

# Lecture 4: Sequence Models I

Alan Ritter

(many slides from Greg Durrett, Dan Klein, Vivek Srikumar, Chris Manning, Yoav Artzi)

# This Lecture

---

- ▶ Sequence modeling
- ▶ HMMs for POS tagging
- ▶ HMM parameter estimation
- ▶ Viterbi, forward-backward

# Linguistic Structures

---

- ▶ Language is tree-structured

# Linguistic Structures

---

- ▶ Language is tree-structured

*I ate the spaghetti with chopsticks*

# Linguistic Structures

---

- ▶ Language is tree-structured

*I ate the spaghetti with chopsticks*

*I ate the spaghetti with meatballs*

# Linguistic Structures

---

- ▶ Language is tree-structured



# Linguistic Structures

---

- ▶ Language is tree-structured



# Linguistic Structures

---

- ▶ Language is tree-structured



- ▶ Understanding syntax fundamentally requires trees – the sentences have the same shallow analysis

# Linguistic Structures

---

- ▶ Language is tree-structured



- ▶ Understanding syntax fundamentally requires trees – the sentences have the same shallow analysis

PRP VBZ DT NN IN NNS  
*I ate the spaghetti with chopsticks*

PRP VBZ DT NN IN NNS  
*I ate the spaghetti with meatballs*

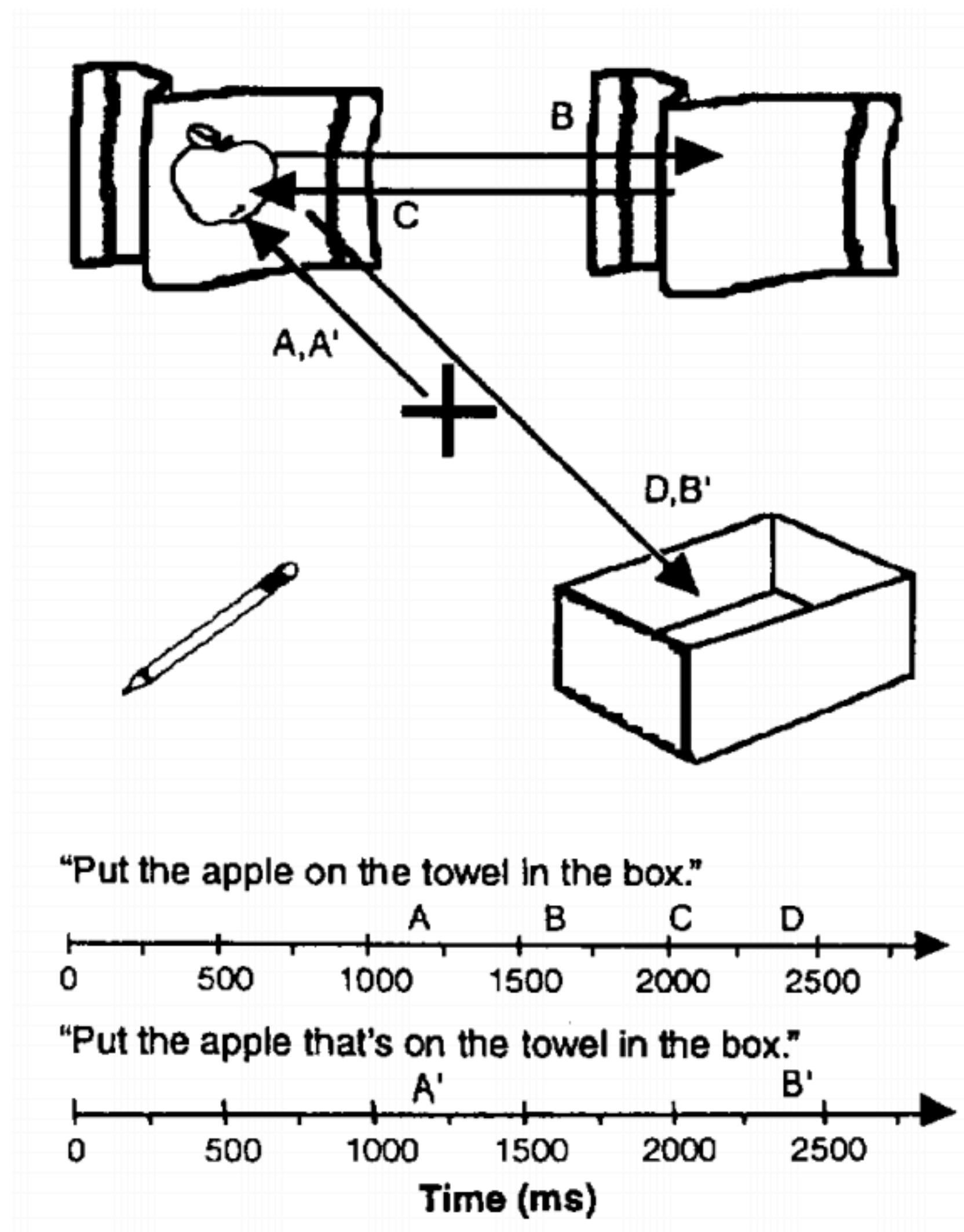
# Linguistic Structures

---

- ▶ Language is sequentially structured: interpreted in an online way

# Linguistic Structures

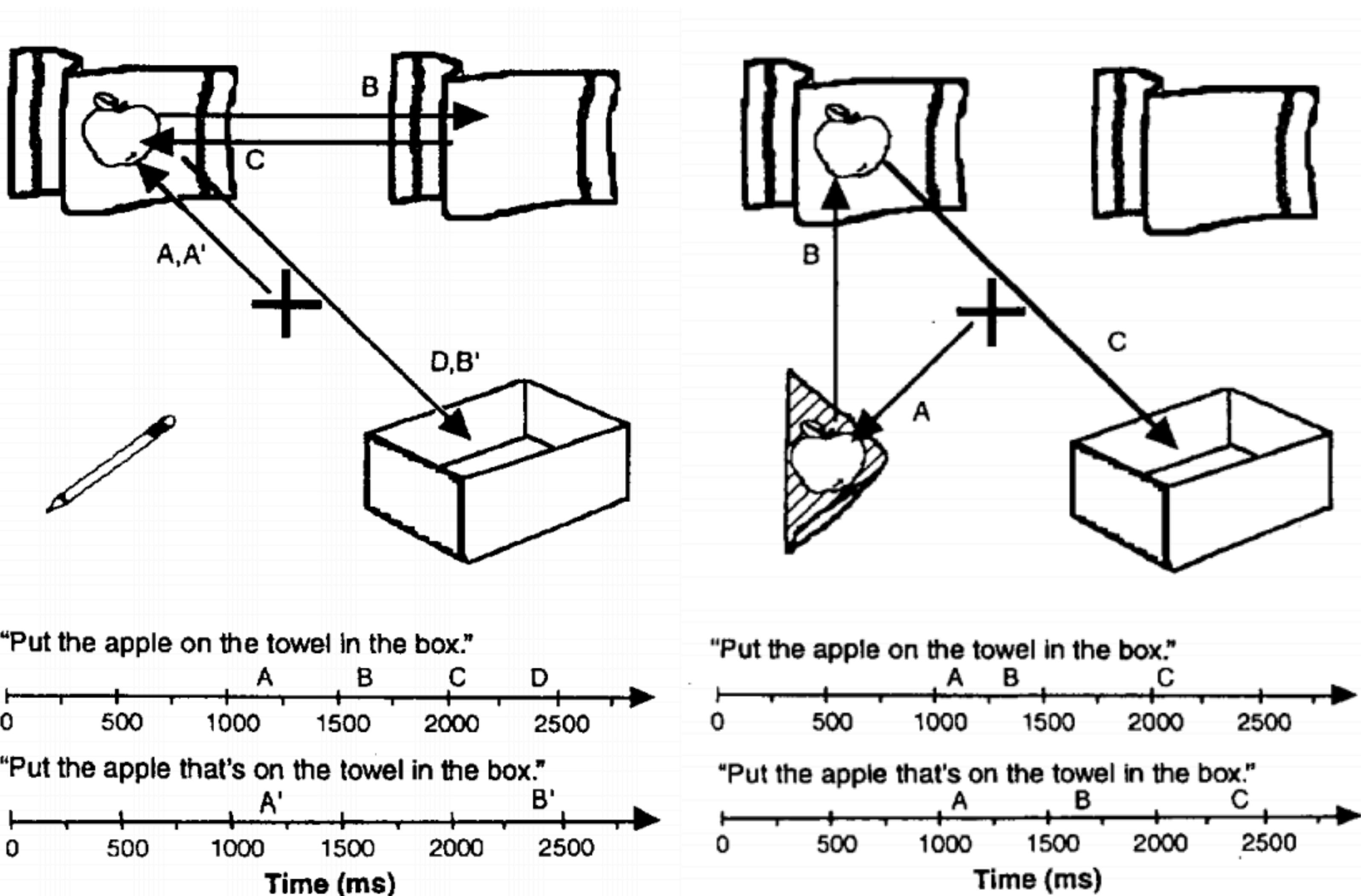
- ▶ Language is sequentially structured: interpreted in an online way



Tanenhaus et al. (1995)

# Linguistic Structures

- ▶ Language is sequentially structured: interpreted in an online way



Tanenhaus et al. (1995)

# POS Tagging

---

- ▶ What tags are out there?

*Ghana's ambassador should have set up the big meeting in DC yesterday .*

# POS Tagging

## Open class (lexical) words

### Nouns

#### Proper

*IBM*

*Italy*

#### Common

*cat / cats*

*snow*

### Verbs

#### Main

*see*

*registered*

### Adjectives

*yellow*

### Adverbs

*slowly*

### Numbers

*122,312*

*one*

*... more*

## Closed class (functional)

### Determiners

*the some*

### Conjunctions

*and or*

### Pronouns

*he its*

### Auxiliary

*can*

*had*

### Prepositions

*to with*

### Particles

*off up*

*... more*

# POS Tagging

---

# POS Tagging

---

*Fed raises interest rates 0.5 percent*

# POS Tagging

---

*Fed raises interest rates 0.5 percent*

I hereby  
increase interest  
rates 0.5%



# POS Tagging

---

VBD

VBN

NNP

*Fed raises interest rates 0.5 percent*

I hereby  
increase interest  
rates 0.5%



# POS Tagging

---

VBD

VBN VBZ

NNP NNS

*Fed raises interest rates 0.5 percent*

I hereby  
increase interest  
rates 0.5%



# POS Tagging

---

VBD	VB	
VBN	VBZ	VBP
NNP	NNS	NN

*Fed raises interest rates 0.5 percent*

I hereby  
increase interest  
rates 0.5%



# POS Tagging

---

VBD	VB		
VBN	VBZ	VBP	VBZ
NNP	NNS	NN	NNS

*Fed raises interest rates 0.5 percent*

I hereby  
increase interest  
rates 0.5%



# POS Tagging

---

VBD	VB				
VBN	VBZ	VBP	VBZ		
NNP	NNS	NN	NNS	CD	NN

*Fed raises interest rates 0.5 percent*

I hereby  
increase interest  
rates 0.5%



# POS Tagging

---

VBD            VB  
VBN **VBZ**    VBP    VBZ  
**NNP** NNS    NN      NNS CD NN

*Fed raises interest rates 0.5 percent*

I hereby  
increase interest  
rates 0.5%

VBD            VB  
VBN VBZ    **VBP**    VBZ  
**NNP** NNS    NN      NNS CD NN

*Fed raises interest rates 0.5 percent*



# POS Tagging

---

VBD            VB  
VBN **VBZ**    VBP    VBZ  
**NNP** NNS    NN      NNS CD    NN

*Fed raises interest rates 0.5 percent*

I hereby  
increase interest  
rates 0.5%



VBD            VB  
VBN VBZ    **VBP**    VBZ  
**NNP** NNS    NN      NNS CD    NN

*Fed raises interest rates 0.5 percent*

I'm 0.5% interested  
in the Fed's raises!



# POS Tagging

VBD            VB  
VBN **VBZ**    VBP    VBZ  
**NNP** NNS    NN      NNS CD NN

*Fed raises interest rates 0.5 percent*

I hereby  
increase interest  
rates 0.5%



VBD            VB  
VBN VBZ    **VBP**    VBZ  
**NNP** NNS    NN      NNS CD NN

*Fed raises interest rates 0.5 percent*

I'm 0.5% interested  
in the Fed's raises!



- ▶ Other paths are also plausible but even more semantically weird...

# POS Tagging

VBD	VB				
VBN	<b>VBZ</b>	VBP	VBZ		
<b>NNP</b>	NNS	<b>NN</b>	<b>NNS</b>	CD	NN

*Fed raises interest rates 0.5 percent*

I hereby  
increase interest  
rates 0.5%



VBD	VB				
VBN	VBZ	<b>VBP</b>	VBZ		
<b>NNP</b>	NNS	<b>NN</b>	<b>NNS</b>	CD	NN

*Fed raises interest rates 0.5 percent*

I'm 0.5% interested  
in the Fed's raises!



- ▶ Other paths are also plausible but even more semantically weird...
- ▶ What governs the correct choice? Word + context

# POS Tagging

VBD	VB				
VBN	Vbz	VBP	VBZ		
NNP	NNS	NN	NNS	CD	NN

*Fed raises interest rates 0.5 percent*

I hereby  
increase interest  
rates 0.5%



VBD	VB				
VBN	VBZ	VBP	VBZ		
NNP	NNS	NN	NNS	CD	NN

*Fed raises interest rates 0.5 percent*

I'm 0.5% interested  
in the Fed's raises!



- ▶ Other paths are also plausible but even more semantically weird...
- ▶ What governs the correct choice? Word + context
  - ▶ Word identity: most words have  $\leq 2$  tags, many have one (*percent, the*)
  - ▶ Context: nouns start sentences, nouns follow verbs, etc.

# POS Tagging

---

<b>CC</b>	conjunction, coordinating	and both but either or
<b>CD</b>	numeral, cardinal	mid-1890 nine-thirty 0.5 one
<b>DT</b>	determiner	a all an every no that the
<b>EX</b>	existential there	there
<b>FW</b>	foreign word	gemeinschaft hund ich jeux
<b>IN</b>	preposition or conjunction, subordinating	among whether out on by if
<b>JJ</b>	adjective or numeral, ordinal	third ill-mannered regrettable
<b>JJR</b>	adjective, comparative	braver cheaper taller
<b>JJS</b>	adjective, superlative	bravest cheapest tallest
<b>MD</b>	modal auxiliary	can may might will would
<b>NN</b>	noun, common, singular or mass	cabbage thermostat investment subhumanity
<b>NNP</b>	noun, proper, singular	Motown Cougar Yvette Liverpool
<b>NNPS</b>	noun, proper, plural	Americans Materials States
<b>NNS</b>	noun, common, plural	undergraduates bric-a-brac averages
<b>POS</b>	genitive marker	' s
<b>PRP</b>	pronoun, personal	hers himself it we them
<b>PRP\$</b>	pronoun, possessive	her his mine my our ours their thy your
<b>RB</b>	adverb	occasionally maddeningly adventurously
<b>RBR</b>	adverb, comparative	further gloomier heavier less-perfectly
<b>RBS</b>	adverb, superlative	best biggest nearest worst
<b>RP</b>	particle	aboard away back by on open through
<b>TO</b>	"to" as preposition or infinitive marker	to
<b>UH</b>	interjection	huh howdy uh whammo shucks heck
<b>VB</b>	verb, base form	ask bring fire see take
<b>VBD</b>	verb, past tense	pleaded swiped registered saw
<b>VBG</b>	verb, present participle or gerund	stirring focusing approaching erasing
<b>VBN</b>	verb, past participle	dilapidated imitated reunified unsettled
<b>VBP</b>	verb, present tense, not 3rd person singular	twist appear comprise mold postpone
<b>VBZ</b>	verb, present tense, 3rd person singular	bases reconstructs marks uses
<b>WDT</b>	WH-determiner	that what whatever which whichever
<b>WP</b>	WH-pronoun	that what whatever which who whom
<b>WP\$</b>	WH-pronoun, possessive	whose
<b>WRB</b>	Wh-adverb	however whenever where why

# What is this good for?

---

# What is this good for?

---

- ▶ Text-to-speech: *record, lead*

# What is this good for?

---

- ▶ Text-to-speech: *record, lead*
- ▶ Preprocessing step for syntactic parsers

# What is this good for?

---

- ▶ Text-to-speech: *record, lead*
- ▶ Preprocessing step for syntactic parsers
- ▶ Domain-independent disambiguation for other tasks

# What is this good for?

---

- ▶ Text-to-speech: *record, lead*
- ▶ Preprocessing step for syntactic parsers
- ▶ Domain-independent disambiguation for other tasks
- ▶ (Very) shallow information extraction

# Sequence Models

---

# Sequence Models

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$

# Sequence Models

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$
- ▶ POS tagging:  $x$  is a sequence of words,  $y$  is a sequence of tags

# Sequence Models

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$
- ▶ POS tagging:  $\mathbf{x}$  is a sequence of words,  $\mathbf{y}$  is a sequence of tags
- ▶ Today: generative models  $P(\mathbf{x}, \mathbf{y})$ ; discriminative models next time

# Hidden Markov Models

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$

# Hidden Markov Models

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$
- ▶ Model the sequence of  $y$  as a Markov process (dynamics model)

# Hidden Markov Models

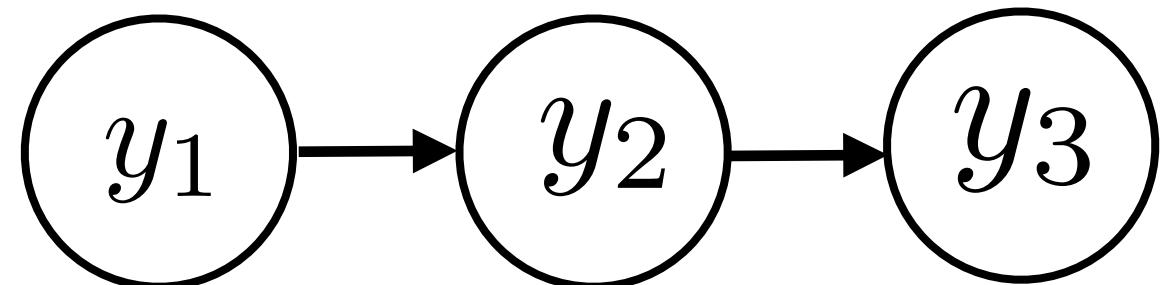
---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$
- ▶ Model the sequence of  $y$  as a Markov process (dynamics model)
- ▶ Markov property: future is conditionally independent of the past given the present

# Hidden Markov Models

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$
- ▶ Model the sequence of  $y$  as a Markov process (dynamics model)
- ▶ Markov property: future is conditionally independent of the past given the present

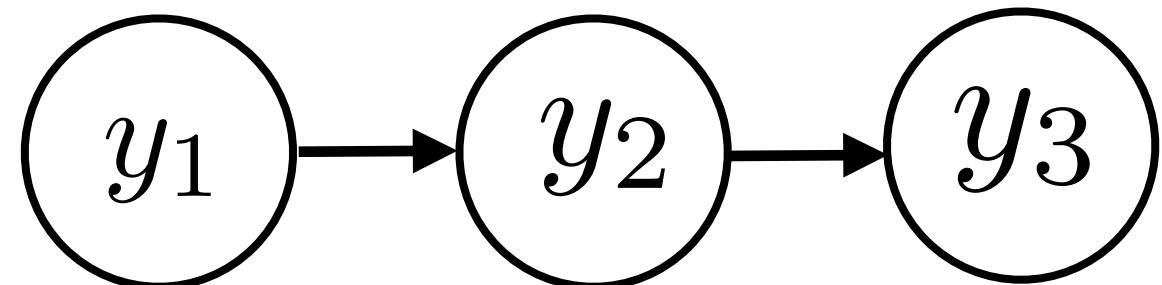


$$P(y_3|y_1, y_2) = P(y_3|y_2)$$

# Hidden Markov Models

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$
- ▶ Model the sequence of  $y$  as a Markov process (dynamics model)
- ▶ Markov property: future is conditionally independent of the past given the present



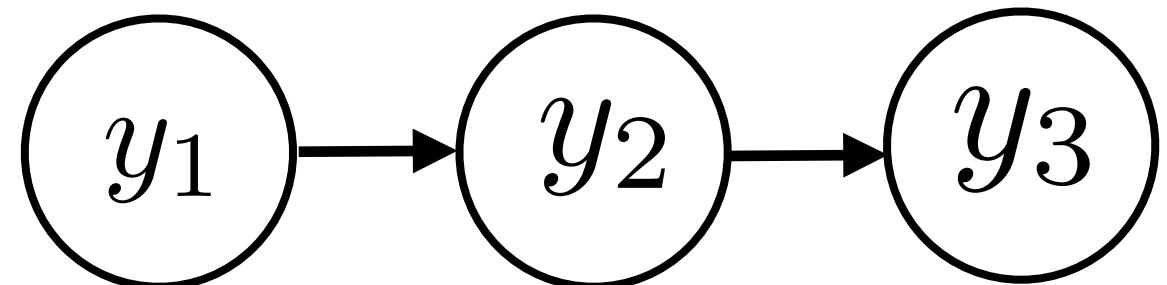
$$P(y_3|y_1, y_2) = P(y_3|y_2)$$

- ▶ Lots of mathematical theory about how Markov chains behave

# Hidden Markov Models

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$
- ▶ Model the sequence of  $y$  as a Markov process (dynamics model)
- ▶ Markov property: future is conditionally independent of the past given the present



$$P(y_3|y_1, y_2) = P(y_3|y_2)$$

- ▶ Lots of mathematical theory about how Markov chains behave
- ▶ If  $y$  are tags, this roughly corresponds to assuming that the next tag only depends on the current tag, not anything before

# Hidden Markov Models

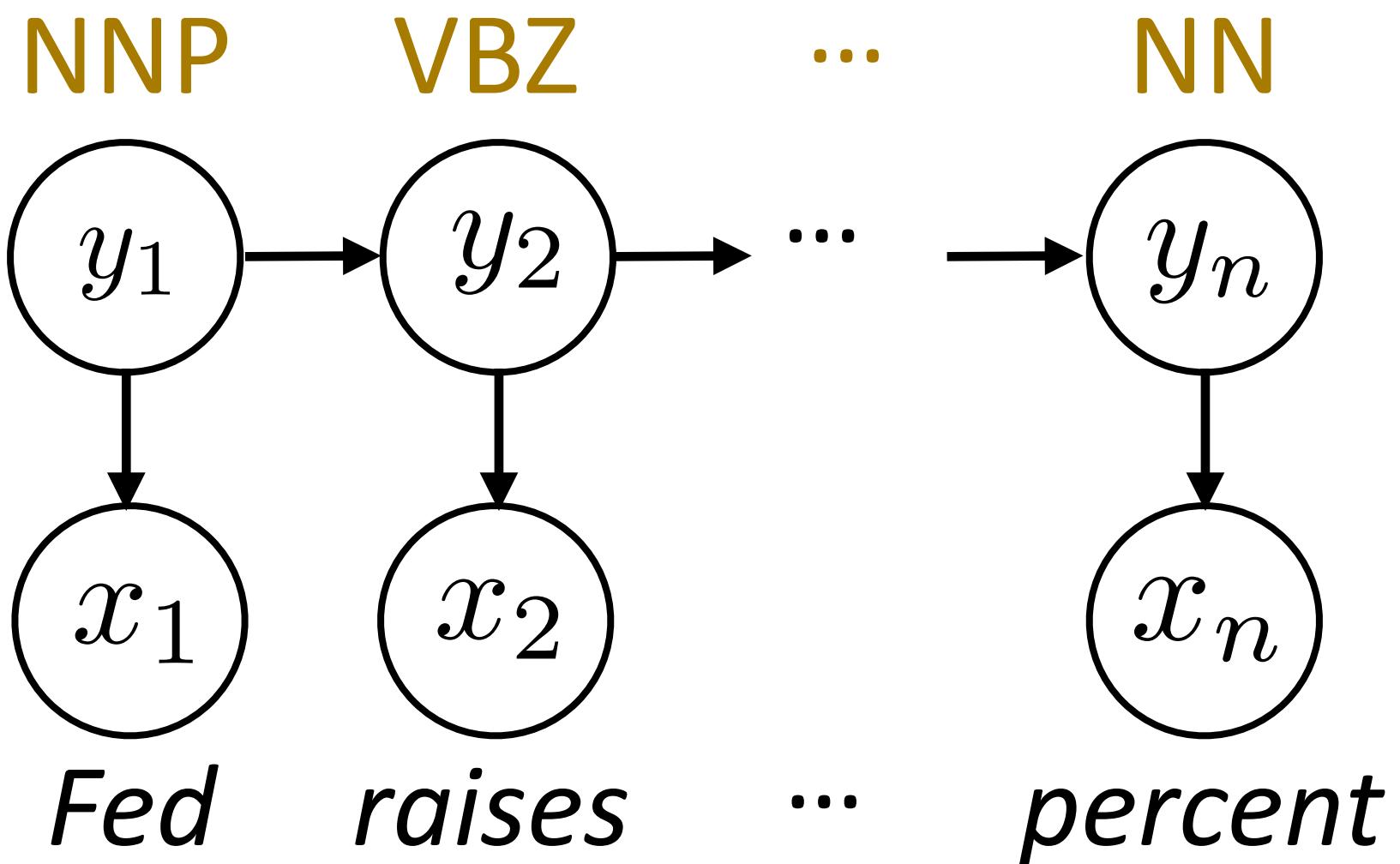
---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$

# Hidden Markov Models

---

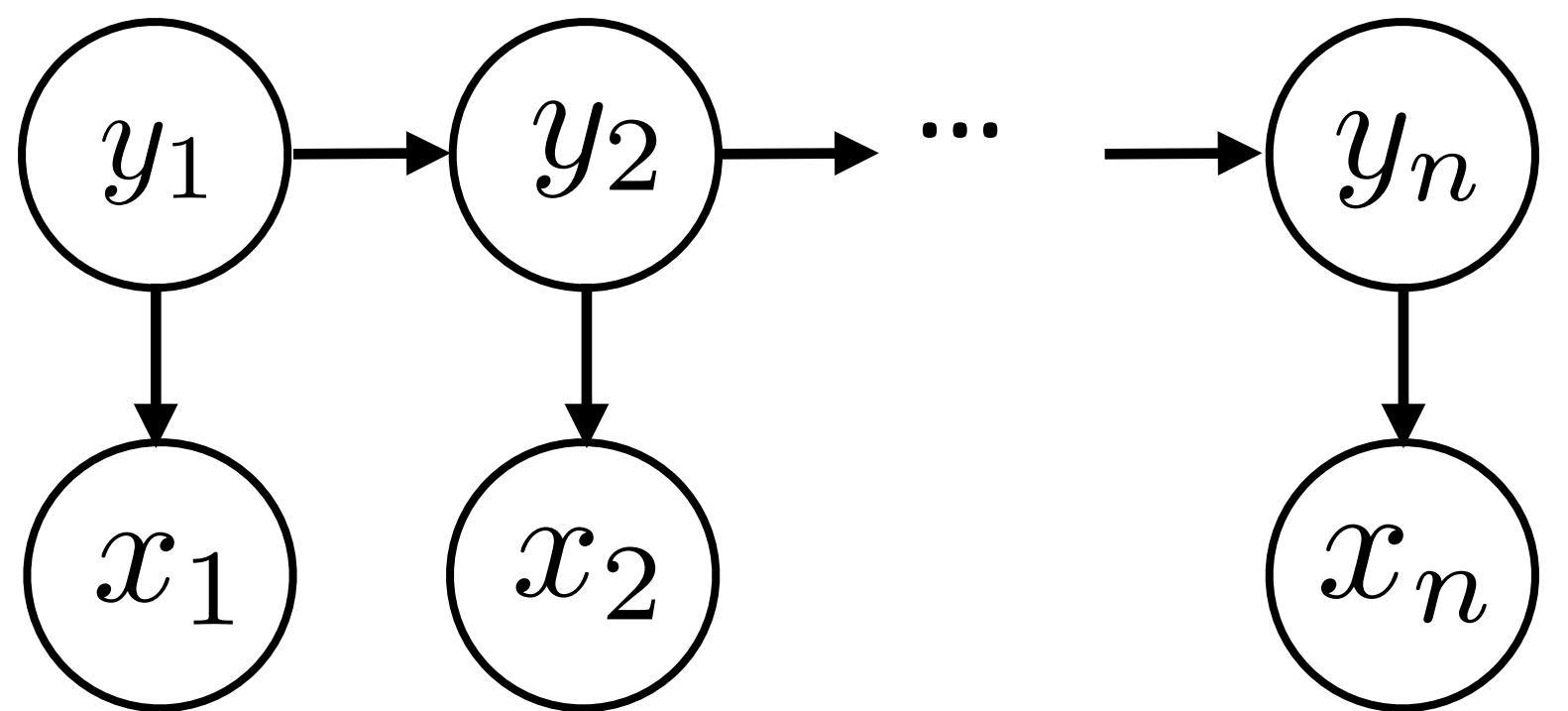
- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$       Output  $\mathbf{y} = (y_1, \dots, y_n)$



# Hidden Markov Models

---

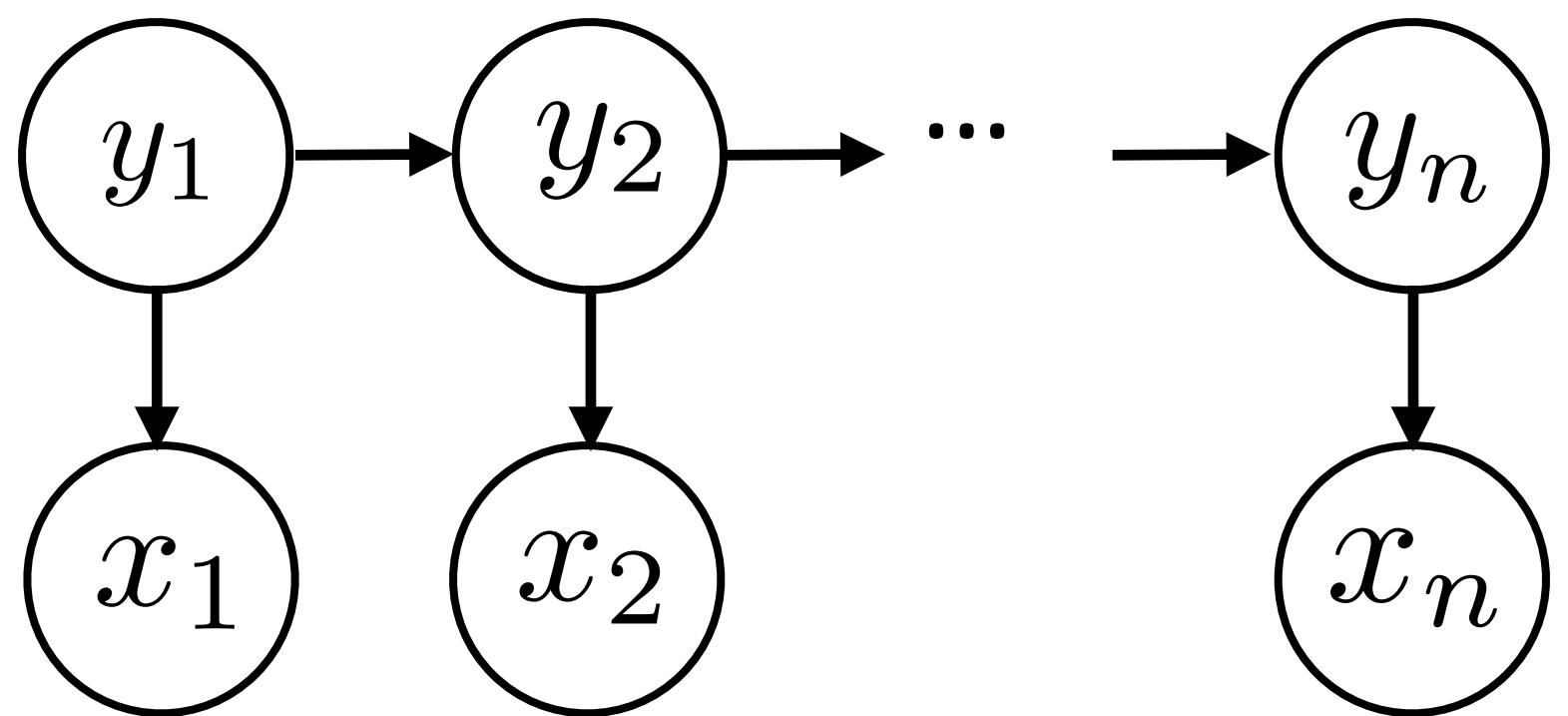
- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$



