Universität
Augsburg
University

Bayesian Learning

Two Roles for Bayesian Methods (1)

1. Role: Practical learning algorithms

- Naive Bayes learning
- Bayesian network learning

- Requires prior probabilities
- Combine prior knowledge (=domain knowledge) by means of prior probabilities with observed data
- Models uncertainty

- Considers all hypotheses simultaneously (➔ $p(h|D)$)
- Probabilities can be updated as new data becomes available
  - Each observed training examples incrementally increases or decreases the estimated probability that a hypothesis is correct. This provides a more flexible approach than completely eliminating a hypothesis.

Two Roles for Bayesian Methods (2)

– New instances can be classified by combing the predictions of multiple hypotheses weighted by their probability p(h|D)

– Usually computationally expensive

2. Role: Useful conceptual framework

– Provides “gold standard” for evaluating other learning algorithms
– Additional insight into Occam's razor

Outline

- Recap of basic formulas for probabilities
- Bayes Theorem
- MAP / ML hypotheses
- MAP / ML learners
- Minimum description length principle (MDL)
- Bayes optimal classifier
- Naive Bayes learner
- Example: Learning over text data
- Expectation Maximization (EM) algorithm

Basic Formulas for Probabilities (1)

$P(X = x) = P(x)$:= probability of random variable X taking value x

- e.g. relative frequency
- Example: *X* is outcome of fair coin toss: *P(X="heads")= P(X="tails")=0.5*

$P(x \wedge y) = P(x, y)$:= probability of $X = x$ AND $Y = y$
(at the same time)
= joint probability of $X = x$ and $Y = y$

$P(x \vee y)$:= probability of $X = x$ OR $Y = y$ (either one or both)

Basic Formulas for Probabilities (2)

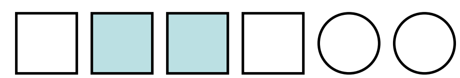
$p(x) \hat{=}$ pior prob.
$p(x|y) \hat{=}$ a posteriori prob.

$P(x|y)$:= probability of $X = x$ given $Y = y$
= conditional probability

– *For* $P(y) \neq 0$ $$P(x \mid y) = \frac{P(x \wedge y)}{P(y)} = \frac{P(x, y)}{P(y)}$$

– *For* $P(y) = 0$ $$P(x \mid y) = P(x)$$

Example: Randomly draw one of the following objects:



$$P(\text{"}blue\text{"}) = \frac{1}{3} \qquad P(\text{"}blue\text{"} \mid \text{"}square\text{"}) = \frac{P(\text{"}blue\text{"} \wedge \text{"}square\text{"})}{P(\text{"}square\text{"})} = \frac{1}{3} / \frac{2}{3} = \frac{1}{2}$$

Basic Formulas for Probabilities (3)

- *Product Rule:* Probability of a conjunction of two events *a* and *b*:

$$P(a \wedge b) = P(a, b) = P(a|b)P(b) = P(b|a)P(a)$$

- *Sum Rule:* Probability of a disjunction of two events a and b:

$$P(a \vee b) = P(a) + P(b) - P(a, b)$$

- *Theorem of total probability:* If events $a_1, \ldots, a_n$ are mutually exclusive with $\sum_{i=1}^{n} P(a_i) = 1$, then

$$P(b) = \sum_{i=1}^{n} P(b, a_i) = \sum_{i=1}^{n} P(b|a_i)\, P(a_i)$$

Bayes Theorem (1)

$$P(a \mid b) = \frac{P(b \mid a)P(a)}{P(b)}, \quad P(b) \neq 0, \quad P(a) \neq 0$$

posterior probability for a given b

prior probability for a

prior probability for b/ normalization constant $P(b) = \sum_i P(b \mid a_i)P(a_i)$

a,b are events or values of random variables

$$P(a,b) = P(a \mid b)P(b)$$

$$P(a,b) = P(b,a) = P(b \mid a)P(a)$$

$$\Rightarrow P(a \mid b) = \frac{P(b \mid a)P(a)}{P(b)}$$

Bayes Theorem (2)

For hypotheses given training data:

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

- *P(h)* = prior probability of hypothesis *h*

  = prior knowledge about the chance that h is correct
- *P(D)* = prior probability of training data *D*

  = prior knowledge that training data D will be observed
- *P(h|D)* = (posterior) probability of *h* given *D*
- *P(D|h)* = probability of *D* given *h*

Choosing Best Hypothesis

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

Generally, we want the most probable hypothesis given the training data

Maximum a posteriori (MAP) hypothesis h_{MAP}:

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(h \mid D) \\ &= \arg\max_{h \in H} \frac{P(D \mid h)P(h)}{P(D)} \\ &= \arg\max_{h \in H} P(D \mid h)P(h) \end{aligned}$$

If assume $P(h_i) = P(h_j)$ for all i,j then we can further simplify, and choose the *Maximum likelihood* (ML) hypothesis

$$h_{ML} = \arg\max_{h_i \in H} P(D \mid h_i)$$

Universität Augsburg University

Does patient have cancer or not?

