2.1 Logistic回归

CSDN学院

大纲

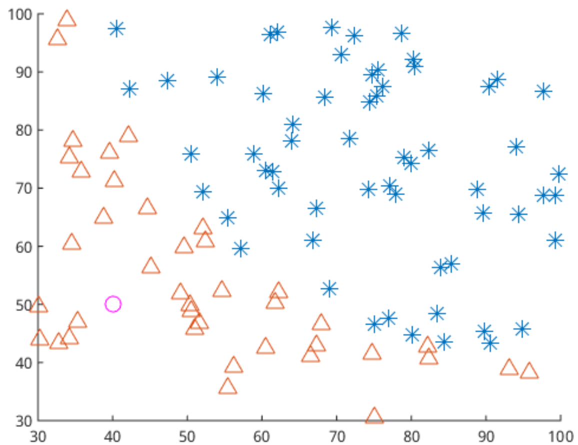
- Logistic回归基本原理
- 多类Logistic回归
- Scikit learn 中的Logistic回归实现
- 分类模型的评价
- 模型选择与参数调优
- 案例分析

▶分类

- 给定训练数据 $\mathcal{D}=\{\mathbf{x}_i, y_i\}_{i=1}^N$，分类任务学习一个从输入$\mathbf{x}$到输出$y$的映射$f$：

$$\hat{y}=f(\mathbf{x})=\arg\max_c p(y=c \mid \mathbf{x}, \mathcal{D})$$

- 其中y为离散值，其取值范围称为标签空间：$\mathcal{Y}=\{1,2,\ldots,C\}$



▶分类

- 分类： $\hat{y} = f(\mathbf{x}) = \arg\max_{c} p(y = c \mid \mathbf{x}, \mathcal{D})$
- 当C=2时，为两类分类问题，计算出$p(y=1 \mid \mathbf{x})$即可。此时分布为Bernoulli分布:

$$p(y \mid \mathbf{x}) = \mathrm{Ber}(y \mid \mu(\mathbf{x}))$$

  – 其中 $\mu(\mathbf{x}) = \mathbb{E}(y \mid \mathbf{x}) = p(y = 1 \mid \mathbf{x})$

► Recall: Bernoulli分布

- Bernoulli分布又名两点分布或者0-1分布。若Bernoulli试验成功，则Bernoulli随机变量X取值为1，否则X为0。记试验成功概率为θ，我们称X服从参数为θ的Bernoulli分布，记为: $X \sim Ber(\theta)$, 概率函数（pmf）为：

$$p(x) = \theta^x (1-\theta)^{1-x} = \begin{cases} \theta & \text{if } x = 1 \\ 1-\theta & \text{if } x = 0 \end{cases}$$

Jakob I. Bernoulli
(1654—1705)

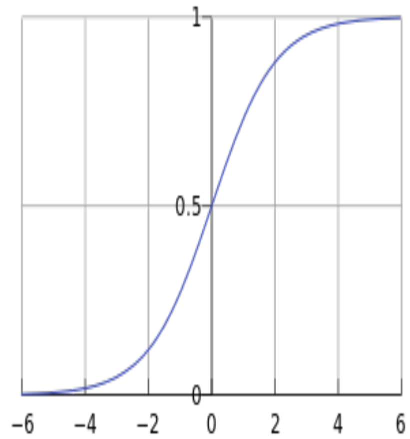
- Bernoulli分布的均值：$\mu = \theta$
- 方差：$\sigma^2 = \theta \times (1-\theta)$

▶ Logistic回归模型

- Logistic回归模型同线性回归模型类似，也是一个线性模型，只是条件概率$p(y|\mathbf{x})$的形式不同：

$$p(y|\mathbf{x}) = \mathrm{Ber}(y|\mu(\mathbf{x})),$$

$$\mu(\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x})$$



- 其中sigmoid函数（S形函数）定义为

$$\sigma(a) = \frac{1}{1+\exp(-a)} = \frac{\exp(a)}{\exp(a)+1}$$

- 亦被称为logistic函数或logit函数，将实数a变换到[0,1]区间。
  - 因为概率取值在[0,1]区间
  - Logistic回归亦被称为logit回归

