中国科学技术大学
University of Science and Technology of China

《人工智能数学原理与算法》
第2章：机器学习基础

2.5 逻辑回归

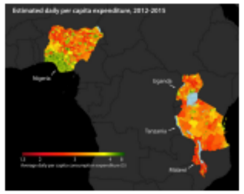
王翔

xiangwang@ustc.edu.cn

回顾：机器学习中的回归问题（Regression）

$$x \longrightarrow \boxed{f} \longrightarrow y \in \mathbb{R}$$

响应（response）



贫困地图:　卫星图像 $\longrightarrow$ 资产财富指数

房价估计:　房屋信息（位置，面积）$\longrightarrow$ 房价

到达时间:　目的地，天气，时间 $\longrightarrow$ 到达时间

回顾：机器学习中的分类问题（Classification）

分类器（classifier）

$$x \longrightarrow \boxed{f} \longrightarrow y \in \{+1, -1\}$$

标签（label）

欺诈检测: 信用卡交易信息 → 是否欺诈

评论检测: 评论信息 → 是否有害

粒子对撞: 测量到的粒子对撞信息 → 粒子衰变还是背景噪音

问：分类和回归之间的关键区别是什么？

- 分类有**离散的输出**
- 回归有**连续的输出**

扩展：多分类问题 $y \in \{1, \ldots, K\}$

回顾：线性回归（Linear Regression）

模型向量表示： $f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x)$　　$\mathbf{w} = [w_1, w_2]$　　$\phi(x) = [1, x]$

参数向量/模型参数　特征提取器　特征向量

假设类： $\mathcal{F} = \{f_{\mathbf{w}} : \mathbf{w} \in \mathbb{R}^2\}$ （预测器 f 的集合）

损失函数： $\text{Loss}(x, y, \mathbf{w}) = (f_{\mathbf{w}}(x) - y)^2$ 平方损失（squared loss）

$$\text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y)\in\mathcal{D}_{\text{train}}} \text{Loss}(x, y, \mathbf{w})$$

均方误差（MSE, mean squared error）

$$\arg\max_{\mathbf{w}} \ln p(\mathbf{y}|\mathbf{x}, \mathbf{w}) = \arg\min_{\mathbf{w}} \sum_{(x,y)\in\mathcal{D}} (y - \phi(x)\mathbf{w})^2$$

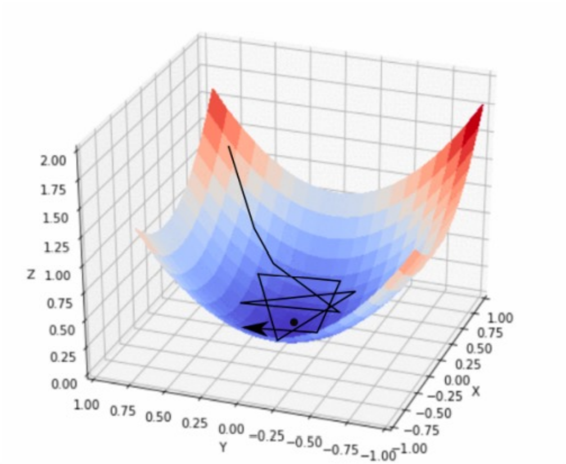
高斯假设下的最大似然估计 = 最小化平方误差

回顾：线性回归（Linear Regression）

优化算法:　$\min_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})$

Definition: gradient

The gradient $\nabla_{\mathbf{w}}\text{TrainLoss}(\mathbf{w})$ is the direction that increases the training loss the most.



Algorithm: gradient descent

Initialize $\mathbf{w} = [0, \ldots, 0]$

For $t = 1, \ldots, T$: epochs

$$\mathbf{w} \leftarrow \mathbf{w} - \underbrace{\eta}_{\text{step size}} \underbrace{\nabla_{\mathbf{w}}\text{TrainLoss}(\mathbf{w})}_{\text{gradient}}$$

目录

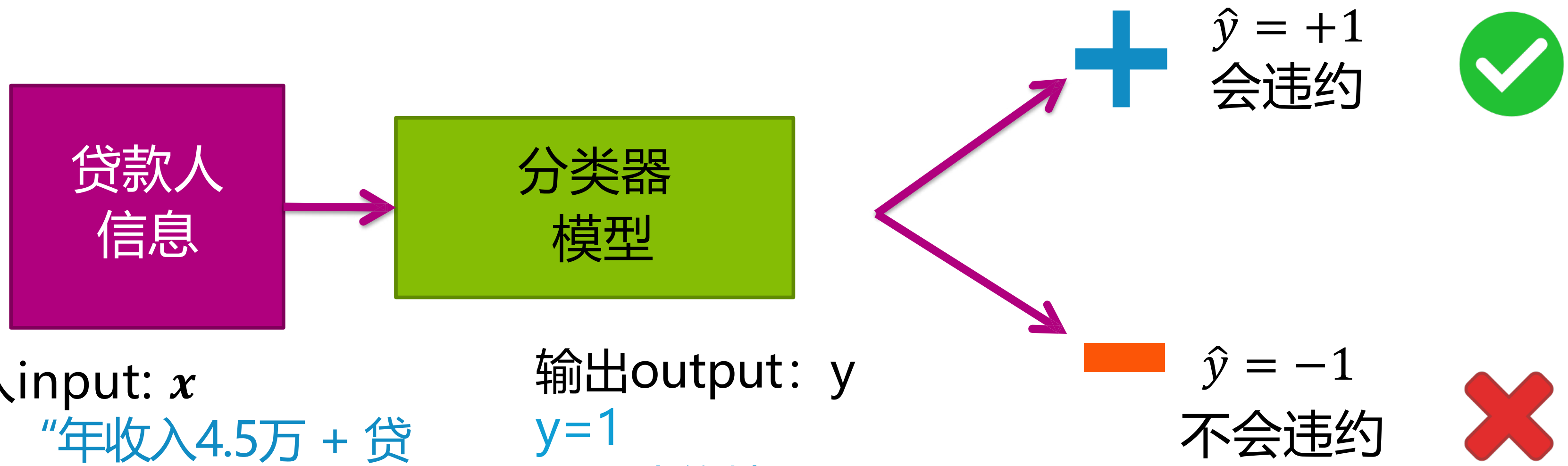
目录

课程安排

假设类	+	损失函数	+	优化算法
↓		↓		↓
特征简单加权和		最大似然估计		梯度上升
↓		↓		↓
逻辑回归		对数似然估计		随机梯度下降

二分类问题：贷款违约检测

模型预测结果



输入input: $\boldsymbol{x}$

如：“年收入4.5万 + 贷款4万，利率13.67%".