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in only 98% of the cases in which the disease is actually present, and a correct negative result in only 97% of the cases in which the disease is not present. Furthermore, 0.8% of the entire population have this cancer.

Suppose, the patient receives a positive result. What is the probability that the patient does have cancer?

$$P(cancer) = \qquad P(\neg cancer) =$$

$$P(+ \mid cancer) = \qquad P(- \mid cancer) =$$

$$P(+ \mid \neg cancer) = \qquad P(- \mid \neg cancer) =$$

Bayes Theorem - Example

$$P(cancer) = 0.008 \qquad P(\neg cancer) = 0.992$$

$$P(+ \mid cancer) = 0.98 \qquad P(- \mid cancer) = 0.02$$

$$P(+ \mid \neg cancer) = 0.03 \qquad P(- \mid \neg cancer) = 0.97$$

Suppose, the patient receives a positive result. What is the probability that the patient does have cancer?

$$h_{MAP} = arg \max_{h \in \{cancer, \neg cancer\}} P(h|+)$$

$$= arg \max_{h \in \{cancer, \neg cancer\}} P(+|h)P(h)$$

$$= arg \max\{P(+|cancer) \cdot P(cancer), P(+|\neg cancer)P \cdot (\neg cancer)\}$$

$$= arg \max\{0.98 \cdot 0.008, 0.03 \cdot 0.992\}$$

$$= arg \max\{0.00784, 0.0376\}$$

$$h_{MAP} = \neg cancer$$

Brute Force MAP Hypothesis Learner

1. For each hypothesis *h* in *H*, calculate the posterior probability

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

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

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

Used as „gold“ standard!

Relation to Concept Learning

Consider our usual concept learning task

- instance space X, hypothesis space H, training examples D
- consider the FindS learning algorithm (outputs most specific hypothesis from the version space $VS_{H,D}$)

What would Bayes rule produce as the MAP hypothesis?

Does FindS output a MAP hypothesis?

Relation to Concept Learning

Assume fixed set of instances $\langle x_1, \ldots, x_m \rangle$

Assume $D = \langle c(x_1), \ldots c(x_m) \rangle$ is the set of classifications

Choose $P(D|h)$:

- We postulate that *h* holds, i.e. is a correct description of target concept
- Training data is noise-free
- Thus if *h* classifies each instance as observed, i.e. $h(x_i) = c(x)$ for all i), then the probability $P(D|h)$ of observing D is 1
- If not $h(x_i) = c(x)$ for all i, then $P(D|h)$ is 0

Relation to Concept Learning

Assume fixed set of instances $\langle x_1, \ldots, x_m \rangle$

Assume $D = \langle c(x_1), \ldots c(x_m) \rangle$ is the set of classifications

Choose $P(D|h)$:

$$P(D|h) = \begin{cases} 1 & if\ h \in VS_{H,D} \\ 0 & otherwise \end{cases}$$

Choose *P(h)* to be *uniform* distribution

- $P(h) = \frac{1}{|H|}$ for all h in H

Then,

$$P(h|D) = \begin{cases} \frac{1}{|VS_{H,D}|} & \text{if } h \text{ is consistent with } D \\ 0 & \text{otherwise} \end{cases}$$

