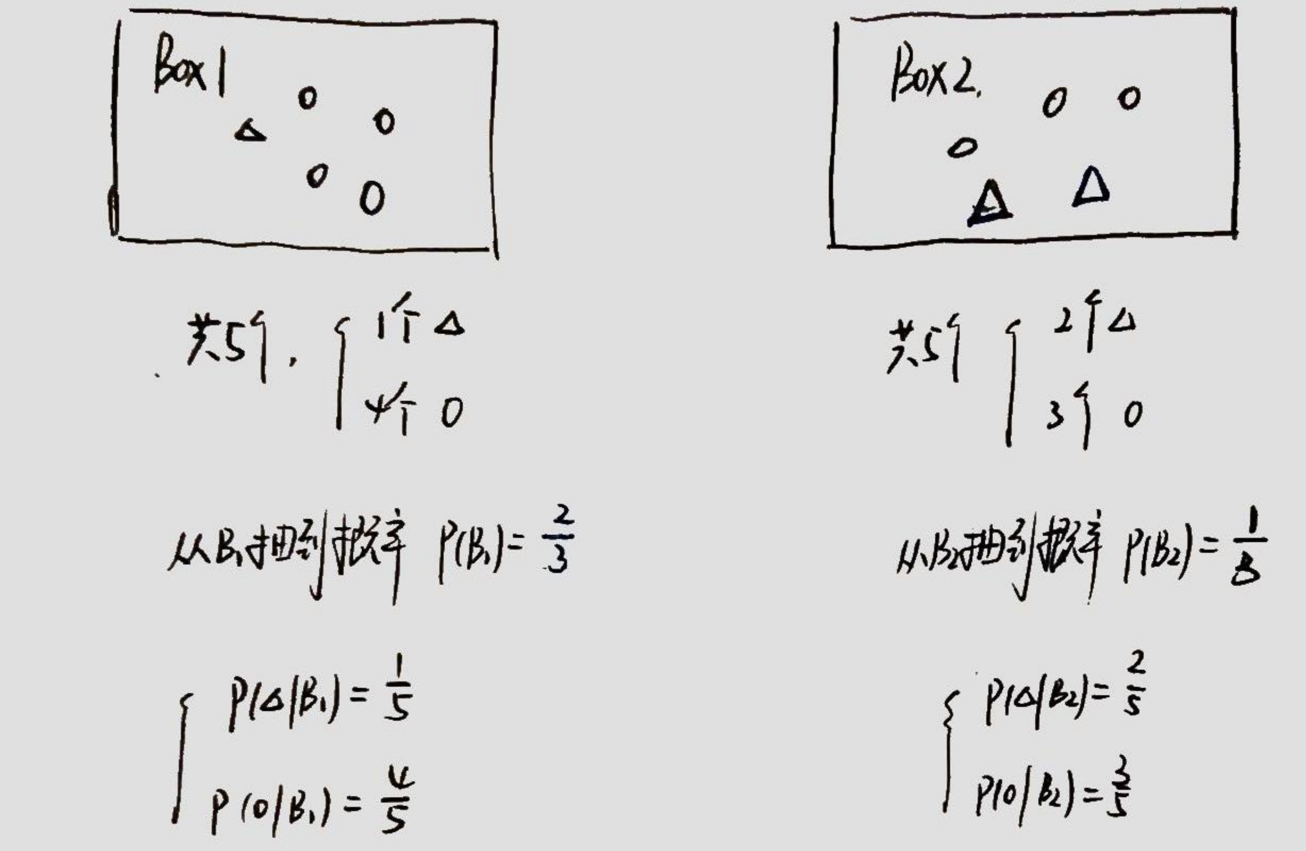
分类问题-Logistic Regression

汇报人：王茂南

时间：2019年4月29日

例子引入

- 现在有两个Box，里面装有一些○和▲（个数如下图所示）；
- 抽到Box1的概率为2/3，抽到Box2的概率为1/3；



例子引入

- 现在的问题是，如果抽到了一个▲，问是来自Box1还是Box2；
- 我们可以使用贝叶斯公式进行求解；

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B|A)P(A) + P(B|A^C)P(A^C)}$$

