CSC384 Summerr 2018

Week 10 - Probabilistic Reasoning Temporal Models

Ilir Dema

University of Toronto

Jul 30, 2018

UNIVERSITY OF TORONTO MISSISSAUGA

Overview

Markov Models

Independence in a Bayes Net

- Another piece of information we can obtain from a Bayes net is the "structure" of relationships in the domain.
- The structure of the BN means: every X_i is *conditionally independent of all of its nondescendants given it parents*:

$$\Pr(X_i \mid S \cup \text{Par}(X_i)) = \Pr(X_i \mid \text{Par}(X_i))$$

for any subset $S \subseteq \text{NonDescendents}(X_i)$

More generally

- Many conditional independencies hold in a given BN.
- These independencies are useful in computation, explanation, etc.
- Some of these independencies can be detected using a graphical condition called D-Separation.

More generally

- How do we determine if two variables X, Y are independent given a set of variables E?
- Simple graphical property: D-separation
  - A set of variables E d-separates X and Y if it blocks every undirected path in the BN between X and Y. (We'll define *blocks* next.)
  - X and Y are conditionally independent given evidence E if E d-separates X and Y
  - thus BN gives us an easy way to tell if two variables are independent ($E = \emptyset$) or cond. independent given E.

Blocking in D-Separation

- We say E blocks path P iff there is some node Z on the path P such that:

  Case 1: $Z \in E$ and one arc on P enters (goes into) Z and one leaves (goes out of Z); or
  Case 2: $Z \in E$ and both arcs on P leave Z; or
  Case 3: both arcs on P enter Z and neither Z, nor any of its descendants, are in E.

Blocking: Graphical View

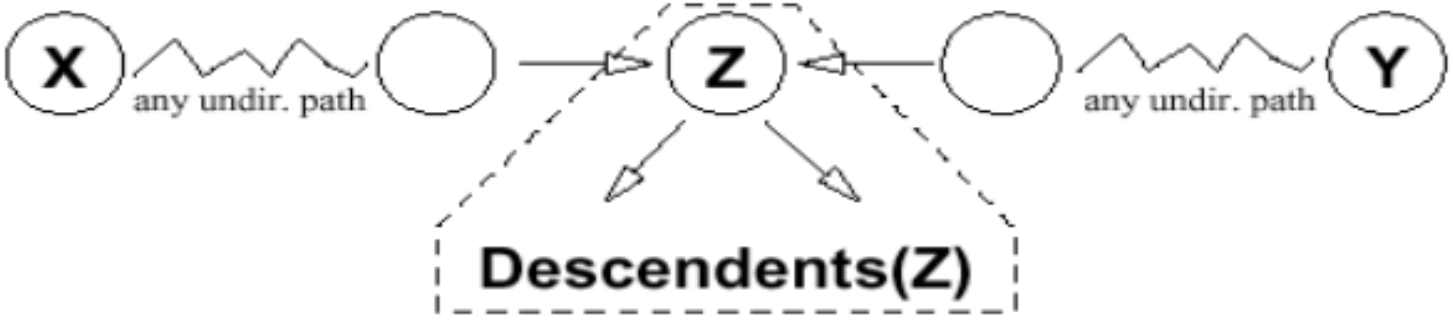
(1)

If Z in evidence, the path between X and Y blocked

(2)

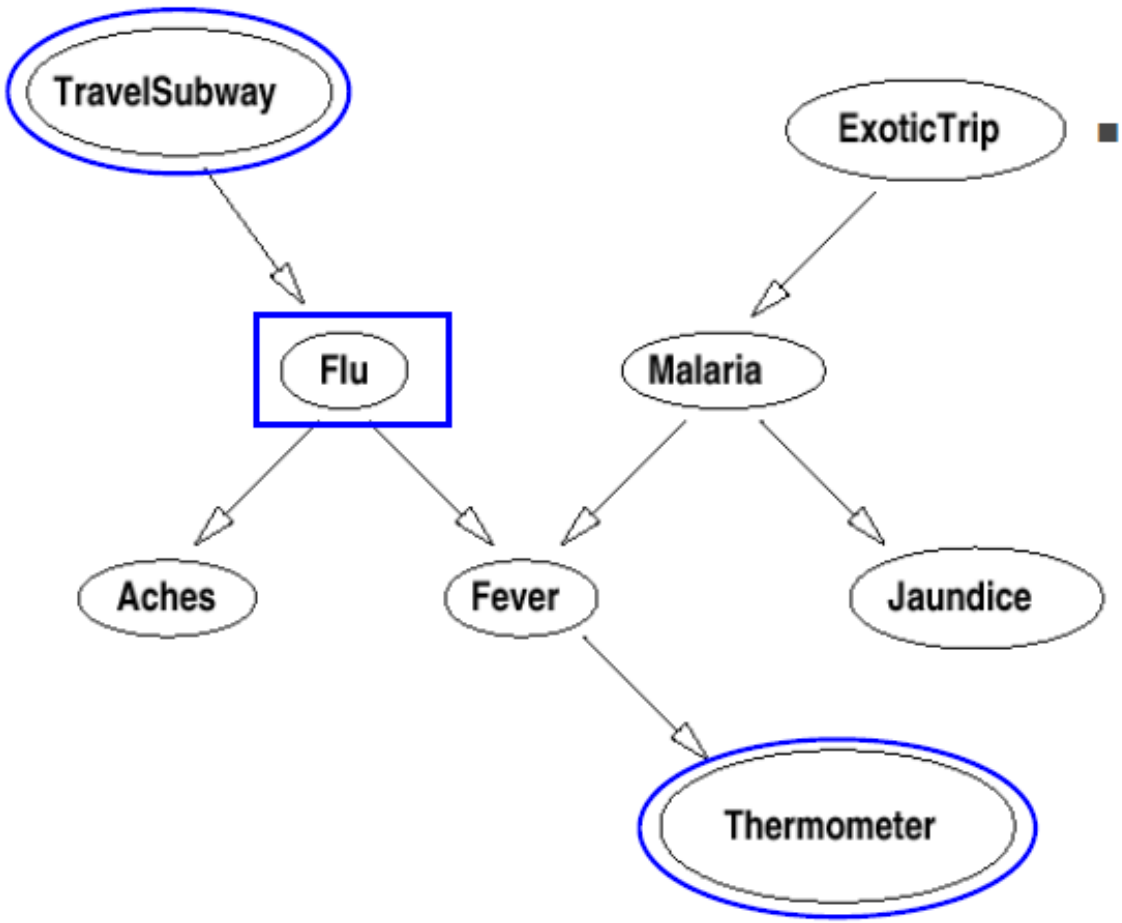
If Z in evidence, the path between X and Y blocked

(3)



