

EM Theory and Latent-Variable Models

Lecture Notes Consolidated with Claude AI — CS6140 Machine Learning

*Optional companion to the **EM_MIXTURES** note. This note proves what that one stated: why the ELBO is a bound, why the E step makes it tight, and why the log-likelihood cannot decrease. It then follows the latent variable in two directions — giving it structure (Bayes nets, HMMs, topic models) and making it continuous (variational inference, the VAE). Notation is identical to **EM_MIXTURES**.*

1 Why EM Works

The basic note stated three things without proof: that the ELBO is a lower bound, that the E step makes it tight, and that the log-likelihood never decreases. All three come from one inequality and one identity, in that order.

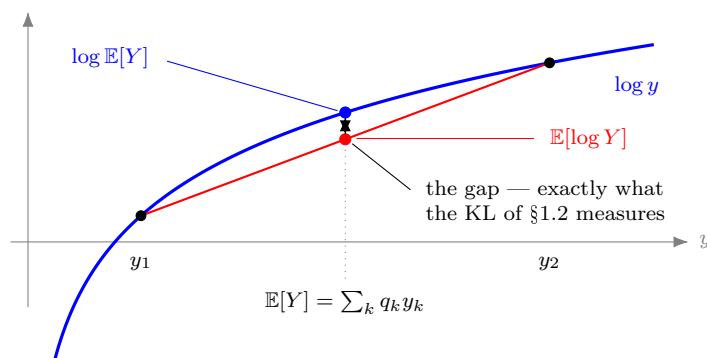
1.1 Why There Is a Bound at All: Jensen

The whole theory rests on one inequality. A function f is **concave** if its chord never lies above its graph; log is the example we need. **Jensen's inequality** says that for a concave f and any random variable Y ,

$$f(\mathbb{E}[Y]) \geq \mathbb{E}[f(Y)]. \quad (1)$$

In words: *the function of the average is at least the average of the function*. For log and a finite set of weights $q_k \geq 0$ summing to 1,

$$\log \left(\sum_k q_k y_k \right) \geq \sum_k q_k \log y_k. \quad (2)$$



Jensen for log. Averaging two points and *then* taking the log lands on the curve; taking logs first and averaging lands on the chord, which is lower. Equality holds exactly when the two points coincide — i.e. when Y is constant.

Key point: Equation 2 is the formal version of the move made twice in **EM_MIXTURES**: trading a *log of a sum* (intractable) for a *weighted sum of logs* (separable). The price is that you get a lower bound instead of the thing itself. The art is choosing the weights q_k so the price is zero.

Deriving the ELBO. Start from the observed-data log-likelihood of one datapoint and insert an arbitrary distribution $q(z)$ with $q(z) > 0$ wherever $p(\mathbf{x}, z) > 0$:

$$\begin{aligned}\log p(\mathbf{x}) &= \log \sum_z p(\mathbf{x}, z) = \log \sum_z q(z) \frac{p(\mathbf{x}, z)}{q(z)} = \log \mathbb{E}_q \left[\frac{p(\mathbf{x}, z)}{q(z)} \right] \\ &\geq \mathbb{E}_q \left[\log \frac{p(\mathbf{x}, z)}{q(z)} \right] \equiv \mathcal{L}(q, \theta).\end{aligned}\tag{3}$$

The multiply-and-divide by $q(z)$ is the entire trick: it turns the sum into an expectation, which is what Jensen needs. $\mathcal{L}(q, \theta)$ is the **ELBO**. Splitting the log two ways gives the two forms used in practice:

$$\mathcal{L}(q, \theta) = \mathbb{E}_q[\log p(\mathbf{x}, z)] + H(q) \quad (\text{the } \textit{ELL plus entropy} \text{ form}) \tag{4}$$

$$= \mathbb{E}_q[\log p(\mathbf{x} | z)] - \text{KL}(q(z) \| p(z)) \quad (\text{the } \textit{reconstruction minus KL} \text{ form}) \tag{5}$$

where $H(q) = -\sum_z q(z) \log q(z)$. Equation 5 is the form quoted in the homework and in the VAE note; Eq. 4 is the one that makes the M step obvious, since $H(q)$ does not involve θ at all — maximizing $\mathcal{L}$ over θ is maximizing the expected complete-data log-likelihood $Q(\theta)$.

1.2 The KL Decomposition, and Why the E Step Makes the Bound Tight

Jensen tells us a gap exists. This identity tells us exactly what it is.

