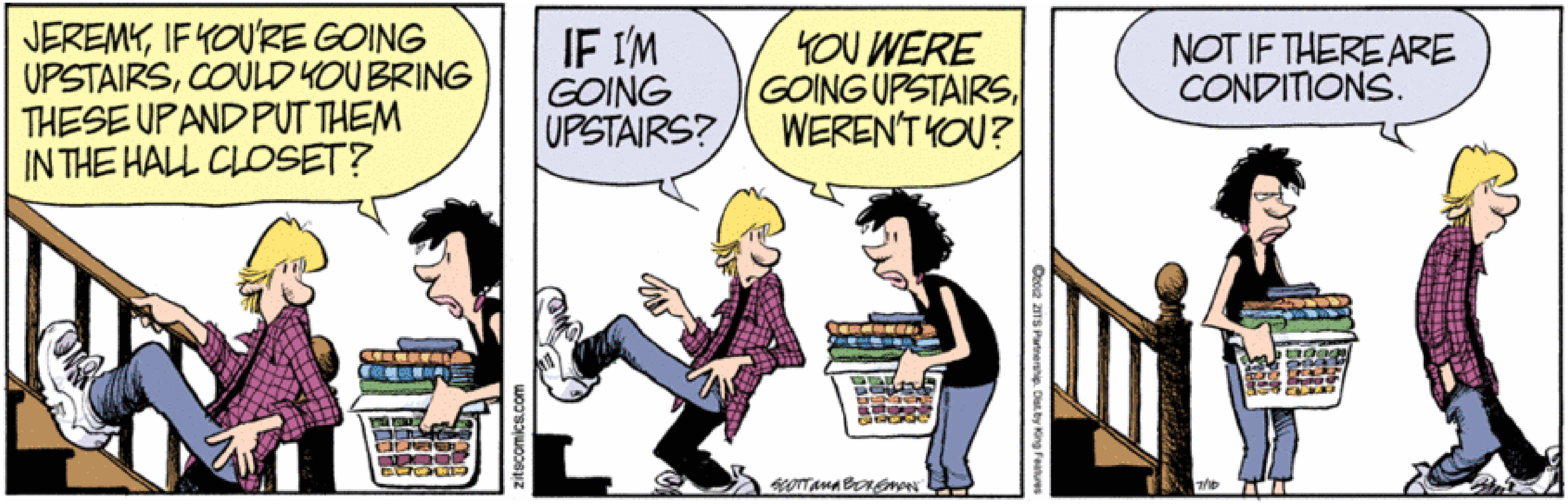

Aprendre amb Bayes simplísticament
Naïve Bayes

Coneixement, Raonament i Incertesa.

El contingut d'aquest document s'ha derivat de material provinent de
Tom Mitchell, William Cohen, Andrew Moore, Aarti Singh, Eric Xing, Carlos Guestrin.

On som?

1. Necessitem 2^m files en la joint distribution per poder fer inferencia (m és el número de variables)

   Solució? No sempre podem assegurar independència

2. No sempre tenim informació de tots els casos

   Solució? Buscar maneres alternatives a la ‘joint distribution’

D’on surten les ‘Joint Distribution’

- Idea 1: Humans Experts
- Idea 2: fets probabilistics simples + algebra

Exemple: Suposem que coneixem P(A) = 0.7

P(B|A) = 0.2　　　P(B|~A) = 0.1

P(C|A^B) = 0.1　　　P(C|A^~B) = 0.8

P(C|A^~B) = 0.8　　　P(C|~A^B) = 0.3

P(C|~A^~B) = 0.1

Llavors podem calcular la JD usant la regla de la cadena

P(A=x ^ B=y ^ C=z) =

P(C=z|A=x^ B=y) P(B=y|A=x) P(A=x)

