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

2.4 线性回归

王翔

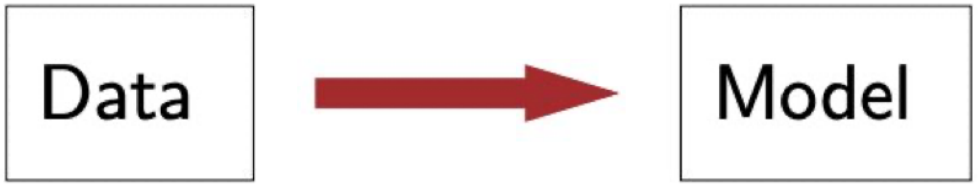
xiangwang@ustc.edu.cn

目录

目录

机器学习（Machine Learning）

- 将数据转化为模型的过程
- 使用该模型进行推断与预测



数据　　模型

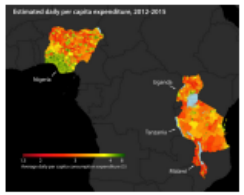
机器学习（Machine Learning）

- x:通常为实向量（句子或图像写成向量表示）
- y:取决于任务类型

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 → 资产财富指数

房价估计：　　房屋信息（位置，面积） → 房价

到达时间：　　目的地，天气，时间 → 到达时间

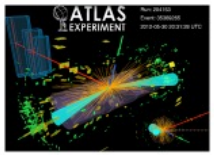
二分类问题（Binary Classification）

分类器
(classifier)

$x \rightarrow f \rightarrow y \in \{+1, -1\}$ 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 $y \in \{1, \ldots, K\}$

结构化预测（Structured Prediction）

$x \rightarrow \boxed{f} \rightarrow y$ 输出为复杂对象

机器对话: 历史对话记录 → 下一句话

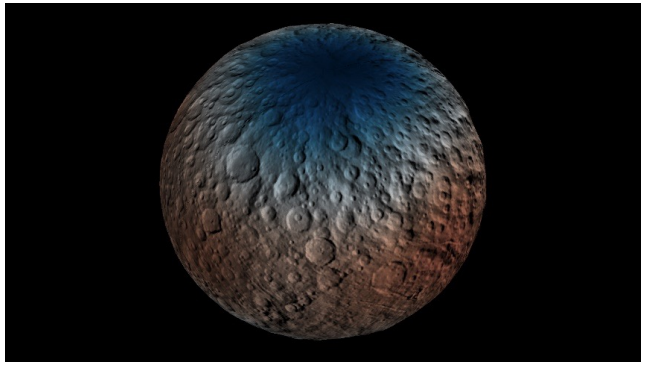
图像字幕: 图像 → 描述图像内容

图像分割: 图像 → 分割图

谷神星的发现历程

1801 年天文学家皮亚齐发现谷神星并在它被太阳强光遮蔽之前进行了 19 次位置观测，每个数据点都包含一个时间戳，以及赤经和赤纬。

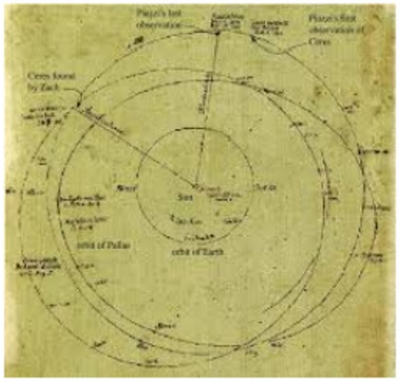
Time	Right ascension	Declination
Jan 01, 20:43:17.8	50.91	15.24
Jan 02, 20:39:04.6	50.84	15.30
...	...	...
Feb 11, 18:11:58.2	53.51	18.43



谷神星何时何地会再次出现?

高斯的胜利

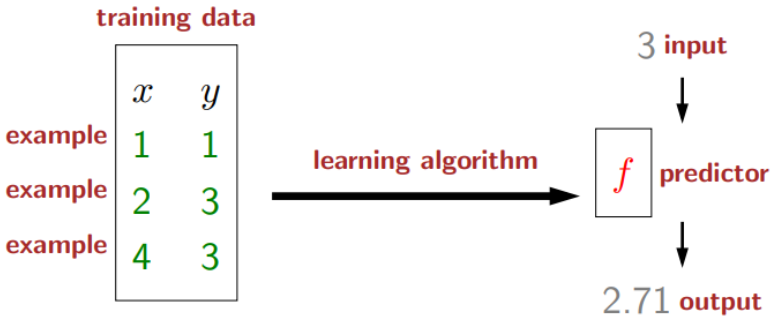
1801年9月：德国著名数学家高斯利用皮亚齐观测的数据，建立了谷神星轨道的模型，并用它来计算预测谷神星的出现位置。

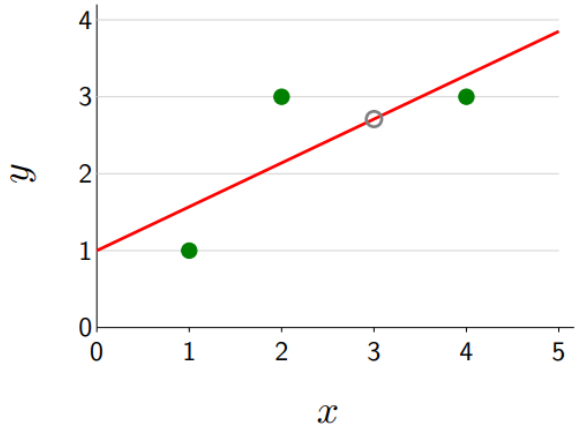


1801年12 月 7 日，谷神星出现在高斯预测的半度范围内，比其他天文学家更准确。

方法：最小二乘线性回归
(least squares linear regression)

线性回归框架





使用什么模型? 假设类（Hypothesis Class）

怎么确定一个模型的好坏? 损失函数（Loss Function）

怎么求解最优的模型? 优化算法（Optimization Algorithm）

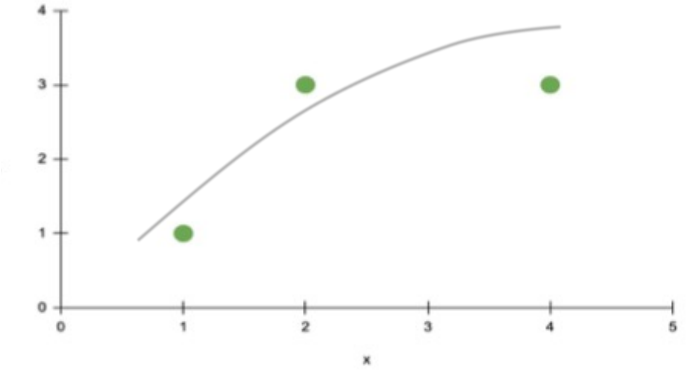
目录

课程安排

假设类	损失函数	优化算法
线性回归	平方损失	梯度下降
多项式回归	为什么使用平方损失	随机梯度下降

假设类（Hypothesis Class）

- 模型：我们对于世界如何运行的假设
- 假设：输入x和输出y之间的关系
- 回归模型



$$y = f(x) + \epsilon, \quad \mathrm{E}(\epsilon) = 0$$

y: 真实值 （target or ground-truth to be predicted）

$\hat{y} = f(x)$: 预测值 （prediction or estimation）

$\epsilon = y - \hat{y}$: 误差 （error or residual）

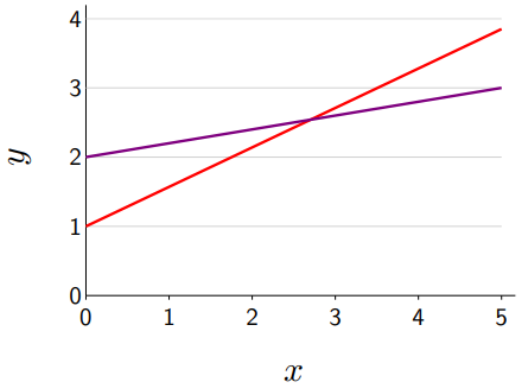
- 注意：我们假设的模型不一定必须是线性

假设类：线性回归模型（Linear Regression Model）

$$f(x) = 1 + 0.57x$$

$$f(x) = 2 + 0.2x$$

$$f(x) = w_1 + w_2 x$$



向量表示：　参数向量/模型参数　　特征提取器　特征向量

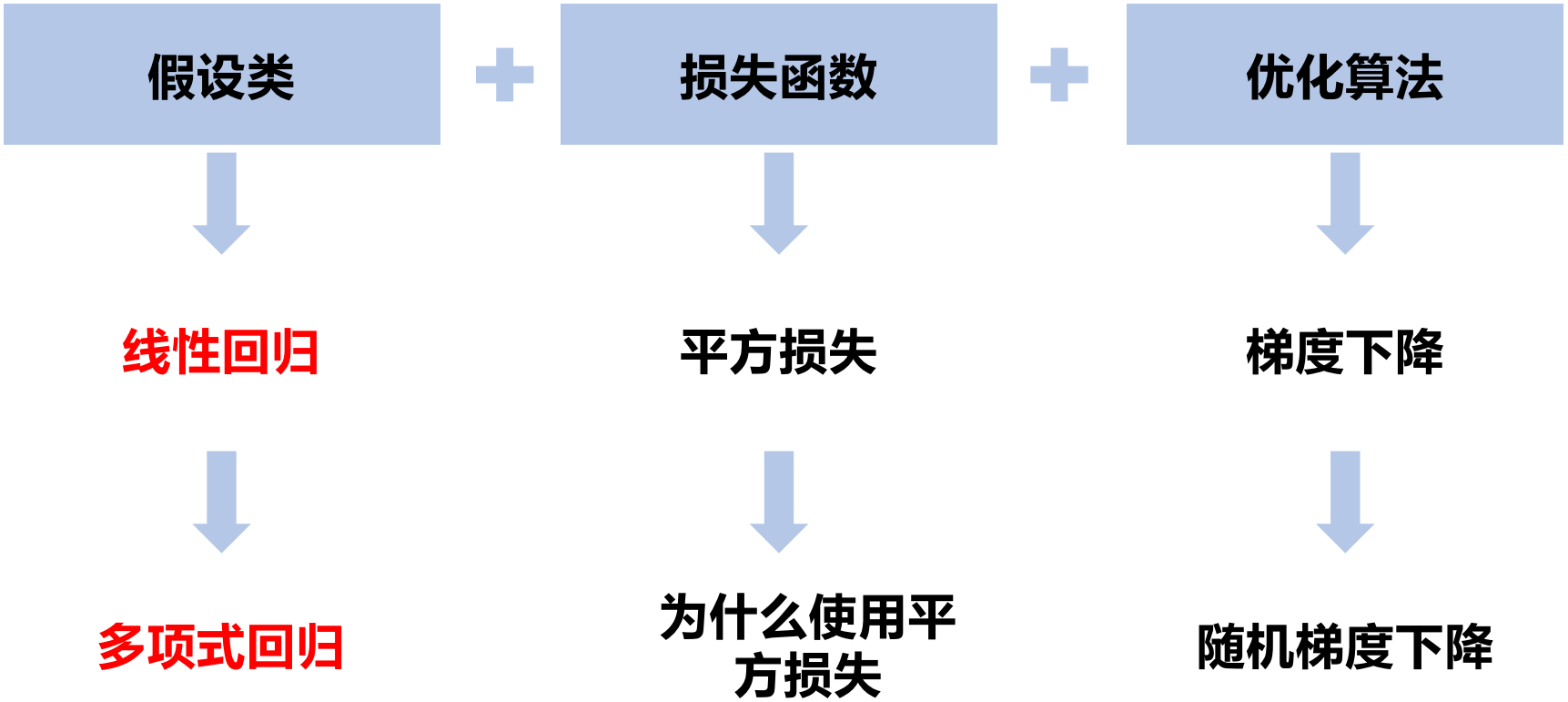
$$\mathbf{w} = [w_1, w_2] \qquad \phi(x) = [1, x]$$

