Probabilistic Reasoning

Ole-Christoffer Granmo

University of Agder

E-mail: `ole.granmo@uia.no`

August 31, 2012

Outline

- Bayesian Inference
- Probability Theory
- Naive Bayesian Classifier
- Text Classification

Bayesian Inference I

- **Bayesian inference:** Statistical inference in which *observations* are used to update the probability that a *hypothesis* may be *true*
- *Remark:* We are interested in using Bayesian inference for Pattern Recognition

Bayesian Inference II

- **In the courtroom**
  - Hypothesis: *The defendant is guilty*
  - Observations: *E.g., DNA evidence*
- **Text analysis**
  - Hypothesis: *The article discusses mobile phones*
  - Observations: *Words occurring in article*
- **Medical diagnosis**
  - Hypothesis: ?
  - Observations: ?

Motivating a Bayesian Approach I

- **Cox's Theorem:** Any system for *plausible reasoning* intended to ensure

  — consistency with classical deductive logic

  — correspondence with commonsense reasoning

  is *isomorphic* to probability theory

- **Question:** Why consider anything else?

Motivating a Bayesian Approach II

> *"I spent about six months writing software that looked for individual spam features before I tried the statistical approach. What I found was that recognizing that last few percent of spams got very hard, and that as I made the filters stricter I got more false positives. [...] I don't know why I avoided trying the statistical approach for so long. I think it was because I got addicted to trying to identify spam features myself, as if I were playing some kind of competitive game with the spammers. [...] When I did try statistical analysis, I found immediately that it was much cleverer than I had been."*

Paul Graham — Author of a Plan for Spam

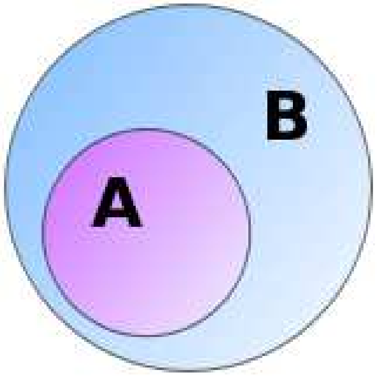
Motivating a Bayesian Approach

- Bayesian inference provides a basis for practical pattern recognition and inference algorithms:
  - Naive Bayes classifier
  - Bayesian belief networks
- Bayesian inference provides a useful conceptual framework
  - Provides "gold standard" for evaluating other pattern recognition and learning algorithms

Probability Theory

- Probability is the likelihood that something *is the case* or *will happen*
- Probability theory is used extensively in areas such as statistics, mathematics, science and philosophy
- The purpose is to draw conclusions about the likelihood of potential events and the underlying mechanics of complex systems

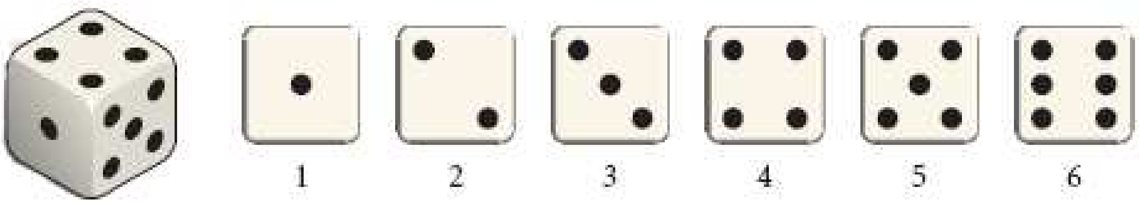
Sample Space and Events



- A sample space is a set of all possible outcomes for an activity or experiment
  - — **Rolling a Die:** $\{1, 2, 3, 4, 5, 6\}$
  - — **Tossing a Coin:** $\{Heads, Tails\}$
  - — **Randomly Selecting a Word from a Document:** $\{A, An, Able, Ability, Abler, Ablest, Ably, \ldots\}$
- Any *subset* of the sample space is usually called an event
  - — **Example of event:** Rolling an even number with a die, i.e., $\{2, 4, 6\}$
- **Question:** What is the sample space when two words are selected at random from a document?

