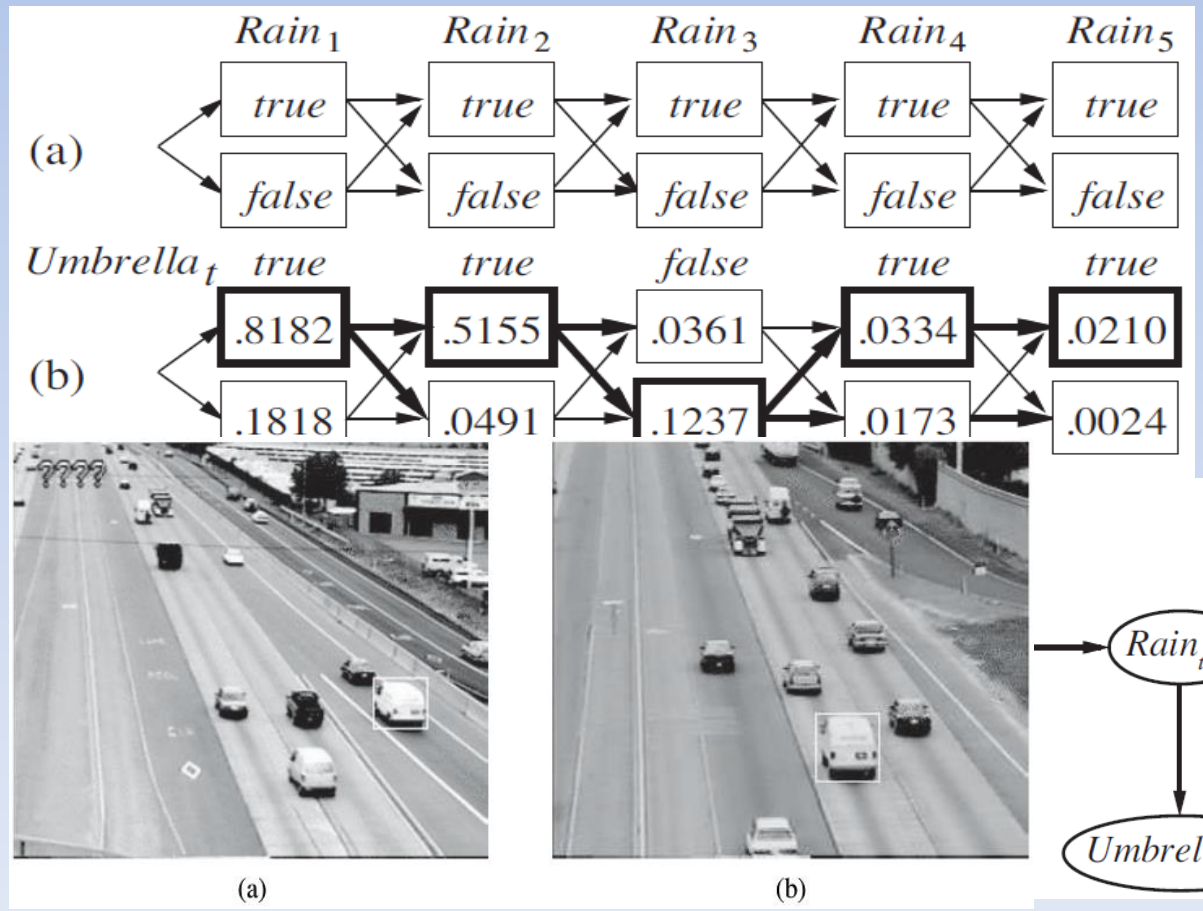
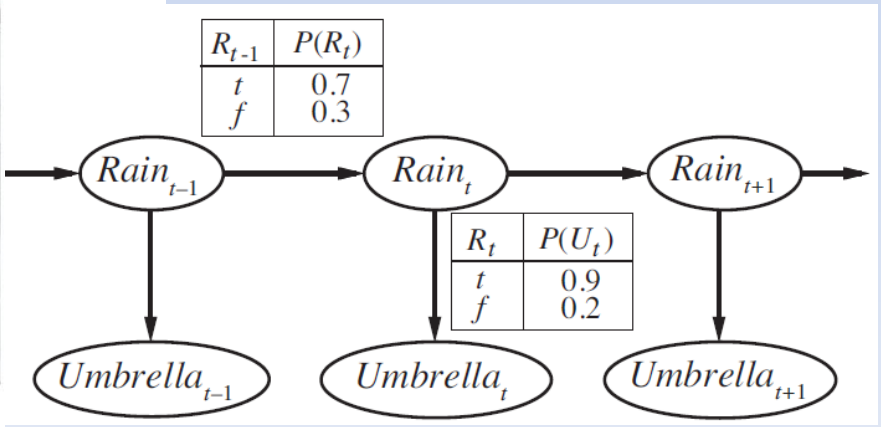
Principles of AI

Chapter 15:
Probabilistic Reasoning over Time





Probability Recap

- Conditional probability

$$P(x|y) = \frac{P(x,y)}{P(y)}$$

- Product rule

$$P(x,y) = P(x|y)P(y)$$

- Chain rule

$$\begin{aligned} P(X_1, X_2, \ldots X_n) &= P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)\ldots \\ &= \prod_{i=1}^{n} P(X_i|X_1, \ldots, X_{i-1}) \end{aligned}$$

- X, Y independent if and only if:

$$\forall x, y : P(x,y) = P(x)P(y)$$

- X and Y are conditionally independent given Z if and only if: $X \perp\!\!\!\perp Y|Z$

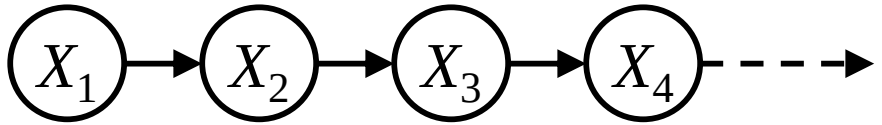
$$\forall x, y, z : P(x,y|z) = P(x|z)P(y|z)$$

Temporal Models

- Reason about world evolving over time.
- System State will be a snapshot of the state of the world.
  - Hidden Variables
    - Disease State, True Speed, True Location
  - Observed Variables
    - Test Results, Sensor Location Data
- State represented as a set of random variables.

Markov Models

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



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

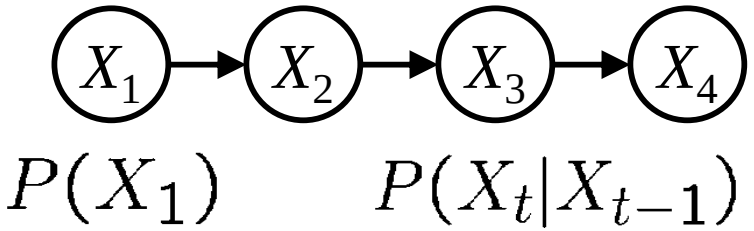
- Parameters: called transition probabilities or dynamics, specify how the state evolves over time (also, initial state probabilities)
- Stationarity assumption: transition probabilities the same at all times
- Same as MDP transition model, but no choice of action

Andrey (Andrei) Andreyevich Markov

- Markov Assumption:
- The FUTURE is INDEPENDENT of the PAST given the PRESENT.

Born	14 June 1856 N.S. Ryazan, Russian Empire
Died	20 July 1922 (aged 66) Petrograd, Russian SFSR
Residence	Russia
Nationality	Russian
Fields	Mathematician
Institutions	St. Petersburg University
Alma mater	St. Petersburg University
Doctoral advisor	Pafnuty Chebyshev
Doctoral students	Abram Besicovitch Nikolai Günther Veniamin Kagan Jacob Tamarkin J. V. Uspensky Georgy Voronoy
Known for	Markov chains; Markov processes; stochastic

Joint Distribution of a Markov Model



- Joint distribution:

$$P(X_1, X_2, X_3, X_4) = P(X_1)P(X_2|X_1)P(X_3|X_2)P(X_4|X_3)$$

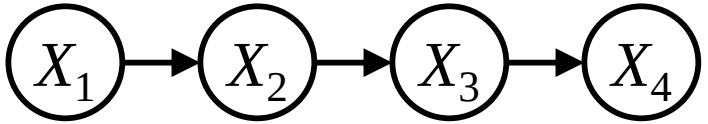
- More generally:

$$P(X_1, X_2, \ldots, X_T) = P(X_1)P(X_2|X_1)P(X_3|X_2)\ldots P(X_T|X_{T-1})$$

$$= P(X_1)\prod_{t=2}^{T} P(X_t|X_{t-1})$$

- Questions to be resolved:
  - Does this indeed define a joint distribution?
  - Can every joint distribution be factored this way, or are we making some assumptions about the joint distribution by using this factorization?

Chain Rule and Markov Models



- From the chain rule, every joint distribution over X_1, X_2, X_3, X_4 can be written as:

$$P(X_1, X_2, X_3, X_4) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)P(X_4|X_1, X_2, X_3)$$

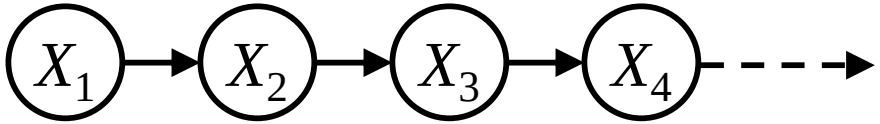
- Assuming that

$$X_3 \perp\!\!\!\perp X_1 \mid X_2 \text{ and } \quad X_4 \perp\!\!\!\perp X_1, X_2 \mid X_3$$

results in the expression posited on the previous slide:

$$P(X_1, X_2, X_3, X_4) = P(X_1)P(X_2|X_1)P(X_3|X_2)P(X_4|X_3)$$

Chain Rule and Markov Models



- From the chain rule, every joint distribution over $X_1, X_2, \ldots, X_T$ can be written as:

$$P(X_1, X_2, \ldots, X_T) = P(X_1) \prod_{t=2}^{T} P(X_t | X_1, X_2, \ldots, X_{t-1})$$

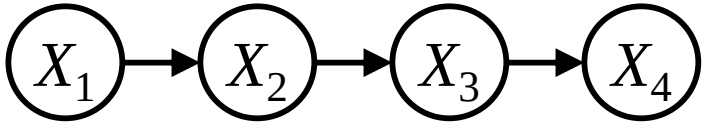
- Assuming that for all *t*:

$$X_t \perp\!\!\!\perp X_1, \ldots, X_{t-2} \mid X_{t-1}$$

gives us the expression posited on the earlier slide:

$$P(X_1, X_2, \ldots, X_T) = P(X_1) \prod_{t=2}^{T} P(X_t | X_{t-1})$$

Implied Conditional Independencies



- We assumed: $X_3 \perp\!\!\!\perp X_1 \mid X_2$ and $X_4 \perp\!\!\!\perp X_1, X_2 \mid X_3$

- Do we also have $X_1 \perp\!\!\!\perp X_3, X_4 \mid X_2$?
  - Yes!
  - Proof:

$$
\begin{aligned}
P(X_1 \mid X_2, X_3, X_4) &= \frac{P(X_1, X_2, X_3, X_4)}{P(X_2, X_3, X_4)} \\
&= \frac{P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_2)P(X_4 \mid X_3)}{\sum_{x_1} P(x_1)P(X_2 \mid x_1)P(X_3 \mid X_2)P(X_4 \mid X_3)} \\
&= \frac{P(X_1, X_2)}{P(X_2)} \\
&= P(X_1 \mid X_2)
\end{aligned}
$$

Markov Models Recap

- Explicit assumption for all t : $X_t \perp\!\!\!\perp X_1, \ldots, X_{t-2} \mid X_{t-1}$
- Consequence, joint distribution can be written as:

$$P(X_1, X_2, \ldots, X_T) = P(X_1)P(X_2|X_1)P(X_3|X_2)\ldots P(X_T|X_{T-1})$$

$$= P(X_1)\prod_{t=2}^{T} P(X_t|X_{t-1})$$

- Implied conditional independencies:
  - Past variables independent of future variables given the present

  i.e., if $t_1 < t_2 < t_3$ or $t_1 > t_2 > t_3$ then: $X_{t_1} \perp\!\!\!\perp X_{t_3} \mid X_{t_2}$
- Additional explicit assumption: $P(X_t \mid X_{t-1})$ is the same for all t

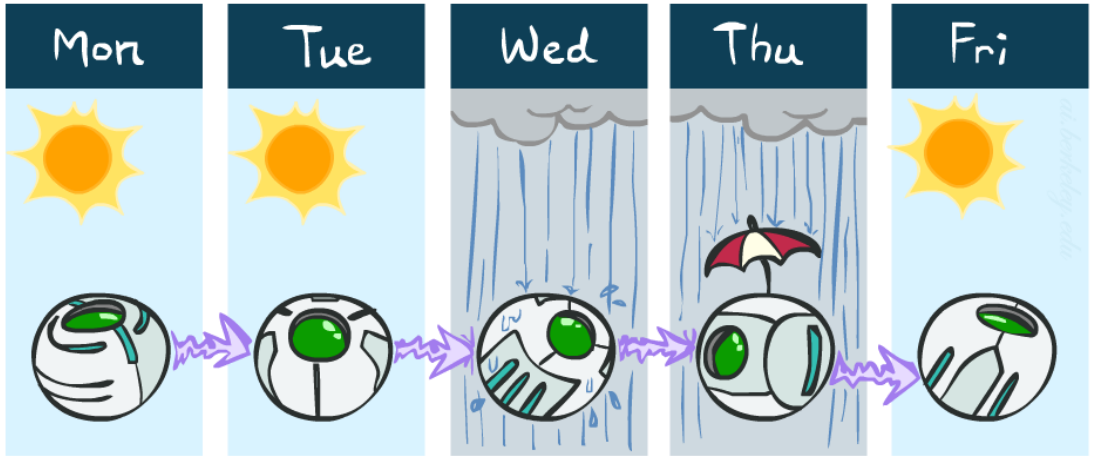
Conditional Independence

- Basic conditional independence:
  - Past and future independent of the present
  - Each time step only depends on the previous
  - This is called the (first order) Markov property

- Note that the chain is just a (growable) BN
  - We can always use generic BN reasoning on it if we truncate the chain at a fixed length

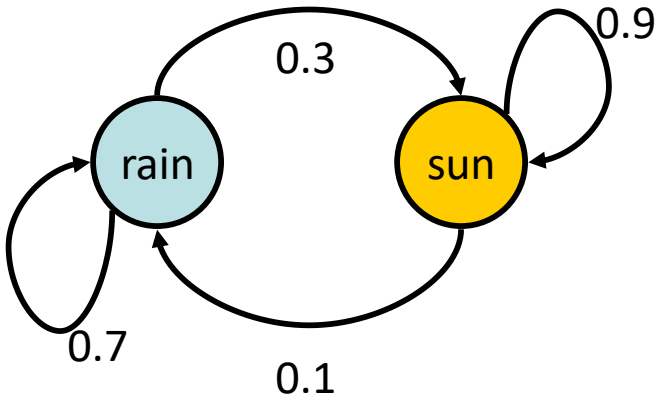
Example Markov Chain: Weather

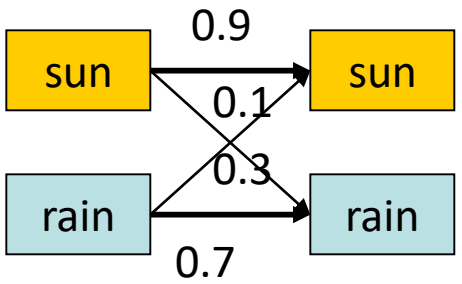
- States: X = {rain, sun}
- Initial distribution: 1.0 sun
- CPT $P(X_t \mid X_{t-1})$:



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

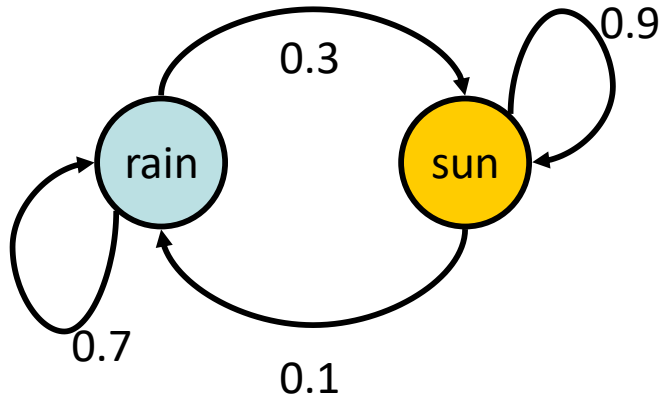
Two new ways of representing the same CPT





Example Markov Chain: Weather

- Initial distribution: 1.0 sun



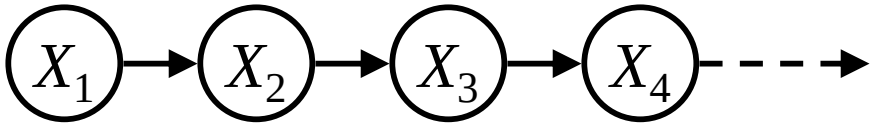
- What is the probability distribution after one step?

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

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

Mini-Forward Algorithm

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

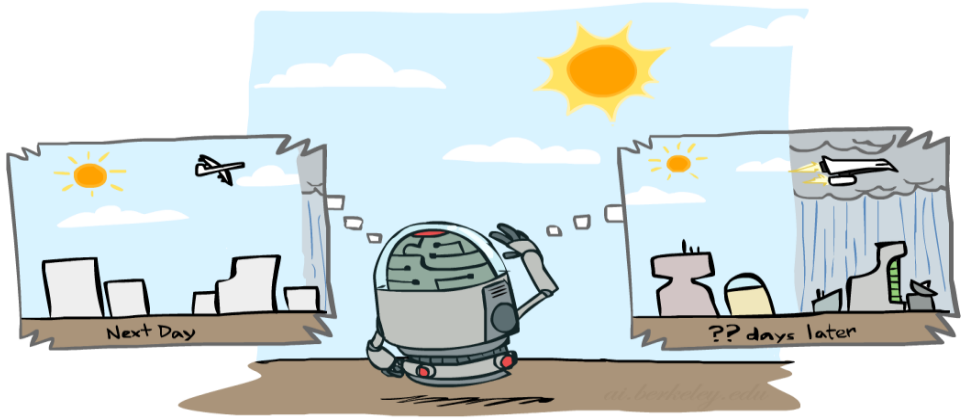


$$P(x_1) = \text{known}$$

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

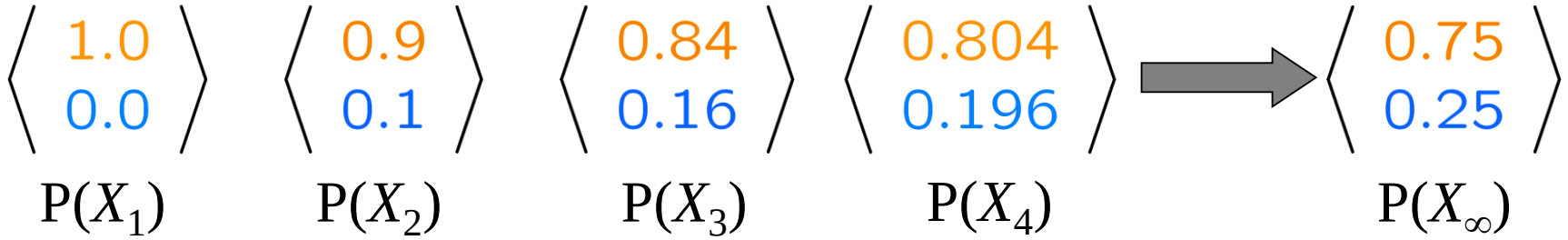
$$= \sum_{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1})$$

Forward simulation

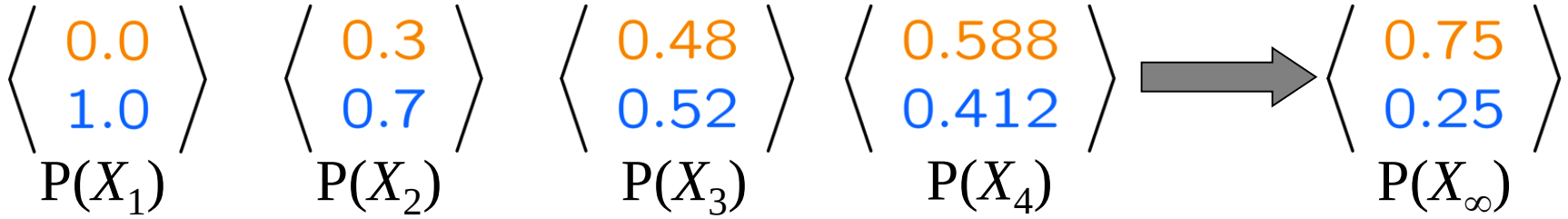


Example Run of Mini-Forward Algorithm

- From initial observation of sun



- From initial observation of rain



- From yet another initial distribution $P(X_1)$:

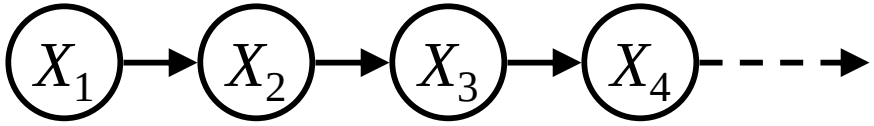
Stationary Distributions

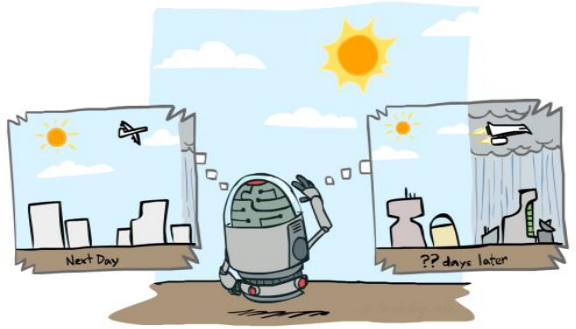
- For most chains:
  - Influence of the initial distribution gets less and less over time.
  - The distribution we end up in is independent of the initial distribution
- Stationary distribution:
  - The distribution we end up with is called the stationary distribution of the chain P_∞
  - It satisfies

$$P_\infty(X) = P_{\infty+1}(X) = \sum_x P(X|x)P_\infty(x)$$

Example: Stationary Distributions

- Question: What's P(X) at time t = infinity?





$$P_\infty(sun) = P(sun|sun)P_\infty(sun) + P(sun|rain)P_\infty(rain)$$
$$P_\infty(rain) = P(rain|sun)P_\infty(sun) + P(rain|rain)P_\infty(rain)$$

$$P_\infty(sun) = 0.9P_\infty(sun) + 0.3P_\infty(rain)$$
$$P_\infty(rain) = 0.1P_\infty(sun) + 0.7P_\infty(rain)$$

