SME3006 Machine Learning – 2025 Fall

Gaussian Mixture Model and Expectation Maximization

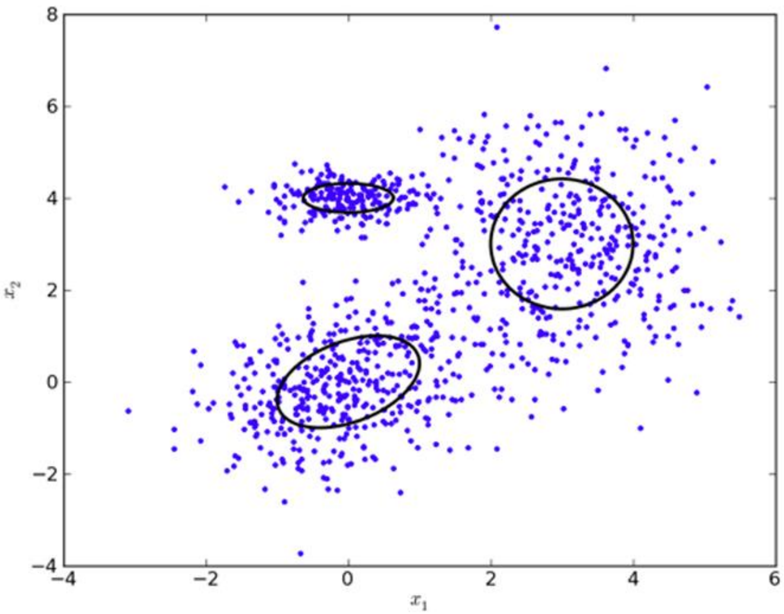
INHA UNIVERSITY

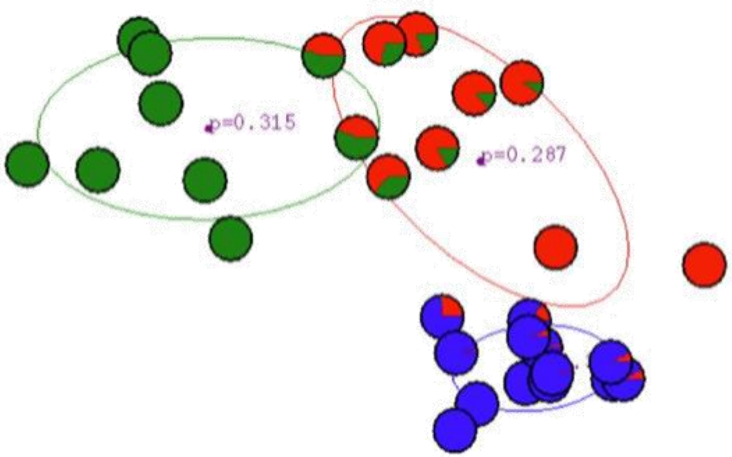
Overview

- Motivation and background
  - soft clustering
- GMM
- EM
- Jensen’s inequality, MLE, convexity
- Extensions
  - HMM
  - Factor analysis
  - Mixture of experts

Clustering

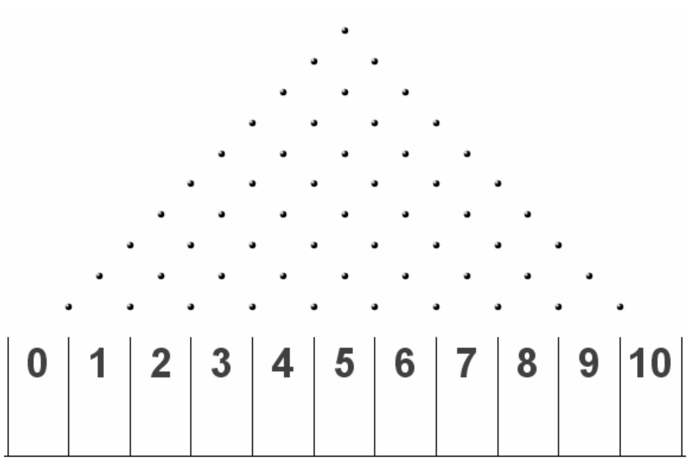
- Hard clustering can be difficult
  - each object belongs to only one cluster
  - K-means, DBSCAN, Hierarchical clustering
- Soft clustering
  - probability that an object belongs to a cluster

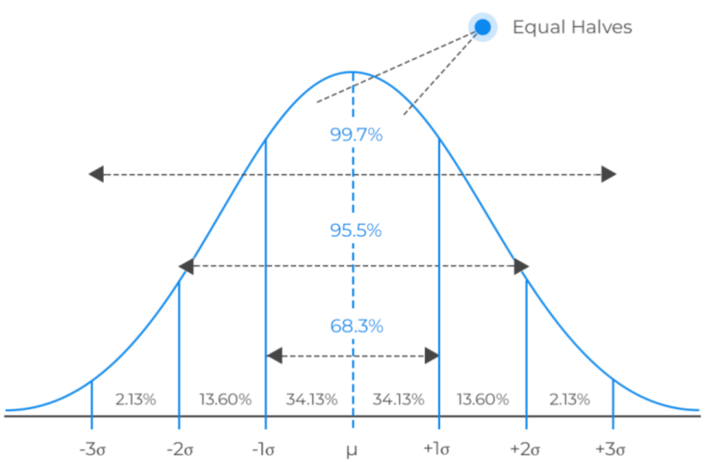




Gaussian distribution

> Perhaps the most used distribution in all of science
  - Central limit theorem: things that are the result of the addition of small effects tend to become Gaussian