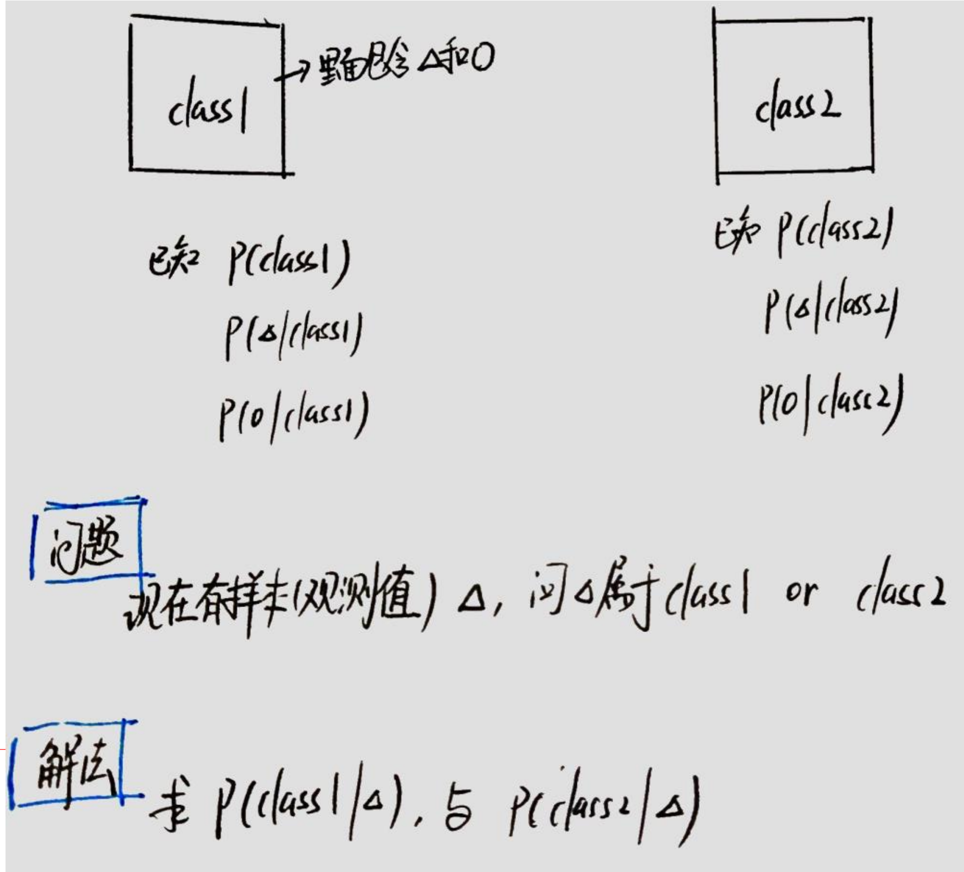
若现在抽一个方△，问来自B_1还是B_2.

发生的(观察到的现象)

即求 $$P(B_1|\triangle) = \frac{P(\triangle|B_1)\cdot P(B_1)}{P(\triangle|B_1)P(B_1)+P(\triangle|B_2)\cdot P(B_2)}$$

转换为分类问题

- 如果在分类的问题里面，上面的Box1和Box2代表class1和class2；
- 如果我们抽到的是三角，我们只需要计算P(class1|三角)和P(class2|三角)的概率大小进行比较即可。



数学式子求解

- 于是，求解分类问题，转换为求解上面的贝叶斯式子，根据解决的方法，可以分为两个类别：
  - Generative Model：P(x)=P(x|c1)*P(c1)+P(x|c2)*P(c2)，我们可以求出x的概率分布，从而模拟x的生成；
  - Logistic Model：与线性回归思想类似；

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

Generative Model. (计算 $P(x|C_1)\cdot P(x|C_2)$)

Logistic Regression.

Generative Model

- Generative Model的想法是求出表达式中的所有概率;
  - P(c1),P(c2)
  - P(x|c1),P(x|c2)