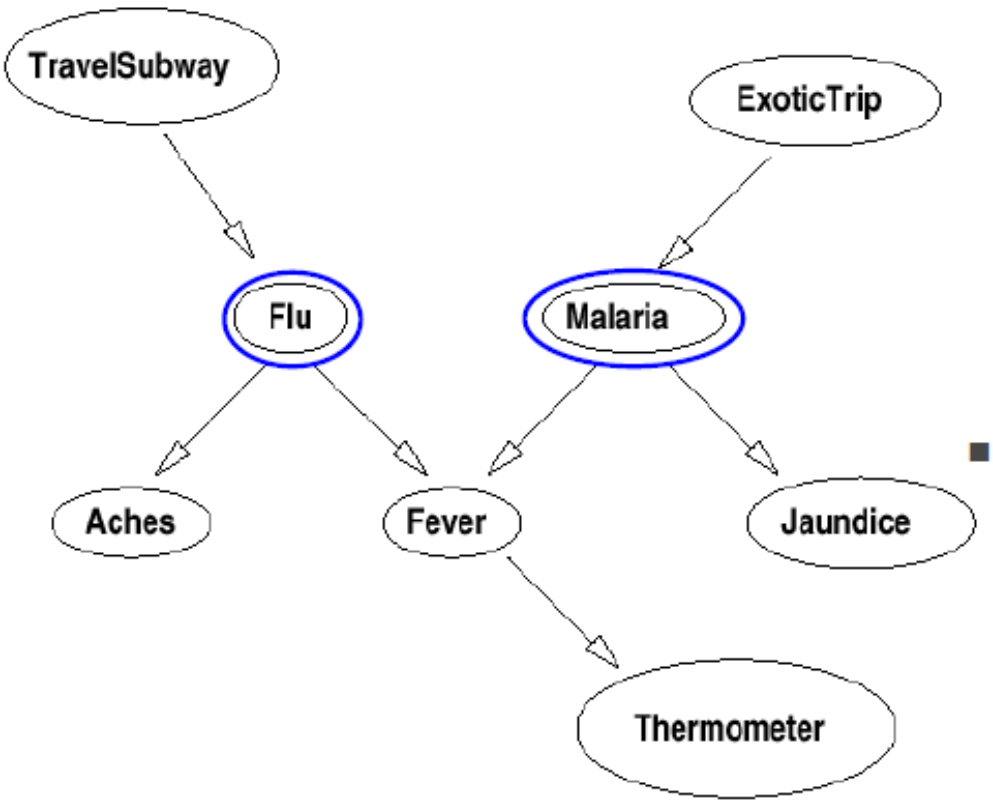
If Z is ***not*** in evidence and***no*** descendent of Z is in evidence, then the path between X and Y is blocked

EXAMPLE



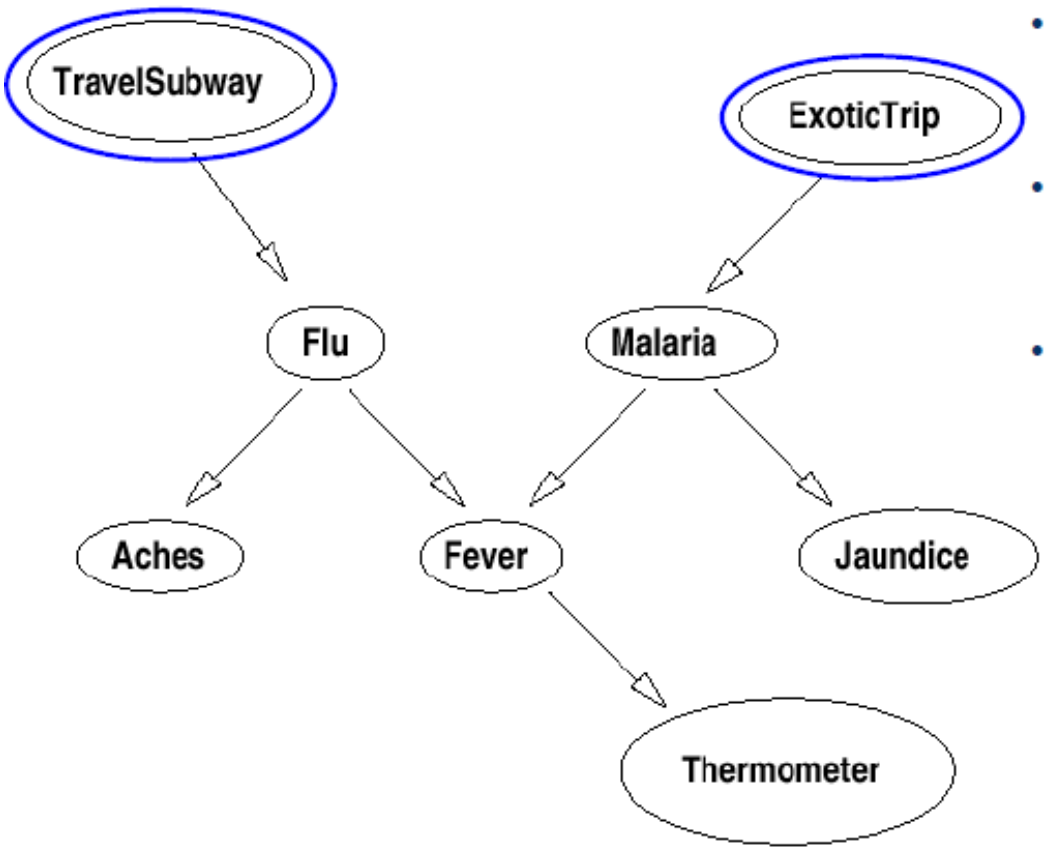
- Subway and Thermometer are **dependent**; but are **independent given Flu** (since Flu blocks the only path)

EXAMPLE



- Flu and Mal are **independent (given no evidence):** Fever blocks the path, since it is *not in evidence*, nor is its decsendant Therm.
- Flu and Mal are **dependent** given Fever (or given Therm): nothing blocks path now. **What's the intuition?**

EXAMPLE

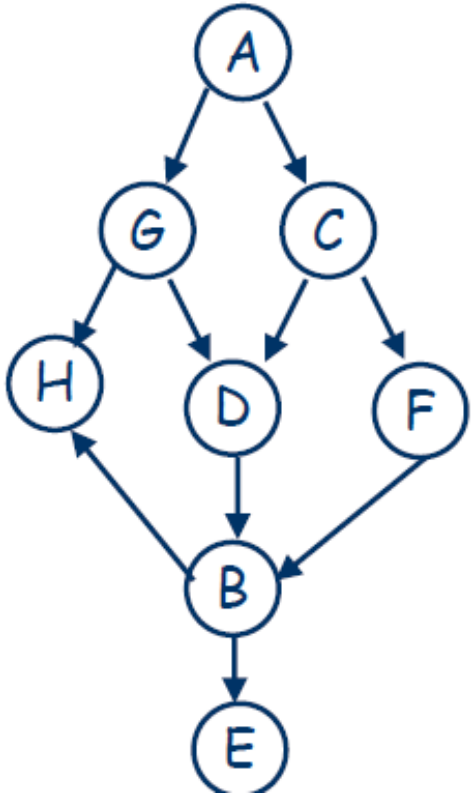


- Subway, ExoticTrip are **independent**;
- They are **dependent given Therm**;
- They are **independent given Therm and Malaria**. This for exactly the same reasons for Flu/Mal above.

