Generative *vs.* Discriminative Classifiers

-
  Want to Learn: $h: \mathbf{X} \mapsto Y$
  - **X** – features
  - Y – target classes
- **Generative classifier**, e.g., Naïve Bayes:

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

  - 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**, $P(X) = \sum_y P(y)\, P(X|y)$
-

Generative *vs.* Discriminative Classifiers

- **Want to Learn**: h:**X** $\mapsto$ Y
  - **X** – features
  - Y – target classes
- **Generative classifier**, e.g., Naïve Bayes:

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

  - 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

Remember Gaussian Naïve Bayes?

Sometimes Assume Variance

– is independent of Y (i.e., σ_i),

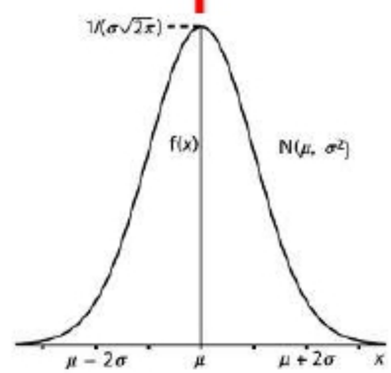
– or independent of X_i (i.e., σ_k)

– or both (i.e., σ)

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

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

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



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

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

Ratio of class-conditional probabilities

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

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

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

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

$$\cdots$$

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

Ratio of class-conditional probabilities

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

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

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

$$\cdots$$

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

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

Linear function! Coefficients expressed with original Gaussian parameters!

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

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

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

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

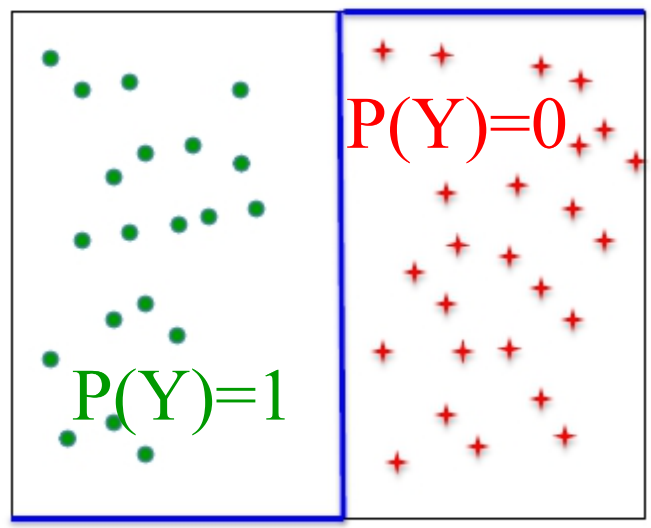
$$P(Y=1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^n w_i X_i)}$$

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

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

Logistic Regression

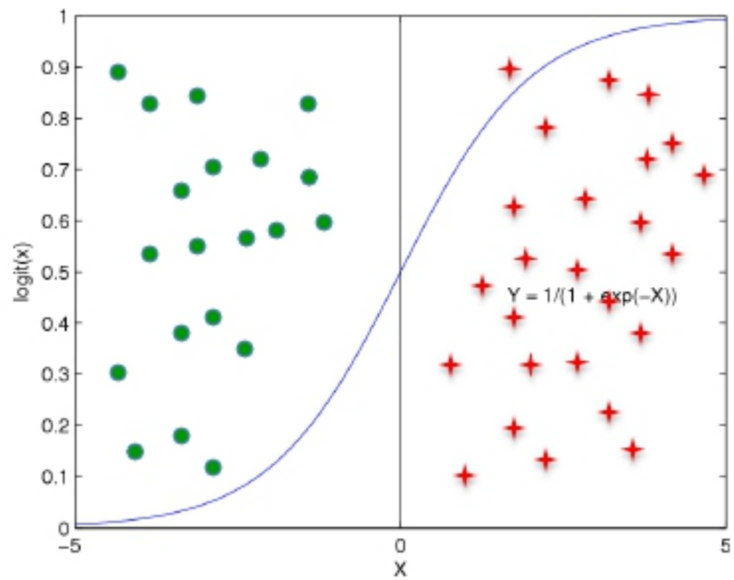
- Learn P(Y|**X**) directly!
  - Assume a particular functional form
  - ☹ ***Not differentiable…***



