
# 模式识别
# Pattern Recognition

李泽榉，复旦大学 生物医学工程与技术创新学院

# 目录

# 任课老师介绍

李泽榉，生物医学工程与技术创新学院
个人主页：https://zerojumpline.github.io/

**研究兴趣：**
医学图像处理，机器学习，计算神经学

**办公室：**
江湾校区交叉学科二号楼C2008

**邮箱：**
zejuli@fudan.edu.cn

**工作经历：**

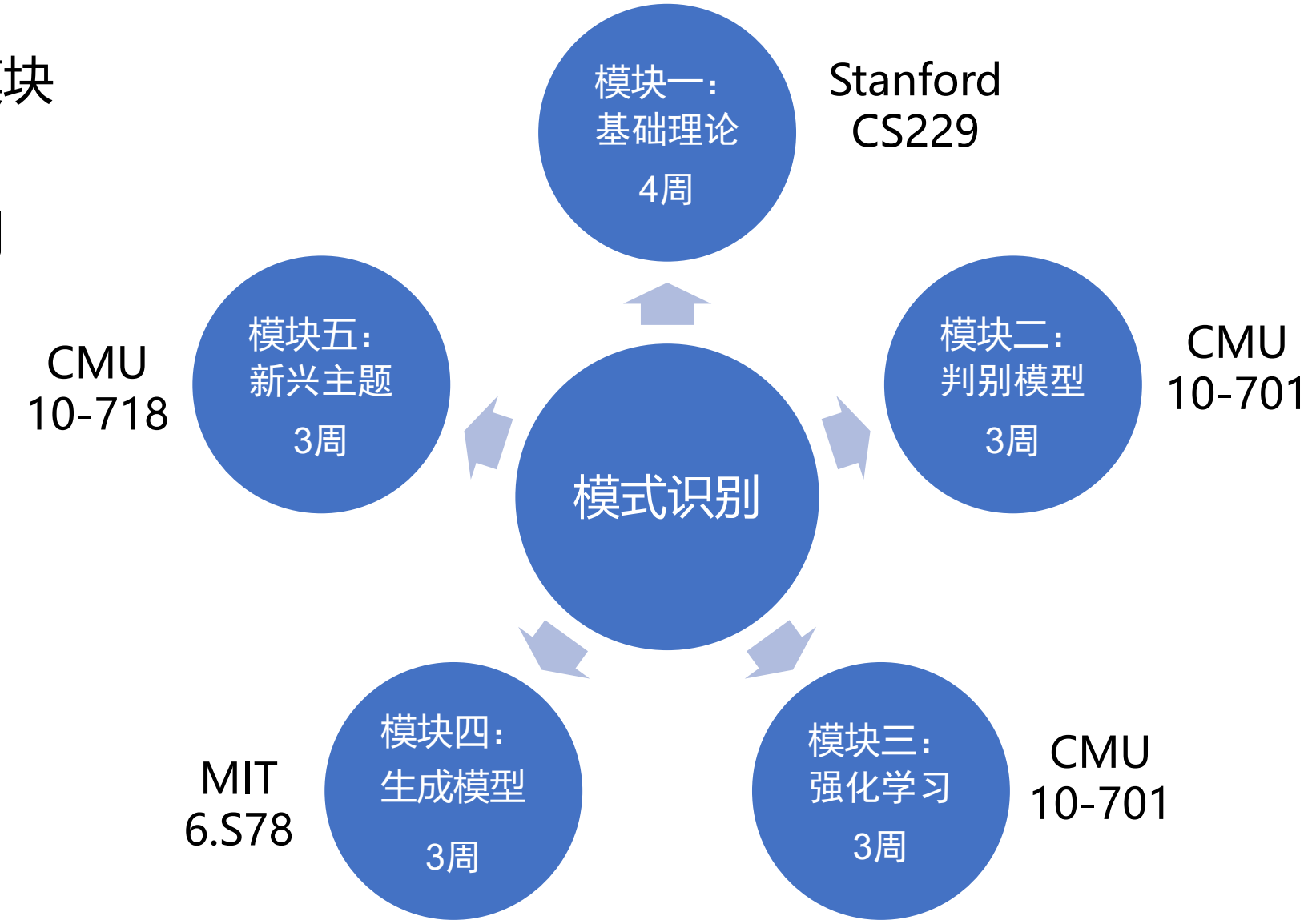
| | | |
|---|---|---|
| 2025至今 | 复旦大学 | 青年研究员 |
| 2023-2025 | 牛津大学 | 博后 |

**学习经历：**

| | | | |
|---|---|---|---|
| 2023 | 博士 | 帝国理工学院 | 计算机科学 |
| 2018 | 硕士 | 复旦大学 | 生物医学工程 |
| 2015 | 本科 | 复旦大学 | 电子工程 |

# 课程设置

- 课程涵盖5个模块
- 每个模块3-4周



# 对同学的期望

**学员基础要求：**

- **身份**：在读博士生
- **设备**：自备笔记本电脑
- **技能**：
  - 熟练掌握Python编程语言
  - 具备数学与人工智能的基础知识
- **网络**：能够稳定访问Google服务（因课程可能使用Colab等工具）

# 对同学的期望

**课程考核要求：**


- **课堂参与**：需进行一次论文分享报告

- **最终成果**：需完成一篇课程论文（形式可以是科学博客文章或学术论文初稿）

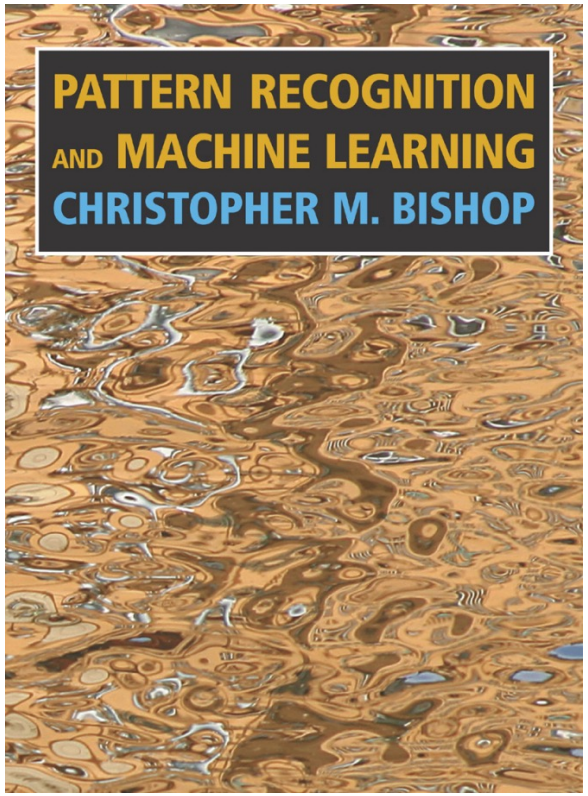
# 基础信息

## 课程网页

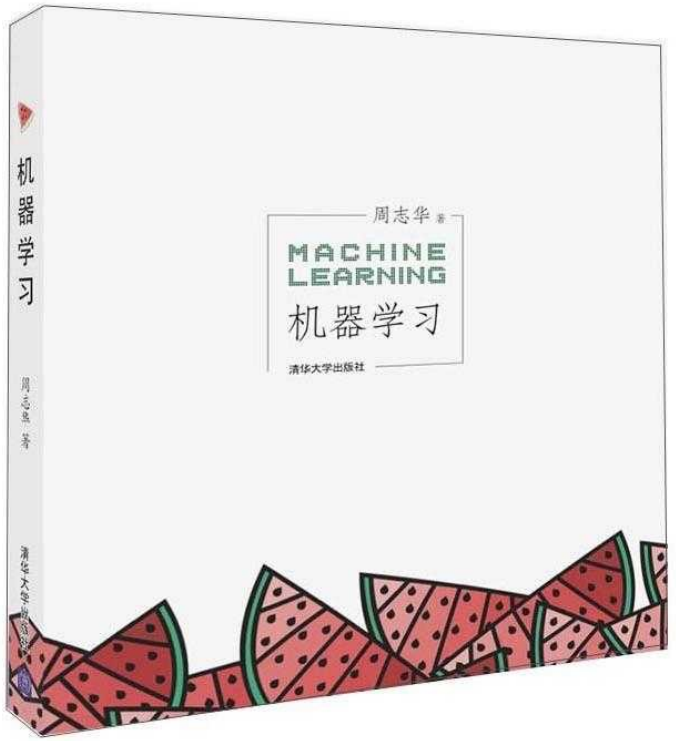
## 课程微信群

# 推荐教材-基础概念

## 基础概念

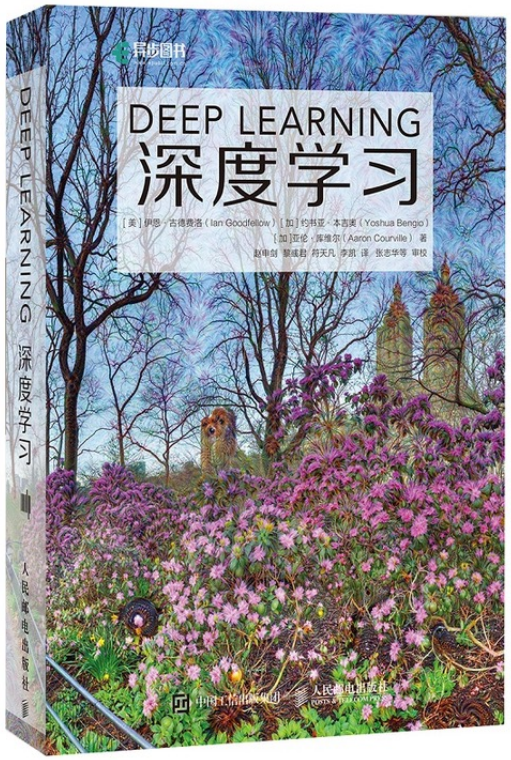



## 中文教材

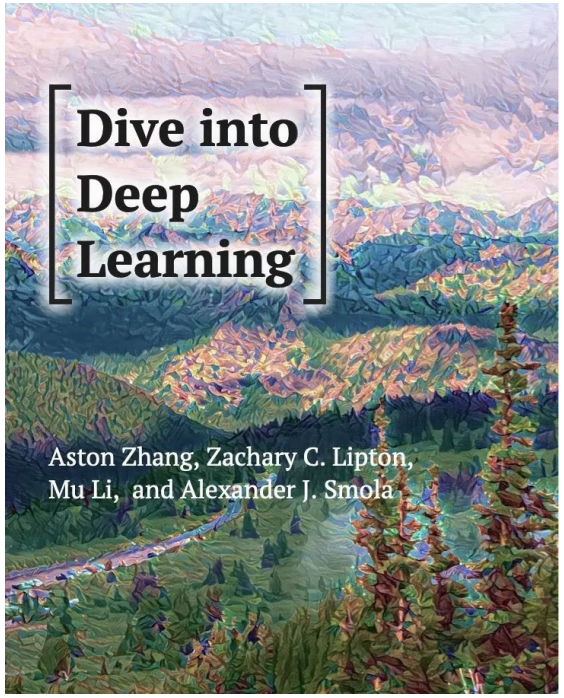


# 推荐教材-实践应用

## 深度学习理论

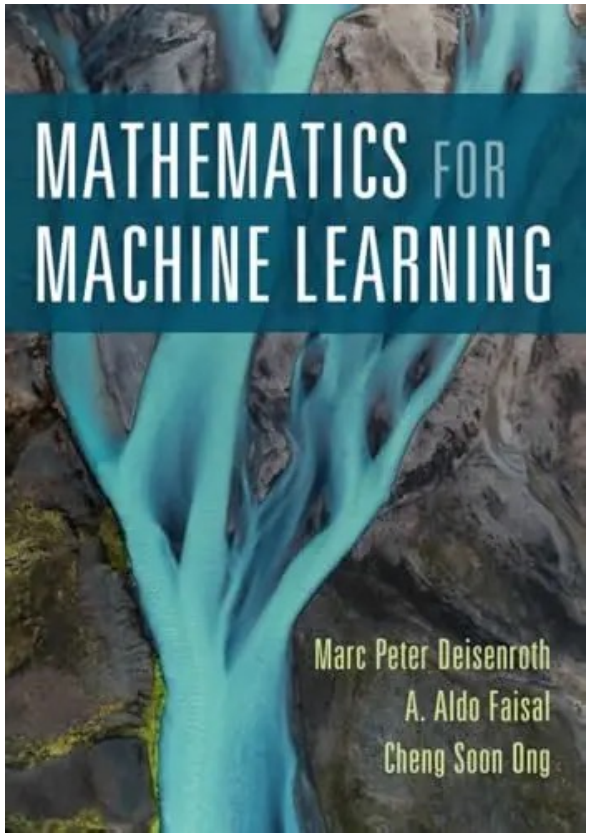



## 深度学习应用

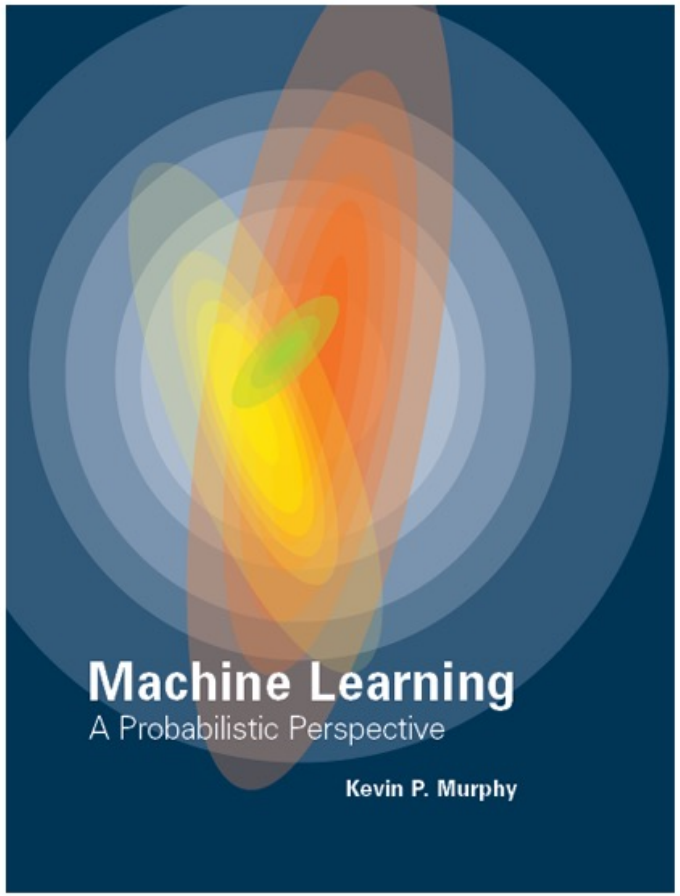


# 推荐教材-数学定义

## 数学定义

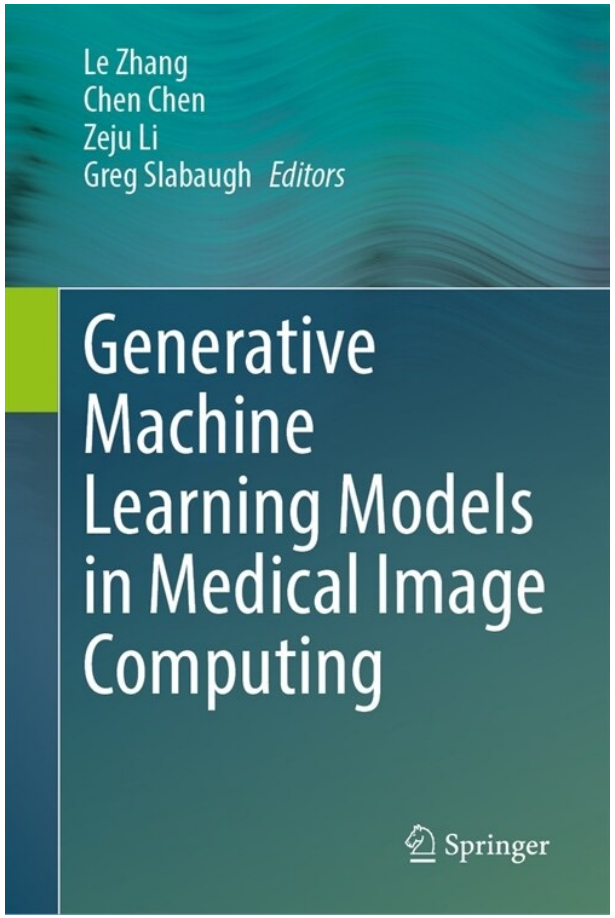



## 概率论定义



# 推荐教材-领域应用

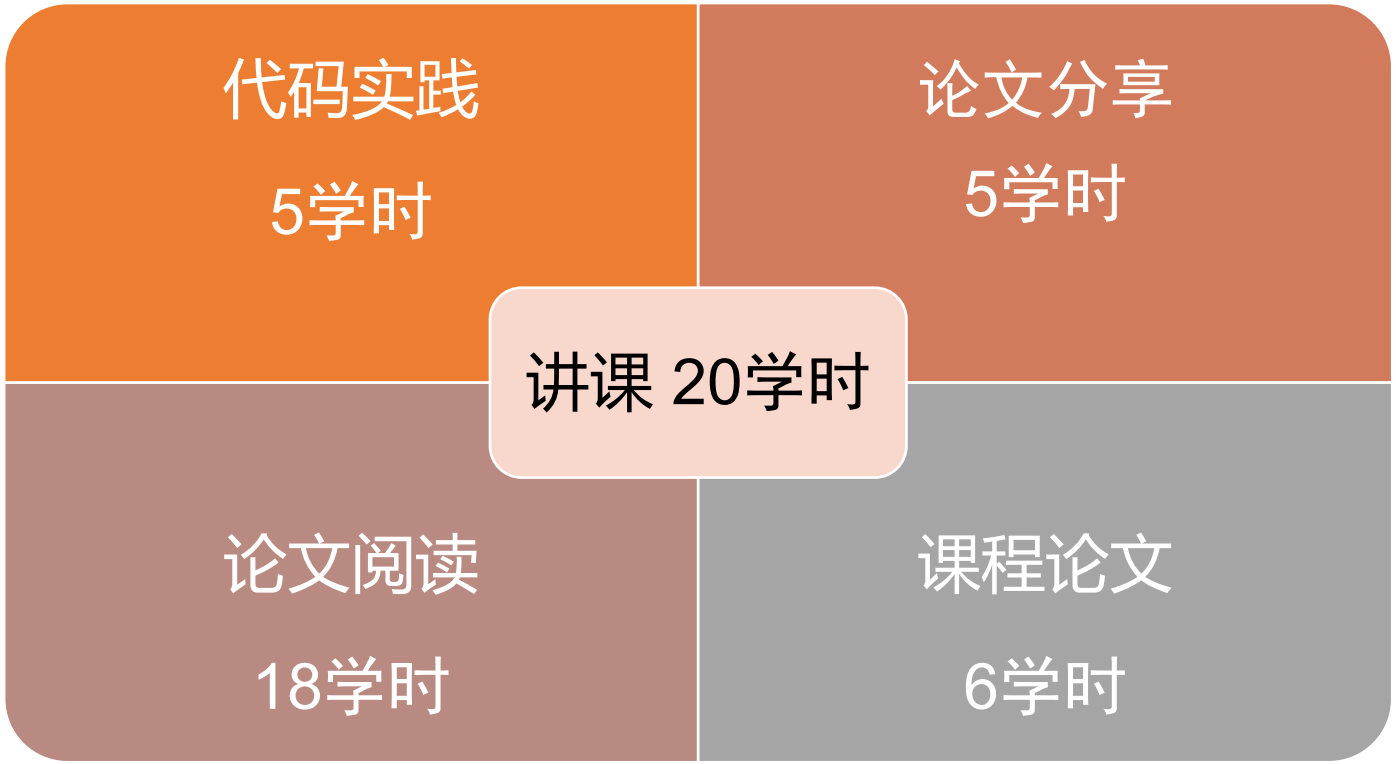
## 生成模型在医学中应用




## 机器学习在MRI中应用

# 课程形式

- 18周课程，54学时一共有四种上课形式



# 讲课

- 任课老师会按照课程大纲授课，但也会进行微调
- 如果对课件感兴趣，可以在上课结束后当天在课程网站上下载

# 代码实践

- 任课老师将提供Jupyter Notebook教程
- 首次代码实践的内容包括PCA、Linear Regression和Gaussian Mixture Models
- 同学们首要目标是理解讲解的内容，代码可以在本地的Jupyter环境或Google Colab上运行
- 因此，不强制要求大家在课内完成所有练习

# 论文分享

- 每个课程模块将安排一次论文分享会
- 由学生分组进行汇报，请同学组队登记
- 每次会议安排2至3组，每组时长15分钟
- 汇报表现将计入课程总成绩

# 论文阅读

- 课程设有18课时的自主文献阅读环节，时长共6周
- 任课老师已将每周对应的阅读论文清单公布在课程主页
- 学生需从中选择**2篇**论文撰写阅读笔记，并与课程论文一并提交（截止日期是第17周）

