Gaussian Mixture Models

Probabilistic Decision Making — Lecture 8

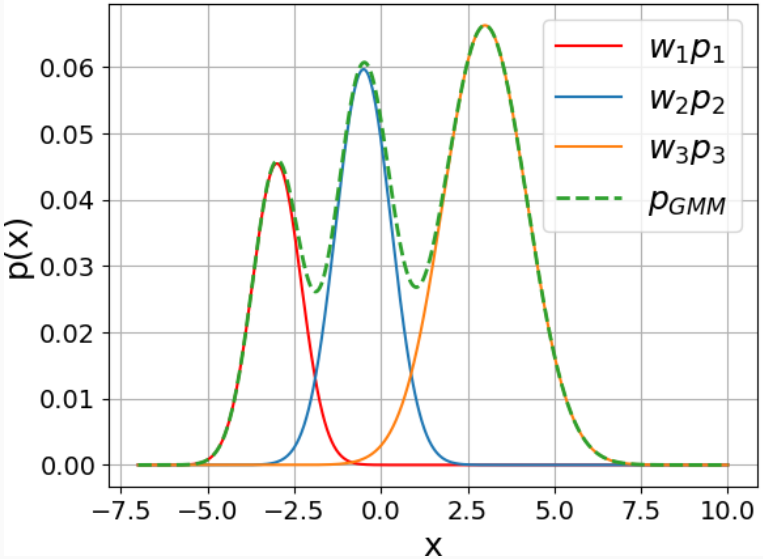
19th November 2025

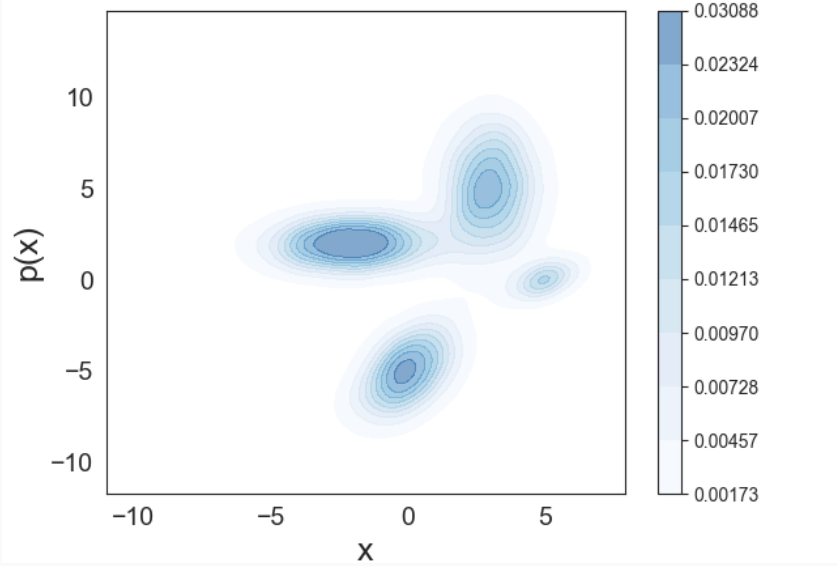
Robert Peharz
Institute of Machine Learning and Neural Computation
Graz University of Technology

The **Gaussian mixture model (GMM)** parametrized by $\boldsymbol{\theta} = \{w_k, \boldsymbol{\mu}_k, \Sigma_k\}_{k=1}^{K}$ is given as

$$p_{\text{GMM}}(\boldsymbol{x} \mid \boldsymbol{\theta}) = \sum_{k=1}^{K} w_k \, p(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \Sigma_k)$$

where $p(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \Sigma_k)$ is a Gaussian with parameters $\boldsymbol{\mu}_k, \Sigma_k$.





GMMs are Universal Approximators

Theorem

GMMs are universal approximators of densities, that is, any density can be approximated arbitrarily well by a GMM.
This is true even when:

- Gaussians are restricted to diagonal covariances,
- the diagonal is constant,
- and the mixture weights w_k are constant

However, we don't know how large K needs to be and how to set the parameters.

Note the parallel to multi layer perceptrons (MLPs), which are universal approximators for continuous functions.

A simple way to "learn" a GMM is to place an **isotropic** (constant diagonal) Gaussian component on each data point.

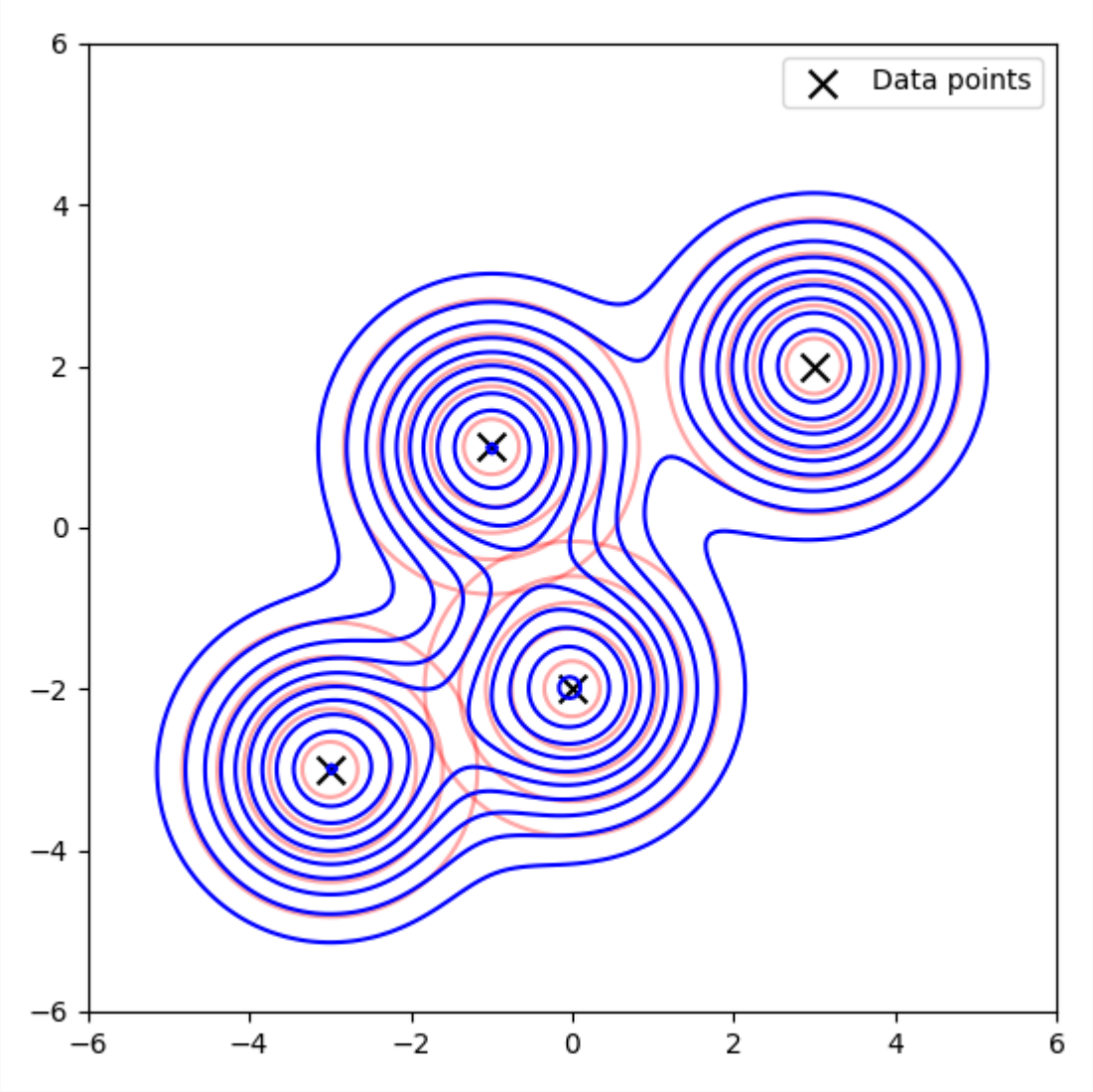
- let $\mathcal{D} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\}$ be a dataset
- the GMM defined as

$$p_{\text{KDE}}(\boldsymbol{x}) = \sum_{i=1}^{N} \frac{1}{N} p_{\text{gauss}}(\boldsymbol{x} \mid \boldsymbol{x}^{(i)}, \sigma I)$$

  is called **kernel density estimator (KDE)** for data $\mathcal{D}$
- here, σ is some **bandwidth parameter**

