Learning with Hidden Variables

Handout Notes - Generative and Deep Learning (GDL)

Davide Bacciu - University of Pisa

Notation. Random variables are uppercase (e.g., X, Z) and observed values are lowercase (e.g., x, z). A dataset is $\mathcal{D} = \{x_1, \ldots, x_N\}$ with $N = |\mathcal{D}|$. Parameters are denoted by θ. The log-likelihood is $\ell(\theta) = \log p(\mathcal{D} \mid \theta)$. In mixture models we use M components and a discrete latent assignment $Z_i \in \{1, \ldots, M\}$ per datum; *responsibilities* are $r_{im} = p(Z_i = m \mid x_i, \theta)$.

1 Why hidden variables?

Many probabilistic models explain observed data X by introducing additional variables Z that are not observed. Latent variables can represent unobserved causes (e.g., cluster identity), missing structure (e.g., a topic in a document), or compressed representations. The resulting model can be expressive and interpretable, but learning becomes harder because the likelihood of the observed data involves summing or integrating over the unobserved variables.

Two common forms of incompleteness are:

- **Missing entries in X:** some measurements are absent (a data issue).
- **Latent variables Z:** the model posits variables we never observe (a modeling choice).

This handout focuses on the second case, which is where Expectation–Maximization (EM) is classically applied.

2 Latent-variable models: joint and marginal likelihood

A latent-variable model specifies a joint distribution

$$p(x, z \mid \theta) = p(x \mid z, \theta)\, p(z \mid \theta).$$

The likelihood of an observed point x is obtained by marginalizing out z:

$$p(x \mid \theta) = \sum_z p(x, z \mid \theta) \quad \text{(discrete } Z\text{)}, \qquad p(x \mid \theta) = \int p(x, z \mid \theta)\, dz \quad \text{(continuous } Z\text{)}.$$

For i.i.d. observations $\mathcal{D} = \{x_1, \ldots, x_N\}$,

$$p(\mathcal{D} \mid \theta) = \prod_{i=1}^{N} p(x_i \mid \theta), \qquad \ell(\theta) = \sum_{i=1}^{N} \log p(x_i \mid \theta).$$

The core difficulty is that each $p(x_i \mid \theta)$ contains a sum/integral over z_i *inside a log*, producing a "log-of-sum" structure that is awkward to optimize directly.

3 Complete-data vs. incomplete-data learning

If we had access to the complete data $\{(x_i, z_i)\}_{i=1}^{N}$, learning would often be straightforward, because the complete-data likelihood typically factorizes cleanly:

$$p(\mathcal{D}, \mathcal{Z} \mid \theta) = \prod_{i=1}^{N} p(x_i, z_i \mid \theta), \quad \text{where} \quad \mathcal{Z} = \{z_1, \ldots, z_N\}.$$

EM exploits this by iterating:

Infer the latent variables probabilistically under the current parameters, then update the parameters as if the latent variables were observed—but weighted by those posterior probabilities.

This "soft completion" viewpoint is the classical intuition behind EM.

4 The EM algorithm (classical statement)

Fix a current parameter value $\theta^{(k)}$. For any candidate θ, define

$$Q(\theta \mid \theta^{(k)}) \;=\; \mathbb{E}_{\mathcal{Z}\sim p(\mathcal{Z}\mid\mathcal{D},\theta^{(k)})}\big[\,\log p(\mathcal{D},\mathcal{Z}\mid\theta)\,\big].$$

In the E-step we compute the *current* posterior under $\theta^{(k)}$. Then, in the M-step we choose $\theta^{(k+1)}$ to increase $Q(\theta \mid \theta^{(k)})$.

