Bayesian Learning

PROF. JACQUES SAVOY
UNIVERSITY OF NEUCHATEL

Overview

Bayes' Theorem

Naïve Bayes

Smoothing

Missing Value

Numeric Attribute

Text Classification

Bayes Learning

Learning based on the probability theory.

Represent our knowledge through probability estimates for each feature and each category.

Various methods to estimate those probabilities.

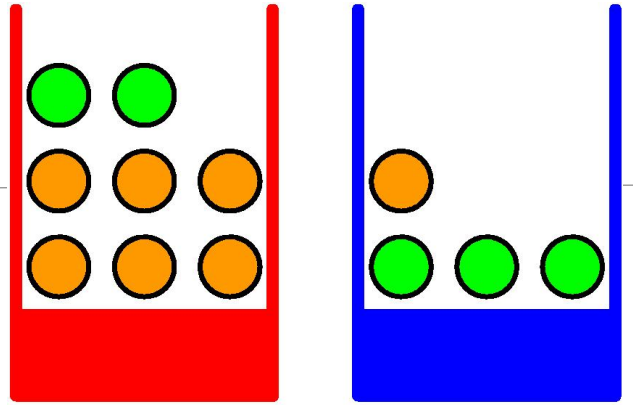
We assume that all provided features are useful for the prediction.

We assume independence between the features.

Simple approach, rather effective in practice.

Use as baseline.

Probability Theory



Basic probability formulae

$0 \leq Prob[A] \leq 1$

Sum rule $Prob[A \cup B] = Prob[A] + Prob[B] - Prob[A \cap B]$

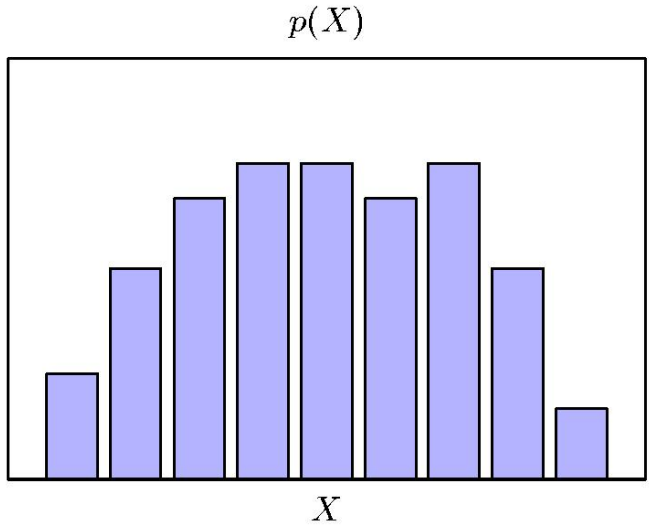
Frequentist approach: an event's probability as the limit of its relative frequency in a large number of trials.

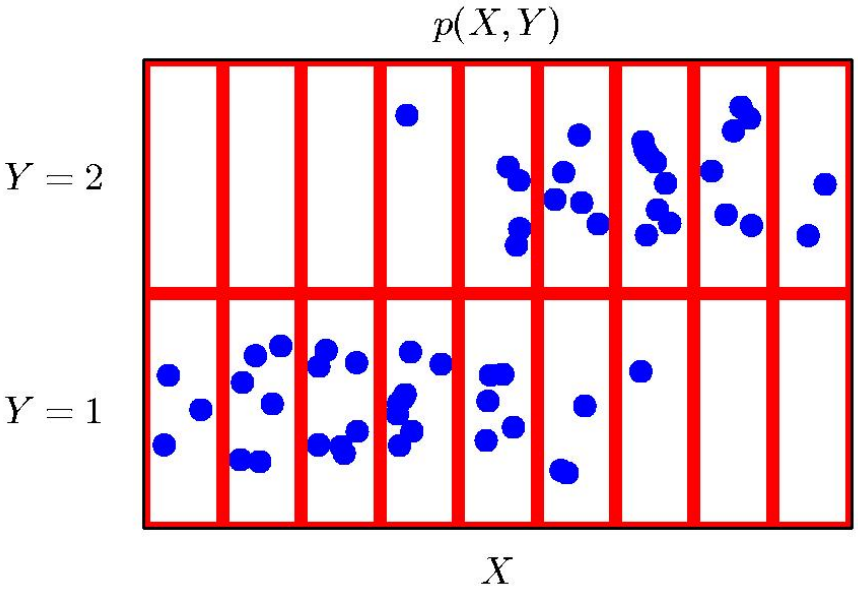
But we cannot always repeat the underling trials (e.g., reliability of a nuclear plant)

Represent as a set of prob. (dice) or as a distribution.

Probability Theory

Consider the distribution of X (prior evidence, without knowing the value of Y).

