Machine Learning – Lecture 4

Probability Density Estimation III

22.10.2017

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

• Exam dates
  - According to rwth online, the exam dates are
  - 1\textsuperscript{st} try Sat 02.03.2019 10:30 – 12:00h
  - 2\textsuperscript{nd} try Thu 21.03.2019 13:30 – 15:30h
  - Exam registration will start in early December…
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 \( X = \{x_1, \ldots, x_n\} \) 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 \( \theta \) (Learning)**
  - Maximize the likelihood (=minimize the negative log-likelihood)
    \[
    \frac{\partial}{\partial \theta} E(\theta) = - \sum_{n=1}^{N} \frac{\partial}{\partial \theta} \frac{p(x_n|\theta)}{p(x_n|\theta)} \overset{!}{=} 0
    \]

_Recap: Maximum Likelihood Approach_

- **Computation of the likelihood**
  - Single data point: \( p(x_n|\theta) \)
  - Assumption: all data points \( X = \{x_1, \ldots, x_n\} \) 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 \( \theta \) (Learning)**
  - Maximize the likelihood (=minimize the negative log-likelihood)
    \[
    \frac{\partial}{\partial \theta} E(\theta) = - \sum_{n=1}^{N} \frac{\partial}{\partial \theta} \frac{p(x_n|\theta)}{p(x_n|\theta)} \overset{!}{=} 0
    \]

_Slide credit: Bernt Schiele_
Recap: Histograms

• Basic idea:
  - Partition the data space into distinct bins with widths $\Delta_i$ 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

- Approximation formula:
  \[
  p(x) \approx \frac{K}{NV}
  \]

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

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

Slide adapted from Bernt Schiele
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
Mixture of Gaussians (MoG)

- Sum of $M$ individual Normal distributions

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

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

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

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

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

- **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”

\[ p(x) = \sum_{j=1}^{M} p(x|\theta_j)p(j) \]

"Weight" of mixture component

Mixture component

Mixture density

\[ p(j) = \pi_j \]

Slide credit: Bernt Schiele
Mixture of Multivariate Gaussians

(a) 0.5  0.3  0.2
(b) 0.5  0.3
(c)

Image source: C.M. Bishop, 2006
Mixture of Multivariate Gaussians

- Multivariate Gaussians

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

\[
p(x|\theta_j) = \frac{1}{(2\pi)^{D/2}|\Sigma_j|^{1/2}} \exp \left\{ -\frac{1}{2}(x - \mu_j)^{T}\Sigma_j^{-1}(x - \mu_j) \right\}
\]

- Mixture weights / mixture coefficients:

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

- Parameters:

\[
\theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_M, \mu_M, \Sigma_M)
\]
Mixture of Multivariate Gaussians

- “Generative model”

\[
p(j) = \pi_j
\]

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

Slide credit: Bernt Schiele

Image source: C.M. Bishop, 2006
Mixture of Gaussians – 1st Estimation Attempt

- **Maximum Likelihood**
  
  - Minimize \( E = -\ln L(\theta) = -\sum_{n=1}^{N} \ln p(x_n|\theta) \)
  
  - Let’s first look at \( \mu_j \):
    
    \[
    \frac{\partial E}{\partial \mu_j} = 0
    \]

  - We can already see that this will be difficult, since
    
    \[
    \ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\}
    \]

    This will cause problems!

Slide adapted from Bernt Schiele
Mixture of Gaussians – 1\textsuperscript{st} Estimation Attempt

- Minimization:

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

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

- We thus obtain

\[
\Rightarrow \mu_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n) x_n}{\sum_{n=1}^{N} \gamma_j(x_n)}
\]

\[
\gamma_j(x_n) \text{ “responsibility” of component } j \text{ for } x_n
\]

\[
\frac{\partial}{\partial \mu_j} \mathcal{N}(x_n | \mu_k, \Sigma_k) = \\
\Sigma^{-1}(x_n - \mu_j) \mathcal{N}(x_n | \mu_k, \Sigma_k)
\]
Mixture of Gaussians – 1\textsuperscript{st} Estimation Attempt

• But…

\[ \mu_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n)x_n}{\sum_{n=1}^{N} \gamma_j(x_n)} \quad \gamma_j(x_n) = \frac{\pi_j \mathcal{N}(x_n; \mu_j, \Sigma_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n; \mu_k, \Sigma_k)} \]

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

\[ \frac{\partial E}{\partial \mu_j} = f(\pi_1, \mu_1, \Sigma_1, \ldots, \pi_M, \mu_M, \Sigma_M) \]