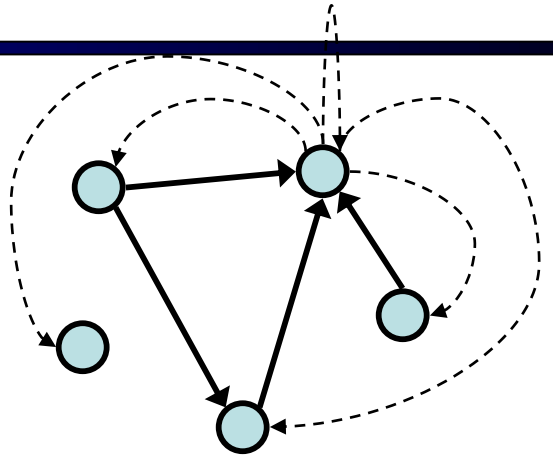
$$P_\infty(sun) = 3P_\infty(rain)$$
$$P_\infty(rain) = 1/3P_\infty(sun)$$

Also: $P_\infty(sun) + P_\infty(rain) = 1$

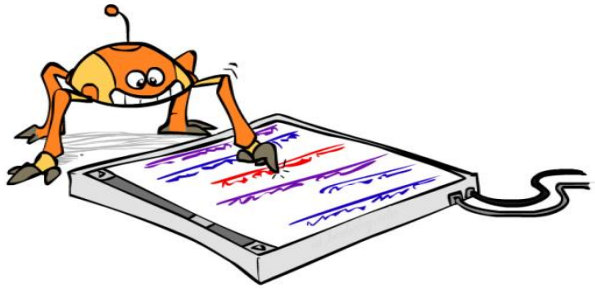
$$P_\infty(sun) = 3/4$$
$$P_\infty(rain) = 1/4$$

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

Application of Stationary Distribution: Web Link Analysis



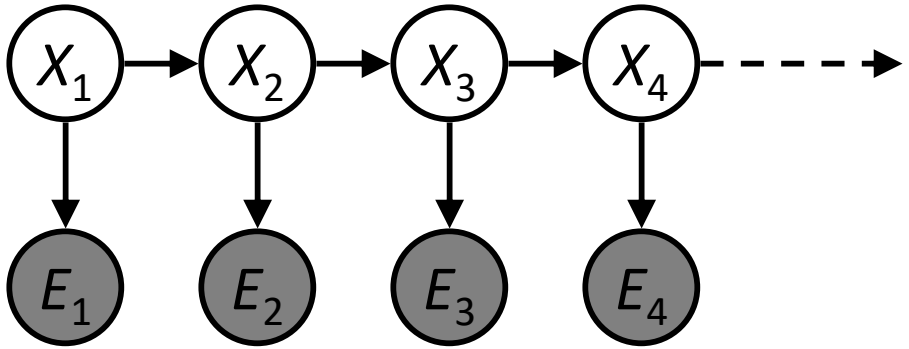
- PageRank over a web graph
  - Each web page is a state
  - Initial distribution: uniform over pages
  - Transitions:
    - With prob. c, uniform jump to a random page (dotted lines, not all shown)
    - With prob. 1-c, follow a random outlink (solid lines)



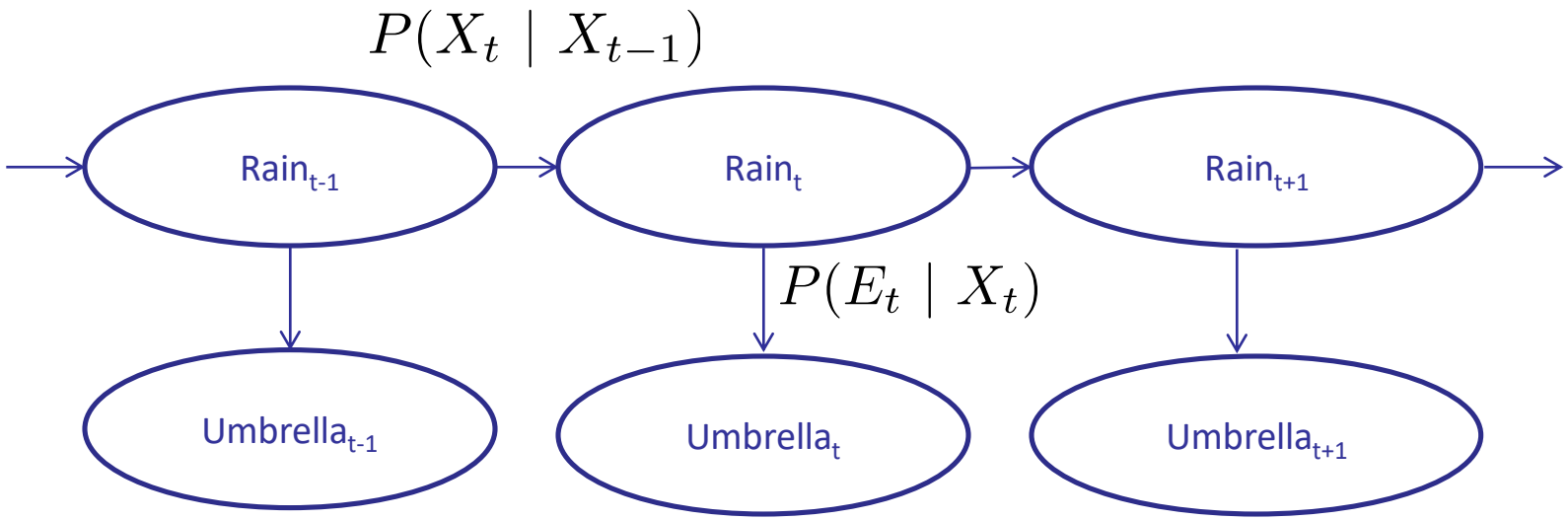
- Stationary distribution
  - Will spend more time on highly reachable pages
  - E.g. many ways to get to the Acrobat Reader download page
  - Somewhat robust to link spam
  - Google 1.0 returned the set of pages containing all your keywords in decreasing rank, now all search engines use link analysis along with many other factors (rank actually getting less important over time)

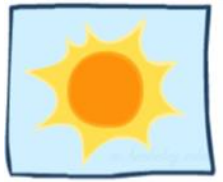
Hidden Markov Models

- Markov chains not so useful for most agents
  - Need observations to update your beliefs
- Hidden Markov models (HMMs)
  - Underlying Markov chain over states X
  - You observe outputs (effects) at each time step



Example: Weather HMM



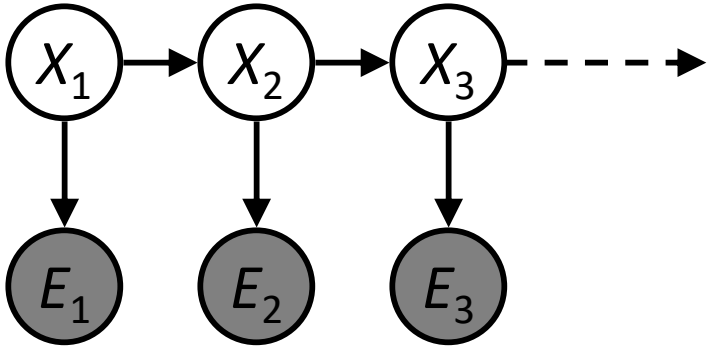


R_t	U_t	$P(U_t \mid R_t)$
+r	+u	0.9
+r	-u	0.1
-r	+u	0.2
-r	-u	0.8

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

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

Joint Distribution of an HMM



- Joint distribution:

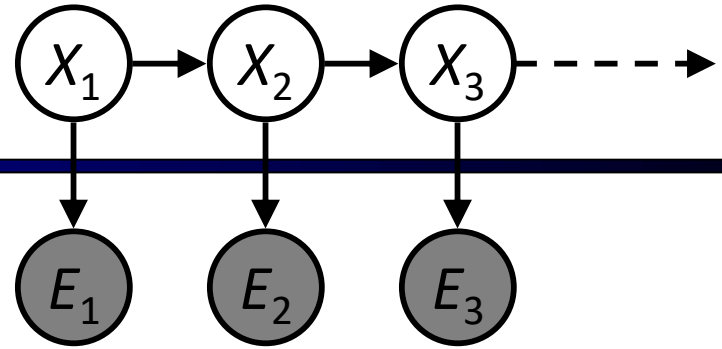
$$P(X_1, E_1, X_2, E_2, X_3, E_3) = P(X_1)P(E_1|X_1)P(X_2|X_1)P(E_2|X_2)P(X_3|X_2)P(E_3|X_3)$$

- More generally:

$$P(X_1, E_1, \ldots, X_T, E_T) = P(X_1)P(E_1|X_1)\prod_{t=2}^{T} P(X_t|X_{t-1})P(E_t|X_t)$$

- Questions to be resolved:
  - Does this indeed define a joint distribution?
  - Can every joint distribution be factored this way, or are we making some assumptions about the joint distribution by using this factorization?

Chain Rule and HMMs



- From the chain rule, *every* joint distribution over $X_1, E_1, X_2, E_2, X_3, E_3$ can be written as:

$$P(X_1, E_1, X_2, E_2, X_3, E_3) = P(X_1)P(E_1|X_1)P(X_2|X_1, E_1)P(E_2|X_1, E_1, X_2)$$
$$P(X_3|X_1, E_1, X_2, E_2)P(E_3|X_1, E_1, X_2, E_2, X_3)$$

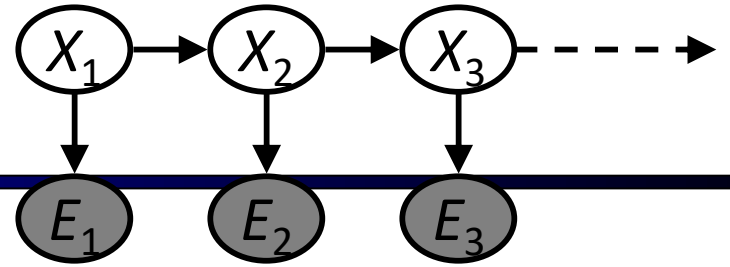
- *Assuming* that

$$X_2 \perp\!\!\!\perp E_1 \mid X_1, \quad E_2 \perp\!\!\!\perp X_1, E_1 \mid X_2, \quad X_3 \perp\!\!\!\perp X_1, E_1, E_2 \mid X_2, \quad E_3 \perp\!\!\!\perp X_1, E_1, X_2, E_2 \mid X_3$$

gives us the expression posited on the previous slide:

$$P(X_1, E_1, X_2, E_2, X_3, E_3) = P(X_1)P(E_1|X_1)P(X_2|X_1)P(E_2|X_2)P(X_3|X_2)P(E_3|X_3)$$

Chain Rule and HMMs



- From the chain rule, *every* joint distribution over $X_1, E_1, \ldots, X_T, E_T$ can be written as:

$$P(X_1, E_1, \ldots, X_T, E_T) = P(X_1)P(E_1|X_1)\prod_{t=2}^{T} P(X_t|X_1, E_1, \ldots, X_{t-1}, E_{t-1})P(E_t|X_1, E_1, \ldots, X_{t-1}, E_{t-1}, X_t)$$

- *Assuming* that for all *t*:
  - State independent of all past states and all past evidence given the previous state, i.e.:

$$X_t \perp\!\!\!\perp X_1, E_1, \ldots, X_{t-2}, E_{t-2}, E_{t-1} \mid X_{t-1}$$

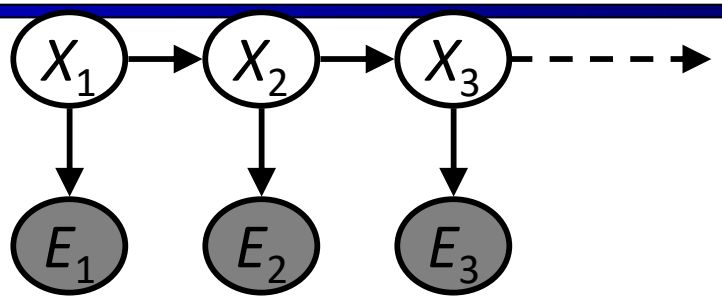
  - Evidence is independent of all past states and all past evidence given the current state, i.e.:

$$E_t \perp\!\!\!\perp X_1, E_1, \ldots, X_{t-2}, E_{t-2}, X_{t-1}, E_{t-1} \mid X_t$$

gives us the expression posited on the earlier slide:

$$P(X_1, E_1, \ldots, X_T, E_T) = P(X_1)P(E_1|X_1)\prod_{t=2}^{T} P(X_t|X_{t-1})P(E_t|X_t)$$

Implied Conditional Independencies



- Many implied conditional independencies, e.g.,

$$E_1 \perp\!\!\!\perp X_2, E_2, X_3, E_3 \mid X_1$$

- To prove them
  - Approach 1: follow similar (algebraic) approach to what we did in the Markov models
  - Approach 2: directly from the graph structure
    - Intuition: If path between U and V goes through W, then $U \perp\!\!\!\perp V \mid W$

HMMs Recap

- Explicit assumption for all t : $X_t \perp\!\!\!\perp X_1, \ldots, X_{t-2} \mid X_{t-1}$
- Consequence, joint distribution can be written as:

$$P(X_1, X_2, \ldots, X_T) = P(X_1)P(X_2|X_1)P(X_3|X_2)\ldots P(X_T|X_{T-1})$$

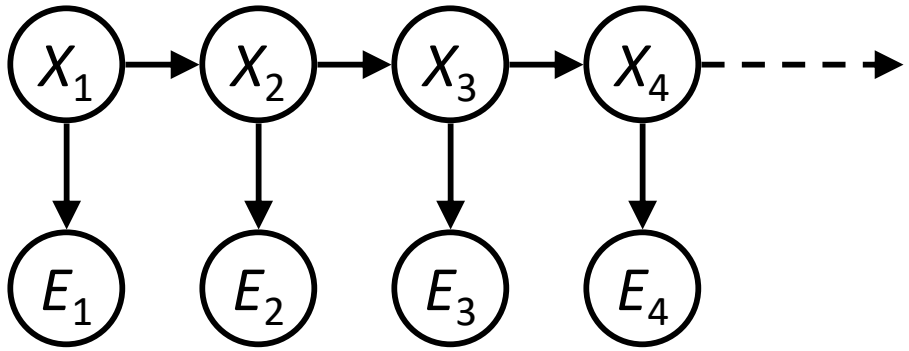
$$= P(X_1)\prod_{t=2}^{T} P(X_t|X_{t-1})$$

- Implied conditional independencies:
  - Past variables independent of future variables given the present

  i.e., if $t_1 < t_2 < t_3$ or $t_1 > t_2 > t_3$ then: $X_{t_1} \perp\!\!\!\perp X_{t_3} \mid X_{t_2}$
- Additional explicit assumption: $P(X_t \mid X_{t-1})$ is the same for all t

Conditional Independence

- HMMs have two important independence properties:
  - Markov hidden process: future depends on past via the present
  - Current observation independent of all else given current state



- Quiz: does this mean that evidence variables are guaranteed to be independent?
  - [No, they tend to correlated by the hidden state]

Real HMM Examples

- Speech recognition HMMs:
  - Observations are acoustic signals (continuous valued)
  - States are specific positions in specific words (so, tens of thousands)
- Machine translation HMMs:
  - Observations are words (tens of thousands)
  - States are translation options
- Robot tracking:
  - Observations are range readings (continuous)
  - States are positions on a map (continuous)

Inference Tasks

- Filtering: $P(X_t | e_{1:t})$
  - 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;
  - like filtering without the evidence
- 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

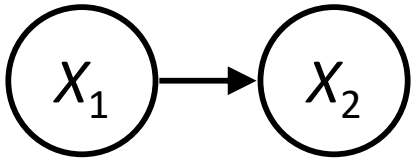
Filtering / Monitoring

- 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)
- The Kalman filter was invented in the 60's and first implemented as a method of trajectory estimation for the Apollo program

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 passes:

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

- Or compactly:

$$B'(X_{t+1}) = \sum_{x_t} P(X'|x_t) B(x_t)$$

- Basic idea: beliefs get "pushed" through the transitions
  - With the "B" notation, we have to be careful about what time step t the belief is about, and what evidence it includes

Example: Passage of Time

- As time passes, uncertainty "accumulates"

(Transition model: ghosts usually go clockwise)

<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	1.00	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

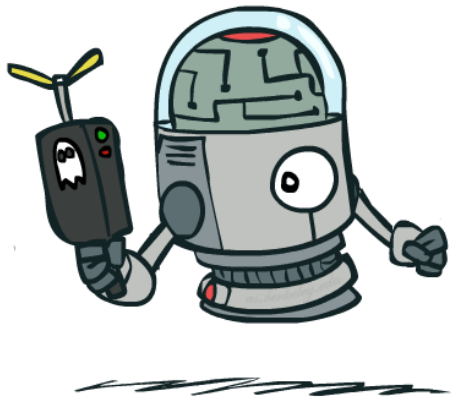
T = 1

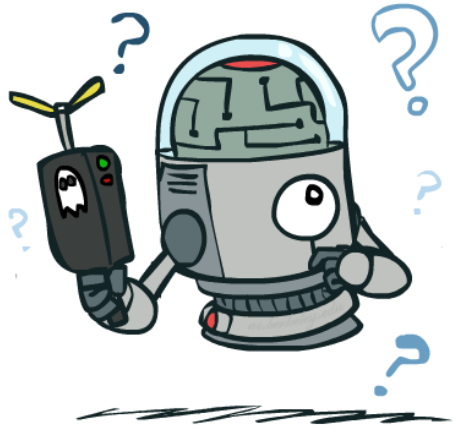
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	0.06	<0.01	<0.01	<0.01
<0.01	0.76	0.06	0.06	<0.01	<0.01
<0.01	<0.01	0.06	<0.01	<0.01	<0.01

T = 2

0.05	0.01	0.05	<0.01	<0.01	<0.01
0.02	0.14	0.11	0.35	<0.01	<0.01
0.07	0.03	0.05	<0.01	0.03	<0.01
0.03	0.03	<0.01	<0.01	<0.01	<0.01

T = 5





Observation

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

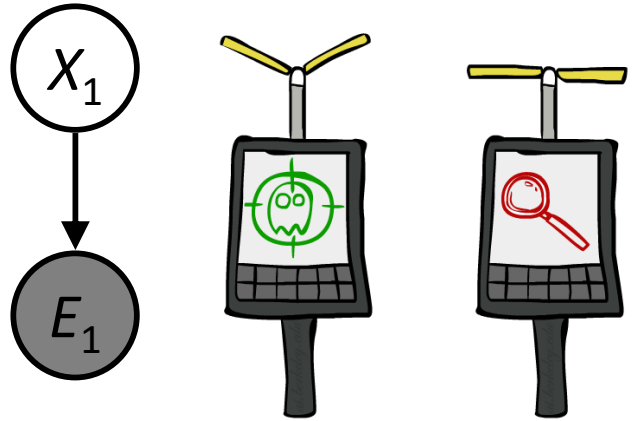
$$B'(X_{t+1}) = P(X_{t+1}|e_{1:t})$$

- Then, after evidence comes in:

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

- Or, compactly:

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



- Basic idea: beliefs "reweighted" by likelihood of evidence
- Unlike passage of time, we have to renormalize

Example: Observation

- As we get observations, beliefs get reweighted, uncertainty "decreases"

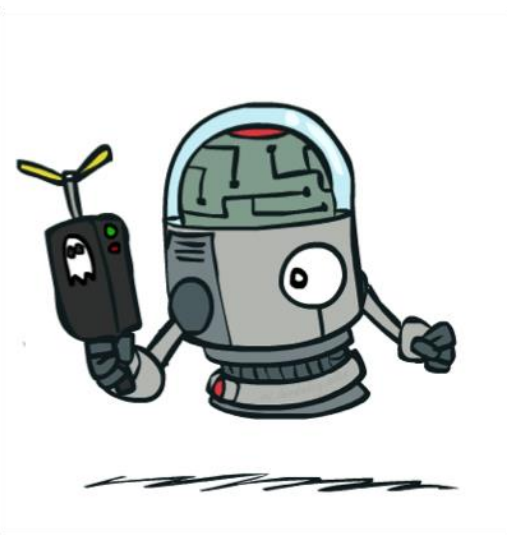
0.05	0.01	0.05	<0.01	<0.01	<0.01
0.02	0.14	0.11	0.35	<0.01	<0.01
0.07	0.03	0.05	<0.01	0.03	<0.01
0.03	0.03	<0.01	<0.01	<0.01	<0.01

Before observation

<0.01	<0.01	<0.01	<0.01	0.02	<0.01
<0.01	<0.01	<0.01	0.83	0.02	<0.01
<0.01	<0.01	0.11	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

After observation

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



Stationary Distribution

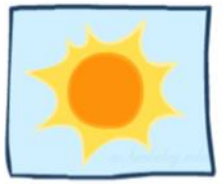
- $P(0|0) = 0.4$, $P(1|0) = 0.6$
- $P(0|1) = 0.8$, $P(1|1) = 0.2$

- Calculate Stationary Distribution

- B(0) = 0.4B(0) + 0.8B(1)
- B(1) = 0.6B(0) + 0.2B(1)

- 0.6B(0) = 0.8B(1)
- B(1) = 6/8(B(0))
- B(0) +6/8(B(0)=1
- B(0) = 8/14

Example: Weather HMM



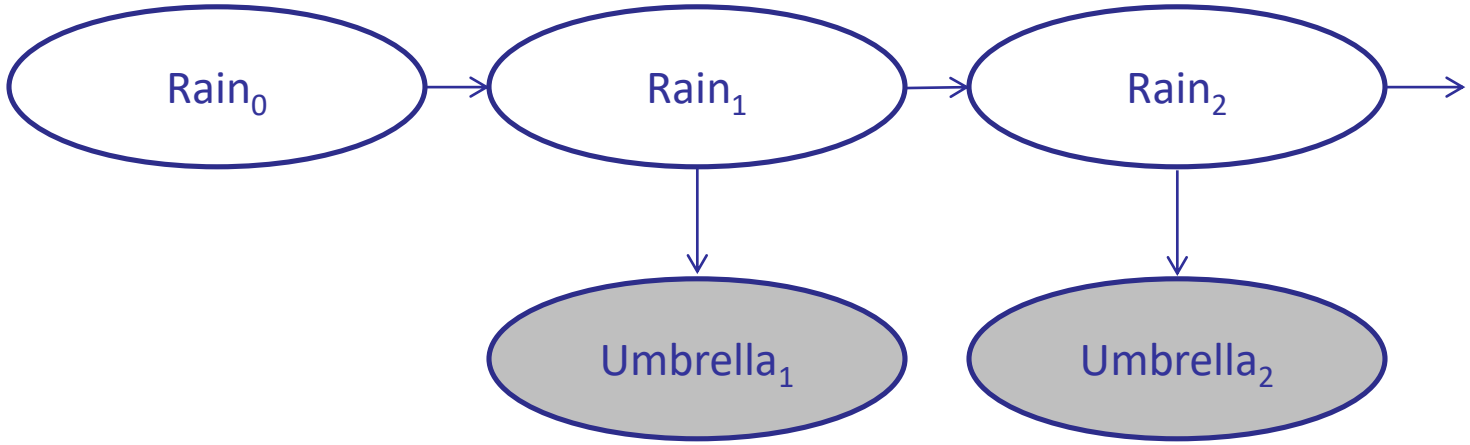
B(+r) = 0.5
B(-r) = 0.5

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

Example: Weather HMM

- $P(0|0) = 0.7$, $P(1|0) = 0.3$
- $P(0|1) = 0.3$, $P(1|1) = 0.7$

- $B(0) = 0.5$
- $B_1(0) = 0.7B(0) + 0.3B(1) = 0.5$
- $B_1(1) = 0.3B(0) + 0.7B(1) = 0.5$

Example: Weather HMM

- $P(u|0) = 0.2$
- $P(u|1) = 0.9$

- $B_1(0) = 0.5$
- $B_1(1) = 0.5$
- $B_1(0) = B_1(0) * 0.2 = 0.1/0.55 = 0.181818$
- $B_1(1) = B_1(1) * 0.9 = 0.45/0.55 = 0.81818$