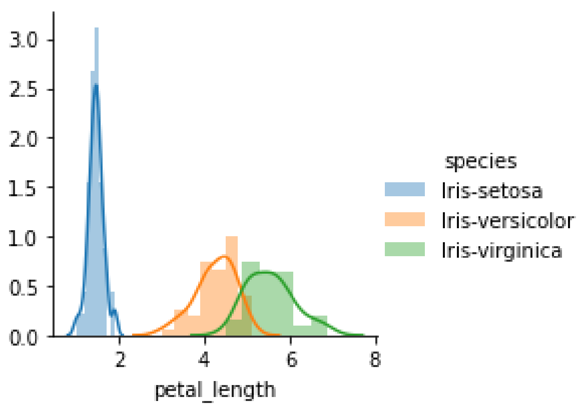


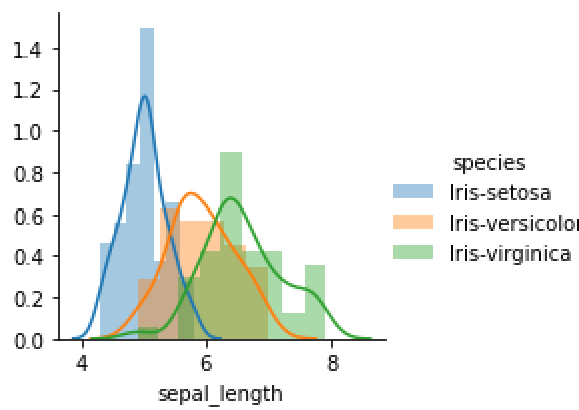
- $p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, high\ dimensional: p(x) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^\top\Sigma^{-1}(x-\mu)}$

Gaussian distribution

- Most distribution in nature follow a Gaussian distribution
  - Example: iris dataset
    - sepal length, width, petal length, width

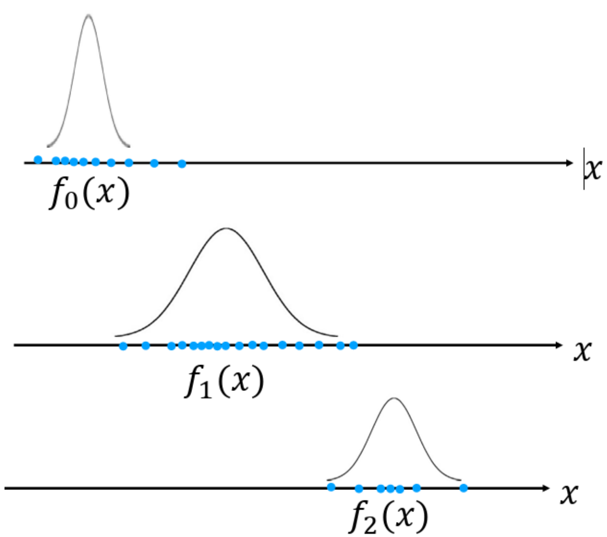
Iris Versicolor **Iris Setosa** **Iris Virginica**



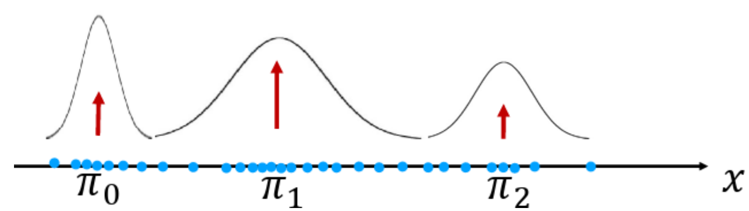


Mixture model

- Each element would be probably Gaussian, but there might be several Gaussian distributions
- How can we express the distribution of mixture of Gaussian?
  - mixture model is the weighted sum of a number of pdfs
  - $p(x) = \pi_0 f_0(x) + \pi_1 f_1(x) + \cdots + \pi_k f_k(x), \quad where \ \sum_{i=0}^{k} \pi_i = 1$



