

Exact Inference for Hidden Markov Models

Michael Gutmann

Probabilistic Modelling and Reasoning (INFR11134)
School of Informatics, University of Edinburgh

Spring Semester 2019

Recap

- ▶ Assuming a factorisation / set of statistical independencies allowed us to efficiently represent the pdf or pmf of random variables
- ▶ Factorisation can be exploited for inference
 - ▶ by using the distributive law
 - ▶ by re-using already computed quantities
- ▶ Inference for general factor graphs (variable elimination)
- ▶ Inference for factor trees
- ▶ Sum-product and max-product message passing

Program

1. Markov models
2. Inference by message passing

Program

1. Markov models

- Markov chains
- Transition distribution
- Hidden Markov models
- Emission distribution
- Mixture of Gaussians as special case

2. Inference by message passing

Applications of (hidden) Markov models

Markov and hidden Markov models have many applications, e.g.

- ▶ speech modelling (speech recognition)
- ▶ text modelling (natural language processing)
- ▶ gene sequence modelling (bioinformatics)
- ▶ spike train modelling (neuroscience)
- ▶ object tracking (robotics)

Markov chains

- ▶ Chain rule with ordering $x_1, \dots, x_d$

$$p(x_1, \dots, x_d) = \prod_{i=1}^d p(x_i | x_1, \dots, x_{i-1})$$

- ▶ If p satisfies ordered Markov property, the number of variables in the conditioning set can be reduced to a subset $\pi_i \subseteq \{x_1, \dots, x_{i-1}\}$
- ▶ Not all predecessors but only subset π_i is “relevant” for x_i .
- ▶ L -th order Markov chain: $\pi_i = \{x_{i-L}, \dots, x_{i-1}\}$

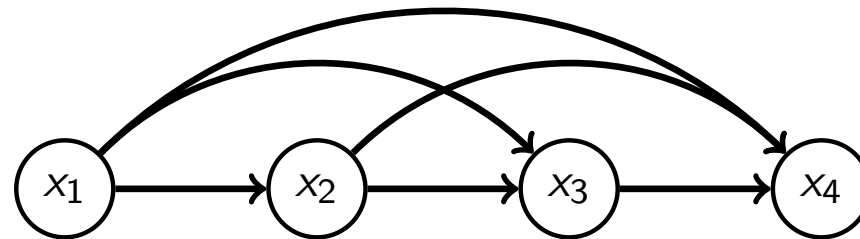
$$p(x_1, \dots, x_d) = \prod_{i=1}^d p(x_i | x_{i-L}, \dots, x_{i-1})$$

- ▶ 1st order Markov chain: $\pi_i = \{x_{i-1}\}$

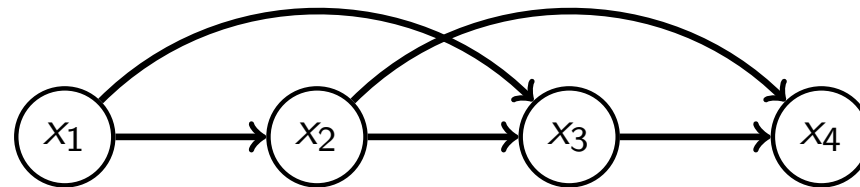
$$p(x_1, \dots, x_d) = \prod_{i=1}^d p(x_i | x_{i-1})$$

Markov chain — DAGs

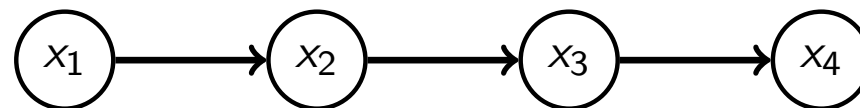
Chain rule



Second-order Markov chain



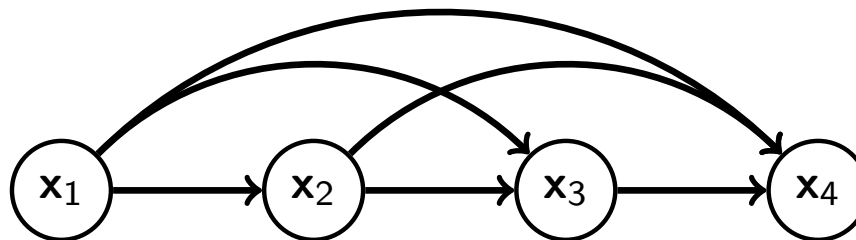
First-order Markov chain



Vector-valued Markov chains

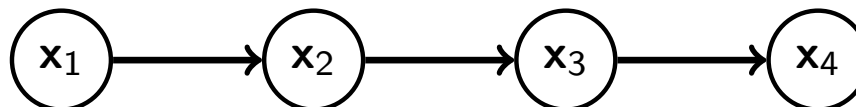
- ▶ While not explicitly discussed, the graphical models extend to vector-valued variables
- ▶ Chain rule with ordering $\mathbf{x}_1, \dots, \mathbf{x}_d$

$$p(\mathbf{x}_1, \dots, \mathbf{x}_d) = \prod_{i=1}^d p(\mathbf{x}_i | \mathbf{x}_1, \dots, \mathbf{x}_{i-1})$$



- ▶ 1st order Markov chain:

$$p(\mathbf{x}_1, \dots, \mathbf{x}_d) = \prod_{i=1}^d p(\mathbf{x}_i | \mathbf{x}_{i-1})$$



Modelling time series

- ▶ Index i may refer to time t
- ▶ L -th order Markov chain of length T :

$$p(x_1, \dots, x_T) = \prod_{t=1}^T p(x_t | x_{t-L}, \dots, x_{t-1})$$

Only the recent past of L time points $x_{t-L}, \dots, x_{t-1}$ is relevant for x_t

- ▶ 1st order Markov chain of length T :

$$p(x_1, \dots, x_T) = \prod_{t=1}^T p(x_t | x_{t-1})$$

Only the last time point x_{t-1} is relevant for x_t .

