Expectation Maximization (EM) Algorithm

Machine Learning Course - CS-433
Nov 26, 2024
Nicolas Flammarion

Gaussian Mixture Models

Data is modeled as a mixture of K Gaussian distributions

Instance space is $\mathcal{X} = \mathbb{R}^D$ and each sample $\mathbf{x}$ is generated as follows:

1. Choose a random component $k \in \{1,2,...,K\}$, i.e., let z be a multinomial r.v. with $p(z = k) = \pi_k$
2. Sample $\mathbf{x}$, given the value of $z = k$, from a Gaussian distribution:

$$p(\mathbf{x} \mid z = k) = \frac{1}{(2\pi)^{D/2} \mid \Sigma_k \mid^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma_k^{-1} (\mathbf{x} - \mu_k)\right)$$

Gaussian Mixture Models

Then we can write the distribution of **x** as:

mixture components

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(z = k)p(\mathbf{x} \mid z = k)$$

$$= \sum_{k=1}^{K} \pi_k \mathcal{N}\left(x \mid \mu_k, \Sigma_k\right)$$

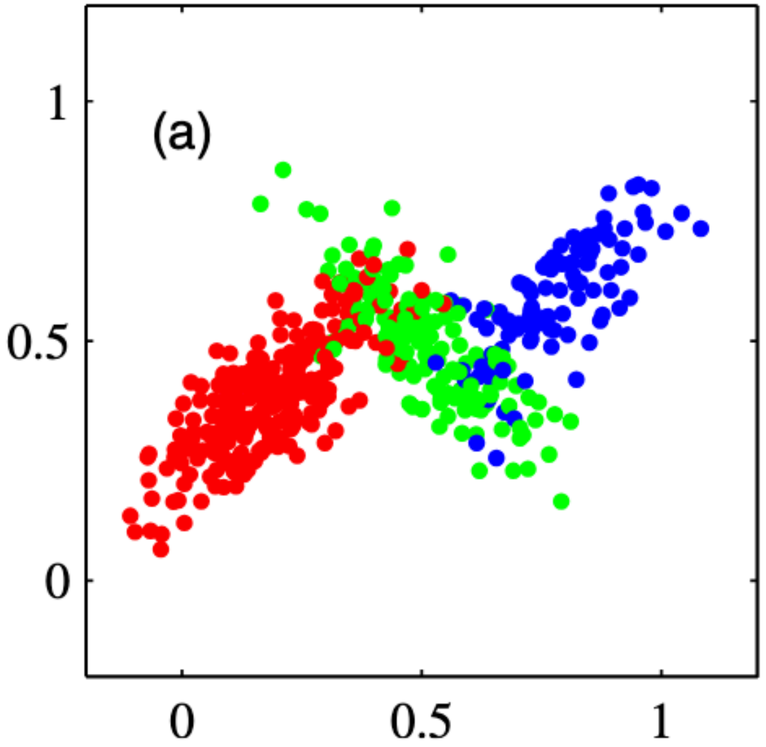
mixing proportion

$$= \sum_{k=1}^{K} \frac{\pi_k}{(2\pi)^{D/2} \mid \Sigma_k \mid^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma_k^{-1}(\mathbf{x} - \mu_k)\right)$$

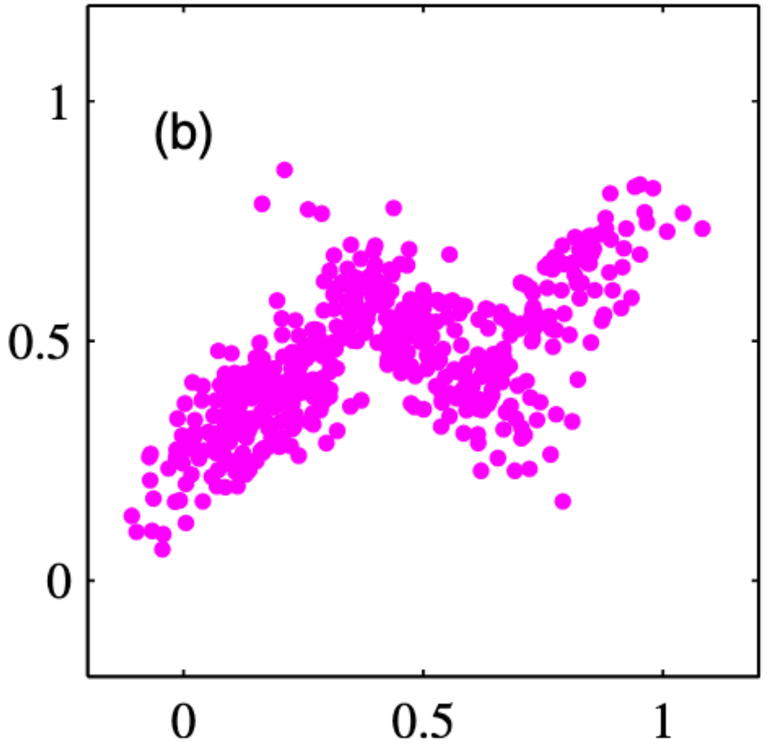
z is a latent (hidden) variable that is not directly observed

We introduce z because it helps us describe $p(\mathbf{x})$ in a simple parametric form

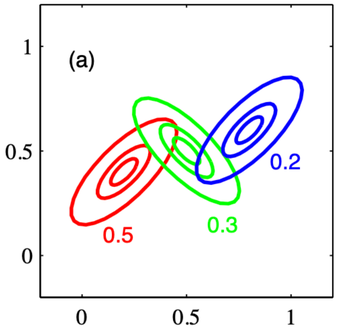
Visualizing the Joint and Marginal Distributions of GMM



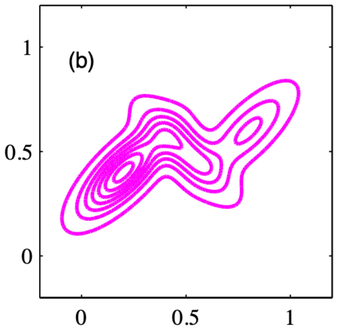
Samples from the joint distribution $p(\mathbf{x}, z)$, with z represented by colors



Samples from the marginal distribution $p(\mathbf{x})$



Joint distribution $p(\mathbf{x}, z)$, with z represented by colors



Marginal distributions $p(\mathbf{x})$

MLE with Latent Variables

unobserved

Given an i.i.d. sample $\mathscr{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and a latent variable model $p(\mathbf{x}, z \,|\, \theta)$, we want to maximize the log-likelihood of $\mathscr{D}$, i.e. $\hat{\theta}_{MLE} = \arg\max_\theta \log p(\mathscr{D} \,|\, \theta)$:

$$
\begin{aligned}
\log p(\mathscr{D} \,|\, \theta) &= \log \prod_{n=1}^{N} p\left(\mathbf{x}_n \,|\, \theta\right) \\
&= \sum_{n=1}^{N} \log p\left(\mathbf{x}_n \,|\, \theta\right) \\
&= \sum_{n=1}^{N} \log \left(\sum_{z_n=1}^{K} p\left(\mathbf{x}_n, z_n \,|\, \theta\right) \right) \\
&= \sum_{n=1}^{N} \log \left(\sum_{z_n=1}^{K} p\left(z_n \,|\, \theta\right) p\left(\mathbf{x}_n \,|\, z_n, \theta\right) \right)
\end{aligned}
$$

latent variables $(z_1, \cdots, z_N)$

MLE with Latent Variables

unobserved

Given an i.i.d. sample $\mathscr{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and a latent variable model $p(\mathbf{x}, z \,|\, \theta)$, we want to maximize the log-likelihood of $\mathscr{D}$

$$\log p(\mathscr{D} \,|\, \theta) = \log \prod_{n=1}^{N} p\left(\mathbf{x}_n \,|\, \theta\right)$$

$$= \sum_{n=1}^{N} \log p\left(\mathbf{x}_n \,|\, \theta\right)$$

$$= \sum_{n=1}^{N} \log \left(\sum_{z_n=1}^{K} p\left(\mathbf{x}_n, z_n \,|\, \theta\right) \right)$$