# Hidden Markov Models

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$

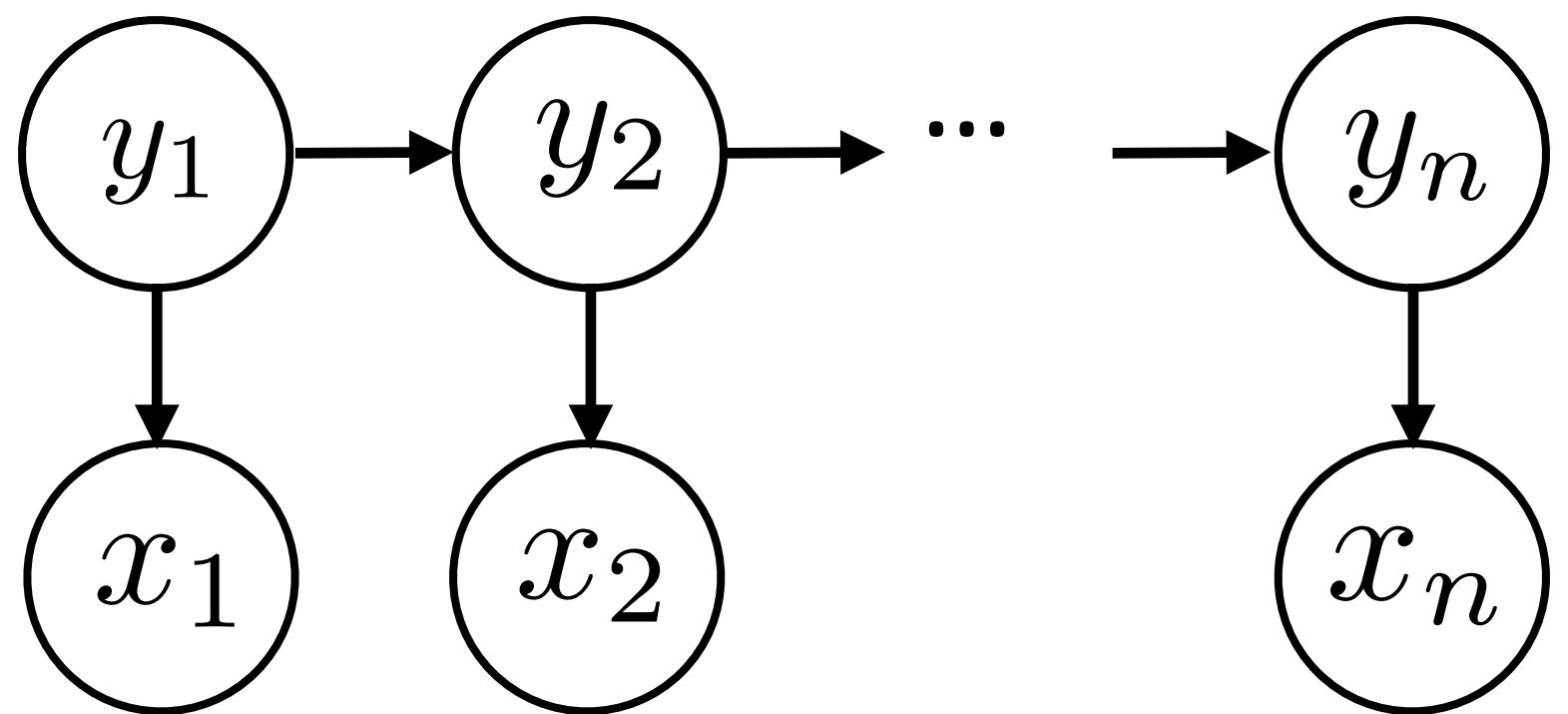


$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \prod_{i=2}^n P(y_i | y_{i-1}) \prod_{i=1}^n P(x_i | y_i)$$

# Hidden Markov Models

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$

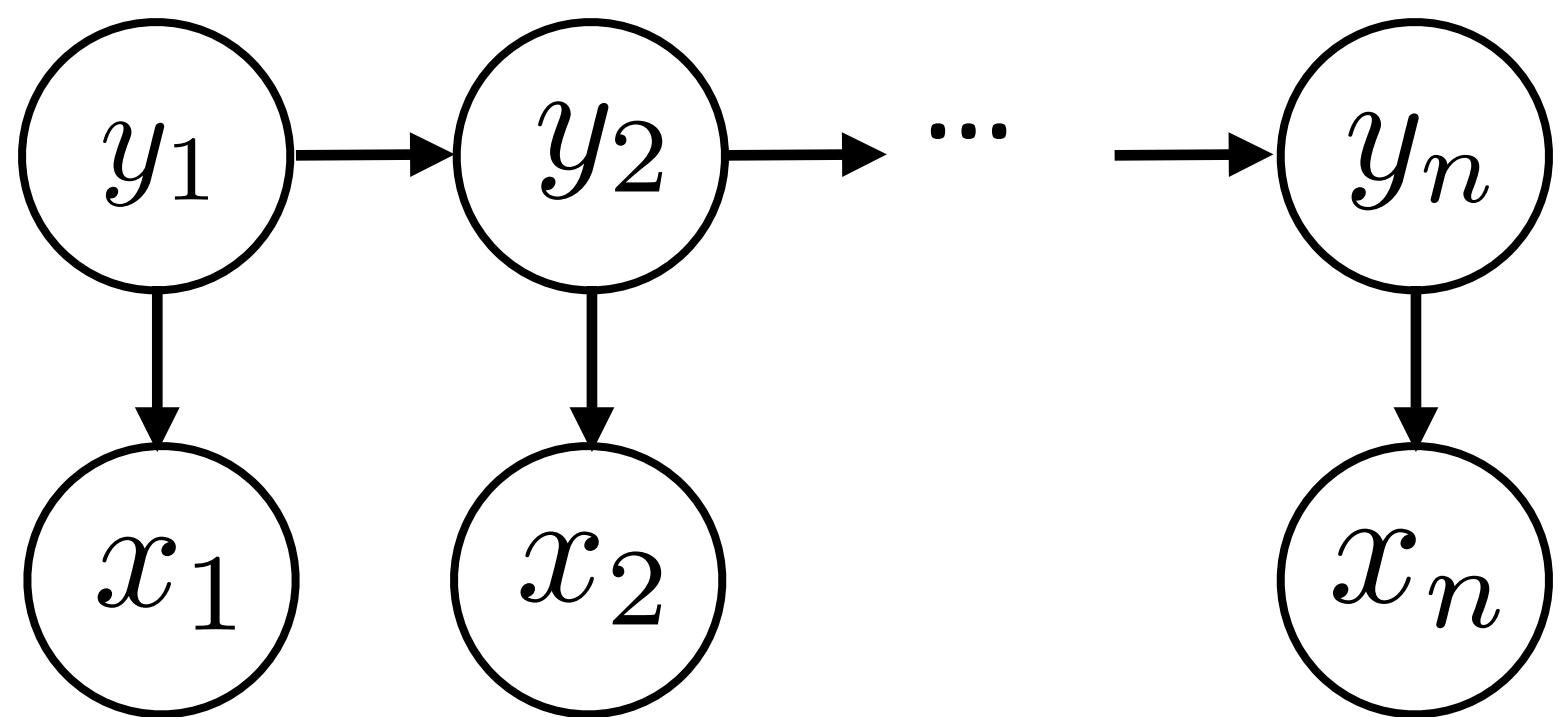


$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \underbrace{\prod_{i=2}^n P(y_i | y_{i-1})}_{\text{Initial distribution}} \prod_{i=1}^n P(x_i | y_i)$$

# Hidden Markov Models

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$

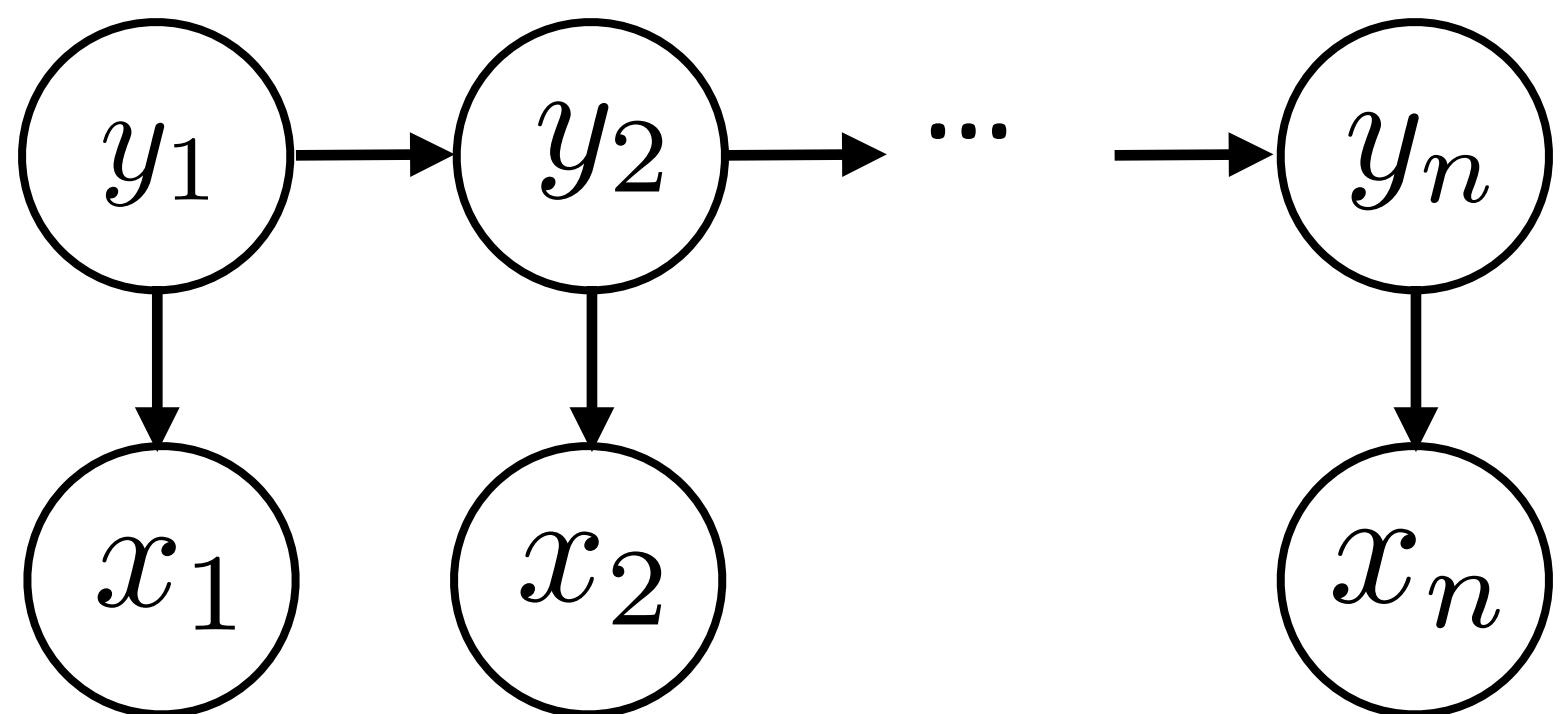


$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \underbrace{\prod_{i=2}^n P(y_i | y_{i-1})}_{\text{Initial distribution probabilities}} \underbrace{\prod_{i=1}^n P(x_i | y_i)}_{\text{Transition probabilities}}$$

# Hidden Markov Models

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$

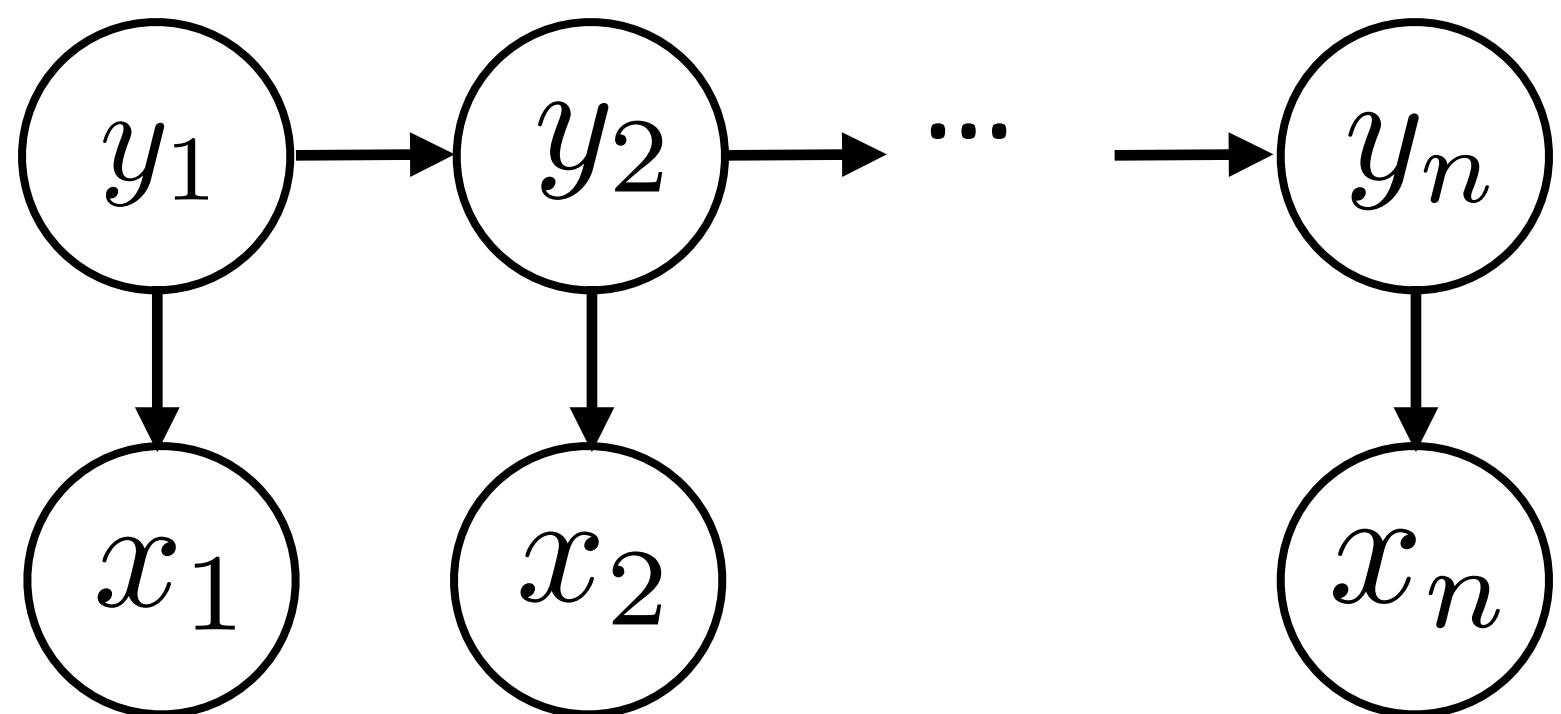


$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \underbrace{\prod_{i=2}^n P(y_i | y_{i-1})}_{\text{Initial distribution probabilities}} \underbrace{\prod_{i=1}^n P(x_i | y_i)}_{\text{Transition probabilities Emission probabilities}}$$

# Hidden Markov Models

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$



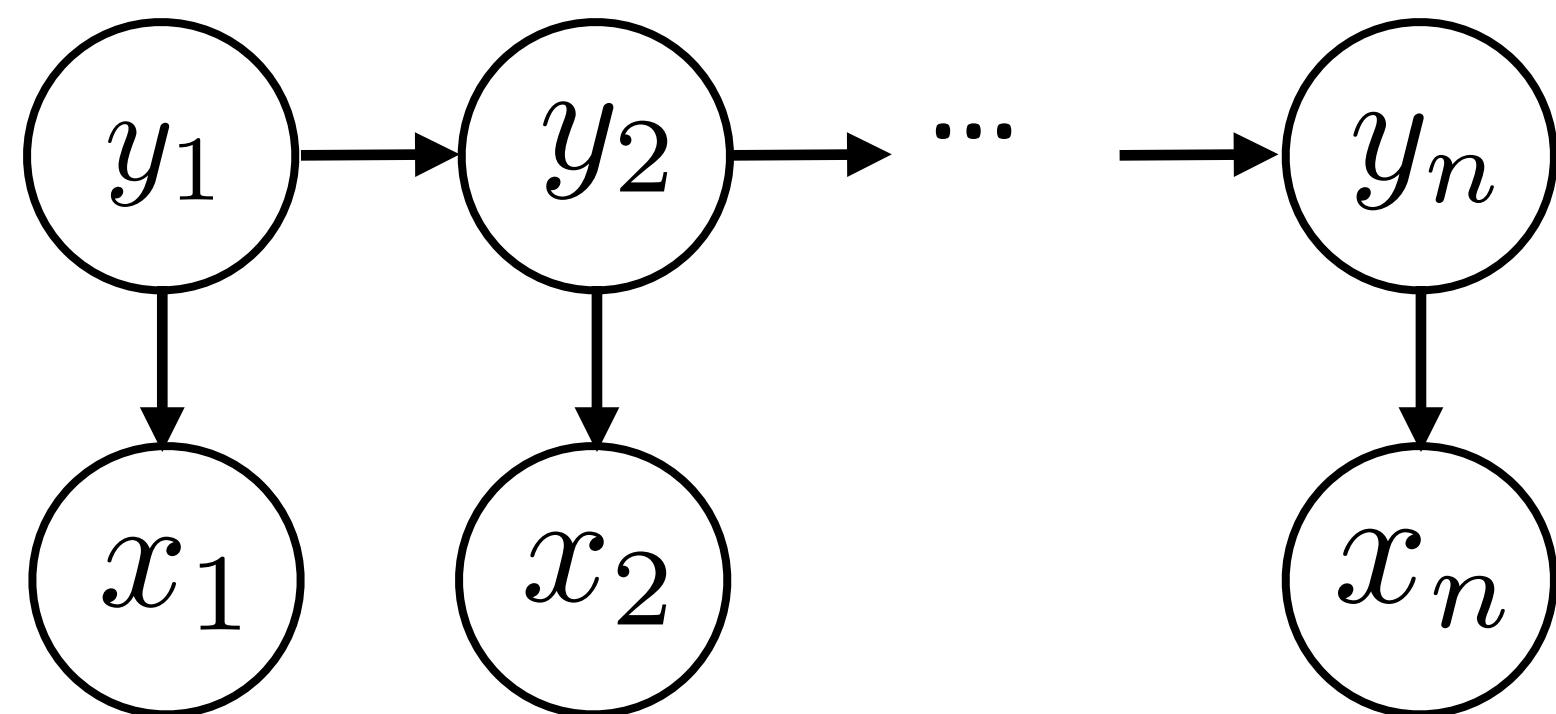
- ▶ Observation ( $x$ ) depends only on current state ( $y$ )

$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \underbrace{\prod_{i=2}^n P(y_i | y_{i-1})}_{\text{Initial distribution probabilities}} \underbrace{\prod_{i=1}^n P(x_i | y_i)}_{\text{Transition Emission probabilities}}$$

# Hidden Markov Models

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$

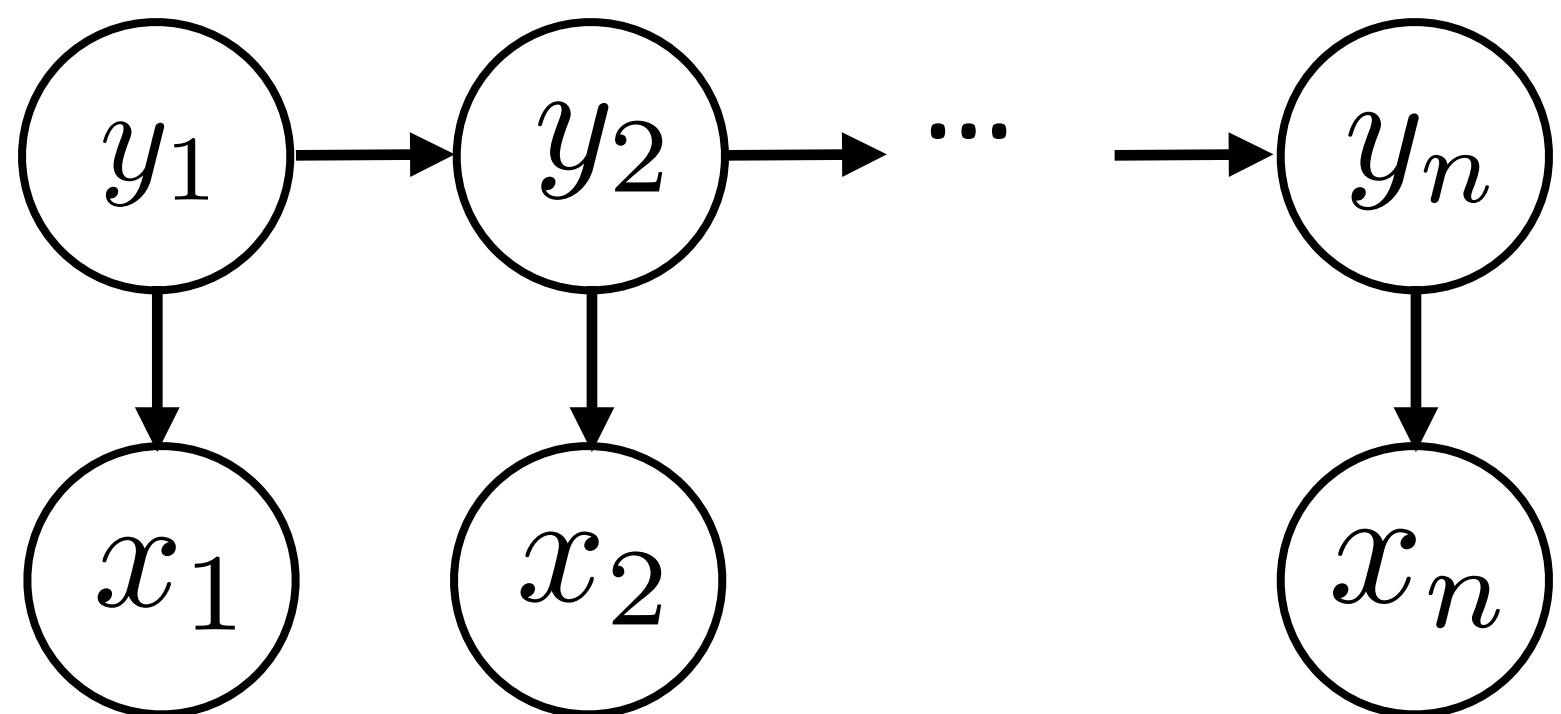


$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \underbrace{\prod_{i=2}^n P(y_i | y_{i-1})}_{\text{Initial distribution probabilities}} \underbrace{\prod_{i=1}^n P(x_i | y_i)}_{\text{Transition probabilities Emission probabilities}}$$

- ▶ Observation ( $x$ ) depends only on current state ( $y$ )
- ▶ Multinomials: tag x tag transitions, tag x word emissions

# Hidden Markov Models

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$    Output  $\mathbf{y} = (y_1, \dots, y_n)$



$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \underbrace{\prod_{i=2}^n P(y_i | y_{i-1})}_{\text{Initial distribution probabilities}} \underbrace{\prod_{i=1}^n P(x_i | y_i)}_{\text{Transition Emission probabilities}}$$

- ▶ Observation ( $x$ ) depends only on current state ( $y$ )
- ▶ Multinomials: tag x tag transitions, tag x word emissions
- ▶  $P(x|y)$  is a distribution over all words in the vocabulary
  - not a distribution over features (but could be!)

# Transitions in POS Tagging

---

- Dynamics model  $P(y_1) \prod_{i=2}^n P(y_i|y_{i-1})$
- VBD            VB  
VBN **VBZ**    VBP      VBZ  
**NNP** NNS     **NN**      **NNS** CD **NN** .

*Fed raises interest rates 0.5 percent.*

**NNP** - proper noun, singular  
**VBZ** - verb, 3rd ps. sing. present  
**NN** - noun, singular or mass

# Transitions in POS Tagging

---

- Dynamics model  $P(y_1) \prod_{i=2}^n P(y_i|y_{i-1})$
- VBD              VB  
VBN VBZ    VBP      VBZ  
NNP NNS    NN      NNS CD    NN .
- Fed raises interest rates 0.5 percent.*

NNP - proper noun, singular

VBZ - verb, 3rd ps. sing. present

NN - noun, singular or mass
- $P(y_1 = \text{NNP})$  likely because start of sentence

# Transitions in POS Tagging

---

- Dynamics model  $P(y_1) \prod_{i=2}^n P(y_i|y_{i-1})$
- VBD            VB  
VBN **VBZ**    VBP      VBZ  
**NNP** **NNS**    **NN**     **NNS** **CD** **NN** .

*Fed raises interest rates 0.5 percent.*

**NNP** - proper noun, singular  
**VBZ** - verb, 3rd ps. sing. present  
**NN** - noun, singular or mass

- $P(y_1 = \text{NNP})$  likely because start of sentence
- $P(y_2 = \text{VBZ}|y_1 = \text{NNP})$  likely because verb often follows noun

# Transitions in POS Tagging

---

- Dynamics model  $P(y_1) \prod_{i=2}^n P(y_i|y_{i-1})$
- VBD            VB  
VBN **VBZ**    VBP      VBZ  
**NNP** NNS     **NN**      **NNS** CD **NN** .

*Fed raises interest rates 0.5 percent.*

**NNP** - proper noun, singular  
**VBZ** - verb, 3rd ps. sing. present  
**NN** - noun, singular or mass

- $P(y_1 = \text{NNP})$  likely because start of sentence
- $P(y_2 = \text{VBZ}|y_1 = \text{NNP})$  likely because verb often follows noun
- $P(y_3 = \text{NN}|y_2 = \text{VBZ})$  direct object follows verb, other verb rarely follows past tense verb (main verbs can follow modals though!)

# Estimating Transitions

---

NNP VBZ NN NNS CD NN .

*Fed raises interest rates 0.5 percent.*

# Estimating Transitions

---

NNP VBZ NN NNS CD NN .

*Fed raises interest rates 0.5 percent.*

- ▶ Similar to Naive Bayes estimation: maximum likelihood solution = normalized counts (with smoothing) read off supervised data

# Estimating Transitions

---

NNP VBZ NN NNS CD NN .

*Fed raises interest rates 0.5 percent.*

- ▶ Similar to Naive Bayes estimation: maximum likelihood solution = normalized counts (with smoothing) read off supervised data
- ▶  $P(\text{tag} \mid \text{NN})$

# Estimating Transitions

---

NNP VBZ NN NNS CD NN .

*Fed raises interest rates 0.5 percent.*

- ▶ Similar to Naive Bayes estimation: maximum likelihood solution = normalized counts (with smoothing) read off supervised data
- ▶  $P(\text{tag} \mid \text{NN}) = (0.5 \text{ .}, 0.5 \text{ NNS})$

# Estimating Transitions

---

NNP VBZ NN NNS CD NN .

*Fed raises interest rates 0.5 percent.*

- ▶ Similar to Naive Bayes estimation: maximum likelihood solution = normalized counts (with smoothing) read off supervised data
- ▶  $P(\text{tag} \mid \text{NN}) = (0.5 \text{ .}, 0.5 \text{ NNS})$
- ▶ How to smooth?

# Estimating Transitions

---

NNP VBZ NN NNS CD NN .

*Fed raises interest rates 0.5 percent.*

- ▶ Similar to Naive Bayes estimation: maximum likelihood solution = normalized counts (with smoothing) read off supervised data
- ▶  $P(\text{tag} \mid \text{NN}) = (0.5 \text{ .}, 0.5 \text{ NNS})$
- ▶ How to smooth?
- ▶ One method: smooth with unigram distribution over tags

# Estimating Transitions

---

NNP VBZ NN NNS CD NN .

*Fed raises interest rates 0.5 percent.*

- ▶ Similar to Naive Bayes estimation: maximum likelihood solution = normalized counts (with smoothing) read off supervised data
- ▶  $P(\text{tag} \mid \text{NN}) = (0.5 \text{ .}, 0.5 \text{ NNS})$
- ▶ How to smooth?
- ▶ One method: smooth with unigram distribution over tags

$$P(\text{tag} | \text{tag}_{-1}) = (1 - \lambda) \hat{P}(\text{tag} | \text{tag}_{-1}) + \lambda \hat{P}(\text{tag})$$

# Estimating Transitions

---

NNP VBZ NN NNS CD NN .

*Fed raises interest rates 0.5 percent.*

- ▶ Similar to Naive Bayes estimation: maximum likelihood solution = normalized counts (with smoothing) read off supervised data
- ▶  $P(\text{tag} \mid \text{NN}) = (0.5 \text{ .}, 0.5 \text{ NNS})$
- ▶ How to smooth?
- ▶ One method: smooth with unigram distribution over tags

$$P(\text{tag} | \text{tag}_{-1}) = (1 - \lambda) \hat{P}(\text{tag} | \text{tag}_{-1}) + \lambda \hat{P}(\text{tag})$$

$\hat{P}$  = empirical distribution (read off from data)

# Emissions in POS Tagging

---

NNP VBZ NN NNS CD NN .

*Fed raises interest rates 0.5 percent.*

# Emissions in POS Tagging

---

NNP VBZ NN NNS CD NN .

*Fed raises interest rates 0.5 percent.*

- ▶ Emissions  $P(x | y)$  capture the distribution of words occurring with a given tag

# Emissions in POS Tagging

---

NNP VBZ NN NNS CD NN .

*Fed raises interest rates 0.5 percent.*

- ▶ Emissions  $P(x | y)$  capture the distribution of words occurring with a given tag
- ▶  $P(\text{word} | \text{NN}) = (0.05 \text{ person}, 0.04 \text{ official}, 0.03 \text{ interest}, 0.03 \text{ percent} \dots)$

# Emissions in POS Tagging

---

NNP VBZ NN NNS CD NN .

*Fed raises interest rates 0.5 percent.*

- ▶ Emissions  $P(x | y)$  capture the distribution of words occurring with a given tag
- ▶  $P(\text{word} | \text{NN}) = (0.05 \text{ person}, 0.04 \text{ official}, 0.03 \text{ interest}, 0.03 \text{ percent} \dots)$
- ▶ When you compute the posterior for a given word's tags, the distribution favors tags that are more likely to generate that word

# Emissions in POS Tagging

---

NNP VBZ NN NNS CD NN .