Putting the above into algorithmic form:

- **Initialize:** choose $\theta^{(0)}$.
- For $k = 0, 1, 2, \ldots$ until convergence:
  - **E-step:** compute the posterior over latents under current parameters:

$$p(\mathcal{Z} \mid \mathcal{D}, \theta^{(k)}).$$

  - **M-step:** update parameters by maximizing the expected complete-data log-likelihood:

$$\theta^{(k+1)} = \arg\max_{\theta}\; Q(\theta \mid \theta^{(k)}) = \arg\max_{\theta}\; \mathbb{E}_{\mathcal{Z}\sim p(\mathcal{Z}\mid\mathcal{D},\theta^{(k)})}\big[\log p(\mathcal{D},\mathcal{Z}\mid\theta)\big].$$

In many models, the E-step involves computing marginal posteriors of the form $p(z_i \mid x_i, \theta^{(k)})$, and the M-step resembles complete-data ML with counts replaced by their expectations under those posteriors.

5 Warm-up: ML for a univariate Gaussian (complete data)

Before mixtures, recall ML for a single Gaussian with complete data. Assume $x_1, \ldots, x_N \in \mathbb{R}$ are i.i.d. from $\mathcal{N}(\mu, \sigma^2)$ with parameters $\theta = (\mu, \sigma^2)$. Ignoring constants,

$$\ell(\mu, \sigma^2) = -\frac{N}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i - \mu)^2 + \text{const.}$$

Worked example — MLE for μ and σ^2 in a univariate Gaussian

Differentiate w.r.t. μ:

$$\frac{\partial \ell}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{N}(x_i - \mu) = 0 \implies \boxed{\mu_{\text{ML}} = \frac{1}{N}\sum_{i=1}^{N} x_i.}$$

Differentiate w.r.t. σ^2:

$$\frac{\partial \ell}{\partial \sigma^2} = -\frac{N}{2}\frac{1}{\sigma^2} + \frac{1}{2(\sigma^2)^2}\sum_{i=1}^{N}(x_i - \mu)^2 = 0 \implies \boxed{(\sigma^2)_{\text{ML}} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_{\text{ML}})^2.}$$

This is the template that will reappear in the M-step for Gaussian mixtures, except that the effective sample size becomes a *soft* count.

6 Gaussian mixture models (GMMs)

A Gaussian mixture model assumes each observation is generated by one of M Gaussian components, but the component identity is unobserved. This missing identity is captured by a latent variable.

6.1 Model definition

For each datum $x_i \in \mathbb{R}^d$:

1. Draw an assignment $Z_i \in \{1, \dots, M\}$ with mixing weights $\pi_1, \dots, \pi_M$:

$$p(Z_i = m \mid \theta) = \pi_m, \qquad \pi_m \geq 0, \quad \sum_{m=1}^{M} \pi_m = 1.$$

2. Draw the observation from the assigned Gaussian:

$$p(x_i \mid Z_i = m, \theta) = \mathcal{N}(x_i \mid \mu_m, \Sigma_m),$$

with $\mu_m \in \mathbb{R}^d$ and $\Sigma_m \succ 0$.

The marginal mixture density is

$$p(x_i \mid \theta) = \sum_{m=1}^{M} \pi_m \mathcal{N}(x_i \mid \mu_m, \Sigma_m),$$

and the observed-data log-likelihood is

$$\ell(\theta) = \sum_{i=1}^{N} \log \left(\sum_{m=1}^{M} \pi_m \mathcal{N}(x_i \mid \mu_m, \Sigma_m) \right).$$

7 EM for GMMs

7.1 E-step: responsibilities

The E-step computes the posterior $p(Z_i \mid x_i, \theta^{(k)})$ for each i. Because Z_i is discrete and the model is simple, this posterior is available in closed form.

