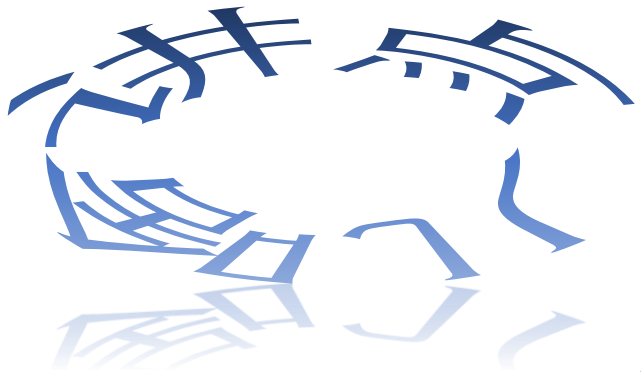
第4章 逻辑回归

Logistic Regression

电子工程学院、人工智能学院
college of Electronic Engineering、college of Artificial Intelligence



- Logistic回归模型
- Logistic回归的损失函数
- Logistic回归的优化算法
- Logistic回归的多类分类任务
- 分类任务中的样本不均衡问题
- 分类模型的评价指标

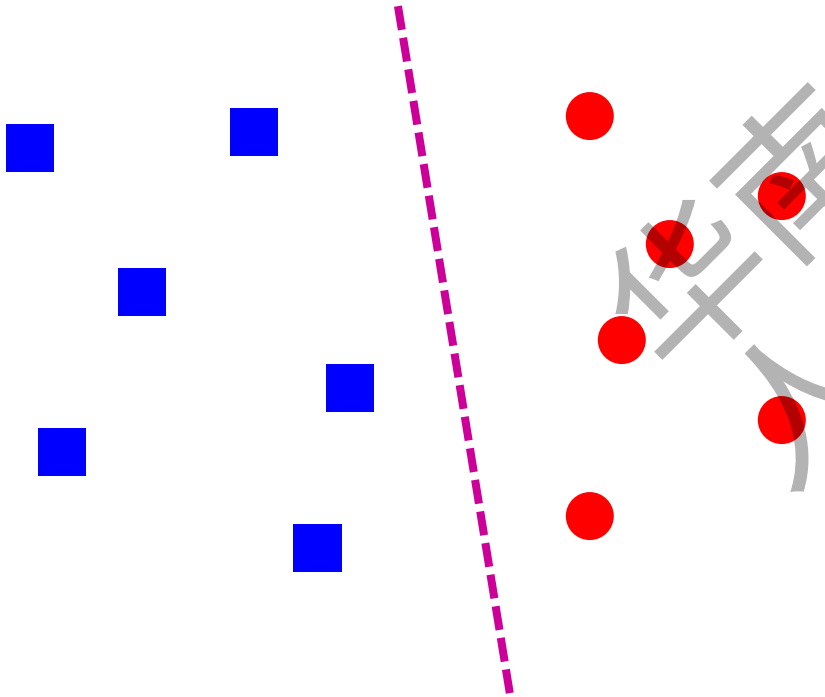
1. Sigmoid函数

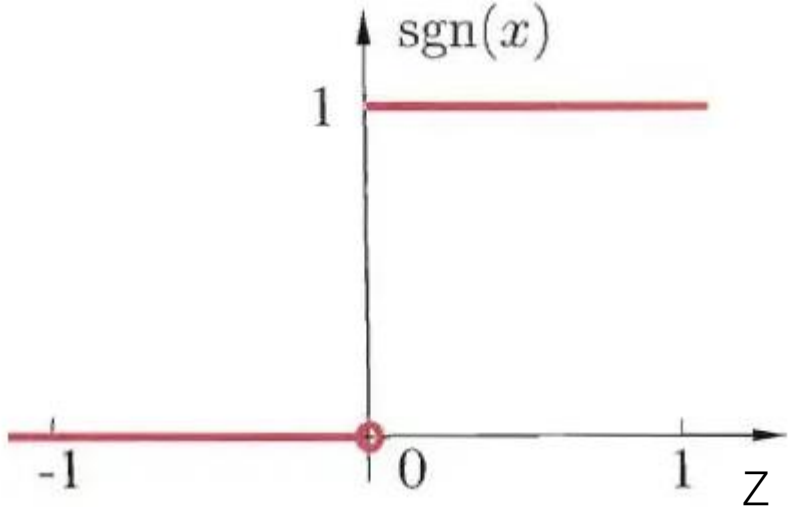
线性分类器Linear classifier

$$z = b + \sum w_i x_i$$

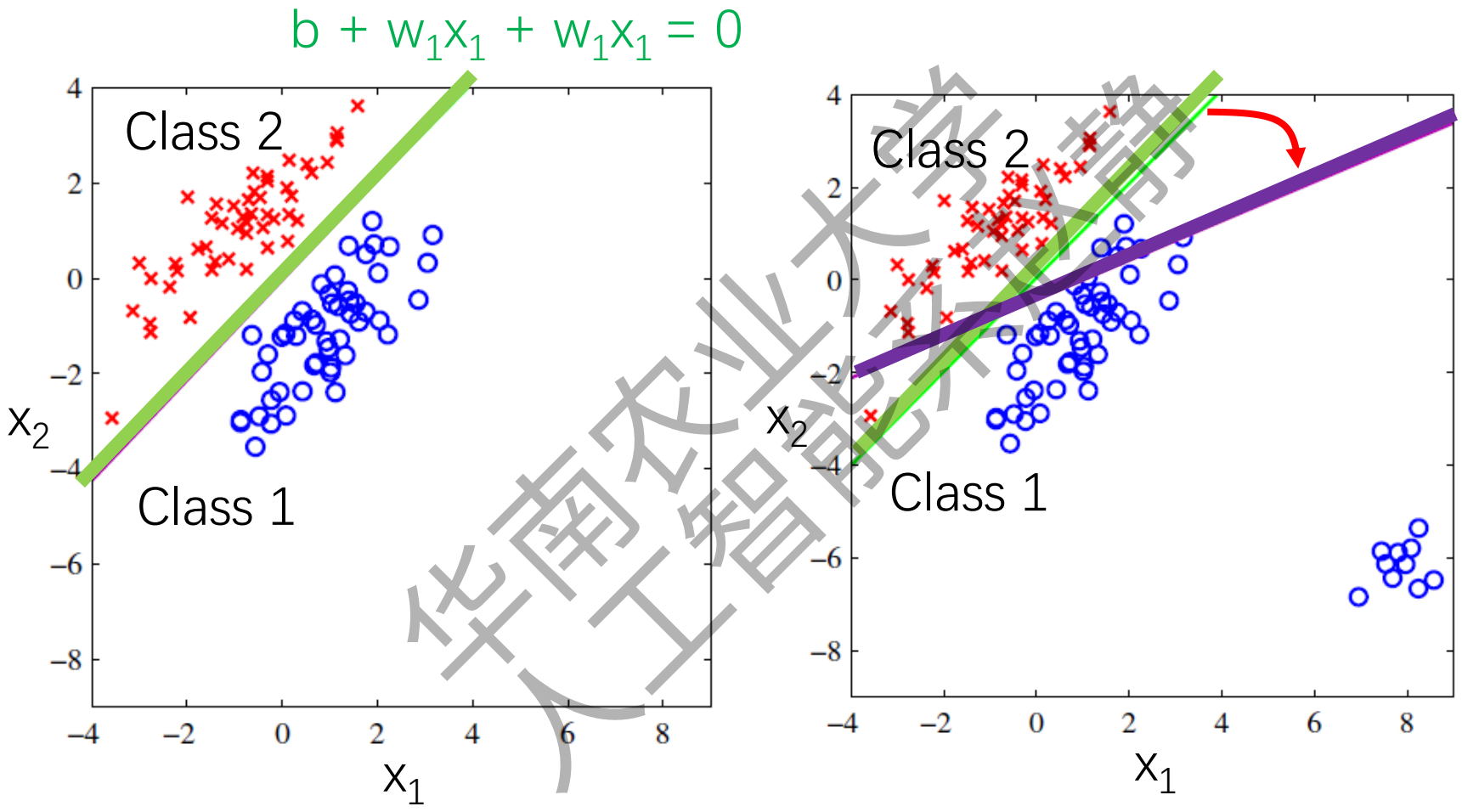
$$z = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$$
$$y \in \{0, 1\}$$

$$y = \begin{cases} 0, & z < 0; \\ 0.5, & z = 0; \\ 1, & z > 0, \end{cases}$$





阶跃函数缺点



‘soft’ binary classification:

$f(\mathbf{x}) = P(+1|\mathbf{x}) \in [0, 1]$

实际数据

$\mathbf{x}1, y1 =$ ○	$\sim P(y|\mathbf{x}1)$	$P(+1|\mathbf{x}1) = 0.9$
$\mathbf{x}2, y2 =$ ×	$\sim P(y|\mathbf{x}2)$	$P(+1|\mathbf{x}2) = 0.2$
.		.
.		.
.		.
$\mathbf{x}N, yN =$ ×	$\sim P(y|\mathbf{x}_N)$	$P(+1|\mathbf{x}N) = 0.6$

预测得心脏病的概率?

