BAYESian Decision Theory and Parametric Density Estimation

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

Summer term 2025

Find all slides on GitHub (DaWe1992/Applied_ML_Fundamentals)

Lecture Overview

Agenda for this Unit

Section:

BAYESian Decision Theory

Introduction
Problem Definition
BAYES' Theorem and its components
Error Minimization: BAYES optimal Classifiers
Risk Minimization

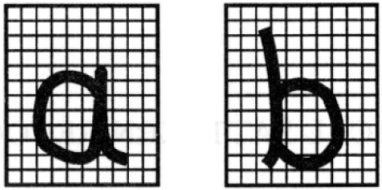
Statistical Methods

- Statistical methods assume that the process that 'generates' the data is governed by the **rules of probability**
- The data is understood to be a set of **random samples** from some underlying **probability distribution**
- This is the reason for the name **statistical machine learning**

The basic assumption about how the data is generated is always there, even if you don't see a single probability distribution!

Running Example: Optical Character Recognition (OCR)

A labeled dataset contains hand-written examples of two letters $\mathfrak{a}$ and $\mathfrak{b}$:



Our Goal: Classify new letters so as to minimize the probability of a wrong classification!

Problem Definition

- This is a **binary classification problem** involving the two classes $\mathcal{C}_1 := \mathfrak{a}$ and $\mathcal{C}_2 := \mathfrak{b}$
- Let $\boldsymbol{x}'$ be the feature vector of an unknown example
- Given $\boldsymbol{x}'$, we want to compute the probability of the classes $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively:

$$p(\mathcal{C}_1|\boldsymbol{x}') = ??? \qquad p(\mathcal{C}_2|\boldsymbol{x}') = ??? \tag{1}$$

Problem: We cannot compute these probabilities directly from the dataset, because we do not observe the feature vector x' in the training dataset!

BAYES' Theorem

- We can make use of **BAYES' theorem** to compute the relevant probabilities
- This theorem is one of the **most important formulas** used in machine learning (It gives rise to techniques collectively known as **BAYESian machine learning**)

BAYES' theorem:

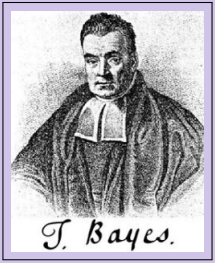
$$\overbrace{p(\mathcal{C}_k|\boldsymbol{x}')}^{\textcircled{1}} = \frac{\overbrace{p(\boldsymbol{x}'|\mathcal{C}_k)}^{\textcircled{2}} \cdot \overbrace{p(\mathcal{C}_k)}^{\textcircled{3}}}{\underbrace{p(\boldsymbol{x}')}_{\textcircled{4}}} = \frac{p(\boldsymbol{x}'|\mathcal{C}_k) \cdot p(\mathcal{C}_k)}{\sum_{j=1}^{K} p(\boldsymbol{x}'|\mathcal{C}_j) \cdot p(\mathcal{C}_j)} \tag{2}$$

Portrait: THOMAS BAYES

THOMAS BAYES (ca. 1701 – 7 April 1761) was an English statistician, philosopher and Presbyterian minister who is known for formulating a specific case of the theorem that bears his name: **BAYES' theorem**.

BAYES never published what would become his most famous accomplishment; his notes were edited and published posthumously by RICHARD PRICE.

(Wikipedia)



BAYES' Theorem Components

- The components of BAYES' theorem in equation (2) are:
  1. **Class posterior probability** (this is what we actually want to compute)
  2. **Class conditional probability**
  3. **Class prior probability**
  4. **Normalization constant**
- We will have a closer look into these components on the following slides.

Notation: In the following we will interpret features as random variables denoted by F_m $(1 \leqslant m \leqslant M)$.

In our example F_1 could be the number of black pixels and F_2 the height-width ratio, etc.

Class Conditional Probabilities

- Let $\boldsymbol{x}' := \left(x_1', x_2', \ldots, x_M'\right)^\top$ be a feature vector we wish to classify
- The **class conditional probability** of the feature vector $\boldsymbol{x}'$ given the class $\mathcal{C}_k$, $k \in \{1, 2\}$, is formally written as:

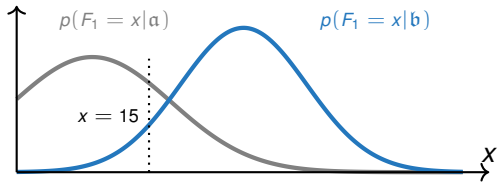
$$p(\boldsymbol{x}'|\mathcal{C}_k) = p(F_1 = x_1' \cap F_2 = x_2' \cap \ldots \cap F_M = x_M'|\mathcal{C}_k) \in [0, 1] \tag{3}$$

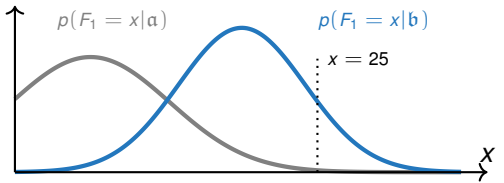
Unlike the class posterior probability, we can easily estimate the class conditional probabilities from a given labeled dataset (with some assumptions)!

(Later we shall see how this is done)

Class Conditional Probabilities (Ctd.)

We assume each image is described by one feature only: $F_1 := \#\ \texttt{black pixels}$





It is more likely to observe the value $F_1 = 15$ for instances of class $\mathfrak{a}$, while it is more likely to observe the value $F_1 = 25$ for instances of class $\mathfrak{b}$.

Class Prior Probabilities

- The **class prior probability** is equivalent to a **prior belief** in the class label, i. e. **before** taking any feature contributions into account
- These probabilities could be given by

$$p(\mathfrak{a}) := 0.75, \qquad p(\mathfrak{b}) := 0.25$$

- Class $\mathfrak{a}$ is in general more probable than class $\mathfrak{b}$ *(at least in English texts)*

Unlike the class posterior probability, we can easily estimate the class prior probabilities from a given labeled dataset!

A Priori versus a Posteriori

A Priori

A **belief** or conclusion **based on assumptions** or reasoning of some sort rather than actual experience or empirical evidence.

Before actually encountering, experiencing, or observing a fact.

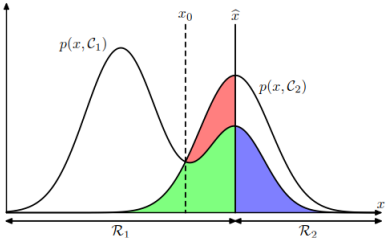
A Posteriori

A fact, belief, or argument that is **based on actual experience**, experiment, or observation.