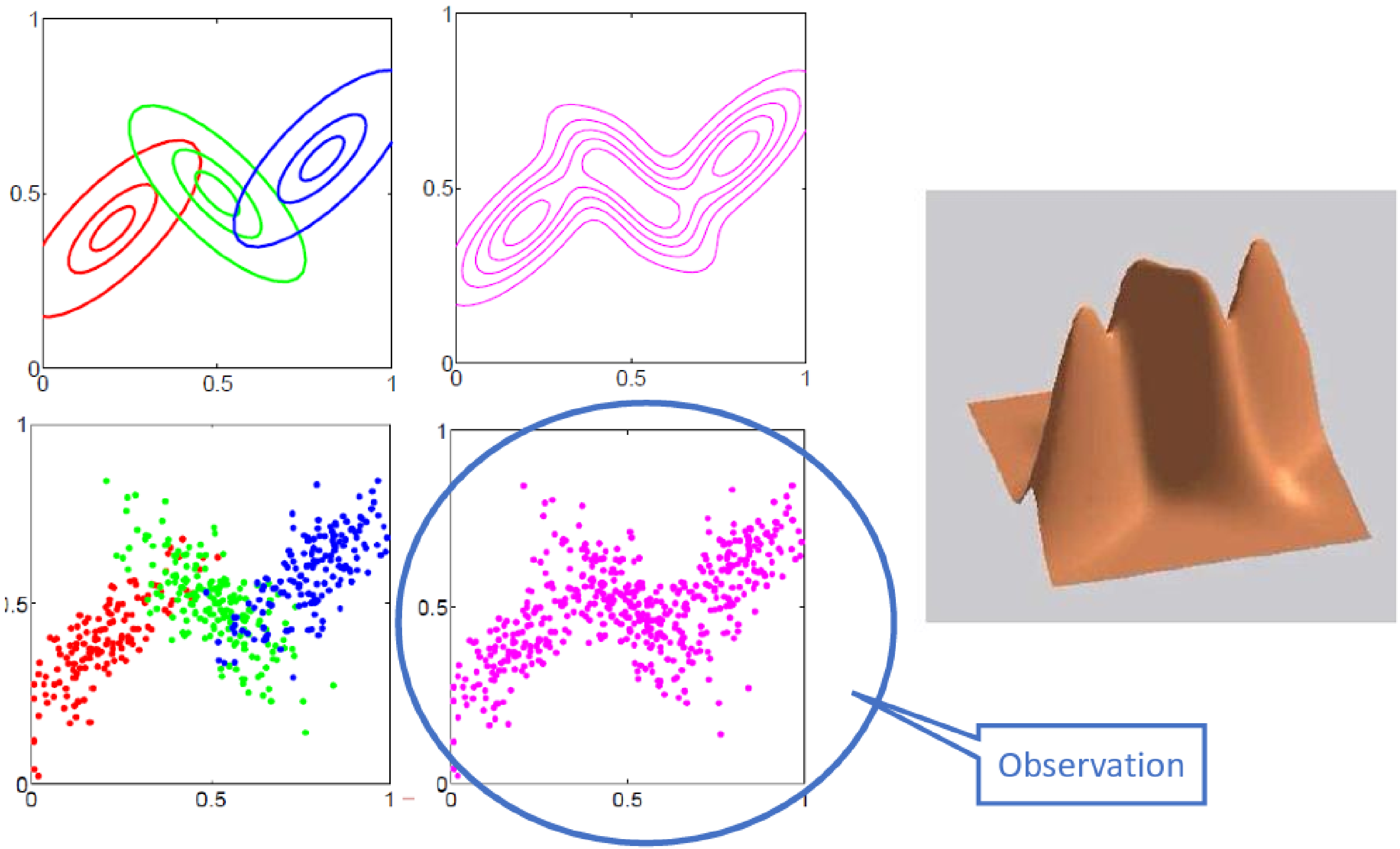
single Gaussian distribution



Mixture of Gaussian distributions

Mixture model

> Examples



Gaussian mixture model

> From the dataset, we want to find a Gaussian mixture distributions

- dataset $X = \{x_1, x_2, \ldots, x_n\}$
- find the best θ that maximizes the probability of $p(X|\theta)$ w.r.t $\theta = \{w, \mu, \Sigma\}$
  - cluster probability w, cluster mean μ, cluster covariance Σ

- maximal likelihood estimator (MLE)

$$\theta^* = \arg\max_{\theta} p(X|\theta) = \arg\max_{\theta} \prod_{i=1}^{n} p(x_i|\theta)$$

- recall: Bayesian

$$p(belief|data) = \frac{p(data|belief)p(belief)}{p(data)}$$

Gaussian mixture model

- For data points x_i, the probability is a mixture of Gaussian
  - this is a Gaussian mixture model (GMM)

$$p(x_i|\theta) = \sum_{j=1}^{K} w_j \, \mathcal{N}\left(x_i|\mu_j, \Sigma_j\right), \;\; where \; \sum_{j=1}^{K} w_j = 1$$

  - it is an universal approximator of densities (if we have enough Gaussians)

- Introduce latent variable
  - z_i is the Gaussian cluster ID indicates which Gaussian x_i comes from
  - joint distribution $p(x,z) = p(x|z)p(z)$
  - $p(z_i = j) = w_j \quad \rightarrow$ prior
  - $p(x_i|z_i = j) = \mathcal{N}(\mu_j, \Sigma_j)$
  - $p(x_i|\theta) = \sum_{j=1}^{K} p(z_i = j) p(x_i|z_i = j)$

Gaussian mixture model

> MLE

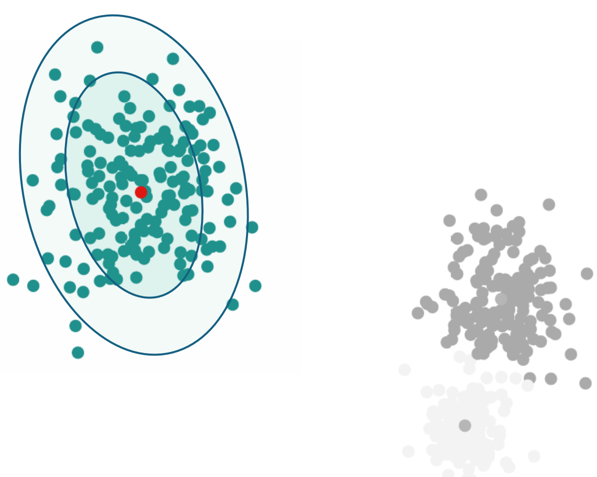
- $\theta^* = \arg\max_{\theta} p(X|\theta) = \arg\max_{\theta} \prod_{i=1}^{n} p(x_i|\theta)$
  $= \arg\max_{\theta} \log \prod_{i=1}^{n} p(x_i|\theta) = \arg\max_{\theta} \sum_{i=1}^{n} \log p(x_i|\theta)$
- a log-likelihood function: $l(\theta) = \sum_{i=1}^{n} \log p(x_i|\theta)$
- $l(\theta) = \sum_{i=1}^{n} \log p(x_i|\theta) = \sum_{i=1}^{n} \log \sum_{j=1}^{K} w_j \mathcal{N}\left(x_i|\mu_j, \Sigma_j\right)$

- Issues
  - singularities: arbitrarily large likelihood when a Gaussian explains a single point (spike Gaussian → infinite likelihood)
  - identifiability: solution is invariant to permutations ($K!$ equivalent solutions)
  - non-convex

- How can we optimize this?
  - we should consider constraints: $\sum_{j=1}^{K} w_j = 1$ and Σ_j is positive definite

Gaussian mixture model

- MLE: $l(\theta) = \sum_{i=1}^{n} \log \sum_{j=1}^{K} w_j \, \mathcal{N}\left(x_i | \mu_j, \Sigma_j\right)$
  - If we knew z_i for every x_i (cluster assignment of each point), the problem becomes easy
  - $l(\theta) = \sum_{i=1}^{n} \log w_{z_i} \mathcal{N}\left(x_i | \mu_{z_i}, \Sigma_{z_i}\right) = \sum_{i=1}^{n} \log \mathcal{N}\left(x_i | \mu_{z_i}, \Sigma_{z_i}\right) + \log w_{z_i}$
  - The procedure is the same as fitting a single Gaussian distribution
    - w_{z_i} will converge to the proportion of points in that cluster $w_{z_i} \to \frac{n_{z_i}}{n}$
    - μ_{z_i}, Σ_{z_i} are mean and covariance of data points in that cluster



