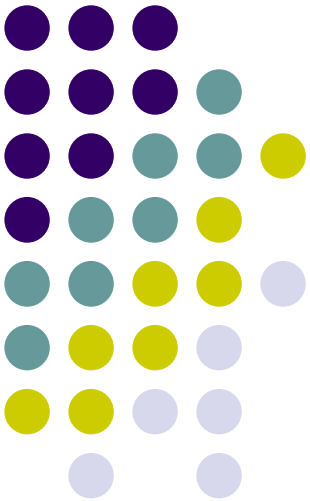
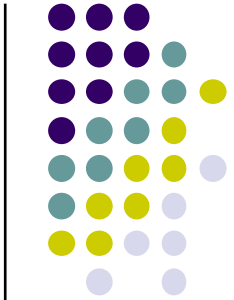
Bayesian Learning

J. Savoy
Université de Neuchâtel



I. H. Witten, E. Frank, M.A. Hall: Data Mining. Practical Machine Learning Tools and Techniques. Morgan Kaufmann.

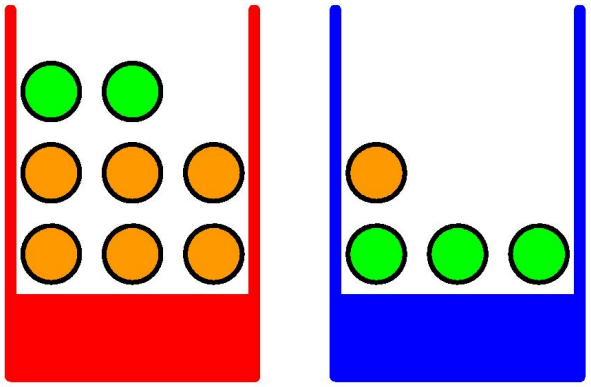
Tom M. Mitchell: Machine Learning. McGraw-Hill, 1997.



