数据科学概论
Introduction to Data Science

龚舒凯

中国人民大学

Renmin University of China

School of Applied Economics/ School of Statistics

shukai_gong@ruc.edu.cn

目录

题型

（每题 3 分）10 个不定项选择，每个不定项选择 5 个选项
（每题 5 分）简答题 ×5
（每题 15 分）论述题 ×3

考试重点

第一章：绪论

1. 基本数据科学方法：
2. 数据科学与统计学的关系：

- 数据科学有机结合了诸多领域中的理论和技术，包括数学、统计学、模式识别、机器学习、数据可视化、数据库、高性能计算
- 数据科学的前身是统计学。

3. 统计学发展历史

- 统计学划分为三个时期：古典记录统计学、近代描述统计学、现代推断统计学

第二章：编程

1.Python 选择题

第三章：数学基础

1. 概念性、定义性的概率论数理统计

第四章：统计原则

1. 有个大题
2. 怎么度量变异性?
3. 怎么度量稳定性?
4. 可计算原则里：数据量很大的时候、维数很高的时候要做什么?

第五章：可视化

1. 各种图是为了解决什么问题
2. 不同数据类型应该画什么图

第六章：数据挖掘和机器学习

1. 无监督学习有哪些?
2. 有监督学习有哪些?

第七章：人工智能

1. 感知机思想 + 感知机的训练过程
2.BP 算法为什么可以往后?
3. 神经网络的构造是什么?
4. 激活函数有哪些？性质如何?

第八章：行业运用

1 2020-2021《数据科学》概论真题

(5 分不定项选择) 今有一组数据 $(x_1, x_2..., x_n)$，将数字 y 和每个数据的差异称为离差，则使得所有离差的绝对值之和最小的 y 是这组数据的 (　　)
A. 偏度　　B. 标准差　　C. 中位数　　D. 众数　　E. 均值

(5 分不定项选择) 在本学期的研究中，哪种研究不完全是数据驱动的 (　　)
A. 特征学习　　B. 人工智能　　C. 深度学习　　D. 机器学习　　E. 以上皆不是

(5 分简答题) 请根据自己的理解简述数据挖掘过程中的必要步骤
答:

- 问题理解：明确问题、清晰地定义问题并确定目标
- 数据理解：了解相关行业情况、掌握相关知识，理解数据的业务意义并以此为基础进一步探索教据
- 数据准备：对数据进行清洗、整理和转换 (如处理缺失值、异常值；或数据结构的转换)
- 数据建模：分析建模过程，根据实际情况选择合适的模型
- 模型评估：从模型本身效果 (技术)、是否适用于实际 (业务) 的角度对模型进行评估
- 模型部署：将模型应用到实际的生活场景中

(5 分简答题) 简要说明数据科学的发展历史
答：分为四个阶段，分别是古典记录统计学 (Petty，Achenwall，Pascal，Fermat，Bayes)；近代描述统计学 (Moivre，Laplace，Quetelet，Gauss，Bayes)；现代推断统计学 (Gosset，Fisher，Lasso，Bayes)；当代统计学—数据科学。数据科学的发展历程经历了以上四个发展阶段，**数据科学以传统统计学为基础、结合计算机科学，尽可能地使数据价值最大化。**

(5 分简答题) 人工智能有哪些研究领域和应用领域?
答:（来自高瓴人工智能“学院介绍”）

- **研究领域：计算机视觉** (图像识别、图像理解、无人驾驶); **语音** (声学模型；语言合成与语音识别);**NLP**(机器翻译、情感分析、语义理解，认知策略层面); **优化运筹** (路径规划、智能调度、智能运维); **图形图像学** (avatar 的生成，VR，AR，辅助设计);
- **应用领域：**金融、医疗、电商、机器人、互联网...

（15 分论述题）列举 3 个数据科学在金融中的应用场景，并作简单介绍并举出对应的实例

答：

- 1. 数据商业链: 在数据时代的大背景下，金融业中所有客户的信息通过社交网络得以生成和传播，然后被例如百度等其他大型接索引擎所组织排序和检索，再通过数据科学家进行教据分析最终形成有价值的社交商业链。实例：阿里巴巴入股新浪微博，这一事件实现了社交媒体与电商平台的合作，阿里还在此基础上推出了阿里小贷。

- 风险管理与控制: 金融企业通过数据科学领域中的机器学习算法（如 LDA，Logistic 回归）对教据进行训练和分析，进一步建立相关的风险评级模型，进而增加风险管理的效率和可持续性。实例：京东金融、网易金融等金融企业根据自己的数据库建立了反款诈模型和交易行为风险模型等，在一定程度上有效的解决或避免了信贷欺诈问题。

- 量化投资介绍：在该场景下，金融企业以数据科学和经济金融理论为基础，对投资想法构建教学模型，利用历史教据对模型进行训练和回观验证，从中选出最优的模型来指导投资决策。实例：国泰君安推出的量化交易平台上提供智能投顾的功能，利用数据科学的优点并结合投资人对风验厌恶程度、期望收益和市场动态，采用多种算法和模型为客户设计投资组合，为其提供综合的资产配置服务。

（15 分论述题）假设我们利用样本中位数估计总体中位数，请描述如何利用 Bootstrap 方法计算该统计量的变异性

答：利用 Bootstrap 法计算样本中位数变异性的步骤如下：

（1）在总体中随机抽样出一份样本 $\boldsymbol{Y} = (Y_1, ..., Y_n)$

（2）从样本 $\boldsymbol{Y} = (Y_1, ..., Y_n)$ 进行 n 次有放回的抽样，得到一个 Bootstrap 样本 ${\boldsymbol{Y_1}}^{*(1)}$（注：这样一个 Bootstrap 样本 ${\boldsymbol{Y_1}}^{*(1)}$ 里面有 n 个数 Y_i）

（3）计算出这个 Bootstraps 样本 ${\boldsymbol{Y_1}}^{*(1)}$ 的中位数 m_1

（4）将上述（2）（3）过程重复 B 次，得到 B 个 Bootstrap 中位数 $(m_1, m_2, ..., m_B)$

（5）计算这 B 个 Bootstrap 中位数的标准差 s_B，公式如下：

$$\mathrm{Var}[\hat{S}(\boldsymbol{Y})] = \frac{1}{B-1}\sum_{b=1}^{B}(S(\boldsymbol{Y}^{*(b)}) - \bar{S}^{*})^2$$

其中 $S(\boldsymbol{Y}^{*(b)})$ 是第 b 个 Bootstrap 样本的中位数，$\bar{S}^{*}$ 是 B 个 Bootstrap 样本中位数的中位数

2 数据科学的数学基础

2.1 线性代数

1.Hadamard 乘积：$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11} \cdot b_{11} & a_{12} \cdot b_{12} & \ldots & a_{1n} \cdot b_{1n} \\ a_{21} \cdot b_{21} & a_{22} \cdot b_{22} & \ldots & a_{2n} \cdot b_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{m1} \cdot b_{n1} & a_{m2} \cdot b_{n2} & \ldots & a_{mn} \cdot b_{mn} \end{bmatrix}$

2. 非奇异阵：（就是可逆阵）

3. 正交矩阵：使得 $\boldsymbol{QQ^T = E}$ 的矩阵 $\boldsymbol{Q}$

- 矩阵中任意两个不相同的向量内积为 0，说明这两个向量垂直（正交）

4. 矩阵的迹：矩阵对角线上的元素的和称为迹，记作 $\mathrm{tr}\boldsymbol{A}$

- $\mathrm{tr}(\boldsymbol{A} \pm \boldsymbol{B}) = \mathrm{tr}(\boldsymbol{A}) \pm \mathrm{tr}(\boldsymbol{B})$
- $\mathrm{tr}(\boldsymbol{AB}) = \mathrm{tr}(\boldsymbol{BA})$