Error Minimization

$$p(err) = p(x \in \mathcal{R}_1 \cap \mathcal{C} = \mathcal{C}_2) + p(x \in \mathcal{R}_2 \cap \mathcal{C} = \mathcal{C}_1)$$

$$= \overbrace{\int_{\mathcal{R}_1} p(x|\mathcal{C}_2) \cdot p(\mathcal{C}_2)\,\mathrm{d}x}^{\textbf{red} + \textbf{green} \text{ area}} + \overbrace{\int_{\mathcal{R}_2} p(x|\mathcal{C}_1) \cdot p(\mathcal{C}_1)\,\mathrm{d}x}^{\textbf{blue} \text{ area}}$$



- The **red** area is reducible, **blue** and **green** areas are not due to class overlap!
- $\widehat{x}$ is a suboptimal decision boundary, x_0 is the optimal one

cf. BISHOP.2006, page 40

BAYES optimal Classifiers

To minimize the classification error, decide for class $\mathcal{C}_1$ over $\mathcal{C}_2$ if:

$$
\begin{aligned}
& p(\mathcal{C}_1|\boldsymbol{x}') > p(\mathcal{C}_2|\boldsymbol{x}') \\
\iff \quad & \frac{p(\boldsymbol{x}'|\mathcal{C}_1) \cdot p(\mathcal{C}_1)}{p(\boldsymbol{x}')} > \frac{p(\boldsymbol{x}'|\mathcal{C}_2) \cdot p(\mathcal{C}_2)}{p(\boldsymbol{x}')} \\
\iff \quad & p(\boldsymbol{x}'|\mathcal{C}_1) \cdot p(\mathcal{C}_1) > p(\boldsymbol{x}'|\mathcal{C}_2) \cdot p(\mathcal{C}_2) \\
\iff \quad & \frac{p(\boldsymbol{x}'|\mathcal{C}_1)}{p(\boldsymbol{x}'|\mathcal{C}_2)} > \frac{p(\mathcal{C}_2)}{p(\mathcal{C}_1)}
\end{aligned}
\tag{4}
$$

BAYES optimal Classifiers (Ctd.)

- Any classification model obeying rule (4) is called a **BAYES optimal classifier**
- Such classifiers minimize the misclassification rate

Minimizing the misclassification rate may not always be the best approach as we will see on the next slides!

Error Minimization is not equivalent to Risk Minimization

- **False positives** and **false negatives** might be associated with different costs
- Some classical examples include:
  - **Smoke detector:**
    - If there is a fire, we must make sure to detect it
    - If there is not, an occasional false alarm may be acceptable
  - **Medical diagnosis:**
    - If the patient is sick, we have to detect the disease
    - If they are healthy, it can be okay to classify them as sick (conduct further tests)

Minimizing the error is not necessarily equal to minimizing the risk!

Expected Loss

- The key idea is to **incorporate the costs** connected to a misclassification into our decision rule
- For this we define a loss function

$$\ell(\mathcal{C}_i|\mathcal{C}_k) =: \ell_{ik}$$

which returns the costs of errenously deciding for class $\mathcal{C}_i$ over class $\mathcal{C}_k$

- The **expected loss (risk)** of making a decision for class $\mathcal{C}_i$ is then defined according to

$$R(\mathcal{C}_i|\boldsymbol{x}') := \sum_{k=1}^{K} \ell(\mathcal{C}_i|\mathcal{C}_k) p(\mathcal{C}_k|\boldsymbol{x}') \tag{5}$$

Risk Minimization

- We consider the binary case
- We therefore have **two possibilities**: Deciding for class $\mathcal{C}_1$ or class $\mathcal{C}_2$
- According to (5) we get:

$$R(\mathcal{C}_1|\boldsymbol{x}') = \ell_{11} p(\mathcal{C}_1|\boldsymbol{x}') + \ell_{12} p(\mathcal{C}_2|\boldsymbol{x}')$$
$$R(\mathcal{C}_2|\boldsymbol{x}') = \ell_{21} p(\mathcal{C}_1|\boldsymbol{x}') + \ell_{22} p(\mathcal{C}_2|\boldsymbol{x}')$$

- The values of $\ell_{11}, \ell_{12}, \ell_{21}, \ell_{22}$ are **hyperparameters**. Usually we set $\ell_{ii} = 0$

New decision rule: Decide for class $\mathcal{C}_1$ if $R(\mathcal{C}_2|\boldsymbol{x}') > R(\mathcal{C}_1|\boldsymbol{x}')$

Risk Minimization (Ctd.)

$$R(\mathcal{C}_2|\boldsymbol{x}') > R(\mathcal{C}_1|\boldsymbol{x}')$$

$$\iff \quad \ell_{21}p(\mathcal{C}_1|\boldsymbol{x}') + \ell_{22}p(\mathcal{C}_2|\boldsymbol{x}') > \ell_{11}p(\mathcal{C}_1|\boldsymbol{x}') + \ell_{12}p(\mathcal{C}_2|\boldsymbol{x}')$$

$$\iff \quad (\ell_{21} - \ell_{11})p(\mathcal{C}_1|\boldsymbol{x}') > (\ell_{12} - \ell_{22})p(\mathcal{C}_2|\boldsymbol{x}')$$

$$\iff \quad \frac{\ell_{21} - \ell_{11}}{\ell_{12} - \ell_{22}} > \frac{p(\mathcal{C}_2|\boldsymbol{x}')}{p(\mathcal{C}_1|\boldsymbol{x}')} = \frac{p(\boldsymbol{x}'|\mathcal{C}_2)p(\mathcal{C}_2)}{p(\boldsymbol{x}'|\mathcal{C}_1)p(\mathcal{C}_1)}$$

$$\iff \quad \frac{p(\boldsymbol{x}'|\mathcal{C}_1)}{p(\boldsymbol{x}'|\mathcal{C}_2)} > \frac{\ell_{12} - \ell_{22}}{\ell_{21} - \ell_{11}} \frac{p(\mathcal{C}_2)}{p(\mathcal{C}_1)}$$

Error Minimization as a special Case of Risk Minimization

$$\frac{p(\boldsymbol{x}'|\mathcal{C}_1)}{p(\boldsymbol{x}'|\mathcal{C}_2)} > \frac{\ell_{12} - \ell_{22}}{\ell_{21} - \ell_{11}} \frac{p(\mathcal{C}_2)}{p(\mathcal{C}_1)} \tag{6}$$

- We directly see that **error minimization is a special case of risk minimization**
- To obtain (4) from (6), we set

$$\ell_{ik} := \begin{cases} 0 & \text{if } i = k \\ 1 & \text{if } i \neq k \end{cases} \tag{7}$$