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

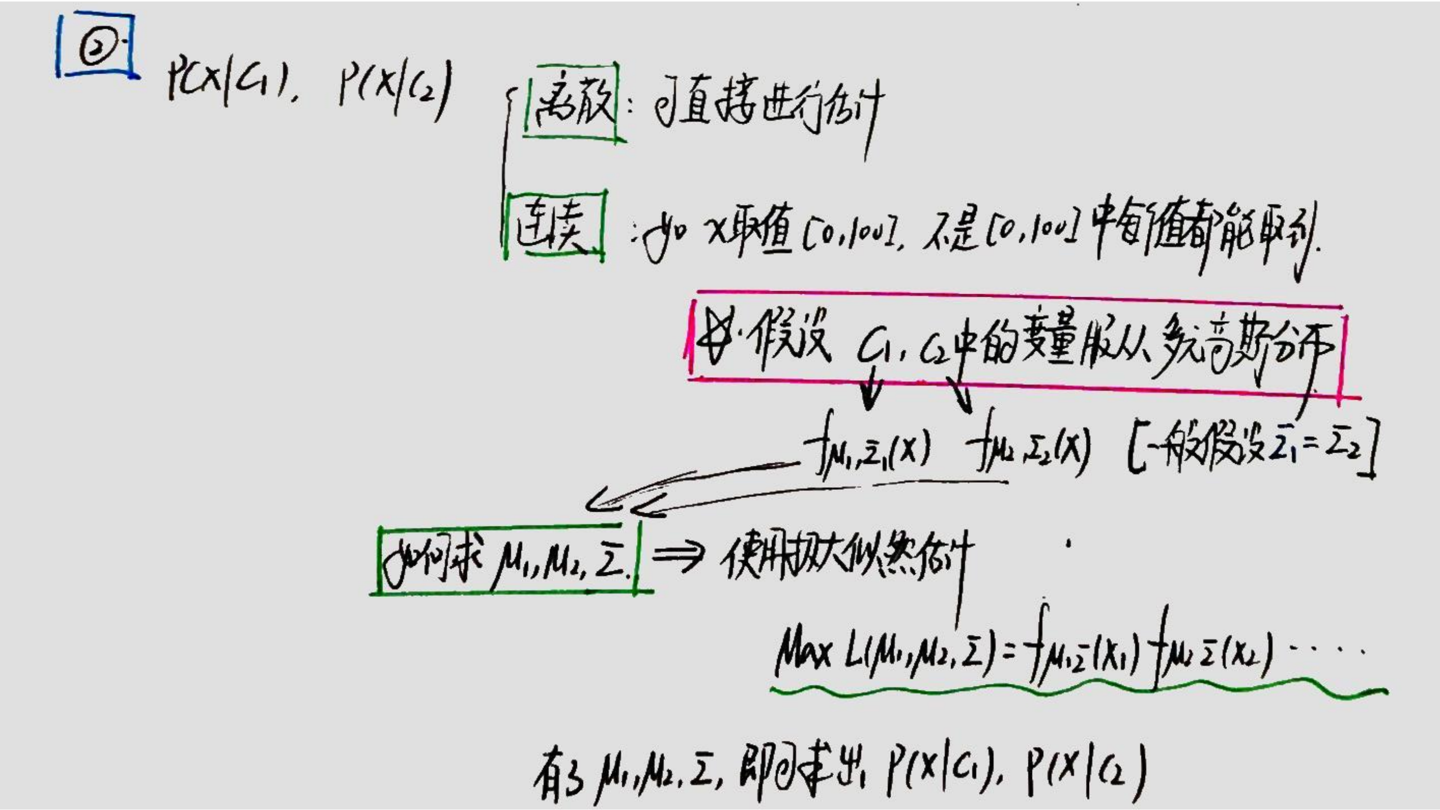
现在需要从样本计算 $\begin{cases} P(C_1),\ P(C_2) \\ P(x|C_1),\ P(x|C_2) \end{cases}$