Kernel Density Estimator

Example



KDE is simple but a **consistent density estimator**. In particular, when

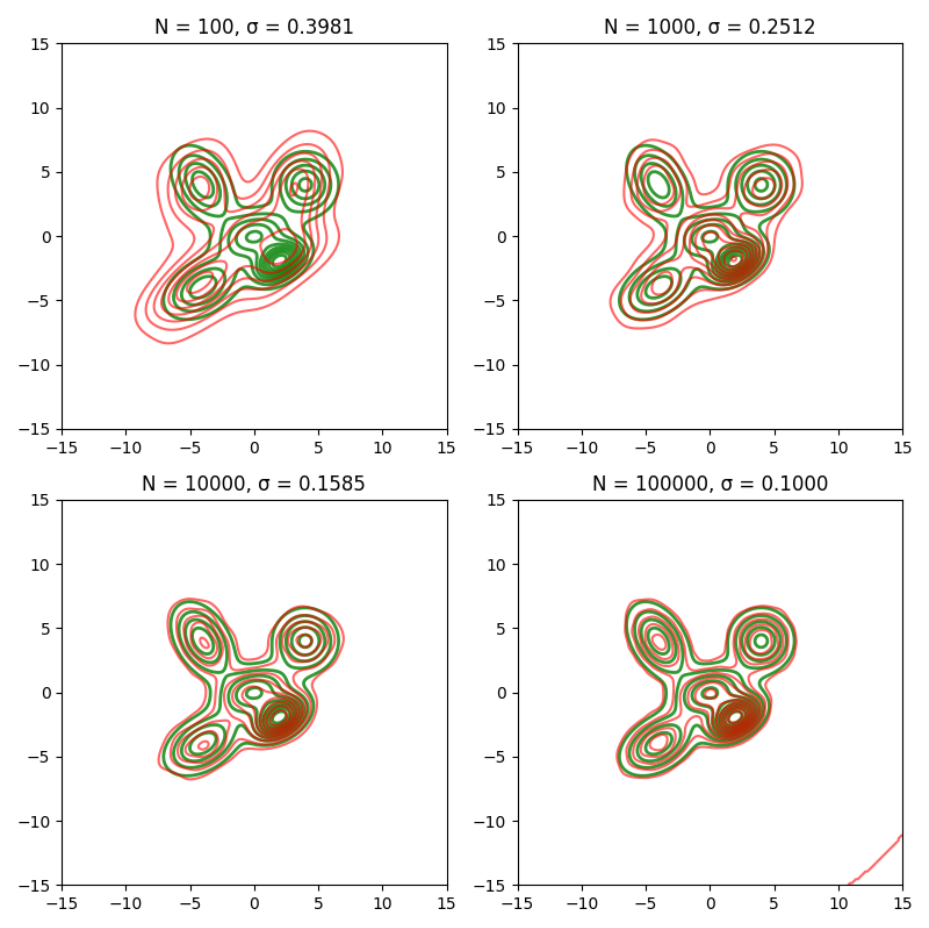
- number of data samples $N \to \infty$
- bandwidth $\sigma \to 0$
- but "σ goes to 0 slowly", i.e. $\sigma N \to \infty$,

then the KDE converges to p^*.

- this is also true for arbitrary bounded and symmetric density components, not only Gaussians
- theoretically optimal convergence rate for $\sigma = N^{-0.2}$

Consistency of KDE

Example



We use a ground truth p^* (here a 5 component GMM) from which samples can be drawn (green). In red we show the KDE fit for a varying number of samples.

GMMs as Latent Variable Models

GMMs as Latent Variable Models

Recall the GMM definition: $p(\boldsymbol{x}) = \sum_{k=1}^{K} w_k \, p(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \Sigma_k)$

- the weights $w_k \geq 0$, $\sum_k w_k = 1$ look like a **Categorical distribution** of some **latent** (unobserved) random variable Z
- $p(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \Sigma_k)$ might be interpreted as a conditional distribution $p(\boldsymbol{x} \mid z = k)$

This yields a joint via the product rule:

$$p(\boldsymbol{x}, z) = \overbrace{p(z)}^{w_z} \overbrace{p(\boldsymbol{x} \mid z)}^{p(\boldsymbol{x} \mid \boldsymbol{\mu}_z, \Sigma_z)}$$

Since Z is unobserved, we **marginalize**:

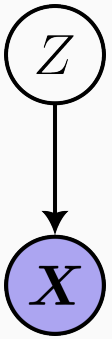
$$p(\boldsymbol{x}) = \sum_z p(\boldsymbol{x}, z) = \sum_z p(z) \, p(\boldsymbol{x} \mid z)$$

retrieving exactly the GMM definition above!

GMMs as Latent Variable Models cont'd

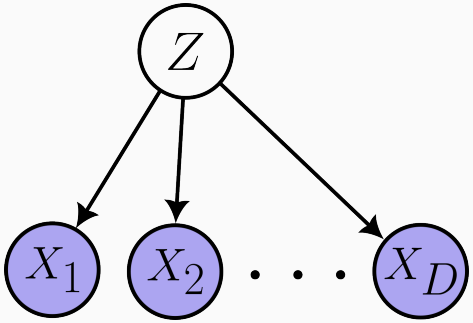
The GMM might be understood as a joint over $\boldsymbol{X}$ and a latent Z.

$$Z \sim \text{Categorical}(w_1, w_2, \ldots, w_K)$$
$$\boldsymbol{X} \sim \text{Gaussian}(\boldsymbol{\mu}_Z, \boldsymbol{\Sigma}_Z)$$



If the Gaussian components have diagonal covariances:

$$Z \sim \text{Categorical}(w_1, w_2, \ldots, w_K)$$
$$X_d \sim \text{Gaussian}(\mu_{Zd}, \sigma_{Zd}), \; d = 1 \ldots D$$



Sampling GMMs

How can we draw samples from a GMM?

Sampling GMMs

How can we draw samples from a GMM?

We combine two previous ideas:

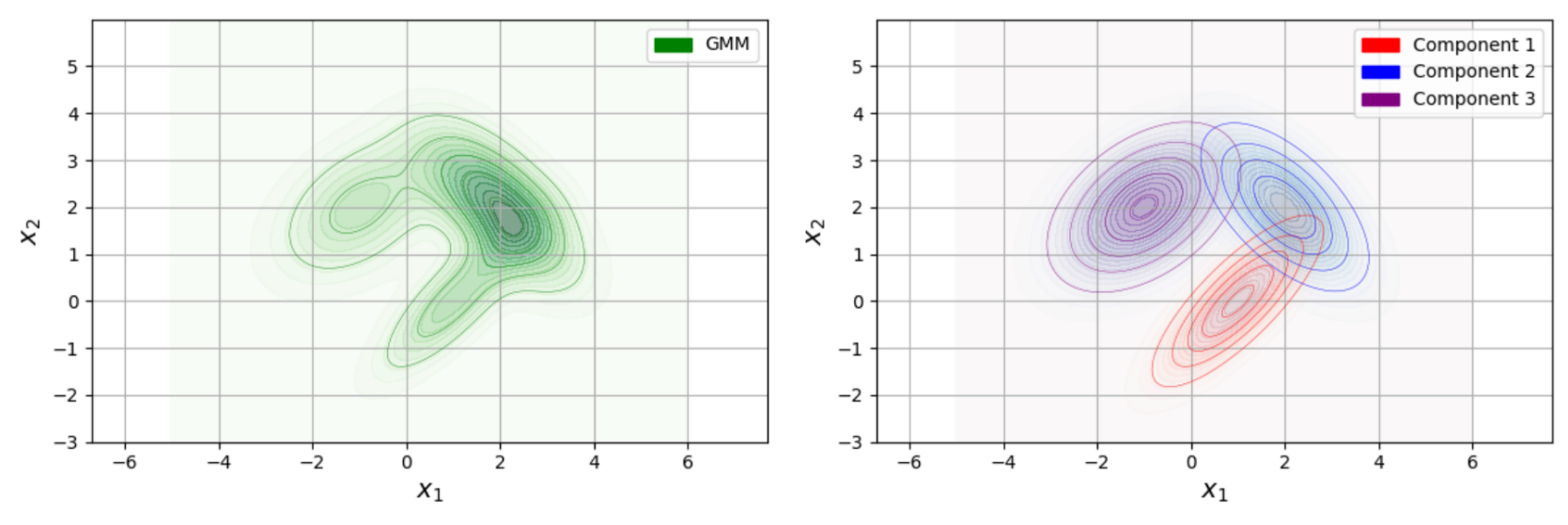
- **ancestral sampling** in BNs to draw a joint (Lecture 7)
  - sampling a Categorical Z
  - sampling $\boldsymbol{X}$ from the Gaussian selected by Z
- having a joint sample, a **marginal** sample $\boldsymbol{X}$ is obtained by discarding marginalized dimensions Z (Lecture 4)

Sampling in GMMs

Example

Sampling a GMM?

- $Z \sim \text{Categorical}(w_1, w_2, w_3)$
- $\boldsymbol{X} \sim \text{Gaussian}(\mu_z, \Sigma_z)$
- discard Z



$\boldsymbol{X}$ is an **exact sample** from p_{GMM}

Learning GMMs

Learning with Latent Variables?

A GMM can be seen as a joint over $X_1, X_2, \ldots, X_D$ and Z. However, since Z is **latent**, we never observe its value, i.e. our dataset is **incomplete** and actually looks like this:

i	x_1	x_2	$\ldots$	x_D	z
1	-0.34	0.04	$\ldots$	-0.53	?
2	1.33	-0.60	$\ldots$	0.94	?
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
N	1.64	-1.80	$\ldots$	1.99	?

Assuming Complete Data

Let us ignore the problem that Z is latent (missing) for a moment, and pretend we actually had observed it:

i	x_1	x_2	$\dots$	x_D	z
1	-0.34	0.04	$\dots$	-0.53	2
2	1.33	-0.60	$\dots$	0.94	1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
N	1.64	-1.80	$\dots$	1.99	2

How do we learn the GMM parameters θ in this case?