$$f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x)$$

假设类：　$\mathcal{F} = \{f_{\mathbf{w}} : \mathbf{w} \in \mathbb{R}^2\}$

（预测器f的集合）

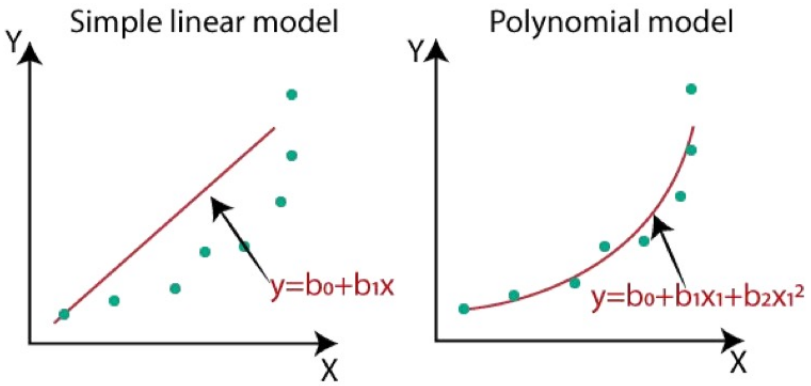
课程安排

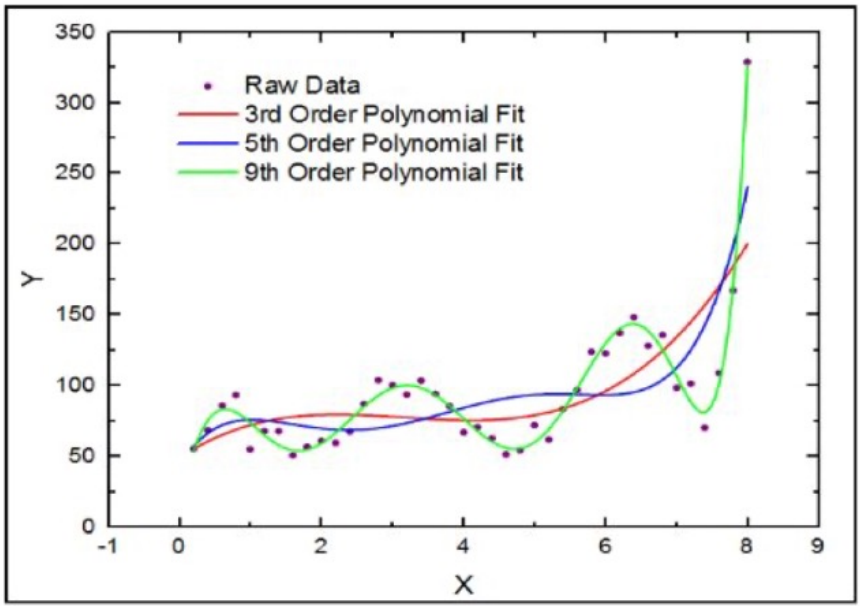


非线性特征

当面对更复杂的数据时，简单的线性模型可能难以拟合数据，这时我们可以考虑使用非线性特征。

线性回归 vs 多项式回归





二阶特征

$$\phi(x) = [1, x, x^2]$$

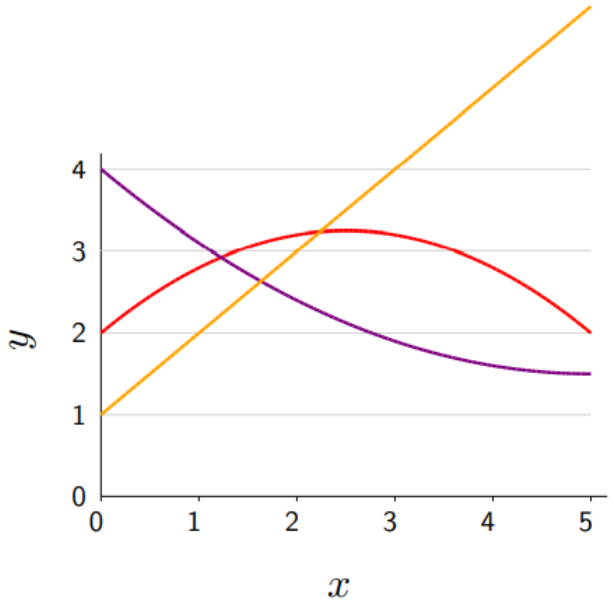
Example: $\phi(3) = [1, 3, 9]$

$$f(x) = [2, 1, -0.2] \cdot \phi(x)$$

$$f(x) = [4, -1, 0.1] \cdot \phi(x)$$

$$f(x) = [1, 1, 0] \cdot \phi(x)$$

$$\mathcal{F} = \{f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x) : \mathbf{w} \in \mathbb{R}^3\}$$



改变特征提取器$\phi(x)$获取非线性特征

多项式回归（Polynomial Regression）

$$\phi(x) = [1, x, x^2, x^3, \cdots, x^{d-1}]$$

$$f(x) = \mathbf{w} \cdot \phi(x)$$

将$\phi(x)$不同维度的值当做不同的特征

$$\text{feature}_1 = 1(\text{constant})$$

$$\text{feature}_2 = x$$

$$\text{feature}_3 = x^2$$

$$\cdots$$

$$\text{feature}_d = x^{d-1}$$

具有周期性结构的特征

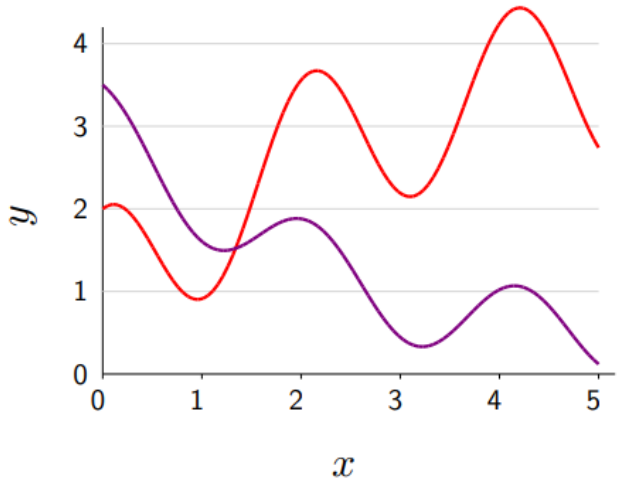
$$\phi(x) = [1, x, x^2, \cos(3x)]$$

Example: $\phi(2) = [1, 2, 4, 0.96]$

$$f(x) = [1, 1, -0.1, 1] \cdot \phi(x)$$

$$f(x) = [3, -1, 0.1, 0.5] \cdot \phi(x)$$

$$\mathcal{F} = \{f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x) : \mathbf{w} \in \mathbb{R}^4\}$$



使用任何你希望的特征

分段常数特征

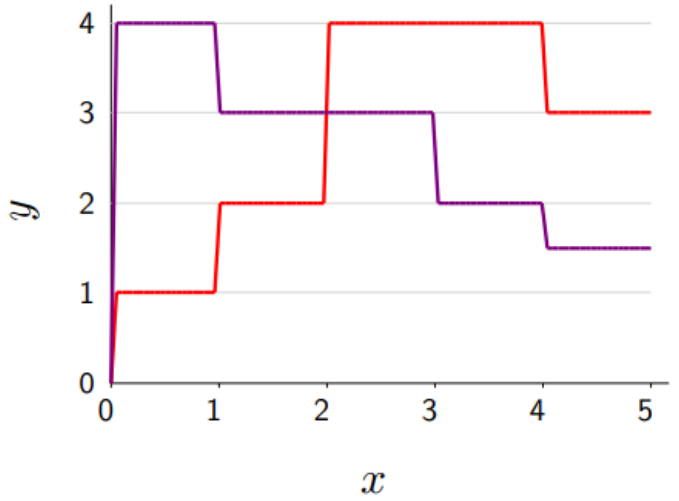
$$\phi(x) = [\mathbf{1}[0 < x \leq 1], \mathbf{1}[1 < x \leq 2], \mathbf{1}[2 < x \leq 3], \mathbf{1}[3 < x \leq 4], \mathbf{1}[4 < x \leq 5]]$$

Example: $\phi(2.3) = [0, 0, 1, 0, 0]$

$$f(x) = [1, 2, 4, 4, 3] \cdot \phi(x)$$

$$f(x) = [4, 3, 3, 2, 1.5] \cdot \phi(x)$$

$$\mathcal{F} = \{f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x) : \mathbf{w} \in \mathbb{R}^5\}$$



划分输入空间获得分段常数特征

“线性”体现在哪?

预测值: $f_w(x) = w \cdot \phi(x)$

关于w线性吗? Yes

关于$\phi(x)$线性吗? Yes

关于x线性吗? No!

