Logistic Regression

Rishabh Iyer
University of Texas at Dallas

based on the slides of Nick Rouzzi and Vibhav Gogate

Last Time

- Supervised learning via naive Bayes
  - Use MLE to estimate a distribution $p(x, y) = p(y)p(x|y)$
  - Classify by looking at the conditional distribution, $p(y|x)$
- Today: logistic regression

Discrete Feat Cont. Feat

$p(y|\theta)$ Dirichlet Dirichlet

$p(x|y, \theta)$ Dirichlet Gaussian

$p(y)$ θ_1 θ_2 -- θ_K.

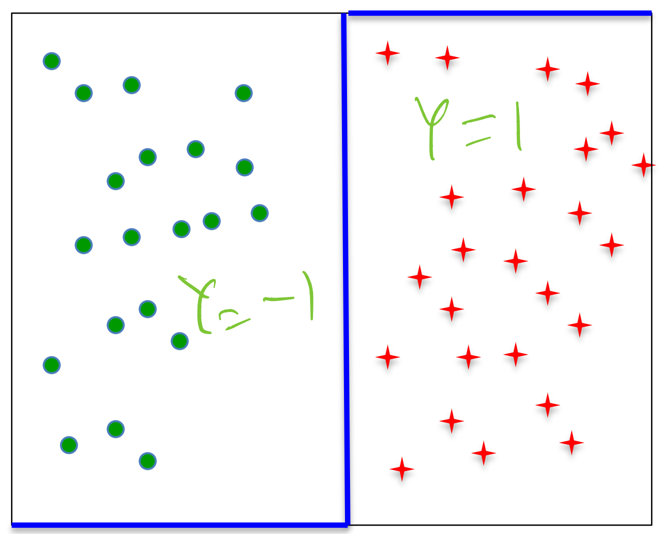
$$\theta_i = \frac{\#\text{ Class } i}{\#\text{ Total}}$$

Logistic Regression

- Learn $p(Y|X)$ directly from the data
  - Assume a particular functional form, e.g., a linear classifier $p(Y = 1|x) = 1$ on one side and 0 on the other
  - Not differentiable…
    - Makes it difficult to learn
    - Can't handle noisy labels

$p(Y=-1|x)=1$

$p(Y = 1|x) = 0$



$p(Y = 1|x) = 1$

$p(Y=-1|x)=0$

Logistic Regression

- Learn $p(y|x)$ directly from the data
  - Assume a particular functional form

$$p(Y=-1|x) = \frac{1}{1+\exp(w^T x + b)}$$

+

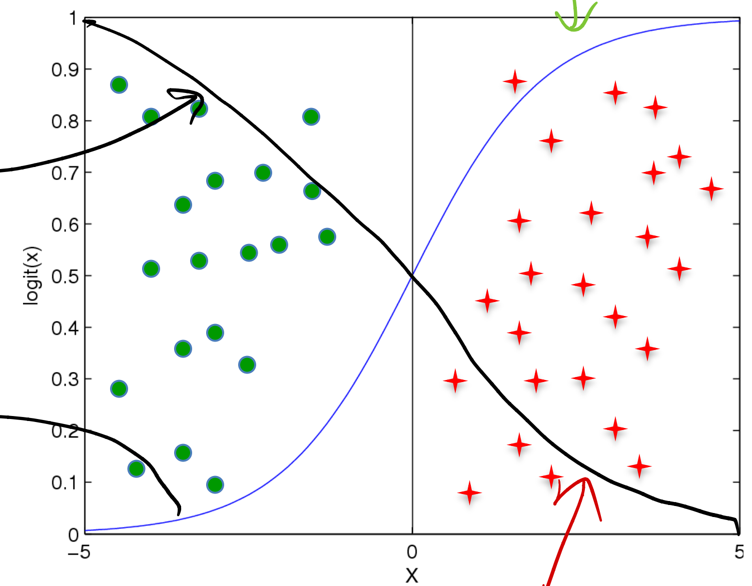
$$p(Y=1|x) = \frac{\exp(w^T x + b)}{1+\exp(w^T x + b)}$$