Transition distribution

(Consider 1st order Markov chain.)

- ▶ $p(x_i|x_{i-1})$ is called the transition distribution
- ▶ For discrete random variables, $p(x_i|x_{i-1})$ is defined by a transition matrix $\mathbf{A}^i$

$$p(x_i = k|x_{i-1} = k') = A_{k,k'}^i$$

- ▶ For continuous random variables, $p(x_i|x_{i-1})$ is a conditional pdf, e.g.

$$p(x_i|x_{i-1}) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(x_i - f_i(x_{i-1}))^2}{2\sigma_i^2}\right)$$

for some function f_i

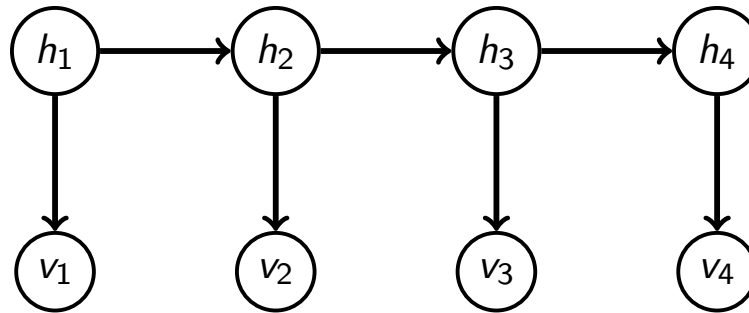
- ▶ Homogeneous Markov chain: $p(x_i|x_{i-1})$ does not depend on i , e.g.

$$\mathbf{A}^i = \mathbf{A} \quad \sigma_i = \sigma, \quad f_i = f$$

- ▶ Inhomogeneous Markov chain: $p(x_i|x_{i-1})$ does depend on i

Hidden Markov model

DAG:



- ▶ 1st order Markov chain on hidden (latent) variables h_i .
- ▶ Each visible (observed) variable v_i only depends on the corresponding hidden variable h_i
- ▶ Factorisation

$$p(h_{1:d}, v_{1:d}) = p(v_1|h_1)p(h_1) \prod_{i=2}^d p(v_i|h_i)p(h_i|h_{i-1})$$

- ▶ The visibles are d-connected if hiddens are not observed
- ▶ Visibles are d-separated (independent) given the hiddens
- ▶ The h_i model/explain all dependencies between the v_i

Emission distribution

- ▶ $p(v_i|h_i)$ is called the emission distribution
- ▶ Discrete-valued v_i and h_i :
 $p(v_i|h_i)$ can be represented as a matrix
- ▶ Discrete-valued v_i and continuous-valued h_i :
 $p(v_i|h_i)$ is a conditional pmf.
- ▶ Continuous-valued v_i : $p(v_i|h_i)$ is a density
- ▶ As for the transition distribution, the emission distribution $p(v_i|h_i)$ may depend on i or not.
- ▶ If neither the transition nor the emission distribution depend on i , we have a stationary (or homogeneous) hidden Markov model.

Gaussian emission model with discrete-valued latents

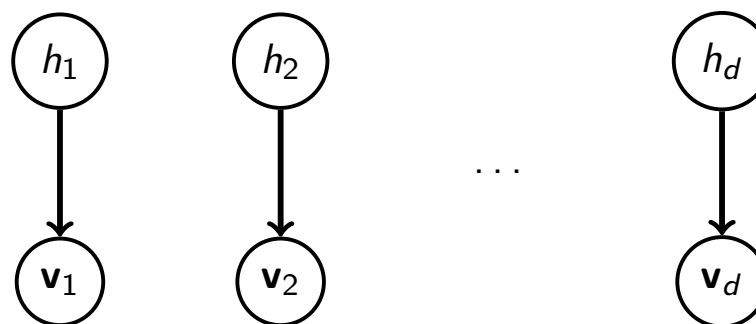
- ▶ Special case: $h_i \perp\!\!\!\perp h_{i-1}$, and $\mathbf{v}_i \in \mathbb{R}^m, h_i \in \{1, \dots, K\}$

$$p(h = k) = p_k$$

$$p(\mathbf{v} | h = k) = \frac{1}{|\det 2\pi \Sigma_k|^{1/2}} \exp \left(-\frac{1}{2} (\mathbf{v} - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1} (\mathbf{v} - \boldsymbol{\mu}_k) \right)$$

for all h_i and $\mathbf{v}_i$.