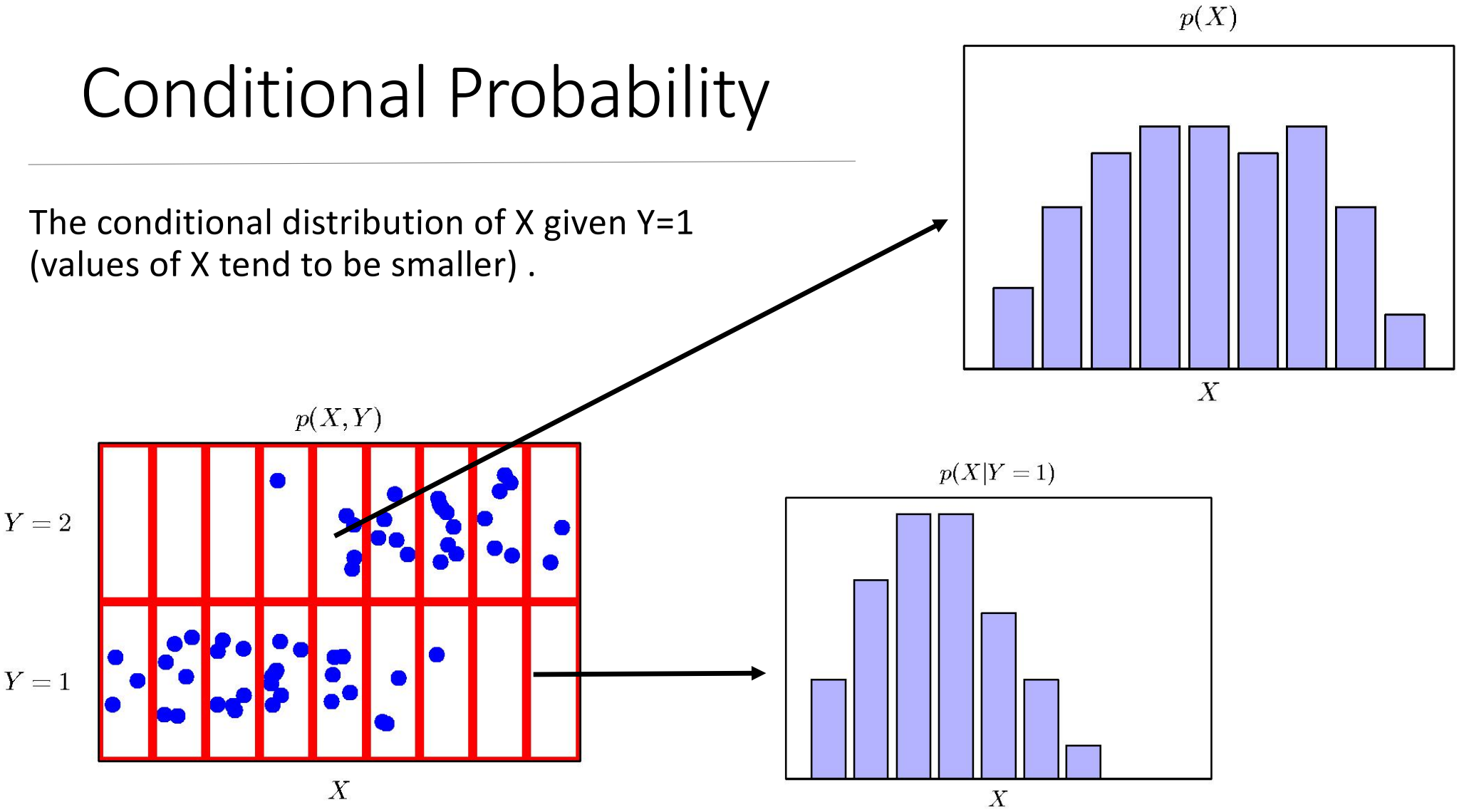




When considering the two possible value for Y, you can guess a better estimate for X.

Conditional Probability

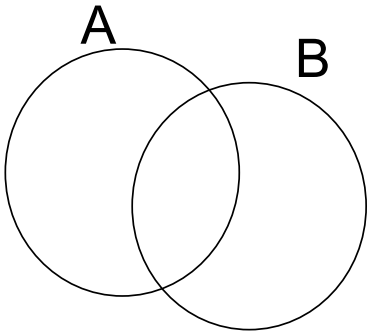
The conditional distribution of X given Y=1 (values of X tend to be smaller) .

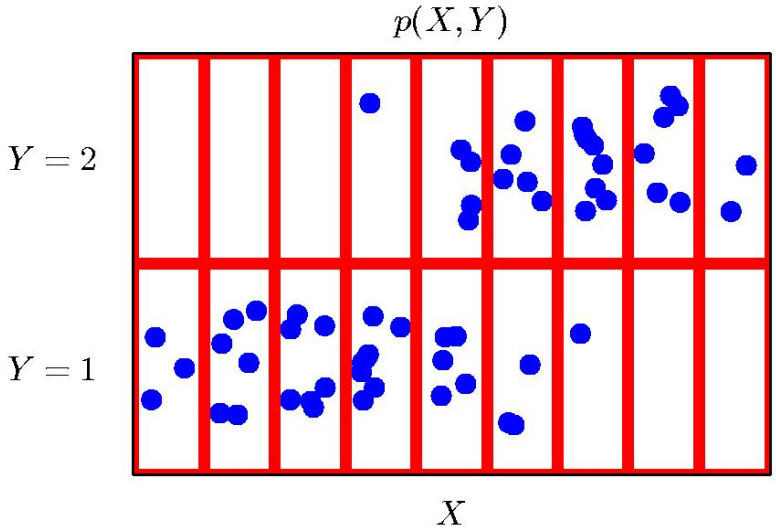


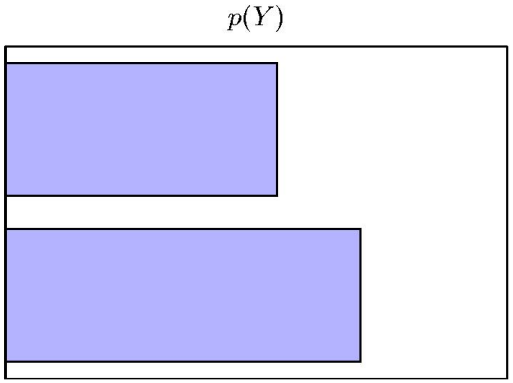
Probability Theory

Another view is the conditional probability

$$Prob[A|B] = \frac{Prob[A \cap B]}{Prob[B]}$$







Probability Theory

Product rule

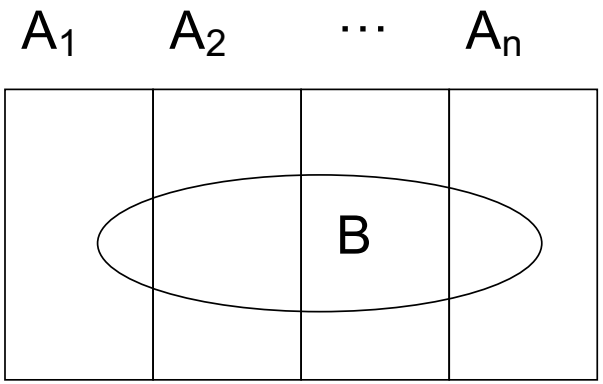
$$Prob[A|B] = \frac{Prob[A \cap B]}{Prob[B]}$$

From

$$Prob[A \cap B] = Prob[A|B] \cdot Prob[B]$$
$$= Prob[B|A] \cdot Prob[A]$$

Total probabilities

$$Prob[B] = \sum_{j=1}^{n} Prob[B|A_j] \cdot Prob[A_j]$$



Bayes' Rule

Thomas Bayes (1702-1761)

Probability of event *H* given evidence *E:*

$$Prob[H|E] = \frac{Prob[E|H] \cdot Prob[H]}{Prob[E]}$$

1. *A priori* probability of *H*: Prob[H]

   Probability of event *before* (any) evidence is seen.

2. *Likelihood* probability: Prob[E|H]

   Probability of the evidence knowing the hypothesis.

3. *A posteriori* probability of *H*: Prob[H|E]

   Probability of event *after* evidence is seen.

Example

In the bar, a person said: “I win with a 7!”

Question: Does this person win when rolling a pair of dice or spinning a roulette?

To have an estimate your need to compute
Prob[dice | "7”] and Prob[roulette | "7”]

How to do that? $$Prob[A|B] = \frac{Prob[A \cap B]}{Prob[B]}$$

How to estimate Prob[dice ∩ "7”] ?

Example

In the bar, a person said: “I win with a 7!”
Does this person win when rolling a pair of dice or spinning a roulette?

Compute Prob[dice | "7”] and Prob[roulette | "7”]

The prior: There is 6 tables, and in 2 they are playing with a roulette.

- Prob[h_{dice}] = 4/6
- Prob[$h_{roulette}$] = 2/6

$$Prob[H|E] = \frac{Prob[E|H] \cdot Prob[H]}{Prob[E]}$$

Evidence:

- What is the chance to obtain a "7" with the dice and the roulette?
  Prob["7" | dice], Prob["7" | roulette]?

Example

We have the prior:

- Prob[h_{dice}] = 4/6
- Prob[$h_{roulette}$] = 2/6

We need to compute the evidence
(having a "7" according to the two hypothesis) :

- Prob["7" | dice] = Prob[e | h_{dice}] = 6/36
- Prob["7" | roulette] = Prob[e | $h_{roulette}$] = 1/37

Next we need to combine these two sources the prior and the likelihood (evidence).

Bayes Theorem

Combining prior probabilities and the likelihood of the data (according to the hypothesis *H*):

$$Prob[H|E] = \frac{Prob[E|H] \cdot Prob[H]}{Prob[E]} \propto Prob[E|H] \cdot Prob[H]$$

In some cases, we just need to determine the most probable hypothesis (and not its corresponding probability).

