Fundamentals of Machine Learning

LOGISTIC REGRESSION

Amit K Roy-Chowdhury

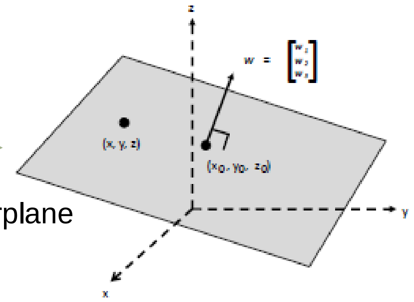
Acknowledgments: Adapted from slides at https://probml.github.io/pml-book/teaching1.html by Prof. Saw Shier Nee

Linear Classifier

The prediction can be written as

$$f(x) = \mathbb{I}\left(p(y=1|x) > p(y=0|x)\right) = \mathbb{I}\left(\log \frac{p(y=1|x)}{p(y=0|x)} > 0\right) = \mathbb{I}\left(a > 0\right)$$

$$f(x;\theta) = b + \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}$$



hyperplane

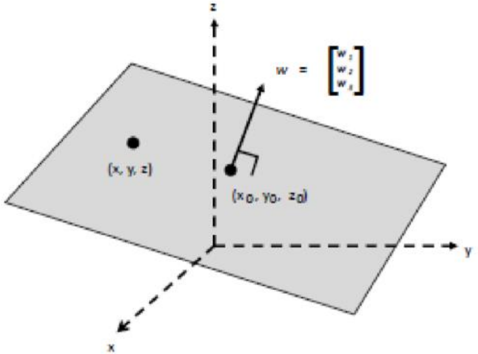
normal vector $w \in \mathbb{R}^D$ and an offset $b \in \mathbb{R}$ from the origin

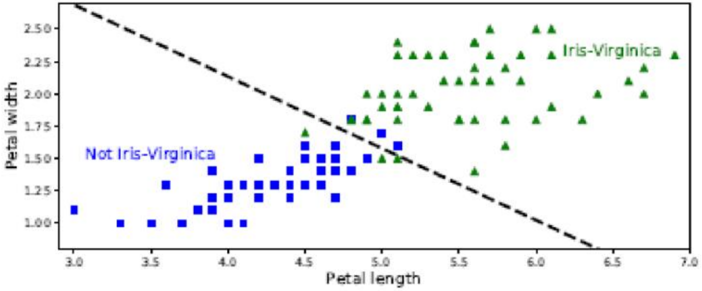
This linear hyperplane (decision boundary) separates 3d space into half spaces.

Linear Classifier

$$f(x) = \mathbb{I}\left(p(y=1|x) > p(y=0|x)\right) = \mathbb{I}\left(\log \frac{p(y=1|x)}{p(y=0|x)} > 0\right) = \mathbb{I}\left(a > 0\right)$$

$$a = \boldsymbol{w}^{\mathsf{T}} \boldsymbol{x} + b$$





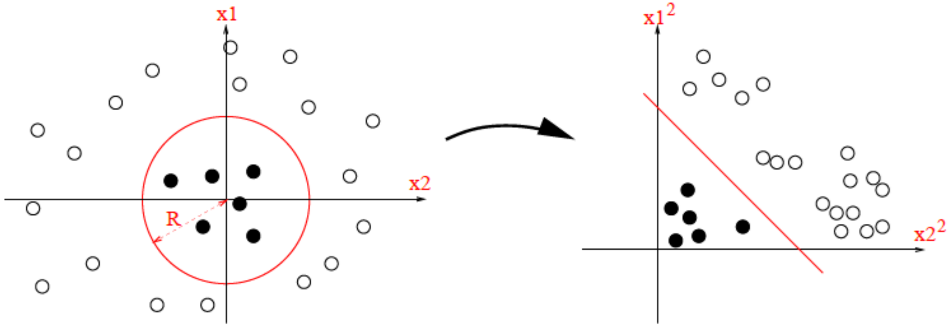
Non Linear Classifier

Transform input features in suitable way

$$\phi(x_1, x_2) = [1, x_1^2, x_2^2]$$

$$\boldsymbol{w} = [-R^2, 1, 1]. \text{ Then } \boldsymbol{w}^\top \phi(\boldsymbol{x}) = x_1^2 + x_2^2 - R^2$$