Recordar

$$P(X \mid Y) = \frac{P(Y \mid X)P(X)}{P(Y)}$$

Recordar:

posterior

likelihood

priori

$$P(C=c \mid X) = \frac{P(X \mid C=c)P(C=c)}{P(X)}$$

normalitzador

$C = c$ mostra pertany a la classe c

$X = < x_1, x_2, \ldots, x_n >$ mostra amb n característiques

Classificador Naïve Bayes

Donada una funció objectiu f: X→C, on cada instancia x descrita pels atributs <a1, a2, …., an>. El valor més probable de f(x) és:

$$c = \arg\max_{cj \in V} P(c_j \mid a_1, a_2 \ldots. a_n)$$

$$= \arg\max_{vj \in V} \frac{P(a_1, a_2 \ldots. a_n \mid c_j) P(c_j)}{P(a_1, a_2 \ldots. a_n)}$$

$$= \arg\max_{cj \in V} P(a_1, a_2 \ldots. a_n \mid c_j) P(c_j)$$

Regla de la cadena

$$P(a_1, a_2 \ldots. a_n \mid c_j) = \frac{P(a_1, a_2 \ldots. a_n, c_j)}{P(c_j)} =$$

$$\frac{P(a_1 \mid a_2 \ldots. a_n, c_j) P(a_2 \ldots. a_n, c_j)}{P(c_j)} = \frac{P(a_1 \mid a_2 \ldots. a_n, c_j) P(a_2 \mid a_3 \ldots a_n, c_j) P(a_3 \ldots. a_n, c_j)}{P(c_j)} = \cdots$$

$$\cdots = \frac{P(a_1 \mid a_2 \ldots. a_n, c_j) P(a_2 \mid a_3 \ldots a_n, c_j) \cdots P(a_n \mid c_j) P(c_j)}{P(c_j)} =$$

$$= P(a_1 \mid a_2 \ldots. a_n, c_j) P(a_2 \mid a_3 \ldots a_n, c_j) \cdots P(a_n \mid c_j)$$

Dificil d'obtenir !!!!

Classificador Naïve Bayes

Donada una funció objectiu f: X→C, on cada instancia x descrita pels atributs <a1, a2, …., an>. El valor més probable de f(x) és:

$$c = \arg\max_{cj \in V} P(c_j \mid a_1, a_2 \ldots. a_n)$$

$$= \arg\max_{vj \in V} \frac{P(a_1, a_2 \ldots. a_n \mid c_j) P(c_j)}{P(a_1, a_2 \ldots. a_n)}$$

$$= \arg\max_{cj \in V} P(a_1, a_2 \ldots. a_n \mid c_j) P(c_j)$$

Assumció del Naïve Bayes :

$$P(a_1, a_2 \ldots. a_n \mid c_j) = \prod_i P(a_i \mid c_j)$$ **els atributs són condicionalment independents**

$$c = \arg\max_{cj \in V} \prod_i P(a_i \mid c_j) P(c_j)$$

$$= \arg\max_{cj \in V} P(c_j) \prod_i P(a_i \mid c_j)$$

Naïve Bayesian Classification

assumció Naïve : independencia cond d'atributs
$P(x_1,\ldots,x_k|C) = P(x_1|C)\cdot\ldots\cdot P(x_k|C)$

- Si el i-èssim atribut és categòric:
  $P(x_i|C)$ s'estima com la frequencia relativa de mostres que tenen valor x_i en el i-èssim atribut en la classe C (funcio de massa de probabilitat)
- Si el i-èssim atribut és continuu:
  $P(x_i|C)$ s'estima a partir d'una funció de densitat de probabilitat (Gaussiana?)

Computacionalment fàcil !!

Exemple: estimació de $P(x_i|C)$

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	Y
rain	mild	high	false	Y
rain	cool	normal	false	Y
rain	cool	normal	true	N
overcast	cool	normal	true	Y
sunny	mild	high	false	N
sunny	cool	normal	false	Y
rain	mild	normal	false	Y
sunny	mild	normal	true	Y
overcast	mild	high	true	Y
overcast	hot	normal	false	Y
rain	mild	high	true	N

P(y) = 9/14
P(n) = 5/14

outlook	
P(sunny|y = 2/9	P(sunny|n) = 3/5
P(overcast|y) = 4/9	P(overcast|n) = 0
P(rain|y) = 3/9	P(rain|n) = 2/5
Temperature	
P(hot|y) = 2/9	P(hot|n) = 2/5
P(mild|y) = 4/9	P(mild|n) = 2/5
P(cool|y) = 3/9	P(cool|n) = 1/5
Humidity	
P(high|y) = 3/9	P(high|n) = 4/5
P(normal|y) = 6/9	P(normal|n) = 2/5
Windy	
P(true|y) = 3/9	P(true|n) = 3/5
P(false|y) = 6/9	P(false|n) = 2/5

