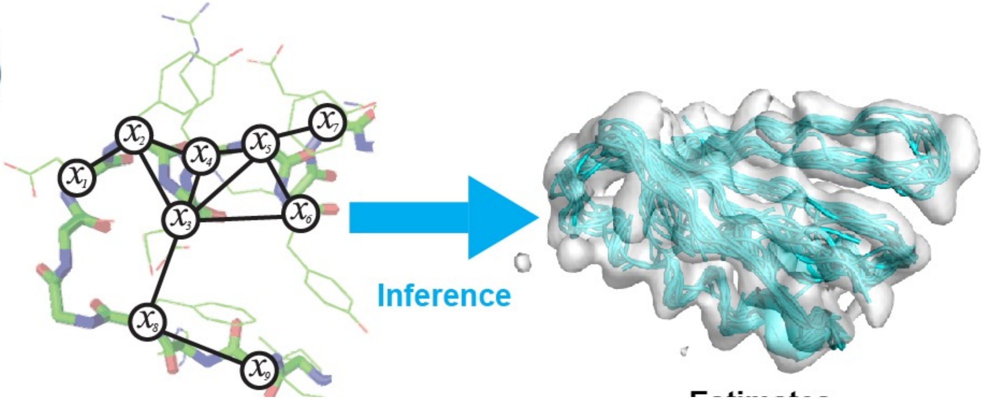
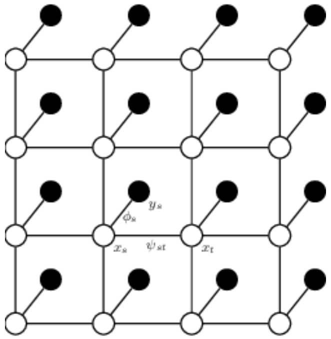
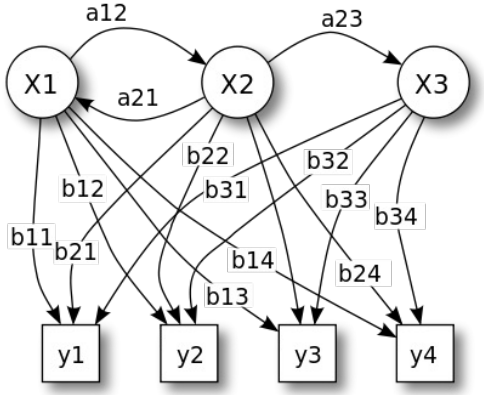
AI3603: Artificial Intelligence: Principles and Applications

Probabilistic Graphical Model III : Markov Chain & HMM

Yue Gao
Shanghai Jiao Tong University



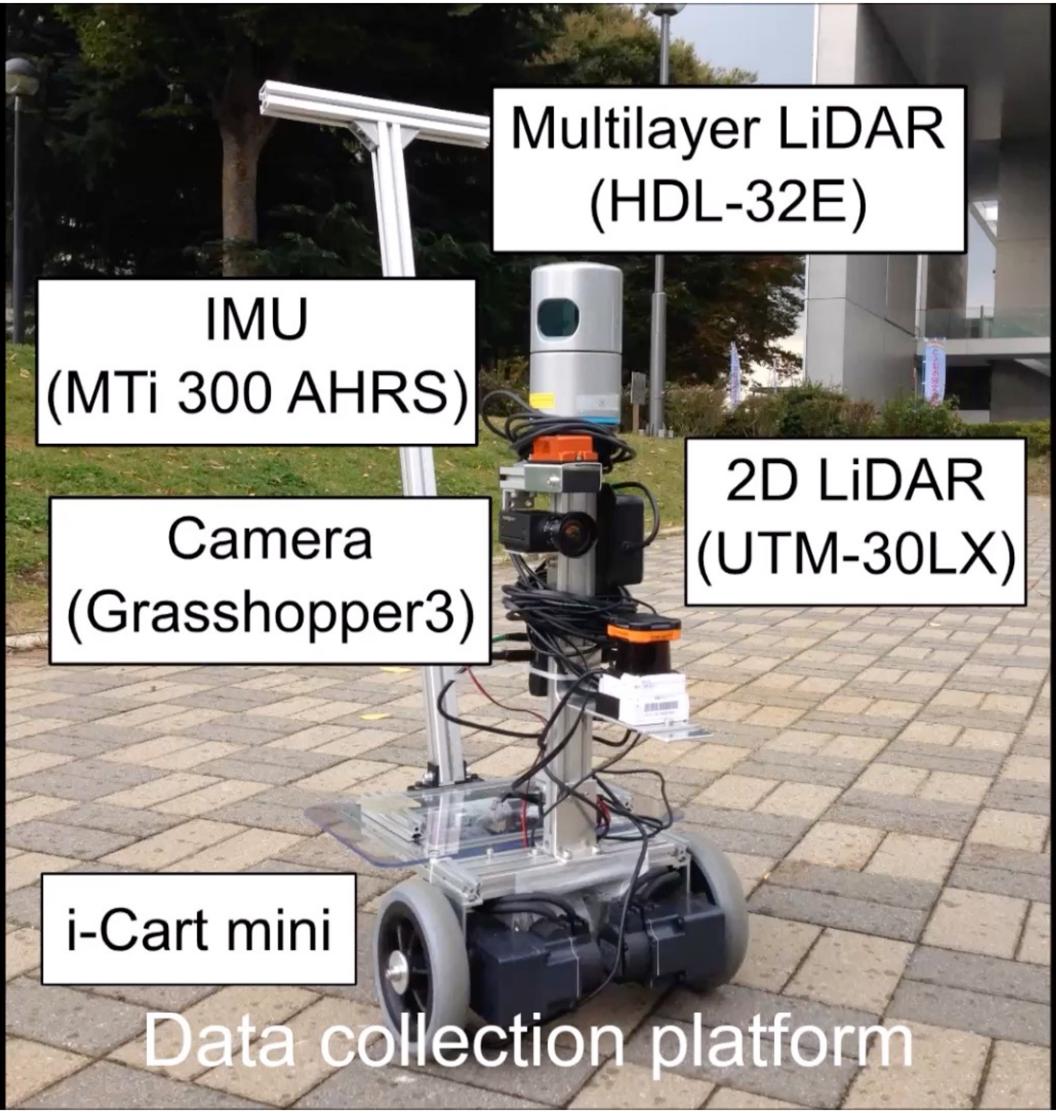




Reasoning over Time or Space

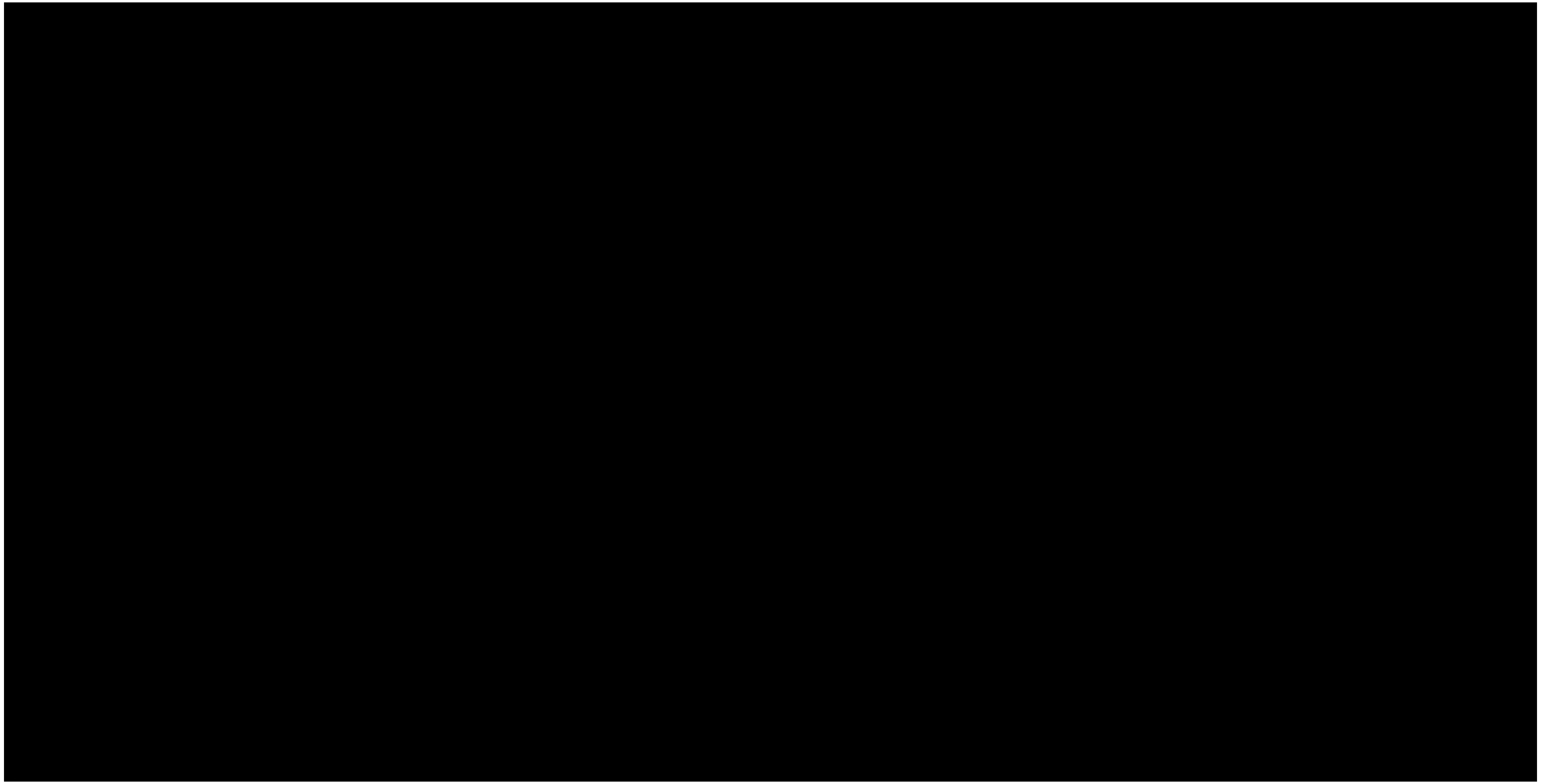
- Often, we want to reason about a sequence of observations
  - Speech recognition
  - Robot localization
  - Medical monitoring
- Need to introduce time (or space) into our models

Robot Localization

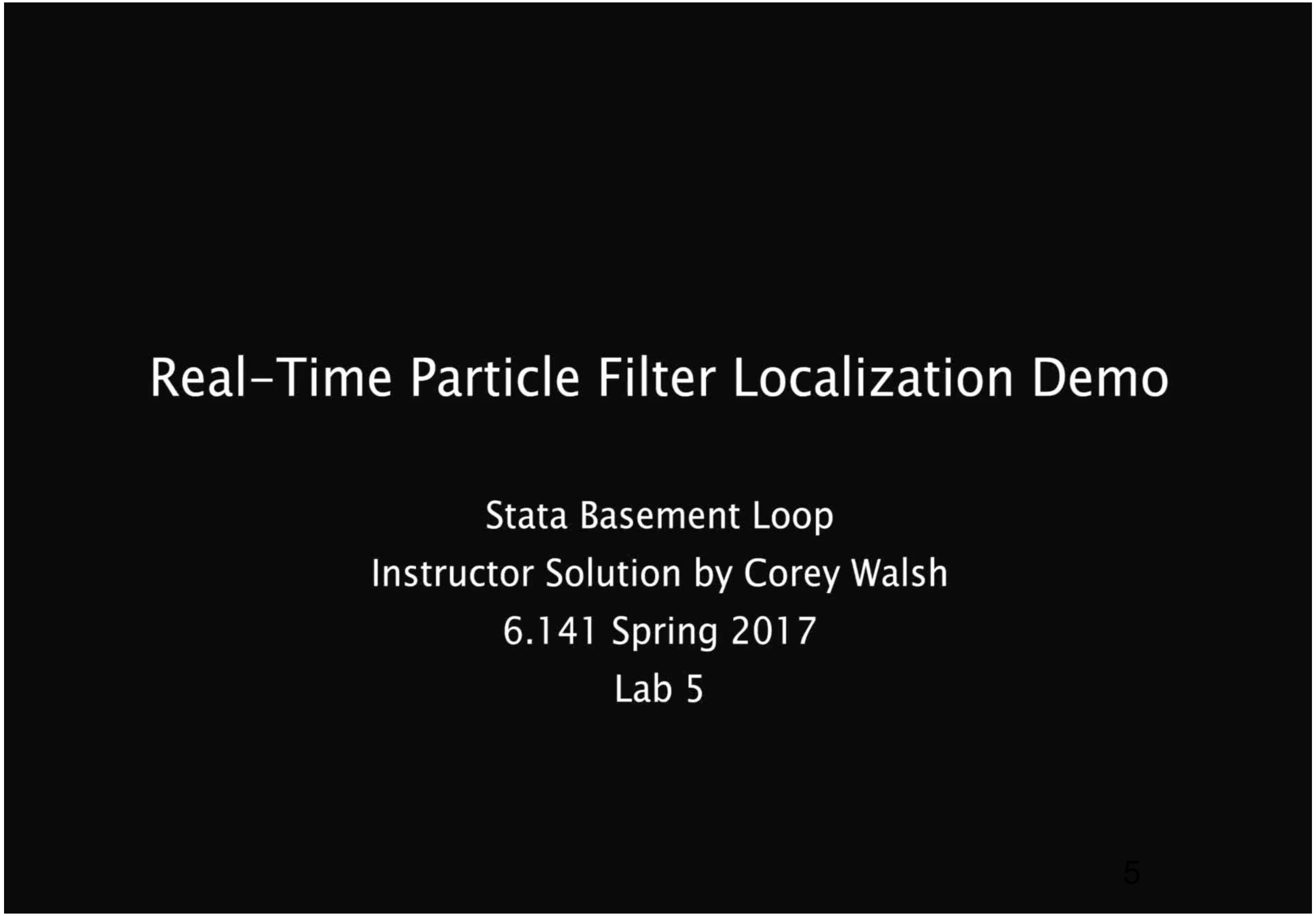


SLAM

- SLAM（Simultaneous Localization And Mapping）



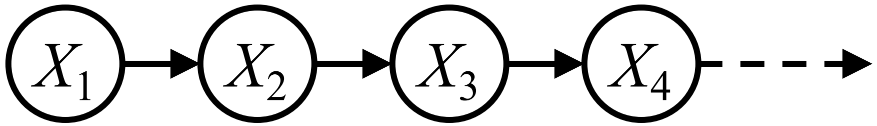
Robot Localization



Contents

Markov Models

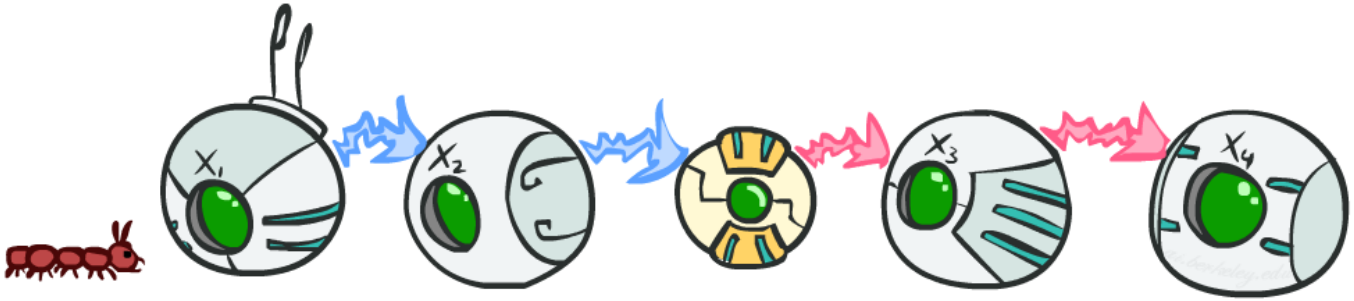
- Value of X at a given time is called the state



