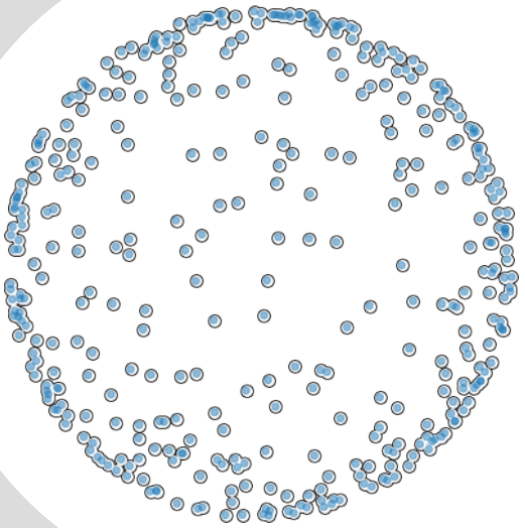
WiSe 2024/25

Machine Learning for Data Science

Lecture by G. Montavon

FREIE UNIVERSITÄT BERLIN



Lecture 9b

Probabilistic Models (cont.)

Outline

Modeling Sequential Data

- Motivations (recap)
- Tikhonov regularization (recap)
- Probabilistic approach: Gaussian processes

Modeling Clustered Data

- Motivations (recap)
- The K-means algorithm (recap)
- Probabilistic approach: Gaussian mixture model

Part 1

Modeling Sequential Data

Recap: Motivations

- Sequential data is found in many practical applications.
- It can be time series, images, geographical maps, etc.
- Sequential data exhibits correlation between adjacent instances $\rightarrow$ iid. assumption does not hold, and we need ad-hoc ML models.

Browse Through: **38** Data Sets

Table View List View

Default Task
Classification (27)
Regression (23)
Clustering (7)
Other (0)

Attribute Type
Categorical (0)
Numerical (37)
Mixed (1)

Data Type - Undo
Multivariate (137)
Univariate (5)
Sequential (21)
Time-Series (38)
Text (10)
Domain-Theory (2)
Other (3)

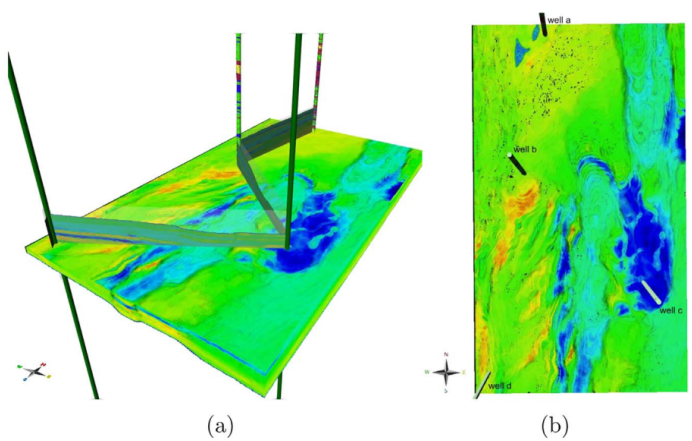
Area
Life Sciences (3)
Physical Sciences (5)
CS / Engineering (22)
Social Sciences (1)
Business (1)
Game (0)
Other (6)

 # Attributes - Undo
Less than 10 (27)
10 to 100 (38)
Greater than 100 (21)

 # Instances - Undo
Less than 100 (3)
100 to 1000 (8)
Greater than 1000 (38)

Format Type
Matrix (31)
Non-Matrix (7)

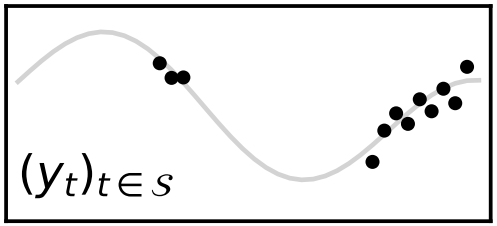
Name	Data Types	Default Task	Attribute Types	# Instances	# Attributes	Year
AI4I 2020 Predictive Maintenance Dataset	Multivariate, Time-Series	Classification, Regression, Causal-Discovery	Real	10000	14	2020
Air Quality	Multivariate, Time-Series	Regression	Real	9358	15	2016
Air quality	Multivariate, Time-Series	Regression	Real	9358	15	2016
Appliances energy prediction	Multivariate, Time-Series	Regression	Real	19735	29	2017
Australian Sign Language signs	Multivariate, Time-Series	Classification	Categorical, Real	6650	15	1999
Australian Sign Language signs (High Quality)	Multivariate, Time-Series	Classification	Real	2565	22	2002
Beijing Multi-Site Air-Quality Data	Multivariate, Time-Series	Regression	Integer, Real	420768	18	2019
Beijing PM2.5 Data	Multivariate, Time-Series	Regression	Integer, Real	43824	13	2017
BitcoinHeistRansomwareAddressDataset	Multivariate, Time-Series	Classification, Clustering	Integer, Real	2916697	10	2020
BLE RSSI Dataset for Indoor localization and Navigation	Multivariate, Sequential, Time-Series	Classification, Clustering	Integer	6611	15	2018



(a) (b)

Image source: DOI:10.1016/j.cageo.2017.05.004

Recap: Tikhonov Regularization



Tikhonov regularization:

- Learn a surrogate model $\boldsymbol{z}$ of the true sequence $\boldsymbol{y}$, and enforce correlation between adjacent elements in the sequence through a penalty term.

$$\mathcal{E}(\boldsymbol{z}) = \|M\boldsymbol{z} - \boldsymbol{y}_S\|^2 + \|\Gamma \boldsymbol{z}\|^2$$

The matrix M selects elements of $\boldsymbol{z}$ for which we have observations of $\boldsymbol{y}$, and Γ is a matrix designed by the user and whose rows are typically of the type:

$$\Gamma_{i,:} = (0, \ldots, 0, -1, 1, 0, \ldots, 0) \qquad \text{(penalizing slope)}$$
$$\Gamma_{i,:} = (0, \ldots, 0, -1, 2, -1, 0, \ldots, 0) \qquad \text{(penalizing curvature)}$$

