Math 4570 Matrix Methods for DA and ML

Section 11. Logistic Regression

1. Logistic Regression (binary)
2. Softmax Regression (multiclass)

Instructor: He Wang
Department of Mathematics
Northeastern University

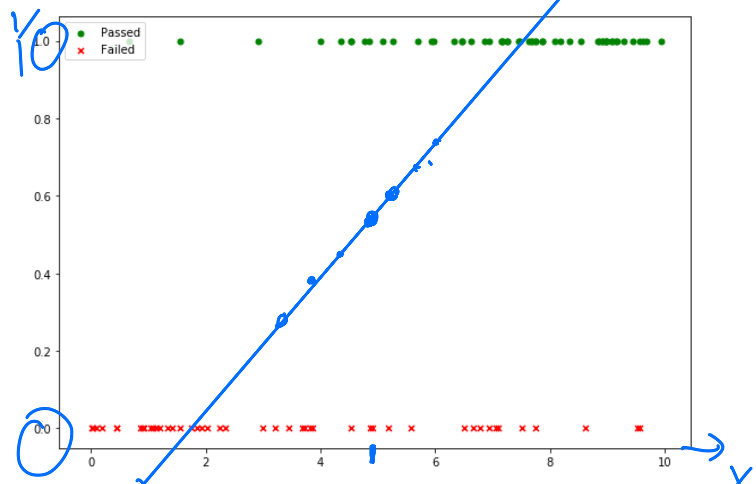
- Example: Data of students sleep time, study time, and pass/fail.

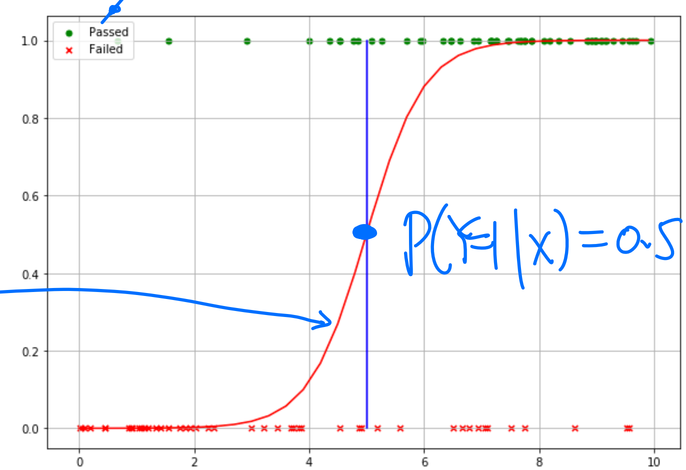
If we know the test scores, we can use linear regression to predict the test scores.

