*** Applied Machine Learning Fundamentals ***
Bayesian Decision Theory

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

Winter term 2023/2024

Find all slides on GitHub (DaWe1992/Applied_ML_Fundamentals)

Lecture Overview

Unit I	Machine Learning Introduction
Unit II	Mathematical Foundations
Unit III	**Bayesian Decision Theory**
Unit IV	Regression
Unit V	Classification I
Unit VI	Evaluation
Unit VII	Classification II
Unit VIII	Clustering
Unit IX	Dimensionality Reduction

Agenda for this Unit

1. Bayesian Decision Theory
2. (Multinomial) Naïve Bayes
3. Gaussian Naïve Bayes
4. Wrap-Up

Section:

Bayesian Decision Theory

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

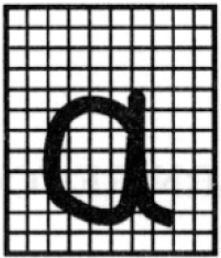
Statistical Methods

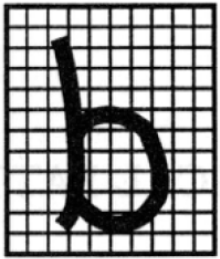
- Statistical methods assume that the process that 'generates' the data is governed by the **rules of probability**
- The data is understood to be a set of **random samples** from some underlying **probability distribution**
- This is the reason for the name **statistical machine learning**

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 $\mathfrak{b}$:





Goal: Classify a new letter so that the probability of a wrong classification is minimized

Problem Definition

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

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

However, we cannot compute these probabilities directly from the dataset, because we do not observe the feature vector x_q in the dataset!

Bayes' Theorem

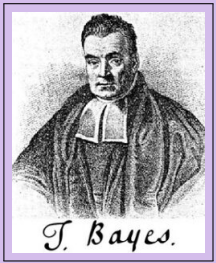
- We can make use of **Bayes' theorem** to compute the probabilities we are interested in
- This theorem is one of the most important formulas used in machine learning (It gives rise to techniques collectively known as **Bayesian machine learning**)

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

Bayes' Theorem (Ctd.)

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**.



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 $\mathcal{X}_j$ $(1 \leqslant j \leqslant m)$. In our example $\mathcal{X}_1$ could be the number of black pixels and $\mathcal{X}_2$ the height-width ratio, etc.

Class Conditional Probabilities

- Let $\boldsymbol{x}_q := (x_1, x_2, \ldots, x_m)^\intercal$ be the feature vector
- The **class conditional probability** of the feature vector $\boldsymbol{x}_q$ given the class $\mathcal{C}_k$, where $k \in \{1, 2\}$, is formally written as:

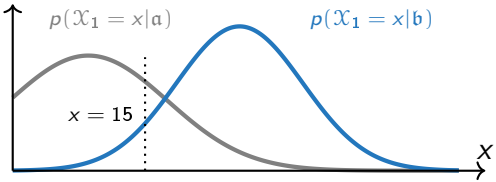
$$p(\boldsymbol{x}_q|\mathcal{C}_k) = p(\mathcal{X}_1 = x_1 \cap \mathcal{X}_2 = x_2 \cap \ldots \cap \mathcal{X}_m = x_m|\mathcal{C}_k) \in [0, 1] \quad (3)$$

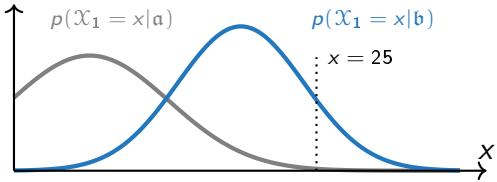
Unlike the class posterior probability, we can easily estimate the class conditional probabilities from a given labeled dataset (with some assumptions)!

(Later we shall see how this is done)

Class Conditional Probabilities (Ctd.)

We assume each image is described by one feature only: $\mathcal{X}_1 := \#$ black pixels





It is more probable to observe the feature value $\mathcal{X}_1 = 15$ for instances of class $\mathfrak{a}$, while it is more probable to observe the feature value $\mathcal{X}_1 = 25$ 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.