Take Eq. 3's expectation apart using $p(\mathbf{x}, z) = p(z | \mathbf{x})p(\mathbf{x})$:

$$\begin{aligned}\mathcal{L}(q, \theta) &= \mathbb{E}_q \left[\log \frac{p(z | \mathbf{x}) p(\mathbf{x})}{q(z)} \right] = \log p(\mathbf{x}) + \mathbb{E}_q \left[\log \frac{p(z | \mathbf{x})}{q(z)} \right] \\ &= \log p(\mathbf{x}) - \text{KL}(q(z) \| p(z | \mathbf{x})),\end{aligned}\tag{6}$$

so, rearranged,

$$\boxed{\log p(\mathbf{x}) = \mathcal{L}(q, \theta) + \text{KL}(q(z) \| p(z | \mathbf{x}, \theta))} \tag{7}$$

for *every* q . This is the central identity of EM, and it says three things at once.

1. **The bound is a bound.** $\text{KL} \geq 0$ always (Gibbs' inequality), so $\log p(\mathbf{x}) \geq \mathcal{L}(q, \theta)$ — Jensen again, now with the gap named.
2. **The gap is exactly a KL.** It measures how far the guess q is from the true posterior $p(z | \mathbf{x}, \theta)$. Nothing else.
3. **The gap can be closed.** $\text{KL}(q \| p) = 0$ if and only if $q = p$. So choosing

$$q^*(z) = p(z | \mathbf{x}, \theta^{\text{old}}) \tag{8}$$

makes the bound *touch*: $\mathcal{L}(q^*, \theta^{\text{old}}) = \log p(\mathbf{x} | \theta^{\text{old}})$.

Key point: Equation 8 is the E step, and now it is not a recipe but a *derivation*: the posterior is not one reasonable choice of q among many, it is the unique choice that makes the bound exact. For a Gaussian mixture, $p(z_{ik}=1 | \mathbf{x}_i, \theta)$ is the responsibility $\langle z_{ik} \rangle$ — which is why **EM_MIXTURES** could compute the E step with Bayes' rule and stop. Everything else in this note follows from asking what happens when that posterior is *not* available in closed form (§6.2).

1.3 The Monotonicity Theorem

Theorem. Let $\theta^{(t+1)} = \arg \max_{\theta} Q(\theta \mid \theta^{(t)})$ where $Q(\theta \mid \theta^{(t)}) = \mathbb{E}_{p(z \mid X, \theta^{(t)})}[\log p(X, z \mid \theta)]$. Then

$$\ell(\theta^{(t+1)}) \geq \ell(\theta^{(t)}), \quad \ell(\theta) = \log p(X \mid \theta),$$

with equality only if $\theta^{(t)}$ is a stationary point of ℓ . The M step need not maximize Q ; it suffices that $Q(\theta^{(t+1)} \mid \theta^{(t)}) \geq Q(\theta^{(t)} \mid \theta^{(t)})$ — this weaker version is called **generalized EM**.

Proof. Write $q^{(t)}(z) = p(z \mid X, \theta^{(t)})$, the E-step's distribution. Apply Eq. 7 twice.

Step 1 — the bound is tight at $\theta^{(t)}$. By construction $\text{KL}(q^{(t)} \parallel p(z \mid X, \theta^{(t)})) = 0$, so

$$\ell(\theta^{(t)}) = \mathcal{L}(q^{(t)}, \theta^{(t)}). \quad (9)$$

Step 2 — the M step raises the bound. By Eq. 4, $\mathcal{L}(q^{(t)}, \theta) = Q(\theta \mid \theta^{(t)}) + H(q^{(t)})$, and $H(q^{(t)})$ does not depend on θ . So maximizing Q over θ is maximizing $\mathcal{L}(q^{(t)}, \cdot)$, and since $\theta^{(t+1)}$ is the maximizer (or merely an improvement, for generalized EM),

$$\mathcal{L}(q^{(t)}, \theta^{(t+1)}) \geq \mathcal{L}(q^{(t)}, \theta^{(t)}). \quad (10)$$

Step 3 — the bound still bounds. Apply Eq. 7 at the *new* parameters with the *old* q :

$$\ell(\theta^{(t+1)}) = \mathcal{L}(q^{(t)}, \theta^{(t+1)}) + \underbrace{\text{KL}(q^{(t)} \parallel p(z \mid X, \theta^{(t+1)}))}_{\geq 0} \geq \mathcal{L}(q^{(t)}, \theta^{(t+1)}). \quad (11)$$

Chaining Eqs. 11, 10, 9:

$$\ell(\theta^{(t+1)}) \stackrel{(11)}{\geq} \mathcal{L}(q^{(t)}, \theta^{(t+1)}) \stackrel{(10)}{\geq} \mathcal{L}(q^{(t)}, \theta^{(t)}) \stackrel{(9)}{=} \ell(\theta^{(t)}). \quad \blacksquare$$

Key point: Read the three steps as the three parts of the staircase picture in `EM_MIXTURES` §6.1. Step 1 is the bound touching; step 2 is climbing the bound; step 3 is the observation that the true curve is still above the bound after the climb. The likelihood gains *at least* what the bound gained, and it gains strictly more by the KL term in Eq. 11 — which is the slack the next E step will collect.