Overview

- **Bayes' Theorem**
- Naïve Bayes
- Smoothing
- Numeric Attribute

Probability Theory

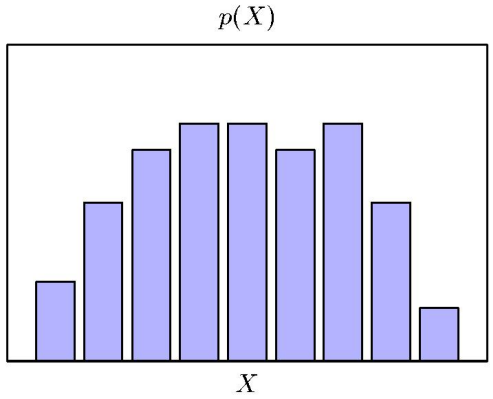


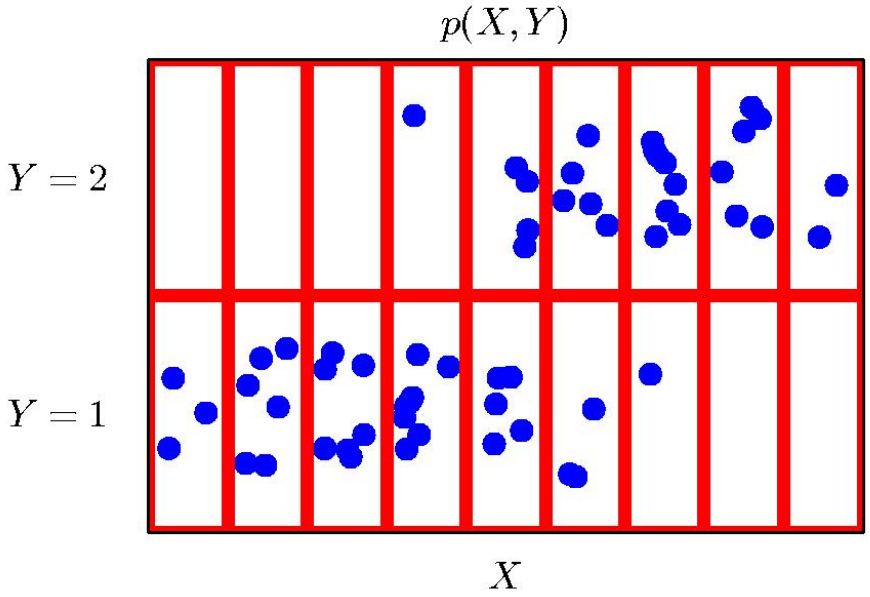
Basic probability formulae

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

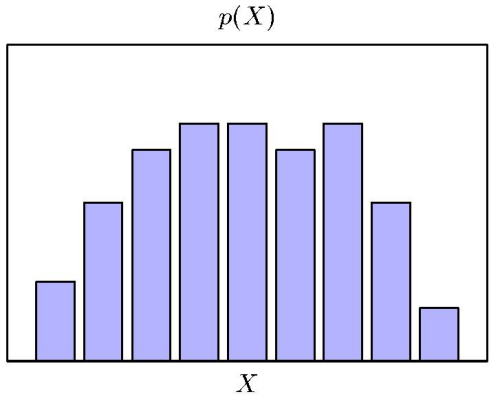


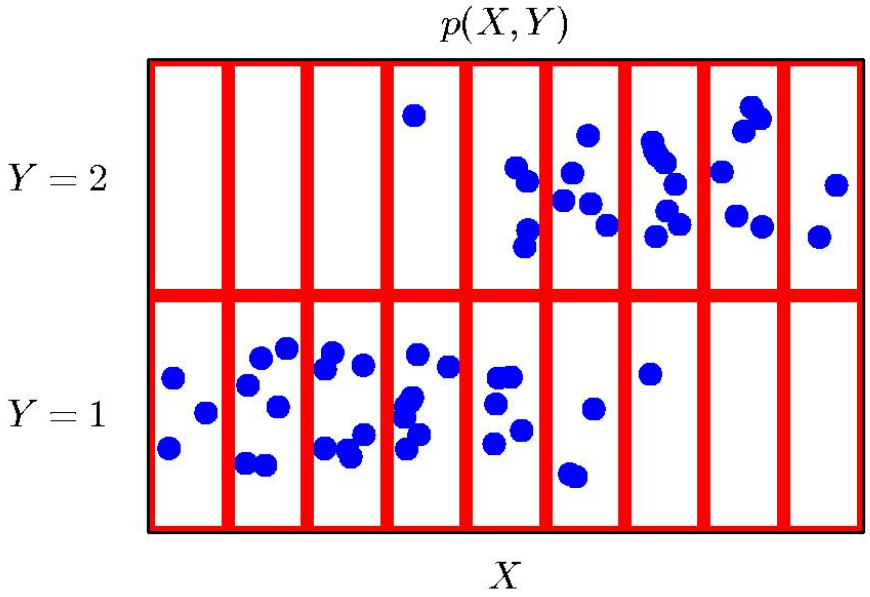


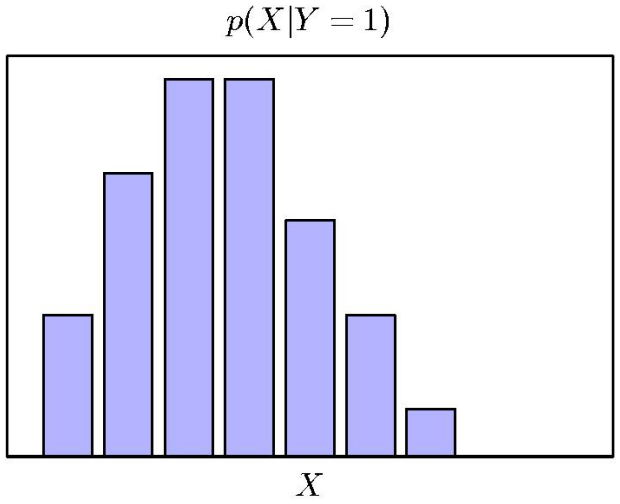
- When considering the two possible value for Y, you can guess a better estimate for X

Probability Theory

- Consider the distribution of X
- Then the conditional distribution of X given Y=1
  (values of X tend to be smaller)



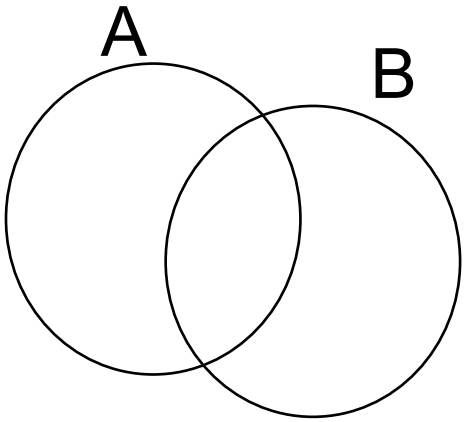


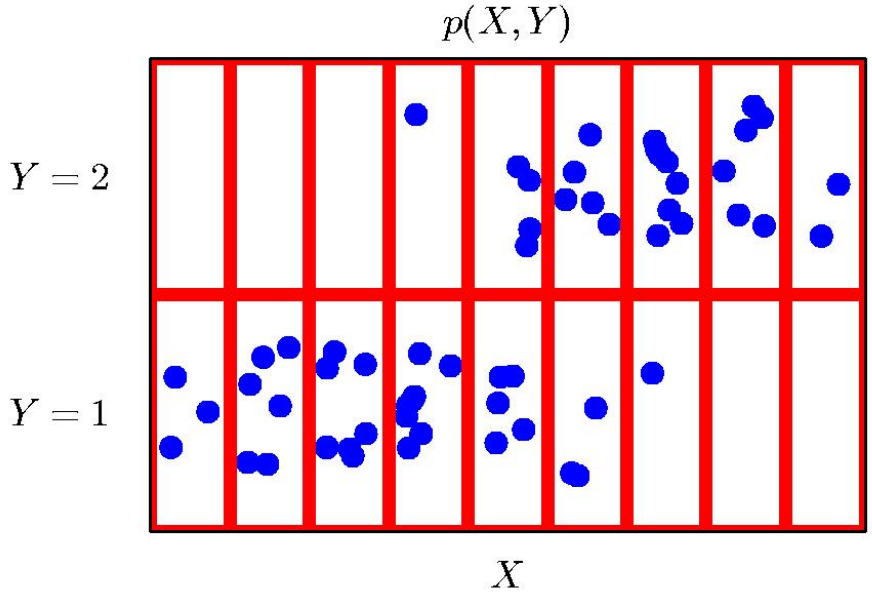


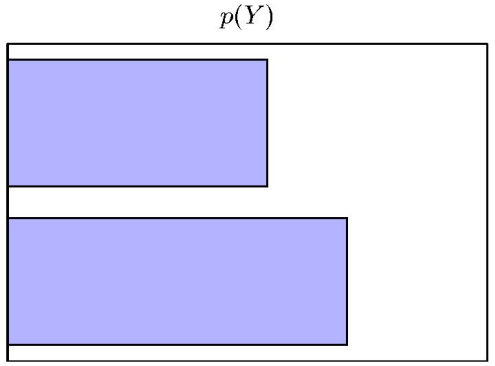
Probability Theory

- Another view is the conditional probability

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







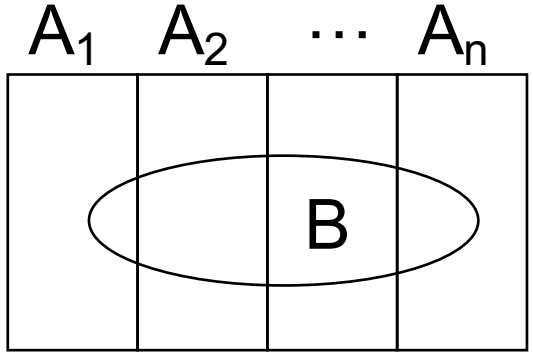
Probability Theory

- Product rule from $Prob[A|B] = \frac{Prob[A \cap B]}{Prob[B]}$

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

- Total probabilities

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



Bayes' Rule

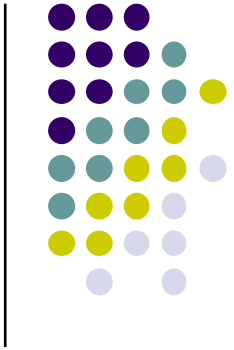
- Probability of event *H* given evidence *E:*