输出output：y

y=1

个人违约情况：，违约”

分类器：简单线性分类器

特征	系数
…	…

贷款人信息

输入input: $\boldsymbol{x}$

简单线性分类器:

Score(x) = 各特征的加权和

If Score(x) >0:

$\hat{y} = +1$

Else:

$\hat{y} = -1$

分类器：简单线性分类器

□ 一个简单的例子

- 假设每个特征的系数已知

特征	系数
年收入	1.0
贷款金额	-0.8
贷款利率	-3.3
…	…

输入$\boldsymbol{x}$：

贷款人年收入4.5万
贷款金额4万
贷款利率13.76%

$$\text{Score}(x) = 1.0 * 4.5 - 0.8 * 4 - 3.3 * 0.1376 > 0$$

$$\hat{y} = +\mathbf{1}$$

线性分类器：建模

$$\hat{y} = \text{sign}\left(\text{Score}(\boldsymbol{x})\right)$$

$$\text{Score}(\boldsymbol{x}) = w_0 \cdot \phi(\boldsymbol{x})_0 + w_1 \cdot \phi(\boldsymbol{x})_1 + \ \ldots + w_d \cdot \phi(\boldsymbol{x})_d$$

$$= \sum_i w_i \cdot \phi(\boldsymbol{x})_i = \boldsymbol{w}^\intercal \boldsymbol{\phi}(\boldsymbol{x})$$

特征提取器$\boldsymbol{\phi}$：

特征1 = $\phi(x)_0$ (e.g., 1)

特征2 = $\phi(x)_1$ (e.g., $x[1]$=年收入)

特征3 = $\phi(x)_2$ (e.g., $x[2]$=贷款金额

or $\log(x[2])$

or $\log(x[2]/x[1])$)

特征d = $\phi(x)_d$ (其他关于$\boldsymbol{x}$的函数)