*Fed raises interest rates 0.5 percent.*

- ▶ Emissions  $P(x | y)$  capture the distribution of words occurring with a given tag
- ▶  $P(\text{word} | \text{NN}) = (0.05 \text{ person}, 0.04 \text{ official}, 0.03 \text{ interest}, 0.03 \text{ percent} \dots)$
- ▶ When you compute the posterior for a given word's tags, the distribution favors tags that are more likely to generate that word
- ▶ How should we smooth this?

# Estimating Emissions

---

NNP VBZ NN NNS CD NN

Fed raises interest rates 0.5 percent

# Estimating Emissions

---

NNP VBZ NN NNS CD NN

Fed raises interest rates 0.5 percent

- ▶  $P(\text{word} \mid \text{NN}) = (0.5 \textit{interest}, 0.5 \textit{percent})$  – hard to smooth!

# Estimating Emissions

---

NNP VBZ NN NNS CD NN

Fed raises interest rates 0.5 percent

- ▶  $P(\text{word} \mid \text{NN}) = (0.5 \text{ } interest, 0.5 \text{ } percent)$  — hard to smooth!
- ▶ Can interpolate with distribution looking at word shape  
 $P(\text{word shape} \mid \text{tag})$  (e.g.,  $P(\text{capitalized word of len} \geq 8 \mid \text{tag})$ )

# Estimating Emissions

---

NNP VBZ NN NNS CD NN

Fed raises interest rates 0.5 percent

- ▶  $P(\text{word} \mid \text{NN}) = (0.5 \text{ } interest, 0.5 \text{ } percent)$  — hard to smooth!
- ▶ Can interpolate with distribution looking at word shape  
 $P(\text{word shape} \mid \text{tag})$  (e.g.,  $P(\text{capitalized word of len} \geq 8 \mid \text{tag})$ )
- ▶ Alternative: use Bayes' rule

# Estimating Emissions

---

NNP VBZ NN NNS CD NN

Fed raises interest rates 0.5 percent

- ▶  $P(\text{word} \mid \text{NN}) = (0.5 \textit{interest}, 0.5 \textit{percent})$  — hard to smooth!
- ▶ Can interpolate with distribution looking at word shape  
 $P(\text{word shape} \mid \text{tag})$  (e.g.,  $P(\text{capitalized word of len } \geq 8 \mid \text{tag})$ )
- ▶ Alternative: use Bayes' rule

$$P(\text{word}|\text{tag}) = \frac{P(\text{tag}|\text{word})P(\text{word})}{P(\text{tag})}$$

# Estimating Emissions

---

NNP VBZ NN NNS CD NN

Fed raises interest rates 0.5 percent

- ▶  $P(\text{word} \mid \text{NN}) = (0.5 \textit{interest}, 0.5 \textit{percent})$  — hard to smooth!
  - ▶ Can interpolate with distribution looking at word shape  
 $P(\text{word shape} \mid \text{tag})$  (e.g.,  $P(\text{capitalized word of len } \geq 8 \mid \text{tag})$ )
  - ▶ Alternative: use Bayes' rule
- $$P(\text{word}|\text{tag}) = \frac{P(\text{tag}|\text{word})P(\text{word})}{P(\text{tag})}$$
- ▶ Fancy techniques from language modeling, e.g. look at type fertility
    - $P(\text{tag}|\text{word})$  is flatter for some kinds of words than for others)

# Estimating Emissions

---

NNP VBZ NN NNS CD NN

Fed raises interest rates 0.5 percent

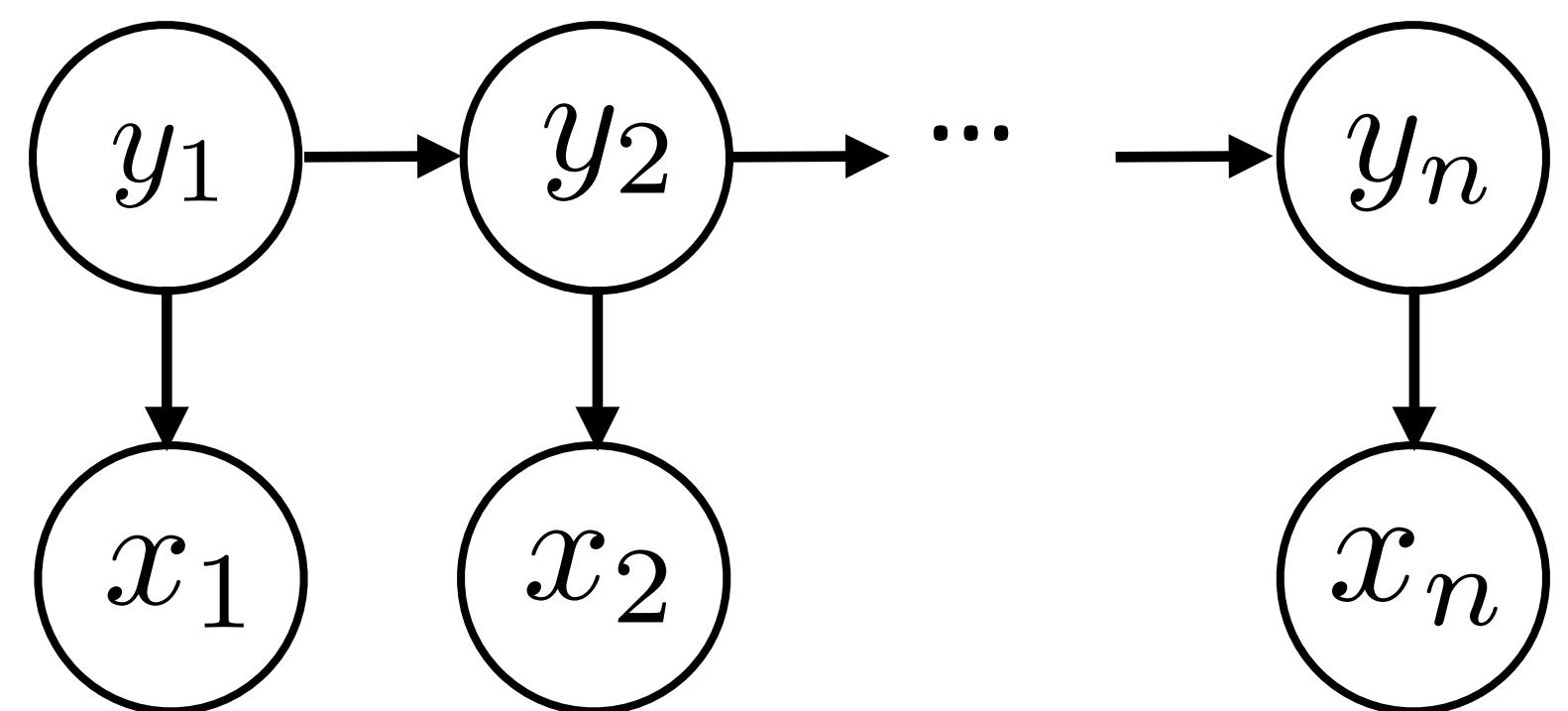
- ▶  $P(\text{word} \mid \text{NN}) = (0.5 \textit{interest}, 0.5 \textit{percent})$  — hard to smooth!
  - ▶ Can interpolate with distribution looking at word shape  
 $P(\text{word shape} \mid \text{tag})$  (e.g.,  $P(\text{capitalized word of len } \geq 8 \mid \text{tag})$ )
  - ▶ Alternative: use Bayes' rule
- $$P(\text{word}|\text{tag}) = \frac{P(\text{tag}|\text{word})P(\text{word})}{P(\text{tag})}$$
- ▶ Fancy techniques from language modeling, e.g. look at type fertility
    - $P(\text{tag}|\text{word})$  is flatter for some kinds of words than for others)
  - ▶  $P(\text{word}|\text{tag})$  can be a log-linear model — we'll see this in a few lectures

# Inference in HMMs

---

► Input  $\mathbf{x} = (x_1, \dots, x_n)$

Output  $\mathbf{y} = (y_1, \dots, y_n)$



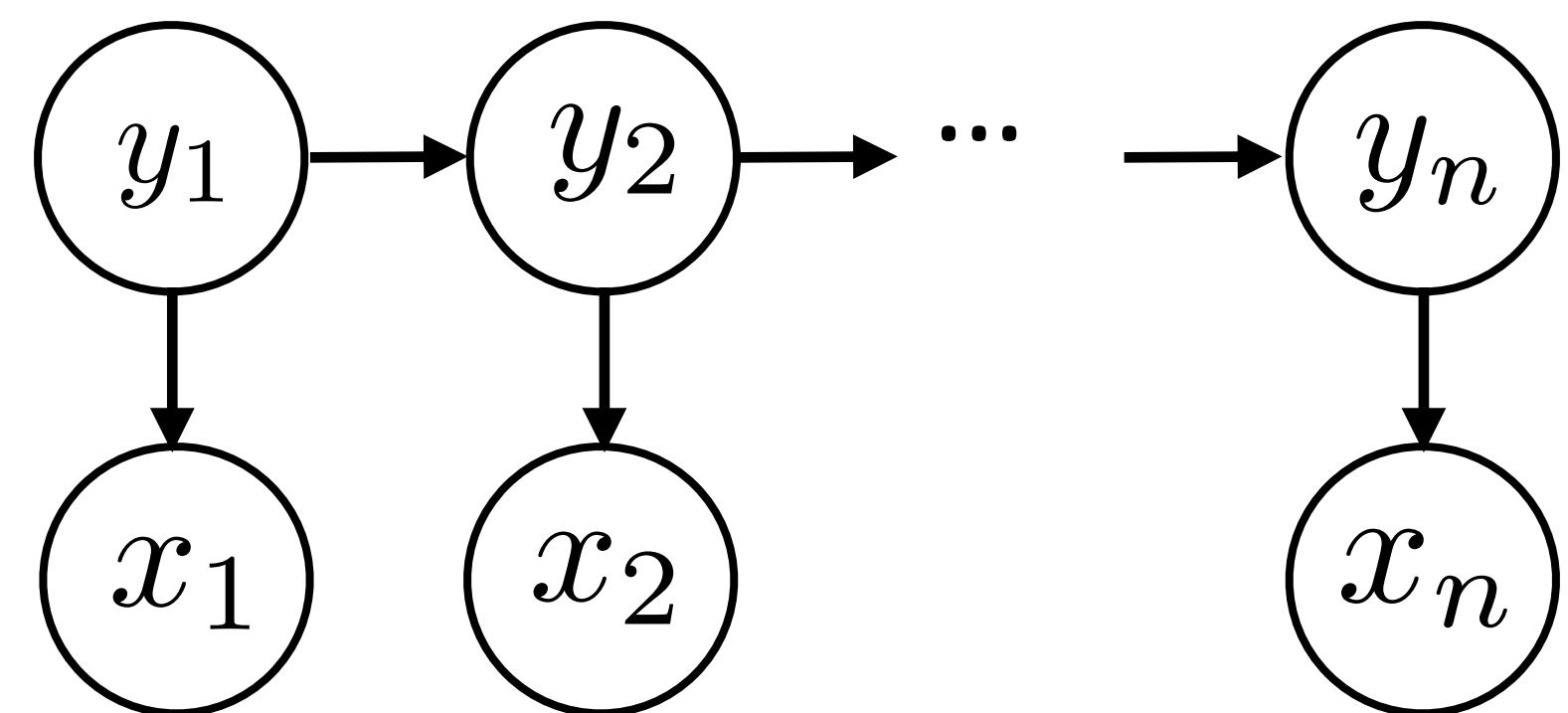
$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \prod_{i=2}^n P(y_i|y_{i-1}) \prod_{i=1}^n P(x_i|y_i)$$

# Inference in HMMs

---

► Input  $\mathbf{x} = (x_1, \dots, x_n)$

Output  $\mathbf{y} = (y_1, \dots, y_n)$



$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \prod_{i=2}^n P(y_i|y_{i-1}) \prod_{i=1}^n P(x_i|y_i)$$

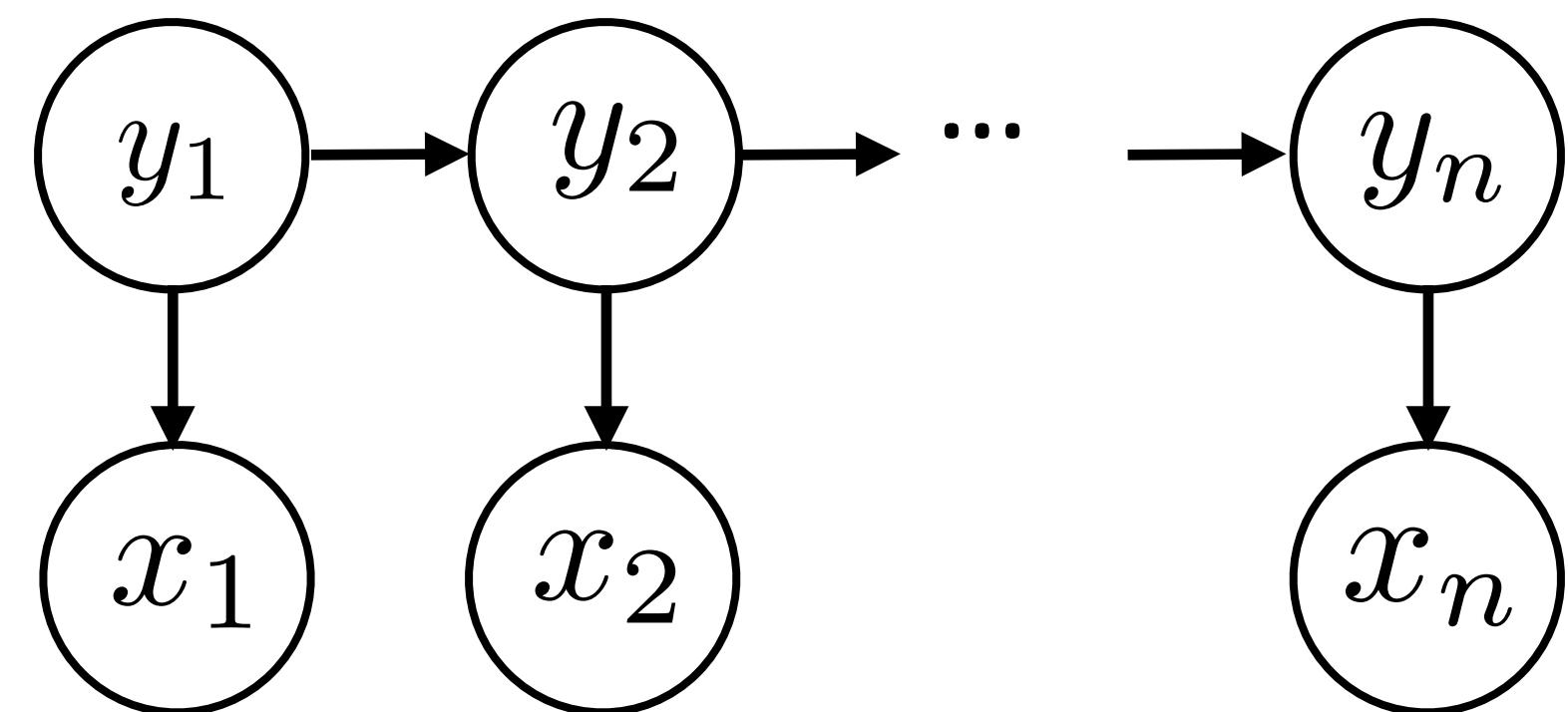
► Inference problem:  $\text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) = \text{argmax}_{\mathbf{y}} \frac{P(\mathbf{y}, \mathbf{x})}{P(\mathbf{x})}$

# Inference in HMMs

---

► Input  $\mathbf{x} = (x_1, \dots, x_n)$

Output  $\mathbf{y} = (y_1, \dots, y_n)$



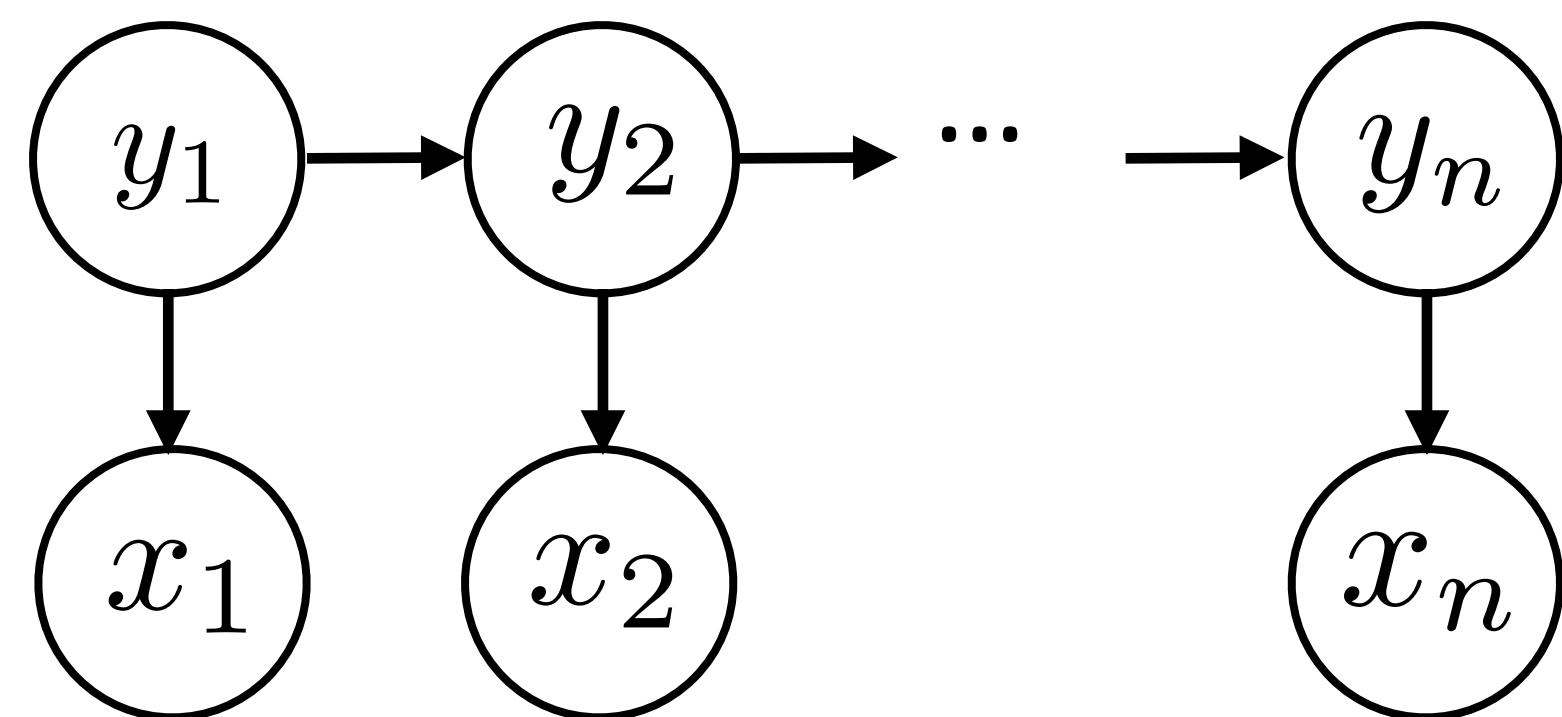
$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \prod_{i=2}^n P(y_i|y_{i-1}) \prod_{i=1}^n P(x_i|y_i)$$

► Inference problem:  $\text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) = \text{argmax}_{\mathbf{y}} \frac{P(\mathbf{y}, \mathbf{x})}{P(\mathbf{x})}$

# Inference in HMMs

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$       Output  $\mathbf{y} = (y_1, \dots, y_n)$



$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \prod_{i=2}^n P(y_i|y_{i-1}) \prod_{i=1}^n P(x_i|y_i)$$

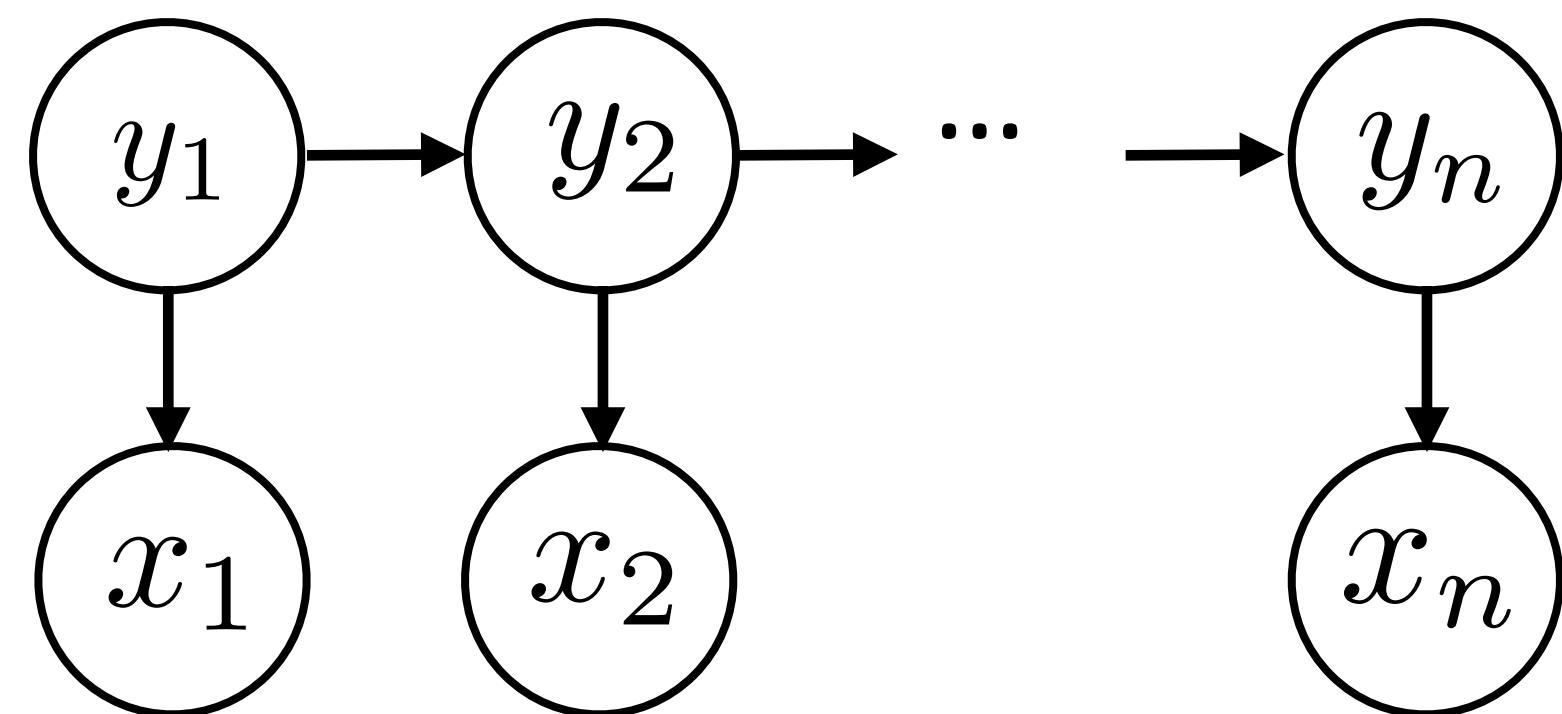
- ▶ Inference problem:  $\text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) = \text{argmax}_{\mathbf{y}} \frac{P(\mathbf{y}, \mathbf{x})}{P(\mathbf{x})}$
- ▶ Exponentially many possible  $\mathbf{y}$  here!

# Inference in HMMs

---

► Input  $\mathbf{x} = (x_1, \dots, x_n)$

Output  $\mathbf{y} = (y_1, \dots, y_n)$

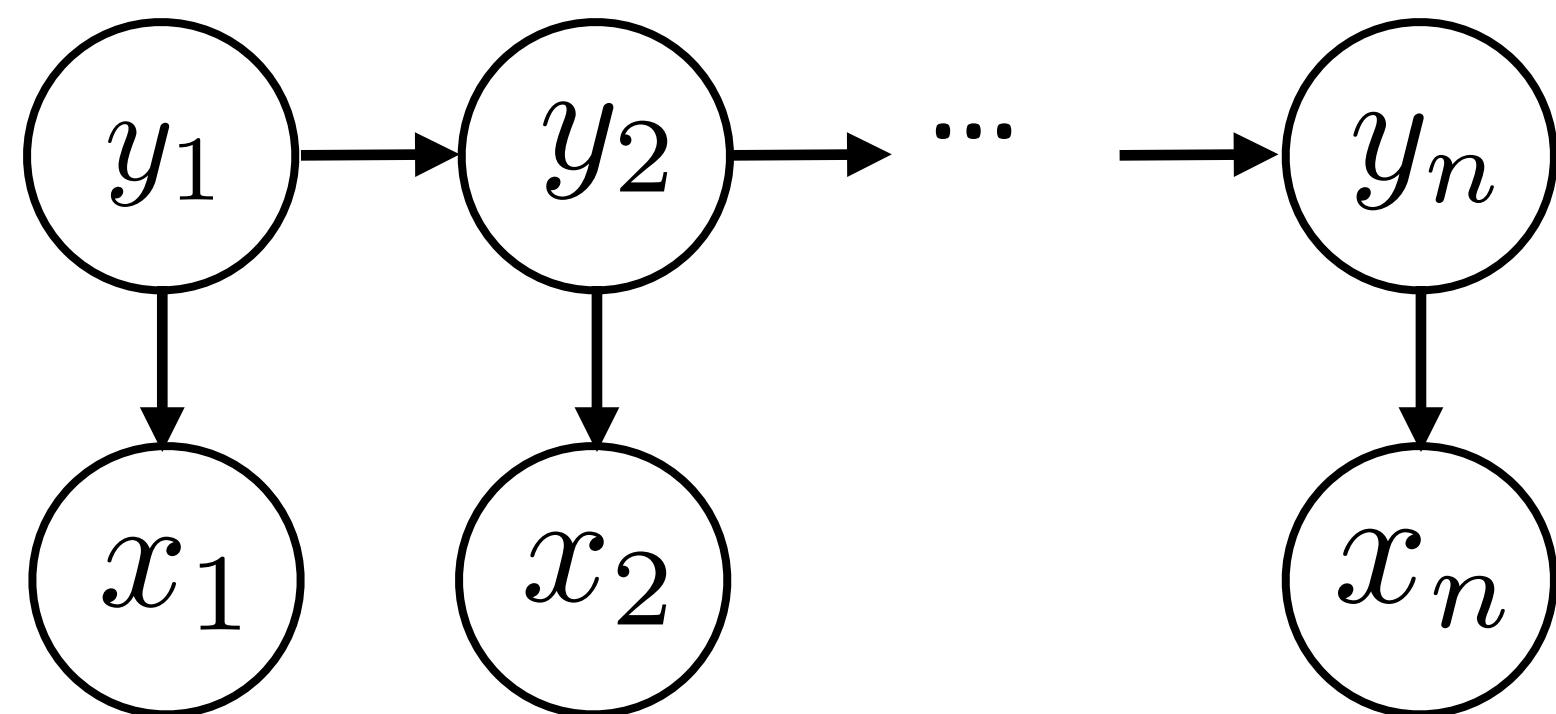


$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \prod_{i=2}^n P(y_i | y_{i-1}) \prod_{i=1}^n P(x_i | y_i)$$

- Inference problem:  $\operatorname{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) = \operatorname{argmax}_{\mathbf{y}} \frac{P(\mathbf{y}, \mathbf{x})}{P(\mathbf{x})}$
- Exponentially many possible  $\mathbf{y}$  here!
- Solution: dynamic programming (possible because of **Markov structure!**)

# Inference in HMMs

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$



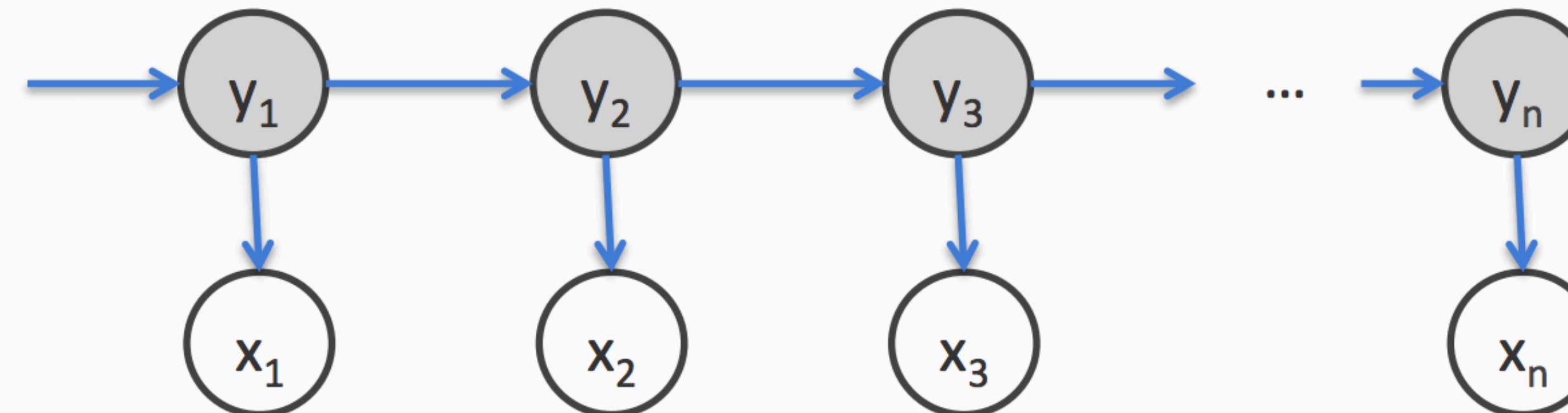
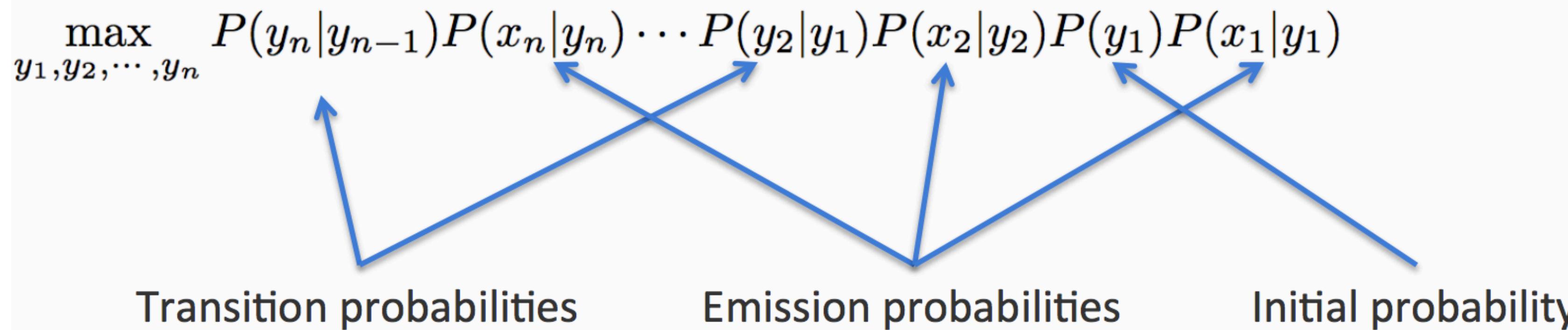
- ▶ Output  $\mathbf{y} = (y_1, \dots, y_n)$

$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \prod_{i=2}^n P(y_i|y_{i-1}) \prod_{i=1}^n P(x_i|y_i)$$

