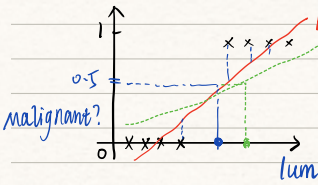
Classification ~ 逻辑回归 (Logistic Regression Model)

二分类 $y \in \{0,1\}$ Binary classification problem



$h_\theta(x) = \theta^T x$

(1) 通过 linear regression 解决 classification 问题:

设置 "threshold" $\begin{cases} h_\theta(x) \geq 0.5 \rightarrow \hat{y}=1 \\ h_\theta(x) < 0.5 \rightarrow \hat{y}=0 \end{cases}$

缺陷: ① 线性回归结果很容易受影响 → 当不能线性可分时, 自然不适用线性回归

② $h_\theta(x)$ 可能 >1, <0 while $y=0$ or 1

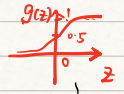
(2) 通过 Logistic Regression 解决 (注: 名称虽为"回归", 但实际上解决的是标签值为离散型的分类问题)

可以满足 $h_\theta(x) \in [0,1]$

Hypothesis

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

$g(\cdot): g(z) = \frac{1}{1+e^{-z}}$



Sigmoid function / logistic function

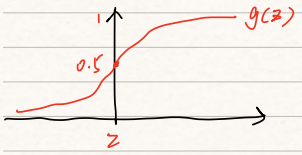
$y=0$ or 1

$P(y=1|x;\theta) + P(y=0|x;\theta) = 1$

output 介于 $[0,1]$ 之间, 对此输出的解释是: 代表了患病的概率: $h_\theta(x) = P(y=1|x;\theta)$

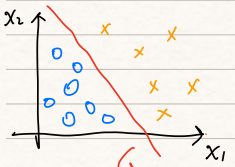
$h_\theta(x)$ give us the probability that our output is 1 eg: $h_\theta(x)=0.7$ 代表该病人有 70% 可能得病.

Decision Boundary



$h_\theta(x) = g(\theta^T x) = P(y=1|x;\theta)$ $\begin{cases} \geq 0.5, & "y=1" \Rightarrow \theta^T x \geq 0 \\ < 0.5, & "y=0" \Rightarrow \theta^T x < 0 \end{cases}$

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



$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$ → 预测 $y=1$, 当 $\theta_0 + \theta_1 x_1 + \theta_2 x_2 \geq 0$ 时.

↳ decision Boundary: 决策边界

(是假设 hypothesis 这一个属性, 决定了参数即决定边界)

而非训练集的属性! (可以有很多个决策边界).

Non-linear decision boundaries:

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

$\theta = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$ 则当 $-1 + x_1^2 + x_2^2 \geq 0$ 时, 预测 $y=1$

decision boundary

可能有更复杂的决策边界.

Cost Function

Loss function：在整个训练集上的损失

Cost function：在单个训练样本上的损失

对于线性回归，采用 MSE $\left[\frac{1}{2}\sum(y_i-\hat{y})^2\right]$ 作为 Cost function

但如果 logistic 回归也用这个作为 cost function，那会导致其非凸！

因此，我们采用另一种 cost function

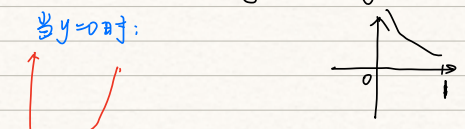
$$Cost(h_\theta(x),y)=\begin{cases}-\log(h_\theta(x)), & \text{if } y=1\\ -\log(1-h_\theta(x)), & \text{if } y=0\end{cases}$$

($y\in\{0,1\}$)

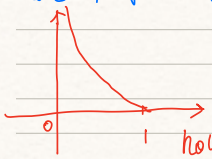
$$J(\theta)=\frac{1}{m}\sum_{i=1}^{m}Cost(h_\theta(x^{(i)}),y^{(i)})$$

其图像为：(右表)

可以先记 $-\log(x)$ 的图像



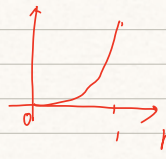
当 $y=1$ 时：



若预测为 1，则 cost=0

0，cost=∞

当 $y=0$ 时：



若预测为 1，则 cost=∞

0，=0

采用了这样的 cost function 后，$J(\theta)$ 就是凸的了（可以用 GD 得到最优解）

(没有 MSE 这样的多峰)

PS：

另个角度解释为何采用这样的 Cost function：MLE 的角度

$$\begin{cases}P(y=1\mid x;\theta)=h_\theta(x)\\ P(y=0\mid x;\theta)=1-h_\theta(x)\end{cases}$$

则：$P(y\mid x;w)=h_\theta(x)^y\cdot[1-h_\theta(x)]^{1-y}$ $\begin{cases}y=0\\ y=1\end{cases}$ 分别是各自的式子。通过指数巧妙地把分段函数写成一行.