Decision boundary (where f(x) = 0) defines a circle with radius R

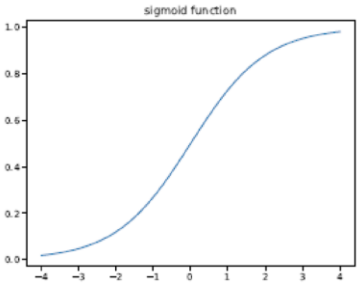


Sigmoid (Logistic) Function

Predict binary random variable y given inputs x.

$$p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \text{Ber}(y|f(\boldsymbol{x}; \boldsymbol{\theta}))$$

$$p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \text{Ber}(y|\sigma(f(\boldsymbol{x}; \boldsymbol{\theta})))$$



Logistic/Sigmoid function

$$\sigma(a) \triangleq \frac{1}{1+e^{-a}} = \frac{e^a}{1+e^a}$$

$$p(y=1|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{1}{1+e^{-a}} = \frac{e^a}{1+e^a} = \sigma(a)$$

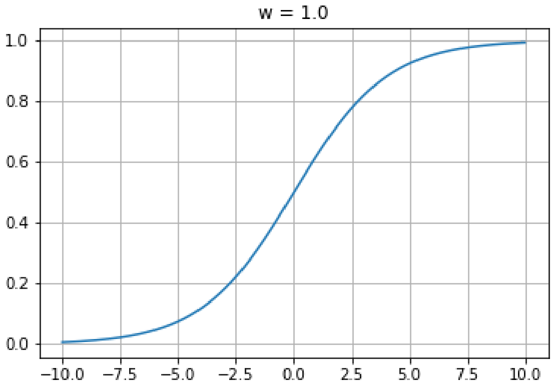
$$p(y=0|\boldsymbol{x}, \boldsymbol{\theta}) = 1 - \frac{1}{1+e^{-a}} = \frac{e^{-a}}{1+e^{-a}} = \frac{1}{1+e^a} = \sigma(-a)$$

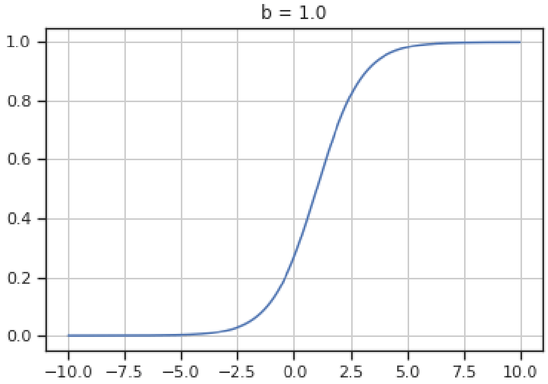
Logit function

$$a = \text{logit}(p) = \sigma^{-1}(p) \triangleq \log\left(\frac{p}{1-p}\right)$$

Sigmoid Function

Sigmoid function $\sigma(a) = \dfrac{1}{1 + e^{-a}},\ \text{where}\ a = \log\dfrac{p}{1-p} = b + \boldsymbol{w}^T\boldsymbol{x}$





Logistic Regression

Binary logistic regression often follows the following model

$$p(y|x;\theta) = \underbrace{\mathrm{Ber}}_{\text{Bernoulli}}(y|\underbrace{\sigma}_{\text{sigmoid}}(\underbrace{w}_{\text{weight}}^{\mathsf{T}}x + \underbrace{b}_{\text{bias}}))$$

$$f(k;p) = \begin{cases} p & \text{if } k = 1, \\ q = 1 - p & \text{if } k = 0. \end{cases}$$

$$p(y=1|x;\theta) = \sigma(a) = \frac{1}{1+e^{-a},}\ , where\ a = log\,\frac{p}{1-p}$$