- ▶ Inference problem:  $\operatorname{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) = \operatorname{argmax}_{\mathbf{y}} \frac{P(\mathbf{y}, \mathbf{x})}{P(\mathbf{x})}$
- ▶ Exponentially many possible  $\mathbf{y}$  here!
- ▶ Solution: dynamic programming (possible because of **Markov structure!**)
  - ▶ Many neural sequence models depend on entire previous tag sequence, need to use approximations like beam search

# Viterbi Algorithm

$$P(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n) = P(y_1) \prod_{i=1}^{n-1} P(y_{i+1}|y_i) \prod_{i=1}^n P(x_i|y_i)$$



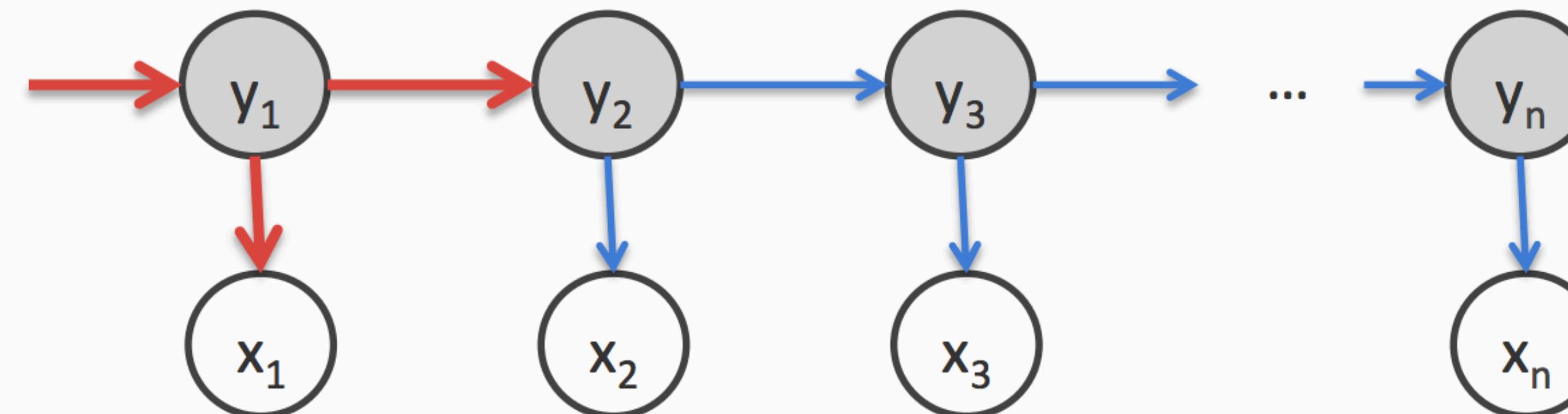
slide credit: Vivek Srikumar

# Viterbi Algorithm

$$P(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n) = P(y_1) \prod_{i=1}^{n-1} P(y_{i+1}|y_i) \prod_{i=1}^n P(x_i|y_i)$$

$$\begin{aligned} & \max_{y_1, y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \end{aligned}$$

The only terms that depend on  $y_1$



slide credit: Vivek Srikumar

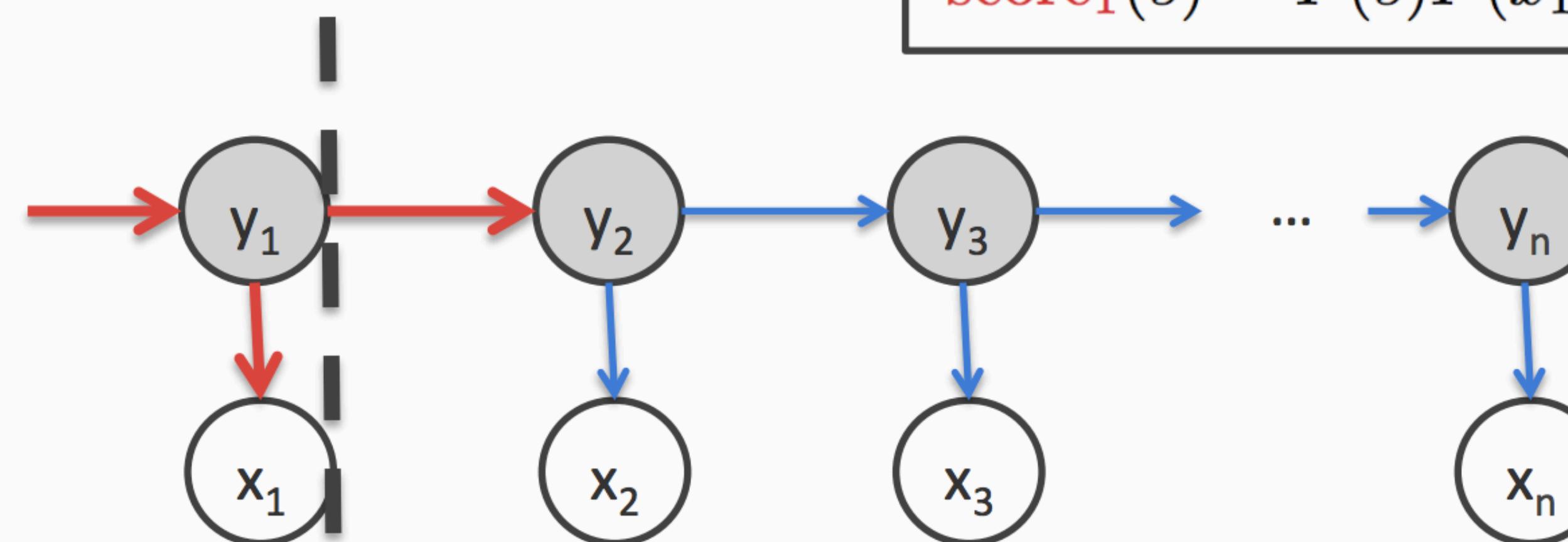
# Viterbi Algorithm

$$P(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n) = P(y_1) \prod_{i=1}^{n-1} P(y_{i+1}|y_i) \prod_{i=1}^n P(x_i|y_i)$$

$$\begin{aligned} & \max_{y_1, y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2) \text{score}_1(y_1) \end{aligned}$$

Abstract away the score for all decisions till here into **score**

$$\text{score}_1(s) = P(s)P(x_1|s)$$



# Viterbi Algorithm

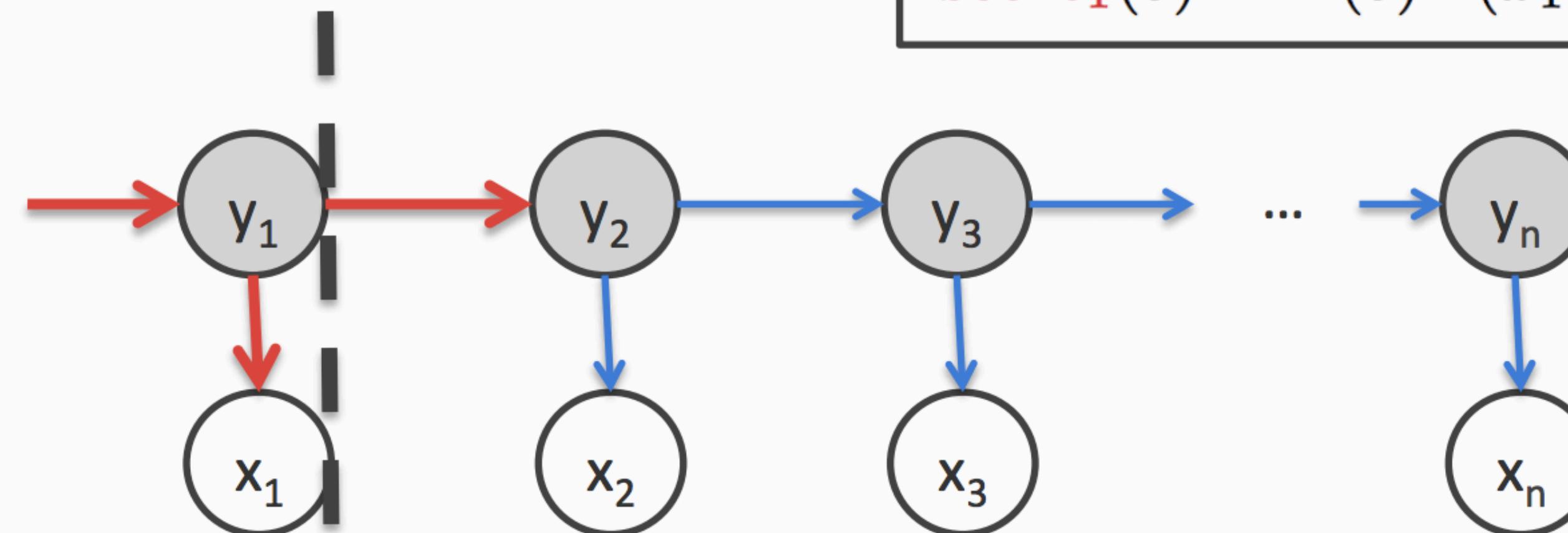
$$P(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n) = P(y_1) \prod_{i=1}^{n-1} P(y_{i+1}|y_i) \prod_{i=1}^n P(x_i|y_i)$$

$$\begin{aligned} & \max_{y_1, y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2) \text{score}_1(y_1) \end{aligned}$$

best (partial) score for  
a sequence ending in state  $s$

Abstract away the score for all  
decisions till here into **score**

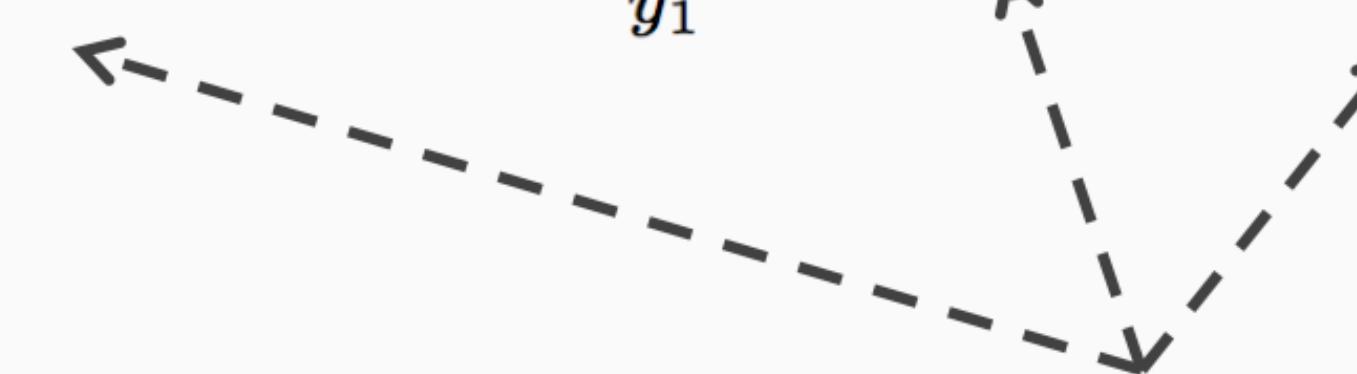
$$\text{score}_1(s) = P(s)P(x_1|s)$$



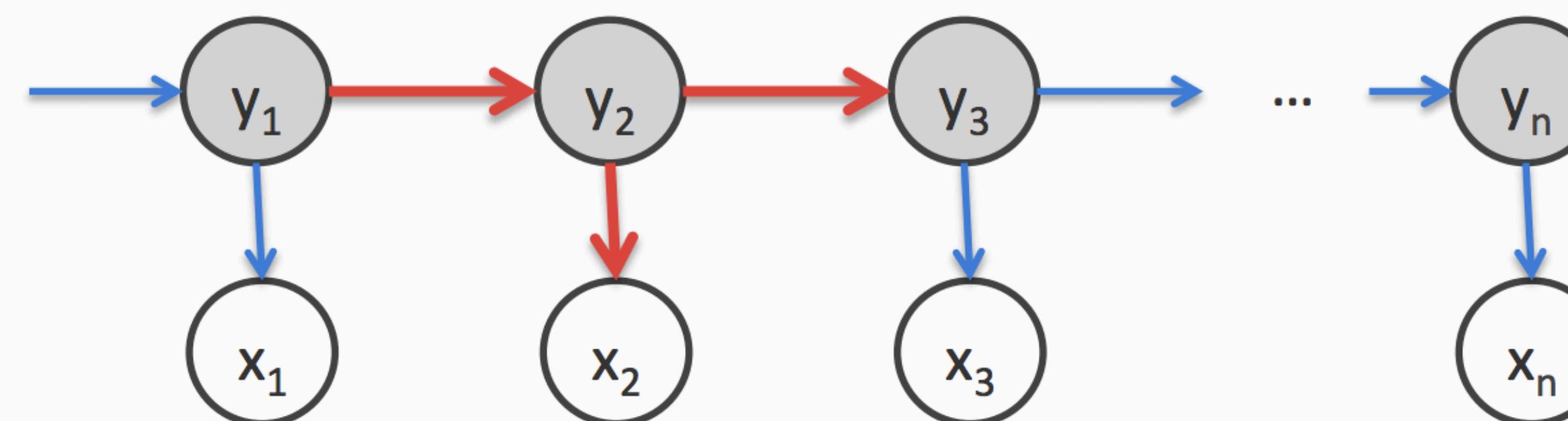
# Viterbi Algorithm

$$P(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n) = P(y_1) \prod_{i=1}^{n-1} P(y_{i+1}|y_i) \prod_{i=1}^n P(x_i|y_i)$$

$$\begin{aligned} & \max_{y_1, y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2)\text{score}_1(y_1) \\ &= \max_{y_3, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_2} P(y_3|y_2)P(x_3|y_3) \max_{y_1} P(y_2|y_1)P(x_2|y_2)\text{score}_1(y_1) \end{aligned}$$



Only terms that depend on  $y_2$

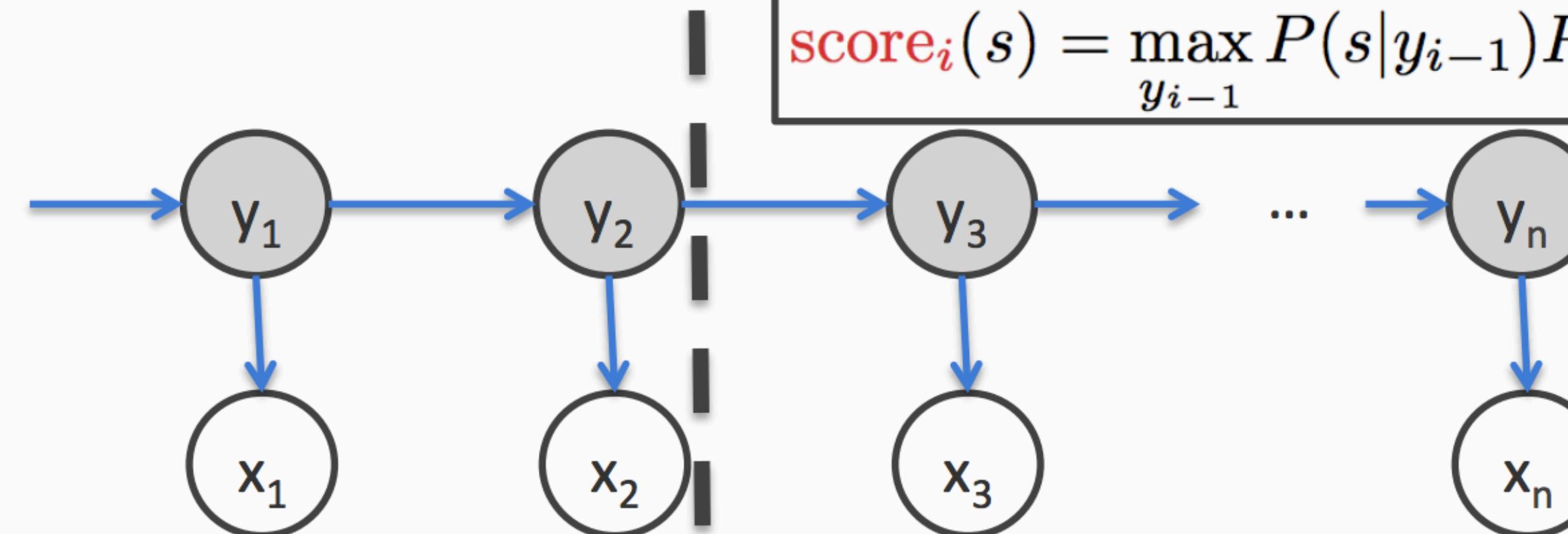


# Viterbi Algorithm

$$P(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n) = P(y_1) \prod_{i=1}^{n-1} P(y_{i+1}|y_i) \prod_{i=1}^n P(x_i|y_i)$$

$$\begin{aligned} & \max_{y_1, y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2) \text{score}_1(y_1) \\ &= \max_{y_3, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_2} P(y_3|y_2)P(x_3|y_3) \max_{y_1} P(y_2|y_1)P(x_2|y_2) \text{score}_1(y_1) \\ &= \max_{y_3, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_2} P(y_3|y_2)P(x_3|y_3) \text{score}_2(y_2) \end{aligned}$$

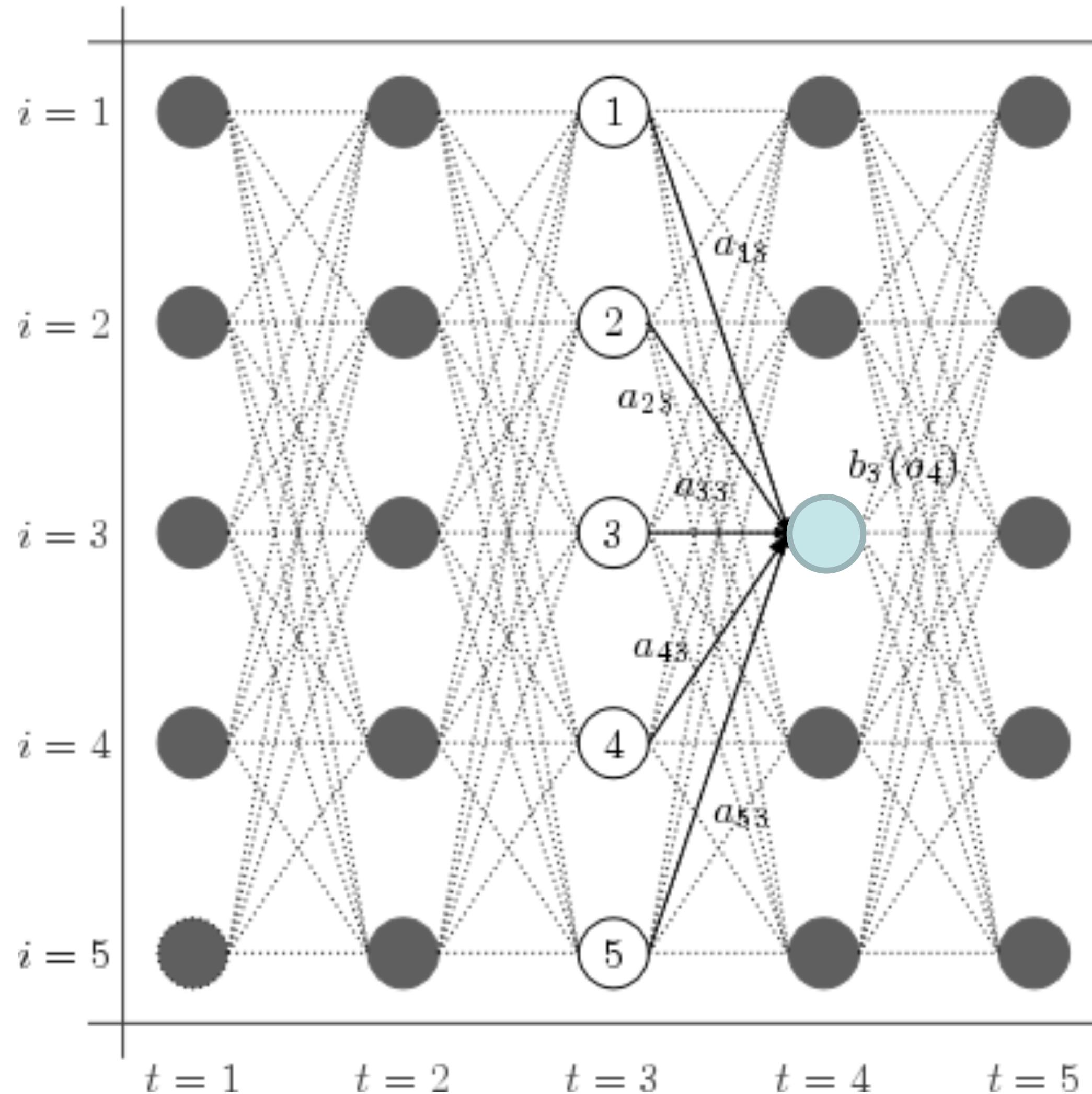
$$\text{score}_i(s) = \max_{y_{i-1}} P(s|y_{i-1})P(x_i|s) \text{score}_{i-1}(y_{i-1})$$



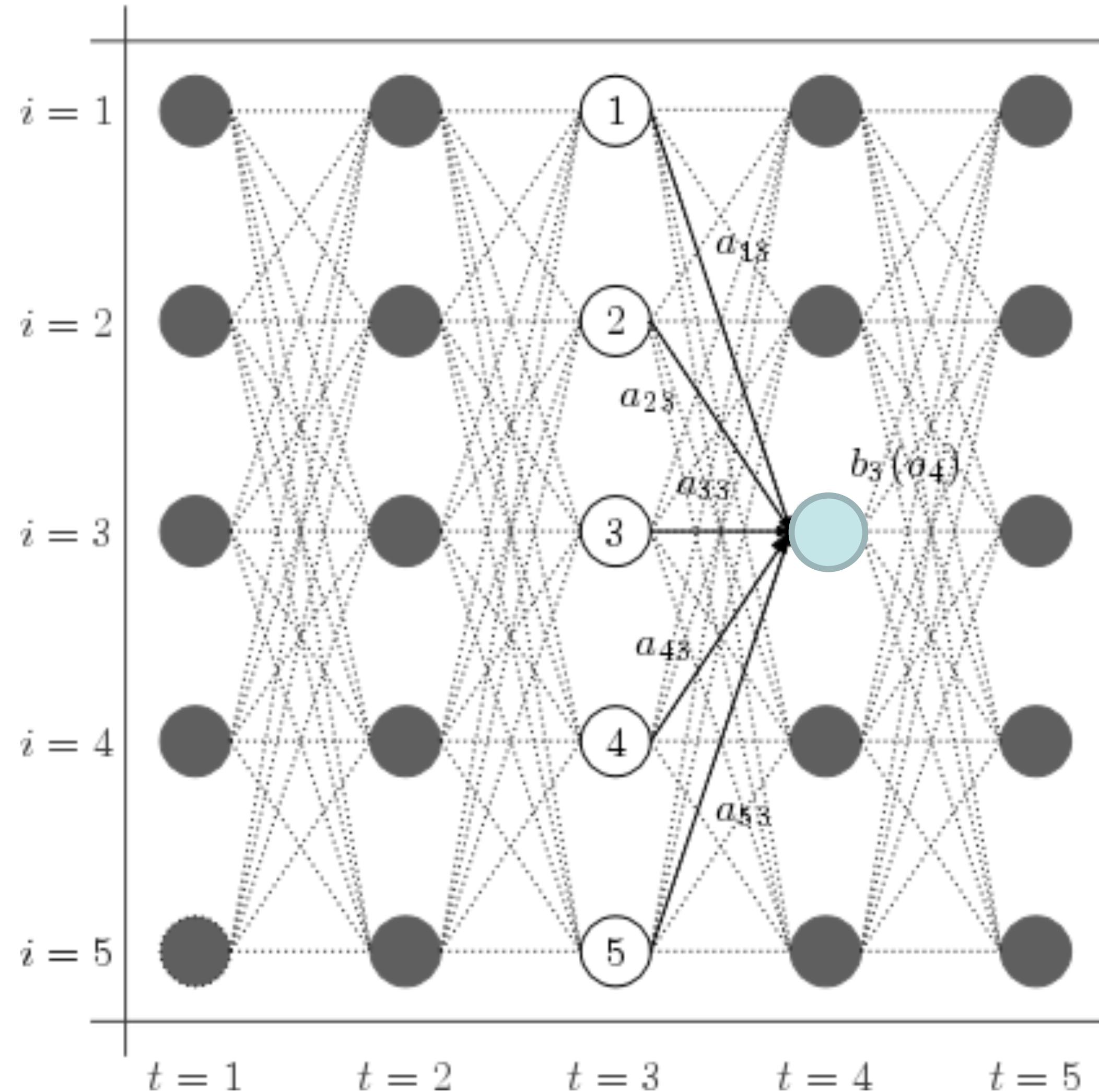
Abstract away the score for all decisions till here into **score**

slide credit: Vivek Srikumar

# Viterbi Algorithm



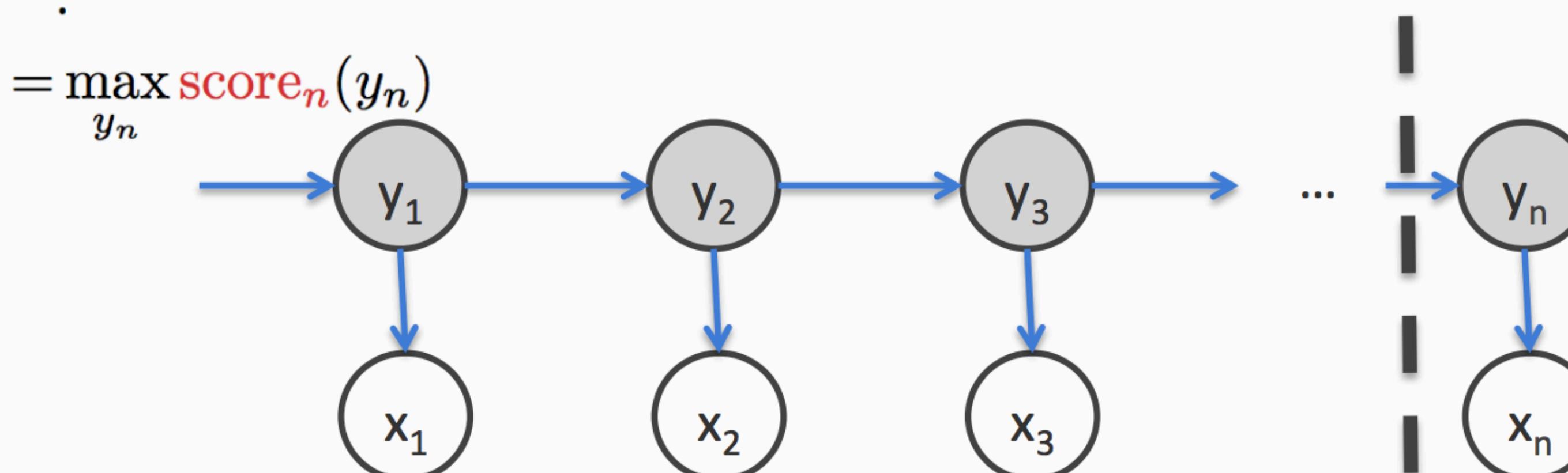
# Viterbi Algorithm



- ▶ “Think about” all possible immediate prior state values. Everything before that has already been accounted for by earlier stages.