2.2 概率论与数理统计

1. 史实：Pascal 创立；Laplace 古典概率论完善；Kolmogorov（柯尔莫哥洛夫）概率论公理化系统，现代概率论

2. 随机试验：

- 可以在相同的条件下重复进行;
- 每次试验的可能结果不止一个，并且能事先明确试验的所有可能结果;
- 进行一次试验之前不能确定会出现哪一个结果;

对于某个随机试验 E，把所有可能结果组成的集合称为 E 的样本空间。

3.Bernoulli 试验：

- 试验 E 只有两种可能结果：A，$\bar{A}$
- n 重 Bernoulli 试验：重复地做 n 次的试验构成了一个新试验
- $P(B_k) = C_n^k p^k (1-p)^{n-k}, 0 \leq k \leq n$

4. 随机变量：对于该试验中每个 B_k 的次数可以用变量来描述，这个变量的取值都在样本空间 S 中，而且有很好的方式来描述其中的概率，把这一类变量称为随机变量。

5. 随机事件：样本空间的子集

6. 分布：即分布函数 $F(X) = P(X \leq x), -\infty < x < +\infty$，称 X 服从分布 $F(X)$

7. 离散型随机变量：X 取有限个/可列个值. 如果设

$$p_k = P(X = x_k)$$

则 p_k 称为 X 的分布律（理解为一个数列）

8. 连续型随机变量：存在 $(-\infty,+\infty)$ 上的非负实值函数 $f(x)$ 使得分布函数：

$$F(x)=\int_{-\infty}^{x} f(y)dy, -\infty<x<+\infty$$

则称 X 为连续型随机变量，$F(X)$ 为连续性分布函数，$f(x)$ 为 X 的概率密度函数。连续型随机变量没有分布律。

9. 正态分布：记作 $X\ N(\mu,\sigma^2)$，标准正态分布 $X\ N(0,1)$

- μ 影响钟形曲线对称轴位置，σ^2 影响钟形曲线的扁平程度（σ^2 越大越平）
- $f(x)_{\max}=f(\mu)=\dfrac{1}{\sqrt{2\pi}\sigma}$
- $f(x)$ 在 $x=\mu\pm\sigma$ 处有拐点

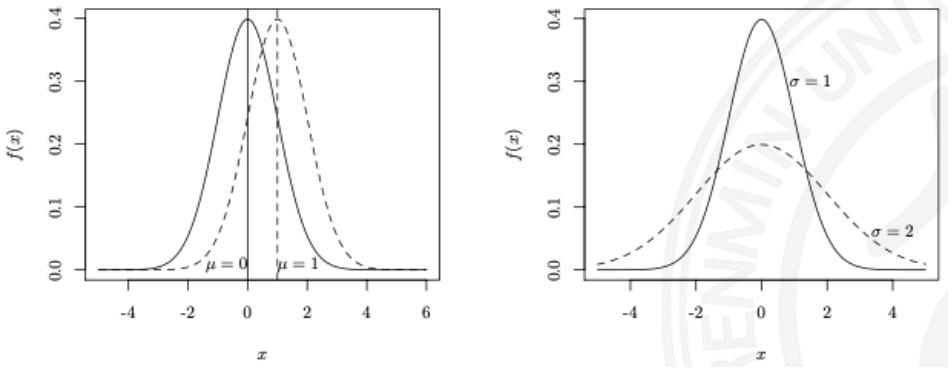


图 1:

10. 总体与个体：被研究的对象的全体称为总体，组成总体的元素称为个体。总体是一个具有确定分布的随机变量 X，需要研究其分布函数 $F(X)$

11. 总体与样本：总体的 $F(X)$ 未知，用样本来推断 $F(X)$

- 从总体中抽取 n 个个体进行观察或试验的过程称为抽样;
- 得到的 n 个随机变量**彼此独立并且与 X 具有相同的分布**，称之为样本;
- 该样本容量为 n

12. 随机变量的期望

离散型随机变量：$\sum\limits_{k\geq 1} x_k p_k$（如果 $\sum\limits_{k\geq 1}|x_k|p_k$ 收敛）

连续型随机变量：$\int_{-\infty}^{+\infty} xf(x)\mathrm{d}x$（如果 $\int_{-\infty}^{+\infty}|x|f(x)\mathrm{d}x$ 收敛）

计算规则如下：

$$E(k)=k,\ E(kX)=kE(X),\ E(X_1+X_2)=E(X_1)+E(X_2)$$

13. 统计量的定义：
对于一组样本 $\{X_1, X_2, ..., X_n\}$ 来说，如果存在某个函数 $g(x_1, x_2, ..., x_n)$ 是关于 $x_1, x_2, ..., x_n$ 的连续函数，并且不再包含任何其他未知的参数，则称 $g(x_1, x_2, ..., x_n)$ 为统计量。

- 若 $\{x_1, x_2, ..., x_n\}$ 为样本观察值，则 $g(x_1, x_2, ..., x_n)$ 为统计量的观察值
- 统计量也是随机变量

14. 顺序统计量：
假设 $\{X_1, X_2, ..., X_n\}$ 是来自总体 X 的样本，定义一个统计量 $X(k)$，对任意一组样本观察值 $\{x_1, x_2, ..., x_n\}$，将其从小到大排成 $x_(1) \leq x_(2) \leq ... \leq x_(n)$，它总是取其中的第 k 个值 $x(k)$，则称 $X(k)$ 是样本 $\{X_1, X_2, ..., X_n\}$ 的第 k 位顺序统计量。

- 最大值、最小值、中位数都属于顺序统计量。

Pearson 相关系数：设变量 x, y 有 n 组观测：(x_i, y_i)，$i = 1, 2, ..., n$，则 x, y 的 Pearson 相关系数为：

$$P = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \cdot \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

容易证明 $P \in [-1, 1]$

2.3 最优化方法

1. 最优化方法定义：最优化方法是在所有的可行方案中找出最优方案的方法。

- 最优化方法严格来说并不是数据分析的方法，因为它不是从历史数据中发现规律和建立模型。但最优化方法是数据科学中不可或缺的重要方法。
- 可以根据模型的数学公式推导求得精确的解析解
- 通过一些算法来求得近似的数值解。

2. 最优化算法实现：将要解决的实际问题转化为数学问题，然后建立最优化模型并调用算法进行求解。（R，Lingo，Python 包）

3. 无约束的非线性规划：对于任意非线性函数，求其最小值/最大值是普遍存在的问题。比如用**最小二乘法估计二元回归模型系数要求残差的平方和最小**，即:

$$\min z = \sum_{i=1}^{n}(y_i - (\alpha + \beta_1 x_{1i} + \beta_2 x_{2i}))^2$$

求得最优解时的 $\hat{\alpha}, \hat{\beta}_1, \hat{\beta}_2$ 的值就是回归系数的估计值。

-

3 数据科学的统计原则

Bin Yu 教授指出：数据科学是计算与统计思维的必然融合
数据科学的三个统计原则（RPC）：

- 可重复原则 (Reproducibility)
- 可预测原则 (Predictability)
- 可计算原则 (Computability)