- 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: 
    - Unobserved = “hidden variable”: $j|x$

\[
\begin{align*}
h(j = 1 | x_n) &= \begin{bmatrix} 1 & 111 \\ 0 & 0 \end{bmatrix} \\
h(j = 2 | x_n) &= \begin{bmatrix} 0 & 000 \\ 1 & 1 \end{bmatrix}
\end{align*}
\]
Mixture of Gaussians – Other Strategy

- Assuming we knew the values of the hidden variable...

\[ f(x) \]

\[ x \]

ML for Gaussian #1

\[ h(j = 1 \mid x_n) = 1 \quad 1 \quad 1 \quad 1 \]
\[ h(j = 2 \mid x_n) = 0 \quad 0 \quad 0 \quad 0 \]

\[ \mu_1 = \frac{\sum_{n=1}^{N} h(j = 1 \mid x_n) x_n}{\sum_{i=1}^{N} h(j = 1 \mid x_n)} \]

ML for Gaussian #2

\[ h(j = 2 \mid x_n) = 1 \quad 1 \quad 1 \quad 1 \]

\[ \mu_2 = \frac{\sum_{n=1}^{N} h(j = 2 \mid x_n) x_n}{\sum_{i=1}^{N} h(j = 2 \mid x_n)} \]

Slide credit: Bernt Schiele

B. Leibe
Mixture of Gaussians – Other Strategy

- Assuming we knew the mixture components...

\[ f(x) \]

\[ p(j = 1|x) \] \[ p(j = 2|x) \]

1 111 22 2 2

- 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

• 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
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.

Slide credit: Bernt Schiele
K-Means – Example with K=2

Image source: C.M. Bishop, 2006
K-Means Clustering

- K-Means optimizes the following objective function:

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

where

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

- I.e., \( r_{nk} \) is an indicator variable that checks whether \( \mathbf{\mu}_k \) 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

Take each pixel as one data point.

Set the pixel color to the cluster mean.

K-Means Clustering

Image source: C.M. Bishop, 2006
Example Application: Image Compression

\( K = 2 \) \hspace{1cm} \( K = 3 \) \hspace{1cm} \( K = 10 \) \hspace{1cm} Original image

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

\[
p(j|x) = \begin{cases} 
0.99 & 0.8 & 0.2 & 0.01 \\
0.01 & 0.2 & 0.8 & 0.99 
\end{cases}
\]
EM Clustering

- Clustering with “soft assignments”
  - Maximization step of the EM algorithm

Maximum Likelihood estimate

$$\mu_j = \frac{\sum_{n=1}^{N} p(j|x_n)x_n}{\sum_{n=1}^{N} p(j|x_n)}$$
Credit Assignment Problem

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

  “responsibility” of component $j$ for $x$. 

Slide credit: Bernt Schiele
EM Algorithm

• **Expectation-Maximization (EM) Algorithm**
  - **E-Step**: softly assign samples to mixture components
    \[
    \gamma_j(x_n) \leftarrow \frac{\pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)} \quad \forall j = 1, \ldots, K, \ n = 1, \ldots, N
    \]
  - **M-Step**: re-estimate the parameters (separately for each mixture component) based on the soft assignments
    \[
    \hat{N}_j \leftarrow \sum_{n=1}^{N} \gamma_j(x_n) = \text{soft number of samples labeled } j
    \]
    \[
    \hat{\pi}_j \leftarrow \frac{\hat{N}_j}{N}
    \]
    \[
    \hat{\mu}_j \leftarrow \frac{1}{\hat{N}_j} \sum_{n=1}^{N} \gamma_j(x_n) x_n
    \]
    \[
    \hat{\Sigma}_j \leftarrow \frac{1}{\hat{N}_j} \sum_{n=1}^{N} \gamma_j(x_n)(x_n - \hat{\mu}_j)(x_n - \hat{\mu}_j)^T
    \]

Slide adapted from Bernt Schiele

B. Leibe
EM Algorithm – An Example

Image source: C.M. Bishop, 2006
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 $\Sigma_k = \sigma_k^2 I$ (this 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
    $\mathcal{N}(x_n | x_n, \sigma_j^2 I) = \frac{1}{\sqrt{2\pi}\sigma_j}$
  - 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 $\Sigma^{-1}$, use $(\Sigma + \sigma_{\text{min}} I)^{-1}$

Image source: C.M. Bishop, 2006
EM – Technical Advice (2)

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

⇒ 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 $\mu_j$ to the corresponding cluster mean from k-Means.
    - Initialize $\Sigma_j$ 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 $\sigma^2$

---

*Slide credit: Bernt Schiele*
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.

Image source: C.M. Bishop, 2006
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.


Image Source: Daniel Roth, Tobias Jäggi
Application: Image 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:
  ➢ EM tutorial: