Machine Learning – Lecture 3

Probability Density Estimation II

19.10.2017

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

• Exam dates
  ➢ We’re in the process of fixing the first exam date

• Exercises
  ➢ The first exercise sheet is available on L2P now
  ➢ First exercise lecture on 30.10.2017
  ⇒ Please submit your results by evening of 29.10. via L2P
    (detailed instructions can be found on the exercise sheet)
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
Topics of This Lecture

• Recap: Parametric Methods
  - Gaussian distribution
  - Maximum Likelihood approach

• Non-Parametric Methods
  - Histograms
  - Kernel density estimation
  - K-Nearest Neighbors
  - k-NN for Classification

• Mixture distributions
  - Mixture of Gaussians (MoG)
  - Maximum Likelihood estimation attempt
Recap: Gaussian (or Normal) Distribution

- **One-dimensional case**
  - Mean $\mu$
  - Variance $\sigma^2$

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

- **Multi-dimensional case**
  - Mean $\mu$
  - Covariance $\Sigma$

  $$N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma|}} \exp\left\{ -\frac{1}{2}(x - \mu)^T\Sigma^{-1}(x - \mu) \right\}$$

Image source: C.M. Bishop, 2006
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)
  \( \Rightarrow \) Take the derivative and set it to zero.
  \[
  \frac{\partial}{\partial \theta} E(\theta) = - \sum_{n=1}^{N} \frac{\partial}{\partial \theta} p(x_n|\theta) \frac{p(x_n|\theta)}{p(x_n|\theta)} = 0
  \]
Maximum Likelihood Approach

• When applying ML to the Gaussian distribution, we obtain

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

“sample mean”

• In a similar fashion, we get

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

“sample variance”

• \( \hat{\theta} = (\hat{\mu}, \hat{\sigma}) \) is the Maximum Likelihood estimate for the parameters of a Gaussian distribution.

• This is a very important result.

• Unfortunately, it is wrong…
Maximum Likelihood Approach

- Or not wrong, but rather **biased**...

- Assume the samples $x_1, x_2, \ldots, x_N$ come from a true Gaussian distribution with mean $\mu$ and variance $\sigma^2$
  
  - We can now compute the expectations of the ML estimates with respect to the data set values. It can be shown that
    
    $$
    \mathbb{E}(\mu_{\text{ML}}) = \mu
    $$

    $$
    \mathbb{E}(\sigma^2_{\text{ML}}) = \left( \frac{N-1}{N} \right) \sigma^2
    $$

    ⇒ The ML estimate will underestimate the true variance.

- Corrected estimate:

  $$
  \tilde{\sigma}^2 = \frac{N}{N-1} \sigma^2_{\text{ML}} = \frac{1}{N-1} \sum_{n=1}^{N} (x_n - \hat{\mu})^2
  $$
Maximum Likelihood – Limitations

- Maximum Likelihood has several significant limitations
  - It systematically underestimates the variance of the distribution!
  - E.g. consider the case
    \[ N = 1, \, X = \{ x_1 \} \]

  \[ \Rightarrow \text{Maximum-likelihood estimate:} \]
  \[ \hat{\sigma}^2 = \frac{1}{N} \sum_{n=1}^{N} (x_n - \hat{\mu})^2 \]

  - We say ML overfits to the observed data.
  - We will still often use ML, but it is important to know about this effect.
Deeper Reason

- Maximum Likelihood is a **Frequentist** concept
  - In the **Frequentist view**, probabilities are the frequencies of random, repeatable events.
  - These frequencies are fixed, but can be estimated more precisely when more data is available.

- This is in contrast to the **Bayesian** interpretation
  - In the **Bayesian view**, probabilities quantify the uncertainty about certain states or events.
  - This uncertainty can be revised in the light of new evidence.

- Bayesians and Frequentists do not like each other too well…
Bayesian vs. Frequentist View

• To see the difference…
  ➢ Suppose we want to estimate the uncertainty whether the Arctic ice cap will have disappeared by the end of the century.
  ➢ This question makes no sense in a Frequentist view, since the event cannot be repeated numerous times.
  ➢ In the Bayesian view, we generally have a prior, e.g. from calculations how fast the polar ice is melting.
  ➢ If we now get fresh evidence, e.g. from a new satellite, we may revise our opinion and update the uncertainty from the prior.

