Naïve Bayes (Continued)
Naïve Bayes with Continuous (variables)
Logistic Regression

Machine Learning – 10701/15781

Carlos Guestrin

Carnegie Mellon University

January 29th, 2007

Announcements

- ~~Recitations stay on Thursdays~~
  - ~~5-6:30pm in Wean 5409~~
  - ~~This week: Naïve Bayes & Logistic Regression~~
- ~~**Extension** for the first homework:~~
  - ~~**Due Wed. Feb 8th** beginning of class~~
  - ~~Mitchell's chapter is most useful reading~~
- ~~Go to the AI seminar:~~
  - ~~Tuesdays 3:30pm, Wean 5409~~
  - ~~http://www.cs.cmu.edu/~aiseminar/~~
  - ~~This week's seminar very relevant to what we are covering in class~~

ignore, unless you are taking the class last year...

Optimal classification $P(Y|X)$

- **Theorem:** Bayes classifier h_{Bayes} is optimal! if you know $P(Y|X)$ exactly

$h_{Bayes}(x) = y^*(x) \leftarrow \arg\max_y P(Y=y|X=x)$

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

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

$$\min_h p(error_h|x) = \begin{cases} P(Y=f|x), & h(x)=t \\ P(Y=t|x), & h(x)=f \end{cases}$$

$\equiv$ max prob get right $\equiv$ guess answer with highest prob $\equiv \arg\max_y P(Y=y|X=x)$

How hard is it to learn the optimal classifier?

X Y

- 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): $k-1$ ← probably ok
    - Suppose Y is composed of *k* classes
  - Likelihood, P(**X**|Y): $\boxed{k(2^n-1)}$ ← get you into trouble
    - Suppose **X** is composed of *n* binary features

$\#\text{param}(P(X|Y=y)) = 2^n - 1$ for each y

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

$T \perp R \mid L$ = Thunder independent of rain given lightning

- Equivalent to: $X \perp Y \mid Z$

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

The Naïve Bayes assumption

$Z \equiv m$ values

- Naïve Bayes assumption: $X_1 \perp X_2 \mid Y$
  - Features are independent given class:

likelihood

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

$P(Z) =$

Z	$P(Z)$
1	0.1
2	0.2
⋮	⋮
m	0.06

$\sum_j P(Z=j) = 1$

saves one param

  - More generally: $\forall i \ X_i \perp \{X_1 \ldots X_{i-1}, X_{i+1}, \ldots X_n\} \mid Y$ ← class variable

$$\boxed{P(X_1 \ldots X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)}$$

binary likelihood k values prod

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

full table $k \cdot (2^n - 1)$ parameters

with NB assumption $n \cdot k \cdot (2-1) = nk$

The Naïve Bayes Classifier

- Given: $\{Spam, \neg Spam\}$
  - Prior P(Y) 0.8 0.2
  - *n* conditionally independent features **X** given the class Y words images ⋮
  - For each X_i, we have likelihood $P(X_i|Y)$

compute for each feature

- Decision rule:

prior likelihood

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

- If assumption holds, NB is optimal classifier!

MLE for the parameters of NB

- Given dataset D
  - Count(A=a,B=b) ← number of examples where A=a and B=b

Proof path: write likelihood of data, take log, take derivative, set to zero. ⋮

- MLE for NB, simply:
  - Prior: P(Y=y) = $\frac{Count(Y=y)}{|D|}$
  - Likelihood: $P(X_i=x_i|Y_i=y_i)$ = $\frac{Count(X_i=x_i, Y_i=y_i)}{Count(Y_i=y_i)}$

Subtleties of NB classifier 1 – Violating the NB assumption

- Usually, features are not conditionally independent:

  ~~pretty~~ always the case

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

- Actual probabilities P(Y|**X**) often biased towards 0 or 1 see an example in homework!
- 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

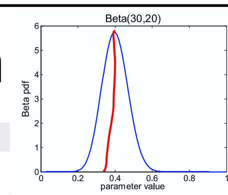
Subtleties of NB classifier 2 – Insufficient training data

- What if you never see a training instance where X_1=a when Y=b? $P(X_1=a|Y=b) = \emptyset$
  - 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:
  - $P(Y=b \mid X_1=a, X_2,\ldots,X_n) = 0$ $\cancel{=0}$

$$\propto P(Y=b) \cdot \cancel{P(X_1=a|Y=b)} \cdot \prod_{i=2}^{n} P(X_i|Y=b) = 0$$

- What now???

