逻辑回归

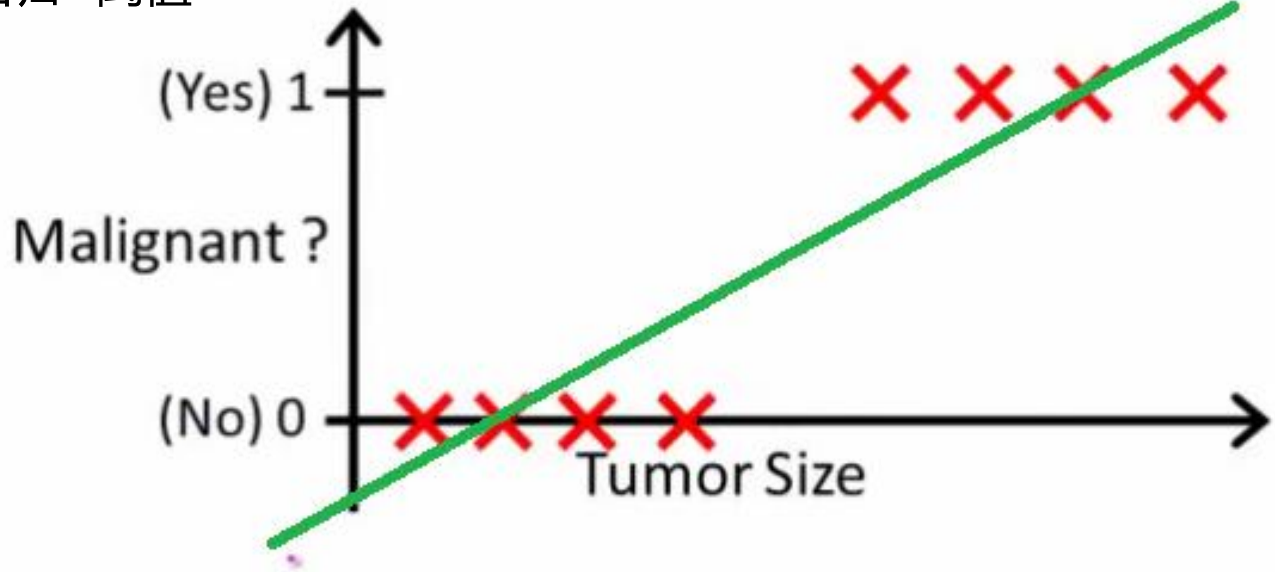
01

PART ONE

Logistic回归基本原理

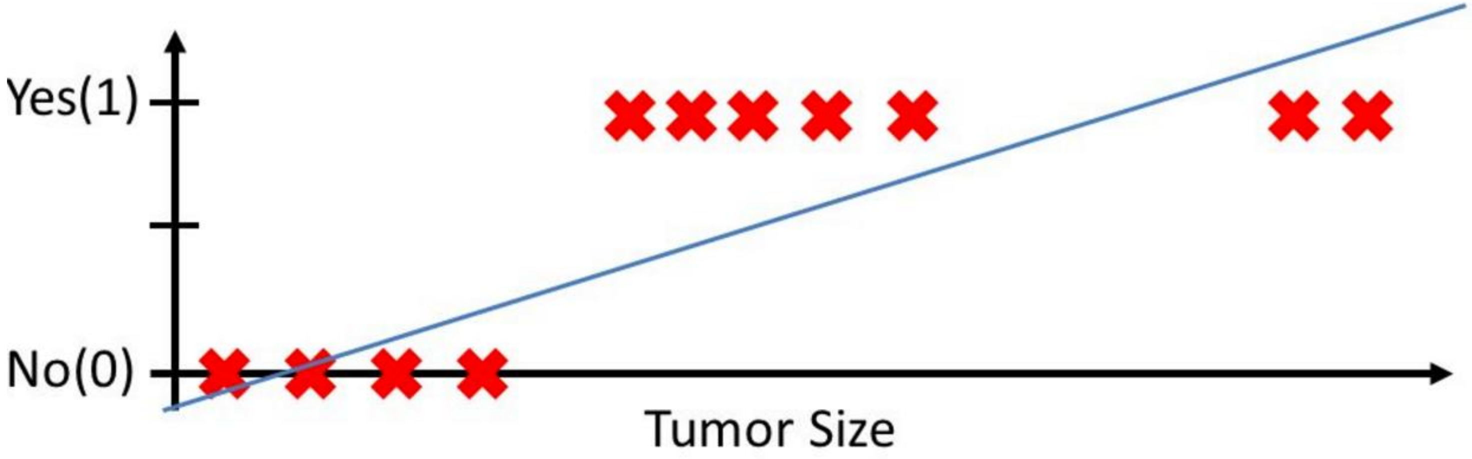
从线性回归到逻辑回归

