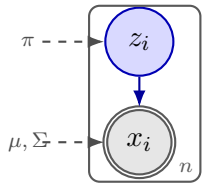
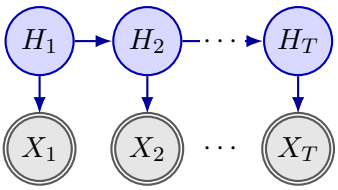
Optional Lecture Notes

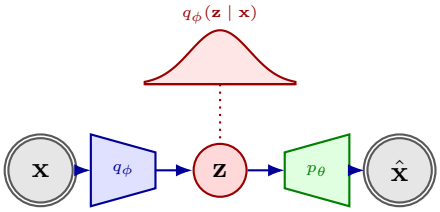
Latent Variable Models



GMM



HMM



VAE

BUSI70575

Imperial College Business School

May 13, 2026

Abstract

These optional lecture notes present a unified treatment of latent variable models. We begin with the Expectation Maximization (EM) algorithm and apply it to Gaussian Mixture Models with full M-step derivations. We then develop Hidden Markov Models with both continuous (Gaussian) and discrete (product-multinomial) emissions, deriving the Forward-Backward algorithm and the complete EM updates. We close with prediction of the next hidden state and the next observation, and an exercise extending the framework to Variational Autoencoders.

Contents

1 The Expectation Maximization Algorithm **4**

1.1 Setup . . . 4

1.2 The Evidence Lower Bound (ELBO) . . . 4

1.3 Closing the Bound: the Posterior . . . 5

1.4 The EM Algorithm . . . 5

2 Application: Gaussian Mixture Models **7**

2.1 Model . . . 7

2.2 Deriving EM for GMM . . . 7

3 Hidden Markov Models **11**

3.1 Motivation: from Mixture to Markov Chain . . . 11

3.2 Model Specification . . . 11

3.3 Notation: Slice Notation for Sequences . . . 12

3.4 Joint Distribution . . . 12

3.5 Filtering and Smoothing Probabilities . . . 12

3.6 Forward Pass: the α Variables . . . 12

3.7 Backward Pass: the β Variables . . . 14

3.8 Smoothing Probabilities in Closed Form . . . 15

3.8.1 Single-state smoothing $\psi(t, h)$ . . . 15

3.8.2 Pairwise smoothing $\phi(t, h, h')$ . . . 16

3.9 Recap of the Forward-Backward Algorithm . . . 17

3.10 Learning the HMM Parameters via EM . . . 18

3.10.1 Complete-Data Log-Likelihood . . . 18

3.10.2 E-Step . . . 18
3.10.3 M-Step: Update of π . . . 19
3.10.4 M-Step: Update of Q . . . 19
3.10.5 M-Step: Gaussian Emissions . . . 20
3.10.6 M-Step: Product-Multinomial Emissions . . . 21
3.11 Recap of the EM Loop for HMM . . . 22

4 Prediction: the Next Hidden State and Observation **24**
4.1 Predicting the Next Hidden State . . . 24
4.2 Predicting the Next Observation (Discrete Case) . . . 25

5 Variational Autoencoder: a Continuous Latent-Variable Model **26**

1 The Expectation Maximization Algorithm

1.1 Setup

We consider statistical models with two kinds of variables: **observed** variables x and **latent** (or hidden) variables z. The joint distribution $p_\theta(x, z)$ is parameterized by θ. Latent variables are not directly observed but rather inferred through the observed data x.

Marginal log-likelihood objective

Given i.i.d. observations $x_1, \ldots, x_n$ with associated latent variables $z_1, \ldots, z_n$, the goal of EM is to maximize the marginal log-likelihood:

$$\log p_\theta(x) = \sum_{i=1}^{n} \log p_\theta(x_i) = \sum_{i=1}^{n} \log \sum_{z_i} p_\theta(x_i, z_i).$$

The log of a sum makes this optimization non-trivial. The EM algorithm sidesteps the difficulty by constructing a tractable lower bound and alternately maximizing it in two variables.

1.2 The Evidence Lower Bound (ELBO)

Jensen's inequality

If $f : \mathbb{R} \to \mathbb{R}$ is convex and X is an integrable random variable,

$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X]),$$

with equality iff X is constant a.s. when f is strictly convex.

For each i, introduce a probability mass function $z_i \mapsto q(z_i)$ with $q(z_i) \geq 0$ and $\sum_{z_i} q(z_i) = 1$. Applying Jensen's inequality to $-\log$ (which is convex):

$$\begin{aligned}
\log p_\theta(x) &= \sum_{i=1}^{n} \log \sum_{z_i} p_\theta(x_i, z_i) \\
&= \sum_{i=1}^{n} \log \sum_{z_i} q(z_i) \frac{p_\theta(x_i, z_i)}{q(z_i)} \\
&\geq \sum_{i=1}^{n} \sum_{z_i} q(z_i) \log \frac{p_\theta(x_i, z_i)}{q(z_i)} \\
&= \sum_{i=1}^{n} \underbrace{\mathbb{E}_{q(z_i)}\left[\log \frac{p_\theta(x_i, z_i)}{q(z_i)}\right]}_{=: \mathcal{L}(q(z_i), \theta)}.
\end{aligned}$$

The quantity $\mathcal{L}(q, \theta) = \sum_i \mathcal{L}(q(z_i), \theta)$ is the **Evidence Lower Bound (ELBO)**. The EM algorithm alternately maximizes it in q (E-step) and θ (M-step).

1.3 Closing the Bound: the Posterior

Exercise

Show that the gap between the marginal log-likelihood and the ELBO is zero when $q(z_i) = p_\theta(z_i \mid x_i)$ for all i.

Solution

Let $d = \log p_\theta(x) - \sum_{i=1}^n \mathcal{L}(q(z_i), \theta)$ denote the gap.

Step 1. Expand the gap term by term.

$$d = \sum_{i=1}^n \left(\log p_\theta(x_i) - \sum_{z_i} q(z_i) \log \frac{p_\theta(x_i, z_i)}{q(z_i)} \right).$$

Step 2. Insert $\sum_{z_i} q(z_i) = 1$ in front of $\log p_\theta(x_i)$ and combine logarithms.

$$d = \sum_{i=1}^n \sum_{z_i} q(z_i) \left(\log p_\theta(x_i) - \log \frac{p_\theta(x_i, z_i)}{q(z_i)} \right) = \sum_{i=1}^n \sum_{z_i} q(z_i) \log \frac{q(z_i)\, p_\theta(x_i)}{p_\theta(x_i, z_i)}.$$

Step 3. Use Bayes' rule $p_\theta(x_i, z_i) = p_\theta(z_i \mid x_i)\, p_\theta(x_i)$ to recognize a KL divergence.

$$d = \sum_{i=1}^n \sum_{z_i} q(z_i) \log \frac{q(z_i)}{p_\theta(z_i \mid x_i)} = \sum_{i=1}^n \mathrm{KL}(q(z_i) \,\|\, p_\theta(z_i \mid x_i)) \;\geq\; 0.$$

Step 4 (conclusion). The KL divergence is zero iff its two arguments coincide, so

$$d = 0 \iff q(z_i) = p_\theta(z_i \mid x_i) \quad \forall i.$$

This is called the **posterior distribution**.

1.4 The EM Algorithm

At the M-step, since $\sum_{z_i} q(z_i) \log q(z_i)$ does not depend on θ, maximizing $\mathcal{L}(q, \theta)$ with respect to θ is equivalent to maximizing the **expected complete-data log-likelihood**:

$$\mathcal{Q}(\theta, \theta_t) := \mathbb{E}_{Z|X,\theta_t}[\log p_\theta(X, Z)]\,.$$

Algorithm 1 EM algorithm

Require: Observations $x_1, \ldots, x_n$
Ensure: Locally optimal parameters $\theta^\star$
1: Initialize $\theta^{(0)}$; set $t = 0$
2: **while** not converged **do**
3: **E-step:** compute the posterior $q^{(t+1)}(z_i) = p_{\theta^{(t)}}(z_i \mid x_i)$ for each i
4: **M-step:** update $\theta^{(t+1)} = \arg\max_\theta \mathbb{E}_{Z|X,\theta^{(t)}}[\log p_\theta(X, Z)]$
5: $t \leftarrow t + 1$
6: **end while**
7: **return** $\theta^{(t)}$

Properties.