passed Y=1, failed Y=0

x_1

studied	Y	Z
7.40	1	95
3.93	0	50
0.72	0	44
3.89	1	80
8.19	1	90
...	...	⋮



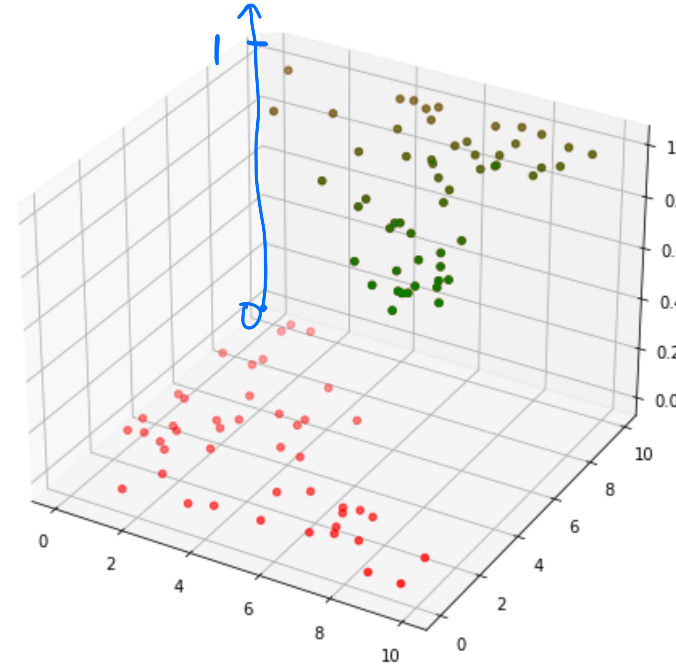


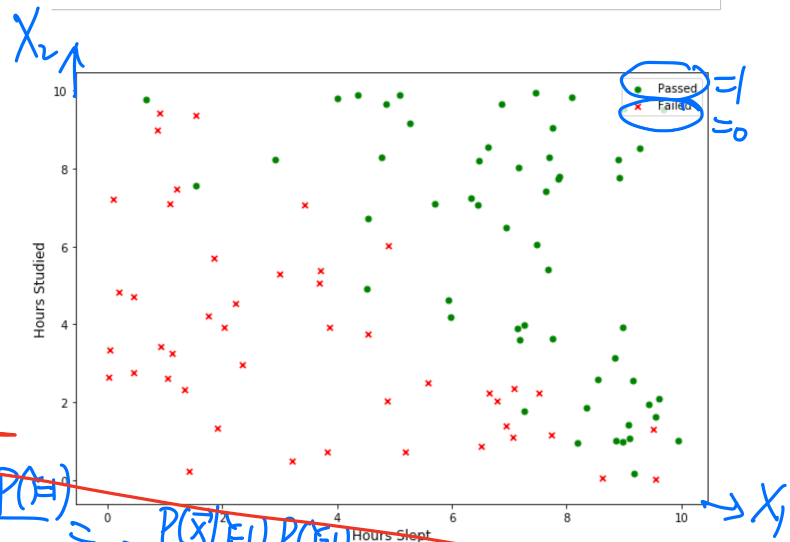
$P(Y=1 \mid x)=?$

passed Y=1, failed Y=0

X_1 X_2 Y

slept	studied	Y
7.63	7.40	1
2.03	3.93	0
3.82	0.72	0
7.15	3.89	1
6.47	8.19	1
...	...	...





Model: $\vec{x} \in \mathbb{R}^2 \xrightarrow{f_\theta} \{0,1\}$

$h_\theta(\vec{x}) \to [0,1]$

$\parallel$

Goal $\boxed{P(Y=1 \mid \vec{x})}$

method 1: model

method 2: $\frac{P(\vec{x} \mid Y=1)P(Y=1)}{P(\vec{x})} = P(\vec{x} \mid Y=1)P(Y=1)$

$P(\vec{x}|Y=1)P(Y=1)+P(\vec{x}|Y=0)P(Y=0)$

GDA $\begin{cases} \text{LDA} \\ \text{QDA} \end{cases}$

➢ Logistic regression

Logistic regression is a **classification** algorithm, used to predict probabilities based on given set of independent variables.

$\boxed{\text{Data matrix } X}$

$\vec{y}$

Data: $D = \{(\vec{x}^{(i)}, y^{(i)}), i = 1, \dots n\}$ $y^{(i)} \in \{0, 1\}$,

Goal: Find conditional (posterior) probability

$$P(Y = k \,|\, \vec{X} = \vec{x}) \quad \text{for } k = 0, 1$$

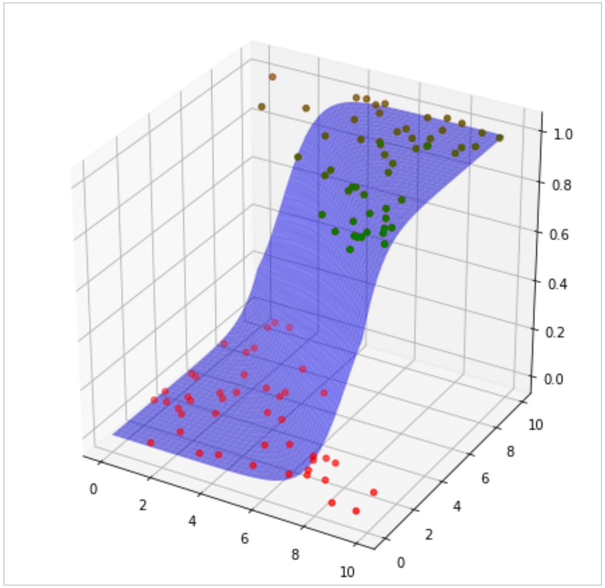
model

➢ Bayes Decision Boundary

logistic regression prediction function returns a probability between 0 and 1, in order to predict which class this data belongs we need to set a threshold.

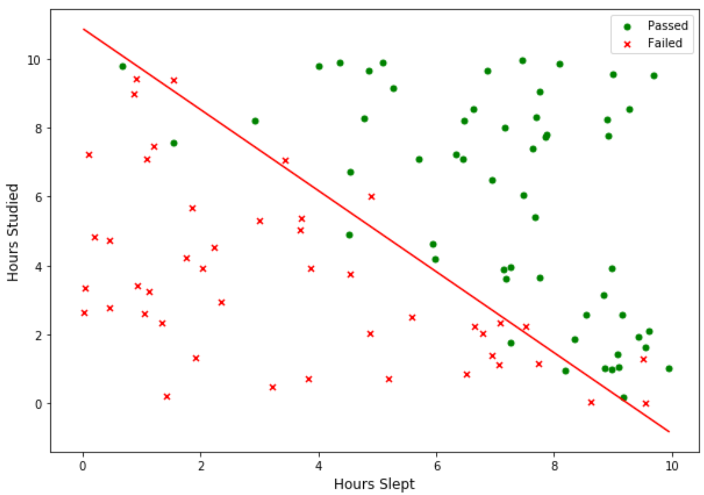
Bayes Boundary $P(Y = 0 \,|\, \vec{x}) = P(Y = 1 \,|\, \vec{x})$ $= 0.5$

