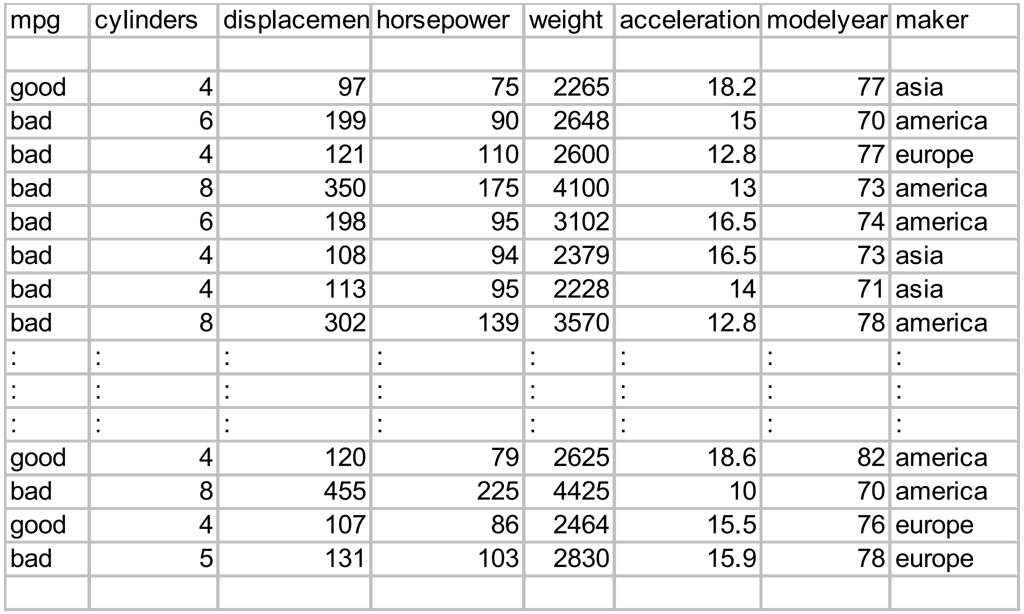
CSE446: Logistic Regression
Spring 2017

Ali Farhadi
Hessam Bagherinezhad

Slides adapted from Carlos Guestrin and Luke Zettlemoyer

Lets take a(nother) probabilistic approach!!!

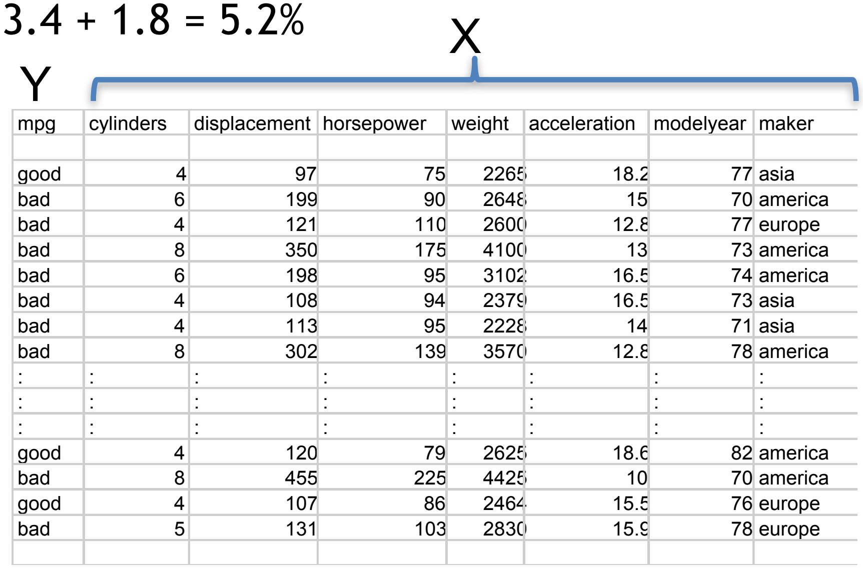
- Previously: directly estimate the data distribution P(X,Y)!
  - challenging due to size of distribution!
  - make Naïve Bayes assumption: only need $P(X_i|Y)$!
- But wait, we classify according to:
  - $\max_Y P(Y|X)$
- Why not learn P(Y|X) directly?



mpg	cylinders	displacemen	horsepower	weight	acceleration	modelyear	maker
good	4	97	75	2265	18.2	77	asia
bad	6	199	90	2648	15	70	america
bad	4	121	110	2600	12.8	77	europe
bad	8	350	175	4100	13	73	america
bad	6	198	95	3102	16.5	74	america
bad	4	108	94	2379	16.5	73	asia
bad	4	113	95	2228	14	71	asia
bad	8	302	139	3570	12.8	78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
good	4	120	79	2625	18.6	82	america
bad	8	455	225	4425	10	70	america
good	4	107	86	2464	15.5	76	europe
bad	5	131	103	2830	15.9	78	europe

What does that mean tho?