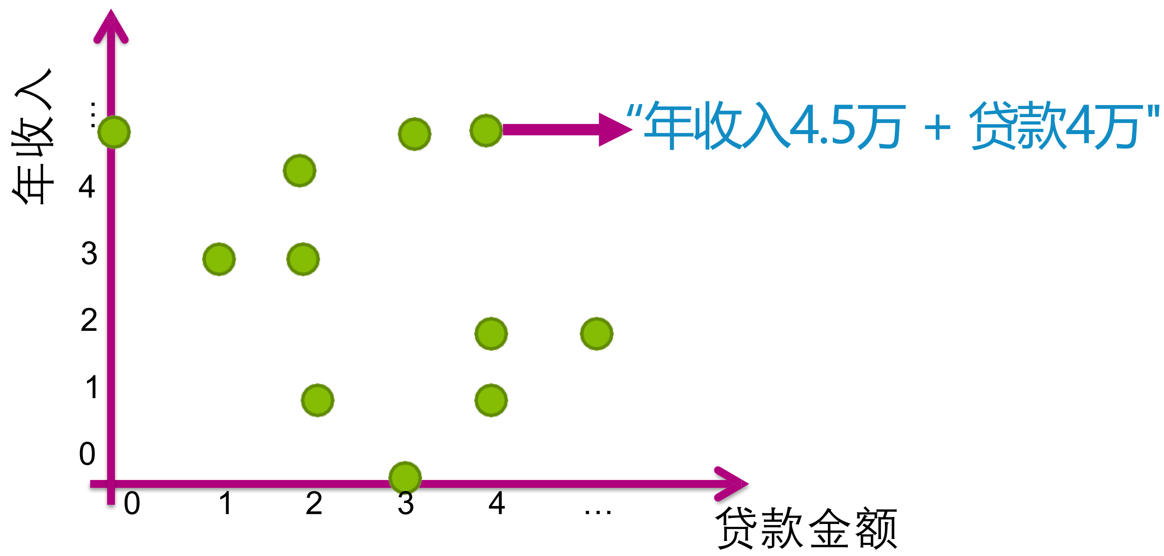
决策边界

- 假设只有两个特征具有非零系数

特征	系数	值
	w_0	0.0
贷款金额	w_1	1.0
年收入	w_2	-1.5

$$\text{Score}(\boldsymbol{x}) = 1.0 * \text{贷款金额} - 1.5 * \text{年收入}$$

贷款人ID	贷款金额	年收入
1	4.0	4.5
2	0.0	3.0
3	2.0	1.0
4	3.0	5.3
…	…	…

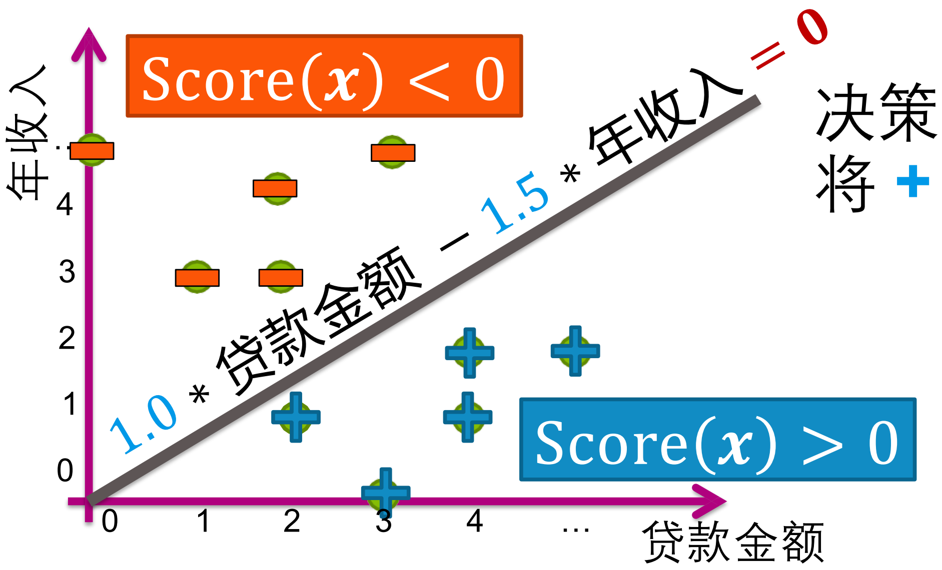


决策边界

- 假设只有两个特征具有非零系数

特征	系数	值
	w_0	0.0
贷款金额	w_1	1.0
年收入	w_2	-1.5

$\text{Score}(\boldsymbol{x}) = 1.0 * \text{贷款金额} - 1.5 * \text{年收入}$



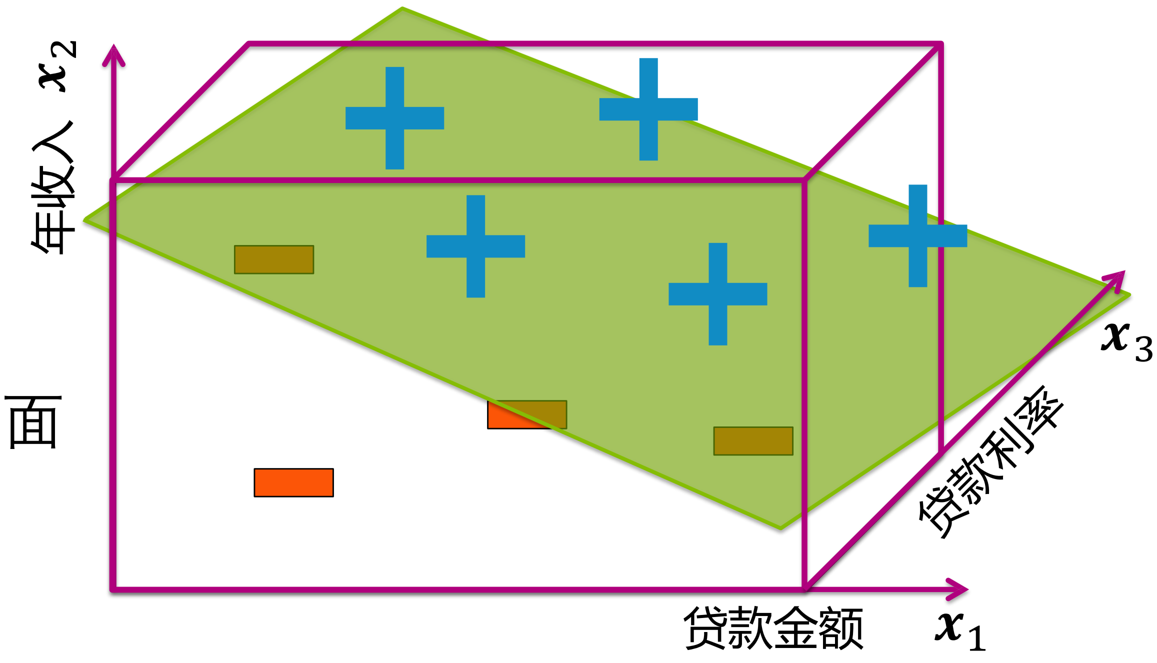
决策边界:
将 + 和 - 区分开的线

决策边界

- 对于更多线性特征

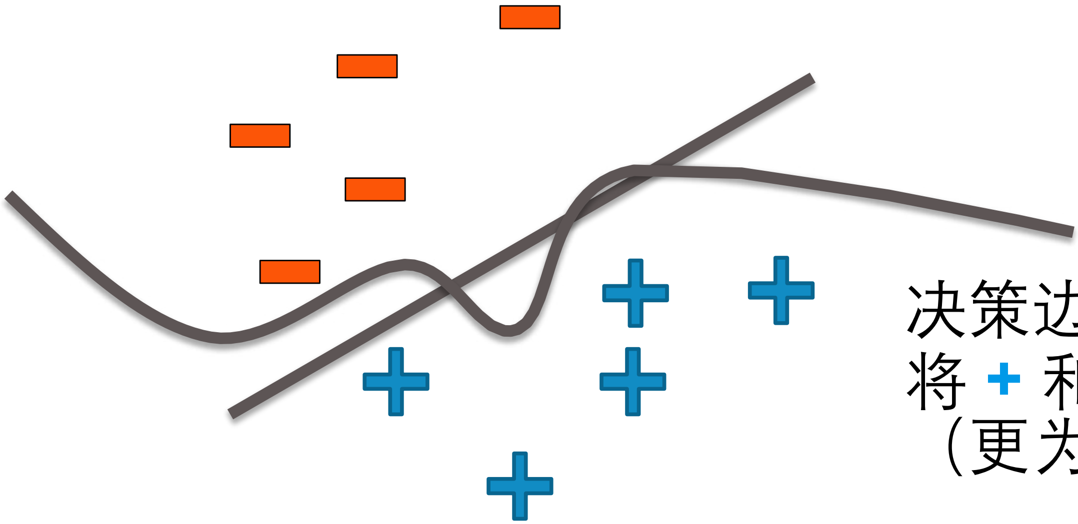
$$\text{Score}(\boldsymbol{x}) = w_0 + w_1 * \text{贷款金额} + w_2 * \text{年收入} + w_3 * \text{贷款利率}$$

决策边界:
将 + 和 - 区分开的面



决策边界

- 对于更一般的特征...



决策边界:
将 + 和 - 区分开的曲面
（更为复杂的形状）

目录

对预测有多确定呢?

- **一个简单分类器可以获得+1或-1的预测**
  - 但对于这个预测到底有多确定呢?

“年收入4.5万 + 贷款4万，利率13.67%"

↓

很确定： **+1**

“年收入7.5万 + 贷款2.8万，利率20%”

↓

不确定

在分类中使用概率

□ 在分类中使用概率

$\boldsymbol{x}$ = “年收入4.5万 + 贷款4万， 利率13.67%"

很确定： +1

$P(y = +1|\boldsymbol{x}) = 0.99$

$\boldsymbol{x}$ =“年收入7.5万 + 贷款2.8万， 利率20%”

不确定

$P(y = +1|\boldsymbol{x}) = 0.55$

许多分类器提供了一定程度的确定性：

输出类别 → $P(y|\boldsymbol{x})$ ← 输入特征

目标：从训练数据中学习条件概率

- **训练数据：** N个 **观测值** $\{(x, y)\}$

贷款金额	年收入	y = 是否违约
317.96k	635k	-1
866.1k	305k	+1
136.08k	45k	+1
95.21k	100k	-1
…	…	…

在训练数据上学习
（拟合训练数据分布）

通过寻找最优参数$\hat{w}$，
确定最佳模型$\hat{P}$，
可用来预测$\hat{y}$

使用概率估计做预测

□ 使用概率估计做预测可以提升可解释性：

- 预测 $\hat{y}$，并提供有多确定

贷款人信息

输入input: $\boldsymbol{x}$

使用概率估计做预测：

$\hat{P}(\mathrm{y}|\mathrm{x})$ = 对类别概率的估计

If $\hat{P}(\mathrm{y}=+1|\mathrm{x}) > 0.5$:

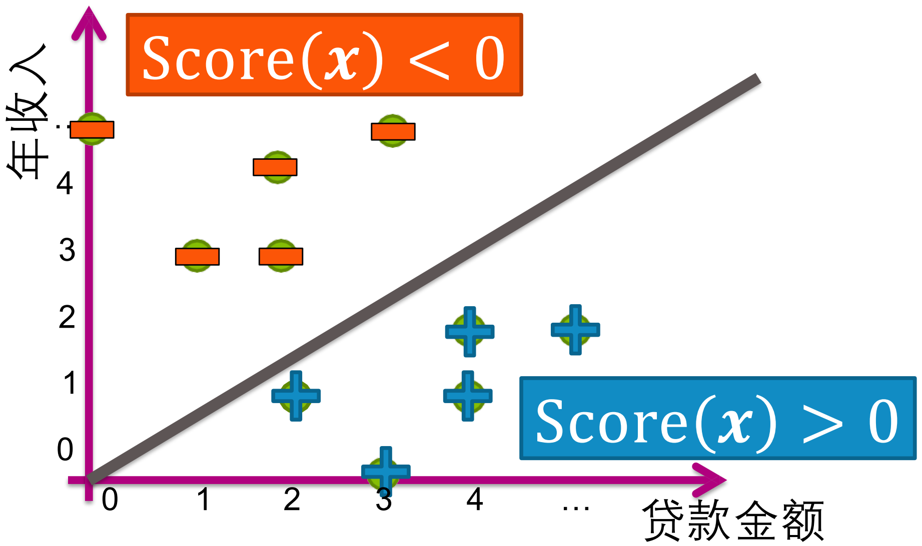
$\quad\quad \hat{y} = +\mathbf{1}$

Else:

$\quad\quad \hat{y} = -\mathbf{1}$

决策边界

$$\text{Score}(\boldsymbol{x}) = w_0 \cdot \phi(\boldsymbol{x})_0 + \ldots + w_d \cdot \phi(\boldsymbol{x})_d$$
$$= \sum_i w_i \cdot \phi(\boldsymbol{x})_i = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x})$$



如何将 $\text{Score}(\boldsymbol{x})$与 $\hat{P}(y|x)$关联起来呢?

如何理解Score(x)?

$$\text{Score}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x})$$

$-\infty$ | 0.0 | $+\infty$

$\hat{y} = -\mathbf{1}$ | $\hat{y} = +\mathbf{1}$

非常确定	不太确定	非常确定
$\hat{y} = -\mathbf{1}$	$\hat{y} = -\mathbf{1}$或$+\mathbf{1}$	$\hat{y} = +\mathbf{1}$
$\hat{P}(\mathrm{y} = +1 \mid \mathrm{x}) = 0$	$\hat{P}(\mathrm{y} = +1 \mid \mathrm{x}) = 0.5$	$\hat{P}(\mathrm{y} = +1 \mid \mathrm{x}) = 1$

什么不直接使用回归来构建分类器?

$-\infty < \text{Score}(\boldsymbol{x}) < +\infty$

$$\text{Score}(\boldsymbol{x}) = w_0 \cdot \phi(\boldsymbol{x})_0 + \ldots + w_d \cdot \phi(\boldsymbol{x})_d = \boldsymbol{w}^\intercal \boldsymbol{\phi}(\boldsymbol{x})$$

-∞　　　0.0　　　+∞

我们如何将
-∞,+∞
映射到 0, 1?

0.0　　　0.5　　　1.0

$\hat{P}(\text{y} = +1|\text{x})$

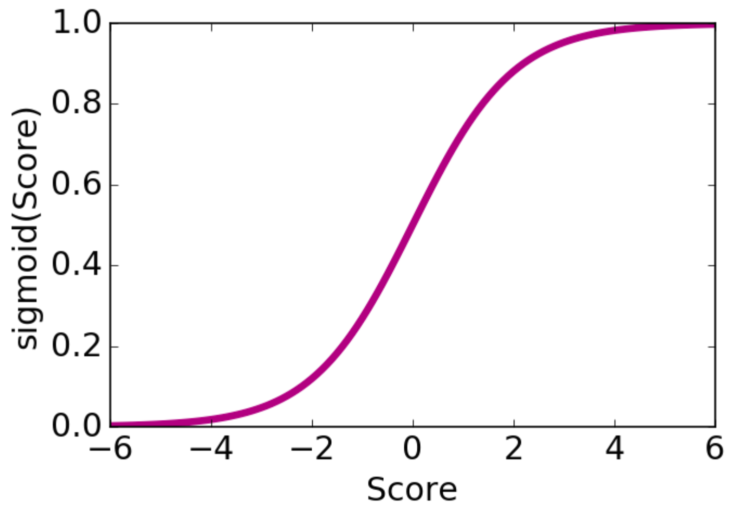
但概率处于 0 到 1 之间

逻辑回归：逻辑函数

□ 逻辑函数 (Logistic，也称sigmoid, logit)

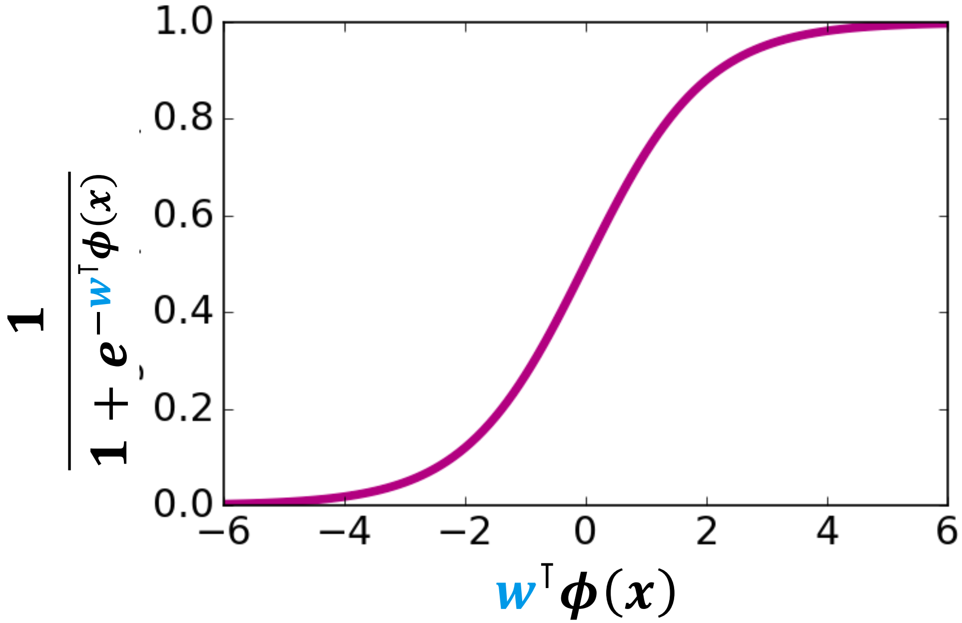
$$\boldsymbol{sigmoid}(\textbf{Score}) = \frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{e}^{-\textbf{Score}}}$$