The ranking of the hypotheses is enough.

Example

We have the prior:

- Prob[h_{dice}] = 4/6
- Prob[$h_{roulette}$] = 2/6

Evidence:

- Prob["7" | dice] = Prob[e | h_{dice}] = 6/36
- Prob["7" | roulette] = Prob[e | $h_{roulette}$] = 1/37

Posteriori:

$$Prob[h_{dice}|e] \propto \frac{6}{36} \cdot \frac{4}{6} = 0.111$$

$$Prob[h_{roulette}|e] \propto \frac{1}{37} \cdot \frac{2}{6} = 0.009$$

Another Example

The candy manufacturer produces large bags of candies.
No specification about the content is given.

10%	h_1	100% cherry
20%	h_2	75% cherry + 25% lime
40%	h_3	50% cherry + 50% lime
20%	h_4	25% cherry + 75% lime
10%	h_5	100% lime

Prior: each hypothesis does not have the same probability.

Evidence: e_i (random variable) is the flavor of the *i*th candy selected from the bag
e_i = *cherry* or *lime* *(in fact, here we will have each e_i = lime).*

Question: predict the flavor of the next piece of candy.

Prior

In our candy example (having a lime candy).

The prior are:

100% cherry	Prob[h_1] = 0.1
75% cherry & 25% lime	Prob[h_2] = 0.2
50% cherry & 50% lime	Prob[h_3] = 0.4
25% cherry & 75% lime	Prob[h_4] = 0.2
100% lime	Prob[h_5] = 0.1

The evidence: the candy found was a lime

Compute the likelihood P[e|h_i]

Posteriori

In our candy example (having e_1 = lime candy).

What is Prob[e|h_i] ? Directly given in the description of the problem.

Combine the likelihood with the prior.

$$Prob[h_1|e] \propto Prob[e|h_1] \cdot Prob[h_1] = 0.0 \cdot 0.1 = 0.0$$
$$Prob[h_2|e] \propto Prob[e|h_2] \cdot Prob[h_2] = 0.25 \cdot 0.2 = 0.05$$
$$Prob[h_3|e] \propto Prob[e|h_3] \cdot Prob[h_3] = 0.5 \cdot 0.4 = 0.2$$
$$Prob[h_4|e] \propto Prob[e|h_4] \cdot Prob[h_4] = 0.75 \cdot 0.2 = 0.15$$
$$Prob[h_5|e] \propto Prob[e|h_5] \cdot Prob[h_5] = 1.0 \cdot 0.1 = 0.1$$

This is proportional to the real probabilities.

Bayes Theorem

In our candy example:

The *most probable* hypothesis (the hypothesis h_i that maximizes P[h_i|e]) is called *maximum a posteriori* or MAP and denoted h_{MAP}.

In our example, h_{MAP} depends on the evidence E.

with $e_1 \rightarrow h_3$ (the first is a lime)
with $e_2 \rightarrow h_4$ (the first two are lime)
with $e_3 \rightarrow h_5$ (forever, we will find only lime candies)

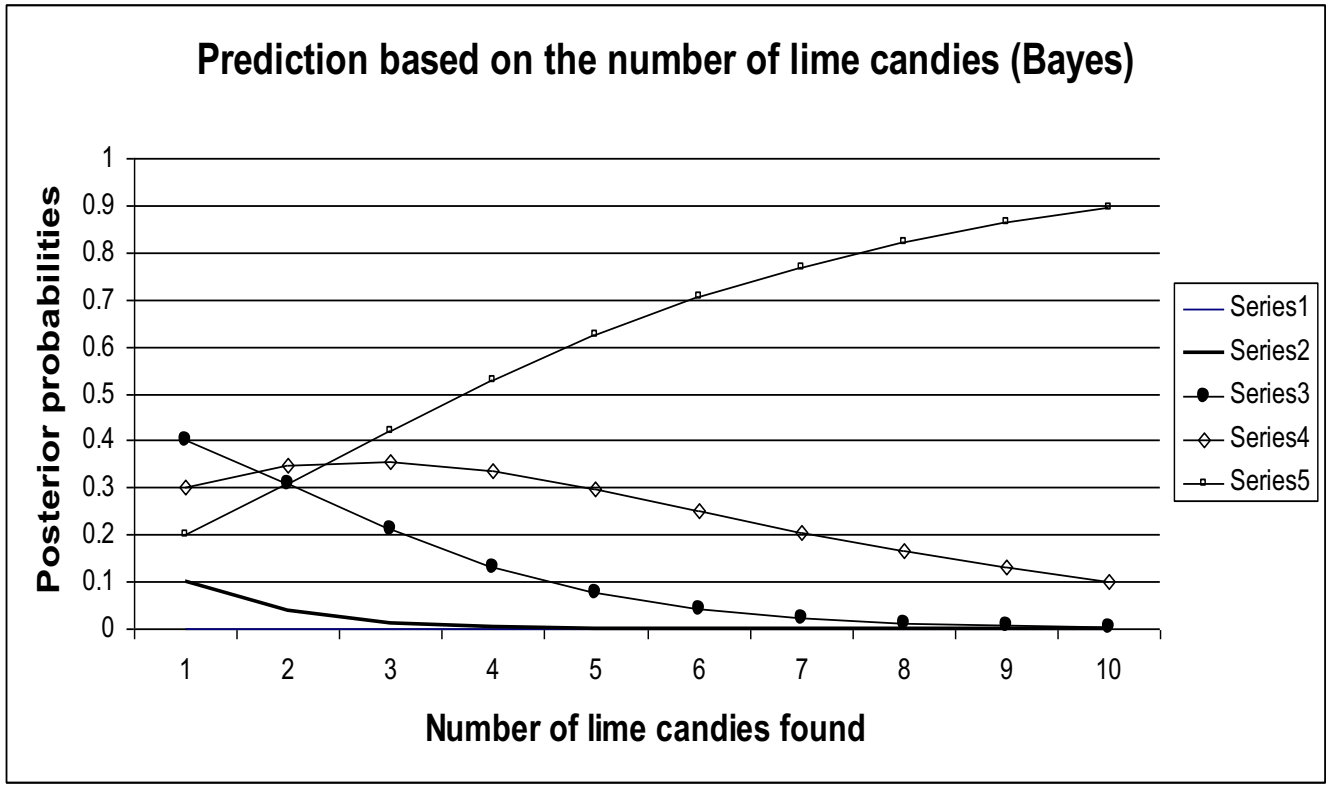
If we consider only the hypothesis *h* that maximizes the likelihood P[e|h_i], such an hypothesis will be denoted h_{ML} and called *maximum likelihood* (ML) hypothesis.

Bayes Theorem

In our candy example

with a sequence of evidence (always lime, lime, lime, …).

The likelihood increases with P[H_i] and P[E|H_i].



Overview

Bayes' Theorem

Naïve Bayes

Smoothing

Missing Value

Numeric Attribute

Text Classification

Bayes Learning

Classification learning: what's the probability of the class given an (new) instance?

- Evidence *E* = new instance
- Event *H* = class value for this new instance

In general, the evidence can be divided into parts (e_i = feature-value),
i.e., the various features E= e_1, e_2, …, e_n, namely e_1 *and* e_2 *and* … *and* e_n.

and we need to classify the new instance.