Worked example — E-step responsibilities in a GMM

By Bayes' rule,

$$r_{im}^{(k)} \;=\; p(Z_i = m \mid x_i, \theta^{(k)}) = \frac{p(x_i \mid Z_i = m, \theta^{(k)})\, p(Z_i = m \mid \theta^{(k)})}{\sum_{m'=1}^{M} p(x_i \mid Z_i = m', \theta^{(k)})\, p(Z_i = m' \mid \theta^{(k)})}.$$

Substitute the mixture terms:

$$\boxed{r_{im}^{(k)} = \frac{\pi_m^{(k)}\, \mathcal{N}(x_i \mid \mu_m^{(k)}, \Sigma_m^{(k)})}{\sum_{m'=1}^{M} \pi_{m'}^{(k)}\, \mathcal{N}(x_i \mid \mu_{m'}^{(k)}, \Sigma_{m'}^{(k)})}}.$$

These satisfy $0 \leq r_{im}^{(k)} \leq 1$ and $\sum_{m=1}^{M} r_{im}^{(k)} = 1$ for each i.

The responsibilities are the classical "soft assignments" of points to components.

7.2 M-step: step-by-step derivation of the update equations

In the M-step we maximize the auxiliary function

$$Q(\theta \mid \theta^{(k)}) = \mathbb{E}_{\mathcal{Z}\sim p(\mathcal{Z}\mid\mathcal{D},\theta^{(k)})}\big[\log p(\mathcal{D},\mathcal{Z}\mid\theta)\big],$$

where, for a GMM, the expectation replaces the (unobserved) component indicators with their posterior expectations (responsibilities) $r_{im}^{(k)} = p(Z_i = m \mid x_i, \theta^{(k)})$.

Step 1: Write the complete-data log-likelihood. Introduce one-hot indicators $z_{im} \in \{0,1\}$ with $\sum_{m=1}^{M} z_{im} = 1$ and $z_{im} = 1 \Leftrightarrow Z_i = m$. For one observation,

$$p(x_i, z_i \mid \theta) = \prod_{m=1}^{M} \Big(\pi_m \mathcal{N}(x_i \mid \mu_m, \Sigma_m)\Big)^{z_{im}},$$

so the complete-data log-likelihood for $(\mathcal{D},\mathcal{Z})$ is

$$\log p(\mathcal{D},\mathcal{Z}\mid\theta) = \sum_{i=1}^{N}\sum_{m=1}^{M} z_{im}\Big[\log\pi_m + \log\mathcal{N}(x_i \mid \mu_m, \Sigma_m)\Big].$$

Step 2: Take expectation w.r.t. the current posterior over latents. Under the current parameters $\theta^{(k)}$, we have

$$\mathbb{E}[z_{im} \mid x_i, \theta^{(k)}] = r_{im}^{(k)}.$$

Hence,

$$Q(\theta \mid \theta^{(k)}) = \sum_{i=1}^{N}\sum_{m=1}^{M} r_{im}^{(k)}\Big[\log\pi_m + \log\mathcal{N}(x_i \mid \mu_m, \Sigma_m)\Big].$$

Define the effective membership (soft count) of component m:

$$N_m^{(k)} = \sum_{i=1}^{N} r_{im}^{(k)}.$$

We now maximize Q w.r.t. $\{\pi_m, \mu_m, \Sigma_m\}_{m=1}^{M}$ subject to the constraints

$$\pi_m \geq 0, \qquad \sum_{m=1}^{M}\pi_m = 1, \qquad \Sigma_m \succ 0.$$

Because Q decomposes over mixture components, we can optimize π_m, μ_m, Σ_m componentwise (with the exception that the π_m are coupled by the simplex constraint).

