Machine Learning

Mixture Models and EM - Lecture IV

Course Outline

Basic Concepts

- Parametric Method,
- Bayesian Learning and Nonparametric Methods

Classical Approaches

- Clustering and Mixture of Gaussians
- Linear Discriminants

Ensemble Methods

- Ensemble Methods and Boosting
- Randomized Trees, Forest

Reinforcement Learning

- Classical Reinforcement Learning

Neural Networks and Deep Learning

- Foundations
- Optimization

Videos for This Lecture

- Repetition Video (Part 0)
- Mixture of Distributions (Part 1)
- K-means (Part 2)
- Expectation Maximisation (Part 3)
- Applications (Part 4)

Bayesian Learning Approach

Bayesian view:

- Consider the parameter vector ɵ as a random variable.
- When estimating the parameters from a dataset X, we compute

$$p(x|X) = \int p(x, \theta|X) d\theta$$

Assumption: given ɵ, this doesn't depend on X anymore

$$p(x, \theta|X) = p(x|\theta, \cancel{X}) p(\theta|X)$$

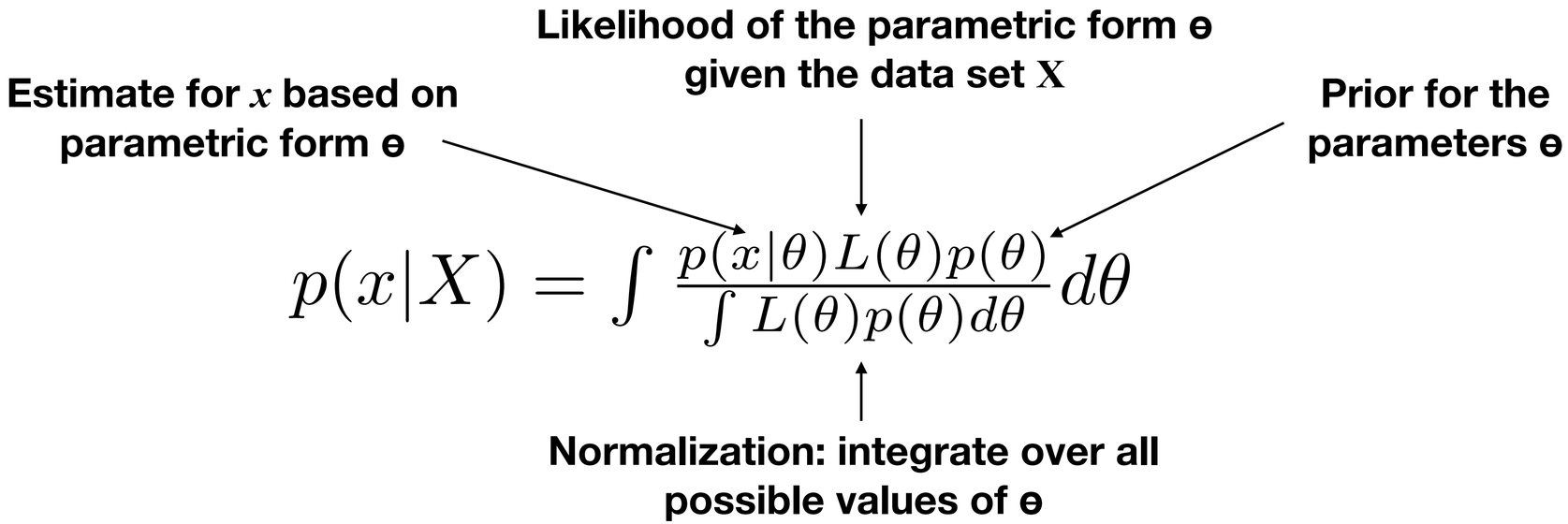
$$p(x|X) = \int p(x|\theta) p(\theta|X) d\theta$$

This is entirely determined by the parameter ɵ (i.e., by the parametric form of the pdf).

Slide source: B. Leibe Slides ML, 2015

Bayesian Learning Approach

Discussion



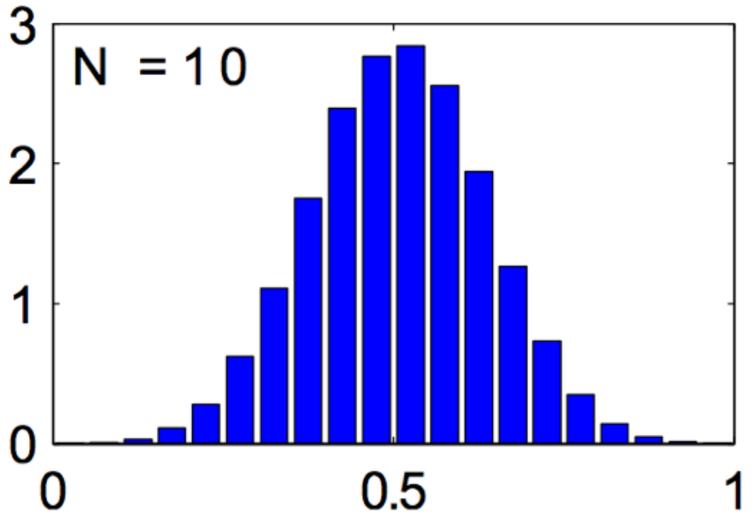
If we now plug in a (suitable) prior $p(\theta)$, we can estimate $p(x|X)$ from the data set X.

Slide source: B. Leibe Slides ML, 2015

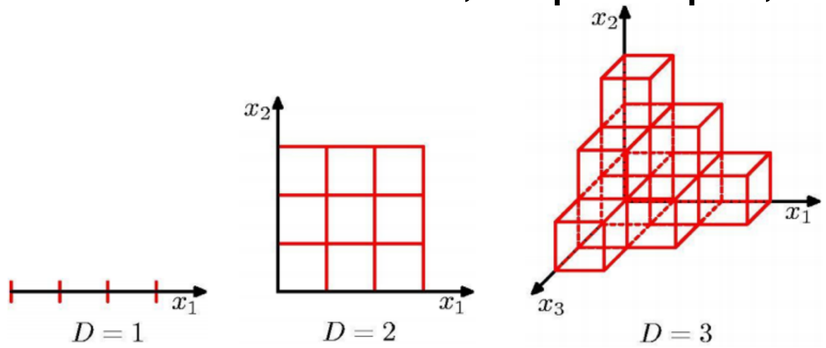
Recap: Histograms

- Basic idea:
  - Partition the data space into distinct bins with width Δ_i and count the number of observations, n_i, in each bin

$$p_i = \frac{n_i}{N\Delta_i}$$



  - Often, the same width is used for all bins, $\Delta_i = \Delta$
  - This can be done, in principle, for any dimensionality *D…*



…but the required number of bins grows exponentially with D!

Slide source: B. Leibe Slides ML, 2015

Recap: Statistically Better-Founded Approach

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

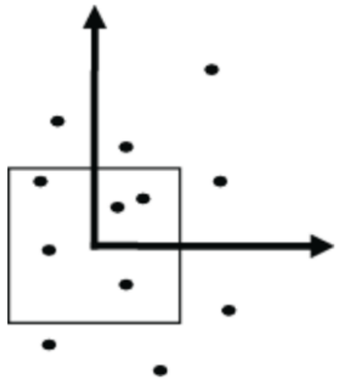
fixed V
determine K

fixed K
determine V

Kernel Methods

K-Nearest Neighbor

- **Kernel methods**
  - Example: Determine the number K of data points inside a fixed hypercube



- **K-Nearest Neighbor**
  - Increase the volume V until the K next data points are found.

Slide source: B. Leibe Slides ML, 2015

Today’s topics

Mixture distributions

- Mixture of Gaussians (MoG)
- Maximum Likelihood estimation attempt

K-Means Clustering

- Algorithm
- Applications

EM Algorithm

- Credit assignment problem
- MoG estimation
- EM Algorithm
- Interpretation of K-Means
- Technical advice