=

1

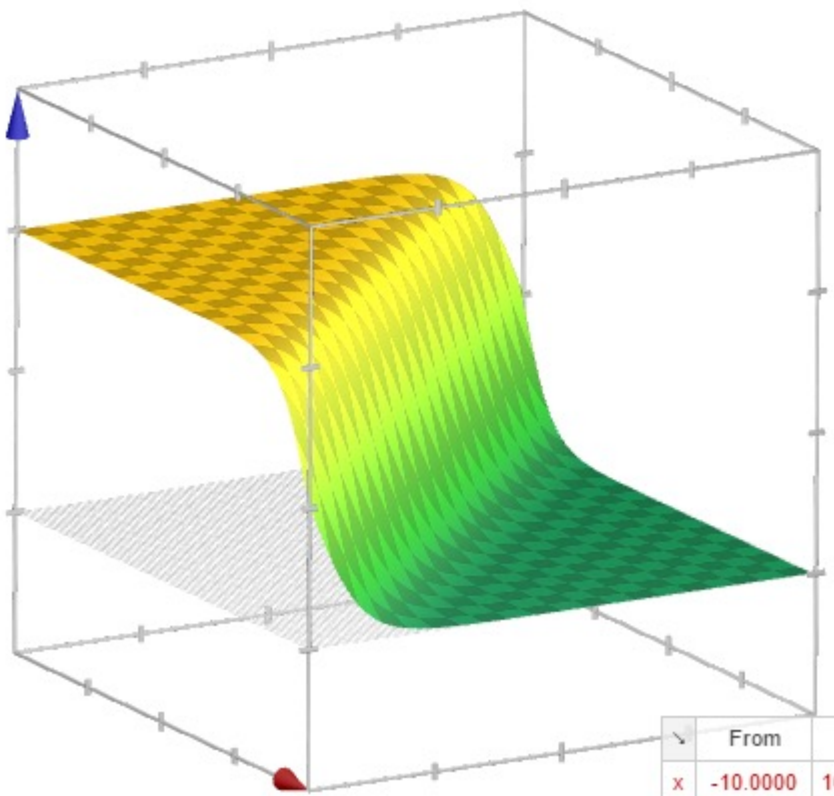
$$\frac{p(Y=1|x)}{p(Y=-1|x)} = \exp(w^T x + b)$$

$p(Y=1|x)$



$p(Y=-1|x)$

Logistic Function in m Dimensions



↘	From	To
x	-10.0000	10.0000
y	-10.0000	10.0000
z	-0.499119	1.49970

$$p(Y = -1|x) = \frac{1}{1 + \exp(w^T x + b)}$$

Can be applied to discrete and continuous features

Discrete → One-hot encoding

Zip Code

$z_1 \; . \; . \; z_{10}$

$[1 \; 0 \; . \; . \; 0]$

Functional Form: Two classes

- Given some w and b, we can classify a new point x by assigning the label 1 if $p(Y = 1|x) > p(Y = -1|x)$ and -1 otherwise
  - This leads to a linear classification rule:
    - Classify as a 1 if $w^T x + b > 0$
    - Classify as a -1 if $w^T x + b < 0$

$$\frac{\exp(w^T x + b)}{\cancel{1 + \exp(w^T x + b)}} > \frac{1}{\cancel{1 + \exp(w^T x + b)}} \Rightarrow w^T x + b > 0$$

$p(Y=1|x)$

$p(Y=-1|x)$

Learning the Weights

$p(y|x)$

- To learn the weights, we maximize the conditional likelihood

$$(w^*, b^*) = \arg\max_{w,b} \prod_{i=1}^{N} \underline{p(y^{(i)}|x^{(i)}}, w, b)$$

- This is the not the same strategy that we used in the case of naive Bayes
  - For naive Bayes, we maximized the log-likelihood

Generative vs. Discriminative Classifiers

Generative classifier:
(e.g., Naïve Bayes)

- Assume some **functional form** for $\underline{p(x|y)}, \underline{p(y)}$
- Estimate parameters of $p(x|y)$, $p(y)$ directly from training data
- Use Bayes rule to calculate $p(y|x) = \frac{p(x,y)}{p(x)} \propto p(x,y) = p(x|y)p(y)$
- 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** $\underline{p(y|x)}$
- Estimate parameters of $p(y|x)$ directly from training data
- This is a discriminative model
  - Directly learn $p(y|x)$
  - But **cannot obtain a sample of the data** as $p(x)$ is not available
  - Useful for discriminating labels

No Generation possible.