Log-Likelihood Decomposes

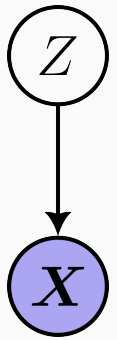
Recall from last lecture that in Bayesian Networks, the log-likelihood decomposes over RV families:

$$\mathcal{L} = \sum_{n=1}^{N} \log \prod_{i=1}^{D} p(x_i^{(n)} \mid \boldsymbol{x}_{\mathbf{pa}_i}^{(n)}, \boldsymbol{\theta}_i) = \sum_{n=1}^{N} \sum_{i=1}^{D} \log p(x_i^{(n)} \mid \boldsymbol{x}_{\mathbf{pa}_i}^{(n)}, \boldsymbol{\theta}_i)$$

$$= \sum_{i=1}^{D} \underbrace{\sum_{n=1}^{N} \log p(x_i^{(n)} \mid \boldsymbol{x}_{\mathbf{pa}_i}^{(n)}, \boldsymbol{\theta}_i)}_{\mathcal{L}_i\text{: local log-likelihood for } X_i}$$

Now, since a GMM **is** a Bayesian network over Z and $\boldsymbol{X}$, we can learn **independently**

- $p(z)$
- $p(\boldsymbol{x} \mid z)$



Learning GMM with Complete Data

Example

Recall the MLE for Categorical:

$$\hat{w}_k = \frac{\sum_{i=1}^{N} \mathbb{1}[z^{(i)} = k]}{N}$$

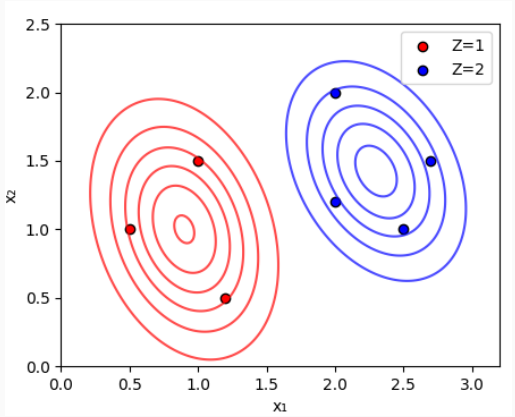
and for multivariate Gaussian

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}^{(i)} \quad \hat{\Sigma} = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{x}^{(i)} - \hat{\boldsymbol{\mu}})(\boldsymbol{x}^{(i)} - \hat{\boldsymbol{\mu}})^T$$

We fit Gaussians for each value of Z.

X_1	X_2	Z
1.2	0.5	1
2.7	1.5	2
2	1.2	2
0.5	1	1
1	1.5	1
2	2	2
2.5	1	2

	$z = 1$	$z = 2$
w	$3/7$	$4/7$
$\boldsymbol{\mu}$	$(0.9, 1)^T$	$(2.3, 1.425)^T$
$\boldsymbol{\Sigma}$	$\begin{pmatrix} 0.13 & -0.05 \\ -0.05 & 0.25 \end{pmatrix}$	$\begin{pmatrix} 0.17 & -0.05 \\ -0.05 & 0.19 \end{pmatrix}$



Globally optimal solution for complete data!

How to deal with missing Z's?

However, Z's are latent and hence missing.

A common heuristic is to "invent" suitable Z's, for example by **K-means clustering**.

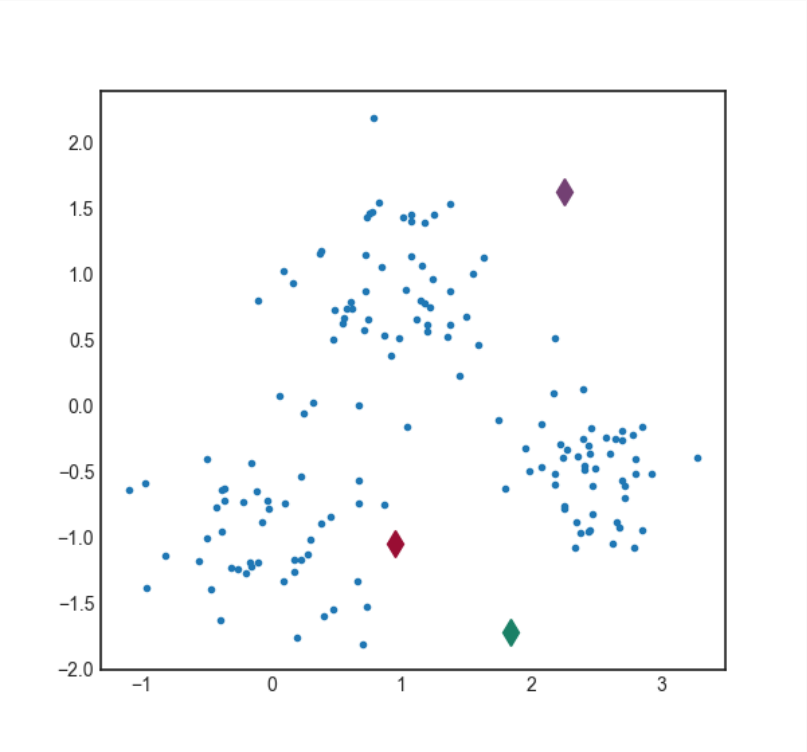
A Crash-Course in K-Means

- the **K-means** algorithm aims to cluster a dataset $\mathcal{D} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\}$ into K clusters (groups of similar items)
- it introduces **cluster centers** $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$
- k^{th} cluster is the set of samples closest to $\boldsymbol{\mu}_k$:

$$\mathcal{D}_k = \left\{ \boldsymbol{x} \in \mathcal{D} \,|\, k = \arg\min_l \|\boldsymbol{x} - \boldsymbol{\mu}^{(l)}\|_2 \right\}$$

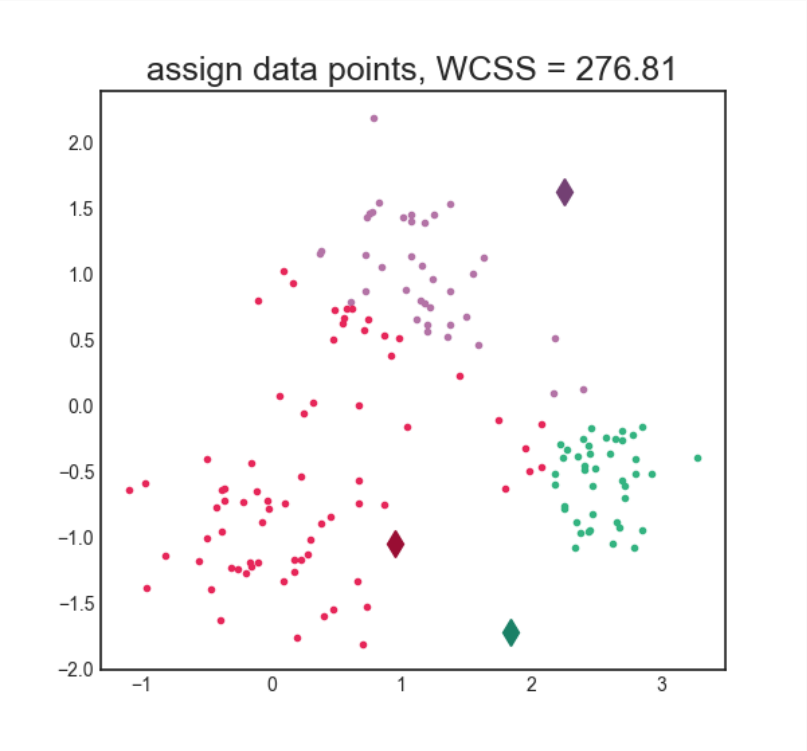
- **chicken-egg problem**:
  - if $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ were known, clusters are determined
  - if cluster were known, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ are simply the cluster means
- K-means initializes randomly and iterates between these steps

We pick $K = 3$ and initialize $\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \boldsymbol{\mu}^{(3)}$, e.g. by drawing them uniformly between minimal and maximal range in each dimension.



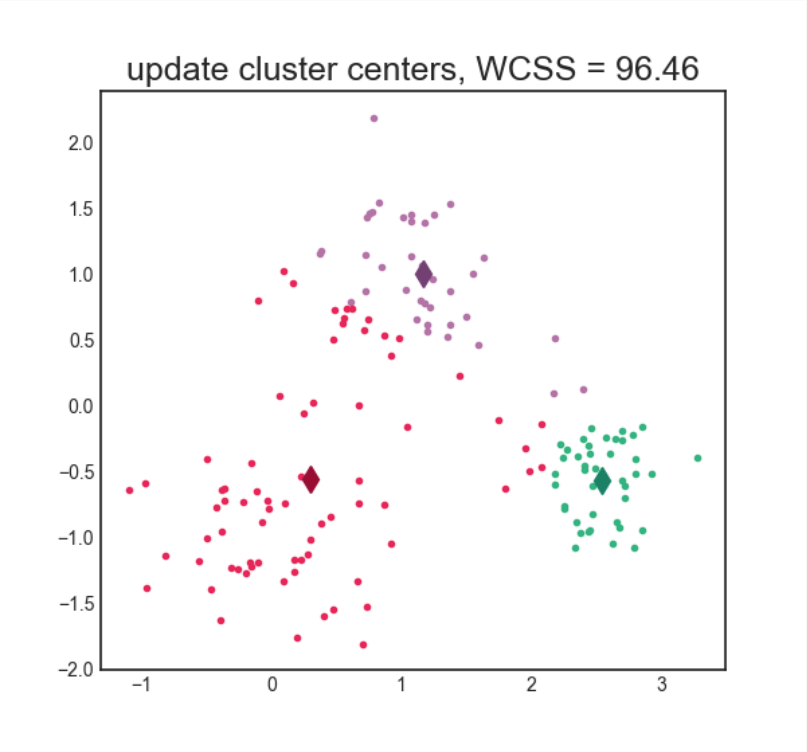
Given $\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \boldsymbol{\mu}^{(3)}$, we set assignment indicators

$$z_{i,k} = \begin{cases} 1 & \text{if cluster center } \boldsymbol{\mu}^{(k)} \text{ is closest to } \boldsymbol{x}^{(i)} \\ 0 & \text{otherwise} \end{cases}$$



