机器学习
统计学习方法

主讲：蔡 波
武汉大学网络安全学院

第六章 补充

主讲：蔡 波

武汉大学网络安全学院

【超平面的性质】

$$\boxed{w^T(x_1 - x_2) = 0}$$

$$\boxed{x = x_p + r\frac{w}{\|w\|}}$$

式中:

x_p -- x 在H上的投影

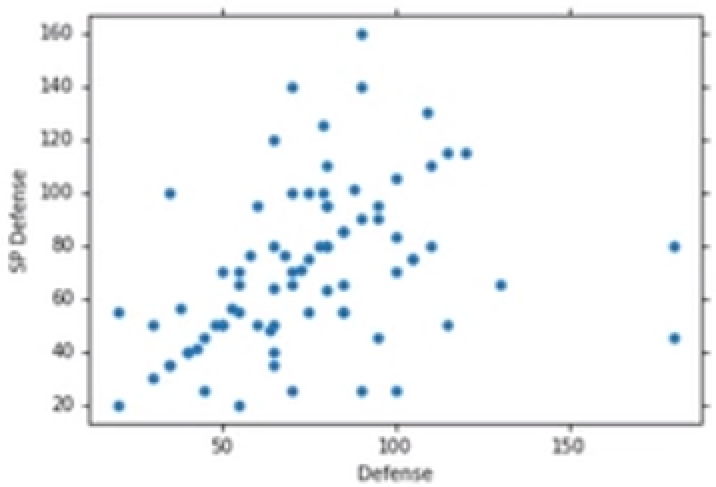
r -- x 到H的垂直距离

若x为原点，则：g（x）＝w 0为原点到超平面的距离。

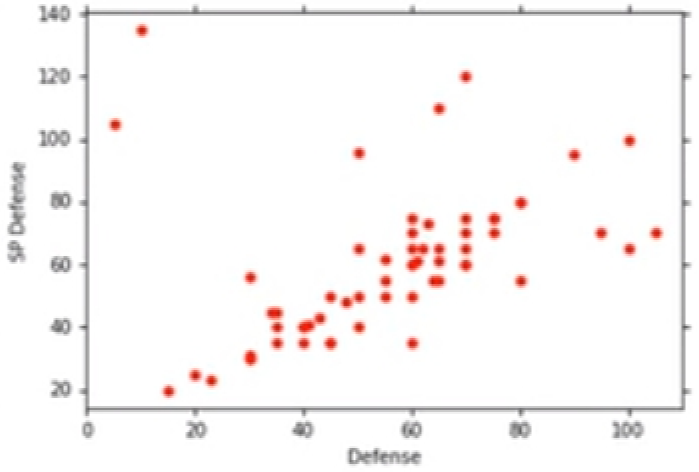
若$w_0>0$，则原点在H的正侧；
若$w_0<0$，则原点在H的负侧；
若$w_0=0$，则H通过原点，具有齐次形式，超平面过原点。

Maximum Likelihood

Class 1: Water



Class 2: Normal



$$\mu^1 = \begin{bmatrix} 75.0 \\ 71.3 \end{bmatrix} \quad \Sigma^1 = \begin{bmatrix} 874 & 327 \\ 327 & 929 \end{bmatrix} \qquad \mu^2 = \begin{bmatrix} 55.6 \\ 59.8 \end{bmatrix} \quad \Sigma^2 = \begin{bmatrix} 847 & 422 \\ 422 & 685 \end{bmatrix}$$

Now we can do classification ☺

$$f_{\mu^1,\Sigma^1}(x) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma^1|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu^1)^T(\Sigma^1)^{-1}(x-\mu^1)\right\}$$

P(C1)
= 79 / (79 + 61) =0.56

$$\mu^1 = \begin{bmatrix} 75.0 \\ 71.3 \end{bmatrix} \quad \Sigma^1 = \begin{bmatrix} 874 & 327 \\ 327 & 929 \end{bmatrix}$$