项目	值
年龄	40
性别	男
身高	180cm
体重	200kg
胆固醇	240
血糖	7.3

线性函数 $z = w^T x$ 分类预测结果需要得到[0,1]的概率值

设p为随机事件发生的概率，p的范围为[0,1]

事件概率比（**the odds of experiencing an event**）事件发生与事件不发生的概率之比 $\frac{p}{1-p}$

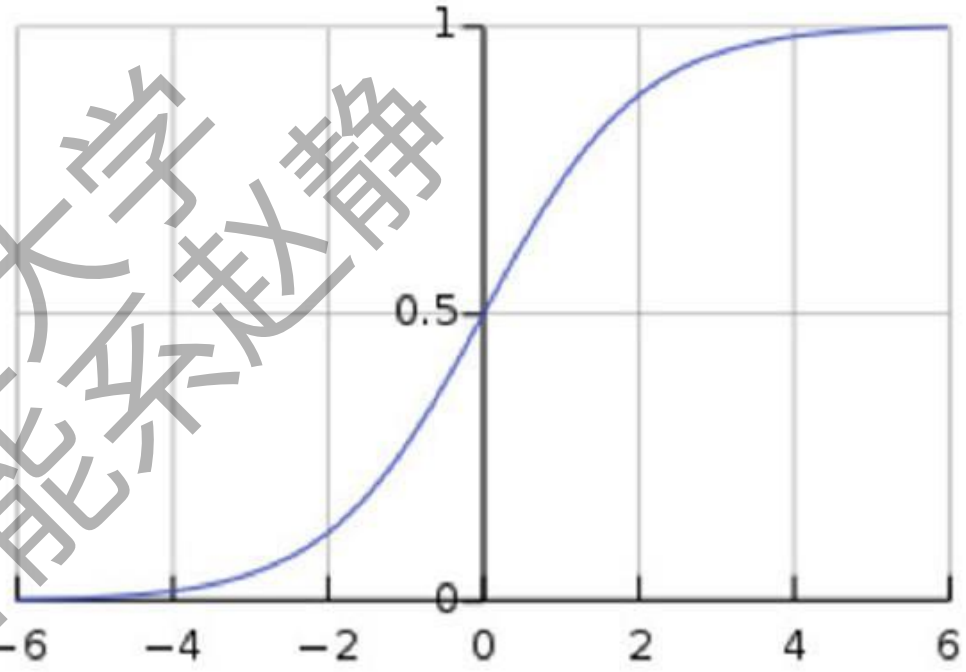
取对数得到：$\log\frac{p}{1-p}$，而$\log\frac{p}{1-p} = w^T x = z$

求解得到：$p = \frac{1}{1+e^{-w^T x}} = \frac{1}{1+e^{-z}}$

➢ Sigmoid 函数

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

$$z=w^T x + b$$



当$\sigma(z)$大于等于0.5时，预测 y=1

当$\sigma(z)$小于0.5时，预测 y=0

特点：

单调递增，$\sigma(-\infty)$，$\sigma(0)$，$\sigma(+\infty)$

平滑，处处可导

$1-\sigma(z)=\sigma(-z)$

优点：

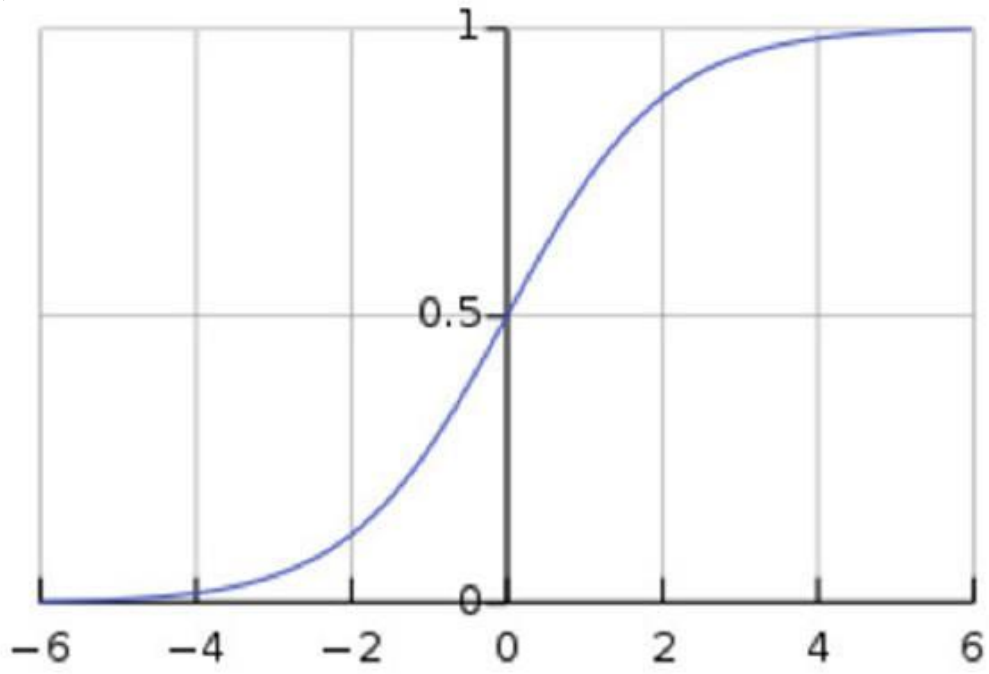
- 无需事先假设数据分布
- 可得到“类别”的近似概率预测
- 可直接应用现有数值优化算法求取最优解

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

def sigmoid(x):
    return 1/(1+np.exp(-x))

x = np.arange(-5., 5., 0.2)
y = sigmoid(x)