$\boxed{①}$ $P(C_1)$, $P(C_2)$ 直接计算

如 $P(C_1) = \dfrac{\text{样本中}C_1\text{的个数}}{\text{样本总个数}}$

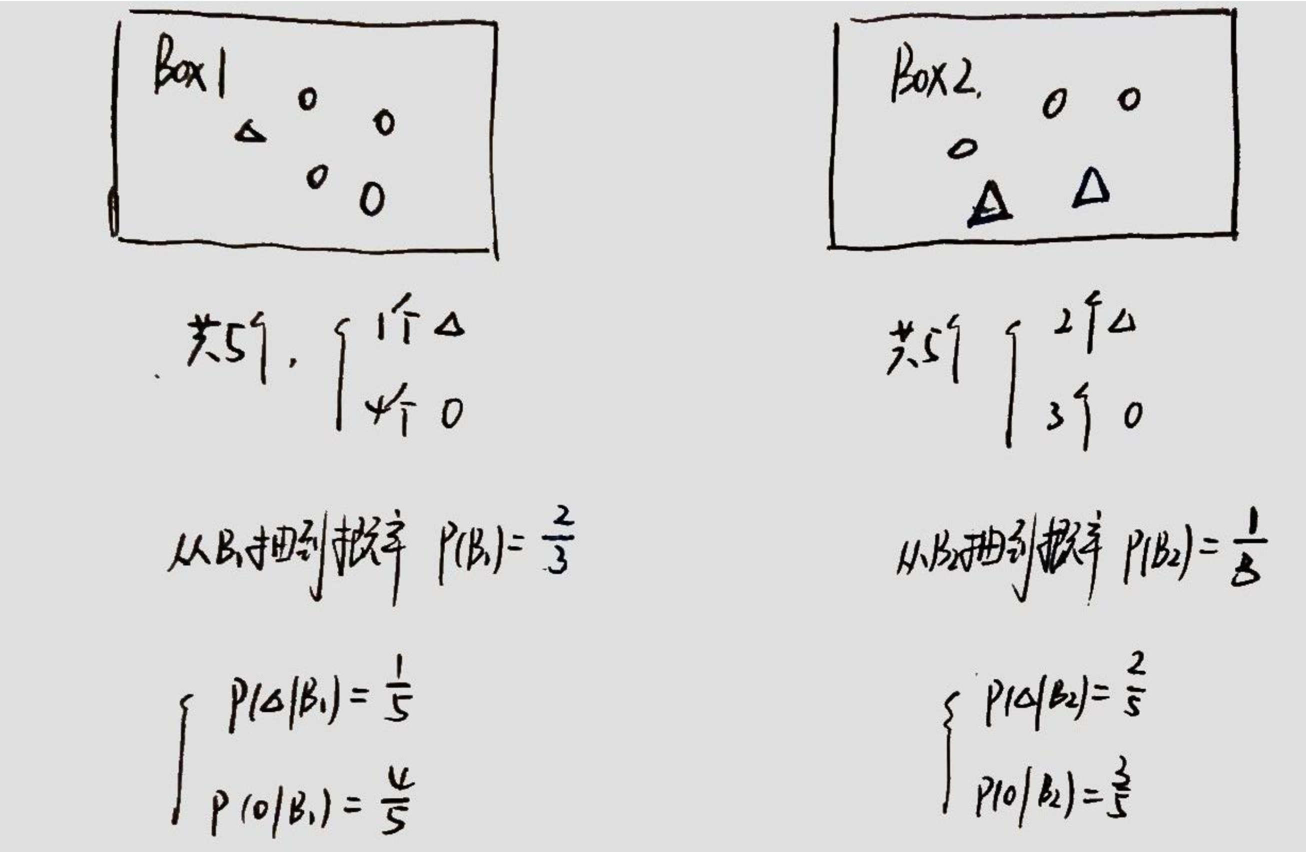
Generative Model

- Generative Model的想法是求出表达式中的所有概率;
  - P(c1),P(c2)
  - P(x|c1),P(x|c2)
    - 离散值
    - 连续值



离散变量解释

- 现在有两个Box，里面装有一些○和▲（个数如下图所示）；
- 抽到Box1的概率为2/3，抽到Box2的概率为1/3；



Logistic Model

- 我们从贝叶斯的表达式出发进行化简：

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

$$\overset{\text{同除分子}}{=} \frac{1}{1 + \frac{P(x|C_2)\cdot P(C_2)}{P(x|C_1)\cdot P(C_1)}}$$

令 $z = \ln \frac{P(x|C_1)\cdot P(C_1)}{P(x|C_2)\cdot P(C_2)}$ 则 $P(C_1|x) = \frac{1}{1+e^{-z}}$

Logistic Model

- 我们从贝叶斯的表达式出发进行化简；
  - 注意：这是其中一种的解释方式，我们可以从别的解释同样推导出logistic model的表达式；
  - 如：我们可以认为Logistic Model需要输出一个概率，于是我们使用Sigmoid函数做压缩，把(负无穷，正无穷)的值压缩到(0,1)

如何用高斯分布 $f_{\mu_1,\Sigma}(x)$. $f_{\mu_2,\Sigma}(x)$ 代入z，之可以化简如下表达式

$$z = \underbrace{(\mu_1-\mu_2)\Sigma^{-1}}_{w} x \underbrace{- \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_2 + \frac{1}{2}\mu_2^T\Sigma^{-1}\mu_1 + \ln\frac{N_1}{N_2}}_{b}$$

这是两个参数

$$\therefore P(C_1|x) = \frac{1}{1+e^{wx+b}} = \sigma(z),\quad 其中 \begin{cases} \sigma(x) = \frac{1}{1+e^{-x}} \\ z = wx+b \end{cases}$$

详细推导过程

下面对z进行化简

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

$$= \ln \frac{P(x|C_1)}{P(x|C_2)} + \ln \frac{P(C_1)}{P(C_2)} \qquad P(C_1) \to \frac{N_1}{N_1+N_2},\quad P(C_2) \to \frac{N_2}{N_1+N_2}$$

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

将 $P(x|C_1)=f_{\mu_1,\Sigma_1}(x)$, $P(x|C_2)=f_{\mu_2,\Sigma_2}(x)$ 代入

$$= \ln \frac{f_{\mu_1,\Sigma_1}(x)}{f_{\mu_2,\Sigma_2}(x)} + \ln \frac{N_1}{N_2}$$

$$= \ln \frac{\cancel{\frac{1}{(2\pi)^{D/2}}}\frac{1}{|\Sigma_1|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)}}{\cancel{\frac{1}{(2\pi)^{D/2}}}\frac{1}{|\Sigma_2|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)}} + \ln \frac{N_1}{N_2}$$

$$= \ln \left[\frac{|\Sigma_2|^{\frac{1}{2}}}{|\Sigma_1|^{\frac{1}{2}}} e^{-\frac{1}{2}\left[(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)-(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)\right]}\right] + \ln \frac{N_1}{N_2}$$

$$= \ln \frac{|\Sigma_2|^{\frac{1}{2}}}{|\Sigma_1|^{\frac{1}{2}}} \underbrace{- \frac{1}{2}\left[(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)-(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)\right]}_{\text{展开}} + \ln \frac{N_1}{N_2}$$

$$= \ln \frac{|\Sigma_2|^{\frac{1}{2}}}{|\Sigma_1|^{\frac{1}{2}}} - \frac{1}{2}x^T(\Sigma_1)^{-1}x + (\mu_1)^T\Sigma_1^{-1}x - \frac{1}{2}\mu_1^T\Sigma_1^{-1}\mu_1$$
$$+ \frac{1}{2}x^T(\Sigma_2)^{-1}x - (\mu_2)^T(\Sigma_2)^{-1}x + \frac{1}{2}\mu_2^T\Sigma_2^{-1}\mu_2$$
$$+ \ln \frac{N_1}{N_2}$$