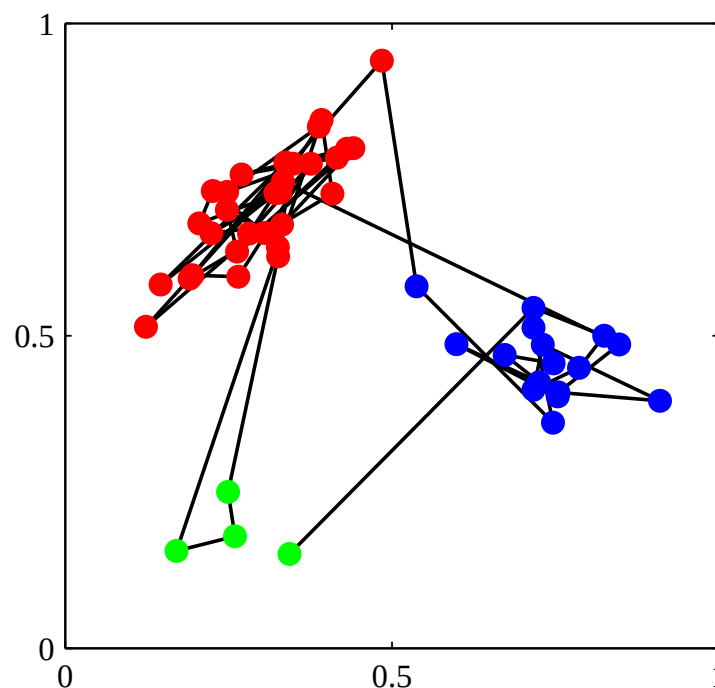
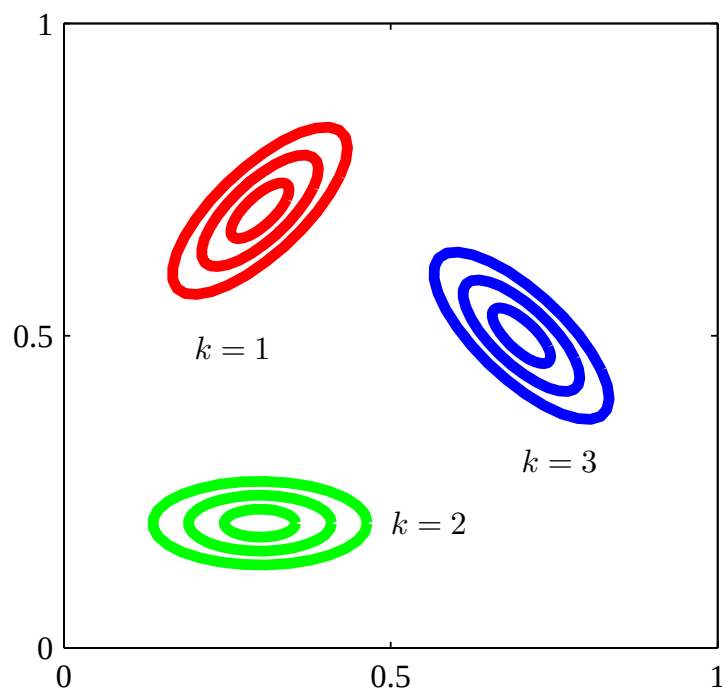
- ▶ DAG



- ▶ Corresponds to d iid draws from a Gaussian mixture model with K mixture components
 - ▶ Mean $\mathbb{E}[\mathbf{v} | h = k] = \boldsymbol{\mu}_k$
 - ▶ Covariance matrix $\mathbb{V}[\mathbf{v} | h = k] = \Sigma_k$

Gaussian emission model with discrete-valued latents

The HMM is a generalisation of the Gaussian mixture model where cluster membership at “time” i (the value of h_i) generally depends on cluster membership at “time” $i - 1$ (the value of h_{i-1}).



Example for $\mathbf{v}_i \in \mathbb{R}^2$, $h_i \in \{1, 2, 3\}$. Left: $p(\mathbf{v}|h = k)$. Right: samples

(Bishop, Figure 13.8)

Program

1. Markov models

- Markov chains
- Transition distribution
- Hidden Markov models
- Emission distribution
- Mixture of Gaussians as special case

2. Inference by message passing

Program

1. Markov models

2. Inference by message passing

- Inference: filtering, prediction, smoothing, Viterbi
- Filtering: Sum-product message passing yields the alpha-recursion from the HMM literature
- Smoothing: Sum-product message passing yields the alpha-beta recursion from the HMM literature
- Sum-product message passing for prediction, inference of most likely hidden path, and for inference of joint distributions

The classical inference problems

(Considering the index i to refer to time t)

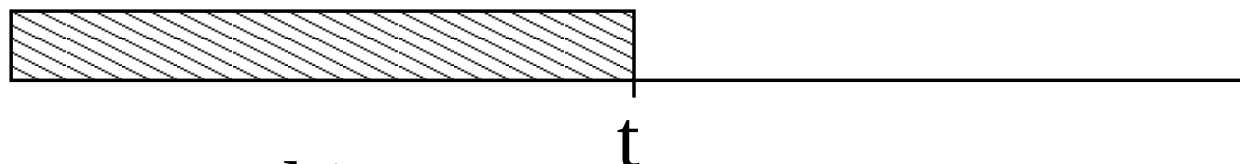
Filtering	(Inferring the present)	$p(h_t v_{1:t})$	
Smoothing	(Inferring the past)	$p(h_t v_{1:u})$	$t < u$
Prediction	(Inferring the future)	$p(h_t v_{1:u})$	$t > u$
Most likely Hidden path	(Viterbi alignment)	$\operatorname{argmax}_{h_{1:t}} p(h_{1:t} v_{1:t})$	

For prediction, one is also often interested in $p(v_t | v_{1:u})$ for $t > u$.

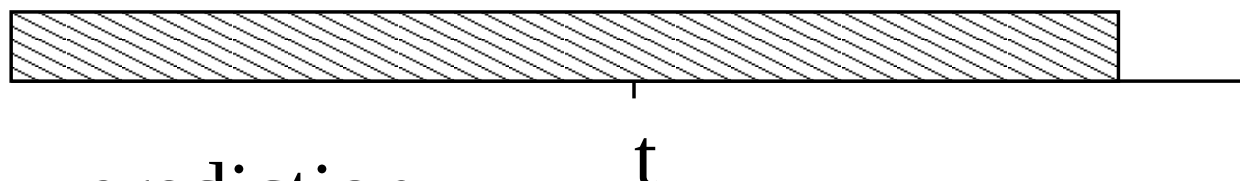
(slide courtesy of David Barber)

The classical inference problems

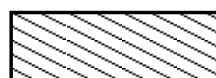
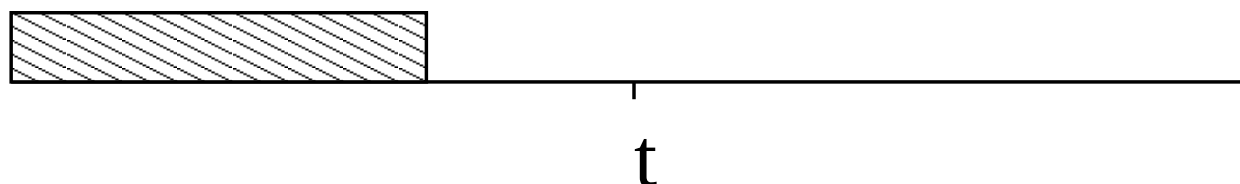
filtering



smoothing



prediction



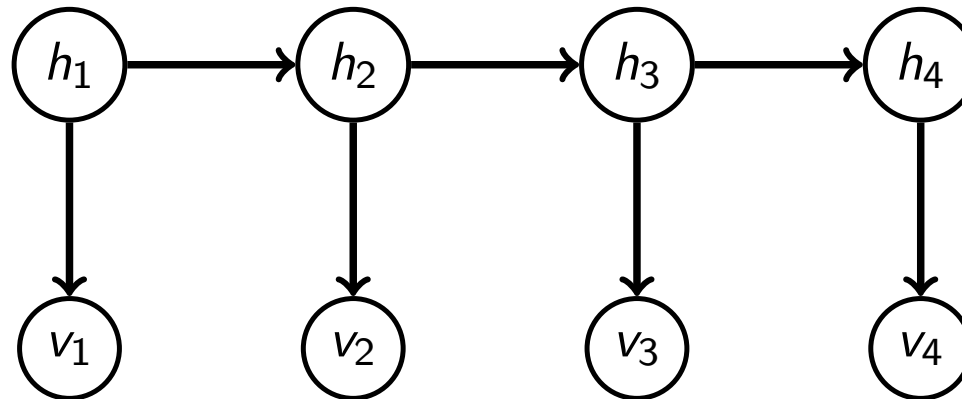
denotes the extent of data
available

(slide courtesy of Chris Williams)

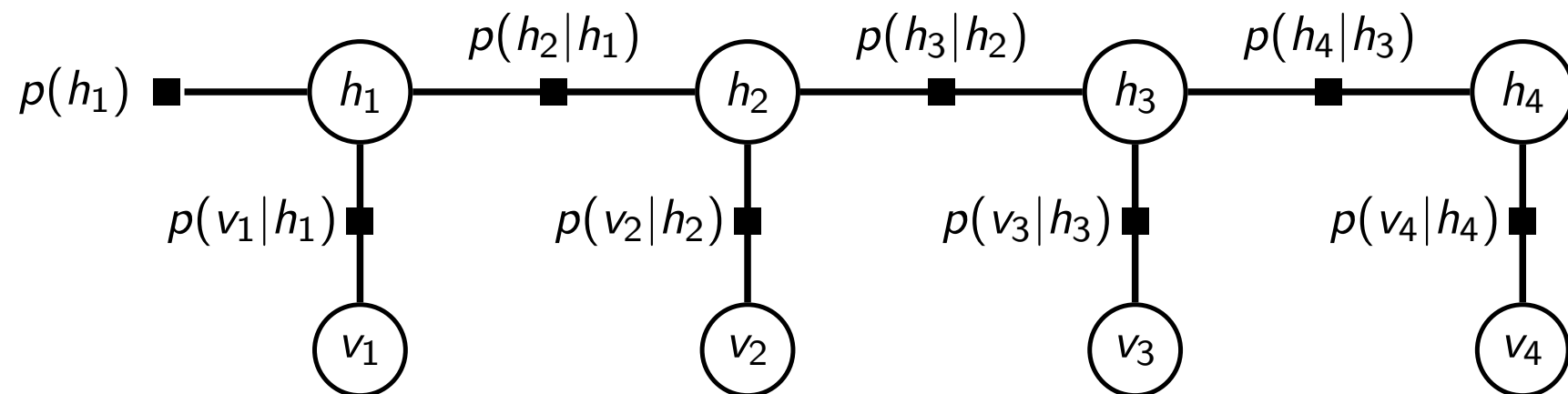
Factor graph for hidden Markov model

(see tutorial 3)

DAG:

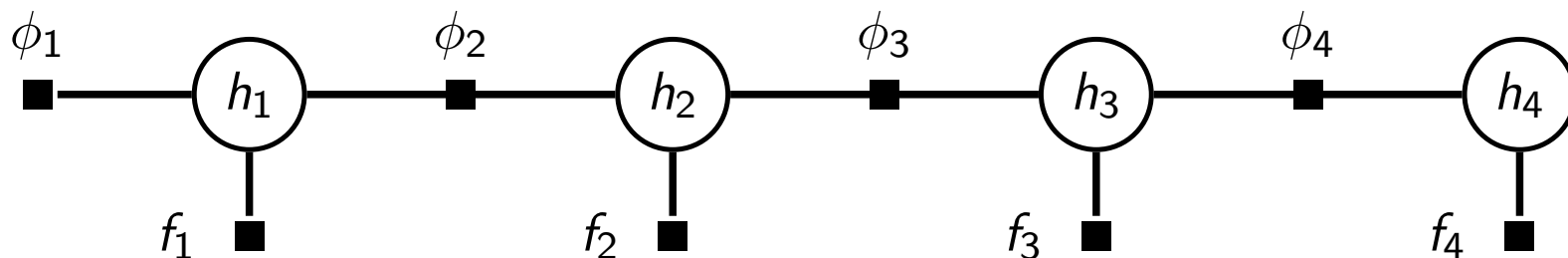


Factor graph:

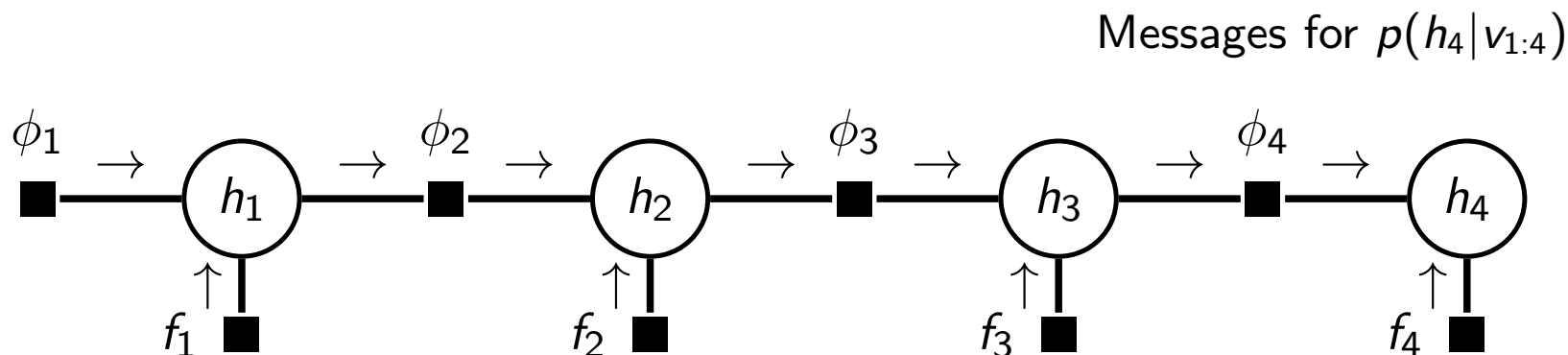


Filtering $p(h_t|v_{1:t})$

- ▶ When computing $p(h_t|v_{1:t})$, the $v_{1:t} = (v_1, \dots, v_t)$ are assumed known and are kept fixed
- ▶ Factors $p(v_s|h_s)$ depend on h_s only ($s = 1, \dots, t$).
- ▶ Different options (give the same results):
 - ▶ Work with (combined) factors
 $\phi_s(h_s, h_{s-1}) \propto p(v_s|h_s)p(h_s|h_{s-1})$ and $\phi_1(h_1) = p(v_1|h_1)p(h_1)$.
 - ▶ Work with factors $\phi_s(h_s, h_{s-1}) = p(h_s|h_{s-1})$, $f_s(h_s) = p(v_s|h_s)$, and $\phi_1(h_1) = p(h_1)$.
- ▶ Factor graph for second option



Filtering $p(h_t|v_{1:t})$



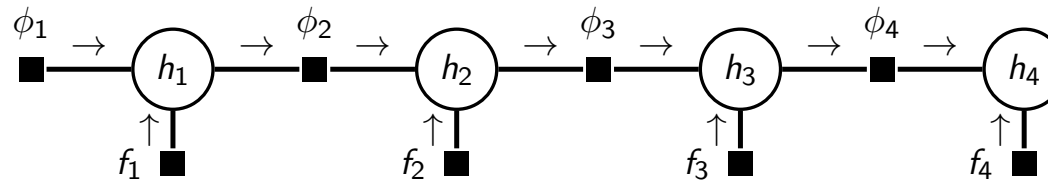
Marginal posterior:

$$p(h_t|v_{1:t}) \propto \mu_{\phi_t \rightarrow h_t}(h_t) \mu_{f_t \rightarrow h_t}(h_t)$$

Messages:

- ▶ $\mu_{f_i \rightarrow h_i}(h_i) = f_i(h_i)$ and $\mu_{\phi_1 \rightarrow h_1}(h_1) = \phi_1(h_1)$
- ▶ $\mu_{h_1 \rightarrow \phi_2}(h_1) = \mu_{\phi_1 \rightarrow h_1}(h_1) \cdot \mu_{f_1 \rightarrow h_1}(h_1)$
- ▶ $\mu_{\phi_2 \rightarrow h_2}(h_2) = \sum_{h_1} \phi_2(h_2, h_1) \mu_{h_1 \rightarrow \phi_2}(h_1)$
- ▶ $\vdots$
- ▶ $\mu_{\phi_s \rightarrow h_s}(h_s) = \sum_{h_{s-1}} \phi_s(h_s, h_{s-1}) \mu_{h_{s-1} \rightarrow \phi_s}(h_{s-1})$
- ▶ $\mu_{h_s \rightarrow \phi_{s+1}}(h_s) = \mu_{\phi_s \rightarrow h_s}(h_s) \cdot \mu_{f_s \rightarrow h_s}(h_s)$