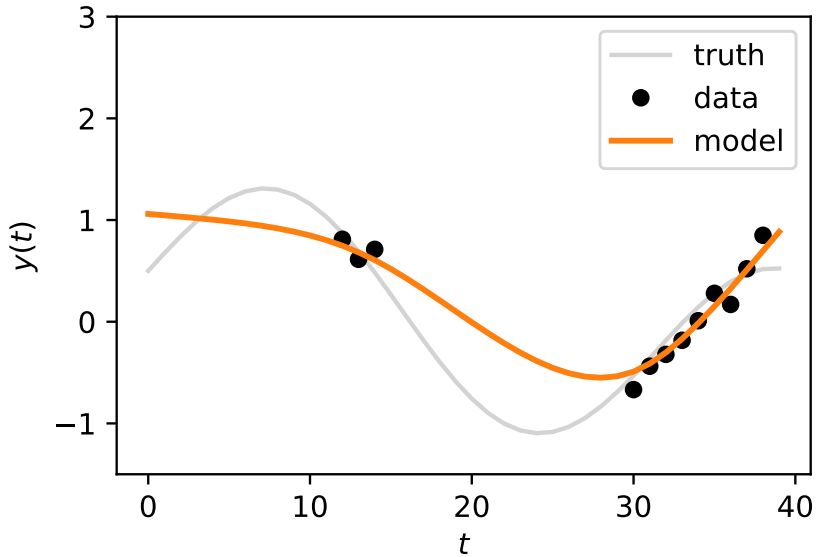
- A minimum of $\mathcal{E}(\boldsymbol{z})$ can be obtained in closed form as:

$$\boldsymbol{z} = \underbrace{(M^\top M + \Gamma^\top \Gamma)^{-1} M^\top}_{W} \boldsymbol{y}_S$$

Recap: Tikhonov Regularization

Example:

- True sequence, observed data, and sequence model with *Tikhonov regularization*.



Observations:

- The model's predicted sequence implements local correlations, and provides plausible interpolations in regions without data.
- It is hard however to assess the level of uncertainty of the model without seeing the groud-truth.

Question:

- Can we use the probabilistic modeling approach to obtain an estimate of predictive uncertainty?

Gaussian Processes

Idea:

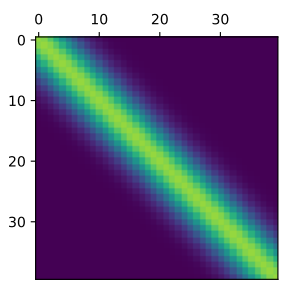
- View the predicted sequence as high-dimensional multivariate Gaussian distribution.

$$\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$$

  with $\boldsymbol{\mu} \in \mathbb{R}^T$ and $\Sigma \in \mathbb{R}^{T \times T}$, and T is the number of time steps in the sequence.

- The parameter $\boldsymbol{\mu}$ is e.g. a constant sequence set to the mean of $\boldsymbol{y}_{\mathcal{S}}$, or something more complex learned from the data.
- The matrix Σ models the covariance between adjacent elements of the sequence, e.g.

$$\Sigma_{ij} = \alpha \cdot \exp(-\gamma \cdot \|i - j\|^2) \qquad \Sigma =$$

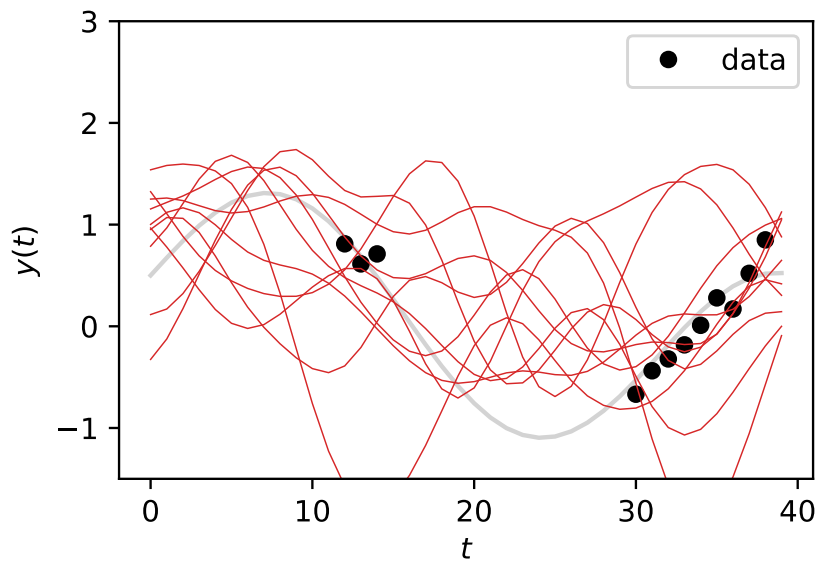


  where i and j are positions in the sequence and (α, γ) are hyperparameters.

Gaussian Process

Example:

- Sequences $\boldsymbol{z}$ sampled from $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$, and shown in red:



Note:

- In practice, to avoid overfitting, we still need account for some possible uncorrelated intrinsic or measurement noise. We can achieve this by adding a diagonal term to our model of convariance: $\Sigma \leftarrow \Sigma + \epsilon^2 I$, where ϵ is the standard deviation of the noise.

Gaussian Process: Inferences

- To 'learn' from the available observations, we condition the Gaussian process to the observed data. We first partition $\boldsymbol{z}$ in terms of observed and unobserved elements (resp. $\boldsymbol{z}_\mathcal{S}$ and $\boldsymbol{z}_\mathcal{T}$):

$$\underbrace{\begin{bmatrix}\boldsymbol{z}_\mathcal{S}\\ \boldsymbol{z}_\mathcal{T}\end{bmatrix}}_{\boldsymbol{z}} \sim \mathcal{N}\Big(\underbrace{\begin{bmatrix}\boldsymbol{\mu}_\mathcal{S}\\ \boldsymbol{\mu}_\mathcal{T}\end{bmatrix}}_{\boldsymbol{\mu}}, \underbrace{\begin{bmatrix}\Sigma_{\mathcal{S}\mathcal{S}} & \Sigma_{\mathcal{S}\mathcal{T}}\\ \Sigma_{\mathcal{T}\mathcal{S}} & \Sigma_{\mathcal{T}\mathcal{T}}\end{bmatrix}}_{\Sigma}\Big)$$

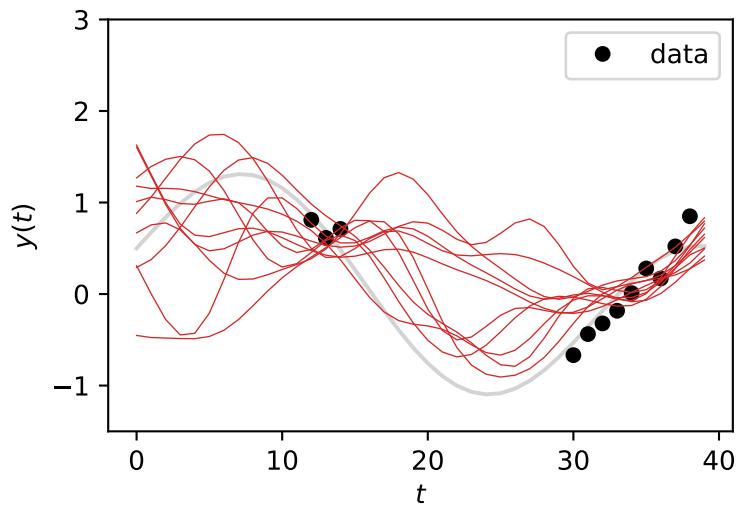
- We now apply the formula for conditioning Gaussian distribution:

$$\boldsymbol{z}_\mathcal{T}|\boldsymbol{z}_\mathcal{S} \sim \mathcal{N}(\boldsymbol{\mu}', \Sigma')$$