Applying the probability theory, we can formulate this as:

Bayes Learning

This new instance (e_1, e_2, …, e_n) is classified according to:

$$
\begin{aligned}
h_{MAP} &= \arg\max_{h_j \in H} \; Prob[h_j | e_1, e_2, \ldots e_n] \\
h_{MAP} &= \arg\max_{h_j \in H} \; \frac{Prob[e_1, e_2, \ldots e_n | h_j] \cdot Prob[h_j]}{Prob[e_1, e_2, \ldots e_n]} \\
&= \arg\max_{h_j \in H} \; Prob[e_1, e_2, \ldots e_n | h_j] \cdot Prob[h_j]
\end{aligned}
$$

The computation of $Prob[e_1, e_2, \ldots e_n | h_j]$

is in a *general* case too complex.

Naïve Bayes

The naïve Bayes classifier: the direct estimation is not possible (estimations too imprecise).

Adopt a simplification hypothesis: conditionally independence:

$$Prob[e_1, e_2, \ldots e_n | h_j] \rightarrow \prod_{i=1}^{n} Prob[e_i | h_j]$$

and combined with the prior:

$$h_{NB} = \arg \max_{h_j \in H} \; Prob[h_j] \cdot \prod_{i=1}^{n} Prob[e_i | h_j]$$

Yes the independence assumption is rather unrealistic.

Example: Weather problem

Outlook	Temperature	Humidity	Windy	*Play*
sunny	hot	high	false	*no*
sunny	hot	high	true	*no*
overcast	hot	high	false	*yes*
rainy	mild	high	false	*yes*
rainy	cool	normal	false	*yes*
rainy	cool	normal	true	*no*
overcast	cool	normal	true	*yes*
sunny	mild	high	false	*no*
sunny	cool	normal	false	*yes*
rainy	mild	normal	false	*yes*
sunny	mild	normal	true	*yes*
overcast	mild	high	true	*yes*
overcast	hot	normal	false	*yes*
rainy	mild	high	true	*no*

Example

Prior estimation:
Decision Play: *yes* 9, *no* 5.

Prob[H=yes] = 9/14

Prob[H=no] = 5/14

For the likelihood Prob[e|H] (or Prob[feature-value|H])

Feature by feature

Outlook	yes	no	Temperature	yes	no
sunny	2	3	*high*	2	2
overcast	4	0	*mild*	4	2
rainy	3	2	*cool*	3	1
sunny	2/9	3/5	*high*	2/9	2/5
overcast	4/9	0/5	*mild*	4/9	2/5
rainy	3/9	2/5	*cool*	3/9	1/5

Example

The last two features

Humidity	yes	no	Windy	yes	no
high	3	4	*false*	6	2
normal	6	1	*true*	3	3
high	3/9	4/5	*false*	6/9	2/5
normal	6/9	1/5	*true*	3/9	3/5

The new instance E

Outlook	Temperature	Humidity	Windy	play
sunny	*cool*	*high*	*true*	?

Probabilities

Without the independence assumption, we need to estimate (for our example)

Prob[outlook = *sunny* Λ temperature = *cool* Λ humidity = *high* Λ windy = *true* | H=*yes*]

Prob[outlook = *sunny* Λ temperature = *cool* Λ humidity = *high* Λ windy = *true* | H=*no*]

Too many data are needed to obtain an accurate estimation
(even for binary dependence e.g., "outlook = *sunny* Λ temperature = *cool*").

Thus, we assume that knowing the value of one attribute says nothing about the value of another (not fully realistic).

Probabilities

Thus for our example, we compute the likelihood x prior of the possible outcomes

Prob[*yes*|E] = Prob[outlook = *sunny* | *yes*] ·
Prob[temperature = *cool* | *yes*] ·
Prob[humidity = *high* | *yes*] ·
Prob[windy = *true* | *yes*] ·
Prob[H=*yes*] / Prob[E]

$$\text{For } yes = \left[\frac{2}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14}\right] / Prob[E]$$

Probabilities

But we need to consider the other outcomes

$$\text{For } yes = \frac{2}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14} = 0.0053$$

$$\text{For } no = \frac{3}{5} \cdot \frac{1}{5} \cdot \frac{4}{5} \cdot \frac{3}{5} \cdot \frac{5}{14} = 0.0206$$

We can stop here (take the max) or convert them into probability (normalization)

$$Prob[yes] = \frac{0.0053}{(0.0053+0.0206)} = 0.205$$

$$Prob[no] = \frac{0.0206}{(0.0053+0.0206)} = 0.795$$

Naïve Bayes

Problem: decide whether to wait for a table at a restaurant, based on the following attributes:

1. Alternate: is there an alternative restaurant nearby?
2. Bar: is there a comfortable bar area to wait in?
3. Fri/Sat: is today Friday or Saturday?
4. Hungry: are we hungry?
5. Patrons: number of people in the restaurant (None, Some, Full).
6. Price: price range ($, $$, $$$).
7. Raining: is it raining outside?
8. Reservation: have we made a reservation?
9. Type: kind of restaurant (French, Italian, Thai, Burger).
10. WaitEstimate: estimated waiting time (0-10, 10-30, 30-60, >60).

Naïve Bayes

Example with a (simplified) restaurant

situations where I will / won't wait for a table

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

Classification of examples is positive (T) or negative (F).

Naïve Bayes

Example with the (simplified) restaurant
Feature selection: remove some uninteresting attributes (noisy?)

Attribute:

```
     Patron, Hungry, Type,     Fri/Sat
E = (full,    true,  French,   false)
```

Prior WillWait = *yes* or *no*
12 cases Prob[*yes*] = 6/12 = 0.5
Prob[*no*] = 6/12 = 0.5

Naïve Bayes

Likelihood:

$$Prob[Patron = full|yes] = \frac{|Patron = full \cap Wait = yes|}{|Wait = yes|}$$

Prob[Patron = *full* | *yes*] = 2/6
Prob[Patron = *full* | *no*] = 4/6

Prob[Hungry = *yes* | *yes*] = 5/6
Prob[Hungry = *yes* | *no*] = 2/6

Prob[Type = *French* | *yes*] = 1/6
Prob[Type = *French* | *no*] = 1/6
Prob[Fri/Sat = *false* | *yes*] = 4/6
Prob[Fri/Sat = *false* | *no*] = 3/6

Naïve Bayes

Likelihood:

Prob[Patron = *full* | *yes*] · Prob[Hungry = *yes* | *yes*] · Prob[Type = *French* | *yes*] · Prob[Fri/Sat = *false* | *yes*] = 2/6 · 5/6 · 1/6 · 2/6 = 20 / 1296

Prob[Patron = *full* | *no*] · Prob[Hungry = *yes* | *no*] · Prob[Type = *French* | *no*] · Prob[Fri/Sat = *false* | *no*]
= 4/6 · 2/6 · 1/6 · 3/6 = 24 / 1296

Posterior:

Prob[*yes*] · Prob[E | *yes*] = ½ · 20 / 1296
Prob[*no*] · Prob[E| *yes*] = ½ · 24 / 1296