Classical Definition of Probability

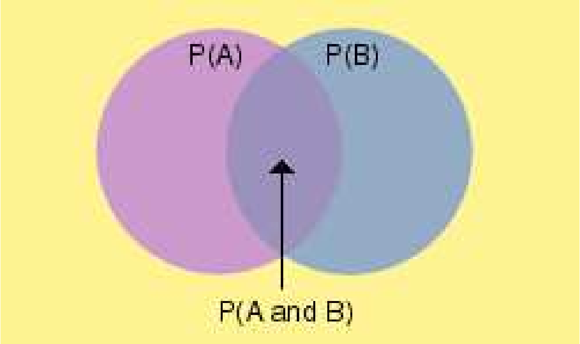
If A random experiment can result in N mutually exclusive and equally likely outcomes;
And N_A of these outcomes result in the occurrence of the event A
Then The probability of A is defined by $P(A) = \frac{N_A}{N}$



Example:

$$P(``Rolling\ an\ even\ number\ with\ a\ die'') = P(\{2,4,6\}) = \frac{3}{6} = 0.5$$

Joint Probability



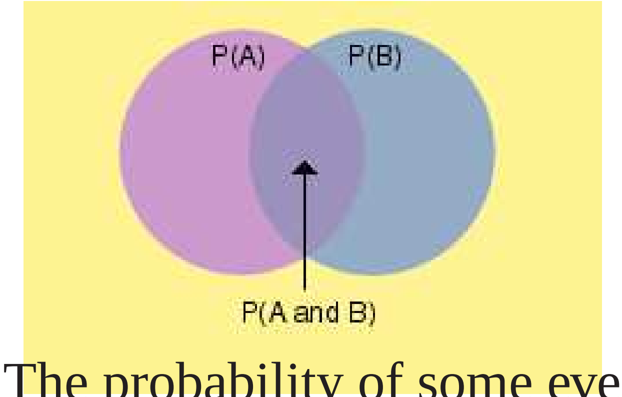
Joint Probability: The probability of two events in conjunction (both events together)

The joint probability of A and B is written $P(A \wedge B)$ or $P(A, B)$

Example:

$$P(\text{``}Rolling\ an\ even\ number'' \wedge \text{``}Rolling\ 2'') = P(\{2,4,6\} \wedge \{2\}) = ?$$

Conditional Probability



Conditional Probability: The probability of some event A, given the occurrence of some other event B

Conditional probability is written $P(A|B)$, and is read *"the probability of A, given B"*

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

Question: What is the probability of getting a 2 when tossing a die, given that the outcome of the toss is even?

$$P(\{2\}|\{2,4,6\}) = \frac{P(\{2\} \wedge \{2,4,6\})}{P(\{2,4,6\})} = ?$$

Conditional Independence

- Two events A and B are independent if and only if $P(A \wedge B) = P(A)P(B)$
- Two events A and B are conditionally independent given a third event C precisely if A and B are independent events given C:

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

Bayes' Theorem

Bayes' theorem tells how to update or revise beliefs in light of new evidence

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

- $P(h)$ = prior probability of hypothesis h
- $P(o)$ = prior probability of observations o
- $P(h|o)$ = probability of h given o
- $P(o|h)$ = probability of o given h

Choosing Hypotheses

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

We generally want to identify the most probable hypothesis, which we call the *maximum a posteriori* hypothesis h_{MAP}:

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

Example: Bayesian Inference in the Courtroom I

- Let h be the event that the defendant is guilty and $\neg h$ be the event that he is innocent
- Let o be the event that the defendant's DNA matches DNA found at the crime scene
- Let $P(o|h) = 1.0$ be the probability of seeing event o assuming that the defendant is guilty
- Let $P(h) = 0.3$ be the juror's personal estimate of the probability that the defendant is guilty, based on the evidence other than the DNA match
- Let $P(o|\neg h) = 10^{-6}$ be the probability that an innocent person chosen at random would have DNA that matched that at the crime scene

Bayes' Theorem tells us that we can calculate $P(h|o)$ — the probability that the defendant is guilty assuming the DNA match event o:

$$P(h|o) = \frac{P(h)P(o|h)}{P(o)} = \frac{P(h)P(o|h)}{P(o,h) + P(o,\neg h)} = \frac{P(h)P(o|h)}{P(h)P(o|h) + P(\neg h)P(o|\neg h)}$$

Example: Bayesian Inference in the Courtroom II

$$P(h|o) = \frac{0.3 \times 1.0}{0.3 \times 1.0 + 0.7 \times 10^{-6}} = 0.99999766667$$

Naive Bayes Classifier I

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
{Sunny, Rainy}	{Warm, Cold}	{Normal, High}	{Weak, Strong}	{Cool, Warm}	{Change, Same}	{Yes, No}