- P(Y|X): P(mpg=good | cylinders=6, maker=europe, …)
  - If I randomly pick a European car with 6 cylinders, what's the probability that it has a good mpg?
    - Possible answer: 70%
    - And, of course, P(mpg=bad | cylinders=6, maker=europe, …) = 30%
- P(X,Y): P(mpg=good, cylinders=6, maker=europe, …)
  - If I pick a car randomly, what's the probability it's European, has 6 cylinders and a good mpg?
    - Possible answer: 3.4%
    - Let's say P(mpg=good, cylnd=6, mkr=eu, …) = 1.8%
    - Now we know P(cylnd=6, mkr=eu, …) = 3.4 + 1.8 = 5.2%
  - This has way more information!
    - And is harder to train.

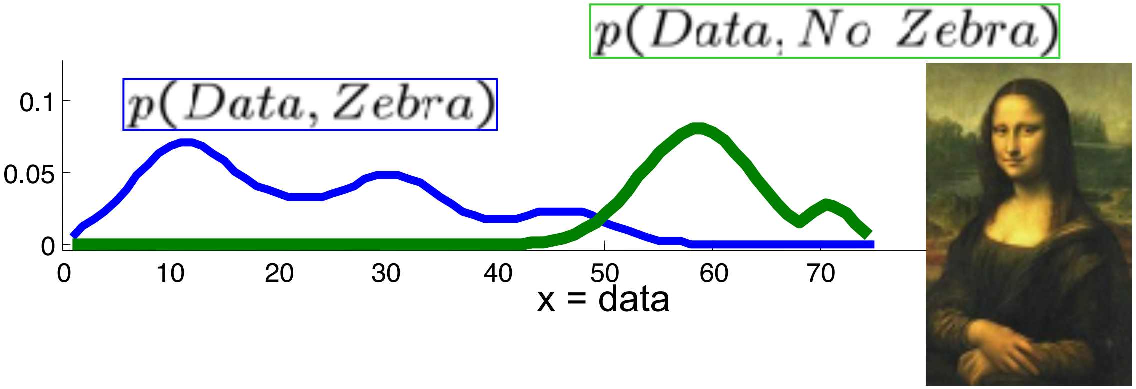


mpg	cylinders	displacement	horsepower	weight	acceleration	modelyear	maker
good	4	97	75	2265	18.2	77	asia
bad	6	199	90	2648	15	70	america
bad	4	121	110	2600	12.8	77	europe
bad	8	350	175	4100	13	73	america
bad	6	198	95	3102	16.5	74	america
bad	4	108	94	2379	16.5	73	asia
bad	4	113	95	2228	14	71	asia
bad	8	302	139	3570	12.8	78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
good	4	120	79	2625	18.6	82	america
bad	8	455	225	4425	10	70	america
good	4	107	86	2464	15.5	76	europe
bad	5	131	103	2830	15.9	78	europe

Discriminative vs. generative

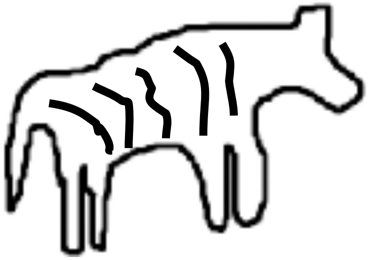
- Generative model

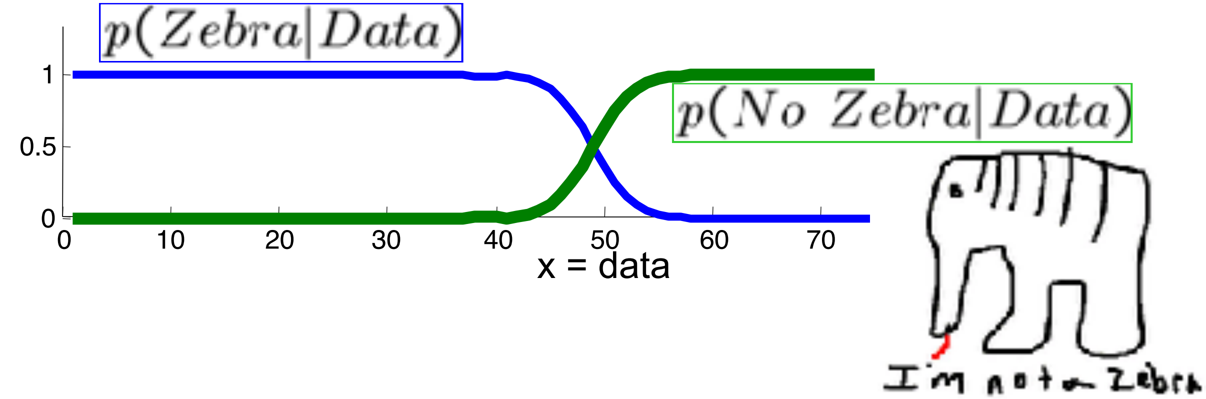
  (*The artist*)