Applications

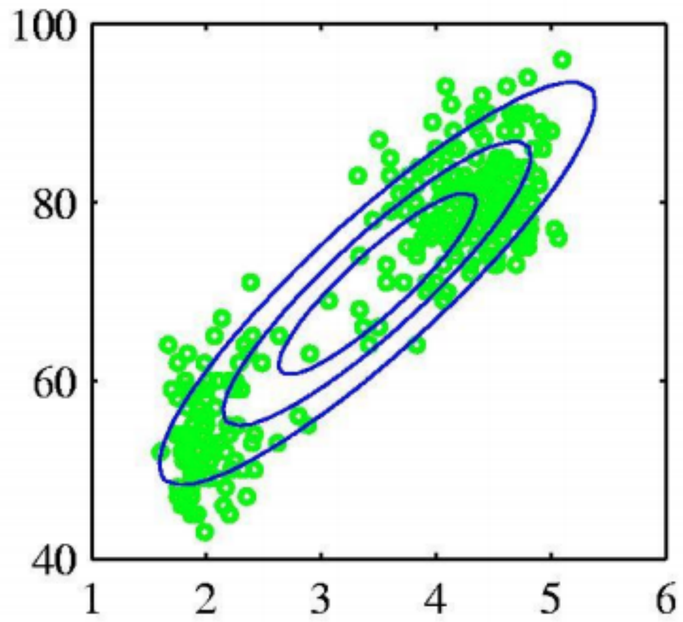
Part 1, Video MoGandEM_p1

- Mixture of Gaussians
- Maximum Likelihood estimation attempt

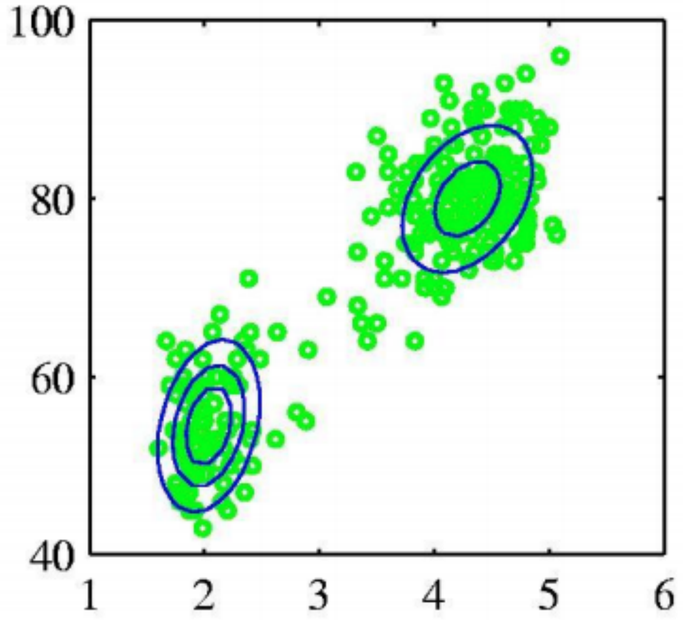
Mixture Distributions

A single parametric distribution is often not sufficient

- E.g. multimodel data



Single Gaussian

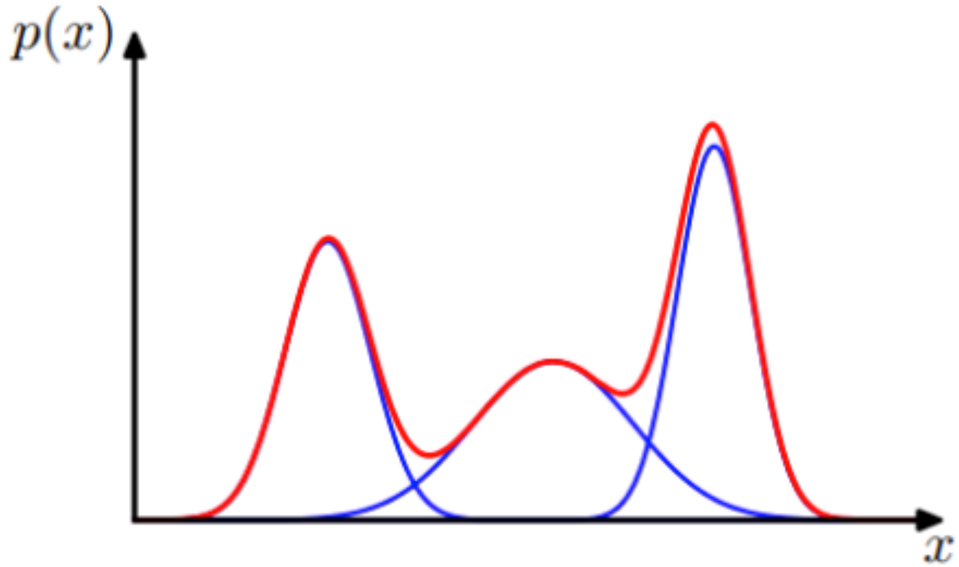


Mixture of two Gaussian

Image source: C.M. Bishop, 2006

Mixture of Gaussians (MoG)

Summer of M individual Normal distributions



In the limit, every smooth distribution can be approximated this way (if M is large enough)

$$p(x|\theta) = \sum_{j=1}^{M} p(x|\theta_j)p(j)$$

Image source: C.M. Bishop, 2006

Mixture of Gaussians (MoG)

$$p(x|\theta) = \sum_{j=1}^{M} p(x|\theta_j)p(j)$$

Likelihood of measurement x given mixture component j

$$p(x|\theta_j) = \mathcal{N}(x|\mu_i, \sigma_j^2) = \frac{1}{\sqrt{2\pi}\sigma_j} exp\{-\frac{(x-\mu_j)^2}{2\sigma_j^2}\}$$

$$p(j) = \pi_j \text{ with } 0 < \pi_j < 1 \text{ and } \sum_{j=1}^{M} \pi_j = 1$$

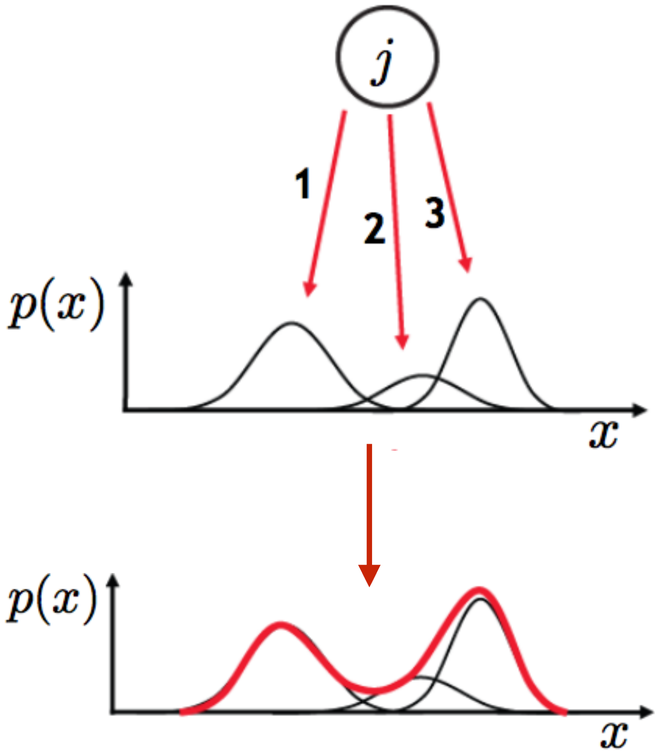
Prior of component j

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

Image source: B. Leibe Slides ML, 2015

Mixture of Gaussians (MoG)

Generative Model



$$p(j) = \pi_j$$

WEIGHT of mixture component

$$p(x|\theta_j)$$

Mixture component

Mixture density

$$p(x|\theta) = \sum_{j=1}^{M} p(x|\theta_j)p(j)$$

Image source: B. Leibe Slides ML, 2015

Mixture of Multivariate Gaussians

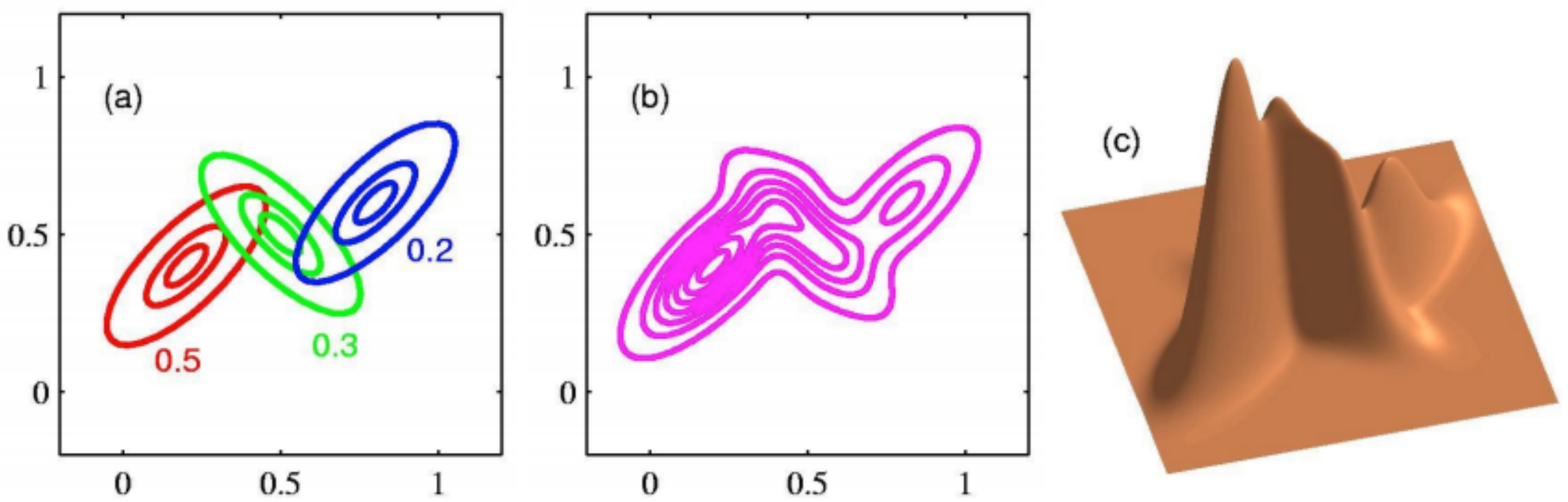


Image source: C.M. Bishop, 2006

Mixture of Multivariate Gaussians

Multivariate Gaussians

$$p(\mathbf{x}|\theta) = \sum_{j=1}^{M} p(\mathbf{x}|\theta_j)p(j)$$

$$p(\mathbf{x}|\theta_j) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}_j|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^{\mathrm{T}}\mathbf{\Sigma}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)\right\}$$

Mixture weights / mixture coefficients:

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

Parameters

$$\theta = (\pi_1, \mu_1, \sigma_1, \ldots, \pi_M, \mu_M, \sigma_M)$$

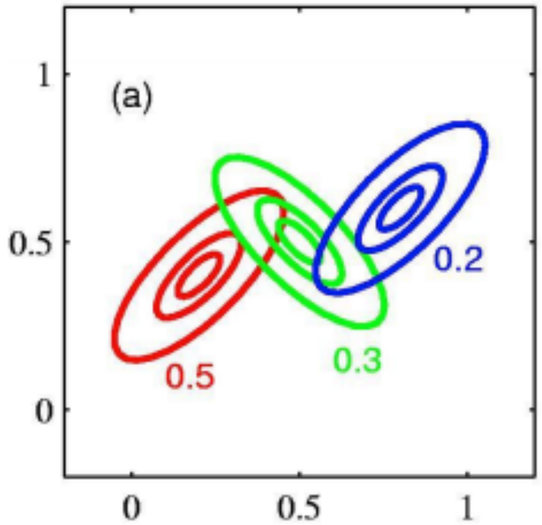


Image source: C.M. Bishop, 2006

Mixture of Multivariate Gaussians

Generative Model

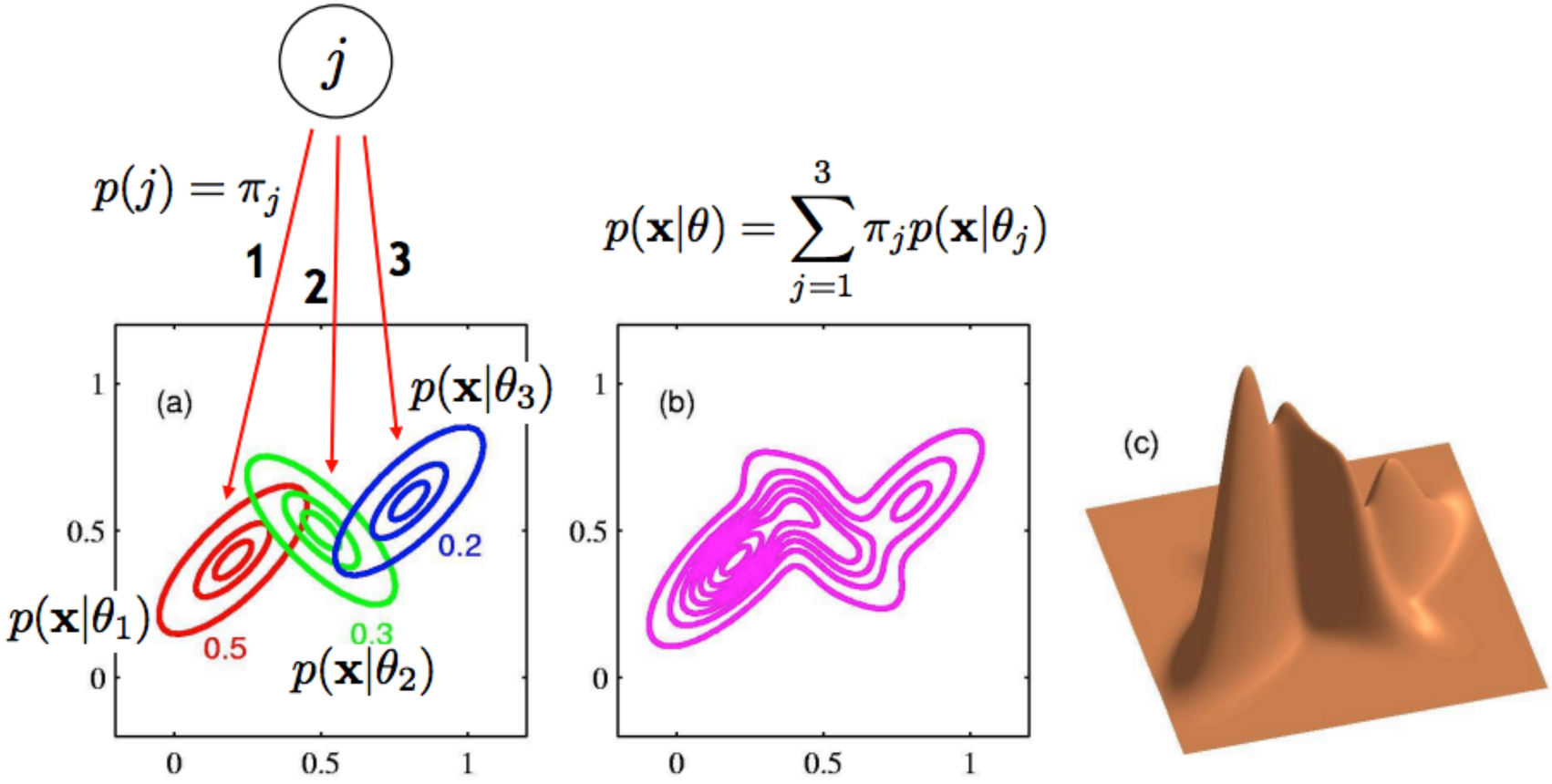


Image source: B. Leibe Slides ML, 2015

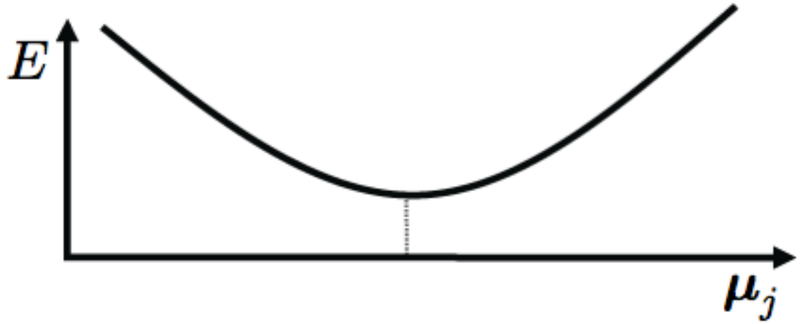
Mixture of Gaussians - 1st Estimation Attempt

Maximum Likelihood

$$\text{Minimize } E = -lnL(\theta) = -\sum_{j=1}^{N} \ln\, p(x_n|\theta)$$

Let us look at μ_j :

$$\frac{\partial E}{\partial \mu_j} = 0$$



We can already see that this will be difficult, since

$$\ln p(X|\pi,\mu,\Sigma) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)\}$$

This will cause problems!

Slide adapted from B. Leibe, ML, 2017

Mixture of Gaussians - 1st Estimation Attempt

$$\frac{\partial E}{\partial \mu_j} = -\sum_{n=1}^{N} \frac{\frac{\partial}{\partial \mu_j} p(x_n|\theta_j)}{\sum_{k=1}^{K} p(x_n|\theta_k)}$$

$$\frac{\partial}{\partial \mu_j}\mathcal{N}(x_n|\mu_k,\Sigma_k) = \Sigma^{-1}(x_n-\mu_j)\mathcal{N}(x_n|\mu_k,\Sigma_k)$$

$$= -\sum_{n=1}^{N} \left(\Sigma^{-1}(x_n-\mu_j)\frac{p(x_n|\theta_j)}{\sum_{k=1}^{K} p(x_n|\theta_k)}\right)$$

$$= -\cancel{\Sigma^{-1}} \sum_{n=1}^{N} (x_n-\mu_j)\frac{\pi_j \mathcal{N}(x_n|\mu_j,\Sigma_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)} = 0$$

$$= \gamma_j(x_n)$$

"responsibility" of component j for x_n

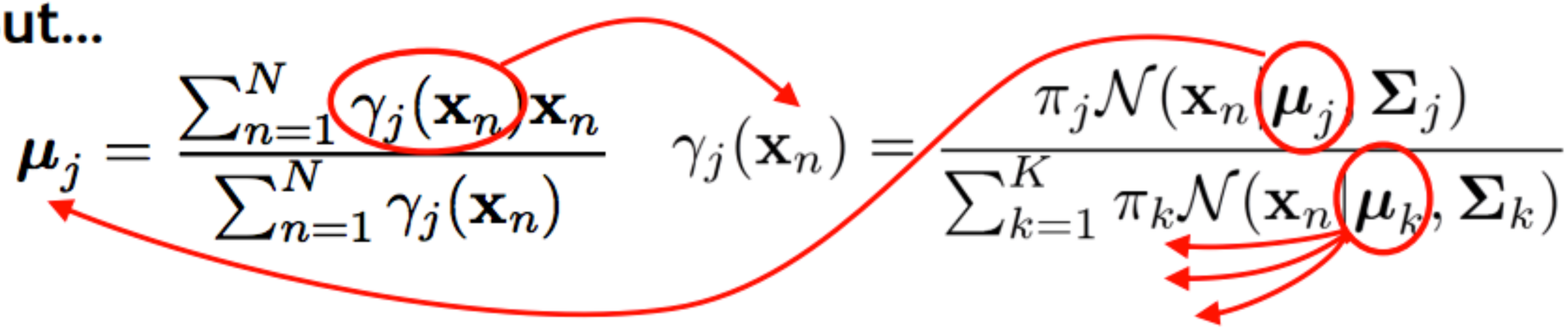
$$\Rightarrow \mu_j = \frac{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)}$$

Slide is taken from B. Leibe, ML, 2017

Mixture of Gaussians - 1st Estimation Attempt

But...

$$\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)} \qquad \gamma_j(\mathbf{x}_n) = \frac{\pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$



I.e. there is no direct analytical solution!

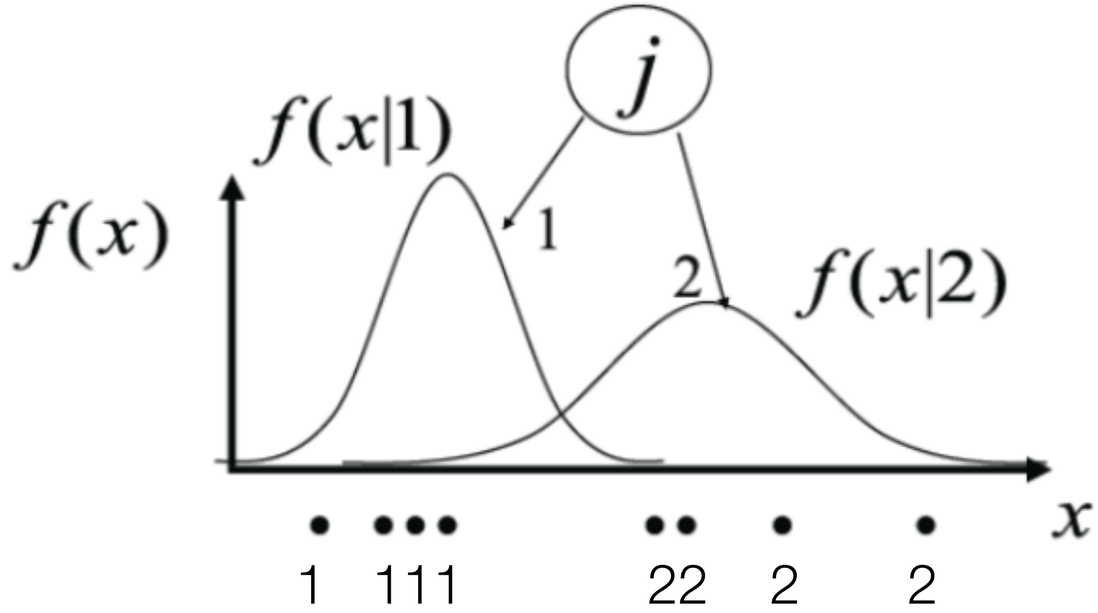
$$\frac{\partial E}{\partial \boldsymbol{\mu}_j} = f\left(\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_M, \boldsymbol{\mu}_M, \boldsymbol{\Sigma}_M\right)$$

- **Complex gradient function (non-linear mutual dependencies)**
- **Optimization of one Gaussian depends on all other Gaussians**
- **It is possible to apply iterative numerical optimization here, but in the following, we will see a simpler method.**

Slide is taken from B. Leibe, ML, 2017

Mixture of Gaussians - Other Strategy

Other strategy



Observed data:

Unobserved data:

Unobserved = '***hidden variable***': **j|x**

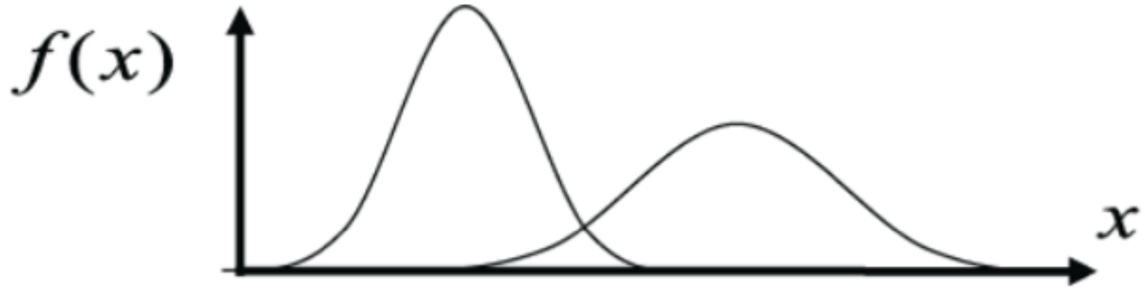
$$h(j=1|x_n) = \textit{1 111} \quad \textit{00} \quad \textit{0} \quad \textit{0}$$

$$h(j=2|x_n) = \textit{0 000} \quad \textit{11} \quad \textit{1} \quad \textit{1}$$

