大数据

入门
机器学习

青年AI自强项目—情况更新

学堂在线
xuetangx.com

直播平台改变

录播发布机制改变

作业发布及回收机制改变

所有作业的截止时间改为：

最后一次讲座之后两周，固定周期评分

为了满足大家单纯的学习愿望

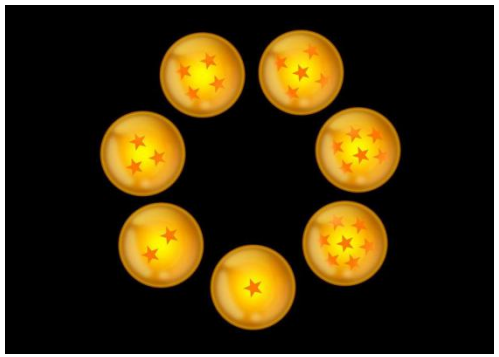
序号-形式	讲座/活动主题
1-讲座	AI 鸟瞰与升级指南
2-讲座	机器学期入门
3-讲座	经典神经网络
4-讲座	深度神经网络
5-讲座	卷积神经网络
6-讲座	分类任务
7-讲座	探测任务
8-讲座	实例与调参
9-任务	转化挑战

希望大家能够坚持下去

ML是一座能令你学会“召唤技能”的神奇乐园

游玩项目→“知识景点”

游玩目标→到达“里程碑点” →学会召唤技能→召

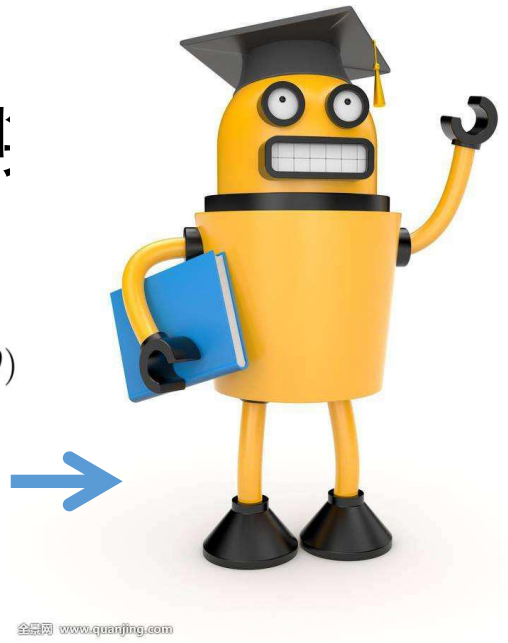


法宝+咒语

$$\arg\min_{\theta} F(\mathcal{D};\theta) = \mathcal{L}\left(\{x_i, y_i\}_{i=1}^{n};\theta\right) + \Omega(\theta)$$

base.py	2018/10/8 14:25	PY 文件	5 KB
branch1.py	2018/10/6 9:52	PY 文件	4 KB
branch2.py	2018/10/9 16:11	PY 文件	4 KB
branch3.py	2018/10/9 17:54	PY 文件	3 KB
branch4.py	2018/10/8 16:42	PY 文件	2 KB
branch5.py	2018/10/8 16:23	PY 文件	1 KB
branch6.py	2018/10/8 17:55	PY 文件	2 KB
data.csv	2018/10/5 1:43	Microsoft Excel ...	25 KB
data.xls	2018/10/5 1:43	Microsoft Excel ...	25 KB

数学公式+代码



计算机小弟

召唤一个“女朋友” 当小弟？

我们可以想办法让知识变得更有趣

美女机器人“佳佳”

美女机器人“Android ‘U’ ”

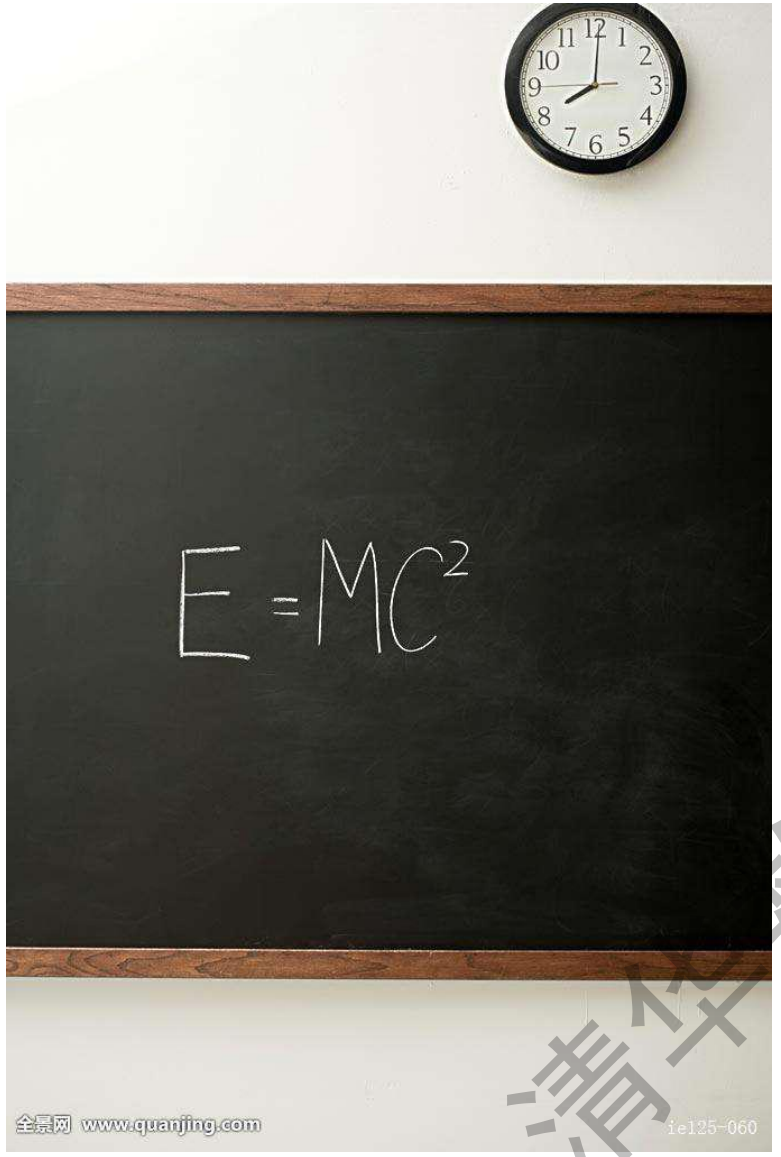
要有当小弟的觉悟

以游览清华为例

原则-不让细节牵绊你

目标：绝不“呆萌”的成为懵逼触发者

描述挂表的位置



自然语言→定型描述：黑板上边

数学语言→定量描述：（1.5,2.5）

数学语言：
简洁、准确、完备

数学之美-啥是机器学习?