EXERCISE

- In the following network determine if A and E are independent given the evidence:

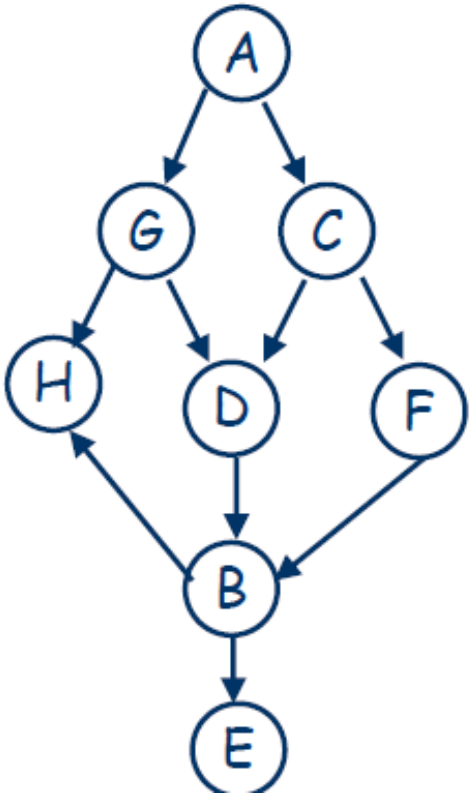
1. A and E given no evidence?
2. A and E given {C}?
3. A and E given {G,C}?
4. A and E given {G,C,H}?
5. A and E given {G,F}?
6. A and E given {F,D}?
7. A and E given {F,D,H}?
8. A and E given {B}?
9. A and E given {H,B}?
10. A and E given {G,C,D,H,D,F,B}?



EXERCISE

- In the following network determine if A and E are independent given the evidence:

1. A and E given no evidence? No
2. A and E given {C}? No
3. A and E given {G,C}? Yes
4. A and E given {G,C,H}? Yes
5. A and E given {G,F}? No
6. A and E given {F,D}? Yes
7. A and E given {F,D,H}? No
8. A and E given {B}? Yes
9. A and E given {H,B}? Yes
10. A and E given {G,C,D,H,D,F,B}? Yes



Uncertainty

- In many practical problems we want to reason about a sequence of observations
  - Speech recognition
  - Robot localization
  - User attention
  - Medical monitoring
- Need to introduce time (or space) into our models

Markov Models

- Have one feature X (perhaps with a very large number of possible states). We want to track the probability of different values of X (the probability distribution over X) as it changes over time
- We make multiple copies of X, one for each time point i (we use a discrete model of time): $X_1, \ldots, X_n$.
- A Markov Model is specified by the two following assumptions
  1. The current state X_t is conditionally independent of the earlier states given the previous state.

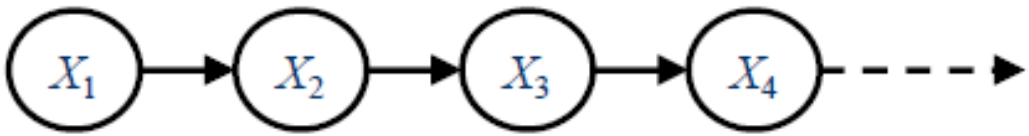
$$Pr(X_t|X_1, \ldots, X_{t-1}) = Pr(X_t|X_{t-1})$$

  2. The transitions between X_{t-1} and X_t are determined by probabilities that do not change over time (stationary probabilities).
     $Pr(X_t|X_{t-1})$ is the same for all points in time t.

MARKOV MODELS

Markov Models

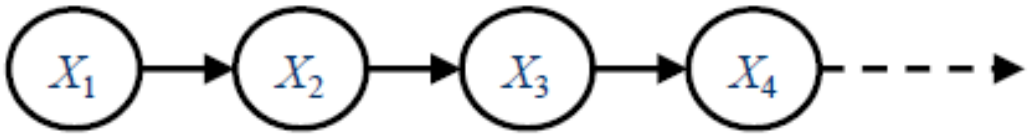
- These assumptions give rise to the following Bayesian Network



- $Pr(X_1,X_2,X_3,....) = Pr(X_1)Pr(X_2|X_1)Pr(X_3|X_2)$ … (Assumption 1)
- And all the CPTs (except $Pr(X_1)$) are the same (Assumption 2)

MARKOV MODELS

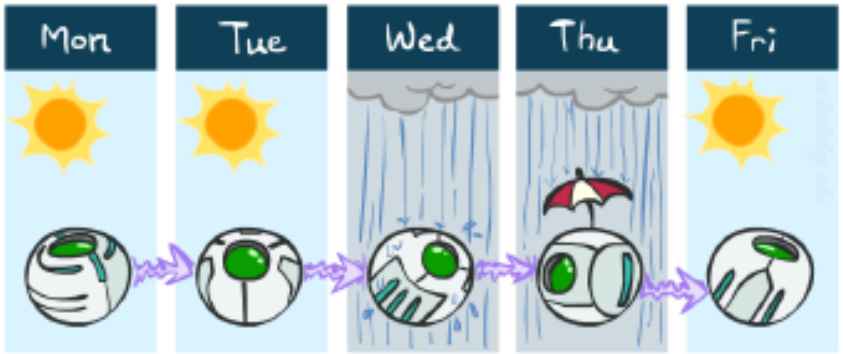
Markov Models



- D-Separation also shows us that X_{t-1} is conditionally independent of X_{t+1}, X_{t+2}, … given X_t
  - The current state separates the past from the future.

Markov Models

Example Markov Chain Weather



- States: X = {rain, sun}

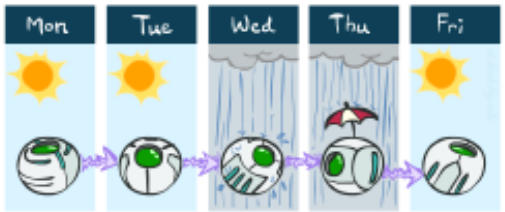
- Initial distribution:
  $Pr(X_1=sun) = 1.0$

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

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

Markov Models

Example Markov Chain Weather



- $\Pr(X_1=\text{sun}) = 1.0$
- What is the probability distribution after one step, $\Pr(X_2)$?
- Use summing out rule with X_1

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

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

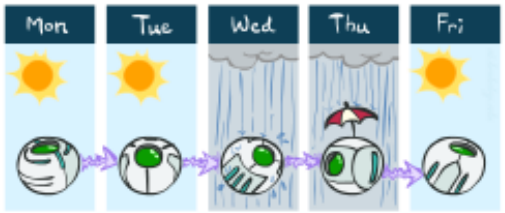
CPT Pr(Xt | Xt-1):

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

[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.

Markov Models

Example Markov Chain Weather



- What is the probability distribution on the day t, $Pr(X_t)$?
- Sum out over X_{t-1}

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

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

Forward simulation
Compute $Pr(X_2)$ then $Pr(X_3)$ then $Pr(X_4)$ …

CPT Pr(Xt | Xt-1):

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

[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.

Markov Models

Stationary Distributions

- For most Markov 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
  - It satisfies

$$\Pr_\infty(X = d') = \sum_{d \in Dom[X]} \Pr(X = d' | X = d) \Pr_\infty(X = d')$$

  - That is the stationary distribution does not change on an forward progression
  - We can compute it by solving these simultaneous equations (or by forward simulating the system many times; forward simulation is generally computationally more effective)

EXAMPLE

Web Link Analysis

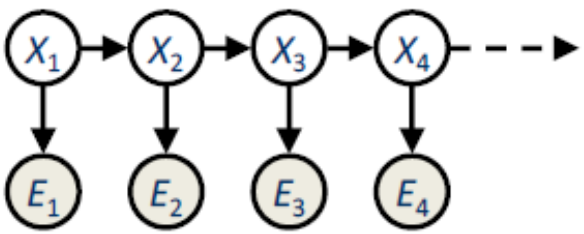
- PageRank over a web graph
  - Each web page is a state
  - Initial distribution: uniform over pages
  - Transitions:
    - With prob. c, uniform jump to a random page
    - With prob. 1-c, follow a random outlink
- Stationary distribution
  - Will spend more time on highly reachable pages
  - E.g. 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)

[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.

