Density Estimation and Expectation Maximization

Maschinelles Lernen 1 - Grundverfahren WS20/21

Prof. Gerhard Neumann

KIT, Institut für Anthrophomatik und Robotik

Learning Outcomes

What will we learn today?

- Understand the density estimation problem
- Recap parametric density estimation and its Bayesian variant
- Understand non-parametric density estimation and its advantages
- Understand mixture models and how to train it using EM
- Analysis of the EM algorithm and why it converges

Agenda for today

Parametric Density Estimation

- Maximum Likelihood
- Bayesian Estimation (see Bayesian Learning lecture...)

Non-Parametric Density Estimation

- Histograms
- Kernel-density estimates
- Nearest neighbour density estimates

Mixture Models

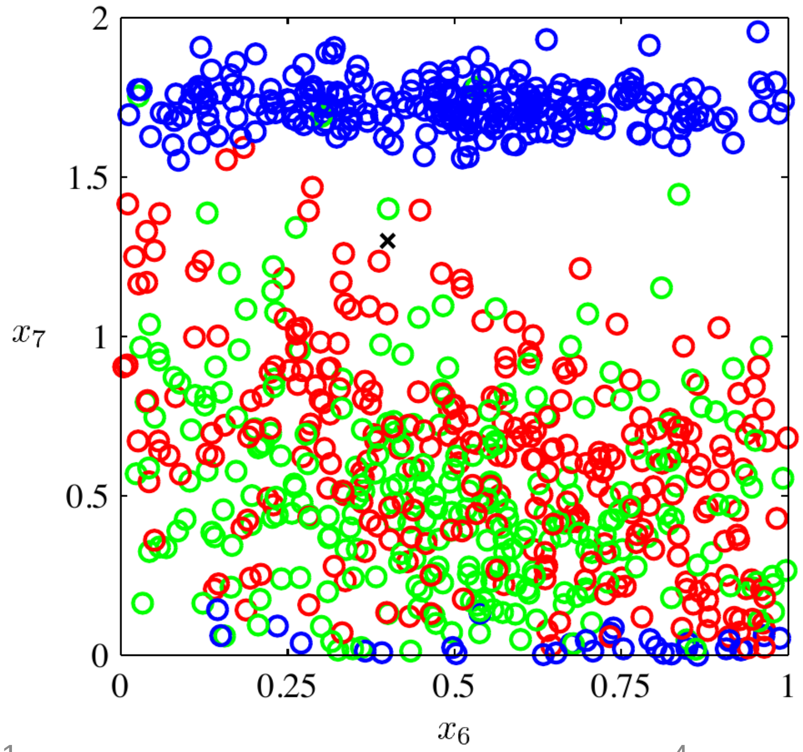
- Gaussian Mixture Models (GMM)

The Expectation Maximization Algorithm

- EM decomposition
- E- and M-step
- Convergence analysis
- EM for GMMs

Density estimation

How do we get the probability distributions from this?



Applications:

- Classify (generative approaches)
- Outlier / unseen event detection
- Generate new data

Probability Density Estimation

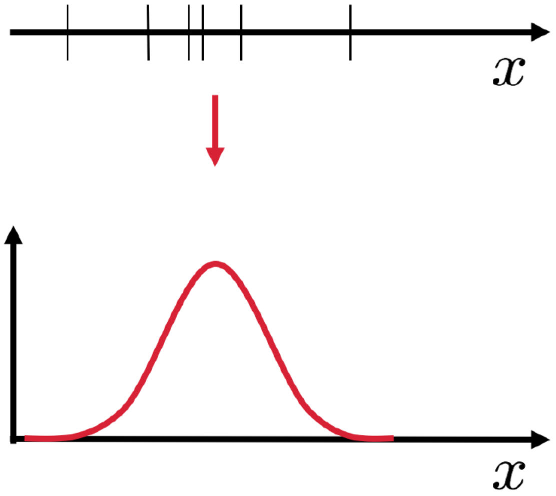
- **Training data**

$$\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$$

- **Estimation**

$$p(\boldsymbol{x})$$

- **Methods**
  - Parametric model
  - Non-parametric model
  - Mixture models



Recap: Parametric models

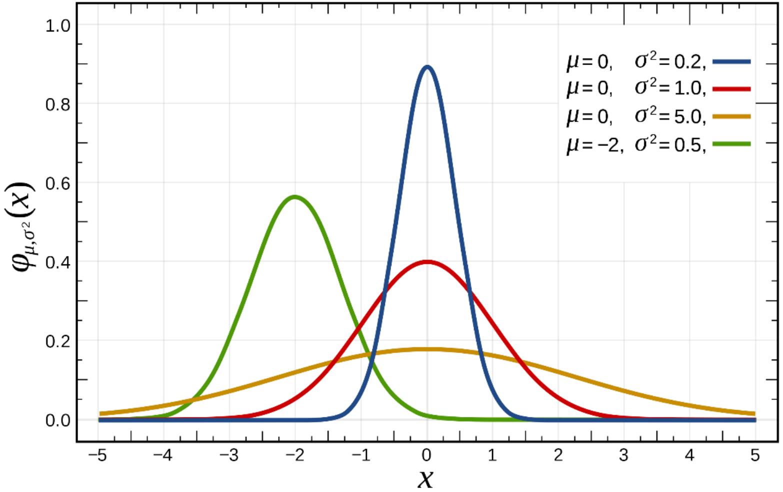
Most commonly used: Gaussian distribution

Parametric model:

$$p_{\boldsymbol{\theta}}(x) = \mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

2 Parameters: $\boldsymbol{\theta} = \{\mu, \sigma\}$

- Mean μ
- Variance σ^2



Recap: Maximum Likelihood Estimation (MLE)

Maximize the Log-likelihood:

$$\mathrm{loglik}(\boldsymbol{\theta}; D) = \sum_i \log p_{\boldsymbol{\theta}}(x_i)$$

- **Assumption:** Independently identically distributed (iid.) dataset
- Sums are “nicer” to optimize than products
- Log cancels exponential form (most distributions are in the exponential family)