这是一个分类问题
线性回归+阈值



从线性回归到逻辑回归

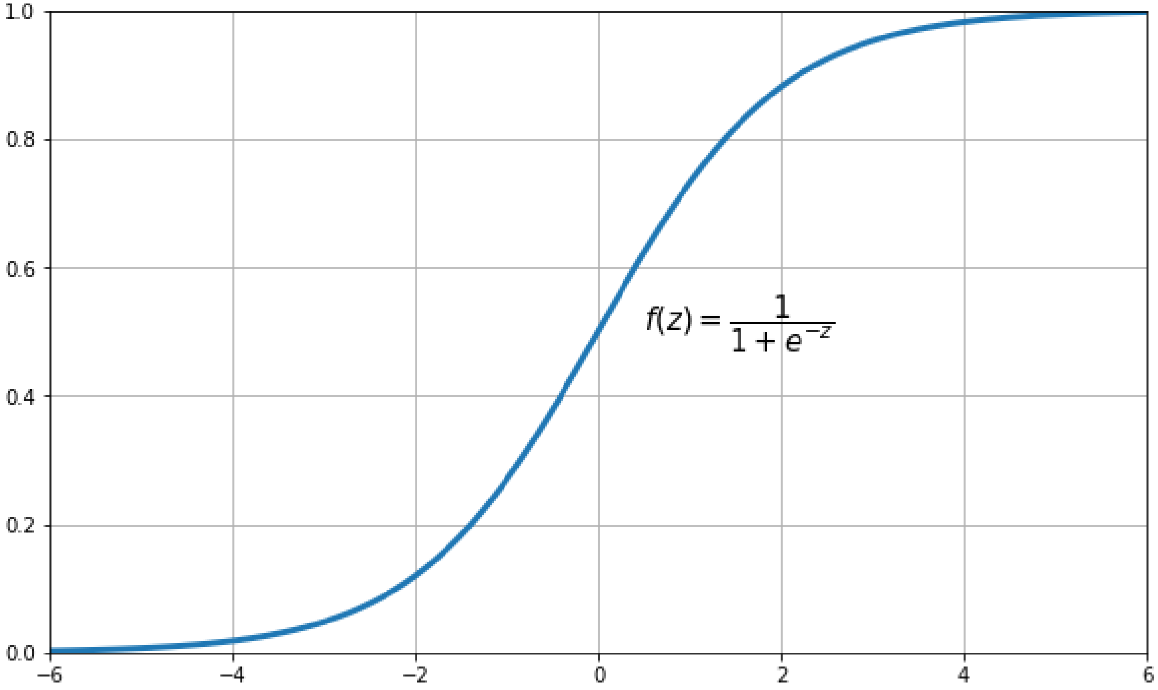
如果有噪声存在，很难界定阈值



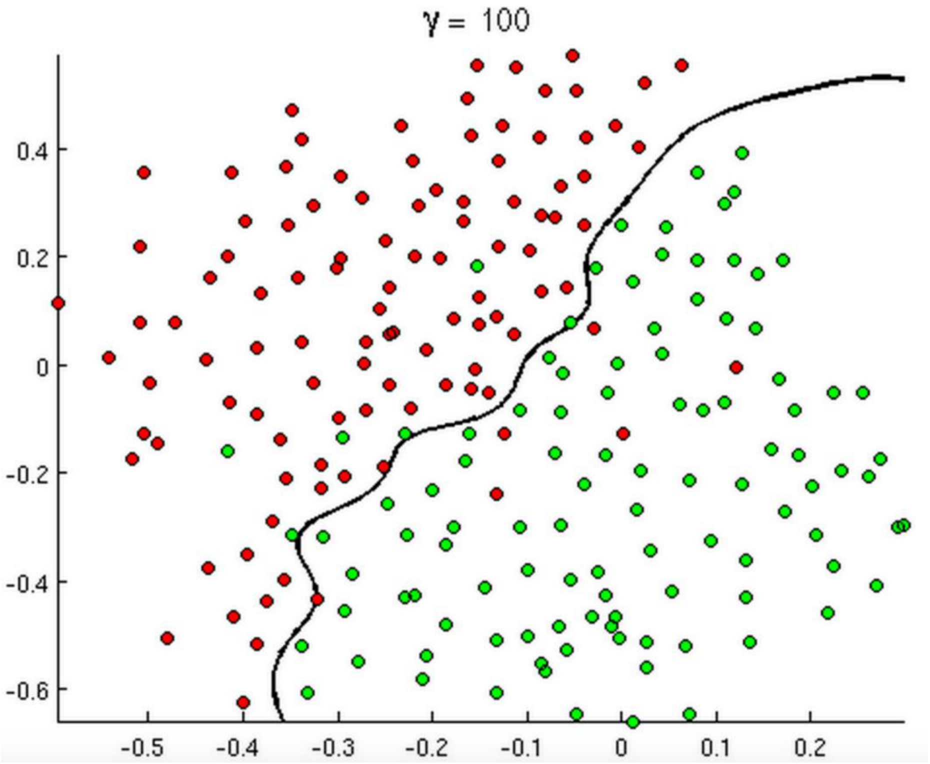