- This loss function is called **0-1 loss**

Section:

(Multinomial) Naïve BAYES

Assumptions and Algorithm
Maximum Likelihood Estimation for the binomial Distribution
A simple Example
LAPLACE Smoothing

Introduction

- For now we consider **categorical data** only
- We want to compute $p(\mathcal{C}_k|\boldsymbol{x'})$ according to (2)

$$p(\mathcal{C}_k|\boldsymbol{x'}) = \frac{p(\boldsymbol{x'}|\mathcal{C}_k) \cdot p(\mathcal{C}_k)}{p(\boldsymbol{x'})} \tag{8}$$

- We notice that we can omit the normalization constant $p(\boldsymbol{x'})$ in the denominator **(why?)**, so that

$$p(\mathcal{C}_k|\boldsymbol{x'}) \propto p(\boldsymbol{x'}|\mathcal{C}_k) \cdot p(\mathcal{C}_k) \tag{9}$$

(The symbol $\propto$ means 'is proportional to')

A naïve Assumption

- In equation (3) we have seen how to compute

$$p(\boldsymbol{x'}|\mathcal{C}_k) = p(F_1 = x_1' \cap F_2 = x_2' \cap \ldots \cap F_M = x_M'|\mathcal{C}_k)$$

- Estimating this probability is rather cumbersome *(and needs an exponential amount of data)*
- We therefore assume the features to be **pairwise conditionally independent (PCI)** given the class label **(this is a naïve assumption)**
- **Recall:** Two random variables $\mathcal{A}$ and $\mathcal{B}$ are said to be independent if $p(\mathcal{A} \cap \mathcal{B}) = p(\mathcal{A}) \cdot p(\mathcal{B})$

A naïve Assumption (Ctd.)

We use the chain rule for probabilities and the independence assumption **(PCI)** to rewrite equation (3):

$$
\begin{aligned}
p(\boldsymbol{x}'|\mathcal{C}_k) &= p(F_1 = x_1' \cap F_2 = x_2' \cap F_3 = x_3' \cap \ldots \cap F_M = x_M'|\mathcal{C}_k) \\
&= p(F_1 = x_1'|\mathcal{C}_k) \cdot p(F_2 = x_2'|\mathcal{C}_k \cap F_1 = x_1') \cdot \ldots \\
&= p(F_1 = x_1'|\mathcal{C}_k) \cdot p(F_2 = x_2'|\mathcal{C}_k) \cdot p(F_3 = x_3'|\mathcal{C}_k) \cdot \ldots \\
&= \prod_{m=1}^{M} p(F_m = x_m'|\mathcal{C}_k) \qquad (10)
\end{aligned}
$$

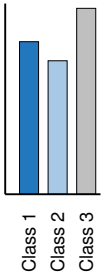
Naïve Bayes Decision Rule

Given a new instance $\boldsymbol{x'} = (x'_1, x'_2, \ldots, x'_M)^\top$ and a set of classes $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$, predict the most probable class $\mathcal{C}^{\text{MAP}}$ *(maximum aposteriori)* according to the rule

$$
\begin{aligned}
\mathcal{C}^{\text{MAP}} &:= \arg\max_{\mathcal{C}_k \in \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}} p(\mathcal{C}_k | \boldsymbol{x'}) \\
&= \arg\max_{\mathcal{C}_k \in \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}} p(\boldsymbol{x'} | \mathcal{C}_k) \cdot p(\mathcal{C}_k) \\
&= \arg\max_{\mathcal{C}_k \in \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}} \left(\prod_{m=1}^{M} p(F_m = x'_m | \mathcal{C}_k) \right) \cdot p(\mathcal{C}_k) \qquad (11)
\end{aligned}
$$

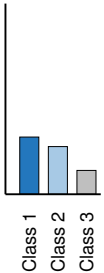
Visualization: From Prior Probability to Posterior Probability

Apriori Probabilities $p(\mathcal{C}_k)$

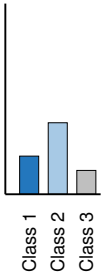


x

Feature Contributions $p(F_1 = x_1'|\mathcal{C}_k)$

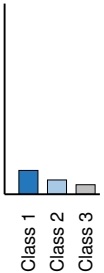


x ... x $p(F_M = x_M'|\mathcal{C}_k)$



=

Aposteriori Probabilities $p(\mathcal{C}_k|\boldsymbol{x}')$



How to estimate the Probabilities?

- At this point **we know how to compute the class posterior probability** from known class prior and class conditional probabilities
- However, we have not yet seen how we can obtain $p(\mathcal{C}_k)$ and $p(F_m = x'_m|\mathcal{C}_k)$ $(1 \leqslant m \leqslant M \text{ and } 1 \leqslant k \leqslant K)$ from a labeled training dataset

How to do this?

Determining these probabilities for **categorical data** is rather simple!

How to estimate the Probabilities? (Ctd.)

Count Count's advice:

How to estimate the Probabilities? (Ctd.)

- **Solution**: Simply count the occurrences:

$$p(\mathcal{C}_k) \approx \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\{y_n = \mathcal{C}_k\} \tag{12}$$

$$p(F_m = x'_m | \mathcal{C}_k) \approx \frac{\sum_{n=1}^{N} \mathbf{1}\left\{x_m^{(n)} = x'_m \wedge y_n = \mathcal{C}_k\right\}}{\sum_{n=1}^{N} \mathbf{1}\{y_n = \mathcal{C}_k\}} \tag{13}$$

- $\mathbf{1}\{bool\}$ is the **indicator function** which returns 1 if *bool* is true, and 0 otherwise
- **Example:** $\mathbf{1}\{1 + 1 = 2\} = 1$, but $\mathbf{1}\{3 = 2\} = 0$

Maximum Likelihood Estimation (MLE)

We can justify the 'counting approach' by applying a method called **maximum likelihood estimation (MLE)**

- In statistics, this method is used to estimate the unknown parameters $\boldsymbol{\theta}$ of a given probability distribution from data
- In the following we verify formula (12) for the binary case, i. e. we consider the two classes $\mathcal{C}_1$ and $\mathcal{C}_2$ (the K-class case for $K > 2$ can be shown analogously)
- Formula (13) can be shown analogously

MLE for the Class Prior Probability

- Let θ be the unknown probability of observing class $\mathcal{C}_1$ *(also known as the probability of success)*
- Suppose further that $\mathcal{C}_1$ occurs k times in a training dataset comprising N **independent training instances**
- The probability of observing k occurrences of class $\mathcal{C}_1$ in the dataset is then distributed according to a **binomial distribution:**

$$p(\theta) := b(k; N, \theta) = \binom{N}{k} \cdot \theta^k \cdot (1-\theta)^{N-k} \tag{14}$$

MLE for the Class Prior Probability (Ctd.)