Gaussian mixture model

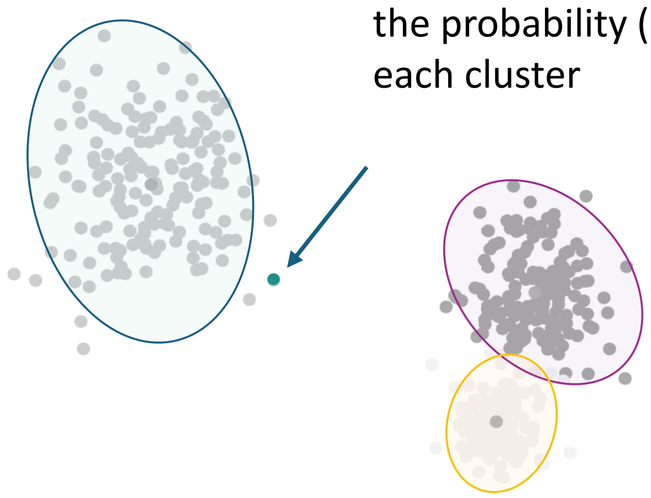
> We previously assumed that the cluster assignments were known

> How can we evaluate the probability of each data point belongs to each cluster

- Given the parameter $\theta = \{w_j, \mu_j, \Sigma_j\}$, the posterior distribution of each latent variable z_i can be inferred as

$$\gamma_{j,i} = p(z_i = j | x_i; \theta) = \frac{p(x_i, z_i = j|\theta)}{p(x_i|\theta)} = \frac{w_j \mathcal{N}(x_i|\mu_j, \Sigma_j)}{\sum_{k=1}^{K} w_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}$$

- we call $\gamma_{j,i}$ the responsibility - the responsibility that cluster j takes for data x_i



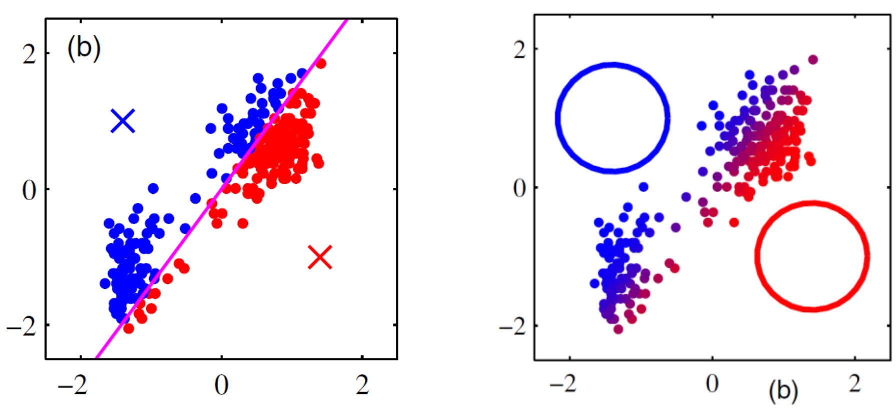
the probability (responsibility) that a point belongs to each cluster

Expectation maximization

- Optimization uses the expectation maximization (EM)
  - iterate two steps
    - E-step: compute the posterior probability over z given our current model i.e., how much do we think each Gaussian generates each datapoint
    - M-step: assuming that the data really was generated this way, change the parameters of each Gaussian to maximize the MLE