Or $P(Y = 1 \,|\, \vec{x}) = 0.5$



Want

$$h_{\vec{\theta}}(\vec{x}) =: P(Y = 1 \mid \vec{x})$$



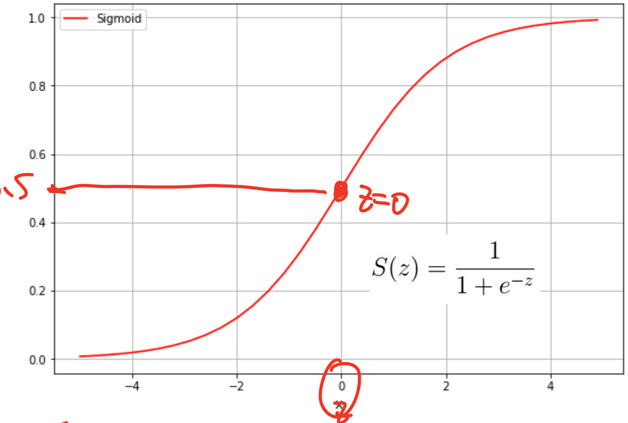
$$h_{\vec{\theta}}(\vec{x}) = 0.5$$

$\vec{x} \in \mathbb{R}^d$ $\qquad -\infty < \vec{\theta}^T \vec{x} = \theta_0 + \theta_1 x_1 + \dots + \theta_d x_d < \infty$

➢ Logistics regression.

The **sigmoid function** maps any real value into a value in [0,1].

$$0 < S(z) = \frac{1}{1+e^{-z}} < 1$$



- **Logistics regression "assumption:"**

$$P(Y = 1 \mid \vec{x}) := h_{\vec{\theta}}(\vec{x}) := S(\vec{\theta}^T \vec{x}) = \boxed{\frac{1}{1+e^{-\vec{\theta}^T \vec{x}}}}$$

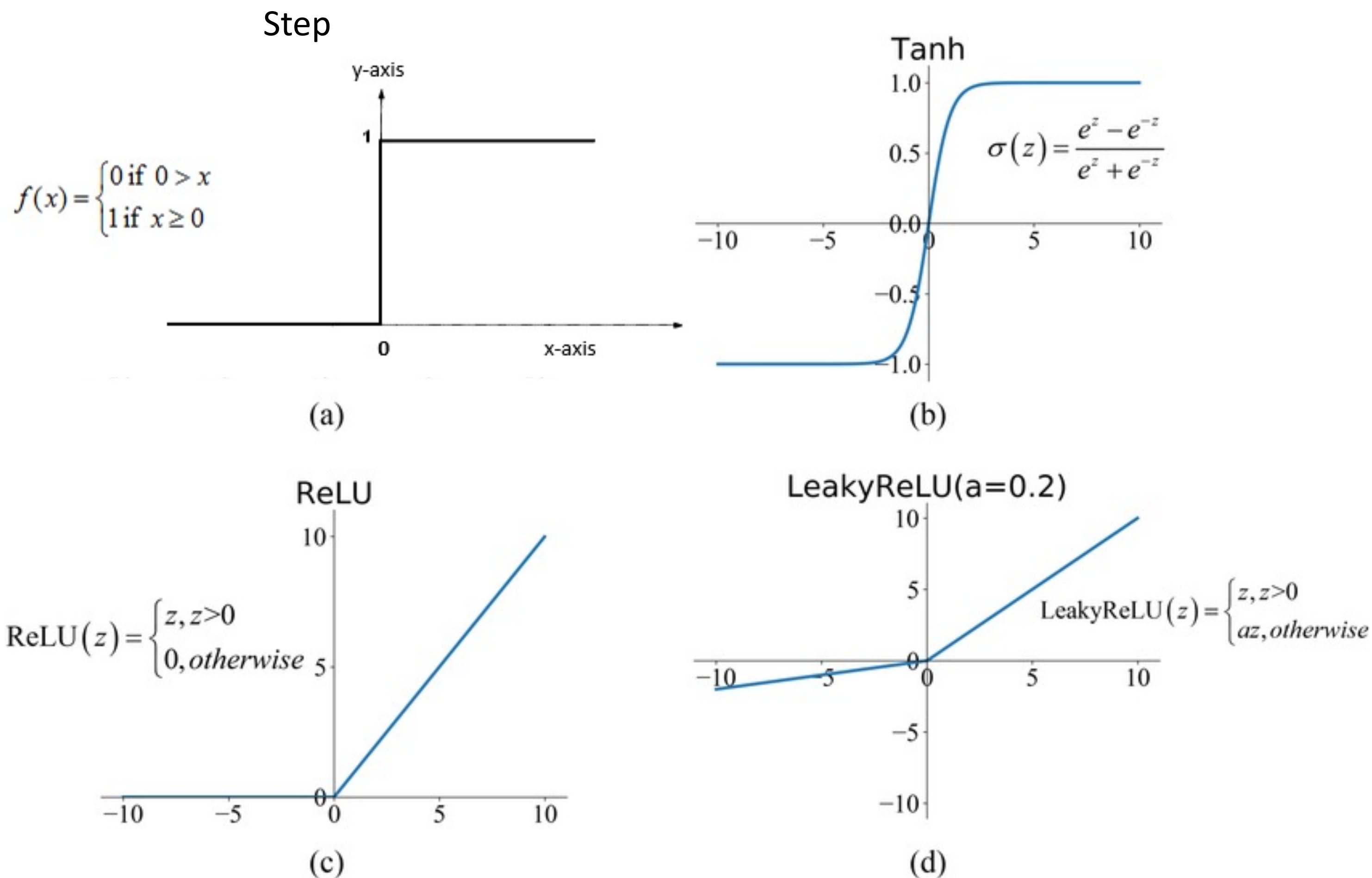
- **Prediction**:

$$C(\vec{x}) = \begin{cases} 1, & if\ h(\vec{x}) \geq 0.5 \\ 0, & if\ h(\vec{x}) < 0.5 \end{cases}$$

- **Bayes Decision Boundary** $\quad h(\vec{x}) = 0.5 \iff \vec{\theta}^T \vec{x} = 0$

$$\vec{\theta}^T \vec{x} = 0$$

- Other activation functions



Rectified Linear Unit (ReLU)

- $P(Y=1 \mid \vec{x}) = h_{\vec{\theta}}(\vec{x}) := \frac{1}{1 + e^{-\vec{\theta}^T \vec{x}}}$

➢ Maximize Likelihood method:

Logistics regression Assumption (with label space $\boxed{\mathcal{C} = \{0,1\}}$):

$$\begin{cases} P(Y = 1 \mid \vec{x}; \vec{\theta}) = h_{\vec{\theta}}(\vec{x}) & \text{if } Y=1 \\ P(Y = 0 \mid \vec{x}; \vec{\theta}) = 1 - h_{\vec{\theta}}(\vec{x}) & \text{if } Y=0 \end{cases}$$

Equivalently,

$$\boxed{P(Y = y \mid \vec{x}; \vec{\theta}) = h_{\vec{\theta}}(\vec{x})^{y} \left(1 - h_{\vec{\theta}}(\vec{x})\right)^{1-y}}$$

$y = 0$, or, 1

The above random variable Y is the **Bernoulli Distribution** with probability $p = h_{\vec{\theta}}$ depending on $\vec{x}$ and parameter $\vec{\theta}$.

(1). Data $(\vec{x}^{(i)}, y^{(i)})$, $X, \vec{y}$

(2). Model $h_{\theta}(\vec{x})$

$\boxed{\text{(3). cost of } h_{\theta}(\vec{x}) \quad J(\vec{\theta})}$

(4) $\underset{\vec{\theta}}{\arg\min}\,(J(\vec{\theta}))$

Given labeled data: $(\boldsymbol{X}, \vec{\boldsymbol{y}})$ $\quad$ $y^{(i)} \in \{0, 1\}$

$$\boxed{P(\text{"Data"})}$$

Maximize

Likelihood function:

$$L(\vec{\theta}) := \boxed{P(\vec{y} \mid X; \vec{\theta})} = \frac{P(\vec{X}, \vec{y})}{\cancel{P(\vec{X})}}$$

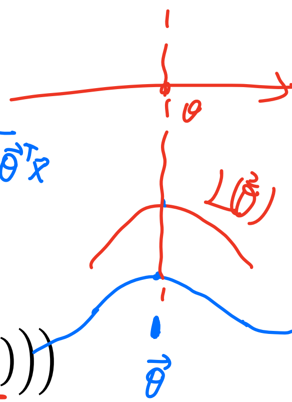
$$\vec{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix} \qquad X = \begin{bmatrix} \vec{x}^{(1)T} \\ \vdots \\ \vec{x}^{(n)T} \end{bmatrix}$$

independent

$$= \prod_{i=1}^{n} P\left(y^{(i)} \mid \vec{x}^{(i)} ; \vec{\theta}\right)$$

$$= \prod_{i=1}^{n} h_{\vec{\theta}}\left(\vec{x}^{(i)}\right)^{y^{(i)}} \left(1 - h_{\vec{\theta}}\left(\vec{x}^{(i)}\right)\right)^{1-y^{(i)}}$$

$$h_{\vec{\theta}} = \frac{1}{1+e^{-\vec{\theta}^T x}}$$



Log Likelihood function:

$$l(\vec{\theta}) = \log L(\vec{\theta})$$

$$= \sum_{i=1}^{n} \left(y^{(i)} \, \ln h_{\vec{\theta}}\left(\vec{x}^{(i)}\right) + \left(1 - y^{(i)}\right) \ln \left(1 - h_{\vec{\theta}}\left(\vec{x}^{(i)}\right)\right)\right)$$

