

Machine Learning 1 – Fundamentals

Bayesian Learning

Prof. Dr. J. M. Zöllner, M.Sc. Marcus Fechner, M.Sc. Nikolai Polley



Outline

- Motivation
- Bayes' Theorem
- MAP- / ML-Hypothesis
- Bayes Optimal Classifier
- Naive Bayes Classifier
- Example: Text Classification
- Bayesian Networks
- Expectation-Maximization Algorithm
- Summary

What is Bayesian Learning?

■ Statistical Learning Method:

- Combines existing knowledge (a priori probabilities) with observed data
- Each hypothesis has an associated probability
- Each new observation/example can increase or decrease the probability of an existing hypothesis
 - No existing hypothesis is excluded or completely removed
- Several possible hypotheses can be evaluated together to obtain a more accurate result

What is Bayesian Learning?

- **Note:** „Hypothesis“ has different semantics, e.g. ...
 - **Statistics:** An assumption (about relationships/data) that holds with a certain probability
 - **Machine Learning:** A candidate model, which approximates a target function (Maps input \rightarrow output)

Why Bayesian Learning?

- **Successful learning methods:**

- Voting Gibbs (Bayes optimal classifier)
- Naive Bayes classifier
- Bayesian networks

- **Analysis of other learning methods:**

- „Gold standard“ for the assessment of (non-statistical) learning methods

- **Used in advanced learning methods such as semi-supervised learning and hyperparameter tuning of neural networks (→ ML2)**

Bayesian Learning: Challenges

■ Practical Problems:

- Prior knowledge of many probabilities/distributions necessary
- **But:** We can often estimate this prior knowledge based on background knowledge, existing data, etc.
- Significant computational effort (e.g. for Bayes optimal hypothesis in the general case)
 - Linear with number of possible hypotheses
 - **But:** In special cases, significant reduction of computational effort is possible

Outline

- Motivation
- Bayes' Theorem
- MAP- / ML-Hypothesis
- Bayes Optimal Classifier
- Naive Bayes Classifier
- Example: Text Classification
- Bayesian Networks
- Expectation-Maximization Algorithm
- Summary

Probability Theory

- **Product rule:** Conjunction of two events A and B

$$P(A \wedge B) = P(A|B)P(B) = P(B|A)P(A)$$

- **Sum rule:** Disjunction of two events A and B

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$


- **Law of total probability:** For mutually exclusive events A_1, \dots, A_n with $\sum_{i=1}^n P(A_i) = 1$

$$P(B) = \sum_{i=1}^n P(B|A_i)P(A_i)$$

- **Bayes' theorem:**

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

Bayes' Theorem (for ML)


$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- $P(h)$: A priori probability, that h of H is valid, before observing data D .
- $P(D)$: Probability of observing data D , without knowledge about valid hypotheses.
- $P(D|h)$: Likelihood, probability of observing data D under the model/hypothesis h .
- $P(h|D)$: A posteriori probability, that h is valid given the observed data D and the prior knowledge about h

Conditional Independence

- **Definition:** X is conditionally independent of Y given Z , if the probability distribution of X is independent of the value of Y , given the value of Z :

$$P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k), \quad \forall x_i, y_j, z_k$$

- Or:

$$P(X | Y, Z) = P(X | Z)$$

- **Example:** Thunder is conditionally independent of *Rain* given *Lightning*.

$$P(\text{Thunder} | \text{Rain}, \text{Lightning}) = P(\text{Thunder} | \text{Lightning})$$

Outline

- Motivation
- Bayes' Theorem
- MAP- / ML-Hypothesis
- Bayes Optimal Classifier
- Naive Bayes Classifier
- Example: Text Classification
- Bayesian Networks
- Expectation-Maximization Algorithm
- Summary

Selection of Hypotheses

- **Goal:** Find the hypothesis h of H with the highest probability given the observed data. This is the **maximum a posteriori (MAP) hypothesis**.

$$\begin{aligned} h_{MAP} &= \arg \max_{h \in H} P(h|D) \\ &= \arg \max_{h \in H} \frac{P(D|h)P(h)}{P(D)} && \text{Bayes} \\ &= \arg \max_{h \in H} P(D|h)P(h) && P(D) = \text{const.} \end{aligned}$$

- Under the assumption $P(h_i) = P(h_j)$, we can simplify this to the **maximum likelihood (ML) hypothesis**:

$$h_{ML} = \arg \max_{h \in H} P(D|h)$$

Brute Force Learning of MAP Hypotheses

Algorithm

1. For each hypothesis $h \in H$, calculate the posterior probability:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

2. Calculate the hypothesis h_{MAP} with the highest posterior probability:

$$h_{MAP} = \arg \max_{h \in H} P(D|h)P(h)$$

Example: Medical Diagnosis

■ Given:

- 0.8% of the population suffer from cancer.
- Given cancer, the probability of a positive test is 98%.
- Given no cancer, the test creates false positives with a probability of 3%.

■ Wanted:

- Probability that a person with positive test has cancer?