Logistic Regression

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

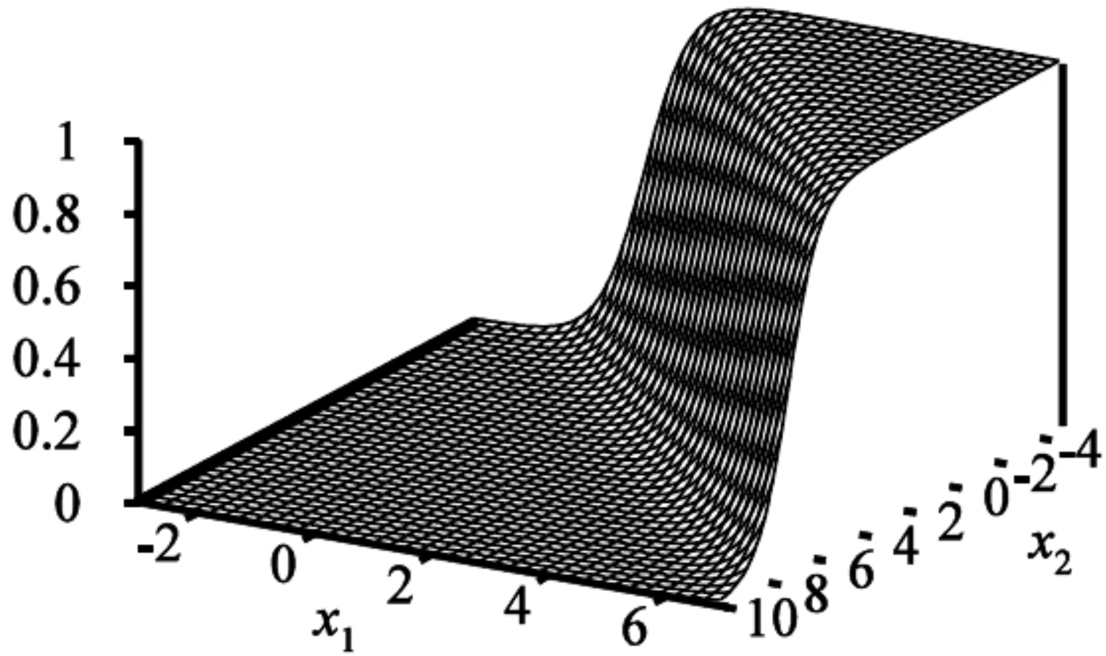
$$\frac{1}{1+exp(-z)}$$



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:

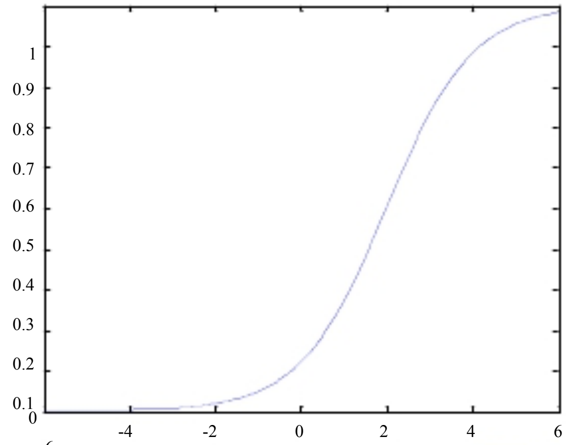


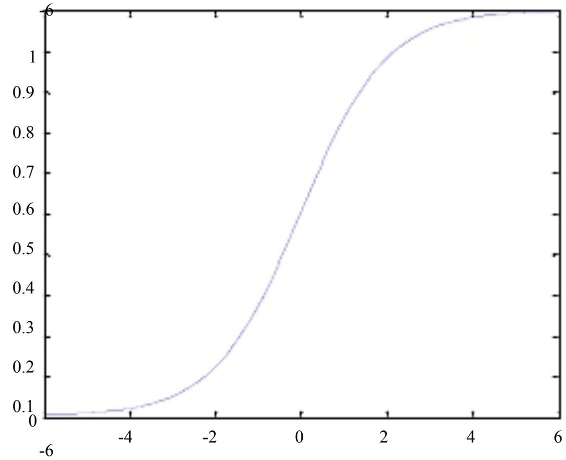
Features can be discrete or continuous!

Understanding Sigmoids

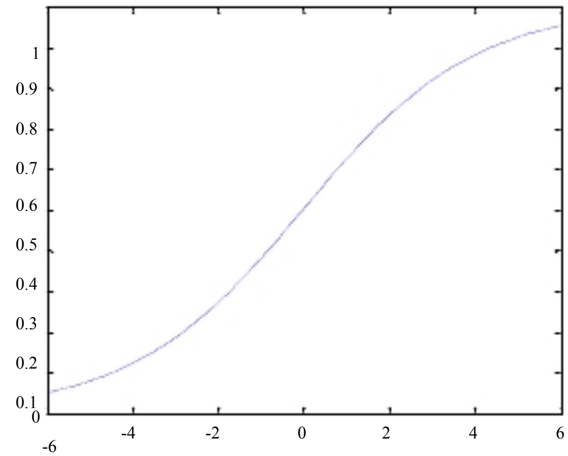
$$g(w_0 + \sum_i w_i x_i) = \frac{1}{1 + e^{w_0 + \sum_i w_i x_i}}$$

w0=-2, w1=-1





w0=0, w1=-1



w0=0, w1= -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)?

Very convenient!

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

implies

$$P(Y=0|X=<X_1,...X_n>) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

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

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

implies

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

linear classification rule!

Likelihood vs. Conditional Likelihood

Generative (Naïve Bayes) maximizes **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 (Logistic Regr.) maximizes **Conditional Data Likelihood**

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

Discriminative models *can't* compute P($\mathbf{x}_j$|**w**)!

Or, … "They don't *waste effort* learning P(**X**)"

Focus only on P(Y|**X**) - all that matters for classification

Maximizing Conditional Log Likelihood

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

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

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

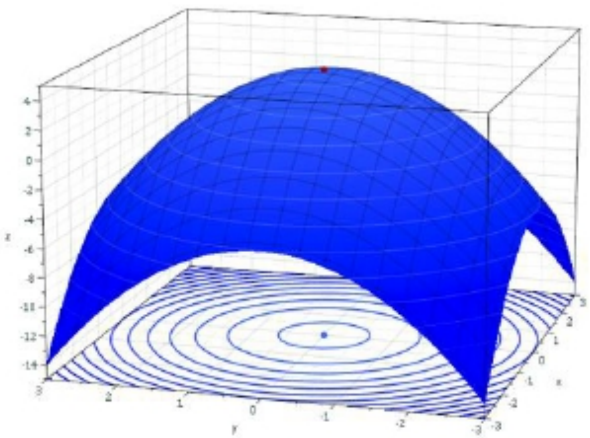
Good news: $l(\mathbf{w})$ is concave function of $\mathbf{w}$!

No local minima

Concave functions easy to optimize

Optimizing concave function – Gradient ascent

- Conditional likelihood for Logistic Regression is concave !



