Statistical Machine Learning

Lecture 06: Probability Density Estimation

Kristian Kersting
TU Darmstadt

Summer Term 2020

Today's Objectives

- Make you understand how to do find $p(x)$
- Covered Topics
  - Density Estimation
  - Maximum Likelihood Estimation
  - Non-Parametric Methods
  - Mixture Models
  - Expectation Maximization

Outline

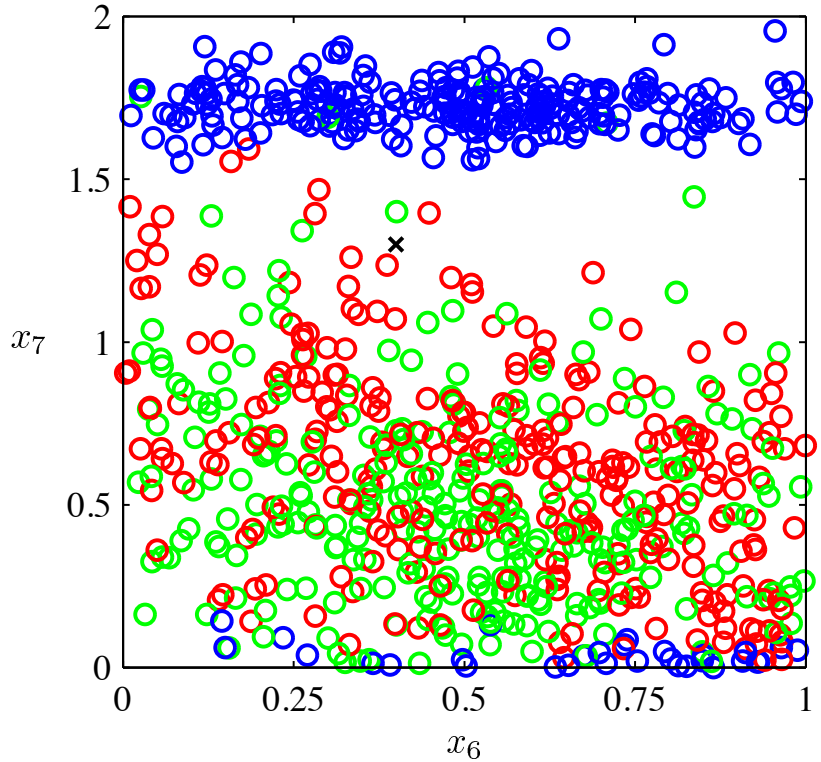
1. **Probability Density**
2. **Parametric models**
   - Maximum Likelihood Method
3. **Non-Parametric Models**
   - Histograms
   - Kernel Density Estimation
   - K-nearest Neighbors
4. **Mixture models**
5. **Wrap-Up**

TECHNISCHE UNIVERSITÄT DARMSTADT

Outline

TECHNISCHE UNIVERSITÄT DARMSTADT

Training Data



- How do we get the probability distributions from this so that we can classify with them?

TECHNISCHE UNIVERSITÄT DARMSTADT

Probability Density Estimation

- So far we have seen:
  - Bayes optimal classification, based on probability distributions $p(x \mid C_k)p(C_k)$
- The prior $p(C_k)$ is easy to deal with. We can "just count" the number of occurrences of each class in the training data
- We need to estimate (learn) the class-conditional probability density $p(x \mid C_k)$
  - Supervised training: we know the input data points and their true labels (classes)
  - Estimate the density separately for each class C_k
  - "Abbreviation": $p(x) = p(x \mid C_k)$

Probability Density Estimation

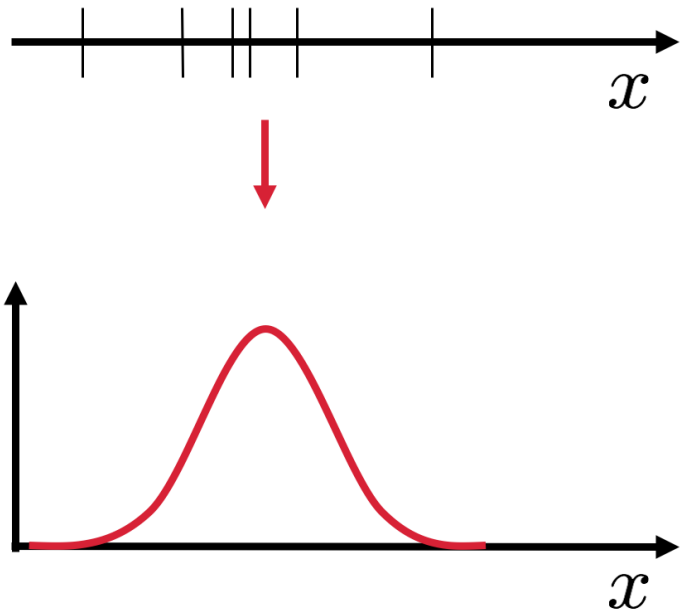
- Training data

$$x_1, x_2, x_3, \ldots$$

- Estimation

$$p(x)$$

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



TECHNISCHE UNIVERSITÄT DARMSTADT

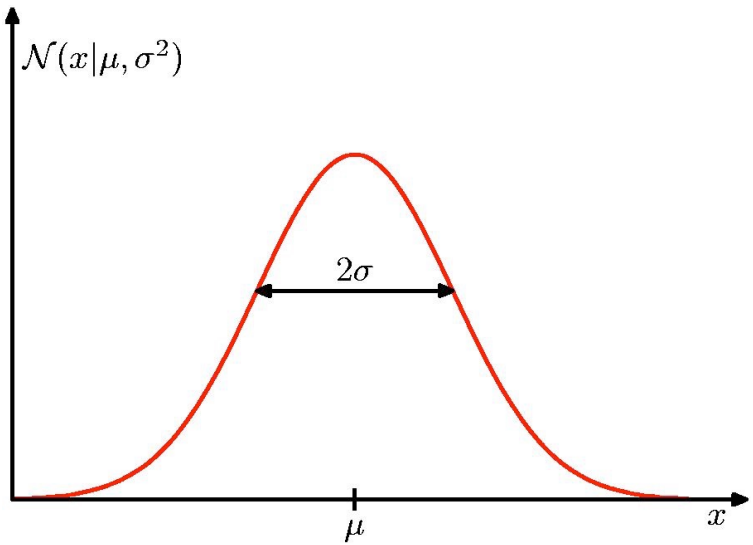
Outline

TECHNISCHE UNIVERSITÄT DARMSTADT

2. Parametric models

- Simple case: Gaussian Distribution

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



- Is governed by two parameters: mean and variance. That is, if we know these parameters we can fully describe $p(x)$

TECHNISCHE UNIVERSITÄT DARMSTADT

2. Parametric models

- Notation for parametric density models

$$x \sim p(x \mid \theta)$$

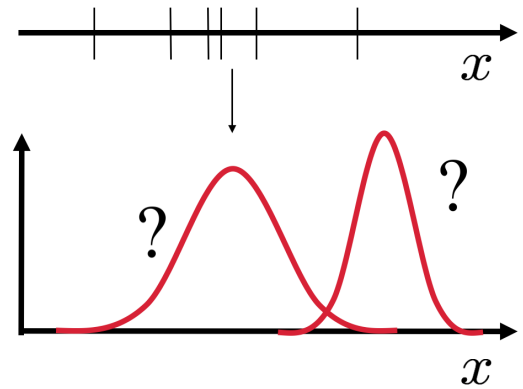
- For the Gaussian distribution

$$\theta = (\mu, \sigma)$$
$$x \sim p\left(x \middle| \mu, \sigma\right)$$

TECHNISCHE UNIVERSITÄT DARMSTADT

2. Parametric models

- Learning means to estimate the parameters θ given the training data $X = \{x_1, x_2, \ldots\}$



- Likelihood of θ is defined as the probability that the data X was generated from the probability density function with parameters θ

$$L(\theta) = p(X \mid \theta)$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Maximum Likelihood Method

- Consider a set of points $X = \{x_1, \ldots, x_N\}$
- Computing the likelihood
  - Of a single datum? $p(x_n|\theta)$
  - Of all data?
- Assumption: the data is i.i.d. (independent and identically distributed)
  - The random variables x_1 and x_2 are independent if

$$P(x_1 \leq \alpha, x_2 \leq \beta) = P(x_1 \leq \alpha)\, P(x_2 \leq \beta) \quad \forall \alpha, \beta \in \mathbb{R}$$

  - The random variables x_1 and x_2 are identically distributed if

$$P(x_1 \leq \alpha) = P(x_2 \leq \alpha) \quad \forall \alpha \in \mathbb{R}$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Maximum Likelihood Method

- Likelihood

$$L(\theta) = p(X \mid \theta) = p\left(x_1, \ldots, x_N \,\middle|\, \theta\right)$$

(using the i.i.d. assumption)

$$= p(x_1 \mid \theta) \cdot \ldots \cdot p(x_n \mid \theta)$$

$$= \prod_{n=1}^{N} p(x_n \mid \theta)$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Maximum log-Likelihood Method

- Maximize the (log-)likelihood w.r.t. θ

$$\log L(\theta) = \log p(X \mid \theta) = \log \prod_{n=1}^{N} p(x_n \mid \theta) = \sum_{n=1}^{N} \log p(x_n \mid \theta)$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Maximum Likelihood Method - Gaussian

- Maximum likelihood estimation of a Gaussian

$$\hat{\mu}, \hat{\sigma} = \arg\max_{\mu,\sigma} \log L(\theta) = \log p(X \mid \theta) = \sum_{n=1}^{N} \log p\left(x_n \,\middle|\, \mu, \sigma\right)$$

