Python 3 玩儿转机器学习

liuyubobobo

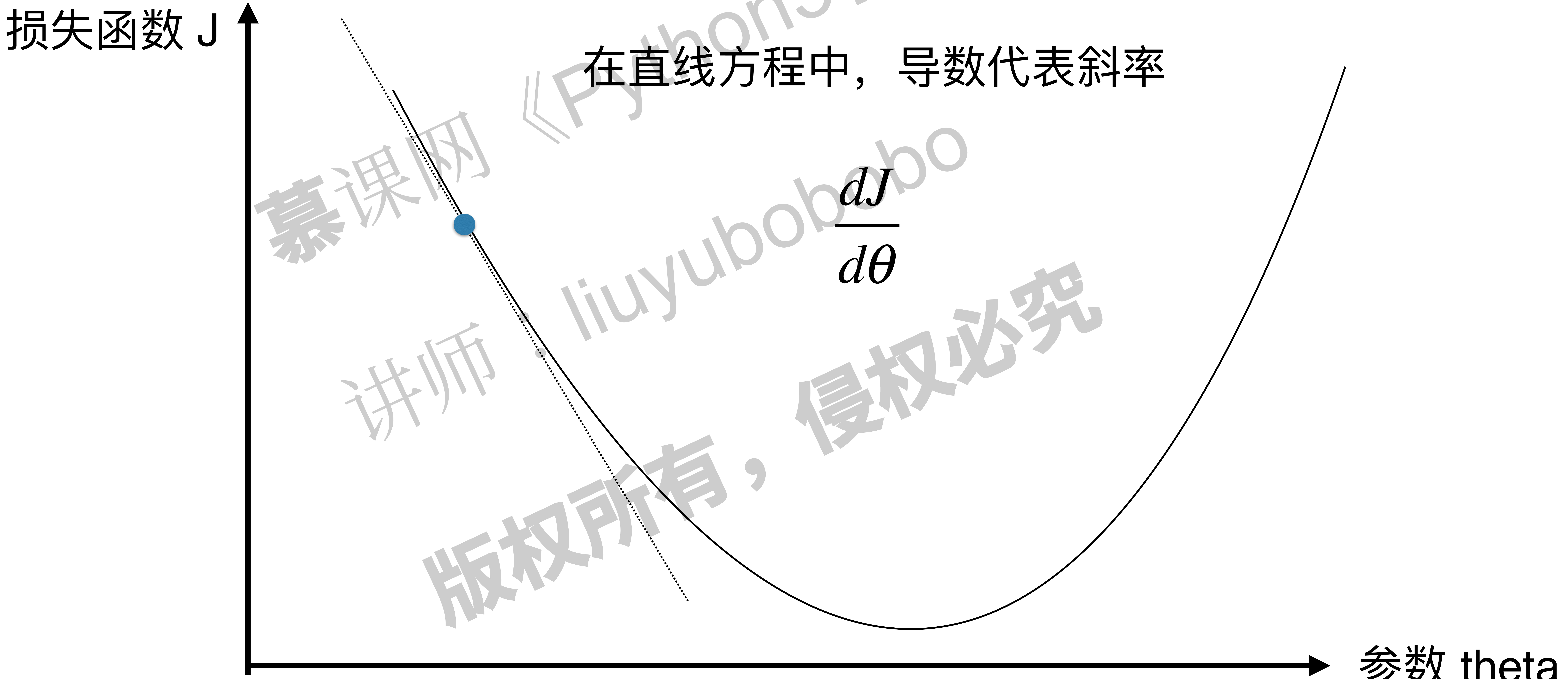
梯度下降法

Gradient Descent

梯度下降法

- 不是一个机器学习算法
- 是一种基于搜索的最优化方法
- 作用：最小化一个损失函数
- 梯度上升法：最大化一个效用函数

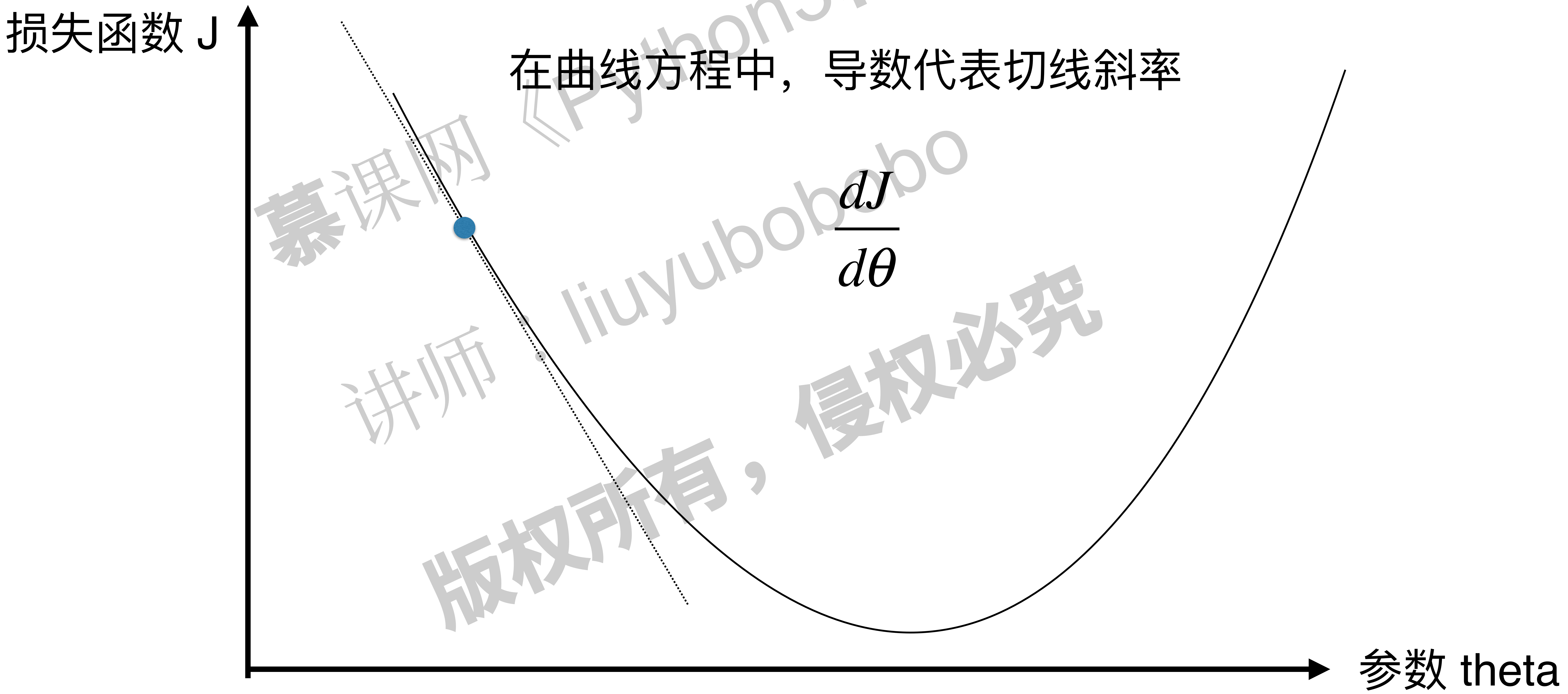
梯度下降法



在直线方程中，导数代表斜率

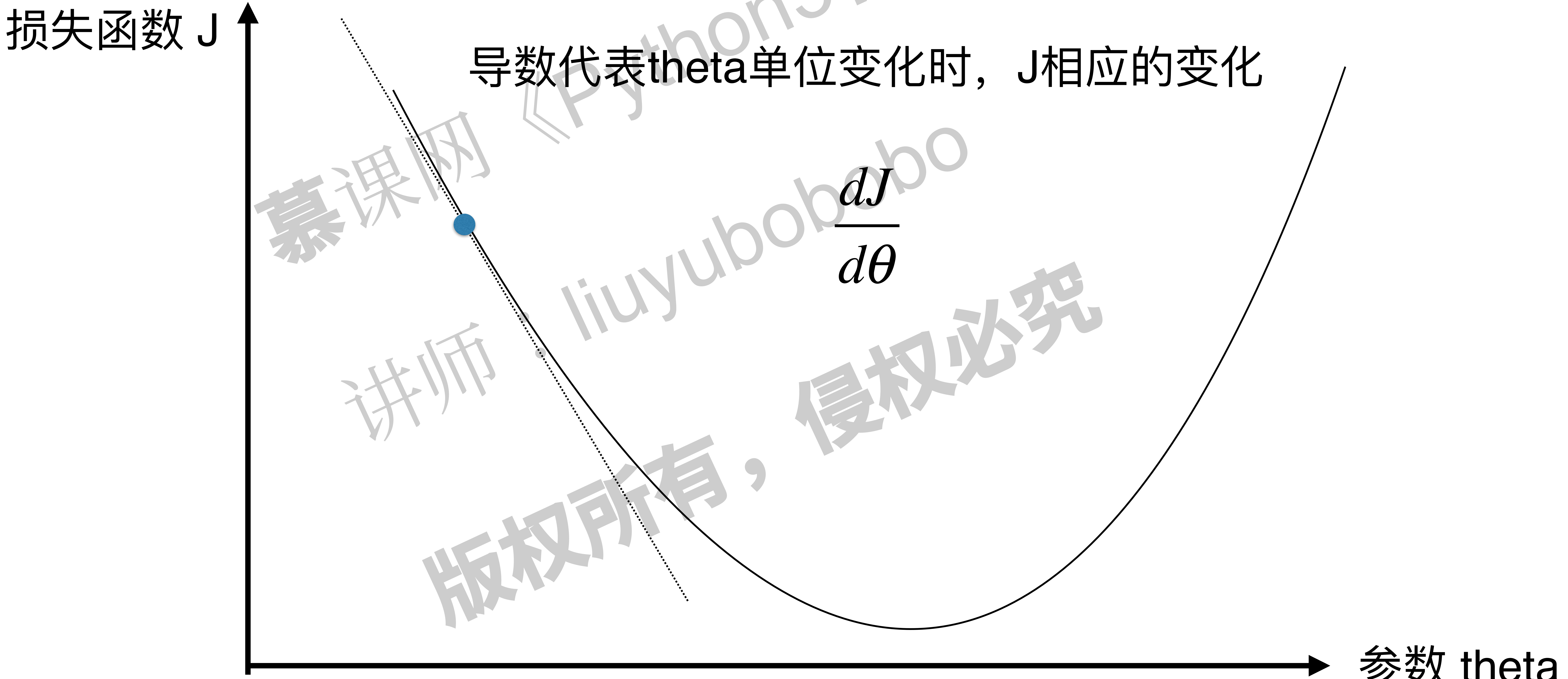
$$\frac{dJ}{d\theta}$$

梯度下降法



$$\frac{dJ}{d\theta}$$

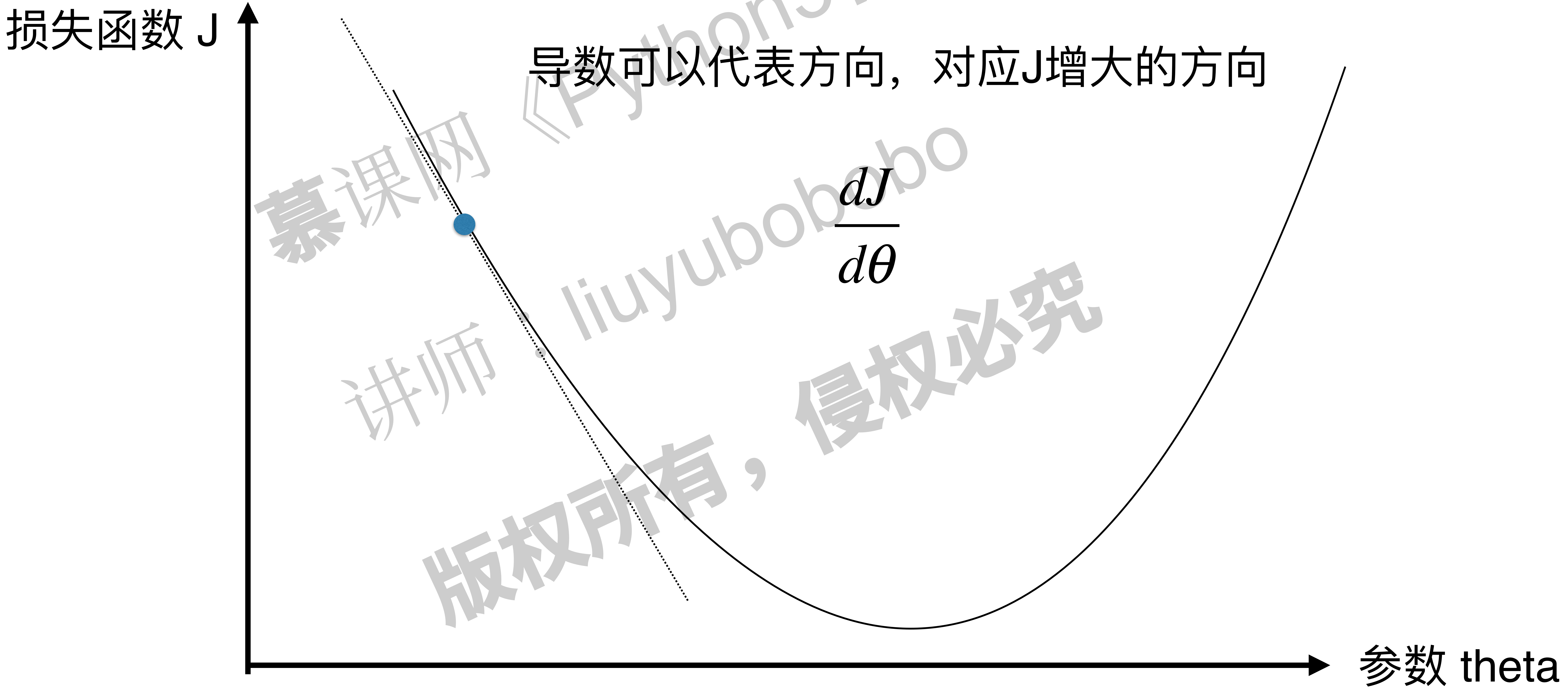
梯度下降法



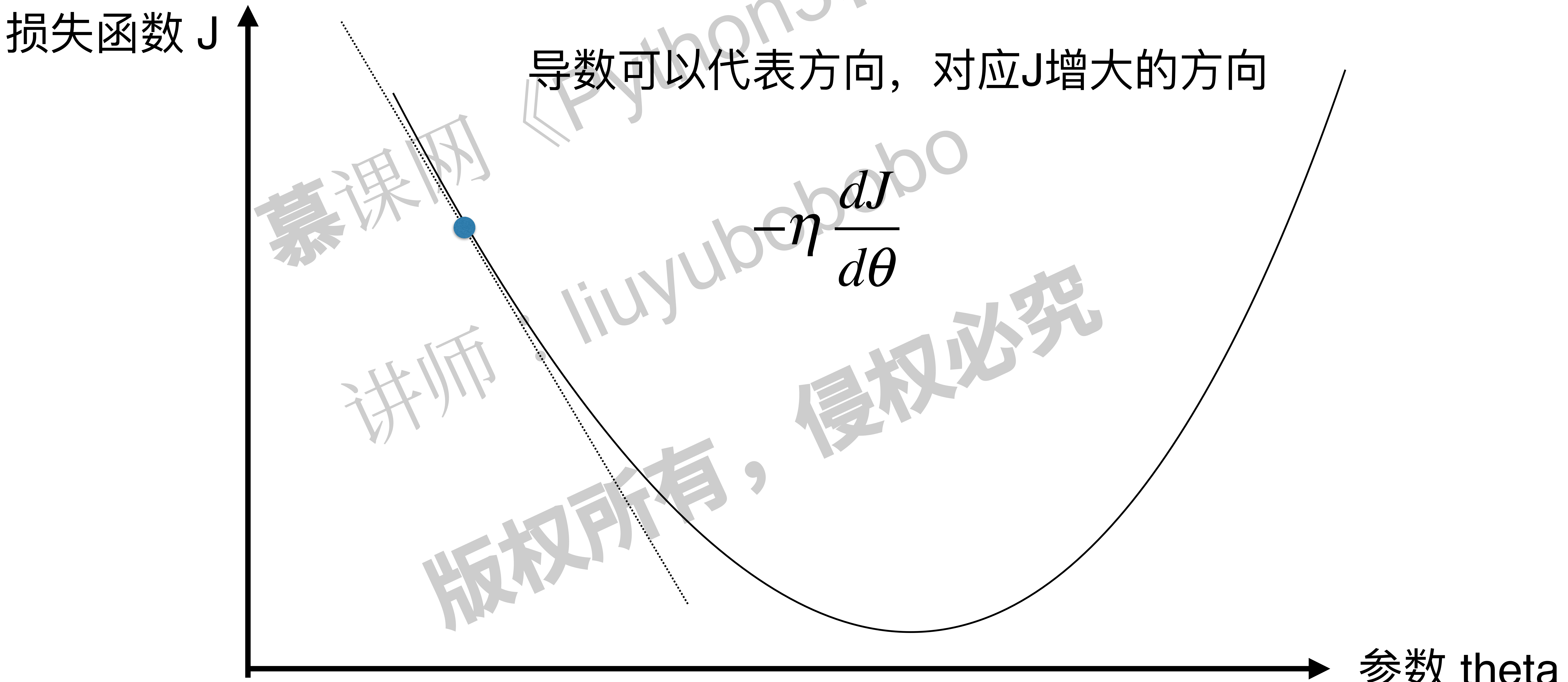
