BITS Pilani
Pilani Campus

Machine Learning

Dr. Sugata Ghosal
sugata.ghosal@pilani.bits-pilani.ac.in

BITS Pilani
Pilani Campus

Lecture No. – 12 | Bayesian Learning
Date – 26/08/2023
Time: 2 PM – 4 PM

Grateful Acknowledgement : These slides were assembled leveraging the content created by the many instructors who made their course materials freely available online.

Agenda

- Naïve Bayes Classifier
- Gaussian Naïve Bayes Classifier
- Image Classification Example
- Text Classification Example
- Optimal Bayes Classifier
- Regression from Bayesian Perspective

Learning Function Approximation ?

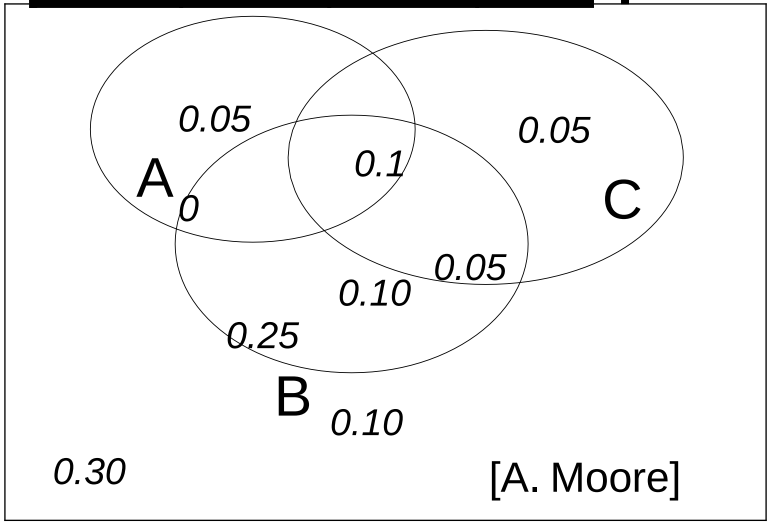
- instead of $F: X \rightarrow Y$
  learn $P(Y \mid X)$

The Joint Distribution

Recipe for making a joint distribution of M variables:

Example: Boolean variables A, B, C

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



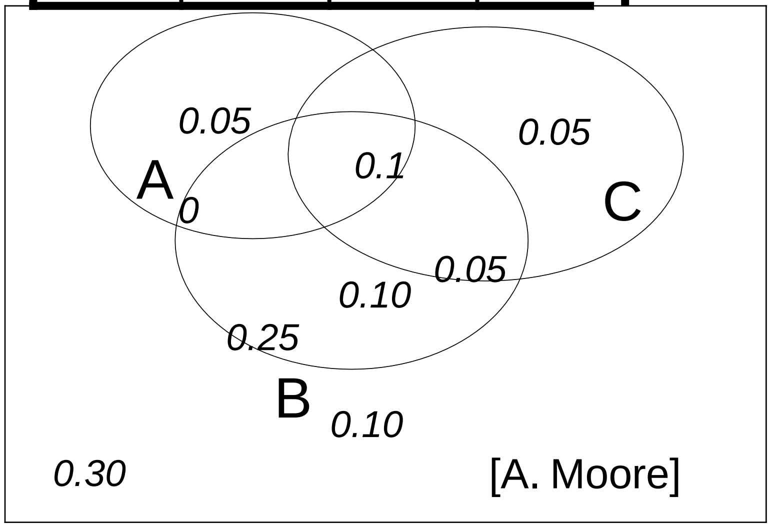
The Joint Distribution

Recipe for making a joint distribution of M variables:

1. Make a truth table listing all combinations of values (M Boolean variables $\rightarrow 2^M$ rows).

Example: Boolean variables A, B, C

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



The Joint Distribution

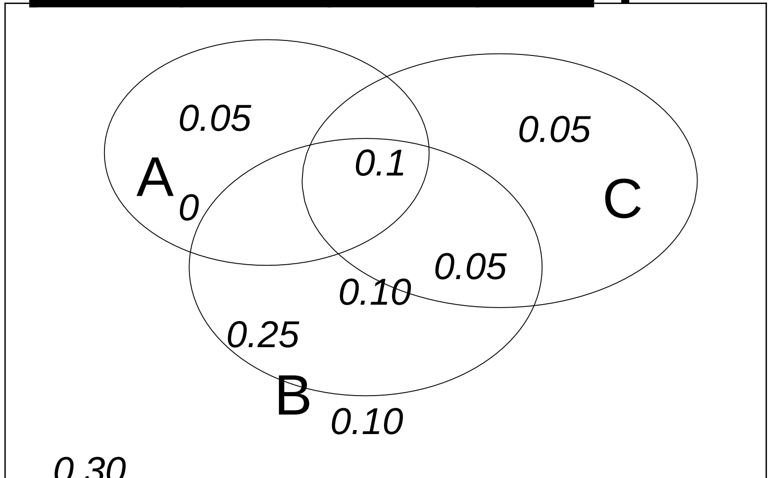
Example: Boolean variables A, B, C

Recipe for making a joint distribution of M variables:

1. Make a truth table listing all combinations of values (M Boolean variables → 2^M rows).

2. For each combination of values, say how probable it is.

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



[A. Moore]

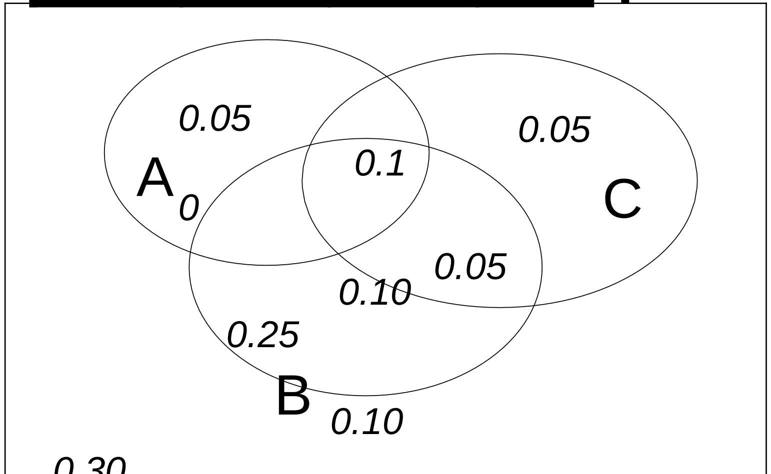
The Joint Distribution

Recipe for making a joint distribution of M variables:

1. Make a truth table listing all combinations of values (M Boolean variables →2^M rows).
2. For each combination of values, say how probable it is.
3. If you subscribe to the axioms of probability, those probabilities must sum to 1.

