

Logistic Regression

Sargur N. Srihari

University at Buffalo, State University of New York
USA

Topics in Linear Classification using Probabilistic Discriminative Models

- Generative vs Discriminative
 1. Fixed basis functions
 2. Logistic Regression (two-class)
 3. Iterative Reweighted Least Squares (IRLS)
 4. Multiclass Logistic Regression
 5. Probit Regression
 6. Canonical Link Functions

Topics in Logistic Regression

- Logistic Sigmoid and Logit Functions
- Parameters in discriminative approach
- Determining logistic regression parameters
 - Error function
 - Gradient of error function
 - Simple sequential algorithm
 - An example

Logistic Sigmoid and Logit Functions

- In two-class case, *posterior* of class C_1 can be written as as a logistic sigmoid of feature vector ϕ

$$p(C_1|\phi) = y(\phi) = \sigma(w^T\phi)$$

with $p(C_2|\phi) = 1 - p(C_1|\phi)$

Here $\sigma(\cdot)$ is the logistic sigmoid function

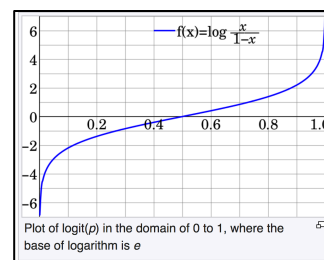
– Known as logistic regression^w in statistics

- Although a model for classification rather than for regression

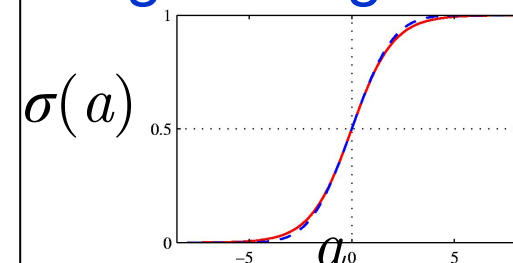
- Logit function:

– It is the log of the odds ratio

- It links the probability to the predictor variables



Logistic Sigmoid



Properties:

A. Symmetry

$$\sigma(-a) = 1 - \sigma(a)$$

B. Inverse

$$a = \ln(\sigma / (1 - \sigma))$$

known as *logit*.

Also known as *log odds* since it is the ratio

$$\ln[p(C_1|x) / p(C_2|x)]$$

C. Derivative

$$d\sigma / da = \sigma(1 - \sigma)$$

Fewer Parameters in Linear Discriminative Model

- Discriminative approach (Logistic Regression)
 - For M -dim feature space ϕ :
 - M adjustable parameters
- Generative based on Gaussians (Bayes/NB)
 - $2M$ parameters for mean
 - $M(M+1)/2$ parameters for shared covariance matrix
 - Two class priors
 - Total of $M(M+5)/2 + 1$ parameters
 - Grows quadratically with M
 - If features assumed independent (naïve Bayes) still needs $M+3$ parameters

Determining Logistic Regression parameters

- Maximum Likelihood Approach for Two classes
- For a data set (ϕ_n, t_n) where $t_n \in \{0,1\}$ and $\phi_n = \phi(\mathbf{x}_n)$, $n = 1, \dots, N$
- Likelihood function can be written as

$$p(\mathbf{t} | \mathbf{w}) = \prod_{n=1}^N y_n^{t_n} \{1 - y_n\}^{1-t_n}$$

where $\mathbf{t} = (t_1, \dots, t_N)^T$ and $y_n = p(C_1 | \phi_n)$

y_n is the probability that $t_n = 1$

Error Fn for Logistic Regression

- Likelihood function is

$$p(t | \mathbf{w}) = \prod_{n=1}^N y_n^{t_n} \{1 - y_n\}^{1-t_n}$$

- By taking negative logarithm we get the *Cross-entropy Error Function*

$$E(\mathbf{w}) = -\ln p(t | \mathbf{w}) = -\sum_{n=1}^N \left\{ t_n \ln y_n + (1 - t_n) \ln(1 - y_n) \right\}$$

where $y_n = \sigma(a_n)$ and $a_n = \mathbf{w}^T \boldsymbol{\phi}_n$

- We need to minimize $E(\mathbf{w})$

At its minimum, derivative of $E(\mathbf{w})$ is zero

So we need to solve for \mathbf{w} in the equation

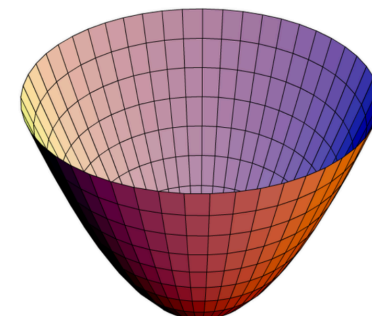
$$\nabla E(\mathbf{w}) = 0$$

Gradient of Error Function

Error function

$$E(\mathbf{w}) = -\ln p(t | \mathbf{w}) = -\sum_{n=1}^N \left\{ t_n \ln y_n + (1 - t_n) \ln(1 - y_n) \right\}$$

where $y_n = \sigma(\mathbf{w}^T \boldsymbol{\phi}_n)$



Using Derivative of logistic sigmoid $d\sigma/da = \sigma(1-\sigma)$

Gradient of the error function

$$\nabla E(\mathbf{w}) = \sum_{n=1}^N \underbrace{(y_n - t_n)}_{\text{Error}} \boldsymbol{\phi}_n$$