▶ 为什么用logistic函数？

- 在神经科学中，
  - 神经元对其输入进行加权和：$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$
  - 如果该和大于某阈值 $f(\mathbf{x}) > \tau$，神经元发放脉冲
- 在Logistic回归，定义Log Odds Ratio:

$$\mathrm{LOR}(\mathbf{x}) = \log \frac{p(y=1 \mid \mathbf{x}, \mathbf{w})}{p(y=0 \mid \mathbf{x}, \mathbf{w})} = \log\left[\frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})} \frac{1+\exp(-\mathbf{w}^T\mathbf{x})}{\exp(-\mathbf{w}^T\mathbf{x})}\right]$$

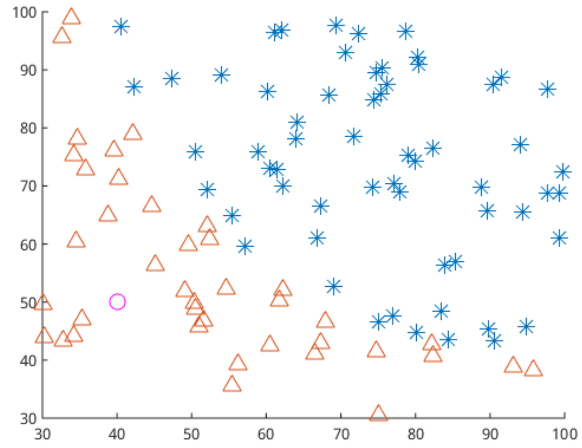
$$= \log\left[\exp(\mathbf{w}^T\mathbf{x})\right] = \mathbf{w}^T\mathbf{x}$$

- 因此*iff* $\mathrm{LOR}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} > 0$，神经元发放脉冲，即

$$p(y=1 \mid \mathbf{x}, \mathbf{w}) > p(y=0 \mid \mathbf{x}, \mathbf{w})$$

▶线性决策函数

- 在Logistic回归中
  - $\mathrm{LOR}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} > 0$，$\hat{y} = 1$
  - $\mathrm{LOR}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} < 0$, $\hat{y} = 0$
  - $\mathbf{w}^T\mathbf{x} = 0$: 决策面



- 因此 $a(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ 分类决策面
  - 因此Logistic回归是一个线性分类器

▶极大似然估计

$$\mathrm{Ber}(x\mid\theta)=\theta^{x}(1-\theta)^{1-x}$$

- Logistic回归： $p(y\mid\mathbf{x},\mathbf{w})=\mathrm{Ber}(y\mid\mu(\mathbf{x}))$， $\mu(\mathbf{x})=\sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x})$

- 令$\mu_i=\mu(\mathbf{x}_i)$，则负log似然为

$$J(\mathbf{w})=NLL(\mathbf{w})=-\sum_{i=1}^{N}\log\left[(\mu_i)^{y_i}\times(1-\mu_i)^{(1-y_i)}\right]$$

$$=\sum_{i=1}^{N}-\Big[\underbrace{y_i\log(\mu_i)+(1-y_i)\log(1-\mu_i)}_{\text{Logistic损失}}\Big]$$

极大似然估计 等价于 最小Logistic损失

优化求解：梯度下降 / 牛顿法

▶ 梯度

- 目标函数为

$$J(\mathbf{w})=\sum_{i=1}^{N}-\left[y_i\log(\mu_i)+(1-y_i)\log(1-\mu_i)\right]$$

- 梯度为

$$g(\mathbf{w})=\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}=\frac{\partial}{\partial \mathbf{w}}\left[\sum_{i=1}^{N}-\left[y_i\log(\mu_i)+(1-y_i)\log(1-\mu_i)\right]\right]$$

$$J(\mathbf{w}) = -\sum_{i=1}^{N}\left[y_i \log(\mu_i) + (1-y_i)\log(1-\mu_i)\right]$$

