UNIVERSITY OF CAMBRIDGE

Efficient segmental features for speech recognition

Rogier van Dalen
with Mark Gales and Anton Ragni

Google, Mountain View, 23 May 2013

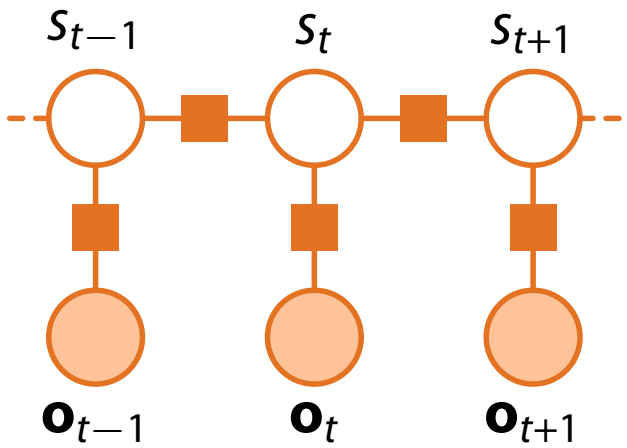
University of Cambridge, Department of Engineering

Outline

Outline

Hidden Markov models

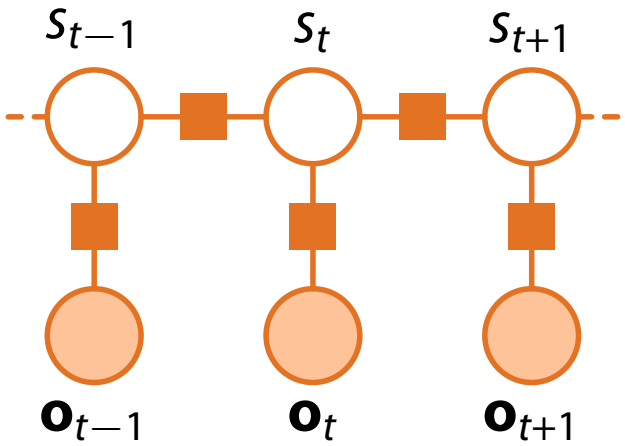
Hidden Markov model:



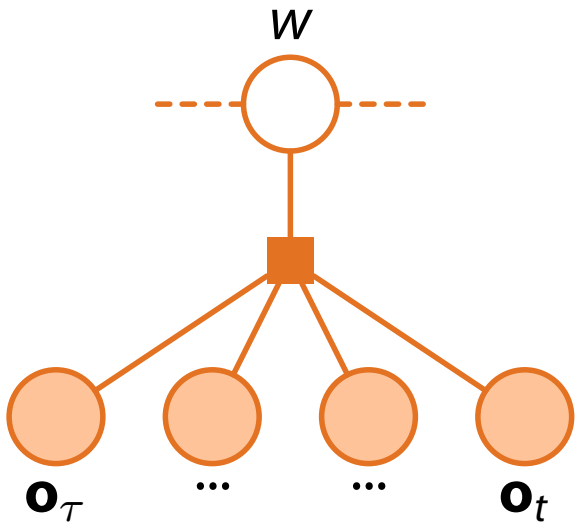
- Markov assumption;
- features (frame-level observations) conditionally independent.
- Training and decoding feasible;
- works in practice, adaptation known.
- Does not model temporal aspects of speech well.

Structured discriminative models

Hidden Markov model



Structured discriminative model

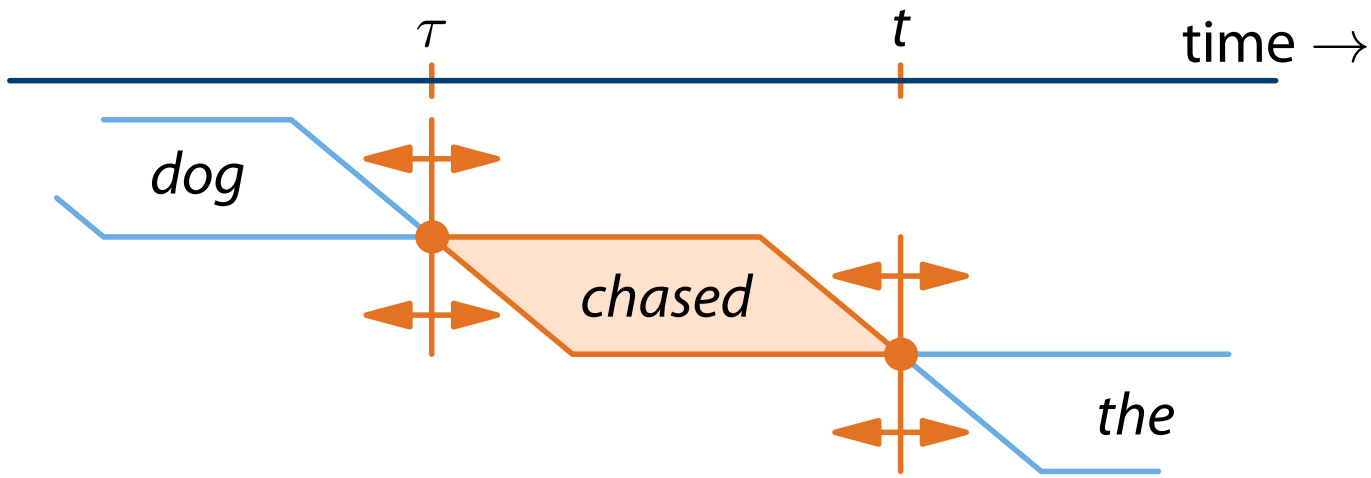


Extract features for a word w (or a phone/triphone).

Segmentations

Extract features for a word w (or a phone/triphone).

$\Rightarrow$ Unknown start and end times.



- $1 \leq \tau \leq t \leq T \Rightarrow \Theta(T^2)$ possible segments.
- Average length of segment is $\frac{1}{2}T$.
- Feature depends on all $\frac{1}{2}T$ observations in the segment.

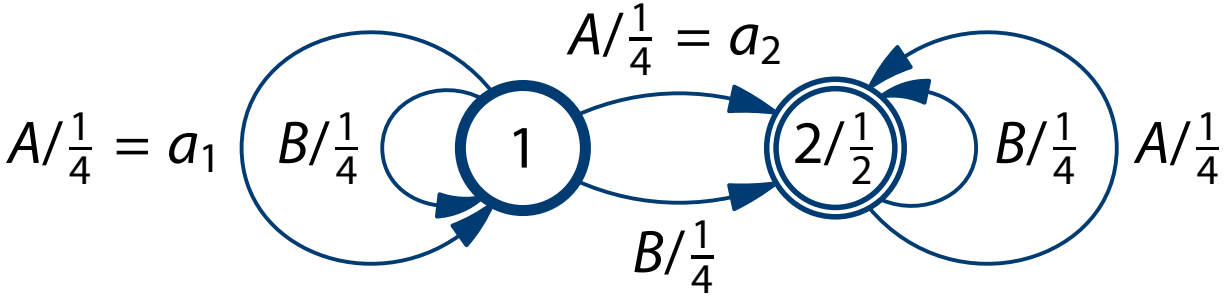
$\Rightarrow$ Extracting features for all segments normally $\Omega(T^3)$.

Outline

Generative score-spaces: toy example

Two-class, two-symbol problem:

- class 1: *AAAA*, *BBBB*;
- class 2: *AABB*, *BBAA*.

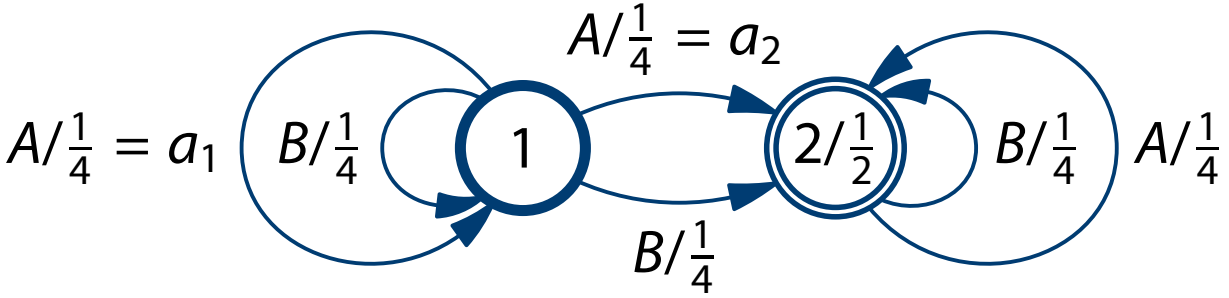


	Class 1		Class 2	
Feature	*AAAA*	*BBBB*	*AABB*	*BBAA*
Log-likelihood	−2.1	−2.1	−2.1	−2.1
∇_{a_1}	6	0	5	1
$\nabla_{a_1}\nabla_{a_2}$	0	0	8	8

Generative score-spaces: toy example

Two-class, two-symbol problem:

- class 1: *AAAA*, *BBBB*;
- class 2: *AABB*, *BBAA*.

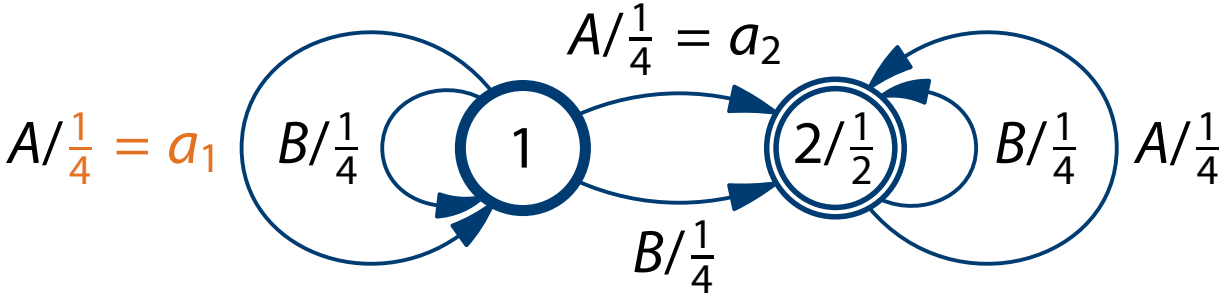


	Class 1		Class 2		
Feature	*AAAA*	*BBBB*	*AABB*	*BBAA*	
Log-likelihood	−2.1	−2.1	−2.1	−2.1	not separable
∇_{a_1}	6	0	5	1	
$\nabla_{a_1}\nabla_{a_2}$	0	0	8	8	

Generative score-spaces: toy example

Two-class, two-symbol problem:

- class 1: *AAAA*, *BBBB*;
- class 2: *AABB*, *BBAA*.

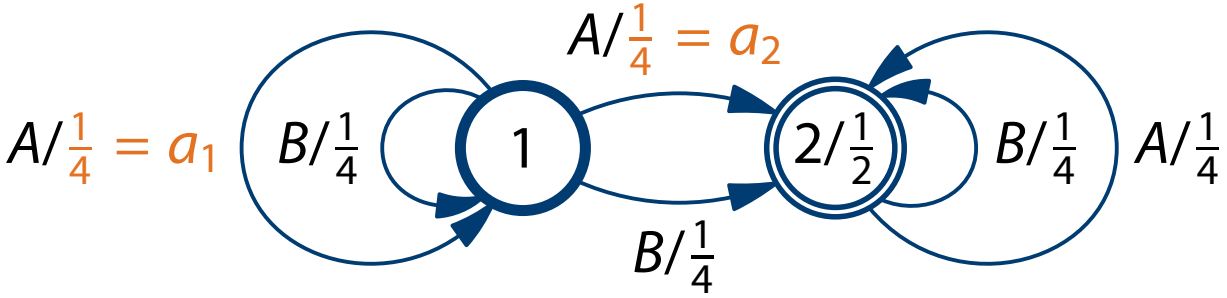


	Class 1		Class 2		
Feature	*AAAA*	*BBBB*	*AABB*	*BBAA*	
Log-likelihood	−2.1	−2.1	−2.1	−2.1	not separable
∇_{a_1}	6	0	5	1	separable, not linearly
$\nabla_{a_1}\nabla_{a_2}$	0	0	8	8	

Generative score-spaces: toy example

Two-class, two-symbol problem:

- class 1: *AAAA*, *BBBB*;
- class 2: *AABB*, *BBAA*.