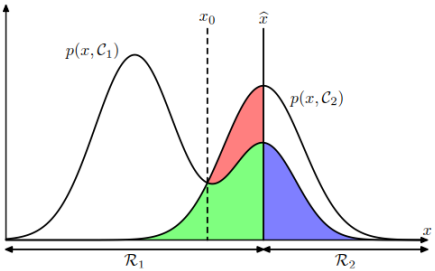
A Posteriori

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

Error Minimization

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

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



The red area is reducible, blue and green 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:

Any classification model obeying rule (4) on the right-hand side is called a **Bayes optimal classifier**!

$$
\begin{aligned}
& p(\mathcal{C}_1|x_q) > p(\mathcal{C}_2|x_q) \\
\iff \quad & \frac{p(x_q|\mathcal{C}_1) \cdot p(\mathcal{C}_1)}{p(x_q)} > \frac{p(x_q|\mathcal{C}_2) \cdot p(\mathcal{C}_2)}{p(x_q)} \\
\iff \quad & p(x_q|\mathcal{C}_1) \cdot p(\mathcal{C}_1) > p(x_q|\mathcal{C}_2) \cdot p(\mathcal{C}_2) \\
\iff \quad & \frac{p(x_q|\mathcal{C}_1)}{p(x_q|\mathcal{C}_2)} > \frac{p(\mathcal{C}_2)}{p(\mathcal{C}_1)} \qquad (4)
\end{aligned}
$$

Error Minimization is not equivalent to Risk Minimization

- False positives and false negatives might be associated to 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 (order 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

$$\ell(\mathcal{C}_i|\mathcal{C}_j) \equiv \ell_{ij}$$

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

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

$$R(\mathcal{C}_i|\boldsymbol{x}_q) := \sum_j \ell(\mathcal{C}_i|\mathcal{C}_j) p(\mathcal{C}_j|\boldsymbol{x}_q) \tag{5}$$

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:

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

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

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

Risk Minimization (Ctd.)

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

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

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

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

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

Error Minimization as a special Case of Risk Minimization

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

- We directly see that error minimization is a special case of risk minimization
- For this we set

$$\ell_{ij} := \begin{cases} 0 \text{ if } i = j \\ 1 \text{ if } i \neq j \end{cases}$$

in (6) to obtain the decision rule (4). 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 $p(\mathcal{C}_k|\boldsymbol{x}_q)$ according to Bayes' theorem

$$p(\mathcal{C}_k|\boldsymbol{x}_q) = \frac{p(\boldsymbol{x}_q|\mathcal{C}_k) \cdot p(\mathcal{C}_k)}{p(\boldsymbol{x}_q)} \tag{7}$$

- First of all, we can omit the normalization constant $p(\boldsymbol{x}_q)$ in the denominator (why?), so that

$$p(\mathcal{C}_k|\boldsymbol{x}_q) \propto p(\boldsymbol{x}_q|\mathcal{C}_k) \cdot p(\mathcal{C}_k) \tag{8}$$

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

A naïve Assumption

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

$$p(\boldsymbol{x}_q|\mathcal{C}_k) = p(\mathcal{X}_1 = x_1 \cap \mathcal{X}_2 = x_2 \cap \ldots \cap \mathcal{X}_m = x_m|\mathcal{C}_k)$$

- 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 independence assumption to rewrite equation (3):

$$
\begin{aligned}
p(\boldsymbol{x}_q|\mathcal{C}_k) &= p(\mathcal{X}_1 = x_1 \cap \mathcal{X}_2 = x_2 \cap \ldots \cap \mathcal{X}_m = x_m|\mathcal{C}_k) \\
&\overset{(\triangle)}{=} p(\mathcal{X}_1 = x_1|\mathcal{C}_k) \cdot p(\mathcal{X}_2 = x_2|\mathcal{C}_k \cap \mathcal{X}_1 = x_1) \cdot p(\mathcal{X}_3 = x_3|\mathcal{C}_k \cap \mathcal{X}_1 = x_1 \cap \mathcal{X}_2 = x_2) \cdot \ldots \\
&\overset{PCI}{=} p(\mathcal{X}_1 = x_1|\mathcal{C}_k) \cdot p(\mathcal{X}_2 = x_2|\mathcal{C}_k) \cdot p(\mathcal{X}_3 = x_3|\mathcal{C}_k) \cdot \ldots \\
&= \prod_{j=1}^{m} p(\mathcal{X}_j = x_j|\mathcal{C}_k) \qquad (9)
\end{aligned}
$$