# Viterbi Algorithm

$$\begin{aligned} P(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n) &= P(y_1) \prod_{i=1}^{n-1} P(y_{i+1}|y_i) \prod_{i=1}^n P(x_i|y_i) \\ &\max_{y_1, y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2)\text{score}_1(y_1) \\ &= \max_{y_3, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_2} P(y_3|y_2)P(x_3|y_3) \max_{y_1} P(y_2|y_1)P(x_2|y_2)\text{score}_1(y_1) \\ &= \max_{y_3, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_2} P(y_3|y_2)P(x_3|y_3)\text{score}_2(y_2) \\ &\vdots \\ &= \max_{y_n} \text{score}_n(y_n) \end{aligned}$$



Abstract away the score for all decisions till here into **score**

slide credit: Vivek Srikumar

# Viterbi Algorithm

$$\begin{aligned} P(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n) &= P(y_1) \prod_{i=1}^{n-1} P(y_{i+1}|y_i) \prod_{i=1}^n P(x_i|y_i) \\ &\max_{y_1, y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2)P(y_1)P(x_1|y_1) \\ &= \max_{y_2, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_1} P(y_2|y_1)P(x_2|y_2)\text{score}_1(y_1) \\ &= \max_{y_3, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_2} P(y_3|y_2)P(x_3|y_3) \max_{y_1} P(y_2|y_1)P(x_2|y_2)\text{score}_1(y_1) \\ &= \max_{y_3, \dots, y_n} P(y_n|y_{n-1})P(x_n|y_n) \cdots \max_{y_2} P(y_3|y_2)P(x_3|y_3)\text{score}_2(y_2) \\ &\vdots \\ &= \max_{y_n} \text{score}_n(y_n) \end{aligned}$$

$$\text{score}_1(s) = P(s)P(x_1|s)$$

$$\text{score}_i(s) = \max_{y_{i-1}} P(s|y_{i-1})P(x_i|s)\text{score}_{i-1}(y_{i-1})$$

# Viterbi Algorithm

1. **Initial:** For each state  $s$ , calculate

$$\text{score}_1(s) = P(s)P(x_1|s) = \pi_s B_{x_1,s}$$

2. **Recurrence:** For  $i = 2$  to  $n$ , for every state  $s$ , calculate

$$\begin{aligned}\text{score}_i(s) &= \max_{y_{i-1}} P(s|y_{i-1})P(x_i|s)\text{score}_{i-1}(y_{i-1}) \\ &= \max_{y_{i-1}} A_{y_{i-1},s}B_{s,x_i}\text{score}_{i-1}(y_{i-1})\end{aligned}$$

3. **Final state:** calculate

$$\max_{\mathbf{y}} P(\mathbf{y}, \mathbf{x}|\pi, A, B) = \max_s \text{score}_n(s)$$

$\pi$ : Initial probabilities  
A: Transitions  
B: Emissions

This only calculates the max. To get final answer (*argmax*),

- keep track of which state corresponds to the max at each step
- build the answer using these back pointers

# Forward-Backward Algorithm

---

# Forward-Backward Algorithm

---

- ▶ In addition to finding the best path, we may want to compute marginal probabilities of paths  $P(y_i = s | \mathbf{x})$

# Forward-Backward Algorithm

---

- ▶ In addition to finding the best path, we may want to compute marginal probabilities of paths  $P(y_i = s | \mathbf{x})$

$$P(y_i = s | \mathbf{x}) = \sum_{y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_n} P(\mathbf{y} | \mathbf{x})$$

# Forward-Backward Algorithm

---

- ▶ In addition to finding the best path, we may want to compute marginal probabilities of paths  $P(y_i = s | \mathbf{x})$

$$P(y_i = s | \mathbf{x}) = \sum_{y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_n} P(\mathbf{y} | \mathbf{x})$$

- ▶ What did Viterbi compute?  $P(\mathbf{y}_{\max} | \mathbf{x}) = \max_{y_1, \dots, y_n} P(\mathbf{y} | \mathbf{x})$

# Forward-Backward Algorithm

---

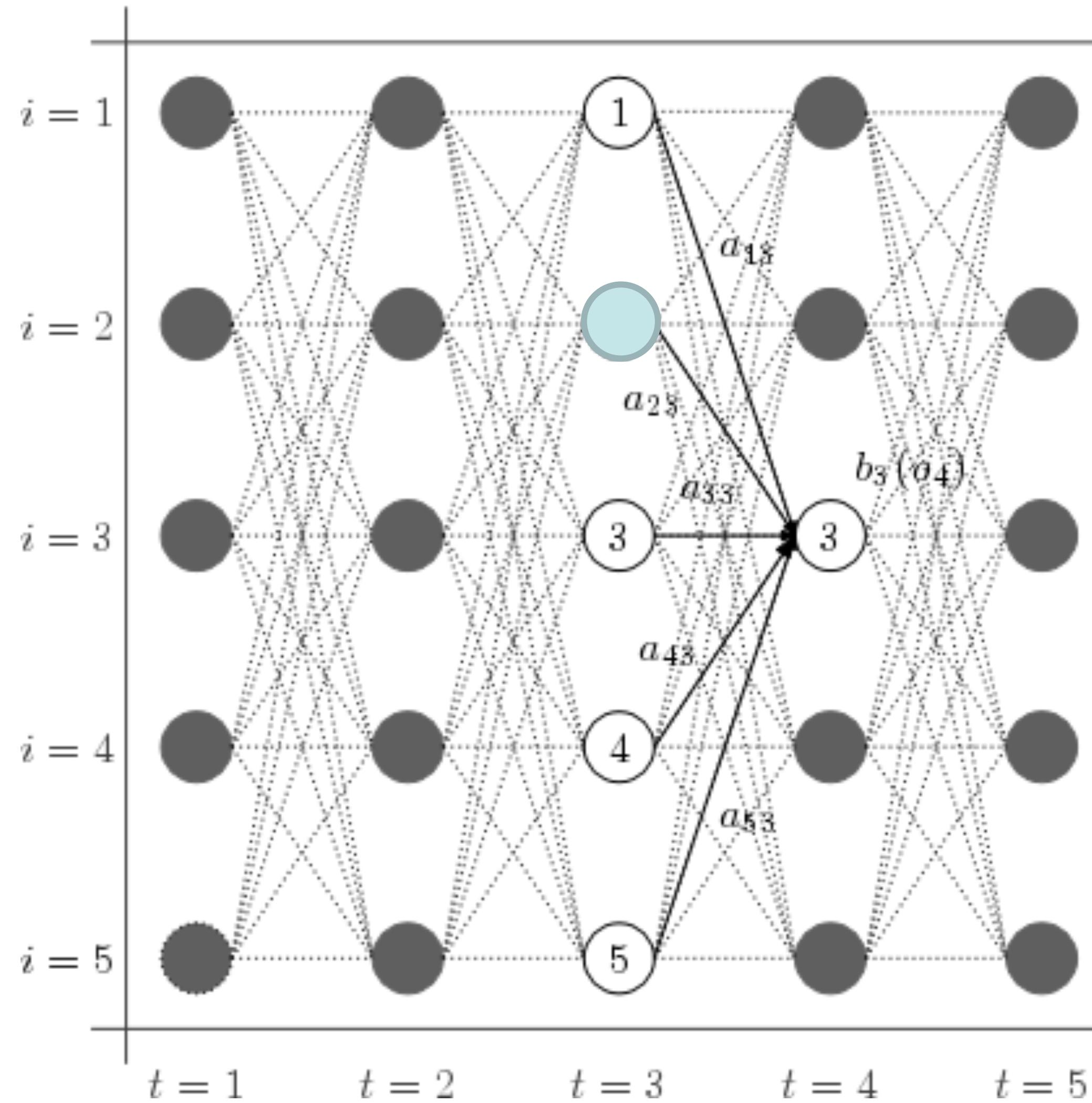
- ▶ In addition to finding the best path, we may want to compute marginal probabilities of paths  $P(y_i = s | \mathbf{x})$

$$P(y_i = s | \mathbf{x}) = \sum_{y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_n} P(\mathbf{y} | \mathbf{x})$$

- ▶ What did Viterbi compute?  $P(\mathbf{y}_{\max} | \mathbf{x}) = \max_{y_1, \dots, y_n} P(\mathbf{y} | \mathbf{x})$
- ▶ Can compute marginals with dynamic programming as well using an algorithm called forward-backward

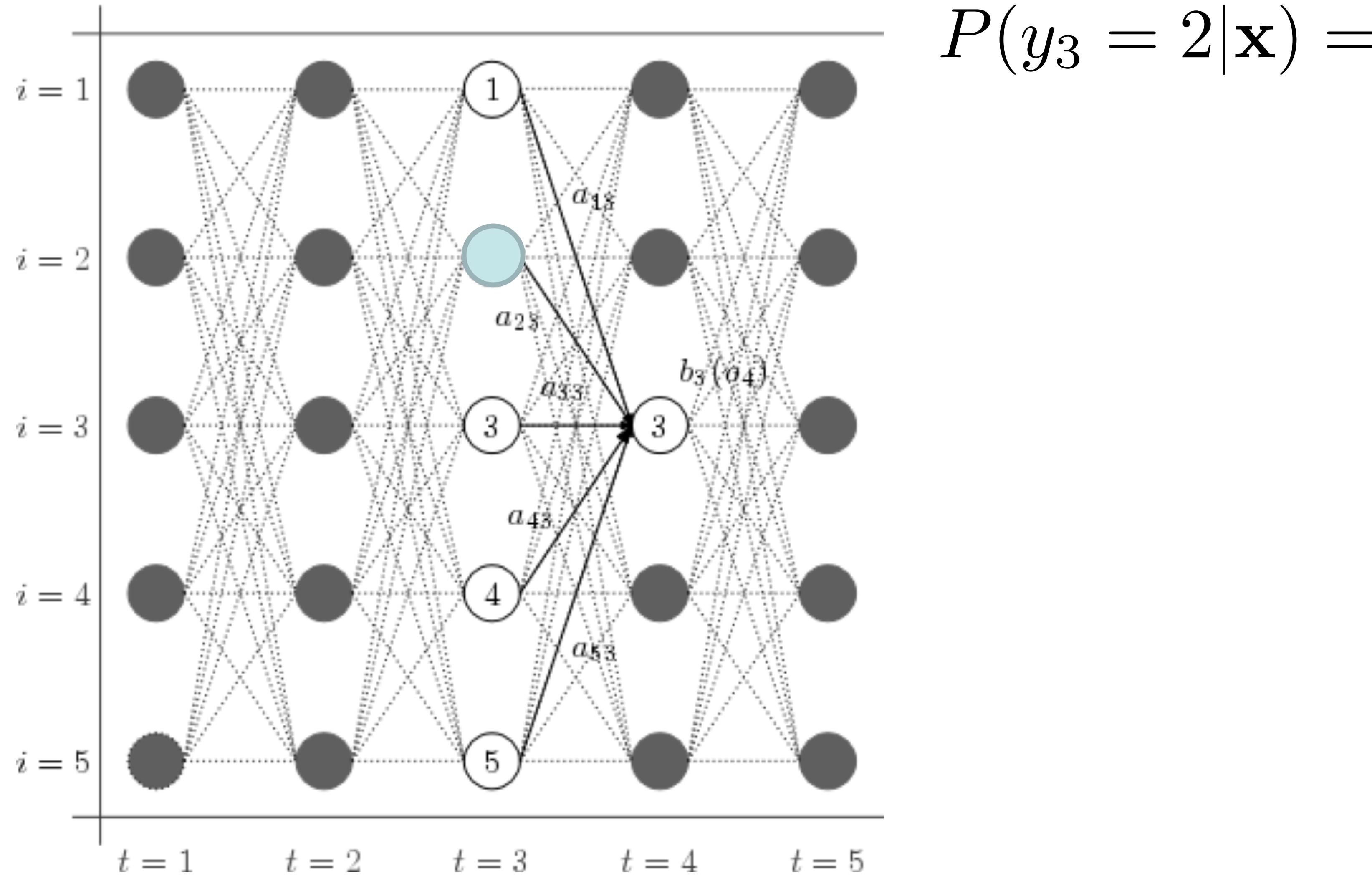
# Forward-Backward Algorithm

---

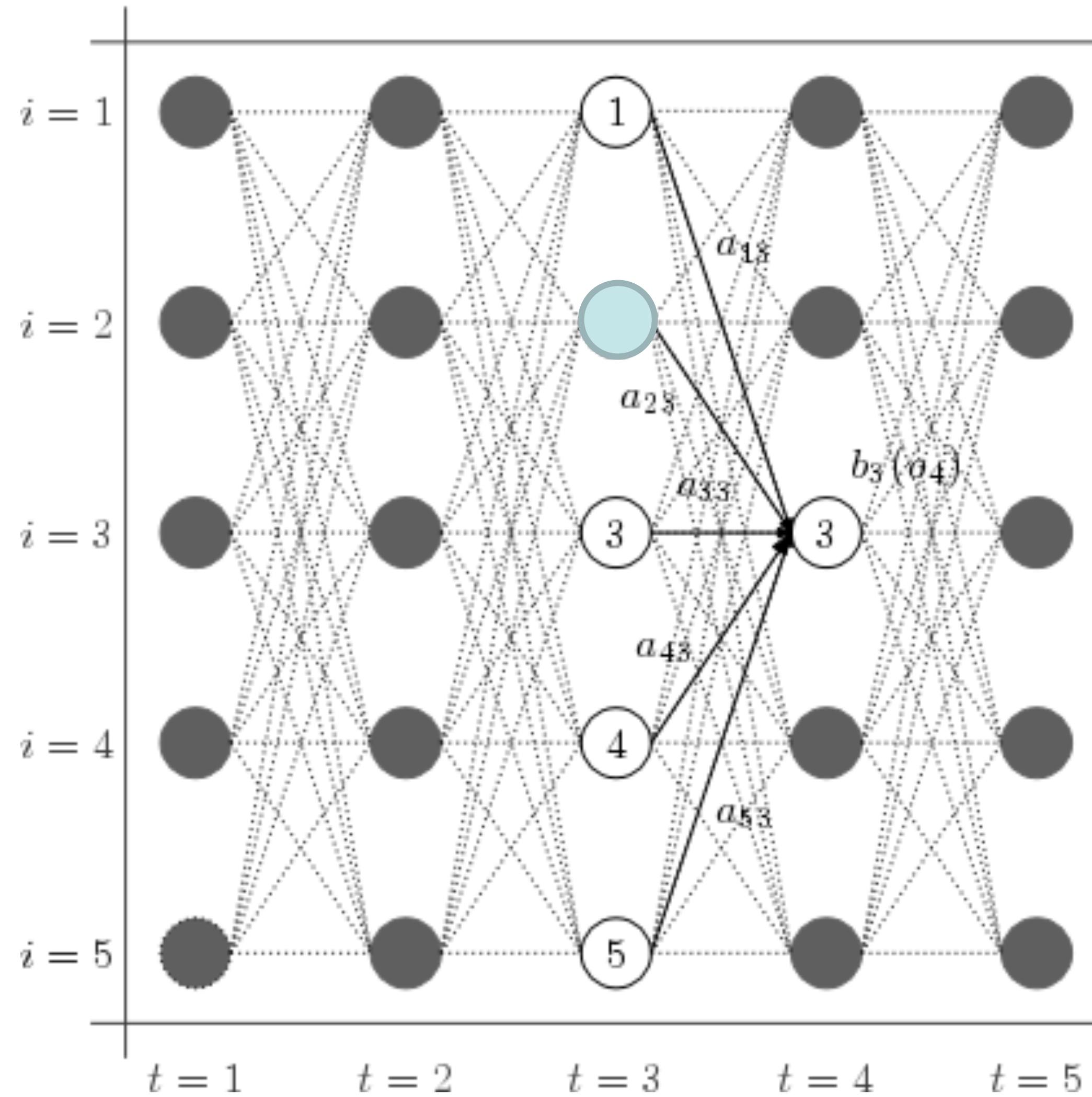


# Forward-Backward Algorithm

---



# Forward-Backward Algorithm

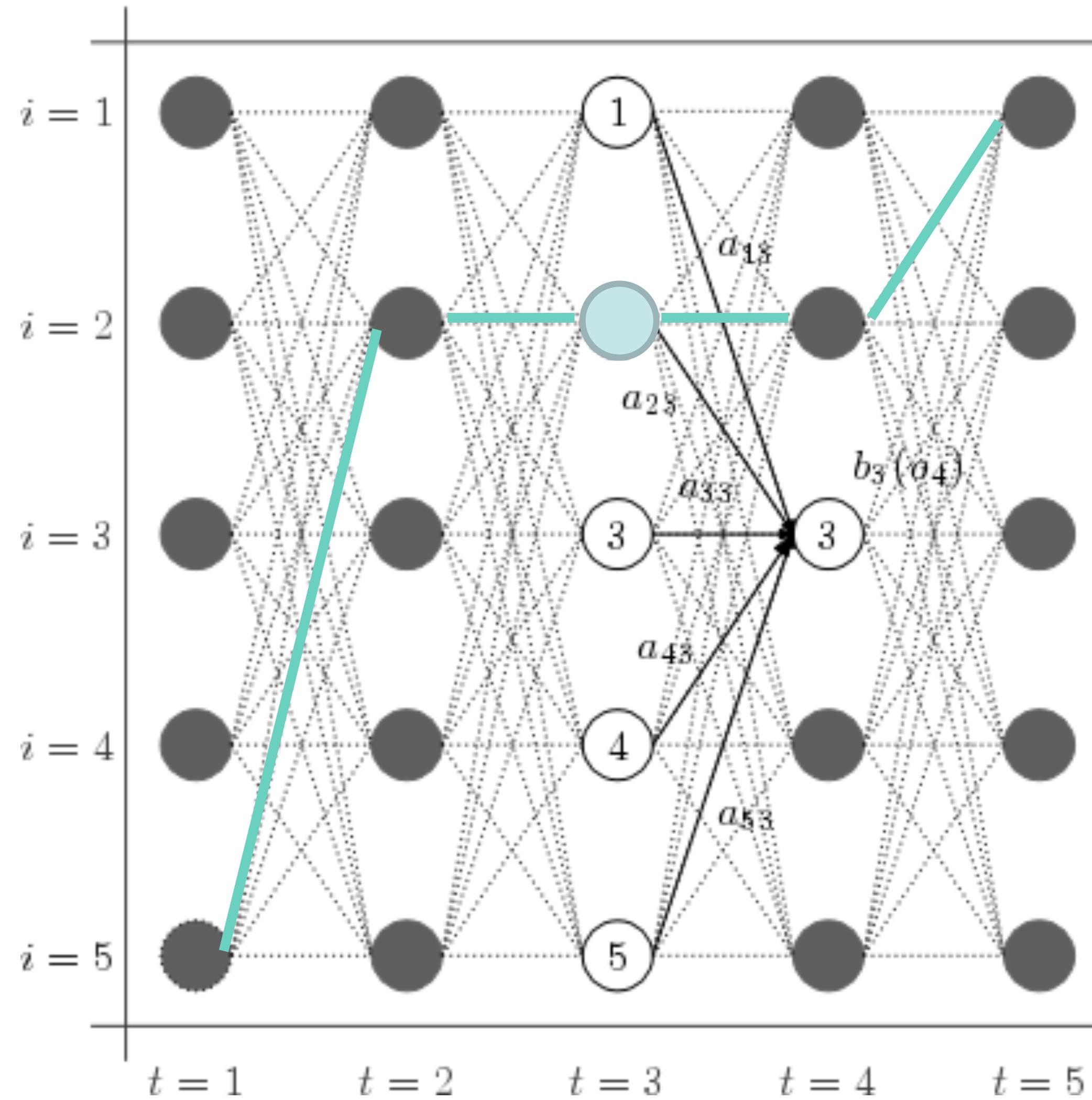


$$P(y_3 = 2 | \mathbf{x}) =$$

sum of all paths through state 2 at time 3

sum of all paths

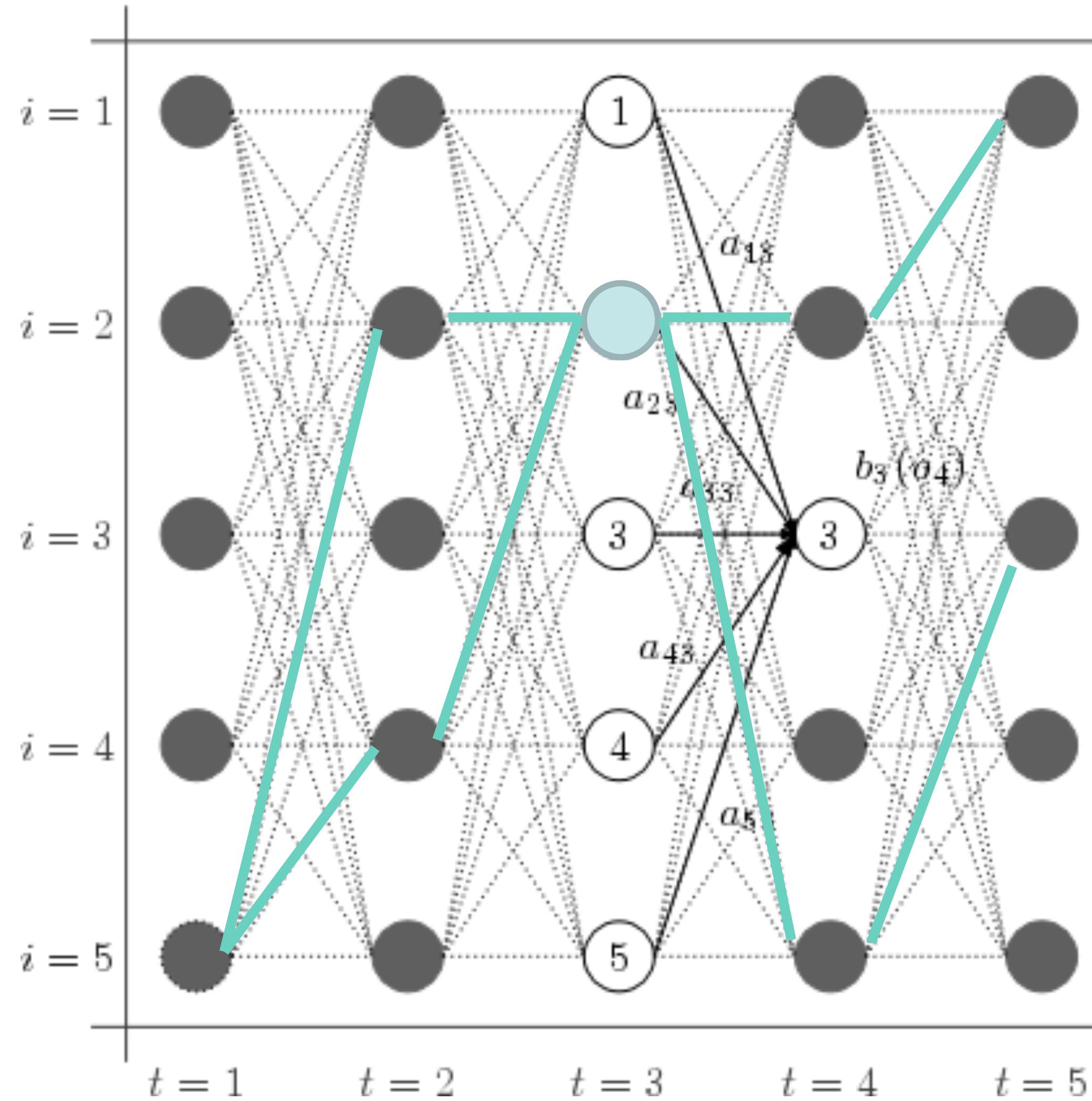
# Forward-Backward Algorithm



$$P(y_3 = 2 | \mathbf{x}) =$$

$$\frac{\text{sum of all paths through state 2 at time 3}}{\text{sum of all paths}}$$

# Forward-Backward Algorithm

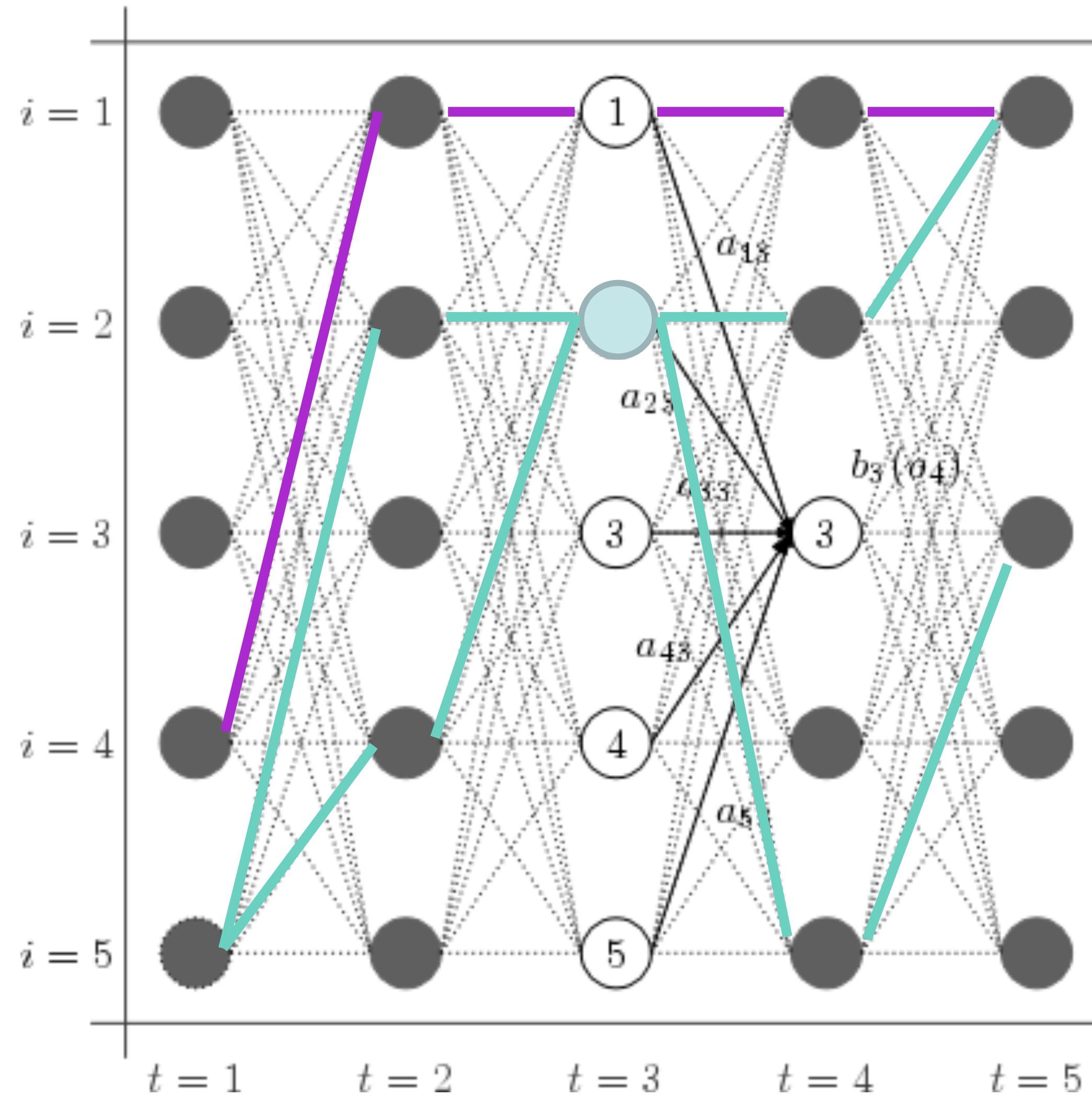


$$P(y_3 = 2 | \mathbf{x}) =$$

sum of all paths through state 2 at time 3

sum of all paths

# Forward-Backward Algorithm

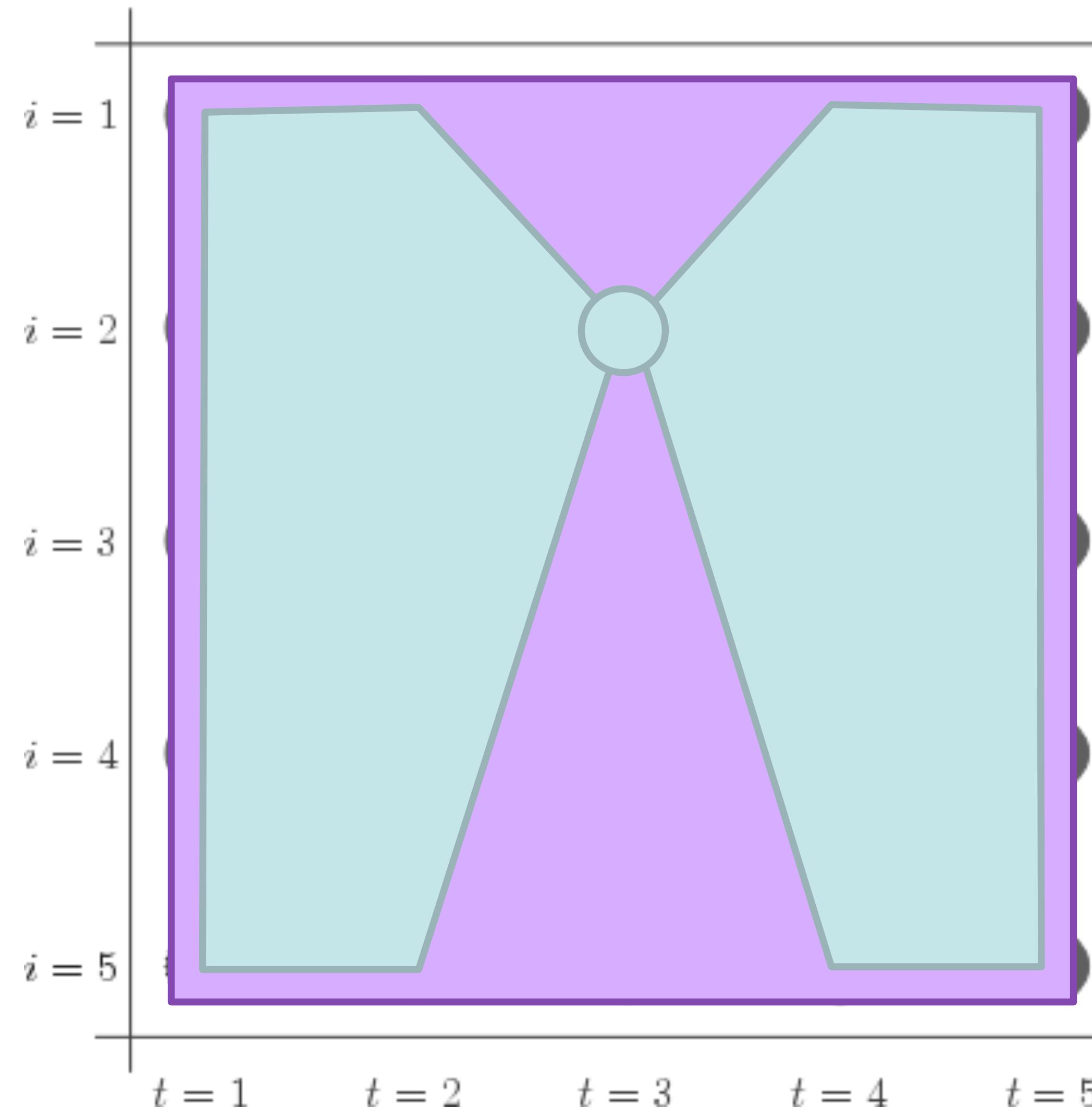


$$P(y_3 = 2 | \mathbf{x}) =$$

$$\frac{\text{sum of all paths through state 2 at time 3}}{\text{sum of all paths}}$$