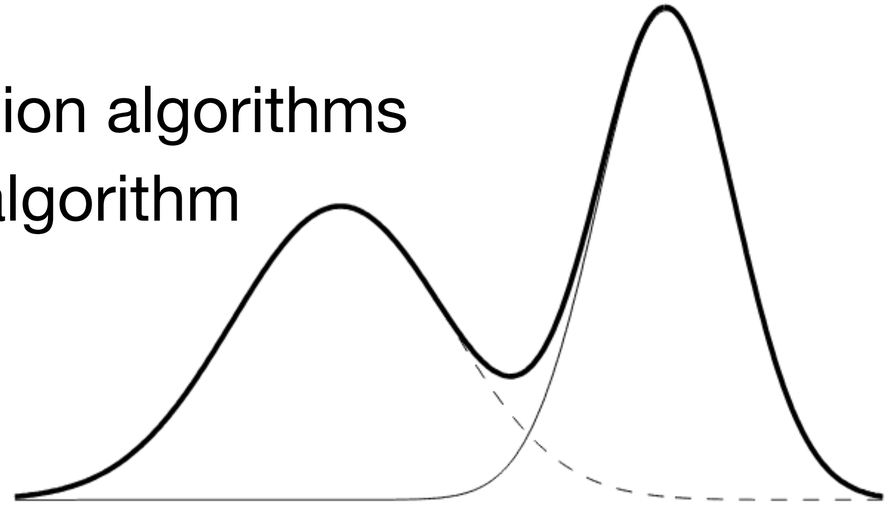
$$= \sum_{n=1}^{N} \log \left(\sum_{z_n=1}^{K} p\left(z_n \,|\, \theta\right) p\left(\mathbf{x}_n \,|\, z_n, \theta\right) \right)$$

In many cases, the summation inside the log makes the optimization problem computationally hard. The Expectation-Maximization (EM) is designed for cases in which, had we known the values of the latent variables z, the MLE would have been computationally tractable.

MLE Objective for Gaussian Mixtures

$$\max_\theta \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N} \left(\mathbf{x}_n \mid \mu_k, \Sigma_k \right) \right)$$

- This is a **non-convex** objective
- There is **no unique optimum** (permutation of K clusters)
- The objective is **unbounded**
- Can be solved directly using black-box optimization algorithms
- Main focus: How to solve this problem with EM algorithm



Average of two probability distributions of two Gaussian for which it is natural to introduce a mixture model

MLE with access to Labels for GMM

Assume observing the *complete data* $\mathcal{D}_c = \{(\mathbf{x}_n, z_n)\}_{n \in [N]}$, the *complete log likelihood is:*

$$\mathcal{L}_c(\theta \mid \mathcal{D}_c) = \sum_{n=1}^{N} \log(\pi_{z_n} \mathcal{N}(\mathbf{x}_n \mid \mu_{z_n}, \Sigma_{z_n}))$$ → **Convex!**

We could easily compute MLE in closed form:

$$\hat{\pi}_k = \frac{1}{N} \sum_n \mathbf{1}\{z_n = k\} \qquad \hat{\mu}_k = \frac{\sum_n \mathbf{1}\{z_n = k\}\, \mathbf{x}_n}{\sum_n \mathbf{1}\{z_n = k\}}$$

$$\hat{\Sigma}_k = \frac{1}{\sum_n \mathbf{1}\{z_n = k\}} \sum_n \mathbf{1}\{z_n = k\}\, (\mathbf{x}_n - \hat{\mu}_k)(\mathbf{x}_n - \hat{\mu}_k)^\top$$

➡ If we had access to the labels, the problem simplifies

Hard EM Algorithm

Given dataset $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, initialize the parameters $\theta^{(0)} = \{\pi_k, \mu_k, \Sigma_k\}_{k \in [K]}$

For $t = 1, 2, \ldots$

E-step: Predict the most likely class for each data point

$$z_n^{(t)} = \arg\max_z p(z \mid \mathbf{x}_n, \theta^{(t-1)}) = \arg\max_z p(z \mid \theta^{(t-1)}) p(\mathbf{x}_n \mid z, \theta^{(t-1)})$$

$$= \arg\max_z \pi_z \mathcal{N}(\mathbf{x}_n \mid \mu_z, \Sigma_z)$$

Now we have complete labeled data! Let $\mathcal{D}_c^{(t)} = \{(\mathbf{x}_n, z_n^{(t)})\}_{n \in [N]}$

M-step: Compute MLE of θ using $\mathcal{D}_c^{(t)}$

$$\theta^{(t)} = \arg\max_\theta p(\mathcal{D}_c^{(t)} \mid \theta) = \arg\max_\theta \sum_{n=1}^N \log(\pi_{z_n^{(t)}} \mathcal{N}(\mathbf{x}_n \mid \mu_{z_n^{(t)}}, \Sigma_{z_n^{(t)}}))$$

Hard EM & K-Means

K-Means is a special case of Hard-EM where optimization is over the means μ_k with fixed uniform weights ($\pi_k = 1/K \ \forall k$) and fixed identical spherical covariances ($\Sigma_k = \sigma^2 I \ \forall k$).

E-step:

$$z_n^{(t)} = \arg\max_k \pi_k \mathcal{N}(\mathbf{x}_n \mid \mu_k, \Sigma_k) = \arg\max_k \frac{1}{K} \frac{1}{\sqrt{(2\pi\sigma^2)^d}} \exp(-\|\mathbf{x}_n - \mu_k\|^2 / 2\sigma^2)$$

$$= \arg\min_k \|\mathbf{x}_n - \mu_k\|^2$$

M-step:

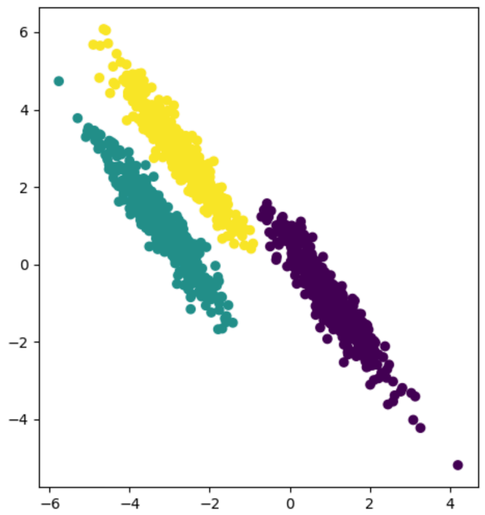
$$\theta^{(t)} = \arg\max_\theta \sum_{n=1}^N \log(\pi_{z_n^{(t)}} \ \mathcal{N}(\mathbf{x}_n \mid \mu_{z_n^{(t)}}, \Sigma_{z_n^{(t)}})) = \arg\max_\theta \sum_{n=1}^N \|\mathbf{x}_n - \mu_{z_n^{(t)}}\|^2$$

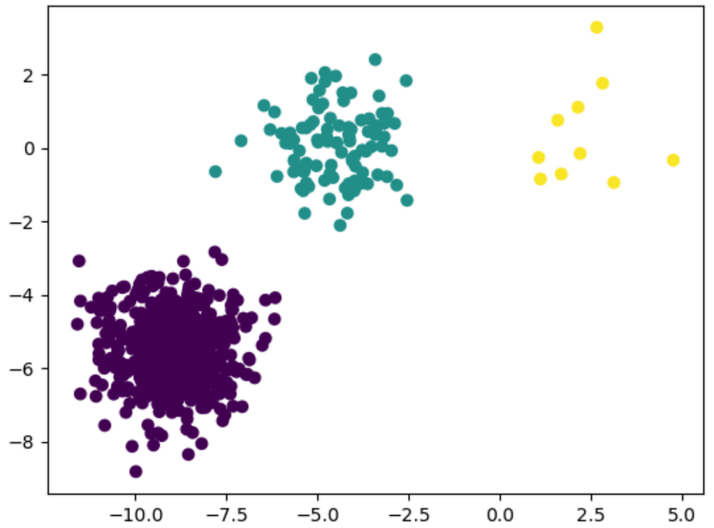
$$\implies \mu_k^{(t)} = \frac{1}{\sum_n \mathbf{1}\{z_n^{(t)} = k\}} \sum_n \mathbf{1}\{z_n^{(t)} = k\} \, \mathbf{x}_n$$

Hard EM vs K-Means

Hard-EM handles clusters of **different shapes and sizes**, unlike K-means

It allows flexible choices of Σ_k and π_k





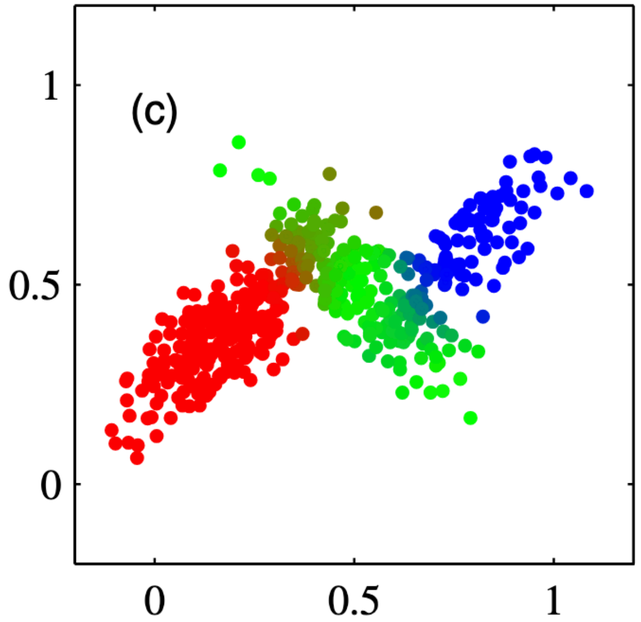
However, similar to K-means, Hard-EM assigns fixed labels to points, even when the model is uncertain