Proof

Relation to Concept Learning

Thus, every consistent hypothesis has the same probability given D

$\Rightarrow$ every consistent hypothesis is a MAP hypothesis

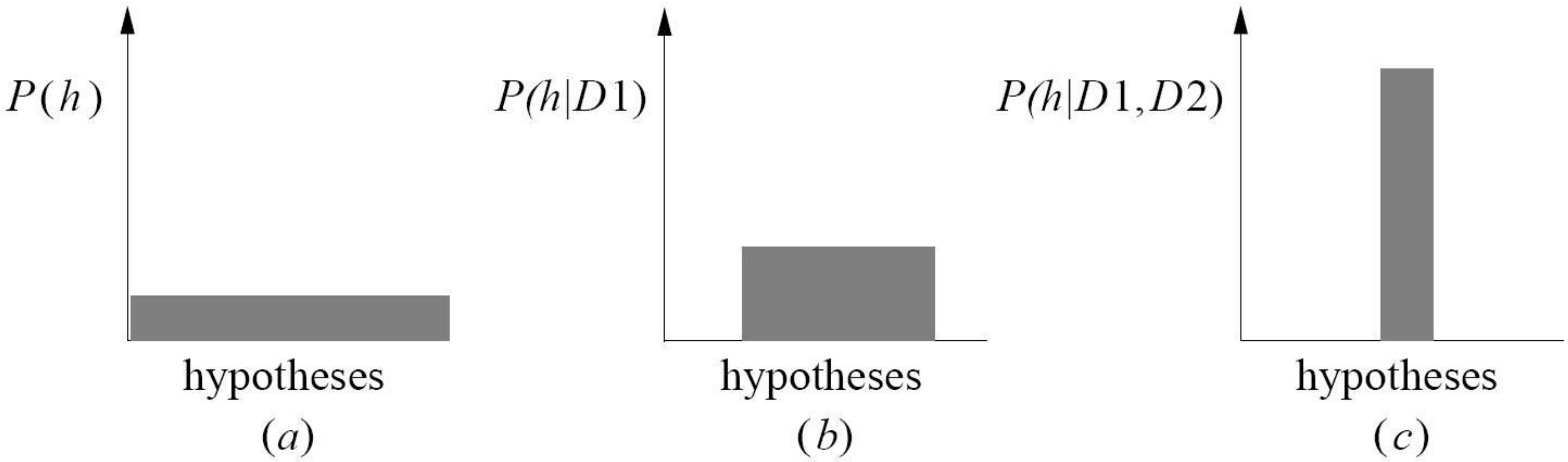
What would Bayes rule produce as the MAP hypothesis?

One of the consistent hypothesis

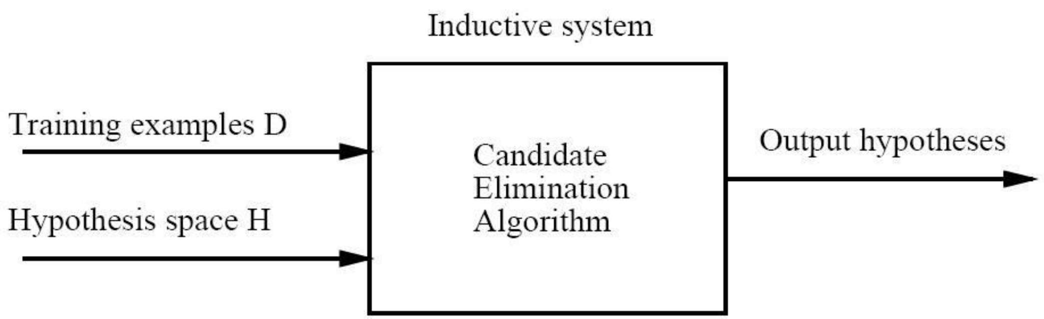
Does FindS output a MAP hypothesis?

Yes (it finds a consistent hypothesis).