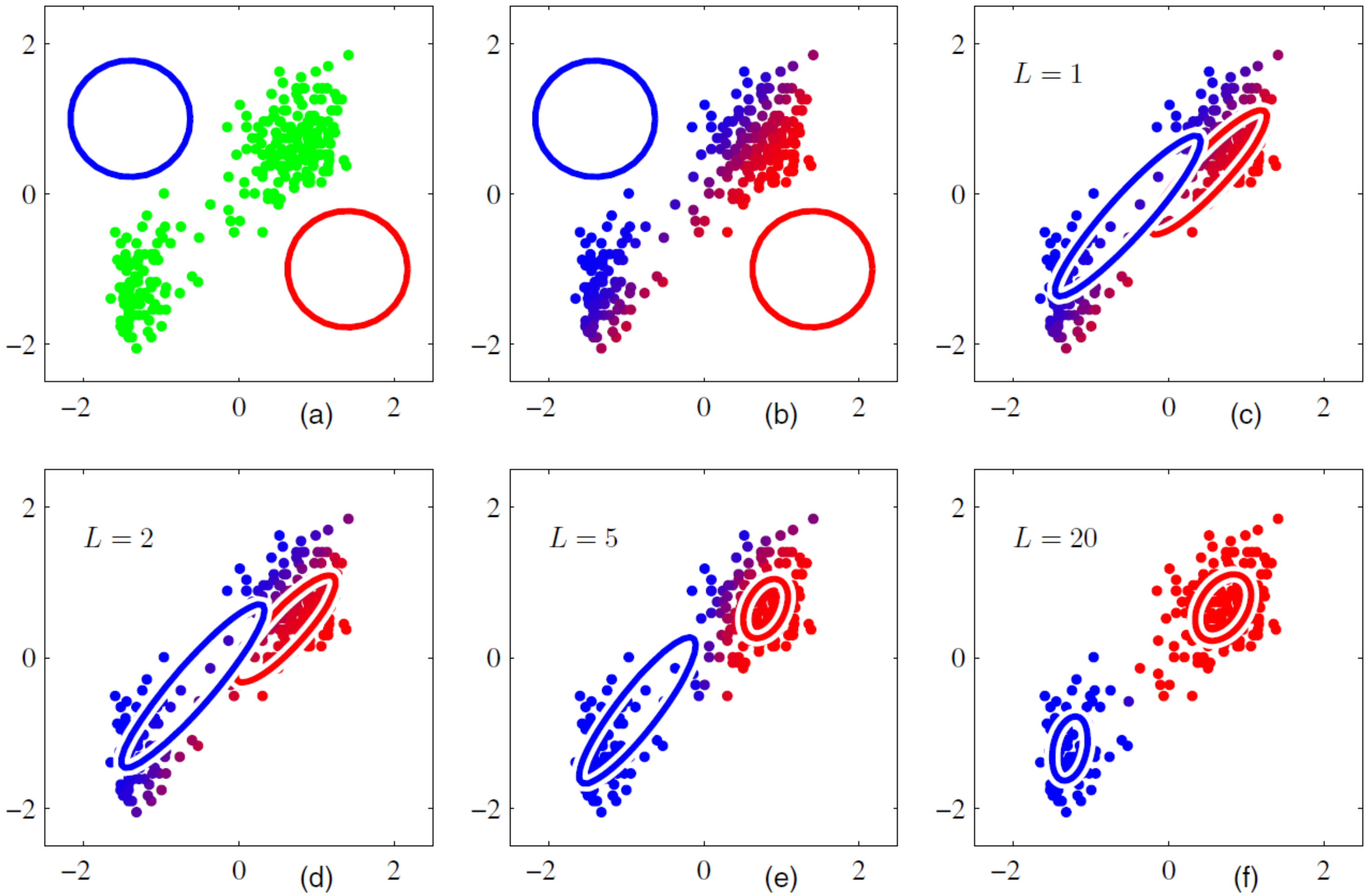
Expectation maximization

- Compared to K-means algorithm
  - K-means
    - Assignment: assign each data point to the closest cluster
    - Refitting: move each cluster center to the center of gravity of the data assigned
  - EM: soft version of K-means with fixed priors and covariance
    - E-step: compute the posterior probability over z given our current model
    - M-step: maximize the probability that it would generate the data
    - each center moved by weight means of the data (in K-means, weights are 0 or 1)



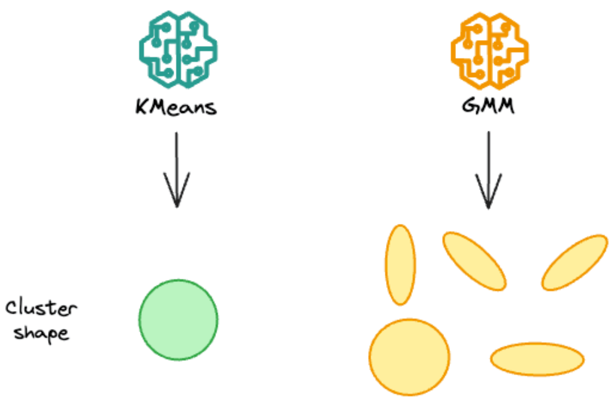
Expectation maximization

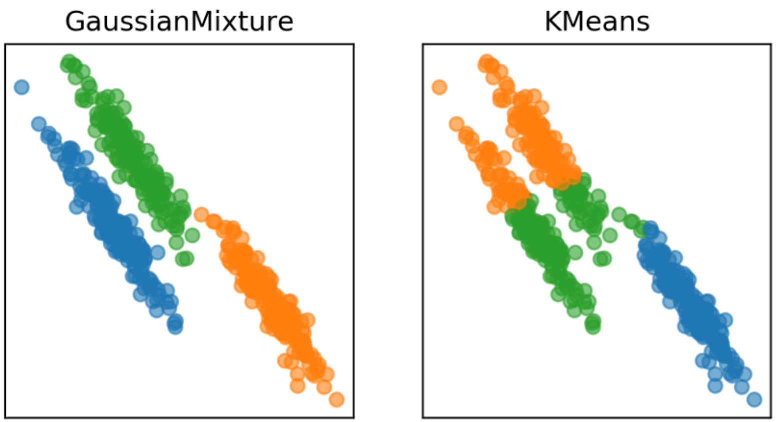
> Procedures

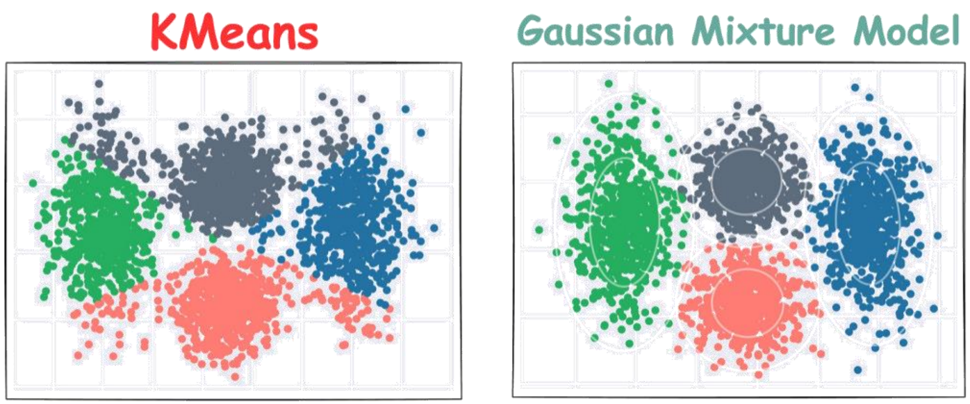


Expectation maximization

> Performance







Expectation maximization

- Issues
  - the process will be converged to local optimum
  - very sensitive to initial conditions
  - required predefined number of Gaussians
    - we may use information-theoretic criteria to obtain the optimal number
    - Minimal description length (MDL) or other criteria such as AIC, BIC, MML, …