Score	-∞	-2	0.0	+2	+∞
Sigmoid (Score)	0	0.12	0.5	0.88	1



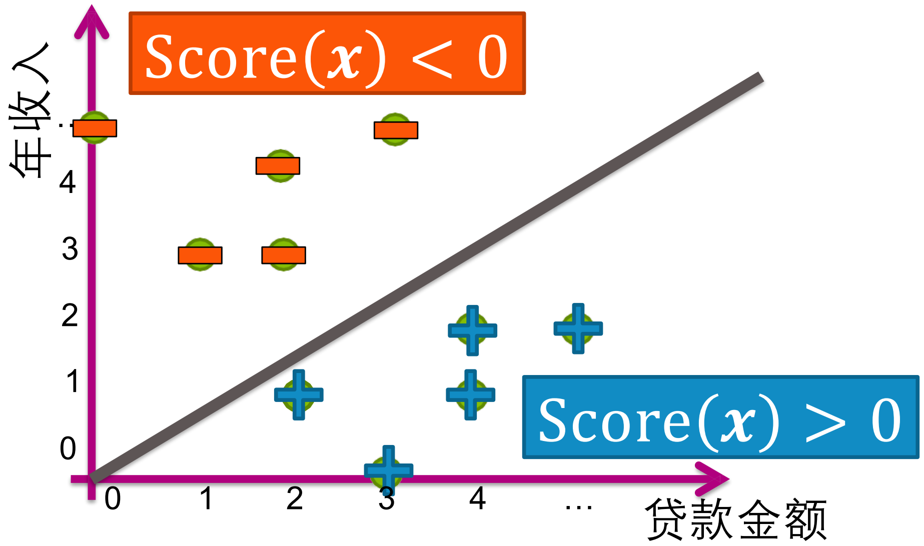
逻辑回归

$$\hat{P}(\mathrm{y}=+1|\mathrm{x}) = \mathrm{sigmoid}\big(\mathrm{Score}(\boldsymbol{x})\big) = \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{e}^{-\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x})}}$$

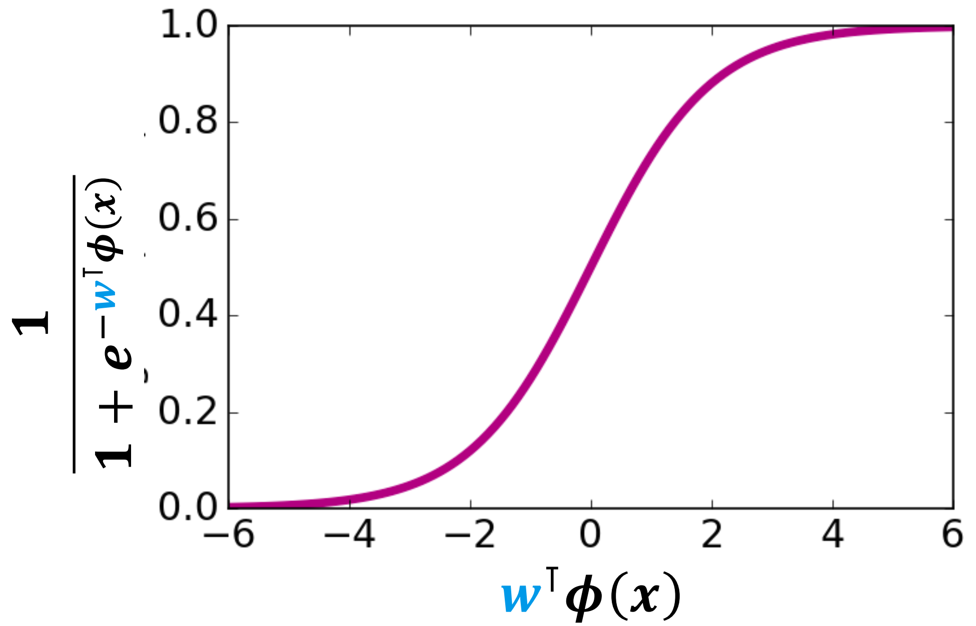


Score(x)	P(y=+1|x,w)
0	0.5
-2	0.12
2	0.88
4	0.98

逻辑回归：线性决策边界



Score(x) < 0
Score(x) > 0
年收入
贷款金额
0
1
2
3
4
…



$\frac{1}{1+e^{-w^{\top}\phi(x)}}$
$w^{\top}\phi(x)$
1.0
0.8
0.6
0.4
0.2
0.0
−6
−4
−2
0
2
4
6

逻辑回归：模型参数的影响

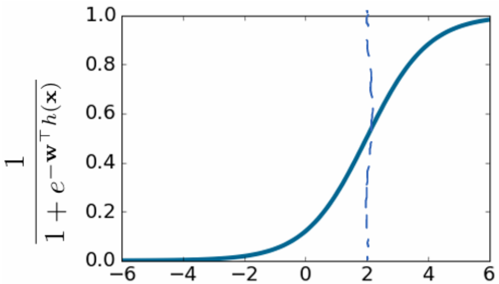
□ 逻辑回归模型系数的影响

$$\text{Score}(\boldsymbol{x}) = w_0 + w_1 * 贷款金额 + w_2 * 年收入 = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x})$$