导数代表theta单位变化时，J相应的变化

$$\frac{dJ}{d\theta}$$

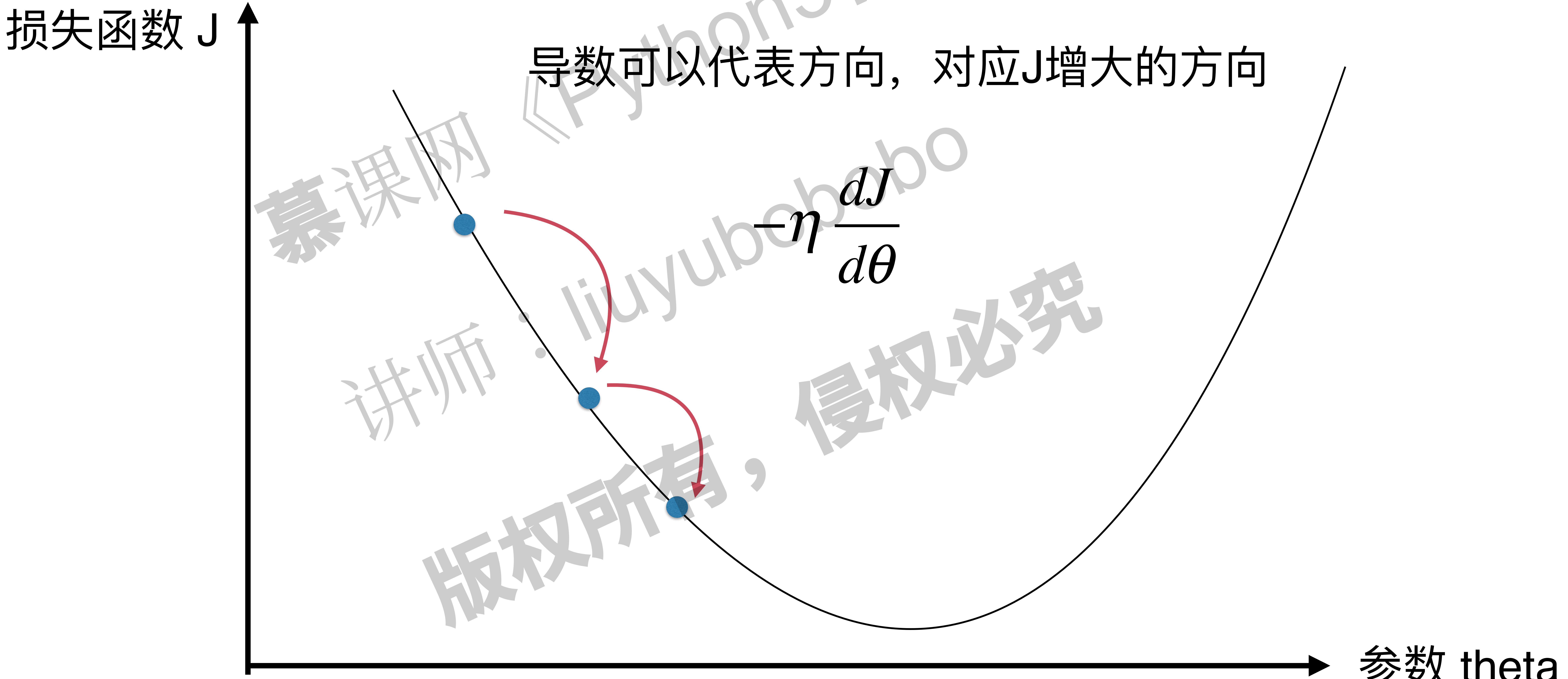
梯度下降法



梯度下降法



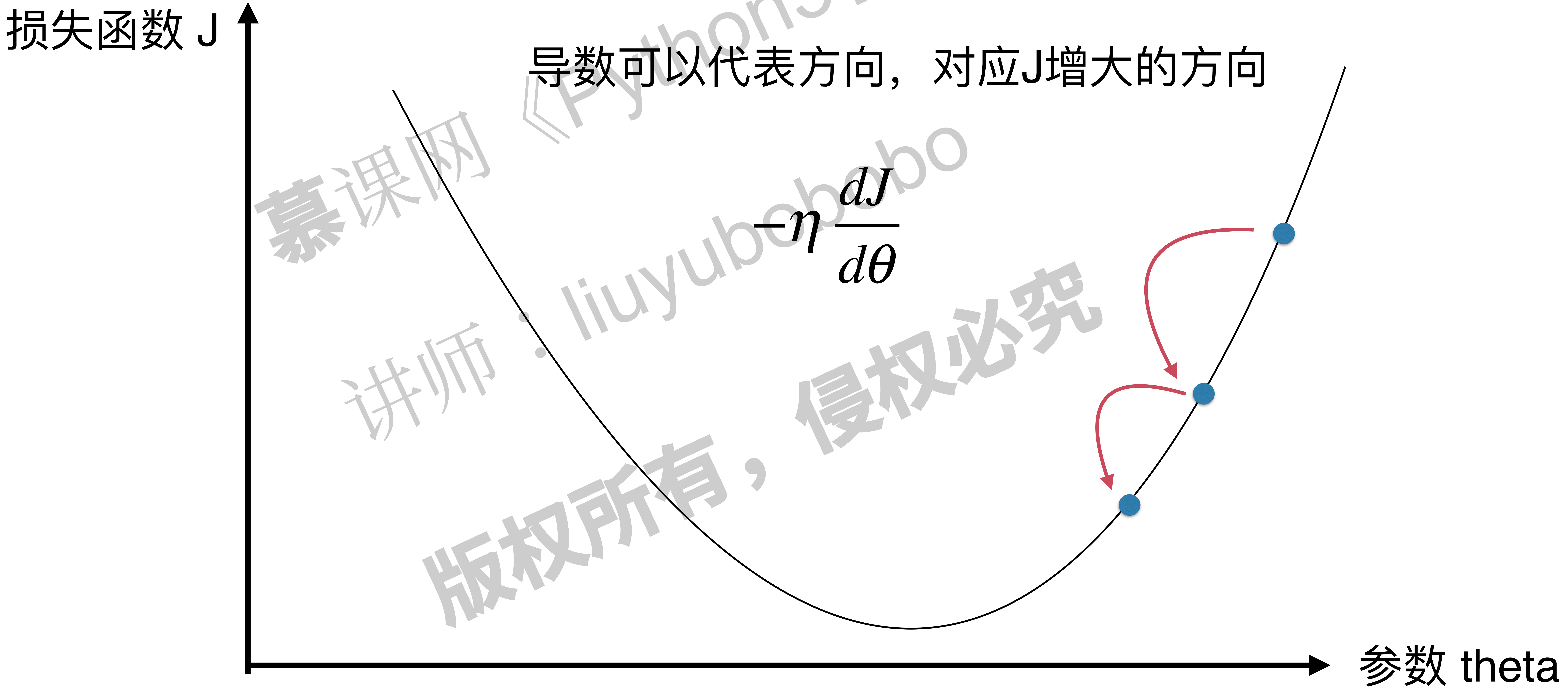
梯度下降法



导数可以代表方向，对应J增大的方向

$$-\eta\frac{dJ}{d\theta}$$

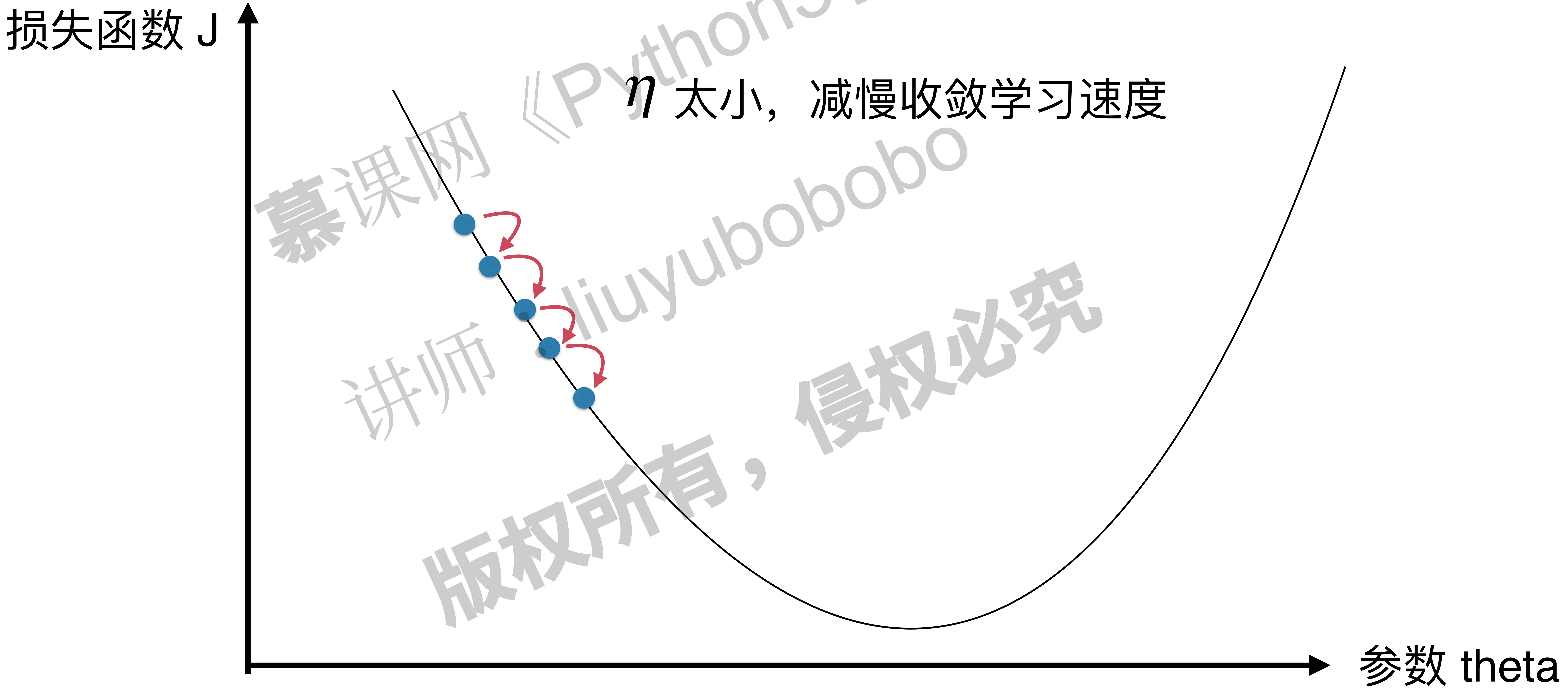
梯度下降法



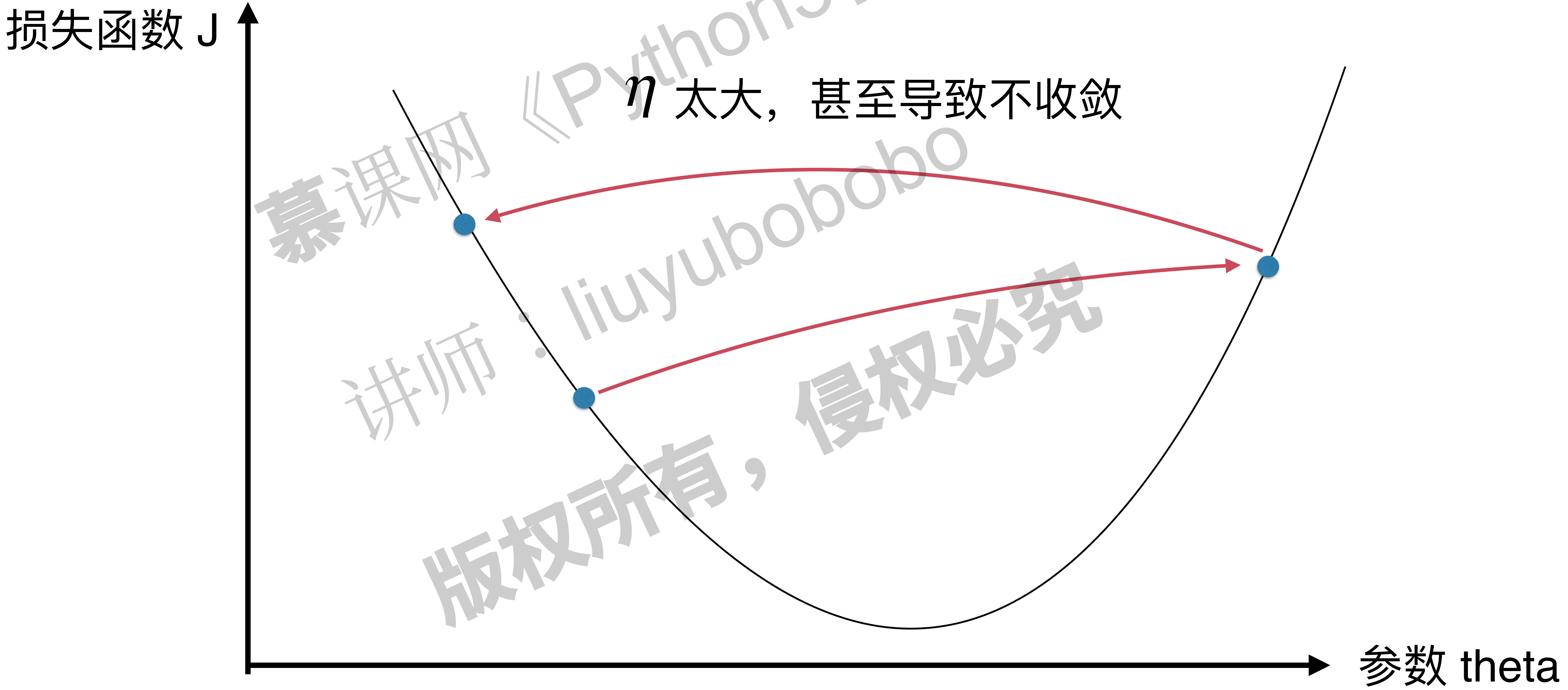
梯度下降法

- η 称为学习率(learning rate)
- η 的取值影响获得最优解的速度
- η 取值不合适，甚至得不到最优解
- η 是梯度下降法的一个超参数

梯度下降法



梯度下降法



梯度下降法

