Classification: Naïve Bayes' Classifier

- References

- http://www.cs.cmu.edu/~ninamf/courses/601sp15/slides/04_NBayes-1-26-2015.pptx.pdf
- https://www3.cs.stonybrook.edu/~cse634/19Bayes2.pdf
- http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

Let's learn classifiers by learning P(Y|X)

Consider Y=Wealth, X=<Gender, HoursWorked>

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

Gender	HrsWorked	P(rich | G,HW)	P(poor | G,HW)
F	<40.5	.09	.91
F	>40.5	.21	.79
M	<40.5	.23	.77
M	>40.5	.38	.62

How many parameters must we estimate?

Suppose $X = \langle X_1, \ldots X_n \rangle$
where X_i and Y are boolean RV's

Gender	HrsWorked	P(rich | G,HW)	P(poor | G,HW)
F	<40.5	.09	.91
F	>40.5	.21	.79
M	<40.5	.23	.77
M	>40.5	.38	.62

To estimate $P(Y| X_1, X_2, \ldots X_n)$

If we have 30 boolean X_i's: $P(Y \mid X_1, X_2, \ldots X_{30})$

How many parameters must we estimate?

Suppose $X = \langle X_1, \ldots X_n \rangle$
where X_i and Y are boolean RV's

Gender	HrsWorked	P(rich | G,HW)	P(poor | G,HW)
F	<40.5	.09	.91
F	>40.5	.21	.79
M	<40.5	.23	.77
M	>40.5	.38	.62

To estimate $P(Y|X_1, X_2, \ldots X_n)$

2^n

If we have 30 X_i's instead of 2?

$2^{30} \sim 1$ Billion

Bayes Rule

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

Which is shorthand for:

$$(\forall i,j) P(Y = y_i | X = x_j) = \frac{P(X = x_j | Y = y_i) P(Y = y_i)}{P(X = x_j)}$$

Equivalently:

$$(\forall i,j) P(Y = y_i | X = x_j) = \frac{P(X = x_j | Y = y_i) P(Y = y_i)}{\sum_k P(X = x_j | Y = y_k) P(Y = y_k)}$$

Can we reduce params using Bayes Rule?

Suppose $X = \langle X_1, \ldots X_n \rangle$
where X_i and Y are boolean RV's

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

How many parameters to define $P(X_1, \ldots X_n \mid Y)$?

How many parameters to define P(Y)?

Can we reduce params using Bayes Rule?

Suppose $X = \langle X_1, \ldots X_n \rangle$
where X_i and Y are boolean RV's

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

how many params for $P(X_1 \ldots X_n|Y)$ $(2^n - 1) \cdot 2$

how many for $P(Y) = 1$

Naïve Bayes

Naïve Bayes assumes

$$P(X_1 \ldots X_n|Y) = \prod_i P(X_i|Y)$$

i.e., that X_i and X_j are conditionally independent given Y, for all $i \neq j$

Conditional Independence

Definition: X is conditionally independent of Y given Z, if the probability distribution governing X is independent of the value of Y, given the value of Z

$$(\forall i, j, k) P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

Which we often write

$$P(X|Y, Z) = P(X|Z)$$

E.g.,

$$P(Thunder|Rain, Lightning) = P(Thunder|Lightning)$$

Naïve Bayes uses assumption that the X_i are conditionally independent, given Y. E.g., $P(X_1|X_2, Y) = P(X_1|Y)$

Given this assumption, then:

$$P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y)$$
$$= P(X_1|Y)P(X_2|Y)$$

in general: $P(X_1 \ldots X_n|Y) = \prod_i P(X_i|Y)$

How many parameters to describe $P(X_1 \ldots X_n|Y)$? $P(Y)$?

- Without conditional indep assumption?
- With conditional indep assumption?

Naïve Bayes uses assumption that the X_i are conditionally independent, given Y

Given this assumption, then:

$$P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y)$$
$$= P(X_1|Y)P(X_2|Y)$$

Chain rule
Cond. Indep.

in general: $P(X_1 \ldots X_n|Y) = \prod_i P(X_i|Y)$

How many parameters to describe $P(X_1 \ldots X_n|Y)$? $P(Y)$?