plt.plot(x, y)
```



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

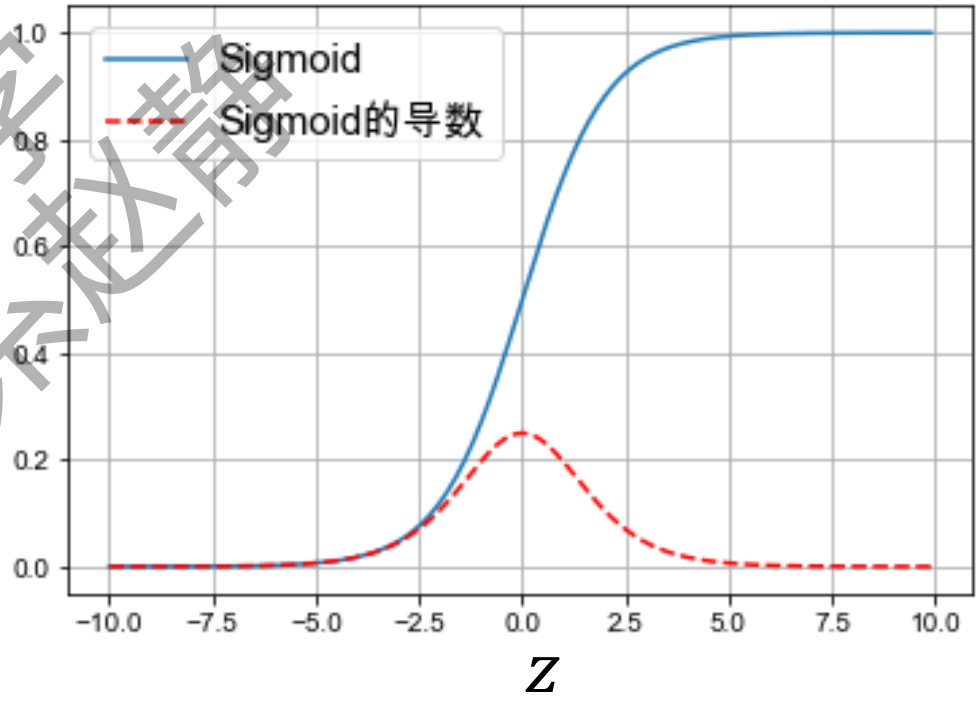
$$\sigma'(z) = \left(\frac{1}{1+e^{-z}}\right)'$$

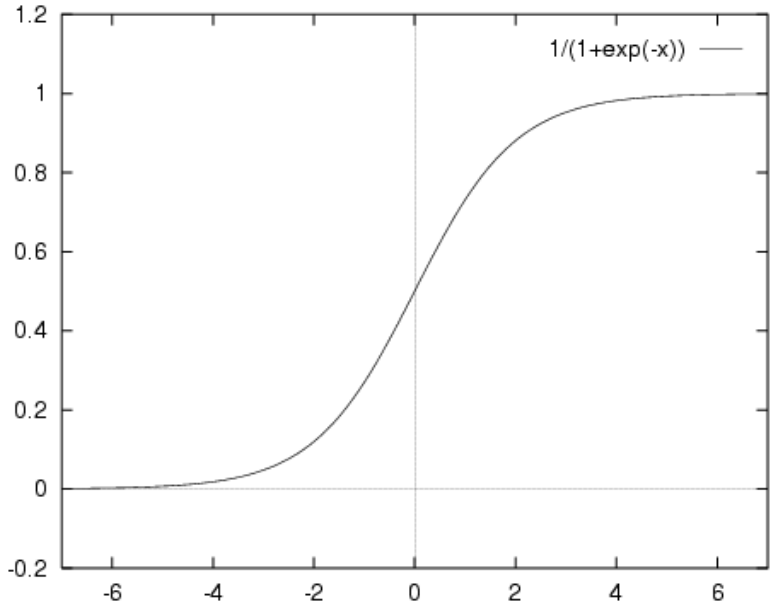
$$= \frac{e^{-z}}{(1+e^{-z})^2}$$

$$= \frac{1+e^{-z}-1}{(1+e^{-z})^2}$$

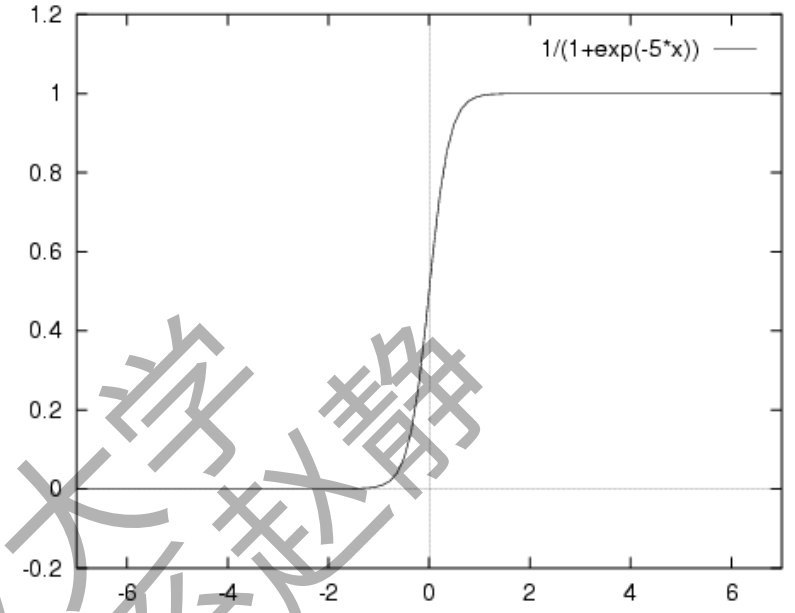
$$= \frac{1}{(1+e^{-z})}\left(1-\frac{1}{(1+e^{-z})}\right)$$

$$= \sigma(z)(1-\sigma(z))$$

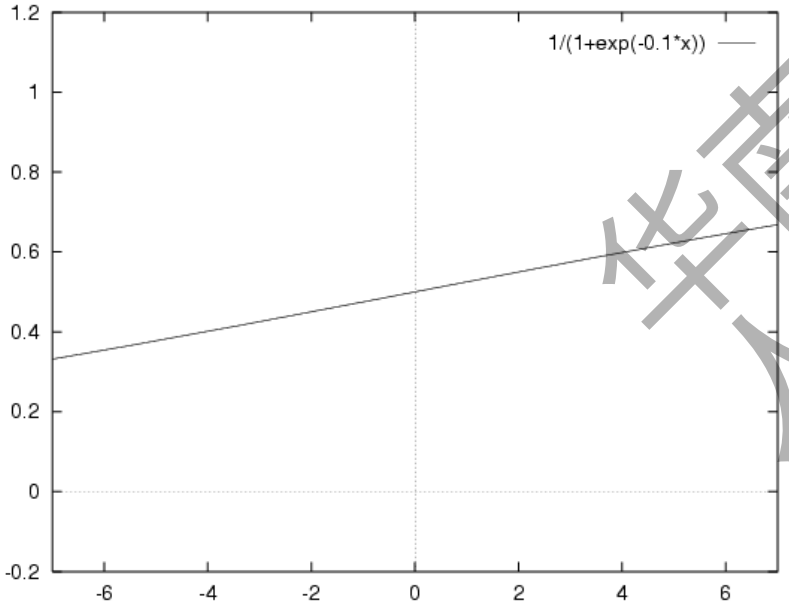




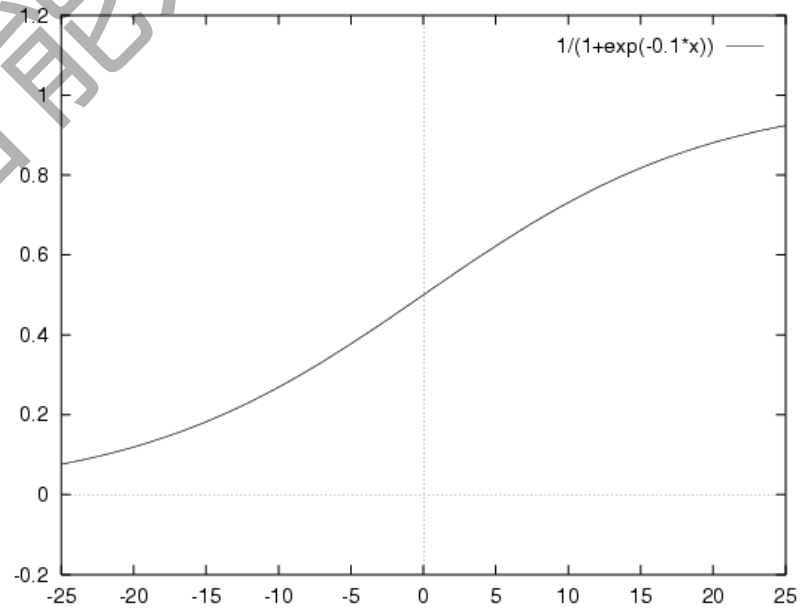
W=1



W=5



W=0.1



W=0.1

➢ 决策边界 (decision boundaries)

- 事件概率比(odds)：

$$\frac{p(y=1|\boldsymbol{x})}{p(y=0|\boldsymbol{x})}=\frac{\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x})}{1-\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x})}=\frac{1/\left(1+e^{-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}}\right)}{e^{-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}}/\left(1+e^{-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}}\right)}=e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}}$$

- 两边同取ln运算，得到**对数几率**：

$$\ln\frac{p(y=1|\boldsymbol{x})}{p(y=0|\boldsymbol{x})}=\ln\left(e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}}\right)=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$$

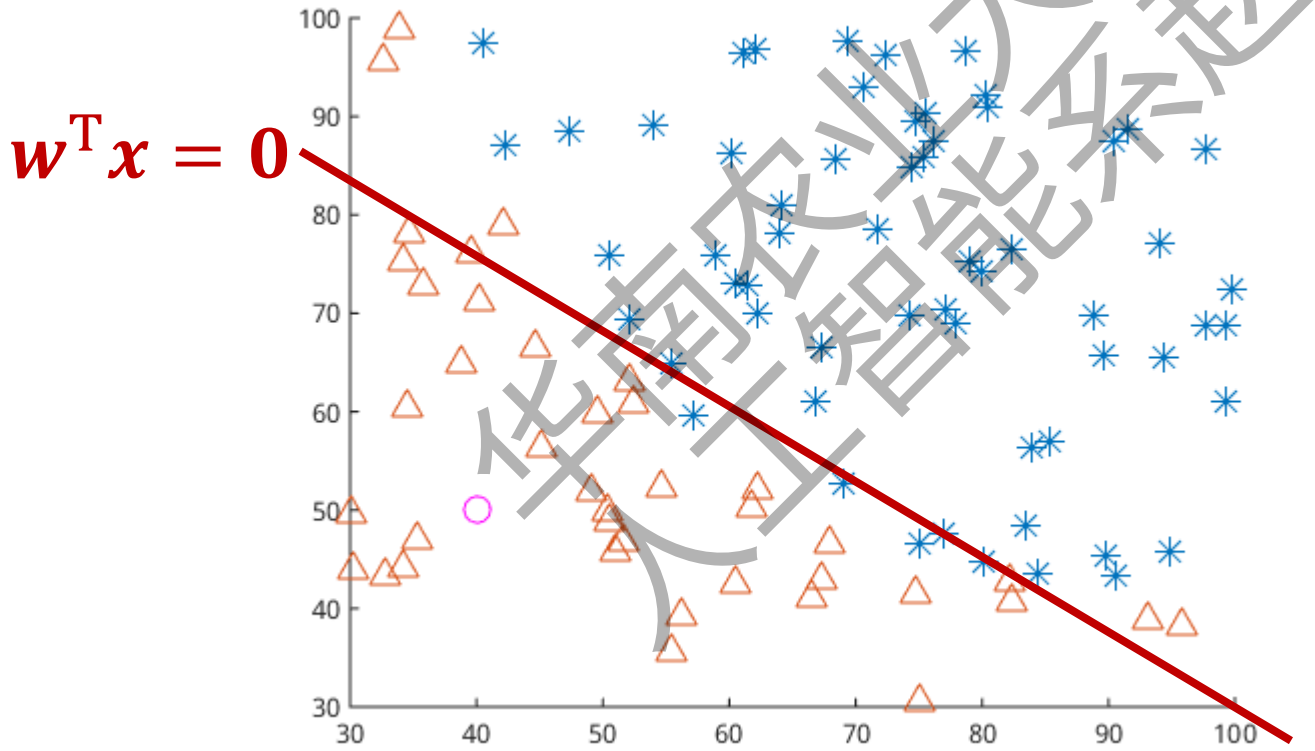
对数几率：$\ln \frac{p(y=1|\boldsymbol{x})}{p(y=0|\boldsymbol{x})} = \ln\left(e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}}\right) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$

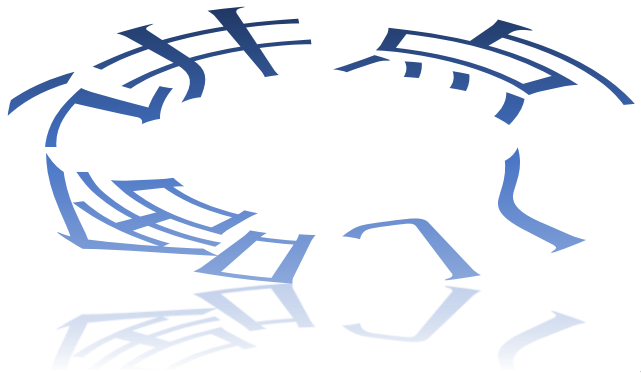
- 当$p(y=1|\boldsymbol{x}) > p(y=0|\boldsymbol{x})$时，$\mathbf{x}$的类别取$y=1$：

$$\frac{p(y=1|\mathbf{v})}{p(y=0|\boldsymbol{x})} > 1，\ \ln \frac{p(y=1|\boldsymbol{x})}{p(y=0|\boldsymbol{x})} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} > 0。$$

- 当$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} > 0$时，$\mathbf{x}$的类别取$y=1$；
- 当$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} < 0$时，$\mathbf{x}$的类别取$y=0$；
- 当$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} = 0$时，$y=1$的概率和$y=0$的概率相等，$x$位于决策面上。

决策函数$f(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$根据$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$的符号将输入空间$\mathcal{X}$分出两个区域。





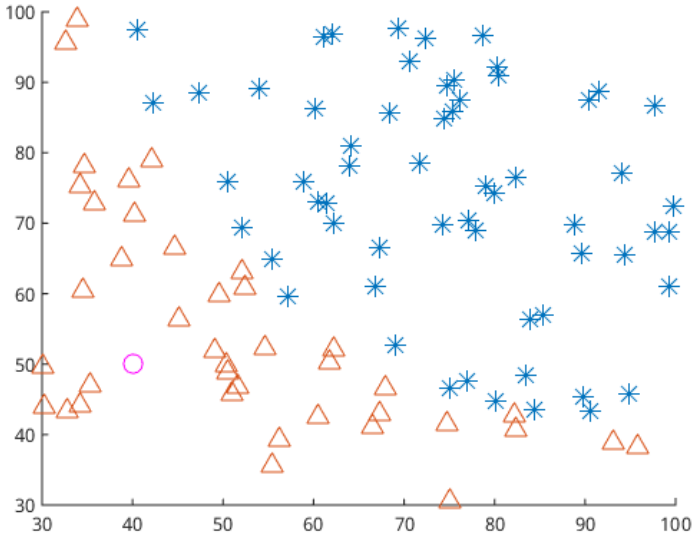
- Logistic回归模型
- Logistic回归的目标函数
- Logistic回归的优化算法
- Logistic回归的多类分类任务
- 分类任务中的样本不均衡问题
- 分类模型的评价指标

2. 目标函数

分类任务的损失函数

0/1 损失：预测类别正确损失为0，否则为1，记为