$$\begin{aligned}
g(\mathbf{w}) = \frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) &= \sum_{i=1}^{N}\left[-y_i \times \frac{1}{\mu(\mathbf{x}_i)}\frac{\partial}{\partial \mathbf{w}}\mu(\mathbf{x}_i) + (1-y_i)\times\frac{1}{1-\mu(\mathbf{x}_i)}\frac{\partial}{\partial \mathbf{w}}\mu(\mathbf{x}_i)\right] \\
&= \sum_{i=1}^{N}\left[-y_i \times \frac{1}{\mu(\mathbf{x}_i)} + (1-y_i)\times\frac{1}{1-\mu(\mathbf{x}_i)}\right]\frac{\partial}{\partial \mathbf{w}}\mu(\mathbf{x}_i) \\
&= \sum_{i=1}^{N}\left[-y_i \times \frac{1}{\mu(\mathbf{x}_i)} + (1-y_i)\times\frac{1}{1-\mu(\mathbf{x}_i)}\right]\mu(\mathbf{x}_i)\left(1-\mu(\mathbf{x}_i)\right)\mathbf{x}_i \\
&= \sum_{i=1}^{N}\left[-y_i \times \left[1-\mu(\mathbf{x}_i)\right] + (1-y_i)\mu(\mathbf{x}_i)\right]\mathbf{x}_i \\
&= \sum_{i=1}^{N}\left[-y_i + \mu(\mathbf{x}_i)\right]\mathbf{x}_i \\
&= \sum_{i=1}^{N}\left[\mu(\mathbf{x}_i) - y_i\right]\mathbf{x}_i
\end{aligned}$$

$$\boxed{\frac{\partial}{\partial \mathbf{w}}\mu(\mathbf{x}) = \mu(\mathbf{x})\left(1-\mu(\mathbf{x})\right)\mathbf{x}}$$

$$\boxed{\begin{aligned}\mu(\mathbf{x}) &= \frac{\exp(\mathbf{w}^T\mathbf{x})}{\exp(\mathbf{w}^T\mathbf{x})+1}\\ 1\text{-}\ \mu(\mathbf{x}) &= \frac{1}{\exp(\mathbf{w}^T\mathbf{x})+1}\end{aligned}}$$

$$\frac{\partial}{\partial \mathbf{w}}\mu(\mathbf{x}) = \frac{\frac{\partial}{\partial \mathbf{w}}\left[\exp(\mathbf{w}^T\mathbf{x})\right]\left(\exp(\mathbf{w}^T\mathbf{x})+1\right) - \exp(\mathbf{w}^T\mathbf{x})\frac{\partial}{\partial \mathbf{w}}\left[\exp(\mathbf{w}^T\mathbf{x})+1\right]}{\left[\exp(\mathbf{w}^T\mathbf{x})+1\right]^2}$$

$$= \frac{\exp(\mathbf{w}^T\mathbf{x})\left(\exp(\mathbf{w}^T\mathbf{x})+1\right)\frac{\partial}{\partial \mathbf{w}}(\mathbf{w}^T\mathbf{x}) - \exp(\mathbf{w}^T\mathbf{x})\exp(\mathbf{w}^T\mathbf{x})\frac{\partial}{\partial \mathbf{w}}(\mathbf{w}^T\mathbf{x})}{\left[\exp(\mathbf{w}^T\mathbf{x})+1\right]^2}$$

$$= \frac{\exp(\mathbf{w}^T\mathbf{x})}{\left[\exp(\mathbf{w}^T\mathbf{x})+1\right]^2}\mathbf{x} = \mu(\mathbf{x})\left(1-\mu(\mathbf{x})\right)\mathbf{x}$$

$$\boxed{\frac{\partial}{\partial \mathbf{w}}(\mathbf{w}^T\mathbf{x}) = \frac{\partial}{\partial \mathbf{w}}(\mathbf{x}^T\mathbf{w}) = \mathbf{x}}$$

$$\boxed{\begin{aligned}\mu(\mathbf{x}) &= \frac{\exp(\mathbf{w}^T\mathbf{x})}{\exp(\mathbf{w}^T\mathbf{x})+1}\\ 1\text{-}\ \mu(\mathbf{x}) &= \frac{1}{\exp(\mathbf{w}^T\mathbf{x})+1}\end{aligned}}$$

▶梯度

算法与线性回归 $g(\mathbf{w})=\sum_{i=1}^{N}\left(f(\mathbf{x}_i)-y_i\right)\mathbf{x}_i$
看起来一样！
当然 $f(\mathbf{x})$不同（线性回归中 $f(\mathbf{x})=\mathbf{w}^T\mathbf{x}$）
事实上所有的线性模型的梯度都是如此

