COL333/671: Introduction to AI

Semester I, 2023-24

Probabilistic Reasoning over Time

Acknowledgement

These slides are intended for teaching purposes only. Some material has been used/adapted from web sources and from slides by Doina Precup, Dorsa Sadigh, Percy Liang, Mausam, Dan Klein, Anca Dragan, Nicholas Roy and others.

Outline

- Background
  - Decision making relies on the knowledge of the agent's state. Often the state is not available to us directly.
  - Often, we reason over time. We have some knowledge of how the agent's state is likely to evolve.
- Hidden Markov Models
  - Representation (conditional independencies)
  - Queries (most likely estimate of the current state).
- Particle Filtering
  - How to deal with very large state spaces?
- Viterbi Algorithm
  - What if we want not just the current state estimate a an estimate of all states given observations.
- Applications

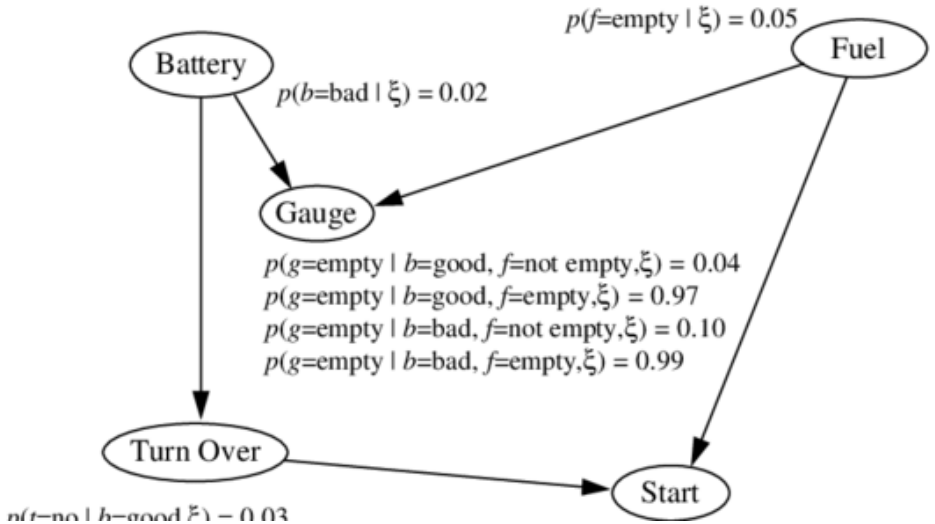
Uncertainty in “Knowing” the State

- Decision making relies on knowledge of the agent’s state
  - Location of a robot in a grid for path planning
  - A persons’ intent for a recommendation system
  - What was spoken for a conversational agent
- But, how does one “know” the agent’s state?
- In practice, there are cues/features that we see.
  - Observe/measure the position of the robot in a grid
  - Record text/clicks that a person is typing during online shopping
  - Record audio spoken by a person
- Such *observations* or *measurements* are a “noisily” linked to the actual state (which we can’t see directly)

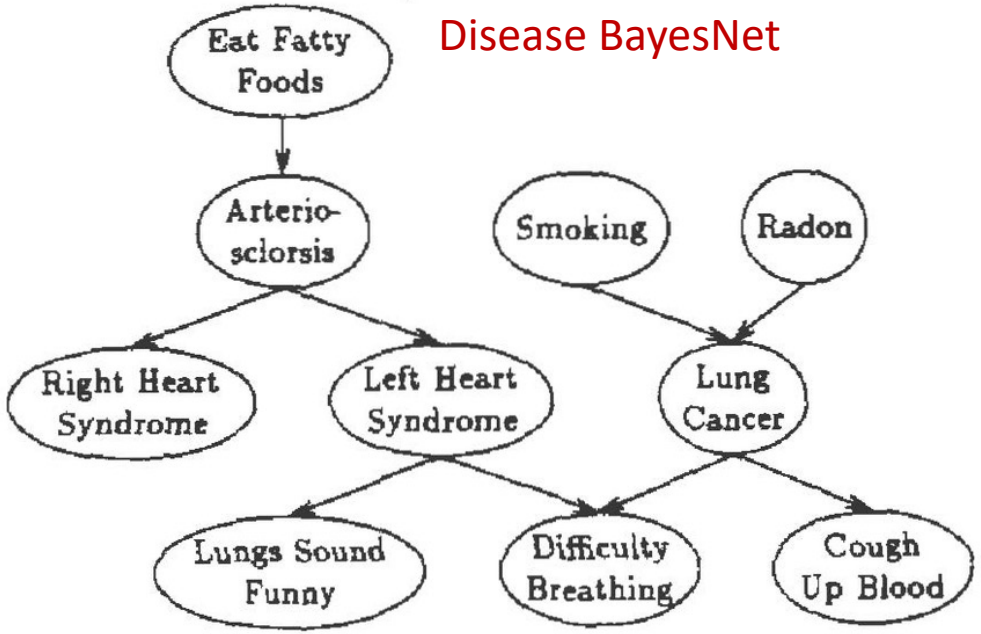
Bayes Net

- Decision making till now was “one off” or a specific time step.
- Stochastic linking between variables encoded in a Bayes Net.
- The latent variables were “inferred” **given** observed features.

Faulty car BayesNet



Disease BayesNet



Reasoning over Time

- Often, we take decisions over time
  - We reason with a ***sequence*** of observations.

Predicting student attrition

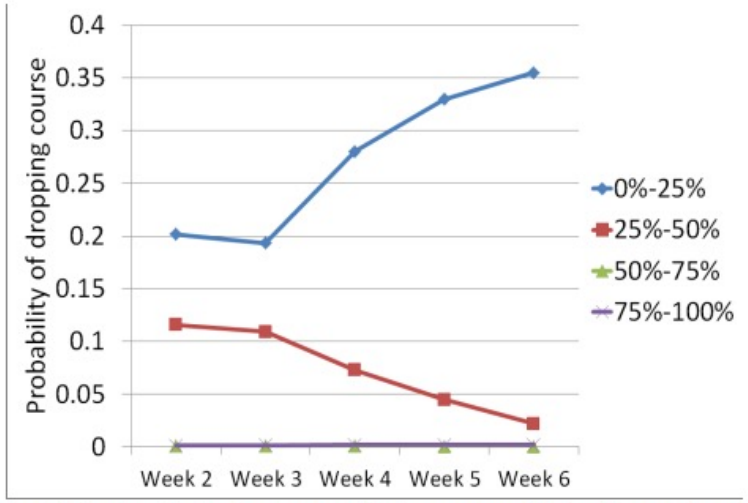


Figure 4 Attrition with time for students who view a consistent percentage of lecture videos each week. As an example, if a student is active in the course up until week 4, and views 25%-50% of lecture minutes each week during that period, their likelihood of dropping the course in week 4 is about 8%, as shown by the red line.

Features that indicate engagement

2.2.3 Forum Interaction Features

The forum is the primary means of student support and interaction during the course. The forum's basic software mechanisms allow us to observe the following useful features:

1. Number of threads viewed this week, where a thread can only be viewed once a day. Since most active students undertake this passive interaction it is an important metric of engagement.
2. Number of threads followed this week, which is a slightly more active sign of engagement than 1.
3. Number of upvotes given this week, indicating posts students found to be useful, which is also a more active sign of engagement.
4. Number of posts made this week. Although most students aren't active on the forum, for those who are this feature is a strong indicator of engagement and sense of community.
5. Number of replies received this week to any post previously made. This is very important as it directly correlates with how much belonging a student feels in the course.
6. Number of upvotes received this week to any post previously made. This is important for the same reasons as 5.

2.2.4 Assignment Features

Students are exposed to ungraded lecture problems that are intertwined with lecture videos, as well as graded quizzes and homeworks that assess their understanding of the material. Graded assignments are conveniently due at the end of a week. Since these types of problems carry very different weights, we define them individually as follows:

1. Cumulative percentage score on homework problems that are due at the end of this week, or have been due in previous weeks. When monitoring this value from week to week, we again get a good gauge on how far up-to-date a student is on the course.
2. Cumulative percentage score on quiz problems that are due at the end of this week, or have been due in previous weeks.
3. Cumulative percentage score on lecture problems that are available from the start of the course until this week. The difference between this and features 1 and 2 is that there is no due date for lecture problems, so a student actually has the possibility to catch up on them at any point in the course.
4. Percentage score on homework problems that are only due this week. The score that a student

Reasoning over Time

- Often, we take decisions over time
  - We reason with a ***sequence*** of observations.
- Example:
  - Diagnosing Covid
  - State = {Covid, Non Covid}
  - Observation at time t = {fever = T, running_nose = False, ……}
- Note: the state evolves over time
  - The disease has a certain progression (state variables are correlated)

Examples

Predicting student attrition

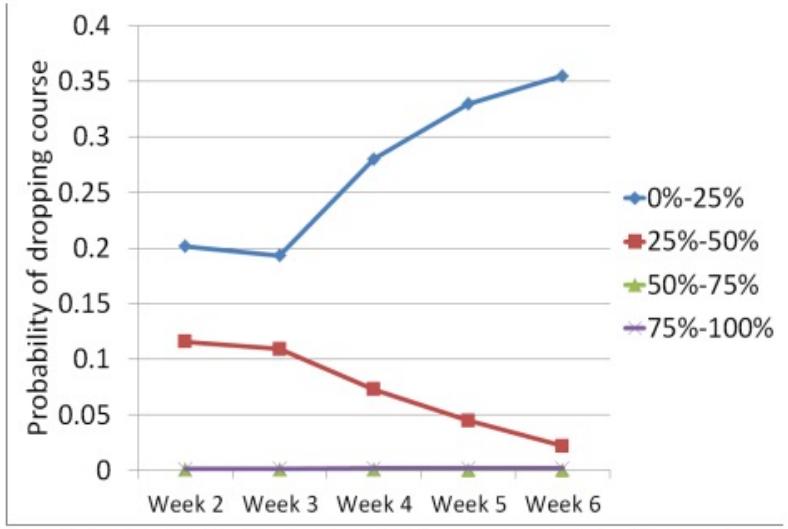


Figure 4 Attrition with time for students who view a consistent percentage of lecture videos each week. As an example, if a student is active in the course up until week 4, and views 25%-50% of lecture minutes each week during that period, their likelihood of dropping the course in week 4 is about 8%, as shown by the red line.

2.2.3 Forum Interaction Features

The forum is the primary means of student support and interaction during the course. The forum's basic software mechanisms allow us to observe the following useful features:

1. Number of threads viewed this week, where a thread can only be viewed once a day. Since most active students undertake this passive interaction it is an important metric of engagement.
2. Number of threads followed this week, which is a slightly more active sign of engagement than 1.
3. Number of upvotes given this week, indicating posts students found to be useful, which is also a more active sign of engagement.
4. Number of posts made this week. Although most students aren't active on the forum, for those who are this feature is a strong indicator of engagement and sense of community.
5. Number of replies received this week to any post previously made. This is very important as it directly correlates with how much belonging a student feels in the course.
6. Number of upvotes received this week to any post previously made. This is important for the same reasons as 5.