Overview

Bayes' Theorem

Naïve Bayes

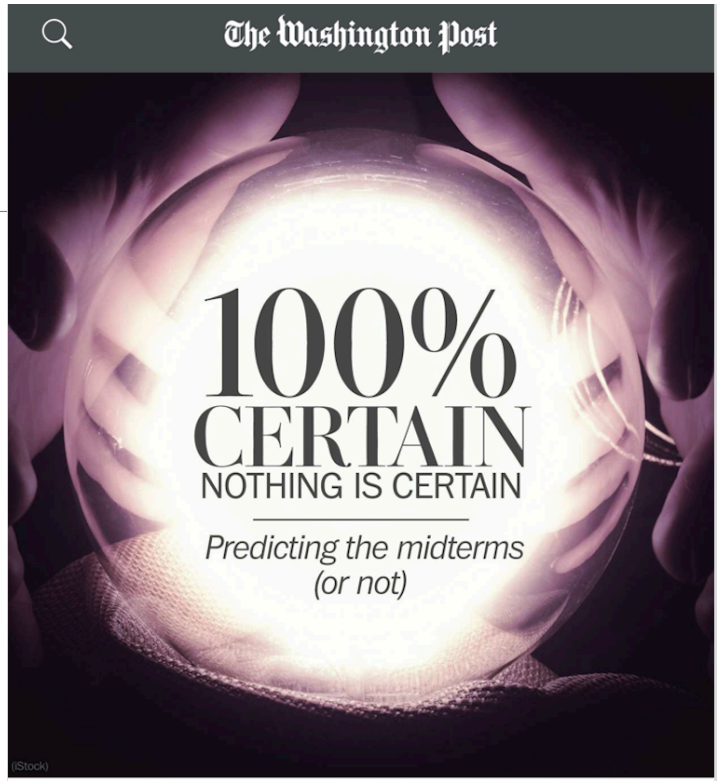
Smoothing

Missing Value

Numeric Attribute

Text Classification

Better Probability Estimates



What if a pair feature-value does not occur with every class value?

(Using our forecast problem:
e.g. "Outlook= overcast" for class "no").

Probability will be zero!

Prob[humidity=*high* | *yes*] = 0.0

A posteriori probability will also be zero!

Prob[*yes* | E] = 0.0 (No matter how likely the other values are!)
(similar with Prob[*no* | E] = 1.0)

Really realistic?

Better Probability Estimates

Apply a smoothing method.

Various possible approaches.

Simple: Laplace (or Add-one).
Add 1 to the count for every feature-value class combination (*Laplace estimator).*

Result: probabilities will never be zero! (also: stabilizes probability estimates).

Very important when dealing with text categorization problem (see later).

Better Probability Estimates

Smoothing techniques

Remedy: add 1 to the count for every attribute value class combination (*Laplace estimator)*

Result: probabilities will never be zero! (also: stabilizes probability estimates).

Before	With Laplace smoothing
Prob[sunny | Yes] = 2 / 9	Prob[sunny | Yes] = (2+1) / (9+3)
Prob [overcast | Yes] = 4 / 9	Prob [overcast | Yes] = (4+1) / (9+3)
Prob[rainy | Yes] = 3 / 9	Prob[rainy | Yes] = (3+1) / (9+3)
Prob [overcast | No] = 0 / 5	Prob [overcast | No] = (0+1) / (5+3)

Better Probability Estimates

In some cases adding a constant different from 1 might be more appropriate (below $\mu = 1$)

Example: attribute *outlook* for class *yes*

sunny	overcast	rainy
$\frac{2+\mu/3}{9+\mu}$	$\frac{4+\mu/3}{9+\mu}$	$\frac{3+\mu/3}{9+\mu}$

Weights don't need to be equal (but they must however sum to 1)

sunny	overcast	rainy
$\frac{2+\mu \cdot p_1}{9+\mu}$	$\frac{4+\mu \cdot p_2}{9+\mu}$	$\frac{3+\mu \cdot p_3}{9+\mu}$

Overview

Bayes' Theorem

Naïve Bayes

Smoothing

Missing Value

Numeric Attribute

Text Classification

Missing Value

Training: instance is not included in frequency count for attribute value-class combination.

Classification: attribute will be omitted from calculation.

Example:

outlook	temperature	humidity	windy	play
?	*cold*	*high*	*true*	?

Ignore it?

Remove the corresponding records?

Use NA as a possible value?

Missing Value

Ignore the missing attribute

The corresponding likelihood:

$$\text{For } yes = \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14} = 0.0238$$

$$\text{For } no = \frac{1}{5} \cdot \frac{4}{5} \cdot \frac{3}{5} \cdot \frac{5}{14} = 0.0343$$

The resulting probabilities

$$Prob[yes] = \frac{0.0238}{(0.0238+0.0343)} = 0.41$$

$$Prob[no] = \frac{0.0343}{(0.0238+0.0343)} = 0.59$$

Overview

Bayes' Theorem

Naïve Bayes

Smoothing

Missing Value

Numeric Attribute

Text Categorization

Numeric Attribute

Usual assumption: attributes have a *normal* or *Gaussian* probability distribution (given the class)

The *probability density function* for the normal distribution is defined by two parameters:

- *Sample mean* μ
- *Standard deviation* σ

Then the density function *f(x) is*

$$f(x) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} e^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}}$$

with $\quad -\infty < x < +\infty$

Statistics for the Weather Data

Temperature (in C)		Humidity	
yes	no	*yes*	no
29, 28, 21, 20, …	27, 22, 24, 22, …	*65, 70, 70, 75…*	70, 85, 90, 91…
μ = 22.3	μ = 24.4	μ = 79	μ = 86
σ = 3.7	σ = 2.2	σ = 10.2	σ = 9.7

With R:
Gaussian distribution is assumed
mean and standard deviation given

$$f(temperature = 27|yes) = \frac{1}{3.7 \cdot \sqrt{2 \cdot \pi}} \cdot e^{-\frac{(27-22.3)^2}{2 \cdot 3.7^2}} = 0.04812$$

Missing values during training are not included in calculation of mean and standard deviation

Numeric Attribute

A new day

Outlook	Temperature	Humidity	Windy	Play
sunny	66	90	*true*	?

The corresponding likelihood:

$$Prob[E|yes] \propto \frac{2}{9} \cdot 0.048 \cdot 0.0219 \cdot \frac{3}{9} \cdot \frac{9}{14} = 0.0000501$$

$$Prob[E|no] \propto \frac{3}{5} \cdot 0.0902 \cdot 0.0378 \cdot \frac{3}{5} \cdot \frac{5}{14} = 0.0004384$$

The resulting probabilities

$$Prob[E|yes] = \frac{0.0000501}{0.00003501 + 0.0004384} = 0.10256$$

$$Prob[E|no] = \frac{0.0004384}{0.0000501 + 0.0004384} = 0.89744$$

Numeric Attribute

Usually the previous technique is working well, without being fully correct.

The density function f(x) is not directly a probability, taking a small surface ($x \pm \varepsilon$, with $\varepsilon = 0.5$).

A correct way : the repartition function F(x) which corresponds to Prob[$x \leq X$] = F(x).

Need to standardize the values to obtain the Normal distribution Z with mean = 0, standard deviation = 1.

$$Z = \frac{x-\mu}{\sigma} = \frac{x-\bar{x}}{S_x}$$

Overview

Bayes' Theorem

Naïve Bayes

Smoothing

Missing Value

Numeric Attribute

Text Classification

Text Classification

Example: Spam detection

Spam

"Dear sir,
We want to transfer to overseas $ 126,000.000.00 USD (one hundred and twenty six million United States Dollars) from a Bank in Africa, I want to ask you to quietly look for a reliable and honest person who will be capable and fit to provide either an existing …"

Legitimate email (Ham).

