MACHINE LEARNING

CHAPTER 3: LOGISTIC REGRESSION

LOGISTIC REGRESSION

1. The concept of logistic regression
2. Optimizing-Newton method
3. Bayesian logistic regression

Classification

Learn a function: $\mathbf{x} \longrightarrow p\left(t = k|\mathbf{x}\right)$

- $\boldsymbol{x}$ - features
- t - target classes

Suppose you know $p\left(t|\mathbf{x}\right)$ exactly, how should you classify?

$$\mathrm{argmax}_k p\left(t = k|\mathbf{x}\right)$$

Logistic regression

For two-class classification problems, a target coding scheme:

$$t = 1, \mathbf{x} \in \mathcal{C}_1$$

$$t = 0, \mathbf{x} \in \mathcal{C}_2$$

Target values is in $\{0, 1\}$.

Logistic regression

Learn $p\left(t|\mathbf{x}\right)$ directly!

Assume a particular function form

Sigmoid applied to a linear function of the data:

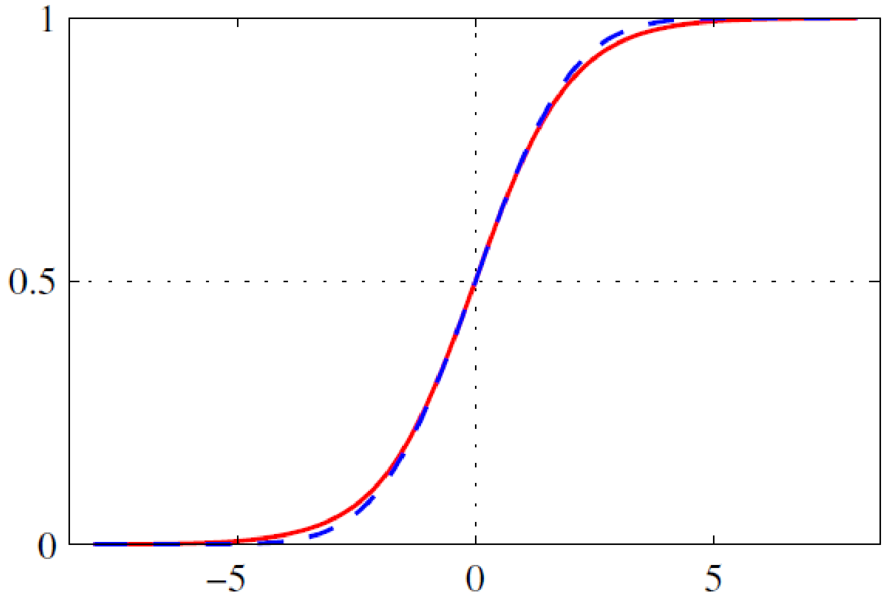
$$p\left(t=1|\mathbf{x}\right)=\sigma\left(\mathbf{w}^{T}\boldsymbol{\phi}\left(\mathbf{x}\right)\right)=\frac{1}{1+exp\left(-\mathbf{w}^{T}\boldsymbol{\phi}\left(\mathbf{x}\right)\right)}$$

$$p\left(t=0|\mathbf{x}\right)=1-p\left(t=1|\mathbf{x}\right)$$

Features can be discrete or continuous!

Logistic sigmoid function

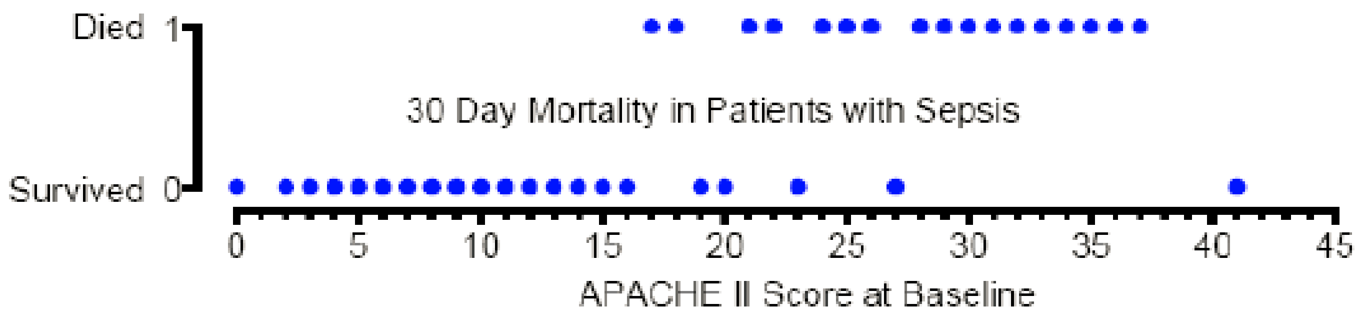
$$\sigma\left(a\right) = \frac{1}{1 + exp\left(-a\right)}$$



Examples

a) Example: APACHE II Score and Mortality in Sepsis

The following figure shows 30 day mortality in a sample of septic patients as a function of their baseline APACHE II Score. Patients are coded as 1 or 0 depending on whether they are dead or alive in 30 days, respectively.