- Take the partial derivatives and set them to zero

$$\frac{\partial L}{\partial \mu} = 0, \frac{\partial L}{\partial \sigma} = 0$$

- This leads to a closed form solution

$$\hat{\mu} = \frac{1}{N} \sum_{n=1}^{N} x_n$$

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{n=1}^{N} (x_n - \hat{\mu})^2$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Maximum Likelihood Method - Gaussian

Given a set of i.i.d. data $X = \{x_1, \ldots, x_N\}$ drawn from $\mathcal{N}(x; \mu, \Sigma)$, we want to estimate (μ, Σ) by MLE. The log-likelihood function is

$$\ln p(X|\mu, \Sigma) = -\frac{N}{2} \ln |\Sigma| - \frac{1}{2} \sum_{n=1}^{N} (x_n - \mu)^T \Sigma^{-1} (x_n - \mu) + \text{const}$$

Taking its derivative w.r.t. μ and setting it to zero we have

$$\hat{\mu} = \frac{1}{N} \sum_{n=1}^{N} x_n$$

Rewrite the log-likelihood using "trace trick",

$$\begin{aligned}
\ln p(X|\mu, \Sigma) &= -\tfrac{N}{2} \ln |\Sigma| - \tfrac{1}{2} \sum_{n=1}^{N} (x_n - \mu)^T \Sigma^{-1} (x_n - \mu) + \text{const} \\
&\propto -\tfrac{N}{2} \ln |\Sigma| - \tfrac{1}{2} \sum_{n=1}^{N} \text{Trace}\left(\Sigma^{-1} (x_n - \mu)(x_n - \mu)^T\right) \\
&= -\tfrac{N}{2} \ln |\Sigma| - \tfrac{1}{2} \text{Trace}\left(\Sigma^{-1} \sum_{n=1}^{N} [(x_n - \mu)(x_n - \mu)^T]\right)
\end{aligned}$$

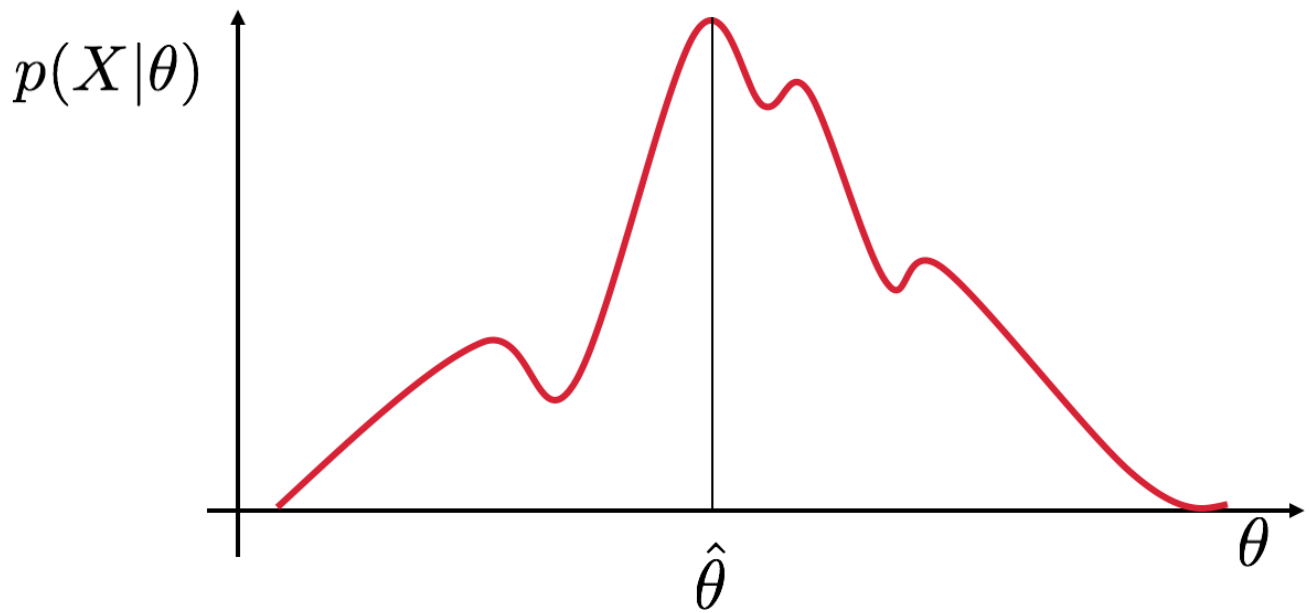
Taking the derivative w.r.t. Σ^{-1}, and using 1) $\frac{\partial}{\partial A} \log |A| = A^{-T}$; 2) $\frac{\partial}{\partial A} \mathtt{Tr}[AB] = \frac{\partial}{\partial A} \mathtt{Tr}[BA] = B^T$, we obtain

$$\hat{\Sigma} = \frac{1}{N} \sum_{n=1}^{N} (x_n - \hat{\mu}) (x_n - \hat{\mu})^T.$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Likelihood

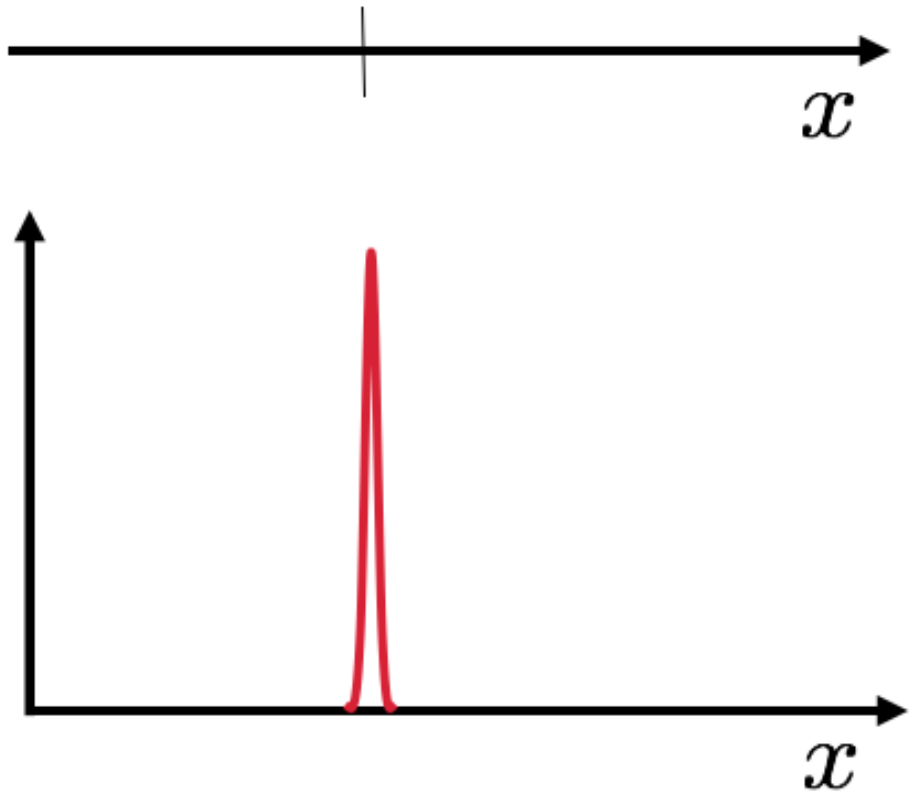
$$L(\theta) = p(X \mid \theta) = \prod_{n=1}^{N} p(x_n \mid \theta)$$



TECHNISCHE UNIVERSITÄT DARMSTADT

Degenerate case

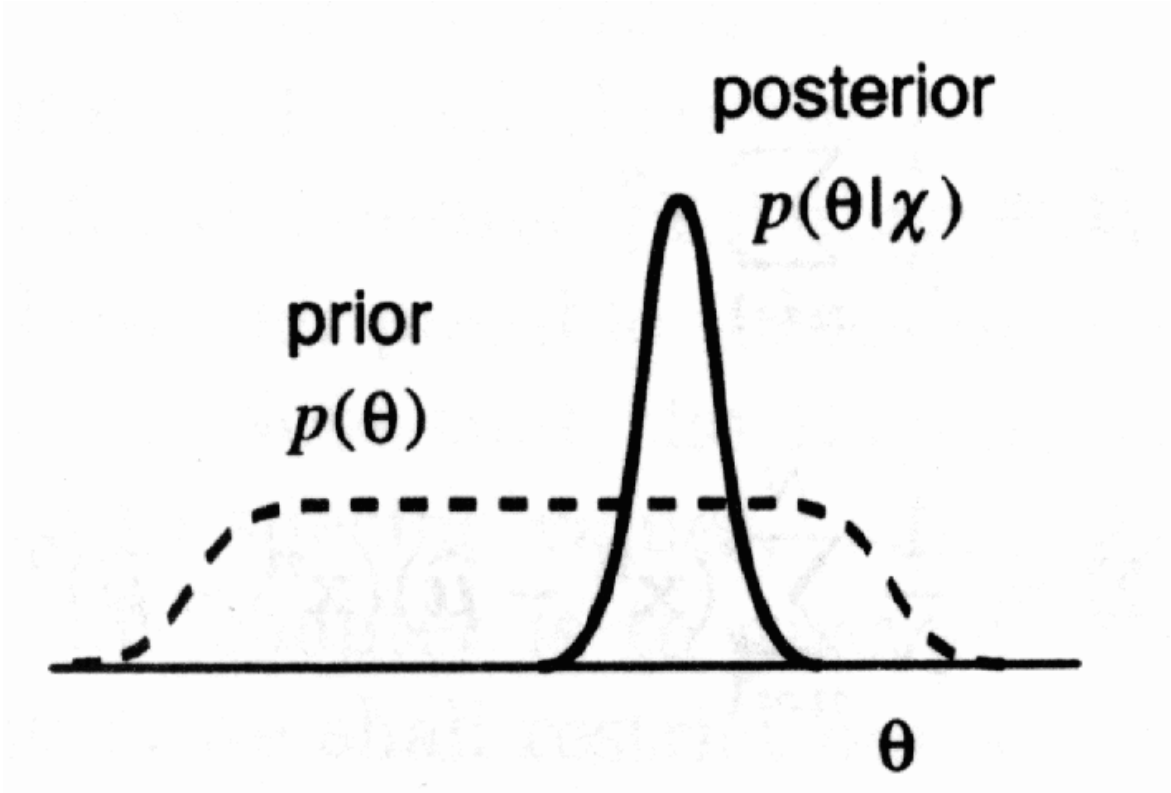
- If $N = 1$, $X = \{x_1\}$, the resulting Gaussian looks like



TECHNISCHE UNIVERSITÄT DARMSTADT

Degenerate case

- What can we do to still get a useful estimate?
- We can put a prior on the mean!