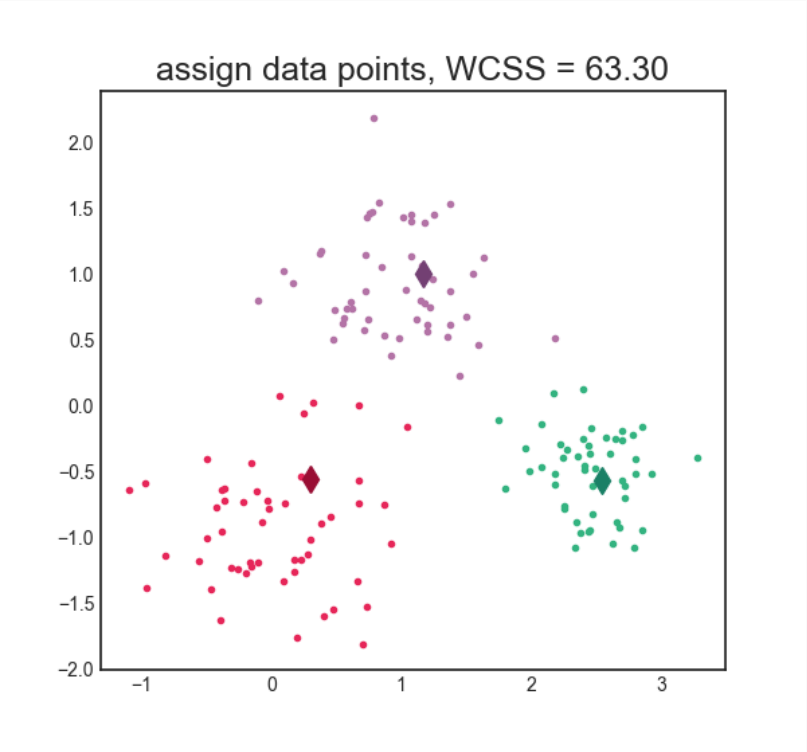
Update cluster centers:

$$\boldsymbol{\mu}^{(k)} \leftarrow \frac{1}{|\mathcal{D}_k|} \sum_{\boldsymbol{x} \in \mathcal{D}_k} \boldsymbol{x}$$



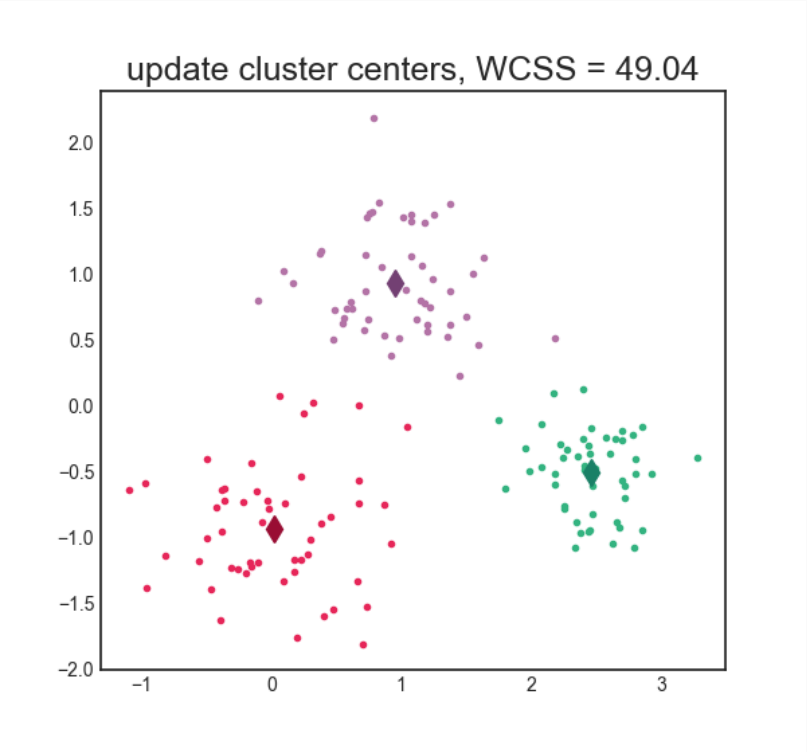
Update assignments:

$$z_{i,k} = \begin{cases} 1 & \text{if cluster center } \boldsymbol{\mu}^{(k)} \text{ is closest to } \boldsymbol{x}^{(i)} \\ 0 & \text{otherwise} \end{cases}$$



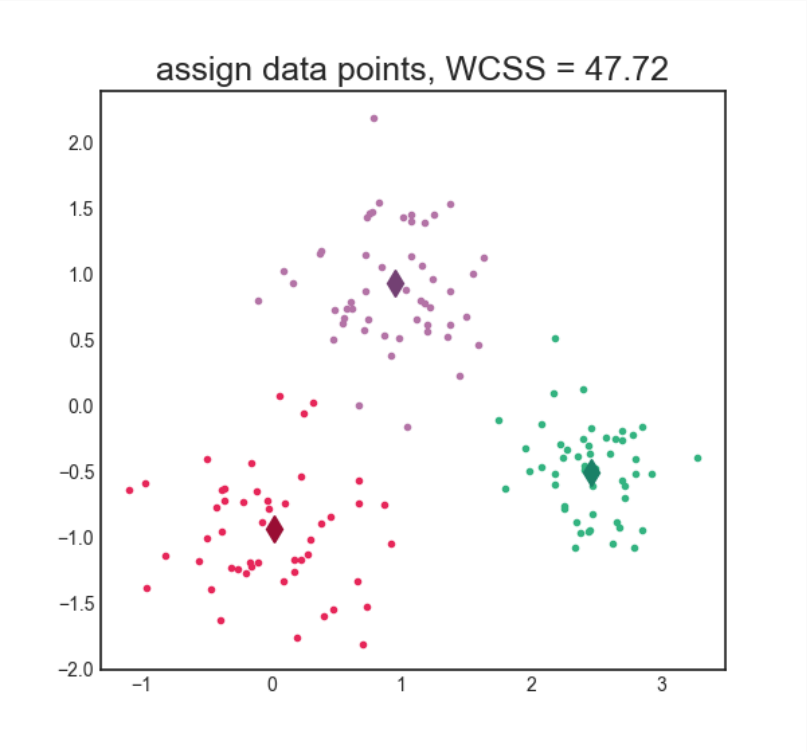
Again, update mean vectors:

$$\boldsymbol{\mu}^{(k)} \leftarrow \frac{1}{|\mathcal{D}_k|} \sum_{\boldsymbol{x} \in \mathcal{D}_k} \boldsymbol{x}$$



Again, the data points to clusters:

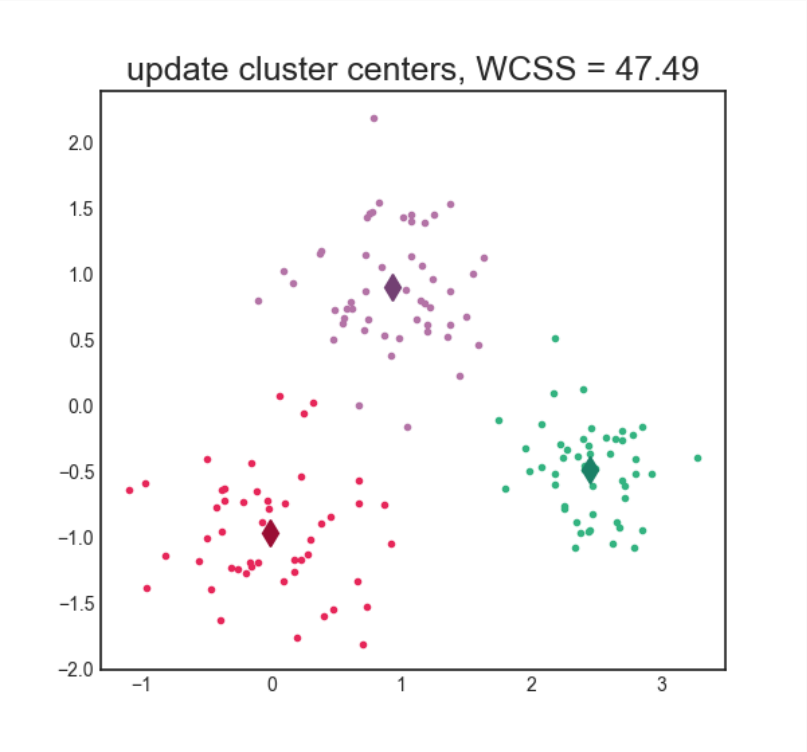
$$z_{i,k} = \begin{cases} 1 & \text{if cluster center } \boldsymbol{\mu}^{(k)} \text{ is closest to } \boldsymbol{x}^{(i)} \\ 0 & \text{otherwise} \end{cases}$$



Update means:

$$\boldsymbol{\mu}^{(k)} \leftarrow \frac{1}{|\mathcal{D}_k|} \sum_{\boldsymbol{x} \in \mathcal{D}_k} \boldsymbol{x}$$

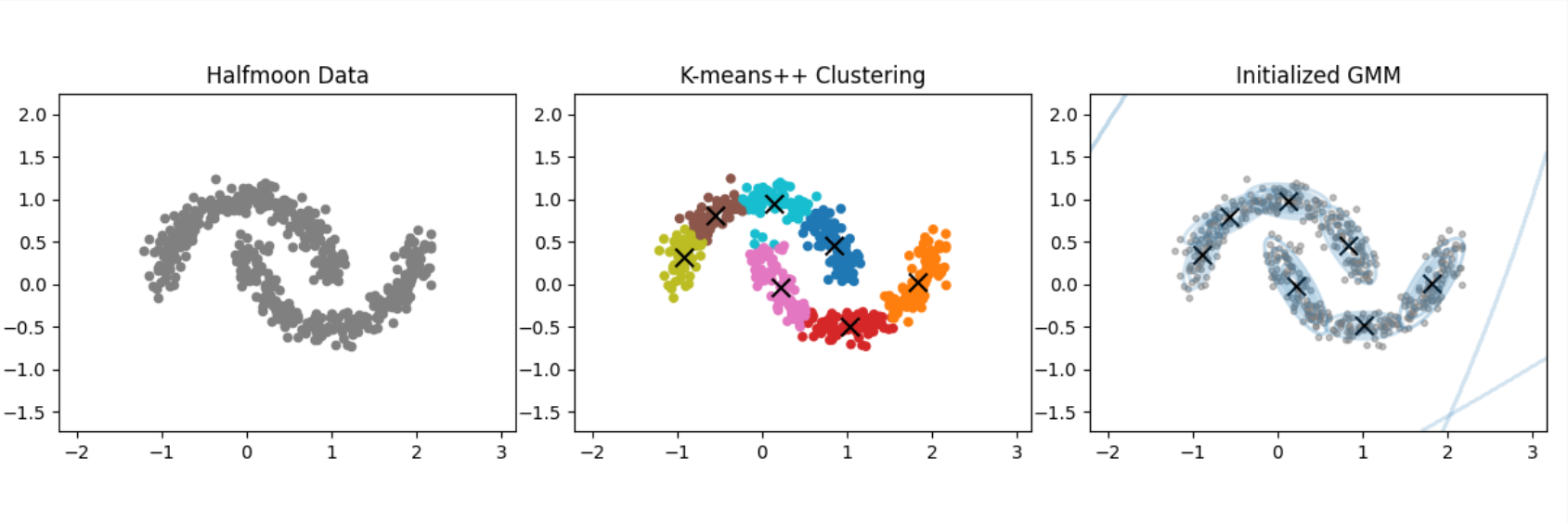
After that, the assignments $\boldsymbol{Z}$ won't change any more. Hence, also $\boldsymbol{\mu}^{(k)}$ won't change any more. The algorithm has **converged.**