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

- *A priori* probability of *H* : Prob[H]
  - Probability of event *before* evidence is seen
- *A posteriori* probability of *H* : Prob[H|E]
  - Probability of event *after* evidence is seen

Thomas Bayes (1702-1761)

Example



The candy manufacturer produces large bags of candies

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

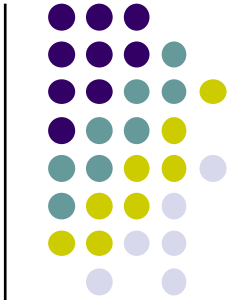
Prior: each hypothesis does not have the same probability

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

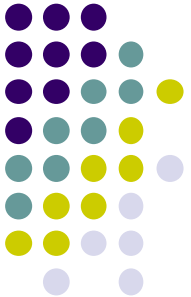
Example



- 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 $\cap$ "7”] ?

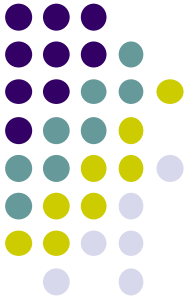
Example

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



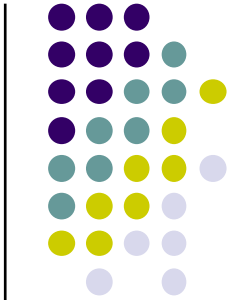
- In the bar, a person said: “I win with a 7!”
  Does this person win when rolling a pair of dice or spinning a roulette?
- Compute Prob[dice | "7”] and Prob[roulette | "7”]
- The prior: There is 6 tables, and in 2 they are playing with a roulette.
  - Prob[h_{dice}] = 4/6
  - Prob[$h_{roulette}$] = 2/6
- 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

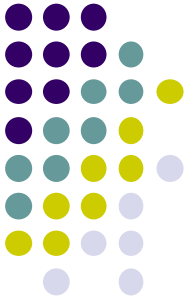


- Combining prior probabilities and the likelihood of the data (according to the hypothesis *H*)

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

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

Example

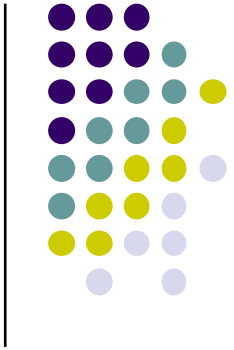


- We have the prior:
  - Prob[h_{dice}] = 4/6
  - Prob[$h_{roulette}$] = 2/6
- Evidence:
  - Prob["7" | dice] = Prob[e | h_{dice}] = 6/36
  - Prob["7" | roulette] = Prob[e | $h_{roulette}$] = 1/37
- Posteriori:

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

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

Example



The candy manufacturer produces large bags of candies

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

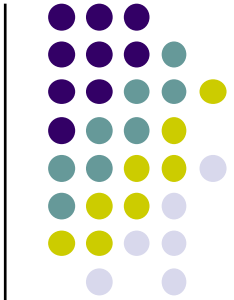
Prior: each hypothesis does not have the same probability

