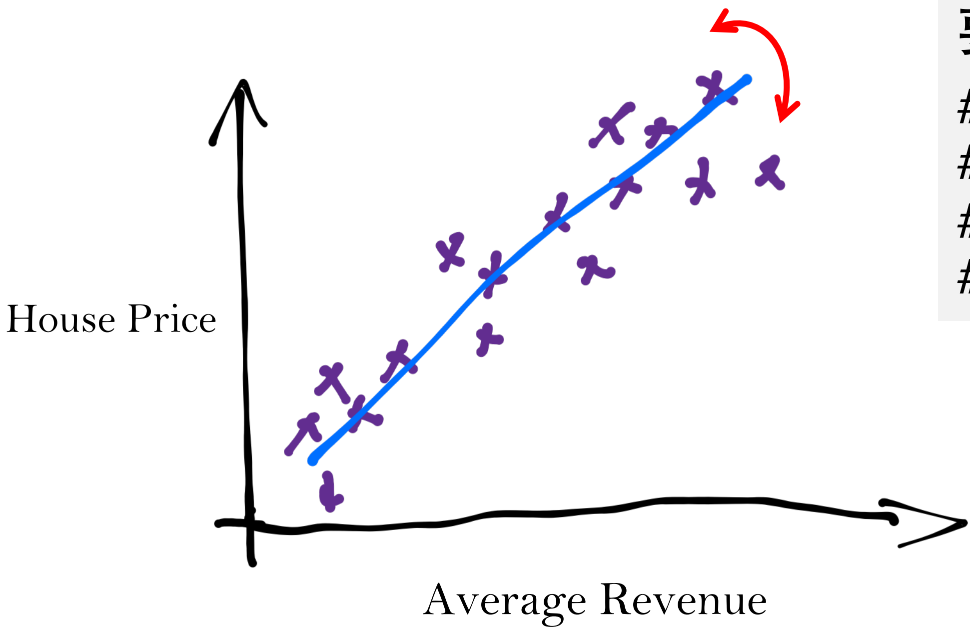
机器学习

第二讲: 线性回归

2025 春

授课老师：顾小东

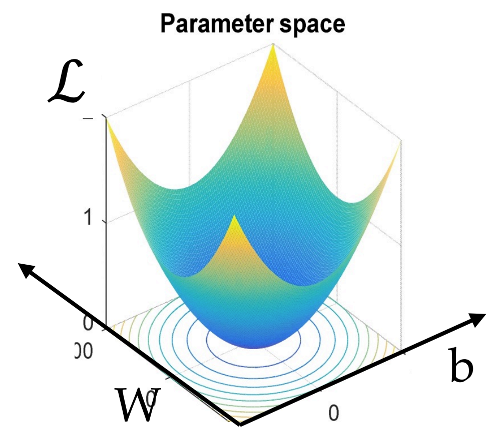
回忆: 机器学习核心要素



要素:

#1 数据 (Experience)
#2 模型 (Hypothesis)
#3 损失函数 (Objective)
#4 优化算法 (Improve)

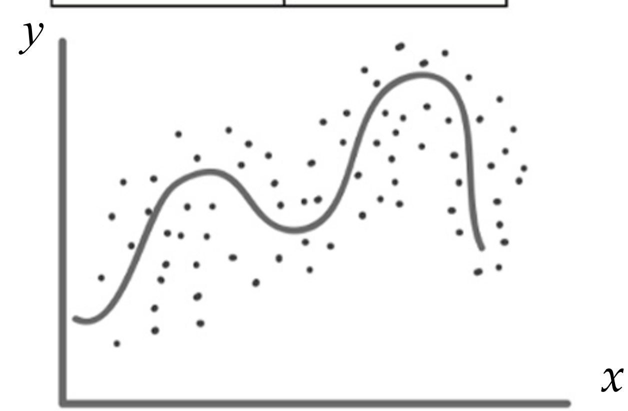
$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}(\theta|\mathcal{D})$$



机器学习中的回归问题

机器学习
- 监督学习
  - 回归 (✔)
  - 分类
  - …
- 无监督学习
- 强化学习

Advertisement	Sales
$90	$1000
$120	$1300
$150	$1800
$100	$1200
$130	$1380
$200	??