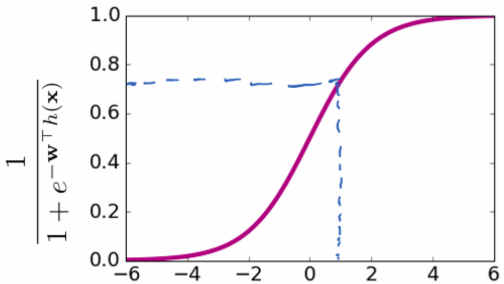
w_0	−2
w_1	+1
w_2	−1

w_0	0
w_1	+1
w_2	−1

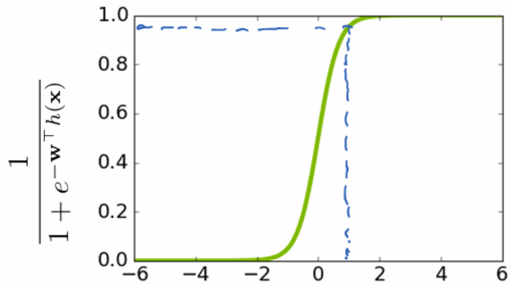
w_0	0
w_1	+3
w_2	−3



#贷款金额 - #年收入



#贷款金额 - #年收入



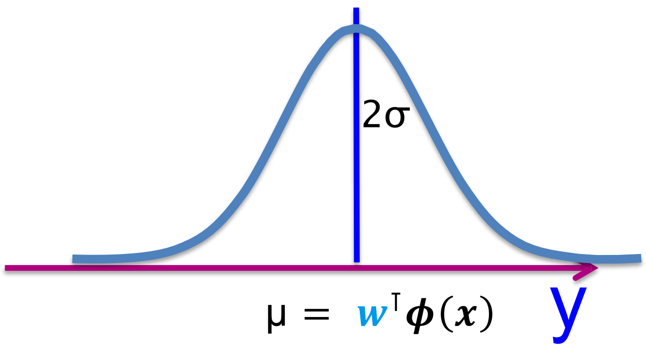
#贷款金额 - #年收入

逻辑回归：与回归模型的不同

- 具有高斯误差的线性回归

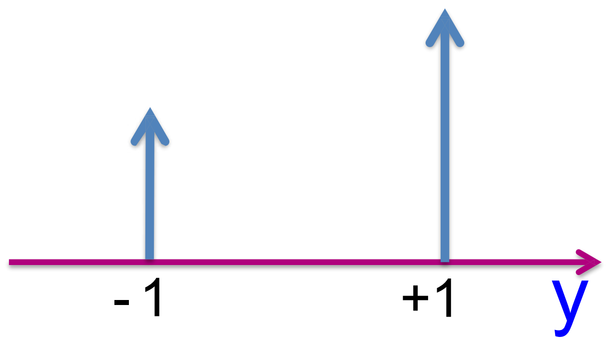
$$y = \boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}) + \epsilon,\ \epsilon \sim N(0, \sigma^2)$$

$$\Rightarrow \mathrm{p}(y|\boldsymbol{x}, \boldsymbol{w}) = N(y;\ \boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}), \sigma^2)$$



- 逻辑回归

$$P(y|x,w) = \begin{cases} \dfrac{1}{1 + e^{-\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x})}} & y = +1 \\ \dfrac{e^{-\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x})}}{1 + e^{-\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x})}} & y = -1 \end{cases}$$

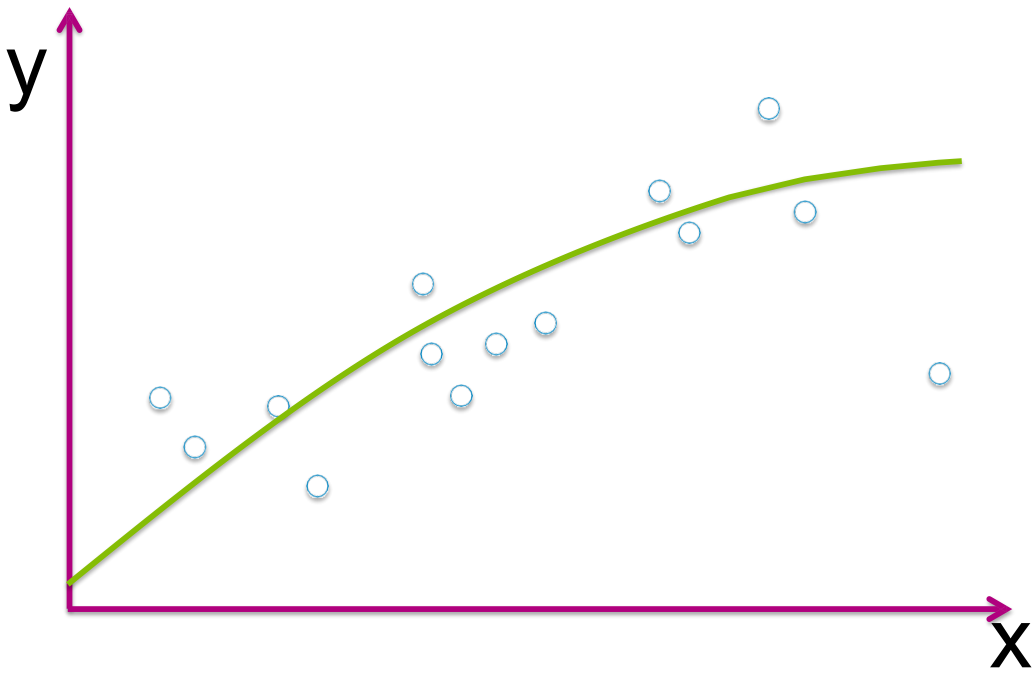


目录

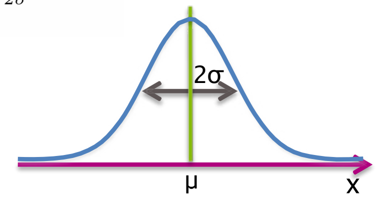
课程安排

假设类	损失函数	优化算法
特征简单加权和	最大似然估计	梯度上升
逻辑回归	对数似然估计	随机梯度下降

回顾：（高斯）线性回归模型



$$\underbrace{N(\mu, \sigma^2)}_{\text{参数}} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$



- ϵ:
  - 基础假设： $\mathrm{E}(\epsilon) = 0$
  - 进一步假设： $\epsilon \sim N(0, \sigma^2)$
- y:
  - $y = w^\top \phi(x) + \epsilon$
  - $y|x, w \sim N(w^\top \phi(x), \sigma^2)$

回顾：最大似然估计

Maximum likelihood estimate w.r.t. w:　最大似然估计

$$\arg\max_{\mathbf{w}} p(\mathbf{y}|\mathbf{x},\mathbf{w}) = \arg\max_{\mathbf{w}} \ln p(\mathbf{y}|\mathbf{x},\mathbf{w})$$

Maximize log-likelihood estimate w.r.t. w:　对数似然

$$\begin{aligned}\arg\max_{\mathbf{w}} \ln p(\mathbf{y}|\mathbf{x},\mathbf{w}) &= \arg\max_{\mathbf{w}} \ln \prod_{(x,y)\in\mathcal{D}} p(y|x,\mathbf{w}) \\ &= \arg\max_{\mathbf{w}} \ln \prod_{(x,y)\in\mathcal{D}} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y-\phi(x)\mathbf{w})^2} \\ &= \arg\min_{\mathbf{w}} \sum_{(x,y)\in\mathcal{D}} (y-\phi(x)\mathbf{w})^2\end{aligned}$$

高斯假设下的最大似然估计 = 最小化平方误差

逻辑回归：需要寻找最佳参数

贷款金额	年收入	y = 是否违约
317.96k	635k	-1
866.1k	305k	+1
136.08k	45k	+1
95.21k	100k	-1
...	...	...

逻辑回归：需要寻找最佳参数

贷款金额	年收入	y = 是否违约
866.1k	305k	+1
136.08k	45k	+1
...	...	...

贷款金额	年收入	y = 是否违约
317.96k	635k	-1
95.21k	100k	-1
...	...	...

