Introduction to Probabilistic Graphical Models

Christoph Lampert

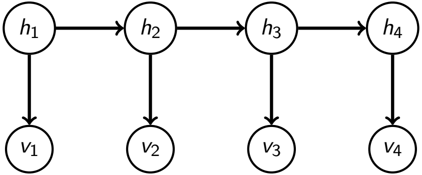
IST Austria (Institute of Science and Technology Austria)

Inference in Hidden Markov Models

Hidden Markov Models

Reminder: a hidden Markov model (HMM) consists of

- a discrete Markov chain of hidden (or 'latent') variables $h_{1:T}$
- one observable (continous or discrete) variable v_i for each hidden variable h_i



$$p(h_{1:T}, v_{1:T}) = p(v_1|h_1)p(h_1)\prod_{t=2}^{T} p(v_t|h_t)p(h_t|h_{t-1})$$

We call the HMM stationary if

- the transition distribution $p(h_{t+1} = i'|h_t = i)$ and the emission distribution $p(v_t = j|h_t = i)$ do not depend on the position t, but only one the values i, i' and j

HMM parameters

Transition Distribution

For a stationary HMM the transition distribution $p(h_{t+1}|h_t)$ is defined by the $H \times H$ transition matrix

$$A_{i',i} = p(h_{t+1} = i'|h_t = i)$$

and an initial distribution

$$a_i = p(h_1 = i).$$

Emission Distribution

For a stationary HMM and emission distribution $p(v_t|h_t)$ with discrete states $v_t \in \{1, \dots, V\}$, we define a $V \times H$ emission matrix

$$B_{i,j} = p(v_t = i|h_t = j)$$

For continuous outputs, h_t selects one of H possible output distributions $p(v_t|h_t)$, $h_t \in \{1, \dots, H\}$.

The classical inference problems

Filtering	(Inferring the present)	$p(h_t \mid v_{1:t})$	
Prediction	(Inferring the future)	$p(h_t \mid v_{1:s})$	for $t > s$
	sometimes also	$p(v_t \mid v_{1:s})$	for $t > s$
Smoothing	(Inferring the past)	$p(h_t \mid v_{1:u})$	for $t < u$
Likelihood		$p(v_{1:T})$	
Most likely Hidden path	(Viterbi alignment)	$\text{argmax}\, h_{1:T}\, p(h_{1:T} \mid v_{1:T})$	
Learning	(Parameter estimation)	$\mathcal{D} \rightarrow A_{i,i'}, a_i, B_{i,j}$	

The Burglar Scenario

You're asleep upstairs in your house and awoken by noises from downstairs. You realise that a burglar is on the ground floor and attempt to understand where he his from listening to his movements.

The Burglar Scenario

You're asleep upstairs in your house and awoken by noises from downstairs. You realise that a burglar is on the ground floor and attempt to understand where he his from listening to his movements.

The HMM view

- You mentally partition the ground floor into a 5×5 grid.
- For each grid position you know the probability that if someone is in that position the floorboard will creak.
- Similarly you know for each position the probability that someone will bump into something in the dark.
- The floorboard creaking and bumping into objects can occur independently.
- In addition you assume that the burglar will move only one grid square – forwards, backwards, left or right in a single timestep.

Can you infer the burglar's position from the sounds?

The Burglar Scenario: Example

'creaks'

'bumps'

observations:

creaks	n	y	n	y	n	y	y	y	y	y
bumps	y	n	n	y	n	y	n	n	y	y

- latent variable $h_t \in \{1, \ldots, 25\}$ denotes the positions on 5×5 grid
  dark squares means probability 0.9, light means probability 0.1
- observed variables: $v_t = (c_t, b_t) \in \{(n, n), (n, y), (y, n), (y, y)\}$
- observed probability factorizes $p(v|h) = p(c|h)p(b|h)$

Burglar

Localising the burglar through time for 10 time steps

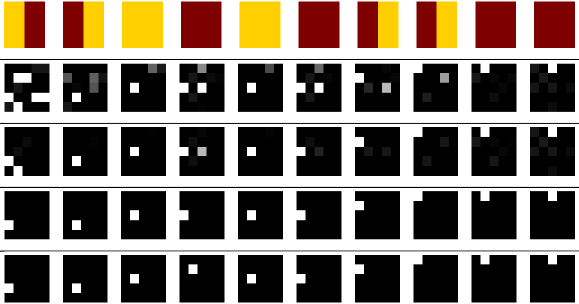
(a) Creaks and Bumps v_t

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

(c) Smoothing $p(h_t|v_{1:T})$

(d) Viterbi $\text{argmax}_{h_{1:T}}\, p(h_{1:T}|v_{1:T})$

(e) True Burglar position $h^*_{1:T}$



Note:

- (b) is computed on-the-fly in every time step
- (c) and (d) are computed offline after all observations are available

Real-world example

https://www.youtube.com/watch?v=4Z3shNPOdQA

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

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

$$
\begin{aligned}
p(h_t, v_{1:t}) &= \sum_{h_{t-1}} p(h_t, h_{t-1}, v_{1:t-1}, v_t) \\
&= \sum_{h_{t-1}} p(v_t|\cancel{v_{1:t-1}}, h_t, \cancel{h_{t-1}})p(h_t|\cancel{v_{1:t-1}}, h_{t-1})p(v_{1:t-1}, h_{t-1}) \\
&= \sum_{h_{t-1}} p(v_t|h_t)p(h_t|h_{t-1})p(h_{t-1}, v_{1:t-1})
\end{aligned}
$$

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

$$
\begin{aligned}
p(h_t, v_{1:t}) &= \sum_{h_{t-1}} p(h_t, h_{t-1}, v_{1:t-1}, v_t) \\
&= \sum_{h_{t-1}} p(v_t|\cancel{v_{1:t-1}}, h_t, \cancel{h_{t-1}})p(h_t|\cancel{v_{1:t-1}}, h_{t-1})p(v_{1:t-1}, h_{t-1}) \\
&= \sum_{h_{t-1}} p(v_t|h_t)p(h_t|h_{t-1})p(h_{t-1}, v_{1:t-1})
\end{aligned}
$$

Hence if we define $\alpha(h_t) \equiv p(h_t, v_{1:t})$ the above gives the α-recursion

$$
\alpha(h_t) = \overbrace{p(v_t|h_t)}^{\text{corrector}} \overbrace{\sum_{h_{t-1}} p(h_t|h_{t-1})\alpha(h_{t-1})}^{\text{predictor}}, \quad \text{with} \quad \alpha(h_1) = p(h_1, v_1) = p(v_1|h_1)p(h_1)
$$

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

$$\begin{aligned}
p(h_t, v_{1:t}) &= \sum_{h_{t-1}} p(h_t, h_{t-1}, v_{1:t-1}, v_t) \\
&= \sum_{h_{t-1}} p(v_t|\cancel{v_{1:t-1}}, h_t, \cancel{h_{t-1}})p(h_t|\cancel{v_{1:t-1}}, h_{t-1})p(v_{1:t-1}, h_{t-1}) \\
&= \sum_{h_{t-1}} p(v_t|h_t)p(h_t|h_{t-1})p(h_{t-1}, v_{1:t-1})
\end{aligned}$$

Hence if we define $\alpha(h_t) \equiv p(h_t, v_{1:t})$ the above gives the α-recursion

$$\alpha(h_t) = \overbrace{p(v_t|h_t)}^{\text{corrector}} \overbrace{\sum_{h_{t-1}} p(h_t|h_{t-1})\alpha(h_{t-1})}^{\text{predictor}}, \quad \text{with} \quad \alpha(h_1) = p(h_1, v_1) = p(v_1|h_1)p(h_1)$$

Filtered posterior follows by normalization: $p(h_t|v_{1:t}) = \frac{p(h_t, v_{1:t})}{\sum_{\bar{h}_t} p(\bar{h}_t, v_{1:t})} = \frac{\alpha(h_t)}{\sum_{\bar{h}_t} \alpha(\bar{h}_t)}$

Likelihood $p(v_{1:T})$

Likelihood $p(v_{1:T})$

$$p(v_{1:T}) = \sum_{h_T} p(h_T, v_{1:T}) = \sum_{h_T} \alpha(h_T)$$

Smoothing $p(h_t|v_{1:T})$

Smoothing $p(h_t|v_{1:T})$

To compute the smoothed quantity we consider how h_t partitions the series into the past and future:

$$\begin{aligned} p(h_t, v_{1:T}) &= p(h_t, v_{1:t}, v_{t+1:T}) \\ &= \underbrace{p(h_t, v_{1:t})}_{\text{past}} \underbrace{p(v_{t+1:T}|h_t, v_{1:t})}_{\text{future}} = \alpha(h_t)\beta(h_t) \end{aligned}$$

Smoothing $p(h_t|v_{1:T})$

To compute the smoothed quantity we consider how h_t partitions the series into the past and future:

$$
\begin{aligned}
p(h_t, v_{1:T}) &= p(h_t, v_{1:t}, v_{t+1:T}) \\
&= \underbrace{p(h_t, v_{1:t})}_{\text{past}} \underbrace{p(v_{t+1:T}|h_t, v_{1:t})}_{\text{future}} = \alpha(h_t)\beta(h_t)
\end{aligned}
$$

Forward. The term $\alpha(h_t)$ is obtained from the 'forward' α recursion.

