中山大学软件工程学院

SCHOOL OF SOFTWARE ENGINEERING

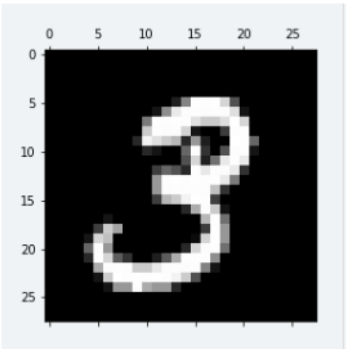
线性回归

廖国成

liaogch6@mail.sysu.edu.cn

人工智能
专业术语

人工智能模型的形式化表达



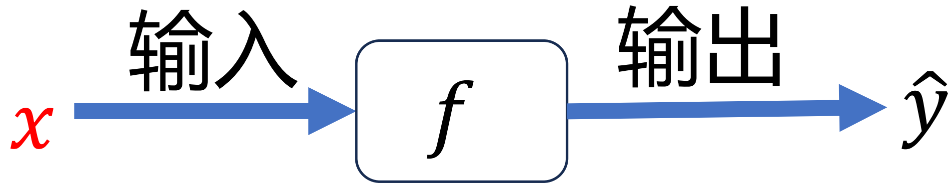
输入 → 模型 → 输出 数字3

输入 → 模型 → 输出 猫

广告投入、
价格、口碑等

输入 → 模型 → 输出 产品月销量

人工智能模型的形式化表达



➢人工智能模型可以用以下函数公式进行概括：

$$\hat{y} = f(x; \omega)$$

- x：输入
- $\hat{y}$：模型的预测输出
- $f(x; \omega)$：模型函数，描述输入x与输出$\hat{y}$之间的关系
- ω：模型参数

专业术语：特征

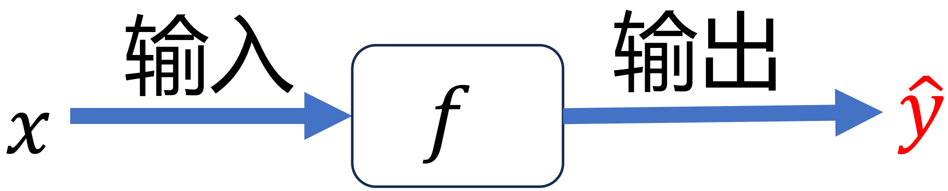
$$x \xrightarrow{\text{输入}} \boxed{f} \xrightarrow{\text{输出}} \hat{y}$$

特征 （或者属性）：

- 图片内容识别：特征是图片像素值
- 产品月销量预测：特征是广告投入、价格和口碑
- 房价预测：特征是房子面积、楼龄等信息

房价（万）	房间平方（m^2）	楼层（层）	房龄（年）	配套电梯
253	121	18	21	1
370	229	6	7	0
135	75	18	21	1
165	89	32	8	1
270	132	33	21	1
143	73	30	8	1
275	127	32	9	1

专业术语：标签

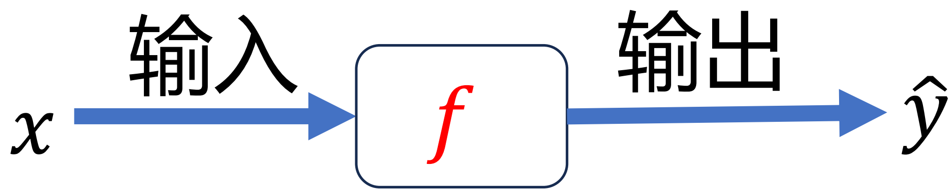


标签 (label)：

- 图片内容识别：标签是图片的内容，例如猫、狗
- 产品月销量预测：标签是产品月销量
- 房价预测：标签是房价

房价（万）	房间平方（m²）	楼层（层）	房龄（年）	配套电梯
253	121	18	21	1
370	229	6	7	0
135	75	18	21	1
165	89	32	8	1
270	132	33	21	1
143	73	30	8	1
275	127	32	9	1

专业术语：模型



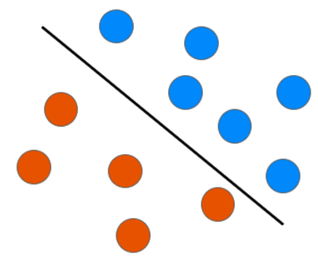
模型：

- 人工智能算法通过数据训练得到模型$f(x;\omega)$，用于在新数据进行预测
- 希望模型$f(x;\omega)$能够准确地预测标签

分类

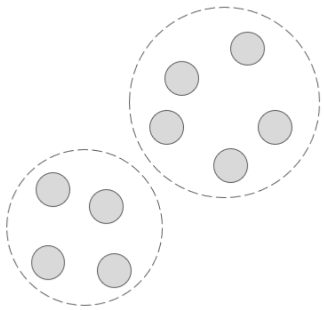
➢ 监督学习：使用带有标签的数据进行训练

- 回归：标签为连续值，例如房价预测
- 分类：标签为离散值类别，例如垃圾邮件检测



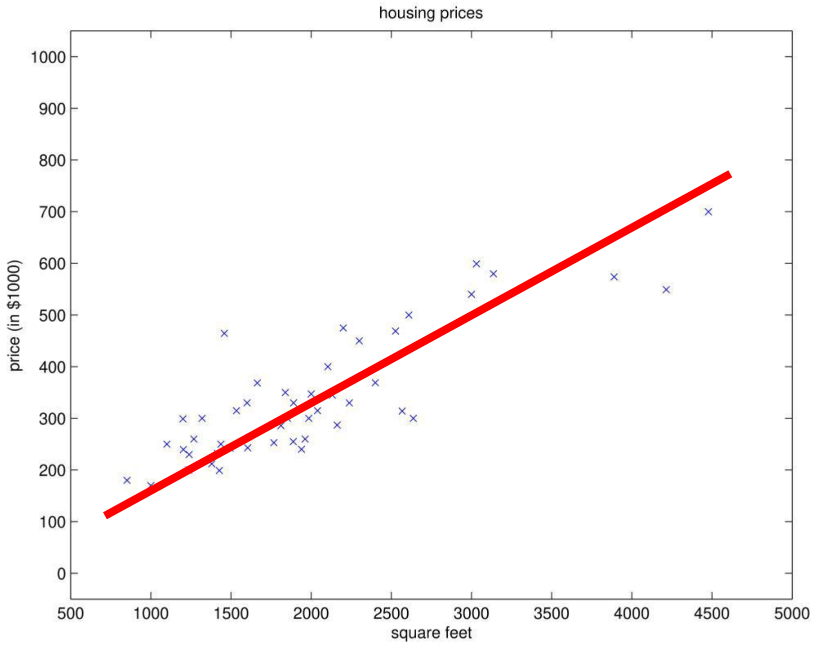
➢ 无监督学习：使用无标签的数据进行训练

- 聚类：将相似的数据点聚为一组
- 降维：减少数据维度



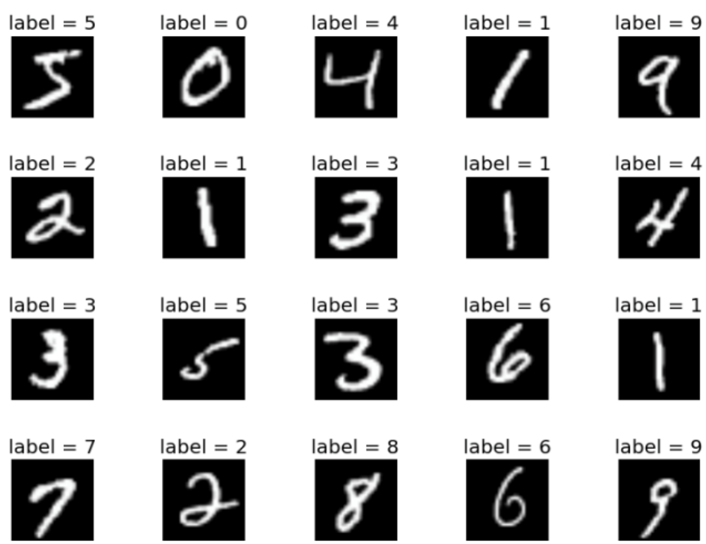
监督学习

- 回归：标签为连续值
  - 房价
  - 产品月销量



监督学习

- 分类：标签为离散值
  - 判断照片里的是哪个数字
  - 检测邮件是否为垃圾邮件
  - 判断是否患病



随堂小测 （5%的平时成绩）

给定大量动物的照片，训练模型能够识别照片中为哪种动物。此任务属于（）

A. 分类
B. 回归

随堂小测 （5%的平时成绩）

根据某公司的财务数据，预测其股票价格。
此任务属于（）

A. 分类

B. 回归

线性回归

- 一元线性回归
- 多元线性回归
- 梯度下降

举例

房价预测

产品销量预测

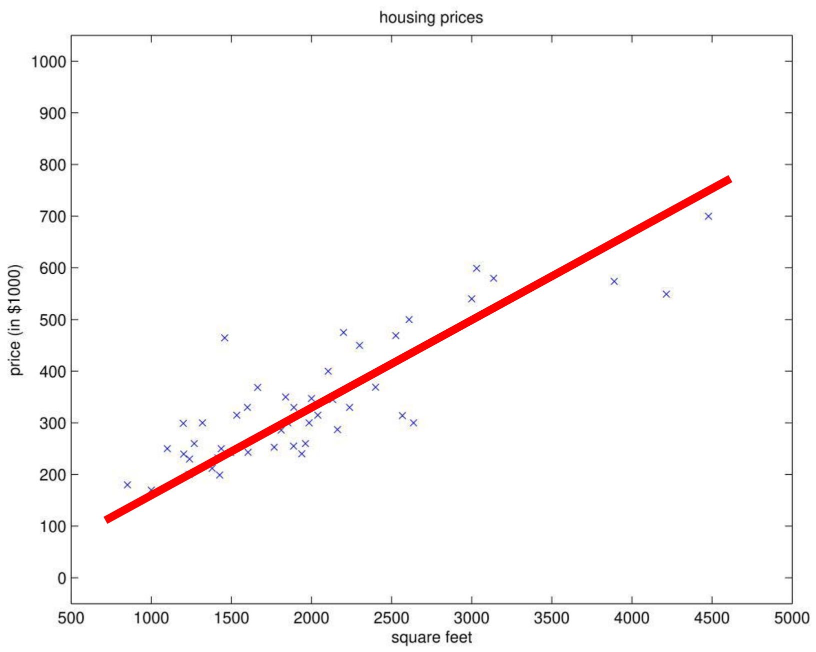
农作物产量预测

线性回归

➢ 回归：研究特征（输入）和标签（输出）之间的关系

➢ 回归模型：从特征到标签的映射函数

➢ 线性回归：假设特征和标签之间为线性关系



一元线性回归

模型表达式

$$y = \omega_0 + \omega_1 x$$

- x：特征，如房屋面积
- y：预测的标签，如房价
- ω_0：模型参数，对应截距
- ω_1：模型参数，对应斜率，表示 x每增加一个单位 y的变化量

示例：房价与房屋面积

房价（万）	房间平方（m^2）
253	121
370	229
135	75
165	89
270	132
143	73
275	127
130	73
264.5	131
130	73
120	66
258	129
185	90
182	90
165	89
278	128
107	69