$0 \leq \quad \leq 1$

$(f \cdot g)' = f'g + fg'$

$(f+g)' = f' + g'$

$\log(z)$

① Data $(\vec{X}^{(i)}, y^{(i)})$ or $(X, \vec{y})$

② Model: $h_\theta(\vec{x}) : \mathbb{R}^d \longrightarrow \mathbb{R}$

③ Cost: $J(\vec{\theta})$

$\rightarrow \frac{1}{n}\| h(X) - \vec{y} \|^2$ least-square

or $P(\vec{y} \mid X)$

$\rightarrow -\frac{1}{n} \ln\left(P(\text{"data"})\right)$ ← cross-entropy

likelihood

Naive Bayes

LDA/QDA

linear regression

④ $\vec{\theta}^{opt.} = \underline{\arg\min}(J(\vec{\theta}))$?

Solution: (i) Solve $\boxed{\nabla J(\vec{\theta}) = 0}$

→ there is a solu.

→ can not find a formula.

gradient descent

Newton's method

• For any $\vec{\theta}^{current}$,

the fastest decreasing direction

is $-\nabla J(\vec{\theta}^{current})$

$$\vec{\theta}^{next} = \vec{\theta} - \alpha \nabla J(\vec{\theta})$$

Optimization: (Maximize Likelihood)

$$\arg\max \; L(\vec{\theta})$$

$$= \arg\max \; l(\vec{\theta})$$

$y^{(i)} = 0, 1$

$\mathbb{1}(y^{(i)}=1)$ $\quad$ $\mathbb{1}(y^{(i)}=0)$

$$= \arg\mathbf{min} \; -\frac{\mathbf{1}}{\mathbf{n}} \sum_{i=1}^{n} \left(y^{(i)} \ln h_{\vec{\theta}}\left(\vec{x}^{(i)}\right) + \left(1 - y^{(i)}\right) \ln\left(1 - h_{\vec{\theta}}\left(\vec{x}^{(i)}\right)\right)\right)$$

(4) $\nabla J(\theta) = 0$

Cross-entropy Loss $J(\vec{\theta})$

Or log-cost function

$\mathbb{1}(\text{true}) = 1$

$\mathbb{1}(\text{false}) = 0$

Cost for each individual point $\vec{x}^{(i)}, y^{(i)}$:

$$J(\vec{\theta}; \vec{x}^{(i)}) = \begin{cases} -\ln h_{\vec{\theta}}\left(\vec{x}^{(i)}\right) & if \;\; y^{(i)} = 1 \\ -\ln\left(1 - h_{\vec{\theta}}\left(\vec{x}^{(i)}\right)\right) & if \;\; y^{(i)} = 0 \end{cases}$$

- **Gradient descent for Cross-entropy Loss**

$$\nabla J(\vec{\theta}) = \begin{bmatrix} \frac{\partial J(\vec{\theta})}{\partial \theta_0} \\ \vdots \\ \frac{\partial J(\vec{\theta})}{\partial \theta_d} \end{bmatrix} \qquad \frac{\partial J(\vec{\theta})}{\partial \theta_i} = ?$$

Recall: $h_{\vec{\theta}}(\vec{x}) := S(\vec{\theta}^T\vec{x}) = \dfrac{1}{1+e^{-\vec{\theta}^T\vec{x}}}$

$$S(z) = \frac{1}{1+e^{-z}}$$

$$\frac{d\,S(z)}{d\,z} = S(z)(1-S(z))$$

$$\frac{\partial h_{\vec{\theta}}(\vec{x}^{(i)})}{\partial \theta_j} = S(z)(1-S(z))x_j^{(i)} \qquad z = \vec{\theta}^T\vec{x}$$

$$= \frac{d\,S(z)}{dz} \cdot \frac{\partial z}{\partial \theta_j}$$

$$= \theta_0 + \theta_1 x_1 + \dots + \theta_d x_d$$

$$\frac{\partial J(\vec{\theta})}{\partial \theta_i} = -\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)}\frac{1}{S(z)}\,S(z)\big(1-S(z)\big)x_j^{(i)} - \left(1-y^{(i)}\right)\frac{1}{1-S(z)}\,S(z)\big(1-S(z)\big)x_j^{(i)}\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(S(\vec{\theta}^T\vec{x}) - y^{(i)}\right)x_j^{(i)}$$

$$\nabla J = \begin{bmatrix} \frac{\partial J}{\partial \theta_0} \\ \frac{\partial J}{\partial \theta_1} \\ \vdots \\ \frac{\partial J}{\partial \theta_d} \end{bmatrix}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(h(\vec{x}^{(i)}) - y^{(i)}\right)x_j^{(i)}$$

