A Refresher on Probabilities, K-means Clustering and Gaussian Mixture Models

Laura Manduchi, Dario Pavllo

ETH Zürich

4–5 April 2019

Overview

A Refresher on Probabilities

K-means Clustering

Gaussian mixture models

- Why EM works?

Sample spaces and probabilities

- A *sample space* Ω is the set of outcomes of a random experiment.
- Subsets $A \subseteq \Omega$ are called *events*.
- For example, consider the experiment of tossing a fair coin twice.
  - Sample space: $\Omega = \{HH, HT, TH, TT\}$
  - Event of at least one "head" occurring: $A = \{HH, HT, TH\}$.
- A *probability distribution* is a function that assigns a real number $\Pr[A]$ to each event $A \subseteq \Omega$.

Random variables

- Usually, we do not deal directly with sample spaces. Instead, we define *random variables* and probability distributions on those.
- A random variable is a function $X : \Omega \to \mathbb{R}$.
- For example, if $X :=$ "the number of heads in two coin tosses", then

$$X(HH) = 2$$
$$X(HT) = 1$$
$$X(TH) = 1$$
$$X(TT) = 0$$

Probabilities of random variables

- If we denote by $\mathcal{X}$ the set of values a random variable X can take, we can define probabilities directly on $\mathcal{X}$.
- In the above example, $\mathcal{X} = \{0, 1, 2\}$ and we define

$$\begin{aligned} \Pr[X = 0] &:= \Pr[\{TT\}] \\ \Pr[X = 1] &:= \Pr[\{HT, TH\}] \\ \Pr[X = 2] &:= \Pr[\{HH\}] \end{aligned}$$

- In practice, we often completely forget about the sample space and work only with random variables.

Discrete random variables

- X is called a *discrete random variable* if $\mathcal{X}$ is a finite or countably infinite set.

- Examples:
  - $\mathcal{X} = \{0, 1\}$
  - $\mathcal{X} = \mathbb{N}$
  - $\mathcal{X} = \mathbb{N}^d$

- The corresponding probability distribution

$$P(x) := \Pr[X = x]$$

  is called a *probability mass function*.

- Non-negativity: $P(x) \geq 0, \ \forall x \in \mathcal{X}$

- Normalization: $\sum_{x \in \mathcal{X}} P(x) = 1$

Continuous random variables

- X is called a *continuous random variable* if $\mathcal{X}$ is an uncountably infinite set.
- Examples:
  - $\mathcal{X} = [0, 1]$
  - $\mathcal{X} = \mathbb{R}$
  - $\mathcal{X} = \mathbb{R}^d$
- The corresponding probability distribution $p(x)$ is called a *probability density function*.
- Non-negativity: $p(x) \geq 0, \ \forall x \in \mathcal{X}$
- Normalization: $\int_{\mathcal{X}} p(x)dx = 1$

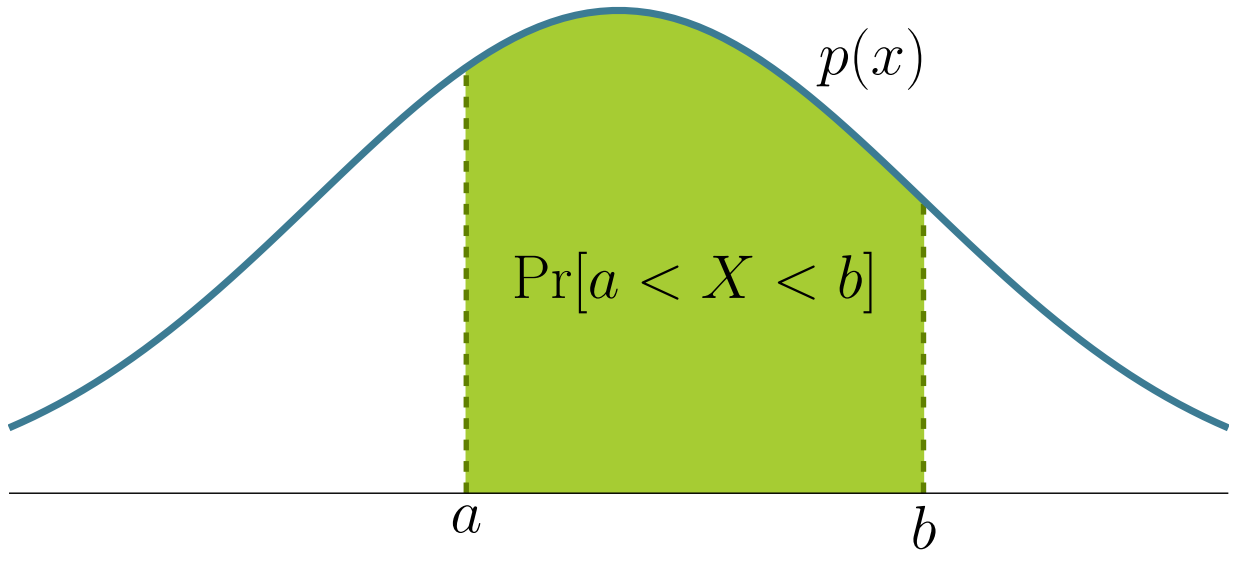
The meaning of density

- *Important:* For continuous random variables

$$p(x) \neq \Pr[X = x] = 0$$

- To acquire a probability, we have to integrate p over the desired set

$$\Pr[a < X < b] = \int_a^b p(x)dx$$



Joint distributions

- For two random variables $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, their *joint distribution* is defined as

$$P(x, y) := \Pr[X = x, Y = y]$$

- Non-negativity: $P(x, y) \geq 0$
- Normalization: $\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P(x, y) = 1$
- For example, assume we throw two fair six-sided dice and define $X :=$ "the number on the first die" and $Y :=$ "the number on the second die".
  - $\mathcal{X} = \mathcal{Y} = \{1, 2, 3, 4, 5, 6\}$
  - $P(6, 6) = \Pr[X = 6, Y = 6] = \frac{1}{36}$

Marginal and conditional distributions

Let $P(x, y)$ be a joint distribution of random variables X and Y.

- The *marginal distribution* of X is defined as

$$P(x) := \Pr[X = x] := \sum_{y \in \mathcal{Y}} P(x, y)$$

- The *conditional distribution* of X given that Y has a known value y is defined as

$$\begin{aligned} P(x|y) &:= \Pr[X = x | Y = y] \\ &:= \frac{P(x, y)}{P(y)} \qquad \text{(defined if } P(y) > 0\text{)} \end{aligned}$$

- Note that for any fixed y, $P(x|y)$ is a distribution over x, i.e.

$$\sum_{x \in \mathcal{X}} P(x|y) = 1, \ \forall y \in \mathcal{Y}$$

The chain rule

- By definition of conditional distributions, we can *always* write a joint distribution of X and Y as a product of conditionals:

$$P(x, y) = P(x|y)P(y)$$

- We can do the same for an arbitrary number of random variables $X_1, \ldots, X_n$:

$$P(x_1, \ldots, x_n) = P(x_1|x_2, \ldots, x_n) \ldots P(x_{n-1}|x_n)P(x_n)$$

- Consistency of marginals and conditionals:

$$\begin{aligned} \sum_{y \in \mathcal{Y}} P(x, y) &= \sum_{y \in \mathcal{Y}} P(y|x)P(x) && \text{(chain rule)} \\ &= P(x) \sum_{y \in \mathcal{Y}} P(y|x) \\ &= P(x) && \text{(normalization)} \end{aligned}$$

Bayes' rule

- For two random variables X and Y, by definition of the conditional distribution of X given Y:

$$P(x|y) = \frac{P(x,y)}{P(y)}$$

- Also, by the chain rule:

$$P(x,y) = P(y|x)P(x)$$

- Combining the above we get Bayes' rule:

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

Independence

- Two random variables X and Y are called *independent*, if knowing the value of X does not give any additional information about the distribution of Y (and vice versa):

$$\begin{aligned} & P(x|y) = P(x) \\ \Leftrightarrow\ & P(y|x) = P(y) \end{aligned}$$

- Equivalently, X and Y are independent if their joint distribution factorizes:

$$P(x, y) = P(x|y)P(y) = P(x)P(y)$$

IID

- IID := *I*ndependent and *I*dentically *D*istributed
- Random variables $X_1, ..., X_n$ are called IID if
  - Each of them has the same (marginal) distribution
  - They are mutually independent
- Note that if $X_1, ..., X_n$ are IID, then

$$\begin{aligned} P(x_1, ..., x_n) &= P(x_1)...P(x_n) \\ &= \prod_{i=1}^{n} P(x_i) \end{aligned}$$

Expectation

- The *expectation* of a random variable X is defined as

$$\mu_X := \mathrm{E}[X] := \sum_{x \in \mathcal{X}} xP(x)$$

- Note that the expectation $\mathrm{E}[X]$ is *not* the same as the most likely value $\max_{x \in \mathcal{X}} P(x)$.