随堂小测 （5%的平时成绩）

以下模型中，是关于特征x的线性模型的有（）

A. $y = \omega_0 + \omega_1 x$

B. $y = \omega_0 + \omega_1^2 x$

C. $y = \omega_0 + \omega_1 x^2$

D. $y = \omega_0 + \omega_1 x^3$

随堂小测 （5%的平时成绩）

以下模型中，是关于特征x^2的线性模型的有（）

A. $y = \omega_0 + \omega_1 x$

B. $y = \omega_0 + \omega_1^2 x^2$

C. $y = \omega_0 + \omega_1 x^2$

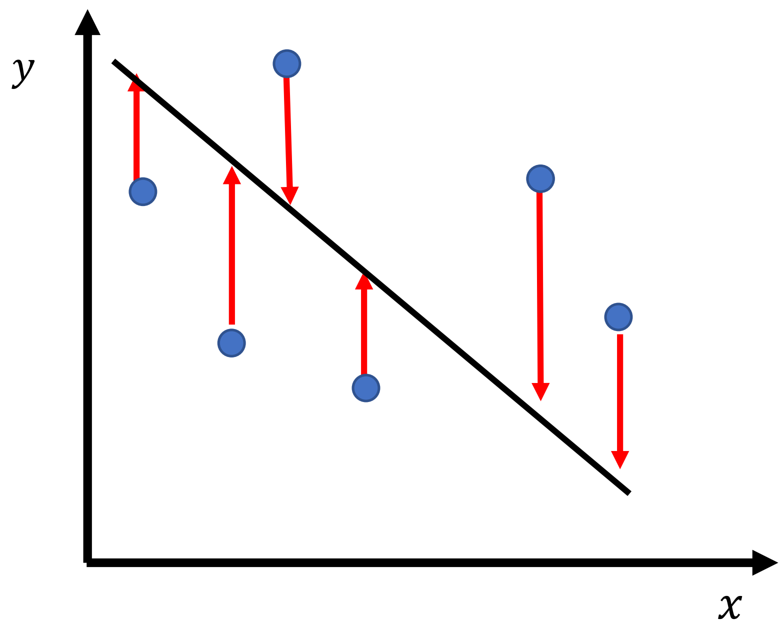
D. $y = \omega_0 + \omega_1 x^3$

一元线性回归

给定数据集$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$

线性回归的目的是学得一个模型以尽可能准确地预测真实值

$$\omega_0 + \omega_1 x^{(i)} \simeq y^{(i)},\ i = 1,2,\ldots,m$$



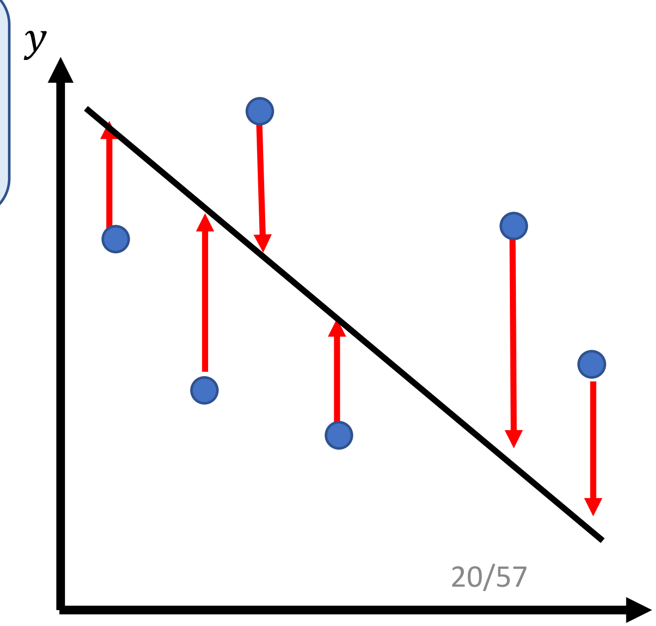
线性回归问题

给定数据集 $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$

均方误差损失函数： $L(\omega_0, \omega_1) = \frac{1}{m}\sum_{i=1}^{m}\left(\underbrace{\boxed{\omega_0 + \omega_1 x^{(i)}}}_{\text{预测值}} - \underbrace{\boxed{y^{(i)}}}_{\text{真实值}}\right)^2$

回归问题 $\min_{\omega_0, \omega_1} L(\omega_0, \omega_1) = \frac{1}{m}\sum_{i=1}^{m}\left(\omega_0 + \omega_1 x^{(i)} - y^{(i)}\right)^2$

找到(ω_0, ω_1)使得均方误差最小



最小二乘法

$L(\omega_0,\omega_1)$是关于ω_0,ω_1的凸函数，当它关于ω_0,ω_1的导数均为零时，得到ω_0,ω_1的最优解。

$$\frac{\partial L(\omega_0,\omega_1)}{\partial \omega_1}=\frac{2}{m}\left(\omega_1\sum_{i=1}^{m}x^{(i)^2}-\sum_{i=1}^{m}\left(y^{(i)}-\omega_0\right)x^{(i)}\right)=0$$

$$\frac{\partial L(\omega_0,\omega_1)}{\partial \omega_0}=\frac{2}{m}\left(m\omega_0-\sum_{i=1}^{m}\left(y_i-\omega_1x^{(i)}\right)\right)=0$$

最小二乘法

$L(\omega_0,\omega_1)$是关于ω_0,ω_1的凸函数，当它关于ω_0,ω_1的导数均为零时，得到ω_0,ω_1的最优解。

$$\omega_1 = \frac{\sum_{i=1}^{m} y^{(i)}\left(x^{(i)} - \bar{x}\right)}{\sum_{i=1}^{m} {x^{(i)}}^2 - \frac{1}{m}\left(\sum_{i=1}^{m} x^{(i)}\right)^2}$$

其中，$\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$

$$\omega_0 = \frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - \omega_1 x^{(i)}\right)$$

最终学得的一元线性回归模型为：$f(x) = \omega_0 + \omega_1 x$

线性回归

- 一元线性回归
- 多元线性回归
- 梯度下降

多元线性回归

- 房价与很多因素有关
- 产品销量跟很多因素有关

房价（万）	房间平方（m^2）	楼层（层）	房龄（年）	配套电梯
253	121	18	21	1
370	229	6	7	0
135	75	18	21	1
165	89	32	8	1
270	132	33	21	1
143	73	30	8	1
275	127	32	9	1
130	73	33	9	1
264.5	131	32	9	1
130	73	33	6	1
120	66	18	9	1
258	129	18	11	1
185	90	30	9	1
182	90	32	9	1
165	89	32	8	1
278	128	6	9	0
107	69	28	9	1

多元线性回归

模型表达式

$$y = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_d x_d$$

- $x_1, x_2, \ldots, x_d$：特征，例如包括房屋面积，楼层，楼龄
- y：想要预测的标签，如房价
- $\omega_0, \omega_1, \ldots \omega_d$：模型参数

多元线性回归

向量形式

$$
\begin{aligned}
y &= \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_d x_d \\
&= \boldsymbol{\omega}^T \boldsymbol{x}
\end{aligned}
$$

- $\boldsymbol{\omega} = \begin{bmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_d \end{bmatrix}$: $d+1$维向量
- $\boldsymbol{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}$: $d+1$维向量 $\quad x_0 = 1$

模型特点

$$
\begin{aligned}
y &= \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_d x_d \\
&= \boldsymbol{\omega}^T \boldsymbol{x}
\end{aligned}
$$

- 形式简单、易于建模
- 可解释性
- 非线性模型的基础

最小化均方误差损失函数

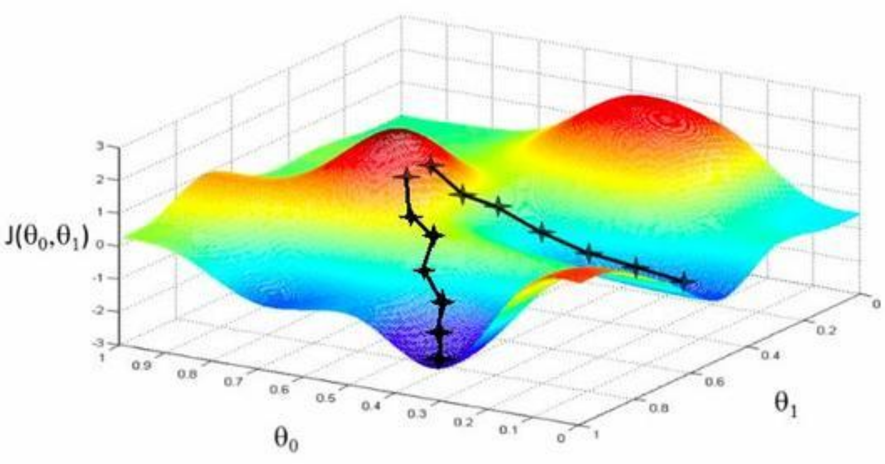
给定数据集$\{(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \cdots, (\boldsymbol{x}^{(m)}, y^{(m)})\}$

