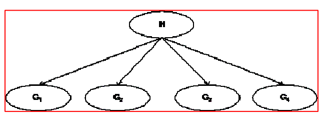
Machine Learning

10-701/15-781, Spring 2008

Naïve Bayes Classifier

Eric Xing

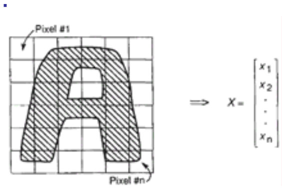
Lecture 3, January 23, 2006



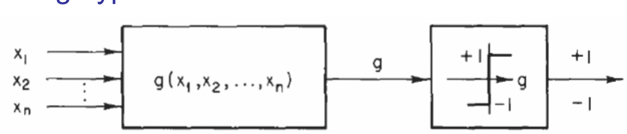
Reading: Chap. 4 CB and handouts

Classification

- Representing data:



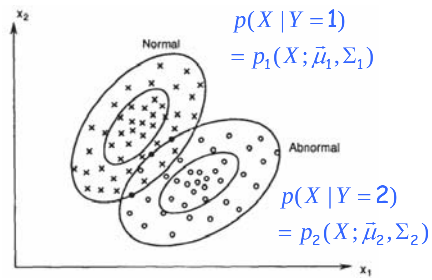
- Choosing hypothesis class



- Learning: h:**X** $\mapsto$ Y
  - **X** – features
  - Y – target classes

Suppose you know the following …

- Classification-specific Dist.: P(X|Y)



Bayes classifier:

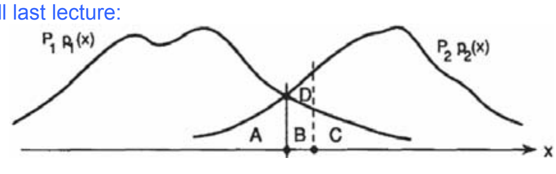
- Class prior (i.e., "weight"): P(Y)
- This is a generative model of the data!

Optimal classification

- **Theorem:** Bayes classifier is optimal!
  - That is

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

- **Proof**:
  - Recall last lecture:



- How to learn a Bayes classifier?

Recall 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:

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

Recall Bayes Rule

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

Which is shorthand for:

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

Common abbreviation:

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

Learning Bayes Classifier

- Training data:

$$\overbrace{\text{Sky Temp Humid Wind Water Forecst}}^{X}\ \overbrace{\text{EnjoySpt}}^{Y}$$

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

- Learning = estimating P(X|Y), and P(Y)

- Classification = using Bayes rule to calculate $P(Y \mid X_{new})$

How hard is it to learn the optimal classifier?

- How do we represent these? How many parameters?
  - Prior, P(Y):
    - Suppose Y is composed of *k* classes

$$\overbrace{\text{Sky Temp Humid Wind Water Forecst}}^{X}\ \overbrace{\text{EnjoySpt}}^{Y}$$

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

  - Likelihood, P(**X**|Y):
    - Suppose **X** is composed of *n* binary features

- Complex model $\rightarrow$ High variance with limited data!!!

Naïve Bayes:

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

assuming that X_i and X_j are conditionally independent given Y, for all i≠j

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

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

Which we often write

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

- e.g.,

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

- Equivalent to:

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

Summary

- ***Bayes classifier*** is the best classifierwhich minimizes the probability of classification error.
- Nonparametric and parametric classifier
- A nonparametric classifier does not rely on any assumption concerning the structure of the underlying density function.
- A classifier becomes the ***Bayes classifier*** if the density estimates converge to the true densities
  - when an infinite number of samples are used
  - The resulting error is the ***Bayes error,*** the smallest achievable error given the underlying distributions.

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:

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

- 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:

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

- If assumption holds, NB is optimal classifier!

Naïve Bayes Algorithm

- 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})

$$Y^{new} \leftarrow \arg\max_{y_k} P(Y = y_k) \prod_i P(X_i = x_{ij}|Y = y_k)$$
$$Y^{new} \leftarrow \arg\max_{y_k} \pi_k \prod_i \theta_{ijk}$$

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

Learning NB: parameter estimation

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

$$\hat{\theta} = \arg\max_{\theta} P(\mathcal{D}|\theta)$$

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

$$\begin{aligned}\hat{\theta} &= \arg\max_{\theta} P(\theta|\mathcal{D}) \\ &= \arg\max_{\theta} \frac{P(\mathcal{D}|\theta)\mathcal{P}(\theta)}{P(\mathcal{D})}\end{aligned}$$

MLE for the parameters of NB

Discrete features:

- Maximum likelihood estimates (MLE's): $\hat{\theta} = \arg\max_{\theta} P(\mathcal{D}|\theta)$
- Given dataset
  - Count(A=a,B=b) $\leftarrow$ number of examples where A=a and B=b

Subtleties of NB classifier 1 – Violating the NB assumption

- Often the X_i are not really conditionally independent
- We use Naïve Bayes in many cases anyway, and it often works pretty well
  - often the right classification, even when not the right probability (see [Domingos&Pazzani, 1996])
  - But the resulting probabilities $P(Y|X_{new})$ are biased toward 1 or 0 (why?)