K-Means Initialization

K-means initialization for GMMs:

- apply clustering to data $\mathcal{D} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\}$
- use the cluster indices as proxy for Z
- learn GMM on this completed dataset



This is a common and powerful initialization for GMMs.

Maximum Likelihood under Missing Data

- when Z's are observed, we got a simple closed-from solution for the **complete data log-likelihood** $\sum_i \log p(\boldsymbol{x}^{(i)}, z^{(i)})$
  - MLE for $Z \sim$ Categorical
  - MLE for conditional $\boldsymbol{X}|Z \sim$ Gaussian
- K-means heuristic: impute missing Z's and then apply complete data MLE
- however, in fact Z's are not observed and the goal would be to optimize the **marginal log-likelihood**:

$$\sum_{i=1}^{N} \log p(\boldsymbol{x}^{(i)}) = \sum_{i=1}^{N} \log \sum_{z} \underbrace{p(z)}_{w_z} \underbrace{p(\boldsymbol{x}^{(i)} \mid z)}_{p(\boldsymbol{x}^{(i)} \mid \boldsymbol{\mu}_z, \Sigma_z)},$$

Optimizing the Marginal Log-Likelihood

- the (marginal) log-likelihood in GMMs is non-convex
- finding a global optimum: **NP-hard** problem
- first idea:
  - initialize parameters randomly
  - apply gradient ascent
- this idea indeed works, but care needs to be taken about constraints

$$w_k \geq 0 \quad \sum_k w_k = 1 \quad \text{(softmax re-parameterization)}$$

$$\Sigma_k \text{ positive definite} \quad \text{(low-rank + diagonal re-parameterization)}$$

Expectation-Maximization

- **expectation maximization (EM)** is a general and powerful approach to MLE under missing data
- it applies to **any parametric model** and **any missingness pattern**
- does not introduce hyperparameters such as stepsize
- updates are **monotonic** in the marginal log-likelihood, i.e. they can only increase the log-likelihood
- in GMMs, it is yields simple update rules and takes care about constraints
- coincidentally, it is very similar to K-means

Expectations Maximization Setting (in general

- assume some parametric joint model (not necessarily GMM):

$$p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta})$$

  over **observed** RVs $\boldsymbol{X}$ and **missing**/**latent** RVs $\boldsymbol{Z}$
- in general, the **missingness pattern does not need to be fixed**—the role of observed and missing RVs is allowed to change between samples! (here we assume it fixed though, for simplicity)
- **assumption:** the **complete log-likelihood** is easy to optimize, i.e. we can solve

$$\arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{N} \log p(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)} \mid \boldsymbol{\theta})$$

Expectations Maximization (in general)

- **initialize** parameters $\boldsymbol{\theta}$ (e.g. randomly, heuristically)
- **iterate** until marginal log-likelihood converges:
  - **step 1** — using the current model, **infer** the posterior over missing RVs $\boldsymbol{Z}^{(i)}$ given observed variables $\boldsymbol{X}^{(i)}$:

$$p(\boldsymbol{z}^{(i)} \mid \boldsymbol{x}^{(i)}, \boldsymbol{\theta}_{\text{cur}})$$

  - **step 2** — use posterior to compute **expected complete log-likelihood**:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{cur}}) = \sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{z}^{(i)} \mid \boldsymbol{x}^{(i)}, \boldsymbol{\theta}_{\text{cur}}} \left[\log p(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)} \mid \boldsymbol{\theta}) \right]$$

  - **step 3** — maximize:

$$\boldsymbol{\theta}_{\text{new}} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{cur}}) \quad \text{and let } \boldsymbol{\theta}_{\text{cur}} \leftarrow \boldsymbol{\theta}_{\text{new}}$$

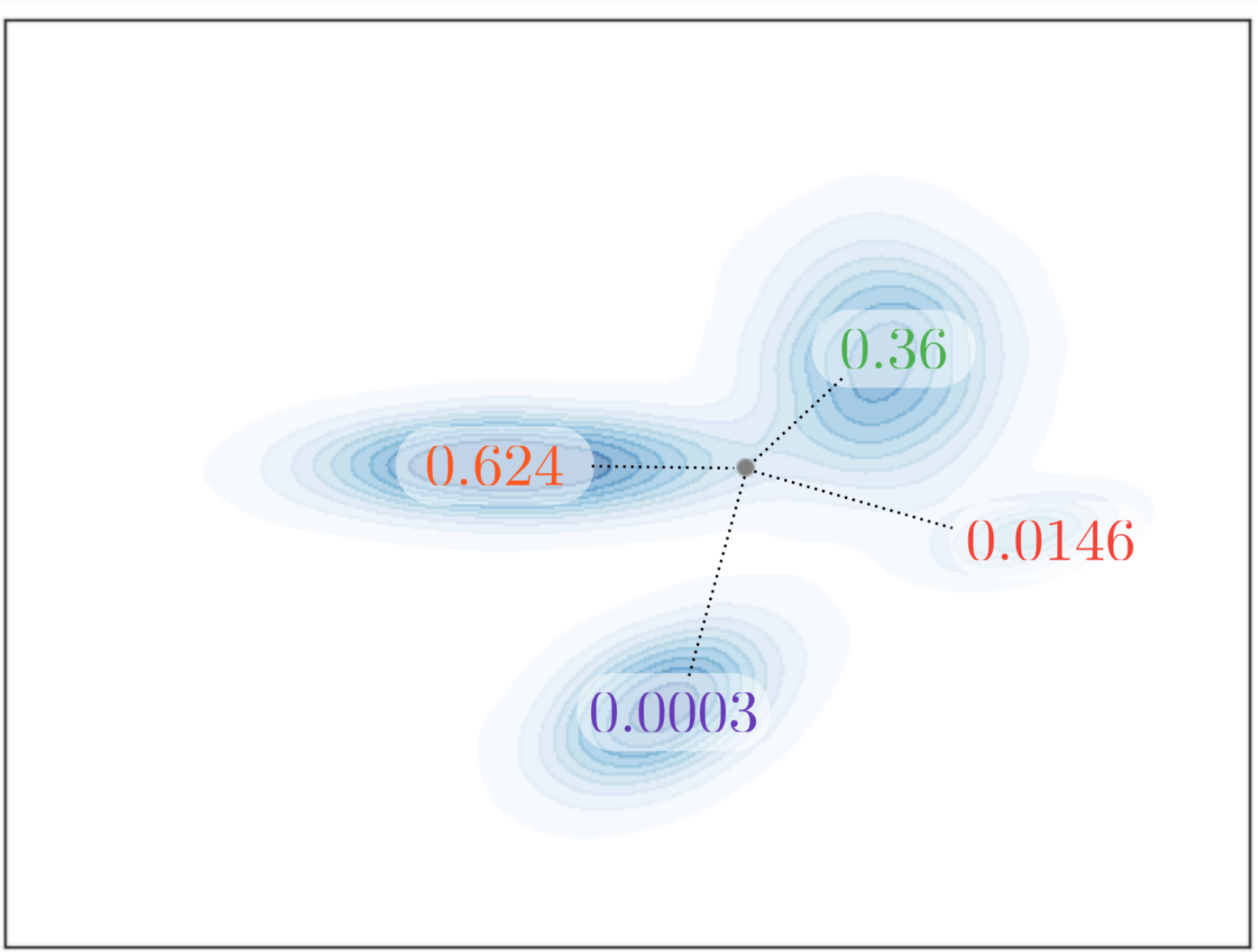
Expectation Maximization (for GMMs)

Step 1 — compute posterior $p(z^{(i)} \mid \boldsymbol{x}^{(i)}, \boldsymbol{\theta}_{\text{cur}})$

In GMMs, there is just a single missing $Z \in \{1, \ldots, K\}$. Bayes rule yields a closed-form:

$$p(z^{(i)} = k \mid \boldsymbol{x}^{(i)}) = \frac{\overbrace{p(z^{(i)} = k)}^{w_k} \; \overbrace{p(\boldsymbol{x}^{(i)} \mid z^{(i)} = k)}^{k^{\text{th}} \text{ Gaussian eval. on } \boldsymbol{x}^{(i)}}}{\underbrace{\sum_{z'} p(z')\, p(\boldsymbol{x}^{(i)} \mid z')}_{\text{GMM eval. on } \boldsymbol{x}^{(i)}}}$$

Let us use a shorthand $\gamma_{i,k} = p(z^{(i)} = k \mid \boldsymbol{x}^{(i)})$, often called **components responsibilities**. They might be interpreted as "fractions of a sample".


