

Gaussian mixtures: complexity and model selection

Hicham Janati

Fall 2026

Chapter 1 closed on its own limit: one Gaussian describes one cluster (one center, one ellipsoid), and many datasets gather around *several* centers at once. A marketing team suspects a few distinct “types” among thousands of unlabeled customers; a biologist looks for unknown cell subtypes in gene-expression data. Discovering groups in unlabeled data is *clustering*, our first *unsupervised* problem: no labels exist, and no “right answers”, only structure to be found. This chapter builds the family that models such data, several Gaussians at once: the Gaussian mixture. Its likelihood cannot be maximized directly as in chapter 1, and the algorithm that maximizes it, EM, is one of the most used algorithms in statistics.

1 From one Gaussian to several groups

Two groups, one histogram. Begin in one dimension, with no units attached. Figure 1 shows the histogram of a sample of 400 values: two clearly separated bumps of equal size. No single Gaussian density can produce this shape (a $\mathcal{N}(\mu, \sigma^2)$ has one peak), yet each bump, taken alone, looks perfectly Gaussian. Suppose for a moment we *knew* which points belong to which bump. Then nothing new would be needed: split the sample in two and fit one Gaussian per group by chapter 1’s maximum likelihood: a mean and a variance ($\hat{\mu}_1, \hat{\sigma}_1^2$) for the left group, another pair for the right. The two fits also combine into a single model for a point whose group we ignore. The split being half-and-half, a random point is produced by a fair coin flip between the two Gaussians, and its density is the corresponding *superposition*, half of each bump:

$$f(x) = \frac{1}{2} \varphi(x; \mu_1, \sigma_1^2) + \frac{1}{2} \varphi(x; \mu_2, \sigma_2^2), \quad (1)$$

where $\varphi(\cdot; \mu, \sigma^2)$ denotes the $\mathcal{N}(\mu, \sigma^2)$ density. This is the solid curve of figure 1: a two-peaked density, assembled from two one-peaked ones.

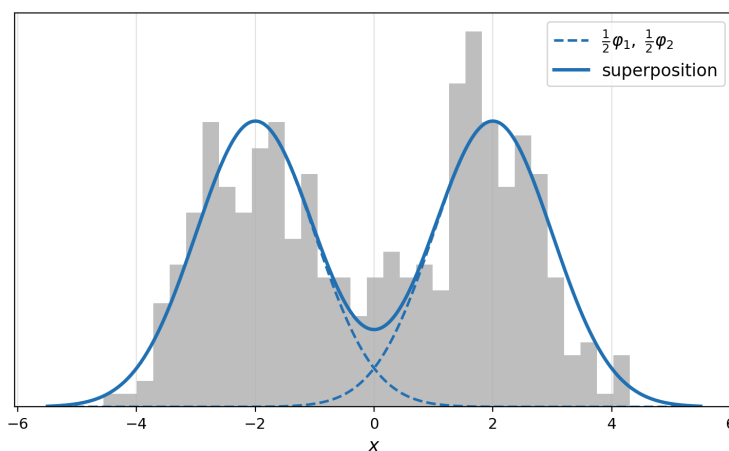


Figure 1: A sample with two equal groups (gray histogram), the two half-weighted Gaussian densities (dashed), and their superposition (solid): a two-peaked density that no single Gaussian can produce.

Remark

Equation (1) mixes *densities*, not random variables: it is not the distribution of the average $\frac{1}{2}(X_1 + X_2)$ of two independent Gaussian variables. That average is again Gaussian, $\mathcal{N}(\frac{\mu_1 + \mu_2}{2}, \frac{\sigma_1^2 + \sigma_2^2}{4})$, with a single peak between the two centers. Under the density (1), an observation is drawn from *one* of the two Gaussians, chosen at random, never from both at once.

A concrete example. Figure 2 shows the kind of data this machinery is for: a store’s loyalty-card database: one row per customer, monthly visits against monthly spend, exactly as the file presents it. The eye needs no help: there are clearly *three* groups of customers: a large cloud of occasional shoppers in the lower left (a few visits, small baskets); a group of regulars (around ten visits a month, medium spend); and a smaller group of bulk buyers, who come less often than the regulars but spend the most of anyone. And each group, taken alone, has exactly the shape chapter 1’s conclusion described: a single cluster, roughly symmetric around its center, well approximated by an ellipsoid: a Gaussian cluster.

