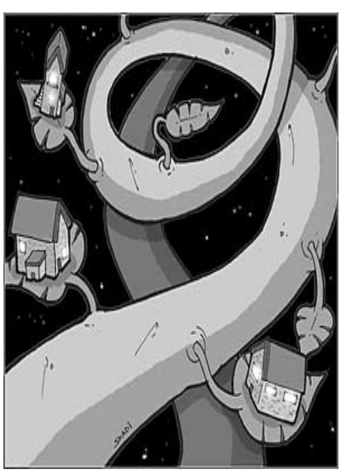
Bayes optimal classifier
Naïve Bayes

Machine Learning – 10701/15781
Carlos Guestrin (presented by Eric Xing)
Carnegie Mellon University

September 17th, 2007

Machine learning for apartment hunting



- Now you've moved to Pittsburgh!!
  And you want to find the **most overall satisfying** apartment for you to **move in:**

  square-ft., # of bedroom, distance to campus, rent, …

Living area (ft^2)	# bedroom	Rent ($)	Yes/No
230	1	600	yes
506	2	1000	yes
433	2	1100	no
109	1	500	no
…			
150	1	500	?
270	1.5	1200	?

Classification

- **Learn**: h:**X** $\mapsto$ Y
  - **X** – features (Rent, Ames, dist; ...)
  - Y – target classes 1/0

- Suppose you know P(Y|**X**) exactly, how should you classify?
  - Bayes classifier: $\arg\max_y P(Y|X)$

$X_1\ X_2 \cdots X_n \mid Y=1,\ Y=0 \quad P(X|Y)$

$h(y|x) = P(X|y)P(Y).$

if $P(x|y=1)P(Y=1) \geq P(X|y=0)P(y=0)$. then $Y=1$.

- **Why?**

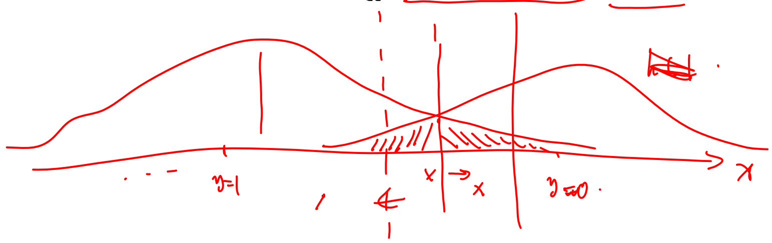
Optimal classification

- **Theorem:** Bayes classifier h_{Bayes} is optimal!

$P(x|y=1)P(y=1) = P(x|y=0)P(y=0).$ $\quad p(x|y)\,p(y).$

  - That is $error_{true}(h_{Bayes})) \leq error_{true}(h), \quad \forall h(\mathbf{x})$

- **Proof**: $p(error) = \int_x p(error|x)p(x)dx$



Bayes Rule

$$\underline{P(Y|X)} = \frac{P(X|Y)P(Y)}{P(X)} = \sum_{y} P(X|Y)\,P(Y)$$

Which is shorthand for:

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

How hard is it to learn the optimal classifier?

- Data =

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

- How do we represent these? How many parameters?
  - Prior, P(Y): $= \prod_k \theta_k^{\delta(y=k)} = \begin{cases} \theta_k & \text{if } y = k \end{cases}$
    - Suppose Y is composed of *k* classes

      for K outcome & need K-1 numbers
  - Likelihood, P(**X**|Y):
    - Suppose **X** is composed of *n* binary features

y.	f_1	f_2	- - - - -			f_6	
	+	+	+	+	+	+	π_1
1	+	-	+	+	+	+	
							0

- Complex model → High variance with limited data!!!

Conditional Independence

- 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

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

- e.g., $P(Thunder | Rain, Lightning) = P(Thunder | Lightning)$

- Equivalent to:

$$P(X, Y \mid Z) = P(X \mid Z) P(Y \mid Z)$$

What if features are independent?