从线性回归到逻辑回归

主要归功于Sigmod函数

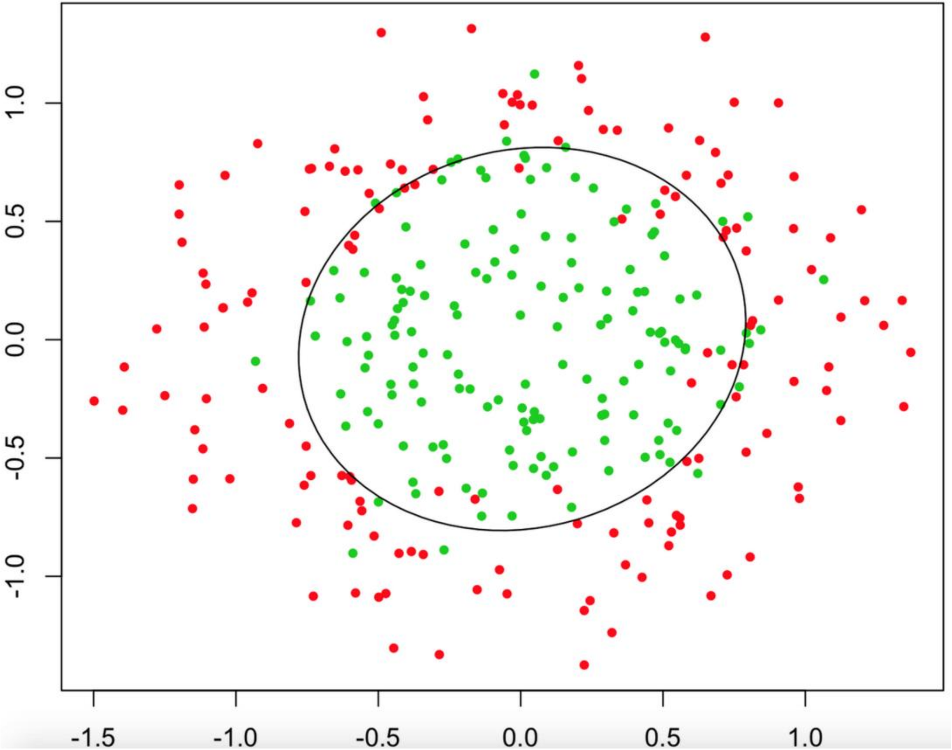
$$h_\theta(X) = g(\theta^T X) = \frac{1}{1+e^{-\theta^T X}}$$