Evolution of Posterior Probabilities

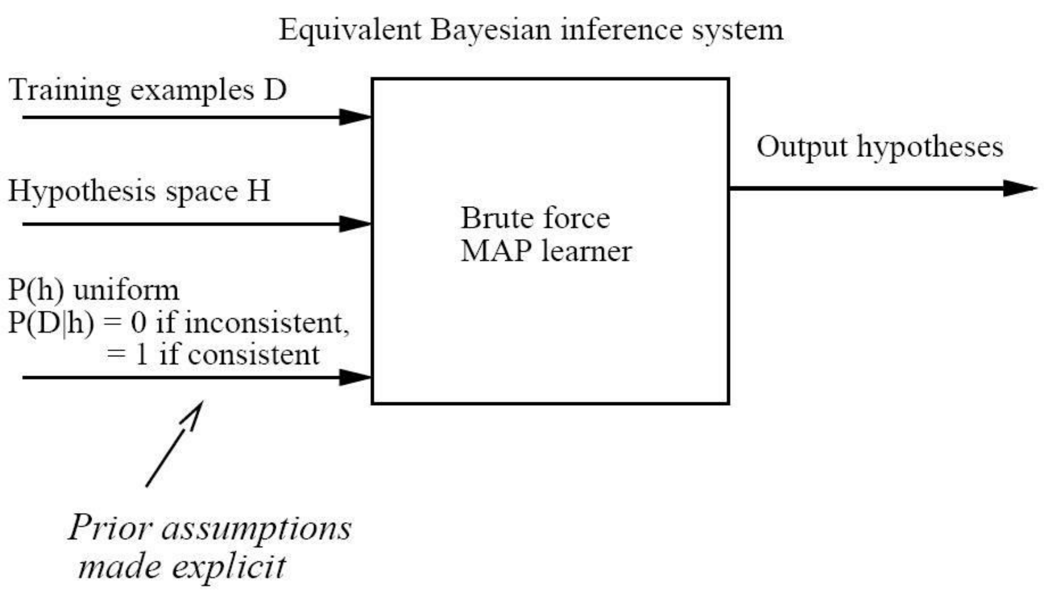


Characterizing Learning Algorithm by Equivalent MAP Learners



Instead of modeling inductive inference by an *equivalent deductive system*

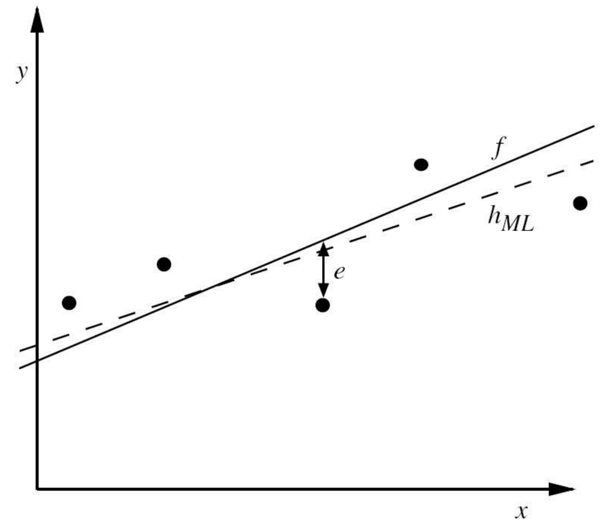
use



an *equivalent probabilistic reasoning system* based on Bayes theorem

Learning A Real Valued Function

Consider any real-valued target function f. Given are training Examples $\langle x_i, d_i \rangle$ where d_i is noisy training value



- $d_i = f(x_i) + e_i$
- e_i is random variable (noise) drawn independently for each x_i according to some Gaussian distribution with mean=0

Then the maximum likelihood hypothesis h_{ML} is the one that minimizes the sum of squared errors:

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

Learning A Real Valued Function

$$h_{ML} = \arg\max_{h\in H} p(D|h) = \arg\max_{h\in H} \prod_{i=1}^{m} p(d_i|h)$$

$$= \arg\max_{h\in H} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2}$$

Maximize natural log of this instead...

$$h_{ML} = \arg\max_{h\in H} \sum_{i=1}^{m} \left[\ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2\right]$$

$$= \arg\max_{h\in H} \sum_{i=1}^{m} -\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2$$

$$= \arg\max_{h\in H} \sum_{i=1}^{m} -\left(d_i - h(x_i)\right)^2$$

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

Minimum Description Length (MDL) Principle

$$h_{MAP} = \arg\max_{h\in H} P(D|h)P(h) = \arg\max_{h\in H} \log_2 P(D|h) + \log_2 P(h)$$
$$= \arg\min_{h\in H} -\log_2 P(D|h) - \log_2 P(h) \qquad (1)$$

Interesting fact from information theory:
If we need to transmit a message we encounter with probability p, the optimal (shortest) expected coding length L_C for this message is $-\log_2 p$ bits.

So interpret (1):

- $-\log_2 P(h)$ is length L_{C_H} of h under optimal code C_H
- $-\log_2 P(D|h)$ is length $L_{C_{D|h}}$ of D given h under optimal code $C_{D|h}$
  - Since we transmit h, we only must transmit the classifications of the instances which h misclassifies

$\Rightarrow$ prefer the hypothesis h that minimizes

$$length(h) + length(misclassifications)$$

Minimum Description Length Principle

Occam's razor: prefer the shortest hypothesis

MDL: prefer the hypothesis *h* that minimizes

$$h_{MDL} = \arg\min_{h \in H} L_{C_1}(h) + L_{C_2}(D \mid h)$$

where $L_C(x)$ is the description length of x under encoding C

If C_1 and C_2 are optimal encodings C_H and $C_{D|h}$, $h_{MDL} = h_{MAP}$

Example: H = decision trees, D = training data labels

- $L_{C1}(h)$ is # bits to describe tree h
- $L_{C2}(D|h)$ is # bits to describe D given h
  - Note $L_{C2}(D|h) = 0$ if examples classified perfectly by h. Need only describe exceptions
- Hence h_{MDL} trades off tree size for training errors

Most Probable Classification of New Instances

So far we've sought the most probable *hypothesis* given the data *D* (i.e., h_{MAP})

Given new instance *x*, what is its most probable *classification*?

- $h_{MAP}(x)$ is not the most probable classification!

Consider:

- Three possible hypotheses:

$$P(h_1 \mid D) = .4, \quad P(h_2 \mid D) = .3, \quad P(h_3 \mid D) = .3$$

- Given new instance *x*,

$$h_1(x) = +, \quad h_2(x) = -, \quad h_3(x) = -$$

- What's most probable classification of *x*?