Learning the Weights

$$\ell(w, b) = \ln \prod_{i=1}^{N} p(y^{(i)} | x^{(i)}, w, b)$$

$$= \sum_{i=1}^{N} \ln p(y^{(i)} | x^{(i)}, w, b)$$

Learning the Weights

$$\ell(w,b) = \ln \prod_{i=1}^{N} p(y^{(i)}|x^{(i)},w,b)$$

$$= \sum_{i=1}^{N} \ln p(y^{(i)}|x^{(i)},w,b)$$

$$= \sum_{i=1}^{N} \underbrace{\frac{y^{(i)}+1}{2}}_{\begin{cases} =1 & \text{if } y^{(i)}=1 \\ =0 & \text{if } y^{(i)}=-1 \end{cases}} \ln p(Y=1|x^{(i)},w,b) + \underbrace{\left(1-\frac{y^{(i)}+1}{2}\right)}_{\begin{cases} =1 & \text{if } y^{(i)}=-1 \\ =0 & \text{if } y^{(i)}=+1 \end{cases}} \ln p(Y=-1|x^{(i)},w,b)$$

Learning the Weights

$$
\begin{aligned}
\ell(w,b) &= \ln \prod_{i=1}^{N} p(y^{(i)}|x^{(i)},w,b) \\
&= \sum_{i=1}^{N} \ln p(y^{(i)}|x^{(i)},w,b) \\
&= \sum_{i=1}^{N} \frac{y^{(i)}+1}{2} \ln p(Y=1|x^{(i)},w,b) + \left(1-\frac{y^{(i)}+1}{2}\right) \ln p(Y=-1|x^{(i)},w,b) \\
&= \sum_{i=1}^{N} \frac{y^{(i)}+1}{2} \ln \frac{p\left(Y=1\middle|x^{(i)},w,b\right)}{p\left(Y=-1\middle|x^{(i)},w,b\right)} + \ln p(Y=-1|x^{(i)},w,b)
\end{aligned}
$$

Learning the Weights

$$\ell(w,b) = \ln \prod_{i=1}^{N} p(y^{(i)}|x^{(i)},w,b)$$

$$= \sum_{i=1}^{N} \ln p(y^{(i)}|x^{(i)},w,b)$$

$$= \sum_{i=1}^{N} \frac{y^{(i)}+1}{2} \ln p(Y=1|x^{(i)},w,b) + \left(1 - \frac{y^{(i)}+1}{2}\right) \ln p(Y=-1|x^{(i)},w,b)$$

$$= \sum_{i=1}^{N} \frac{y^{(i)}+1}{2} \ln \frac{p\left(Y=1\middle|x^{(i)},w,b\right)}{p\left(Y=-1\middle|x^{(i)},w,b\right)} + \ln p(Y=-1|x^{(i)},w,b)$$

$$= \sum_{i=1}^{N} \frac{y^{(i)}+1}{2} \left(w^T x^{(i)} + b\right) - \ln\left(1 + \exp\left(w^T x^{(i)} + b\right)\right)$$

Learning the Weights

$$
\begin{aligned}
\ell(w,b) &= \ln \prod_{i=1}^{N} p(y^{(i)}|x^{(i)},w,b) \\
&= \sum_{i=1}^{N} \ln p(y^{(i)}|x^{(i)},w,b) \\
&= \sum_{i=1}^{N} \frac{y^{(i)}+1}{2} \ln p(Y=1|x^{(i)},w,b) + \left(1-\frac{y^{(i)}+1}{2}\right) \ln p(Y=-1|x^{(i)},w,b) \\
&= \sum_{i=1}^{N} \frac{y^{(i)}+1}{2} \ln \frac{p\left(Y=1\middle|x^{(i)},w,b\right)}{p\left(Y=-1\middle|x^{(i)},w,b\right)} + \ln p(Y=-1|x^{(i)},w,b) \\
&= \sum_{i=1}^{N} \frac{y^{(i)}+1}{2}\left(w^T x^{(i)}+b\right) - \ln\left(1+\exp\left(w^T x^{(i)}+b\right)\right)
\end{aligned}
$$

This is concave in w and b: take derivatives and solve!