Logistic Regression – Cost Function

Maximize Maximum Likelihood Estimation / Minimize Negative Log Likelihood

$$\mathrm{NLL}(\boldsymbol{w}) = -\frac{1}{N}\log p(\mathcal{D}|\boldsymbol{w}) = -\frac{1}{N}\log\prod_{n=1}^{N}\mathrm{Ber}(y_n|\mu_n)$$

$$= -\frac{1}{N}\sum_{n=1}^{N}\log[\mu_n^{y_n}\times(1-\mu_n)^{1-y_n}]$$

$$= -\frac{1}{N}\sum_{n=1}^{N}[y_n\log\mu_n + (1-y_n)\log(1-\mu_n)]$$

$$= \frac{1}{N}\sum_{n=1}^{N}\mathbb{H}(y_n,\mu_n)$$

$\mu_n = \sigma(a_n)$ is the probability of class 1

$a_n = \boldsymbol{w}^\mathsf{T}\boldsymbol{x}_n$ is the **logit**

No of Samples

probability

$$\mathbb{H}(p,q) = -[p\log q + (1-p)\log(1-q)]$$

Binary cross-entropy

Logistic Regression – Cost Function

To find the MLE, we must solve

$$\nabla_{\boldsymbol{w}} \mathrm{NLL}(\boldsymbol{w}) = \boldsymbol{g}(\boldsymbol{w}) = \boldsymbol{0}$$

Use chain rule to work out derivatives

$$\begin{aligned}
\nabla_{\boldsymbol{w}} \mathrm{NLL}(\boldsymbol{w}) &= -\frac{1}{N}\sum_{n=1}^{N}\left[y_n(1-\mu_n)\boldsymbol{x}_n - (1-y_n)\mu_n\boldsymbol{x}_n\right] \\
&= -\frac{1}{N}\sum_{n=1}^{N}\left[y_n\boldsymbol{x}_n - y_n\boldsymbol{x}_n\mu_n - \boldsymbol{x}_n\mu_n + y_n\boldsymbol{x}_n\mu_n)\right] \\
&= \frac{1}{N}\sum_{n=1}^{N}\underbrace{(\mu_n - y_n)}_{\text{Error}}\boldsymbol{x}_n
\end{aligned}$$

Here, we can see that the gradient is weighed by the error for each input

Calculating Derivatives

$$a_n = \boldsymbol{w}^\mathsf{T}\boldsymbol{x}_n \text{ and } \mu_n = \sigma(a_n)$$

$$\frac{d\mu_n}{da_n} = \sigma(a_n)(1 - \sigma(a_n))$$

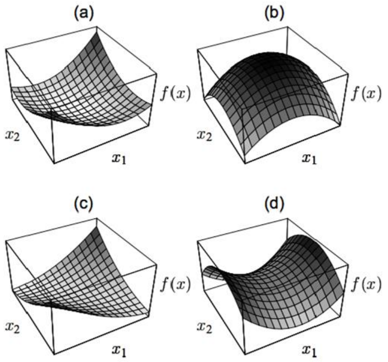
(try! nice exercise in taking derivatives)

$$\frac{\partial}{\partial w_d}\mu_n = \frac{\partial}{\partial w_d}\sigma(\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_n) = \frac{\partial}{\partial a_n}\sigma(a_n)\frac{\partial a_n}{\partial w_d} = \mu_n(1 - \mu_n)x_{nd}$$

$$\nabla_{\boldsymbol{w}} \log(\mu_n) = \frac{1}{\mu_n}\nabla_{\boldsymbol{w}}\mu_n = (1 - \mu_n)\boldsymbol{x}_n$$

$$\nabla_{\boldsymbol{w}} \log(1 - \mu_n) = \frac{-\mu_n(1 - \mu_n)\boldsymbol{x}_n}{1 - \mu_n} = -\mu_n\boldsymbol{x}_n$$