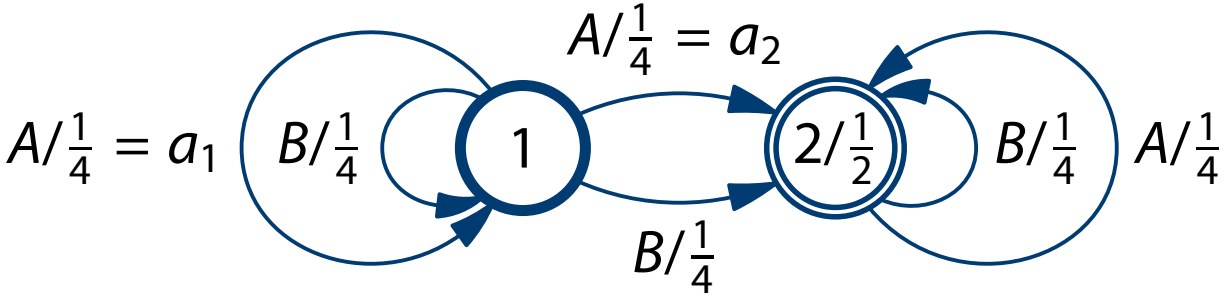


	Class 1		Class 2		
Feature	*AAAA*	*BBBB*	*AABB*	*BBAA*	
Log-likelihood	−2.1	−2.1	−2.1	−2.1	not separable
∇_{a_1}	6	0	5	1	separable, not linearly
$\nabla_{a_1}\nabla_{a_2}$	0	0	8	8	linearly separable

Generative score-spaces: toy example

Two-class, two-symbol problem:

- class 1: *AAAA*, *BBBB*;
- class 2: *AABB*, *BBAA*.



	Class 1		Class 2		
Feature	*AAAA*	*BBBB*	*AABB*	*BBAA*	
Log-likelihood	−2.1	−2.1	−2.1	−2.1	not separable
∇_{a_1}	6	0	5	1	separable, not linearly
$\nabla_{a_1}\nabla_{a_2}$	0	0	8	8	linearly separable

Derivatives introduce extra dependencies between observations.

Generative score-spaces

Generative score-spaces:
Likelihood of word HMM $\ell(\mathbf{o}_{\tau:t}; \boldsymbol{\lambda}_w)$ with parameters $\boldsymbol{\lambda}_w$.
For every word w in the vocabulary:

$$\phi_w(\mathbf{o}_{\tau:t}) = \begin{bmatrix} \log \ell(\mathbf{o}_{\tau:t}; \boldsymbol{\lambda}_w) \\ \nabla_{\boldsymbol{\lambda}} \log \ell(\mathbf{o}_{\tau:t}; \boldsymbol{\lambda}_w) \end{bmatrix}.$$

Derivatives with respect to HMM parameters:

- Introduce dependencies between observations.
- Feature extraction for all $\Theta(T^2)$ segments normally $\Theta(T^3)$.
- Here: exploit assumptions of HMM.

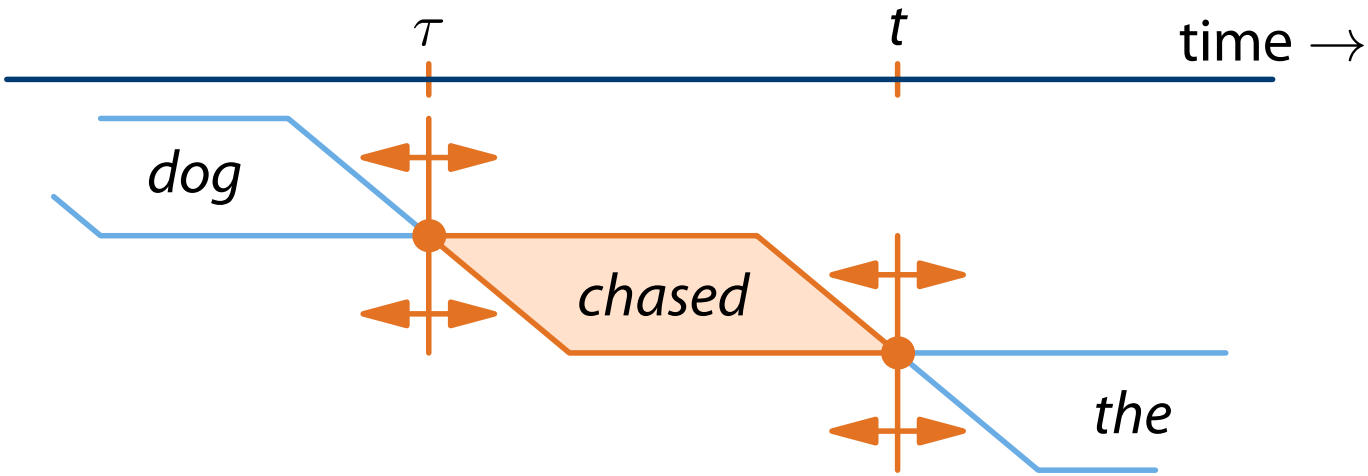
$\Rightarrow$ Feature extraction for $\Theta(T^2)$ segments $\Theta(T^2)$.

Log-linear model

w is a word sequence, **s** is a segmentation of the audio.

$$\arg\max_{\mathbf{w},\mathbf{s}} P(\mathbf{w},\mathbf{s}|\mathbf{O};\alpha) = \arg\max_{\mathbf{w},\mathbf{s}} \frac{1}{Z(\mathbf{O})} \exp\left(\sum_i \alpha^{\mathsf{T}} \phi(\mathbf{O}_{s_i}, w_i)\right).$$

$\phi(\mathbf{O}_{s_i}, w_i)$ contains HMM log-likelihoods and their derivatives.



Assume the language model fixed.

- Decoding is $\Theta(T^2)$ if features for all segments are known.

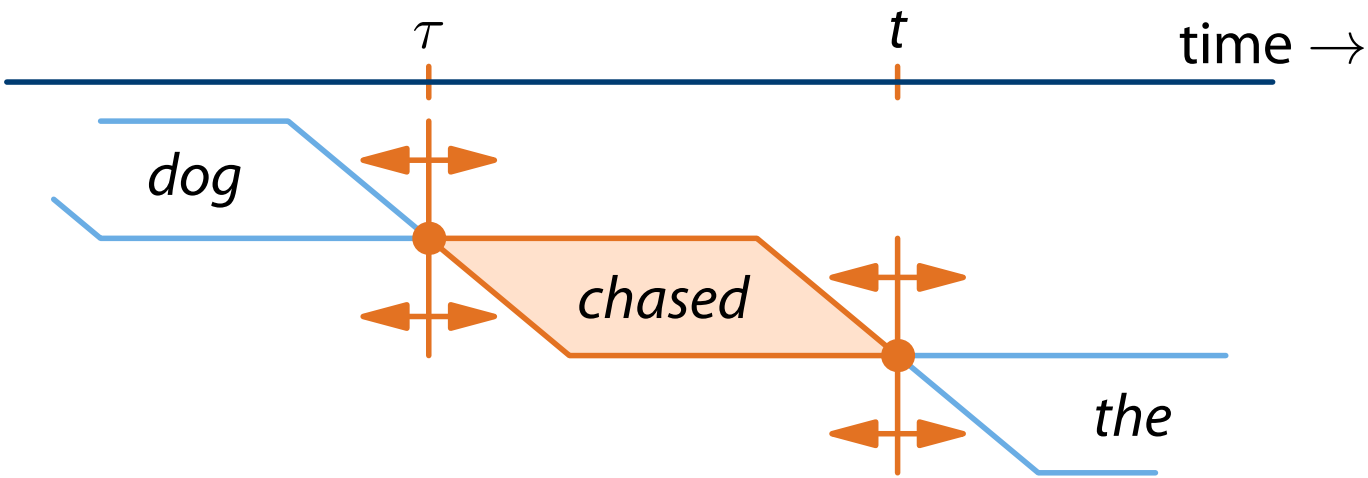
$\Rightarrow$ Feature extraction is the bottleneck.

Log-linear model

w is a word sequence, **s** is a segmentation of the audio.

$$\arg\max_{\mathbf{w},\mathbf{s}} P(\mathbf{w}, \mathbf{s}|\mathbf{O}; \alpha) = \arg\max_{\mathbf{w},\mathbf{s}} \sum_i \alpha^{\mathsf{T}} \phi(\mathbf{O}_{s_i}, w_i).$$

$\phi(\mathbf{O}_{s_i}, w_i)$ contains HMM log-likelihoods and their derivatives.



Assume the language model fixed.

- Decoding is $\Theta(T^2)$ if features for all segments are known.

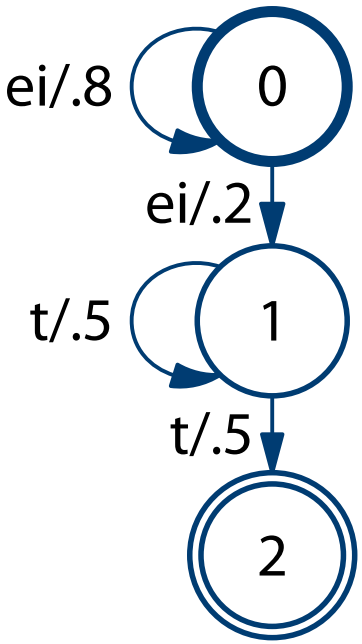
$\Rightarrow$ Feature extraction is the bottleneck.

Outline

Trellis

Likelihood of an HMM: marginalise out over discrete sequence.

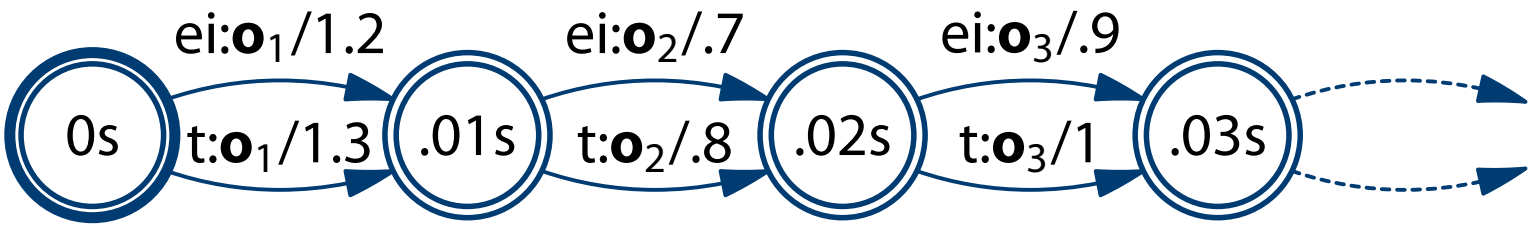
Produce discrete symbol sequence.

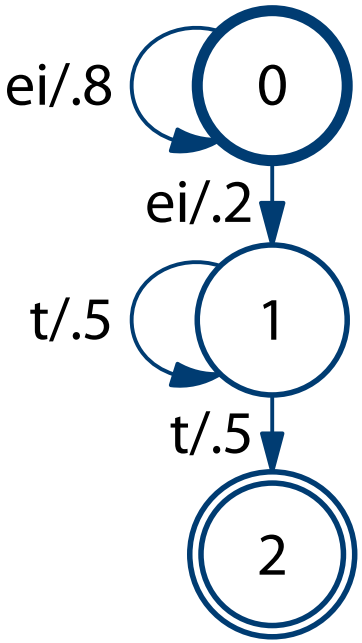


Trellis

Likelihood of an HMM: marginalise out over discrete sequence.

Produce observations (GMM output densities).