时 $\Sigma_1=\Sigma_2$，可以进一步化简

$$z = \underline{\overset{1\times n}{(\mu_1-\mu_2)}\,\overset{n\times n}{\Sigma^{-1}}}\,x \underbrace{- \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \frac{1}{2}\mu_2^T\Sigma^{-1}\mu_2 + \ln\frac{N_1}{N_2}}_{b}$$

Logistic Model

- 接下来需要求解Logistic Model中的w和b

$\boxed{\text{关于评价函数}}$

现在问题：$f_{w,b}(x) = \frac{1}{1+e^{-z}}$ 其中 $z=wx+b$ [$f_{w,b}(x)$表示 x 属于 C_1 的概率]

要求 w, b，可以用最大似然估计.

如：有数据如下：

data	x_1	x_2	x_3	...
label	C_1	C_2	C_1	...

$$\boxed{\text{Max}}\ L(w,b) = f_{w,b}(x_1)\cdot[1-f_{w,b}(x_2)]\,f_{w,b}(x_3)\ \cdots$$

Logistic Model

接下来需要求解Logistic Model中的w和b => Cross Entropy(交叉熵)

$$\boxed{Max}\ L(w,b) = f_{w,b}(x_1) \cdot [1 - f_{w,b}(x_2)]\, f_{w,b}(x_3) \cdots$$

乘法转加法，极大转极小

$$\boxed{Min}\ -\ln L(w,b) = -\ln f_{w,b}(x_1) \xleftrightarrow{\text{等价}} -[y_1 \ln f(x_1) + (1-y_1)\ln(1-f(x_1))]$$

$$-\ln(1 - f_{w,b}(x_2)) \xleftrightarrow{\text{等价}} -[y_2 \ln f(x_2) + (1-y_2)\ln(1-f(x_2))]$$

$$-\ln f_{w,b}(x_3) \xleftrightarrow{\text{等价}} -[y_3 \ln f(x_3) + (1-y_3)\ln(1-f(x_3))]$$

于是，即要求

$$\boxed{Min\ -\ln L(w,b) = \sum_{i=1}^{n} -y_i \ln f(x_i) - (1-y_i)\ln(1-f(x_i))}\ \boxed{\text{Cross Entropy}}$$

使用梯度下降可求出最优解.

	Logistic Regression	Linear Regression
Step 1:	$f_{w,b}(x) = \sigma\left(\sum_i w_i x_i + b\right)$ Output: between 0 and 1	$f_{w,b}(x) = \sum_i w_i x_i + b$ Output: any value
Step 2:	Training data: $(x^n, \hat{y}^n)$ $\boxed{\hat{y}^n\text{: 1 for class 1, 0 for class 2}}$ $L(f) = \sum_n C(f(x^n), \hat{y}^n)$	Training data: $(x^n, \hat{y}^n)$ $\hat{y}^n$: a real number $L(f) = \frac{1}{2}\sum_n (f(x^n) - \hat{y}^n)^2$

Step 3:

Logistic regression: $w_i \leftarrow w_i - \eta \sum_n -\underline{\left(\hat{y}^n - f_{w,b}(x^n)\right)} x_i^n$

Linear regression: $w_i \leftarrow w_i - \eta \sum_n -\underline{\left(\hat{y}^n - f_{w,b}(x^n)\right)} x_i^n$