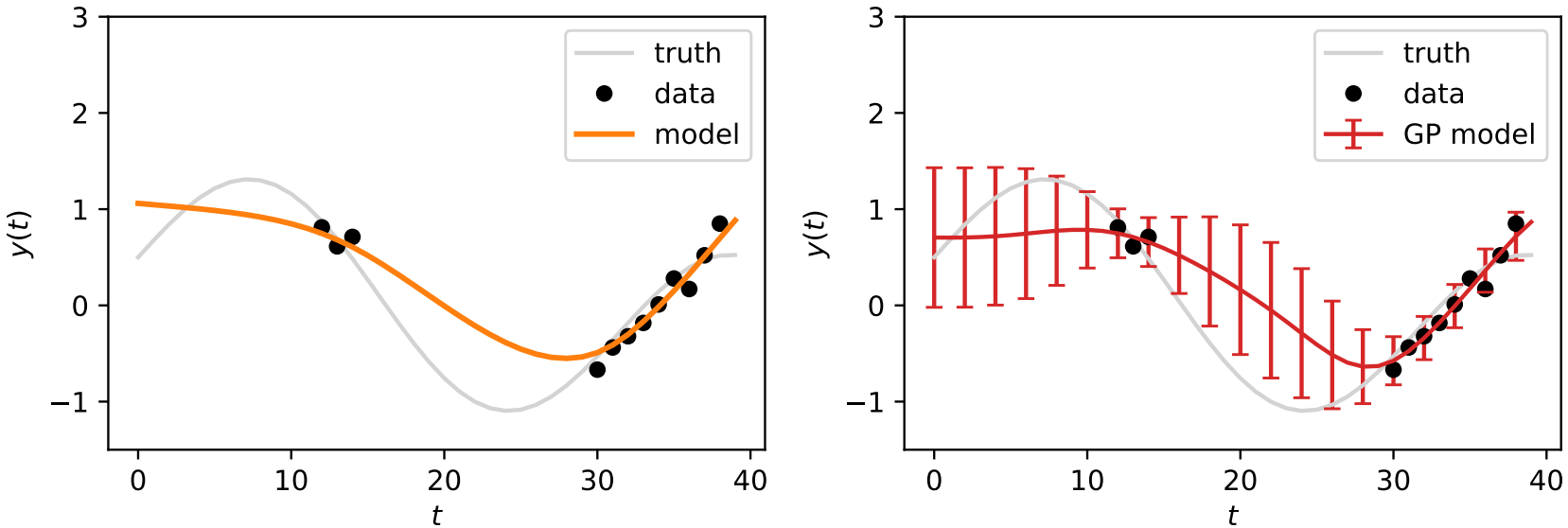
with

$$\boldsymbol{\mu}' = \boldsymbol{\mu}_\mathcal{T} + \Sigma_{\mathcal{T}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}(\boldsymbol{z}_\mathcal{S} - \boldsymbol{\mu}_\mathcal{S})$$
$$\Sigma' = \Sigma_{\mathcal{T}\mathcal{T}} - \Sigma_{\mathcal{T}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{T}}$$

Sequences $\boldsymbol{z}_\mathcal{T}|\boldsymbol{z}_\mathcal{S}$ more closely follow the data, and remain heterogeneous outside the data.



Tikhonov-Regularized Model vs. Gaussian Processes



- Tikhonov-regularized models require the user to provide a matrix Γ specifying how local variations are penalized.
- Gaussian processes require the user to provide a matrix Σ (a model of covariance), and a model of mean $\boldsymbol{\mu}$.
- Gaussian processes provide to the user not only a prediction for missing observations but also a model of predictive uncertainty (shown here as error bars, and where error bars show the standard deviation $\sqrt{\Sigma'_{ii}}$.