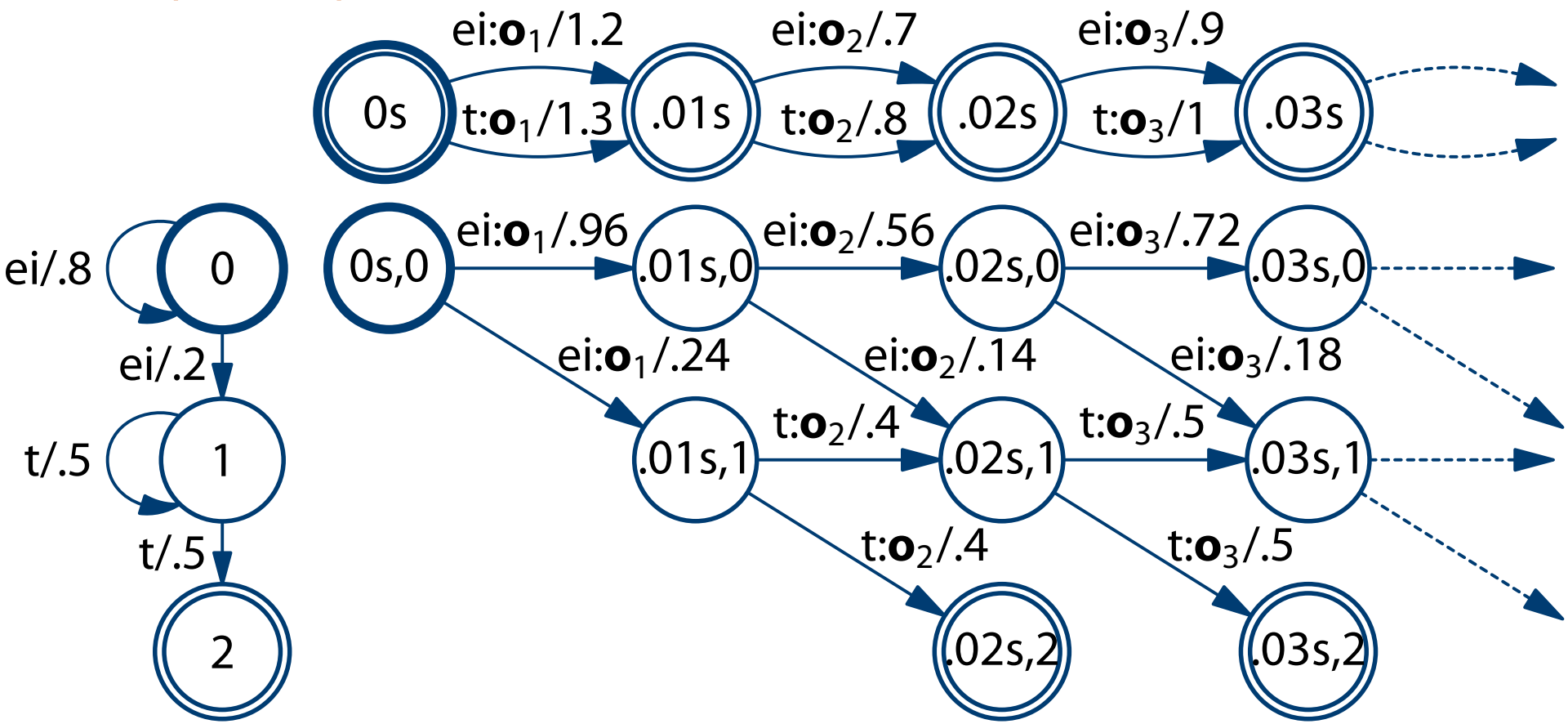




Trellis

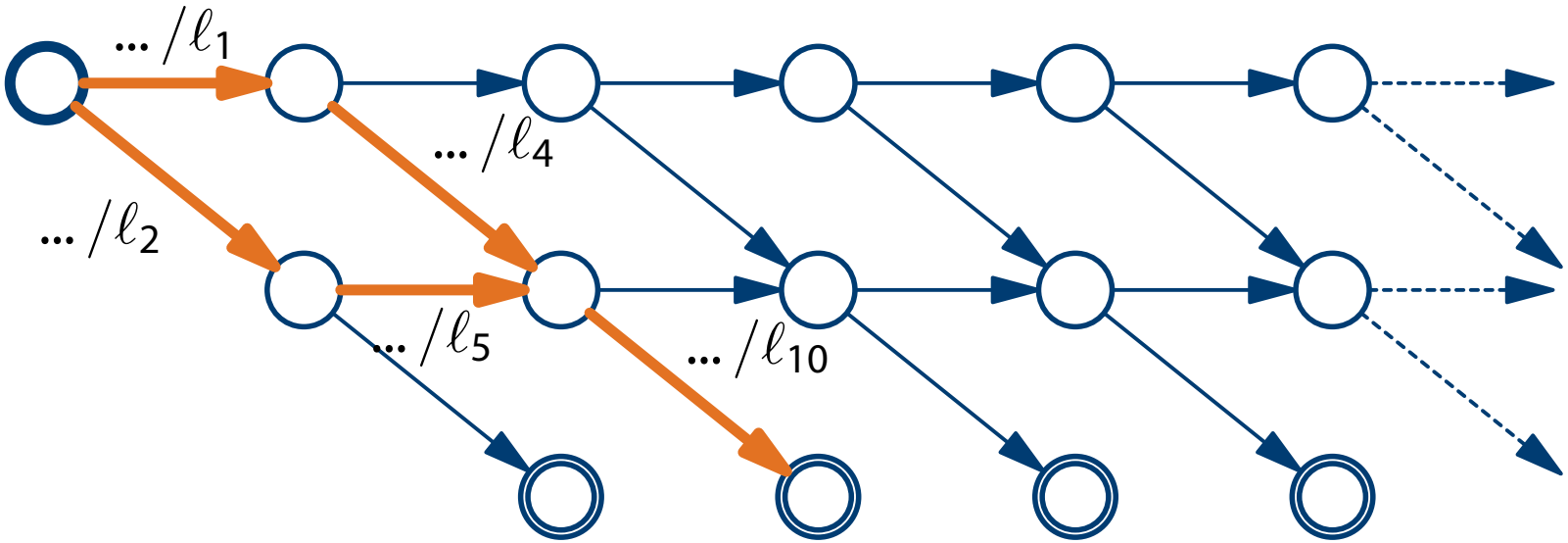
Likelihood of an HMM: marginalise out over discrete sequence.

Compose to produce trellis.



Likelihood

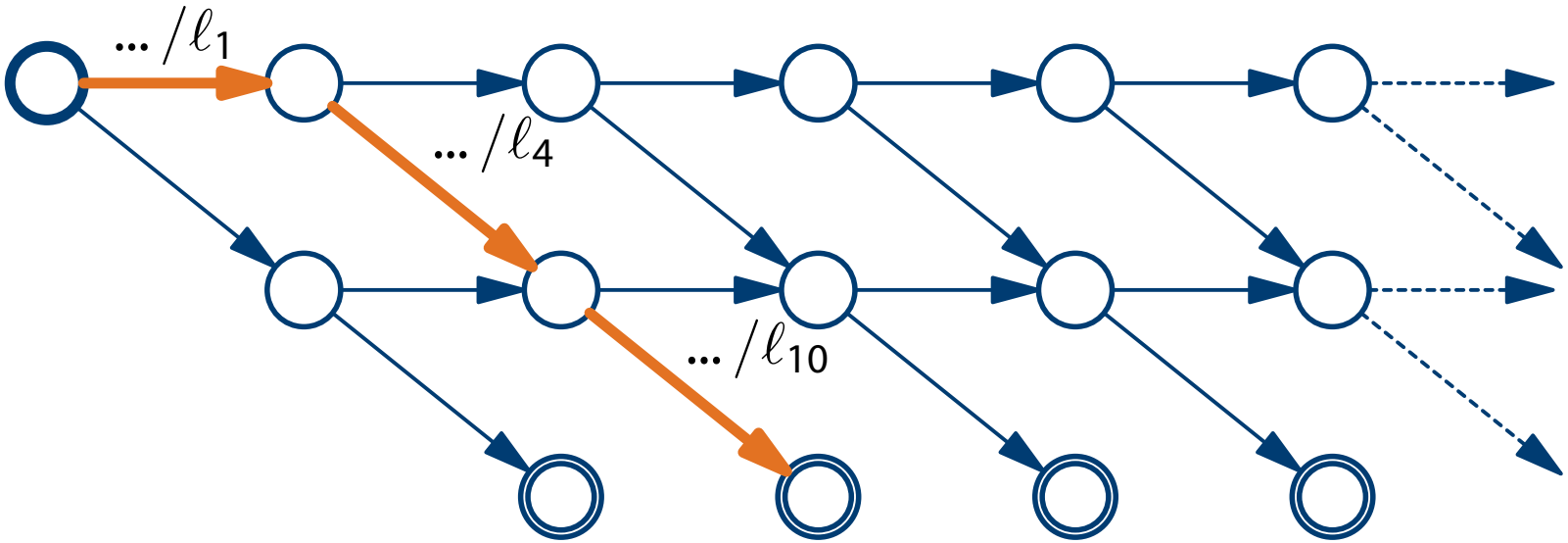
Likelihood of an HMM: sum out over paths in the trellis.



$$\begin{aligned}\ell(\mathbf{o}_{1:3};\boldsymbol{\lambda}) &= \ell_1 \cdot \ell_4 \cdot \ell_{10} + \ell_2 \cdot \ell_5 \cdot \ell_{10} \\ &= (\ell_1 \cdot \ell_4 + \ell_2 \cdot \ell_5) \cdot \ell_{10}.\end{aligned}$$

Likelihood

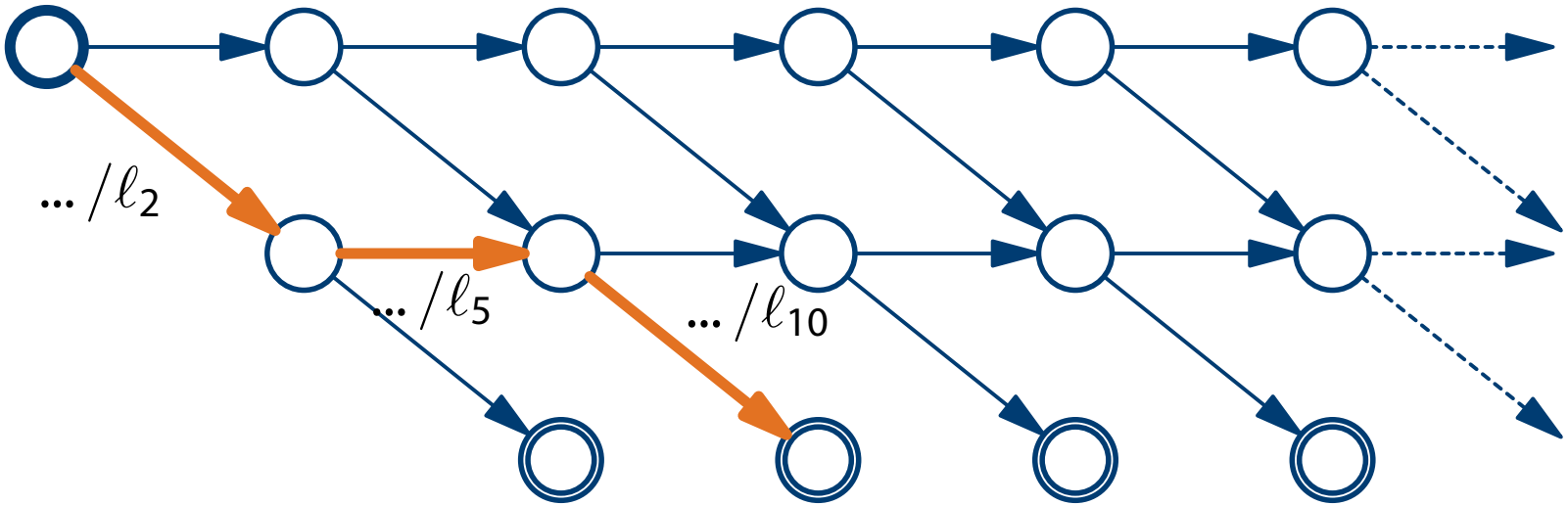
Likelihood of an HMM: sum out over paths in the trellis.



$$\begin{aligned}\ell(\mathbf{o}_{1:3};\boldsymbol{\lambda}) &= \ell_1 \cdot \ell_4 \cdot \ell_{10} + \ell_2 \cdot \ell_5 \cdot \ell_{10} \\ &= (\ell_1 \cdot \ell_4 + \ell_2 \cdot \ell_5) \cdot \ell_{10}.\end{aligned}$$

Likelihood

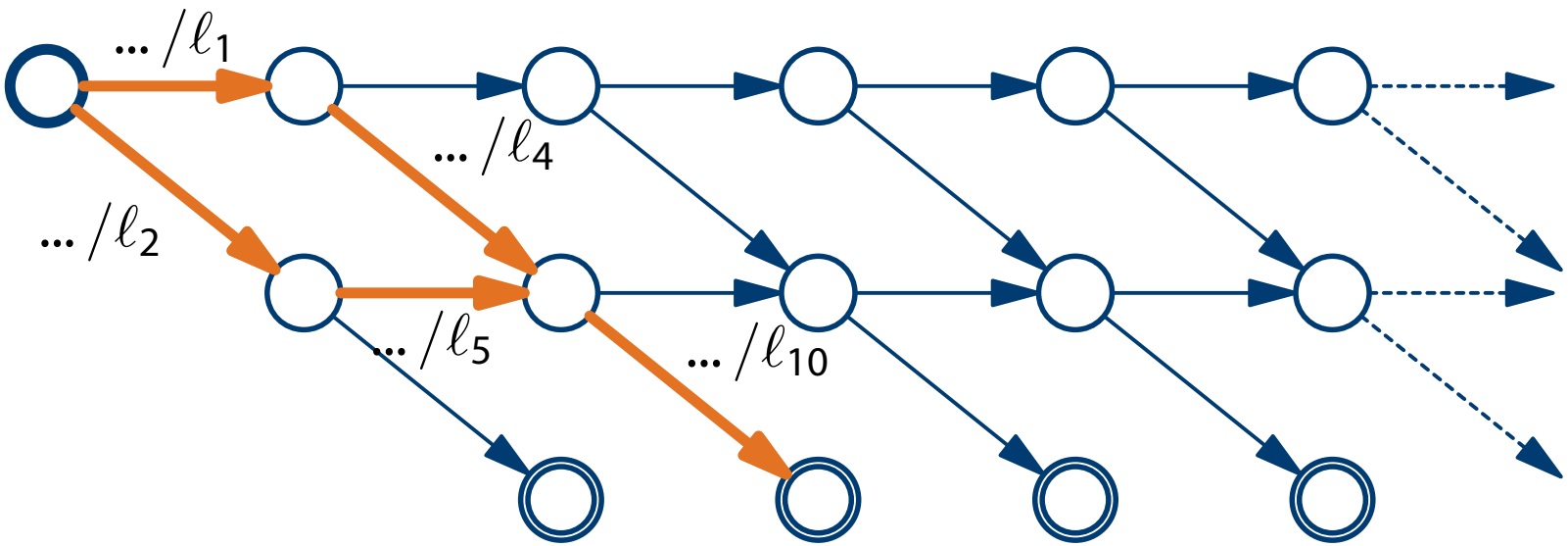
Likelihood of an HMM: sum out over paths in the trellis.



$$\begin{aligned}\ell(\mathbf{o}_{1:3};\boldsymbol{\lambda}) &= \ell_1 \cdot \ell_4 \cdot \ell_{10} + \ell_2 \cdot \ell_5 \cdot \ell_{10} \\ &= (\ell_1 \cdot \ell_4 + \ell_2 \cdot \ell_5) \cdot \ell_{10}.\end{aligned}$$

Likelihood

Likelihood of an HMM: sum out over paths in the trellis.

