Announcements

Midterm

- Grades out today or tomorrow

Assignments

- HW7: Thu, 11/19, 11:59 pm

Introduction to Machine Learning

Hidden Markov Models

Instructor: Pat Virtue

Outline

1. Probability primer
2. Generative stories and Bayes nets
3. Learning HMM parameters
   - MLE for categorical distribution
4. Inference in Bayes Nets and HMMs
   - Forward algorithm (Markov chains)
   - HMM Queries
   - Message passing algorithms
     - Forward algorithm
     - Forward-backward algorithm
     - Viterbi algorithm

Markov Chain Inference

$$Y_1 \rightarrow Y_2 \rightarrow Y_3 \rightarrow Y_4 \dashrightarrow$$

If you know the transition probabilities, $P(Y_t \mid Y_{t-1})$, and you know $P(Y_4)$, write an equation to compute $P(Y_5)$.

$$P(Y_5) = \sum_{y_4} P(y_4, Y_5)$$
$$= \sum_{y_4} P(Y_5 \mid y_4) P(y_4)$$

Wouldn't it be quicker to just compute this from the joint? (No.)

$$P(Y_5) = \sum_{y_1, y_2, y_3, y_4} P(y_1, y_2, y_3, y_4, Y_5)$$

Forward algorithm (simple form)

Transition model

Probability from previous iteration

What is the state at time *t*?

$$P(Y_t) = \sum_{y_{t-1}} P(Y_{t-1}=y_{t-1}, Y_t)$$
$$= \sum_{y_{t-1}} P(Y_t \mid Y_{t-1}=y_{t-1})\, P(Y_{t-1}=y_{t-1})$$

Iterate this update starting at *t*=1

Inference: Hidden Markov Models

Image: http://ai.berkeley.edu/

HMM as Probability Model

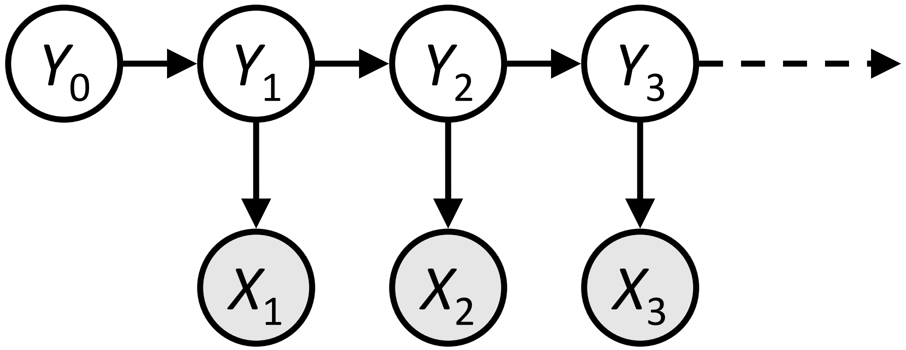
- Joint distribution for Markov model:

$$P(Y_0,\ldots, Y_T) = P(Y_0) \prod_{t=1:T} P(Y_t \mid Y_{t-1})$$

- Joint distribution for hidden Markov model:

$$P(Y_0, Y_1,X_1, \ldots, Y_T,X_T) = P(Y_0) \prod_{t=1:T} P(Y_t \mid Y_{t-1})\, P(X_t \mid Y_t)$$

- Future states are independent of the past given the present
- Current evidence is independent of everything else given the current state
- Are evidence variables independent of each other?



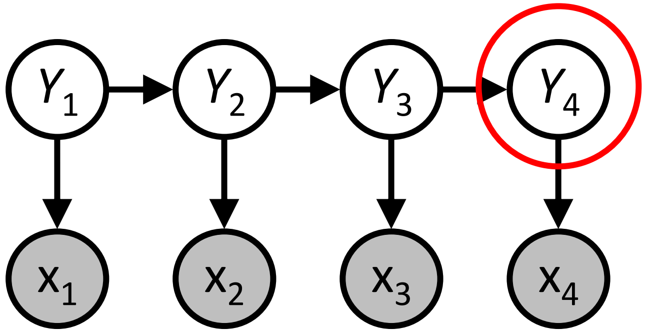
Notation alert!

Useful notation: $X_{a:b} = X_a, X_{a+1}, \ldots, X_b$

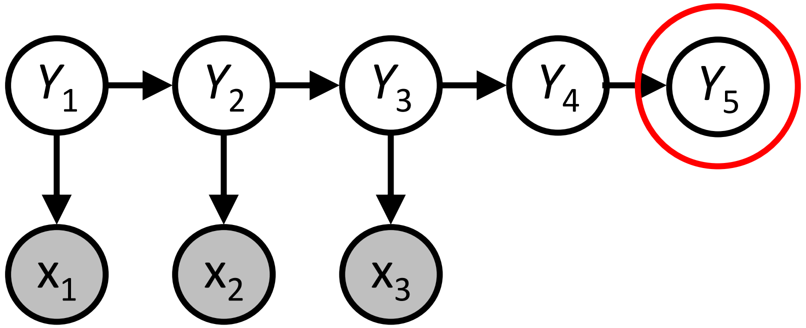
For example: $P(Y_{1:2} \mid x_{1:3}) = P(Y_1, Y_2 \mid x_1, x_2, x_3)$

HMM Queries

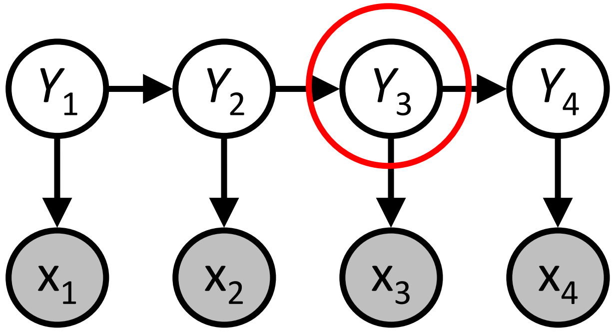
Filtering: $P(Y_t | x_{1:t})$



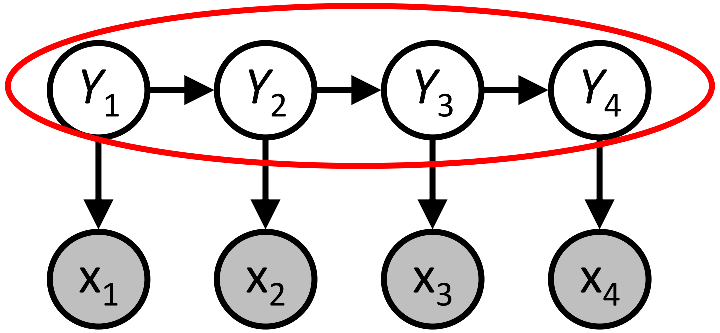
Prediction: $P(Y_{t+k} | x_{1:t})$



Smoothing: $P(Y_k | x_{1:t})$, $k<t$



Explanation: $P(Y_{1:t} | x_{1:t})$



Inference Tasks

Filtering: $P(Y_t \mid x_{1:t})$

- Belief state—input to the decision process of an autonomous agent

Prediction: $P(Y_{t+k} \mid x_{1:t})$ for $k > 0$

- Evaluation of possible action sequences; like filtering without the evidence

Smoothing: $P(Y_k \mid x_{1:t})$ for $0 \leq k < t$

- Better estimate of past states, essential for learning

Most likely explanation: $\text{argmax}_{y_{1:t}} P(y_{1:t} \mid x_{1:t})$

- Speech recognition, decoding with a noisy channel

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)

Molecular biology:

- Observations are nucleotides ACGT
- States are coding/non-coding/start/stop/splice-site etc.

Danielle Belgrave, Microsoft Research

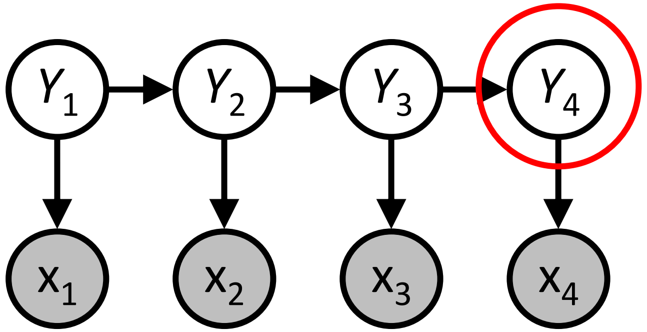
Microsoft | Research

https://www.microsoft.com/en-us/research/people/dabelgra/

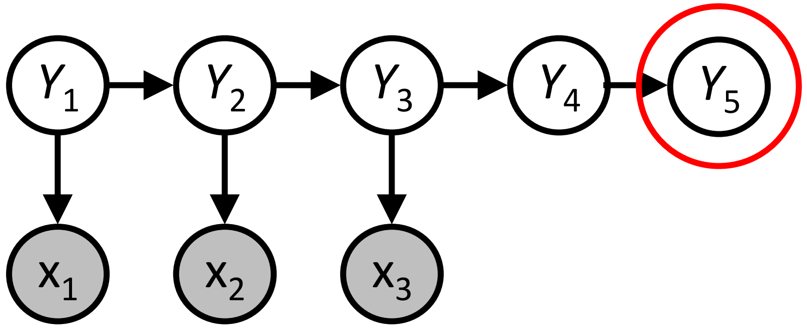
Developmental Profiles of Eczema, Wheeze, and Rhinitis:
Two Population-Based Birth Cohort Studies
Danielle Belgrave, et al. *PLOS Medicine*, 2014
https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.1001748

HMM Queries

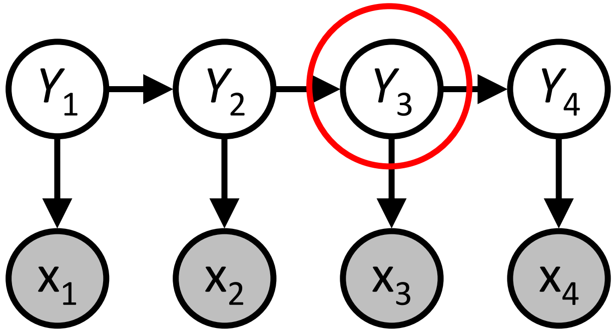
Filtering: $P(Y_t | x_{1:t})$



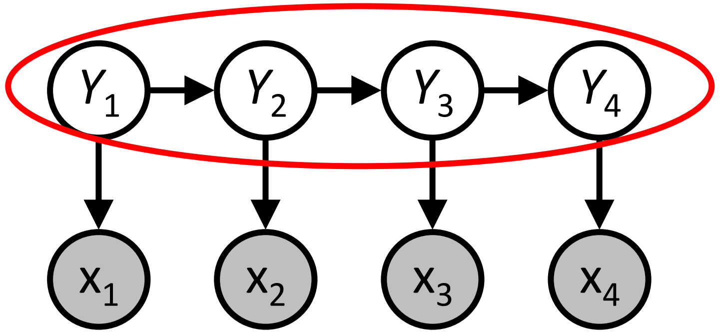
Prediction: $P(Y_{t+k} | x_{1:t})$



Smoothing: $P(Y_k | x_{1:t})$, $k<t$



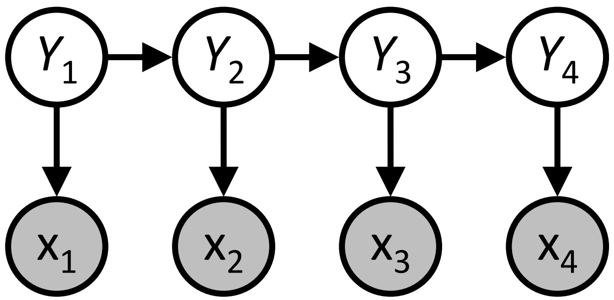
Explanation: $P(Y_{1:t} | x_{1:t})$



HMM Queries

Joint distribution: $P(Y_0, Y_1, X_1, \ldots, Y_T, X_T) = P(Y_0) \prod_{t=1:T} P(Y_t \mid Y_{t-1})\, P(X_t \mid Y_t)$

Filtering: $P(Y_t | x_{1:t})$

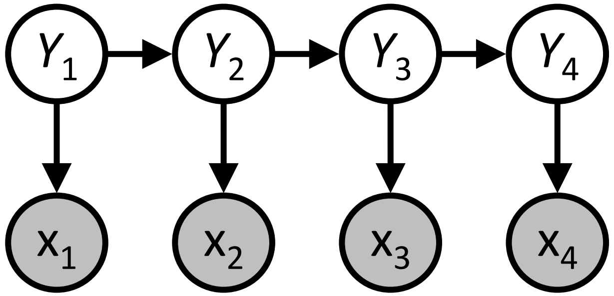


$$P(Y_t \mid x_{1:t}) = \frac{1}{z} \sum_{y_1, \cdots, y_{t-1}} P(Y_0, Y_1, X_1, \cdots Y_T, X_T)$$

HMM Queries

Joint distribution: $P(Y_0, Y_1, X_1, \ldots, Y_T, X_T) = P(Y_0) \prod_{t=1:T} P(Y_t \mid Y_{t-1}) \, P(X_t \mid Y_t)$

Smoothing: $P(Y_t | x_{1:T})$

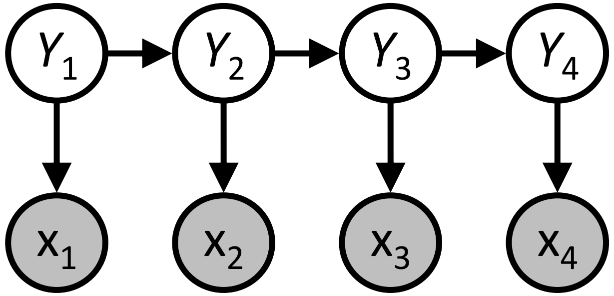


$$P(Y_t \mid x_{1:T}) = \frac{1}{z} \sum_{y_1, \cdots, y_{t-1}, y_{t+1}, \cdots, y_T} P(Y_0, Y_1, X_1, \cdots Y_T, X_T)$$

HMM Queries

Joint distribution: $P(Y_0, Y_1, X_1, \ldots, Y_T, X_T) = P(Y_0) \prod_{t=1:T} P(Y_t \mid Y_{t-1})\, P(X_t \mid Y_t)$

Explanation: $\underset{y_1,\cdots,y_T}{\operatorname{argmax}} P(y_{1:T} \mid x_{1:T})$



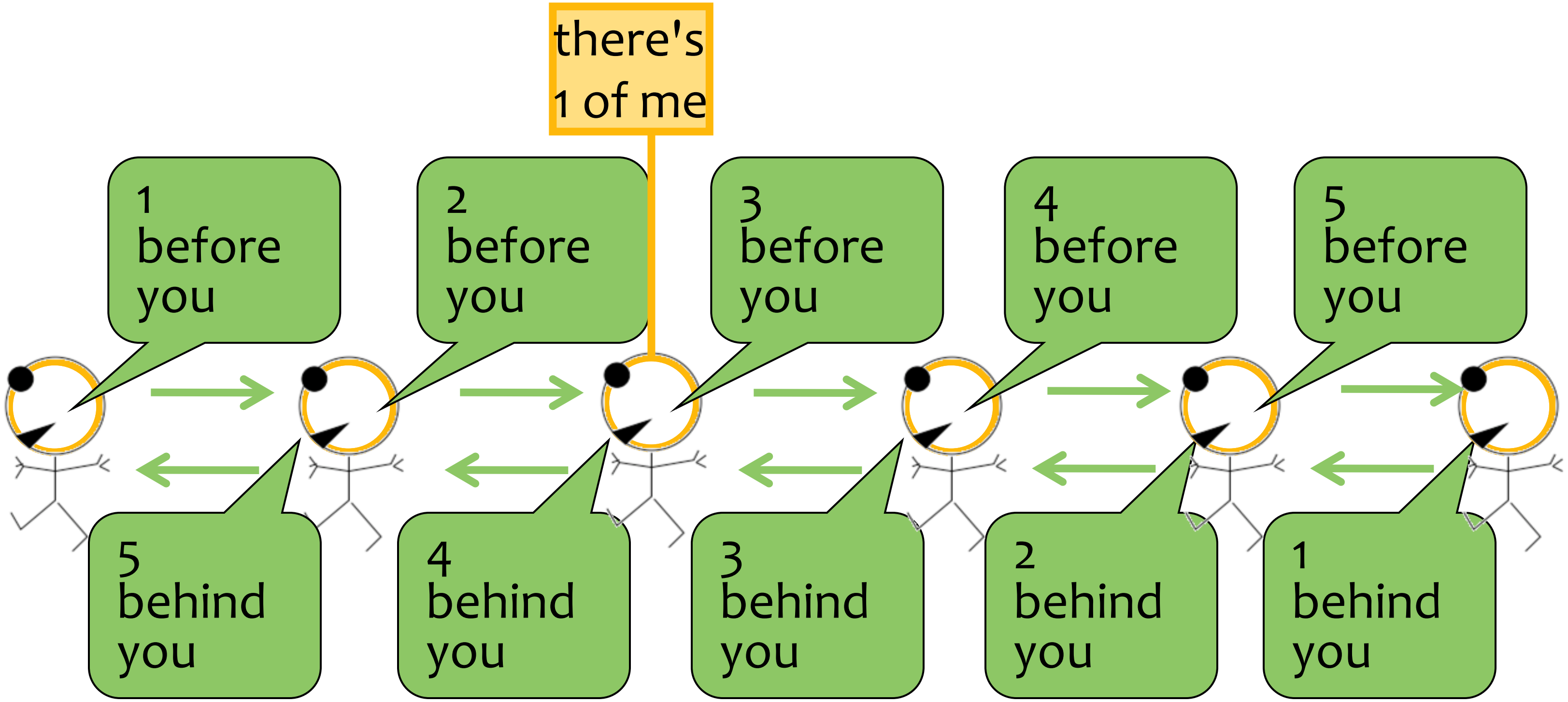
$$\underset{y_1,\cdots,y_T}{\operatorname{argmax}} P(y_{1:T} \mid x_{1:T})$$