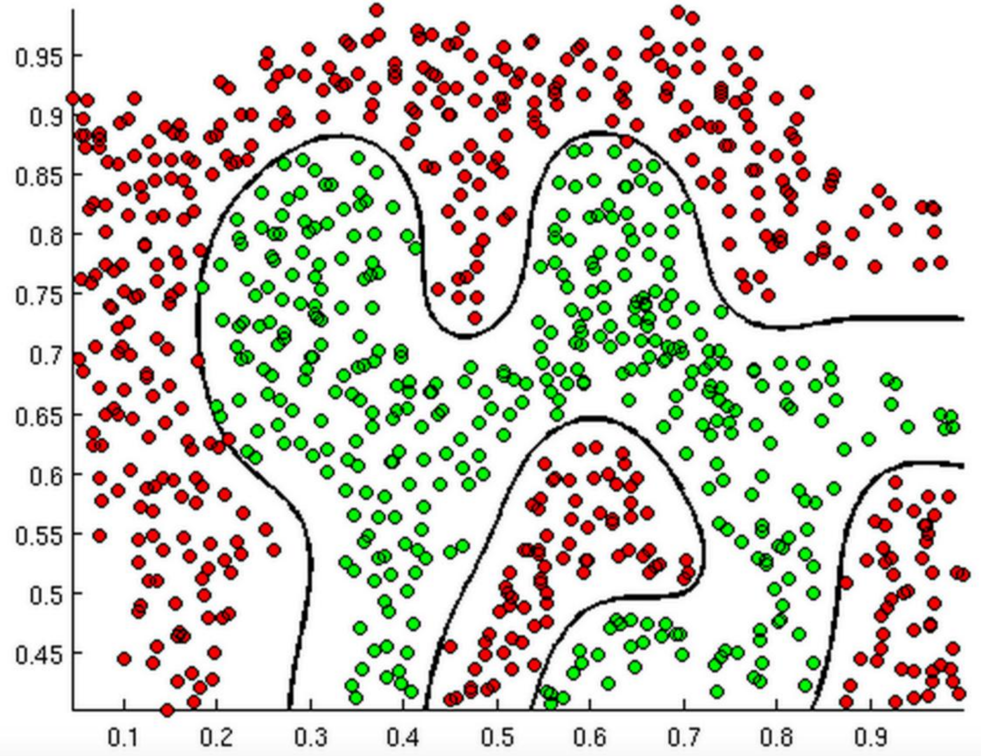
判定边界



判定边界

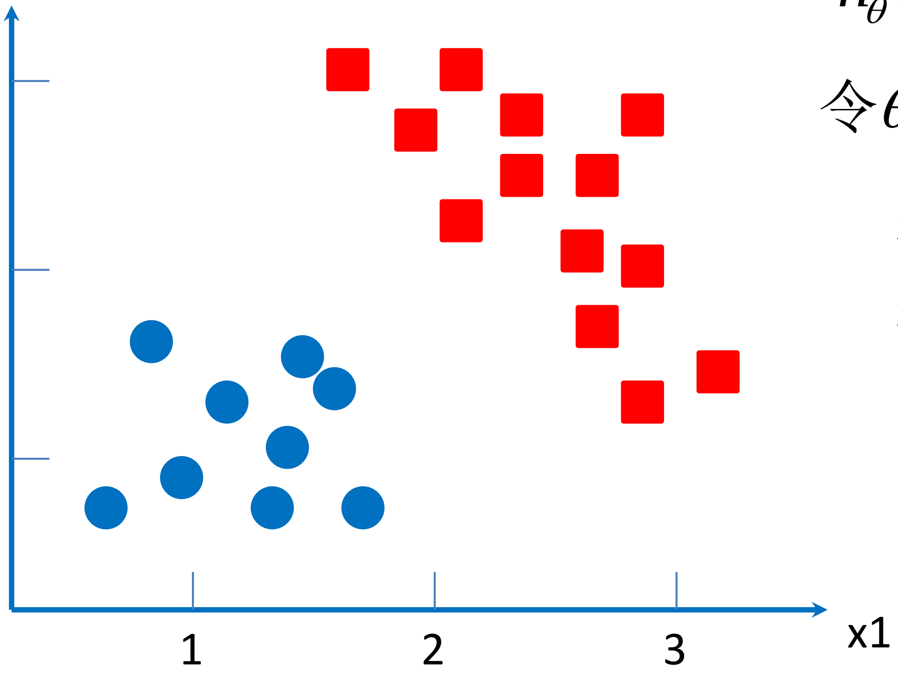


判定边界



逻辑回归的判定边界

线性判断边界



$$h_\theta(X) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

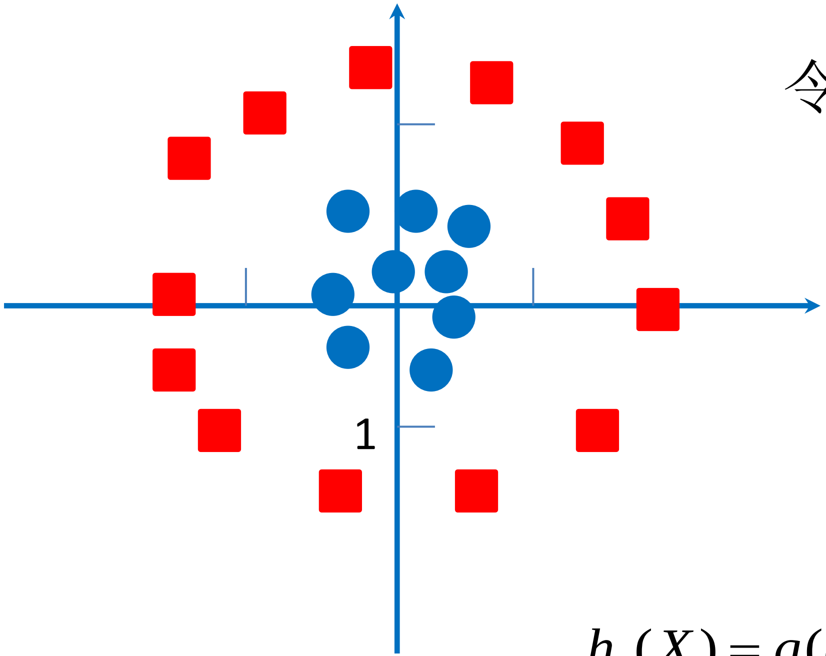
令$\theta_0 = -3$，$\theta_1 = 1$，$\theta_2 = 1$

若$3 + x_1 + x_2 \geq 0$，则$y = 1$

若$3 + x_1 + x_2 < 0$，则$y = 0$

逻辑回归的判定边界

非线性判断边界



$$h_\theta(X) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

令$\theta_0 = 1$，$\theta_1 = 1$，$\theta_2 = 1$，$\theta_3 = 1$，$\theta_4 = 1$

若$x_1^2 + x_2^2 - 1 \geq 0$，则$y = 1$

若$x_1^2 + x_2^2 - 1 < 0$，则$y = 0$

$$h_\theta(X) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1^2 x_2^2 + \ldots)$$