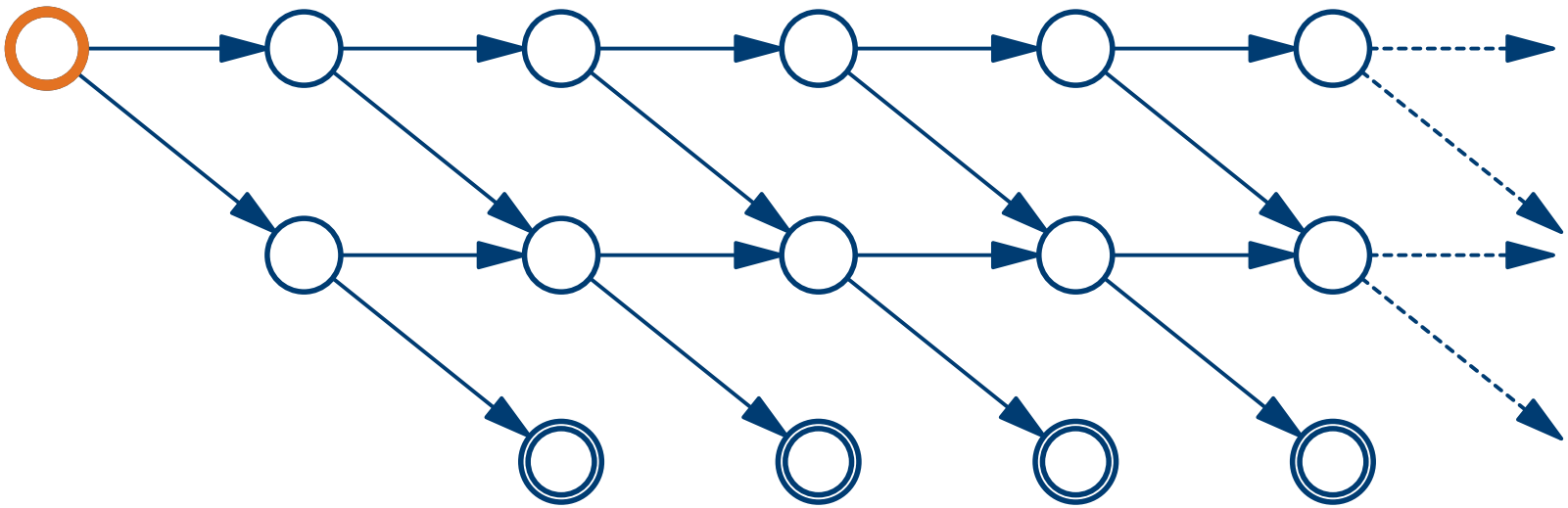


$$\begin{aligned}\ell(\mathbf{o}_{1:3};\boldsymbol{\lambda}) &= \ell_1 \cdot \ell_4 \cdot \ell_{10} + \ell_2 \cdot \ell_5 \cdot \ell_{10} \\ &= (\ell_1 \cdot \ell_4 + \ell_2 \cdot \ell_5) \cdot \ell_{10}.\end{aligned}$$

$\Rightarrow$ This allows dynamic programming.

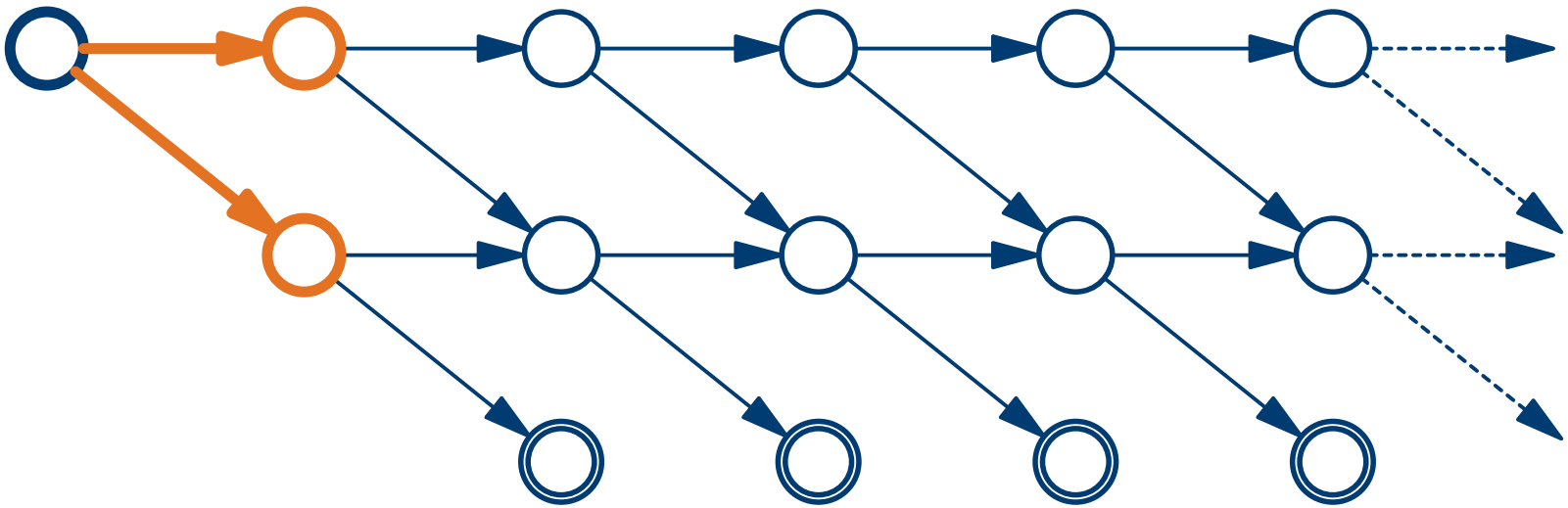
Forward algorithm

Dynamic programming: "frontier" of weights moves forward per time step.



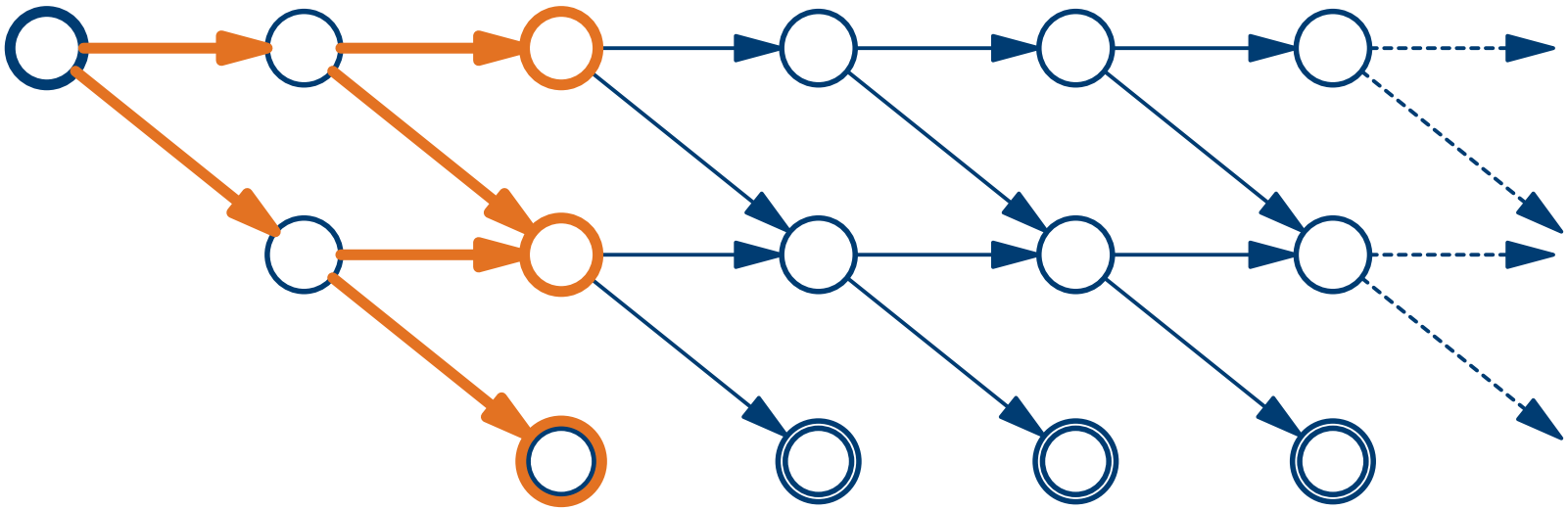
Forward algorithm

Dynamic programming: "frontier" of weights moves forward per time step.



Forward algorithm

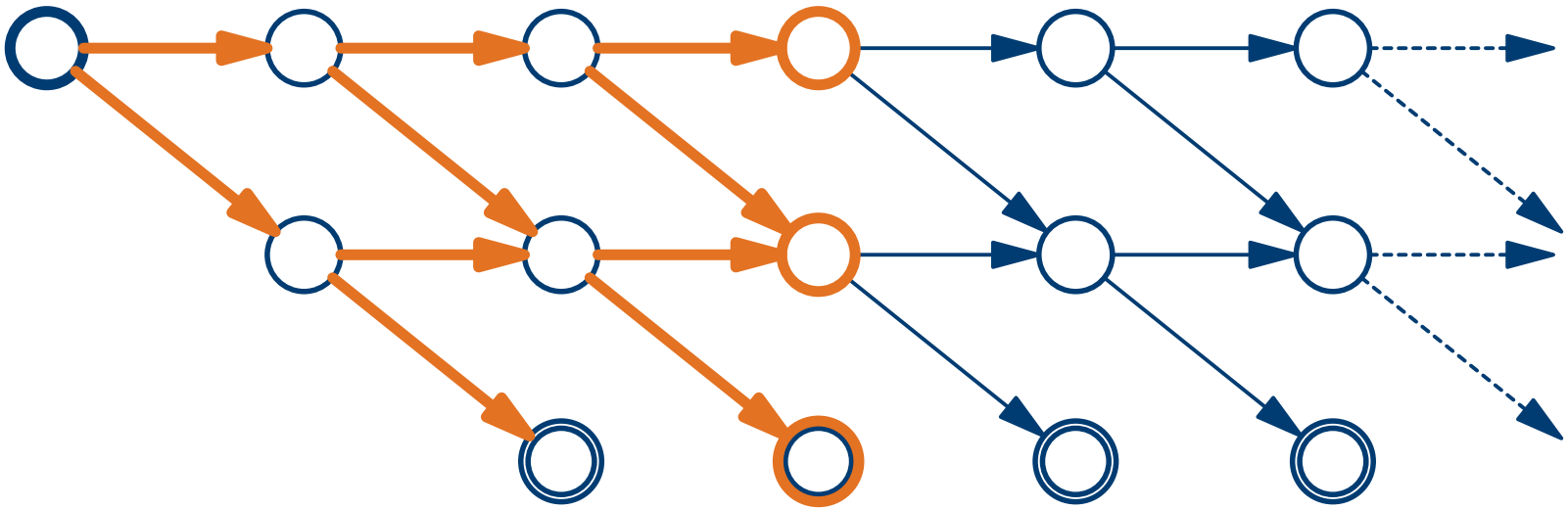
Dynamic programming: "frontier" of weights moves forward per time step.