$X\vec{\theta}$

Vector notation of the gradient:

logistic

$$\boxed{\nabla_{\vec{\theta}}\, J = \frac{1}{n}X^T\left(h_{\vec{\theta}}(X) - \vec{y}\,\right)} = 0$$

$$h_{\vec{\theta}}(\vec{x}) = \frac{1}{1+e^{-\vec{\theta}^T\vec{x}}}$$

linear regression

$$\nabla J = 2\,X^T\left(h(X) - \vec{y}\right)$$

$$h(\vec{x}) = \vec{\theta}^T\vec{x}$$

Gradient Descent and Newton's method for Logistics Regression

- **Gradient Descent:**

$$\boxed{\vec{\theta}_{k+1} = \vec{\theta}_k - \alpha \nabla_{\vec{\theta}_k} J} = \vec{\theta}_k - \alpha \frac{1}{n} X^T \left(h_{\vec{\theta}_k}(X) - \vec{y} \right)$$

- **Newton's method:**

$$\vec{\theta}_{k+1} = \vec{\theta}_k - H^{-1} \nabla J(\vec{\theta}_k)$$

Here H is the Hessian matrix $H = \begin{bmatrix} \frac{\partial^2 J}{\partial \theta_1{}^2} & \cdots & \frac{\partial^2 J}{\partial \theta_1 \partial \theta_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 J}{\partial \theta_d \partial \theta_1} & \cdots & \frac{\partial^2 J}{\partial \theta_d{}^2} \end{bmatrix}$

with $\quad H_{jk} = \dfrac{\partial^2 J}{\partial \theta_j \partial \theta_k} = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} h(\vec{x}^{(i)}) \left(1 - h(\vec{x}^{(i)})\right) x_j^{(i)} x_k^{(i)}$

Matrix Notation for $H = \frac{1}{n} X^T A X$, where A=**diag** $\left[h(\vec{x}^{(i)}) \left(1 - h(\vec{x}^{(i)})\right) \right]$

Question: If $y \in \{-1, 1\}$,

$$P(Y = 1 \mid \vec{x}; \vec{\theta}) = h_{\vec{\theta}}(\vec{x})$$

$$P(Y = -1 \mid \vec{x}; \vec{\theta}) = 1 - h_{\vec{\theta}}(\vec{x})$$

Equivalently,

$$P(Y = y \mid \vec{x}; \vec{\theta}) = h_{\vec{\theta}}(y\vec{x}) \qquad \text{(why?)}$$

1. Find $J(\vec{\theta})$.
2. Calculate gradient $\nabla_{\vec{\theta}} J$
3. Calculate Hessian matrix.

$P(Y=1|\vec{x}) = h(\vec{x}) = ?$

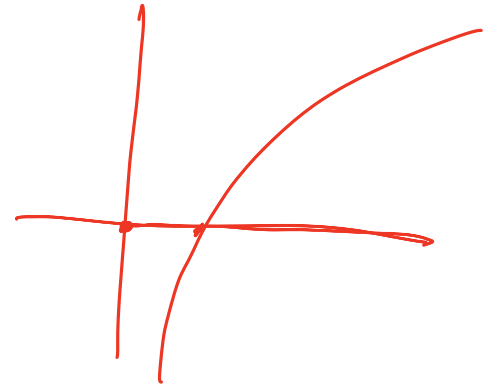
Odds Ratio: A ratio of two probabilities.

Log Odd Ratio: logarithm of an odds ratio.

$$-\infty \leq \log \frac{P(Y=1|\vec{x})}{P(Y=0|\vec{x})} \leq \infty$$

$$0 \leq \frac{P(Y=1|\vec{x})}{P(Y=0|\vec{x})} \leq \infty$$

$$\log \frac{P(Y=1|\vec{x})}{P(Y=0|\vec{x})} = \log \frac{h(\vec{x})}{1-h(\vec{x})} := \vec{\theta}^T \vec{x}$$



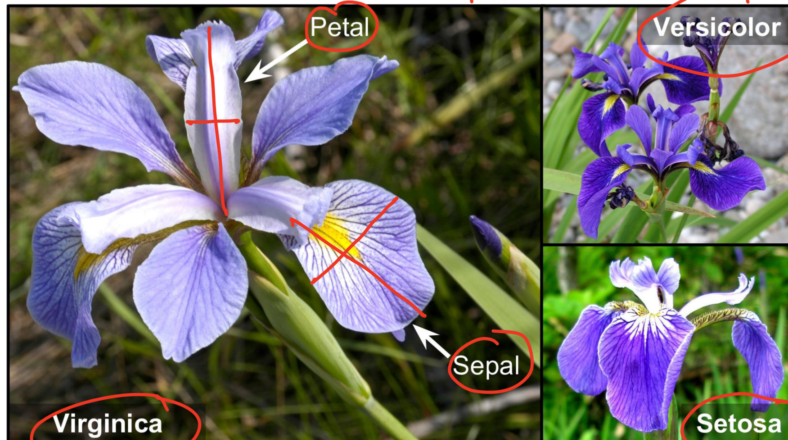
$\Updownarrow$

Logistic Regression assumption $h_{\vec{\theta}}(\vec{x}) := \dfrac{1}{1+e^{-\vec{\theta}^T \vec{x}}}$

- **Softmax Regression** (Multinomial Logistic Regression) binary $\{0,1\}$

  - Flowers of three iris plant species:

The famous Iris database, first used by Sir R.A. Fisher(1936), is best known database to be found in the pattern recognition literature. It contains the **sepal** and **petal length** and **width** of 150 iris flowers of three different species: Iris-Setosa, Iris-Versicolor, and Iris-Virginica.