# Forward-Backward Algorithm

---



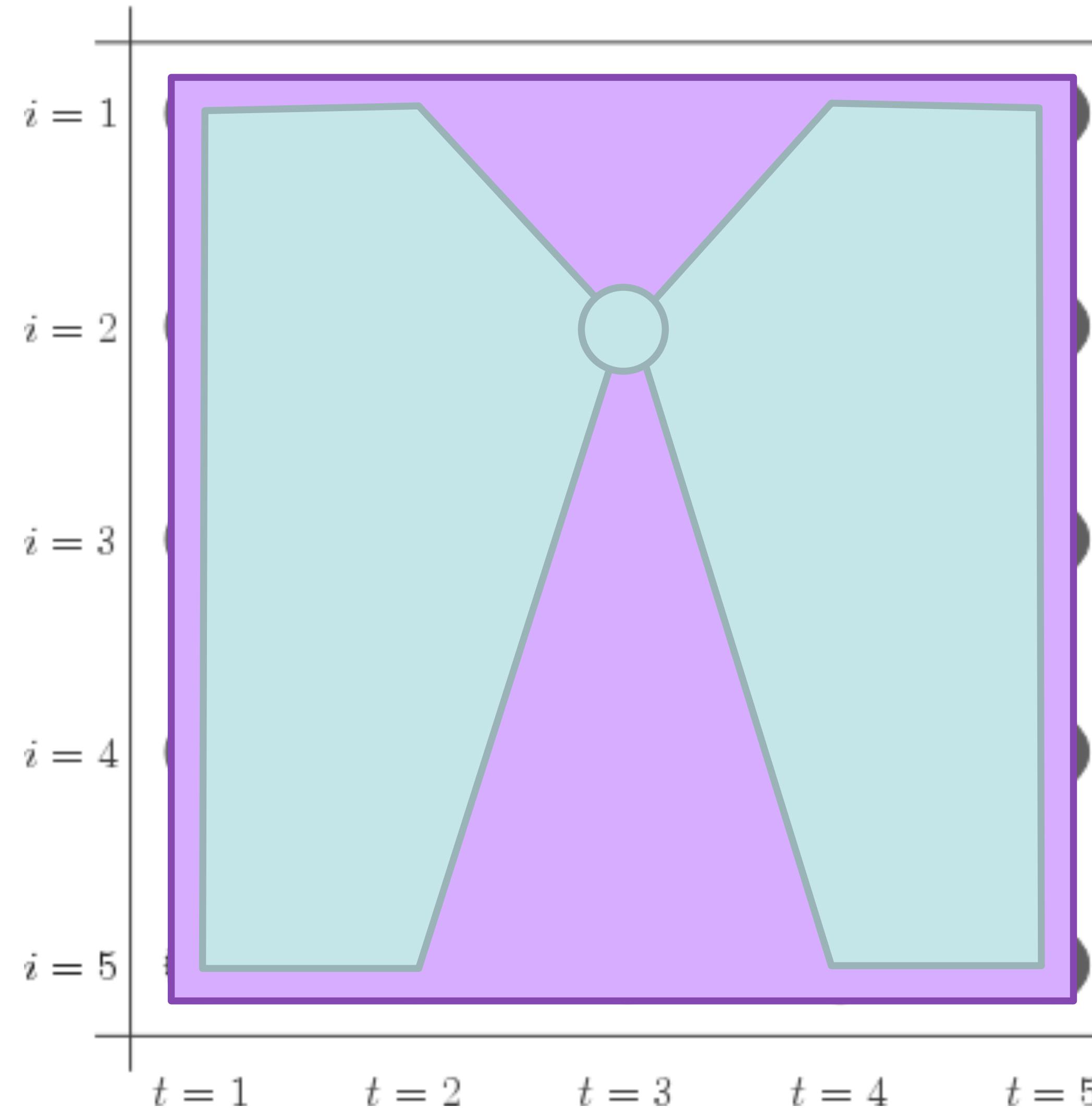
$$P(y_3 = 2 | \mathbf{x}) =$$

sum of all paths through state 2 at time 3

---

sum of all paths

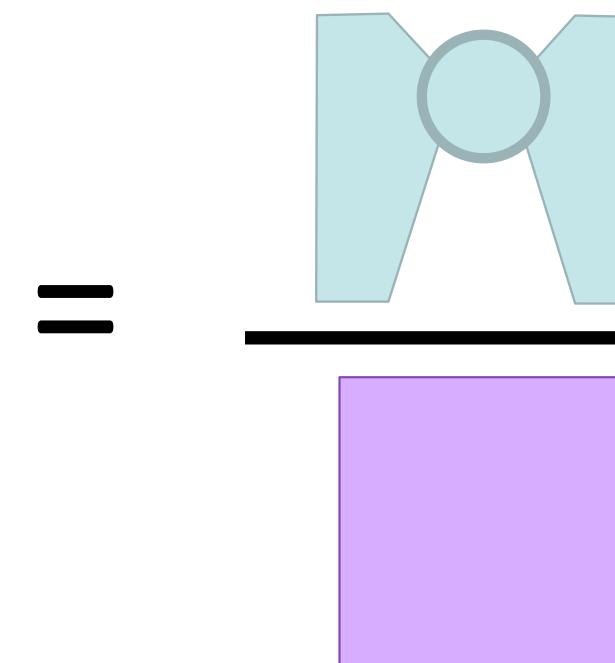
# Forward-Backward Algorithm



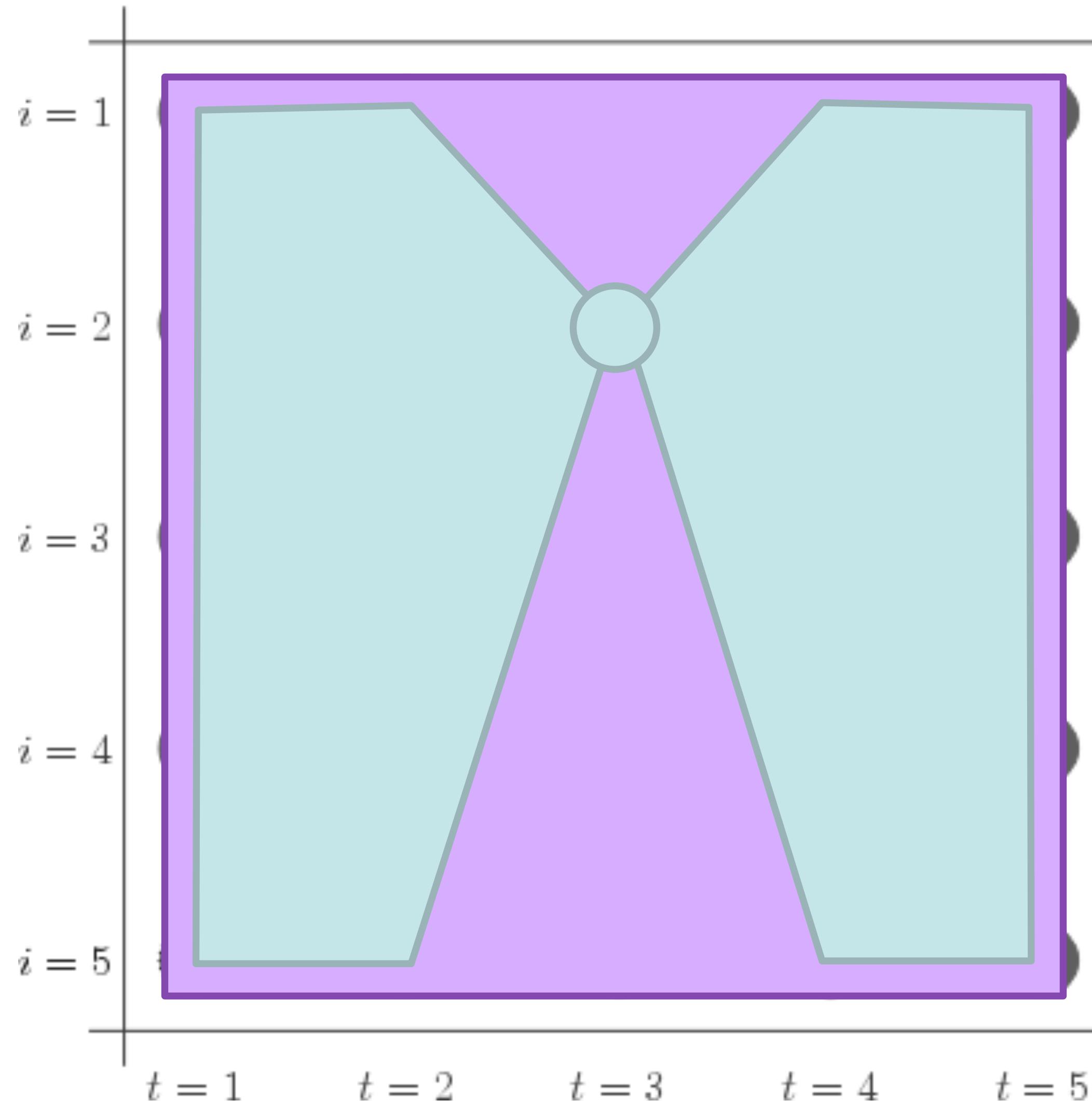
$$P(y_3 = 2 | \mathbf{x}) =$$

sum of all paths through state 2 at time 3

sum of all paths

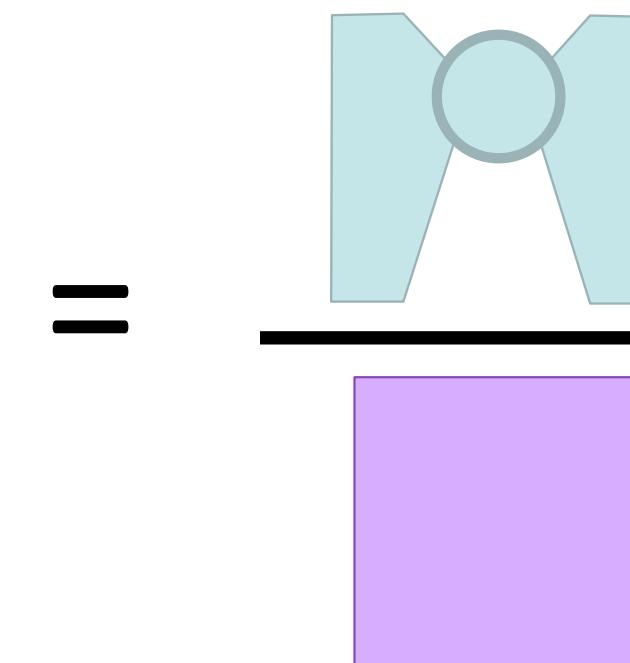


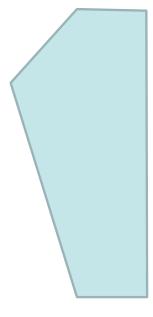
# Forward-Backward Algorithm



$$P(y_3 = 2 | \mathbf{x}) =$$

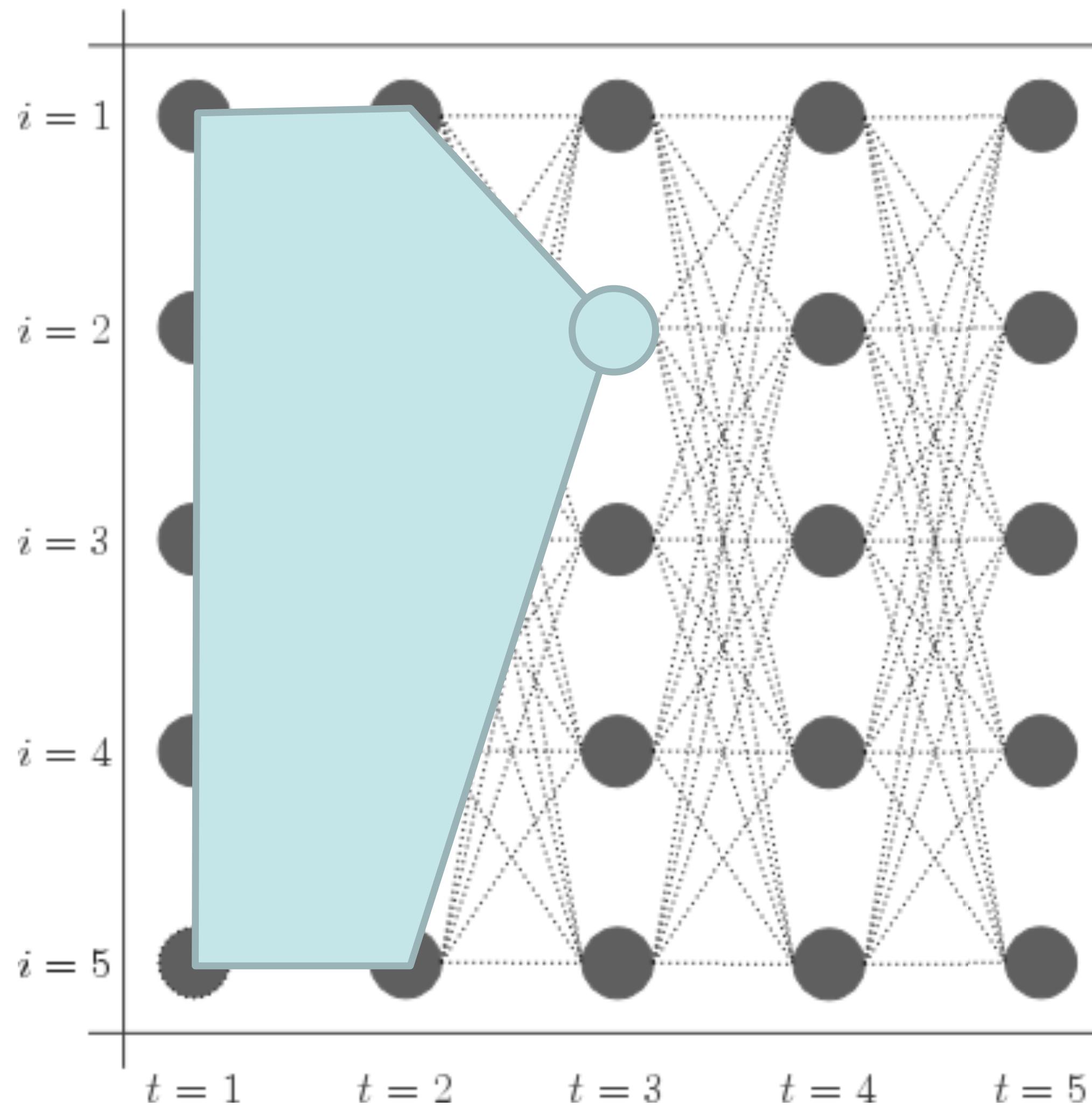
$$\frac{\text{sum of all paths through state 2 at time 3}}{\text{sum of all paths}}$$



- ▶ Easiest and most flexible to do one pass to compute  and one to compute 

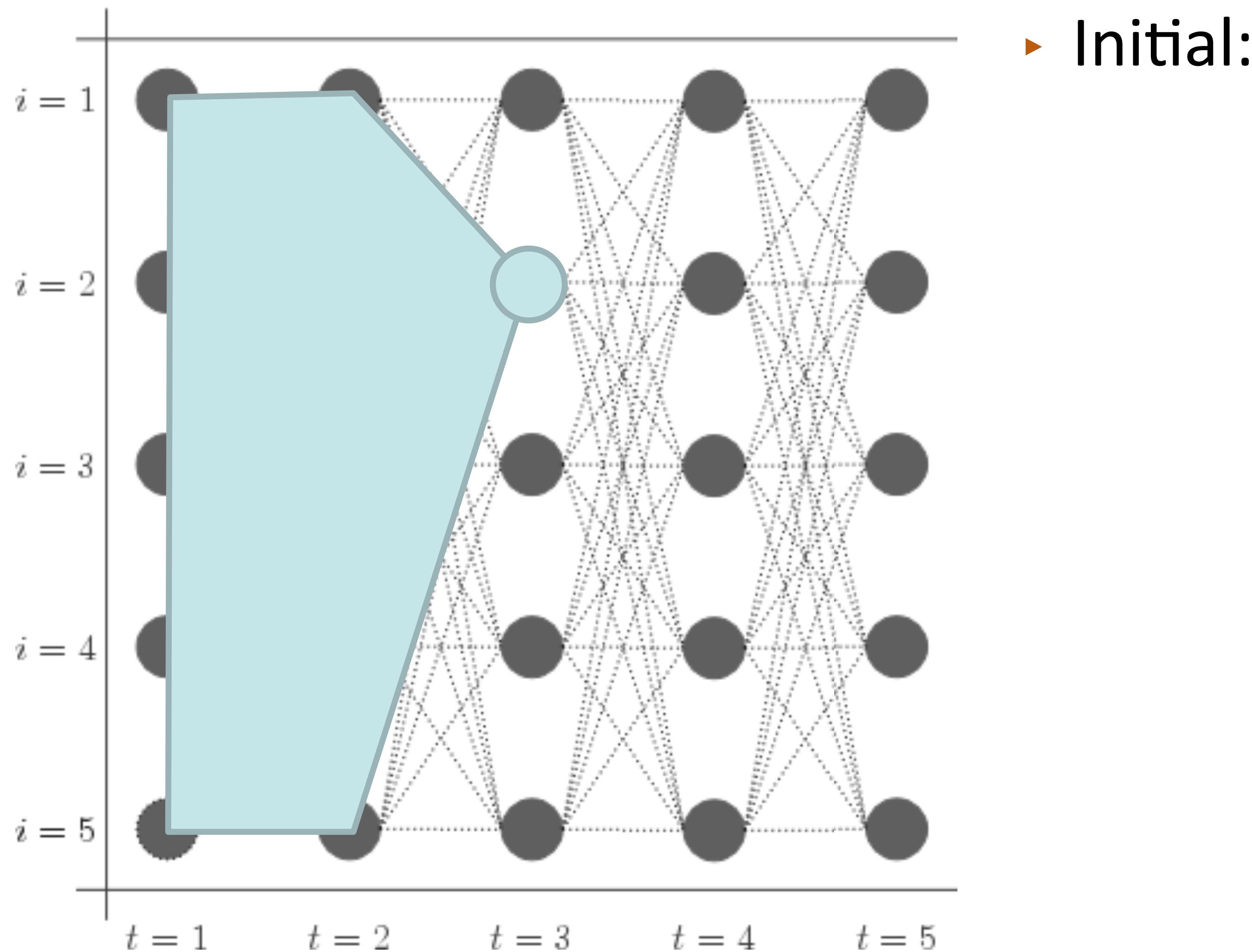
# Forward-Backward Algorithm

---



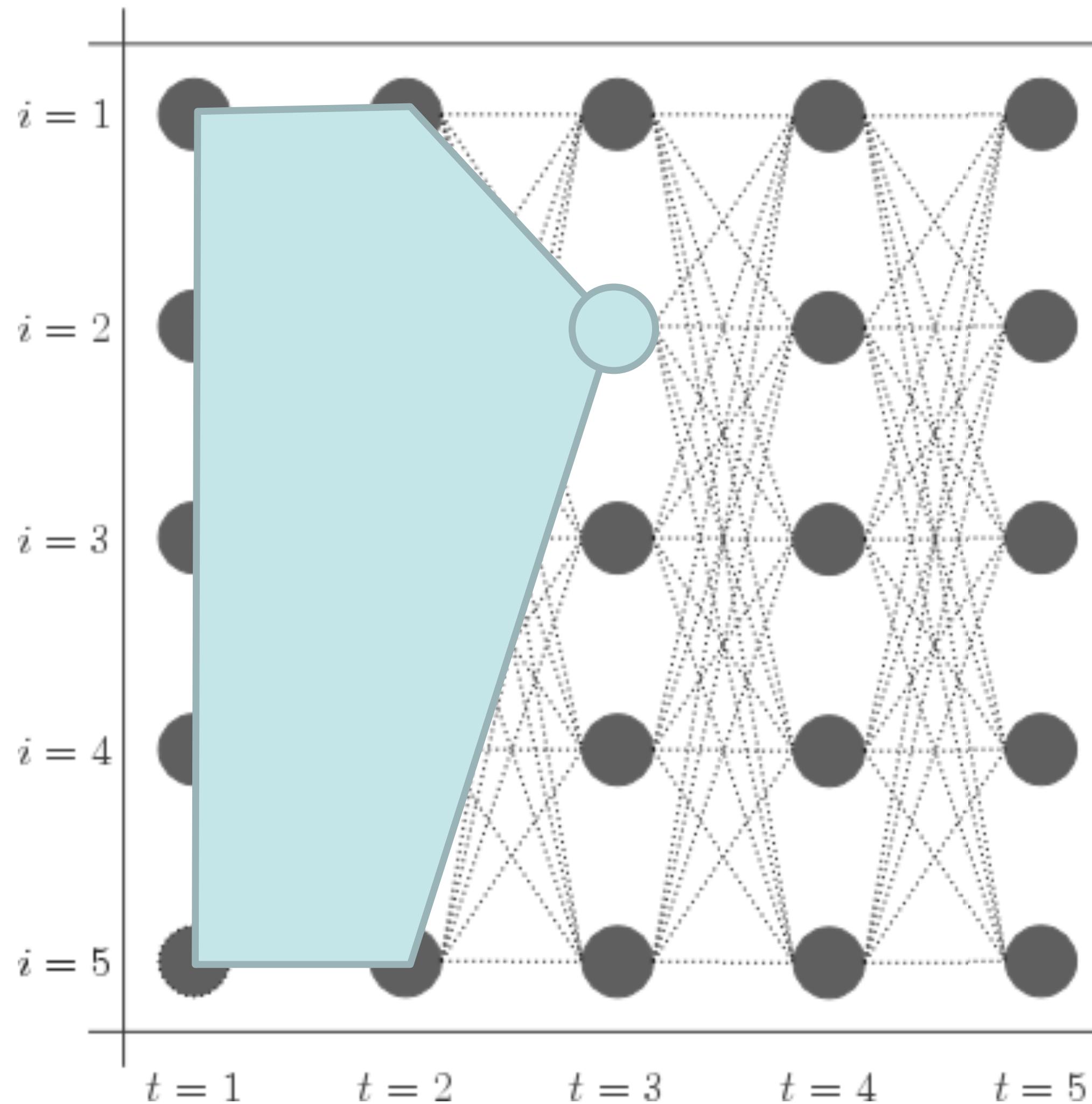
# Forward-Backward Algorithm

---



# Forward-Backward Algorithm

---

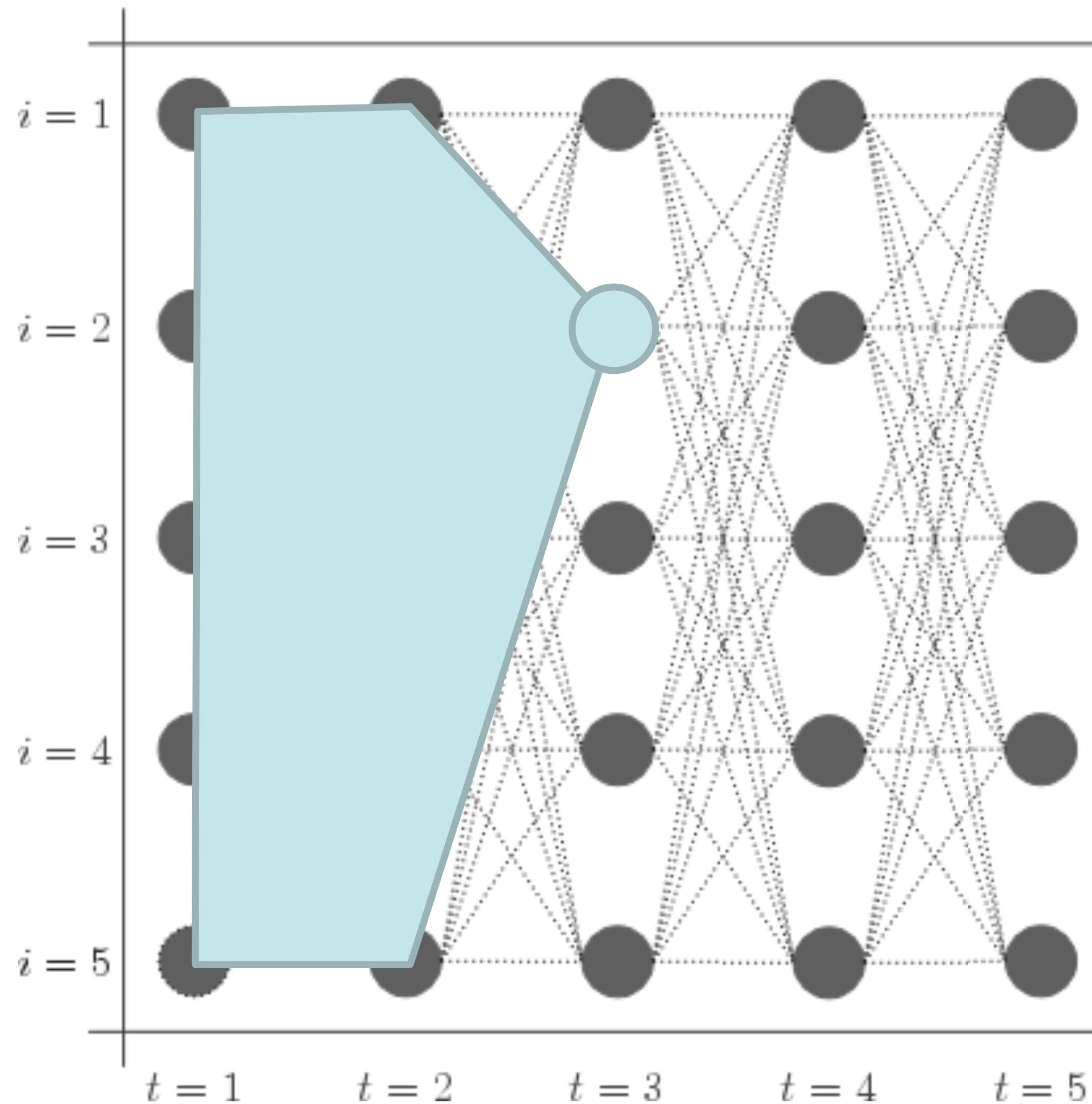


► Initial:

$$\alpha_1(s) = P(s)P(x_1|s)$$

# Forward-Backward Algorithm

---



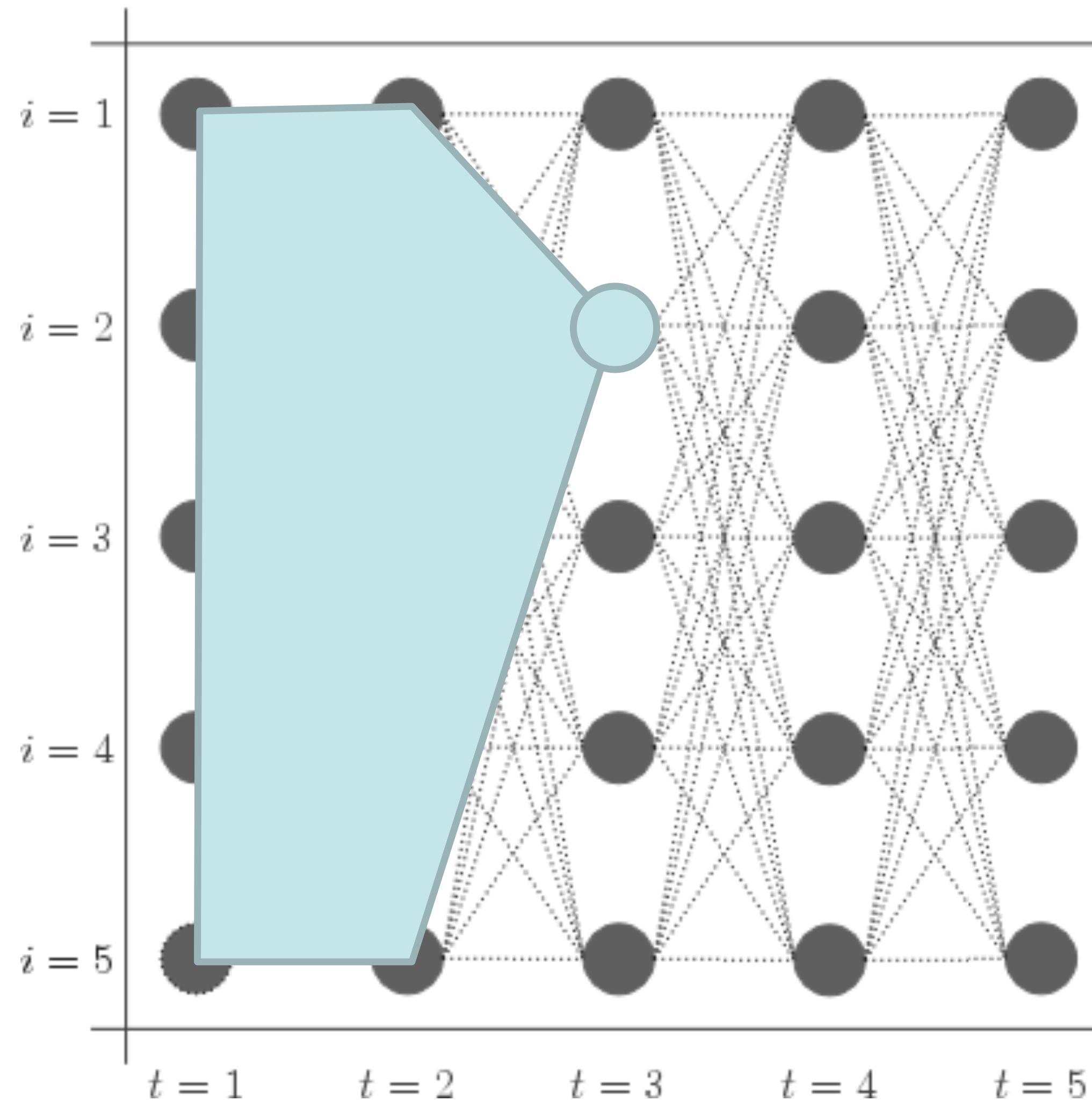
► Initial:

$$\alpha_1(s) = P(s)P(x_1|s)$$

► Recurrence:

# Forward-Backward Algorithm

---



► Initial:

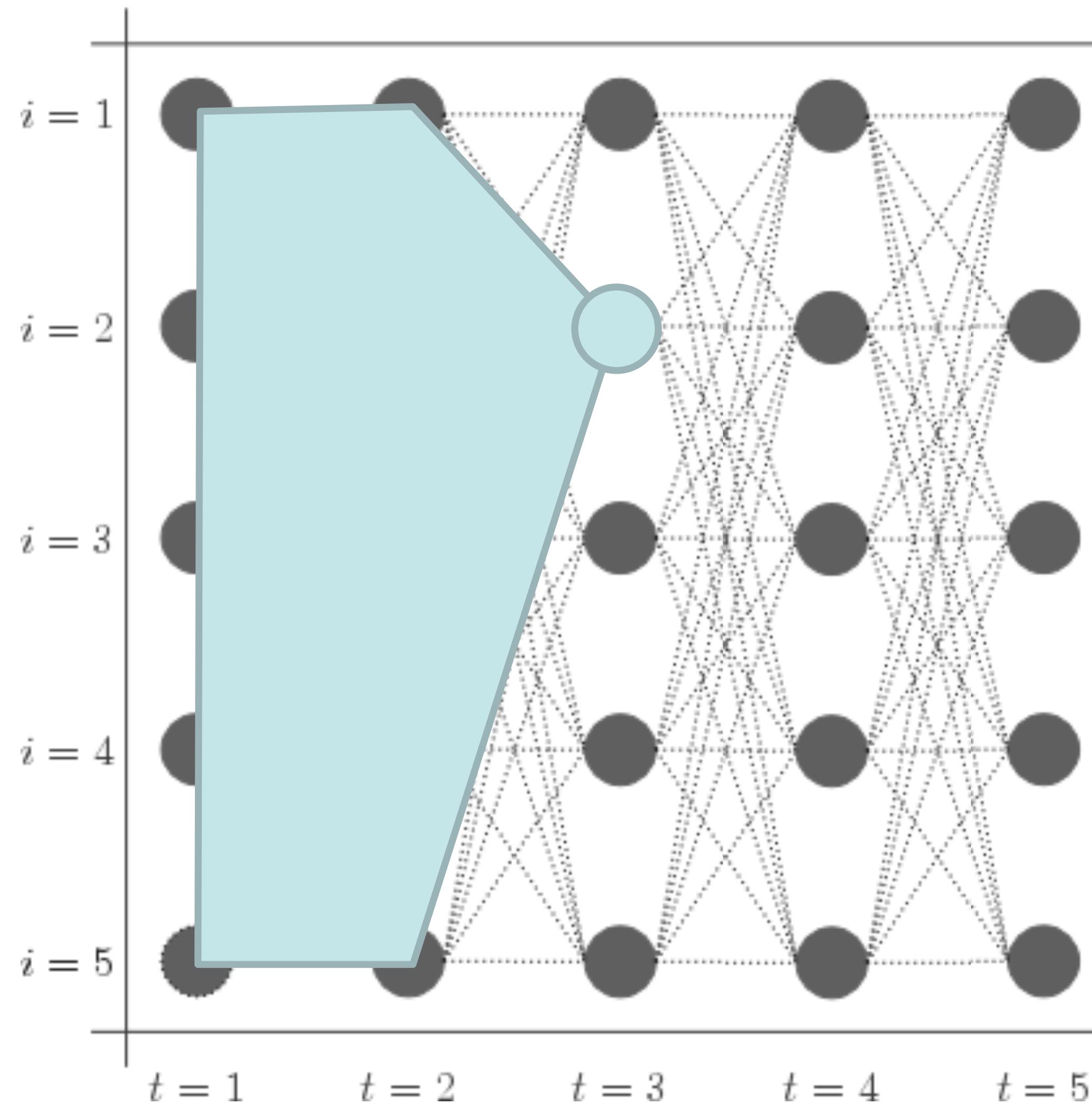
$$\alpha_1(s) = P(s)P(x_1|s)$$

► Recurrence:

$$\alpha_t(s_t) = \sum_{s_{t-1}} \alpha_{t-1}(s_{t-1})P(s_t|s_{t-1})P(x_t|s_t)$$

# Forward-Backward Algorithm

---



► Initial:

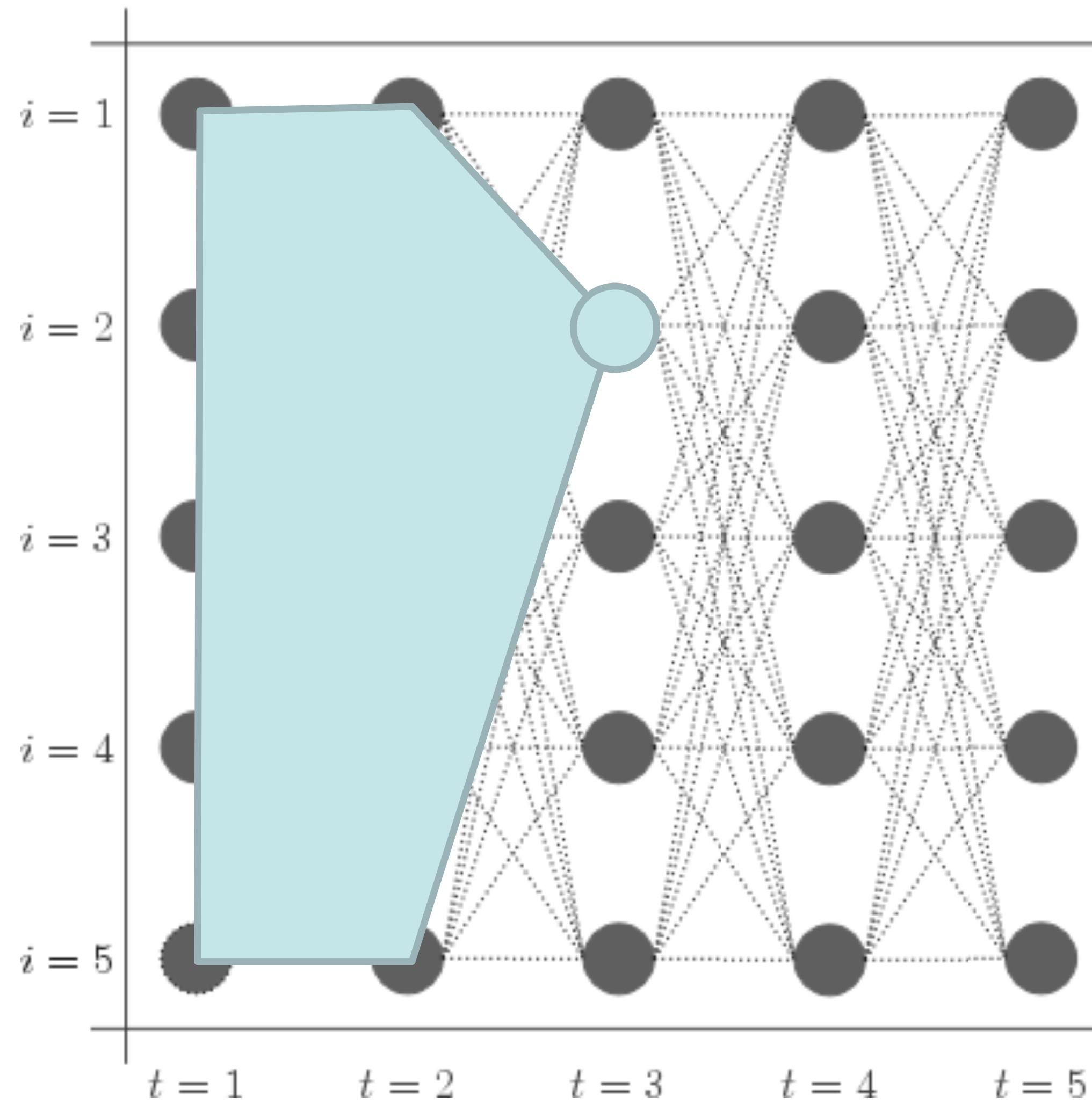
$$\alpha_1(s) = P(s)P(x_1|s)$$

► Recurrence:

$$\alpha_t(s_t) = \sum_{s_{t-1}} \alpha_{t-1}(s_{t-1})P(s_t|s_{t-1})P(x_t|s_t)$$

► Same as Viterbi but summing instead of maxing!