$\ell(\mathbf{o}_{1:2}; \boldsymbol{\lambda})$

Forward algorithm

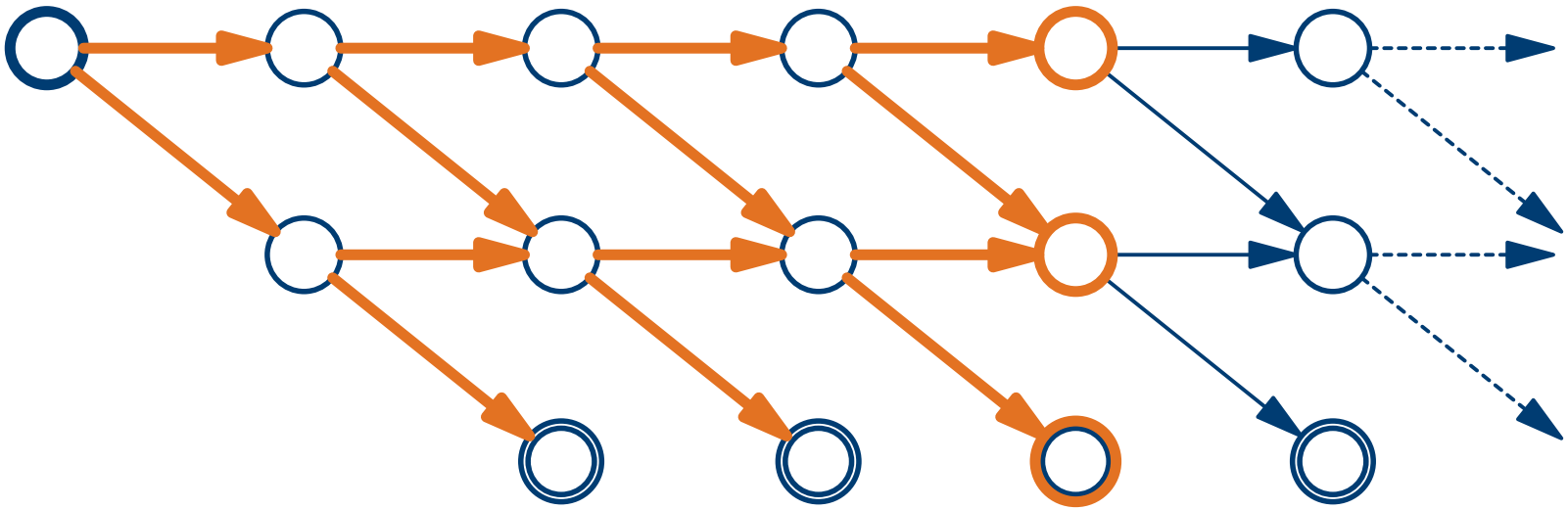
Dynamic programming: "frontier" of weights moves forward per time step.



$$\ell(\mathbf{o}_{1:2}; \boldsymbol{\lambda}) \, ; \ell(\mathbf{o}_{1:3}; \boldsymbol{\lambda})$$

Forward algorithm

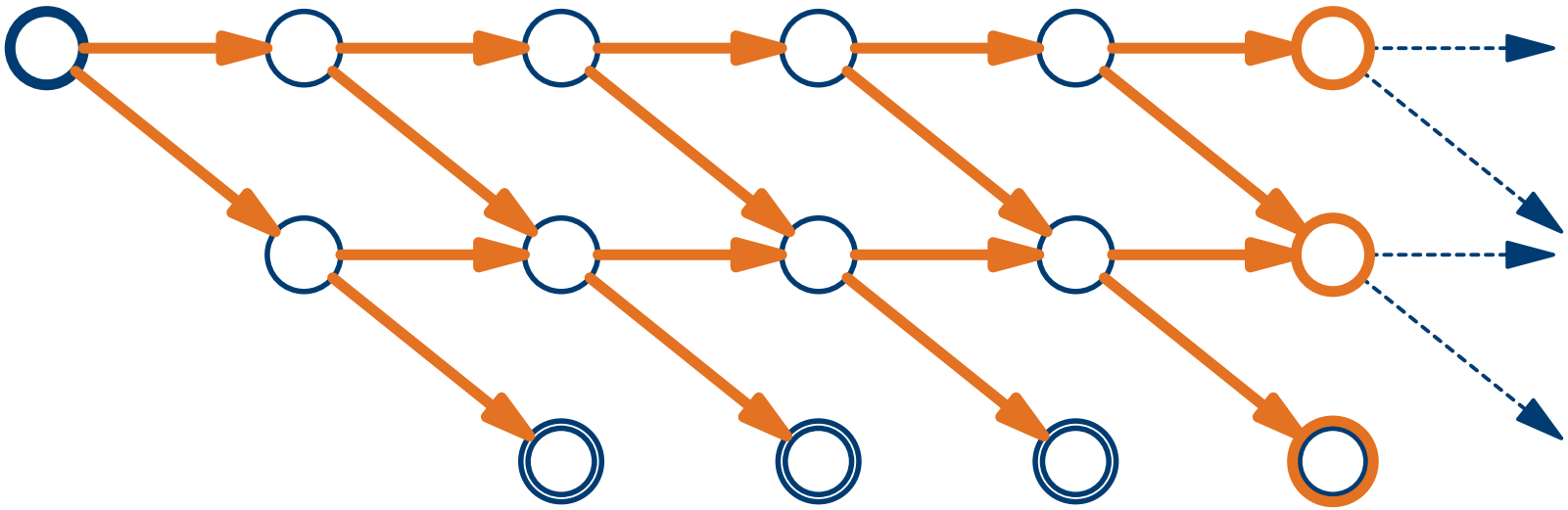
Dynamic programming: "frontier" of weights moves forward per time step.



$$\ell(\mathbf{o}_{1:2}; \boldsymbol{\lambda}) \, ; \ell(\mathbf{o}_{1:3}; \boldsymbol{\lambda}) \, ; \ell(\mathbf{o}_{1:4}; \boldsymbol{\lambda})$$

Forward algorithm

Dynamic programming: "frontier" of weights moves forward per time step.



$$\ell(\mathbf{o}_{1:2}; \boldsymbol{\lambda}) \, ; \ell(\mathbf{o}_{1:3}; \boldsymbol{\lambda}) \, ; \ell(\mathbf{o}_{1:4}; \boldsymbol{\lambda}) \, ; \ell(\mathbf{o}_{1:5}; \boldsymbol{\lambda}) \, .$$

- Likelihoods, for all T segments starting at $\tau = 1$, in $\Theta(T)$ time.
- Repeat this for all $\tau = 1 \ldots T$:
  $\Rightarrow$ $\Theta(T^2)$ likelihoods in $\Theta(T^2)$ time.

Generative score-spaces

Add derivatives:

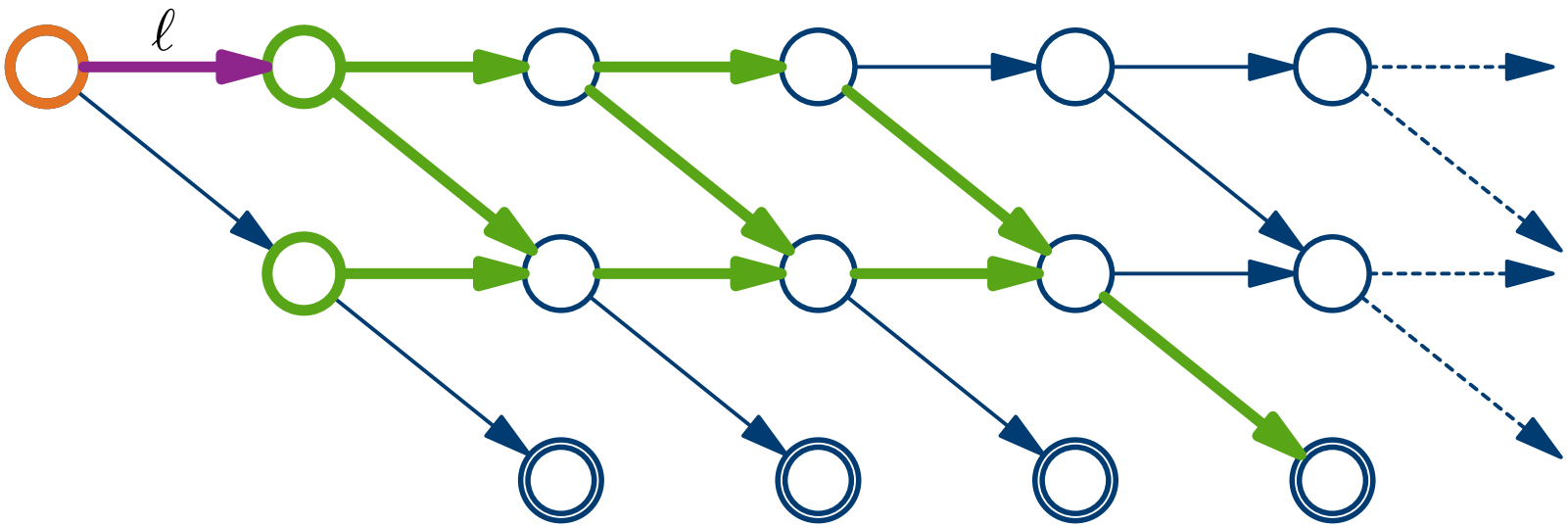
$$\phi_w(\mathbf{o}_{\tau:t}) = \begin{bmatrix} \log \ell(\mathbf{o}_{\tau:t}; \boldsymbol{\lambda}_w) \\ \nabla_{\boldsymbol{\lambda}} \log \ell(\mathbf{o}_{\tau:t}; \boldsymbol{\lambda}_w) \end{bmatrix}, \qquad \nabla_{\boldsymbol{\lambda}} \log \ell = \frac{\nabla_{\boldsymbol{\lambda}} \ell}{\ell}.$$

$$\nabla_{\boldsymbol{\lambda}} \ell(\mathbf{o}_{\tau:t}; \boldsymbol{\lambda}_w) = \sum_e \gamma_{\tau:t}(e) \frac{\nabla_{\boldsymbol{\lambda}} \ell[e; \boldsymbol{\lambda}_w]}{\ell[e; \boldsymbol{\lambda}_w]}.$$

$\gamma_{\tau:t}(e)$ is the arc posterior.
This uses the forward-backward algorithm.

Forward-backward algorithm

Forward-backward algorithm to compute the arc posterior $\gamma_{\tau:t}(e)$:

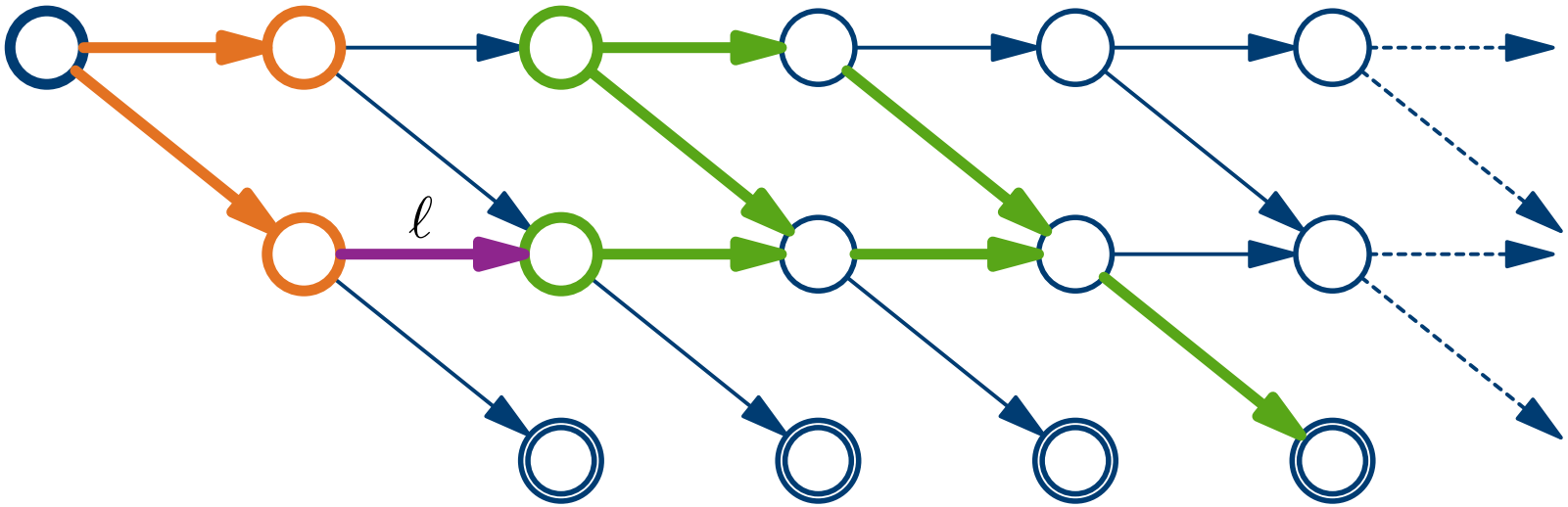


$$\gamma_{\tau:t}(e) \propto \text{forwd}(\ldots) \cdot \ell \cdot \text{backwd}(\ldots).$$

- Derivatives introduce dependencies between observations.
- This multiplication must be redone for every segment:
  - $\Theta\left(T^3\right)$ for all segments.