Regression: predicts real-valued labels

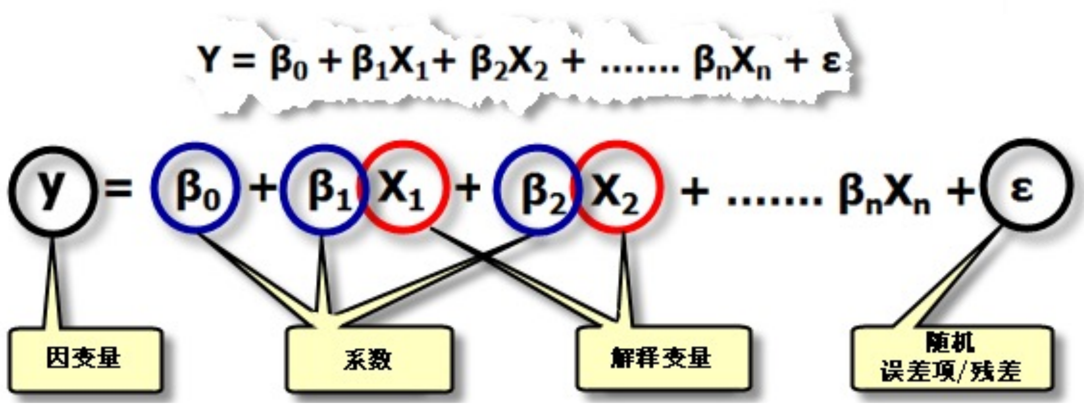
回归问题

Why regression?

Regress the true value of a statistical variable through many experimentally observed values.

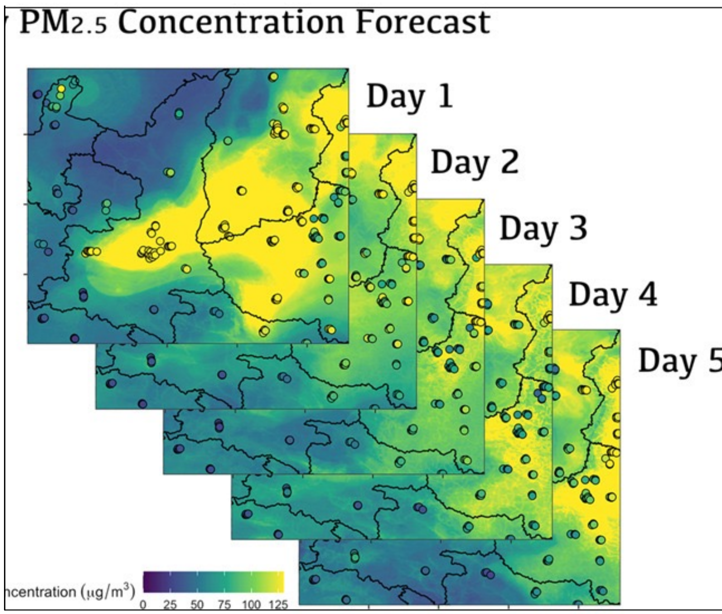
What is regression?

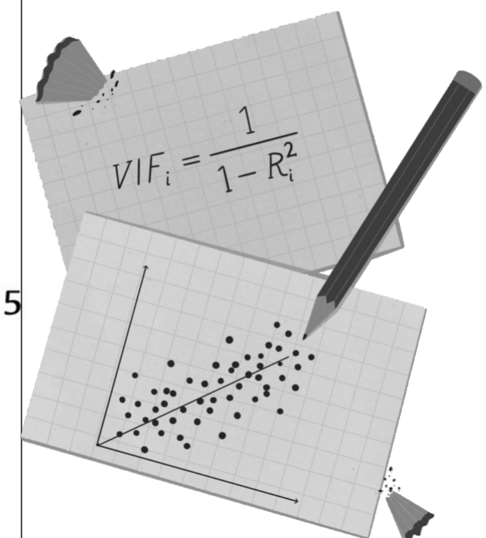
A function that describe the relationship between one **dependent** variable and a series of other (**independent**) variables.