# 课程论文

- 课程论文可选择以下两种形式**之一**提交（截止日期是第17周）：

- **形式一：论文初稿**
  该形式要求进行一项相对严肃的学术研究。允许对经典论文进行复现，但必须辅以细致的性能分析（如不同超参数、数据集下的表现对比等）。最终成果应有望达到本领域国际研讨会（Workshop）的录用水平。

- **形式二：科学博客**
  该形式要求选择一个前沿概念，用有趣、生动的方式进行阐释，目标读者为普通科研工作者。写作风格和内容质量可参考Distill 或者 ICLR Blog上的文章。

# 评分标准

- **一、论文阅读与分享（共40%）**
- 课程参与（10%）
  - 提交两篇带阅读笔记的论文PDF，每篇占比5%。
- 论文分享（30%）
  - 内容清晰度（10%）
  - 与主题相关度（10%）
  - 演讲表达能力（10%）
- **二、课程作业（60%）**
- 选题意义（20%）
- 研究深度（20%）
- 写作规范（20%）

# 小心!

- 这是由我主讲的第一门课程，讲解上可能还有不清晰之处，请大家见谅
- 课件中也可能存在一些不准确或疏漏，**非常欢迎大家随时提出和讨论**

# 目录

# 基础运算

- x为一个向量，并且有n的元素

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

# 基础运算

- 向量内积（点积）

$$x^T y \in \mathbb{R} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i.$$

- 向量外积

$$xy^T \in \mathbb{R}^{m \times n} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix} = \begin{bmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_n \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_n \\ \vdots & \vdots & \ddots & \vdots \\ x_m y_1 & x_m y_2 & \cdots & x_m y_n \end{bmatrix}.$$

# 基础运算

- A为一个矩阵，m个行和n个列
- 一般表示为m个样本和n个特征

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} | & | & & | \\ a^1 & a^2 & \cdots & a^n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} — & a_1^T & — \\ — & a_2^T & — \\ & \vdots & \\ — & a_m^T & — \end{bmatrix}.$$

# 基础运算

- 矩阵和向量的乘法

$$y = Ax = \begin{bmatrix} — & a_1^T & — \\ — & a_2^T & — \\ & \vdots & \\ — & a_m^T & — \end{bmatrix} x = \begin{bmatrix} a_1^T x \\ a_2^T x \\ \vdots \\ a_m^T x \end{bmatrix}.$$

- 另外的角度

$$y = Ax = \begin{bmatrix} | & | & & | \\ a^1 & a^2 & \cdots & a^n \\ | & | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \\ a^1 \\ \\ \end{bmatrix} x_1 + \begin{bmatrix} \\ a^2 \\ \\ \end{bmatrix} x_2 + \ldots + \begin{bmatrix} \\ a^n \\ \\ \end{bmatrix} x_n \ .$$

# 基础运算

- 矩阵和矩阵的乘法

$$C = AB = \begin{bmatrix} — & a_1^T & — \\ — & a_2^T & — \\ & \vdots & \\ — & a_m^T & — \end{bmatrix} \begin{bmatrix} | & | & & | \\ b^1 & b^2 & \cdots & b^p \\ | & | & & | \end{bmatrix} = \begin{bmatrix} a_1^T b^1 & a_1^T b^2 & \cdots & a_1^T b^p \\ a_2^T b^1 & a_2^T b^2 & \cdots & a_2^T b^p \\ \vdots & \vdots & \ddots & \vdots \\ a_m^T b^1 & a_m^T b^2 & \cdots & a_m^T b^p \end{bmatrix}.$$

- 另外的角度

$$C = AB = \begin{bmatrix} | & | & & | \\ a^1 & a^2 & \cdots & a^p \\ | & | & & | \end{bmatrix} \begin{bmatrix} — & b_1^T & — \\ — & b_2^T & — \\ & \vdots & \\ — & b_p^T & — \end{bmatrix} = \sum_{i=1}^{p} a^i b_i^T \quad .$$

# 基础运算

- 结合律

$$(AB)C = A(BC).$$

- 分配律

$$A(B + C) = AB + AC.$$

# 基础运算

- 方阵 A 的迹（记为 trA）是指矩阵中对角线元素之和：

$$\mathrm{tr}A = \sum_{i=1}^{n} A_{ii}.$$

- 迹具有以下性质：

For $A \in \mathbb{R}^{n\times n}$, $\mathrm{tr}A = \mathrm{tr}A^T$.

For $A, B \in \mathbb{R}^{n\times n}$, $\mathrm{tr}(A+B) = \mathrm{tr}A + \mathrm{tr}B$.

For $A \in \mathbb{R}^{n\times n}$, $t \in \mathbb{R}$, $\mathrm{tr}(tA) = t\,\mathrm{tr}A$.

For $A, B$ such that $AB$ is square, $\mathrm{tr}AB = \mathrm{tr}BA$.

# 基础运算

- 向量的范数代表向量的“长度”
- 最常见的$l_2$范数

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

- $l_1$范数

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

# 矩阵运算

- 向量求导

$$\nabla_x f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}.$$

- 向量求导性质

$$\nabla_x(f(x) + g(x)) = \nabla_x f(x) + \nabla_x g(x).$$

For $t \in \mathbb{R}$, $\nabla_x(t\, f(x)) = t\nabla_x f(x)$.

# 矩阵运算

- 矩阵求导，对于 $f : \mathbb{R}^{m \times n} \to \mathbb{R}$

$$\nabla_A f(A) \in \mathbb{R}^{m \times n} = \begin{bmatrix} \frac{\partial f(A)}{\partial A_{11}} & \frac{\partial f(A)}{\partial A_{12}} & \cdots & \frac{\partial f(A)}{\partial A_{1n}} \\ \frac{\partial f(A)}{\partial A_{21}} & \frac{\partial f(A)}{\partial A_{22}} & \cdots & \frac{\partial f(A)}{\partial A_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(A)}{\partial A_{m1}} & \frac{\partial f(A)}{\partial A_{m2}} & \cdots & \frac{\partial f(A)}{\partial A_{mn}} \end{bmatrix}$$

i.e., an $m \times n$ matrix with

$$(\nabla_A f(A))_{ij} = \frac{\partial f(A)}{\partial A_{ij}}.$$

# 矩阵运算

- Hessian矩阵 $\mathbb{R}^n \to \mathbb{R}$

$$\nabla_x^2 f(x) \in \mathbb{R}^{n\times n} = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}.$$

i.e., $\nabla_x^2 f(x) \in \mathbb{R}^{n\times n}$, with

$$(\nabla_x^2 f(x))_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}.$$

# 矩阵运算

- 方阵A重写为

$$\begin{bmatrix} — & a_1^T & — \\ — & a_2^T & — \\ & \vdots & \\ — & a_n^T & — \end{bmatrix},$$

- 定义线性变换对

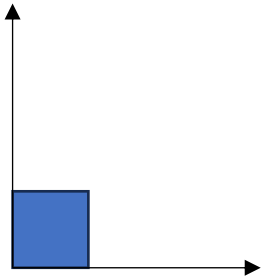
$$S = \{v \in \mathbb{R}^n : v = \sum_{i=1}^{n} \alpha_i a_i \text{ where } 0 \leq \alpha_i \leq 1, i = 1, \ldots, n\}.$$

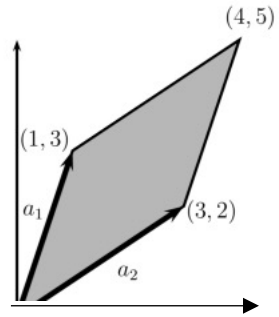
- 方阵A行列式的值是线性变换对 **n 维体积**的缩放比例
  - **|det(A)| = 1**：变换是**保距**的（如旋转、反射），体积不变。
  - **|det(A)| > 1**：变换**放大**了体积。
  - **|det(A)| < 1**：变换**缩小**了体积。
  - **|det(A)| = 0**： **不可逆**的变换

# 矩阵运算

- 例如2x2的方阵A

$$A = \begin{bmatrix} 1 & 3 \\ 3 & 2 \end{bmatrix}$$






- 其行列式计算为： (1)(2) - (3)(3) = 2 - 9 = -7

- **绝对值 | -7 | = 7**：表示任何一块区域的面积经过这个变换后，都会变成原来的 **7倍**。它是一个“放大”变换

- **符号为负**：表示这个变换在放大的同时，还**改变了空间的 “取向”**，列向量的相对方向是**反向**的（顺时针）

# 特征值和特征向量

- 对于一个给定的方阵 $A$，特征值分解试图找到一组**特征向量（v）** 和对应的**特征值（λ）**，使得：

- $A\mathbf{v} = \lambda\mathbf{v}$

- 矩阵 $A$ 对特征向量 $\mathbf{v}$ 的变换效果，等同于只是简单地将其拉长或缩短了 $\lambda$ 倍（如果 $\lambda$ 为负，则还包括反向）

- 特征向量在变换中**方向保持不变**

# 特征值和特征向量

- **特征值求解**
- 1、将定义式改写：

$$A\vec{v} - \lambda\vec{v} = \vec{0}$$
$$(A - \lambda I)\vec{v} = \vec{0}$$

其中 $I$是 n×n 的单位矩阵。我们寻找的是**非零解 v**，这意味着矩阵 **(**$A - \lambda I$**)** 必须是**奇异矩阵**（不可逆矩阵），即它的行列式必须为零。

- **2、求解特征多项式**
令行列式为零，得到**特征方程**：$\det(A - \lambda I)=0$

- **3、求特征方程的根**
解这个关于 λ 的方程，得到的**所有根（解）就是矩阵 A 的全部特征值。**

# 特征值和特征向量

- 将所有这些特征向量和特征值组织起来，就得到了特征值分解的矩阵形式：

- $A = P\Lambda P^{-1}$ （因为$AP = P\Lambda$）

- 其中：
- $\Lambda$ 是一个对角矩阵，对角线上的元素就是特征值 $\lambda_1, \lambda_2, \ldots, \lambda_n$。
- $P$ 的列就是对应的特征向量 $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$。
- $P^{-1}$ 是 $P$ 的逆矩阵。

# 特征值和特征向量

- 矩阵的幂和指数计算是最直接的应用。计算 $A^k$（k很大）非常困难，但利用分解 $A = P\Lambda P^{-1}$，则有：

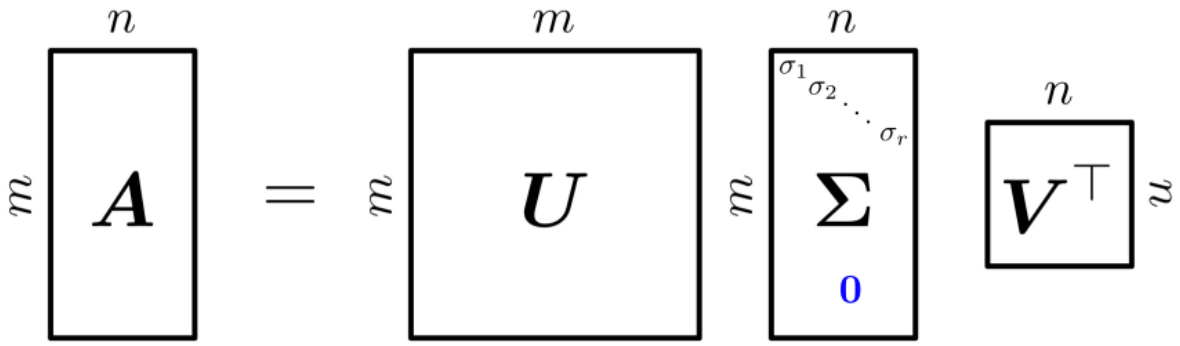
- $A^k = \left(P\Lambda P^{-1}\right)^k = P\Lambda^k P^{-1}$

- 而 $\Lambda^k$ 就是一个对角线上元素为 $\lambda_i^k$ 的对角矩阵，极易计算。

- 但是要求矩阵 $A$ 有 $n$ 个线性无关的特征向量（否则$P^{-1}$ 不存在）

# 奇异值分解（SVD）

$\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\text{rank}(\boldsymbol{A}) = r \leq \min(m, n)$:




- $\boldsymbol{U} \in \mathbb{R}^{m \times m}$ with orthogonal columns vectors $\boldsymbol{u}_i$, $i = 1, \ldots, m$.
- $\boldsymbol{V} \in \mathbb{R}^{n \times n}$ with orthogonal columns vectors $\boldsymbol{v}_j$, $j = 1, \ldots, n$.
- $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times n}$ with $\Sigma_{ii} = \sigma_i \geq 0$ and $\Sigma_{ij} = 0$ for $i \neq j$.
  - $\sigma_i$: singular values; $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0$.
  - $\boldsymbol{u}_i$: left-singular vectors;
    $\boldsymbol{v}_j$: right-singular vectors;

# 奇异值分解（SVD）

- 对于任意一个 $m \times n$ 的实数矩阵 $A$：
- 计算 $A^T A$（一个 $n \times n$ 的方阵，半正定矩阵），它的特征值都是非负的。
- 对 $A^T A$ 进行**特征值分解**，得到特征值和特征向量：

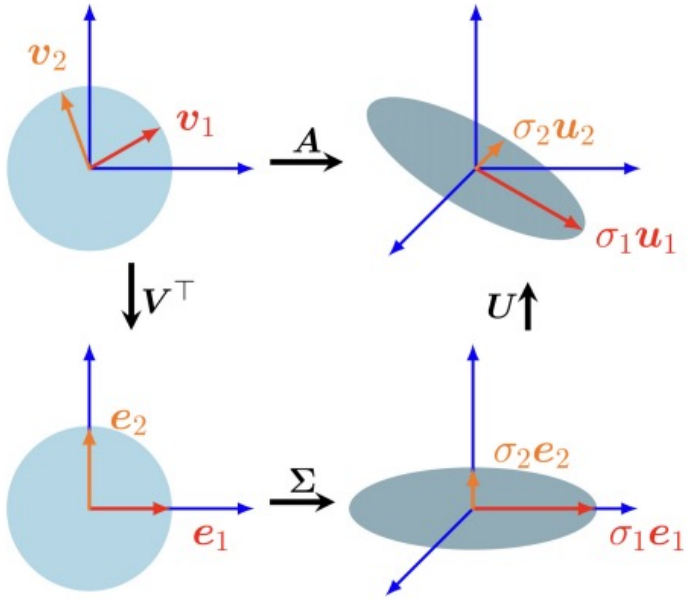
$$(A^T A)\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

  - 这些特征向量 $\mathbf{v}_i$ 是标准正交的，它们构成了 SVD 中的**右奇异向量**$V$。
  - 特征值 $\lambda_i$ 的平方根就是**奇异值**，即 $\sigma_i = \sqrt{\lambda_i}$。

- 类似地，计算 $AA^T$（一个 $m \times m$ 的方阵），并对其进行**特征值分解**，得到的特征向量就是 SVD 中的**左奇异向量**$U$。

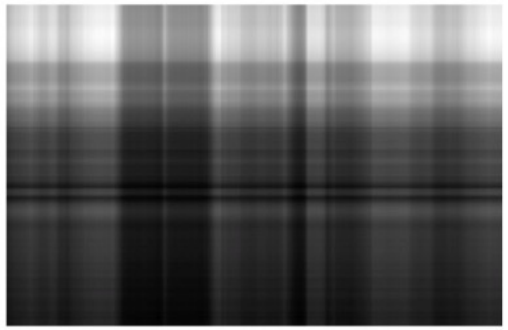
# 奇异值分解（SVD）

$$
\begin{aligned}
\boldsymbol{A} &= \begin{bmatrix} 1 & -0.8 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top} \\
&= \begin{bmatrix} -0.79 & 0 & -0.62 \\ 0.38 & -0.78 & -0.49 \\ -0.48 & -0.62 & 0.62 \end{bmatrix} \begin{bmatrix} 1.62 & 0 \\ 0 & 1.0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -0.78 & 0.62 \\ -0.62 & -0.78 \end{bmatrix}
\end{aligned}
$$

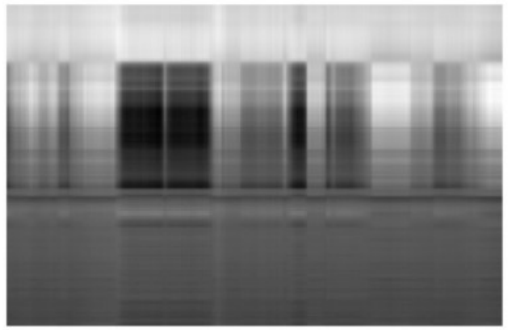


# 奇异值分解（SVD）

(a) Original image $\boldsymbol{A}$.



(b) Rank-1 approximation $\widehat{\boldsymbol{A}}(1)$.



(c) Rank-2 approximation $\widehat{\boldsymbol{A}}(2)$.

(d) Rank-3 approximation $\widehat{\boldsymbol{A}}(3)$.

(e) Rank-4 approximation $\widehat{\boldsymbol{A}}(4)$.

(f) Rank-5 approximation $\widehat{\boldsymbol{A}}(5)$.

# 小结

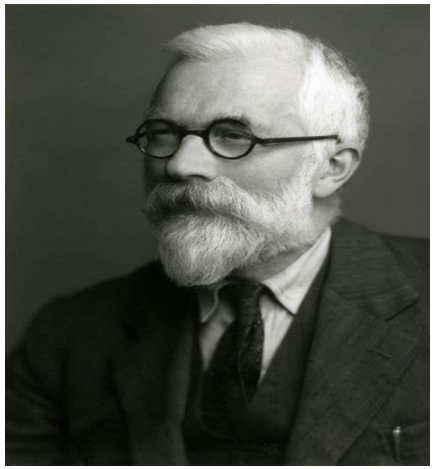
- 熟悉矩阵的定义和运算

- 特征值分解像是为方阵量身定做的“特权分解”，它揭示了矩阵在自身特征向量方向上如何缩放空间。但它要求高（方阵且可对角化），且结果可能不稳定（特征向量不正交、特征值为复数）。

- 奇异值分解（SVD） 是一个“万能分解”，它对任何矩阵都有效。它巧妙地通过计算 $A^TA$ 和 $AA^T$ 的特征值分解，来构建自身的分解。SVD 总是提供一组标准正交基和非负的奇异值，这使得它在数值计算上非常稳定，应用范围广泛。

# 目录

# 机器学习(machine learning)的由来



**化繁为简、大巧不工**

**(The object of statistical methods is the reduction of data)**

**——罗纳德·费希尔（Ronald Fisher）**

机器学习是一种**“数据驱动学习（data-driven learning）”**的范式，它从数据出发来学习数据中所蕴含的模式，对数据进行抽象。统计学家罗纳德·艾尔默·费希尔（Ronald Aylmer Fisher）将这一过程概括为“化繁为简（the object of statistical methods is the reduction of data）”。

1959 年7 月，IBM 公司的工程师阿瑟·塞缪尔（Arthur Samuel） 第一次使用了“机器学习（machine learning）”，将其定义为“**让机器具有不需要明确编程而具有的一种学习能力**（the ability to learn without being explicitly programmed）”的研究，其目标是构造一种学习机器（learning machine），使之像人一样具有自我学习能力， 而非按部就班完成预设任务。