0.36
0.624
0.0146
0.0003

Expectation Maximization (for GMMs) cont'd

Step 2 — expected complete log-likelihood

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{cur}}) = \sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{z}^{(i)}|\boldsymbol{x}^{(i)},\boldsymbol{\theta}_{\text{cur}}} \left[\log p(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)} \mid \boldsymbol{\theta}) \right]$$
$$= \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{i,k} \log p(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)} \mid \boldsymbol{\theta})$$

Note: in the case of observed Z (i.e. when GMM learning is easy) the complete log-likelihood can **be written exactly the same,** where

- $\gamma_{i,k}$ are "hard", i.e. a one-hot encoding, indicating to which component $\boldsymbol{x}^{(i)}$ belongs to
- $\gamma_{i,k}$ are fixed and do not depend on $\boldsymbol{\theta}_{\text{cur}}$

Expectation Maximization (for GMMs) cont'd

Step 3 — maximize

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{cur}}) = \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{i,k} \log p(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)} \mid \boldsymbol{\theta})$$

This is really just a weighted version of the maximum likelihood problem for complete data. Without proof, the MLE is

$$\hat{w}_k = \frac{\sum_{i=1}^{N} \gamma_{i,k}}{N}$$

$$\hat{\boldsymbol{\mu}}_k = \frac{\sum_{i=1}^{N} \gamma_{i,k} \boldsymbol{x}^{(i)}}{\sum_{i=1}^{N} \gamma_{i,k}}$$

$$\hat{\boldsymbol{\Sigma}}_k = \frac{\sum_{i=1}^{N} \gamma_{i,k} (\boldsymbol{x}^{(i)} - \hat{\boldsymbol{\mu}}_k)(\boldsymbol{x}^{(i)} - \hat{\boldsymbol{\mu}}_k)^T}{\sum_{i=1}^{N} \gamma_{i,k}}$$

Note: in the case of observed Z, the solution has actually the same form, just with "hard" $(0, 1)$ and fixed $\gamma_{i,k}$'s

Expectation Maximization Algorithm for GMMs

input: Data $\mathcal{D} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}\}$

output: Learned GMM parameters $\boldsymbol{\theta} = \{w_k, \boldsymbol{\mu}_k, \Sigma_k\}_{k=1}^{K}$

Initialize $\boldsymbol{\theta} = \{w_k, \boldsymbol{\mu}_k, \Sigma_k\}_{k=1}^{K}$ randomly

for *counter* $= 1 \ldots$ *max number of iterations* **do**

  E-Step

  for $i = 1 \ldots N$, $k = 1 \ldots K$ **do**

    let $\gamma_{i,k} \leftarrow \frac{p(\boldsymbol{x}^{(i)} \mid \boldsymbol{\mu}_k, \Sigma_k)\, w_k}{\sum_{k'} p(\boldsymbol{x}^{(i)} \mid \boldsymbol{\mu}_{k'}, \Sigma_{k'})\, w_{k'}}$ ▷ cluster responsibilities

  end

  M-Step

  for $k = 1 \ldots K$ **do**

    $\boldsymbol{\mu}_k \leftarrow \frac{\sum_{i=1}^{N} \gamma_{i,k}\, \boldsymbol{x}^{(i)}}{\sum_{i=1}^{N} \gamma_{i,k}}$ ▷ weighted mean

    $\Sigma_k \leftarrow \frac{\sum_{i=1}^{N} \gamma_{i,k} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_k)(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_k)^T}{\sum_{i=1}^{N} \gamma_{i,k}}$ ▷ weighted covariance

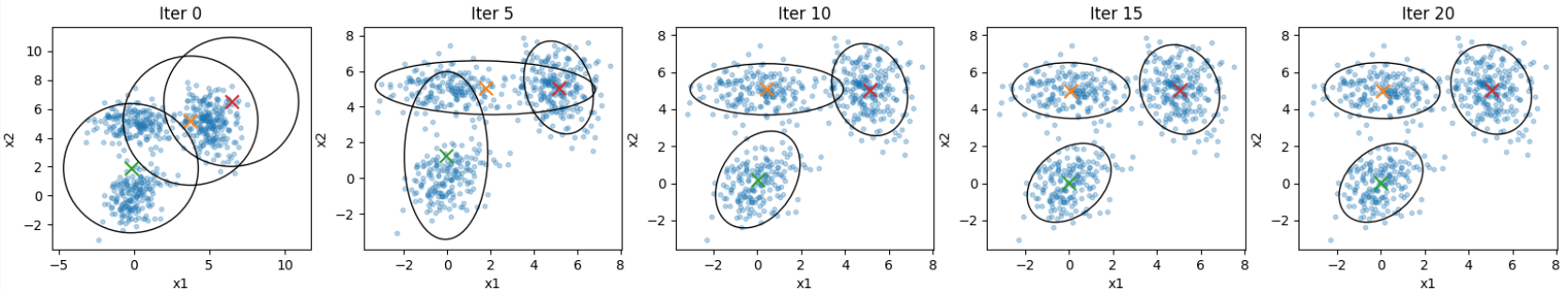
    $w_k \leftarrow \frac{\sum_{i=1}^{N} \gamma_{i,k}}{N}$ ▷ weighted empirical frequencies

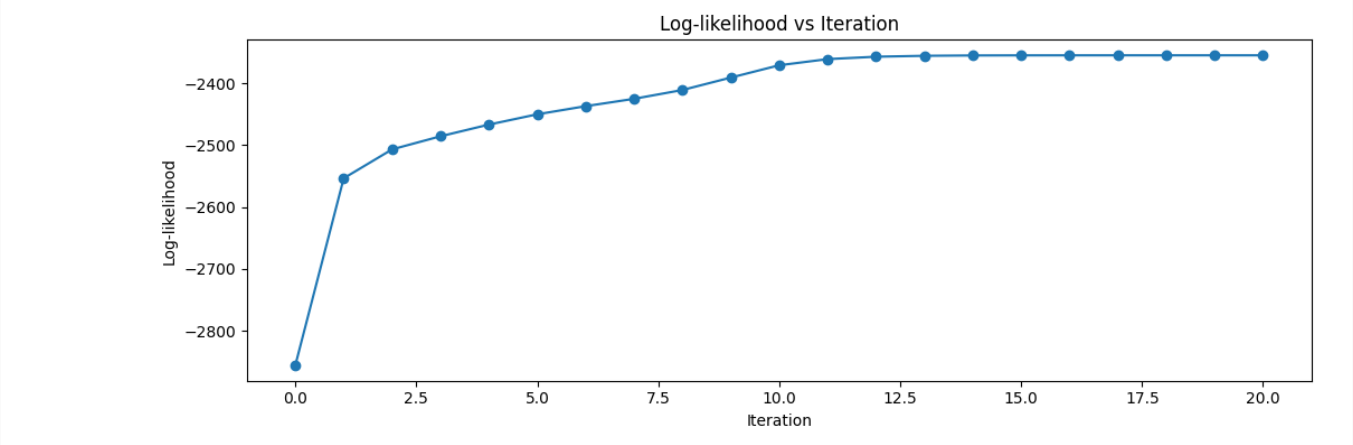
  end

end

Initialization

- uniform weights
- random data points as means
- identity matrix as covariances

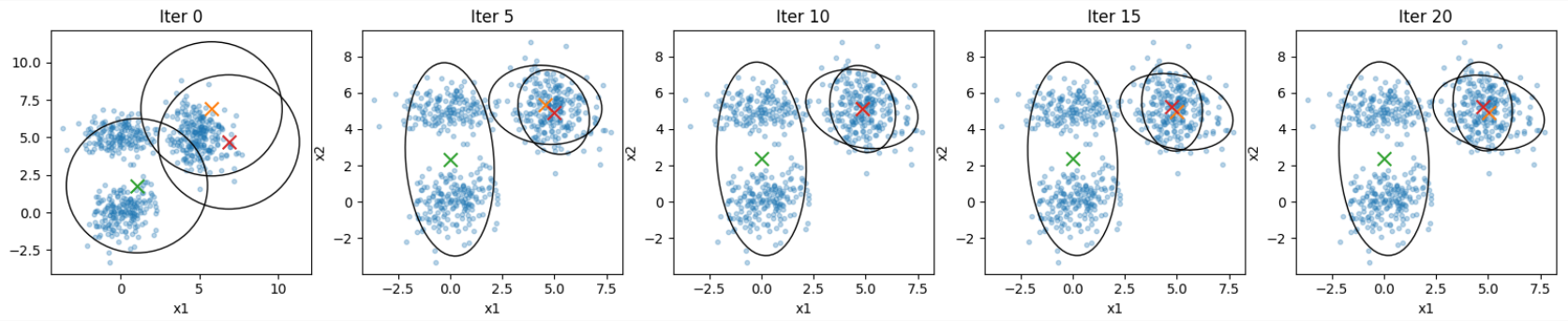


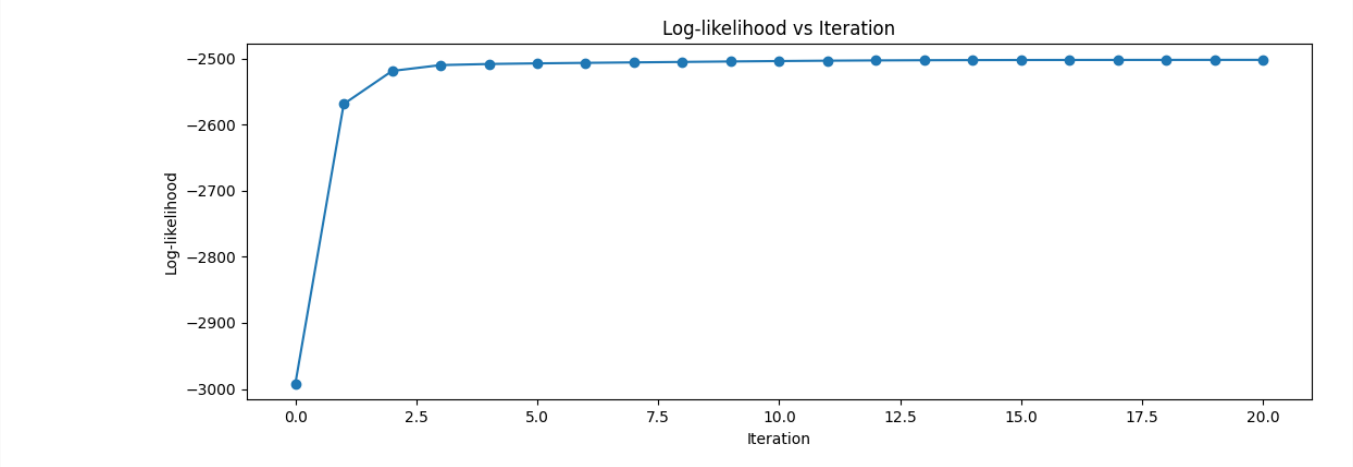


Bad Local Maxima

Example

- EM monotonically increases the log-likelihood
- but it can run into pretty bad local maxima
- mitigation:
  - initialization with K-means++
  - random restarts