$$L(y,\hat{y}) = \begin{cases} 0 & y = \hat{y} \\ 1 & y \neq \hat{y} \end{cases}$$



最小二乘 $\min \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

➢ 交叉熵损失函数（Cross Entropy Loss）

$$\mu(\boldsymbol{x}) = \sigma\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}\right)$$

$$p(y = 1|x; w) = \mu(x)$$

$$p(y = 0|x; w) = 1 - \mu(x)$$

- 似然函数为：

$$p(y|\boldsymbol{x}; \mu(\boldsymbol{x})) = \mu(\boldsymbol{x})^{y}(1 - \mu(\boldsymbol{x}))^{(1-y)}$$

- log似然函数为：

$$\ell(\mu) = \ln p(\mathcal{D}) = \sum_{i=1}^{N} \ln p(y_i|\boldsymbol{x}_i) = \sum_{i=1}^{N} \ln\left(\mu(\boldsymbol{x}_i)^{y_i}(1 - \mu(\boldsymbol{x}_i))^{(1-y_i)}\right)$$

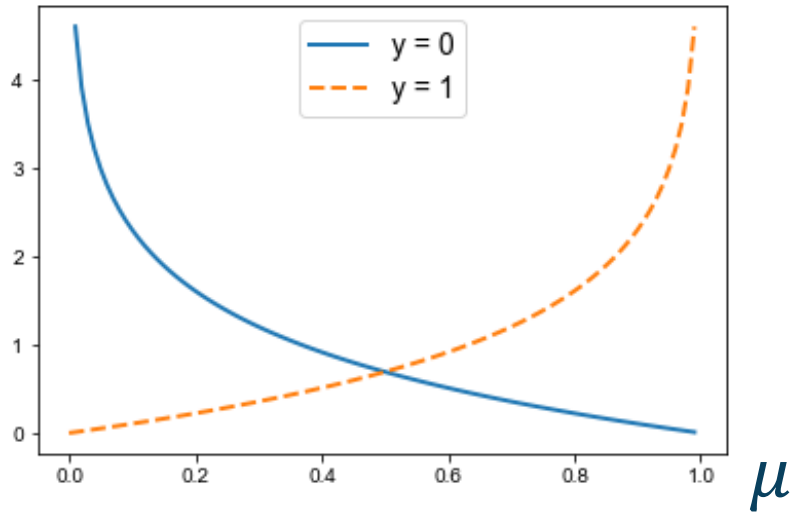
- 负log似然损失为：

$$L\big(y,\mu(\boldsymbol{x})\big)=-y\ln\big(\mu(\boldsymbol{x})\big)-(1-y)\ln\big(1-\mu(\boldsymbol{x})\big)$$

- 极大似然估计等价于训练集上的最小负log似然损失，即交叉熵损失：

$$J(\mu)=\sum_{i=1}^{N}L\big(y_i,\mu(\boldsymbol{x}_i)\big)$$

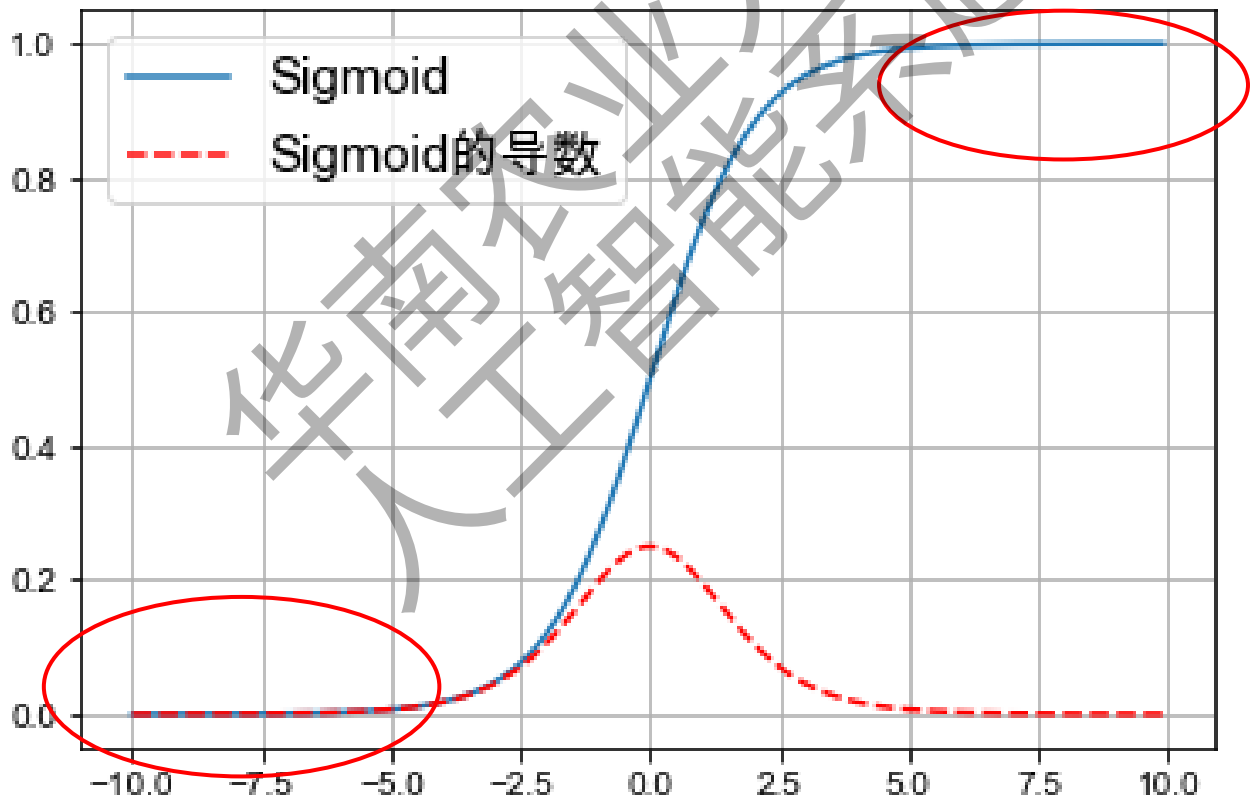
$$=\sum_{i=1}^{N}-y_i\ln\big(\mu(\boldsymbol{x}_i)\big)-(1-y_i)\ln\big(1-\mu(\boldsymbol{x}_i)\big)$$

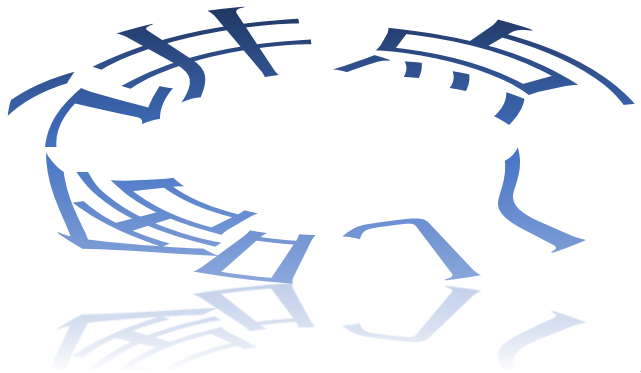


➢ Logistic回归中的正则项

- 线性回归：目标函数可以不加正则项（OLS）
- Logistic回归：必须加正则

$$J(\boldsymbol{w};C)=C\sum_{i=1}^{N}L\left(y_i,\mu(\boldsymbol{x}_i;\boldsymbol{w})\right)+R(\boldsymbol{w})$$





- Logistic回归模型
- Logistic回归的目标函数
- Logistic回归的优化算法
- Logistic回归的多类分类任务
- 分类任务中的样本不均衡问题
- 分类模型的评价指标
- Sklearn中的Logistic回归API
- Logistic回归案例分析

3. 优化求解——梯度下降法

$$\mu(\boldsymbol{x},\boldsymbol{w})=\sigma\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}\right)$$
$$\sigma(z)=\frac{1}{1+e^{-z}}$$
$$\frac{d\sigma(z)}{dz}=\sigma(z)(1-\sigma(z))$$
$$\frac{\partial\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}\right)}{\partial\boldsymbol{w}}=\boldsymbol{x}$$

- 损失函数之和求解

$$J\ \ (\boldsymbol{w})=\sum_{i=1}^{N}\left(-y_i\ln\left(\mu(\boldsymbol{x}_i,\boldsymbol{w})\right)-(1-y_i)\ln\left(1-\mu(\boldsymbol{x}_i,\boldsymbol{w})\right)\right)$$

$$g\ \ (\boldsymbol{w})=\frac{\partial J\ \ (\boldsymbol{w})}{\partial\boldsymbol{w}}=-\sum_{i=1}^{N}\left(y_i\frac{1}{\mu(\boldsymbol{x}_i,\boldsymbol{w})}-(1-y_i)\frac{1}{1-\mu(\boldsymbol{x}_i,\boldsymbol{w})}\right)\frac{\partial\mu(\boldsymbol{x}_i,\boldsymbol{w})}{\partial\boldsymbol{w}}$$

$$=-\sum_{i=1}^{N}\left(y_i\frac{1}{\mu(\boldsymbol{x}_i,\boldsymbol{w})}-(1-y_i)\frac{1}{1-\mu(\boldsymbol{x}_i,\boldsymbol{w})}\right)\mu(\boldsymbol{x}_i,\boldsymbol{w})\left(1-\mu(\boldsymbol{x}_i,\boldsymbol{w})\right)\frac{\partial\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i\right)}{\partial\boldsymbol{w}}$$

