

# Logistic Regression

## Lecture 19

David Sontag  
New York University

Slides adapted from Vibhav Gogate, Luke Zettlemoyer,  
Carlos Guestrin, and Dan Weld

# Naïve Bayes

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

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

- Decision rule for classification:

$$\begin{aligned} y^* = h_{NB}(\mathbf{x}) &= \arg \max_y P(y) P(x_1, \dots, x_n | y) \\ &= \arg \max_y P(y) \prod_i P(x_i | y) \end{aligned}$$

# What about if there is missing data?

- One of the key strengths of Bayesian approaches is that they can naturally handle missing data
- Suppose don't have value for some attribute  $X_i$ 
  - applicant's credit history unknown
  - some medical test not performed on patient
  - how to compute  $P(X_1=x_1 \dots X_j=? \dots X_d=x_d | y)$

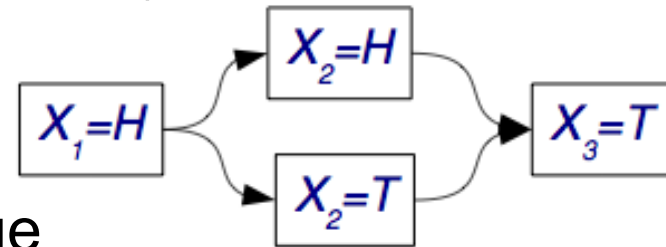
- Easy with Naïve Bayes

- ignore attribute in instance where its value is missing
- compute likelihood based on observed attributes
- no need to “fill in” or explicitly model missing values
- based on conditional independence between attributes

$$P(x_1 \dots X_j \dots x_d | y) = \prod_{i \neq j}^d P(x_i | y)$$

# What about if there is missing data?

- Ex: three coin tosses: Event =  $\{X_1=H, X_2=?, X_3=T\}$ 
  - event = head, unknown (either head or tail), tail
  - event =  $\{H,H,T\} + \{H,T,T\}$
  - $P(\text{event}) = P(H,H,T) + P(H,T,T)$
- General case:  $X_j$  has missing value



$$P(x_1 \dots x_j \dots x_d | y) = P(x_1 | y) \cdots P(x_j | y) \cdots P(x_d | y)$$

$$\begin{aligned} \sum_{x_j} P(x_1 \dots x_j \dots x_d | y) &= \sum_{x_j} P(x_1 | y) \cdots P(x_j | y) \cdots P(x_d | y) \\ &= P(x_1 | y) \cdots \left[ \sum_{x_j} P(x_j | y) \right] \cdots P(x_d | y) \\ &= P(x_1 | y) \cdots [1] \cdots P(x_d | y) \end{aligned}$$

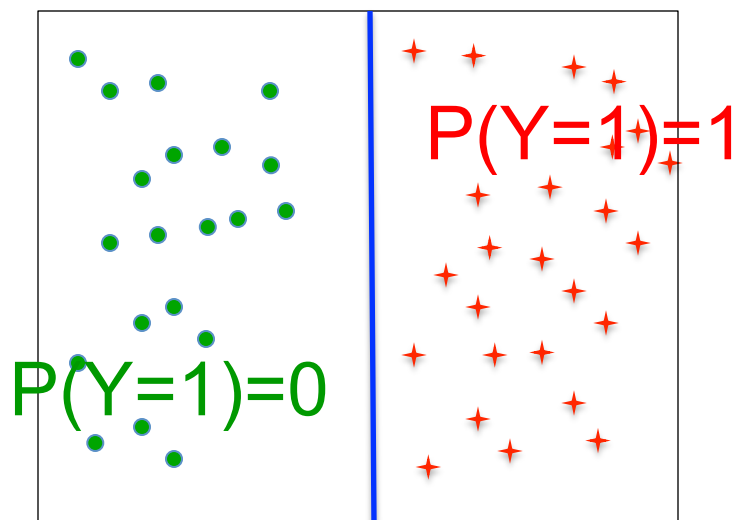
# Naive Bayes = Linear Classifier

- **Theorem:** assume that  $x_i \in \{0, 1\}$  for all  $i \in [1, N]$ . Then, the Naive Bayes classifier is defined by