# 机器学习的本质——没有免费午餐定理

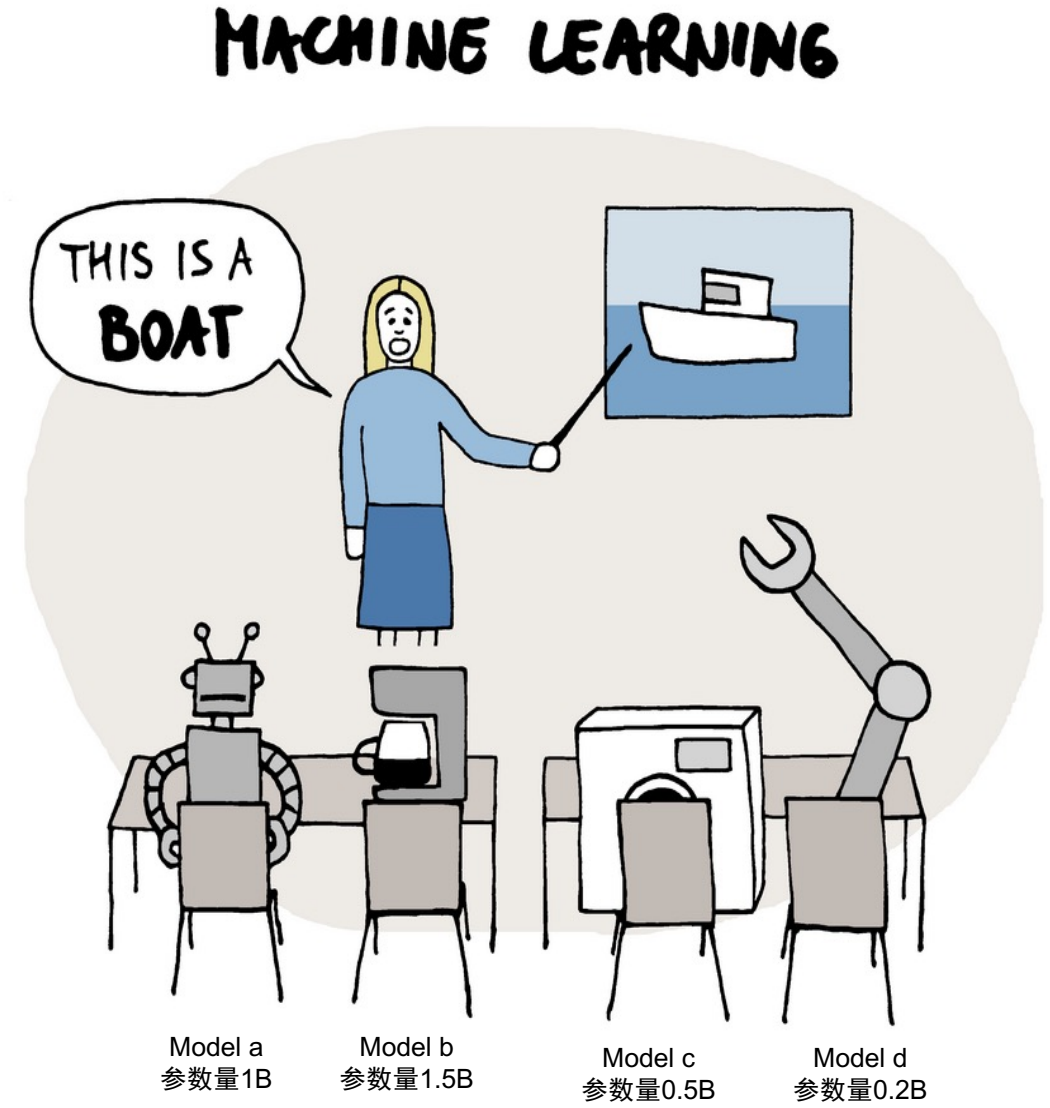
- 为了在训练优化针对不同的任务，往往需要采用不同机器学习模型，1995年，David Wolpert等学者提出了“没有免费午餐定理(No Free Lunch Theorem)”指出：**任何一个机器学习模型如果在一些训练集以外的样本误差小（off-training set error），那么必然在另外一些训练集以外的样本上表现欠佳，任何模型在平均意义上而言其性能都是一样的，即没有放之四海而皆准的最好算法。**
- 似乎这一定理给机器学习带来了一个令人沮丧的事实（即针对某一域的所有问题，所有算法的期望性能是相同的），但是这一定理也告诉我们，离开具体场景和问题去讨论采用哪种机器学习算法是毫无意义的，应该在机器学习中合理引入已有先验假设对模型进行约束，以提升模型效果，如在自然语言理解中引入句子中单词和单词之间的上下文关联（诸如n-gram文法）、在视觉图像分析引入像素点之间的空间依赖（诸如卷积算子）等。

# 机器学习：概念与分类

- 机器学习通过对数据的优化学习，**建立能够刻画数据中所蕴含语义概念或分布结构等信息的模型**。在模型学习过程中，采用合适手段来利用有标签数据或无标签数据， 对模型参数不断进行优化，从而提升模型性能。

- 从数据利用的角度，可将机器学习划分为**监督学习（supervised learning）、无监督学习（unsupervised learning）及半监督学习（semi-supervised learning）**等。

# 机器学习：人类教育




用机器学习算法去训练模型的过程，

**就像你教侄子、侄女算算数** 😄

# 有监督学习：小学、初中、高中

- 老师会手把手的教你
- 每道题都有明确的答案
- 用考试来衡量你学的对不对

# 无监督学习：大学

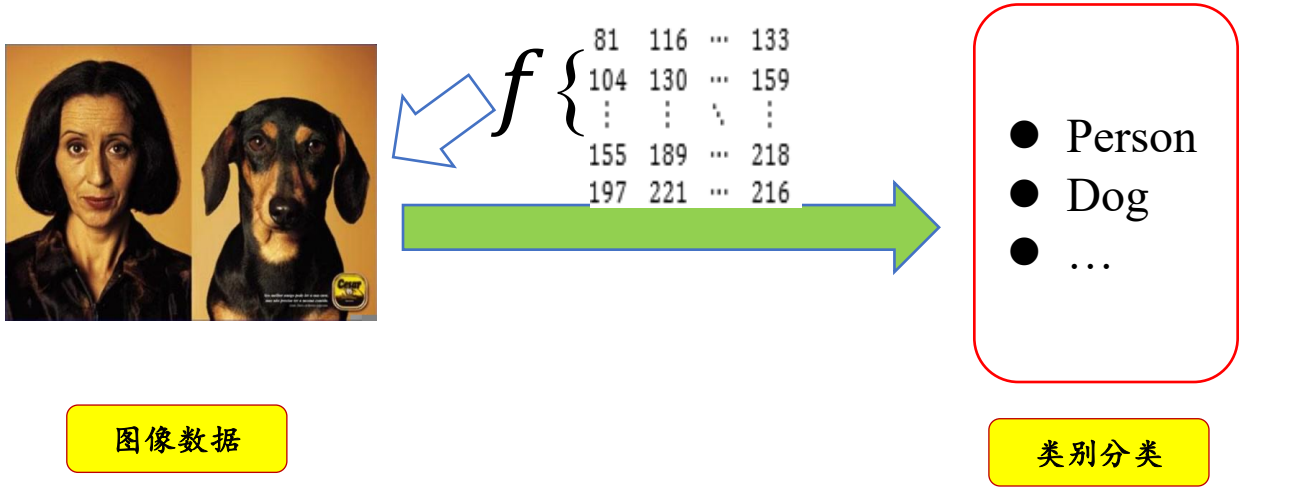
- 老师会告诉你学什么但不会告诉你怎么学
- 没人管你，需要自学
- 并不是每道题都有对与错
- 考试分为A/B/C

# 强化学习：走向社会，参加工作

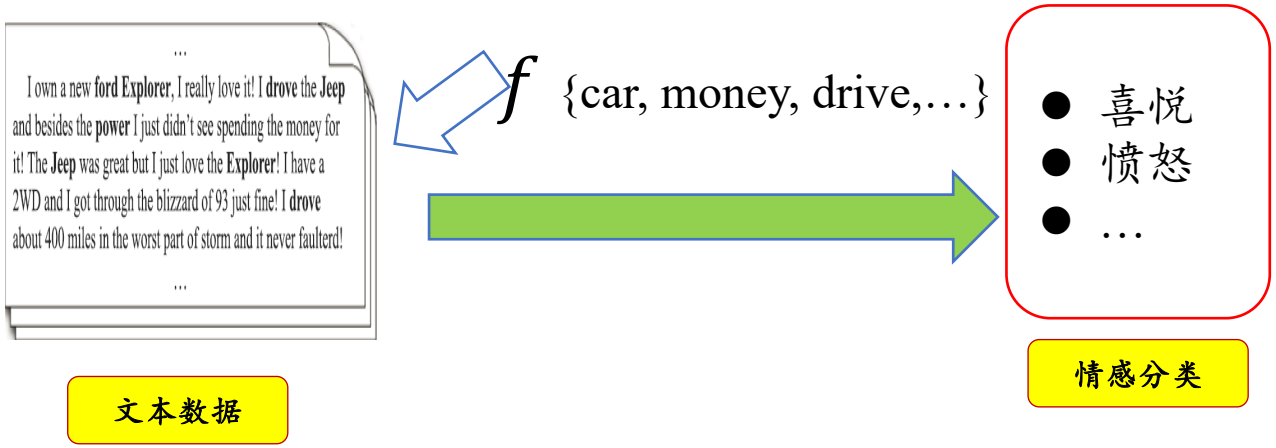
- 没人告诉你每天做什么、怎么做
- 需要自己探索这个世界
- 没有考试，但是会有工资上的差异、升职加薪等奖励差异
- 多年后不同的人之间会出现巨大差异

# 机器学习——监督学习

- 监督学习是一种在实践中运用最为广泛的一种机器学习方法，其目标是给定带有标签信息数据的训练集$\mathcal{D}=\{(x_i,y_i)\}_{i=1}^{n}$，学习一个从输入$x_i$到输出$y_i$的映射。$x_i$可是文档、图像、音频或蛋白质基因等数据或者数据的特征表达，$y_i$为所对应的论文类别、人脸对象、歌曲语音或生命功能等语义内容，其中$\mathcal{D}$被称为训练集，$n$是训练样例的数量。



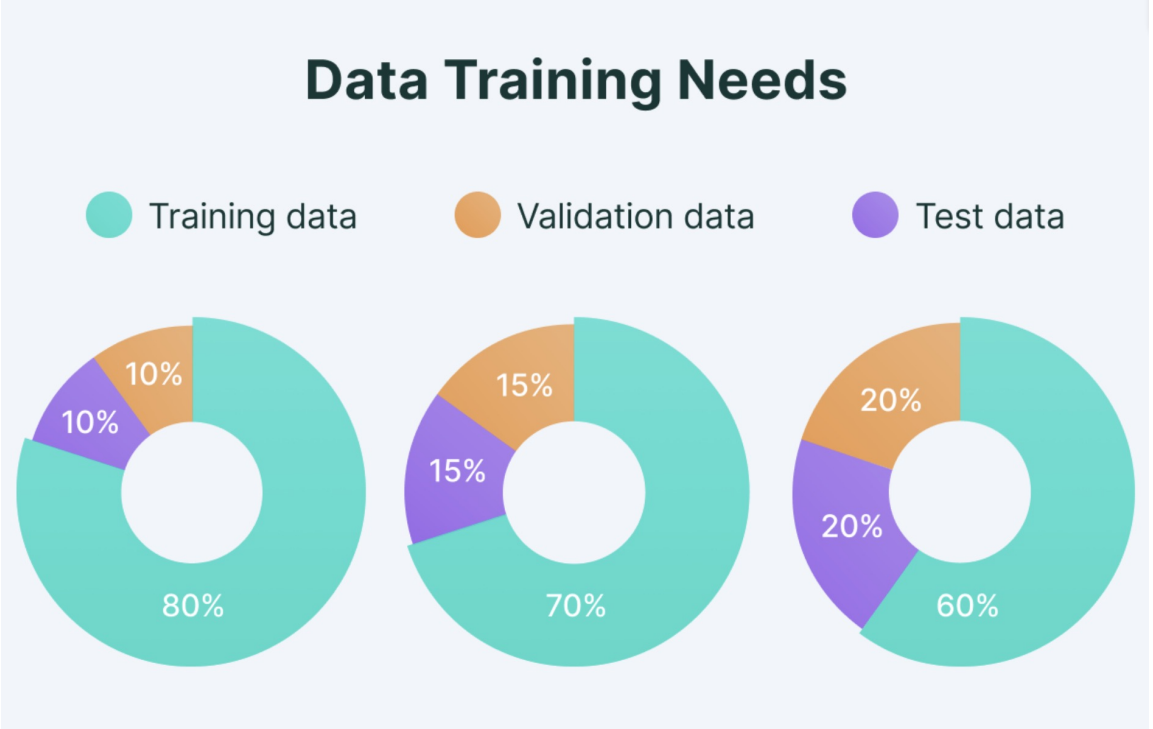

- 监督学习算法从假设空间（hypothesis space）学习得到一个**最优映射函数$f$（又称决策函数）**，映射函数$f$将输入数据映射到语义标注空间，实现数据的分类和识别。无监督学习则是直接从无标签数据$\{x_i, i=1,\ldots,n\}$出发学习映射函数，而半监督学习在学习映射函数过程中使用的一部分数据有标签、一部分数据没有标签。



# 有监督学习——训练集、验证集、测试集

- 一旦在**训练集**上完成了模型参数优化后，需要在测试数据集上对模型性能进行测试。为了在训练优化过程中挑选更好的模型参数，一般可将训练集中一部分数据作为**验证集（validation set）**。在训练集上训练模型的同时会在验证集上对模型进行评估，以便得到最佳参数，最后在**测试集**上进行测试，将测试结果作为模型性能最终结果。

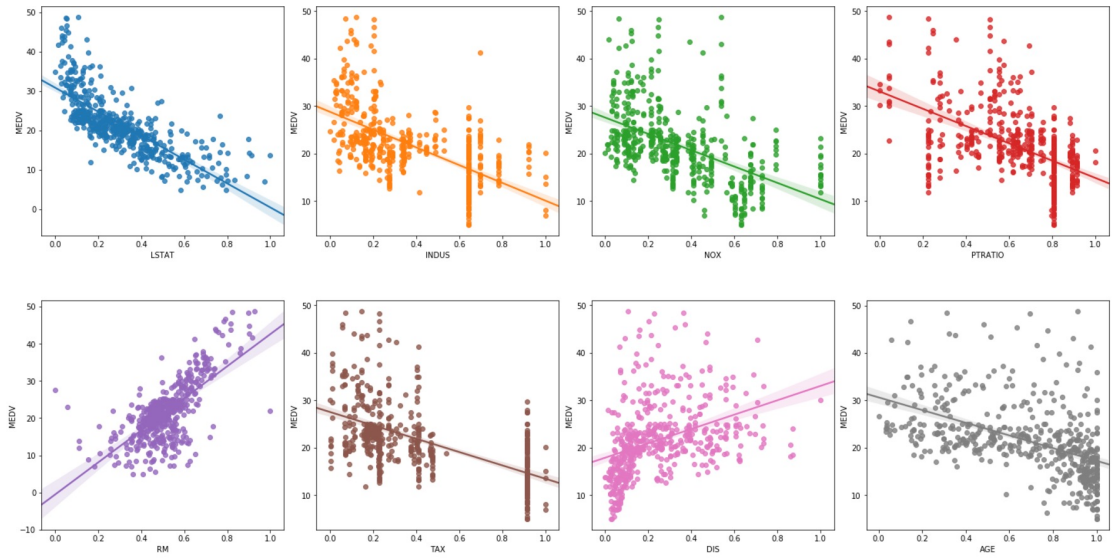
- 要注意的是，训练集、验证集和测试集所包含数据之间没有任何交叉。可以说，训练集用于模型训练（好比学生的练习册）、验证集用于评估模型以调整相应参数（好比学生的模拟考卷或小测验）、测试集用于得到模型的优劣水平（好比真正考试）。




**训练集、验证集和测试集三种数据集中数据比例**

# 有监督学习——经典数据集

波士顿房价数据集包含506个样本，每个样本都有13个特征，这些特征描述了影响房价的各种因素。目标变量是每个区域的房价中位数（MEDV，Median Value）。




**波士顿房价数据集部分特征的分布统计**

**波士顿房价数据集特征描述**

| 特征 | 描述 |
|---|---|
| CRIM | 城镇的犯罪率（每人犯罪率） |
| ZN | 住宅区面积的比例，面积大于25,000平方英尺的区域所占的比例 |
| INDUS | 城镇中非零售商用土地的比例 |
| CHAS | 查尔斯河虚拟变量（如果边界在查尔斯河旁，值为1，否则为0） |
| NOX | 氮氧化物浓度，单位为每千万分之一 |
| RM | 每栋住宅的平均房间数 |
| AGE | 1940年之前建造的自住房比例 |
| DIS | 到五个波士顿就业中心的加权距离 |
| RAD | 靠近高速公路的可达性指数（1~24） |
| TAX | 每10,000美元的财产税率 |
| PTRATIO | 城镇师生比例 |
| B | 指示该区域的黑人居民比例 |
| LSTAT | 低社会经济状态的比例（%） |

# 模型评估与参数估计手段：损失函数

## 泛化能力（generalization）

**在机器学习中，需要保证模型在训练集上所取得性能与在测试集上所取得性能保持一致，即模型具有泛化能力（generalization）。**

表 4.1　常见损失函数的定义

| 损失函数名称 | 损失函数定义 |
| --- | --- |
| 0−1 损失函数 | $\mathrm{Loss}(y_i, f(x_i)) = \begin{cases} 1, f(x_i) \neq y_i \\ 0, f(x_i) = y_i \end{cases}$ |
| 平方损失函数 | $\mathrm{Loss}(y_i, f(x_i)) = (y_i - f(x_i))^2$ |
| 绝对损失函数 | $\mathrm{Loss}(y_i, f(x_i)) = \lvert y_i - f(x_i) \rvert$ |
| 对数损失函数 / 对数似然函数 | $\mathrm{Loss}(y_i, P(y_i \mid x_i)) = -\log P((y_i \mid x_i))$ |

将映射函数记为$f$、第$i$个训练数据记为$(x_i, y_i)$以及$f$对$x_i$的预测结果记为$\hat{y}_i$（即$\hat{y}_i = f(x_i)$，可定义损失函数$Loss(f(x_i), y_i)$来估量预测值$\hat{y}_i$和真实值$y_i$之间差异。很显然，在训练过程中希望映射函数在训练集上累加差异最小，即$\min \sum_{i=1}^{n} Loss(f(x_i), y_i)$。

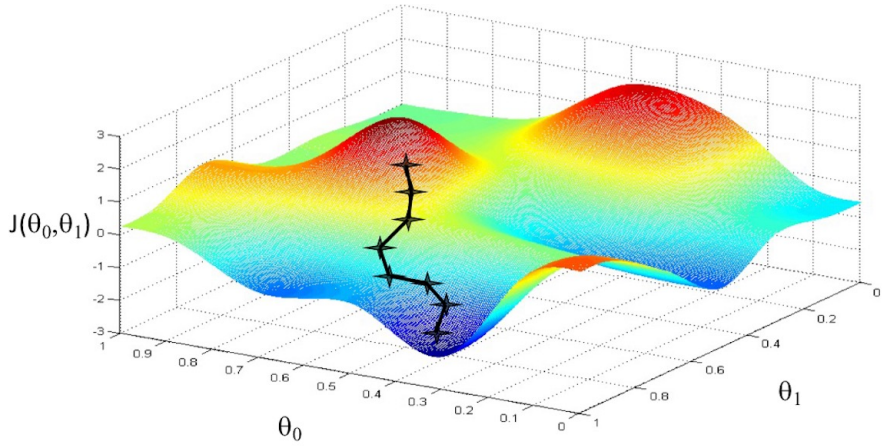
# 优化方法：梯度下降

- 梯度下降法是一种通过不断调整参数来最小化损失函数（或目标函数）的方法。
- 在每一步，算法通过计算损失函数相对于模型参数的梯度（即导数），然后根据梯度的方向更新参数。
- 目标是通过多次迭代找到损失函数的最小值点（即局部最小值或全局最小值）。

梯度下降法公式：

$$\omega_{t+1} = \omega_t - \eta_t \cdot \nabla L(\omega_t)$$

- $\omega_t$：模型参数
- $\eta_t$：学习率（步长），控制每次更新的幅度
- $\nabla L(\omega_t)$：损失函数的梯度（导数）



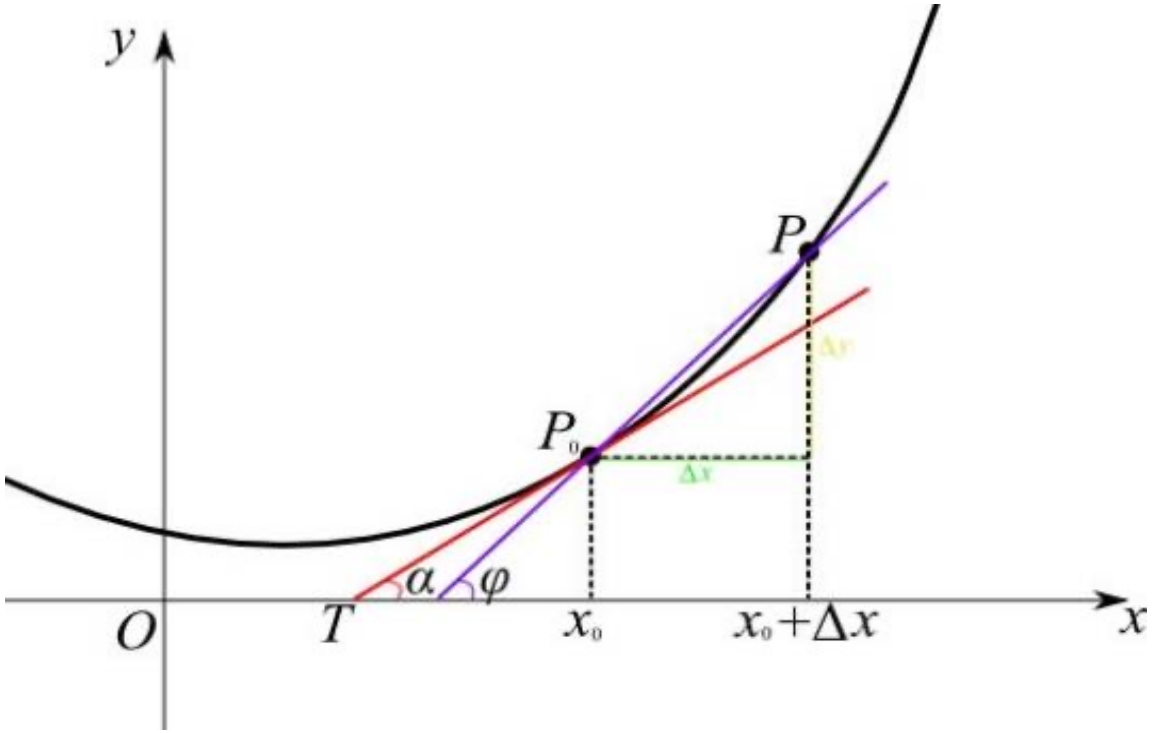

梯度下降法原理：

- 梯度（Gradient）：损失函数相对于参数的导数，表示损失函数在当前点的变化率。
- 梯度下降的核心思想：朝着损失函数最陡下降的方向（即梯度的反方向）更新参数。