Remark. Three consequences the basic note asserted without proof, now visible in the algebra. (i) The theorem never mentions Gaussians — it holds for any component family, which is why `EM_MIXTURES` §7 could swap in Binomials and Poissons unchanged. (ii) Generalized EM is enough. If the M step has no closed form you may take a single gradient step on Q and monotonicity survives; this is what makes EM usable for models where Eq. 4 cannot be maximized exactly. (iii) Monotone and bounded above implies $\ell(\theta^{(t)})$ converges. It does not imply $\theta^{(t)}$ converges, nor that the limit is a global maximum — only that it is a stationary point of ℓ under mild regularity (Wu, 1983). §2 is about the gap between those statements.

1.4 EM Is Coordinate Ascent on the ELBO

Equation 7 supports a reformulation that is worth internalizing, because it is the form that generalizes.

Regard $\mathcal{L}(q, \theta)$ as a single objective in *two* arguments: the distribution q and the parameters θ . Then

	optimize over	solution
E-step	q , with θ fixed	$q = p(z \mid X, \theta)$, closing the KL to zero
M-step	θ , with q fixed	$\theta = \arg \max_{\theta} \mathbb{E}_q[\log p(X, z \mid \theta)]$

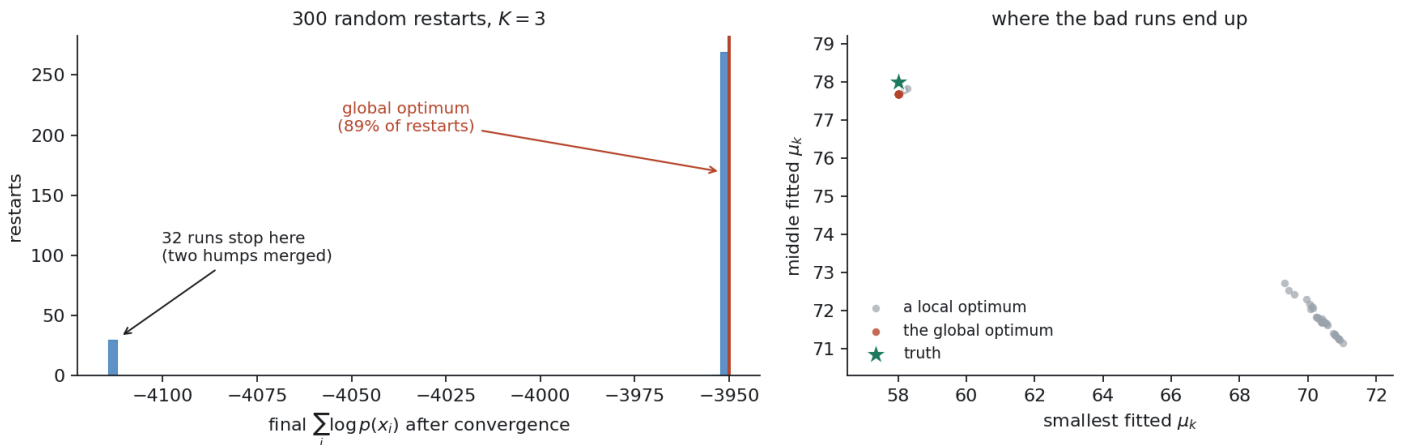
EM is coordinate ascent on $\mathcal{L}$. Both steps increase the same objective; neither ever decreases it; the procedure converges to a stationary point of $\mathcal{L}$, which by Eq. 7 is a stationary point of ℓ .

How this maps onto the board’s route. In lecture the derivation goes: write the complete-data likelihood with the indicator exponent Π_{ik} ; observe it is too hard to optimize because Π_{ik} is unknown; replace Π_{ik} by its expectation $\langle \Pi_{ik} \rangle$ to get the *expected log-likelihood* (ELL); maximize that. The present section says the same thing in a different order. The ELL is $Q(\theta \mid \theta^{\text{old}})$; by Eq. 4 it differs from $\mathcal{L}$ by the θ -free constant $H(q)$; and the substitution $\Pi_{ik} \rightsquigarrow \langle \Pi_{ik} \rangle$ is legitimate precisely because $\log p(X, z \mid \theta)$ is *linear* in the indicators, so $\mathbb{E}[\cdot]$ passes through. Nothing is approximated at that step. If the complete-data log-likelihood were *not* linear in z — and for exponential-family components it always is — plugging in $\mathbb{E}[z]$ would be an approximation rather than an identity, and the theorem of §1.3 would not apply.