Posterior Probabilities for Latent Variables in GMM

For each data point, we compute the posterior over cluster memberships given the parameters $\theta = \{\pi_k, \mu_k, \Sigma_k\}_{k \in [K]}$

This models distributions over latent variables z rather than assigning a single (hard) label

$$q_k(\mathbf{x}_n) = p(z_n = k \,|\, \mathbf{x}_n, \theta)$$
$$= \frac{\pi_k \mathcal{N}(\mathbf{x}_n \,|\, \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n \,|\, \mu_j, \Sigma_j)}$$



Soft cluster assignments as posterior probabilities

Soft assignments capture uncertainty in cluster memberships (unlike K-means)

MLE for GMM: Coupled Parameter Updates

The MLE estimates $(\mu^\star, \Sigma^\star, \pi^\star) = \arg\max_{(\mu,\Sigma,\pi)} \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}_n \,|\, \mu_k, \Sigma_k\right) \right)$ satisfy:

$$\pi_k^\star = \frac{1}{N} \sum_{n=1}^{N} q_k(\mathbf{x}_n) \qquad \mu_k^\star = \frac{\sum_{n=1}^{N} q_k(\mathbf{x}_n)\, \mathbf{x}_n}{\sum_{n=1}^{N} q_k(\mathbf{x}_n)}$$

$$\Sigma_k^\star = \frac{\sum_{n=1}^{N} q_k(\mathbf{x}_n)(\mathbf{x}_n - \mu_k^\star)(\mathbf{x}_n - \mu_k^\star)^\top}{\sum_{n=1}^{N} q_k(\mathbf{x}_n)}$$

Chicken-and-egg problems: These equations are coupled!
Solving them jointly is difficult.

MLE for GMM: Coupled Parameter Updates

The MLE estimates $(\mu^\star, \Sigma^\star, \pi^\star) = \arg \max$ satisfy:

$$\pi_k^\star = \frac{1}{N} \sum_{n=1}^{N} q_k(\mathbf{x}_n)$$

$$\Sigma_k^\star = \frac{\sum_{n=1}^{N} q_k(\mathbf{x}_n)(}{\sum}$$

Chicken-and-egg problems: These ec
Solving them jointly is difficult.

Proof:

$$\frac{\partial l}{\partial \mu_k} = \frac{\partial}{\partial \mu_k} \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}_n \,|\, \mu_k, \Sigma_k\right) \right)$$

$$= \sum_{n=1}^{N} \frac{\pi_k}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\mathbf{x}_n \,|\, \mu_j, \Sigma_j\right)} \frac{\partial}{\partial \mu_k} \mathcal{N}\left(\mathbf{x}_n \,|\, \mu_k, \Sigma_k\right)$$

$$= \sum_{n=1}^{N} \frac{q_k(\mathbf{x}_n)}{\mathcal{N}\left(\mathbf{x}_n \,|\, \mu_k, \Sigma_k\right)} \frac{\partial}{\partial \mu_k} \mathcal{N}\left(\mathbf{x}_n \,|\, \mu_k, \Sigma_k\right)$$

$$= -\sum_{n=1}^{N} q_k(\mathbf{x}_n) \Sigma_k^{-1} (\mathbf{x}_n - \mu_k)$$

Setting to zero yields $\mu_k^\star = \frac{\sum_{n=1}^{N} q_k(\mathbf{x}_n)\, \mathbf{x}_n}{\sum_{n=1}^{N} q_k(\mathbf{x}_n)}$

(Soft) EM Algorithm for GMMs

Initialize the parameters $\theta^{(0)} = \{\pi_k, \mu_k, \Sigma_k\}_{k\in[K]}$. For $t = 1, 2, \ldots$:

E-step: Compute posterior cluster assignments

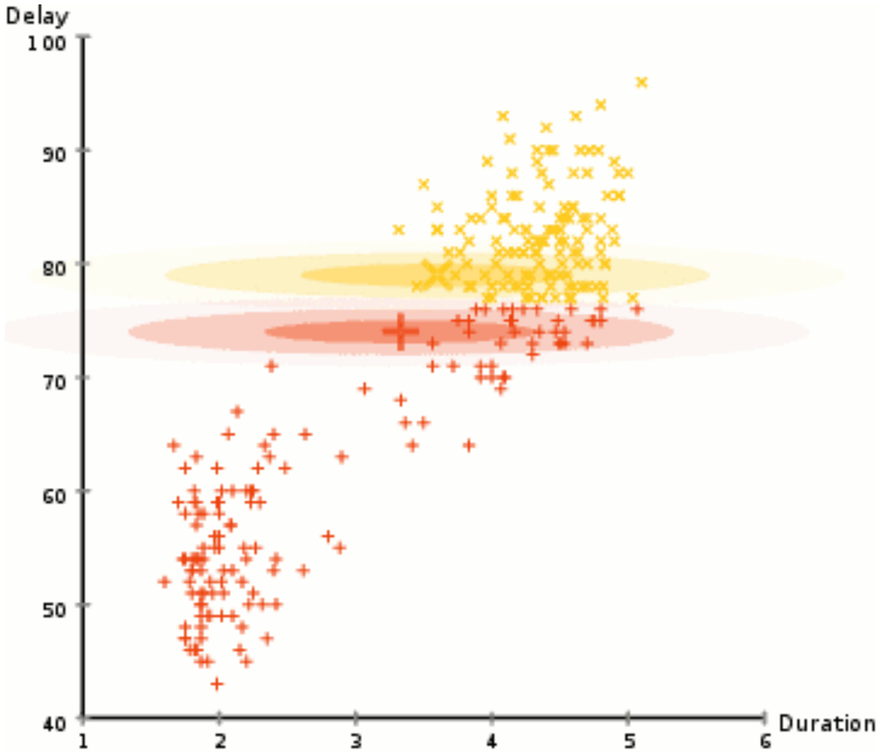
$$q_k^{(t)}(\mathbf{x}_n) = p(z_n = k \,|\, \mathbf{x}_n, \theta^{(t-1)}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n; \mu_k^{(t-1)}, \Sigma_k^{(t-1)})}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n; \mu_j^{(t-1)}, \Sigma_j^{(t-1)})}$$

M-step: Compute MLE for θ using the calculated $q_k^{(t)}(\mathbf{x}_n)$

$$\pi_k^{(t)} = \frac{1}{N}\sum_n q_k^{(t)}(\mathbf{x}_n), \;\; \mu_k^{(t)} = \frac{\sum_{n=1}^N q_k^{(t)}(\mathbf{x}_n)\,\mathbf{x}_n}{\sum_{n=1}^N q_k^{(t)}(\mathbf{x}_n)}, \;\; \Sigma_k^{(t)} = \frac{\sum_{n=1}^N q_k^{(t)}(\mathbf{x}_n)(\mathbf{x}_n - \mu_k^{(t)})(\mathbf{x}_n - \mu_k^{(t)})^\top}{\sum_{n=1}^N q_k^{(t)}(\mathbf{x}_n)}$$