Example: Boolean variables A, B, C

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



[A. Moore]

Using the Joint Distribution

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

One you have the JD you can ask for the probability of **any** logical expression involving these variables

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

[A. Moore]

Using the Joint Distribution

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	

P(Poor Male) = 0.4654

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

[A. Moore]

Using the Joint Distribution

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

P(Poor) = 0.7604

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

[A. Moore]

Inference with the Joint Distribution

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

$$P(E_1 \mid E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{\text{rows matching } E_1 \text{ and } E_2} P(\text{row})}{\sum_{\text{rows matching } E_2} P(\text{row})}$$

P(Male | Poor) = 0.4654 / 0.7604 = 0.612

[A. Moore]

Learning and the Joint Distribution

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

Suppose we want to learn the function f: <G, H> → W
Equivalently, P(W | G, H)

Solution: learn joint distribution from data, calculate P(W | G, H)

e.g., P(W=rich | G = female, H = 40.5-) =

[A. Moore]

innovate achieve lead

sounds like the solution to learning F : X → Y or P(Y | X)

Main problem: learning P(Y|X) can require more data than we have

consider learning Joint Dist. with 100 attributes

# of rows in this table? $2^{100} \sim 10^{30}$

# of people on earth? 10^{9}

fraction of rows with 0 training examples? 0.9999

What to do?

1. Be smart about how we estimate probabilities from sparse data
   - maximum likelihood estimates
   - maximum a posteriori estimates

1. Be smart about how we estimate probabilities

Estimating Probability of Heads

- I show you the above coin X, and hire you to estimate the probability that it will turn up heads $(X = 1)$ or tails $(X = 0)$
- You flip it repeatedly, observing
  - it turns up heads α_1 times
  - it turns up tails α_0 times
- Your estimate for $P(X = 1)$ is....?

Estimating θ = P(X=1)

X=1 X=0

Case A:

– 100 flips: 51 Heads (X=1), 49 Tails (X=0)

Case B:

– 3 flips: 2 Heads (X=1), 1 Tails (X=0)

Case C: (online learning)

– keep flipping, want single learning algorithm that gives reasonable estimate after each flip

Principles for Estimating Probabilities

Principle 1 (maximum likelihood):

- choose parameters θ that maximize **P(data | θ)**

Principle 2 (maximum a posteriori prob.):

- choose parameters θ that maximize **P(θ | data)**

Maximum Likelihood Estimation

P(X=1) = θ   P(X=0) = (1-θ)

X=1   X=0

Data D:

$$\{ 0, 1, 1, 1, 0, \ldots.\}$$

Flips produce data D with α_1 heads, α_0 tails

- flips are independent, identically distributed 1's and 0's (Bernoulli)
- α_1 and α_0 are counts that sum these outcomes (Binomial)

$$P(D|\theta) = P(\alpha_1, \alpha_0|\theta) = \theta^{\alpha_1}(1-\theta)^{\alpha_0}$$

Maximum Likelihood Estimate for Θ

$$\hat{\theta} = \arg\max_{\theta} \ln P(\mathcal{D} \mid \theta)$$
$$= \arg\max_{\theta} \ln \theta^{\alpha_H}(1-\theta)^{\alpha_T}$$

- Set derivative to zero: $\boxed{\frac{d}{d\theta} \ln P(\mathcal{D} \mid \theta) = 0}$

$$\hat{\theta} = \arg\max_{\theta} \ln P(D|\theta)$$
$$= \arg\max_{\theta} \ln \left[\theta^{\alpha_1}(1-\theta)^{\alpha_0}\right]$$

- Set derivative to zero: $\boxed{\frac{d}{d\theta} \ln P(\mathcal{D} \mid \theta) = 0}$

hint: $\frac{\partial \ln \theta}{\partial \theta} = \frac{1}{\theta}$

[C. Guestrin]

Summary: Maximum Likelihood Estimate

X=1 X=0

P(X=1) = θ

P(X=0) = 1-θ

(Bernoulli)

- Each flip yields boolean value for X

$$X \sim \text{Bernoulli: } P(X) = \theta^X (1-\theta)^{(1-X)}$$

- Data set D of independent, identically distributed (iid) flips produces α_1 ones, α_0 zeros (Binomial)

$$P(D|\theta) = P(\alpha_1, \alpha_0|\theta) = \theta^{\alpha_1}(1-\theta)^{\alpha_0}$$

$$\hat{\theta}^{MLE} = \text{argmax}_\theta\, P(D|\theta) = \frac{\alpha_1}{\alpha_1+\alpha_0}$$

Principles for Estimating Probabilities

Principle 1 (maximum likelihood):

- choose parameters θ that maximize $P(\text{data} \mid \theta)$

Principle 2 (maximum a posteriori prob.):

- choose parameters θ that maximize

$$P(\theta \mid \text{data}) = \frac{P(\text{data} \mid \theta)\, P(\theta)}{P(\text{data})}$$

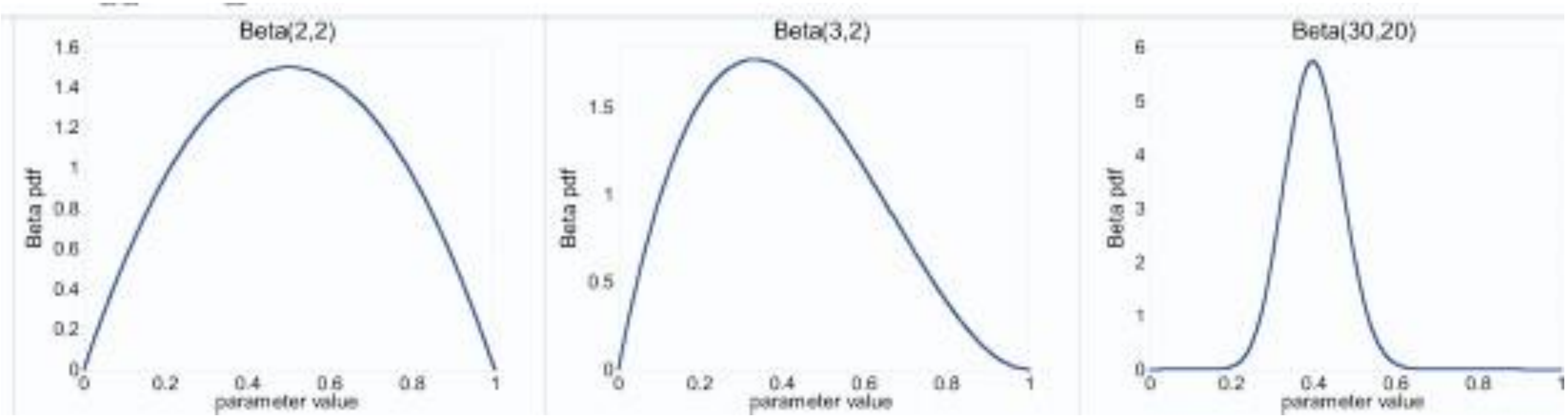
Example prior distribution – P(θ)

$$P(\theta) = \frac{\theta^{\beta_H - 1}(1-\theta)^{\beta_T - 1}}{B(\beta_H, \beta_T)} \sim Beta(\beta_H, \beta_T)$$