# 梯度下降｜偏导数与梯度

导数：衡量函数在某一点变化率的概念。对于一个单变量函数 $f(x)$，其导数$f$描述的是函数$f(x)$在某一点的瞬时变化率，即曲线在这一点的斜率。导数$f'(x)$的定义

$$f'(x) = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$




常用函数导数公式

| 函数 | 导数 |
|---|---|
| $f(x) = c$ | $f'(x) = 0$ |
| $f(x) = x^n$ | $f'(x) = nx^{n-1}$ |
| $f(x) = e^x$ | $f'(x) = e^x$ |
| $f(x) = \ln(x)$ | $f'(x) = \frac{1}{x}$ |
| $f(x) = \sin(x)$ | $f'(x) = \cos(x)$ |
| $f(x) = \cos(x)$ | $f'(x) = -\sin(x)$ |
| $f(x) = \tan(x)$ | $f'(x) = \sec^2(x)$ |
| $f(x) = \arcsin(x)$ | $f'(x) = \frac{1}{\sqrt{1-x^2}}$ |
| $f(x) = \arccos(x)$ | $f'(x) = -\frac{1}{\sqrt{1-x^2}}$ |
| $f(x) = \arctan(x)$ | $f'(x) = \frac{1}{1+x^2}$ |
| $f(x) = \sinh(x)$ | $f'(x) = \cosh(x)$ |
| $f(x) = \cosh(x)$ | $f'(x) = \sinh(x)$ |
| $f(x) = \log_a(x)$ | $f'(x) = \frac{1}{x \ln(a)}$ |

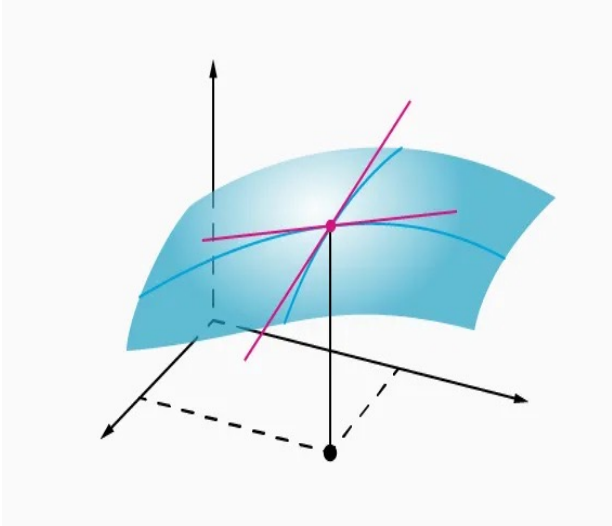
# 梯度下降｜偏导数与梯度

- 偏导数：偏导数是多变量函数中，衡量一个变量变化时，其他变量不变的情况下，函数值的变化率。假设有一个多变量函数 $f(x_1, x_2, \ldots, x_n)$，其中有多个自变量，偏导数是在其他自变量保持不变时，函数相对于其中一个自变量的变化率。

  几何意义：偏导数表示的是在某一点，函数沿某个特定方向（某个变量变化的方向）的变化速率。

  定义：如果 $f(x_1, x_2, \ldots, x_n)$ 是一个多变量函数，则其对 $x_i$ 的偏导数表示为：

$$\frac{\partial f}{\partial x_i} = \lim_{\Delta x_i \to 0} \frac{f(x_1, x_2, \ldots, x_i + \Delta x_i, \ldots, x_n) - f(x_1, x_2, \ldots, x_n)}{\Delta x_i}$$



- 梯度：是一个向量，包含了函数相对于所有自变量的偏导数。梯度描述了函数在多维空间中的最大上升方向。在机器学习和优化中，梯度常用于寻找最优解，例如在梯度下降算法中，梯度用来指导参数调整的方向。

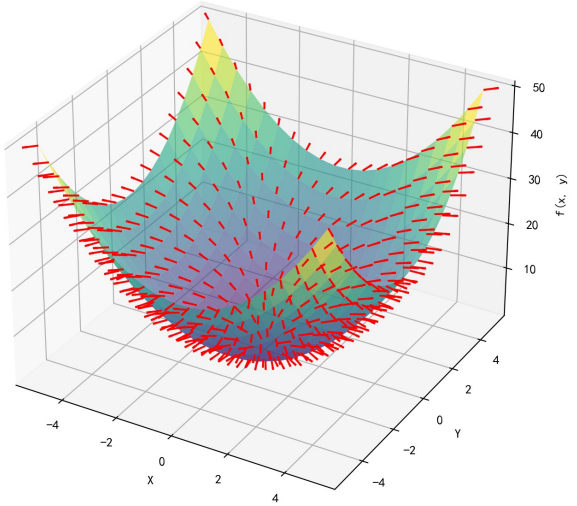
  几何意义：梯度的方向是函数增速最快的方向，而梯度的大小表示增速的快慢。

  定义：对于多变量函数 $f(x_1, x_2, \ldots, x_n)$，其梯度是一个向量，由函数对每个自变量的偏导数组成：

$$\nabla h_t heta(x_1, x_2, \ldots, x_n) = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right)$$

梯度向量指出了在某一点，函数值变化最快的方向。

$\nabla f(x, y)$



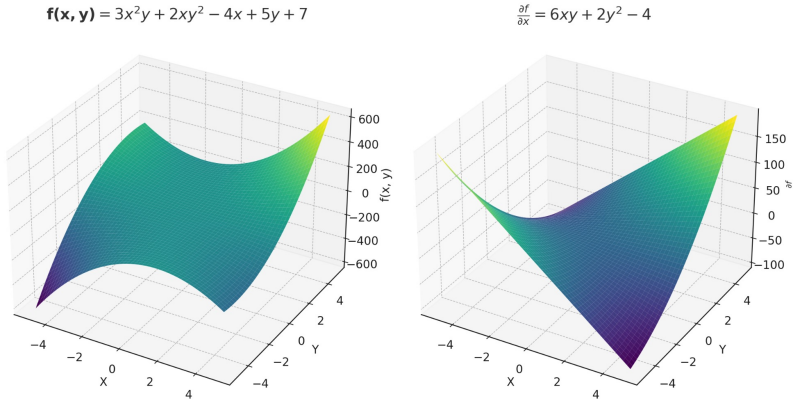
# 梯度下降｜偏导数与梯度

给定一个多变量的多项式函数：

$$f(x, y) = 3x^2y + 2xy^2 - 4x + 5y + 7$$

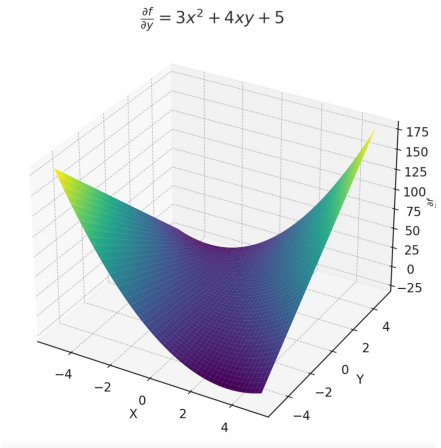
对 x 的偏导数：

$$\begin{aligned}\frac{\partial f}{\partial x} &= \frac{\partial}{\partial x}(3x^2y + 2xy^2 - 4x + 5y + 7) \\ &= 6xy + 2y^2 - 4\end{aligned}$$




对 y 的偏导数：

$$\begin{aligned}\frac{\partial f}{\partial y} &= \frac{\partial}{\partial y}(3x^2y + 2xy^2 - 4x + 5y + 7) \\ &= 3x^2 + 4xy + 5\end{aligned}$$



# 常见的梯度下降方法

- **梯度下降（Gradient Descent，GD）**

原理：每次迭代时，计算整个训练集的梯度，并基于该梯度更新模型参数。

缺点：计算资源消耗大，尤其在数据量很大的时候，计算梯度时需要遍历整个数据集，效率较低。

- **随机梯度下降（Stochastic Gradient Descent，SGD）**

原理：每次迭代时，随机选择一个样本计算梯度，并用这个梯度更新模型参数。

缺点：更新过程比较噪声大，容易产生震荡，导致收敛不稳定。

- **小批量梯度下降（Mini-batch Gradient Descent）**

原理：每次迭代时，随机选择一个小批量的样本（如32个或64个），计算这些样本的平均梯度并更新参数。

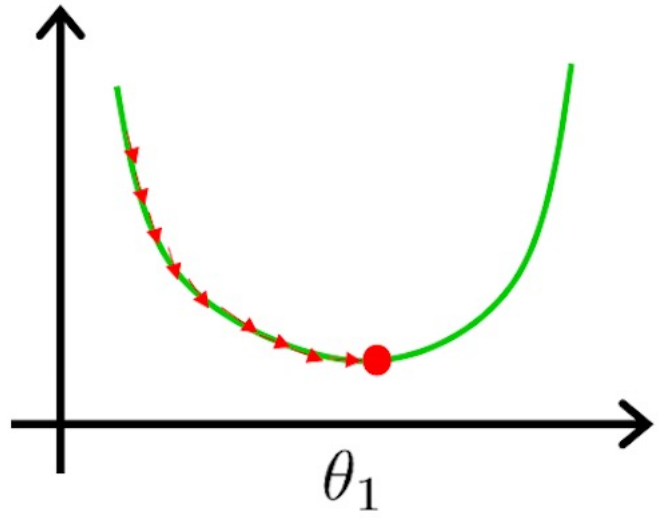
优点：结合了批量梯度下降和随机梯度下降的优点，既能加速计算，又能在一定程度上避免噪声问题。通常在深度学习中广泛使用。

缺点：选择合适的批量大小（batch size）需要调优，过大或过小都会影响收敛速度和效果。

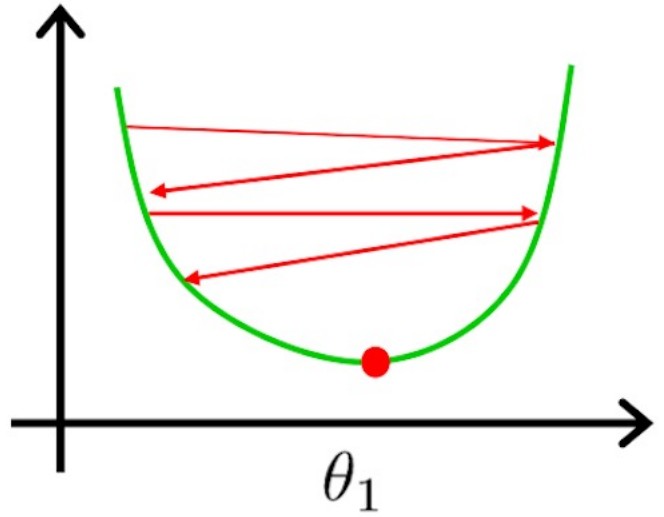
# 梯度下降｜学习率

$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

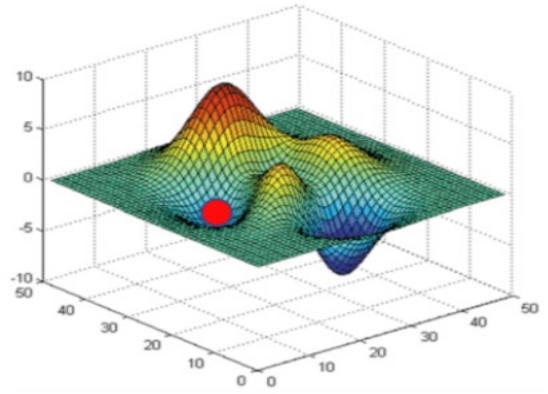
如果学习率$\alpha$过小，梯度下降的过程较慢




如果学习率$\alpha$过大，模型会在全局最小点周围震荡，从而无法收敛

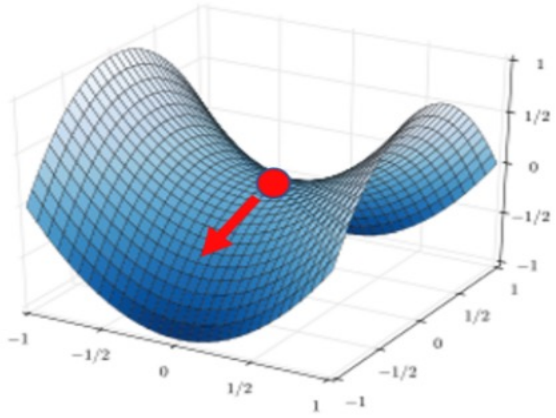


# 梯度下降 | 可能出现的问题



局部最小值

**局部最小值**：梯度下降法可能会停留在损失函数的局部最小值上，而不是找到全局最小值。为了解决这个问题，可以通过使用不同的初始参数来增加找到全局最小值的概率。



鞍点

**鞍点**：梯度下降可能会遇到鞍点，这意味着梯度为但不是最小值。现代优化方法（如Adam）能有效地避免这一问题。

# 梯度下降的几个优化算法

- **动量梯度下降算法（Momentum optimization）**

原理：在更新参数时，考虑之前梯度更新的累积效果，即在当前梯度的基础上加入之前梯度的加权和，更新时不仅依赖当前的梯度，还受过去梯度的影响。

- **AdaGrad（Adaptive Gradient Algorithm）**

原理：根据每个参数的历史梯度调整学习率，频繁更新的参数使用较小的学习率，更新较少的参数使用较大的学习率。

- **RMSprop（Root Mean Square Propagation）**

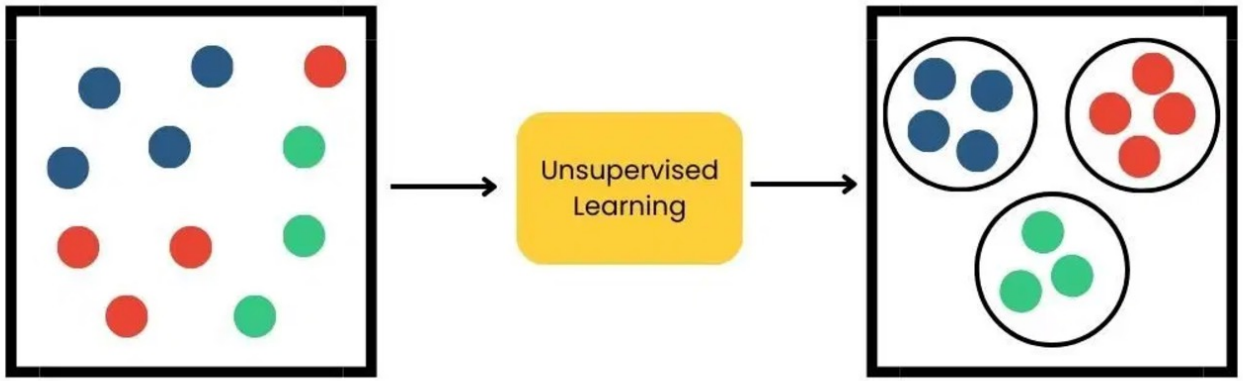
原理：对梯度平方进行加权平均，避免了AdaGrad早期学习率下降过快的问题。使用指数加权移动平均来更新每个参数的学习率。

- **Adam（Adaptive Moment Estimation）**

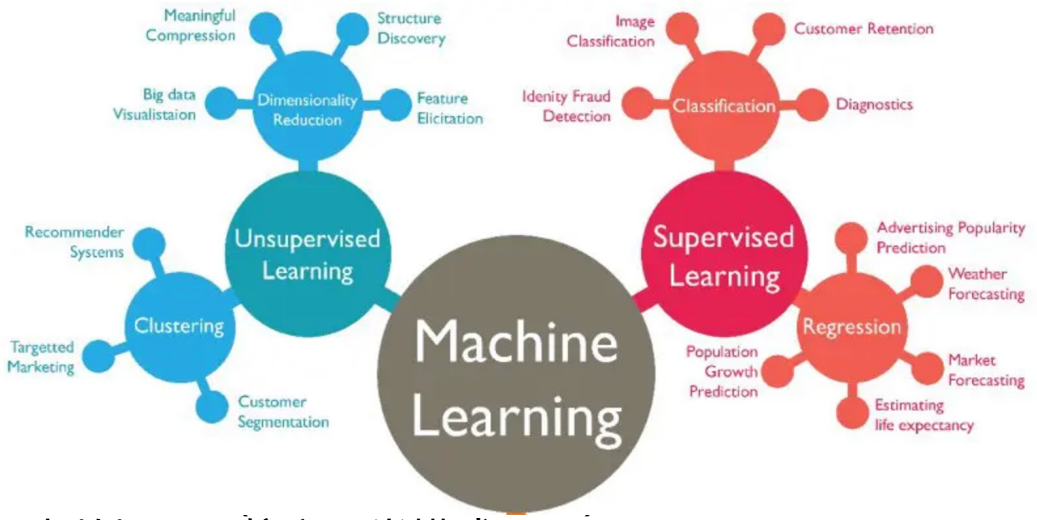
原理：结合了动量法和RMSprop的思想，既考虑梯度的动量，又考虑梯度的平方加权平均。通过一阶矩和二阶矩的估计来动态调整每个参数的学习率。

# 机器学习——无监督学习

- **无监督学习（Unsupervised Learning）**是一种在没有标签的训练数据下进行学习的机器学习方法。目标是从数据中自动发现隐藏的模式或结构。与监督学习相比，无监督学习没有标签数据，仅依赖于数据本身的内在结构。




- 无监督学习的基本任务包括：聚类：将数据分组，类似的对象被放在同一个组里；降维：减少数据的特征维度，保留数据的主要信息；异常检测：识别与大多数数据不同的异常点；关联规则学习：发现数据中的关联规则，例如在购物篮分析中找到商品之间的关联。




本数据是否偏离正常模式。

# 机器学习——无监督学习

## 常见无监督学习算法

**1. 聚类算法**：

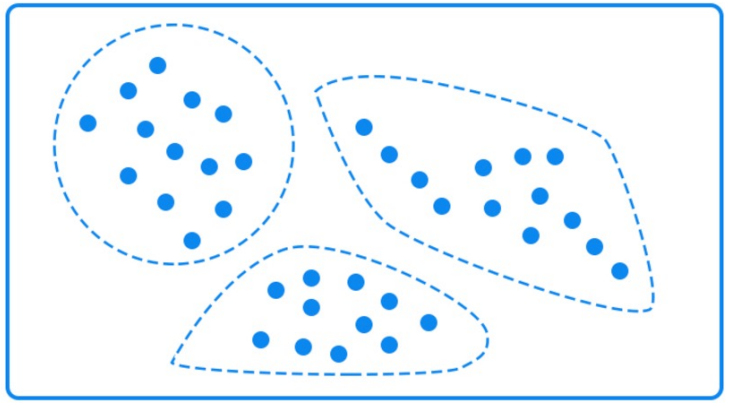
- **K-Means**：通过将数据分成K个簇，每个簇的中心点最小化簇内的距离。
- **层次聚类**：构建一个树状结构，表示数据之间的层次关系。
- **DBSCAN**：基于密度的聚类方法，可以发现任意形状的簇，并能自动识别噪声点。

**2. 降维算法**：

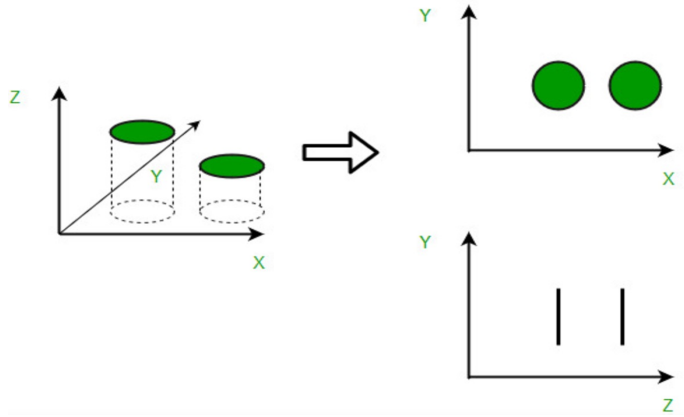
- **PCA（主成分分析）**：通过线性变换减少数据的维度，保留数据中最大的方差。
- **t-SNE**：非线性降维方法，用于高维数据的可视化。

**3. 异常检测**：

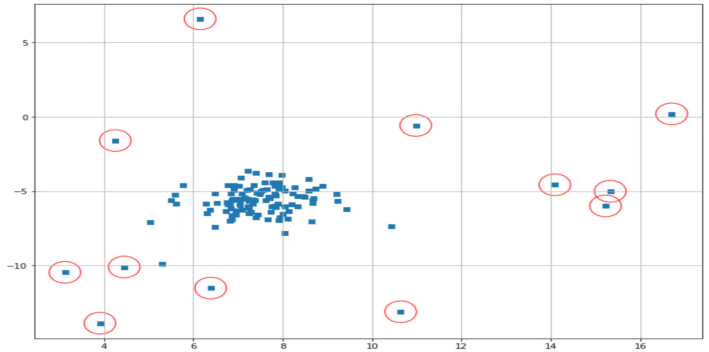
- **孤立森林（Isolation Forest）**：基于树结构检测数据中的异常点。
- **单类SVM（One-Class SVM）**：通过边界框架检测样本数据是否偏离正常模式。