The MLE solution is given by:

$$\boldsymbol{\theta}_{\mathrm{ML}} = \mathrm{argmax}_{\boldsymbol{\theta}} \mathrm{loglik}(\boldsymbol{\theta}; D)$$

Maximum Likelihood for a Gaussian

- ML estimation for a Gaussian

$$\boldsymbol{\mu}, \boldsymbol{\Sigma} = \mathrm{argmax}_{\boldsymbol{\theta}} \mathrm{loglik}(\boldsymbol{\theta}; D) = \sum_{i=1}^{N} \log \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- Take the partial derivatives and set it to 0

$$\frac{\partial \mathrm{loglike}(\boldsymbol{\theta}; \mathcal{D})}{\partial \boldsymbol{\mu}} = \mathbf{0} \qquad \frac{\partial \mathrm{loglike}(\boldsymbol{\theta}; \mathcal{D})}{\partial \boldsymbol{\Sigma}} = \mathbf{0}$$

- Which leads to the closed form solution (also see homework 1)

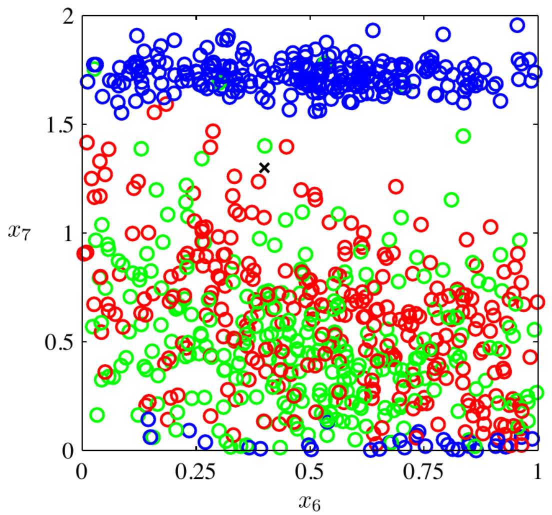
$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i \qquad \boldsymbol{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T$$

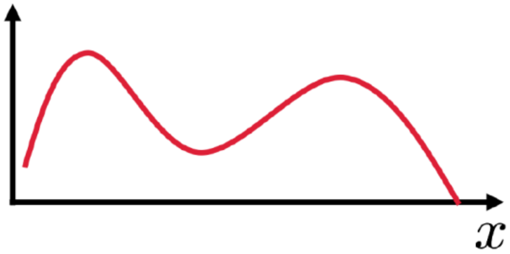
Non-parametric Models

- **Does this look Gaussian?**

- **No!** Indeed most data-sets are not Gaussian distributed. Typically we have:
  - Multi-Modality
  - Non-symmetric
  - No infinite support

- **We need more complex representations:**
  - Non-parametric
  - Mixture models





Non-parametric models

Why use Non-parametric representations?

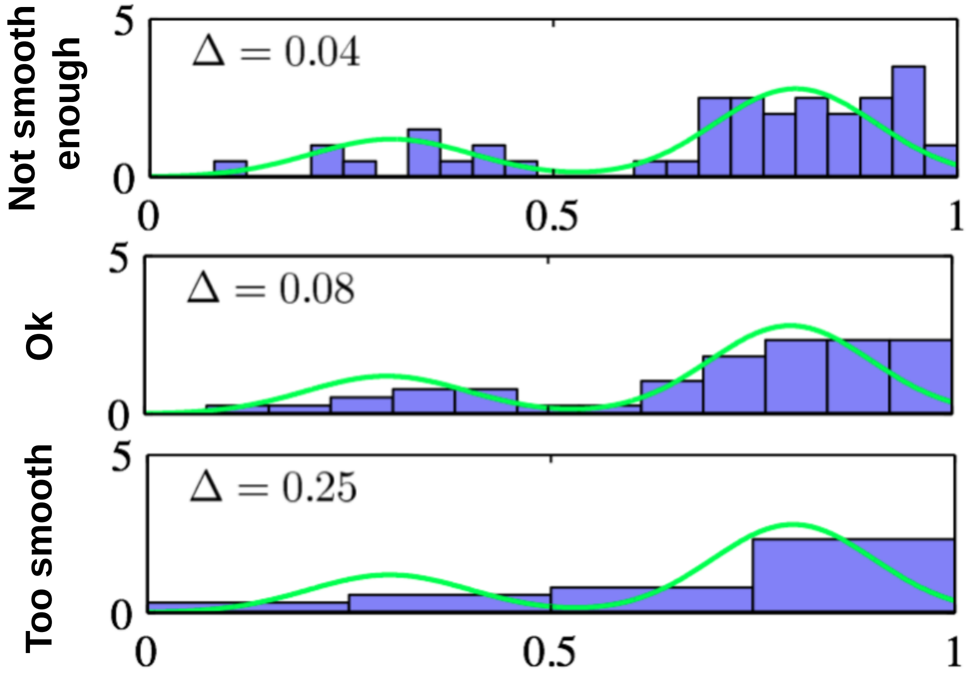
- Often we do not know what functional form the class-conditional density takes (or we do not know what class of function we need)

Probability density is estimated directly from the data (i.e. without an explicit parametric model)

- Histograms
- Kernel density estimation (Parzen windows)
- K-nearest neighbors

Histograms

- Discretize the input space into bins
- Count the samples per bin



Histograms

Properties

- They are very general, because in the infinite data limit any probability density can be approximated arbitrarily well
- At the same time it is a Brute-force method

Problems

- High-dimensional feature spaces
- Exponential increase in the number of bins
- Hence requires exponentially much data
- Commonly known as the curse of dimensionality
- How to choose the size of the bins?
  - This is again a model-selection problem!