Part 2

Modeling Clustered Data

Recap: Motivations

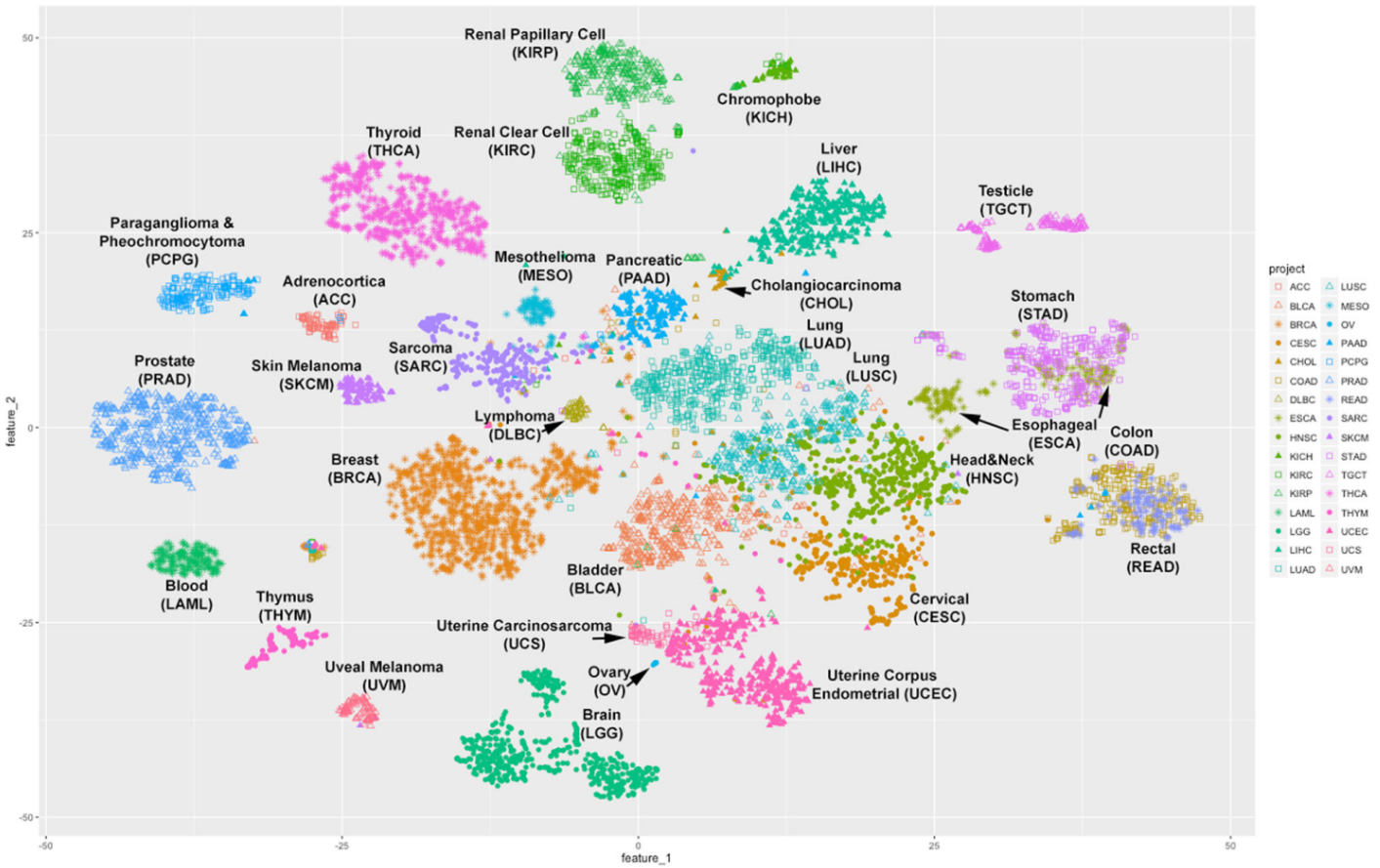


Image source: `https://doi.org/10.1038/s41467-021-21254-9`

- Methylation profiles of cancer cells organize into clusters, which often correlate with oncology categories.
- This cluster analysis is useful to verify that existing taxonomies are supported by the data.

Recap: K-means Clustering

Notation:

- Let $(\boldsymbol{x}_i)_{i=1}^N$ be our dataset of N instances.
- Let $(\boldsymbol{\mu}_k)_{k=1}^C$ be the C cluster centroids.
- Let $c : \{1, \ldots, N\} \to \{1, \ldots, K\}$ assign instances to clusters.

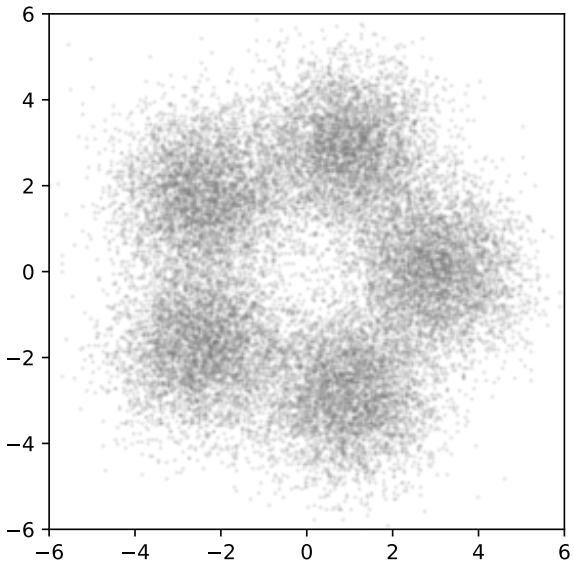
Objective:

- K-means clustering seeks to find cluster centroids and an assignment onto clusters that minimize the objective

$$J(\boldsymbol{\mu}, c) = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{\mu}_{c(i)}\|^2$$

- Each data point is assigned to one cluster as a result.

Limits of Clustering



Observations:

- Instances truly organize into distinct groups but these groups do not form well-separable clusters in $\mathbb{R}^d$ (here $\mathbb{R}^2$).
- If one would train a clustering algorithm, it would either find a single cluster, or produce a solution that do not have the desired separability property (cf. e.g. Dunn's index).

Idea:

- Adopt a probabilistic approach to account for group uncertainty.

Gaussian Mixture Model

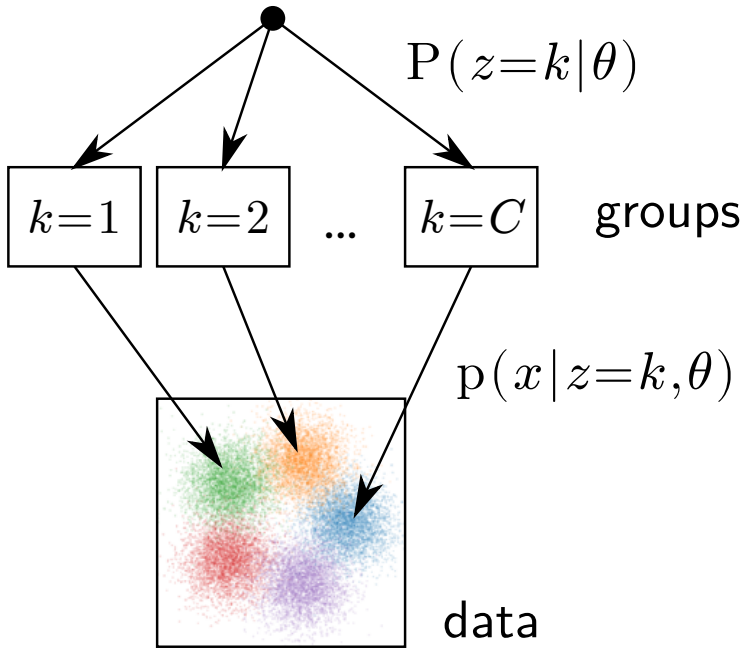
Data is generated via a two-step process:

step 1:

$$P(z \mid \theta) = \begin{cases} \alpha_1 & \text{if } z = 1 \\ \alpha_2 & \text{if } z = 2 \\ \vdots & \\ \alpha_C & \text{if } z = C \end{cases}$$

step 2:

$$p(\boldsymbol{x} \mid z = k, \theta) \sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$$



where $\theta = (\alpha_k, \boldsymbol{\mu}_k, \Sigma_k)_{k=1}^{C}$ are the parameters of the model.