Bayesian Estimation

- Bayesian estimation / learning of parametric distributions, assumes that the parameters are not fixed, but are random variables too
- This allows us to use prior knowledge about the parameters
- How do we achieve that?
  - What do we want? A density model for x, $p(x)$
  - What do we have? Data X

Bayesian Estimation

- Formalize this as a conditional probability $p\left(x \middle| X\right)$

$$p\left(x \middle| X\right) = \int p\left(x, \theta \middle| X\right) \mathrm{d}\theta$$

$$p\left(x, \theta \middle| X\right) = p\left(x \middle| \theta, X\right) p\left(\theta \middle| X\right)$$

- $p(x)$ can be fully determined with the parameters θ, i.e., θ is a sufficient statistic
- Hence, we have $p\left(x \middle| \theta, X\right) = p\left(x \middle| \theta\right)$

$$p\left(x \middle| X\right) = \int p\left(x \middle| \theta\right) p\left(\theta \middle| X\right) \mathrm{d}\theta$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Bayesian Estimation

$$p\left(x\middle|X\right) = \int p\left(x\middle|\theta\right) p\left(\theta\middle|X\right) \mathrm{d}\theta$$

$$p\left(\theta\middle|X\right) = \frac{p\left(X\middle|\theta\right) p\left(\theta\right)}{p\left(X\right)} = L\left(\theta\right) \frac{p\left(\theta\right)}{p\left(X\right)}$$

$$p\left(X\right) = \int p\left(X\middle|\theta\right) p\left(\theta\right) \mathrm{d}\theta = \int L\left(\theta\right) p\left(\theta\right) \mathrm{d}\theta$$

$$p\left(x\middle|X\right) = \frac{1}{p\left(X\right)} \int p\left(x\middle|\theta\right) L\left(\theta\right) p\left(\theta\right) \mathrm{d}\theta$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Bayesian Estimation

$$p\left(x \middle| X\right) = \int p\left(x \middle| \theta\right) p\left(\theta \middle| X\right) \mathrm{d}\theta$$

- The probability $p\left(\theta \middle| X\right)$ makes it explicit how the parameter estimation depends on the training data
- If $p\left(\theta \middle| X\right)$ is small in most places, but large for a single $\hat{\theta}$ then we can approximate

$$p\left(x \middle| X\right) \approx p\left(x \middle| \hat{\theta}\right)$$

  - Sometimes referred to as the Bayes point
- The more uncertain we are about estimating $\hat{\theta}$, the more we average

TECHNISCHE UNIVERSITÄT DARMSTADT

Bayesian Estimation

- Problem: In general, it is intractable to integrate out the parameters θ (or only possible to do so numerically)
- Example with closed form solution
  - Gaussian data distribution, the variance is known and fixed
  - We estimate the distribution of the mean

$$p\left(\mu \middle| X\right) = \frac{p\left(X \middle| \mu\right) p\left(\mu\right)}{p\left(X\right)}$$

  - With prior

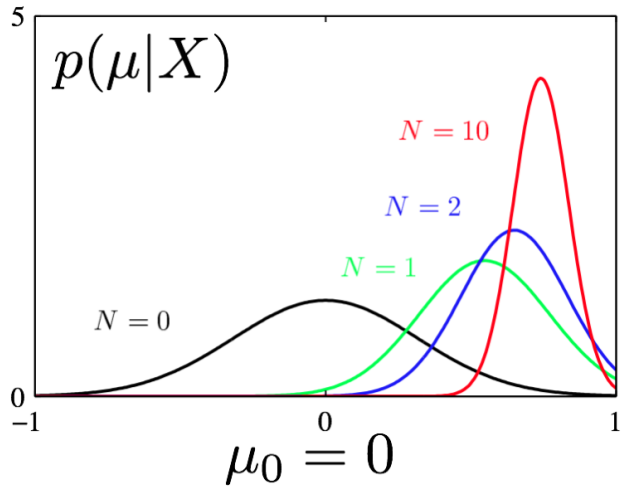
$$p\left(\mu\right) = \mathcal{N}\left(\mu_0, \sigma_0^2\right)$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Bayesian Estimation

- Sample mean

$$\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n$$



- Bayesian estimation

$$p\left(\mu \middle| X\right) \sim \mathcal{N}\left(\mu_N, \sigma_N^2\right)$$

$$\mu_N = \frac{N\sigma_0^2 \bar{x} + \sigma^2 \mu_0}{N\sigma_0^2 + \sigma^2}, \qquad \frac{1}{\sigma_N^2} = \frac{N}{\sigma^2} + \frac{1}{\sigma_0^2}$$

- Check what happens when N grows to infinity...

Conjugate Priors

- Conjugate Priors are prior distributions for the parameters that do not "change" the type of the parametric model
- For example, as we saw that a Gaussian prior on the mean is conjugate to the Gaussian model. This works here because...
  - The product of two Gaussians is a Gaussian
  - The marginal of a Gaussian is a Gaussian
- In general, it is not as easy!

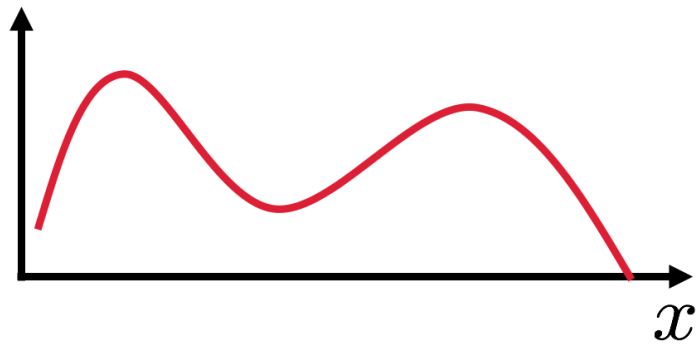
TECHNISCHE UNIVERSITÄT DARMSTADT

Outline

TECHNISCHE UNIVERSITÄT DARMSTADT

3. 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

TECHNISCHE UNIVERSITÄT DARMSTADT

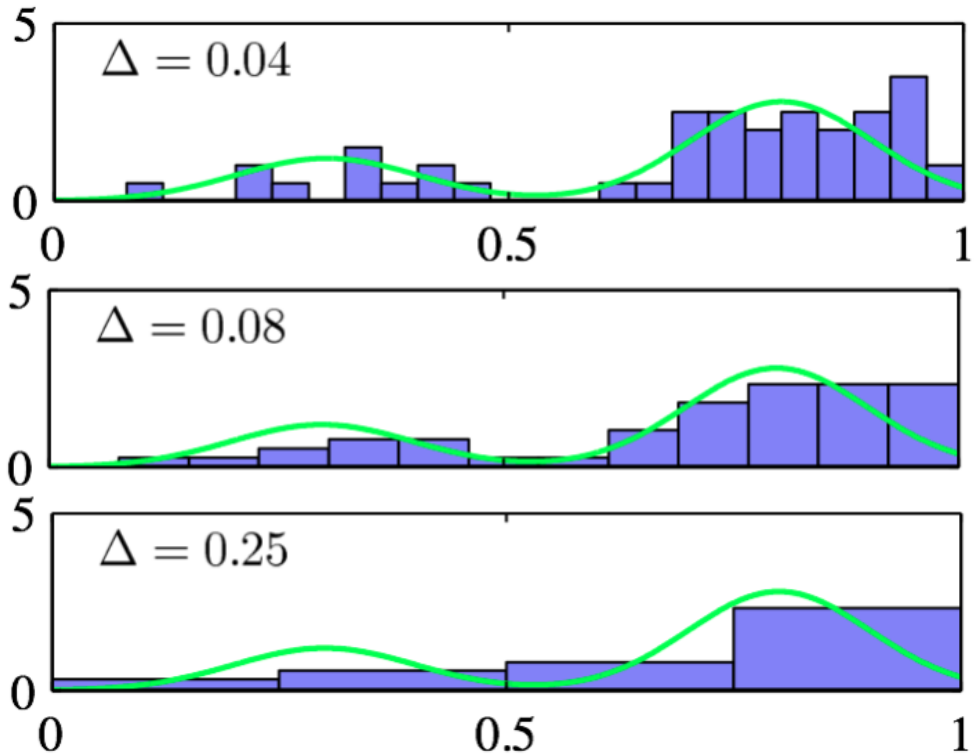
Histograms

- Discretize the feature space into bins

Not smooth enough

About right

Too smooth



TECHNISCHE UNIVERSITÄT DARMSTADT

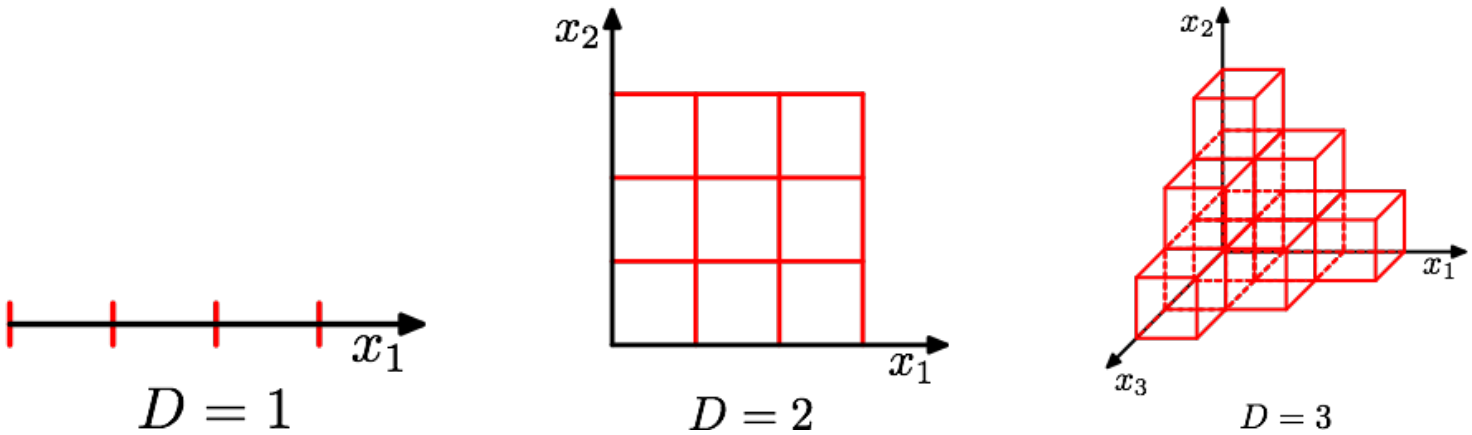
Histograms

- 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?