Worked example — M-step derivation for the mixing weights π_m (with Lagrange multipliers)

Extract from Q only the terms involving π_m:

$$Q_\pi(\pi) = \sum_{i=1}^{N}\sum_{m=1}^{M} r_{im}^{(k)} \log \pi_m = \sum_{m=1}^{M}\Big(\sum_{i=1}^{N} r_{im}^{(k)}\Big)\log \pi_m = \sum_{m=1}^{M} N_m^{(k)} \log \pi_m.$$

We maximize $Q_\pi(\pi)$ subject to $\sum_{m=1}^{M}\pi_m = 1$ using a Lagrange multiplier λ:

$$\mathcal{J}(\pi,\lambda) = \sum_{m=1}^{M} N_m^{(k)} \log \pi_m + \lambda\Big(\sum_{m=1}^{M}\pi_m - 1\Big).$$

Differentiate w.r.t. π_m and set to zero:

$$\frac{\partial \mathcal{J}}{\partial \pi_m} = \frac{N_m^{(k)}}{\pi_m} + \lambda = 0 \quad\Longrightarrow\quad \pi_m = -\frac{N_m^{(k)}}{\lambda}.$$

Enforce the constraint $\sum_{m=1}^{M}\pi_m = 1$:

$$1 = \sum_{m=1}^{M}\pi_m = -\frac{1}{\lambda}\sum_{m=1}^{M} N_m^{(k)} = -\frac{1}{\lambda}\sum_{i=1}^{N}\sum_{m=1}^{M} r_{im}^{(k)}.$$

Since $\sum_{m=1}^{M} r_{im}^{(k)} = 1$ for each i, we have $\sum_m N_m^{(k)} = \sum_i 1 = N$, so $\lambda = -N$ and therefore

$$\boxed{\pi_m^{(k+1)} = \frac{N_m^{(k)}}{N}.}$$

To derive the updates for μ_m and Σ_m, we need an explicit expression for the Gaussian log-density. For $x \in \mathbb{R}^d$,

$$\log \mathcal{N}(x \mid \mu, \Sigma) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu).$$

Constants independent of (μ, Σ) can be dropped when taking argmax.

Worked example — M-step derivation for the means μ_m (weighted least squares)

Fix a component m and collect all terms in Q that depend on μ_m (holding Σ_m fixed):

$$Q_{\mu_m}(\mu_m) = \sum_{i=1}^{N} r_{im}^{(k)} \log \mathcal{N}(x_i \mid \mu_m, \Sigma_m) = -\frac{1}{2} \sum_{i=1}^{N} r_{im}^{(k)} (x_i - \mu_m)^\top \Sigma_m^{-1} (x_i - \mu_m) + \text{const.}$$

Maximizing Q_{μ_m} is equivalent to minimizing the weighted quadratic form

$$\sum_{i=1}^{N} r_{im}^{(k)} (x_i - \mu_m)^\top \Sigma_m^{-1} (x_i - \mu_m).$$

Differentiate w.r.t. μ_m (using $\nabla_\mu (x-\mu)^\top A (x-\mu) = -2A(x-\mu)$ for symmetric A):

$$\nabla_{\mu_m} Q_{\mu_m} = -\frac{1}{2} \sum_{i=1}^{N} r_{im}^{(k)} \big(-2\Sigma_m^{-1}(x_i - \mu_m) \big) = \sum_{i=1}^{N} r_{im}^{(k)} \Sigma_m^{-1} (x_i - \mu_m).$$

Set to zero:

$$\sum_{i=1}^{N} r_{im}^{(k)} \Sigma_m^{-1} (x_i - \mu_m) = 0 \quad \Longrightarrow \quad \Sigma_m^{-1} \Big(\sum_{i=1}^{N} r_{im}^{(k)} x_i - \mu_m \sum_{i=1}^{N} r_{im}^{(k)} \Big) = 0.$$

Since Σ_m^{-1} is invertible, we obtain

$$\mu_m \sum_{i=1}^{N} r_{im}^{(k)} = \sum_{i=1}^{N} r_{im}^{(k)} x_i \quad \Longrightarrow \quad \boxed{\mu_m^{(k+1)} = \frac{1}{N_m^{(k)}} \sum_{i=1}^{N} r_{im}^{(k)} x_i.}$$

Worked example — M-step derivation for the covariances Σ_m (weighted covariance)