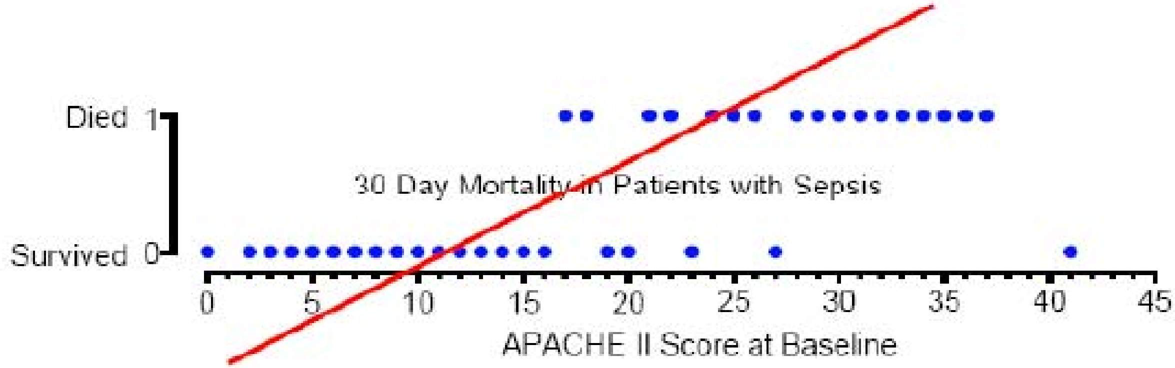


Examples

We wish to predict death from baseline APACHE II score in these patients.

Let $\pi(x)$ be the probability that a patient with score x will die.

Note that linear regression would not work well here since it could produce probabilities less than zero or greater than one.



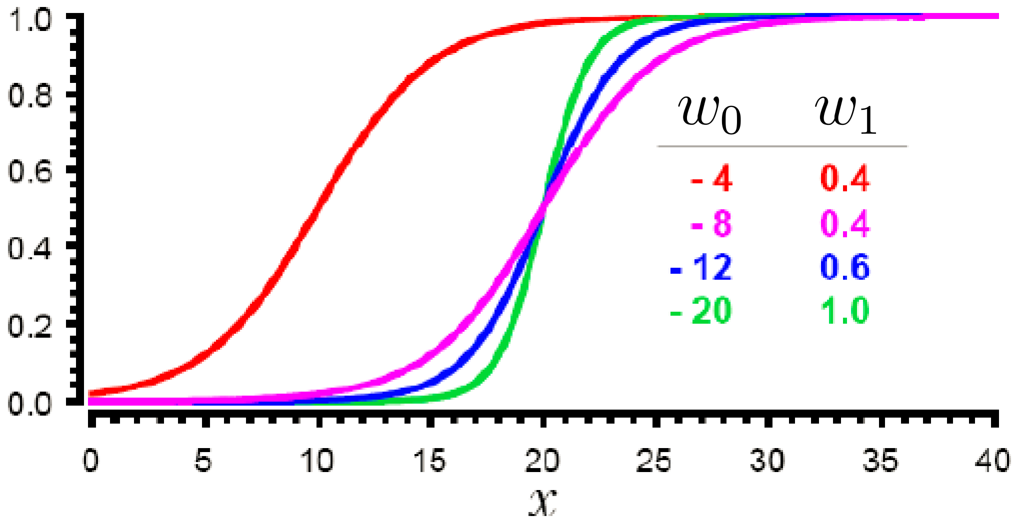
Examples

Parameters control shape and location of sigmoid curve

– w_0 controls location of midpoint

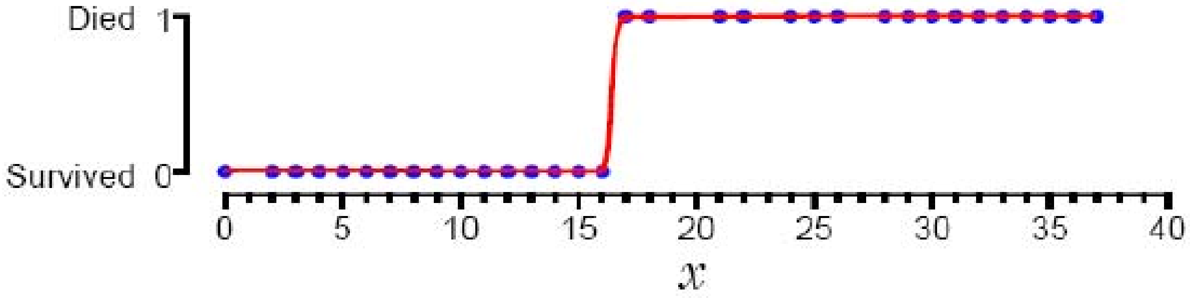
– w_1 controls slope of rise

$$p\left(t=1|\mathbf{x}\right)=\frac{1}{1+exp\left(-w_0-w_1x\right)}$$

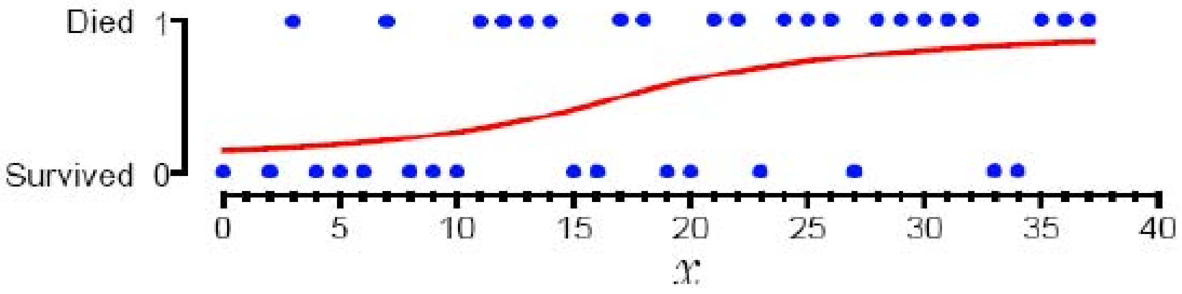


Examples

Data that has a sharp survival cut off point between patients who live or die should have a large value of w_1