Text Classification

Hypotheses: {Spam, Ham}.

Evidence: a document

- The document is treated as a set (or bag) of words.

Knowledge

- P(Spam)
  - The prior probability of an e-mail message being a spam.
  - How to estimate this probability?
- P(w|Spam)
  - the probability that a word is *w* if we know *w* is chosen from a spam.
  - How to estimate this probability?

Text Classification

Based on a document (web page) can you assign to it one of these categories (no overlap)

20 categories

comp.graphics
comp.os_ms-window
comp.sys_ibm.pc
comp.sys.mac
comp.windows.x
sci.space
sci.electronics
misc.forsale
rec.authors
rec.motorcycles
rec.sport.baseball
rec.sprot.hockey
sci.crypt
sci.med
soc.religion.christian
talk.politics.guns
talk.politics.mideast
talk.politics.misc
talk.religion.misc
alt.atheism

Authorship Attribution

Did Shakespeare write all of his plays?

- Various authors including Bacon and Marlowe are said to have written parts or all of several plays
- "Shakespeare" may even be a nom-de-plume for a group of writers?

Plays written by more than one author

- *Edward III* – Shakespeare? & Kyd?
- *Two Noble Kinsmen* – Shakespeare & Fletcher
- *Henry VIII* – Shakespeare & Fletcher?

Other ex. E-mail, Web page, Twitter

Profiling: Male/Female?

Text 1 (blog)

Yesterday we had our second jazz competition. Thank God we weren't competing. We were sooo bad. Like, I was so ashamed, I didn't even want to talk to anyone after. I felt so rotton, and I wanted to cry, but...it’s ok.

Profiling: Male/Female?

Text 2 (blog)

My gracious boss had agreed to let me have one week off of "work." He did finally give me my report back after eight freakin' days! Now I only have the rest of this week and then one full week after my vacation to finish this damned thing.

Sentiment Analysis

Movie review

- Unbelievably disappointing

- Full of zany characters and richly applied satire, and some great plot twists

- This is the greatest screwball comedy ever filmed

- It was pathetic. The worst part about it was the boxing scenes.

Bot or Human?

Is this tweet sent by a bot or a human?

1. RT @DeepLearn007 "Fintech trends: The rise of AI | Fintech 2017 Recap #AI #MachineLearning #BigData #Fintech #ML #Banking #tech
2. yewwNEWS CNNgo - https://t.co/613g5WGzZ2 https://t.co/gdApg378LY, see more https://t.co/Ue2IiIqto1
3. Learn About GPS Technology and Fitness Watches http://t.co/lwVbKuJG30
4. I'm going on a buying trip to Paris tomorrow with my boss!!! So grateful to have a job that I love! 🗼🇫🇷🥐🍷🛍️👜👛 https://t.co/s8qmNs6Pko
5. The best thing about today! 🎈🎈🎈 https://t.co/3w0gGKPFSL
6. Important piece on the death of democracy in Hong Kong. https://t.co/35SVTBcRnC

Text Classification

From a news: “We were really concerned” said the Canadian PM “that the relationship between our countries …”

How can we define the features needed to classify it?

Using the words? with their position? Part-of-speech?

Idea: limited to content-bearing tokens

Remove very frequent tokens (the, in, of, is, …)
(determinants, prepositions, conjunctions, pronouns)

May remove words occurring only once or twice

May remove the final ‘-s’ (and ‘-ed’ or ‘-ing’)

Do not account for token position.

Text Classification

From the news article

"We were really concerned" said the Canadian PM "that the relationship between our countries …"

we obtain:

" concern " said Canadian PM " relationship country …"

Can the punctuation be useful?

For example classify between opinioned and non-opinionated sentences (or between positive, negative, mixed opinionated sentences).

Be careful: Large number of predictors (p) and in some cases n < p!

Naive Bayes for Text Categorization

Different implementations of the naïve Bayes model.

1. Multivariate Bernoulli model: both presence and *absence* of terms are taken into account.
2. Multinomial model: term frequency (*tf*) is taken into account (and not the absence).
3. Simplification of the multinomial model: the *tf* component is replaced by a binary weight.

Multivariate Bernoulli Model

The estimation of the prior probabilities is the same (relative frequency of each class).

For words (features), we will use the fraction of the documents in the category with this word (feature).

We will ignore the number of occurrences (only the *presence* / *absence* information is used). Thus the term Bernoulli.

DocID	Words in documents	Category
1	swiss cheese cheese	1
2	swiss watch jewelry	1
3	swiss swiss	1
4	paris watch paris	2
Query	swiss watch swiss	?

Multivariate Bernoulli Model

The prior probabilities (two categories)

Prob[Category=1] = 3/4

Prob[Category=2] = ¼

Distribution of the words in the two categories

Word	Category 1	Category 2
swiss	I I I	
cheese	I	
watch	I	I
jewelry	I	
paris		I
$\|C\| = 2$	$\|doc_1\| = 3$	$\|doc_2\| = 1$

Multivariate Bernoulli Model

Estimating the probabilities (for two categories) + Laplace smoothing

$$\text{Prob}[\text{word} \mid \text{Cat}=1] = (df_{w1} + 1) / (|doc_1| + 2)$$

$$\text{Prob}[\text{word} \mid \text{Cat}=2] = (df_{w2} + 1) / (|doc_2| + 2)$$

$$\text{And Prob}[\neg\text{word} \mid \text{Cat}=1] = 1 - \text{Prob}[\text{word} \mid \text{Cat}=1]$$

Word	Prob in Cat=1	Prob in Cat=2
swiss	(3+1) / (3+2)	(0+1) / (1+2)
cheese	(1+1) / (3+2)	(0+1) / (1+2)
watch	(1+1) / (3+2)	(1+1) / (1+2)
jewelry	(1+1) / (3+2)	(0+1) / (1+2)
paris	(0+1) / (3+2)	(1+1) / (1+2)

Two cases:
present / absent

Multivariate Bernoulli Model

Computing the probability of each category for the query text “swiss watch swiss” and this query ignores the terms “cheese, jewelry, paris”

The query is then “swiss watch ¬cheese ¬jewelry ¬paris”

$$\text{Prob}[\text{Cat}=1 \mid \text{query}] \propto \tfrac{3}{4} \cdot 4/5 \cdot 2/5 \cdot (1 - 2/5) \cdot (1 - 2/5) \cdot (1 - 1/5) = 0.06912$$

$$\text{Prob}[\text{Cat}=2 \mid \text{query}] \propto \tfrac{1}{4} \cdot 1/3 \cdot 2/3 \cdot (1 - 1/3) \cdot (1 - 1/3) \cdot (1 - 2/3) = 0.00823$$

$$\text{Prob}[\text{Cat}=1 \mid \text{query}] = 0.06912 / (0.06912 + 0.00823) = 0.894$$

$$\text{Prob}[\text{Cat}=2 \mid \text{query}] = 0.00823 / (0.06912 + 0.00823) = 0.106$$

Text Preprocessing

"Dear Sir,
We want to transfer to overseas $ 126,000.000.00 USD (one hundred and twenty six million United States Dollars) from a Bank in Africa, I want to ask you to quietly look for a reliable and honest person who will be capable and fit to provide either an existing …
Have a nice day."