# Forward-Backward Algorithm



► Initial:

$$\alpha_1(s) = P(s)P(x_1|s)$$

► Recurrence:

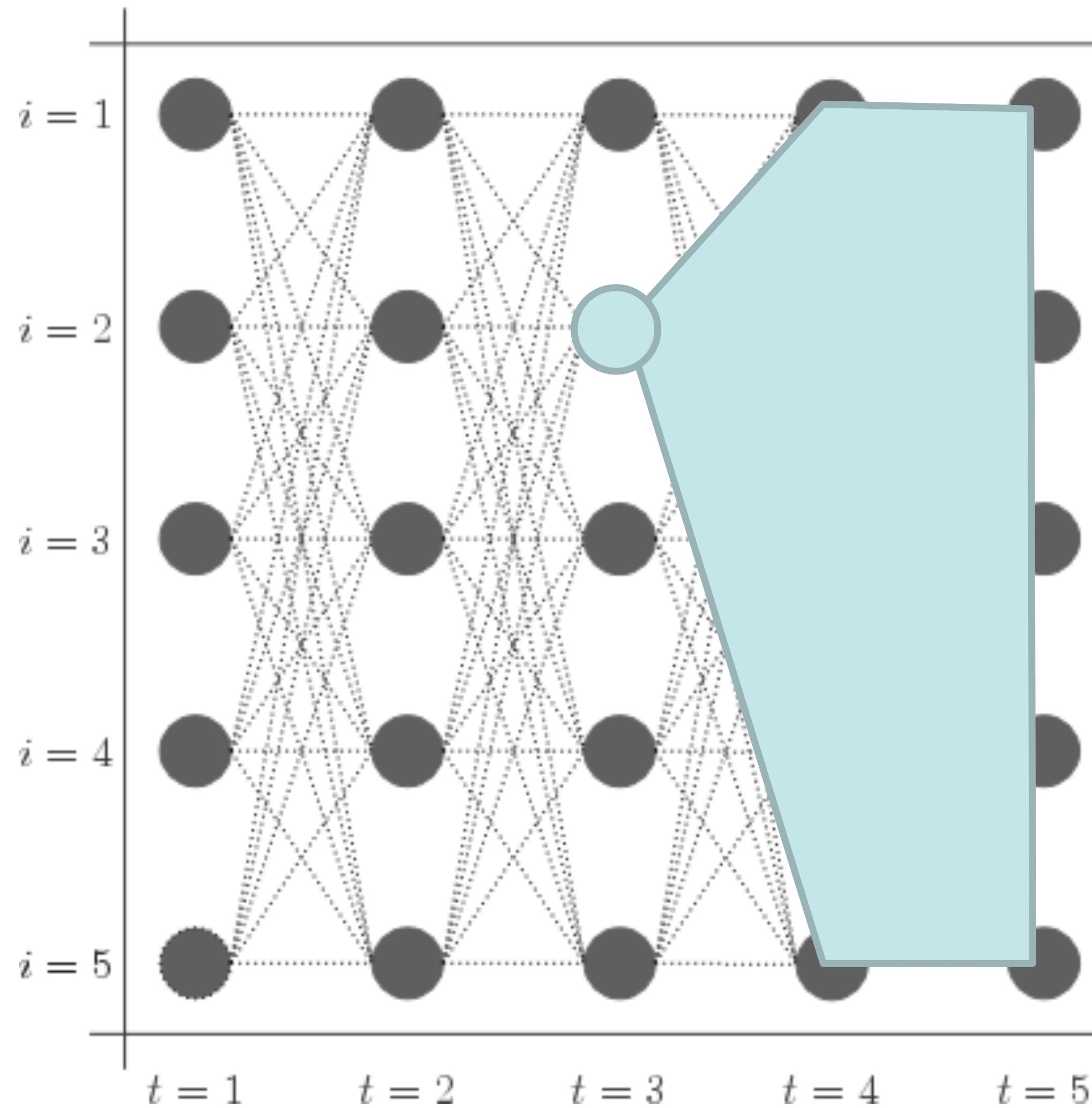
$$\alpha_t(s_t) = \sum_{s_{t-1}} \alpha_{t-1}(s_{t-1})P(s_t|s_{t-1})P(x_t|s_t)$$

► Same as Viterbi but summing instead of maxing!

► These quantities get very small!  
Store everything as log probabilities

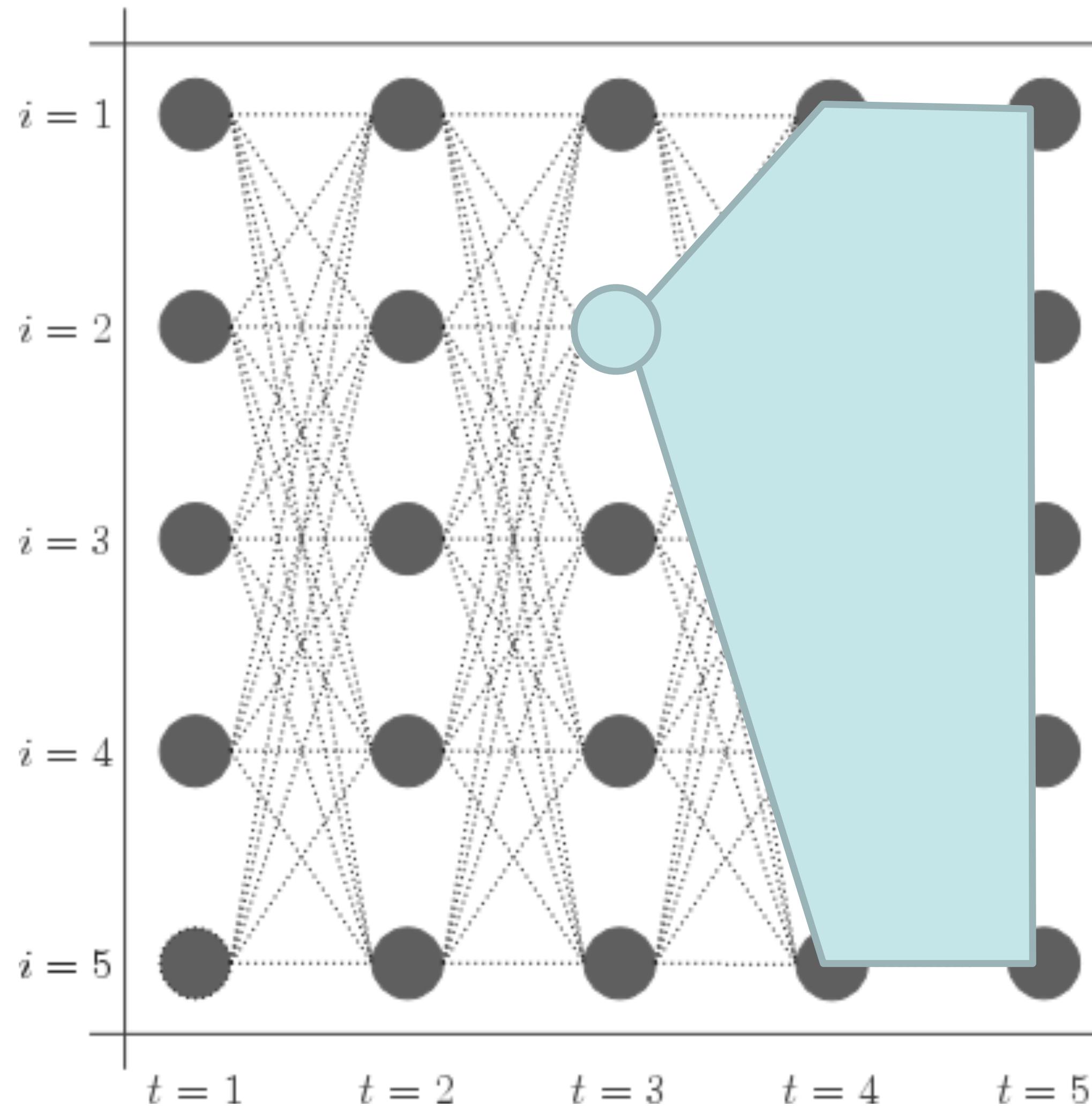
# Forward-Backward Algorithm

---



# Forward-Backward Algorithm

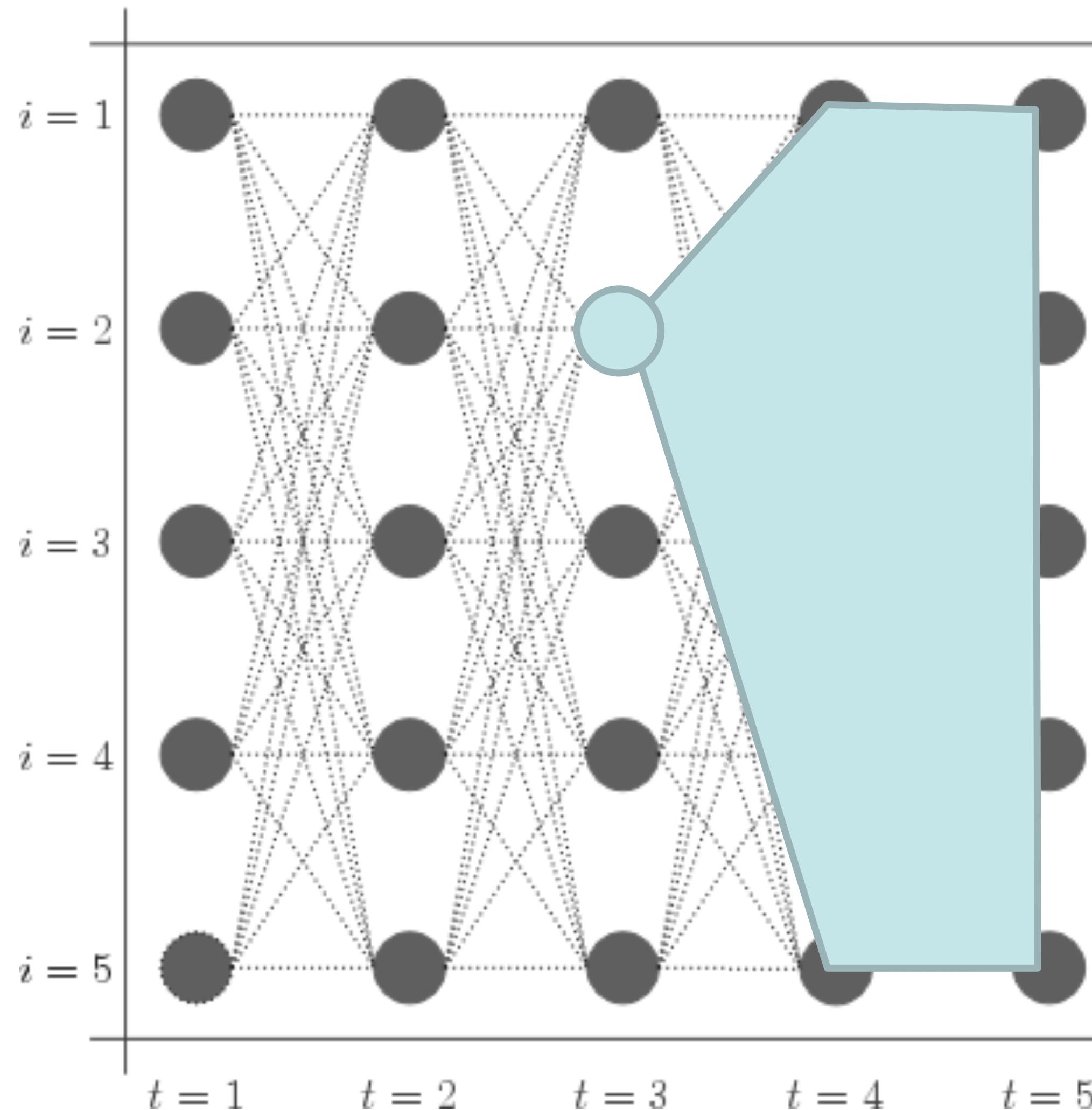
---



► Initial:

# Forward-Backward Algorithm

---

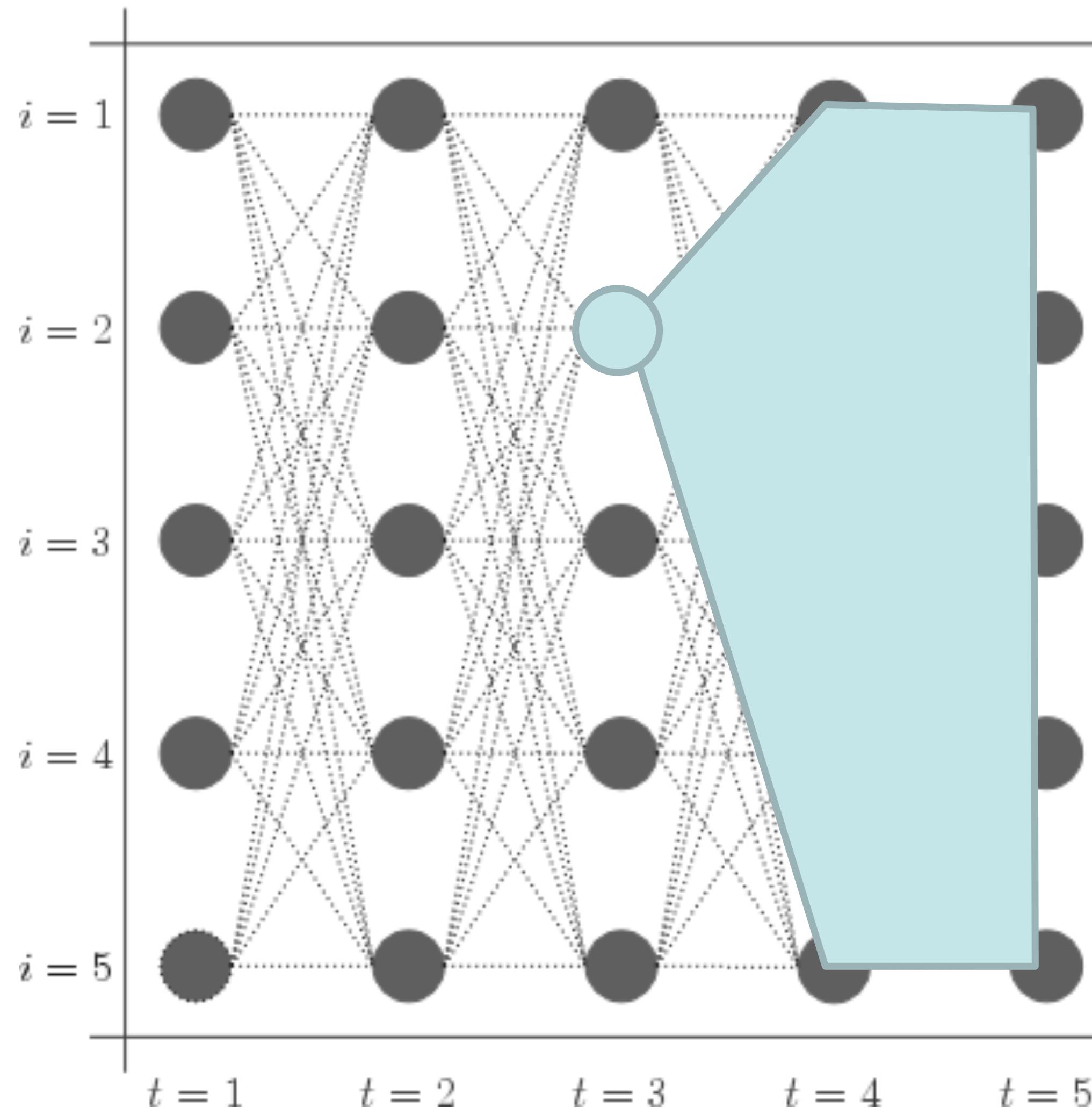


► Initial:

$$\beta_n(s) = 1$$

# Forward-Backward Algorithm

---



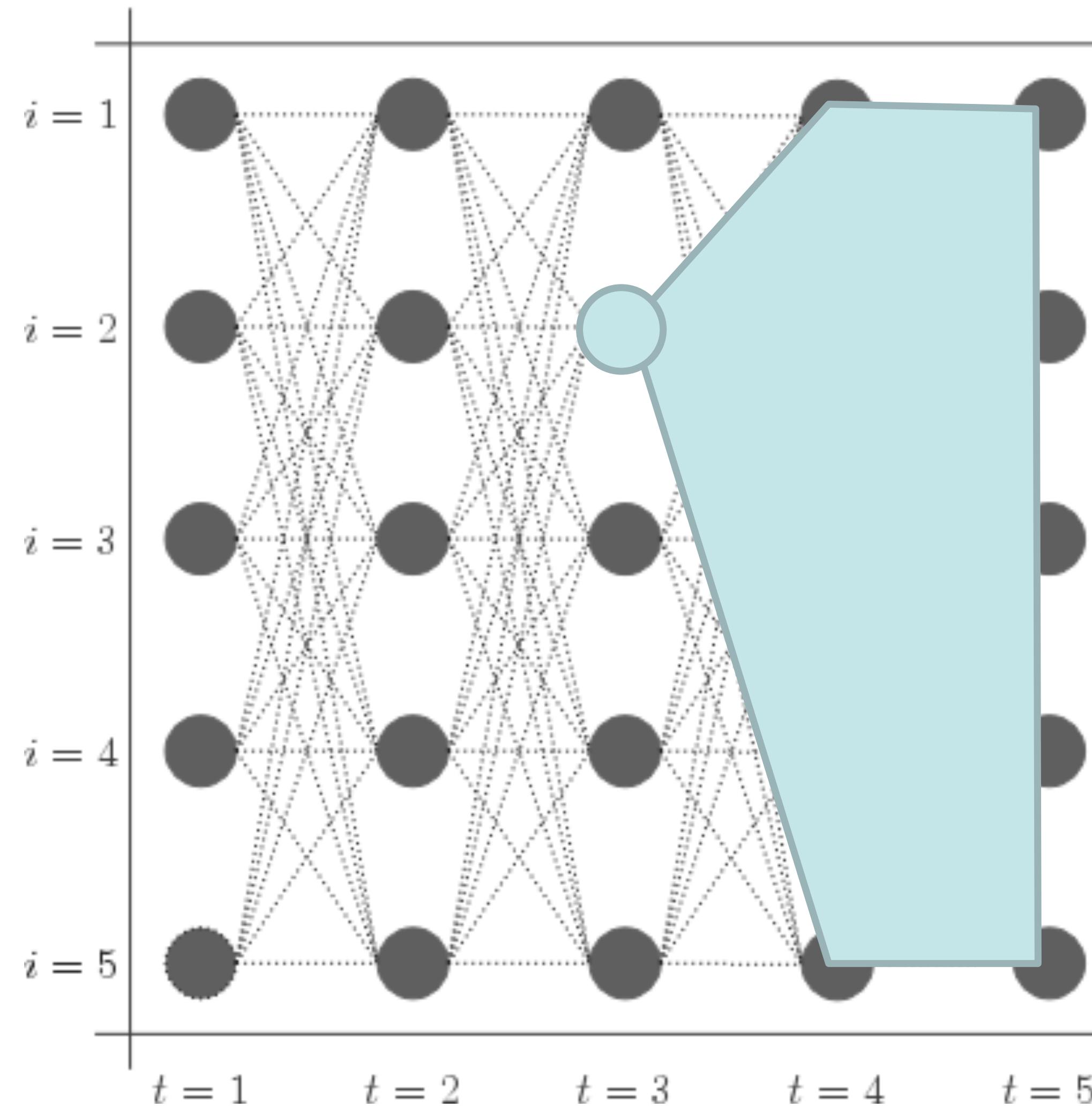
► Initial:

$$\beta_n(s) = 1$$

► Recurrence:

# Forward-Backward Algorithm

---



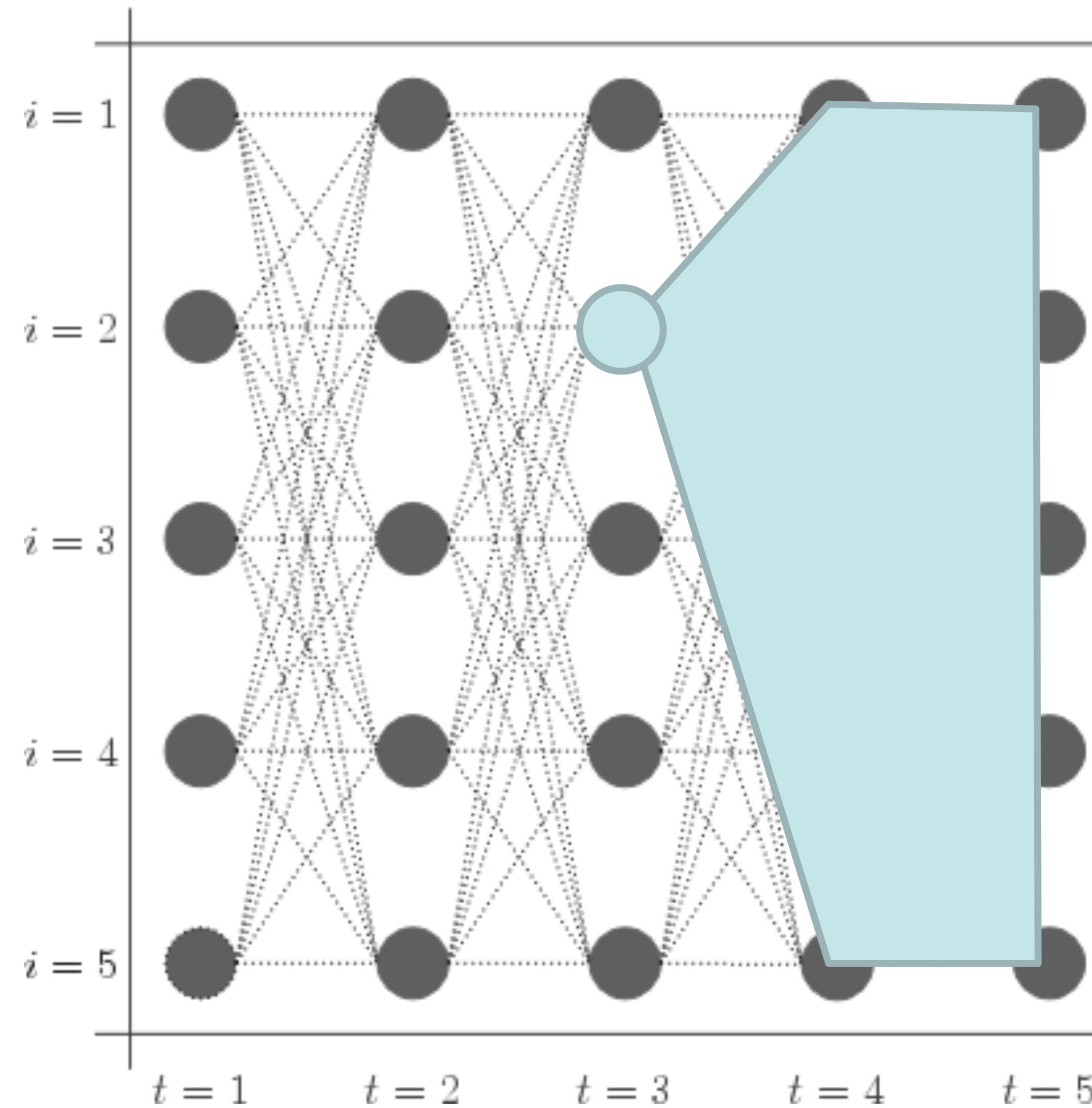
► Initial:

$$\beta_n(s) = 1$$

► Recurrence:

$$\beta_t(s_t) = \sum_{s_{t+1}} \beta_{t+1}(s_{t+1}) P(s_{t+1}|s_t) P(x_{t+1}|s_{t+1})$$

# Forward-Backward Algorithm



► Initial:

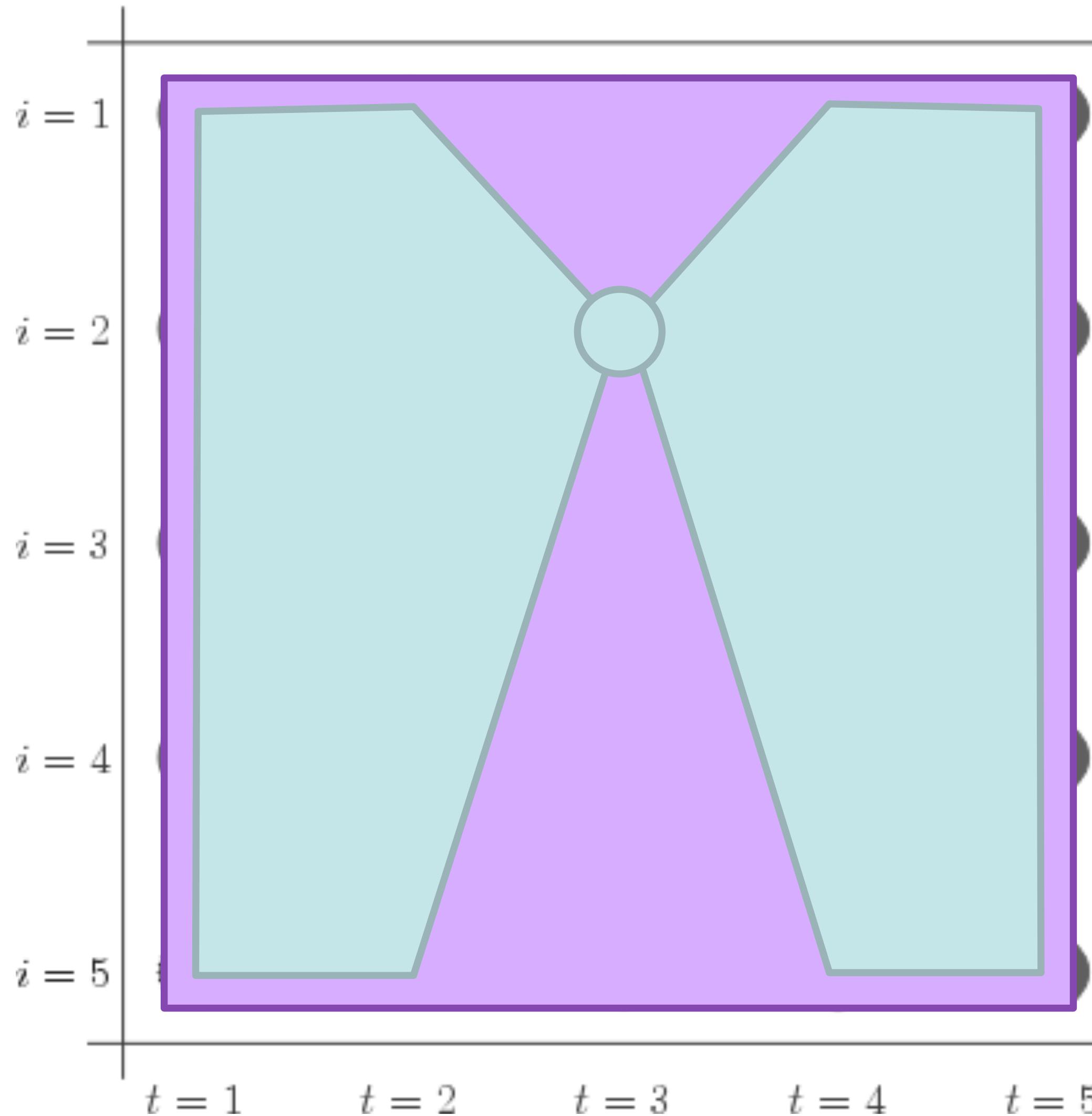
$$\beta_n(s) = 1$$

► Recurrence:

$$\beta_t(s_t) = \sum_{s_{t+1}} \beta_{t+1}(s_{t+1}) P(s_{t+1}|s_t) P(x_{t+1}|s_{t+1})$$

► Big differences: count emission for the *next timestep* (not current one)