自然语言：

机器学习算法的本质是找到一些特征，来描述一些数据/问题，并做出一些预测。具体的办法是将找到的特征输入到一个模型中，通过对准备好的数据集进行学习，来更新模型中的参数。为了评价模型现有参数的好坏并确定更新后的参数是否更优，我们需要借助一把“尺子”来度量目前模型参数与“最优解”之间的差距，更新模型中参数的目标就是使这种差距减到最小。另一方面，我们还需要引入正则手段来防止模型中的参数“过分”追求现有数据集上的“最优解”，“过犹不及”这个道理同样适用于机器学习。

数学语言：

$$\arg\min F(\mathcal{D};\theta) = \mathcal{L}\left(\{x_i, y_i\}_{i=1}^{n};\theta\right) + \Omega(\theta)$$

本次课程的目标

利用简单的机器学习算法回答：什么样的男生会受到小姐姐的青睐

沿着一个个由机器学习的关键“知识景点”标识出来的最有效路径一路走到首个“里程碑点”

特征与数据集的描述

序号	身高	月薪（	是否有兴趣尝试交往
1	1.71	4704	0
2	1.64	10139	0
3	1.58	5490	0
4	1.75	1236	0
5	1.95	8963	1
6	1.47	2573	0
7	1.64	10195	0
8	1.7	3565	0
9	1.82	3867	0
10	1.74	11527	1
11	1.75	6875	1
12	1.6	4690	0
13	1.9	1541	0
14	1.83	6842	1
15	1.83	3166	0
16	1.67	1400	0
17	1.73	6002	0

结构化数据
样本集和标签集

$$Y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \dots \\ y^{(50)} \end{bmatrix}$$

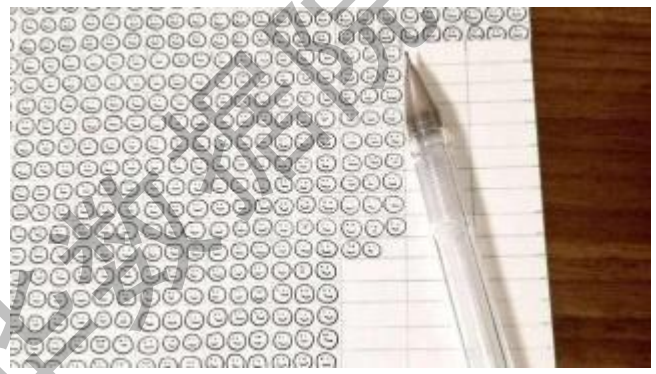
数学语言描述

x表示小哥哥
最熟悉的未知数

$x^{(1)}$代表1号小哥哥
为样本（小哥哥）编号

x_1代表1号特征“身高”
为特征（身高/收入）编号

$x_1^{(3)}$代表啥？
引用所有元素



计算机科学家强迫症

$$X = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} \\ x_1^{(2)} & x_2^{(2)} \\ \dots & \dots \\ x_1^{(50)} & x_2^{(50)} \end{bmatrix}$$

整齐的摞起来写

如果有人问：
你的样本集是啥？

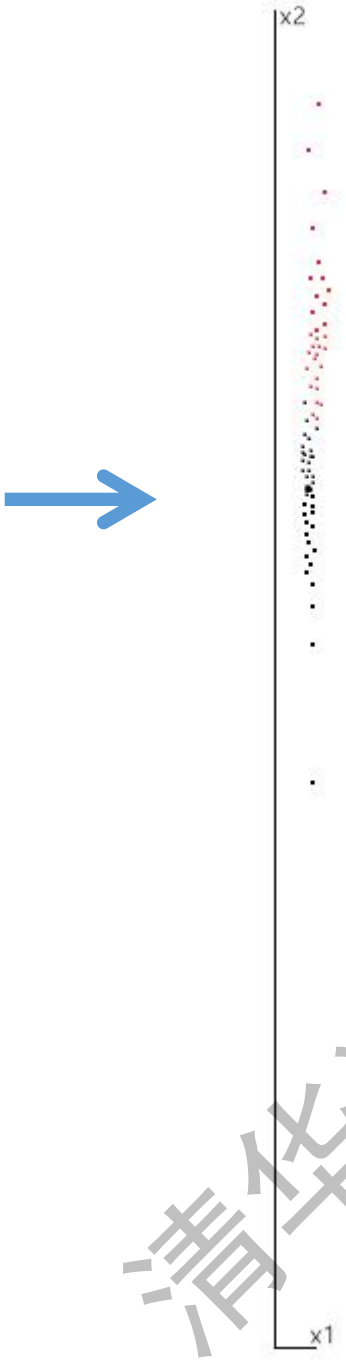
数据可视化
（Data Visualization）

x_1：横轴
x_2：纵轴

- →不受欢迎
- →受欢迎

不考虑单位影响

同例尺画出图像



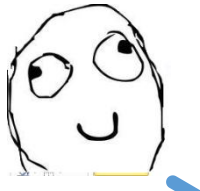
看不清？→在黑板上画

原因：
x_1的取值范围：$[1.59, 1.87]$
x_2的取值范围：$[1794, 12128]$
二者分布范围相差了4个数量级
所以代表样本的小点必定分布在一个极为窄高的区域内
左边的图像也是示例，在保证人眼能分辨的情况下真实的图像恐怕要画到一张0.002米宽，1.3米高的白纸上。

归一化（Normalization）

归一化公式

解决办法



很土的读法

往±1里凑一凑

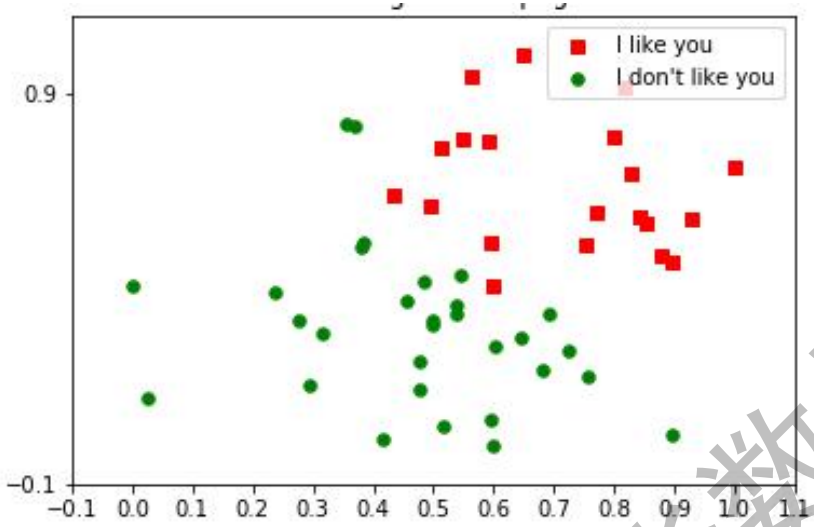
$$x^i_{norm} = \frac{x^i - \mu}{\sigma}$$

1号“法宝”

```
# 归一化公式
def normalization(X):
    mu = np.mean(X, axis=0)
    sigma = np.std(X, axis=0)
    X_norm = (X - mu) / sigma
    return X_norm
```

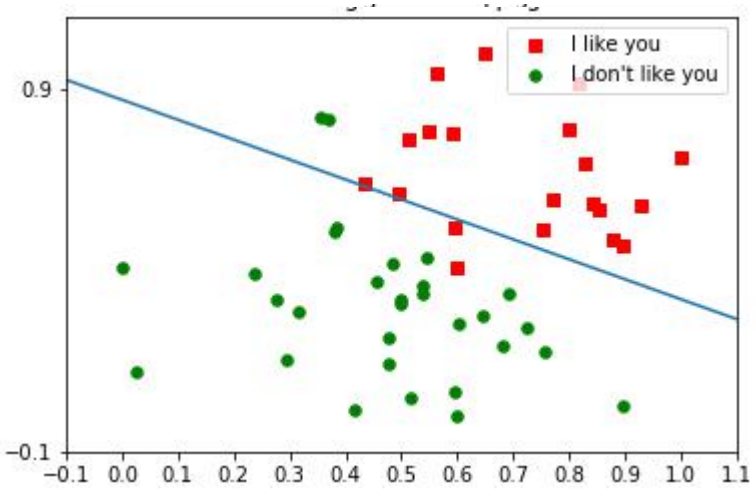
口诀

召唤



归一化处理后的数据

决策边界
（Decision Boundary）



画一条线来分界

$y = kx + b$ → $0 = b + (-1)y + kx$ → 机器学习中：$0 = \theta_0 + \theta_1 x_1 + \theta_2 x_2$