More formal definition

- Data point **x** is sampled from probability density p(**x**)
- Probability that **x** falls in region *R*

$$p(\boldsymbol{x} \in R) = \int_R p(\boldsymbol{x}) d\boldsymbol{x}$$

- If R is is sufficiently small, with volume V , then p(**x**) is almost constant

$$p(\boldsymbol{x} \in R) = \int_R p(\boldsymbol{x}) d\boldsymbol{x} \approx p(\boldsymbol{x}) V$$

- We can also compute $p(\boldsymbol{x} \in R)$ from samples (If we have sufficiently large dataset)

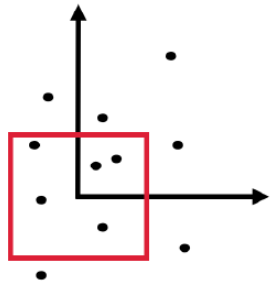
$$p(\boldsymbol{x} \in R) \approx \frac{K}{N} \Rightarrow p(\boldsymbol{x}) \approx \frac{K}{NV}$$

where N is the number of total points and K is the number of points falling in the region R

Regions

$$p(\boldsymbol{x}) \approx \frac{K}{NV}$$

- For histograms, the regions are of equal size and span across the whole input space.
- Can we find a more adaptive representation of regions?
  - Yes, make the region always centred on the input **x**!

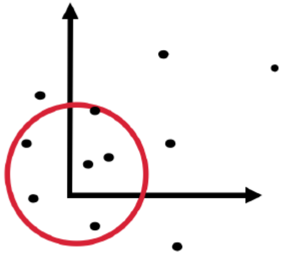


Kernel density estimation

- Fix V and determine K
- Example: determine the number of data points K in a fixed hypercube

K-nearest neighbor

- Fix K and determine V
- Example: increase the size of a sphere until K data points fall into the sphere



Kernel density estimation

A kernel $k(\boldsymbol{x}, \boldsymbol{y})$ **"compares" two samples x and y**

- Required properties for density estimation:
  - Non-negative: $k(\boldsymbol{x}, \boldsymbol{y}) \geq 0$
  - Distance-dependent: $k(\boldsymbol{x}, \boldsymbol{y}) = g(\underbrace{\boldsymbol{x} - \boldsymbol{y}}_{\text{difference } \boldsymbol{u}})$
- **Volume:** $V = \int g(\boldsymbol{u}) d\boldsymbol{u}$
- **Summed kernel activation:** $K(\boldsymbol{x}_*) = \sum_{i=1}^{N} g(\boldsymbol{x}_* - \boldsymbol{x}_i)$
- **Estimated density:** $p(\boldsymbol{x}_*) \approx \frac{K(\boldsymbol{x}_*)}{NV} = \frac{1}{NV} \sum_{i=1}^{N} g(\boldsymbol{x}_* - \boldsymbol{x}_i)$
- A more formal definition can be found in the kernel lecture

Parzen Window

- **Kernel function:** Hypercubes in d dimensions with edge length h

$$g(\boldsymbol{u}) = \begin{cases} 1, & |u_j| \leq h/2, \; j = 1 \ldots d \\ 0, & \text{else} \end{cases}$$

bandwidth

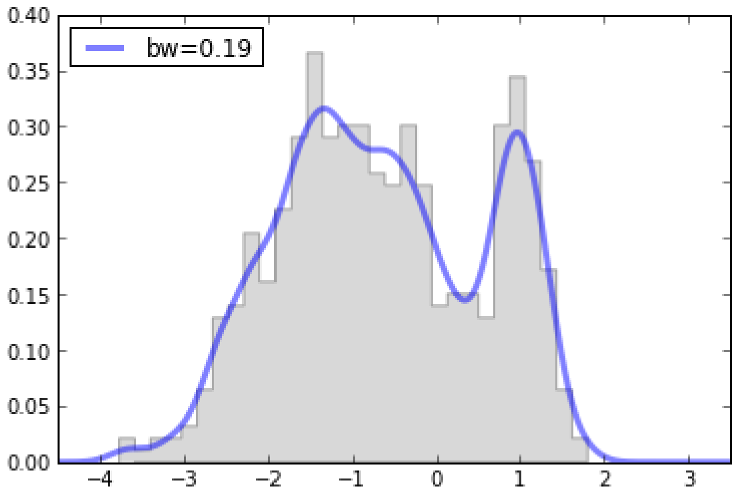
dimensionality

- **Volume:**

$$V = \int g(\boldsymbol{u}) d\boldsymbol{u} = h^d$$

- **Estimated Density:**

$$p(\boldsymbol{x}_*) \approx \frac{1}{N h^d} \sum_{i=1}^{N} g(\boldsymbol{x}_* - \boldsymbol{x}_i)$$



✓ Simple to compute

× Not very smooth

Gaussian kernel…

- **Kernel function:**

$$g(\boldsymbol{u}) = \exp\left(-\frac{\|\boldsymbol{u}\|^2}{2h}\right)$$

bandwidth

- **Volume:**

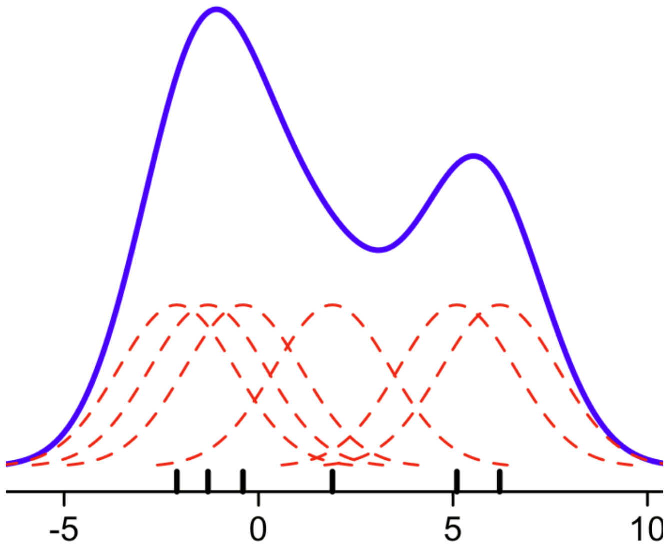
$$V = \int g(\boldsymbol{u}) d\boldsymbol{u} = \sqrt{2\pi h^d}$$

dimensionality

- **Estimated Density:**

$$p(\boldsymbol{x}_*) \approx \frac{1}{NV} \sum_{i=1}^{N} g(\boldsymbol{x}_* - \boldsymbol{x}_i)$$

$$= \frac{1}{N\sqrt{2\pi h^d}} \sum_{i=1}^{N} \exp\left(-\frac{\|\boldsymbol{x}_* - \boldsymbol{x}_i\|^2}{2h}\right)$$



