Machine Learning – Lecture 4

Probability Density Estimation III

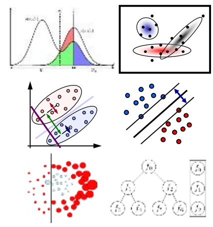
17.10.2019

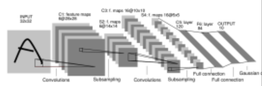
Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de

leibe@vision.rwth-aachen.de

Course Outline

- Fundamentals
  - Bayes Decision Theory
  - Probability Density Estimation
- Classification Approaches
  - Linear Discriminants
  - Support Vector Machines
  - Ensemble Methods & Boosting
  - Randomized Trees, Forests & Ferns
- Deep Learning
  - Foundations
  - Convolutional Neural Networks
  - Recurrent Neural Networks





Recap: Maximum Likelihood Approach

- Computation of the likelihood
  - Single data point: $p(x_n|\theta)$
  - Assumption: all data points $X = \{x_1, \dots, x_n\}$ are independent

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

  - Log-likelihood

$$E(\theta) = -\ln L(\theta) = -\sum_{n=1}^{N} \ln p(x_n|\theta)$$

- Estimation of the parameters θ (Learning)
  - Maximize the likelihood (=minimize the negative log-likelihood)

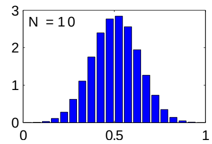
  $\Rightarrow$ Take the derivative and set it to zero.

$$\frac{\partial}{\partial\theta} E(\theta) = -\sum_{n=1}^{N} \frac{\frac{\partial}{\partial\theta} p(x_n|\theta)}{p(x_n|\theta)} \stackrel{!}{=} 0$$

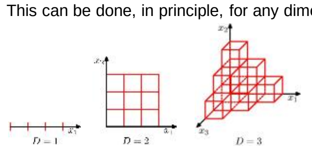
Recap: Histograms

- Basic idea:
  - Partition the data space into distinct bins with widths 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!

Recap: Kernel Density Estimation

see Exercise 1.5

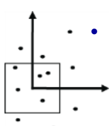
- Approximation formula:

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

fixed V, determine K → Kernel Methods

fixed K, determine V → K-Nearest Neighbor

- Kernel methods
  - Place a *kernel window* k at location $\mathbf{x}$ and count how many data points fall inside it.



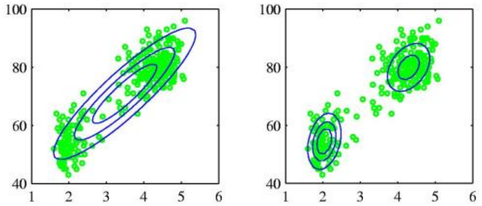
- K-Nearest Neighbor
  - Increase the volume V until the K nearest data points are found.

Topics of This Lecture

- 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

Mixture Distributions

- A single parametric distribution is often not sufficient
  - E.g. for multimodal data

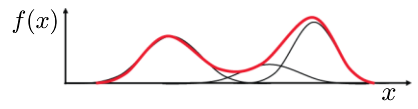


Single Gaussian

Mixture of two Gaussians

Mixture of Gaussians (MoG)

- Sum of M individual Normal distributions



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

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

Mixture of Gaussians

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

Likelihood of measurement x given mixture component j

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

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

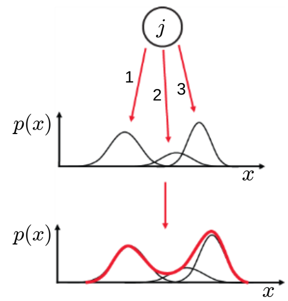
Prior of component j

- Notes
  - The mixture density integrates to 1: $\int p(x)dx = 1$
  - The mixture parameters are

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

Mixture of Gaussians (MoG)

- "Generative model"



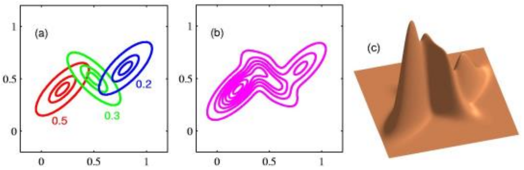
$p(j) = \pi_j$ "Weight" of mixture component