逻辑回归的判定边界

深层次的理解一下

对数机会比 (Log Odds Rati)

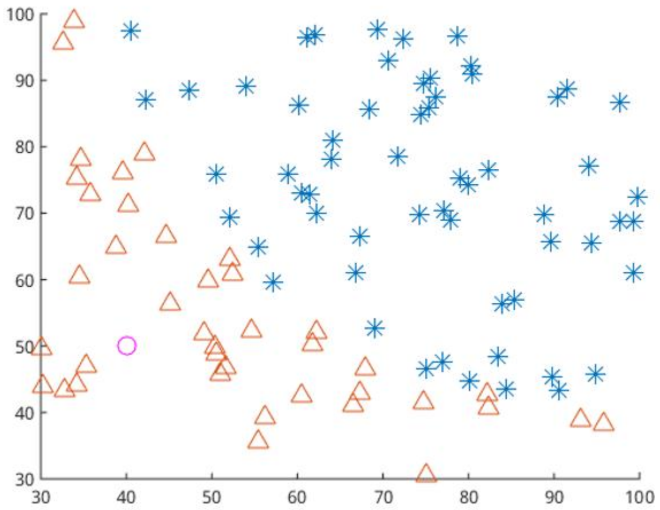
$$\mathrm{LOR}(X) = \ln \frac{y}{1-y} = \theta^T X$$

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

把y看做类后验概率估计p(y=1|X)，则有：

$$LOR(X) = \ln \frac{p(y=1|X)}{p(y=0|X)} = \theta^T X$$

$$\hat{y} = 1$$

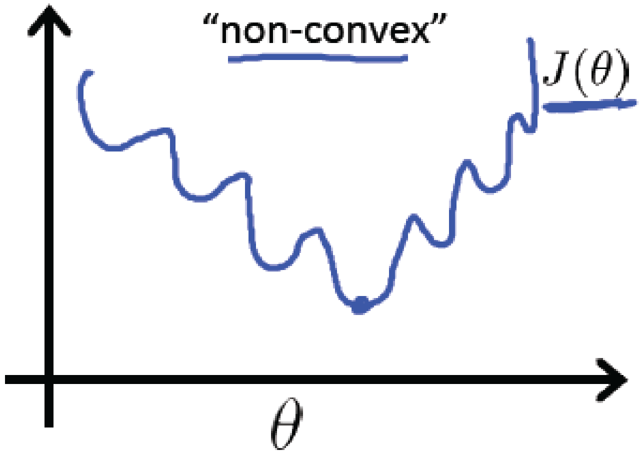


分类判定边界

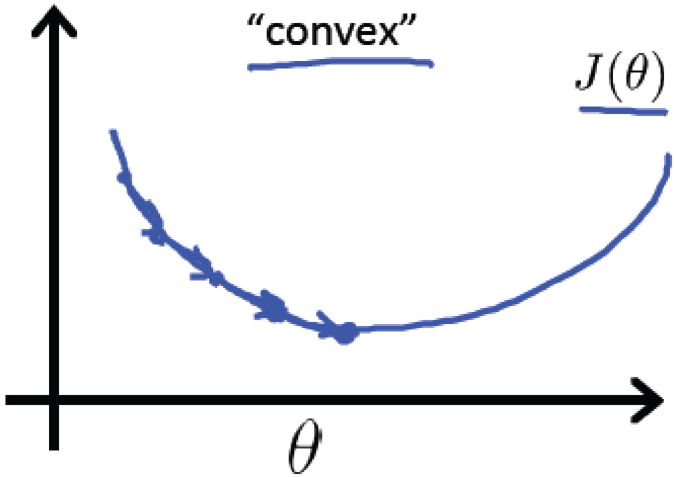
- 在逻辑回归里面
  - $\mathrm{LOR}(X) = \theta^T X > 0$，$\hat{y} = 1$
  - $\mathrm{LOR}(X) = \theta^T X < 0$，$\hat{y} = 0$
  - $\theta^T X = 0$：判定边界
- a(X)= θ^TX为判定边界
  - 因此Logistic回归是一个线性分类器