- Predict 10701Grade
- From two **conditionally Independent** features
  - HomeworkGrade X_1
  - ClassAttendance X_2

$P(y)\, P(X | y)$

$\downarrow$

$\cancel{P(X|y)}$ $P(x_1\, x_2 | y) \leftarrow$

$= P(x_1 | y) P(x_2 | y)$

The Naïve Bayes assumption

- Naïve Bayes assumption:
  - Features are independent given class:

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

  - More generally:

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

2^7

$\downarrow$

12

- How many parameters now?
  - Suppose **X** is composed of *n* binary features

The Naïve Bayes Classifier

- Given:
  - Prior P(Y)
  - *n* conditionally independent features **X** given the class Y
  - For each X_i, we have likelihood $P(X_i|Y)$
- Decision rule:

$$
\begin{aligned}
y^* = h_{NB}(\mathbf{x}) &= \arg\max_y P(y)P(x_1, \ldots, x_n \mid y) \\
&= \arg\max_y P(y) \prod_i P(x_i|y)
\end{aligned}
$$

- If assumption holds, NB is optimal classifier!

MLE for the parameters of NB

- Given dataset
  - Count(A=a,B=b) ← number of examples where A=a and B=b
- MLE for NB, simply:
  - Prior: P(Y=y) = $\frac{\#(Y=1)}{\sum_k \#(Y=k)}$
  - Likelihood: $P(X_i=x_i|Y_i=y_i)$ = $\frac{\#(X=x_i, Y_i=y_i)}{\#(Y_i=y_i)}$ $\times P(y)$ $P(X, y)$

$\sum_k \#(X_i=k, Y_i=y_i)$

Subtleties of NB classifier 1 – Violating the NB assumption

- Usually, features are not conditionally independent:

$$\boxed{P(X_1 ... X_n|Y) \neq \prod_i P(X_i|Y)}$$

- Actual probabilities P(Y|**X**) often biased towards 0 or 1
- Nonetheless, NB is the single most used classifier out there
  - NB often performs well, even when assumption is violated
  - [Domingos & Pazzani '96] discuss some conditions for good performance

Subtleties of NB classifier 2 – Insufficient training data

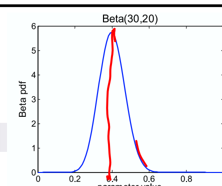
- What if you never see a training instance where X_1=a when Y=b?
  - e.g., Y={SpamEmail}, X_1={'Enlargement'}
  - $P(X_1=a \mid Y=b) = 0$
- Thus, no matter what the values $X_2,\ldots,X_n$ take:
  - $P(Y=b \mid X_1=a, X_2,\ldots,X_n) = 0$

$\propto P(Y=b)\,P(x \ldots x_n \mid Y=b)$

$= P(Y=b)\prod_i P(X_i \mid Y=b) = 0$

- What now???

MAP for Beta distribution



$$P(\theta \mid \mathcal{D}) = \frac{\theta^{\beta_H+\alpha_H-1}(1-\theta)^{\beta_T+\alpha_T-1}}{B(\beta_H+\alpha_H, \beta_T+\alpha_T)} \sim Beta(\beta_H+\alpha_H, \beta_T+\alpha_T)$$

$\alpha_1 = \#(y=1)$ $\quad \alpha_2 = \#(y=0)$

$P(y) = \begin{cases}\theta \\ 1-\theta\end{cases} = \theta^{\delta(y=1)}(1-\theta)^{1-\delta(y=1)}$

- MAP: use most likely parameter:

$$\widehat{\theta} = \arg\max_\theta P(\theta \mid \mathcal{D}) = \frac{\alpha_1+\beta_1-1}{\alpha_1+\beta_1-1+\alpha_0+\beta_0-1}$$