HIDDEN MARKOV MODELS

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
  - But you also observe outputs (effects) at each time step

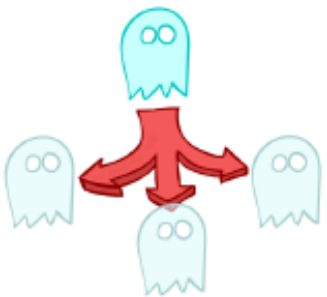


[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.

EXAMPLE

Example: Ghostbusters HMM

- $Pr(X_1)$ = uniform
- $Pr(X|X')$ = usually move clockwise, but sometimes move in a random direction or stay in place
- $Pr(D|X)$ = Observe distance to ghost (using a noisy sonar)

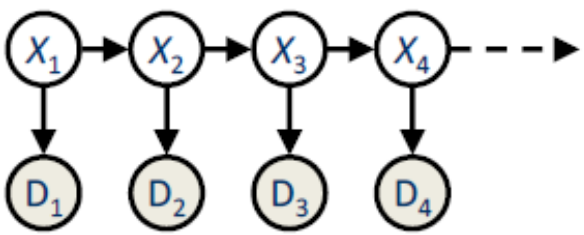


1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

$P(X_1)$

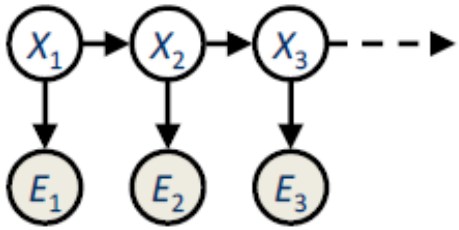
1/6	1/6	1/2
0	1/6	0
0	0	0

$P(X|X'=<1,2>)$



JOINT DISTRIBUTION OF A HMM

Joint Distribution of an HMM

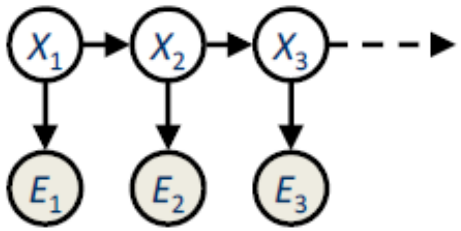


Assumptions:

1. $\Pr(X_t|X_{t-1}, E_{t-1}, \cdots, E_1, X_1) = \Pr(X_t|X_{t-1})$
   The current state Xt is conditionally independent of the earlier states and evidence given the previous state.
2. $\Pr(X_t|X_{t-1})$ is the same for all time points t
   The transitions between X_{t-1} and X_t are determined by probabilities that do not change over time (stationary probabilities).
3. $\Pr(E_t|X_t, E_{t-1}, X_{t-1}, \cdots, E_1, X_1) = \Pr(E_t|X_t)$
   The current evidence is conditionally independent of all earlier states and evidence given the current state.

Joint Distribution of a HMM

Joint Distribution of an HMM



Independencies

As with Markov Chains, the past is independent of the future (and vice versa) given the current state. (Easy to see by D-Separation)

But note that two evidence items are not independent, unless one of the intermediate states is known.

EXAMPLES

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)

[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.

FILTERING/MONITORING

- Filtering, or monitoring, is the task of tracking $\Pr(X_t \mid e_1, \ldots, e_t)$ over time: the probability distribution over feature X updated by all evidence accumulated so far.
- $P(X_1)$ is the initial distribution over feature X. (Usually we start with a uniform distribution, $\Pr(X_1 = d)$ is the same for all d)
- As time passes, and we get observations, we update our distribution over X, i.e., we move from $\Pr(X_{t-1}|e_{t-1}, e_{t-2}, \cdots, e_1)$ to $\Pr(X_t|e_t, e_{t-1}, \cdots, e_1)$

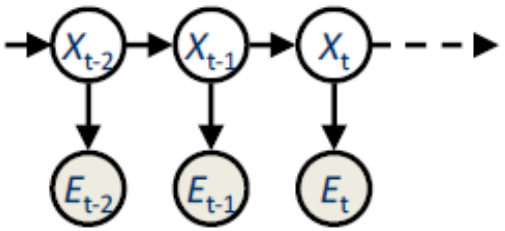
Update rules for HMM

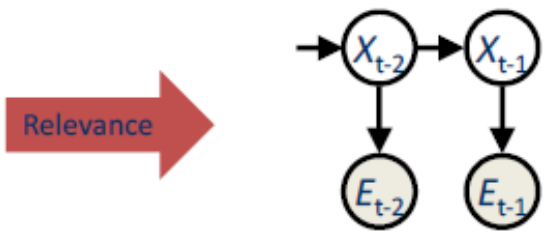
- HMMs are a type of Bayes Net so we can use Variable Elimination to compute $\Pr(X_t|e_t, e_{t-1}, \cdots, e_1)$ at all time points t.
- If we compute $\Pr(X_{t-1}|e_{t-1}, \cdots, e_1)$ using VE we will see that most of the information we need to compute the X_t has already been done.
- Reusing this information allows us to derive simple update rules.

VARIABLE ELIMINATION

VE for $\Pr(X_{t-1}|e_{t-1},\cdots,e_1)$

Relevance reasoning shows that all future variables $X_t, E_t, X_{t+1}, E_{t+1}, \cdots$ are irrelevant. X_{t-1} is the query, and e_{t-1} is evidence: Only ancestors of X_{t-1}



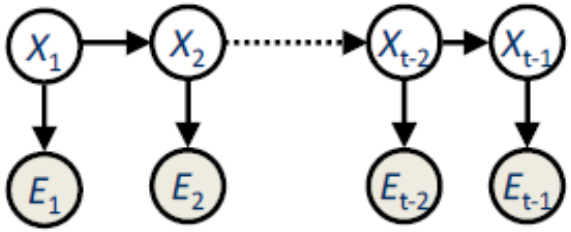


VARIABLE ELIMINATION

VE for $\Pr(X_{t-1}|e_{t-1},\cdots,e_1)$

Order of Elimination X_1, …, X_{t-1} (all E_i variables have been instantiated with values e_i)

X_1: $\Pr(X_1)\Pr(e_1|X_1)\Pr(X_2|X_1)$
X_2: $\Pr(e_2|X_2)\Pr(X_3|X_2)$
…
X_{t-2}: $\Pr(e_{t-2}|X_{t-2})\Pr(X_{t-1}|X_{t-1})$
X_{t-1}: $\Pr(e_{t-1}|X_{t-1})$



VARIABLE ELIMINATION

VE for $\Pr(X_{t-1}|e_{t-1},\cdots,e_1)$

Summing out X_1 we get a factor over X_2, summing out over X_2 we get a factor over X_3 … summing out X_{t-2} we get a factor over X_{t-1}

X_1: $\Pr(X_1)\Pr(e_1|X_1)\Pr(X_2|X_1)$
X_2: $\Pr(e_2|X_2)\Pr(X_3|X_2)\,F_2(X_2)$
…
X_{t-2}: $\Pr(e_{t-2}|X_{t-2})\Pr(X_{t-1}|X_{t-1})\,F_{t-2}(X_{t-2})$
X_{t-1}: $\Pr(e_{t-1}|X_{t-1})\,F_{t-1}(X_{t-1})$