Error x Feature Vector

Contribution to gradient by data point n is error between target t_n and prediction $y_n = \sigma(\mathbf{w}^T \boldsymbol{\phi}_n)$ times basis $\boldsymbol{\phi}_n$

Proof of gradient expression

Let $z = z_1 + z_2$

where $z_1 = t \ln \sigma(w\phi)$ and $z_2 = (1 - t) \ln[1 - \sigma(w\phi)]$

$$\frac{dz_1}{dw} = \frac{t \sigma(w\phi) [1 - \sigma(w\phi)] \phi}{\sigma(w\phi)} \quad \frac{d\sigma}{da} = \sigma(1 - \sigma)$$

and

$$\frac{dz_2}{dw} = \frac{(1 - t) \sigma(w\phi) [1 - \sigma(w\phi)] (-\phi)}{[1 - \sigma(w\phi)]} \quad \text{Using } \frac{d}{dx}(\ln ax) = \frac{a}{x}$$

Therefore $\frac{dz}{dw} = (\sigma(w\phi) - t) \phi$

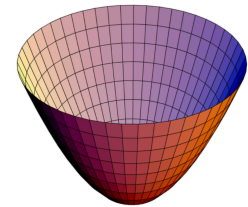
Simple Sequential Algorithm

- Given Gradient of error function

$$\nabla E(\mathbf{w}) = \sum_{n=1}^N (y_n - t_n) \phi_n \quad \text{where } y_n = \sigma(\mathbf{w}^T \phi_n)$$

- Solve using an iterative approach

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} - \eta \nabla E_n$$



- where

$$\nabla E_n = \underbrace{(y_n - t_n) \phi_n}_{\text{Error x Feature Vector}}$$

Error x Feature Vector

Takes precisely same form as
Gradient of Sum-of-squares
error for linear regression

Samples are presented one at a time in
which each each of the weight vectors is updated

ML solution can over-fit

- Severe over-fitting for linearly separable data

- Because ML solution occurs at $\sigma = 0.5$

- With $\sigma > 0.5$ and $\sigma < 0.5$ for the two classes

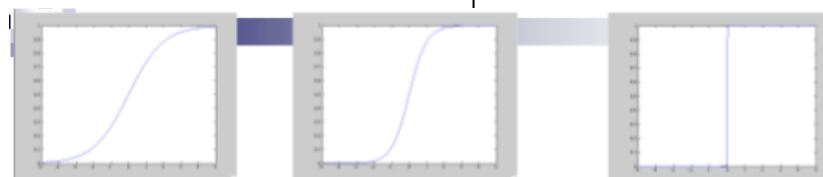
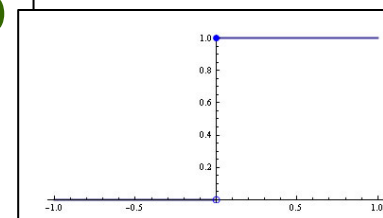
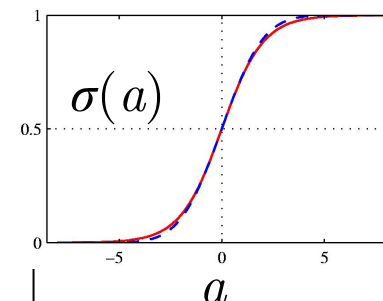
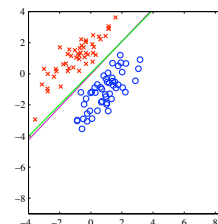
- Solution equivalent to $a = w^T \phi = 0$

- Logistic sigmoid becomes infinitely steep

- A Heavyside step function

- $\|w\|$ goes to infinity

- Penalizing wts can avoid this



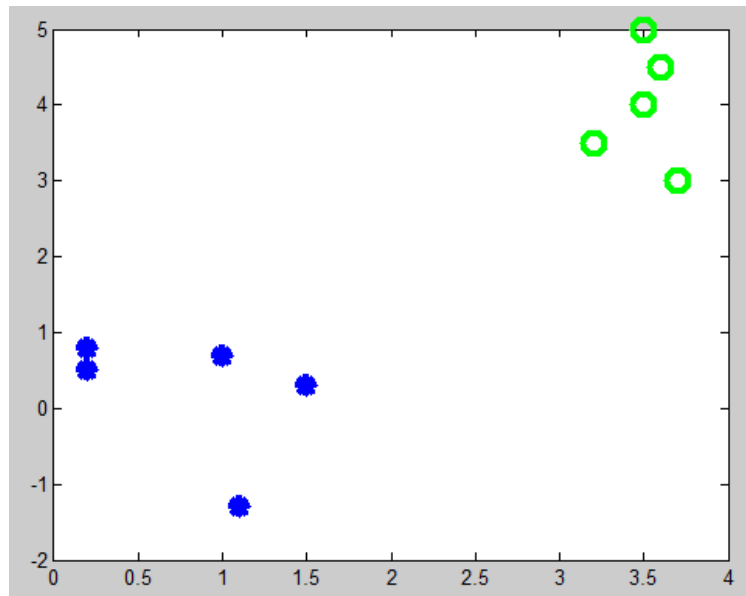
$\frac{1}{1 + e^{-x}}$	$\frac{1}{1 + e^{-2x}}$	$\frac{1}{1 + e^{-100x}}$
------------------------	-------------------------	---------------------------