- 并不是所有函数都有唯一的极值点

梯度下降法

梯度下降法

- 并不是所有函数都有唯一的极值点

解决方案：

- 多次运行，随机化初始点
- 梯度下降法的初始点也是一个超参数

线性回归中使用梯度下降法

目标：使 $\sum_{i=1}^{m}(y^{(i)}-\hat{y}^{(i)})^2$ 尽可能小

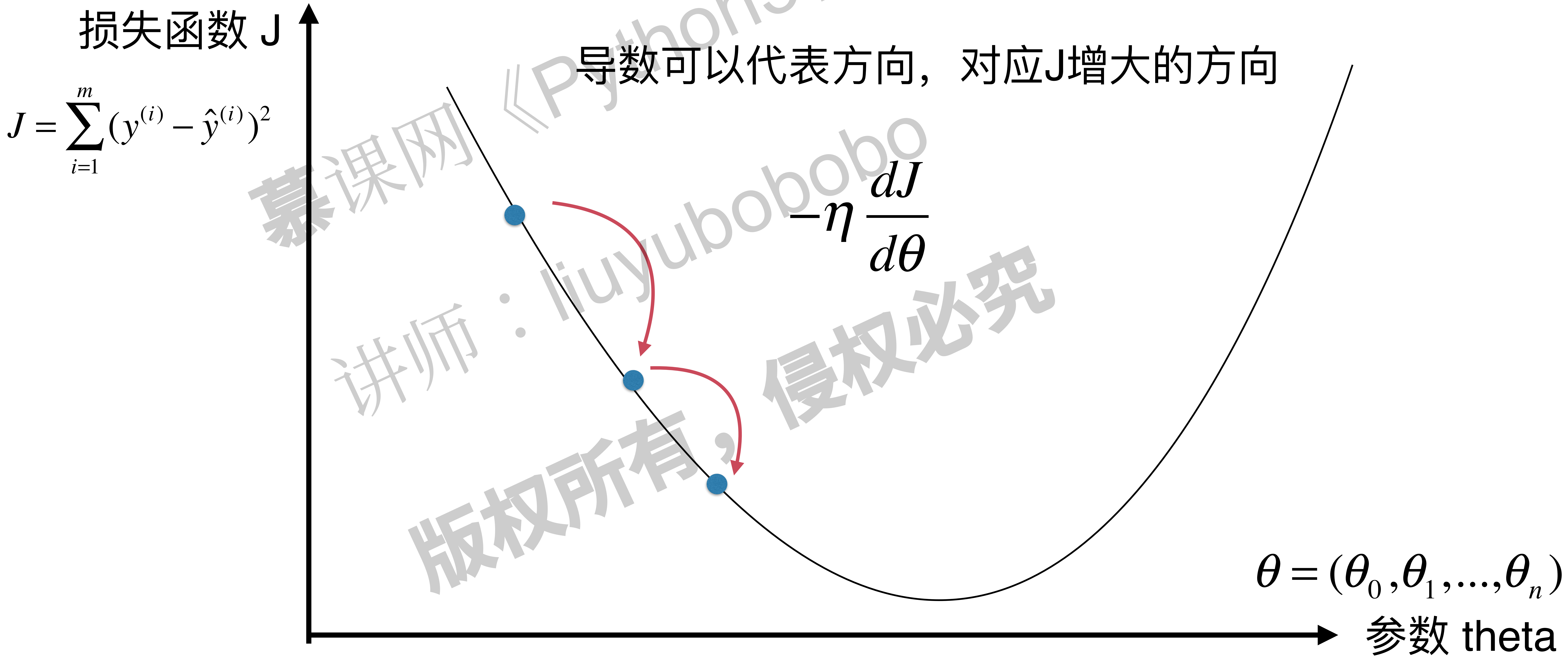
线性回归法的损失函数具有唯一的最优解

模拟梯度下降法

实践：模拟梯度下降法

多元线性回归中的梯度下降法

梯度下降法

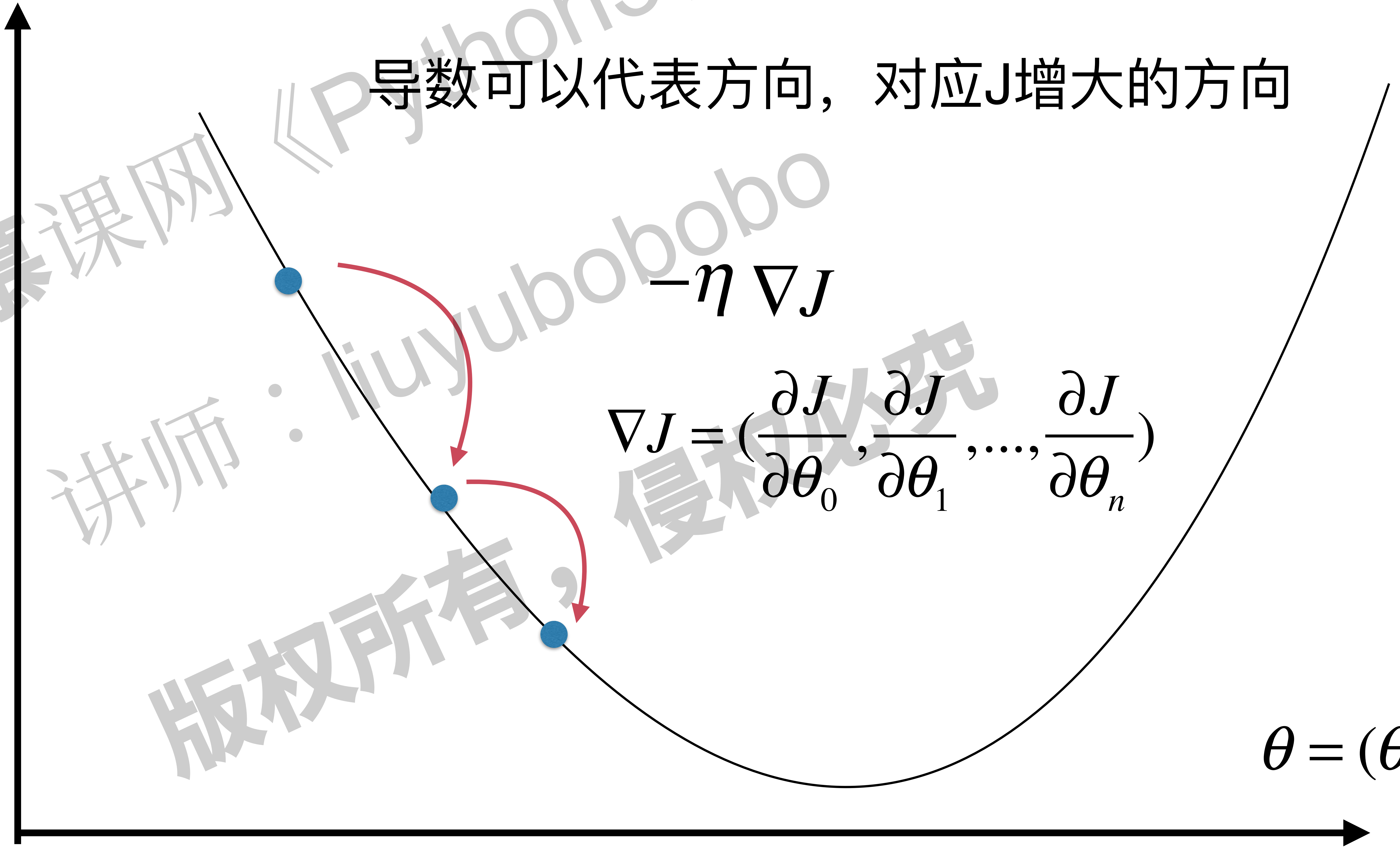


损失函数 J
$J = \sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2$
导数可以代表方向，对应J增大的方向
$-\eta \frac{dJ}{d\theta}$