- 目标函数为

$$J(\mathbf{w})=\sum_{i=1}^{N}-\left[y_i\log(\mu_i)+(1-y_i)\log(1-\mu_i)\right]$$

- 梯度为

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

- 二阶Hessian矩阵为

$$\mathbf{H}(\mathbf{w})=\frac{\partial}{\partial \mathbf{w}}\left[\mathbf{g}(\mathbf{w})^T\right]=\sum_{i=1}^{N}\left(\frac{\partial}{\partial \mathbf{w}}\mu_i\right)\mathbf{x}_i^T$$

$$=\sum_{i=1}^{N}\mu_i(1-\mu_i)\mathbf{x}_i\mathbf{x}_i^T=\mathbf{X}^T\underbrace{diag\left(\mu_i(1-\mu_i)\right)}_{\mathbf{S}}\mathbf{X}=\mathbf{X}^T\mathbf{S}\mathbf{X}$$

正定矩阵，凸优化

牛顿法

- 亦称牛顿-拉夫逊（ Newton-Raphson ）方法
  - 牛顿在17世纪提出的一种近似求解方程的方法
  - 使用函数$f(x)$的泰勒级数的前面几项来寻找方程 $f(x)=0$的根
- 在求极值问题中，求 $g(\mathbf{w})=\dfrac{\partial J(\mathbf{w})}{\partial \mathbf{w}}=0$的根
  - 对应处$J(\mathbf{w})$ 取极值

▶ 牛顿法

一阶泰勒展开：$f(x) = f(x^{(t)}) + f'(x^{t})(x - x^{(t)})$

- 将导数 $\mathbf{g}(\mathbf{w})$ 在 $\mathbf{w}^t$ 处进行Taylor展开：

$$0 = \mathbf{g}(\hat{\mathbf{w}}) = g(\mathbf{w}^t) + (\hat{\mathbf{w}} - \mathbf{w}^t)\mathbf{H}(\mathbf{w}^t) + Op(\hat{\mathbf{w}} - \mathbf{w}^t)$$

- 去掉高阶无穷小 $Op(\hat{\mathbf{w}} - \mathbf{w}^t)$，从而得到

$$g(\mathbf{w}^t) + (\hat{\mathbf{w}} - \mathbf{w}^t)\mathbf{H}(\mathbf{w}^t) = 0 \quad \Rightarrow \quad \hat{\mathbf{w}} = \mathbf{w}^t - \mathbf{H}^{-1}(\mathbf{w}^t)\mathbf{g}(\mathbf{w}^t)$$

- 因此迭代机制为：

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \mathbf{H}^{-1}(\mathbf{w}^t)\mathbf{g}(\mathbf{w}^t)$$

  – 也被称为二阶梯度下降法，移动方向：$\mathbf{d} = -\left(\mathbf{H}(\mathbf{w}^t)\right)^{-1}\mathbf{g}(\mathbf{w}^t)$
  – Vs. 一阶梯度法，移动方向：$\mathbf{d} = -\mathbf{g}(\mathbf{w}^t)$ 移动

Iterratively Reweighted Least Squares（IRLS）

引入记号：

$$\mathbf{g}^t(\mathbf{w}) = \mathbf{X}^T(\boldsymbol{\mu}^t - \mathbf{y}),\quad \mu_i^t = \sigma\left(\left(\mathbf{w}^t\right)^T \mathbf{x}_i\right)$$

$$\mathbf{H}^t(\mathbf{w}) = \mathbf{X}^T\mathbf{S}^t\mathbf{X},\quad \mathbf{S}^t := \mathrm{diag}\left(\mu_1^t\left(1-\mu_1^t\right),\ldots,\mu_N^t\left(1-\mu_N^t\right)\right)$$