VARIABLE ELIMINATION

VE for $\Pr(X_{t-1}|e_{t-1},\cdots,e_1)$

X_1: $\Pr(X_1)\Pr(e_1|X_1)\Pr(X_2|X_1)$
X_2: $\Pr(e_2|X_2)\Pr(X_3|X_2)\,F_2(X_2)$
...
X_{t-2}: $\Pr(e_{t-2}|X_{t-2})\Pr(X_{t-1}|X_{t-1})\,F_{t-2}(X_{t-2})$
X_{t-1}: $\Pr(e_{t-1}|X_{t-1})\,F_{t-1}(X_{t-1})$

$$\Pr(X_{t-1}|e_{t-1},\cdots,e_1) = normalize(\Pr(e_{t-1}|X_{t-1})\,F_{t-1}(X_{t-1}))$$

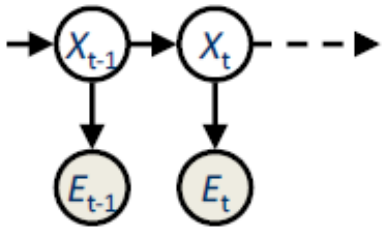
At time step t we have already computed this vector of values for time step t-1. (One number for each value of X_{t-1})

Base case is $\Pr(X_1|e_1) = normalize(\Pr(e_1|X_1)\Pr(X_1))$

VARIABLE ELIMINATION

VE for $\Pr(X_t|e_{t-1},\cdots,e_1)$

Update Rule #1: Time has passed but new observation not yet made.



X_1: $\Pr(X_1)\Pr(e_1|X_1)\Pr(X_2|X_1)$

X_2: $\Pr(e_2|X_2)\Pr(X_3|X_2)$

...

X_{t-2}: $\Pr(e_{t-2}|X_{t-2})\Pr(X_{t-1}|X_{t-1})$

X_{t-1}: $\Pr(e_{t-1}|X_{t-1})\Pr(X_t|X_{t-1})$

X_t:

VE buckets are the same as before except

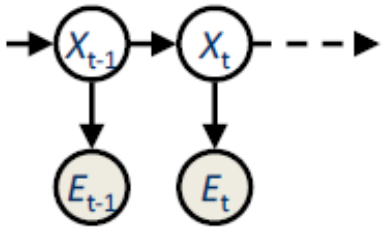
a) Add a bucket for X_t (this is initially empty)

b) Extra factor $\Pr(X_t|X_{t-1})$ in bucket for X_{t-1}

VARIABLE ELIMINATION

VE for $\Pr(X_t|e_{t-1},\cdots,e_1)$

Sum out variables as before obtaining exactly the same factors



X_1: $\Pr(X_1)\Pr(e_1|X_1)\Pr(X_2|X_1)$
X_2: $\Pr(e_2|X_2)\Pr(X_3|X_2)\,F_2(X_2)$
...
X_{t-2}: $\Pr(e_{t-2}|X_{t-2})\Pr(X_{t-1}|X_{t-1})\,F_{t-2}(X_{t-1})$
X_{t-1}: $\Pr(e_{t-1}|X_{t-1})\,F_{t-1}(X_{t-1})\Pr(X_t|X_{t-1})$
X_t: $F_t(X_t)$

We obtain a factor over X_t

$$F_t(X_t) = \sum_{d\in Dom[X_{t-1}]} \Pr(e_{t-1}|X_{t-1})\,F_{t-1}(X_{t-1})\Pr(X_t|X_{t-1})$$

VARIABLE ELIMINATION

VE for $\Pr(X_t | e_{t-1}, \cdots, e_1)$

$$F_t(X_t) = \sum_{d \in Dom[X_{t-1}]} \Pr(e_{t-1} | X_{t-1}) \, F_{t-1}(X_{t-1}) \Pr(X_t | X_{t-1})$$

$$= \sum_{d \in Dom[X_{t-1}]} \alpha \Pr(X_{t-1} | e_{t-1}, \cdots, e_1) \Pr(X_t | X_{t-1})$$

Since we already computed that

$$\Pr(X_{t-1} | e_{t-1}, \cdots, e_1) = \text{normalize}(\Pr(e_{t-1} | X_{t-1}) \, F_{t-1}(X_{t-1}))$$
$$= \Pr(e_{t-1} | X_{t-1}) \, F_{t-1}(X_{t-1})) \, / \alpha$$

(α was the normalization constant)

VARIABLE ELIMINATION

Update rule for $\Pr(X_t | e_{t-1}, \cdots, e_1)$

Finally:

$$\begin{aligned}
&\Pr(X_t | e_{t-1}, \cdots, e_1) \\
&\quad = \text{normalize}(\, F_t(X_t)\,) \\
&\quad = \text{normalize}(\textstyle\sum_{d \in Dom[X_{t-1}]} \alpha \Pr(X_{t-1} | e_{t-1}, \cdots, e_1) \Pr(X_t | X_{t-1})) \\
&\quad = \text{normalize}(\textstyle\sum_{d \in Dom[X_{t-1}]} \Pr(X_{t-1} | e_{t-1}, \cdots, e_1) \Pr(X_t | X_{t-1}))
\end{aligned}$$

We can drop α because we are normalizing.

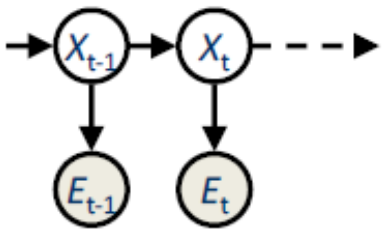
PROGRESS IN TIME

$$\begin{aligned}
&\Pr(X_t | e_{t-1}, \cdots, e_1) = \\
&\quad \text{normalize}(\textstyle\sum_{d \in Dom[X_{t-1}]} \Pr(X_{t-1} | e_{t-1}, \cdots, e_1) \Pr(X_t | X_{t-1}))
\end{aligned}$$

VARIABLE ELIMINATION

VE for $\Pr(X_t|e_t,\cdots,e_1)$

Now we deal with new evidence e_t
Variable Elimination will look similar except



X_1: $\Pr(X_1)\Pr(e_1|X_1)\Pr(X_2|X_1)$
X_2: $\Pr(e_2|X_2)\Pr(X_3|X_2)F_2(X_2)$
...
X_{t-2}: $\Pr(e_{t-2}|X_{t-2})\Pr(X_{t-1}|X_{t-1})F_{t-2}(X_{t-1})$
X_{t-1}: $\Pr(e_{t-1}|X_{t-1})F_{t-1}(X_{t-1})\Pr(X_t|X_{t-1})$
X_t: $F_t(X_t)\Pr(e_t|X_t)$

But we add a new factor $\Pr(e_t|X_t)$ to the X_t bucket then normalize

VARIABLE ELIMINATION

VE for $\Pr(X_t|e_t,\cdots,e_1)$

Hence $\Pr(X_t|e_{t-1},\cdots,e_1) = normalize(F_t(X_t)\Pr(e_t|X_t))$

We already saw that

$$\text{normalize}(F_t(X_t)) = \Pr(X_t|e_{t-1},\cdots,e_1)$$

So

$$F_t(X_t) = \alpha\Pr(X_t|e_{t-1},\cdots,e_1)$$

And

$$\Pr(X_t|e_{t-1},\cdots,e_1) = normalize(\alpha\Pr(X_t|e_{t-1},\cdots,e_1)\Pr(e_t|X_t))$$

We can remove α as we are normalizing

VARIABLE ELIMINATION

Update rule for $\Pr(X_t|e_t, \cdots, e_1)$

Finally:

Observe

$$\Pr(X_t|e_t, \cdots, e_1) = \text{normalize}(\Pr(X_t|e_{t-1}, \cdots, e_1)\Pr(e_t|X_t))$$

VARIABLE ELIMINATION

HMM update rules, recap.