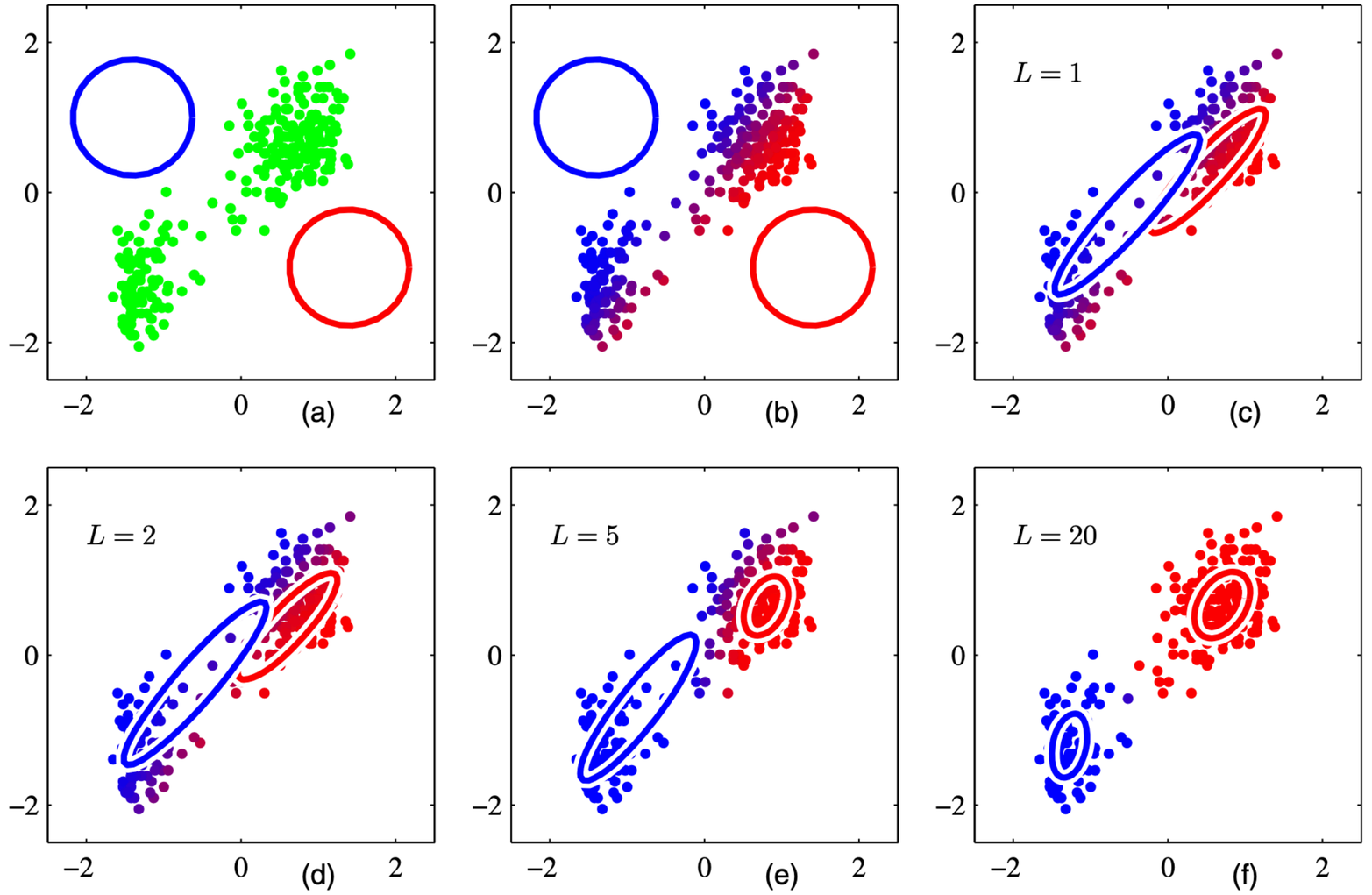
Iteration cost: $O(NKD^3)$

(Soft) EM Algorithm



EM Clustering of Old Faithful data

(Soft) EM Algorithm



Comparison of EM and K-Means

EM typically requires more iterations to converge compared to K-means

Each iteration is computationally more expensive ($O(NKD^3)$ vs $O(NKD)$)

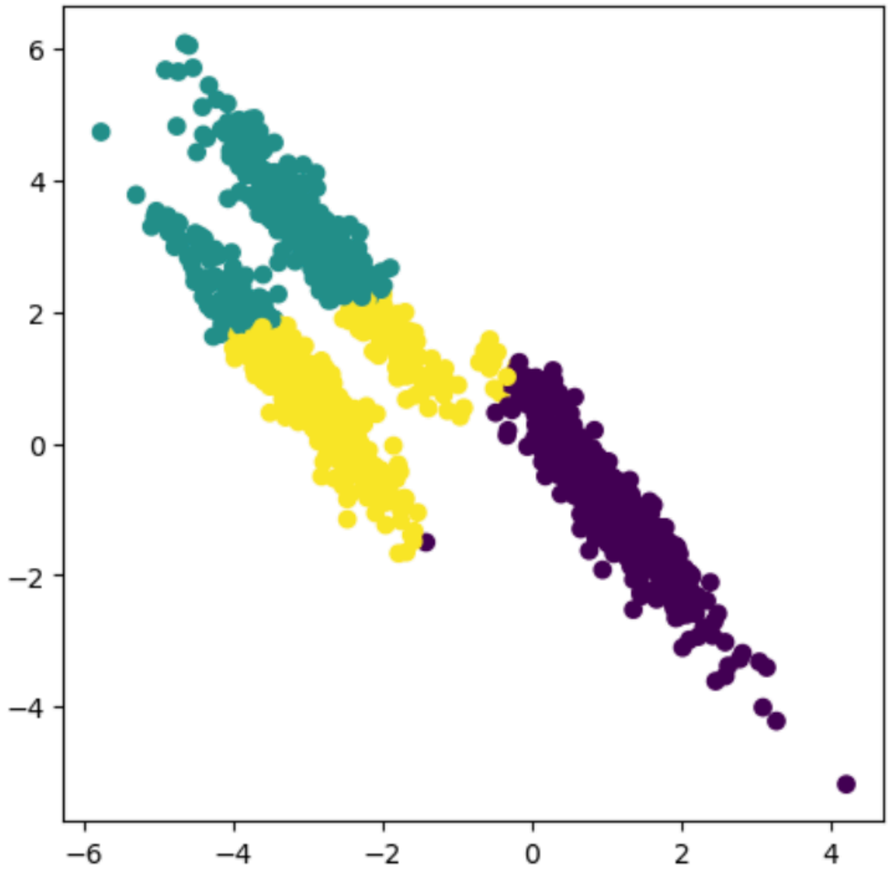
K-means is often used to initialize EM for faster convergence

EM outperforms K-means on challenging cases:

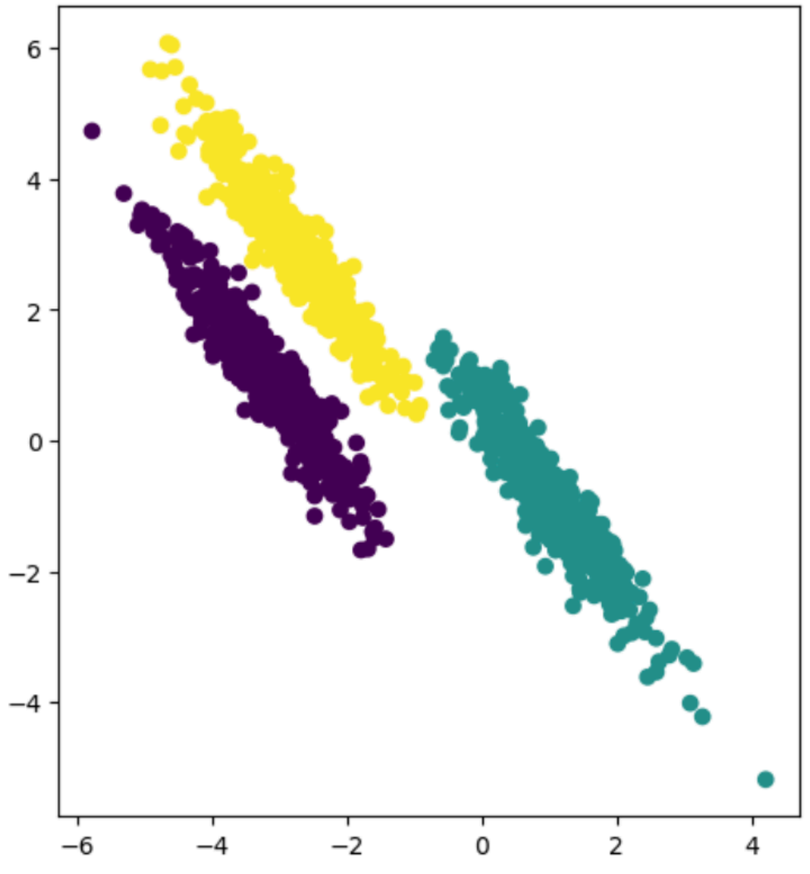
- Anisotropic clusters
- Unequal variances of clusters
- Unevenly sized clusters

➡ EM is more flexible than K-means but is slower and more computationally expensive, making it better suited for complex clustering tasks