聚类




降维



异常检测

# 无监督学习：K-means聚类

- **k-means算法**的目标是将$n$个$d$维数据$\{x_i, i = 1, \ldots, n\}$划分为$K$个聚簇，使得簇内方差最小化。由于原始数据可能的聚类结果数量巨大，要求一个特定的聚类算法总是能达到最优是不切实际的。所以，k-means算法找到的是一个“局部”最优，即没有任何其他的聚类结果，能够让簇内的方差更小，但不能保证找到全局最优[Hartigan 1979]。k-means同时也是一个易受初始值影响的迭代算法，可以用不同的初始值重复几次，以达到上述的“局部”最优，常用的初始化方法包括Forgy和Random Partition[Hamerly 2002]。

**算法 4.1** $k$-means 聚类

**输入：** $n$ 个 $d$ 维数据 $\{x_i,\ i=1,\ \cdots,\ n\}$，聚簇数目 $K$

**输出：** 每个数据所属聚簇标签

**算法步骤：**

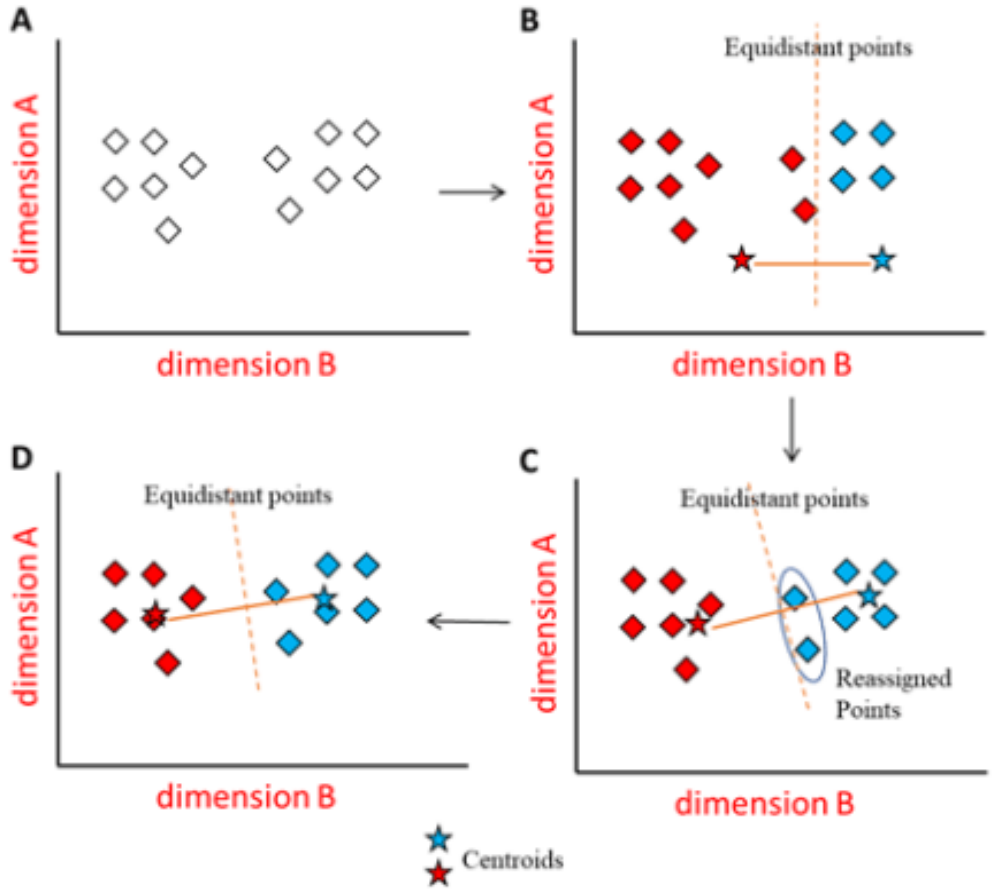
(1) 初始化聚类质心。

(2) 根据预定的相似度 / 距离函数（通常为欧氏距离）对数据进行聚类。

(3) 更新聚类质心。

(4) 重复 (2) 和 (3)，直到收敛。

# 无监督学习：K-means聚类




1.选定质心：
选择 K 个初始簇中心（也称为质心，centroids）

2. 分配簇
对于数据集中的每一个数据点，计算它与 K 个簇中心的距离，通常使用欧氏距离。将每个数据点分配给距离它最近的簇中心。

3. 更新簇中心
在所有数据点都被分配到一个簇中后，计算每个簇的新簇中心。新簇中心是簇中所有数据点的均值。

4.迭代
重复步骤 2 和 3，直到满足停止条件：
簇中心不再变化，或者变化范围小于给定值；
达到最大迭代次数。

# 无监督学习：主成分分析

- 主成分分析（principal component analysis）是一种特征降维方法，在消除数据噪声、冗余等方面具有广泛应用

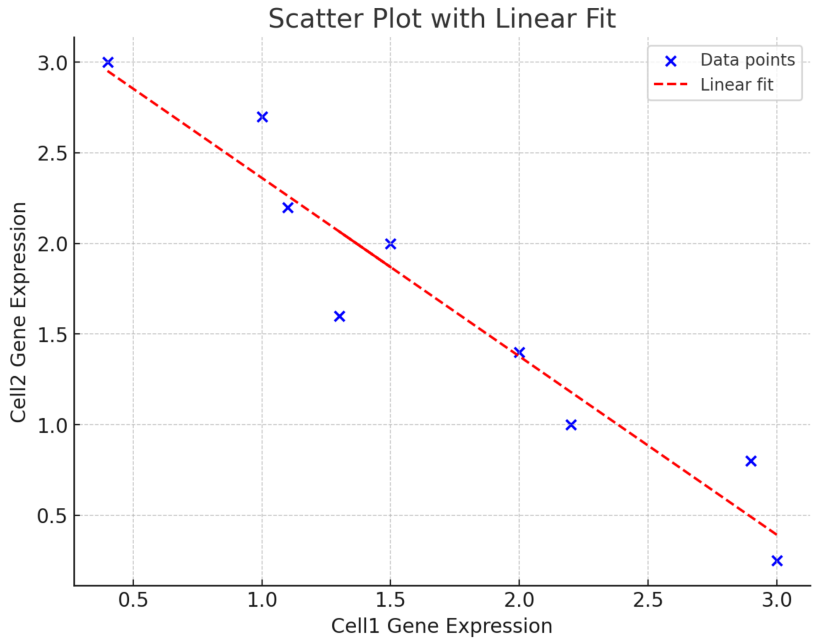
- 顾名思义，主成分分析即通过分析找到数据特征的主要成分，使用这些主要成分来代替原始数据。这样一方面可以加深对数据本身的理解（认识到数据的主要成分）；另一方面，简化后的数据（主要成分）在用于下游的其他任务时，有着噪声少、易于处理计算的特点。

- 主成分分析要求“降维后的结果要保持原始数据的原有结构”，例如对于图像数据，要求保持视觉对象区域构成的空间分布；对于文本数据，要求保持单词之间的（共现）相似或不相似的特性。更准确地说，主成分分析要求最大限度保持原始高维数据的总体方差结构。

# 无监督学习：主成分分析

| | Cell1 | Cell2 | Cell3 | Cell4 | … |
|---|---|---|---|---|---|
| Gene1 | 3 | 0.25 | 2.8 | 0.1 | … |
| Gene2 | 2.9 | 0.8 | 2.2 | 1.8 | … |
| Gene3 | 2.2 | 1 | 1.5 | 3.2 | … |
| Gene4 | 2 | 1.4 | 2 | 0.3 | … |
| Gene5 | 1.3 | 1.6 | 1.6 | 0 | … |
| Gene6 | 1.5 | 2 | 2.1 | 3 | … |
| Gene7 | 1.1 | 2.2 | 1.2 | 2.8 | … |
| Gene8 | 1 | 2.7 | 0.9 | 0.3 | … |
| Gene9 | 0.4 | 3 | 0.6 | 0.1 | … |

| | Cell1 | Cell2 |
|---|---|---|
| Gene1 | 3 | 0.25 |
| Gene2 | 2.9 | 0.8 |
| Gene3 | 2.2 | 1 |
| Gene4 | 2 | 1.4 |
| Gene5 | 1.3 | 1.6 |
| Gene6 | 1.5 | 2 |
| Gene7 | 1.1 | 2.2 |
| Gene8 | 1 | 2.7 |
| Gene9 | 0.4 | 3 |

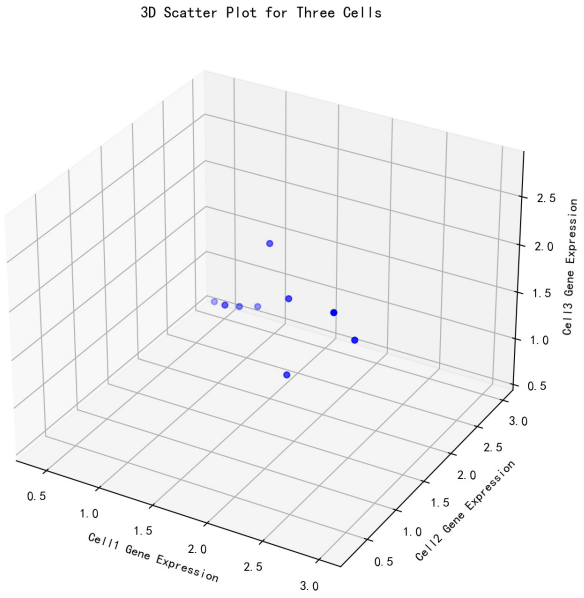



给定一组基因数据，每列展示的是每个基因被转录到每个细胞的程度

假设只有两个细胞，绘图分析它们的转录情况

# 无监督学习：主成分分析

| | Cell1 | Cell2 | Cell3 |
|---|---|---|---|
| Gene1 | 3 | 0.25 | 2.8 |
| Gene2 | 2.9 | 0.8 | 2.2 |
| Gene3 | 2.2 | 1 | 1.5 |
| Gene4 | 2 | 1.4 | 2 |
| Gene5 | 1.3 | 1.6 | 1.6 |
| Gene6 | 1.5 | 2 | 2.1 |
| Gene7 | 1.1 | 2.2 | 1.2 |
| Gene8 | 1 | 2.7 | 0.9 |
| Gene9 | 0.4 | 3 | 0.6 |



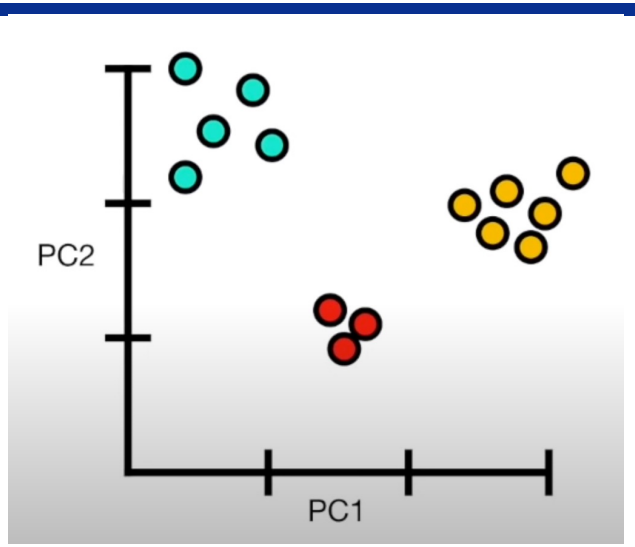

| | Cell1 | Cell2 | Cell3 | Cell4 |
|---|---|---|---|---|
| Gene1 | 3 | 0.25 | 2.8 | 0.1 |
| Gene2 | 2.9 | 0.8 | 2.2 | 1.8 |
| Gene3 | 2.2 | 1 | 1.5 | 3.2 |
| Gene4 | 2 | 1.4 | 2 | 0.3 |
| Gene5 | 1.3 | 1.6 | 1.6 | 0 |
| Gene6 | 1.5 | 2 | 2.1 | 3 |
| Gene7 | 1.1 | 2.2 | 1.2 | 2.8 |
| Gene8 | 1 | 2.7 | 0.9 | 0.3 |
| Gene9 | 0.4 | 3 | 0.6 | 0.1 |

假设有三个细胞，绘图分析他们的转录情况

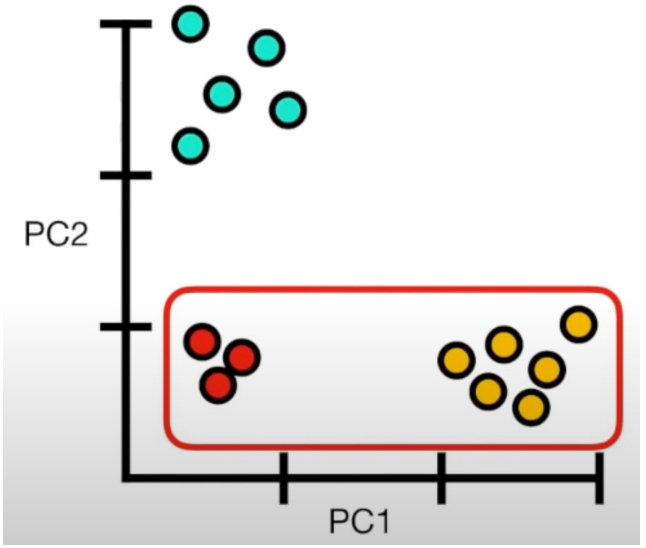
如果有四个以上的细胞，如何分析它们的转录情况？

# 无监督学习：主成分分析

| | Cell1 | Cell2 | Cell3 | Cell4 |
|---|---|---|---|---|
| Gene1 | 3 | 0.25 | 2.8 | 0.1 |
| Gene2 | 2.9 | 0.8 | 2.2 | 1.8 |
| Gene3 | 2.2 | 1 | 1.5 | 3.2 |
| Gene4 | 2 | 1.4 | 2 | 0.3 |
| Gene5 | 1.3 | 1.6 | 1.6 | 0 |
| Gene6 | 1.5 | 2 | 2.1 | 3 |
| Gene7 | 1.1 | 2.2 | 1.2 | 2.8 |
| Gene8 | 1 | 2.7 | 0.9 | 0.3 |
| Gene9 | 0.4 | 3 | 0.6 | 0.1 |




主成分分析图a




主成分分析图b

坐标轴按照显著性顺序排列

PC1: **第一主成分**

PC2: **第二主成分**

两个群体，在横纵坐标相同距离的情况下，沿第一主成分轴线的差异比沿第二主成分轴线的差异更明显

# 主成分分析方式：降维

主成分分析的思想是将$d$维特征数据映射到$l$维空间（一般$d \gg l$），去除原始数据之间的冗余性。

将原始数据向这些数据方差最大的方向进行投影。一旦发现了方差最大的投影方向，则继续寻找保持方差第二的方向且进行投影。其目标是使得数据每一维的方差都尽可能大。

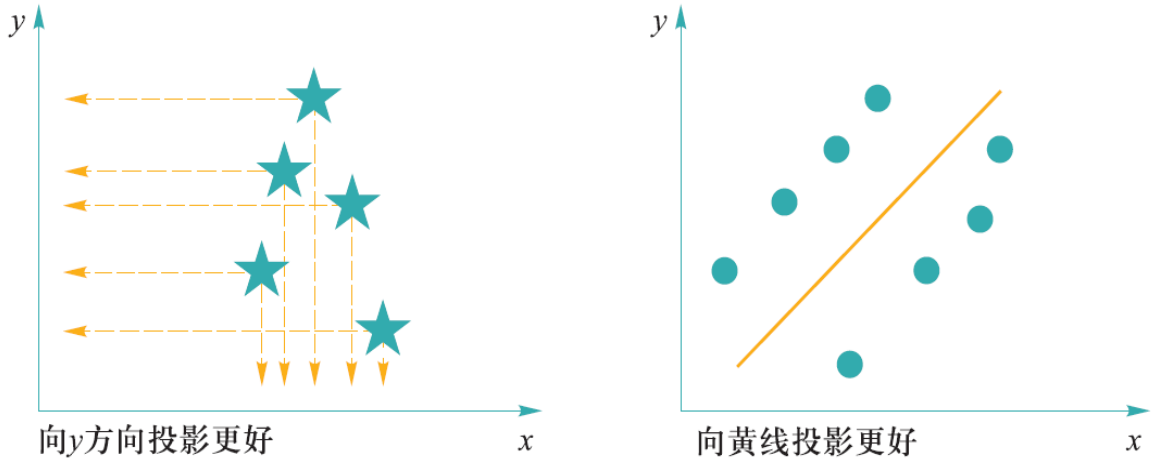



图 4.5　PCA 降维示意

**降维需要尽可能将数据向方差最大的方向进行投影，使得数据所蕴含的信息丢失得尽可能少**。如图4.5左图所示，向$y$方向投影（使得二维数据映射为一维）就比向$x$方向投影结果在降维这个意义上而言要好；图4.5右图则是黄线方向投影要好。这样的投影结果更好的保留了未降维前数据的离散程度。

# 小结

- 机器学习：利用统计方法通过从数据中学习规律和模式
- 所有模型都是错的，但有些确实有用
- 设计损失函数通过训练集的表现来优化模型参数
- 可以通过梯度下降来优化损失函数
- PCA的计算可以等价地通过数据矩阵的SVD来完成，是一种广为流行的特征降维方法

# 目录

# 模型泛化

模型训练


模型泛化

# 模型泛化能力 | 经验风险与期望风险

## 经验风险

映射函数$f$在训练集上所产生损失一般被称为**经验风险**$\Re_{emp}$ (empirical risk)。经验风险越小说明模型对训练集数据拟合程度越好。经验风险被定义为：

$$\frac{1}{n}\sum_{i=1}^{n} Loss(y_i, f(x_i))$$

$\neq$

## 期望风险

如果知道某一任务包含的所有数据，则可以从所有数据中计算模型产生的损失，这一误差损失被称为**期望风险**$\Re$（expected risk），即真实风险或真实误差。记该任务中所有数据的联合分布为$\mathrm{P(x,y)}$，期望风险被定义为：

$$\int_{\mathrm{x\times y}} \mathrm{Loss\big(y,f(x)\big)P(x,y)dxdy}$$

# 经验风险最小化

当然，由于无法事先就得到任何任务所对应的所有数据分布（如无法采取世界中所有人脸图像来笃信完成人脸识别），使得计算期望风险这一目标可望不可及。因此，机器学习中模型优化目标一般为**经验风险最小化（empirical risk minimization）**，虽然机器学习的目标是追求期望风险最小化，即不断提升模型泛化能力。

# 经验风险最小化

期望风险$\mathfrak{R}$与经验风险$\mathfrak{R}_{emp}$之间存在如下关系：

$$\mathfrak{R} \leq \mathfrak{R}_{emp} + err$$

期望风险　经验风险

其中，$err$ 取值与机器学习模型的复杂程度和训练集样本数目有关。在模型训练过程中，如果使用同一批训练数据反复训练，模型会变得越复杂，虽然经验风险$\mathfrak{R}_{emp}$会降低，但是$err$取值会越大，导致期望风险$\mathfrak{R}$增加，这一现象被称为**过拟合**（overfitting）。

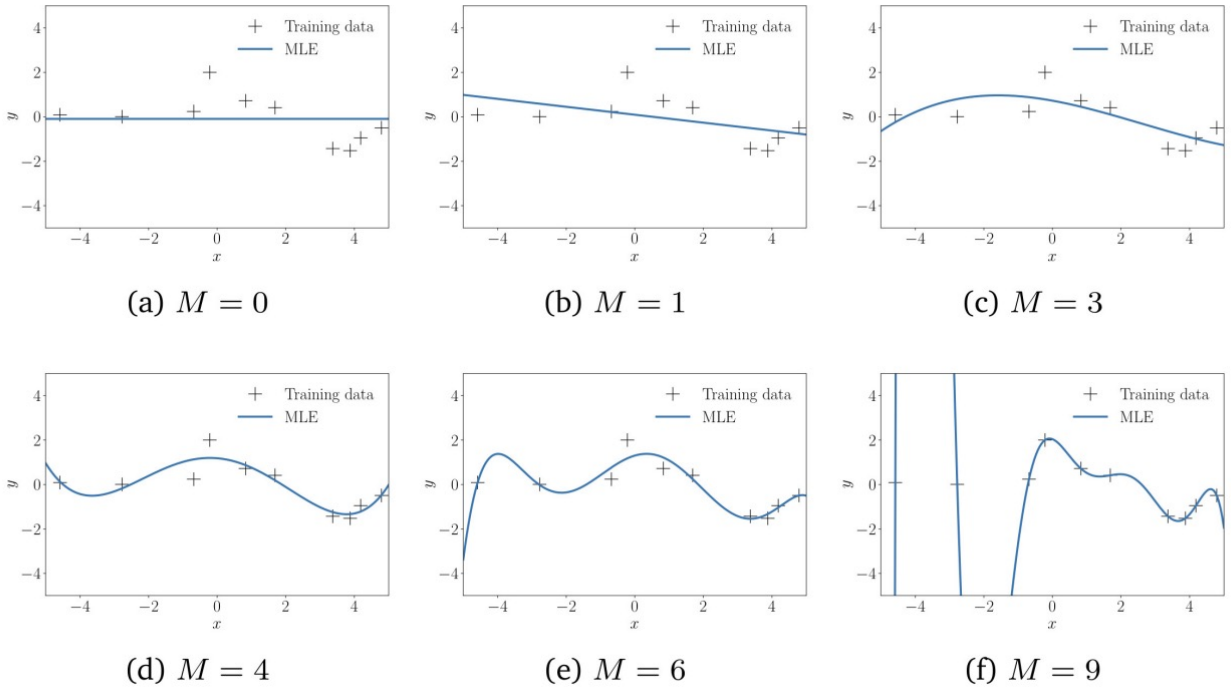
# 模型泛化能力与经验风险和期望风险之间关系

表 4.2　模型泛化能力与经验风险、期望风险之间的关系

| 经验风险 | 期望风险 | 模型泛化能力 |
| --- | --- | --- |
| 经验风险小<br>（训练集上表现好） | 期望风险小<br>（所有数据上表现好） | 泛化能力强 |
| 经验风险小<br>（训练集上表现好） | 期望风险大<br>（所有数据上表现不好） | 过学习<br>（模型过于复杂） |
| 经验风险大<br>（训练集上表现不好） | 期望风险大<br>（所有数据上表现不好） | 欠学习 |
| 经验风险大<br>（训练集上表现不好） | 期望风险小<br>（所有数据上表现好） | “神仙算法”或“黄粱美梦” |