3.1 可重复原则

1. 定义：数据或模型发生一定程度的扰动时，分析结果依然能够保持相对一致。

2. 扰动的来源有：

- **分析数据的扰动**：（1）原始数据的采集过程中的测量误差（2）数据清洗和整理过程中出现的扰动（3）抽象变异性
- **分析工具/方法的扰动：**

3. 统计推断中的几个基本概念：

- **总体**：总体可以通过一个或多个参数进行刻画。比如

  （1）总体的期望 $E(Y)$

  （2）多个变量间的关系：如多元线性回归模型：

  $$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \varepsilon$$

  其中：Y 在该模型中被称为**响应变量**，$X_1, ..., X_p$ 被称为**协变量**，ε 是**随机误差项**，通常假定其 $\sim N(0, \sigma^2)$，而 $\beta_0, \beta 1, ..., \beta_p$ 和 σ^2 则是研究者感兴趣的**总体参数**。

- **抽样**：抽样所得到的试验单元集合构成总体的一个样本
- **统计量**：由样本信息构造的**函数**称为统计量（统计量是一个函数，比如 $\bar{x}, \sigma$ 等等）
  统计量只依赖于样本信息，它不含总体的任何未知参数，可以记作 $s(y_1, ..., y_n)$ 或者 $s[(y_1, \boldsymbol{x_1}), ..., (y_n, \boldsymbol{x_n})]$，比如：

  （1）**样本均值**：$\bar{y} = \frac{1}{n} \sum \lim_{i=1}^{n} y_i$ $\quad \Rightarrow$ 只依赖于样本信息 $(x_1, ..., x_n)$

  （2）**最小二乘估计**：总体参数 $\hat{\beta}$ 只取决于样本信息 $\boldsymbol{Y} = (y_1, ..., y_n)^T$，$\boldsymbol{X} = (x_1, ..., x_n)^T$

  $$\hat{\beta} = \arg\min ||\boldsymbol{Y} - \boldsymbol{X\beta}||^2$$

  其中 $\boldsymbol{Y} = (y_1, ..., y_n)^T$，$\boldsymbol{X} = (x_1, ..., x_n)^T$，可以解得 $\hat{\beta} = \boldsymbol{(X^T X)^{-1} X^T Y}$

4. 抽样变异性：由于抽样过程的随机性，研究者无法保证每一次抽取的样本是完全一样，这就会导致基于样本计算的统计量的取值在每一次抽样过程中具有变异性。

5. 抽样分布：统计量的分布也称抽样分布，对抽样分布的刻画主要有两种方式：

- **精确分布**：通过总体或误差项的正态假定来到处抽样分布，例如

  假定 $Y \sim N(\mu, \sigma^2)$，则 $\bar{Y} \sim N(\mu, \sigma^2/n)$（回顾：抽样分布的方差与总体的方差关系满足：$\sigma_x^2 = \frac{\sigma^2}{n}$）

- **渐进分布**：通过中心极限定理导出抽样分布。
  （回顾**中心极限定理**：设随机变量序列 $X_1, X_2, ...$ 独立同分布，且具有有限期望和方差，即 $E(X) = \mu$, $D(X) = \sigma^2$，那么 $\bar{X}$ 的分布将近似于正态分布 $N(\mu, \sigma^2/n)$

6. 抽样变异性的度量：

- **统计量的变异性**：计算简单，但对抽样总体有严格假定，复杂统计量的抽样分布难以得到

  - **参数方法**：利用抽样分布计算统计量的样本方差，比如：
    （i）均值的样本方差为：$\hat{\sigma}^2/n$
    （ii）最小二乘估计的样本方差为：$(\boldsymbol{X^T X})^{-1}\hat{\sigma}^2$
  - **非参数方法：利用 Bootstrap 法**计算统计量 $S(\cdot)$ 的样本方差，这个方法适用于抽样分布无法准确求出的情形，对总体的假定更宽松，但计算量较大。
    步骤如下：

    * 从样本 $\boldsymbol{Y} = (Y_1, ..., Y_n)$ 中进行 n 次有放回的随机抽样，得到一个 Bootstrap 样本 $\boldsymbol{Y^{*(1)}}$(注：这个 $\boldsymbol{Y^{*(1)}}$ 中有 n 个 Y_i)
    * 重复该过程 B 次得到 B 个 Bootstrap 样本（每个 Bootstrap 样本里都有 n 个 Y_i，只是顺序/个数不同，这样就可以得到 B 个互异的 Bootstrap 样本）。
    *
    $$\mathrm{Var}[\hat{S}(\boldsymbol{Y})] = \frac{1}{B-1}\sum_{b=1}^{B}(S(\boldsymbol{Y}^{*(b)}) - \bar{S}^*)^2$$

    其中，$\bar{S}^* = \frac{1}{B}\sum_{b=1}^{B} S(\boldsymbol{Y}^{*(b)})$.

- **变量选择结果的稳定性**：
  纳入众多**无关协变量**的回归模型:（1）模型复杂，容易产生过拟合；（2）多重共线性，统计推断失效；（3）湮没真正重要的协变量的影响，解释性差
  从众多协变量中筛选出真正重要的变量的过程就是变量选择。
  变量选择结果的稳定性可以通过以下三种方式进行度量：

  - **序列不稳定性：用于度量当样本量减小时，变量选择结果的稳定性。操作如下：**
    * 首先基于完整样本选择出一个重要变量的集合;
    * 然后随机地从完整样本中删去部分样本，再重新进行变量选择;
    * 计算不完全样本所得到的重要变量的集合和基于所有样本得到的重要变量集合的对称差;
    * 将上述过程重复进行 1000 次，求其平均值就可以得到序列不稳定性。
  - **Bootstrap 不稳定性**：
    * 首先基于完整样本利用模型选择方法建模，得到 n 个样本观测的拟合值 $\hat{y}_i$, $i = 1, ..., n$ 以及误差项方差的估计值 $\hat{\sigma}^2$，记 $\bar{y} = \frac{1}{n}\sum_{i=1}^{n}\hat{y}_i$

    - 然后利用从分布 $N(\bar{y}, \hat{\sigma}^2)$ 中进行再抽样，得到 Bootstrap 样本 $(\boldsymbol{x_i}, y_i^*)$
    - $(\boldsymbol{x_i}, y_i^*)$ 再利用模型选择方法基于样本 $(\boldsymbol{x_i}, y_i^*)$ 选择出重要变量，计算 Bootstrap 样本所得到的重要变量的集合和基于原始样本得到的重要变量集合的对称差;
    - 将上述过程重复进行 1000 次，求其平均值就可以得到 Bootstrap 不稳定性。
  - **扰动不稳定性**:
    - 从正态分布 $N(0, \tau\sigma^2)$ 中产生扰动误差 ξ_i，其中参数 $0 < \tau < 1$ 控制着扰动量的大小，σ^2 是基于原始数据所得到的误差项方差的估计
    - 对于每一个 τ，我们利用 $\tilde{y}_i = y_i + \xi_i$ 重复产生扰动数据 100 次，然后将变量选择方法用在扰动数据 $(x_i, \tilde{y}_i)$，$1 \le i \le n$ 上进行变量选择
    - 对于每一个 τ，研究者可以计算基于扰动数据的变量选择结果与基于原始数据的变量选择结果的对称差，然后将其绘制在坐标图中，如果线条越靠近左上角，那么模型选择结果的不稳定性将会越大。

7.Bootstrap 组合方法:

- **Bagging 算法**：利用样本数据 $Z = [(y_1, \boldsymbol{x_1}), ..., (y_n, \boldsymbol{x_n})]$ 来推断总体回归模型，获得回归函数的估计 $\hat{f}(x)$ Bagging 算法的基本思想是基于 Bootstrap 样本训练出来的**多个回归函数的估计进行平均**，由此减少估计的变异程度。
  - 利用 Bootstrap 方法从原始样本 Z 抽取 Bootstrap 样本 $Z^{*(b)}$
  - 基于该 Bootstrap 样本训练得到回归模型 $\hat{f}^{*(b)}(x)$
  - 最终 Bagging 估计量定义为:
$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*(b)}(x)$$
- **随机森林 (Random Forest) 算法**:
  - 从原始数据集中有放回随机抽取一个样本量为 N 的 Bootstrap 样本 Z
  - 利用 Bootstrap 样本数据训练一棵随机森林模型 T_b，训练一棵树的基本操作流程如下:
    - 随机选择 m 个特征（或称为变量），通常是将总特征数的根号（或 $\log_2$）向下取整，作为**该决策树的候选特征集合**。
    - 对于每个**节点**（即数据集的某个子集），从 m 个候选特征中选择**最优的特征**，根据**该特征的取值**将节点**分裂成两个子节点**。
    - 不断递归这个过程，直到满足某个停止条件（例如达到预设的最小节点数 n）为止。
  - 重复 B 次对每棵树的操作，得到 B 棵树
  - 输出 B 棵随机森林树 $T_b^{(i)}, i = 1, 2, ..., B$
  - 对于回归问题，随机森林算法得到的最终估计为
$$\hat{f}_{\text{rf}}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B}T_b(x)$$
  - 对于分类问题，最终的预测类别可以通过 B 棵随机森林树投票来确定

8. 模型的扰动：统计推断需要对总体给予一定的假定（如在回归模型中，我们假定误差项服从正态分布）。**模型的扰动**是为了探究当总体分布或模型**偏离于原定的假设时数据分析结果的变化**。分析结果的变化较小的方法具有稳健性。

3.2 可预测原则

1. 可预测原则的定义：数据：协变量数据集 $\boldsymbol{X}$，因变量数据集 $\boldsymbol{y}$，目标：建立预测模型 f，使得预测结果 $\hat{\boldsymbol{y}} = f(\boldsymbol{X})$ 与 $\boldsymbol{y}$ 的差距很小

2. 过拟合：导致可预测性不良的原因之一。

3. 交叉验证法：数据科学中常用的**可预测性评价方法**，可以指导研究者在不同的模型间进行选择。

- 核心思想是**将"建立模型"和"评价预测"的数据分开**
- 实现提升拟合效果的同时避免过拟合的目标。

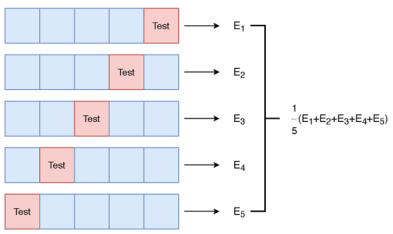
要素有：

- **训练集**：用于建立模型的数据集
- **测试集**：用于评价预测的数据集
- **损失函数**：评价模型拟合好坏的函数，如平方损失函数 $L = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

概括一下，交叉验证法的思路就是把数据分成两半，一部分数据作为训练集，一部分数据作为测试集，基于训练集练模型，基于数据集测试模型看它好不好。

4. 常用的交叉验证方法（本质上关心怎么划分训练集和测试集）：

- **保留交叉验证 (Handout Method)**：将数据**随机划分**为训练集和测试集，在训练集上估计模型，在测试集上评价预测效果
  - 一般研究中训练集：测试集 =7:3
  - 要求多次保留交叉验证
  - 用损失函数评估模型的预测准确度
- k **折交叉验证 (K-folds Cross Validation)**：将数据随机等分为 k 份，使用其中一份作为测试集，剩余 $k-1$ 份作为训练集，并将上述过程重复 k 次，每次使用不同的测试集。最后取 k 次的损失函数平均值模型来看拟合效果（相当于测试集轮流坐庄，要跑 k 次模型）



- **留一交叉验证 (Leave-one-out)**：看作 k 折交叉验证的特例，将数据中的每一个样本轮流用于测试，其余的 $n-1$ 个样本轮流作为数据集

3.3 可计算原则

1. 狭义定义：模型或算法是否具备可计算性；
广义定义：数据分析的整个过程（如数据清洗、预处理）中的计算问题

- 最小一乘估计：$\sum_{i=1}^{n}|Y-(\beta_0+\beta_1 x+\varepsilon)|$ 不可导

2. 大数据时代的数据特征：数据量大；数据维度高
3. 大规模数据的处理方法：

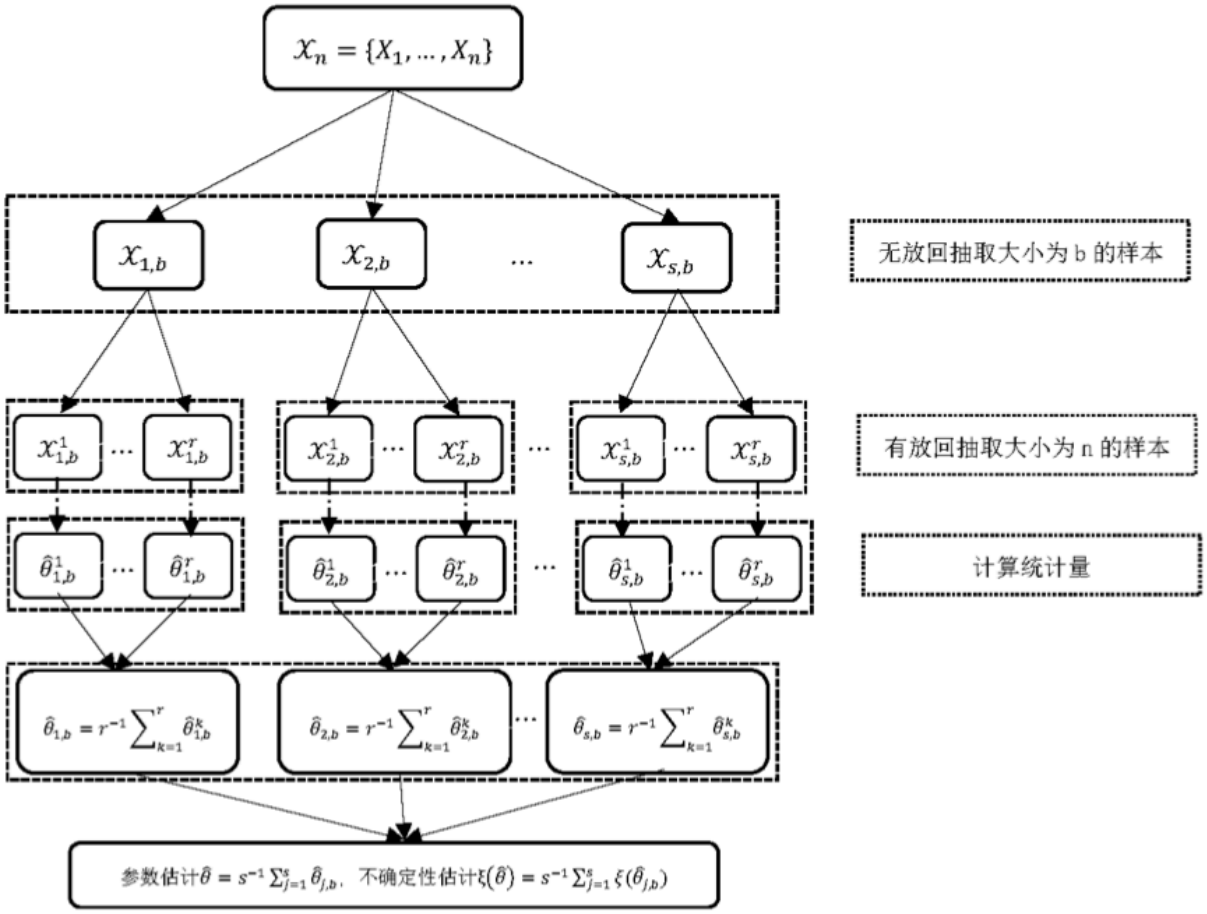
- **分布式储存**：通过网络使用企业中的每台机器上的磁盘空间，并将这些分散的存储资源构成一个虚拟的存储设备，数据分散的存储在企业的各个角落。(Hadoop HDFS, Alluxio)
- **并行计算**：基本思想是用多个处理器来协同求解同一问题，可以分为时间上的并行（流水线技术）/空间上的并行（多个处理器并发的执行计算）
- 传统的 Bootstrap 法进行抽样时，传统的自助法中每一组经验样本平均包含原样本中 63% 的样本单元，属于同一计算数量级，因此不能有效降低计算复杂度
  - n 个样本单元有放回抽取，某个样本单元没被抽到的概率是 $(1-\frac{1}{n})^n \to \frac{1}{e}=37\%\ (n\to\infty)$