Backward. The term $\beta(h_t)$ we will obtain using a 'backward' β recursion as we show next.

Smoothing $p(h_t|v_{1:T})$

To compute the smoothed quantity we consider how h_t partitions the series into the past and future:

$$\begin{aligned} p(h_t, v_{1:T}) &= p(h_t, v_{1:t}, v_{t+1:T}) \\ &= \underbrace{p(h_t, v_{1:t})}_{\text{past}} \underbrace{p(v_{t+1:T}|h_t, v_{1:t})}_{\text{future}} = \alpha(h_t)\beta(h_t) \end{aligned}$$

Forward. The term $\alpha(h_t)$ is obtained from the 'forward' α recursion.

Backward. The term $\beta(h_t)$ we will obtain using a 'backward' β recursion as we show next.

The forward and backward recursions are independent and may therefore be run in parallel, with their results combined to obtain the smoothed posterior.

$$p(h_t|v_{1:T}) \equiv \gamma(h_t) = \frac{\alpha(h_t)\beta(h_t)}{\sum_{\bar{h}_t} \alpha(\bar{h}_t)\beta(\bar{h}_t)}$$

"Parallel Smoothing"

The β recursion

$$
\begin{aligned}
p(v_{t:T}|h_{t-1}) &= \sum_{h_t} p(v_t, v_{t+1:T}, h_t|h_{t-1}) \\
&= \sum_{h_t} p(v_t|\cancel{v_{t+1:T}}, h_t, \cancel{h_{t-1}})p(v_{t+1:T}, h_t|h_{t-1}) \\
&= \sum_{h_t} p(v_t|h_t)p(v_{t+1:T}|h_t, \cancel{h_{t-1}})p(h_t|h_{t-1})
\end{aligned}
$$

Defining $\beta(h_t) \equiv p(v_{t+1:T}|h_t)$ gives the β-recursion

$$
\beta(h_{t-1}) = \sum_{h_t} p(v_t|h_t)p(h_t|h_{t-1})\beta(h_t), \quad \text{for } 2 \leq t \leq T \quad \text{and} \quad \beta(h_T) = 1.
$$

Together the $\alpha - \beta$ recursions are called the Forward-Backward algorithm.

Smoothing $p(h_t|v_{1:T})$

"Correction Smoothing":

$$p(h_t|v_{1:T}) = \sum_{h_{t+1}} p(h_t, h_{t+1}|v_{1:T}) = \sum_{h_{t+1}} p(h_t|h_{t+1}, v_{1:t}, \cancel{v_{t+1:T}})p(h_{t+1}|v_{1:T})$$

This gives a recursion for $\gamma(h_t) \equiv p(h_t|v_{1:T})$:

$$\gamma(h_t) = \sum_{h_{t+1}} p(h_t|h_{t+1}, v_{1:t})\gamma(h_{t+1})$$

with $\gamma(h_T) \propto \alpha(h_T)$. The term $p(h_t|h_{t+1}, v_{1:t})$ may be computed using the filtered results $p(h_t|v_{1:t})$:

$$p(h_t|h_{t+1}, v_{1:t}) \propto p(h_{t+1}, h_t|v_{1:t}) \propto p(h_{t+1}|h_t)p(h_t|v_{1:t})$$

where the proportionality constant is found by normalisation. This is sequential since we need to first complete the α recursions, after which the γ recursion may begin. This 'corrects' the filtered result. Interestingly, once filtering has been carried out, the evidential states $v_{1:T}$ are not needed during the subsequent γ recursion.