- Without conditional indep assumption? $2(2^n - 1) + 1$
- With conditional indep assumption? $2n + 1$

Naïve Bayes in a Nutshell

Bayes rule:

$$P(Y = y_k | X_1 \dots X_n) = \frac{P(Y = y_k) P(X_1 \dots X_n | Y = y_k)}{\sum_j P(Y = y_j) P(X_1 \dots X_n | Y = y_j)}$$

Assuming conditional independence among X_i's:

$$P(Y = y_k | X_1 \dots X_n) = \frac{P(Y = y_k) \prod_i P(X_i | Y = y_k)}{\sum_j P(Y = y_j) \prod_i P(X_i | Y = y_j)}$$

So, to pick most probable Y for $X^{new} = < X_1, \dots, X_n >$

$$Y^{new} \leftarrow \arg\max_{y_k} \; P(Y = y_k) \prod_i P(X_i^{new} | Y = y_k)$$

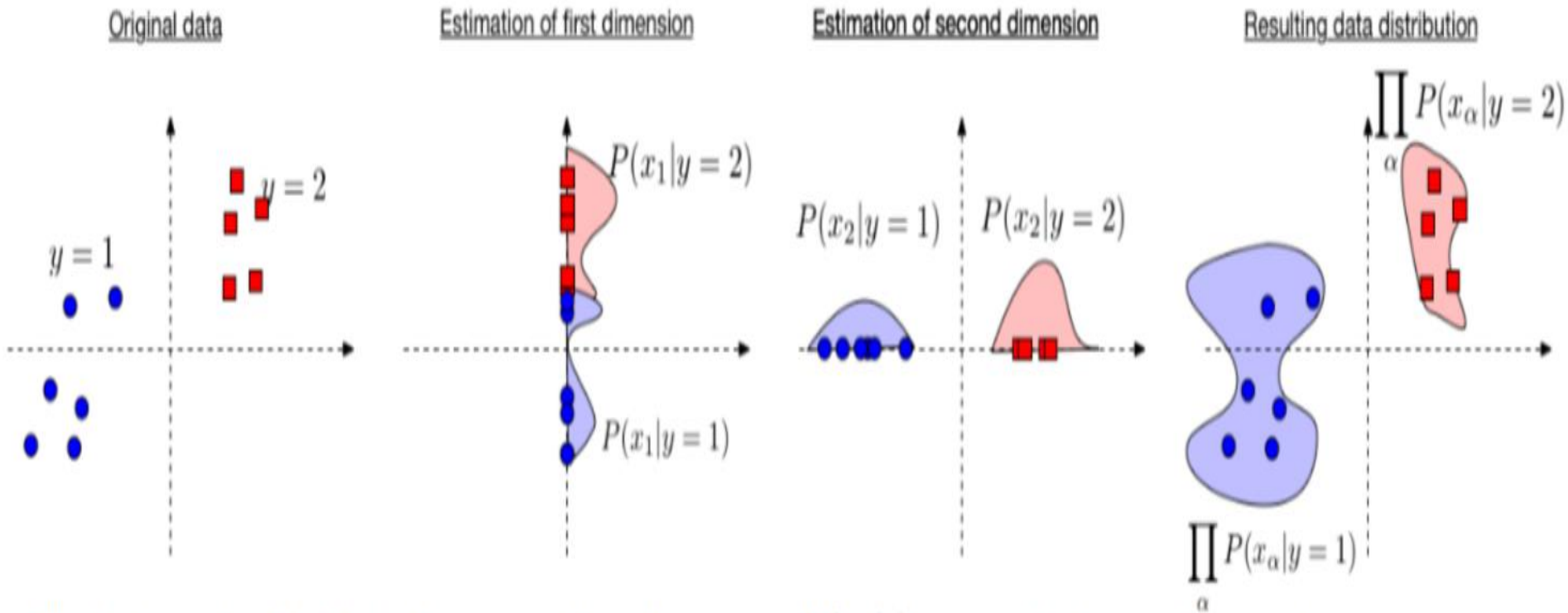

Illustration behind the Naive Bayes algorithm. We estimate $P(x_\alpha|y)$ independently in each dimension (middle two images) and then obtain an estimate of the full data distribution by assuming conditional independence $P(\mathbf{x}|y) = \prod_\alpha P(x_\alpha|y)$ (very right image).

https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote05.html

Naïve Bayes Classifier

- A naïve Bayes (NB) classifier is a simple probabilistic classifier based on: (a) Bayes theorem, (b) strong (naïve) independence assumptions, and (c) independent feature models.
- It is also an important mining classifier for pattern classification and applied in many real world classification problems because of its high classification performance.
- A NB classifier can easily handle missing attribute values by simply omitting the corresponding probabilities for those attributes when calculating the likelihood of membership for each class.
- The NB classifier also requires the class conditional independence, i.e. the effect of an attribute on a given class is independent of those of other attributes.