Slide is taken from B. Leibe, ML, 2017

Mixture of Gaussians - Other Strategy

Assume we knew the values of the hidden variable…



ML for Gaussian #1 **ML for Gaussian #2**

assumed known → **1 111 22 2 2 j**

$h(j=1|x_n) =$ ***1 111 00 0 0***

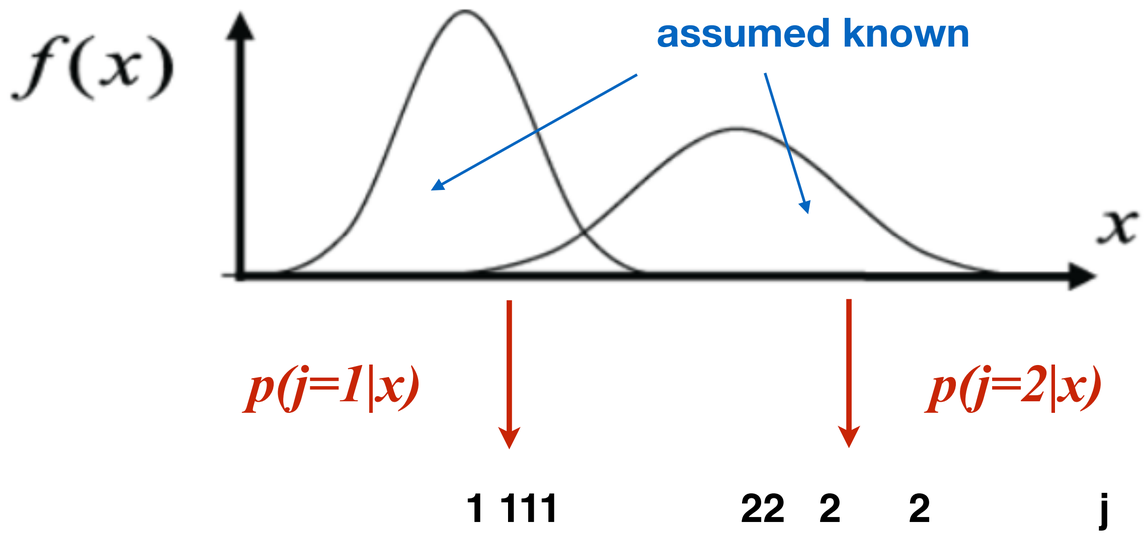
$h(j=2|x_n) =$ ***0 000 11 1 1***

$$\mu_1 = \frac{\sum_{n=1}^{N} h(j=1|x_n)x_n}{\sum_{i=1}^{N} h(j=1|x_n)} \qquad \mu_2 = \frac{\sum_{n=1}^{N} h(j=2|x_n)x_n}{\sum_{i=1}^{N} h(j=2|x_n)}$$

Slide is taken from B. Leibe, ML, 2017

Mixture of Gaussians - Other Strategy

Assume we knew the values of the hidden variable…



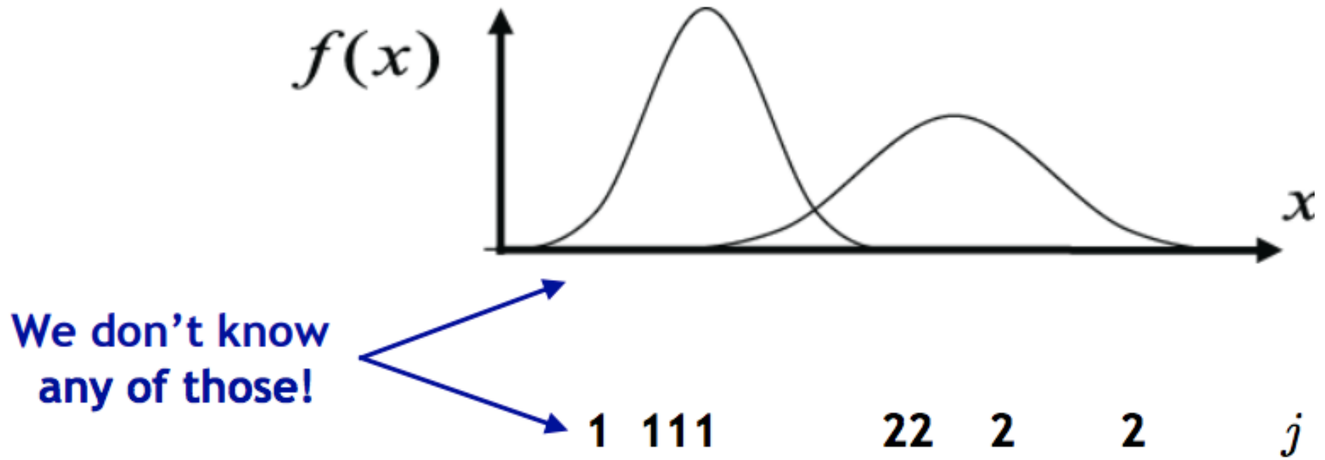
Bayes decision rule:

$$\text{Decide } j = 1 \text{ if } p(j = 1|x_n) > p(j = 2|x_n)$$

Slide is taken from B. Leibe, ML, 2017

Mixture of Gaussians - Other Strategy

Chicken and egg problem - what comes first?



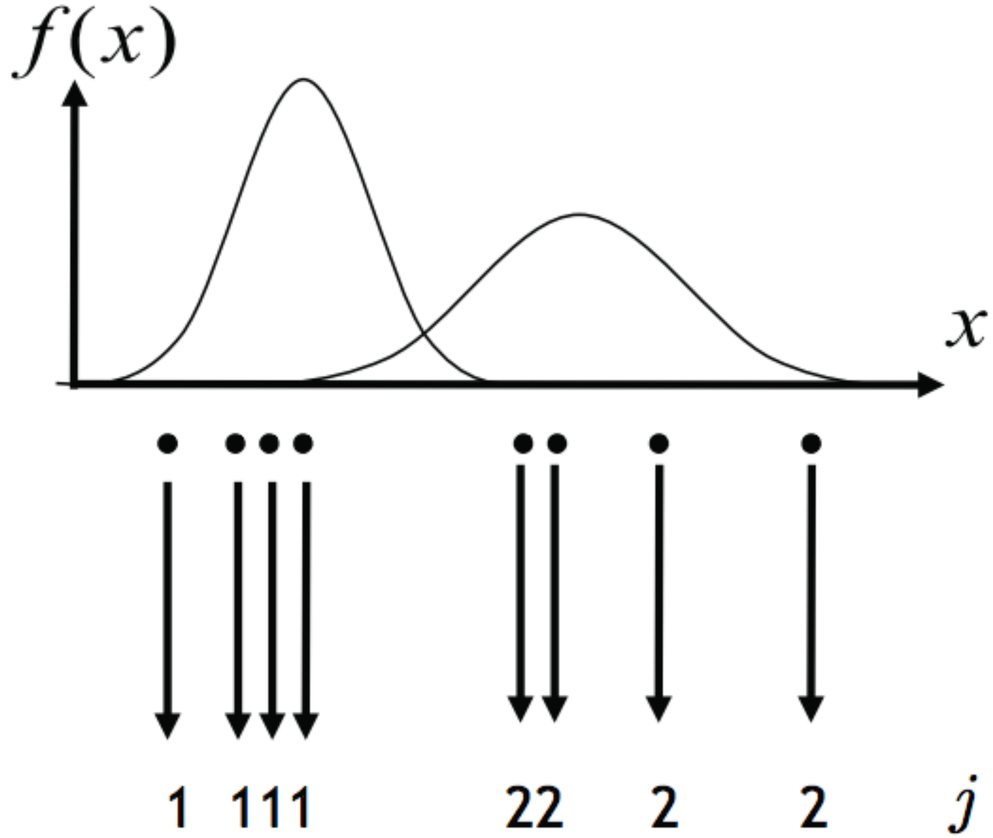
In order to break the loop, we need an estimate for *j*

- For example by Clustering…

Slide is taken from B. Leibe, ML, 2017

Clustering with Hard Assignments

Let us first look at clustering with “hard assignments”



Slide is taken from B. Leibe, ML, 2017

Today’s topics

Mixture distributions

- Mixture of Gaussians (MoG)
- Maximum Likelihood estimation attempt

K-Means Clustering

- Algorithm
- Applications

EM Algorithm

- Credit assignment problem
- MoG estimation
- EM Algorithm
- Interpretation of K-Means
- Technical advice

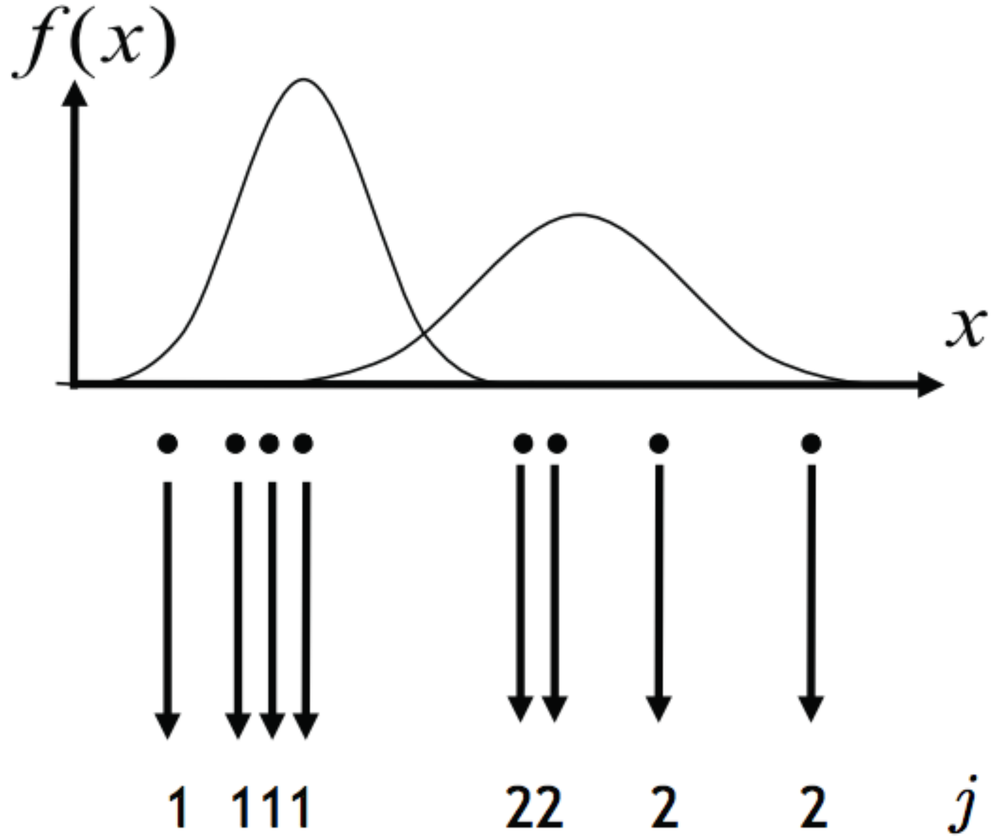
Applications

Part 2, Video MoGandEM_p2

- K-Means Clustering
- Example Application
- Summary

Clustering with Hard Assignments

Let us first look at clustering with “hard assignments”