Computing the pairwise marginal $p(h_t, h_{t+1}|v_{1:T})$

To implement the EM algorithm for learning, we require terms such as $p(h_t, h_{t+1}|v_{1:T})$.

$$\begin{aligned} p(h_t, h_{t+1}|v_{1:T}) &\propto p(v_{1:t}, v_{t+1}, v_{t+2:T}, h_{t+1}, h_t) \\ &= p(v_{t+2:T}|\cancel{v_{1:t}, v_{t+1}, h_t}, h_{t+1})p(v_{1:t}, v_{t+1}, h_{t+1}, h_t) \\ &= p(v_{t+2:T}|h_{t+1})p(v_{t+1}|\cancel{v_{1:t}, h_t}, h_{t+1})p(v_{1:t}, h_{t+1}, h_t) \\ &= p(v_{t+2:T}|h_{t+1})p(v_{t+1}|h_{t+1})p(h_{t+1}|\cancel{v_{1:t}}, h_t)p(v_{1:t}, h_t) \end{aligned}$$

After rearranging:

$$p(h_t, h_{t+1}|v_{1:T}) \propto \alpha(h_t)p(v_{t+1}|h_{t+1})p(h_{t+1}|h_t)\beta(h_{t+1})$$

Prediction

Predicting the future hidden variable:

$$p(h_{t+1}|v_{1:t}) =$$

Prediction

Predicting the future hidden variable:

$$p(h_{t+1}|v_{1:t}) = \sum_{h_t} p(h_{t+1}|h_t) \underbrace{p(h_t|v_{1:t})}_{\textit{filtering}}$$

Prediction

Predicting the future hidden variable:

$$p(h_{t+1}|v_{1:t}) = \sum_{h_t} p(h_{t+1}|h_t) \underbrace{p(h_t|v_{1:t})}_{\textit{filtering}}$$

Predicting the future observation The one-step ahead predictive distribution is given by

$$p(v_{t+1}|v_{1:t}) =$$

Prediction

Predicting the future hidden variable:

$$p(h_{t+1}|v_{1:t}) = \sum_{h_t} p(h_{t+1}|h_t) \underbrace{p(h_t|v_{1:t})}_{\textit{filtering}}$$

Predicting the future observation The one-step ahead predictive distribution is given by

$$p(v_{t+1}|v_{1:t}) = \sum_{h_{t+1}} p(v_{t+1}|h_{t+1}) \underbrace{p(h_{t+1}|v_{1:t})}_{\textit{prediction}}$$

Most likely joint state

The most likely path $h_{1:T}$ of $p(h_{1:T}|v_{1:T})$ is the same as the most likely state of

$$p(h_{1:T}, v_{1:T}) = \prod_t p(v_t|h_t)p(h_t|h_{t-1}) \qquad \text{with } h_0 = \emptyset$$

Consider

$$\max_{h_T} \prod_{t=1}^{T} p(v_t|h_t)p(h_t|h_{t-1})$$
$$= \left\{ \prod_{t=1}^{T-1} p(v_t|h_t)p(h_t|h_{t-1}) \right\} \underbrace{\max_{h_T} p(v_T|h_T)p(h_T|h_{T-1})}_{\mu(h_{T-1})}$$

The "message" $\mu(h_{T-1})$ conveys information from the end of the chain to the penultimate timestep.

Most likely joint state

We can continue in this manner, defining the recursion

$$\mu(h_{t-1}) = \max_{h_t} p(v_t|h_t)p(h_t|h_{t-1})\mu(h_t), \quad \text{for } 2 \leq t \leq T \quad \text{and} \quad \mu(h_T) = 1.$$

The effect of maximising over $h_2, \ldots, h_T$ is compressed into a message $\mu(h_1)$
$\rightarrow$ the first entry most likely state, h_1^*, is given by

$$h_1^* = \operatorname*{argmax}_{h_1} p(v_1|h_1)p(h_1)\mu(h_1)$$

Once computed, backtracking gives the remaining entries:

$$h_t^* = \operatorname*{argmax}_{h_t} p(v_t|h_t)p(h_t|h_{t-1}^*)\mu(h_t)$$

Learning Hidden Markov Models

Learning HMMs

Setting:

- given: data $\mathcal{V} = \{\mathbf{v}^1, \dots, \mathbf{v}^N\}$ of N sequences, each sequence $\mathbf{v}^N = v^N_{1:T_N}$ is of length T_n
- goal: maximum-likelihood of HMM parameters $\theta = (\mathbf{A}, \mathbf{B}, \mathbf{a})$, where
  - $\mathbf{A}$ is the HMM transition matrix, $p(h_{t+1}|h_t)$
  - $\mathbf{B}$ is the emission matrix, $p(v_t|h_t)$
  - $\mathbf{a}$ is the vector of initial state probabilities, $p(h_1)$.
- assumption: the sequences are i.i.d. (within sequences, data are still dependent, of course)
- assumption: the number of hidden states H and observable states V is known and finite