2.2.4 Assignment Features

Students are exposed to ungraded lecture problems that are intertwined with lecture videos, as well as graded quizzes and homeworks that assess their understanding of the material. Graded assignments are conveniently due at the end of a week. Since these types of problems carry very different weights, we define them individually as follows:

1. Cumulative percentage score on homework problems that are due at the end of this week, or have been due in previous weeks. When monitoring this value from week to week, we again get a good gauge on how far up-to-date a student is on the course.
2. Cumulative percentage score on quiz problems that are due at the end of this week, or have been due in previous weeks.
3. Cumulative percentage score on lecture problems that are available from the start of the course until this week. The difference between this and features 1 and 2 is that there is no due date for lecture problems, so a student actually has the possibility to catch up on them at any point in the course.
4. Percentage score on homework problems that are only due this week. The score that a student

Wildlife Monitoring

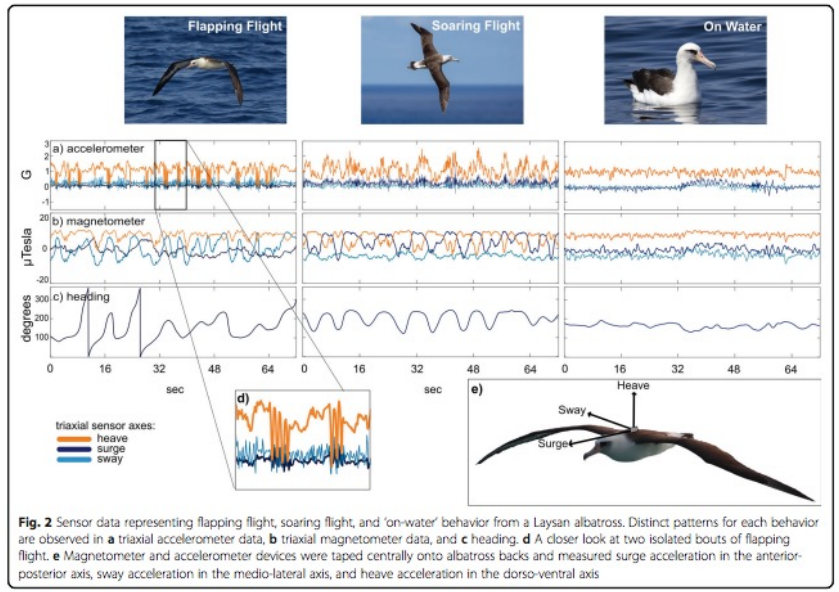
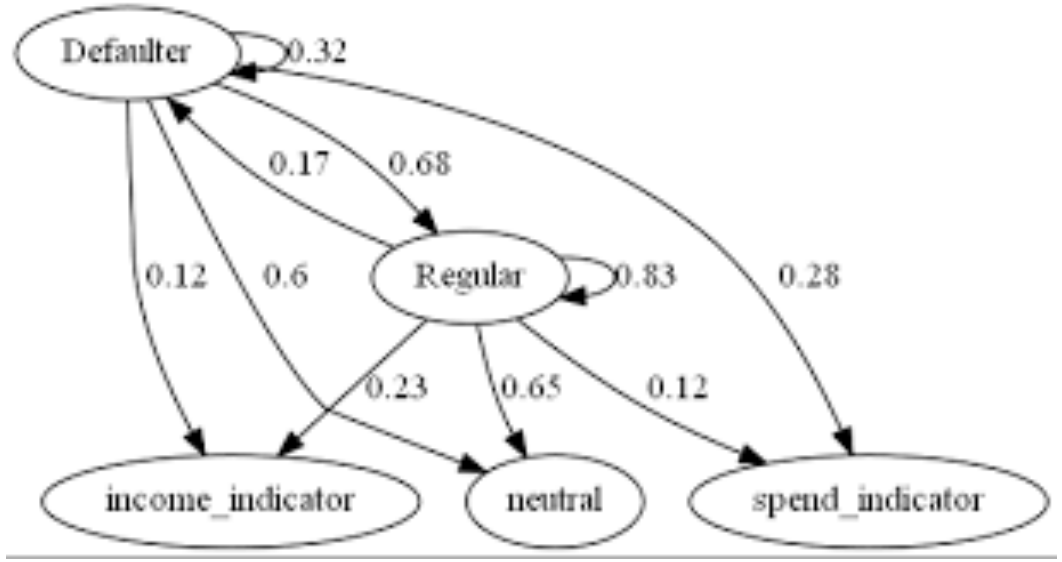


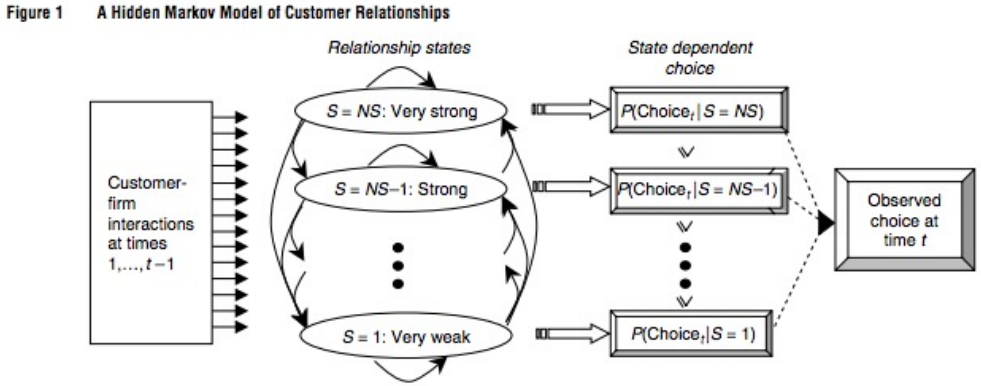
Fig. 2 Sensor data representing flapping flight, soaring flight, and 'on-water' behavior from a Laysan albatross. Distinct patterns for each behavior are observed in **a** triaxial accelerometer data, **b** triaxial magnetometer data, and **c** heading. **d** A closer look at two isolated bouts of flapping flight. **e** Magnetometer and accelerometer devices were taped centrally onto albatross backs and measured surge acceleration in the anterior-posterior axis, sway acceleration in the medio-lateral axis, and heave acceleration in the dorso-ventral axis

https://movementecologyjournal.biomedcentral.com/articles/10.1186/s40462-021-00243-z

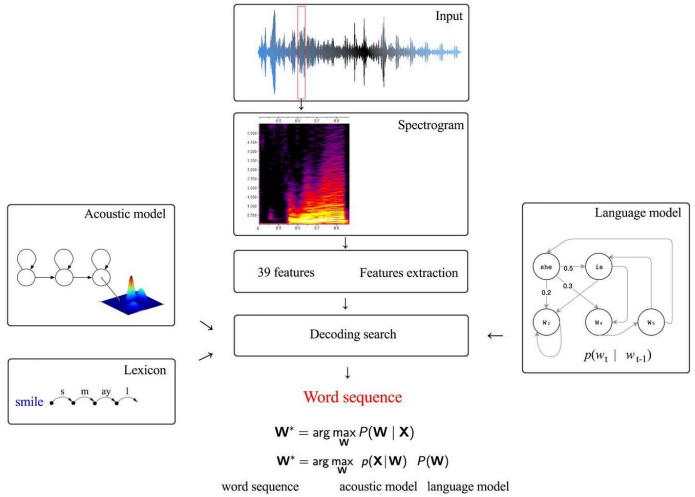
Loan Monitoring



Predicting Alumni (dis)Engagement



Speech Recognition

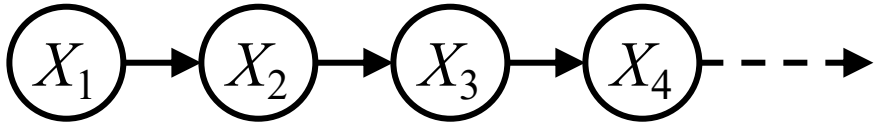


Core of reasoning over time

- What do we model?
  - Modelling how the “state” transitions
    - Markov models
  - Modelling the link between state and observations
    - How the observations are correlated with the state
- What is observed? (Evidence)
  - Measurements collected over time
- What is to be inferred? (Probabilistic Queries)
  - Given the measurements (+ initial state distribution) what is the current state
  - Or what will be the state after some time
  - Or what is the best prediction for a state in the past

Transition Models: Markov Models

- Value of X at a given time is called the **state**.



- **Transition probabilities** or dynamics,
  - Specify how the state evolves over time
  - Initial state probabilities
  - Stationarity assumption: transition probabilities the same at all times.

$$P(X_1) \qquad P(X_t|X_{t-1})$$

- (First order) Markov Property
  - Past and future independent given the present
  - Each time step only depends on the previous

- Note: there can be higher-order dependencies (they are more complex to reason with, first order works well).

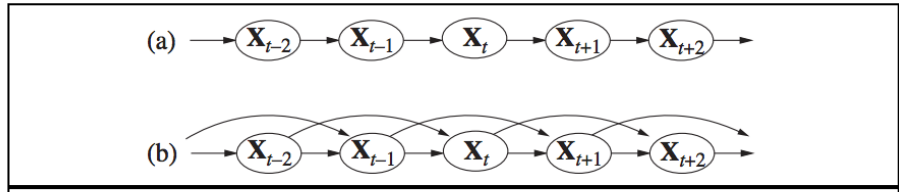


Figure 15.1 (a) Bayesian network structure corresponding to a first-order Markov process with state defined by the variables $\mathbf{X}_t$. (b) A second-order Markov process.

Transition Models: An example

What is the probability that the state = sunny given state – rainy the previous day?

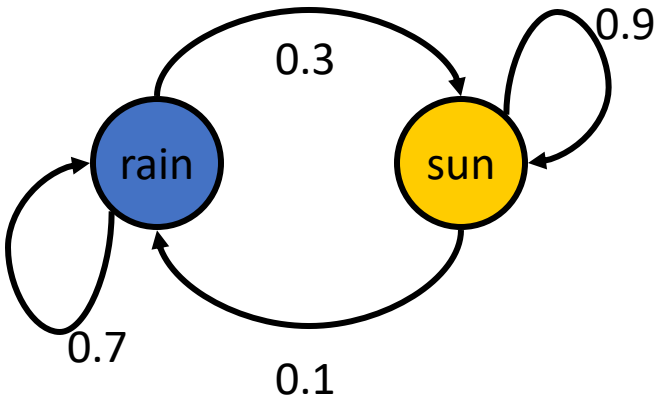
States: X = {rain, sun}

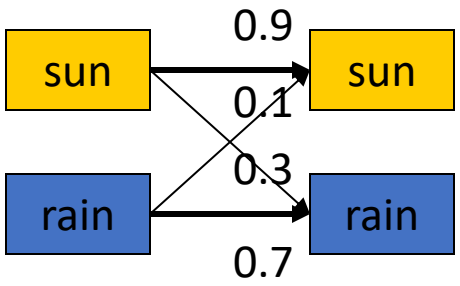
Initial distribution: 1.0 sun

CPT $P(X_t \mid X_{t-1})$:

X_{t-1}	X_t	$P(X_t \mid X_{t-1})$
sun	sun	0.9
sun	rain	0.1
rain	sun	0.3
rain	rain	0.7

Diagrammatically representing the transition model.

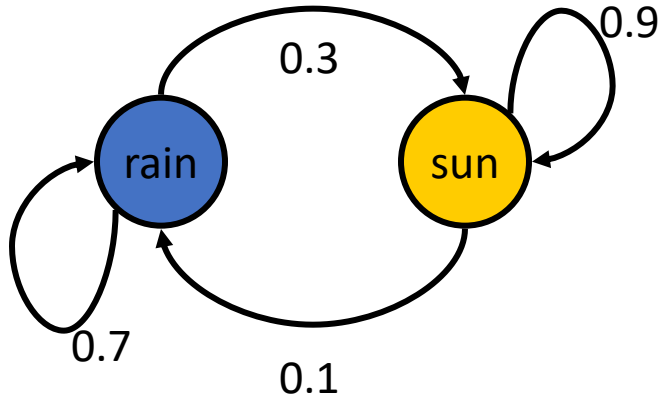




Weather has a natural evolution. S, R, S, R are less likely to occur.

Computing Likelihoods over Time

- Initial distribution: 1.0 sun



- What is the probability distribution after one step?

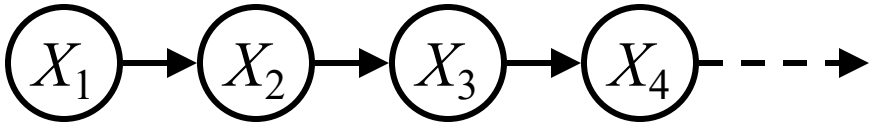
$$P(X_2 = \text{sun}) = P(X_2 = \text{sun}|X_1 = \text{sun})P(X_1 = \text{sun}) + P(X_2 = \text{sun}|X_1 = \text{rain})P(X_1 = \text{rain})$$

$$0.9 \cdot 1.0 + 0.3 \cdot 0.0 = 0.9$$

Essentially, for predicting a future state, marginalizing out the possible states in the past.

Forward Algorithm for a Markov Chain

- What's P(X) on some day t?



$$P(x_1) = \text{known}$$

$$P(x_t) = \sum_{x_{t-1}} P(x_{t-1}, x_t)$$

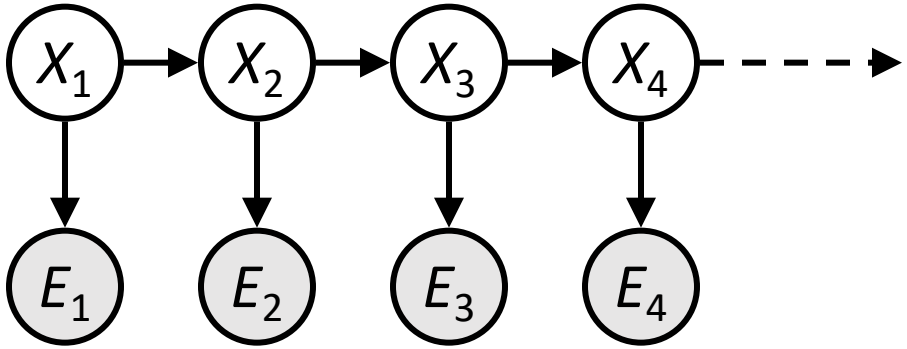
$$= \sum_{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1})$$

The second variable is marginalized out from the joint distribution.