- The log-likelihood is non-decreasing across iterations (ascent algorithm).
- The sequence of log-likelihoods converges, but only to a *local* maximum since the problem is generally non-convex.
- It is wise to run EM from several initializations and keep the run with highest likelihood.
- For Gaussian mixtures, a common initialization is K-means.

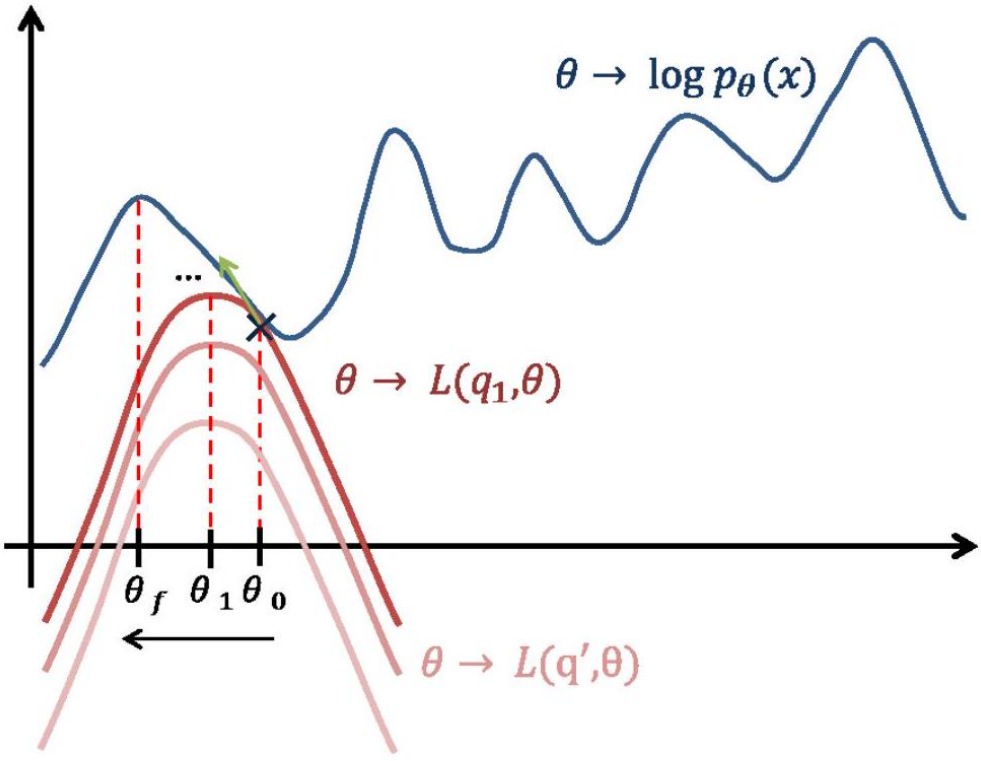


Figure 1: An illustration of the EM algorithm that converges to a local minimum.

2 Application: Gaussian Mixture Models

2.1 Model

A Gaussian Mixture Model (GMM) represents a population as a convex combination of K Gaussian subpopulations. Each observation is assumed to come from one of K components, but the component index is unobserved.

GMM with K components

For $i \in [\![1, n]\!]$, the latent variable z_i takes values in $\{1, \dots, K\}$ and is drawn from a multinomial:

$$z_i \sim \mathcal{M}(1; \pi_1, \dots, \pi_K), \qquad \sum_{j=1}^{K} \pi_j = 1,$$
$$x_i \mid z_i = j \sim \mathcal{N}(\mu_j, \Sigma_j), \qquad x_i \in \mathbb{R}^p.$$

The parameters are $\theta = (\pi, \mu, \Sigma)$ with $\pi \in \mathbb{R}^K$, $\mu \in \mathbb{R}^{K \times p}$, $\Sigma \in \mathbb{R}^{K \times p \times p}$.

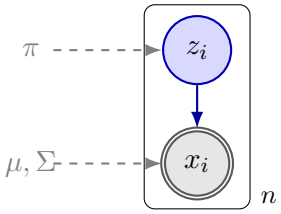


Figure 2: Graphical model of a GMM. The hidden cluster index z_i generates the observation x_i, repeated n times.

2.2 Deriving EM for GMM

Step 1: posterior of z_i given x_i. Using Bayes' rule with $\theta_t = (\pi_t, \mu_t, \Sigma_t)$ at iteration t:

$$\tau_i^j(\theta_t) := p_{\theta_t}(z_i = j \mid x_i) = \frac{\pi_{j,t}\, \mathcal{N}(x_i \mid \mu_{j,t}, \Sigma_{j,t})}{\sum_{j'=1}^{K} \pi_{j',t}\, \mathcal{N}(x_i \mid \mu_{j',t}, \Sigma_{j',t})}.$$

The quantity τ_i^j is called the *responsibility* of component j for observation i.

Step 2: complete-data log-likelihood with indicator variables. The key technical step is to introduce **one-hot indicators**

$$z_i^j := \mathbf{1}\{z_i = j\} \in \{0, 1\}, \qquad \sum_{j=1}^{K} z_i^j = 1,$$

which turn the categorical variable z_i into a tractable sum. We rewrite each term of the complete-data log-likelihood:

$$\log p_\theta(z_i = j) = \log \pi_j = \sum_{j'=1}^{K} z_i^{j'} \log \pi_{j'},$$

and similarly

$$\log p_\theta(x_i \mid z_i = j) = \log \mathcal{N}(x_i \mid \mu_j, \Sigma_j) = \sum_{j'=1}^{K} z_i^{j'} \log \mathcal{N}(x_i \mid \mu_{j'}, \Sigma_{j'}).$$

Summing over all i gives the full complete-data log-likelihood:

$$\begin{aligned}\log p_\theta(x, z) &= \sum_{i=1}^{n} \log p_\theta(z_i) + \sum_{i=1}^{n} \log p_\theta(x_i \mid z_i) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{K} z_i^j \log \pi_j + \sum_{i=1}^{n}\sum_{j=1}^{K} z_i^j \log \mathcal{N}(x_i \mid \mu_j, \Sigma_j).\end{aligned}$$

Step 3: E-step. The indicator variables z_i^j are the only random objects under $p_{\theta_t}(z \mid x)$, with

$$\mathbb{E}_{Z|X,\theta_t}[z_i^j] = p_{\theta_t}(z_i = j \mid x_i) = \tau_i^j(\theta_t).$$

Therefore the expected complete-data log-likelihood is

GMM expected complete-data log-likelihood

$$\mathcal{Q}(\theta, \theta_t) = \sum_{i=1}^{n}\sum_{j=1}^{K} \tau_i^j \log \pi_j + \sum_{i=1}^{n}\sum_{j=1}^{K} \tau_i^j \log \mathcal{N}(x_i \mid \mu_j, \Sigma_j).$$

Step 4: M-step (full derivations). We maximize $\mathcal{Q}(\theta, \theta_t)$ with respect to $\theta = (\pi, \mu, \Sigma)$. The three blocks decouple.

(a) Update of π We solve

$$\max_{\pi} \sum_{j=1}^{K} \left(\sum_{i=1}^{n} \tau_i^j \right) \log \pi_j \qquad \text{s.t.} \quad \sum_{j=1}^{K} \pi_j = 1.$$

Lagrangian:

$$\mathcal{L}(\pi, \lambda) = \sum_{j=1}^{K} \left(\sum_{i=1}^{n} \tau_i^j \right) \log \pi_j - \lambda \left(\sum_{j=1}^{K} \pi_j - 1 \right).$$

First-order condition:

$$\frac{\partial \mathcal{L}}{\partial \pi_j} = \frac{\sum_{i=1}^{n} \tau_i^j}{\pi_j} - \lambda = 0 \implies \pi_j = \frac{1}{\lambda} \sum_{i=1}^{n} \tau_i^j.$$

Summing over j and using $\sum_j \tau_i^j = 1$ for all i gives $\lambda = n$, hence

$$\boxed{\pi_{j,t+1} = \frac{1}{n} \sum_{i=1}^{n} \tau_i^j.}$$

(b) Update of μ_j We expand the Gaussian log-density:

$$\log \mathcal{N}(x_i \mid \mu_j, \Sigma_j) = -\frac{p}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma_j| - \frac{1}{2} (x_i - \mu_j)^\top \Sigma_j^{-1} (x_i - \mu_j).$$