$$= \underset{y_1,\cdots,y_T}{\operatorname{argmax}} P(y_{1:T}, x_{1:T})$$

$$= P(Y_0, Y_1, X_1, \cdots Y_T, X_T)$$

Great Ideas in ML: Message Passing

Count the soldiers

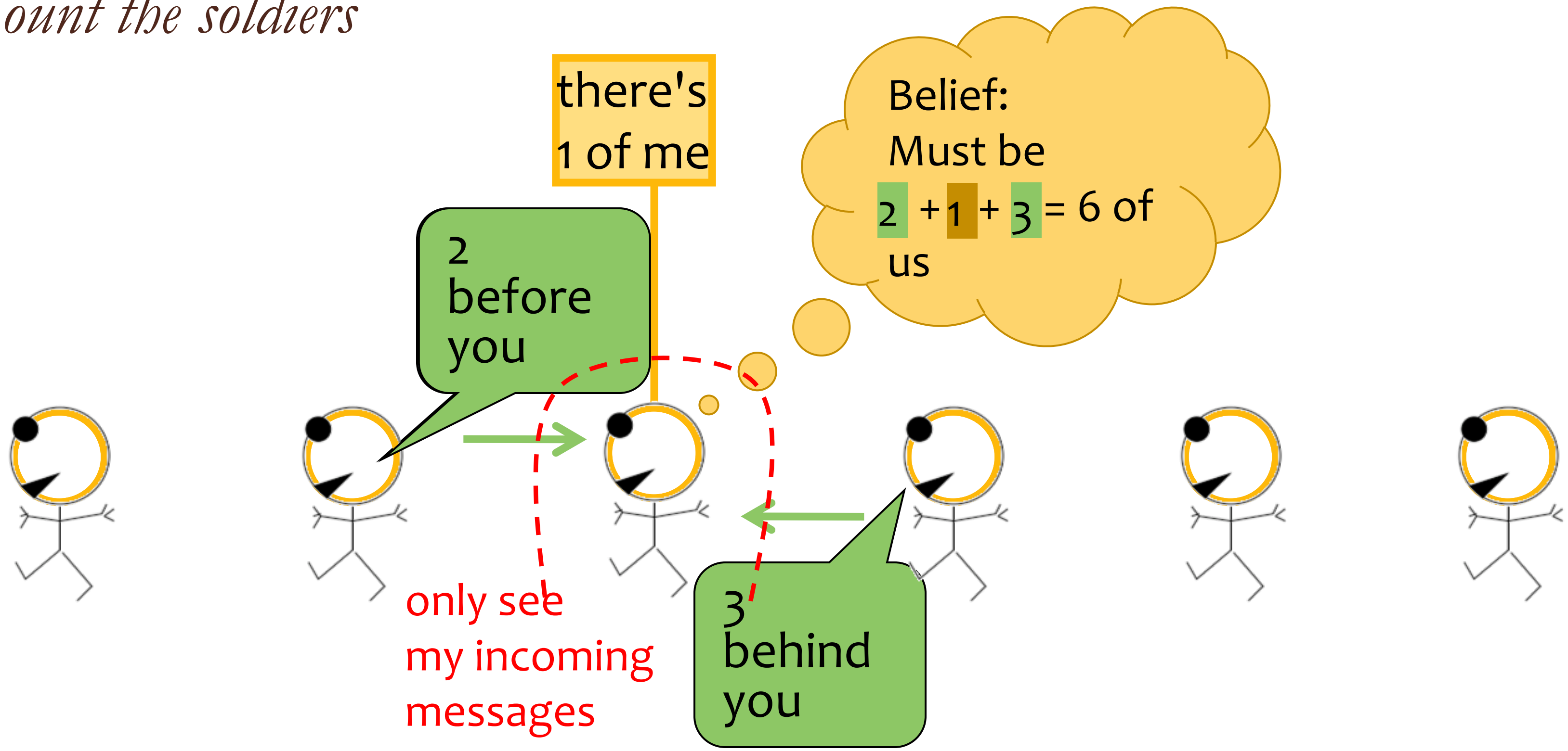


adapted from MacKay (2003) textbook

Slide credit: CMU MLD, Matt Gormley

Great Ideas in ML: Message Passing

Count the soldiers

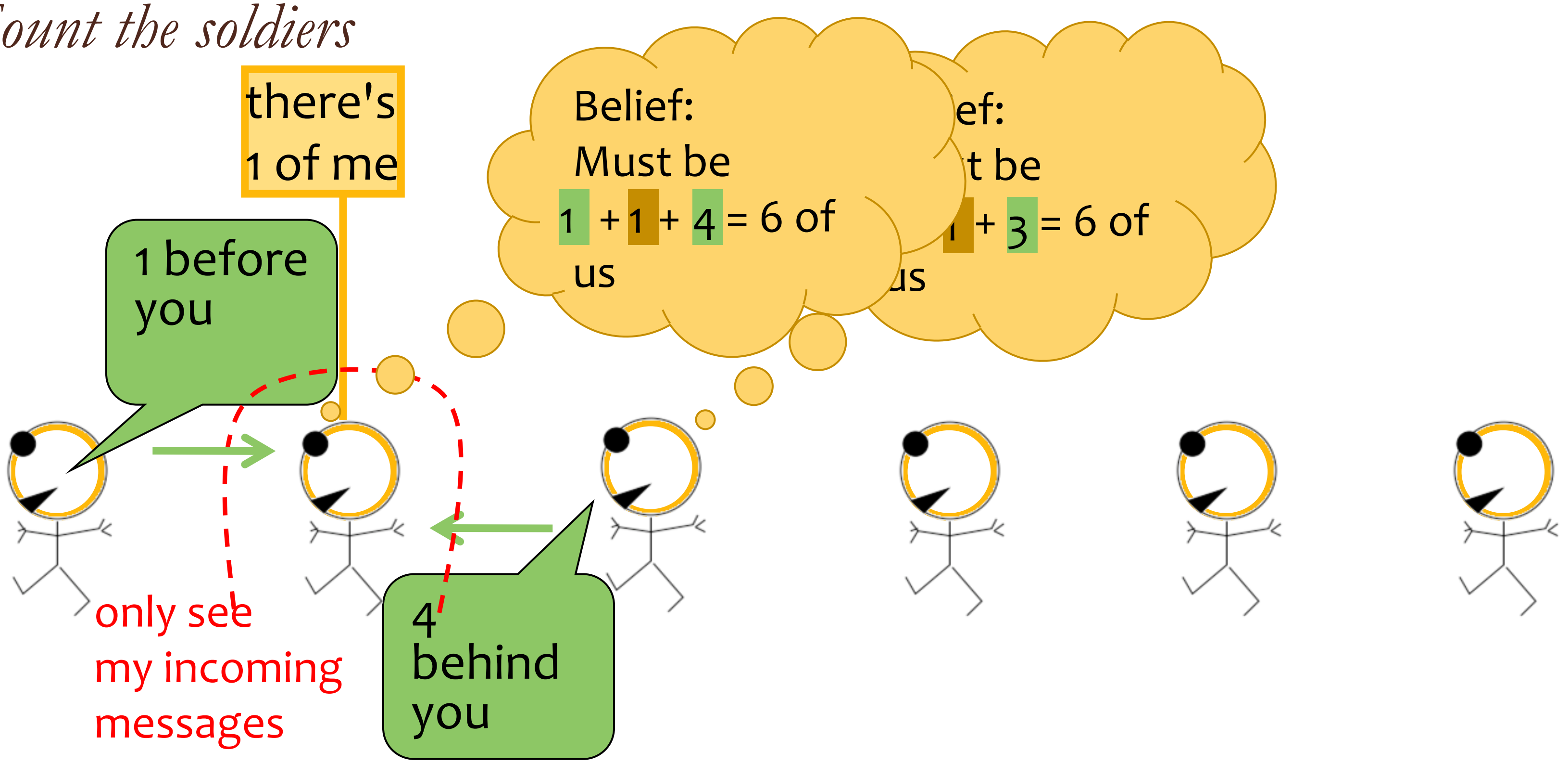


adapted from MacKay (2003) textbook

Slide credit: CMU MLD, Matt Gormley

Great Ideas in ML: Message Passing

Count the soldiers

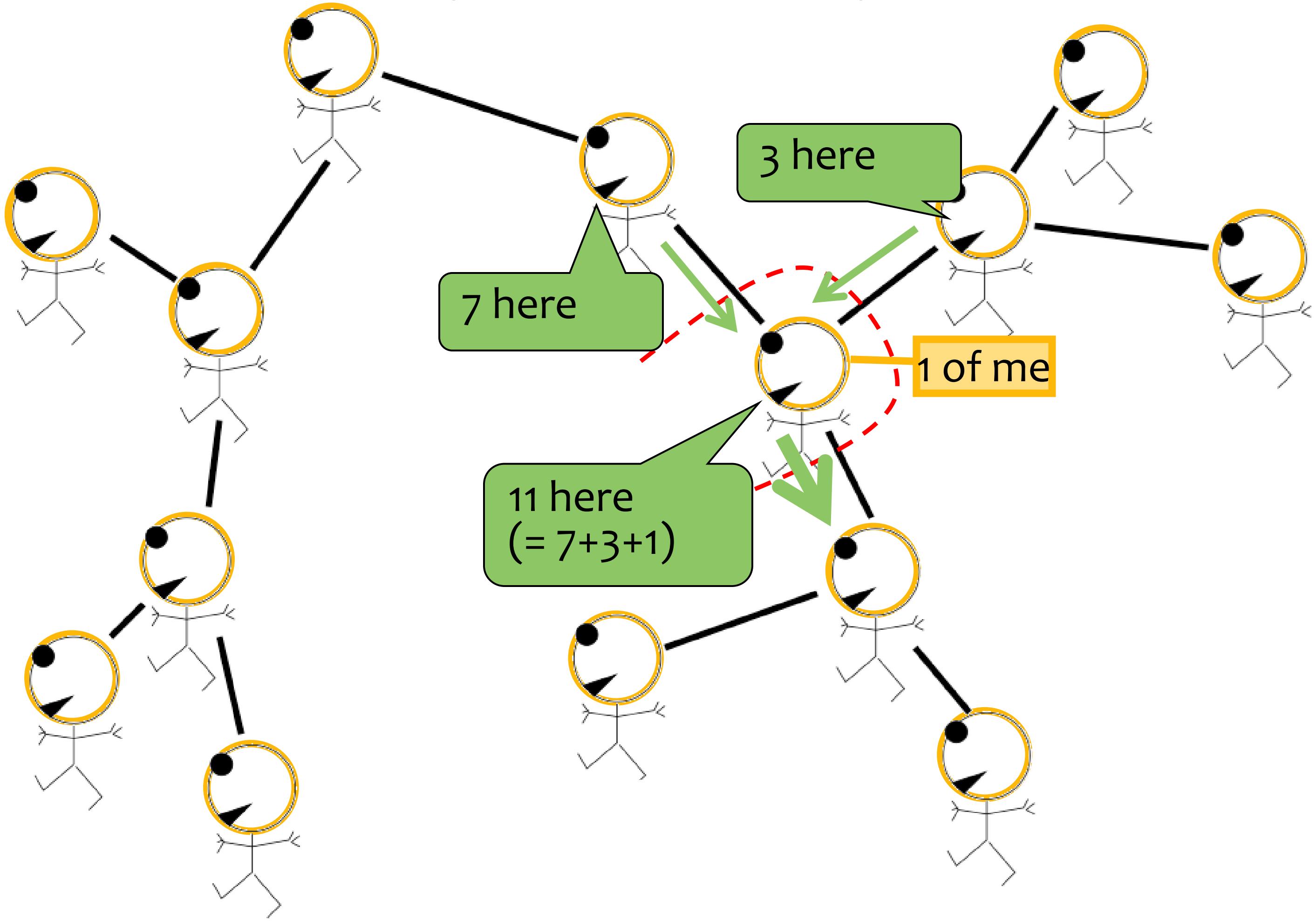


adapted from MacKay (2003) textbook

Slide credit: CMU MLD, Matt Gormley

Great Ideas in ML: Message Passing

Each soldier receives reports from all branches of tree

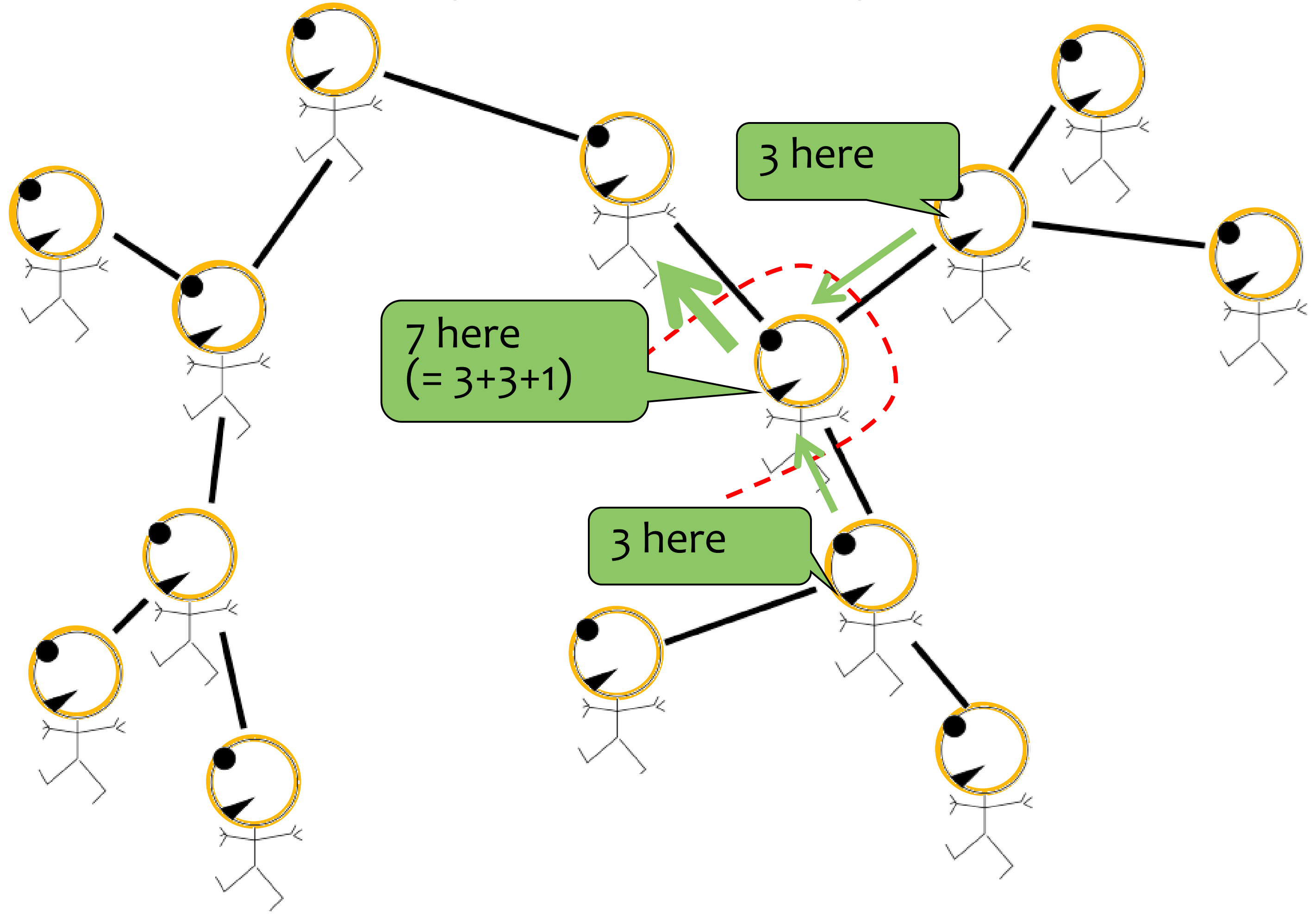


adapted from MacKay (2003) textbook

Slide credit: CMU MLD, Matt Gormley

Great Ideas in ML: Message Passing

Each soldier receives reports from all branches of tree

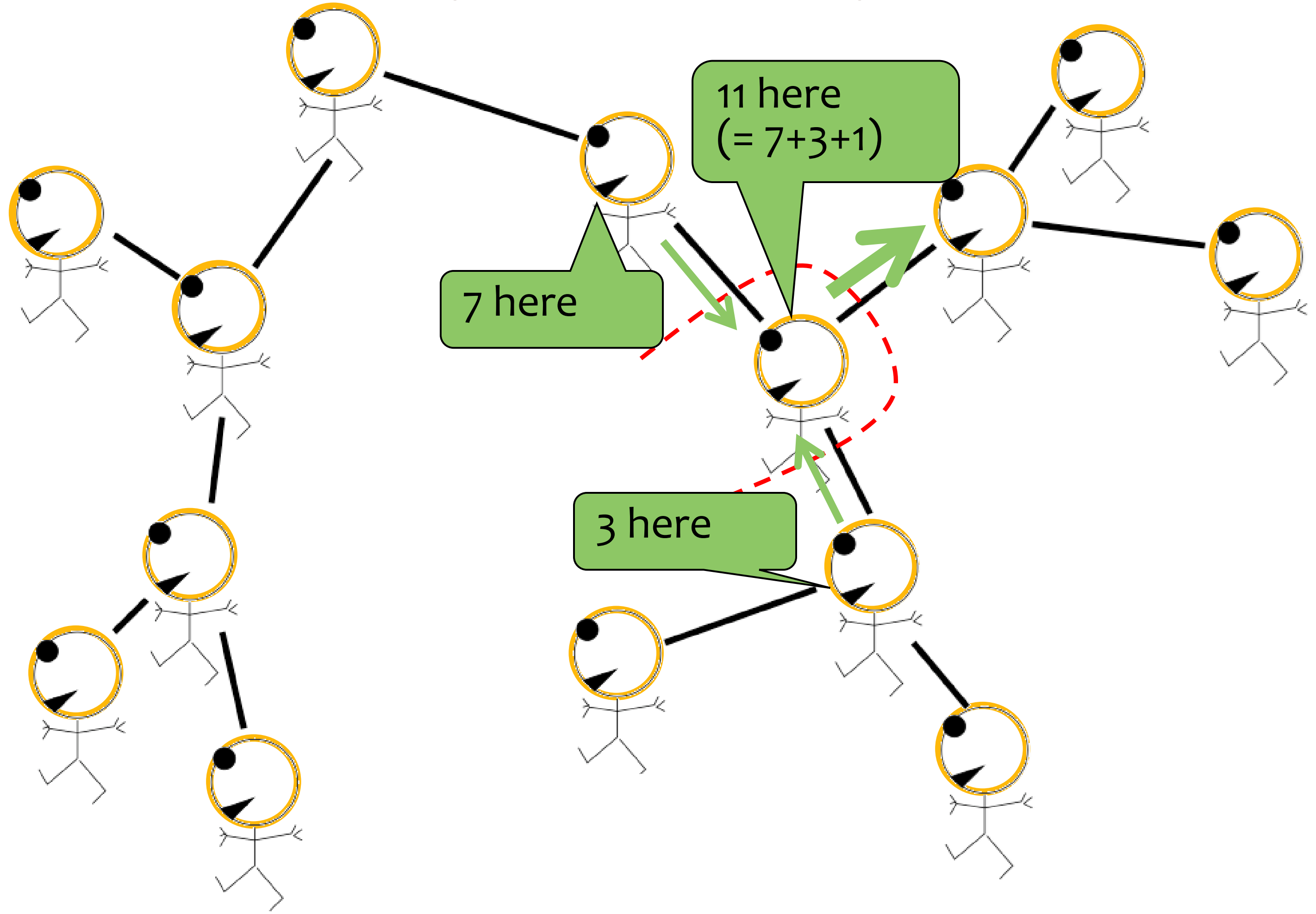


adapted from MacKay (2003) textbook

Slide credit: CMU MLD, Matt Gormley

Great Ideas in ML: Message Passing

Each soldier receives reports from all branches of tree

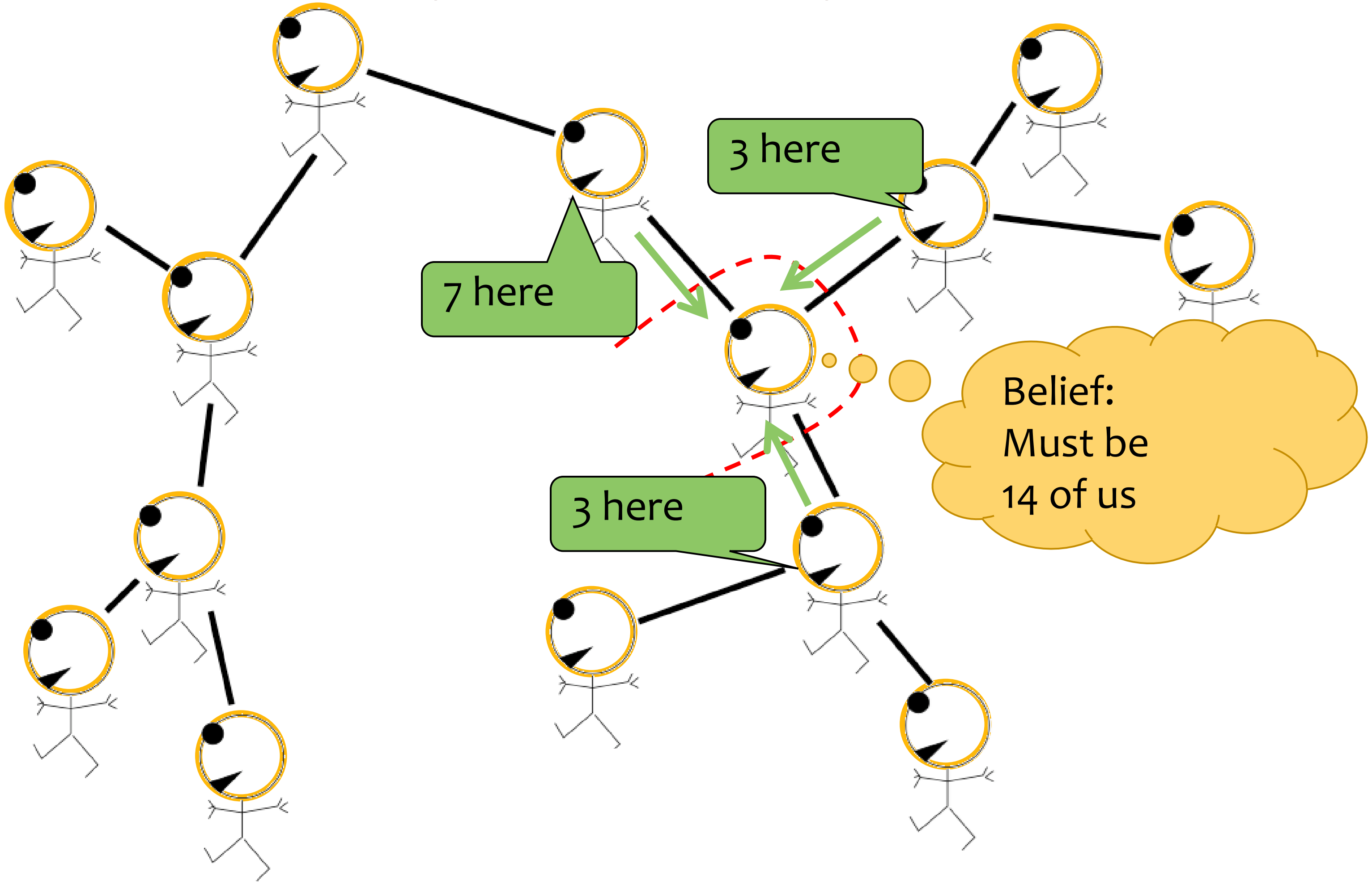


adapted from MacKay (2003) textbook

Slide credit: CMU MLD, Matt Gormley

Great Ideas in ML: Message Passing

Each soldier receives reports from all branches of tree

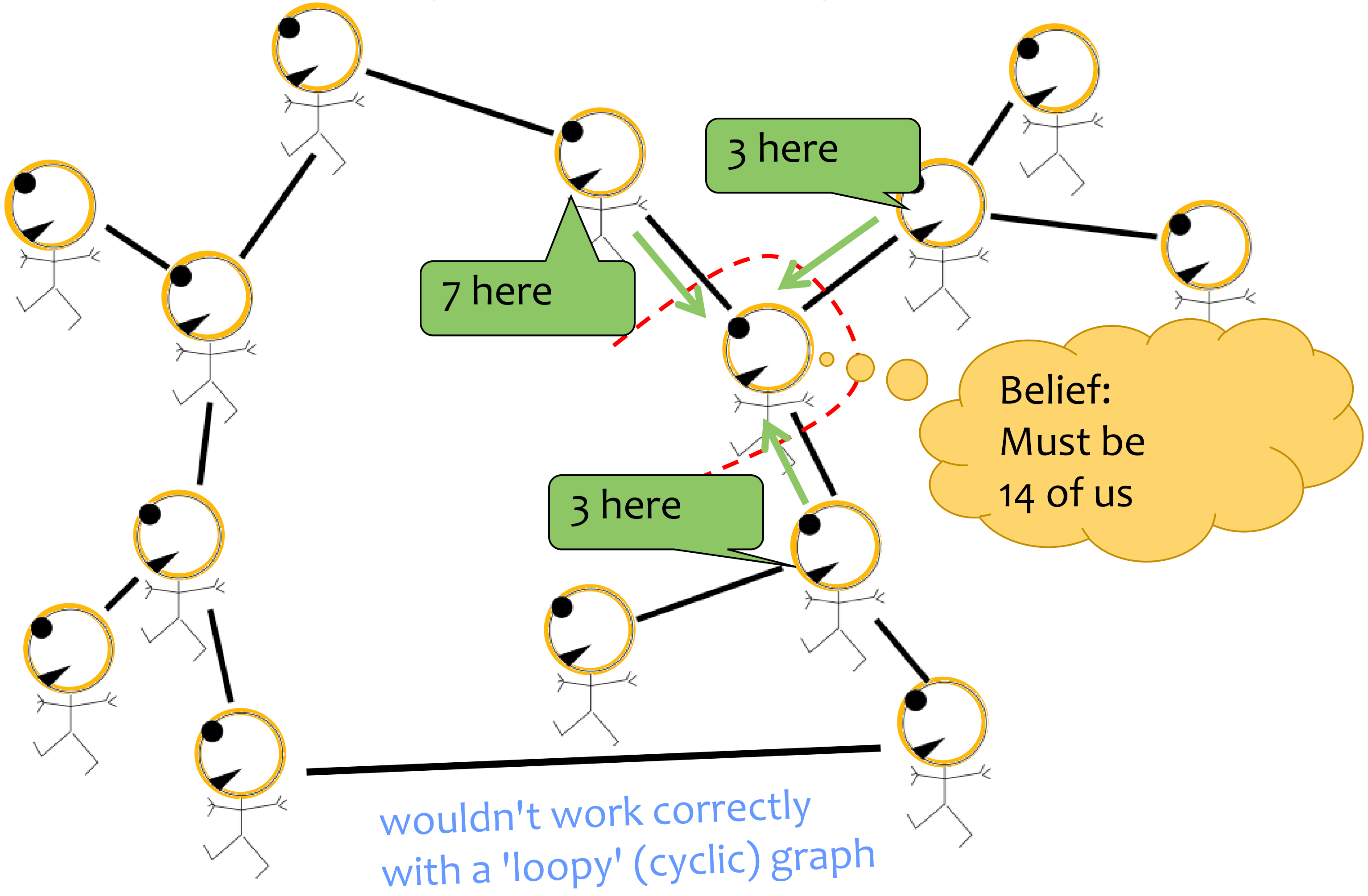


adapted from MacKay (2003) textbook

Slide credit: CMU MLD, Matt Gormley

Great Ideas in ML: Message Passing

Each soldier receives reports from all branches of tree



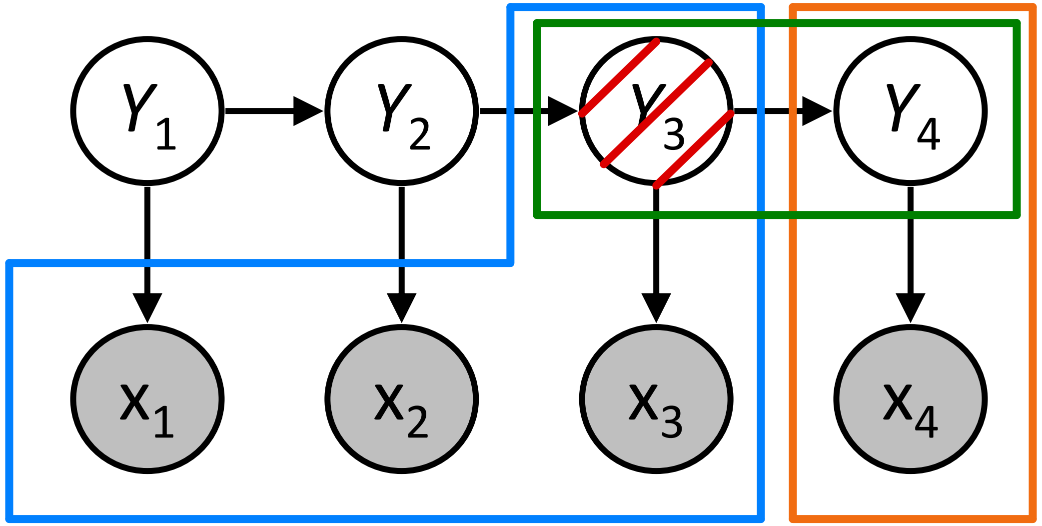
adapted from MacKay (2003) textbook

Slide credit: CMU MLD, Matt Gormley

Filtering: Forward Algoritm