下面几页PPT出处：http://speech.ee.ntu.edu.tw/~tlkagk/courses_ML17_2.html

多分类问题—Softmax介绍

C$_1$: w^1, b_1　　$z_1 = w^1 \cdot x + b_1$

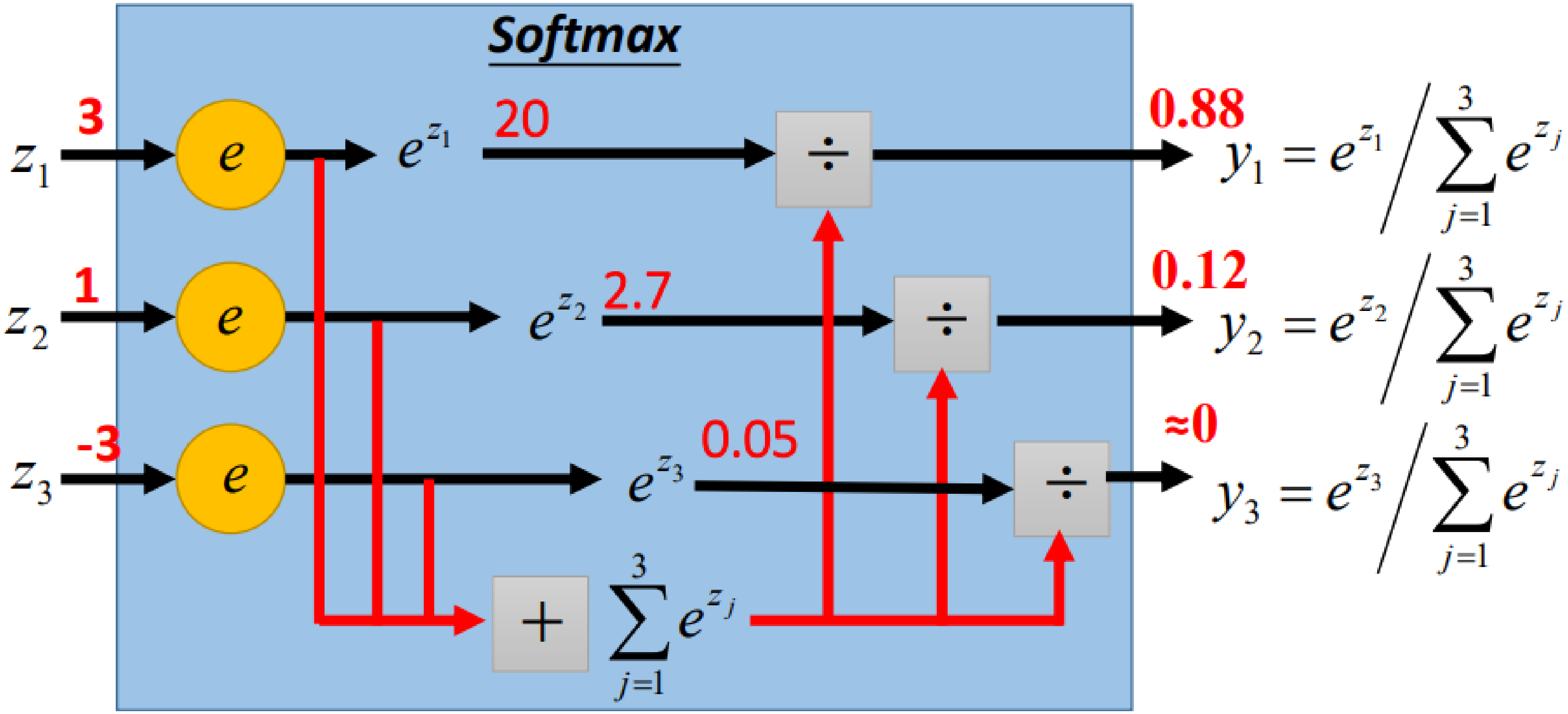
C$_2$: w^2, b_2　　$z_2 = w^2 \cdot x + b_2$

C$_3$: w^3, b_3　　$z_3 = w^3 \cdot x + b_3$

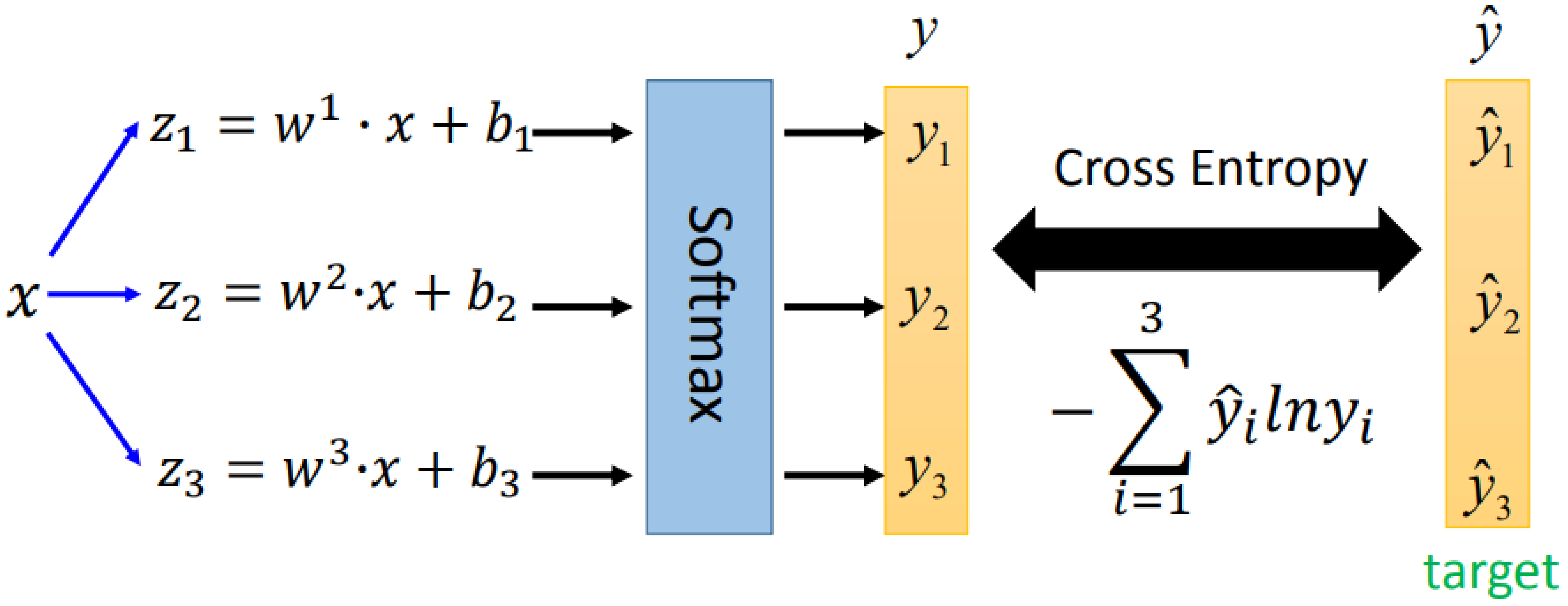
Probability:

- $1 > y_i > 0$
- $\sum_i y_i = 1$

$$y_i = P(C_i \mid x)$$



多分类问题—Softmax介绍



If x ∈ class 1

$$\hat{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

If x ∈ class 2

$$\hat{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

If x ∈ class 3

$$\hat{y} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

二分类与多分类关系

$$z_1 = w_1 x + b_1 \longrightarrow y_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2}}$$

$$z_2 = w_2 x + b_2 \longrightarrow y_2 = \frac{e^{z_2}}{e^{z_1} + e^{z_2}}$$

$\boxed{\text{在两个变量，只需比较 } z_1, z_2 \text{ 大小}}$

$\frac{z_1}{z_2} > 1$ 则 class1

$$z = \frac{z_1}{z_2} = wx + b \longrightarrow y_1 = \frac{e^z}{e^z + 1} \xlongequal{\text{同除}e^z} \frac{1}{1 + e^{-z}}$$

$$1 \longrightarrow y_2 = \frac{1}{e^z + 1}$$

⇒ 这样有了 y 的表达式.

二分类与多分类关系

Cross Entropy (一个样本时)

$$-\sum_{i=1}^{2} \hat{y}_i \ln y_i \quad \frac{\text{时 } \hat{y}_1+\hat{y}_2=1}{y_1+y_2=1} \quad \hat{y}\ln y + (1-\hat{y})\ln y$$

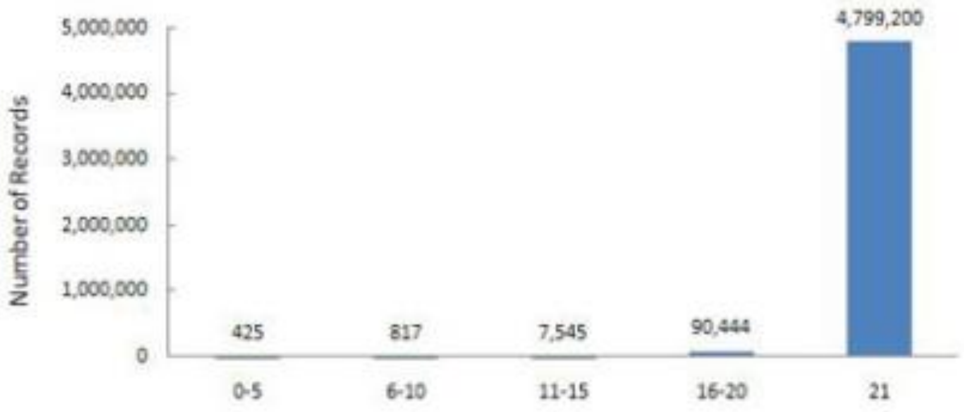
$$\Rightarrow \begin{cases} \text{其中 } \hat{y} \text{ 为 label} \\ y = f(x) \end{cases}$$

Logistic Model实验

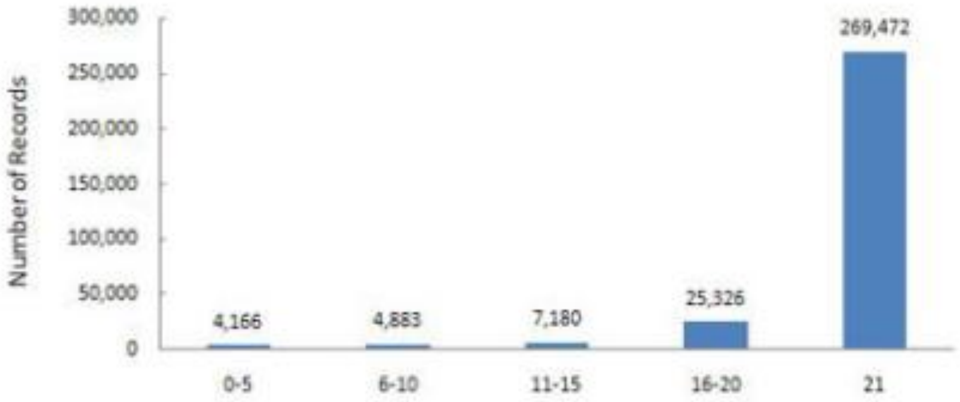
- 数据集:The NSL-KDD Data Set
- 训练集：The full NSL-KDD train set including all difficulty level in CSV format.
  - 离散变量使用One-hot编码;
  - 离散变量不使用One-hot编码;
- 测试集
  - KDDTest：The full NSL-KDD test set including attack-type labels and difficulty level in CSV format.
  - KDDTest-21：A subset of the KDDTest+.txt file which does not include records with difficulty level of 21 out of 21
- 关于difficulty level介绍
  - 作者在NSL-KDD的时候，对每一条数据集跑了21个算法；
  - difficulty level个数表示分类成功的次数;
  - 即difficulty level越小，这条数据被分对的次数越小(也就是越难);