Example: Weather HMM

B'(+r) = 0.5 B'(+r) = 0.627

Let us illustrate the filtering process for two steps in the basic umbrella example (Figure 15.2.) That is, we will compute $\mathbf{P}(R_2 \mid u_{1:2})$ as follows:

- On day 0, we have no observations, only the security guard's prior beliefs; let's assume that consists of $\mathbf{P}(R_0) = \langle 0.5, 0.5 \rangle$.
- On day 1, the umbrella appears, so $U_1 = true$. The prediction from $t = 0$ to $t = 1$ is

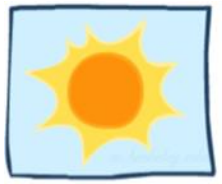
$$\begin{aligned}\mathbf{P}(R_1) &= \sum_{r_0} \mathbf{P}(R_1 \mid r_0) P(r_0) \\ &= \langle 0.7, 0.3 \rangle \times 0.5 + \langle 0.3, 0.7 \rangle \times 0.5 = \langle 0.5, 0.5 \rangle \ .\end{aligned}$$

Then the update step simply multiplies by the probability of the evidence for $t = 1$ and normalizes, as shown in Equation (15.4):

$$\begin{aligned}\mathbf{P}(R_1 \mid u_1) &= \alpha\, \mathbf{P}(u_1 \mid R_1)\mathbf{P}(R_1) = \alpha\, \langle 0.9, 0.2 \rangle \langle 0.5, 0.5 \rangle \\ &= \alpha\, \langle 0.45, 0.1 \rangle \approx \langle 0.818, 0.182 \rangle \ .\end{aligned}$$

$J_t \mid R_t$
0.9
0.1
0.2
0.8

Example: Weather HMM



B'(+r) = 0.5 B'(+r) = 0.627

- On day 2, the umbrella appears, so $U_2 = true$. The prediction from $t = 1$ to $t = 2$ is

$$\begin{aligned}\mathbf{P}(R_2 \,|\, u_1) &= \sum_{r_1} \mathbf{P}(R_2 \,|\, r_1) P(r_1 \,|\, u_1) \\ &= \langle 0.7, 0.3 \rangle \times 0.818 + \langle 0.3, 0.7 \rangle \times 0.182 \approx \langle 0.627, 0.373 \rangle \ ,\end{aligned}$$

and updating it with the evidence for $t = 2$ gives

$$\begin{aligned}\mathbf{P}(R_2 \,|\, u_1, u_2) &= \alpha\, \mathbf{P}(u_2 \,|\, R_2) \mathbf{P}(R_2 \,|\, u_1) = \alpha\, \langle 0.9, 0.2 \rangle \langle 0.627, 0.373 \rangle \\ &= \alpha\, \langle 0.565, 0.075 \rangle \approx \langle 0.883, 0.117 \rangle \ .\end{aligned}$$

-r	-r	0.7

-r	-u	0.8

The Forward Algorithm

- 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

$$P(x_t|e_{1:t}) \propto_X P(x_t, e_{1:t})$$

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

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

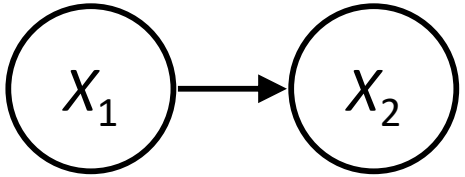
$$= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) P(x_{t-1}, e_{1:t-1})$$

We can normalize as we go if we want to have P(x|e) at each time step, or just once at the end…

Online Belief Updates

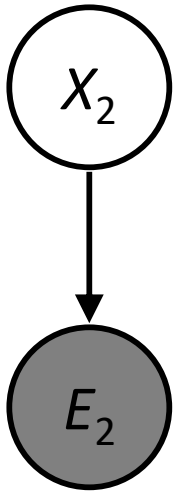
- Every time step, we start with current P(X | evidence)
- We update for time:

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



- We update for evidence:

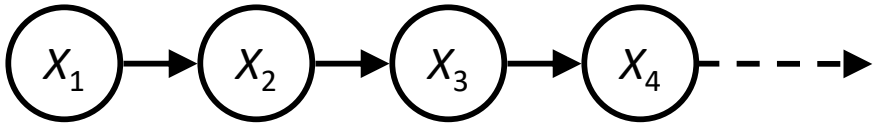
$$P(x_t|e_{1:t}) \propto_X P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$$



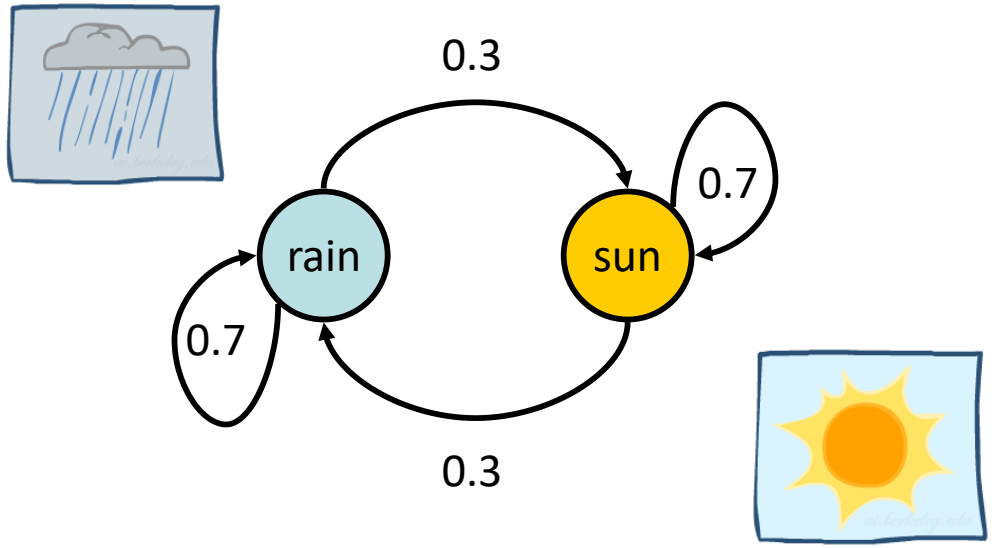
- The forward algorithm does both at once (and doesn't normalize)

Recap: Reasoning Over Time

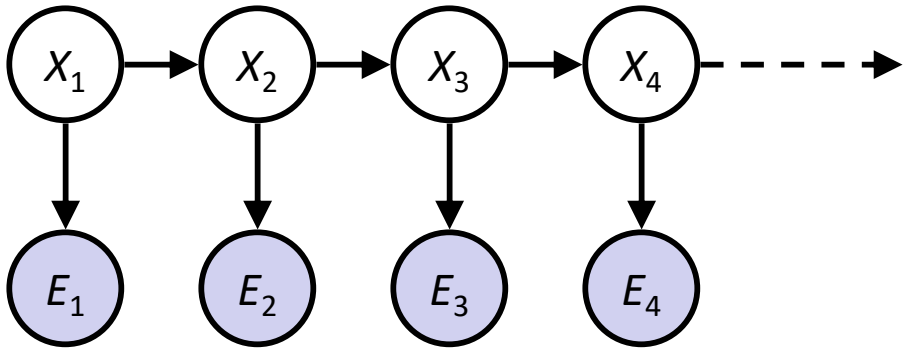
- Markov models



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



- Hidden Markov models



$$P(E|X)$$

X	E	P
rain	umbrella	0.9
rain	no umbrella	0.1
sun	umbrella	0.2
sun	no umbrella	0.8

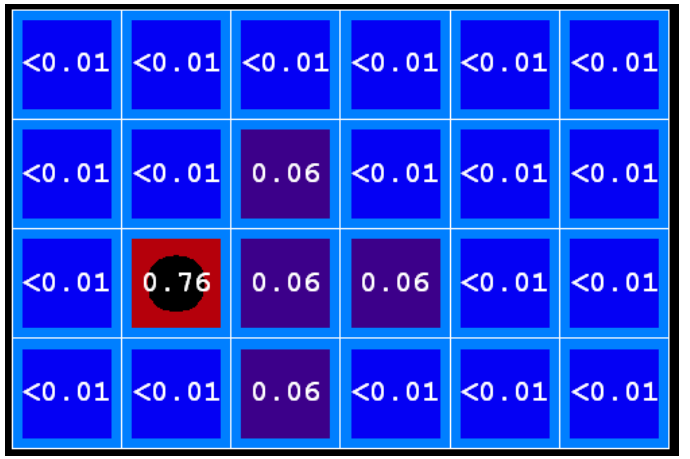
Recap: Filtering

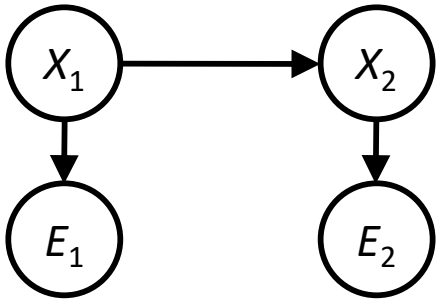
Elapse time: compute P(X_t | $e_{1:t-1}$)

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

Observe: compute P(X_t | $e_{1:t}$)

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



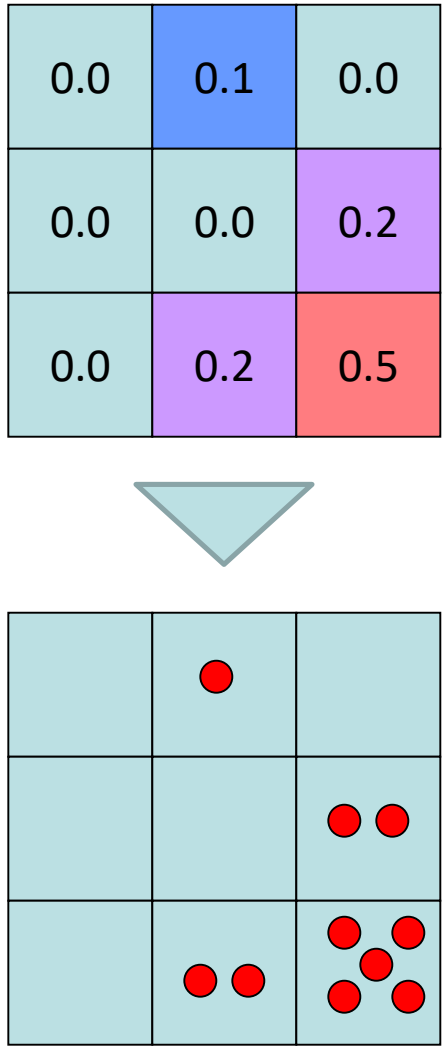


Belief: <P(rain), P(sun)>

$P(X_1)$	<0.5, 0.5>	*Prior on X_1*
$P(X_1 \mid E_1 = umbrella)$	<0.82, 0.18>	*Observe*
$P(X_2 \mid E_1 = umbrella)$	<0.63, 0.37>	*Elapse time*
$P(X_2 \mid E_1 = umb, E_2 = umb)$	<0.88, 0.12>	*Observe*

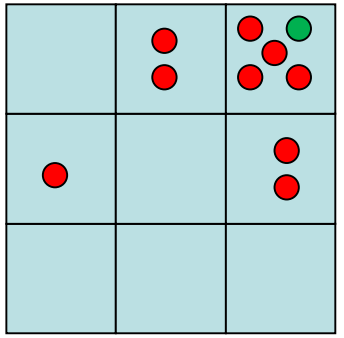
Particle Filtering

- Filtering: approximate solution
- Sometimes |X| is too big to use exact inference
  - |X| may be too big to even store B(X)
  - E.g. X is continuous
- Solution: approximate inference
  - Track samples of X, not all values
  - Samples are called particles
  - Time per step is linear in the number of samples
  - But: number needed may be large
  - In memory: list of particles, not states
- This is how robot localization works in practice
- Particle is just new name for sample



Representation: Particles

- Our representation of P(X) is now a list of N particles (samples)
  - Generally, N << |X|
  - Storing map from X to counts would defeat the point
- P(x) approximated by number of particles with value x
  - So, many x may have P(x) = 0!
  - More particles, more accuracy
- For now, all particles have a weight of 1



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

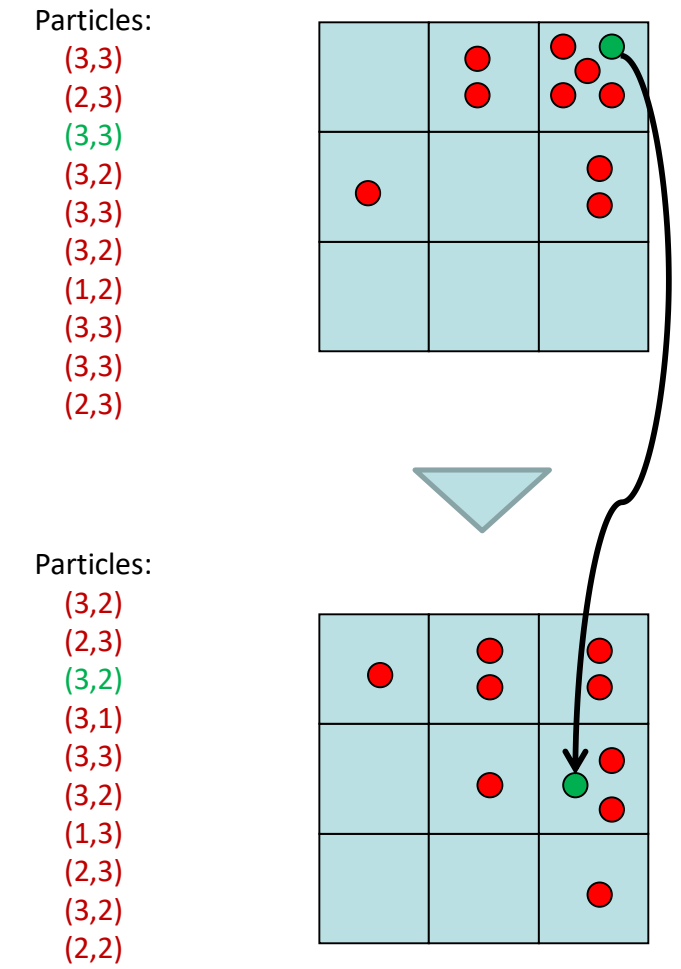
Particle Filtering: Elapse Time

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

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

  - This is like prior sampling – samples' frequencies reflect the transition probabilities
  - Here, most samples move clockwise, but some move in another direction or stay in place

- This captures the passage of time
  - If enough samples, close to exact values before and after (consistent)



Particle Filtering: Observe

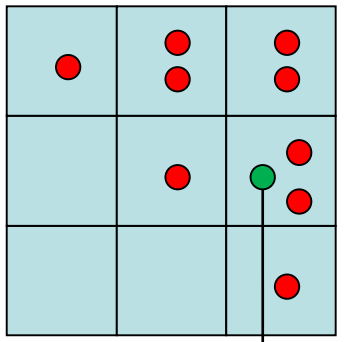
- Slightly trickier:
  - Don't sample observation, fix it
  - Similar to likelihood weighting, downweight samples based on the evidence

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

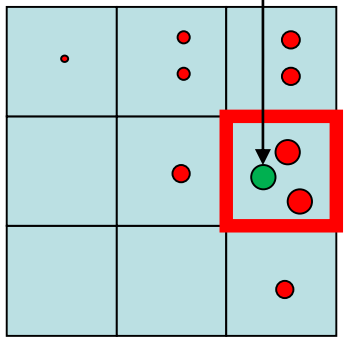
$$B(X) \propto P(e|X)B'(X)$$

  - As before, the probabilities don't sum to one, since all have been downweighted (in fact they now sum to (N times) an approximation of P(e))

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

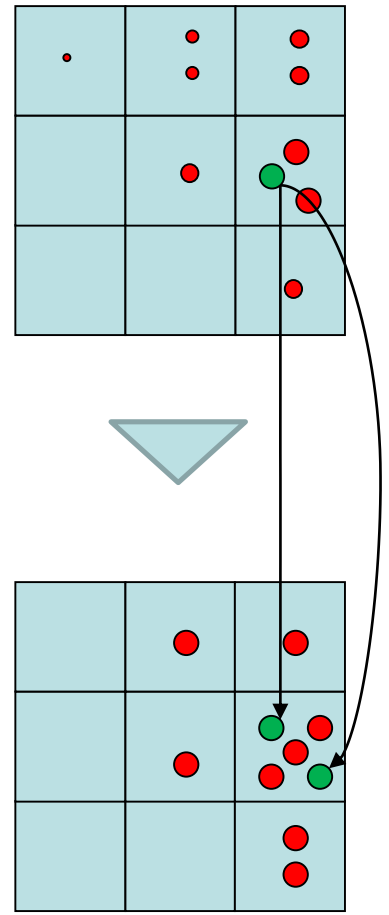


Particle Filtering: Resample

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

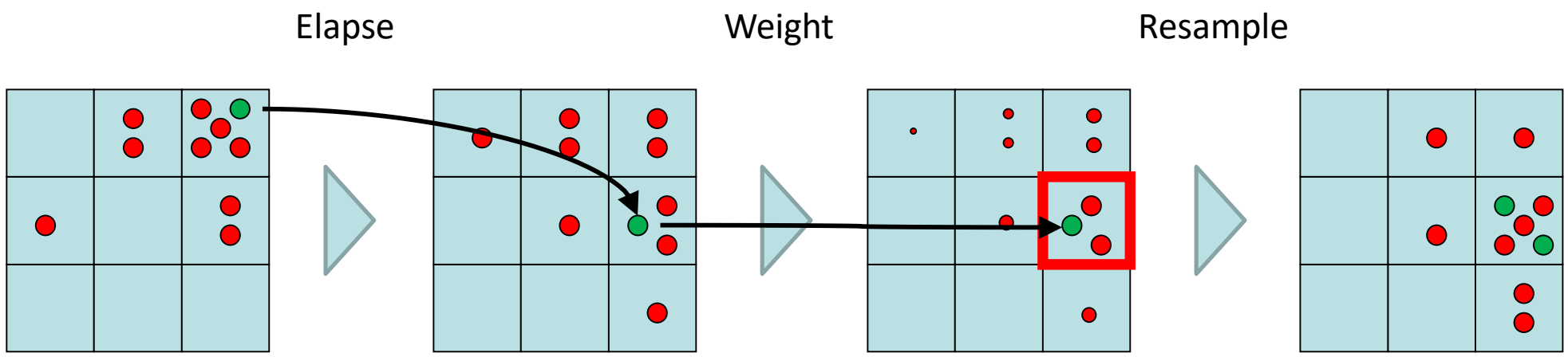
Particles:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4

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



Recap: Particle Filtering

- Particles: track samples of states rather than an explicit distribution



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

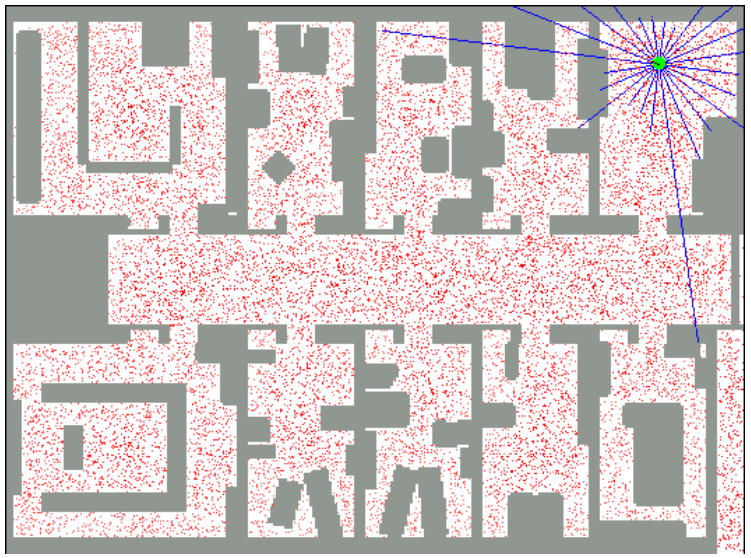
Particles:
(3,2)
(2,3)
(3,2)
(3,1)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(2,2)

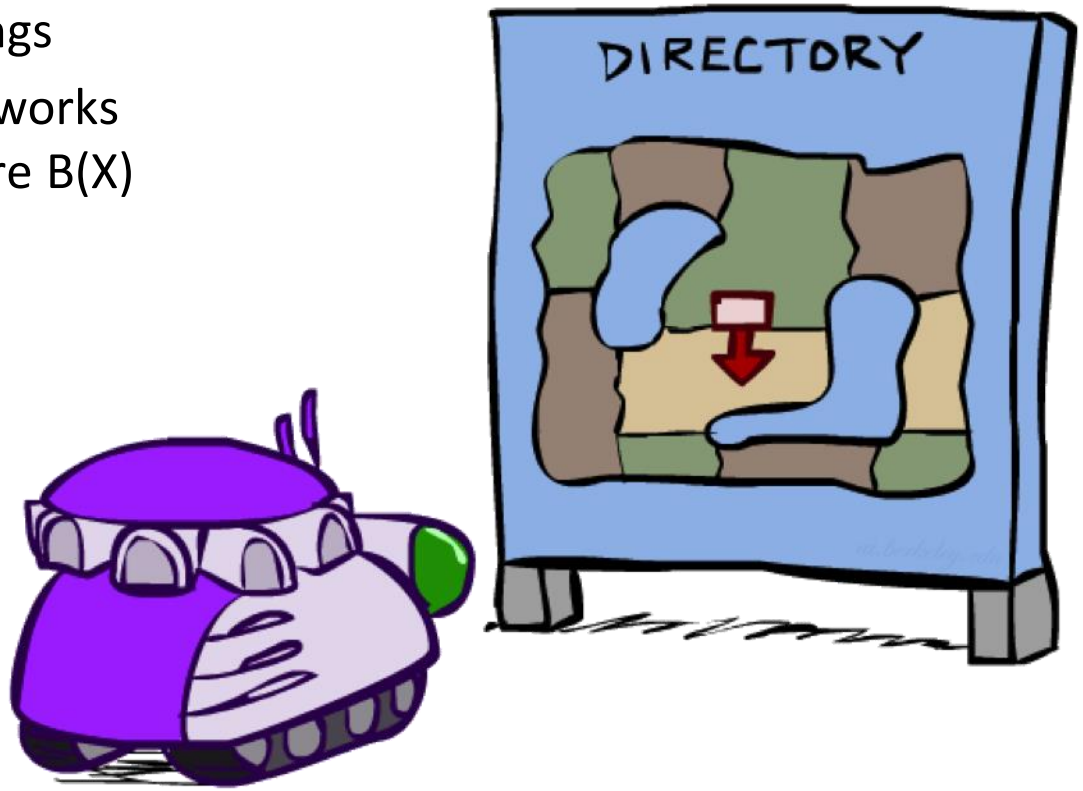
Particles:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4

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

Robot Localization

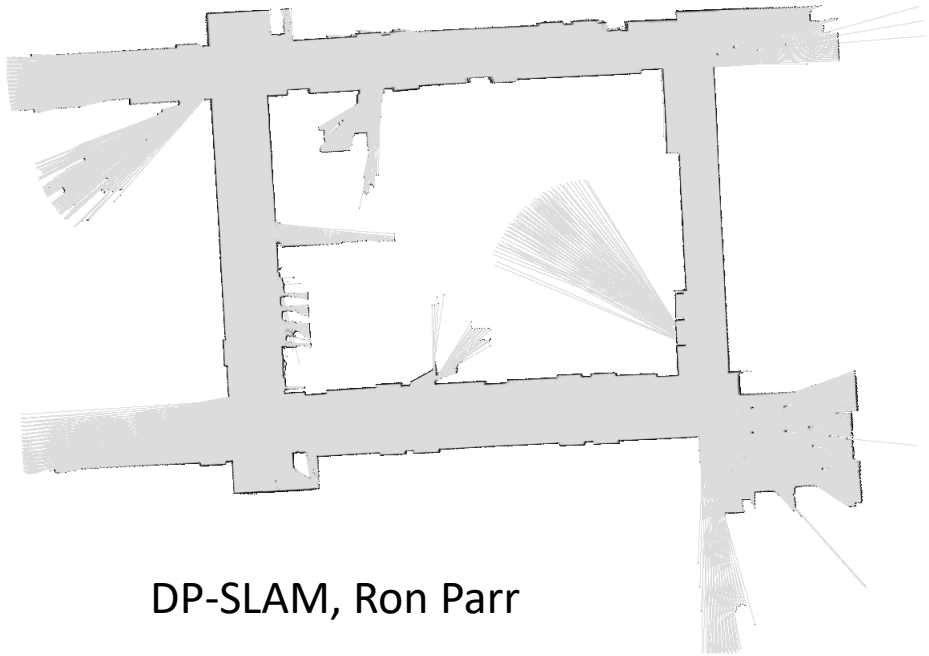
- In robot localization:
  - We know the map, but not the robot's position
  - Observations may be vectors of range finder readings
  - State space and readings are typically continuous (works basically like a very fine grid) and so we cannot store B(X)
  - Particle filtering is a main technique



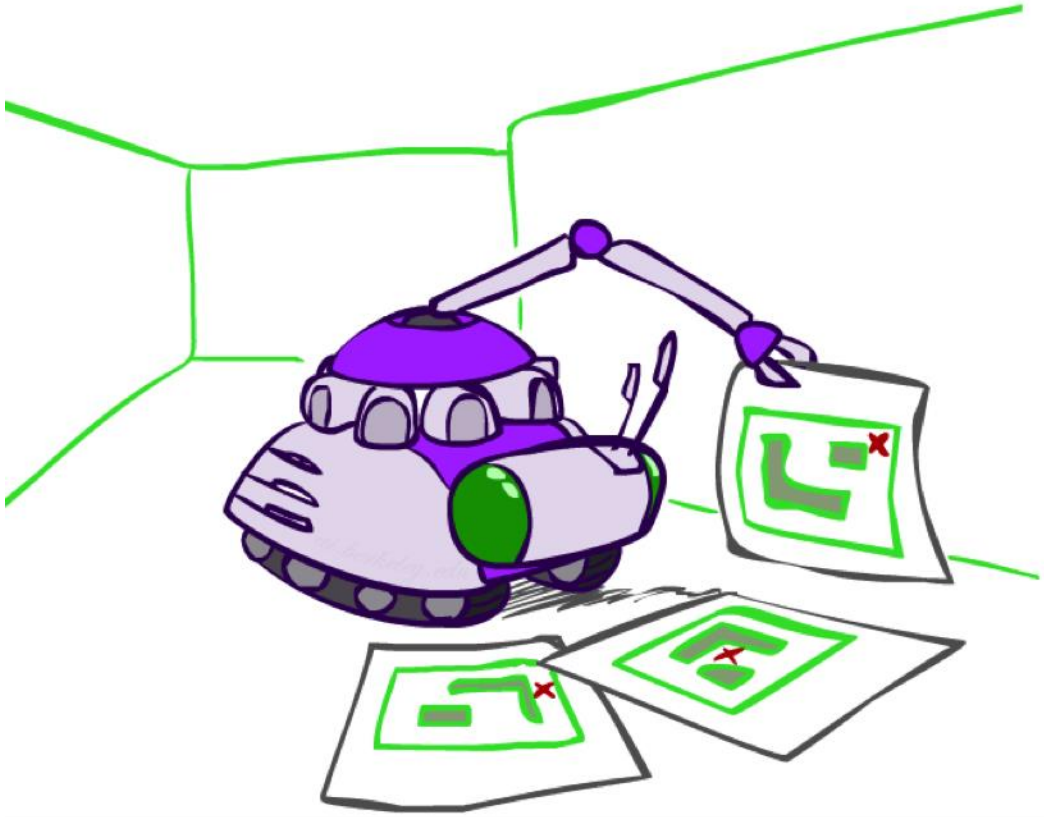


Robot Mapping

- SLAM: Simultaneous Localization And Mapping
  - We do not know the map or our location
  - State consists of position AND map!
  - Main techniques: Kalman filtering (Gaussian HMMs) and particle methods

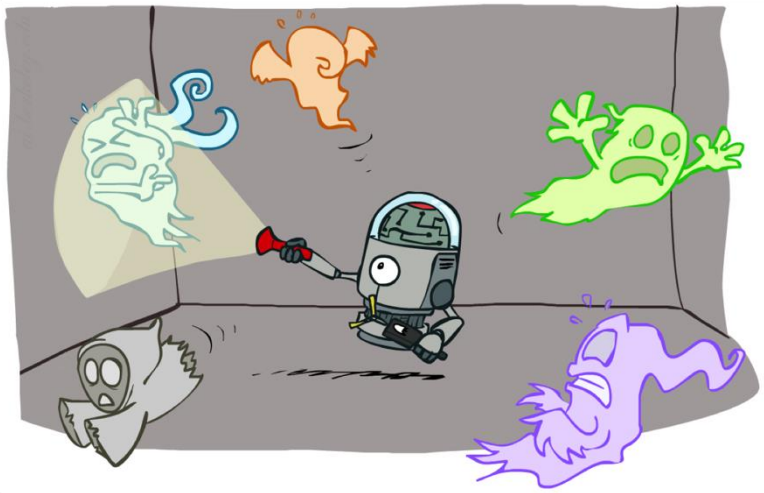


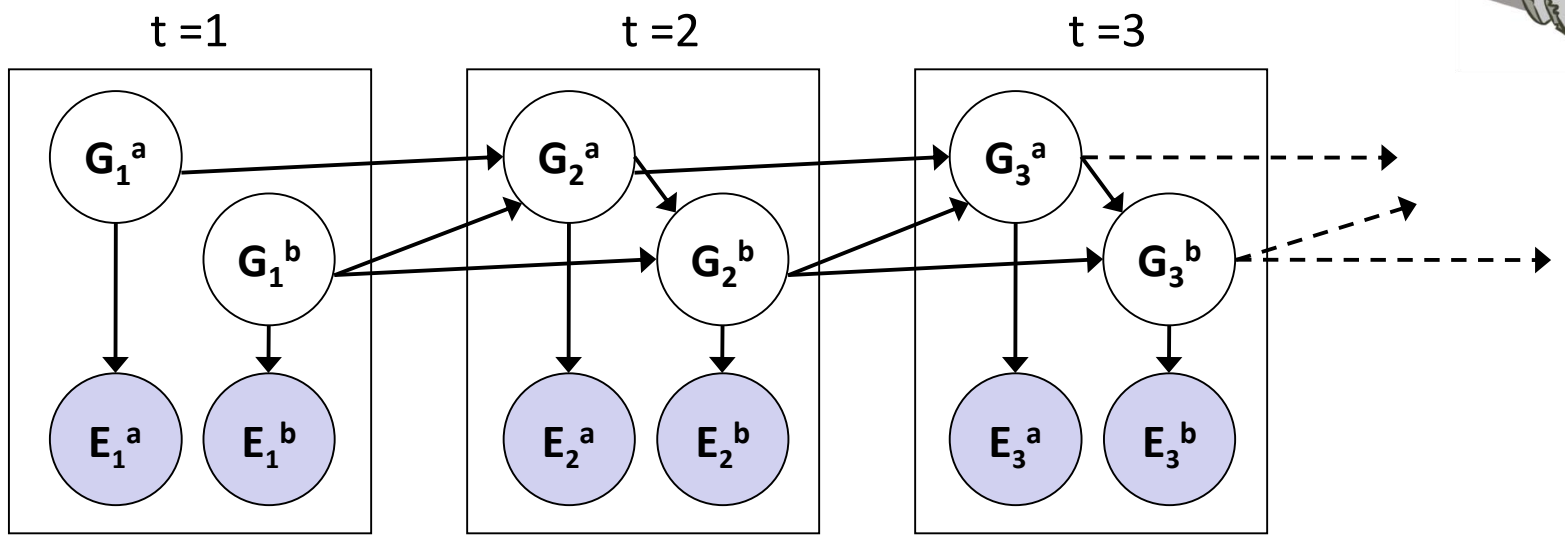
DP-SLAM, Ron Parr



Dynamic Bayes Nets (DBNs)

- We want to track multiple variables over time, using multiple sources of evidence
- Idea: Repeat a fixed Bayes net structure at each time
- Variables from time *t* can condition on those from *t-1*





- Dynamic Bayes nets are a generalization of HMMs

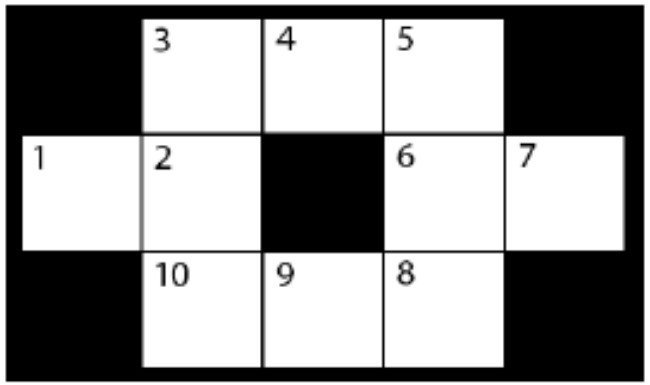
DBN Particle Filters

- A particle is a complete sample for a time step
- **Initialize**: Generate prior samples for the t=1 Bayes net
  - Example particle: $\mathbf{G}_1^{\mathbf{a}} = (3,3)$ $\mathbf{G}_1^{\mathbf{b}} = (5,3)$
- **Elapse time**: Sample a successor for each particle
  - Example successor: $\mathbf{G}_2^{\mathbf{a}} = (2,3)$ $\mathbf{G}_2^{\mathbf{b}} = (6,3)$
- **Observe**: Weight each *entire* sample by the likelihood of the evidence conditioned on the sample
  - Likelihood: $P(\mathbf{E}_1^{\mathbf{a}} \mid \mathbf{G}_1^{\mathbf{a}}) * P(\mathbf{E}_1^{\mathbf{b}} \mid \mathbf{G}_1^{\mathbf{b}})$
- **Resample:** Select prior samples (tuples of values) in proportion to their likelihood

Particle Filter Example

QUESTION 5: PARTICLE FILTERING (4 points possible)

In this question, we will use a particle filter to track the state of a robot that is lost in the small map below:



The robot's state is represented by an integer $1 \leq X_t \leq 10$ corresponding to its location in the map at time t. We will approximate our belief over this state with $N = 8$ particles.

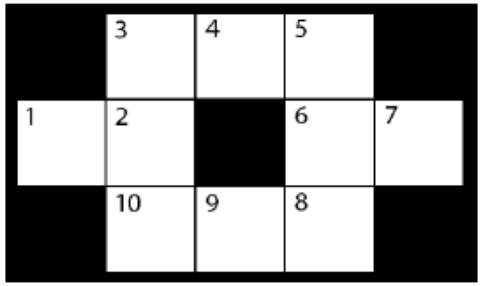
You have no control over the robot's actions. At each timestep, the robot either stays in place, or moves to any one of its neighboring locations, all with equal probability. For example, if the robot starts in state $X_t = 7$, it will move to state $X_{t+1} = 6$ with probability $\frac{1}{2}$ or $X_{t+1} = 7$ with probability $\frac{1}{2}$. Similarly, if the robot starts in state $X_t = 2$, the next state X_{t+1} can be any element of $\{1, 2, 3, 10\}$, and each occurs with probability $\frac{1}{4}$.

At each time step, a sensor on the robot gives a reading $E_t \in \{H, C, T, D\}$ corresponding to the *type* of state the robot is in. The possible types are:

Particle Filter Example

QUESTION 5: PARTICLE FILTERING (4 points possible)

In this question, we will use a particle filter to track the state of a robot that is lost in the small map below:



The robot's state is represented by an integer $1 \leq X_t \leq 10$ corresponding to its location in the map at time t. We will approximate our belief over this state with $N = 8$ particles.

You have no control over the robot's actions. At each timestep, the robot either stays in place, or moves to any one of its neighboring locations, all with equal probability. For example, if the robot starts in state $X_t = 7$, it will move to state $X_{t+1} = 6$ with probability $\frac{1}{2}$ or $X_{t+1} = 7$ with probability $\frac{1}{2}$. Similarly, if the robot starts in state $X_t = 2$, the next state X_{t+1} can be any element of $\{1, 2, 3, 10\}$, and each occurs with probability $\frac{1}{4}$.

At each time step, a sensor on the robot gives a reading $E_t \in \{H, C, T, D\}$ corresponding to the *type* of state the robot is in. The possible types are:

- **Hallway (H)** for states bordered by two parallel walls (4,9).
- **Corner (C)** for states bordered by two orthogonal walls (3,5,8,10).
- **Tee (T)** for states bordered by one wall (2,6).
- **Dead End (D)** for states bordered by three walls (1,7).

The sensor is not very reliable: it reports the correct type with probability $\frac{1}{2}$, but gives erroneous readings the rest of the time, with probability $\frac{1}{6}$ for each of the three other possible readings.

Particle Filter Example

Part 1: Sensor Model

Fill in the sensor model below:

Sensor Reading	State Type	P(Sensor | State Type)
H	H	
C	H	
T	H	
D	H	
H	C	
C	C	
T	C	
D	C	

At

Sensor Reading	State Type	P(Sensor | State Type)
H	T	
C	T	
T	T	
D	T	
H	D	
C	D	
T	D	
D	D	

this point, please make sure that you have checked your answers for Part 1.

Particle Filter Example

Part 2: Belief State for $t = 0$

Now we will sample the starting positions for our particles. For each particle p_i, we have generated a random number r_i sampled uniformly from $[0, 1)$. Your job is to use these numbers to sample a starting location for each particle. As a reminder, locations are integers from the range $[1, 10]$, as shown in the map. You should assume that the locations go in ascending order and that each location has equal probability. The random number generated for particle i, denoted by r_i, is provided in the table below. Please fill in the locations of the eight particles.

Particle	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
r_i	0.914	0.473	0.679	0.879	0.212	0.024	0.458	0.154
Location: x_0								

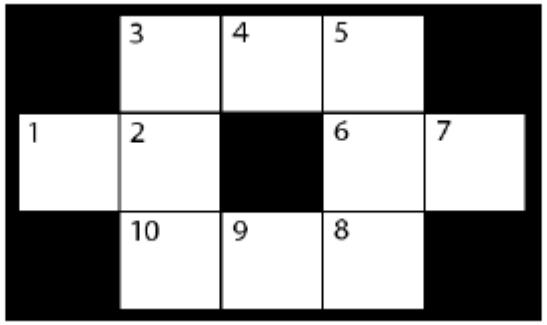
Particle Filter Example

(4 points possible)

Part 3: Time update from $t = 0 \rightarrow t = 1$

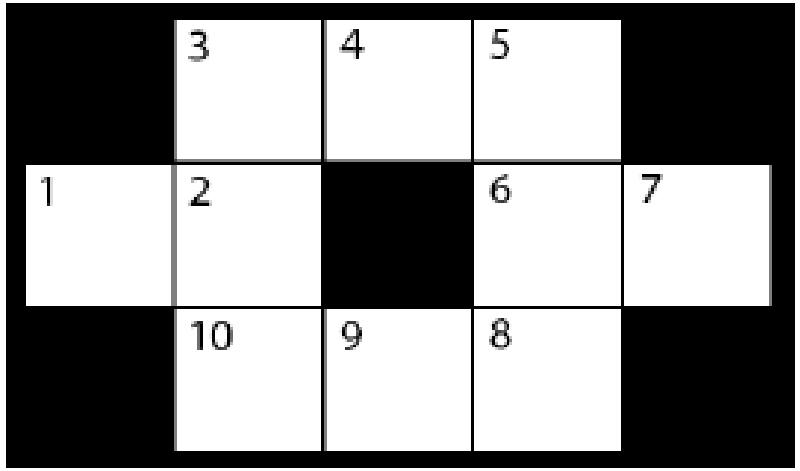
Now we'll perform a time update using the transition model. Stated again, the transition model is as follows: At each timestep, the robot either stays in place, or moves to any one of its neighboring locations, all with equal probability.

For each particle, take the starting position you found in Part 2, and perform the time update for that particle. You should again sample from the range $[0, 1)$, where the bins are the possible locations **sorted in ascending numerical order** . As an example, if $X_t = 2$, the next state can be one of $\{1, 2, 3, 10\}$, each with equal probability, so the $[0, 0.25)$ bin would be for $X_{t+1} = 1$, the $[0.25, 0.5)$ bin would be for $X_{t+1} = 2$, the $[0.5, 0.75)$ bin would be for $X_{t+1} = 3$, and the $[0.75, 1)$ bin would be for $X_{t+1} = 10$.



Particle Filter Example

The map is shown again below:



Particle	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
r_i	0.674	0.119	0.748	0.802	0.357	0.736	0.425	0.058
Location: x_1								

Particle Filter Example

Part 4: Probability Distribution Induced by the Particles

Recall that a particle filter just keeps track of a list of particles, but at any given time, we can compute a probability distribution from these particles. Using the current newly updated set of particles (that you found in Part 3) , give the estimated probability that the robot is in each location.

x_1	1	2	3	4	5
$P(x_1)$					

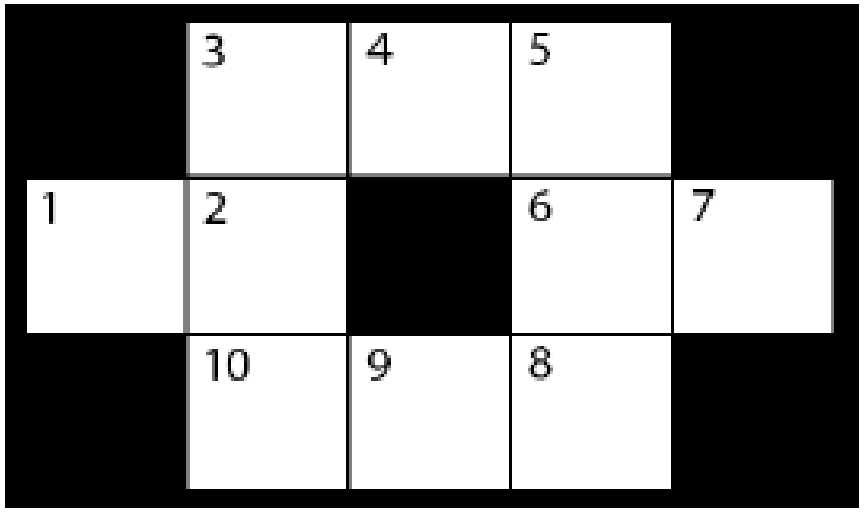
x_1	6	7	8	9	10
$P(x_1)$					

Particle Filter Example

Part 5: Incorporating Evidence at $t = 1$

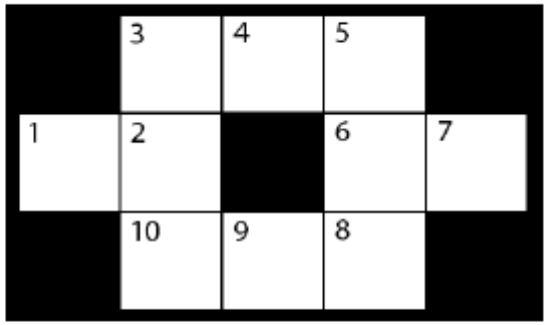
The sensor reading at $t = 1$ is: $E_1 = D$:

Using the sensor model you specified in Part 1, incorporate the evidence by reweighting the particles. Also enter the normalized and cumulative weights for each particle. The normalized weight for a specific particle can be calculated by taking that particle's weight and dividing by the sum of all the particle weights. The cumulative weight keeps track of a running sum of all the weights of the particles seen so far (meaning, particle i at will have a cumulative weight equal to the sum of the weights of all particles j such that $j \leq i$).



Particle Filter Example

The map is shown again below:



Particle	p_1	p_2	p_3	p_4
Weight				
Normalized Weight				
Cumulative Weight				

Particle	p_5	p_6	p_7	p_8
Weight				
Normalized Weight				
Cumulative Weight				

Particle Filter Example

(4 points possible)

Part 6: Resampling

Finally, we'll resample the particles. This reallocates resources to the most relevant parts of the state space in the next time update step.

Notice that your cumulative weights effectively tell you where the bins used in resampling the particles lie. For example, for particle 1, you calculated the cumulative weight to be some value, p. Then, on a random value draw, if a value between 0 and p was chosen, you would generate a new particle where particle 1 is. Use these bounds to resample the eight particles. In the "New Particle" row, enter the particle corresponding to the bin that the random value chose. In the "New Location" row, enter the location corresponding to this new particle. You may need to look back at Part 3 to get the locations of the particles.