$$P(Y_{t+1} | x_{1:t+1}) = \underbrace{\frac{1}{z}}_{\text{Normalize}} \; \underbrace{P(x_{t+1} | Y_{t+1})}_{\text{Update}} \; \underbrace{\sum_{y_t} P(Y_{t+1} | y_t)\, P(y_t \mid x_{1:t})}_{\text{Predict}}$$

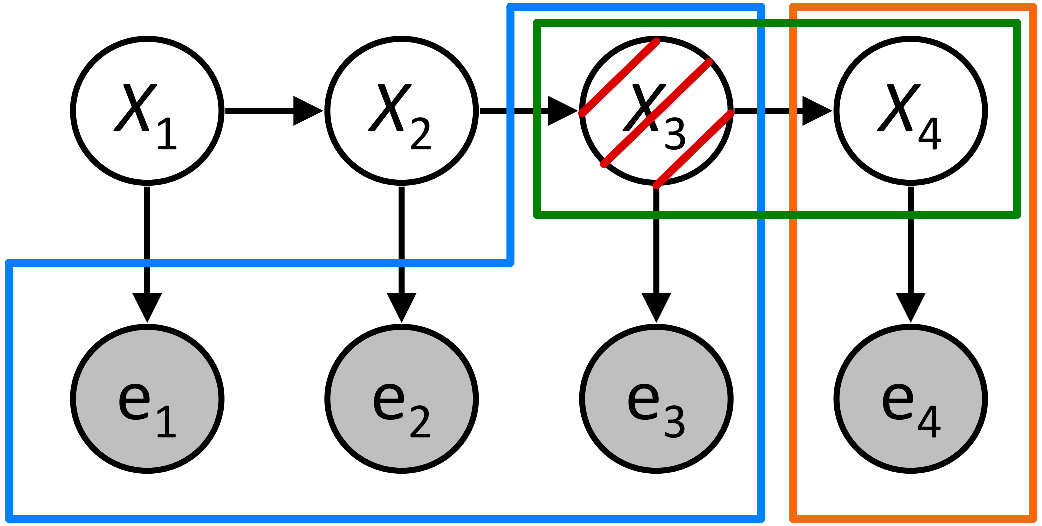
$$\boldsymbol{f}_{1:t+1} = \textsc{Forward}(\boldsymbol{f}_{1:t}\,,\, x_{t+1})$$



Filtering: Forward Algoritm

$$P(X_{t+1} | e_{1:t+1}) = \underbrace{\frac{1}{z}}_{\text{Normalize}} \underbrace{P(e_{t+1} | X_{t+1})}_{\text{Update}} \underbrace{\sum_{x_t} P(X_{t+1} | x_t)\, P(x_t | e_{1:t})}_{\text{Predict}}$$

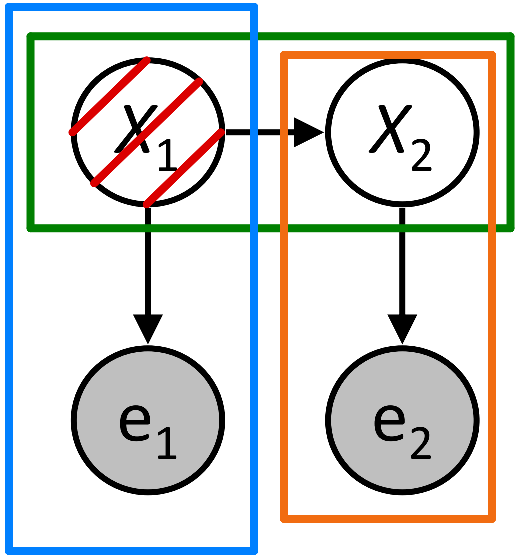
$\boldsymbol{f}_{1:t+1}$ = FORWARD($\boldsymbol{f}_{1:t}$, e_{t+1})



Filtering **Algorithm**

Query: What is the current state, given all of the current and past evidence?

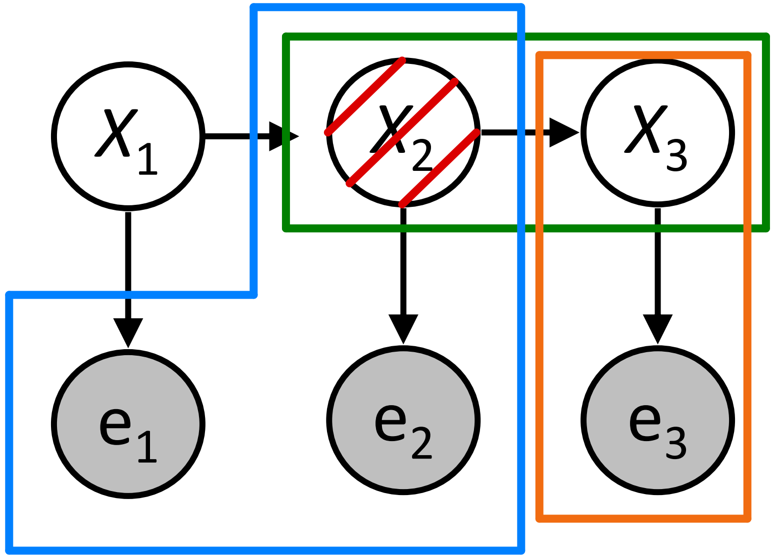
Marching **forward** through the HMM network



Filtering **Algorithm**

Query: What is the current state, given all of the current and past evidence?

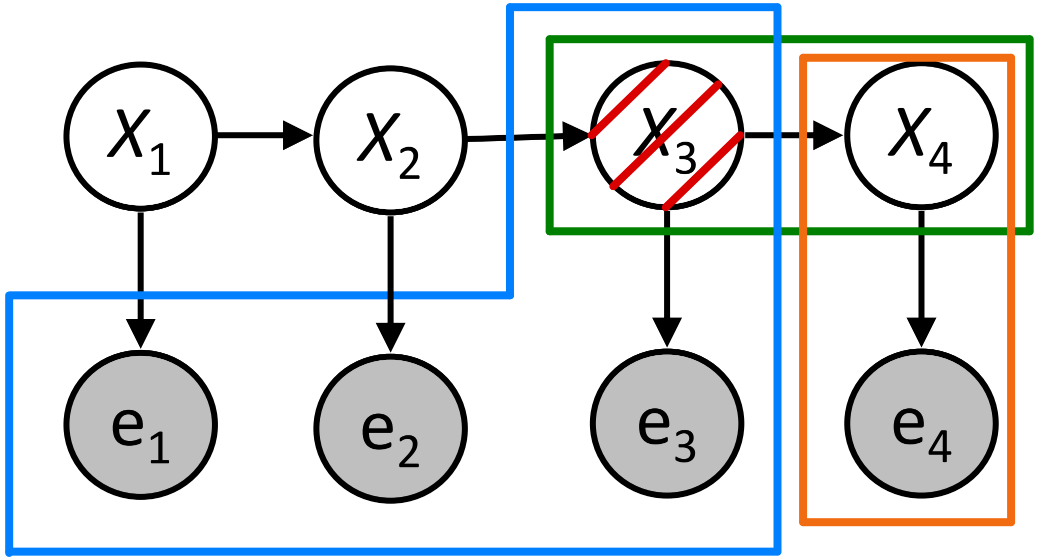
Marching **forward** through the HMM network



Filtering **Algorithm**

Query: What is the current state, given all of the current and past evidence?

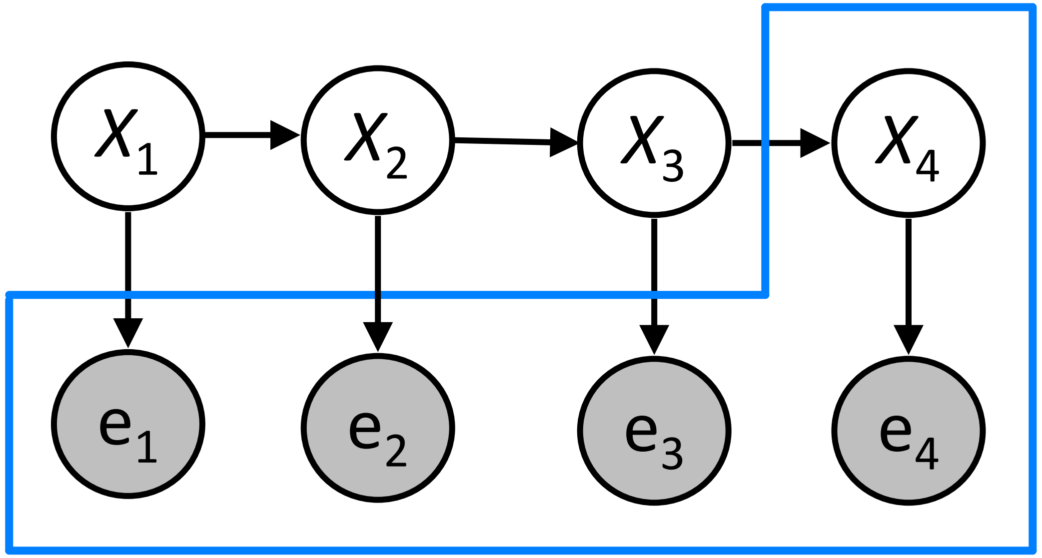
Marching **forward** through the HMM network



Filtering **Algorithm**

Query: What is the current state, given all of the current and past evidence?

Marching **forward** through the HMM network



Filtering Algorithm

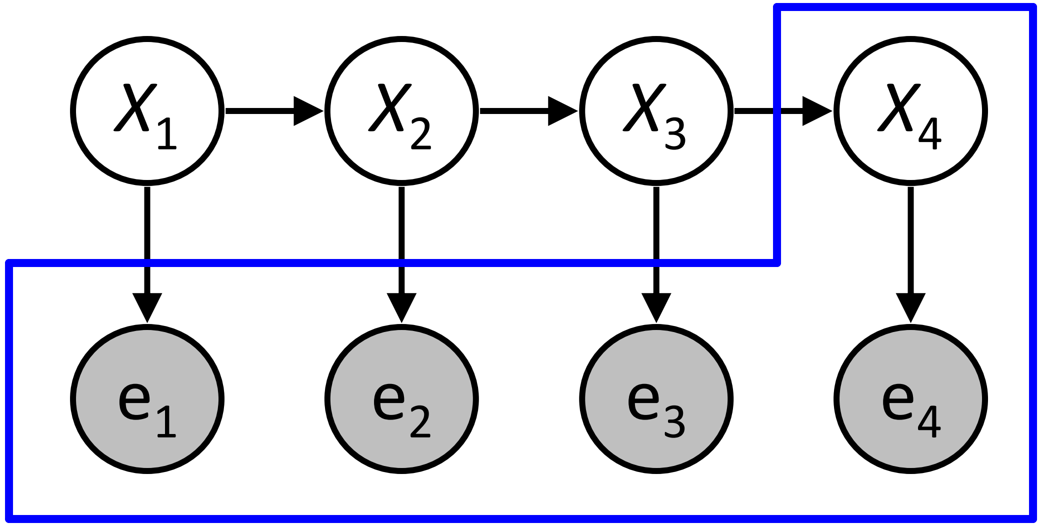
$$P(X_{t+1} | e_{1:t+1}) = \underbrace{\frac{1}{z}}_{\text{Normalize}} \underbrace{P(e_{t+1} | X_{t+1})}_{\text{Update}} \underbrace{\textstyle\sum_{x_t} P(X_{t+1} | x_t)\, P(x_t \mid e_{1:t})}_{\text{Predict}}$$

Filtering Algorithm

Query: What is the current state, given all of the current and past evidence?