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

Gradient Ascent for LR

Gradient ascent algorithm: (learning rate $\eta > 0$)

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 to n: (iterate over weights)

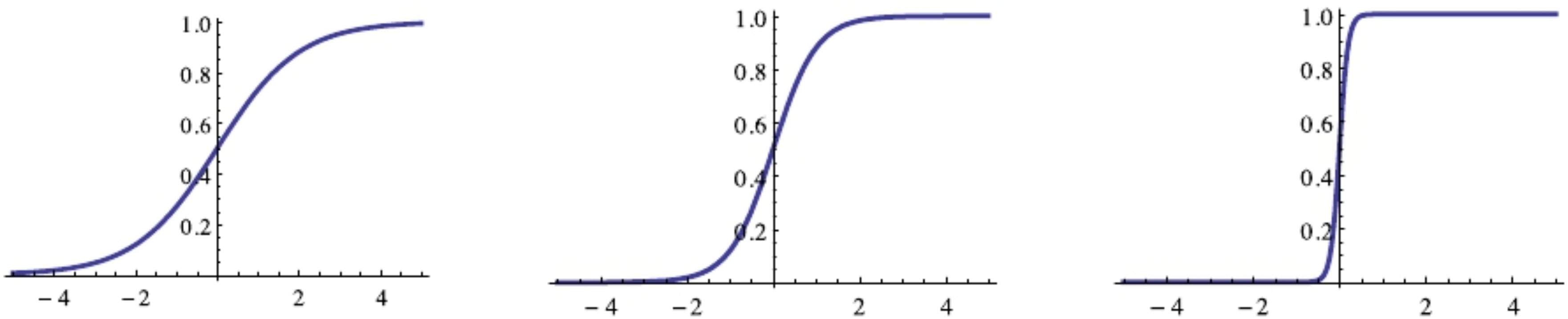
$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w})]$$

until "change" < ε

Loop over training examples!

Large parameters…

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



a=1 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

That's all MCLE. How about MCAP?

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

- One common approach is to define priors on **w**
  - Normal distribution, zero mean, identity covariance
  - "Pushes" parameters towards zero

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

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

- MAP estimate:

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

MCAP as Regularization

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

- 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, like we did in linear regression

MCLE vs. MCAP

- 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

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

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

Naïve Bayes *vs.* Logistic Regression

Learning: h:**X** $\mapsto$ Y

X – features
Y – target classes

Generative

- Assume functional form for
  - P(X|Y) **assume cond indep**
  - P(Y)
  - Est params from train data
- Gaussian NB for cont features
- Bayes rule to calc. P(Y|X= x)
  - P(Y | **X**) $\propto$ P(**X** | Y) P(Y)
- **Indirect** computation
  - Can also generate a sample of the data

Discriminative

- Assume functional form for
  - P(Y|X) **no assumptions**
  - Est params from training data
- Handles discrete & cont features
- Directly calculate P(Y|X=x)
  - Can't generate data sample

Gaussian Naïve Bayes *vs.* Logistic Regression

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

Generative

- Assume functional form for
  - P(X|Y) assume X_i cond indep given Y
  - P(Y)
  - Est params from train data
- Gaussian NB for continuous features
  - model P(X_i | Y = y_k) as **Gaussian** N(μ_{ik},σ_i)
  - model P(Y) as **Bernoulli** (θ,1-θ)
- Bayes rule to calc. P(Y|X= x)
  - P(Y | **X**) $\propto$ P(**X** | Y) P(Y)

What can we say about the form of P(Y=1 | …X_i…)?

$$\boxed{\frac{1}{1+exp(w_0+\sum_i w_i X_i)}}$$

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, $X = \langle X_1 \ldots X_n \rangle$ continuous

Number of parameters:

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

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**

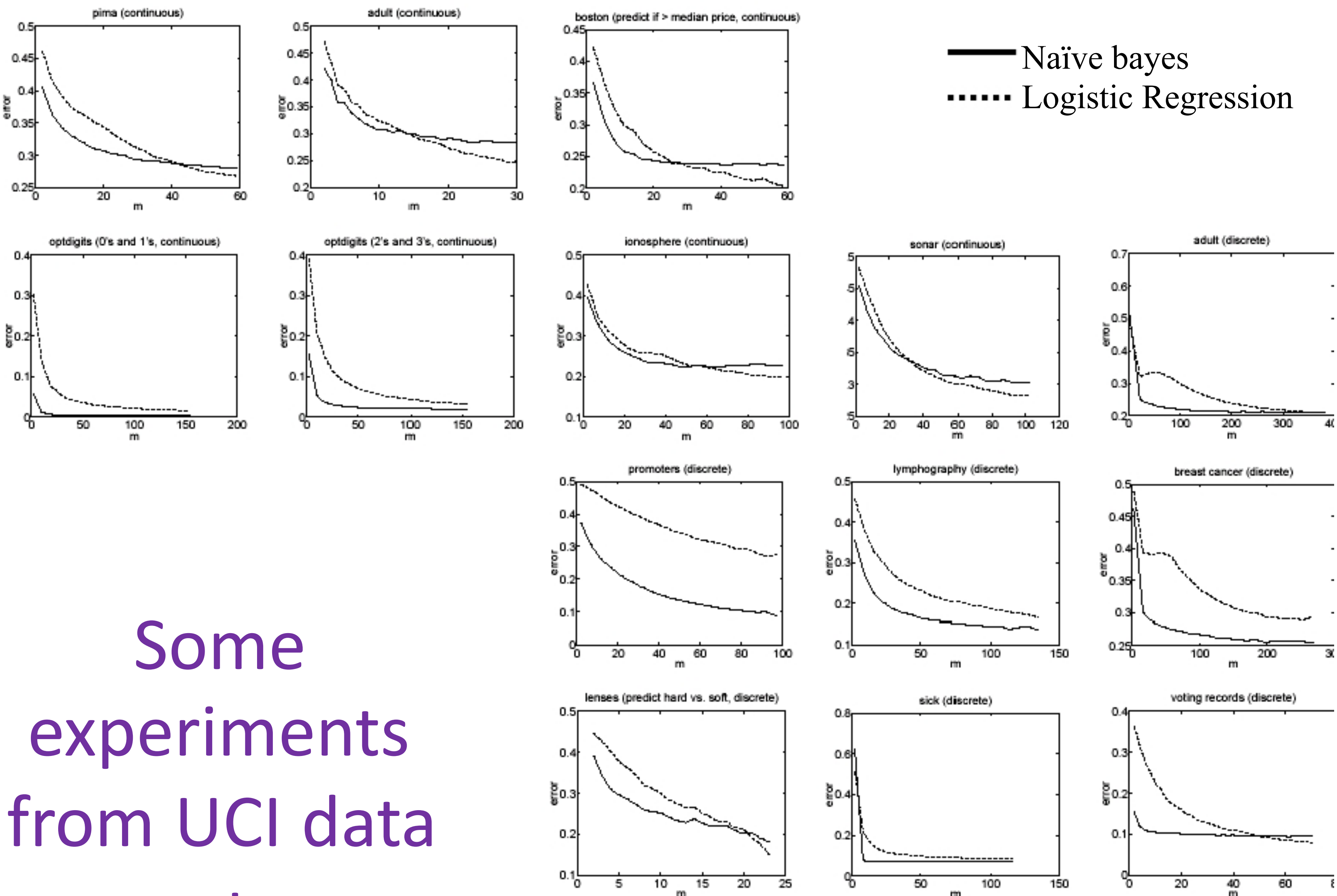
Naïve Bayes vs. Logistic Regression

[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

LR: MCAP Algorithm

- Given Data matrix of size m x (n+2) (n attributes and m examples)
  - Data[i][n+1] gives the class for example i
  - Data[i][0] is the dummy threshold attribute always set to 1.
- Arrays Pr[0..m-1] and w[0..n] initialized to random values
- Until convergence do
  - For each example i
    - Compute Pr[i]=Pr(class=1|Data[i],w)
  - Array dw[0..n] initialized to zero
  - For i=0 to n // Go over all the weights
    - For j=0 to m-1 // Go over all the training examples
      - dw[i]=dw[i]+Data[j][i]*(Data[j][n+1]-Pr[j])
  - For i=0 to n
    - w[i]=w[i]+η(dw[i]-λw[i])