关键点：非线性

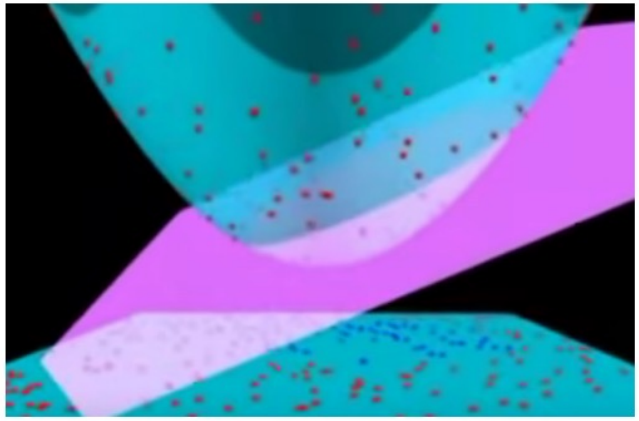
- 表达能力：$w \cdot \phi(x)$是x的非线性函数
- 效率：$w \cdot \phi(x)$是w的线性函数

小结

$$f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x)$$

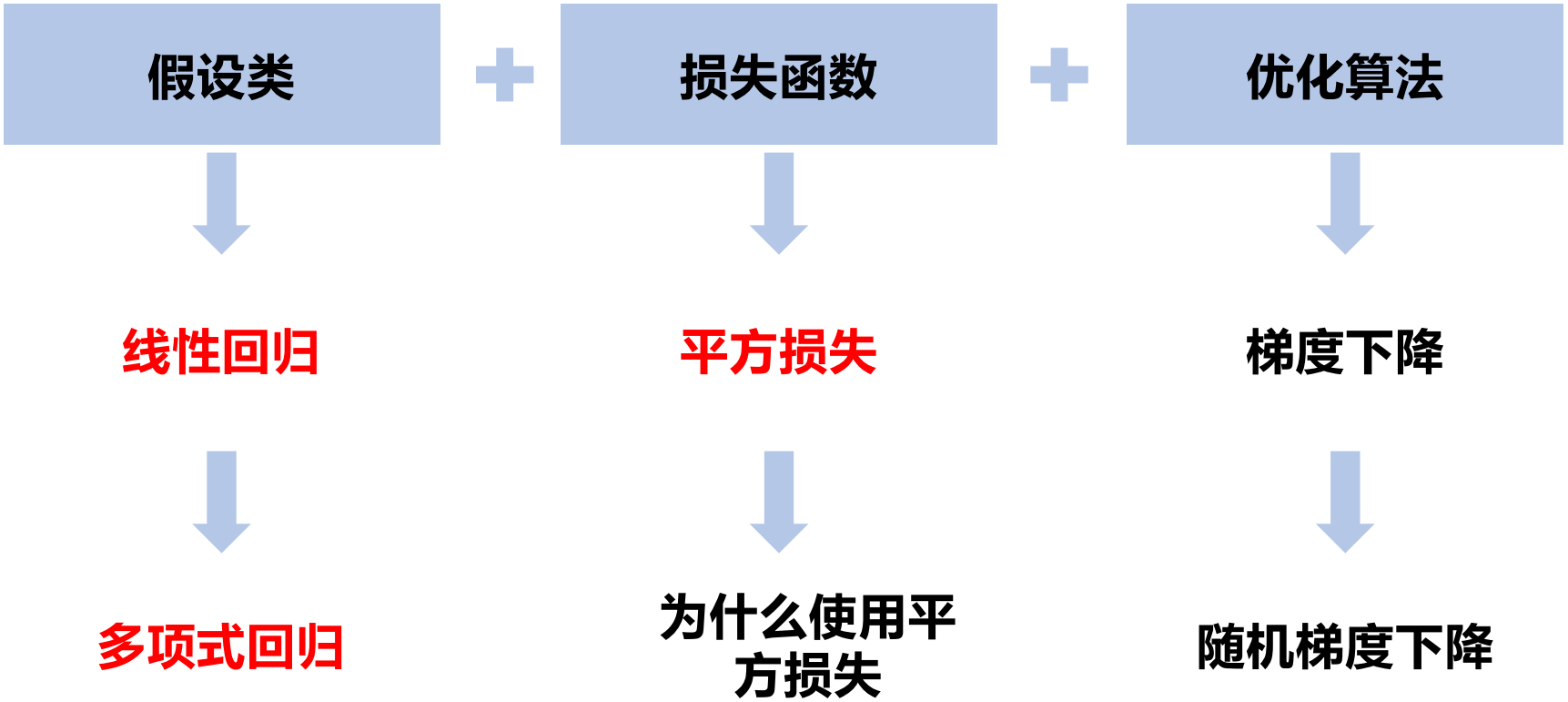
linear in $\mathbf{w}, \phi(x)$

non-linear in x



目录

课程安排

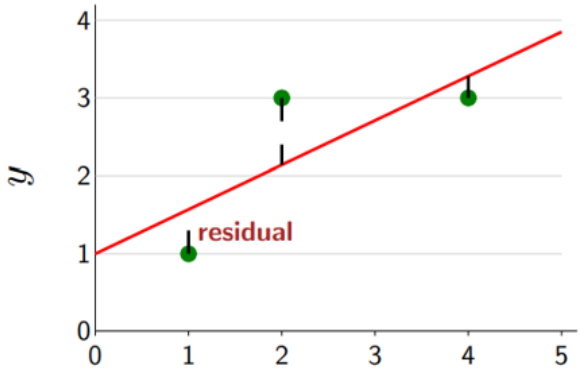


损失函数

$$f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x)$$
$$\mathbf{w} = [1, 0.57]$$
$$\phi(x) = [1, x]$$

训练数据$\mathcal{D}_{\text{train}}$

x	y
1	1
2	3
4	3



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

$$\text{Loss}(1, 1, [1, 0.57]) = ([1, 0.57] \cdot [1, 1] - 1)^2 = 0.32$$
$$\text{Loss}(2, 3, [1, 0.57]) = ([1, 0.57] \cdot [1, 2] - 3)^2 = 0.74$$
$$\text{Loss}(4, 3, [1, 0.57]) = ([1, 0.57] \cdot [1, 4] - 3)^2 = 0.08$$

均方误差（MSE, mean squared error） $\text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} \text{Loss}(x, y, \mathbf{w})$

$\text{TrainLoss}([1, 0.57]) = 0.38$

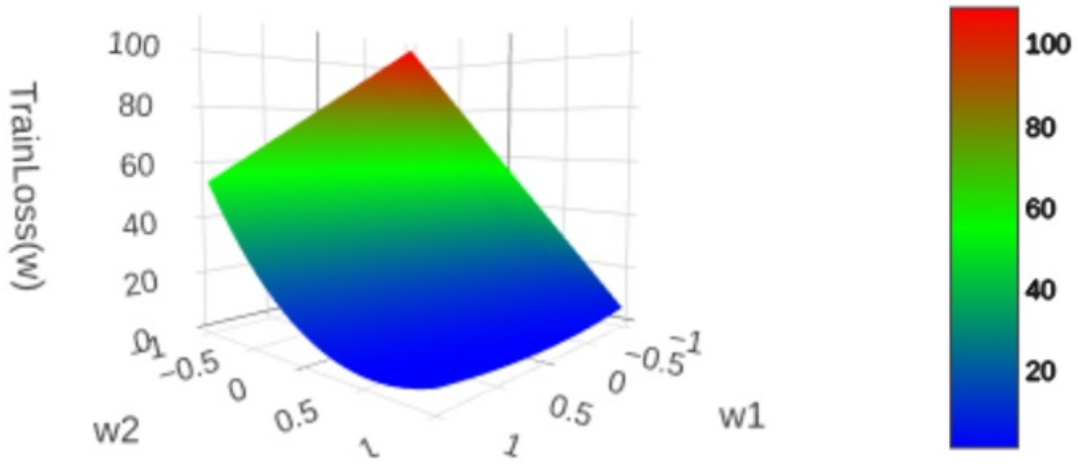
- 问题： 为什么使用平方损失而不是绝对值损失?

损失函数的可视化

优化目标：

$$\text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y)\in\mathcal{D}_{\text{train}}} (f_{\mathbf{w}}(x) - y)^2$$

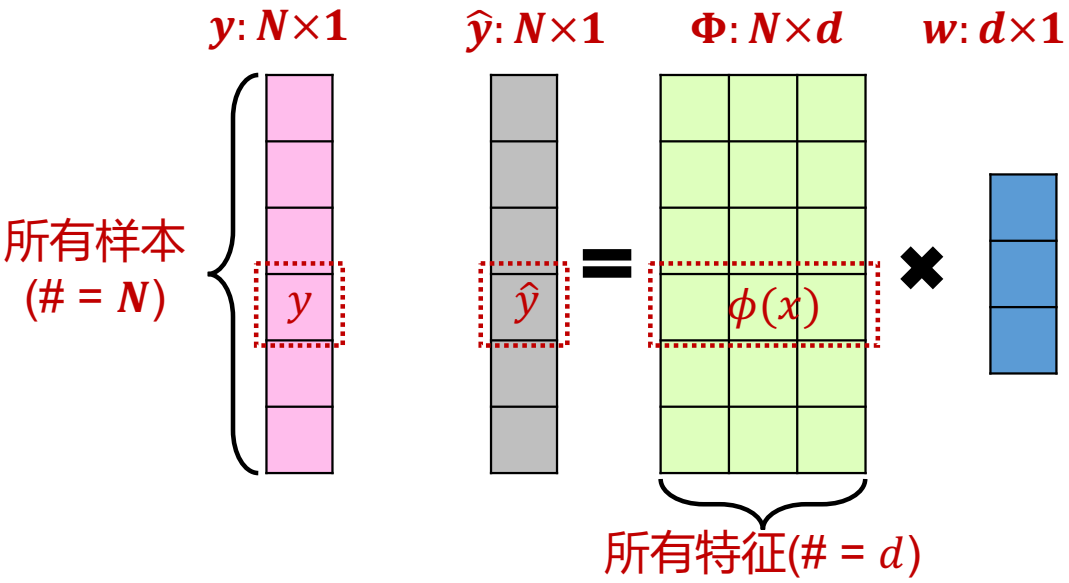
$$\min_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})$$