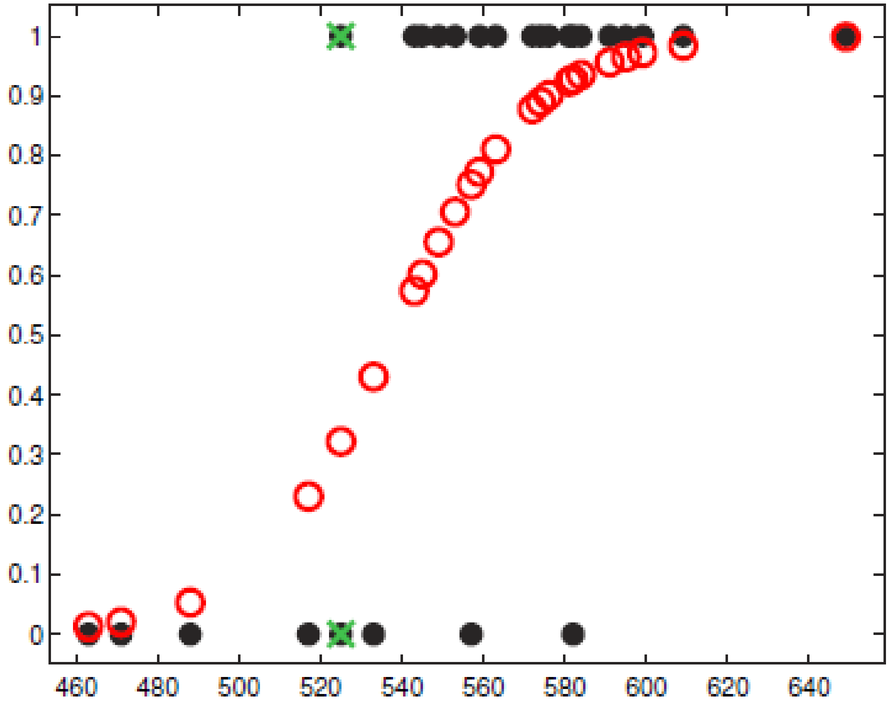


Data with a lengthy transition from survival to death should have a low value of w_1



Logistic regression

SAT score vs. being admitted to MIT



Logistic regression

Given the SAT score x

being admitted to MIT y=1

Not being admitted to MIT y=0

We choose

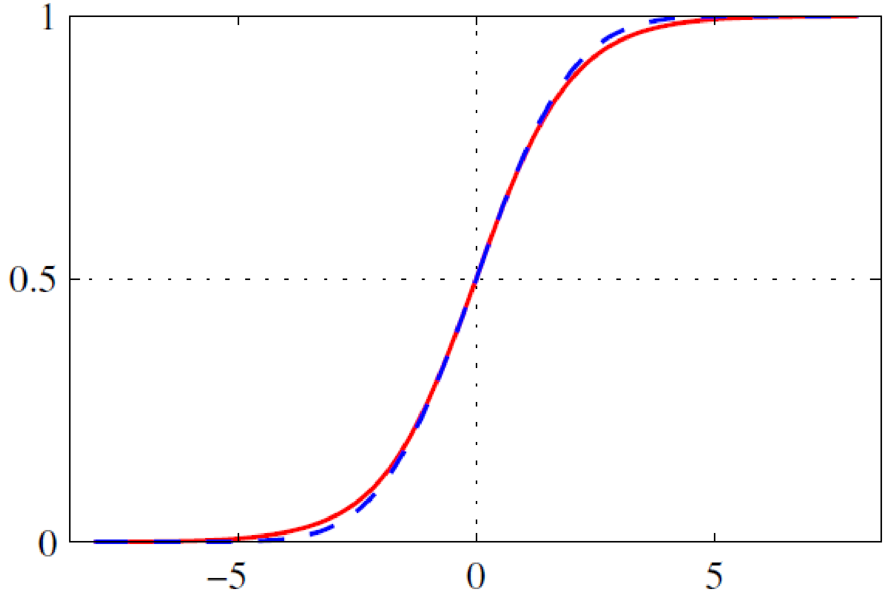
$$\boldsymbol{\phi}(\mathbf{x}) = \begin{pmatrix} 1 \\ \mathrm{x} \end{pmatrix} \qquad \boldsymbol{w} = \begin{pmatrix} \mathrm{w}_0 \\ \mathrm{w}_1 \end{pmatrix}$$

$$\boldsymbol{w}^T \boldsymbol{\phi}(\mathbf{x}) = w_0 + w_1 x$$

Logistic regression

If $w_0 + w_1 x > 0, \sigma(w_0 + w_1 x) > 0.5$

we have $p(t = 1|x) > 0.5$, x is prone to be admitted to MIT.