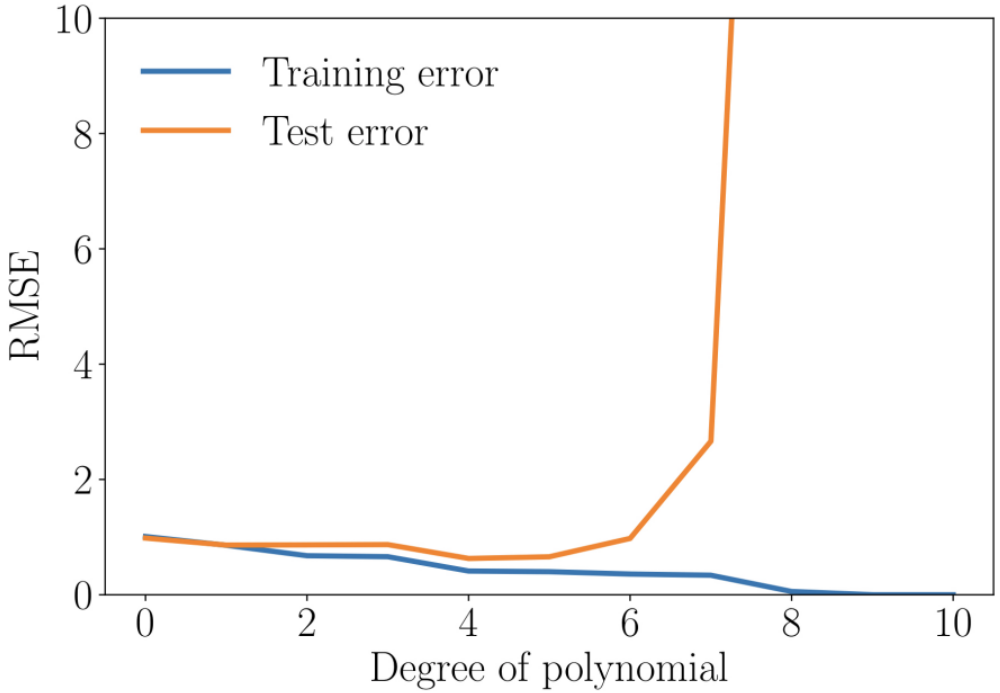
# 模型泛化能力与经验风险和期望风险之间关系

欠学习




(a) $M = 0$ (b) $M = 1$ (c) $M = 3$

(d) $M = 4$ (e) $M = 6$ (f) $M = 9$




过学习

# 模型泛化能力与经验风险和期望风险之间关系

如前“没有免费午餐定理”所指出，在模型优化中引入恰当先验约束可提升模型性能。为了防止过学习，结构风险最小化(structural risk minimization)引入正则化 (regularizer)或惩罚项(penalty term) 来降低模型模型复杂度，既最小化经验风险、又力求降低模型复杂度，在两者之间寻找平衡：

$$\frac{1}{n}\sum_{i=1}^{n} Loss(y_i, f(x_i)) + \lambda J(f)$$

其中$J(f)$是正则化因子或惩罚项因子，$\lambda$是用来调整惩罚强度的系数。哲学领域的奥卡姆剃刀定律（Occam's Razor, Ockham's Razor）阐明了“如无必要，勿增实体”的意义，即“简单有效原理”。老子《道德经》曾写道，“万物之始，大道至简，衍化至繁”。在模型中加入约束（如约束模型系数稀疏等），使得从数据到模型的建模过程中，能够“化繁为简、大巧不工”。

# 模型度量方法

机器学习模型需要若干性能度量指标来判断其性能优劣。下面以二分类问题（正类、负类）为例，介绍几种主要度量方法。$n$为训练样例的总数，正例总数和负例总数分别是$P$(positive)和$N$(negative)。机器模型预测类别可分为如下四类：真正例（True Positive, TP）、假正例（False Positive, FP）、真反例（True Negative, TN）与假反例（False Negative, FN），令$TP$、$FP$、$TN$、$FN$分别表示其对应的样例数。

**准确率(accuracy)：$ACC = \frac{TP+TN}{P+N}$。很显然，如果正负样本比例不平衡，$ACC$ 不是一个度量模型好的方法。比如，某一种恶性疾病很罕见（如1万个疑似患者中仅有1人罹患该疾病），机器学习模型可将所有患者均识别为负类，从而保证$ACC$取值极高，但这就忽略了这一模型应该关注的问题。**

**错误率（error rate）：$errorRate = \frac{FP+FN}{P+N}$，显然有$errorRate = 1 - ACC$。**

**精确率（precision）：$precision = \frac{TP}{TP+FP}$，也叫查准率，表示被模型预测为正例的样本中实际为正例的比例。**

**召回率（Recall）：$recall = \frac{TP}{TP+FN}$，也叫查全率，表示所有正例样本中被模型预测为正例的比例。**

# 模型度量方法

在实际应用中，精确率和召回率之间是相互矛盾的，比如可以将所有样本分类为正例使得召回率为100%而精确率极低。因此为了综合考虑精确率和召回率，可采用F1-score这一综合分类率：

$$\text{综合分类率（F1-score）}:\frac{2}{\frac{1}{Precision}+\frac{1}{Recall}}=2\times\frac{Precision\times Recall}{Precision+Recall}$$

F1-score是精确率和召回率的调和平均数，在尽可能提高精确率和召回率同时，也希望两者之间的差异尽可能小。

# 参数优化：频率学派与贝叶斯学派

## 频率学派

在频率学派中，频率是概率的经验基础，概率表示的是事件发生频率的极限值。**当重复试验的次数趋近于无穷大时，事件发生的频率会收敛到真实概率**，即“频率依概率收敛于概率”。从频率学派角度而言，对模型参数优化学习的结果就是得到使观测数据发生概率最大的模型参数，又称为最大似然估计（maximum likelihood estimation，MLE）。这里的最大似然可理解为通过调整模型参数使得模型能够最大化样本情况出现的概率。

## 贝叶斯学派

在贝叶斯学派中，**事件发生的频率既与当前观测数据有关，又与对该事件已获得的历史先验知识有关**。从贝叶斯学派角度而言，对模型参数优化学习的结果就是似然概率（模型参数产生数据的概率）与先验概率（没有任何实验数据时对模型参数的经验判断）乘积最大，又称为最大后验估计（maximum a posteriori estimation, MAP）。这里的最大后验估计可理解为最大化在给定数据样本的情况下模型参数的后验概率。

- MLE

**The negative log-likelihood**

$$-\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) = -\log \prod_{i=1}^{N} p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}) = -\sum_{i=1}^{N} \log p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}).$$

⋆ **Note:** the independence assumption on the training set applies here.

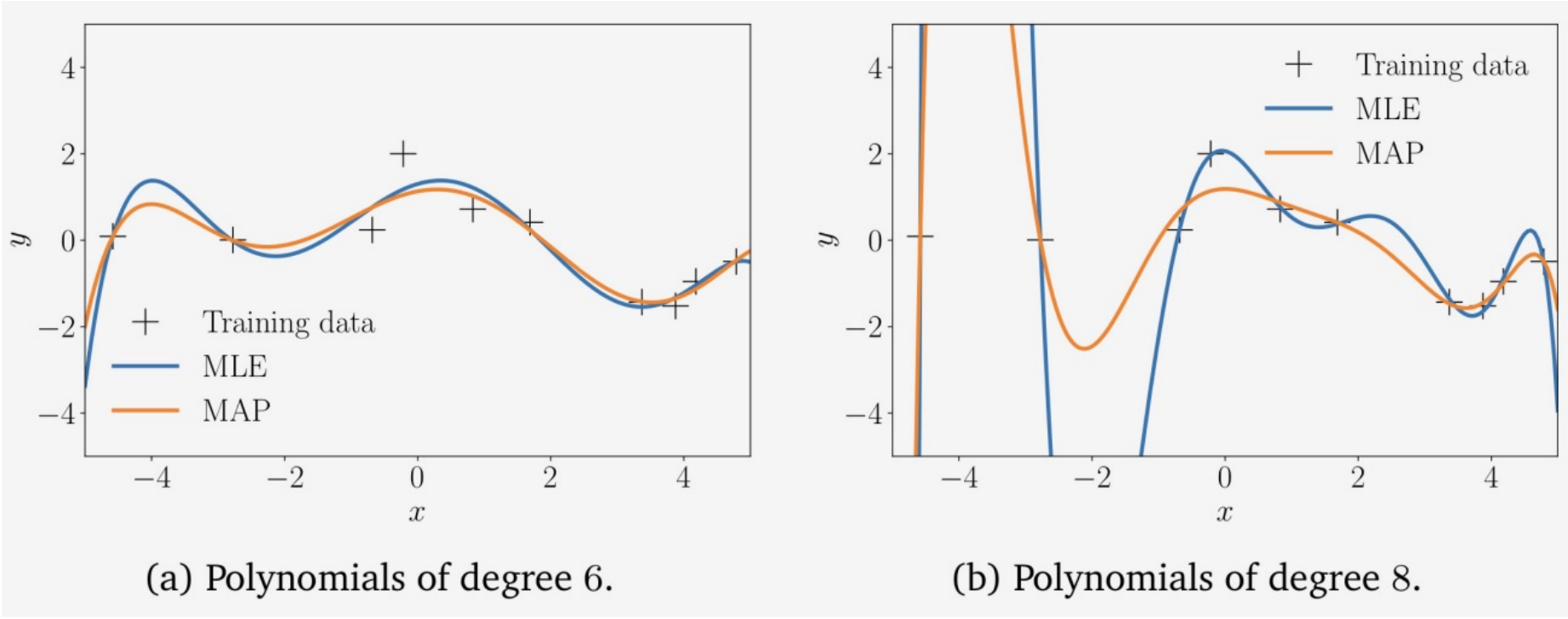
$$\log p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}) = -\frac{1}{2\sigma^2}(y_i - \mathbf{x}^\top \boldsymbol{\theta})^2 + \text{constant}_{\text{indepent of } \boldsymbol{\theta}}.$$

- MAP

The log-transformation of the posterior:

$$\log p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) = \log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) + \text{constant}$$

# 参数优化：频率学派与贝叶斯学派



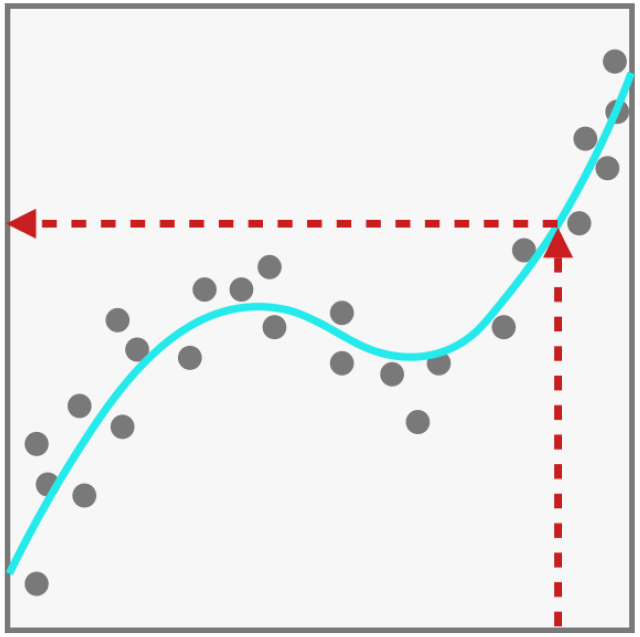

(a) Polynomials of degree 6.

(b) Polynomials of degree 8.

# 监督学习两类问题|回归与分类

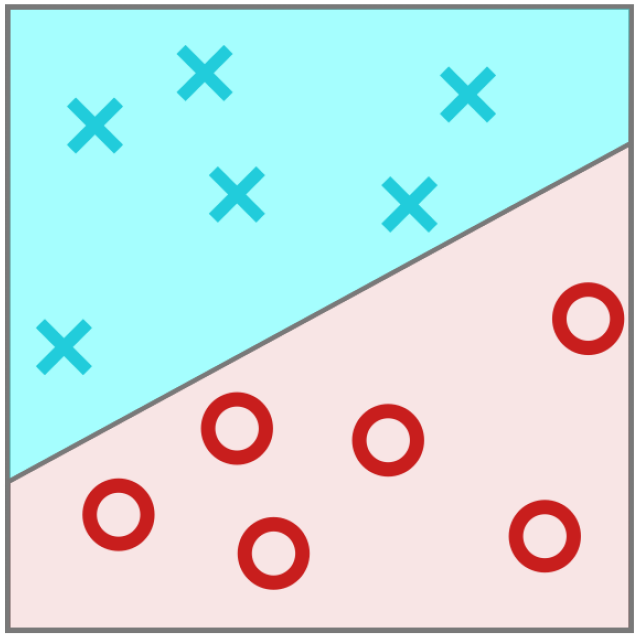
## 回归问题

通过已知的输入数据（特征）预测一个连续的数值输出。回归问题的目标是建立一个模型，使其能够根据输入的特征预测一个数值型的目标变量



## 分类问题

将输入数据分配到预定类别。分类问题的目标是通过学习训练数据中的特征，建立一个分类器，从而能够将新样本正确地分到一个或多个类别中

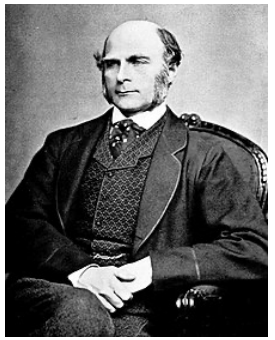


# 回归分析

在现实生活中，往往需要分析若干变量之间的关系，如碳排放量与气候变暖之间的关系，某一商品广告投入量与该商品销售量之间的关系等，这种分析不同变量之间存在关系的研究叫作回归分析，刻画不同变量之间关系的模型称为回归模型

$$y = 33.73(\text{英寸}) + 0.516x$$

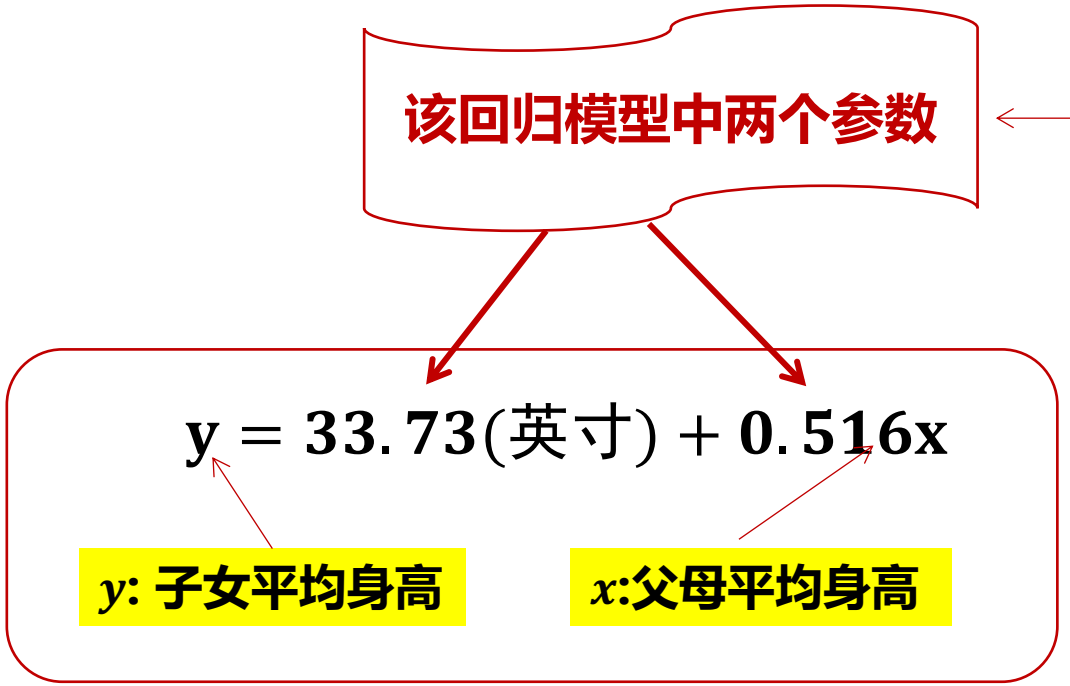
**$y$: 子女平均身高**　　**$x$:父母平均身高**



英国著名生物学家兼统计学家高尔顿
Sir Francis Galton
(1822-1911)

- **父母平均身高每增加一个单位，其成年子女平均身高只增加0.516个单位，它反映了这种“衰退(regression)”效应（“回归”到正常人平均身高)**
- **虽然$x$和$y$之间并不总是具有“衰退”（回归）关系，但是“线性回归”这一名称就保留了下来了**

# 回归分析




- 给出任意一对父母平均身高，则可根据上述方程，计算得到其子女平均身高
- 从父母平均身高来预测其子女平均身高
- 如何求取上述线性方程（预测方程）的参数?

# 回归分析

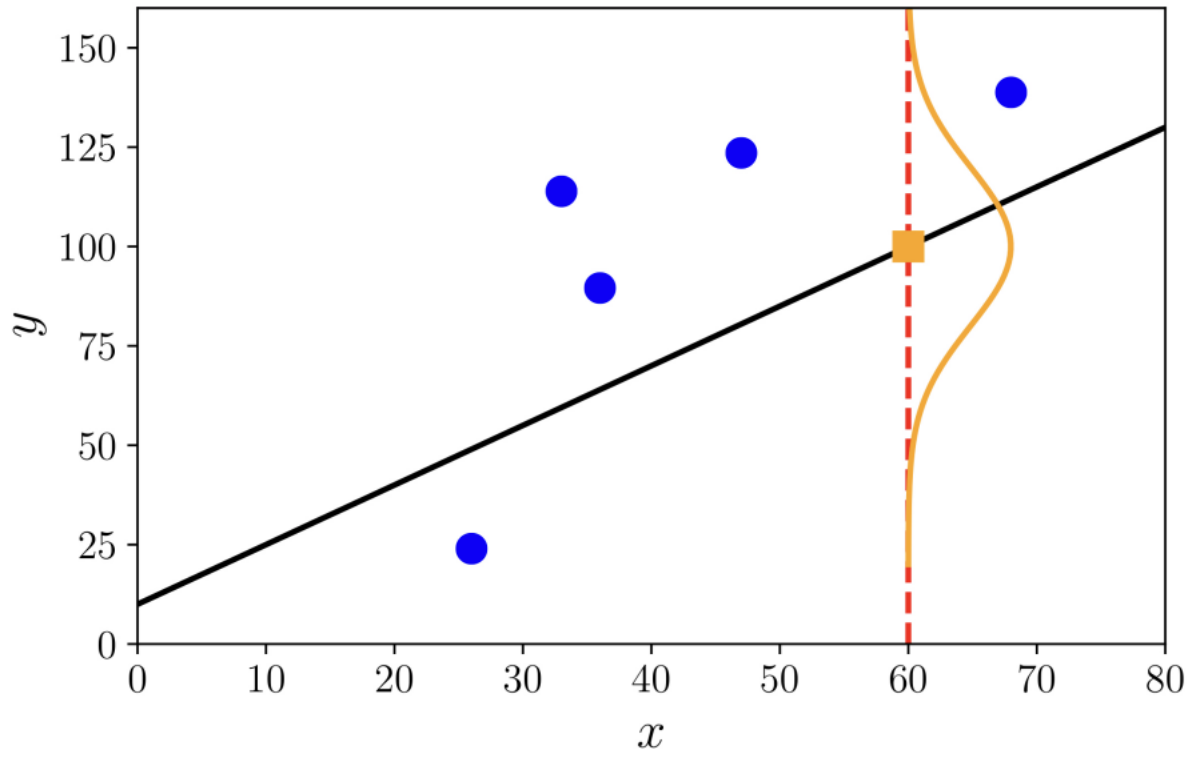
- Given a training set $\mathcal{D} := \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$ for $i = 1, \ldots, N$.
- By the independence of the input, the likelihood factorizes:

$$\begin{aligned} p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) &= p(y_1, \ldots, y_N \mid \mathbf{x}_1, \ldots, \mathbf{x}_N, \boldsymbol{\theta}) \\ &= \prod_{i=1}^{N} p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}) = \prod_{i=1}^{N} \mathcal{N}(y_i \mid \mathbf{x}_i^\top \boldsymbol{\theta}, \sigma^2). \end{aligned}$$

  The likelihood and the factors $p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta})$ are Gaussian due to the noise distribution.

