机器学习与数据挖掘

线性回归

课程大纲

- 引言
- 单特征情况
- 多特征情况
- 数值优化

Introduction

- 什么是回归？

  基于给定的特征，预测感兴趣变量的值

- 示例：房价预测

特征　　　　　　　　　　　　　　　　感兴趣的变量

大小（英尺） x_1	# 卧室数 x_2	# 楼层 x_3	# 房龄（年） x_4	价格 (\$ 1000) y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
….	….	….	….	….

特征: 1) 大小; 2) # 卧室数; 3) # 楼层; 4) # 房龄

- 从数学上讲，回归旨在学习一个函数 $f(\cdot)$ ，用以对输入数据 x 和输出值 y 之间的关系进行建模

$$\hat{y} = f(x_1,\ x_2,\ x_3, x_4)$$

- 线性回归

将函数族 $f(\cdot)$ 限制为线性形式，即：

$$f(x_1, x_2 \cdots x_m) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_m x_m$$

  - w_k: 模型参数
  - m: 特征数量

- 目标

找到一组参数 $\{w_k\}_{k=1}^{m}$ 使得预测值

$$\hat{y} = f(x_1, x_2, \cdots, x_m)$$

在训练集中，尽可能接近所有数据样本的真实值 y

	大小（英尺）x_1	# 卧室数 x_2	# 楼层 x_3	# 房龄（年）x_4	价格 ($ 1000) y
样本 1	2104	5	1	45	460
样本 2	1416	3	2	40	232
样本 3	1534	3	2	30	315
样本 4	852	2	1	36	178
	….	….	….	….	….

课程大纲

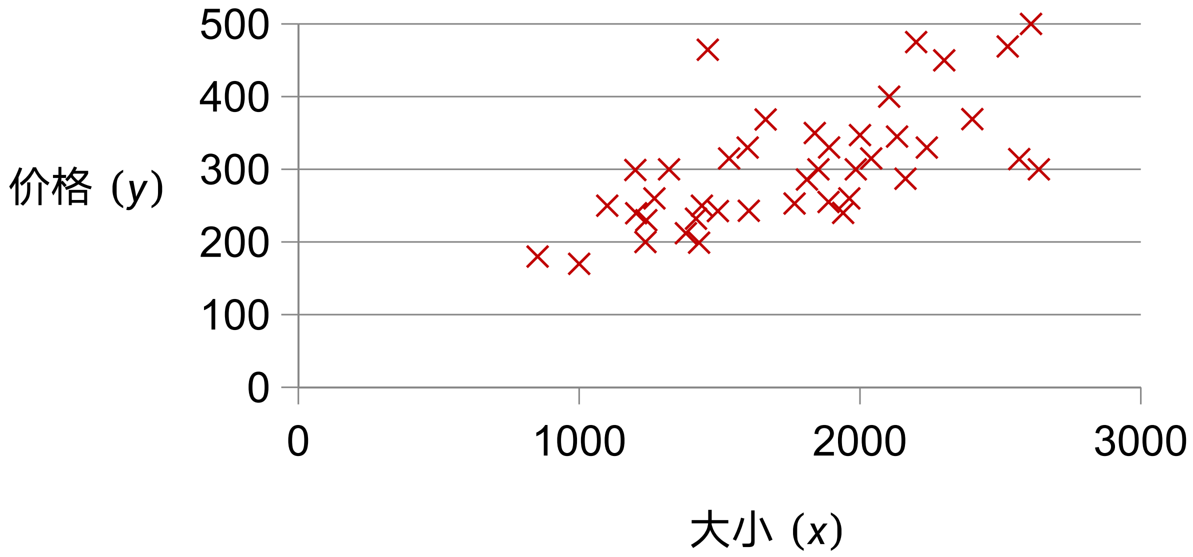
- 引言
- 单特征情况
- 多特征情况
- 数值优化

模型

- 为了简化，我们首先只考虑一个特征
- 例如：房屋大小

大小（英尺） x	价格 ($ 1000) y
2104	460
1416	232
852	178
….	….

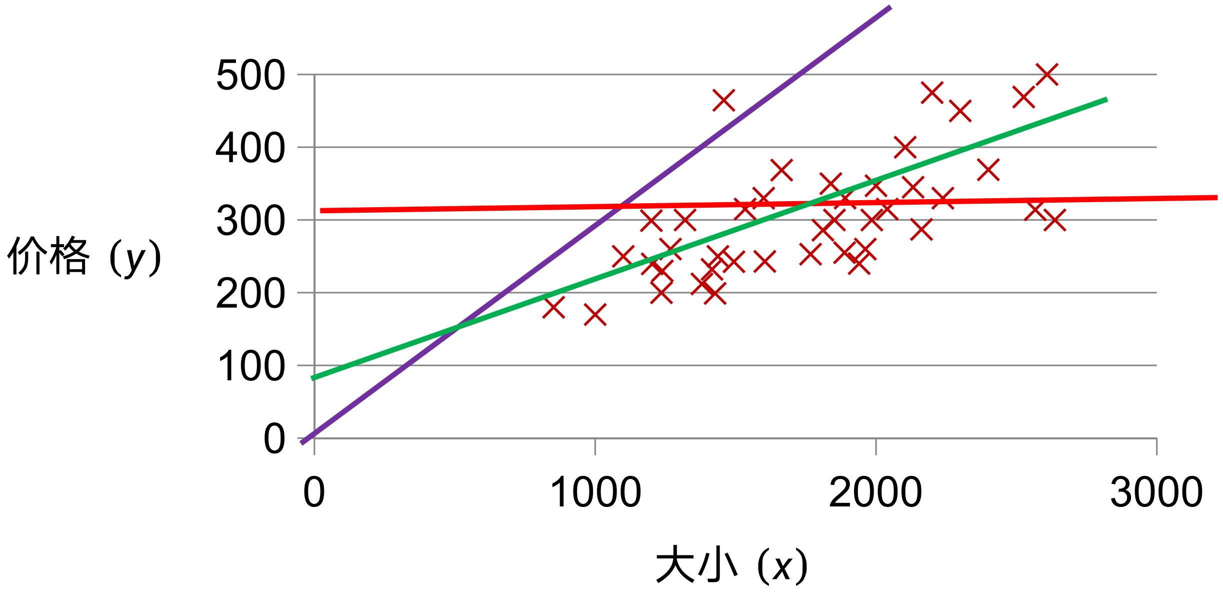
- 将 (x, y) 数据对绘制在平面上



- 预测函数简化为：

$$f(x) = w_0 + w_1 x$$

- 对于不同的 w_0 和 w_1 值，函数 $f(x)$ 代表不同的直线



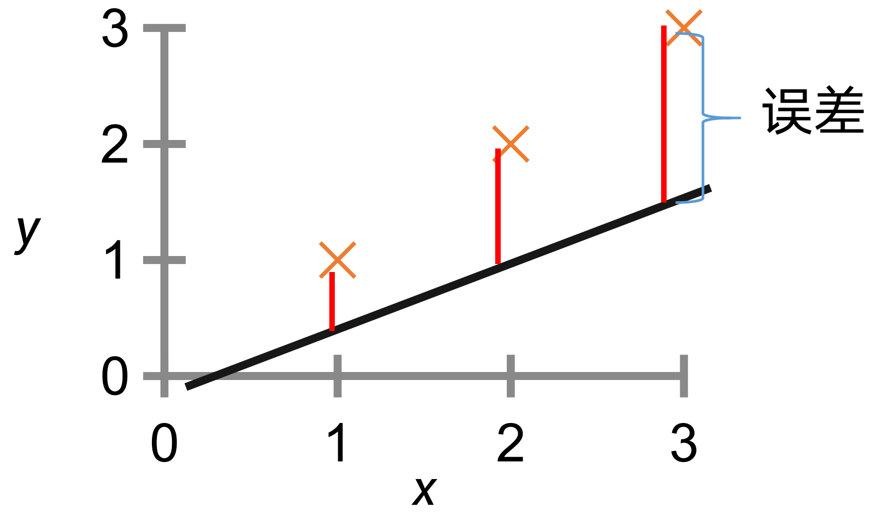
- 目标是找到一组合适的 w_0 和 w_1，使得这条直线尽可能地拟合所有给定 x 对应的真实 y 值

代价/损失函数

- 从数学上讲，我们的目标可以被表述为最小化代价（损失）函数

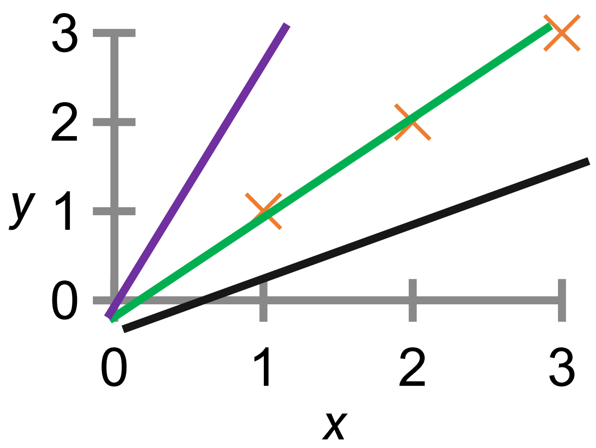
$$L(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}\left(f\left(x^{(i)}\right) - y^{(i)}\right)^2$$