Only the last term depends on μ_j. Differentiating w.r.t. μ_j:

$$\nabla_{\mu_j} (x_i - \mu_j)^\top \Sigma_j^{-1} (x_i - \mu_j) = -2\Sigma_j^{-1} (x_i - \mu_j).$$

Therefore

$$\nabla_{\mu_j} \mathcal{Q} = \sum_{i=1}^{n} \tau_i^j \, \Sigma_j^{-1} (x_i - \mu_j) = 0 \implies \sum_{i=1}^{n} \tau_i^j (x_i - \mu_j) = 0,$$

giving

$$\boxed{\mu_{j,t+1} = \frac{\sum_{i=1}^{n} \tau_i^j \, x_i}{\sum_{i=1}^{n} \tau_i^j}.}$$

(c) Update of Σ_j Let $\Omega_j = \Sigma_j^{-1}$. Then $\log |\Sigma_j| = -\log |\Omega_j|$, and the relevant part of $\mathcal{Q}$ is

$$\frac{1}{2} \sum_{i=1}^{n} \tau_i^j \left[\log |\Omega_j| - (x_i - \mu_j)^\top \Omega_j (x_i - \mu_j) \right].$$

Using the matrix identities $\nabla_\Omega \log |\Omega| = \Omega^{-1}$ and $\nabla_\Omega (v^\top \Omega v) = v v^\top$, we get

$$\nabla_{\Omega_j} \mathcal{Q} = \frac{1}{2} \sum_{i=1}^{n} \tau_i^j \left[\Omega_j^{-1} - (x_i - \mu_j)(x_i - \mu_j)^\top \right] = 0,$$

which yields

$$\boxed{\Sigma_{j,t+1} = \frac{\sum_{i=1}^{n} \tau_i^j \, (x_i - \mu_{j,t+1})(x_i - \mu_{j,t+1})^\top}{\sum_{i=1}^{n} \tau_i^j}.}$$

GMM EM updates (summary)

$$\tau_i^j \leftarrow \frac{\pi_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)}{\sum_{j'} \pi_{j'} \mathcal{N}(x_i \mid \mu_{j'}, \Sigma_{j'})} \qquad \text{(E-step)}$$

$$\pi_j \leftarrow \frac{1}{n} \sum_{i=1}^{n} \tau_i^j \qquad \text{(M-step)}$$

$$\mu_j \leftarrow \frac{\sum_i \tau_i^j x_i}{\sum_i \tau_i^j} \qquad \text{(M-step)}$$

$$\Sigma_j \leftarrow \frac{\sum_i \tau_i^j (x_i - \mu_j)(x_i - \mu_j)^\top}{\sum_i \tau_i^j} \qquad \text{(M-step)}$$

Variants of Σ_j.

- **Isotropic:** $\Sigma_j = \sigma_j^2 I_p$, one parameter per component (spherical cluster).
- **Diagonal:** Σ_j diagonal, p parameters per component (axis-aligned ellipse).
- **Full:** Σ_j general PSD, $p(p+1)/2$ parameters per component (general ellipse).

3 Hidden Markov Models

3.1 Motivation: from Mixture to Markov Chain

In a GMM the latent variables are i.i.d. In many sequential settings (speech, finance, biology), the latent state evolves over time: the current state depends on the previous one. A Hidden Markov Model (HMM) generalizes the mixture model by replacing the i.i.d. prior over z_i with a first-order Markov chain.

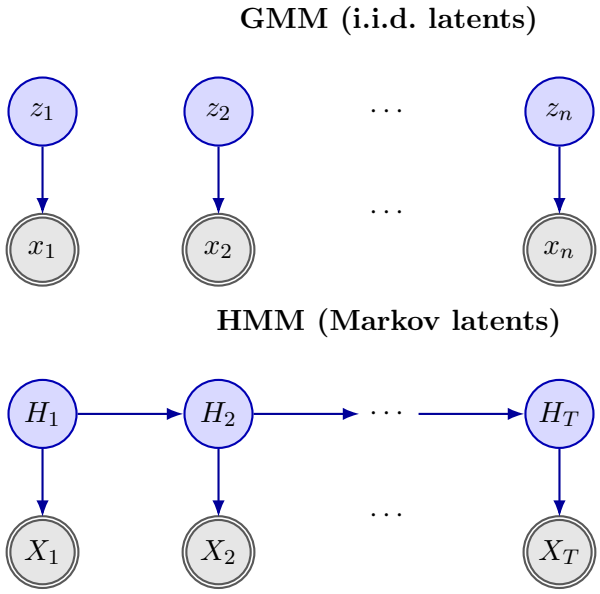


Figure 3: Comparison of the graphical models: GMM (top) versus HMM (bottom). The HMM adds horizontal arrows between latent states.

3.2 Model Specification

Hidden Markov Model

Let $H_t \in \{1, \dots, M\}$ denote the hidden state at time t and X_t the observation. The HMM is specified by:

- **Initial distribution** $\pi \in \mathbb{R}^M$: $\pi_m = p(H_1 = m)$,
- **Transition matrix** $Q \in \mathbb{R}^{M \times M}$ (assumed homogeneous): $Q_{m,m'} = p(H_{t+1} = m' \mid H_t = m)$,
- **Emission distribution** $p(X_t \mid H_t = m)$, also assumed homogeneous in t.

The full parameter vector is $\theta = (\pi, Q, \phi_{\text{emission}})$.

We will treat two emission families in parallel:

- **Gaussian (continuous):** $X_t \mid H_t = m \sim \mathcal{N}(\mu_m, \Sigma_m)$ with $X_t \in \mathbb{R}^d$.

- **Product-multinomial (discrete):** $X_t = (X_t^1, \ldots, X_t^d)$ with each $X_t^c \in \{1, \ldots, K\}$ and

$$p(X_t \mid H_t = m) = \prod_{c=1}^{d} O_{m,c,X_t^c},$$

with $\sum_k O_{m,c,k} = 1$ for every (m, c).

3.3 Notation: Slice Notation for Sequences

To keep equations compact, we use the slice notation $X_{1:t} := (X_1, X_2, \ldots, X_t)$. We also write $X_{1:T}$ for the full sequence.

3.4 Joint Distribution

By the graph structure,

$$p_\theta(H_{1:T}, X_{1:T}) = p(H_1) \prod_{t=1}^{T-1} p(H_{t+1} \mid H_t) \prod_{t=1}^{T} p(X_t \mid H_t)$$

$$= \pi_{H_1} \prod_{t=1}^{T-1} Q_{H_t, H_{t+1}} \prod_{t=1}^{T} p(X_t \mid H_t).$$

The marginal $p_\theta(X_{1:T}) = \sum_{H_1,\ldots,H_T} p_\theta(H_{1:T}, X_{1:T})$ has M^T terms; naive computation is intractable. The **Forward-Backward algorithm** computes it (and the smoothing posteriors) in $O(TM^2)$ time.

3.5 Filtering and Smoothing Probabilities

Posterior quantities of interest

$$\begin{aligned}
\text{Filtering:} \quad & \xi(t, h) = p(H_t = h \mid X_{1:t}), \\
\text{Smoothing (one state):} \quad & \psi(t, h) = p(H_t = h \mid X_{1:T}), \\
\text{Smoothing (pair of states):} \quad & \phi(t, h, h') = p(H_t = h, H_{t+1} = h' \mid X_{1:T}).
\end{aligned}$$

We assemble these into vectors and matrices: $\xi_t, \psi_t \in \mathbb{R}^M$ and $\phi_t \in \mathbb{R}^{M \times M}$.

3.6 Forward Pass: the α Variables

Define

$$\alpha(t, h) := p(X_{1:t}, H_t = h), \qquad \alpha_t := (\alpha(t, 1), \ldots, \alpha(t, M))^\top.$$

The recursion is illustrated below: each $\alpha(t, h)$ is obtained by summing over all possible values of the previous hidden state H_{t-1}.

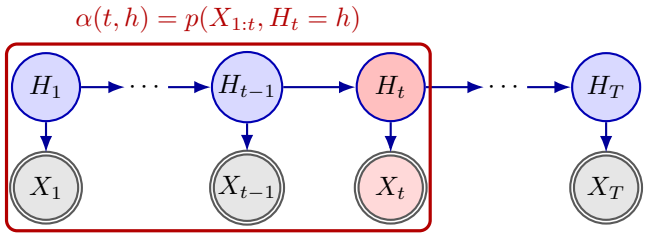


Figure 4: The forward variable $\alpha(t,h)$ encompasses everything in the past up to time t: the observations $X_{1:t}$ and the current state $H_t = h$.