- Likelihood function: $P(\mathcal{D} \mid \theta) = \theta^{\alpha_H}(1-\theta)^{\alpha_T}$
- Posterior: $P(\theta \mid \mathcal{D}) \propto P(\mathcal{D} \mid \theta) P(\theta)$

Beta prior distribution –P(θ)

$$P(\theta) = \frac{\theta^{\beta_H - 1}(1-\theta)^{\beta_T - 1}}{B(\beta_H, \beta_T)} \sim Beta(\beta_H, \beta_T)$$



[C. Guestrin]

Eg. 1 Coin flip problem

Likelihood is ~ Binomial

$$P(\mathcal{D} \mid \theta) = \theta^{\alpha_H}(1-\theta)^{\alpha_T}$$

If prior is Beta distribution,

$$P(\theta) = \frac{\theta^{\beta_H - 1}(1-\theta)^{\beta_T - 1}}{B(\beta_H, \beta_T)} \sim Beta(\beta_H, \beta_T)$$

Then posterior is Beta distribution

$$P(\theta|D) \sim Beta(\alpha_H + \beta_H, \alpha_H + \beta_H)$$

and MAP estimate is therefore

$$\hat{\theta}^{MAP} = \frac{\alpha_H + \beta_H - 1}{(\alpha_H + \beta_H - 1) + (\alpha_T + \beta_T - 1)}$$

Eg. 2 Dice roll problem (6 outcomes instead of 2)

Likelihood is ~ Multinomial($\theta = \{\theta_1, \theta_2, \ldots, \theta_k\}$)

$$P(\mathcal{D} \mid \theta) = \theta_1^{\alpha_1} \theta_2^{\alpha_2} \ldots \theta_k^{\alpha_k}$$

If prior is Dirichlet distribution,

$$P(\theta) = \frac{\theta_1^{\beta_1 - 1}\ \theta_2^{\beta_2 - 1} \ldots \theta_k^{\beta_k - 1}}{B(\beta_1, \ldots, \beta_k)} \sim \text{Dirichlet}(\beta_1, \ldots, \beta_k)$$

Then posterior is Dirichlet distribution

$$P(\theta|D) \sim \text{Dirichlet}(\beta_1 + \alpha_1, \ldots, \beta_k + \alpha_k)$$

and MAP estimate is therefore

$$\hat{\theta}_i^{MAP} = \frac{\alpha_i + \beta_i - 1}{\sum_{j=1}^{k}(\alpha_j + \beta_j - 1)}$$

Aside: Some terminology

- Likelihood function: P(data | θ)
- Prior: P(θ)
- Posterior: P(θ | data)

- **Conjugate prior**: P(θ) is the conjugate prior for likelihood function P(data | θ) if the forms of P(θ) and P(θ | data) are the same.

Two Principles for Estimating Parameters

- Maximum Likelihood Estimate (MLE): choose θ that maximizes probability of observed data $\mathcal{D}$

$$\widehat{\theta} = \arg\max_{\theta} \; P(\mathcal{D} \mid \theta)$$

- Maximum a Posteriori (MAP) estimate: choose θ that is most probable given prior probability and the data

$$\widehat{\theta} = \arg\max_{\theta} \; P(\theta \mid \mathcal{D})$$

$$= \arg\max_{\theta} \; = \frac{P(\mathcal{D} \mid \theta) P(\theta)}{P(\mathcal{D})}$$

Maximum Likelihood Estimate

X=1 X=0

P(X=1) = θ

P(X=0) = 1-θ

(Bernoulli)

- Each flip yields boolean value for X

$$X \sim \text{Bernoulli: } P(X) = \theta^X (1-\theta)^{(1-X)}$$

- Data set D of independent, identically distributed (iid) flips produces α_1 ones, α_0 zeros

$$P(D|\theta) = P(\alpha_1, \alpha_0|\theta) = \theta^{\alpha_1}(1-\theta)^{\alpha_0}$$

$$\hat{\theta}^{MLE} = \arg\max_{\theta} P(D|\theta) = \frac{\alpha_1}{\alpha_1 + \alpha_0}$$

Maximum A Posteriori (MAP) Estimate

X=1 X=0

- Data set D of independent, identically distributed (iid) flips produces α_1 ones, α_0 zeros

$$P(D|\theta) = P(\alpha_1, \alpha_0|\theta) = \theta^{\alpha_1}(1-\theta)^{\alpha_0}$$

- Assume prior $P(\theta) = Beta(\beta_1, \beta_0) = \frac{1}{B(\beta_1,\beta_0)}\, \theta^{\beta_1 - 1}(1-\theta)^{\beta_0 - 1}$
- Then

$$\hat{\theta}^{MAP} = \arg\max_{\theta} P(D|\theta)P(\theta) = \frac{\alpha_1 + \beta_1 - 1}{(\alpha_1 + \beta_1 - 1) + (\alpha_0 + \beta_0 - 1)}$$

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

innovate achieve lead

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?

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

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

To estimate $P(Y| X_1, X_2, \ldots X_n)$
how many parameters do we need to estimate?

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

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:

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

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 <u>conditionally independent</u> 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:

$$\begin{aligned} P(X_1, X_2|Y) &= P(X_1|X_2, Y)P(X_2|Y) \\ &= P(X_1|Y)P(X_2|Y) \end{aligned}$$

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

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

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

Naïve Bayes in a Nutshell

Bayes rule:

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

Assuming conditional independence among X_i's:

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

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

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

Naïve Bayes Algorithm – discrete X_i

- Train Naïve Bayes (examples)

  for each* value y_k

  estimate $\pi_k \equiv P(Y = y_k)$

  for each* value x_{ij} of each attribute X_i

  estimate $\theta_{ijk} \equiv P(X_i = x_{ij} | Y = y_k)$

- Classify (X^{new})

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

$$Y^{new} \leftarrow \arg\max_{y_k} \; \pi_k \prod_i \theta_{ijk}$$

* probabilities must sum to 1, so need estimate only n-1 of these...