Forward simulation

Hidden Markov Models (HMMs)

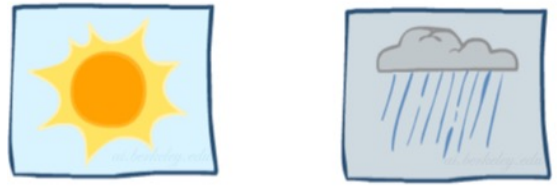
- Markov Chains
  - Assume that we observe the state directly.
  - Often this is not the case. We only have noisy observations of the state.



- **"Hidden"** Markov Models
  - Underlying Markov chain over states X
  - You observe outputs (effects) at each time step

Note: the actual state is latent.
- We observe observations. The observations are correlated with the actual state.
- Given the observation we try to predict what the latent state was?

Is the weather on the day rainy or not rainy (sunny)?



Weather HMM

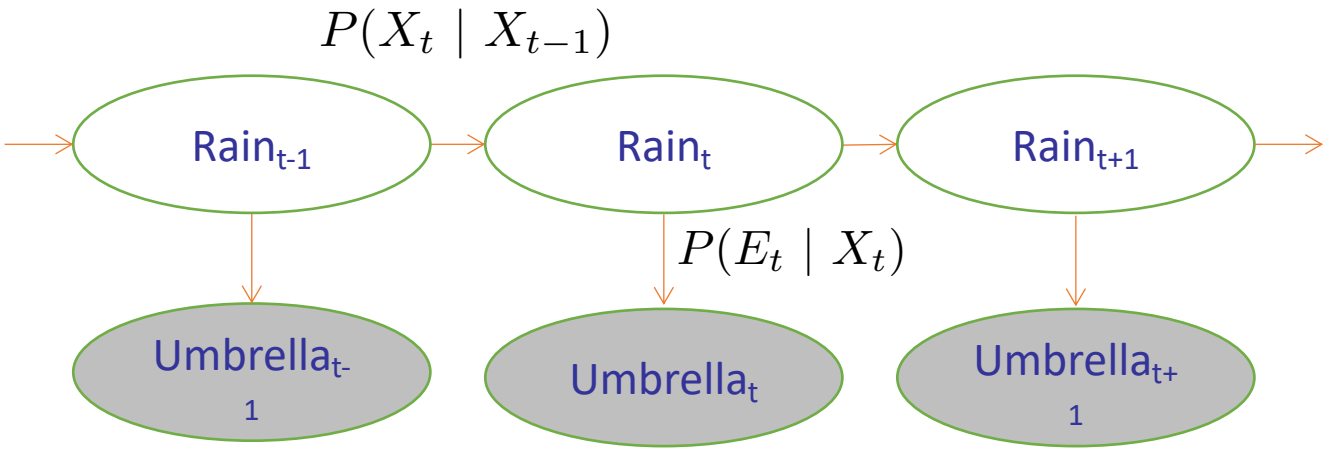
State: The world state (rainy or sunny) is not directly observed. You are inside an underground building.

Observations: Observe a person coming such as a person carrying an umbrella or not. Note: *Carrying an umbrella is likely when it is rainy, but people tend to carry umbrellas even on a sunny day.*

Inference: Once we observe some persons bring umbrella or not, what can we say about the weather today?

HMM characterized by:

- Initial distribution: $P(X_1)$
- Transitions: $P(X_t \mid X_{t-1})$
- Emissions: $P(E_t \mid X_t)$



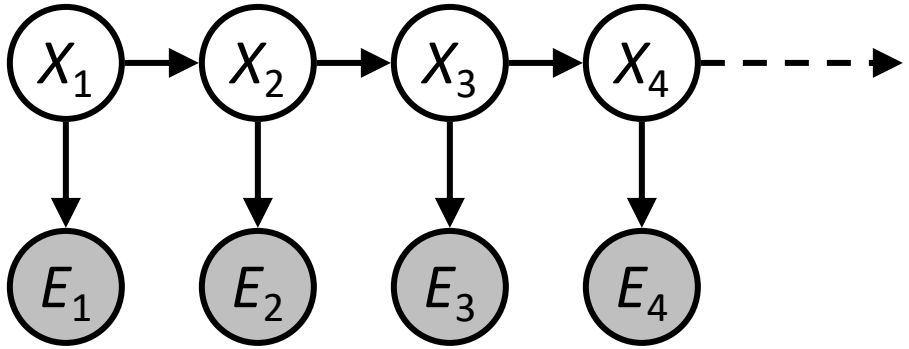
R_{t-1}	R_t	$P(R_t \mid R_{t-1})$
+r	+r	0.7
+r	-r	0.3
-r	+r	0.3
-r	-r	0.7

R_t	U_t	$P(U_t \mid R_t)$
+r	+u	0.9
+r	-u	0.1
-r	+u	0.2
-r	-u	0.8

What conditional independences are encoded?

HMMs make two important independence assumptions.

- Future state depends on past states via the present state.
- The current observation is independent of all else given current state

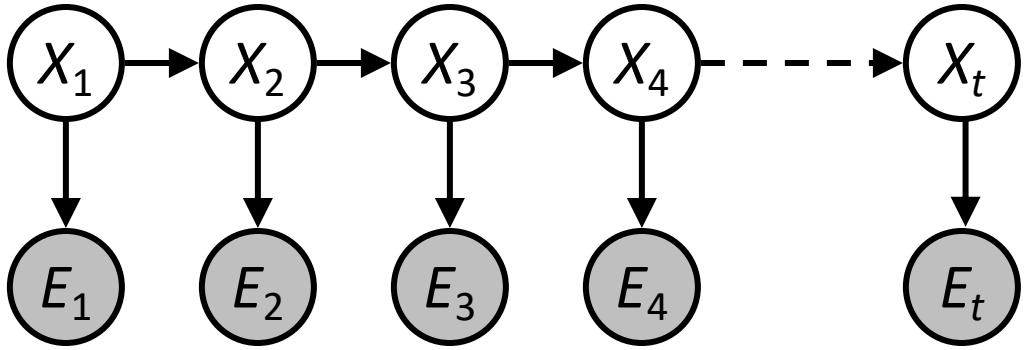


$$\mathbf{P}(\mathbf{X}_t \mid \mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t \mid \mathbf{X}_{t-1})$$

$$\mathbf{P}(\mathbf{E}_t \mid \mathbf{X}_{0:t}, \mathbf{E}_{0:t-1}) = \mathbf{P}(\mathbf{E}_t \mid \mathbf{X}_t)$$

Inference: Filtering or Monitoring Task

- Filtering, or monitoring, is the task of tracking the distribution
  - $B_t(X) = P_t(X_t \mid e_1, \ldots, e_t)$ (the belief state) over time
- We start with $B_1(X)$ in an initial setting, usually uniform
- As time passes, or we get observations, we update B(X)

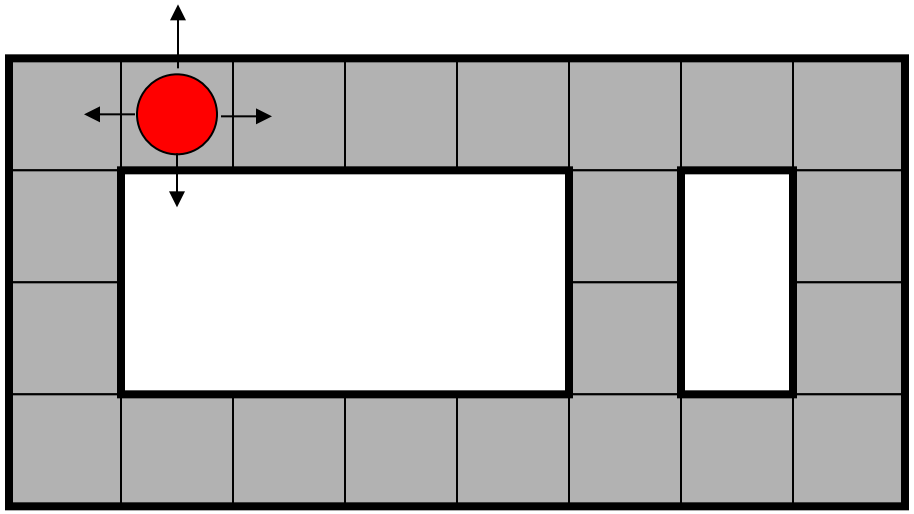


Example: Given the fever levels on four days, what is the probability that the person's covid status on day 5?

Example: Robot Localization

Robotic vacuum cleaners.

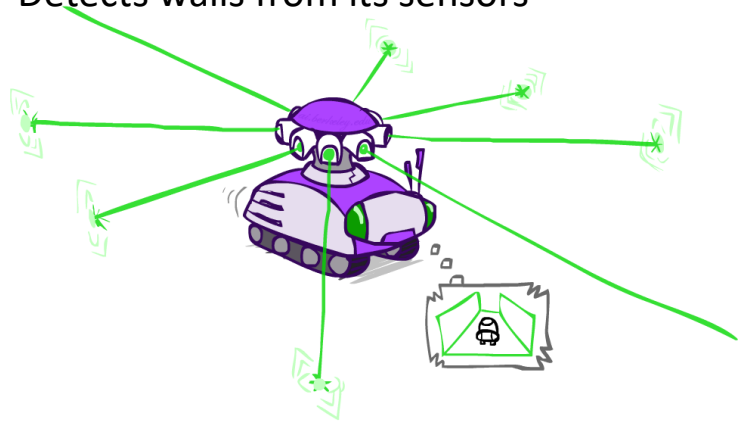
A robot vacuum cleaner has four sensors that are noisy. Can it figure out where it is in the room?



Prob 0 1

t=0

Robot can take actions of taking one step to N, S, E, W directions.
Detects walls from its sensors

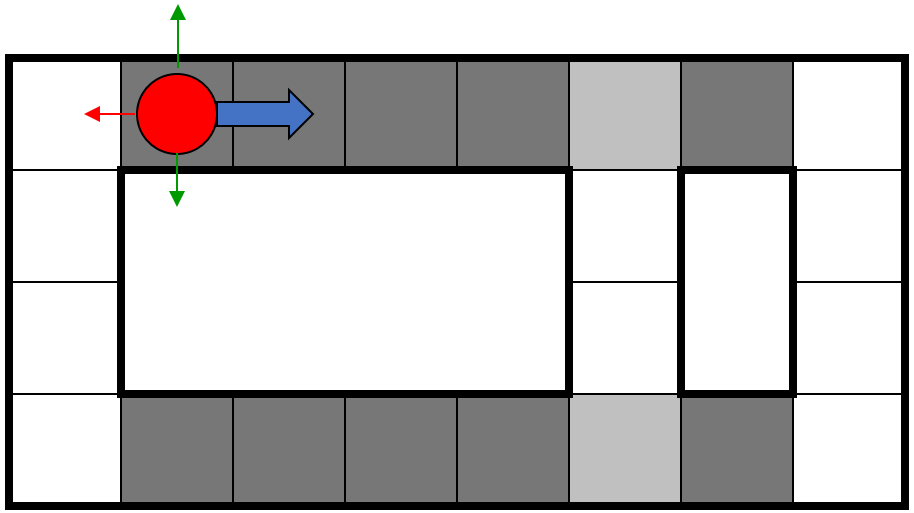


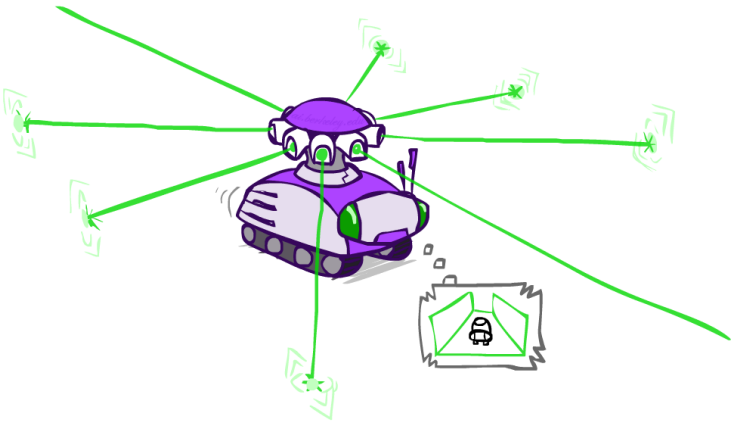
Q: What is an observation?
Q: What is the state?

Sensor model: can read in which directions there is a wall, never more than 1 mistake.

Motion model: may not execute action with small probability, so can move or stay there (stochastically). Most of the times move in the correct direction.