$\theta = (\theta_0, \theta_1, ..., \theta_n)$
参数 theta

梯度下降法

损失函数 J

$$J = \sum_{i=1}^{m} (y^{(i)} - \hat{y}^{(i)})^2$$



导数可以代表方向，对应J增大的方向

$$-\eta \, \nabla J$$

$$\nabla J = (\frac{\partial J}{\partial \theta_0}, \frac{\partial J}{\partial \theta_1}, \ldots, \frac{\partial J}{\partial \theta_n})$$

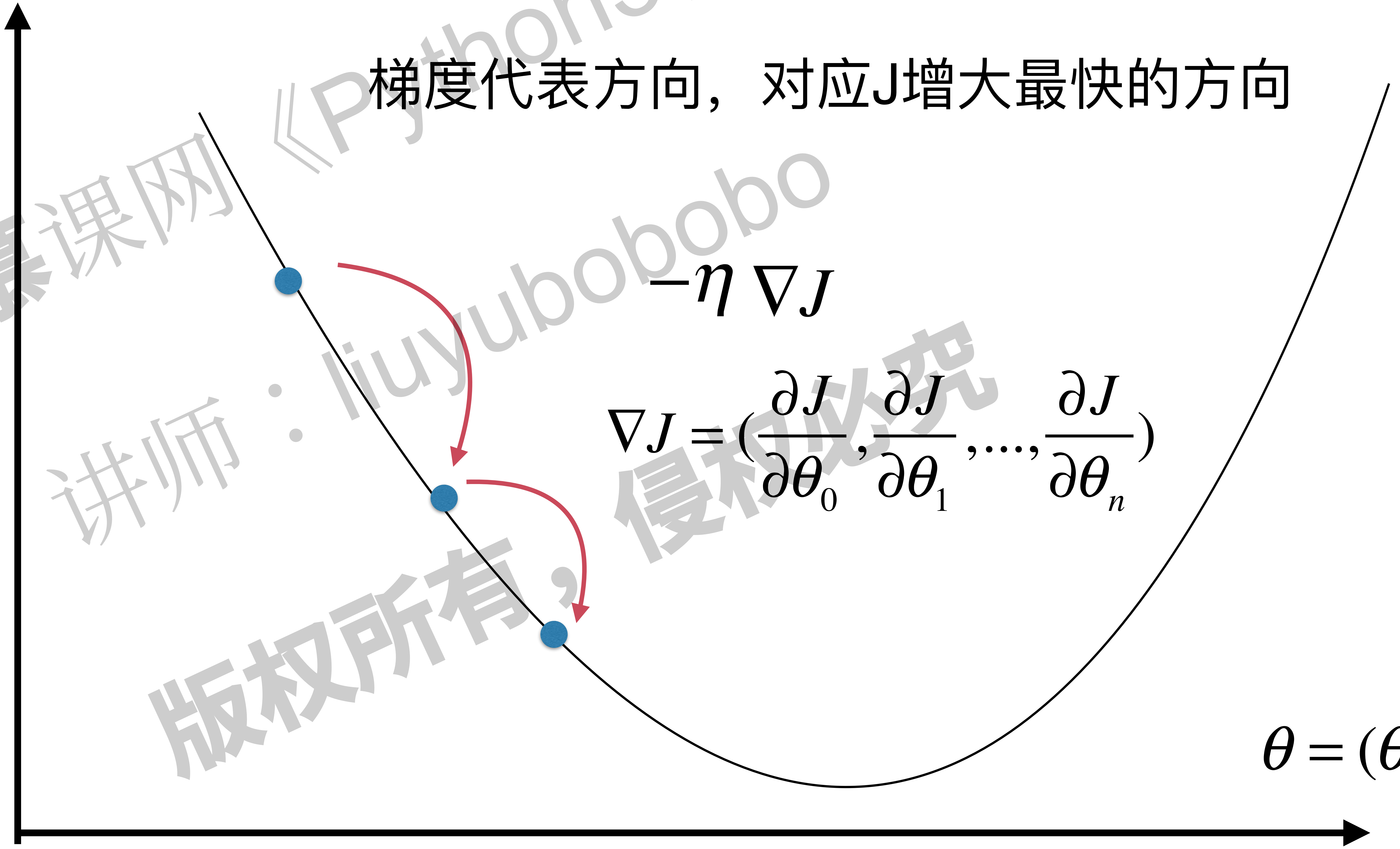
$$\theta = (\theta_0, \theta_1, \ldots, \theta_n)$$