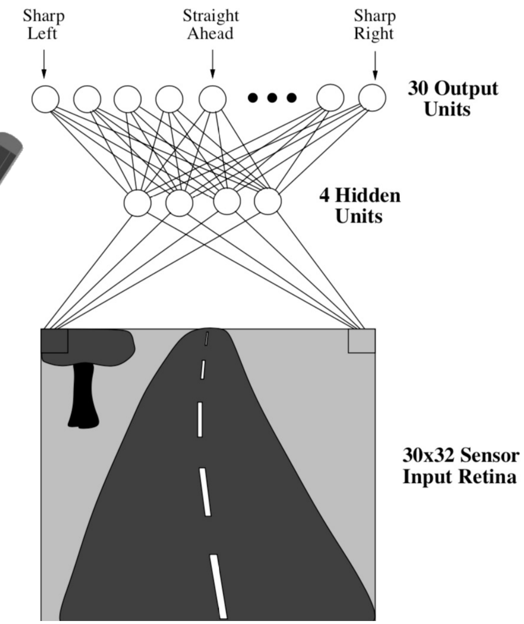


回归问题的应用

预测 | 归因分析 | 控制 | …



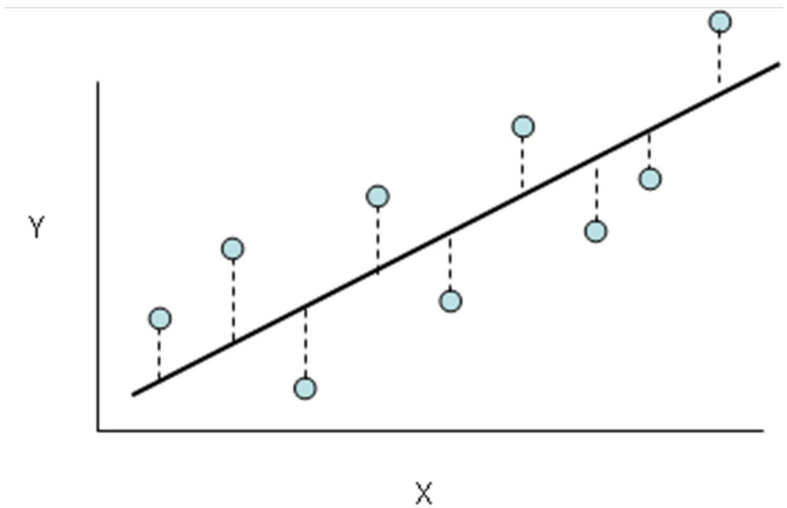


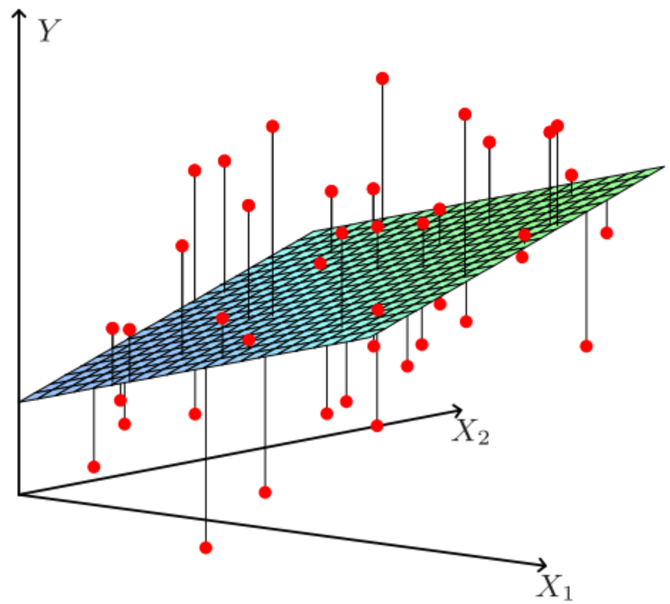


线性回归

使用一个线性函数进行回归

$$y = f(x) = \mathbf{w}^T x + w_0$$

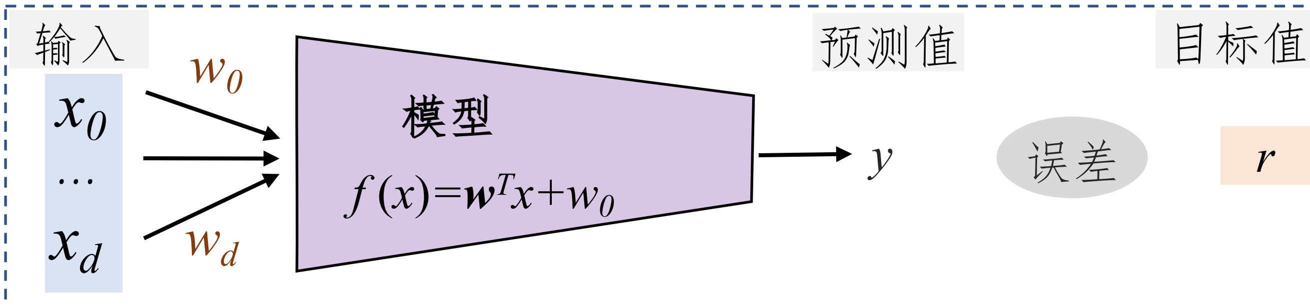




Linear model for regression is a (d+1)-dimensional hyperplane

模型结构

线性回归模型的核心是一个简单的**线性函数**



- **训练过程**:
  - 根据数据误差，估算合适的参数 $\boldsymbol{w}$ 和 w_0
- **预测过程**:
  - 对于输入的新数据 x, 计算 $f(x) = \boldsymbol{w}^T x + w_0$.，得到回归值y

损失函数

- 对于任一输入样本x, 模型输出预测值 y. 令 $r \in \mathrm{R}$ 为该样本对应的目标值, 则相应的平方误差定义为:

$$l(\boldsymbol{w}, w_0 \mid x, r) = (r - y)^2$$

- 对于完整数据集 $D = \{(x^{(1)}, r^{(1)}), \ldots, (x^{(\mathrm{N})}, r^{(\mathrm{N})})\}$, 损失函数定义为均方误差 (MSE):

$$L(\boldsymbol{w}, w_0 \mid D) = \frac{1}{2\mathrm{N}} \sum_{\ell=1}^{N} (r^{(\ell)} - y^{(\ell)})^2$$

训练优化

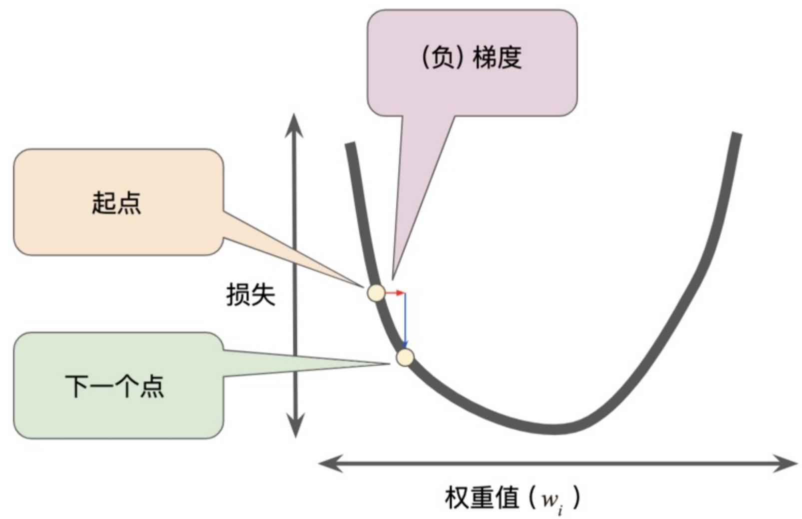
一般采用**梯度下降法**(Gradient Descend)优化损失函数:

- 优化目标:

$$\min_w L(\boldsymbol{w})$$

- 迭代步骤:

$$\boldsymbol{w}_{\mathrm{t}+1} = \boldsymbol{w}_{\mathrm{t}} - \eta_{\mathrm{t}} \frac{\partial L}{\partial w}$$



如何计算 $\dfrac{\partial L}{\partial w}$?