- ✓ Smooth
- × Infinite support
- × Requires a lot of computation

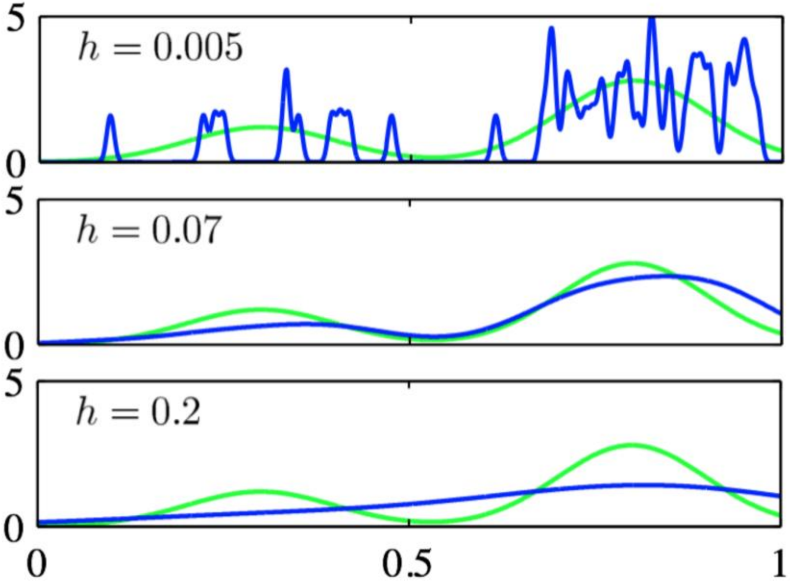
Gaussian KDE Example

- **Problem with kernel methods:** We have to select the kernel bandwidth h appropriately

Not smooth enough

About right

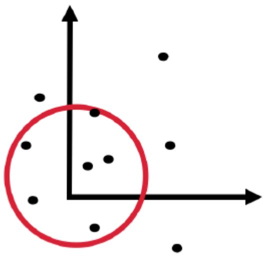
Too smooth



K-nearest neighbour density estimation

K-nearest neighbour: Fix K and determine V

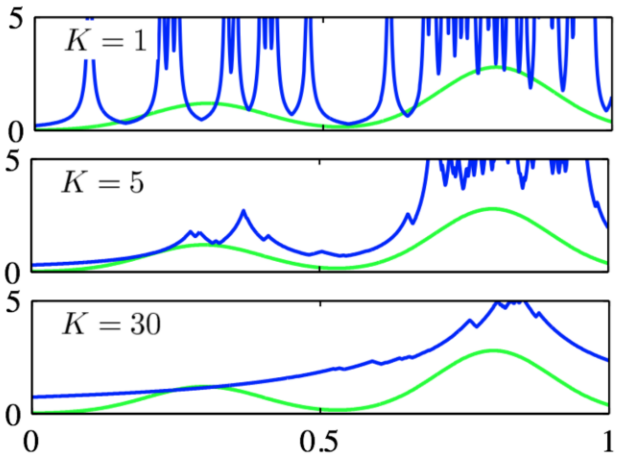
- Example: increase the size of a sphere until K data points fall into the sphere



Not smooth enough

About right

Too smooth



Model-Selection

Nonparametric probability density estimation

- **Histograms:** Size of the bins?
  - too large: too smooth
  - too small: not smooth enough
- **Kernel density estimation:** Kernel bandwidth?
  - h too large: too smooth
  - h too small: not smooth enough
- **K-nearest neighbor:** Number of neighbors?
  - K too large: too smooth
  - K too small: not smooth enough

A general problem of many density estimation approaches

- Select via cross-validation: Select model with highest likelihood on test-set

Agenda for today

Parametric Density Estimation
- Maximum Likelihood
- Bayesian Estimation

Non-Parametric Density Estimation
- Histograms
- Kernel-density estimates
- Nearest neighbour density estimates

Mixture Models
- Gaussian Mixture Models (GMM)

The Expectation Maximization Algorithm
- EM decomposition
- E- and M-step
- Convergence analysis
- EM for GMMs

Mixture Models

Mixture Models

Parametric models

- Gaussian, Neural Networks, ...
- ✓ Good analytic properties
- ✓ Simple
- ✓ Small memory requirements
- ✓ Fast
- × Limited representation power (most parametric distributions have only one mode)

Non-Parametric models

- Kernel-density estimation, k-NN
- ✓ General (can represent any distribution)
- × Curse of dimensionality
- × Large memory requirements
- × Slow

- Mixture models combine the advantages of both worlds
- **Key idea:** Create a complex distribution by combining simple ones (e.g. Gaussians)

Mixture model

A mixture distribution is the sum of individual distributions:

Number of components

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} p(k) p(\boldsymbol{x}|k)$$

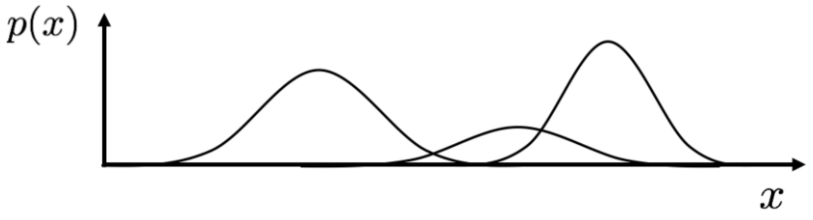
Mixture coefficient

k-th mixture component

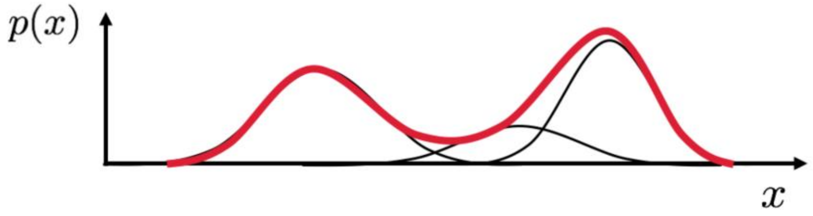
- In the limit with many / infinite components, this can approximate any smooth density