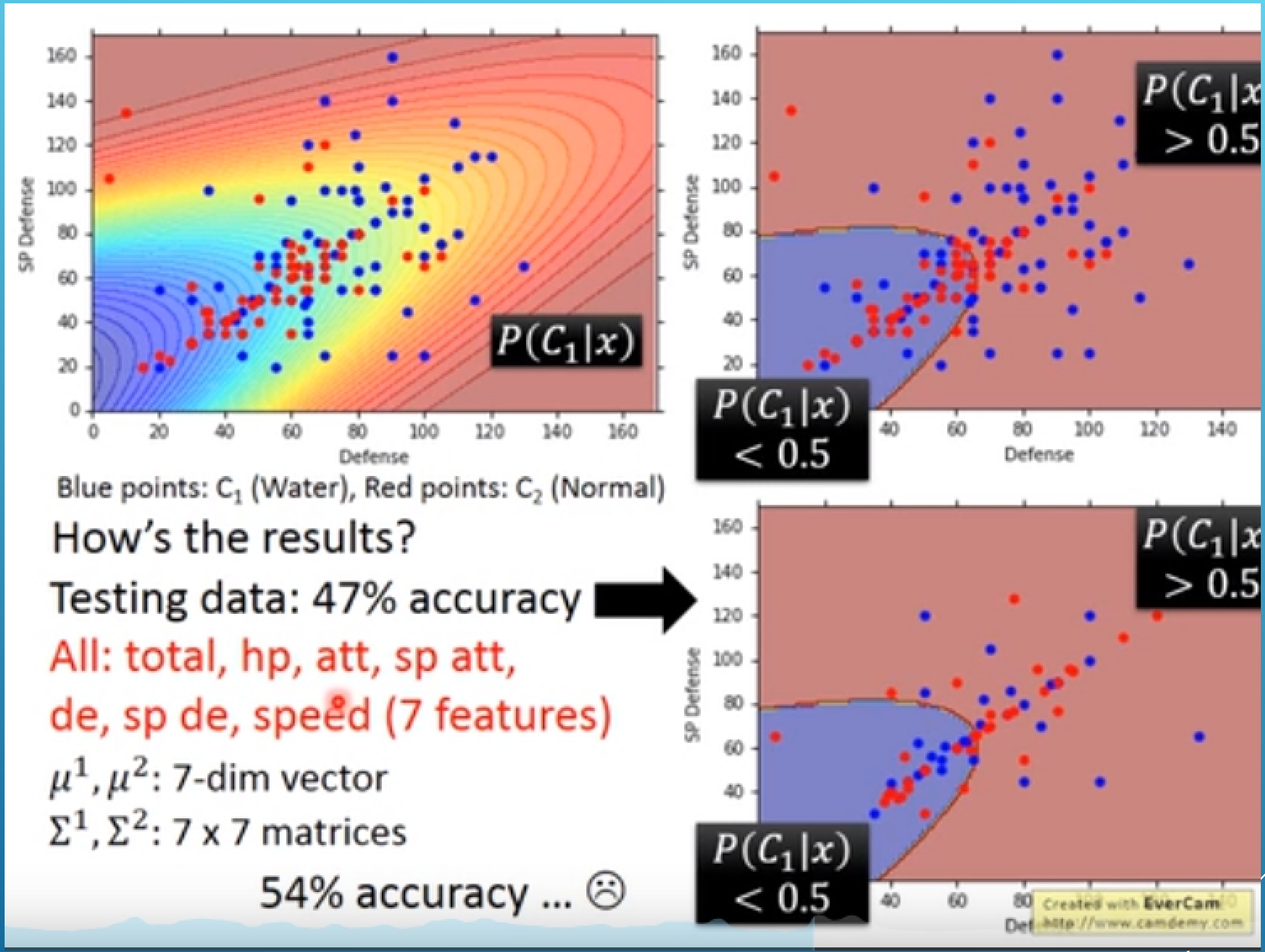
$$P(C_1|x) = \frac{P(x|C_1)P(C_1)}{P(x|C_1)P(C_1) + P(x|C_2)P(C_2)}$$

$$f_{\mu^2,\Sigma^2}(x) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma^2|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu^2)^T(\Sigma^2)^{-1}(x-\mu^2)\right\}$$