Dataset

- Given a training dataset, $D = \{X_1, X_2, \cdots, X_n\}$, each data record is represented as, $X_i = \{x_1, x_2, \cdots, x_n\}$.
- D contains the following attributes $\{A_1, A_2, \cdots, A_n\}$ and each attribute A_i contains the following attribute values $\{A_{i1}, A_{i2}, \cdots, A_{ih}\}$.
- The attribute values can be discrete or continuous.
- D also contains a set of classes $C = \{C_1, C_2, \cdots, C_m\}$. Each training instance, $X \in D$, has a particular class label C_i.
- For a test instance, X, the classifier will predict that X belongs to the class with the highest posterior probability, conditioned on X.

NB classifier

The NB classifier predicts that the instance X belongs to the class C_i, if and only if $P(C_i|X) > P(C_j|X)$ for $1 \leq j \leq m, j \neq i$. The class C_i for which $P(C_i|X)$ is maximized is called the Maximum Posteriori Hypothesis.

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \tag{9}$$

In Bayes theorem shown in Eq. 9, as $P(X)$ is a constant for all classes, only $P(X|C_i)P(C_i)$ needs to be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, $P(C_1) = P(C_2) = \cdots = P(C_m)$, and therefore maximize $P(X|C_i)$. Otherwise, maximize $P(X|C_i)P(C_i)$. The class prior probabilities are calculated by $P(C_i) = |C_{i,D}|/|D|$, where $|C_{i,D}|$ is the number of training instances belonging to the class C_i in D.

NB classifier (con.)

To compute $P(X|C_i)$ in a dataset with many attributes is extremely computationally expensive. Thus, the naïve assumption of class conditional independence is made in order to reduce computation in evaluating $P(X|C_i)$. The attributes are conditionally independent of one another, given the class label of the instance. Thus, Eq. 10 and Eq. 11 are used to produce $P(X|C_i)$.

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \tag{10}$$

$$P(X|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i) \tag{11}$$

In Eq. 10, x_k refers to the value of attribute A_k for instance X. Therefore, these probabilities $P(x_1|C_i), P(x_2|C_i), \cdots, P(x_n|C_i)$ can be easily estimated from the training instances.

NB classifier (con.)

Moreover, the attributes in training datasets can be categorical or continuous-valued. If the attribute value, A_k, is categorical, then $P(x_k|C_i)$ is the number of instances in the class $C_i \in D$ with the value x_k for A_k, divided by $|C_{i,D}|$, i.e. the number of instances belonging to the class $C_i \in D$. If A_k is a continuous-valued attribute, then A_k is typically assumed to have a Gaussian distribution with a mean μ and standard deviation σ, defined respectively by the following two equations:

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) \tag{12}$$

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-p)^2}{2\sigma^2}} \tag{13}$$

In Eq. 12, μ_{C_i} is the mean and σ_{C_i} is the standard deviation of the values of the attribute A_k for all training instances in the class C_i. Now we can bring these two quantities to Eq. 13, together with x_k, in order to estimate $P(x_k|C_i)$.

Algorithm 1 Naïve Bayes classifier

Input: $D = \{x_1, x_2, \cdots, x_n\}$ // Training data.
Output: A naïve Bayes Model.
Method:

1: **for** each class, $C_i \in D$, **do**
2: Find the prior probabilities, $P(C_i)$.
3: **end for**
4: **for** each attribute, $A_i \in D$, **do**
5: **for** each attribute value, $A_{ij} \in A_i$, **do**
6: Find the class conditional probabilities, $P(A_{ij}|C_i)$.
7: **end for**
8: **end for**
9: **for** each instance, $x_i \in D$, **do**
10: Find the posterior probability, $P(C_i|x_i)$;
11: **end for**

NB classifier (con.)

To predict the class label of instance X, $P(X|C_i)P(C_i)$ is evaluated for each class $C_i \in D$. The NB classifier predicts that the class label of instance X is the class C_i, if and only if

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \tag{14}$$

In Eq. 14, $1 \leq j \leq m$ and $j \neq i$. That is the predicted class label is the class C_i for which $P(X|C_i)P(C_i)$ is the maximum probability. Algorithm 1 outlines the naïve Bayes classifier algorithm.