Logistic regression – a linear classifier

$$\text{if } p(t=1|\mathbf{x}) > 0.5,\ \ \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}) > 0, \mathbf{x} \in \mathcal{C}_1$$

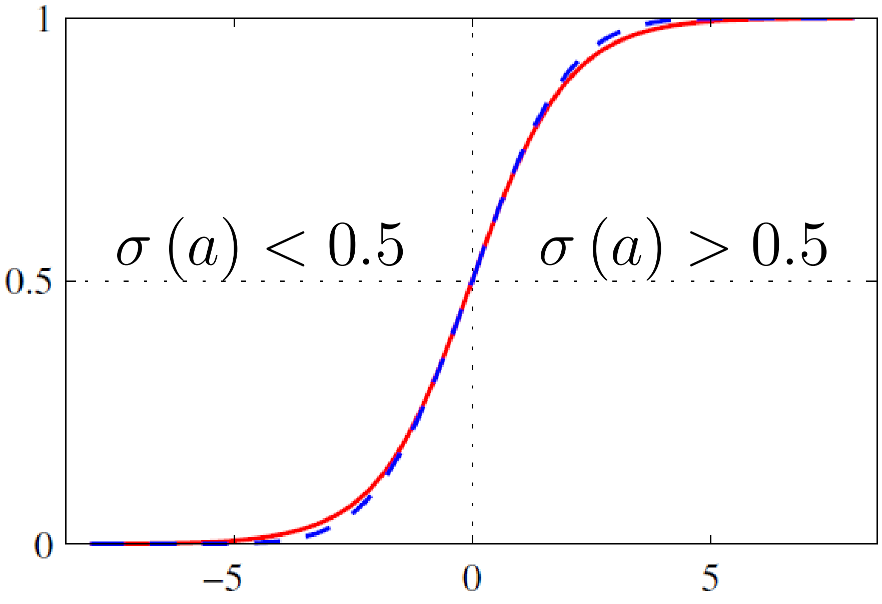
$$\text{if } p(t=1|\mathbf{x}) < 0.5,\ \ \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}) < 0, \mathbf{x} \in \mathcal{C}_2$$

We have:

$$p(t=1|\mathbf{x}) = \sigma\left(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x})\right)$$

$$= \frac{1}{1+exp\left(-\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x})\right)}$$

$$p(t=0|\mathbf{x}) = 1 - p(t=1|\mathbf{x})$$



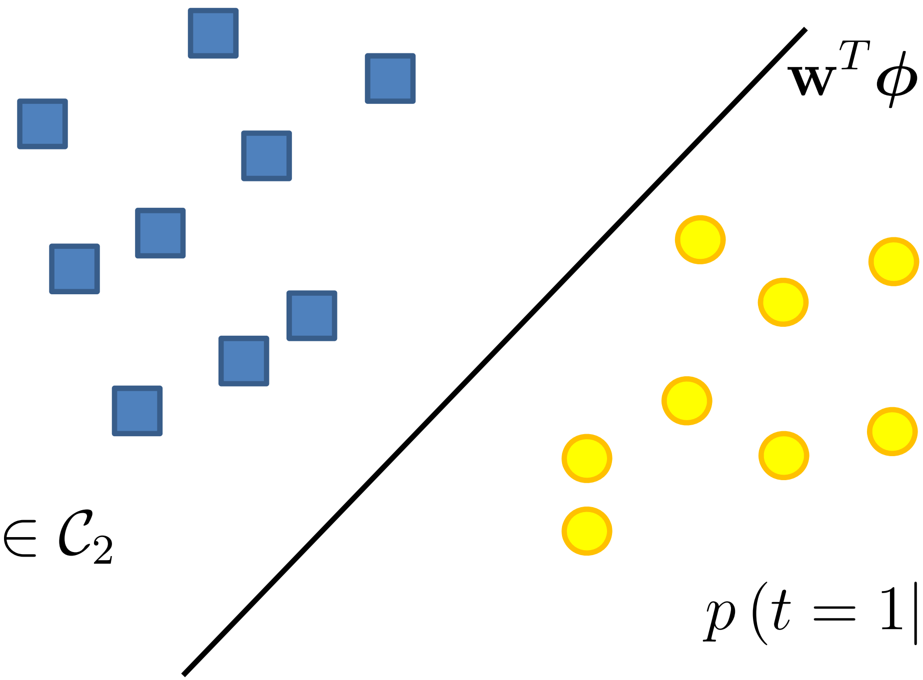
Logistic regression – a linear classifier

$p\left(t=1|\mathbf{x},\mathbf{w}\right)<0.5$

$\mathbf{w}^T\boldsymbol{\phi}\left(\mathbf{x}\right)>0, \mathbf{x}\in\mathcal{C}_1$

predict $t=0$

predict $t=1$



$\mathbf{w}^T\boldsymbol{\phi}\left(\mathbf{x}\right)<0, \mathbf{x}\in\mathcal{C}_2$

$p\left(t=1|\mathbf{x},\mathbf{w}\right)>0.5$

$\mathbf{w}^T\boldsymbol{\phi}\left(\mathbf{x}\right)=0, \text{hyperplane}$

Logistic regression – a linear classifier

The goal of logistic regression is to learn the weights of a linear classifier!

$$\boxed{\begin{aligned}&\text{if } p\left(t=1|\mathbf{x}\right) > 0.5,\ \mathbf{w}^T\boldsymbol{\phi}\left(\mathbf{x}\right) > 0, \mathbf{x} \in \mathcal{C}_1\\&\text{if } p\left(t=1|\mathbf{x}\right) < 0.5,\ \mathbf{w}^T\boldsymbol{\phi}\left(\mathbf{x}\right) < 0, \mathbf{x} \in \mathcal{C}_2\end{aligned}}$$