In step $(\triangle)$ we have used the chain rule/product rule for probabilities

Naïve Bayes Decision Rule

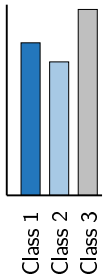
Given a new instance $\boldsymbol{x}_q = (x_1, x_2, \ldots, x_m)^\intercal$ and a finite set of classes $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$, predict the most probable class $\mathcal{C}_{MAP}$ (maximum aposteriori) according to the **naïve Bayes decision rule**:

$$\begin{aligned} \mathcal{C}_{MAP} &= \underset{\mathcal{C}_k \in \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}}{\arg\max} \; p(\mathcal{C}_k | \boldsymbol{x}_q) = \underset{\mathcal{C}_k \in \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}}{\arg\max} \; p(\boldsymbol{x}_q | \mathcal{C}_k) \cdot p(\mathcal{C}_k) \\ &= \underset{\mathcal{C}_k \in \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}}{\arg\max} \left(\prod_{j=1}^{m} p(\mathcal{X}_j = x_j | \mathcal{C}_k) \right) \cdot p(\mathcal{C}_k) \end{aligned} \tag{10}$$

Visualization: From Prior Probability to Posterior Probability

Apriori Probabilities

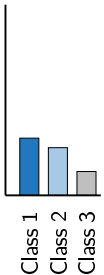
$p(\mathcal{C}_k)$



x

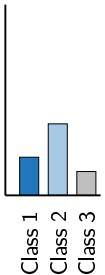
Feature Contributions

$p(\mathcal{X}_1 = x_1|\mathcal{C}_k)$



x ... x

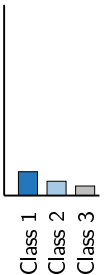
$p(\mathcal{X}_m = x_m|\mathcal{C}_k)$



=

Aposteriori Probabilities

$p(\mathcal{C}_k|\boldsymbol{x}_q)$



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 $p(\mathcal{X}_j = x_j|\mathcal{C}_k)$ 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

$$p(\mathcal{C}_k) \approx \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{y^{(i)} = \mathcal{C}_k\} \tag{11}$$

$$p(\mathcal{X}_j = x_j | \mathcal{C}_k) \approx \frac{\sum_{i=1}^{n} \mathbb{1}\{\mathcal{X}_j^{(i)} = x_j \wedge y^{(i)} = \mathcal{C}_k\}}{\sum_{i=1}^{n} \mathbb{1}\{y^{(i)} = \mathcal{C}_k\}} \tag{12}$$

- $\mathbb{1}\{bool\}$ is the **indicator function**
  (returns 1 if *bool* is true, 0 otherwise. E. g.: $\mathbb{1}\{1+1=2\} = 1$, but $\mathbb{1}\{3=2\} = 0$)

Maximum Likelihood Estimation

- We can justify the 'counting approach' using a method called **maximum likelihood estimation (MLE)**
- In statistics this method is used to estimate parameters $\boldsymbol{\vartheta}$ of a given probability distribution from data
- In the following we verify formula (11) for the binary case, i. e. we consider the two classes $\mathcal{C}_1$ and $\mathcal{C}_2$
- Formula (12) can be shown analogously

MLE for the Class Prior Probability

- Let ϑ be the unknown probability of observing class $\mathcal{C}_1$ (ϑ is called 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**

$$b(k|n,\vartheta) = \binom{n}{k} \cdot \vartheta^k \cdot (1-\vartheta)^{n-k} =: \rho_k(\vartheta) \tag{13}$$

MLE for the Class Prior Probability (Ctd.)

- Our goal is to find ϑ so as to maximize the **likelihood function** (13)
- As the logarithm is a strictly monotonous function, we can consider the **log-likelihood function** (this will make maths more convenient)

$$\log\big(p_k(\vartheta)\big) = \log\left[\binom{n}{k} \cdot \vartheta^k \cdot (1-\vartheta)^{n-k}\right]$$

$$\overset{(\triangle)}{=} \log\binom{n}{k} + k \cdot \log(\vartheta) + (n-k) \cdot \log(1-\vartheta) \qquad (14)$$

In step $(\triangle)$ we have used the logarithmic rules $\log(ab) = \log a + \log b$ and $\log(a^b) = b \cdot \log a$

MLE for the Class Prior Probability (Ctd.)