Example: Robot Localization



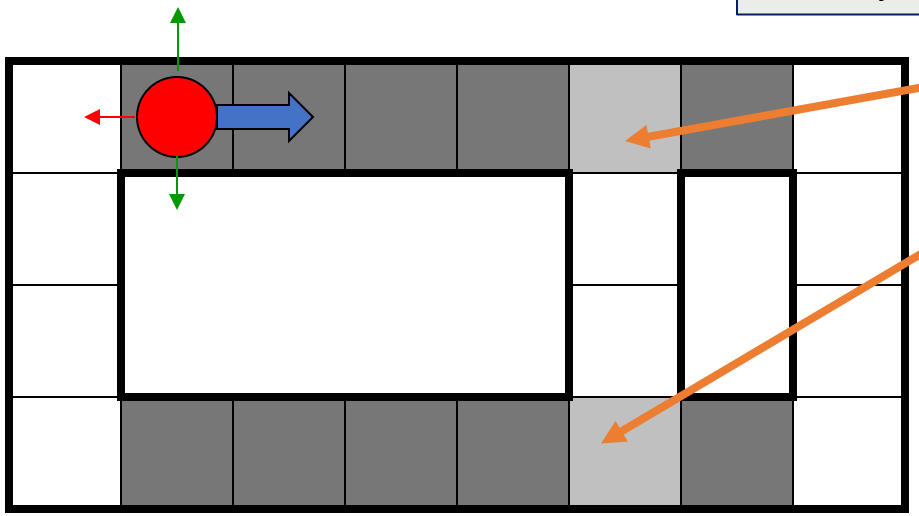


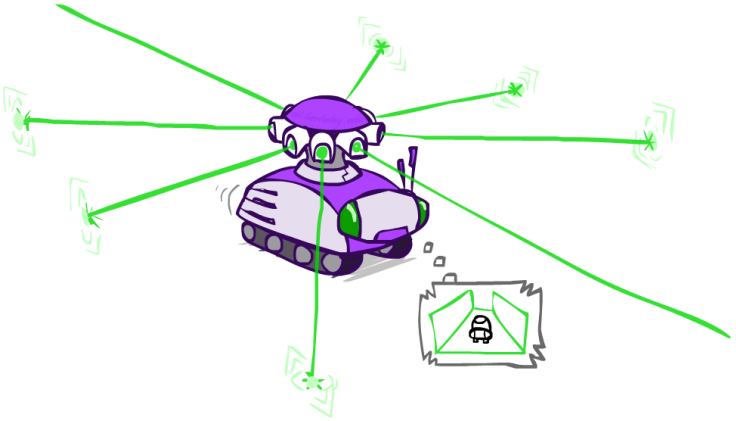
Prob 0 1

t=1

Example: Robot Localization

Q: Why are these light grey? Why is there symmetry?



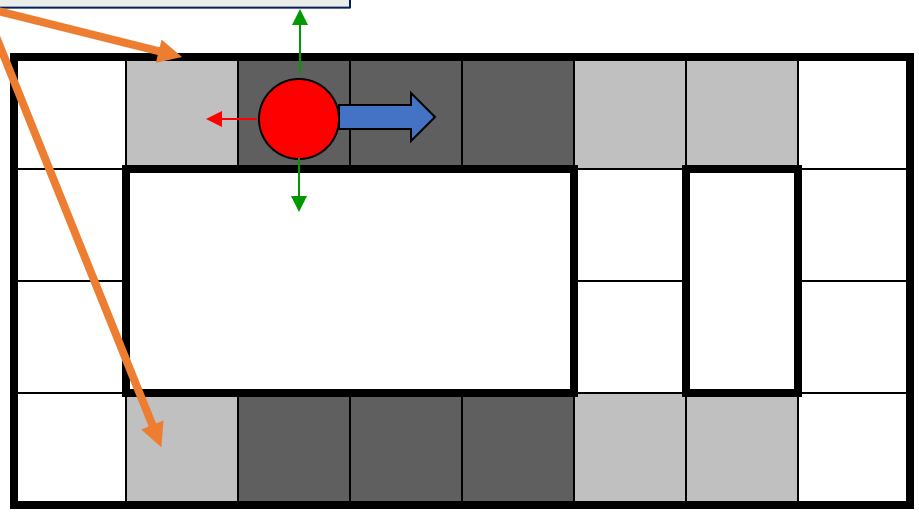


Prob 0 1

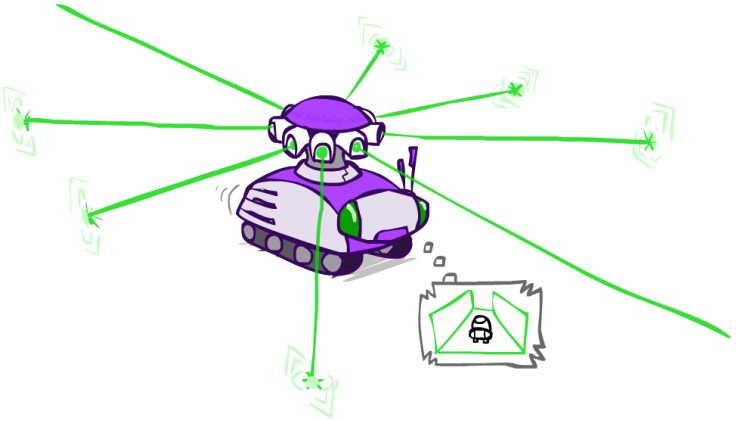
t=1

Example: Robot Localization

Q: These cells become light. Why?

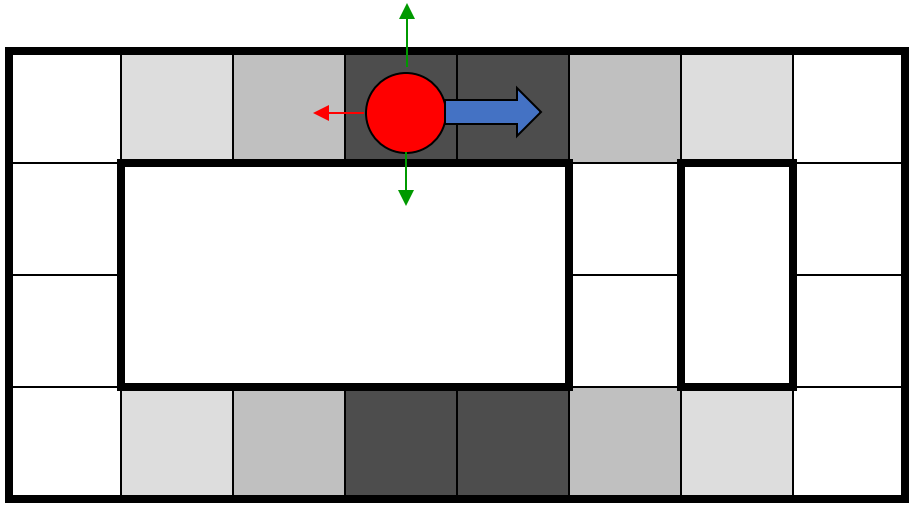


Prob 0 1

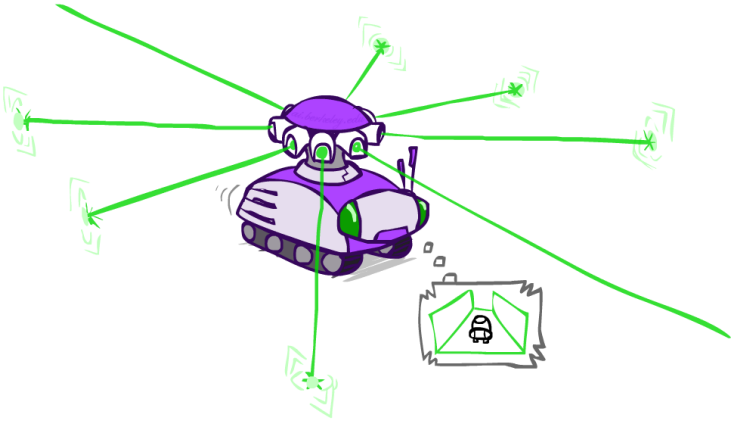


t=2

Example: Robot Localization

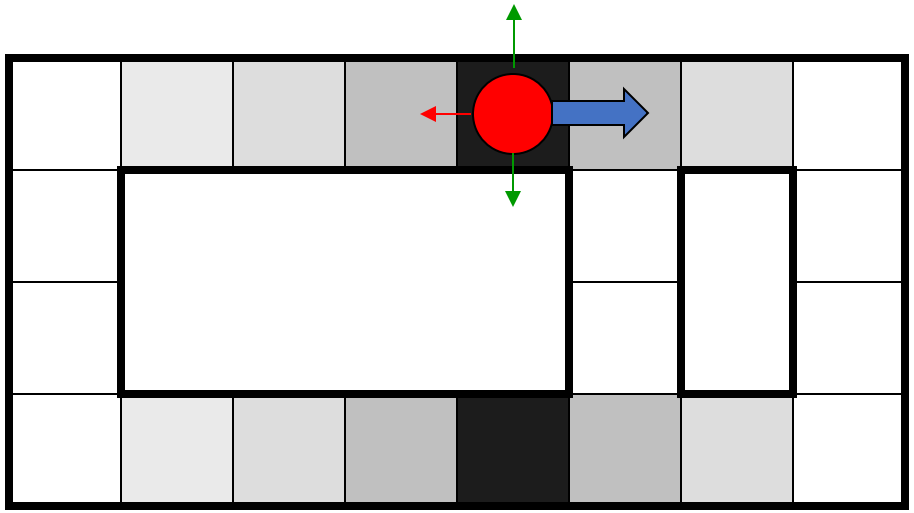


Prob 0 1



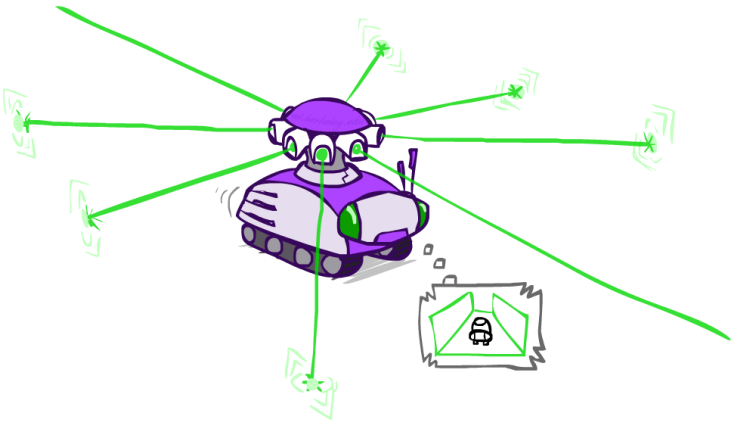
t=3

Example: Robot Localization



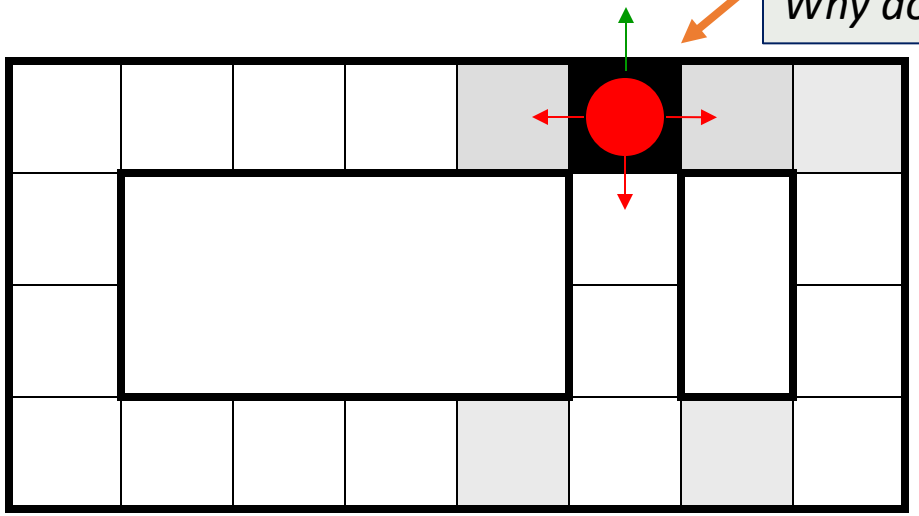
Prob 0 1

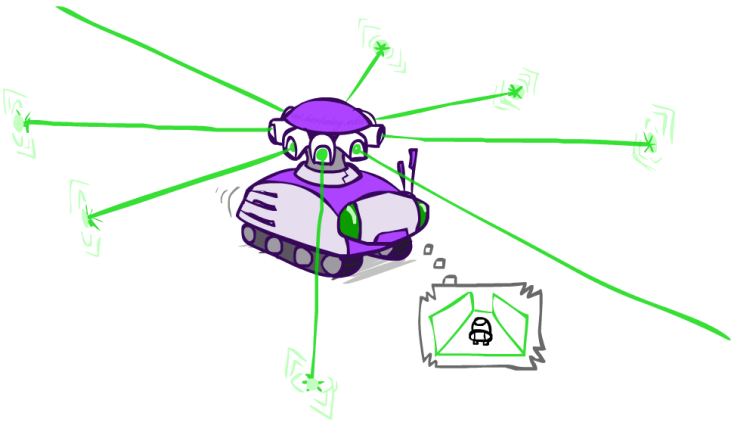
t=4



Example: Robot Localization

The likelihood peaks to a single cell.
Why does the symmetry break?