  Brute Force MAP learner:

$$h_{MAP}(x) = h_1(x) = +$$

Bayes optimal classification:

$$v_{opt} = \arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j \mid h_i) P(h_i \mid D) \qquad V = \{+,-\}$$

Example:

$$P(h_1 \mid D) = .4, \quad P(- \mid h_1) = 0, \quad P(+ \mid h_1) = 1$$
$$P(h_2 \mid D) = .3, \quad P(- \mid h_2) = 1, \quad P(+ \mid h_2) = 0$$
$$P(h_3 \mid D) = .3, \quad P(- \mid h_3) = 1, \quad P(+ \mid h_3) = 0$$

Therefore

$$\sum_{h_i \in H} P(+ \mid h_i) P(h_i \mid D) = .4$$

$$\sum_{h_i \in H} P(- \mid h_i) P(h_i \mid D) = .6$$

and

$$v_{opt} = \arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j \mid h_i) P(h_i \mid D) = -$$

Gibbs Classifier

Bayes optimal classifier provides best result, but can be expensive if many hypotheses.

Gibbs algorithm:

1. Choose one hypothesis at random, according to *P(h|D)*
2. Use this to classify new instance

Surprising fact: Assume target concepts are drawn at random according to priors assumed by learner. Then:

$$E[error_{Gibbs}] \leq 2E[error_{BayesOptimal}]$$

Suppose correct, uniform prior distribution over *H*, then

- Pick any hypothesis from *VS*, with uniform probability
- Its expected error no worse than twice Bayes optimal

Naive Bayes Classifier (1)

Along with decision trees, neural networks & nearest neighbor, one of the most practical learning methods

When to use

- Moderate or large training set available
- Attributes that describe instances are conditionally independent given classification

Successful applications:

- Diagnosis
- Classifying text documents

Naive Bayes Classifier (2)

Assume target function $f: X \rightarrow V$ where each instance x described by attributes $\langle a_1, a_2, \ldots, a_n \rangle$. Most probable value of $f(x)$ is:

$$v_{MAP} = arg \max_{v_j \in V} P\left(v_j | a_1, a_2, \cdots, a_n\right)$$

$$= arg \max_{v_j \in V} \frac{P\left(a_1, a_2, \cdots, a_n | v_j\right) P(v_j)}{P(a_1, a_2, \cdots, a_n)}$$

$$= arg \max_{v_j \in V} P\left(a_1, a_2, \cdots, a_n | v_j\right) P(v_j)$$

Naive Bayes assumption:

$$P\left(a_1, a_2, \cdots, a_n | v_j\right) = \prod_i P(a_i | v_j)$$

(= Attribute values are conditionally independent given the target value)

which gives

Naive Bayes classifier: $v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i \mid v_j)$

Note: $v_{NB} == v_{MAP}$ iff conditional independence holds

Naive Bayes Algorithm

Universität Augsburg University

NAIVE_BAYES_LEARN (*examples*)

For each target value v_j

$$\hat{P}(v_j) \leftarrow \text{estimate } \boxed{P(v_j)}$$

For each attribute value a_i of each attribute A

$$\hat{P}(a_i \mid v_j) \leftarrow \text{estimate } \boxed{P(a_i \mid v_j)}$$

Determined by relative frequency over training data

CLASSIFY_NEW_INSTANCE(*x*)

$$v_{NB} = \arg\max_{v_j \in V} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i \mid v_j)$$

Naive Bayes: Example

Consider *PlayTennis* again, and new instance

$$\langle Outlook = sun, Temp = cool, Humid = high, Wind = strong \rangle$$

Want to compute:

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

$$P(y)P(sun \mid y)P(cool \mid y)P(high \mid y)P(strong \mid y) = .005$$

$$P(n)P(sun \mid n)P(cool \mid n)P(high \mid n)P(strong \mid n) = .021$$

$$\Rightarrow \quad v_{NB} = n$$

Vorrechnen

Naive Bayes: Subtleties (1)

1. Conditional independence assumption is often violated

$$P(a_1, a_2 \ldots a_n \mid v_j) = \prod_i P(a_i \mid v_j)$$

- …but it works surprisingly well in practice.