Example: Mixture of Gaussians (MoG)

- Individual Gaussians



- Sum of Gaussians



Mixture of Gaussians

- **Mixture coefficient:**

$$p(k) = \pi_k, \text{ with } 0 \leq \pi_k \leq 1, \; \sum_k \pi_k = 1$$

- **Mixture component:**

$$p(\boldsymbol{x}|k) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- **Mixture distribution:**

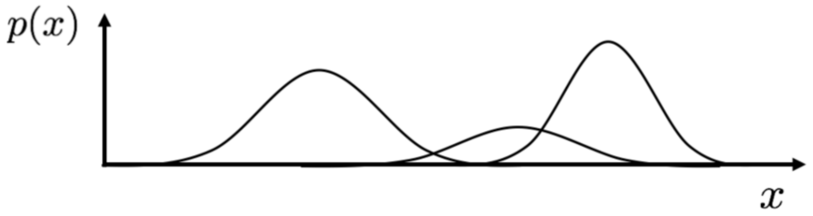
$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

  - Always integrates to 1
  - Parameters of the mixture

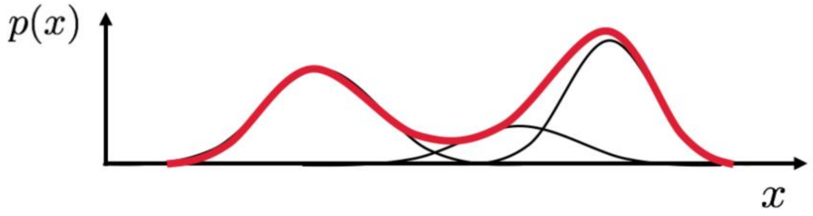
$$\boldsymbol{\theta} = \{\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \dots, \pi_K, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K\}$$

Example: Mixture of Gaussians (MoG)

- Individual Gaussians



- Sum of Gaussians



Maximum Likelihood of a mixture

- (Marginal-)Log-Likelihood with N iid. points

$$\mathcal{L} = \log L(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log \underbrace{p(\boldsymbol{x}_i|\boldsymbol{\theta})}_{\text{marginal}} = \sum_{i=1}^{N} \underbrace{\log\left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right)}_{\text{non-exponential family}}$$

- Q: Can we do gradient descent?

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_j} = \sum_{i=1}^{N} \frac{\pi_j \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)$$

$$= \sum_{i=1}^{N} \frac{p(j)p(\boldsymbol{x}_i|j)}{p(\boldsymbol{x}_i)} \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)$$

$$= \sum_{i=1}^{N} \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_j) p(j|\boldsymbol{x}_i)$$

- × Gradient depends on all other components (cyclic dependency)
- × No closed form solution
- × Typically very slow convergence
- ► A: Yes, but the sum (marginalization) does not go well with the log

Agenda for today

Parametric Density Estimation
- Maximum Likelihood
- Bayesian Estimation

Non-Parametric Density Estimation
- Histograms
- Kernel-density estimates
- Nearest neighbour density estimates

Mixture Models
- Gaussian Mixture Models (GMM)

The Expectation Maximization (EM) Algorithm
- EM decomposition
- E- and M-step
- Convergence analysis
- EM for GMMs

Basics: Kullback-Leibler Divergences

The KL-divergence is a important similarity measure for distributions

$$\mathrm{KL}(q(\boldsymbol{x})||p(\boldsymbol{x})) = \sum_{\boldsymbol{x}} q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p(\boldsymbol{x})}$$

- **Its always non-negative** $\mathrm{KL}(q||p) \geq 0$
- **If its zero, both distributions are the same:** $\mathrm{KL}(q||p) = 0 \iff q = p$
- **It is non-symmetric** (hence, its not a distance metric): $\mathrm{KL}(q||p) \neq \mathrm{KL}(p||q)$

- Can be used to find different approximations of distributions
- Used a lot in Variational Inference, Reinforcement Learning, Information theory…

Latent Variable Models

Mixture models are an instance of latent variable models

- Examples: Missing data, latent factors, mixtures, …
- Observed variables: **x**, Latent variables: **z** (e.g., index of mixture component)
- Parametric model: $p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})$
- Marginal distribution: $\underbrace{p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_z p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}),}_{\text{discrete latent variable}} \quad \underbrace{p(\boldsymbol{x}|\boldsymbol{\theta}) = \int_z p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) d\boldsymbol{z}}_{\text{continuous latent variable}}$

(Marginal) Log-Likelihood:

$$\mathcal{L} = \log L(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \sum_{i=1}^{N} \log \left(\sum_z p(\boldsymbol{x}_i, \boldsymbol{z}) \right)$$

… which is hard to optimize for all latent variable models (due to log of a sum)

Expectation-Maximization (EM)

Expectation-Maximization (EM) is a general algorithm for estimating latent variable models

- **Most common application:** Gaussian Mixture models
- … but many other (deep) models as well
- Its extension is called Variational Bayes, which is underlying variational auto-encoder and other variational inference techniques
- Very hot research topic… pays off to look into the math of it

EM can be derived in 2 ways:

- Jensen's inequality (not covered)
- Decomposition in lower-bound and KL-term

Expectation-Maximization (EM)

EM uses a lower bound of the marginal log-likelihood for the optimization

- For simplicity, lets consider only a single data-point first

$$\underbrace{\log p(\boldsymbol{x}|\boldsymbol{\theta})}_{\text{marginal log-like}} = \underbrace{\sum_z q(z) \log \frac{p(\boldsymbol{x}, z|\boldsymbol{\theta})}{q(z)}}_{\text{Lower Bound } \mathcal{L}(q,\boldsymbol{\theta})} + \underbrace{\sum_z q(z) \log \frac{q(z)}{p(z|\boldsymbol{x})}}_{\text{KL Divergence: } \mathrm{KL}(q(z)||p(z|\boldsymbol{x}))}$$