Laplace Correction

- A zero probability cancels the effects of all the other (posteriori) probabilities (on C_i) involved in the product.
- We can assume that our training database, D, is so large that adding one to each count that we need would only make a negligible difference in the estimated probability value, yet would conveniently avoid the case of probability values of zero.
- This technique estimation is known as the **Laplacian correction** or **Laplace estimator**, named after Pierre Laplace, a French mathematician who lived from 1749 to 1827.

unsmoothed

$$p(X_1 = T \mid Y = \text{spam}) = \frac{\text{freq. of T in spam data}}{\text{total \# of spam instances}}$$

smoothed

$$p(X_1 = T \mid Y = \text{spam}) = \frac{\text{freq. of T in spam} + 1}{\text{total \# of spam instances} + v}$$

Classification Example

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- A married person with income 120K did not refund the loan previously
- *Can we trust him*?

Naïve Bayes Classifier

- Assume independence among attributes A_i when class is given:
  - $P(A_1, A_2, \ldots, A_n \mid C) = P(A_1 \mid C_j)\, P(A_2 \mid C_j) \ldots P(A_n \mid C_j)$
  - Can estimate $P(A_i \mid C_j)$ for all A_i and C_j.
  - *the new pattern* is classified to C_j if $P(C_j)\, \Pi\, P(A_i \mid C_j)$ is maximal.

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Class: $P(C) = N_c/N$
  - e.g., $P(No) = 7/10$, $P(Yes) = 3/10$

- For discrete attributes:

$$P(A_i \mid C_k) = |A_{ik}| / N_c$$

  - where $|A_{ik}|$ is number of instances having attribute A_i and belongs to class C_k
  - Examples:

P(Status=Married|No) = 4/7
P(Refund=Yes|Yes)=0

How to Estimate Probabilities from Data?

- For continuous attributes:
  - Discretize the range into bins
    - one ordinal attribute per bin
    - violates independence assumption
  - Two-way split: $(A < v)$ or $(A > v)$
    - choose only one of the two splits as new attribute
  - Probability density estimation:
    - Assume attribute follows a normal distribution
    - Use data to estimate parameters of distribution (e.g., mean and standard deviation)
    - Once probability distribution is known, can use it to estimate the conditional probability $P(A_i|c)$

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Normal distribution:

$$P(A_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

  - One for each (A_i,c_i) pair

- For (Income, Class=No):
  - If Class=No
    - sample mean = 110K
    - sample variance = 2975

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Normal distribution:

$$P(A_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

  - One for each (A_i,c_i) pair

- For (Income, Class=No):
  - If Class=No
    - sample mean = 110K (770/7) = 110
    - sample variance = 2975

$$P(Income = 120 \mid No) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

Example of Naïve Bayes Classifier

Given a Test Record: $X = (\text{Refund} = \text{No}, \text{Married}, \text{Income} = 120\text{K})$

naive Bayes Classifier:

P(Refund=Yes|No) = 3/7
P(Refund=No|No) = 4/7
P(Refund=Yes|Yes) = 0
P(Refund=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No)=1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/7
P(Marital Status=Divorced|Yes)=1/7
P(Marital Status=Married|Yes) = 0

For taxable income:
If class=No: sample mean=110
sample variance=2975
If class=Yes: sample mean=90
sample variance=25

- P(X|Class=No) = P(Refund=No|Class=No)
  × P(Married| Class=No)
  × P(Income=120K| Class=No)
  = 4/7 × 4/7 × 0.0072 = 0.0024

- P(X|Class=Yes) = P(Refund=No| Class=Yes)
  × P(Married| Class=Yes)
  × P(Income=120K| Class=Yes)
  = 1 × 0 × 1.2 × 10^{-9} = 0

Since P(X|No)P(No) > P(X|Yes)P(Yes)

Therefore P(No|X) > P(Yes|X)

=> Class = No

Example of Naïve Bayes Classifier

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

Example of Naïve Bayes Classifier

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

A: attributes

M: mammals

N: non-mammals

$$P(A \mid M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A \mid N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A \mid M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A \mid N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

P(A|M)P(M) > P(A|N)P(N)

=> Mammals

Simple demonstration of Spam detection

- Reviews{Non-spam(+), Spam(-)}
- Represent each review by vector of words
- Learning: Use training examples to estimate

- P(+)
- P(-)
- P(rev|+)
- P(rev|-)

- Naïve Bayes condition independence assumption.