- Our goal is to find θ so as to maximize the **likelihood function** $\rho(\theta)$ given by equation (14)
- As the logarithm is a strictly monotonous function, we can consider the **log-likelihood function** *(this will make maths more convenient)*

$$\begin{aligned}\mathcal{L}(\theta) := \log\big(\rho(\theta)\big) &= \log\left[\binom{N}{k} \cdot \theta^k \cdot (1-\theta)^{N-k}\right] \\ &= \log\binom{N}{k} + k \cdot \log(\theta) + (N-k) \cdot \log(1-\theta) \qquad (15)\end{aligned}$$

- Remember the logarithmic rules

MLE for the Class Prior Probability (Ctd.)

We compute the first-order and second-order derivatives of $\mathcal{L}(\theta)$ with respect to the unknown parameter θ:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\theta}\mathcal{L}(\theta) &= \frac{\mathrm{d}}{\mathrm{d}\theta}\left[\log\binom{N}{k} + k\cdot\log(\theta) + (N-k)\cdot\log(1-\theta)\right] \\ &= \frac{k}{\theta} - \frac{N-k}{1-\theta} \end{aligned} \tag{16}$$

$$\frac{\mathrm{d}^2}{\mathrm{d}\theta^2}\mathcal{L}(\theta) = \frac{\mathrm{d}}{\mathrm{d}\theta}\left[\frac{k}{\theta} - \frac{N-k}{1-\theta}\right] = -\frac{k}{\theta^2} - \frac{N-k}{(1-\theta)^2} \tag{17}$$

MLE for the Class Prior Probability (Ctd.)

- Setting (16) to zero yields:

$$\frac{k}{\theta} - \frac{N-k}{1-\theta} \stackrel{!}{=} 0 \iff \boxed{\theta = \frac{k}{N}} =: p(\mathcal{C}_1) \tag{18}$$

- Since the second-order derivative (17) $< 0 \; \forall \; \theta \in [0, 1]$ and $k < N \in \mathbb{N}$, we know that we have found a **maximum of the likelihood function**
- Finally, we define

$$p(\mathcal{C}_2) := 1 - \frac{k}{N} = \frac{N-k}{N}$$

Summary: Maximum Likelihood Estimation

Prerequisite: The data is **independent and identically distributed (i. i. d.)**

1. Choose a probability distribution you want to fit to the data
2. Set up the likelihood function $\rho(\theta)$
3. *Optional:* Apply the logarithm to obtain the log-likelihood function $\mathcal{L}(\theta)$ and apply the logarithmic rules to rewrite the log-likelihood function
4. Differentiate with respect to the parameter θ you want to optimize, set the derivative to zero, and subsequently solve for θ
5. Check for a maximum using the second-order derivative

The Multinomial Distribution

- We can use a similar approach in the non-binary case by considering the **multinomial distribution** instead of the binomial distribution:

$$m(k_1, \ldots, k_L; N, p_1, \ldots, p_L) := \binom{N}{k_1, k_2, \ldots, k_L} \prod_{\ell=1}^{L} p_\ell^{k_\ell}, \tag{19}$$

with $k_1 + \ldots + k_L = N$

- The **multinomial coefficient** is given by

$$\binom{N}{k_1, k_2, \ldots, k_L} := \frac{N!}{k_1! \cdot k_2! \cdot \ldots \cdot k_L!} \tag{20}$$

Example Dataset

New instance: x'

Outlook	= sunny
Temperature	= cool
Humidity	= high
Wind	= strong

What is its class?

Outlook	Temperature	Humidity	Wind	PlayGolf
sunny	hot	high	weak	**no**
sunny	hot	high	strong	**no**
overcast	hot	high	weak	**yes**
rainy	mild	high	weak	**yes**
rainy	cool	normal	weak	**yes**
rainy	cool	normal	strong	**no**
overcast	cool	normal	strong	**yes**
sunny	mild	high	weak	**no**
sunny	cool	normal	weak	**yes**
rainy	mild	normal	weak	**yes**
sunny	mild	normal	strong	**yes**
overcast	mild	high	strong	**yes**
overcast	hot	normal	weak	**yes**
rainy	mild	high	strong	**no**
sunny	cool	high	strong	**???**

Let's estimate the Probabilities from the Dataset!

Class prior probabilities:

$$p(\texttt{PlayGolf} = \text{yes}) = 9/14 = 0.64$$
$$p(\texttt{PlayGolf} = \text{no}) = 5/14 = 0.36$$

Class conditional probabilities:

$$p(\texttt{Outlook} = \text{sunny}|\texttt{PlayGolf} = \text{yes}) = 2/9 = 0.22$$
$$p(\texttt{Temp.} = \text{cool}|\texttt{PlayGolf} = \text{yes}) = 3/9 = 0.33$$
$$p(\texttt{Humidity} = \text{high}|\texttt{PlayGolf} = \text{yes}) = 3/9 = 0.33$$
$$p(\texttt{Wind} = \text{strong}|\texttt{PlayGolf} = \text{yes}) = 3/9 = 0.33$$

$$p(\texttt{Outlook} = \text{sunny}|\texttt{PlayGolf} = \text{no}) = 3/5 = 0.60$$
$$p(\texttt{Temp.} = \text{cool}|\texttt{PlayGolf} = \text{no}) = 1/5 = 0.20$$
$$p(\texttt{Humidity} = \text{high}|\texttt{PlayGolf} = \text{no}) = 4/5 = 0.80$$
$$p(\texttt{Wind} = \text{strong}|\texttt{PlayGolf} = \text{no}) = 3/5 = 0.60$$

Computation of the Class Posterior Probabilities

$$
\begin{aligned}
p(\texttt{PlayGolf} = \text{yes}|\boldsymbol{x}') &= p(\text{sunny}|\text{yes})p(\text{cool}|\text{yes})p(\text{high}|\text{yes})p(\text{strong}|\text{yes})p(\text{yes}) \\
&= 0.22 \cdot 0.33 \cdot 0.33 \cdot 0.33 \cdot 0.64 \\
&= \boxed{0.0053}
\end{aligned}
$$

$$
\begin{aligned}
p(\texttt{PlayGolf} = \text{no}|\boldsymbol{x}') &= p(\text{sunny}|\text{no})p(\text{cool}|\text{no})p(\text{high}|\text{no})p(\text{strong}|\text{no})p(\text{no}) \\
&= 0.60 \cdot 0.20 \cdot 0.80 \cdot 0.60 \cdot 0.36 \\
&= \boxed{0.0207}
\end{aligned}
$$

Classification: $\mathcal{C}^{\text{MAP}} = \text{no}$ *(so no golf today...)*

Scaling the Output

But wait! These probabilities don't sum up to one!?!?