Fix a component m and collect all terms in Q that depend on Σ_m (with μ_m fixed to $\mu_m^{(k+1)}$):

$$Q_{\Sigma_m}(\Sigma_m) = \sum_{i=1}^{N} r_{im}^{(k)} \log \mathcal{N}(x_i \mid \mu_m, \Sigma_m) = -\frac{1}{2} \sum_{i=1}^{N} r_{im}^{(k)} \Big[\log |\Sigma_m| + (x_i - \mu_m)^\top \Sigma_m^{-1} (x_i - \mu_m) \Big] + \text{const.}$$

Define the weighted scatter matrix

$$S_m = \sum_{i=1}^{N} r_{im}^{(k)} (x_i - \mu_m)(x_i - \mu_m)^\top.$$

Using $\operatorname{tr}(AB) = \operatorname{tr}(BA)$ and $(x-\mu)^\top \Sigma^{-1} (x-\mu) = \operatorname{tr}\big(\Sigma^{-1}(x-\mu)(x-\mu)^\top\big)$, we can rewrite

$$Q_{\Sigma_m}(\Sigma_m) = -\frac{1}{2} \Big[N_m^{(k)} \log |\Sigma_m| + \operatorname{tr}\big(\Sigma_m^{-1} S_m\big) \Big] + \text{const.}$$

To differentiate cleanly, set $A = \Sigma_m^{-1}$ (so $\Sigma_m = A^{-1}$). Then $\log |\Sigma_m| = -\log |A|$ and

$$Q_{\Sigma_m}(A) = -\frac{1}{2} \Big[N_m^{(k)} (-\log |A|) + \operatorname{tr}(A S_m) \Big] + \text{const} = \frac{N_m^{(k)}}{2} \log |A| - \frac{1}{2} \operatorname{tr}(A S_m) + \text{const.}$$

Differentiate w.r.t. A using $\nabla_A \log |A| = (A^{-1})^\top$ and $\nabla_A \operatorname{tr}(A S_m) = S_m^\top = S_m$:

$$\nabla_A Q_{\Sigma_m}(A) = \frac{N_m^{(k)}}{2} (A^{-1})^\top - \frac{1}{2} S_m.$$

Set to zero:

$$\frac{N_m^{(k)}}{2} (A^{-1})^\top = \frac{1}{2} S_m \quad \Longrightarrow \quad N_m^{(k)} (A^{-1})^\top = S_m.$$

Because S_m is symmetric and $A^{-1} = \Sigma_m$ is symmetric at the optimum, this yields

$$N_m^{(k)} \Sigma_m = S_m \quad \Longrightarrow \quad \boxed{\Sigma_m^{(k+1)} = \frac{1}{N_m^{(k)}} \sum_{i=1}^{N} r_{im}^{(k)} (x_i - \mu_m^{(k+1)})(x_i - \mu_m^{(k+1)})^\top.}$$

Summary of the M-step. Given responsibilities $r_{im}^{(k)}$ from the E-step, define $N_m^{(k)} = \sum_i r_{im}^{(k)}$. Then the M-step updates are

$$\pi_m^{(k+1)} = \frac{N_m^{(k)}}{N}, \qquad \mu_m^{(k+1)} = \frac{1}{N_m^{(k)}} \sum_{i=1}^{N} r_{im}^{(k)} x_i, \qquad \Sigma_m^{(k+1)} = \frac{1}{N_m^{(k)}} \sum_{i=1}^{N} r_{im}^{(k)} (x_i - \mu_m^{(k+1)})(x_i - \mu_m^{(k+1)})^\top.$$

They are exactly the complete-data ML estimators, with hard component membership indicators replaced by their posterior expectations (soft assignments).

Practical remarks. EM for GMMs is sensitive to initialization and can converge to local optima. Common practices include: (i) initializing means with k-means, (ii) multiple random restarts, and (iii) covariance regularization (e.g., $\Sigma_m \leftarrow \Sigma_m + \lambda I$) to avoid singularities.