Initial

$\Pr(X_1) = initial\ distribution$ (usually uniform)

Observe

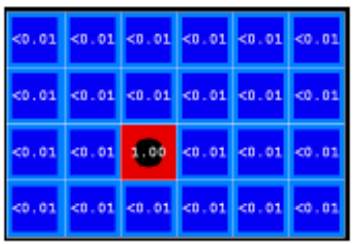
$$\Pr(X_t|e_t, \cdots, e_1) = \text{normalize}(\Pr(X_t|e_{t-1}, \cdots, e_1)\Pr(e_t|X_t))$$

PROGRESS IN TIME

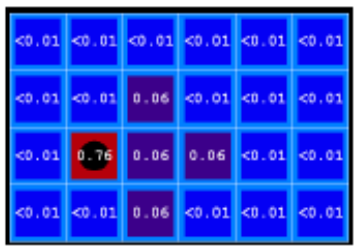
$$\Pr(X_t|e_{t-1}, \cdots, e_1) = \text{normalize}(\sum_{d \in Dom[X_{t-1}]} \Pr(X_{t-1}|e_{t-1}, \cdots, e_1)\Pr(X_t|X_{t-1}))$$

EXAMPLE: PASSAGE OF TIME

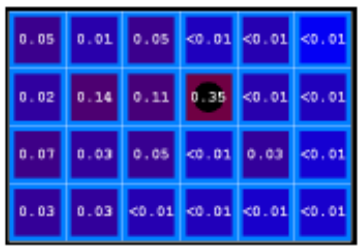
- As time passes, uncertainty "accumulates"
  - (Transition model: ghosts usually go clockwise)



T = 1



T = 2

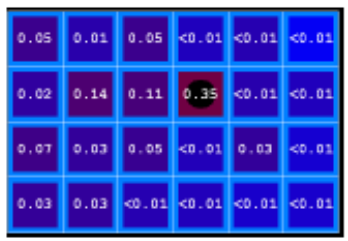


T = 5

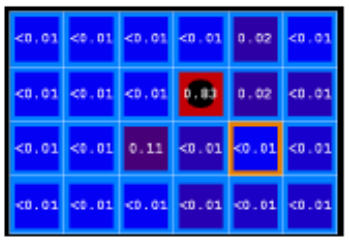
[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.

EXAMPLE: OBSERVATION

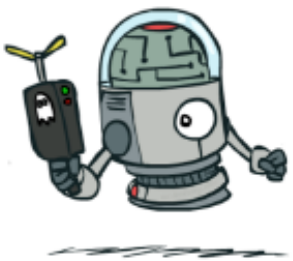
- As we get observations, beliefs get reweighted, uncertainty "decreases"



Before observation



After observation



$$\Pr(X_t|e_t, \dots, e_1) \propto \Pr(e_t|X_t) \Pr(X_{t-1}|e_{t-1}, \dots, e_1)$$

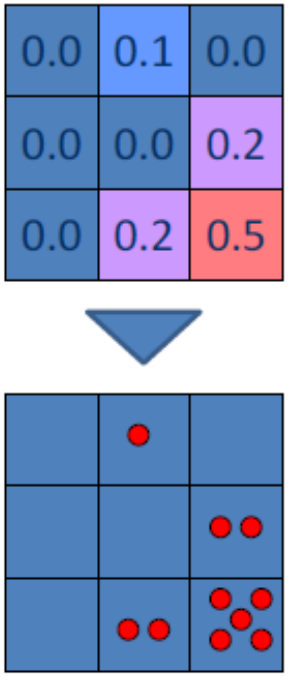
Approximate Inference

- Often the Bayes net is not solvable by Variable Elimination: under any ordering of the variables we end up with a factor that is too large to compute (or store).
- Since we are trying to compute a probability (which only predicts the likelihood of an event occurring) it is natural to consider approximating answer.
- Approximation can also be used in HMMs.

PARTICLE FILTERING

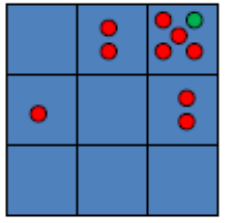
Particle Filtering

- 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, Pr(X)
  - Samples are called particles, each sample specifies one particular value that X might have.
  - Time per step is linear in the number of samples
  - But: number needed may be large
  - In memory: list of particles, not distributions over X
- This is how robot localization works in practice



REPRESENTATION: PARTICLES

- We approximate Pr(X) by a list of N particles. Each particle corresponds to single value for X: (1,1), (1,2), (2,2), (2,3) etc. These particles serve as a representation for Pr(X).
  - Generally, N << |X|
  - So the list of particles have size more manageable than storing all possible values for feature X.



- Pr(X=d) approximated by number of particles with value d (note than more than one particle can have the same value.
  - For many values X=d we might have Pr(X=d) = 0!
  - More particles, more accuracy

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

- Initially all particles have equal weight of 1

- In our examples the values for X are positions of the ghost on in the pacman grid.

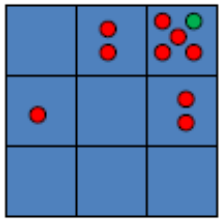
REPRESENTATION: PARTICLES

Representation: Particles

Pr(X=d) ≈ (aproximately equal)

(#particles with value d)/

(Total # of particles)



Particles:

(3,3)

(2,3)

(3,3)

(3,2)

(3,3)

(3,2)

(1,2)

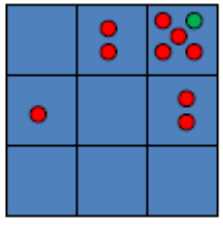
(3,3)

(3,3)

(2,3)

REPRESENTATION: PARTICLES

- We approximate Pr(X) by a list of N particles. Each particle corresponds to single value for X: (1,1), (1,2), (2,2), (2,3) etc. These particles serve as a representation for Pr(X).
  - Generally, N << |X|
  - So the list of particles have size more manageable than storing all possible values for feature X.



- Pr(X=d) approximated by number of particles with value d (note than more than one particle can have the same value.
  - For many values X=d we might have Pr(X=d) = 0!
  - More particles, more accuracy

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

- Initially all particles have equal weight of 1

- In our examples the values for X are positions of the ghost on in the pacman grid.