$P(y=+1|x,w) = 1.0$

$P(y=-1|x,w) = 1.0$

我们希望 $\hat{w}$ 满足

逻辑回归：利用最大似然估计学习逻辑回归

数据点	贷款金额	年收入	y = 是否违约
x_1, y_1	317.96k	635k	-1
x_2, y_2	866.1k	305k	+1
x_3, y_3	136.08k	45k	+1
x_4, y_4	95.21k	100k	-1
...	...	...	...

$$\ell(\mathrm{w}) = \underbrace{P(y_1|x_1,\mathrm{w}) \quad P(y_2|x_2,\mathrm{w}) \quad P(y_3|x_3,\mathrm{w}) \quad P(y_4|x_4,\mathrm{w})}_{\prod_{i=1}^{N} P(y_i \mid \mathbf{x}_i, \mathbf{w})}$$

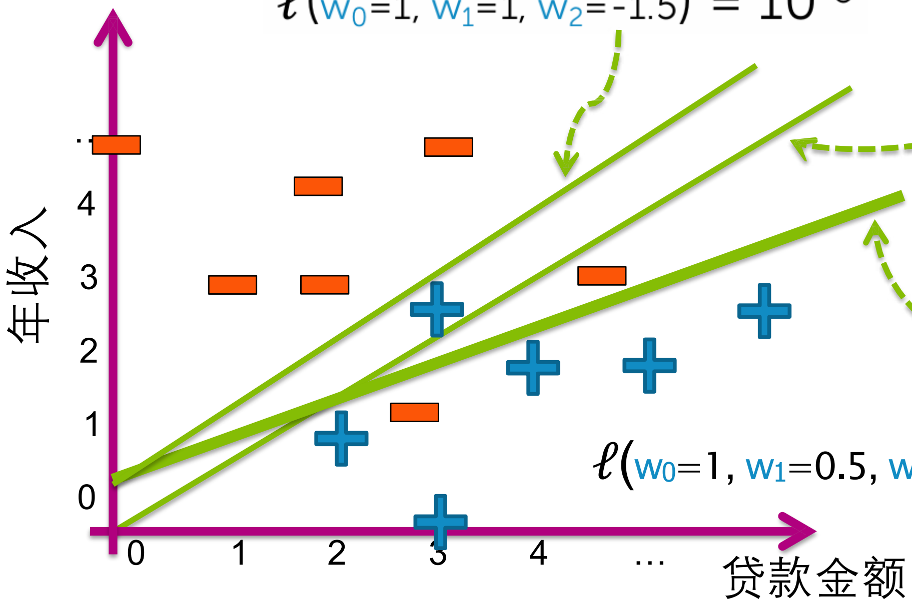
逻辑回归：根据最大似然估计确定“最佳”分类器

对于所有可能的w_0,w_1,w_2，选择似然性最大的

$$\ell(\mathbf{w}) = \prod_{i=1}^{N} P(y_i \mid \mathbf{x}_i, \mathbf{w})$$

$\ell(w_0=0, w_1=1, w_2=-1.5) = 10^{-6}$

$\ell(w_0=1, w_1=1, w_2=-1.5) = 10^{-5}$



$\ell(w_0=1, w_1=0.5, w_2=-1.5) = 10^{-4}$

最佳模型:

最大化似然函数 $\ell(\mathbf{w})$

$\hat{\mathbf{w}} = (w_0=1, w_1=0.5, w_2=-1.5)$

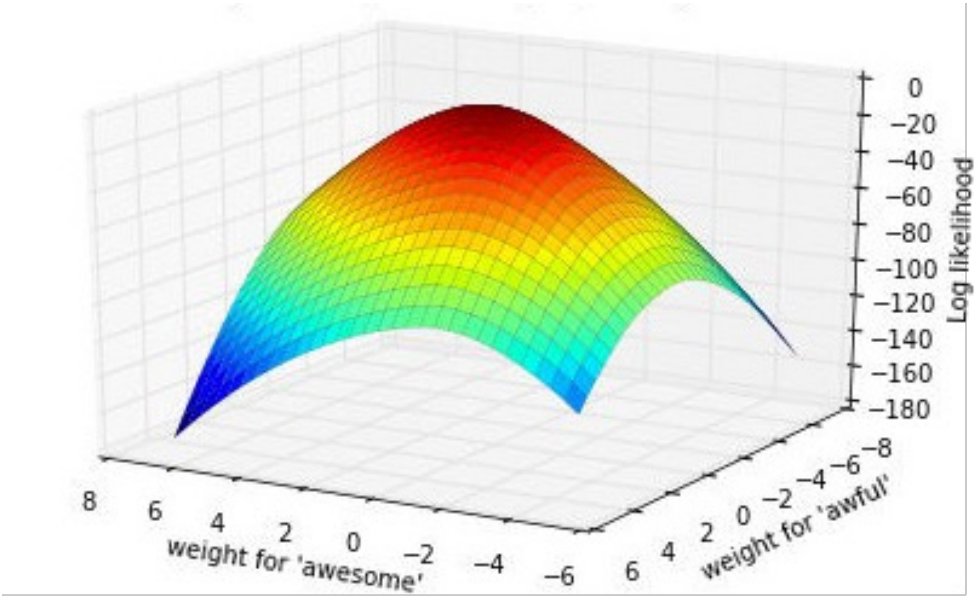
课程安排

假设类 + 损失函数 + 优化算法

假设类	损失函数	优化算法
特征简单加权和	最大似然估计	梯度上升
逻辑回归	对数似然估计	随机梯度下降

逻辑回归：梯度上升优化MLE

□ 最大化似然估计



对于所有可能的w_0,w_1,w_2，最大化函数

$$\max_{w_0,w_1,w_2} \underbrace{\prod_{i=1}^{N} P(y_i \mid \mathbf{x}_i, \mathbf{w})}_{\ell(w_0,w_1,w_2)\text{ 为三变量函数}}$$

逻辑回归：对数似然估计

$$\mathrm{P}(y=+1|x)=\mathrm{sigmoid}\left(\mathrm{Score}(\boldsymbol{x})\right)=\sigma\left(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x})\right)=\frac{1}{1+e^{-\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x})}}$$

对于包含 N 个样本的数据集，即 $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$，似然（likelihood)估计为：

$$\max_{w} l(w)=\prod_{i=1}^{N} P(y_i|\boldsymbol{x}_i, w)$$

重写为对数似然(log-likelihood)估计：

$$\max_{w} ll(w)=\sum_{i=1}^{N} \log(P(y_i|\boldsymbol{x}_i, w))$$

$$ll(w)=\frac{1}{2}\sum_{i=1}^{N}(1+y_i)\cdot\log(P(y_i=+1|\boldsymbol{x}_i, w))+(1-y_i)\cdot\log\left(P(y_i=-1|\boldsymbol{x}_i, w)\right)$$

$$=\frac{1}{2}\sum_{i=1}^{N}(1+y_i)\cdot\log(\sigma\left(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x})\right))+(1-y_i)\cdot\log\left(1-\sigma\left(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x})\right)\right)$$