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

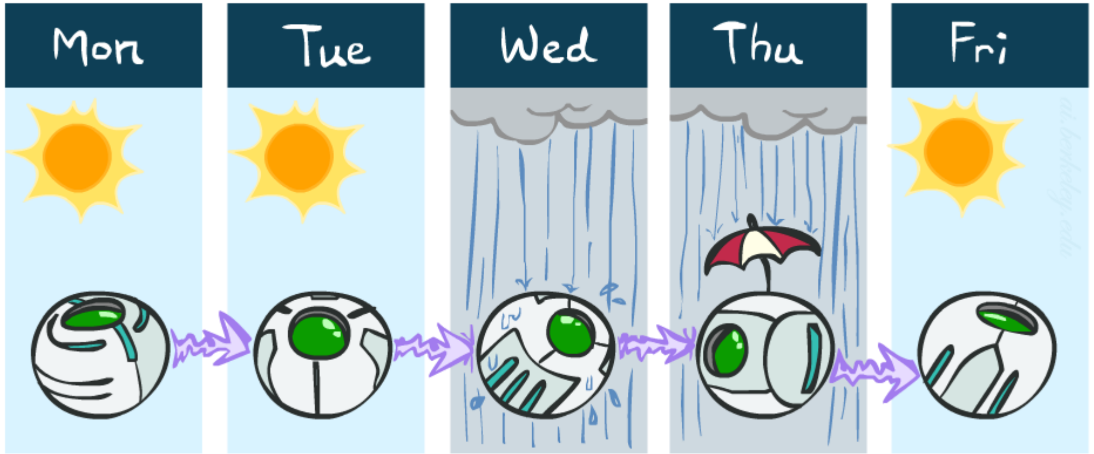
- Parameters: called transition probabilities or dynamics, specify how the state evolves over time (also, initial state probabilities)
- Stationarity assumption: transition probabilities the same at all times
- Same as MDP transition model, but no choice of action

Conditional Independence



- Basic conditional independence:
  - Past and future independent given the present(**D-separation)**
  - Each time step only depends on the previous
  - This is called the (first order) Markov property
  - For second order Markov, combine current with previous state as a new state X

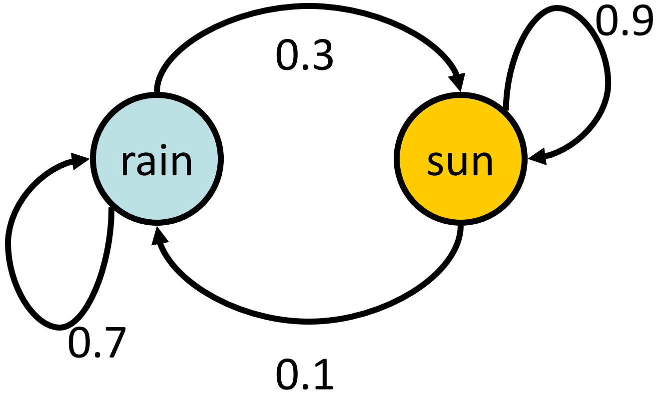
Example Markov Chain: Weather

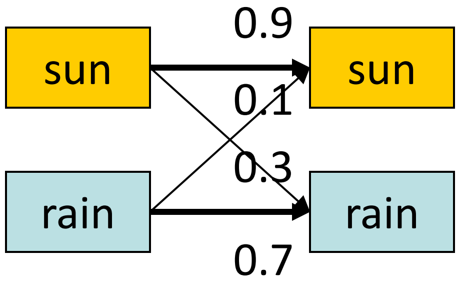


- States: X = {rain, sun}
- Initial distribution: 1.0 sun
- CPT $P(X_t \mid X_{t-1})$:

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

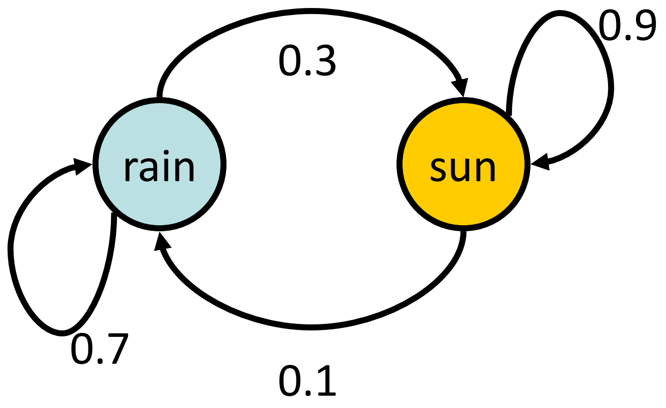




Example Markov Chain: Weather

- Initial distribution: 1.0 sun

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



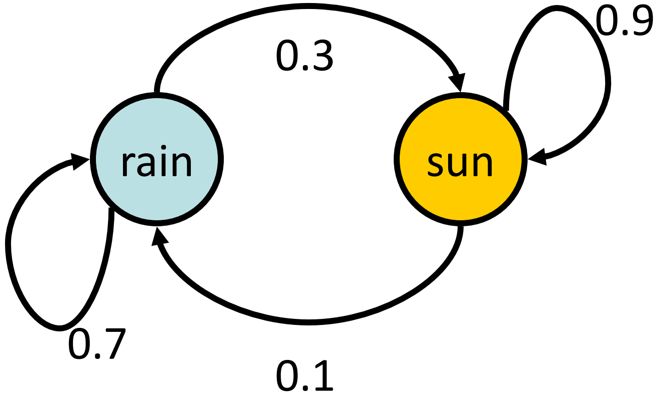
- What is the probability distribution after one step?

$$P(X_2 = sun)$$

Example Markov Chain: Weather

- Initial distribution: 1.0 sun

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



- What is the probability distribution after one step?

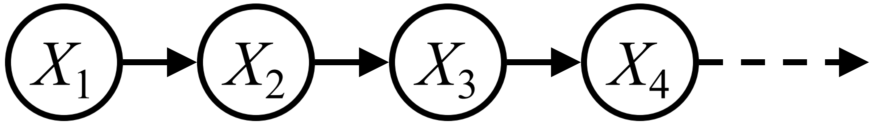
$$P(X_2 = sun) = \sum_{x_1} P(x_1, X_2 = sun) = \sum_{x_1} P(X_2 = sun|x_1)P(x_1)$$

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

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

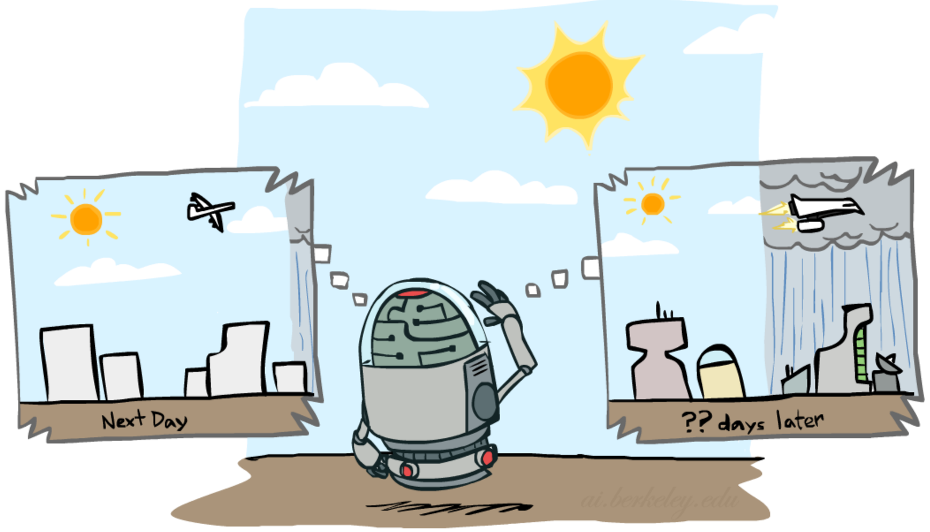
Mini-Forward Algorithm

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



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

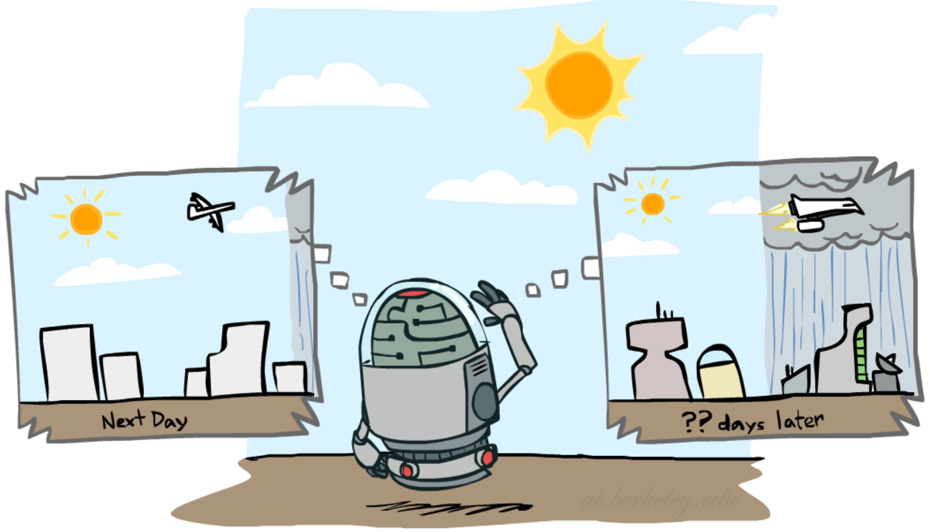
$$P(x_t) =$$



Mini-Forward Algorithm

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

$$X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \dashrightarrow$$



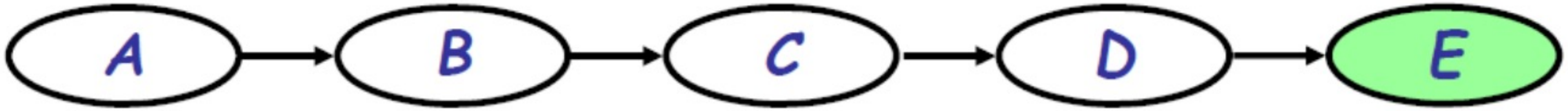
$$P(x_1) = \text{known}$$

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

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

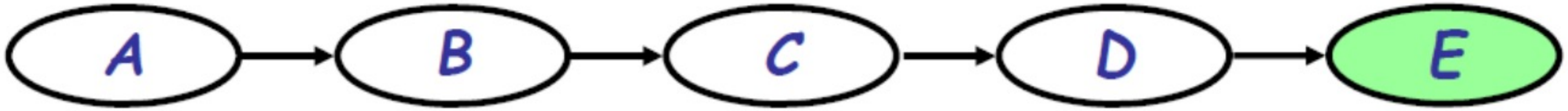
Forward simulation

Mini-Forward Algorithm: Variable Elimination



$$P(e) = \sum_{a,b,c,d} P(a,b,c,d)$$
$$= \sum_{a,b,c,d} P(a)P(b|a)P(c|b)P(d|c)P(e|d)$$

Mini-Forward Algorithm: Variable Elimination



$$P(e) = \sum_{a,b,c,d} P(a,b,c,d)$$
$$= \sum_{a,b,c,d} P(a)P(b|a)P(c|b)P(d|c)P(e|d)$$

$$P(e) = \sum_{a,b,c,d} P(a)P(b|a)P(c|b)P(d|c)P(e|d)$$
$$= \sum_{b,c,d} P(c|b)P(d|c)P(e|d) \sum_{a} P(a)P(b|a)$$
$$= \sum_{b,c,d} P(c|b)P(d|c)P(e|d)P(b)$$

$$P(e) = \sum_{b,c,d} P(c|b)P(d|c)P(e|d)P(b)$$
$$= \sum_{c,d} P(d|c)P(e|d) \sum_{b} P(c|b)P(b)$$
$$= \sum_{c,d} P(d|c)P(e|d)P(c)$$

$$P(e) = \sum_{c,d} P(d|c)P(e|d)P(c)$$
$$= \sum_{d} P(e|d) \sum_{c} P(d|c)P(c)$$
$$= \sum_{d} P(e|d)P(d)$$

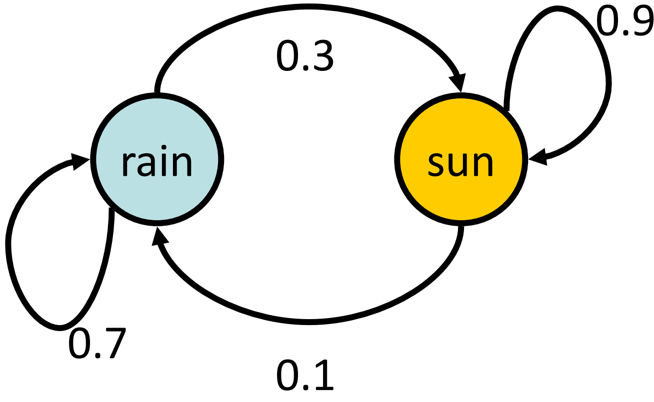
Example Run of Mini-Forward Algorithm

- From initial observation of sun

$$\underset{P(X_1)}{\left\langle \begin{array}{c} 1.0 \\ 0.0 \end{array} \right\rangle} \quad \underset{P(X_2)}{\left\langle \begin{array}{c} 0.9 \\ 0.1 \end{array} \right\rangle} \quad \underset{P(X_3)}{\left\langle \begin{array}{c} 0.84 \\ 0.16 \end{array} \right\rangle} \quad \underset{P(X_4)}{\left\langle \begin{array}{c} 0.804 \\ 0.196 \end{array} \right\rangle} \Rightarrow \underset{P(X_\infty)}{\left\langle \begin{array}{c} 0.75 \\ 0.25 \end{array} \right\rangle}$$