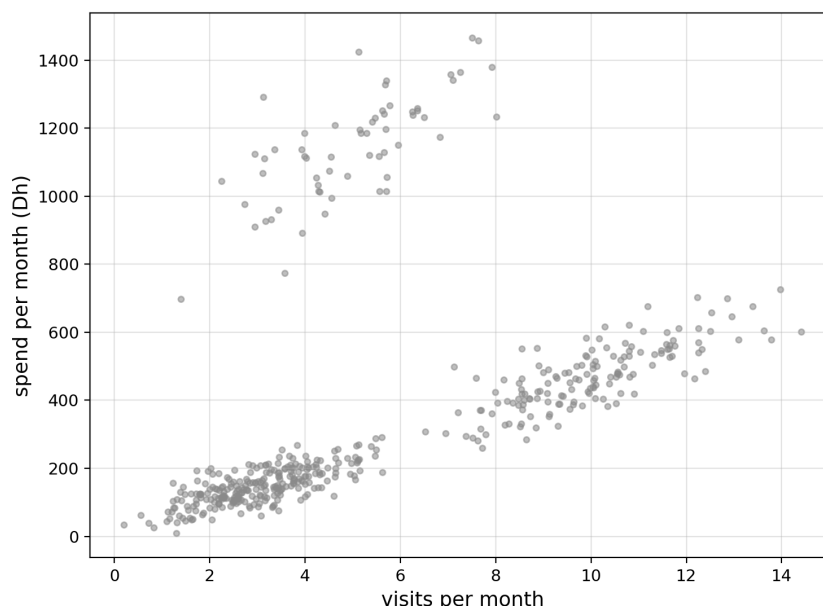


Figure 2: A store’s loyalty-card customers: monthly visits against monthly spend, as the file presents them, with no labels. Three groups of different sizes are plainly visible, and each one, taken alone, looks like a Gaussian cluster.

Determining these Gaussians is worth real money. The three means are three *customer profiles* (the typical occasional, regular and bulk customer) around which the store can build segments, offers and stock decisions; the three covariances say how variable each segment is; the group sizes say how much of the customer base each represents. And once the groups are determined, each customer can be *assigned*: which segment does this card belong to, and with what confidence? The natural model is the superposition idea of equation (1), upgraded to what the picture demands: several Gaussians on $\mathbb{R}^2$ rather than two on $\mathbb{R}$; weights that are *not* equal, since the three groups have visibly different sizes; and, above all, *no labels*: no column of the file says which customer belongs to which group, and nobody has ever annotated them. Can the parameters of the Gaussians, the proportions, and the memberships all be recovered from the gray cloud alone? The groups remain visible without any annotation, so the information survives in the shape of the data; the rest of the chapter turns this observation into an estimator.

2 Latent variables and the mixture model

Hidden labels as random variables. To reason about the missing labels, give them a name: let $Z \in \{1, \dots, K\}$ be the group of an observation, a genuine random variable, with $\mathbb{P}(Z = k) = \pi_k$ and $\mathbf{X} \mid Z = k \sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$, that the model contains but the data do not show: the variable Z is not observed in the data. Such a variable is called *latent*, and the model is a *latent variable model*. The process that generates one observation is assumed to be as follows:

1. *pick a group*: choose k with probability π_k ;
2. *draw the point*: $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$;
3. *forget the group*: record $\mathbf{X}$, discard k .

Step 3 is exactly what the customer file shows: the groups exist in the process that produced the data, but not in the empirical data.

Marginalizing the label. What distribution does the recipe produce for the $\mathbf{X}$ we actually observe? The law of total probability answers this. The events $\{Z = 1\}, \dots, \{Z = K\}$ partition the sample space, so for any region $A \subset \mathbb{R}^d$,

$$\begin{aligned} \mathbb{P}(\mathbf{X} \in A) &= \sum_{k=1}^K \mathbb{P}(Z = k) \mathbb{P}(\mathbf{X} \in A \mid Z = k) \\ &= \sum_{k=1}^K \pi_k \int_A \varphi(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k) d\mathbf{x} = \int_A \left(\sum_{k=1}^K \pi_k \varphi(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k) \right) d\mathbf{x}, \end{aligned}$$

where step 2 of the recipe supplies the conditional distribution $\mathbf{X} \mid Z = k \sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$. The probability of every region is obtained by integrating one fixed function, which is therefore the marginal density of $\mathbf{X}$:

$$f(\mathbf{x}) = \sum_{k=1}^K \pi_k \varphi(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k). \quad (2)$$

This is a *mixture* of Gaussians, a weighted superposition of bumps. For two groups of equal weight, equation (2) recovers the half-and-half superposition (1); the general formula frees both the number of groups K and their proportions π_k , which need not be equal (the customer cloud of figure 2, with its three groups of visibly different sizes, needs $K = 3$ and three different weights) and need not be known.

Definition 2.1 (Gaussian mixture model [1])

A random vector $\mathbf{X}$ on $\mathbb{R}^d$ follows a *Gaussian mixture distribution* with K components and parameters

$$\theta = (\pi_1, \dots, \pi_K, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K, \Sigma_1, \dots, \Sigma_K), \quad \pi_k > 0, \quad \sum_k \pi_k = 1,$$

when its density is the mixture (2). The *Gaussian mixture model* assumes that the observations $\mathbf{X}_1, \dots, \mathbf{X}_n$ are i.i.d. draws from such a distribution, with θ unknown.