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

Bayes Theorem

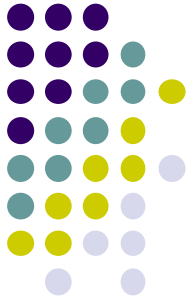


- In our candy example (having a lime candy)
- The prior are

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 $P[e|h_i]$

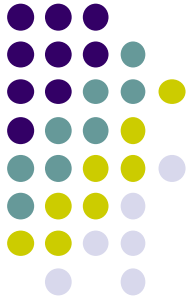
Bayes Theorem



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

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

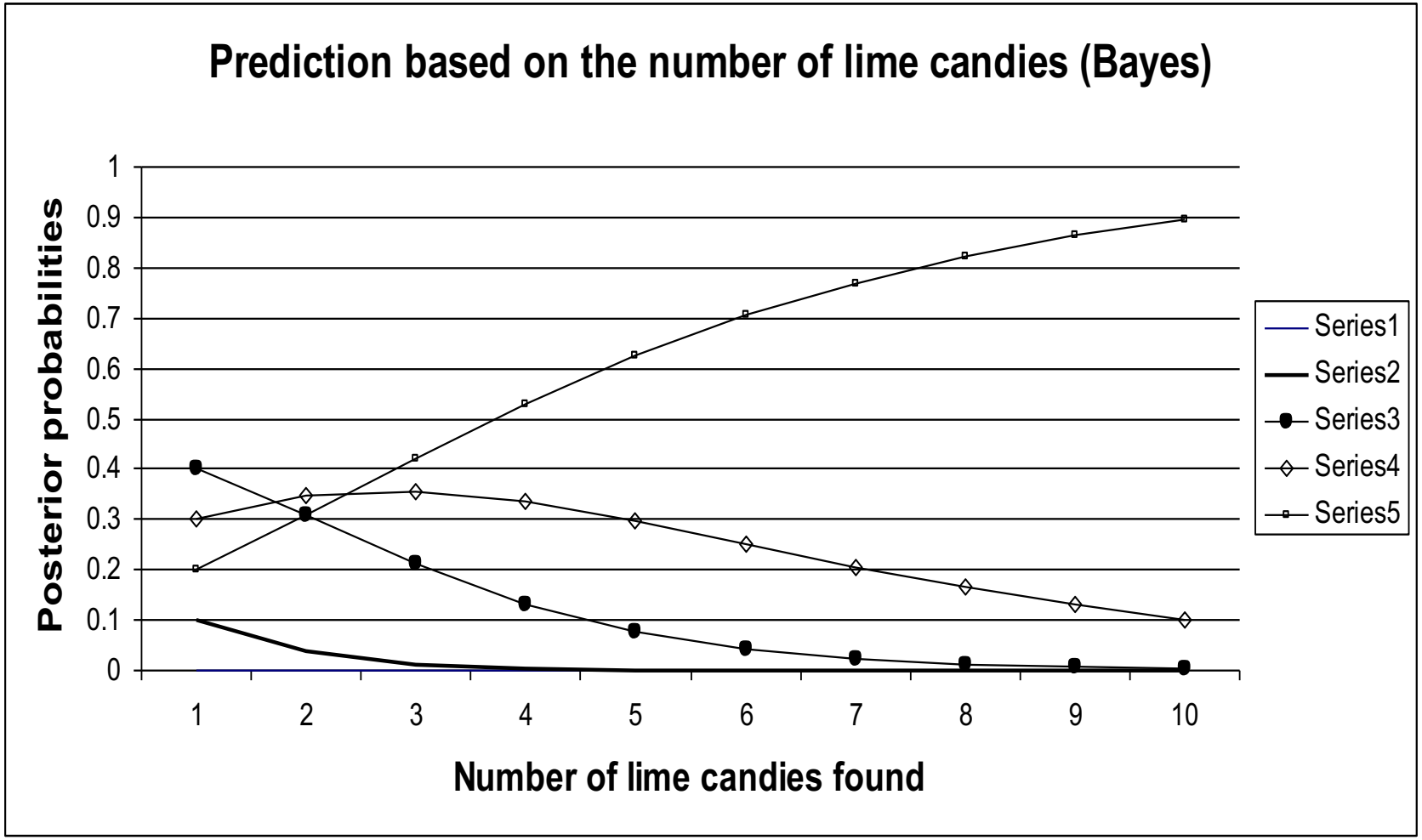
Bayes Theorem

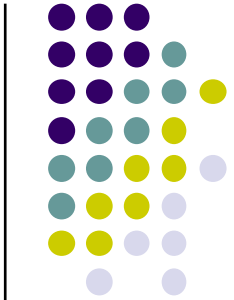


- In our candy example
  The *most probable* hypothesis (the hypothesis h_i that maximizes P[h_i|e]) is called *maximum a posteriori* or MAP and denoted h_{MAP}
- In our example, h_{MAP} depends on the evidence E
  with $e_1 \rightarrow h_3$ (the first is a lime)
  with $e_2 \rightarrow h_4$ (the first 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

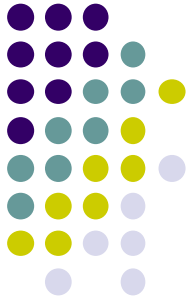




Overview

- Bayes' Theorem
- **Naïve Bayes**
- Smoothing
- Numeric Attribute

Bayes Learning



- Classification learning: what's the probability of the class given an instance?
  - Evidence *E* = new instance
  - Event *H* = class value for this new instance
- The evidence can be divided into parts (i.e., the various attributes E= e_1, e_2, …e_n)
- and we need to classify it

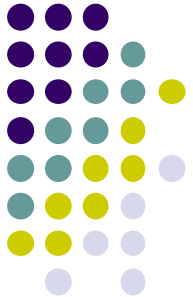
Bayes Learning

- 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



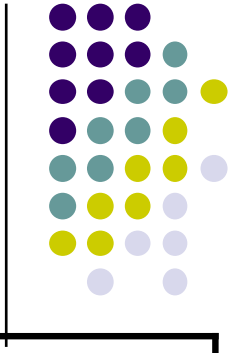
- The naïve Bayes classifier (conditionally independence)

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

and thus

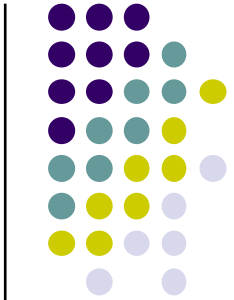
$$h_{NB} = \arg \max_{h_j \in H} \; Prob[h_j] \cdot \prod_{i=1}^{n} Prob[e_i | h_j]$$

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



Weather problem:
Decision Play: *yes* 9, *no* 5

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

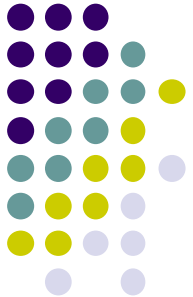
Weather problem

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

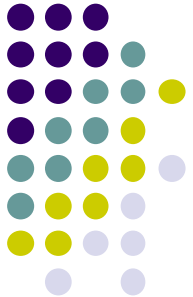
  Prob[outlook = *sunny* Λ temperature = *cool* Λ
  humidity = *high* Λ windy = *true* | *yes*]

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

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

- Thus, knowing the value of one attribute says nothing about the value of another (not always very realistic)

Probabilities

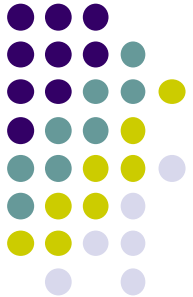


- Compute the likelihood of the possible outcomes

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

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

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

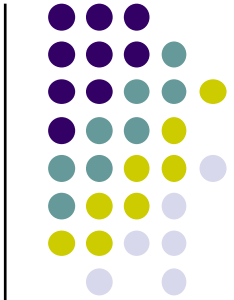
$$\text{For } no = \frac{3}{5} \cdot \frac{1}{5} \cdot \frac{4}{5} \cdot \frac{3}{5} \cdot \frac{5}{14} = 0.0206$$