高中知识　　改写　　解析几何中：$0 = b + (-1)y + kx$

两类符号，换个马甲

“模型”=“参数的结构”+“变量的输入方法”

CS很“懒”

公式太长 → $0 = \theta_0 + \theta_1 x_1 + \theta_2 x_2$

要写短点 → $0 = \theta X$

矩阵运算

$$0 = \theta X \quad \rightarrow \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} \quad X = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \quad \rightarrow \quad \theta^T X = [\theta_0 \quad \theta_1 \quad \theta_2] \times \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$$

真的能这么写吗？

θ与X都是矩阵

写起来需要多一个角标T

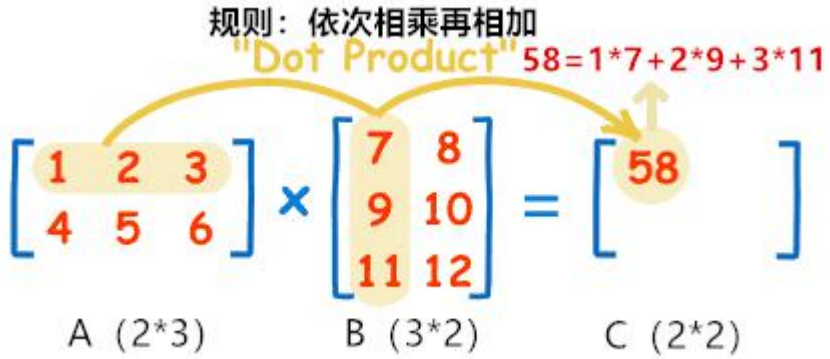
$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} \quad \text{vs} \quad \theta^T = [\theta_0 \quad \theta_1 \quad \theta_2]$$

矩阵的转置

1、将原矩阵沿对角线做镜面反转

2、i行、j列的元素，转移到j行、i列处

3、转置仅改变矩阵的尺寸，元素取值不变



矩阵的乘法

1、用 c_{ij}代表矩阵C中第i行j列的元素

2、c_{ij}等于矩阵A第i行与矩阵B第j列的元素，**依次相乘并相加**

3、矩阵乘法没有交换律，尺寸对不上就没办法“**依次相乘**”

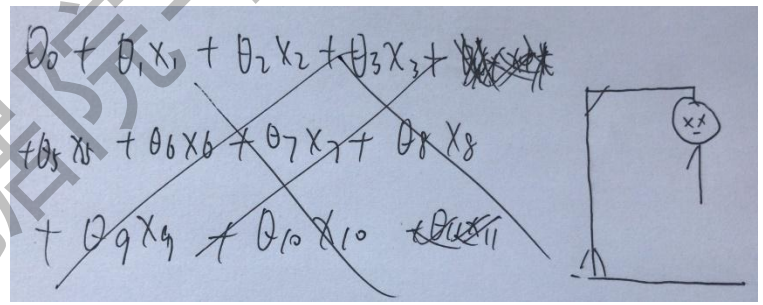
决策边界
（Decision Boundary）

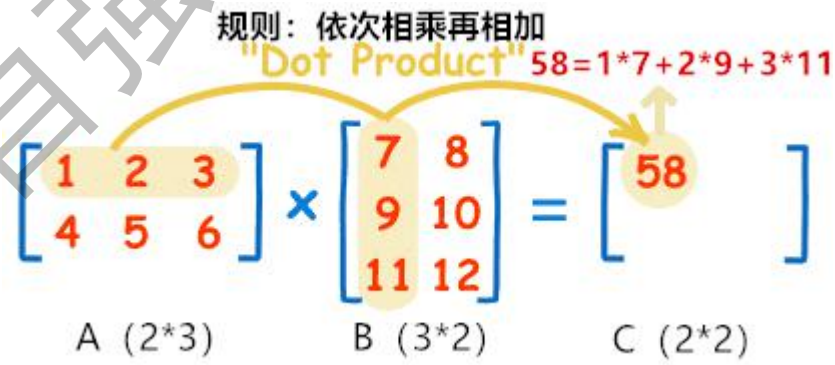
这是什么独特的“犯懒”姿势？

数学不好不要骗我

$$\rightarrow \theta_0 + \theta_1 x_1 + \theta_2 x_2 = \theta^T X \rightarrow$$

为了写短一小点





要发明这么复杂的规则吗😓

实际问题：
50万个特征 →

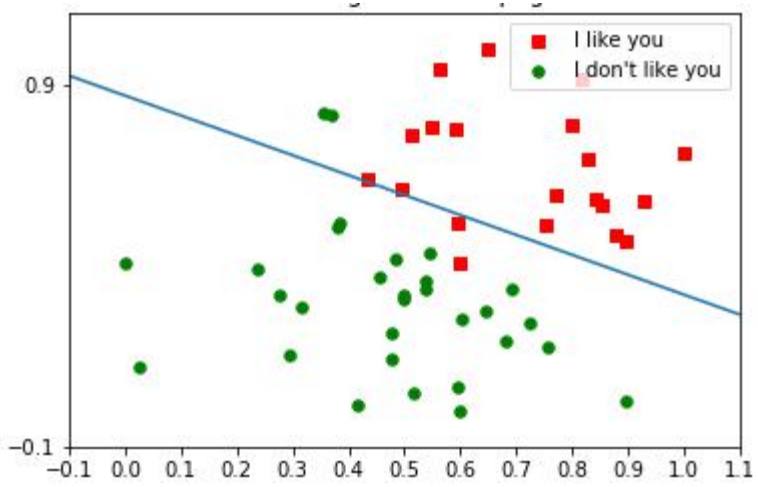
如果有人问：
你的模型是啥？

$$\theta^T X$$

2号法宝：矩阵化表示

不会矩阵化表示的小菜鸟
念到世界的尽头……

如何做预测



刚刚画出的决策边界

$$\theta^T X = \theta_0 + \theta_1 x_1 + \theta_2 x_2 = -5.6 + 4.2x_1 + 6x_2$$

具体参数如上

验证：

1、坐标系中任意一点带入直线的表达式$\theta^T X$；
2、如果得到负值则表示这一点在直线下方，反之则表示其在直线上方
3、且绝对值越大表示该点距离直线越远
4、如果得到0，则说明这一点恰好落到了直线上。

结论：

1、新的样本，只需要将其输入表示决策边界的模型$\theta^T X$中
2、并通过输出数值的正负来判断这位小哥哥是否受欢迎

Sigmiod函数

新的问题：
如果一个小哥哥来做预测
我们告诉他“你受欢迎的数字是2”

```
while(1){
    ;
}
```

死循环懵逼

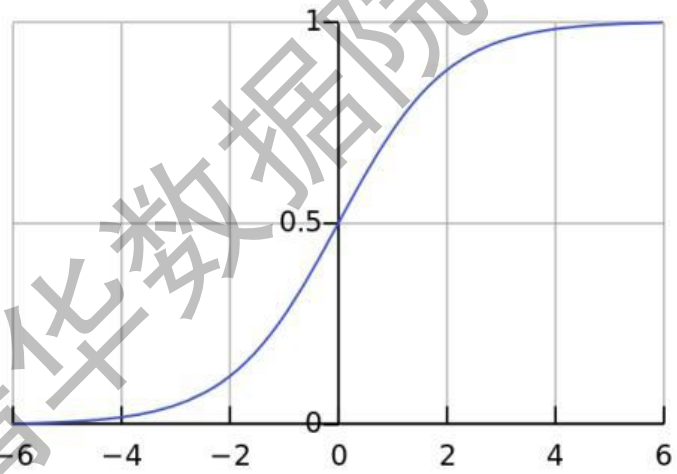
急需新法宝

要求：

1、输出的取值范围是（0，1）
2、输入可以是（-∞,+∞）中的任意数字；
3、输入0时输出时0.5，输入负值时输出小于0.5，且输入负值的绝对值越大，输出的数值约接近于0，反之亦然；