We compute the first-order and second-order derivatives of $\log\big(\rho_k(\vartheta)\big)$ with respect to ϑ:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\vartheta}\log\big(\rho_k(\vartheta)\big) &= \frac{\mathrm{d}}{\mathrm{d}\vartheta}\left[\log\binom{n}{k} + k\cdot\log(\vartheta) + (n-k)\cdot\log(1-\vartheta)\right] \\ &= \frac{k}{\vartheta} - \frac{n-k}{1-\vartheta} \qquad (15)\end{aligned}$$

$$\frac{\mathrm{d}^2}{(\mathrm{d}\vartheta)^2}\log\big(\rho_k(\vartheta)\big) = \frac{\mathrm{d}}{\mathrm{d}\vartheta}\left[\frac{k}{\vartheta} - \frac{n-k}{1-\vartheta}\right] = -\frac{k}{\vartheta^2} - \frac{n-k}{(1-\vartheta)^2} \qquad (16)$$

MLE for the Class Prior Probability (Ctd.)

- Setting (15) to zero yields:

$$\frac{k}{\vartheta} - \frac{n-k}{1-\vartheta} \stackrel{!}{=} 0 \iff \boxed{\vartheta = \frac{k}{n}} =: p(\mathcal{C}_1) \tag{17}$$

- Since the second-order derivative (16) < 0 $\forall \vartheta \in [0, 1]$ 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

1. Choose a probability distribution you want to fit to the data
2. Set up the likelihood function $\rho(\vartheta)$
3. *Optional:* Apply the logarithm to obtain the log-likelihood function $\log\big(\rho(\vartheta)\big)$ and apply the logarithmic rules to rewrite the log-likelihood function (this will usually make maths more convenient)
4. Differentiate for 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:

$$\binom{n}{k_1, k_2, \ldots, k_\ell} \prod_{j=1}^{\ell} p^{k_j} \qquad \text{with: } k_1 + \ldots + k_\ell = n \tag{18}$$

- Where the **multinomial coefficient** is given by

$$\binom{n}{k_1, k_2, \ldots, k_\ell} := \frac{n!}{k_1! \cdot k_2! \cdot \ldots \cdot k_\ell!} \tag{19}$$

Example Dataset

New instance: x_q

$$\begin{aligned} \texttt{Outlook} &= \text{sunny} \\ \texttt{Temperature} &= \text{cool} \\ \texttt{Humidity} &= \text{high} \\ \texttt{Wind} &= \text{strong} \end{aligned}$$

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:

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

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

Computation of the Class Posterior Probabilities

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

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

Classification: $\mathcal{C}_{MAP} = \text{no}$ (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 $p(\boldsymbol{x}_q)$
- Scaling can fix this:

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

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

- Scaling does **not** change the prediction

Laplace Smoothing

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

$$p(\mathcal{X}_j = x_j | \mathcal{C}_k) \approx \frac{\sum_{i=1}^{n} \mathbb{1}\{\mathcal{X}_j^{(i)} = x_j \wedge y^{(i)} = \mathcal{C}_k\} + 1}{\sum_{i=1}^{n} \mathbb{1}\{y^{(i)} = \mathcal{C}_k\} + K} \tag{20}$$

- Laplace smoothing adds a tiny probability to unknown feature values

Section:

Gaussian Naïve Bayes

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

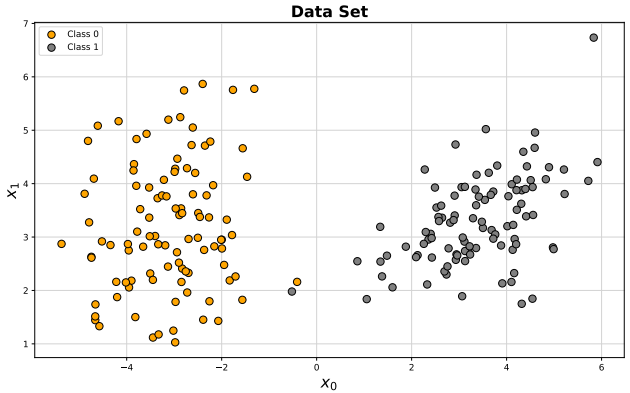
Handling of continuous Features

- Let us now assume **continuous features**
- Multinomial naïve Bayes can only be used for **discrete features** (why?)