An Example of 2-class Logistic Regression

- Input Data

C1 =	
3.7000	3.0000
3.2000	3.5000
3.5000	5.0000
3.6000	4.5000
3.5000	4.0000

C2 =	
1.1000	-1.3000
0.2000	0.5000
1.5000	0.3000
0.2000	0.8000
1.0000	0.7000



$\phi_0(\mathbf{x})=1$, dummy feature

Initial Weight Vector, Gradient and Hessian (2-class)

- Weight vector

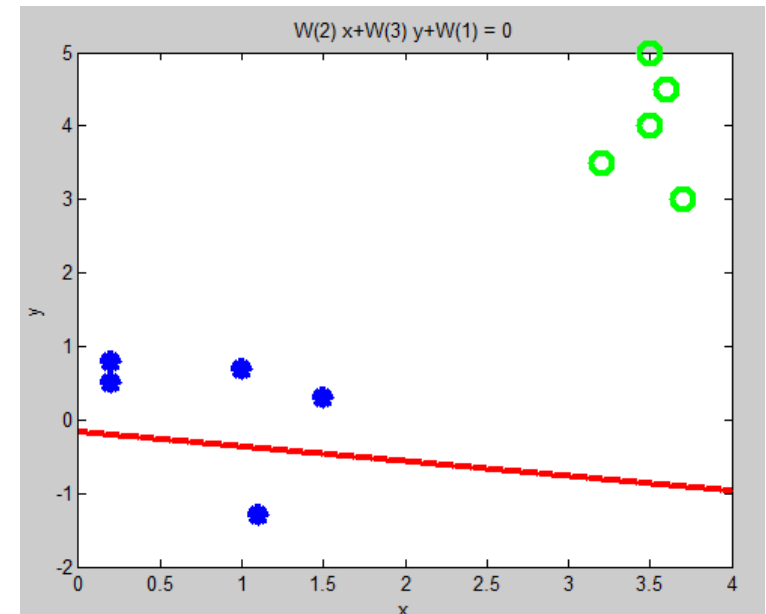
```
W =
    0.1117
    0.1363
    0.6787
```

- Gradient

```
Delta_E =
           0
    6.7500
    9.5000
```

- Hessian

```
H =
    3.5000    5.3750    5.2500
    5.3750   17.4825   17.4950
    5.2500   17.4950   22.4150
```



Final Weight Vector, Gradient and Hessian (2-class)

- Weight Vector

W =

704.5915
-20.9086
-337.6170

- Gradient

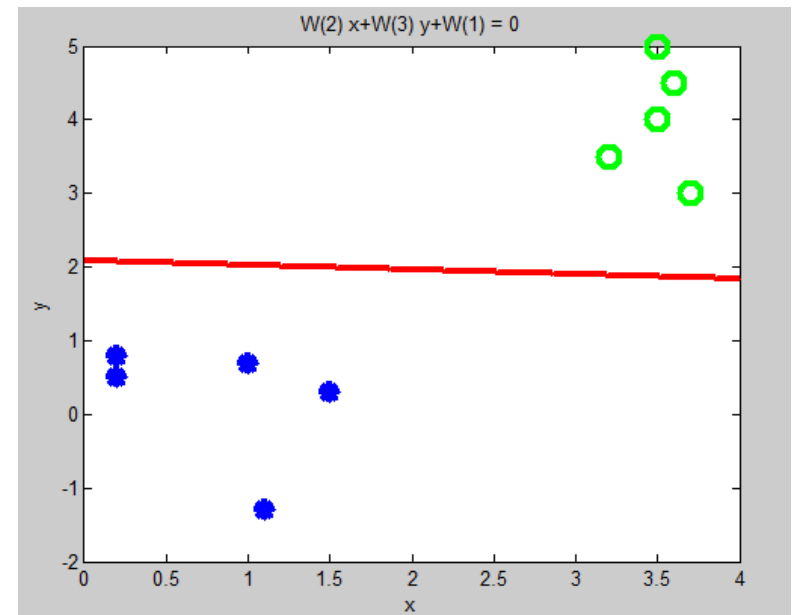
Delta_E =

-12.3917
-1.6321
4.9025

- Hessian

H =

1.0000	0.0000	0.0000
0.0000	1.0000	0.0000
0.0000	0.0000	1.0000



Number of iterations : 10

Error (Initial and Final): 15.0642, 1.0000e-009

Logistic Regression is a special architecture of a neural network