P(C2)
= 61 / (79 + 61)
=0.44

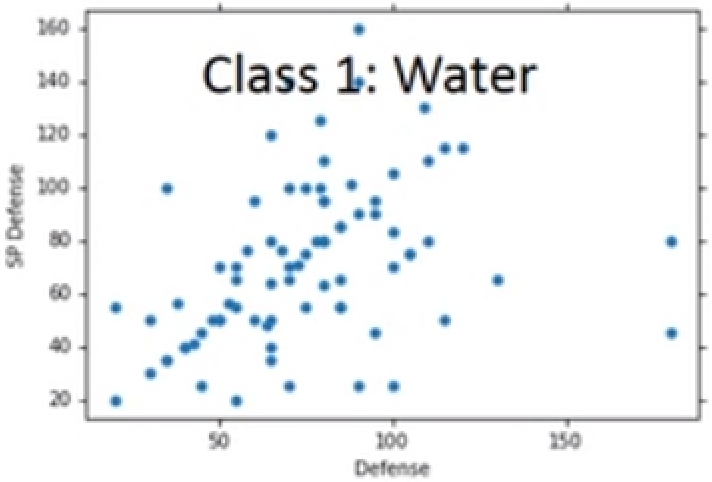
$$\mu^2 = \begin{bmatrix} 55.6 \\ 59.8 \end{bmatrix} \quad \Sigma^2 = \begin{bmatrix} 847 & 422 \\ 422 & 685 \end{bmatrix}$$

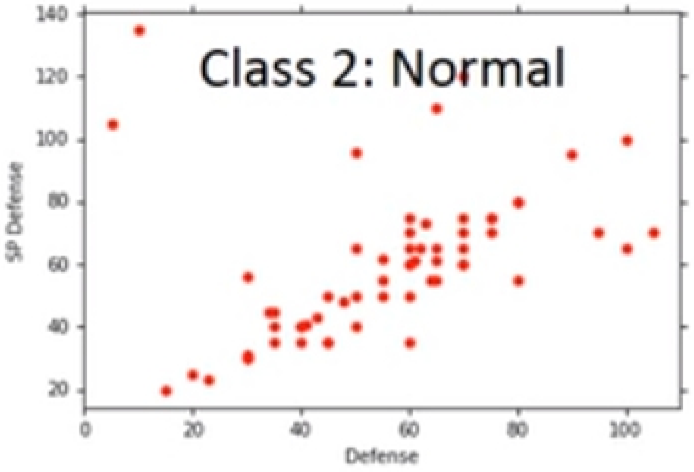
If $P(C_1|x) > 0.5$ ➡ x belongs to class 1 (Water)



SP Defense
Defense
$P(C_1|x)$
$P(C_1|x)$ > 0.5
$P(C_1|x)$ < 0.5
Blue points: C_1 (Water), Red points: C_2 (Normal)
How's the results?
Testing data: 47% accuracy
All: total, hp, att, sp att,
de, sp de, speed (7 features)
μ^1, μ^2: 7-dim vector
Σ^1, Σ^2: 7 x 7 matrices
54% accuracy … ☹
Created with EverCam
http://www.camdemy.com

Modifying Model





$$\mu^1 = \begin{bmatrix} 75.0 \\ 71.3 \end{bmatrix} \quad \Sigma^1 = \begin{bmatrix} 874 & 327 \\ 327 & 929 \end{bmatrix} \qquad \mu^2 = \begin{bmatrix} 55.6 \\ 59.8 \end{bmatrix} \quad \Sigma^2 = \begin{bmatrix} 847 & 422 \\ 422 & 685 \end{bmatrix}$$