Subtleties of NB classifier 2 – Insufficient training data

- What if you never see a training instance where X_{1000}=a when Y=b?
  - e.g., Y={SpamEmail or not}, X_{1000}={'Rolex'}
  - $P(X_{1000}=T \mid Y=T) = 0$
- Thus, no matter what the values $X_2,\ldots,X_n$ take:
  - $P(Y=T \mid X_1,X_2,\ldots,X_{1000}=T, \ldots, X_n) = 0$
- What now???

MAP for the parameters of NB

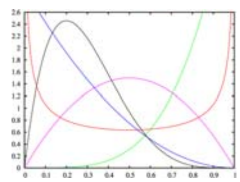
Discrete features:

- Maximum *a Posteriori* (MAP) estimate: (MAP's):

$$\hat{\theta} = \arg\max_{\theta} \frac{P(\mathcal{D}|\theta)\mathcal{P}(\theta)}{P(\mathcal{D})}$$

- Given prior:
  - Consider binary feature
  - θ is a *Bernoulli* rate

$$P(\theta;\alpha_T,\alpha_F) = \frac{\Gamma(\alpha_T+\alpha_F)}{\Gamma(\alpha_T)\Gamma(\alpha_F)}\theta^{\alpha_T-1}(1-\theta)^{\alpha_F-1} = \frac{\theta^{\alpha_T-1}(1-\theta)^{\alpha_F-1}}{B(\alpha_T,\alpha_F)}$$



  - Let β_a=Count(X=a) $\leftarrow$ number of examples where X=a

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

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

- **Posterior distribution of θ**
  - Bernoulli: $P(\theta \mid \mathcal{D}) = \frac{\theta^{\beta_T+\alpha_T-1}(1-\theta)^{\beta_F+\alpha_F-1}}{B(\beta_T+\alpha_T,\beta_F+\alpha_F)} \sim Beta(\beta_H+\alpha_H, \beta_T+\alpha_T)$
  - Multinomial $P(\theta \mid \mathcal{D}) = \frac{\prod_{j=1}^{K}\theta_j^{\beta_j+\alpha_j-1}}{B(\beta_1+\alpha_1,\ldots,\beta_K+\alpha_K)} \sim Dirichlet(\beta_1+\alpha_1,\ldots,\beta_K+\alpha_K)$
- MAP estimate:

$$\hat{\theta} = \arg\max_{\theta} P(\theta|\mathcal{D}) =$$

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

MAP for the parameters of NB

- Dataset of *N* examples
  - Let β_{iab}=Count(X_i=a,Y=b) $\leftarrow$ number of examples where X_i=a and Y=b
  - Let γ_b=Count(Y=b)
- Prior

  Q(X_i|Y) $\propto$ Multinomial(α_{i1}, …, α_{iK}) or Multinomial(α/K)

  Q(Y) $\propto$ Multinomial(τ_{i1}, …, τ_{iM}) or Multinomial(τ/M)

  m "virtual" examples
- MAP estimate

$$\hat{\pi}_k = \arg\max_{\pi_k} \prod_k P(Y = y_k; \pi_k) P(\pi_k | \vec{\tau}) = ?$$

$$\hat{\theta}_{ijk} = \arg\max_{\theta_{ijk}} \prod_j P(X_i = x_{ij} | Y = y_k; \theta_{ijk}) P(\theta_{ijk} | \vec{\alpha}_{ik}) = ?$$

  - 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, **X**=$\{X_1,\ldots,X_{1000}\}$
  - 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.
- 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 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)
- Test phase:
  - For each document
    - Use naïve Bayes decision rule

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

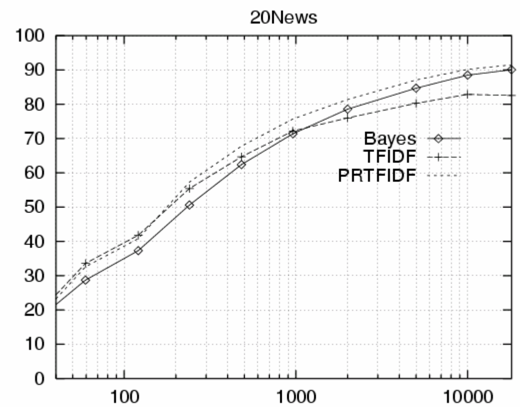
Twenty News Groups results

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

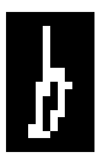
Learning curve for Twenty News Groups

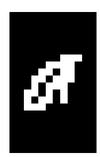


Accuracy vs. Training set size (1/3 withheld for test)

What if we have continuous X_i ?

- Eg., character recognition: X_i is ith pixel





- Gaussian Naïve Bayes (GNB):

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

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, Xi 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

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

Gaussian Naïve Bayes

Example: GNB for classifying mental states [Mitchell et al.]