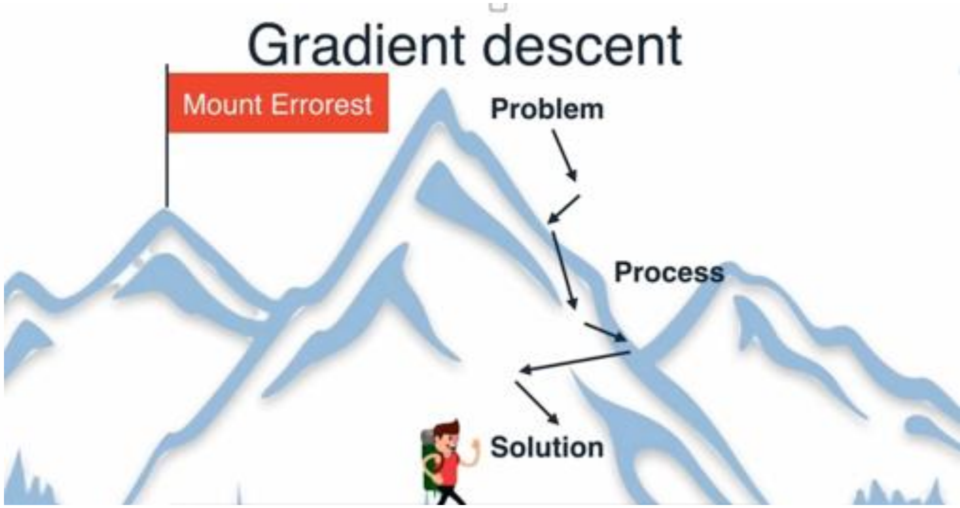
回归问题 $$\min_{\boldsymbol{\omega}} L(\boldsymbol{\omega}) = \frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{\omega}^T\boldsymbol{x}^{(i)} - y^{(i)}\right)^2$$

梯度下降法

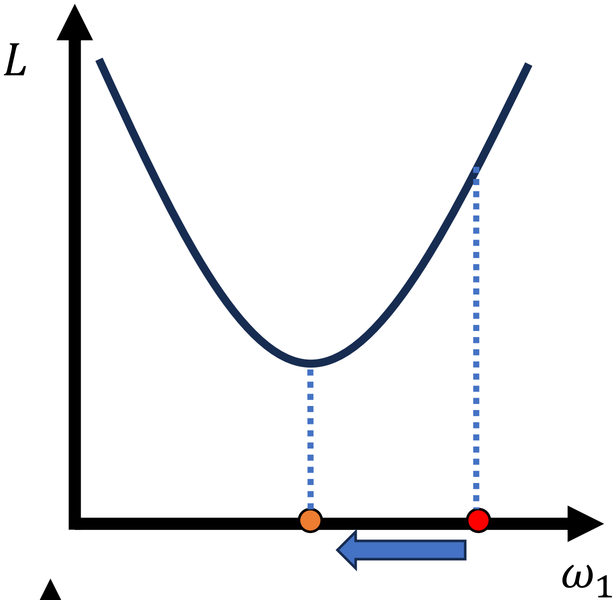
梯度下降法（gradient descent method）

- 经典的数值优化算法：
- 迭代算法



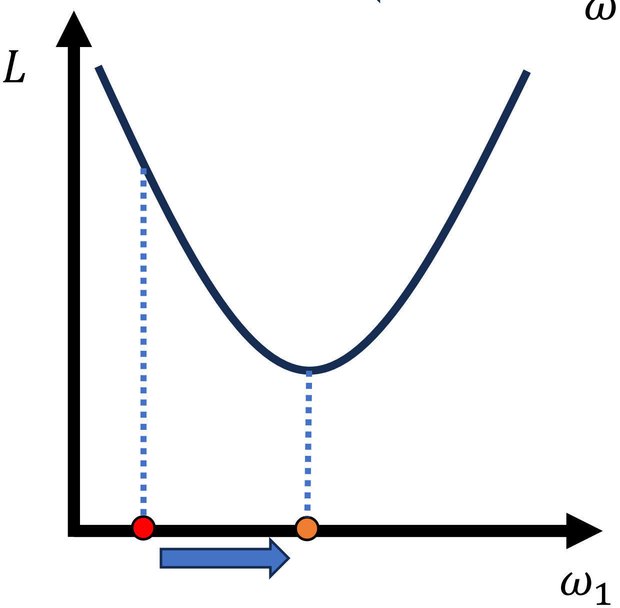


梯度下降法



ω_1在极值点右边， $\frac{\partial L}{\partial \omega_1} > 0$

需要减少ω_1



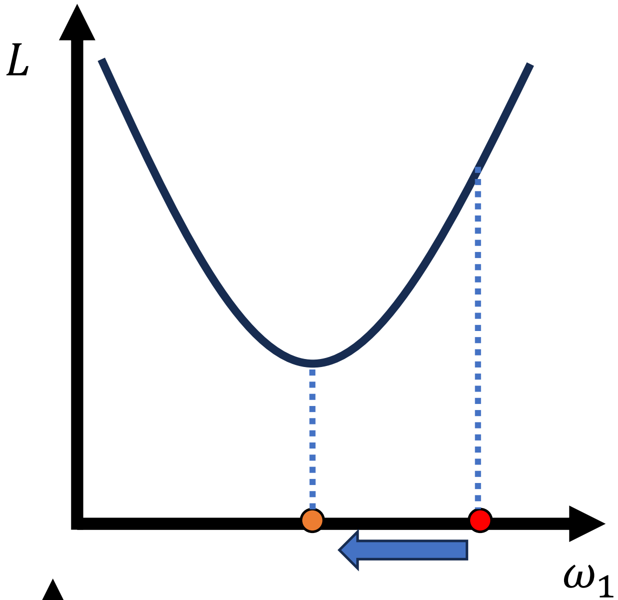
ω_1在极值点左边， $\frac{\partial L}{\partial \omega_1} < 0$

需要增大ω_1

梯度下降法

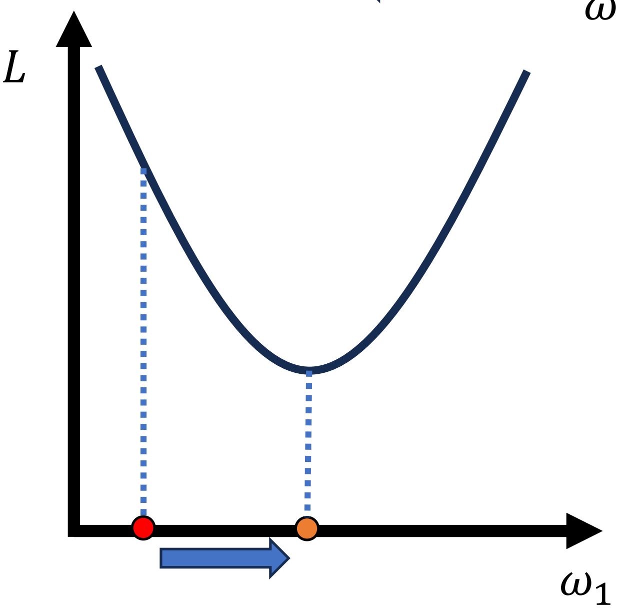
更新公式：$\omega_1 = \omega_1 - \alpha \frac{\partial L}{\partial \omega_1}$

- α为步长（学习率），为很小的正数



ω_1在极小点右边， $\frac{\partial L}{\partial \omega_1} > 0$

$\omega_1 = \omega_1 - \alpha \frac{\partial L}{\partial \omega_1}$使得$\omega_1$左移



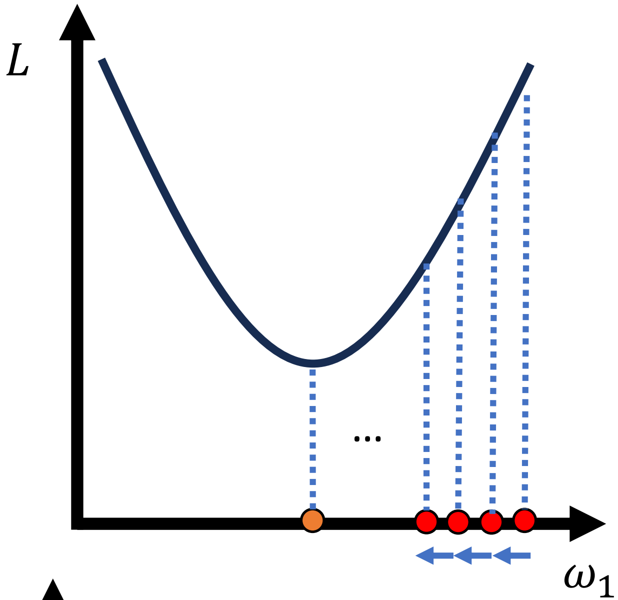
ω_1在极小点左边， $\frac{\partial L}{\partial \omega_1} < 0$

$\omega_1 = \omega_1 - \alpha \frac{\partial L}{\partial \omega_1}$使得$\omega_1$右移

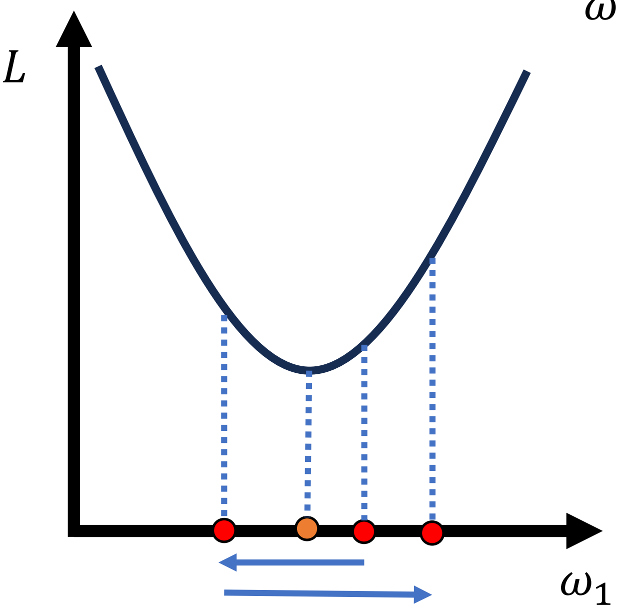
步长

更新公式： $\omega_1 \leftarrow \omega_1 - \alpha \dfrac{\partial L}{\partial \omega_1}$