Recursion.

Proof

Step 1. Marginalize over the previous hidden state H_{t-1}.

$$\alpha(t,h) = p(X_{1:t}, H_t = h) = \sum_{h'=1}^{M} p(X_{1:t}, H_t = h, H_{t-1} = h').$$

Step 2. Factor the joint by conditioning on H_t and H_{t-1} and apply graph independencies (given H_t, the observation X_t is independent of the past; given H_{t-1}, the state H_t is independent of $X_{1:t-1}$):

$$p(X_{1:t}, H_t = h, H_{t-1} = h') = \underbrace{p(X_t \mid H_t = h)}_{\text{emission}} \cdot \underbrace{p(H_t = h \mid H_{t-1} = h')}_{Q_{h',h}} \cdot \underbrace{p(X_{1:t-1}, H_{t-1} = h')}_{\alpha(t-1,h')}.$$

Step 3. Collect terms.

$$\alpha(t,h) = p(X_t \mid H_t = h) \sum_{h'=1}^{M} Q_{h',h}\, \alpha(t-1,h').$$

Introducing the diagonal **emission matrix**

$$\Gamma(t) := \operatorname{diag}\big(p(X_t \mid H_t = 1), \ldots, p(X_t \mid H_t = M)\big) \in \mathbb{R}^{M \times M},$$

the recursion is

Forward recursion

$$\alpha_1 = \Gamma(1)\pi, \qquad \alpha_t = \Gamma(t)\, Q^\top \alpha_{t-1} \quad \text{for } t \geq 2.$$

Filtering probabilities. By Bayes' rule,

$$\xi(t,h) = \frac{\alpha(t,h)}{\sum_{h'} \alpha(t,h')} \iff \xi_t = \frac{\alpha_t}{\mathbf{1}_M^\top \alpha_t}.$$

3.7 Backward Pass: the β Variables

Define

$$\beta(t,h) := p(X_{t+1:T} \mid H_t = h), \qquad \beta_t := (\beta(t,1), \dots, \beta(t,M))^\top.$$

By convention, $p(X_{T+1:T} \mid \cdot) = 1$, so $\beta_T = \mathbf{1}_M$.

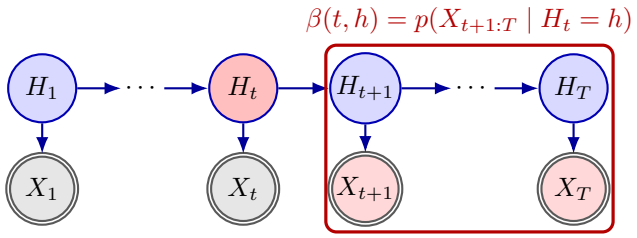


Figure 5: The backward variable $\beta(t,h)$ encompasses everything in the future: the observations $X_{t+1:T}$, given that we are in state $H_t = h$.

Recursion.

Proof

Step 1. Marginalize over the next hidden state H_{t+1}.

$$\beta(t,h) = \sum_{h'=1}^{M} p(X_{t+1:T}, H_{t+1} = h' \mid H_t = h).$$

Step 2. Factor using the chain rule and graph independencies:

$$p(X_{t+1:T}, H_{t+1} = h' \mid H_t = h) = \underbrace{p(H_{t+1} = h' \mid H_t = h)}_{Q_{h,h'}} \cdot \underbrace{p(X_{t+1:T} \mid H_{t+1} = h')}_{\text{decomposed below}}.$$

Step 3. Split $X_{t+1:T} = (X_{t+1}, X_{t+2:T})$ and use that, given H_{t+1}, the observation X_{t+1} is independent of the future:

$$p(X_{t+1:T} \mid H_{t+1} = h') = \underbrace{p(X_{t+1} \mid H_{t+1} = h')}_{\Gamma(t+1)_{h',h'}} \cdot \underbrace{p(X_{t+2:T} \mid H_{t+1} = h')}_{\beta(t+1,h')}.$$

Step 4. Combine.

$$\beta(t,h) = \sum_{h'=1}^{M} Q_{h,h'}\, \Gamma(t+1)_{h',h'}\, \beta(t+1,h').$$

Backward recursion

$$\beta_T = \mathbf{1}_M, \qquad \beta_t = Q\,\Gamma(t+1)\,\beta_{t+1} \quad \text{for } t = T-1, \dots, 1.$$

3.8 Smoothing Probabilities in Closed Form

3.8.1 Single-state smoothing $\psi(t, h)$

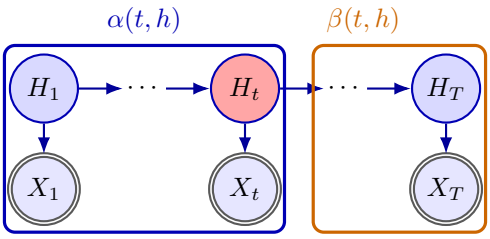


Figure 6: Single-state smoothing splits the sequence at time t: the past contributes $\alpha(t, h)$ and the future contributes $\beta(t, h)$.

Proof

Step 1. Apply Bayes' rule.

$$\psi(t,h) = p(H_t = h \mid X_{1:T}) = \frac{p(H_t = h, X_{1:T})}{p(X_{1:T})}.$$

Step 2. Split the joint $p(H_t = h, X_{1:T}) = p(H_t = h, X_{1:t})\, p(X_{t+1:T} \mid H_t = h, X_{1:t})$, then use the independence $X_{t+1:T} \perp X_{1:t} \mid H_t$:

$$p(H_t = h, X_{1:T}) = \underbrace{p(H_t = h, X_{1:t})}_{\alpha(t,h)} \cdot \underbrace{p(X_{t+1:T} \mid H_t = h)}_{\beta(t,h)}.$$

Step 3. The denominator is $p(X_{1:T}) = \sum_{h''} \alpha(T, h'') = \mathbf{1}_M^\top \alpha_T$. Therefore

$$\psi(t,h) = \frac{\alpha(t,h)\,\beta(t,h)}{\mathbf{1}_M^\top \alpha_T}.$$

Compactly,

$$\boxed{\psi_t = \frac{\alpha_t \circ \beta_t}{\mathbf{1}_M^\top \alpha_T},}$$

where $\circ$ denotes the Hadamard (entrywise) product.

3.8.2 Pairwise smoothing $\phi(t, h, h')$

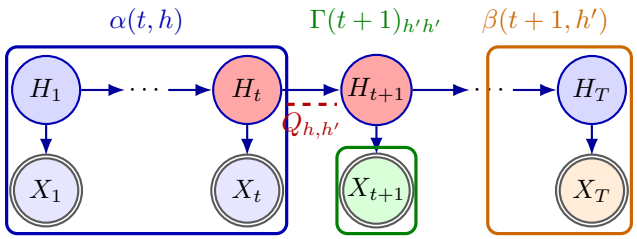


Figure 7: Pairwise smoothing decomposes into four factors: the past trajectory (α), the transition (Q), the emission at $t+1$ (Γ), and the future trajectory (β).

Proof

Step 1. Apply Bayes' rule.

$$\phi(t, h, h') = p(H_t = h, H_{t+1} = h' \mid X_{1:T}) = \frac{p(X_{1:T}, H_t = h, H_{t+1} = h')}{p(X_{1:T})}.$$

Step 2. Split the observation sequence around time t:

$$X_{1:T} = (X_{1:t}, X_{t+1}, X_{t+2:T}).$$

Conditioning on $(H_t = h, H_{t+1} = h')$ and using d-separation:

- $X_{1:t} \perp \{X_{t+1}, X_{t+2:T}\} \mid H_t$, so $p(X_{1:t} \mid H_t = h, H_{t+1} = h') = p(X_{1:t} \mid H_t = h)$.
- $X_{t+1}, X_{t+2:T}$ are independent given H_{t+1}.