$p(x|\theta_j)$ Mixture component

Mixture density

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

Mixture of Multivariate Gaussians



Mixture of Multivariate Gaussians

- Multivariate Gaussians

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

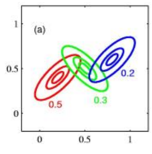
$$p(\mathbf{x}|\theta_j) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}_j|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^{\mathrm{T}}\mathbf{\Sigma}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)\right\}$$

  - Mixture weights / mixture coefficients:

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

  - Parameters:

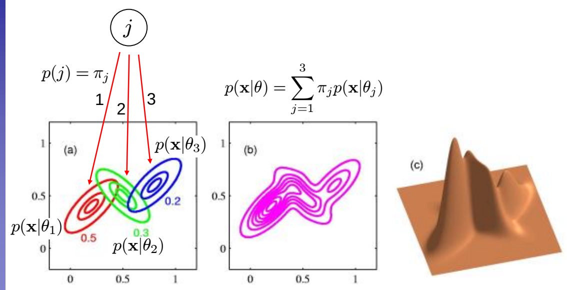
$$\theta = (\pi_1, \boldsymbol{\mu}_1, \mathbf{\Sigma}_1, \ldots, \pi_M, \boldsymbol{\mu}_M, \mathbf{\Sigma}_M)$$



Mixture of Multivariate Gaussians

- "Generative model"

$$p(\mathbf{x}|\theta) = \sum_{j=1}^{3} \pi_j p(\mathbf{x}|\theta_j)$$



Mixture of Gaussians – 1st Estimation Attempt

- Maximum Likelihood
  - Minimize $E = -\ln L(\theta) = -\sum_{n=1}^{N} \ln p(\mathbf{x}_n|\theta)$
  - Let's first look at $\boldsymbol{\mu}_j$:

$$\frac{\partial E}{\partial \boldsymbol{\mu}_j} = 0$$

  - We can already see that this will be difficult, since

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

This will cause problems!

Mixture of Gaussians – 1st Estimation Attempt

- Minimization:

$$\boxed{\frac{\partial}{\partial \boldsymbol{\mu}_j} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_j)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

$$\begin{aligned}
\frac{\partial E}{\partial \boldsymbol{\mu}_j} &= -\sum_{n=1}^{N} \frac{\frac{\partial}{\partial \boldsymbol{\mu}_j} p(\mathbf{x}_n|\theta_j)}{\sum_{k=1}^{K} p(\mathbf{x}_n|\theta_k)} \\
&= -\sum_{n=1}^{N} \left(\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_j) \frac{p(\mathbf{x}_n|\theta_j)}{\sum_{k=1}^{K} p(\mathbf{x}_n|\theta_k)} \right) \\
&= -\cancel{\boldsymbol{\Sigma}^{-1}} \sum_{n=1}^{N} (\mathbf{x}_n - \boldsymbol{\mu}_j) \underbrace{\frac{\pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}}_{= \gamma_j(\mathbf{x}_n)} \overset{!}{=} 0
\end{aligned}$$

"responsibility" of component j for $\mathbf{x}_n$

- We thus obtain

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

Mixture of Gaussians – 1st Estimation Attempt

- But…

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

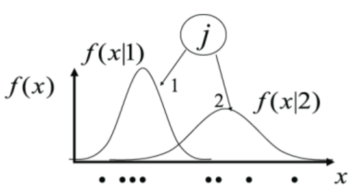
- I.e. there is no direct analytical solution!

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

  - 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.

Mixture of Gaussians – Other Strategy

- Other strategy:



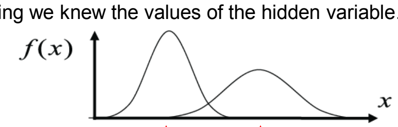
  - Observed data:
  - Unobserved data: 1 111 22 2 2
    - Unobserved = "hidden variable": j|x

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

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

Mixture of Gaussians – Other Strategy

- Assuming we knew the values of the hidden variable…



ML for Gaussian #1 ML for Gaussian #2

assumed known → 1 111 22 2 2 j

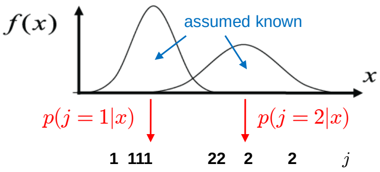
$$h(j=1|x_n) = \quad 1\ 111 \quad 00\ \ 0 \quad 0$$