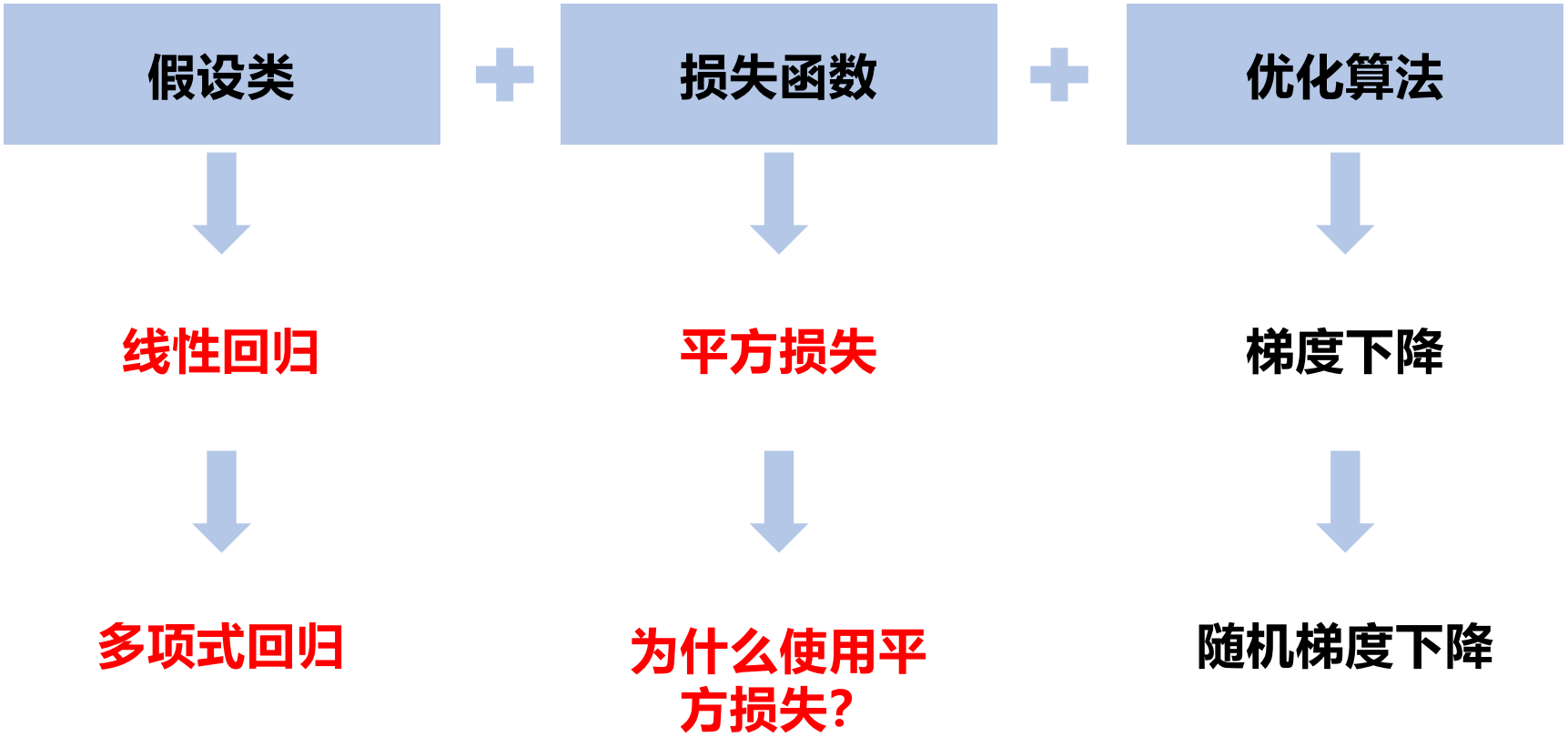
寻找参数w最小化损失函数

损失函数的矩阵形式

$$\begin{aligned}\text{TrainLoss}(\mathbf{w}) &= \frac{1}{|\mathcal{D}_{\text{train}}|}\sum_{(x,y)\in\mathcal{D}_{\text{train}}}(f(x)-y)^2 \\ &= \frac{1}{|\mathcal{D}_{\text{train}}|}\sum_{(x,y)\in\mathcal{D}_{\text{train}}}(\mathbf{w}\cdot\phi(x)-y)^2 \\ &= \frac{1}{|\mathcal{D}_{\text{train}}|}(\mathbf{y}-\Phi\mathbf{w})^\top(\mathbf{y}-\Phi\mathbf{w})\end{aligned}$$



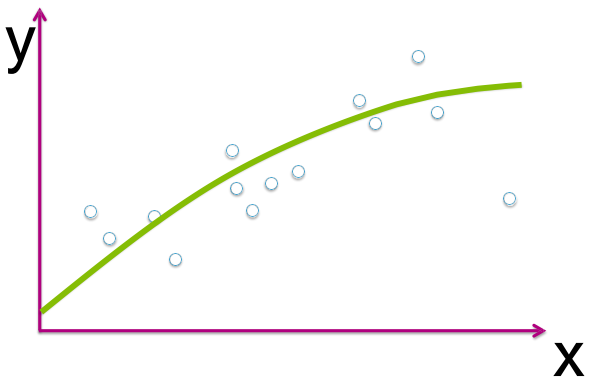
课程安排



为什么最小化平方误差?

输入x 和输出 y的关系

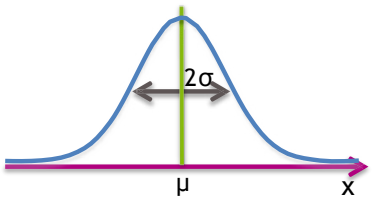
$$y = f(x) + \epsilon, \quad \mathrm{E}(\epsilon) = 0$$



对于噪声\偏差 ϵ:

- 我们已经有: $E(\epsilon) = 0$
- 进一步我们假设其服从高斯分布，即$\epsilon \sim N(\mu, \sigma^2)$

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



于是我们得到y的分布如下:

- $y = \phi(x)w + \epsilon$
- $y|x; w \sim N(\phi(x)w, \sigma^2)$

为什么最小化平方误差?

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

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

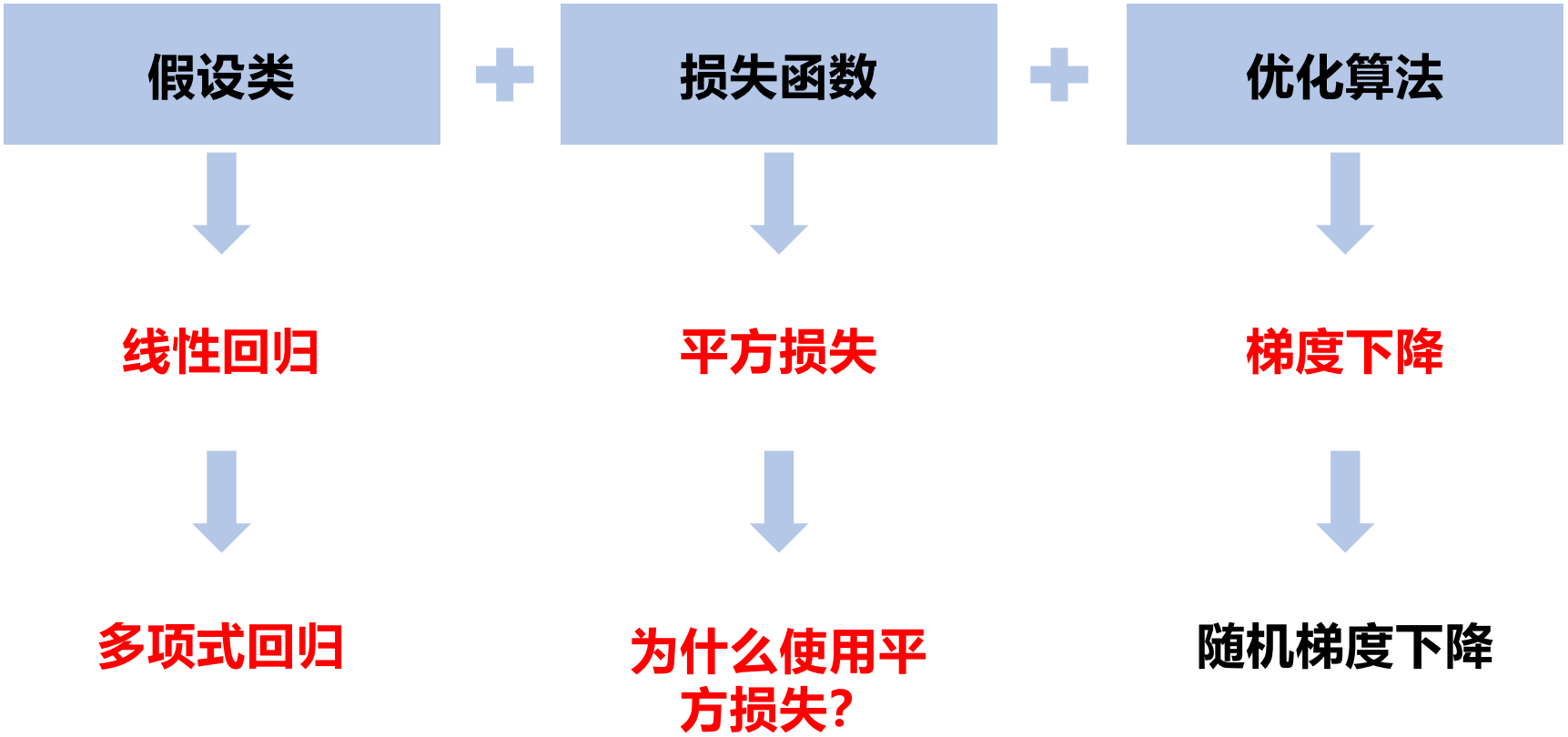
Maximize log-likelihood estimate w.r.t. w: 对数似然