TECHNISCHE UNIVERSITÄT DARMSTADT

Curse of Dimensionality

- For histograms



- We will see that it is a general issue that we have to keep in mind

TECHNISCHE UNIVERSITÄT DARMSTADT

More formally

- Data point $\mathbf{x}$ is sampled from probability density $p(\mathbf{x})$
- Probability that $\mathbf{x}$ falls in region R

$$P(\mathbf{x} \in R) = \int_R p(\mathbf{x}) \, \mathrm{d}\mathbf{x}$$

- If R is sufficiently small, with volume V, then $p(\mathbf{x})$ is almost constant

$$P(\mathbf{x} \in R) = \int_R p(\mathbf{x}) \, \mathrm{d}\mathbf{x} \approx p(\mathbf{x}) \, V$$

- If R is sufficiently large

$$P(\mathbf{x} \in R) = \frac{K}{N} \implies p(\mathbf{x}) \approx \frac{K}{NV}$$

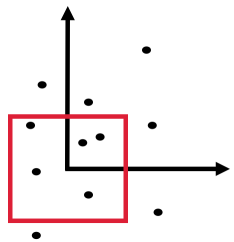
where N is the number of total points and K is the number of points falling in the region R

TECHNISCHE UNIVERSITÄT DARMSTADT

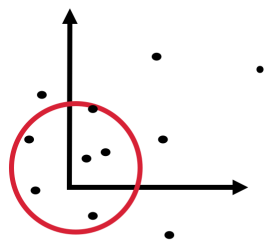
More formally

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

- 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



Parzen Window

- Hypercubes in d dimensions with edge length h

$$H(\mathbf{u}) = \begin{cases} 1 & \left|u_j\right| \leq \frac{h}{2},\ j = 1, \ldots, d \\ 0 & \text{otherwise} \end{cases}$$

$$V = \int H(\mathbf{u})\, \mathrm{d}\mathbf{u} = h^d$$

$$K(\mathbf{x}) = \sum_{n=1}^{N} H\left(\mathbf{x} - \mathbf{x}^{(n)}\right)$$

$$p(\mathbf{x}) \approx \frac{K(\mathbf{x})}{NV} = \frac{1}{Nh^d} \sum_{n=1}^{N} H\left(\mathbf{x} - \mathbf{x}^{(n)}\right)$$

Gaussian Kernel

$$H(\mathbf{u}) = \frac{1}{\left(\sqrt{2\pi h^2}\right)^d} \exp\left\{-\frac{\|\mathbf{u}\|^2}{2h^2}\right\}$$

$$V = \int H(\mathbf{u})\,\mathrm{d}\mathbf{u} = 1$$

$$K(\mathbf{x}) = \sum_{n=1}^{N} H\left(\mathbf{x} - \mathbf{x}^{(n)}\right)$$

$$p(\mathbf{x}) \approx \frac{K(\mathbf{x})}{NV} = \frac{1}{N\left(\sqrt{2\pi h^2}\right)^d} \sum_{n=1}^{N} \exp\left\{-\frac{\left\|\mathbf{x} - \mathbf{x}^{(n)}\right\|^2}{2h^2}\right\}$$

TECHNISCHE UNIVERSITÄT DARMSTADT

General formulation - arbitrary kernel

$$k(u) \geq 0, \quad \int k(u)\,\mathrm{d}u = 1$$

$$V = h^d$$

$$K(\mathbf{x}) = \sum_{n=1}^{N} k\left(\frac{\left\|\mathbf{x} - \mathbf{x}^{(n)}\right\|}{h}\right)$$

$$p(\mathbf{x}) \approx \frac{K(\mathbf{x})}{NV} = \frac{1}{Nh^d} \sum_{n=1}^{N} k\left(\frac{\left\|\mathbf{x} - \mathbf{x}^{(n)}\right\|}{h}\right)$$

Common Kernels

- Gaussian Kernel

$$k(u) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}u^2\right\}$$

  - Problem: kernel has infinite support
  - Requires a lot of computation

- Parzen window

$$k(u) = \begin{cases} 1 & |u| \leq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

  - Not very smooth results

Common Kernels

- Epanechnikov kernel

$$k(u) = \max\left\{0, \frac{3}{4}(1-u)^2\right\}$$

  - Smoother, but finite support
- Problem with kernel methods: We have to select the kernel bandwidth h appropriately