- **Goal:** Find optimal parameters $\boldsymbol{\theta}^* \in \mathbb{R}^D$.
- Then we can make predictions for an arbitrary test input $\mathbf{x}_*$ and get target $y_*$ with $p(y_* \mid \mathbf{x}_*, \boldsymbol{\theta}^*) = \mathcal{N}(y_* \mid \mathbf{x}_*^\top \boldsymbol{\theta}^*, \sigma^2)$.

# 回归分析



# 回归分析

- 对于回归任务的MLE估计本质就是高斯假设下的最小二乘法

**Example (contd.)**

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &= -\sum_{i=1}^{N} \log p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}) = -\sum_{i=1}^{N} \log \mathcal{N}(y_i \mid \mathbf{x}_i^\top \boldsymbol{\theta}, \sigma^2) \\
&= -\sum_{i=1}^{N} \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \mathbf{x}_i^\top \boldsymbol{\theta})^2}{2\sigma^2}\right) \\
&= -\sum_{i=1}^{N} \log \exp\left(-\frac{(y_i - \mathbf{x}_i^\top \boldsymbol{\theta})^2}{2\sigma^2}\right) - \sum_{i=1}^{N} \log \frac{1}{\sqrt{2\pi\sigma^2}} \\
&= \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - \mathbf{x}_i^\top \boldsymbol{\theta})^2 - \sum_{i=1}^{N} \log \frac{1}{\sqrt{2\pi\sigma^2}}.
\end{aligned}
$$

The second term is constant.

$\Longrightarrow$ minimizing $\mathcal{L}(\boldsymbol{\theta})$ $\Rightarrow$ solving the least-squares problem.

# 回归分析：一元线性回归模型

表4.3给出了芒提兹尼欧（Montesinho）地区发生森林火灾的部分历史数据，表中列举了每次发生森林火灾时的气温温度取值$x$和受到火灾影响的森林面积$y$。

表 4.3　芒提兹尼欧地区发生森林火灾的部分数据

| 气温 $x$ | 5.1 | 8.2 | 11.5 | 13.9 | 15.1 | 16.2 | 19.6 | 23.3 |
|---|---|---|---|---|---|---|---|---|
| 火灾影响面积 $y$ | 2.14 | 4.62 | 8.24 | 11.24 | 13.99 | 16.33 | 19.23 | 28.74 |

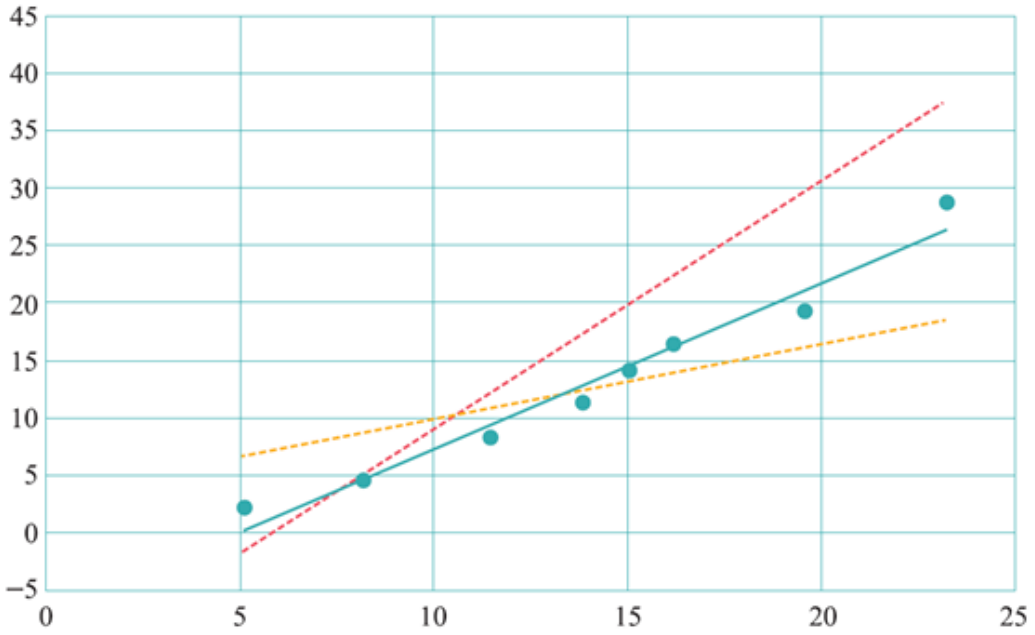



图 4.1　火灾影响的森林面积与气温之间的关系

一元线性回归实际上就是寻找一条用$y = ax + b$表达的直线，使得这条直线尽可能靠近或穿过这8组$(x, y)$数据，即能够以最小误差来拟合这8组$(x, y)$数据。

# 回归分析：一元线性回归模型参数求解

最佳回归模型将使得残差平方和的平均值$\frac{1}{N}\sum(y-\tilde{y})^2$最小，残差即预测值和真实值之间的差值。残差平方和的平均值最小只与参数$a$和$b$有关，最优解即是使得残差最小所对应的$a$和$b$的值。

一般而言，回归模型$y_i = ax_i + b\ (1 \le i \le n)$的参数求解过程为：记在当前参数下第$i$个训练样本$x_i$的预测值为$\hat{y}_i$，

计算$x_i$的标注值（实际值）$y_i$与预测值$\hat{y}_i$之差的平方$(y_i - \hat{y}_i)^2$，

计算训练集中$n$个样本所产生误差总和$L(a,b) = \sum_{i=1}^{n}(y_i - ax_i - b\ )^2$，

使用最小二乘法找到误差总和最小，要使函数具有最小值，

可对$L(a,b)$参数$a$和$b$分别求导，令其导数值为零，再求取参数$a$和$b$的取值。

# 回归分析：一元线性回归模型参数求解

**优化目标：**

$$\min_{a,b} L(a,b) = \sum_{i=1}^{n} (y_i - ax_i - b)^2$$

**损失函数对b求偏导：**

$$\frac{\partial L(a,b)}{\partial b} = \sum_{i=1}^{n} 2(y_i - ax_i - b)(-1) = 0$$

$$\Rightarrow \sum_{i=1}^{n} (y_i - ax_i - b) = 0$$

$$\Rightarrow \sum_{i=1}^{n} (y_i) - a\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} b = 0$$

$$\Rightarrow n\bar{y} - na\bar{x} - nb = 0$$

$$\Rightarrow b = \bar{y} - a\bar{x}$$

**这样就得到了参数b的计算公式。**

**损失函数对a求偏导：**

$$\frac{\partial L(a,b)}{\partial a} = \sum_{i=1}^{n} 2(y_i - ax_i - b)(-x_i) = 0$$

**将**$b = \bar{y} - a\bar{x}$ **(其中**$\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$, $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$**)代入上式**

$$\Rightarrow \sum_{i=1}^{n} (y_i - ax_i - \bar{y} + a\bar{x})(x_i) = 0$$

$$\Rightarrow \sum_{i=1}^{n} (y_i x_i - ax_i x_i - \bar{y}x_i + a\bar{x}x_i) = 0$$

$$\Rightarrow \sum_{i=1}^{n} (y_i x_i - \bar{y}x_i) - a\sum_{i=1}^{n} (x_i x_i - \bar{x}x_i) = 0$$

$$\Rightarrow (\sum_{i=1}^{n} x_i y_i - n\bar{y}\bar{x}) - a(\sum_{i=1}^{n} (x_i x_i - n\bar{x}^2) = 0$$

$$\Rightarrow a = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\,\bar{y}}{\sum_{i=1}^{n} x_i x_i - n\bar{x}^2}$$

# 回归分析：一元线性回归模型参数求解

可以看出只要给出了训练样本$(x_i, y_i)(i = 1, \ldots, n)$，我们就可以从训练样本出发，建立一个线性回归方程，使得对训练样本数据而言，该线性回归方程预测的结果与样本标注结果之间的差值和最小。

这样，对于上面的案例，可以求得参数a和b分别为：

$$a = \frac{x_1 y_1 + x_2 y_2 + \cdots + x_8 y_8 - 8\bar{x}\bar{y}}{x_1^2 + x_2^2 + \cdots + x_8^2 - 8\bar{x}^2} = 1.428$$

$$b = \bar{y} - a\bar{x} = -7.09$$

即预测芒提兹尼欧地区火灾所影响森林面积与气温温度之间的一元线性回归模型为“火灾所影响的森林面积 $= 1.428\times$气温温度 $- 7.09$”，即$y = 1.428x - 7.09$。

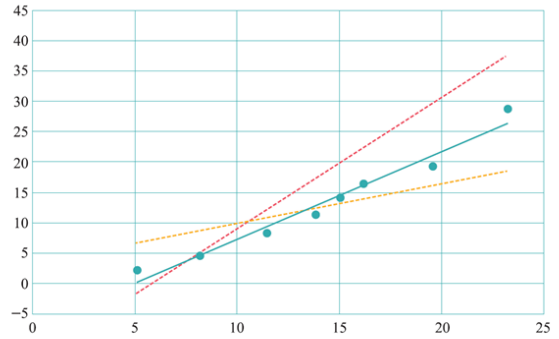


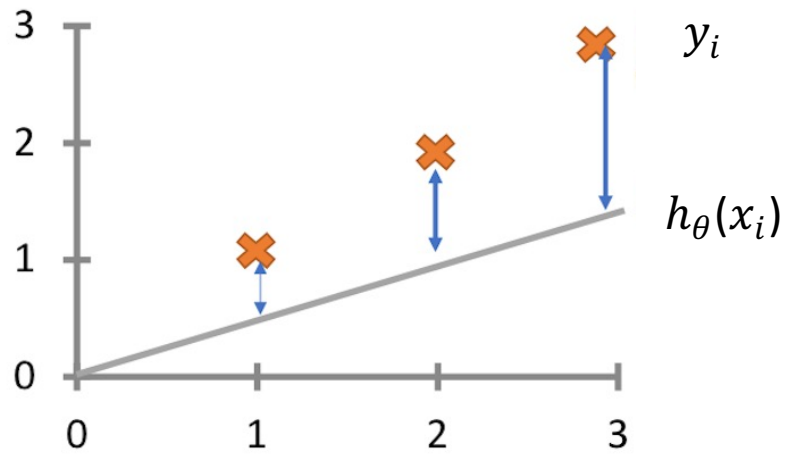
图 4.1　火灾影响的森林面积与气温之间的关系

表 4.3　芒提兹尼欧地区发生森林火灾的部分数据

| 气温 $x$ | 5.1 | 8.2 | 11.5 | 13.9 | 15.1 | 16.2 | 19.6 | 23.3 |
|---|---|---|---|---|---|---|---|---|
| 火灾影响面积 $y$ | 2.14 | 4.62 | 8.24 | 11.24 | 13.99 | 16.33 | 19.23 | 28.74 |

需要求解的回归模型$y_i = ax_i + b\ (1 \le i \le n)$

# 回归分析：梯度下降法求解一元线性回归




优化目标：最小化模型预测值与真实值之间的误差，使用的误差衡量标准是均方误差（MSE），它是每个数据点的预测值与实际值之间差的平方的平均值。

设$h_\theta(x)$为回归值，y 为预测值

优化目标表达式：$J(\theta_0,\theta_1)=\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i)-y_i)^2$

假设：$h_\theta(x)=\theta_0+\theta_1 x$

参数：$\theta_0,\theta_1$

代价函数：$J(\theta_0,\theta_1)=\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i)-y_i)^2$

优化目标：$\underset{\theta_0,\theta_1}{\text{minimize}}\, J(\theta_0,\theta_1)$

# 回归分析：梯度下降法求解一元线性回归

假设$h_\theta(x) = \theta_0 + \theta_1 x$

代价函数：$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2$

对代价函数求偏导：$\frac{\partial}{\partial\theta_j}J(\theta_0, \theta_1) = \frac{\partial}{\partial\theta_j}\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2$

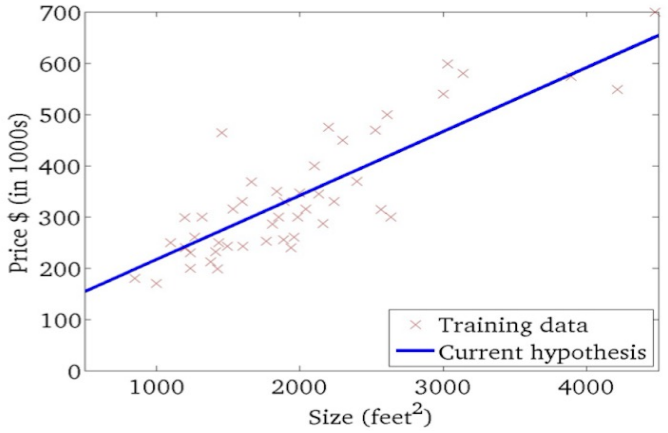
其中，当$j = 0$ 时:$\frac{\partial}{\partial\theta_0}J(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)$

当$j = 1$ 时:$\frac{\partial}{\partial\theta_1}J(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}((h_\theta(x_i) - y_i)\cdot x_i)$

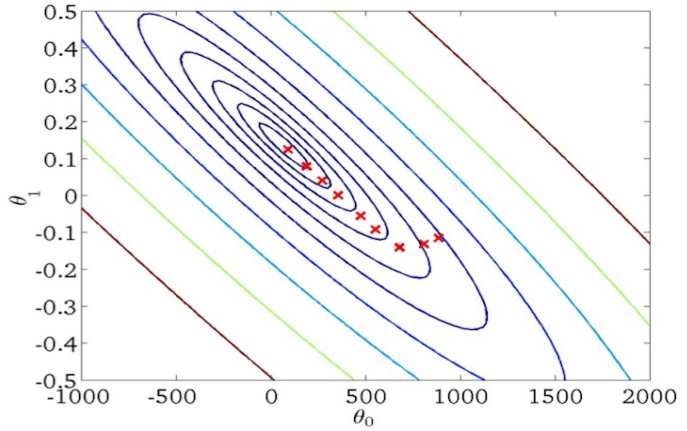
参数更新公式：

$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)$$

$$\theta_1 := \theta_1 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)\cdot x_i$$




$h_\theta(x) = \theta_0 + \theta_1 x$，一个关于x的函数




$J(\theta_0, \theta_1)$的梯度下降过程

# 回归分析：从一元线性回归到多元线性回归

接下来扩展到数据特征的维度是多维的情况，在上述数据中增加一个影响火灾影响面积的潜在因素—风力。

表 4.4　芒提兹尼欧地区历史森林火灾的部分数据（加入风力因素）

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 气温 $x$ | 5.1 | 8.2 | 11.5 | 13.9 | 15.1 | 16.2 | 19.6 | 23.3 |
| 风力 $z$ | 4.5 | 5.8 | 4.0 | 6.3 | 4.0 | 7.2 | 6.3 | 8.5 |
| 火灾影响面积 $y$ | 2.14 | 4.62 | 8.24 | 11.24 | 13.99 | 16.33 | 19.23 | 28.74 |

多维数据特征中线性回归的问题定义如下：假设总共有$m$个训练数据$\{(\boldsymbol{x_i}, y_i)\}_{i=1}^{m}$，其中$\boldsymbol{x_i} = [x_{i,1}, x_{i,2}, \dots, x_{i,D}] \in \mathbb{R}^D$，$D$为数据特征的维度，线性回归就是要找到一组参数$\boldsymbol{a} = [a_0, a_1, \dots, a_D]$，使得线性函数：

$$f(\boldsymbol{x_i}) = a_0 + \sum_{j=1}^{D} a_j\, x_{i,j} = a_0 + \boldsymbol{a^T x_i}$$

最小化均方误差函数：$J_m = \frac{1}{m}\sum_{i=1}^{m}\big(y_i - f(\boldsymbol{x_i})\big)^2$

# 回归分析：从一元线性回归到多元线性回归

为了方便，我们使用**矩阵**来表示所有的训练数据和数据标签。

$$X = [x_1, \dots, x_m], \qquad \boldsymbol{y} = [y_1, \dots, y_m]$$

均方误差函数可以表示为：$J_m(\boldsymbol{a}) = (\boldsymbol{y} - X^T\boldsymbol{a})^T(\boldsymbol{y} - X^T\boldsymbol{a})$

**特征维度=1时线性回归有闭式解，多维情况下同样存在：**


均方误差函数$J_n(\boldsymbol{a})$对所有参数$\boldsymbol{a}$求导可得：

$$\nabla J(\boldsymbol{a}) = -2X(y - X^T\boldsymbol{a})$$

因为均方误差函数$J_n(\boldsymbol{a})$是一个二次的凸函数，所以函数只存在一个极小值点，也同样是最小值点，所以**令$\nabla J(\boldsymbol{a}) = 0$**可得

$$XX^T\boldsymbol{a} = X\boldsymbol{y}; \quad \boldsymbol{a} = (XX^T)^{-1}X\boldsymbol{y}$$

对于上面的例子，转化为矩阵的表示形式为：

$$X = \begin{bmatrix} 5.1 & 8.2 & 11.5 & 13.9 & 15.1 & 16.2 & 19.6 & 23.3 \\ 4.5 & 5.8 & 4. & 6.3 & 4. & 7.2 & 6.3 & 8.5 \\ 1. & 1. & 1. & 1. & 1. & 1. & 1. & 1. \end{bmatrix}$$

$$\mathbf{y} = [2.14 \quad 4.62 \quad 8.24 \quad 11.24 \quad 13.99 \quad 16.33 \quad 19.23 \quad 28.74]^{\mathrm{T}}$$

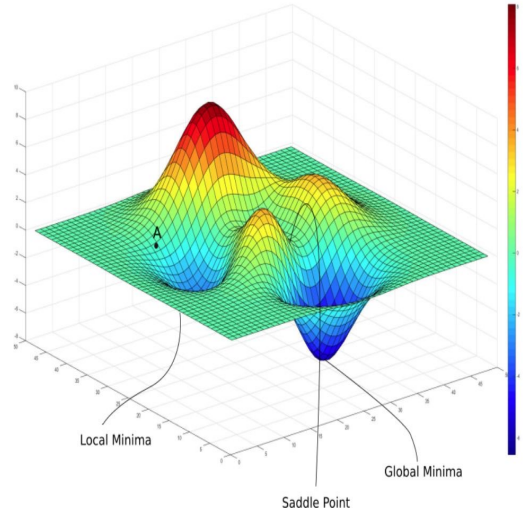
为了**将常数项$a_0$**也纳入矩阵，**矩阵$X$多出一行全1。**

由上式计算可得

$$\boldsymbol{a} = [1.312 \quad 0.626 \quad -9.103]$$

所以对应的线性函数为：

$$y = -9.103 + 1.312x + 0.626z$$



# 回归分析：梯度下降法求解多元线性回归

假设$h_\theta(x_i) = \theta_0 x_{i0} + \theta_1 x_{i1} + \theta_2 x_{i2} + \theta_3 x_{i3} + \ldots + \theta_n x_{in}$

其中$x_i = \begin{bmatrix} x_{i0} \\ x_{i1} \\ x_{i2} \\ x_{i3} \\ \cdots \\ x_{in} \end{bmatrix}, x_{i0} = 1, y_i \in R$

优化目标：$J(\theta) = J(\theta_0; \theta_1; \theta_2 \ldots; \theta_n) = \frac{1}{2m}\sum_{i=1}^{n}(h_\theta(x_i) - y_i)^2$

对代价函数求偏导：

$$\begin{aligned}
\frac{\partial}{\partial\theta_j}J(\theta) &= \frac{\partial}{\partial\theta_j}\frac{1}{2}(h_\theta(x) - y)^2 \\
&= 2 \cdot \frac{1}{2}(h_\theta(x) - y) \cdot \frac{\partial}{\partial\theta_j}(h_\theta(x) - y) \\
&= (h_\theta(x) - y) \cdot \frac{\partial}{\partial\theta_j}\left(\sum_{i=0}^{n}\theta_i x_i - y\right) \\
&= (h_\theta(x) - y)\, x_j
\end{aligned}$$