- Can also be defined for a function f of X:

$$\mathrm{E}[f(X)] := \sum_{x \in \mathcal{X}} f(x)P(x)$$

Variance

- The *variance* of a random variable X is defined as

$$\mathrm{Var}[X] := \mathrm{E}[(X - \mu_X)^2] := \sum_{x \in \mathcal{X}} (x - \mu_X)^2 P(x)$$

- $\mathrm{Var}[X] \geq 0$

- The *standard deviation* of X is defined as

$$\sigma_X := \sqrt{\mathrm{Var}[X]}$$

Multidimensional moments

Let $\boldsymbol{X} = (X_1, \ldots, X_n)$ be a vector of random variables.

- The expectation of $\boldsymbol{X}$ is defined as

$$\mathrm{E}[\boldsymbol{X}] := (\mathrm{E}[X_1], \ldots, \mathrm{E}[X_n])$$

- The covariance of variables X_i and X_j is defined as

$$\mathrm{Cov}[X_i, X_j] := \mathrm{E}[(X_i - \mu_{X_i})(X_j - \mu_{X_j})]$$

  - $\mathrm{Cov}[X_i, X_i] = \mathrm{Var}[X_i]$
  - X_i, X_j independent $\Rightarrow \mathrm{Cov}[X_i, X_j] = 0$
  - $\mathrm{Cov}[X_i, X_j] > 0$ roughly means that X_i and X_j increase and decrease together.
  - $\mathrm{Cov}[X_i, X_j] < 0$ roughly means that when X_i increases X_j decreases (and vice versa).

Covariance matrix

For a random vector $\boldsymbol{X} = (X_1, \ldots, X_n)$ we define its $n \times n$ *covariance matrix* as follows:

$$\Sigma_{\boldsymbol{X}} = \begin{bmatrix} \mathrm{Var}[X_1] & \mathrm{Cov}[X_1, X_2] & \cdots & \mathrm{Cov}[X_1, X_n] \\ \mathrm{Cov}[X_2, X_1] & \mathrm{Var}[X_2] & \cdots & \mathrm{Cov}[X_2, X_n] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[X_n, X_1] & \mathrm{Cov}[X_n, X_2] & \cdots & \mathrm{Var}[X_n] \end{bmatrix}$$

- The diagonal elements are the variances of each random variable $\mathrm{Cov}[X_i, X_i] = \mathrm{Var}[X_i]$.
- $\Sigma_{\boldsymbol{X}}$ is symmetric, because $\mathrm{Cov}[X_i, X_j] = \mathrm{Cov}[X_j, X_i]$.
- $\Sigma_{\boldsymbol{X}}$ is positive semi-definite.
- What does it mean if $\Sigma_{\boldsymbol{X}}$ is diagonal?

K-means Clustering

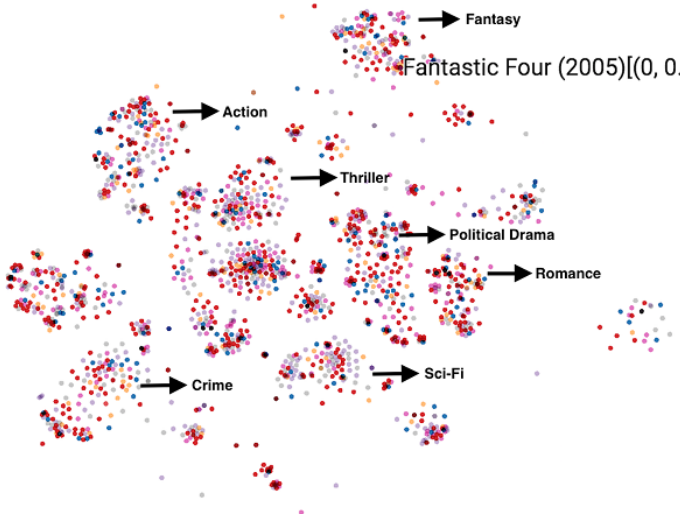
The clustering problem

- Also known as *vector quantization*, depending on the application.
- Consider N data points in a D-dimensional space, i.e. each data point is a D-dimensional vector $\boldsymbol{x}_n$, $n = 1, \ldots, N$.
- Our goal is to partition the data set into K clusters.
- In other words, find K representative vectors (centroids) $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K$, one for each cluster, that best fit the data according to some distance metric.

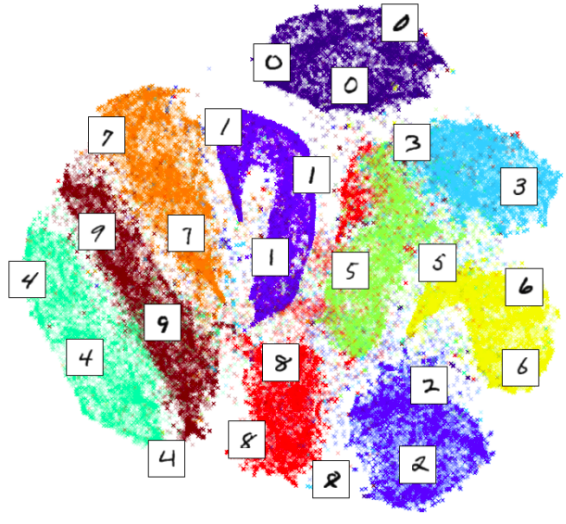
(Some) applications of clustering

Compression

Semantic segmentation



Topic modeling



Pattern recognition

K-means

One of the many vector quantization algorithms.

- Arguably the most famous and the simplest
- The distance metric is the *squared* Euclidean distance
- **Not** the Euclidean distance, which results in another algorithm (K-medoids)

Objective

Minimize the following cost function

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{k,n} \|\boldsymbol{x}_n - \boldsymbol{u}_k\|_2^2.$$

- Data points: $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^D$
- Centroids: $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K \in \mathbb{R}^D$
- Assignments: $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N \in \mathbb{R}^K$ (with $z_{k,n} := (\boldsymbol{z}_n)_k$)

K-means constraints

Hard assignment constraints

Each point $\boldsymbol{x}_n$ is assigned to exactly one cluster:

- $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N \in \{0, 1\}^K$
- $\sum_{k=1}^{K} z_{k,n} = 1, \ \forall n \in \{1, \ldots, N\}$

In practice

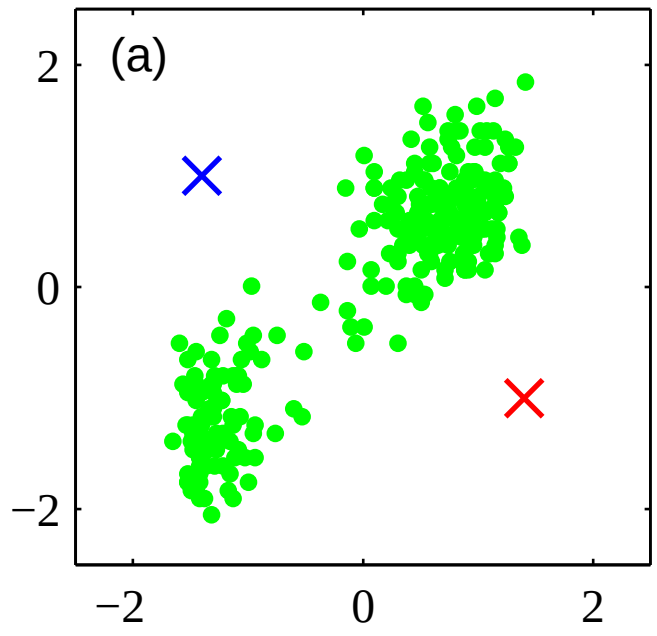
- K-means builds a dictionary that maps code words to points and vice versa.
- The assignment matrix $\mathbf{Z}$ can be just implemented as a list of indices.

Challenges

- The objective function J is **non-convex**
- Finding the global optimum is NP-Hard
  - Only possible by brute-forcing all assignments
  - Exception: 1D data (dynamic programming solution)
- In practice: local minima are good enough.