The same Σ

Less parameters

Modifying Model

Ref: Bishop, chapter 4.2.2

- Maximum likelihood

"Water" type Pokémons: $x^1, x^2, x^3, \ldots\ldots, x^{79}$

"Normal" type Pokémons: $x^{80}, x^{81}, x^{82}, \ldots\ldots, x^{140}$

μ^1 $\qquad$ Σ $\qquad$ μ^2

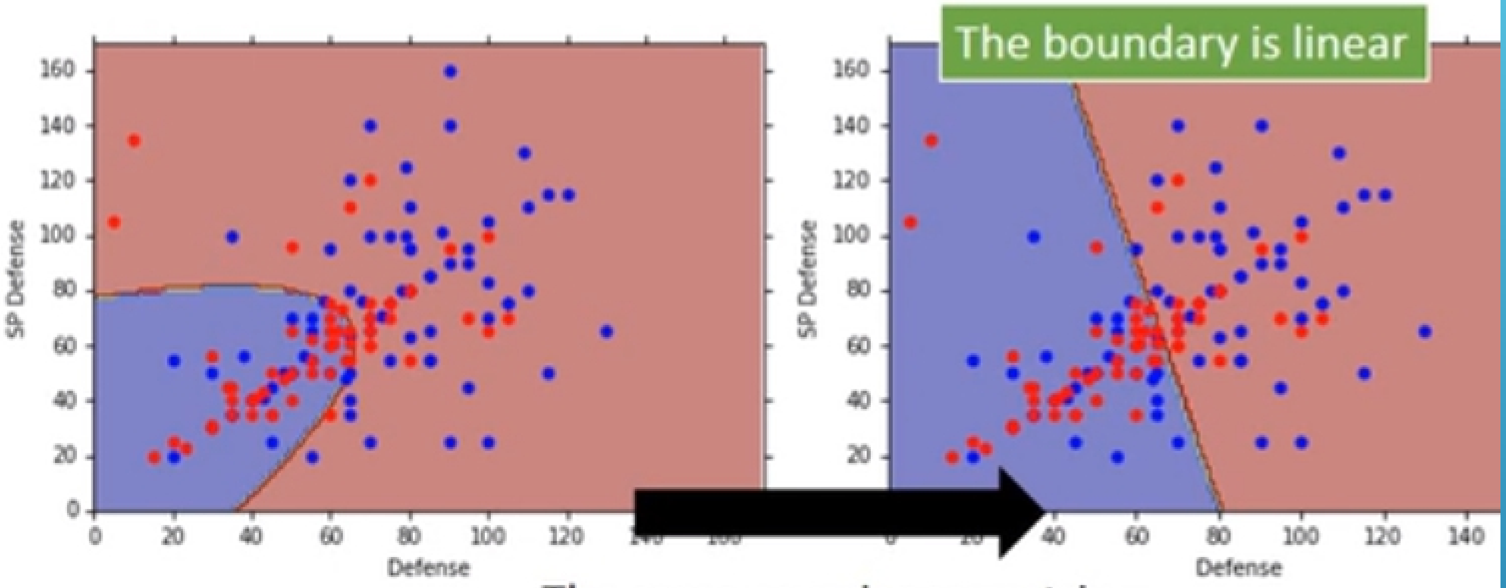
Find μ^1, μ^2, Σ maximizing the likelihood $L(\mu^1, \mu^2, \Sigma)$

$$L(\mu^1,\mu^2,\Sigma) = f_{\mu^1,\Sigma}(x^1) f_{\mu^1,\Sigma}(x^2) \cdots f_{\mu^1,\Sigma}(x^{79}) \\ \times f_{\mu^2,\Sigma}(x^{80}) f_{\mu^2,\Sigma}(x^{81}) \cdots f_{\mu^2,\Sigma}(x^{140})$$

μ^1 and μ^2 is the same

$$\Sigma = \frac{79}{140}\Sigma^1 + \frac{61}{140}\Sigma^2$$

Modifying Model



The same covariance matrix

All: total, hp, att, sp att, de, sp de, speed

54% accuracy → 73% accuracy

Three Steps

- Function Set (Model):

$$x \Rightarrow \quad P(C_1|x) = \frac{P(x|C_1)P(C_1)}{P(x|C_1)P(C_1) + P(x|C_2)P(C_2)}$$