Key point: Two sentences worth remembering. “Plug in $\mathbb{E}[z]$ ” is correct because of linearity, not because expectations are generally substitutable — $\mathbb{E}[f(z)] \neq f(\mathbb{E}[z])$ in general, which is Jensen again. And EM is coordinate ascent, so everything you know about coordinate ascent transfers: it can be slow near ill-conditioned optima, and it stops at stationary points of whatever surface it is climbing.

2 Local Optima, MAP-EM, and Regularized Covariance

How bad are the local optima, really? The honest answer is “it depends entirely on separation,” and it is worth measuring rather than guessing. On `2gaussian.txt` and `3gaussian.txt` the components are far enough apart that *every* random restart finds the same optimum — there is effectively nothing to worry about. On the deliberately overlapping three-hump data of `EM_MIXTURES` §1.1, there is:



300 random starts, $K = 3$. **Left:** the final log-likelihoods take only a few distinct values — 89% land on the best one, 11% on a solution about 164 nats worse. **Right:** the bad runs are not scattered noise, they are a *specific* failure — one component covering the two left humps at $\mu \approx 70$ while the other two split the right hump. EM converged there monotonically and correctly; it is simply a different stationary point.

Key point: Local optima in mixtures are usually a small number of qualitatively distinct, interpretable solutions, not a rough landscape. That is good news: a handful of restarts is genuinely enough, and when two restarts disagree you can usually look at the two answers and see what happened. Restarts, not cleverer optimization, is the standard answer.

The unbounded-likelihood degeneracy. Worse than a local optimum: the GMM likelihood is *unbounded above*. Put $\mu_k = \mathbf{x}_i$ for a single point and shrink $\Sigma_k \rightarrow 0$; the density at $\mathbf{x}_i$ diverges and so does ℓ . The global maximum of the likelihood is therefore a spike with no statistical content, and the useful answer is always a *local* maximum. This is not a flaw in EM — it is a flaw in maximum likelihood for this model.

MAP-EM is the fix. Put a prior on the parameters and maximize the posterior instead: replace $Q(\theta)$ by $Q(\theta) + \log p(\theta)$ in the M step. Everything in §1.3 goes through with ℓ replaced by $\log p(X | \theta) + \log p(\theta)$, so monotonicity is preserved. With an inverse-Wishart prior on each Σ_k the M step becomes

$$\Sigma_k = \frac{\sum_i \langle z_{ik} \rangle (\mathbf{x}_i - \mu_k)^T (\mathbf{x}_i - \mu_k) + \Psi}{N_k + \nu}, \quad (12)$$

for prior parameters Ψ (a matrix) and ν (a scalar count). Compare this with the $\Sigma_k + \epsilon I$ patch that `EM_MIXTURES` §5 recommends for numerical reasons: with $\Psi = \epsilon(N_k + \nu)I$ they coincide. The ad-hoc regularizer *is* a prior, exactly as additive smoothing is a Beta prior in `GDA_NB_ADVANCED` §7. Choosing Ψ, ν rather than ϵ has the advantage that the degeneracy is provably gone: the posterior is bounded.

3 k -Means as a Limit, and Choosing K

k -means is zero-temperature EM. Fix every $\Sigma_k = \sigma^2 I$ and every $\pi_k = 1/K$, and let $\sigma^2 \rightarrow 0$. The responsibility

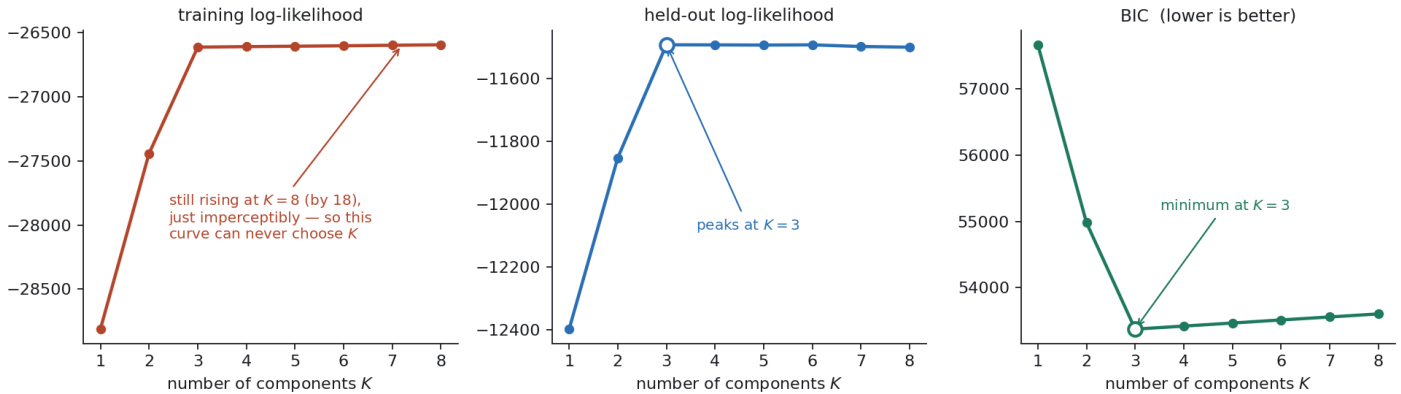
$$\langle z_{ik} \rangle = \frac{\exp(-\|\mathbf{x}_i - \mu_k\|^2 / 2\sigma^2)}{\sum_\ell \exp(-\|\mathbf{x}_i - \mu_\ell\|^2 / 2\sigma^2)}$$