Prob 0 1

t=5

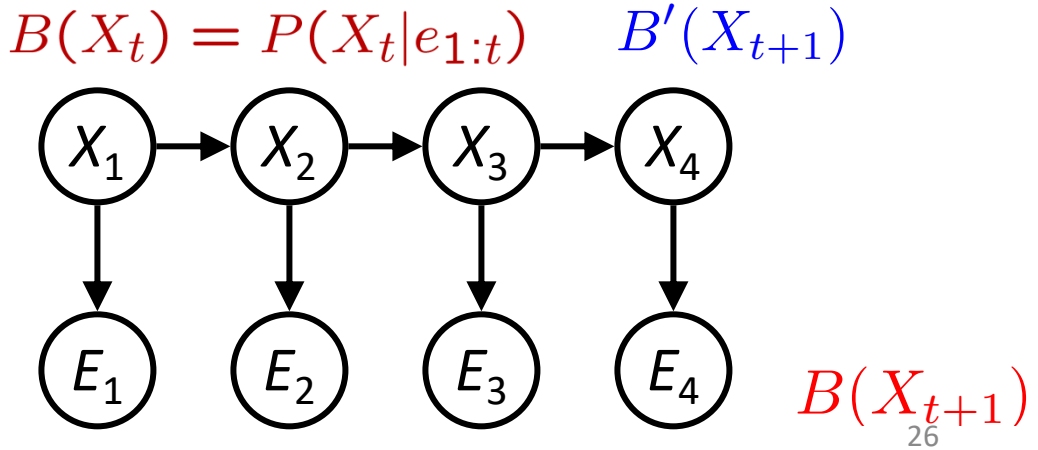
Inference: Estimate State Given Evidence

- We are given evidence at each time and want to know

$$B_t(X) = P(X_t|e_{1:t})$$

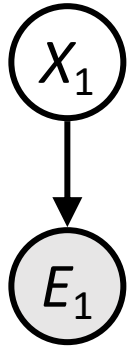
- Approach: start with $P(X_1)$ and derive B_t in terms of B_{t-1}
  - Equivalently, derive B_{t+1} in terms of B_t

- Two Steps:
  - Passage of time
  - Evidence incorporation



Estimating State Given Evidence: Base Cases

- Evidence incorporation
  - Incorporating noisy observations of the state.

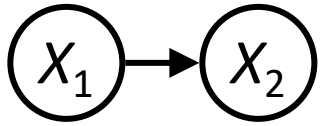


$$P(X_1|e_1)$$

$$P(X_1|e_1) = \frac{P(X_1, e_1)}{\sum_{x_1} P(x_1, e_1)}$$

$$P(X_1|e_1) = \frac{P(e_1|X_1)P(X_1)}{\sum_{x_1} P(e_1|x_1)P(x_1)}$$

- Passage of time
  - The system state at the next time step given transition model



$$P(X_2)$$

$$P(X_2) = \sum_{x_1} P(x_1, X_2)$$

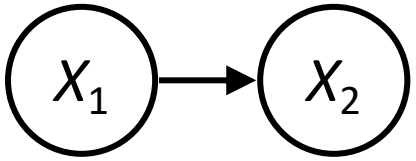
$$P(X_2) = \sum_{x_1} P(X_2|x_1)P(x_1)$$

Next, perform these two computations repeatedly over each time step

Passage of Time

Assume we have current belief P(X | evidence to date)

$$B(X_t) = P(X_t|e_{1:t})$$



Then, after one time step:

$$P(X_{t+1}|e_{1:t}) = \sum_{x_t} P(X_{t+1}, x_t|e_{1:t})$$
$$= \sum_{x_t} P(X_{t+1}|x_t, e_{1:t})P(x_t|e_{1:t})$$
$$= \sum_{x_t} P(X_{t+1}|x_t)P(x_t|e_{1:t})$$

Introduce the state at the previous time step.

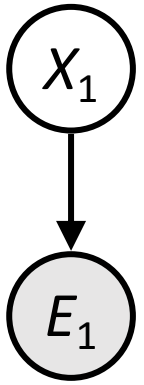
Account for (via marginalization) the likelihood of each value and the likelihood of transition to arrive at the current value.

Basic idea: the beliefs get "pushed" through the transitions

Incorporating Observations

Assume we have current belief P(X | previous evidence):

$$B'(X_{t+1}) = P(X_{t+1}|e_{1:t})$$



Then, after evidence comes in:

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

Given the observation update the likelihood of the state.

Invoke Bayes Rule.

View it as a "correction" of the belief using the observation

$$B(X_{t+1}) \propto_{X_{t+1}} P(e_{t+1}|X_{t+1})B'(X_{t+1})$$

Inference: Weather HMM

Passage of time and correction at each stage.

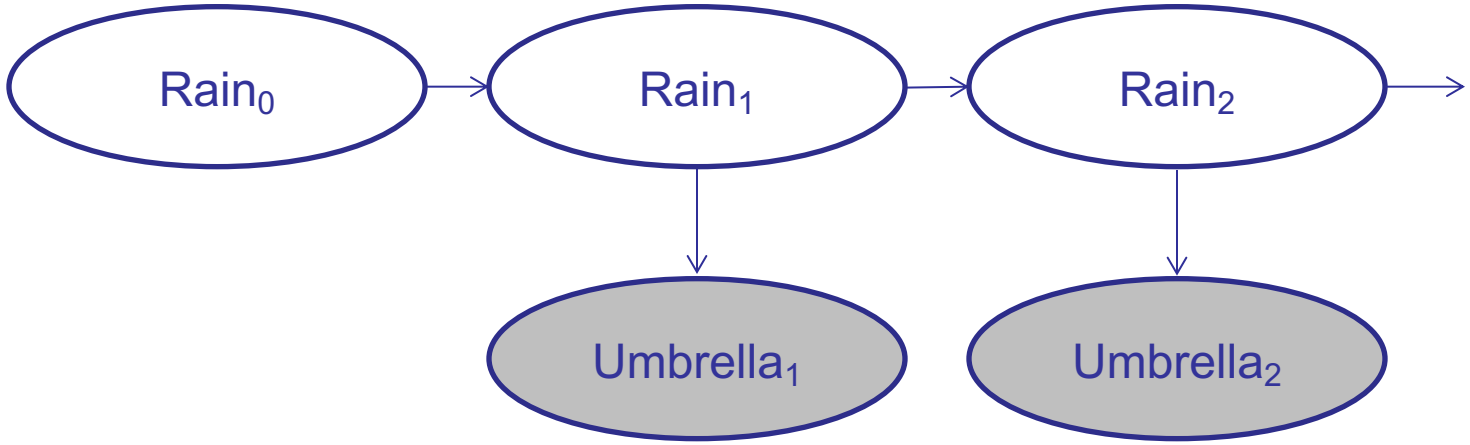
B(+r) = 0.5
B(-r) = 0.5

B'(+r) = 0.5
B'(-r) = 0.5

B(+r) = 0.818
B(-r) = 0.182

B'(+r) = 0.627
B'(-r) = 0.373

B(+r) = 0.883
B(-r) = 0.117



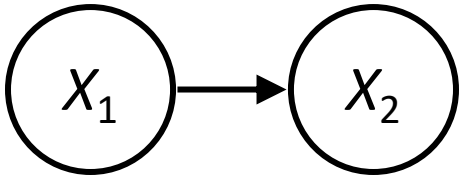
R_t	R_{t+1}	$P(R_{t+1} \mid R_t)$
+r	+r	0.7
+r	-r	0.3
-r	+r	0.3
-r	-r	0.7

R_t	U_t	$P(U_t \mid R_t)$
+r	+u	0.9
+r	-u	0.1
-r	+u	0.2
-r	-u	0.8

Recursive structure to the computation

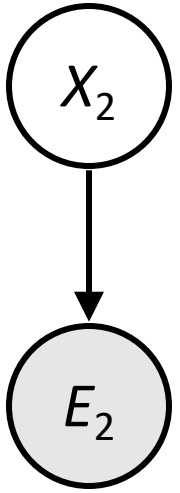
- At each time, propagate the belief forward and correct with observations.
- Every time step, we start with current P(X | evidence)
- We update for time:

$$P(x_t|e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1}|e_{1:t-1}) \cdot P(x_t|x_{t-1})$$



- We update for evidence:

$$P(x_t|e_{1:t}) \propto_X P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$$



- Works online

Inference

We are given evidence at each time and want to know

$$B_t(X) = P(X_t|e_{1:t})$$

We can derive the following updates

$$\begin{aligned} P(x_t|e_{1:t}) &\propto_{X_t} P(x_t, e_{1:t}) \\ &= \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t}) \\ &= \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1}) P(x_t|x_{t-1}) P(e_t|x_t) \\ &= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) P(x_{t-1}, e_{1:t-1}) \end{aligned}$$

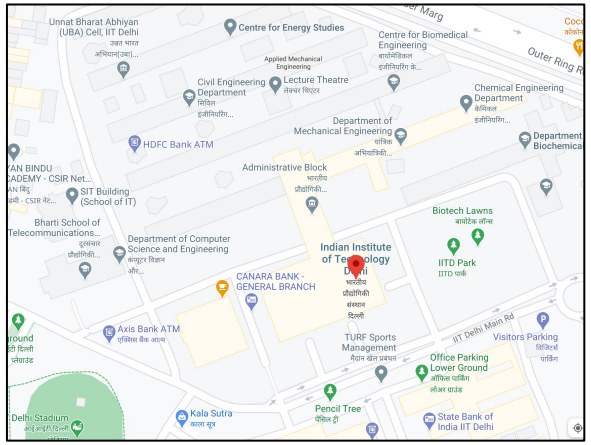
Normalization can be at each step if the exact likelihood is needed at each step or at the end.

Q: what happens if there are no observations at a time step?

What if the state space is “really” large?

Problem: Sometimes |X| is too big to use exact inference

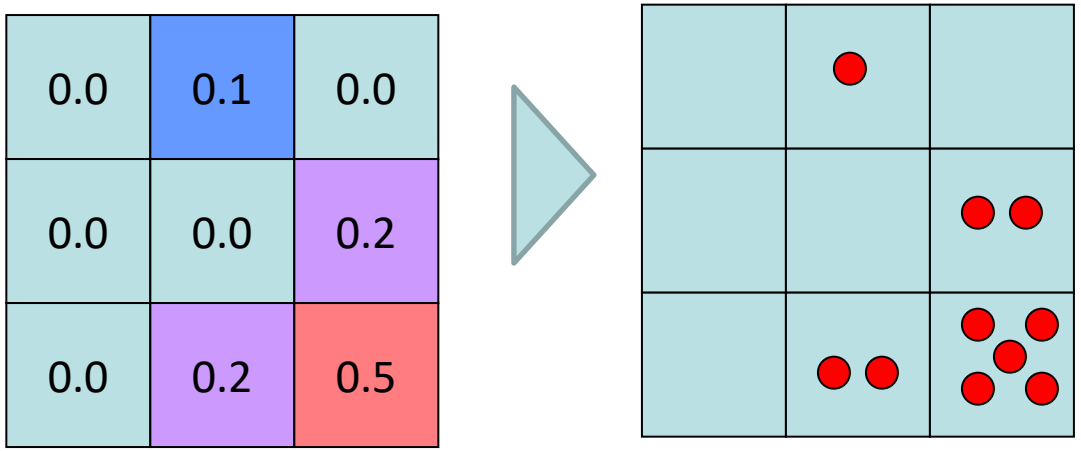
- Example: grid cells may be too many
- |X| may be too big to even store B(X)
- E.g. X is continuous (though here we focus on the discrete case)



Problem: localize the agent in a grid which is “city scale”. There are too many states! Representing B(X) will be challenging.