Posterior $\propto$ Likelihood $\times$ Prior

➢ This generally allows to get better uncertainty estimates for many situations.

• Main Frequentist criticism
  ➢ The prior has to come from somewhere and if it is wrong, the result will be worse.
Bayesian Approach to Parameter Learning

• Conceptual shift
  ➢ Maximum Likelihood views the true parameter vector \( \theta \) to be unknown, but fixed.
  ➢ In Bayesian learning, we consider \( \theta \) to be a random variable.

• This allows us to use knowledge about the parameters \( \theta \)
  ➢ i.e. to use a prior for \( \theta \)
  ➢ Training data then converts this prior distribution on \( \theta \) into a posterior probability density.

  ➢ The prior thus encodes knowledge we have about the type of distribution we expect to see for \( \theta \).
Bayesian Learning

• Bayesian Learning is an important concept
  ➢ However, it would lead to far here.
  ⇒ I will introduce it in more detail in the Advanced ML lecture.
Topics of This Lecture

• Recap: Parametric Methods
  ➢ Gaussian distribution
  ➢ Maximum Likelihood approach

• Non-Parametric Methods
  ➢ Histograms
  ➢ Kernel density estimation
  ➢ K-Nearest Neighbors
  ➢ k-NN for Classification

• Mixture distributions
  ➢ Mixture of Gaussians (MoG)
  ➢ Maximum Likelihood estimation attempt
Non-Parametric Methods

• Non-parametric representations
  ▶ Often the functional form of the distribution is unknown

• Estimate probability density from data
  ▶ Histograms
  ▶ Kernel density estimation (Parzen window / Gaussian kernels)
  ▶ k-Nearest-Neighbor

Slide credit: Bernt Schiele
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 $D$!
Histograms

- The bin width $\Delta$ acts as a smoothing factor.

  not smooth enough

  about OK

  too smooth

Image source: C.M. Bishop, 2006
Summary: Histograms

• Properties
  - Very general. In the limit \( N \to \infty \), every probability density can be represented.
  - No need to store the data points once histogram is computed.
  - Rather brute-force

• Problems
  - High-dimensional feature spaces
    - \( D \)-dimensional space with \( M \) bins/dimension will require \( M^D \) bins!
    \( \Rightarrow \) Requires an exponentially growing number of data points
    \( \Rightarrow \) “Curse of dimensionality”
  - Discontinuities at bin edges
  - Bin size?
    - too large: too much smoothing
    - too small: too much noise
Statistically Better-Founded Approach

• Data point $x$ comes from pdf $p(x)$
  ➢ Probability that $x$ falls into small region $\mathcal{R}$
  
  $$P = \int_{\mathcal{R}} p(y) \, dy$$

• If $\mathcal{R}$ is sufficiently small, $p(x)$ is roughly constant
  ➢ Let $V$ be the volume of $\mathcal{R}$
  
  $$P = \int_{\mathcal{R}} p(y) \, dy \approx p(x)V$$

• If the number $N$ of samples is sufficiently large, we can estimate $P$ as

  $$P = \frac{K}{N} \quad \Rightarrow \quad p(x) \approx \frac{K}{NV}$$

Slide credit: Bernt Schiele
Statistically Better-Founded Approach

\[ p(x) \approx \frac{K}{NV} \]

- Kernel Methods
- K-Nearest Neighbor

- Kernel methods
  - Example: Determine the number \( K \) of data points inside a fixed hypercube...

Slide credit: Bernt Schiele
Kernel Methods

- Parzen Window
  - Hypercube of dimension $D$ with edge length $h$:

$$k(u) = \begin{cases} 
1, & |u_i| \leq \frac{1}{2}h, \\
0, & \text{else}
\end{cases}$$

“Kernel function”

$$K = \sum_{n=1}^{N} k(x - x_n) \quad V = \int k(u)du = h^D$$

- Probability density estimate:

$$p(x) \approx \frac{K}{NV} = \frac{1}{Nh^D} \sum_{n=1}^{N} k(x - x_n)$$

Slide credit: Bernt Schiele
Kernel Methods: Parzen Window

• Interpretations
  1. We place a kernel window \( k \) at location \( x \) and count how many data points fall inside it.
  2. We place a kernel window \( k \) around each data point \( x_n \) and sum up their influences at location \( x \).
     \( \Rightarrow \) Direct visualization of the density.