Identifiability. Take $K = 2$ and any parameter vector $\theta = (\pi_1, \pi_2, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma_1, \Sigma_2)$, and let θ' be the same parameters with the two components numbered the other way, $\theta' = (\pi_2, \pi_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_1, \Sigma_2, \Sigma_1)$. Substituting θ' into (2),

$$f(\mathbf{x}; \theta') = \pi_2 \varphi(\mathbf{x}; \boldsymbol{\mu}_2, \Sigma_2) + \pi_1 \varphi(\mathbf{x}; \boldsymbol{\mu}_1, \Sigma_1) = f(\mathbf{x}; \theta),$$

because addition is commutative: $\theta \neq \theta'$ as soon as the two components differ, yet the two parameter vectors define exactly the same distribution. The map $\theta \mapsto f(\cdot; \theta)$ is therefore not one-to-one (the parameters are said to be *non-identifiable*), and the same argument works for any permutation of the K components. The practical consequence is that the *indexing* of the components is not reliable: nothing in the data makes one of them “component 1”, and two runs of an estimation procedure may return the same fitted mixture with the labels swapped. Section 8 examines what this does and does not break.

Membership probabilities. The model can also answer the assignment question of section 1: which group produced a given observation? Elementary conditional probability is enough. The joint weight of the event “group k and a point at $\mathbf{x}$ ” is $\pi_k \varphi(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k)$ (the probability of picking the group times the density it puts at $\mathbf{x}$), and the total weight of “a point at $\mathbf{x}$ ”, whatever the group, is their sum: the mixture density $f(\mathbf{x})$. Dividing the one by the other is Bayes’ rule, written first with its ingredients named and then expanded using the model:

$$\gamma_k(\mathbf{x}) = \mathbb{P}(Z = k \mid \mathbf{X} = \mathbf{x}) = \frac{\mathbb{P}(Z = k) f(\mathbf{x} \mid Z = k)}{f(\mathbf{x})} = \frac{\pi_k \varphi(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \varphi(\mathbf{x}; \boldsymbol{\mu}_j, \Sigma_j)}, \quad (3)$$

the denominator being the mixture density (2). These posterior probabilities are called *responsibilities*: a *soft* group assignment summing to one. A customer deep inside the occasional cloud has responsibility near 1 for that component; a customer sitting between two groups comes out near one half for each: the soft assignment records the ambiguity instead of hiding it. Clustering is now an inference problem: estimate θ , then read each customer’s group off its responsibilities.

3 The EM algorithm

A chicken-and-egg problem. Section 1 left us with two half-problems, each easy *on its own*. If the labels were known (as we briefly pretended for the two-bump sample), estimating the parameters would be chapter 1’s MLE applied group by group. If the parameters were known, computing each point’s membership probabilities would be the responsibilities formula of section 2. Each half is easy given the other, and neither is available. The EM algorithm [2] breaks this circle by *alternating*: start from a guess of the parameters, compute the memberships they imply, re-estimate the parameters treating those soft memberships as if they were data, and repeat.

Why direct maximization fails. The log-likelihood of a mixture,

$$\ell(\theta) = \sum_{i=1}^n \log \left(\sum_{k=1}^K \pi_k \varphi(\mathbf{x}_i; \boldsymbol{\mu}_k, \Sigma_k) \right), \quad (4)$$

has a *logarithm of a sum* where chapter 1’s Gaussian log-likelihood had a sum of logarithms. Each of those logarithms was an explicit function of $(\boldsymbol{\mu}, \Sigma)$, which is what let chapter 1 maximize the likelihood in closed form; with the sum trapped inside the logarithm this simplification no longer occurs, and setting gradients to zero yields coupled equations with no closed form.

Component indicators and the complete-data likelihood. To make the chicken-and-egg idea exact, encode the hidden label of observation i as *indicator variables*

$$z_{ik} = \mathbf{1}\{Z_i = k\} \quad (z_{ik} \in \{0, 1\}, \sum_k z_{ik} = 1) :$$

the one-hot vector $(z_{i1}, \dots, z_{iK})$ says which component produced point i . Suppose the indicators *were* observed alongside the data: the sample would then consist of the pairs $(\mathbf{x}_i, z_i)$, and its

likelihood, the *complete-data likelihood* L_c , is built from the *joint* density of $(\mathbf{X}, Z)$, evaluated at those pairs. This is a different object from the observed-data likelihood (4), which uses only the marginal density of $\mathbf{X}$. The joint density of one pair follows from the generative recipe of section 2 (pick the group, then draw the point), so for a pair with label $z_i = k$,

$$p(\mathbf{x}_i, z_i; \theta) = \mathbb{P}(Z_i = k) f(\mathbf{x}_i | Z_i = k) = \pi_k \varphi(\mathbf{x}_i; \boldsymbol{\mu}_k, \Sigma_k).$$

The indicators write this without case distinctions: exactly one z_{ik} equals 1 and the others are 0, so raising each component's candidate factor to the power z_{ik} keeps the right one and turns every other into $[\cdot]^0 = 1$:

$$p(\mathbf{x}_i, z_i; \theta) = \prod_{k=1}^K [\pi_k \varphi(\mathbf{x}_i; \boldsymbol{\mu}_k, \Sigma_k)]^{z_{ik}}.$$

The pairs being i.i.d., the complete-data likelihood is the product of these joint densities over the sample,

$$L_c(\theta) = \prod_{i=1}^n p(\mathbf{x}_i, z_i; \theta) = \prod_{i=1}^n \prod_{k=1}^K [\pi_k \varphi(\mathbf{x}_i; \boldsymbol{\mu}_k, \Sigma_k)]^{z_{ik}}, \quad (5)$$

and taking the logarithm turns both products into sums and each exponent into a factor:

$$\ell_c(\theta) = \log L_c(\theta) = \sum_{i=1}^n \sum_{k=1}^K z_{ik} [\log \pi_k + \log \varphi(\mathbf{x}_i; \boldsymbol{\mu}_k, \Sigma_k)]. \quad (6)$$

Compare (6) with the observed-data log-likelihood (4): no log-of-sum anywhere. The sum over k sits *outside* the logarithm, because the indicators have already decided who belongs where. Maximizing ℓ_c is chapter 1's problem K times over: with the counts $n_k = \sum_i z_{ik}$, the maximizer is $\hat{\pi}_k = n_k/n$, $\hat{\boldsymbol{\mu}}_k$ the mean of group k 's points, and $\hat{\Sigma}_k$ their scatter. This is the split-and-fit estimation of section 1, in formulas.

The E-step: expected indicators. The indicators are missing. EM's move is the simplest one available: replace each missing z_{ik} by its *conditional expectation* given what we do observe, under the current parameter guess $\theta^{(t)}$. Since z_{ik} is a 0/1 variable, its expectation is a probability, and one we have already computed:

$$\mathbb{E}[z_{ik} | \mathbf{x}_i; \theta^{(t)}] = \mathbb{P}(Z_i = k | \mathbf{x}_i; \theta^{(t)}) = \gamma_{ik}^{(t)},$$

the responsibilities of section 2. Because ℓ_c is *linear* in the z_{ik} , substituting expectations for indicators gives exactly the expected complete-data objective

$$Q(\theta | \theta^{(t)}) = \mathbb{E}[\ell_c(\theta) | \mathbf{x}_1, \dots, \mathbf{x}_n; \theta^{(t)}] = \sum_{i=1}^n \sum_{k=1}^K \gamma_{ik}^{(t)} [\log \pi_k + \log \varphi(\mathbf{x}_i; \boldsymbol{\mu}_k, \Sigma_k)]. \quad (7)$$

The M-step: fractional counts. Now maximize Q over θ . Compare Q with ℓ_c : they are the *same function*, with the hard indicators $z_{ik} \in \{0, 1\}$ replaced by the soft weights $\gamma_{ik} \in [0, 1]$. The maximization therefore goes through verbatim, with *fractional counts* in place of integer ones: writing $n_k = \sum_i \gamma_{ik}$ for component k 's effective number of points,

$$\pi_k^{new} = \frac{n_k}{n}, \quad \boldsymbol{\mu}_k^{new} = \frac{1}{n_k} \sum_{i=1}^n \gamma_{ik} \mathbf{x}_i, \quad \Sigma_k^{new} = \frac{1}{n_k} \sum_{i=1}^n \gamma_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k^{new})(\mathbf{x}_i - \boldsymbol{\mu}_k^{new})^\top \quad (8)$$

that is, group proportions together with chapter 1's means and scatters, each observation counted in every group according to its responsibility. Given the decision to replace the missing indicators by their expectations, the updates are forced rather than postulated. The algorithm is now just the alternation of these two computations:

Definition 3.1 (EM for Gaussian mixtures)

Initialize $\theta^{(0)}$; alternate until convergence:

- *E-step*: given $\theta^{(t)}$, compute the responsibilities $\gamma_{ik} = \mathbb{P}(Z_i = k \mid \mathbf{x}_i; \theta^{(t)})$ and form $Q(\theta \mid \theta^{(t)})$;
- *M-step*: set $\theta^{(t+1)} = \arg \max_{\theta} Q(\theta \mid \theta^{(t)})$ (the fractional-count updates above).

What makes EM more than a plausible heuristic is that each round provably improves the actual objective:

Theorem 3.1 (Monotonicity of EM)

The observed-data log-likelihood never decreases along EM iterations: $\ell(\theta^{(t+1)}) \geq \ell(\theta^{(t)})$.

Proof. Fix $\theta^{(t)}$ and write $q_{ik} = \gamma_{ik}^{(t)}$ for the E-step responsibilities, which satisfy $q_{ik} > 0$ and $\sum_k q_{ik} = 1$. For any θ , insert q_{ik}/q_{ik} inside each inner sum of ℓ and apply Jensen’s inequality to the concave logarithm ($\log \mathbb{E} \geq \mathbb{E} \log$ for the distribution q_i):

$$\ell(\theta) = \sum_{i=1}^n \log \sum_{k=1}^K q_{ik} \frac{\pi_k \varphi(\mathbf{x}_i; \boldsymbol{\mu}_k, \Sigma_k)}{q_{ik}} \geq \sum_{i=1}^n \sum_{k=1}^K q_{ik} \log \frac{\pi_k \varphi(\mathbf{x}_i; \boldsymbol{\mu}_k, \Sigma_k)}{q_{ik}} = Q(\theta \mid \theta^{(t)}) + H,$$

where Q is the expected complete-data objective defined above and $H = -\sum_{i,k} q_{ik} \log q_{ik}$ does not depend on θ . Two facts finish the proof. First, at $\theta = \theta^{(t)}$ the bound is an *equality*: plugging the definition of γ_{ik} , each ratio $\pi_k \varphi(\mathbf{x}_i)/q_{ik}$ equals the mixture density $f(\mathbf{x}_i; \theta^{(t)})$, a constant in k , and Jensen is tight on constants. Second, the M-step maximizes Q by construction (the exact updates (8)). Chaining the two:

$$\ell(\theta^{(t+1)}) \geq Q(\theta^{(t+1)} \mid \theta^{(t)}) + H \geq Q(\theta^{(t)} \mid \theta^{(t)}) + H = \ell(\theta^{(t)}). \quad \square$$

The proof uses the M-step only through $Q(\theta^{(t+1)} \mid \theta^{(t)}) \geq Q(\theta^{(t)} \mid \theta^{(t)})$. The regularized covariance updates of section 4 are not the exact maximizer of Q , so for them the guarantee holds only as long as each update still increases Q . In the run of figure 3, which uses such an update, the likelihood does increase at every iteration.

Figure 3 shows the algorithm at work on the customer data: from a poor random start ($\ell = -4683$), a single round of soft reassignment already reshapes the components ($\ell = -4508$), and by iteration 25 the three ellipses have locked onto the groups ($\ell = -3720$), with the points sitting between groups shared between components, in blended colors, rather than assigned to one group.

4 What can go wrong?

Monotone improvement is a strong property, but it is a *local* guarantee, and fitted mixtures fail in reproducible, avoidable ways.

Local maxima and initialization. Theorem 3.1 promises only that the likelihood never decreases. At best EM settles at a stationary point of ℓ , typically a local maximum, and mixture likelihoods have many (different clusterings can each be locally optimal); at worst it runs off toward the degenerate spikes described next. Where EM lands depends entirely on $\theta^{(0)}$. Standard practice runs EM from several initializations (random data points as initial means, or the K-means solution of section 7) and keeps, among the runs that did not degenerate, the one with the best final likelihood, stopping each run when ℓ stalls between iterations.

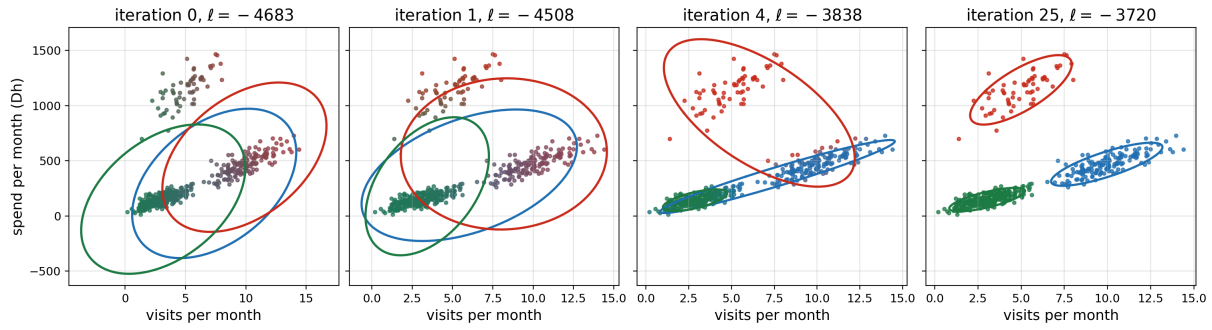


Figure 3: One EM run ($K = 3$) on the customers cloud: point colors blend the component colors in proportion to responsibilities; ellipses are the current component covariances. The log-likelihood increases at every step, as theorem 3.1 promises for the exact M-step (the run adds the tiny ridge of section 4).

The likelihood is unbounded. A second failure is more serious: the global maximum of the likelihood *does not exist*. Center one component on a single data point, $\mu_1 = \mathbf{x}_1$, and let $\Sigma_1 \rightarrow 0$: that component’s density at $\mathbf{x}_1$ explodes while every other point remains covered by the remaining components, so $\ell(\theta) \rightarrow +\infty$. This behavior, called *component collapse*, occurs in practice: a run whose component captures one or two points can enter this regime, its likelihood increasing while the component degenerates into a spike around a single observation. What we seek is a good *local* maximizer away from these degeneracies. Figure 4 shows the pull toward collapse on a crescent-shaped cloud whose curved geometry no small- K mixture fits well, inviting many components. As K grows, the fitted density wraps the sample ever more tightly and the likelihood climbs at every step. At $K = 12$ the improvement is no longer geometric: several components have contracted onto a handful of nearly aligned points and flattened onto the line through them, visible as needle-thin ridges of the density. These are the first steps of the unbounded path above: a component’s likelihood gain grows like $-d \log \sigma$ when it shrinks to a point spike of scale σ , and like $-\log \sigma_{\min}$ when only its smallest standard deviation $\sigma_{\min}$ shrinks, as for these needles.