Step 3. Identify each factor as a known quantity.

$$p(X_{1:T}, H_t = h, H_{t+1} = h') = \underbrace{p(X_{1:t}, H_t = h)}_{\alpha(t,h)} \cdot \underbrace{p(H_{t+1} = h' \mid H_t = h)}_{Q_{h,h'}} \cdot \underbrace{p(X_{t+1} \mid H_{t+1} = h')}_{\Gamma(t+1)_{h',h'}} \cdot \underbrace{p(X_{t+2:T} \mid H_{t+1} = h')}_{\beta(t+1,h')}.$$

Step 4. Combine and normalize.

$$\phi(t, h, h') = \frac{\alpha(t, h)\, Q_{h,h'}\, \Gamma(t+1)_{h',h'}\, \beta(t+1, h')}{\mathbf{1}_M^\top \alpha_T}.$$

In matrix form,

$$\boxed{\phi_t = \frac{\operatorname{diag}(\alpha_t)\, Q\, \Gamma(t+1)\, \operatorname{diag}(\beta_{t+1})}{\mathbf{1}_M^\top \alpha_T}.}$$

3.9 Recap of the Forward-Backward Algorithm

We have computed two recursive sequences, α moving forward in time and β moving backward, and combined them into the filtering and smoothing probabilities. The diagram below summarizes the full pipeline.

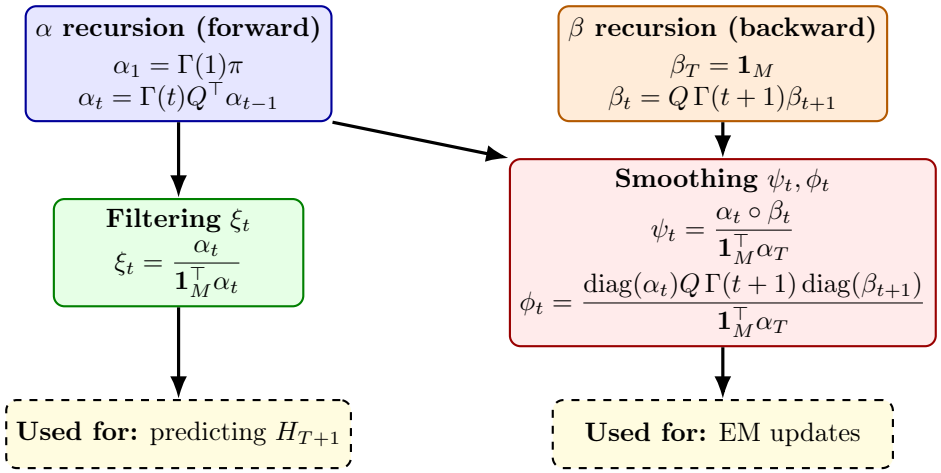


Figure 8: Recap of the Forward-Backward algorithm. The forward variables α and backward variables β are computed recursively, then combined: α alone yields the filtering distribution ξ, while α and β together yield the smoothing distributions ψ and ϕ. **Filtering** feeds the prediction step (next hidden state); **smoothing** feeds the EM updates.

Algorithm 2 Forward-Backward algorithm

Require: Observations $X_{1:T}$; parameters $\theta = (\pi, Q, \Gamma(1), \ldots, \Gamma(T))$
Ensure: Smoothing probabilities (ψ_t), (ϕ_t); marginal likelihood $p(X_{1:T})$

Forward pass:
$\alpha_1 \leftarrow \Gamma(1)\,\pi$
for $t = 2, \ldots, T$ **do**
  $\alpha_t \leftarrow \Gamma(t)\,Q^\top \alpha_{t-1}$
end for
$Z \leftarrow \mathbf{1}_M^\top \alpha_T$ ▷ marginal likelihood $p(X_{1:T})$
Backward pass:
$\beta_T \leftarrow \mathbf{1}_M$
for $t = T-1, \ldots, 1$ **do**
  $\beta_t \leftarrow Q\,\Gamma(t+1)\,\beta_{t+1}$
end for
Smoothing:
for $t = 1, \ldots, T$ **do**
  $\psi_t \leftarrow (\alpha_t \circ \beta_t)/Z$
end for
for $t = 1, \ldots, T-1$ **do**
  $\phi_t \leftarrow \mathrm{diag}(\alpha_t)\,Q\,\Gamma(t+1)\;\mathrm{diag}(\beta_{t+1})/Z$
end for
return $(\psi_t), (\phi_t), Z$

3.10 Learning the HMM Parameters via EM

3.10.1 Complete-Data Log-Likelihood

Using indicator variables to expand the categorical H_t (the same trick as for GMM):

$$\log \pi_{H_1} = \sum_{m=1}^{M} \mathbf{1}\{H_1 = m\} \log \pi_m,$$

$$\log Q_{H_t, H_{t+1}} = \sum_{m,m'} \mathbf{1}\{H_t = m, H_{t+1} = m'\} \log Q_{m,m'},$$

$$\log p(X_t \mid H_t) = \sum_{m=1}^{M} \mathbf{1}\{H_t = m\} \log p(X_t \mid H_t = m).$$

Therefore,

$$\begin{aligned} \log p_\theta(H_{1:T}, X_{1:T}) = &\sum_{m} \mathbf{1}\{H_1 = m\} \log \pi_m \\ &+ \sum_{t=1}^{T-1} \sum_{m,m'} \mathbf{1}\{H_t = m, H_{t+1} = m'\} \log Q_{m,m'} \\ &+ \sum_{t=1}^{T} \sum_{m} \mathbf{1}\{H_t = m\} \log p(X_t \mid H_t = m). \end{aligned}$$

3.10.2 E-Step

Taking the expectation under the posterior $p_{\theta_i}(H_{1:T} \mid X_{1:T})$ and using

$$\mathbb{E}[\mathbf{1}\{H_t = m\}] = \psi(t, m), \qquad \mathbb{E}[\mathbf{1}\{H_t = m, H_{t+1} = m'\}] = \phi(t, m, m'),$$

we obtain the expected complete-data log-likelihood:

HMM expected complete-data log-likelihood

$$\begin{aligned} \mathcal{Q}(\theta, \theta_i) = &\underbrace{\sum_{m=1}^{M} \psi(1, m) \log \pi_m}_{\text{initial term}} \\ &+ \underbrace{\sum_{t=1}^{T-1} \sum_{m,m'} \phi(t, m, m') \log Q_{m,m'}}_{\text{transition term}} \\ &+ \underbrace{\sum_{t=1}^{T} \sum_{m=1}^{M} \psi(t, m) \log p(X_t \mid H_t = m)}_{\text{emission term}}. \end{aligned}$$

The smoothing probabilities ψ and ϕ are computed by Algorithm 2.

3.10.3 M-Step: Update of π

We solve

$$\max_{\pi} \sum_{m=1}^{M} \psi(1,m) \log \pi_m \qquad \text{s.t.} \quad \sum_{m=1}^{M} \pi_m = 1.$$

Proof

Step 1 (Lagrangian).

$$\mathcal{L}(\pi, \lambda) = \sum_{m=1}^{M} \psi(1,m) \log \pi_m - \lambda \left(\sum_{m=1}^{M} \pi_m - 1 \right).$$

Step 2 (first-order condition).

$$\frac{\partial \mathcal{L}}{\partial \pi_m} = \frac{\psi(1,m)}{\pi_m} - \lambda = 0 \implies \pi_m = \frac{\psi(1,m)}{\lambda}.$$

Step 3 (determine λ). Summing $\pi_m = 1$ and using $\sum_m \psi(1,m) = 1$:

$$1 = \sum_{m=1}^{M} \pi_m = \frac{1}{\lambda} \sum_{m=1}^{M} \psi(1,m) = \frac{1}{\lambda} \implies \lambda = 1.$$

Step 4 (conclusion).

$$\boxed{\pi_m^{(i+1)} = \psi(1,m).}$$

3.10.4 M-Step: Update of Q

We solve

$$\max_{Q} \sum_{t=1}^{T-1} \sum_{m,m'} \phi(t,m,m') \log Q_{m,m'} \qquad \text{s.t.} \quad \sum_{m'=1}^{M} Q_{m,m'} = 1 \ \ \forall m.$$

Proof

Step 1 (Lagrangian, one multiplier per row).

$$\mathcal{L}(Q, \lambda) = \sum_{t=1}^{T-1} \sum_{m,m'} \phi(t,m,m') \log Q_{m,m'} - \sum_{m=1}^{M} \lambda_m \left(\sum_{m'=1}^{M} Q_{m,m'} - 1 \right).$$

Step 2 (first-order condition).

$$\frac{\partial \mathcal{L}}{\partial Q_{m,m'}} = \frac{\sum_{t=1}^{T-1} \phi(t,m,m')}{Q_{m,m'}} - \lambda_m = 0 \implies Q_{m,m'} = \frac{\sum_{t=1}^{T-1} \phi(t,m,m')}{\lambda_m}.$$

Step 3 (determine λ_m). Using the marginalization identity $\sum_{m'} \phi(t, m, m') = \psi(t, m)$:

$$1 = \sum_{m'=1}^{M} Q_{m,m'} = \frac{1}{\lambda_m} \sum_{t=1}^{T-1} \underbrace{\sum_{m'=1}^{M} \phi(t, m, m')}_{\psi(t,m)} = \frac{\sum_{t=1}^{T-1} \psi(t, m)}{\lambda_m}.$$

Therefore $\lambda_m = \sum_{t=1}^{T-1} \psi(t, m)$.