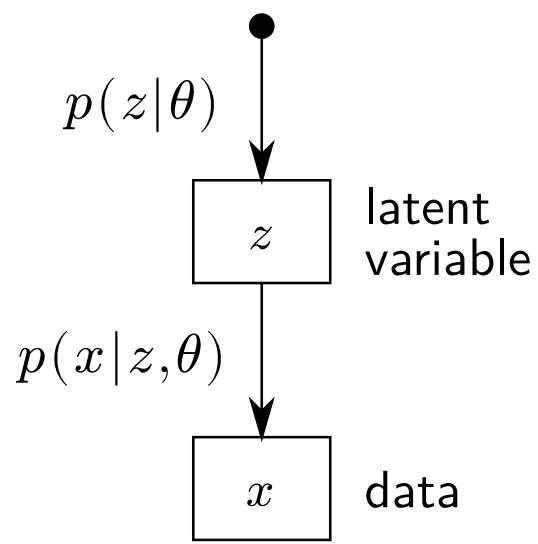
Gaussian Mixture Model

The marginal distribution $p(\boldsymbol{x}\,|\theta)$ of the Gaussian mixture model can be written using the law of total probabilities as:

$$p(\boldsymbol{x}\,|\,\theta) = \sum_{k=1}^{C} \underbrace{P(z=k\,|\,\theta)}_{\alpha_k} \cdot \underbrace{p(\boldsymbol{x}\,|\,z=k,\theta)}_{\sim\,\mathcal{N}(\boldsymbol{\mu}_k,\,\Sigma_k)}$$

More generally, the marginal distribution of a model with some discrete latent variable $\boldsymbol{z} \in \mathcal{Z}$ can be written as:

$$p(\boldsymbol{x}\,|\,\theta) = \sum_{z\in\mathcal{Z}} p(z\,|\,\theta) \cdot p(\boldsymbol{x}\,|\,z,\theta)$$



Learning a Mixture Model

Assuming a dataset $\mathcal{D}$, and considering examples in the dataset to be iid, the log-likelihood of the data according to the model can be written as:

$$\begin{aligned}\log P(\mathcal{D}|\theta) &= \log \prod_{\boldsymbol{x}\in\mathcal{D}} p(\boldsymbol{x}|\theta) \\ &= \sum_{\boldsymbol{x}\in\mathcal{D}} \log p(\boldsymbol{x}|\theta)\end{aligned}$$

and we wish to maximize that quantity. Injecting the latent variable model into the objective, we get:

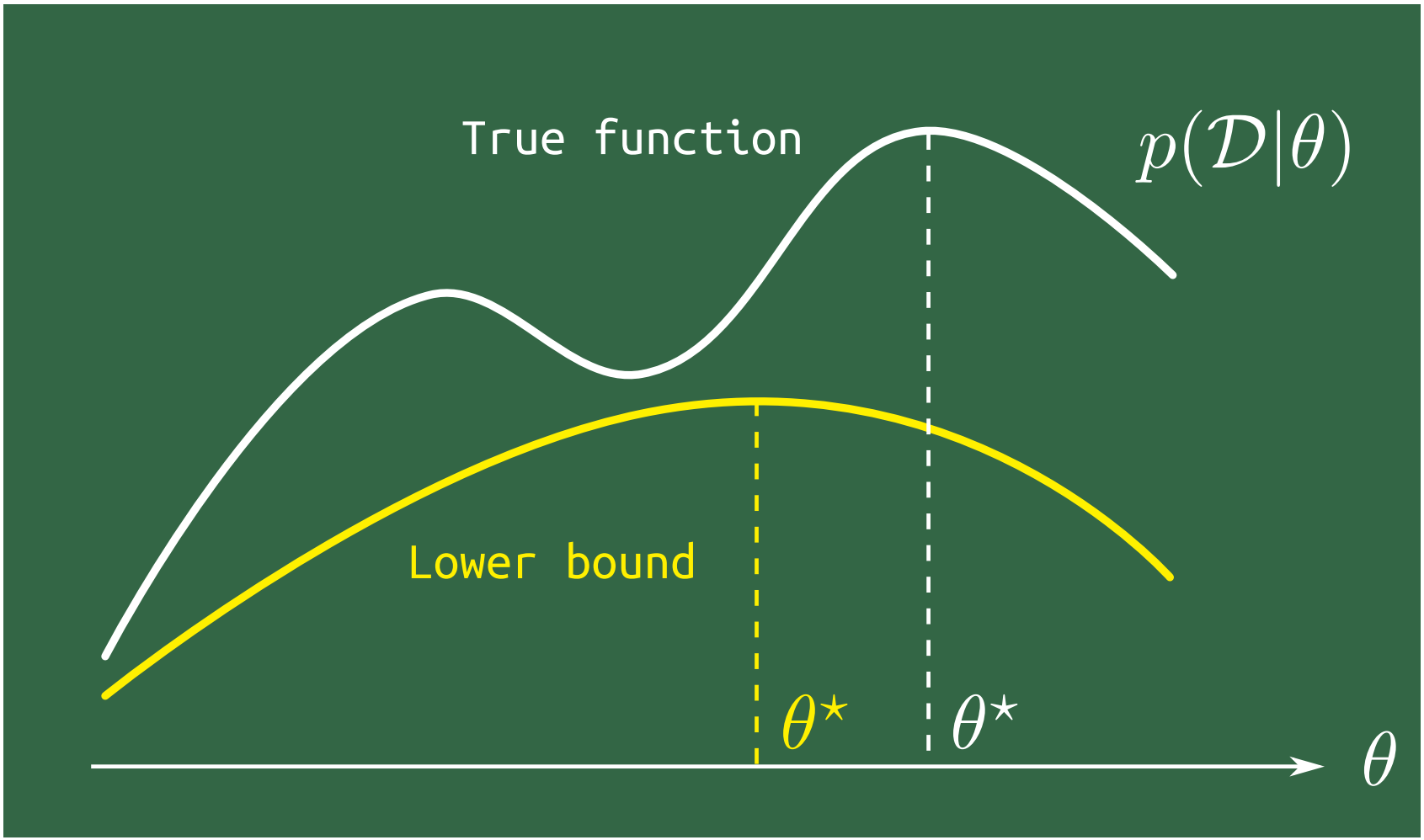
$$= \sum_{\boldsymbol{x}\in\mathcal{D}} \log \sum_{z\in\mathcal{Z}} p(z \mid \theta) \cdot p(\boldsymbol{x} \mid z, \theta)$$

Problem:

- The maximum of $\log p(\boldsymbol{x} \mid \theta)$ cannot be found analytically.

Learning a Latent Variable Model

Strategy: If the function $p(\mathcal{D}|\theta)$ cannot be easily optimized, find a lower-bound of that function that is easier to optimize.



Building a Lower-Bound

Jensen's inequality

For any element $\boldsymbol{\lambda}$ of the d-dimensional simplex (i.e. $\boldsymbol{\lambda} \succeq \mathbf{0}$ and $\boldsymbol{\lambda}^\top \mathbf{1} = 1$, and any concave function $f : \mathbb{R}^d \to \mathbb{R}$, we have

$$f(\sum_{i=1}^{d} \lambda_i a_i) \geq \sum_{i=1}^{d} \lambda_i f(a_i)$$

Application to the latent variable model:

- Let $\boldsymbol{\lambda}$ be some probability distribution $(q(z|\boldsymbol{x}))_{z \in \mathcal{Z}}$ independent from θ.
- Because our objective

$$\sum_{\boldsymbol{x} \in \mathcal{D}} \log \sum_{z \in \mathcal{Z}} p(z|\theta) \cdot p(\boldsymbol{x}|z, \theta)$$

does not contain such terms $q(z|\boldsymbol{x})$ independent from θ, create them!