其中 $x^{(i)}$ 和 $y^{(i)}$ 分别表示第 i 个特征和目标值；n 是训练样本的数量



- 将 $f\left(x^{(i)}\right)=w_0+w_1x^{(i)}$ 代入 $L(w_0,w_1)$ 得到：

$$L(w_0,w_1)=\frac{1}{n}\sum_{i=1}^{n}\left(w_0+w_1x^{(i)}-y^{(i)}\right)^2$$



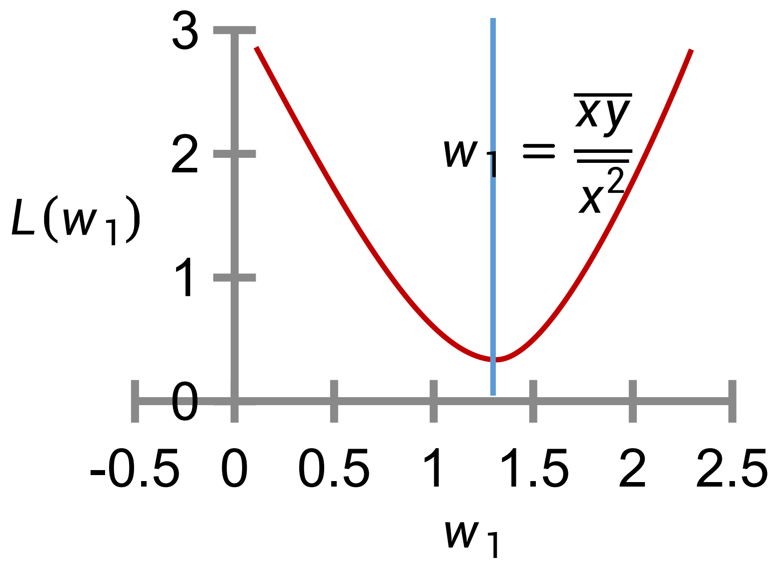
备注：为了更好地理解这个代价函数，我们通过令 $w_0=0$ 来简化它

- 然后，代价函数变为：

$$\boxed{L(w_1)=\overline{x^2}w_1^2-2\overline{xy}w_1+\overline{y^2}}$$

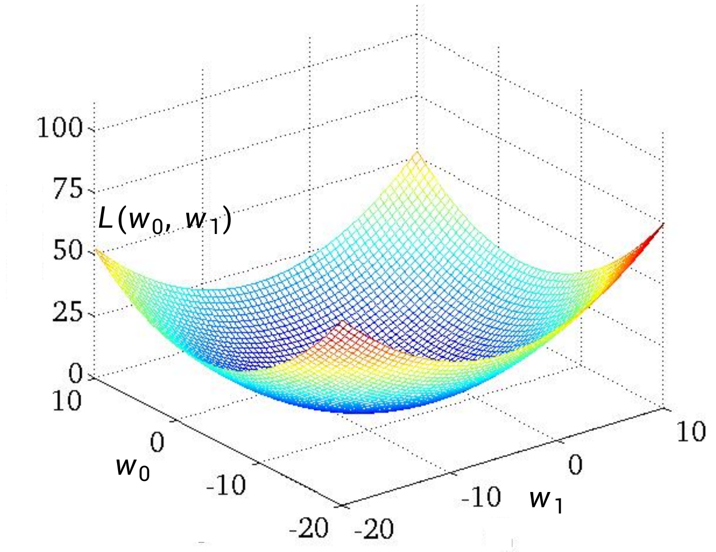
其中 $\overline{x^2}=\frac{\sum_{i=1}^{n}\left(x^{(i)}\right)^2}{n}$, $\overline{xy}=\frac{\sum_{i=1}^{n}x^{(i)}y^{(i)}}{n}$ 且 $\overline{y^2}=\frac{\sum_{i=1}^{n}\left(y^{(i)}\right)^2}{n}$

- 代价函数是关于 w_1 的二次函数



$$L(w_1) = \overline{x^2}w_1^2 - 2\overline{xy}w_1 + \overline{y^2}$$

- 如果将 w_0 也考虑在内，代价函数 $L(w_0, w_1)$ 仍然是一个二次函数，但变成了二维的



$$L(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}\left(w_0 + w_1 x^{(i)} - y^{(i)}\right)^2$$

- 最优的 w_0 和 w_1 可以通过将导数置为零来找到

$$\frac{\partial L}{\partial w_0} = \frac{2}{n}\sum_{i=1}^{n}\left(w_0 + w_1 x^{(i)} - y^{(i)}\right) = 0$$

$$\frac{\partial L}{\partial w_1} = \frac{2}{n}\sum_{i=1}^{n}\left(w_0 + w_1 x^{(i)} - y^{(i)}\right) x^{(i)} = 0$$

- 通过求解该线性方程组，可以得到最优的 w_0 和 w_1

$$w_0 = \frac{\overline{xy}\,\bar{x} - \overline{x^2}\,\bar{y}}{\bar{x}^2 - \overline{x^2}} \qquad w_1 = \frac{\bar{x}\bar{y} - \overline{xy}}{\bar{x}^2 - \overline{x^2}}$$

课程大纲

- 引言
- 单特征情况
- 多特征情况
- 数值优化

- 从单特征到多特征情况的训练数据

大小（英尺）	价格 ($ 1000)
x	y
2104	460
1416	232
1534	315
852	178
….	….

大小（英尺）	# 卧室数	# 楼层	# 房龄（年）	价格 ($ 1000)
x_1	x_2	x_3	x_4	y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
….	….	….	….	….

- 一般线性回归的函数是：

$$f(x_1, x_2 \cdots x_m) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_m x_m$$

  - x_i 是第 i 个特征

- 使用标量形式很繁琐。将其重新表述为矩阵形式，得到：

$$f(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{w}$$

  - $\boldsymbol{x} = [1, x_1, x_2, \cdots, x_m]$ 是特征行向量
  - $\boldsymbol{w} = [w_0, w_1, w_2, \cdots, w_m]^T$ 是参数列向量

通过将 $\mathbf{x}$ 的第一个元素设置为1，可以将 w_0 像其他参数 w_k 一样处理

代价函数

- 目标仍然是找到一个 $\boldsymbol{w}$ ，使得预测值

$$f\left(\boldsymbol{x}^{(i)}\right) = \boldsymbol{x}^{(i)}\boldsymbol{w}$$

尽可能接近真实值 $y^{(i)}$ ，其中 $\boldsymbol{x}^{(i)}$ 和 $y^{(i)}$ 分别是第 i 个特征向量和目标值

大小（英尺）x_1	# 卧室数 x_2	# 楼层 x_3	# 房龄（年）x_4	价格 (\$ 1000) y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
….	….	….	….	….

- 因此，代价函数可以表示为：

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

- 代价函数可以进一步写为：

$$L(\boldsymbol{w}) = \frac{1}{n}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2$$

其中 $\boldsymbol{X}$ 是特征矩阵，定义为：

$$\boldsymbol{X} \triangleq \begin{bmatrix} \boldsymbol{x}^{(1)} \\ \vdots \\ \boldsymbol{x}^{(n)} \end{bmatrix}$$

	大小（英尺）	# 卧室数	# 楼层	# 房龄（年）	价格 ($ 1000)
x_0	x_1	x_2	x_3	x_4	y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178
1	….	….	….	….	….

$\boldsymbol{X}$ $\boldsymbol{y}$

- 代价函数关于 $\boldsymbol{w}$ 的梯度为：

$$\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}} = \frac{2}{n}\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$$

- 由于 $L(\boldsymbol{w})$ 是一个凸函数，其最优解可以通过令其导数（梯度）为零来找到：

$$\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}} = \frac{2}{n}\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}) = 0$$

- 求解该方程可得：

$$\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

- 可以验证，当特征数量为 1 时，该结果可以简化为：

$$w_0 = \frac{\overline{xy}\bar{x} - \overline{x^2}\bar{y}}{\bar{x}^2 - \overline{x^2}}, \qquad w_1 = \frac{\bar{x}\bar{y} - \overline{xy}}{\bar{x}^2 - \overline{x^2}}$$

补充：函数关于向量或矩阵的梯度

- 函数关于向量或矩阵的梯度的含义

➢ $L(\cdot)$ 是一个标量函数

$$\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} \triangleq \begin{bmatrix} \frac{\partial L(\mathbf{w})}{\partial w_1} \\ \vdots \\ \frac{\partial L(\mathbf{w})}{\partial w_m} \end{bmatrix} \qquad \frac{\partial L(\mathbf{X})}{\partial \mathbf{X}} \triangleq \begin{bmatrix} \frac{\partial L(\mathbf{X})}{\partial x_{11}} & \cdots & \frac{\partial L(\mathbf{X})}{\partial x_{1m}} \\ \vdots & \ddots & \vdots \\ \frac{\partial L(\mathbf{X})}{\partial x_{m1}} & \cdots & \frac{\partial L(\mathbf{X})}{\partial x_{mm}} \end{bmatrix}$$

函数关于向量或矩阵的梯度，只是函数关于每个元素的梯度的紧凑记法

➢ $L(\cdot)$ 可以是任何函数，例如范数 $\|\cdot\|^2$ 、求和 $\sum_{i=1}^{m} w_i$ 、迹 $trace(\cdot)$、行列式 $\det(\cdot)$ 等

$$\frac{\partial \|\mathbf{w}\|^2}{\partial \mathbf{w}} = 2\mathbf{w} \qquad \frac{\partial \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2}{\partial (\mathbf{X}\mathbf{w} - \mathbf{y})} = 2(\mathbf{X}\mathbf{w} - \mathbf{y})$$