- Where $q(z)$ is called the variational / auxiliary distribution
  - This decomposition holds for any $q(z)$
  - By introducing q(z), the optimization will become much simpler
- Why is that the same?
  - We can use Bayes rule for $p(z|\boldsymbol{x}) = \dfrac{p(\boldsymbol{x}, z|\boldsymbol{\theta})}{p(\boldsymbol{x}|\boldsymbol{\theta})}$ and all terms except $p(\boldsymbol{x}|\boldsymbol{\theta})$ cancel

EM-Decomposition

Derivation:

$$\log p(\boldsymbol{x}) = \sum_z q(z) \log p(\boldsymbol{x})$$

$$= \sum_z q(\boldsymbol{z}) \big(\log p(\boldsymbol{x}, z) - \log p(\boldsymbol{z}|\boldsymbol{x}) \big)$$

$$= \sum_z q(z) \big(\log p(\boldsymbol{x}, z) - \log q(z)$$

$$+ \log q(z) - \log p(\boldsymbol{z}|x) \big)$$

$$= \underbrace{\sum_z q(z) \log \frac{p(\boldsymbol{x}, z)}{q(z)}}_{\text{Lower Bound } \mathcal{L}(q)} + \underbrace{\sum_z q(z) \log \frac{q(z)}{p(z|\boldsymbol{x})}}_{\text{KL Divergence: } \mathrm{KL}(q(z)||p(z|\boldsymbol{x}))}$$

1. Introduce variational distribution q(z)
2. Use Bayesian theorem
   $$p(\boldsymbol{x}) = \frac{p(\boldsymbol{x}, z)}{p(\boldsymbol{z}|x)}$$
3. Add and substract $\log q(z)$
4. Write as 2 sums

EM Decomposition

Marginal Likelihood decomposes in 2 terms: $\log p(\boldsymbol{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + \mathrm{KL}(q(z)||p(z|\boldsymbol{x}))$

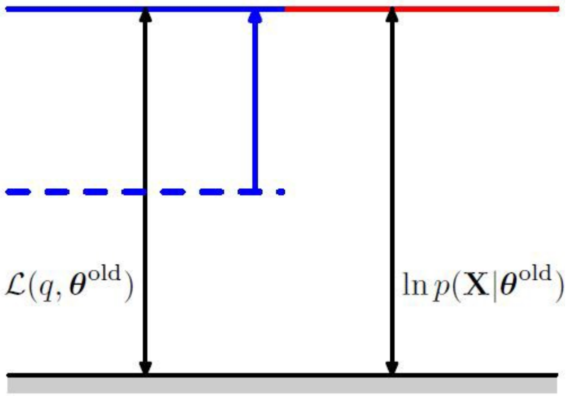
- **Lower bound** $\mathcal{L}(q, \boldsymbol{\theta}) = \sum_z q(z) \log p(\boldsymbol{x}, z|\boldsymbol{\theta}) - \sum_z q(z) \log q(z)$
  - Contains $\log p(\boldsymbol{x}, z|\boldsymbol{\theta})$ instead of $\log p(\boldsymbol{x}|\boldsymbol{\theta}) = \log \sum_z p(\boldsymbol{x}, z|\boldsymbol{\theta})$
  - … which is much easier to optimize (convex for most distributions)
  - Each $\log p(\boldsymbol{x}, z|\boldsymbol{\theta})$ is weighted by *q(z)*
- Why is it a lower bound?
  - Since $\mathrm{KL}(q||p) \geq 0$ it follow that $\mathcal{L}(q, \boldsymbol{\theta}) \leq \log p(\boldsymbol{x}|\boldsymbol{\theta})$

Expectation-Maximization Steps

EM iteratively applies 2 steps:

- **(E)xpectation-step:** $q(z) = \arg\min_q \mathrm{KL}(q(z)||p(z|\boldsymbol{x}))$
  - Find q(z) that minimizes KL
  - Can be done in closed form for discrete z:

$$q(z) = p(z|\boldsymbol{x}, \boldsymbol{\theta}_{\text{old}}) = \frac{p(\boldsymbol{x}, z|\boldsymbol{\theta}_{\text{old}})}{\sum_z p(\boldsymbol{x}, z|\boldsymbol{\theta}_{\text{old}})}$$



- **Observations:**
  - The marginal log-likelihood $\log p(\boldsymbol{x}|\boldsymbol{\theta})$ is unaffected by the E-step
  - As KL is minimized, lower bound has to go up
  - After the E-step $\mathrm{KL}(q(z)||p(z|\boldsymbol{x})) = 0$ and therefore, the lower bound is tight, i.e.:

$$\log p(\boldsymbol{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta})$$

Expectation-Maximization Steps

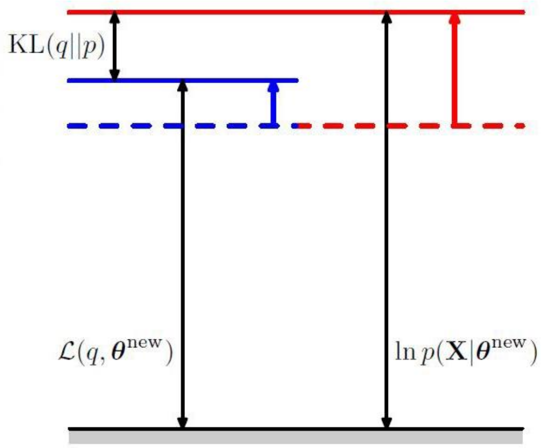
EM iteratively applies 2 steps:

- **(M)aximization-step:**

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(q, \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_z q(z) \log p(\boldsymbol{x}, z|\boldsymbol{\theta}) + \text{const}$$

  - Maximize lower bound with respect to $\boldsymbol{\theta}$
  - Also called the complete-data likelihood
  - Each possible value of the missing data is weighted by

$$q(z) = p(z|\boldsymbol{x}, \boldsymbol{\theta}_{\text{old}})$$



EM Convergence Properties

- **EM improves the lower bound**

$$\mathcal{L}(q_{\text{new}}, \boldsymbol{\theta}_{\text{new}}) \geq \mathcal{L}(q_{\text{old}}, \boldsymbol{\theta}_{\text{old}})$$