Logistic regression for more than 2 classes

Logistic regression in more general cases,

where $t \in \{1, ..., K\}$

3 classes: $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$

$$p\left(t = 1 | \mathbf{x}\right) \propto exp\left\{\mathbf{w}_1^T \boldsymbol{\phi}\left(\mathbf{x}\right)\right\}$$

$$p\left(t = 2 | \mathbf{x}\right) \propto exp\left\{\mathbf{w}_2^T \boldsymbol{\phi}\left(\mathbf{x}\right)\right\}$$

$$p\left(t = 3 | \mathbf{x}\right) = 1 - p\left(t = 1 | \mathbf{x}\right) - p\left(t = 2 | \mathbf{x}\right)$$

Logistic regression more generally

Logistic regression in more general case, where $t \in \{1, ..., K\}$

For $k < K$

$$p\left(t = k|\mathbf{x}\right) = \frac{exp\left\{\mathbf{w}_k^T \boldsymbol{\phi}\left(\mathbf{x}\right)\right\}}{1 + \sum_{j=1}^{K-1} exp\left\{\mathbf{w}_j^T \boldsymbol{\phi}\left(\mathbf{x}\right)\right\}}$$

For $k = K$ (normalization, so no weights for this class)

$$p\left(t = k|\mathbf{x}\right) = \frac{1}{1 + \sum_{j=1}^{K-1} exp\left\{\mathbf{w}_j^T \boldsymbol{\phi}\left(\mathbf{x}\right)\right\}}$$

Loss function: conditional likelihood

Data likelihood

$$\ln p(\mathcal{D}|\mathbf{w}) = \sum_{n=1}^{N} \ln p(\mathbf{x}_n, t_n|\mathbf{w})$$
$$= \sum_{n=1}^{N} \ln p(t_n|\mathbf{x}_n, \mathbf{w}) + \sum_{n=1}^{N} \ln p(\mathbf{x}_n|\mathbf{w})$$

Discriminative model can not compute $p(\mathbf{x}_n|\mathbf{w})$.

Loss function: conditional likelihood

Conditional data likelihood:

$$\ln p\left(D_Y \middle| D_X, \mathbf{w}\right) = \sum_{n=1}^{N} \ln p\left(t_n \middle| \mathbf{x}_n, \mathbf{w}\right)$$

Doesn’t waste effort learning $p\left(D_X\right)$

Loss function: conditional likelihood

The conditional likelihood:

$$l\left(\mathbf{w}\right)=\sum_{n}\ln p\left(t_n|\mathbf{x}_n,\mathbf{w}\right)$$

As we know:

$$p\left(t_n=1|\mathbf{x}_n,\mathbf{w}\right)=\frac{1}{1+exp\left(-\mathbf{w}^T\boldsymbol{\phi}\left(\mathbf{x}_n\right)\right)}$$

$$p\left(t_n=0|\mathbf{x}_n,\mathbf{w}\right)=\frac{exp\left(-\mathbf{w}^T\boldsymbol{\phi}\left(\mathbf{x}_n\right)\right)}{1+exp\left(-\mathbf{w}^T\boldsymbol{\phi}\left(\mathbf{x}_n\right)\right)}$$

Loss function: conditional likelihood

The conditional likelihood:

$$l(\mathbf{w}) = \sum_n t_n \ln p(t_n = 1|\mathbf{x}_n, \mathbf{w}) + (1 - t_n) \ln p(t_n = 0|\mathbf{x}_n, \mathbf{w})$$

$$= \sum_n -(1 - t_n)\left(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\right) - \ln\left(1 + \exp\left(-\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\right)\right)$$

$E(\mathbf{w}) = -l(\mathbf{w})$ is a convex function of **w**!

No closed-form solution to maximize $E(\mathbf{w})$

Convex function is easy to optimize.

LOGISTIC REGRESSION

1. The concept of logistic regression
2. Optimizing-Newton method
3. Bayesian logistic regression

Optimizing-Newton method

The Newton-Raphson method:

$$\boxed{\mathbf{w}^{(\mathrm{new})} = \mathbf{w}^{(\mathrm{old})} - \mathbf{H}^{-\mathbf{1}} \nabla E\left(\mathbf{w}\right)}$$

Where **H** is the Hessian matrix whose element comprise the second derivatives of E(**w**)

Linear regression model revisited

The objective function of linear regression model:

$$E\left(\mathbf{w}\right)=\frac{1}{2}\sum_{n=1}^{N}\left\{\mathbf{w}^{T}\boldsymbol{\phi}\left(\mathbf{x}_{\boldsymbol{n}}\right)-t_{n}\right\}^{2}$$

can be rewritten as

$$E\left(\mathbf{w}\right)=\frac{1}{2}\left\|\mathbf{t}-\mathbf{\Phi}\mathbf{w}\right\|_{2}^{2}$$

Linear regression model revisited

The gradient and Hessian of the objective function in linear regression are given by

$$\nabla E\left(\mathbf{w}\right) = \mathbf{\Phi}^T\mathbf{\Phi}\mathbf{w} - \mathbf{\Phi}^T\mathbf{t}$$

$$\mathbf{H} = \nabla^2 E\left(\mathbf{w}\right) = \mathbf{\Phi}^T\mathbf{\Phi}$$

Linear regression model revisited

The Newton-Raphson update

$$\mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - \left(\mathbf{\Phi}^{\mathbf{T}}\mathbf{\Phi}\right)^{\mathbf{-1}} \left\{\mathbf{\Phi}^{T}\mathbf{\Phi}\mathbf{w}^{(\text{old})} - \mathbf{\Phi}^{T}\mathbf{t}\right\}$$

$$= \left(\mathbf{\Phi}^{T}\mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^{T}\mathbf{t}$$

which is the same as the standard least-squares solution.