$$\mathbf{x} \mapsto \text{sgn}(\mathbf{w} \cdot \mathbf{x} + b),$$

# Logistic Regression

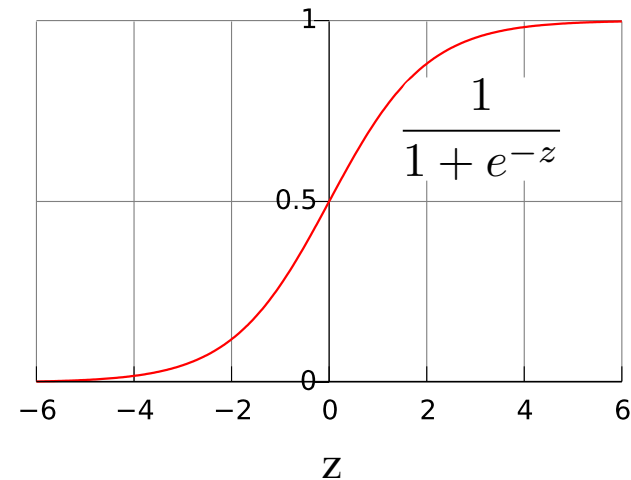
- Learn  $P(Y|\mathbf{X})$  directly!
  - Assume a particular functional form
  - ★ Linear classifier? On one side we say  $P(Y=1|X)=1$ , and on the other  $P(Y=1|X)=0$
  - ★ ***But, this is not differentiable (hard to learn)... doesn't allow for label noise...***



# Logistic Regression

Logistic function (Sigmoid):

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



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

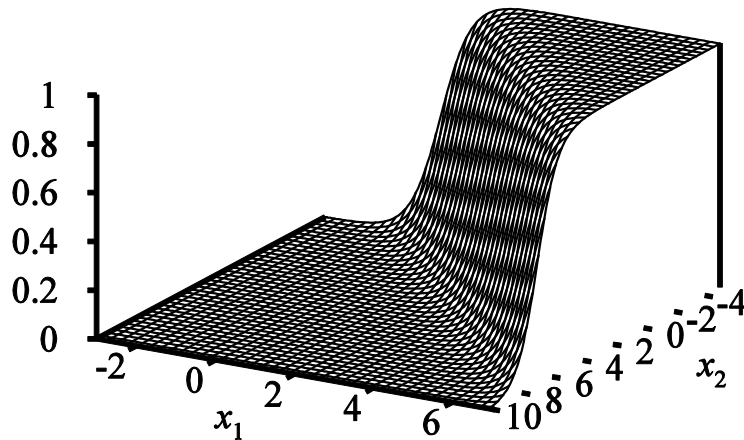
$$P(Y = 0|X) = \frac{\exp(w_0 + \sum_{i=1}^n w_i X_i)}{1 + \exp(w_0 + \sum_{i=1}^n w_i X_i)}$$

**Features can be discrete or continuous!**

# 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!**



# Logistic Regression: decision boundary

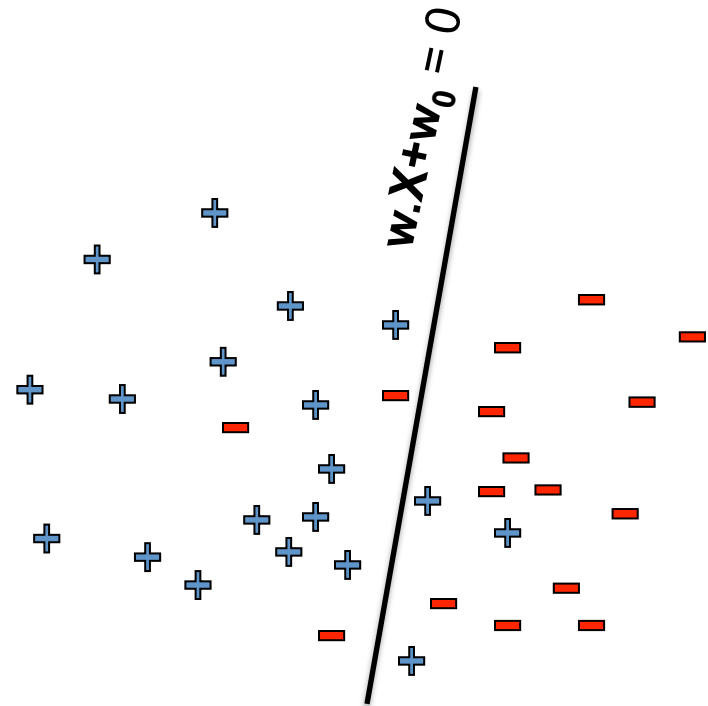
$$P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^n w_i X_i)} \quad P(Y = 0|X) = \frac{\exp(w_0 + \sum_{i=1}^n w_i X_i)}{1 + \exp(w_0 + \sum_{i=1}^n w_i X_i)}$$