Step 4 (conclusion).

$$\boxed{Q_{m,m'}^{(i+1)} = \frac{\sum_{t=1}^{T-1} \phi(t, m, m')}{\sum_{t=1}^{T-1} \psi(t, m)}.}$$

3.10.5 M-Step: Gaussian Emissions

For $X_t \mid H_t = m \sim \mathcal{N}(\mu_m, \Sigma_m)$, the emission term is

$$J(\mu, \Sigma) = -\sum_{t=1}^{T} \sum_{m=1}^{M} \psi(t, m) \log \mathcal{N}(X_t \mid \mu_m, \Sigma_m),$$

with $\log \mathcal{N}(X_t \mid \mu_m, \Sigma_m) = -\frac{d}{2} \log(2\pi) - \frac{1}{2} \log \det \Sigma_m - \frac{1}{2}(X_t - \mu_m)^\top \Sigma_m^{-1}(X_t - \mu_m)$.

Update of μ_m.

Proof

Step 1 (gradient identity). Recall that $\nabla_\mu (v - \mu)^\top \Sigma^{-1}(v - \mu) = -2\Sigma^{-1}(v - \mu)$.

Step 2 (first-order condition).

$$\nabla_{\mu_m} J = -\sum_{t=1}^{T} \psi(t, m)\, \Sigma_m^{-1}(X_t - \mu_m) = 0.$$

Step 3 (solve). Left-multiply by Σ_m:

$$\sum_{t=1}^{T} \psi(t, m)(X_t - \mu_m) = 0 \implies \boxed{\mu_m^{(i+1)} = \frac{\sum_{t=1}^{T} \psi(t, m)\, X_t}{\sum_{t=1}^{T} \psi(t, m)}.}$$

Update of Σ_m.

Proof

Step 1 (reparameterize). Set $\Omega_m = \Sigma_m^{-1}$. Then $\log\det\Sigma_m = -\log\det\Omega_m$, and

$$J(\mu,\Omega) = \sum_{t,m} \psi(t,m)\left[-\frac{1}{2}\log\det\Omega_m + \frac{1}{2}(X_t-\mu_m)^\top\Omega_m(X_t-\mu_m)\right] + \text{const.}$$

Step 2 (matrix gradients). Use the standard matrix identities $\nabla_\Omega \log\det\Omega = \Omega^{-1} = \Sigma$ and $\nabla_\Omega(v^\top\Omega v) = vv^\top$.

Step 3 (first-order condition).

$$\nabla_{\Omega_m} J = \sum_{t=1}^{T} \psi(t,m)\left[-\frac{1}{2}\Sigma_m + \frac{1}{2}(X_t-\mu_m)(X_t-\mu_m)^\top\right] = 0.$$

Step 4 (solve).

$$\boxed{\Sigma_m^{(i+1)} = \frac{\sum_{t=1}^{T}\psi(t,m)\,(X_t-\mu_m^{(i+1)})(X_t-\mu_m^{(i+1)})^\top}{\sum_{t=1}^{T}\psi(t,m)}.}$$

3.10.6 M-Step: Product-Multinomial Emissions

For categorical observations $X_t = (X_t^1,\dots,X_t^d)$ with $X_t^c \in \{1,\dots,K\}$ and emission parameters $O_{m,c,k}$, the emission term decomposes:

$$\log p(X_t \mid H_t = m) = \sum_{c=1}^{d} \log O_{m,c,X_t^c} = \sum_{c=1}^{d}\sum_{k=1}^{K} \mathbf{1}\{X_t^c = k\}\log O_{m,c,k}.$$

The identity above is valid because exactly one indicator $\mathbf{1}\{X_t^c = k\}$ equals 1 for any fixed (t,c).

The emission term of $\mathcal{Q}$ thus becomes

$$\sum_{t=1}^{T}\sum_{c=1}^{d}\sum_{m=1}^{M}\sum_{k=1}^{K} \psi(t,m)\,\mathbf{1}\{X_t^c = k\}\log O_{m,c,k}.$$

Update of O.

Proof

Step 1 (Lagrangian, one multiplier per (m,c)).

$$\mathcal{L}(O,\lambda) = \sum_{t,c,m,k} \psi(t,m)\,\mathbf{1}\{X_t^c = k\}\log O_{m,c,k} - \sum_{m,c}\lambda_{m,c}\left(\sum_k O_{m,c,k} - 1\right).$$

Step 2 (first-order condition).

$$\frac{\sum_t \psi(t,m)\,\mathbf{1}\{X_t^c = k\}}{O_{m,c,k}} = \lambda_{m,c} \quad \Longrightarrow \quad O_{m,c,k} = \frac{\sum_t \psi(t,m)\,\mathbf{1}\{X_t^c = k\}}{\lambda_{m,c}}.$$

Step 3 (determine $\lambda_{m,c}$). Sum over k and use $\sum_k \mathbf{1}\{X_t^c = k\} = 1$:

$$1 = \sum_k O_{m,c,k} = \frac{\sum_t \psi(t,m)}{\lambda_{m,c}} \quad \Longrightarrow \quad \lambda_{m,c} = \sum_{t=1}^{T} \psi(t,m).$$

Step 4 (conclusion).

$$\boxed{O_{m,c,k}^{(i+1)} = \frac{\sum_{t=1}^{T} \psi(t,m)\,\mathbf{1}\{X_t^c = k\}}{\sum_{t=1}^{T} \psi(t,m)}.}$$

Numerator: expected number of times feature c took value k while in state m. Denominator: expected number of visits to state m. The update is a posterior-weighted multinomial MLE.

3.11 Recap of the EM Loop for HMM

We have all the pieces. The HMM is trained by repeating the following cycle until the marginal log-likelihood stops increasing.

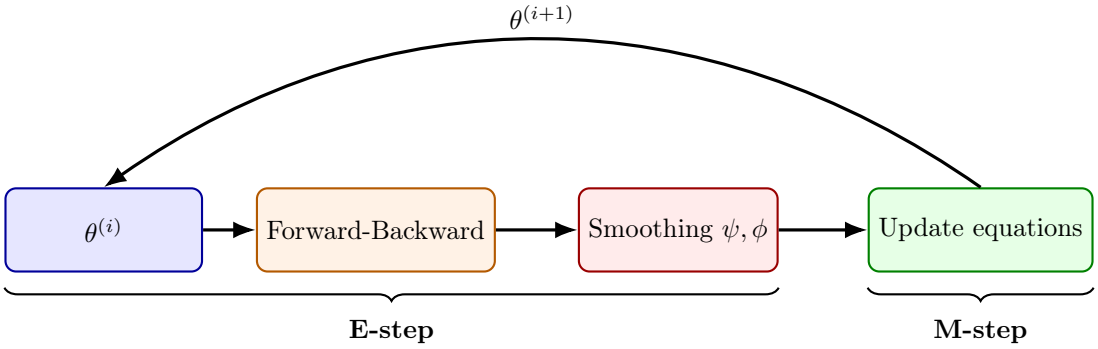


Figure 9: One EM iteration for the HMM. The E-step computes the Forward-Backward variables and the smoothing probabilities; the M-step applies the closed-form update equations; the updated parameters $\theta^{(i+1)}$ are fed back as the new $\theta^{(i)}$ and the loop repeats until convergence.