How do we get the probabilities in the continuous case?

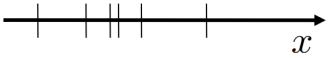
- The class prior probabilities $p(\mathcal{C}_k)$ are still easy to compute
- The estimation of the class conditional probabilities $p(\boldsymbol{x}_q|\mathcal{C}_k)$ is more involved
- Our goal is to model $p(\boldsymbol{x}|\mathcal{C}_k)$ using a **Gaussian distribution**

Training Data Example



General Approach (1-dimensional Data)

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



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

Example: Gaussian Distribution

- The **Gaussian distribution** is given by *(you should know it by heart!)*:

$$p(x_q|\mathcal{C}_k) := \mathcal{N}(x_q|\mu_k, \sigma_k^2) := \frac{1}{\sqrt{2\pi\sigma_k^2}} \cdot \exp\left(-\frac{(x_q - \mu_k)^2}{2\sigma_k^2}\right) \quad (21)$$

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

the Parameters using MLE

- Learning means estimating the parameters $\theta_k := \left\{\mu_k, \sigma_k^2\right\}$ given the data $\boldsymbol{X}_k$
- **Fundamental assumption:** The data is assumed to be **i. i. d.** (independent and identically distributed):
  - Two random variables $\mathcal{A}$ and $\mathcal{B}$ are independent, if

$$P(\mathcal{A} \leqslant \alpha \cap \mathcal{B} \leqslant \beta) = P(\mathcal{A} \leqslant \alpha) \cdot P(\mathcal{B} \leqslant \beta) \qquad \forall \alpha, \beta \in \mathbb{R} \tag{22}$$

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

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

Setting up the Likelihood Function

$$p(\boldsymbol{X}_k|\boldsymbol{\theta}_k) = p\big(x^{(1)} \cap x^{(2)} \cap \ldots \cap x^{(n_k)}|\boldsymbol{\theta}_k\big)$$

data is independent; refer to definition (22):

$$= p\big(x^{(1)}|\boldsymbol{\theta}_k\big) \cdot p\big(x^{(2)}|\boldsymbol{\theta}_k\big) \cdot \ldots \cdot p\big(x^{(n_k)}|\boldsymbol{\theta}_k\big)$$

data is identically distributed (follows the same Gaussian distribution); refer to definition (23):

$$= \prod_{i=1}^{n_k} p\big(x^{(i)}|\boldsymbol{\theta}_k\big) \tag{24}$$

Derivation of the Log-Likelihood Function

- Again we apply the logarithm to make maths more convenient
- The log-likelihood function is therefore given by:

$$\log\big(p(\boldsymbol{X}_k|\boldsymbol{\theta}_k)\big) = \log\left(\prod_{i=1}^{n_k} p\big(x^{(i)}|\boldsymbol{\theta}_k\big)\right)$$

$$= \sum_{i=1}^{n_k} \log\Big(p\big(x^{(i)}|\boldsymbol{\theta}_k\big)\Big) \qquad \textit{(What have we done in this step?)}$$

$$= \sum_{i=1}^{n_k} \log\left[\frac{1}{\sqrt{2\pi\sigma_k^2}} \cdot \exp\left(-\frac{(x_q-\mu_k)^2}{2\sigma_k^2}\right)\right] \tag{25}$$

Derivation of the Log-Likelihood Function (Ctd.)