- Conversion into a 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

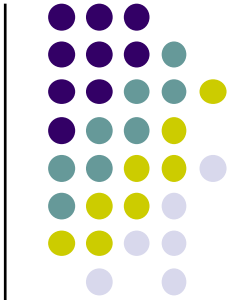
Example with a (simplified) restaurant

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:

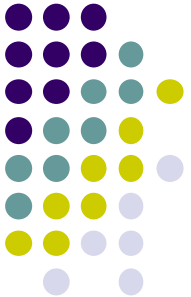
$$Prob[Patron = full|yes] = \frac{|Patron = full \cap Wait = yes|}{|Wait = yes|}$$

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

Naïve Bayes

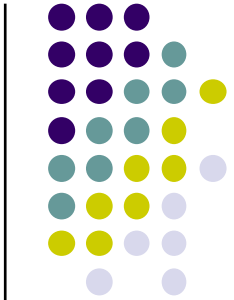


- Another example: Spam detection
- Spam

  "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 …"

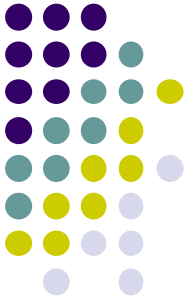
- Legitimate email (Ham)

Naïve Bayes



- 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



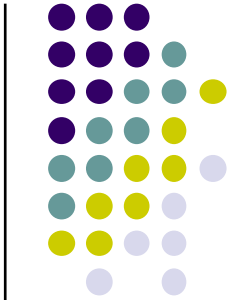
- Based on a document (web page) can you assign to it one of these categories (no overlap)
- 20 Categories

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

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 the author

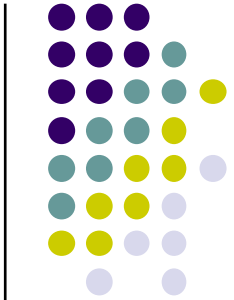
Profiling: Male/Female?



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.

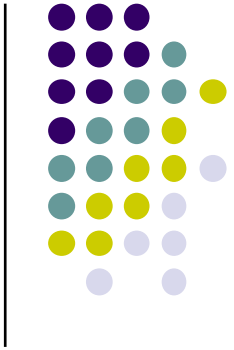
Profiling: Male/Female?



Text 2 (blog)

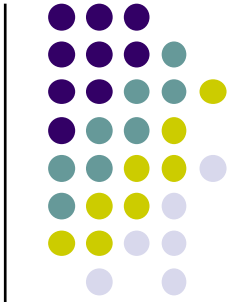
My gracious boss had agreed to let me have one week off of "work." He did finally give me my report back after eight freakin' days! Now I only have the rest of this week and then one full week after my vacation to finish this damned thing.

Text Classification



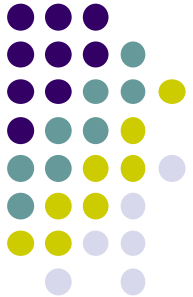
- From a news:
  “We were really concerned” said the Canadian PM “that the relationship between our countries …”
- How can we define the features needed to classify it?
- Using the words? with their position? Part-of-speech?
- Idea: limited to content-bearing tokens
- Remove very frequent tokens (the, 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)

Sentiment Analysis



Text 1

”The rains have been falling since Saturday and much of the town is under water now,” police said.

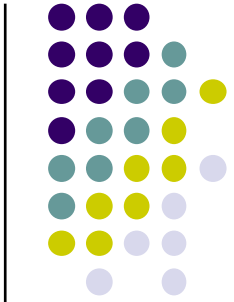
Text 2

Away from friends and family, Japan, he thought, would be the place to let him focus on putting the pieces back to together.

Text 3

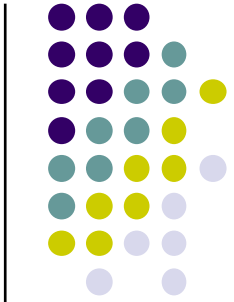
The iPhone is great, but not the microphone!

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*)
- Each article can be viewed as composed of different word (token) w_k, k = 1, 2, … |Voc| belonging to a vocabulary Voc (or size |Voc|)
- We use the notation $P[a_i = w_k|v_j]$ to indicate the probability of finding in the corresponding article, in position a_i for i = 1, 2, …, *n* the token w_k (the *k*th item extracted from Voc) knowing that this article belongs to the class v_j (in short $P[w_k \mid v_j]$)

Text Classification



The general formulation is therefore

$$v_{MAP} = arg\ max_{v_j \in V}\ Prob[v_j | w_1, w_2, \ldots, w_n]$$

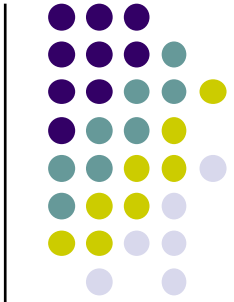
$$= arg\ max_{v_j \in V}\ \frac{Prob[w_1, w_2, \ldots, w_n | v_j] \cdot Prob[v_j]}{Prob[w_1, w_2, \ldots, w_n]}$$

$$= arg\ max_{v_j \in V}\ Prob[w_1, w_2, \ldots, w_n | v_j] \cdot Prob[v_j]$$

Applying the naïve Bayes approach

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

Text Classification



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

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

$$v_{NB} = arg\ max_{v_j \in V}\ Prob[v_j] \cdot Prob[w_1 = "concern"|v_j] \cdot Prob[w_2 = "said"|v_j] \cdots$$

but with the corresponding probability estimates…

How to estimate the underlying probabilities?

Relatively easy for Prob[v_j]

Text Classification (Learning)

1. Collect all words (& punctuation) that occur in the Corpus

   $V \leftarrow$ the set of all distinct words or tokens (selection?, stemming?)