  Note: Don't need estimated posteriors $\hat{P}(v_j|x)$ to be correct; need only that

$$\arg\max_{v_j \in V} \hat{P}(v_j) \prod_i \hat{P}(a_i \mid v_j) = \arg\max_{v_j \in V} P(v_j) P(a_1 \ldots a_n \mid v_j)$$

- Naive Bayes posteriors often unrealistically close to 1 or 0

Naive Bayes: Subtleties (2)

2. what if none of the training instances with target value v_j have attribute value a_i? Then

$$\hat{P}(a_i \mid v_j) = 0, \text{ and...}$$

$$\hat{\mathrm{P}}(\mathrm{v_j}) \prod_i \hat{P}(a_i \mid v_j) = 0$$

Typical solution is Bayesian estimate for $\hat{P}(a_i|v_j)$

$$\hat{P}(a_i \mid v_j) \leftarrow \frac{n_c + mp}{n + m}$$

where

- n is number of training examples for which $v = v_j$,
- n_c number of examples for which $v = v_j$ and $a = a_i$
- p is prior estimate for $\hat{P}(a_i|v_j)$
- m is weight given to prior (i.e. number of “virtual” examples)

Universität
Augsburg
University

Applications:

- Learn which news articles are of interest
- Learn to classify web pages by topic
- Filter spam mails

Naive Bayes is among most effective algorithms

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

Where v is target concept (interesting/not interesting, …)

What attributes a_i shall we use to represent text documents?

Learning to Classify Text (2)

1. Target concept $Interesting: Document \rightarrow V$ with $V = \{+, -\}$
2. Represent each document by vector of words
   - one attribute per word position in document: a_i is word at i-th position
3. Learning: Use training examples to estimate
   - $P(+)$
   - $P(-)$
   - $P(doc|+)$
   - $P(doc|-)$

   Naive Bayes conditional independence assumption

$$P(doc \mid v_j) = \prod_{i=1}^{length(doc)} P(a_i = w_k \mid v_j)$$

   where $P(a_i = w_k|v_j)$ is probability that word in position *i* is w_k, given v_j

One more assumption: $P(a_i = w_k \mid v_j) = P(a_m = w_k \mid v_j), \forall i, m$

Learning to Classify Text (3)

LEARN_NAIVE_BAYES_TEXT (*Examples, V*)

1. *collect all words and other tokens that occur in Examples*
   - Vocabulary ← all distinct words and other tokens in *Examples*
2. *calculate the required* $P(v_j)$ *and* $P(w_k|vj)$ *probability terms*
   - *For each target value* v_j *in V do*
     - $docs_j$ ← subset of *Examples* for which the target value is v_j
     - $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
     - $Text_j$ ← a single document created by concatenating all members of $docs_j \leftarrow$
     - n ← total number of words in $Text_j$ (counting duplicate words multiple times)
     - for each word w_k in *Vocabulary*
       - n_k ← number of times word w_k occurs in $Text_j$
       - $P(w_k | v_j) \leftarrow \frac{n_k + 1}{n + |Vocabulary|}$

$$m = |Vocabulary|$$
$$p = \frac{1}{|Vocabulary|}$$

Universität Augsburg University

CLASSIFY_NAIVE_BAYES_TEXT(*Doc*)

- $positions$ ← all word positions in *Doc* that contain tokens found in V*ocabulary*
- Return v_{NB}, where

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_{i \in positions} P(a_i \mid v_j)$$

Twenty News Groups

Given 667 training documents from each group (20 groups in total)
Learn to classify new documents according to which newsgroup it came from (target concept: which news group?)

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	misc.forsale rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey
alt.atheism soc.religion.christian talk.religion.misc talk.politics.mideast talk.politics.misc talk.politics.guns	sci.space sci.crypt sci.electronics sci.med

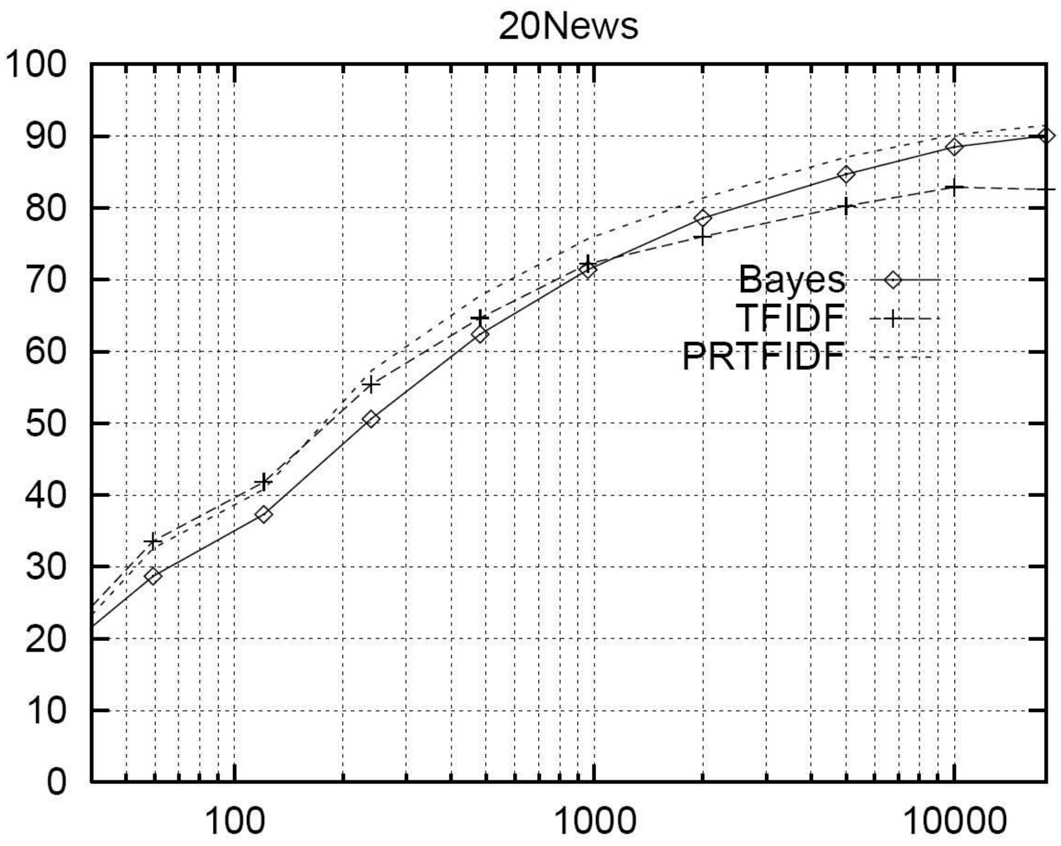
Naive Bayes: 89% classification accuracy on 333 test articles of each group