Particle	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
r_i	0.403	0.218	0.217	0.826	0.717	0.460	0.794	0.016
New Particle								
New Location								

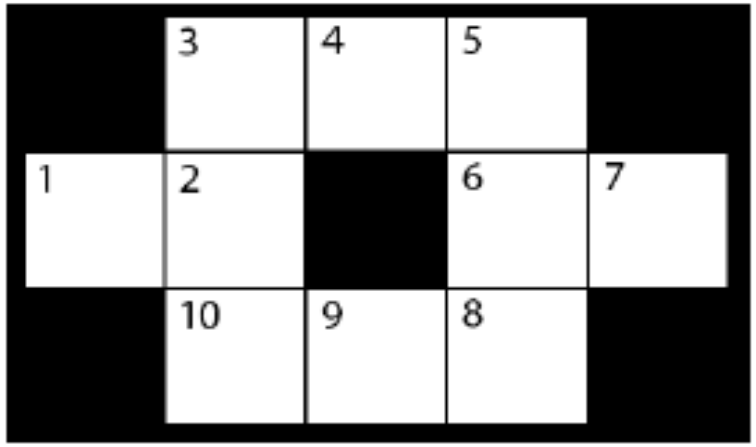
Particle Filter Example

(1 point possible)

Part 7: Analysis

The sensor provided a reading $E_1 = D$. What fraction of the particles are now on a dead end?

The map is shown again below:



This completes everything for the first time step, $t = 0 \rightarrow t = 1$. Of course, we would now continue by repeating the time update, evidence incorporation by reweighting, and resampling. We'll leave that to the computers, though.

Particle Filter Example

Part 1: Sensor Model

Fill in the sensor model below:

Sensor Reading	State Type	P(Sensor | State Type)
H	H	
C	H	
T	H	
D	H	
H	C	
C	C	
T	C	
D	C	

At

Sensor Reading	State Type	P(Sensor | State Type)
H	T	
C	T	
T	T	
D	T	
H	D	
C	D	
T	D	
D	D	

this point, please make sure that you have checked your answers for Part 1.

Sensor Reading	State Type	P(Sensor | State Type)
H	H	0.5 ✔
C	H	1/6 ✔
T	H	1/6 ✔
D	H	1/6 ✔
H	C	1/6 ✔
C	C	0.5 ✔
T	C	1/6 ✔
D	C	1/6 ✔

this point, please make sure that you have checked

Sensor Reading	State Type	P(Sensor | State Type)
H	T	1/6 ✔
C	T	1/6 ✔
T	T	0.5 ✔
D	T	1/6 ✔
H	D	1/6 ✔
C	D	1/6 ✔
T	D	1/6 ✔
D	D	0.5 ✔

your answers for Part 1.

Particle Filter Example

Part 2: Belief State for $t = 0$

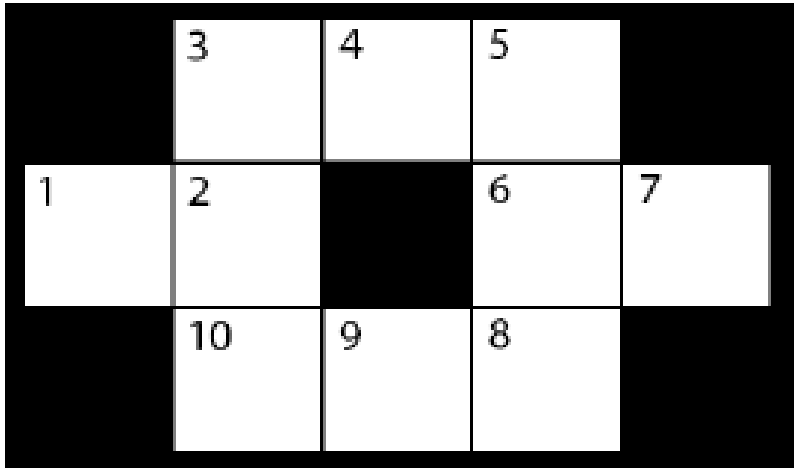
Now we will sample the starting positions for our particles. For each particle p_i, we have generated a random number r_i sampled uniformly from $[0, 1)$. Your job is to use these numbers to sample a starting location for each particle. As a reminder, locations are integers from the range $[1, 10]$, as shown in the map. You should assume that the locations go in ascending order and that each location has equal probability. The random number generated for particle i, denoted by r_i, is provided in the table below. Please fill in the locations of the eight particles.

Particle	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
r_i	0.914	0.473	0.679	0.879	0.212	0.024	0.458	0.154
Location: x_0								

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
0.914	0.473	0.679	0.879	0.212	0.024	0.458	0.154
10 ✔	5 ✔	7 ✔	9 ✔	3 ✔	1 ✔	5 ✔	2 ✔

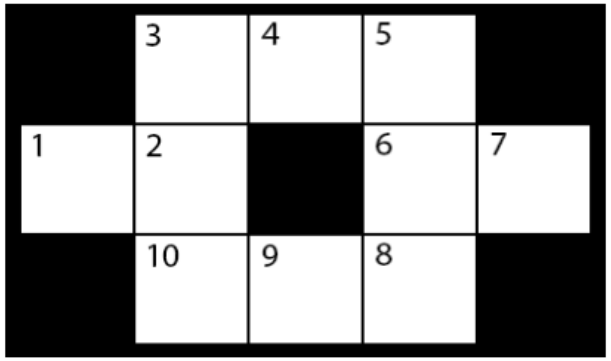
Particle Filter Example

The map is shown again below:

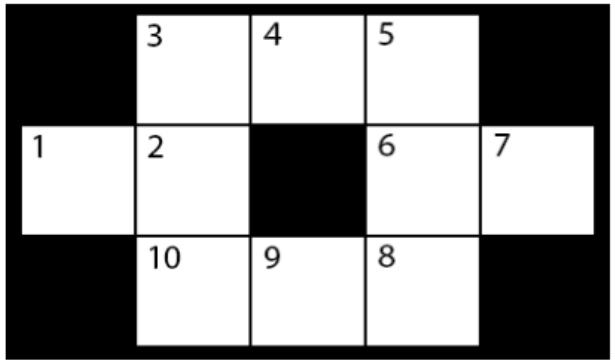


Particle	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
r_i	0.674	0.119	0.748	0.802	0.357	0.736	0.425	0.058
Location: x_1								

Support



p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
0.674	0.119	0.748	0.802	0.357	0.736	0.425	0.058
10	4	7	10	3	3	5	1
✔	✔	✔	✔	✔	✖	✔	✔



Support

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
0.674	0.119	0.748	0.802	0.357	0.736	0.425	0.058
10	4	7	10	3	2	5	1
✔	✔	✔	✔	✔	✔	✔	✔

Particle Filter Example

Part 4: Probability Distribution Induced by the Particles

Recall that a particle filter just keeps track of a list of particles, but at any given time, we can compute a probability distribution from these particles. Using the current newly updated set of particles (that you found in Part 3) , give the estimated probability that the robot is in each location.

x_1	1	2	3	4	5
$P(x_1)$					

x_1	6	7	8	9	10
$P(x_1)$					

Part 4: Probability Distribution Induced by the Particles

Recall that a particle filter just keeps track of a list of particles, but at any given time, we can compute a probability distribution from these particles. Using the current newly updated set of particles (that you found in Part 3) , give the estimated probability that the robot is in each location.

x_1	1	2	3	4	5
$P(x_1)$	1/8 ✔	1/8 ✔	1/8 ✔	1/8 ✔	1/8 ✔

x_1	6	7	8	9	10
$P(x_1)$	0 ✔	1/8 ✔	0 ✔	0 ✔	1/4 ✔

At this point, please make sure that you have checked your answers for Part 4.

Submit

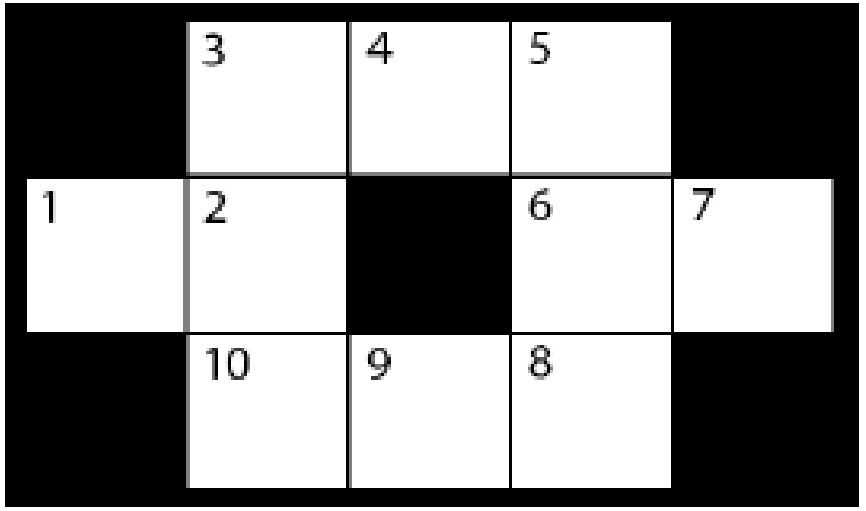
Reset

Particle Filter Example

Part 5: Incorporating Evidence at $t = 1$

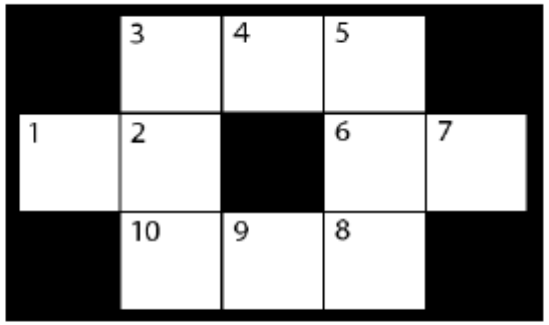
The sensor reading at $t = 1$ is: $E_1 = D$:

Using the sensor model you specified in Part 1, incorporate the evidence by reweighting the particles. Also enter the normalized and cumulative weights for each particle. The normalized weight for a specific particle can be calculated by taking that particle's weight and dividing by the sum of all the particle weights. The cumulative weight keeps track of a running sum of all the weights of the particles seen so far (meaning, particle i at will have a cumulative weight equal to the sum of the weights of all particles j such that $j \leq i$).



Particle Filter Example

The map is shown again below:



Particle	p_1	p_2	p_3	p_4
Weight				
Normalized Weight				
Cumulative Weight				

Particle	p_5	p_6	p_7	p_8
Weight				
Normalized Weight				
Cumulative Weight				

Weight	1/6 ✔	1/6 ✔	0.5 ✔	1/6 ✔
Normalized Weight	1/12 ✔	1/12 ✔	0.25 ✔	1/12 ✔
Cumulative Weight	1/12 ✔ in decimal: 0.083	2/12 ✔ in decimal: 0.167	5/12 ✔ in decimal: 0.417	6/12 ✔ in decimal: 0.5

Particle	p_5	p_6	p_7	p_8
Weight	1/6 ✔	1/6 ✔	1/6 ✔	0.5 ✔
Normalized Weight	1/12 ✔	1/12 ✔	1/12 ✔	0.25 ✔
Cumulative	7/12	8/12	9/12	12/12

Particle Filter Example

(4 points possible)

Part 6: Resampling

Finally, we'll resample the particles. This reallocates resources to the most relevant parts of the state space in the next time update step.

Notice that your cumulative weights effectively tell you where the bins used in resampling the particles lie. For example, for particle 1, you calculated the cumulative weight to be some value, p. Then, on a random value draw, if a value between 0 and p was chosen, you would generate a new particle where particle 1 is. Use these bounds to resample the eight particles. In the "New Particle" row, enter the particle corresponding to the bin that the random value chose. In the "New Location" row, enter the location corresponding to this new particle. You may need to look back at Part 3 to get the locations of the particles.

Particle	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
r_i	0.403	0.218	0.217	0.826	0.717	0.460	0.794	0.016
New Particle								
New Location								

Particle	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
	0.403	0.218	0.217	0.826	0.717	0.460	0.794	0.01
New Particle	3 ✔	3 ✔	3 ✔	8 ✔	7 ✔	4 ✔	8 ✔	1 ✔
New Locatio	7 ✔	7 ✔	7 ✔	1 ✔	5 ✔	10 ✔	1 ✔	10 ✔

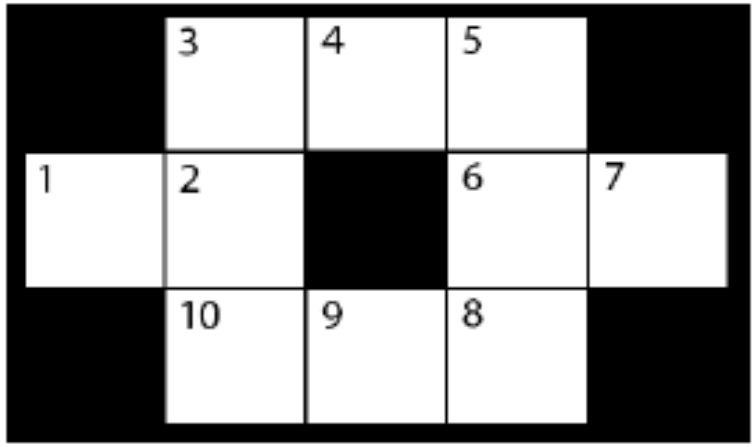
Particle Filter Example

(1 point possible)

Part 7: Analysis

The sensor provided a reading $E_1 = D$. What fraction of the particles are now on a dead end?

The map is shown again below:



This completes everything for the first time step, $t = 0 \rightarrow t = 1$. Of course, we would now continue by repeating the time update, evidence incorporation by reweighting, and resampling. We'll leave that to the computers, though.

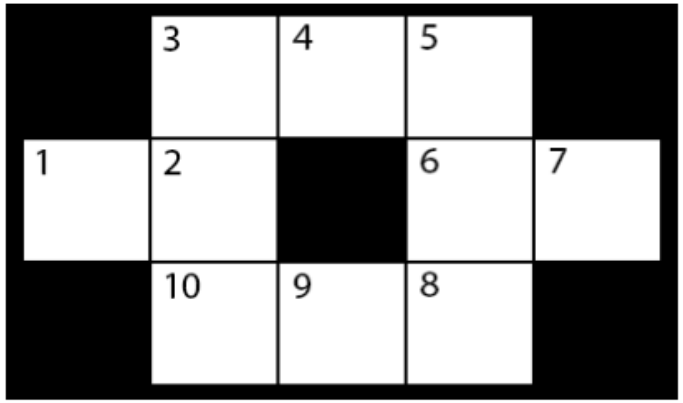
problem

1/1 point (graded)

Part 7: Analysis

The sensor provided a reading $E_1 = D$. What fraction of the particles are now on a dead end?

The map is shown again below:



5/8 ✔

This completes everything for the first time step, $t = 0 \rightarrow t = 1$. Of course, we would now continue by repeating the time update, evidence incorporation by reweighting, and resampling. We'll leave that to the computers, though.

ヾ(⌐■_■)ノ♪

Smoothing

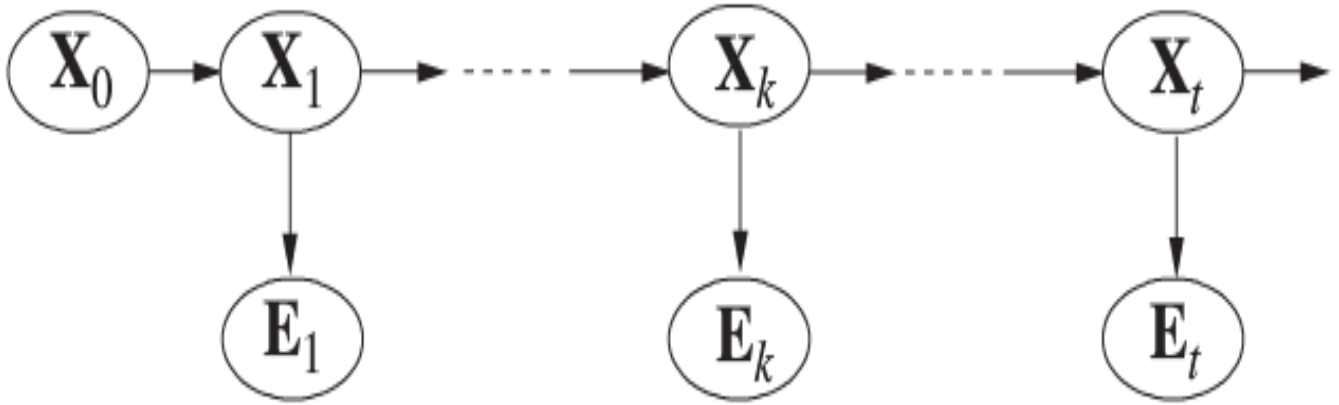
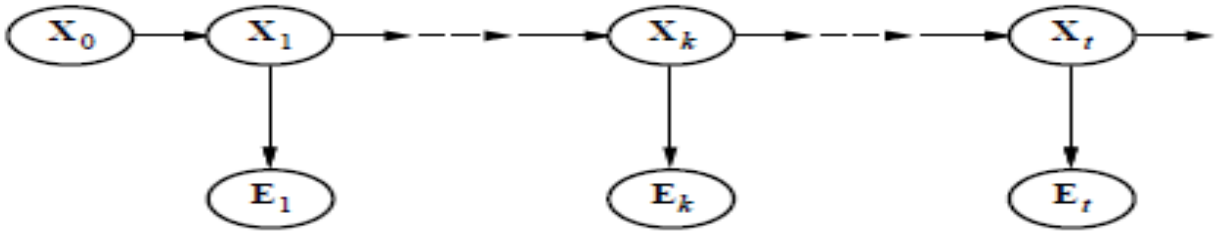


Figure 15.3 Smoothing computes $\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t})$, the posterior distribution of the state at some past time k given a complete sequence of observations from 1 to t.

Smoothing



Divide evidence $\mathbf{e}_{1:t}$ into $\mathbf{e}_{1:k}$, $\mathbf{e}_{k+1:t}$:

$$\begin{aligned}\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t}) &= \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k},\mathbf{e}_{k+1:t}) \\ &= \alpha\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k,\mathbf{e}_{1:k}) \\ &= \alpha\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) \\ &= \alpha\mathbf{f}_{1:k}\mathbf{b}_{k+1:t}\end{aligned}$$

Backward message computed by a backwards recursion:

$$\begin{aligned}\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) &= \textstyle\sum_{\mathbf{x}_{k+1}}\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k,\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k) \\ &= \textstyle\sum_{\mathbf{x}_{k+1}}P(\mathbf{e}_{k+1:t}|\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k) \\ &= \textstyle\sum_{\mathbf{x}_{k+1}}P(\mathbf{e}_{k+1}|\mathbf{x}_{k+1})P(\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)\end{aligned}$$

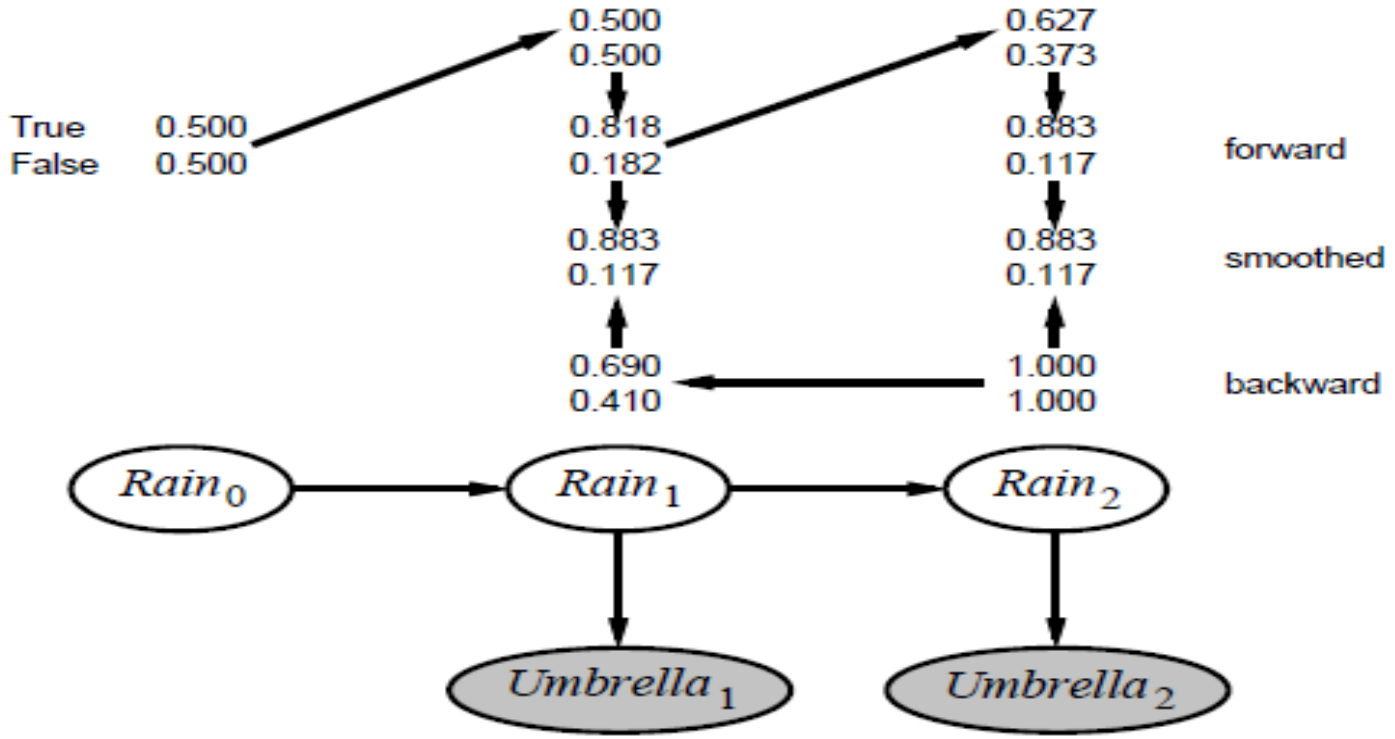
Smoothing w/ Backward Algorithm

Backward message computed by a backwards recursion:

$$\begin{aligned}\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) &= \sum_{\mathbf{x}_{k+1}} \mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k, \mathbf{x}_{k+1}) \mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k) \\ &= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1:t}|\mathbf{x}_{k+1}) \mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k) \\ &= \sum_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1}|\mathbf{x}_{k+1}) P(\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1}) \mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)\end{aligned}$$

- $\mathbf{b}_{k+1:t}$=Backward($\mathbf{b}_{k+1:t}$, $\mathbf{e}_{k+1}$)

Smoothing w/ Forward-Backward



Forward–backward algorithm: cache forward messages along the way
Time linear in t (polytree inference), space $O(t|\mathbf{f}|)$