MAP for Beta distribution



multinomial-like

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

$\alpha_H = 3$
$\alpha_T = 2$
β_H, β_T extra data

- MAP: use most likely parameter:

$$\widehat{\theta} = \arg\max_{\theta} P(\theta \mid \mathcal{D}) = \frac{\beta_H + \alpha_H - 1}{\beta_H + \alpha_H + \beta_T + \alpha_T - 2}$$

from prior

- Beta prior equivalent to extra thumbtack flips
- As $N \rightarrow \infty$, 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 (assume Dirichlet prior) (generalization of Beta for Multinomials)
  - "distribution" $Q(X_i,Y)$, $Q(Y)$, typically $Q \sim$ uniform
  - m "virtual" examples, $m = 1, 2 \ldots 5 \ldots 10$

strength

- MAP estimate
  - $P(X_i = a | Y = b) = \dfrac{Count(X_i = a, Y = b) + m \cdot Q(X_i = a, Y = b)}{Count(Y = b) + m \cdot Q(Y = b)}$

e.g., if Q is uniform, $|Y| = k$, $|X_i| = \ell$

$Q(Y = b) = \frac{1}{k}$ $\quad Q(X_i = a, Y = b) = \frac{1}{k \cdot \ell}$

- 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
```

ith word

$X_i = \{\text{Kings}\}$

$|X_i| = 10{,}000$

$\hookrightarrow 100{,}000$

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.

explicitly $\#_{params}(P(X|Y)) = K\,(10{,}000^{1000} - 1)$ (K ← classes)

- 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

$1000 \cdot (10{,}000 - 1) \cdot K$

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

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 (+ smoothing 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

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

Twenty News Groups results

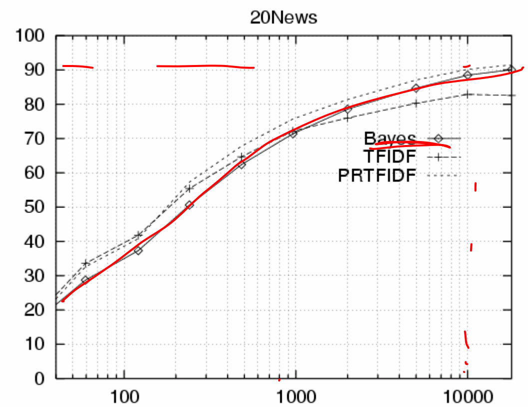
Given 1000 training documents from each group
Learn to classify new documents according to which newsgroup it came from

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



amount

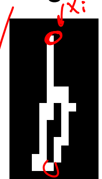
Accuracy vs. Training set size (1/3 withheld for test)

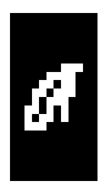
What if we have continuous X_i ?

Eg., character recognition: X_i is i[th] pixel

$Y \in \{a, b, \cdots, z\}$

x_i

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



$x_i \in \{0, \cdots, 256\}$

Gaussian Naïve Bayes (GNB):

$x_i \in (-\infty, +\infty)$

μ_{ik} = for feature i class k

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

feature   class k σ_{ik} = for features class k

Sometimes assume variance

- is independent of Y (i.e., σ_i), ← doesn't depend class
- or independent of X_i (i.e., σ_k) ← doesn't depend on feature
- or both (i.e., σ) ← nothing

Estimating Parameters: Y discrete, X_i continuous

Maximum likelihood estimates:

jth training example

$$\widehat{\mu}_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_{j=1}^{|D|} X_i^j \delta(Y^j = y_k)$$