梯度下降法优化模型

$$L(\boldsymbol{w}, w_0 \mid D) = -\, {}^1\!/_{2\mathrm{N}} \sum_{\ell=1}^{N} (r^{(\ell)} - y^{(\ell)})^2$$

怎么求 $\dfrac{\partial L}{\partial w}$?

对于任意w_j (j =1,…,d):

$$\frac{\partial L}{\partial w_j} = -\frac{1}{N}\sum_{\ell} \underbrace{\left(r^{(\ell)} - y^{(\ell)}\right)\frac{\partial y^{(\ell)}}{\partial w_j}}_{\textbf{Chain rule}} = -\frac{1}{N}\sum_{\ell}\left(r^{(\ell)} - y^{(\ell)}\right)x^{(\ell)}$$

迭代规则： $\boldsymbol{w}_{\mathrm{new}} = \boldsymbol{w}_{\mathrm{old}} + \dfrac{1}{N} \sum_{\ell=1}^{N} (r^{(\ell)} - y^{(\ell)})\, x^{(\ell)}$

算法总结

采用梯度下降法训练线性回归模型

Input: $D = \{(\mathrm{x}^{(l)}, \mathrm{r}^{(l)})\}\ (l = 1{:}N)$

for $j = 0, \ldots, \mathrm{d}$

  $w_{\mathrm{j}} \leftarrow rand\ (-0.01, 0.01)$

repeat

  for $j = 0, \ldots, d$

    $\Delta w_{\mathrm{j}} \leftarrow 0$

  for $l = 1, \ldots, N$

    $y \leftarrow 0$

    for $j = 0, \ldots, d$

      $y \leftarrow y + w_{\mathrm{j}} x_{\mathrm{j}}^{(l)}$

    $\Delta w_{\mathrm{j}} \leftarrow \Delta w_{\mathrm{j}} + (r^{(l)} - y) x_{\mathrm{j}}^{(l)}$

  $\Delta w_{\mathrm{j}} = \Delta w_{\mathrm{j}} / \mathrm{N}$

  for $j = 0, \ldots, d$

    $w_{\mathrm{j}} \leftarrow w_{\mathrm{j}} + \eta \Delta w_{\mathrm{j}}$

until convergence

线性回归的矩阵形式

令 $\boldsymbol{X}=\begin{bmatrix}\boldsymbol{x}^{(1)}\\ \boldsymbol{x}^{(2)}\\ \vdots\\ \boldsymbol{x}^{(N)}\end{bmatrix}=\begin{bmatrix}x_0^{(1)} & x_1^{(1)} & x_2^{(1)} & \cdots & x_d^{(1)}\\ x_0^{(2)} & x_1^{(2)} & x_2^{(2)} & \cdots & x_d^{(2)}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ x_0^{(N)} & x_1^{(N)} & x_2^{(N)} & \cdots & x_d^{(N)}\end{bmatrix}$ $\quad \boldsymbol{w}=\begin{bmatrix}w_1\\ w_2\\ \vdots\\ w_d\end{bmatrix}$ $\quad \boldsymbol{r}=\begin{bmatrix}r^{(1)}\\ r^{(2)}\\ \vdots\\ r^{(N)}\end{bmatrix}$

- 预测值: $y=X\boldsymbol{w}=\begin{bmatrix}x^{(1)}w\\ x^{(2)}w\\ \vdots\\ x^{(N)}w\end{bmatrix}$
- 损失函数: $L(w)=\dfrac{1}{2}(r-y)^T(r-y)=\dfrac{1}{2}(r-Xw)^T(r-Xw)$