损失函数

$$J(\theta)=\frac{1}{2m}\sum_{i=1}^{m}\left(y_i-h_\theta(x_i)\right)^2$$

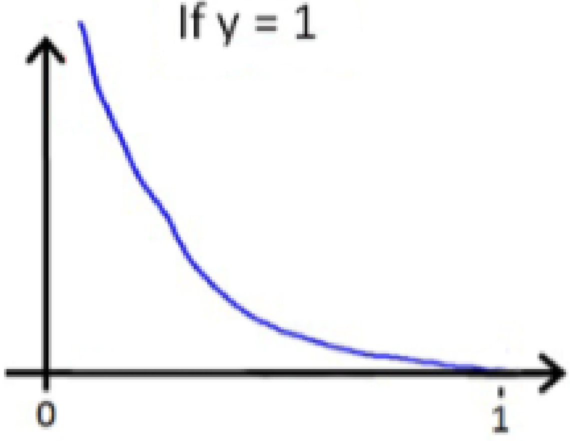


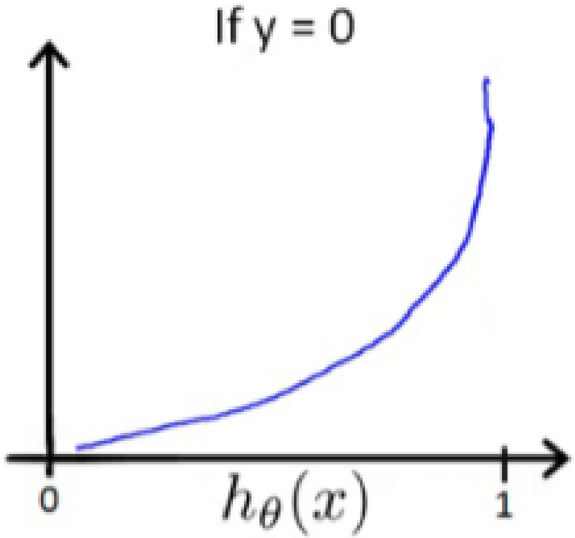
我们希望的下面的样子：



损失函数

$$Cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)), & \text{if } y=1 \\ -\log(1-h_\theta(x)), & \text{if } y=0 \end{cases}$$





损失函数

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}(y_i \log h_\theta(x_i) + (1-y_i)\log(1-h_\theta(x_i)))\right]$$

$$\Big\Downarrow \text{ 加正则化项}$$

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}(y_i \log h_\theta(x_i) + (1-y_i)\log(1-h_\theta(x_i)))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

梯度下降求参数

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}(y_i \log h_\theta(x_i) + (1-y_i)\log(1-h_\theta(x_i)))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

$$\theta_j = \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta)$$

02

PART TWO

多分类问题

多分类问题

我们已经知道，普通的logistic回归只能针对二分类(Binary Classification)问题，要想实现多个类别的分类，我们必须要改进logistic回归，让其适应多分类问题。主要有两个解决思路：

- One vs. Rest

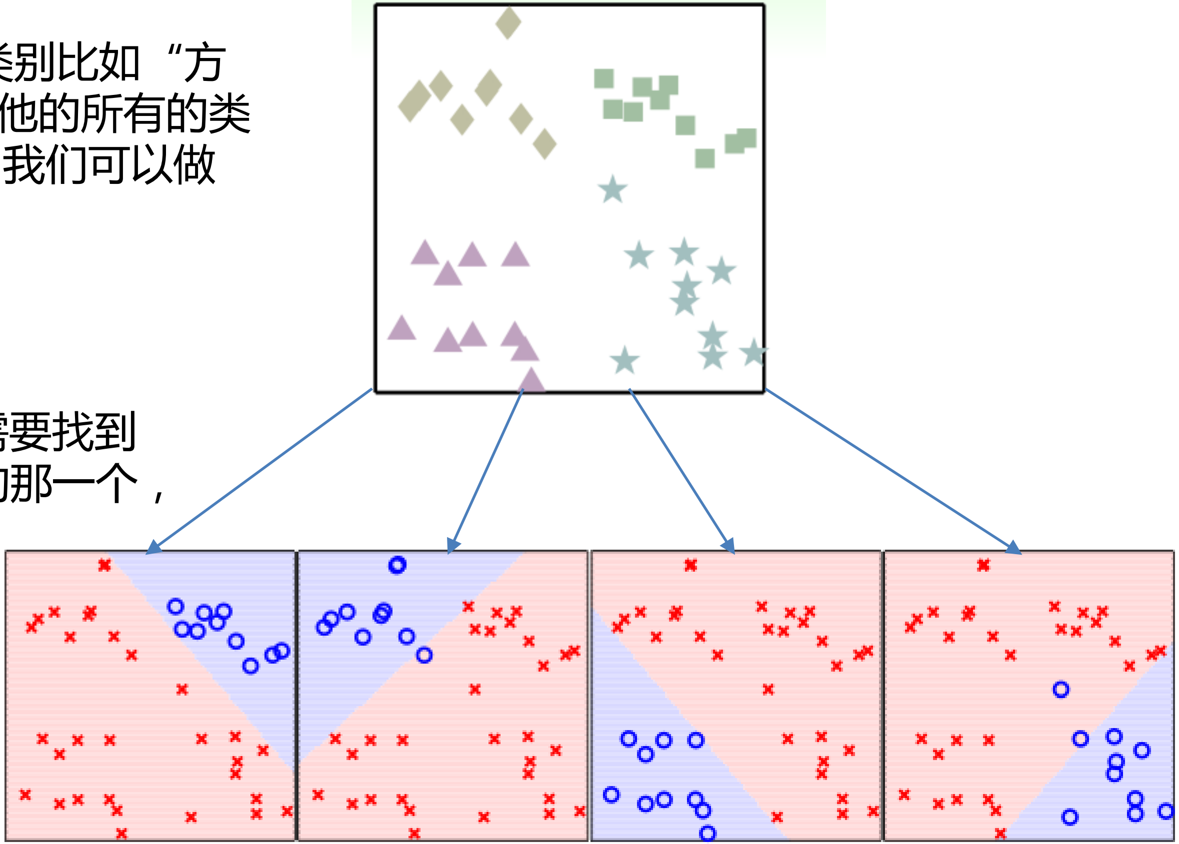
  直接根据每个类别，都建立一个二分类器。假如我们k有个类别，最后我们就得到了k个针对不同标记的普通的logistic分类器。