Learning the Weights

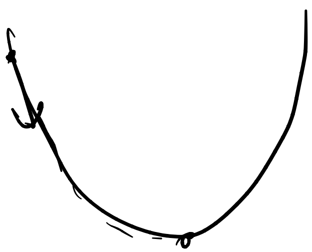
$$
\begin{aligned}
\ell(w,b) &= \ln \prod_{i=1}^{N} p(y^{(i)}|x^{(i)},w,b)\\
&= \sum_{i=1}^{N} \ln p(y^{(i)}|x^{(i)},w,b)\\
&= \sum_{i=1}^{N} \frac{y^{(i)}+1}{2} \ln p(Y=1|x^{(i)},w,b) + \left(1-\frac{y^{(i)}+1}{2}\right) \ln p(Y=-1|x^{(i)},w,b)\\
&= \sum_{i=1}^{N} \frac{y^{(i)}+1}{2} \ln \frac{p\left(Y=1\middle|x^{(i)},w,b\right)}{p\left(Y=-1\middle|x^{(i)},w,b\right)} + \ln p(Y=-1|x^{(i)},w,b)\\
&= \sum_{i=1}^{N} \frac{y^{(i)}+1}{2}\left(w^T x^{(i)}+b\right) - \ln\left(1+\exp\left(w^T x^{(i)}+b\right)\right)
\end{aligned}
$$

No closed form solution ☹

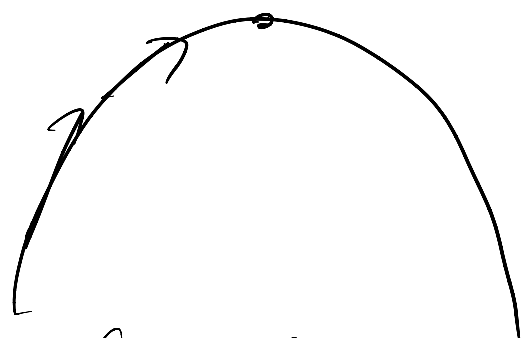
Likelihood Maximization

$$\ell(w,b) \qquad = \sum_{i=1}^{N} \frac{y^{(i)}+1}{2}\left(w^T x^{(i)} + b\right) - \ln\left(1 + \exp\left(w^T x^{(i)} + b\right)\right)$$

The above is the Likelihood which we maximize!



Convex minimization



Concave maximization

Learning the Weights

- Can apply gradient ascent to maximize the conditional likelihood

$$\frac{\partial \ell}{\partial b} = \sum_{i=1}^{N} \left[\frac{y^{(i)} + 1}{2} - p(Y = 1 | x^{(i)}, w, b) \right]$$

$$\frac{\partial \ell}{\partial w_j} = \sum_{i=1}^{N} x_j^{(i)} \left[\frac{y^{(i)} + 1}{2} - p(Y = 1 | x^{(i)}, w, b) \right]$$

Gradient Ascent

w^0, b^0 Initialize.

For $t = 1 : T$

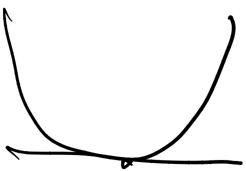
$$w^{t+1} \leftarrow w^t + r_t \nabla_w L$$

$$b^{t+1} \leftarrow b^t + r_t \nabla_b L$$

end For.