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

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

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

   $Text_j$ = concatenation of all members of doc_j

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

   for each word w_k in Voc

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

   $P[w_k|v_j] = (tf_{kj}+1) / (n_j + |Voc|)$ (better than direct tf_{kj} / n_j, see later)

Text Classification (Classify)

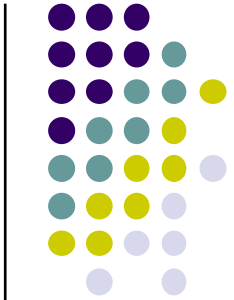
Return the estimated target value for document D

pos ← all word position in D that contain tokens in Voc
compute and return the max

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

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

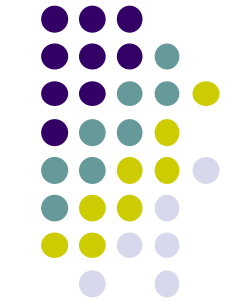
why? Try to multiply *n* times small prob…



Overview

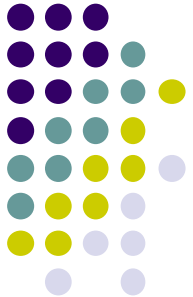
- Bayes' Theorem
- Naïve Bayes
- **Smoothing**
- Numeric Attribute

Better Probability Estimates



- What if an attribute value doesn't occur with every class value? (Using our forecast problem)
  (e.g. "Humidity = high" for class "yes")
  - 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!)

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)

Better Probability Estimates

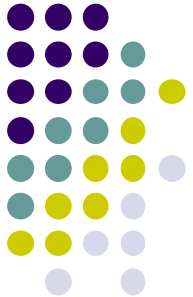
- In some cases adding a constant different from 1 might be more appropriate (below $\mu = 1$)
- Example: attribute *outlook* for class *yes*

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

Multinomial Model



Using the previous estimations for text classification, we can apply it for a small example. We have a set of documents with known categories.

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	?

Multinomial Model

The prior probabilities (two categories)

Prob[Category=1] = 3/4

Prob[Category=2] = 1/4

Distribution of the words in the two categories

Word	Category 1	Category 2
swiss	I I I I	
cheese	I I	
watch	I	I
jewelry	I	
paris		I I
|V| = 5	$n_1 = 8$	$n_2 = 3$

Multinomial Model

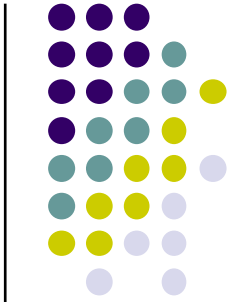
Estimating the probabilities (for two categories)

$$\text{Prob[word | Cat=1]} = (tf_{w1} + 1) / (n_1 + |V|)$$

$$\text{Prob[word | Cat=2]} = (tf_{w2} + 1) / (n_2 + |V|)$$

Word	Prob in Cat=1	Prob in Cat=2
swiss	(4+1) / (8+5)	(0+1) / (3+5)
cheese	(2+1) / (8+5)	(0+1) / (3+5)
watch	(1+1) / (8+5)	(1+1) / (3+5)
jewelry	(1+1) / (8+5)	(0+1) / (3+5)
paris	(0+1) / (8+5)	(2+1) / (3+5)

Multinomial Model



Computing the probability of each category for the query text “swiss watch swiss”

Prob[Cat=1 | query] $\propto$ ¾ · 5/13 · 2/13 · 5/13 = 0.01707

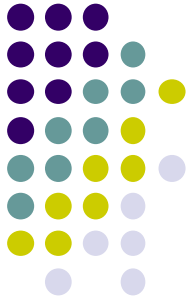
Prob[Cat=2 | query] $\propto$ ¼ · 1/8 · 2/8 · 1/8 = 0.0009766

If you want the real probabilities,

Prob[Cat=1 | query] = 0.01707 / (0.01707+0.0009766) =
= 0.946

Prob[Cat=2 | query] = 0. 0009766 / (0.01707+0.0009766) =
= 0.054

Multivariate Bernoulli Model



The estimation of the prior probabilities is the same.

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).

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	?

Bernoulli Model

The prior probabilities (two categories)

Prob[Category=1] = 3/4

Prob[Category=2] = 1/4

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
$\|C\| = 2$	$\|doc_1\| = 3$	$\|doc_2\| = 1$

Bernoulli Model

Estimating the probabilities (for two categories)

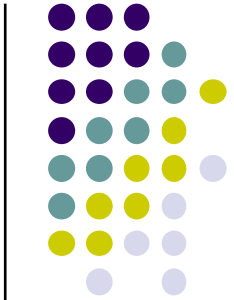
$$\text{Prob[word | Cat=1]} = (df_{w1} + 1) / (|doc_1| + |C|)$$

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

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

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)

Bernoulli Model



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

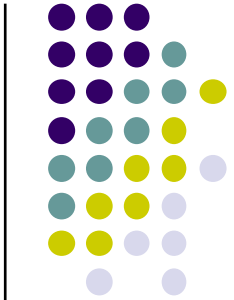
The query is then “swiss watch ¬cheese ¬jewelry ¬paris”

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

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

Prob[Cat=1 | query] = 0.06912 / (0.06912 + 0.00823) = 0.894
Prob[Cat=2 | query] = 0.00823 / (0.06912 + 0.00823) = 0.106

Missing Value

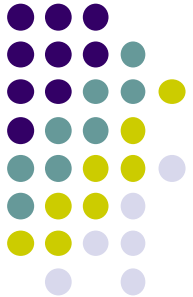


- Training: instance is not included in frequency count for attribute value-class combination
- Classification: attribute will be omitted from calculation
- Example:

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

- Ignore it?

