Universität Stuttgart

Pattern Recognition
Chapter 10: Bayesian Classification

Prof. Dr.-Ing. Uwe Sörgel
soergel@ifp.uni-stuttgart.de

ifp

Contents

Bayesian Classification

- Generative approach:
  - The posterior probability $p(C|\mathbf{x})$ is maximized.
  - Posterior $p(C|\mathbf{x})$ is modelled indirectly according to the Theorem of Bayes.
  - This requires a model of the joint distribution $p(C,\mathbf{x})$ of the data $\mathbf{x}$ and the class labels C.
  - It is possible to *generate* synthetic data sets by sampling from the joint distribution.

- Strong theoretical foundation:
  - If the required distributions are known, Bayesian classification will deliver the result with the lowest proportion of classification errors!

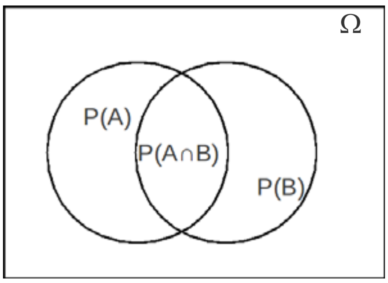
Motivation: Recap probabilities I

- A subset A of a population Ω suffers from cancer. By normalization we yield a probability that a person we sample carries this disease:

$$\frac{|A|}{|\Omega|} = P(A)$$

- A drug company invents some screening test, which is either "positive" (indicating cancer) for some people (set B) and "negative" for the rest:

$$\frac{|B|}{|\Omega|} = P(B)$$



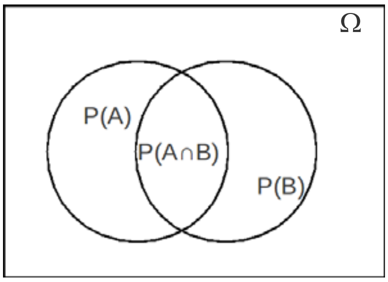
https://oscarbonilla.com/2009/05/visualizing-bayes-theorem/

Motivation: Recap probabilities II

- The joint probability A,B (shorthand $A \cap B$) is: $\frac{|A,B|}{|\Omega|} = P(A,B)$

- We ask: “Given that the test is positive for a randomly selected individual, what is the probability that said individual has cancer?”

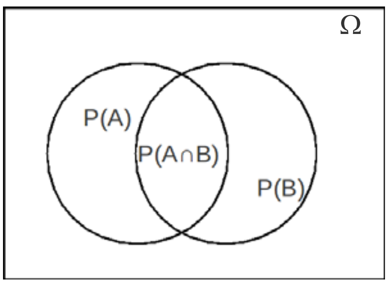
  - This is a conditional probability $P(A|B) = \frac{P(A,B)}{P(B)}$



Motivation: Recap probabilities III

- Now let us ask “Given that a randomly selected individual has cancer (event A), what is the probability that the test is positive for that individual (event AB)?”
  - This is of course again a conditional probability: $P(B|A) = \frac{P(A,B)}{P(A)}$
  - We have now: $P(A|B) = \frac{P(A,B)}{P(B)}$ and $P(B|A) = \frac{P(A,B)}{P(A)}$

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$



Theorem of Bayes: Derivation for our purpose

- For the joint distribution $p(\mathbf{x},C)$ of data $\mathbf{x}$ and classes C the product rule applies:

$$p(\mathbf{x},C) = p(C|\mathbf{x}) \cdot p(\mathbf{x})$$

- Likewise: $$p(C,\mathbf{x}) = p(\mathbf{x}|C) \cdot p(C)$$

- Due to $p(\mathbf{x},C) = p(C,\mathbf{x})$:

$$p(C|\mathbf{x}) \cdot p(\mathbf{x}) = p(\mathbf{x}|C) \cdot p(C)$$

- Therefore: $$p(C|\mathbf{x}) = \frac{p(\mathbf{x}|C) \cdot p(C)}{p(\mathbf{x})}$$

Theorem of Bayes

Theorem of Bayes: Interpretation

- causal relation between object type and observed features: the observed features are a function of the object type.

- Usually it is easier to deduce the effect from the cause, i.e., it would seem to be easier to deduce the features from the object type.

- The theorem of Bayes allows inverse reasoning: derive information about the cause (the object type) from the effect (the observed features).

$$p(C|\mathbf{x}) = \frac{p(\mathbf{x}|C) \cdot p(C)}{p(\mathbf{x})}$$

Theorem of Bayes: Meaning of the terms I

- $p(C)$: prior probability

$$p(C|\mathbf{x}) = \frac{p(\mathbf{x}|C) \cdot p(C)}{p(\mathbf{x})}$$

- Corresponds to knowledge (bias) for the occurrence of C.
- If no information is available: Uniform Distribution
  → MAP becomes Maximum-Likelihood (ML)
- $p(C)$ can be determined iteratively:
  1. Classification under the assumption of a uniform distribution of the occurrence of the individual classes.
  2. Determination of $p(C)$ from the relative frequencies of occurrence of the individual classes C^k.
  3. Classification according to the theorem of Bayes.

Theorem of Bayes: Meaning of the terms II

- $p(\mathbf{x}|C)$: likelihood

$$p(C|\mathbf{x}) = \frac{p(\mathbf{x}|C) \cdot p(C)}{p(\mathbf{x})}$$

- Probability to observe $\mathbf{x}$ if it is known to belong to class C.
- Note: the Likelihood is no probability density function **of the Classes *C*!**
- For each class C^k there is a model for $p(x|C = C^k)$, which describes the distribution of the features for this class.
- Determination from data in training areas
- *Non-parametric Models*: Direct determination of $p(\mathbf{x}|C)$ from the training data
- *Parametric Models*: Based on the assumption of an **analytical model** for $p(\mathbf{x}|C)$, whose parameters are estimated from the training data.

Paranthesis:
Likelihood function vs. probability density function

- **Probability density function :**
  - We have a set of n samples $x_1, \ldots, x_n$ of n independent and identically distributed random variables $X_1, \ldots, X_n$.
  - We know the joint probability density $p(\mathbf{x}, \theta)$ governed by fixed (given) parameters θ.
  - Then we may **factorize** the joint probability density like this:

$$p\left(x_1,\ldots,x_n\middle|\theta\right) = p\left(x_1\middle|\theta\right)p\left(x_2\middle|\theta\right)\ldots p\left(x_n\middle|\theta\right) = \prod_{i=1}^{n} p\left(x_i\middle|\theta\right)$$

- Likelihood **function** L:
  - We kind of turn the tables by considering now $x_1, \ldots, x_n$ as given and θ as random variables.
  - However, eventually we yield the exact same factorization as above:

$$L\left(\theta\middle|x_1,\ldots,x_n\right) = p\left(x_1,\ldots,x_n\middle|\theta\right) = p\left(x_1\middle|\theta\right)\ldots p\left(x_n\middle|\theta\right) = \prod_{i=1}^{n} p\left(x_i\middle|\theta\right)$$

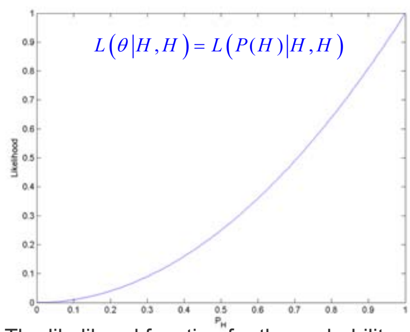
- Comments
  - The Likelihood function **is no probability density function**:
    - In case we integrate over parameter space θ, the integral is usually unequal to 1.
  - Important application: **Maximum Likelihood Method** (Search for best θ).

Example for Likelihood function: Coin tossing

- Two possible outcomes:
  - Head (H) or tail (T): $P(H) = \theta$ and $P(T) = 1 - \theta$
  - Let us toss two times: In general: $L\left(\theta\middle|x_1, x_2\right) = p\left(x_1\middle|\theta\right)\cdot p\left(x_2\middle|\theta\right)$
    - Observation: HH
    - Case 1: Fair coin: $\theta = 0.5 \rightarrow L\left(0.5\middle|H,H\right) = p\left(H\middle|0.5\right)\cdot p\left(H\middle|0.5\right) = 0.5^2 = 0.25$
    - Case 2: Biased coin, e.g. $\theta = 0.3 \rightarrow L\left(0.3\middle|H,H\right) = p\left(H\middle|0.3\right)\cdot p\left(H\middle|0.3\right) = 0.3^2 = 0.09$

$$HH \rightarrow \int_0^1 L\left(\theta\middle|H,H\right)d\theta = \int_0^1 p\left(H\middle|\theta\right)^2 d\theta =$$

$$\int_0^1 \theta^2 d\theta = \left[\frac{1}{3}\theta^3\right]_0^1 = \frac{1}{3}$$



The likelihood function for the probability of a coin landing heads-up (without prior knowledge) after observing HH (Wikipedia).

Theorem of Bayes: Meaning of the terms III

- $p(\mathbf{x})$: probability of the data (also called evidence)

$$p(C|\mathbf{x}) = \frac{p(\mathbf{x}|C)\cdot p(C)}{p(\mathbf{x})}$$

- Equal for all values of C becauses it does not depend on C.
  $\Rightarrow$ MAP can also be applied without knowing $p(\mathbf{x})$:

$$p(C|\mathbf{x}) \propto p(\mathbf{x}|C)\cdot p(C)$$

$$\Rightarrow \max\left(p(C|\mathbf{x})\right) = \max\left(p(\mathbf{x}|C)\cdot p(C)\right)$$

- $p(\mathbf{x})$ ensures that $p(C|\mathbf{x})$ can be interpreted as a probability and can be used as such in further probabilistic processes.

- $p(\mathbf{x})$ can be determined as the **marginal distribution** of $p(\mathbf{x},C)$:

$$p(\mathbf{x}) = \sum_k p(\mathbf{x}|C^k)\cdot p(C^k)$$

Theorem of Bayes: Example

- It is known that from 100000 people 20 suffer from a certain severe illness:

$$p(K = ill) = 0.0002,\ p(\bar{K} = healthy) = 0.9998$$

- It exists a screening method for this disease:
  - Sensitivity of the tests: 95% of all ill persons are detected ($T=1$):

$$p(T|K) = 0.95,\ p(\bar{T}|K) = 0.05$$

  - Unfortunately, the test delivers false positive result for 1% of healthy persons:

$$p(T|\bar{K}) = 0.01,\ p(\bar{T}|\bar{K}) = 0.99$$

- We may be interested in the portion of ill persons in the set of all persons with positive test results:

$$p(K|T) = \frac{p(T|K)\,p(K)}{p(T|K)\,p(K) + p(T|\bar{K})\,p(\bar{K})} = \underline{\qquad\qquad} = 0.0186$$

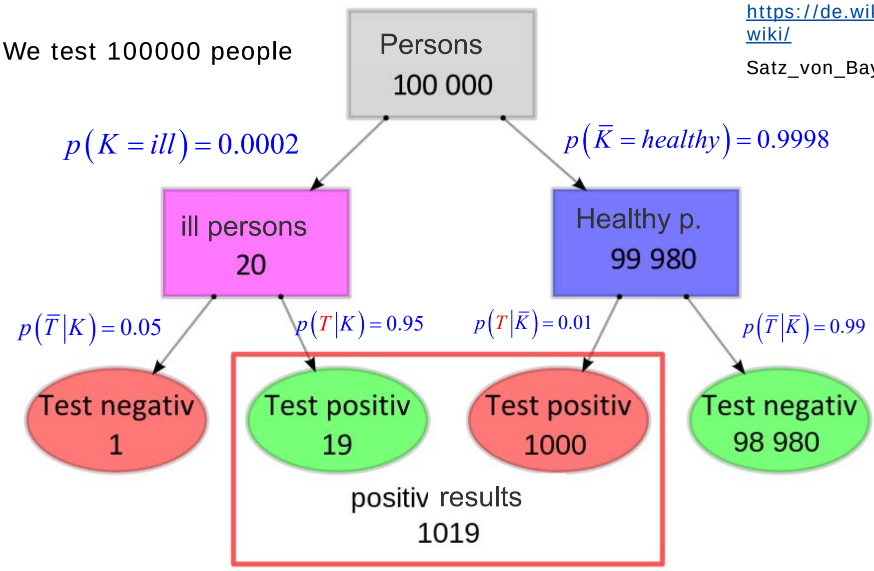
$p(T)$: Sum over all classes (here: 2)

https://de.wikipedia.org/wiki/Satz_von_Bayes

Visualization using event tree I

We test 100000 people

https://de.wikipedia.org/wiki/Satz_von_Bayes

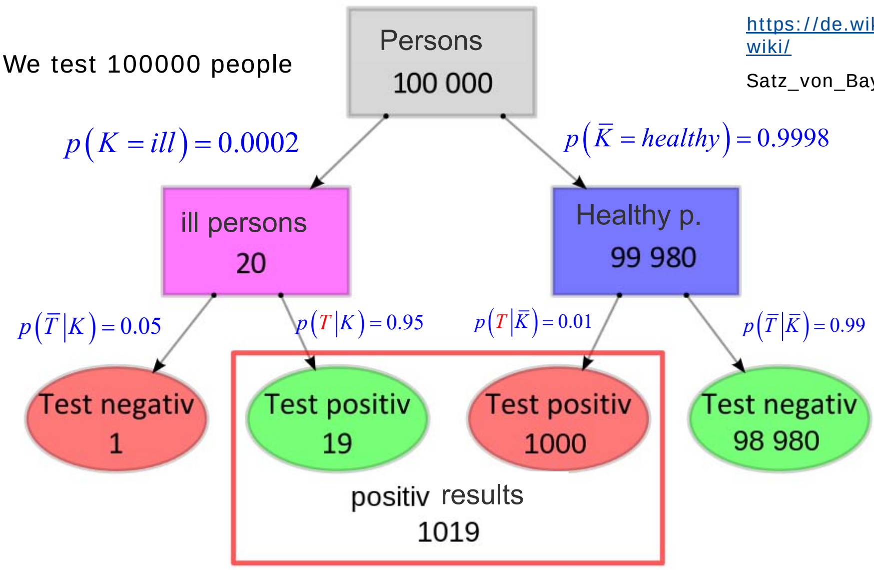


$$p(K|T) = \frac{p(T|K)\,p(K)}{p(T|K)\,p(K) + p(T|\bar{K})\,p(\bar{K})} = \frac{0.95 \cdot 0.0002}{0.95 \cdot 0.0002 + 0.01 \cdot 0.9998} \approx 0.0186$$

Visualization using event tree II

We test 100000 people

https://de.wikipedia.org/wiki/Satz_von_Bayes



Other conclusion possible, e.g.:

$$p(\bar{K}|T) = 1 - p(K|T) = \frac{p(T|\bar{K})\,p(\bar{K})}{p(T|K)\,p(K) + p(T|\bar{K})\,p(\bar{K})} = \frac{0.01 \cdot 0.9998}{0.95 \cdot 0.0002 + 0.01 \cdot 0.9998} \approx 0.9814$$

Workflow of Bayesian classification

- Given:
  - Models for the likelihoods $p(\mathbf{x}|C^k)$ of all classes C^k
  - Priori probabilities $p(C^k)$ of all classes C^k
  - A feature vector $\mathbf{x}$ to be classified

- Wanted: Class C_{map} of $\mathbf{x}$ according to the MAP criterion.

- Procedure:
  1. For all C^k: calculate $p(\mathbf{x}, C^k) = p(\mathbf{x}|C^k) \cdot p(C^k)$
  2. Calculate $p(\mathbf{x}) = \sum_k p(\mathbf{x}|C^k) \cdot p(C^k)$
  3. For all C^k: calculate $p(C^k|\mathbf{x}) = p(\mathbf{x}, C^k) / p(\mathbf{x})$
  4. C_{map} results as the label C^k for which $p(C^k|\mathbf{x})$ is a maximum.

Training

- **Training: provision of examples**
  - User marks image regions which correspond to a class C^k.
  - Assumption: all pixels in the selected region belong to C^k.
  - Training areas must be provided for all classes
  - The training data must be representative for all classes
- **Modelling of the likelihood for the classes:**
  - Based on training data
  - Different for parametric and non-parametric methods.

Contents

- Theorem of Bayes
- Modelling of the likelihood function
  - Non-parametric techniques
  - Parametric techniques
- Modelling of the prior probability
- Discussion

Likelihood: Non-parametric methods

- Likelihood $p(\mathbf{x}|C)$: Conditional probability to observe the data $\mathbf{x}$ if the class C is known.
- Non-parametric techniques for modelling:
  - Histograms
  - Kernel density estimation
  - Techniques based on nearest neighbors

Likelihood based on Histograms: 1D Case

- Discrete variables (e.g. gray values g):

$$p\left(x = g \middle| C^k\right) = K_k / N_k$$

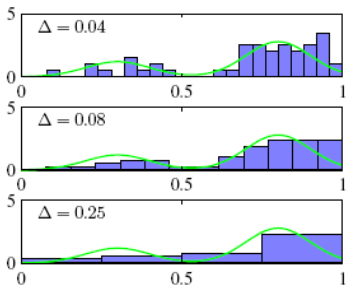
  - K_k ... Number of pixels in the training areas of the class C^k with grey value g
  - N_k ... Number of pixels in the training areas for the class C^k
  - Implementation via lookup tables $L_k(g)$ for each class C^k
  - Fast both for training and classification

Likelihood based on Histograms: 1D Case

- Continuous variables: discretization with grid width Δ.
- For the estimation of probability at location i the step size Δ_i needs to be considered:

$$p_i(x) = \frac{K_i}{N \cdot \Delta_i}, \quad \int p_i(x)\,dx = 1$$

- Usually equidistant step size Δ
- Quality of the approximation depends on Δ:
  - N_k = 50 samples drawn from a bimodal distribution (green)
  - Blue: histograms of the approximation
  - If Δ is too small: noisy approximation
  - If Δ is too large: smoothing too strong



© Bishop, 2006

- Problem: how to select the optimal value of Δ?

Likelihood from histograms: Multi-dimensional case

- Example (two grey values g_1, g_2, $\Delta_{x1} = \Delta_{x2} = \Delta$):

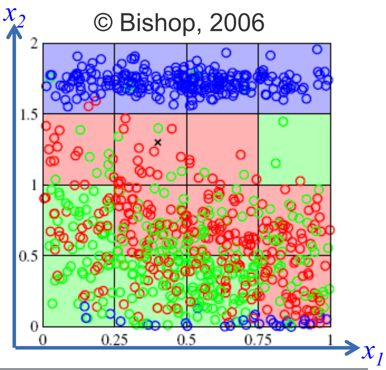
$$p\left(x_1 = g_1, x_2 = g_2 \middle| C^k\right) = \frac{K_k}{N_k \cdot \Delta^2}$$

  - K_k ... Number of pixels in the training areas of class C^k with the grey value combination (g_1, g_2)
  - N_k ... Number of pixels in the training areas for C^k

- Q possible values for each feature

  → Q^2 sub-squares, in which $p(x_1,x_2| C^k)$ is to be determined

  (Example in Figure.: $Q=4$)



Likelihood from histograms: Multi-dimensional case

- If we have D features with Q possible values per feature

  → Q^D probabilities need to be determined!

- This means that Q^D parameters have to be determined from training data.

- Practically impossible for $D > 2$

  - „Curse of Dimensionality"

  - „Hughes phenomenon" [Hughes, 1968 (!)]:
    - Beyond a certain point, the classification accuracy is reduced by using additional features!

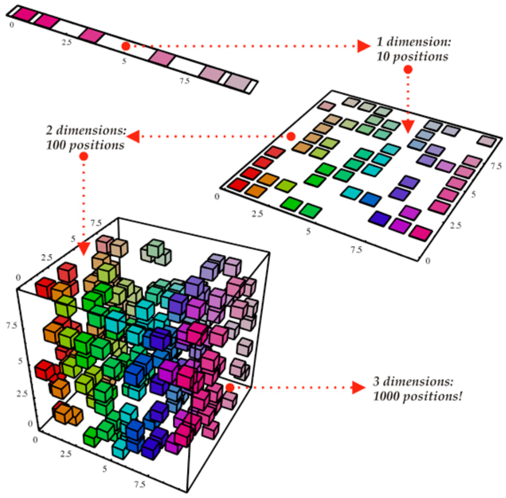
Multi-dimensional histograms: Curse of Dimensionality

- If we have D features with Q possible values per feature

  → Q^D probabilities need to be determined!

In order to maintain the same density of training data in the feature space, the data volume increases exponentially with dimension D, here ($Q=10$):

- 1-dim: 10^1
- 2-dim: 10^2
- 3-dim: 10^3



Multi-dimensional histograms: Curse of Dimensionality

- Examples for „Curse of Dimensionality“:
  - RGB image: 256 possible values for (R,G,B)
    - → $256^3 = 16.777.216$ probabilities
  - Feature vectors with $D = 40$ elements:
    Quantisation with 8 bit (256 possible values per feature)
    - → $256^{40} = 2.1 \cdot 10^{96}$ probabilities
    - Comparison: number of protons in the universe: $1.57 \cdot 10^{80}$!
- Can the problem be simplified by determination of the probabilities for each feature independently?

Likelihood from histograms: Multi-dimensional case

- Example for two features x_1, x_2:

$$p\left(x_1, x_2, C^k\right) = p\left(x_1, x_2 \middle| C^k\right) \cdot p\left(C^k\right)$$
$$= p\left(x_1 \middle| x_2, C^k\right) \cdot p\left(x_2, C^k\right)$$
$$= p\left(x_1 \middle| x_2, C^k\right) \cdot p\left(x_2 \middle| C^k\right) \cdot p\left(C^k\right)$$

  - thus $\Rightarrow p\left(x_1, x_2 \middle| C^k\right) = p\left(x_1 \middle| x_2, C^k\right) \cdot p\left(x_2 \middle| C^k\right)$

- In general, one cannot split ("factorize") $p(x_1,x_2|C^k)$ into a product of the form $p(x_1|C^k) \cdot p(x_2|C^k)$!

- **Exception**: the two variables are conditional independent

Conditional independence

- Two features x_1, x_2 are **conditionally independent** if $p(x_1|x_2,C^k)$ does not depend on x_2, i.e., if :

$$p\left(x_1 \middle| x_2, C^k\right) = p\left(x_1 \middle| C^k\right)$$

and, therefore,

$$p\left(x_1, x_2 \middle| C^k\right) = p\left(x_1 \middle| C^k\right) \cdot p\left(x_2 \middle| C^k\right)$$

- „Conditionally independent" means that x_1 *and* x_2 are statistically independent while **that C_k has occurred.**

- It does **not** mean that x_1 *und* x_2 are statistically independent in the general meaning of the word.

Conditional independence and the Naive Bayes Model

- If the features of a multidimensional feature vector $\mathbf{x}$ are conditionally independent, the likelihood can be factorized:

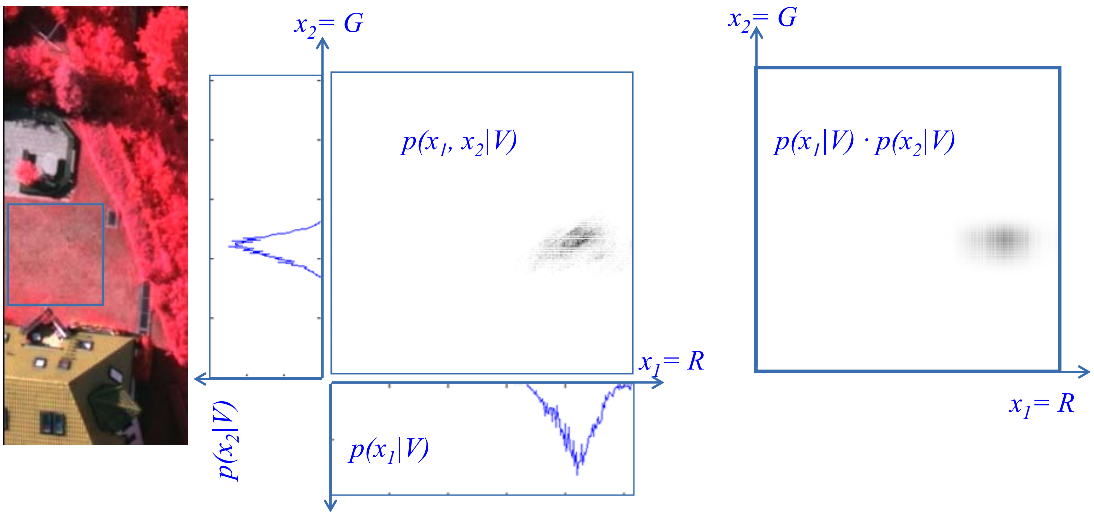
$$p\left(\mathbf{x}\middle|C^k\right) = p\left(x_1\middle|C^k\right)\cdot p\left(x_2\middle|C^k\right)\cdot\ldots\cdot p\left(x_D\middle|C^k\right)$$

- Consequence: the likelihood can be determined from the marginal distributions $p(x_i \mid C^k)$ → $Q \cdot D$ **instead of** Q^D **parameters**!

- This is called the naive Bayes model
  - Statistical dependencies between the features are neglected.
  - In general: too strong simplification.
  - May be justified if the features are determined from independent sensors.

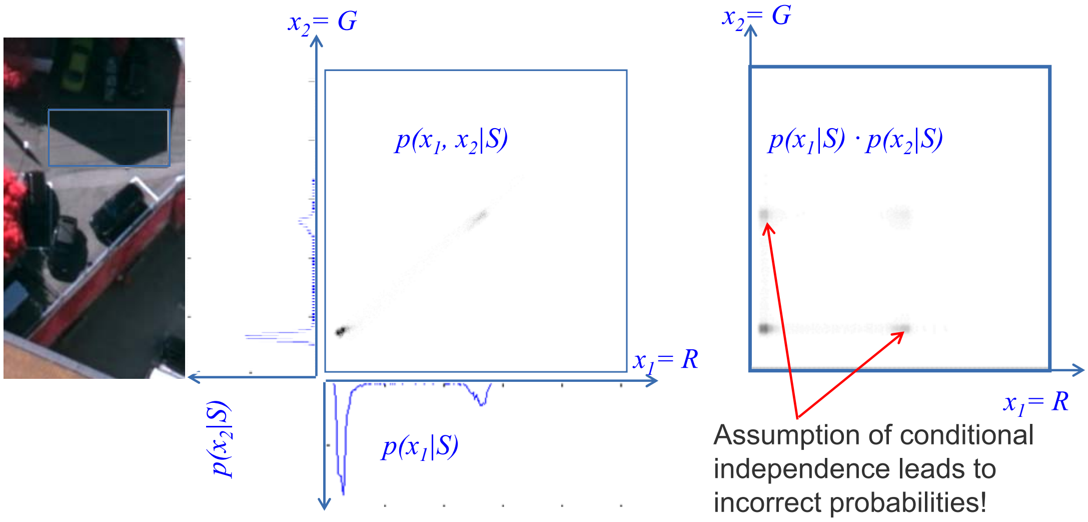
Example of impact of the Naive Bayes Model

- Aerial image with training area for „vegetation“ (V) (87 x 85 = 7395 pixels)



Example of impact of the Naive Bayes Model

- Aerial image with training area for „street“ (S)
  (49 x 102 = 4998 Pixel)



Other non-parametric techniques

- Probability P that a point $\mathbf{x}$ falls into a region R:

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

- If the volume V of R is so small that $p(\mathbf{x})$ is almost constant in R, one can approximate P by:

$$P \approx p(\mathbf{x}) \cdot V$$

- For a large number N of training samples $\mathbf{x}_i$ one can expect that $K \approx P \cdot N$ of these samples fall into R:

$$K \approx P \cdot N \approx p(\mathbf{x}) \cdot V \cdot N$$

$$\Rightarrow p(\mathbf{x}) \approx \frac{K}{N \cdot V}$$

Other non-parametric techniques

- Methods for the determination of the likelihood based on the approximation

$$p(\mathbf{x}) \approx \frac{K}{N \cdot V}$$

- Kernel density estimation:
  1. Define R (and, consequently, V)
  2. Count the number K of the points in $R \rightarrow p(\mathbf{x})$

- Techniques on the basis of nearest neighbors:
  1. Define K
  2. Determine $V \rightarrow p(\mathbf{x})$

Kernel density estimation I

- Definition of R as unit cube of side length 1 in feature space:

$$k(\mathbf{x}) = \begin{cases} 1, & |x_i| \leq 1/2 \\ 0, & sonst \end{cases}$$

- $k(\mathbf{x})$ is an example of a kernel function.
- Number K of the points inside a cube of side length h at point $\mathbf{x}$:

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

Therefore, using $V = h^D$ for the volume of the cube:

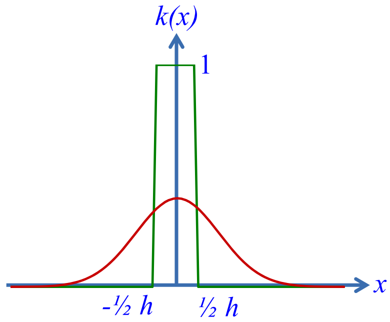
$$p(\mathbf{x} \mid C^k) = \frac{1}{N_k} \cdot \sum_{i=1}^{N_k} \frac{1}{h^D} \cdot k\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$$

with N_k ... Number of training points for the class C^k

Kernel density estimation II

- The proposed kernel function is not continuous at the boundaries of the cube.
- Transition to a smooth kernel, e.g., Gaussian kernel with width h:

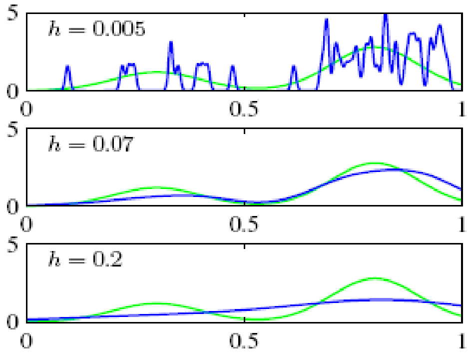
$$p\left(\mathbf{x} \mid C^k\right) = \frac{1}{N_k} \cdot \sum_{i=1}^{N_k} \frac{1}{\sqrt{2\pi} \cdot h} \cdot e^{-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2h^2}}$$



- No training in the sense of the determination of parameters required.
- For the classification of a new feature vector $\mathbf{x}$, the sum has to be evaluated using all training points $\mathbf{x}_i$ of the class C^k
  → slow for a large number of training points
- For classification, all training points must be available in RAM.

Kernel density estimation III

- Influence of the parameter h: Smoothing of the estimated probability density function.
- Example (Bishop, 2006):
  - $N_k = 50$ samples drawn from a bimodal distribution (green).
  - Blue curve: approximation of the probability density for different values of h.



© Bishop, 2006

- Choice of h is critical for success!
- Possible choice of h:
  - For each training sample: determine distance to its nearest neighbor in feature space
  - Set h to half the average distance.

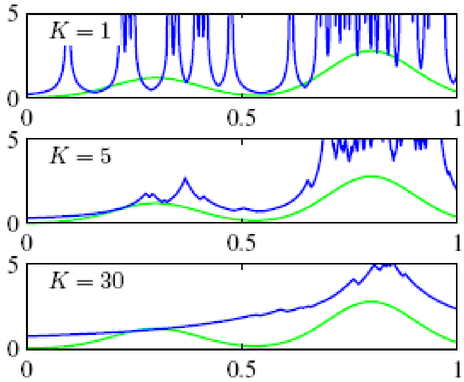
Nearest neighbor techniques I

- Remember: $$p(\mathbf{x}) \approx \frac{K}{N \cdot V}$$

- Procedure:

  1. Select K.
  2. Take a point and let the volume V grow until K points are inside..
  3. Calculate $p(\mathbf{x})$.

  → K nearest neighbor (KNN) techniques



© Bishop, 2006

- The parameter K heavily influences the quality of the approximation.

Nearest neighbor techniques II

- Direct classification with the KNN-Method:

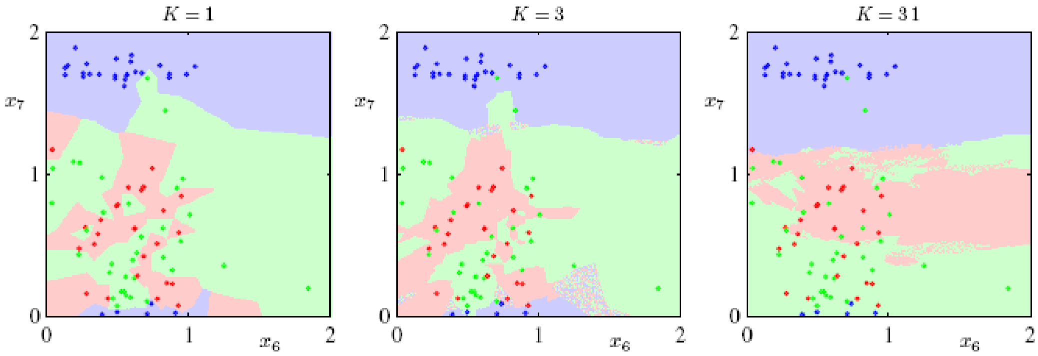
  - For a feature vector $\mathbf{x}$ that is to be classified:
    1. Search for the K nearest feature vectors in feature space among the training samples of all classes.
    2. For each class C^k: Determine the number K_k of the training samples among the K nearest neighbors that belong to C^k.
    3. Assign $\mathbf{x}$ to the class with maximum K_k.

- No training in the sense of the calculation of parameters, but all of the data points must be RAM.

- Requires a spatial index (e.g., kd tree) for efficient nearest neighbor search.

Nearest neighbor techniques III

- Example for the class boundaries as a function of K:



© Bishop, 2006

- KNN classification corresponds to Bayesian classification if the percentages of training samples per class are proportional to the prior probabilities.

Non-parametric techniques: Discussion

- Histograms: Not applicable for $D > 2$; however, 2D histograms are often used for visualization.

- Kernel density estimation and KNN: Are used occasionally

  - Need all training data in RAM at test time
  - Kernel density estimation: Time for classification increases linearly with the number of training samples
  - KNN: Needs efficient indexing
  - Require the choice of a parameter (h or K, which has a strong influence on the result.
  - Possible way to determine of h or K: cross-validation

Cross-Validation: Example kernel density estimation

- The training data are randomly divided into G groups (e.g., $G = 3$)

  - For different values of h that cover the entire possible range of values:

    1. For $g = 1 \ldots G$: Classify the training points of group g using the G-1 remaining groups for training.
    2. Determine the average training error (the number of training points assigned to a wrong class).
    3. Select the value of h for which the training error is a minimum.

- Cross validation can also be used to select one of several classification models.

Contents

- Theorem of Bayes
- Modelling of the likelihood function
  - Non-parametric techniques
  - Parametric techniques
- Modelling of the prior probability
- Discussion

Universität Stuttgart ifp

Likelihood: Parametric methods

- Here, an analytical model for the probability density $p(\mathbf{x}|C)$ is assumed.
- The probability density function $p(\mathbf{x}|C)$ also depends on parameters θ, which are determined from training data, i.e., $p(\mathbf{x}|C) = p(\mathbf{x}|C,\theta)$.
- The number of parameters θ is usually small, therefore the training effort is often considerable lower compared to non-parametric techniques.
- Training areas are required for each class C^k to determine the parameters θ_k of $p(\mathbf{x}|C^k,\theta_k)$.
- The training areas must be representative for the respective class

Training: Determination of the parameters I

- The likelihood $p(\mathbf{x}|C^k)$ is interpreted as $p(\mathbf{x}|C^k,\theta_k)$, where the vector θ_k contains all parameters of $p(\mathbf{x}|C^k)$.
- There are N_k statistically independent data samples $\mathbf{x}_{ik}$ for the class C^k (i.e., all points in the training areas for class C^k).
- Determination of θ_k: Probability $p(\theta_k \mid \mathbf{x}_{1k}, \ldots \mathbf{x}_{nk})$ of the parameter for the given training data should be maximum.
- Theorem of Bayes: $p\left(\theta_k \middle| \mathbf{x}_{1k}, \mathbf{x}_{2k} \ldots \mathbf{x}_{Nk}\right) \propto p\left(\mathbf{x}_{1k}, \mathbf{x}_{2k} \ldots \mathbf{x}_{Nk} \middle| \theta_k\right) \cdot p\left(\theta_k\right)$
- Due to the statistical independence of the samples $\mathbf{x}_{ik}$:

$$p\left(\mathbf{x}_{1k}, \mathbf{x}_{2k} \ldots \mathbf{x}_{Nk} \middle| \theta_k\right) = p\left(\mathbf{x}_{1k} \middle| \theta_k\right) \cdot p\left(\mathbf{x}_{2k} \middle| \theta_k\right) \cdot \ldots \cdot p\left(\mathbf{x}_{Nk} \middle| \theta_k\right)$$

Training: Determination of the parameters II

- Thus:

$$\underbrace{p\left(\theta_k \middle| \mathbf{x}_{1k}, \mathbf{x}_{2k} \ldots \mathbf{x}_{Nk}\right)}_{\text{posterior}} \propto \underbrace{p\left(\mathbf{x}_{1k} \middle| \theta_k\right) \cdot \ldots \cdot p\left(\mathbf{x}_{Nk} \middle| \theta_k\right)}_{\text{likelihood}} \cdot \underbrace{p\left(\theta_k\right)}_{\text{prior}}$$

- Estimation of θ_k according to the **maximum likelihood (ML)** principle:
  - Assumption of a uniform distribution of $p(\theta_k)$, therefore:

$$p\left(\mathbf{x}_{1k} \middle| \theta_k\right) \cdot \ldots \cdot p\left(\mathbf{x}_{Nk} \middle| \theta_k\right) \Rightarrow \max$$

- Bayesian Estimation
  - Requires knowledge about the prior $p(\theta_k)$

Parametric methods

- The choice of an analytical model for the likelihood $p(\mathbf{x}|C^k,\theta_k)$ depends on
  - The nature of the features
  - The expected distribution of the features in the feature space

- Different probability density functions for
  - Binary features
  - Discrete features
  - Continuous features

Binary features

- A feature x, which can take two values ($0,1$)
- Probability that x takes the value 1 or 0, respectively:

$$p(x=1)=\mu \quad \Rightarrow p(x=0)=1-\mu$$

- Bernoulli distribution: $p(x)=\mu^x \cdot (1-\mu)^{1-x}$
  - or in the case of the likelihood function $p(\mathbf{x}|C^k)$:

$$p(x\,|\,C^k,\mu_K)=\mu_k^x \cdot (1-\mu_k)^{1-x}$$

  - This is just another notation for $p(x\,|\,C^k,\mu_k)=\begin{cases}\mu_k & \textit{for} \quad x=1\\ 1-\mu_k & \textit{for} \quad x=0\end{cases}$

- For each class, one parameter μ_k must be determined $\rightarrow \theta_K = \mu_K$

Binary Features: Training

- Given: N_k independent training points x_{ik} for the class C^k
- Wanted: Parameter μ_k of the Bernoulli distribution for C^k
- Determination by the maximum likelihood method:
  - Maximize the probability of x_{ik} for given μ_k:

$$p(x_{1k},\ldots,x_{ik},\ldots x_{N_k k}\,|\,\mu_k)=\prod_{i=1}^{N_k}\mu_k^{x_i}\cdot(1-\mu_k)^{(1-x_i)}\rightarrow \max$$

Binary Features: Training

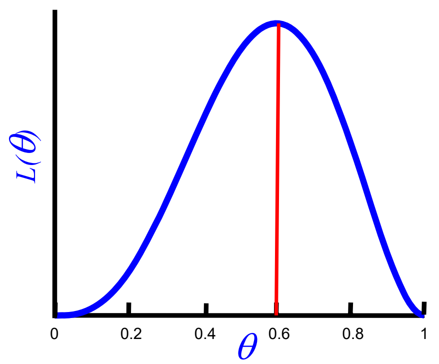
- Maximum likelihood estimation:
  - Equivalent problem: maximize the log-likelihood
    - The location of the maximum stays the same, the advantage is that **products turn to sums and exponents to products.**

$$\ln p\left(x_{1k},\ldots,x_{ik},\ldots x_{N_k k} \mid \mu_k\right) =$$
$$\sum_{i=1}^{N_k}\left[x_i \cdot \ln \mu_k + (1-x_i)\cdot \ln(1-\mu_k)\right] \rightarrow \max$$

- Result: $\mu_k = \frac{1}{N_k}\cdot\sum_{i=1}^{N_k} x_i = \frac{m_k}{N_k}$ with m_k ... Number of x_{ik} with $x_{ik}=1$

Example coin tossing

- Two possible values a feature can take (here parameter θ for μ):
  - Head (H) or tail (T): $P(H)=\theta$ und $P(T)=1-\theta$
  - We toss n times: $x_1 \ldots x_n$ $\quad L(\theta) = p(\mathbf{x} \mid \theta) = \prod_n P(x_i \mid \theta)$
  - Which choice of θ is optimal?
  - Let's assume we toss the coin five times: H, T, T, H, H



$$L(\theta) = \theta\cdot(1-\theta)\cdot(1-\theta)\cdot\theta\cdot\theta$$
$$= \theta^5 - 2\theta^4 + \theta^3$$
$$\frac{L(\theta)}{\partial\theta} =$$
$$\frac{L(\theta)}{\partial\theta} =$$

Multinomial discrete features

- Generalization of binary probabilities
- We desire now to have features that can take more than just 2 discrete values, which are mutually exclusive (i.e., only one possible at a time like rolling a dice).
- Formally:
  - A feature x that can assume W discrete values.
  - 1-in-W representation of x: Vector $\mathbf{x}$ of W binary variables x_j with $\sum_{j=1}^{W} x_j = 1$
- **Example**: ($W = 6$): $x = 2$ is represented by $\mathbf{x} = (0, 1, 0, 0, 0, 0)^T$
- Using $p(x = j) = p(x_j = 1) = \mu_j$ results for likelihood in:

$$p(\mathbf{x} \mid C^k, \boldsymbol{\mu}_k) = \prod_{j=1}^{W} \mu_{kj}^{x_j} \quad \text{or} \quad p(x \mid C^k, \boldsymbol{\mu}_k) = \begin{cases} \mu_{k1} & \text{for} \quad x = 1 \\ \mu_{k2} & \text{for} \quad x = 2 \\ \vdots & \\ \mu_{kW} & \text{for} \quad x = W \end{cases}$$

- For each class the parameter vector $\boldsymbol{\mu}_k$ has to be determined

$$\rightarrow \boldsymbol{\theta}_K = \boldsymbol{\mu}_K \qquad \text{subject to the constraint} \quad \sum_{j=1}^{W} \mu_{kj} = 1$$

Multinomial discrete features: Training

- Here again: Maximum likelihood estimation of $\boldsymbol{\mu}_k$
- In this case, one has to consider the constraint

$$\sum_{j=1}^{W} \mu_{kj} = 1$$

- Result (Derivation see Bishop, 2006): $\mu_{kj} = \dfrac{m_{kj}}{N_k}$

with m_{kj} ... Number of training samples for the class C^k that have the feature value j

N_k ... Total number of training samples for class C^k

Continuous features

- Frequent assumption: Multivariate normal distribution

$$p\left(\mathbf{x} \mid C^k\right) = \frac{1}{(2\pi)^{D/2} \cdot \left\|\mathbf{\Sigma}_k\right\|^{1/2}} \cdot e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \cdot \mathbf{\Sigma}_k^{-1} \cdot (\mathbf{x}-\boldsymbol{\mu}_k)}$$

- Motivation: Central limit theorem (the sum of many random variables is approximately normally distributed).

- Prerequisite: The class C^k only corresponds to one cluster in feature space

- Grey values are considered as continuous features.

- For each class C^k the mean value $\boldsymbol{\mu}_k$ and the covariance matrix $\mathbf{\Sigma}_k$ have to be determined → $\boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \mathbf{\Sigma}_k)$

Normal distribution: Training

- Given: N_k independent training samples x_{ik} for the class C^k
- Wanted: Parameters $\boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \mathbf{\Sigma}_k)$ for C^k

- Determination by **maximum likelihood estimation**:

$$p\left(\mathbf{x}_{1k} \middle| \theta_k\right) \cdot \ldots \cdot p\left(\mathbf{x}_{Nk} \middle| \theta_k\right) \Rightarrow \max$$

- **Log-Likelihood**: $\sum_i \ln p\left(\mathbf{x}_{1k} \middle| \theta_k\right) \Rightarrow \max$

$$\ln p\left(\mathbf{x}_{ik} \mid \boldsymbol{\theta}_k\right) = -\frac{D}{2}\ln(2\pi) - \frac{1}{2}\ln\left(\left\|\mathbf{\Sigma}_k\right\|\right) - \frac{1}{2}\left(\mathbf{x}_{ik} - \boldsymbol{\mu}_k\right)^T \cdot \mathbf{\Sigma}_k^{-1} \cdot \left(\mathbf{x}_{ik} - \boldsymbol{\mu}_k\right)$$

and, therefore, $\sum_i \ln p\left(\mathbf{x}_{ik} \mid \boldsymbol{\theta}_k\right) =$

$$= N \cdot \left[-\frac{D}{2}\ln(2\pi) - \frac{1}{2}\ln\left(\left\|\mathbf{\Sigma}_k\right\|\right)\right] - \frac{1}{2}\sum_i \left(\mathbf{x}_{ik} - \boldsymbol{\mu}_k\right)^T \cdot \mathbf{\Sigma}_k^{-1} \cdot \left(\mathbf{x}_{ik} - \boldsymbol{\mu}_k\right)$$

Normal distribution: Training

- Maximum Likelihood estimation: $\sum_i \ln p\left(\mathbf{x}_{1k} \middle| \theta_k\right) \Rightarrow \max$

- Derivative of $\sum_i \ln p\left(\mathbf{x}_{1k} \middle| \theta_k\right)$ by $\boldsymbol{\mu}_k$ must be 0:

$$\frac{\partial\left[\sum_i \ln p(\mathbf{x}_{ik} \mid \boldsymbol{\theta}_k)\right]}{\partial \boldsymbol{\mu}_k} = \sum_i \left[\boldsymbol{\Sigma}_k^{-1} \cdot (\mathbf{x}_{ik} - \boldsymbol{\mu}_k)\right] = 0$$

and, therefore,

$$\sum_i (\mathbf{x}_{ik} - \boldsymbol{\mu}_k) = \sum_i \mathbf{x}_{ik} - \sum_i \boldsymbol{\mu}_k = \sum_i \mathbf{x}_{ik} - N_k \cdot \boldsymbol{\mu}_k = 0$$

$\Rightarrow$ Result for $\boldsymbol{\mu}_k$: $\boldsymbol{\mu}_k = \frac{1}{N_k} \cdot \sum_i \mathbf{x}_{ik}$

Normal distribution: Training

- Solution for $\boldsymbol{\Sigma}_k$ is more complicated; again, the derivatives of the log-likelihood by the elements of $\boldsymbol{\Sigma}_k$ have to vanish.

- Result: $\boldsymbol{\Sigma}_k^{ML} = \frac{1}{N_k} \cdot \sum_i (\mathbf{x}_{ik} - \boldsymbol{\mu}_k) \cdot (\mathbf{x}_{ik} - \boldsymbol{\mu}_k)^T$

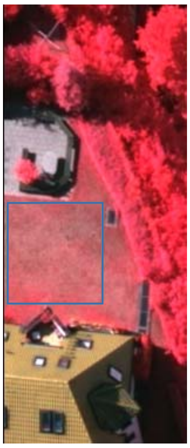
- Caution: While the ML-estimation of $\boldsymbol{\mu}_k$ is **unbiased, this is not the case for** $\boldsymbol{\Sigma}_k^{ML}$ **!**

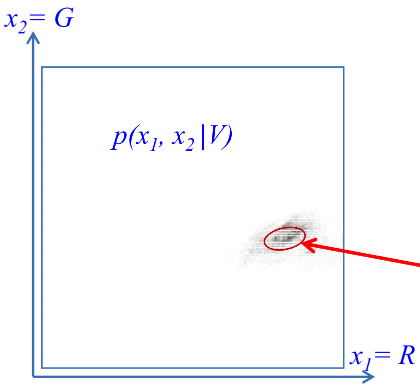
- Unbiased estimation: $\boldsymbol{\Sigma}_k = \frac{1}{N_k - 1} \cdot \sum_i (\mathbf{x}_{ik} - \boldsymbol{\mu}_k) \cdot (\mathbf{x}_{ik} - \boldsymbol{\mu}_k)^T$

- Bayesian estimation: $p(\theta_k)$ corresponds to regularization.

Normal Distribution: Example I

- Aerial image with training area for „vegetation“ (V)
  (87 x 85 = 7395 pixels)





$$\boldsymbol{\mu}_V = \begin{pmatrix} 230.2 \\ 110.5 \end{pmatrix}$$

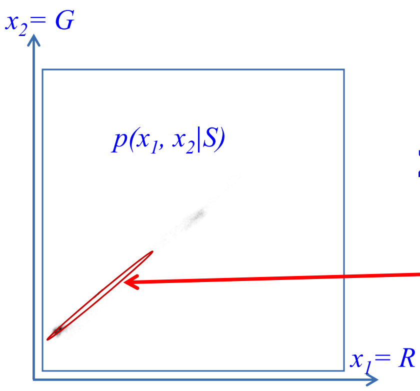
$$\boldsymbol{\Sigma}_V = \begin{pmatrix} 79.4 & 39.2 \\ 39.2 & 255.7 \end{pmatrix}$$

1σ - Ellipse

Good approximation by normal distribution

Normal Distribution: Example II

- Aerial image with training area for „street“ (S)
  (49 x 102 = 4998 Pixel)



$$\boldsymbol{\mu}_S = \begin{pmatrix} 45.1 \\ 58.4 \end{pmatrix}$$

$$\boldsymbol{\Sigma}_S = \begin{pmatrix} 2496.0 & 2076.1 \\ 2076.1 & 1736.4 \end{pmatrix}$$

1σ - Ellipse

Poor approximation by a normal distribution

because there are multiple clusters (shadow / sun)

Classes with multiple clusters

- Option 1: Splitting of a "thematic class" into several sub-classes
  - For example, *street:* → "street with shadow"
    → "street without shadow"
  - Each of these sub-classes corresponds to a single cluster in feature space → can be modelled by a normal distribution
  - Extra effort for the definition of the training data because the user must provide training samples for all sub-classes.
- Option 2: Automatic separation of the training data of a class into multiple clusters and estimation of the parameters of the individual clusters.

Gaussian mixture model

- In the case of N_j clusters, every cluster is described by a normal distribution.
- The total probability density is obtained from the weighted sum of the components :

$$p\left(\mathbf{x} \mid C^k\right) = \sum_{j=1}^{N_j} \pi_j \cdot N\left(\mathbf{x} \mid \boldsymbol{\mu}_{kj}, \boldsymbol{\Sigma}_{kj}\right)$$

with π_j ... Mixture coefficient for cluster j, corresponding to the prior probability for j

$\boldsymbol{\mu}_{kj}$ ... Mean value for cluster j

$\boldsymbol{\Sigma}_{kj}$ ... Covariance matrix for cluster j

$N(x|\boldsymbol{\mu}_{kj}, \boldsymbol{\Sigma}_{kj})$ ... Probability density of the normal distribution for cluster j

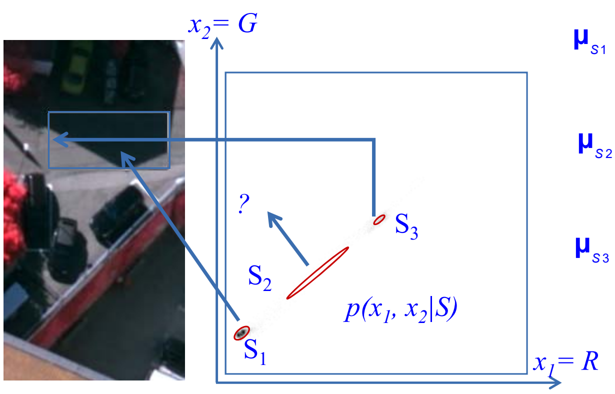
Gaussian mixture model: Training

- Parameters to be estimated: π_j, $\boldsymbol{\mu}_{kj}$, $\boldsymbol{\Sigma}_{kj}$ (1 set per cluster)
- Training of the mixture model requires cluster analysis of the feature space
  → unsupervised classification
- Closed estimation of the parameters is not possible.
- Method: „Expectation Maximization“ (EM)

  → see lecture “Unsupervised Classification“
- In general, EM requires the number of clusters N_j to be known in advance.

Gaussian mixture model: Example

- Aerial image with training area for „street“ (S)
  (49 x 102 = 4998 Pixel)



EM with three clusters

$$\boldsymbol{\mu}_{S1} = \begin{pmatrix} 13.4 \\ 32.4 \end{pmatrix} \quad \boldsymbol{\Sigma}_{S1} = \begin{pmatrix} 10.6 & 4.1 \\ 4.1 & 5.7 \end{pmatrix}$$

$$\boldsymbol{\mu}_{S2} = \begin{pmatrix} 65.3 \\ 73.3 \end{pmatrix} \quad \boldsymbol{\Sigma}_{S2} = \begin{pmatrix} 1559.6 & 1440.5 \\ 1440.5 & 1349.8 \end{pmatrix}$$

$$\boldsymbol{\mu}_{S3} = \begin{pmatrix} 129.3 \\ 128.5 \end{pmatrix} \quad \boldsymbol{\Sigma}_{S3} = \begin{pmatrix} 26.6 & 18.4 \\ 18.4 & 19.3 \end{pmatrix}$$

$$\pi_{S1} = 0.669 \quad \pi_{S2} = 0.105 \quad \pi_{S3} = 0.226$$

Good approximation by three components

Likelihood: Discussion

- Assumption of a normal distribution is often justified due to the central limit theorem.

- With inhomogeneous feature vectors (e.g. characteristics of data from different sensors) or discrete features one must make different assumptions.

- Assumption of a normal distribution is not justified for distributions having multiple clusters → mixture models

  - Example: streets in the shadow or in the sun correspond to different clusters in feature space.

- In many cases, one tries to avoid explicit modelling of probability densities

  → discriminative methods

Contents

- Theorem of Bayes
- Modelling of the likelihood function
  - Non-parametric techniques
  - Parametric techniques
- Modelling of the prior probability
- Discussion

Types of priors

- Origin of the priors $p(C^k)$:

  1) **From experiments**, e.g. in the case of sequential data: the prior for the classification at time *t* depends on the state at time *t-1*.

  2) **"Uninformed" / subjective:** from prior knowledge (... from whichever source)

- These two types of prior information are modelled in different ways.

Priors from Experiments: Maximum likelihood

- Requirement: the prior distribution should have the same algebraic form as the likelihood function →

- Example: Estimation of the parameter μ of a Bernoulli distribution with

$$p(x) = \mu^x \cdot (1 - \mu)^{(1-x)}$$

  - N experiments
    - in n_+ cases the result is "1"
    - in n_- cases the result is "0"
    - with: $n_+ + n_- = N$

  → Maximum Likelihood estimation: $\mu = n_+ / N$
  Can lead to overfitting → prior for μ?

Priors from Experiments: Bayesian estimation

- Bayesian estimation of μ: $p(\mu \mid n_+) \propto p(n_+ \mid \mu) \cdot p(\mu)$
- $p(n_+ \mid \mu)$ follows a binomial distribution:

$$p(n_+ \mid \mu) = \frac{N!}{n_+! \cdot (N - n_+)!} \mu^{n_+} \cdot (1-\mu)^{N-n_+}$$

- Prior distribution for μ?
  - Conjugate prior: *Beta distribution* with hyperparameters a,b:

$$p(\mu) = p(\mu \mid a,b) = \frac{\Gamma(a+b)}{\Gamma(a) \cdot \Gamma(b)} \cdot \mu^{a-1} \cdot (1-\mu)^{b-1}$$

- Resulting posterior:

$$p(\mu \mid n_+) \propto p(n_+ \mid \mu) \cdot p(\mu) \propto \mu^{n_+ + a-1} \cdot (1-\mu)^{N-n_+ + b-1}$$

Priors from Experiments: Bayesian estimation

- Resulting posterior:

$$p(\mu \mid n_+) \propto p(n_+ \mid \mu) \cdot p(\mu) \propto \mu^{n_+ + a-1} \cdot (1-\mu)^{N-n_+ + b-1}$$

- Interpretation:
  - $a - 1$ ... The number of trials with $x = 1$ from “earlier experiments“ which formed the basis of the prior.
  - $b - 1$ ... The number of trials with $x = 0$ from “earlier experiments“ which formed the basis of the prior.
- Simplifies the processing of sequential data.

Priors from Experiments

- Conjugate priors for other distributions:

Likelihood	Parameter	Conjugate prior	Hyper-parameter	Posterior parameter
Binomial	μ	Beta	a,b	$a+n_+$, $b+(N-n_+)$
Multinomial	$\boldsymbol{\mu}$ ($\Sigma\mu_i = 1$)	Dirichlet	$\mathbf{a}$	a_i+n_{i+}
Normal, σ known	$\boldsymbol{\mu}$	Normal	μ_0, σ_0^2	$\dfrac{\mu_0/\sigma_0^2 + \sum x_i/\sigma^2}{1/\sigma_0^2 + 1/\sigma^2}$
Normal, μ known	w (Precision)	Gamma	α, β	$\alpha+n/2$, $\beta+1/2\ \Sigma(x_i-\mu)^2$

Application: Generation of synthetic data

- We desire to train a Bayesian classier based on synthetic data, which is nothing else than to derive the (artificial) evidence:

$$p(\mathbf{x}) = \sum_k p\left(\mathbf{x} \middle| C^k\right) \cdot p\left(C^k\right)$$

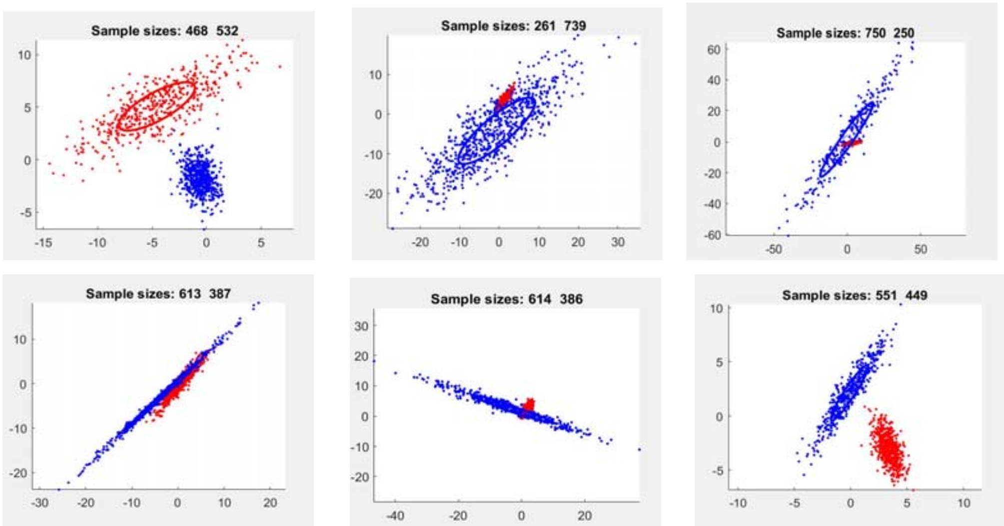
- For example, we look at a binary decision that is governed by Gaussian likelihood functions (embedded into a 2D feature space $\mathbf{x}$), thus:

$$p(\mathbf{x}) = N_1(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \cdot p\left(C^1\right) + N_2(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \cdot \left(1 - p\left(C^1\right)\right)$$

- Hence, we need priors for following parameters, which luckily can be at least coarsely be narrowed down to certain range due to **prior knowledge**:
  - $p(C^1)$: Beta distribtion (a, b)
  - $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$:
  - Σ_i or precision $\mathbf{W}_i := \Sigma_i^{-1}$: Gamma distribtion of major axis (w_1, w_2) and uniform distribtion of orientation angle θ in $[0,\pi]$.

Generation of synthetic data: Example

- Set of samples: $N = 1000$,
- Prior probabilities: $p(C^1) \sim 0.5$,
- Coordinates of cluster centers: $\mu_i \sim 1$, $\sigma_{\mu i} \sim 3$,
- Precision or Covariance of centers, respectively: w_1, w_2: $\mu_{wi} \sim 2$, $\sigma_{wi} \sim 3$



Uninformed priors

- A priori probabilities from minimal additional information
- Subjective priors (without measurements / experiments)

  → Principle of Maximum Entropy (ME):

$$p_{ME} = \arg\max_p \int_x -p(x)\log_2 p(x)\,dx$$

- Prior knowledge concerning the value range or moments of the distribution can be used to formulate of constraints for p_{ME}.

Uninformed Priors

- Example for ME-Priors:
  - Known value range with $a \leq x \leq b$: $$\int_{x=a}^{b} p(x)\,dx = 1$$

    → Uniform distribution in the interval (a,b)
    → Also applies for $(-\infty, +\infty)$ → in this case: **ML classification!**

  - Known expected value m, $x \geq 0$: $$\int_x x \cdot p(x)\,dx = m$$

    → Exponential distribution: $p(x) = \frac{1}{m} \cdot e^{-\frac{x}{m}}$

  - Known expected value m, known variance s^2:

    $$\int_x x \cdot p(x)\,dx = m \qquad \int_x (x-m)^2 \cdot p(x)\,dx = s^2$$

    → Normal distribution $N(\mu, \sigma^2)$

Contents

- Theorem of Bayes
- Modelling of the likelihood function
  - Non-parametric techniques
  - Parametric techniques
- Modelling of the prior probability
- Discussion

Bayesian classification: Discussion I

- Bayesian classification (and extensions) has many applications.
- There are many variants depending on the models used for the individual components.
- Bayesian classification delivers optimal results **if**
  - The assumptions about the likelihood function and the priors are correct.
  - The training data are representative for the classes.
  - There are enough training data to estimate the parameters of the models reliably.
- Problems occur when one of these assumptions is not justified…

Bayesian classification: Discussion II

- Examples of problems:
  - Assumption: the assumptions about the likelihood function and the priors are correct

    → Possible problem: unknown / wrong number of clusters for one or more classes in feature space.
  - Assumption: The training data are representative

    → Possible problem: training data only for objects in the sun, not for objects in the shadow.
  - Assumption: There are enough training data

    → Possible problem: not enough training data → reliable determination of the parameters may be impossible

Bayesian classification: Discussion III

- There is no mechanism to take into account uncertainties in the probabilities.

→ If the requirements are not fulfilled, Bayesian classification may yield suboptimal results.

- How to describe the quality of the results?
- How to determine the priors?
- Modelling the distribution of the data may require more parameters and, therefore, more training data than direct models of the posterior distribution

→ Discriminative methods: Only the class boundaries have to be learned

Literature

- Bishop, C. : Pattern Recognition and Machine Learning. 1st edition, Springer, New York, USA, 2006.
- Duda, R. O., Hart, P. E., Stork, D. G.: Pattern Classification. 2nd edition, Wiley & Sons, New York, USA, 2001.
- Klein, L. A.: Data and sensor fusion: a tool for information assessment and decision making. SPIE Optical Engineering, Bellingham, WA, USA, 2004.
- Förstner, W., 2012: Probabilistic data analysis using graphical models. Tutorial, lecture notes, ISPRS Congress Melbourne.