Optimizing-Newton method

$$E(\mathbf{w}) = \sum_{n} (1 - t_n)\left(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\right) + \ln\left(1 + exp\left(-\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\right)\right)$$

The gradient of the error function in logistic regression

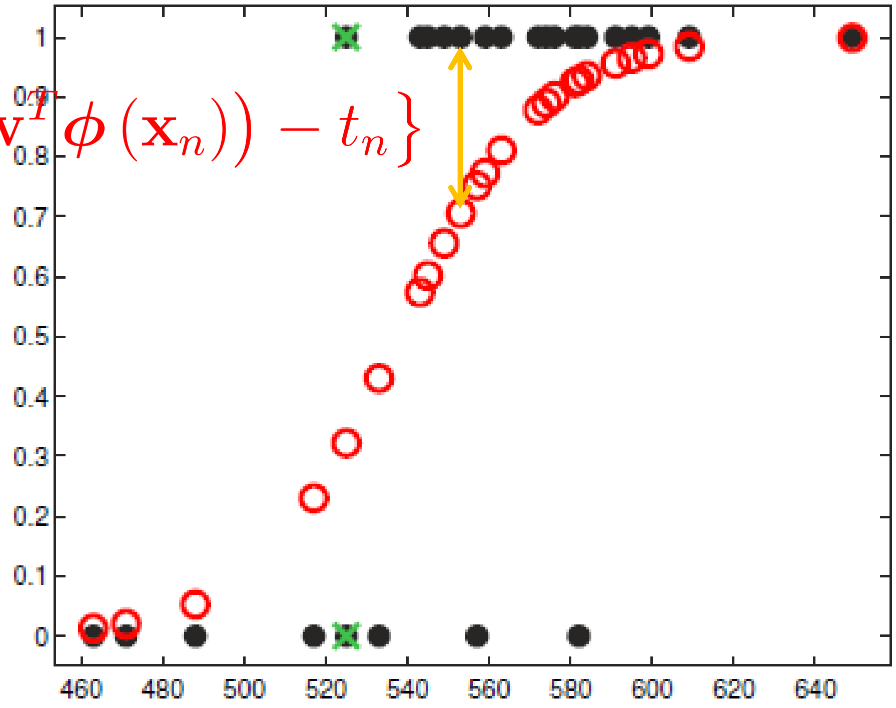
$$\nabla E(\mathbf{w}) = \sum_{n} \{\sigma\left(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\right) - t_n\} \boldsymbol{\phi}(\mathbf{x}_{\boldsymbol{n}})$$

$\sigma\left(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\right) - t_n$ is the difference between target value and the prediction of the model.

Optimizing-Newton method

the difference between target value and the prediction of the model

$$\left\{\sigma\left(\mathbf{w}^T\phi\left(\mathbf{x}_n\right)\right)-t_n\right\}$$



Optimizing-Newton method

The gradient of the error function in logistic regression

$$\nabla E(\mathbf{w}) = \sum_n \{y_n - t_n\} \boldsymbol{\phi}(\mathbf{x}_{\boldsymbol{n}})$$

$$= \boldsymbol{\Phi}^T (\mathbf{y} - \mathbf{t})$$

Where $\mathbf{y} = [y_1, ..., y_N]^T$ and $y_n = \sigma\left(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\right)$

Optimizing-Newton method

The Hessian of the error function in logistic regression

$$\mathbf{H} = \nabla^2 E(\mathbf{w})$$

$$= \sum_j y_n \{1 - y_n\} \boldsymbol{\phi}(\mathbf{x}_{\boldsymbol{n}}) \boldsymbol{\phi}(\mathbf{x}_{\boldsymbol{n}})^T$$

$$= \mathbf{\Phi}^T \mathbf{R} \mathbf{\Phi}$$

Optimizing-Newton method

R is a $N \times N$ diagonal matrix with elements

$$R_{nn} = y_n \left(1 - y_n\right)$$

Using the property that $0 < y_n < 1$, the Hessian matrix H is positive definite.