- From initial observation of rain

$$\underset{P(X_1)}{\left\langle \begin{array}{c} 0.0 \\ 1.0 \end{array} \right\rangle} \quad \underset{P(X_2)}{\left\langle \begin{array}{c} 0.3 \\ 0.7 \end{array} \right\rangle} \quad \underset{P(X_3)}{\left\langle \begin{array}{c} 0.48 \\ 0.52 \end{array} \right\rangle} \quad \underset{P(X_4)}{\left\langle \begin{array}{c} 0.588 \\ 0.412 \end{array} \right\rangle} \Rightarrow \underset{P(X_\infty)}{\left\langle \begin{array}{c} 0.75 \\ 0.25 \end{array} \right\rangle}$$



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

$$\underset{P(X_1)}{\left\langle \begin{array}{c} p \\ 1-p \end{array} \right\rangle} \quad \ldots \Rightarrow \underset{P(X_\infty)}{\left\langle \begin{array}{c} 0.75 \\ 0.25 \end{array} \right\rangle}$$

Contents

Stationary Distributions

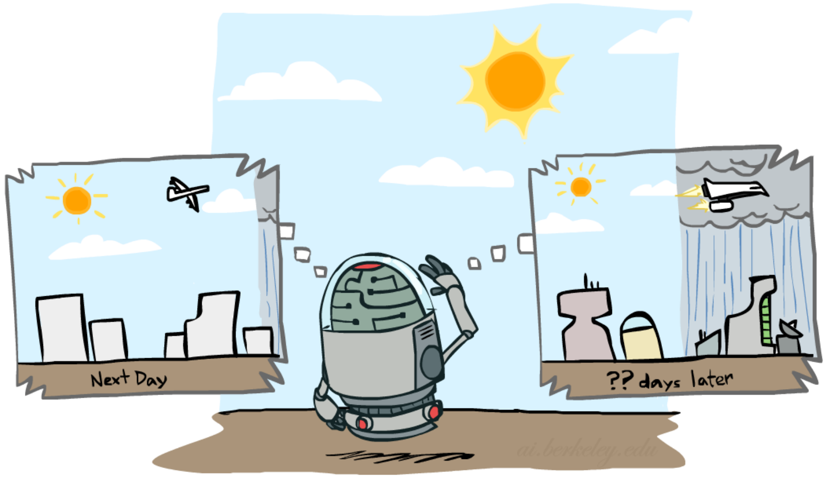
- For most chains:
  - Influence of the initial distribution gets less and less over time.
  - The distribution we end up in is independent of the initial distribution
- Stationary distribution:
  - The distribution we end up with is called the stationary distribution P_∞ of the chain
  - It satisfies

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

Example: Stationary Distributions

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

$$X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \dashrightarrow$$



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

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

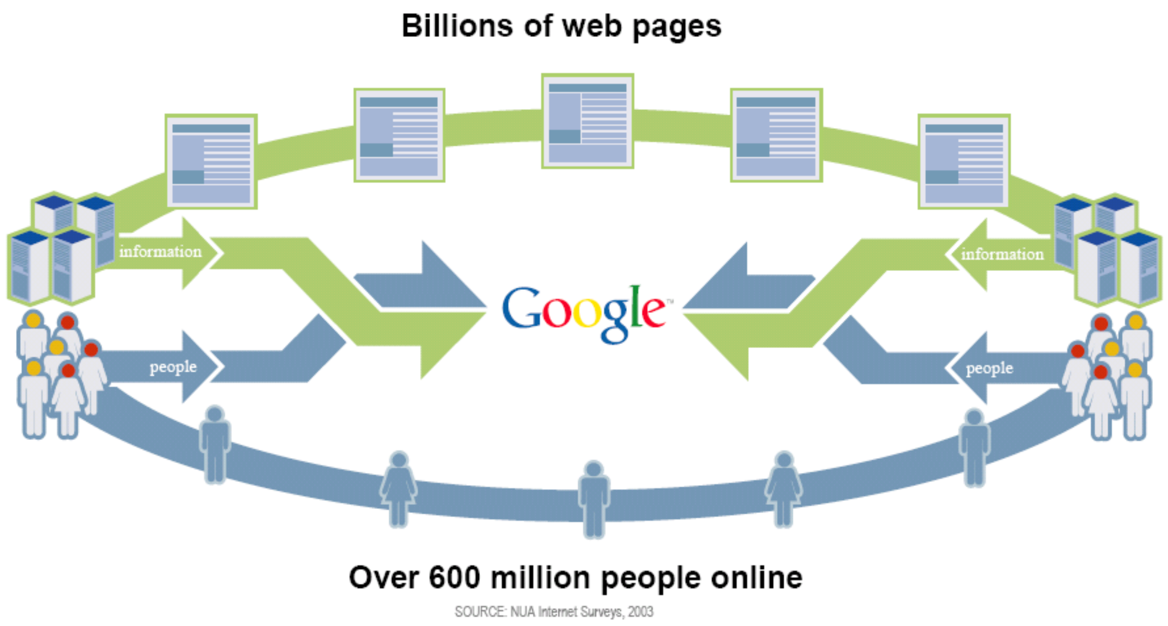
$$P_\infty(sun) = 3P_\infty(rain)$$
$$P_\infty(rain) = 1/3P_\infty(sun)$$

Also: $P_\infty(sun) + P_\infty(rain) = 1$ $\Rightarrow$ $P_\infty(sun) = 3/4$, $P_\infty(rain) = 1/4$

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

Web Search Engine

- Google's Mission： To organize the world's information and make it universally accessible and useful
- Scaling with the web
  - Improved Search Quality
  - Academic Search Engine Research



[redacted]

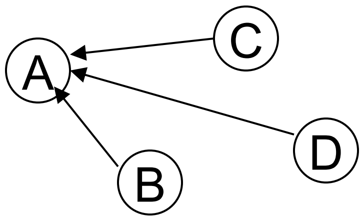
System Features

- It makes use of the link structure of the Web to calculate a quality ranking for each web page, called PageRank
  - PageRank is a trademark of Google. The PageRank process has been patented.
- Google utilizes link to improve search results

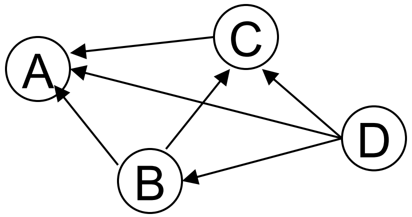
Application of Stationary Distribution: Web Link Analysis

- PageRank is a link analysis algorithm which assigns a numerical weighting to each Web page, with the purpose of "measuring" relative importance.
- Based on the hyperlinks map
- An excellent way to prioritize the results of web keyword searches

Simplified PageRank algorithm

- Assume four web pages: A, B,C and D. Let each page would begin with an estimated PageRank of 0.25.
- 

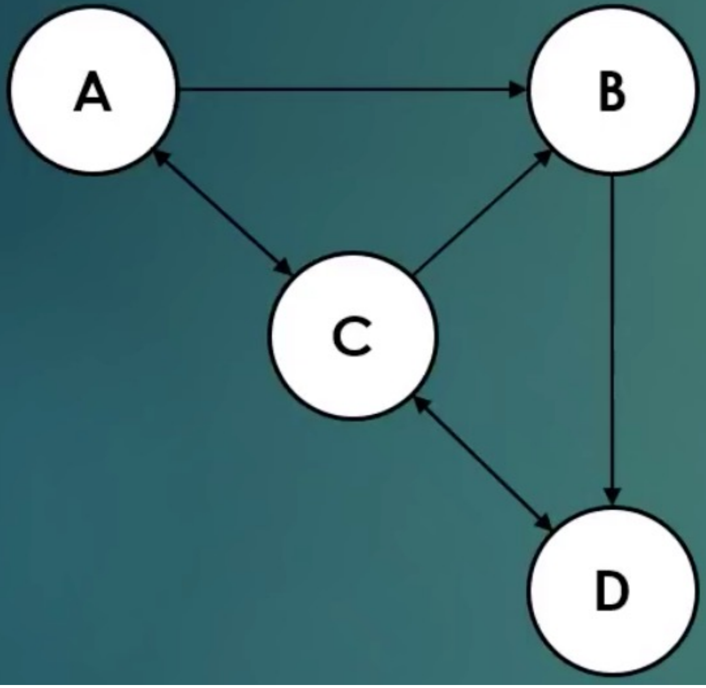
$$PR(A) = PR(B) + PR(C) + PR(D).$$



$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}.$$

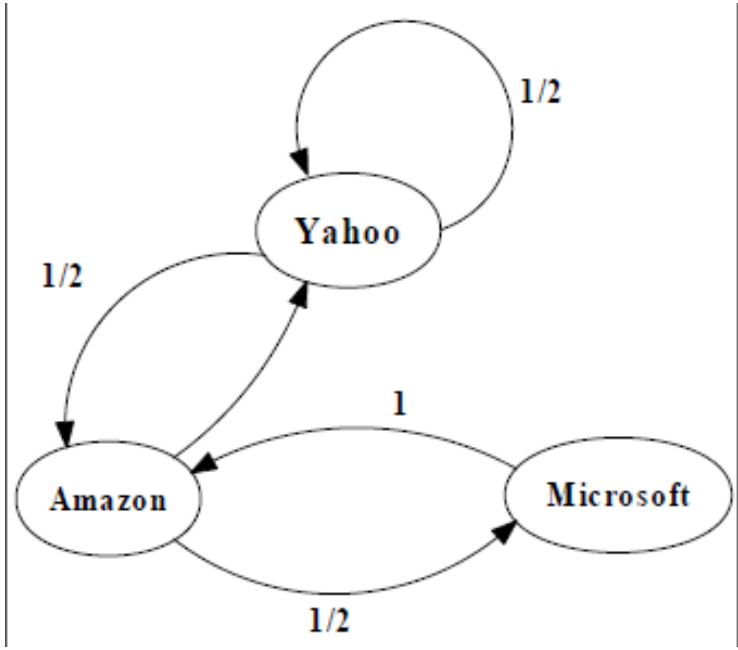
- L(A) is defined as the number of links going out of page A. The PageRank of a page A is given as follows:

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}.$$



	Iteration 0	Iteration 1	Iteration 2	PageRank
A				
B				
C				
D				

An example of Simplified PageRank



M =

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix} = \qquad \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

PageRank Calculation: first iteration

PageRank

The PageRank Citation Ranking: Bringing Order to the Web

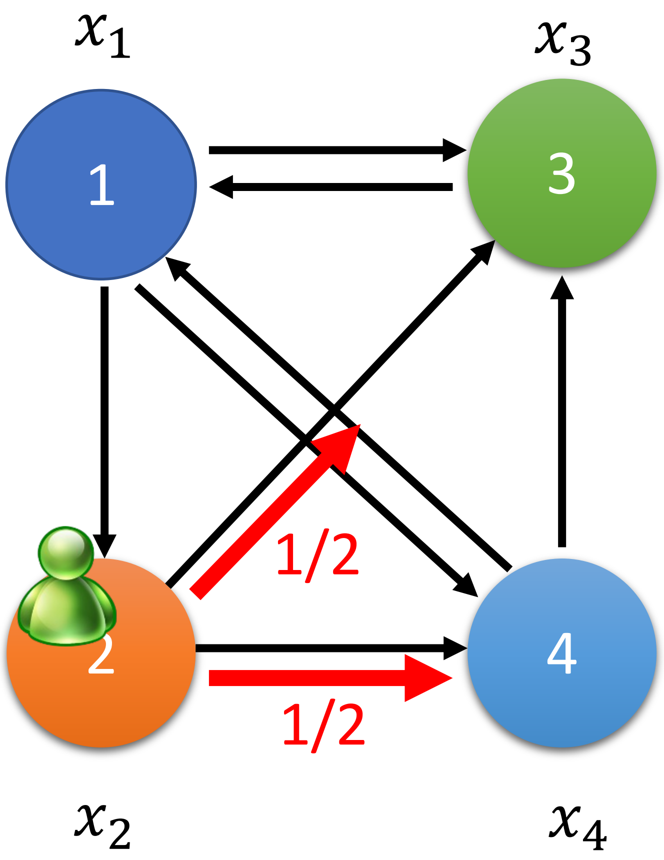
January 29, 1998

Abstract

The importance of a Web page is an inherently subjective matter, which depends on the readers interests, knowledge and attitudes. But there is still much that can be said objectively about the relative importance of Web pages. This paper describes PageRank, a method for rating Web pages objectively and mechanically, effectively measuring the human interest and attention devoted to them.

We compare PageRank to an idealized random Web surfer. We show how to efficiently compute PageRank for large numbers of pages. And, we show how to apply PageRank to search and to user navigation.

PageRank



$$x_1 = x_3 + \frac{1}{2} x_4$$

$$x_2 = \frac{1}{3} x_1$$

$$x_3 = \frac{1}{3} x_1 + \frac{1}{2} x_2 + \frac{1}{2} x_4$$

$$x_4 = \frac{1}{3} x_1 + \frac{1}{2} x_2$$

Consider a random surfer

PageRank

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$\boxed{\begin{aligned} x_1 &= x_3 + \frac{1}{2}x_4 \\ x_2 &= \frac{1}{3}x_1 \\ x_3 &= \frac{1}{3}x_1 + \frac{1}{2}x_2 + \frac{1}{2}x_4 \\ x_4 &= \frac{1}{3}x_1 + \frac{1}{2}x_2 \end{aligned}}$$

$$Ax = x$$

The solution x is in the eigenspace of eigenvalue $\lambda = 1$

Column-stochastic matrix always have eigenvalue $\lambda = 1$

Users scan right to the **bottom** of the pages!

Baidu

Users didn't scan behind the **4th** listing.