Logistic Regression – Cost Function



Ensure NLL has bowl shape (global minimum)
check Hessian matrix

$$\mathbf{H}(\boldsymbol{w}) = \nabla_{\boldsymbol{w}} \nabla_{\boldsymbol{w}}^{\mathsf{T}} \mathrm{NLL}(\boldsymbol{w}) = \frac{1}{N} \sum_{n=1}^{N} (\mu_n (1 - \mu_n) \boldsymbol{x}_n) \boldsymbol{x}_n^{\mathsf{T}} = \frac{1}{N} \mathbf{X}^{\mathsf{T}} \mathbf{S} \mathbf{X}$$

$$\mathbf{S} \triangleq \mathrm{diag}(\mu_1(1-\mu_1), \ldots, \mu_N(1-\mu_N))$$

Check convexity

We see that $\mathbf{H}$ is positive definite, since for any nonzero vector $\boldsymbol{v}$, we have

$$\boldsymbol{v}^{\mathsf{T}} \mathbf{X}^{\mathsf{T}} \mathbf{S} \mathbf{X} \boldsymbol{v} = (\boldsymbol{v}^{\mathsf{T}} \mathbf{X}^{\mathsf{T}} \mathbf{S}^{\frac{1}{2}})(\mathbf{S}^{\frac{1}{2}} \mathbf{X} \boldsymbol{v}) = ||\boldsymbol{v}^{\mathsf{T}} \mathbf{X}^{\mathsf{T}} \mathbf{S}^{\frac{1}{2}}||_2^2 > 0$$

$$\mathbf{H}_f = \frac{\partial^2 f}{\partial \boldsymbol{x}^2} = \nabla^2 f = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ & \vdots & \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

This follows since $\mu_n > 0$ for all n, because of the use of the sigmoid function. Consequently the NLL is strictly convex. However, in practice, values of μ_n which are close to 0 or 1 might cause the Hessian to be close to singular. We can avoid this by using ℓ_2 regularization, as we discuss in Section 10.2.7.

Logistic Regression – Cost Function

Our goal is to solve the following optimization problem

$$\hat{\boldsymbol{w}} \triangleq \underset{\boldsymbol{w}}{\operatorname{argmin}} \mathcal{L}(\boldsymbol{w})$$

where $\mathcal{L}(\boldsymbol{w})$ is the loss function, in this case the negative log likelihood:

$$\mathrm{NLL}(\boldsymbol{w}) = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \mu_n + (1 - y_n)\log(1 - \mu_n)\right]$$

where $\mu_n = \boldsymbol{\sigma}(a_n)$ is the probability of class 1, and $a_n = \boldsymbol{w}^\mathsf{T}\boldsymbol{x}_n$ is the log odds.

Stochastic gradient descent with minibatch of size 1

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \nabla_{\boldsymbol{w}} \mathrm{NLL}(\boldsymbol{w}_t) = \boldsymbol{w}_t - \eta_t(\mu_n - y_n)\boldsymbol{x}_n$$

Logistic Regression – Optimizer

1. First order method
   - Stochastic Gradient Descent

Slow convergence, when gradient is small

2. Second order method
   - Newton Method (Iteratively reweighted least squares)

$$w_{t+1} = w_t - \eta_t \mathbf{H}_t^{-1} g_t$$

$$\text{where} \quad \mathbf{H}_t \triangleq \nabla^2 \mathcal{L}(w)|_{w_t} = \nabla^2 \mathcal{L}(w_t) = \mathbf{H}(w_t)$$

$$\begin{aligned} w_{t+1} &= w_t - \mathbf{H}^{-1} g_t \\ &= w_t + (\mathbf{X}^\mathsf{T}\mathbf{S}_t\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}(y - \mu_t) \\ &= (\mathbf{X}^\mathsf{T}\mathbf{S}_t\mathbf{X})^{-1}\left[(\mathbf{X}^\mathsf{T}\mathbf{S}_t\mathbf{X})w_t + \mathbf{X}^\mathsf{T}(y - \mu_t)\right] \\ &= (\mathbf{X}^\mathsf{T}\mathbf{S}_t\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\left[\mathbf{S}_t\mathbf{X}w_t + y - \mu_t\right] \\ &= (\mathbf{X}^\mathsf{T}\mathbf{S}_t\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{S}_t z_t \end{aligned}$$