- This is because we dropped the normalization term $p(\boldsymbol{x}')$
- **Scaling** can fix this *(it does **not** change the prediction)*:

$$p(\text{yes}|\boldsymbol{x}')_{norm} := \frac{0.0053}{0.0053 + 0.0207} = \mathbf{0.204}$$

$$p(\text{no}|\boldsymbol{x}')_{norm} := \frac{0.0207}{0.0053 + 0.0207} = \mathbf{0.796}$$

LAPLACE Smoothing

What if there is a feature value $F = v^\star$ in the test data, not seen during training?
Problem: The probability $p(F = v^\star|\mathcal{C}_k) = 0$ for all classes. **Therefore, the class posterior probabilities become zero as well...**

Solution: **LAPLACE smoothing** adds a tiny probability to unknown feature values:

$$p(F_m = x'_m|\mathcal{C}_k) \approx \frac{\sum_{n=1}^{N} \mathbf{1}\left\{x_m^{(n)} = x'_m \wedge y_n = \mathcal{C}_k\right\} + 1}{\sum_{n=1}^{N} \mathbf{1}\left\{y_n = \mathcal{C}_k\right\} + K} \tag{21}$$

Section:

Gaussian Naïve Bayes

Handling of continuous Data
Maximum Likelihood Estimation for the (univariate) Gaussian Distribution
Maximum Likelihood Estimation for the (multivariate) Gaussian Distribution
Generative vs. Discriminative Models

Handling of continuous Features

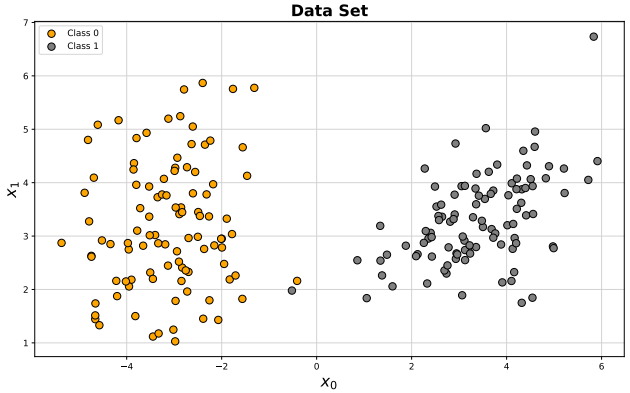
- Let us now turn towards **continuous features**
- Multinomial naïve BAYES cannot estimate useful probabilities in this case **(why?)**

How do we get the probabilities in the continuous case?

- The class prior probabilities $p(\mathcal{C}_k)$ are still easy to compute
- The estimation of the class conditional probabilities $p(\boldsymbol{x}|\mathcal{C}_k)$ is trickier
- We have to consider a continuous probability distribution

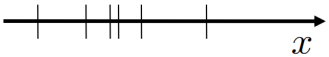
Goal: Model $p(\boldsymbol{x}|\mathcal{C}_k)$ using a **GAUSSian distribution**

Training Data Example

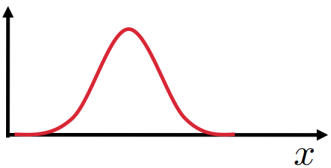


General Approach (1-dimensional Data)

- Let $\boldsymbol{X}_k := \left\{x_n\right\}_{n=1}^{N_k}$ be the set of data points belonging to class $\mathcal{C}_k$



- Estimate $p(x|\mathcal{C}_k)$ using a fixed parametric form (here: GAUSSian distribution)



GAUSSian Distribution

GAUSSian distribution:

$$p(x|\mathcal{C}_k) := \mathcal{N}\left(x; \mu_k, \sigma_k^2\right) := \frac{1}{\sqrt{2\pi\sigma_k^2}} \cdot \exp\left(-\frac{(x-\mu_k)^2}{2\sigma_k^2}\right) \tag{22}$$

Remarks:

- The GAUSSian distribution is governed by two parameters: μ (mean), σ^2 (variance)
- We learn a separate GAUSSian distribution for each class $\mathcal{C}_k$
- Therefore we have to estimate the mean μ_k and the variance σ_k^2 for each class

Learning the Parameters using MLE

Goal: Estimate the parameters $\boldsymbol{\theta}^k := \left(\mu_k, \sigma_k^2\right)^\top$ from the data $\boldsymbol{X}_k$ using MLE

Assume data to be **i. i. d.**: Two random variables $\mathcal{A}$ and $\mathcal{B}$ are **independent**, if

$$P(\mathcal{A} \leqslant \alpha \cap \mathcal{B} \leqslant \beta) = P(\mathcal{A} \leqslant \alpha) \cdot P(\mathcal{B} \leqslant \beta) \qquad \forall\, \alpha, \beta \in \mathbb{R} \tag{23}$$

Two random variables $\mathcal{A}$ and $\mathcal{B}$ are **identically distributed**, if

$$P(\mathcal{A} \leqslant \alpha) = P(\mathcal{B} \leqslant \alpha) \qquad \forall\, \alpha \in \mathbb{R} \tag{24}$$

Setting up the Likelihood Function

❶ **Likelihood function:** We use the i. i. d. assumption to rewrite the likelihood function

$$p\left(\boldsymbol{X}_k; \boldsymbol{\theta}^k\right) = p\left(x_1 \cap x_2 \cap \ldots \cap x_{N_k}; \boldsymbol{\theta}^k\right)$$

$$\overset{(23)}{=} p\left(x_1; \boldsymbol{\theta}^k\right) \cdot p\left(x_2; \boldsymbol{\theta}^k\right) \cdot \ldots \cdot p\left(x_{N_k}; \boldsymbol{\theta}^k\right)$$

$$\overset{(24)}{=} \prod_{n=1}^{N_k} p\left(x_n; \boldsymbol{\theta}^k\right) \tag{25}$$

Derivation of the Log-Likelihood Function

❷ Again we apply the logarithm to obtain the **log-likelihood function**:

$$\mathcal{L}\left(\boldsymbol{\theta}^k\right) := \log\left(\prod_{n=1}^{N_k} p\left(x_n; \boldsymbol{\theta}^k\right)\right)$$

$$= \sum_{n=1}^{N_k} \log\left(p\left(x_n; \boldsymbol{\theta}^k\right)\right) \qquad \textit{(What have we done in this step?)}$$

$$\overset{(22)}{=} \sum_{n=1}^{N_k} \log\left[\frac{1}{\sqrt{2\pi\sigma_k^2}} \cdot \exp\left(-\frac{\left(x_n - \mu_k\right)^2}{2\sigma_k^2}\right)\right] \tag{26}$$

Derivation of the Log-Likelihood Function (Ctd.)