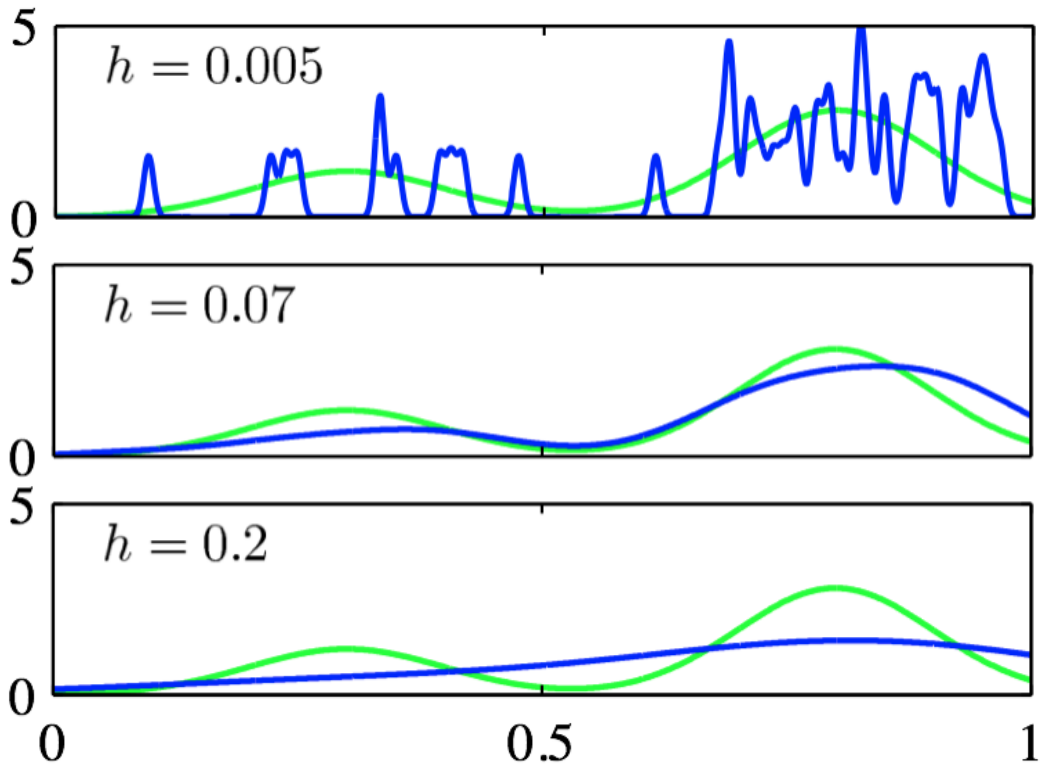
TECHNISCHE UNIVERSITÄT DARMSTADT

Gaussian KDE Example

Not smooth enough

About right

Too smooth

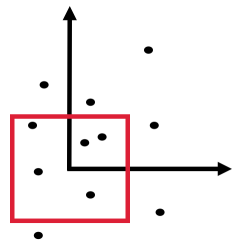


TECHNISCHE UNIVERSITÄT DARMSTADT

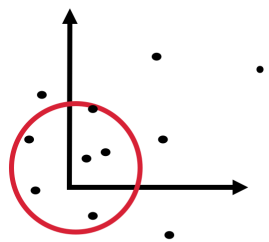
Again to our definition

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

- 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



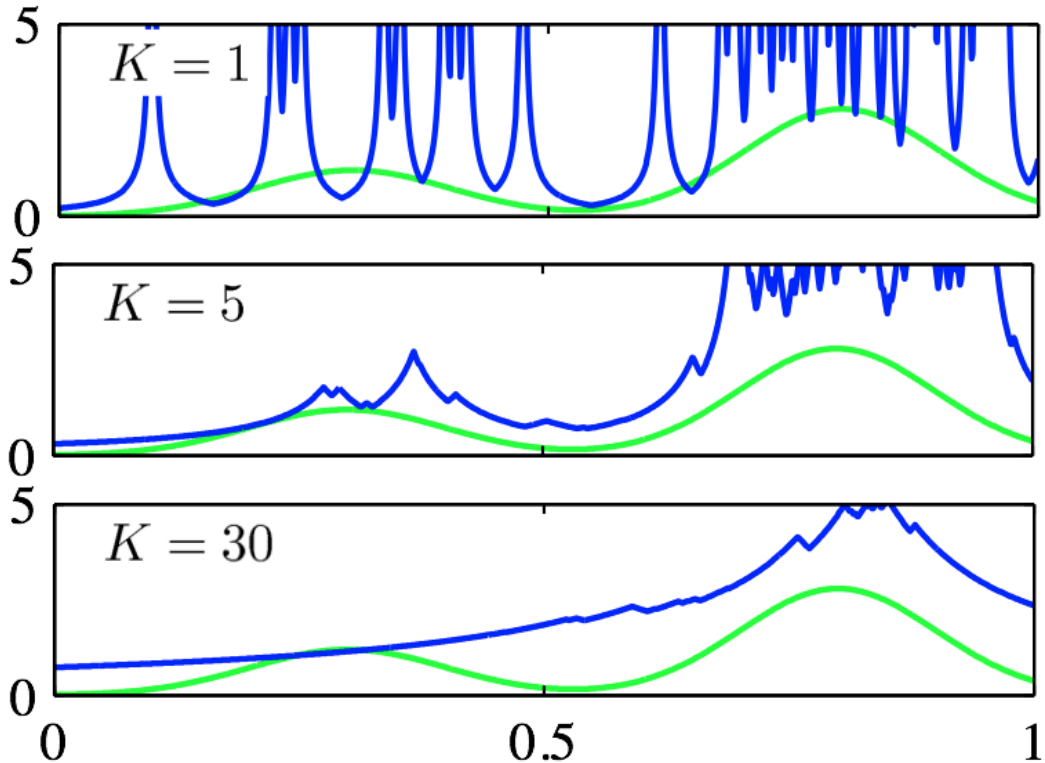
TECHNISCHE UNIVERSITÄT DARMSTADT

K-Nearest Neighbors (kNN)

Not smooth enough

About right

Too smooth



- Note: Blue rescaled for visualization

K-Nearest Neighbors (kNN)

- Bayesian classification

$$P\left(C_j \middle| x\right) = \frac{P\left(x \middle| C_j\right) P\left(C_j\right)}{P\left(x\right)}$$

- k-Nearest Neighbors classification
  - Assume we have a dataset of N points, where N_j is the number of data points in class C_j and $\sum_j N_j = N$. To classify a point x we draw a sphere centered in x that contains K points (from any classes). Assume the sphere has volume V and contains K_j points of class C_j

$$P\left(x\right) \approx \frac{K}{NV}, \quad P\left(x \middle| C_j\right) \approx \frac{K_j}{N_j V}, \quad P\left(C_j\right) \approx \frac{N_j}{N}$$

$$P\left(C_j \middle| x\right) \approx \frac{K_j}{N_j V} \frac{N_j}{N} \frac{NV}{K} = \frac{K_j}{K}$$

Bias-Variance Problem

- 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, including parametric and mixture models

TECHNISCHE UNIVERSITÄT DARMSTADT

Outline

TECHNISCHE UNIVERSITÄT DARMSTADT

4. Mixture models

Parametric models

- Gaussian, Neural Networks, ...
- Good analytic properties
- Simple
- Small memory requirements
- Fast

Nonparametric models

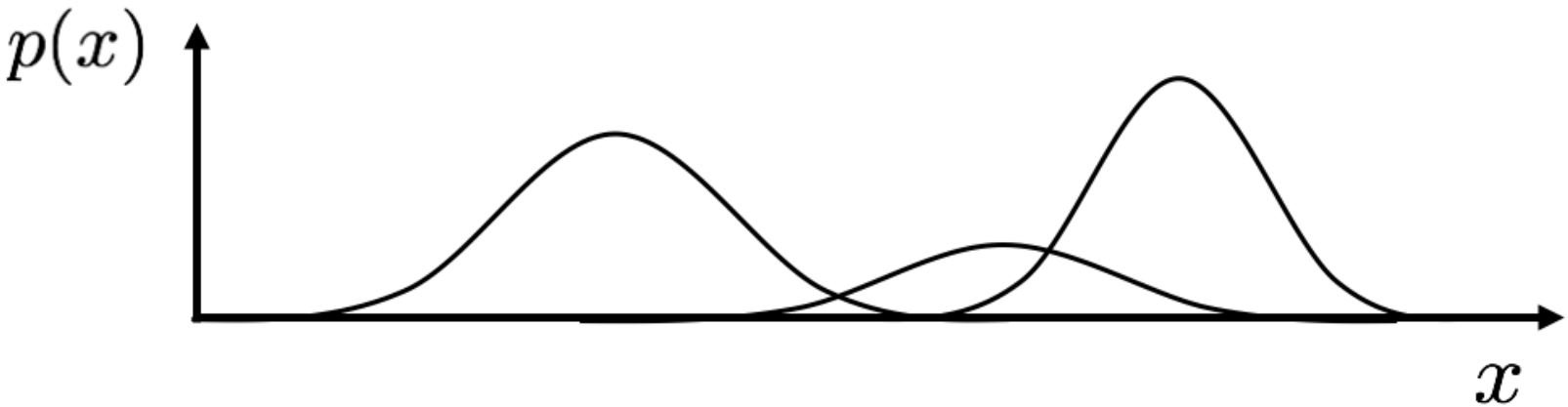
- Kernel Density Estimation, k-Nearest Neighbors, ...
- General
- Large memory requirements
- Slow

Mixture models are a mix of parametric and nonparametric models

TECHNISCHE UNIVERSITÄT DARMSTADT

Mixture of Gaussians (MoG)

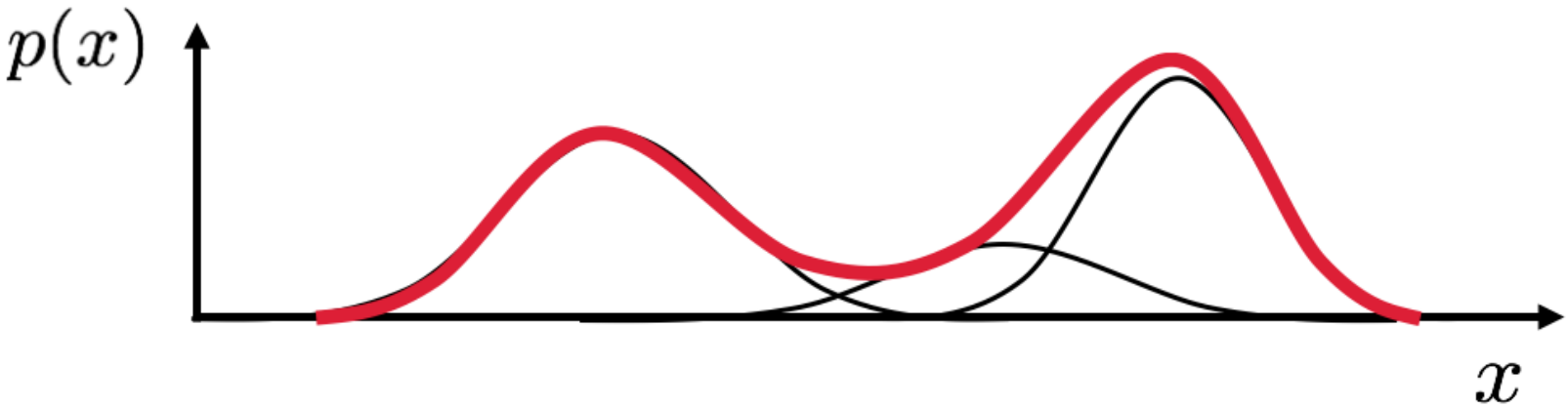
- Sum of individual Gaussian distributions



TECHNISCHE UNIVERSITÄT DARMSTADT

Mixture of Gaussians

- Sum of individual Gaussian distributions



- In the limit (i.e. with many mixture components) this can approximate every (smooth) density

$$p(x) = \sum_{j=1}^{M} p\left(x \middle| j\right) p(j)$$

Mixture of Gaussians

$$p(x) = \sum_{j=1}^{M} p\left(x \middle| j\right) p(j)$$

$$p\left(x \middle| j\right) = \mathcal{N}\left(x \middle| \mu_j, \sigma_j\right) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left\{-\frac{(x-\mu_j)^2}{2\sigma_j^2}\right\}$$

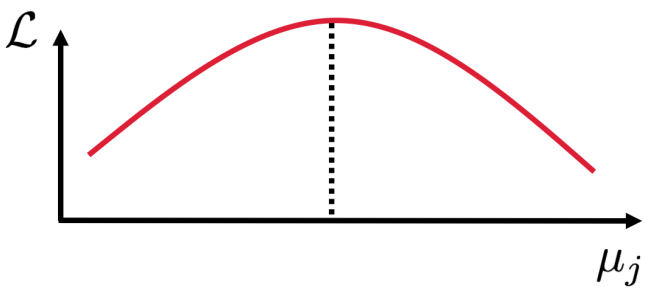
$$p(j) = \pi_j \quad \text{with} \quad 0 \leq \pi_j \leq 1, \sum_{j=1}^{M} \pi_j = 1$$

- Remarks
  - The mixture density integrates to 1: $\int p(x)\,\mathrm{d}x = 1$
  - The mixture parameters are: $\theta = \{\mu_1, \sigma_1, \pi_1, \ldots, \mu_M, \sigma_M, \pi_M\}$

Mixture of Gaussians - MLE

- Maximum (log-)Likelihood Estimation
  - Dataset with N i.i.d. points $\{x_1, \ldots, x_N\}$

$$\mathcal{L} = \log L(\theta) = \sum_{n=1}^{N} \log p\left(x_n \middle| \theta\right)$$



$$\frac{\partial \mathcal{L}}{\partial \mu_j} = 0$$

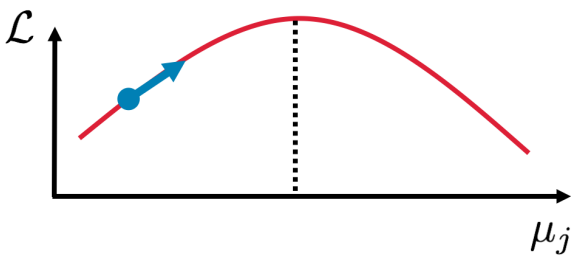
$$\mu_j = \frac{\sum_{n=1}^{N} p\left(j \middle| x_n\right) x_n}{\sum_{n=1}^{N} p\left(j \middle| x_n\right)}$$

- What is the problem with this approach?
- Circular dependency - No analytical solution!