由$\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta)$,得

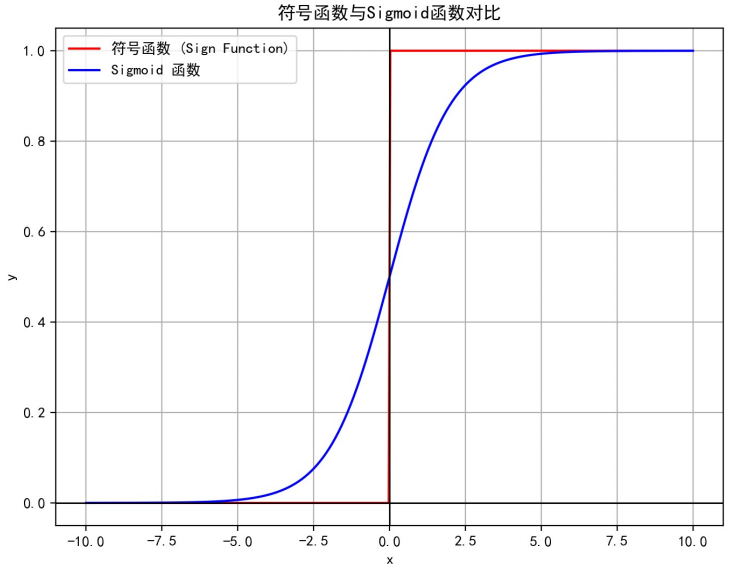
$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)\, x_{i0}$$

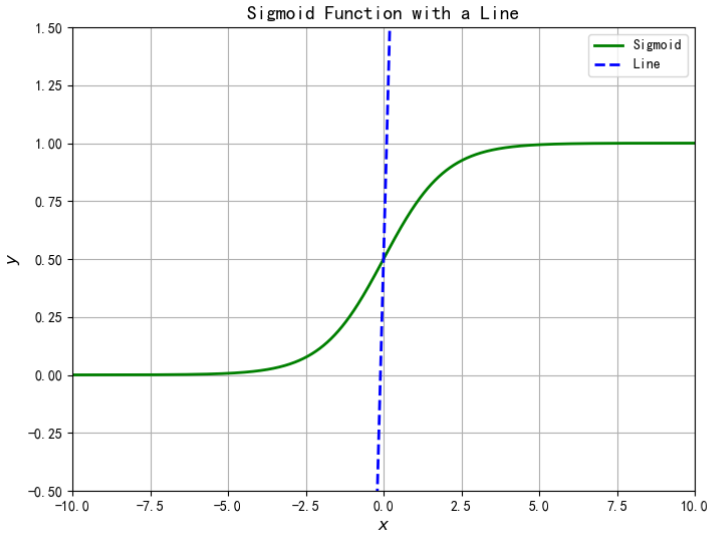
$$\theta_1 := \theta_1 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)\, x_{i1}$$

$$\vdots$$

$$\theta_n := \theta_n - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)\, x_{in}$$

# 分类问题：Sigmoid函数




Sigmoid函数： $\sigma(z) = \frac{1}{1+e^{-z}}$

逻辑回归概率：$f(x)=\sigma(\omega^T x) = \frac{1}{1+e^{-\omega^T x}}$

通过对符号函数的“软化”，Sigmoid函数成为一个连续的函数，从而解决符号函数不可求导的问题

通过分析模型的预测概率，完成分类任务，例如：

概率 > 0.5，则预测为类别1。

概率 < 0.5，则预测为类别0。

# 分类与回归问题：损失函数

- **均方误差**（Mean Squared Error, MSE）

MSE 是回归问题中最常用的损失函数之一。它计算预测值与实际值之间差异的平方，并对所有样本取平均。

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **交叉熵损失**（Cross-Entropy Loss）

交叉熵损失广泛应用于二分类和多分类问题，尤其是在深度学习中。它衡量了真实类别分布与预测类别分布之间的差异。

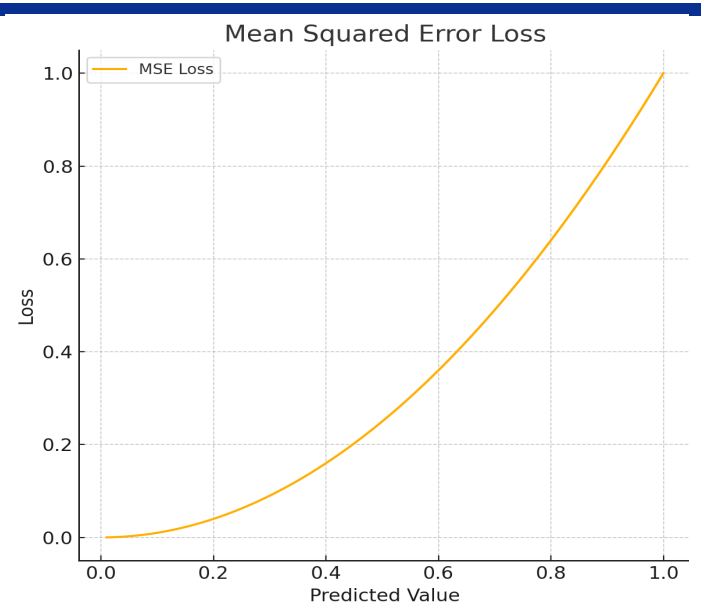
二分类交叉熵（Binary Cross-Entropy）：

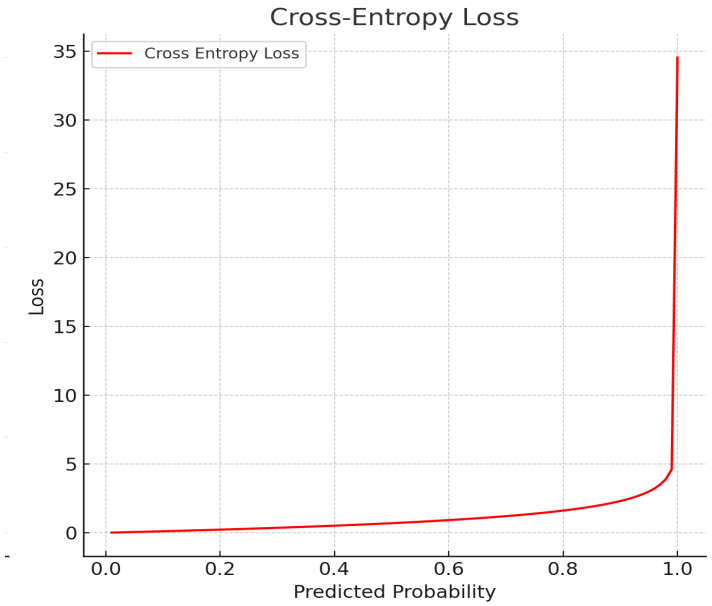
$$\text{BCE} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

**多分类交叉熵**（Categorical Cross-Entropy）：

$$\text{CCE} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c})$$

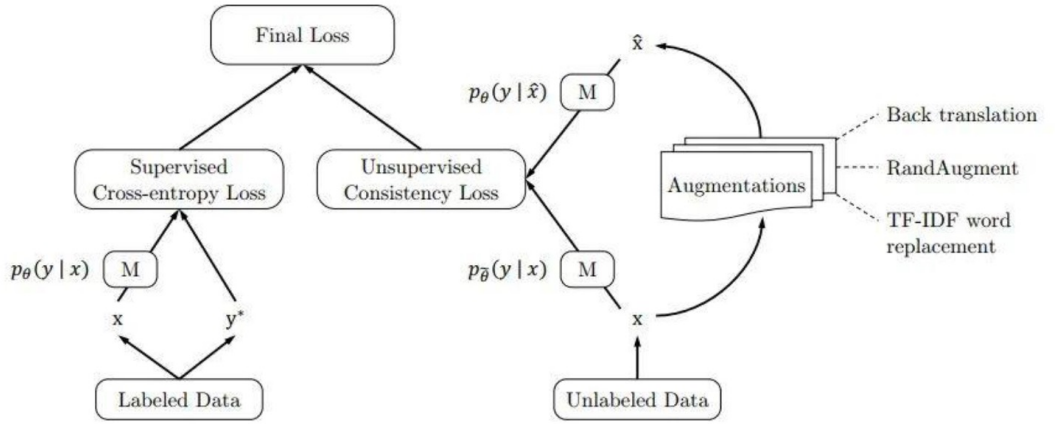
其中 $C$ 是类别数，$y_{\{i,c\}}$ 是样本 $i$ 在类别 $c$ 上的真实标签。



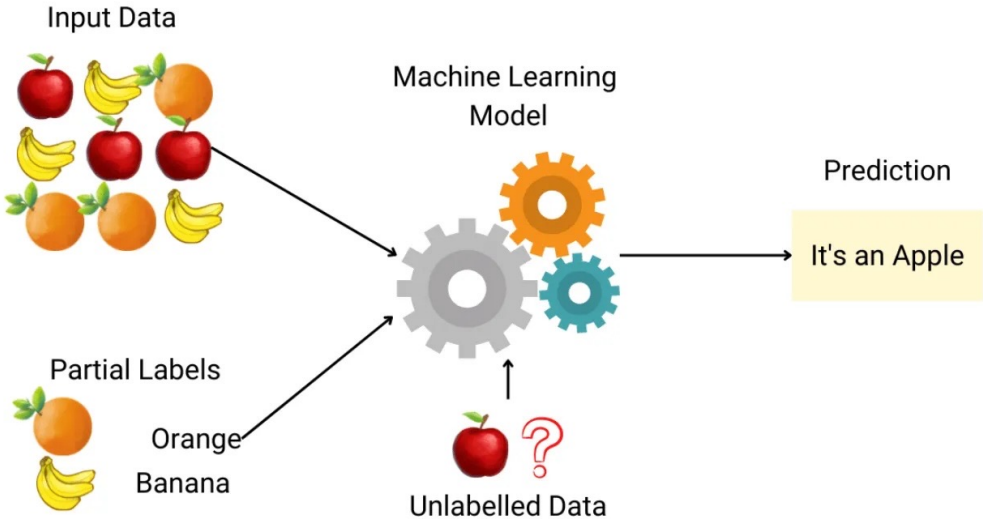



# 机器学习——半监督学习

- **半监督学习（Semi-supervised Learning）**是一种利用少量标注数据和大量未标注数据进行训练的学习方法。它通过在标注数据上进行有监督学习，并通过未标注数据推断潜在的结构或模式，来改善学习效果。




- 半监督学习的基本思想：假设数据集具有某种形式的结构，标注数据和未标注数据之间存在内在的关联性。假设包括：平滑性假设，相似的数据点具有相似的标签；簇假设，相同类别的数据点往往聚集在一起；低密度分离假设，类别的边界位于低密度区域，标注数据集中较少的样本通常位于这些边界区域。



# 机器学习——半监督学习

**常见半监督学习方法：**

- **基于图的模型（Graph-based Models）：**

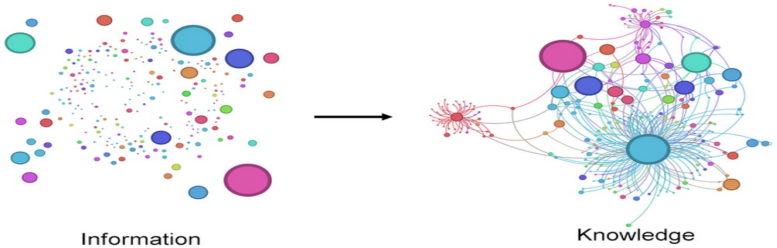
利用数据的图结构表示数据点之间的关系，传播标签信息。

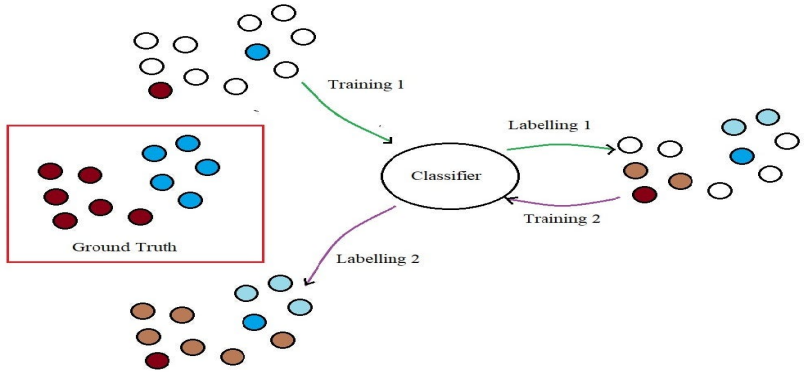
- **自训练（Self-training）：**

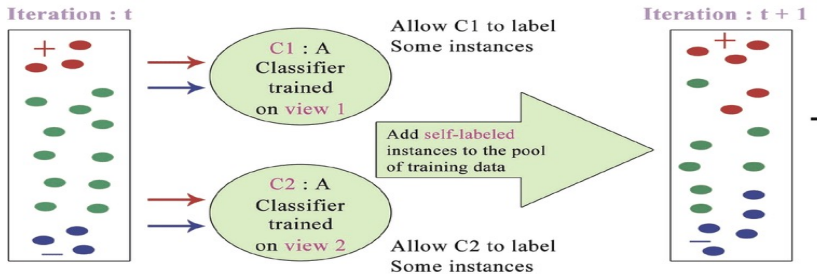
首先用少量标注数据训练一个初始模型，然后用该模型对未标注数据进行预测，并将高置信度的预测结果添加为新的标签，反复迭代直到收敛。

- **协同训练（Co-training）：**

将数据集分为两个不同的视角，利用这两个视角分别训练两个分类器，彼此互相标注未标注数据。



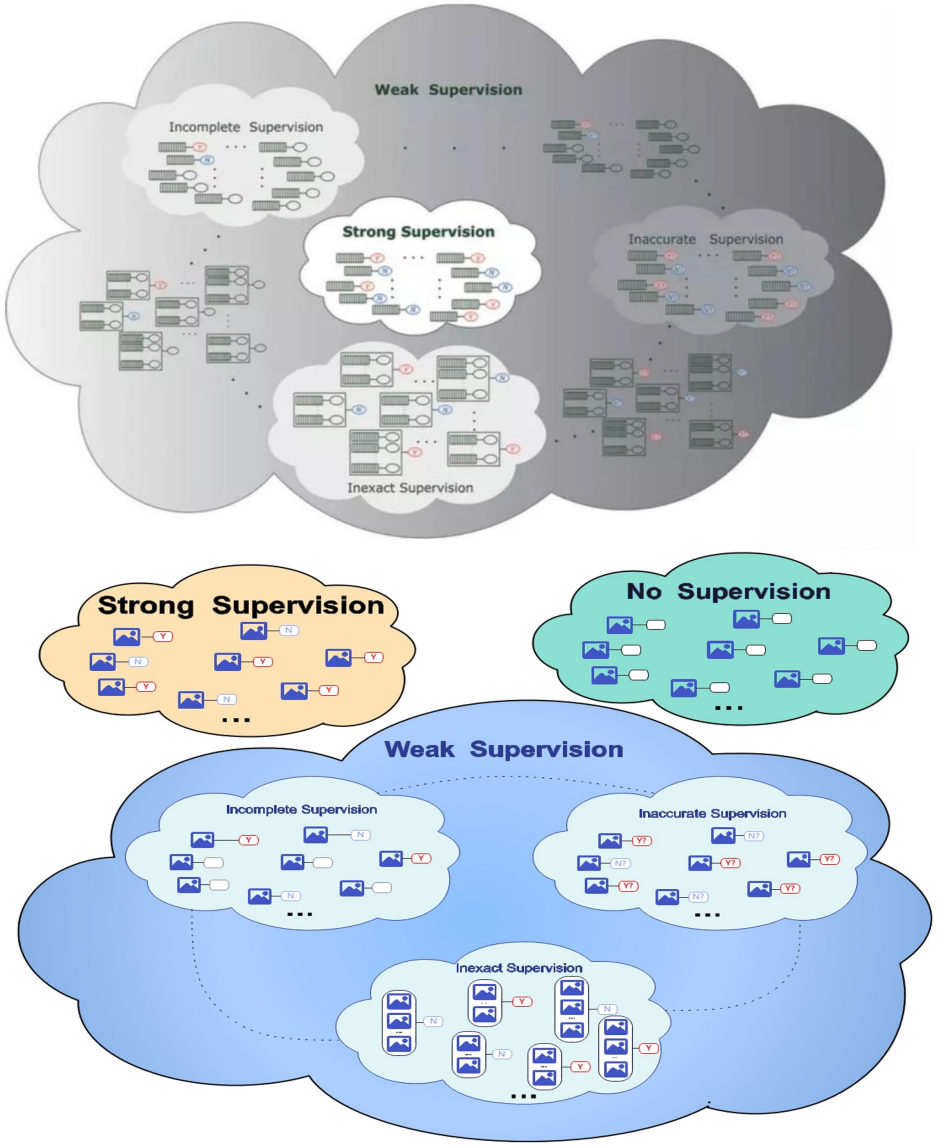
# 机器学习——弱监督学习

- **弱监督学习（Weakly Supervised Learning）**是一类使用部分或不完全标签来训练机器学习模型的技术。它通常利用不完全标注、低质量标注或者无标签的辅助信息来推断出更加准确的预测。
- **弱监督可以分为三类：**

  **1. 不完全监督（incomplete supervision）**，只有训练集的一个较小子集有标签，其他数据则没有标签。这种情况发生在各类任务中。

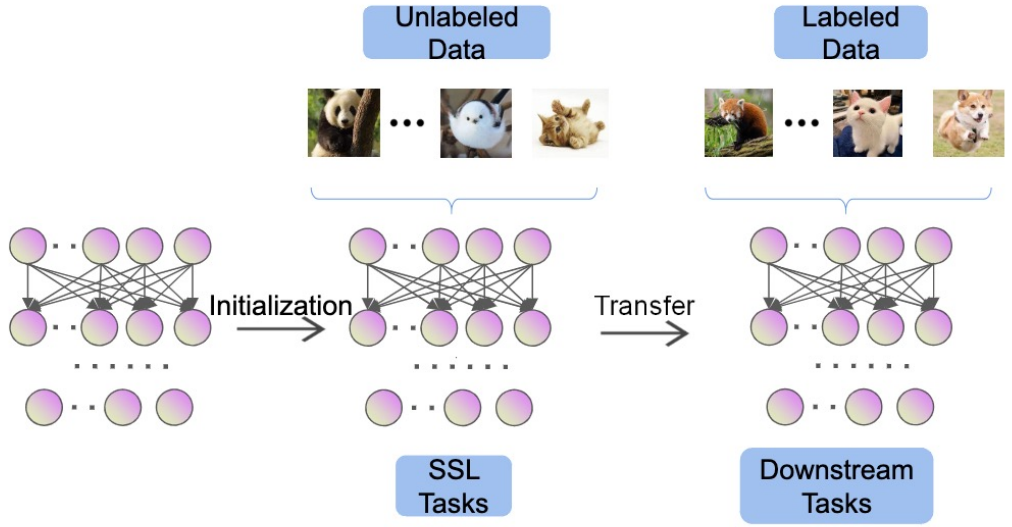
  **2. 不确切监督（inexact supervision）**，只有粗粒度的标签，粗粒度的标签对于细粒度的任务来说帮助很有限。

  **3. 不准确的监督（inaccurate supervision）**，模型给出的标签不总是真实的。



# 机器学习——自监督学习

- **自监督学习（Self-supervised Learning，SSL）**是一种无监督学习方法，不同于传统的无监督学习，自监督学习通过设计合适的任务（pretext task），从未标记的数据中自我生成标签，进而进行学习。自监督学习通常依赖于生成“伪标签”或“辅助任务”（auxiliary tasks）。




自监督学习的经典流程包括以下几个步骤：

**1.预设目标任务：**

设计一个与原始数据相关的预测任务，这些任务可以是数据的一部分的预测，或是数据内部某种结构的恢复。

**2.生成伪标签：**

根据设计的目标任务从原始数据中生成“伪标签”。这些伪标签是通过对输入数据进行某种形式的转换或遮挡生成的。

**3.训练模型：**

使用原始数据和伪标签训练模型。目标是通过训练学习到一个能够根据输入数据生成准确预测的模型。

**4.迁移到下游任务：**

训练好的模型可以迁移到其他下游监督学习任务，比如分类、回归等。

# 小结

- 通过最小化经验风险ERM来优化模型参数
- 选择MLE和MAP优化模型都可以，选择缘于是否对模型参数有预先假设
- 线性回归可以用最小二乘法（精确但复杂度高）也可以用梯度下降法（通用但可能不是全局最优）
- 广泛来讲，机器学习模型可以基于MSE或者交叉熵进行训练
- 现实场景中，除全监督学习之外半监督、弱监督和自监督学习应用更广泛

# 本节课小结

1959年7月，塞缪尔发表了一篇名为“通过国际跳棋进行机器学习的研究”论文中，第一次使用了“机器学习”这一术语[Samuel 1959]。在人工智能发展早期，使用国际跳棋而非国际象棋或围棋，是因为国际跳棋中对落子的选择和判断相对简单。

**从数据中学习概念或模式形成判断和决策能力是机器学习的一个基本目标。**监督学习从标注数据出发，学习得到一个映射函数，将原始输出映射到语义任务空间，架构起了“从底层特征到高层语义”的“桥梁”。

如何形成**数据依赖灵活、且在学习过程中有效利用知识或先验**，是机器学习今后发展的重要方向。同时，一个任务是否可以**被学习（Learnability）**仍然是学术界的热点和难点。