ith feature
kth class

$\delta(x)$=1 if x true, else 0

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

Example: GNB for classifying mental states

[Mitchell et al.]

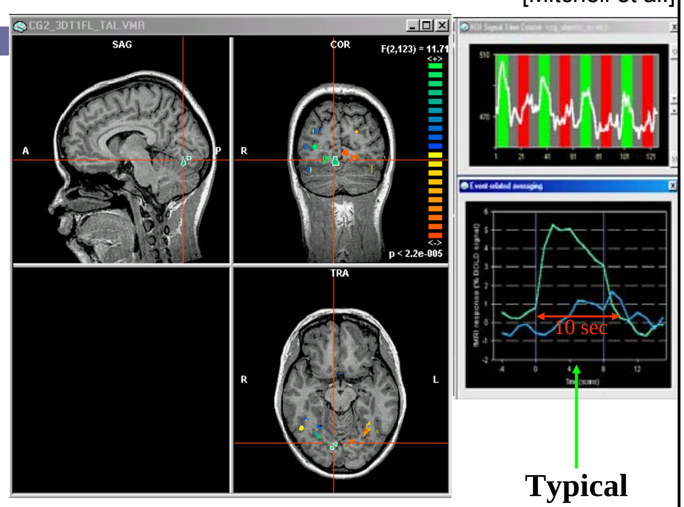
~1 mm resolution

~2 images per sec.

15,000 voxels/image

non-invasive, safe

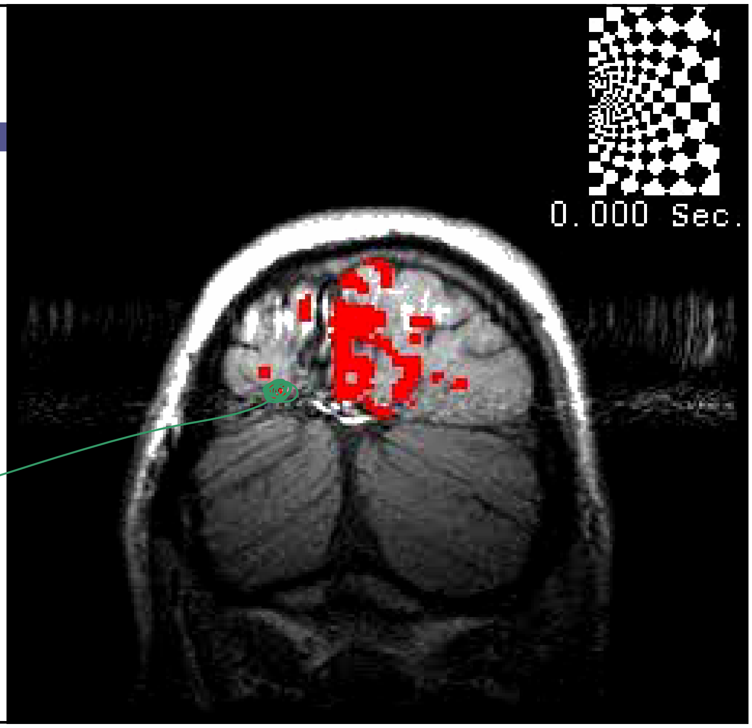
measures Blood Oxygen Level Dependent (BOLD) response



Typical impulse response

Brain scans can track activation with precision and sensitivity

each voxel is an X_i

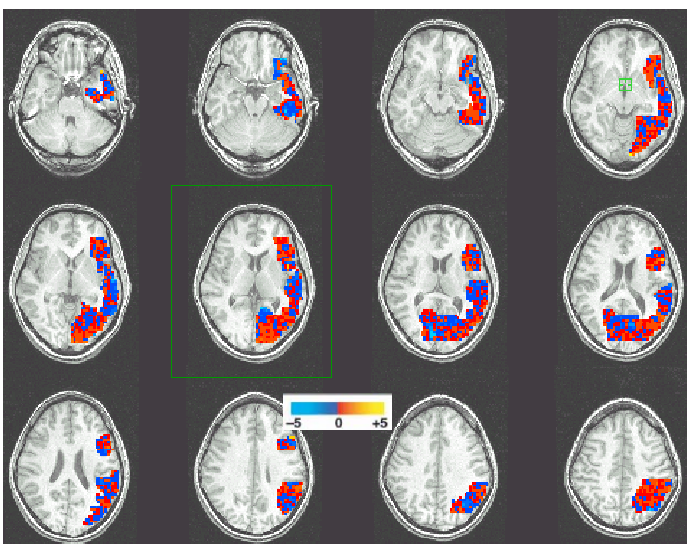


[Mitchell et al.]

Gaussian Naïve Bayes: Learned $\mu_{voxel,word}$

P(BrainActivity | WordCategory = {People,Animal})

[Mitchell et al.]



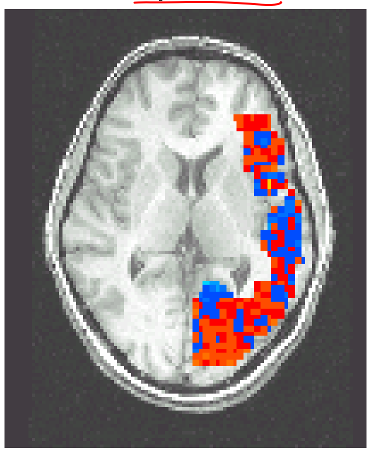
Learned Bayes Models – Means for P(BrainActivity | WordCategory)

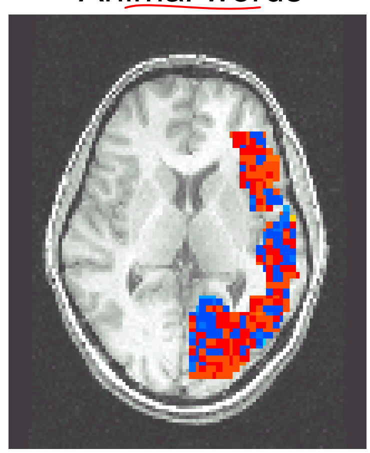
[Mitchell et al.]

Pairwise classification accuracy: 85%

People words

Animal words





What you need to know about Naïve Bayes

- Types of learning problems
  - Learning is (just) function approximation!
- Optimal decision using Bayes Classifier P(y|x)
- Naïve Bayes classifier P(Y), P(X|Y), use Bayes rule...
  - What's the assumption NB
  - 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

Generative v. Discriminative classifiers – Intuition

- **Want to Learn**: h:**X** ↦ Y
  - **X** – features
  - Y – target classes
- **Bayes optimal classifier** – P(Y|**X**)
- **Generative classifier**, e.g., Naïve Bayes:
  - Assume some **functional form for P(X|Y), P(Y)**
  - Estimate parameters of P(X|Y), P(Y) directly from training data
  - Use Bayes rule to calculate P(Y|X= x)
  - This is a '***generative*' model**
    - **Indirect** computation of P(Y|X) through Bayes rule
    - But, **can generate a sample of the data**, $P(X) = \sum_y P(y) P(X|y)$
- **Discriminative classifiers**, e.g., Logistic Regression:
  - Assume some **functional form for P(Y|X)**
  - Estimate parameters of P(Y|X) directly from training data
  - This is the '***discriminative*' model**
    - Directly learn P(Y|X)
    - But **cannot obtain a sample of the data**, because P(X) is not available

if you have

$h_{Bayes}(x) = \arg\max_y P(y|x)$

NB

table

P(Y)

P(X|Y)

sample image

first sample class

then sample pixel values

$\propto P(Y) \cdot P(X=x|Y)$

if you want to classify

have use

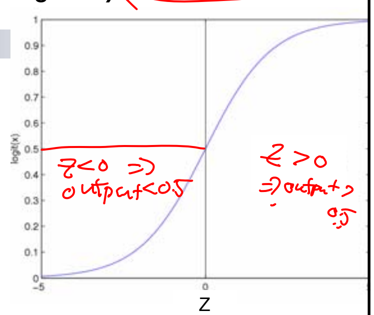
directly model P(Y|X)

discriminate classes, e.g.,

person v. animal

Logistic Regression

Logistic function (or Sigmoid): $\frac{1}{1+exp(-z)}$



- Learn P(Y|**X**) directly!
  - Assume a particular functional form
  - Sigmoid applied to a linear function of the data:

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

linear function of X, with parameters w

$P(Y=0|X) = 1 - P(Y=1|X)$

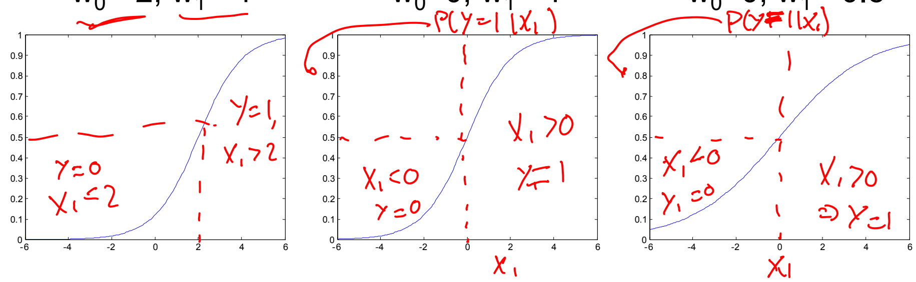
Features can be discrete or continuous!

Understanding the sigmoid

$$g(w_0 + \sum_i w_i x_i) = \frac{1}{1 + e^{w_0 + \sum_i w_i x_i}}$$

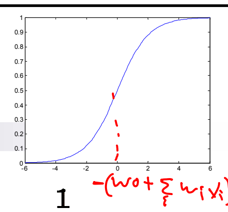
one feature X_1:

w_0=-2, w_1=-1 $\quad$ w_0=0, w_1=-1 $\quad$ w_0=0, w_1=-0.5

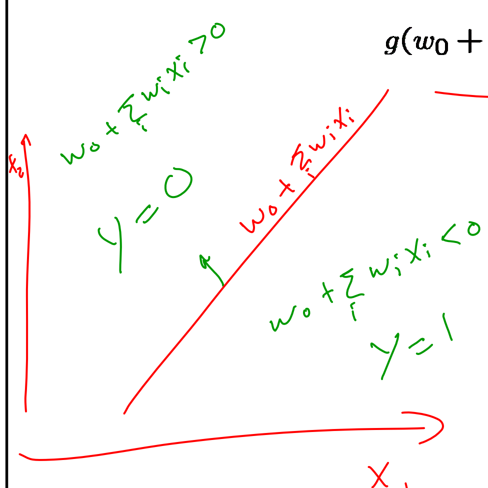


 31

Logistic Regression – a Linear classifier



$$g(w_0 + \sum_i w_i x_i) = \frac{1}{1 + e^{w_0 + \sum_i w_i x_i}}$$



$P(Y=1|X) > 0.5$

when

$-(w_0+\sum_i w_i x_i) > 0$

$\Leftrightarrow w_0+\sum_i w_i x_i < 0$

$P(Y=0|X) > 0.5$

$\cdot\ w_0+\sum_i w_i x_i > 0$

 32

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

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

Logistic regression v. Naïve Bayes

- Consider learning f: X → Y, where
  - X is a vector of real-valued features, < X1 … Xn >
  - Y is boolean
- Could use a Gaussian Naïve Bayes classifier
  - 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(θ,1-θ)
- What does that imply about the form of P(Y|X)?

Logistic regression v. Naïve Bayes

- Consider learning f: X → Y, where
  - X is a vector of real-valued features, < X1 … Xn >
  - Y is boolean
- Could use a Gaussian Naïve Bayes classifier
  - 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(θ,1-θ)
- What does that imply about the form of P(Y|X)?

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

Cool!!!!

Derive form for P(Y|X) for continuous X_i

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

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

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

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

Ratio of class-conditional probabilities

$$\ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)}$$

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

Derive form for P(Y|X) for continuous X_i

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

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

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

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

Gaussian Naïve Bayes v. Logistic Regression

Set of Gaussian Naïve Bayes parameters (feature variance independent of class label)

Set of Logistic Regression parameters

- Representation equivalence
  - **But only in a special case!!!** (GNB with class-independent variances)
- But what's the difference???
- **LR makes no assumptions about** P(**X**|Y) **in learning**!!!
- **Loss function!!!**
  - Optimize different functions → Obtain different solutions

Logistic regression for more than 2 classes