Filtering $p(h_t|v_{1:t})$



- Recursion:

$$\mu_{h_1 \rightarrow \phi_2}(h_1) = \phi_1(h_1) \cdot f_1(h_1)$$

$$\mu_{\phi_s \rightarrow h_s}(h_s) = \sum_{h_{s-1}} \phi_s(h_s, h_{s-1}) \mu_{h_{s-1} \rightarrow \phi_s}(h_{s-1})$$

$$\mu_{h_s \rightarrow \phi_{s+1}}(h_s) = \mu_{\phi_s \rightarrow h_s}(h_s) \cdot \mu_{f_s \rightarrow h_s}(h_s)$$

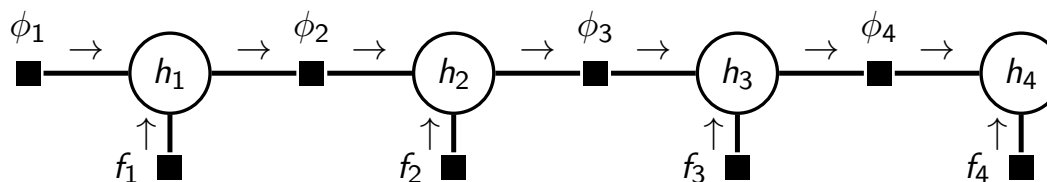
- Inserting the definition of the factors gives:

$$\mu_{h_1 \rightarrow \phi_2}(h_1) = p(h_1) \cdot p(v_1|h_1)$$

$$\mu_{\phi_s \rightarrow h_s}(h_s) = \sum_{h_{s-1}} p(h_s|h_{s-1}) \mu_{h_{s-1} \rightarrow \phi_s}(h_{s-1})$$

$$\mu_{h_s \rightarrow \phi_{s+1}}(h_s) = \mu_{\phi_s \rightarrow h_s}(h_s) \cdot p(v_s|h_s)$$

Filtering $p(h_t|v_{1:t})$



- Write recursion in terms of $\mu_{h_s \rightarrow \phi_{s+1}}$ only

$$\mu_{h_1 \rightarrow \phi_2}(h_1) = p(h_1) \cdot p(v_1|h_1)$$

$$\mu_{h_s \rightarrow \phi_{s+1}}(h_s) = p(v_s|h_s) \sum_{h_{s-1}} p(h_s|h_{s-1}) \mu_{h_{s-1} \rightarrow \phi_s}(h_{s-1})$$

- Called “alpha-recursion”: With $\alpha(h_s) = \mu_{h_s \rightarrow \phi_{s+1}}(h_s)$

$$\alpha(h_1) = p(h_1) \cdot p(v_1|h_1)$$

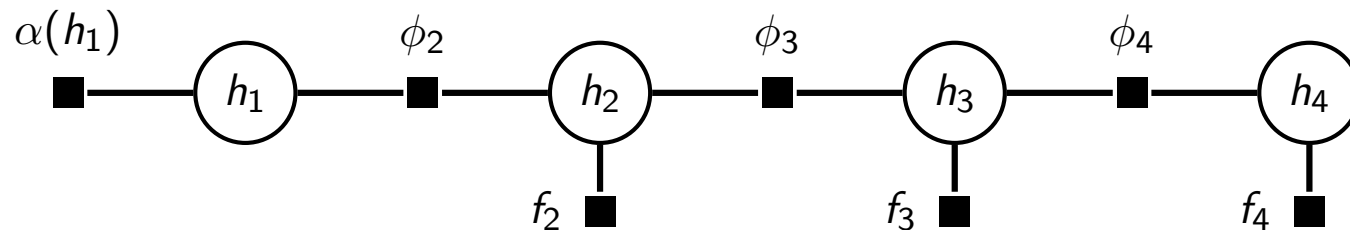
$$\alpha(h_s) = p(v_s|h_s) \sum_{h_{s-1}} p(h_s|h_{s-1}) \alpha(h_{s-1})$$

- Marginal posterior:

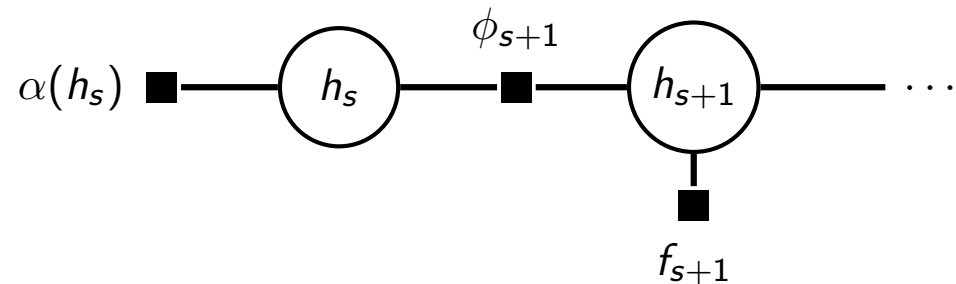
$$p(h_t|v_{1:t}) \propto \alpha(h_t)$$

Filtering $p(h_t|v_{1:t})$ – more on the alpha-recursion

- ▶ $\alpha(h_s) = \mu_{h_s \rightarrow \phi_{s+1}}(h_s)$ is an effective factor.
- ▶ $\alpha(h_1) = p(h_1)p(v_1|h_1) = p(h_1, v_1) \propto p(h_1|v_1)$



- ▶ For $\alpha(h_s)$



- ▶ We now prove by induction that

$$\alpha(h_s) = p(h_s, v_{1:s}) \propto p(h_s|v_{1:s})$$

Filtering $p(h_t|v_{1:t})$ – more on the alpha-recursion

$$\alpha(h_s) = p(v_s|h_s) \sum_{h_{s-1}} p(h_s|h_{s-1})\alpha(h_{s-1})$$

- ▶ Independencies in the model: $p(h_s|h_{s-1}) = p(h_s|h_{s-1}, v_{1:s-1})$
- ▶ With $\alpha(h_{s-1}) = p(h_{s-1}, v_{1:s-1})$ (holds for $s = 2$!)