参数 theta

梯度下降法

损失函数 J

$$J=\sum_{i=1}^{m}(y^{(i)}-\hat{y}^{(i)})^2$$



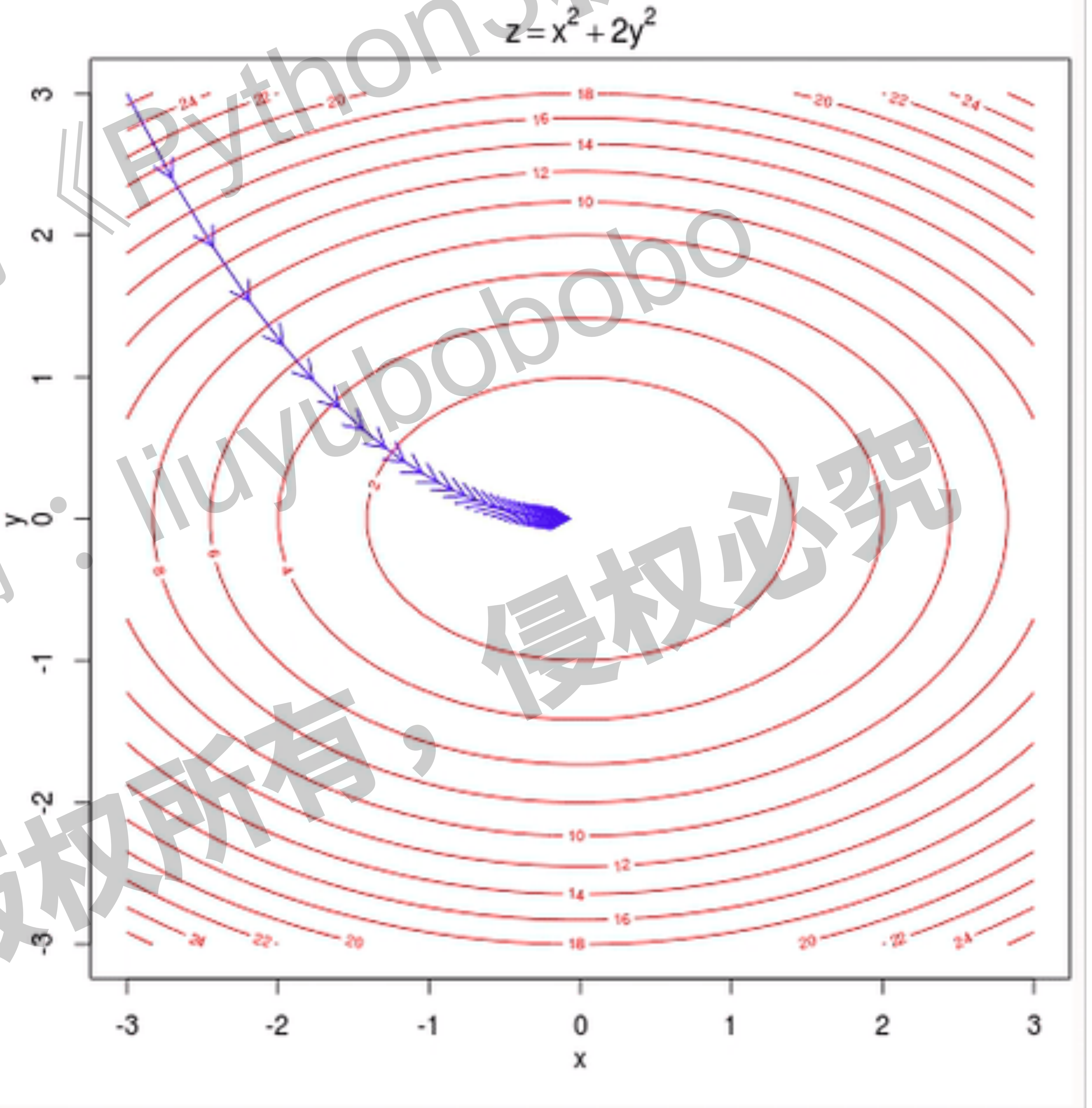
$$-\eta\,\nabla J$$

$$\nabla J=\left(\frac{\partial J}{\partial\theta_0},\frac{\partial J}{\partial\theta_1},\ldots,\frac{\partial J}{\partial\theta_n}\right)$$

$$\theta=(\theta_0,\theta_1,\ldots,\theta_n)$$

参数 theta

梯度下降法



线性回归中使用梯度下降法

目标：使 $\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2$ 尽可能小

$$\hat{y}^{(i)} = \theta_0 + \theta_1 X_1^{(i)} + \theta_2 X_2^{(i)} + \ldots + \theta_n X_n^{(i)}$$

目标：使 $\sum_{i=1}^{m}(y^{(i)} - \theta_0 - \theta_1 X_1^{(i)} - \theta_2 X_2^{(i)} - \ldots - \theta_n X_n^{(i)})^2$ 尽可能小

线性回归中使用梯度下降法

目标：使 $\sum_{i=1}^{m}(y^{(i)}-\theta_0-\theta_1X_1^{(i)}-\theta_2X_2^{(i)}-\ldots-\theta_nX_n^{(i)})^2$ 尽可能小

$$\nabla J(\theta) = \begin{pmatrix} \partial J / \partial\theta_0 \\ \partial J / \partial\theta_1 \\ \partial J / \partial\theta_2 \\ \ldots \\ \partial J / \partial\theta_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m} 2(y^{(i)}-X_b^{(i)}\theta)\cdot(-1) \\ \sum_{i=1}^{m} 2(y^{(i)}-X_b^{(i)}\theta)\cdot(-X_1^{(i)}) \\ \sum_{i=1}^{m} 2(y^{(i)}-X_b^{(i)}\theta)\cdot(-X_2^{(i)}) \\ \ldots \\ \sum_{i=1}^{m} 2(y^{(i)}-X_b^{(i)}\theta)\cdot(-X_n^{(i)}) \end{pmatrix} = 2\cdot\begin{pmatrix} \sum_{i=1}^{m} (X_b^{(i)}\theta-y^{(i)}) \\ \sum_{i=1}^{m} (X_b^{(i)}\theta-y^{(i)})\cdot X_1^{(i)} \\ \sum_{i=1}^{m} (X_b^{(i)}\theta-y^{(i)})\cdot X_2^{(i)} \\ \ldots \\ \sum_{i=1}^{m} (X_b^{(i)}\theta-y^{(i)})\cdot X_n^{(i)} \end{pmatrix}$$

线性回归中使用梯度下降法

目标：使 $\sum_{i=1}^{m}(y^{(i)}-\theta_0-\theta_1X_1^{(i)}-\theta_2X_2^{(i)}-\ldots-\theta_nX_n^{(i)})^2$ 尽可能小

$$\nabla J(\theta)=\begin{pmatrix} \partial J/\partial\theta_0 \\ \partial J/\partial\theta_1 \\ \partial J/\partial\theta_2 \\ \ldots \\ \partial J/\partial\theta_n \end{pmatrix}=\frac{2}{m}\cdot\begin{pmatrix} \sum_{i=1}^{m}(X_b^{(i)}\theta-y^{(i)}) \\ \sum_{i=1}^{m}(X_b^{(i)}\theta-y^{(i)})\cdot X_1^{(i)} \\ \sum_{i=1}^{m}(X_b^{(i)}\theta-y^{(i)})\cdot X_2^{(i)} \\ \ldots \\ \sum_{i=1}^{m}(X_b^{(i)}\theta-y^{(i)})\cdot X_n^{(i)} \end{pmatrix}$$

线性回归中使用梯度下降法

目标：使 $\frac{1}{m}\sum_{i=1}^{m}(y^{(i)}-\hat{y}^{(i)})^2$ 尽可能小

$$J(\theta)=MSE(y,\hat{y})$$

有时取：$J(\theta)=\frac{1}{2m}\sum_{i=1}^{m}(y^{(i)}-\hat{y}^{(i)})^2$

$$\nabla J(\theta)=\begin{pmatrix} \partial J/\partial\theta_0 \\ \partial J/\partial\theta_1 \\ \partial J/\partial\theta_2 \\ \dots \\ \partial J/\partial\theta_n \end{pmatrix}=\frac{2}{m}\cdot\begin{pmatrix} \sum_{i=1}^{m}(X_b^{(i)}\theta-y^{(i)}) \\ \sum_{i=1}^{m}(X_b^{(i)}\theta-y^{(i)})\cdot X_1^{(i)} \\ \sum_{i=1}^{m}(X_b^{(i)}\theta-y^{(i)})\cdot X_2^{(i)} \\ \dots \\ \sum_{i=1}^{m}(X_b^{(i)}\theta-y^{(i)})\cdot X_n^{(i)} \end{pmatrix}$$