- **Prediction:** Output the Y with highest  $P(Y|X)$ 
  - For binary Y, output  $Y=0$  if

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

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

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



**A Linear Classifier!**

# Likelihood vs. Conditional Likelihood

Generative (Naïve Bayes) maximizes **Data likelihood**

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

Discriminative (Logistic Regr.) maximizes **Conditional Data Likelihood**

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

Focuses only on learning  $P(Y | \mathbf{X})$  - all that matters for classification

# Maximizing Conditional Log Likelihood

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

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

0 or 1!

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

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

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

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

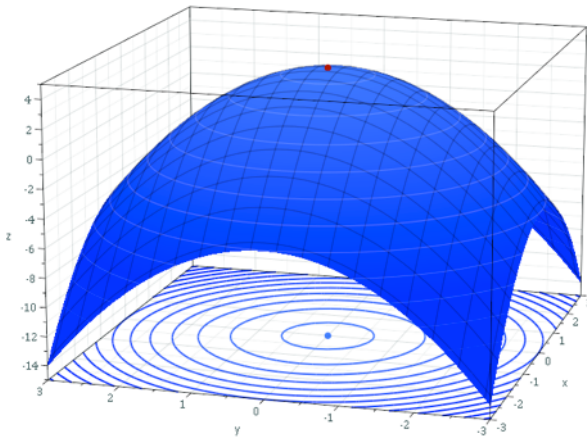
No local minima

Concave functions easy to optimize

# Optimizing concave function – Gradient ascent

- Conditional likelihood for Logistic Regression is concave →

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



Learning rate,  $\eta > 0$

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

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

# Maximize Conditional Log Likelihood: Gradient ascent

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

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

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

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

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

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

# 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 features)

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

until “change”  $< \epsilon$

Loop over training examples!



# That's all MLE. 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  $\mathbf{w}$ 
  - 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:

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

# MAP as Regularization

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

- Add  $\log p(w)$  to objective:

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

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

**Penalizes high weights, just like we did with SVMs!**



$$l(D; \vec{w}) = \sum_{i=1}^N \log \left( \frac{1}{1 + e^{-y_i(\vec{w}_0 + \vec{w} \cdot \vec{x}_i)}} \right)$$

MAP estimation of LR:

$$\max_{\vec{w}} \sum_{i=1}^N \log \left( \frac{1}{1 + e^{-y_i(\vec{w} \cdot \vec{x}_i)}} \right) - \lambda \|\vec{w}\|^2$$

$$\min_{\vec{w}} \sum_{i=1}^N \log \left( 1 + e^{-y_i(\vec{w} \cdot \vec{x}_i)} \right) + \lambda \|\vec{w}\|^2$$