Forward-backward algorithm

Forward-backward algorithm to compute the arc posterior $\gamma_{\tau:t}(e)$:

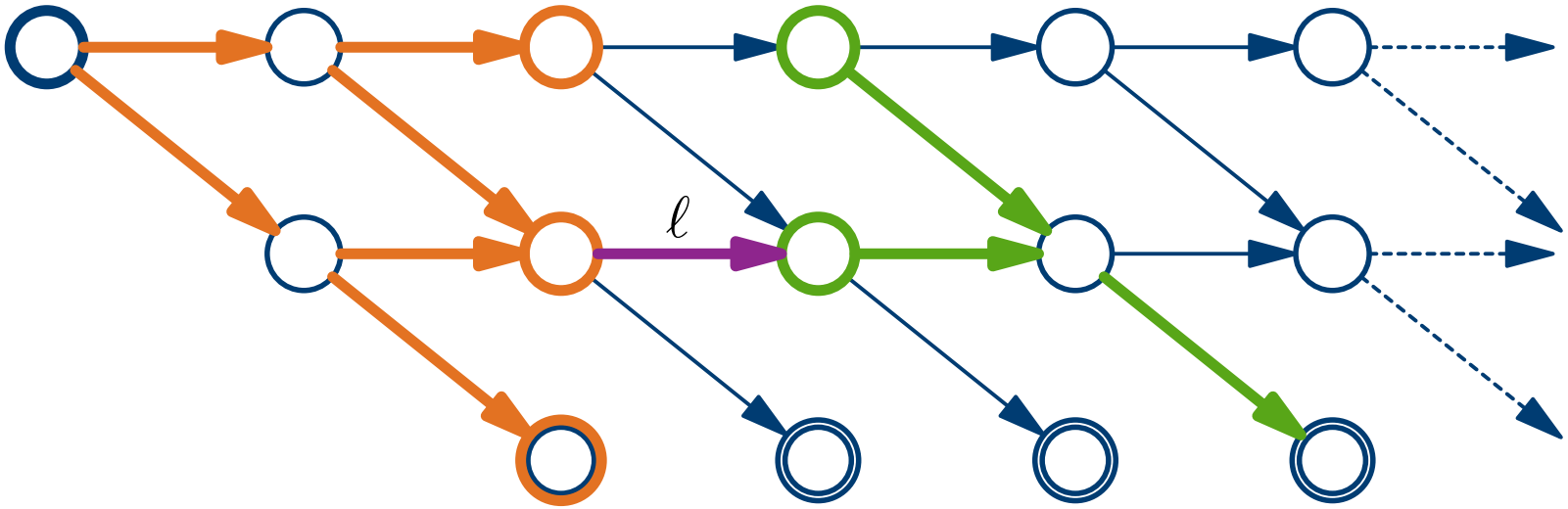


$$\gamma_{\tau:t}(e) \propto \text{forwd}(\ldots) \cdot \ell \cdot \text{backwd}(\ldots).$$

- Derivatives introduce dependencies between observations.
- This multiplication must be redone for every segment:
  - $\Theta\left(T^3\right)$ for all segments.

Forward-backward algorithm

Forward-backward algorithm to compute the arc posterior $\gamma_{\tau:t}(e)$:

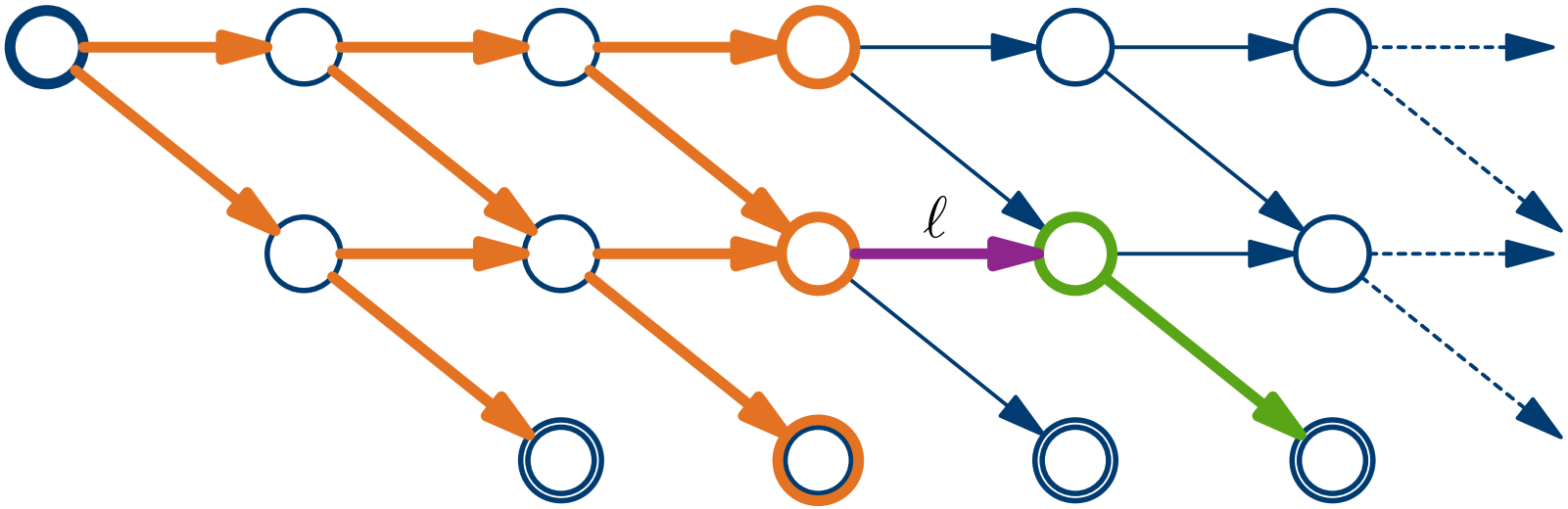


$$\gamma_{\tau:t}(e) \propto \text{forwd}(\ldots) \cdot \ell \cdot \text{backwd}(\ldots).$$

- Derivatives introduce dependencies between observations.
- This multiplication must be redone for every segment:
  - $\Theta\left(T^3\right)$ for all segments.

Forward-backward algorithm

Forward-backward algorithm to compute the arc posterior $\gamma_{\tau:t}(e)$:

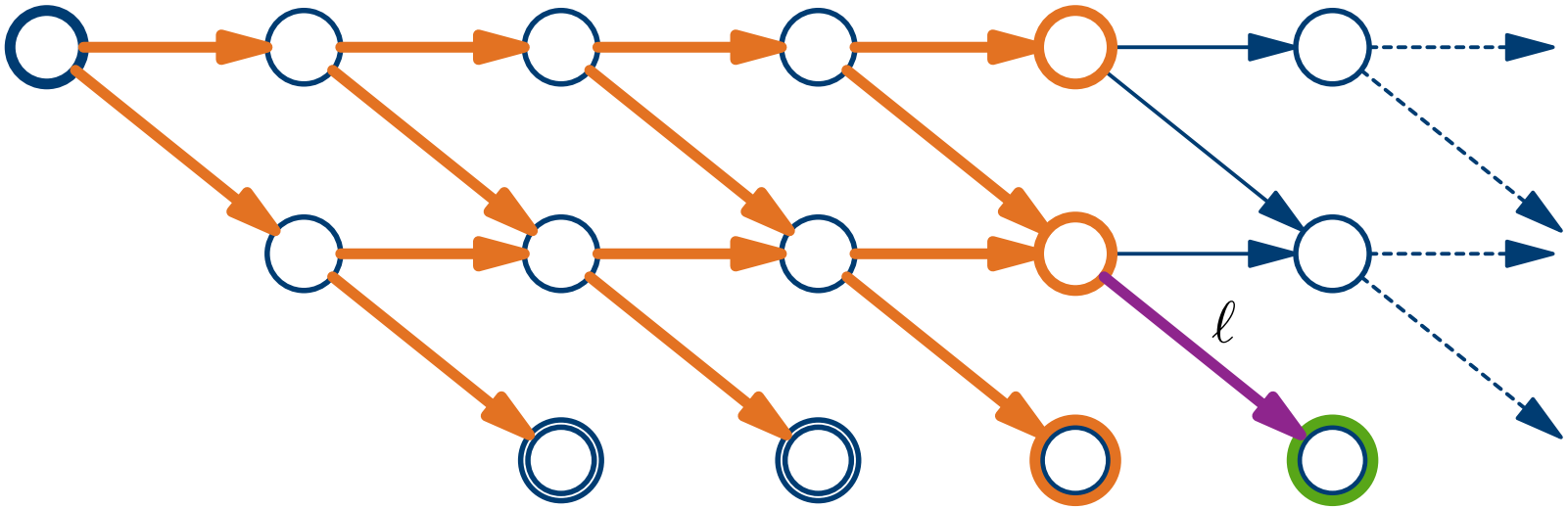


$$\gamma_{\tau:t}(e) \propto \text{forwd}(\ldots) \cdot \ell \cdot \text{backwd}(\ldots).$$

- Derivatives introduce dependencies between observations.
- This multiplication must be redone for every segment:
  - $\Theta\left(T^3\right)$ for all segments.

Forward-backward algorithm

Forward-backward algorithm to compute the arc posterior $\gamma_{\tau:t}(e)$:

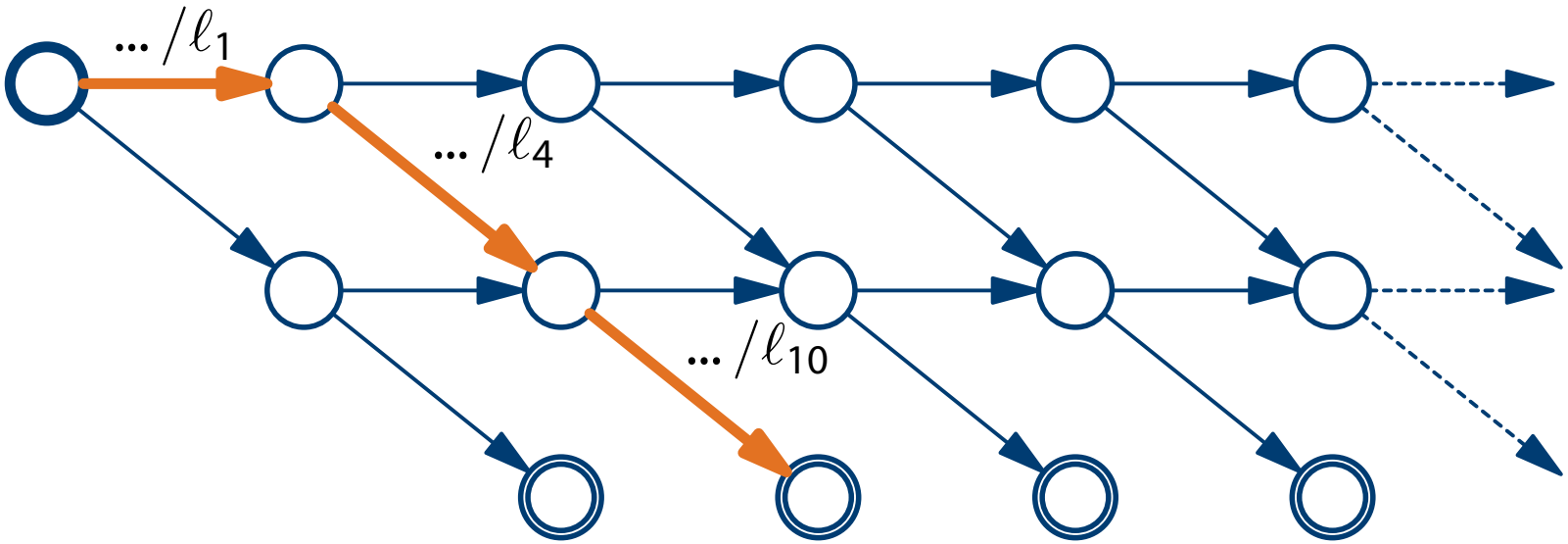


$$\gamma_{\tau:t}(e) \propto \text{forwd}(\ldots) \cdot \ell \cdot \text{backwd}(\ldots).$$

- Derivatives introduce dependencies between observations.
- This multiplication must be redone for every segment:
  - $\Theta\left(T^3\right)$ for all segments.