Estimating Parameters: *Y, X_i* discrete

Maximum likelihood estimates (MLE's):

$$\hat{\pi}_k = \hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\}}{|D|}$$

$$\hat{\theta}_{ijk} = \hat{P}(X_i = x_{ij}|Y = y_k) = \frac{\#D\{X_i = x_{ij} \wedge Y = y_k\}}{\#D\{Y = y_k\}}$$

Number of items in dataset D for which $Y=y_k$

Estimating Parameters: Y, X_i discrete

MAP estimates (Beta, Dirichlet priors):

$$\hat{\pi}_k = \hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\} + (\beta_k - 1)}{|D| + \sum_m (\beta_m - 1)}$$

Only difference:
"imaginary" examples

$$\hat{\theta}_{ijk} = \hat{P}(X_i = x_j | Y = y_k) = \frac{\#D\{X_i = x_j \wedge Y = y_k\} + (\beta_k - 1)}{\#D\{Y = y_k\} + \sum_m (\beta_m - 1)}$$

Naïve Bayes Classification Example 1

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

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

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

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

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

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

Issues with Naïve Bayes Classifier

Naïve 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) = 2/3
P(Marital Status = Divorced | Yes) = 1/3
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(Yes) = 3/10
  P(No) = 7/10
- P(Yes | Married) = 0 x 3/10 / P(Married)
  P(No | Married) = 4/7 x 7/10 / P(Married)

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

Issues with Naïve Bayes Classifier

Consider the table with Tid = 7 deleted

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
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Naïve Bayes Classifier:

P(Refund = Yes | No) = 2/6
P(Refund = No | No) = 4/6
P(Refund = Yes | Yes) = 0
P(Refund = No | Yes) = 1
P(Marital Status = Single | No) = 2/6
P(Marital Status = Divorced | No) = 0
P(Marital Status = Married | No) = 4/6
P(Marital Status = Single | Yes) = 2/3
P(Marital Status = Divorced | Yes) = 1/3
P(Marital Status = Married | Yes) = 0/3
For Taxable Income:
If class = No: sample mean = 91
sample variance = 685
If class = No: sample mean = 90
sample variance = 25

Given X = (Refund = Yes, Divorced, 120K)

P(X | No) = 2/6 X 0 X 0.0083 = 0

P(X | Yes) = 0 X 1/3 X 1.2 X 10^{-9} = 0

Naïve Bayes will not be able to classify X as Yes or No!

Issues with Naïve Bayes Classifier

- If one of the conditional probabilities is zero, then the entire expression becomes zero
- Need to use other estimates of conditional probabilities than simple fractions
- Probability estimation:

$$\text{Original}: P(A_i \mid C) = \frac{N_{ic}}{N_c}$$

$$\text{Laplace}: P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c}$$

$$\text{m-estimate}: P(A_i \mid C) = \frac{N_{ic} + mp}{N_c + m}$$

c: number of classes

p: prior probability of the class

m: parameter

N_c: number of instances in the class

N_{ic}: number of instances having attribute value A_i in class c

A Simple Example

Text	Tag
“A great game”	Sports
“The election was over”	Not sports
“Very clean match”	Sports
“A clean but forgettable game”	Sports
“It was a close election”	Not sports

Which tag does the sentence *A very close game* belong to? i.e. P(sports| *A very close game*)

Feature Engineering: Bag of words i.e use word frequencies without considering order

Using Bayes Theorem:

P(sports| ***A very close game*****)**

$$= \frac{P(\textit{A very close game} \mid \text{sports})\, P(\text{sports})}{P(\textit{A very close game})}$$

We assume that every word in a sentence is **independent** of the other ones

“close” doesn’t appear in sentences of sports tag, So P(close | sports) = 0, which makes product 0

Laplace smoothing

- Laplace smoothing: we add 1 or in general constant k to every count so it’s never zero.
- To balance this, we add the number of possible words to the divisor, so the division will never be greater than 1
- In our case, the possible words are ['a', 'great', 'very', 'over', 'it', 'but', 'game', 'election', 'clean', 'close', 'the', 'was', 'forgettable', 'match'].

Apply Laplace Smoothing

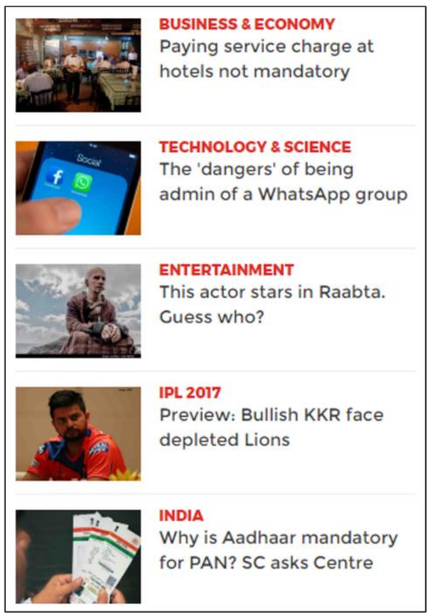
Word	P(word | Sports)	P(word | Not Sports)
a	2+1 / 11+14	1+1 / 9+14
very	1+1 / 11+14	0+1 / 9+14
close	0+1 / 11+14	1+1 / 9+14
game	2+1 / 11+14	0+1 / 9+14

$$P(a|Sports) \times P(very|Sports) \times P(close|Sports) \times P(game|Sports) \times P(Sports)$$
$$= 2.76 \times 10^{-5}$$
$$= 0.0000276$$

$$P(a|Not\ Sports) \times P(very|Not\ Sports) \times P(close|Not\ Sports) \times P(game|Not\ Sports) \times P(Not\ Sports)$$
$$= 0.572 \times 10^{-5}$$
$$= 0.00000572$$

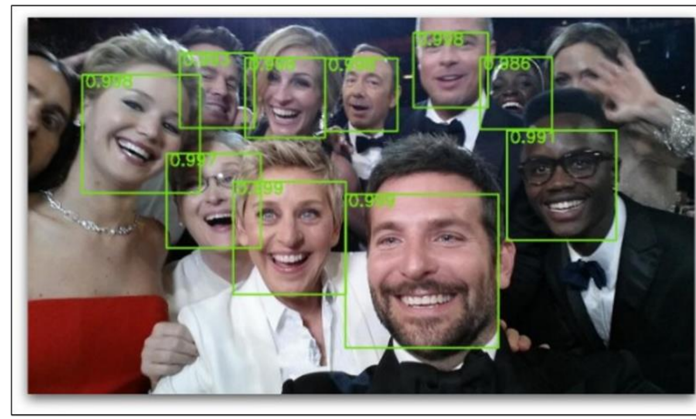
Naïve Bayes Classifier Applications

Categorizing News



Email Spam Detection

Face Recognition



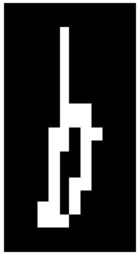
Sentiment Analysis

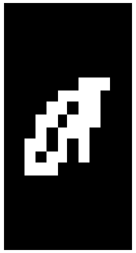
Estimating Parameters: X_i

Continuous

What if features are continuous?

- E.g., character recognition: X_i is intensity at ith pixel





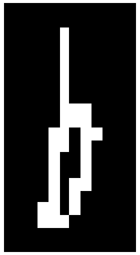
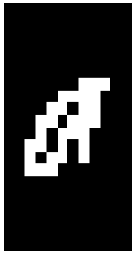
- Gaussian Naïve Bayes (GNB):

$$P(X_i = x \mid Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}}\, e^{-\frac{(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

distribution of feature X_i is Gaussian with a mean and variance that can depend on the label y_k and which feature X_i it is

What if features are continuous?