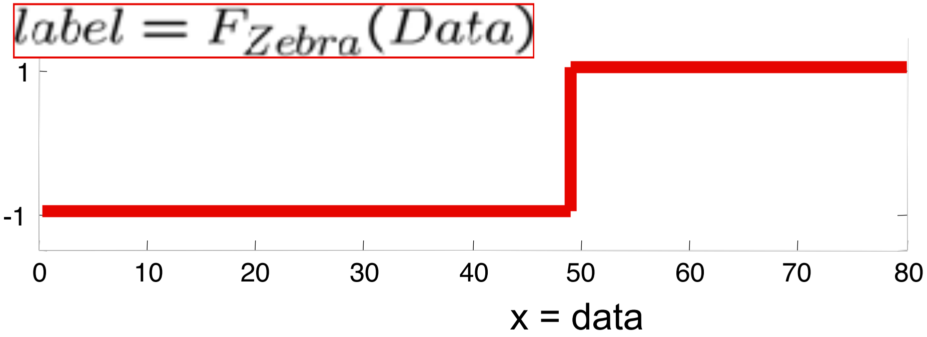
- Discriminative model

  (The lousy painter)





- Classification function

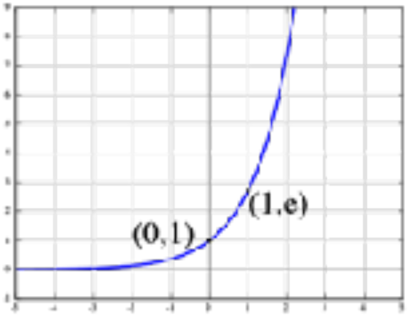


Logistic Regression

- Learn P(Y|**X**) directly!
  - Reuse ideas from regression, but let y-intercept define the probability

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

Exponential:



$$y = e^x = exp(x)$$

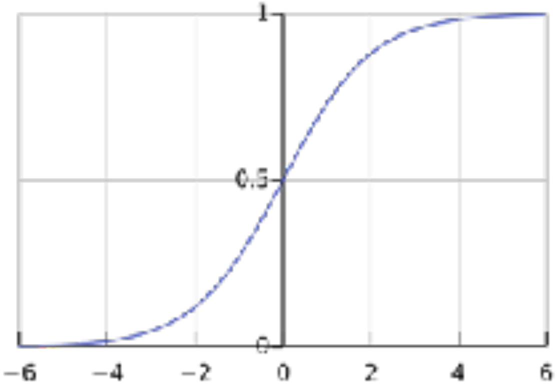
  - With normalization constants:

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

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

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

Logistic function



$$\sigma(x) = \frac{\exp(x)}{1+\exp(x)}$$

$$P(Y=1|X,w) = \sigma(w_0 + \sum_i w_i X_i)$$

Logistic Regression: decision boundary

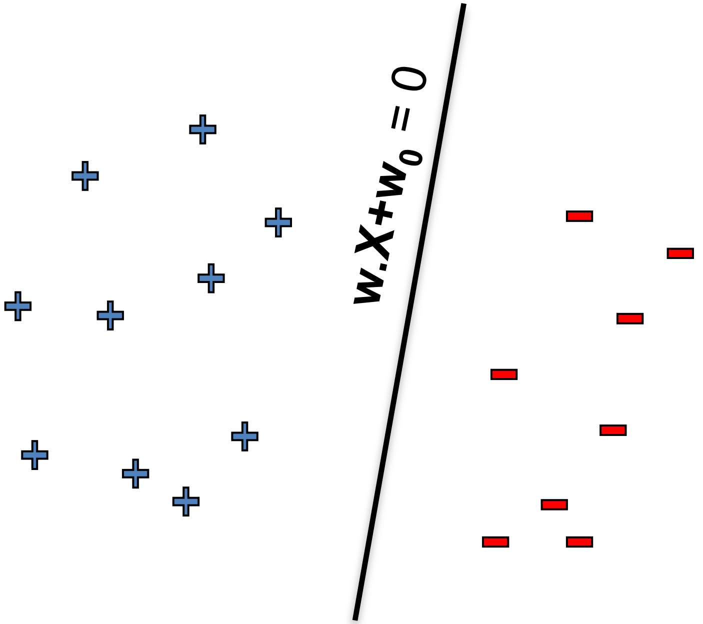
$$P(Y=0|\mathbf{X},\mathbf{w}) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)} \qquad P(Y=1|\mathbf{X},\mathbf{w}) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

- Prediction: Output the Y with highest P(Y|X)
  - Output Y=1 if

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

$$1 < \exp(w_0 + \sum_{i=1}^{n} w_i X_i)$$

$$0 < w_0 + \sum_{i=1}^{n} w_i X_i$$

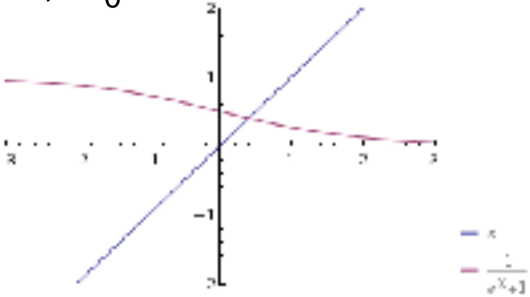


A Linear Classifier!