is a softmax over $-\|\mathbf{x}_i - \mu_k\|^2$ with temperature σ^2 . As the temperature goes to zero the softmax becomes an arg max: $\langle z_{ik} \rangle \rightarrow 1$ for the nearest centre and 0 otherwise. The M step then averages over hard assignments. That is k -means exactly, and it explains three of its properties at once: it converges (it is EM), it hill-climbs a likelihood (the $\sigma^2 \rightarrow 0$ limit of the GMM one), and it is bad at overlapping clusters (it has thrown away the very information that EM keeps).

Choosing K — the likelihood cannot. Add a component and the maximized likelihood can only go up: the K -component family contains every $(K-1)$ -component solution (set $\pi_K = 0$). So training log-likelihood is monotone in K and says nothing. Two standard answers:

- **Held-out log-likelihood.** Fit on a training split, evaluate $\sum_i \log p(\mathbf{x}_i)$ on data the fit never saw. This is cross validation with likelihood instead of accuracy as the score, and it is the method with the fewest assumptions.

- **BIC** (Bayesian information criterion): $\text{BIC} = -2\ell + P \log N$, where P is the number of free parameters — for a full-covariance GMM, $P = (K - 1) + KD + K \frac{D(D+1)}{2}$. Pick the K minimizing BIC. The $P \log N$ term is a penalty for complexity that comes out of a Laplace approximation to the marginal likelihood $p(X | K)$; see `BAYES_REGRESSION` for the same quantity (the “evidence”) computed exactly in a case where it can be.



`3gaussian.txt`, $K = 1, \dots, 8$, best of twelve restarts each, 70/30 split. **Left:** training log-likelihood rises from $K = 1$ to 3 and then keeps creeping up forever — it has no opinion about K . **Middle:** held-out log-likelihood peaks at $K = 3$ and declines. **Right:** BIC bottoms out at $K = 3$. Both criteria recover the number of components the data was actually generated with.

Remark. BIC is cheap (one fit per K) and held-out likelihood is honest; when they disagree, trust the held-out number, since BIC’s derivation assumes a regularity condition — that the maximum likelihood estimate is a well-separated interior point — that mixtures violate (label switching alone means the optimum is never unique). On well-separated data such as this they agree, which is the usual outcome.

4 EM’s Original Problem: Missing Data

Dempster, Laird and Rubin’s 1977 paper that named EM was not about clustering. It was about *incomplete data*, and the mixture case is one instance: the missing quantity is the component label.

Suppose each row $\mathbf{x}_i$ is split into an observed part $\mathbf{x}_i^{\text{obs}}$ and a missing part $\mathbf{x}_i^{\text{mis}}$ — a survey where respondents skipped different questions, a sensor array where different sensors failed. Treat $\mathbf{x}^{\text{mis}}$ as the latent variable and the recipe is unchanged:

- **E step:** compute the conditional distribution of each missing entry given the observed entries of that row and the current parameters. For a Gaussian model this is the conditional Gaussian, in closed form: if $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$ is partitioned into blocks, then $\mathbf{x}^{\text{mis}} | \mathbf{x}^{\text{obs}}$ is Gaussian with mean $\mu_m + \Sigma_{mo} \Sigma_{oo}^{-1} (\mathbf{x}^{\text{obs}} - \mu_o)^T$ and covariance $\Sigma_{mm} - \Sigma_{mo} \Sigma_{oo}^{-1} \Sigma_{om}$.
- **M step:** the usual mean and covariance updates, with each missing entry replaced by its conditional expectation — plus, for the covariance, the conditional *variance*, which the naive “fill in the mean and refit” procedure forgets and which is why that procedure underestimates the spread.

Key point: Contrast this with the `GDA_NB` note’s treatment of missing values, where marginalizing a missing feature out of a Naive Bayes product was free because independence made the missing factor simply drop. Here the features are correlated, so a missing entry must be *imputed from* the observed ones. EM does the imputation and the fitting jointly, which is the right thing: imputing once with a wrong model and then fitting is not the same procedure.