EM vs K-Means: Anisotropic Clusters

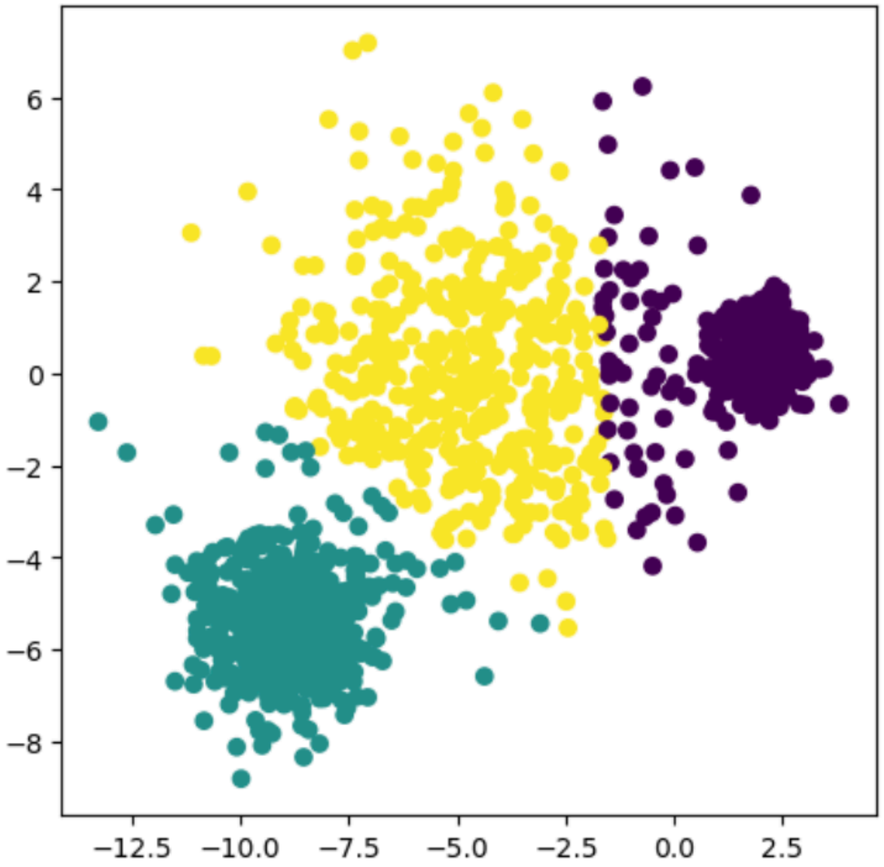


K-means

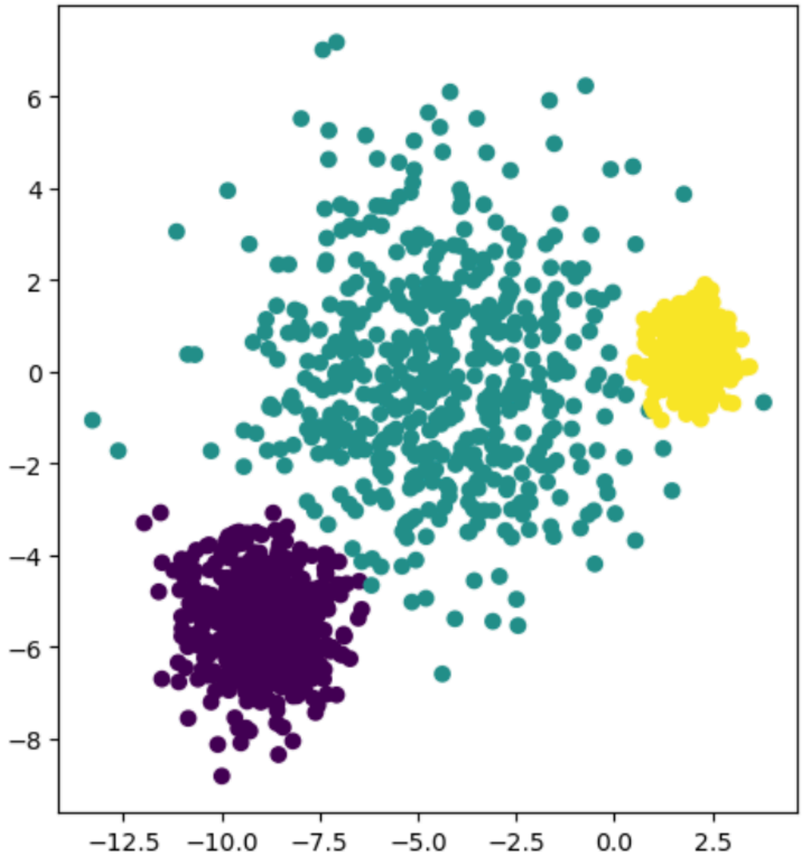


EM

EM vs K-Means: Unequal Variances

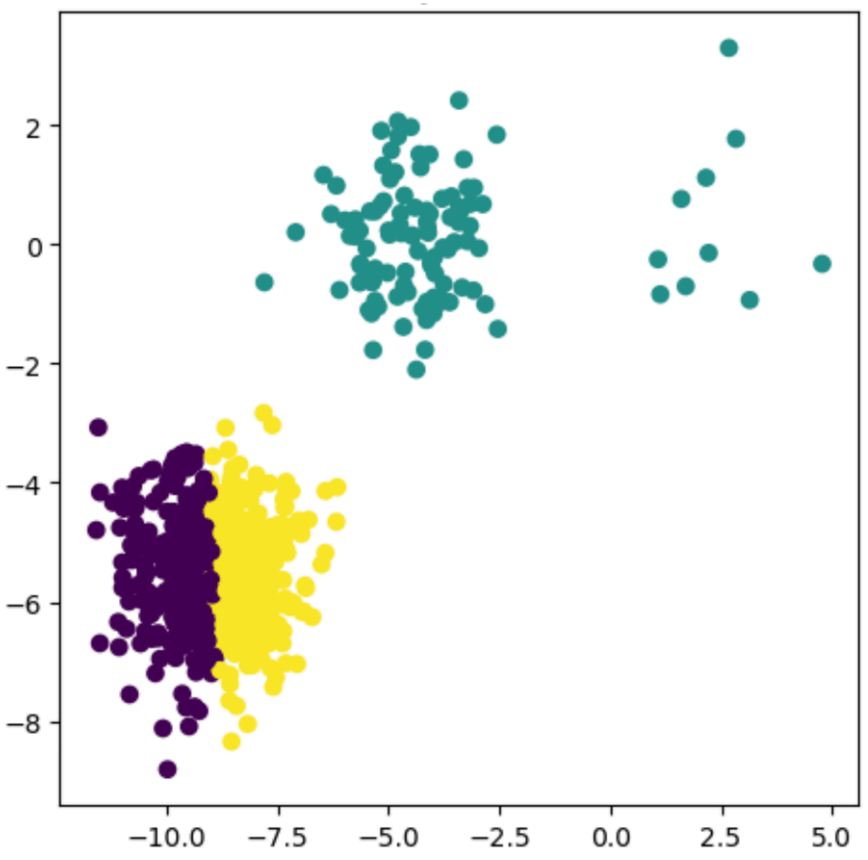


K-means

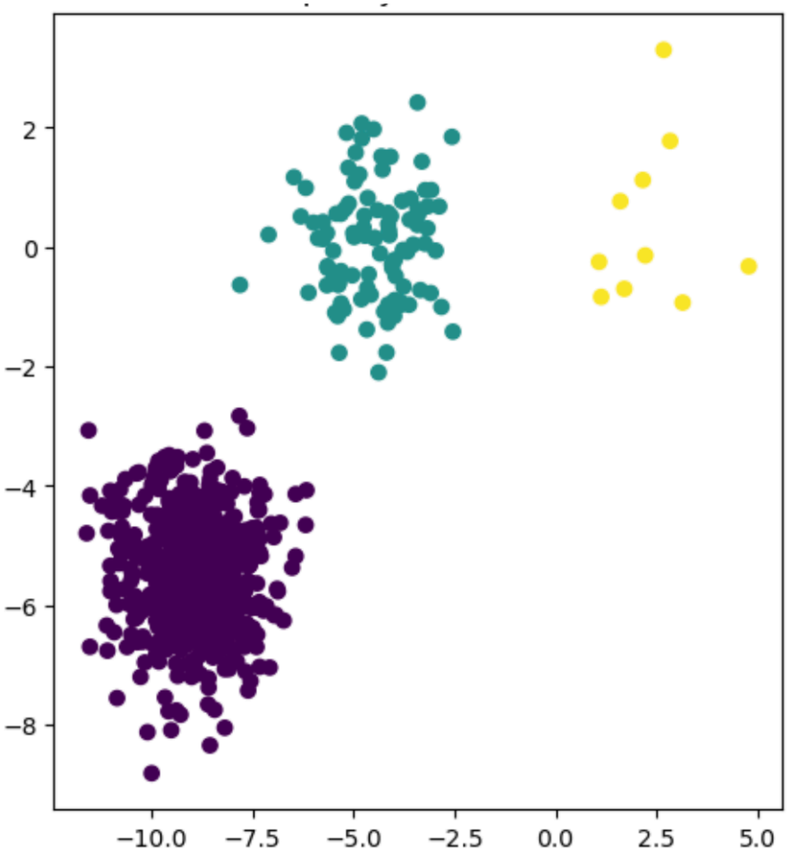


EM

EM vs K-Means: Uneven Sizes



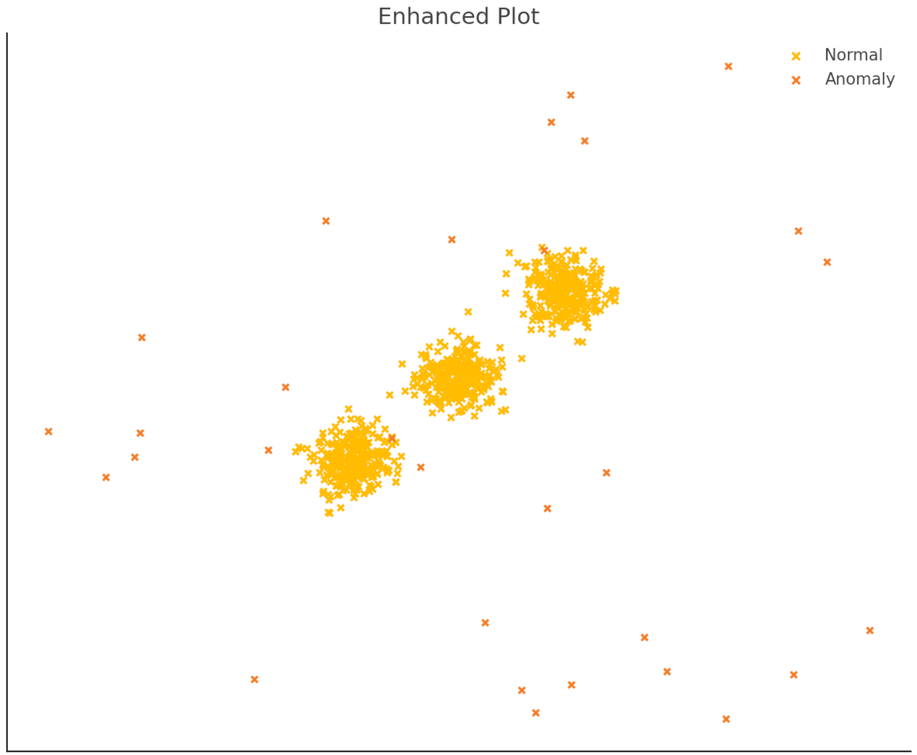
K-means



EM

Anomaly Detection with GMMs

After training a GMM, we compute the likelihood $p(\mathbf{x})$ for any new data point:



Likelihood-based anomaly detection: red points are flagged as anomalies based on the GMM threshold

$$p(\mathbf{x}) = \sum_{k=1}^{K} \frac{\pi_k \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_k)^T \Sigma_k^{-1}(\mathbf{x}-\mu_k)\right)}{(2\pi)^{d/2}\,|\,\Sigma_k\,|^{1/2}}$$

If $p(x) \leq \tau \implies$ **anomaly**

EM Algorithm

Maximum Likelihood from Incomplete Data via the *EM* Algorithm

By A. P. DEMPSTER, N. M. LAIRD and D. B. RUBIN

Harvard University and Educational Testing Service

[Read before the ROYAL STATISTICAL SOCIETY at a meeting organized by the RESEARCH SECTION on Wednesday, December 8th, 1976, Professor S. D. SILVEY in the Chair]

SUMMARY

A broadly applicable algorithm for computing maximum likelihood estimates from incomplete data is presented at various levels of generality. Theory showing the monotone behaviour of the likelihood and convergence of the algorithm is derived. Many examples are sketched, including missing value situations, applications to grouped, censored or truncated data, finite mixture models, variance component estimation, hyperparameter estimation, iteratively reweighted least squares and factor analysis.