❸ We further rewrite the equation using the logarithmic rules:

$$(26) = \sum_{n=1}^{N_k} \left[\log(1) - \frac{1}{2} \log(2\pi\sigma_k^2) - \frac{1}{2\sigma_k^2} \left(x_n - \mu_k\right)^2 \right]$$

$$= \boxed{-\frac{N_k}{2} \log(2\pi\sigma_k^2) - \frac{1}{2\sigma_k^2} \sum_{n=1}^{N_k} \left(x_n - \mu_k\right)^2} \tag{27}$$

Next, we have to differentiate equation (27) with respect to μ_k and σ_k^2
(we will demonstrate this for the mean μ_k on the following slides)

Maximum Likelihood Solution for the Mean

❹ We compute the partial derivative of equation (27) with respect to μ_k:

$$\frac{\partial}{\partial \mu_k} \mathcal{L}\left(\boldsymbol{\theta}^k\right) = \frac{\partial}{\partial \mu_k} \left[-\frac{N_k}{2} \log(2\pi\sigma_k^2) - \frac{1}{2\sigma_k^2} \sum_{n=1}^{N_k} \left(x_n - \mu_k\right)^2 \right]$$

$$= -\frac{1}{2\sigma_k^2} \sum_{n=1}^{N_k} \frac{\partial}{\partial \mu_k} \left[\left(x_n - \mu_k\right)^2 \right]$$

$$= \frac{1}{\sigma_k^2} \sum_{n=1}^{N_k} \left(x_n - \mu_k\right) = \frac{1}{\sigma_k^2} \left(\sum_{n=1}^{N_k} x_n - N_k \mu_k \right)$$

Maximum Likelihood Solution for the Mean (Ctd.)

❺ We set the derivative to zero and solve for μ_k:

$$\frac{1}{\sigma_k^2}\left(\sum_{n=1}^{N_k} x_n - N_k\mu_k\right) \stackrel{!}{=} 0 \iff \sum_{n=1}^{N_k} x_n = N_k\mu_k$$

$$\iff \mu_k = \frac{1}{N_k}\sum_{n=1}^{N_k} x_n = \overline{x}, \tag{28}$$

where $\overline{x} := \frac{1}{N_k}\sum_{n=1}^{N_k} x_n$ is the arithmetic mean of the data

Maximum Likelihood Solution for the Mean (Ctd.)

❻ The second-order derivative of equation (27) with respect to μ_k is given by:

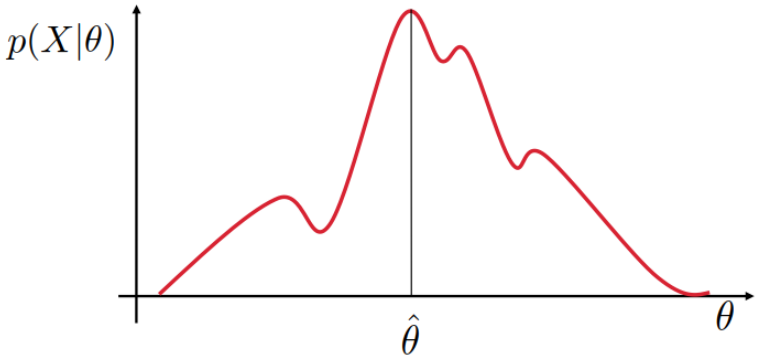
$$\frac{\partial^2}{\partial \mu_k^2} \mathcal{L}\left(\boldsymbol{\theta}^k\right) = -\frac{N_k}{\sigma_k^2} < 0 \; \forall \; \mu_k \in \mathbb{R} \tag{29}$$

Therefore, we have found a maximum and

$$\mu_k^{\mathrm{ML}} := \frac{1}{N_k} \sum_{n=1}^{N_k} x_n$$

is the **maximum likelihood solution** for the mean of the Gaussian distribution.

Visualization: Maximization of the Likelihood



Maximum Likelihood Solution for Gaussian naïve Bayes

Estimate the mean and the variance for each class according to:

$$\mu_k^{\mathrm{ML}} := \frac{1}{N_k} \sum_{n=1}^{N_k} x_n \quad \text{and} \quad \left(\sigma_k^2\right)^{\mathrm{ML}} \overset{(*)}{:=} \frac{1}{N_k} \sum_{n=1}^{N_k} \left(x_n - \mu_k^{\mathrm{ML}}\right)^2 \tag{30}$$

(*) This is left to the reader as an exercise!

Using these estimates we can use Bayes' theorem to predict the most probable class labels *(see next slide for the general procedure!)*

Summary: GAUSSian Naïve BAYES

Prerequisite: The data is **independent and identically distributed (i. i. d.)**

1. Estimate the class **prior probabilities** $p(\mathcal{C}_k)$ using the previously introduced counting approach *(as in the discrete case)*
2. Compute the **class conditional probabilities** $p(x|\mathcal{C}_k)$:
   - Estimate μ_k and σ_k^2 according to the formulas in (30)
   - Compute $p(x|\mathcal{C}_k)$ using the GAUSSian distribution (22)
3. Compute the **class posterior probabilities** $p(\mathcal{C}_k|x)$ using BAYES' theorem (2)
4. Decide for the class which maximizes the posterior probability

Biased vs. Unbiased Estimators

- We can show that $\left(\sigma_k^2\right)^{\text{ML}}$ is a **biased estimator**
- On average it **underestimates** the true variance of the data
- The **empirical variance** is an **unbiased estimator**

$$\left(\sigma_k^2\right)^{\text{Emp}} := \frac{1}{N_k - 1} \sum_{n=1}^{N_k} \left(x_n - \mu_k^{\text{ML}}\right)^2 \tag{31}$$

- Please find a derivation $\Rightarrow$ here
- We can use both estimators in practical applications

Multivariate GAUSSian Distribution

- The solution above is for 1-dimensional data
- **Question: What if we have more dimensions?**

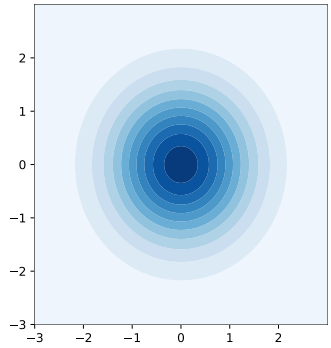
Multivariate GAUSSian distribution:

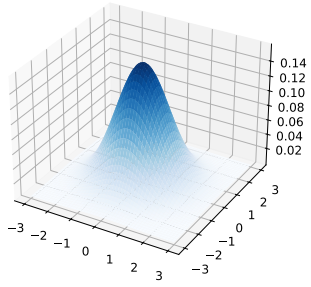
$$\mathcal{N}_D\left(\boldsymbol{x}; \boldsymbol{\mu}^k, \boldsymbol{\Sigma}_k\right) := \frac{1}{\sqrt{(2\pi)^D \det\left(\boldsymbol{\Sigma}_k\right)}} \cdot \exp\left(-\frac{1}{2}\left(\boldsymbol{x} - \boldsymbol{\mu}^k\right)^\top \boldsymbol{\Sigma}_k^{-1}\left(\boldsymbol{x} - \boldsymbol{\mu}^k\right)\right)$$

The multivariate GAUSSian distribution is governed by the two parameters $\boldsymbol{\mu} \in \mathbb{R}^D$ *(mean)* and $\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}$ *(covariance matrix)*

Visualization of a multivariate GAUSSian Distribution

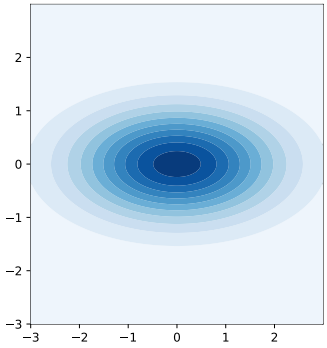
$$\boldsymbol{\Sigma} = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix}$$

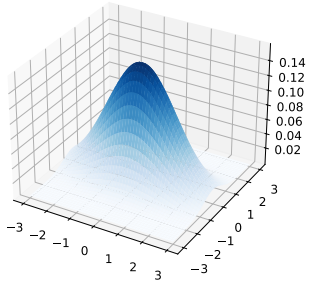




Visualization of a multivariate Gaussian Distribution (Ctd.)

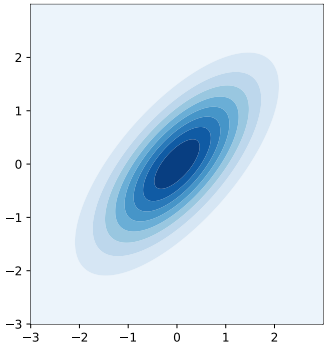
$$\boldsymbol{\Sigma} = \begin{pmatrix} 2.0 & 0.0 \\ 0.0 & 0.5 \end{pmatrix}$$

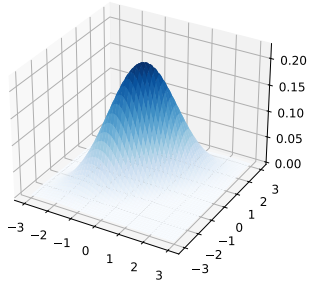




Visualization of a multivariate GAUSSian Distribution (Ctd.)

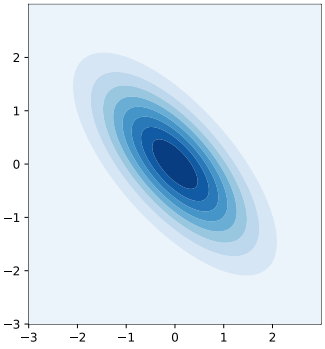
$$\boldsymbol{\Sigma} = \begin{pmatrix} 1.0 & 0.7 \\ 0.7 & 1.0 \end{pmatrix}$$

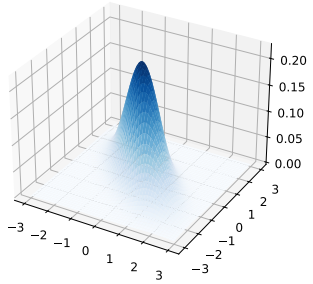




Visualization of a multivariate Gaussian Distribution (Ctd.)

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1.0 & -0.7 \\ -0.7 & 1.0 \end{pmatrix}$$





Maximum Likelihood Estimation for the multivariate GAUSSian

- **Luckily, the results do not change**
- MLE for the mean $\boldsymbol{\mu}^k$:

$$\left(\boldsymbol{\mu}^k\right)^{\mathrm{ML}} := \frac{1}{N_k} \sum_{n=1}^{N_k} \boldsymbol{x}^n \tag{32}$$

- MLE for the covariance $\boldsymbol{\Sigma}_k$:

$$\boldsymbol{\Sigma}_k^{\mathrm{ML}} := \frac{1}{N_k} \sum_{n=1}^{N_k} \left(\boldsymbol{x}^n - \left(\boldsymbol{\mu}^k\right)^{\mathrm{ML}}\right) \left(\boldsymbol{x}^n - \left(\boldsymbol{\mu}^k\right)^{\mathrm{ML}}\right)^\top \tag{33}$$

GAUSSian naïve BAYES – Final Model

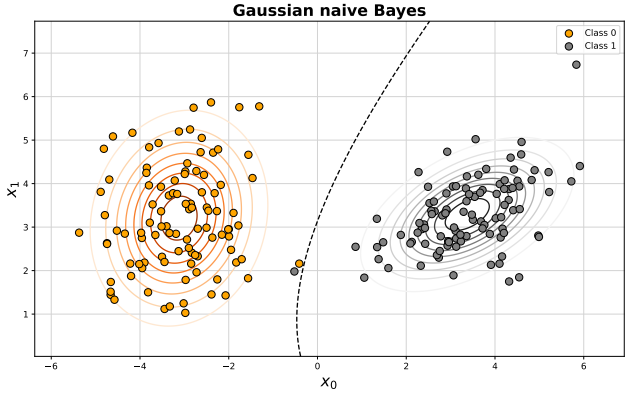
The multivariate GAUSSian model is given by

$$p(\mathcal{C}_k|\boldsymbol{x}') := \mathcal{N}_D\left(\boldsymbol{x}'; \boldsymbol{\mu}^k, \boldsymbol{\Sigma}_k\right) \cdot p(\mathcal{C}_k)$$

Remarks:

- $\mathcal{N}_D\left(\boldsymbol{x}'; \boldsymbol{\mu}^k, \boldsymbol{\Sigma}_k\right)$ is the multivariate GAUSSian distribution for class $\mathcal{C}_k$
- $\mathcal{N}_D\left(\boldsymbol{x}'; \boldsymbol{\mu}^k, \boldsymbol{\Sigma}_k\right)$ is still a scalar number!
- $p(\mathcal{C}_k)$ is the prior probability of class $\mathcal{C}_k$ *(as in the discrete case)*