Solution: approximate inference

- Track samples of X, not all values.
- Samples are called “particles”
- Time spent per step is linear in the number of samples
- Keep the list of particles in memory, not states
- Larger the number of particles, the better is the approximation.

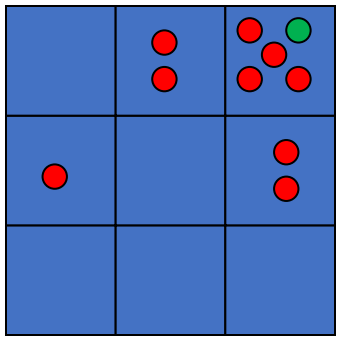


Instead of representing a probability for each state, represent only a constant number of particles.

Representing Belief using Particles

- Our representation of P(X) is now a list of N particles (samples)
  - Generally, N << |X|

- P(x) approximated by number of particles with value x
  - Several x can have P(x) = 0. Note that (3,3) has half the number of particles.
  - Larger the number of particles, better is the approximation.



Particles:
(3,3)
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(1,2)
(3,3)
(3,3)
(2,3)

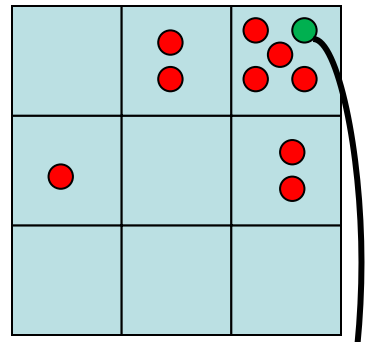
Representation: Passage of Time

Each particle is moved by sampling its next position from the transition model

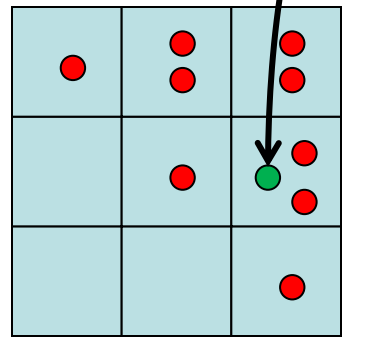
$$x' = \text{sample}(P(X'|x))$$

- Perform simulation or sampling
  - The samples' frequencies reflect the transition probabilities
- In the example, most samples move clockwise, but some move in another direction or stay in place.
  - This is an outcome of the probabilistic transition model.

Particles:
(3,3)
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(1,2)
(3,3)
(3,3)
(2,3)



Particles:
(3,2)
(2,3)
(3,2)
(3,1)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(2,2)



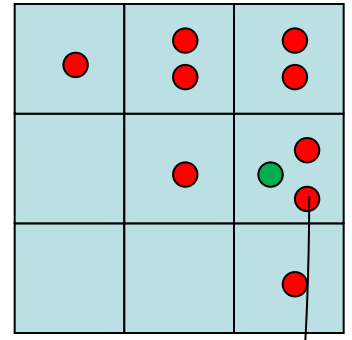
Representation: Incorporate Evidence

- Observation: the agent can measure which grid cell it is in. The measurement is noisy.
- How to adjust the likelihood for each particle?
- As seen previously, incorporating evidence adjusts or weighs the probabilities.
- Attach a weight to each sample. Weigh the samples based on the likelihood of the evidence.

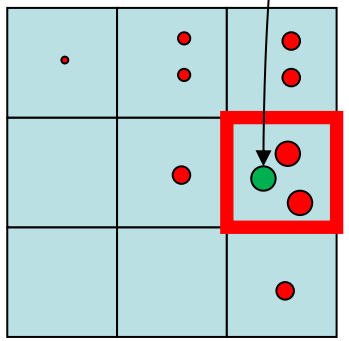
$$w(x) = P(e|x)$$

$$B(X) \propto P(e|X)B'(X)$$

Particles:
(3,2)
(2,3)
(3,2)
(3,1)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(2,2)



Particles:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4

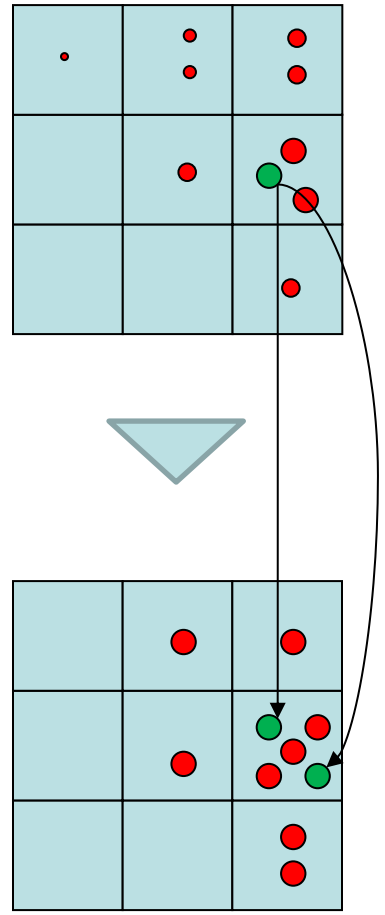


Representation: Resample

- Rather than tracking weighted samples, we resample
- N times, we choose from our weighted sample distribution (i.e. draw with replacement)
- Now the update is complete for this time step, continue with the next one

Particles:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4

(New) Particles:
(3,2)
(2,2)
(3,2)
(2,3)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(3,2)

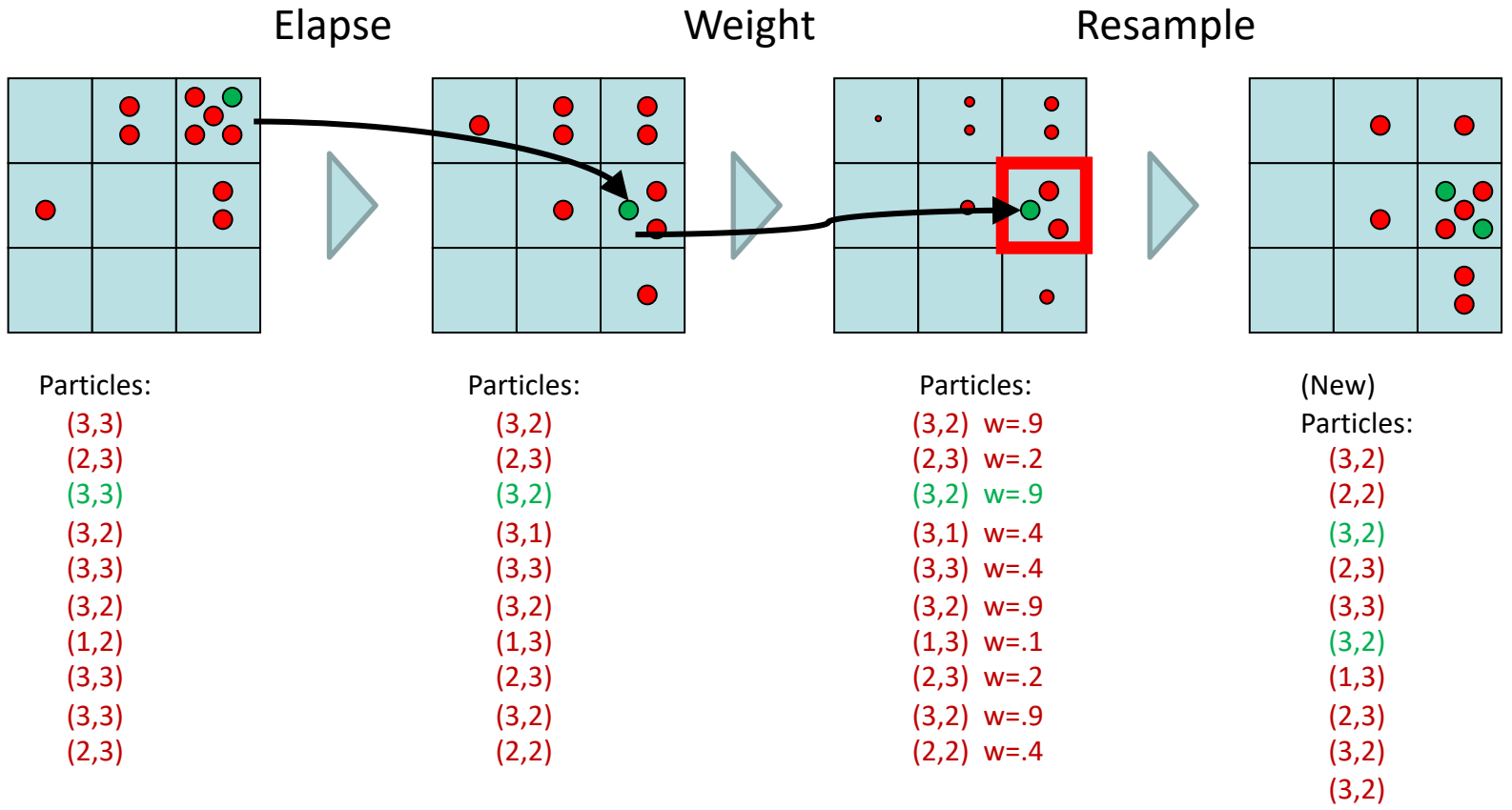


Particle Filtering

Summary

- Each sample is propagated forward by sampling the next state value given the current state value for the sample.
- Each sample is weighted by the likelihood it assigns to the new observation.
- The population is re-sampled to generate a new population of N samples (probability proportional to weight). The new samples are unweighted.

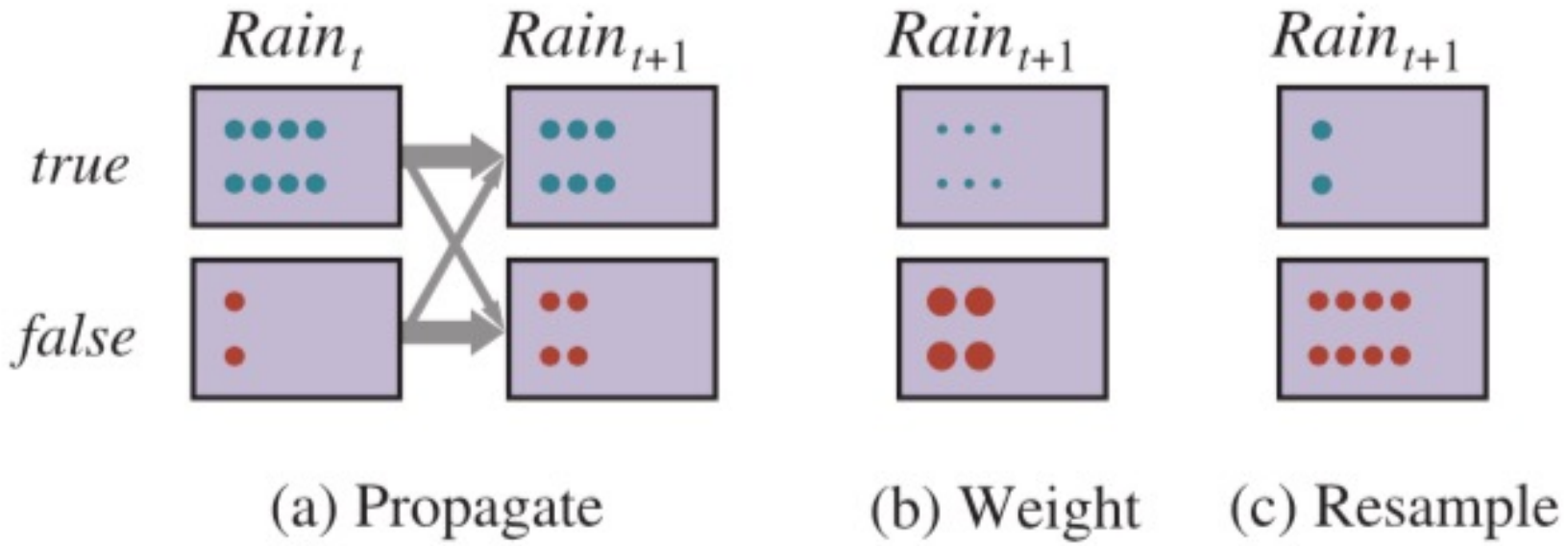
Particles: track samples of states rather than an explicit distribution



In essence: particles represent the “belief” over the state.