logit loss

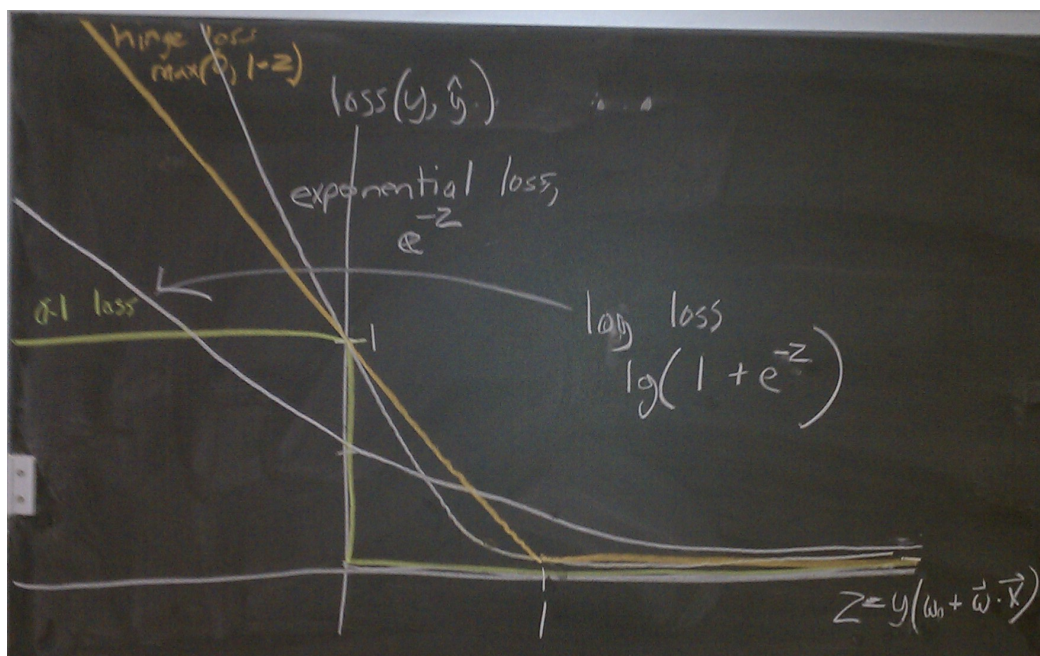
$y_i(\vec{w} \cdot \vec{x}_i) \geq 1 - \xi_i$

SVM

$$\min_{\vec{w}} \sum_{i=1}^N \underbrace{\max(0, 1 - y_i(\vec{w} \cdot \vec{x}_i))}_{\text{hinge loss}} + \lambda \|\vec{w}\|^2$$

Learning:

$$\min_{\vec{w}} \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{y}_i(\vec{x}_i, \vec{w})) + \mathcal{R}(\vec{w})$$



# 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.  $P(Y|X=x)$ :
  - $P(Y | \mathbf{X}) \propto P(\mathbf{X} | Y) P(Y)$
- **Indirect** computation
  - Can generate a sample of the data
  - **Can easily handle missing 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

# Naïve Bayes vs. Logistic Regression

[Ng & Jordan, 2002]

- Generative vs. Discriminative classifiers
- Asymptotic comparison  
(# training examples  $\rightarrow$  infinity)
  - when model correct
    - NB, Linear Discriminant Analysis (with class independent variances), and Logistic Regression produce identical classifiers
  - when model incorrect
    - LR is less biased – does not assume conditional independence
      - **therefore LR expected to outperform NB**

# 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
  - Naïve Bayes converges more quickly to its (*perhaps less helpful*) asymptotic estimates