Building a Lower-Bound

Step 1: Add $q(z|\boldsymbol{x})$ both on the numerator and denominator:

$$\log p(\mathcal{D}|\theta) = \sum_{\boldsymbol{x}\in\mathcal{D}} \log \sum_{z} \frac{p(z|\theta)\cdot p(\boldsymbol{x}|z,\theta)}{q(z|\boldsymbol{x})} \cdot q(z|\boldsymbol{x})$$

Step 2: Applying the Jensen inequality

$$\geq \sum_{\boldsymbol{x}\in\mathcal{D}} \sum_{z} \log\left(\frac{p(z|\theta)\cdot p(\boldsymbol{x}|z,\theta)}{q(z|\boldsymbol{x})}\right) \cdot q(z|\boldsymbol{x})$$

... and verify the bound:

$$= \sum_{\boldsymbol{x}\in\mathcal{D}} \sum_{z} \log\left(\frac{p(\boldsymbol{x}|\theta)\cdot p(z|\boldsymbol{x},\theta)}{q(z|\boldsymbol{x})}\right) \cdot q(z|\boldsymbol{x})$$

$$= \underbrace{\sum_{\boldsymbol{x}\in\mathcal{D}} \sum_{z} \log\Big(p(\boldsymbol{x}|\theta)\Big) \cdot q(z|\boldsymbol{x})}_{\log P(\mathcal{D}|\theta)} - \sum_{\boldsymbol{x}\in\mathcal{D}} \underbrace{\sum_{z} \log\left(\frac{q(z|\boldsymbol{x})}{p(z|\boldsymbol{x},\theta)}\right) \cdot q(z|\boldsymbol{x})}_{\mathrm{KL}(q(z|\boldsymbol{x}) \,\|\, p(z|\boldsymbol{x},\theta)) \geq 0}$$

Building a Lower-Bound

Question: How to ensure that

$$\mathrm{KL}(q(z|\boldsymbol{x}) \,\|\, p(z|\boldsymbol{x}, \theta))$$

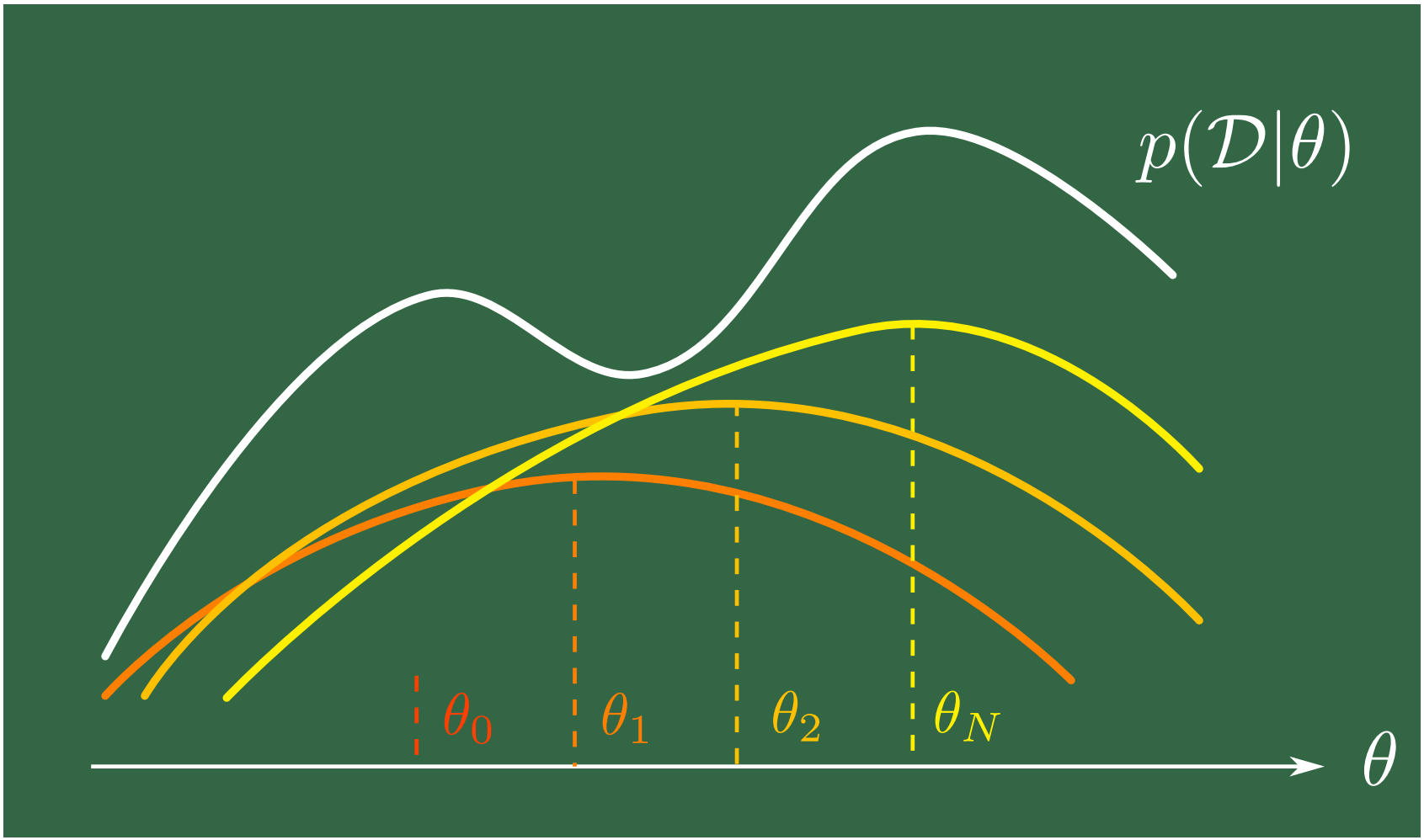
is small, so that maximizing the lower-bound with θ gives a good approximation of the true log-likelihood?

Chicken & egg problem:

- If we use a simple q, e.g. uniformly distributed, we get a loose lower-bound from which we get a bad θ.
- To get a tight lower-bound, we need to choose $q(z|\boldsymbol{x}) \approx p(z|\boldsymbol{x}, \theta)$ which requires that we know θ.

The Expectation-Maximization (EM) Algorithm

Strategy: Start with some random solution θ and alternately estimate q and θ, until we converge.