- E.g., character recognition: X_i is intensity at ith pixel

- Gaussian Naïve Bayes (GNB):

$$P(X_i = x|Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}}\, e^{-\frac{(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

- Different mean and variance for each class k and each pixel i.
- Sometimes assume variance:
  - Is independent of Y (i.e., just have σ_i)
  - Or independent of X (i.e., just have σ_k)
  - Or both (i.e., just have σ)

Estimating parameters: Y discrete, X_i continuous

- Maximum likelihood estimates:

$$\hat{\mu}_{MLE} = \frac{1}{N}\sum_{j=1}^{N} x_j$$

$$\hat{\mu}_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j X_i^j \delta(Y^j = y_k)$$

kth class

jth training image

ith pixel in jth training image

$$\hat{\sigma}^2_{unbiased} = \frac{1}{N-1}\sum_{j=1}^{N}(x_j - \hat{\mu})^2$$

$$\hat{\sigma}^2_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k) - 1} \sum_j (X_i^j - \hat{\mu}_{ik})^2 \delta(Y^j = y_k)$$

Naive Bayes Classifier for Text

- Along with decision trees, neural networks, one of the most practical learning methods.
- When to use
  - Moderate or large training set available
  - Attributes that describe instances are conditionally independent given classification
- Successful applications:
  - Diagnosis
  - Classifying text documents

Learning to Classify Text

- Why?
  - Learn which news articles are of interest
  - Learn to classify web pages by topic
- Naive Bayes is among most effective algorithms
- What attributes shall we use to represent text documents??

Baseline: Bag of Words Approach


the world of
TOTAL

All About The Company
Global Activities
Corporate Structure
TOTAL's Story
Upstream Strategy
Downstream Strategy
Chemicals Strategy
TOTAL Foundation
Homepage

all about the company

Our energy exploration, production, and distribution operations span the globe, with activities in more than 100 countries.

At TOTAL, we draw our greatest strength from our fast-growing oil and gas reserves. Our strategic emphasis on natural gas provides a strong position in a rapidly expanding market.

Our expanding refining and marketing operations in Asia and the Mediterranean Rim complement already solid positions in Europe, Africa, and the U.S.

Our growing specialty chemicals sector adds balance and profit to the core energy business.

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

Case Study: Text Classification

- Classify e-mails
  - $Y = \{$Spam, NotSpam$\}$
- Classify news articles
  - $Y =$ what is the topic of the article?
- Classify webpages
  - $Y = \{$student, professor, project, …$\}$

- What about the features X?
  - The text!