■ Definition:

- Hypothesis: h = cancer (and h' = no cancer)
- Observation: Positive (\oplus) or negative test (\ominus)

Example: Medical Diagnosis

- Prior knowledge of specific cancer diseases / tests:

$$P(\text{Cancer}) = 0.008$$

$$P(\neg \text{Cancer}) = 0.992$$

$$P(\oplus | \text{Cancer}) = 0.98$$

$$P(\ominus | \text{Cancer}) = 0.02$$

$$P(\oplus | \neg \text{Cancer}) = 0.03$$

$$P(\ominus | \neg \text{Cancer}) = 0.97$$

- **Given:**

$$P(\text{Cancer} | \oplus)$$

- **Solution:**

$$P(\text{Cancer} | \oplus) = \frac{P(\oplus | \text{Cancer})P(\text{Cancer})}{P(\oplus)}$$

$$\begin{aligned}
 &= \frac{P(\oplus | \text{Cancer})P(\text{Cancer})}{P(\oplus | \text{Cancer})P(\text{Cancer}) + P(\oplus | \neg \text{Cancer})P(\neg \text{Cancer})} \\
 &= 0.98 * 0.008 / (0.98 * 0.008 + 0.03 * 0.992) \\
 &= 0.21
 \end{aligned}$$

Example: Medical Diagnosis

$$h_{MAP} = \arg \max_{h \in H} P(D|h)P(h)$$

- Prior knowledge of specific cancer diseases / tests:

$$P(\text{Cancer}) = 0.008$$

$$P(\neg \text{Cancer}) = 0.992$$

$$P(\oplus | \text{Cancer}) = 0.98$$

$$P(\ominus | \text{Cancer}) = 0.02$$

$$P(\oplus | \neg \text{Cancer}) = 0.03$$

$$P(\ominus | \neg \text{Cancer}) = 0.97$$

This example illustrates how, even with seemingly highly accurate tests, results can often be misleading. In real-world scenarios, conditions are quite different, leading to more reliable outcomes.

- **Observation:** New patient, test \oplus

$$\begin{aligned} h_{MAP} &= \arg \max_{h \in \{\text{Cancer}, \neg \text{Cancer}\}} \{ (\oplus | \text{Cancer})P(\text{Cancer}), (\oplus | \neg \text{Cancer})P(\neg \text{Cancer}) \} \\ &= \arg \max_{h \in \{\text{Cancer}, \neg \text{Cancer}\}} \{ 0.98 * 0.008 = 0.0078, 0.03 * 0.992 = 0.0298 \} \\ &= \neg \text{Cancer} \end{aligned}$$

← h_{MAP}

Analysis Concept Learning 1

■ Problem formulation:

- Given: Observations X with target values $\{(x_1, d_1), \dots, (x_m, d_m)\}$
- Wanted: Hypothesis $h: X \rightarrow \{0,1\}$ from the finite hypothesis space H (of defined Boolean functions) for the so-called target concept $c: X \rightarrow \{0,1\}$

■ Assumption:

- Training data D has no noise/error (that is $d_i = c(x_i)$)
- Target concept c is included in H
- The prior probability of all hypotheses is equal \rightarrow Uniform distribution

- **Question:** Does a method that creates the version space (such as the search from the specific to the general, see Mitchell) provide a MAP hypothesis?

Analysis Concept Learning 2

■ **Prior knowledge:** $P(D|h) = \begin{cases} 1, & \text{if } h(x_i) = d_i, \forall d_i \in D \\ 0, & \text{else} \end{cases}$ and $P(h) = \frac{1}{|H|}$

■ Calculation of the a posteriori probability:

■ If (consistent hypotheses):

$$P(h|D) = \frac{1 \cdot \frac{1}{|H|}}{P(D)} = \frac{1 \cdot \frac{1}{|H|}}{\sum_{h \text{ consistent}} P(D|h)P(h)} = \frac{1 \cdot \frac{1}{|H|}}{\frac{|VS_{H,D}|}{|H|}} = \frac{1}{|VS_{H,D}|}$$

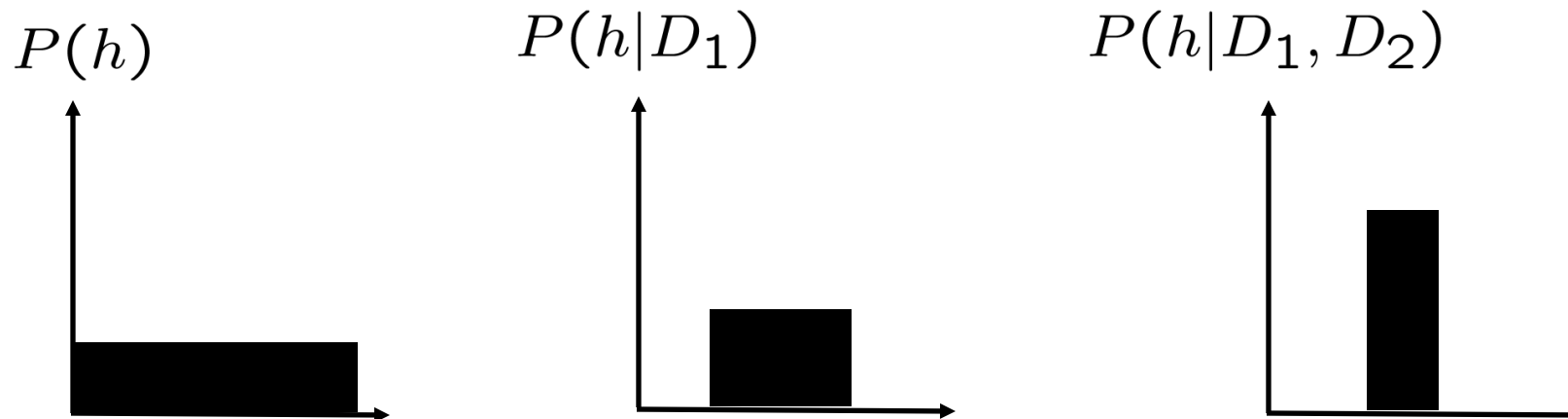
$VS_{H,D} = \{h \in H | h \text{ consistent with } D\}$ (Version-space of H)

■ Else: $P(h|D) = \frac{0 \cdot P(h)}{P(D)} = 0$

■ **Result:** $P(h|D) = \begin{cases} \frac{1}{|VS_{H,D}|}, & \text{if } h \in VS_{H,D} \text{ (consistent with } D) \\ 0, & \text{else} \end{cases}$