Article from rec.sport.hockey

```
Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.edu!ogicse!uwm.edu
From: xxx@yyy.zzz.edu (John Doe)
Subject: Re: This year's biggest and worst (opinion)...
Date: 5 Apr 93 09:53:39 GMT

I can only comment on the Kings, but the most
obvious candidate for pleasant surprise is Alex
Zhitnik. He came highly touted as a defensive
defenseman, but he's clearly much more than that.
Great skater and hard shot (though wish he were
more accurate). In fact, he pretty much allowed
the Kings to trade away that huge defensive
liability Paul Coffey. Kelly Hrudey is only the
biggest disappointment if you thought he was any
good to begin with. But, at best, he's only a
mediocre goaltender. A better choice would be
Tomas Sandstrom, though not through any fault of
his own, but because some thugs in Toronto decided
```

Learning Curve for 20 Newsgroups



TFIDF=
Term Frequency
Inverse Document
Frequency

Accuracy vs. Training set size (1/3 withheld for test)

Expectation Maximization (EM) Algorithm

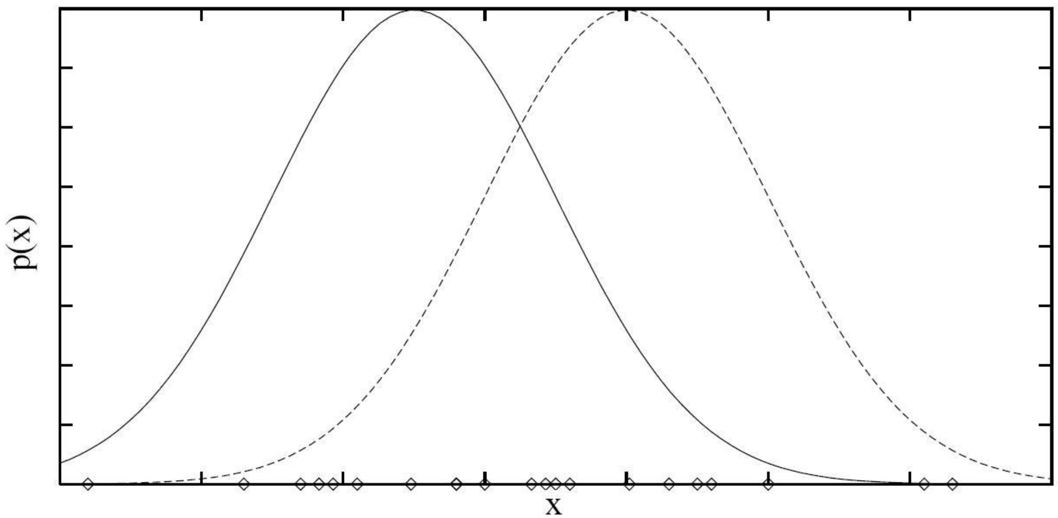
When to use:

- Data is only partially observable
  - Unsupervised clustering (target value unobservable)
  - Supervised learning (some instance attributes unobservable)

Some uses:

- K-Means Clustering
- pLSA
- Train Bayesian Belief Networks
- Unsupervised clustering (AUTOCLASS)
- Learning Hidden Markov Models

Generating Data from Mixture of *k* Gaussians



Example:
x denotes the height of a person. Male and female height distributions are modelled each by a Gaussian distribution. However, we only have height values, but no gender attributes. Task: We want to estimate the gender-specific height distributions.