Features X are entire document - X_i for ith word in article

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
```

Naïve Bayes for Text Classification

- **Naïve Bayes assumption helps a lot!**
  - $P(X_i = x_i | Y = y)$ is just the probability of observing word x_i at the ith position in a document on topic y.
  - Assume X_i is independent of all other words in document given the label y: $P(X_i = x_i | Y = y, X_{-i}) = P(X_i = x_i | Y = y)$.

$$h_{NB}(x) = \arg\max_y P(y) \prod_{i=1}^{\text{lengthDoc}} P(X_i = x_i | y)$$

  - For each label y, have 1000 distributions of size 10000 to estimate.
  - This is 10000×1000 items, which is big but much less than 10000^{1000}...

Bag of Words Model

- Typical additional assumption – **Position in document doesn't matter**:

$$P(X_i = x_i \mid Y = y) = P(X_k = x_i \mid Y = y)$$

the probability distributions of words are the same at each position: $P_i = P_j$ for all i, j.

- **"Bag of Words"** model – order of words in the document is ignored
- Now, only 10000 quantities $P(x_i|y)$ to estimate for each label y (the 10000 possible values that x_i can be) plus the prior.

$$h_{NB}(x) = \arg\max_y P(y) \prod_{i=1}^{1000} P(x_i|y)$$

Bag of Words model

- Typical additional assumption – **Position in document doesn't matter**:

$$P(X_i = x_i \mid Y = y) = P(X_k = x_i \mid Y = y)$$

- **"Bag of Words"** model – order of words on the page ignored

Can simplify further:

$$\prod_{i=1}^{lengthDoc} P(x_i|y) = \prod_{w=1}^{W} P(w|y)^{count(w)}$$

Bag of Words representation

- Since we are assuming the order of words doesn't matter, an alternative representation of document is as vector of counts:
  - $x^{(j)}$ = number of occurrences of word j in document x.
  - Typical document: $[0\ 0\ 1\ 0\ 0\ 3\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 2\ 0\ 0\ \ldots]$
  - Called "bag of words" or "term vector" or "vector space model" representation

Naïve Bayes with Bag of Words for text classification

- Learning phase
  - Class Prior $P(Y)$
  - $P(X_i|Y)$
- Test phase:
  - For each document
    - Use naïve Bayes decision rule

$$h_{NB}(x) = \arg\max_y P(y) \prod_{i=1}^{1000} P(x_i|y)$$

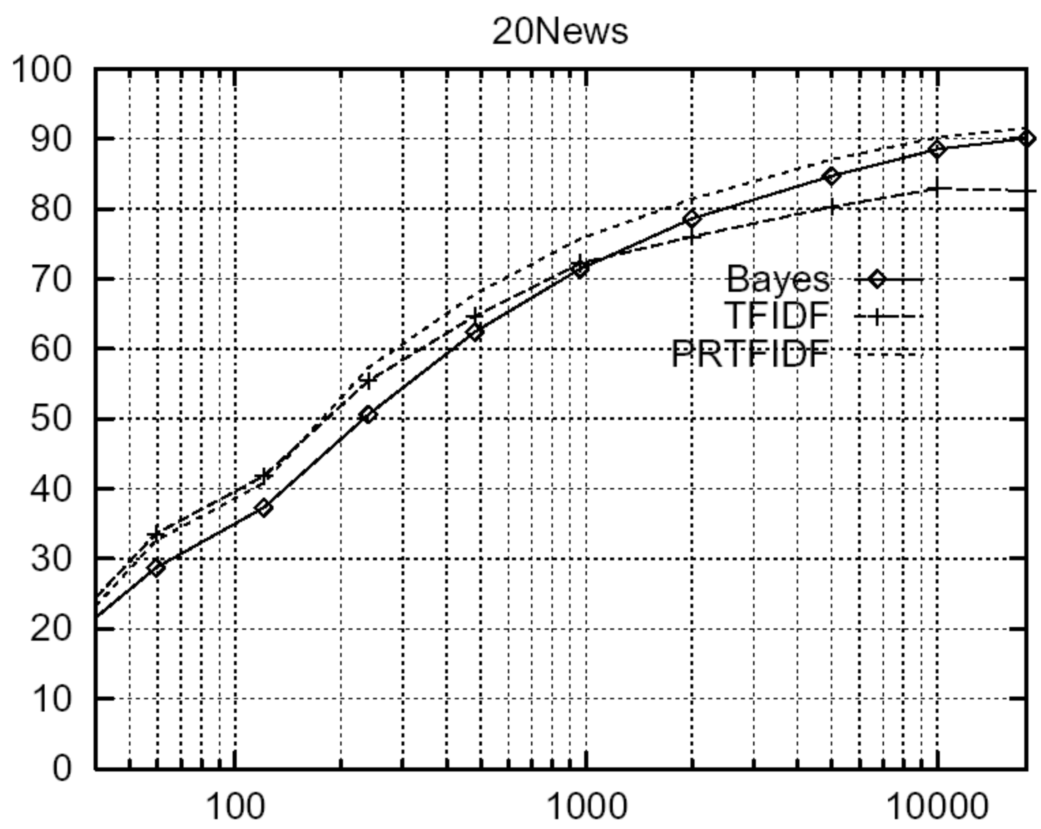
Twenty NewsGroups

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

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

- Naive Bayes: 89% classification accuracy

Learning Curve for 20 Newsgroups



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

Summary: 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 $P(a_i = w_k \mid v_j)$ is probability that word in position *i* is w_k, given v_j

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

Summary: Learning to Classify Text

LEARN_NAIVE_BAYES_TEXT (*Examples*, V)

1. *collect all words and other tokens that occur in Examples*

- *Vocabulary* ← all distinct words and other tokens in *Examples*

2. *calculate the required $P(v_j)$ and $P(w_k \mid v_j)$ probability terms*

- For each target value v_j in V do
  - $docs_j$ ← subset of *Examples* for which the target value is v_j
  - $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
  - $Text_j$ ← a single document created by concatenating all members of $docs_j$

Summary: Learning to Classify Text

- $n \leftarrow$ 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 $v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_{i \in positions} P(a_i|v_j)$

Probabilistic Generative Model versus Probabilistic Discriminative Model

Generative	Discriminative
Ex: Naïve Bayes	Ex: Logistic Regression
Estimate $P(Y)$ and $P(X\|Y)$	Finds class label directly $P(Y\|X)$
Prediction $\hat{y} = \text{argmax}_y\, P(Y = y)P(X = xnew\|Y = y)$	Prediction $\hat{y} = P(Y = y\|X = xnew)$

Most Probable Classification of New Instances

- So far we've sought the most probable *hypothesis* given the data D (i.e., h_{MAP})
- Given new instance x, what is its most probable *classification*?
  - $h_{MAP}(x)$ is not the most probable classification!
- Consider:
  - Three possible hypotheses:

$$P(h_1|D) = .4,\ P(h_2|D) = .3,\ \ P(h_3|D) = .3$$

  - Given new instance x,

$$h_1(x) = +,\ \ h_2(x) = -,\ \ h_3(x) = -$$

  - What's most probable classification of x?

Bayes Optimal Classifier

- **Bayes optimal classification:**

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

- Example:

$$P(h_1|D) = .4,\ P(-|h_1) = 0,\ \ P(+|h_1) = 1$$
$$P(h_2|D) = .3,\ P(-|h_2) = 1,\ \ P(+|h_2) = 0$$
$$P(h_3|D) = .3,\ P(-|h_3) = 1,\ \ P(+|h_3) = 0$$

therefore

$$\sum_{h_i \in H} P(+|h_i)P(h_i|D) = .4$$
$$\sum_{h_i \in H} P(-|h_i)P(h_i|D) = .6$$

and

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) = -$$

Gibbs Classifier

- Bayes optimal classifier provides best result, but can be expensive if many hypotheses.
- Gibbs algorithm:
  1. Choose one hypothesis at random, according to $P(h|D)$
  2. Use this to classify new instance
- Surprising fact: Assume target concepts are drawn at random from H according to priors on H. Then:

$$E[error_{Gibbs}] \leq 2E\,[error_{BayesOptional}]$$

Features of Bayesian learning

- Each observed training example can incrementally decrease or increase the estimated probability that a hypothesis is correct.
- Flexible approach to learning than algorithms that completely eliminate a hypothesis if it is found to be inconsistent with any single example.
- Bayesian methods can accommodate hypotheses that make probabilistic predictions (e.g., hypotheses such as "this pneumonia patient has a 93% chance of complete recovery").

Features of Bayesian learning

- Prior knowledge can be combined with observed data to determine the final probability of a hypothesis.
- Prior knowledge is provided by asserting
  - prior probability for each candidate hypothesis, and
  - probability distribution over observed data for each possible hypothesis.
- New instances can be classified by combining the predictions of multiple hypotheses, weighted by their probabilities.

Practical Issues of Bayesian learning

- Require initial knowledge of many probabilities
  - Often estimated based on background knowledge, previously available data, and assumptions about the form of the underlying distributions.
- Significant computational cost required to determine the Bayes optimal hypothesis in the general case (linear in the number of candidate hypotheses)

Logistic Regression from Bayesian Perspective

- Consider learning f: X →Y, where
  - X is a vector of real-valued features, < $X_1 \ldots X_n$ >
  - Y is boolean
  - assume all X_i are conditionally independent given Y
  - model $P(X_i \mid Y = y_k)$ as Gaussian $N(\mu_{ik},\sigma_i)$
  - model P(Y) as Bernoulli ($P(Y=1) = \pi$)

- What does that imply about the form of P(Y|X)?

innovate achieve lead

Derive form for P(Y|X) for Gaussian $P(X_i | Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

$$P(Y=1|X) = \frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1) + P(Y=0)P(X|Y=0)}$$

$$= \frac{1}{1 + \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}}$$

$$P(x \mid y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} \; e^{\frac{-(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

$$= \frac{1}{1 + \exp(\ln \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)})}$$

$$P(Y=1) = \pi$$

$$= \frac{1}{1 + \exp(\ (\ln \frac{1-\pi}{\pi}) + \boxed{\sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)}})}$$

$$\boxed{\sum_i \left(\frac{\mu_{i0} - \mu_{i1}}{\sigma_i^2} X_i + \frac{\mu_{i1}^2 - \mu_{i0}^2)}{2\sigma_i^2} \right)}$$

$$P(Y=1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

Very convenient!

$$P(Y=1|X=<X_1,...X_n>) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

implies

$$P(Y=0|X=<X_1,...X_n>) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

implies

$$\frac{P(Y=0|X)}{P(Y=1|X)} = exp(w_0+\sum_i w_i X_i)$$

linear classification rule!

implies

$$\ln\frac{P(Y=0|X)}{P(Y=1|X)} = w_0+\sum_i w_i X_i$$

Training Logistic Regression: MCLE

- Choose parameters $W=<w_0, \ldots w_n>$ to <u>maximize conditional likelihood</u> of training data

  where
$$P(Y=0|X,W) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$
$$P(Y=1|X,W) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

- Training data D = $\{\langle X^1, Y^1\rangle, \ldots \langle X^L, Y^L\rangle\}$
- Data likelihood = $\prod_l P(X^l, Y^l|W)$
- Data <u>conditional</u> likelihood = $\prod_l P(Y^l|X^l, W)$

$$W_{MCLE} = \arg\max_W \prod_l P(Y^l|W, X^l)$$

Expressing Conditional Log Likelihood

$$l(W) \equiv \ln \prod_l P(Y^l|X^l, W) = \sum_l \ln P(Y^l|X^l, W)$$

$$P(Y=0|X,W) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

$$P(Y=1|X,W) = \frac{exp(w_0 + \sum_i w_i X_i)}{1 + exp(w_0 + \sum_i w_i X_i)}$$

$$\begin{aligned} l(W) &= \sum_l Y^l \ln P(Y^l = 1|X^l, W) + (1 - Y^l) \ln P(Y^l = 0|X^l, W) \\ &= \sum_l Y^l \ln \frac{P(Y^l = 1|X^l, W)}{P(Y^l = 0|X^l, W)} + \ln P(Y^l = 0|X^l, W) \\ &= \sum_l Y^l (w_0 + \sum_i^n w_i X_i^l) - \ln(1 + exp(w_0 + \sum_i^n w_i X_i^l)) \end{aligned}$$

MLE vs MAP

- Maximum conditional likelihood estimate

$$W \leftarrow \arg\max_W \ \ln \prod_l P(Y^l|X^l, W)$$

$$\boxed{w_i \leftarrow w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))}$$

- Maximum a posteriori estimate with prior W~N(0,σI)

$$W \leftarrow \arg\max_W \ \ln[P(W) \ \prod_l P(Y^l|X^l, W)]$$

$$\boxed{w_i \leftarrow w_i - \eta\lambda w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))}$$

MAP estimates and Regularization

- Maximum a posteriori estimate with prior W~N(0,σI)

$$W \leftarrow \arg\max_W \ \ln[P(W) \prod_l P(Y^l|X^l, W)]$$