Mixture of Gaussians - MLE Gradient Ascent

- Maximum (log-)Likelihood Estimation
  - Dataset with N i.d.d. points $\{x_1, \ldots, x_N\}$

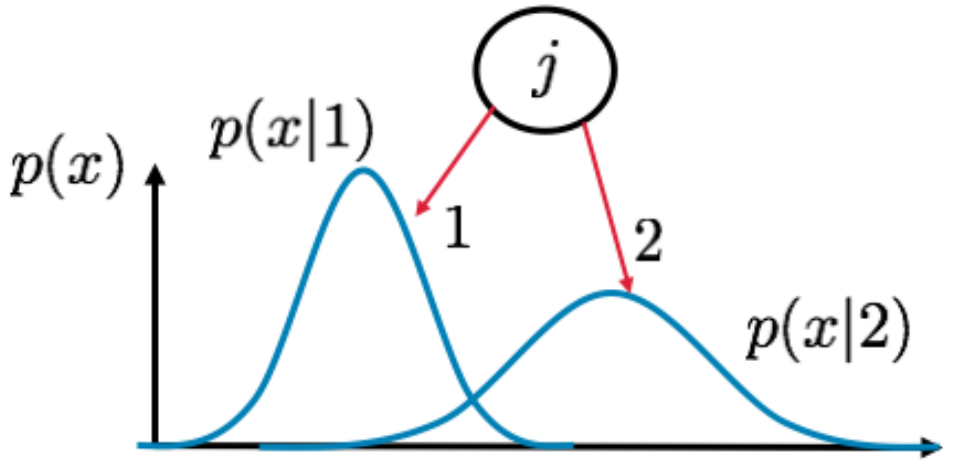
$$\mathcal{L} = \log L(\theta) = \sum_{n=1}^{N} \log p\left(x_n \middle| \theta\right)$$



$$\frac{\partial \mathcal{L}}{\partial \mu_j} = 0$$

- Gradient ascent
  - Complex gradient (nonlinear, circular dependencies)
  - Optimization of one Gaussian component depends on all other components

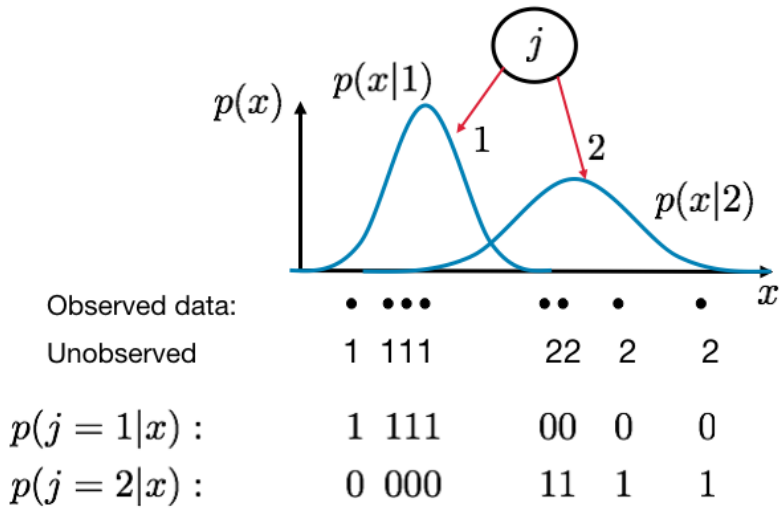
Mixture of Gaussians - Different strategy



Observed data:	•	•••	••	•	•
Unobserved	1	111	22	2	2
$p(j=1\|x):$	1	111	00	0	0
$p(j=2\|x):$	0	000	11	1	1

Unobserved := hidden or latent variables $(j|x)$

Mixture of Gaussians - Different strategy

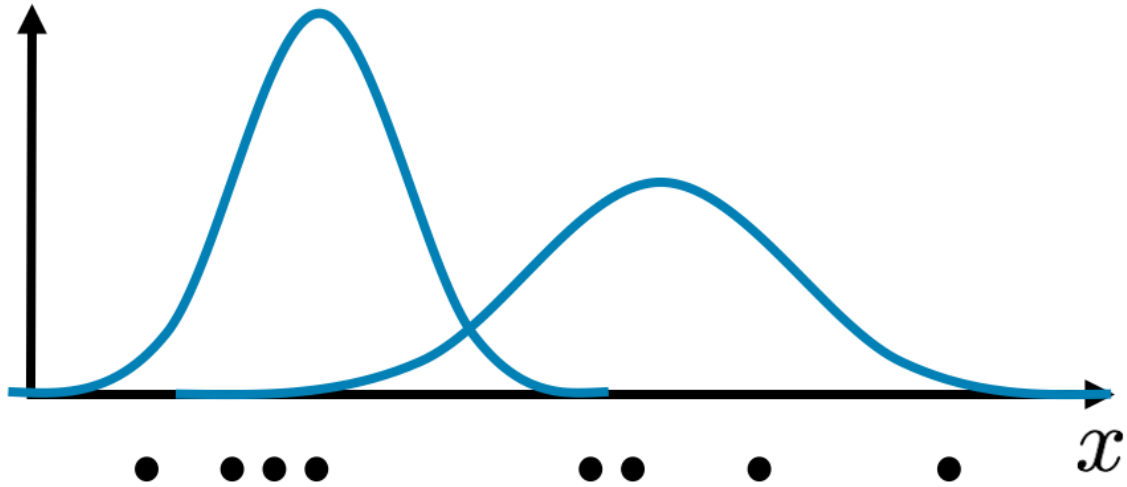


- Suppose we knew the observed and unobserved dataset (also called the *complete* dataset)
- Then we can compute the maximum likelihood solution of components 1 and 2

$$\mu_1 = \frac{\sum_{n=1}^{N} p\left(1 \middle| x_n\right) x_n}{\sum_{n=1}^{N} p\left(1 \middle| x_n\right)} \quad \mu_2 = \frac{\sum_{n=1}^{N} p\left(2 \middle| x_n\right) x_n}{\sum_{n=1}^{N} p\left(2 \middle| x_n\right)}$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Mixture of Gaussians - Different strategy



- Suppose we knew the distributions
- We can infer the unobserved data using Bayes Decision Rule. Namely we decide 1 if

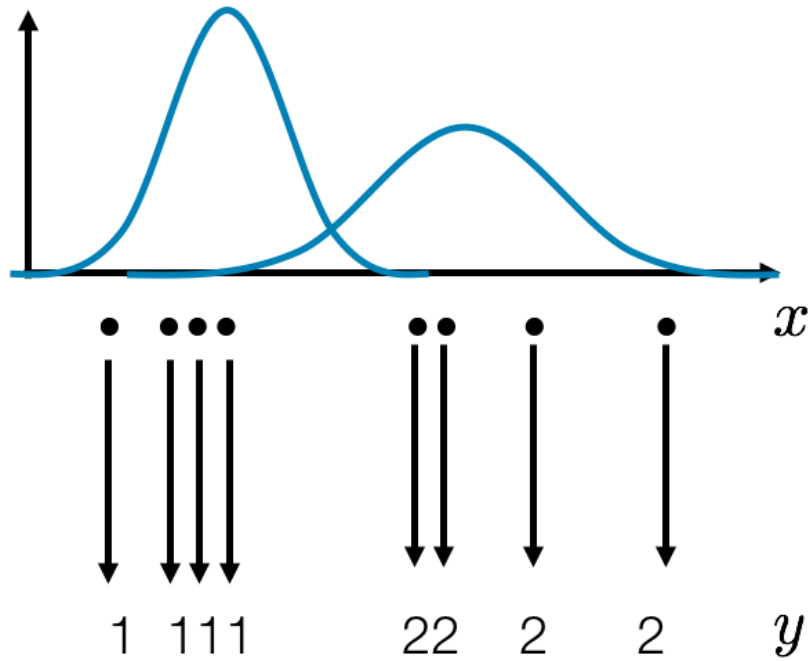
$$p\left(j=1\middle|x\right) > p\left(j=2\middle|x\right)$$

Mixture of Gaussians - Chicken and Egg problem

- We have big problem at hand... we neither know the distribution nor the unobserved data!
- To break this loop, we need some estimation of the unobserved data j
- Temporary solution: Clustering (to be replaced soon)

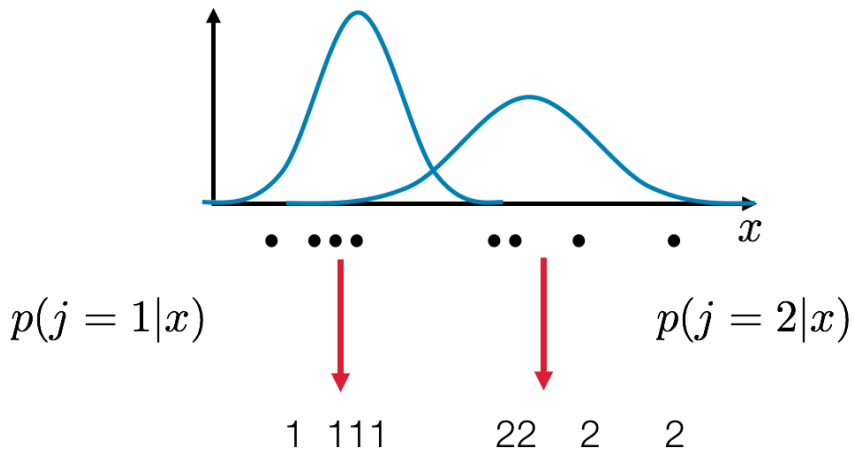
Estimation using Clustering

- Clustering with *hard* assignments
- Somehow assign mixture labels to each data point
- Estimate the mixture component only from its data