$$\begin{aligned} \sum_{h_{s-1}} p(h_s|h_{s-1})\alpha(h_{s-1}) &= \sum_{h_{s-1}} p(h_s|h_{s-1}, v_{1:s-1})p(h_{s-1}, v_{1:s-1}) \\ &= \sum_{h_{s-1}} p(h_s, h_{s-1}, v_{1:s-1}) \\ &= p(h_s, v_{1:s-1}) \end{aligned}$$

- ▶ Independencies in the model: $p(v_s|h_s) = p(v_s|h_s, v_{1:s-1})$

$$\begin{aligned} \alpha(h_s) &= p(v_s|h_s, v_{1:s-1})p(h_s, v_{1:s-1}) \\ &= p(h_s, v_{1:s}) \end{aligned}$$

which completes the proof.

Filtering $p(h_t|v_{1:t})$ – more on the alpha-recursion

- ▶ This kind of approach allows one to obtain the alpha-recursion without message passing (see Barber).
- ▶ Interpretation of the alpha-recursion in terms of “prediction and correction”

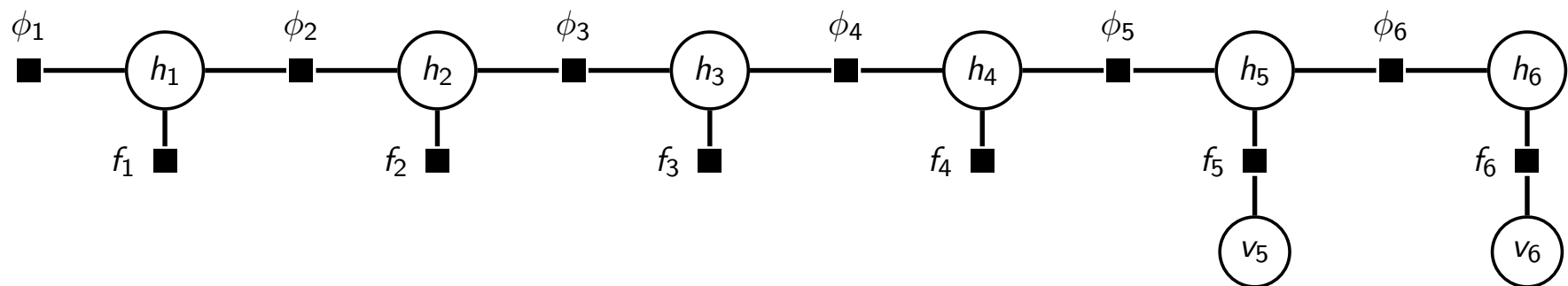
$$\begin{aligned}\alpha(h_s) &= p(v_s|h_s) \sum_{h_{s-1}} p(h_s|h_{s-1})\alpha(h_{s-1}) \\ &= p(v_s|h_s)p(h_s, v_{1:s-1}) \\ &\propto \underbrace{p(v_s|h_s)}_{\text{correction}} \underbrace{p(h_s|v_{1:s-1})}_{\text{prediction}} \\ &\propto p(h_s|v_{1:s})\end{aligned}$$

- ▶ The correction term updates the predictive distribution of h_s given $v_{1:s-1}$ to include the new data v_s .

Smoothing $p(h_t|v_{1:u}), t < u$

Consider:

- ▶ Hidden Markov model with variables $(h_1, \dots, h_6, v_1, \dots, v_6)$
- ▶ Observed $v_{1:4} = (v_1, \dots, v_4)$
- ▶ Interest: $p(h_2|v_{1:4})$



Factor graph with factors ϕ_i and $f_1, \dots, f_4$ defined as before. Factors f_5 and f_6 are: $f_5(h_5, v_5) = p(v_5|h_5)$ and $f_6(h_6, v_6) = p(v_6|h_6)$.

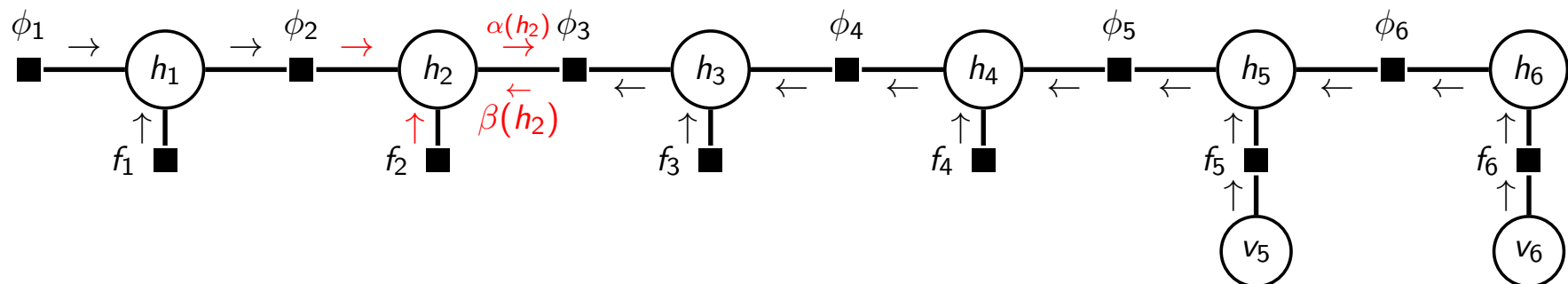
Smoothing $p(h_t|v_{1:u}), t < u$

- ▶ $p(h_2|v_{1:4})$ is given by incoming messages

$$p(h_2|v_{1:4}) \propto \underbrace{\mu_{\phi_2 \rightarrow h_2}(h_2) \mu_{f_2 \rightarrow h_2}(h_2)}_{\mu_{h_2 \rightarrow \phi_3}(h_2) = \alpha(h_2)} \mu_{\phi_3 \rightarrow h_2}(h_2)$$