If $P(C_1|x) > 0.5$, output: class 1

Otherwise, output: class 2

- Goodness of a function:
  - The mean μ and covariance Σ that maximizing the likelihood (the probability of generating data)
- Find the best function: easy

Probability Distribution

- You can always use the distribution you like ☺

$$P(x|C_1) = P(x_1|C_1)\ P(x_2|C_1)\ \cdots\cdots\ P(x_k|C_1)\ \cdots\cdots$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \\ \vdots \\ x_K \end{bmatrix}$$

1-D Gaussian

For binary features, you may assume they are from Bernoulli distributions.

If you assume all the dimensions are independent, then you are using *Naive Bayes Classifier*.

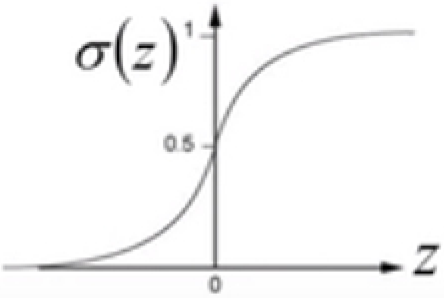
Posterior Probability

$$P(C_1|x) = \frac{P(x|C_1)P(C_1)}{P(x|C_1)P(C_1) + P(x|C_2)P(C_2)}$$

$$= \frac{1}{1 + \frac{P(x|C_2)P(C_2)}{P(x|C_1)P(C_1)}} = \frac{1}{1 + exp(-z)} = \sigma(z)$$

Sigmoid function

$$z = ln\frac{P(x|C_1)P(C_1)}{P(x|C_2)P(C_2)}$$



Posterior Probability

$$P(C_1|x) = \sigma(z) \quad \text{sigmoid} \qquad z = ln\frac{P(x|C_1)P(C_1)}{P(x|C_2)P(C_2)}$$

$$z = ln\frac{P(x|C_1)}{P(x|C_2)} + ln\boxed{\frac{P(C_1)}{P(C_2)}} \Rightarrow \frac{\frac{N_1}{N_1+N_2}}{\frac{N_2}{N_1+N_2}} = \frac{N_1}{N_2}$$

$$P(x|C_1) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\Sigma^1|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu^1)^T(\Sigma^1)^{-1}(x-\mu^1)\right\}$$

$$P(x|C_2) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\Sigma^2|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu^2)^T(\Sigma^2)^{-1}(x-\mu^2)\right\}$$

$$z = \boxed{\ln \frac{P(x|C_1)}{P(x|C_2)}} + \ln \boxed{\frac{P(C_1)}{P(C_2)}} = \frac{N_1}{N_2}$$

$$P(x|C_1) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma^1|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu^1)^T(\Sigma^1)^{-1}(x-\mu^1)\right\}$$

$$P(x|C_2) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma^2|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu^2)^T(\Sigma^2)^{-1}(x-\mu^2)\right\}$$

$$\ln \frac{\cancel{\frac{1}{(2\pi)^{D/2}}} \boxed{\frac{1}{|\Sigma^1|^{1/2}}} \boxed{exp\left\{-\frac{1}{2}(x-\mu^1)^T(\Sigma^1)^{-1}(x-\mu^1)\right\}}}{\cancel{\frac{1}{(2\pi)^{D/2}}} \boxed{\frac{1}{|\Sigma^2|^{1/2}}} \boxed{exp\left\{-\frac{1}{2}(x-\mu^2)^T(\Sigma^2)^{-1}(x-\mu^2)\right\}}}$$

$$= \boxed{\ln \frac{|\Sigma^2|^{1/2}}{|\Sigma^1|^{1/2}}} \boxed{exp\left\{-\frac{1}{2}\left[(x-\mu^1)^T(\Sigma^1)^{-1}(x-\mu^1) - (x-\mu^2)^T(\Sigma^2)^{-1}(x-\mu^2)\right]\right\}}$$