Learning HMMs

Setting:

- given: data $\mathcal{V} = \{\mathbf{v}^1, \dots, \mathbf{v}^N\}$ of N sequences, each sequence $\mathbf{v}^N = v^N_{1:T_N}$ is of length T_n
- goal: maximum-likelihood of HMM parameters $\theta = (\mathbf{A}, \mathbf{B}, \mathbf{a})$, where
  - $\mathbf{A}$ is the HMM transition matrix, $p(h_{t+1}|h_t)$
  - $\mathbf{B}$ is the emission matrix, $p(v_t|h_t)$
  - $\mathbf{a}$ is the vector of initial state probabilities, $p(h_1)$.
- assumption: the sequences are i.i.d. (within sequences, data are still dependent, of course)
- assumption: the number of hidden states H and observable states V is known and finite

Find θ that maximizes

$$p(\mathbf{v}^1, \dots, \mathbf{v}^N; \theta) = \sum_{\mathbf{h}^1, \dots, \mathbf{h}^N} p(\mathbf{v}^1, \dots, \mathbf{v}^N, \mathbf{h}^1, \dots, \mathbf{h}^N; \theta) = \prod_{n=1}^{N} \sum_{\mathbf{h}^n} p(\mathbf{v}^n, \mathbf{h}^n; \theta)$$

Learning HMMs

Setting:

- given: data $\mathcal{V} = \{\mathbf{v}^1, \ldots, \mathbf{v}^N\}$ of N sequences,
  each sequence $\mathbf{v}^N = v^N_{1:T_N}$ is of length T_n
- goal: maximum-likelihood of HMM parameters $\theta = (\mathbf{A}, \mathbf{B}, \mathbf{a})$, where
  - $\mathbf{A}$ is the HMM transition matrix, $p(h_{t+1}|h_t)$
  - $\mathbf{B}$ is the emission matrix, $p(v_t|h_t)$
  - $\mathbf{a}$ is the vector of initial state probabilities, $p(h_1)$.
- assumption: the sequences are i.i.d. (within sequences, data are still dependent, of course)
- assumption: the number of hidden states H and observable states V is known and finite

Find θ that maximizes

$$p(\mathbf{v}^1, \ldots, \mathbf{v}^N; \theta) = \sum_{\mathbf{h}^1, \ldots, \mathbf{h}^N} p(\mathbf{v}^1, \ldots, \mathbf{v}^N, \mathbf{h}^1, \ldots, \mathbf{h}^N; \theta) = \prod_{n=1}^{N} \sum_{\mathbf{h}^n} p(\mathbf{v}^n, \mathbf{h}^n; \theta)$$

How?

Learning HMMs

Setting:

- given: data $\mathcal{V} = \{\mathbf{v}^1, \dots, \mathbf{v}^N\}$ of N sequences, each sequence $\mathbf{v}^N = v^N_{1:T_N}$ is of length T_n
- goal: maximum-likelihood of HMM parameters $\theta = (\mathbf{A}, \mathbf{B}, \mathbf{a})$, where
  - $\mathbf{A}$ is the HMM transition matrix, $p(h_{t+1}|h_t)$
  - $\mathbf{B}$ is the emission matrix, $p(v_t|h_t)$
  - $\mathbf{a}$ is the vector of initial state probabilities, $p(h_1)$.
- assumption: the sequences are i.i.d. (within sequences, data are still dependent, of course)
- assumption: the number of hidden states H and observable states V is known and finite

Find θ that maximizes

$$p(\mathbf{v}^1, \dots, \mathbf{v}^N; \theta) = \sum_{\mathbf{h}^1, \dots, \mathbf{h}^N} p(\mathbf{v}^1, \dots, \mathbf{v}^N, \mathbf{h}^1, \dots, \mathbf{h}^N; \theta) = \prod_{n=1}^{N} \sum_{\mathbf{h}^n} p(\mathbf{v}^n, \mathbf{h}^n; \theta)$$

How? EM-algorithm (for HMMs called Baum-Welch algorithm for historic reasons)