Exemple : Naïve Bayes

Predir si jugarem a tenis en un dia amb les següents condicions
<sunny, cool, high, strong> (P(C| o=sunny, t= cool, h=high w=strong))

sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	Y
rain	mild	high	false	Y
rain	cool	normal	false	Y
rain	cool	normal	true	N
overcast	cool	normal	true	Y
sunny	mild	high	false	N
sunny	cool	normal	false	Y
rain	mild	normal	false	Y
sunny	mild	normal	true	Y
overcast	mild	high	true	Y
overcast	hot	normal	false	Y
rain	mild	high	true	N

outlook	
P(sunny|y) = 2/9	**P(sunny|n) = 3/5**
P(overcast|y) = 4/9	**P(overcast|n) = 0**
P(rain|y) = 3/9	**P(rain|n) = 2/5**
Temperature	
P(hot|y) = 2/9	**P(hot|n) = 2/5**
P(mild|y) = 4/9	**P(mild|n) = 2/5**
P(cool|p) = 3/9	**P(cool|n) = 1/5**
Humidity	
P(high|y) = 3/9	**P(high|n) = 4/5**
P(normal|y) = 6/9	**P(normal|n) = 2/5**
Windy	
P(true|y) = 3/9	**P(true|n) = 3/5**
P(false|y) = 6/9	**P(false|n) = 2/5**

P(y) = 9/14
P(n) = 5/14

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

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

Exemple : Naïve Bayes

The weather data, with counts and probabilities

outlook			temperature			humidity			windy			play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

Un nou dia

outlook	temperature	humidity	windy	play
sunny	cool	high	true	?

Exemple : Naïve Bayes continuu

The numeric weather data with summary statistics

outlook			temperature		humidity		windy			play	
	yes	no	yes	no	yes	no		yes	no	yes	no
sunny	2	3	83	85	86	85	false	6	2	9	5
overcast	4	0	70	80	96	90	true	3	3		
rainy	3	2	68	65	80	70					
			64	72	65	95					
			69	71	70	91					
			75		80						
			75		70						
			72		90						
			81		75						

Ho aproximem amb una funció de densitat de probabilitat.
Per exemple: gaussiana

Exemple : Naïve Bayes continuu

- Si $x_1, x_2, \ldots, x_n$ son els valors d'un atribut numèric en el conjunt d'aprenentatge, llavors la distribució normal que els fita s'aproxima per:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \mu\right)^2$$

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

Exemple : Naïve Bayes continuu

The numeric weather data with summary statistics

outlook			temperature			humidity			windy			play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3		83	85		86	85	false	6	2	9	5
overcast	4	0		70	80		96	90	true	3	3		
rainy	3	2		68	65		80	70					
				64	72		65	95					
				69	71		70	91					
				75			80						
				75			70						
				72			90						
				81			75						
sunny	2/9	3/5	mean	73	74.6	mean	79.1	86.2	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	std dev	6.2	7.9	std dev	10.2	9.7	true	3/9	3/5		
rainy	3/9	2/5											

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

- Per exemple,

$$f\left(\text{temperature} = 66 \mid \text{Yes}\right) = \frac{1}{6.2\sqrt{2\pi}} e^{-\frac{(66-73)^2}{2(6.2)^2}} = 0.0340$$

- Likelihood de Yes = $\frac{2}{9} \times 0.0340 \times 0.0221 \times \frac{3}{9} \times \frac{9}{14} = 0.000036$

- Likelihood de No = $\frac{3}{5} \times 0.0291 \times 0.038 \times \frac{3}{5} \times \frac{5}{14} = 0.000136$

Algorisme Naïve Bayes

Naïve_Bayes_Learn (examples)
  Per a cada valor objectiu Cj
    estimar P(Cj)
  per a cada valor ai del atribut a
    estimar P(ai | Cj)

Classificar_nova_instancia (x) seguint

$$c = \arg\max_{C_j \in C} P(C_j) \prod_{a_i \in x} P(a_i \mid C_j)$$

Si tenim 10000 variables el producte provocarà imprecisió numèrica.
Solució: usar logaritmes

$$c = \arg\max_{C_j \in C} \left(\log P(C_j) + \sum_{a_i \in x} \log P(a_i \mid C_j) \right)$$