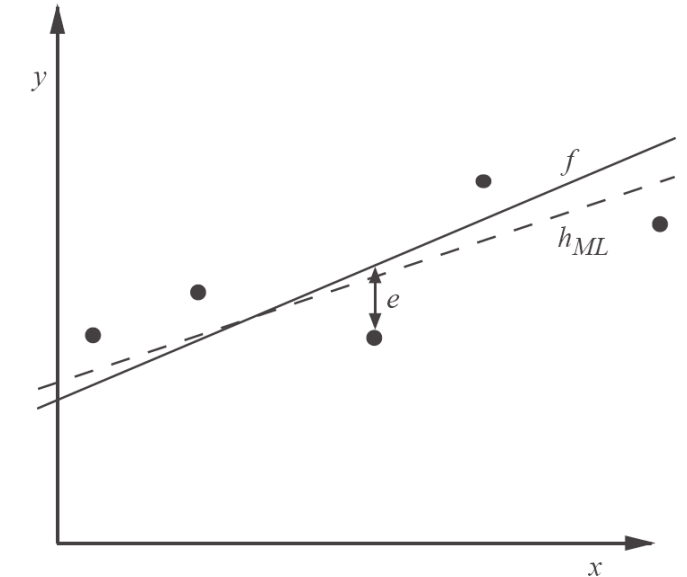
Conclusion: Consistent Learner

- **Definition:** A learning method is a consistent learner if it provides a hypothesis that **does not make mistakes on the training data**.
- Under the above assumptions, each consistent learner outputs a MAP hypothesis
- Method to express inductive bias (Probability of the hypothesis)
- Development of the a posteriori probabilities with increasing number of training data: for inconsistent hypotheses $P(h) = 0$



Learning a Real-Valued Function 1

- **Wanted:** Real-valued target function f
- **Given:** Examples (x_i, d_i) with noise e_i
$$d_i = f(x_i) + e_i$$
- e_i is a random variable (noise), that is distributed according to a Gaussian with mean $\mu = 0$. It is sampled independent of x_i .



- The maximum likelihood hypothesis $h_{ML} \in H$ (real-valued functions) is the hypothesis that minimizes the mean squared error:

$$h_{ML} = \arg \min_{h \in H} \sum_{i=1}^m (d_i - h(x_i))^2$$

Learning a Real-Valued Function 2

$$\begin{aligned}h_{ML} &= \arg \max_{h \in H} P(D|h) \\&= \arg \max_{h \in H} \prod_{i=1}^m P(d_i|h) \\&= \arg \max_{h \in H} \prod_{i=1}^m \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{d_i-h(x_i)}{\sigma}\right)^2} \\&= \arg \max_{h \in H} \sum_{i=1}^m \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2} \left(\frac{d_i-h(x_i)}{\sigma}\right)^2 \\&= \arg \min_{h \in H} \sum_{i=1}^m (d_i - h(x_i))^2\end{aligned}$$

d_i independent
product rule

noise
normal distribution

monotonic

Outline

- Motivation
- Bayes' Theorem
- MAP- / ML-Hypothesis
- Bayes Optimal Classifier
- Naive Bayes Classifier
- Example: Text Classification
- Bayesian Networks
- Expectation-Maximization Algorithm
- Summary

Bayes Optimal Classifier: Definition

■ Given:

- Hypotheses $h_i: X \rightarrow V$ that describe a classification into V classes
- A posteriori probability $p(h_i|D)$ of hypotheses h_i given data D .

■ **Wanted:** What is the most probable class $v_j \in V$ of an example $x \in X$, that is $p(v_j|x)$?

■ **Optimal classification according to Bayes:**

$$v_{OB} = \arg \max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

■ **Caution:** Output of $h_{MAP}(x)$ is not the most probable classification!

Example


■ **Given:** $\{h_1, h_2, h_3\}, V = \{\ominus, \oplus\}$ with


$$P(h_1|D) = 0.4, \quad P(h_2|D) = 0.3, \quad P(h_3|D) = 0.3$$

$$h_1(\mathbf{x}) = \oplus, \quad h_2(\mathbf{x}) = \ominus, \quad h_3(\mathbf{x}) = \ominus$$

■ **Wanted:** $v_{OB}|\mathbf{x}$

$$v_{OB} = \arg \max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

$P(h_1 D) = 0.4$	$P(\ominus h_1) = 0$	$P(\oplus h_1) = 1$		$\sum_{h_i \in H} P(\oplus h_i)P(h_i D) = 0.4$
$P(h_2 D) = 0.3$	$P(\ominus h_2) = 1$	$P(\oplus h_2) = 0$		$\sum_{h_i \in H} P(\ominus h_i)P(h_i D) = 0.6$
$P(h_3 D) = 0.3$	$P(\ominus h_3) = 1$	$P(\oplus h_3) = 0$		



■ **Solution:** $v_{OB} = \ominus$ (however $h_{MAP} = h_1 = \oplus$)

Bayes Optimal Classifier: Analysis

- **Advantage:** No other classification method (with the same hypothesis space and prior knowledge) performs better on average!
- **Disadvantage:** Intractable with large number of hypotheses!
- **Gibbs Algorithm:**

Algorithm

1. Choose h of H according to $P(h|D)$
2. Use $h(x)$ as classification (resp. class probability) of x

- **Extension:** Voting Gibbs (multiple samples and majority decision)
- **Property:** The following applies (see lit. [Haussler](#)):

$$E[\text{error}_{\text{Gibbs}}] \leq 2E[\text{error}_{\text{BayesOptimal}}]$$

Outline

- Motivation
- Bayes' Theorem
- MAP- / ML-Hypothesis
- Bayes Optimal Classifier
- Naive Bayes Classifier
- Example: Text Classification
- Bayesian Networks
- Expectation-Maximization Algorithm
- Summary

