CSE 446
Gaussian Naïve Bayes & Logistic Regression
Winter 2012

Dan Weld

Some slides from Carlos Guestrin, Luke Zettlemoyer

Last Time

- Learning Gaussians
- Naïve Bayes

Today

- Gaussians Naïve Bayes
- Logistic Regression

2

Text Classification
Bag of Words Representation

→

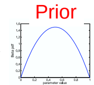
aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
…	
Zaire	0

Bayesian Learning

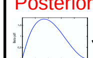
Use Bayes rule:

Data Likelihood

Prior



Posterior



$$P(Y \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid Y)\, P(Y)}{P(\mathbf{X})}$$

Normalization

Or equivalently: $P(Y \mid \mathbf{X}) \propto P(\mathbf{X} \mid Y)\, P(Y)$

Naïve Bayes

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

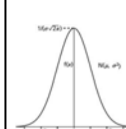
$$P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y)$$
$$= P(X_1|Y)P(X_2|Y)$$

  - More generally:

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

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

The Distributions We Love

	Discrete		Continuous
	Binary {0, 1}	k Values	
Single Event	Bernouilli		
Sequence (N trials) $N= \alpha_H+\alpha_T$	$P(\mathcal{D} \mid \theta) = \theta^{\alpha_H}(1-\theta)^{\alpha_T}$ Binomial ↓	Multinomial ↓	
Conjugate Prior	Beta	Dirichlet	

6

NB with Bag of Words for Text Classification

- Learning phase:
  - Prior $P(Y_m)$
    - Count how many documents from topic m / total # docs
  - $P(X_i|Y_m)$
    - Let B_m be a bag of words formed from all the docs in topic m
    - Let #(i, B) be the number of times word i is in bag B
    - $P(X_i \mid Y_m) = (\#(i, B_m)+1) / (W+\Sigma_j\#(j, B_m))$ where W=#unique words
- Test phase:
  - For each document
    - Use naïve Bayes decision rule

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

Easy to Implement

- But…
- If you do… it probably won't work…

8

Probabilities: Important Detail!

- $P(\text{spam} \mid X_1 \ldots X_n) = \prod_i P(\text{spam} \mid X_i)$

Any more potential problems here?

- We are multiplying lots of small numbers
  Danger of underflow!
  - $0.5^{57} = 7$ E -18
- Solution? Use logs and add!
  - $p_1 * p_2 = e^{\log(p1)+\log(p2)}$
  - Always keep in log form

Naïve Bayes Posterior Probabilities

- Classification results of naïve Bayes
  - I.e. the class with maximum posterior probability…
  - Usually fairly accurate (?!?!?)
- However, due to the inadequacy of the conditional independence assumption…
  - Actual posterior-***probability*** estimates ***not*** accurate.
  - Output probabilities generally very close to 0 or 1.

10

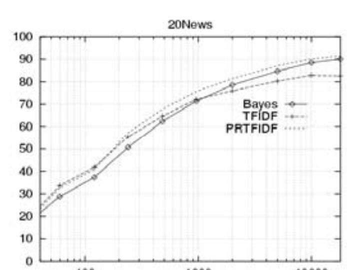
Twenty News Groups results

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

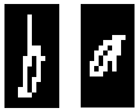
Learning curve for Twenty News Groups



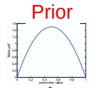
Accuracy vs. Training set size (1/3 withheld for test)

Bayesian Learning
What if Features are Continuous?

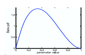
Eg., Character Recognition: X_i is i^{th} pixel



Prior



Posterior

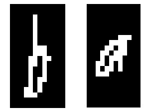


$P(Y \mid \mathbf{X}) \propto P(\mathbf{X} \mid Y)\, P(Y)$

Data Likelihood

Bayesian Learning
What if Features are Continuous?

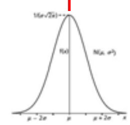
Eg., Character Recognition: X_i is i^{th} pixel



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

$P(X_i = x \mid Y = y_k) = N(\mu_{ik}, \sigma_{ik})$

$$N(\mu_{ik}, \sigma_{ik}) = \frac{1}{\sigma_{ik}\sqrt{2\pi}}\, e^{\frac{-(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$



Gaussian Naïve Bayes

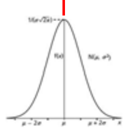
Sometimes Assume Variance

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

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

$P(X_i = x \mid Y = y_k) = N(\mu_{ik}, \sigma_{ik})$

$$N(\mu_{ik}, \sigma_{ik}) = \frac{1}{\sigma_{ik}\sqrt{2\pi}}\, e^{\frac{-(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$



Learning Gaussian Parameters

Maximum Likelihood Estimates:

- Mean:

$$\hat{\mu}_{MLE} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

- Variance:

$$\hat{\sigma}^2_{MLE} = \frac{1}{N}\sum_{i=1}^{N} (x_i - \hat{\mu})^2$$

Learning Gaussian Parameters

Maximum Likelihood Estimates:

- Mean:

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

j^{th} training example

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

- Variance:

$$\hat{\sigma}^2_{MLE} = \frac{1}{N}\sum_{i=1}^{N} (x_i - \hat{\mu})^2$$

Learning Gaussian Parameters

Maximum Likelihood Estimates:

- Mean:

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

- Variance:

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

Example: GNB for classifying mental states

[Mitchell et al.]

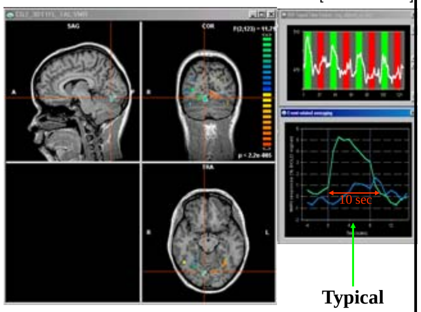
~1 mm resolution

~2 images per sec.

15,000 voxels/image

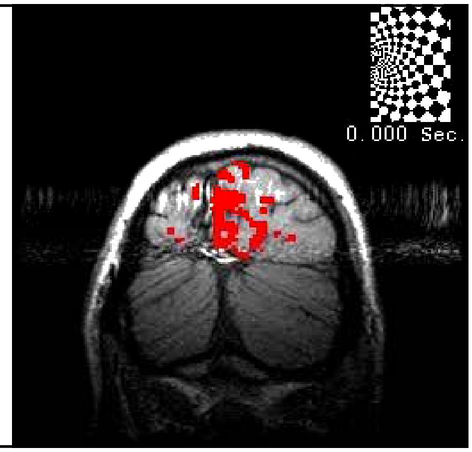
non-invasive, safe

measures Blood Oxygen Level Dependent (BOLD) response



Typical impulse response

Brain scans can track activation with precision and sensitivity

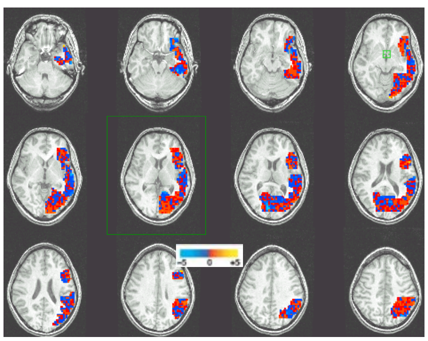


[Mitchell et al.]

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

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

[Mitchell et al.]

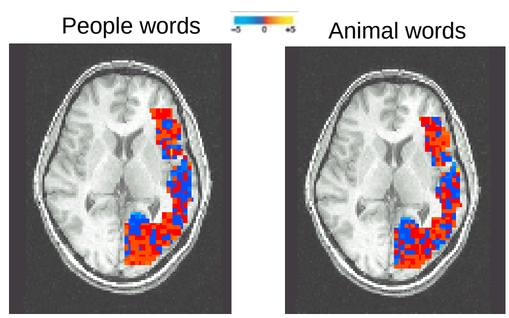


Gaussian Naïve Bayes: Learned $\mu_{voxel,word}$

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

[Mitchell et al.]

Pairwise classification accuracy: 85%



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
- Text Classification
  - Bag of words model
- Gaussian NB
  - Features still conditionally independent
  - Features have Gaussian distribution given class

What's (supervised) learning more formally

- Given:
  - **Dataset**: Instances $\{\langle \mathbf{x}_1;t(\mathbf{x}_1)\rangle,\ldots,\langle \mathbf{x}_N;t(\mathbf{x}_N)\rangle\}$
    - e.g., $\langle \mathbf{x}_i;t(\mathbf{x}_i)\rangle = \langle$(GPA=3.9,IQ=120,MLscore=99);150K$\rangle$
  - **Hypothesis space**: *H*
    - e.g., polynomials of degree 8
  - **Loss function**: measures quality of hypothesis $h \in H$
    - e.g., squared error for regression
- Obtain:
  - **Learning algorithm**: obtain $h \in H$ that minimizes loss function
    - e.g., using closed form solution if available
    - Or greedy search if not
    - Want to minimize prediction error, but can only minimize error in dataset

24

Types of (supervised) learning problems, revisited

- **Decision Trees**, e.g.,
  - **dataset**: ⟨votes; party⟩
  - **hypothesis space**:
  - **Loss function**:
- **NB Classification**, e.g.,
  - **dataset**: ⟨brain image; {verb v. noun}⟩
  - **hypothesis space**:
  - **Loss function**:
- **Density estimation**, e.g.,
  - **dataset**: ⟨grades⟩
  - **hypothesis space**:
  - **Loss function**:

Learning is (simply) function approximation!

- The general (supervised) learning problem:
  - Given some data (including features), hypothesis space, loss function
  - Learning is no magic!
  - Simply trying to find a function that fits the data
- **Regression**
- **Density estimation**
- **Classification**
- (Not surprisingly) Seemly different problem, very similar solutions…

What you need to know about **supervised learning**

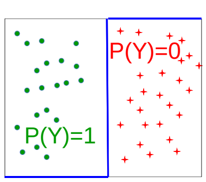
- Learning is function approximation
- What functions are being optimized?

Generative *vs.* **Discriminative** Classifiers

- **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
    - As a result, **can also 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

Logistic Regression

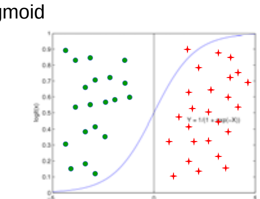
Learn P(Y|**X**) directly!
- Assume a particular functional form
- ☹ ***Not differentiable…***



Logistic Regression

Learn P(Y|**X**) directly!
- Assume a particular functional form
- Logistic Function
  - Aka Sigmoid

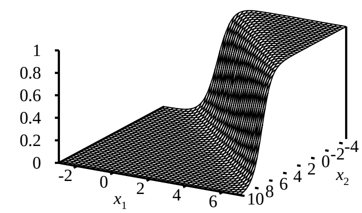
$$\frac{1}{1 + exp(-z)}$$



Logistic Function in n Dimensions

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

Sigmoid applied to a linear function of the data:

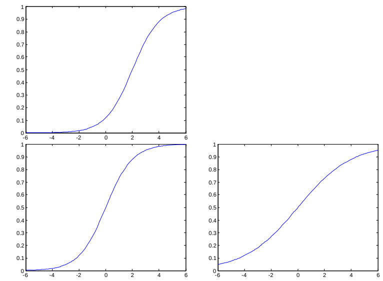


Features can be discrete or continuous!

Understanding Sigmoids

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

w_0=-2, w_1=-1



w_0=0, w_1=-1

w_0=0, w_1= -0.5

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

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

Loss functions: Likelihood vs. Conditional Likelihood

- Generative (Naïve Bayes) Loss function:
  Data likelihood

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

- 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

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

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

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

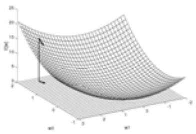
Good news: l(**w**) is concave function of **w** ! no locally optimal solutions

Bad news: no closed-form solution to maximize l(**w**)

Good news: concave functions easy to optimize

Optimizing concave function – Gradient ascent

- Conditional likelihood for Logistic Regression is concave ! 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: $\Delta \mathbf{w} = \eta \nabla_{\mathbf{w}} l(\mathbf{w})$

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \frac{\partial l(\mathbf{w})}{\partial w_i}$$

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

 38

Maximize Conditional Log Likelihood: Gradient ascent

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

 39

Gradient Descent for LR

Gradient ascent algorithm: iterate until change < ε

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \sum_j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w})]$$

For i=1,…,n,

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

repeat

 40

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
  - More on this later in the semester
- MAP estimate

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

 41

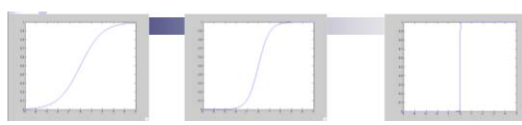
M(C)AP as Regularization

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

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

Penalizes high weights, also applicable in linear regression

Large parameters → Overfitting



$\frac{1}{1+e^{-x}}$ $\frac{1}{1+e^{-2x}}$ $\frac{1}{1+e^{-100x}}$

- If data is linearly separable, weights go to infinity
- Leads to overfitting:

- Penalizing high weights can prevent overfitting…
  - again, more on this later in the semester

 43

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

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

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

- Maximum conditional a posteriori estimate

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

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

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

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

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

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

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

Ratio of class-conditional probabilities

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

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

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

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

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

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

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

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

Naïve Bayes vs Logistic Regression

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

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 1

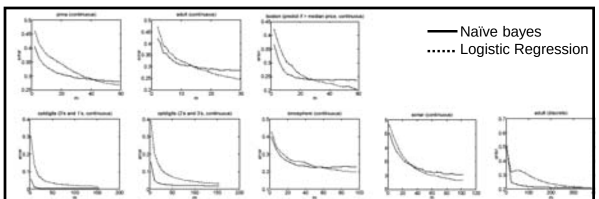
[Ng & Jordan, 2002]

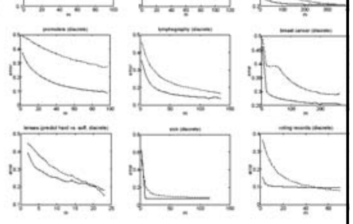
- Generative and Discriminative classifiers
- Asymptotic comparison (# training examples → infinity)
  - when model correct
    - GNB, LR produce identical classifiers
  - when model incorrect
    - LR is less biased – does not assume conditional independence
      - **therefore LR expected to outperform GNB**

G. Naïve Bayes vs. Logistic Regression 2

[Ng & Jordan, 2002]

- Generative and Discriminative classifiers
- Non-asymptotic analysis
  - convergence rate of parameter estimates, n = # of attributes in X
    - Size of training data to get close to infinite data solution
    - GNB needs *O*(log n) samples
    - LR needs *O*(n) samples
  - **GNB converges more quickly to its (perhaps less helpful) asymptotic estimates**





Some experiments from UCI data sets

Figure 1: Results of 15 experiments on datasets from the UCI Machine Learning repository. Plots are of generalization error vs. m (averaged over 1000 random train/test splits). Dashed line is logistic regression; solid line is naive Bayes.

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
- Convergence rates
  - GNB (usually) needs less data
  - LR (usually) gets to better solutions in the limit