- α为步长（学习率），为很小的正数



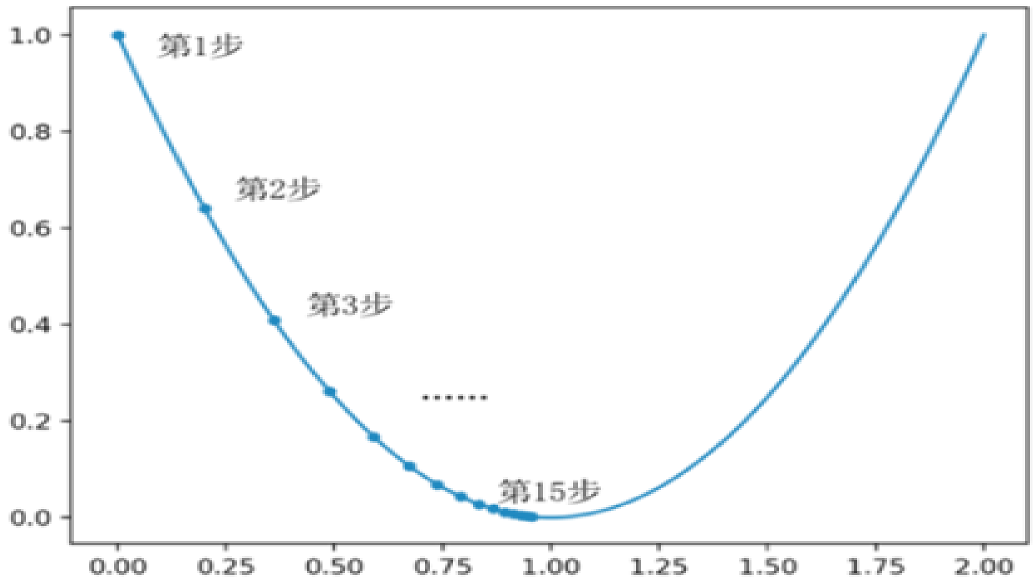
若步长α太小，则收敛速度较慢



若步长α太大，则难以收敛

梯度下降法

- 梯度下降算法从空间中任一给定初始点开始进行迭代
- 在每一次迭代中，计算目标函数在当前点的梯度，并沿着与梯度相反的方向按照一定步长移动到下一可行点。



梯度

$$L(\boldsymbol{\omega}) = \frac{1}{2m}\sum_{i=1}^{m}\left(\boldsymbol{\omega}^T \boldsymbol{x}^{(i)} - y^{(i)}\right)^2$$

损失函数对ω_j的偏导为

$$\frac{\partial L}{\partial \omega_j} = \frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{\omega}^T x^{(i)} - y^{(i)}\right) x_j^{(i)} \qquad j = 0,1,2,\ldots,d$$

梯度 $\nabla L = \begin{bmatrix} \frac{\partial L}{\partial \omega_0} \\ \frac{\partial L}{\partial \omega_1} \\ \vdots \\ \frac{\partial L}{\partial \omega_d} \end{bmatrix}$

随堂小测 （5%的平时成绩）

$L(\omega_0,\omega_1)=\frac{1}{2m}\sum_{i=1}^{m}\left(\omega_0+\omega_1 x_1^{(i)}+\omega_2 x_2^{(i)}-y^{(i)}\right)^2$关于$\omega_0,\omega_1,\omega_2$的偏导是（）

A. $\frac{\partial L}{\partial \omega_0}=\frac{1}{m}\sum_{i=1}^{m}\left(\omega_0+\omega_1 x_1^{(i)}+\omega_2 x_2^{(i)}-y^{(i)}\right),$

$\frac{\partial L}{\partial \omega_1}=\frac{1}{m}\sum_{i=1}^{m}\left(\omega_0+\omega_1 x_1^{(i)}+\omega_2 x_2^{(i)}-y^{(i)}\right)\omega_1$

$\frac{\partial L}{\partial \omega_2}=\frac{1}{m}\sum_{i=1}^{m}\left(\omega_0+\omega_1 x_1^{(i)}+\omega_2 x_2^{(i)}-y^{(i)}\right)\omega_2$

B. $\frac{\partial L}{\partial \omega_0}=\frac{1}{m}\sum_{i=1}^{m}\left(\omega_0+\omega_1 x_1^{(i)}+\omega_2 x_2^{(i)}-y^{(i)}\right),$

$\frac{\partial L}{\partial \omega_1}=\frac{1}{m}\sum_{i=1}^{m}\left(\omega_0+\omega_1 x_1^{(i)}+\omega_2 x_2^{(i)}-y^{(i)}\right)x_1^{(i)}$

$\frac{\partial L}{\partial \omega_2}=\frac{1}{m}\sum_{i=1}^{m}\left(\omega_0+\omega_1 x_1^{(i)}+\omega_2 x_2^{(i)}-y^{(i)}\right)x_2^{(i)}$

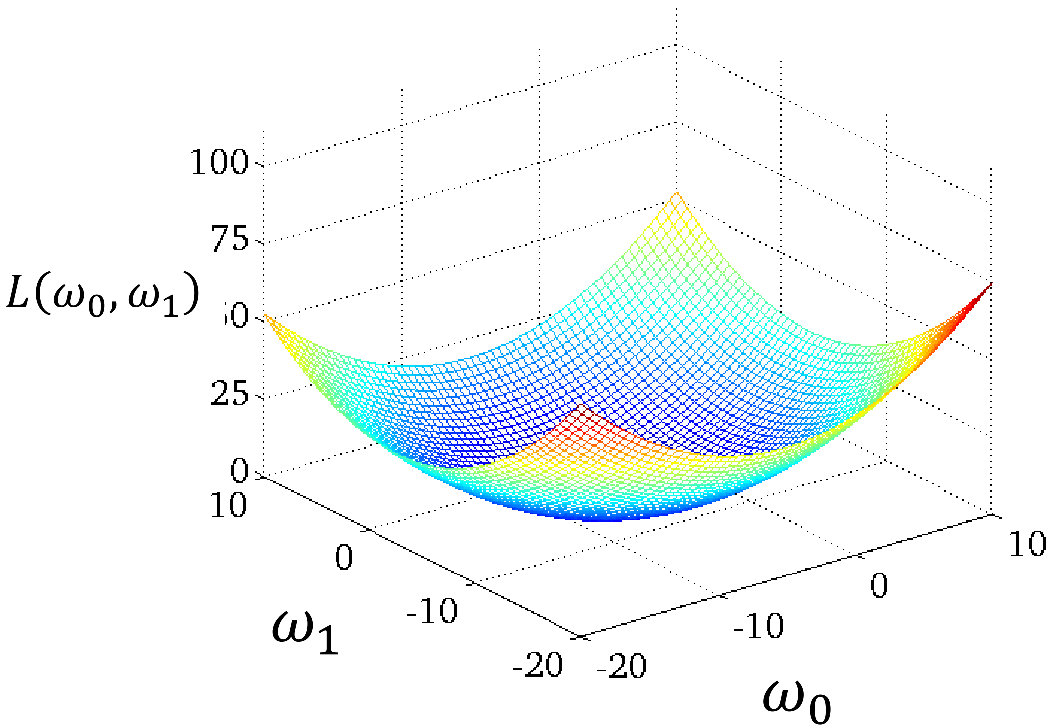
线性回归：梯度下降法

- 初始化参数$\boldsymbol{\omega}$
- 重复以下更新直至满足停止条件（如参数或者损失值的变化小于某阈值）

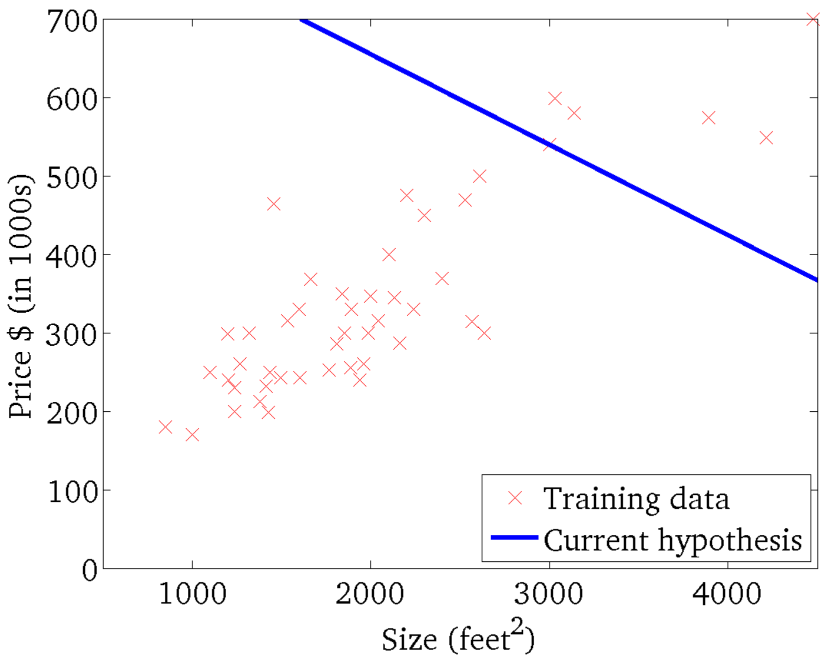
  更新各个参数$\omega_0, \omega_1, \ldots, \omega_d$
  $$\omega_j = \omega_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(\boldsymbol{\omega}^T x^{(i)} - y^{(i)} \right) x_j^{(i)}$$