Matching math with Bayes net

$$P(X_t \mid e_{1:t}) = P(X_t \mid e_t, e_{1:t-1})$$
$$= \alpha\, P(X_t, e_t \mid e_{1:t-1})$$



Def. of cond. probability with X_t, e_t

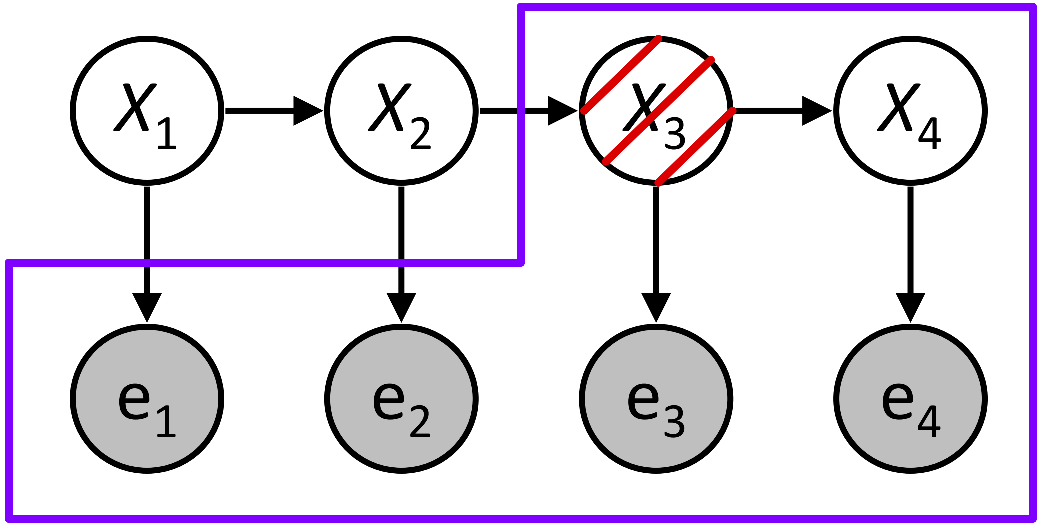
Filtering Algorithm

Query: What is the current state, given all of the current and past evidence?

Matching math with Bayes net

$$P(X_t \mid e_{1:t}) = P(X_t \mid e_t, e_{1:t-1})$$
$$= \alpha\, P(X_t, e_t \mid e_{1:t-1})$$
$$= \alpha \sum_{x_{t-1}} P(x_{t-1}, X_t, e_t \mid e_{1:t-1})$$

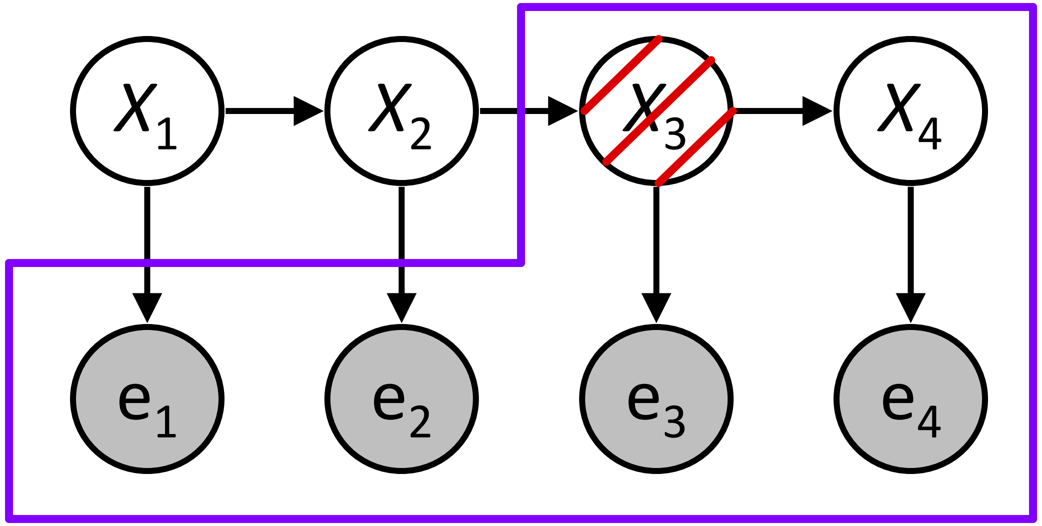
Summation over variable X_{t-1}



Filtering Algorithm

Query: What is the current state, given all of the current and past evidence?

Matching math with Bayes net



$$P(X_t \mid e_{1:t}) = P(X_t \mid e_t, e_{1:t-1})$$

$$= \alpha\, P(X_t, e_t \mid e_{1:t-1})$$

$$= \alpha \sum_{x_{t-1}} P(x_{t-1}, X_t, e_t \mid e_{1:t-1})$$

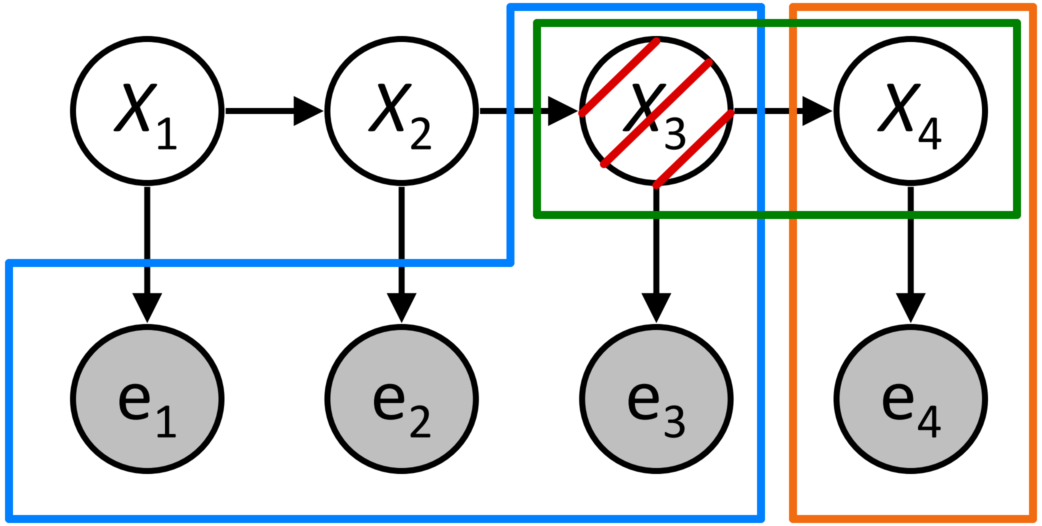
$$= \alpha \sum_{x_{t-1}} P(x_{t-1} \mid e_{1:t-1})\, P(X_t \mid x_{t-1}, e_{1:t-1})\, P(e_t \mid X_t, x_{t-1}, e_{1:t-1})$$

Chain rule with x_{t-1}, X_t, and e_t

Filtering Algorithm

Query: What is the current state, given all of the current and past evidence?

Matching math with Bayes net



$$P(X_t \mid e_{1:t}) = P(X_t \mid e_t, e_{1:t-1})$$

$$= \alpha\, P(X_t, e_t \mid e_{1:t-1})$$

$$= \alpha \sum_{x_{t-1}} P(x_{t-1}, X_t, e_t \mid e_{1:t-1})$$

$$= \alpha \sum_{x_{t-1}} P(x_{t-1} \mid e_{1:t-1})\, P(X_t \mid x_{t-1}, e_{1:t-1})\, P(e_t \mid X_t, x_{t-1}, e_{1:t-1})$$

Chain rule with x_{t-1}, X_t, and e_t

Filtering Algorithm

Query: What is the current state, given all of the current and past evidence?

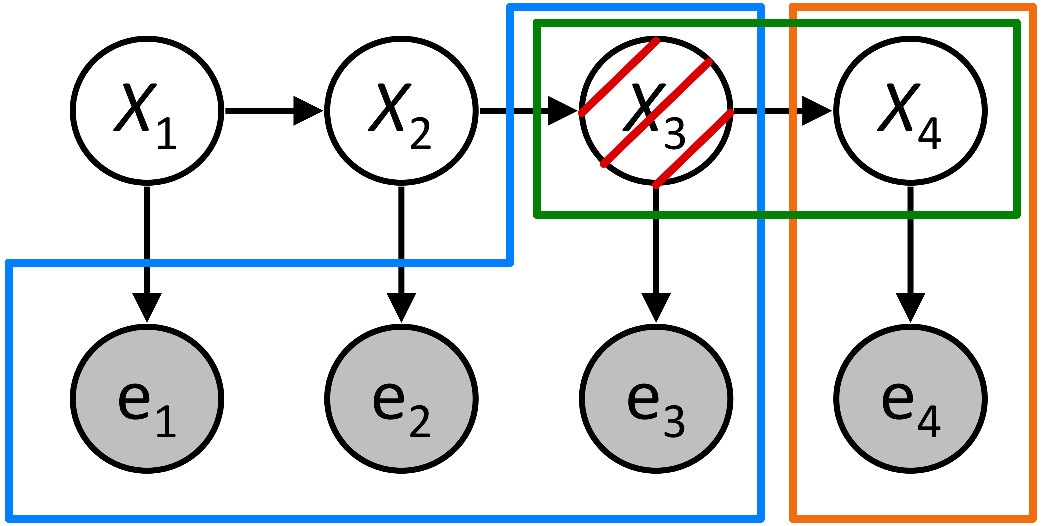
Matching math with Bayes net

$$P(X_t \mid e_{1:t}) = P(X_t \mid e_t, e_{1:t-1})$$

$$= \alpha \, P(X_t, e_t \mid e_{1:t-1})$$

$$= \alpha \sum_{x_{t-1}} P(x_{t-1}, X_t, e_t \mid e_{1:t-1})$$

$$= \alpha \sum_{x_{t-1}} P(x_{t-1} \mid e_{1:t-1}) \, P(X_t \mid x_{t-1}) \, P(e_t \mid X_t)$$



Bayes net conditional independence

Filtering Algorithm

Query: What is the current state, given all of the current and past evidence?

Matching math with Bayes net

$$P(X_t \mid e_{1:t}) = P(X_t \mid e_t, e_{1:t-1})$$

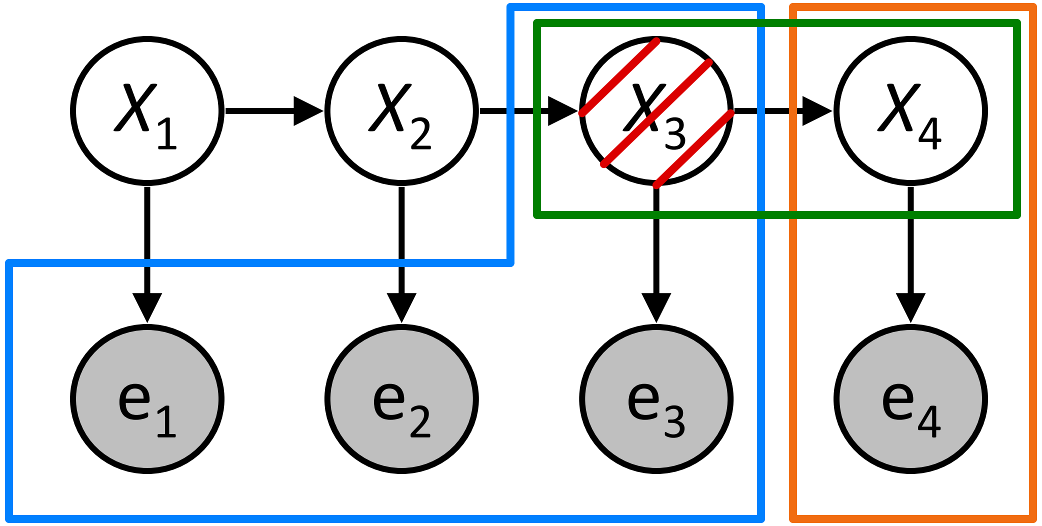
$$= \alpha\, P(X_t, e_t \mid e_{1:t-1})$$

$$= \alpha \sum_{x_{t-1}} P(x_{t-1}, X_t, e_t \mid e_{1:t-1})$$

$$= \alpha \sum_{x_{t-1}} P(x_{t-1} \mid e_{1:t-1})\, P(X_t \mid x_{t-1})\, P(e_t \mid X_t)$$

$$= \alpha\, P(e_t \mid X_t) \sum_{x_{t-1}} P(X_t \mid x_{t-1})\, P(x_{t-1} \mid e_{1:t-1})$$

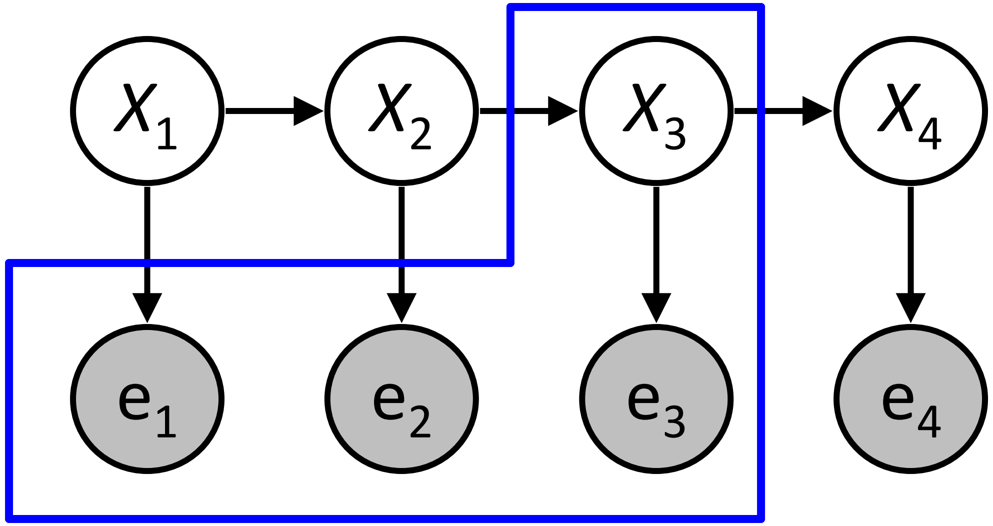
Pulling $P(e_t \mid X_t)$ out of the summation



Filtering Algorithm

Query: What is the current state, given all of the current and past evidence?

Matching math with Bayes net



$$P(X_t \mid e_{1:t}) = P(X_t \mid e_t, e_{1:t-1})$$

$$= \alpha\, P(X_t, e_t \mid e_{1:t-1})$$

$$= \alpha \sum_{x_{t-1}} P(x_{t-1}, X_t, e_t \mid e_{1:t-1})$$

$$= \alpha \sum_{x_{t-1}} P(x_{t-1} \mid e_{1:t-1})\, P(X_t \mid x_{t-1})\, P(e_t \mid X_t)$$

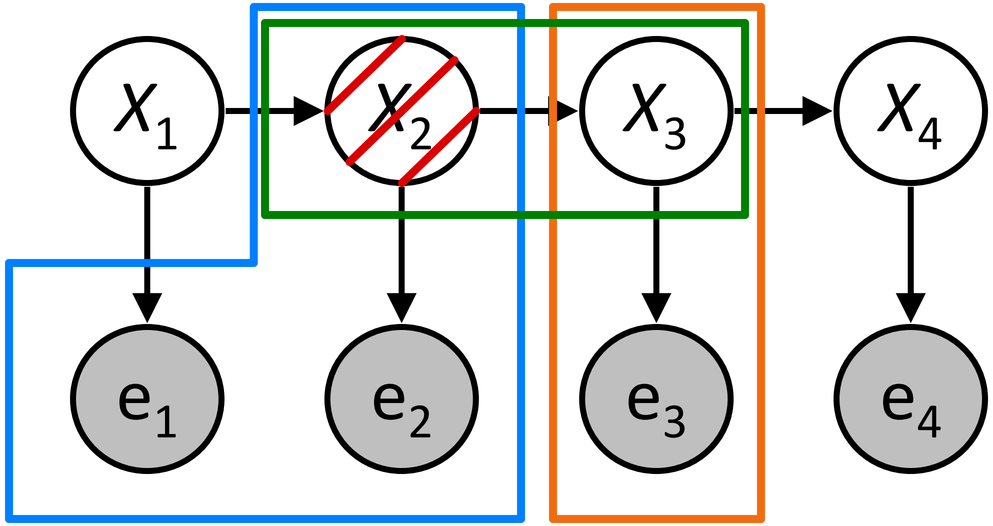
$$= \alpha\, P(e_t \mid X_t) \sum_{x_{t-1}} P(X_t \mid x_{t-1})\, P(x_{t-1} \mid e_{1:t-1})$$

Recursion!

Filtering Algorithm

Query: What is the current state, given all of the current and past evidence?

Matching math with Bayes net



$$P(X_t \mid e_{1:t}) = P(X_t \mid e_t, e_{1:t-1})$$

$$= \alpha\, P(X_t, e_t \mid e_{1:t-1})$$

$$= \alpha \sum_{x_{t-1}} P(x_{t-1}, X_t, e_t \mid e_{1:t-1})$$

$$= \alpha \sum_{x_{t-1}} P(x_{t-1} \mid e_{1:t-1})\, P(X_t \mid x_{t-1})\, P(e_t \mid X_t)$$

$$= \alpha\, P(e_t \mid X_t) \sum_{x_{t-1}} P(X_t \mid x_{t-1})\, P(x_{t-1} \mid e_{1:t-1})$$

Recursion!

Filtering Algorithm

$$P(X_{t+1} | e_{1:t+1}) = \underbrace{\frac{1}{z}}_{\text{Normalize}} \underbrace{P(e_{t+1} | X_{t+1})}_{\text{Update}} \underbrace{\sum_{x_t} P(X_{t+1} | x_t)\, P(x_t \mid e_{1:t})}_{\text{Predict}}$$

$\boldsymbol{f}_{1:t+1} = \text{FORWARD}(\boldsymbol{f}_{1:t}, e_{t+1})$

Cost per time step: $O(|X|^2)$ where $|X|$ is the number of states

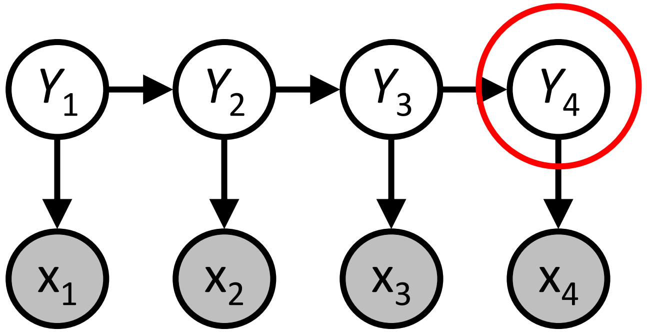
Time and space costs are ***constant***, independent of t

$O(|X|^2)$ is infeasible for models with many state variables

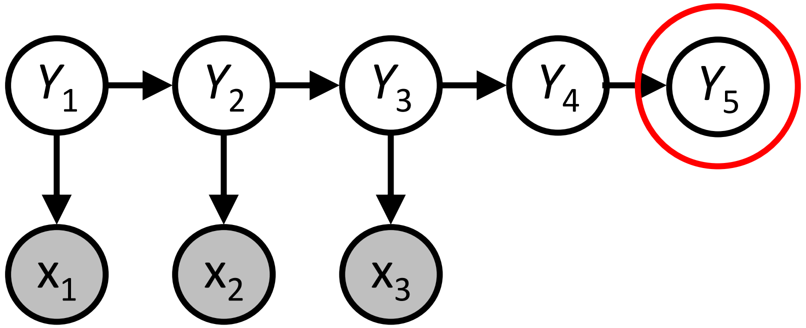
We get to invent really cool approximate filtering algorithms