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

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

Mixture of Gaussians – Other Strategy

- Assuming we knew the mixture components…

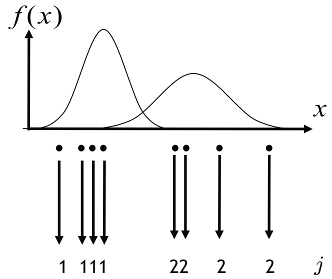


- Bayes decision rule: Decide $j = 1$ if

$$p(j=1|x_n) > p(j=2|x_n)$$

Clustering with Hard Assignments

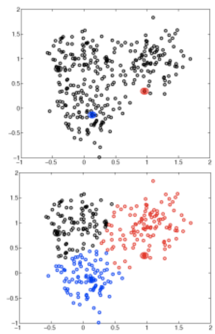
- Let's first look at clustering with "hard assignments"



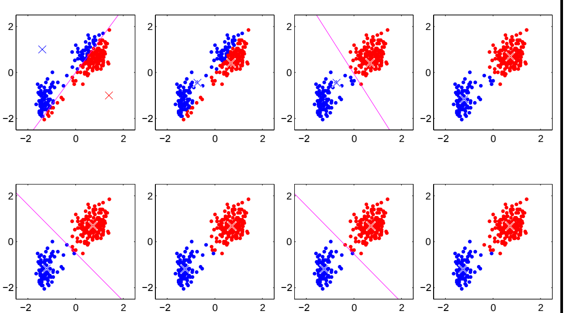
Topics of This Lecture

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 after finite #iterations.
  - Local optimum
  - Final result depends on initialization.



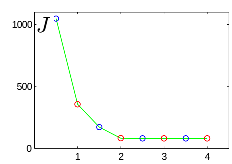
K-Means – Example with K=2



K-Means Clustering

- K-Means optimizes the following objective function:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{x}_n - \boldsymbol{\mu}_k||^2$$



  - where

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j ||\mathbf{x}_n - \boldsymbol{\mu}_j||^2 \\ 0 & \text{otherwise.} \end{cases}$$

  - I.e., r_{nk} is an indicator variable that checks whether $\boldsymbol{\mu}_k$ is the nearest cluster center to point $\mathbf{x}_n$.
  - In practice, this procedure usually converges quickly to a local optimum.

Example Application: Image Compression

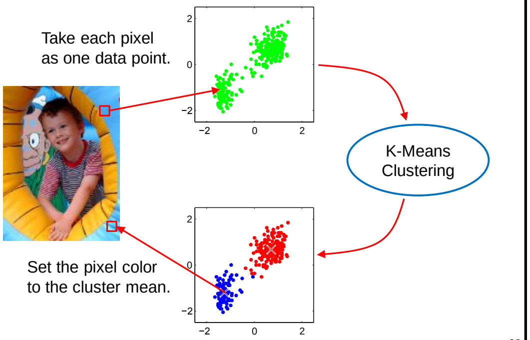


Image source: C.M. Bishop, 2006

Example Application: Image Compression

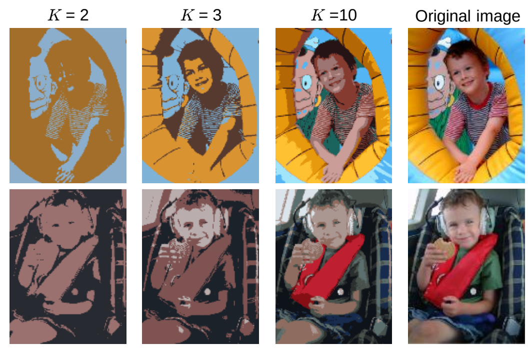
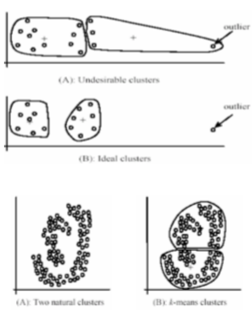