# Forward-Backward Algorithm



$$\alpha_1(s) = P(s)P(x_1|s)$$

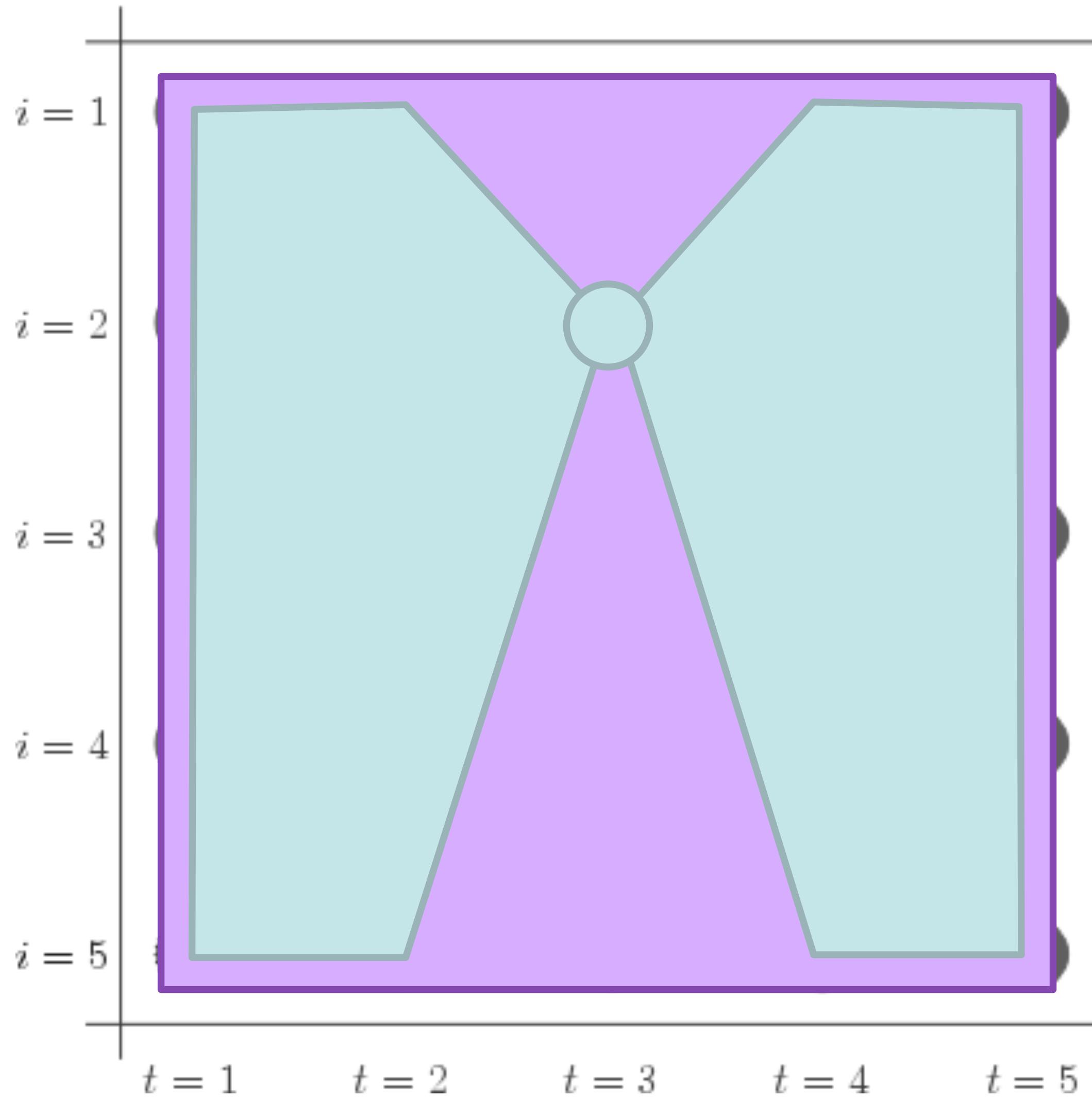
$$\alpha_t(s_t) = \sum_{s_{t-1}} \alpha_{t-1}(s_{t-1})P(s_t|s_{t-1})P(x_t|s_t)$$

$$\beta_n(s) = 1$$

$$\beta_t(s_t) = \sum_{s_{t+1}} \beta_{t+1}(s_{t+1})P(s_{t+1}|s_t)P(x_{t+1}|s_{t+1})$$

- ▶ Big differences: count emission for the *next timestep* (not current one)

# Forward-Backward Algorithm

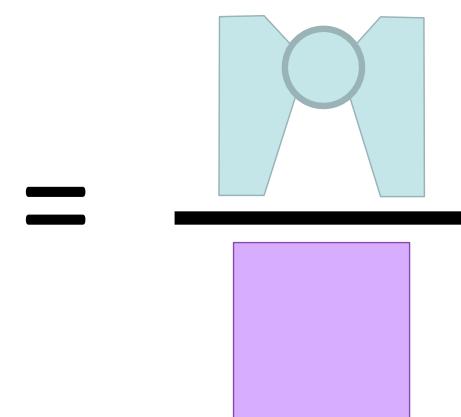


$$\alpha_1(s) = P(s)P(x_1|s)$$

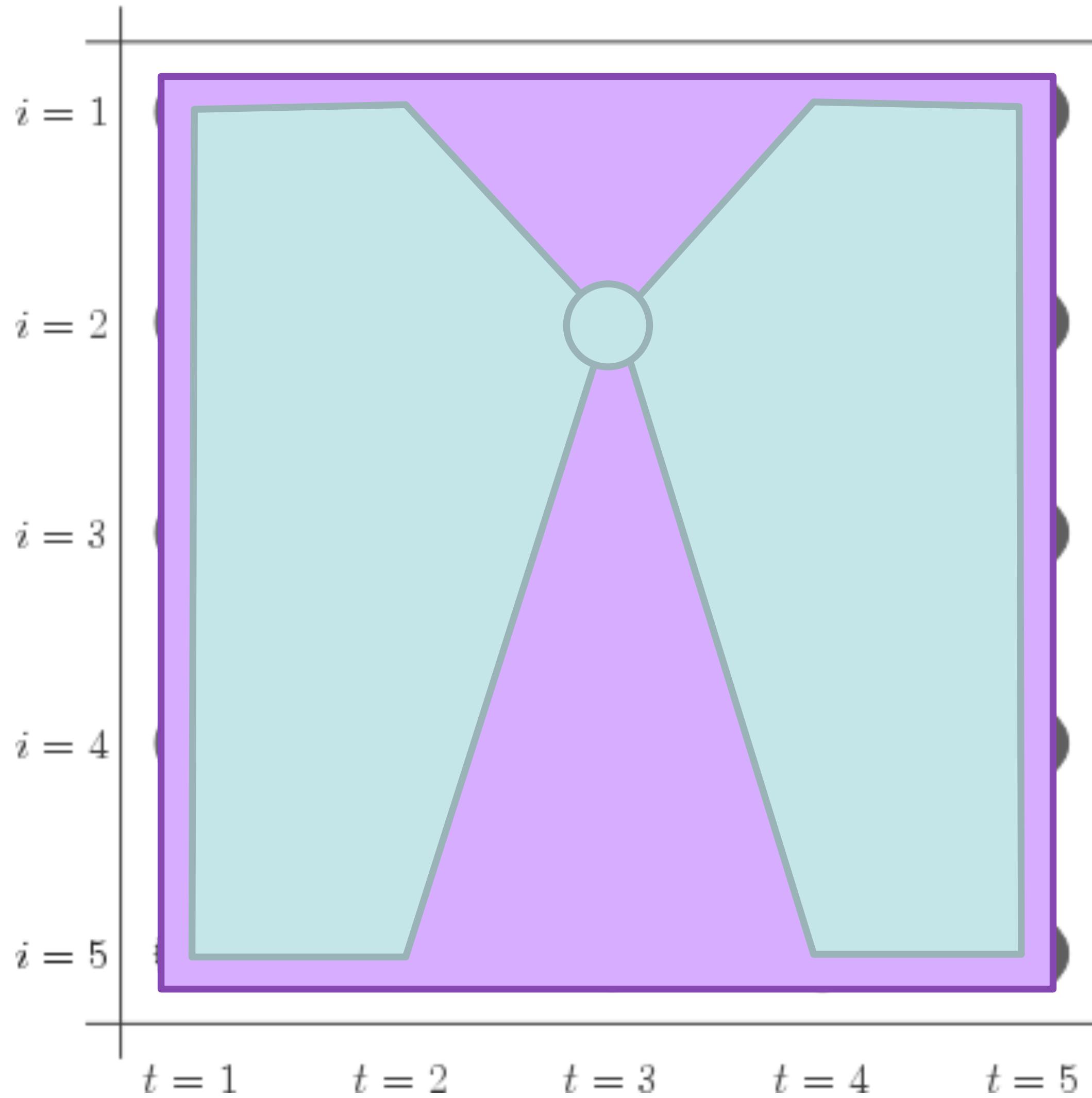
$$\alpha_t(s_t) = \sum_{s_{t-1}} \alpha_{t-1}(s_{t-1})P(s_t|s_{t-1})P(x_t|s_t)$$

$$\beta_n(s) = 1$$

$$\beta_t(s_t) = \sum_{s_{t+1}} \beta_{t+1}(s_{t+1})P(s_{t+1}|s_t)P(x_{t+1}|s_{t+1})$$



# Forward-Backward Algorithm



$$\alpha_1(s) = P(s)P(x_1|s)$$

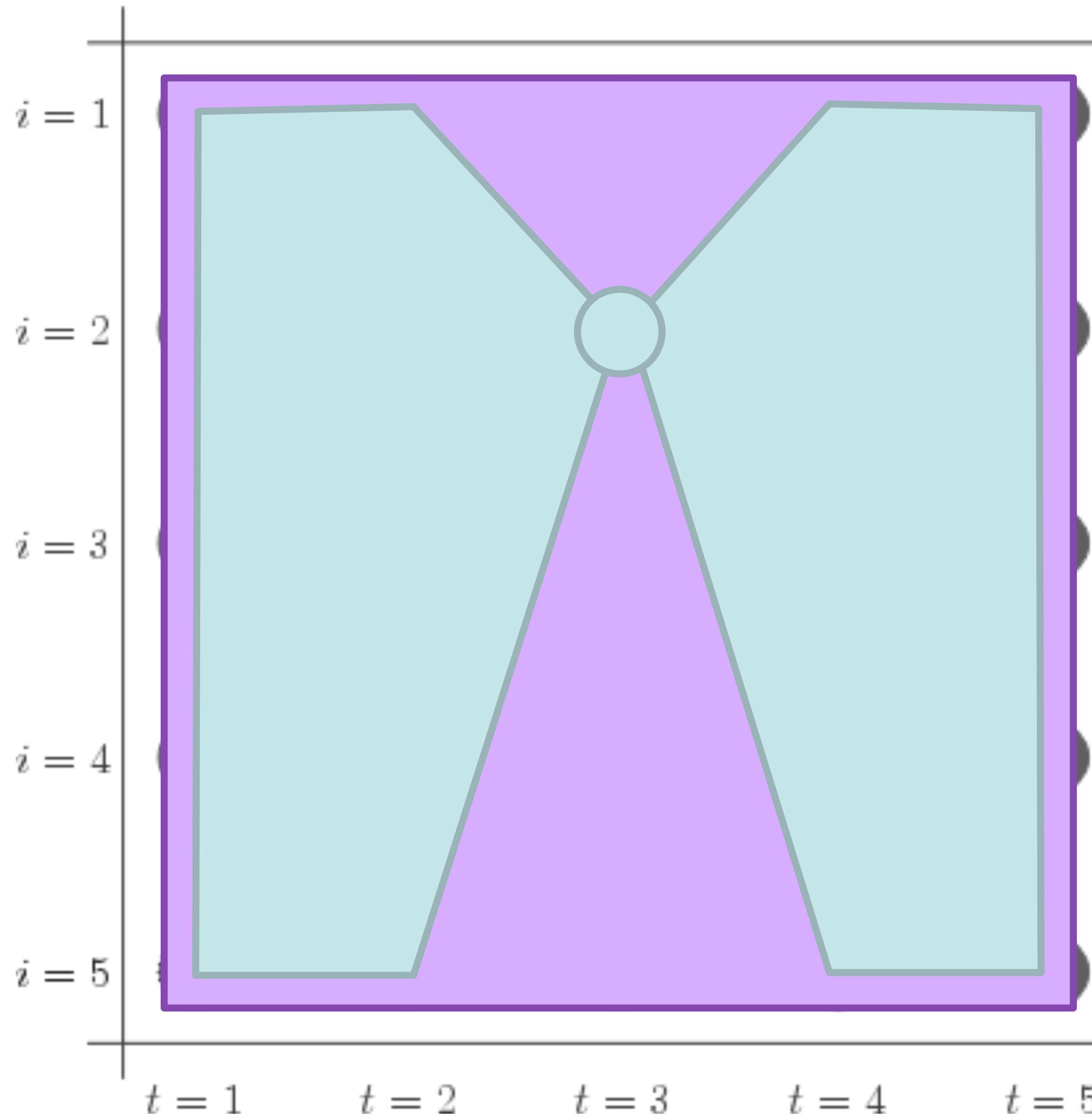
$$\alpha_t(s_t) = \sum_{s_{t-1}} \alpha_{t-1}(s_{t-1})P(s_t|s_{t-1})P(x_t|s_t)$$

$$\beta_n(s) = 1$$

$$\beta_t(s_t) = \sum_{s_{t+1}} \beta_{t+1}(s_{t+1})P(s_{t+1}|s_t)P(x_{t+1}|s_{t+1})$$

$$P(s_3 = 2 | \mathbf{x}) = \frac{\alpha_3(2)\beta_3(2)}{\sum_i \alpha_3(i)\beta_3(i)} = \frac{\text{[Diagram of a green shaded area under a purple shaded area]}}{\text{[Diagram of a purple shaded area under a green shaded area]}}$$

# Forward-Backward Algorithm



$$\alpha_1(s) = P(s)P(x_1|s)$$

$$\alpha_t(s_t) = \sum_{s_{t-1}} \alpha_{t-1}(s_{t-1})P(s_t|s_{t-1})P(x_t|s_t)$$

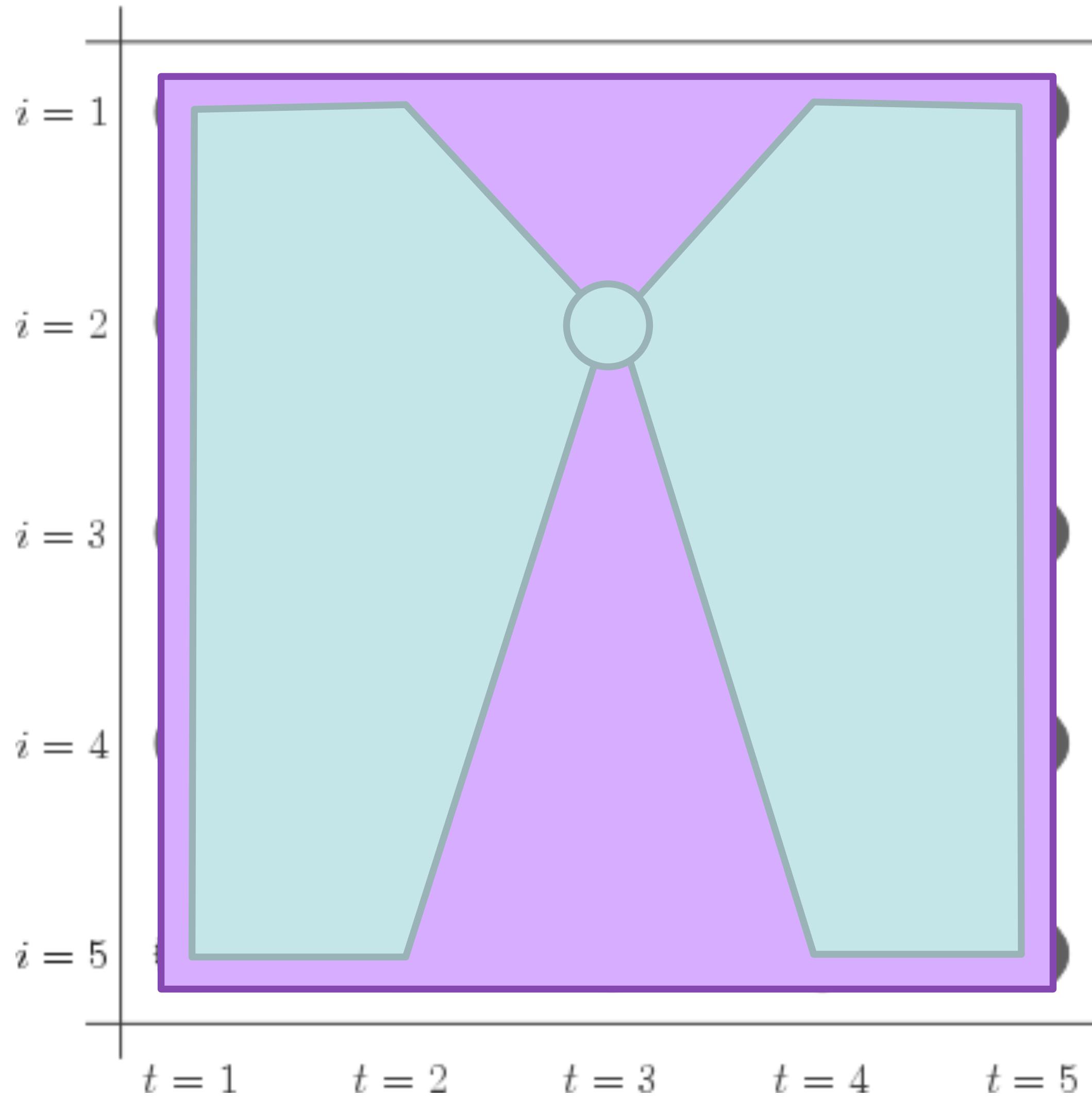
$$\beta_n(s) = 1$$

$$\beta_t(s_t) = \sum_{s_{t+1}} \beta_{t+1}(s_{t+1})P(s_{t+1}|s_t)P(x_{t+1}|s_{t+1})$$

$$P(s_3 = 2 | \mathbf{x}) = \frac{\alpha_3(2)\beta_3(2)}{\sum_i \alpha_3(i)\beta_3(i)} = \frac{\text{[Diagram showing the ratio of the highlighted green and purple areas]}}{\text{[Diagram showing the total area under the curves]}}$$

► What is the denominator here?

# Forward-Backward Algorithm



$$\alpha_1(s) = P(s)P(x_1|s)$$

$$\alpha_t(s_t) = \sum_{s_{t-1}} \alpha_{t-1}(s_{t-1})P(s_t|s_{t-1})P(x_t|s_t)$$

$$\beta_n(s) = 1$$

$$\beta_t(s_t) = \sum_{s_{t+1}} \beta_{t+1}(s_{t+1})P(s_{t+1}|s_t)P(x_{t+1}|s_{t+1})$$

$$P(s_3 = 2|\mathbf{x}) = \frac{\alpha_3(2)\beta_3(2)}{\sum_i \alpha_3(i)\beta_3(i)} = \frac{\text{[Diagram showing a light blue trapezoid divided by a vertical line, with a small circle at the top right representing } \beta_3(2)]}{\text{[Diagram showing a pink trapezoid divided by a vertical line]}}$$

► What is the denominator here?  $P(\mathbf{x})$

# HMM POS Tagging

---

# HMM POS Tagging

---

- ▶ Baseline: assign each word its most frequent tag: ~90% accuracy

# HMM POS Tagging

---

- ▶ Baseline: assign each word its most frequent tag: ~90% accuracy
- ▶ Trigram HMM: ~95% accuracy / 55% on unknown words

# Trigram Taggers

---

NNP VBZ NN NNS CD NN

Fed raises interest rates 0.5 percent

# Trigram Taggers

---

NNP VBZ NN NNS CD NN

Fed raises interest rates 0.5 percent

- ▶ Trigram model:  $y_1 = (<S>, \text{NNP})$ ,  $y_2 = (\text{NNP}, \text{VBZ})$ , ...

# Trigram Taggers

---

NNP VBZ NN NNS CD NN

Fed raises interest rates 0.5 percent

- ▶ Trigram model:  $y_1 = (<S>, \text{NNP})$ ,  $y_2 = (\text{NNP}, \text{VBZ})$ , ...
- ▶  $P((\text{VBZ}, \text{NN}) \mid (\text{NNP}, \text{VBZ}))$  — more context! Noun-verb-noun S-V-O

# Trigram Taggers

---

NNP VBZ NN NNS CD NN

Fed raises interest rates 0.5 percent

- ▶ Trigram model:  $y_1 = (<S>, \text{NNP})$ ,  $y_2 = (\text{NNP}, \text{VBZ})$ , ...
- ▶  $P((\text{VBZ}, \text{NN}) \mid (\text{NNP}, \text{VBZ}))$  — more context! Noun-verb-noun S-V-O
- ▶ Tradeoff between model capacity and data size — trigrams are a “sweet spot” for POS tagging

# HMM POS Tagging

---

- ▶ Baseline: assign each word its most frequent tag: ~90% accuracy
- ▶ Trigram HMM: ~95% accuracy / 55% on unknown words

# HMM POS Tagging

---

- ▶ Baseline: assign each word its most frequent tag: ~90% accuracy
- ▶ Trigram HMM: ~95% accuracy / 55% on unknown words
- ▶ TnT tagger (Brants 1998, tuned HMM): 96.2% accuracy / 86.0% on unks

# HMM POS Tagging

---

- ▶ Baseline: assign each word its most frequent tag: ~90% accuracy
- ▶ Trigram HMM: ~95% accuracy / 55% on unknown words
- ▶ TnT tagger (Brants 1998, tuned HMM): 96.2% accuracy / 86.0% on unks
- ▶ State-of-the-art (BiLSTM-CRFs): 97.5% / 89%+

# Errors

---

	JJ	NN	NNP	NNPS	RB	RP	IN	VB	VBD	VBN	VBP	Total
JJ	0	177	56	0	61	2	5	10	15	108	0	488
NN	244	0	103	0	12	1	1	29	5	6	19	525
NNP	107	106	0	132	5	0	7	5	1	2	0	427
NNPS	1	0	110	0	0	0	0	0	0	0	0	142
RB	72	21	7	0	0	16	138	1	0	0	0	295
RP	0	0	0	0	39	0	65	0	0	0	0	104
IN	11	0	1	0	169	103	0	1	0	0	0	323
VB	17	64	9	0	2	0	1	0	4	7	85	189
VBD	10	5	3	0	0	0	0	3	0	143	2	166
VBN	101	3	3	0	0	0	0	3	108	0	1	221
VBP	5	34	3	1	1	0	2	49	6	3	0	104
Total	626	536	348	144	317	122	279	102	140	269	108	3651

# Errors

	JJ	NN	NNP	NNPS	RB	RP	IN	VB	VBD	VBN	VBP	Total
JJ	0	<b>177</b>	<b>56</b>	0	<b>61</b>	2	5	10	15	<b>108</b>	0	488
NN	<b>244</b>	0	<b>103</b>	0	12	1	1	29	5	6	19	525
NNP	<b>107</b>	<b>106</b>	0	<b>132</b>	5	0	7	5	1	2	0	427
NNPS	1	0	<b>110</b>	0	0	0	0	0	0	0	0	142
RB	<b>72</b>	21	7	0	0	16	<b>138</b>	1	0	0	0	295
RP	0	0	0	0	<b>39</b>	0	<b>65</b>	0	0	0	0	104
IN	11	0	1	0	<b>169</b>	<b>103</b>	0	1	0	0	0	323
VB	17	<b>64</b>	9	0	2	0	1	0	4	7	<b>85</b>	189
VBD	10	5	3	0	0	0	0	3	0	<b>143</b>	2	166
VBN	<b>101</b>	3	3	0	0	0	0	3	<b>108</b>	0	1	221
VBP	5	34	3	1	1	0	2	<b>49</b>	6	3	0	104
Total	626	536	348	144	317	122	279	102	140	269	108	3651

JJ/**NN**      NN  
*official knowledge*

# Errors

	JJ	NN	NNP	NNPS	RB	RP	IN	VB	VBD	VBN	VBP	Total
JJ	0	177	56	0	61	2	5	10	15	108	0	488
NN	244	0	103	0	12	1	1	29	5	6	19	525
NNP	107	106	0	132	5	0	7	5	1	2	0	427
NNPS	1	0	110	0	0	0	0	0	0	0	0	142
RB	72	21	7	0	0	16	138	1	0	0	0	295
RP	0	0	0	0	39	0	65	0	0	0	0	104
IN	11	0	1	0	169	103	0	1	0	0	0	323
VB	17	64	9	0	2	0	1	0	4	7	85	189
VBD	10	5	3	0	0	0	0	3	0	143	2	166
VBN	101	3	3	0	0	0	0	3	108	0	1	221
VBP	5	34	3	1	1	0	2	49	6	3	0	104
Total	626	536	348	144	317	122	279	102	140	269	108	3651

JJ/**NN**      NN  
*official knowledge*

(NN NN: *tax cut, art gallery, ...*)

Slide credit: Dan Klein / Toutanova + Manning (2000)

# Errors

	JJ	NN	NNP	NNPS	RB	RP	IN	VB	VBD	VBN	VBP	Total
JJ	0	177	56	0	61	2	5	10	15	108	0	488
NN	244	0	103	0	12	1	1	29	5	6	19	525
NNP	107	106	0	132	5	0	7	5	1	2	0	427
NNPS	1	0	110	0	0	0	0	0	0	0	0	142
RB	72	21	7	0	0	16	138	1	0	0	0	295
RP	0	0	0	0	39	0	65	0	0	0	0	104
IN	11	0	1	0	169	103	0	1	0	0	0	323
VB	17	64	9	0	2	0	1	0	4	7	85	189
VBD	10	5	3	0	0	0	0	3	0	143	2	166
VBN	101	3	3	0	0	0	0	3	108	0	1	221
VBP	5	34	3	1	1	0	2	49	6	3	0	104
Total	626	536	348	144	317	122	279	102	140	269	108	3651

JJ/**NN**      NN  
*official knowledge*

VBD RP/**IN** DT NN  
*made up the story*

(NN NN: *tax cut, art gallery, ...*)

Slide credit: Dan Klein / Toutanova + Manning (2000)

# Errors

---

	JJ	NN	NNP	NNPS	RB	RP	IN	VB	VBD	VBN	VBP	Total
JJ	0	177	56	0	61	2	5	10	15	108	0	488
NN	244	0	103	0	12	1	1	29	5	6	19	525
NNP	107	106	0	132	5	0	7	5	1	2	0	427
NNPS	1	0	110	0	0	0	0	0	0	0	0	142
RB	72	21	7	0	0	16	138	1	0	0	0	295
RP	0	0	0	0	39	0	65	0	0	0	0	104
IN	11	0	1	0	169	103	0	1	0	0	0	323
VB	17	64	9	0	2	0	1	0	4	7	85	189
VBD	10	5	3	0	0	0	0	3	0	143	2	166
VBN	101	3	3	0	0	0	0	3	108	0	1	221
VBP	5	34	3	1	1	0	2	49	6	3	0	104
Total	626	536	348	144	317	122	279	102	140	269	108	3651

JJ/**NN**      NN  
*official knowledge*

VBD RP/**IN** DT NN  
*made up the story*

RB      VBD/**VBN** NNS  
*recently sold shares*

(NN NN: *tax cut, art gallery, ...*)

# Remaining Errors

---

Manning 2011 “Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?”

# Remaining Errors

---

- ▶ Lexicon gap (word not seen with that tag in training) 4.5%

# Remaining Errors

---

- ▶ Lexicon gap (word not seen with that tag in training) 4.5%
- ▶ Unknown word: 4.5%

# Remaining Errors

---

- ▶ Lexicon gap (word not seen with that tag in training) 4.5%
- ▶ Unknown word: 4.5%
- ▶ Could get right: 16% (many of these involve parsing!)

# Remaining Errors

---

- ▶ Lexicon gap (word not seen with that tag in training) 4.5%
- ▶ Unknown word: 4.5%
- ▶ Could get right: 16% (many of these involve parsing!)
- ▶ Difficult linguistics: 20%

# Remaining Errors

---

- ▶ Lexicon gap (word not seen with that tag in training) 4.5%
- ▶ Unknown word: 4.5%
- ▶ Could get right: 16% (many of these involve parsing!)
- ▶ Difficult linguistics: 20%

VBD / VBP? (past or present?)

*They set up absurd situations, detached from reality*

# Remaining Errors

---

- ▶ Lexicon gap (word not seen with that tag in training) 4.5%
- ▶ Unknown word: 4.5%
- ▶ Could get right: 16% (many of these involve parsing!)
- ▶ Difficult linguistics: 20%

VBD / VBP? (past or present?)

*They set up absurd situations, detached from reality*

- ▶ Underspecified / unclear, gold standard inconsistent / wrong: 58%

# Remaining Errors

---

- ▶ Lexicon gap (word not seen with that tag in training) 4.5%
- ▶ Unknown word: 4.5%
- ▶ Could get right: 16% (many of these involve parsing!)
- ▶ Difficult linguistics: 20%

VBD / VBP? (past or present?)

*They set up absurd situations, detached from reality*

- ▶ Underspecified / unclear, gold standard inconsistent / wrong: 58%

adjective or verbal participle? JJ / VBN?

*a \$ 10 million fourth-quarter charge against discontinued operations*

# Other Languages

---

Language	Source	# Tags	O/O	U/U	O/U
Arabic	PADT/CoNLL07 (Hajič et al., 2004)	21	96.1	96.9	97.0
Basque	Basque3LB/CoNLL07 (Aduriz et al., 2003)	64	89.3	93.7	93.7
Bulgarian	BTB/CoNLL06 (Simov et al., 2002)	54	95.7	97.5	97.8
Catalan	CESS-ECE/CoNLL07 (Martí et al., 2007)	54	98.5	98.2	98.8
Chinese	Penn ChineseTreebank 6.0 (Palmer et al., 2007)	34	91.7	93.4	94.1
Chinese	Sinica/CoNLL07 (Chen et al., 2003)	294	87.5	91.8	92.6
Czech	PDT/CoNLL07 (Böhmová et al., 2003)	63	99.1	99.1	99.1
Danish	DDT/CoNLL06 (Kromann et al., 2003)	25	96.2	96.4	96.9
Dutch	Alpino/CoNLL06 (Van der Beek et al., 2002)	12	93.0	95.0	95.0
English	PennTreebank (Marcus et al., 1993)	45	96.7	96.8	97.7
French	FrenchTreebank (Abeillé et al., 2003)	30	96.6	96.7	97.3
German	Tiger/CoNLL06 (Brants et al., 2002)	54	97.9	98.1	98.8
German	Negra (Skut et al., 1997)	54	96.9	97.9	98.6
Greek	GDT/CoNLL07 (Prokopidis et al., 2005)	38	97.2	97.5	97.8
Hungarian	Szeged/CoNLL07 (Cséndes et al., 2005)	43	94.5	95.6	95.8
Italian	ISST/CoNLL07 (Montemagni et al., 2003)	28	94.9	95.8	95.8
Japanese	Verbmobil/CoNLL06 (Kawata and Bartels, 2000)	80	98.3	98.0	99.1
Japanese	Kyoto4.0 (Kurohashi and Nagao, 1997)	42	97.4	98.7	99.3
Korean	Sejong ( <a href="http://www.sejong.or.kr">http://www.sejong.or.kr</a> )	187	96.5	97.5	98.4
Portuguese	Floresta Sintá(c)tica/CoNLL06 (Afonso et al., 2002)	22	96.9	96.8	97.4
Russian	SynTagRus-RNC (Boguslavsky et al., 2002)	11	96.8	96.8	96.8
Slovene	SDT/CoNLL06 (Džeroski et al., 2006)	29	94.7	94.6	95.3
Spanish	Ancora-Cast3LB/CoNLL06 (Civit and Martí, 2004)	47	96.3	96.3	96.9
Swedish	Talbanken05/CoNLL06 (Nivre et al., 2006)	41	93.6	94.7	95.1
Turkish	METU-Sabancı/CoNLL07 (Oflazer et al., 2003)	31	87.5	89.1	90.2

# Next Time

---

# Next Time

---

- ▶ CRFs: feature-based discriminative models

# Next Time

---

- ▶ CRFs: feature-based discriminative models
- ▶ Structured SVM for sequences

# Next Time

---

- ▶ CRFs: feature-based discriminative models
- ▶ Structured SVM for sequences
- ▶ Named entity recognition