HMM Queries

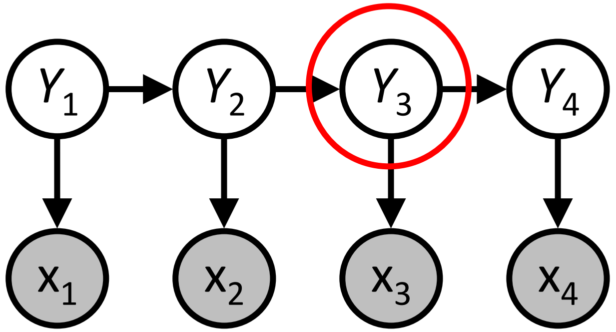
Filtering: $P(Y_t | x_{1:t})$



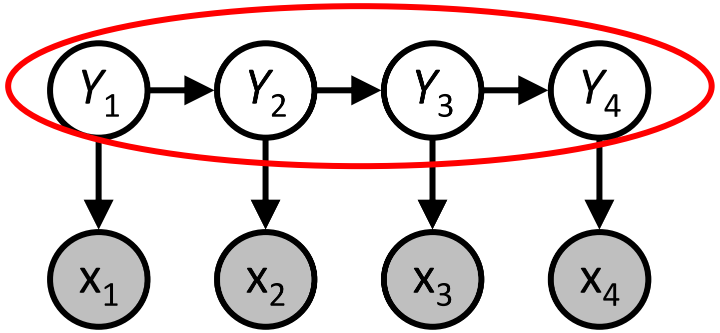
Prediction: $P(Y_{t+k} | x_{1:t})$



Smoothing: $P(Y_k | x_{1:t})$, $k<t$



Explanation: $P(Y_{1:t} | x_{1:t})$



Smoothing: Forward-Backward Algorithm

1. Forward pass from beginning to end

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

2. Backward pass from end to t

$$P(x_{t+1:T} \mid Y_t) = \sum_{y_{t+1}} P(x_{t+1} \mid y_{t+1}) \, P(y_{t+1} \mid Y_t) \, P(x_{t+2:T} \mid y_{t+1})$$

3. Combine forward and backward to answer query

$$P(Y_t \mid x_{1:T}) = \frac{1}{z} P(Y_t, x_{1:t}) \, P(x_{t+1:T} \mid Y_t)$$

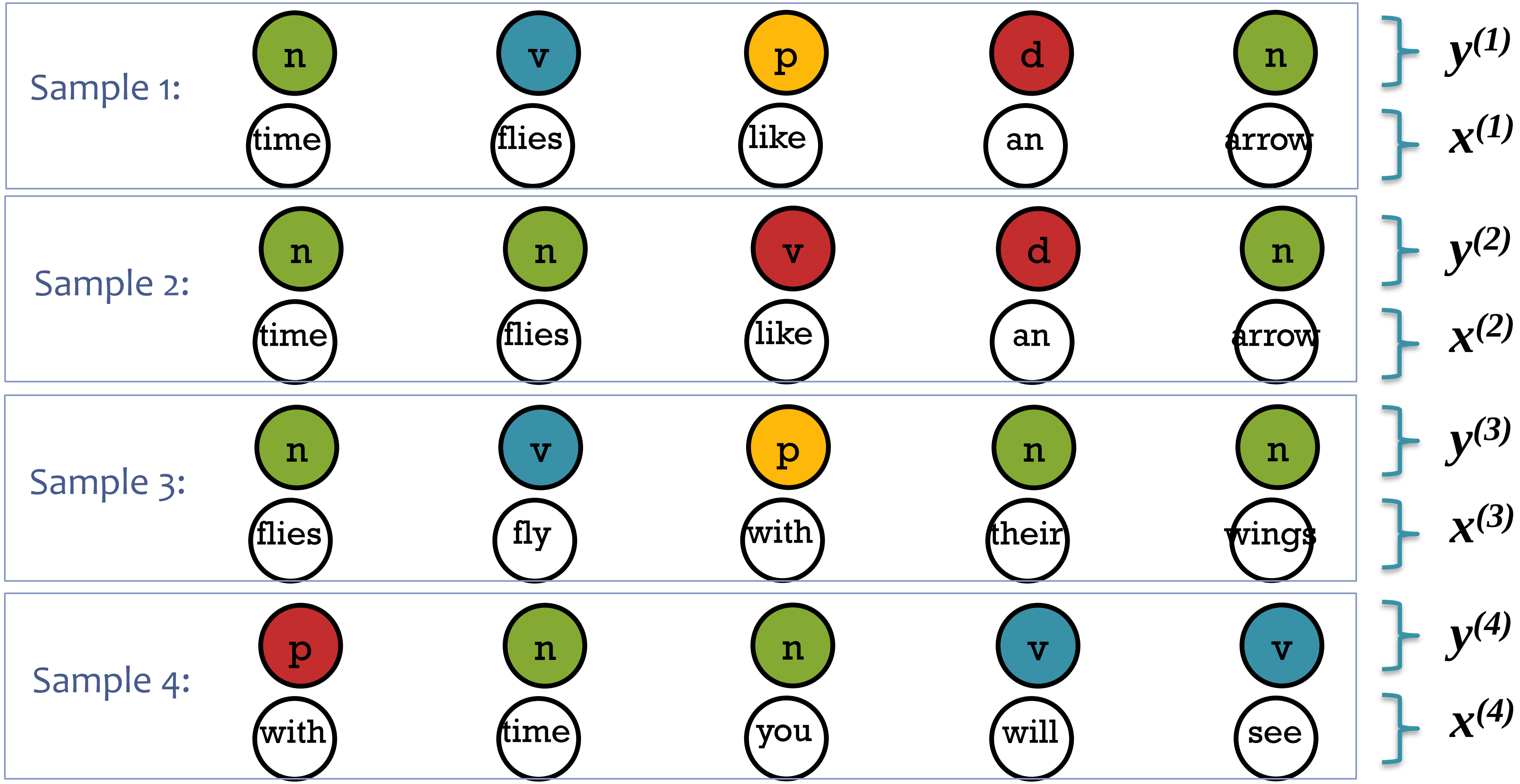
Course Survey(s)

See Piazza for course survey

(Some of you: see e-mail for research feedback survey)

Dataset for Supervised Part-of-Speech (POS) Tagging

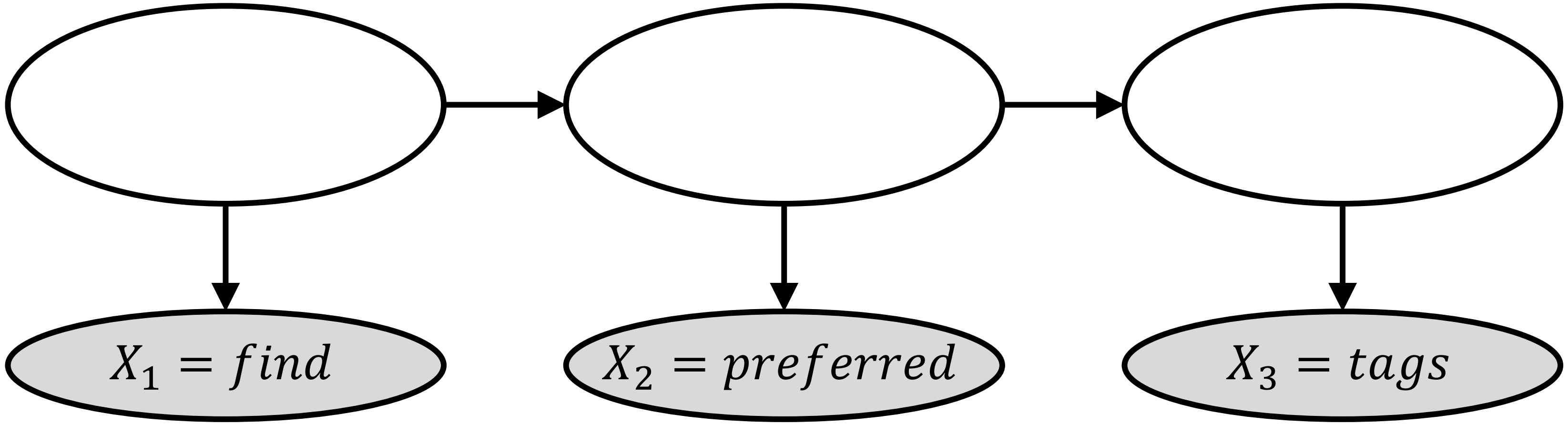
Data: $\mathcal{D} = \{\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}\}_{n=1}^{N}$



Slide credit: CMU MLD, Matt Gormley

Queries for Part-of-Speech (POS) Tagging

What are the POS tags for the sentence $X_{1:3}$ = "find preferred tags"?



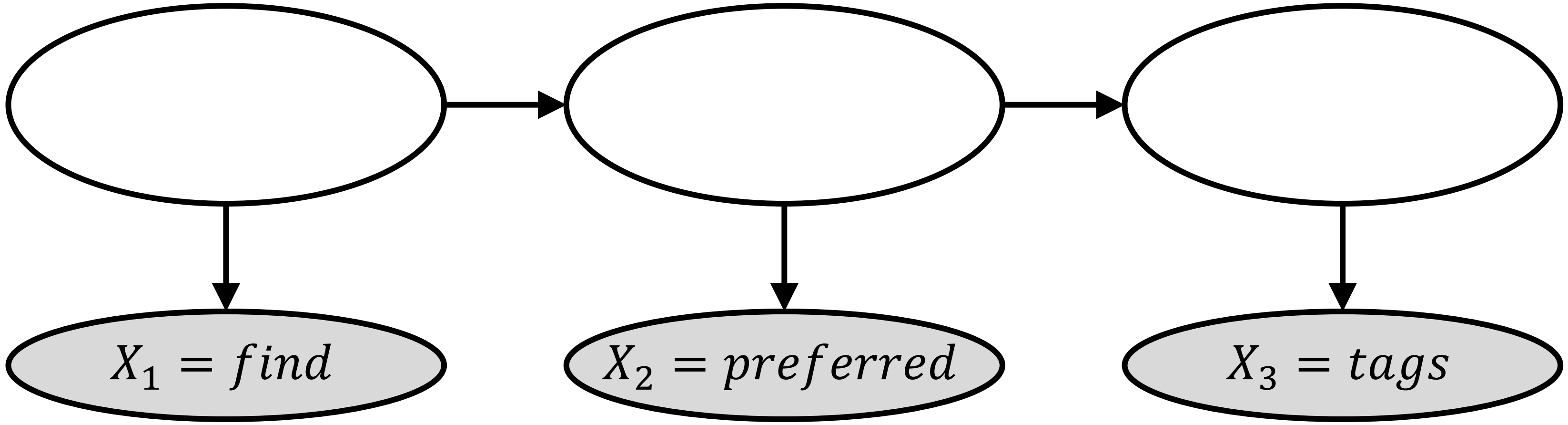
Could be verb or noun

Could be adjective or verb

Could be noun or verb

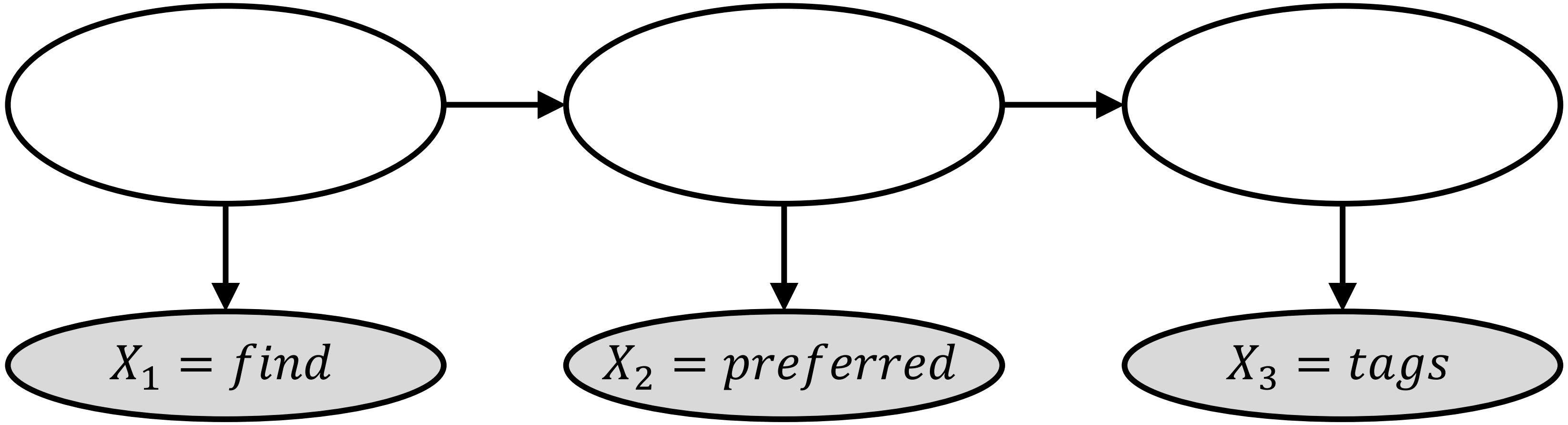
Filtering $P(Y_2|x_{1:2})$ vs Smoothing $P(Y_2|x_{1:3})$

Filtering (forward algorithm, then possibly argmax)



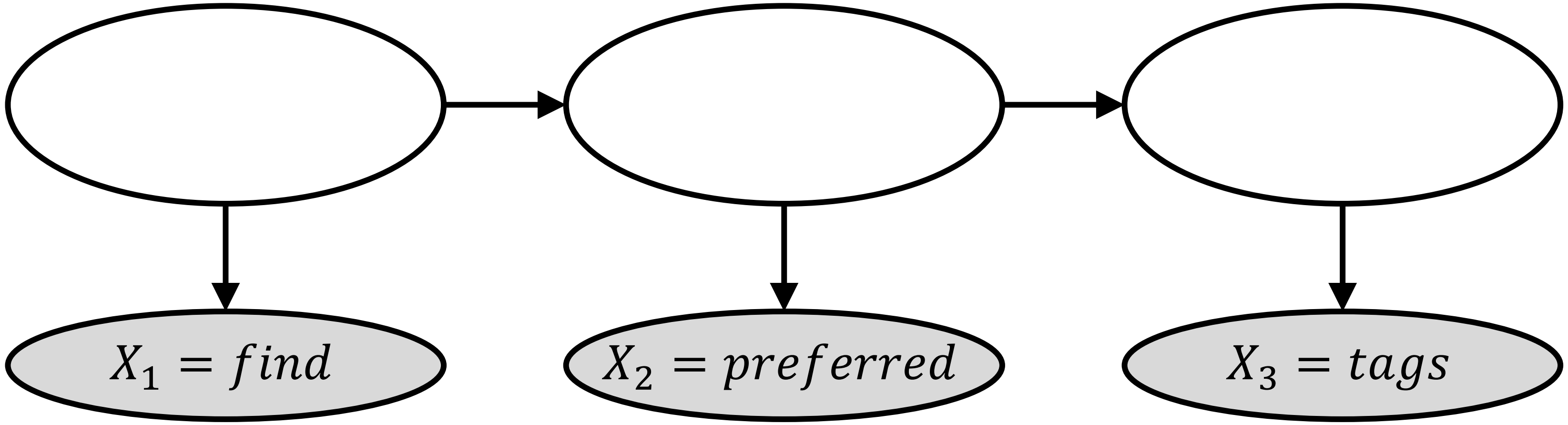
Filtering $P(Y_2 | x_{1:2})$ vs Smoothing $P(Y_2 | x_{1:3})$

Smoothing (forward-backward algorithm, then possibly argmax)



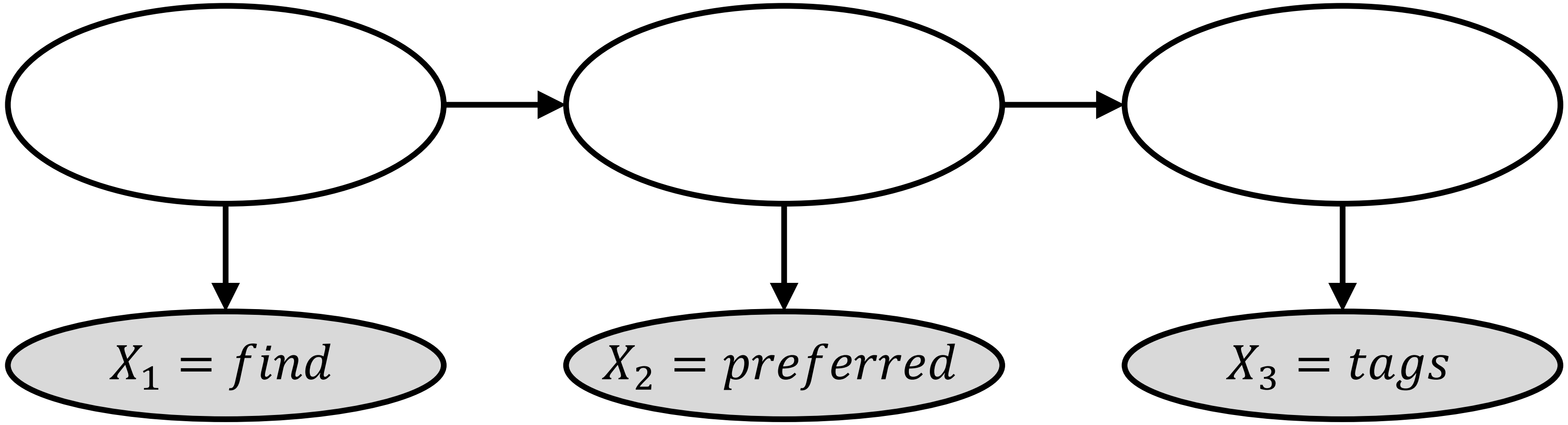
Smoothing $P(Y_2 | x_{1:3})$ vs Explanation $P(Y_{1:3} | x_{1:3})$

Smoothing (forward-backward algorithm, then argmax)



Smoothing $P(Y_2|x_{1:3})$ vs Explanation $P(Y_{1:3}|x_{1:3})$

Explanation (Viterbi algorithm)



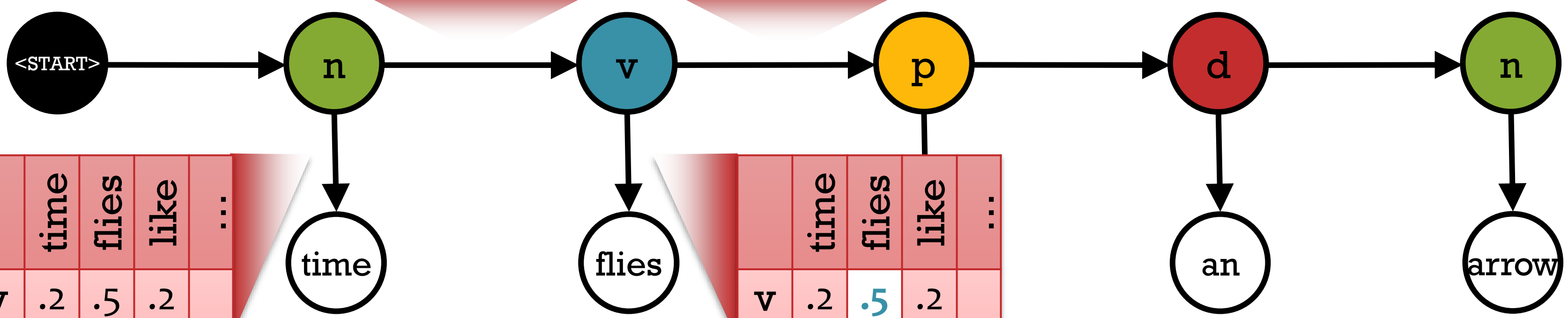
Hidden Markov Model

A Hidden Markov Model (HMM) provides a joint distribution over the the sentence/tags with an assumption of dependence between adjacent tags.