根据牛顿法的结果：

$$\begin{aligned}
\mathbf{w}^{t+1} &= \mathbf{w}^t - \left(\mathbf{H}^t\right)^{-1}\mathbf{g}^t \\
&= \mathbf{w}^t + \left(\mathbf{X}^T\mathbf{S}^t\mathbf{X}\right)^{-1}\mathbf{X}^T(\mathbf{y}-\boldsymbol{\mu}^t) \\
&= \left(\mathbf{X}^T\mathbf{S}^t\mathbf{X}\right)^{-1}\left[\left(\mathbf{X}^T\mathbf{S}^t\mathbf{X}\right)\mathbf{w}^t + \mathbf{X}^T\left(\mathbf{y}-\boldsymbol{\mu}^t\right)\right] \\
&= \left(\mathbf{X}^T\mathbf{S}^t\mathbf{X}\right)^{-1}\mathbf{X}^T\left[\mathbf{S}^t\mathbf{X}\mathbf{w}^t + \mathbf{y} - \boldsymbol{\mu}^t\right] \\
&= \left(\mathbf{X}^T\mathbf{S}^t\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{S}^t\left[\mathbf{X}\mathbf{w}^t + \left(\mathbf{S}^t\right)^{-1}\left(\mathbf{y}-\boldsymbol{\mu}^t\right)\right] \qquad \textit{where } \mathbf{z}^t = \mathbf{X}\mathbf{w}^t + \left(\mathbf{S}^t\right)^{-1}(\mathbf{y}-\boldsymbol{\mu}^t) \\
&= \left(\mathbf{X}^T\mathbf{S}^t\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{S}^t\mathbf{z}
\end{aligned}$$

加权最小二乘问题的标准方程

但权重矩阵S不是常数，而是依赖参数向量w。因此我们必须使用标准方程来迭代计算，每次使用新的权向量w来修正权重矩阵S。因此该算法被称为迭代再加权最小二乘（iterative reweighted least squares，IRLS）。

回忆最小二乘： $\hat{\mathbf{w}}_{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$

IRLS（cont.）

不止于代码

- 回忆最小二乘问题：
  - 目标函数：$J(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - \mathbf{w}^T \mathbf{x}\right)^2 = \left(\mathbf{y} - \mathbf{X}\mathbf{w}\right)^T \left(\mathbf{y} - \mathbf{X}\mathbf{w}\right)$
  - 解：$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$
- 回忆加权最小二乘问题：（Σ^{-1}： 权重矩阵）
  - 目标函数：$J(\mathbf{w}) = \left(\mathbf{y} - \mathbf{X}\mathbf{w}\right)^T \Sigma^{-1} \left(\mathbf{y} - \mathbf{X}\mathbf{w}\right)$
  - 解： $\hat{\mathbf{w}} = (\mathbf{X}^T \Sigma^{-1} \mathbf{X})^{-1} \mathbf{X}^T \Sigma^{-1} \mathbf{y}$
- IRLS中， $\mathbf{w}^{t+1} = \left(\mathbf{X}^T \mathbf{S}^t \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{S}^t \left[\mathbf{X}\mathbf{w}^t + \left(\mathbf{S}^t\right)^{-1} \left(\mathbf{y} - \boldsymbol{\mu}^t\right)\right]$
  - 相当于权重矩阵为 $\Sigma^{-1} = \mathbf{S}^t$
  - 由于$\mathbf{S}^t$是对角阵，$\mathbf{S}^t$相当于给每个样本的权重为 $S_{ii}^t = \mu_i^t \left(1 - \mu_i^t\right)$ ，

$$\mathbf{z}_i^t = \left(\mathbf{w}^t\right)^T \mathbf{x}_i + \frac{y_i - \mu_i^t}{S_{ii}^t}$$

Iteratively Reweighted Least Squares (cond)

- Iteratively reweighted least squares(IRLS)

$\mathbf{w} = \mathbf{0}_D$

$w_0 = \log(\bar{y} / (1 - \bar{y}))$

repeat

$\quad a_i = w_0 + \mathbf{w}^T \mathbf{x}_i$

$\quad \mu_i = \sigma(a_i)$

$\quad s_i = \mu_i (1 - \mu_i)$

$\quad z_i = a_i + \dfrac{y_i - \mu_i}{s_i}$

$\quad \mathbf{S} = \mathrm{diag}(\mathrm{s}_{1:\mathrm{N}})$

$\quad \mathbf{w} = \left(\mathbf{X}^T \mathbf{S} \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{S} \mathbf{z}$

until *converged*

$$\mathbf{S} = \mathrm{diag}\left(\mu_1\left(1 - \mu_1\right), \dots, \mu_N\left(1 - \mu_N\right)\right)$$

$$\mathbf{z}_i = \mathbf{w}^T \mathbf{x}_i + \frac{y_i - \mu_i}{\mu_i\left(1 - \mu_i\right)}$$