General latent variable model

- Two sets of random variables z, x
  - z consists of unobserved hidden variables
  - x consists of observed variables
  - joint probability model parameterized by θ, $\mathrm{p}(\mathrm{X}, \mathrm{z}|\theta)$
  - we call $p(X)$ the marginal likelihood $p(X) = \sum_z p(X, z)$

  - Def) a latent variable model is a probability model for which certain variables are never observed
    - GMM is a latent variable model

  - Learning problem: given incomplete dataset $D = X = \{x_1, x_2, \dots, x_n\}$
    find $\theta^* = \arg\max_\theta p(X|\theta)$
  - Inference problem: given X, find conditional distribution over z
    $p(z_i|x_i, \theta)$

General latent variable model

- Optimizing MLE
  - We are not assuming the Gaussian distribution
  - maximize marginal log-likelihood

    $$l(\theta) = \log p(X|\theta) = \log \sum_z p(X, z|\theta) = \log \sum_z q(z) \frac{p(X,z|\theta)}{q(z)}$$

    - where $q(z)$ be any probability mass function (PMF)
    - $\sum_z q(z) = 1$
  - By Jensen's inequality,

    $$\log \sum_z q(z) \frac{p(X,z|\theta)}{q(z)} \geq \sum_z q(z) \log(\frac{p(X,z|\theta)}{q(z)})$$

    - why?

Recall: convexity

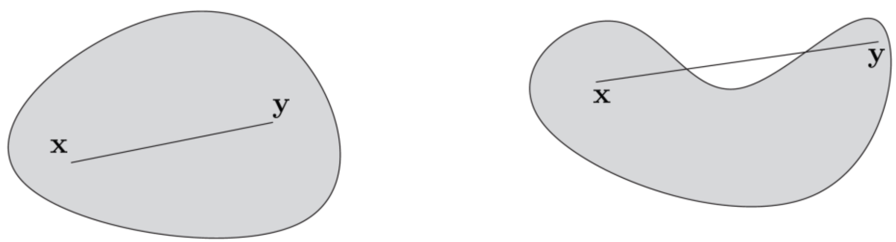
> Convex sets:

A set $C \subseteq \mathbb{R}^n$ is convex if for $x, y \in C$ and any $\alpha \in [0,1]$,

$$\alpha x + (1 - \alpha) y \in C$$

convex

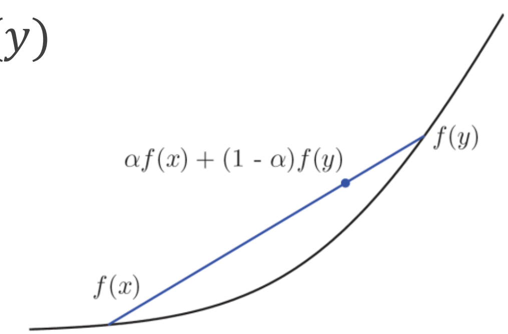
non-convex



> Convex function

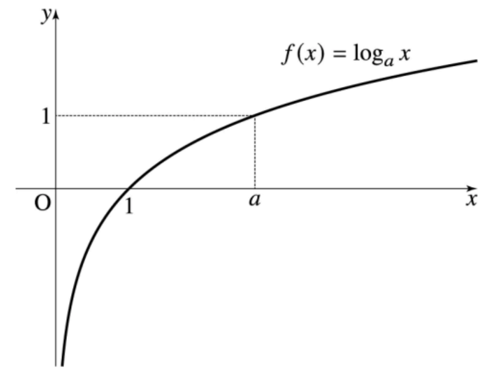
- A function $f: \mathbb{R}^n \to \mathbb{R}$ is convex if for $x, y \in dom\ f$ and any $\alpha \in [0,1]$,

$$f(\alpha x + (1 - \alpha) y) \leq \alpha f(x) + (1 - \alpha) f(y)$$

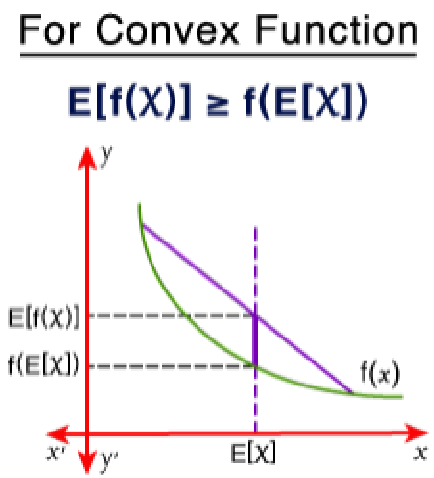


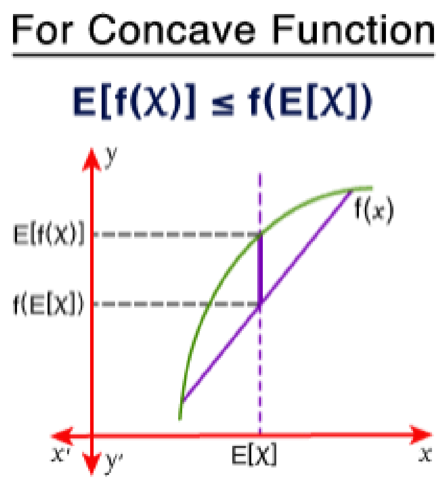
Convexity

- Convexity f$(\alpha x + (1-\alpha)y) \le \alpha f(x) + (1-\alpha) f(y)$
  - if $f''(x) \ge 0 \ \rightarrow f$ is convex
  - g is concave if and only if $-g$ is convex
  - e.g., $g(x) = \log x$ is concave