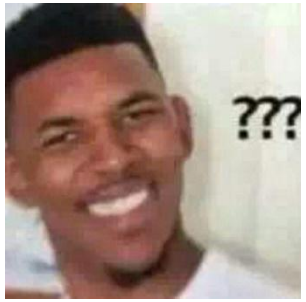
$$g(z) = \frac{1}{1+e^{-z}}$$

3号法宝：sigmoid函数



sigmoid函数图像

来历：
最大熵
拉格朗日乘数法



有兴趣的同学可以自行寻求懵逼

“逻辑回归”（Logistic regression）

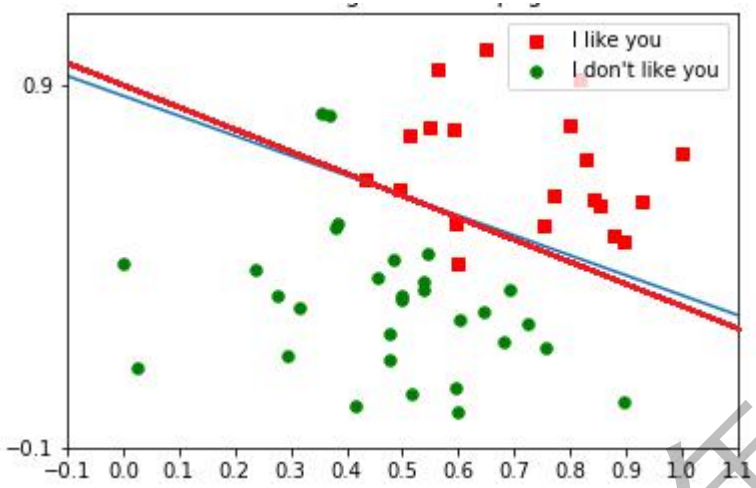
$$\theta^T X = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \quad + \quad g(z) = \frac{1}{1+e^{-z}} \quad = \quad h_\theta(x) = g(\theta^T X) = \frac{1}{1+e^{-\theta^T X}}$$

线性模型　　sigmoid函数　　逻辑回归模型

今天的主角：“逻辑回归”又作“对率回归”

如何评价参数的好坏

凭感觉画的线再来一条
如何评价好坏？



$$Y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \dots \\ y^{(50)} \end{bmatrix}$$

可以与正确答案做对比

求新法宝

要求：

1、将现有的模型输入，能够得到一个分布在（0，+∞）范围内的数值，来表示该模型误差的大小；
2、模型的预测值约接近正确答案，这个表示误差的数值应该越小；
3、模型的预测值离正确答案越远，这个表示误差的数值应该越大。

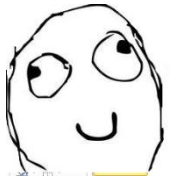
“损失函数”（Loss function）

4号法宝：损失函数

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

符号看不懂？公式好长好吓人？

σ　　∑



换个马甲　　求和符号　　也有很土的读法

都加起来

估计是用来偷懒的

$$x^1+x^2+x^3+.....+x^{(i)}+.....+x^{50}$$

可以简写成 $\sum_{i=1}^{50}x^{(i)}$

1个符号胜过千言万语

$(1-y^{(i)})\log(1-h_\theta(x^{(i)}))$

$(y^{(i)})\log(h_\theta(x^{(i)}))$

公式拆成两半

如果y=1

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

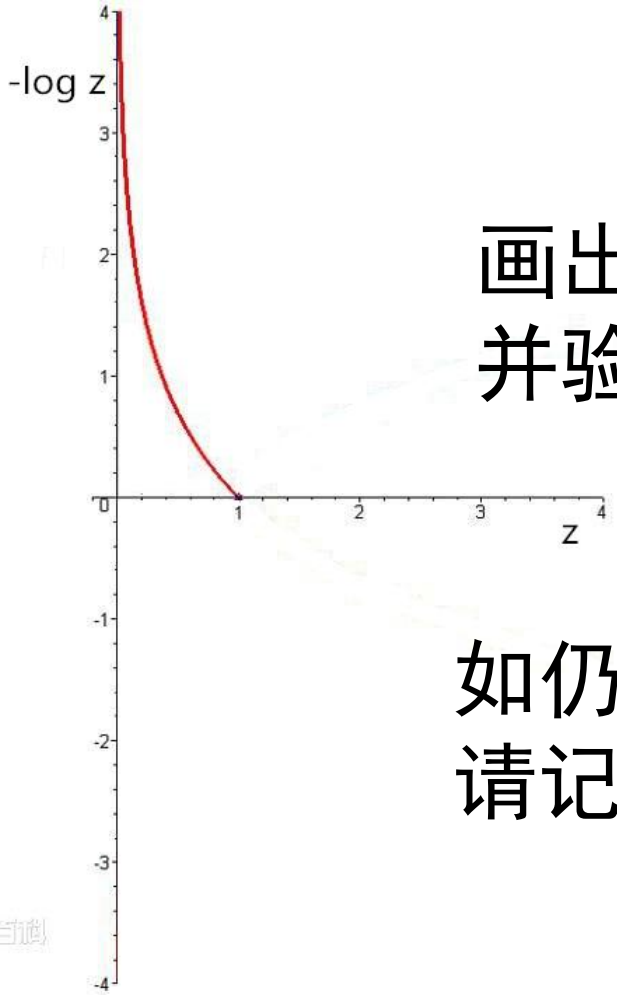
只看关键部分

$$-\log(h_\theta(x^{(i)}))$$

令$h_\theta(x^{(i)})$=z

$$-\log z$$

“损失函数”（Loss function）



$$-\log z$$

画出图像并验证：

1、它的输出是一个可以表示损失大小的数值；
2、此时的正确答案是1，模型的预测值接近1，此函数的输出越接近于0；
3、反之亦然。

如仍懵逼请记住：

1、损失函数的本质是：以训练集为依据，度量目前模型与“最优解”之间的差距（即“损失”）。
3、机器学习的直接目标，就是通过尝试改变输入参数θ的取值组合来使损失函数的输出最小，用公式可以表示为：$\min_{\theta} J(\theta)$