$$z_t \triangleq \mathbf{X}w_t + \mathbf{S}_t^{-1}(y - \mu_t)$$

Logistic Regression – Optimizer

$$w_{t+1} = w_t - \eta_t \mathbf{H}_t^{-1} g_t$$

where $\quad \mathbf{H}_t \triangleq \nabla^2 \mathcal{L}(w)|_{w_t} = \nabla^2 \mathcal{L}(w_t) = \mathbf{H}(w_t)$

$$\begin{aligned}
w_{t+1} &= w_t - \mathbf{H}^{-1} g_t \\
&= w_t + (\mathbf{X}^\mathsf{T} \mathbf{S}_t \mathbf{X})^{-1} \mathbf{X}^\mathsf{T} (y - \mu_t) \\
&= (\mathbf{X}^\mathsf{T} \mathbf{S}_t \mathbf{X})^{-1} \left[(\mathbf{X}^\mathsf{T} \mathbf{S}_t \mathbf{X}) w_t + \mathbf{X}^\mathsf{T} (y - \mu_t) \right] \\
&= (\mathbf{X}^\mathsf{T} \mathbf{S}_t \mathbf{X})^{-1} \mathbf{X}^\mathsf{T} \left[\mathbf{S}_t \mathbf{X} w_t + y - \mu_t \right] \\
&= (\mathbf{X}^\mathsf{T} \mathbf{S}_t \mathbf{X})^{-1} \mathbf{X}^\mathsf{T} \mathbf{S}_t z_t
\end{aligned}$$

where $\quad z_t \triangleq \mathbf{X} w_t + \mathbf{S}_t^{-1} (y - \mu_t)$

This is a iteratively weighted least squares problem: $\displaystyle\sum_{n=1}^{N} S_{t,n} (z_{t,n} - w_t^\mathsf{T} x_n)^2$

Least Squares

$$J(\boldsymbol{\theta}) = \sum_{n=0}^{N-1} \left(x[n] - s[n;\boldsymbol{\theta}] \right)^2$$

$$= (\mathbf{x} - \mathbf{H}\boldsymbol{\theta})^T (\mathbf{x} - \mathbf{H}\boldsymbol{\theta})$$

$$\boxed{\hat{\boldsymbol{\theta}}_{LS} = \left(\mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{x}}$$

Multinomial Logistic Regression

	Binary Logistic Regression	Multiple Logistic Regression
Probability	$p(y\|x;\theta) = \mathrm{Ber}(y\|\sigma(w^\top x + b))$	$p(y\|x;\theta) = \prod_{c=1}^{C} \mathrm{Ber}(y_c\|\sigma(w_c^\top x))$
Activation function	σ = sigmoid activation	σ = softmax activation
	$p(y=1\|x;\theta) = \sigma(a) = \frac{1}{1+e^{-a}}$	$\mathrm{softmax}(a) \triangleq \left[\frac{e^{a_1}}{\sum_{c'=1}^{C} e^{a_{c'}}}, \dots, \frac{e^{a_C}}{\sum_{c'=1}^{C} e^{a_{c'}}}\right]$
Cost function	$\frac{1}{N}\sum_{n=1}^{N} \mathbb{H}_{ce}(y_n, \mu_n)$	$\frac{1}{N}\sum_{n=1}^{N} \mathbb{H}_{ce}(y_n, \mu_n)$
	$\mathbb{H}_{ce}(p,q) = -[p\log q + (1-p)\log(1-q)]$	$\mathbb{H}_{ce}(p,q) = -\sum_{c=1}^{C} p_c \log q_c$