Slide is taken from B. Leibe, ML, 2017

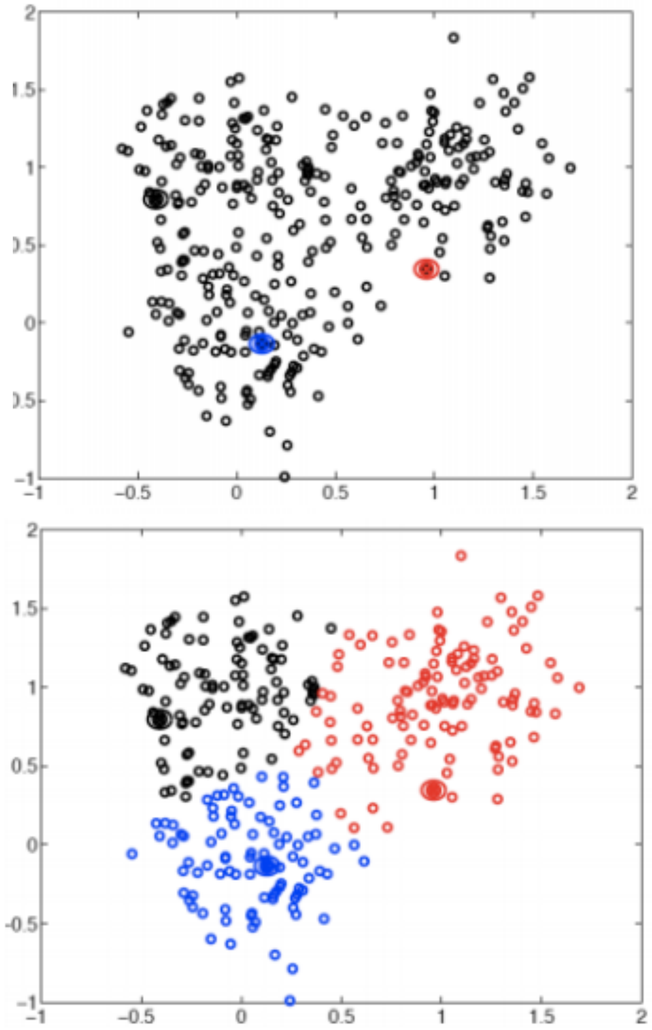
K-Means Clustering

Iterative procedure

1. Initialization: pick K arbitrary centroids (cluster means)
2. Assign each sample to the closest centroid.
3. Adjust the centroids to be the means of the samples assigned to them.
4. Go to step 2 (until no change)

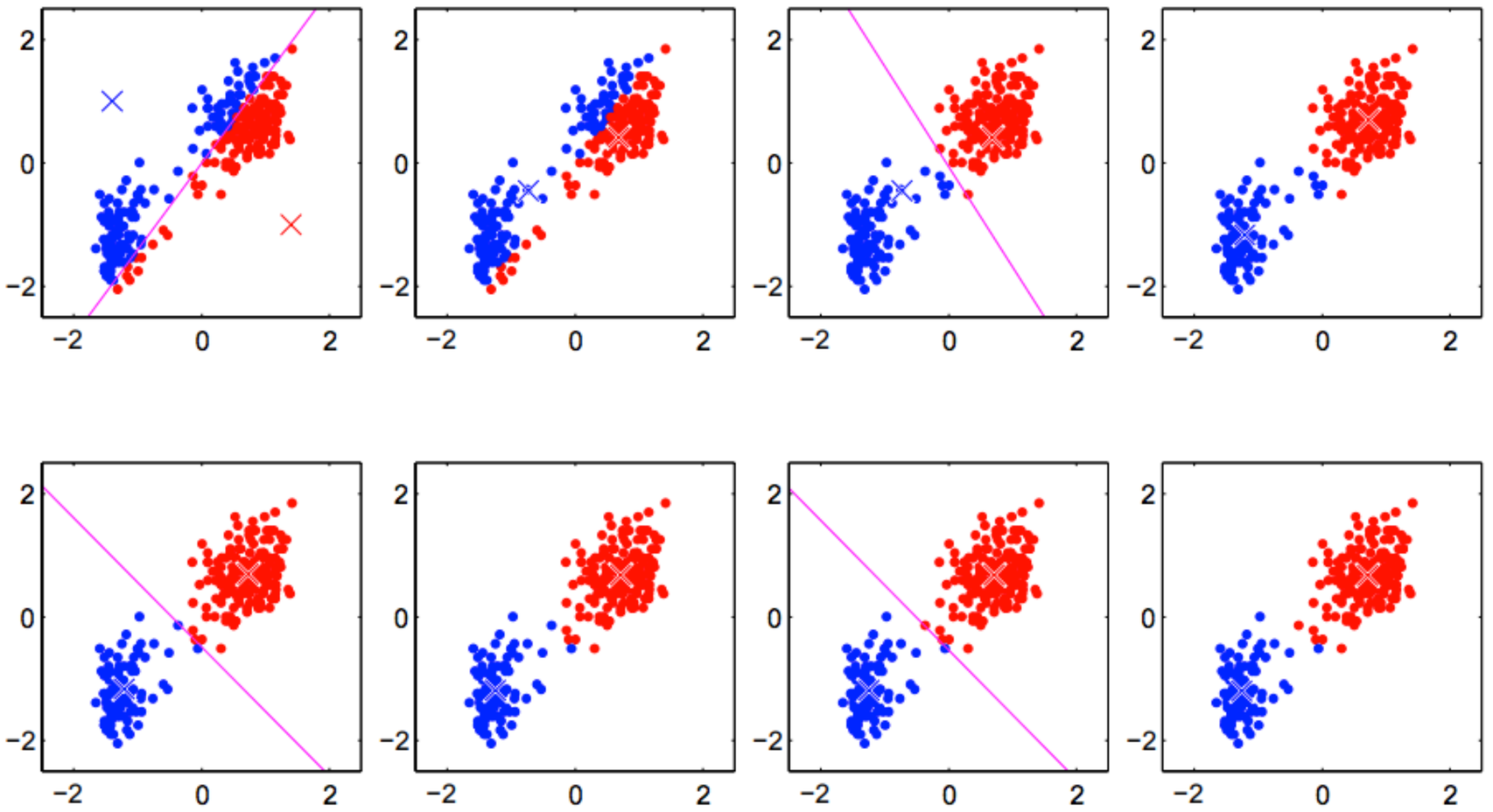
Algorithm is guaranteed to converge

- Local optimum
- Final result depends on initial estimate



Slide is taken from B. Leibe, ML, 2017

K-Means - Example with K=2

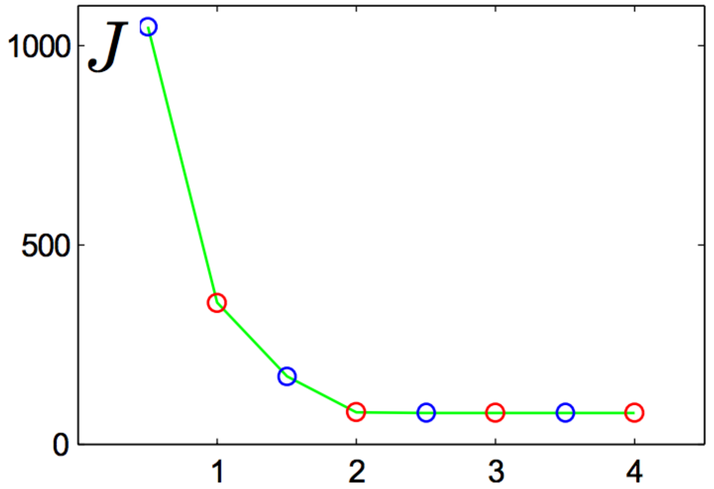


Slide is taken from B. Leibe, ML, 2017

K-Means Clustering

K-Means optimizes the following objective function:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2$$



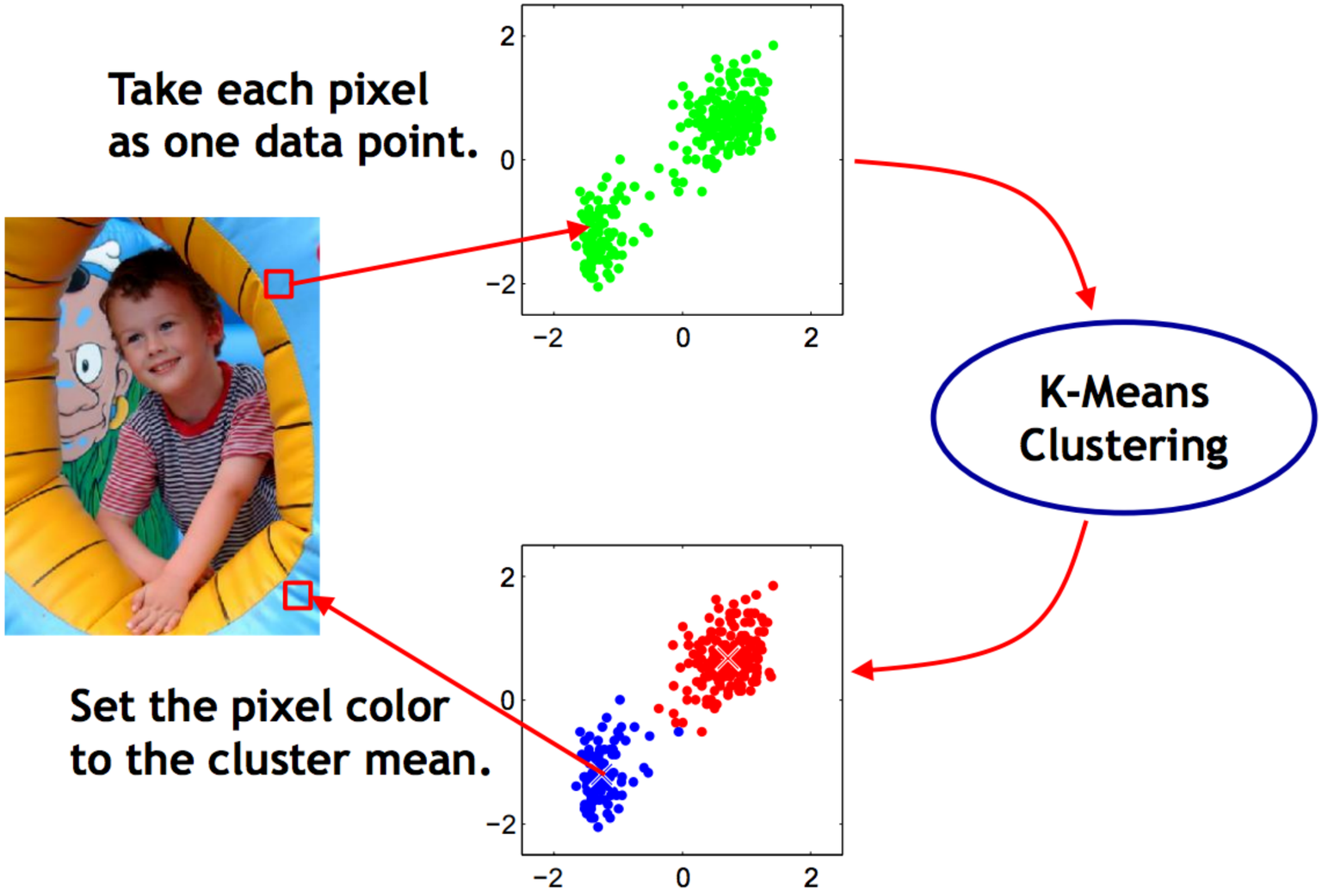
where

$$r_{nk} = \begin{cases} 1 & \text{if } k = \text{argmin}_j ||x_n - \mu_j||^2 \\ 0 & \text{otherwise .} \end{cases}$$

In practice, this procedure usually converges quickly to a local optimum.