Weighted least square

▶拟牛顿法

- 牛顿法比一般的梯度下降法收敛速度快，但是在高维情况下，计算目标函数的二阶偏导数的复杂度很大，而且有时候目标函数的海森矩阵无法保持正定，不存在逆矩阵，此时牛顿法将不再能使用。
- 因此，人们提出了拟牛顿法。其基本思想是：不用二阶偏导数而构造出可以近似Hessian矩阵(或Hessian矩阵的逆矩阵)的正定对称矩阵，进而再逐步优化目标函数。不同的构造方法就产生了不同的拟牛顿法（Quasi-Newton Methods）
  - BFGS / LBFGS / Newton-CG

►正则化的Logistic回归

不 止 于 代 码

- 若损失函数取logistic损失，则Logistic回归的目标函数为

$$J(\mathbf{w}) = \sum_{i=1}^{N} -\left[y_i \log(\mu_i) + (1-y_i)\log(1-\mu_i) \right]$$

- 同线性回归类似，Logistic回归亦可加上L2正则

$$J(\mathbf{w}) = \sum_{i=1}^{N} -\left[y_i \log(\mu_i) + (1-y_i)\log(1-\mu_i) \right] + \lambda \|\mathbf{w}\|_2^2$$

- 或L1正则

$$J(\mathbf{w}) = \sum_{i=1}^{N} -\left[y_i \log(\mu_i) + (1-y_i)\log(1-\mu_i) \right] + \lambda |\mathbf{w}|$$

▶L2正则的Logistic回归求解

$$J(\mathbf{w}) = \sum_{i=1}^{N} -\left[y_i \log(\mu_i) + (1-y_i)\log(1-\mu_i) \right] + \lambda \|\mathbf{w}\|_2^2$$

- 梯度为: $g_{L2}(\mathbf{w}) = g(\mathbf{w}) + \lambda \mathbf{w} = \sum_{i=1}^{N} \left(\mu(\mathbf{x}_i) - y_i \right) \mathbf{x}_i + \lambda \mathbf{w} = \mathbf{X}^T (\boldsymbol{\mu} - \mathbf{y}) + \lambda \mathbf{w}$

- Hessian矩阵为： $\mathbf{H}_{L2}(\mathbf{w}) = \mathbf{H}(\mathbf{w}) + \lambda \mathbf{I} = \mathbf{X}^T \mathbf{S} \mathbf{X} + \lambda \mathbf{I}$

- 类似不带正则的Logistic回归，可采用（随机）梯度下降、牛顿法或拟牛顿法求解。

▶L1正则的Logistic回归求解

$$J(\mathbf{w}) = \sum_{i=1}^{N} -\left[y_i \log(\mu_i) + (1-y_i)\log(1-\mu_i) \right] + \lambda |\mathbf{w}|$$

- L1正则项的在0处不可导
- 在此我们L1正则的Logistic回归的牛顿法（IRLS）求解
  - 随机梯度下降（在线学习）在CTR预估部分讲解
- Recall：IRLS

$$\mathbf{w}^{t+1} = \left(\mathbf{X}^T \mathbf{S}^t \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{S}^t \mathbf{z} = \arg\min_{\mathbf{w}} \left\| \left(\mathbf{S}^t\right)^{1/2} \mathbf{X}\mathbf{w} - \left(\mathbf{S}^t\right)^{1/2} \mathbf{z} \right\|_2^2$$

- L1正则的Logistic回归在每次迭代中可视为一个再加权的Lasso问题：

$$\mathbf{w}^{t+1} = \arg\min_{\mathbf{w}} \left\| \left(\mathbf{S}^t\right)^{1/2} \mathbf{X}\mathbf{w} - \left(\mathbf{S}^t\right)^{1/2} \mathbf{z} \right\|_2^2, s.t. \|\mathbf{w}\|_1 < t$$

▶小结

- Logistic回归：
  - 损失函数：负log似然损失
  - 正则：L2/L1正则
  - 优化：梯度下降 / 牛顿法 / 拟牛顿法

THANK YOU