Algorithm 3 EM for HMM (Gaussian or Product-Multinomial emissions)

Require: Observations $X_{1:T}$, number of states M, iterations I
Ensure: Optimized parameters $\theta^\star$

Initialize $\theta^{(0)}$ (e.g. via K-means for Gaussian, uniform for multinomial)
for $i = 0, 1, \ldots, I-1$ **do**
E-step (Forward-Backward):
    Build emission matrices $\Gamma(t)_{mm} = p_{\theta^{(i)}}(X_t \mid H_t = m)$
    Run Algorithm 2 to obtain $\psi(t, m)$, $\phi(t, m, m')$, and $Z = p(X_{1:T})$
M-step:
    $\pi_m^{(i+1)} \leftarrow \psi(1, m)$
    $Q_{m,m'}^{(i+1)} \leftarrow \dfrac{\sum_{t=1}^{T-1} \phi(t, m, m')}{\sum_{t=1}^{T-1} \psi(t, m)}$
    if Gaussian **then**
        $\mu_m^{(i+1)} \leftarrow \dfrac{\sum_t \psi(t, m) X_t}{\sum_t \psi(t, m)}$
        $\Sigma_m^{(i+1)} \leftarrow \dfrac{\sum_t \psi(t, m)(X_t - \mu_m^{(i+1)})(X_t - \mu_m^{(i+1)})^\top}{\sum_t \psi(t, m)}$
    else if Product-multinomial **then**
        $O_{m,c,k}^{(i+1)} \leftarrow \dfrac{\sum_t \psi(t, m) \mathbf{1}\{X_t^c = k\}}{\sum_t \psi(t, m)}$
    end if
end for
return $\theta^{(I)}$

4 Prediction: the Next Hidden State and Observation

Once the HMM is trained, we typically want to forecast either the **next hidden state** or the **next observation** given the observed sequence.

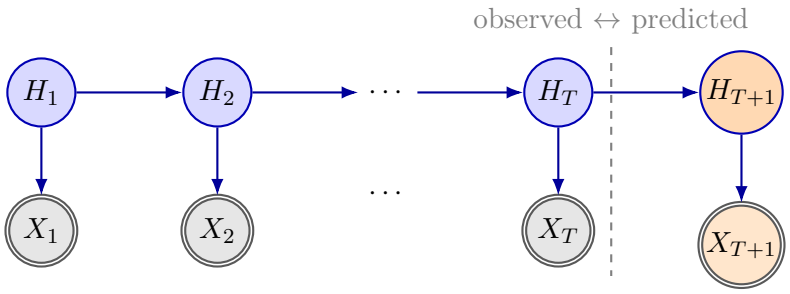


Figure 10: Prediction setting: given $X_{1:T}$, forecast the next hidden state and the next observation.

4.1 Predicting the Next Hidden State

We want $p(H_{T+1} = h \mid X_{1:T})$ for each h.

Proof

Step 1. Marginalize over the current hidden state H_T.

$$p(H_{T+1} = h \mid X_{1:T}) = \sum_{h'=1}^{M} p(H_{T+1} = h, H_T = h' \mid X_{1:T}).$$

Step 2. Factor using the Markov property: given H_T, the next state H_{T+1} is independent of past observations.

$$p(H_{T+1} = h, H_T = h' \mid X_{1:T}) = \underbrace{p(H_{T+1} = h \mid H_T = h')}_{Q_{h',h}} \cdot \underbrace{p(H_T = h' \mid X_{1:T})}_{\xi(T,h')}.$$

Step 3. Combine.

$$p(H_{T+1} = h \mid X_{1:T}) = \sum_{h'=1}^{M} Q_{h',h}\, \xi(T, h').$$

Next-hidden-state prediction

In vector form,

$$p(H_{T+1} \mid X_{1:T}) = Q^\top \xi_T,$$

where ξ_T is the filtering distribution at the final time (output of the forward pass).

4.2 Predicting the Next Observation (Discrete Case)

We focus on the **product-multinomial** emission model, where $X_{T+1} = (X_{T+1}^1, \ldots, X_{T+1}^d)$ with each $X_{T+1}^c \in \{1, \ldots, K\}$ and emission parameters $O_{m,c,k}$.

Proof

Step 1. Apply the tower property by conditioning on the next hidden state.

$$p(X_{T+1}^c = k \mid X_{1:T}) = \sum_{h=1}^{M} p(X_{T+1}^c = k \mid H_{T+1} = h) \cdot p(H_{T+1} = h \mid X_{1:T}).$$

Step 2. Use the emission model $p(X_{T+1}^c = k \mid H_{T+1} = h) = O_{h,c,k}$ and write $\eta_h := p(H_{T+1} = h \mid X_{1:T}) = (Q^\top \xi_T)_h$.

$$p(X_{T+1}^c = k \mid X_{1:T}) = \sum_{h=1}^{M} \eta_h \, O_{h,c,k}.$$

Next-observation prediction (discrete case)

For each feature $c \in \{1, \ldots, d\}$ and each value $k \in \{1, \ldots, K\}$,

$$\boxed{p(X_{T+1}^c = k \mid X_{1:T}) = \sum_{h=1}^{M} \eta_h \, O_{h,c,k}, \qquad \eta = Q^\top \xi_T.}$$

This is a mixture: the predictive emission for feature c is an η-weighted combination of the per-state multinomials.

5 Variational Autoencoder: a Continuous Latent-Variable Model

So far we have worked with discrete latents (the cluster index in a GMM, the state in an HMM). We now consider a continuous latent variable $\mathbf{z} \in \mathbb{R}^d$ and a Gaussian observation model. The posterior $p_\theta(\mathbf{z} \mid \mathbf{x})$ is no longer tractable; instead, we approximate it with a parametric family $q_\phi(\mathbf{z} \mid \mathbf{x})$. The exercise below walks through the derivation of the Variational Autoencoder (VAE) objective.

Exercise: VAEs

We consider a model where each observation $\mathbf{x} \in \mathbb{R}^p$ is generated using an unobserved continuous vector $\mathbf{z} \in \mathbb{R}^d$.

Generative model.

$$\mathbf{z} \sim \mathcal{N}(0, I_d), \qquad \mathbf{x} \mid \mathbf{z} \sim \mathcal{N}(\mu_\theta(\mathbf{z}), \sigma^2 I_p),$$

where $\mu_\theta(\mathbf{z})$ is a function from $\mathbb{R}^d$ to $\mathbb{R}^p$, parameterized by θ. In practice, μ_θ is a neural network that takes $\mathbf{z}$ as input and outputs the mean of the Gaussian distribution of $\mathbf{x}$.
The joint density is

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}) = \mathcal{N}(\mathbf{x} \mid \mu_\theta(\mathbf{z}), \sigma^2 I_p)\, \mathcal{N}(\mathbf{z} \mid 0, I_d).$$

(1) Write the marginal likelihood $p_\theta(\mathbf{x})$ by integrating out $\mathbf{z}$, and then write the log-marginal $\log p_\theta(\mathbf{x})$.

(2) Let $q(\mathbf{z} \mid \mathbf{x})$ be any density on $\mathbb{R}^d$. Show that

$$\log p_\theta(\mathbf{x}) = \mathcal{L}(\theta, q; \mathbf{x}) + \mathrm{KL}(q(\mathbf{z} \mid \mathbf{x}) \,\|\, p_\theta(\mathbf{z} \mid \mathbf{x})),$$

where

$$\mathcal{L}(\theta, q; \mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}, \mathbf{z})] - \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log q(\mathbf{z} \mid \mathbf{x})].$$

Conclude that $\mathcal{L}(\theta, q; \mathbf{x})$ is a lower bound (ELBO) on $\log p_\theta(\mathbf{x})$.

(3) For fixed θ, show that the lower bound is maximized over q when $q^*(\mathbf{z} \mid \mathbf{x}) = p_\theta(\mathbf{z} \mid \mathbf{x})$.

In practice, $p_\theta(\mathbf{z} \mid \mathbf{x})$ is not tractable, so we approximate it by a family

$$q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}\big(\mathbf{z} \mid m_\phi(\mathbf{x}), \mathrm{diag}(s_\phi^2(\mathbf{x}))\big),$$

where $m_\phi(\mathbf{x})$ and $s_\phi(\mathbf{x})$ are functions from $\mathbb{R}^p$ to $\mathbb{R}^d$, parameterized by ϕ (the encoder network*).*

(4) Show that the ELBO can be rewritten as

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] - \mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z})),$$

where $p(\mathbf{z}) = \mathcal{N}(\mathbf{z} \mid 0, I_d)$.