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

高斯假设下的最大似然估计 = 最小化平方误差

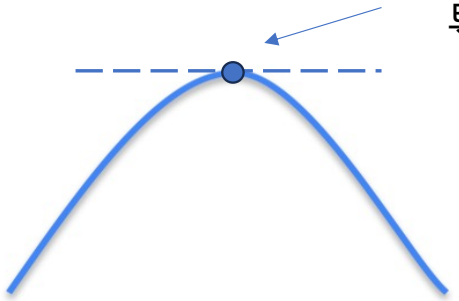
目录

课程安排



回顾：凸/凹函数

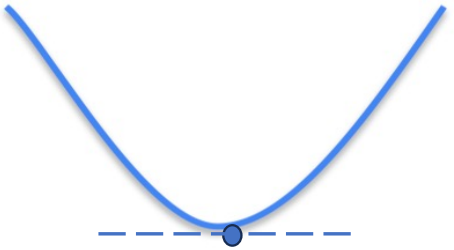
凹函数
CONCAVE



导数等于0

$$\max_{w} \; g(w)$$

凸函数
CONVEX

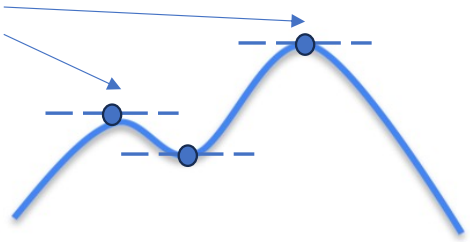


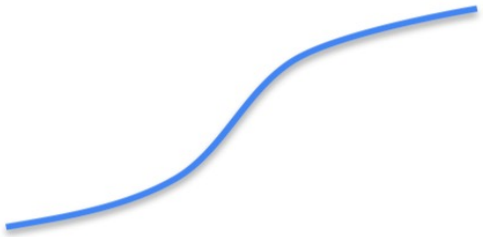
$$\min_{w} \; g(w)$$

导数等于0

NEITHER

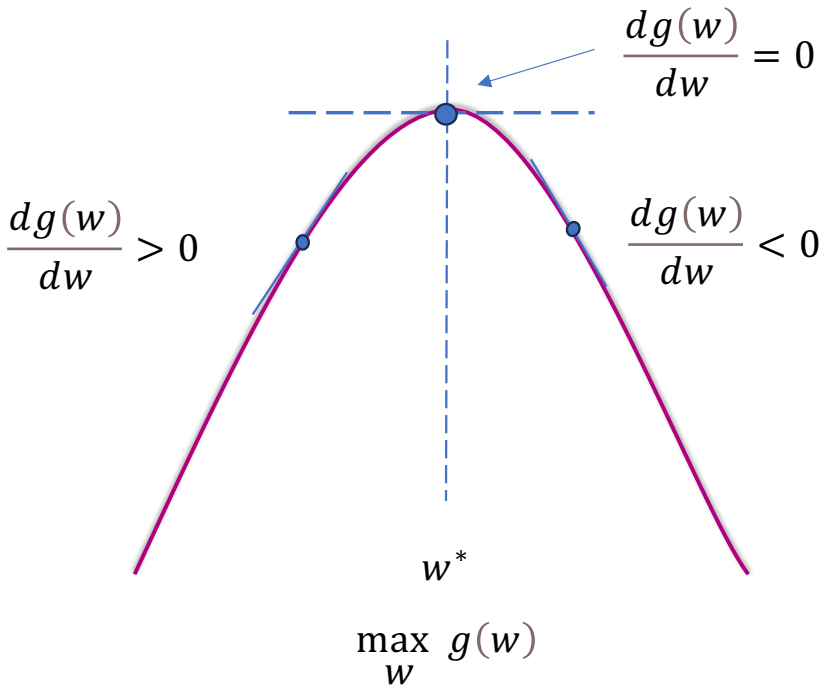
导数等于0的解有多个





导数等于0无解

回顾：利用梯度上升找最大值



我们如何判断应该增大还是减小w的值呢?

当 $\frac{dg(w)}{dw} > 0$ 时， 增大w

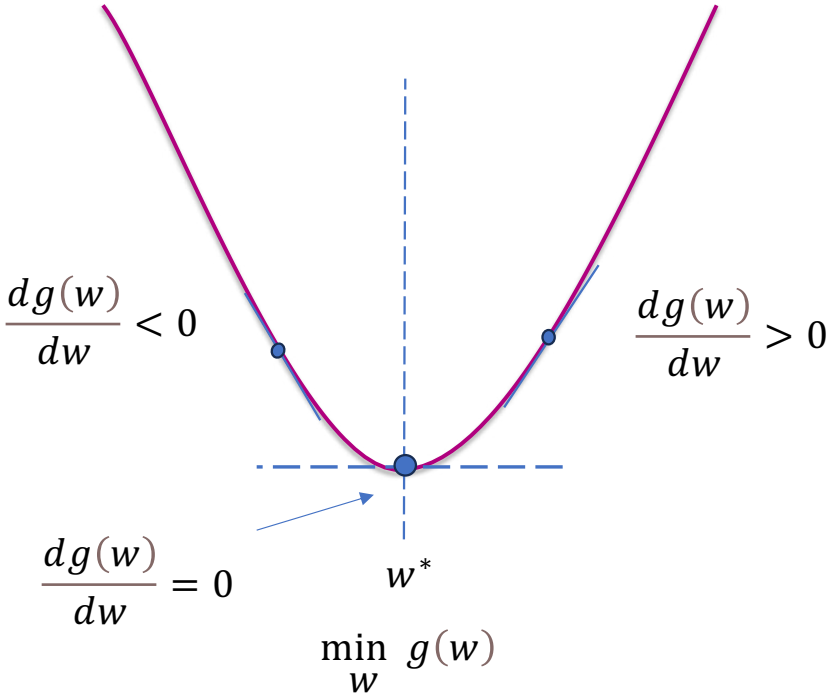
当 $\frac{dg(w)}{dw} < 0$ 时， 减小w

第t次迭代

$$w^{(t+1)} \leftarrow w^{(t)} + \eta \frac{dg(w)}{dw}$$

步长

回顾：利用梯度下降找最小值



当 $\frac{dg(w)}{dw} > 0$ 时，　增大w

当 $\frac{dg(w)}{dw} < 0$ 时，　减小w

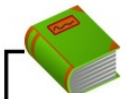
第t次迭代

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \frac{dg(w)}{dw}$$

步长

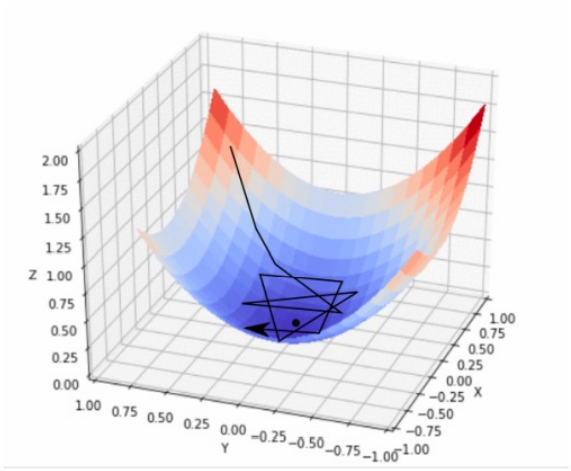
如何优化损失函数?

目标：$\min_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})$



定义：梯度

梯度 $\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})$ 是训练损失上升最快的方向



梯度下降

Algorithm: gradient descent

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

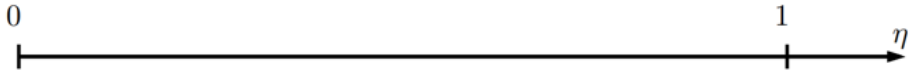
For $t = 1, \ldots, T$: epochs

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

步长（学习率/learning rate）

$$\mathbf{w} \leftarrow \mathbf{w} - \underbrace{\eta}_{\text{step size}} \nabla_{\mathbf{w}} \text{Loss}(x, y, \mathbf{w})$$

问题：我们该怎样选取步长？



保守，更稳定　　激进，更快

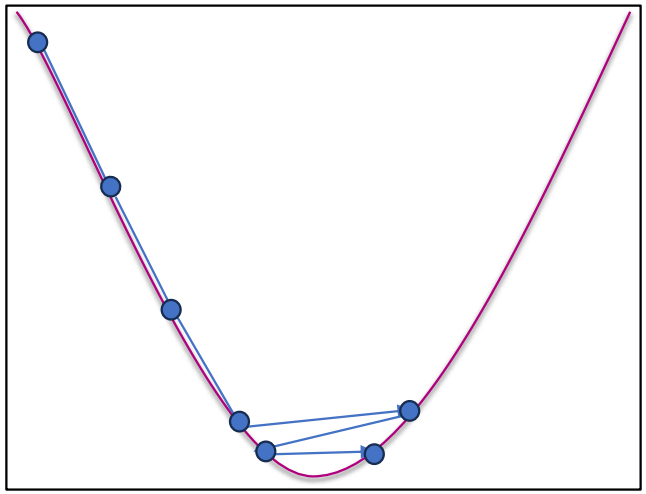
策略：

- 步长为常数：$\eta = 0.1$
- 步长逐渐减小：$\eta = 1/\sqrt{\text{当前已更新次数}}$

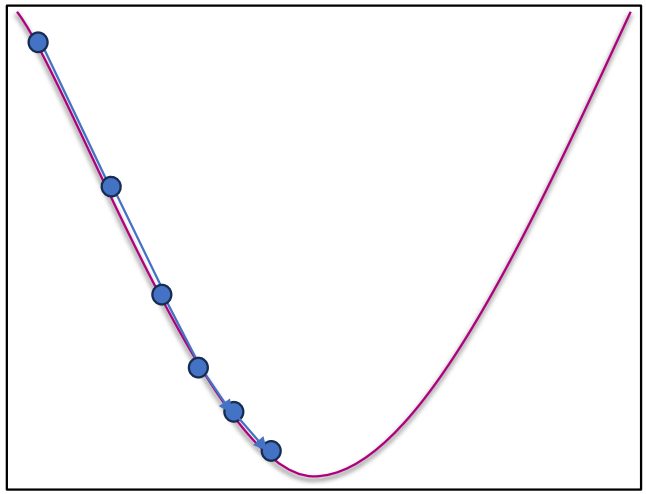
- 一些其他策略：
  - learning rate decay
  - Adagrad
  - **Adam**

选择合适的步长

- 步长太大?
- 步长太小?
- 通常的策略: 常数 vs 不断减小



constants



shrinking step size

$$\eta(t) = \frac{\alpha}{t}$$

$$\eta(t) = \frac{\alpha}{\sqrt{t}}$$

停止/收敛

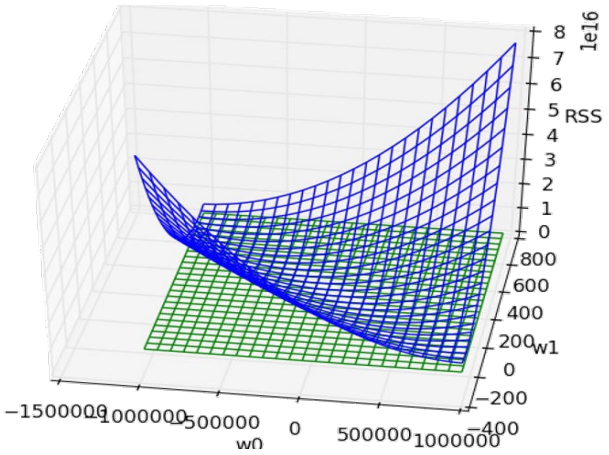
- 理论上对于凸函数，当梯度为0时算法收敛
- 在实际应用当中，我们会设定一个阈值ε或者最大迭代次数T，当梯度的模长满足

$$\|\nabla_{\mathbf{w}} \mathrm{TrainLoss}(\mathbf{w})\| < \epsilon$$

或者迭代次数达到T时，停止迭代。

多维梯度

$$\nabla_{\mathbf{w}}\text{TrainLoss}(\mathbf{w}) = \begin{bmatrix} \frac{d\,\text{TrainLoss}}{d\,w_1} \\ \frac{d\,\text{TrainLoss}}{d\,w_2} \\ \cdots \\ \frac{d\,\text{TrainLoss}}{d\,w_d} \end{bmatrix}$$