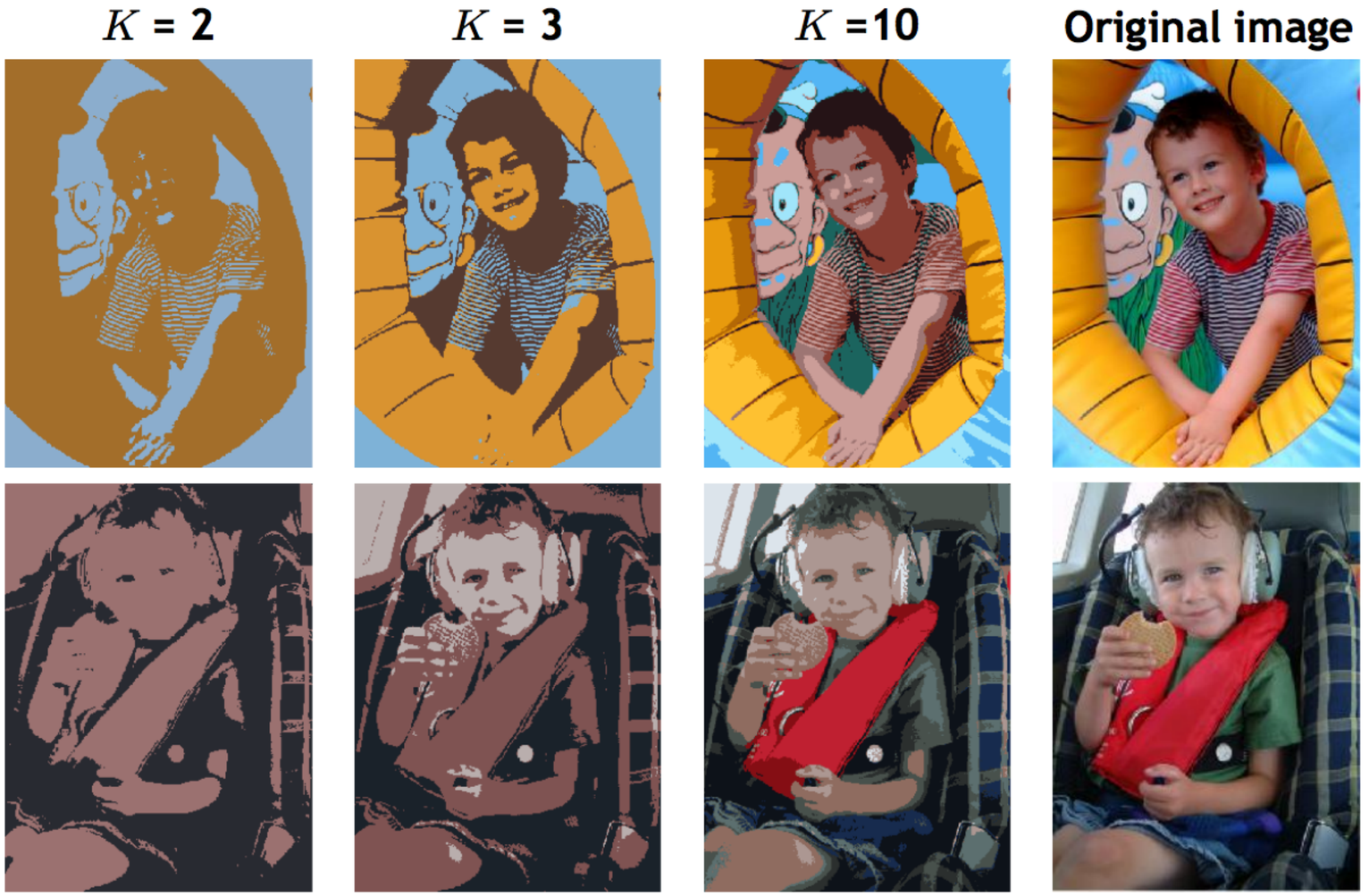
Slide is taken from B. Leibe, ML, 2017

Example Application: Image Compression



Slide is taken from B. Leibe, ML, 2017

Example Application: Image Compression



Slide is taken from B. Leibe, ML, 2017

Summary K-Means

Pros

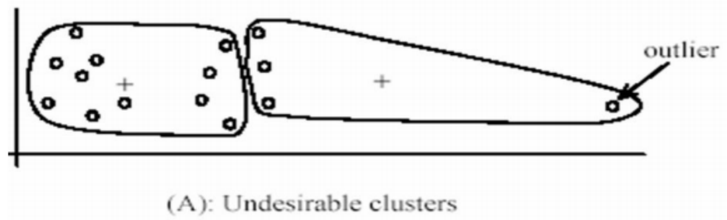
- Simple, fast to compute
- Converges to local minimum of within-cluster squared error

Problem cases

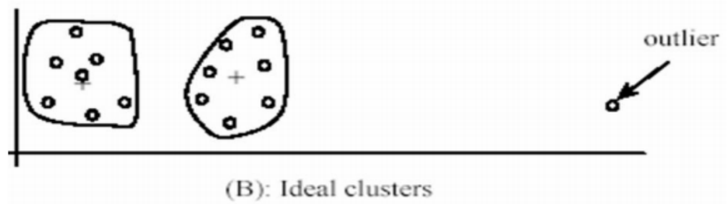
- Setting k?
- Sensitive to initial centres
- Sensitive to outliers
- Detects spherical clusters only

Extensions

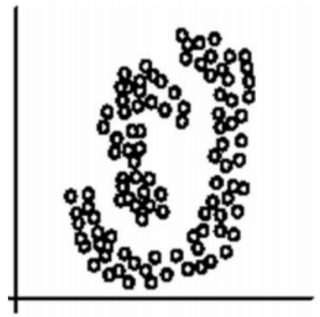
- Speed-ups possible through efficient search structures
- General distance measures: k-medoids

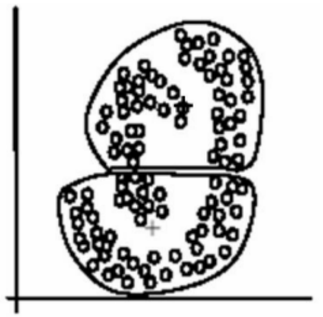


(A): Undesirable clusters



(B): Ideal clusters


(A): Two natural clusters


(B): *k*-means clusters

Slide is taken from B. Leibe, ML, 2017

Part 3, Video MoGandEM_p3

- EM Algorithm
- EM Technical Advise
- Summary

Today’s topics

Mixture distributions

- Mixture of Gaussians (MoG)
- Maximum Likelihood estimation attempt

K-Means Clustering

- Algorithm
- Applications

EM Algorithm

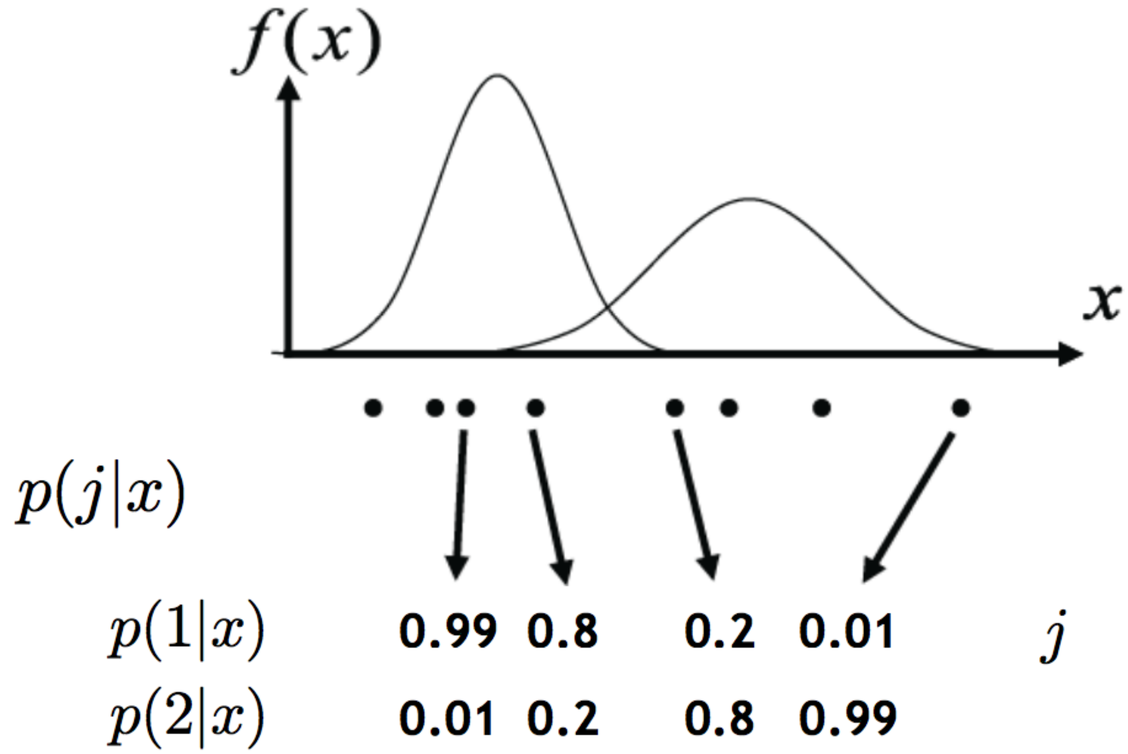
- **Credit assignment problem**
- **MoG estimation**
- **EM Algorithm**
- **Interpretation of K-Means**
- **Technical advice**

Applications

EM Clustering

Clustering with "soft assignment"

- **Expectation step** of the EM algorithm

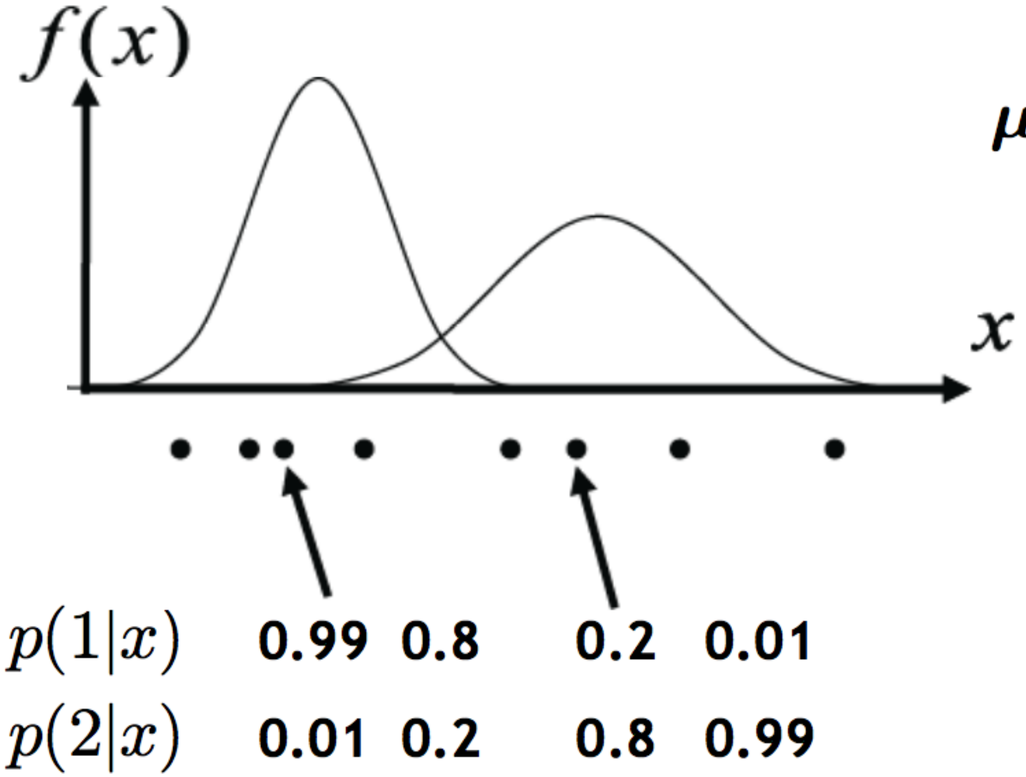


Slide is taken from B. Leibe, ML, 2017

EM Clustering

Clustering with "soft assignment"