1. Reduce to lowercase.
2. Remove stopwords (the, a, is, and, or, but, for, in, was, want, can, …).
3. Remove the plural form (stemming).

"transfer oversea $ 126,000.000.00 usd (hundred twenty six million united states dollar) bank africa, quietly look reliable honest person capable provide existing … nice day"

Text Classification

Having a training corpus composed of articles dealing with different categories (modeled as having a target value v_j, j = 1, 2, … *m*).

Assume a binary case: v_j ={positive, negative} or {yes, no}.

Each article can be viewed as composed of sequence of words (tokens) w_k, k = 1, 2, … |V| belonging to a vocabulary V (or size |V|).

We can use the notation $P[a_i = w_k | v_j]$ to indicate the probability of finding in the corresponding document, in position a_i for i = 1, 2, …, *n* the token w_k (the *k*th item extracted from V) knowing that this document belongs to the class v_j
In short, $P[w_k \mid v_j]$, ignoring the position. Only word and target category are useful.

Text Classification

With the naïve Bayes model and having a document, we may compute

$$v_{NB} = arg\ max_{v_j \in V}\ Prob[v_j] \cdot \prod_{i=1}^{n} Prob[w_i|v_j]$$

$$v_{NB} = arg\ Max_{v_j \in V}\ \ Prob[v_j] \cdot Prob[w_1 = "transfer"|v_j] \cdot$$
$$Prob[w_2 = "oversea"|v_j] \cdot \quad \cdot Prob[w_n = "day"|v_j]$$

But with the corresponding probability estimates…

How to estimate the underlying probabilities?

Relatively easy for the prior Prob[v_j].

Multinomial Naïve Bayes

1. Collect all words (& punctuation, emoticons, …) occuring in the corpus C.

   V ← the set of all distinct tokens (or vocabulary) (selection?, stemming?).

2. Compute the probability estimate $P[v_j]$ and $P[w_k|v_j]$ as:

   Prior: $doc_j \leftarrow$ the subset of documents from C having the target value is v_j

   $P[v_j] = |doc_j| \,/\, |C|$

   Likelihood:

   $Text_j$ = concatenation of all documents in the set doc_j

   $n_j \leftarrow$ total number of words in $Text_j$

   for each word w_i in Voc

   $tf_{kj} \leftarrow$ number of times word w_i occurs in $Text_j$

   $P[w_i|v_j] = (tf_{ij} + 1) \,/\, (n_j + |Voc|)$ (better than direct $tf_{ij} \,/\, n_j$)

Multinomial Naïve Bayes

With the naïve Bayes model and having a document, we may compute

$$v_{NB} = arg\ max_{v_j \in V}\ Prob[v_j] \cdot \prod_{i=1}^{n} Prob[w_i|v_j]$$

A good idea could be to add the log(Prob[]) instead of multiplying them

why?

Try to multiply *n* times small prob…

Working with Text

Difficult to work directly with a set of text (corpus) in R. Use predefined libraries (stringR, tm)

Example: Spam detection
Classify new incoming e-mail as spam (trash) or ham (legitimate e-mail)

"Hope you are having a good week. Just checking in"
"Am also doing in cbe only. But have to pay."

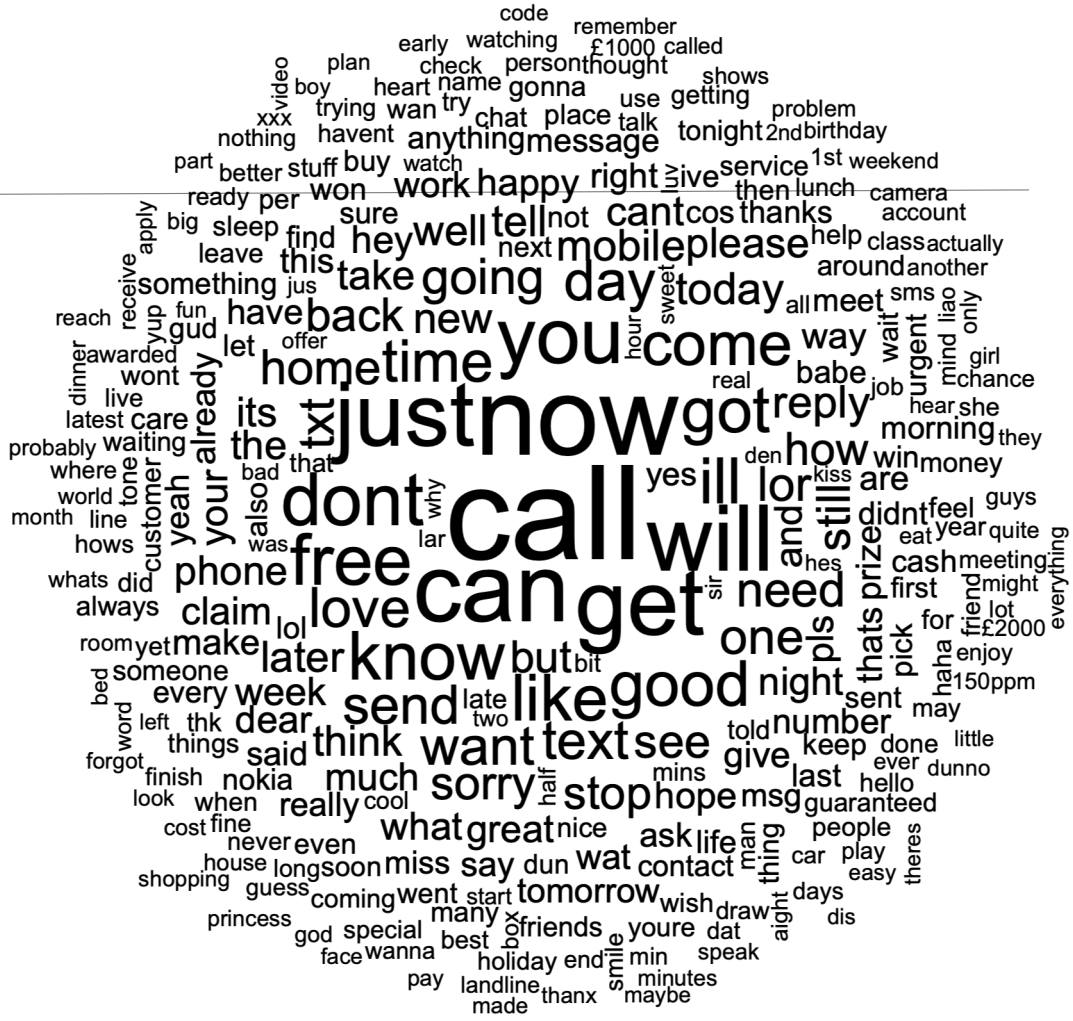
"complimentary 4 STAR Ibiza Holiday or £10,000 cash needs your URGENT collection. 09066364349 NOW from Landline not to lose out!"
"okmail: Dear Dave this is your final notice to collect your 4* Tenerife Holiday or #5000 CASH award! Call 09061743806 from landline."

Gomez J.M., Almeida, T.A., Yamakami, A. 2012. On the validity of a new SMS spam collection. *Proceedings of the 11 IEEE conference on Machine Learning and Applications*.

Feinerer, I., Hornik, K. Meier, D. 2008. Text mining infrastructures in R. *Journal of Statistical Software*. 25, 1-54.