Google

Highest Page Rank

PageRank 10

Currently **12** domains

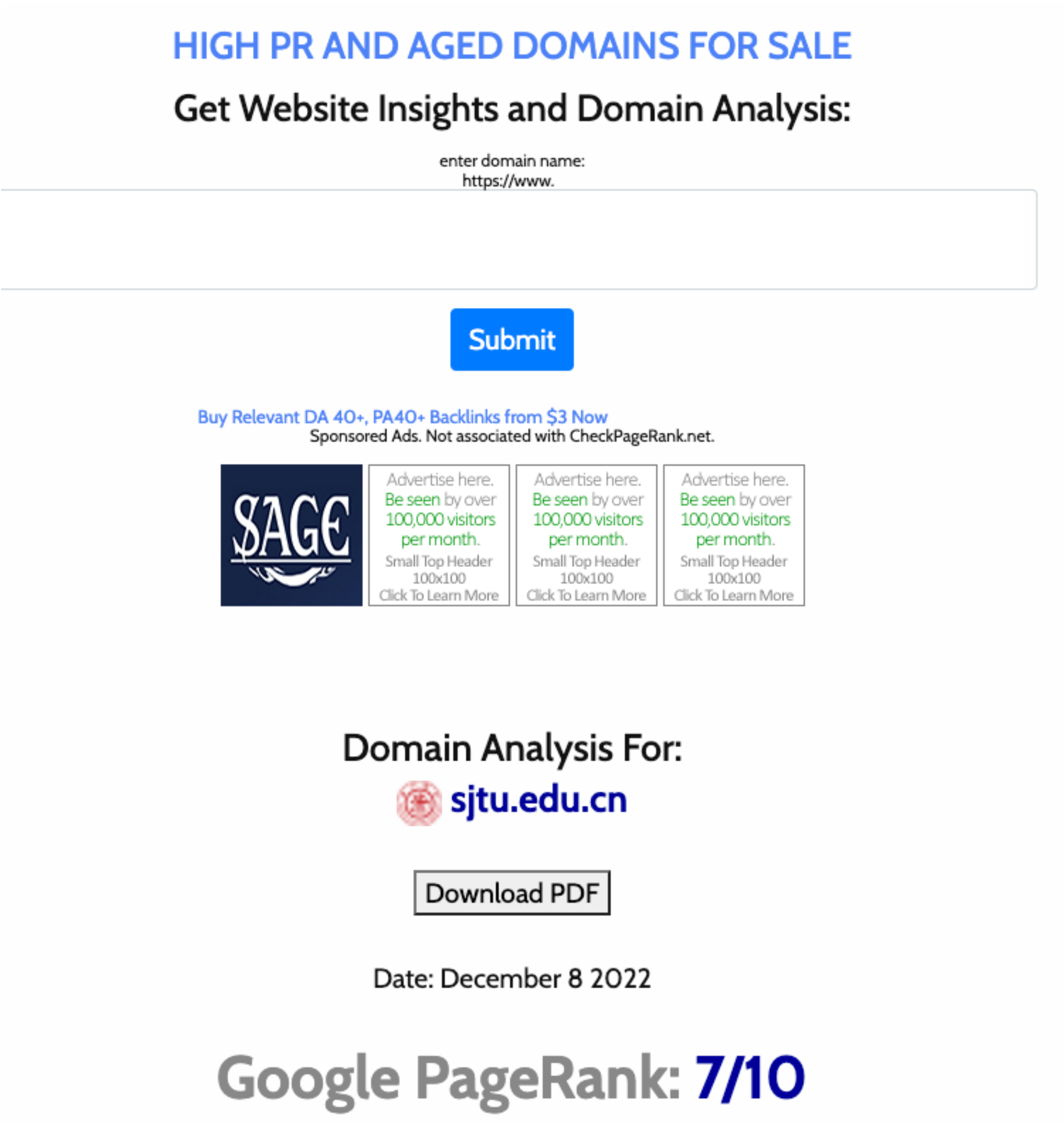
- Google.com - Google, the most popular Internet search engine, and the one that assigns this PageRank we're rambling on about.
- W3.org - World Wide Web Consortium, the organization behind most web specifications like HTML, CSS and so on.
  - http://jigsaw.w3.org/css-validator/ (CSS validator)
- USA.gov - The U.S. Government's official web portal.
- Adobe.com - The main domain is no longer PR10, but these two pages are:
  - get.adobe.com/flashplayer/ (Adobe Flash Player)
  - get.adobe.com/reader/ (Adobe PDF Reader)
- India.gov.in - Government of India
- HHS.gov - United States Department of Health and Human Services
- Recovery.gov - U.S Government recovery board
- TheEuropeanLibrary.org - The European Library - searches the content of European national libraries.
- Europeana.eu - Europeana - "The cultural collections of Europe".
- CNN.com - Cable News Network, American television news channel
- Miibeian.gov.cn - Ministry of Information Industry Records in China
- AddThis.com - AddThis bookmarking/sharing service - this page is linked to by every page using the system (as StatCounter.com used to be), hence the PR10.

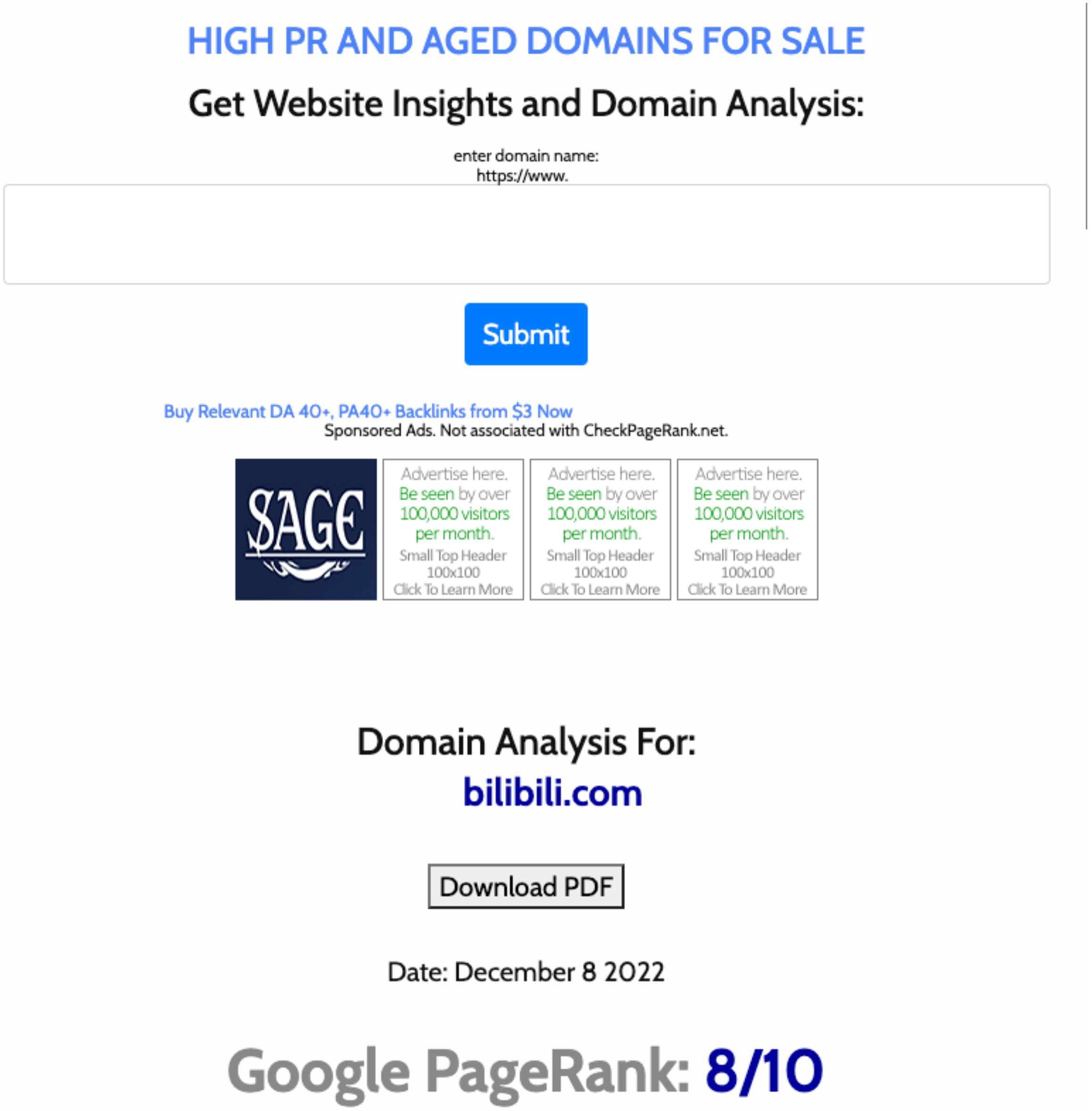
PageRank 9

Currently **148** domains

- AAAS.org - American Association for the Advancement of Science
- ACM.org - Association for Computing Machinery
- AltaVista.com - AltaVista, search engine
- Amazon.com - Amazon.com, shopping site
- AOL.com - America Online, portal
- APS.org - American Physical Society
- Archive.org - Internet Archive
- Arizona.edu - University of Arizona
- ArXiv.org - ArXiv.org e-print archive
- ASU.edu - Arizona State University
- BarnesAndNoble.com - Barnes & Noble, bookseller
- BBC.co.uk - British Broadcasting Corporation
- Berkeley.edu - University of California, Berkeley
- Blogger.com - Blogger, a weblog publishing tool
- Bloglines.com - Bloglines, web-based news aggregator
- Brown.edu - Brown University
- BU.edu - Boston University
- Cam.ac.uk - Cambridge University
- CBC.ca - Canadian Broadcasting Corporation
- CERN.ch - CERN
- CMU.edu - Carnegie Mellon University
- CNET.com - Cnet, technology portal
- CNRS.fr - Centre National de la Recherche Scientifique
- Cisco.com - Cisco Systems
- Colorado.edu - University of Colorado at Boulder
- Columbia.edu - Columbia University
- Computer.org - IEEE Computer Society
- Copyright.gov - U.S. Copyright Office
- Cornell.edu - Cornell University
- CreativeCommons.org - Creative Commons
- Debian.org - Debian, open-source operating system
- DHHS.gov - Department of Health and Human Services
- DHS.gov/dhspublic/ - Department of Homeland Security
- DOI.gov - U.S. Department of the Interior
- Duke.edu - Duke University
- eBay.com - EBay, auction site
- Economist.com - news site
- Elsevier.com - Elsevier, publisher of scientific and medical literature
- EnergyStar.gov - Energy Star
- EPA.gov - Environmental Protection Agency

PageRank

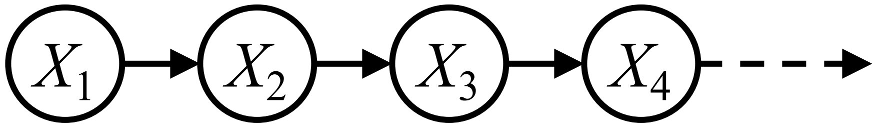

HIGH PR AND AGED DOMAINS FOR SALE
Get Website Insights and Domain Analysis:
enter domain name:
https://www.
Submit
Buy Relevant DA 40+, PA40+ Backlinks from $3 Now
Sponsored Ads. Not associated with CheckPageRank.net.
SAGE
Advertise here. Be seen by over 100,000 visitors per month. Small Top Header 100x100 Click To Learn More
Advertise here. Be seen by over 100,000 visitors per month. Small Top Header 100x100 Click To Learn More
Advertise here. Be seen by over 100,000 visitors per month. Small Top Header 100x100 Click To Learn More
Domain Analysis For:
sjtu.edu.cn
Download PDF
Date: December 8 2022
Google PageRank: 7/10


HIGH PR AND AGED DOMAINS FOR SALE
Get Website Insights and Domain Analysis:
enter domain name:
https://www.
Submit
Buy Relevant DA 40+, PA40+ Backlinks from $3 Now
Sponsored Ads. Not associated with CheckPageRank.net.
SAGE
Advertise here. Be seen by over 100,000 visitors per month. Small Top Header 100x100 Click To Learn More
Advertise here. Be seen by over 100,000 visitors per month. Small Top Header 100x100 Click To Learn More
Advertise here. Be seen by over 100,000 visitors per month. Small Top Header 100x100 Click To Learn More
Domain Analysis For:
bilibili.com
Download PDF
Date: December 8 2022
Google PageRank: 8/10

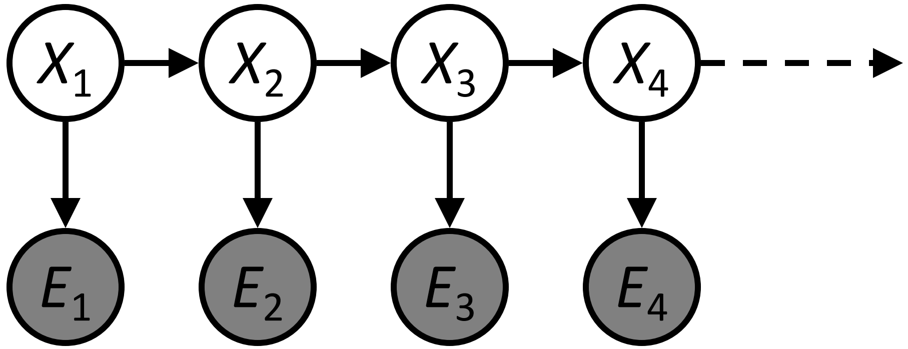
Contents

Hidden Markov Models

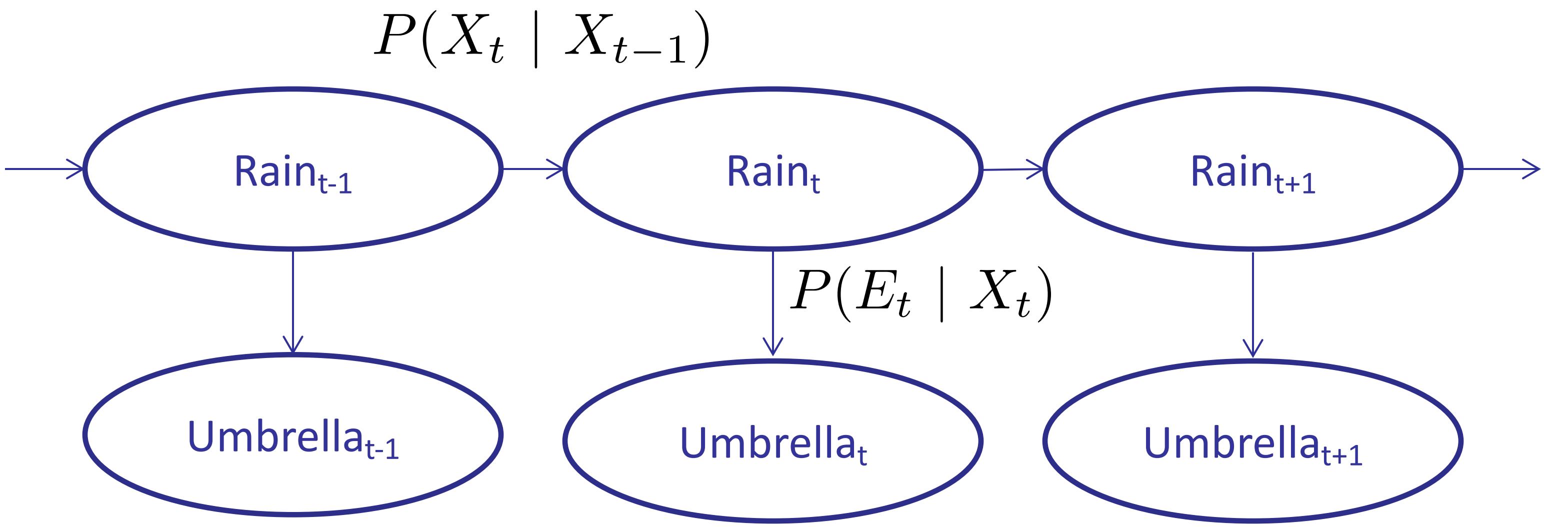
- Markov chains not so useful for most agents
  - Need observations to update your beliefs