Learning HMMs

Like for GMM, construct a lower bound using a distribution $q(\mathbf{h}^1, \dots, \mathbf{h}^N)$

$$\log p(\mathbf{v}^1, \dots, \mathbf{v}^N; \theta) = \log \sum_{\mathbf{h}^1, \dots, \mathbf{h}^N} p(\mathbf{v}^1, \dots, \mathbf{v}^N, \mathbf{h}^1, \dots, \mathbf{h}^N; \theta)$$

$$\leq \mathop{\mathbb{E}}_{(\mathbf{h}^1, \dots, \mathbf{h}^N) \sim q} \log p(\mathbf{v}^1, \dots, \mathbf{v}^N, \mathbf{h}^1, \dots, \mathbf{h}^N; \theta) - \mathop{\mathbb{E}}_{(\mathbf{h}^1, \dots, \mathbf{h}^N) \sim q} \log q(\mathbf{h}^1, \dots, \mathbf{h}^N)$$

EM algorithm:

initialize θ^0
for $t = 1, 2, \dots,$ until convergence **do**
$q^t \leftarrow \text{argmax}_q \; G(\theta^{t-1}, q)$ // E-step
$\theta^t \leftarrow \text{argmax}_\theta \; G(\theta, q^t)$ // M-step
end for

Learning HMMs

Like for GMM, construct a lower bound using a distribution $q(\mathbf{h}^1, \dots, \mathbf{h}^N)$

$$\log p(\mathbf{v}^1, \dots, \mathbf{v}^N; \theta) = \log \sum_{\mathbf{h}^1,\dots,\mathbf{h}^N} p(\mathbf{v}^1, \dots, \mathbf{v}^N, \mathbf{h}^1, \dots, \mathbf{h}^N; \theta)$$

$$\leq \mathop{\mathbb{E}}_{(\mathbf{h}^1,\dots,\mathbf{h}^N)\sim q} \log p(\mathbf{v}^1, \dots, \mathbf{v}^N, \mathbf{h}^1, \dots, \mathbf{h}^N; \theta) - \mathop{\mathbb{E}}_{(\mathbf{h}^1,\dots,\mathbf{h}^N)\sim q} \log q(\mathbf{h}^1, \dots, \mathbf{h}^N) \quad =: G(\theta, q)$$

EM algorithm:

initialize θ^0
for $t = 1, 2, \dots,$ until convergence **do**
  $q^t \leftarrow \text{argmax}_q \; G(\theta^{t-1}, q)$ // E-step
  $\theta^t \leftarrow \text{argmax}_\theta \; G(\theta, q^t)$ // M-step
end for

E-step, Part 1

$$q \leftarrow \underset{q}{\operatorname{argmax}} \; G(\theta^{t-1}, q)$$

- as for GMMs:

$$q^t \leftarrow p(\mathbf{h}^1, \dots, \mathbf{h}^N | \mathbf{v}^1, \dots, \mathbf{v}^N; \theta^{t-1}) \overset{i.i.d.}{=} \prod_{n=1}^{N} p(\mathbf{h}^n | \mathbf{v}^n; \theta^{t-1})$$

E-step, Part 1

$$q \leftarrow \underset{q}{\operatorname{argmax}} \; G(\theta^{t-1}, q)$$

- as for GMMs:

$$q^t \leftarrow p(\mathbf{h}^1, \ldots, \mathbf{h}^N | \mathbf{v}^1, \ldots, \mathbf{v}^N; \theta^{t-1}) \overset{i.i.d.}{=} \prod_{n=1}^{N} \underbrace{p(\mathbf{h}^n | \mathbf{v}^n; \theta^{t-1})}_{=: q^n(\mathbf{h}^n)}$$

E-step, Part 1

$$q \leftarrow \underset{q}{\operatorname{argmax}} \; G(\theta^{t-1}, q)$$

- as for GMMs:

$$q^t \leftarrow p(\mathbf{h}^1, \ldots, \mathbf{h}^N | \mathbf{v}^1, \ldots, \mathbf{v}^N; \theta^{t-1}) \overset{i.i.d.}{=} \prod_{n=1}^{N} \underbrace{p(\mathbf{h}^n | \mathbf{v}^n; \theta^{t-1})}_{=: q^n(\mathbf{h}^n)} = \prod_{n=1}^{N} q^n(\mathbf{h}^n)$$

later more...