8 EM for Gaussian Mixture Models: pseudocode

Worked example — EM algorithm for learning a GMM (classical pseudocode)

Input: dataset $\mathcal{D} = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$, number of components M, convergence tolerance ε, max iterations $K_{\max}$.
Output: parameters $\theta = (\{\pi_m\}_{m=1}^M, \{\mu_m\}_{m=1}^M, \{\Sigma_m\}_{m=1}^M)$.

Initialize: Choose $\pi_m^{(0)} \geq 0$ with $\sum_{m=1}^M \pi_m^{(0)} = 1$, choose $\mu_m^{(0)} \in \mathbb{R}^d$, choose $\Sigma_m^{(0)} \succ 0$.
Set $k \leftarrow 0$ and compute initial log-likelihood

$$\ell(\theta^{(0)}) = \sum_{i=1}^N \log \left(\sum_{m=1}^M \pi_m^{(0)} \mathcal{N}(x_i \mid \mu_m^{(0)}, \Sigma_m^{(0)}) \right).$$

Repeat until convergence or $k = K_{\max} - 1$:

E-step (compute responsibilities):
For each $i = 1, \dots, N$ and each $m = 1, \dots, M$,

$$r_{im}^{(k)} \leftarrow \frac{\pi_m^{(k)} \mathcal{N}(x_i \mid \mu_m^{(k)}, \Sigma_m^{(k)})}{\sum_{m'=1}^M \pi_{m'}^{(k)} \mathcal{N}(x_i \mid \mu_{m'}^{(k)}, \Sigma_{m'}^{(k)})}.$$

M-step (update parameters):
Compute soft counts for each component:

$$N_m^{(k)} \leftarrow \sum_{i=1}^N r_{im}^{(k)} \qquad \text{for } m = 1, \dots, M.$$

Update mixing weights:

$$\pi_m^{(k+1)} \leftarrow \frac{N_m^{(k)}}{N}.$$

Update means:

$$\mu_m^{(k+1)} \leftarrow \frac{1}{N_m^{(k)}} \sum_{i=1}^N r_{im}^{(k)} \, x_i.$$

Update covariances:

$$\Sigma_m^{(k+1)} \leftarrow \frac{1}{N_m^{(k)}} \sum_{i=1}^N r_{im}^{(k)} \, (x_i - \mu_m^{(k+1)})(x_i - \mu_m^{(k+1)})^\top.$$

(Optionally, regularize: $\Sigma_m^{(k+1)} \leftarrow \Sigma_m^{(k+1)} + \lambda I$ with small $\lambda > 0$.)

Convergence check:
Compute the new log-likelihood

$$\ell(\theta^{(k+1)}) = \sum_{i=1}^N \log \left(\sum_{m=1}^M \pi_m^{(k+1)} \mathcal{N}(x_i \mid \mu_m^{(k+1)}, \Sigma_m^{(k+1)}) \right).$$

If $\left| \ell(\theta^{(k+1)}) - \ell(\theta^{(k)}) \right| \leq \varepsilon$, **stop**.
Else set $k \leftarrow k + 1$ and continue.

Return $\theta^{(k+1)}$.

Implementation notes. In practice, it is numerically safer to compute responsibilities using log-densities and the log-sum-exp trick, especially in high dimensions. Multiple random restarts (or k-means initialization) are also commonly used to mitigate convergence to poor local optima.

9 When the E-step is not tractable

The classical EM template requires computing $p(\mathcal{Z} \mid \mathcal{D}, \theta^{(k)})$ (or at least the expectations it implies). In many models this posterior cannot be computed exactly. Common strategies then include:

- **Sampling-based EM:** approximate E-step expectations with samples.
- **Variational EM:** use a simplified family of approximate posteriors to keep computations tractable.

These methods preserve the central EM idea—alternating inference about hidden variables with parameter updates—while relaxing the exactness of the E-step.