- Hidden Markov models (HMMs)
  - Underlying Markov chain over states X
  - You observe outputs (effects) at each time step



Example: Weather HMM



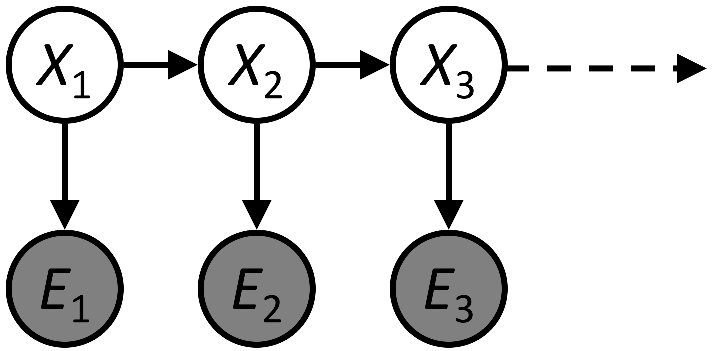
- An HMM is defined by:
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X_t \mid X_{t-1})$
  - Emissions: $P(E_t \mid X_t)$

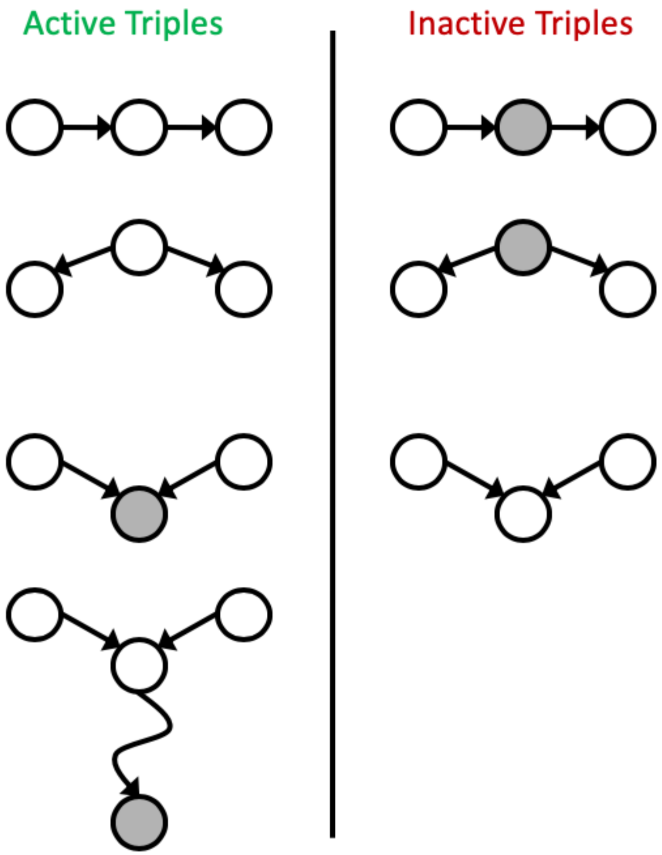
R_{t-1}	R_t	$P(R_t \mid R_{t-1})$
+r	+r	0.7
+r	-r	0.3
-r	+r	0.3
-r	-r	0.7

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

Conditional Independence

- How many Implied conditional independencies?

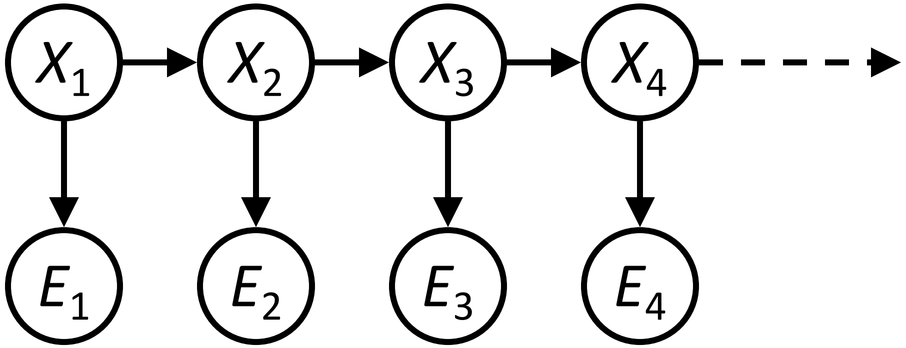




- E1⊥⊥X2, E2,X3,E3|X1

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)

Filtering

- Filtering 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 NASA Apollo program

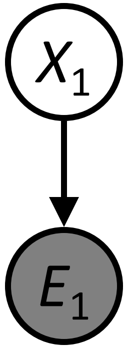
Inference: Find State Given Evidence

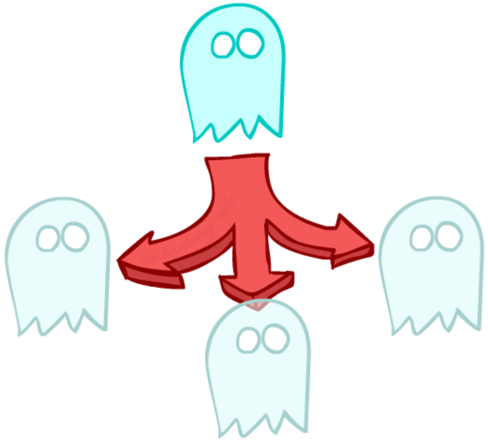
- We are given evidence at each time and want to know

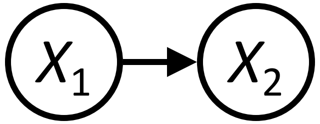
$$B_t(X) = P(X_t | e_{1:t})$$

- Idea: 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

Inference: Base Cases







$$P(X_1|e_1)$$

$$\begin{aligned} P(x_1|e_1) &= P(x_1, e_1)/P(e_1) \\ &\propto_{X_1} P(x_1, e_1) \\ &= P(x_1)P(e_1|x_1) \end{aligned}$$

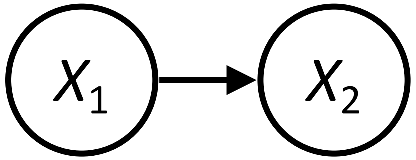
$$P(X_2)$$

$$\begin{aligned} P(x_2) &= \sum_{x_1} P(x_1, x_2) \\ &= \sum_{x_1} P(x_1)P(x_2|x_1) \end{aligned}$$

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:

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

- Or compactly:

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

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

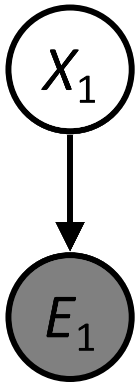
Observation

- 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:

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



- 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: Weather HMM

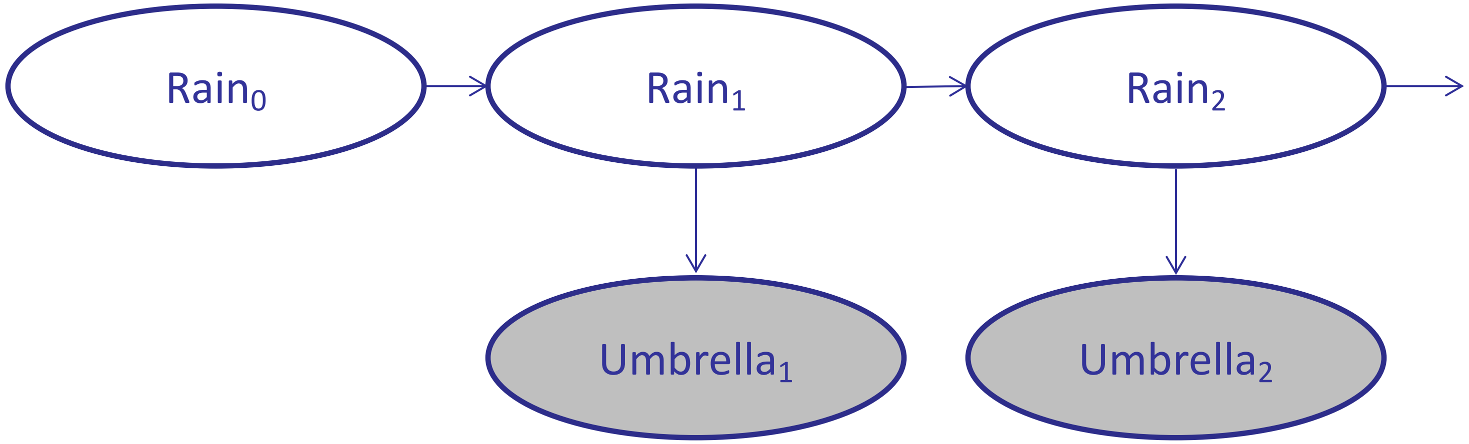
$B(+r) = 0.5$
$B(-r) = 0.5$

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

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

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

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



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

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

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

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

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

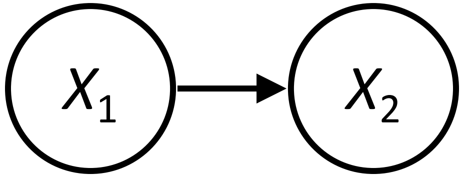
$$= \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1}) P(x_t | x_{t-1}) P(e_t | x_t)$$

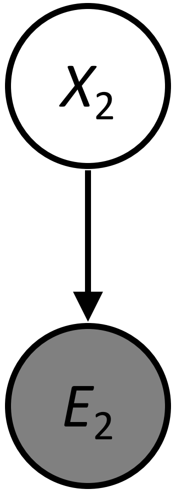
$$= P(e_t | x_t) \sum_{x_{t-1}} P(x_t | x_{t-1}) P(x_{t-1}, e_{1:t-1})$$

Online Belief Updates

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

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

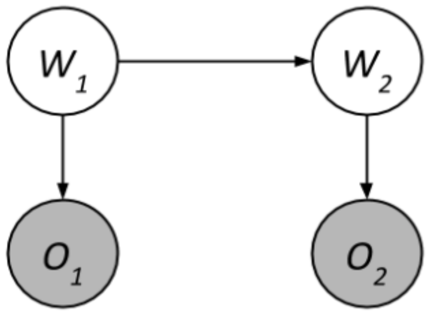




- We update for evidence:

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

- The forward algorithm doesn't normalize



W_1	$P(W_1)$
0	0.3
1	0.7

W_t	W_{t+1}	$P(W_{t+1}\|W_t)$
0	0	0.4
0	1	0.6
1	0	0.8
1	1	0.2

W_t	O_t	$P(O_t\|W_t)$
0	a	0.9
0	b	0.1
1	a	0.5
1	b	0.5

Suppose that we observe $O_1 = a$ and $O_2 = b$.
Using the forward algorithm, compute the probability distribution $P(W_2|O_1 = a, O_2 = b)$ one step at a time.

1- Compute P(W1,O1=a)

2 - Compute P(W2,O1=a)

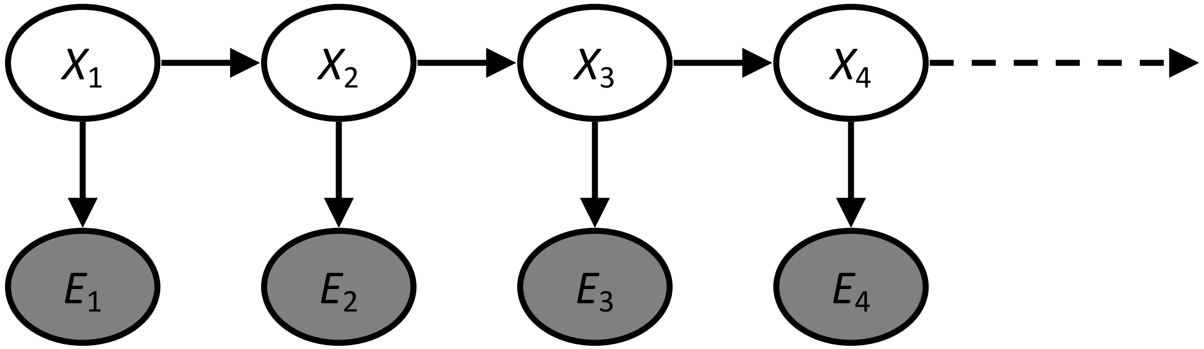
3- Compute P(W2, O1=a,O2=b)

P(W2, O1=a, O2 = b) / P(O1=a, O2 = b)

4- Compute P(W2|O1=a,O2=b)

HMMs: Most Likely Explanation Queries

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

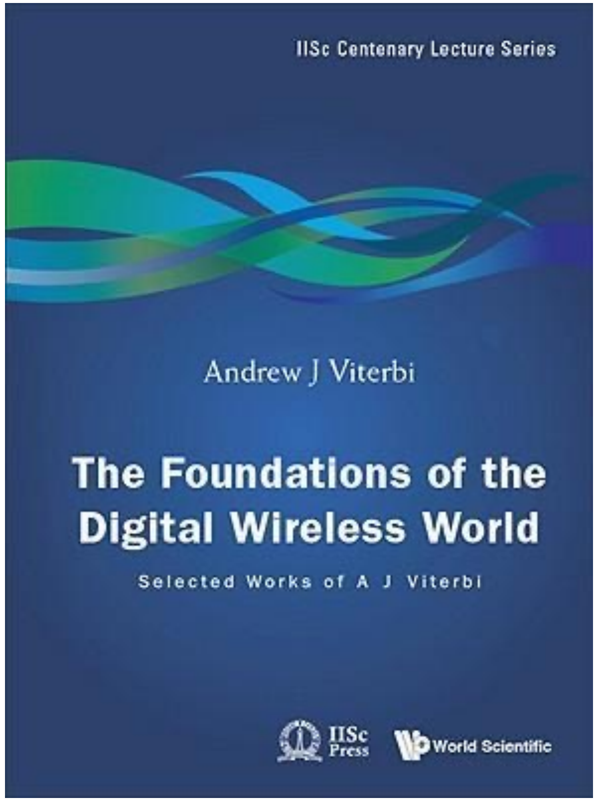


- New query: most likely explanation: $\arg\max_{x_{1:t}} P(x_{1:t}|e_{1:t})$

- New method: the Viterbi algorithm

Andrew Viterbi

- The Viterbi algorithm 1968
- The key to communications
- Co-founder of Qualcomm