M-step

M-step

$$\mathop{\mathbb{E}}_{(\mathbf{h}^1,\dots,\mathbf{h}^N)\sim q} \log p(\mathbf{v}^1,\dots,\mathbf{v}^N,\mathbf{h}^1,\dots,\mathbf{h}^N;\theta)$$

$$\overset{i.i.d.}{=} \mathop{\mathbb{E}}_{(\mathbf{h}^1,\dots,\mathbf{h}^n\sim q} \sum_{n=1}^{N} \log p(\mathbf{v}^n,\mathbf{h}^n;\theta) = \sum_{n=1}^{N} \mathop{\mathbb{E}}_{\mathbf{h}\sim q^n} \log p(v^n_{1:T^n}, h_{1:T^n};\theta)$$

$$\overset{\text{HMM graph}}{=} \sum_{n=1}^{N} \mathop{\mathbb{E}}_{\mathbf{h}\sim q^n} \log \left[p(h_1;a) \prod_{t=2}^{T^n} p(h_t|h_{t-1};A) \prod_{t=1}^{T^n} p(v^n_t|h_t;B) \right]$$

$$= \underbrace{\sum_{n=1}^{N} \mathop{\mathbb{E}}_{\mathbf{h}\sim q^n} \log p(h_1;a)}_{\mathcal{L}_{\text{initial}}(a)} + \underbrace{\sum_{n=1}^{N}\sum_{t=2}^{T^n} \mathop{\mathbb{E}}_{\mathbf{h}\sim q^n} \log p(h_t|h_{t-1};A)}_{\mathcal{L}_{\text{transition}}(A)} + \underbrace{\sum_{n=1}^{N}\sum_{t=1}^{T^n} \mathop{\mathbb{E}}_{\mathbf{h}\sim q^n} \log p(v^n_t|h_t;B)}_{\mathcal{L}_{\text{emission}}(B)}$$

sum of independent terms $\rightarrow$ we can optimize for a, A and B separately

$$\mathcal{L}_{\text{initial}}(a) = \sum_{n=1}^{N} \mathop{\mathbb{E}}_{h_{1:T_n} \sim q^n} \log p(h_1; a) = \sum_{n=1}^{N} \mathop{\mathbb{E}}_{h_1 \sim q^n} \log a_{h_1}$$

a is a discrete probability distribution over H states, *i.e.* $\sum_i a_i = 1$. Use Langragian:

$$\mathfrak{L}(a, \lambda) = \mathcal{L}_{\text{initial}}(a) - \lambda\Big(\sum_i a_i - 1\Big)$$

$$\frac{d\mathcal{L}_{\text{initial}}(a)}{da_i}(a) = \frac{d}{da_i} \sum_{n=1}^{N} \mathop{\mathbb{E}}_{h_1 \sim q^n} \sum_{i'=1}^{H} [\![h_1 = i']\!] \log a_{i'} = \sum_{n=1}^{N} \mathop{\mathbb{E}}_{h_1 \sim q^n} [\![h_1 = i]\!] \frac{1}{a_i} = \frac{1}{a_i} \sum_{n=1}^{N} q^n(h_1)$$

$$0 = \frac{d\mathfrak{L}(a, \lambda)}{da_i}(\hat{a}, \hat{\lambda}) = \frac{1}{\hat{a}_i} \sum_{n=1}^{N} q^n(h_1 = i) - \lambda \quad \rightarrow \quad \hat{a}_i = \frac{1}{\hat{\lambda}} \sum_{n=1}^{N} q^n(h_1)$$

$$0 = \frac{d\mathfrak{L}(a, \lambda)}{d\lambda}(\hat{a}, \hat{\lambda}) = -1 + \sum_{i=1}^{H} \frac{1}{\hat{\lambda}} \sum_{n=1}^{N} q^n(h_1 = i) = -1 + \sum_{i=1}^{H} \frac{1}{\hat{\lambda}} \quad \rightarrow \quad \hat{\lambda} = n$$

$$\mathcal{L}_{\text{transition}}(A) = \sum_{n=1}^{N}\sum_{t=2}^{T^n} \mathop{\mathbb{E}}_{\mathbf{h}\sim q^n} \log p(h_t | h_{t-1}; A)$$
$$= \sum_{n=1}^{N}\sum_{t=2}^{T^n} \mathop{\mathbb{E}}_{h_{1:T^n}\sim q^n} \sum_{i,i'=1}^{H} [\![h_t = i \wedge h_{t-1} = i']\!] \log A_{i,i'}$$
$$= \sum_{n=1}^{N}\sum_{t=2}^{T^n} \sum_{i,i'=1}^{H} q^n(h_t = i, h_{t-1} = i') \log A_{i,i'}$$