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

如果模型是胖子
损失函数就是秤

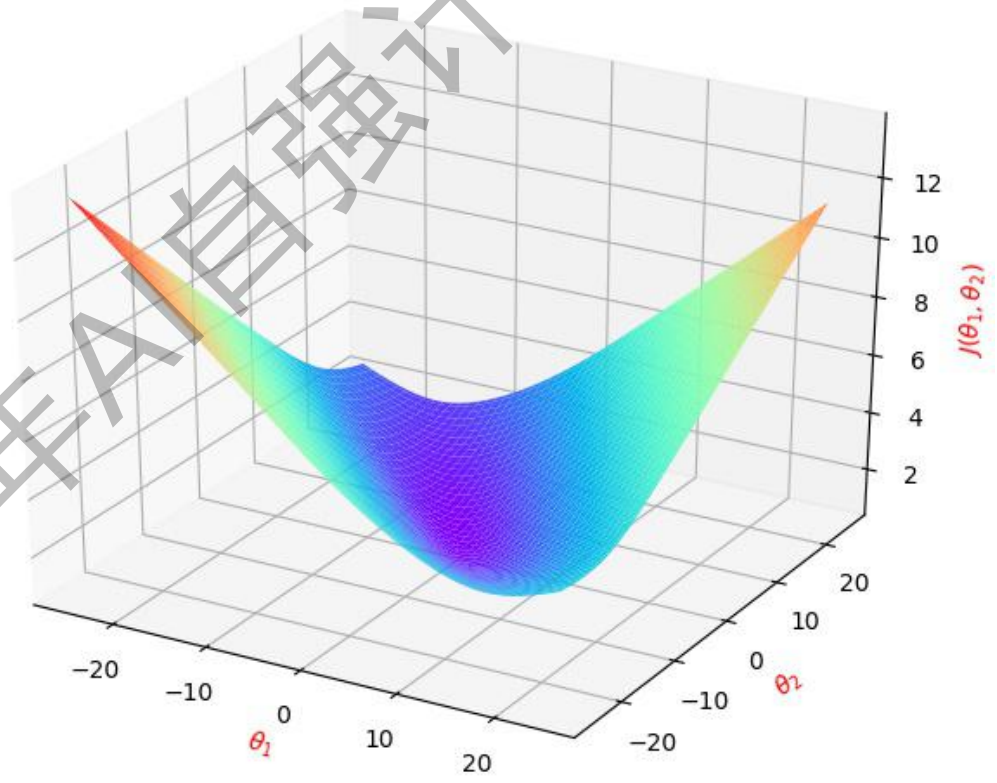
如何更新参数

$$\theta^T X = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

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

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

想要减少损失
不如可视化看看



为了能画三维图像，舍弃一个参数θ_0

“梯度下降法”（Gradient Descent）

新的问题：
损失看起来像山
那计算机怎么下山？

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

5号法宝：梯度下降法

公式好短好高兴

还是不大懂

$\theta_j = \theta_j -$ 

小白眼里

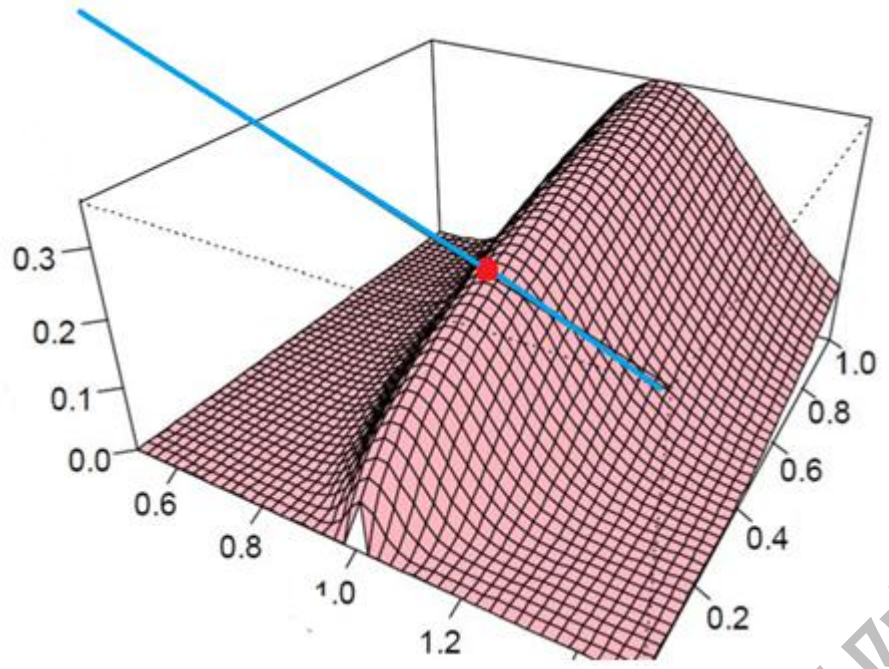
只看关键部分

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

一个提前选定的常数：α

一个偏导数：$\frac{\partial J(\theta)}{\partial \theta_j}$

“梯度下降法”（Gradient Descent）



$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

求导的集合意义：找到下降方向

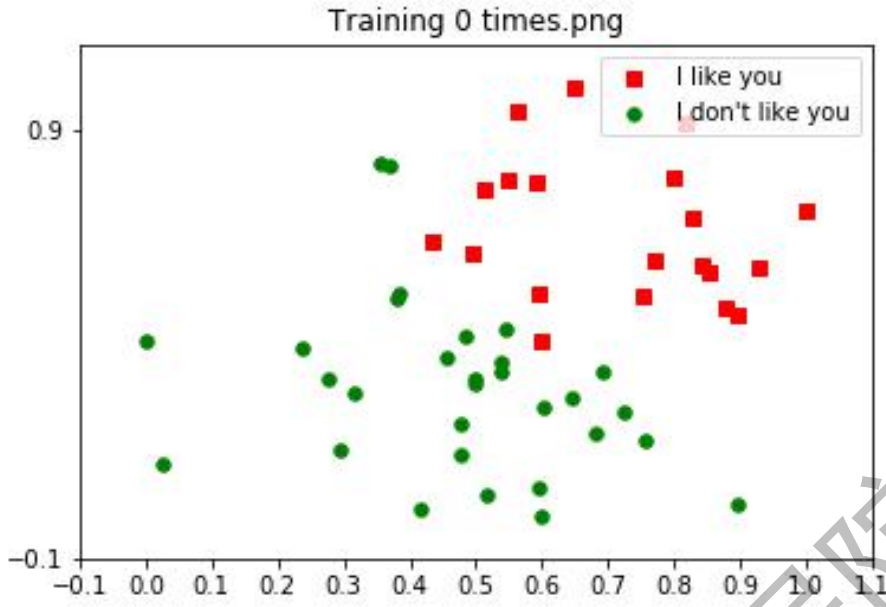
一件小事：求偏导公式

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

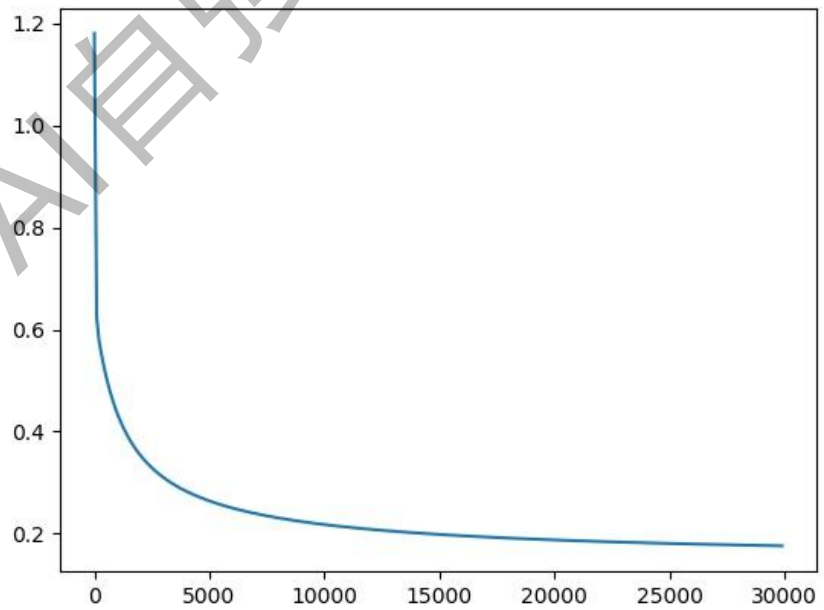
梯度下降法小结

1、算法给了模型一个“水平仪”来指出向下的方向；
2、同时给了我们一个遥控器α，来控制模型每走一步（更新一次参数）的大小。
3、α有一个好听的名字“学习率”（learning rate），它在一定程度上决定了模型“学习”最优解的快慢
4、不能让计算机无限次的走下去，所以还需要指定一个“迭代次数”（iterations），即指定“走多少步”（更新多少次参数）后停下来。；