HMMs: Most Likely Explanation Queries

Figure 1: Example observation from data set: actual word is "commanding" with the first letter removed

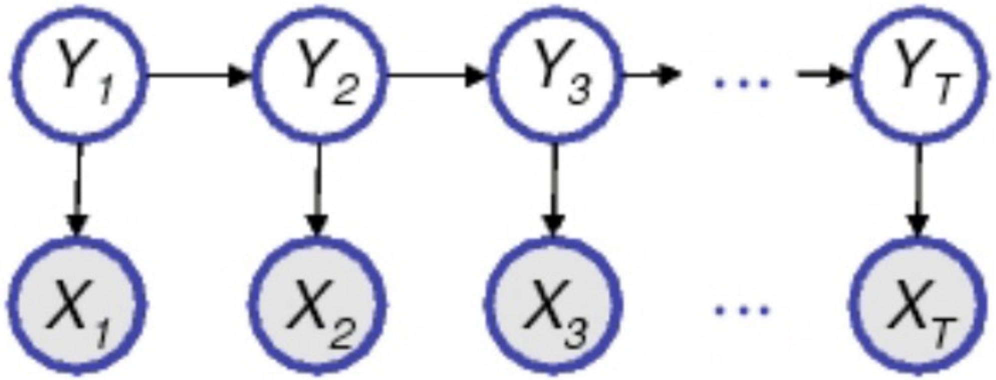
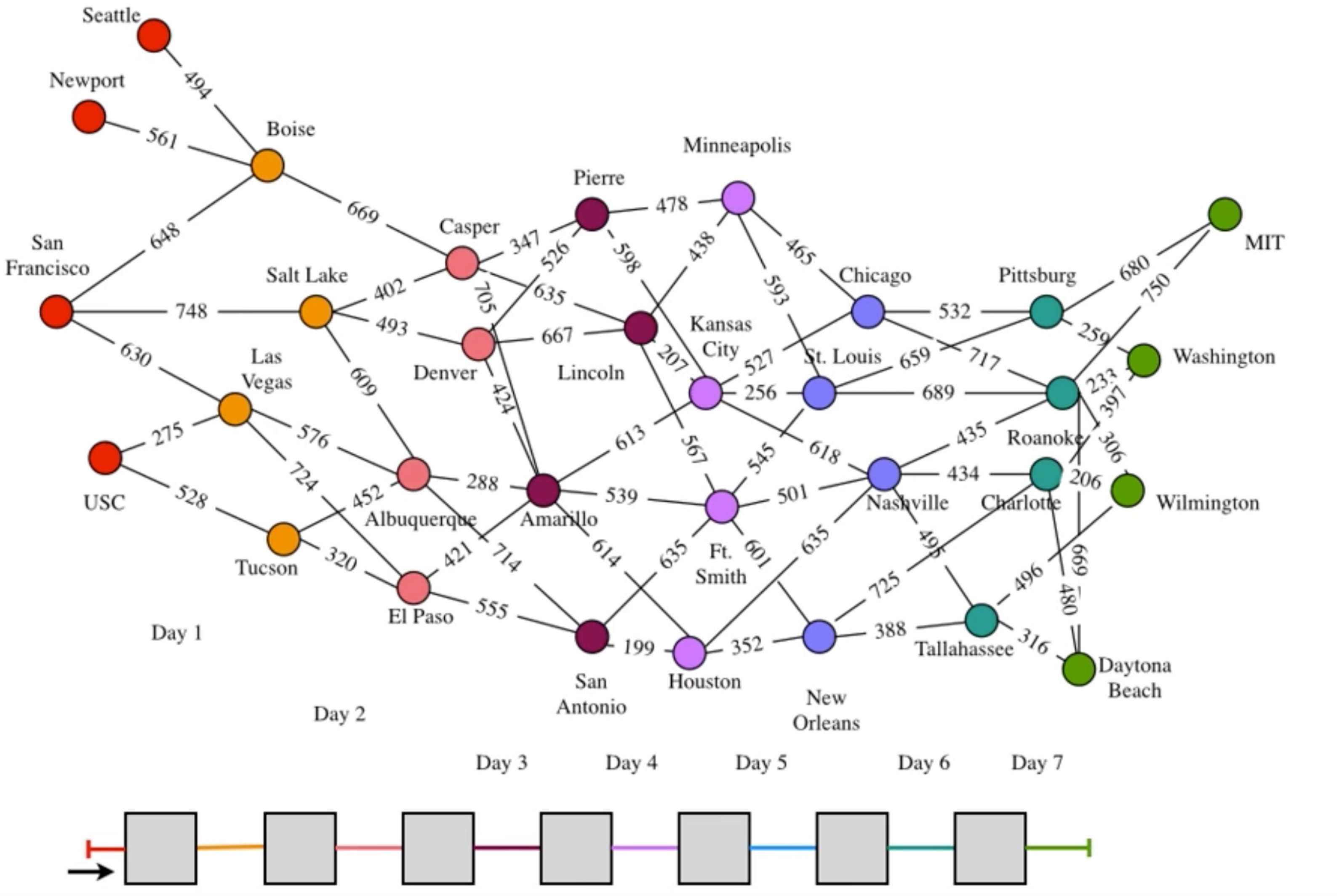
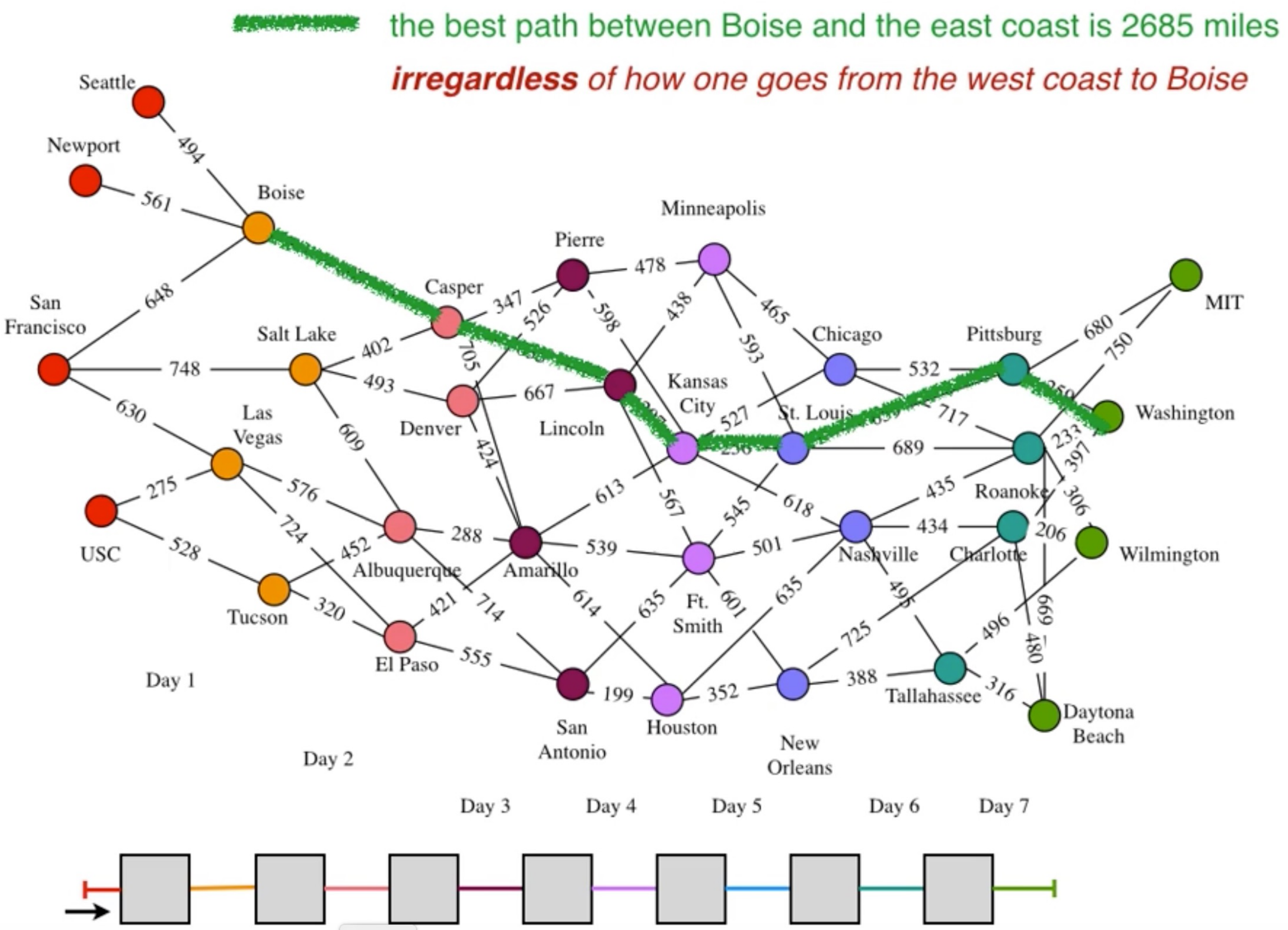


Figure 2: In our Markov model, the hidden variables *Yt* are the 26 letters of the English alphabet and the observed variables are the bitmap images

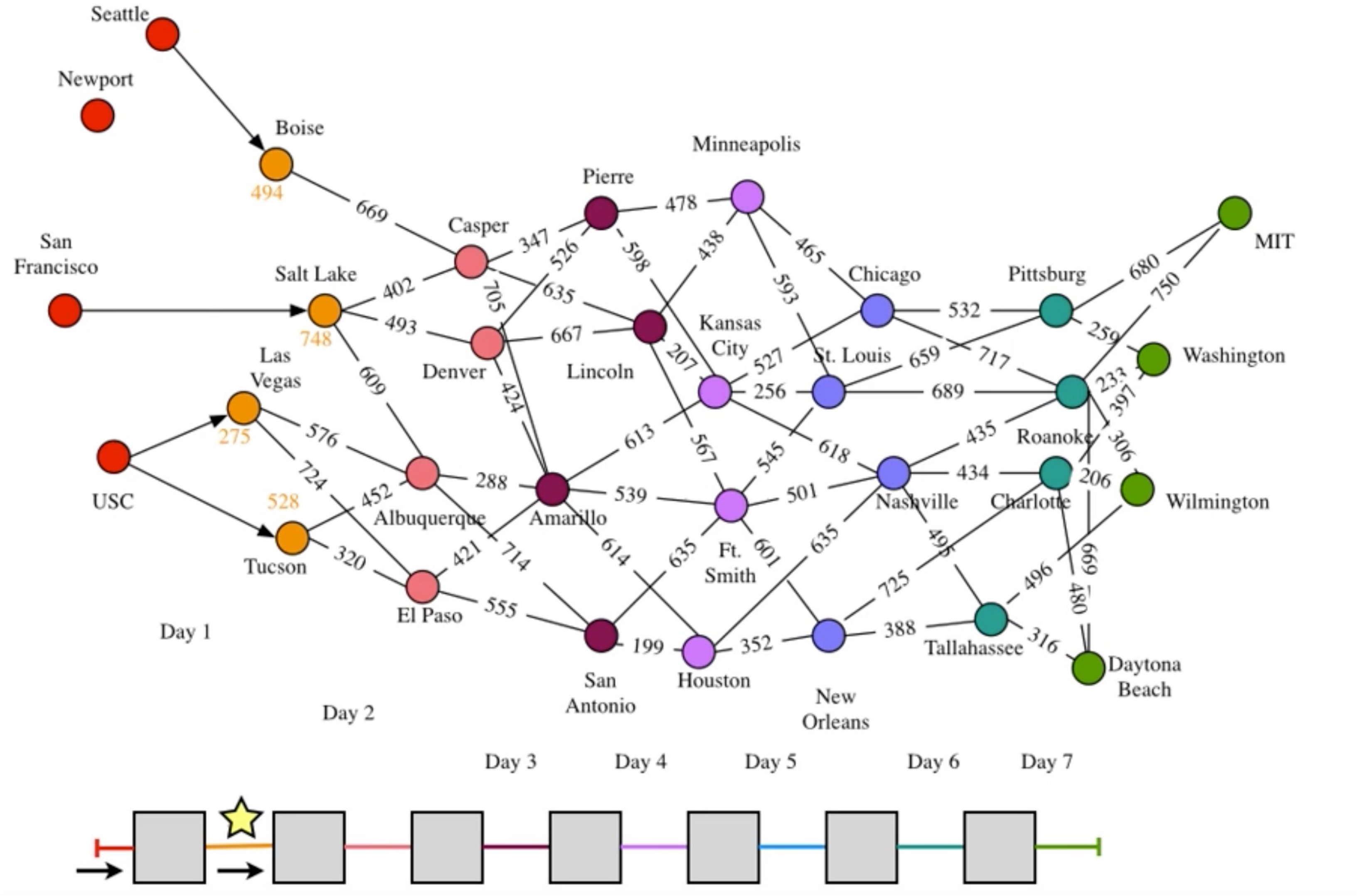
Viterbi Algorithms：Shortest path from Red to Green