We further rewrite the equation using the logarithmic rules:

$$\sum_{i=1}^{n_k} \log\left[\frac{1}{\sqrt{2\pi\sigma_k^2}} \cdot \exp\left(-\frac{\left(x^{(i)}-\mu_k\right)^2}{2\sigma_k^2}\right)\right] = \sum_{i=1}^{n_k}\left[\log(1) - \frac{1}{2}\log(2\pi\sigma_k^2) - \frac{1}{2\sigma_k^2}\left(x^{(i)}-\mu_k\right)^2\right]$$

$$= -\frac{n_k}{2}\log(2\pi\sigma_k^2) - \frac{1}{2\sigma_k^2}\sum_{i=1}^{n_k}\left(x^{(i)}-\mu_k\right)^2 \qquad (26)$$

As a next step we have to differentiate equation (26) 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

We compute the partial derivative w. r. t. μ_k:

$$\frac{\partial}{\partial \mu_k} \log\left(p(\boldsymbol{X}_k|\boldsymbol{\theta}_k)\right) = \frac{\partial}{\partial \mu_k}\left[-\frac{n_k}{2}\log(2\pi\sigma_k^2) - \frac{1}{2\sigma_k^2}\sum_{i=1}^{n_k}\left(x^{(i)} - \mu_k\right)^2\right]$$

$$= -\frac{1}{2\sigma_k^2}\sum_{i=1}^{n_k}\frac{\partial}{\partial \mu_k}\left[\left(x^{(i)} - \mu_k\right)^2\right]$$

$$= \frac{1}{\sigma_k^2}\sum_{i=1}^{n_k}\left(x^{(i)} - \mu_k\right) = \boxed{\frac{1}{\sigma_k^2}\left(\sum_{i=1}^{n_k} x^{(i)} - n_k\mu_k\right)}$$

Maximum Likelihood Solution for the Mean (Ctd.)

We set the derivative to zero and solve for μ_k:

$$\frac{1}{\sigma_k^2}\left(\sum_{i=1}^{n_k} x^{(i)} - n_k\mu_k\right) \stackrel{!}{=} 0 \iff \sum_{i=1}^{n_k} x^{(i)} = n_k\mu_k$$

$$\iff \mu_k = \frac{1}{n_k}\sum_{i=1}^{n_k} x^{(i)} = \overline{x} \tag{27}$$

(where $\overline{x} := \frac{1}{n_k}\sum_{i=1}^{n_k} x^{(i)}$ is the arithmetic mean of the data)

Maximum Likelihood Solution for the Mean (Ctd.)

The second-order derivative is given by:

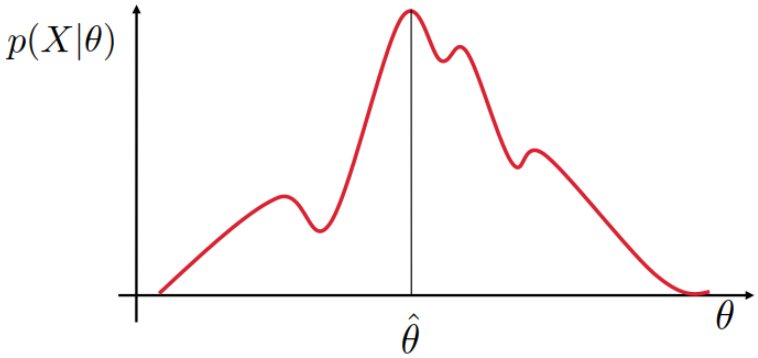
$$\frac{\partial^2}{(\partial \mu_k)^2} \log\big(p(\boldsymbol{X}_k|\boldsymbol{\theta}_k)\big) = -\frac{n_k}{\sigma_k^2} < 0 \ \forall \mu_k \in \mathbb{R} \tag{28}$$

Therefore, we have found a maximum and

$$\mu_k^{\mathrm{ML}} := \frac{1}{n_k} \sum_{i=1}^{n_k} x^{(i)}$$

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:

$$\mu_k^{\mathrm{ML}} := \frac{1}{n_k} \sum_{i=1}^{n_k} x^{(i)} \qquad \left(\sigma^2\right)^{\mathrm{ML}} \overset{(*)}{:=} \frac{1}{n_k} \sum_{i=1}^{n_k} \left(x^{(i)} - \mu_k^{\mathrm{ML}}\right)^2$$

- Using these probabilities we can use Bayes' theorem to predict the most probable class labels