Each column of A is a (conditional) distribution over the rows, *i.e.* $\sum_i A_{i,i'} = 1$ for any $i' \in \{1, \ldots, H\}$. We can optimize for any fixed i' independently:

$$\mathfrak{L}(A, \lambda) = \mathcal{L}_{\text{transition}}(A) - \lambda\Big(\sum_i A_{i,i'} - 1\Big)$$

$$\hat{A}_{i,i'} \propto \sum_{n=1}^{n}\sum_{t=2}^{T_n} q^n(h_t = i, h_{t-1} = i') \quad \text{with normalization to make } \hat{A}_{i,i'} = 1 \text{ for each } i'$$

$$\mathcal{L}_{\text{emission}}(A) = \sum_{n=1}^{N}\sum_{t=1}^{T^n} \mathop{\mathbb{E}}_{\mathbf{h}\sim q^n} \log p(v_t^n | h_t; B) = \sum_{n=1}^{N}\sum_{t=1}^{T^n}\sum_{j=1}^{V} [\![v_t^n = j]\!] \mathop{\mathbb{E}}_{h_{1:T^n}\sim q^n} \sum_{i=1}^{H} [\![h_t = i]\!] \log B_{j,i}$$

$$= \sum_{n=1}^{N}\sum_{t=1}^{T^n}\sum_{j=1}^{V} [\![v_t^n = j]\!] \sum_{i=1}^{H} q^n(h_t = i) \log B_{j,i}$$

Each column of B is a (conditional) distribution over the rows, *i.e.* $\sum_j B_{j,i} = 1$ for any $j \in \{1, \ldots, V\}$. We can optimize for any fixed i independently:

$$\mathfrak{L}(B, \lambda) = \mathcal{L}_{\text{emission}}(B) - \lambda\left(\sum_j B_{j,i} - 1\right)$$

$$\hat{B}_{j,i} \propto \sum_{n=1}^{n}\sum_{t=1}^{T_n} [\![v_t^n = j]\!] q^n(h_t = i) \quad \text{with normalization to make } \hat{B}_{j,i} = 1 \text{ for each } i$$

E-step, Part 2

For the M-step we compute:

$$\hat{a}_i \propto \sum_{n=1}^{N} q^n(h_1) \qquad \hat{A}_{i,i'} \propto \sum_{n=1}^{n} \sum_{t=2}^{T_n} q^n(h_t = i, h_{t-1} = i') \qquad \hat{B}_{j,i} \propto \sum_{n=1}^{n} \sum_{t=1}^{T_n} [\![v_t^n = j]\!] q^n(h_t = i)$$

Of $q^n(\mathbf{h}) = p(\mathbf{h}|\mathbf{v}^n; \theta)$ we really only need:

- $q^n(h_1) = p(h_1|v^n_{1:T^n}; \theta)$ for a
- $q^n(h_t, h_{t-1}) = p(h_t, h_{t-1}|v^n_{1:T^n}; \theta)$ for A
- $q^n(h_t) = p(h_t|v^n_{1:T^n}; \theta)$ for B

For computing all of these we have derived efficient ways in the previous section.

EM for HMMs: Initialization

EM algorithm:

initialize θ^0
for $t = 1, 2, \ldots,$ until convergence **do**
$q^t \leftarrow \text{argmax}_q \; G(\theta^{t-1}, q)$ // E-step
$\theta^t \leftarrow \text{argmax}_\theta \; G(\theta, q^t)$ // M-step
end for

Parameter initialisation

- EM algorithm converges to a local maximum of the likelihood,
- in general, there is no guarantee that the algorithm will find the global maximum
- often, the initialization determined how good the found solution is
- practical strategy:
  - first, train non-temporal mixture model for $p(v) = \sum_h p(v|h)p(h)$
  - initialize a and B from this, and assume independence for A

HMM with Continuous observations

For an HMM with continuous observation $\mathbf{v}_t$, we need a model of $p(\mathbf{v}_t|h_t)$, i.e. a continuous distribution for each state of h_t.

Inference

- filtering, smoothing, etc. remain largely unchanged, as everything is conditioned on $\mathbf{v}_{1:T}$

Learning

- learning requires computing normalization constants w.r.t. v
- depending on the model, this might or might not be tractable