Simple demonstration of Spam detection

- Set of reviews and their classification

Doc	Review	Class
1	I loved this hotel	+
2	I hated this hotel	-
3	A great hotel, good hotel	+
4	Bad service	-
5	Great service, a good hotel	+

- 10 unique words

<I, loved, this, hotel, hated, a, great, bad, service, good>

Simple demonstration of Spam detection

- Converting the reviews into feature sets.

Doc	I	loved	this	Hotel	Hated	A	Great	Bad	Service	Good	Class
1	1	1	1	1							+
2	1		1	1	1						-
3				2		1	1			1	+
4								1	1		-
5				1		1	1		1	1	+

Simple demonstration of Spam detection

- Reviews with positive outcomes.

Doc	I	loved	this	Hotel	Hated	A	Great	Bad	Service	Good	Class
1	1	1	1	1							+
3				2		1	1			1	+
5				1		1	1		1	1	+

- P(+)= 3/5= 0.6
- Compute: p(I|+); p(loved|+); p(this|+); p(hotel|+); p(a|+); p(great|+); p(service|+); p(good|+); p(hated|+); p(bad|+).

Simple demonstration of Spam detection

- P(+)=3/5=0.6;

  $p(w_k|+)= (n_k+1)/(n+|vocabulary|)$

  p(I|+): (1+1)/(14+10)= 0.0833;
  p(loved|+): (1+1)/(14+10)= 0.0833;
  p(this|+): (1+1)/(14+10)= 0.0833;
  p(hotel|+): (5+1)/(14+10)= 0.2083;
  p(a|+): (2+1)/(14+10)= 0.125;
  p(great|+): (2+1)/(14+10)= 0.125;
  p(service|+): (1+1)/(14+10)= 0.0833;
  p(good|+): (2+1)/(14+10)= 0.125;
  p(hated|+): (0+1)/(14+10)= 0.0417;
  p(bad|+): (0+1)/(14+10)= 0.0417;

Simple demonstration of Spam detection

- Reviews with negative outcomes.

Doc	I	loved	this	Hotel	Hated	A	Great	Bad	Service	Good	Class
2	1		1	1	1						-
4								1	1		-

- P(-)= 2/5= 0.4
- Compute: p(I|-); p(loved|-); p(this|-); p(hotel|-); p(a|-); p(great|-); p(service|-); p(good|-); p(hated|-); p(bad|-).

Simple demonstration of Spam detection

- P(-)= 2/5= 0.4; **$p(w_k|-)= (n_k+1)/(n+|vocabulary|)$**

  p(I|-): (1+1)/(6+10)= 0.125;
  p(this|-): (1+1)/(6+10)= 0.125;
  p(hotel|-): (1+1)/(6+10)= 0.125;
  p(hated|-): (1+1)/(6+10)= 0.125;
  p(bad|-): (1+1)/(6+10)= 0.125;
  p(service|-): (1+1)/(6+10)= 0.125;
  p(loved|-): (0+1)/(6+10)= 0.0625;
  p(a|-): (0+1)/(6+10)= 0.0625;
  p(great|-): (0+1)/(6+10)= 0.0625;
  p(good|-): (0+1)/(6+10)= 0.0625;

Simple demonstration of Spam detection

- Now training of our classifier is done,
- Lets test the classifier with a test Review:

"**I hated this bad service**"

If v_j= +; P(+)P(I|+)P(hated|+)P(this|+)P(bad|+)P(service|+)=

$$\mathbf{6.03\times 10^{-7}}$$

If v_j= -; P(-)P(I|-)P(hated|-)P(this|-)P(bad|-)P(service|-)=

$$\mathbf{1.22\times 10^{-5}}$$

Spam

A Practice Example

Example 8.4

Class:
C1:buys_computer = 'yes'
C2:buys_computer = 'no'

Data instance
X = (age <=30,
Income = medium,
Student = yes
Credit_rating = fair)

age	income	student	credit_rating	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

A Practice Example

- $P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357

- Compute $P(X|C_i)$ for each class
  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4

- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 × 0.444 × 0.667 × 0.667 = 0.044
P(X|buys_computer = "no") = 0.6 × 0.4 × 0.2 × 0.4 = 0.019

$P(X|C_i)*P(C_i)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")