5 Giving the Latent Variable Structure: Bayes Nets and HMMs

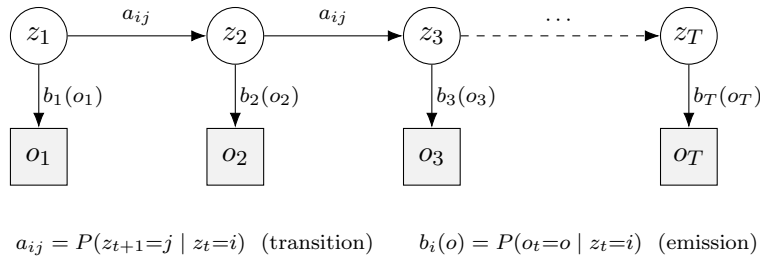
GDA_NB §1.2 listed three ways to make the joint $p(\mathbf{x})$ tractable: assume full independence (Naive Bayes), restrict the shape (GDA, mixtures), or group the features into small dependent clusters. The third option is where this section starts.

“Not-so-naive Bayes” → Bayes networks. Naive Bayes factorizes $p(\mathbf{x} | y) = \prod_j p(x^j | y)$ — every feature its own island. Relax it by allowing each feature one parent besides the class ($p(x^j | y, x^{\text{pa}(j)})$) and you get the *tree-augmented Naive Bayes* classifier; allow an arbitrary acyclic parent structure and you get a general **Bayes network**, a directed acyclic graph in which

$$p(x^1, \dots, x^D) = \prod_{j=1}^D p(x^j | x^{\text{pa}(j)}). \quad (13)$$

The graph is the modelling choice; Eq. 13 is what it buys — a joint over D variables specified by D small conditional tables instead of one table of size V^D . (Recall the 10^{100} -cell argument of GDA_NB §1.2: this is the third escape from it.) When some variables in the network are never observed, fitting it is again EM — the E step is an inference query on the graph.

HMMs: one chain of latent variables. The most useful structured case puts the latent variables in a sequence. A **hidden Markov model** has N hidden states; at each time t a hidden state z_t emits an observation o_t and transitions to z_{t+1} :



A mixture model is the $T = 1$ case: one latent variable, no transitions. Everything in EM_MIXTURES carries over with $\langle z_{ik} \rangle$ replaced by a posterior over whole *paths*, and the only new difficulty is that there are N^T paths — so the E step needs an algorithm rather than a single application of Bayes’ rule.

Forward and backward probabilities. Define

$$\alpha_i(t) = P(o_1, \dots, o_t, z_t = i), \quad \beta_i(t) = P(o_{t+1}, \dots, o_T | z_t = i). \quad (14)$$

Read them carefully: α is a *joint* over the past and the current state; β is a *conditional* on the current state, over the future. Both satisfy recursions obtained by conditioning on the neighbouring state:

$$\alpha_j(t+1) = \left[\sum_{i=1}^N \alpha_i(t) a_{ij} \right] b_j(o_{t+1}), \quad \beta_i(t) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_j(t+1), \quad (15)$$

initialized by $\alpha_i(1) = \pi_i b_i(o_1)$ and $\beta_i(T) = 1$. Each costs $O(N^2 T)$ instead of $O(N^T)$.

Part (A) of the homework’s optional HMM problem asks you to notice that the standard sequence probability $P(O) = \sum_i \alpha_i(T)$ has a mirror image: running the β recursion from the end and combining with the initial distribution gives $P(O) = \sum_i \pi_i b_i(o_1) \beta_i(1)$. The substitution is mechanical — reverse the roles of “past” and “future” in Eq. 14.

Part (B) is the one worth doing carefully. For any t ,

$$P(O) = \sum_{i=1}^N \alpha_i(t) \beta_i(t). \quad (16)$$

Proof: condition on the state at time t . $P(O) = \sum_i P(O, z_t = i)$, and the Markov property makes the past and future conditionally independent given $z_t = i$:

$$P(O, z_t = i) = \underbrace{P(o_1..o_t, z_t = i)}_{\alpha_i(t)} \cdot \underbrace{P(o_{t+1}..o_T \mid z_t = i)}_{\beta_i(t)}.$$

That conditional independence is the *entire* content of “Markov”, and it is the reason the $O(N^2T)$ algorithm exists at all. Note that the index i in Eq. 16 runs over the N hidden states, not over time — the homework statement flags this, because the original DHS printing reuses the symbol T' for both.

Part (C): at $t = T$, $\beta_i(T) = 1$ by definition and Eq. 16 reduces to $P(O) = \sum_i \alpha_i(T)$, the forward algorithm. At $t = 0$ (or $t = 1$ with the convention $\alpha_i(1) = \pi_i b_i(o_1)$) it reduces to the backward form of part (A). The formula interpolates between the two standard algorithms, and the fact that it gives the same answer at every t is a useful unit test for an implementation.