4. 大规模数据抽样方法：BLB 和 SDB 法能够通过**针对子样本的自助法**实现**针对数据变异性的调整**，在降低计算成本的同时**实现数据变异性的还原**，能够较好地度量出估计的不确定性。

- 小自助包算法（Bag of Little Bootstrap）：记样本为 $\{\boldsymbol{X_1}, \boldsymbol{X_2}, ..., \boldsymbol{X_n}\}$
  - 1. 从样本量为 n 的总样本中，随机**无放回**的抽样 s 个大小为 b 的样本（$b<n$）
  - 2. 在子样本中通过自助法有放回的抽取 r 个大小为 n 的蒙特卡洛样本以模拟数据的变异性，每个蒙特卡洛样本实际上是 b 个不同样本单元的加权组合，因此其计算复杂度为 $O(b)$
    （相当于**从大小为 n 的样本**抽到**大小为 b 样本**又抽回**大小为 n 的样本**，但效率从 $O(n)$ 提升到 $O(b)$）

Algorithm 2: BLB 算法

Input: 数据集 $\mathcal{X}_n=\{\boldsymbol{X}_1,\boldsymbol{X}_2,\ldots,\boldsymbol{X}_n\}$；子样本集合大小 b；子样本数量 s；蒙特卡洛迭代次数 r；评估函数 $\xi(\cdot)$。
Output: 统计量的估计 $\hat{\theta}$，不确定性度量 $\xi(\hat{\theta})$。
for $j \leftarrow 1$ *to* s **do**
  从数据集 $\mathcal{X}_n$ 中随机无放回地抽取一个大小为 b 的子样本 $\mathcal{X}_{j,b}$
  for $k \leftarrow 1$ *to* r **do**
    从子样本 $\mathcal{X}_{j,b}$ 中应用自助法有放回地抽取样本量为 n 的蒙特卡洛样本 $\mathcal{X}_{j,b}^k$
    计算样本 $\mathcal{X}_{j,b}^k$ 的参数估计 $\hat{\theta}_{j,b}^k$
  end
  $\hat{\theta}_{j,b} \leftarrow r^{-1}\sum_{k=1}^{r}\hat{\theta}_{j,b}^k$
  $\xi_j \leftarrow \xi\left(\hat{\theta}_{j,b}\right)$
end
$\hat{\theta} \leftarrow s^{-1}\sum_{j=1}^{s}\hat{\theta}_{j,b}$
$\xi(\hat{\theta}) \leftarrow s^{-1}\sum_{j=1}^{s}\xi_j$



[注]：相比于传统抽样方法，当子样本量 b 远小于总样本量 n 时，BLB 算法的计算效率会有明显的提升。

- 子集双重自助算法（Subsampled Double Bootstrap）：在 BLB 的基础上提出 SDB

Algorithm 3: 针对样本独立数据集的 SDB 算法

Input: 数据集 $\mathcal{X}_n = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n\}$；子样本集合大小 b；子样本数量 s；评估函数 $\xi(\cdot)$；根函数 $T_n(\hat{\theta}_n, \theta)$。

Output: 统计量的估计 $\hat{\theta}$，不确定性度量 $\xi(\hat{\theta})$。

for $j \leftarrow 1$ *to* s **do**

(1) 从数据集 $\mathcal{X}_n$ 中无放回地抽取一个大小为 b 的子样本 $\mathcal{X}_{n,b}^{*j}$

(2) 由 $\mathcal{X}_{n,b}^{*j}$ 计算 $\hat{\theta}(\mathcal{X}_{n,b}^{*j})$

(3) 由 $\mathcal{X}_{n,b}^{*j}$ 生成大小为 n 的子样本 $\left(\mathcal{X}_{n,b}^{*j}, \mathcal{W}_{n,b}^{*(j,1)}\right)$

(4) 由子样本 $\left(\mathcal{X}_{n,b}^{*j}, \mathcal{W}_{n,b}^{*(j,1)}\right)$ 计算估计值 $\hat{\theta}\left(\mathcal{X}_{n,b}^{*j}, \mathcal{W}_{n,b}^{*(j,1)}\right)$

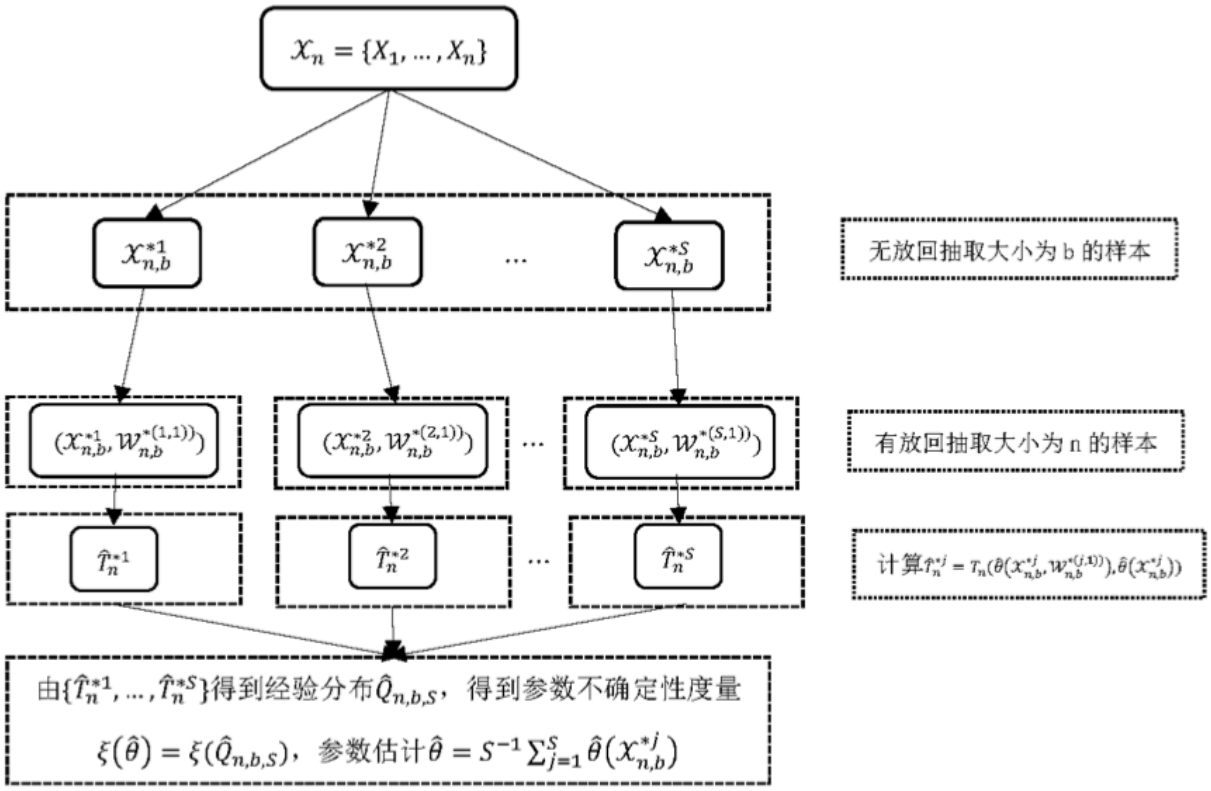
(5) 计算 $T_n^{*j} = T_n\left(\hat{\theta}\left(\mathcal{X}_{n,b}^{*j}, \mathcal{W}_{n,b}^{*(j,1)}\right), \hat{\theta}\left(\mathcal{X}_{n,b}^{*j}\right)\right)$