$$p(\text{n, v, p, d, n, time, flies, like, an, arrow}) = (.3 * .8 * .2 * .5 * \ldots)$$

	v	n	p	d
v	.1	.4	.2	.3
n	.8	.1	.1	0
p	.2	.3	.2	.3
d	.2	.8	0	0

	v	n	p	d
v	.1	.4	.2	.3
n	.8	.1	.1	0
p	.2	.3	.2	.3
d	.2	.8	0	0

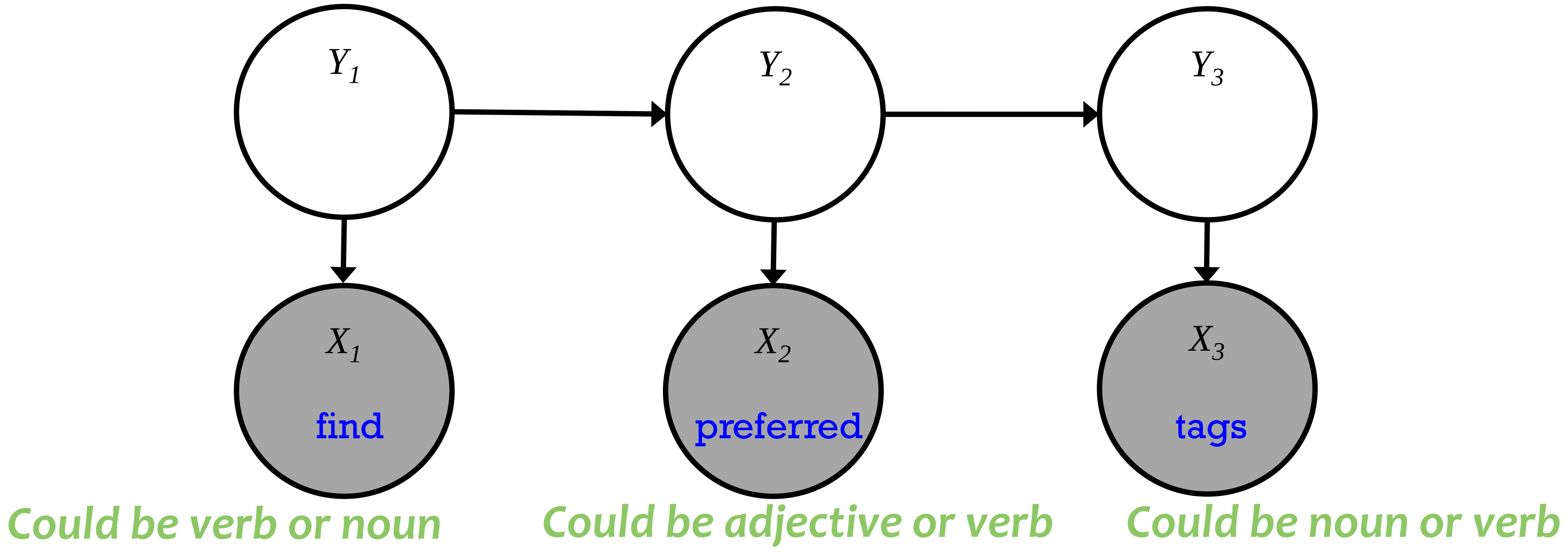


	time	flies	like	…
v	.2	.5	.2	
n	.3	.4	.2	
p	.1	.1	.3	
d	.1	.2	.1	

	time	flies	like	…
v	.2	.5	.2	
n	.3	.4	.2	
p	.1	.1	.3	
d	.1	.2	.1	

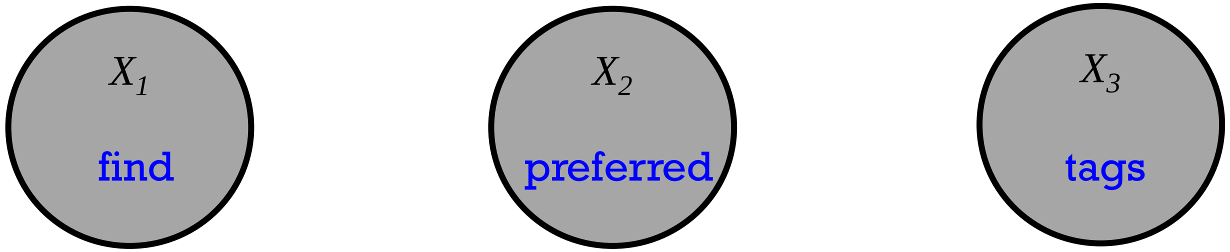
Slide credit: CMU MLD, Matt Gormley

Viterbi Algorithm: Most Probable Assignment



Slide credit: CMU MLD, Matt Gormley

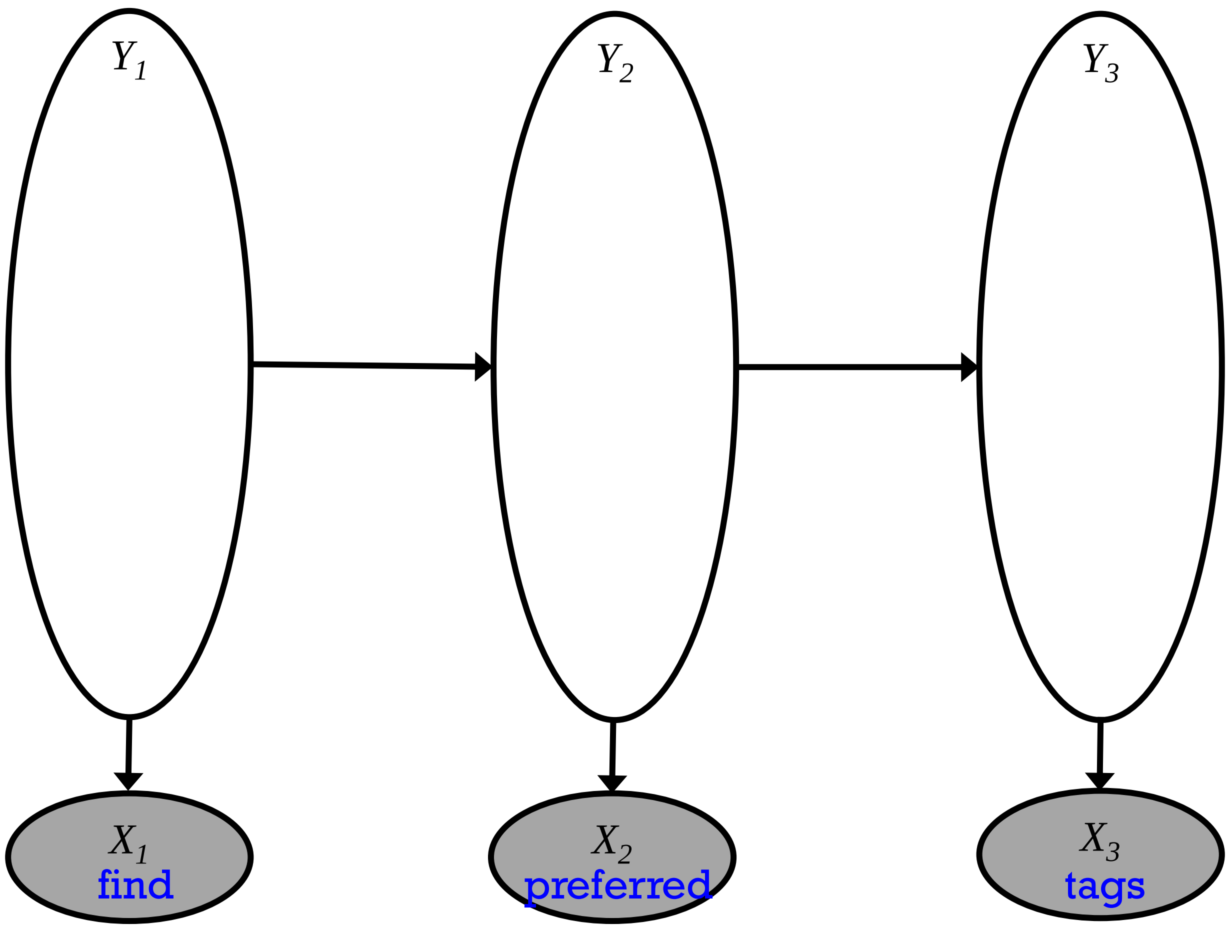
Viterbi Algorithm: Most Probable Assignment



Could be verb or noun *Could be adjective or verb* *Could be noun or verb*

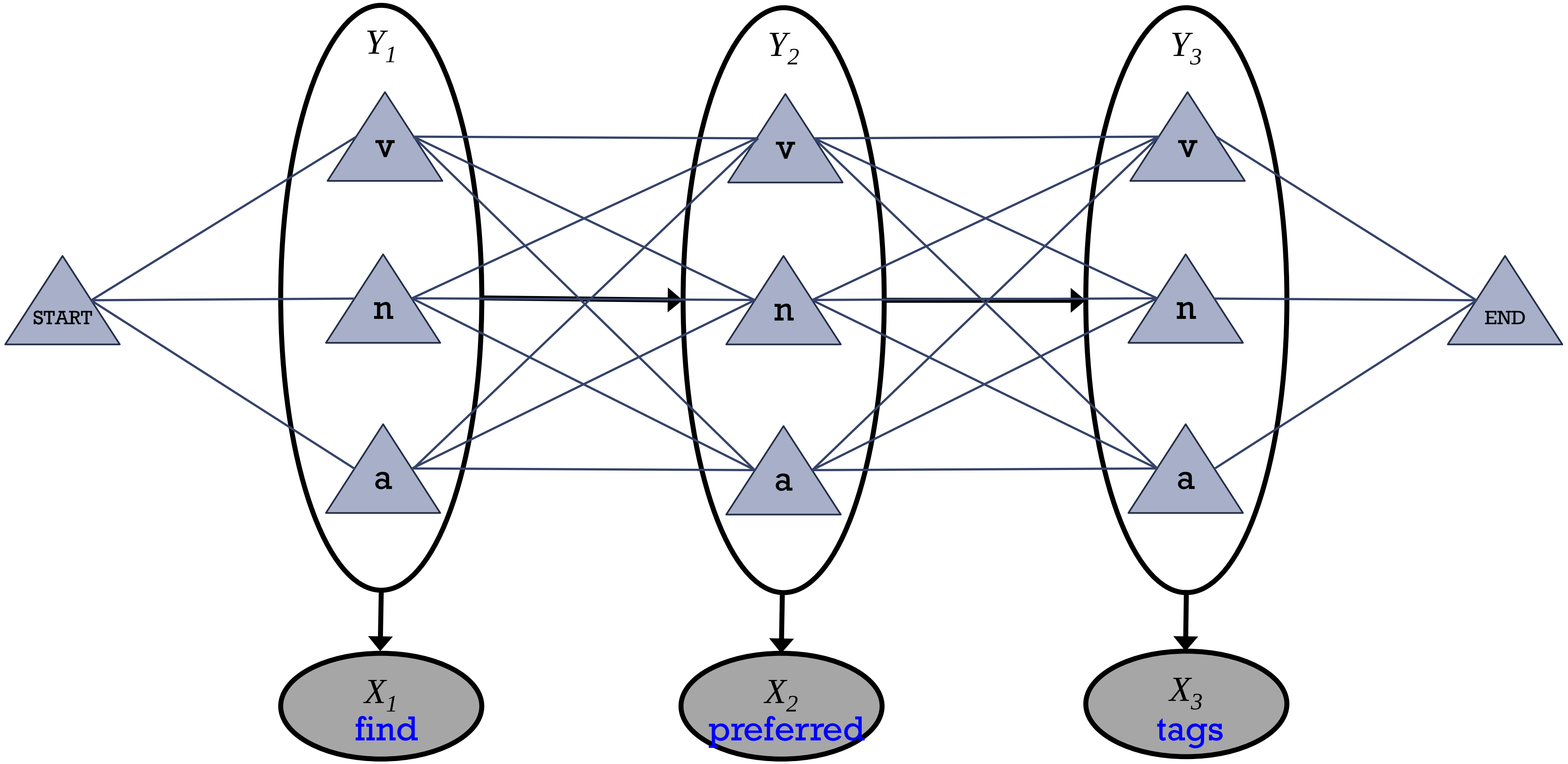
Slide credit: CMU MLD, Matt Gormley

Viterbi Algorithm: Most Probable Assignment



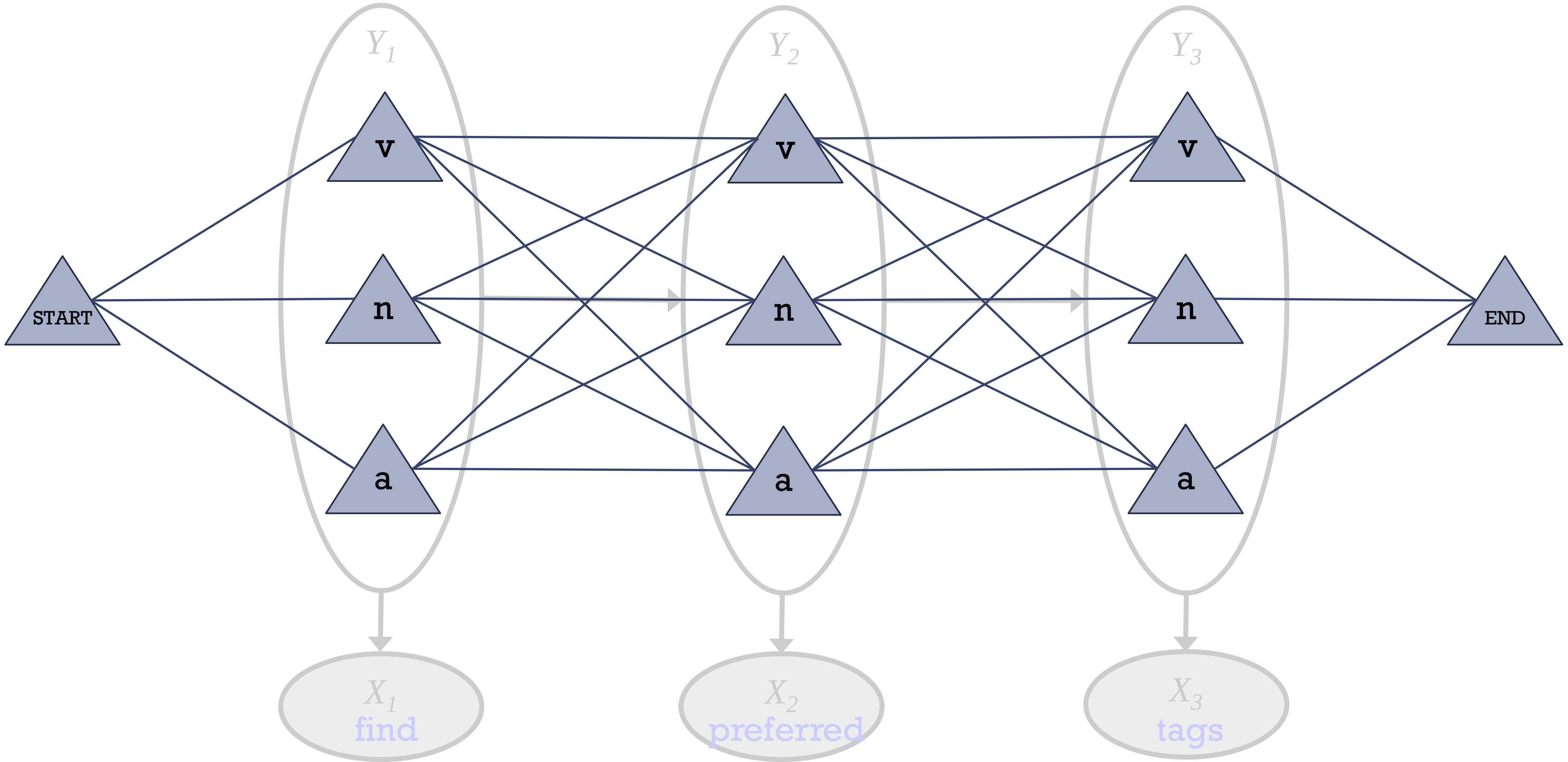
Slide credit: CMU MLD, Matt Gormley

Viterbi Algorithm: Most Probable Assignment



- Let's show the possible *values* for each variable

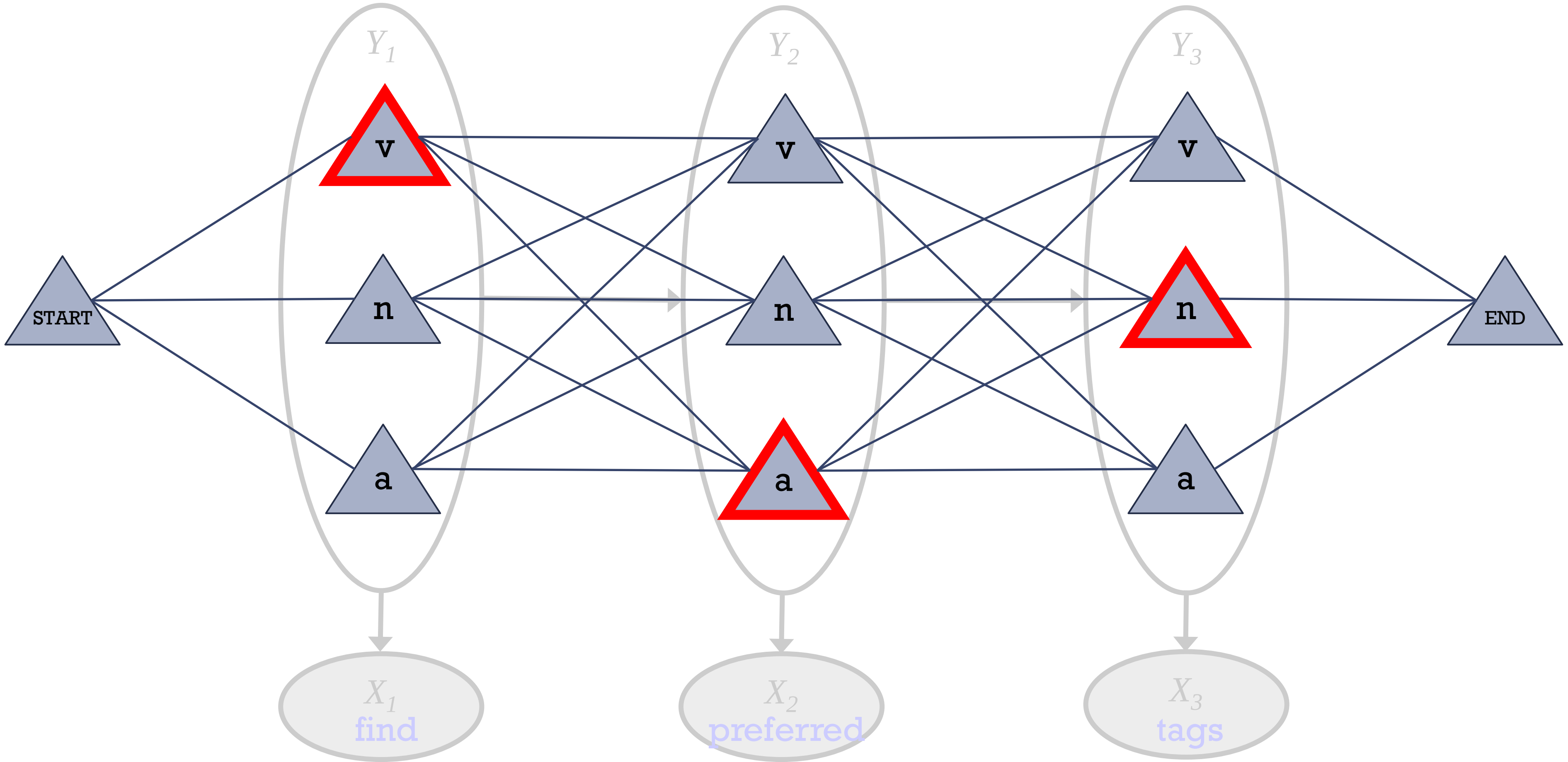
Viterbi Algorithm: Most Probable Assignment