Mixture of Gaussians

- Suppose we had a guess about the distribution, but did not know the unobserved data



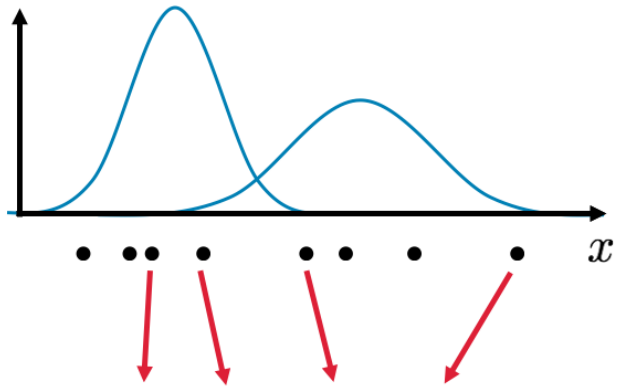
- Compute the probability for each mixture component:

$$p\left(j=1\,\middle|\,x\right)=\frac{p\left(x\,\middle|\,1\right)p\left(1\right)}{p\left(x\right)}=\frac{p\left(x\,\middle|\,1\right)\pi_1}{\sum_{j=1}^{M}p\left(x\,\middle|\,j\right)\pi_j}$$

$$p\left(j=2\,\middle|\,x\right)=\frac{p\left(x\,\middle|\,2\right)p\left(2\right)}{p\left(x\right)}=\frac{p\left(x\,\middle|\,2\right)\pi_2}{\sum_{j=1}^{M}p\left(x\,\middle|\,j\right)\pi_j}$$

Expectation Maximization - Clustering

- Clustering with *soft* assignments
- Expectation-step of the EM-algorithm (shortly)



$p(1\|x)$	0.99	0.8	0.2	0.01
$p(2\|x)$	0.01	0.2	0.8	0.99

- We can determine the means by maximum likelihood estimation

$$\mu_j = \frac{\sum_{n=1}^{N} p\left(j \middle| x_n\right) x_n}{\sum_{n=1}^{N} p\left(j \middle| x_n\right)}$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Expectation Maximization Algorithm

Algorithm

- Initialize with (random) means: $\mu_1, \mu_2, \ldots, \mu_M$
- While stop-condition is not met
  - **E-step**: Compute the posterior distribution for each mixture component and for all data points

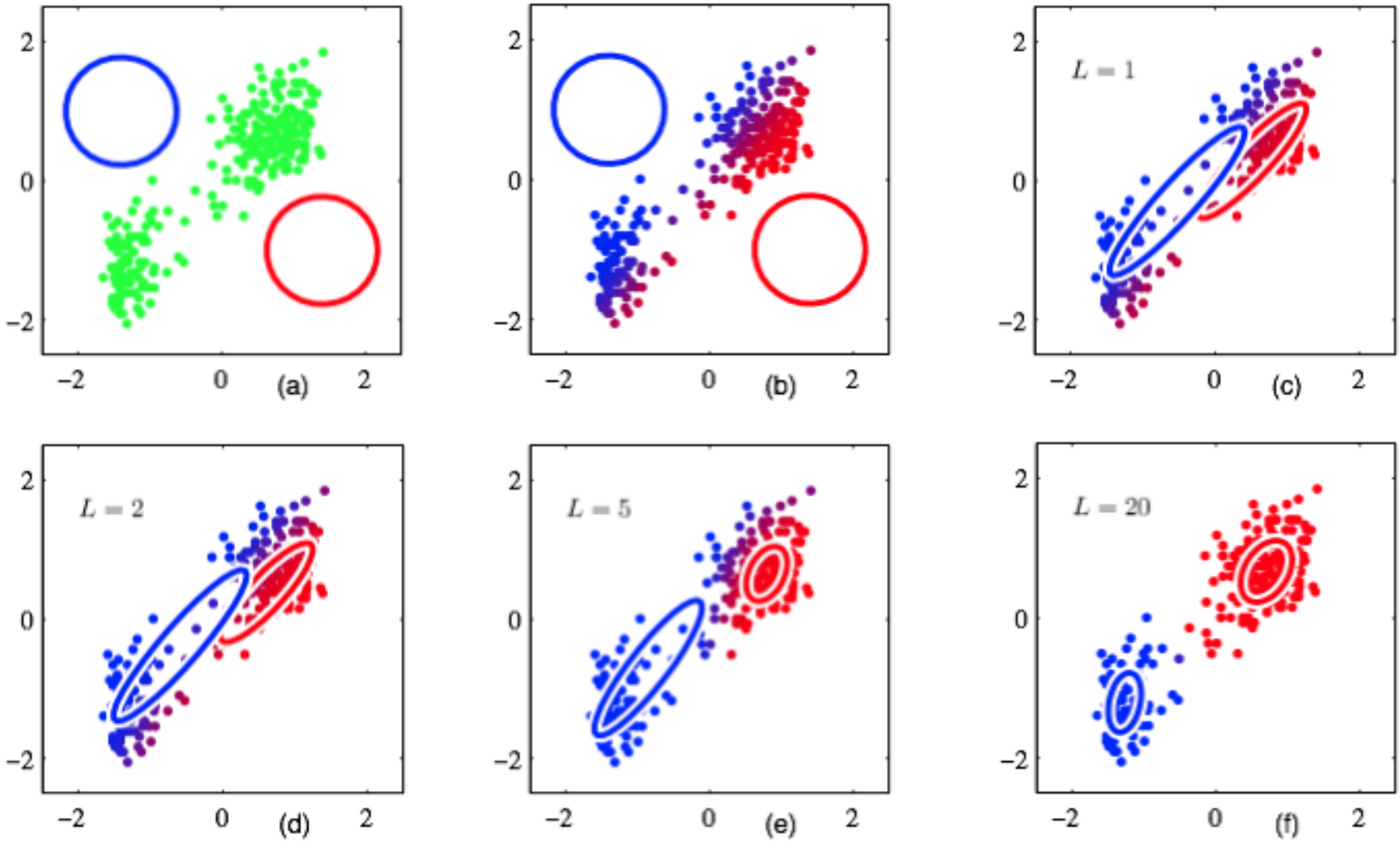
$$p\left(j \middle| x_n\right)$$

  - **M-step**: Compute the new means as the weighted means of all data points

$$\mu_j = \frac{\sum_{n=1}^{N} p\left(j \middle| x_n\right) x_n}{\sum_{n=1}^{N} p\left(j \middle| x_n\right)}$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Expectation Maximization



TECHNISCHE UNIVERSITÄT DARMSTADT

Expectation Maximization (EM) Algorithm

- **Expectation-Maximization (EM) Algorithm**
  - Method for performing maximum likelihood estimation, even when the data is *incomplete* (i.e. we only have access to observed variables)
  - Idea: if we have unknown values in our estimation problem (so-called hidden variables) we can use EM
- Assume:
  - Observed (incomplete) data: $X = \{x_1, \ldots, x_N\}$
  - Unobserved (hidden) data: $Y = \{y_1, \ldots, y_N\}$
- In case of Gaussian mixtures:
  - Association of every data point to one of the mixture components

TECHNISCHE UNIVERSITÄT DARMSTADT

Properties of EM

- Incomplete (observed) data: $X = \{x_1, \ldots, x_N\}$
- Hidden (unobserved) data: $Y = \{y_1, \ldots, y_N\}$
- Complete data: $Z = (X, Y)$
- Joint density

$$p(Z) = p(X, Y) = p\left(Y \middle| X\right) p(X)$$

  - With parameters

$$p\left(Z \middle| \theta\right) = p\left(X, Y \middle| \theta\right) = p\left(Y \middle| X, \theta\right) p\left(X \middle| \theta\right)$$

- In the case of Gaussian mixtures
  - $p\left(X \middle| \theta\right)$ - likelihood of the mixture model
  - $p\left(Y \middle| X, \theta\right)$ - predictions of the mixture component

TECHNISCHE UNIVERSITÄT DARMSTADT

Properties of EM

- Incomplete likelihood

$$\mathcal{L}\left(\theta \middle| X\right) = p\left(X \middle| \theta\right) = \prod_{n=1}^{N} p\left(x_n \middle| \theta\right)$$

- Complete likelihood

$$\begin{aligned}\mathcal{L}\left(\theta \middle| Z\right) &= p\left(Z \middle| \theta\right) = p\left(X, Y \middle| \theta\right) = p\left(Y \middle| X, \theta\right) p\left(X \middle| \theta\right) \\ &= \prod_{n=1}^{N} p\left(y_n \middle| x_n, \theta\right) p\left(x_n \middle| \theta\right)\end{aligned}$$

TECHNISCHE UNIVERSITÄT DARMSTADT

EM Algorithm

- We don't know Y, but if we have the current guess θ^{i-1} of the parameters θ, we can it use that to predict Y
- Formally we compute the expected value of the (complete) log-likelihood given the data X and the current estimation of θ

$$\mathbb{E}_Y \left[\log p \left(X, Y \middle| \theta \right) \middle| X, \theta^{i-1} \right] =: Q \left(\theta, \theta^{i-1} \right)$$

  - X - fixed; Y - random variable; θ - variable; θ^{i-1} - current estimation of the parameters (fixed)