Naive Bayes Classifier 1

- **Often:** Bayes optimal classifier not applicable (e.g. no prior knowledge or computationally expensive)
- **Given:**
 - Examples $x \in X$ as conjunction of attributes $x_1 \in A_1, x_2 \in A_2 \dots x_n \in A_n$
 - Finite set of classes $V = \{v_1, \dots, v_m\}$
 - Set of classified examples
- **Wanted:** Most likely class $v_{MAP} \in V$ for a new example $(a_1, a_2 \dots a_n)$

- **MAP approach:**
$$\begin{aligned} v_{MAP} &= \arg \max_{v_j \in V} P(v_j | a_1, a_2 \dots a_n) = \arg \max_{v_j \in V} \frac{P(a_1, a_2 \dots a_n | v_j) P(v_j)}{P(a_1, a_2 \dots a_n)} \\ &= \arg \max_{v_j \in V} P(a_1, a_2 \dots a_n | v_j) P(v_j) \end{aligned}$$

Example: Play Tennis 1

■ Given: Learning examples; Wanted: Classification for: (sunny, cold, high, ...)

Outlook	Temperature	Humidity	Windy	Tennis?
sunny	hot	high	no	no
sunny	hot	high	yes	no
overcast	hot	high	no	yes
rain	mild	high	no	yes
rain	cold	normal	no	yes
rain	cold	normal	yes	no
overcast	cold	normal	yes	yes
sunny	mild	high	no	no
sunny	cold	normal	no	yes
rain	mild	normal	no	yes
sunny	mild	normal	yes	yes
overcast	mild	high	yes	yes
overcast	hot	normal	no	yes
rain	warm	high	yes	no

Naive Bayes Classifier 2

- $P(v_j)$ can be easily calculated from the occurrence of the class v_j in the training data – simply counting.
- $P(a_1, a_2 \dots a_n | v_j)$ is harder to calculate: Counting all combinations via attribute values requires a huge amount of training.

- **Simplifying assumption** (a_i conditionally independent):

$$P(a_1, a_2 \dots a_n | v_j) = \prod_i P(a_i | v_j)$$

- **Naive Bayes classifier:**

$$v_{NB} = \arg \max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$$

Naive Bayes Classifier 3

■ Procedure:

- $P(v_j)$ and $P(a_i|v_j) \forall i, j$ are estimated based on the occurrence in the training data.
- The probabilities are the parameters of the learned hypothesis.
- New examples are classified under the application of the MAP rule.
- If the assumption, conditional independence of attributes, is satisfied, v_{NB} is equivalent to a MAP classification.

→ No explicit search in the hypothesis space!

Example: Play Tennis 2

- Classification of new examples: (sunny, cold, high, yes)

$$v_{NB} = \arg \max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$$

- We need $P(v_j)$ and $P(a_i | v_j)$

$$P(\text{Tennis} = \text{yes}) = \frac{9}{14} = 0.64$$

$$P(\text{windy} = \text{yes} | \text{Tennis} = \text{yes}) = \frac{3}{9} = 0.33$$

$$P(\text{Tennis} = \text{no}) = \frac{5}{14} = 0.36$$

$$P(\text{windy} = \text{yes} | \text{Tennis} = \text{no}) = \frac{3}{5} = 0.60$$

$$P(\text{yes})P(\text{sunny} | \text{yes})P(\text{cold} | \text{yes})P(\text{high} | \text{yes})P(\text{yes} | \text{yes}) = 0.0053$$

$$P(\text{no})P(\text{sunny} | \text{no})P(\text{cold} | \text{no})P(\text{hoch} | \text{no})P(\text{yes} | \text{no}) = 0.0206$$

→ **Classification:** Tennis = no

- Normalized probability: $\frac{0.0206}{0.0206 + 0.0053} = 0.795$



Estimating Attribute Probabilities

- **Problem:** What happens if we don't have data for a specific class v_j and attribute a_i ?

$$P(a_i|v_j) = 0 \Rightarrow P(v_j) \prod_i P(a_i|v_j) = 0$$

- **Solution:** $\hat{P}(a_i|v_j) \leftarrow \frac{n_c + mp}{n + m}$ (m- Laplace estimator)

- **Where:**

- n - Number of examples with $v = v_j$
- n_c - Number of examples with $v = v_j$ and $x_j = a_i$
- p - a priori probability for $\hat{P}(a_i|v_j)$ e.g. uniform distribution: $p = \frac{1}{|A_i|}$
- m - Number of "virtual examples" weighted by a priori probability p

Outline

- Motivation
- Bayes' Theorem
- MAP- / ML-Hypothesis
- Bayes Optimal Classifier
- Naive Bayes Classifier
- Example: Text Classification
- Bayesian Networks
- Expectation-Maximization Algorithm
- Summary

Example: Text Classification 1

- **Motivation:** Statistical methods are (quite) successful in classifying texts.
- **Applications:**
 - Learning which news are interesting
 - Learning the topic affiliation of web pages, etc.
 - Detection of spam emails
- **Wanted:** Target function Interesting? : Document $\rightarrow \{\oplus, \ominus\}$
- **Approach:** Naive Bayes classifier
- **Question:** Which attributes are suitable to represent text documents?