$P(\theta) \propto \theta^{\beta_1-1}(1-\theta)^{\beta_0-1}$

$\frac{\beta_0}{\beta_1+\beta_0}$

$\theta_{ML} = \frac{\alpha_1}{\alpha_1+\alpha_0} \xleftarrow{N\to\infty}$

$m = \beta_1 + \beta_2$

- Beta prior equivalent to extra thumbtack flips
- As $N \to \infty$, prior is "forgotten"
- **But, for small sample size, prior is important!**

Bayesian learning for NB parameters – a.k.a. smoothing

- Dataset of *N* examples
- Prior
  - "distribution" $Q(X_i,Y)$, $Q(Y)$
  - *m* "virtual" examples

$m_{i/y}$

$p(\theta_i^{k/Y_j}) = Beta(\alpha_i^{k/Y}, \alpha_i^{k/Y})$

$P(\theta) = Beta(\beta_1, \beta_2)$

$\beta_1 + \beta_2 = m$

$\alpha_i^{1/y} + \alpha_i^{0/y} = m_{i/y}$

- MAP estimate
  - $P(X_i|Y_{=1}) = \dfrac{(\#X_i = x_i, y_i = 1) + \alpha_i^{1/Y=1}}{\#(Y=1) + m_{i/y=1}}$

$P(y) = \dfrac{\#(y=1) + \beta_1}{N + m}$

$P(X_i|y)P(y)$

- Now, even if you never observe a feature/class, posterior probability never zero