Filtering with Particles

- We want to approximate exact HMM inference with a collection of particles.
- This means that we must at every stage we must progress the particles to the next time step so that they approximate $\Pr(X_t|e_{t-1}, \cdots, e_1)$
- And then update the particles to account for the next observation so that they approximate $\Pr(X_t|e_t, \cdots, e_1)$

Particle Filtering: Elapse Time

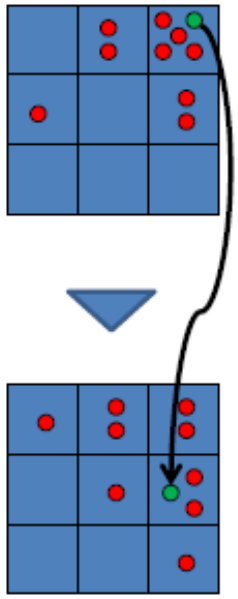
- Each particle is moved to the next step by sampling its next position from the transition model.
  - Each particle **p** is "asserting" that X_t=**p**. If X_t=**p** then it will transition to X_{t+1}=d with probability
    $$\Pr(X_{t+1} = d' | X_t = \boldsymbol{p})$$
  - Hence if we sample randomly from this distribution we will get a new particle
  - For each particle **p** we replace **p** with a new particle drawn randomly from the distribution
    $$\Pr(X_{t+1} | X_t = \boldsymbol{p})$$
    - 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)

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)

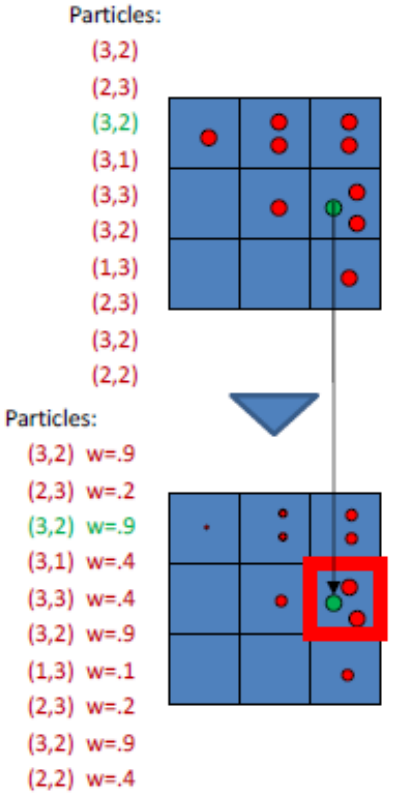


PARTICLE FILTERING: OBSERVE

Particle Filtering: Observe

- Observations are fixed, we don't sample these.
  - Instead we use the observation to reweight the particles.
  - Particles that are unlikely given the observation become less likely.
  - So for each particle **p** we set wt(**p**) = **Pr(e|X=p)**

    That is, **p** asserts that X has value **p**. If it did then the probability we would see observation **e** would be **Pr(e|X=p)**



PARTICLE FILTERING: OBSERVE

Particle Filtering: Observe

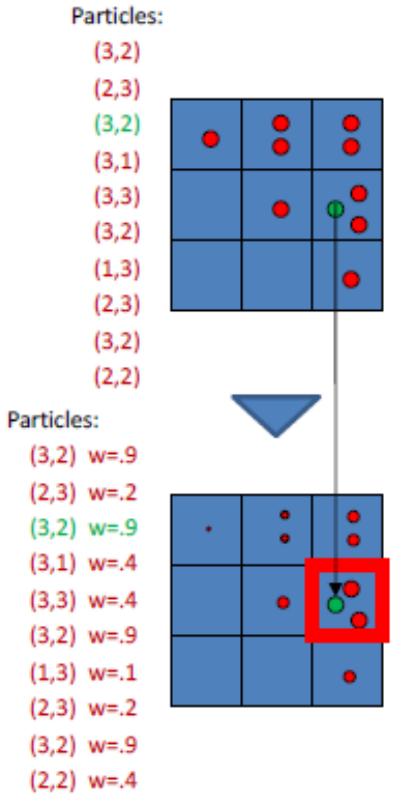
- But now with weighted particles our approximate probabilities
  Pr(X=d) = #particles with value d/
  Total number of particles

  No longer makes sense.

- We could instead use

$$\frac{\sum_{\boldsymbol{p}\ with\ value\ d} \boldsymbol{wt(p)}}{\sum_{all\ \boldsymbol{p}} wt(p)}$$

- As our approximate probability. But then we would also have to transfer the particle weights when time elapses. And we would accumulate many very low weight particles.

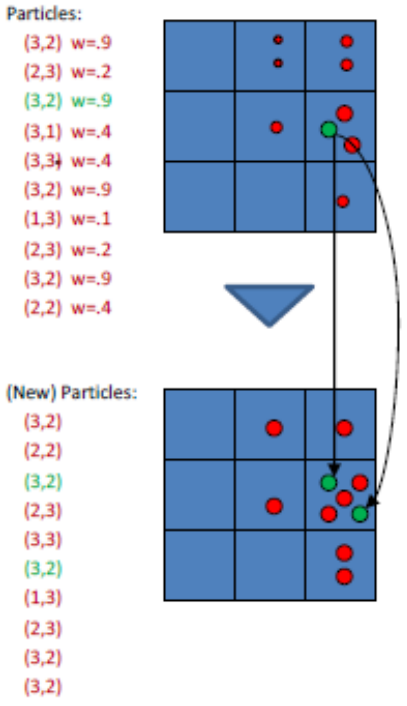


PARTICLE FILTERING: BACK TO UNIT WEIGHTS

- Rather than do this we convert back to weight 1 particles by resampling.
- If we are using N particles, we sample N weighted particles from our set of weighted particles. These is newly chosen weighted particles are given unit weight and used in the next time step. This is called **resampling** the particles.
- Specifically, for N times we select any particle **p** from our weighted particles with probability

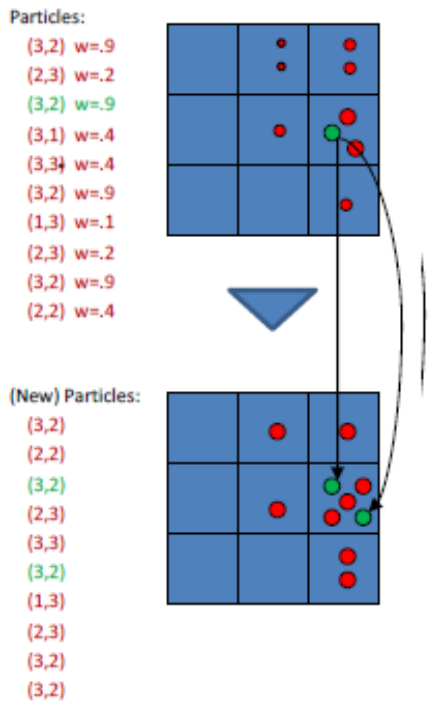
$$wt(\boldsymbol{p}) / \sum_{all\ p'} wt(p')$$

- E.g., we normalize the weights of all particles, and then select randomly from resulting distribution.



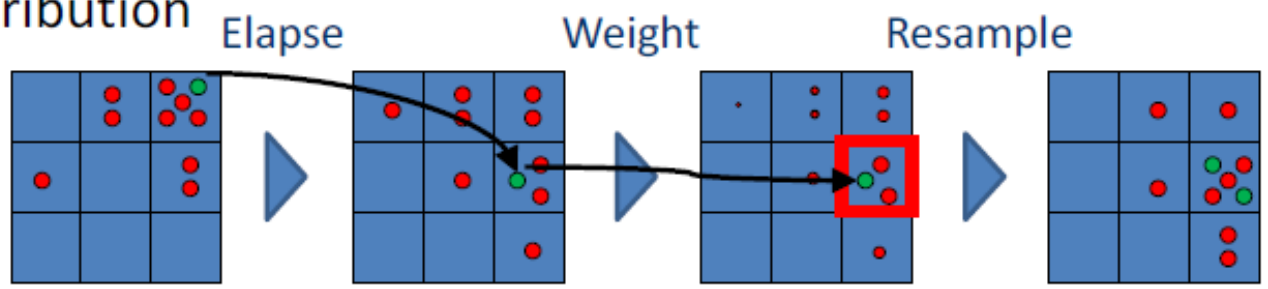
Particle Filtering: Back to unit weights

- E.g., particles = {p1, p2, p3, p4, p5}
- wt(p1) = 0.1, wt(p2) = 0, wt(p3) = .8, wt(p4) = .8, wt(p5) = .5
- Normalize the weights: wt(p1)=0.0455, wt(p2)=0, wt(p3)=0.36, wt(p4)=0.36, wtp5) = 0.23
- Now we sample this set of particles 5 times selecting p1 with probability 0.0455, p2 with probability 0, p3 with probability 0.36, p4 with probability 0.36, and p5 with probability 0.23
- Note that p2 drops out—it will never be sampled. This way we an get rid of samples with very low probability.