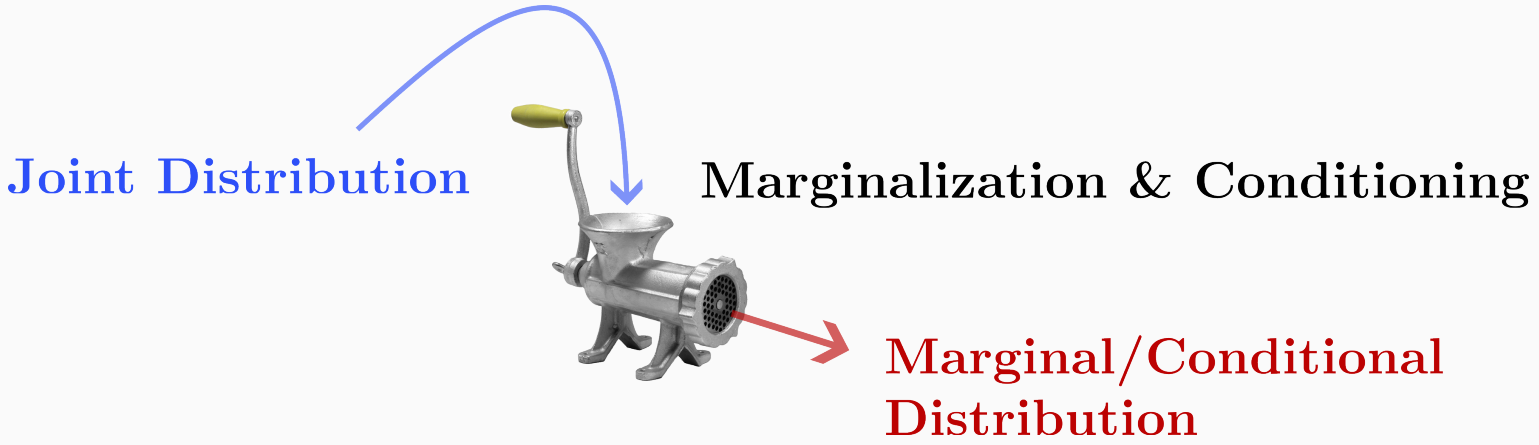




Inference

Inference in GMMs

- GMMs are universal approximators of densities
- we discussed how they can be learned from data
- what about **inference**?



- it turns out that many inference tasks in GMMs are **tractable**, i.e. can be computed **exactly and efficiently**
- they basically "inherit" tractability from their Gaussian components

Gaussian Marginal Distribution

Marginalizing a single Gaussian is easy: one simply needs to discard the parameters mentioning marginalized RVs.

For example, a Gaussian over 5 random variables X_1, X_2, X_3, X_4, X_5, the marginal over X_1, X_4, X_5 is Gaussian with parameters $\boldsymbol{\mu}_q$ and Σ_{qq} obtained by deleting rows and columns corresponding to X_2, X_3:

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \cancel{\mu_2} \\ \cancel{\mu_3} \\ \mu_4 \\ \mu_5 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} v_{1,1} & v_{1,2} & v_{1,3} & v_{1,4} & v_{1,5} \\ \cancel{v_{2,1}} & \cancel{v_{2,2}} & \cancel{v_{2,3}} & \cancel{v_{2,4}} & \cancel{v_{2,5}} \\ \cancel{v_{3,1}} & \cancel{v_{3,2}} & \cancel{v_{3,3}} & \cancel{v_{3,4}} & \cancel{v_{3,5}} \\ v_{4,1} & v_{4,2} & v_{4,3} & v_{4,4} & v_{4,5} \\ v_{5,1} & v_{5,2} & v_{5,3} & v_{5,4} & v_{5,5} \end{pmatrix}$$

$$\boldsymbol{\mu}_q = \begin{pmatrix} \mu_1 \\ \mu_4 \\ \mu_5 \end{pmatrix} \qquad \Sigma_{qq} = \begin{pmatrix} v_{1,1} & v_{1,4} & v_{1,5} \\ v_{4,1} & v_{4,4} & v_{4,5} \\ v_{5,1} & v_{5,4} & v_{5,5} \end{pmatrix}$$

GMM Marginal Distribution

The marginals of GMMs are also easy:

$$
\begin{aligned}
p(\boldsymbol{x}_q) &= \int \sum_k w_k \, p(\boldsymbol{x}_q, \boldsymbol{x}_m \mid \boldsymbol{\theta}_k) \, \mathrm{d}\boldsymbol{x}_m \\
&= \sum_k w_k \underbrace{\int p(\boldsymbol{x}_q, \boldsymbol{x}_m \mid \boldsymbol{\theta}_k) \, \mathrm{d}\boldsymbol{x}_m}_{\text{Gaussian marginal}} \\
&= \sum_k w_k \, p(\boldsymbol{x}_q \mid \boldsymbol{\theta}_k)
\end{aligned}
$$

Here we actually didn't use the fact that components are Gaussian—as long as each component has tractable marginals, the mixture has tractable marginals as well.

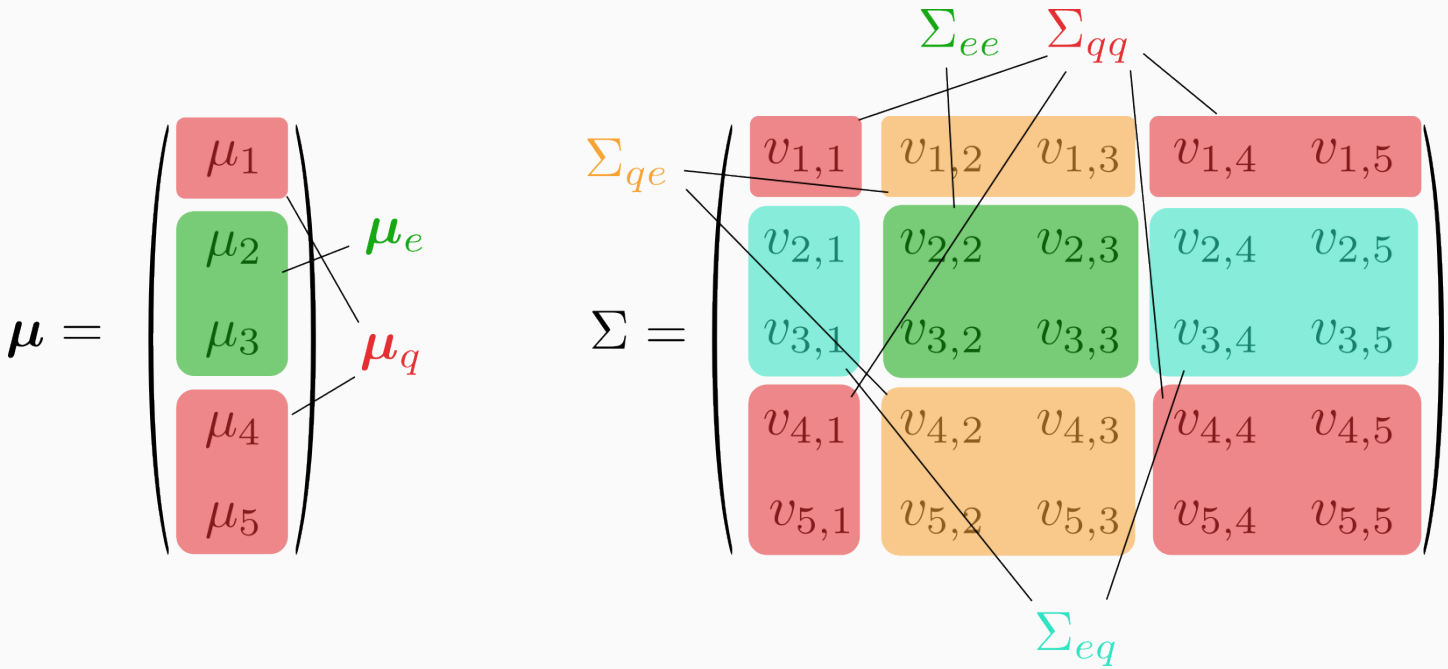
marginal of mixture = mixture of marginals!