- Let $H = h_j \in \{h_1, h_2, \ldots, h_m\}$ be the hypotheses under consideration [†]
- Let $\langle O_1 = o_1, O_2 = o_2, \ldots, O_n = o_n \rangle$ be the different kinds of observations that have been made
- Then the most probable hypothesis is:

$$h_{MAP} = \underset{h_j \in H}{\operatorname{argmax}} P(h_j | o_1, o_2 \ldots o_n)$$

$$\begin{aligned} h_{MAP} &= \underset{h_j \in H}{\operatorname{argmax}} \frac{P(o_1, o_2 \ldots o_n | h_j) P(h_j)}{P(o_1, o_2 \ldots o_n)} \\ &= \underset{h_j \in H}{\operatorname{argmax}} P(o_1, o_2 \ldots o_n | h_j) P(h_j) \end{aligned}$$

[†] We assume that the hypotheses are *mutually exclusive* (cannot occur together) and *exhaustive* (covers all cases)

Naive Bayes Classifier II

Naive Bayes assumption:

$$P(o_1, o_2 \ldots o_n | h_j) = \prod_i P(o_i | h_j)$$

which gives

Naive Bayes classifier: $h_{NB} = \operatorname*{argmax}_{h_j \in H} P(h_j) \prod_i P(o_i | h_j)$

How to Find the Probabilities?

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

- **Solution 1:** Fix the probabilities based on expert knowledge
- **Solution 2:** Estimate the probabilities using a set of example data (training set)
  - $\hat{P}(\text{EnjoySpt} = \text{Yes}) = \frac{3}{4} = 0.75$
  - $\hat{P}(\text{Temp} = \text{Warm}|\text{EnjoySpt} = Yes) = \frac{3}{3} = 1.0$
  - $\hat{P}(\text{Sky} = \text{Sunny}|\text{EnjoySpt} = No) = ?$

Naive Bayes Algorithm

```
Naive_Bayes_Learn($examples$)
    For each target value $h_j$
        $\hat{P}(h_j) \leftarrow$ estimate $P(h_j)$
        For each observation value $o_i$ of observation $O_i$
            $\hat{P}(o_i|h_j) \leftarrow$ estimate $P(o_i|h_j)$
```

```
Classify_New_Instance($x$)
```

$$h_{NB} = \operatorname*{argmax}_{h_j \in H} \hat{P}(h_j) \prod_i \hat{P}(o_i|h_j)$$

Naive Bayes: Example

Consider *PlayTennis*, and new instance

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

Want to compute:

$$h_{NB} = \operatorname*{argmax}_{h_j \in H} P(h_j) \prod_i P(o_i|h_j)$$

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

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

$$\rightarrow h_{NB} = n$$

Naive Bayes: Subtleties

- Conditional independence assumption is often violated

$$P(o_1, o_2 \ldots o_n|h_j) = \prod_i P(o_i|h_j)$$

- ...but it works surprisingly well anyway. Note don't need estimated posteriors $\hat{P}(h_j|x)$ to be correct; need only that

$$\operatorname*{argmax}_{h_j \in H} \hat{P}(h_j) \prod_i \hat{P}(o_i|h_j) = \operatorname*{argmax}_{h_j \in H} P(h_j) P(o_1 \ldots, o_n|h_j)$$

- However, note that Naive Bayes posteriors often are unrealistically close to 1 or 0

Learning to Classify Text

Why?

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

Naive Bayes is among most effective algorithms

What attributes shall we use to represent text documents??

Learning to Classify Text

Target concept $Interesting? : Document \rightarrow \{+, -\}$

1. Represent each document by vector of words
   - one attribute per word position in document
2. Learning: Use training examples to estimate
   - $P(+)$
   - $P(-)$
   - $P(doc|+)$
   - $P(doc|-)$

Naive Bayes conditional independence assumption

$$P(doc|h_j) = \prod_{i=1}^{length(doc)} P(o_i = w_k|h_j)$$

where $P(o_i = w_k|h_j)$ is probability that word in position i is w_k, given h_j

one more assumption: $P(o_i = w_k|h_j) = P(o_m = w_k|h_j), \forall i, m$

Learning to Classify Text

```
LEARN_NAIVE_BAYES_TEXT($Examples, H$)
  1. Collect all words and other tokens that occur in $Examples$:
     $Vocabulary \leftarrow$ all distinct words and other tokens in $Examples$
  2. Calculate the required $P(h_j)$ and $P(w_k|h_j)$ probability terms:
     For each target value $h_j$ in $H$ do:
  — $docs_j \leftarrow$ subset of $Examples$ for which the target value is $h_j$
  — $P(h_j) \leftarrow \frac{|docs_j|}{|Examples|}$
  — $Text_j \leftarrow$ a single document created by concatenating all members of $docs_j$
  — $n \leftarrow$ total number of words in $Text_j$ (counting duplicate words multiple
    times)
  — for each word $w_k$ in $Vocabulary$
      $*$ $n_k \leftarrow$ number of times word $w_k$ occurs in $Text_j$
      $*$ $P(w_k|h_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$
```