- Jensen's inequality: $\mathbb{E}[f(x)] \ge f(\mathbb{E}[x])$





General latent variable model

> Optimizing MLE

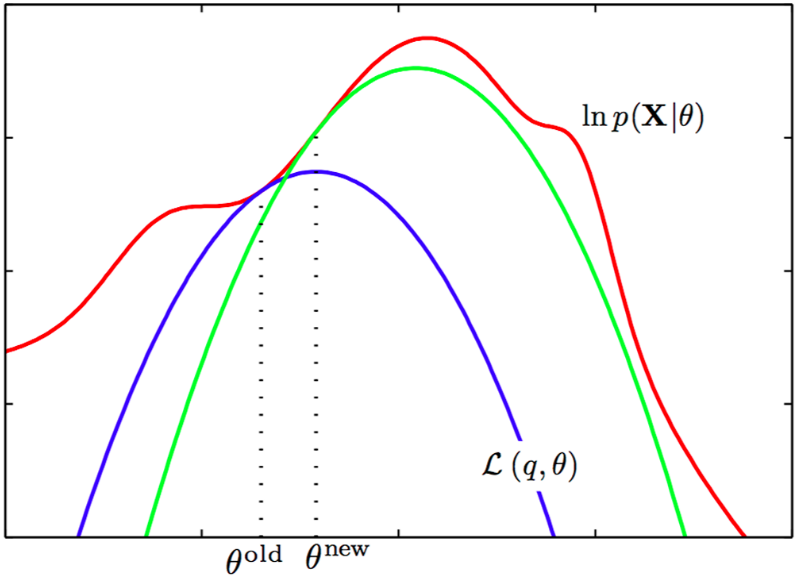
- $l(\theta) = \log p(X|\theta) = \underbrace{\log}_{f} \underbrace{\sum_z q(z)}_{\mathbb{E}} \underbrace{\frac{p(X,z|\theta)}{q(z)}}_{x} \geq \underbrace{\sum_z q(z)}_{\mathbb{E}} \underbrace{\log}_{f} \underbrace{\left(\frac{p(X,z|\theta)}{q(z)}\right)}_{[x]}$

  Jensen's inequality $f(\mathbb{E}[x]) \geq \mathbb{E}[f(x)]$

- marginal log likelihood $\log p(X|\theta)$ also called the evidence
- $\sum_z q(z) \log\left(\frac{p(X,z|\theta)}{q(z)}\right)$ is the evidence lower bound, or ELBO
- we maximize ELBO over q and θ in EM (and variational methods)

General latent variable model

- Optimizing MLE
  - $l(\theta) = \log p(X|\theta) \geq \sum_z q(z) \log\left(\frac{p(X,z|\theta)}{q(z)}\right) = \mathcal{L}(q,\theta)$
  - maximize the ELBO instead of log-likelihood ($q(z)$ is any kind of PMF)
- Algorithm
  - start at θ^{old}
  - find q giving best ELBO
    $q^* = \arg\max_q \mathcal{L}(q, \theta^{old})$
  - find θ giving best ELBO
    $\theta^{new} = \arg\max_\theta \mathcal{L}(q^*, \theta)$
  - iterate



ELBO

- Evidence lower bound (ELBO) in terms of KL divergence and entropy
  - $\mathcal{L}(q,\theta) = \sum_z q(z)\log\left(\frac{p(X,z|\theta)}{q(z)}\right)$

    $= \sum_z q(z)\log\left(\frac{p(z|X,\theta)p(X|\theta)}{q(z)}\right)$

    $= \sum_z q(z)\log\left(\frac{p(z|X,\theta)}{q(z)}\right) + \sum_z q(z)\log p(X|\theta)$

    $= -KL[q(z) \parallel p(z|X,\theta)] + \log p(X|\theta)$

  - we obtained the equation with the marginal likelihood

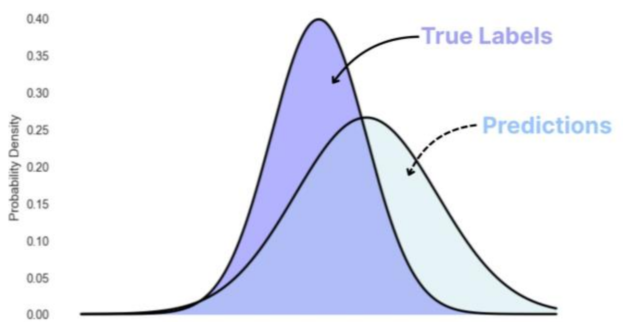
    $\log p(X|\theta) = \mathcal{L}(q,\theta) + KL[q(z) \parallel p(z|X,\theta)]$

Detour: Information theory basics

- Goal of machine learning is to extract meaningful patterns out of data, it is no surprise that there are deep connections there
- Entropy
  - $H(p) = -\sum_x p(x) \log p(x)$
  - a measure of uncertainty of random variable
    = required information amount to remove the uncertainty (coding cost)
  - X has a maximum entropy if it follows the uniform distribution
- Cross entropy
  - $H(p,q) = -\sum_x p(x) \log q(x)$
  - a measure of how well a distribution q approximates the true distribution p
    = a measure of how different two probability distributions are
  - coding cost if you assume the wrong distribution q

Detour: Information theory basics

> Kullback-Leibler divergence (KL divergence)
  - how much a probability distribution p is different from a probability distribution q
  - def: $KL[p \parallel q] = \sum_x p(x) \log \frac{p(x)}{q(x)}$
    $= \sum_x p(x) \log p(x) - \sum_x p(x) \log q(x) = -H(p) + H(p,q)$
  - penalty you pay for using q instead of p
  - KL value is always positive
  - $\sum_x p(x) \log \frac{p(x)}{q(x)} = \sum_x p(x) \left(-\log \frac{q(x)}{p(x)}\right)$
    $\geq -\log \sum_x p(x) \frac{q(x)}{p(x)} = -\log \sum_x q(x) = 0$