K-means algorithm

1. Initialize centroids $\boldsymbol{u}_1^{(0)}, \ldots, \boldsymbol{u}_K^{(0)}$ and $t \leftarrow 1$.

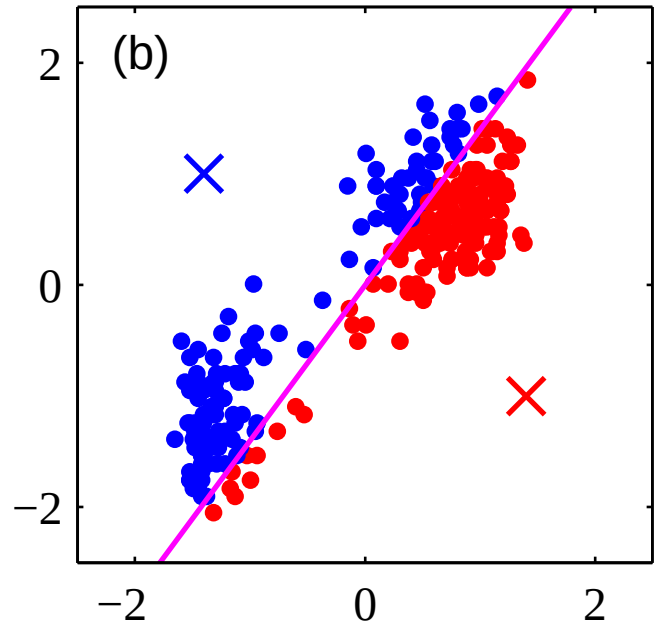


K-means algorithm

2. **Cluster assignment.**

$$k^*(\boldsymbol{x}_n) = \underset{k \in \{1,\ldots,K\}}{\arg\min} \; \|\boldsymbol{x}_n - \boldsymbol{u}_k^{(t-1)}\|_2^2, \;\; \forall n \in \{1,\ldots,N\}$$

$$z_{j,n}^{(t)} = \begin{cases} 1 & \text{, if } j = k^*(\boldsymbol{x}_n) \\ 0 & \text{, otherwise} \end{cases}, \;\; \forall n \in \{1,\ldots,N\}$$

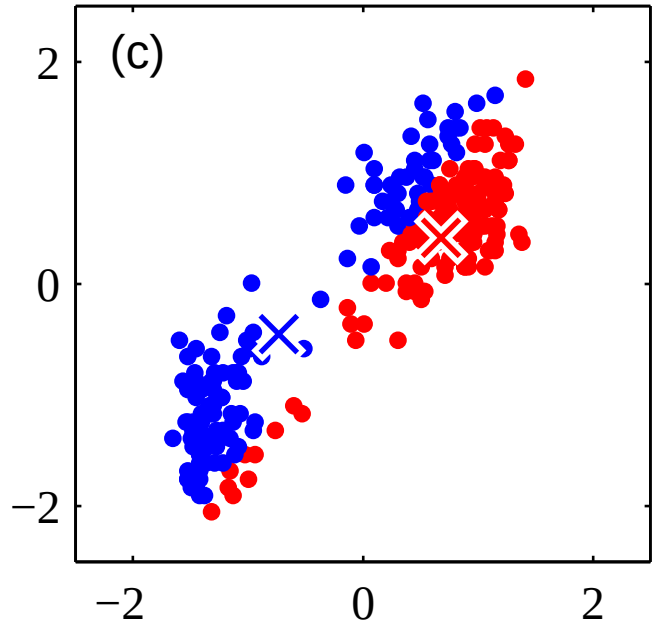


K-means algorithm

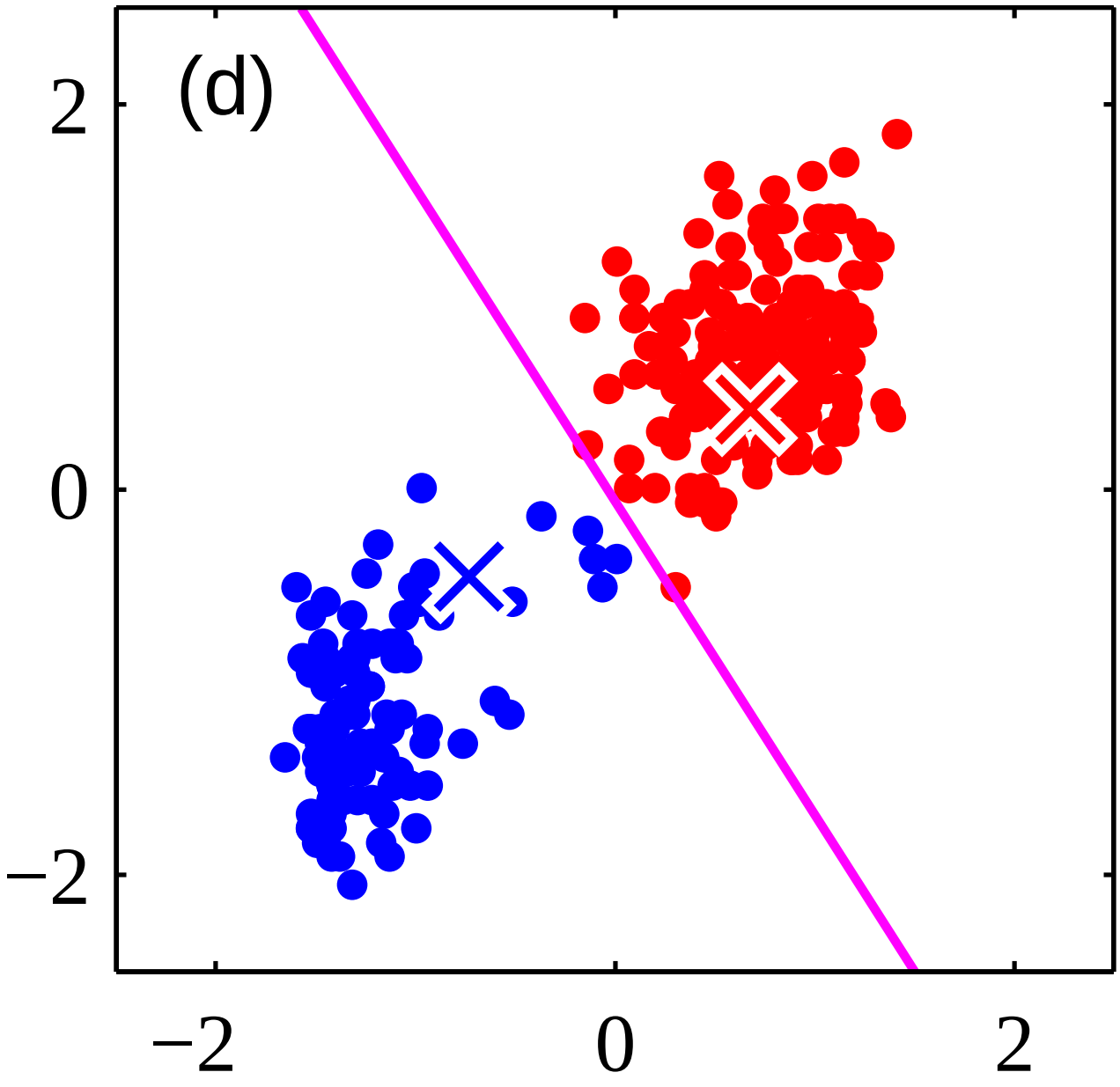
3. **Centroid update.**

$$\boldsymbol{u}_k^{(t)} = \frac{\sum_{n=1}^{N} z_{k,n}^{(t)} \boldsymbol{x}_n}{\sum_{n=1}^{N} z_{k,n}^{(t)}}, \ \forall k \in \{1, \ldots, K\}$$

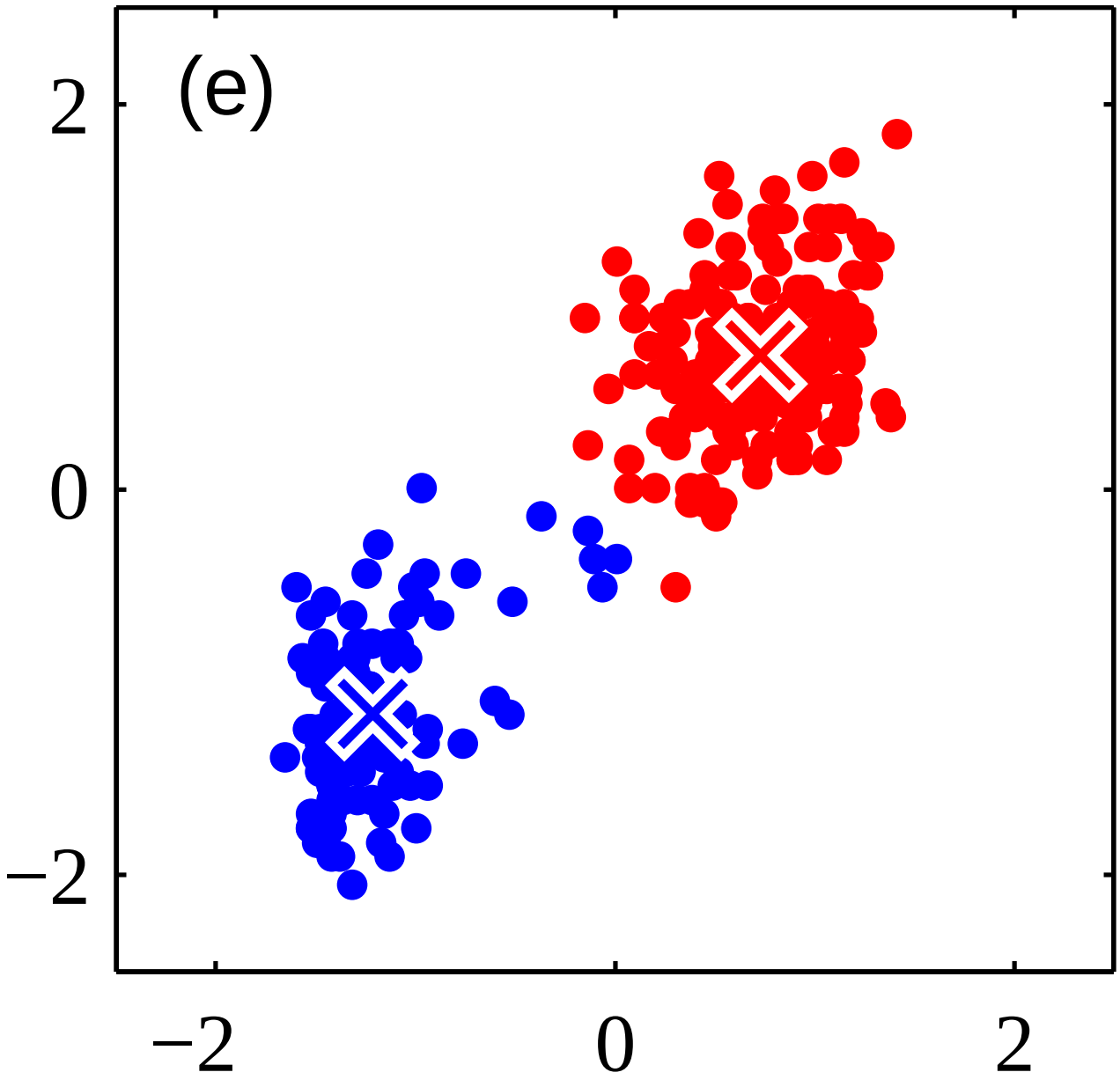
4. If termination condition ($\boldsymbol{u}_k^{(t)} = \boldsymbol{u}_k^{(t-1)}, \ \forall k$) is not met, $t \leftarrow t + 1$ and go to step 2.