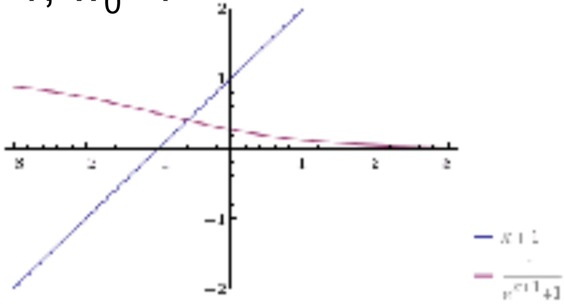
Visualizing 1D inputs

$$P(Y=0|\mathbf{X},\mathbf{w}) = \frac{1}{1+exp(w_0+w_1x_1)}$$

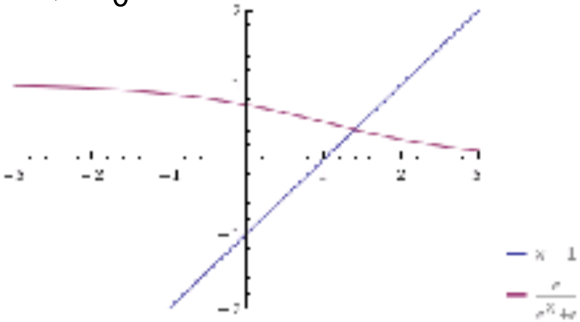
w_1=1, w_0=0



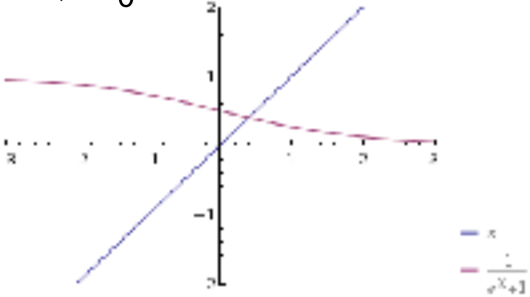
w_1=1, w_0=1



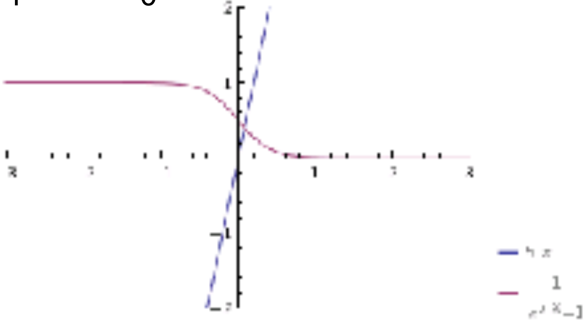
w_1=1, w_0=-1



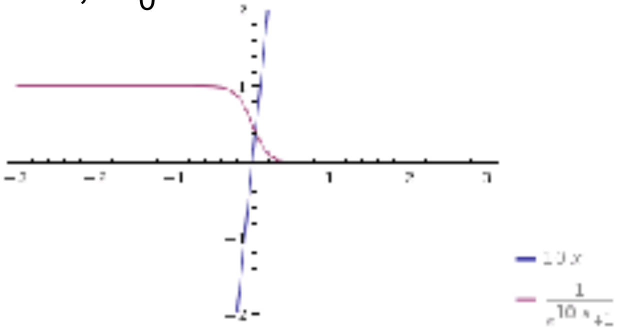
w_1=1, w_0=0



w_1=5, w_0=0



w_1=10, w_0=0



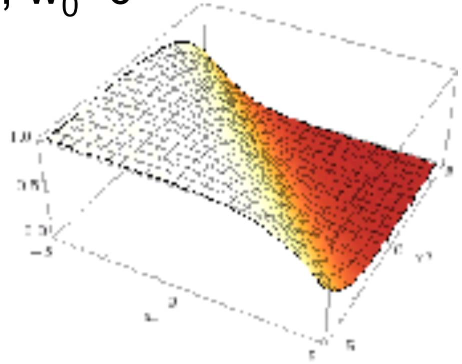
Notes:

- Defines a probability distribution over Y in {0,1} for every possible input X
- Decision boundary: P(Y=0|X,w)=0.5 when at the y=0 point on the line
- Slope of line defines how quickly probabilities go to 0 or 1 around decision boundary

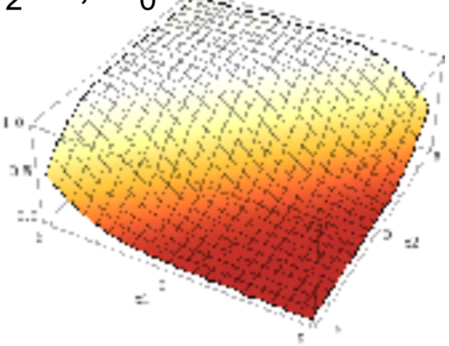
Visualizing 2D inputs

$$P(Y=0|\mathbf{X},\mathbf{w}) = \frac{1}{1+exp(w_0+w_1x_1+w_2x_2)}$$

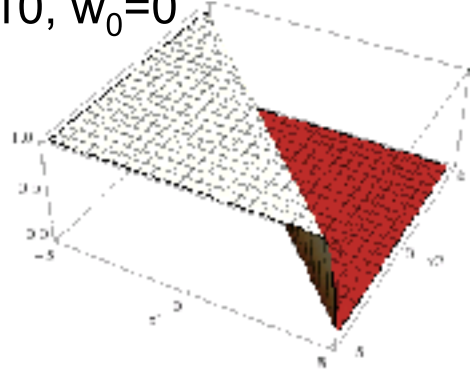
w_1=1, w_2=1, w_0=0



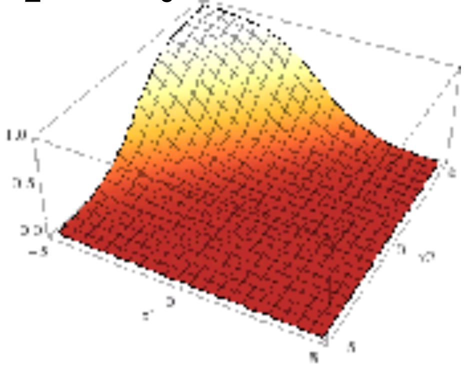
w_1=-1, w_2=1, w_0=0



w_1=10, w_2=10, w_0=0



w_1=-1, w_2=1, w_0=5



What about higher dimensions?