Baum–Welch is EM. With α and β in hand, the E-step quantities are

$$\gamma_i(t) = \frac{\alpha_i(t) \beta_i(t)}{P(O)}, \quad \xi_{ij}(t) = \frac{\alpha_i(t) a_{ij} b_j(o_{t+1}) \beta_j(t+1)}{P(O)}, \quad (17)$$

$\gamma_i(t)$ being the posterior probability of being in state i at time t — the exact analogue of $\langle z_{ik} \rangle$ — and $\xi_{ij}(t)$ the posterior over transitions. The M step is then the usual weighted counting:

$$\hat{a}_{ij} = \frac{\sum_t \xi_{ij}(t)}{\sum_t \gamma_i(t)}, \quad \hat{b}_i(o) = \frac{\sum_{t: o_t=o} \gamma_i(t)}{\sum_t \gamma_i(t)}, \quad \hat{\pi}_i = \gamma_i(1).$$

This is the **Baum–Welch** algorithm, and it is EM with a structured E step. Every guarantee of §1.3 applies unchanged.

Key point: The pattern generalizes past HMMs. The E step is always “compute the posterior over the latent variables”; the M step is always “weighted maximum likelihood with those posteriors as weights”. What changes between models is only *how hard the posterior is to compute*: trivial for a mixture (K terms), a dynamic program for a chain ($O(N^2T)$), message passing for a tree, intractable for a general graph — and at that last point you stop computing it exactly and start approximating, which is §6.2.

Remark. **Viterbi** is the other dynamic program on the same trellis: replace the $\sum$ in Eq. 15’s α recursion by a max and keep backpointers, and you get the single most likely *path* rather than the probability of the observations. The relationship between Viterbi and forward–backward is exactly the relationship between k -means and EM (§3): hard assignment versus soft. Training an HMM with Viterbi assignments instead of γ, ξ is called “Viterbi training” and is the HMM world’s k -means — faster, and not maximizing the likelihood.

6 When the Posterior Is Intractable

Every E step so far has been exact, because the latent variable had finitely many values or a chain structure. This section follows what happens when it does not — first in a model where the posterior merely stops factorizing, then in one where it has no closed form at all.

6.1 Topic Models: Mixtures With a Prior on the Weights

One more direction, because it shows what happens when the mixture weights themselves become latent.

In **latent Dirichlet allocation** (Blei, Ng & Jordan, 2003) a document is a mixture over K topics, each topic is a distribution over the vocabulary, and crucially *each document gets its own mixture weights* $\pi^{(d)}$, drawn from a Dirichlet prior:

$$\pi^{(d)} \sim \text{Dir}(\alpha), \quad z_{dn} \sim \text{Categorical}(\pi^{(d)}), \quad w_{dn} \sim \text{Categorical}(\phi_{z_{dn}}).$$

Compare this with **EM MIXTURES** Eq. (1): the generative recipe is identical — pick a component, draw from it — but the die is re-weighted per document, and the $\pi^{(d)}$ are latent rather than parameters. The Dirichlet prior is the same object as in **GDA_NB_ADVANCED** §7, used here to make the per-document weights well behaved when a document is short.

The consequence is that the posterior $p(\pi^{(d)}, z^{(d)} \mid w^{(d)})$ no longer factorizes and has no closed form: there is no Bayes'-rule E step. LDA is fitted with variational EM or with collapsed Gibbs sampling. This is the first model in this sequence where the E step genuinely breaks, which is the natural lead-in to the last section.

6.2 Variational Inference, and the VAE

The one assumption we have been making. Every E step so far computed $q(z) = p(z \mid \mathbf{x}, \theta)$ exactly. That was possible because z was discrete with few values (mixtures), or discrete with chain structure (HMMs). Make z a continuous vector in $\mathbb{R}^d$ and let $p(\mathbf{x} \mid z)$ be a neural network, and

$$p(z \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid z)p(z)}{\int p(\mathbf{x} \mid z)p(z) dz}$$

has an integral in the denominator that no one can do. The E step is dead.

Variational inference: give up on exactness, keep the bound. Equation 7 holds for *every* q , not just the posterior. So restrict q to a family $\mathcal{Q}$ you can handle and maximize $\mathcal{L}(q, \theta)$ over $q \in \mathcal{Q}$. You no longer close the KL gap completely — you close as much of it as $\mathcal{Q}$ permits — but you still have a valid lower bound and you can still alternate. *Variational EM* is exactly EM with an approximate E step.