$$=-\sum_{i=1}^{N}\left(y_i\left(1-\mu(\boldsymbol{x}_i,\boldsymbol{w})\right)-(1-y_i)\mu(\boldsymbol{x}_i,\boldsymbol{w})\right)\boldsymbol{x}_i\ \ =\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{\mu}-\boldsymbol{y})$$

形式同线性回归模型

- 损失函数之和的梯度为：$g(\boldsymbol{w}) = \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{\mu} - \boldsymbol{y})$

- 梯度下降法参数更新公式为：

$$w_j := w_j - \eta g(\boldsymbol{w})$$

➢ Hessian矩阵

- 损失函数和的梯度为：$g(\boldsymbol{w}) = \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{\mu} - \boldsymbol{y})$
- 则Hessian矩阵为：

$$\boldsymbol{H}(\boldsymbol{w}) = \frac{\partial g(\boldsymbol{w})}{\partial \boldsymbol{w}^{\mathrm{T}}} = \frac{\partial\left(\sum_{i=1}^{N}(\mu(\boldsymbol{x}_i, \boldsymbol{w}) - y_i)\boldsymbol{x}_i\right)}{\partial \boldsymbol{w}^{\mathrm{T}}}$$

$$= \sum_{i=1}^{N} \mu_i(1-\mu_i)\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}$$

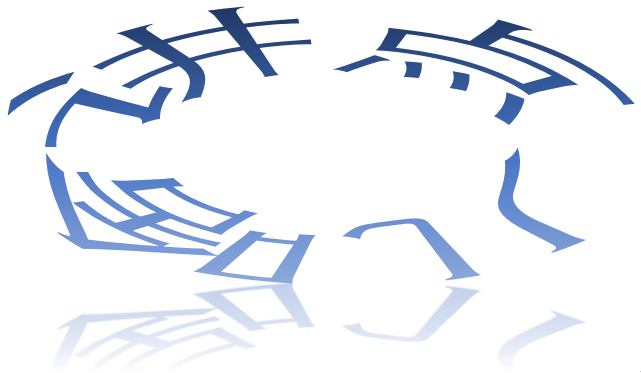
$$= \boldsymbol{X}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{X}$$

$$\frac{\partial(\boldsymbol{a}^T\boldsymbol{y})}{\partial \boldsymbol{y}} = \boldsymbol{a}^{\mathrm{T}}$$

$$\frac{\partial \mu(\boldsymbol{x}_i; \boldsymbol{w})}{\partial \boldsymbol{w}^{\mathrm{T}}} = \boldsymbol{x}_i^{\mathrm{T}}\mu_i(1-\mu_i)$$

$$\boldsymbol{S} \triangleq diag(\mu_i(1-\mu_i))$$

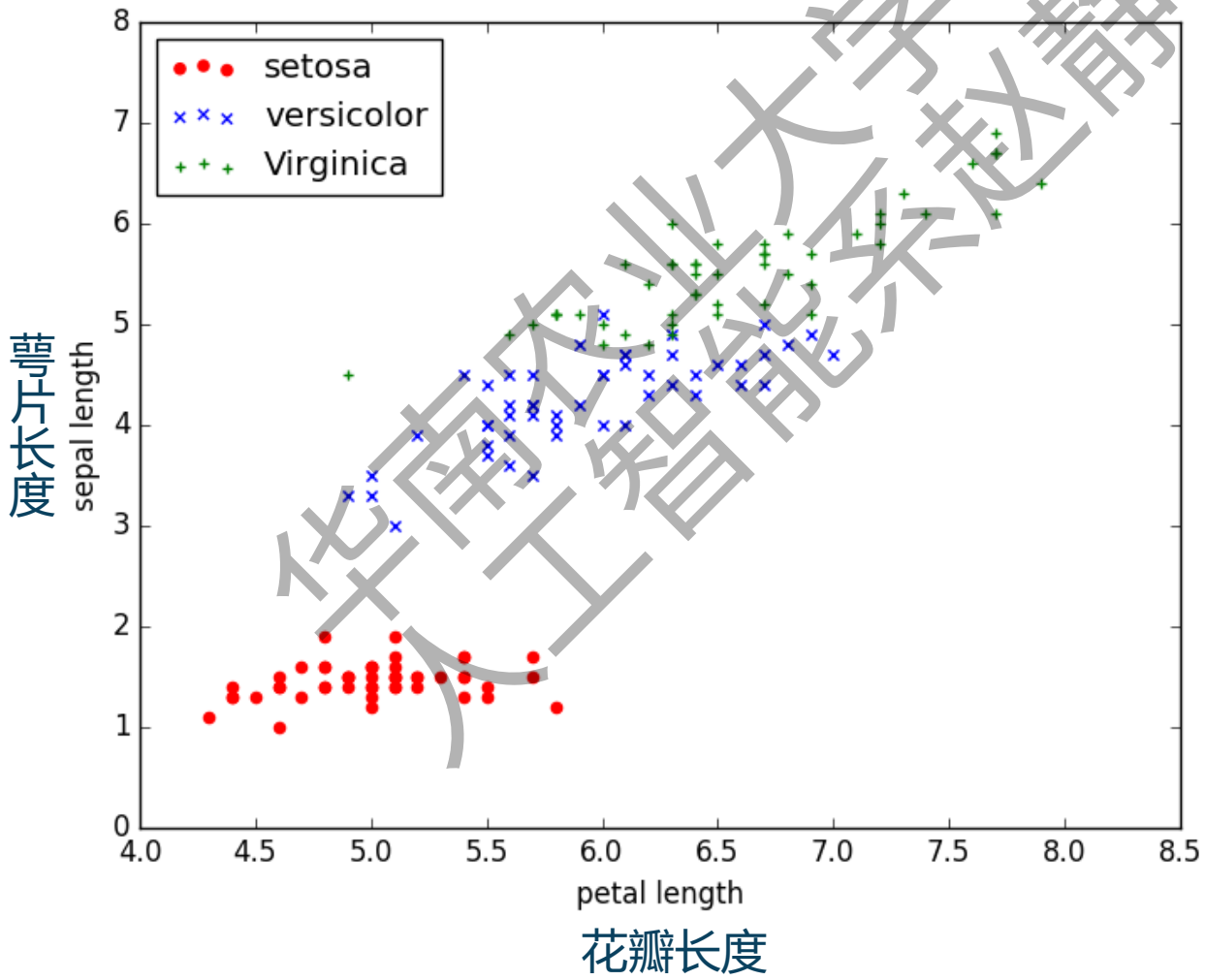
- $\boldsymbol{H}(\boldsymbol{w})$为正定矩阵，梯度下降能找到全局最小值。



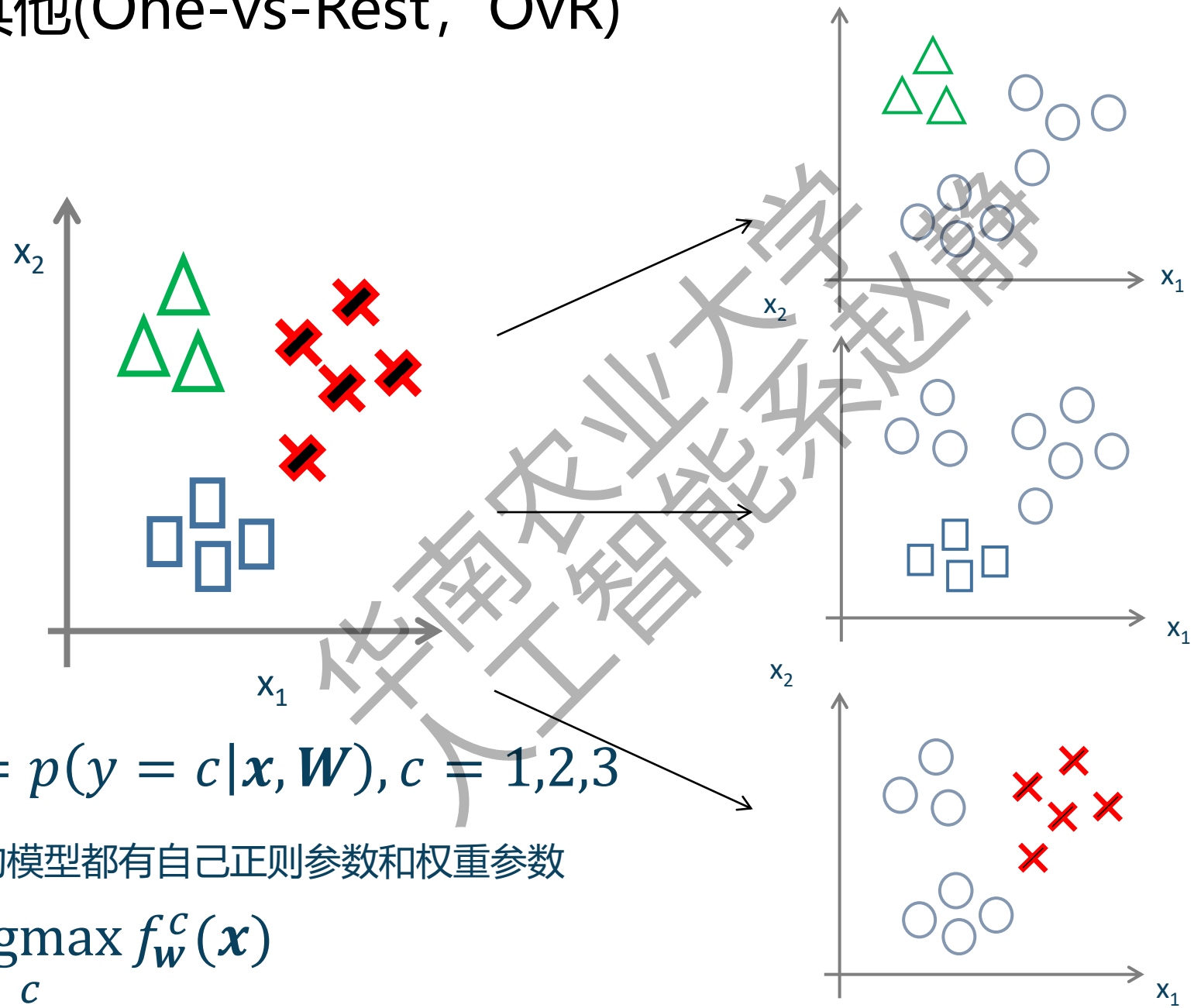
- Logistic回归模型
- Logistic回归的目标函数
- Logistic回归的优化算法
- Logistic回归的多类分类任务
- 分类模型的评价指标
- Sklearn中的Logistic回归API
- Logistic回归案例分析

4. 多类分类任务

例：Iris花分类中，一共有3种鸢尾花



➢ 一对其他(One-vs-Rest，OvR)



$$f_{\boldsymbol{w}}^{c}(\boldsymbol{x}) = p(y = c|\boldsymbol{x}, \boldsymbol{W}), c = 1,2,3$$

每类的模型都有自己正则参数和权重参数

$$\hat{y} = \underset{c}{\operatorname{argmax}}\, f_{\boldsymbol{w}}^{c}(\boldsymbol{x})$$

➢ 多项分布直接实现多类分类

✓ Multinoulli分布

- 将类别y用独热编码，记为向量$\boldsymbol{y}$
- Multinoulli分布的概率函数为：$Cat(\boldsymbol{y};\boldsymbol{\theta}) = \prod_{c=1}^{C} \theta_c^{y_c}$，

$$或：Cat(y;\boldsymbol{\theta}) = \prod_{c=1}^{C} \theta_c^{\mathrm{II}(y=c)}$$

其中y_c表示向量$\boldsymbol{y}$的第c个元素，

其中$\mathrm{II}(.)$为指示函数，当括号中条件满足时函数值为1，否则为0。

➢ 多类分类——Softmax分类器

输出$y = c$的概率可以由$\mathbf{x}$的线性组合再经过sigmoid函数变换得到