• Still, we have artificial discontinuities at the cube boundaries…
  - We can obtain a smoother density model if we choose a smoother kernel function, e.g. a Gaussian
Kernel Methods: Gaussian Kernel

- Gaussian kernel
  - Kernel function
    \[ k(u) = \frac{1}{(2\pi h^2)^{1/2}} \exp \left\{ -\frac{u^2}{2h^2} \right\} \]
  - Probability density estimate
    \[ p(x) \approx \frac{K}{NV} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{(2\pi)^{D/2}h} \exp \left\{ -\frac{||x - x_n||^2}{2h^2} \right\} \]
  
  \[ K = \sum_{n=1}^{N} k(x - x_n) \quad V = \int k(u) du = 1 \]

Slide credit: Bernt Schiele
Gauss Kernel: Examples

- not smooth enough
- about OK
- too smooth

$h$ acts as a smoother.

Image source: C.M. Bishop, 2006
Kernel Methods

- In general
  - Any kernel such that
    \[ k(u) \geq 0, \quad \int k(u) \, du = 1 \]
    can be used. Then
    \[ K = \sum_{n=1}^{N} k(x - x_n) \]
    - And we get the probability density estimate
      \[ p(x) \approx \frac{K}{NV} = \frac{1}{N} \sum_{n=1}^{N} k(x - x_n) \]

Slide adapted from Bernt Schiele
Statistically Better-Founded Approach

\[ p(x) \approx \frac{K}{NV} \]

fixed \( V \)
determine \( K \)

fixed \( K \)
determine \( V \)

Kernel Methods    K-Nearest Neighbor

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

Slide credit: Bernt Schiele
K-Nearest Neighbor

• Nearest-Neighbor density estimation
  - Fix $K$, estimate $V$ from the data.
  - Consider a hypersphere centred on $x$ and let it grow to a volume $V^*$ that includes $K$ of the given $N$ data points.
  - Then
    \[ p(x) \approx \frac{K}{NV^*}. \]

• Side note
  - Strictly speaking, the model produced by K-NN is not a true density model, because the integral over all space diverges.
  - E.g. consider $K = 1$ and a sample exactly on a data point $x = x_j$. 
k-Nearest Neighbor: Examples

- not smooth enough
- about OK
- too smooth

\[ K \text{ acts as a smoother.} \]

Image source: C.M. Bishop, 2006
Summary: Kernel and k-NN Density Estimation

• Properties
  ➢ Very general. In the limit \( N \to \infty \), every probability density can be represented.
  ➢ No computation involved in the training phase
    ⇒ Simply storage of the training set

• Problems
  ➢ Requires storing and computing with the entire dataset.
    ⇒ Computational cost linear in the number of data points.
    ⇒ This can be improved, at the expense of some computation during training, by constructing efficient tree-based search structures.
  ➢ Kernel size / \( K \) in K-NN?
    – Too large: too much smoothing
    – Too small: too much noise
K-Nearest Neighbor Classification

- Bayesian Classification

$$p(C_j|x) = \frac{p(x|C_j)p(C_j)}{p(x)}$$

- Here we have

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

$$p(x|C_j) \approx \frac{K_j}{N_jV}$$

$$p(C_j) \approx \frac{N_j}{N}$$

$$p(C_j|x) \approx \frac{K_j}{N_jV} \frac{N_j}{N} \frac{NV}{K} = \frac{K_j}{K}$$

k-Nearest Neighbor classification

Slide credit: Bernt Schiele
K-Nearest Neighbors for Classification

\[ x_2 \]

\[ x_1 \]

\[ K = 3 \]

\[ K = 1 \]

Image source: C.M. Bishop, 2006
K-Nearest Neighbors for Classification

- Results on an example data set

- $K$ acts as a smoothing parameter.
- Theoretical guarantee
  - For $N \to \infty$, the error rate of the 1-NN classifier is never more than twice the optimal error (obtained from the true conditional class distributions).

Image source: C.M. Bishop, 2006
Bias-Variance Tradeoff

- Probability density estimation
  - Histograms: bin size?
    - $\Delta$ too large: too smooth
    - $\Delta$ too small: not smooth enough
  - Kernel methods: kernel size?
    - $h$ too large: too smooth
    - $h$ too small: not smooth enough
  - K-Nearest Neighbor: $K$?
    - $K$ too large: too smooth
    - $K$ too small: not smooth enough

- This is a general problem of many probability density estimation methods
  - Including parametric methods and mixture models

Too much bias
Too much variance

Slide credit: Bernt Schiele
Discussion

• The methods discussed so far are all simple and easy to apply. They are used in many practical applications.