Solution to the introductory Example



Some Remarks

- It is also conceivable to estimate only one covariance matrix $\boldsymbol{\Sigma}$ which is used for all classes *(instead of estimating one covariance matrix per class)*
- Suppose $y_n \in \{0, 1, 2, \ldots\}$:

$$\boldsymbol{\Sigma}^{\text{ML}} := \frac{1}{N} \sum_{n=1}^{N} \left(\boldsymbol{x}^n - \left(\boldsymbol{\mu}^{y_n} \right)^{\text{ML}} \right) \left(\boldsymbol{x}^n - \left(\boldsymbol{\mu}^{y_n} \right)^{\text{ML}} \right)^{\top} \tag{34}$$

- This approach leads to a **linear decision boundary**
- In the literature, GAUSSian Naïve BAYES is often referred to as **GAUSSian Discriminant Analysis (GDA)**

Generative vs. Discriminative Models

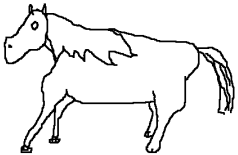
Generative Model – *The artist*

A **generative** algorithm models **how** the data was generated, means the **class conditional probability** distributions $p(\boldsymbol{x}|\mathcal{C}_k)$ and **the priors** $p(\mathcal{C}_k) \Rightarrow$ BAYES' theorem.

vs.

Discriminative Model – *The lousy painter*

A **discriminative** algorithm does not care about how the data was generated. It only knows **how to distinguish the classes**, and models $p(\mathcal{C}_k|\boldsymbol{x})$ directly.



Section:

Exponential Family

Exponential Family Distributions

Exponential family:

$$p(\boldsymbol{x};\boldsymbol{\eta}) = b(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^\top \boldsymbol{T}(\boldsymbol{x}) - a(\boldsymbol{\eta})\right\} \tag{35}$$

Legend:

$\boldsymbol{x}$	Data
$\boldsymbol{\eta}$	Natural parameter
$b(\boldsymbol{x})$	Base measure
$\boldsymbol{T}(\boldsymbol{x})$	Sufficient statistic
$a(\boldsymbol{\eta})$	log-partition function

Many distributions belong to the exponential family:

- BERNOULLI distribution
- GAUSSian distribution
- POISSON distribution
- Exponential distribution
- Beta and Gamma distributions

Example: Bernoulli Distribution

$$x \in \{0, 1\}$$

$$\begin{aligned}
\mathrm{Ber}(x; \mu) &= \mu^x (1-\mu)^{1-x} \\
&= \exp\left\{\log\left(\mu^x (1-\mu)^{1-x}\right)\right\} \\
&= \exp\left\{\log\left(\frac{\mu}{1-\mu}\right) x + \log(1-\mu)\right\}
\end{aligned}$$

$$b(x) = 1$$

$$T(x) = x$$

$$\eta = \log\left(\frac{\mu}{1-\mu}\right) \Leftrightarrow \mu = \frac{1}{1+e^{-\eta}}$$

$$\begin{aligned}
a(\eta) &= -\log(1-\mu) \\
&= -\log\left(1 - \frac{1}{1+e^{-\eta}}\right) \\
&= \log(1+e^{\eta})
\end{aligned}$$

Example: GAUSSian Distribution (fixed Variance)

Assume $\sigma^2 = 1$:

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

$$= \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{x^2}{2}\right\} \exp\left\{\mu x - \frac{1}{2}\mu^2\right\}$$

$$b(x) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{x^2}{2}\right\}$$

$$T(x) = x$$

$$\eta = \mu$$

$$a(\eta) = \frac{\mu^2}{2} = \frac{\eta^2}{2}$$

Properties of Exponential Family Distributions

Distributions belonging to the exponential family have some nice properties:

- The likelihood function with respect to $\boldsymbol{\eta}$ is **concave**, i. e. when doing maximum likelihood estimation, we know that there is **only one global optimum**
- Expectation and variance are easy to compute:

$$\mathbb{E}\{\boldsymbol{x}; \boldsymbol{\eta}\} = \frac{\partial}{\partial \boldsymbol{\eta}} a(\boldsymbol{\eta}) \tag{36}$$

$$\mathbb{V}\{\boldsymbol{x}; \boldsymbol{\eta}\} = \frac{\partial^2}{\partial \boldsymbol{\eta} \partial \boldsymbol{\eta}} a(\boldsymbol{\eta}) \tag{37}$$

Section:

Wrap-Up

Summary
Recommended Literature
Self-Test Questions
Lecture Outlook

Summary

- Important concepts: **Class conditional** and **class prior probabilities**
- Use **BAYES' theorem** to get the **class posteriors**
- **BAYES optimal classifier:** Decide for the most probable class
- Naïve BAYES assumes all features to be **pairwise conditionally independent**
- We can use **parametric models** to estimate the density of the data
- Such models assume a certain **parametric form**, e. g. a GAUSSian distribution
- **Maximum likelihood estimation (MLE)** is an important tool!
- The GAUSSian distribution allows us to work with **continuous features**

Recommended Literature

1. **Bayesian decision theory:**
   - [Bishop.2006], chapter 1.5
2. **Naïve Bayes:**
   - [Murphy.2006], chapters 3.1, 3.2, and 3.5
   - [Murphy.2006], chapter 4.1

(For free PDF versions, see list in GitHub readme!)

Self-Test Questions

1. What are class conditional probabilities?
2. What does BAYES *optimal* mean?
3. How can we incorporate prior knowledge about the class distribution into the classification?
4. What is the naïve assumption which naïve BAYES makes? When might this be a problem?
5. Explain what the term ‘maximum aposteriori’ means!
6. What is maximum likelihood estimation? How can you get the maximum likelihood estimate for a GAUSSian distribution?

What's next...?

	I	Machine Learning Introduction
	II	Optimization Techniques
	III	Bayesian Decision Theory
•	**IV**	Non-parametric Density Estimation
	V	Probabilistic Graphical Models
	VI	Linear Regression
	VII	Logistic Regression
	VIII	Deep Learning
	IX	Evaluation
	X	Decision Trees
	XI	Support Vector Machines
	XII	Clustering
	XIII	Principal Component Analysis
	XIV	Reinforcement Learning
	XV	Advanced Regression

Thank you very much for the attention!

*** * * Artificial Intelligence and Machine Learning * * ***

Topic: BAYESian Decision Theory and Parametric Density Estimation

Term: Summer term 2025

Contact:
Daniel Wehner, M.Sc.
SAP SE / DHBW Mannheim
daniel.wehner@sap.com

Do you have any questions?