Each instance *x* generated by

1. Choosing one of the *k* Gaussians with uniform probability
2. Generating an instance at random according to that Gaussian

Gaussian: $N(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} Exp(-\frac{(x-\mu)^2}{2\sigma^2})$

Gaussian Mixture: $GMM = \sum_{k=1}^{K} w_i N(x_i; \mu_i, \sigma_i)$ mit $\sum_{k=1}^{K} w_i = 1$

EM for Estimating *k* Means

Given:

- Instances from *X* generated by mixture of *k* Gaussian distributions (our example has $k = 2$)
- Unknown means $\langle \mu_1, \dots, \mu_k \rangle$ of the *k* Gaussians
- Don't know which instance x_i was generated by which Gaussian, but each is equally likely (i.e., $w_k = 1/K \ \forall k$)
- Assume, all *k* Gaussians have same variance σ^2, i.e. $\sigma_k^2 = \sigma^2 \ \ \forall k$

Determine:

- Maximum likelihood estimates of $\langle \mu_1, \dots, \mu_k \rangle$, i.e.,

$$h_{ML} = \left\langle \mu_1^{ML}, \dots, \mu_k^{ML} \right\rangle = arg \max_{h \in H} P(D|h)$$

Think of full description of each instance as $y_i = \langle x_i, z_{i1}, \dots, z_{ik} \rangle$, where

- z_{ij} is 1 if x_i generated by jth Gaussian
- x_i observable
- z_{ij} unobservable

EM for Estimating *k* Means

EM Algorithm: Pick random initial $h = \langle \mu_1, \ldots, \mu_k \rangle$, then iterate

- **E step**: Calculate the expected value $E[z_{ij}]$ of each hidden variable z_{ij}, assuming the current hypothesis $h = \langle \mu_1, .., \mu_k \rangle$ holds.

$$E[z_{ij}] = \frac{p(x = x_i \mid \mu = \mu_j)}{\sum_{k=1}^{K} p(x = x_i \mid \mu = \mu_k)} \cdot \frac{1/K}{1/K} = \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2}}{\sum_{n=1}^{2} e^{-\frac{1}{2\sigma^2}(x_i - \mu_k)^2}}$$

- **M step**: Calculate new maximum likelihood hypothesis $h' = \langle \mu_1', \ldots, \mu_k' \rangle$, assuming the value taken on by each hidden variable z_{ij} is its expected value $E[z_{ij}]$calculated above. Replace $h = \langle \mu_1, .., \mu_k \rangle$ by $h' = \langle \mu_1', \ldots, \mu_k' \rangle$.

$$\mu_j \leftarrow \frac{\sum_{i=1}^{m} E[z_{ij}] x_i}{\sum_{i=1}^{m} E[z_{ij}]}$$

Can be proven: Each iteration increases $P(D|h)$ unless it is a local maximum.

vorrechnen

General EM Problem

Given:

- Observed data $X = \{x_1, \ldots, x_m\}$
- Unobserved data $Z = \{z_1, \ldots, z_m\}$ with $z_i = \{z_{i1}, \ldots, z_{ik}\}$
- Parameterized probability distribution $P(Y|h)$, where
  - $Y = \{y_1, \ldots, y_m\}$ is the full data $y_i = x_i \cup z_i$
  - h are the parameters

Determine:

- h that (locally) maximizes $E[\ln P(Y|h)]$

Many uses:

- Train Bayesian belief networks
- Unsupervised clustering (e.g., *k* means)
- Hidden Markov Models

General EM Method

Define likelihood function *Q(h' | h)* which calculates $Y = X \cup Z$ using observed *X* and current parameters *h* to estimate *Z*

$$Q(h' \mid h) \leftarrow E[\ln P(Y \mid h') \mid h, X]$$

EM Algorithm:

Estimation (E) step: Calculate *Q(h'|h)* using the current hypothesis *h* and the observed data *X* to estimate the probability distribution over *Y*.

$$Q(h' \mid h) \leftarrow E[\ln P(Y \mid h') \mid h, X]$$

Maximization (M) step: Replace hypothesis *h* by the hypothesis *h’* that maximizes this *Q* function.

$$h \leftarrow \arg\max_{h'} Q(h' \mid h)$$

EM Algorithm - Summary

Converges to local maximum likelihood h and provides estimates of hidden variables z_{ij}

In fact, local maximum in $E[\ln P(Y|h)]$

- Y is the **complete** (observable plus unobservable variables) data
- $P(Y|h)$ is the likelihood of the **full** data Y given the hypothesis h
- Maximizing $P(Y|h)$ and $\ln P(Y|h)$ yields the same result
- Expected value is taken over all possible values of unobserved variables in Y