Missing Value



- The corresponding likelihood:

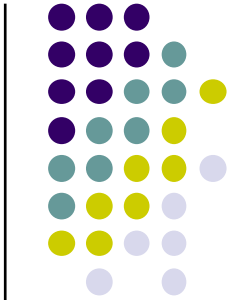
$$\text{For } yes = \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14} = 0.0238$$

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

- 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
- **Numeric Attribute**

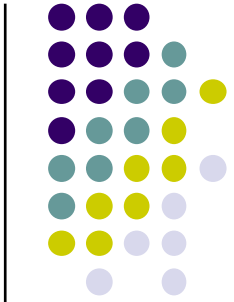
Numeric Attribute

- 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*

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

$$\text{with} \quad -\infty < x < +\infty$$

Statistics for the Weather Data

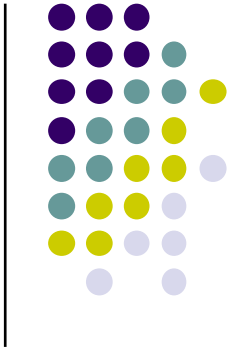


Temperature		Humidity	
yes	no	*yes*	no
64, 68, 69,70…	65, 71, 72, 80…	*65, 70, 70, 75…*	70, 85, 90, 91…
$\mu = 73$	$\mu = 75$	$\mu = 79$	$\mu = 86$
$\sigma = 6.2$	$\sigma = 7.9$	$\sigma = 10.2$	$\sigma = 9.7$

$$f(temperature = 66|yes) = \frac{1}{6.2 \cdot \sqrt{2 \cdot \pi}} e^{-\frac{(66-73)^2}{2 \cdot 6.2^2}}$$
$$= 0.0340$$

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:

$$Prob[E|yes] = \frac{2}{9}\cdot 0.0340\cdot 0.0221\cdot \frac{3}{9}\cdot \frac{9}{14} = 0.000036$$

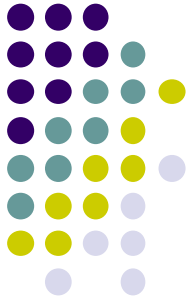
$$Prob[E|no] = \frac{3}{5}\cdot 0.0221\cdot 0.0381\cdot \frac{3}{5}\cdot \frac{5}{14} = 0.000108$$

- The resulting probabilities

$$Prob[yes] = \frac{0.000036}{(0.000036+0.000108)} = 0.25$$

$$Prob[no] = \frac{0.000108}{(0.000036+0.000108)} = 0.75$$

Numeric Attribute

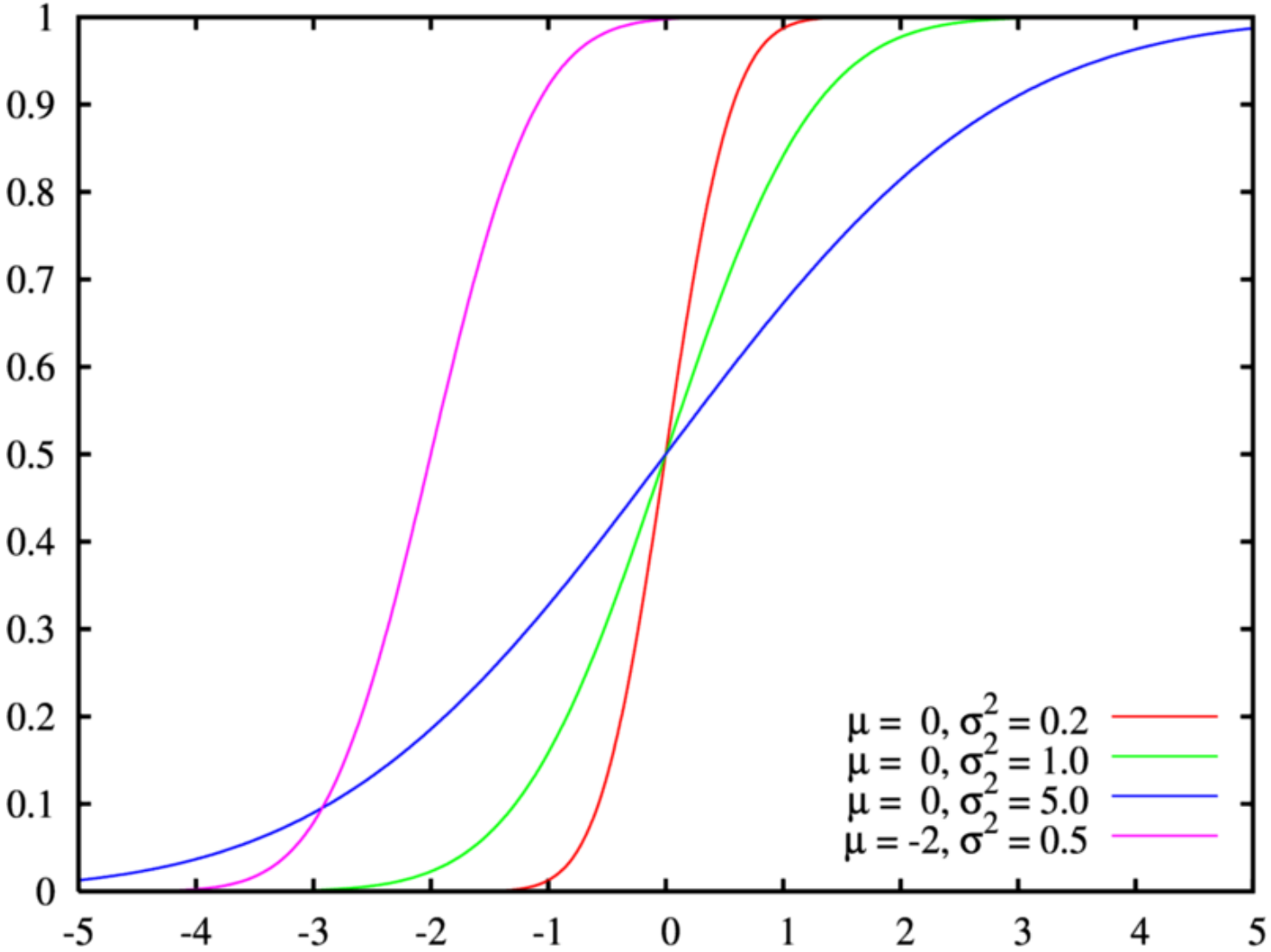


- Usually the previous technique is working well, without being fully correct.
- The density function f(x) is not directly a probability, taking a small surface ($x \pm \varepsilon$, with $\varepsilon = 0.5$)