Optimizing-Newton method

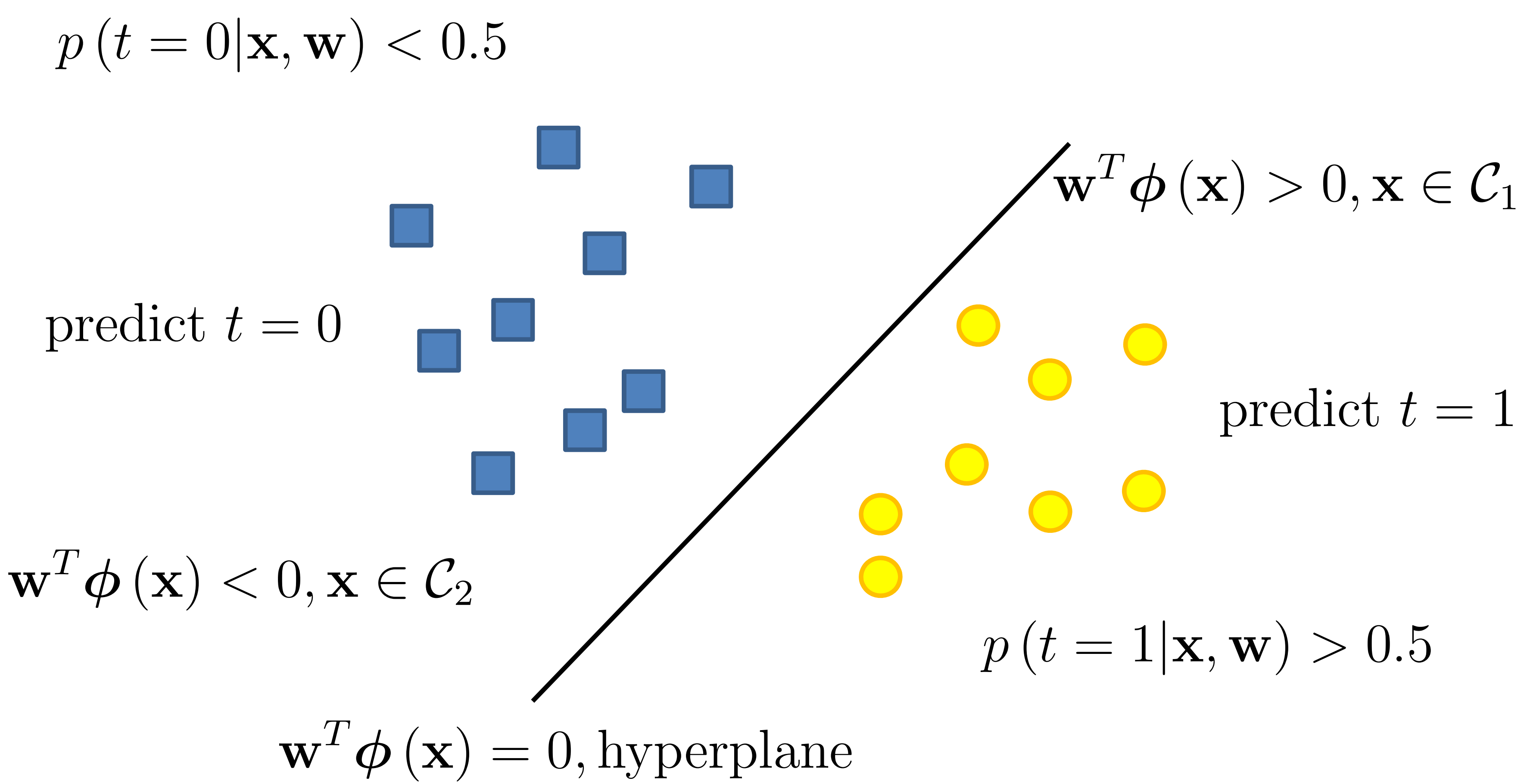
The update rule

$$\mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - \left(\boldsymbol{\Phi}^T \mathbf{R} \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T (\mathbf{y} - \mathbf{t})$$

Because the weight matrix **R** is not constant but depends on the parameter vector **w**, each time using the new weight vector **w** to compute a revised weighting matrix **R**.

Iterative reweighted least squares (IRLS)