- **Maximization step** of the EM algorithm



$$\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} p(j|\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} p(j|\mathbf{x}_n)}$$

Maximum Likelihood estimate

Slide is taken from B. Leibe, ML, 2017

EM Algorithm

Expectation-Maximization (EM) Algorithm

- **E-step:** softly assign samples to mixture components

$$\gamma_j(\mathbf{x}_n) \leftarrow \frac{\pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \qquad \forall j = 1, \ldots, K, \quad n = 1, \ldots, N$$

- **M-step:** re-estimate the parameters (separately for each mixture component) based on the soft assignments

$$\hat{N}_j \leftarrow \sum_{n=1}^{N} \gamma_j(\mathrm{x}_n) = \text{soft number of samples labeled } j$$

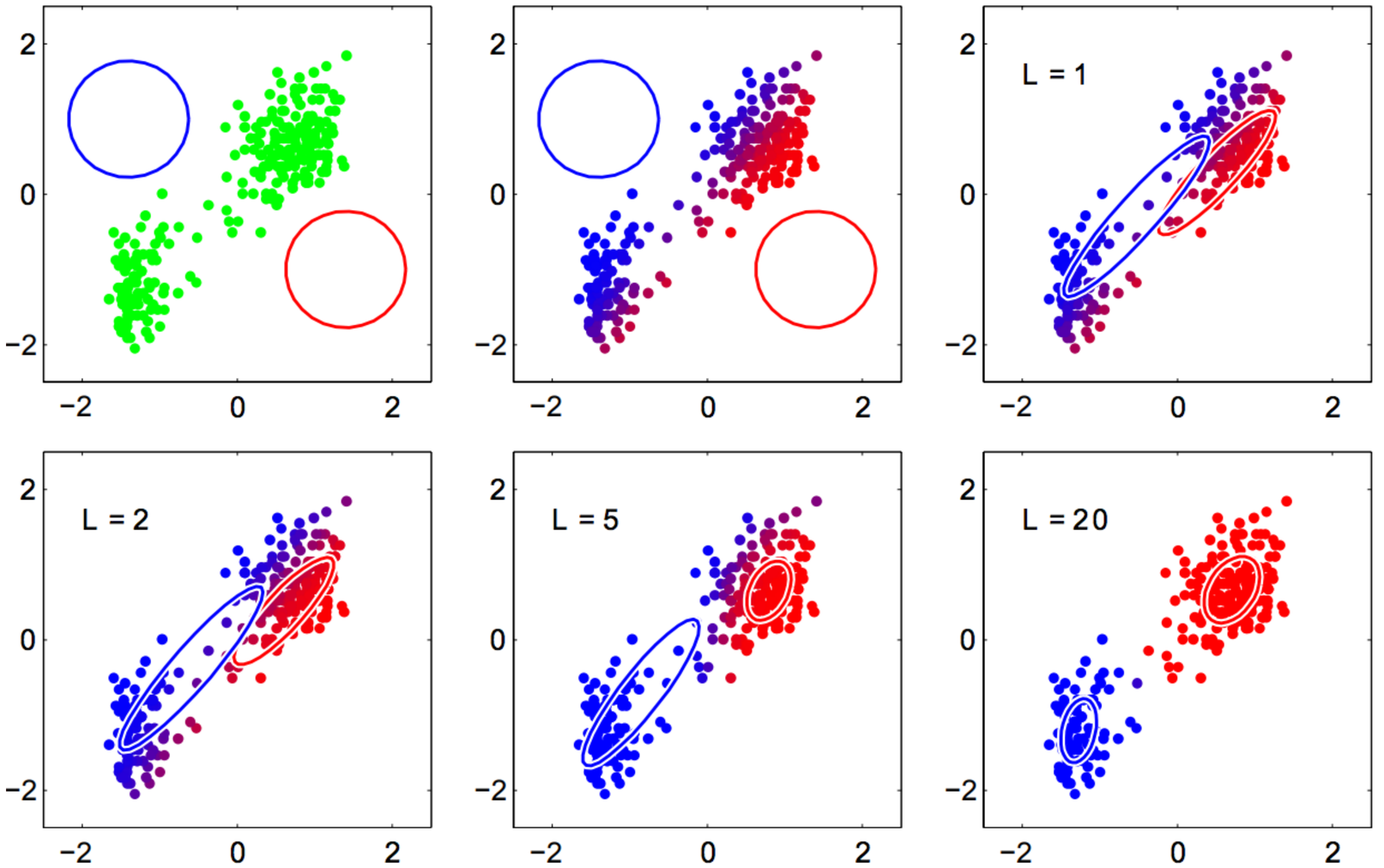
$$\hat{\pi}_j^{new} \leftarrow \frac{\hat{N}_j}{N}$$

$$\hat{\mu}_j^{new} \leftarrow \frac{1}{\hat{N}_j} \sum_{n=1}^{N} \gamma_j(x_n) x_n$$

$$\hat{\Sigma}_j^{new} \leftarrow \frac{1}{\hat{N}_j} \sum_{n=1}^{N} \gamma_j(x_n)(x_n - \hat{\mu}_j^{new})(x_n - \hat{\mu}_j^{new})^T$$

Slide is taken from B. Leibe, ML, 2017

EM Algorithm - An Example



Slide is taken from B. Leibe, ML, 2017

EM - Technical Advice

When implementing EM, we need to take care to avoid singularities in the estimation!

- Mixture components may collapse on single data points.
- E.g. consider the case $\Sigma_k = \sigma_k^2 I$ (which also holds in general)
- Assume component j is exactly centered on data point x_n. This data point will then contribute a term in the likelihood function

$$\mathcal{N}(x_n|x_n, \sigma_j^2 I) = \frac{1}{\sqrt{2\pi}\sigma_j}$$

$$\text{for } \sigma_j \to 0, \text{ this term goes to infinity!}$$

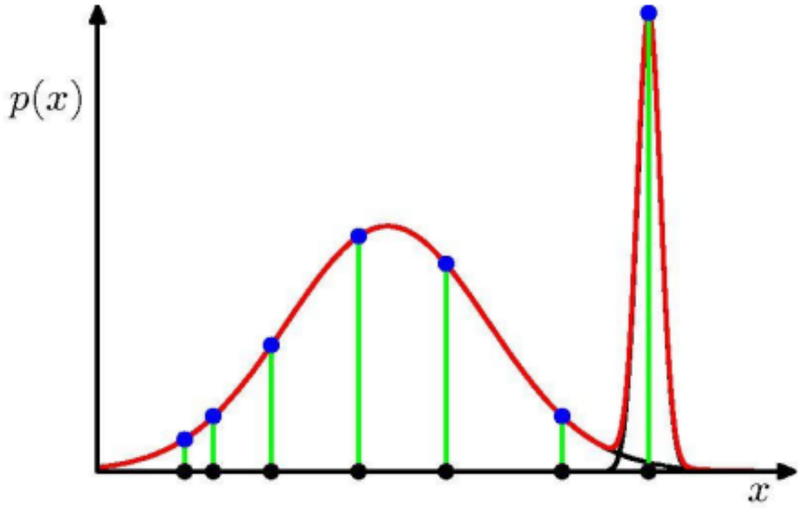
Slide is taken from B. Leibe, ML, 2017

EM - Technical Advice

When implementing EM, we need to take care to avoid singularities in the estimation!

$$\mathcal{N}(x_n|x_n, \sigma_j^2 I) = \frac{1}{\sqrt{2\pi}\sigma_j}$$

for $\sigma_j \to 0$, this term goes to infinity!



Need to introduce regularisation

- Enforce minimum width for the Gaussians

E.g., instead of $\sum^{-1}$ use $(\sum + \sigma_{min} I)^{-1}$

Slide is taken from B. Leibe, ML, 2017

EM - Technical Advice (2)

EM is very sensitive to the initialisation

- Will converge to a local optimum of E.
- Convergence is relatively slow

=> Initialize with k-Means to get better results!

- k-Means is itself initialised randomly, will also only find a local optimum.
- But convergence is much faster.

Slide is taken from B. Leibe, ML, 2017

EM - Technical Advice (2)

Typical procedure

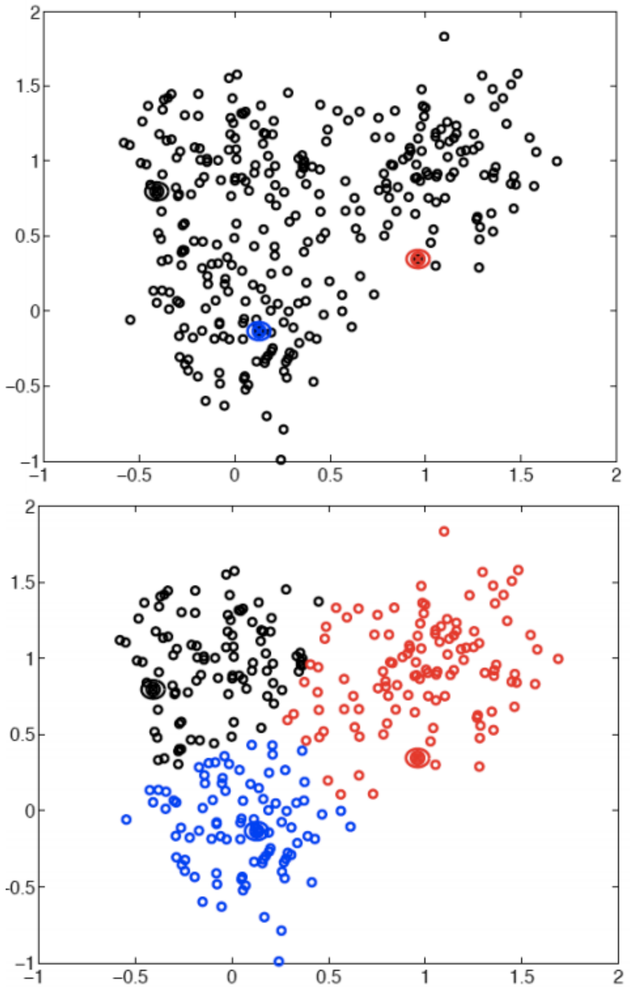
- Run k-Means M times (e.g. M=10-100).
- Pick the best result (lowest error J).
- Use this result to initialize EM
  - Set μ_j to the corresponding cluster mean from k-Means.
  - Initialize $\sum_j$ to the sample covariance of the associated data points.