- 当 $\boldsymbol{L}(\cdot) = [L_1(\boldsymbol{w}), \cdots, L_p(\boldsymbol{w})]$ 是一个行向量函数时

$$\frac{\partial \boldsymbol{L}(\boldsymbol{w})}{\partial \boldsymbol{w}} \triangleq \left[\frac{\partial L_1(\boldsymbol{w})}{\partial \boldsymbol{w}} \quad \cdots \quad \frac{\partial L_p(\boldsymbol{w})}{\partial \boldsymbol{w}}\right] = \begin{bmatrix} \frac{\partial L_1(\boldsymbol{w})}{\partial w_1} & \cdots & \frac{\partial L_p(\boldsymbol{w})}{\partial w_1} \\ \vdots & \cdots & \vdots \\ \frac{\partial L_1(\boldsymbol{w})}{\partial w_m} & \cdots & \frac{\partial L_p(\boldsymbol{w})}{\partial w_m} \end{bmatrix}$$

它仍然是一种紧凑记法

- 然后，我们可以看到

链式法则

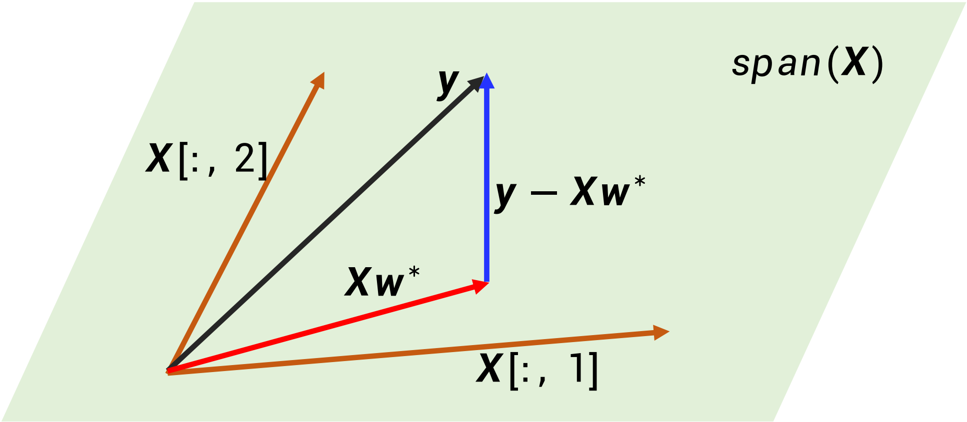
$$\frac{\partial \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2}{\partial \boldsymbol{w}} = \boxed{\frac{\partial (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^T}{\partial \boldsymbol{w}} \frac{\partial \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2}{\partial (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})}} = 2\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$$

几何解释

- 从 $\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w}^* - \boldsymbol{y}) = \boldsymbol{0}$ 这一条件，我们可以看出

$$\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^* \perp span(\boldsymbol{X})$$

- 这个结果表明， $\boldsymbol{X}\boldsymbol{w}^*$ 可以被理解为 $\boldsymbol{y}$ 在由 $\boldsymbol{X}$ 张成的空间上的投影



课程大纲

- 引言
- 单特征情况
- 多特征情况
- 数值优化

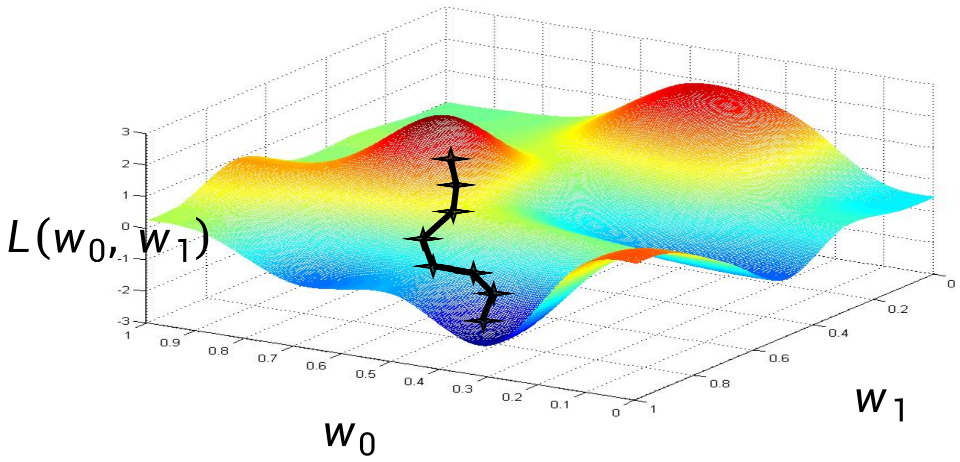
梯度下降

- 解析解*并非总是存在*，或者计算解析式的成本*过于高昂*

- 在这种情况下，我们可以求助于数值方法，例如梯度下降

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - r \cdot \frac{\partial L(\mathbf{w})}{\partial \mathbf{w}}\bigg|_{\mathbf{w}=\mathbf{w}^{(t)}}$$

– r: 学习率

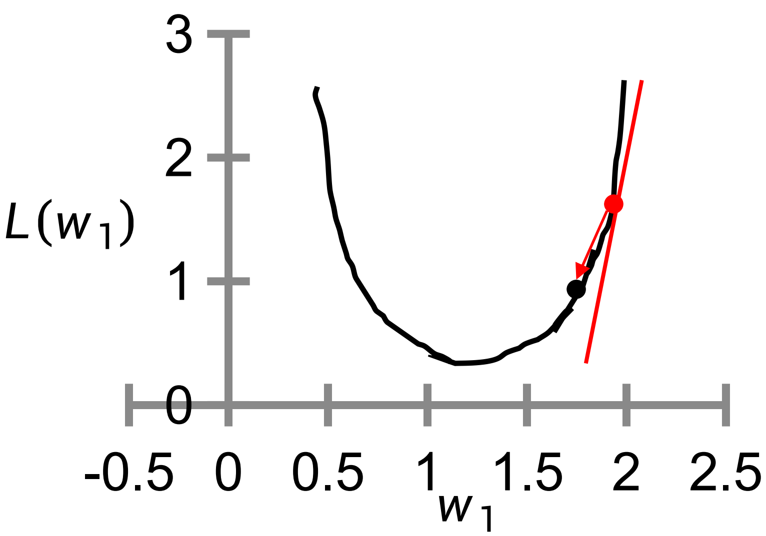


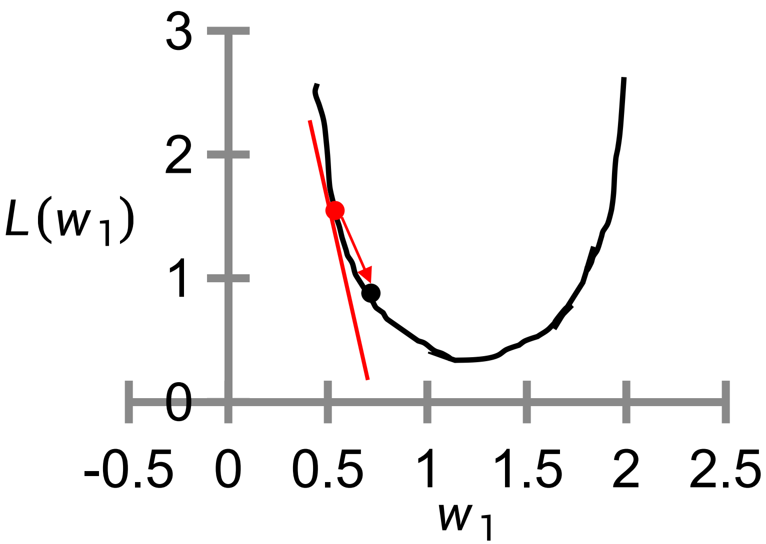
- 让我们以单特征情况为例，并令 $w_0 = 0$，此时损失函数变为：

$$L(w_1) = \frac{1}{n}\sum_{i=1}^{n}\left(w_1 x^{(i)} - y^{(i)}\right)^2$$

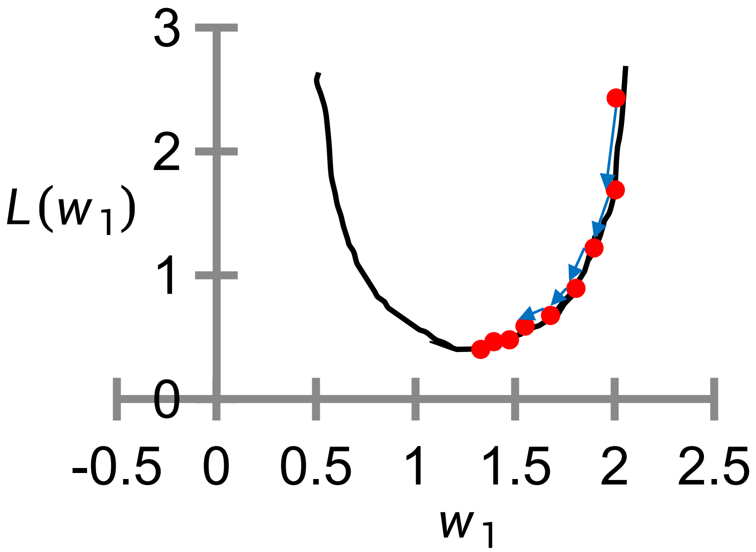
- 参数 w_1 可以按如下方式更新：

$$w_1^{(t+1)} = w_1^{(t)} - r \cdot \left.\frac{\partial L(w_1)}{\partial w_1}\right|_{w_1 = w_1^{(t)}}$$