似然：

$$L(\theta)=\prod_{i=1}^{n}P(y^{(i)}=1\mid x^{(i)};\theta)^{y^{(i)}}\cdot P(y^{(i)}=0\mid x^{(i)};\theta)^{1-y^{(i)}}$$

负对数：

$$-\ln L(\theta)=-\sum_{i=1}^{n}\left[y^{(i)}\ln h_\theta(x^{(i)})+(1-y^{(i)})\ln[1-h_\theta(x^{(i)})]\right]=J(\theta)$$

即：$J(h_\theta(x),y\mid\theta)=\begin{cases}-\ln(h_\theta(x)), & y=1\\ -\ln(1-h_\theta(x)), & y=0\end{cases}$ 一致.

化简 cost function 与 Gradient Descent.

回顾 logistic 回归：

$$h_\theta(x)=\frac{1}{1+e^{-\theta x}}$$

$Cost(x,y)=-y\log(h_\theta(x))-(1-y)\log(1-h_\theta(x))$ 从 $\begin{cases}y=0\\ y=1\end{cases}$ 两个角度来看，与上面是一致的...

整个数据集上：$J(\theta)=\frac{1}{m}\sum_{i=1}^{m}Cost(h_\theta(x^{(i)}),y^{(i)})$

entire cost function $=-\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log(h_\theta(x^{(i)}))+(1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right]$

向量形式：

$h=g(X\theta)$

$J(\theta)=\frac{1}{m}(-y^T\log(h)-(1-y^T)\log(1-h))$

目标：$\min J(\theta)$

Repeat: {

$\theta_j:=\theta_j-\alpha\cdot\frac{\partial}{\partial\theta_j}J(\theta)$ }（simultaneously update all θ_j）

* Repeat: { $\theta_j:=\theta_j-\alpha\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x_j^{(i)}$ 和线性回归的式子一样

向量形式：$\theta:=\theta-\frac{\alpha}{m}X^T(g(X\theta)-\vec{y})$

$(g(\square\square)-\square)$

其他的优化方式（除了梯度下降外）：Conjugate gradient，BFGS，L-BFGS

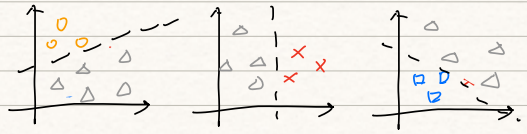
Classification ～多类one vs all (Multiclass Classification)

当 $y\in\{0,1\}$ 二分类

$y\in\{0,1,2,\dots,n\}$ 多类

我们可以将问题视为 $n+1$ 个二分类问题：

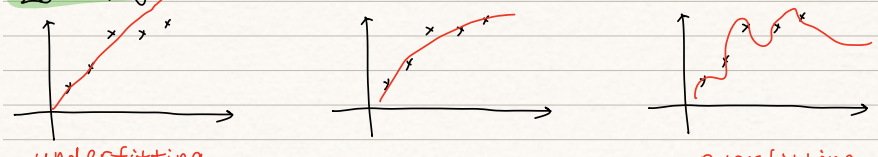
训练 $(n+1)$ 个二分类器（n+1类）



$h_\theta^{(0)}(x)=P\{y=0|x;\theta\}$ $h_\theta^{(1)}(x)=P\{y=0|x;\theta\}$ $h_\theta^{(2)}(x)=P\{y=0|x;\theta\}$

Prediction $=\max_i\{h_\theta^{(i)}(x)\}$

过拟合与欠拟合：



underfitting　　overfitting

1. 欠拟合/高偏差 high bias：模型假设对现有数据拟合的很差。通常是因为模型太简单，利用了较少的特征。

过拟合：模型对现有数据拟合得很好，但泛化能力差，不能很好地预测新数据。；复杂；多

高方差：high variance

如何解决过拟合？两条原则：目标：（降低模型复杂程度？）

(1) 减少特征数量
- 手动选择特征
- 使用模型选择算法

(2) 正则化
- 保留所有特征，但限制参数大小 θ_j
- 。

Regulization 的 Cost function

采用正则化，在不减少特征数且保留原来的假设的情况下，通过减小项的权重，来克服过拟合（项的权重很小 ≈0.0001，就认为没有，则模型简单）。

在 cost function 里加上这些权重，且提高其比重，则可以实现。

1. 对某些项进行正则：

$$\min_\theta \frac{1}{2m}\sum_{i=1}^{m}\left(h^{(i)}(x)-y^{(i)}\right)^2+1000\theta_3^2+1000\theta_4^2$$