- Difficult to visualize!
- P(Y=0|X,w) decreases as $w_0+\Sigma_i w_i x_i$ increases
- Decision boundary is defined by $w_0+\Sigma_i w_i x_i$ =0 hyperplane

Loss functions / Learning Objectives: Likelihood v. Conditional Likelihood

- Generative (Naïve Bayes) Loss function:

  Data likelihood

$$\begin{aligned}\ln P(D|w) &= \sum_j \ln P(x^j, y^j|w) \\ &= \sum_j \ln P(x^j|y^j, w) + \sum_j \ln P(y^j|w)\end{aligned}$$

- 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|Y)
  - Discriminative models cannot compute $P(X^j|Y^j)$!

Conditional Log Likelihood

(the binary case only)

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

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

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

equal because y^j is in {0,1}

$$l(w) = \sum_j y^j \ln P(Y=1|X^j,w) + (1-y^j)\ln P(Y=0|X^j,w)$$

remaining steps: substitute definitions, expand logs, and simplify

$$= \sum_j y^j \ln \frac{e^{w_0+\sum_i w_i X_i}}{1+e^{w_0+\sum_i w_i X_i}} + (1-y^j)\ln \frac{1}{1+e^{w_0+\sum_i w_i X_i}}$$

...

$$= \sum_j y^j(w_0+\sum_i^n w_i x_i^j) - \ln(1+exp(w_0+\sum_i^n w_i x_i^j))$$

Logistic Regression Parameter Estimation: Maximize Conditional Log Likelihood

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

Good news: *l*(**w**) is a concave function of **w**

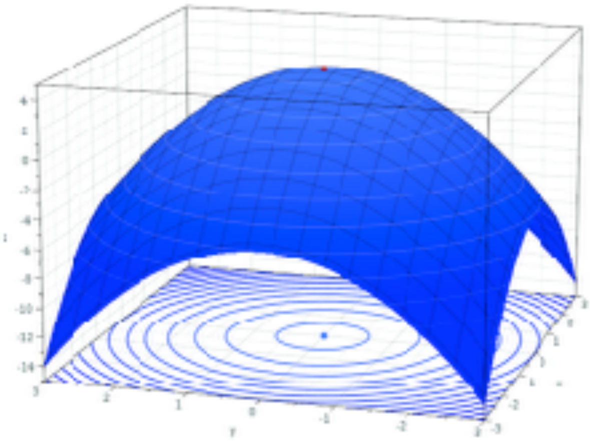
→ no locally optimal solutions!

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

Good news: concave functions “easy” to optimize

Optimizing convex function – Gradient ascent

- Conditional likelihood for Logistic Regression is convex!



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
  - Yet works in most cases

Maximize Conditional Log Likelihood: Gradient ascent

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

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

$$\frac{\partial l(w)}{\partial w_i} = \sum_j \left[\frac{\partial}{\partial w} y^j (w_0 + \sum_i w_i x_i^j) - \frac{\partial}{\partial w} \ln \left(1 + \exp(w_0 + \sum_i w_i x_i^j) \right) \right]$$

$$= \sum_j \left[y^j x_i^j - \frac{x_i^j \exp(w_0 + \sum_i w_i x_i^j)}{1 + \exp(w_0 + \sum_i w_i x_i^j)} \right]$$

$$= \sum_j x_i^j \left[y^j - \frac{\exp(w_0 + \sum_i w_i x_i^j)}{1 + \exp(w_0 + \sum_i w_i x_i^j)} \right]$$

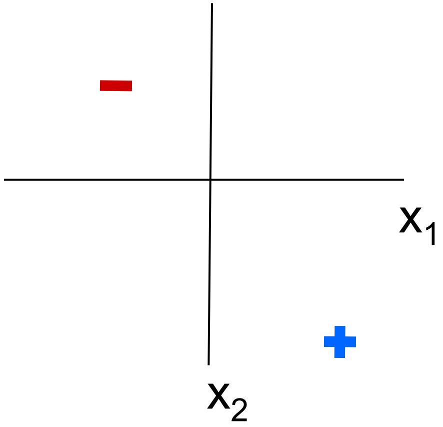
$$\boxed{\frac{\partial l(w)}{\partial w_i} = \sum_j x_i^j \left(y^j - P(Y^j = 1 | x^j, w) \right)}$$

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

$$\frac{\partial l(w)}{\partial w_i} = \sum_j x_i^j \left(y^j - P(Y^j = 1|x^j, w)\right)$$