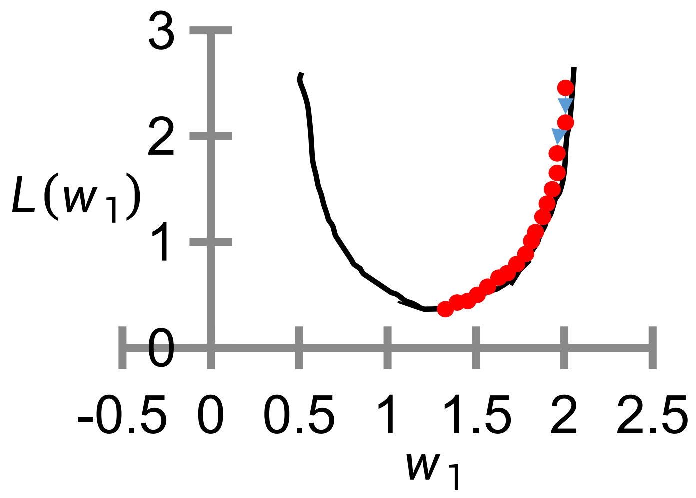


- 通过合适的学习率，模型参数会迭代更新，并最终收敛到最优解

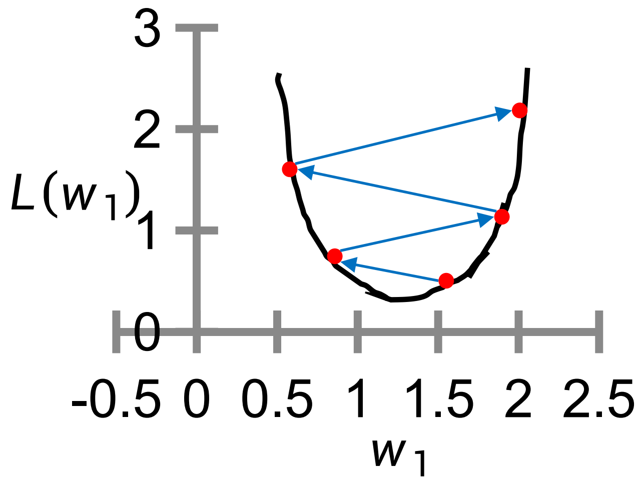


- 当逼近最优解时，梯度会变得越来越小。因此，即使学习率是固定的，只要其设置得当，随着迭代的进行，更新的步长也会趋近于0

- 如果学习率过小，收敛速度会非常慢



- 如果学习率过大，迭代可能会发散

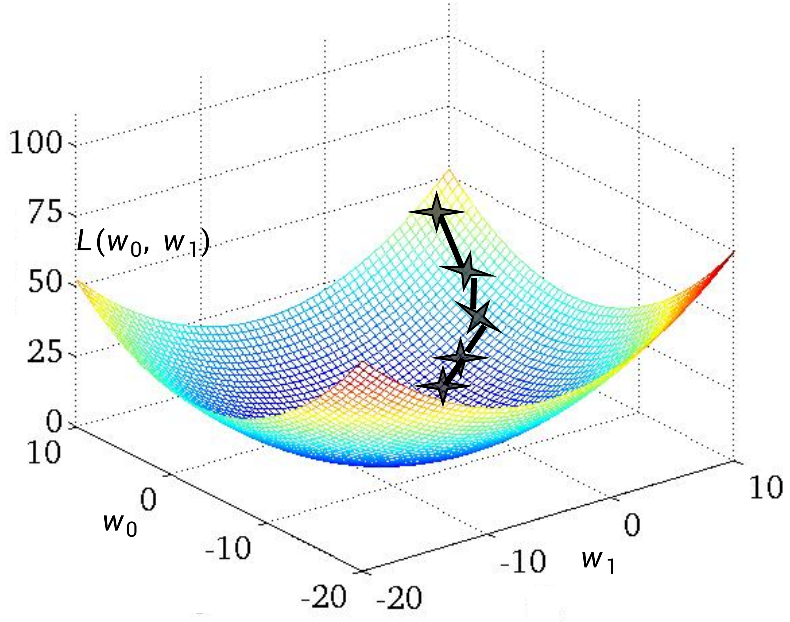


因此，设置合适的学习率非常重要

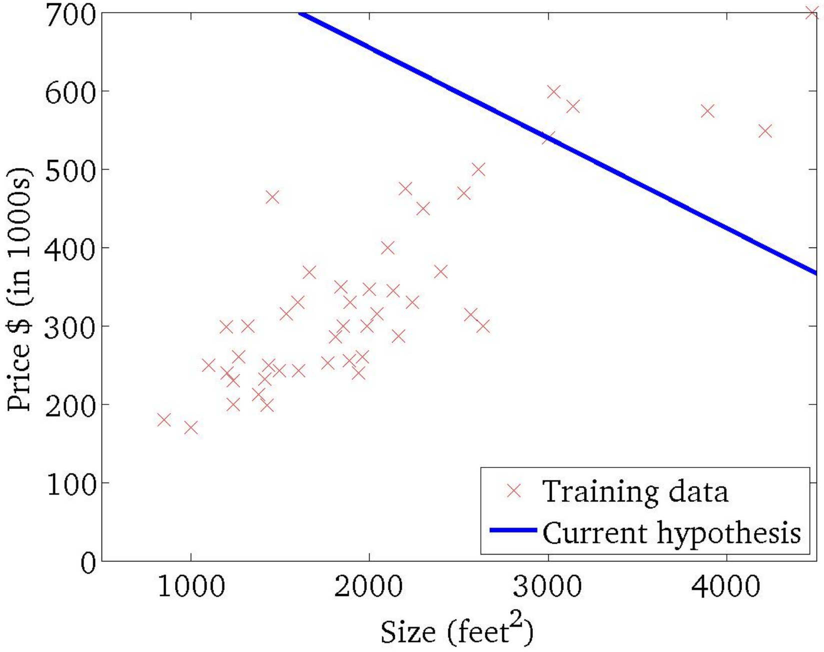
- 现在考虑同时包含 w_0 和 w_1 的情况

$$w_0^{(t+1)} = w_0^{(t)} - r \cdot \frac{\partial L(w_0, w_1)}{\partial w_0}\bigg|_{w_0 = w_0^{(t)},\ w_1 = w_1^{(t)}}$$

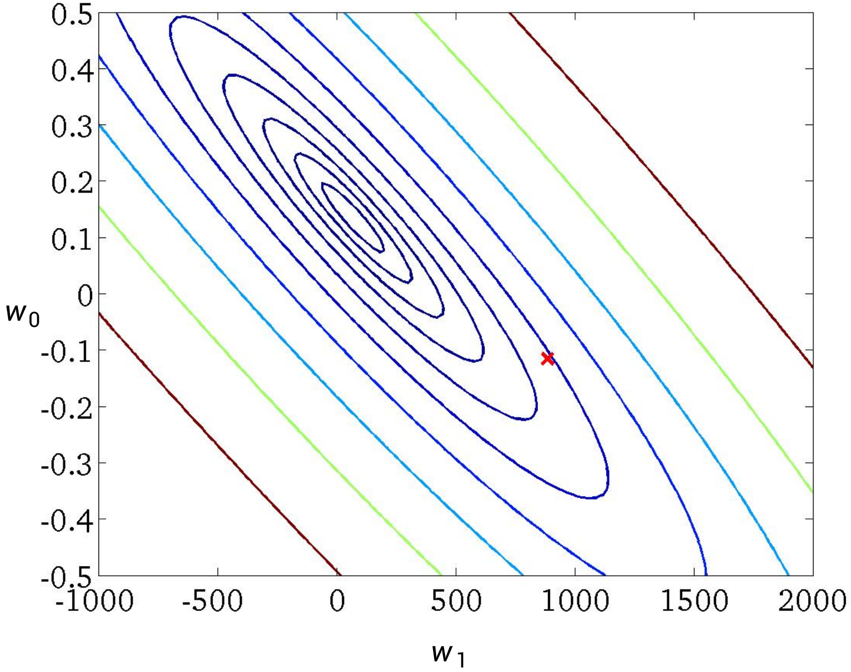
$$w_1^{(t+1)} = w_1^{(t)} - r \cdot \frac{\partial L(w_0, w_1)}{\partial w_1}\bigg|_{w_0 = w_0^{(t)},\ w_1 = w_1^{(t)}}$$