Working with Text

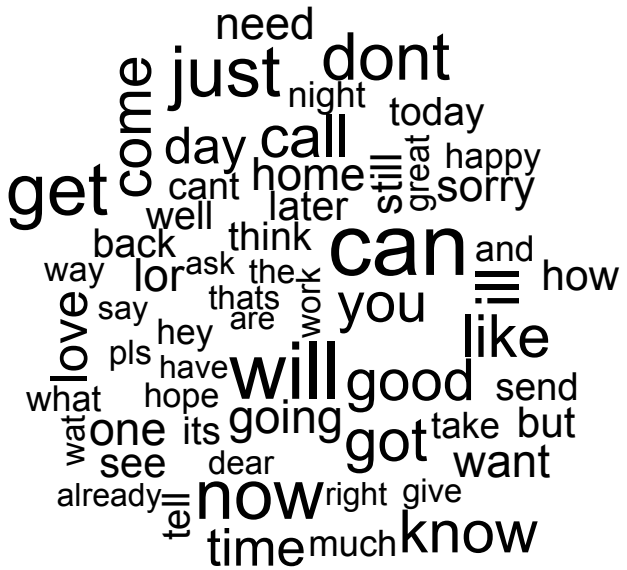
```
> wordcloud(sms_raw$text,
min.freq = 30,
random.order = FALSE)
```



Working with Text

In the spam category

In the ham category



Working with Text

```
> library(tm)
```
load the TM library (in principle already uploaded by wordcloud)

Generate a complex object (Corpus) with the text field

```
> sms_corpus <- Corpus(VectorSource(sms_raw$text))
> inspect(sms_corpus[1:2])

 <<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 2
[1] Hope you are having a good week. Just checking in K..give
back my thanks.
```

Preprocess the text:

1. Ignore the uppercase letters
2. Remove numbers
3. Remove functional words (the, in, is ,or, but, …)
4. Remove the punctuation
5. Stem the words

Working with Text

Preprocess the text:

1. Ignore the uppercase letters.
2. Remove numbers.
3. Remove functional words (the, in, is ,or, but, …).
4. Remove the punctuation.
5. Stem the words.

Working with Text

```
> corpus_clean <- tm_map(sms_corpus, tolower)
> corpus_clean <- tm_map(corpus_clean, removeNumbers)
> corpus_clean <- tm_map(corpus_clean, removeWords, stopwords())
> corpus_clean <- tm_map(corpus_clean, removePunctuation)
> corpus_clean <- tm_map(corpus_clean, stripWhitespace)
> inspect(corpus_clean[1:2])
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 2
[1] hope good week just checking  kgive back thanks
```

Not fully clear to work with strings in R. In the last line, we observe two cleaned SMSs.

Working with Text

```
> stopwords()
```

```
[1] "i"         "me"        "my"        "myself"    "we"        "our"       "ours"       "ourselves"  "you"
[10] "your"     "yours"     "yourself"  "yourselves" "he"       "him"       "his"        "himself"    "she"
[19] "her"      "hers"      "herself"   "it"        "its"       "itself"    "they"       "them"       "their"
[28] "theirs"   "themselves" "what"     "which"     "who"       "whom"      "this"       "that"       "these"
[37] "those"    "am"        "is"        "are"       "was"       "were"      "be"         "been"       "being"
[46] "have"     "has"       "had"       "having"    "do"        "does"      "did"        "doing"      "would"
[55] "should"   "could"     "ought"     "i'm"       "you're"    "he's"      "she's"      "it's"       "we're"
[64] "they're"  "i've"      "you've"    "we've"     "they've"   "i'd"       "you'd"      "he'd"       "she'd"
[73] "we'd"     "they'd"    "i'll"      "you'll"    "he'll"     "she'll"    "we'll"      "they'll"    "isn't"
[82] "aren't"   "wasn't"    "weren't"   "hasn't"    "haven't"   "hadn't"    "doesn't"    "don't"      "didn't"
[91] "won't"    "wouldn't"   "shan't"  "shouldn't" "can't"      "cannot"    "couldn't"   "mustn't"    "let's"
[100] "that's"  "who's"     "what's"    "here's"    "there's"   "when's"    "where's"    "why's"      "how's"
[109] "a"       "an"        "the"       "and"       "but"       "if"        "or"         "because"    "as"
[118] "until"   "while"     "of"        "at"        "by"        "for"       "with"       "about"      "against"
[127] "between" "into"       "through" "during"    "before"    "after"     "above"      "below"      "to"
[136] "from"    "up"        "down"      "in"        "out"       "on"        "off"        "over"       "under"
[145] "again"   "further"   "then"      "once"      "here"      "there"     "when"       "where"      "why"
[154] "how"     "all"       "any"       "both"      "each"      "few"       "more"       "most"       "other"
[163] "some"    "such"      "no"        "nor"       "not"       "only"      "own"        "same"       "so"
[172] "than"    "too"       "very"
```

Working with Text

Stemming: conflate word variants into the same root

```
> library(SnowballC)

> corpus_clean <- tm_map(corpus_clean, stemDocument)

> inspect(corpus_clean[1:2])
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 2
[1] hope good week just check kgive back thank
```

The snowball stemmer is rather aggressive, removing both inflectional and derivational suffixes. In some cases, apply a light stemmer (S-stemmer).

Harman, D. 1991. How effective is stemming? *Journal of the American Society for Information Science*, 42(1):7-15.
Porter, M.F. 1980. An Algorithm for Suffix Stripping. Program, 14(3): 130–137

Working with Text

Stemmed/clean terms →

Doc ID →

TF	call	free	mobile	good	think	…
ID1	0	0	1	0	0	…
ID2	0	1	0	1	0	…
ID3	2	3	0	0	0	…
ID4	0	0	1	0	0	…
ID5	1	0	0	1	0	…
ID6	0	0	0	0	1	…
…	…	…	…	…	…	…

Document x Term matrix

For each document and term the term frequency (TF, or absolute occurrence frequency) is given.

Sparseness is a characteristic of such matrix.

From this is can generate a Boolean Document x Term matrix

Working with Text

Prediction without Laplace smoothing

```
accuracy =  0.975
                sms_test_labels
sms_test_pred  ham spam
         ham  1201   29
         spam    6  154
```

Prediction with Laplace smoothing

```
accuracy = 0.976
                 sms_test_labels
sms_test_pred2  ham spam
          ham  1202   28
          spam    5  155
```

Conclusion

Pros and cons:

- simple, fast, and usually effective.
- produce good results even with noisy data (and missing info).
- based on a good theory (probability estimate).
- don't require a large training dataset.

- the underlying assumptions (independence, same importance) not respected.
- probability estimates are not always reliable.
- not the best method with datasets with many numerical features.

- the probability estimates in a package: not fully clear.

Conclusion

Naïve Bayes works surprisingly well (even if independence assumption is clearly violated).

Why? Because classification doesn't require accurate probability estimates *as long as maximum probability is assigned to correct class.*

"The best ones seem to be naïve Bayes and support vector machines" (Saarikoski et al., 2014), IRJ

However: adding too many redundant attributes will cause problems (e.g., similar attributes).

No feature selection: but probability estimates are closed to zero for irrelevant ones.

Note also: many numeric attributes are not normally distributed, used the other distribution of if we have no idea, see *kernel density estimators.*