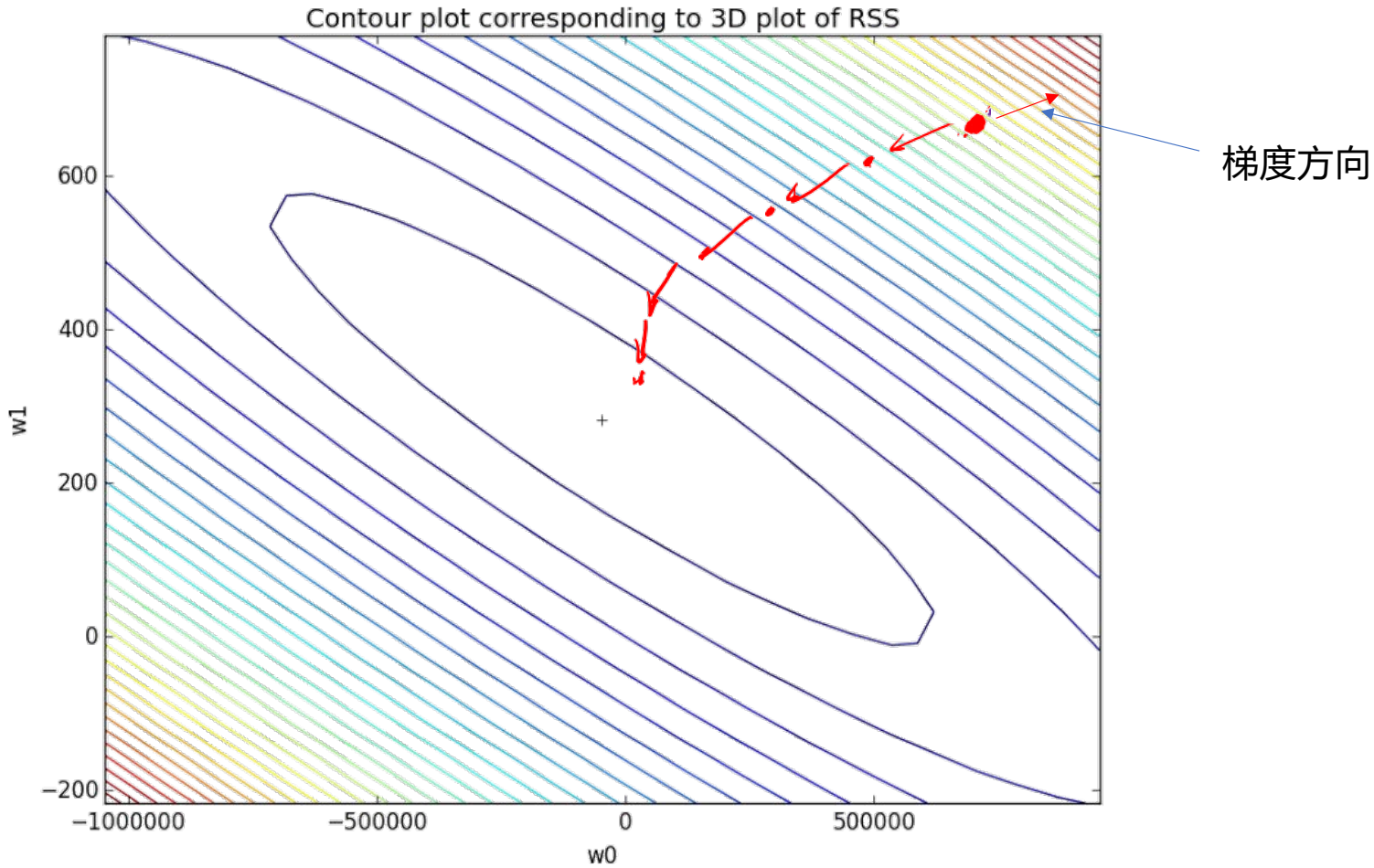


最小值处带有切平面的残差平方和三维图

- 梯度是一个d维的向量，其中d是特征的数量（包括偏移）
- 向量当中的每一项表示损失函数对其中一个参数的偏导。

Q: $g(w) = 5w_1 + 10w_1w_2 + 2w_1^2$, $\nabla g(w)$?

多维梯度



残差平方和（RSS）三维图对应的等高线图

优化例子：线性回归

$$\begin{aligned}\arg\min_{\mathbf{w}} \text{TrainLoss}(\mathbf{w}) &= \arg\min_{\mathbf{w}} \frac{1}{|\mathcal{D}_{\text{train}}|}(\mathbf{y}-\Phi\mathbf{w})^\top(\mathbf{y}-\Phi\mathbf{w}) \\ &= \arg\min_{\mathbf{w}} \underbrace{(\mathbf{y}-\Phi\mathbf{w})^\top(\mathbf{y}-\Phi\mathbf{w})}_{\text{RSS}(\mathbf{w})}\end{aligned}$$

residual sum of squares 残差平方和

$$\begin{aligned}\nabla_{\mathbf{w}}\text{RSS}(\mathbf{w}) &= \nabla_{\mathbf{w}}(\mathbf{y}-\Phi\mathbf{w})^\top(\mathbf{y}-\Phi\mathbf{w}) \\ &= -2\Phi^\top(\mathbf{y}-\Phi\mathbf{w})\end{aligned}$$

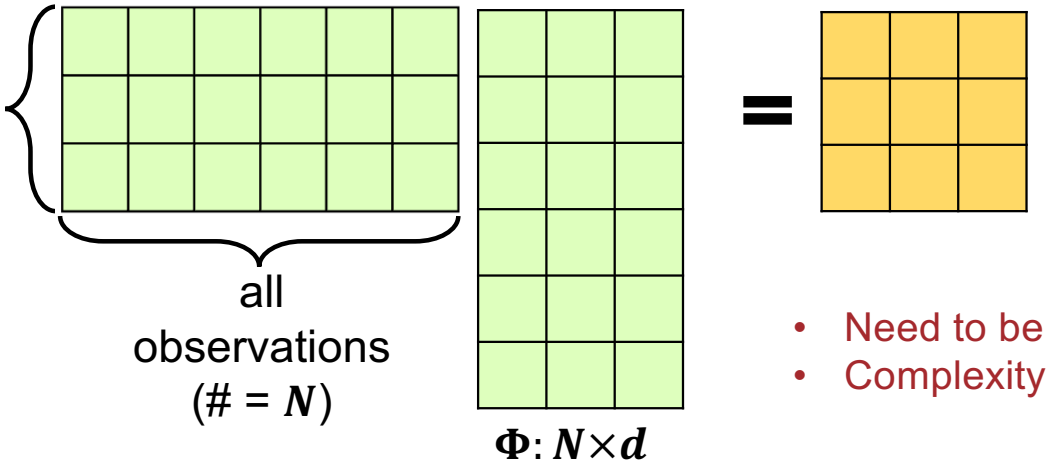
$$\nabla_{\mathbf{w}}\text{TrainLoss}(\mathbf{w}) = \frac{2}{|\mathcal{D}_{\text{train}}|}\Phi^\top(\mathbf{y}-\Phi\mathbf{w})$$

方法一：梯度设置为0

令 $$\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w}) = \frac{2}{|\mathcal{D}_{\text{train}}|} \Phi^{\top}(\mathbf{y} - \Phi\mathbf{w}) = 0$$

我们得到参数向量 w 的闭式解

$$\mathbf{w}^* = (\Phi^{\top}\Phi)^{-1}\Phi^{\top}\mathbf{y}$$



- Need to be invertible（须满足可逆条件）
- Complexity of inverse (矩阵求逆计算复杂度)

方法2：梯度下降

Objective function:

目标函数

$$\text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y)\in\mathcal{D}_{\text{train}}} (\mathbf{w}\cdot\phi(x) - y)^2$$

Gradient (use chain rule):

使用链式法则计算梯度

$$\nabla_{\mathbf{w}}\text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y)\in\mathcal{D}_{\text{train}}} 2(\underbrace{\mathbf{w}\cdot\phi(x) - y}_{\text{prediction}-\text{target}})\phi(x)$$

方法2：梯度下降

training data $\mathcal{D}_{\text{train}}$

x	y
1	1
2	3
4	3

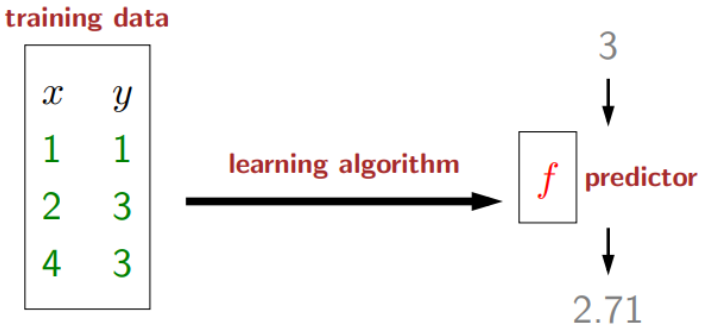
$$\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y)\in\mathcal{D}_{\text{train}}} 2(\mathbf{w}\cdot\phi(x) - y)\phi(x)$$

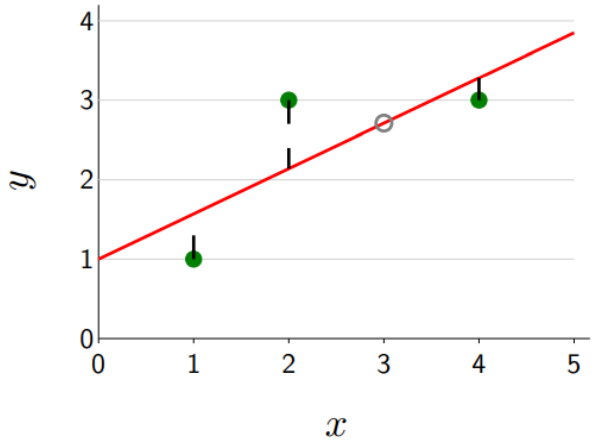
梯度更新： $\mathbf{w} \leftarrow \mathbf{w} - 0.1\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})$

t	$\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})$	$\mathbf{w}$
		$[0,0]$
1	$\underbrace{\frac{1}{3}(2([0,0]\cdot[1,1]-1)[1,1]+2([0,0]\cdot[1,2]-3)[1,2]+2([0,0]\cdot[1,4]-3)[1,4])}_{=[-4.67,-12.67]}$	$[0.47,1.27]$
2	$\underbrace{\frac{1}{3}(2([0.47,1.27]\cdot[1,1]-1)[1,1]+2([0.47,1.27]\cdot[1,2]-3)[1,2]+2([0.47,1.27]\cdot[1,4]-3)[1,4])}_{=[2.18,7.24]}$	$[0.25,0.54]$
...	...	...
200	$\underbrace{\frac{1}{3}(2([1,0.57]\cdot[1,1]-1)[1,1]+2([1,0.57]\cdot[1,2]-3)[1,2]+2([1,0.57]\cdot[1,4]-3)[1,4])}_{=[0,0]}$	$[1,0.57]$