The VAE. The **variational autoencoder** (Kingma & Welling, 2013) makes two specific choices:

1. $\mathcal{Q}$ is the set of diagonal Gaussians, $q_\phi(z \mid \mathbf{x}) = \mathcal{N}(\mu_\phi(\mathbf{x}), \text{diag } \sigma_\phi^2(\mathbf{x}))$, with μ_ϕ and σ_ϕ output by an **encoder** network. Instead of solving a separate optimization for each datapoint's q , one network maps $\mathbf{x} \mapsto q(z \mid \mathbf{x})$ for all $\mathbf{x}$ at once — *amortized* inference.
2. Both ϕ (encoder) and θ (decoder) are trained *jointly* by gradient ascent on $\mathcal{L}$, rather than alternating. The E step and M step become one gradient step on a shared objective.

The objective is Eq. 5 verbatim:

$$\mathcal{L}(\phi, \theta; \mathbf{x}) = \underbrace{\mathbb{E}_{q_\phi(z|\mathbf{x})} [\log p_\theta(\mathbf{x} | z)]}_{\text{reconstruction}} - \underbrace{\text{KL}(q_\phi(z | \mathbf{x}) \| p(z))}_{\text{keep } q \text{ near the prior}}, \quad p(z) = \mathcal{N}(0, I). \quad (18)$$

	Gaussian mixture	VAE
latent z	one of K labels	a vector in $\mathbb{R}^d$
prior $p(z)$	$(\pi_1, \dots, \pi_K)$, <i>learned</i>	$\mathcal{N}(0, I)$, <i>fixed</i>
posterior $q(z \mathbf{x})$	$(\langle z_{i1} \rangle, \dots, \langle z_{iK} \rangle)$, <i>computed exactly</i> by Bayes' rule	$\mathcal{N}(\mu_\phi(\mathbf{x}), \sigma_\phi^2(\mathbf{x}))$, <i>learned</i> by an encoder
decoder $p(\mathbf{x} z)$	$\mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k)$	a neural network
objective	the ELBO	the same ELBO
optimization	alternate E and M, closed form	joint SGD on (ϕ, θ)
gap $\text{KL}(q \ p(z \mathbf{x}))$	exactly 0 after each E step	> 0 ; the price of amortizing

Key point: HW5 Problem 3(C) asks you to explain this table in prose, continuing from HW4 Problem 3(C). The one-sentence version: *the objective is identical; what changes is that the posterior can no longer be computed, so it is approximated by a network trained alongside the decoder.* The second sentence worth having: in the mixture the E step was free and exact, so the bound was tight at every iteration and the algorithm provably climbed the true likelihood — the VAE gives that up and optimizes a bound that stays strictly below $\log p(\mathbf{x})$, which is why a VAE's reported “loss” is a bound and not a likelihood.

Remark. One mechanical detail the VAE note covers and this one only flags: Eq. 18 requires a gradient through a *sample* from q_ϕ , and sampling is not differentiable. The **reparameterization trick** — write $z = \mu_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x}) \odot \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I)$, so the randomness is in ε and the gradient flows through μ_ϕ, σ_ϕ — is what makes the whole thing trainable. There is no analogue in EM, because EM never needed to differentiate through the E step.

7 Further Reading

- **Dempster, Laird & Rubin**, “Maximum Likelihood from Incomplete Data via the EM Algorithm” (JRSS-B 1977) — the paper that named and unified the method; §4 is its actual subject.
- **Wu**, “On the Convergence Properties of the EM Algorithm” (Ann. Statist. 1983) — the careful statement of what §1.3's theorem does and does not give you.
- **Neal & Hinton**, “A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants” (1998) — the source of §1.4's coordinate-ascent reading, and the bridge from EM to variational methods.
- **Rabiner**, “A Tutorial on Hidden Markov Models” (Proc. IEEE 1989) — still the clearest account of §5, including the scaling tricks real implementations need.
- **Duda, Hart & Stork** §3.10 and pp. 154–155 — the source of §5's forward–backward problem.

- **Blei, Ng & Jordan**, “Latent Dirichlet Allocation” (JMLR 2003); **Blei, Kucukelbir & McAuliffe**, “Variational Inference: A Review for Statisticians” (2017) for §6.2.
- **Kingma & Welling**, “Auto-Encoding Variational Bayes” (2013) — the VAE.
- **Bishop**, *PRML* Ch. 9 (mixtures and EM, including the proofs here) and Ch. 10 (variational inference); **Murphy**, *PML* Ch. 8 and 21.
- **Companion notes in this course:** `EM_MIXTURES`, the required material; then `GDA_NB` and its advanced companion, the generative-classifier half of these two weeks; `BAYES_REGRESSION`, where the marginal likelihood (“evidence”) is computed exactly; and the **VAE** note of Week 10.