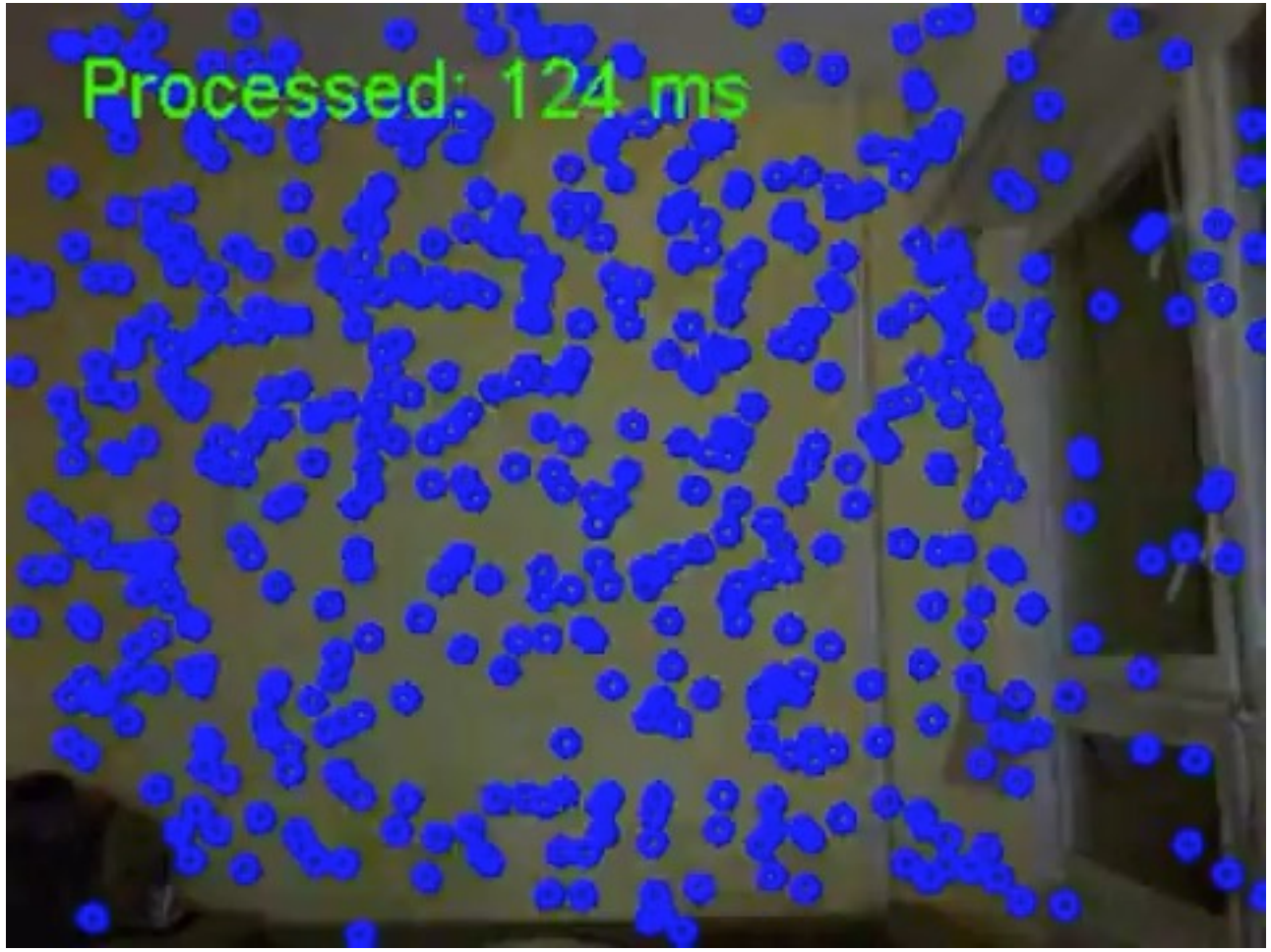
Example:

Weather HMM: Estimating rainy or sunny with umbrella observations. using Particle Filtering.



Note: The observation is not-umbrella in (b). More particles shift to the state corresponding to not rain state.

Particle Filtering Application: Tracking



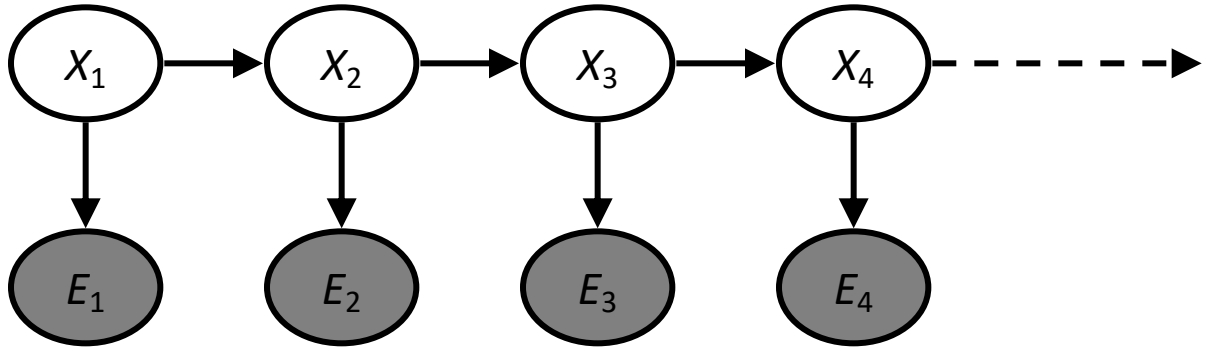
Application: tracking of a red pen. The blue dots indicate the estimated positions.
Video: https://www.youtube.com/watch?v=SV6CmEha51k

Particle Filtering Application: Localization

Other Queries: *Most Likely Explanation*

HMMs defined by

- States X
- Observations E
- Initial distribution: $P(X_1)$
- Transitions: $P(X|X_{-1})$
- Emissions: $P(E|X)$



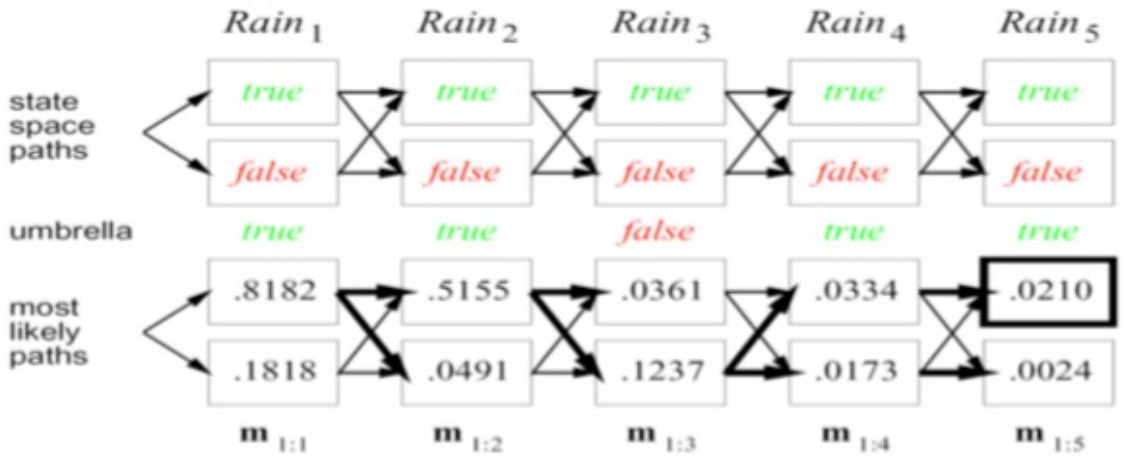
Problem: Most-likely Explanation $\arg\max_{x_{1:t}} P(x_{1:t}|e_{1:t})$

Determine the most likely sequence of states given all the evidence.

Solution: the Viterbi algorithm

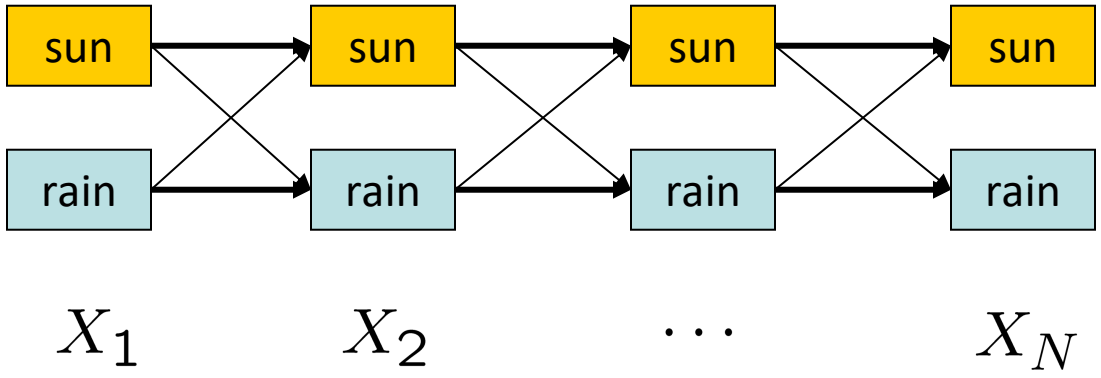
Viterbi Algorithm

- **Observation sequence**: [T, T, F, T, T]
  - Want to determine S/R for *each* of the five days.
  - There are 2^5 sequences possible.
- **State Trellis**
  - View each sequence as a path through a graph whose nodes are the possible states at each time step.



State Trellis

Graph of states and transitions over time



Each arc represents some transition $x_{t-1} \rightarrow x_t$

Each arc has weight $P(x_t|x_{t-1})P(e_t|x_t)$

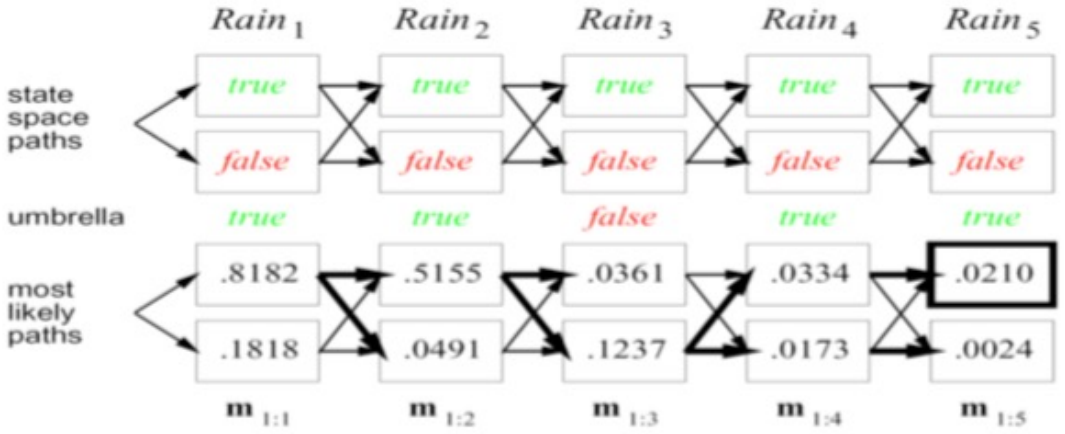
Each path is a sequence of states

The product of weights on a path is that sequence's probability along with the evidence

The most likely explanation query – is like finding the best path in this structure.

Viterbi Algorithm

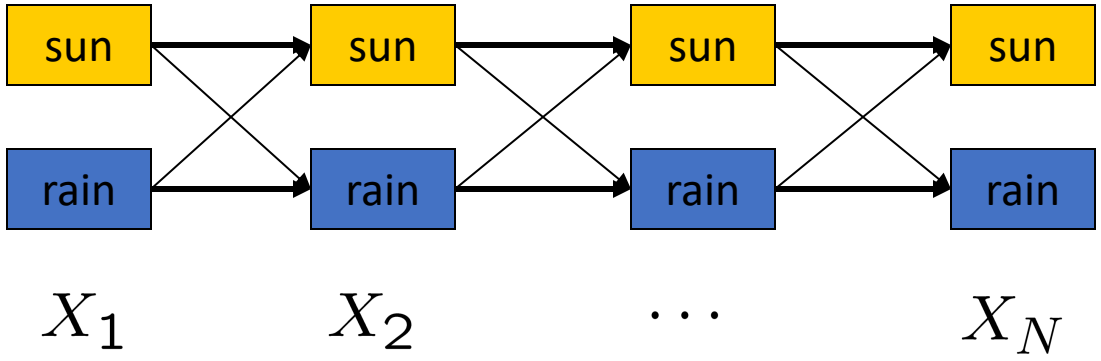
- **Most likely path**
  - Edge is a product of transition probability and the probability of generating the observation.
- **Note**
  - Most likely path to reach $Rain_5$ = True is the most likely path to "some" state at time t=4 and then a transition to $Rain_5$ = True
- **Core Idea**
  - A recursive relation exists between the most likely path to each state in X_{t+1} and the most likely path to each state in X_t



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

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

Viterbi Algorithm



Forward Algorithm (Sum)

$$f_t[x_t] = P(x_t, e_{1:t})$$

$$= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) f_{t-1}[x_{t-1}]$$

Viterbi Algorithm (Max)

$$m_t[x_t] = \max_{x_{1:t-1}} P(x_{1:t-1}, x_t, e_{1:t})$$

$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) m_{t-1}[x_{t-1}]$$

Refer to Russell and Norvig Chapter on Probabilistic Reasoning over Time for Derivation.