Image source: C.M. Bishop, 2006

Summary K-Means

- Pros
  - Simple, fast to compute
  - Converges to local minimum of within-cluster squared error
- Problem cases
  - Setting k?
  - Sensitive to initial centers
  - Sensitive to outliers
  - Detects spherical clusters only
- Extensions
  - Speed-ups possible through efficient search structures
  - General distance measures: k-medoids



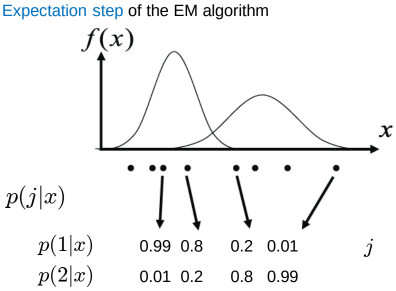
Slide credit: Kristen Grauman

Topics of This Lecture

- Mixture distributions
  - Recap: 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

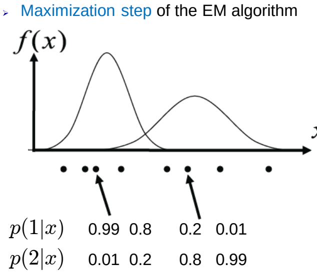
- Clustering with "soft assignments"
  - Expectation step of the EM algorithm



Slide credit: Bernt Schiele

EM Clustering

- Clustering with "soft assignments"
  - 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 credit: Bernt Schiele

Credit Assignment Problem

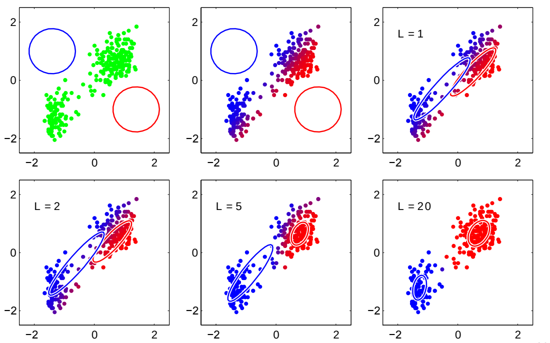
- "Credit Assignment Problem"
  - If we are just given $\mathbf{x}$, we don't know which mixture component this example came from
$$p(\mathbf{x}|\theta) \;=\; \sum_{j=1}^{2} \pi_j p(\mathbf{x}|\theta_j)$$
  - We can however evaluate the posterior probability that an observed $\mathbf{x}$ was generated from the first mixture component.
$$p(j=1|\mathbf{x},\theta) = \frac{p(j=1,\mathbf{x}|\theta)}{p(\mathbf{x}|\theta)}$$
$$p(j=1,\mathbf{x}|\theta) = p(\mathbf{x}|j=1,\theta)p(j=1) = p(\mathbf{x}|\theta_1)p(j=1)$$
$$p(j=1|\mathbf{x},\theta) = \frac{p(\mathbf{x}|\theta_1)p(j=1)}{\sum_{j=1}^{2} p(\mathbf{x}|\theta_j)p(j)} = \gamma_j(\mathbf{x})$$

"responsibility" of component j for $\mathbf{x}$.

EM Algorithm

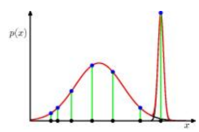
- Expectation-Maximization (EM) Algorithm
  - E-Step: softly assign samples to mixture components
$$\gamma_j(\mathbf{x}_n) \leftarrow \frac{\pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)} \qquad \forall j=1,\ldots,K, \quad n=1,\ldots,N$$
  - M-Step: re-estimate the parameters (separately for each mixture component) based on the soft assignments
$$\hat{N}_j \leftarrow \sum_{n=1}^{N} \gamma_j(\mathbf{x}_n) = \text{soft number of samples labeled } j$$
$$\hat{\pi}_j^{\text{new}} \leftarrow \frac{\hat{N}_j}{N}$$
$$\hat{\boldsymbol{\mu}}_j^{\text{new}} \leftarrow \frac{1}{\hat{N}_j} \sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)\mathbf{x}_n$$
$$\hat{\boldsymbol{\Sigma}}_j^{\text{new}} \leftarrow \frac{1}{\hat{N}_j} \sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_j^{\text{new}})(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_j^{\text{new}})^{\mathrm{T}}$$

EM Algorithm – An Example



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 $\boldsymbol{\Sigma}_k = \sigma_k^2 \mathbf{I}$ (this also holds in general)
  - Assume component j is exactly centered on data point $\mathbf{x}_n$. This data point will then contribute a term in the likelihood function
$$\mathcal{N}(\mathbf{x}_n|\mathbf{x}_n, \sigma_j^2\mathbf{I}) = \frac{1}{\sqrt{2\pi}\sigma_j}$$
  - For $\sigma_j \to 0$, this term goes to infinity!