Derivative

Derivative of the likelihood of an HMM:

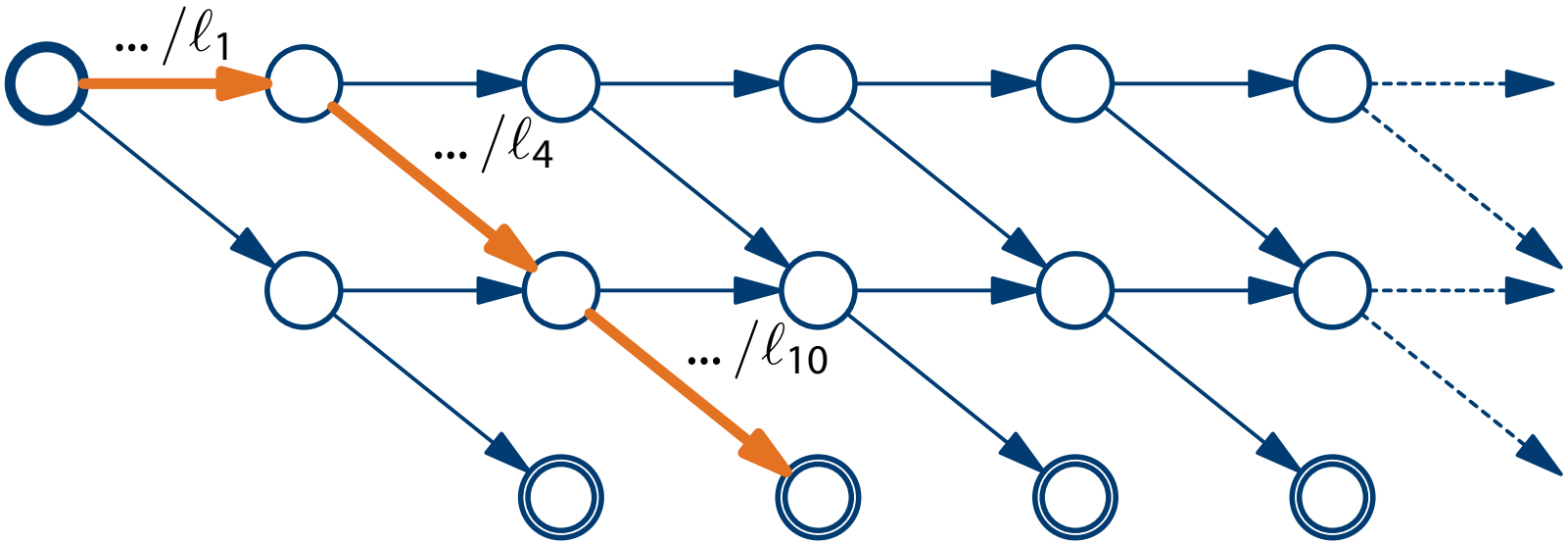


$$\ell(\mathbf{o}_{1:3}; \boldsymbol{\lambda}) = \ell_1 \cdot \ell_4 \cdot \ell_{10} + \ell_2 \cdot \ell_5 \cdot \ell_{10};$$

$$\nabla_{\boldsymbol{\lambda}} \ell(\mathbf{o}_{1:3}; \boldsymbol{\lambda}) = (\nabla_{\boldsymbol{\lambda}} \ell_1) \cdot \ell_4 \cdot \ell_{10} + \ell_1 \cdot \nabla_{\boldsymbol{\lambda}} (\ell_4 \cdot \ell_{10}) + \dots.$$

Derivative

Derivative of the likelihood of an HMM:



$$\ell(\mathbf{o}_{1:3}; \boldsymbol{\lambda}) = \ell_1 \cdot \ell_4 \cdot \ell_{10} + \ell_2 \cdot \ell_5 \cdot \ell_{10};$$
$$\nabla_{\boldsymbol{\lambda}} \ell(\mathbf{o}_{1:3}; \boldsymbol{\lambda}) = (\nabla_{\boldsymbol{\lambda}} \ell_1) \cdot \ell_4 \cdot \ell_{10} + \ell_1 \cdot \nabla_{\boldsymbol{\lambda}} (\ell_4 \cdot \ell_{10}) + \dots.$$

$\Rightarrow$ Generalise this: expectation semiring.

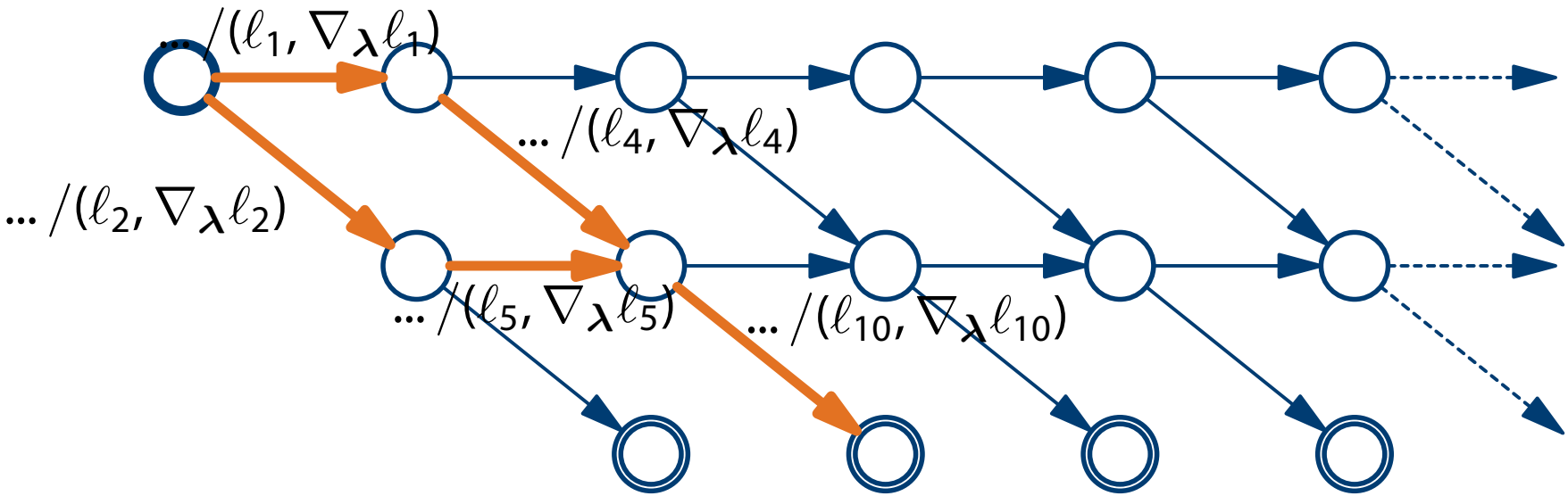
Semirings

What is a semiring?

- Set with operations $\oplus$ and $\otimes$, and values $\overline{0}$ and $\overline{1}$.
- E.g. $\mathbb{R}$ with $+$ and $\times$, and 0 and 1.
- E.g. $\mathbb{R}^+ \cup \{0\}$ with max and $\times$, and 0 and 1
  - turns the forward algorithm into Viterbi.
- E.g. the expectation semiring: a tuple (ℓ, c).

Expectation semiring

Replace weights by tuples $\sigma = (\ell, \nabla_{\boldsymbol{\lambda}} \ell)$ in the expectation semiring:



Define operations $\otimes$ and $\oplus$:

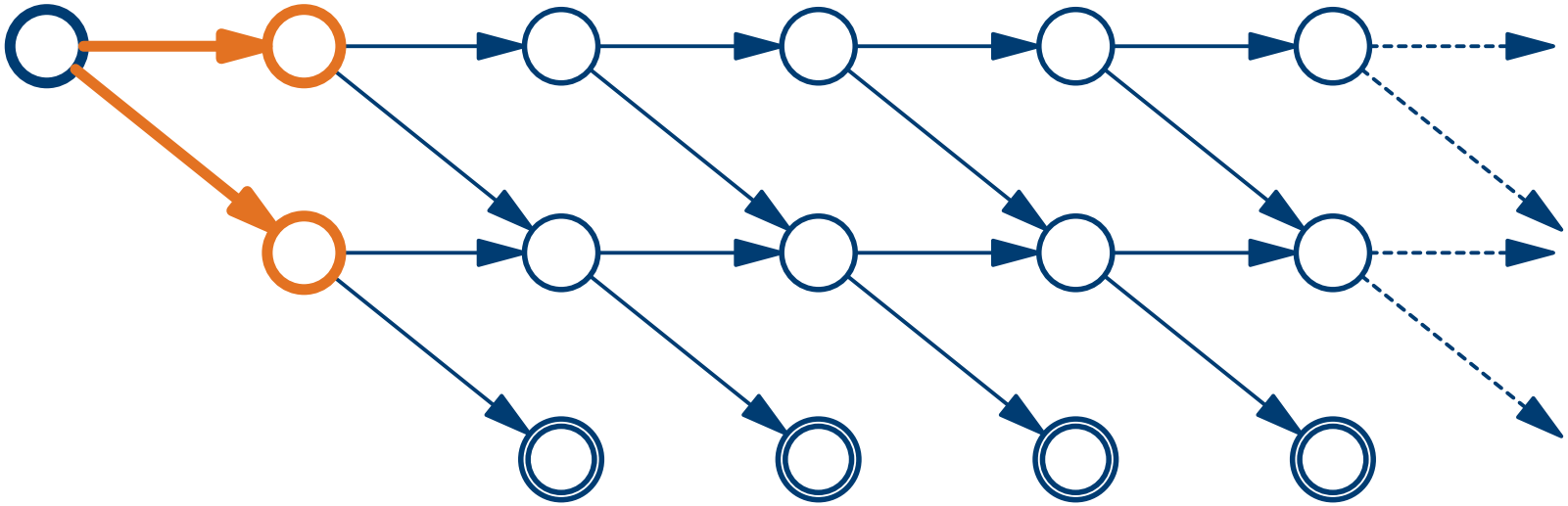
$$(\ell_1, \nabla_{\boldsymbol{\lambda}} \ell_1) \otimes (\ell_4, \nabla_{\boldsymbol{\lambda}} \ell_4) \triangleq (\ell_1 \cdot \ell_4, \ell_1 \cdot \nabla_{\boldsymbol{\lambda}} \ell_4 + \ell_4 \cdot \nabla_{\boldsymbol{\lambda}} \ell_1)$$

$$(\ell_1, \nabla_{\boldsymbol{\lambda}} \ell_1) \oplus (\ell_2, \nabla_{\boldsymbol{\lambda}} \ell_2) \triangleq (\ell_1 + \ell_2, \nabla_{\boldsymbol{\lambda}} \ell_1 + \nabla_{\boldsymbol{\lambda}} \ell_2)$$

$$(\sigma_1 \otimes \sigma_4 \otimes \sigma_{10}) \oplus (\sigma_2 \otimes \sigma_5 \otimes \sigma_{10}) = (\ell(\mathbf{o}_{1:3}; \boldsymbol{\lambda}), \nabla_{\boldsymbol{\lambda}} \ell(\mathbf{o}_{1:3}; \boldsymbol{\lambda})) .$$

Expectation semiring

The forward algorithm with the expectation semiring:

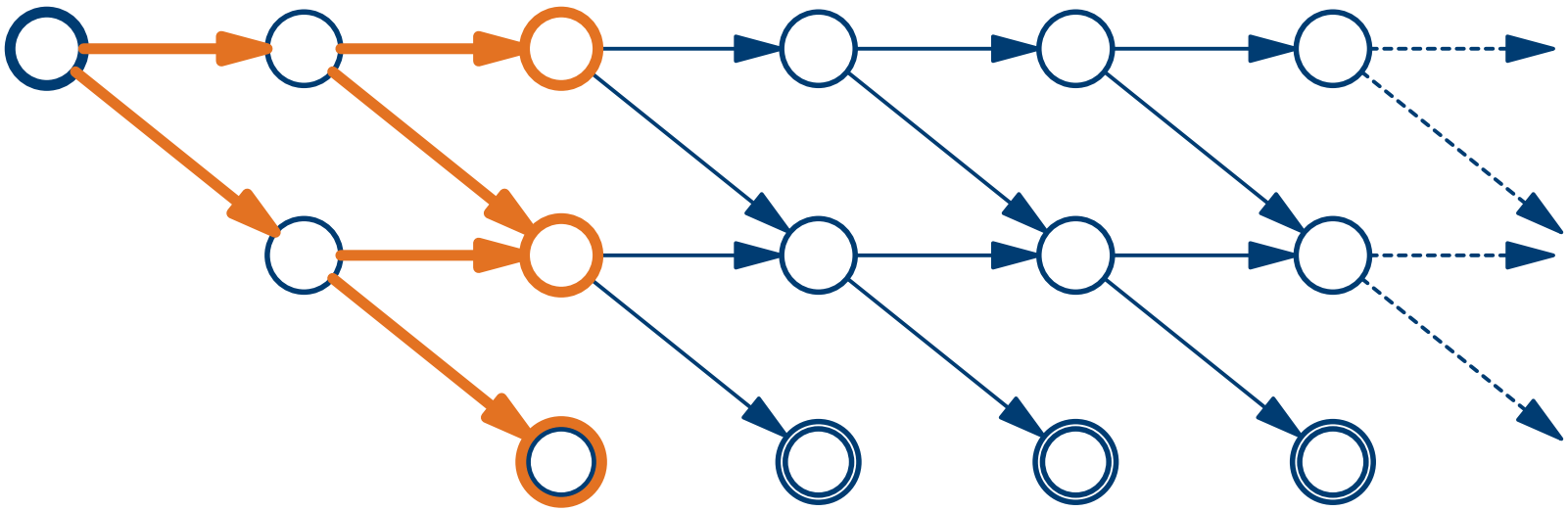


Likelihoods and their derivatives,

- for all T segments starting at $\tau = 1$, in $\Theta(T)$ time.
- for all $\Theta\left(T^2\right)$ segments, in $\Theta\left(T^2\right)$ time.