线性回归中使用梯度下降法

目标：使 $\frac{1}{m}\sum_{i=1}^{m}(y^{(i)}-\hat{y}^{(i)})^2$ 尽可能小

$$J(\theta)=MSE(y,\hat{y})$$

$$\nabla J(\theta)=\begin{pmatrix} \partial J/\partial\theta_0 \\ \partial J/\partial\theta_1 \\ \partial J/\partial\theta_2 \\ \dots \\ \partial J/\partial\theta_n \end{pmatrix}=\frac{2}{m}\cdot\begin{pmatrix} \sum_{i=1}^{m}(X_b^{(i)}\theta-y^{(i)}) \\ \sum_{i=1}^{m}(X_b^{(i)}\theta-y^{(i)})\cdot X_1^{(i)} \\ \sum_{i=1}^{m}(X_b^{(i)}\theta-y^{(i)})\cdot X_2^{(i)} \\ \dots \\ \sum_{i=1}^{m}(X_b^{(i)}\theta-y^{(i)})\cdot X_n^{(i)} \end{pmatrix}$$

实践：线性回归使用梯度下降法训练

线性回归中梯度下降法的向量化

线性回归中使用梯度下降法

$$\nabla J(\theta) = \frac{2}{m} \cdot \begin{pmatrix} \sum_{i=1}^{m} (X_b^{(i)}\theta - y^{(i)}) \\ \sum_{i=1}^{m} (X_b^{(i)}\theta - y^{(i)}) \cdot X_1^{(i)} \\ \sum_{i=1}^{m} (X_b^{(i)}\theta - y^{(i)}) \cdot X_2^{(i)} \\ \dots \\ \sum_{i=1}^{m} (X_b^{(i)}\theta - y^{(i)}) \cdot X_n^{(i)} \end{pmatrix} = \frac{2}{m} \cdot \begin{pmatrix} \sum_{i=1}^{m} (X_b^{(i)}\theta - y^{(i)}) \cdot X_0^{(i)} \\ \sum_{i=1}^{m} (X_b^{(i)}\theta - y^{(i)}) \cdot X_1^{(i)} \\ \sum_{i=1}^{m} (X_b^{(i)}\theta - y^{(i)}) \cdot X_2^{(i)} \\ \dots \\ \sum_{i=1}^{m} (X_b^{(i)}\theta - y^{(i)}) \cdot X_n^{(i)} \end{pmatrix}$$

线性回归中使用梯度下降法

$$= \frac{2}{m} \cdot \begin{pmatrix} \sum_{i=1}^{m} (X_b^{(i)}\theta - y^{(i)}) \cdot X_0^{(i)} \\ \sum_{i=1}^{m} (X_b^{(i)}\theta - y^{(i)}) \cdot X_1^{(i)} \\ \sum_{i=1}^{m} (X_b^{(i)}\theta - y^{(i)}) \cdot X_2^{(i)} \\ \dots \\ \sum_{i=1}^{m} (X_b^{(i)}\theta - y^{(i)}) \cdot X_n^{(i)} \end{pmatrix}$$

$$\frac{2}{m} \cdot (\ X_b^{(1)}\theta - y^{(1)},\quad X_b^{(2)}\theta - y^{(2)},\quad X_b^{(3)}\theta - y^{(3)},\quad \dots\quad X_b^{(m)}\theta - y^{(m)}\ )$$

$$\cdot \begin{pmatrix} X_0^{(1)} & X_1^{(1)} & X_2^{(1)} & \dots & X_n^{(1)} \\ X_0^{(2)} & X_1^{(2)} & X_2^{(2)} & \dots & X_n^{(2)} \\ X_0^{(3)} & X_1^{(3)} & X_2^{(3)} & \dots & X_n^{(3)} \\ \dots & \dots & \dots & \dots & \dots \\ X_0^{(m)} & X_1^{(m)} & X_2^{(m)} & \dots & X_n^{(m)} \end{pmatrix} = \frac{2}{m} \cdot (X_b\theta - y)^T \cdot X_b$$

$$\frac{2}{m} \cdot X_b^T \cdot (X_b\theta - y)$$

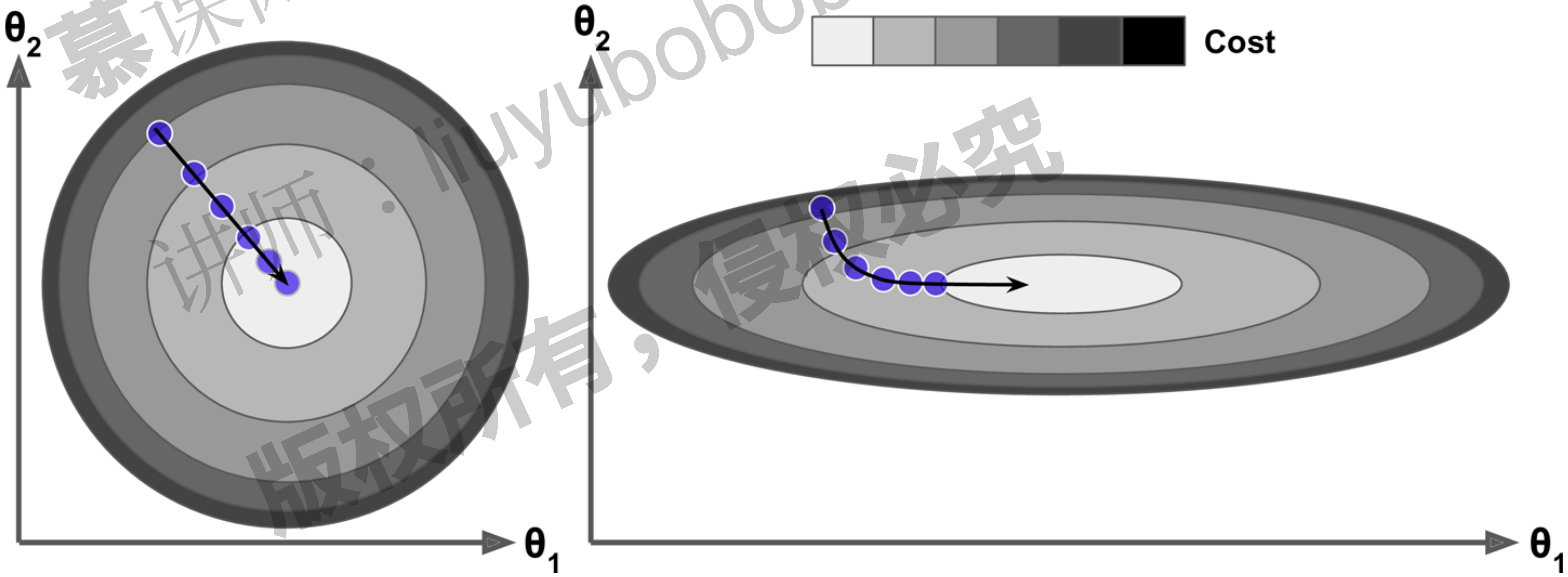
线性回归中使用梯度下降法

$$\nabla J(\theta) = \frac{2}{m} \cdot \begin{pmatrix} \sum_{i=1}^{m} (X_b^{(i)}\theta - y^{(i)}) \cdot X_0^{(i)} \\ \sum_{i=1}^{m} (X_b^{(i)}\theta - y^{(i)}) \cdot X_1^{(i)} \\ \sum_{i=1}^{m} (X_b^{(i)}\theta - y^{(i)}) \cdot X_2^{(i)} \\ \cdots \\ \sum_{i=1}^{m} (X_b^{(i)}\theta - y^{(i)}) \cdot X_n^{(i)} \end{pmatrix} = \frac{2}{m} \cdot X_b^T \cdot (X_b\theta - y)$$

实践：线性回归使用向量化的梯度下降法训练

梯度下降法与数据归一化

使用梯度下降法前，最好进行数据归一化



实践：数据归一化后，使用梯度下降法训练

随机梯度下降法

批量梯度下降法 Batch Gradient Descent

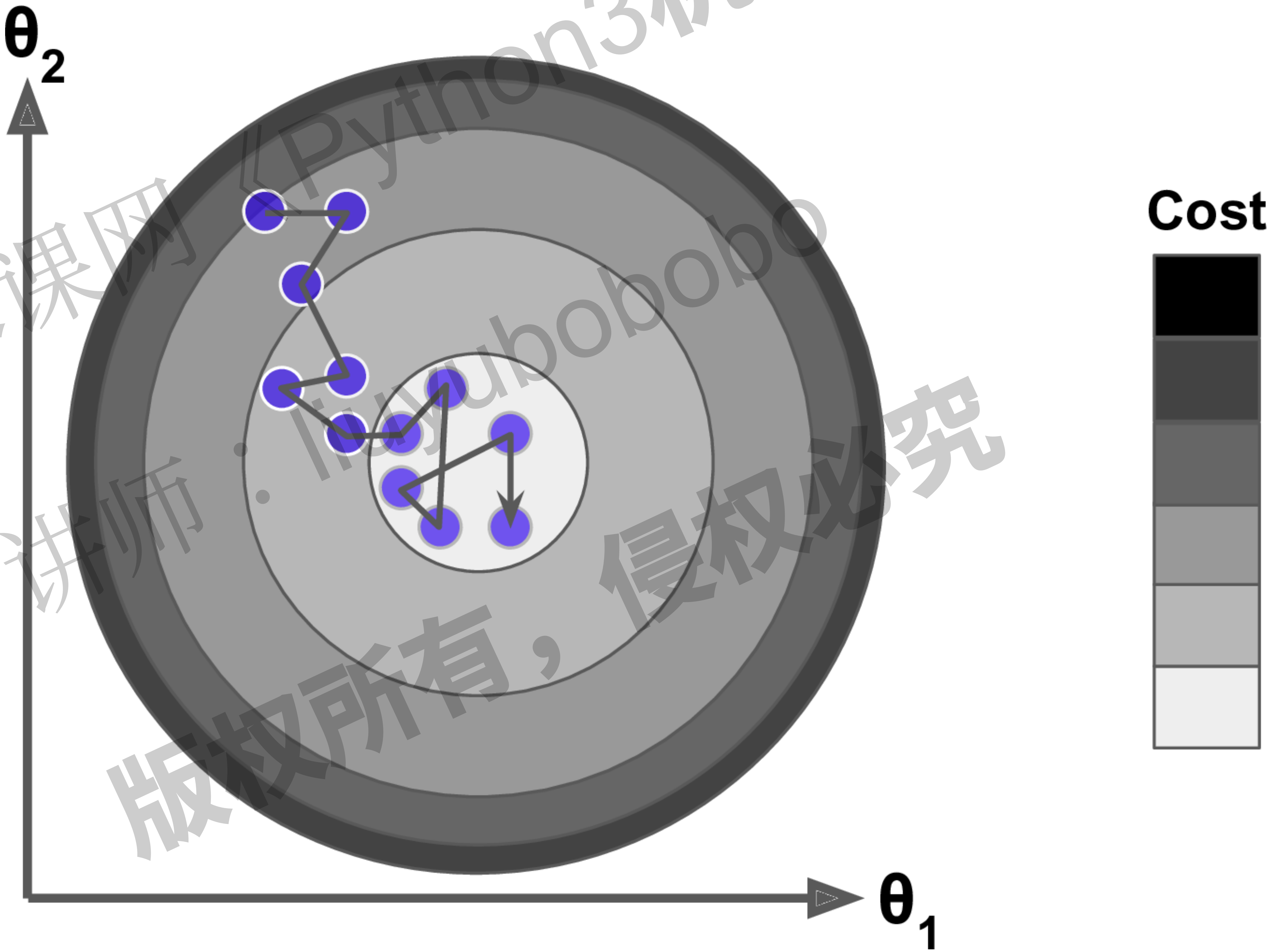
$$\nabla J(\theta) = \frac{2}{m} \cdot \begin{pmatrix} \sum_{i=1}^{m}(X_b^{(i)}\theta - y^{(i)}) \cdot X_0^{(i)} \\ \sum_{i=1}^{m}(X_b^{(i)}\theta - y^{(i)}) \cdot X_1^{(i)} \\ \sum_{i=1}^{m}(X_b^{(i)}\theta - y^{(i)}) \cdot X_2^{(i)} \\ \cdots \\ \sum_{i=1}^{m}(X_b^{(i)}\theta - y^{(i)}) \cdot X_n^{(i)} \end{pmatrix} = \frac{2}{m} \cdot X_b^T \cdot (X_b\theta - y)$$