- 返回参数$\boldsymbol{\omega}$

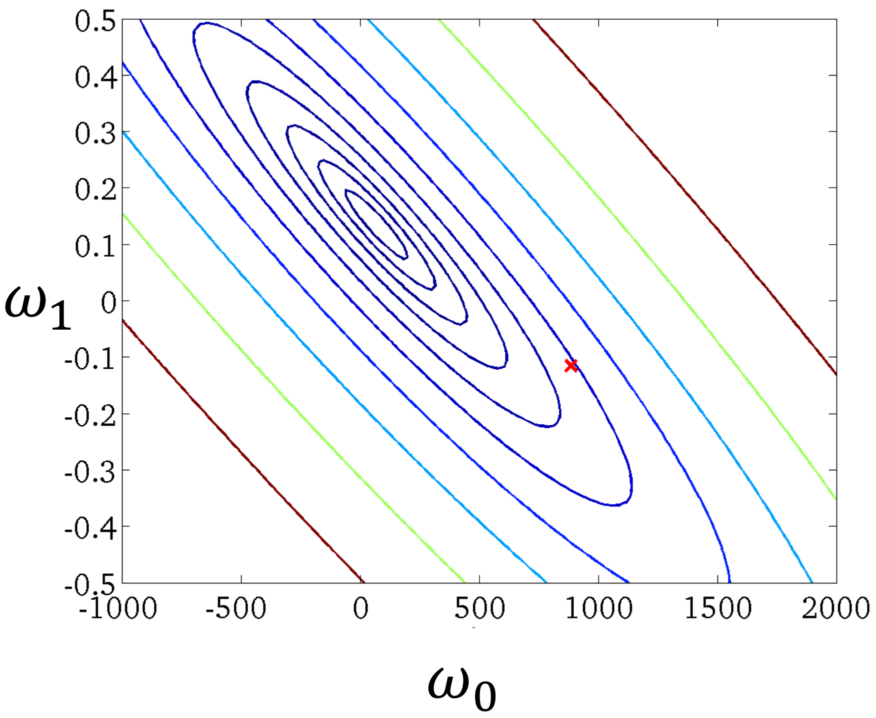
梯度下降法



梯度下降法

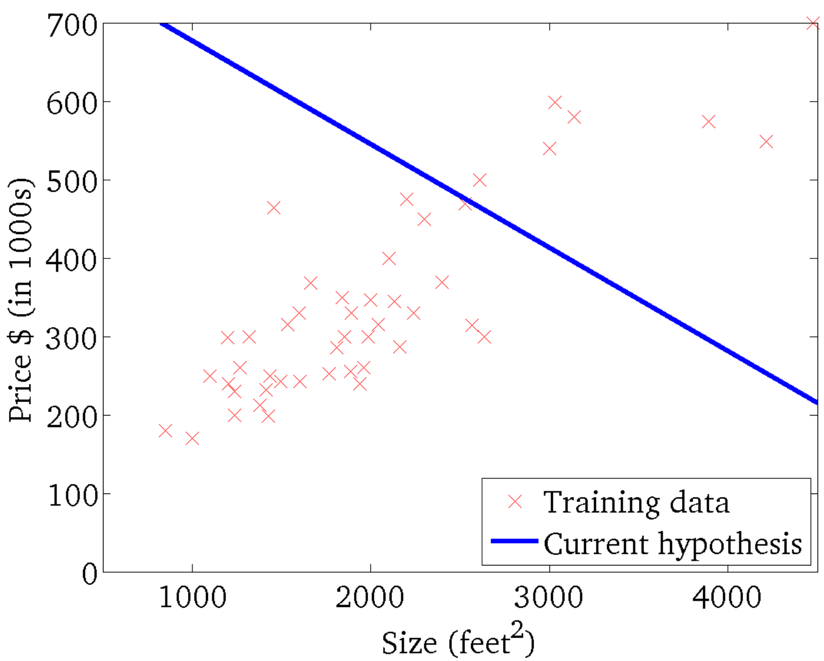


Price $ (in 1000s)
Size (feet^2)
0
100
200
300
400
500
600
700
1000
2000
3000
4000
Training data
Current hypothesis

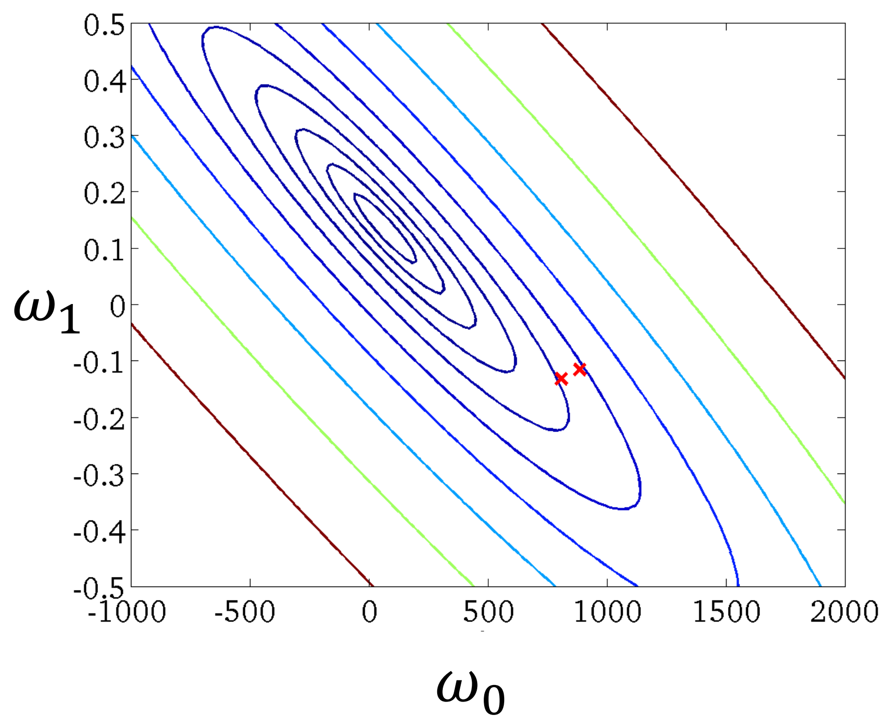


ω_1
ω_0
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1000
-500
0
500
1000
1500
2000

梯度下降法

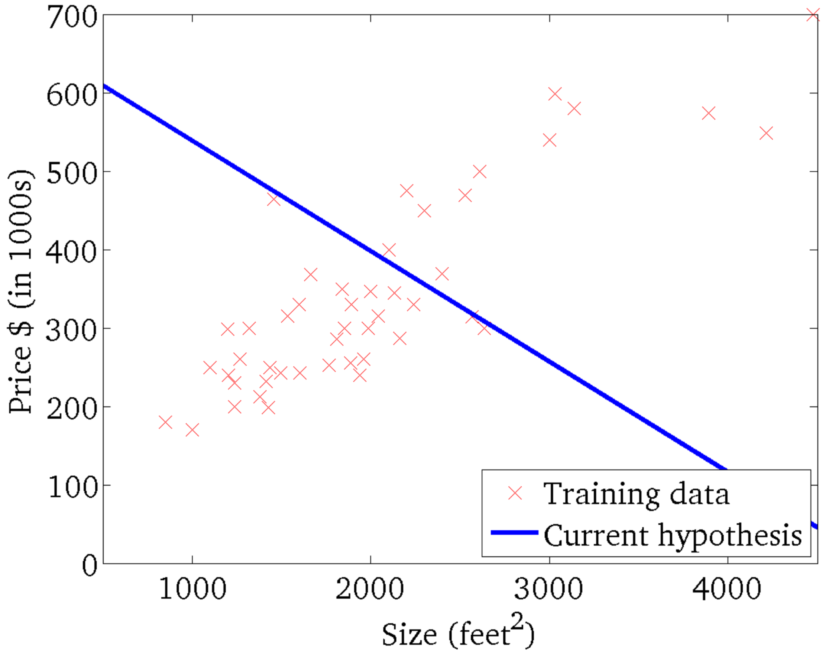


700
600
500
400
300
200
100
0
1000
2000
3000
4000
Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis

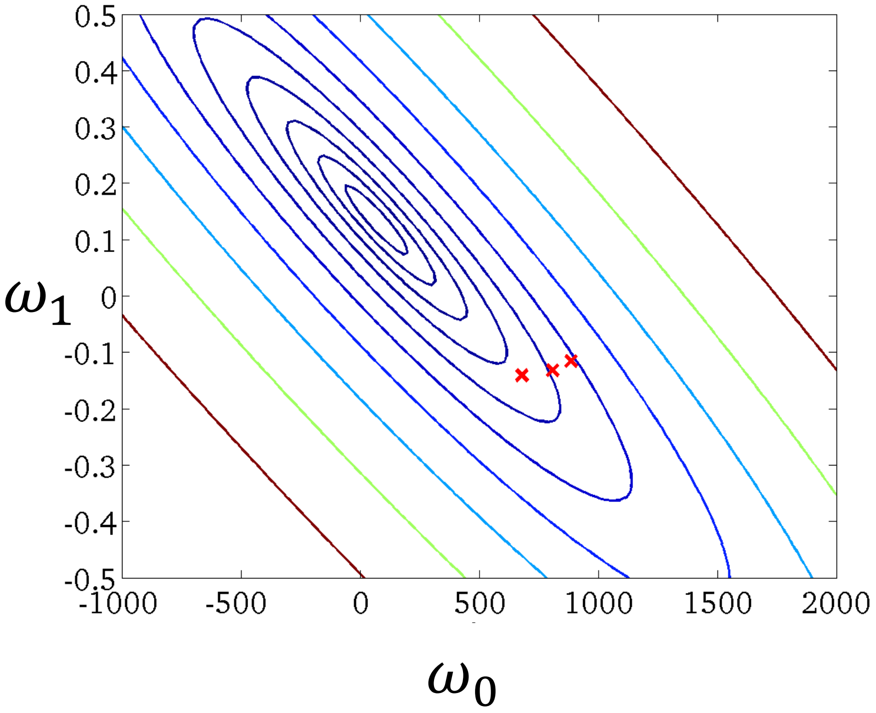


0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1000
-500
0
500
1000
1500
2000
ω_1
ω_0