end

由 $\left\{T_n^{*1}, \ldots, T_n^{*S}\right\}$ 得到经验分布 $\hat{Q}_{n,b,S}$，得到参数不确定性度量 $\xi(\hat{Q}_{n,b,s})$

$\hat{\theta} \leftarrow s^{-1}\sum_{j=1}^{s}\hat{\theta}(\mathcal{X}_{n,b}^{*j})$

$\xi(\hat{\theta}) \leftarrow \xi(\hat{Q}_{n,b,s})$



- 采用 SDB 算法进行抽样估计时只需要进行一层的抽样过程，抽样的总次数为 2s，针对每个子样本进行估计的计算成本为 $2 \times O(b)$，总的计算成本为 $2s \times O(b)$。
- 相比于 BLB 算法，SDB 算法的计算效率明显更高。当给定计算成本时，SDB 算法会把运算资源放在更多的子样本个数 s 上。
- 也就是说，SDB 算法在给定计算成本时通过覆盖更多的总样本单元来提升估计不确定性的精度，且这种优势随着总样本量 n 趋于无穷而逐渐明显。

5. 高维数据的处理方法：降维方法：

- **主成分分析**：构造原始变量的线性组合，形成低维的变量，并使低维变量最大程度地保持原始数据的方差信息

Algorithm 4: PCA 算法

Input: 数据矩阵 X，降维后样本维数 l。
Output: 转换矩阵 $W = (\omega_1, \omega_2, \dots, \omega_l)$。
1：对于 X 中的每一个样本 $\mathbf{x}_i$ 进行中心化处理:

$$\mathbf{x}_i \leftarrow \mathbf{x}_i - \mathbf{m},$$

其中 $\mathbf{m} = \frac{1}{n}\sum_{j=1}^{n}\mathbf{x}_j$ 为样本均值;
2：计算协方差矩阵 $\Sigma = \frac{1}{n-1}\mathbf{X}^T\mathbf{X}$;
3：对协方差矩阵 Σ 做特征值分解并将特征值降序排序：$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$;
4：取最大的前 l 个特征值对应的特征向量 $\omega_1, \omega_2, \dots, \omega_l$ 组成转换矩阵 W。

- **线性判别分析**：利用样本的类别信息找到数据的线性低维表示，使得低维表示最有利于对数据进行分类。

– **基本思路**：通过线性投影将数据降到一维，使得在一维空间中能将数据进行分类

Algorithm 5: LDA 算法

Input: 数据集 $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$，降维后样本维数为 l。
Output: 转换矩阵 $W = (\omega_1, \omega_2, \ldots, \omega_l)$。
1：计算数据集的均值 $\mathbf{m}$ 和每一类数据的均值 $\mathbf{m}_c$：

$$\mathbf{m} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i, \quad \mathbf{m}_c = \frac{1}{n_c}\sum_{i=1}^{n_c}\mathbf{x}_i;$$

2：计算类内离散度矩阵 $\mathbf{S}_\omega = \sum_{c=1}^{C}\frac{n_c}{n}\mathbf{S}_c$，其中 $\mathbf{S}_c$ 为第 c 类样本的类内离散度矩阵；
3：计算类间离散度矩阵 $\mathbf{S}_b$：

$$\mathbf{S}_b = \sum_{c=1}^{C}\frac{n_c}{n}(\mathbf{m}_c - \mathbf{m})(\mathbf{m}_c - \mathbf{m})^T;$$

4：计算矩阵 $\mathbf{S}_\omega^{-1}\mathbf{S}_b$，并对其做特征值分解，将矩阵 $\mathbf{S}_\omega^{-1}\mathbf{S}_b$ 的特征值降序排序；
5：选取前 l 个特征值对应的特征向量 $(\omega_1, \omega_2, \ldots, \omega_l)$ 按列组合成转化矩阵 W。

6. 高维数据的处理方法：变量选择方法

- **最优子集选择法**
  - **基本思想:** 穷举所有可能的变量集合来构造备选模型，利用 AIC，BIC 或者 Mallows C_p 准等来评价模型的好坏，挑选出最佳的模型 (变量集合)。
  - 理论上这是最好的方法，但是当 p 较大时，计算成本巨大。
- **正则化方法**
  - **基本思想:** 在损失函数中加入惩罚项，控制模型的复杂程度，同时实现模型中变量的估计与选择。
  - Lasso 求解以下优化问题

$$\min_{\omega}\frac{1}{2}||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\omega}||_2^2 + \lambda||\boldsymbol{\omega}||_1$$

其中正则化系数 $\lambda > 0$ 是依赖于 C 的调节参数

7. 超高维数据的处理方法

- 首先利用变量筛选 (Screening)，将超高维数据集转化为高维数据集
- 筛选方法主要考察备选变量与响应变量之间的相关性：相关性强的留下，相关性弱的筛走
- 然后再利用已有的正则化方法对高维数据进行处理

4 可视化方法

数据可视化是将数据用图形等视觉效果展现出来的过程与方式。

4.1 基础统计图形

1. 图形设备：指软件系统

- 在图形输出方面，最常见的文件格式分别是位图和矢量图。
- 位图也称为点阵图，简单来说就是由像素点组成的图。对于位图中的每个像素点，可以使用一套色彩模式来描述其颜色，最常用的是 RGB 模式。
- 矢量图使用了另外一套图形描述机制，通过曲线和角度来存储形状特征，无须通过像素。因此，矢量图无论如何放大都不会损失清晰度。
- 在 R 和 Python 中，默认的交互界面自带了图形设备，可以通过执行基础绘图命令来自动调出。

2.ggplot 绘图语言：Leland Wilkinson 创造图形编程语法，Hadley Wickham 基于 R 语言创造了 ggplot2

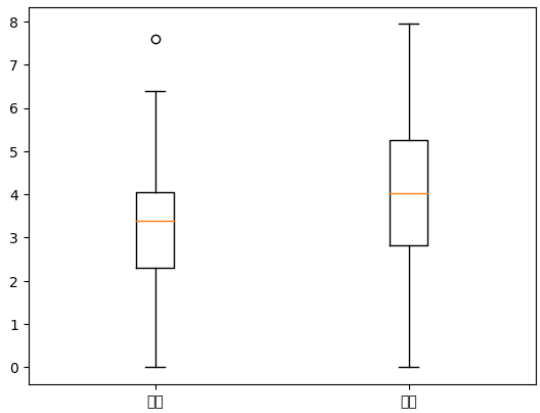
4.2 可视化与数据分析

1. 单变量的分布

- 直方图 (Histogram)：直方图是一种常用的统计图形，它能够帮助研究者**查看连续变量的分布情况**，通过将数据分割后再统计每一组的数目，**得到类似于分布密度图的统计图形**，从而查看连续数据的分布特征。
- 条形图 (Bar chart)：条形图可以非常方便地展示**离散变量的各个类别及其数目**
- 饼图 (Pie chart)：饼图在**表达比例关系**的时候很常用。当类别数目不太多的时候，可以**一目了然地看清不同类别的数值和比例大小**，是最为公众熟知的统计图形之一。但由于人类视觉对角度不太敏感，用饼图来展示比例不如条形图清晰，尤其是当类别过多时，应该尽量避免饼图。