Gaussian Conditional Distributions

In Gaussians with parameters $\boldsymbol{\mu}$ and Σ, any conditional distribution $p(\boldsymbol{x}_q \mid \boldsymbol{x}_e)$ is again Gaussian with **closed form parameters**:

$$\boldsymbol{\mu}_{q|e} = \boldsymbol{\mu}_q + \Sigma_{qe}\Sigma_{ee}^{-1}(\boldsymbol{x}_e - \boldsymbol{\mu}_e)$$
$$\Sigma_{q|e} = \Sigma_{qq} - \Sigma_{qe}\Sigma_{ee}^{-1}\Sigma_{eq}$$

For **q**uery variables X_1, X_4, X_5 and **e**vidence variables X_2, X_3:



GMM Conditional Distribution

The conditionals of GMMs are also quite easy. We use the fact that the conditional is proportional to the joint, $p_{GMM}(\boldsymbol{x}_q \mid \boldsymbol{x}_e) \propto p_{GMM}(\boldsymbol{x}_q, \boldsymbol{x}_e)$:

$$
\begin{aligned}
p_{GMM}(\boldsymbol{x}_q \mid \boldsymbol{x}_e) &\propto \sum_{k=1}^{K} w_k \, p(\boldsymbol{x}_q, \boldsymbol{x}_e \mid \boldsymbol{\theta}_k) \\
&= \sum_{k=1}^{K} w_k \, \overbrace{p(\boldsymbol{x}_q \mid \boldsymbol{x}_e, \boldsymbol{\theta}_k) \, p(\boldsymbol{x}_e \mid \boldsymbol{\theta}_k)}^{\text{chain rule}} \\
&= \sum_{k=1}^{K} \overbrace{w_k \, \underbrace{p(\boldsymbol{x}_e \mid \boldsymbol{\theta}_k)}_{\text{marginal Gauss}}}^{\text{not normalized}} \overbrace{p(\boldsymbol{x}_q \mid \boldsymbol{x}_e, \boldsymbol{\theta}_k)}^{\text{cond. Gauss}}
\end{aligned}
$$

We can normalize by replacing $w_k \, p(\boldsymbol{x}_e \mid \boldsymbol{\theta}_k)$ with $\tilde{w}_k = \frac{w_k \, p(\boldsymbol{x}_e \mid \boldsymbol{\theta}_k)}{\sum_i w_i \, p(\boldsymbol{x}_e \mid \boldsymbol{\theta}_i)}$.
Hence

$$
p_{GMM}(\boldsymbol{x}_q \mid \boldsymbol{x}_e) = \sum_{k=1}^{K} \tilde{w}_k \, p(\boldsymbol{x}_q \mid \boldsymbol{x}_e, \boldsymbol{\theta}_k)
$$

Conditional GMM cont'd

$$p_{GMM}(\boldsymbol{x}_q \mid \boldsymbol{x}_e) = \sum_{k=1}^{K} \tilde{w}_k \, p_k(\boldsymbol{x}_q \mid \boldsymbol{x}_e, \boldsymbol{\theta}_k)$$

Again, we did not use the fact that the components are Gaussian—only that we know how to marginalize and condition them.

Conditional of mixture = a mixture of conditionals!

. . . with modified weights

$$\tilde{w}_k = \frac{w_k \, p(\boldsymbol{x}_e \mid \boldsymbol{\theta}_k)}{\sum_i w_i \, p(\boldsymbol{x}_e \mid \boldsymbol{\theta}_i)}$$

On GANs and GMMs

Eitan Richardson
School of Computer Science and Engineering
The Hebrew University of Jerusalem
Jerusalem, Israel
eitanrich@cs.huji.ac.il

Yair Weiss
School of Computer Science and Engineering
The Hebrew University of Jerusalem
Jerusalem, Israel
yweiss@cs.huji.ac.il

- this paper demonstrates that good old GMMs can deliver decent generative models
- uses **low-rank + diagonal** covariances

https://github.com/eitanrich/gans-n-gmms

Implementation

Covariances

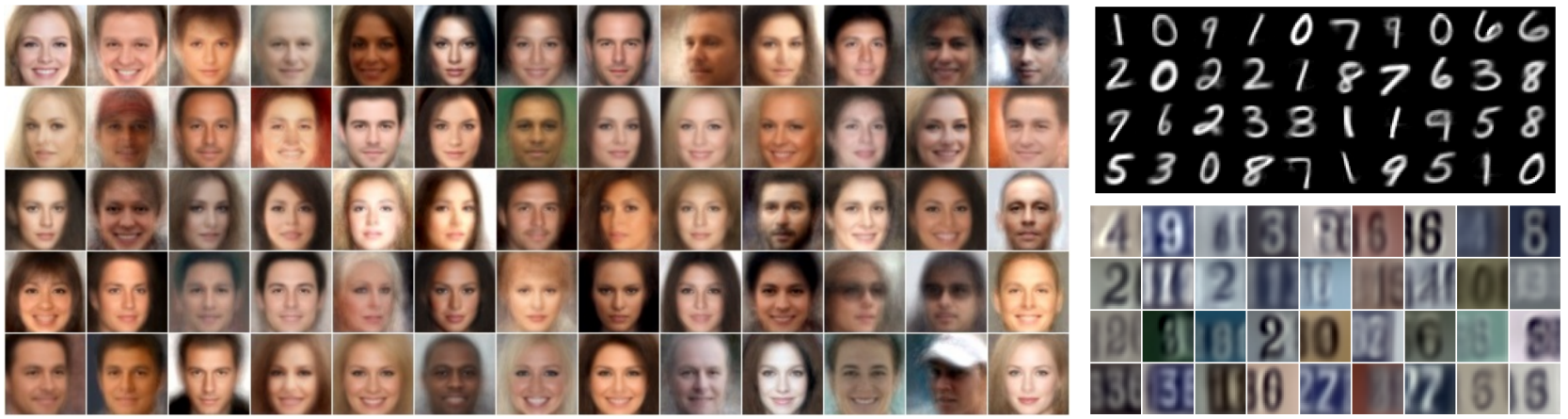
- each covariance matrix is represented as

$$\Sigma = AA^T + \text{diag}(\boldsymbol{d})$$

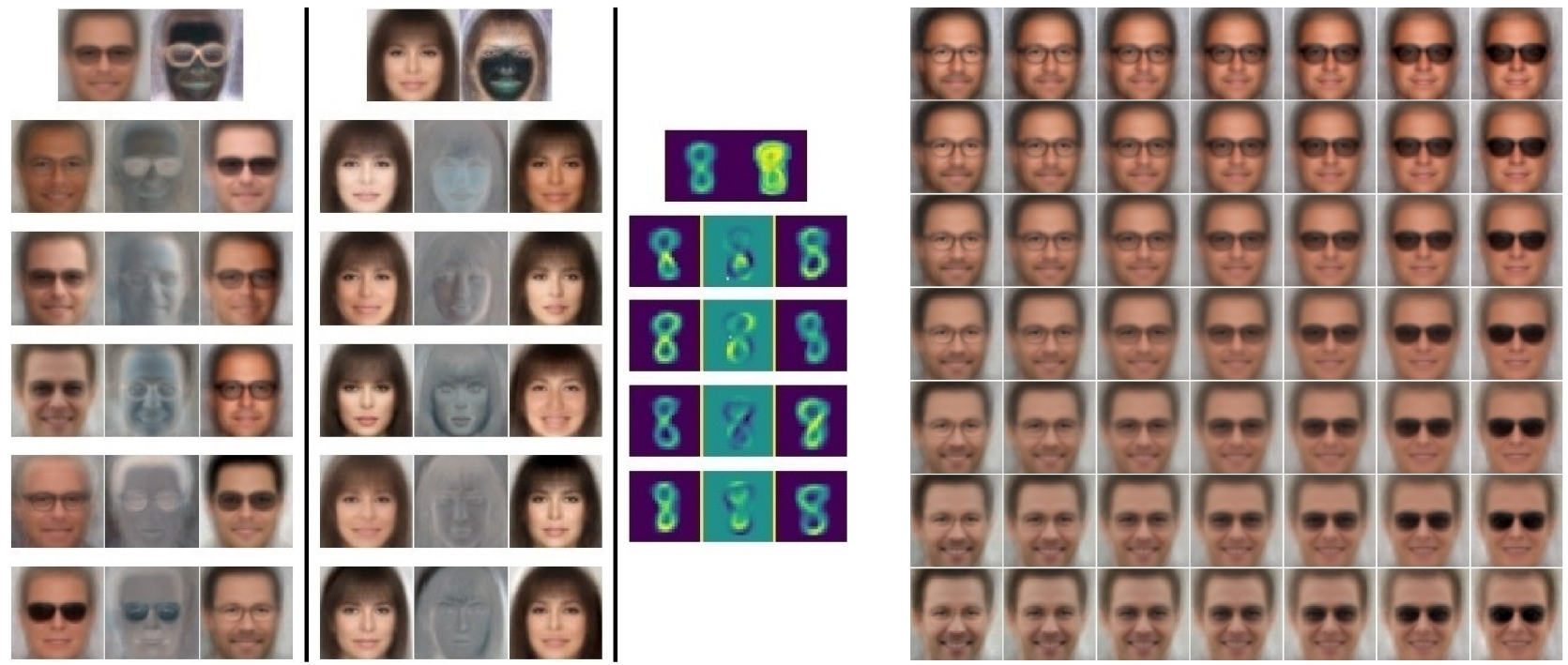
  where $A = \mathbb{R}^{D \times L}$ with $L < D$
- Σ constructed in this way is always positive definite
- adding a diagonal matrix $\text{diag}(\boldsymbol{d})$ with strictly positive elements in the diagonal ensures that Σ is invertible

Training

Maximum likelihood with Adam on GPU. Later versions also include an EM implementation.



Samples on *CelebA*, *MNIST*, and *Street View House Numbers* are decent, albeit a bit blurry and oversmoothed



Left: mean and diagonal of a Gaussian components; each row visualizes a column of the A-matrix (mean minus column, column, mean plus column)

Right: superpositions of two columns

By selecting data samples with low $\log p$ one can detect outliers and peculiar examples.

Image Inpainting

Example

A GMM is one of the few models that allow **tractable inference**, such as computing **exact conditionals**. This can be used for **data restoration** and **inpainting**, by sampling from the conditional, or by taking the conditional expectation.