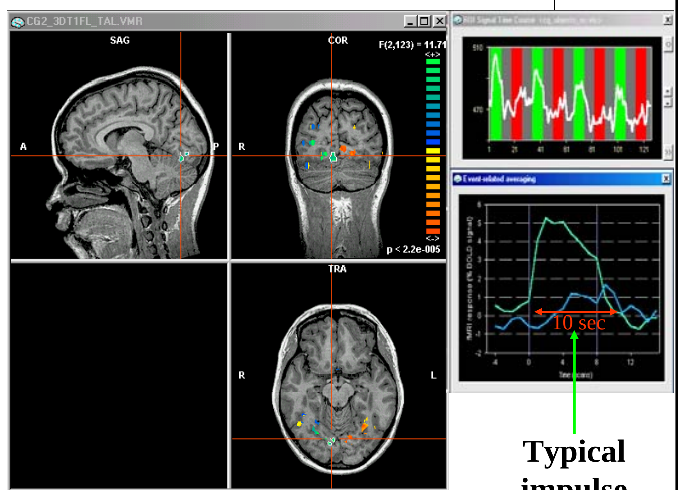
~1 mm resolution

~2 images per sec.

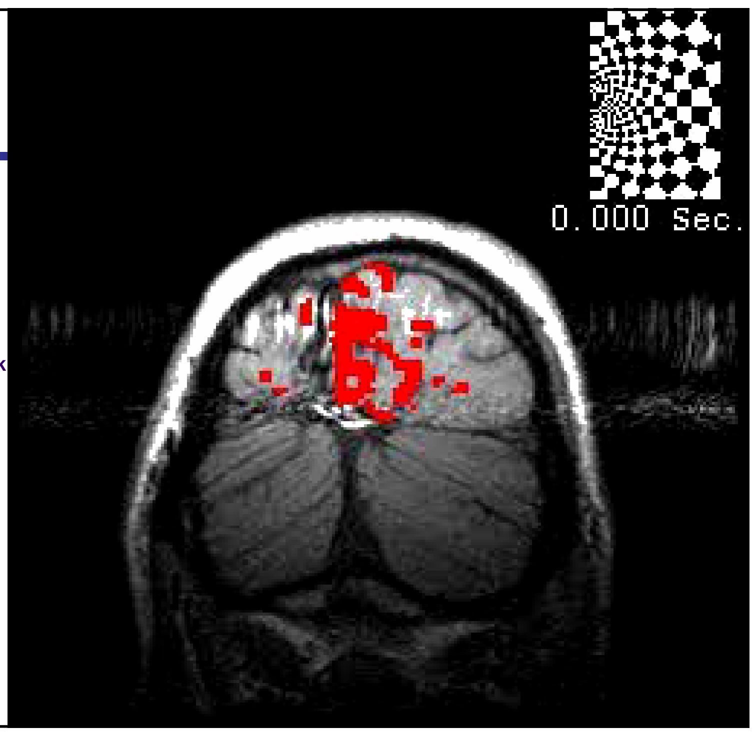
15,000 voxels/image

non-invasive, safe

measures Blood Oxygen Level Dependent (BOLD) response



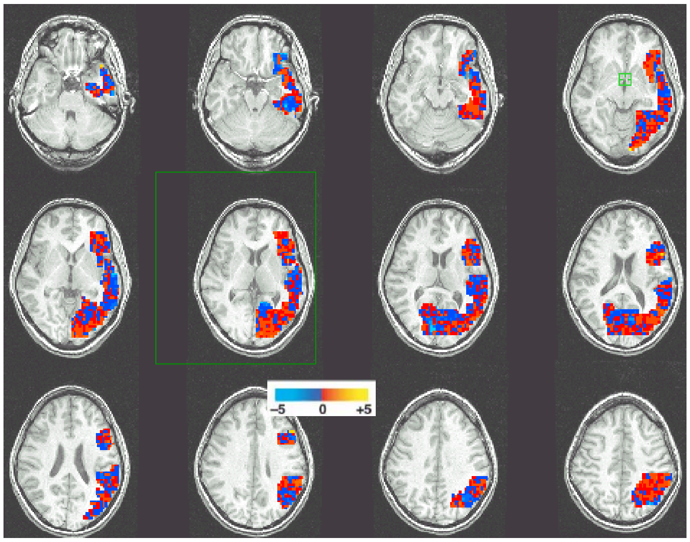
Typical impulse response



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.]

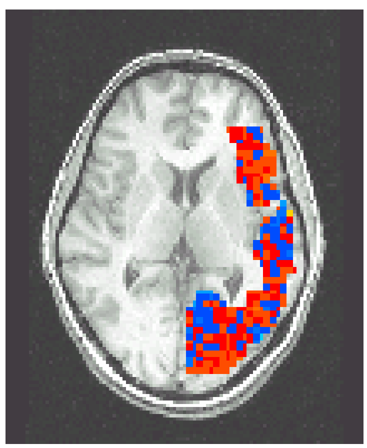


Learned Bayes Models – Means for P(BrainActivity | WordCategory) [Mitchell et al.]

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