$$p(y = c|\boldsymbol{x}, \boldsymbol{W}) = \frac{\exp(\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x})}{\sum_{c'=1}^{C}\exp(\boldsymbol{w}_{c'}^{\mathrm{T}}\boldsymbol{x})}$$

Softmax函数

$$\sigma(z_c) = \frac{e^{z_c}}{\sum_{c'=1}^{C} e^{z_{c'}}}$$

- 将类别y用独热编码为向量$\boldsymbol{y}$：$y_c = I(y = c)$
- 向量$\boldsymbol{\mu}$表示Multinoulli分布的参数：

$$\mu_c = p(y = c|\boldsymbol{x}, \boldsymbol{W}) = \frac{\exp(\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x})}{\sum_{c'=1}^{C}\exp(\boldsymbol{w}_{c'}^{\mathrm{T}}\boldsymbol{x})}$$

- Softmax分类模型的log似然函数为：

$$\ell(\boldsymbol{W}) = \sum_{i=1}^{N}\ln\left(\prod_{c=1}^{C}\mu_{i,c}^{y_{i,c}}\right) = \sum_{i=1}^{N}\sum_{c=1}^{C}y_{i,c}\ln(\mu_{i,c})$$

- 定义Softmax损失为负log似然：

$$L(\boldsymbol{y}, \boldsymbol{\mu}) = -\sum_{c=1}^{C} y_c \ln(\mu_c)$$

- 极大似然估计等价于最小训练集上的Softmax损失/负log似然损失：

$$J(\mathbf{W}) = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \ln(\mu_{i,c})$$

$$= -\sum_{i=1}^{N}\left(\sum_{c=1}^{C} y_{i,c}\left(\mathbf{w}_c^T\mathbf{x} - ln\left(\sum_{c'=1}^{C}\exp\left(\mathbf{w}_{c'}^T\mathbf{x}\right)\right)\right)\right)$$

$$\mu_c = p(y = c|\boldsymbol{x}, \boldsymbol{W}) = \frac{\exp\left(\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}\right)}{\sum_{c'=1}^{C}\exp\left(\boldsymbol{w}_{c'}^{\mathrm{T}}\boldsymbol{x}\right)}$$

➢ 分类任务中的样本不均衡问题——数据解决方案

✓ EasyEnsemble算法

- 对于多数类样本，通过n次有放回抽样生成n份子集
- 少数类样本分别和这n份样本合并训练一个模型：n个模型
- 最终模型：n个模型预测结果的平均值

✓ BalanceCascade（级联）算法

- 从多数类中有效地选择一些样本与少数类样本合并为新的数据集进行训练
- 训练好的模型每个多数类样本进行预测。若预测正确，则不考虑将其作为下一轮的训练样本
- 依次迭代直到满足某一停止条件，最终的模型是多次迭代模型的组合

✓ SMOTE（Synthetic Minority Over-sampling Technique）：

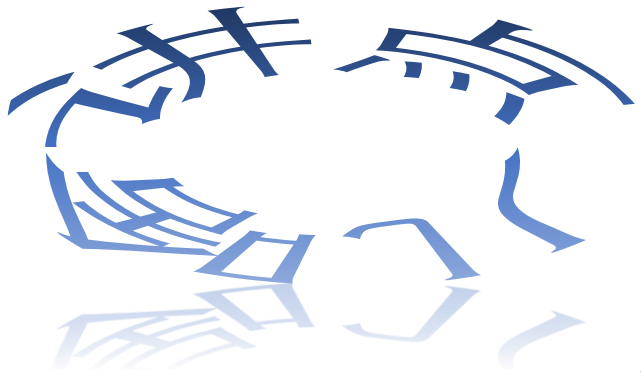
基于“插值”为少数类合成新的样本

➢ 分类任务中的样本不均衡问题——算法解决方案

✓ 代价敏感学习算法(Cost-Sensitive Learning)

代价矩阵（混淆矩阵）

Predict \ True	0	1
0	C_{00}	C_{01}
1	C_{10}	C_{11}



- Logistic回归模型
- Logistic回归的目标函数
- Logistic回归的优化算法
- Logistic回归的多类分类任务
- 分类模型的评价指标
- Sklearn中的Logistic回归API

5. 分类模型的评价指标

1. 正确率：Sklearn中分类任务的默认评价指标

$$\text{Accuracy}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N}\sum_{i=0}^{\mathrm{N}} \mathrm{II}\left(\hat{y}_i = y_i\right)$$

其中 $\mathrm{II}(.)$ 为指示函数，当括号中条件满足时函数值为1，否则为0

2. 负log似然损失（交叉熵损失）

$$\text{logloss}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \ln\left(\hat{y}_{i,c}\right)$$

3. 合页损失（SVM中的损失函数）

$$\mathrm{HingeLoss}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N}\sum_{i=0}^{\mathrm{N}} \max(1, \hat{y}_i y_i)$$

4. 混淆矩阵：

例：手写数字识别的混淆矩阵

真值