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 i^{th} 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 decidec
```

NB for Text classification

- P(**X**|Y) is huge!!!
  - Article at least 1000 words, $\mathbf{X}=\{X_1,\ldots,X_{1000}\}$ $\in$ 10ww 10w
  - X_i represents i^{th} word in document, i.e., the domain of X_i is entire vocabulary, e.g., Webster Dictionary (or more), 10,000 words, etc.

$P(X_i|Y) =$

- NB assumption helps a lot!!!
  - $P(X_i=x_i|Y=y)$ is just the probability of observing word x_i in a document on topic y

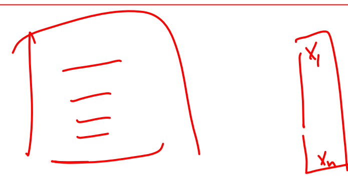
$$h_{NB}(\mathbf{x}) = \arg\max_y P(y) \prod_{i=1}^{LengthDoc} P(x_i|y)$$

Bag of words model

- Typical additional assumption – **Position in document doesn't matter**: $P(X_i=x_i|Y=y) = P(X_k=x_i|Y=y)$
  - "Bag of words" model – order of words on the page ignored
  - Sounds really silly, but often works very well!

$$P(y) \prod_{i=1}^{LengthDoc} P(x_i|y)$$

When the lecture is over, remember to wake up the person sitting next to you in the lecture room.



Bag of words model

- Typical additional assumption – **Position in document doesn't matter**: $P(X_i=x_i|Y=y) = P(X_k=x_i|Y=y)$
  - "Bag of words" model – order of words on the page ignored
  - Sounds really silly, but often works very well!

$$P(y) \prod_{i=1}^{LengthDoc} P(x_i|y)$$

in is lecture lecture next over person remember room sitting the the the to to up wake when you

Bag of Words Approach

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

NB with Bag of Words for text classification

- Learning phase:
  - Prior P(Y)
    - Count how many documents you have from each topic (+ prior)
  - $P(X_i|Y)$
    - For each topic, count how many times you saw word in documents of this topic (+ prior)

Smoothing

- Test phase:
  - For each document
    - Use naïve Bayes decision rule

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

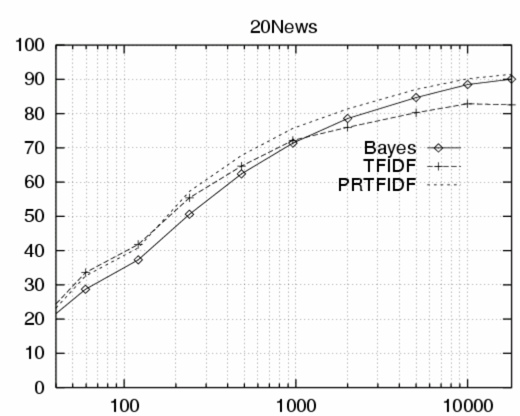
Twenty News Groups results