(5) Explain how a sample from $q_\phi(\mathbf{z} \mid \mathbf{x})$ can be written as

$$\mathbf{z} = m_\phi(\mathbf{x}) + s_\phi(\mathbf{x}) \odot \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I_d),$$

where $\odot$ denotes element-wise multiplication. Then explain why this representation allows us to compute gradients with respect to ϕ using standard calculus and a Monte Carlo approximation.

(6) Assume $p(\mathbf{z}) = \mathcal{N}(\mathbf{z} \mid 0, I_d)$ and $q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}(\mathbf{z} \mid m_\phi(\mathbf{x}), \mathrm{diag}(s_\phi^2(\mathbf{x})))$.

(a) Derive the closed-form expression of $\mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z}))$.

(b) Using one sample $\varepsilon_i \sim \mathcal{N}(0, I_d)$ per datapoint $\mathbf{x}_i$ in a dataset $\{\mathbf{x}_i\}_{i=1}^N$, write the final loss function (negative ELBO) that should be minimized.

Solutions

Solution

1/ The marginal likelihood is obtained by integrating out the latent variable:

$$p_\theta(\mathbf{x}) = \int_{\mathbb{R}^d} p_\theta(\mathbf{x}, \mathbf{z})\, d\mathbf{z} = \int_{\mathbb{R}^d} p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z})\, d\mathbf{z}.$$

Taking the logarithm gives

$$\log p_\theta(\mathbf{x}) = \log \int_{\mathbb{R}^d} p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z})\, d\mathbf{z}.$$

Solution

2/ Insert $q(\mathbf{z} \mid \mathbf{x})$ into the integrand of the marginal:

$$\log p_\theta(\mathbf{x}) = \log \int q(\mathbf{z} \mid \mathbf{x}) \, \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z} \mid \mathbf{x})}\, d\mathbf{z} = \log \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \left[\frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z} \mid \mathbf{x})} \right].$$

By Jensen's inequality applied to log (concave),

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z} \mid \mathbf{x})} \right] = \mathcal{L}(\theta, q; \mathbf{x}).$$

Using Bayes' rule $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{z} \mid \mathbf{x}) p_\theta(\mathbf{x})$, we obtain

$$\mathcal{L}(\theta, q; \mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{z} \mid \mathbf{x}) p_\theta(\mathbf{x})}{q(\mathbf{z} \mid \mathbf{x})} \right] = \log p_\theta(\mathbf{x}) - \mathrm{KL}(q(\mathbf{z} \mid \mathbf{x}) \,\|\, p_\theta(\mathbf{z} \mid \mathbf{x})) \,.$$

Rearranging,

$$\log p_\theta(\mathbf{x}) = \mathcal{L}(\theta, q; \mathbf{x}) + \mathrm{KL}(q(\mathbf{z} \mid \mathbf{x}) \,\|\, p_\theta(\mathbf{z} \mid \mathbf{x})) \,.$$

Since the KL divergence is nonnegative, $\mathcal{L}(\theta, q; \mathbf{x})$ is a lower bound on $\log p_\theta(\mathbf{x})$.

Solution

3/ From the identity

$$\log p_\theta(\mathbf{x}) = \mathcal{L}(\theta, q; \mathbf{x}) + \mathrm{KL}(q(\mathbf{z} \mid \mathbf{x}) \,\|\, p_\theta(\mathbf{z} \mid \mathbf{x}))\,,$$

and the fact that $\log p_\theta(\mathbf{x})$ does not depend on q, maximizing $\mathcal{L}(\theta, q; \mathbf{x})$ over q is equivalent to minimizing the KL divergence. The minimum value of the KL divergence is 0, achieved iff

$$q(\mathbf{z} \mid \mathbf{x}) = p_\theta(\mathbf{z} \mid \mathbf{x}).$$

Interpretation: the best possible choice of the auxiliary density $q(\mathbf{z} \mid \mathbf{x})$ is to set it equal to the true posterior distribution of $\mathbf{z}$ given $\mathbf{x}$ under the generative model.

Solution

4/ Starting from

$$\mathcal{L}(\theta, q; \mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}, \mathbf{z})] - \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log q(\mathbf{z} \mid \mathbf{x})],$$

and using the factorization $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})$:

$$\mathcal{L}(\theta, q; \mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] + \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{z})] - \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log q(\mathbf{z} \mid \mathbf{x})].$$

The last two terms combine into $-\mathrm{KL}(q(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z}))$. Replacing q by q_ϕ gives

$$\boxed{\mathcal{L}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] - \mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z}))\,.}$$

Interpretation: the first term encourages high likelihood of $\mathbf{x}$ given $\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x})$ (reconstruction quality), while the second term penalizes $q_\phi(\mathbf{z} \mid \mathbf{x})$ when it deviates from the prior $p(\mathbf{z})$ (latent regularizer).

Solution

5/ **(the reparameterization trick)** If $\mathbf{z} \sim \mathcal{N}(m, \mathrm{diag}(s^2))$, then we can write

$$\mathbf{z} = m + s \odot \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I_d),$$

because adding a vector of independent standard normal variables scaled by s and shifted by m gives exactly this Gaussian distribution. Thus, for $q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}(\mathbf{z} \mid m_\phi(\mathbf{x}), \mathrm{diag}(s_\phi^2(\mathbf{x})))$:

$$\mathbf{z} = m_\phi(\mathbf{x}) + s_\phi(\mathbf{x}) \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I_d).$$

The expectation becomes

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] = \mathbb{E}_{\varepsilon \sim \mathcal{N}(0, I_d)}\big[\log p_\theta(\mathbf{x} \mid m_\phi(\mathbf{x}) + s_\phi(\mathbf{x}) \odot \varepsilon)\big].$$

We approximate this expectation by a Monte Carlo average over independent samples $\varepsilon^{(1)}, \ldots, \varepsilon^{(L)}$:

$$\frac{1}{L} \sum_{\ell=1}^{L} \log p_\theta\big(\mathbf{x} \mid m_\phi(\mathbf{x}) + s_\phi(\mathbf{x}) \odot \varepsilon^{(\ell)}\big).$$

For each fixed $\varepsilon^{(\ell)}$, the quantity inside the sum is a deterministic differentiable function of θ and ϕ, since m_ϕ and s_ϕ are differentiable in ϕ and the randomness is entirely in $\varepsilon^{(\ell)}$. Therefore, gradients with respect to ϕ can be computed using standard differentiation rules, treating each $\varepsilon^{(\ell)}$ as an input.

Solution

6/ (a) Closed-form KL. For $q = \mathcal{N}(m, \mathrm{diag}(s^2))$ and $p = \mathcal{N}(0, I_d)$,

$$\mathrm{KL}(q \,\|\, p) = \frac{1}{2} \sum_{k=1}^{d} \Big(s_k^2 + m_k^2 - 1 - \log s_k^2 \Big).$$

With $m = m_\phi(\mathbf{x})$ and $s = s_\phi(\mathbf{x})$:

$$\mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z})) = \frac{1}{2} \sum_{k=1}^{d} \Big(s_{\phi,k}^2(\mathbf{x}) + m_{\phi,k}^2(\mathbf{x}) - 1 - \log s_{\phi,k}^2(\mathbf{x}) \Big).$$

6/ (b) Final loss. For a single datapoint $\mathbf{x}_i$ with one noise sample $\varepsilon_i \sim \mathcal{N}(0, I_d)$, set $\mathbf{z}_i = m_\phi(\mathbf{x}_i) + s_\phi(\mathbf{x}_i) \odot \varepsilon_i$. Then

$$\mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}_i \mid \mathbf{z})] \approx \log p_\theta(\mathbf{x}_i \mid \mathbf{z}_i),$$

and the per-sample negative ELBO is

$$\ell(\mathbf{x}_i; \theta, \phi) = -\log p_\theta(\mathbf{x}_i \mid \mathbf{z}_i) + \mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}_i) \,\|\, p(\mathbf{z})) \,.$$

Summing over the dataset, the loss to minimize is

$$\boxed{\mathcal{J}(\theta, \phi) = \frac{1}{N} \sum_{i=1}^{N} \Big(-\log p_\theta(\mathbf{x}_i \mid \mathbf{z}_i) + \mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}_i) \,\|\, p(\mathbf{z})) \Big),}$$

with $\mathbf{z}_i = m_\phi(\mathbf{x}_i) + s_\phi(\mathbf{x}_i) \odot \varepsilon_i$ and $\varepsilon_i \sim \mathcal{N}(0, I_d)$. This is the standard VAE training objective.