Example: Text Classification 2

■ Approach:

- Representation of each text as a vector of words: One attribute x_i per word position in the document $\text{doc} = w_1 \dots w_{\text{length}(\text{doc})}$.
- Training phase: Use the training examples to estimate $P(\oplus), P(\ominus), P(\text{doc} | \oplus), P(\text{doc} | \ominus)$
- The following applies:

$$P(\text{doc} | v_j) = \prod_{i=1}^{\text{length}(\text{doc})} P(x_i = w_k | v_j)$$

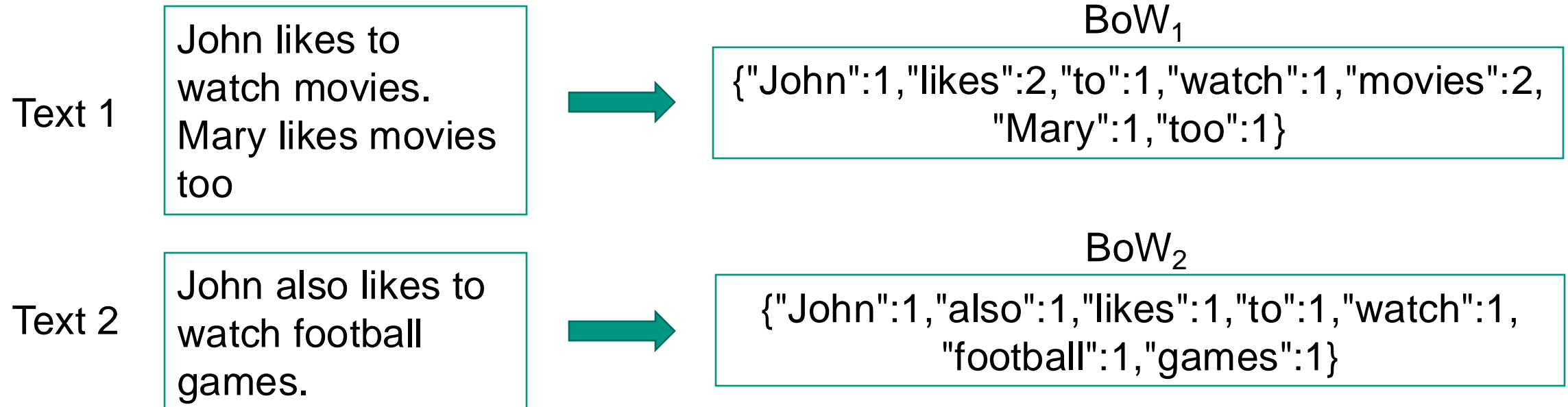
- $P(x_i = w_k | v_j)$: The probability, that word w_k appears at position i , given v_j .

Example: Text Classification 3

■ Approach:

- Additional "softer" assumption (bag of words, BoW): The probability of the occurrence of a particular word is independent of its position in the text.

$$P(x_i = w_k | v_j) = P(x_m = w_k | v_j), \quad \forall i, m$$



Example: Text Classification – Training Phase

■ Define vocabulary:

- Vocabulary: All words and tokens from the training examples

■ Calculate $P(v_j)$ and $P(w_k | v_j)$ for all v_j :

$$P(v_j) = \frac{|docs_j|}{|Examples|} \quad P(w_k | v_j) = \frac{n_k + 1}{n + |Vocabulary|} \quad (\text{Laplace estimator})$$

类别 v_j 的先验概率，即文档属于某类别的概率。

给定类别 v_j 的条件下，某个词 w_k 出现的概率。

■ with:

- $docs_j$: Subset of examples with v_j
- *Examples*: All documents
- n : Total number of word positions in $Text_j = \cup docs_j$
- n_k : Number of occurrences of w_k in $Text_j = \cup docs_j$

Example: Text Classification – Classification

- **Calculate:** For a new $doc = w_1 w_2 \dots w_n$ classify according to

$$v_{NB} = \arg \max_{v_j \in V} P(v_j) \prod_{i \in \text{Positions}} P(w_i | v_j)$$

- **with:**

- Positions: All positions in doc , that contain a token, which appears in *Vocabulary*.
- w_i : The word in position i . That means that tokens/words, which not appear in the training data, are ignored

Text Classification: Application (s. Mitchell)

■ Application:

- Given: 20 newsgroups with approx. 1000 posts each
- Wanted: Assignment of new posts to the newsgroups

■ Classifier:

- Naive Bayes classifier as above, but
 - 100 most common words have been removed from *Vocabulary*
 - Words with Occurrence $_{w_k} < 3$ are removed from the *Vocabulary*

■ Result:

- Classification rate of 89% (compared to 5% for random guessing)

■ Nowadays: Naive Bayes classifiers for text-processing have mostly been replaced by LLMs and other neural network architectures. (→ see later lecture and ML2)