随机梯度下降法 Stochastic Gradient Descent

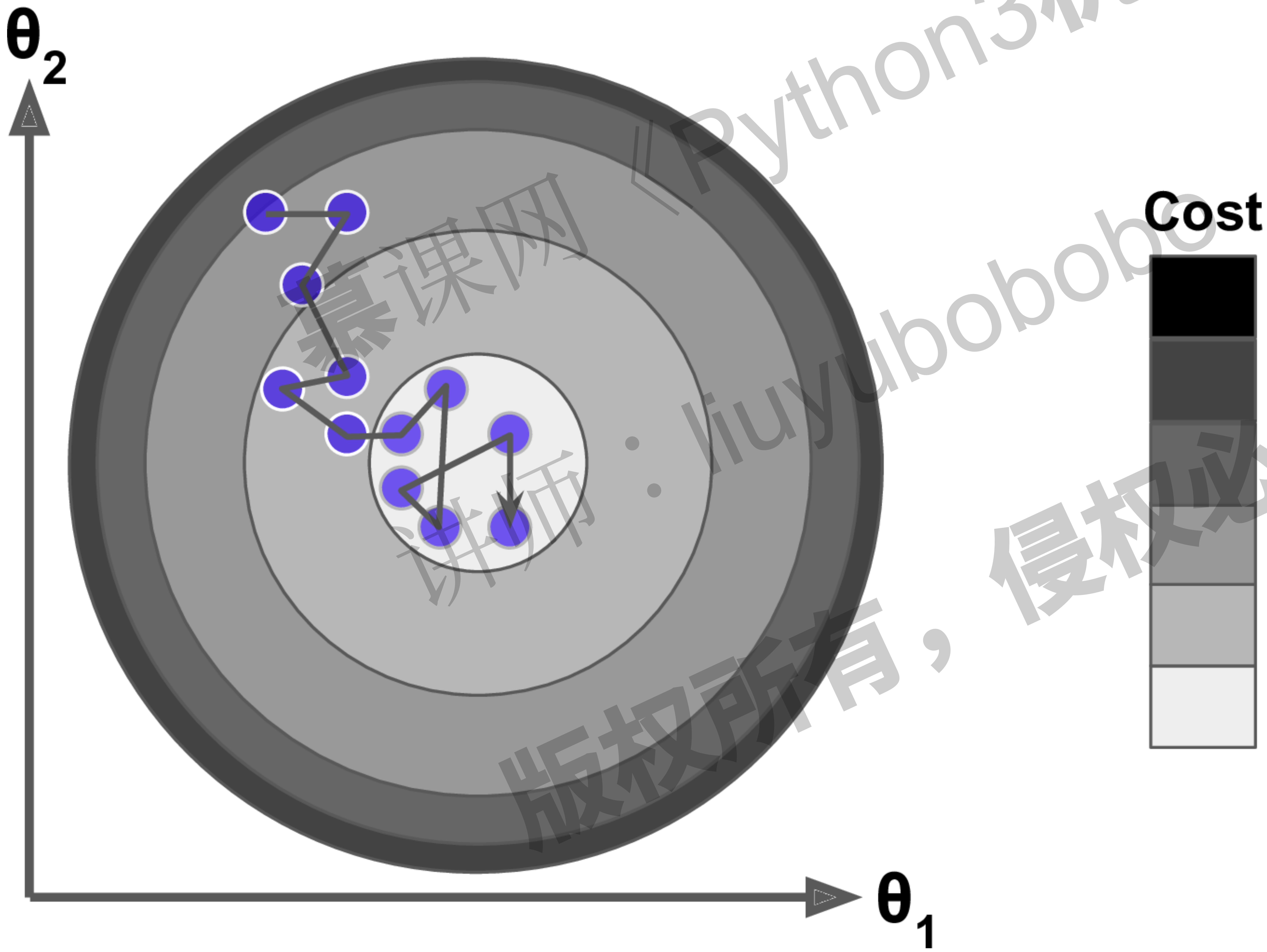
$$= \frac{2}{m} \cdot \begin{pmatrix} \sum_{i=1}^{m}(X_b^{(i)}\theta - y^{(i)}) \cdot X_0^{(i)} \\ \sum_{i=1}^{m}(X_b^{(i)}\theta - y^{(i)}) \cdot X_1^{(i)} \\ \sum_{i=1}^{m}(X_b^{(i)}\theta - y^{(i)}) \cdot X_2^{(i)} \\ \dots \\ \sum_{i=1}^{m}(X_b^{(i)}\theta - y^{(i)}) \cdot X_n^{(i)} \end{pmatrix}$$

$$2 \cdot \begin{pmatrix} (X_b^{(i)}\theta - y^{(i)}) \cdot X_0^{(i)} \\ (X_b^{(i)}\theta - y^{(i)}) \cdot X_1^{(i)} \\ (X_b^{(i)}\theta - y^{(i)}) \cdot X_2^{(i)} \\ \dots \\ (X_b^{(i)}\theta - y^{(i)}) \cdot X_n^{(i)} \end{pmatrix} = 2 \cdot (X_b^{(i)})^T \cdot (X_b^{(i)}\theta - y^{(i)})$$

随机梯度下降法 Stochastic Gradient Descent



随机梯度下降法 Stochastic Gradient Descent



$$\eta = \frac{1}{i\_iters}$$

$$\eta = \frac{1}{i\_iters + b}$$

$$\eta = \frac{a}{i\_iters + b}$$

随机梯度下降法 Stochastic Gradient Descent

模拟退火的思想

$$\eta = \frac{t_0}{i\_iters + t_1}$$

实践：线性回归使用随机梯度下降法训练

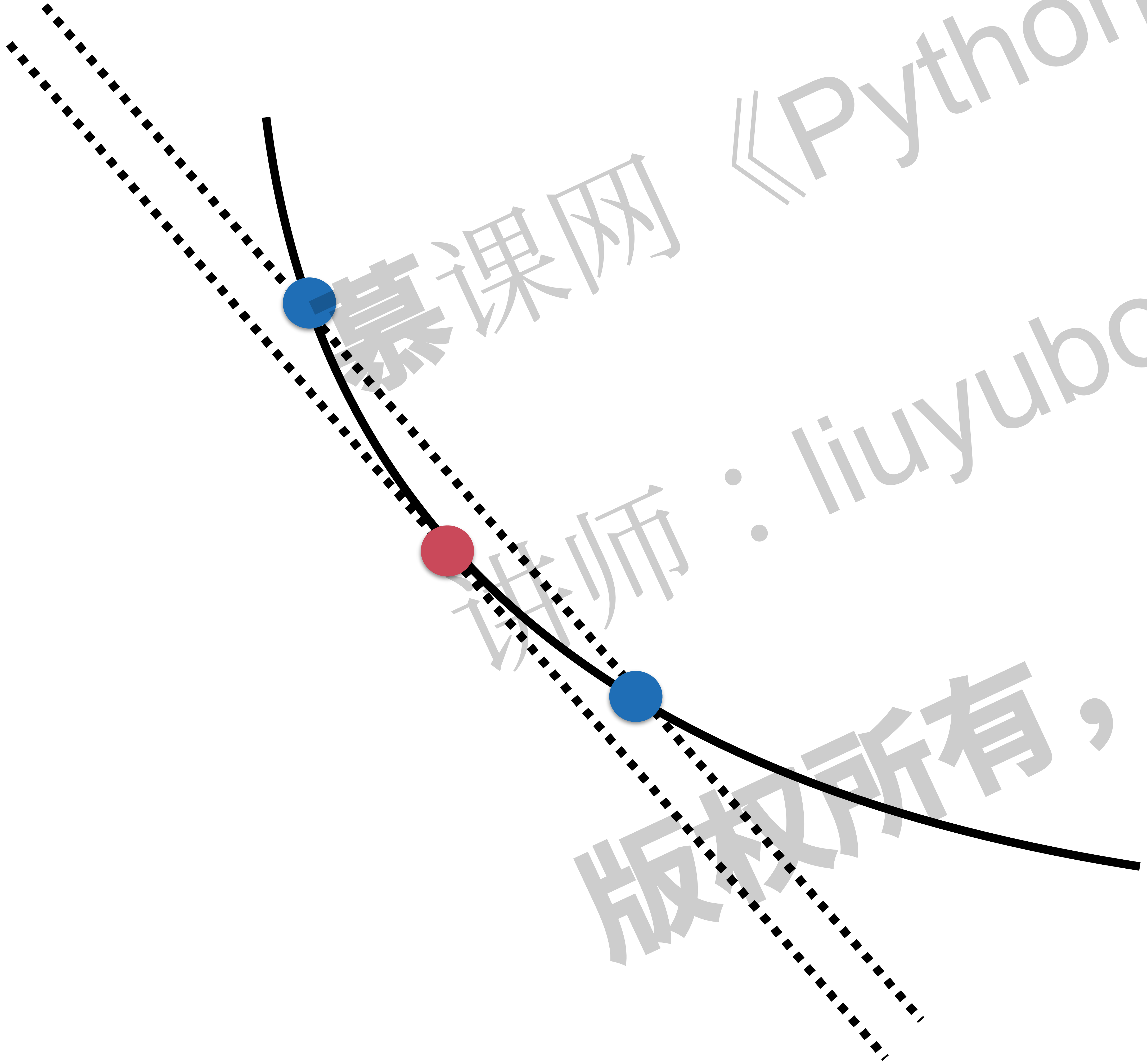
实践：封装我们自己的随机梯度下降法

scikit-learn中的随机梯度下降法

更多有关梯度下降法的讨论

关于梯度的调试

关于梯度的调试



$$\frac{dJ}{d\theta} = \frac{J(\theta+\varepsilon) - J(\theta-\varepsilon)}{2\varepsilon}$$