K-Means: Second E-Step



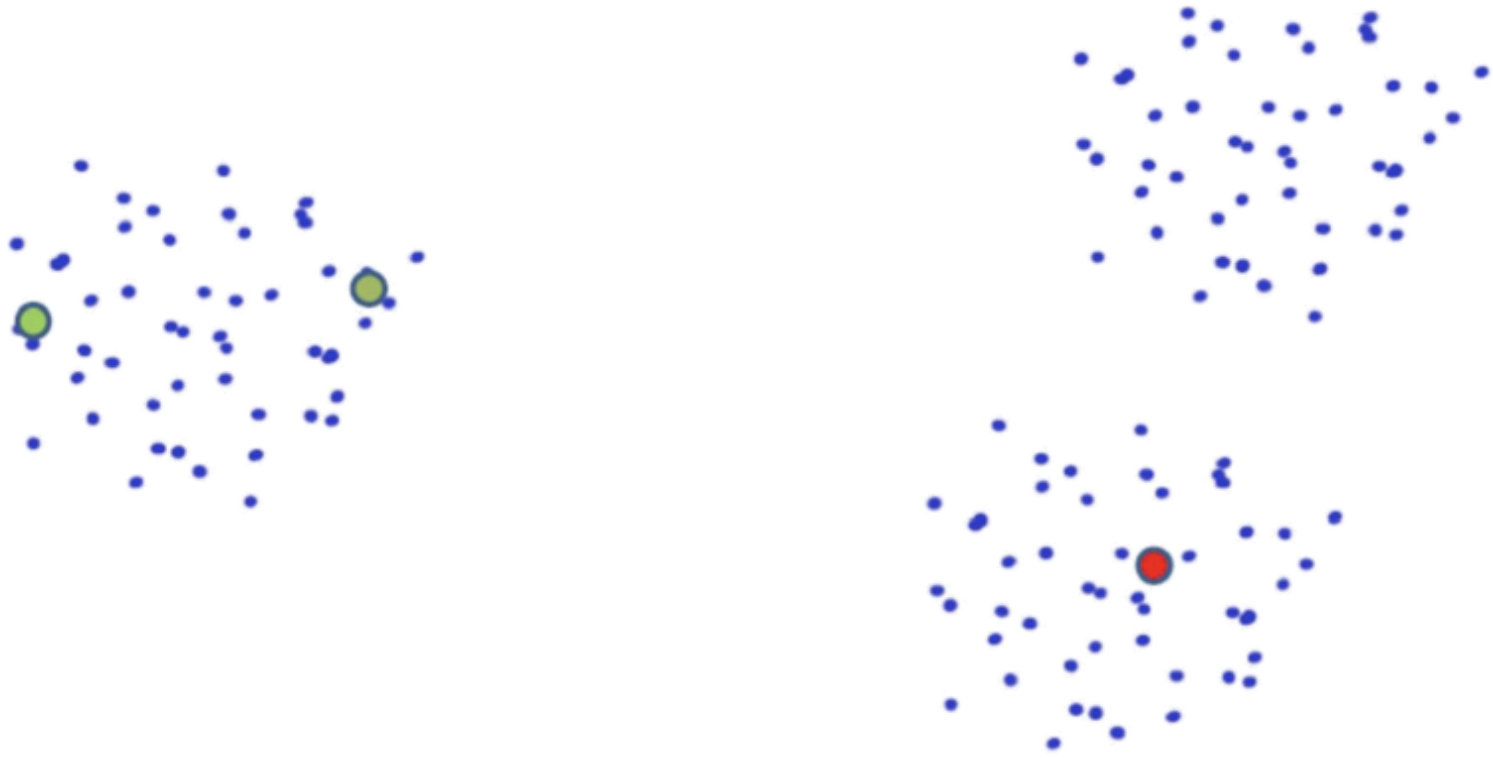
K-Means: Second M-Step



Practical considerations

- Convergence to local minimum **guaranteed**
- Quadratic convergence rate
  - Equivalent to Newton's method
  - In principle, J can also be optimized via (stochastic) gradient descent
- Computational cost: $\mathcal{O}(nkd)$ per iteration
- Issues: convergence to poor minima (less likely with good initialization), empty clusters. Some implementations take the best result out of multiple runs.

Bad initialization



Bad initialization

K-Means convergence proof

Strategy

Prove that steps **2** (E step) and **3** (M step) always result in a decrease (or no change) of the objective function J.

- E step: cluster centroids are fixed, assignments change
- M step: cluster centroids change, assignments are fixed
- What is the minimizer (optimal strategy) of each step?

K-Means convergence proof

E step

Objective function J minimized by definition, since we assign each point to the nearest centroid.

$$k^*(\boldsymbol{x}_n) = \underset{k \in \{1,\ldots,K\}}{\arg\min} \; \|\boldsymbol{x}_n - \boldsymbol{u}_k^{(t-1)}\|_2^2, \; \forall n \in \{1, \ldots, N\}$$

$$z_{j,n}^{(t)} = \begin{cases} 1 & \text{, if } j = k^*(\boldsymbol{x}_n) \\ 0 & \text{, otherwise} \end{cases}, \; \forall n \in \{1, \ldots, N\}$$

K-Means convergence proof

M step

We exploit the property of the mean being the minimizer of the sum of squared distances to all points

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x_n} = \min_{\boldsymbol{\mu}}\sum_{n=1}^{N}\|\mathbf{x_n} - \boldsymbol{\mu}\|^2$$

or for a weighted mean:

$$\boldsymbol{\mu} = \frac{\sum_{n=1}^{N} w_i\mathbf{x_n}}{\sum_{n=1}^{N} w_i} = \min_{\boldsymbol{\mu}}\sum_{n=1}^{N} w_i\|\mathbf{x_n} - \boldsymbol{\mu}\|^2$$

Strategy: derive gradient w.r.t. cluster centroids, set it to zero, and recover closed-form solution. Then show that this is a minimizer by pointing out that the function is convex (given $\mathbf{Z}$ fixed).

K-Means as a matrix factorization problem

K-Means solves the matrix factorization problem:

$$\min \|\mathbf{X} - \mathbf{UZ}\|_F^2$$

where $\mathbf{Z}$ is an indicator matrix.