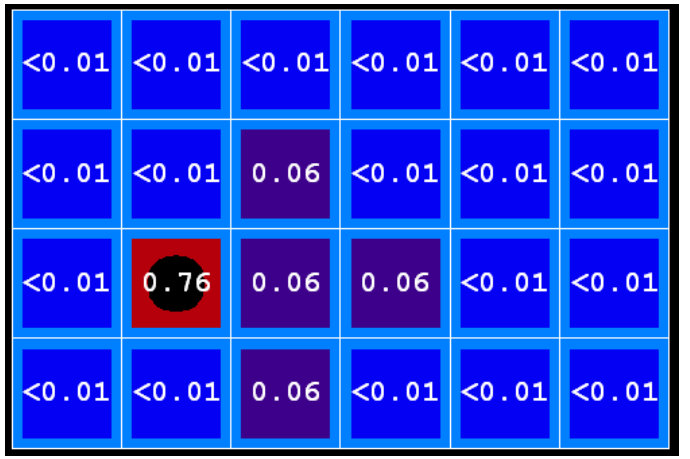
Recap: Filtering

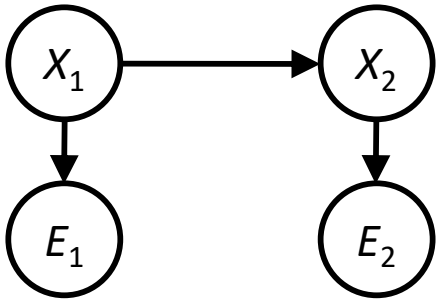
Elapse time: compute P(X_t | $e_{1:t-1}$)

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

Observe: compute P(X_t | $e_{1:t}$)

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





Belief: <P(rain), P(sun)>

$P(X_1)$	<0.5, 0.5>	*Prior on X_1*
$P(X_1 \mid E_1 = umbrella)$	<0.82, 0.18>	*Observe*
$P(X_2 \mid E_1 = umbrella)$	<0.63, 0.37>	*Elapse time*
$P(X_2 \mid E_1 = umb, E_2 = umb)$	<0.88, 0.12>	*Observe*

Smoothing

- $P(R_1 | +u_1, +u_2)$
  - Calculate the smoothed value at time step 1
- $P(R_1 | +u_1, +u_2) = \alpha\ P(R_1 | +u_1)\ P(+u_2 |\ R_1)$
- $P(R_1 | u_1) = \langle 0.818, 0.182 \rangle$

- $P(+u_2 |\ R_1) = P(+u_2 |\ +r_2)\ P(+r_2 | R_1) + P(+u_2 |\ -r_2)\ P(-r_2 | R_1)$

Forward-Backward Algorithm

function FORWARD-BACKWARD(**ev**, $prior$) **returns** a vector of probability distributions
  inputs: **ev**, a vector of evidence values for steps $1, \ldots, t$
    $prior$, the prior distribution on the initial state, $\mathbf{P}(\mathbf{X}_0)$
  local variables: **fv**, a vector of forward messages for steps $0, \ldots, t$
    b, a representation of the backward message, initially all 1s
    sv, a vector of smoothed estimates for steps $1, \ldots, t$

  $\mathbf{fv}[0] \leftarrow prior$
  for $i = 1$ **to** t **do**
    $\mathbf{fv}[i] \leftarrow$ FORWARD$(\mathbf{fv}[i-1], \mathbf{ev}[i])$
  for $i = t$ **downto** 1 **do**
    $\mathbf{sv}[i] \leftarrow$ NORMALIZE$(\mathbf{fv}[i] \times \mathbf{b})$
    $\mathbf{b} \leftarrow$ BACKWARD$(\mathbf{b}, \mathbf{ev}[i])$
  return sv

Simplified Matrix Algorithms w/ HMM's

- X_t is a single, discrete variable (usually E_t is too)
- Domain of X_t is {1, …, S}
- Transition Matrix: $T_{ij}=P(X_t=j|X_{t-1}=i)$
  - $\begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}$
- Sensor Matrix: Ot for each time step: $P(e_t|X_t=i)$
  - Umbrella-true, $O_1 = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix}$

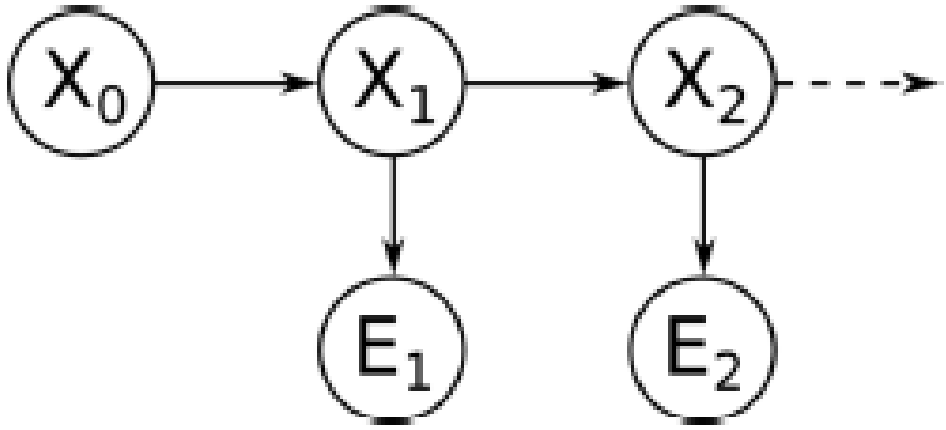
Simplified Matrix Algorithms w/ HMM's

- X_t is a single, discrete variable (usually E_t is too)
- Domain of X_t is {1, …, S}
- Transition Matrix: $T_{ij}=P(X_t=j|X_{t-1}=i)$
  - $\begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}$
- Sensor Matrix: O_t for each time step: $P(e_t|X_t=i)$
  - Umbrella-true, $O_1 = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix}$
- Forward and Backward messages are column vectors:
  - $f_{1:t+1}=\alpha O_{t+1} T^T f$
  - $b_{k+1:t} = T O_{k+1} b_{k+2:t}$

HMM Example

QUESTION 4: HMMS, PART II (18 points possible)

Consider the same HMM.



The prior probability $P(X_0)$, dynamics model $P(X_{t+1}|X_t)$, and sensor model $P(E_t|X_t)$ are as follows:

HMM Example

X_0	$P(X_0)$
0	0.7
1	0.3

X_{t+1}	X_t	$P(X_{t+1}\|X_t)$
0	0	0.1
1	0	0.9
0	1	0.8
1	1	0.2

E_t	X_t	$P(E_t\|X_t)$
a	0	0.4
b	0	0.4
c	0	0.2
a	1	0.7
b	1	0.2
c	1	0.1

In this question we'll assume the sensor is broken and we get no more evidence readings. We are forced to rely on dynamics updates only going forward. In the limit as $t \to \infty$, our belief about X_t should converge to a stationary distribution $\tilde{B}(X_\infty)$ defined as follows:

$$\tilde{B}(X_\infty) := \lim_{t\to\infty} P(X_t|E_1, E_2)$$

Recall that the stationary distribution satisfies the equation

$$\tilde{B}(X_\infty) = \sum_{X_\infty} P(X_{t+1}|X_t)\tilde{B}(X_\infty)$$

for all values in the domain of X.

In the case of this problem, we can write these relations as a set of linear equations of the form

HMM Example

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} \tilde{B}(X_\infty = 0) \\ \tilde{B}(X_\infty = 1) \end{bmatrix} = \begin{bmatrix} \tilde{B}(X_\infty = 0) \\ \tilde{B}(X_\infty = 1) \end{bmatrix}$$

In the spaces below, fill in the coefficients of the linear system. The system you have written has many solutions (consider (0,0), for example), but to get a probability distribution we want the solution that sums to one. Fill in your solution in the table below. (Hint: to check your answer, you can also write some code and run till convergence.)

coefficient	value
a	
b	
c	
d	

X_∞	$\tilde{B}(X_\infty)$
0	
1	

For this problem, you may press "Check" as many times as you want without resetting the problem, so that you don't have to reset the problem for trivial math mistakes.

Check

Most Likely Explanation

Most Likely Explanation

- IS NOT: The sequence of most likely states!!!!
- Most likely path to each x_{t+1}

  = most likely path to some x_t plus one more step
- $\arg \max_{x_1,\ldots xt} P(x_1, \ldots, xt, X_{t+1}|e_{1:\,t+1})$

  = $P(e_{t+1}|X_{t+1}) \max_{x_t}(P(X_{t+1}|\,x_t) \max_{x_1,\ldots xt_{-1}} P(x_1, \ldots, x_{t-1}, x_t, |e_{1:t}))$
- Identical to Filtering, except $f_{1:t}$ replaced by

  $m_{1:t} = \max_{x_1,\ldots,x_{t-1}} P(x_1, \ldots, x_{t-1}, x_t, |e_{1:t})$
- $m_{1:t}$ gives the probability of the most likely path to state i.
  - Update has sum replaced by max, giving the Viterbi algorithm:

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

Most Likely Explanation Viterbi Algorithm

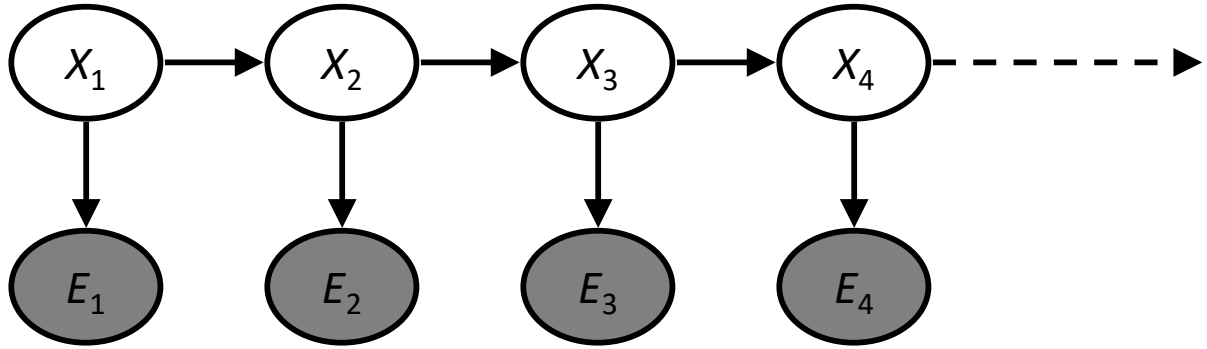
Andrew J. Viterbi

Born	March 9, 1935 (age 79) Bergamo, Italy
Nationality	Italian, American
Education	Massachusetts Institute of Technology (BS, MS) University of Southern California (PhD)
Spouse(s)	Erna Finci (<abbr="married">m. 1958)
Children	Alexander Viterbi (1971–2011) Audrey Viterbi Alan Viterbi
	Engineering career
Engineering discipline	Electrical
Institution memberships	University of Southern California Board of Trustees The Scripps Research Institute Board of Trustees, Sanford Burnham Medical Research Institute
Employer(s)	**Professor:** UC Los Angeles UC San Diego **Founder/Co-founder:** Linkabit Corporation Qualcomm Inc. The Viterbi Group
Significant projects	Viterbi algorithm
Significant advance	Code Division Multiple Access standard for cell phone networks

- Andrew Viterbi
- Co-Founder: Qualcomm
- Venture Capital Company:
  - The Viterbi Group

HMMs: MLE Queries

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



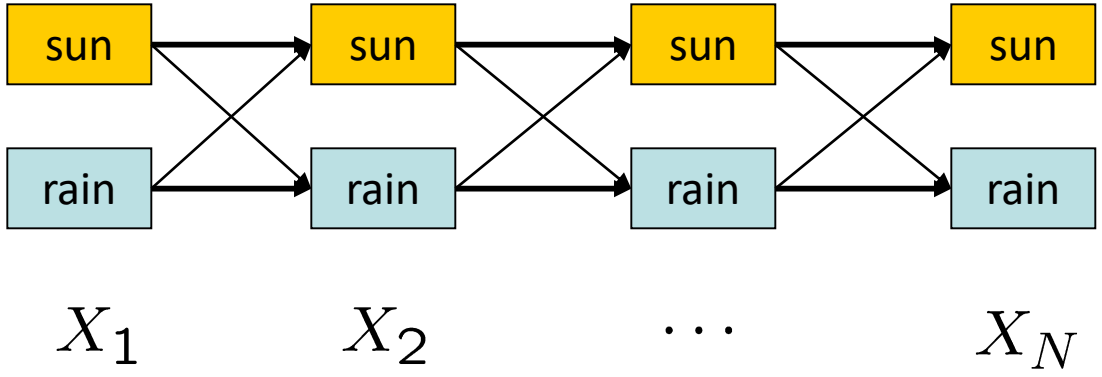
- New query: most likely explanation:

$$\arg\max_{x_{1:t}} P(x_{1:t}|e_{1:t})$$

- New method: the Viterbi algorithm

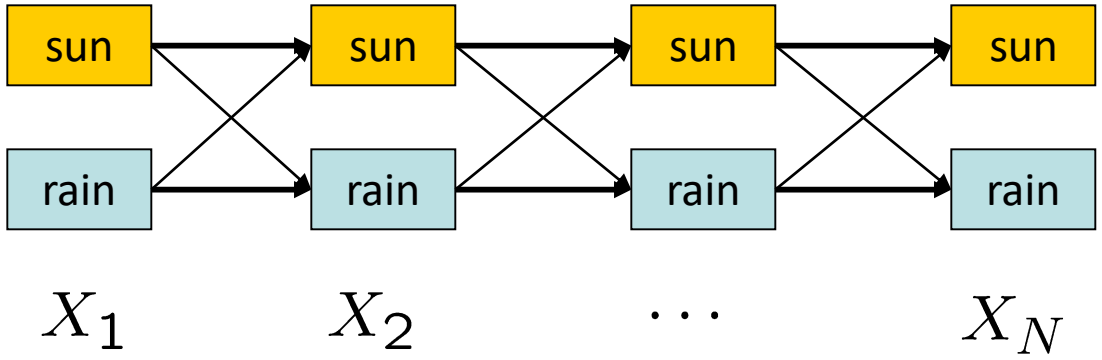
State Trellis

- 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
- Forward algorithm computes sums of paths, Viterbi computes best paths

Forward / Viterbi Algorithms



Forward Algorithm (Sum)

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

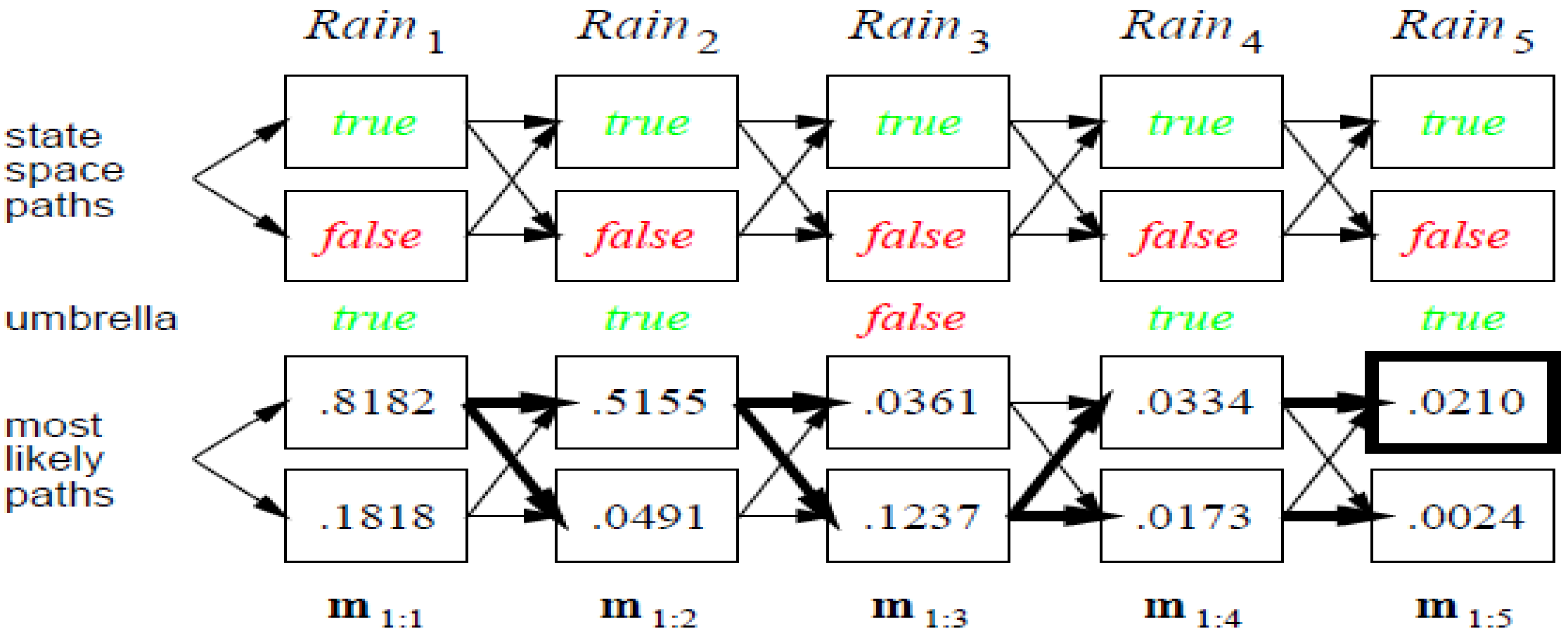
$$= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) f_{t-1}[x_{t-1}]$$

Viterbi Algorithm (Max)

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

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

Veterbi Example



Challenge

- Setting
  - User we want to spy on use HTTPS to browse the internet
- Measurements
  - IP address
  - Sizes of packets coming in
- Goal
  - Infer browsing sequence of that user

- E.g.: medical, financial, legal, …

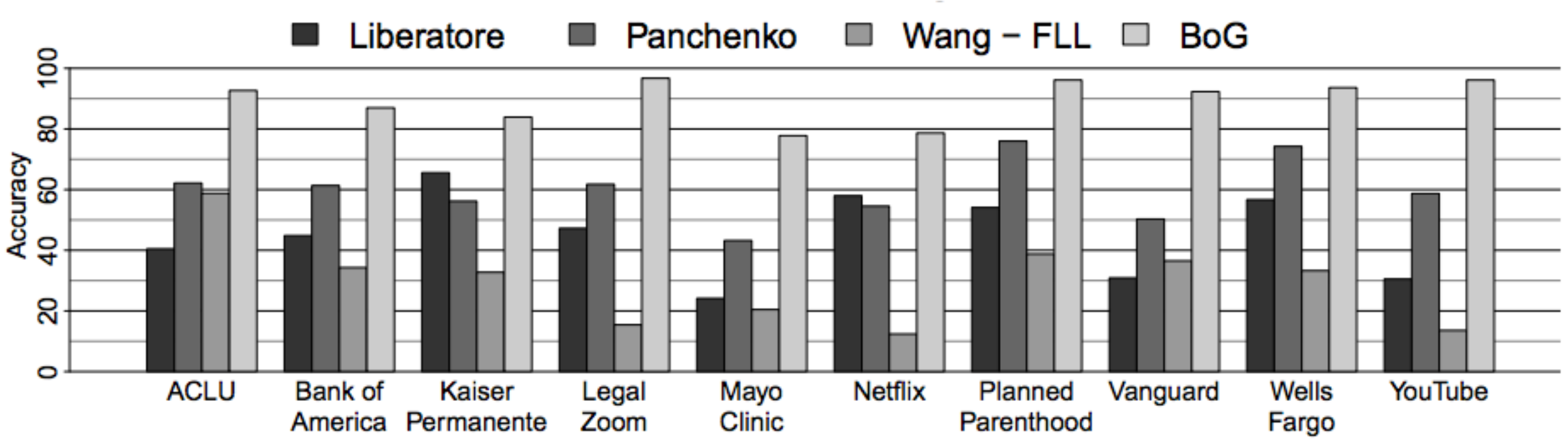
HMM

- Transition model
  - Probability distribution over links on the current page + some probability to navigate to any other page on the site

- Noisy observation model due to traffic variations
  - Caching
  - Dynamically generated content
  - User-specific content, including cookies

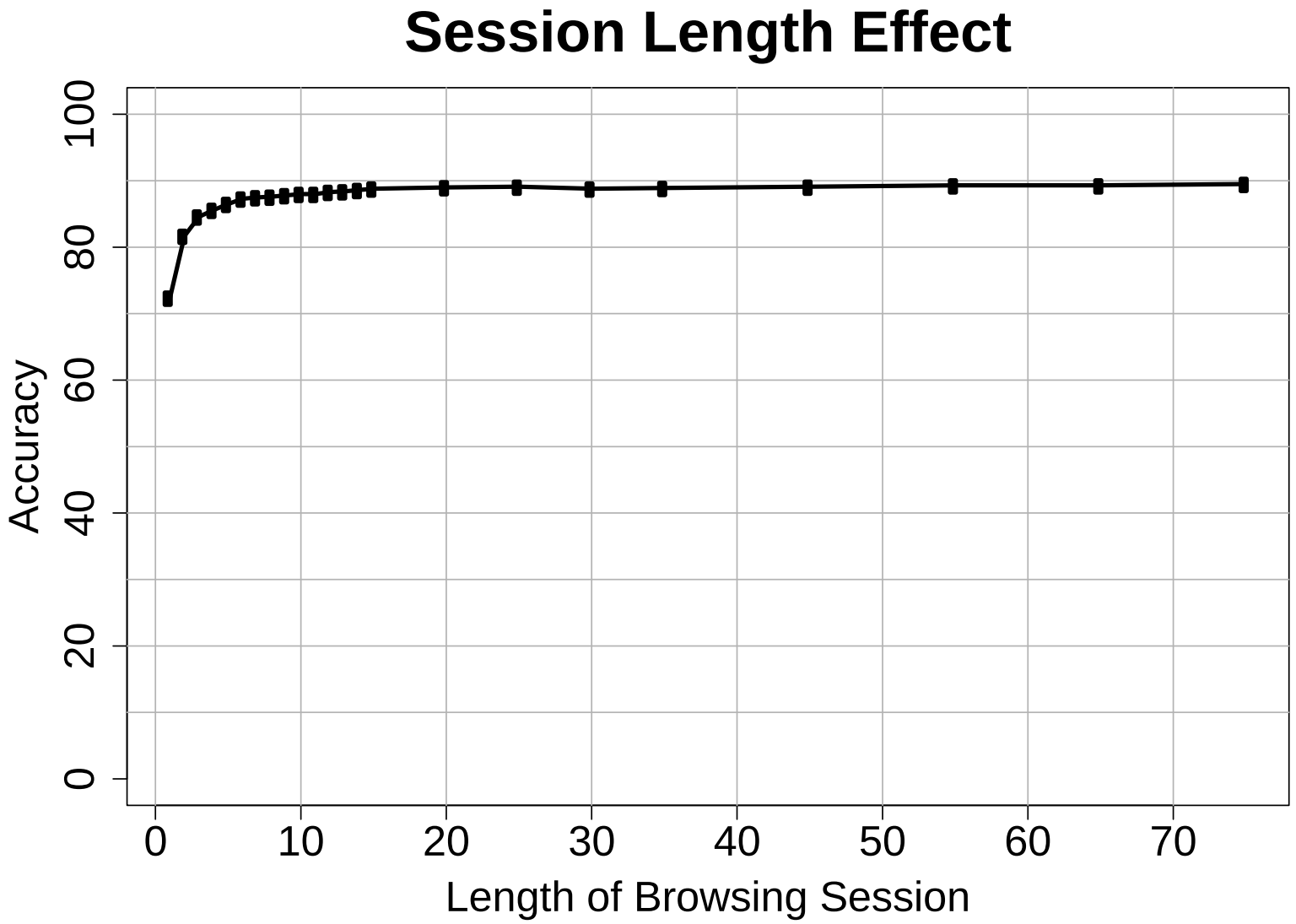
  → Probability distribution P(packet size | page)