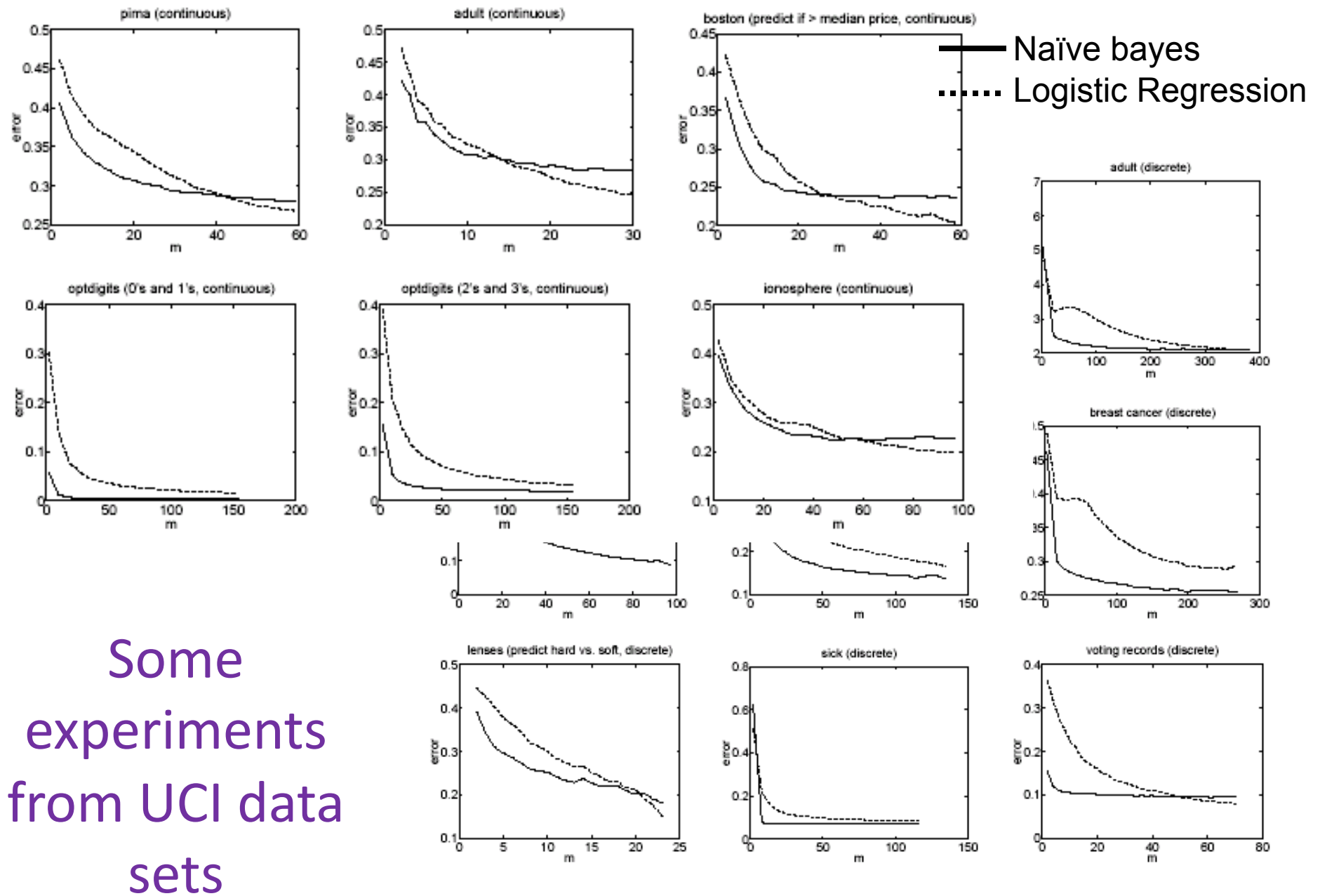


Figure 1: Results of 15 experiments on datasets from the UCI Machine Learning repository. Plots are of generalization error vs.  $m$  (averaged over 1000 random train/test splits). Dashed line is logistic regression; solid line is naive Bayes.

# Logistic regression for discrete classification

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

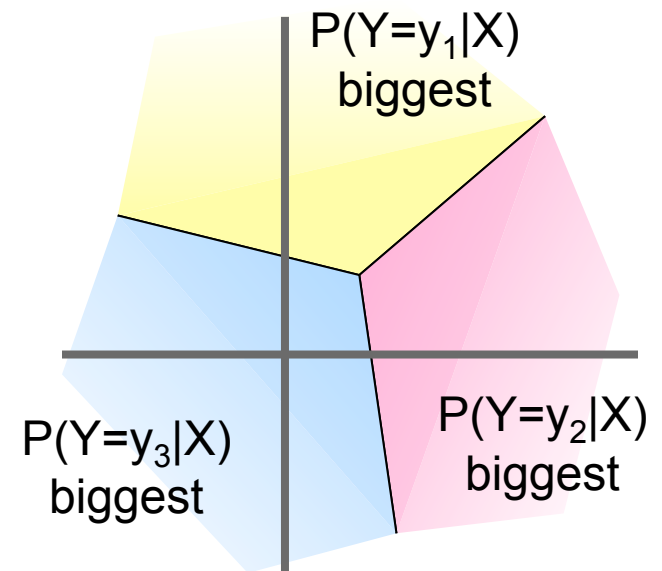
- Define a weight vector  $w_i$  for each  $y_i$ ,  $i=1, \dots, R-1$

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

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

...

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



# Logistic regression for discrete classification

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

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!**