逻辑回归：逻辑对数似然的梯度

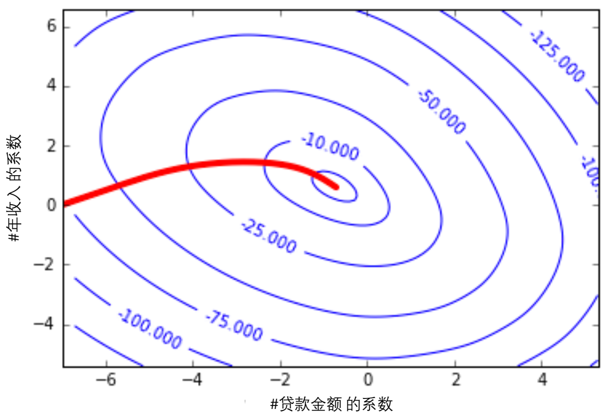
$$\frac{\partial\, ll(w)}{\partial\, w} = \frac{1}{2}\sum_{i=1}^{N} \boldsymbol{\phi}(\boldsymbol{x})\left(1 + y_i - 2\sigma\left(\boldsymbol{w}^{\intercal}\boldsymbol{\phi}(\boldsymbol{x})\right)\right)$$

$$= \sum_{i=1}^{N} \boldsymbol{\phi}(\boldsymbol{x}) \underbrace{\frac{1}{2}\left(1 + y_i - 2P(y_i = +1|\boldsymbol{x}_i, w)\right)}_{\Delta_i}$$

	P(y=+1|xᵢ,w) ≈ 1	P(y=+1|xᵢ,w) ≈ 0
y_i=+1	$\Delta_i \approx 0$, $P(y = +1\|x_i, w)$ 不变	$\Delta_i \approx 1 \rightarrow w$增加 $P(y = +1\|x_i, w)$ 上升
y_i=-1	$\Delta_i \approx -1 \rightarrow w$下降 $P(y = +1\|x_i, w)$ 下降	$\Delta_i \approx 0$, $P(y = +1\|x_i, w)$ 不变

逻辑回归：梯度上升优化MLE

☐ 逻辑回归的梯度上升



init $\mathbf{w}^{(1)}=0$, $t=1$

while $\|\nabla \ell(\mathbf{w}^{(t)})\| > \varepsilon$

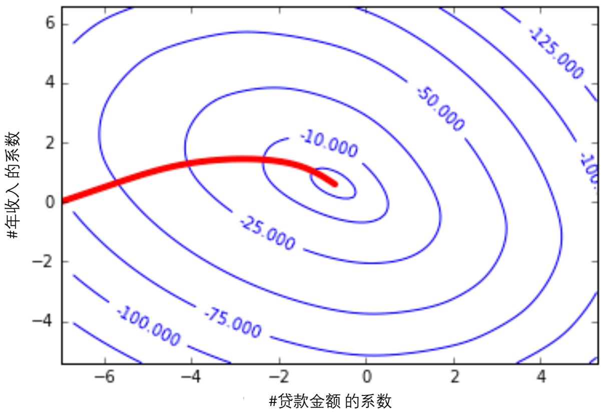
for $j=0,\ldots,D$

$$\text{partial}[j] = \sum_{i=1}^{N} h_j(\mathbf{x}_i)\underbrace{\Big(\mathbb{1}[y_i=+1] - P(y=+1 \mid \mathbf{x}_i, \mathbf{w}^{(t)})\Big)}_{\text{预测与真值的差异}}$$

$w_j^{(t+1)} \leftarrow w_j^{(t)} + \eta\ \text{partial}[j]$

$t \leftarrow t+1$

逻辑回归：梯度上升优化MLE



初始化 $\boldsymbol{w}^{(1)} = \mathbf{0}$ (或者随机初始化)

初始化 t = 1

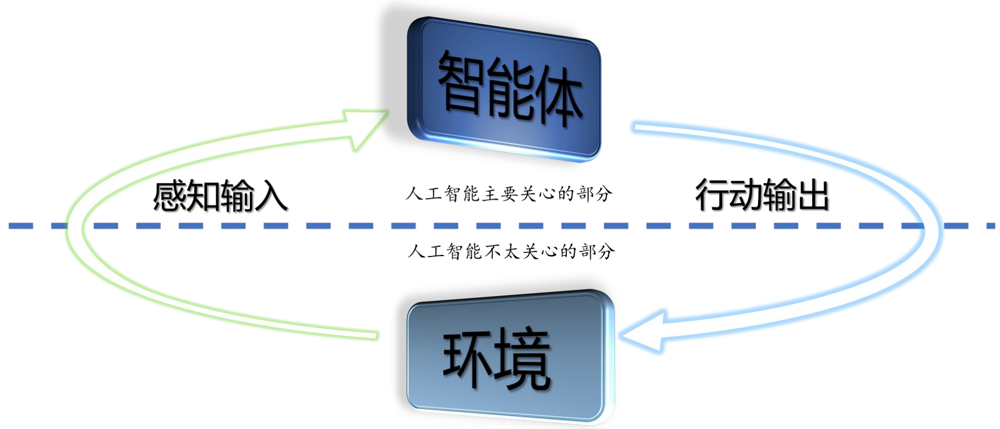
While $\left\|\nabla l(\boldsymbol{w}^{(t)})\right\| > \epsilon$:

$$\boldsymbol{w}^{(t+1)} \leftarrow \boldsymbol{w}^{(t)} + \boldsymbol{\eta} \cdot \frac{\partial\, ll(w)}{\partial\, w}$$

t ← t+1

逻辑回归

- 每一种智能行为X都对应着一种人工X智能，行为X与环境需要进行交互



	贷款违约预测
input	申请人的信息，共47种特征，包含了如贷款金额、贷款年限、贷款利率、年收入、工作年限等
output	申请人贷款是否违约
feedback	正确与否

逻辑回归四要素与数据形态

- 表示：逻辑回归模型长什么样？
  机器编码 f_θ、 input、output、feedback。
- 推理：逻辑回归模型怎么用来解决问题？
  给定input，机器实现f_θ计算output。
- 学习：逻辑回归模型怎么来的？
  基于数据<input, output, feedback>集，
  给定f, 更新计算θ。
- 数据： <input, output, feedback>
  <申请人的信息,是否违约,模型判断正确与否>

1. 算法/模型： f（及部分θ）
2. 计算： f_θ/input/output/feedback转换
3. 数据： <input, output, feedback>
4. 知识： θ（及部分f）

线性分类器和逻辑回归的总结
（以线性分类器为例）

逻辑回归

- 描述决策边界与线性分类器
- 使用类别概率表达预测结果的置信度
- 定义逻辑回归模型
- 将逻辑回归输出结果解释为类别概率
- 分析系数取值对逻辑回归输出的影响
- 使用似然函数衡量分类器质量
- 通过梯度下降法优化负对数似然损失函数来训练逻辑回归模型