- A correct way is to use the repartition function (F(x)) which corresponds to Prob [$x \leq X$] = F(x)
- But we have many normal distributions. We need to standardize the values to obtain the Normal distribution Z with mean = 0, standard deviation = 1

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

Numeric Attribute



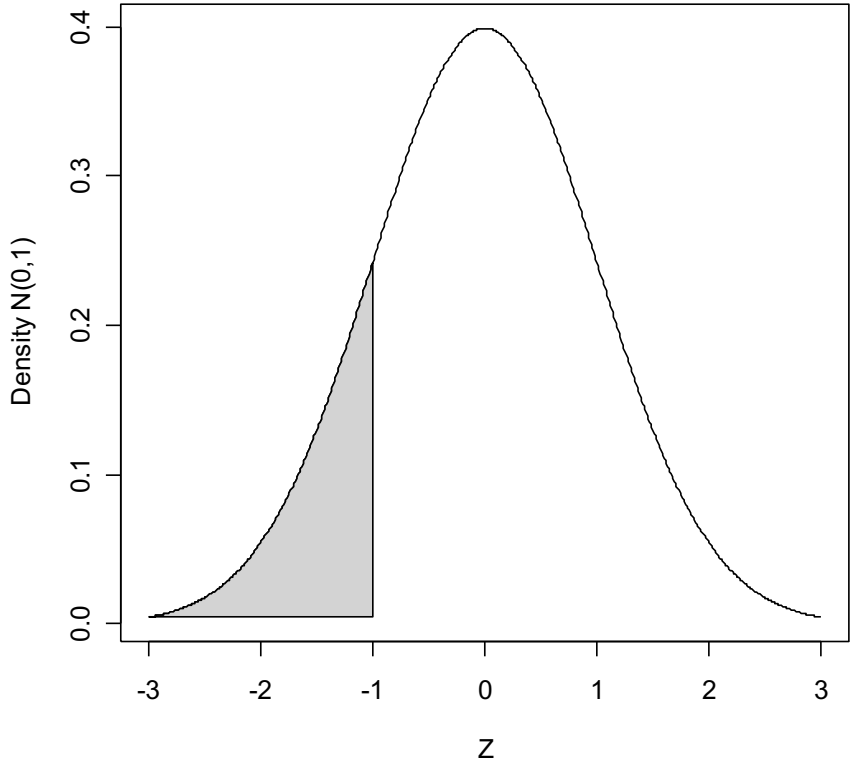
Numeric Attribute

- With Z following a N($\mu = 0$, $\sigma = 1$), we need to use only one distribution.

Example:
Prob[Z < -1] = ?

Solution:

1. Use a dedicated package / function
2. Tabulate the values of the N(0,1)

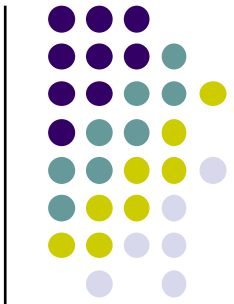


Numeric Attribute

- This is an example of a N(0,1) table

x	F(x)	x	F(x)	x	F(x)
-4	0.0	-0.4	0.34458	0.8	0.78814
-3	0.00135	-0.2	0.42075	1.0	0.84134
-2	0.02275	0	0.5	2	0.97725
-1	0.15865	0.2	0.5796	3	0.99865
-0.8	0.21186	0.4	0.65542	4	1.0

Numeric Attribute



- Example: z = 0.6
  prob = ½ (Prob[z < 0.8] – Prob[z <0.4]) = ½ (0.78814 - 0.65542) = 0.0664

- Example: x = 66 with μ = 73 et σ = 6.2

  z = (66 – 73) / 6.2 = -1.13

  d = Prob[z < -1] – Prob[z < -2] = 0.15865 - 0.02275 = 0.1359

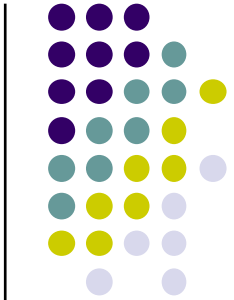
  For the width w = 1, we have 0.1359.

  If the width is smaller, w = 0.13, approximation is

  prob = 0.13 * 0.1359 = 0.0177

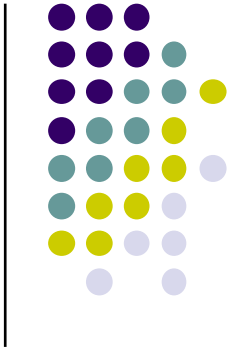
  (more precise table will return prob = 0.021)

Numeric Attribute



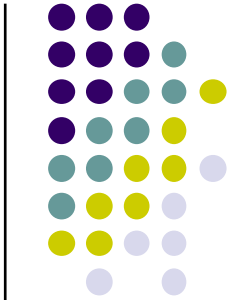
- Usual assumption: attributes have a *normal* or *Gaussian* probability distribution (given the class)
- But you can replace it by a more appropriate (if needed)

Conclusion



- "Opposite" of 1R: use all the attributes
- Two assumptions: Attributes are
  - *equally important*
  - *statistically independent* (given the class value)
    i.e., knowing the value of one attribute says nothing about the value of another (if the class is known)
- Independence assumption is never correct!
- Estimating the probabilities and some could be zero!
- Cannot handle hypotheses of composite hypotheses well
  - Suppose H_1, H_2, …, H_n are independent of each other
  - Consider a composite hypothesis $H_1 \wedge H_2$
  - How to compute the posterior probability?

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