Results

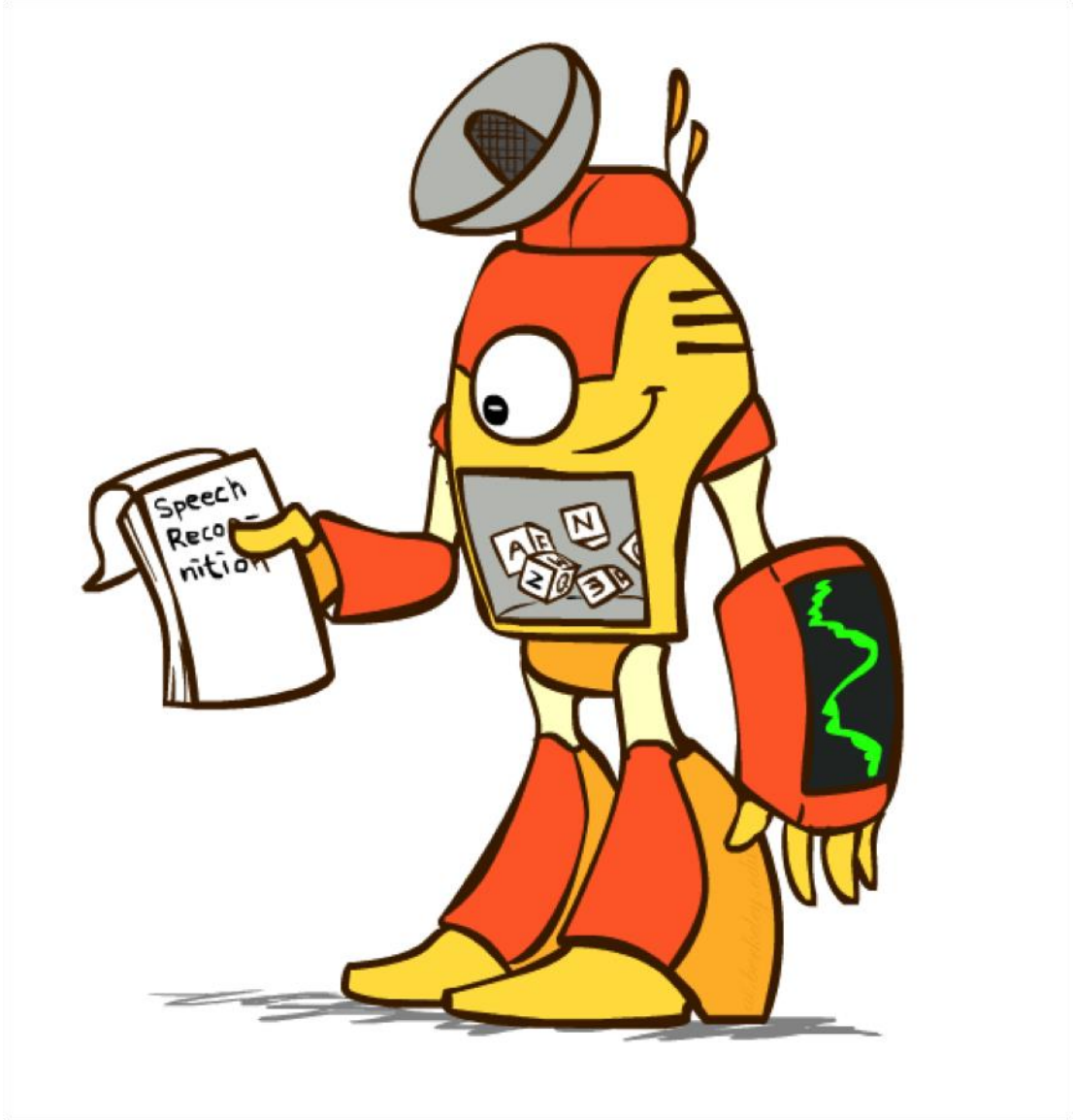


BoG = described approach, others are prior work

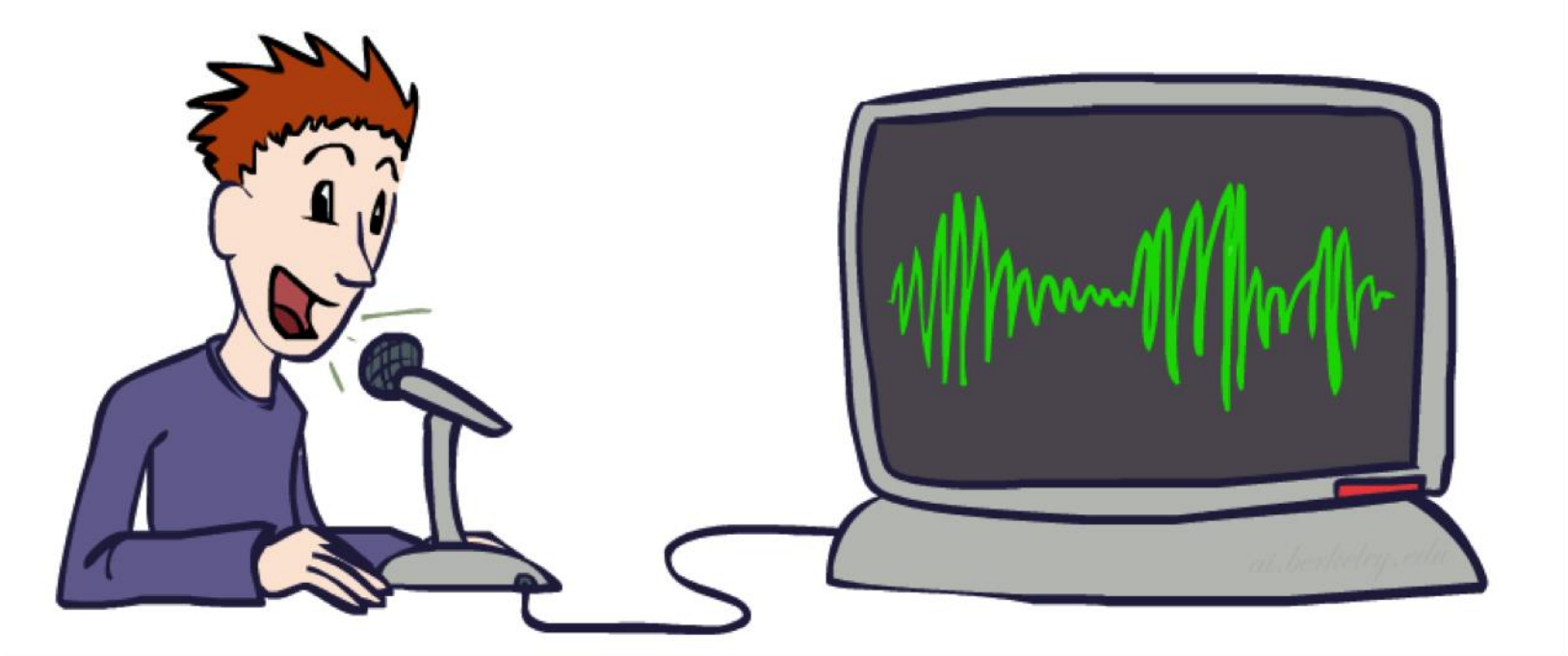
Results


Session Length Effect
Accuracy
0
20
40
60
80
100
0
10
20
30
40
50
60
70
Length of Browsing Session

Speech Recognition



Digitizing Speech



Speech in an Hour

- Speech input is an acoustic waveform

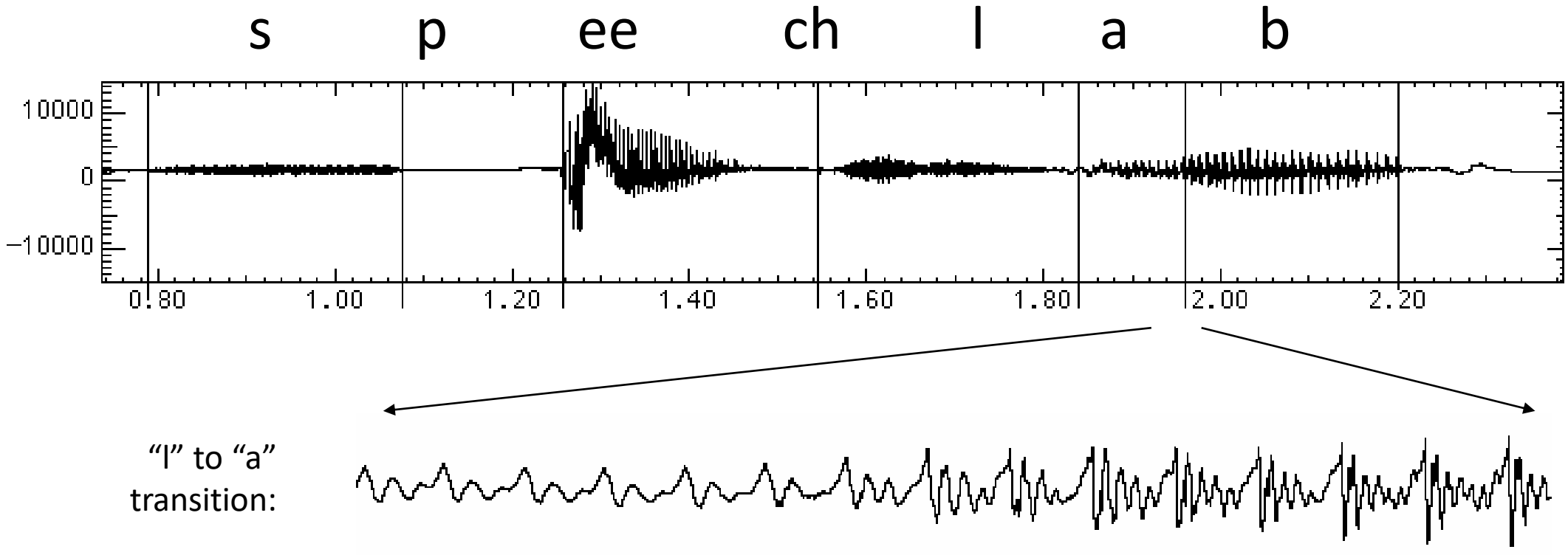
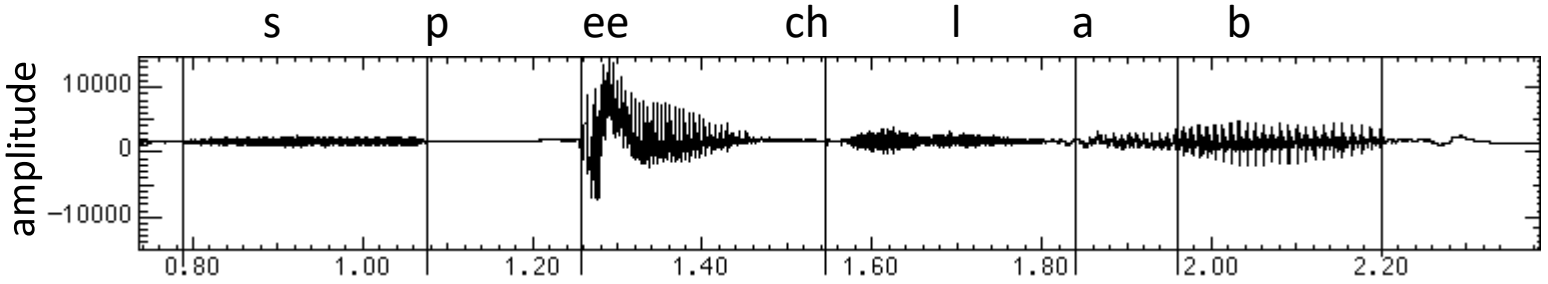
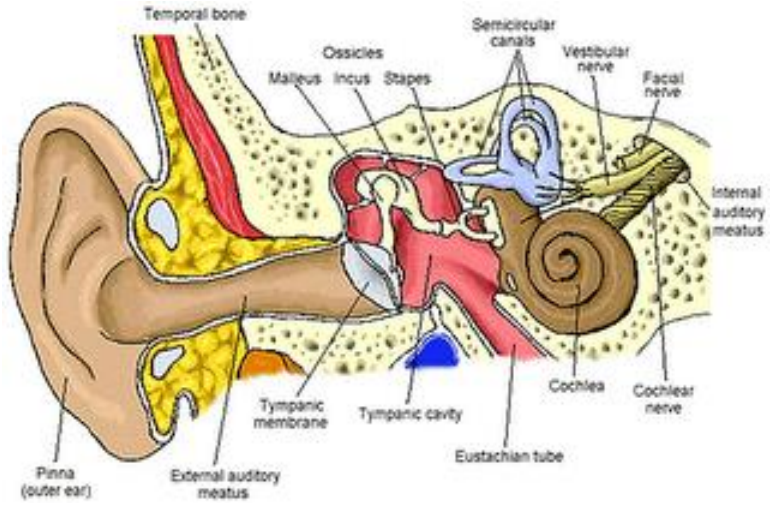


Figure: Simon Arnfield, http://www.psyc.leeds.ac.uk/research/cogn/speech/tutorial/

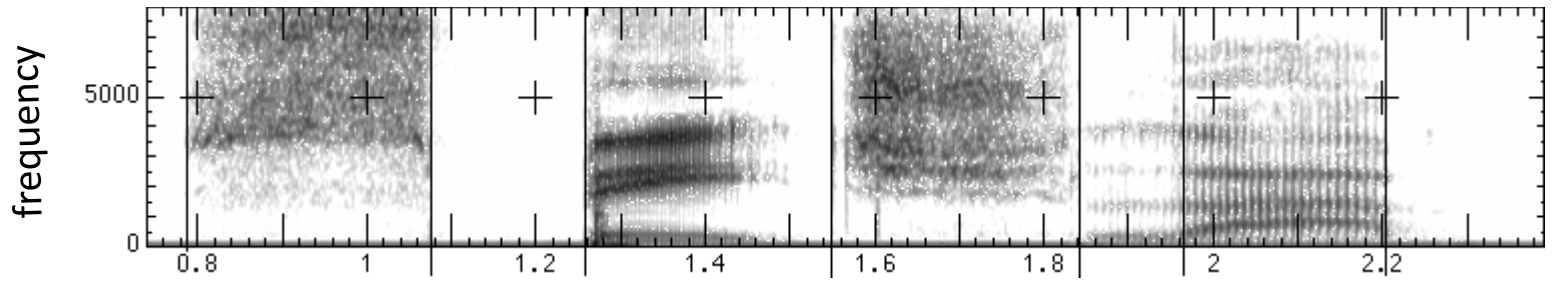
Spectral Analysis

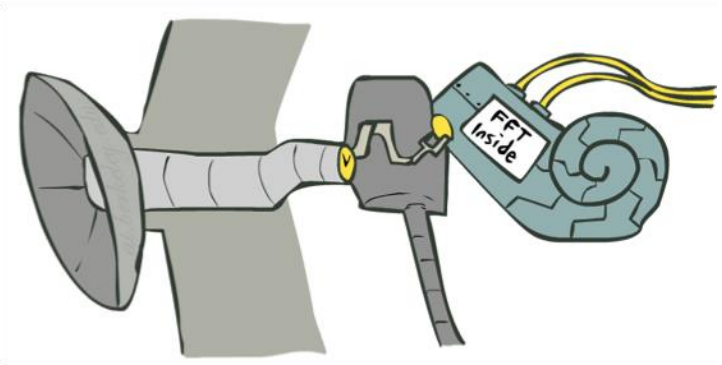
- Frequency gives pitch; amplitude gives volume
  - Sampling at ~8 kHz (phone), ~16 kHz (mic) (kHz=1000 cycles/sec)





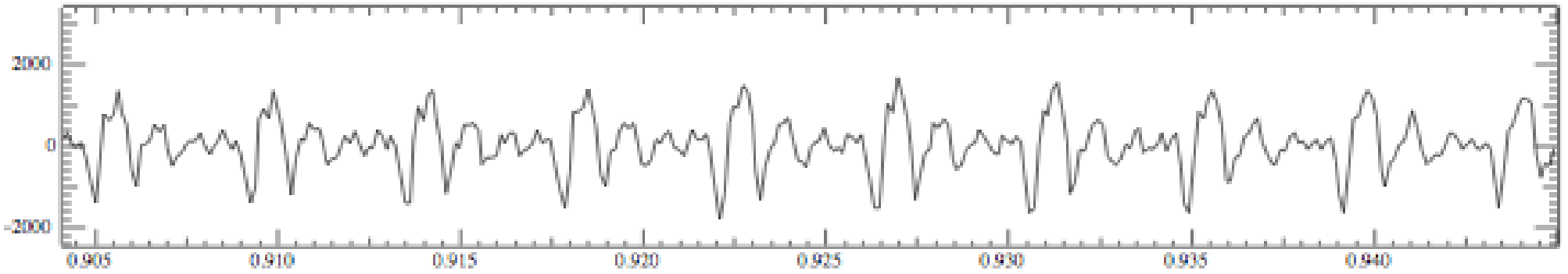
- Fourier transform of wave displayed as a spectrogram
  - Darkness indicates energy at each frequency



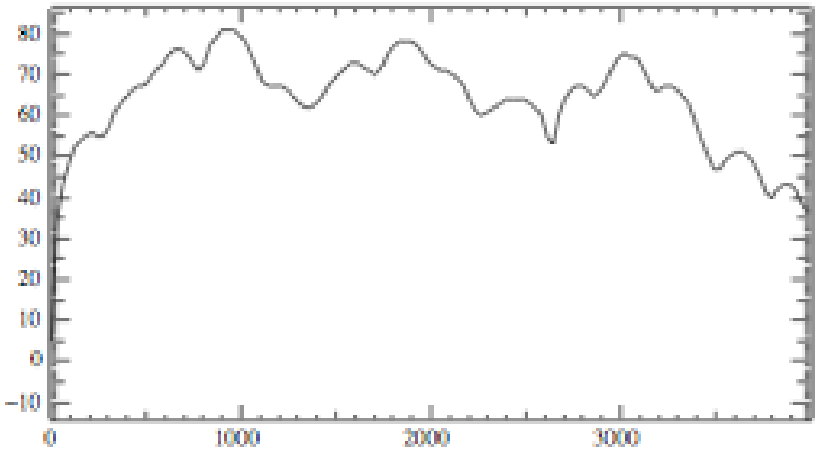


Human ear figure: depion.blogspot.com

Part of [ae] from “lab”

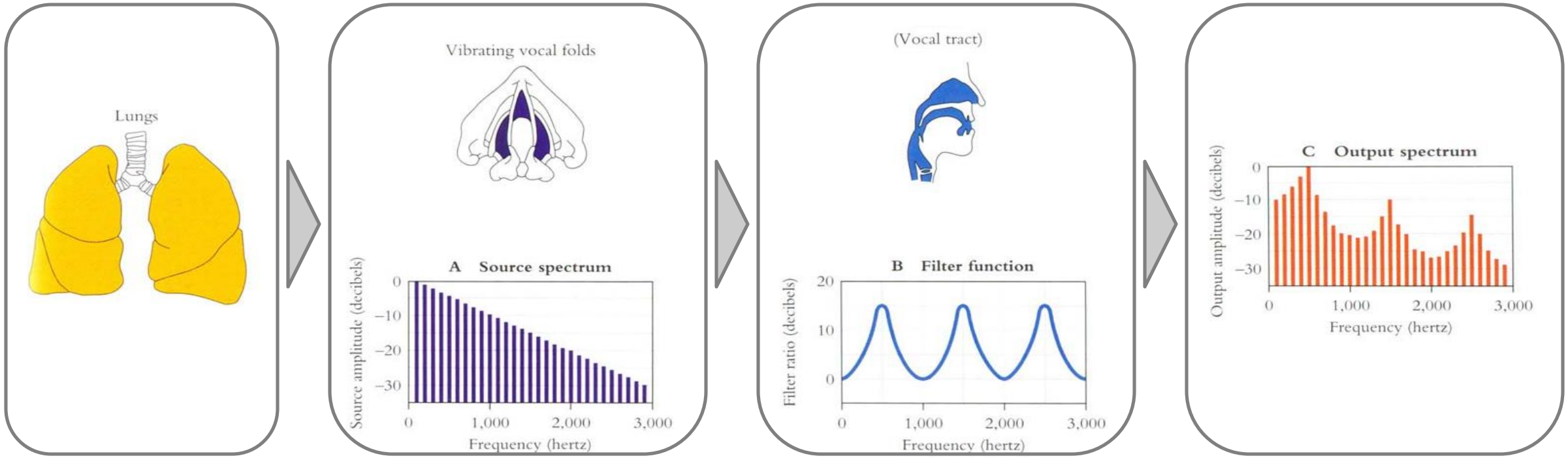


- Complex wave repeating nine times
  - Plus smaller wave that repeats 4x for every large cycle
  - Large wave: freq of 250 Hz (9 times in .036 seconds)
  - Small wave roughly 4 times this, or roughly 1000 Hz



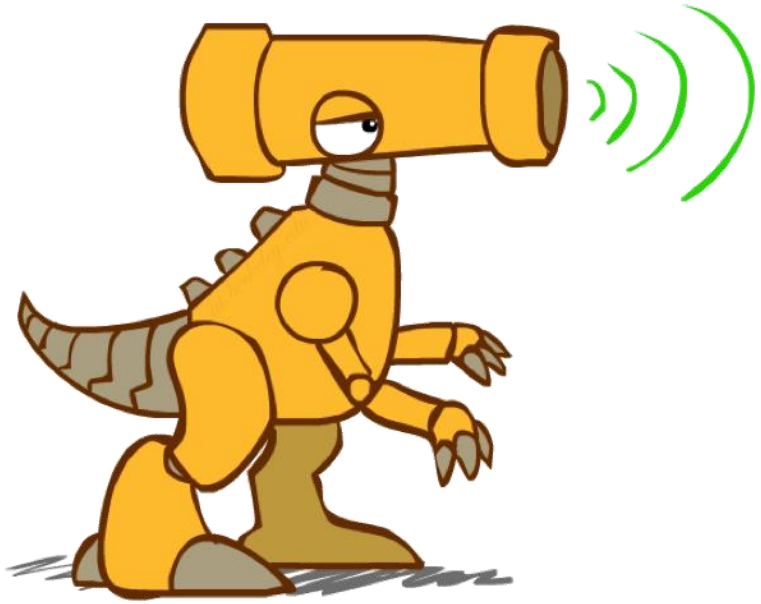
Why These Peaks?