  - M-step: Lower bound is maximized
  - E-step: KL is set to 0, lower bound has to go up

- **EM improves the marginal likelihood**

$$\log p(\boldsymbol{x}|\boldsymbol{\theta}_{\text{new}}) \geq \log p(\boldsymbol{x}|\boldsymbol{\theta}_{\text{old}})$$

  - M-step: Lower bound increases and KL increases (can't get smaller than 0)
  - E-step: Marginal likelihood is unaffected

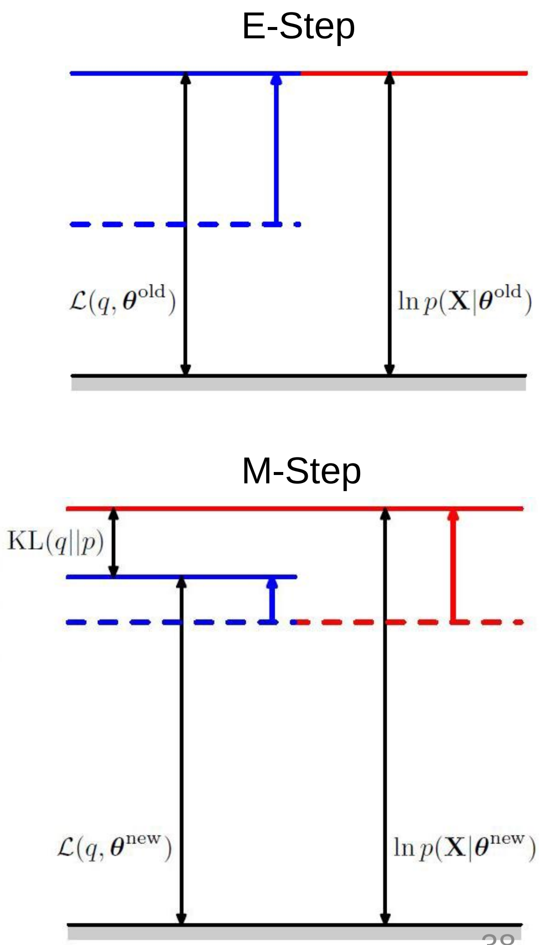
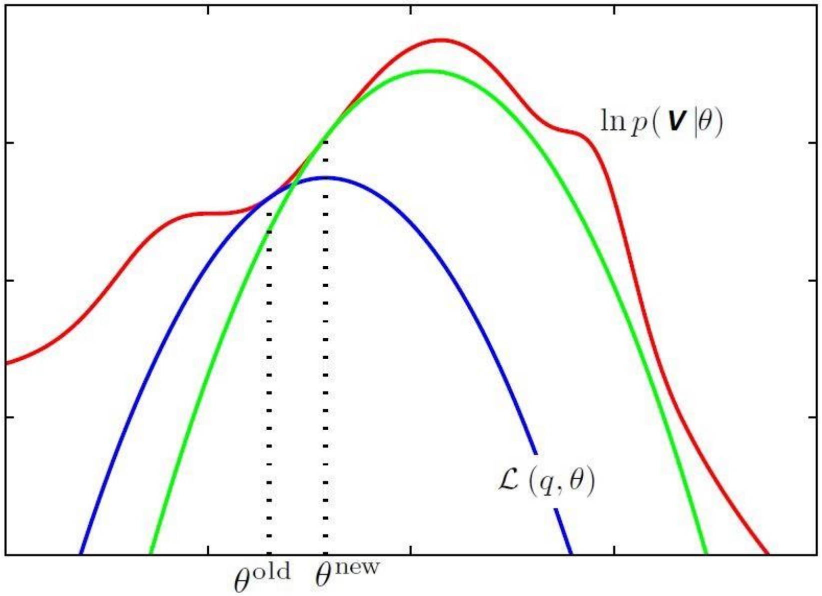


Illustration of EM

- Lower bound (blue curve) is a convex approximation of the marginal likelihood (red curve)
  - Maximum of lower bound can be easily obtained ($\boldsymbol{\theta}^{\text{new}}$)
  - Closed form solutions available, no gradient descent required
- Compute new lower bound for $\boldsymbol{\theta}^{\text{new}}$ (green curve)
- Due to the local approximation of the lower-bound, EM can only find local optima



EM for full dataset

For all data-points, the lower bound is given by:

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_i \sum_z q_i(z) \log p(\boldsymbol{x}_i, z|\boldsymbol{\theta}) - \sum_z q_i(z) \log q_i(z)$$

- One latent variable z_i per data-point
- If z is discrete with K different values, than

$$q_i(z = k) = p(z = k|\boldsymbol{x}_i, \boldsymbol{\theta}_{\text{old}})$$

  can be represented as a N x K matrix
- We will write $q_{ik} = q_i(z = k)$

EM for Mixture of Gaussians

- **Mixture coefficient:**

$$p(k) = \pi_k, \text{ with } 0 \leq \pi_k \leq 1, \sum_k \pi_k = 1$$

- **Mixture component:**

$$p(\boldsymbol{x}|k) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- **Mixture distribution:**

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

  - Always integrates to 1
  - Parameters of the mixture

$$\boldsymbol{\theta} = \{\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_K, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K\}$$

E-Step

- Compute “responsibilities”

$$q_{ik} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

- How much component k contributes to generation of $\mathbf{x}_i$ according to current mixture model

EM for Mixture of Gaussians

- **Mixture coefficient:**

$$p(k) = \lambda_k, \text{ with } 0 \leq \lambda_k \leq 1, \sum_k \lambda_k = 1$$

- **Mixture component:**

$$p(\boldsymbol{x}|k) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- **Mixture distribution:**

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

  - Always integrates to 1
  - Parameters of the mixture

$$\boldsymbol{\theta} = \{\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \dots, \pi_K, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K\}$$

M-Step: $\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} \sum_i \sum_k q_{ik} \log p(k) p(\boldsymbol{x}_i|k)$

- We can separate updates of single components and coefficients
  - just additive objectives in lower bound
- **Update coefficients:**

$$\boldsymbol{\pi} = \arg\max_{\boldsymbol{\pi}} \sum_i \sum_k q_{ik} \log \pi_k$$