- Softmax回归

  修改logistic回归的损失函数，让其适应多分类问题。这个损失函数不再笼统地只考虑二分类非1就0的损失，而是具体考虑每个样本标记的损失。

One vs. Rest

思想：我们每次抽取一个类别比如“方块”，将其作为“o”，其他的所有的类别作为“×”，依次这样，我们可以做出4个分类器。



针对一个测试样本，我们需要找到这个分类函数输出值最大的那一个，即为测试样本的类别。

如果是正则LR，每类的模型都有自己正则参数。

Softmax分类器

从sigmoid函数（对应二项分布）扩展为softmax函数（对应多项 分布 Cat），有k个类别，其中某一类c有：

$$p(y=c|X,\theta)=Cat(y|S(\theta^T X))$$

Softmax 函数类似取最大函数：

$$S(\eta_c)=\frac{e^{\eta_c}}{\sum_{i=1}^{k} e^{\eta_c}}$$

综合起来：

$$p(y=c|X,\theta)=\frac{e^{\theta_c^T X}}{\sum_{i=1}^{k} e^{\theta_i^T X}}$$

Softmax回归

softmax回归算法的代价函数如下所示：

$$J(\theta) = -\sum_{i=1}^{m}\sum_{c=1}^{k} \text{sign}(y^{(i)} = c) \log p(y^{(i)} = c|x^{(i)}, \theta) = -\sum_{i=1}^{m}\sum_{c=1}^{k} \text{sign}(y^{(i)} = c) \log \frac{e^{\theta_c^T x(i)}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}$$

我们可以把logistic回归的损失函数改为如下形式：

$$J(\theta) = -\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) - (1 - y^{(i)}(1 - h_\theta x^{(i)})) = -\sum_{i=1}^{m}\sum_{c=0}^{1} \text{sign}(y^{(i)} = c) \log p(y^{(i)} = c|x^{(i)}, \theta)$$

扩展理解：https://www.cnblogs.com/Rambler1995/p/5467071.html

小结：Softmax分类器Logistic回归

- Softmax分类器能实现多类分类，是对Logistic回归在两类 分类任务上的扩展
- 优化算法和正则与两类分类Logistic回归类似
- 对于选择softmax分类器还是个logistic分类器，取决于所有类别之间是否互斥。所有类别之间明显互斥用softmax分类器，所有类别之间不互斥有交叉的情况下最好用个logistic分类器。

03

PART THREE

Scikitlearn 中的Logistic回归实现

Scikitlearn中的LogisticRegression实现

Scikitlearn提供的LogisticRegression实现为：

LogisticRegression(penalty= ‘l2’,dual=False,tol=0.0001,C=1.0,fit_intercept=True,intercept_scaling=1,class_weight=None,random_state=None,solver=’liblinear’,max_iter=100,multi_class=’ovr’,verbose=0,warm_start=False,n_jobs=1)

– Logistic回归的正则参数：penalty 、 C
– 优化求解参数：dual 、 solver 、 max_iter 、 tol 、 warm_start
– 模型参数：multi_class 、 fit_intercept 、 intercept_scaling 、 class_weight

LogisticRegression参数列表（一）

参数	说明
penalty	惩罚函数 / 正则函数，支持L2正则和L1正则，缺省：L2
dual	原问题（primal）还是对偶问题求解。对偶只支持L2正则和liblinear solver。当样本数n_samples>特征数目n_features时，缺省：False
tol	迭代终止判据的误差范围。 缺省:1e-4
C	C=1/λ，缺省：1
fit_intercept	是否在决策函数中加入截距项。如果数据已经中心化，可以不用。缺省：True
intercept_scaling	截距缩放因子，当fit_intercept为True且liblinear solver有效 所以还是对y做标准化预处理
class_weight	不同类别样本的权重，用户指定每类样本权重或‘balanced’（每类样本权重与该类样本本出现比例成反比）。缺省：None
random_state	混合数据的伪随机数。缺省：None