Logistic Model实验

- 关于difficulty level介绍
  - 作者在NSL-KDD的时候，对每一条数据集跑了21个算法；
  - difficulty level个数表示分类成功的次数;
  - 即difficulty level越小， 这条数据被分对的次数越小(也就是越难);



The distribution of #*successfulPrediction* values for the KDD train set records



The distribution of #*successfulPrediction* values for the KDD test set records

实验结果

	KDDTest-21	KDDTest
使用One-hot编码	*********** Training complete in 0m 31s Best val Acc: 0.974200 *********** 在测试集上进行测试. Accuracy of the network on the 11850 test Data: 57.6456 %	*********** Training complete in 0m 31s Best val Acc: 0.972930 *********** 在测试集上进行测试. Accuracy of the network on the 22544 test Data: 79.1608 %
不使用One-hot编码	*********** Training complete in 0m 32s Best val Acc: 0.954116 *********** 在测试集上进行测试. Accuracy of the network on the 11850 test Data: 56.4895 %	*********** Training complete in 0m 37s Best val Acc: 0.954989 *********** 在测试集上进行测试. Accuracy of the network on the 22544 test Data: 78.3357 %

局限性：Logistic Model是只能进行线性分割的.

实验结果—疑问

- 使用Full的训练集，使用One-hot编码；
- Hidden-layer加到5层数;
- Dropout(0.9), 使用BN;

```
**********
Training complete in 4m 52s
Best val Acc: 0.978011
**********
在测试集上进行测试.
Accuracy of the Best_model network on the 22544 test Data: 78.0075 %
Accuracy of the Final_model network on the 22544 test Data: 77.5949 %
```

```
NeuralNet(
  (inLayer): Linear(in_features=122, out_features=100, bias=True)
  (relu): ReLU()
  (hiddenLayer): Sequential(
    (0): Linear(in_features=100, out_features=100, bias=True)
    (1): BatchNorm1d(100, eps=1e-05, momentum=0.5, affine=True, track_running_stats=True)
    (2): Dropout(p=0.9)
    (3): ReLU()
    (4): Linear(in_features=100, out_features=100, bias=True)
    (5): BatchNorm1d(100, eps=1e-05, momentum=0.5, affine=True, track_running_stats=True)
    (6): Dropout(p=0.9)
    (7): ReLU()
    (8): Linear(in_features=100, out_features=100, bias=True)
    (9): BatchNorm1d(100, eps=1e-05, momentum=0.5, affine=True, track_running_stats=True)
    (10): Dropout(p=0.9)
    (11): ReLU()
    (12): Linear(in_features=100, out_features=100, bias=True)
    (13): BatchNorm1d(100, eps=1e-05, momentum=0.5, affine=True, track_running_stats=True)
    (14): Dropout(p=0.9)
    (15): ReLU()
    (16): Linear(in_features=100, out_features=100, bias=True)
    (17): BatchNorm1d(100, eps=1e-05, momentum=0.5, affine=True, track_running_stats=True)
    (18): Dropout(p=0.9)
    (19): ReLU()
  )
  (outLayer): Linear(in_features=100, out_features=2, bias=True)
)
```

实验结果—疑问

- 用测试集做训练，训练集做测试；

```
***********
Training complete in 4m 52s
Best val Acc: 0.978011
***********
在测试集上进行测试.
Accuracy of the Best_model network on the 22544 test Data: 78.0075 %
Accuracy of the Final_model network on the 22544 test Data: 77.5949 %
```

```
***********
Training complete in 0m 54s
Best val Acc: 0.891748
***********
在测试集上进行测试.
Accuracy of the Best_model network on the 125973 test Data: 88.6420 %
Accuracy of the Final_model network on the 125973 test Data: 85.7652 %
```

- 我自己感觉test中出现了train中没有的数据；
- 在KDD99中，**测试集**中出现了**训练集**中没有的攻击；
- 对于NSL-KDD，还没有验证！！！