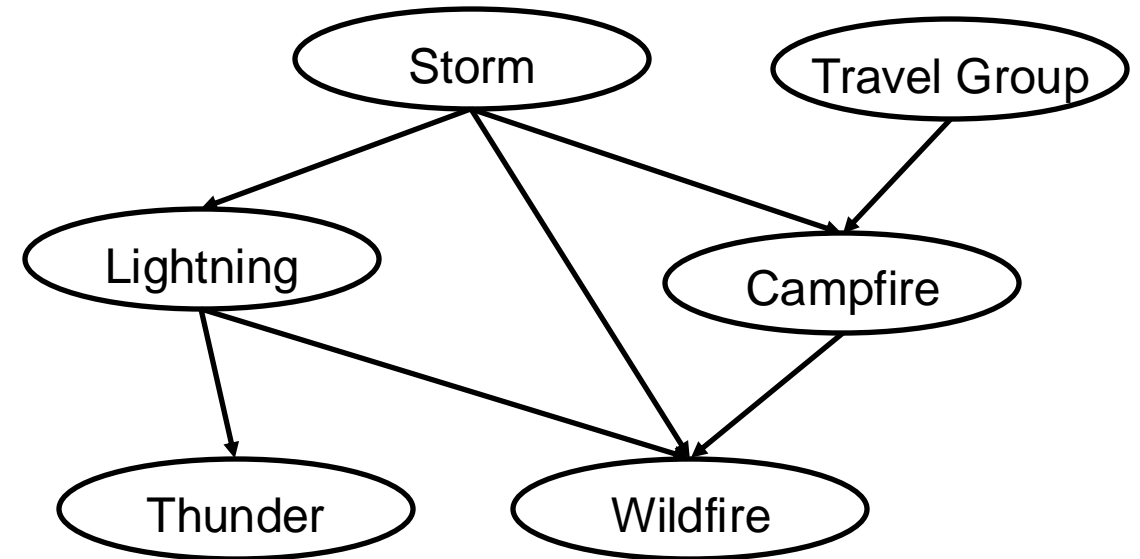
Outline

- Motivation
- Bayes' Theorem
- MAP- / ML-Hypothesis
- Bayes Optimal Classifier
- Naive Bayes Classifier
- Example: Text Classification
- Bayesian Networks
- Expectation-Maximization Algorithm
- Summary

Bayesian Networks 1

■ Motivation:

- Naive Bayes assumption of conditional independence is often too restrictive.
- But without such simplistic assumptions, Bayesian learning is often not computationally tractable.



■ Bayesian networks:

- Describes conditional dependencies (and thus also independence) with respect to subsets of (random) variables.
- Allows the combination of a priori knowledge about conditional (in)dependencies of variables with the observed training data.

Bayesian Networks 2

- Represents a joint probability distribution (of multiple random variables):
 - Directed, acyclic graph
 - Definition: X is the successor of Y if there is a direct path from Y to X
 - Each random variable is represented by a node in the graph, dependencies by edges
 - The edges represent the assurance that a variable is conditionally independent of its non-successors, given its direct predecessors.
 - For discrete random variables: Local tables with conditional probabilities for each variable given their direct predecessors. (A general parameterized distribution is also possible)

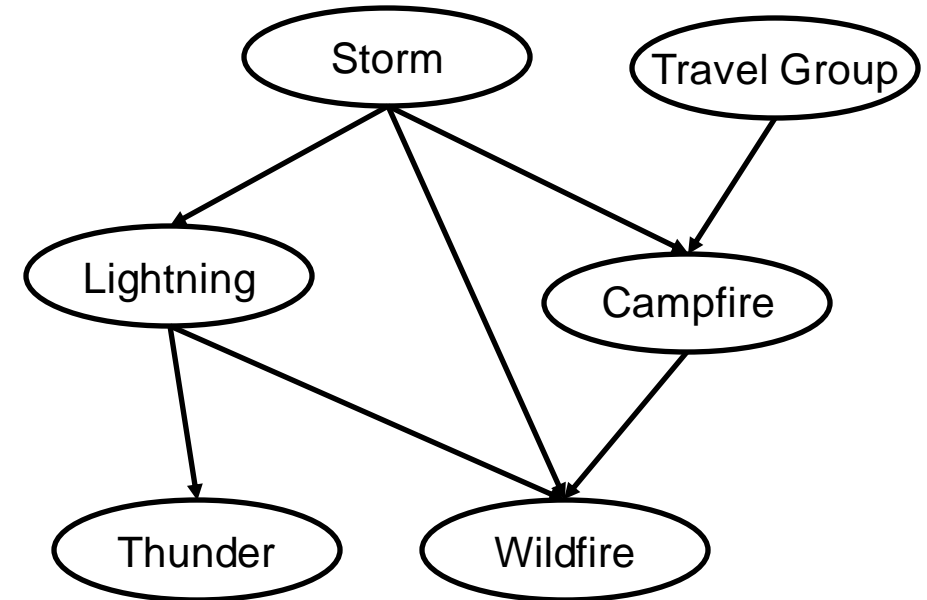
Bayesian Networks 3

- The following applies:

$$P(y_1, \dots, y_n) = \prod_{i=1}^n P(y_i | \text{Predecessor}(Y_i))$$

- Where:

- $\text{Predecessor}(Y_i)$: The set of direct predecessors of Y_i .



		$P(\text{Campfire} \text{Storm}, \text{Travel Group})$			
		S, G	$S, \neg G$	$\neg S, G$	$\neg S, \neg G$
Campfire	C	0.4	0.1	0.8	0.2
	$\neg C$	0.6	0.9	0.2	0.8

Bayesian Networks: Inference

- How to determine the values of one or more network variables, given the observed values of others?
 - Network contains all the information needed
 - Deriving a single random variable is easy
 - But: The general case is NP-complete
- In practice:
 - Some network topologies allow exact inference
 - Use of Monte Carlo methods for random simulation of networks: Calculation of approximate solutions

Bayesian Networks: Training

- Tasks:
 - Structure of the network known or unknown
 - All random variables are directly or only partially observable
- Structure known, all variables observable:
 - Learning (of conditional dependencies) similar to naive Bayes classifier
- Structure known, only some variables observable:
 - Gradient ascent, expectation-maximization algorithm
- Structure unknown:
 - Heuristic methods

Outline

- Motivation
- Bayes' Theorem
- MAP- / ML-Hypothesis
- Bayes Optimal Classifier
- Naive Bayes Classifier
- Example: Text Classification
- Bayesian Networks
- Expectation-Maximization Algorithm
- Summary

Expectation-Maximization (EM) Algorithm

■ General problem definition:

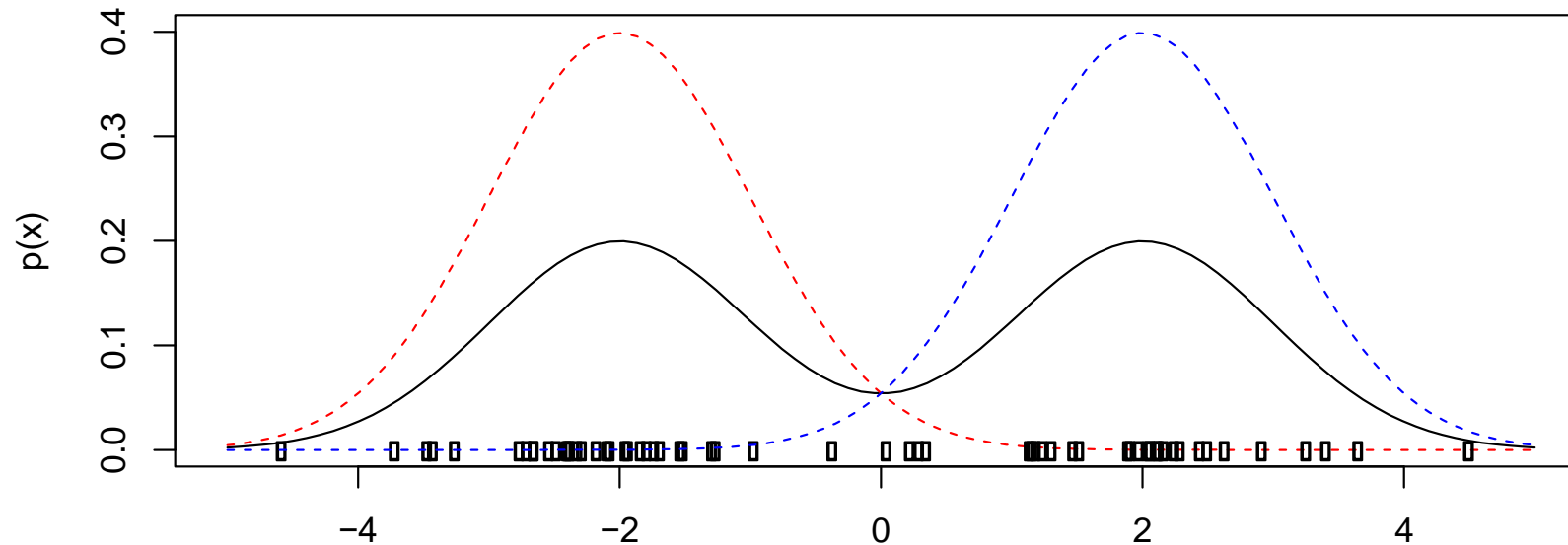
- Data is only partially observable (distributions can be modeled parametrically)
 - Unsupervised clustering (target value is unobservable)
 - Supervised learning (some attributes of the examples are not observable)
- Parameters of a (partial) hypothesis are to be estimated

■ Applications:

- Training of Bayesian networks
- Learning of hidden Markov models

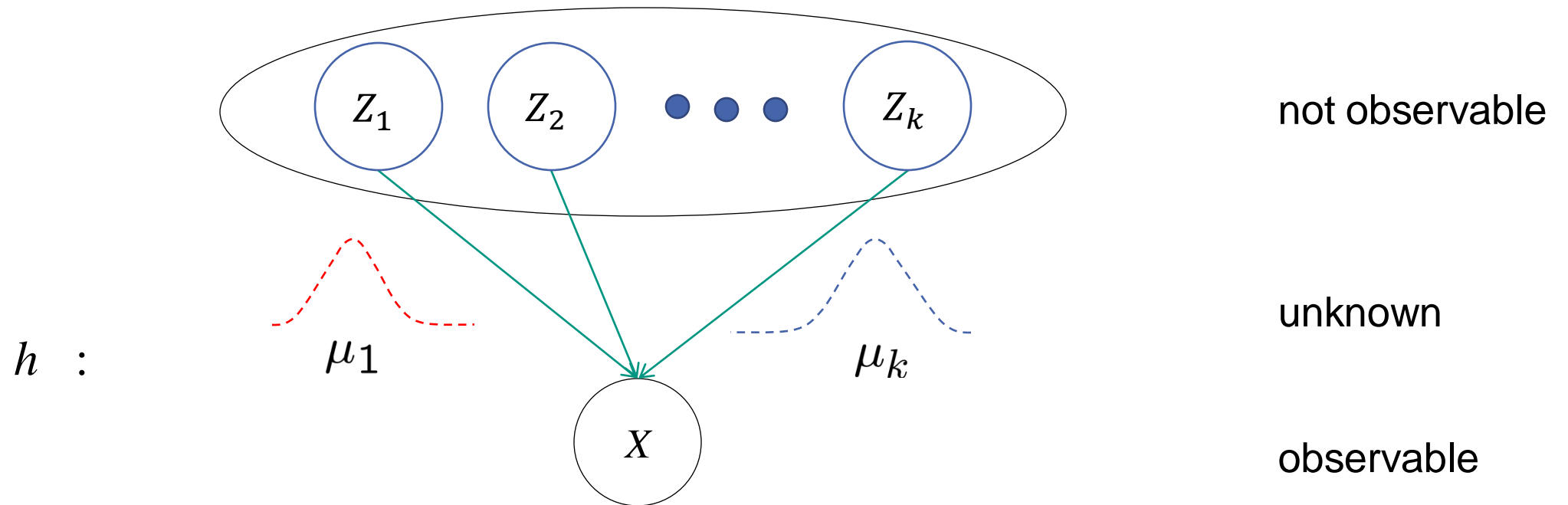
- **Basic idea:** Iterative approach – Estimate unobservable values (E-step) and adjust parameters (M-step)