Proof strategy

Show that the formulation above is equivalent to the original objective function J

$$\|\mathbf{X} - \mathbf{UZ}\|_F^2 = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{k,n} \|\boldsymbol{x}_n - \boldsymbol{u}_k\|_2^2 = J$$

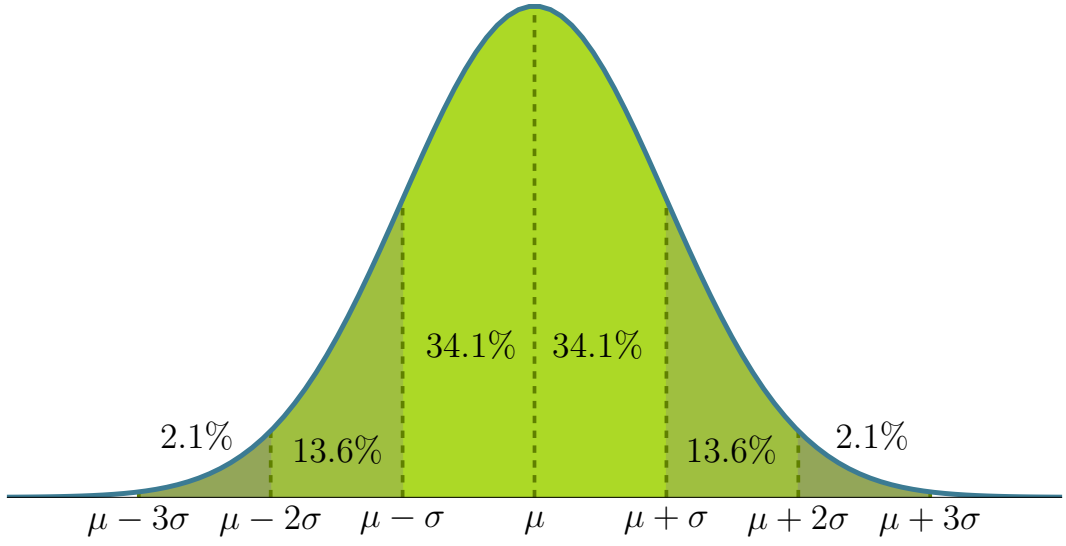
Gaussian mixture models

Gaussian distribution (1-D)

- Random variable X with $\mathcal{X} = \mathbb{R}$
- Probability density function

$$p(x) := \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- $\mathrm{E}[X] = \mu, \ \mathrm{Var}[X] = \sigma^2$

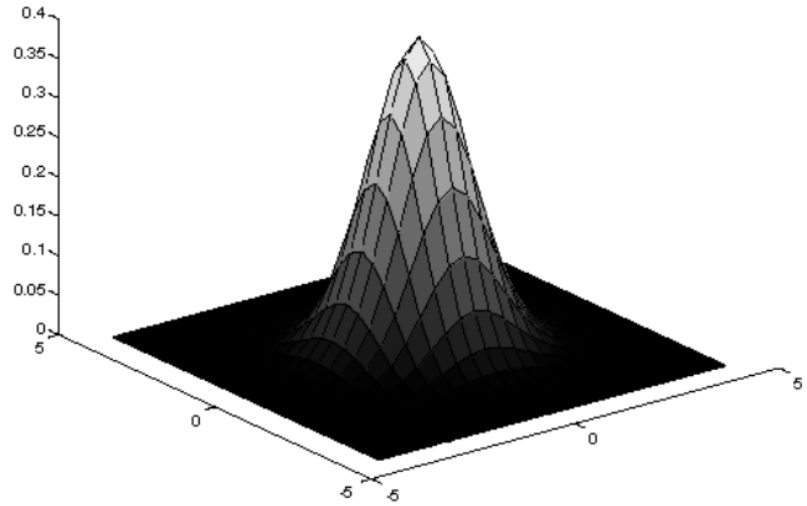


Gaussian Distribution (d-D)

- Random vector $\boldsymbol{X} = (X_1, \ldots, X_d)$ with $\mathcal{X} = \mathbb{R}^d$
- Probability density function

$$p(\boldsymbol{x}) := \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right)$$

- $\mathrm{E}[\boldsymbol{X}] = \boldsymbol{\mu}$
- Σ is the covariance matrix of $\boldsymbol{X}$ and $|\Sigma|$ is its determinant.



Data vs. Distribution

- Data: input
- Distribution: model assumption

- ML methods usually make some general assumption about the distribution (e.g. a parametric family) then try to obtain ("infer") the specifics from the data available.

- Example:
  - **Modeling step:** Assume a Gaussian distribution as model (parameterized by mean and variance)
  - **Inference Step:** Estimate parameters mean & variance from data.

Maximum Likelihood

- Data set: $D = \{\mathbf{x}_n\}, \quad n = 1, \ldots, N$
- Likelihood function: $p(D \mid \boldsymbol{\mu}, \Sigma) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}, \Sigma)$
- Set the parameter by maximizing log likelihood

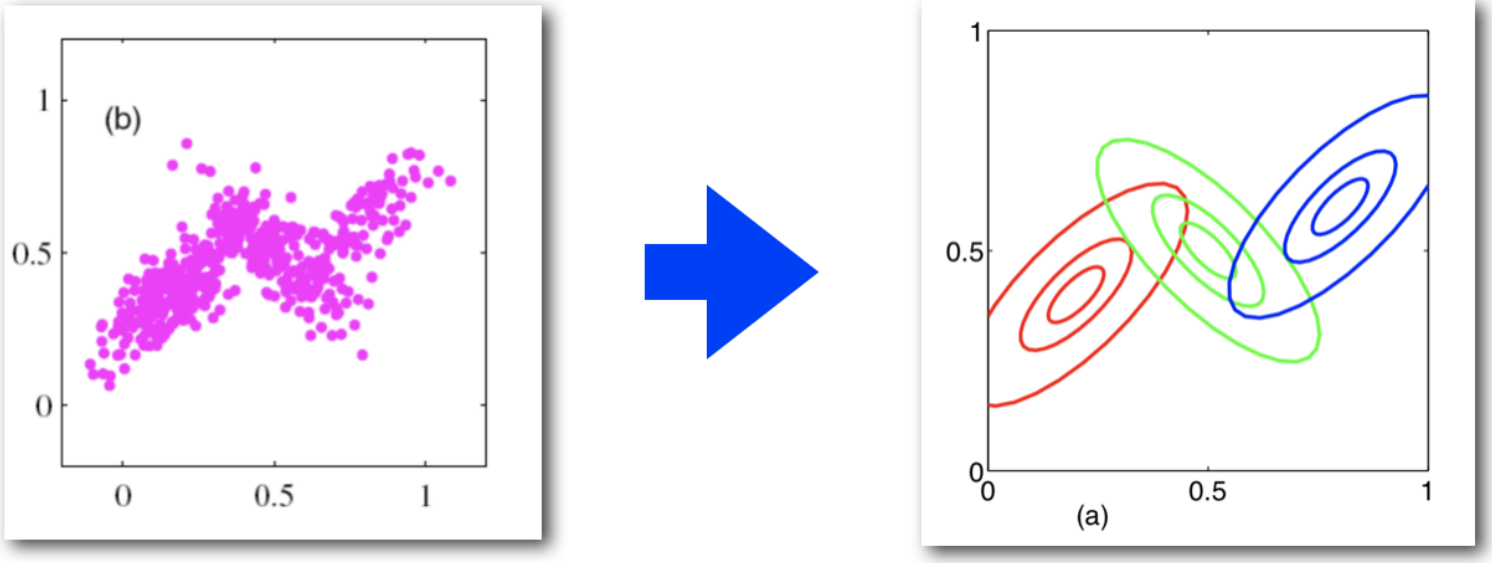
$$\log p(D \mid \boldsymbol{\mu}, \Sigma) = -\frac{N}{2} \log |\Sigma| - \frac{Nd}{2} \log(2\pi) - \frac{1}{2} \sum_{n=1}^{N} (\mathbf{x}_n - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}_n - \boldsymbol{\mu})$$

- Take derivatives and obtain sample mean and sample variance

$$\boldsymbol{\mu}_{ML} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n \qquad \Sigma_{ML} = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}_n - \boldsymbol{\mu}_{ML})(\mathbf{x}_n - \boldsymbol{\mu}_{ML})^T$$

What if...

Our data looks like:



Gaussian Mixture Models

Assume data is generated from a weighted mixture of K Gaussian distributions:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Normalization and positivity require: $\pi_k \geq 0, \ \sum_{k=1}^{K} \pi_k = 1$

Generation Process

- Sample k with probability π_k.
- Sample x with probability $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

Mixing Coefficients

The mixing coefficients (π_k) can be interpret as prior prob:

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

GMM - Parameters

K mixture components with parameters (for $k = 1, \ldots, K$):

- $\boldsymbol{\mu}_k$: mean of the k-th component (similar to centroid $\boldsymbol{u}_k$ in K-means)
- $\boldsymbol{\Sigma}_k$: covariance of the k-th component
- π_k: mixture weight of the k-th component

MLE ?

GMM - Objective

The likelihood of all the data is:

$$p(\boldsymbol{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)$$

Maximize the log-likelihood of the Gaussian mixture model:

$$L(\boldsymbol{X}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) := \ln p(\boldsymbol{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Log of a sum !
It is really hard to optimize with respect to $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$... We need to find an other method to compute them!