TECHNISCHE UNIVERSITÄT DARMSTADT

Properties of the EM Algorithm

- Maximize the expected complete log-likelihood

$$
\begin{aligned}
Q\left(\theta, \theta^{i-1}\right) &= \mathbb{E}_Y\left[\log p\left(X, Y \middle| \theta\right) \middle| X, \theta^{i-1}\right] \\
&= \int p\left(y \middle| X, \theta^{i-1}\right) \log p\left(X, y \middle| \theta\right) \mathrm{d}y
\end{aligned}
$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Properties of the EM Algorithm

$$Q\left(\theta, \theta^{i-1}\right) = \int p\left(y \middle| X, \theta^{i-1}\right) \log p\left(X, y \middle| \theta\right) \mathrm{d}y$$

- **E-step (expectation)**: compute $p\left(y \middle| X, \theta^{i-1}\right)$ to be able to compute the expectation $Q\left(\theta, \theta^{i-1}\right)$
- **M-step (maximization)**: maximize the expected value of the complete log-likelihood

$$\theta^{i} = \arg\max_{\theta} Q\left(\theta, \theta^{i-1}\right)$$

Formal Properties of the EM Algorithm

Dempster (1929-)
Laird (1943-)
Rubin (1942-)

- Main result from Dempster et al, *Maximum Likelihood from Incomplete Data via the EM Algorithm*, 1977
  - The expected complete log-likelihood of the i-th iteration is at least as good as that of the (i-1)-th iteration:

$$Q\left(\theta^i, \theta^{i-1}\right) \geq Q\left(\theta^{i-1}, \theta^{i-1}\right)$$

  - If this expectation is maximized w.r.t. θ^i, then it holds that:

$$L\left(\theta^i \,\middle|\, X\right) \geq L\left(\theta^{i-1} \,\middle|\, X\right)$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Formal Properties of the EM Algorithm

- **Consequence of the previous statements**
  - The incomplete log-likelihood increases in every iteration (or at least stays the same)
  - The incomplete log-likelihood is maximized (locally)
- **In practice**
  - The quality of the results depends on the initialization
  - If we initialize poorly, we may get stuck in poor local optima
  - EM relies on good initialization of the parameters

TECHNISCHE UNIVERSITÄT DARMSTADT

Special case - Gaussian Mixtures

- For mixtures of Gaussians there is a closed form solution
- Look at the fully general case: also estimate the variances of the mixture components and the prior distribution over the mixture components

$$\theta^i = \arg\max_{\theta} Q\left(\theta, \theta^{i-1}\right)$$

EM for Gaussian Mixtures

Algorithm

- Initialize parameters: $\mu_1, \sigma_1, \pi_1 \ldots$
- While stop-condition is not met
  - **E-step**: Compute the posterior distribution, also called *responsibility*, for each mixture component and for all data points

$$\alpha_{nj} = p\left(j \,\middle|\, x_n\right) = \frac{\pi_j \mathcal{N}\left(x_n \,\middle|\, \mu_j, \sigma_j\right)}{\sum_{i=1}^{M} \pi_i \mathcal{N}\left(x_n \,\middle|\, \mu_i, \sigma_i\right)}$$

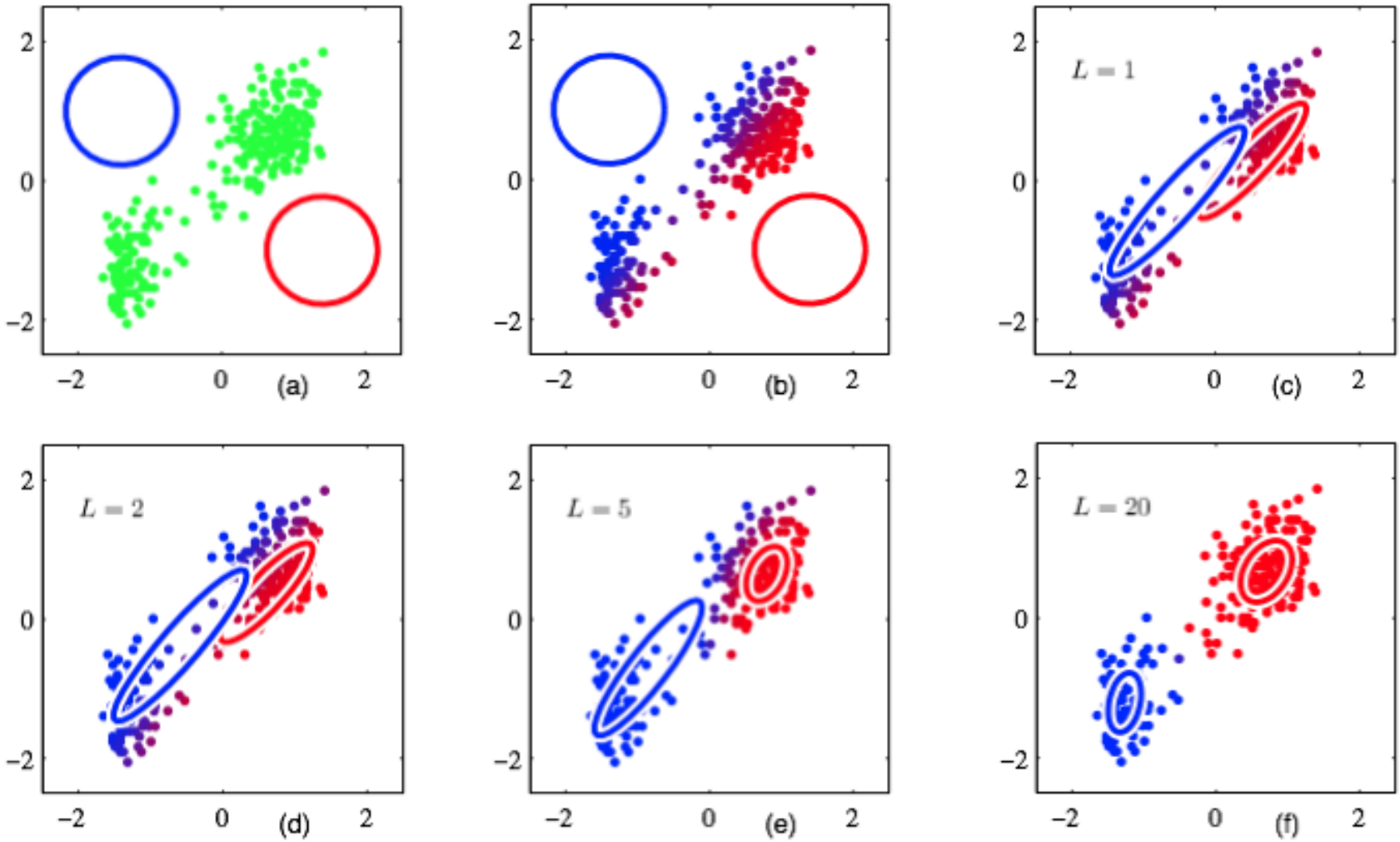
  - **M-step**: Compute the new parameters using weighted estimates

$$\mu_j^{\text{new}} = \frac{1}{N_j} \sum_{n=1}^{N} \alpha_{nj} x_n \qquad \text{with} \quad N_j = \sum_{n=1}^{N} \alpha_{nj}$$

$$\left(\sigma_j^{\text{new}}\right)^2 = \frac{1}{N_j} \sum_{n=1}^{N} \alpha_{nj} \left(x_n - \mu_j^{\text{new}}\right)^2, \quad \pi_j^{\text{new}} = \frac{N_j}{N}$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Expectation Maximization



TECHNISCHE UNIVERSITÄT DARMSTADT

How many components?

- 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!
- (Simple) automatic selection
  - Find K that maximizes the Akaike information criterion

$$\log p\left(X \middle| \theta_{ML}\right) - K$$

    - where K is the number of parameters
  - Or find K that maximizes the Bayesian information criterion

$$\log p\left(X \middle| \theta_{ML}\right) - \frac{1}{2} K \log N$$

    - where N is the number of data points

Before we move on... It is important to understand

- Mixture models are much more general than mixtures of Gaussians
  - One can have mixtures of any parametric distribution, and even mixtures of different parametric distributions
  - Gaussian mixtures are only one of many possibilities, though by far the most common one
- Expectation maximization is not just for fitting mixtures of Gaussians
  - One can fit other mixture models with EM
  - EM is still more general, in that it applies to many other hidden variable models

TECHNISCHE UNIVERSITÄT DARMSTADT

Outline

TECHNISCHE UNIVERSITÄT DARMSTADT

5. Wrap-Up

You know now:

- The difference between parametric and non-parametric models
- More about the likelihood function and how to derive the maximum likelihood estimators for the Gaussian distribution
- What Bayesian estimation is
- Different non-parametric models (histogram, kernel density estimation and k-nearest neighbors)
- What mixture models are
- What the Expectation-Maximization idea and algorithm are

TECHNISCHE UNIVERSITÄT DARMSTADT

Self-Test Questions

- Where do we get the probability of data from?
- 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?

TECHNISCHE UNIVERSITÄT DARMSTADT

Homework

- Reading Assignment for next lecture
  - Clustering: Murphy ch. 25
  - Bias & Variance: Bishop ch. 3.2, Murphy ch. 6.4

TECHNISCHE UNIVERSITÄT DARMSTADT

References

- EM Standard Reference
  - A.P. Dempster, N.M. Laird, D.B. Rubin, *Maximum-Likelihood from incomplete data via EM algorithm*, In Journal Royal Statistical Society, Series B. Vol. 39, 1977
- EM Tutorial
  - Jeff A. Bilmes, *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*, TR-97-021, ICSI, U.C. Berkeley, CA, USA
- Modern interpretation
  - Neal, R.M. and Hinton, G.E., *A view of the EM algorithm that justifies incremental, sparse, and other variants, In Learning in Graphical Models*, M.I. Jordan (editor)