EM Algorithm in General

Assumption: $(\mathbf{x}, z)$ are random variables, where $\mathbf{x}$ is observed (our data) and z is not observed (e.g. cluster membership)
Their joint distribution under a parametric model is given by $p(\mathbf{x}, z \,|\, \theta)$

Goal: Given a dataset $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$, maximize the log likelihood function defined in terms of the latent variables $Z = (z_1, \cdots, z_N)$:

$$\begin{aligned} \mathscr{L}(\theta) &= \log(p(\mathbf{X} \,|\, \theta)) = \log\left(\textstyle\sum_Z p(\mathbf{X}, Z \,|\, \theta) \right) \\ &\uparrow \\ &= \textstyle\sum_{n=1}^{N} \log(p(\mathbf{x}_n \,|\, \theta)) = \sum_{n=1}^{N} \log \sum_{z_n} p(\mathbf{x}_n, z_n \,|\, \theta) \end{aligned}$$

Applications: GMM, HMM and many others!

EM Algorithm in General

We introduce a distribution $q(Z)$ over the latent variables, i.e. $q(Z) \geq 0$ and $\sum_Z q(Z) = 1$. For any such $q(Z)$,

$$\begin{aligned}
\mathscr{L}(\theta) = \log\left(p(\mathbf{X}|\theta)\right) &= \sum_Z q(Z)\log\left(p(\mathbf{X}|\theta)\right) \\
&= \sum_Z q(Z)\log\left(\frac{p(\mathbf{X}|\theta)p(Z|\mathbf{X},\theta)q(Z)}{p(Z|\mathbf{X},\theta)q(Z)}\right) \\
&= \underbrace{\sum_Z q(Z)\log\left(\frac{p(\mathbf{X},Z|\theta)}{q(Z)}\right)}_{:=\mathscr{L}(q,\theta)} + \underbrace{\sum_Z q(Z)\log\left(\frac{q(Z)}{p(Z|\mathbf{X},\theta)}\right)}_{:=KL(q||p)}
\end{aligned}$$

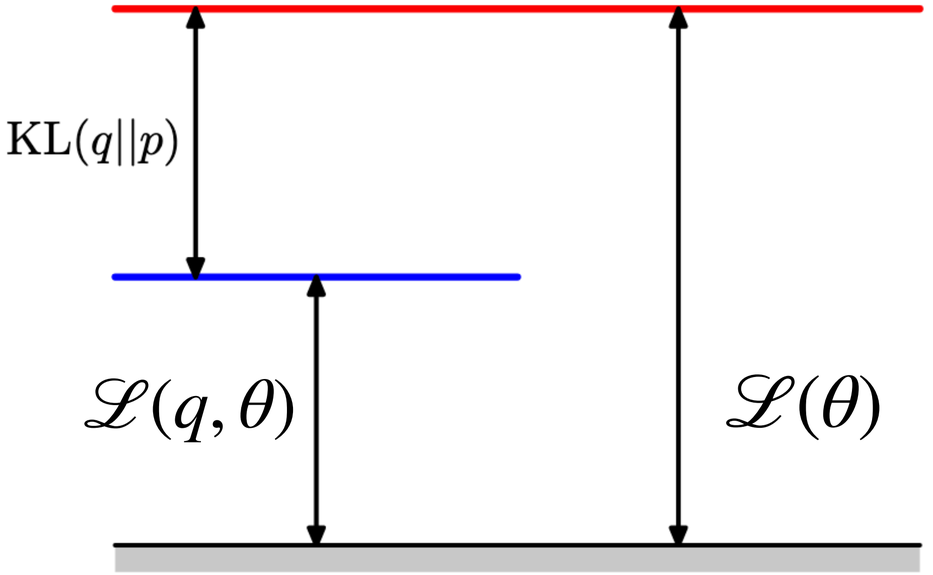
EM Algorithm in General

$$\mathscr{L}(\theta) = \mathscr{L}(q, \theta) + KL(q \,||\, p)$$

Note that for any q, p it holds that $KL(q \,||\, p) \geq 0$, with equality iff $q = p$.