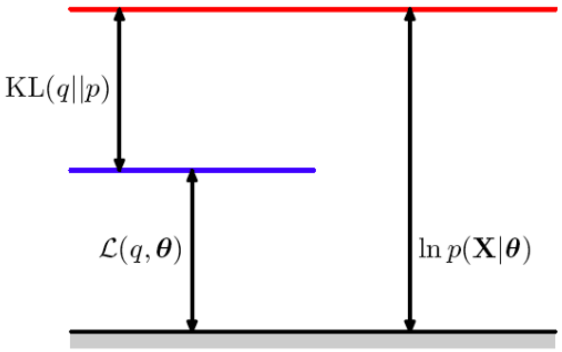


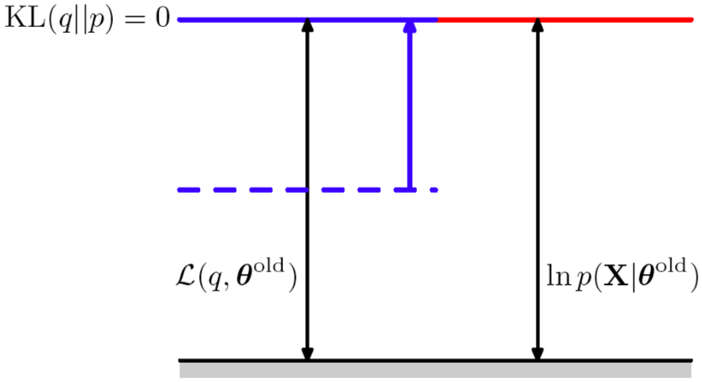
> Note
  - cross entropy and KL divergence are not symmetric
    - $H(p,q) \neq H(q,p), \quad KL[p \parallel q] \neq KL[q \parallel p]$

ELBO

> Optimizing MLE → ELBO
  - $\log p(X|\theta) \geq \mathcal{L}(q,\theta) = \underbrace{-KL[q(z) \parallel p(z|X,\theta)]}_{\geq 0} + \underbrace{\log p(X|\theta)}_{\text{not related to } q}$
  - when θ is fixed
  - best lower bound when $KL[q(z) \parallel p(z|X,\theta)] = 0 \ \rightarrow q(z) = p(z|X,\theta)$
  - lower bound is tight at θ^{old} when we take $q(z) = p(z|X,\theta^{old})$

$\mathcal{L}(q,\Theta) \leq \ln p(\mathbf{X}|\Theta)$

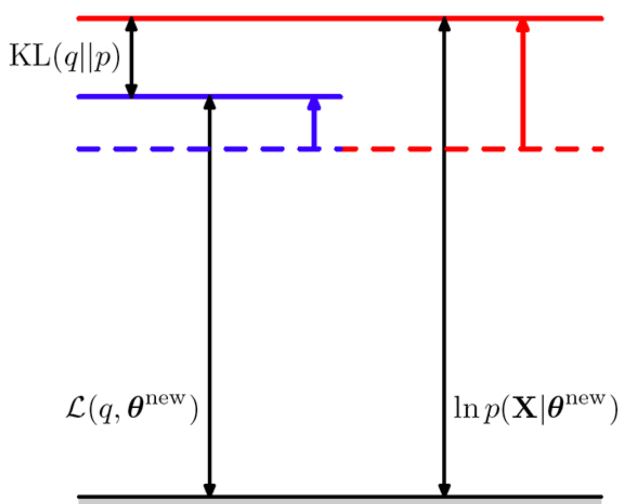




ELBO

> Optimizing MLE → ELBO

- $\log p(X|\theta) \geq \mathcal{L}(q,\theta) = -\underbrace{KL[q(z) \parallel p(z|X,\theta)]}_{\geq 0} + \underbrace{\log p(X|\theta)}_{\text{not related to } q}$
- when q is fixed
- ELBO is maximized with respect to the parameter θ and KL divergence is nonnegative
- log-likelihood is increased by at least as much as the ELBO does



General EM

- General EM algorithm
  - choose initial θ^{old}
  - expectation step
    - $q^*(z) = p(z|X, \theta^{old})$
    - $J(\theta) = \mathcal{L}(q^*, \theta) = \sum_z q^*(z) \log\left(\frac{p(X,z|\theta)}{q^*(z)}\right)$
  - maximization step
    - $\theta^{new} = \arg\max_\theta J(\theta)$

- If we found a global maximum of ELBO $\mathcal{L}(q^*, \theta^*)$,
  then θ^* is a global maximum of log-likelihood $\log p(z|X, \theta^*)$

ELBO

- For EM algorithm, each step can be too hard to do in practice
- Generalized EM algorithm
  - choose initial θ^{old}
  - expectation step
    - $q^*(z) = \arg\min KL[q(z) \parallel p(z|X, \theta^{old})]$
    - $J(\theta) = \mathcal{L}(q^*, \theta) = \sum_z q^*(z) \log\left(\frac{p(X,z|\theta)}{q^*(z)}\right)$
  - maximization step
    - find any θ^{new} for which $J(\theta^{new}) > J(\theta^{old})$
  - we still get monotonically increasing likelihood

GMM and EM

- GMM
  - a probabilistic view of clustering – each cluster is a different Gaussian
  - can replace Gaussian with other distributions
  - optimization is done using the EM algorithm

- EM
  - a general algorithm for optimizing many latent variable models
  - iteratively computes a lower bound then optimizes it
  - converges but maybe to a local minima
  - requires computation of $p(z|X,\theta)$, not possible for complicated models
    - solution: variational inference

Reference

> GMM and EM

- https://davidrosenberg.github.io/mlcourse/Archive/2017Fall/Lectures/13b.mixture-models.pdf
- https://davidrosenberg.github.io/mlcourse/Archive/2017Fall/Lectures/13c.EM-algorithm.pdf
- https://www.cs.toronto.edu/~jlucas/teaching/csc411/lectures/lec15_16_handout.pdf
- https://shuaili8.github.io/Teaching/VE445/L12_gmm.pdf
- https://www.davidinouye.com/course/ece57000-fall-2021/lectures/gaussian-mixtures.pdf
- https://nakulgopalan.github.io/cs4641/course/20-gaussian-mixture-model.pdf