Slide is taken from B. Leibe, ML, 2017

k-Means Clustering Revisited

Interpreting the procedure

1. Initialization: pick K arbitrary centroids (cluster means).
2. Assign each sample to the closest centroid. (**E-Step**)
3. Adjust the centroids to be the means of the samples assigned to them. (**M-Step**)
4. Go to step 2 (until no change)

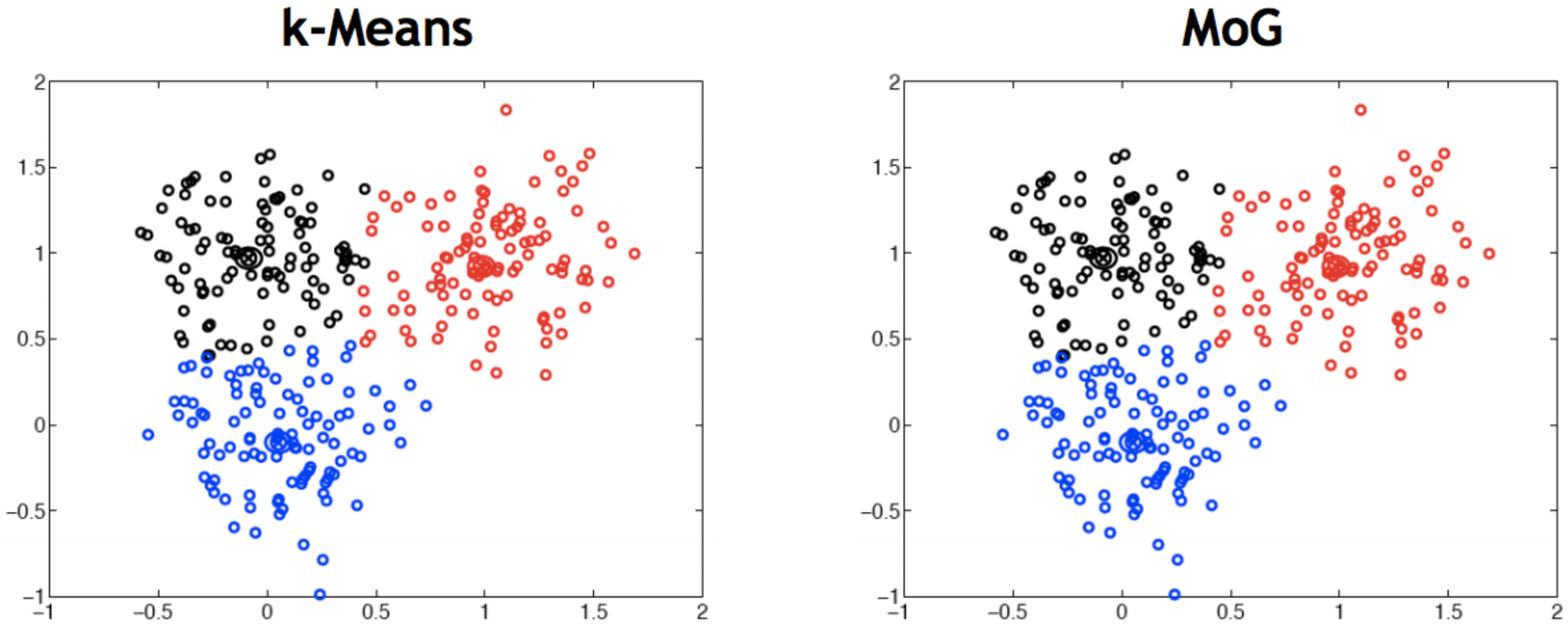


Slide is taken from B. Leibe, ML, 2017

k-Means Clustering Revisited

K-Means clustering essentially corresponds to a Gaussian Mixture Model (MoG or GMM) estimation with EM whenever

- The covariances are of the K Gaussians are set to $\Sigma_j = \sigma^2 I$
- For some small, fixed σ^2



Slide is taken from B. Leibe, ML, 2017

Summary: Gaussian Mixture Models

Properties

- Very general, can represent any (continuous) distribution
- Once trained, very fast to evaluate.
- Can be updated online

Problems/Caveats

- Some numerical issues in the implementation
  - => Need to apply regularisation in order to avoid singularities.
- EM for MoG is computationally expensive
  - Especially for high-dimensional problems!
  - More computational overhead and slower convergence than k_Means
  - Results very sensitive to initialisation
  - => run k-means for some iterations as initialisation.
- Need to select the number of mixture components K.

Slide is taken from B. Leibe, ML, 2017

Part 4, Video MoGandEM_p4

- Background Model for Tracking
- Image Segmentation
- Color-Based Skin Detection

Today’s topics

Mixture distributions

- Mixture of Gaussians (MoG)
- Maximum Likelihood estimation attempt

K-Means Clustering

- Algorithm
- Applications

EM Algorithm

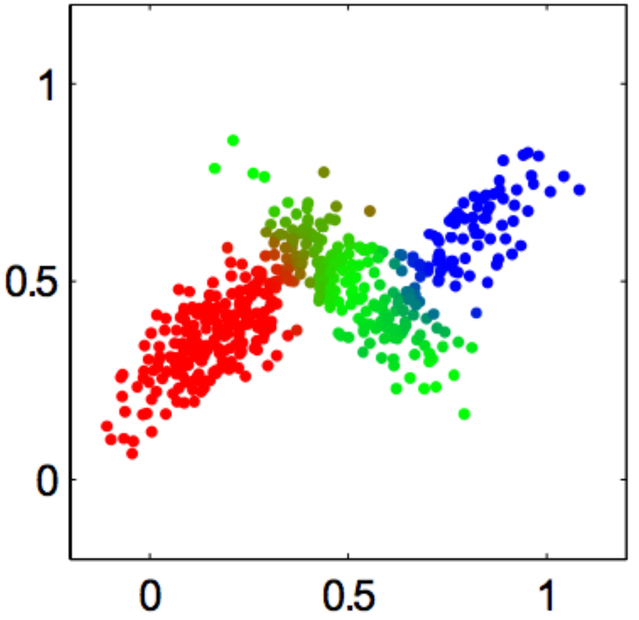
- Credit assignment problem
- MoG estimation
- EM Algorithm
- Interpretation of K-Means
- Technical advice

Applications

Applications

Mixture models are used in many practical applications.

- Where distributions with complex or unknown shapes need to be represented…



Applications in Computer Vision

- Model distributions of pixel colours
- Each pixel is one data point in, e.g., RGB space

  => Learn a MoG to represent the class-conditional densities

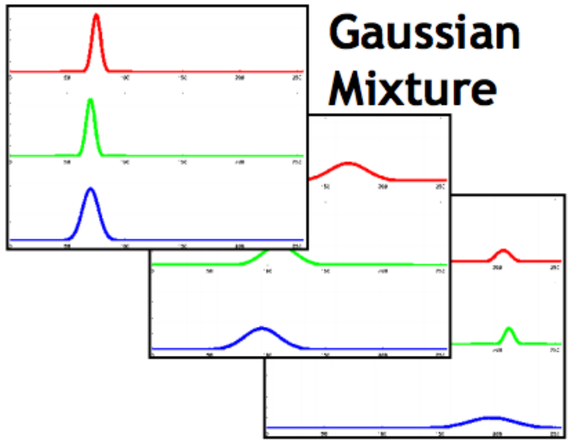
  => Use the learned models to classify other pixels

Slide is taken from B. Leibe, ML, 2017

Application: Background Model for Tracking

Train background MoG for each pixel

- Model “common“ appearance variation for each background pixel.
- Initialization with an empty scene.
- Update the mixtures over time
  - Adapt to lighting changes, etc.



Used in many vision-based tracking applications

- Anything that cannot be explained by the background model is labeled as foreground (=object).
- Easy segmentation if camera is fixed.

C. Stauffer, E. Grimson, Learning Patterns of Activity Using Real-Time Tracking, IEEE Trans. PAMI, 22(8):747-757, 2000.

Slide is taken from B. Leibe, ML, 2017

Application: Image Segmentation

(a) input image (b) user input (c) inferred segmentation

Used in many vision-based tracking applications

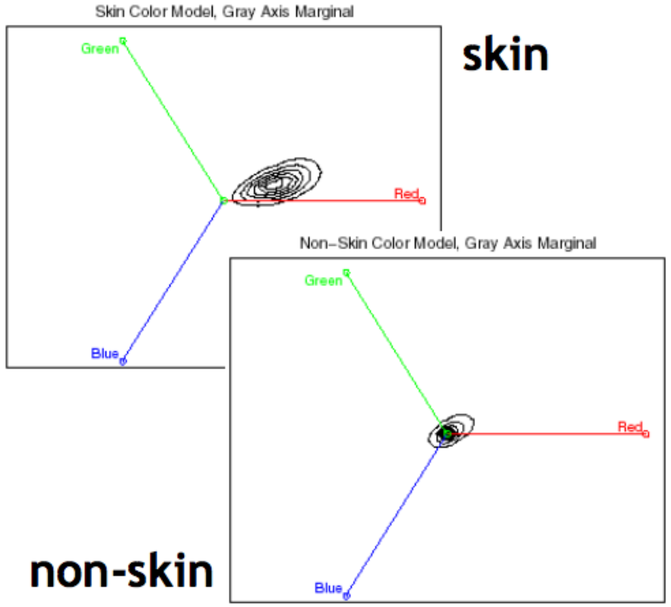
- User marks two regions for foreground and background.
- Learn a MoG model for the color values in each region.
- Use those models to classify all other pixels.

=>Simple segmentation procedure (building block for more complex applications)

Slide is taken from B. Leibe, ML, 2017

Application: Color-Based Skin Detection

- **Collect training samples for skin/non-skin pixels.**
- **Estimate MoG to represent the skin/non-skin densities**



Classify skin color pixels in novel images

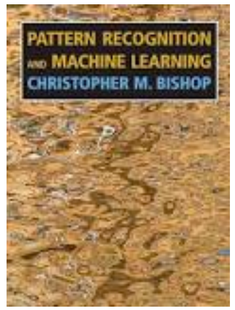
M. Jones and J. Rehg, Statistical Color Models with Application to Skin Detection, IJCV 2002.

Slide is taken from B. Leibe, ML, 2017

Readings

Bishop's book

Chapter 2.3.9 and the entire Chapter 9



Additional information

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

J.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

Slide is taken from B. Leibe, ML, 2017