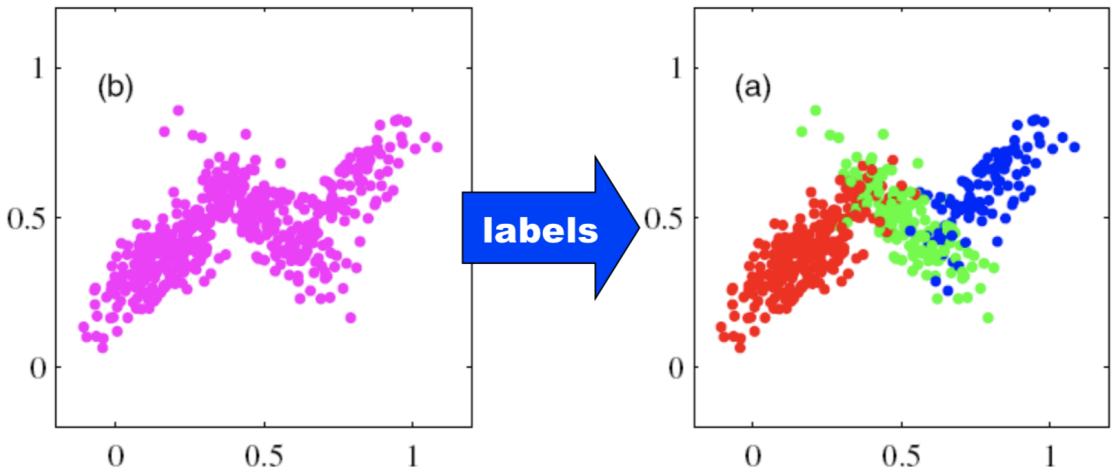
GMM - Latent Variables

- Let's introduce new variables z_k, called **latent variables**, that tell us which point comes from which gaussian.

- For each data point **x**:

$$z_k = \begin{cases} 1 & \text{if } \mathbf{x} \text{ comes from k-th Gaussian component} \\ 0 & \text{ow} \end{cases}$$

- Note: $\sum_{k=1}^{K} z_k = 1$ and $p(z_k = 1) = \pi_k$.

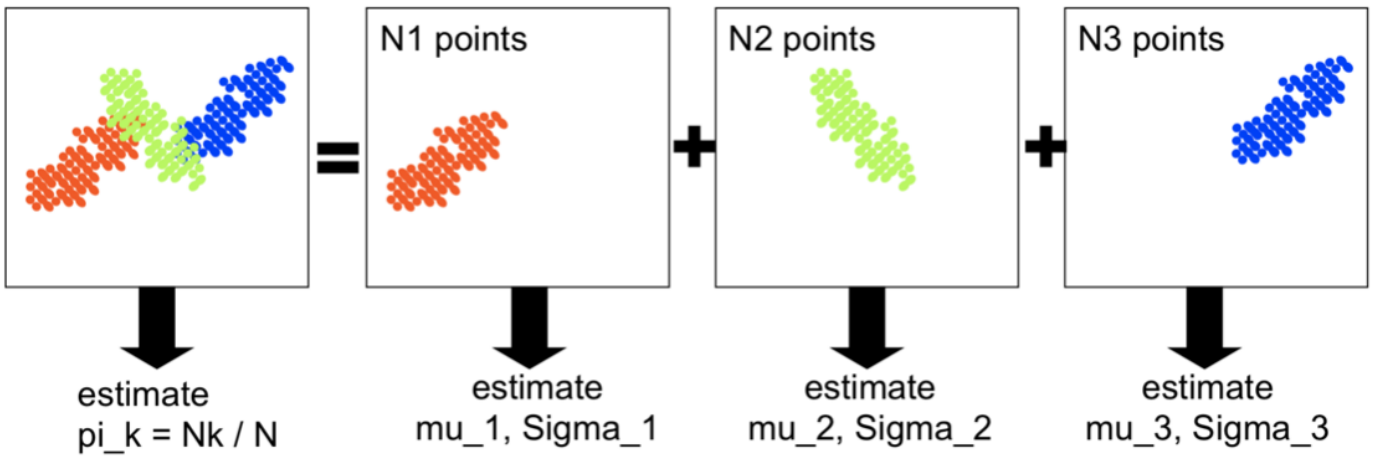


GMM - Latent Variables

- For each data point we define $\mathbf{z} = (z_1, z_2, \ldots, z_K)$, where $z_i = 0$ for all $i \neq k$, and $z_k = 1$.
- Then the conditional distribution of $\boldsymbol{x}$ given a $\mathbf{z}$ is a Gaussian:

$$p(\boldsymbol{x}|\mathbf{z}) = p(\boldsymbol{x}|z_k = 1) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Given $\mathbf{z}$ for each datapoint, the parameter inference is easy!



Complete Log-likeligood

- Remember the log-likelihood...
  $log\ p(\mathbf{x} \mid \pi_k, \boldsymbol{\mu}_k, \Sigma_k) = \log\left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \Sigma_k)\right)$
- The complete log-likelihood for each $\mathbf{x}$:

$$p(\mathbf{x}, \mathbf{z} \mid \pi_k, \boldsymbol{\mu}, \Sigma) = \prod_{k=1}^{K} \pi_k^{z_k} \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \Sigma_k)^{z_k}$$

- The complete log-likelihood for the dataset X then is:

$$\log\ p(\mathbf{X}, \mathbf{Z} \mid \pi_k, \boldsymbol{\mu}, \Sigma) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \Big(\log \pi_k + \log \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \Sigma_k) \Big)$$

The EM Algorithm - Key Idea

- The **EM algorithm** proposes instead to look at the expected complete log-likelihood:

$$E_Z\big[\log p(\mathbf{X}, \mathbf{Z} \mid \pi_k, \boldsymbol{\mu}, \Sigma)\big] = \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma_{nk}\Big(\log \pi_k + \log \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k)\Big)$$

- γ_{nk} is the posterior probability of the latent variables.

$$\gamma_{nk} = E(z_{nk}) = p(z_{nk} = 1) = p(z_k = 1 \mid \mathbf{x}_n)$$

- **Remember:** the expectation of a binary variable is the probability that is equal to 1.

The EM Algorithm - Lower Bound

- Let's find a lower-bound of the log-likelihood:

$$\log p(\mathbf{X};\theta) = \sum_{i=1}^{N} \log \left[\sum_{k=1}^{K} \pi_k p_{\theta_k}(\mathbf{x}_i) \right] = \sum_{i=1}^{N} \log \left[\sum_{k=1}^{K} q_{nk} \frac{\pi_k \; p_{\theta_k}(\mathbf{x}_i)}{q_{nk}} \right]$$
$$\geq \sum_{i=1}^{N} \sum_{k=1}^{K} q_{nk} \left[\log p_{\theta_k}(\mathbf{x}_i) + \log \pi_k - \log q_{nk} \right] = \mathcal{L}(\mathbf{X};\theta)$$

where q_{nk} is any distribution s.t. $\sum_{k=1}^{K} q_{nk} = 1$.

The EM Algorithm - E-Step

- Given that we can't directly maximize the log-likelihood we maximize its lower bound.
- In the lecture you derived the optimal q that makes this lower bound tighter:

$$q^*_{nk} = \frac{\pi_k \; p_{\theta_k}(\mathbf{x}_n)}{\sum_{l=1}^{K} \pi_l \; p_{\theta_l}(\mathbf{x}_n)} = p(z_k = 1 \mid \mathbf{x}_n) = \gamma_{nk}$$

- The optimal q–distribution equals the posterior probability of the latent variables.

The EM Algorithm - M-Step

- Now we maximize the lower bound w.r.t. the parameters $(\pi_k, \boldsymbol{\mu}, \Sigma)$.
- Let's have a closer look at the lower bound with optimal q:

$$\begin{aligned}\mathcal{L}(\mathbf{X}; \theta) &= \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{nk} \left[\log p_{\theta_k}(\mathbf{x}_i) + \log \pi_k - \log \gamma_{nk}\right] \\ &= E_Z\left[\log p(\mathbf{X}, \mathbf{Z} \mid \theta)\right] - \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{nk} \log \gamma_{nk}\end{aligned}$$

- Hence optimizing $\mathcal{L}(\mathbf{X}; \theta)$ w.r.t. θ is equal to maximize the **expected complete data log-likelihood**.

EM Algorithm: M-Step

3. Find the parameters $(\pi_k, \boldsymbol{\mu}, \Sigma)$ that maximize the expected log likelihood.

Blackboard

The EM Algorithm - Overview

The EM algorithm proposes to:

1. Compute the posterior probability of the latent variables.
2. Compute the expected value of the complete log likelihood.
3. Find the parameters $(\pi_k, \boldsymbol{\mu}, \Sigma)$ that maximize the expected complete log likelihood.
4. Iterate.