- ▶ Denote $\mu_{\phi_3 \rightarrow h_2}(h_2)$ by $\beta(h_2)$:

$$p(h_2|v_{1:4}) \propto \alpha(h_2) \beta(h_2)$$



Smoothing $p(h_t|v_{1:u}), t < u$

- ▶ We can compute $\beta(h_2)$ by sum-product message passing.
- ▶ Let $\beta(h_s) = \mu_{\phi_{s+1} \rightarrow h_s}(h_s)$, then (see tutorial 5)

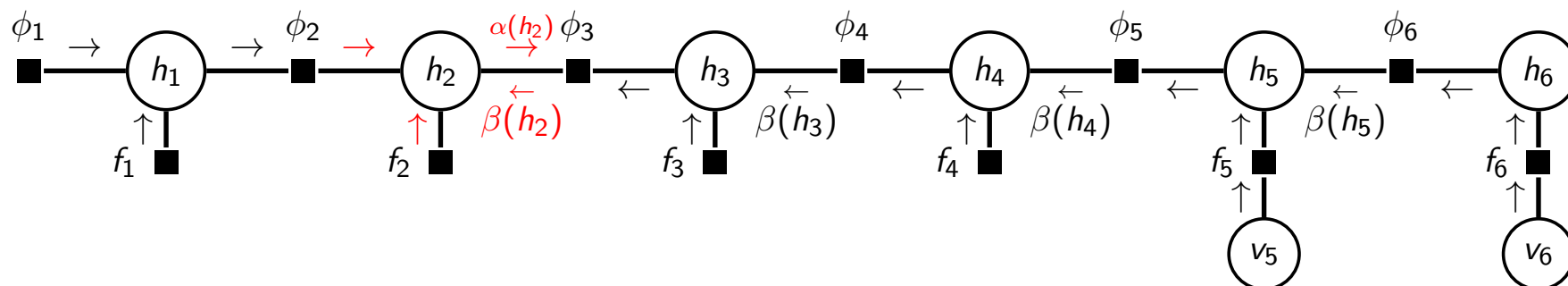
$$\beta(h_4) = \beta(h_5) = 1$$

$$\beta(h_3) = \sum_{h_4} \underbrace{p(h_4|h_3)}_{\phi_4} \underbrace{p(v_4|h_4)}_{f_4} \underbrace{\beta(h_4)}_1$$

⋮

$$\beta(h_s) = \sum_{h_{s+1}} \underbrace{p(h_{s+1}|h_s)}_{\phi_{s+1}} \underbrace{p(v_{s+1}|h_{s+1})}_{f_{s+1}} \beta(h_{s+1}) \quad (s < u)$$

- ▶ From independencies: $\beta(h_s) = p(v_{s+1:u}|h_s)$ (see Barber 23.2.3)



Smoothing $p(h_t|v_{1:u})$, $t < u$

- ▶ Recursive computation of $\beta(h_s)$ via message passing is known as “beta-recursion” in the HMM literature
- ▶ Smoothing via “alpha-beta recursion”

$$p(h_t|v_{1:u}) \propto \alpha(h_t)\beta(h_t)$$

$$\alpha(h_s) = p(v_s|h_s) \sum_{h_{s-1}} p(h_s|h_{s-1})\alpha(h_{s-1})$$

$$\alpha(h_1) = p(h_1)p(v_1|h_1) \propto p(h_1|v_1)$$

$$\beta(h_s) = \sum_{h_{s+1}} p(h_{s+1}|h_s)p(v_{s+1}|h_{s+1})\beta(h_{s+1})$$

$$\beta(h_u) = 1$$

- ▶ Also known as forward-backward algorithm.
- ▶ Due to correspondence to message passing: Knowing all $\alpha(h_s), \beta(h_s) \iff$ knowing all marginals and all joints of neighbouring latents given the observed data $v_{1:u}$.

Prediction, most likely hidden path, and joint distribution

- ▶ Sum-product algorithm can similarly be used for
 - ▶ prediction: $p(h_t|v_{1:u})$ and $p(v_t|v_{1:u})$, with $t > u$
 - ▶ inference of the most likely hidden path: $\operatorname{argmax}_{h_{1:t}} p(h_{1:t}|v_{1:t})$
 - ▶ computing pairwise marginals $p(h_t, h_{t+1}|v_{1:u})$, $u \geq t$ or $u < t$.
- ▶ Can be written in terms of $\alpha(h_t)$ and $\beta(h_t)$
- ▶ See Barber Section 23.2
(does not use message passing)

Program recap

1. Markov models

- Markov chains
- Transition distribution
- Hidden Markov models
- Emission distribution
- Mixture of Gaussians as special case

2. Inference by message passing

- Inference: filtering, prediction, smoothing, Viterbi
- Filtering: Sum-product message passing yields the alpha-recursion from the HMM literature
- Smoothing: Sum-product message passing yields the alpha-beta recursion from the HMM literature
- Sum-product message passing for prediction, inference of most likely hidden path, and for inference of joint distributions