Stopping Criteria for Gradient Descent / Ascent

1. Fixed Number of Iterations : T
2. $\| DL(w^t) \| \leq \varepsilon$
3. $\| w^{t+1} - w^t \| \leq \varepsilon$
4. $| L(w^{t+1}) - L(w^t) | \leq \varepsilon$



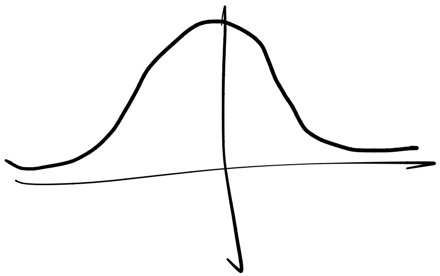
$DL \simeq 0$

Priors

- Can define priors on the weights to prevent overfitting
  - Normal distribution, zero mean, identity covariance

$$p(w) = \prod_j \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{w_j^2}{2\sigma^2}\right)$$

  - "Pushes" parameters towards zero



- Regularization
  - Helps avoid very large weights and overfitting

Priors as Regularization

- The log-MAP objective with this Gaussian prior is then

$$\ln \prod_{i=1}^{N} p\left(y^{(i)} \middle| x^{(i)}, w, b\right) p(w) p(b) = \left[\sum_{i}^{N} \ln p\left(y^{(i)} \middle| x^{(i)}, w, b\right)\right] - \frac{\lambda}{2} \|w\|_2^2 - \frac{\lambda}{2} b^2$$

  - Quadratic penalty: drives weights towards zero
  - Adds a negative linear term to the gradients
  - Different priors can produce different kinds of regularization

Priors as Regularization

- The log-MAP objective with this Gaussian prior is then

$$\ln \prod_{i=1}^{N} p\left(y^{(i)} \middle| x^{(i)}, w, b\right) p(w) p(b) = \left[\sum_{i}^{N} \ln p\left(y^{(i)} \middle| x^{(i)}, w, b\right)\right] - \frac{\lambda}{2} \|w\|_2^2$$

ℓ_2 regularizer

  - Quadratic penalty: drives weights towards zero
  - Adds a negative linear term to the gradients
  - Different priors can produce different kinds of regularization

$$\min_{w} L(w) + \frac{\lambda}{2} \|w\|^2$$

L2 vs L1 Regularization

The Likelihood with L2 Regularization:

$$\ln \prod_{i=1}^{N} p\left(y^{(i)} \middle| x^{(i)}, w, b\right) p(w) p(b) = \left[\sum_{i}^{N} \ln p\left(y^{(i)} \middle| x^{(i)}, w, b\right)\right] - \frac{\lambda}{2} \|w\|_2^2$$

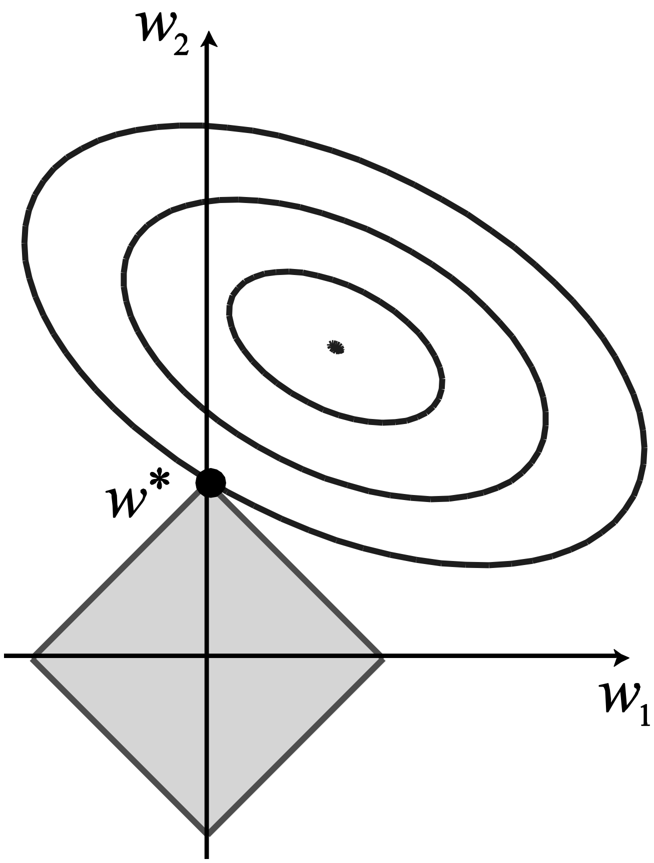
Alternate formulation is L1 Regularization:

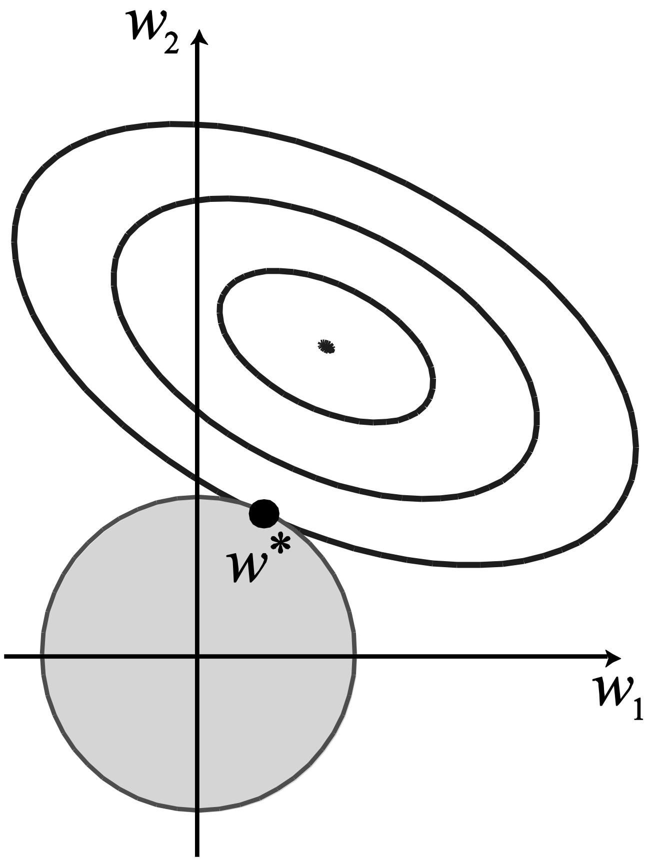
$$\ln \prod_{i=1}^{N} p\left(y^{(i)} \middle| x^{(i)}, w, b\right) p(w) p(b) = \left[\sum_{i}^{N} \ln p\left(y^{(i)} \middle| x^{(i)}, w, b\right)\right] - \frac{\lambda}{2} \|w\|_1$$