The EM algorithm - Overview

1. Initialize $\pi_k^{(0)}$, $\boldsymbol{\mu}_k^{(0)}$, $\boldsymbol{\Sigma}_k^{(0)}$ for $k = 1, \ldots, K$ and $t \leftarrow 1$.
2. **E-step.** Evaluate responsibilities using current parameters:

$$\gamma_{nk} := \frac{\pi_k^{(t-1)} \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k^{(t-1)}, \boldsymbol{\Sigma}_k^{(t-1)})}{\sum_{j=1}^{K} \pi_j^{(t-1)} \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_j^{(t-1)}, \boldsymbol{\Sigma}_j^{(t-1)})}$$

3. **M-step.** Update parameters using new responsibilities:

$$\begin{aligned}
\boldsymbol{\mu}_k^{(t)} &:= \frac{\sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x}_n}{\sum_{n=1}^{N} q_{kn}} \\
\boldsymbol{\Sigma}_k^{(t)} &:= \frac{1}{\sum_{n=1}^{N} \gamma_{nk}} \sum_{n=1}^{N} \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(t)})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(t)})^T \\
\pi_k^{(t)} &:= \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk}
\end{aligned}$$

4. If termination condition is not met, $t := t + 1$ and go to step 2.

Gaussian Mixture Models

Side remark.
Log likelihood for GMM is:

$$L(X, \mu, \Sigma) = \sum_{n=1}^{N} \log\left(\pi_{z_i} \mathcal{N}(\boldsymbol{x}_n, \mu_{z_i}, \Sigma_{z_i})\right)$$

$$L(X, \mu) \sim -\sum_{n=1}^{N} (\boldsymbol{x}_n - \mu_{z_i})^T \Sigma_{z_i}^{-1} (\boldsymbol{x}_n - \mu_{z_i})$$

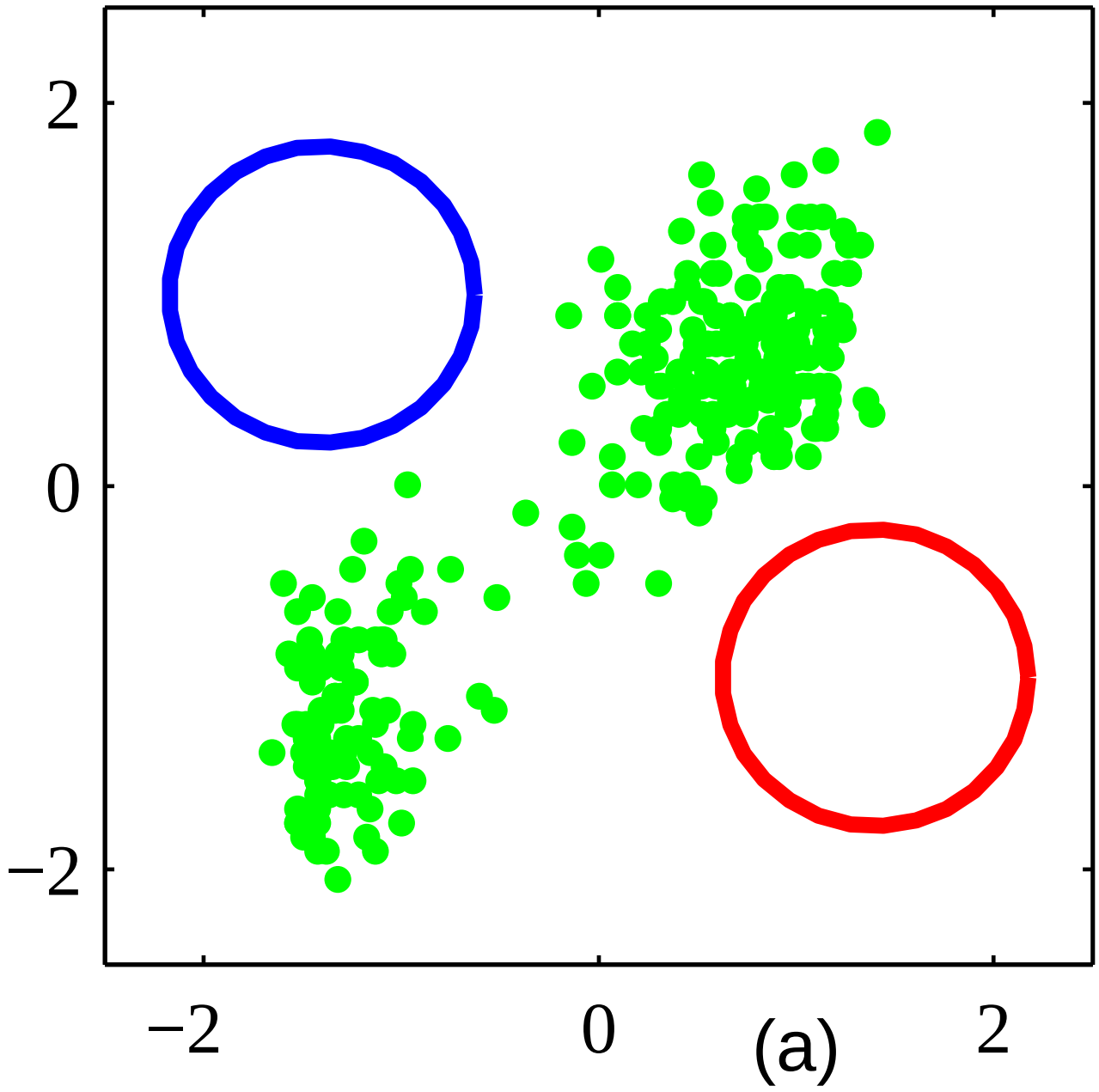
Which looks almost the same as K-means functional with introduced z:

$$L_{\text{K-means}}(X, \mu) = \sum_{n=1}^{N} \min_{j \in 1,\ldots,k} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||_2^2 = \sum_{n=1}^{N} ||\boldsymbol{x}_n - \mu_{z_i}||_2^2$$

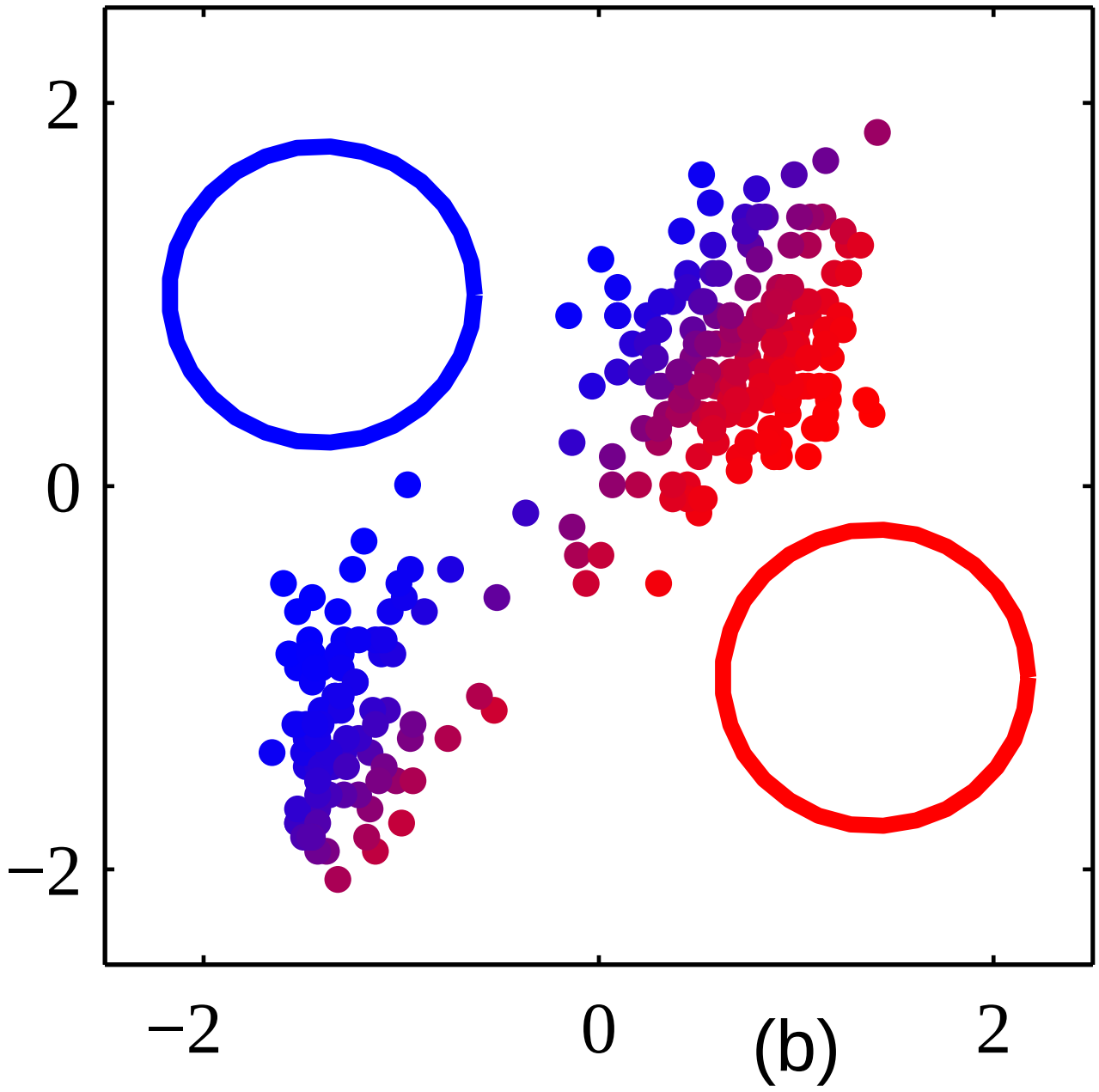
K-means vs. mixture models

- K-means
  - Hard cluster assignments
  - Spherical clusters with uniform prior
  - Fast runtime (can be used to initialize a mixture model)
- Gaussian mixture models
  - Soft cluster assignments $\leftrightarrow$ probabilities of assigments
  - Each cluster has its own covariance ($\boldsymbol{\Sigma_k}$) and "weight" (π_k)
  - Slower runtime