EM – Example: Mixture of k Gaussian distributions



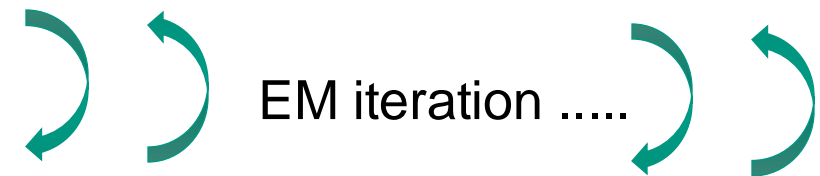
- Generation of each example x :
 - Choose one of the k Gaussian distributions (with equal probability)
 - Randomly generate an example according to the selected Gaussian distribution
 - No weighting here
- **Learning:** Observing examples and inferring Gaussian distributions

EM – Example: „Network Model – Illustration“



E-step: If x and h are known $\rightarrow z_j$ resp. $P(z_j)$

M-step: If z and x are known $\rightarrow h_{\mu_1, \dots, \mu_k}$



EM – Example: Estimation of k Mean Values 1

■ Given:

- Examples $x_i \in X$ are generated according to k Gaussian distributions
- Unknown means μ_1, \dots, μ_k
- It is unknown which instance x_i was generated according to which Gaussian distribution

■ Wanted: Maximum likelihood estimate for h_{μ_1, \dots, μ_k} that is parameters μ_1, \dots, μ_k

■ Approach:

- Observation of the examples through: $y_i = (x_i, z_{i1}, z_{i2}, \dots, z_{ik})$
 - Where:
 - z_{ij} is 1, if x_i is sampled according to the j -th Gaussian distribution, else 0
- x_i observable, z_{ij} not observable

EM – Example: Estimation of $k(=2)$ Mean Values 2

- **Assumption:** $k = 2$ (Number of Gaussian distributions)
- **Initialization:**
 - Choose h_{μ_1, μ_2} with μ_1, μ_2 randomly.
- **E-step:**
 - Calculate the expected value $E[Z_{ij}]$ for each hidden variable Z_{ij} under the assumption that the recent hypothesis h_{μ_1, μ_2} is valid.

$$\begin{aligned} E[Z_{ij}] &= \frac{P(x = x_i | \mu = \mu_j)}{\sum_{n=1}^2 P(x = x_i | \mu = \mu_n)} \\ &= \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2}}{\sum_{n=1}^2 e^{-\frac{1}{2\sigma^2}(x_i - \mu_n)^2}} \end{aligned}$$

■ M-step:

- Calculate a new maximum likelihood hypothesis $h'_{\mu'_1, \mu'_2}$
- Assuming that the values of the hidden variables Z_{ij} take on the expected values calculated in the E-step.

$$\mu'_j = \frac{1}{m} \sum_{i=1}^m E[Z_{ij}] x_i$$

- Replace h_{μ_1, μ_2} by the new hypothesis $h'_{\mu'_1, \mu'_2}$ and iterate.

General EM – Problem

■ Given:

- Observed data $X = \{x_1, \dots, x_m\}$
- Hidden data $Z = \{z_1, \dots, z_m\}$
- Parameterized probability distribution $P(Y|h)$,
- Where:
 - $Y = \{y_1, \dots, y_m\}$ is the complete data $Y = X \cup Z$
 - h the hypothesis (depending on the corresponding parameters)

■ Wanted:

- Hypothesis h , which maximizes $P(Y|h)$, particularly $E[\ln P(Y|h)]$ (local)

General EM – Method

■ E-step:

- Calculate $P(Y | X, h)$ by utilizing the most recent hypothesis h and the observed data X .
- $Y = X \cup Z$, resp. Z is estimated according to $P(Y | X, h)$ from observable data X and known h

■ M-step:

- Replace hypothesis h with hypothesis h' , which maximizes a Q function

$$h' = \arg \max_{h'} Q, \quad \text{with: } Q(h'|X, h) = E[\ln P(Y|h') | X, h]$$

- $Q(h'|X, h)$ is maximized, if h' correctly generates data $Y = X \cup Z$

- Converges to local maximum likelihood hypothesis $h' \rightarrow h_{ML}$ after some iterations

Outline

- Motivation
- Bayes' Theorem
- MAP- / ML-Hypothesis
- Bayes Optimal Classifier
- Naive Bayes Classifier
- Example: Text Classification
- Bayesian Networks
- Expectation-Maximization Algorithm
- Summary

Summary 1

- Bayesian methods determine a posteriori probabilities for hypotheses based on assumed a priori probabilities and observed data.
- Bayesian methods can be used to determine the most probable hypothesis (MAP hypothesis) ("optimal" hypothesis).
- The Bayes optimal classifier determines the most probable classification of a new instance from the weighted predictions of all hypotheses.
- The naive Bayes classifier is a successful learning method.
Assumption: Conditional independence of the attribute values.
- Bayesian methods allow the analysis of other learning algorithms that do not directly apply Bayes' theorem.

Summary 2

- Bayesian networks describe joint probability distributions with the help of a directed graph (and in the discrete case with local probability tables).
- Bayesian networks model conditional independence in subsets of random variables. Less restrictive than the naive Bayes classifier.
- Learning in Bayesian networks in different tasks
- The iterative EM algorithm allows the handling of hidden random variables.

Literature

- [1] *Tom Mitchell: **Machine Learning***. McGraw-Hill, New York, 1997. Chapter 6.
- [2] *S. Russel, P. Norvig: **Artificial Intelligence: A Modern Approach***. Prentice Hall, 2nd Edition, 2003.
- [3] *Christopher M. Bishop: **Pattern Recognition and Machine Learning***. Springer, 2006.