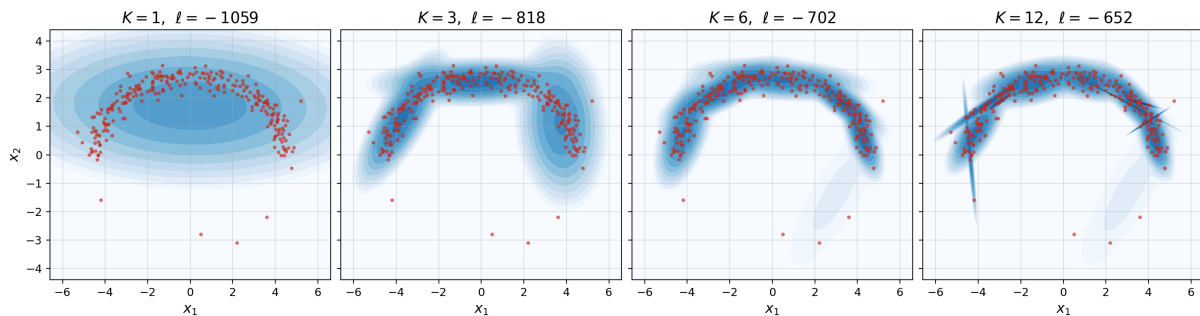


Figure 4: Fitted mixture density (blue, log scale) on a crescent-shaped cloud for a growing number of components. The log-likelihood increases at every step. At $K = 12$ several components have contracted onto a few points (the needle-thin ridges), and further gains come from sharpening these spikes, not from fitting the shape of the cloud: the onset of component collapse.

Singular covariances and regularization. Short of full collapse, a component whose effective sample $n_k = \sum_i \gamma_{ik}$ is not much larger than d produces an M-step covariance as ill-conditioned as any sample covariance estimated from too few points: this is chapter 1’s covariance-estimation problem again, now per component; each Σ_k is estimated from effectively n_k observations, and there are K of them. The remedy is also chapter 1’s, applied inside the M-step: *regularize the estimator* by blending each updated covariance $\hat{\Sigma}_k$ (the update (8)) with a fixed multiple of the

identity,

$$\Sigma_k \leftarrow (1 - \lambda) \hat{\Sigma}_k + \lambda \tau \mathbf{I}_d, \quad \tau = \text{tr}(\hat{\Sigma}_{\text{MLE}})/d, \quad (9)$$

for a small $\lambda \in (0, 1]$, where $\hat{\Sigma}_{\text{MLE}}$ is chapter 1’s sample covariance of the whole dataset; the scale τ , the average variance of the data, is computed once before EM starts and never changes. The eigenvectors of $\hat{\Sigma}_k$ are untouched and every eigenvalue is pulled toward τ , with a floor at $\lambda\tau > 0$ that no iteration can lower: the matrix stays invertible, each component density stays below $(2\pi\lambda\tau)^{-d/2}$, and the collapse $\sigma \rightarrow 0$ becomes *impossible by construction*. The target scale has to be fixed: a scale computed from $\hat{\Sigma}_k$ itself would shrink with the collapsing component and remove the floor exactly when it is needed. Standard EM implementations apply a version of this; ours adds a fixed ridge $10^{-6} \mathbf{I}_d$ to each M-step covariance, a tiny instance of the same fixed floor (tiny enough that the needles of figure 4 formed with it in place).

5 Covariance structures

The parameter bill. The full mixture is expensive. Counting free parameters: $K - 1$ weights, Kd mean coordinates, and one symmetric covariance per component (the dominant term), $d(d + 1)/2$ entries each:

$$p_K = (K - 1) + Kd + K \frac{d(d + 1)}{2}. \quad (10)$$

For $d = 50$ and $K = 10$ that is $9 + 500 + 12\,750 = 13\,259$ parameters, to be estimated from unlabeled data in which each component effectively sees only its share n_k of the observations. This is the regime where the failures of section 4 occur.

Constrained variants. The structural response is chapter 1’s other move, *regularize the problem*: restrict the covariance structure, trading cluster expressiveness for parameter economy. From the most to the least expensive in general (shared is cheaper than diagonal when $K > (d + 1)/2$, dearer otherwise):

- *full*: one free Σ_k per component ($K d(d + 1)/2$ covariance parameters); arbitrary shape and orientation per cluster;
- *shared* (tied): $\Sigma_k = \Sigma$ for all k ($d(d + 1)/2$); all clusters have the same shape, only their centers differ;
- *diagonal*: $\Sigma_k = \text{diag}(\sigma_{k1}^2, \dots, \sigma_{kd}^2)$ (Kd); axis-aligned ellipsoidal clusters, no within-cluster correlations;
- *spherical*: $\Sigma_k = \sigma_k^2 \mathbf{I}_d$ (K); round clusters of per-component radius.