关于梯度的调试

$$\theta = (\theta_0, \theta_1, \theta_2, \ldots, \theta_n)$$

$$\frac{\partial J}{\partial \theta} = (\frac{\partial J}{\partial \theta_0}, \frac{\partial J}{\partial \theta_1}, \frac{\partial J}{\partial \theta_2}, \ldots, \frac{\partial J}{\partial \theta_n})$$

$$\theta_0^+ = (\theta_0 + \varepsilon, \theta_1, \theta_2, \ldots, \theta_n)$$

$$\theta_0^- = (\theta_0 - \varepsilon, \theta_1, \theta_2, \ldots, \theta_n)$$

$$\frac{\partial J}{\partial \theta_0} = \frac{J(\theta_0^+) - J(\theta_0^-)}{2\varepsilon}$$

关于梯度的调试

$$\theta = (\theta_0, \theta_1, \theta_2, \ldots, \theta_n)$$

$$\frac{\partial J}{\partial \theta} = (\frac{\partial J}{\partial \theta_0}, \frac{\partial J}{\partial \theta_1}, \frac{\partial J}{\partial \theta_2}, \ldots, \frac{\partial J}{\partial \theta_n})$$

$$\theta_1^+ = (\theta_0, \theta_1 + \varepsilon, \theta_2, \ldots, \theta_n)$$

$$\theta_1^- = (\theta_0, \theta_1 - \varepsilon, \theta_2, \ldots, \theta_n)$$

$$\frac{\partial J}{\partial \theta_1} = \frac{J(\theta_1^+) - J(\theta_1^-)}{2\varepsilon}$$

实践：关于梯度的调试

梯度下降法

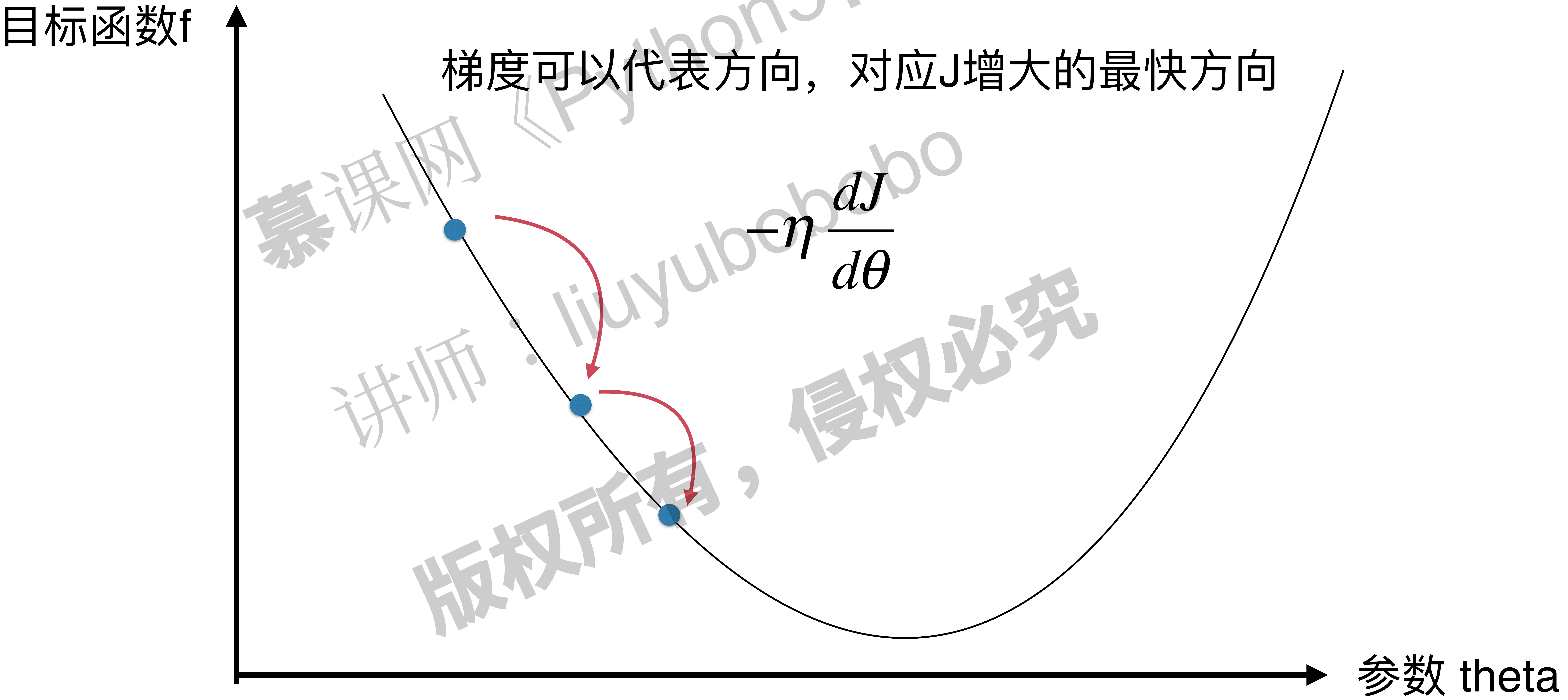
- 批量梯度下降法 Batch Gradient Descent
- 随机梯度下降法 Stochastic Gradient Descent
- 小批量梯度下降法 Mini-Batch Gradient Descent

随机

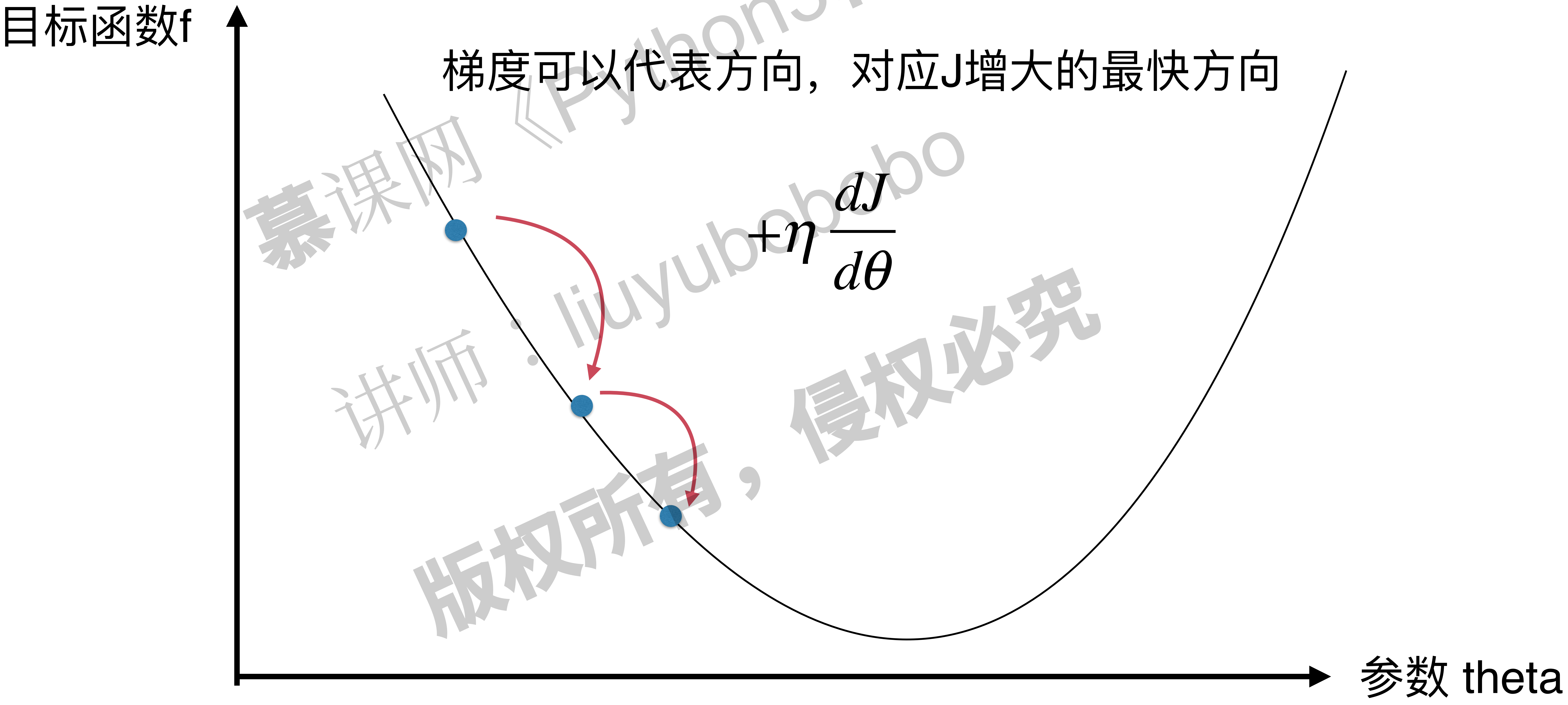
- 跳出局部最优解
- 更快的运行速度
- 机器学习领域很多算法都要使用随机的特点：

  随机搜索；随机森林

梯度上升法



梯度上升法



梯度上升法

梯度可以代表方向，对应J增大的最快方向

$$+\eta \frac{dJ}{d\theta}$$

梯度下降法

- 不是一个机器学习算法
- 是一种基于搜索的最优化方法
- 作用：最小化一个损失函数
- 梯度上升法：最大化一个效用函数

其他

欢迎大家关注我的个人公众号：是不是很酷

Python 3 玩儿转机器学习

liuyubobobo