```
[[87  0  0  0  1  0  0  0  0  0]
 [ 0 88  1  0  0  0  0  0  1  1]
 [ 0  0 85  1  0  0  0  0  0  0]
 [ 0  0  0 79  0  3  0  4  5  0]
 [ 0  0  0  0 88  0  0  0  0  4]
 [ 0  0  0  0  0 88  1  0  0  2]
 [ 0  1  0  0  0  0 90  0  0  0]
 [ 0  0  0  0  0  1  0 88  0  0]
 [ 0  0  0  0  0  0  0  0 88  0]
 [ 0  0  0  1  0  1  0  0  0 90]]
```

预测值

混淆矩阵

真值 \ 预测值	$\hat{y}=1$	$\hat{y}=0$	Σ
$y=1$	TP	FN	N_+
$y=0$	FP	TN	N_-
Σ	$\hat{N}_+$	$\hat{N}_-$	

$$\text{Precison} = \frac{TP}{\hat{N}_+}$$ 精度：检测结果真正为正的比例（质）

$$\text{Recall} = \frac{TP}{N_+}$$ 召回率: 被正确检测到的正样本的比例（量），TPR（True Positive Rate）

$$\text{FPR} = \frac{TP}{N_-}$$ 假阳率: 负样本被预测成了正样本的比例，FPR（False Positive Rate）

$$F1 = \frac{2(\text{Precison} \times \text{Recall})}{\text{Precison} + \text{Recall}}$$ F1分数： Precision 和Recall调和平均值

注意：准确率和召回率是互相影响的。一般情况下准确率高、召回率就低；召回率低、准确率高。

混淆矩阵

真值 \ 预测值	$\hat{y}=1$	$\hat{y}=0$	Σ
$y=1$	TP	FN	N_+
$y=0$	FP	TN	N_-
Σ	$\hat{N}_+$	$\hat{N}_-$	

PR曲线

精度、准确率：预测结果为真的样本中真正为真的比例

$$\text{Precison} = \frac{TP}{\hat{N}_+}$$

召回率：预测结果召回了多少真正的真样本；
真阳率：有多少真正的正样本被预测为真

$$\text{TPR} = \text{Recall} = \frac{TP}{N_+}$$

假阳率：预测结果将多少假的样本预测预测成了真

$$\text{FPR} = \frac{FP}{N_-}$$

ROC曲线

- Matthews相关性系数用一个值综合混淆矩阵，度量真实值和预测值之间的相关性，定义为

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

注：分母中任意一对括号相加之和如果为0，那么整个MCC的值就为0。MCC值在[-1,1]之间

1：完美分类器

0：随机分类器

-1: 分类器是最差，所有预测结果和实际相反

		预测值		
		$\hat{y}=1$	$\hat{y}=0$	Σ
真值	$y=1$	TP	FN	N_+
	$y=0$	FP	TN	N_-
	Σ	$\hat{N}_+$	$\hat{N}_-$	

例：垃圾邮件分类

- 假如我们有100封测试邮件，其中50封为垃圾50封为非垃圾，某分类器的预测结果：TP=40封，TN=40封，FP = 10封，FN = 10封

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} = 0.8$$

$$\text{Precison} = \frac{TP}{TP + FP} = 0.8,\ \text{Recall} = \frac{TP}{TP + FN} = 0.8$$

$$F1 = \frac{2(\text{Precison} \times \text{Recall})}{(\text{Precison}+\text{Recall})} = 0.8$$

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} = 0.6$$

- 还是100封测试邮件，其中垃圾邮件为98封，非垃圾邮件为2封（分布不均衡）。

假设如果有一个傻傻的分类器，永远只做出垃圾邮件判断，那么TP=98，TN=0，FN=0，FP=2

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} = 0.98$$

$$\text{Precison} = \frac{TP}{TP + FP} = 0.98,\ \text{Recall} = \frac{TP}{TP + FN} = 1$$

$$F1 = \frac{2(\text{Precison} \times \text{Recall})}{(\text{Precison}+\text{Recall})} = 0.989$$

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} = 0$$

随机分类器

逻辑回归代码实现

Sigmoid 函数 $\sigma(z) = \frac{1}{1+e^{-z}}$