Learning to Classify Text

```
# Calculates P(O | H)

p_word_given_group = {}
for group in posts.keys():

    p_word_given_group[group] = {}

    # Counts the number of words
    for word in vocabulary.keys():
        p_word_given_group[group][word] = 1.0

    for word in posts[group]:
        if vocabulary.has_key(word):
            p_word_given_group[group][word] += 1.0

    # Calculates probabilities
    for word in vocabulary.keys():
        p_word_given_group[group][word] /= len(posts[group]) +
                                               len(vocabulary)
```

Learning to Classify Text

CLASSIFY_NAIVE_BAYES_TEXT(Doc)

- $positions \leftarrow$ all word positions in Doc that contain tokens found in $Vocabulary$

- Return h_{NB}, where

$$h_{NB} = \operatorname*{argmax}_{h_j \in H} P(h_j) \prod_{i \in positions} P(o_i | h_j)$$

Learning to Classify Text

```
# Finds group with max P(O | H) * P(H)
max_group = 0
max_p = 1
for candidate_group in posts.keys():
    # Calculates P(O | H) * P(H) for candidate group
    p = math.log(p_group[candidate_group])
    for word in post_to_be_classified:
        if vocabulary.has_key(word):
            p += math.log(p_word_given_group[candidate_group][word])

    if p > max_p or max_p == 1:
        max_p = p
        max_group = candidate_group
```

Twenty NewsGroups

Given 1000 training documents from each group

Learn to classify new documents according to which newsgroup it came from

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

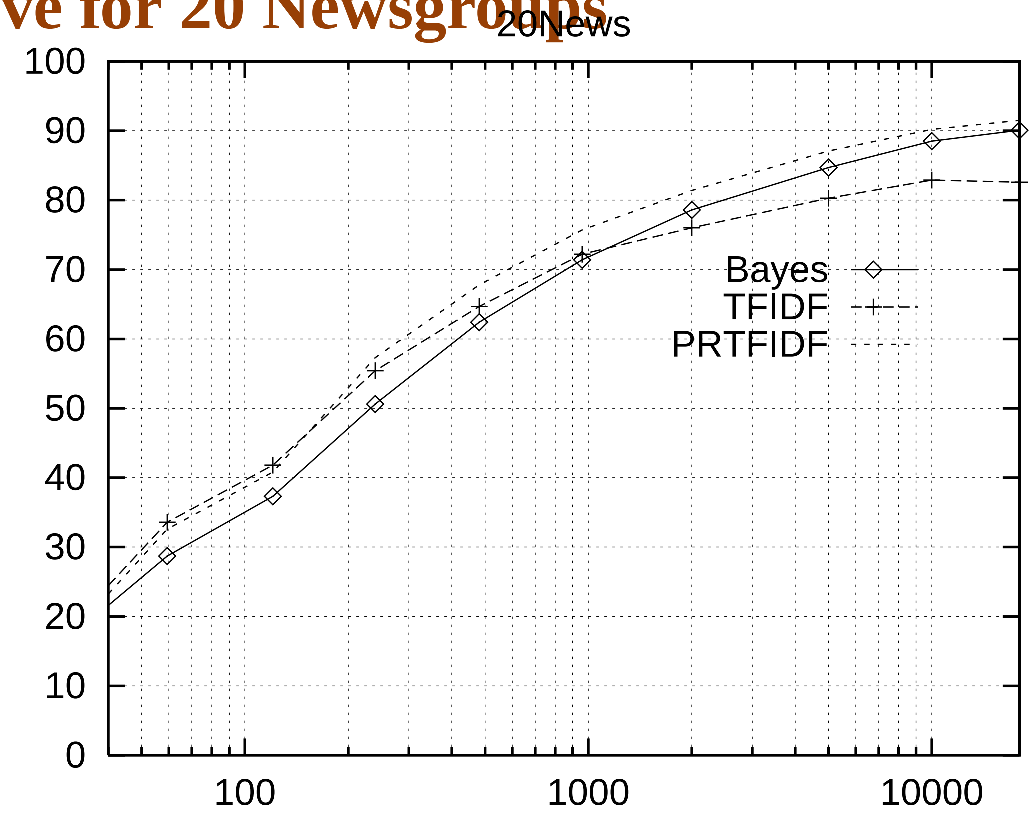
Naive Bayes: 89% classification accuracy

Article from rec.sport.hockey

Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.edu!ogicse!uw
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



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