Slide: not covered in class. Included here for completion.

Viterbi Algorithm

Viterbi Algorithm

Note: there is a max in the estimation.

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

$$\begin{aligned}
\mathbf{m}_{1:t+1} &= \max_{\mathbf{x}_{1:t}} \mathbf{P}(\mathbf{x}_{1:t}, \mathbf{X}_{t+1}, \mathbf{e}_{1:t+1}) = \max_{\mathbf{x}_{1:t}} \mathbf{P}(\mathbf{x}_{1:t}, \mathbf{X}_{t+1}, \mathbf{e}_{1:t}, e_{t+1}) \\
&= \max_{\mathbf{x}_{1:t}} \mathbf{P}(e_{t+1} | \mathbf{x}_{1:t}, \mathbf{X}_{t+1}, \mathbf{e}_{1:t}) \mathbf{P}(\mathbf{x}_{1:t}, \mathbf{X}_{t+1}, \mathbf{e}_{1:t}) \\
&= \mathbf{P}(e_{t+1} | \mathbf{X}_{t+1}) \max_{\mathbf{x}_{1:t}} \mathbf{P}(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_{1:t}, \mathbf{e}_{1:t}) \\
&= \mathbf{P}(e_{t+1} | \mathbf{X}_{t+1}) \max_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1} | \mathbf{x}_t) \max_{\mathbf{x}_{1:t-1}} P(\mathbf{x}_{1:t-1}, \mathbf{x}_t, \mathbf{e}_{1:t})
\end{aligned}$$

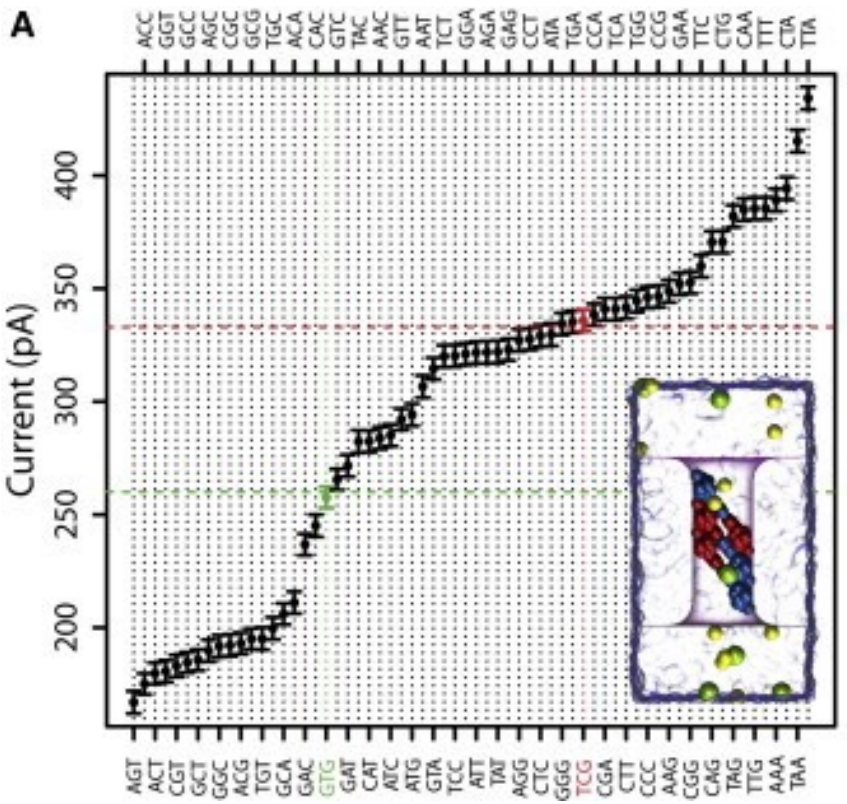
Refer to Russell and Norvig Chapter on Probabilistic Reasoning over Time for Derivation.
AIMA Fourth Edition. Chapter 14. Section 14.2 Page 490.

Viterbi Application: Decoding Genetic Code

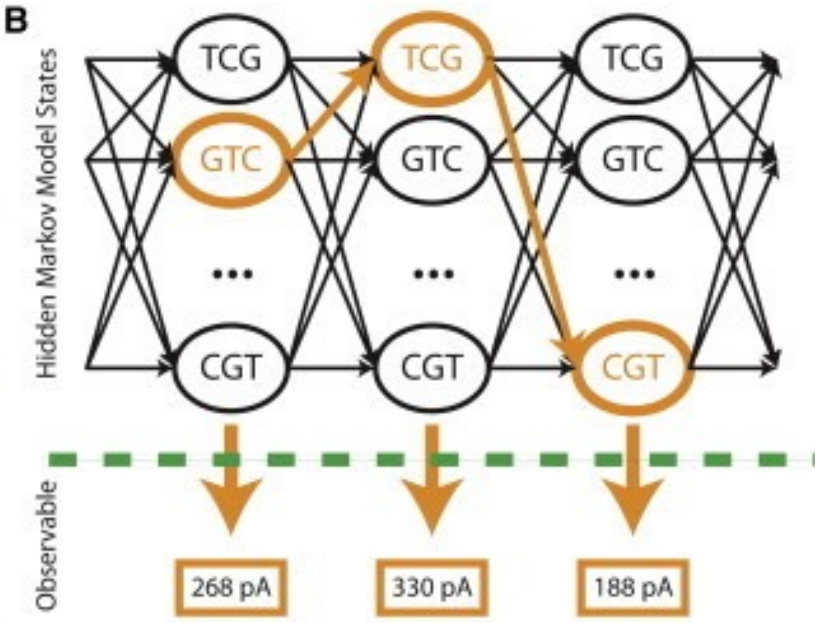
Application to gene sequencing in bio-informatics.

A machine measures current caused by genetic sequences (triplets). The measurement is noisy.

The goal is to predict latent gene sequences from the sequence of current measurements.

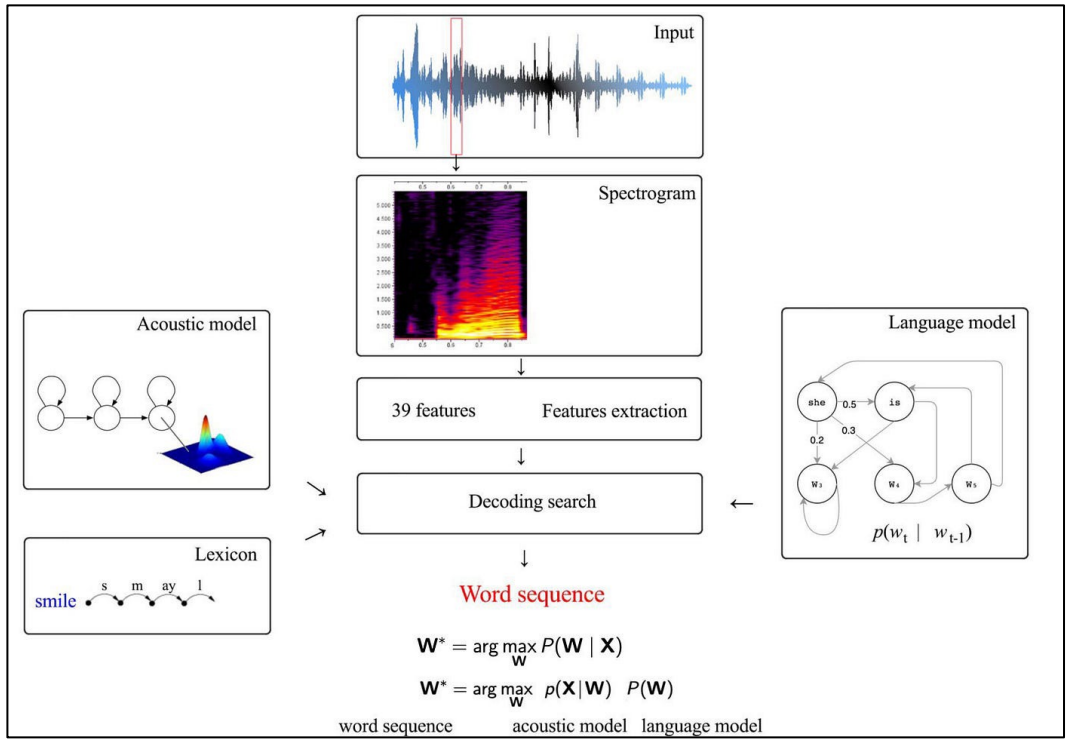


Viterbi algorithm used to identify the most likely latent sequence (yellow).



Source: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3353060/

Viterbi Application: Speech Recognition



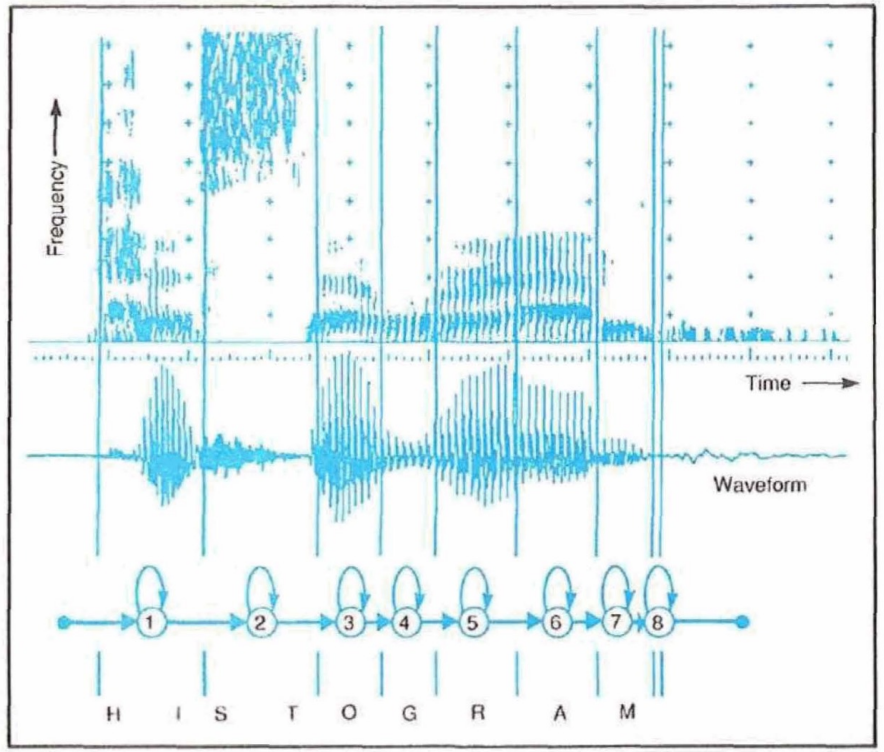


Fig. 3—Viterbi decoder alignment for the word "histogram."

- **Problem:** Given an acoustic signal (spectrogram) infer the words being spoken.
- **Transition Model:** words have a certain likelihood of following (language model).
- **Sensor Model:** Likelihood of a waveform given the word.
- **Inference:** The "sequence of words" from a sequence of waveforms recorded.

Source: https://archive.ll.mit.edu/publications/journal/pdf/vol03_no1/3.1.3.speechrecognition.pdf

HMMs are a special class of Bayes Nets

- HMMs are a class of Bayesian Networks that model uncertainty in state and accumulate observations over time to make inferences.
  - We can analyse their structure via the standard techniques of Bayesian Networks (conditional independencies).
- The specific structure helps in a simple algorithm for estimating the state.
  - Derivation via marginalization and conditioning (standard operations on Bayes Nets).
  - Extends to Most Probable Explanation Queries also.
- HMMs need the Transition and the Observation Model CPTs.
  - There is an algorithm that adapts general EM to the HMM algorithm for estimating the parameters.
  - We did not cover it. But, an interested reader can read here.

Takeaways

- Reasoning over time or space is involved in many practical applications
  - Student engagement, speech recognition, robot localization, bio-informatics etc.
- When the state space is large then it is difficult to maintain the full belief.
  - Maintains a small (constant size) set of belief which are updated with observations as they arrive.
  - Particle filtering applies widely to other inference operations in ML also.
- Connections to other models in AI
  - Bayes Nets that reason over time basis of modern time series models such as RNNs etc.
  - The notion of belief state is used in the (partially-observed) version of MDPs where the state is not known. This model is called POMDPs (pronounced “pomdeepees”).
  - The continuous version of HMMs is Kalman filtering which relies on Gaussian distributions to express the transition and the observation models.