“梯度下降法”（Gradient Descent）



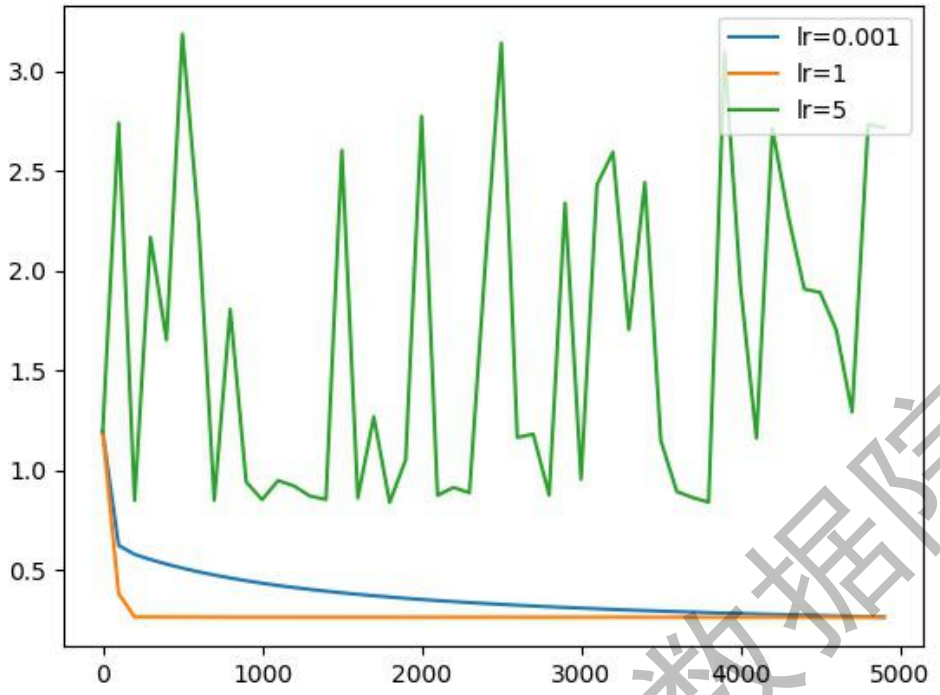
将下山法宝交给计算机
并选择完美的“学习率”和“迭代次数”