梯度下降法

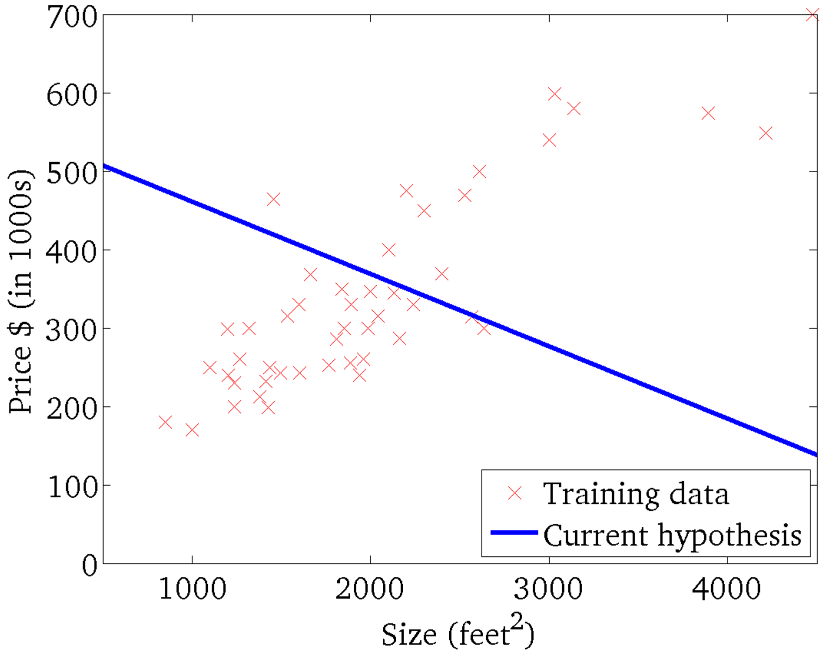


700
600
500
400
300
200
100
0
Price $ (in 1000s)
1000
2000
3000
4000
Size (feet^2)
Training data
Current hypothesis

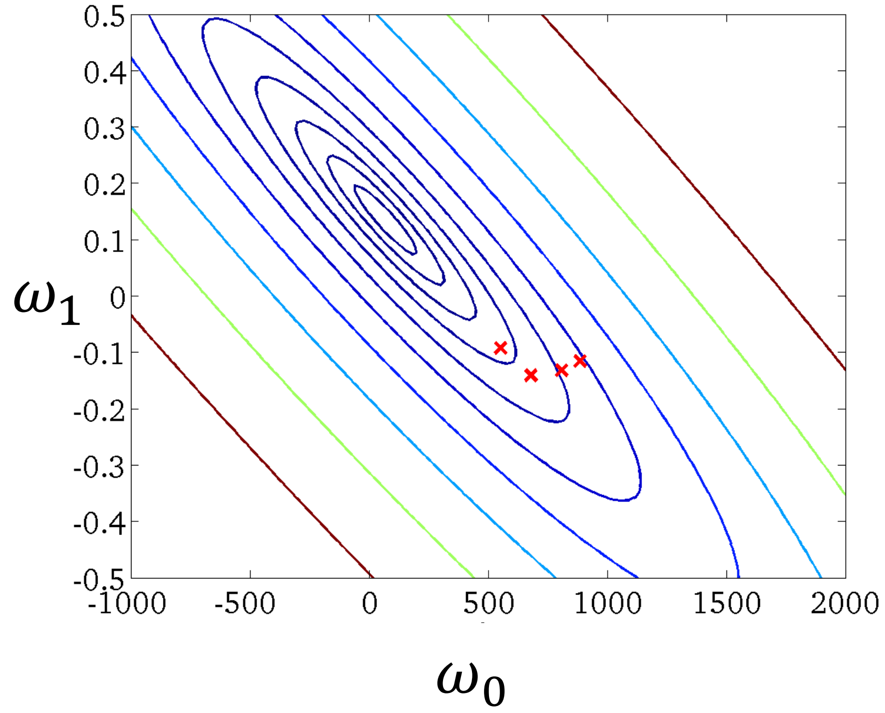


0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
ω_1
-1000
-500
0
500
1000
1500
2000
ω_0

梯度下降法

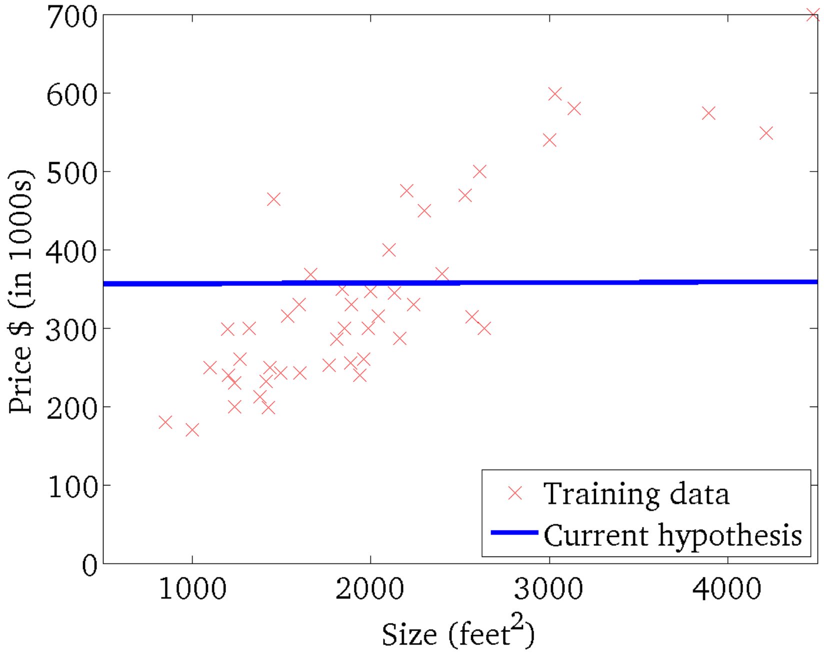


Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis

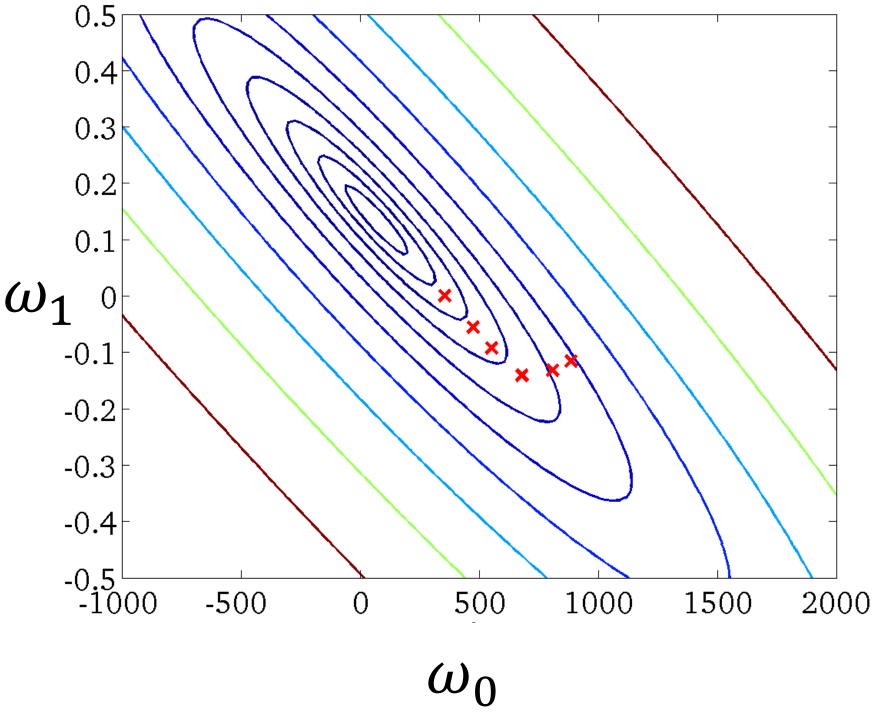


ω_1
ω_0

梯度下降法

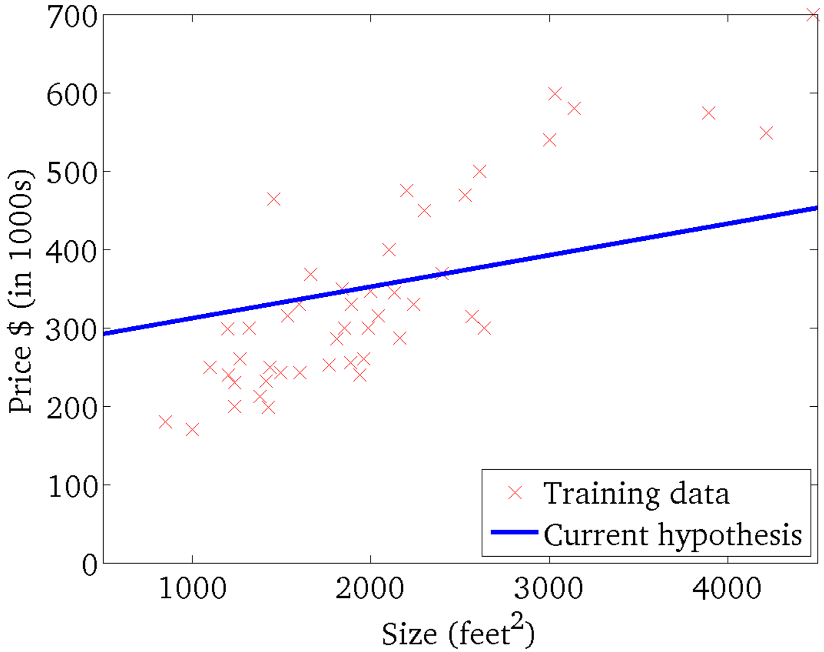


Training data
Current hypothesis
Price $ (in 1000s)
Size (feet^2)
0
100
200
300
400
500
600
700
1000
2000
3000
4000

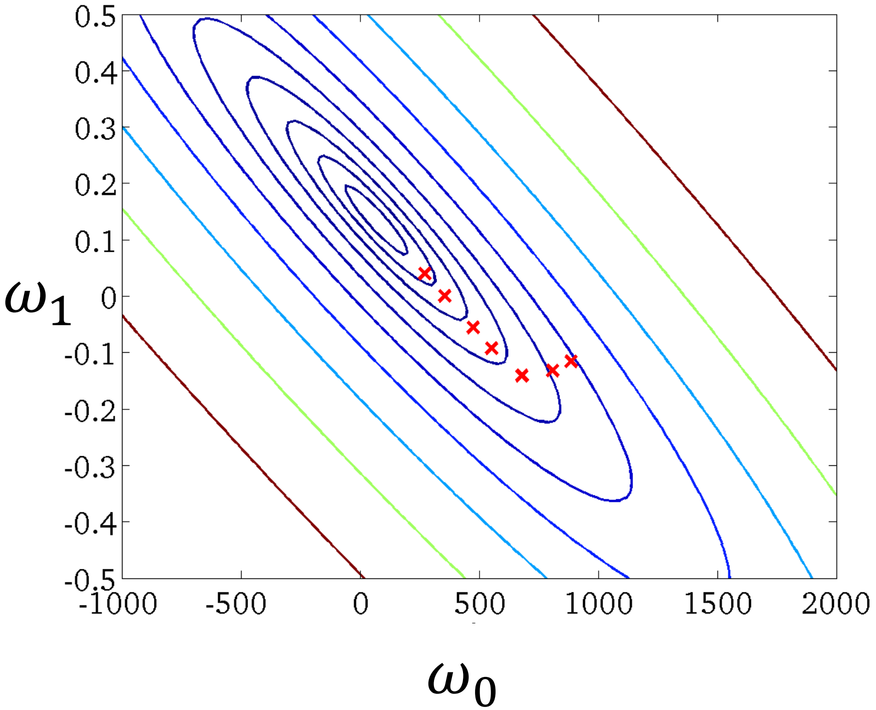


ω_1
ω_0
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1000
-500
0
500
1000
1500
2000