$$= \ln \frac{|\Sigma^2|^{1/2}}{|\Sigma^1|^{1/2}} - \frac{1}{2}\left[(x-\mu^1)^T(\Sigma^1)^{-1}(x-\mu^1) - (x-\mu^2)^T(\Sigma^2)^{-1}(x-\mu^2)\right]$$

$$z = \boxed{ln\frac{P(x|C_1)}{P(x|C_2)}} + ln\boxed{\frac{P(C_1)}{P(C_2)}} = \frac{N_1}{N_2}$$

$$= ln\frac{|\Sigma^2|^{1/2}}{|\Sigma^1|^{1/2}} - \frac{1}{2}\left[\underline{(x-\mu^1)^T(\Sigma^1)^{-1}(x-\mu^1)} - \underline{(x-\mu^2)^T(\Sigma^2)^{-1}(x-\mu^2)}\right]$$

$$(x-\mu^1)^T(\Sigma^1)^{-1}(x-\mu^1)$$

$$= x^T(\Sigma^1)^{-1}x \underline{- x^T(\Sigma^1)^{-1}\mu^1 - (\mu^1)^T(\Sigma^1)^{-1}x} + (\mu^1)^T(\Sigma^1)^{-1}\mu^1$$

$$= x^T(\Sigma^1)^{-1}x \underline{- 2(\mu^1)^T(\Sigma^1)^{-1}x} + (\mu^1)^T(\Sigma^1)^{-1}\mu^1$$

$$(x-\mu^2)^T(\Sigma^2)^{-1}(x-\mu^2)$$

$$= x^T(\Sigma^2)^{-1}x - 2(\mu^2)^T(\Sigma^2)^{-1}x + (\mu^2)^T(\Sigma^2)^{-1}\mu^2$$

$$\boxed{\begin{aligned} z = ln\frac{|\Sigma^2|^{1/2}}{|\Sigma^1|^{1/2}} &- \frac{1}{2}x^T(\Sigma^1)^{-1}x + (\mu^1)^T(\Sigma^1)^{-1}x - \frac{1}{2}(\mu^1)^T(\Sigma^1)^{-1}\mu^1 \\ &+ \frac{1}{2}x^T(\Sigma^2)^{-1}x - (\mu^2)^T(\Sigma^2)^{-1}x + \frac{1}{2}(\mu^2)^T(\Sigma^2)^{-1}\mu^2 + ln\frac{N_1}{N_2} \end{aligned}}$$

$P(C_1|x) = \sigma(z)$

$$\boxed{\begin{aligned} z = & \cancel{ln\frac{|\Sigma^2|^{1/2}}{|\Sigma^1|^{1/2}}} \cancel{-\frac{1}{2}x^T(\Sigma^1)^{-1}x} + (\mu^1)^T(\Sigma^1)^{-1}x - \frac{1}{2}(\mu^1)^T(\Sigma^1)^{-1}\mu^1 \\ & \cancel{+\frac{1}{2}x^T(\Sigma^2)^{-1}x} - (\mu^2)^T(\Sigma^2)^{-1}x + \frac{1}{2}(\mu^2)^T(\Sigma^2)^{-1}\mu^2 + ln\frac{N_1}{N_2} \end{aligned}}$$

$\Sigma_1 = \Sigma_2 = \Sigma$

$$z = \underbrace{(\mu^1 - \mu^2)^T\Sigma^{-1}}_{\boldsymbol{w}^T} x \underbrace{- \frac{1}{2}(\mu^1)^T(\Sigma^1)^{-1}\mu^1 + \frac{1}{2}(\mu^2)^T(\Sigma^2)^{-1}\mu^2 + ln\frac{N_1}{N_2}}_{\text{b}}$$

$P(C_1|x) = \sigma(w \cdot x + b)$ How about directly find ***w*** and b?