LogisticRegression参数列表（二）

参数	说明
solver	优化求解算法，可为‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, ‘saga’。缺省：liblinear
max_iter	最大迭代次数，当newton-cg, sag and lbfgs solvers时有效。缺省：100
multi_class	多类分类处理策略，可为‘ovr’, ‘multinomial’。‘ovr’为1对多，将多类分类转化为多个两类分类问题，multinomial为softmax分类。缺省：‘ovr’
verbose	是否详细输出
warm_start	是否热启动（用之前的结果作为初始化），对liblinear solver无效。缺省：False
n_jobs	多线程控制。缺省值-1，算法自动检测可用CPU核，并使用全部核

多类分类任务

- multi_class参数决定了多类分类的实现方式
- ‘ovr’： 即1对其他（one-vs-rest， OvR），将多类分类转化为多个二类 分类任务。为了完成第c类的分类决策，将所有第c类的样本作为正例， 除了第c类样本以外的所有样本都作为负例。
- ‘multinomial’： 多对多（many-vs-many， MvM）， 即softmax回归模型。
- OvR相对简单，但分类效果相对略差
  - 大多数情况，不排除某些情况下OvR更好
- MvM分类相对精确，但分类速度较OvR慢
- multi_class选择会影响优化算法solver参数的选择
  - OvR： 可用所有的slover
  - Multinomial： 只能选择newton-cg, lbfgs和sag / sag

优化求解算法solver(一)

- liblinear：使用了开源的liblinear库实现，使用坐标轴下降法来迭代优 化损失函数
- sag：随机平均梯度下降（Stochastic Average Gradient），是梯度下降 法的变种，每次迭代仅用一部分的样本来计算梯度，适合于样本多的 情况
- saga：sag的增强版本
- lbfgs：拟牛顿法的一种，利用损失函数二阶导数矩阵（Hessian矩阵）来迭代优化损失函数
- newton-cg：牛顿法家族的一种（共轭梯度）

优化求解算法solver（二）

- 对小数据集， ‘liblinear’ 是一个很好的选择， 而 ‘sag’ 和 ‘saga’ 对大数据集更快。
- 对多类分类问题， 只有 ‘newton-cg’, ‘sag’, ‘saga’ 和 ‘lbfgs’支持MvM（multinomial）， ‘liblinear’ 只支持OvR（one-versus-rest）的方式。
- ‘newton-cg’, ‘lbfgs’ 和 ‘sag’ 支持L2正则， 而 ‘liblinear’ 和 ‘saga’ 支持L1 正则。
- 注意： ‘sag’ 和 ‘saga’ 只有当特征有类似的尺度（scale）时能保证快速收敛。（对数据做标准化预处理）

优化求解算法solver选择

正则	求解算法	应用场景
L1	liblinear	如果模型的特征非常多，希望一些不重要的特征系数归零从而让模型系数稀疏的话，可以使用L1正则化。liblinear适用于小数据集
L1	saga	当数据量较大，且选择L1，只能采用saga
L2	liblinear	libniear只支持多元逻辑回归的OvR，不支持多项分布损失（MvM），但MVM相对精确。
L2	lbfgs/newton-cg/sag	较大数据集，支持OvR和 MvM两种多元logit回归。
L2	sag / saga	如果样本量非常大，sag / sga是第一选择

类别权重class_weight

- class_weight参数用于标示分类模型中各类别样本的权重
- 1.不考虑权重，即所有类别的权重相同
- 2.balanced：自动计算类别权重 – 某类别的样本量越多，其权重越低；样本量越少，则权重越高
  - 类权重计算方法为：n_samples/ (n_classes* np.bincount(y))
    - n_samples为样本数，n_classes为类别数量，np.bincount(y)输出每个类的样本数
- 3.手动指定各个类别的权重
  - 如对于0,1二类分类问题，可以定义class_weight={0:0.9, 1:0.1}，即 类别0的权重为90%，而类别1的权重为10%

04

PART FOUR

评价指标

评价参数

- 损失函数可以作为评价指标(log_loss、 zero_one_loss、hinge_loss)
  - logistic / 负log似然损失（log_loss）：

$$\log loss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} \log p_{ij}$$

    - M为类别数， p_{ij}为二值，当第i个样本为第j类时=1， 否则取0； 为模型预测的第i个样本为第j类的概率

- 0-1损失(zero_one_loss)（错误率、正确率评价指标均与此有关）

$$MCE = \frac{1}{N}\sum_{y_{pred} \neq y} 1$$

05

PART FIVE

项目实战

案列分析

Otto Group Product Classification Challenge

- 竞赛官网：https://www.kaggle.com/c/otto-group-roduct-classification-challenge
- 电商商品分类：
  – Target：共9个商品类别
  – 93个特征：整数型特征

扩展项目

扩展项目：泰坦尼克生还预测