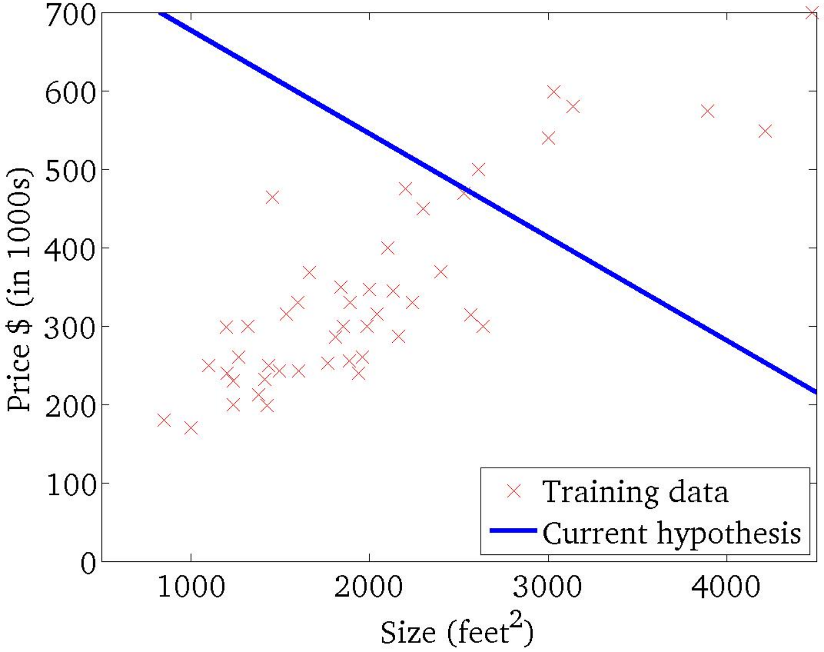
函数 $f(x) = w_0^{(t)} + w_1^{(t)} x$



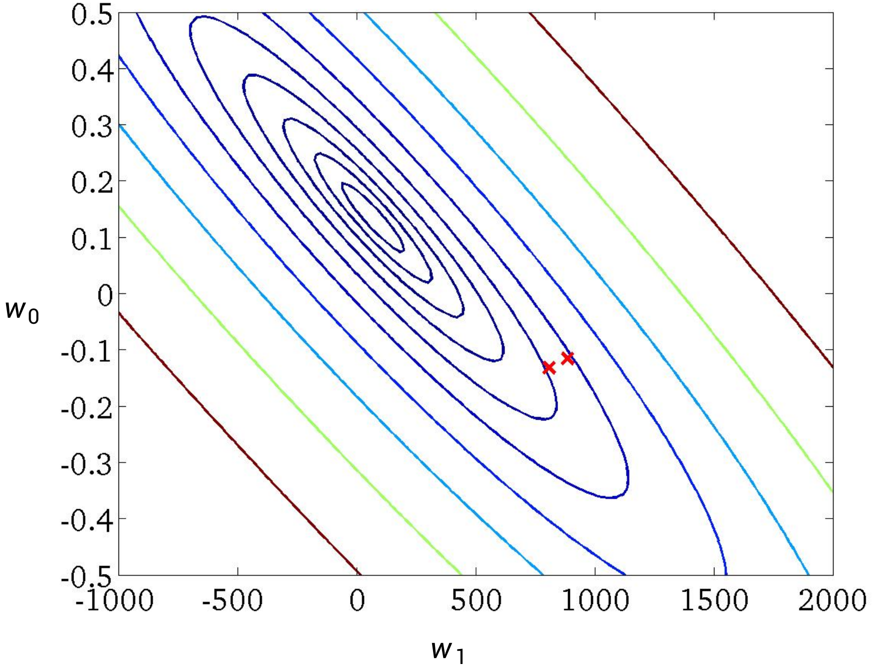
损失函数 $L(w_0, w_1)$ 的等高线图与参数轨迹 $\left((w_0^{(t)}, w_1^{(t)}\right)$



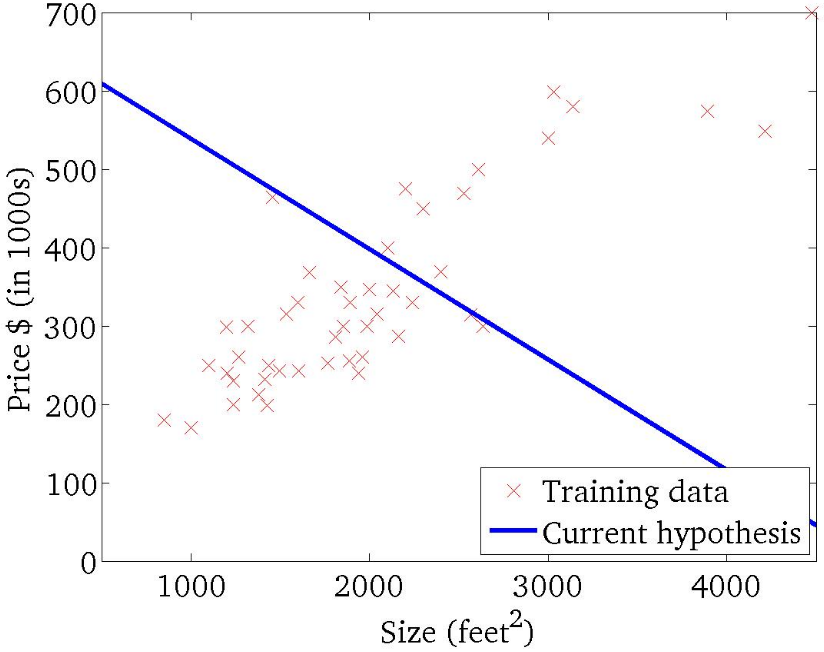
函数 $f(x) = w_0^{(t)} + w_1^{(t)} x$



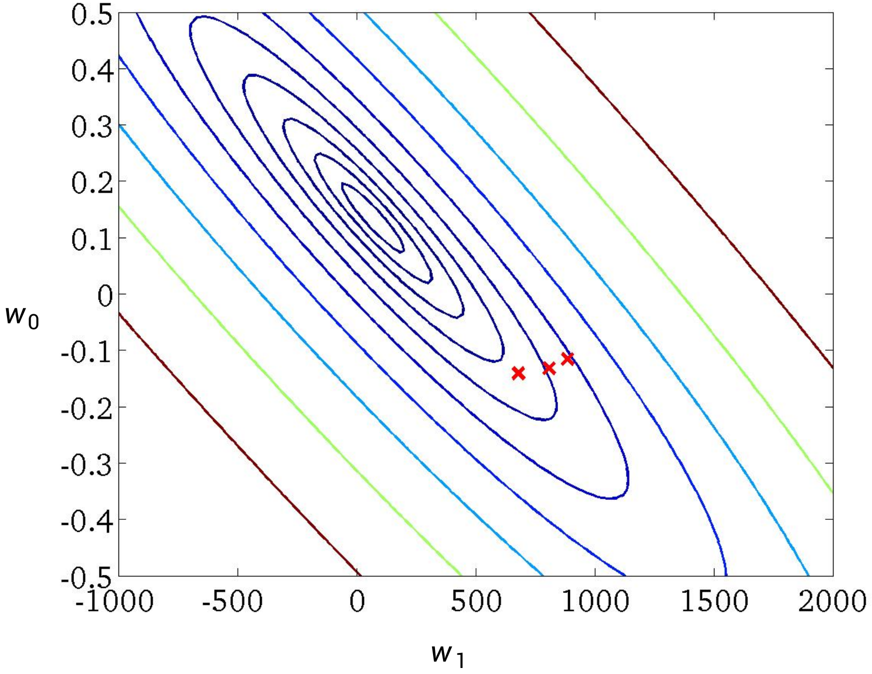
损失函数 $L(w_0, w_1)$ 的等高线图与参数轨迹 $\left((w_0^{(t)}, w_1^{(t)}\right)$