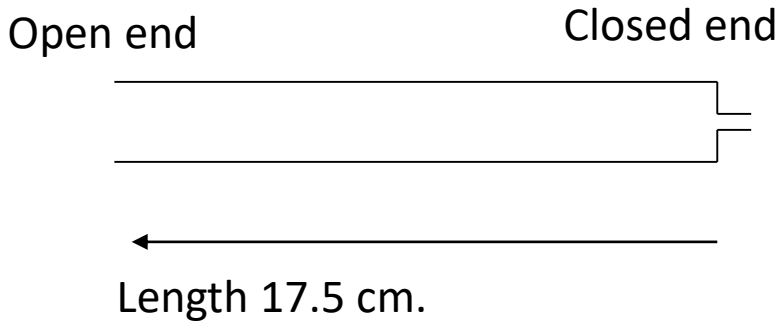
- Articulator process:
  - Vocal cord vibrations create harmonics
  - The mouth is an amplifier
  - Depending on shape of mouth, some harmonics are amplified more than others



Resonances of the Vocal Tract

- The human vocal tract as an open tube





- Air in a tube of a given length will tend to vibrate at resonance frequency of tube
- Constraint: Pressure differential should be maximal at (closed) glottal end and minimal at (open) lip end

Figure: W. Barry Speech Science slides

Spectrum Shapes

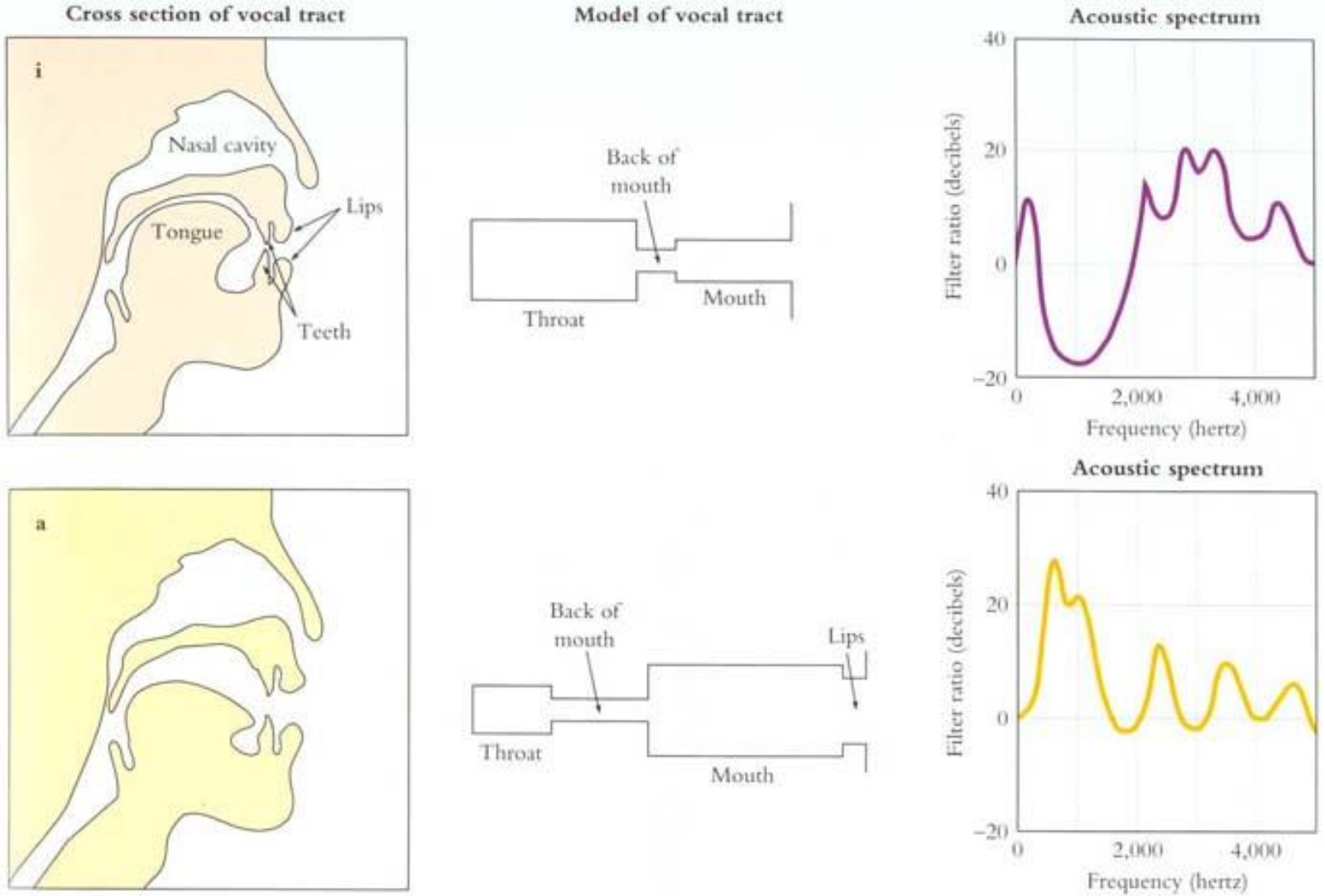
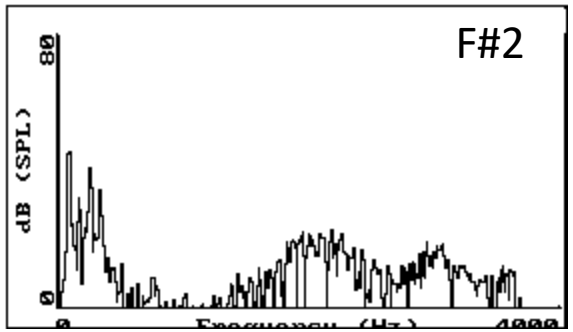
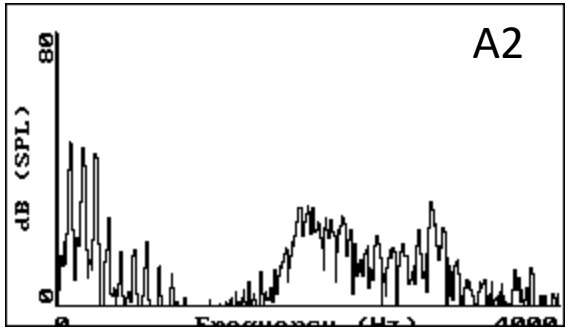


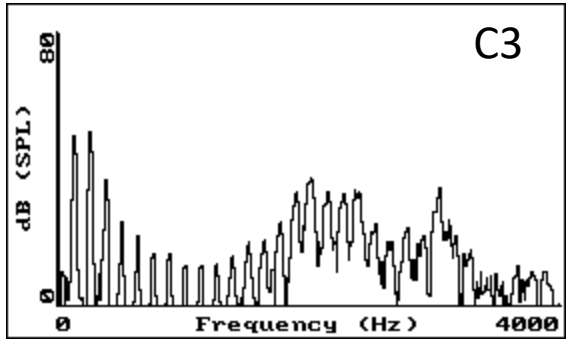
Figure: Mark Liberman

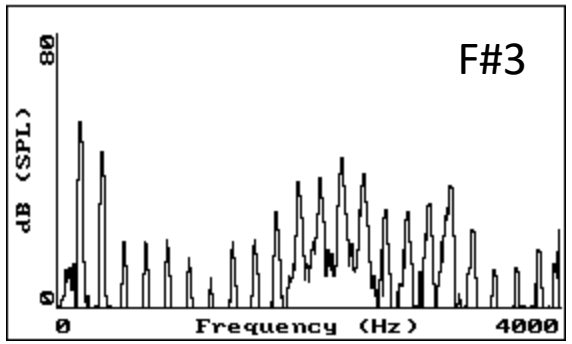
[Demo: speech synthesis]

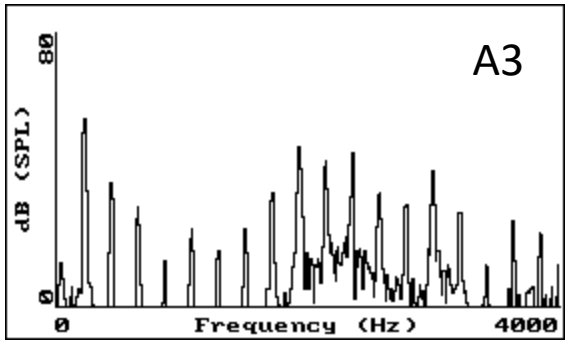
Vowel [i] sung at successively higher pitches

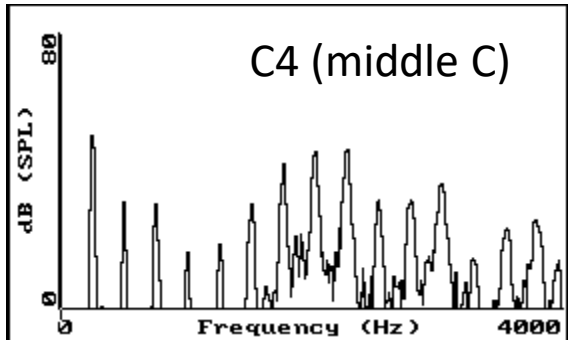


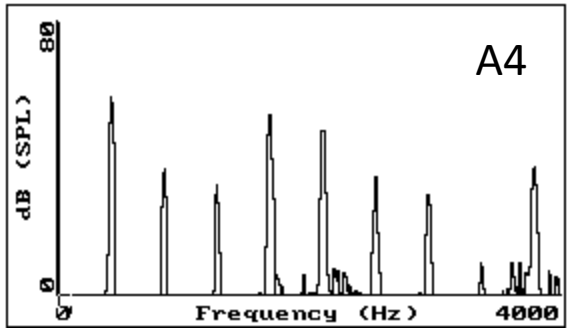


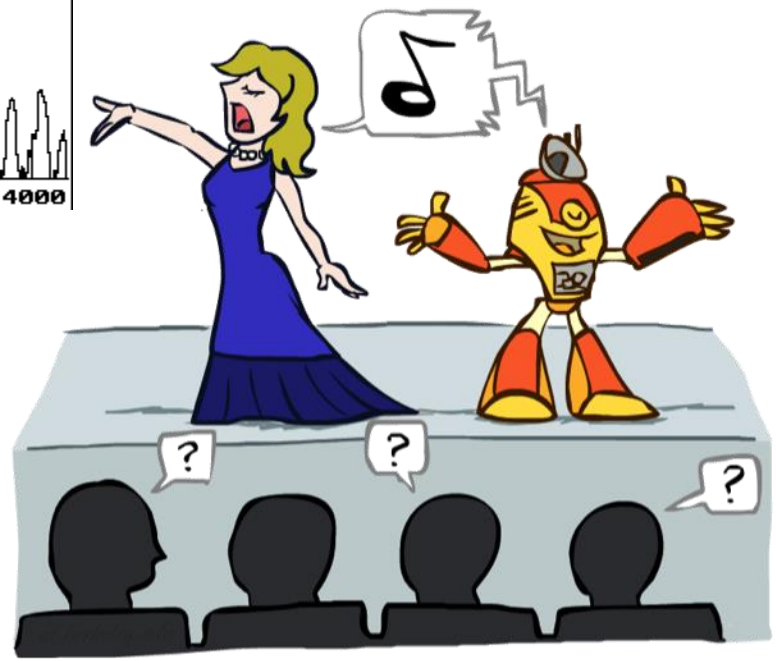








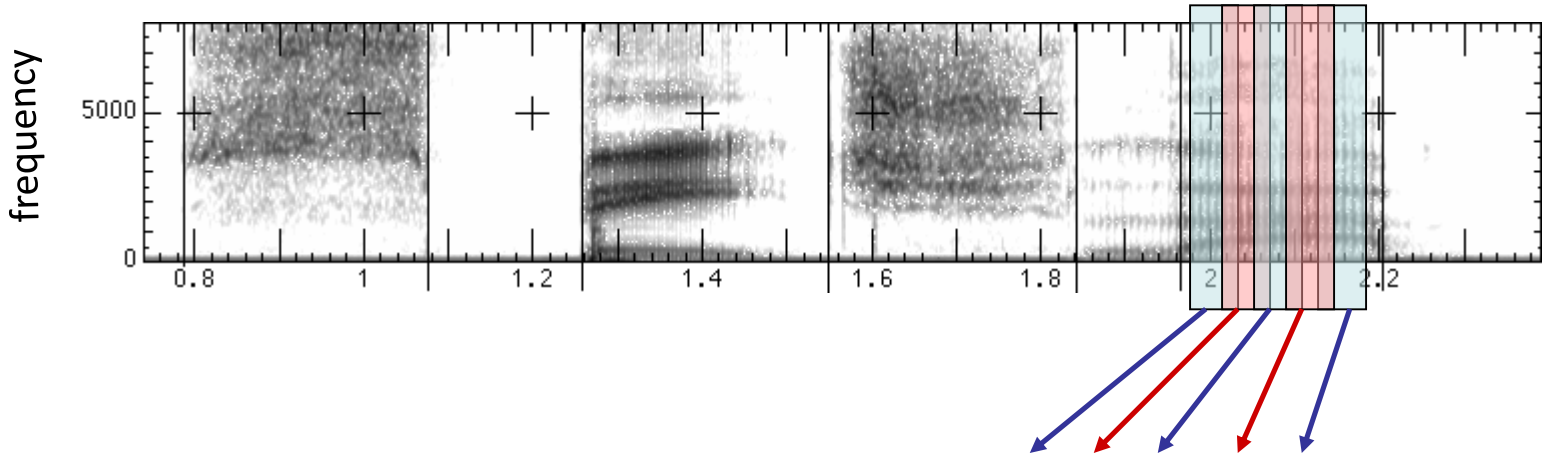




Graphs: Ratree Wayland

Acoustic Feature Sequence

- Time slices are translated into acoustic feature vectors (~39 real numbers per slice)



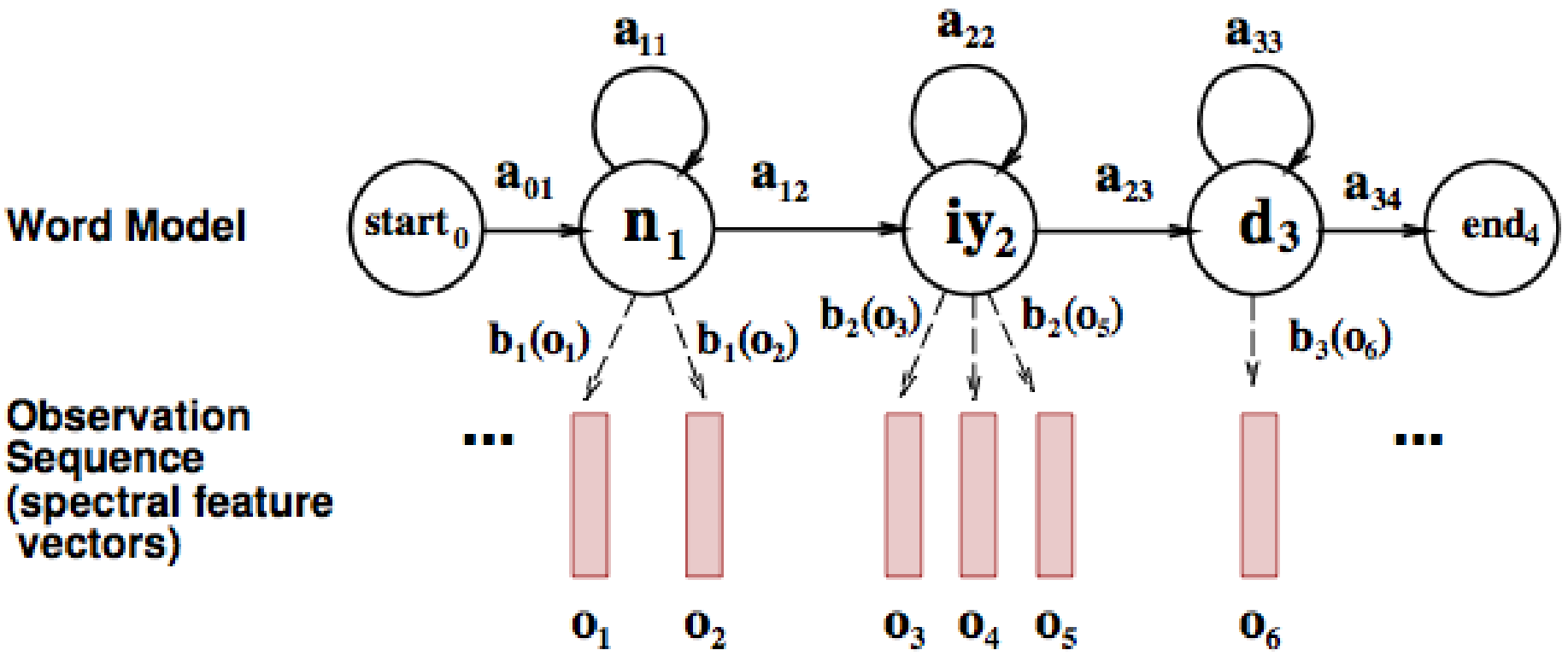
- These are the observations E, now we need the hidden states X

Speech State Space

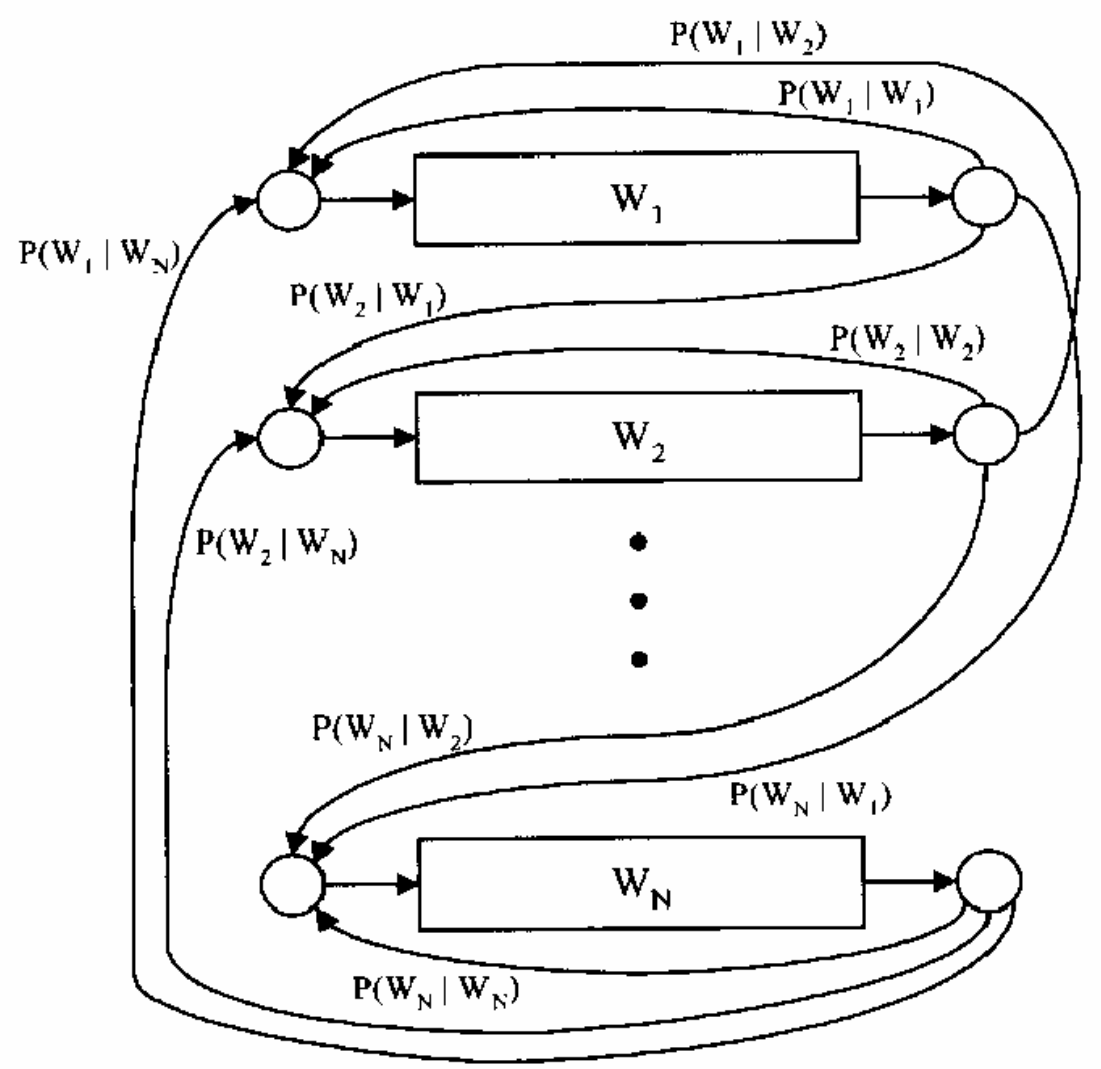
- HMM Specification
  - P(E|X) encodes which acoustic vectors are appropriate for each phoneme (each kind of sound)
  - P(X|X') encodes how sounds can be strung together

- State Space
  - We will have one state for each sound in each word
  - Mostly, states advance sound by sound
  - Build a little state graph for each word and chain them together to form the state space X

States in a Word



Transitions with a Bigram Model



Training Counts

198015222 the first
194623024 the same
168504105 the following
158562063 the world
...
14112454 the door

23135851162 the *

$$\hat{P}(\text{door}|\text{the}) = \frac{14112454}{23135851162}$$

$$= 0.0006$$

Figure: Huang et al, p. 618

Decoding

- Finding the words given the acoustics is an HMM inference problem
- Which state sequence $x_{1:T}$ is most likely given the evidence $e_{1:T}$?

$$x^*_{1:T} = \arg\max_{x_{1:T}} P(x_{1:T}|e_{1:T}) \;\; = \arg\max_{x_{1:T}} P(x_{1:T}, e_{1:T})$$

- From the sequence x, we can simply read off the words

Kalman Filters

- Linear Dynamical Systems:
  - In cases where variables are real-valued
- Dynamic Bayesian Network w/ Continuous Variables and Gaussian Dependencies
- Kalman Filters
  - Heart of tracking systems
  - Radar