```
def sigmoid(z):
    return 1 / (1 + np.exp(-z))
```

损失函数 $J(\boldsymbol{w}) = \sum_{i=1}^{N}\left(-y_i \ln\left(\mu(\boldsymbol{x}_i, \boldsymbol{w})\right) - (1-y_i)\ln\left(1-\mu(\boldsymbol{x}_i, \boldsymbol{w})\right)\right)$

```
def cost(w, X, y):
    w = np.matrix(w)
    X = np.matrix(X)
    y = np.matrix(y)
    first = np.multiply(-y, np.log(sigmoid(X * w.T)))
    second = np.multiply((1 - y), np.log(1 - sigmoid(X * w.T)))
    return np.sum(first - second) / (len(X))
```

L2正则化

$$J(\boldsymbol{w}) = C\sum_{i=1}^{N}\left(-y_i \ln\left(\mu(\boldsymbol{x}_i, \boldsymbol{w})\right) - (1-y_i)\ln\left(1-\mu(\boldsymbol{x}_i, \boldsymbol{w})\right)\right) + \sum_{i=1}^{N} w_i^2$$

Fun Time

在逻辑回归算法中，设w_0=0，η=0.1，w_1=？

已知：$w_j := w_j - \eta \frac{1}{m} \sum_{i=1}^{m} (\boldsymbol{\mu}(x^{(i)}) - y^{(i)}) x_j^{(i)}$

① w_1=0.1 $\frac{1}{m} \sum_{i=1}^{m} y^{(i)} x_j^{(i)}$

② w_1=-0.1 $\frac{1}{m} \sum_{i=1}^{m} y^{(i)} x_j^{(i)}$

③ w_1=0.05 $\frac{1}{m} \sum_{i=1}^{m} y^{(i)} x_j^{(i)}$

④ w_1=-0.05 $\frac{1}{m} \sum_{i=1}^{m} y^{(i)} x_j^{(i)}$

- 函数集合 $\sigma(z)=\frac{1}{1+e^{-z}}$
- 目标函数
  - 损失函数：交叉熵损失
  - 正则项：L2正则、L1正则

$$J(\boldsymbol{w})=C\sum_{i=1}^{N}\left(-y_i\ln\left(\mu(\boldsymbol{x}_i,\boldsymbol{w})\right)-(1-y_i)\ln\left(1-\mu(\boldsymbol{x}_i,\boldsymbol{w})\right)\right)+\sum_{i=1}^{N}w_i^2$$

- 优化方法：梯度下降

$$g(\boldsymbol{w})=\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{\mu}-\boldsymbol{y}) \qquad w_j:=w_j-\eta g(\boldsymbol{w})$$

- 分类模型评价指标
- 类别不均衡的分类