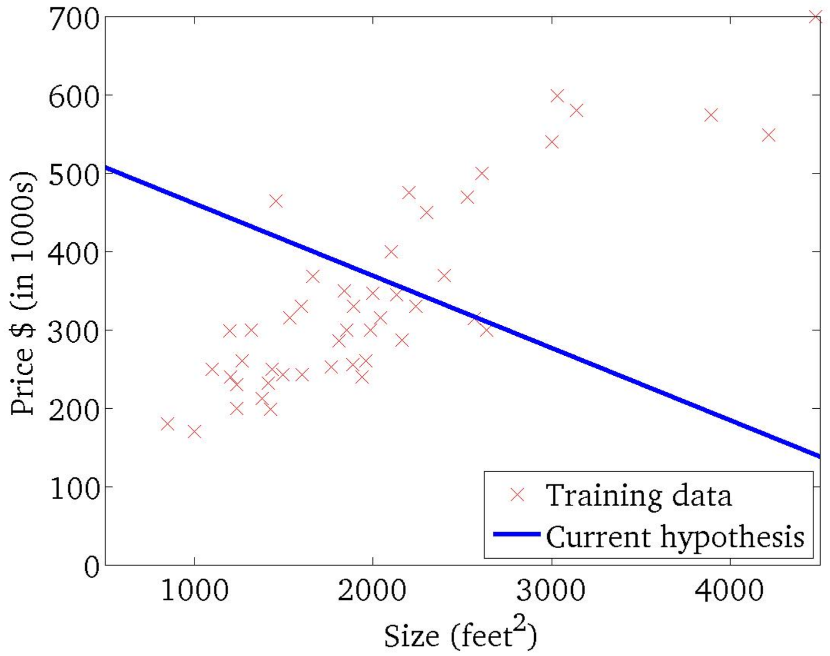
函数 $f(x) = w_0^{(t)} + w_1^{(t)} x$



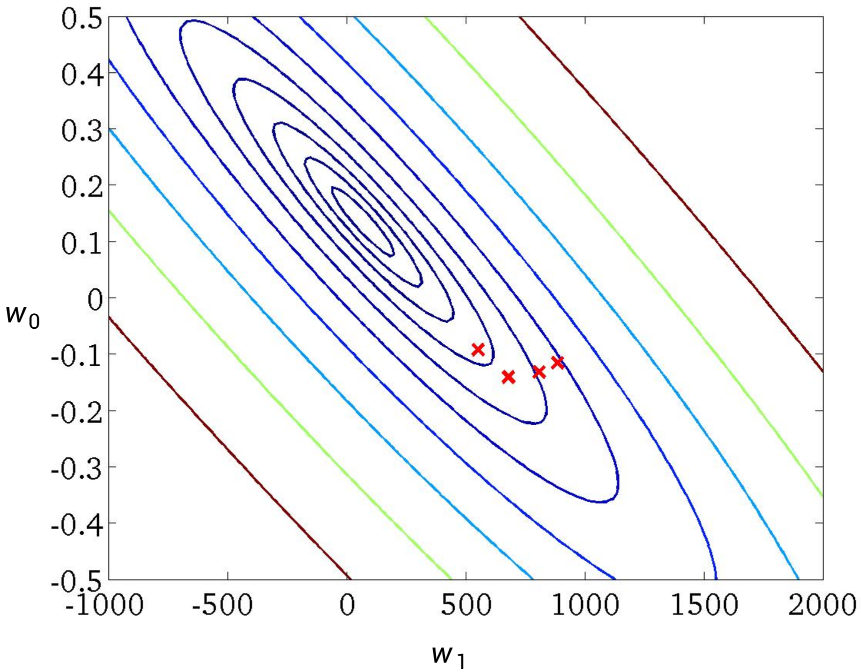
损失函数 $L(w_0, w_1)$ 的等高线图与参数轨迹 $\left((w_0^{(t)}, w_1^{(t)}\right)$



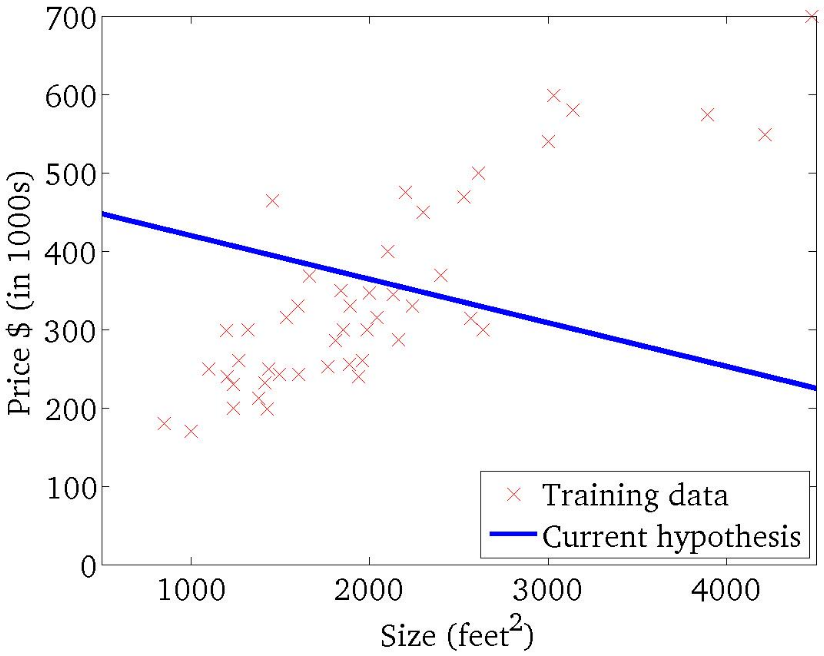
函数 $f(x) = w_0^{(t)} + w_1^{(t)} x$



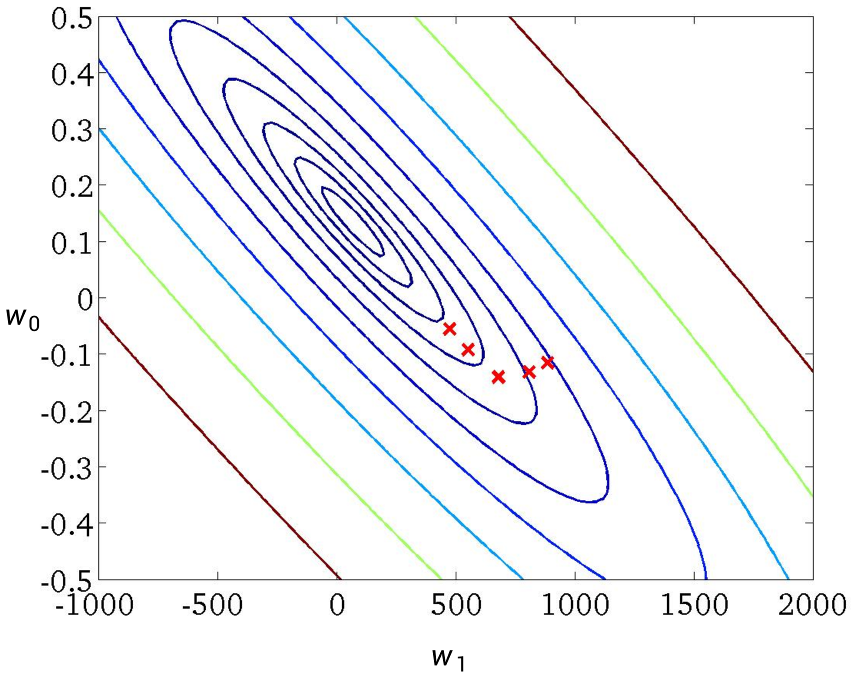
损失函数 $L(w_0, w_1)$ 的等高线图与参数轨迹 $\left((w_0^{(t)}, w_1^{(t)}\right)$



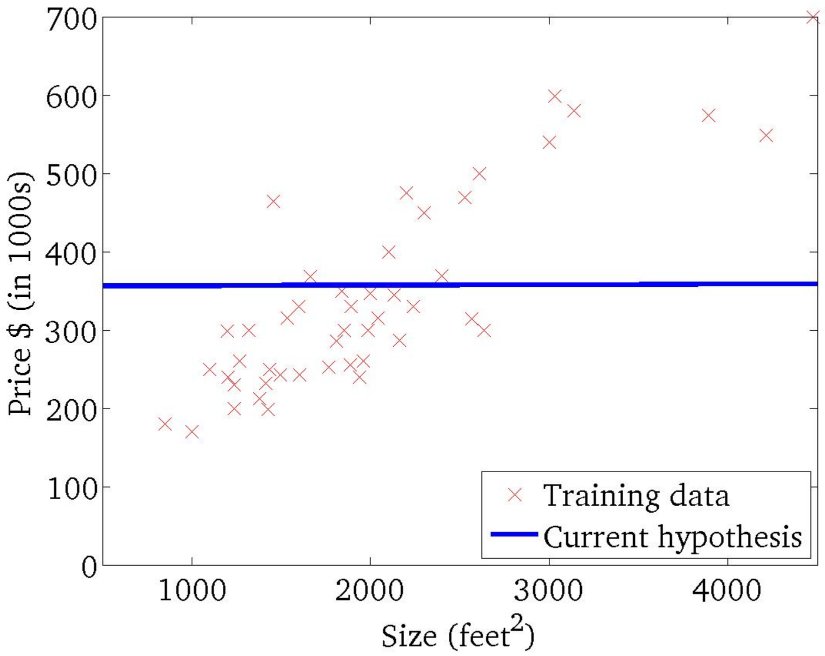
函数 $f(x) = w_0^{(t)} + w_1^{(t)} x$



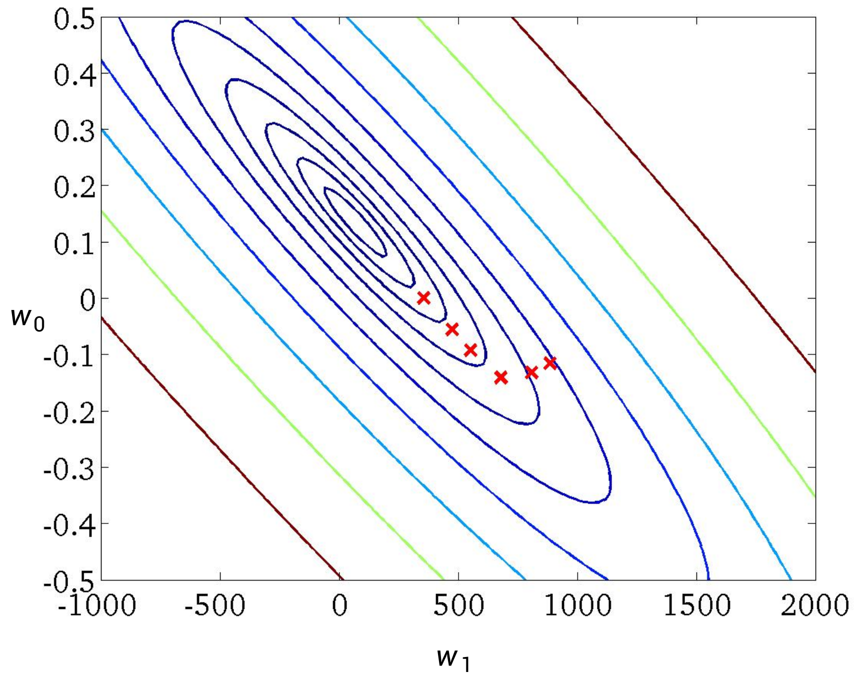
损失函数 $L(w_0, w_1)$ 的等高线图与参数轨迹 $\left((w_0^{(t)}, w_1^{(t)}\right)$