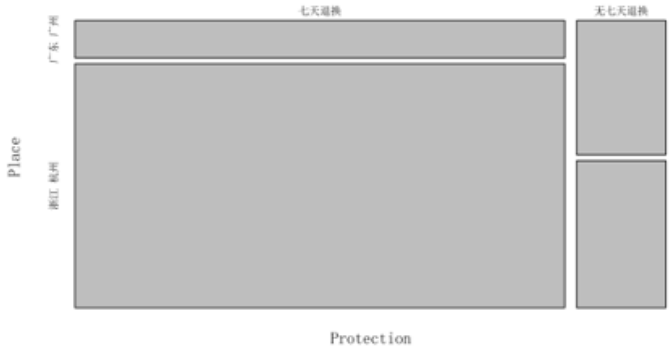
2. 两变量的关系：在数据分析中，研究者处理的变量分为连续变量和离散变量。**两变量间可以构成三种关系：（1）两个连续变量的关系；（2）连续变量和离散变量的关系；（3）两个离散变量的关系**

- **两个连续变量的关系**：可以使用相关分析、回归分析等模型进行深入研究，也可以通过**散点图**来进行直观展现。
- **连续变量和离散变量的关系**：
  - 当离散变量是因变量的时候，**对应机器学习中的分类模型**
  - 当连续变量是因变量的时候，相当于分析不同水平对因变量的影响，**对应统计中的方差分析，也可以使用箱线图来直观展示**。



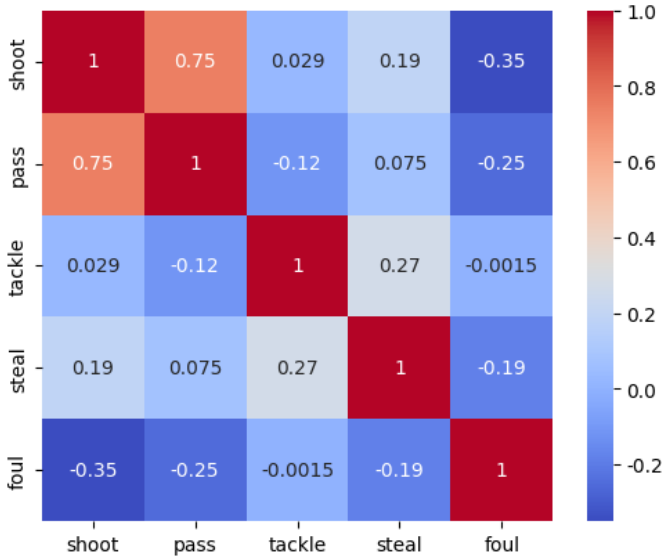
箱子体现了五个关键的值，**上下的两根横线表示上下界，箱子上下边缘表示上下四分位数，中间的粗线表示中位数**。

- **两个离散变量的关系**：
  - 由于数据中离散变量交叉组合后会产生不同组别的频数，可使用列联表进行分析，例如 χ^2 检验。
  - **马赛克**图也可以直观地展示这种关系：
    * 每一块面积代表一个交叉组的频数，面积越大说明该组在总体的频率越高
    * 直观查看各组是否均匀/是否存在明显差异



3. 多变量的关系

- 热力图：通过**相关系数矩阵**图查看两两变量间的关系，考察他们的正相关/负相关性。(比如 python 中调用 cmap = “coolwarm”，越红越正相关，越蓝越负相关)



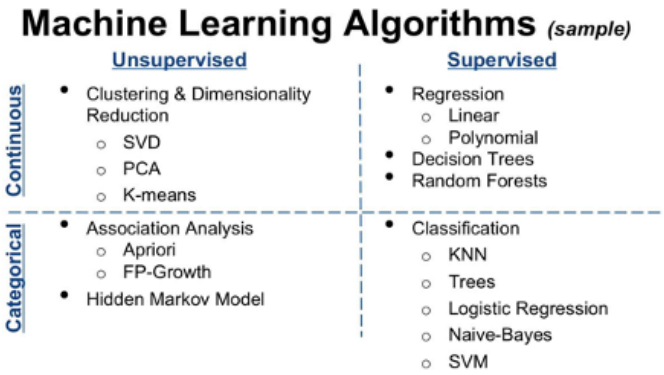
4.3 现代数据可视化方法

1.Buffon 投针实验：在平面上画有一组间距为 D 的平行线，将一根长度为 $L(L \leq a)$ 的针任意投掷在这个平面上，可以证明此针与平行线中任一条相交的概率 p 为

$$p = \frac{2L}{\pi a}$$

如果研究者做一个试验，将针随机投掷一定的次数，然后统计其和平行线相交的次数并基于频率估计概率 p 可以估算出 π 的值。
2. 交互式工具：ECharts, R Shiny 等

5 数据挖掘和机器学习



数据挖掘与机器学习：差异之处

- 机器学习偏重于方法，数据挖掘偏重于流程。
- 通常在业界的应用中不去区分这两个概念，可以简单地认为“使机器学习方法、遵循数据挖掘流程”来进行数据分析。

5.1 从海量数据到大数据

1. 常见的数据挖掘技术：

- 分类；回归；异常检测；聚类；关联规则；序列挖掘

2. 数据挖掘步骤

- 问题理解：清晰的定义问题，站在业务角度考虑，评估可行性
- 数据理解：理解数据业务意义，探索数据，匹配现实世界中事物的特征
- 数据准备：对数据进行整理和转换，数据清洗/转换/特征选择
- 数据建模：分析建模过程，选择模型评价方式（损失函数）
- 模型评估：从技术、业务的角度进行评估，普适性、有用性、可解释性、新颖性
- 模型部署：部署到实际的应用环境中

2. 机器学习：
如果一个计算机程序针对某类任务 T 的用 P 衡量的性能根据经验 E 来自我完善，那么我们称这个计算机程序在从经验 E 中学习，针对某类任务 T，它的性能用 P 来衡量。

- 任务 T 的种类包括分类、回归、聚类、异常检测等。

- 性能的定量度量标准 P，特定于系统执行的任务 T 而言。例如分类任务，可以使用“准确率”(Accuracy) 进行度量。
- 经验 E 通常是指从数据集中获取的经验，根据学习过程中的不同经验，机器学习算法可以大致分类为无监督 (Unsupervised) 学习和有监督 (Supervised) 学习。

[注 1]：确保经验科学性的关键在于可度量。
[注 2]：这套分析的思想也称为数据驱动。
[注 3]：与之相对的是基于演绎推理的可以不依赖数据和学习的专家系统。

4. 特征学习与非特征学习：ML 包含了特征学习和非特征学习

- 很多统计模型 (如回归分析) 都属于非特征学习，需要研究者筛选并指定特征后建立模型;
- 特征学习的方法可以自动学习特征并进行筛选，研究者只需将所有的特征输入即可。
  - 在特征学习中又包含深度学习和浅度学习，当前人工智能的主流技术属于深度学习，基于多层的神经网络来实现。