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

x_1	x_2	y
3	-3	1
-2	2	0



t=0:

$w = [w_0,w_1,w_2] = [0,0,0]$

$P(Y^0=1|x^0,w)$ α exp(0+0*3+0*-3) = 0.5

$P(Y^1=1|x^1,w)$ α exp(0+0*-2+0*2) = 0.5

i=0, j=0: $x_0^0(y^0-P(Y=1|x^0,w))$ = 1(1-0.5) = 0.5

i=0, j=1: $x_0^1(y^1-P(Y=1|x^1,w))$ = 1(0-0.5) = -0.5

i=1, j=0: $x_1^0(y^0-P(Y=1|x^0,w))$ = 3(1-0.5) = 1.5

i=1, j=1: $x_1^1(y^1-P(Y=1|x^1,w))$ = -2(0-0.5) = 1.0

i=2, j=0: $x_2^0(y^0-P(Y=1|x^0,w))$ = -3(1-0.5) = -1.5

i=2, j=1: $x_2^1(y^1-P(Y=1|x^1,w))$ = 2(0-0.5) = -1.0

grad = [0.5-0.5, 1.5+1.0, -1.5-1] = [0,2.5,-2.5

t=1:

η=0.1 → w = [0,0,0] + 0.1 * [0,2.5,-2.5] = [0,0.25,-0.25]

$P(Y^0=1|x^0,w)$ α exp(0+0.25*3-0.25*-3) = 0.82

$P(Y^1=1|x^1,w)$ α exp(0+0.25*-2-0.25*2) = 0.27

i=0, j=0: $x_0^0(y^0-P(Y^0=1|x^0,w))$ = 1(1-0.82) = 0.1

i=0, j=1: $x_0^1(y^1-P(Y^1=1|x^1,w))$ = 1(0-0.27) = -0.27

i=1, j=0: $x_1^0(y^0-P(Y^0=1|x^0,w))$ = 3(1-0.82) = 0.5

i=1, j=1: $x_1^1(y^1-P(Y^1=1|x^1,w))$ = -2(0-0.27) = 0.54

i=2, j=0: $x_2^0(y^0-P(Y^0=1|x^0,w))$ = -3(1-0.82) = -0.54

Gradient Ascent for LR

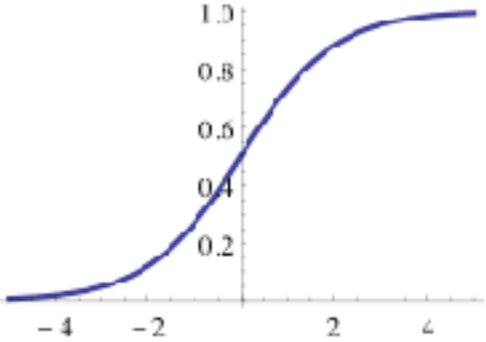
Gradient ascent algorithm: (learning rate η > 0)

```
do:
    $w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \sum_j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w})]$
    For i=1…n: (iterate over weights)
        $w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w})]$
```

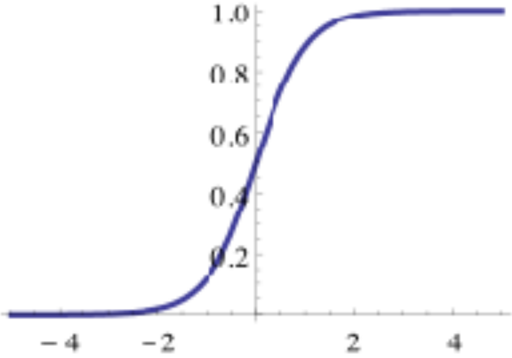
Loop over training examples!

Large parameters…

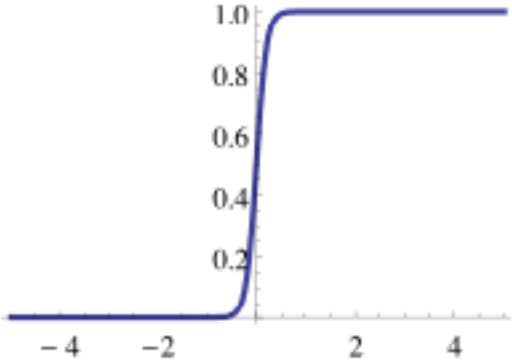
$$\frac{1}{1+e^{-ax}}$$



a=1



a=5



a=10

- Maximum likelihood solution: prefers higher weights
  - higher likelihood of (properly classified) examples close to decision boundary
  - larger influence of corresponding features on decision
  - *can cause overfitting!!!*
- Regularization: penalize high weights
  - again, more on this later in the quarter

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

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

- Often called ***Regularization***
  - Helps avoid very large weights and overfitting

- MAP estimate:

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

M(C)AP as Regularization

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \ln \left[p(\mathbf{w}) \prod_{j=1}^{N} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right] \qquad p(\mathbf{w}) = \prod_i \frac{1}{\kappa\sqrt{2\pi}} \; e^{\frac{-w_i^2}{2\kappa^2}}$$

- Add log p(w) to objective:

$$\ln p(w) \propto -\frac{\lambda}{2} \sum_i w_i^2 \qquad \frac{\partial \ln p(w)}{\partial w_i} = -\lambda w_i$$