Therefore $\forall q, \quad \mathscr{L}(\theta) \geq \mathscr{L}(q, \theta)$

with equality iff $q(Z) = p(Z \,|\, \mathbf{X}, \theta)$



We have introduced an auxiliary function $\mathscr{L}(q, \theta)$ that is always below the $\mathscr{L}(\theta)$

EM Algorithm as Maximizing $\mathscr{L}(q, \theta)$

The EM algorithm is a coordinate ascent algorithm on the function $\mathscr{L}(q, \theta)$

E-step

$$q^{(t+1)} = \arg\max_{q} \mathscr{L}(q, \theta^{(t)})$$

M-step

$$\theta^{(t+1)} = \arg\max_{\theta} \mathscr{L}(q^{(t+1)}, \theta)$$

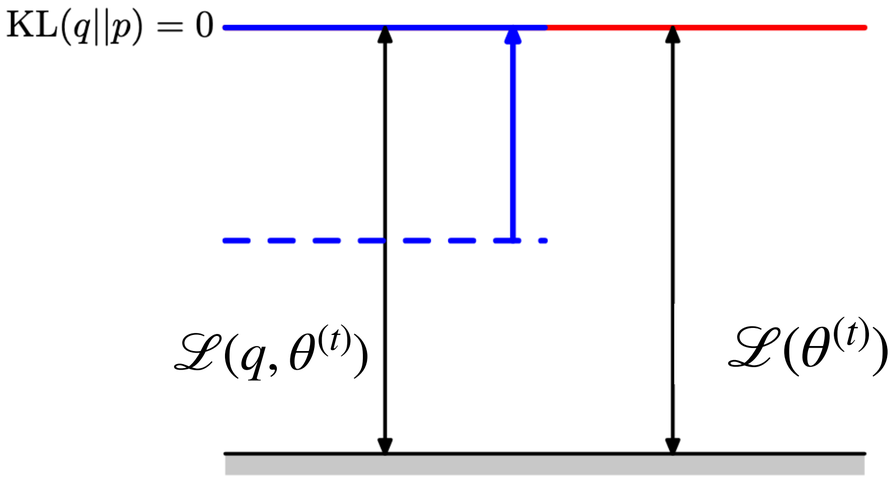
EM Algorithm as Maximizing $\mathscr{L}(q, \theta)$

The E-step of EM maximizes the lower bound $\mathscr{L}(q, \theta^{(t)})$ with respect to q, while keeping $\theta^{(t)}$ fixed.

$$\mathscr{L}(q, \theta^{(t)}) = \log p(\mathbf{X} \mid \theta^{(t)}) - KL(q \,||\, p(Z \mid \mathbf{X}, \theta^{(t)}))$$

Independent of q

Always non-negative and 0 iff $q = p(Z \mid \mathbf{X}, \theta^{(t)})$



$$q^{(t+1)} = p(Z \mid \mathbf{X}, \theta^{(t)})$$

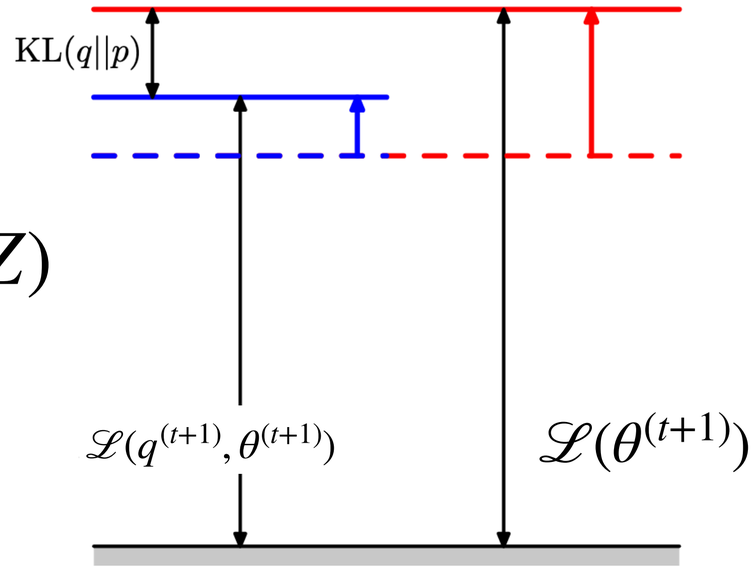
$\Longrightarrow$ **Computes a lower bound for $\mathscr{L}(\theta)$, which is tight at $\theta^{(t)}$: $\mathscr{L}(\theta^{(t)}) = \mathscr{L}(q^{(t+1)}, \theta^{(t)})$**

EM Algorithm as Maximizing $\mathscr{L}(q, \theta)$

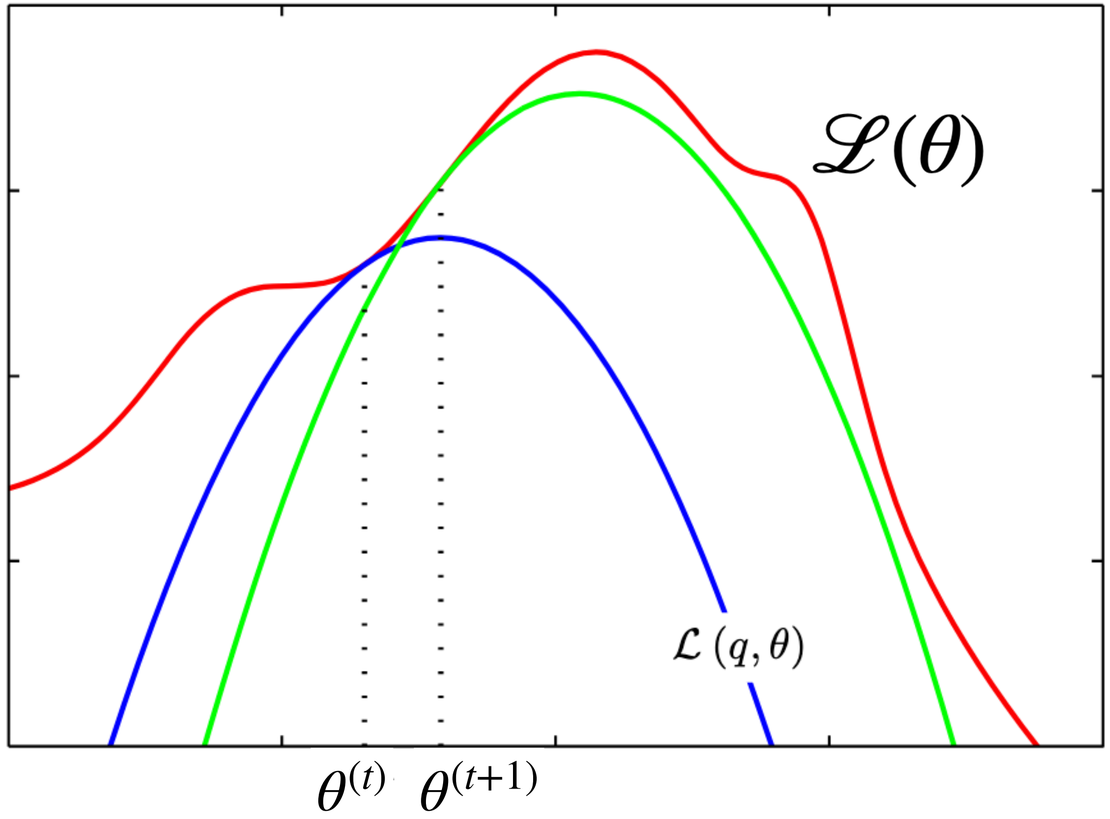
The M-step of EM maximizes the lower bound $\mathscr{L}(q^{(t+1)}, \theta)$ with respect to θ, while keeping $q^{(t+1)}$ fixed:

Note that,

$$\begin{aligned}\mathscr{L}(q, \theta) &= \sum_Z q(Z) \log \frac{p(\mathbf{X}, Z \mid \theta)}{q(Z)} \\ &= \sum_Z q(Z) \log p(\mathbf{X}, Z \mid \theta) - \sum_Z q(Z) \log q(Z) \\ &= \underbrace{\mathbb{E}_{Z \sim q}[\log p(\mathbf{X}, Z \mid \theta)]}_{\text{Maximize the expected complete log likelihood}} + \underbrace{H(q)}_{\text{Independent of } \theta}\end{aligned}$$



EM as an Algorithm Maximizing $\mathscr{L}(q, \theta)$



The EM algorithm involves alternately computing a lower bound on the log likelihood for the current parameter values (E-step) and then maximizing this bound to obtain the new parameter values (M-step).