In generative model, we estimate $N_1, N_2, \mu^1, \mu^2, \Sigma$

Then we have ***w*** and b

softmax函数，也称指数归一化函数，它是一种logistic函数的归一化，可以将［公式］维实数向量压缩成范围（0~1）的［公式］维实数向量函数形式为

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

其中分母指归一化的作用，取指数的原因，第一是模拟max的行为，即使得大的数值更大，第二是，方便运算求导

https://www.codercto.com/a/47514.html

Softmax回归

- softmax回归是logistic回归的一般化，适用于K分类的问题，第k类的参数为向量θ_k，组成的二维矩阵为θ_{k*n}；
- softmax函数的本质就是将一个K维的任意实数向量压缩（映射）成另一个K维的实数向量，其中向量中的每个元素取值都介于（0，1）之间。
- softmax回归概率函数为：

$$p(y=k \mid x;\theta)=\frac{e^{\theta_k^T x}}{\sum_{l=1}^{K} e^{\theta_l^T x}}, k=1,2\cdots,K$$

Softmax算法原理

$$p(y=k\mid x;\theta)=\frac{e^{\theta_k^T x}}{\sum_{l=1}^{K} e^{\theta_l^T x}},k=1,2\cdots,K$$

$$h_\theta(x)=\begin{bmatrix} p\left(y^{(i)}=1\mid x^{(i)};\theta\right) \\ p\left(y^{(i)}=2\mid x^{(i)};\theta\right) \\ \cdots \\ p\left(y^{(i)}=k\mid x^{(i)};\theta\right) \end{bmatrix}=\frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}}\begin{bmatrix} e^{\theta_1^T x} \\ e^{\theta_2^T x} \\ \cdots \\ e^{\theta_k^T x} \end{bmatrix}\Longrightarrow\theta=\begin{bmatrix} \theta_{11} & \theta_{12} & \cdots & \theta_{1n} \\ \theta_{21} & \theta_{22} & \cdots & \theta_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \theta_{k1} & \theta_{k2} & \cdots & \theta_{kn} \end{bmatrix}$$

Softmax算法损失函数

$$J(\theta)=-\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}I\left(y^{(i)}=j\right)\ln\left(\frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k}e^{\theta_l^T x^{(i)}}}\right)\qquad I\left(y^{(i)}=j\right)=\begin{cases}1, & y^{(i)}=j\\0, & y^{(i)}\neq j\end{cases}$$

Softmax算法梯度下降法求解

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k} I\left(y^{(i)} = j\right)\ln\left(\frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right)$$

$$I\left(y^{(i)} = j\right) = \begin{cases} 1, & y^{(i)} = j \\ 0, & y^{(i)} \neq j \end{cases}$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} - I\left(y^{(i)} = j\right)\ln\left(\frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right)$$

$$= \frac{\partial}{\partial \theta_j} - I\left(y^{(i)} = j\right)\left(\theta_j^T x^{(i)} - \ln\left(\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}\right)\right)$$