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)

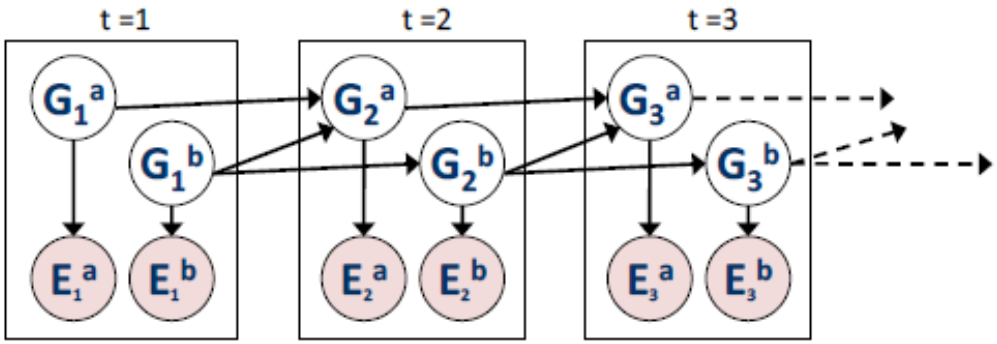
Particles:
- (3,2)
- (2,3)
- (3,2)
- (3,1)
- (3,3)
- (3,2)
- (1,3)
- (2,3)
- (3,2)
- (2,2)

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)

Dynamic Bayes Nets (DBNs)

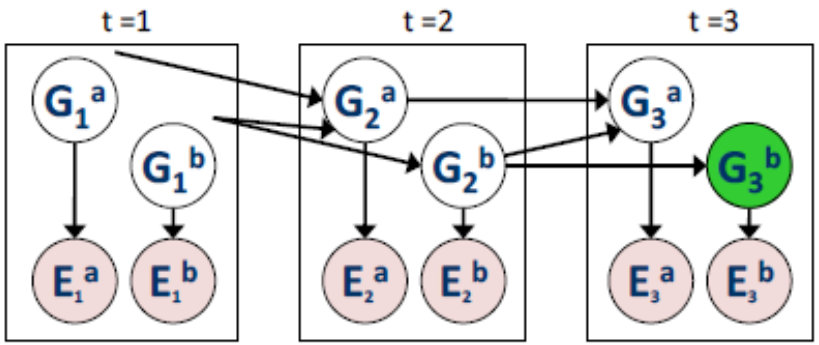
- We want to track multiple variables over time, using multiple sources of evidence
- Idea: Repeat a fixed Bayes net structure at each time
- Variables from time *t* can condition on those from *t-1*



- Dynamic Bayes nets are a generalization of HMMs

Exact Inference in DBNs

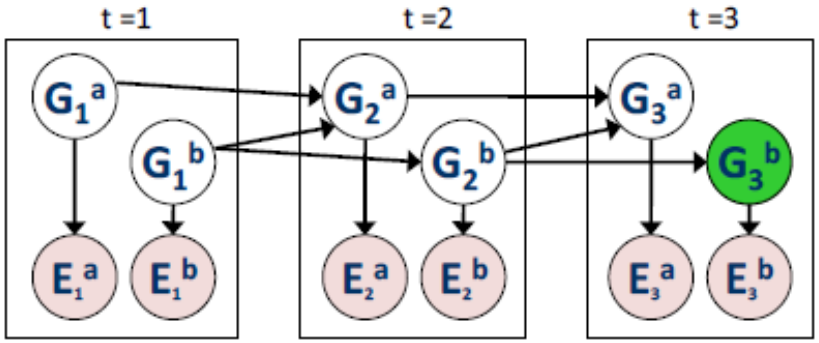
- Variable elimination applies to dynamic Bayes nets
- Procedure: "unroll" the network for T time steps, then eliminate variables until $P(X_T | e_{1:T})$ is computed



- Online belief updates: Eliminate all variables from the previous time step; store factors for current time only (just like HMMs but the updates are not as simple)

PARTICLE FILTERING IN DBN

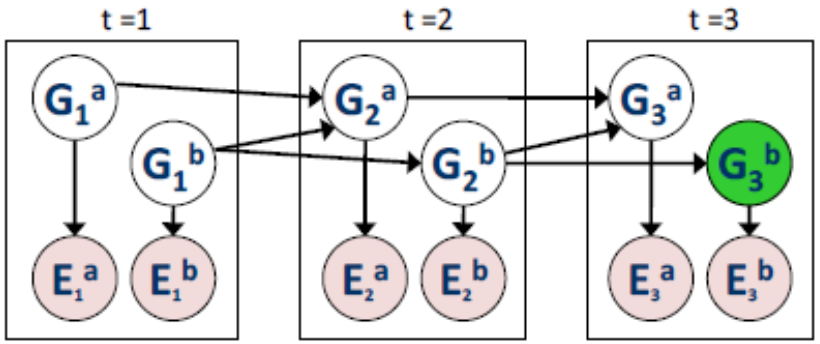
- As with HMMs we can have a set of particles. Now each particle must represent the state of all variables in the Bayes net at time step **t** except for the evidence variables (as these will become known)
- In this example, the particles will have a pair of values ($G_t^a = \mathbf{p}_a, G_t^b = \mathbf{p}_b$)



- To progress time for each particle we would sample a new value for G_{t+1}^a and G_{t+1}^b from the CPT given in the Bayes net: $Pr(G_{t+1}^a | G_t^a = \mathbf{p}_a)$ and $Pr(G_{t+1}^b | G_t^a = \mathbf{p}_a, G_t^b = \mathbf{p}_b)$. Note that we can sample each value $\mathbf{p'}_a$ **and** $\mathbf{p'}_b$ in the new particle **p'** separately since are G_{t+1}^a and G_{t+1}^b are independent in this Bayes net given $G_t^a = \mathbf{p}_a$ and $G_t^b = \mathbf{p}_b$

PARTICLE FILTERING IN DBN

- Then to update by observation, we would weight each particle $\mathbf{p}=(\mathbf{p}_a, \mathbf{p}_b)$ by the product of $Pr(e_{t+1}^a | G_{t+1}^a = \mathbf{p}_a) * Pr(e_{t+1}^b | G_{t+1}^b = \mathbf{p}_b)$
- Again because E_{t+1}^a and E_{t+1}^b are independent of each other given G_{t+1}^a, G_{t+1}^b in this Bayes net.



- If the variables and evidence at each time step t are dependent on each other then the updates to the particles are a bit more complex.