The EM Recipe

Goal: Maximize the incomplete likelihood $\mathscr{L}(\theta) = \log\big(p(\mathbf{X}\,|\,\theta)\big)$

How: Maximize its lower bound $\mathscr{L}(q,\theta) = \log\left(\frac{p(\mathbf{X}, Z\,|\,\theta)}{q(Z)}\right)$

1. E-step:
   - Compute $p(Z\,|\,\mathbf{X}, \theta^{(t)})$ (corresponding to $q^{(t+1)} = \arg\max_q \mathscr{L}(q, \theta^{(t)})$)
   - Write the complete likelihood $\log p(\mathbf{X}, Z\,|\,\theta)$
   - Calculate the conditional expected value of the complete log likelihood $\mathbb{E}_Z[\log p(\mathbf{X}, Z\,|\,\theta)\,|\,\mathbf{X}, \theta^{(t)}]$
2. M-step: Maximize $\theta^{(t+1)} = \arg\max_\theta \mathbb{E}_Z[\log p(\mathbf{X}, Z\,|\,\theta)\,|\,\mathbf{X}, \theta^{(t)}]$

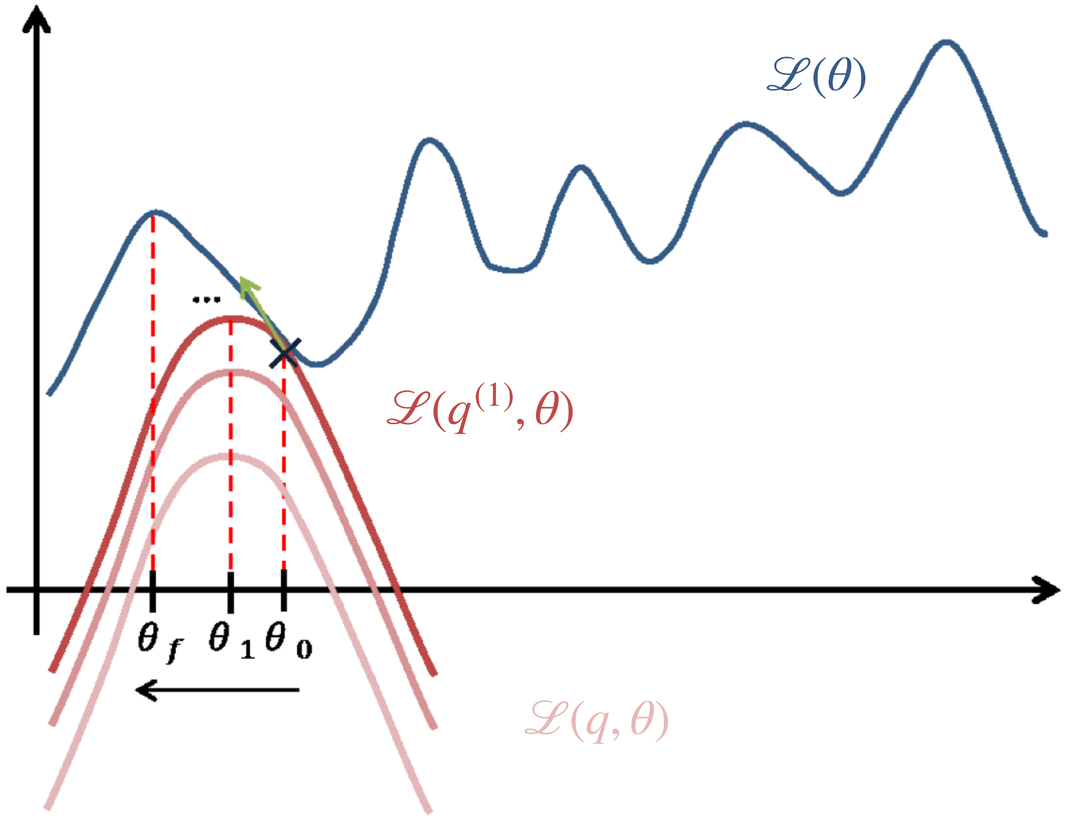
(corresponding to $\theta^{(t+1)} = \arg\max_\theta \mathscr{L}(q^{(t+1)}, \theta)$)

EM convergence properties

- It is an ascent algorithm

$$\mathscr{L}(\theta^{(t)}) \leq \mathscr{L}(\theta^{(t+1)})$$

- The sequence of log-likelihoods converges (under suitable conditions)
- It does not converge to a global maximum but rather to a stationary point (non-convex problem)
- As with K-means, we reiterate the process to increase confidence and keep the one with the highest likelihood



Practicalities: Selecting K

When K is too large ($K = N$), placing one Gaussian at each data point ($\mu_k = x_k$), shrinking variance ($\Sigma_n \to 0_D$) maximizes the log-likelihood

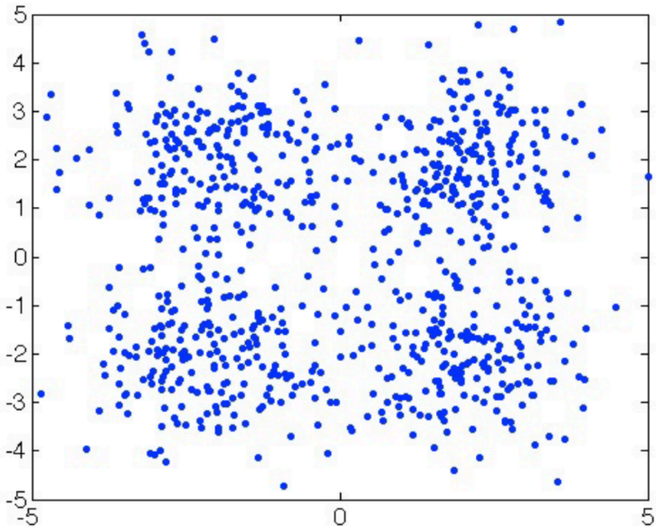
$$\mathscr{L}(\theta) = -\sum_{n=1}^{N} \log((2\pi)^{D/2} |\Sigma_n|^{1/2}) - \log\left(\frac{1}{2}(x_n - \mu_n)^T \Sigma_n^{-1}(x_n - \mu_n)\right) \to \infty$$

as $\Sigma_n \to 0 I_d$ and $\mu_n = x_n$

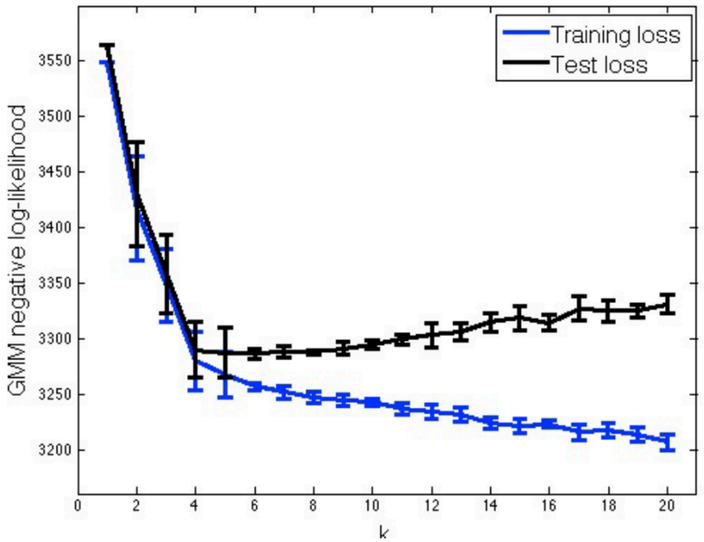
Overfitting results in poor test set log-likelihood!

Selecting K: Using Cross-Validation

Cross-validation selects K by maximizing the log-likelihood on validation dataset
This method was not possible for K-means



Data points for GMM clustering



Training vs. test log-likelihood as K varies

Cross-validation identifies the optimal K by maximizing test log-likelihood

Practicalities: Initialization

For weights:

- Typically use uniform distribution

For means:

- Randomly initialize
- Use K-means++

For variances:

- Initialize as spherical, e.g., according to empirical variance in the data

$$\Sigma_1 = \ldots = \Sigma_D = \begin{bmatrix} \sigma_1^2 & & \\ & \sigma_2^2 & \\ & & \sigma_K^2 \end{bmatrix}, \quad \sigma_d = \frac{1}{N}\sum_{n=1}^{N}(x_n[j] - \bar{x}_j)^2, \quad \bar{x}_j = \frac{1}{N}\sum_{n=1}^{N} x_n[j]$$

Recap

- **Gaussian Mixture Models:**
  - Represents the data as a mixture of Gaussian distributions
  - Useful for clustering, density estimation, and anomaly detection
  - Offers more flexibility in cluster shape and size compared to K-means
- **Expectation Maximization Algorithm:**
  - Goal: Solve models with latent (hidden) variables (e.g. GMMs)
  - Process: Alternate between the E and M steps until convergence
    - E-step: Calculate the posterior of the latent variables == Compute expected complete log likelihood == Maximize $\mathscr{L}(q, \theta)$ over q
    - M-step: Maximize the expected complete log likelihood == Maximize $\mathscr{L}(q, \theta)$ over θ