- Let's show the possible *values* for each variable

Slide credit: CMU MLD, Matt Gormley

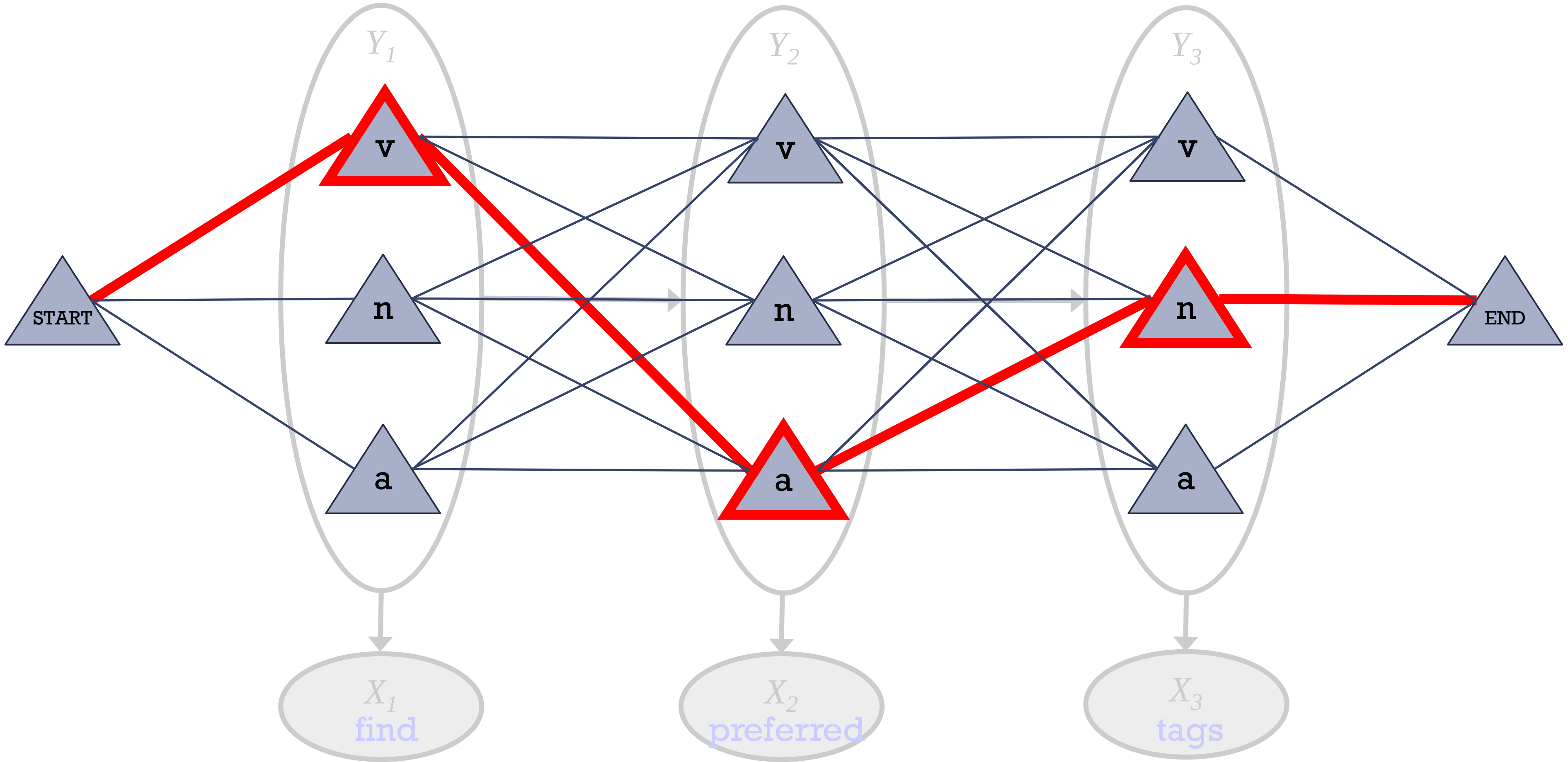
Viterbi Algorithm: Most Probable Assignment



- Let's show the possible *values* for each variable
- One possible assignment

Slide credit: CMU MLD, Matt Gormley

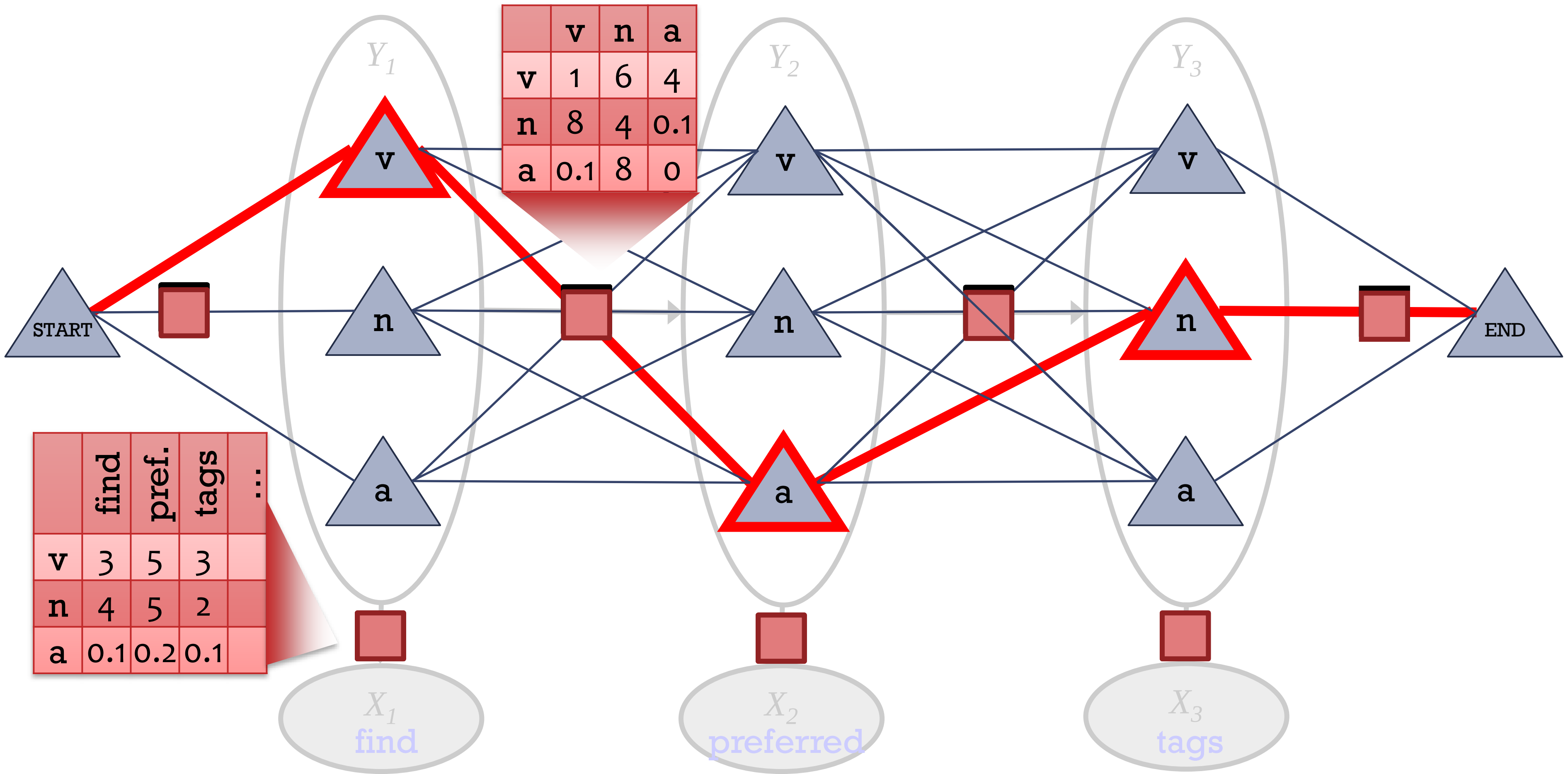
Viterbi Algorithm: Most Probable Assignment



- Let's show the possible *values* for each variable
- One possible assignment
- And what the 7 transition / emission factors think of it …

Slide credit: CMU MLD, Matt Gormley

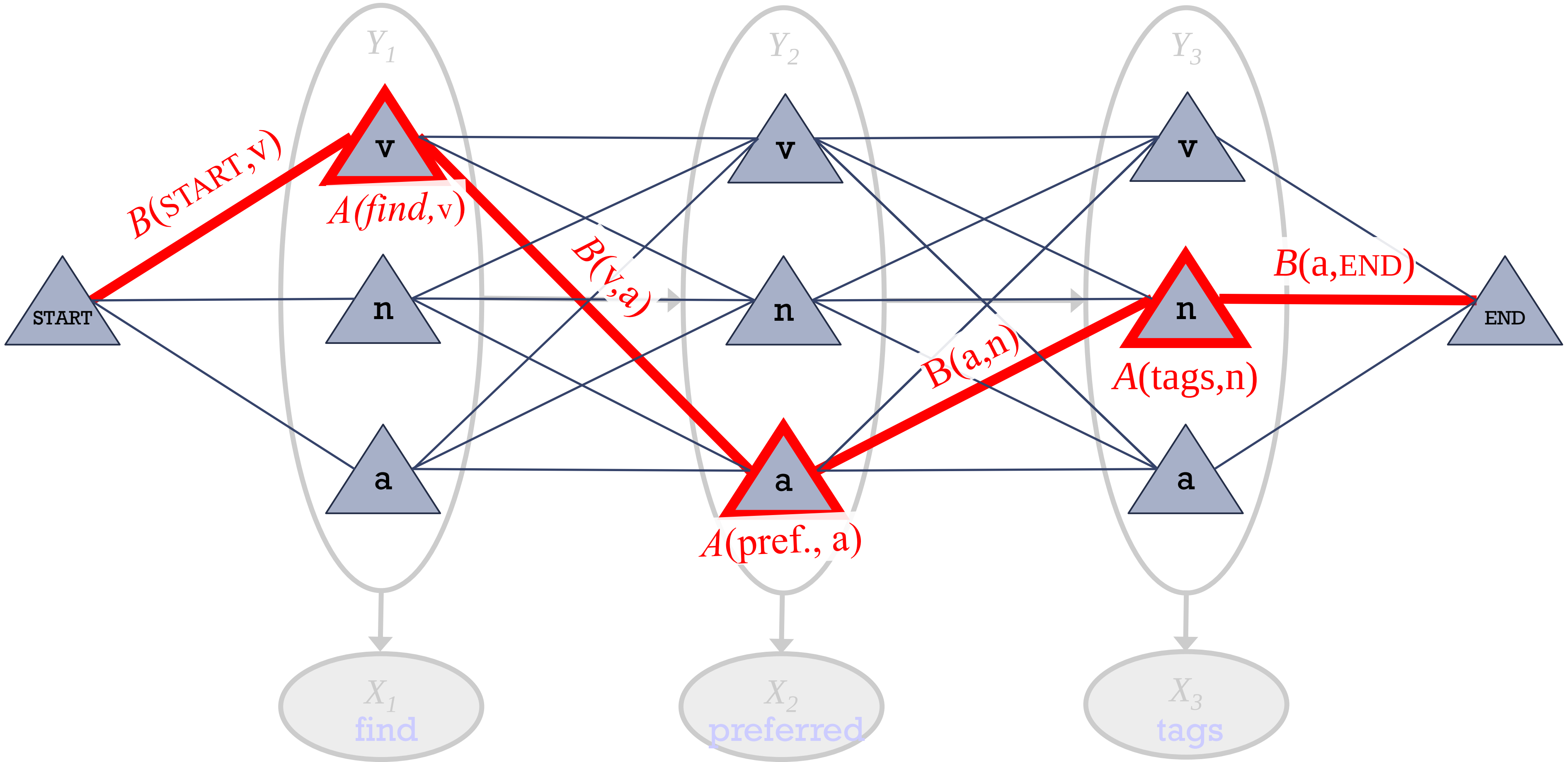
Viterbi Algorithm: Most Probable Assignment



- Let's show the possible *values* for each variable
- One possible assignment
- And what the 7 transition / emission factors think of it …

Slide credit: CMU MLD, Matt Gormley

Viterbi Algorithm: Most Probable Assignment

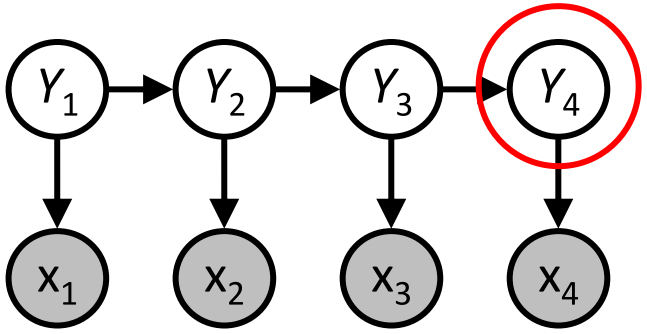


- So p(**v a n** | **x**) = (1/Z) times product of 7 numbers
- Numbers associated with edges and nodes of path
- Most probable assignment = **path with highest product**

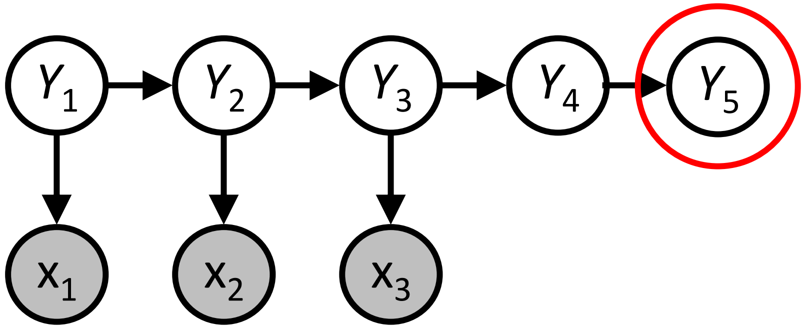
Slide credit: CMU MLD, Matt Gormley

HMM Queries

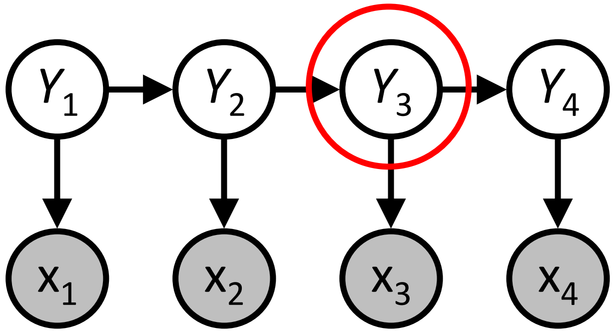
Filtering: $P(Y_t | x_{1:t})$



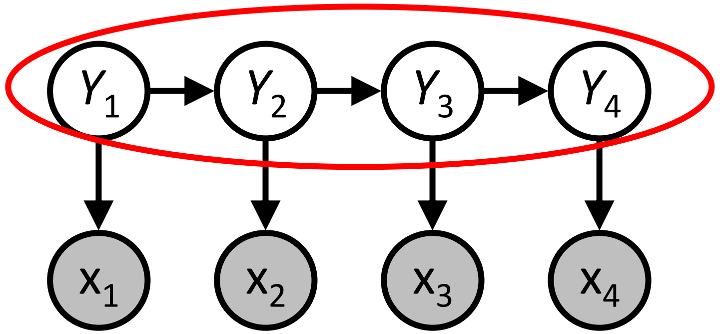
Prediction: $P(Y_{t+k} | x_{1:t})$



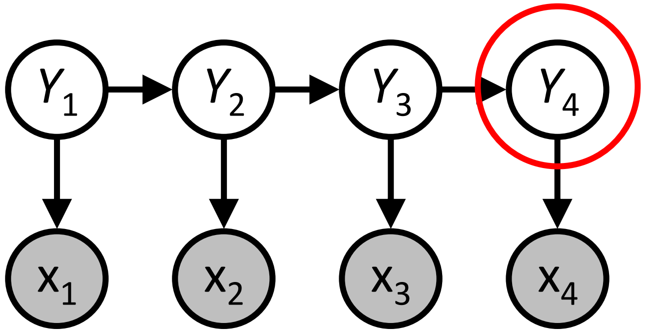
Smoothing: $P(Y_k | x_{1:t})$, $k<t$



Explanation: $P(Y_{1:t} | x_{1:t})$



Forward vs Viterbi

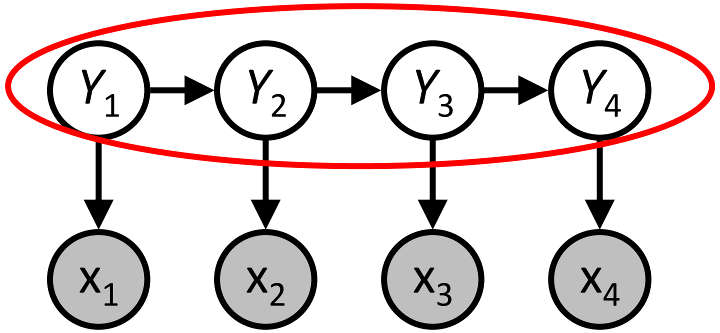


Forward

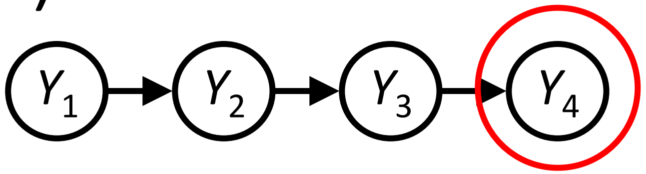
$$p(\, y_t \mid x_{1:t}\,) = \frac{1}{z} \sum_{y_1} \sum_{y_2} \dots \sum_{y_{t-1}} p(x_1, y_1, \dots, x_t, y_t)$$

Viterbi

$$\operatorname{argmax} p(\, y_{1:t} \mid x_{1:t}\,) = \underset{y_1, y_2, \dots, y_t}{\operatorname{argmax}}\, p(x_1, y_1, \dots, x_t, y_t)$$



Forward vs Viterbi (Simple Markov Chain)

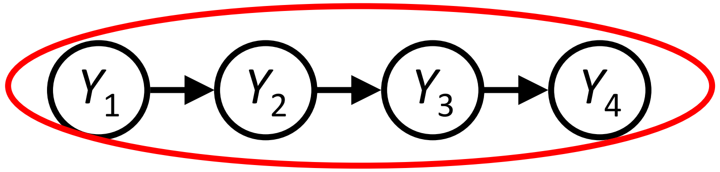


Forward

$$p(y_t) = \frac{1}{z} \sum_{y_1} \sum_{y_2} \dots \sum_{y_{t-1}} p(y_1, \dots, y_t)$$

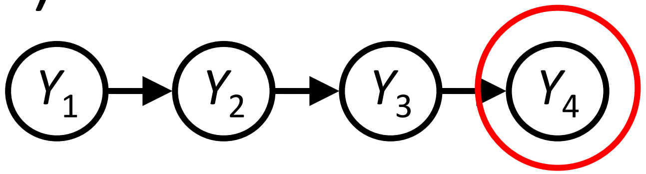
Viterbi

$$\underset{y_1, y_2, \dots, y_t}{\operatorname{argmax}}\ p(y_{1:t}) = \underset{y_1, y_2, \dots, y_t}{\operatorname{argmax}}\ p(y_1, \dots, y_t)$$



Forward vs Viterbi (Simple Markov Chain)

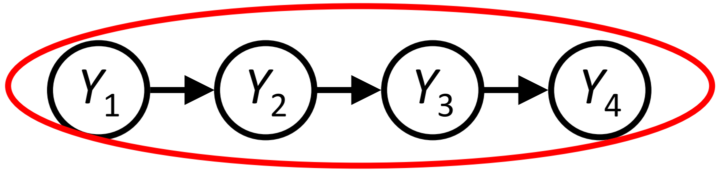
Forward



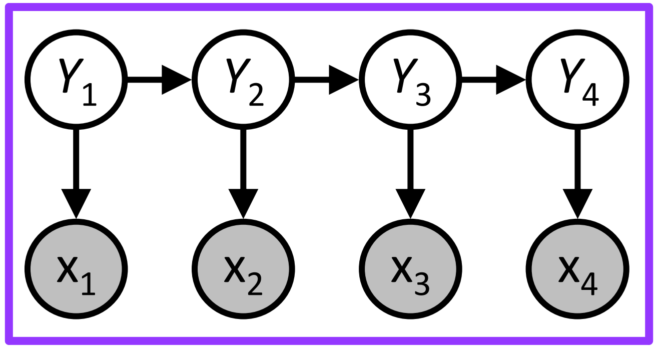
$$p(y_t) = \frac{1}{z}\sum_{y_{t-1}} p(\, y_t \mid y_{t-1}\,) \dots \sum_{y_1} p(\, y_2 \mid y_1\,) p(y_1)$$