- **Update components:**

$$\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k = \arg\max_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k} \sum_i q_{ik} \log \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

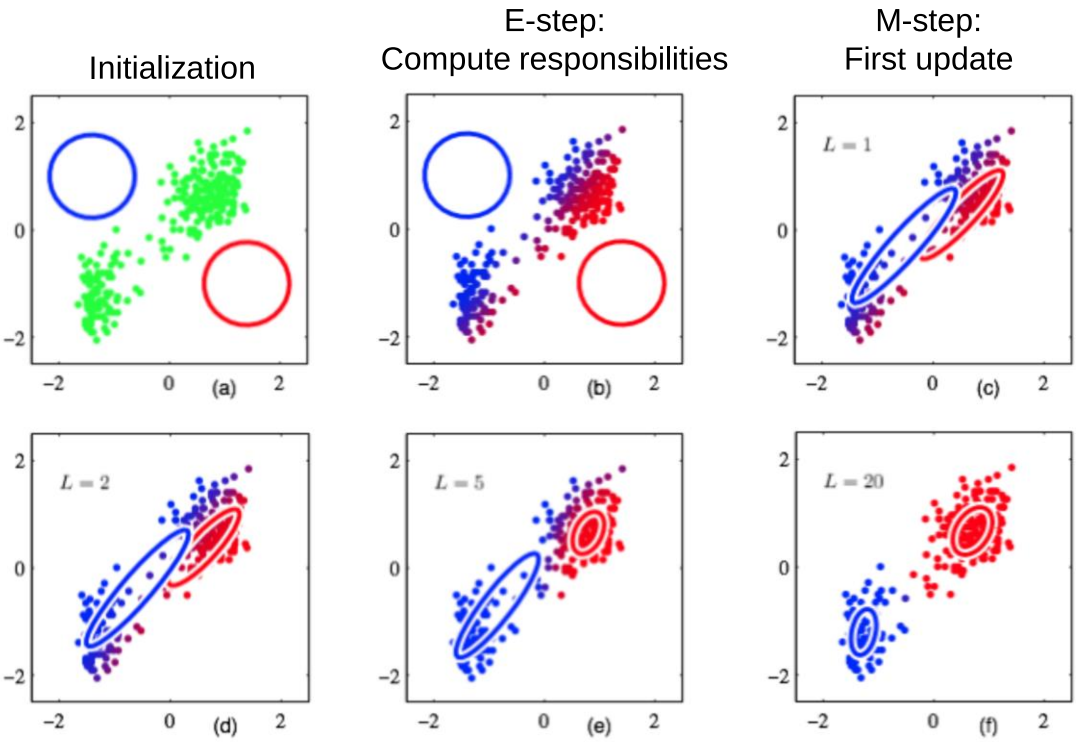
  - Each data-point is weighted by q_{ik}
  - Weighted maximum likelihood estimate

EM for Mixture of Gaussians

Weighted Maximum Likelihood updates:

- **Update coefficients:** $\boldsymbol{\pi} = \arg\max_{\boldsymbol{\pi}} \sum_i \sum_k q_{ik} \log \pi_k$
  - Result: $\pi_k = \frac{\sum_i q_{ik}}{\sum_k \sum_i q_{ik}} = \frac{\sum_i q_{ik}}{N}$
- **Update components:** $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k = \arg\max_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k} \sum_i q_{ik} \log \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
  - Mean: $\boldsymbol{\mu}_k = \frac{\sum_i q_{ik} \boldsymbol{x}_i}{\sum_i q_{ik}}$
  - Covariance: $\boldsymbol{\Sigma}_k = \frac{\sum_i q_{ik} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T}{\sum_i q_{ik}}$

Illustration



- Each component represents a cluster in the data set
- EM is very sensitive to the initialization

EM versus k-means

K-means can be seen as special case of EM with:

- Co-Variances are always set to 0 (in the limit)
- **E-Step / Assignment Step:**
  - responsibilities q_{ik} of nearest cluster k are set to 1, all other values are 0

- **M-Step / Adjustment Step:**
  - Update for the mean is the same
  - Co-Variances are ignored (set to close to 0)

- EM is harder to learn than k-means but it also gives you variances and densities
- Often k-means is used to initialize the means for EM

Practical considerations…

How many mixture components do we need?

- More components will typically lead to a better likelihood
- But are more components necessarily better? Not always, because of overfitting!
- It's again a model-selection problem (cross-validate on a validation-set)
- Bayesian methods can be used to integrate out number of components (tricky to get them to work)

How do we initialize:

- EM can give very poor results with wrong initialization
- Most common approach:
  - Use k-means (simple clustering algorithm) to initialize the centers
  - Use a fixed value for the covariance

Additional Notes

EM assumes that E-step can set the KL to zero:

- Lower bound is tight
- Marginal likelihood always improves

For more complex latent variable models (e.g. continuous z), this is not possible:

- In this case, we also have to use an approximate *q(z)* that minimizes

$$q(\boldsymbol{z}) = \arg\min_q \mathrm{KL}(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x}))$$

- Hence, $\mathrm{KL}(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x})) > 0$ after the E-step
  - No guaranteed improvement of the marginal likelihood
  - Still improvement of the lower bound
- Algorithms that do that are called Variational Bayes / Variational Inference
  - Very active research, underlying algorithm of many deep learning architectures (e.g. variational autoencoder)

Takeaway messages

You know now:

- The difference between parametric and non-parametric models
- Different non-parametric models (histogram, kernel density estimation and k-nearest neighbors)
- What mixture models and latent variable models are
- What the Expectation-Maximization idea and algorithm are
- Why does EM converge
- How to apply EM to GMMs

Self-test questions

- What are parametric methods and how to obtain their parameters?
- How many parameters have non-parametric methods?
- What are mixture models?
- Should gradient methods be used for training mixture models?
- How does the EM algorithm work?
- What is the biggest problem of mixture models?
- How does EM decomposes the marginal likelihood?
- Why does EM always improve the lower bound?
- Why does EM always improve the marginal likelihood