Pytorch或者TensorFlow等深度学习框架集成了自动求导的功能

小结





什么样的模型?
假设类

如何确定模型的好坏?
损失函数

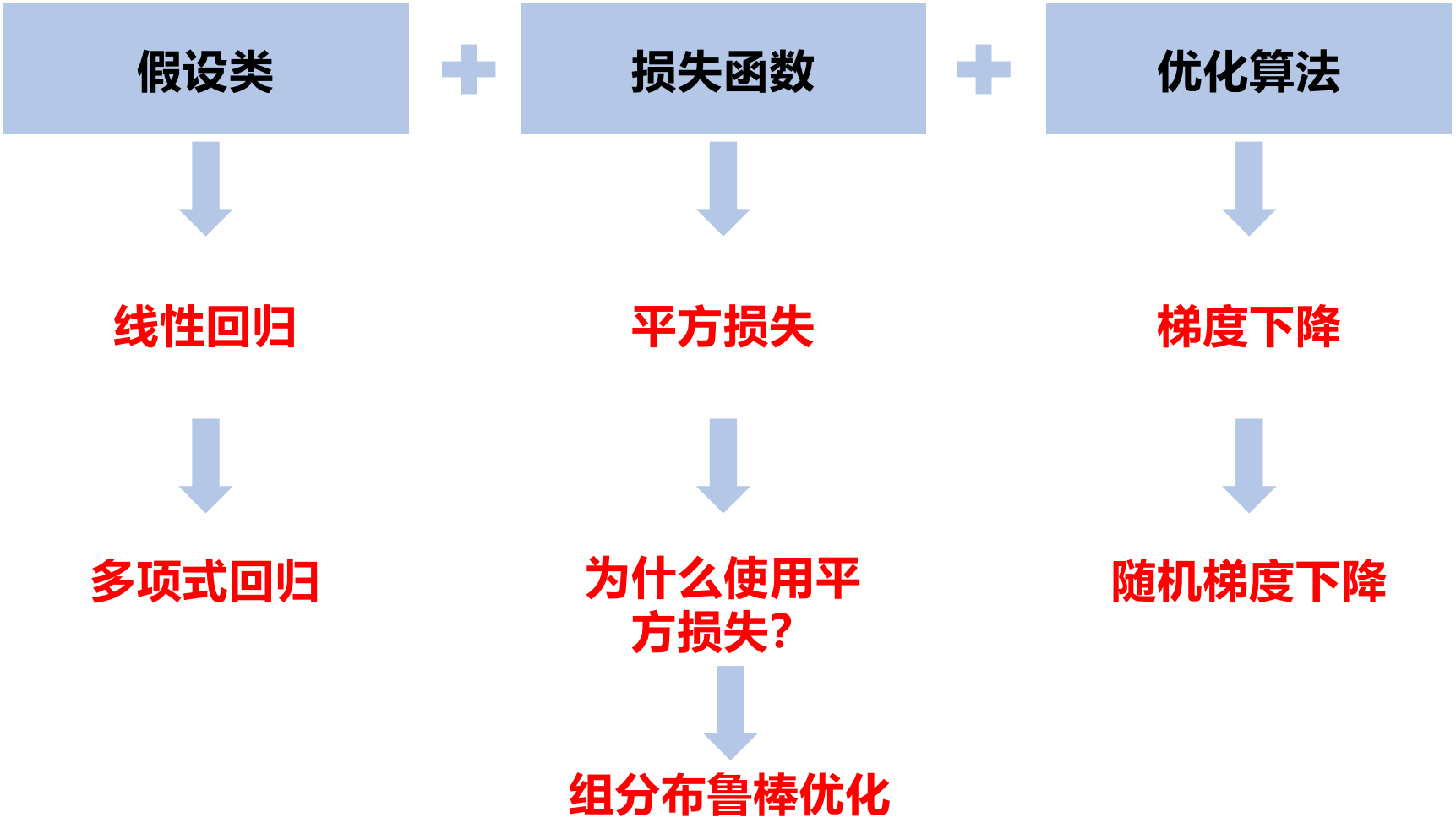
如何找到最好的模型?
优化算法

线性函数
$\mathcal{F} = \{f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x)\}, \phi(x) = [1, x]$

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

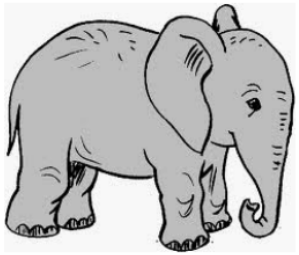
梯度下降
$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla \textbf{TrainLoss}(\mathbf{w})$

课程安排


假设类
损失函数
优化算法
线性回归
平方损失
梯度下降
多项式回归
为什么使用平方损失?
随机梯度下降
组分布鲁棒优化

方法2：梯度下降的缺点

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



Algorithm: gradient descent

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

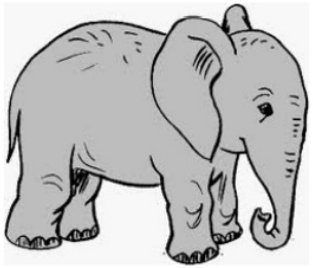
For $t = 1, \ldots, T$:

$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})$

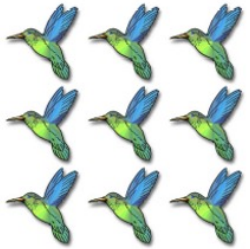
缺点：每次迭代都需要遍历所有训练样本，当训练数据量很大时，需要花费很长时间！

小结

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



梯度下降



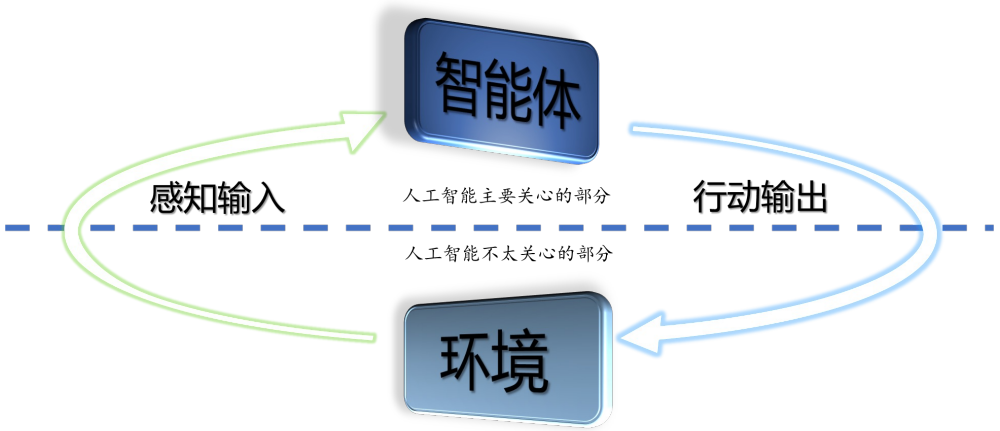
随机梯度下降

关键点：随机更新

数量弥补质量

回顾与总结：线性回归模型

- 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



	线性回归模型
input	特征信息
output	预测值
feedback	预测值与真实值的误差

回顾与总结：人工智能四要素与数据形态

- 表示：线性回归模型包括哪些部分？
  机器编码f_θ、input、output、feedback。
- 推理：得到的线性回归模型如何使用？
  给定input，机器实现f_θ计算output。
- 学习：如何得到最优的线性回归模型？
  基于数据<input, output, feedback>集，
  给定f，更新计算θ。

人工智能四要素（“知识”有待商榷）

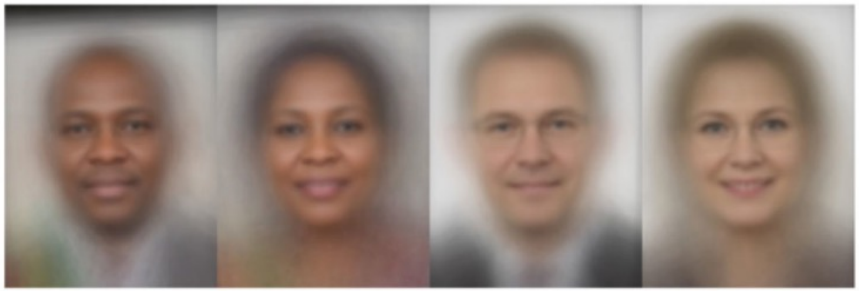
1. 算法/模型：线性模型参数
2. 计算：f_θ/input/output/feedback转换
3. 数据：<input, output, feedback>
4. 知识：θ（及部分f）

- 数据：<input, output, feedback>
  <特征信息，预测值，预测值与真实值的误差>

目录

关于性别的不公平性

Gender Classifier	Darker Male	Darker Female	Lighter Male	Lighter Female	Largest Gap
Microsoft	94.0%	79.2%	100%	98.3%	20.8%
FACE++	99.3%	65.5%	99.2%	94.0%	33.8%
IBM	88.0%	65.3%	99.7%	92.9%	34.4%



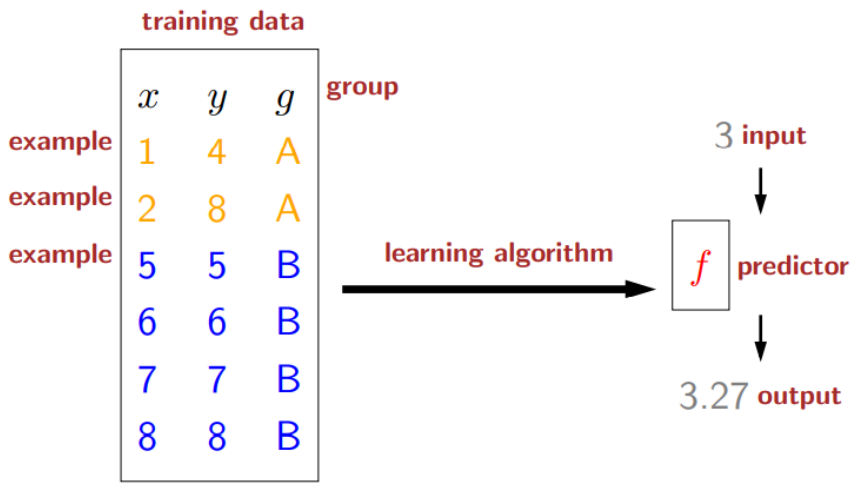
机器学习加剧了数据当中的不公平性

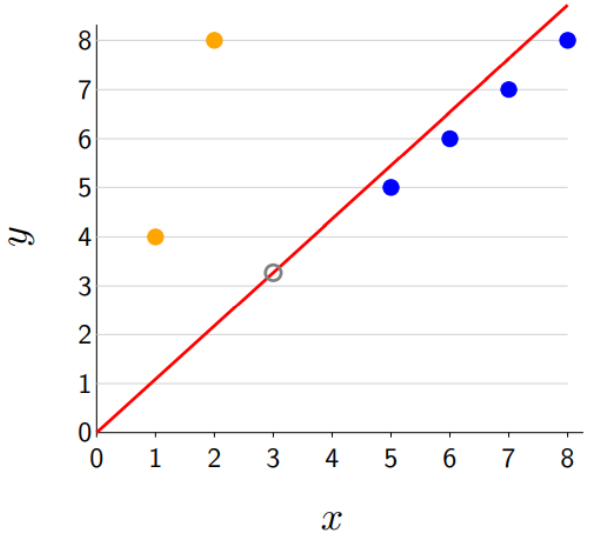
人脸识别造成的冤案

Wrongfully Accused by an Algorithm

In what may be the first known case of its kind, a faulty facial recognition match led to a Michigan man's arrest for a crime he did not commit.