  - Quadratic penalty: drives weights towards zero
  - Adds a negative linear term to the gradients

Penalizes high weights, also applicable in linear regression

MLE vs. MAP

- Maximum conditional likelihood estimate

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

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

- Maximum conditional a posteriori estimate

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

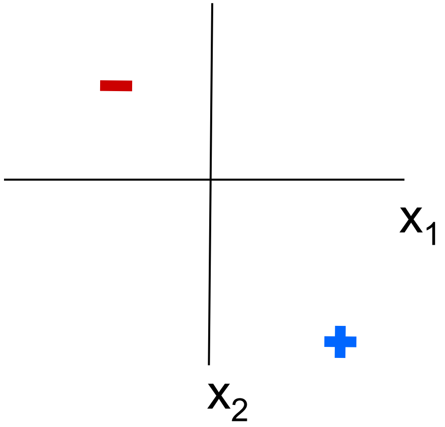
$$\boxed{w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \left\{ -\lambda w_i^{(t)} + \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w})] \right\}}$$

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

$$\frac{\partial l(w)}{\partial w_i} = \sum_j x_i^j \left(y^j - P(Y^j = 1|x^j, w)\right) - \lambda w_i$$

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

x_1	x_2	y
3	-3	1
-2	2	0



t=0:

w = [w_0,w_1,w_2] = [0,0,0]

… see earlier slide, same computations as without regularization…

grad = [0.5-0.5, 1.5+1.0, -1.5-1] = [0,2.5,-2.5]

λ=0.1 → grad -= 0.1 * [0,0,0]

t=1:

η=0.1 → w = [0,0,0] + 0.1 * [0,2.5,-2.5] = [0,0.25,-0.25]

… see earlier slide, same computations as without regularization…

grad = [0.13-0.27, 0.36+0.54, -0.36-0.54] = [-0.14,1,-1]

λ=0.1 → grad -= 0.1 * [0,0.25,-0.25]

t=2:

….

Logistic regression for discrete classification

Logistic regression in more general case, where set of possible *Y* is $\{y_1,\ldots,y_R\}$

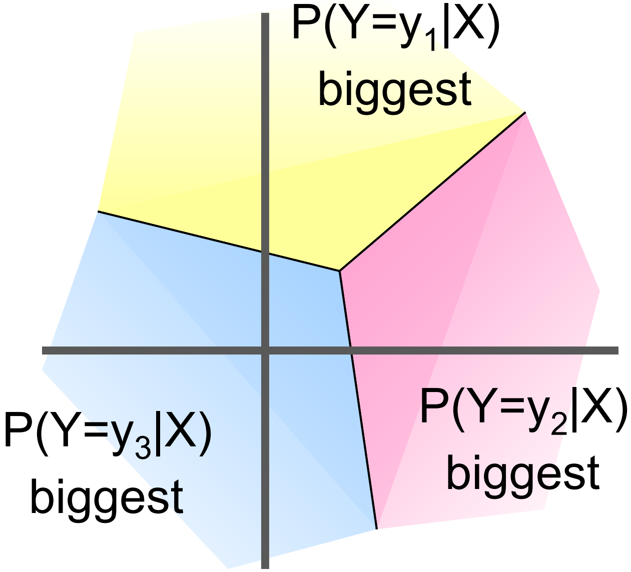
- Define a weight vector w_i for each y_i, i=1,…,R-1

$$P(Y=1|X) \propto \exp(w_{10} + \sum_i w_{1i} X_i)$$

$$P(Y=2|X) \propto \exp(w_{20} + \sum_i w_{2i} X_i)$$

…

$$P(Y=r|X) = 1 - \sum_{j=1}^{r-1} P(Y=j|X)$$



Logistic regression: discrete Y

- Logistic regression in more general case, where *Y* is in the set $\{y_1,\ldots,y_R\}$

  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!

Logistic regression v. Naïve Bayes

- Consider learning f: X → Y, where
  - X is a vector of real-valued features, < X_1 … X_n >
  - 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

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

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

up to now, all arithmetic

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

only for Naïve Bayes models

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

Looks like a setting for w_0?

Can we solve for w_i ?

- Yes, but only in Gaussian case

Ratio of class-conditional probabilities

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

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

$$= \ln \left[\frac{\frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{(x_i-\mu_{i0})^2}{2\sigma_i^2}}}{\frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{(x_i-\mu_{i1})^2}{2\sigma_i^2}}} \right]$$

$$= -\frac{(x_i-\mu_{i0})^2}{2\sigma_i^2} + \frac{(x_i-\mu_{i1})^2}{2\sigma_i^2}$$

$$\cdots$$

$$= \frac{\mu_{i0}+\mu_{i1}}{\sigma_i^2} x_i + \frac{\mu_{i0}^2+\mu_{i1}^2}{2\sigma_i^2}$$

Linear function! Coefficients expressed with original Gaussian parameters!