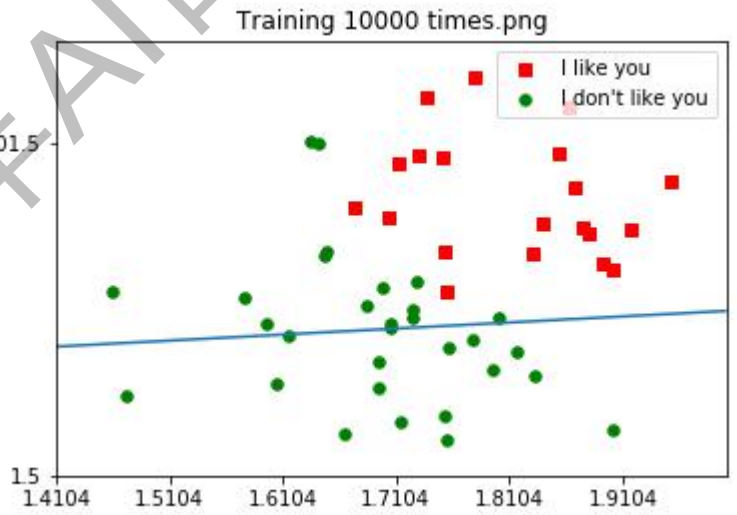
损失函数随着迭代次数的增加而趋于收敛

“梯度下降法”（Gradient Descent）

新的问题：
如果lr选择不完美怎么办？
做一个小实验来验证

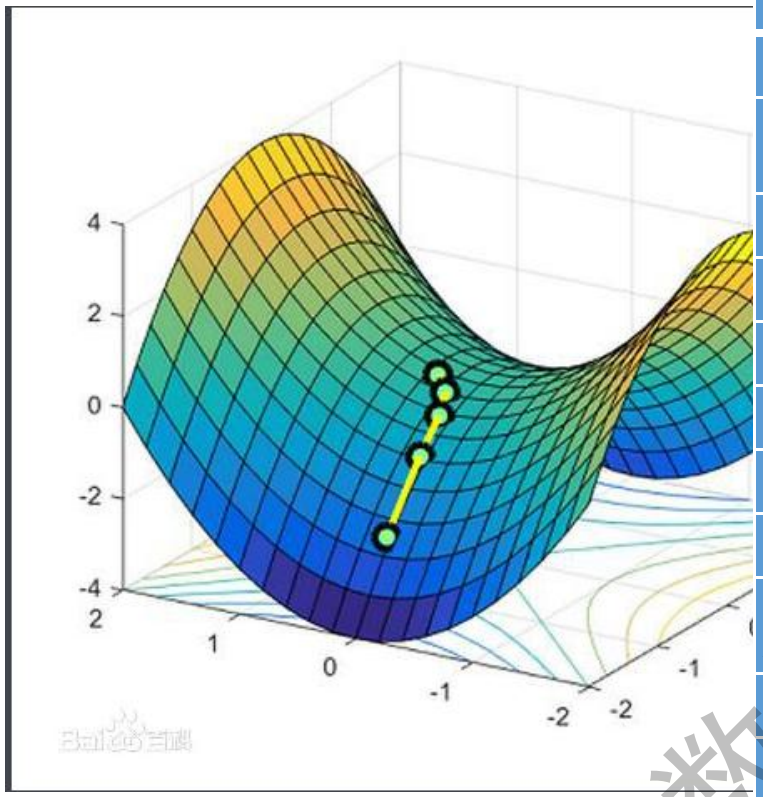


不同lr选择下的loss图像



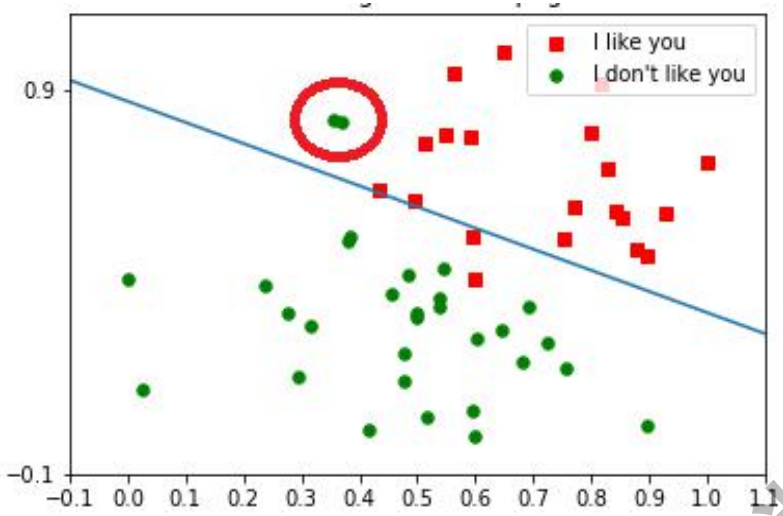
loss“等高线”的另一个发现
归一化的意义在于使loss更快的收敛

“内容延展”（saddle point）



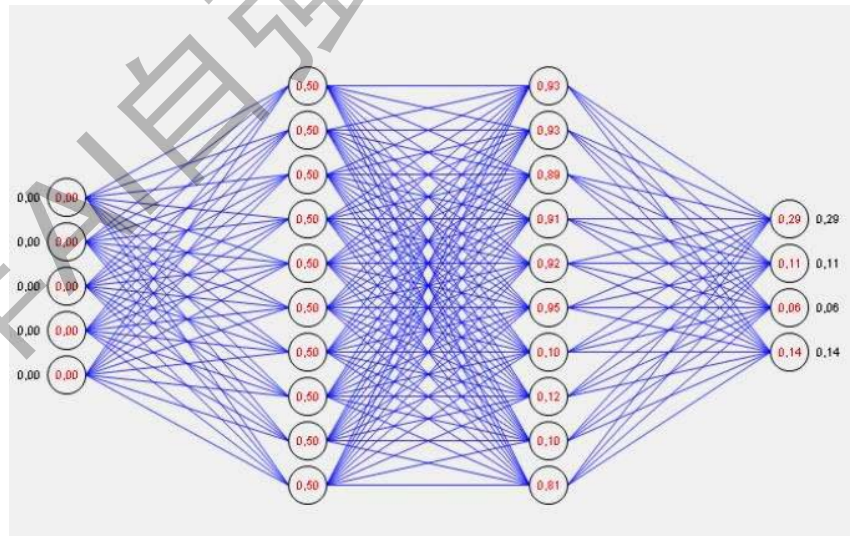
编号	论文	被引用量	发表年份	作者
1	Minimizing Nonconvex Population Risk from Rough Empirical Risk 链接：http://cn.arxiv.org/abs/1803.09357	0	2018	MI Jordan
2	No Spurious Local Minima in Nonconvex Low Rank Problems: A Unified Geometric Analysis 链接：http://cn.arxiv.org/abs/1704.00708	42	2017	Rong Ge
3	Stochastic Cubic Regularization for Fast Nonconvex Optimization 链接：http://cn.arxiv.org/abs/1711.02838	10	2017	MI Jordan
4	How to Escape Saddle Points Efficiently 链接：http://cn.arxiv.org/abs/1703.00887	68	2017	MI Jordan
5	Gradient Descent Can Take Exponential Time to Escape Saddle Point 链接：http://cn.arxiv.org/abs/1705.10412	13	2017	MI Jordan
6	First-order Methods Almost Always Avoid Saddle Points 链接：http://cn.arxiv.org/abs/1710.07406	13	2017	MI Jordan
7	Accelerated Gradient Descent Escapes Saddle Points Faster than Gradient Descent 链接：http://cn.arxiv.org/abs/1711.10456	17	2017	MI Jordan
8	Gradient Descent Converges to Minimizers 链接：https://lanl.arxiv.org/abs/1602.04915	51	2016	MI Jordan
9	Local Maxima in the Likelihood of Gaussian Mixture Models: Structural Results and Algorithmic Consequences 链接：http://cn.arxiv.org/abs/1609.00978	15	2016	Chi Jin
10	Escaping From Saddle Points – Online Stochastic Gradient for Tensor Decomposition 链接：http://cn.arxiv.org/abs/1503.02101	157	2016	Rong Ge
11	Efficient approaches for escaping higher order saddle points in non-convex optimization 链接：http://cn.arxiv.org/abs/1602.05908	20	2016	Rong Ge
12	Structured Prediction, Dual Extragradient and Bregman Projections 链接：http://jmlr.csail.mit.edu/papers/volume7/taskar06a/taskar06a.pdf	104	2006	MI Jordan
13	Structured Prediction via the Extragradient Method 链接：http://papers.nips.cc/paper/2794-structured-prediction-via-the-extragradient-method.pdf	61	2005	MI Jordan
14	Deep Learning without Poor Local Minima 链接：http://cn.arxiv.org/abs/1605.07110	117	2016	Kenji Kawaguchi

欠拟合（underfitting）



红圈处“身高残疾”“收入坚挺”的两位小哥哥未能被正确拟合

解决欠拟合的方法

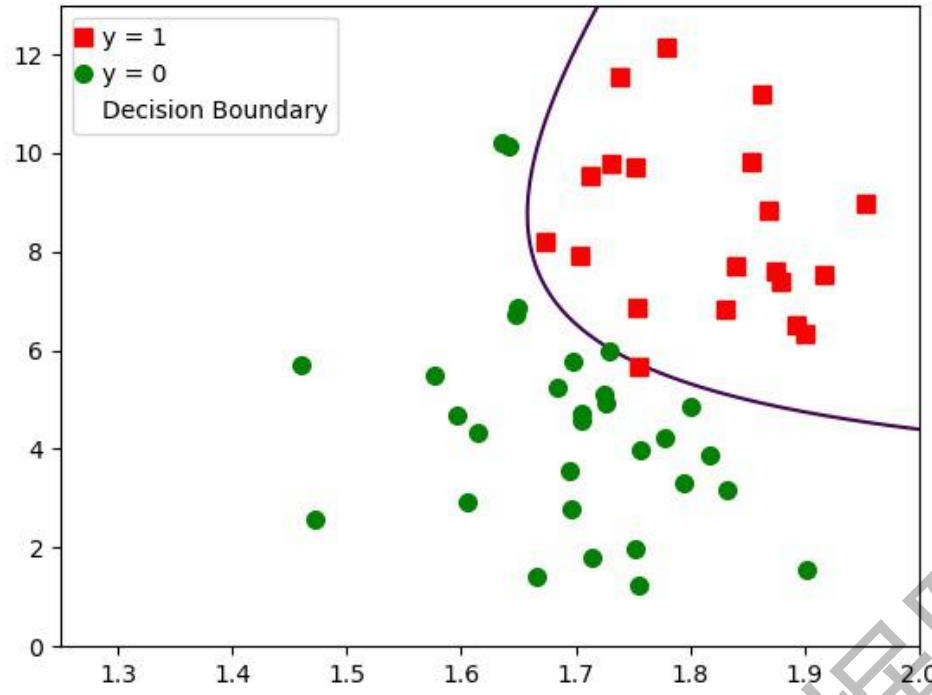


拟合能力更强的模型

$$\theta^T X = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1 x_2 + \theta_5 x_2^2$$

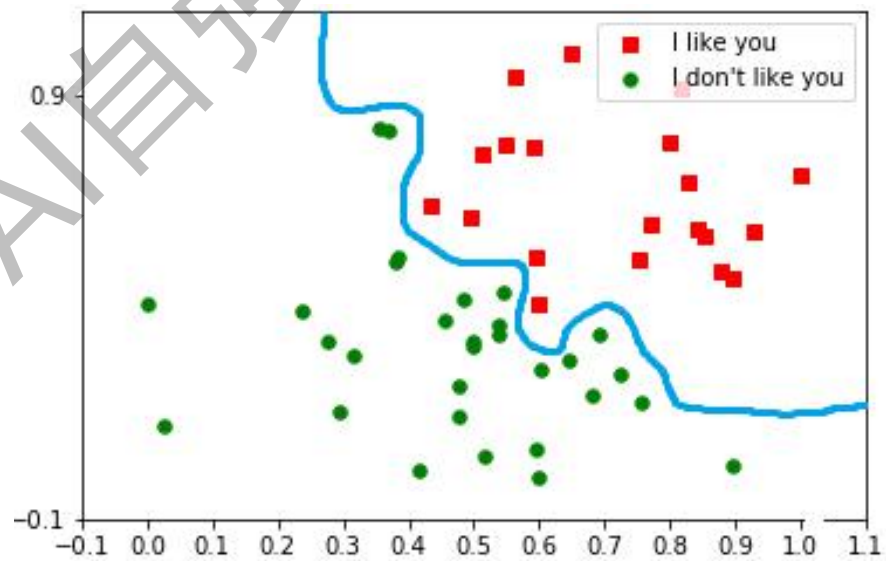
更多的特征（引入2次项）

过拟合(overfitting)&泛化(generalization)