$h(x)=\theta_0+\theta_1x+\theta_2x^2+\theta_3x^3+\theta_4x^4$

2. 对所有项进行正则：

$$\min_\theta \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h^{(i)}x-y^{(i)}\right)^2+\lambda\sum_{j=1}^{n}\theta_j^2\right]$$

λ：正则化参数
- 若太大，则欠拟合
- 若太小，仍然过拟合。

θ_j^2 → 平方项方便求梯度

（ps：θ_0 项不进行正则）

Regularized Linear Regression

应用正则化到线性回归：

(1) 方式一：梯度下降。因为修改了损失函数，所以在用梯度下降法更新参数时会对参数加以限制。特别地，我们分离出 θ_0，因为我们不对 θ_0 进行惩罚。

Repeat {

$$\theta_0 := \theta_0 - \alpha\sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha\left[\sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)x_j^{(i)}+\frac{\lambda}{m}\theta_j\right] \quad j\in\{1,2,\dots,n\}$$

θ_j 是当前更新的参数。

$$\theta_j := \left(1-\alpha\frac{\lambda}{m}\right)\theta_j - \alpha\sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)x_j^{(i)}$$

合并

$\alpha>0$，$\lambda>0$，$m>0$，则 $(1-\alpha\frac{\lambda}{m})<1$，起到缩小 θ_j 的目的。

(2) 方式二：通过 Normal Equation：

$\theta = (X^TX + \lambda \cdot L)^{-1}X^Ty$

$L = \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \end{bmatrix}$。注意若 X^TX 不可逆时，加上 λL 可使其变为可逆。

Regularized Logistic Function

overfitting.

逻辑回归的 cost Function：

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

添加正则项：$+\frac{1}{2m}\sum_{j=1}^{n}\theta_j$

Gradient descent:

Repeat { $\theta_0 := \theta_0 + \sum_{i=1}^{m}($

附注：Logistic 回归梯度下降推导

预备知识：sigmoid 函数求导：

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

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

$$g'(z) = g(z)(1-g(z))$$

求梯度推导：

$$\frac{\partial}{\partial\theta_j}J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\cdot\frac{1}{h_\theta(x^{(i)})}\cdot\frac{\partial h_\theta(x^{(i)})}{\partial\theta_j} + (1-y^{(i)})\cdot\frac{(-1)}{1-h_\theta(x^{(i)})}\cdot\frac{\partial h_\theta(x^{(i)})}{\partial\theta_j}\right]$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\cdot\frac{1}{h_\theta(x^{(i)})} - (1-y^{(i)})\frac{1}{1-h_\theta(x^{(i)})}\right]\frac{\partial h_\theta(x^{(i)})}{\partial\theta_j}$$

$$\frac{\partial h_\theta(x^{(i)})}{\partial\theta_j} = h_\theta(x^{(i)})\cdot[1-h_\theta(x^{(i)})]\cdot[\theta^Tx^{(i)}]'$$

其中 $\theta^Tx^{(i)} = \theta_0 + \theta_1x_1^{(i)} + \theta_2x_2^{(i)}\ldots$ $\therefore [\theta^Tx^{(i)}]$ 关于 θ_j 的导数是 $x_j^{(i)}$，为一个常数。

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\cdot\frac{1}{h_\theta(x^{(i)})} - (1-y^{(i)})\frac{1}{1-h_\theta(x^{(i)})}\right]\cdot h_\theta(x^{(i)})\cdot[1-h_\theta(x^{(i)})]\cdot x_j^{(i)}$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}[1-h_\theta(x^{(i)})] - [1-y^{(i)}]\cdot h_\theta(x^{(i)})\right]\cdot x_j^{(i)}$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} - h_\theta(x^{(i)})\right]x_j^{(i)}$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left[h_\theta(x^{(i)}) - y^{(i)}\right]x_j^{(i)}$$

∴梯度下降法更新参数：$\theta_j := \theta_j - \alpha\cdot\frac{\partial h_\theta(x^{(i)})}{\partial\theta_j}$

$$= \theta_j - \alpha \cdot \frac{1}{m} \sum_{i=1} \left[h_\theta(x^{(i)}) - y^{(i)} \right] x_j^{(i)}$$