线性回归在不同分组上的效果





$$f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x) \qquad \mathbf{w} = [w] \qquad \phi(x) = [x]$$

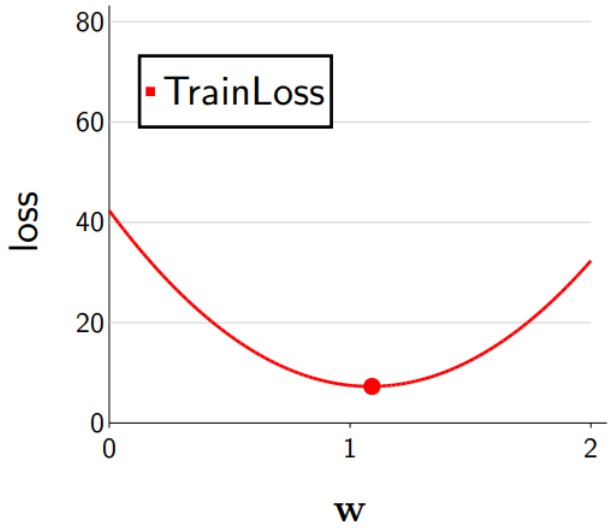
Note：模型f_w没有使用组别信息

尽管模型没有使用组别信息，但不公平性存在!

平均损失

$$\text{Loss}(x, y, \mathbf{w}) = (f_{\mathbf{w}}(x) - y)^2$$

x	y	g
1	4	A
2	8	A
5	5	B
6	6	B
7	7	B
8	8	B



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

$$\text{TrainLoss}(1) = \tfrac{1}{6}((1-4)^2 + (2-8)^2 + (5-5)^2 + (6-6)^2 + (7-7)^2 + (8-8)^2) = 7.5$$

分组计算损失

x	y	g
1	4	A
2	8	A
5	5	B
6	6	B
7	7	B
8	8	B

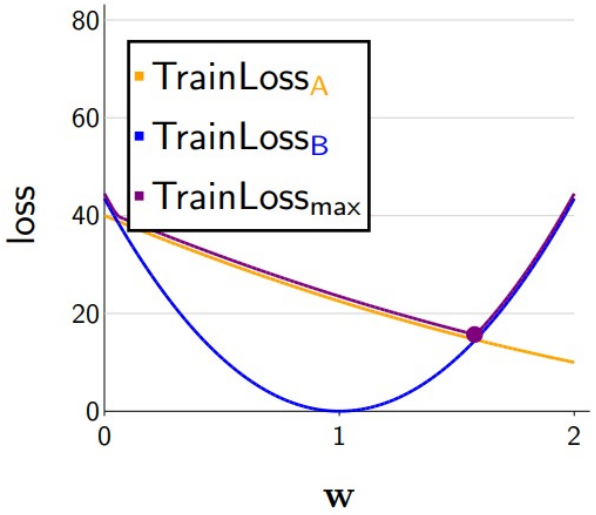
$$\text{TrainLoss}_g(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}(g)|} \sum_{(x,y)\in\mathcal{D}_{\text{train}}(g)} \text{Loss}(x, y, \mathbf{w})$$

$$\text{TrainLoss}_A(1) = \tfrac{1}{2}((1-4)^2 + (2-8)^2) = 22.5$$

$$\text{TrainLoss}_B(1) = \tfrac{1}{4}((5-5)^2 + (6-6)^2 + (7-7)^2 + (8-8)^2) = 0$$

计算最大组内损失

x	y	g
1	4	A
2	8	A
5	5	B
6	6	B
7	7	B
8	8	B



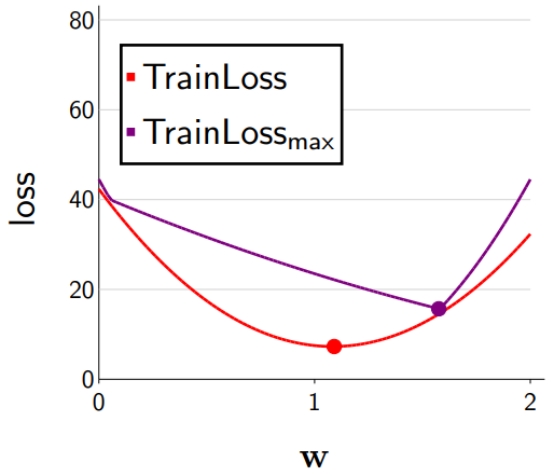
$$\text{TrainLoss}_{\text{max}}(\mathbf{w}) = \max_g \text{TrainLoss}_g(\mathbf{w})$$

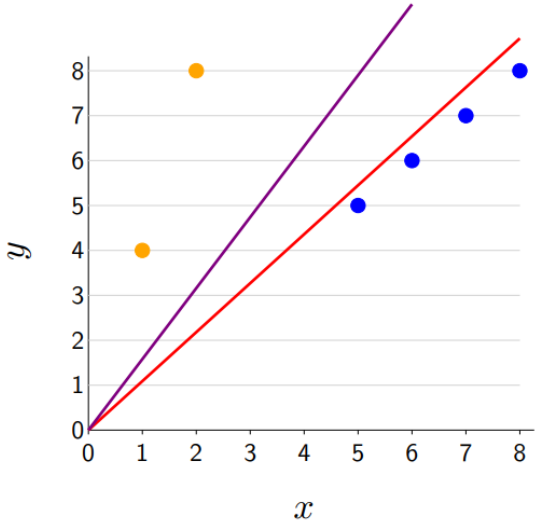
$$\text{TrainLoss}_A(1) = 22.5$$
$$\text{TrainLoss}_B(1) = 0$$
$$\text{TrainLoss}_{\text{max}}(1) = \max(22.5, 0) = 22.5$$

- **组分布鲁棒优化 group distributionally robust optimization** (group DRO)

平均损失 vs 最大组内损失

x	y	g
1	4	A
2	8	A
5	5	B
6	6	B
7	7	B
8	8	B





标准优化方式：

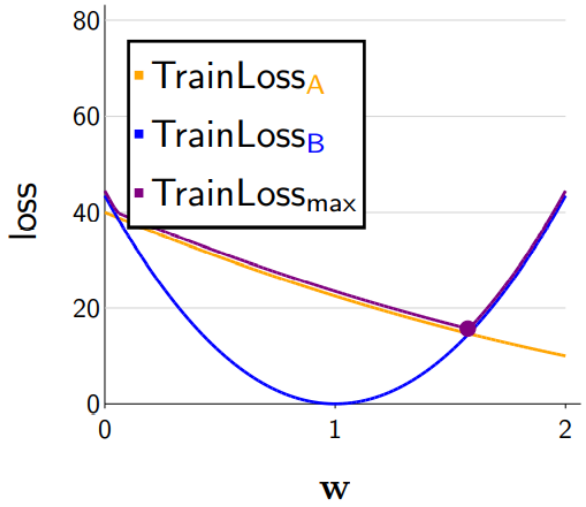
最小化 平均损失：$w = 1.09$

组分布鲁棒优化（group DRO）

最小化 最大组内损失：$w = 1.58$

通过梯度下降法优化

x	y	g
1	4	A
2	8	A
5	5	B
6	6	B
7	7	B
8	8	B



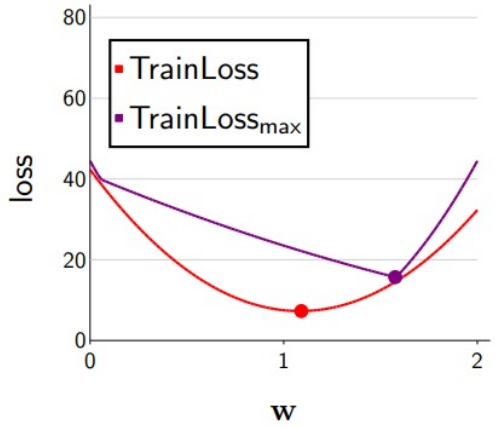
$$\text{TrainLoss}_{\text{max}}(\mathbf{w}) = \max_{g} \text{TrainLoss}_{g}(\mathbf{w})$$

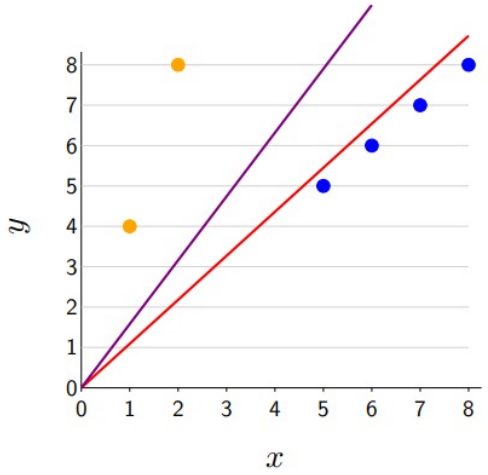
$$\nabla \text{TrainLoss}_{\text{max}}(\mathbf{w}) = \nabla \text{TrainLoss}_{g^*}(\mathbf{w})$$

$$\text{where } g^* = \arg\max_{g} \text{TrainLoss}_{g}(\mathbf{w})$$

小结

x	y	g
1	4	A
2	8	A
5	5	B
6	6	B
7	7	B
8	8	B





- 最大组内损失 ≠ 平均损失
- Group DRO：最小化 最大组内损失
- 更多细微差别：交叉性？ 不知道组信息？过拟合？