Viterbi

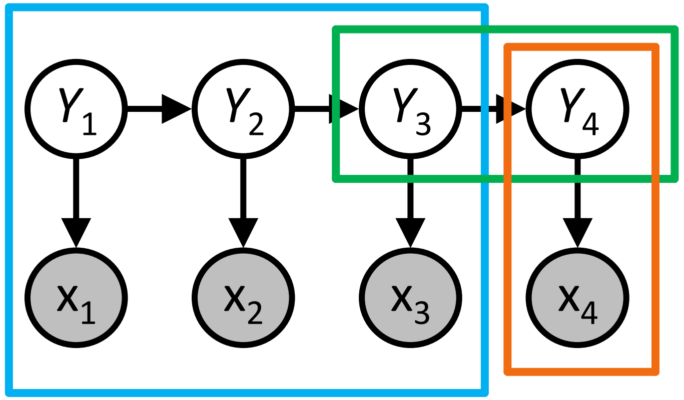
$$\max_{y_1, y_2, \dots, y_t} p(y_{1:t}) = \max_{y_t} \max_{y_{t-1}} p(\, y_t \mid y_{t-1}\,) \dots \max_{y_1} p(\, y_2 \mid y_1\,)\, p(y_1)$$



Viterbi Algorithm



Define:
$$\omega_t(k) = \max_{y_1,\ldots,y_{t-1}} P(x_1,\ldots,x_t,y_1,\ldots,y_{t-1},Y_t = k)$$
$$b_t(k) = \operatorname*{argmax}_{y_1,\ldots,y_{t-1}} P(x_1,\ldots,x_t,y_1,\ldots,y_{t-1},Y_t = k)$$



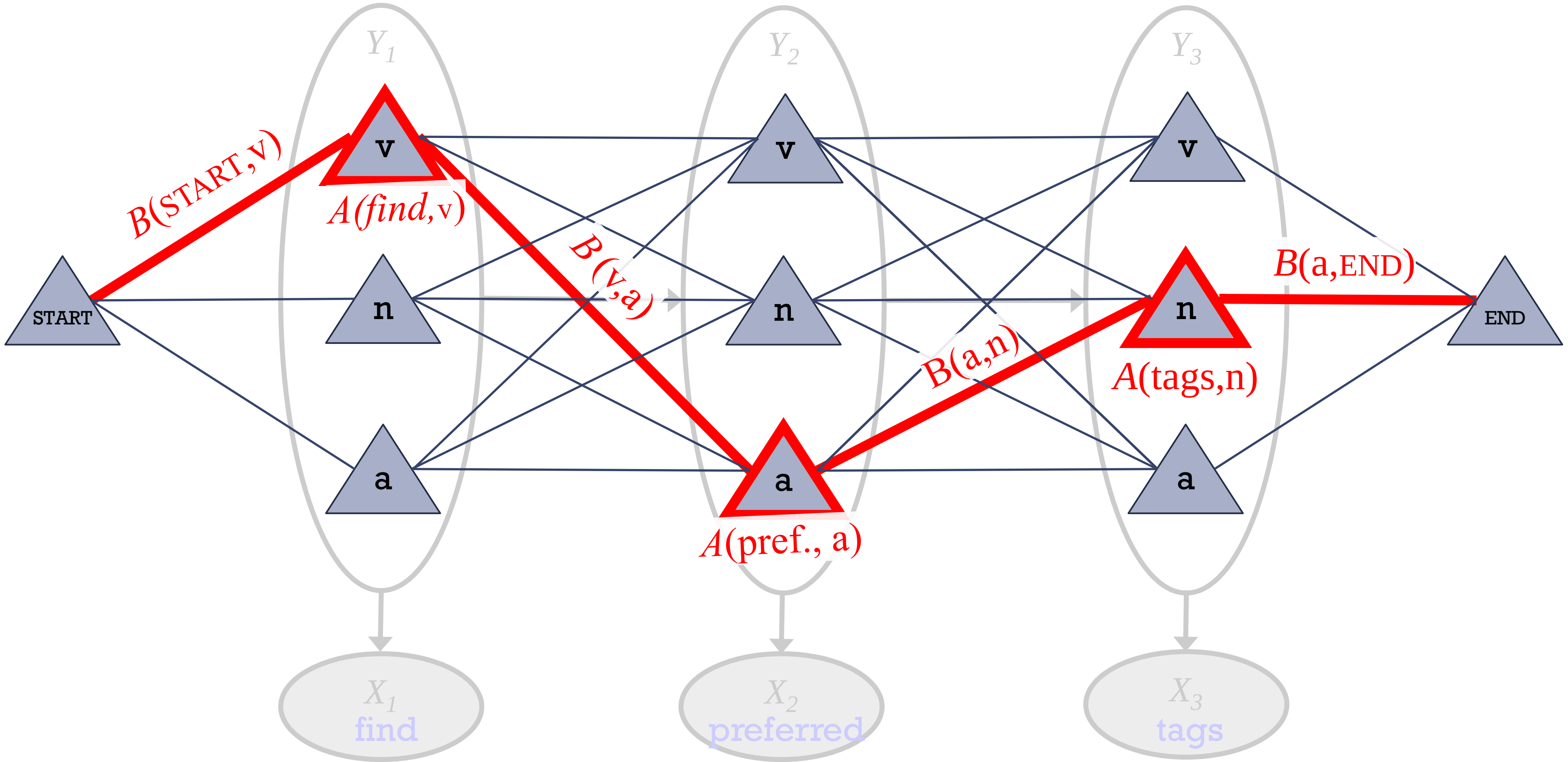
Assume: $y_0 = START$

1. Initialize $\omega_0(START) = 1,\ \omega_0(k) = 0\ \ \forall\, k \neq START$
2. For t = 1…T

   For k = 1…K
$$\omega_t(k) = \max_{j\in\{1,\ldots,K\}} P(\, x_t \mid Y_t = k \,)\, P(\, Y_t = k \mid Y_{t-1} = j \,)\, \omega_{t-1}(j)$$
$$b_t(k) = \operatorname*{argmax}_{j\in\{1,\ldots,K\}} P(\, x_t \mid Y_t = k \,)\, \mathrm{P}(\, Y_t = k \mid Y_{t-1} = j \,)\, \omega_{t-1}(j)$$
3. Compute most probable assignment: $\hat{y}_t = b_{t+1}(END)$

   For t = T-1, … 1
$$\hat{y}_t = b_{t+1}(\hat{y}_{t+1})$$

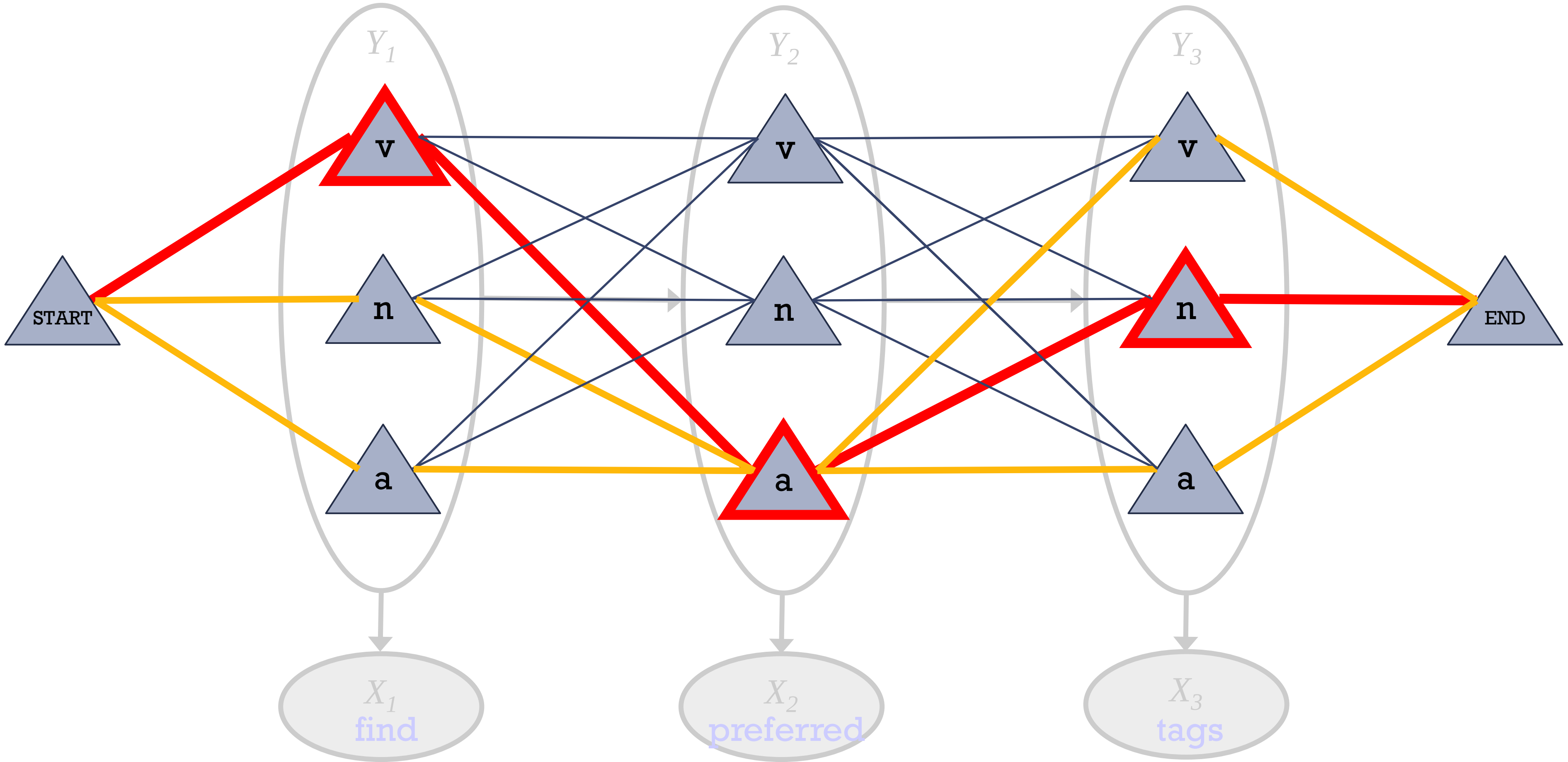
Viterbi Algorithm: Most Probable Assignment

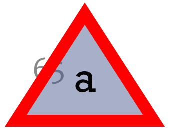


- So p(**v a n** | **x**) = (1/Z) times product weight of one path

Slide credit: CMU MLD, Matt Gormley

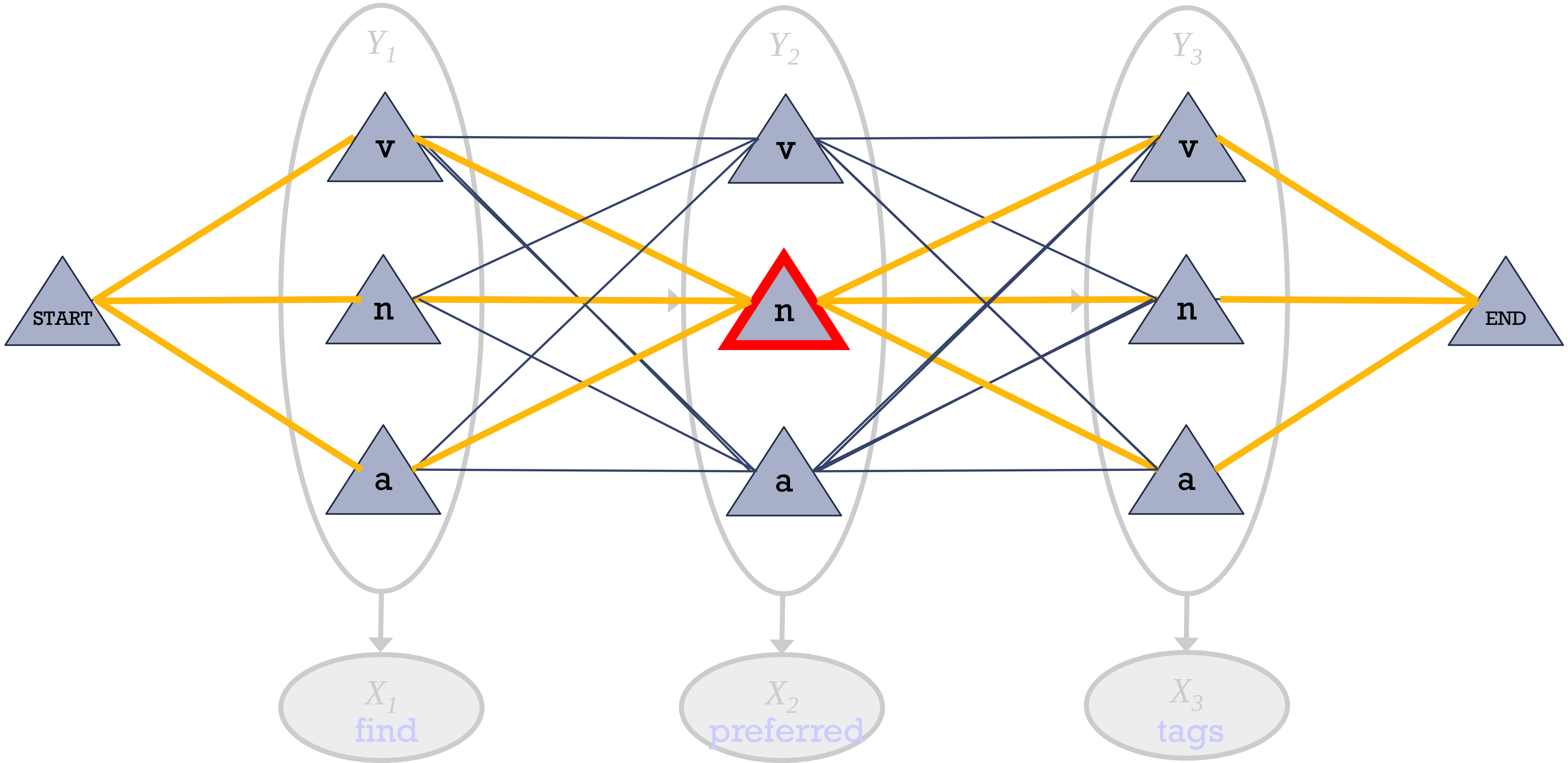
Forward-Backward Algorithm: p(y_t | $\mathbf{x}$)



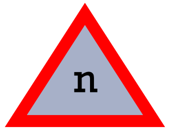
- So p(**v a n** | $\mathbf{x}$) = (1/Z) times product weight of one path
- Probability p(Y_2 = a | $\mathbf{x}$)
  = (1/Z) times total weight of *all* paths through 

Slide credit: CMU MLD, Matt Gormley

Forward-Backward Algorithm: p(y_t | $\mathbf{x}$)

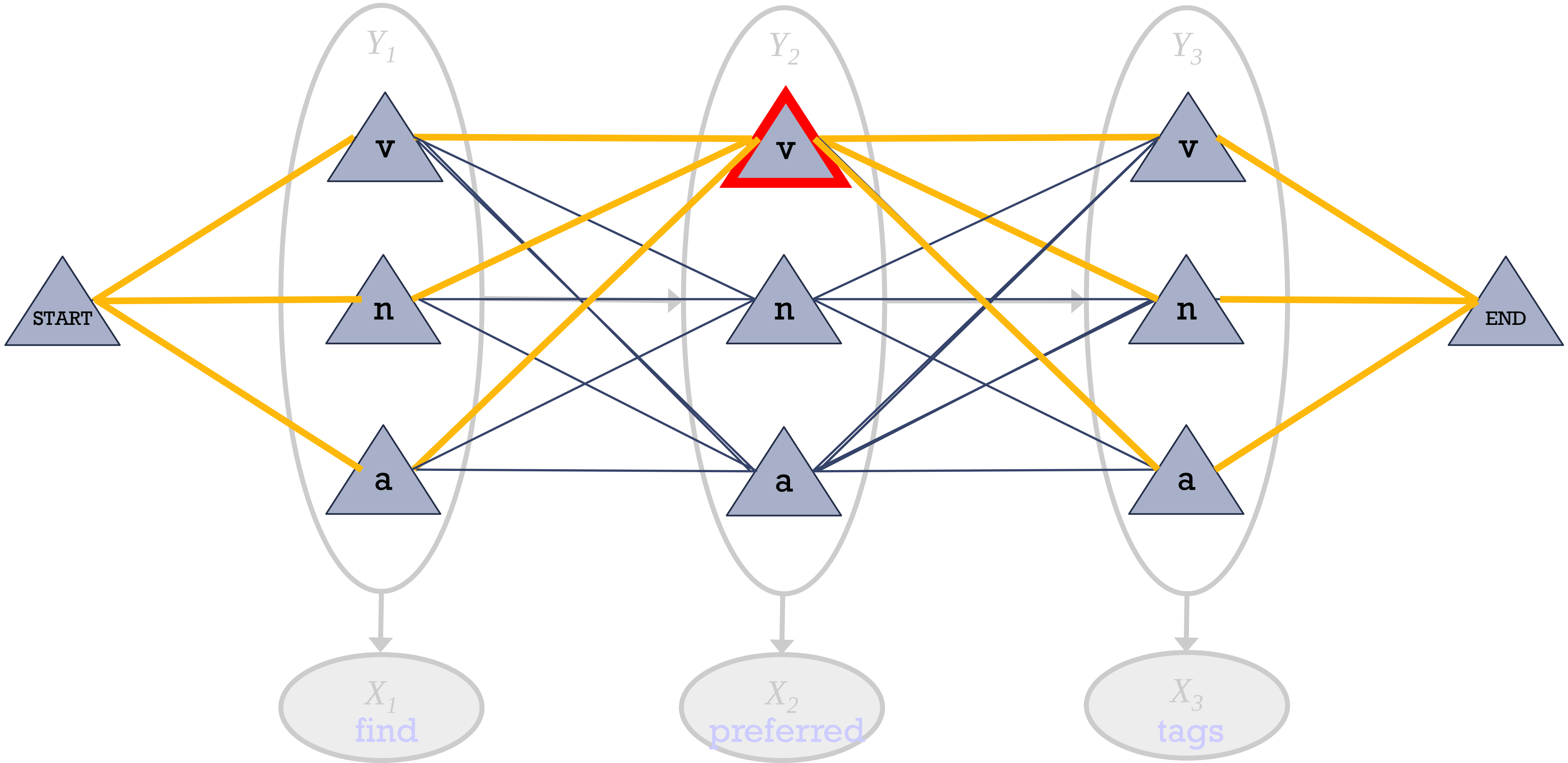


- So p(**v a n** | **x**) = (1/Z) times product weight of one path
- Probability p(Y_2 = n | **x**)
  = (1/Z) times total weight of *all* paths through



Slide credit: CMU MLD, Matt Gormley

Forward-Backward Algorithm: p(y_t | $\mathbf{x}$)



- So p(**v a n** | **x**) = (1/Z) times product weight of one path
- Probability p(Y_2 = v | **x**)
  = (1/Z) times total weight of *all* paths through v

Slide credit: CMU MLD, Matt Gormley

Inference in HMMs

What is the **computational complexity** of inference for HMMs?

- The **naïve** (brute force) computations for *Filtering*, *Smoothing*, and *Explanation* take **exponential time**, $O(K^T)$

- The **forward-backward** algorithm and **Viterbi** algorithm run in **polynomial time**, $O(T*K^2)$
  - Thanks to dynamic programming!

Learning Objectives

Hidden Markov Models

You should be able to…

1. Show that structured prediction problems yield high-computation inference problems
2. Define the first order Markov assumption
3. Draw a Finite State Machine depicting a first order Markov assumption
4. Derive the MLE parameters of an HMM
5. Define the key queries for an HMM: filtering, prediction, smoothing, explanation
6. Derive a dynamic programming algorithm for a key queries of an HMM
7. Interpret the forward-backward algorithm as a message passing algorithm
8. Implement supervised learning for an HMM
9. Implement the forward-backward algorithm for an HMM
10. Implement the Viterbi algorithm for an HMM