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

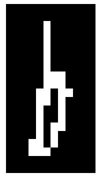
Learning curve for Twenty News Groups

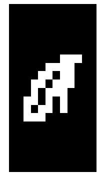


Accuracy vs. Training set size (1/3 withheld for test)

What if we have continuous X_i ?

Eg., character recognition: X_i is i^{th} pixel



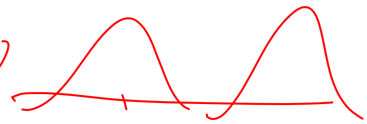


Gaussian Naïve Bayes (GNB):

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

Sometimes assume variance

- is independent of Y (i.e., σ_i),
- or independent of X_i (i.e., σ_k)
- or both (i.e., σ)



Estimating Parameters: Y discrete, X_i continuous

Maximum likelihood estimates:

jth training example

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

δ(x)=1 if x true, else 0

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

Example: GNB for classifying mental states

[Mitchell et al.]

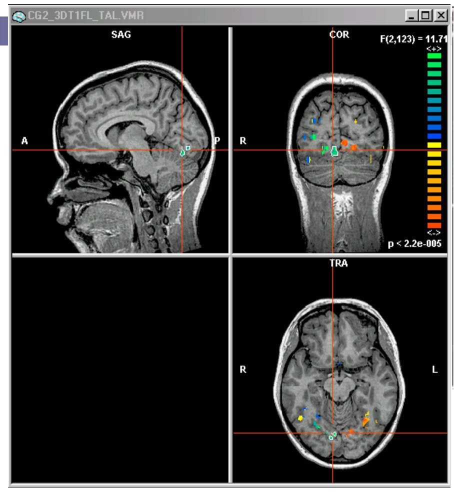
~1 mm resolution

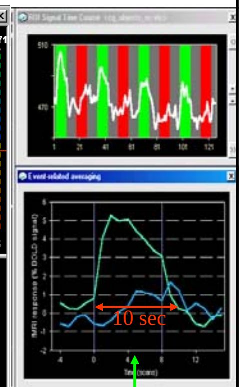
~2 images per sec.

15,000 voxels/image

non-invasive, safe

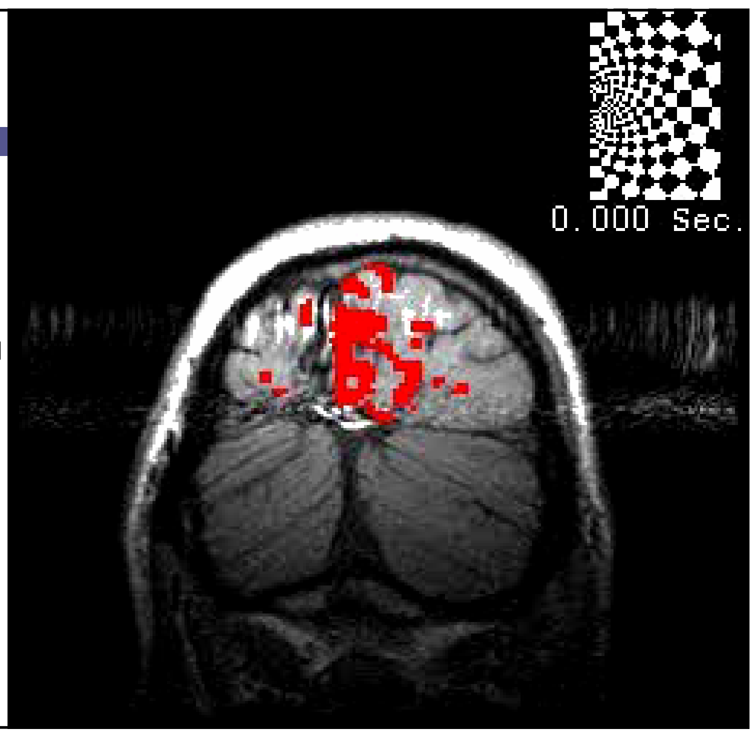
measures Blood Oxygen Level Dependent (BOLD) response





Typical impulse response

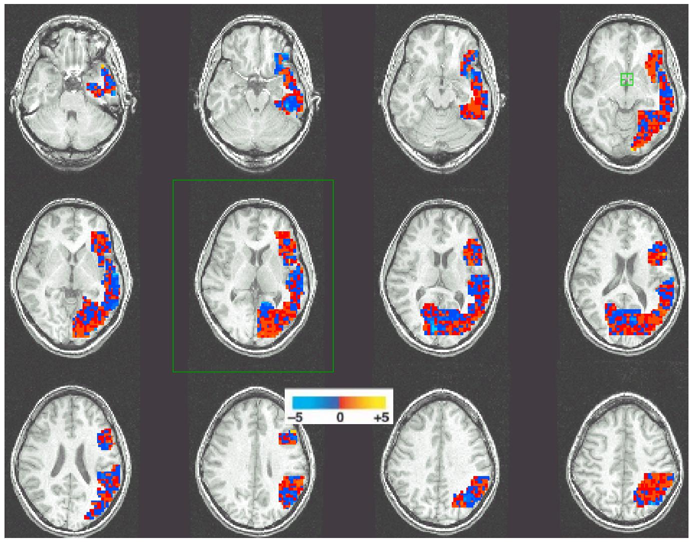
Brain scans can track activation with precision and sensitivity



[Mitchell et al.]

Gaussian Naïve Bayes: Learned $\mu_{voxel,word}$ P(BrainActivity | WordCategory = {People,Animal})

[Mitchell et al.]

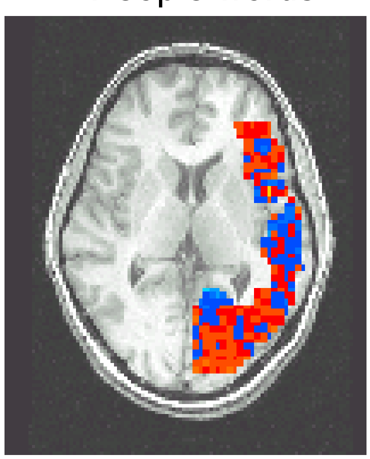


Learned Bayes Models – Means for P(BrainActivity | WordCategory)

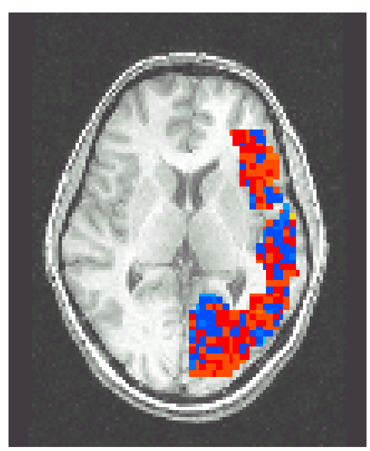
[Mitchell et al.]

Pairwise classification accuracy: 85%

People words



Animal words



What you need to know about Naïve Bayes

- Optimal decision using Bayes Classifier
- Naïve Bayes classifier
  - What's the assumption
  - Why we use it
  - How do we learn it
  - Why is Bayesian estimation important
- Text classification
  - Bag of words model
- Gaussian NB
  - Features are still conditionally independent
  - Each feature has a Gaussian distribution given class