$$\|w\|_2^2 = \left[w_1^2 + w_2^2 + \ldots + w_d^2\right]$$

$$\|w\|_1 = |w_1| + |w_2| + \ldots + |w_d|$$

Regularization





ℓ_1 → Sparsity

ℓ_2

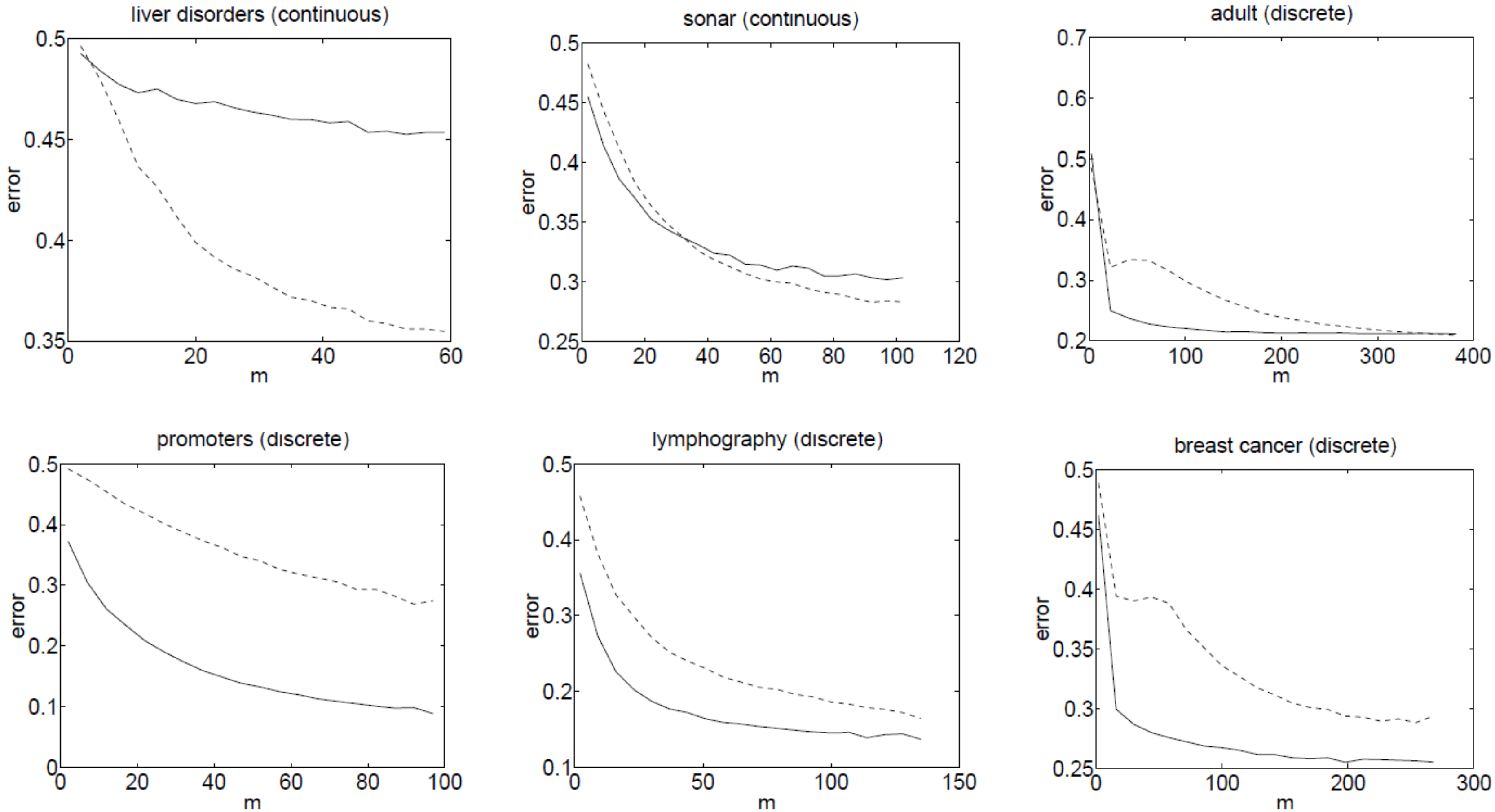
Naïve Bayes vs. Logistic Regression

- Non-asymptotic analysis (for Gaussian NB)
  - Convergence rate of parameter estimates as size of training data tends to infinity (n = # of attributes in X) n = # Features
    - Naïve Bayes needs $O(\log n)$ samples
      - NB converges quickly to its (perhaps less helpful) asymptotic estimates → Cond. Independence
    - Logistic Regression needs $O(n)$ samples
      - LR converges more slowly but makes no independence assumptions (typically less biased)

  ↳ Linear sep

[Ng & Jordan, 2002]

NB vs. LR (on UCI datasets)



Naïve bayes
Logistic Regression

Sample size m

[Ng & Jordan, 2002]

LR in General

- Suppose that $y \in \{1, \dots, R\}$, i.e., that there are R different class labels

- Can define a collection of weights and biases as follows
  - Choose a vector of biases and a matrix of weights such that for $y \neq R$

(w_k, b_k)

$$p(Y = k|x) = \frac{\exp(b_k + \sum_i w_{ki} x_i)}{1 + \sum_{j<R} \exp\left(b_j + \sum_i w_{ji} x_i\right)}$$

$w_k^T x$

and

$$p(Y = R|x) = \frac{1}{1 + \sum_{j<R} \exp\left(b_j + \sum_i w_{ji} x_i\right)}$$

Cond Log Likelihood $D = \{(x^{(1)}, y^{(1)}), \ldots (x^{(N)}, y^{(N)})\}$

$$\log \prod_{i=1}^{N} p(y^{(i)} \mid x^{(i)}, w, b)$$

$$= \sum_{i=1}^{N} \log p(y^{(i)} \mid x^{(i)}, w, b)$$

$$= \sum_{i=1}^{N} \left[\sum_{k=1}^{R-1} I(y^{(i)} = k) \log \frac{\exp(b_k + \sum_j w_{kj} x_j^{(i)})}{\ldots} \right.$$

$$\left. + \; I(y^{(i)} = R) \ldots \right]$$