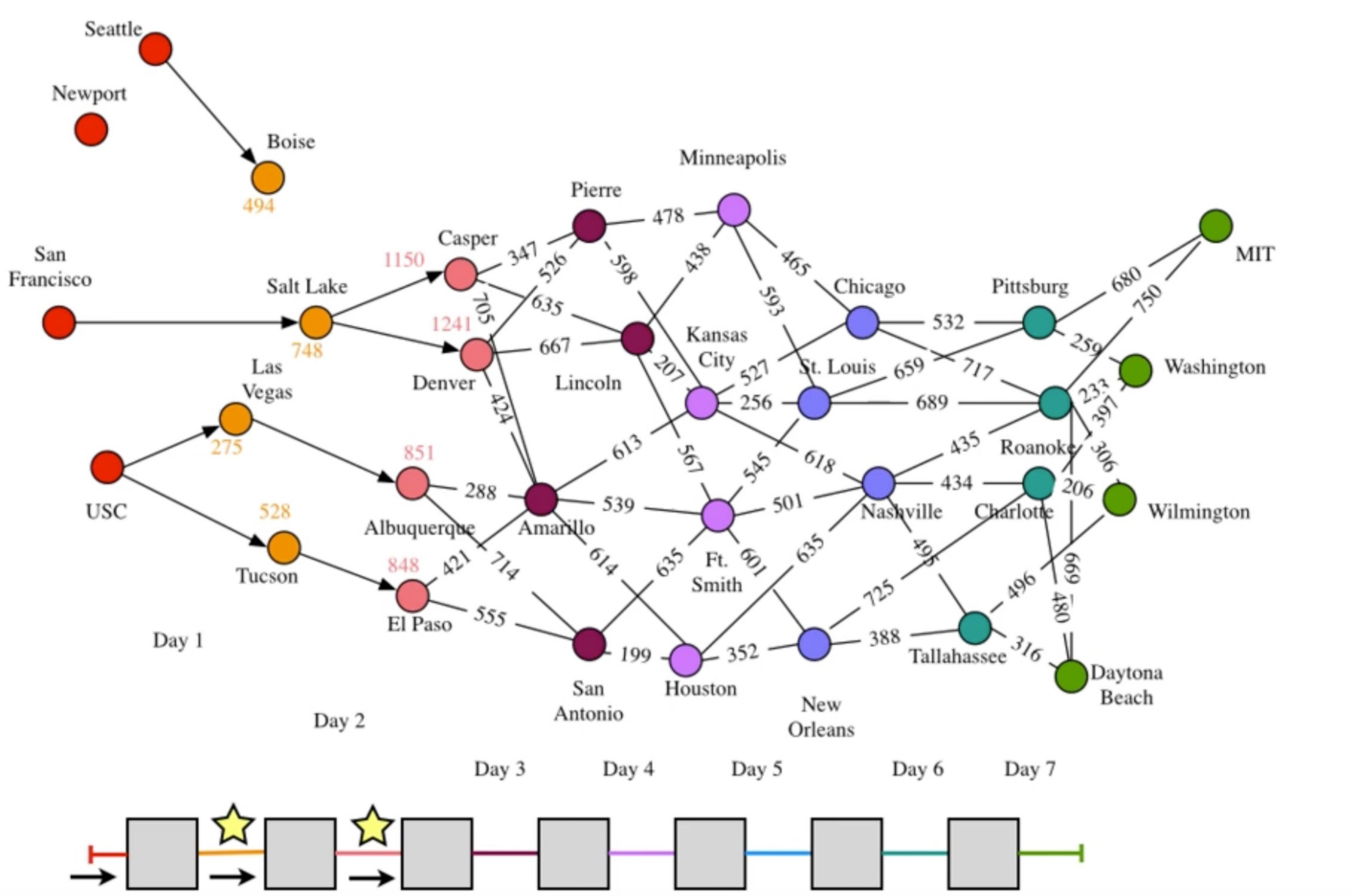
Shortest path to Boise



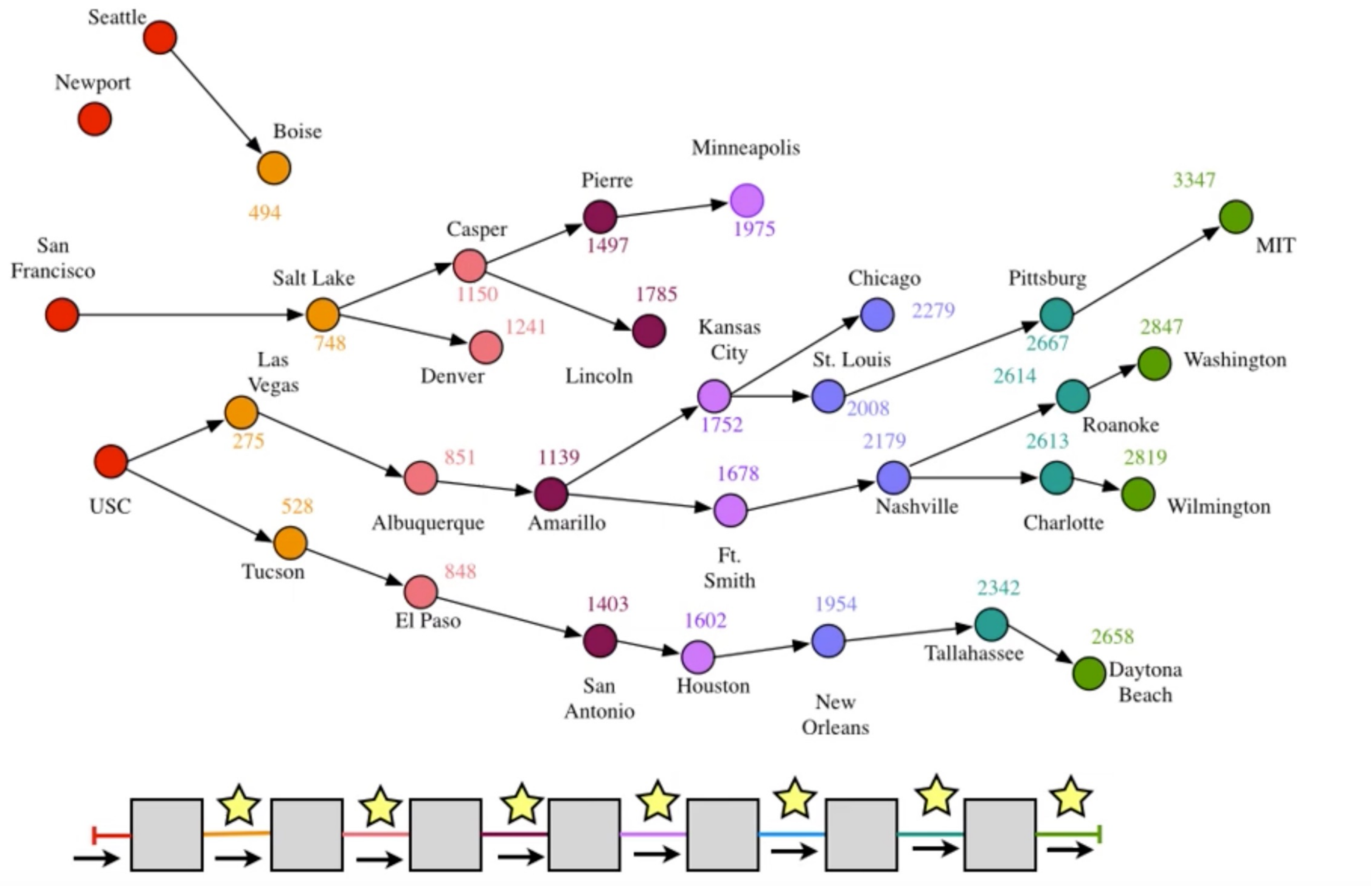
Viterbi Algorithm：Day 1



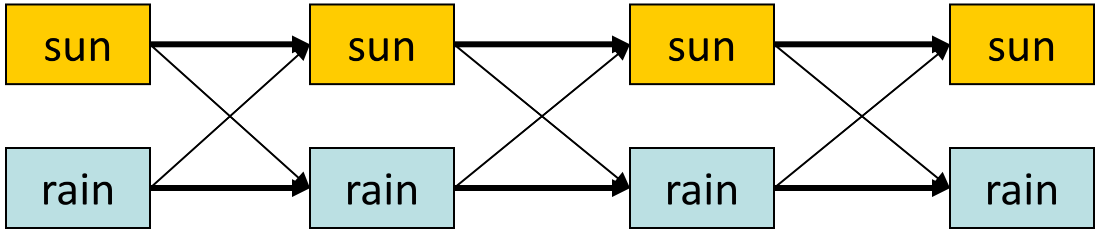
Viterbi Algorithm：Day 2



Viterbi Algorithm ： Day 7



Forward / Viterbi Algorithms



X_1 X_2 $\cdots$ X_N

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

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)

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

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

Viterbi Algorithms

- The algorithm consists of two passes:
- First: runs forward in time and computes the probability of the best path to (state, time) tuple given the evidence observed so far.
- Second pass runs backwards in time: first it finds the terminal state that lies on the path with the highest probability, and then traverses backward through time along the path that leads into this state (which must be the best path).

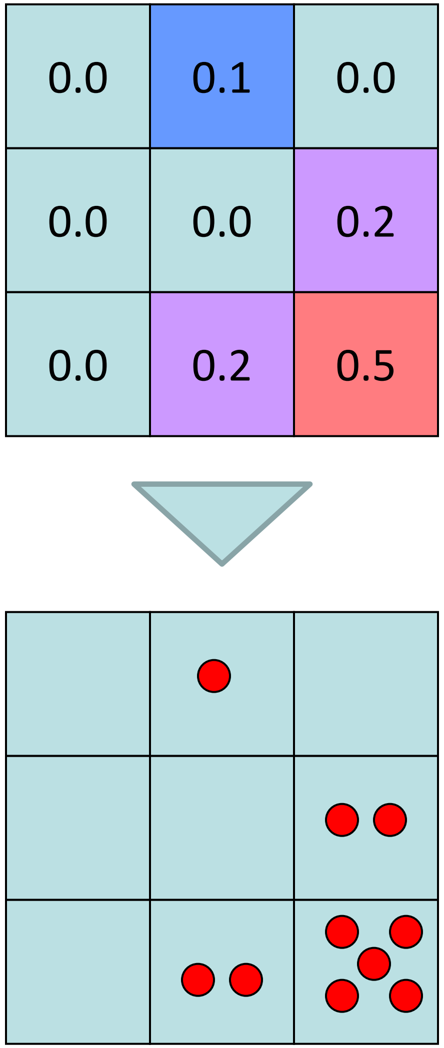
Viterbi Algorithms

```
Result: Most likely sequence of hidden states x*_{1:N}
/* Forward pass                                                    */
for t = 1 to N do
|  for x_t ∈ 𝒳 do
|  |  if t = 1 then
|  |  |  m_t[x_t] = P(x_t)P(e_0|x_t)
|  |  else
|  |  |  a_t[x_t] = argmax_{x_{t-1}} P(x_t|x_{t-1}) m_{t-1}[x_{t-1}];
|  |  |  m_t[x_t] = P(e_t|x_t) P(x_t|a_t[x_t]) m_{t-1}[a_t[x_t]];
|  |  end
|  end
end
/* Find the most likely path's ending point                        */
x*_N = argmax_{x_N} m_N[x_N];
/* Work backwards through our most likely path and find the hidden
   states                                                          */
for t = N to 2 do
|  x*_{t-1} = a_t[x*_t];
end
```

Contents

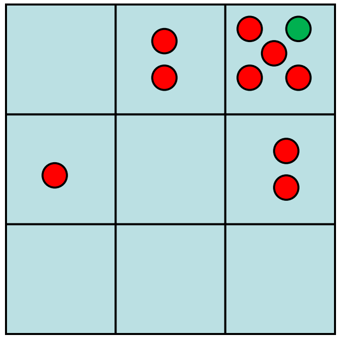
Particle Filtering

- Filtering: approximate solution
- Filtering: compute $P(X_t \mid e_{1:t})$
- 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)

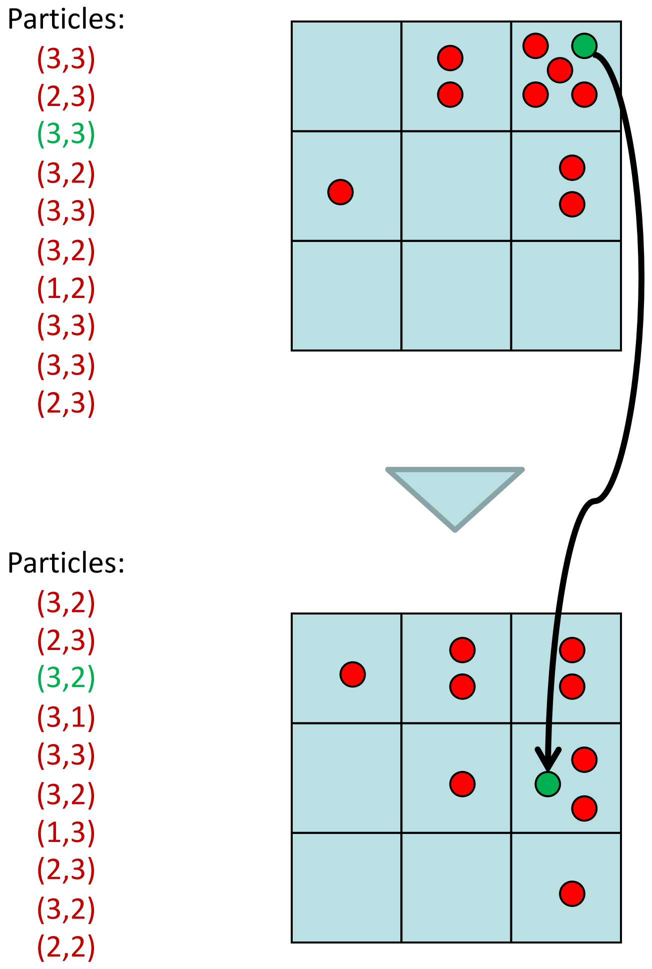
Particle Filtering: Elapse Time

- 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

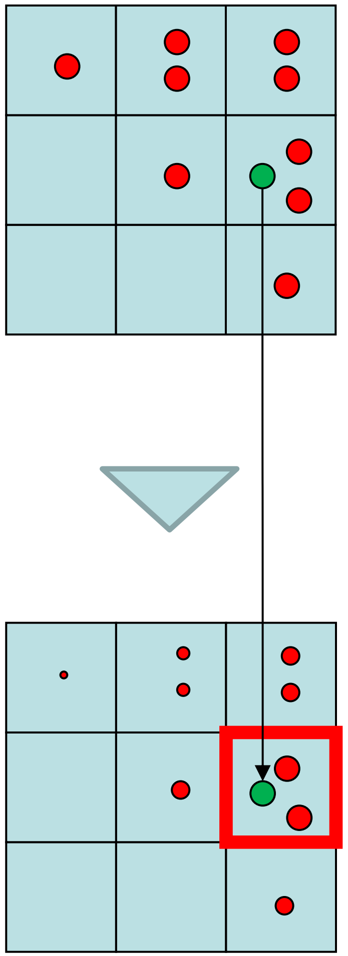
- Slightly trickier:
  - Don't sample observation, fix it
  - Similar to likelihood weighting, downweight samples based on the evidence

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

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

  - 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)