• However…
  - **Histograms** scale poorly with increasing dimensionality.  
    ⇒ Only suitable for relatively low-dimensional data.
  - Both **k-NN** and **kernel density estimation** require the entire data set to be stored.  
    ⇒ Too expensive if the data set is large.
  - Simple **parametric models** are very restricted in what forms of distributions they can represent.  
    ⇒ Only suitable if the data has the same general form.

• **We need density models that are efficient and flexible!**  
  ⇒ Next topic…
Topics of This Lecture

• Recap: Parametric Methods
  ➢ Gaussian distribution
  ➢ Maximum Likelihood approach

• Non-Parametric Methods
  ➢ Histograms
  ➢ Kernel density estimation
  ➢ K-Nearest Neighbors
  ➢ k-NN for Classification

• Mixture distributions
  ➢ Mixture of Gaussians (MoG)
  ➢ Maximum Likelihood estimation attempt
Mixture Distributions

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

![Mixture Distributions Diagram](image-source: C.M. Bishop, 2006)
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

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

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

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

- **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

\[
p(j) = \pi_j
\]

- Mixture component

- Mixture density
Mixture of Multivariate Gaussians

(a) and (b) are two separate plots showing the distribution of data points with ellipses indicating the variance of the Gaussian distributions. The numbers 0.5 and 0.3 likely represent probabilities or weights associated with each Gaussian.

(c) is a 3D representation of the same mixture of Gaussians, showing the density of the distribution in a higher dimension.

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

- Multivariate Gaussians

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

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

- Mixture weights / mixture coefficients:

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

- Parameters:

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

- "Generative model"

\[ p(j) = \pi_j \]

\[ p(x|\theta) = \sum_{j=1}^{3} \pi_j p(x|\theta_j) \]
Mixture of Gaussians – 1\textsuperscript{st} 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 – 1st Estimation Attempt

• Minimization:

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

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

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

• We thus obtain

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

\[
= \gamma_j(x_n)
\]

“responsibility” of component \(j\) for \(x_n\)

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

- But…

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

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

- 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”: \( j | x \)

\[
\begin{align*}
    h(j = 1 | x_n) &= 1 \ 111 \ 00 \ 0 \ 0 \\
    h(j = 2 | x_n) &= 0 \ 000 \ 11 \ 1 \ 1 
\end{align*}
\]
Mixture of Gaussians – Other Strategy

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

![Graph showing two Gaussian distributions](image)

- ML for Gaussian #1
  - Assumed known: 1 111
  - \( h(j = 1|x_n) = 1 111 \)
  - \( h(j = 2|x_n) = 0 000 \)

- ML for Gaussian #2
  - Assumed known: 22 2 2
  - \( h(j = 1|x_n) = 0 0 0 \)
  - \( h(j = 2|x_n) = 1 1 1 \)

\[
\mu_1 = \frac{\sum_{n=1}^{N} h(j = 1|x_n)x_n}{\sum_{i=1}^{N} h(j = 1|x_n)} \\
\mu_2 = \frac{\sum_{n=1}^{N} h(j = 2|x_n)x_n}{\sum_{i=1}^{N} h(j = 2|x_n)}
\]
Mixture of Gaussians – Other Strategy

• Assuming we knew the mixture components…

\[ f(x) \]

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

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

1 111 22 2

• Bayes decision rule: Decide \( j = 1 \) if

\[ p(j = 1 | x_n) > p(j = 2 | x_n) \]
Mixture of Gaussians – Other Strategy

• Chicken and egg problem – what comes first?

\[ f(x) \]

We don’t know any of those!

• In order to break the loop, we need an estimate for \( j \).
  
  ➢ E.g. by clustering…
  
  ⇒ Next lecture…

Slide credit: Bernt Schiele
References and Further Reading

• More information in Bishop’s book
  ➢ Gaussian distribution and ML: Ch. 1.2.4 and 2.3.1-2.3.4.
  ➢ Bayesian Learning: Ch. 1.2.3 and 2.3.6.
  ➢ Nonparametric methods: Ch. 2.5.

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