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

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

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

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

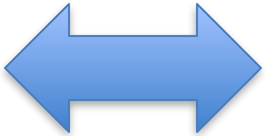
$$P(Y=1|X) = \frac{1}{1+\exp(w_0+\sum_{i=1}^n w_i X_i)}$$

$$\boxed{w_0 = \ln\frac{1-\theta}{\theta} + \frac{\mu_{i0}^2+\mu_{i1}^2}{2\sigma_i^2}}$$

$$\boxed{w_i = \frac{\mu_{i0}+\mu_{i1}}{\sigma_i^2}}$$

Gaussian Naïve Bayes vs. Logistic Regression

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



Can go both ways, we only did one way

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, $X=<X_1 \ldots X_n>$

Number of parameters:

- Naïve Bayes: 4n +1
- Logistic Regression: n+1

Estimation method:

- Naïve Bayes parameter estimates are uncoupled
- Logistic Regression parameter estimates are coupled

Naïve Bayes vs. Logistic Regression

[Ng & Jordan, 2002]

- Generative vs. Discriminative classifiers
- Asymptotic comparison

  (# training examples → infinity)

  - when model correct
    - GNB (with class independent variances) and LR produce identical classifiers
  - when model incorrect
    - LR is less biased – does not assume conditional independence
      - therefore LR expected to outperform GNB

Naïve Bayes vs. Logistic Regression

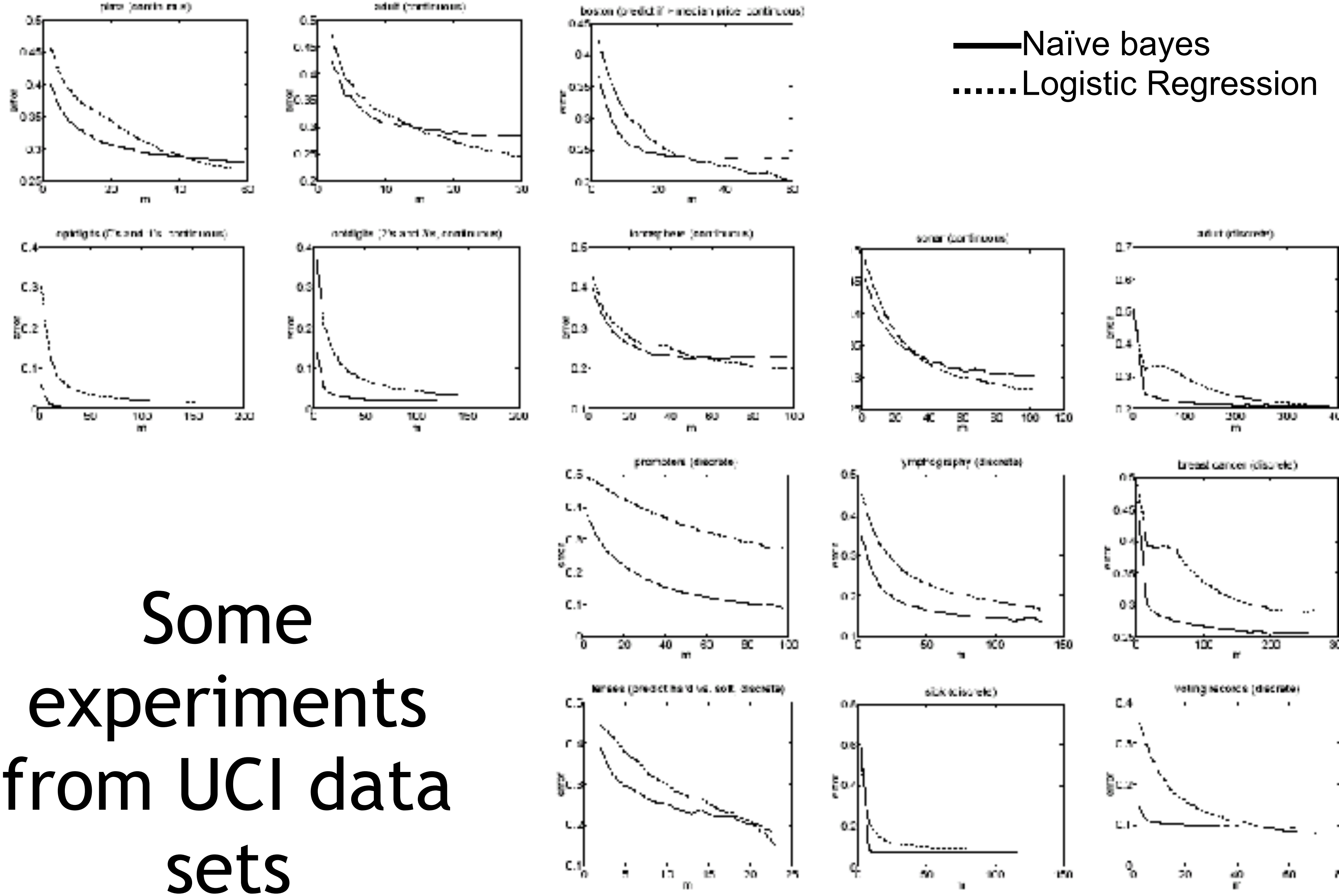
[Ng & Jordan, 2002]

- Generative vs. 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
    - Naïve Bayes needs *O*(log n) samples
    - Logistic Regression 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 Learnin repository. Plots are of generalization error vs. m (averaged over 1000 randor 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