Data features:

$$\left.\begin{array}{ll}\text{Sepal length} & x_1\\ \text{Sepal width} & x_2\\ \text{Petal length} & x_3\\ \text{Petal width} & x_4\end{array}\right] = \vec{x} \in \mathbb{R}^4$$

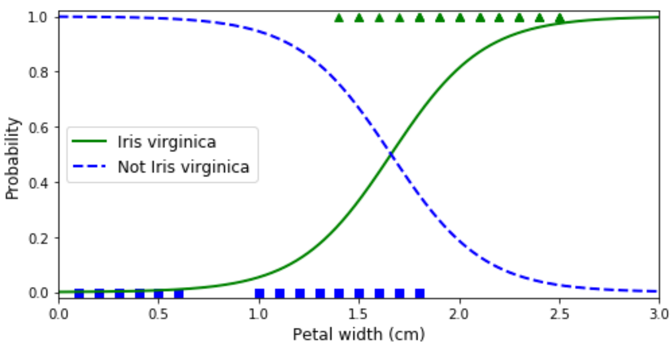
Classes: 0-Iris-Setosa, 1-Iris-Versicolour, 2-Iris-Virginica

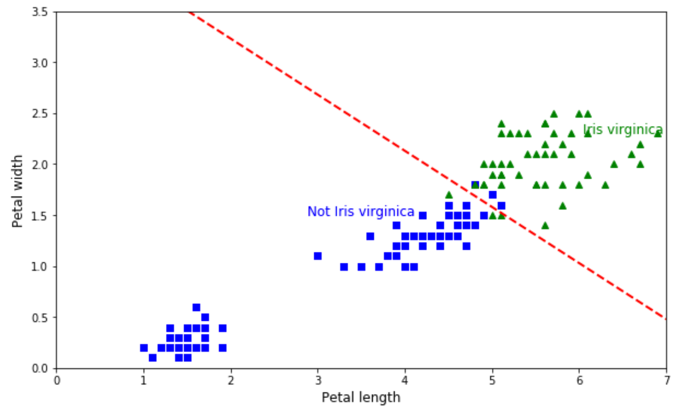
Data: $D = \{(\vec{x}^{(i)}, y^{(i)}), i = 1, \dots n\}$ $\quad y^{(i)} \in \{0, 1, 2\}$ 1, 2, 3

x_1 x_2 x_3 x_4	y
[5.1, 3.5, 1.4, 0.2]	0
[4.9, 3. , 1.4, 0.2]	0
[4.7, 3.2, 1.3, 0.2]	1
[4.6, 3.1, 1.5, 0.2]	0
[5. , 3.6, 1.4, 0.2]	1
[5.4, 3.9, 1.7, 0.4]	2
[4.6, 3.4, 1.4, 0.3]	0
[5. , 3.4, 1.5, 0.2]	1
[4.4, 2.9, 1.4, 0.2]	2
...	

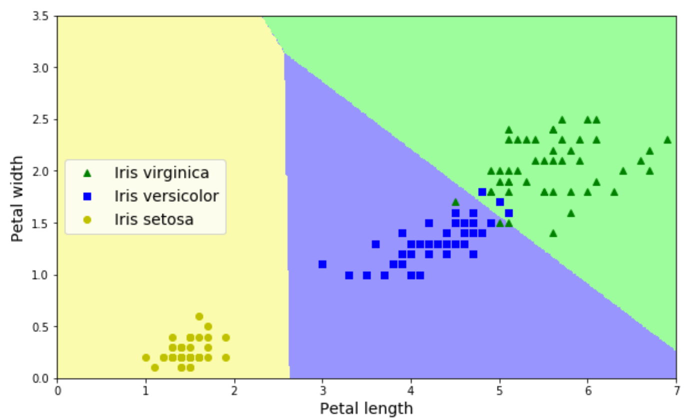
$= \vec{y}$

one v.s. rest





Softmax:



Softmax Regression

$\vec{x}$ $\quad -\infty < \vec{\theta}^T \vec{x} < \infty$

$0 < \exp(\vec{\theta}^T \vec{x}) < \infty$

$h(\vec{x}) = P(Y=1 \mid \vec{x})$

K=1 logistic

$P(Y=0 \mid \vec{x})$

Goal: $P(Y = k \mid \vec{X} = \vec{x}) = ?$ for $k = 0, 1, \ldots, K$

k=2

Assumption:

$$\begin{bmatrix} P(Y=0 \mid \vec{x};\ \vec{\theta}) \\ P(Y=1 \mid \vec{x};\ \vec{\theta}) \\ P(Y=2 \mid \vec{x};\ \vec{\theta}) \end{bmatrix} := \frac{1}{\sum_{j=0}^{K} \exp \vec{\theta}_j^T \vec{x}} \begin{bmatrix} \exp \vec{\theta}_0^T \vec{x} \\ \exp \vec{\theta}_1^T \vec{x} \\ \exp \vec{\theta}_2^T \vec{x} \end{bmatrix} =: h_{\vec{\theta}}(\vec{x}) = \begin{bmatrix} h_{0\vec{\theta}}(\vec{x}) \\ h_{1\vec{\theta}}(\vec{x}) \\ h_{2\vec{\theta}}(\vec{x}) \end{bmatrix}$$

Here $\vec{\theta}_j = \begin{bmatrix} \theta_{j,0} \\ \theta_{j,1} \\ \vdots \\ \theta_{j,d} \end{bmatrix}$ $\quad j = 0, 1, 2.$

So, we have $(K+1)(d+1)$ parameters $\Theta = [\vec{\theta}_1 \ \ldots \vec{\theta}_K]$.

Cross-entropy (log-cost) Loss

3

$$J(\vec{\theta}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{K} \mathbb{I}\left(y^{(i)} = j\right) \ln P\left(y^{(i)} = j \middle| \vec{x}^{(i)}; \vec{\theta}\right)$$

$$= -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{K} \mathbb{I}\left(y^{(i)} = j\right) \ln \frac{\exp \vec{\theta}_j^{\,T} \vec{x}^{(i)}}{\sum_{l=0}^{K} \exp \vec{\theta}_l^{\,T} \vec{x}^{(i)}}$$

$\mathbb{I}(\ )$ is the **indicator function**:

$$\mathbb{I}\,(\text{True}\,) = 1$$

$$\mathbb{I}\,(\text{False}\,) = 0$$

4 $\boxed{\arg\min J(\vec{\theta})}$

$\nabla J(\vec{\theta}) = 0$

- **Gradient Descent:**

The **gradient** of Cross-entropy Loss is

$$\nabla_{\vec{\theta}_j} J(\vec{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\left(h_{\vec{\theta}}\left(\vec{x}^{(i)}\right) - \mathbb{I}\left(y^{(i)} = j\right)\right)\vec{x}^{(i)}$$

Gradient Descent:

$$\vec{\theta}^{next} = \vec{\theta} - \alpha \nabla_{\vec{\theta}} J$$

Hessian is non-invertible in this case, so we can not use Newton's method directly.

➢ Some Remarks:

- Logistics regression with **non-linear** boundaries:

$$\begin{array}{cc} x_1 & x_2 \\ \vdots & \vdots \end{array} \qquad \begin{array}{c} y \\ \vdots \end{array}$$

Similarly, as linear regression, we can introduce new features

$$\boxed{z_1 = x_1, z_2 = x_2}, z_3 = x_1^2, z_4 = x_2^2, z_5 = x_1 x_2, \; z_6 = x_1^3,$$

$$z_7 = x_2^3, z_8 = x_1^2 x_2, \; z_9 = x_1 x_2^2 \;, \ldots$$

$$\begin{array}{ccccc} x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{array} \qquad \begin{array}{c} y \\ \vdots \end{array}$$

Apply logistics regression to the new features, get the boundary and replace back to $x_1, x_2, \; x_1^2, x_2^2, x_1 x_2, \; x_1^3, x_2^3, x_1^2 x_2, x_1 x_2^2 \ldots$

Then we get the non-linear boundary.

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2$$
$$+ \theta_4 x_1 x_2 + \theta_5 x_2^2 = 0$$

- Logistic regression with (ridge/lasso) regularization

Regularization Cost = **Cross-entropy Loss +Penalty**

$$J^{ridge}(\vec{\theta}) = J(\vec{\theta}) + \lambda \sum_{j=1}^{d} {\theta_j}^2$$

$$J^{lasso}(\vec{\theta}) = J(\vec{\theta}) + \lambda \sum_{j=1}^{d} |\theta_j|$$

Convert Categorical Data to Numerical Data

We used **Integer Encoding** for the classification, which means using 0,1,…, K for classes.

Note that in a K-class classification the individual classes can sometimes be usefully represented as K-length binary variables. (**One-Hot Encoding**)

This means we denote class j to be

$$\vec{e}_j = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \in \mathbb{R}^K$$

The binary variables are often called “dummy variables” in statistics.

- **Applications**:

1. Email spam detector
2. Diagnose a person with a set of syndromes as virus carrier or non-carrier.
3. Identify which gene, out of a million genes, is disease-causing or not.
4. Judge if a trading activity is a fraud or not.
5. …