Every variant keeps the same EM algorithm: only the M-step’s covariance update changes, maximizing Q under the constraint (for the shared variant, a responsibility-weighted pooling across components; for the diagonal one, coordinatewise variances). The choice is a modeling decision of the kind flagged in chapter 1: the constrained models gain stability on small groups at the cost of expressiveness (a spherical model cannot represent an elongated, tilted cluster). How to make the choice, and how to choose K itself, is the next section’s question.

6 Choosing the number of components

The number of components cannot be chosen by the likelihood. Since the likelihood is unbounded, write $\hat{\theta}_K$ for the best non-degenerate EM fit with K components. The models are nested (a mixture with $K + 1$ components can imitate any K -component one by splitting one component into two identical halves of weight $\pi_k/2$ each), so the best attainable fit $\ell(\hat{\theta}_K)$ does not decrease

with K , and selecting the best-fitting K would always answer “as many as possible”. This is the first appearance of a distinction that recurs throughout the course: a better fit is not a better model. A mixture with one component per data point fits the sample arbitrarily well while describing nothing beyond it; this failure is called *overfitting*: a flexible model absorbing the sampling noise of the particular dataset. The opposite failure, a model too rigid to represent real structure, is *underfitting*; a single Gaussian forced onto the three-group customer cloud is an example. The standard remedy penalizes the parameter count p_K of section 5: the *Bayesian information criterion* [3]

$$\text{BIC}(K) = -2\ell(\hat{\theta}_K) + p_K \log n \quad (11)$$

is minimized over K . Figure 5 plots both terms on the customer data ($d = 2$, $n = 470$, three latent groups). The fit term $-2\ell(\hat{\theta}_K)$ falls steeply while real structure is being captured (9201 at $K = 1$, 7831 at $K = 2$, 7440 at $K = 3$), then decreases only slightly (7429, 7415, 7410): each extra component improves the fit marginally, by modeling sampling noise, so the fit curve alone cannot select K . BIC sits above it by the penalty $p_K \log n$, the growing gap between the two curves. The penalty grows linearly in p_K while the fit gains shrink, so BIC attains its minimum 7544 at $K = 3$ and increases thereafter, the number of latent groups in the data.

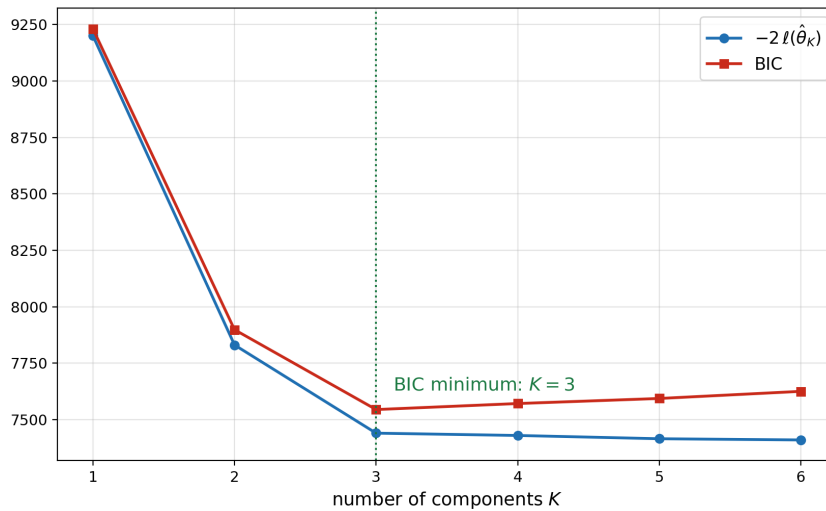


Figure 5: Model selection on the customers cloud (three latent groups). The fit term $-2\ell(\hat{\theta}_K)$ decreases forever; BIC adds the penalty $p_K \log n$ (the widening gap) and turns into a minimum at $K = 3$.

The same criterion applies to the covariance variants, whose p_K differ precisely as section 5 counted: BIC compares them as it compares values of K . It does not apply to the regularization λ of section 4, which leaves p_K unchanged: BIC then reduces to the likelihood, which favors $\lambda \rightarrow 0$. The weight λ is instead chosen by the likelihood of held-out data (cross-validation), a score that also serves for K and the covariance structure when data are plentiful. BIC balances fit against complexity without setting data aside: when the sample is too scarce to hold part of it out, the penalty replaces the held-out score.