The Expectation-Maximization (EM) Algorithm

$$\theta_0 \leftarrow \text{random}()$$

$$q_1(z|\boldsymbol{x}) \leftarrow p(z|\boldsymbol{x}, \theta_0) \qquad \text{(E-step)}$$

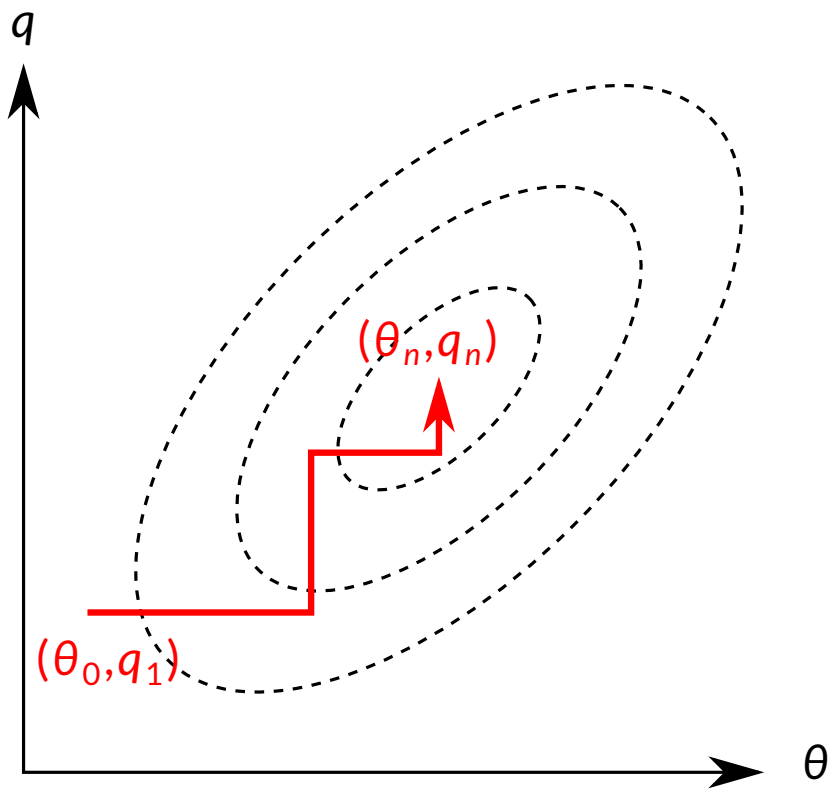
$$\theta_1 \leftarrow \arg\max_{\theta} \sum_{\boldsymbol{x} \in \mathcal{D}} \sum_{z} \log\left(\frac{p(\boldsymbol{x}, z|\theta)}{q_1(z|\boldsymbol{x})}\right) \cdot q_1(z|\boldsymbol{x}) \qquad \text{(M-step)}$$

$$q_2(z|\boldsymbol{x}) \leftarrow p(z|\boldsymbol{x}, \theta_1) \qquad \text{(E-step)}$$

$$\theta_2 \leftarrow \arg\max_{\theta} \sum_{\boldsymbol{x} \in \mathcal{D}} \sum_{z} \log\left(\frac{p(\boldsymbol{x}, z|\theta)}{q_2(z|\boldsymbol{x})}\right) \cdot q_2(z|\boldsymbol{x}) \qquad \text{(M-step)}$$

$$\vdots$$

The Expectation-Maximization (EM) Algorithm



Properties of the algorithm:

- Block coordinate descent
- Locally optimal step size
- The algorithm lands in a local minimum of the function $p(\mathcal{D}|\theta)$.

Advantages of EM compared to gradient descent:

- no learning rate
- no need to compute the gradients.

The Gaussian Mixture Model (GMM)

The GMM is formally defined by the two equations:

$$p(z = k \,|\, \theta) = \alpha_k$$

$$p(\boldsymbol{x} \,|\, z = k, \theta) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} \cdot \exp\Big(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1} (x - \boldsymbol{\mu}_k) \Big)$$

The **parameters** θ of the model to be learned are:

- The mixing coefficients $(\alpha_k)_{k=1}^C$ subject to $\alpha_k \geq 0$ and $\sum_{k=1}^C \alpha_k = 1$
- The mean vectors $(\boldsymbol{\mu}_k)_{k=1}^C$.
- The covariances $(\Sigma_k)_{k=1}^C$ subject to positive semi-definiteness.

Various **simplifications** of the GMM model can be used in practice, e.g. for speed, statistical robustness, or simplicity of implementation.

- diagonal/isotropic/tied/fixed covariance matrices,
- fixed mixing coefficients, etc.

EM for the GMM (simplified)

Consider the simplified GMM:

$$p(z = k|\theta) = \frac{1}{K}$$

$$p(\boldsymbol{x}|z = k, \theta) = \frac{1}{(\pi/\gamma)^{d/2}} \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{\mu}_k\|^2)$$

E-step: (Apply Bayes rule)

$$q(z = k|\boldsymbol{x}) = \frac{p(z = k|\theta) \cdot p(\boldsymbol{x}|z = k, \theta)}{p(\boldsymbol{x}|\theta)} = \frac{\exp(-\gamma \|\boldsymbol{x} - \boldsymbol{\mu}_k\|^2)}{\sum_{k=1}^{C} \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{\mu}_k\|^2)}$$

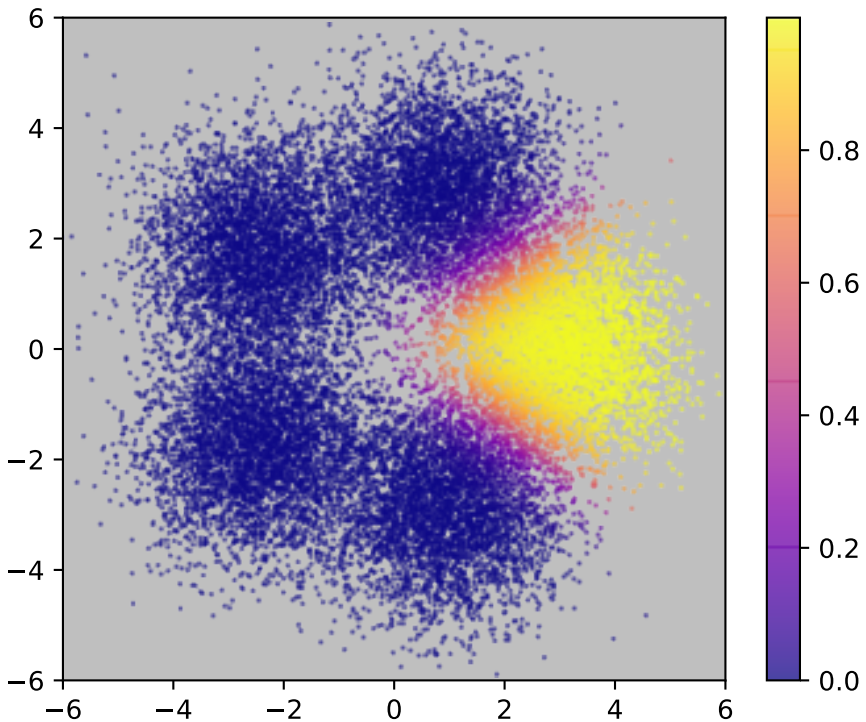
M-step: (Set lower-bound gradient to zero)