梯度下降法

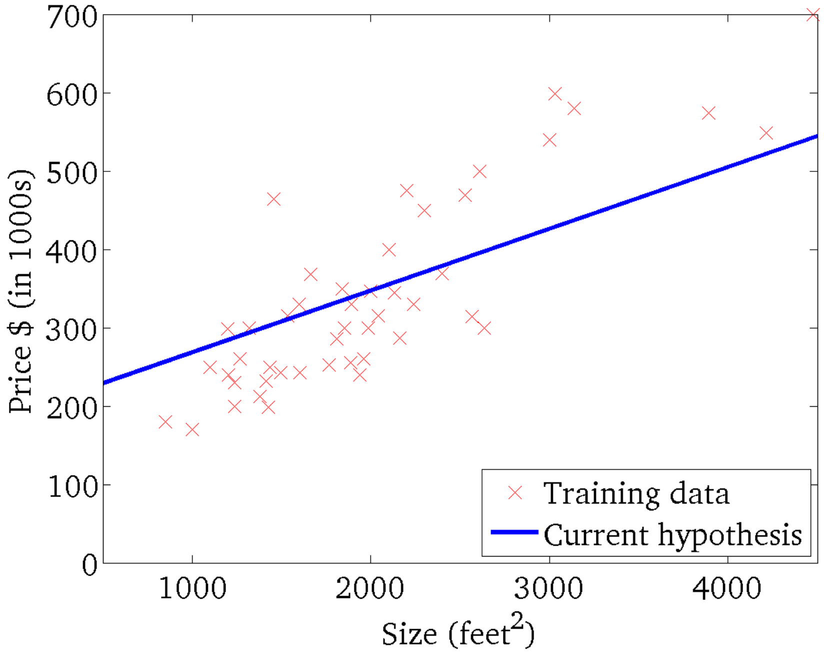


Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis
0
100
200
300
400
500
600
700
1000
2000
3000
4000

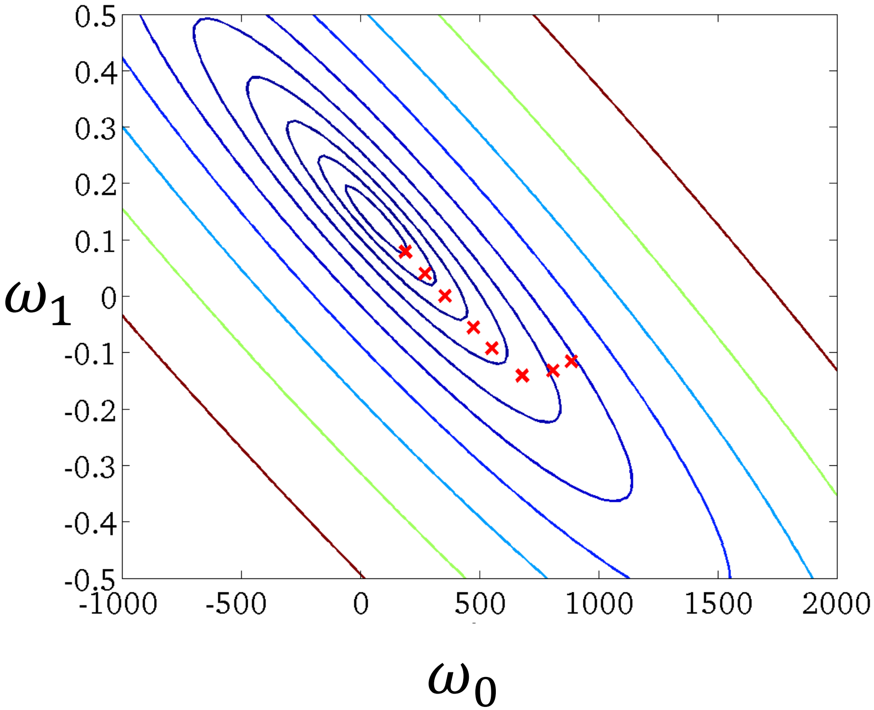


ω_1
ω_0
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1000
-500
0
500
1000
1500
2000

梯度下降法

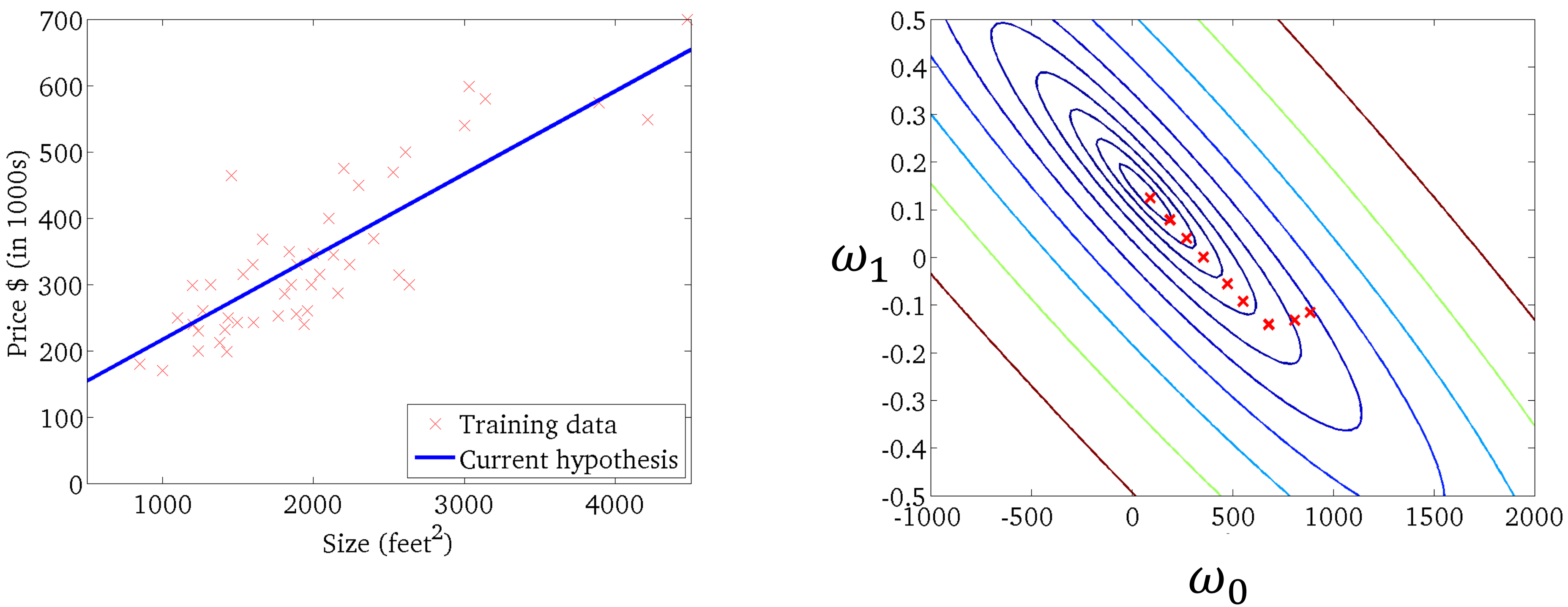


700
600
500
400
300
200
100
0
Price $ (in 1000s)
1000
2000
3000
4000
Size (feet^2)
Training data
Current hypothesis



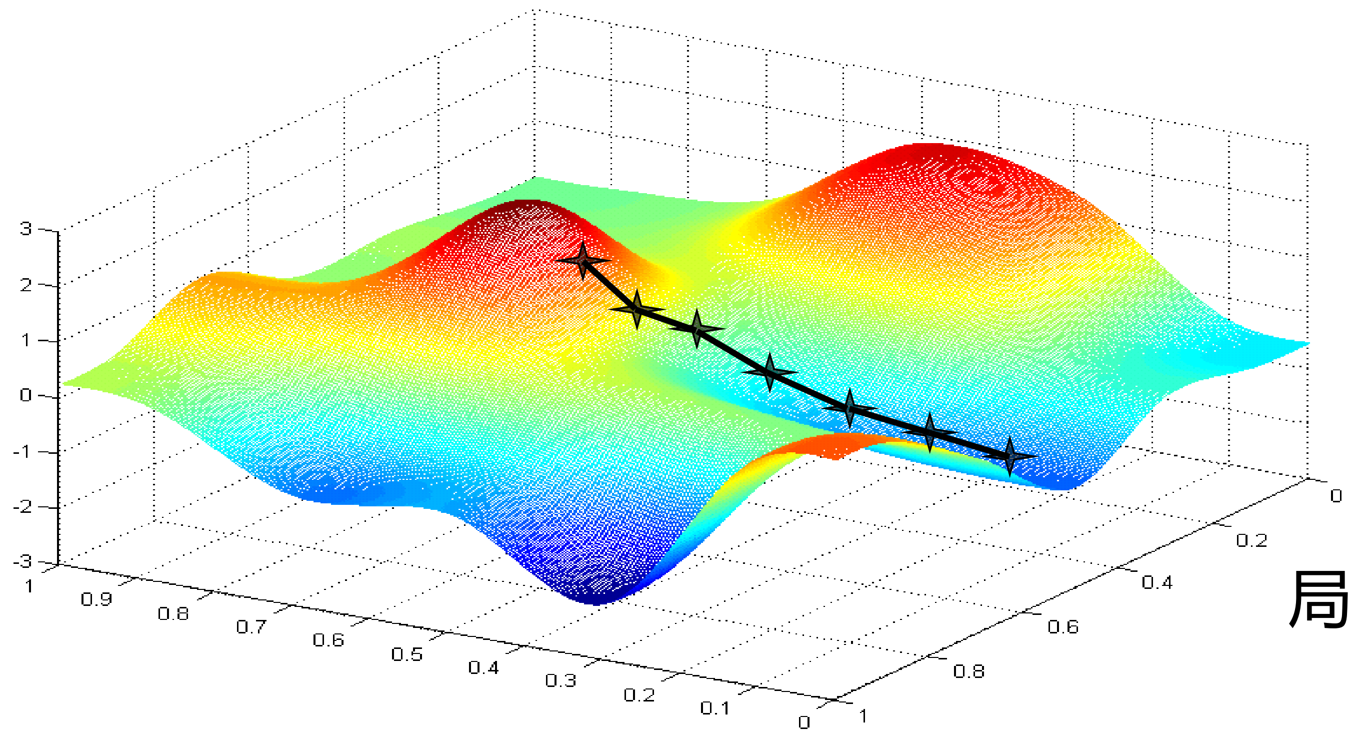
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
ω_1
-1000
-500
0
500
1000
1500
2000
ω_0