$$\boxed{w_i \leftarrow w_i - \eta\lambda w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))}$$

λ is called a "<u>regularization</u>" term

- helps reduce overfitting
- keep weights nearer to zero (if P(W) is zero mean Gaussian prior), or whatever the prior suggests
- used very frequently in Logistic Regression

Naïve Bayes versus Logistic Regression

- Naïve Bayes are Generative Models which Logistic Regression are Discriminative Models
- Naïve Bayes easy to construct
- Naïve Bayes better on smaller datasets
- Naive Bayes also assumes that the features are conditionally independent. Real data sets are never perfectly independent
- When the training size reaches infinity, logistic regression performs better than the generative model Naive Bayes.
  - Optional reading by Ng and Jordan has proofs and experiments
- Logistic regression allows arbitrary features

Naïve Bayes vs Logistic Regression

Consider Y boolean, X_i continuous, $X=<X_1 \ldots X_n>$

Number of parameters:

- NB: 4n +1
- LR: n+1

Estimation method:

- NB parameter estimates are uncoupled
- LR parameter estimates are coupled

G. Naïve Bayes vs. Logistic Regression

[Ng & Jordan, 2002]

Recall two assumptions deriving form of LR from GNBayes:

1. X_i conditionally independent of X_k given Y
2. $P(X_i \mid Y = y_k) = N(\mu_{ik},\sigma_i)$, $\leftarrow$ not $N(\mu_{ik},\sigma_{ik})$

Consider three learning methods:

•GNB (assumption 1 only) -- decision surface can be non-linear
•GNB2 (assumption 1 and 2) – decision surface linear
•LR -- decision surface linear, trained without assumption 1.

Which method works better if we have *infinite* training data, and...

•Both (1) and (2) are satisfied: LR = GNB2 = GNB

•(1) is satisfied, but not (2) : GNB > GNB2, GNB > LR, LR > GNB2

•Neither (1) nor (2) is satisfied: GNB>GNB2, LR > GNB2, LR><GNB

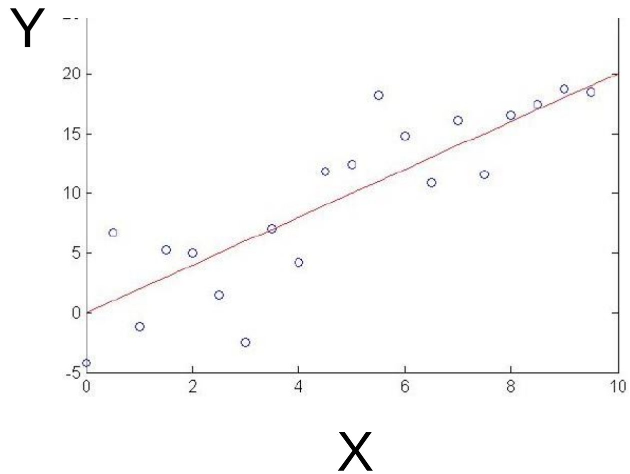
Maximum likelihood and least-squared error hypotheses

- A set of m training examples is provided, where the target value of each example is corrupted by random noise drawn according to a Normal probability distribution.
- Each training example is a pair of the form

(x_i, d_i) where $d_i = f(x_i) + e_i$. Here $f(x_i)$ is the noise-free value of the target function and e_i is a random variable representing the noise.

  – values of the e_i are drawn independently and that they are distributed according to a Normal distribution with zero mean

Choose parameterized form for P(Y|X; θ)



Assume Y is some deterministic f(X), plus random noise

$$y = f(x) + \epsilon \qquad \text{where} \quad \epsilon \sim N(0, \sigma)$$

Therefore Y is a random variable that follows the distribution

$$p(y|x) = N(f(x), \sigma)$$

and the expected value of y for any given x is f(x)

Training Linear Regression : Maximum Conditional Likelihood Estimate (MCLE)

$$p(y|x;W) = N(w_0 + w_1 x, \sigma)$$

How can we learn W from the training data?

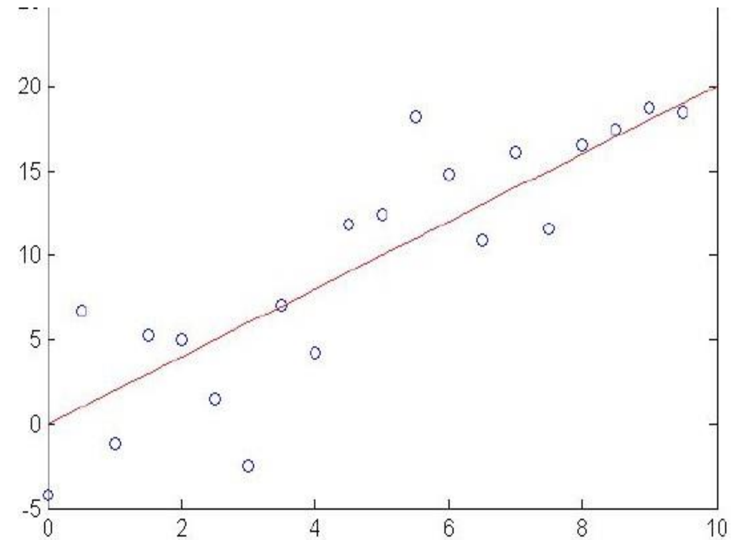
Learn Maximum Conditional Likelihood Estimate!

$$W_{MCLE} = \arg\max_W \prod_l p(y^l|x^l, W)$$

$$W_{MCLE} = \arg\max_W \sum_l \ln p(y^l|x^l, W)$$

where

$$p(y|x;W) = \frac{1}{\sqrt{2\pi\sigma^2}} \; e^{-\frac{1}{2}\left(\frac{y-f(x;W)}{\sigma}\right)^2}$$



Training Linear Regression: MCLE

Learn Maximum Conditional Likelihood Estimate

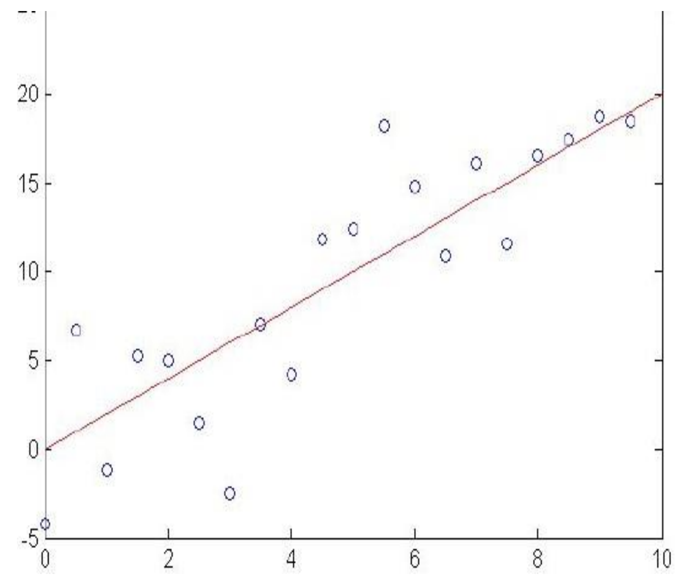
$$W_{MCLE} = \arg\max_{W} \sum_{l} \ln p(y^l | x^l, W)$$

where

$$p(y|x;W) = \frac{1}{\sqrt{2\pi\sigma^2}} \; e^{-\frac{1}{2}\left(\frac{y-f(x;W)}{\sigma}\right)^2}$$

so:

$$W_{MCLE} = \arg\min_{W} \sum_{l} (y - f(x;W))^2$$



Decision Theory

- Suppose **x** is an input vector together with a corresponding vector **t** of target variables
- Goal: predict **t** given a new value for **x.**
- The joint probability distribution ***p(x, t)*** provides a complete summary of the uncertainty associated with these variables.
- Determination of ***p(x, t)*** from a set of training data is called ***inference***

BITS Pilani

Decision Theory

Inference step

Determine either $p(t|\mathbf{x})$ or $p(\mathbf{x}, t)$.

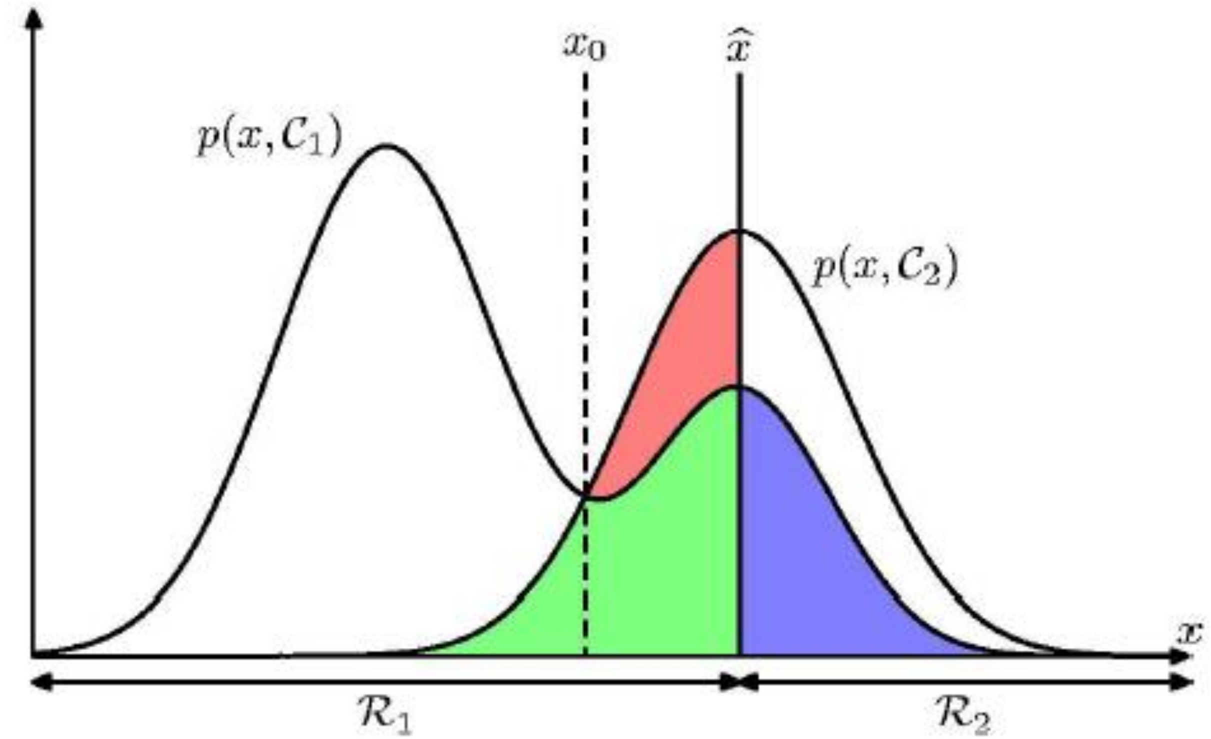
Decision step

For given $\mathbf{x}$, determine optimal t.

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})}$$

BITS Pilani

Minimum Misclassification Rate



$$
\begin{aligned}
p(\text{mistake}) &= p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\
&= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2)\,\mathrm{d}\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1)\,\mathrm{d}\mathbf{x}.
\end{aligned}
$$

BITS Pilani

Minimum Misclassification Rate

$$\begin{aligned} p(\text{correct}) &= \sum_{k=1}^{K} p(\mathbf{x} \in \mathcal{R}_k, \mathcal{C}_k) \\ &= \sum_{k=1}^{K} \int_{\mathcal{R}_k} p(\mathbf{x}, \mathcal{C}_k) \, \mathrm{d}\mathbf{x} \end{aligned}$$

BITS Pilani