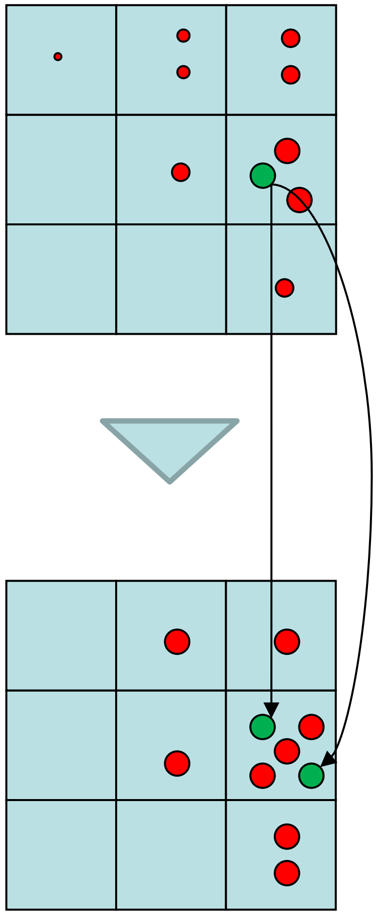
Particles:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4

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

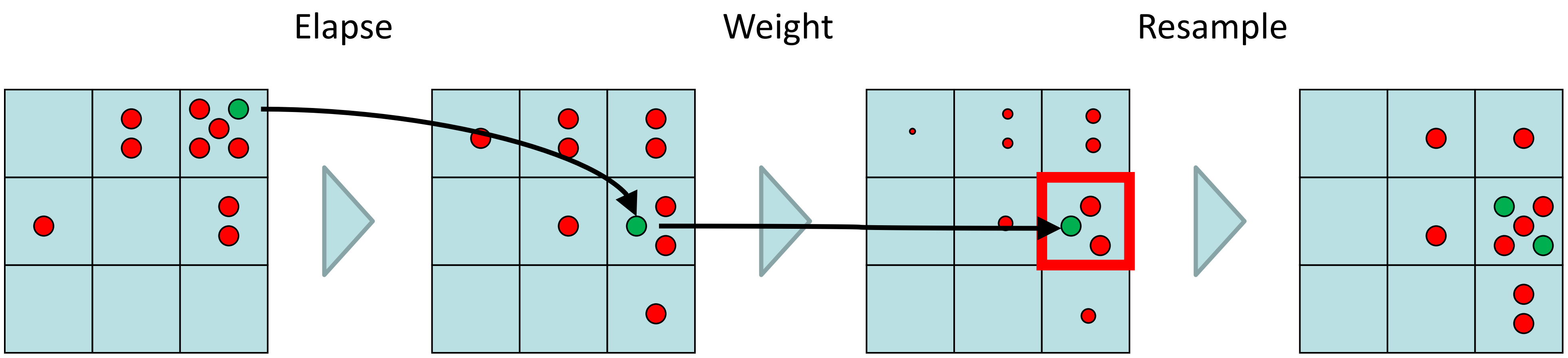
Particles:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4

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



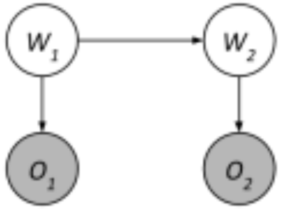
Recap: Particle Filtering

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



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

Let's use Particle Filtering to estimate the distribution of $P(W_2|O_1 = a, O_2 = b)$. Here's the HMM again. O_1 and O_2 are supposed to be shaded.



W_1	$P(W_1)$
0	0.3
1	0.7

W_t	W_{t+1}	$P(W_{t+1}\|W_t)$
0	0	0.4
0	1	0.6
1	0	0.8
1	1	0.2

W_t	O_t	$P(O_t\|W_t)$
0	a	0.9
0	b	0.1
1	a	0.5
1	b	0.5

We start with two particles representing our distribution for W_1.
$P_1 : W_1 = 0$
$P_2 : W_1 = 1$
Use the following random numbers to run particle filtering:

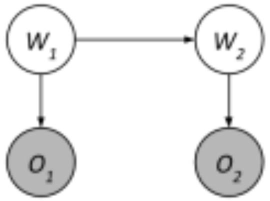
$$[0.22, 0.05, 0.33, 0.20, 0.84, 0.54, 0.79, 0.66, 0.14, 0.96]$$

(a) Observe: Compute the weight of the two particles after evidence $O_1 = a$.

(b) Resample: Using the random numbers, resample P_1 and P_2 based on the weights.

(c) Predict: Sample P_1 and P_2 from applying the time update.

Let's use Particle Filtering to estimate the distribution of $P(W_2|O_1 = a, O_2 = b)$. Here's the HMM again. O_1 and O_2 are supposed to be shaded.



W_1	$P(W_1)$
0	0.3
1	0.7

W_t	W_{t+1}	$P(W_{t+1}\|W_t)$
0	0	0.4
0	1	0.6
1	0	0.8
1	1	0.2

W_t	O_t	$P(O_t\|W_t)$
0	a	0.9
0	b	0.1
1	a	0.5
1	b	0.5

We start with two particles representing our distribution for W_1.
$P_1 : W_1 = 0$
$P_2 : W_1 = 1$
Use the following random numbers to run particle filtering:

$$[0.22, 0.05, 0.33, 0.20, 0.84, 0.54, 0.79, 0.66, 0.14, 0.96]$$

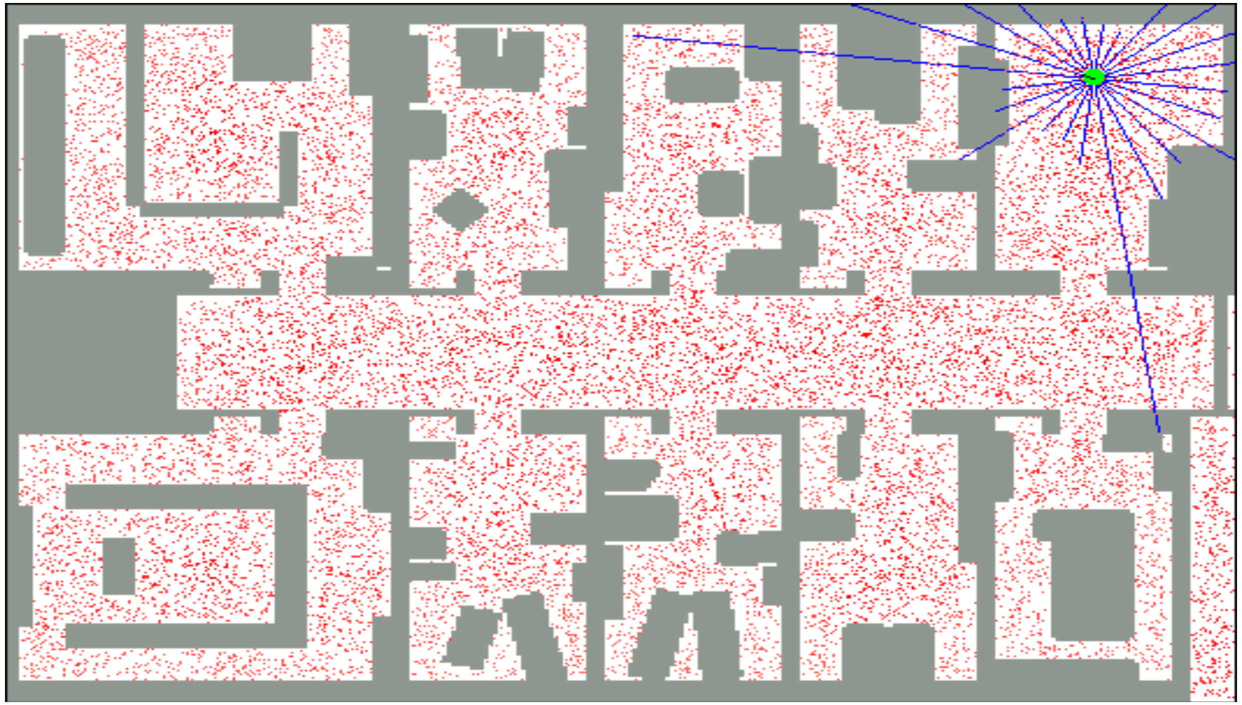
(d) Update: Compute the weight of the two particles after evidence $O_2 = b$.

(e) Resample: Using the random numbers, resample P_1 and P_2 based on the weights.

(f) What is our estimated distribution for $P(W_2|O_1 = a, O_2 = b)$?

Robot Localization

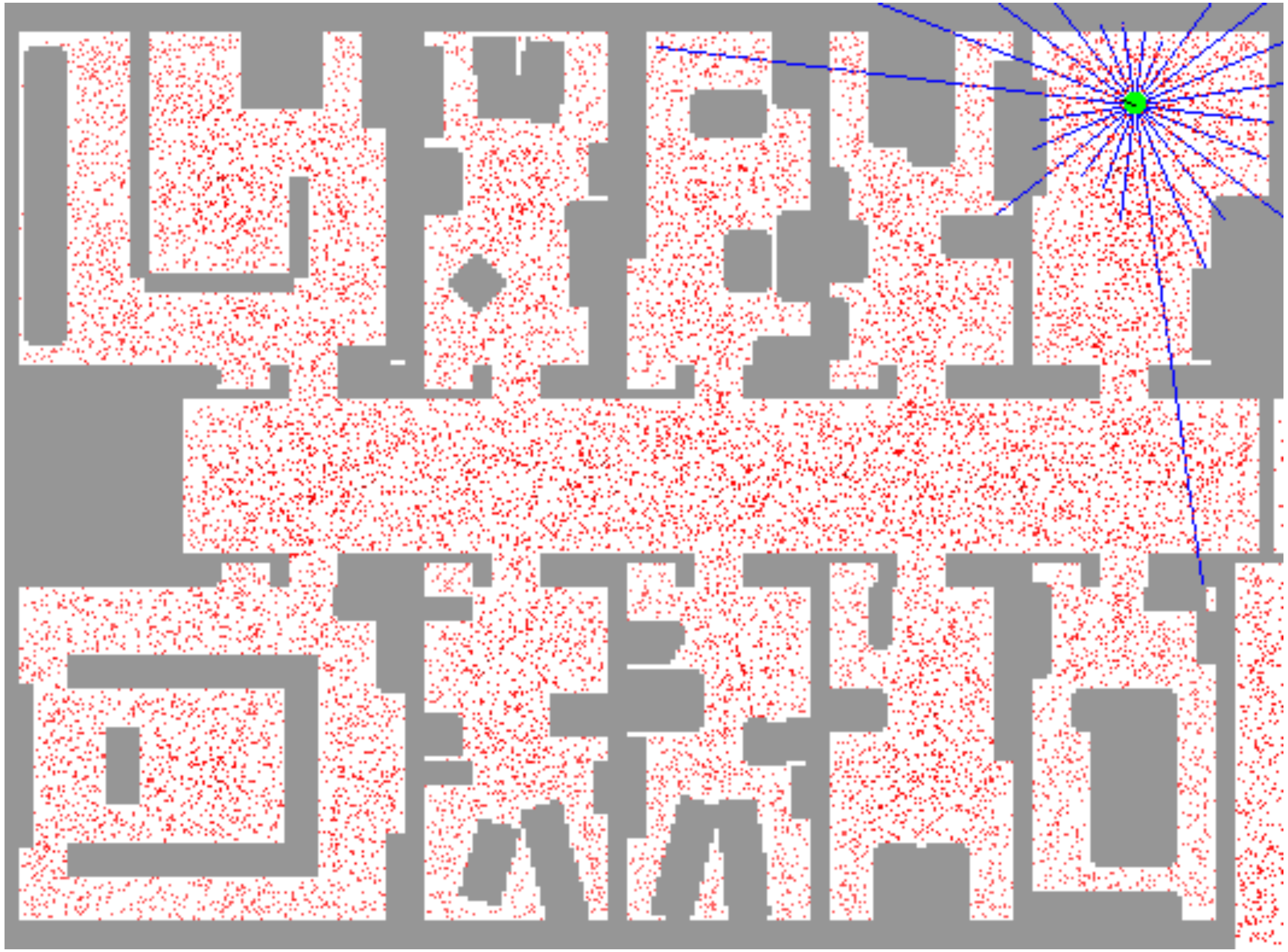
- In robot localization:
  - We know the map, but not the robot's position
  - Observations may be vectors of range finder readings
  - State space and readings are typically continuous (works basically like a very fine grid) and so we cannot store B(X)
  - Particle filtering is a main technique



Particle Filter Localization (Sonar)

[Video: global-sonar-uw-annotated.avi]

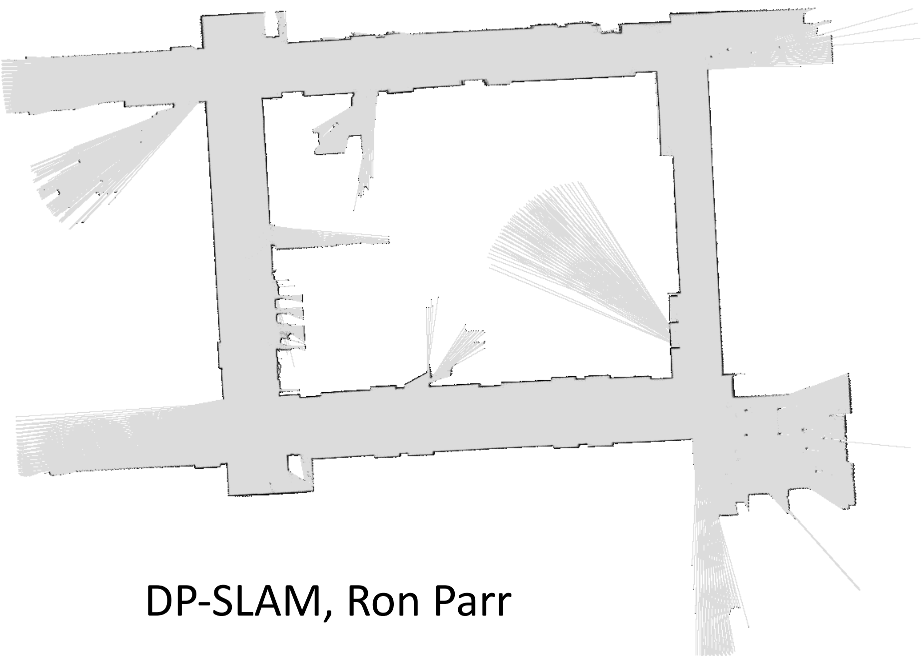
Particle Filter Localization (Laser)



[Video: global-floor.gif]

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

[Demo: PARTICLES-SLAM-mapping1-new.avi]

VisualSLAM