7 GMM vs K-means

The best-known clustering algorithm, *K-means*, predates EM and underlies much of this machinery: pick K centers, assign each point to its nearest center, recompute each center as the mean of its points, repeat. It is exactly what EM degenerates into under three drastic simplifications: force all covariances to $\sigma^2 \mathbf{I}_d$, all priors to $1/K$, and let $\sigma \rightarrow 0$. The responsibilities then collapse to hard 0/1 assignments to the nearest center, and the M-step becomes the plain mean.

K-means, in other words, is a Gaussian mixture that has silently assumed *spherical, equally-sized, equally-weighted* clusters.

The comparison is again a matter of assumptions. K-means is fast, simple and uses few parameters; when its hidden assumptions hold, it is hard to beat. When they fail, the algorithm gives no warning: figure 6 shows two elongated, tilted clusters. K-means, blind to shape, cuts *across* both clusters and misassigns 27% of the points, while the full-covariance GMM recovers the two groups almost perfectly (0.8%). The price of that fidelity was computed in section 5: each full covariance costs $d(d+1)/2$ parameters, so in high dimension the GMM needs data that K-means does not; the constrained variants of section 5 populate the spectrum between the two extremes.

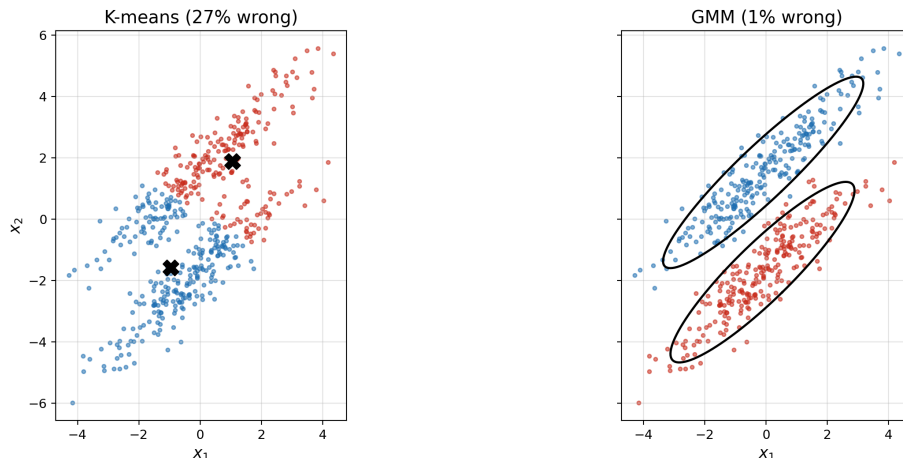


Figure 6: Two elongated correlated clusters. K-means (left) assumes spherical clusters and slices across the true groups; the full-covariance GMM (right) models the shape and recovers them.

8 Interpreting the fitted components

A mixture has now been fitted by EM, with restarts, a sensible covariance structure and a BIC-chosen K . Two cautions are needed before its components are interpreted.

Label switching. The caveat recorded in section 2 can now be stated in full. Permuting the components (relabeling (π_1, μ_1, Σ_1) as component 2 and vice versa) leaves the mixture density f strictly unchanged, so the parameters are identifiable only *up to relabeling*: a K -component mixture has $K!$ equivalent parameterizations. For the clustering itself this is harmless (the *partition* of the data does not depend on the numbering), but it matters the moment components are tracked across runs: “component 1” of one EM run may be “component 2” of the next, and averaging parameters across bootstrap replicates, or comparing “component 1” between two subsamples, is meaningless until the components have been matched (by their means, or by the overlap of their responsibilities).

Components are not always clusters. The model fits a density; the scientific question is usually about subpopulations, and the two need not coincide. A single skewed or heavy-tailed cluster is often fitted by *two* Gaussian components sharing the work: BIC may rightly prefer $K = 3$ on data with two “real” groups, one of them non-Gaussian. The safe reading is that components are building blocks of a density estimate, responsibilities are calibrated membership probabilities under that estimate, and any claim that “component k is a real subtype” (three customer types, two cell subtypes) needs evidence beyond the fit: stability across restarts and subsamples, and ideally external validation.

What the fitted mixture delivers. With those cautions in place, return to the prologue’s five purposes of a fitted distribution, on the customer data that opened the chapter. *Typicality*: the mixture reports three typical customers (one per segment, each with its own spread) instead of one average customer who resembles nobody in the file. *Dependence*: within-group covariances now describe how visits and spend co-vary *inside* a segment, no longer confounded with the differences between segments. *Anomaly*: thresholding the *mixture* density flags customers implausible under *every* segment, and chapter 1’s chi-squared calibration applies within each component. *Prediction*: a missing spend entry is imputed by a responsibility-weighted blend of the per-segment distributions of spend given visits, each one chapter 1’s conditional distribution of a Gaussian vector. *Generation*: synthetic customers are drawn group-first (pick a segment, then a customer from it) and reproduce three clouds rather than one average blur. One family change, motivated by one picture, repaired all five answers.

References

- [1] K. Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London A*, 185:71–110, 1894.
- [2] A. P. Dempster, N. M. Laird and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
- [3] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.