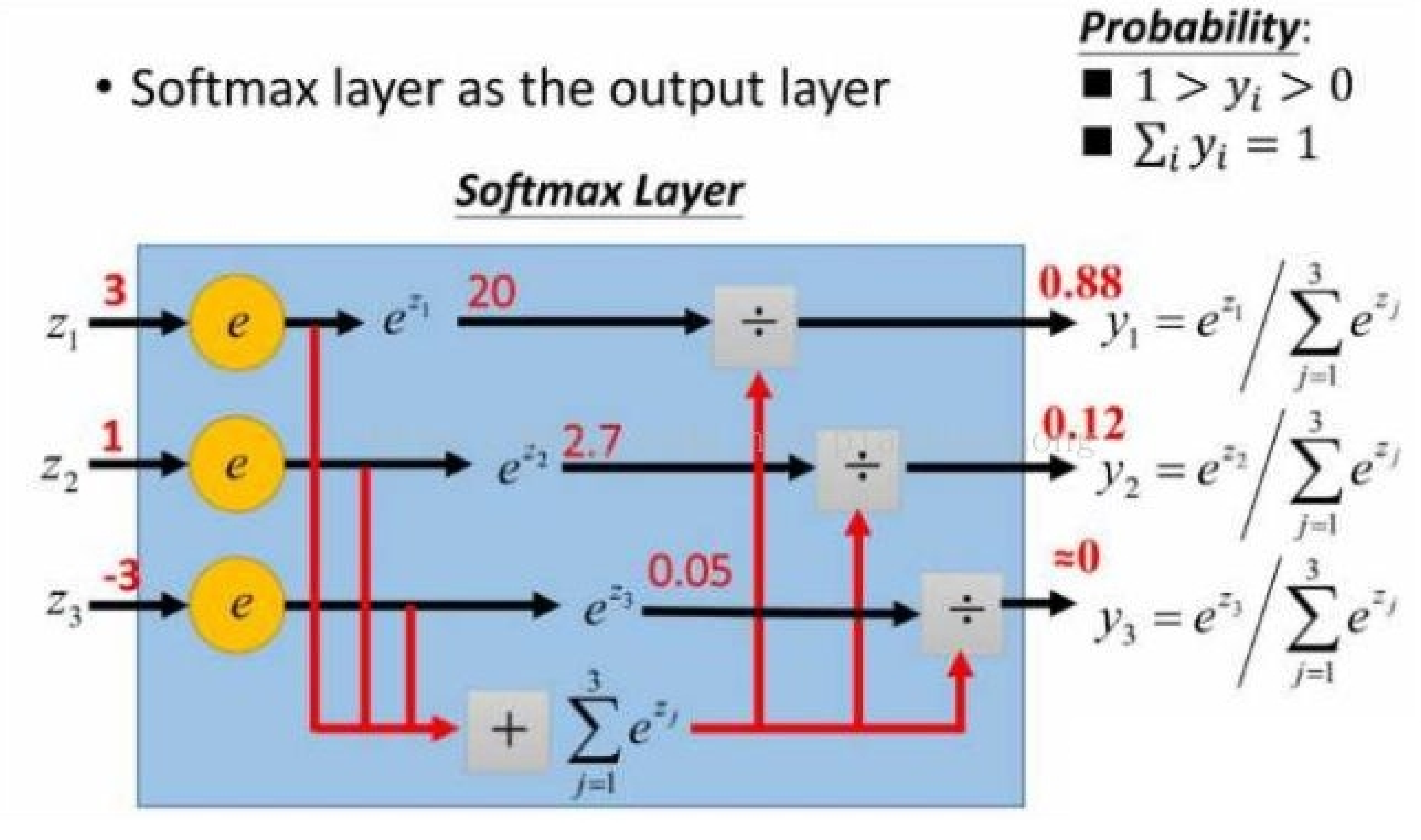
$$= -I\left(y^{(i)} = j\right)\left(1 - \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right)x^{(i)}$$

Softmax算法梯度下降法求解

$$\frac{\partial}{\partial \theta_j} J(\theta) = -I\left(y^{(i)} = j\right)\left(1 - \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right)x^{(i)}$$

$$\theta_j = \theta_j + \alpha \sum_{i=1}^{m} I\left(y^{(i)} = j\right)\left(1 - p\left(y^{(i)} = j \middle| x^{(i)}; \theta\right)\right)x^{(i)}$$

$$\theta_j = \theta_j + \alpha I\left(y^{(i)} = j\right)\left(1 - p\left(y^{(i)} = j \middle| x^{(i)}; \theta\right)\right)x^{(i)}$$


• Softmax layer as the output layer
Probability:
$1 > y_i > 0$
$\sum_i y_i = 1$
Softmax Layer
z_1
3
z_2
1
z_3
-3
e
e
e
e^{z_1}
20
e^{z_2}
2.7
e^{z_3}
0.05
$\sum_{j=1}^{3} e^{z_j}$
0.88
0.12
≈0
$y_1 = e^{z_1} \Big/ \sum_{j=1}^{3} e^{z_j}$
$y_2 = e^{z_2} \Big/ \sum_{j=1}^{3} e^{z_j}$
$y_3 = e^{z_3} \Big/ \sum_{j=1}^{3} e^{z_j}$