$$\frac{\partial}{\partial \theta} \sum_{\boldsymbol{x} \in \mathcal{D}} \sum_{k=1}^{C} \log(\exp(-\gamma \|\boldsymbol{x} - \boldsymbol{\mu}_k\|^2)) \cdot q(z = k|\boldsymbol{x}) = 0$$

$$\Rightarrow \boldsymbol{\mu}_k = \frac{\sum_{\boldsymbol{x} \in \mathcal{D}} \boldsymbol{x} \cdot q(z = k|\boldsymbol{x})}{\sum_{\boldsymbol{x} \in \mathcal{D}} q(z = k|\boldsymbol{x})}$$

Gaussian Mixture Model on Toy Data

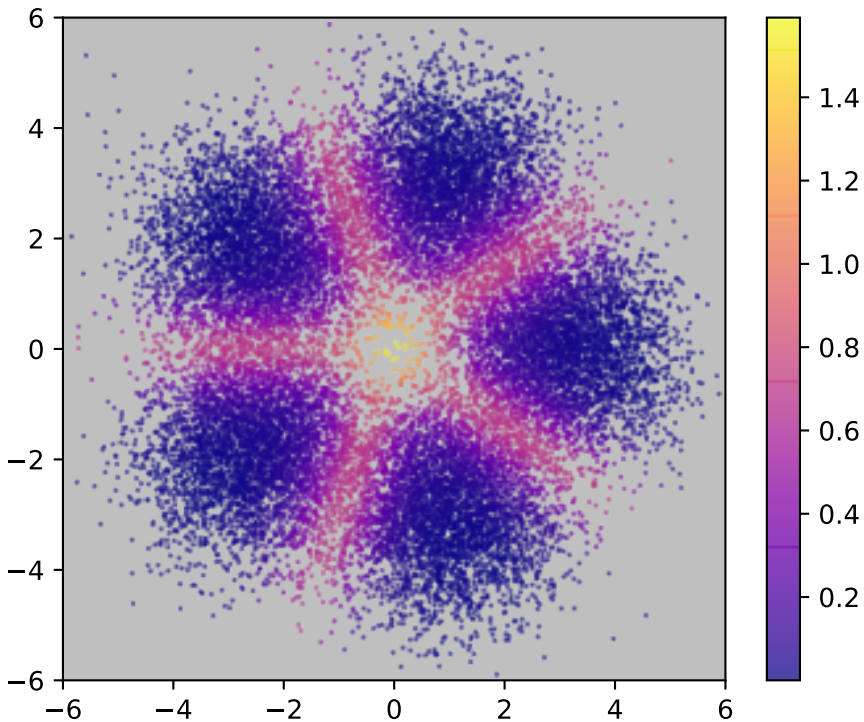
Probability of membership to a given group k, i.e. $q(z = k|\boldsymbol{x})$, as predicted by a GMM with 5 mixture components:



- Unlike K-means, the GMM does not exhibit an abrupt and unnatural transition between predicted group members and non-group members.

Gaussian Mixture Model on Toy Data

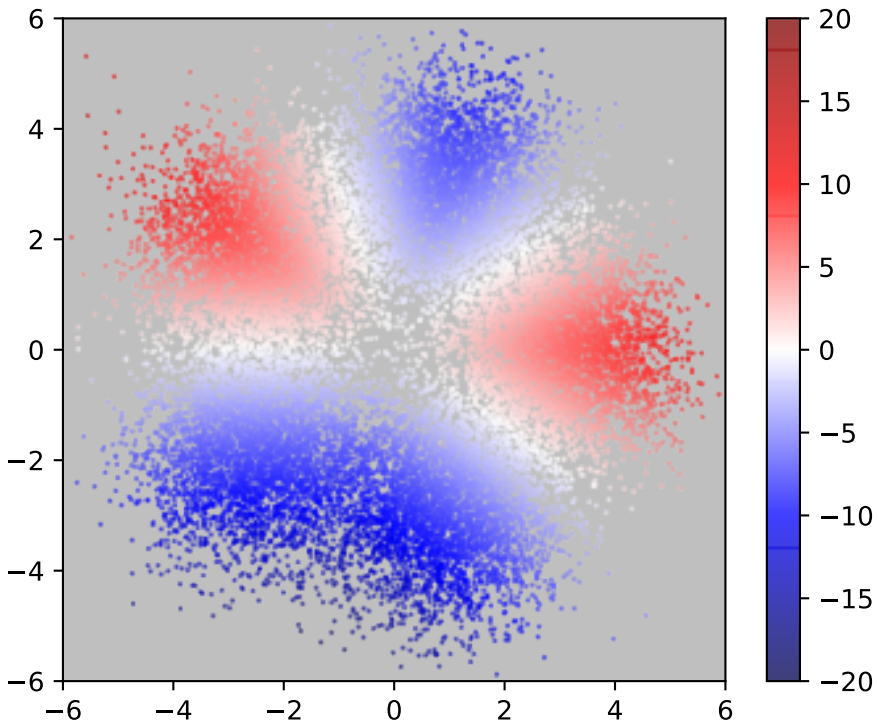
Overall uncertainty measured by the entropy of the vector of probabilities of group memberships, i.e. $-\sum_k q(z=k|\boldsymbol{x}) \log(q(z=k|\boldsymbol{x}))$:



- This analysis can be useful to identify prototypical points for each group, or conversely, unresolved instances (that can be of scientific interest).

Gaussian Mixture Model on Toy Data

Log-probability ratio between two sets of groups $\mathcal{K}, \mathcal{K}'$ ($\log \sum_{k \in \mathcal{K}} q(z = k|\boldsymbol{x}) - \log \sum_{k \in \mathcal{K}'} q(z = k|\boldsymbol{x})$) as predicted by a GMM with 5 mixture components:



- Interesting nonlinear discriminants functions can be generated from the GMM model.

Summary

Summary

- There exists a wide variety of probability models for various tasks (regression, discriminant analysis, interpolation, recovering latent structure, etc.)
- Unlike their 'classical' counterparts, probabilistic models often provide additional functionality (e.g. predictive uncertainty, nonlinearity). This comes at the cost of some extra complexity.
- Some probabilistic models admit closed form solutions whereas other models (such as the Gaussian Mixture Model) require other forms of optimization (e.g. expectation-maximization).