Expectation semiring

The forward algorithm with the expectation semiring:



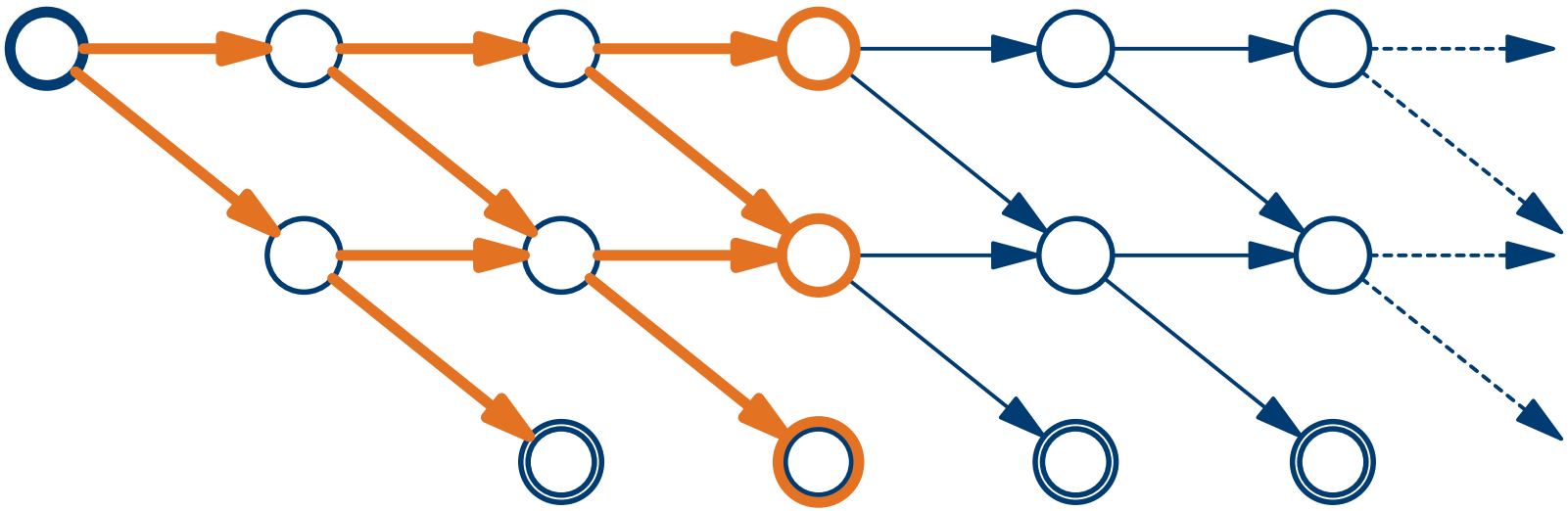
$$\left(\ell(\mathbf{o}_{1:2};\boldsymbol{\lambda})\,,\nabla_{\boldsymbol{\lambda}}\ell(\mathbf{o}_{1:2};\boldsymbol{\lambda})\right)$$

Likelihoods and their derivatives,

- for all T segments starting at $\tau = 1$, in $\Theta(T)$ time.
- for all $\Theta\left(T^2\right)$ segments, in $\Theta\left(T^2\right)$ time.

Expectation semiring

The forward algorithm with the expectation semiring:



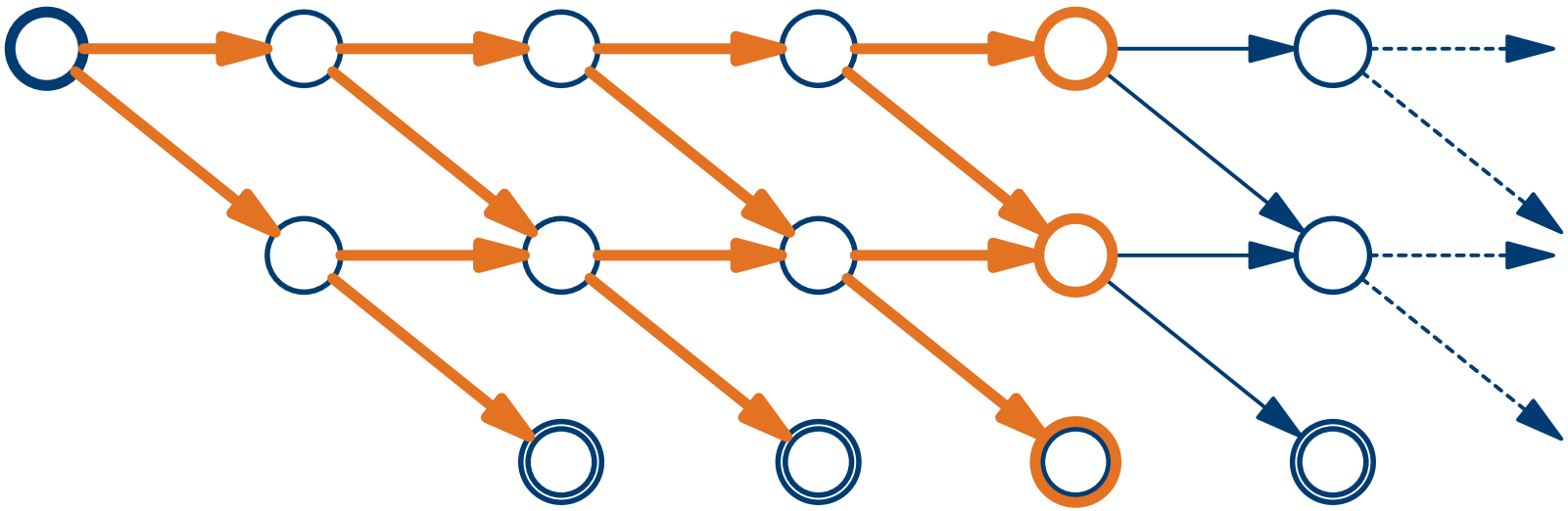
$$(\ell(\mathbf{o}_{1:2};\boldsymbol{\lambda}), \nabla_{\boldsymbol{\lambda}}\ell(\mathbf{o}_{1:2};\boldsymbol{\lambda}));\qquad (\ell(\mathbf{o}_{1:3};\boldsymbol{\lambda}), \nabla_{\boldsymbol{\lambda}}\ell(\mathbf{o}_{1:3};\boldsymbol{\lambda}))$$

Likelihoods and their derivatives,

- for all T segments starting at $\tau = 1$, in $\Theta(T)$ time.
- for all $\Theta(T^2)$ segments, in $\Theta(T^2)$ time.

Expectation semiring

The forward algorithm with the expectation semiring:



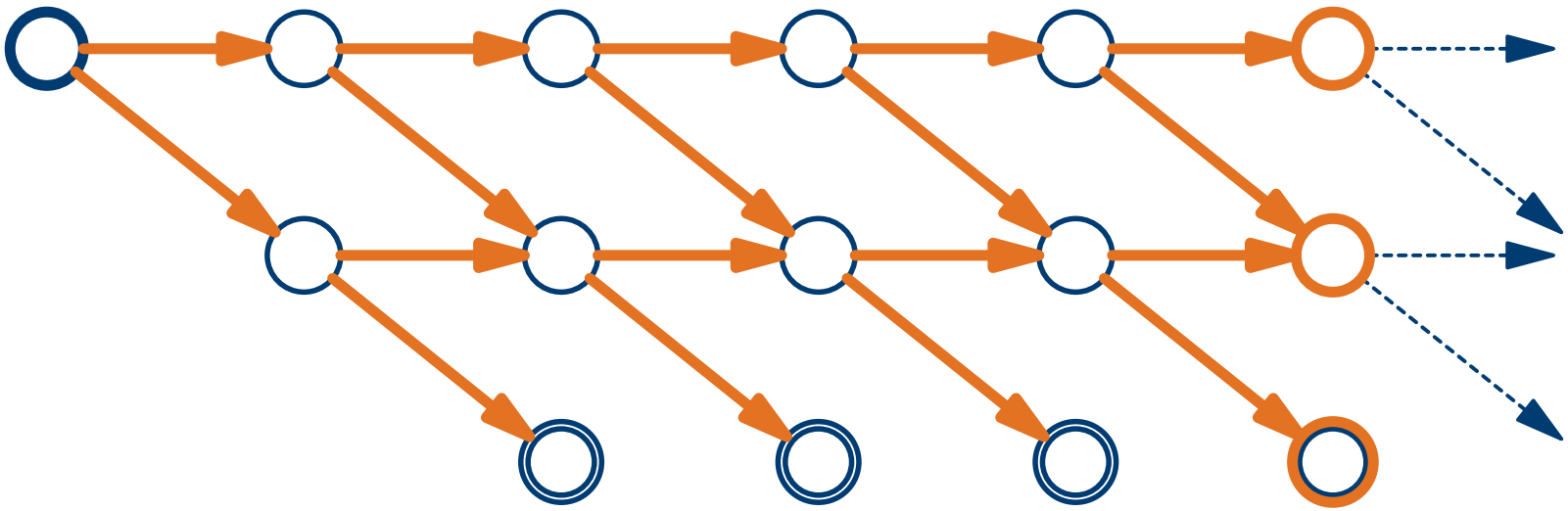
$(\ell(\mathbf{o}_{1:2};\boldsymbol{\lambda}), \nabla_{\boldsymbol{\lambda}}\ell(\mathbf{o}_{1:2};\boldsymbol{\lambda}))$; $(\ell(\mathbf{o}_{1:3};\boldsymbol{\lambda}), \nabla_{\boldsymbol{\lambda}}\ell(\mathbf{o}_{1:3};\boldsymbol{\lambda}))$;
$(\ell(\mathbf{o}_{1:4};\boldsymbol{\lambda}), \nabla_{\boldsymbol{\lambda}}\ell(\mathbf{o}_{1:4};\boldsymbol{\lambda}))$

Likelihoods and their derivatives,

- for all T segments starting at $\tau = 1$, in $\Theta(T)$ time.
- for all $\Theta(T^2)$ segments, in $\Theta(T^2)$ time.

Expectation semiring

The forward algorithm with the expectation semiring:



$$(\ell(\mathbf{o}_{1:2};\boldsymbol{\lambda}),\nabla_{\boldsymbol{\lambda}}\ell(\mathbf{o}_{1:2};\boldsymbol{\lambda}));\qquad (\ell(\mathbf{o}_{1:3};\boldsymbol{\lambda}),\nabla_{\boldsymbol{\lambda}}\ell(\mathbf{o}_{1:3};\boldsymbol{\lambda}));$$
$$(\ell(\mathbf{o}_{1:4};\boldsymbol{\lambda}),\nabla_{\boldsymbol{\lambda}}\ell(\mathbf{o}_{1:4};\boldsymbol{\lambda}));\qquad (\ell(\mathbf{o}_{1:5};\boldsymbol{\lambda}),\nabla_{\boldsymbol{\lambda}}\ell(\mathbf{o}_{1:5};\boldsymbol{\lambda})).$$

Likelihoods and their derivatives,

- for all T segments starting at $\tau = 1$, in $\Theta(T)$ time.
- for all $\Theta(T^2)$ segments, in $\Theta(T^2)$ time.

Experimental results

AURORA 2: noise-corrupted TIDIGITS

- Minimum Bayes risk training on large lattices.
- Results averaged over 0-20 dB.

System	Segmentation	Test set A	B	C	Average
HMM		9.8	9.1	9.5	9.5
ℓ	HMM	8.1	7.4	8.2	7.8
	optimal	7.8	7.3	8.0	7.6
$\begin{bmatrix} \ell \\ \nabla\ell \end{bmatrix}$	HMM	7.0	6.6	7.6	7.0
	optimal	6.8	6.4	7.3	6.7

Outline

Conclusion

- Segmental features: generative score-spaces:
  - derivatives of likelihood;
  - introduce dependencies between observations.
- Extract features in quadratic time.
- Decoding in quadratic time.