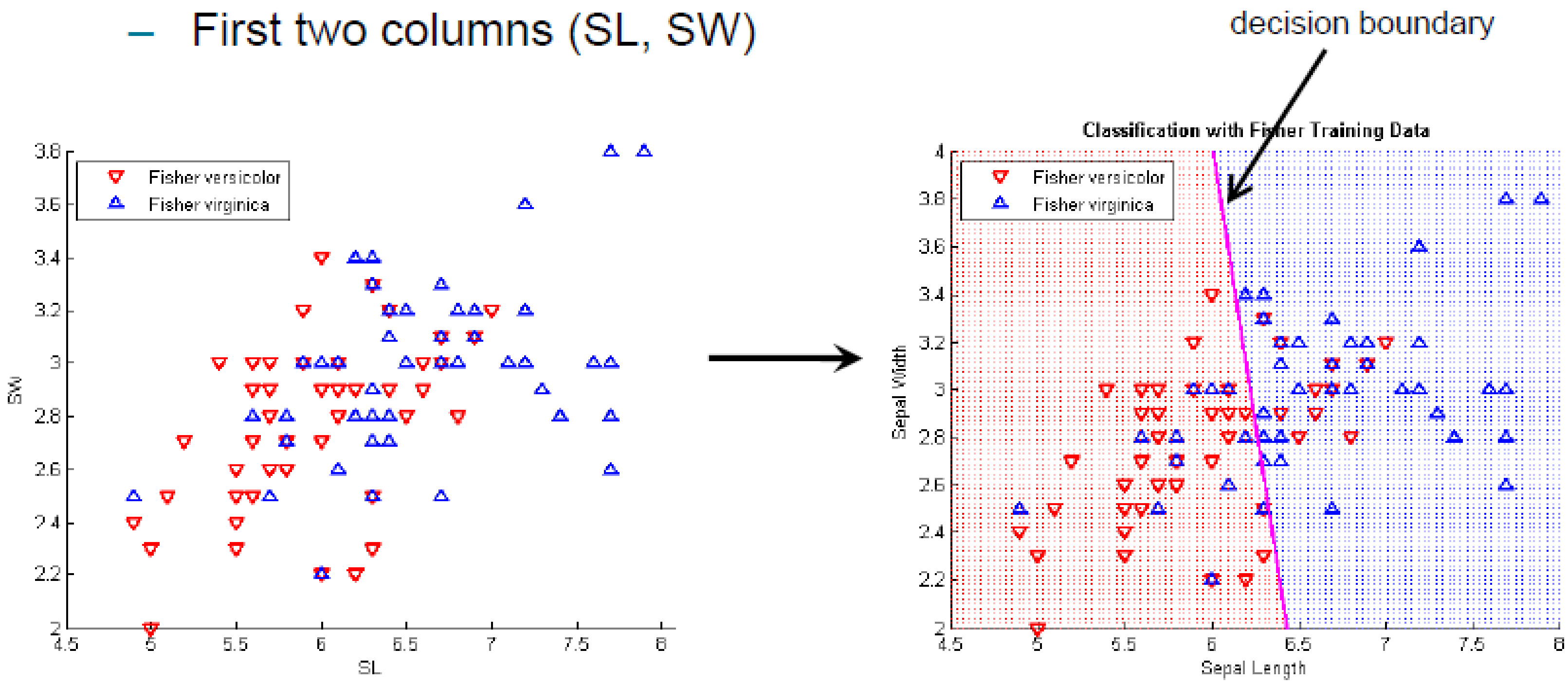
Logistic regression result



Examples

Subset of Fisher iris dataset

- Two classes
- First two columns (SL, SW)

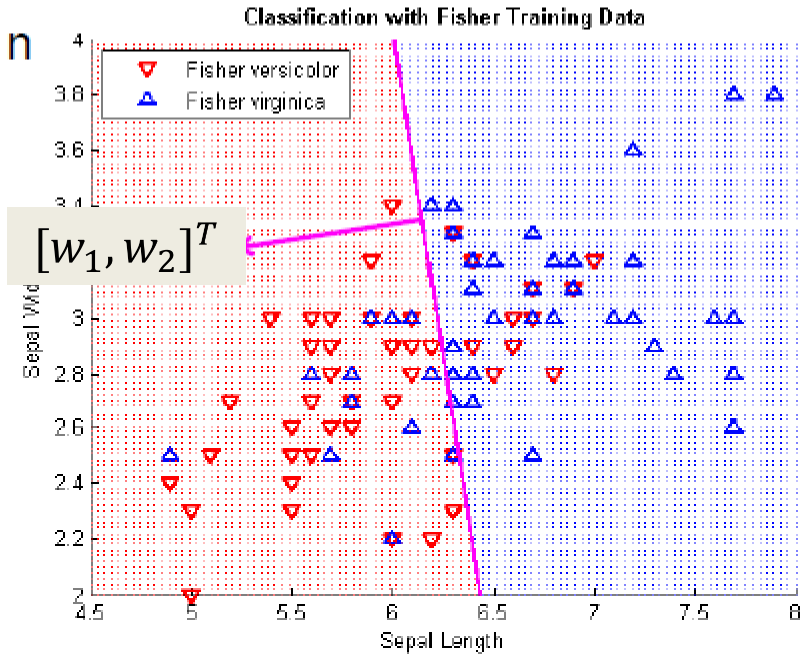


Examples

From MATLAB: **w** = [13.0460 -1.9024 -0.4047]

w_0 determines the distance of decision boundary from origin

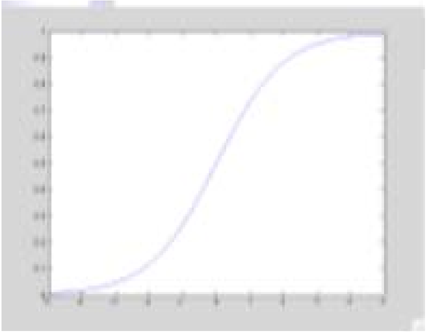
The decision boundary is perpendicular to $[w_1, w_2]^T$



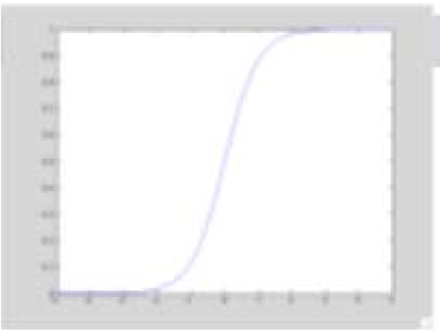
LOGISTIC REGRESSION

1. The concept of logistic regression
2. Optimizing-Newton method
3. Bayesian logistic regression

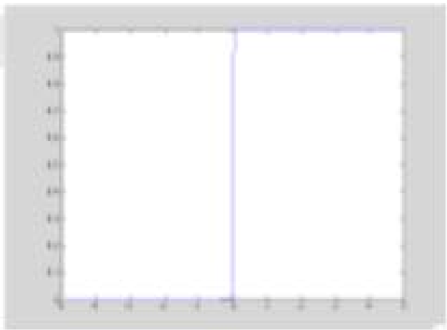
Large parameter, overfitting



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



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



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

If data is linearly separable, weights go to infinity

Leads to overfitting

Penalizing high weights can prevent overfitting

Bayesian logistic regression

Maximum conditional likelihood estimate

$$l\left(\mathbf{w}\right) = \ln \prod_n p\left(t_n|\mathbf{x}_n, \mathbf{w}\right), \mathbf{w}^* = \operatorname{argmax}\, l(\mathbf{w})$$

Maximum conditional a posterior estimate

$$l\left(\mathbf{w}\right) = \ln \prod_n p\left(\mathbf{w}\right) p\left(t_n|\mathbf{x}_n, \mathbf{w}\right), \mathbf{w}^* = \operatorname{argmax}\, l(\mathbf{w})$$