$P(x)$	$\log P(x)$
[[0.08683217],	[[-2.44377805],
[0.02000183],	[-3.9119316],
[0.01367926],	[-4.29187442],
[0.08409573],	[-2.47579952],
[0.0389006],	[-3.24674553],
[0.04847426],	[-3.02672235],
[0.06446187],	[-2.74168135],
[0.05903723],	[-2.82958703],
[0.05259446],	[-2.9451445],
[0.04257019]]	[-3.15660115]]

$\prod_x P(x)$	$\sum_x \log P(x)$
3.2101760910592 51e-14	-31.069865509251503

a=np.random.random((100,1))
np.log(a).sum() → -89.02957210783187
np.prod(a) → 2.1624601219474498e-39

Naïve Bayes: $P(a_i|C_j)$

Si estem de mala sort, la nostra estimació de MLE per a $P(a_i \mid C_j)$ pot ser 0 (e.g., a_{373}= nascut el 30/10/2001)

- Per a que preocuparse per un paràmetre quan en tenim molts?

$$c = \arg\max_{C_j \in C} P(C_j) \prod_{a_i \in x} P(a_i \mid C_j) = 0$$

- Com ho podem evitar?

Naïve Bayes: $P(x_i|C_j)$

Maximum likelihood estimates:

$$P(C = C_k) = \frac{\#D\{C = C_k\}}{|D|}$$

$$P\left(X_i = x_{ij} | C = C_k\right) = \frac{\#D\{X_i = x_{ij} \wedge C = C_k\}}{\#D\{C = C_k\}}$$

MAP estimates (Dirichlet priors):

$$\hat{P}(C = C_k) = \frac{\#D\{C = C_k\} + l}{|D| + lR}$$

Única diferencia: exemples "imaginaris"

$$\hat{P}\left(X_i = x_{ij} | C = C_k\right) = \frac{\#D\{X_i = x_{ij} \wedge C = C_k\} + l}{\#D\{C = C_k\} + lM}$$

$l=1$ s'anomena Laplace Smoothing

R és el número de diferents valors de C, i M el número de diferents valors de X_i

La hipòtesi d'independencia cond…

- … fa possible la computatció
- … dona un classificador òptim si es compleix
- … però rarament es satisfà a la pràctica, ja que els atributs (variables) sovint estan correlacionats.

Com evitar aquesta limitació:

- Bayesian networks, que combinen el raonament bayesià amb relacions de causalitat entre atributs
- Decision trees, que raonen sobre un atribut a cada pas, considerant els atributs més 'importants' primer.

Naive Bayes no és tant Naive

- Aprenentatge i test molt rapid (basicament contar dades)
- Requeriments de memòria baixos
- Molt bo en dominis amb moltes caracteristiques igualment importants
- Més robust a dades irrellevants que molts altres mètodes
- Més robust a ‘concept drift’ (canvi de definició de la classe sobre el temps)
- Naive Bayes va guanyar el primer i segon premi de KDD-CUP 97 entre 16 sistemes

  Goal: Financial services industry direct mail response prediction: Predict if the recipient of mail will actually respond to the advertisement – 750,000 records.

Exemple: aprendre a classificar documents

- Classificar quins emails son spam
- Classifcar quins emails són convocatories de reunions
- Classifcar quines pàgines web són d'estudiants

Com hem de representar els documents de text per a Naïve Bayes?

Article from rec.sport.hockey

```
Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.e
From: xxx@yyy.zzz.edu (John Doe)
Subject: Re: This year's biggest and worst (opinic
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 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

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

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

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

Baseline: Bag of Words Approach

aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
…	
Zaire	0

Twenty NewsGroups

Given 1000 training documents from each group
Learn to classify new documents according to which newsgroup it came from

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

Naive Bayes: 89% classification accuracy

$\text{LEARN\_NAIVE\_BAYES\_TEXT}(Examples, V)$

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 $P(v_j)$ and $P(w_k|v_j)$ probability terms*

- For each target value v_j in V do
  - $docs_j \leftarrow$ subset of $Examples$ for which the target value is v_j
  - $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
  - $Text_j \leftarrow$ 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 $Text_j$
    * $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$

CLASSIFY_NAIVE_BAYES_TEXT(Doc)

- $positions \leftarrow$ all word positions in Doc that contain tokens found in $Vocabulary$
- Return v_{NB}, where

$$v_{NB} = \operatorname*{argmax}_{v_j \in V} P(v_j) \prod_{i \in positions} P(a_i | v_j)$$