引入2次项后的决策边界

引入更高次项？



泛化能力变差

1、模型被训练出来就是为了做泛化预测，所以过拟合将伤害模型精度；
2、在一定范围内，过拟合越严重，泛化能力越差；
3、解决过拟合有很多办法，将在后续课程介绍。

新的问题：
过拟合怎么解决？

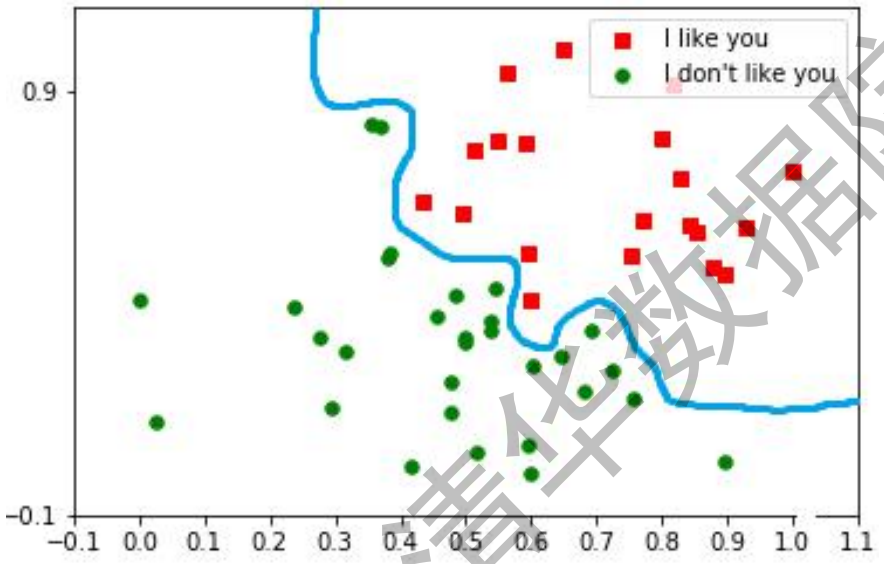
正则化
Regularization

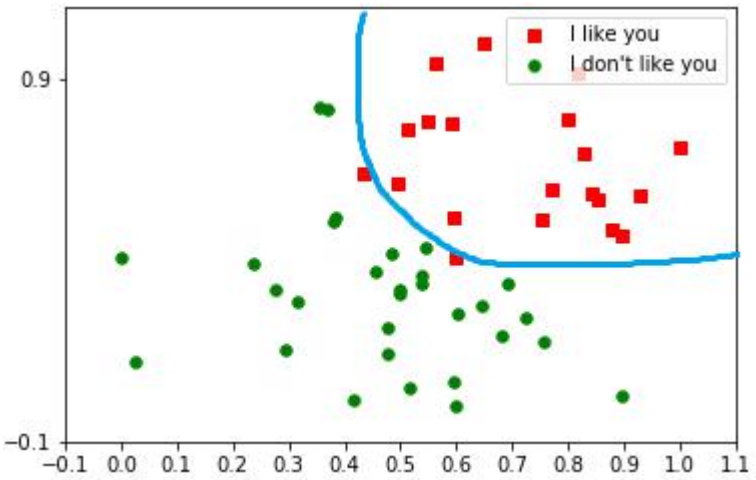
6号法宝：正则化公式

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

$$\frac{\partial J(\theta)}{\partial \theta_j} = (\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}) + \frac{\lambda}{m}\theta_j$$

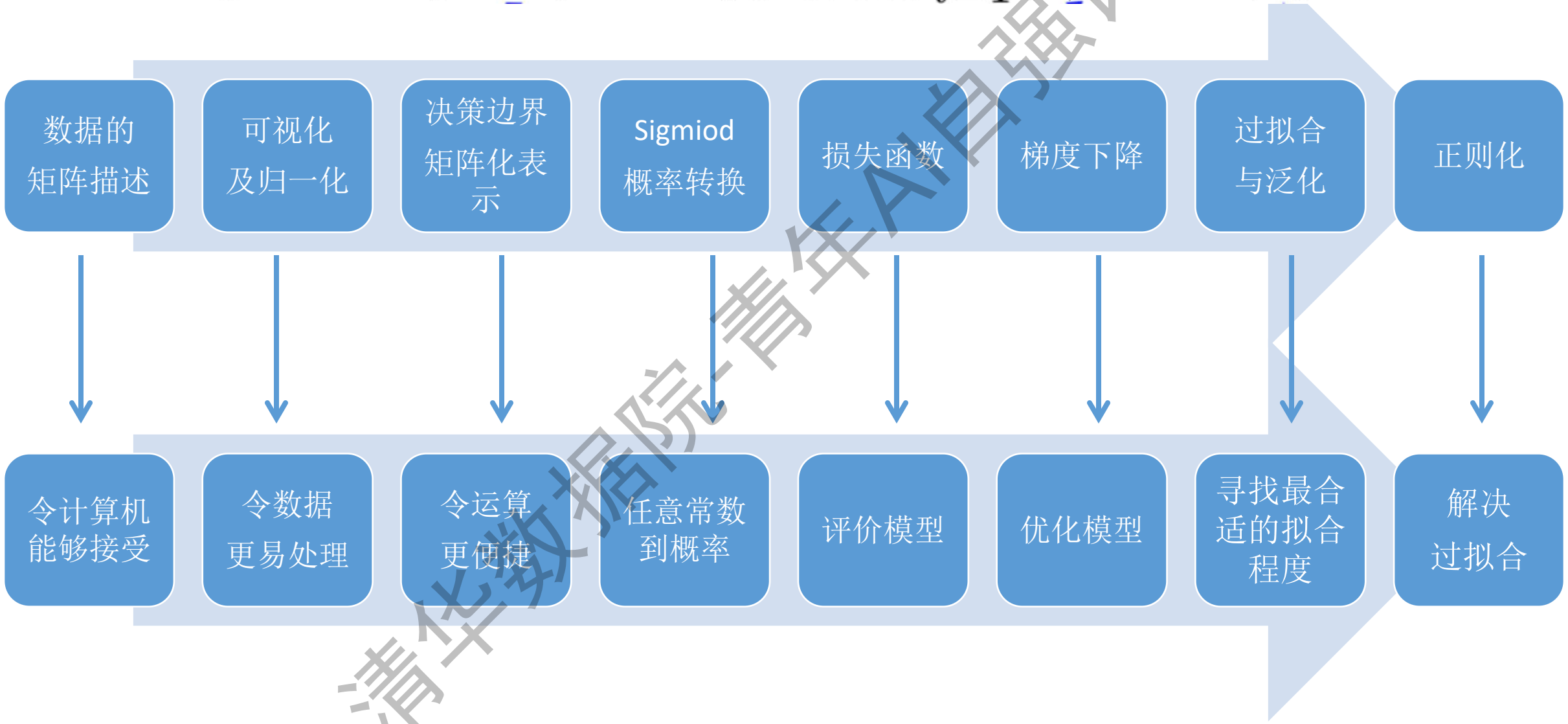
λ的本质是惩罚θ
即λ的取值越大，θ的取值越小





总结

$$\arg\min F(\mathcal{D};\theta) = \mathcal{L}\left(\{x_i, y_i\}_{i=1}^{n};\theta\right) + \Omega(\theta)$$



大数据

关于作业的一点小事

不迷路，不走丢

扫码加好友进群

关注直播间公告