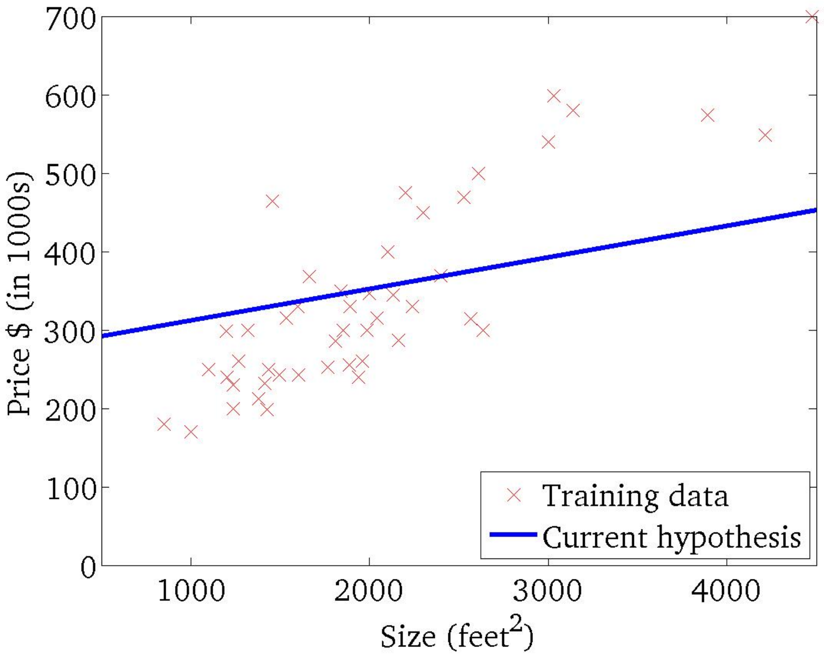
函数 $f(x) = w_0^{(t)} + w_1^{(t)} x$



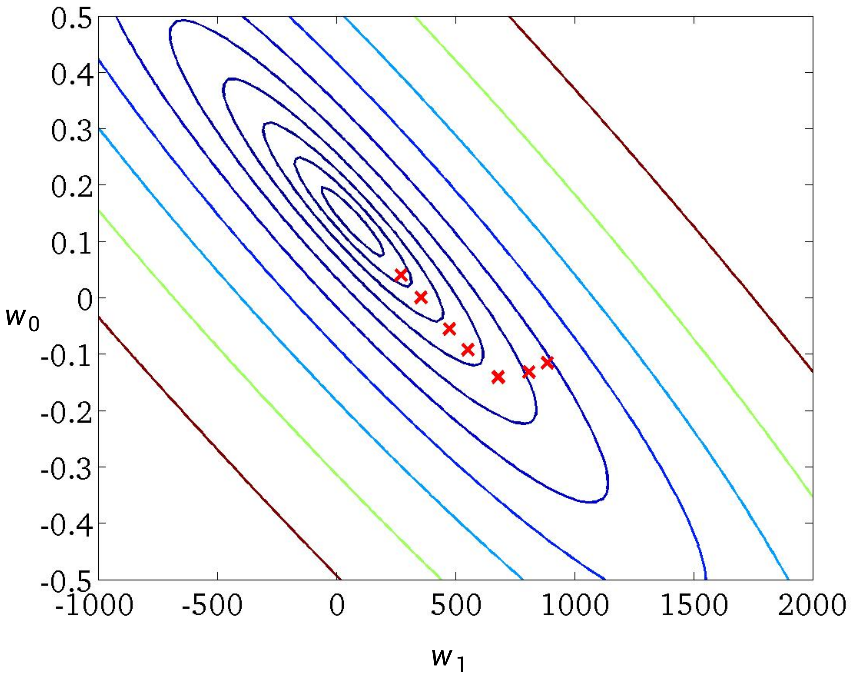
损失函数 $L(w_0, w_1)$ 的等高线图与参数轨迹 $\left((w_0^{(t)}, w_1^{(t)})\right)$



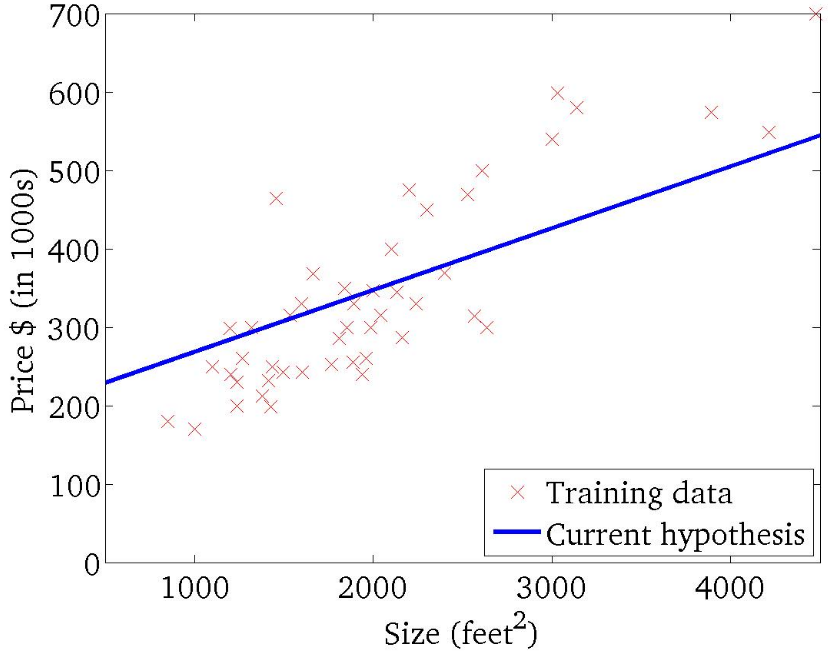
函数 $f(x) = w_0^{(t)} + w_1^{(t)} x$



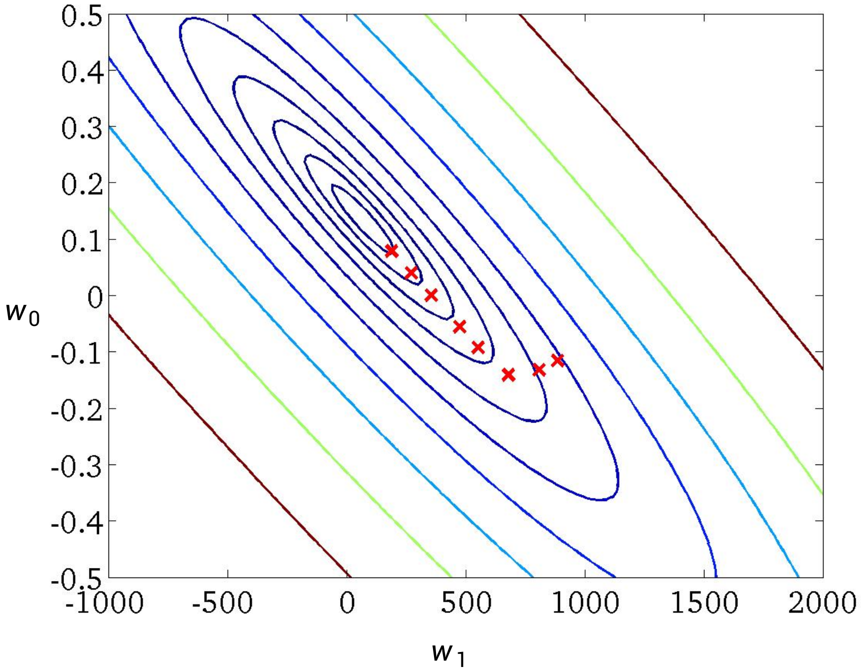
损失函数 $L(w_0, w_1)$ 的等高线图与参数轨迹 $\left((w_0^{(t)}, w_1^{(t)}\right)$