- Logistic regression in more general case, where $Y \in \{Y_1 \ldots Y_R\}$: learn R-1 sets of weights

Logistic regression more generally

- Logistic regression in more general case, where $Y \in \{Y_1 ... Y_R\}$: learn R-1 sets of weights

  for $k<R$

$$P(Y = y_k|X) = \frac{\exp(w_{k0} + \sum_{i=1}^{n} w_{ki}X_i)}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^{n} w_{ji}X_i)}$$

  for $k=R$ (normalization, so no weights for this class)

$$P(Y = y_R|X) = \frac{1}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^{n} w_{ji}X_i)}$$

Features can be discrete or continuous!

Loss functions: Likelihood v. Conditional Likelihood

- Generative (Naïve Bayes) Loss function:

  Data likelihood

$$\begin{aligned} \ln P(\mathcal{D} \mid \mathbf{w}) &= \sum_{j=1}^{N} \ln P(\mathbf{x}^j, y^j \mid \mathbf{w}) \\ &= \sum_{j=1}^{N} \ln P(y^j \mid \mathbf{x}^j, \mathbf{w}) + \sum_{j=1}^{N} \ln P(\mathbf{x}^j \mid \mathbf{w}) \end{aligned}$$

- Discriminative models cannot compute P($\mathbf{x}^j$|**w**)!
- But, discriminative (logistic regression) loss function:

  Conditional Data Likelihood

$$\ln P(\mathcal{D}_Y \mid \mathcal{D}_{\mathbf{X}}, \mathbf{w}) = \sum_{j=1}^{N} \ln P(y^j \mid \mathbf{x}^j, \mathbf{w})$$

  - Doesn't waste effort learning P(X) – focuses on P(Y|**X**) all that matters for classification

Expressing Conditional Log Likelihood

$$l(\mathbf{w}) \equiv \sum_j \ln P(y^j|\mathbf{x}^j, \mathbf{w})$$

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

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

$$l(\mathbf{w}) = \sum_j y^j \ln P(y^j=1|\mathbf{x}^j,\mathbf{w}) + (1-y^j)\ln P(y^j=0|\mathbf{x}^j,\mathbf{w})$$

Maximizing Conditional Log Likelihood

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

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

$$\begin{aligned} l(\mathbf{w}) &\equiv \ln \prod_j P(y^j|\mathbf{x}^j,\mathbf{w}) \\ &= \sum_j y^j(w_0 + \sum_i^n w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^n w_i x_i^j)) \end{aligned}$$

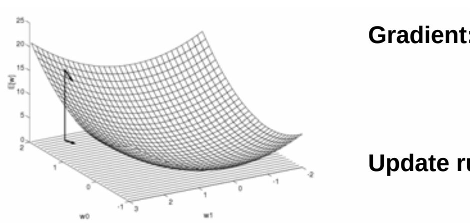
Good news: $l(\mathbf{w})$ is concave function of $\mathbf{w}$ → no locally optimal solutions

Bad news: no closed-form solution to maximize $l(\mathbf{w})$

Good news: concave functions easy to optimize

Optimizing concave function – Gradient ascent

- Conditional likelihood for Logistic Regression is concave $\rightarrow$ Find optimum with gradient ascent



Gradient: $\nabla_{\mathbf{w}} l(\mathbf{w}) = [\frac{\partial l(\mathbf{w})}{\partial w_0}, \ldots, \frac{\partial l(\mathbf{w})}{\partial w_n}]'$

Learning rate, η>0

Update rule: $\Delta \mathbf{w} = \eta \nabla_{\mathbf{w}} l(\mathbf{w})$

$$w_i \leftarrow w_i + \eta \frac{\partial l(\mathbf{w})}{\partial w_i}$$

- Gradient ascent is simplest of optimization approaches
  - e.g., Conjugate gradient ascent much better (see reading)

Maximize Conditional Log Likelihood: Gradient ascent

$$l(\mathbf{w}) \quad = \quad \sum_j y^j (w_0 + \sum_i^n w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^n w_i x_i^j))$$

Gradient ascent algorithm: iterate until change < ε

For all i, $\quad w_i \leftarrow w_i + \eta \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w})]$

repeat

That's all M(C)LE. How about MAP?

$$p(\mathbf{w} \mid Y, \mathbf{X}) \;\propto\; P(Y \mid \mathbf{X}, \mathbf{w})p(\mathbf{w})$$

- One common approach is to define priors on **w**
  - Normal distribution, zero mean, identity covariance
  - "Pushes" parameters towards zero
- Corresponds to ***Regularization***
  - Helps avoid very large weights and overfitting
  - Explore this in your homework
  - More on this later in the semester
- MAP estimate

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \ln \left[p(\mathbf{w}) \prod_{j=1}^{N} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

Gradient of M(C)AP

$$\frac{\partial}{\partial w_i} \ln \left[p(\mathbf{w}) \prod_{j=1}^{N} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

$$p(\mathbf{w}) = \prod_i \frac{1}{\kappa\sqrt{2\pi}} \; e^{\frac{-w_i^2}{2\kappa^2}}$$

MLE vs MAP

- Maximum conditional likelihood estimate

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \ln \left[\prod_{j=1}^{N} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

$$w_i \leftarrow w_i + \eta \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w})]$$

- Maximum conditional a posteriori estimate

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \ln \left[p(\mathbf{w}) \prod_{j=1}^{N} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

$$w_i \leftarrow w_i + \eta \left\{ -\lambda w_i + \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w})] \right\}$$

What you should know about Logistic Regression (LR)

- Gaussian Naïve Bayes with class-independent variances representationally equivalent to LR
  - Solution differs because of objective (loss) function
- In general, NB and LR make different assumptions
  - NB: Features independent given class → assumption on P(**X**|Y)
  - LR: Functional form of P(Y|**X**), no assumption on P(**X**|Y)
- LR is a linear classifier
  - decision rule is a hyperplane
- LR optimized by conditional likelihood
  - no closed-form solution
  - concave → global optimum with gradient ascent
  - Maximum conditional a posteriori corresponds to regularization

Acknowledgements

- Some of the material is the presentation is courtesy of Tom Mitchell