $O_t \in \{o_1, o_2, o_3, \dots, o_M\}$
- For each state we have a probability distribution $b_j(k)$ over the possible outputs:
 - $b_j(k) = P(o_k \text{ at time } t \mid X_t = j)$
the probability of outputting o_k given that we are in state j
- We refer to the entire collection of these distributions as B .

Initial State Probability Distribution

- We also need the probabilities of starting out in each possible state:

$$\pi_i = P(X_1 = i)$$

- Given:
 - A (state transition matrix) for N states
 - B (probabilities for each of the M outputs, for each state)
 - π (Initial state probabilities)
- We can generate a sequence of a fixed number of outputs according to the probabilities.

HMM Issues

- 1) Given a sequence of outputs and a HMM (A, B and π), how do we want to compute the probability of the output sequence.
- 2) Given a sequence of outputs and a HMM, can we come up with an *optimal* state sequence to explain the output sequence ?
- 3) Given a sequence of outputs, can we build a HMM (by defining the value in A, B and π) that could generate the sequence with high probability (or perhaps the best HMM for generating the observed sequence)?

HMM Math

- Problem 1 can be solved exactly. The brute-force approach is to compute all possible sequences of states that could result in the output sequence observed, and to add the probabilities of each of the sequences.
 - This is a large computation! Assuming the output sequence contains T symbols (T time steps), we need to process (roughly) $2TN^T$ operations.
 - Some of these operations will be duplicates, and folks have come up with clever schemes to reduce this to N^2T operations.

Creating a HMM from observations.

- It's not know how to solve problem 3 (construct an *optimal* HMM given an output sequence).
- There are methods that work well (just no guarantee that the resulting HMM is the best HMM to produce the output sequence).
- The methods used involve iterating over the output sequence, adjusting probabilities each time to improve the probability of the output sequence.
 - This will converge, but it won't necessarily be the global optimum...
- Important issue: how many states are there?

HMM usage

- HMMs are used for many recognition problems:
 - Speech recognition (just about all the good ones are based on HMMs).
 - Handwriting recognition.
 - Bioinformatics: protein folding.
- General Usage: given some representative sample of data, build a HMM that models a process that could generate the sample data.
 - generation/recognition are essentially the same.
- Given some new data, see how easily the HMM could produce it.
- Sometimes there are many such HMMs competing to see which is the best match to some observed data.