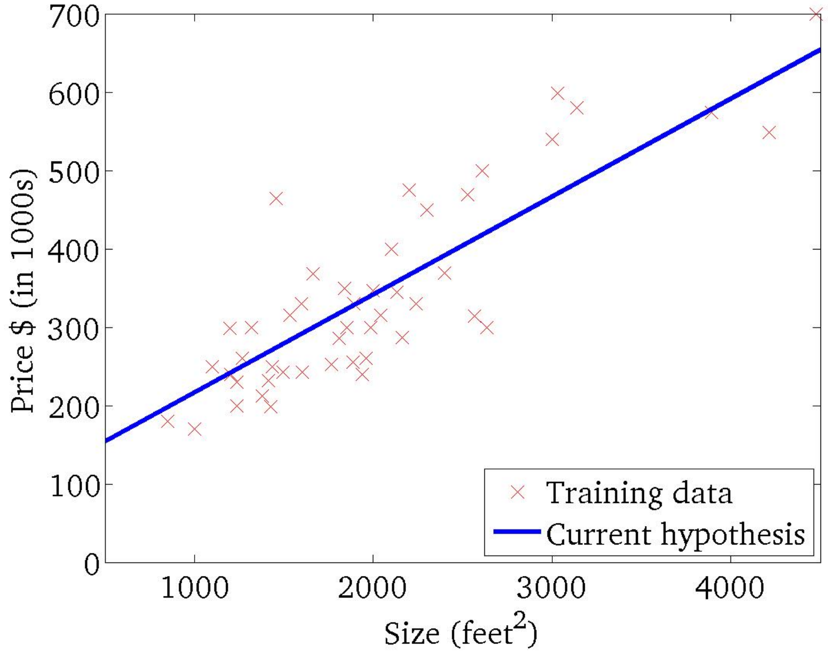
函数 $f(x) = w_0^{(t)} + w_1^{(t)}x$



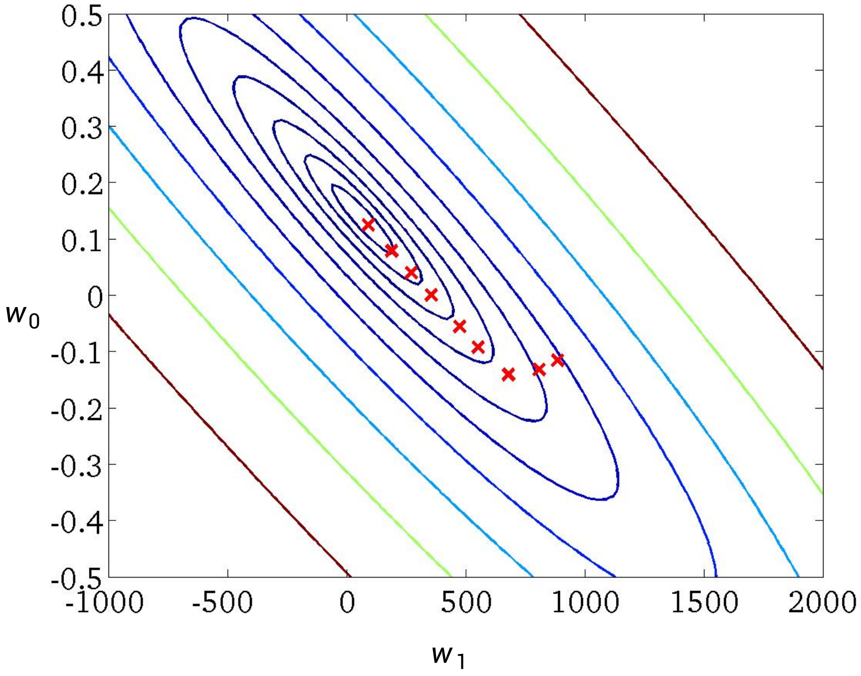
损失函数 $L(w_0, w_1)$ 的等高线图与参数轨迹 $\left((w_0^{(t)}, w_1^{(t)}\right)$



函数 $f(x) = w_0^{(t)} + w_1^{(t)} x$



损失函数 $L(w_0, w_1)$ 的等高线图与参数轨迹 $\left((w_0^{(t)}, w_1^{(t)}\right)$



随机梯度下降

- GD 算法需要在每一次迭代中计算关于模型参数 $\boldsymbol{w}$ 的损失梯度
- 通常，梯度具有以下形式：

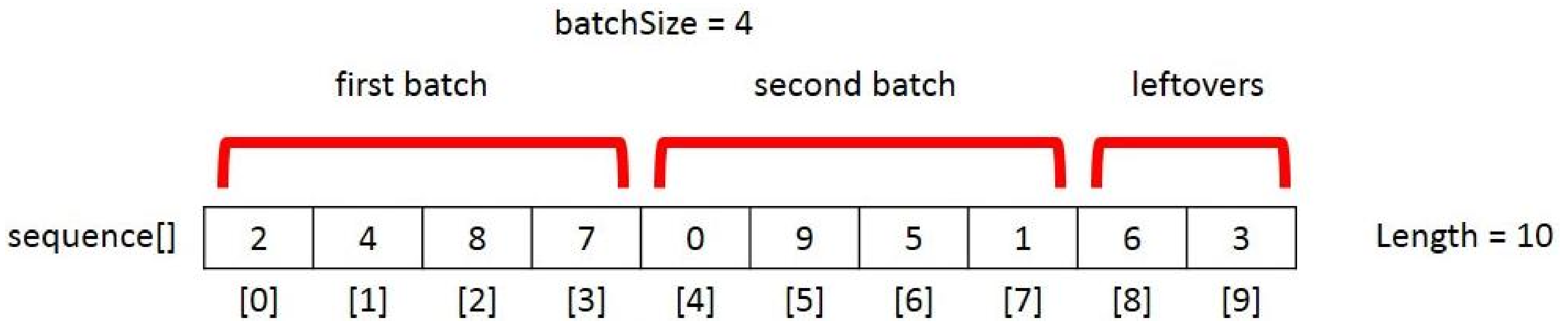
$$\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}} = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial \ell\left(\boldsymbol{w}, \boldsymbol{x}^{(i)}, y^{(i)}\right)}{\partial \boldsymbol{w}}$$

- 每一次迭代都要求在训练数据集上的所有数据样本上计算梯度

对于大型数据集，其复杂度会极高

- 为了降低复杂度，我们可以使用数据集的一小部分（即小批量）来估计梯度 $\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}}$

- 如何获取小批量数据？
  - 乱序
  - 分段



- 更新:

对真实梯度的带噪声估计

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + r \cdot \boxed{\frac{1}{|\mathcal{B}_t|}\sum_{i\in\mathcal{B}_t}\frac{\partial \ell\left(\mathbf{w}, \mathbf{x}^{(i)}, y^{(i)}\right)}{\partial \mathbf{w}}}$$

其中 $\mathcal{B}_t$ 是第 t 次迭代时数据集的一个小批量

- 问题： 随机梯度与下面的真实梯度有什么关系？

$$\frac{1}{|\mathcal{B}_t|}\sum_{i\in\mathcal{B}_t}\frac{\partial\ell\left(\mathbf{w},\,\mathbf{x}^{(i)},\,y^{(i)}\right)}{\partial\mathbf{w}}$$

与下面的真实梯度又有什么关系？

$$\frac{\partial L(\mathbf{w})}{\partial\mathbf{w}}=\frac{1}{n}\sum_{i=1}^{n}\frac{\partial\ell\left(\mathbf{w},\,\mathbf{x}^{(i)},\,y^{(i)}\right)}{\partial\mathbf{w}}$$

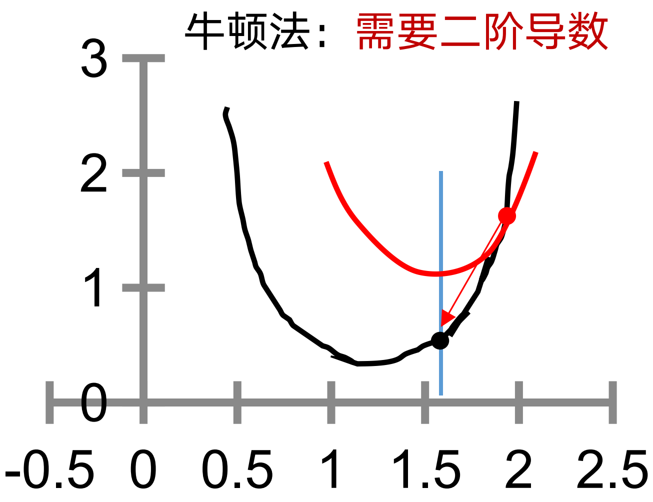
- $\frac{1}{|\mathcal{B}_t|}\sum_{i\in\mathcal{B}_t}\frac{\partial\ell\left(\mathbf{w},\mathbf{x}^{(i)},y^{(i)}\right)}{\partial\mathbf{w}}$ 是对真实梯度 $\frac{\partial L(\mathbf{w})}{\partial\mathbf{w}}$ 的无偏估计，即：

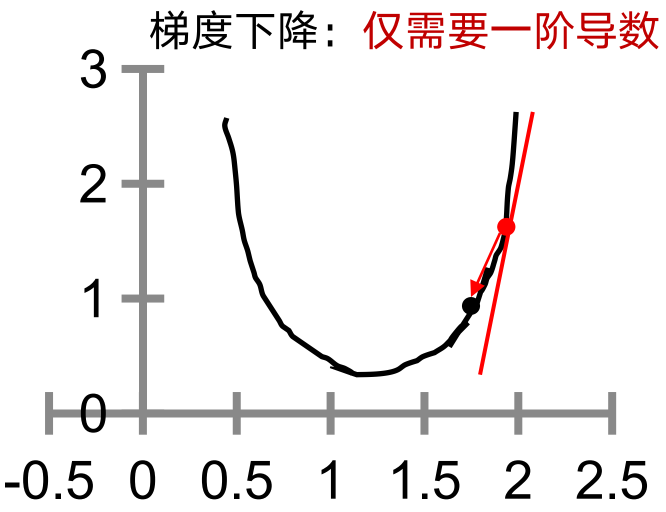
$$\mathbb{E}_{\mathcal{B}_t}\left[\frac{1}{|\mathcal{B}_t|}\sum_{i\in\mathcal{B}_t}\frac{\partial\ell\left(\mathbf{w},\,\mathbf{x}^{(i)},\,y^{(i)}\right)}{\partial\mathbf{w}}\right]=\frac{\partial L(\mathbf{w})}{\partial\mathbf{w}}$$

其他优化方法

- 还有许多其他优化方法

1) 牛顿法





优点

- 无需手动选择学习率
- 收敛速度更快

缺点

- 计算成本更高

2) 拟牛顿法

3) 共轭梯度法

4) 坐标下降法

⋮

这些方法通常比梯度下降法收敛得更快，但计算成本更高