线性回归的矩阵形式

- **梯度**

$$\frac{\partial L(w)}{\partial w} = -X^T(r - Xw)$$

- **最优参数**

$$\frac{\partial L(w)}{\partial w} = 0 \;\Rightarrow X^T(r - Xw) = 0$$

$$\Rightarrow X^T r = \boldsymbol{X}^T X w$$

$$\Rightarrow \boldsymbol{w}^{\star} = (X^T X)^{-1} X^{\mathrm{T}} r$$

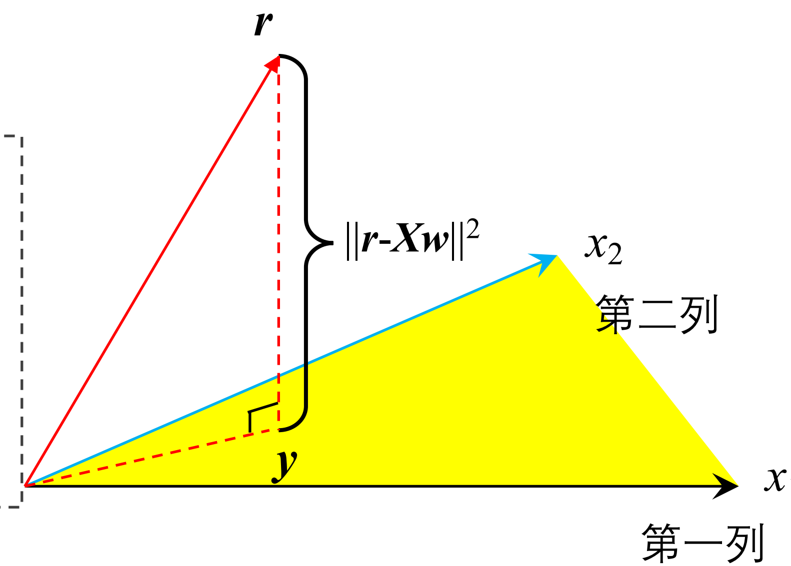
线性回归的矩阵形式

- 将最优参数w^*代入原始模型，得到预测值为

$$y = \underbrace{X(X^T X)^{-1} X^T}_{} r$$
$$= \boldsymbol{H} r$$

几何解释

- 列向量 $[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_d]$ 构成$\mathbb{R}^n$的一个子空间.
- **H** 是向量$\boldsymbol{r}$在该子空间上距离最短的一个投影

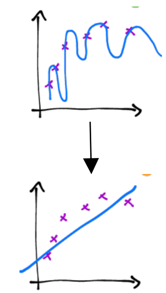


正则化

问题: $X^{\mathrm{T}}X$ 可能不可逆

When some column vectors are not independent (e.g., $\boldsymbol{x}_2=3\boldsymbol{x}_1$), then $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is singular, thus $\boldsymbol{w}^{\star} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{r}$ cannot be directly calculated.

解决方法: 正则化 (对损失函数引入一个惩罚项)



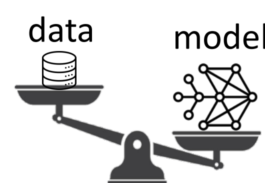
$$L(w)=\frac{1}{2}(r-y)^T(r-y)=\frac{1}{2}(r-Xw)^T(r-Xw)+\frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

新的梯度: $\frac{\partial L(w)}{\partial w}=-X^T(r-Xw)+\lambda w$

新的最优解: $\frac{\partial L(w)}{\partial w}=0 \Rightarrow -X^T(r-Xw)+\lambda w=0$

$$\Rightarrow X^Tr=(X^TX+\lambda \mathrm{I})w$$

$$\Rightarrow w^{\star}=(X^TX+\lambda \mathrm{I})^{-1}X^{\mathrm{T}}r$$



Penalty to model amounts to data augmentation (adding data prior)

敲一敲代码

Tutorial:

Python

https://colab.research.google.com/github/cs231n/cs231n.github.io/blob/master/python-colab.ipynb

Linear regression with Python

https://www.kaggle.com/code/sudhirnl7/linear-regression-tutorial/data?select=insurance.csv

What’ s Next?

Classifications

Find a decision boundary that maximizes the margin between two classes.

Machine Learning
- **Supervised Learning**
  - Regression
  - Classification(✓)
  - …
- Unsupervised Learning
- Reinforcement Learning