梯度下降法



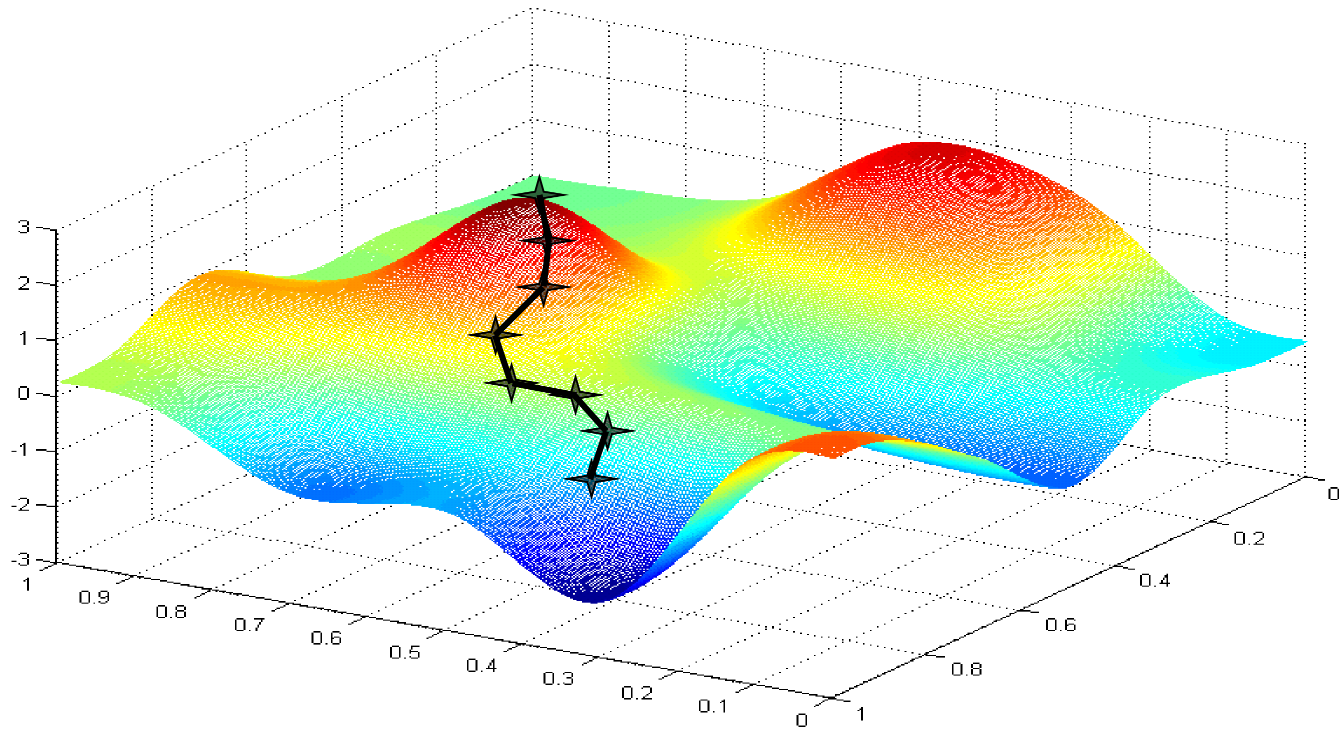
700
600
500
400
300
200
100
0
1000
2000
3000
4000
Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1000
-500
0
500
1000
1500
2000
ω_1
ω_0

梯度下降法



局部极值

梯度下降法



局部极值

随堂小测 （5%的平时成绩）

以下关于梯度下降法的描述错误的是（）

A. 可用于求解一元线性回归问题
B. 沿着负梯度方向更新
C. 梯度下降法返回的是局部最小值点
D. 梯度下降法返回的是全局最小值点

随堂小测 （5%的平时成绩）

函数值上升最快的方向是（）

A. 梯度方向

B. 负梯度方向

梯度下降法

根据所使用样本数量的不同，梯度下降法分为：

- 批量梯度下降法
- 随机梯度下降法
- 小批量梯度下降法

批量梯度下降法

- 所有的样本都有贡献
- 可以达到一个全局最优
- 样本多的情况下收敛的速度慢

$$\omega_j = \omega_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(\boldsymbol{\omega}^T x^{(i)} - y^{(i)} \right) x_j^{(i)}$$

随机梯度下降法

- 在每次更新时用1个样本
- 计算得到的并不是准确的一个梯度
- 整体的方向是全局最优解的方向，最终的结果往往在全局最优解附近
- 方法更快，更快收敛

随机采样一个样本$\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)$:

$$\omega_j = \omega_j - \alpha\left(\boldsymbol{\omega}^T \boldsymbol{x}^{(i)} - y^{(i)}\right) x_j^{(i)}$$

小批量梯度下降法

- 批量梯度下降与随机梯度下降的结合
- 将所有数据分割成 K个小批量
- 每次迭代使用小批量的数据做更新

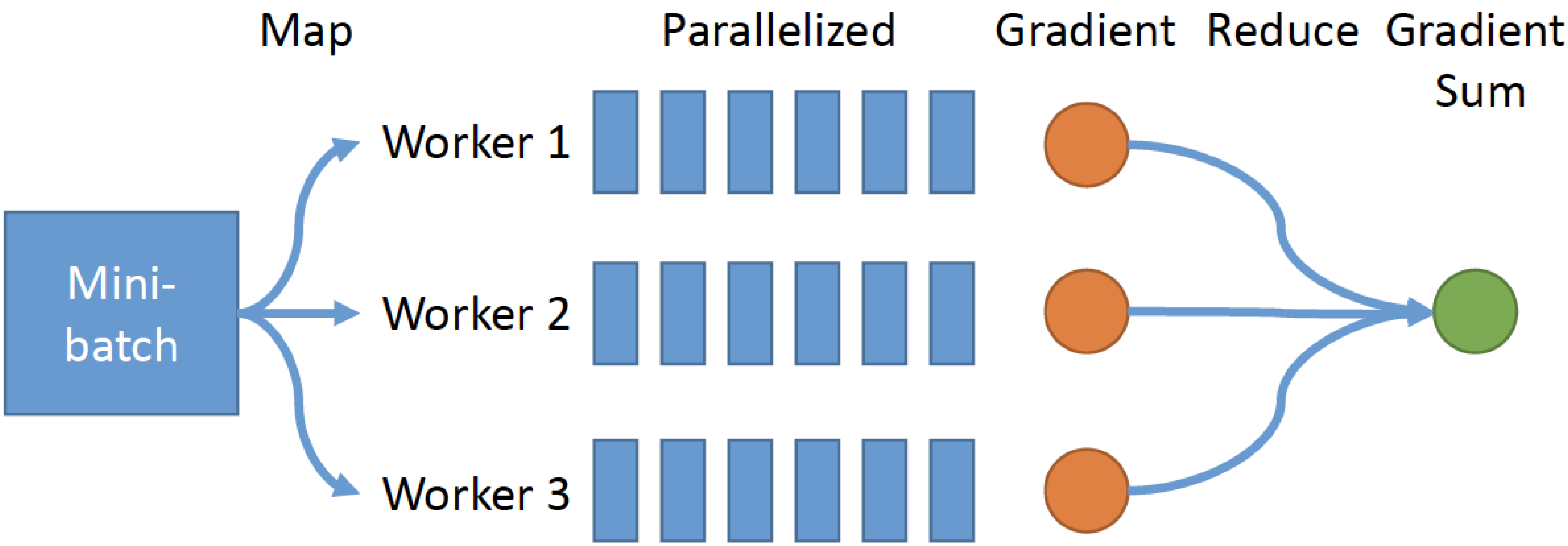
对于第k个小批量

$$\omega_j = \omega_j - \alpha \frac{1}{N} \sum_{i=1}^{N} \left(\boldsymbol{\omega}^T \boldsymbol{x}^{(i)} - y^{(i)} \right) x_j^{(i)}$$

N为小批量数据的数量

比较

- ➢批量梯度下降：学习稳定性较好
- ➢随机梯度下降：计算速度较快
- ➢小批量梯度下降: 两者折中，适合并行化设计
  - 每一个机器负责一个小批量数据



随堂小测 （5%的平时成绩）

哪种梯度下降算法的内存占用最大()

A. 批量梯度下降

B. 随机梯度下降

C. 小批量梯度下降

随堂小测 （5%的平时成绩）

以下关于小批量梯度下降算法的描述，错误的是（）

A. 小批量梯度下降结合了批量梯度下降的稳定性和随机梯度下降的效率

B. 适合在数据集很大的时候使用

C. 小批量梯度下降的路径比随机梯度下降的平滑

D. 以上说法都是错误

总结

- 线性回归模型

$$
\begin{aligned}
y &= \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_d x_d \\
&= \boldsymbol{\omega}^T \boldsymbol{x}
\end{aligned}
$$

- 最小二乘法求解和梯度下降法求解
- 批量梯度下降、随机梯度下降和小批量梯度下降