$\Rightarrow$ Need to introduce regularization
  - Enforce minimum width for the Gaussians
  - E.g., instead of $\boldsymbol{\Sigma}^{-1}$, use $(\boldsymbol{\Sigma} + \sigma_{\min}\mathbf{I})^{-1}$

EM – Technical Advice (2)

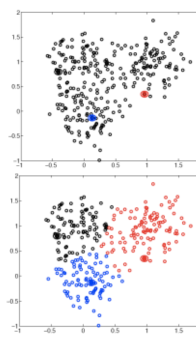
- EM is very sensitive to the initialization
  - Will converge to a local optimum of E.
  - Convergence is relatively slow.

$\Rightarrow$ Initialize with k-Means to get better results!
  - k-Means is itself initialized randomly, will also only find a local optimum.
  - But convergence is much faster.

- 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 Σ_j to the sample covariance of the associated data points.

K-Means Clustering Revisited

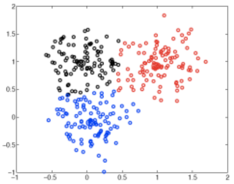
- Interpreting the procedure
  1. Initialization: pick K arbitrary centroids (cluster means)
  2. Assign each sample to the closest centroid. (E-Step)
  3. Adjust the centroids to be the means of the samples assigned to them. (M-Step)
  4. Go to step 2 (until no change)



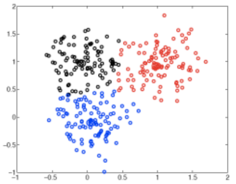
K-Means Clustering Revisited

- 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

k-Means



MoG



Summary: Gaussian Mixture Models

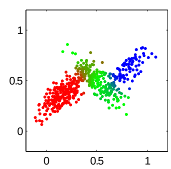
- Properties
  - Very general, can represent any (continuous) distribution.
  - Once trained, very fast to evaluate.
  - Can be updated online.
- Problems / Caveats
  - Some numerical issues in the implementation
    - ⇒ Need to apply regularization in order to avoid singularities.
  - EM for MoG is computationally expensive
    - Especially for high-dimensional problems!
    - More computational overhead and slower convergence than k-Means
    - Results very sensitive to initialization
    - ⇒ Run k-Means for some iterations as initialization!
  - Need to select the number of mixture components K.
    - ⇒ Model selection problem (see later lecture)

Topics of This Lecture

- Mixture distributions
  - Recap: 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

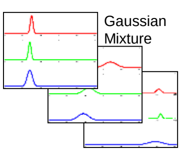
- Mixture models are used in many practical applications.
  - Wherever distributions with complex or unknown shapes need to be represented…



- Popular application in Computer Vision
  - Model distributions of pixel colors.
  - 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.

Application: Background Model for Tracking

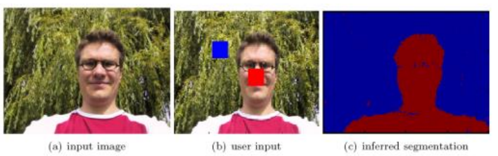
- Train background MoG for each pixel
  - Model "common" appearance variation for each background pixel.
  - Initialization with an empty scene.
  - Update the mixtures over time
    - Adapt to lighting changes, etc.



- 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.

Application: Image Segmentation



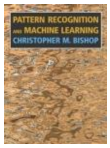
(a) input image (b) user input (c) inferred segmentation

- User assisted image segmentation
  - 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)

References and Further Reading

- More information about EM and MoG estimation is available in Chapter 2.3.9 and the entire Chapter 9 of Bishop's book (recommendable to read).

  Christopher M. Bishop
  Pattern Recognition and Machine Learning
  Springer, 2006



- Additional information
  - Original EM paper:
    - 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:
    - 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