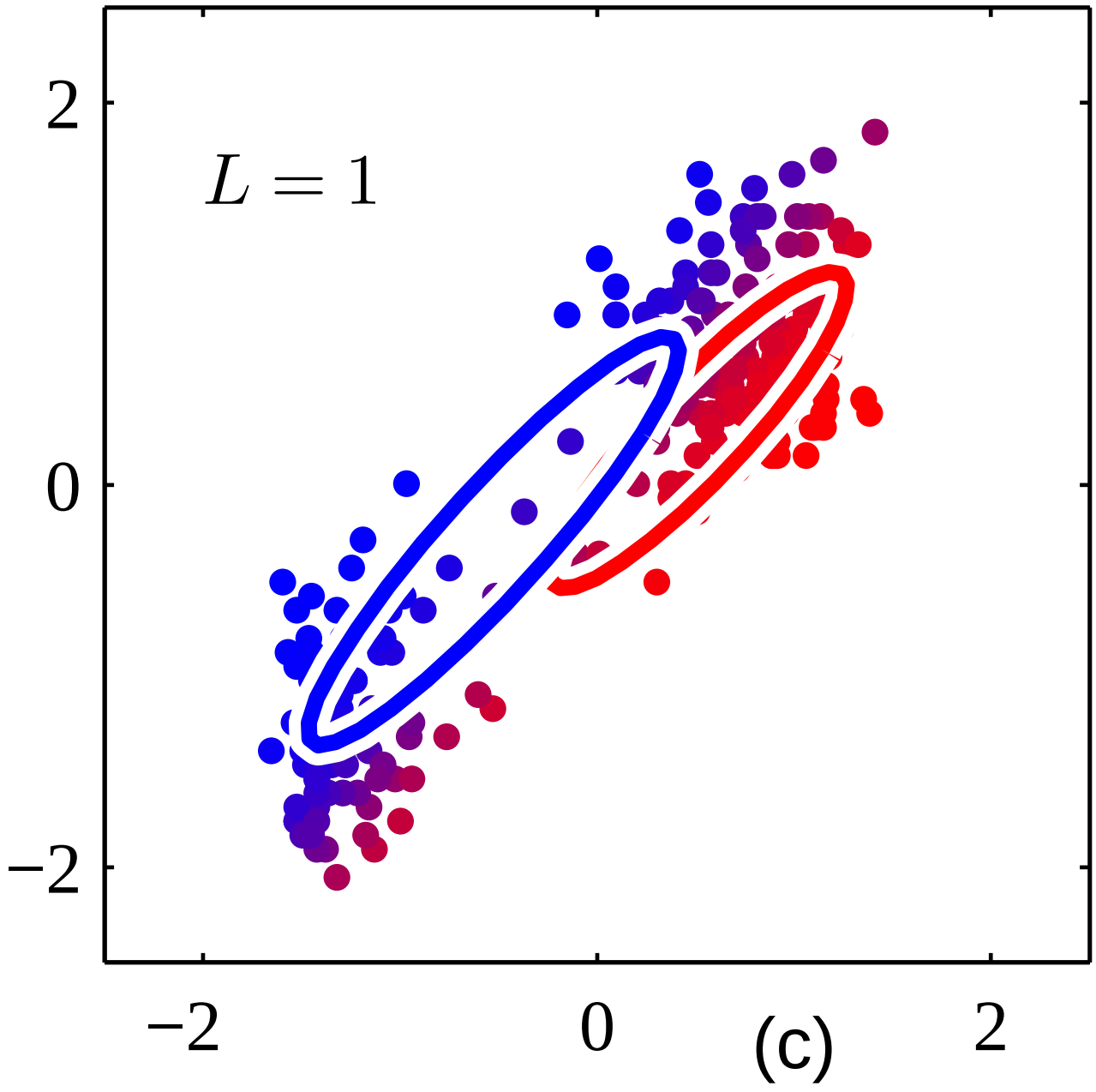
GMM: Initial configuration



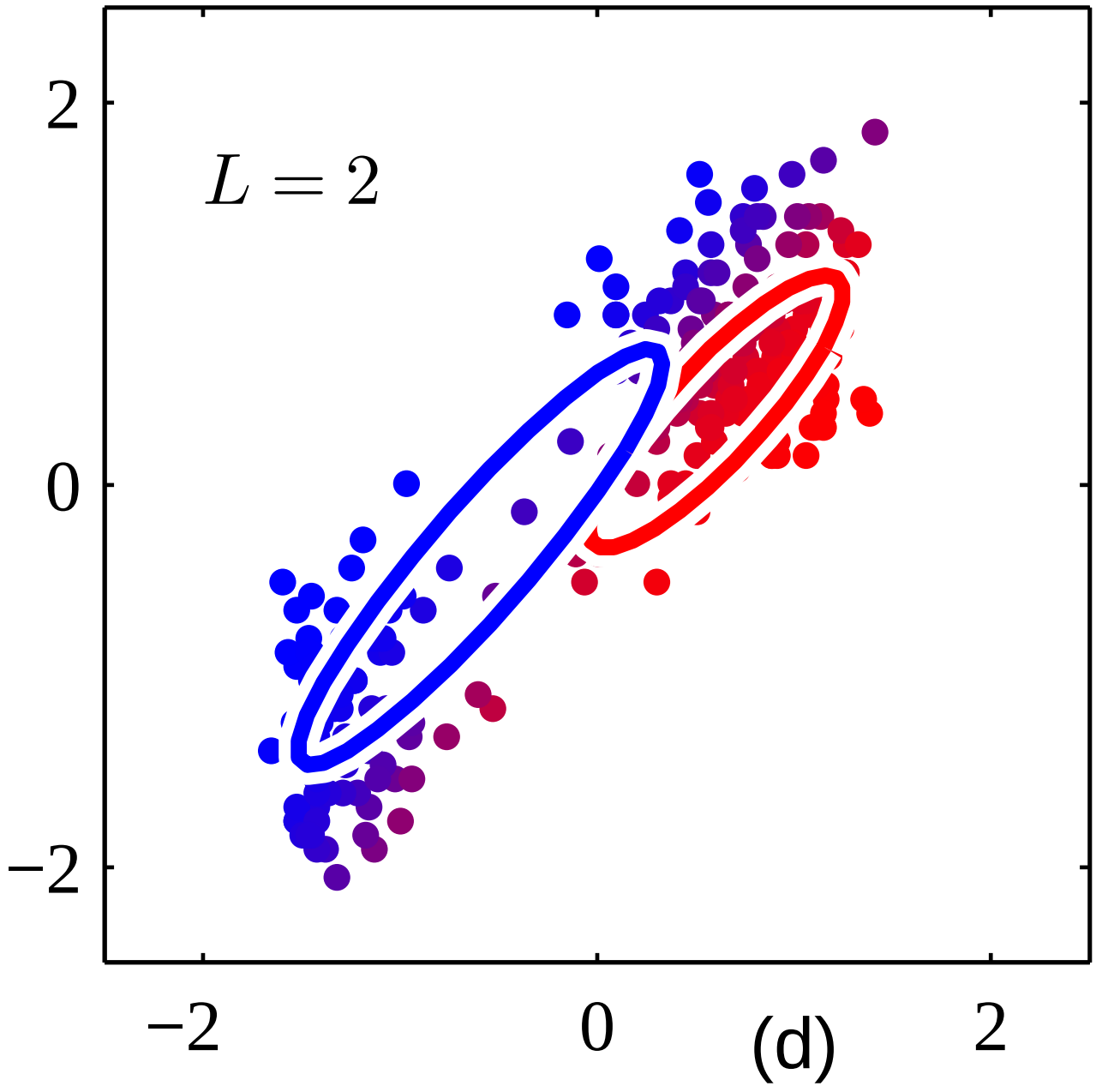
GMM: First E-Step



GMM: First M-Step



GMM: Two EM cycles



GMM: Five EM cycles