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

Biased vs. Unbiased Estimators

- We can show that $\left(\sigma^2\right)^{\mathrm{ML}}$ is a **biased estimator**, i.e. on average this estimator **underestimates** the true variance of the data
- The **empirical variance** defined by

$$\left(\sigma^2\right)^{\mathrm{Emp}} := \frac{1}{n_k - 1} \sum_{i=1}^{n_k} \left(x^{(i)} - \mu_k^{\mathrm{ML}}\right)^2 \tag{29}$$

  is an **unbiased estimator**
- Please find a derivation $\Rightarrow$ here
- We can use both estimators in practical applications

Multivariate Case

- The solution above is for 1-dim. data; What if we have more dimensions?
- **Multivariate Gaussian distribution**:

$$\mathcal{N}_D(\boldsymbol{x}_q|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) := \frac{1}{\sqrt{(2\pi)^D \det(\boldsymbol{\Sigma}_k)}} \cdot \exp\left(-\frac{1}{2}(\boldsymbol{x}_q - \boldsymbol{\mu}_k)^\intercal \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_q - \boldsymbol{\mu}_k)\right) \quad (30)$$

- Luckily, the derivations don't change:

$$\boldsymbol{\mu}_k^{\mathrm{ML}} := \frac{1}{n_k}\sum_{i=1}^{n_k} \boldsymbol{x}^{(i)} \qquad \boldsymbol{\Sigma}_k^{\mathrm{ML}} := \frac{1}{n_k}\sum_{i=1}^{n_k} \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_k^{\mathrm{ML}}\right)\left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_k^{\mathrm{ML}}\right)^\intercal \quad (31)$$

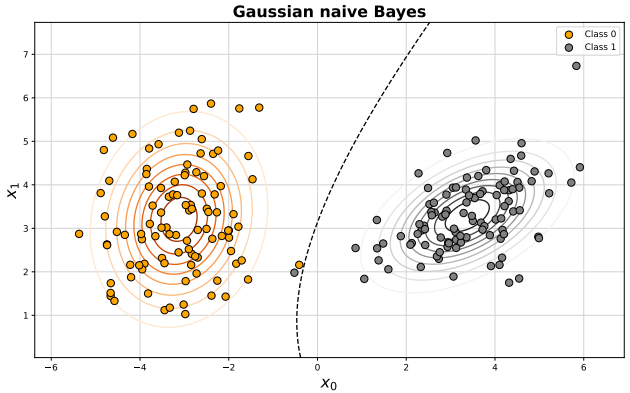
Gaussian naïve Bayes – Final Model

- The multivariate Gaussian model is given by

$$p(\mathcal{C}_k|\boldsymbol{x}_q) := \mathcal{N}_D(\boldsymbol{x}_q|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \cdot p(\mathcal{C}_k)$$

- **Please note:** $\mathcal{N}_D(\boldsymbol{x}_q|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ denotes the multivariate Gaussian distribution representing class $\mathcal{C}_k$; **This is still a scalar number!**
- **Please note:** $p(\mathcal{C}_k)$ is the prior probability of class $\mathcal{C}_k$ (as in the discrete case).

Solution to the introductory Example



Generative vs. Discriminative Models

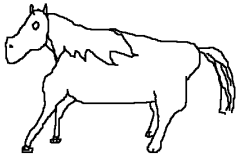
Generative Model

(The artist)

A **generative** algorithm models **how** the data was generated. **It models the class conditional probability distributions $p(\boldsymbol{x}_q|\mathcal{C}_k)$.**

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.**

Section:

Wrap-Up

Summary
Self-Test Questions
Lecture Outlook

Summary

- Important concepts: **Class conditional probabilities** 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. They assume a certain **parametric form**, e. g. a Gaussian distribution
- The Gaussian distribution allows us to work with **continuous features**

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 'maximum aposteriori' means!
6. What is maximum likelihood estimation? How can you get the maximum likelihood estimate for a Gaussian distribution?

What's next...?

Unit I	Machine Learning Introduction
Unit II	Mathematical Foundations
Unit III	Bayesian Decision Theory
Unit IV	**Regression**
Unit V	Classification I
Unit VI	Evaluation
Unit VII	Classification II
Unit VIII	Clustering
Unit IX	Dimensionality Reduction

Thank you very much for the attention!

Topic: *** Applied Machine Learning Fundamentals *** Bayesian Decision Theory
Term: Winter term 2023/2024

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

Do you have any questions?