5. 无监督学习与有监督学习

- 不使用数据集中的标签信息的方法是无监督方法，是纯粹基于数据学习出有用的结构性质，比如聚类分析。
- 而有监督学习利用数据集的标签特征进行学习，比如回归、分类等任务。
- 还有一些方法可以充分利用未标记样本，做一些将未标记样本所揭示的数据分布信息与类别标记相联系的假设来辅助提升学习效果，称为半监督 (Semi-supervised) 学习。

5.2 无监督学习

无监督学习：数据样本中不包含标签信息，因此无需学习现成的模式，而是**从数据自身的规律入手探寻内在的结构**。

5.2.1 主成分分析 (PCA)

当数据集的变量数很多的时候，使用该方法找到前几位的主成分，**可以用来解释大量的变量，实现降维并更好地了解数据中的规律**。

- **悬崖碎石图：展示了每个主成分的方差占比**，“悬崖”越陡峭，说明前面的主成分占比越多，也就意味着解释性越好。

5.2.2 聚类分析

针对变量的聚类称为 R 型聚类，针对样本的聚类称为 Q 型聚类。

1. 层次聚类：将不同距离程度的样本根据层次划分为树状结构

- **描述样本之间的差别**：把样本当成空间中的点，用点的距离来描述样本之间的差别：
  - Euclid 距离/Manhattan 距离/Minkowski 距离/余弦距离
- 从点的距离入手，遍历所有点，**将距离最近的点与点、点与类、类与类连接在一起，形成树状的层次结构**，如此反复迭代，直到汇集到一个类。

– 计算类与类/点与类之间距离：类平均法/最短距离法/最长距离法

2. 原型聚类：聚类结果可以使用一组原型来描述，并通过对原型不断迭代来求解。例如 K-means：

- 制定最终类的个数 K
- 随机选 K 个点作为初始的类中心点，计算各样本点与类中心点的距离，距哪个类中心点更近就归入哪一类
- 所有样本归类完成后，将每一类中所有点的均值作为该类的新中心点
- 重复迭代这个过程直到类中心不再变化

Algorithm 1: K-Means 算法

Input: 数据集 $\mathcal{X}_n = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n\}$；类别数 k；最大迭代次数 $iter.max$。

Output: 类别的划分 $C = \{C_1, C_2, \ldots, C_k\}$；均值向量（中心点）$\boldsymbol{\mu} = \{\mu_1, \mu_2, \ldots, \mu_k\}$。

for $j \leftarrow 1$ *to* $iter.max$ **do**

(1) 从数据集 $\mathcal{X}_n$ 中随机选择 k 个样本作为初始均值向量 $\{\mu_1, \mu_2, \ldots, \mu_k\}$

(2) 计算各样本点 $\boldsymbol{X}_m (1 \leq m \leq n)$ 与各均值向量 $\mu_i (1 \leq i \leq k)$ 的距离，如果 $\boldsymbol{X}_m$ 到 μ_{λ_m} 的距离最小，则将其归入 C_{λ_m} 类。

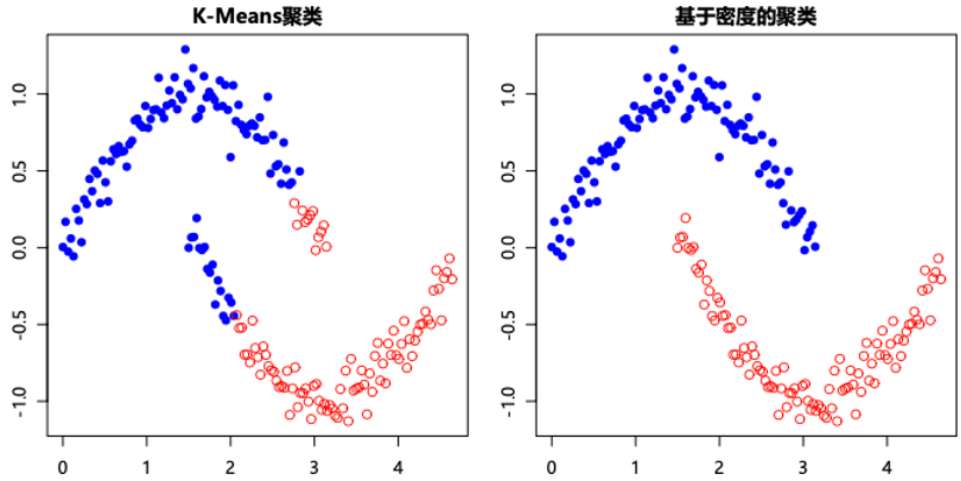
(3) 对于每个类别 C_l，将其中所有点的坐标取均值，得到新的均值向量 μ_l'

(4) 如果 μ_l' 相对于 μ_l 不再更新，终止程序，输出 C 和 $\boldsymbol{\mu}$，否则将当前 μ_l 更新成 μ_l'

end

- 最后根据聚类结果“打标签”

3. 密度聚类：K-Means 聚类这样的基于原型的方法通常假设每个类都是簇状的，因此可以通过每个点到中心点的距离来聚类。但有时数据之间的集中规律并不是簇状的，如下左图。用 K-Means 聚类后，会发现有一些边缘的点被聚到了错误的类中。**当类别结构无法使用原型来描述时，可以用样本的紧密程度来进行描述。最常见的是 DBSCAN 算法，聚类的结果如下图所示。**



流程如下：

- 在 DBSCAN 算法中，首先将数据样本点分成以下三类:

  - 核心点: 如果某个点的邻域内的点的个数超过某个阈值，则它是一个核心点，即表示它位于簇的内部
  - 边界点: 如果某个点不是核心点，但它落在核心点的邻域内，则它是边界点
  - 噪声点: 非核心点也非边界点

- 算法运行时，会基于定义将所有点标记为核心点、边界点或噪声点，并**将任意两个距离小于邻域半径的核心点归为同一个簇。任何与核心点足够近的边界点也放到与之相同的簇中**，从而得到聚类结果。

5.3 有监督学习

5.3.1 回归分析

1. 多元线性回归的方程为：

$$Y = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p + \varepsilon$$

且为保证回归模型参数 $\hat{\boldsymbol{\beta}}$ 的性质，假设：

- 自变量 X 非随机，因变量 Y 随机
- Gauss-Markov 条件：每个随机误差项之间不相关，随机误差项的期望为 0，方差都相等
  - 进而：$E(Y) = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$
- 正态性：$\varepsilon \sim N(\lambda, \sigma^2)$
- 满秩性：样本数量 > 变量数量

对于回归模型 $\boldsymbol{Y = X\beta}$，用最小二乘法求解：

$$\boldsymbol{X^T Y = X^T X\beta}$$
$$\Rightarrow \hat{\boldsymbol{\beta}} = \boldsymbol{(X^T X)^{-1} X^T Y}$$

对于一个完整的回归分析来说，研究者除了参数估计外，还需要对模型进行显著性检验 (F 检验)、对参数进行显著性检验 (t 检验)，并诊断是否存在多重共线性、序列自相关性等。

5.3.2 分类问题与分类性能评估

1. 分类性能评估：以糖尿病为例：
定义预测结果：（1）真阳性 (TP): 把阳性样本正确地分类成阳性。
（2）真阴性 (TN): 把阴性样本正确地分类成阴性。
（3）假阳性 (FP): 把阴性样本错误地分类成阳性。
（4）假阴性 (FN): 把阳性样本错误地分类成阴性。

- **混淆矩阵**：真实值和预测值之间数目的列联表对应的一个矩阵

针对以上 4 种预测结果的各自数目，可以构造出一系列评价指标来判断分类结果的好坏。

- **准确度 (Accurary)**：表示真阳性和真阴性的数目除以所有预测值的个数，计算公式为：(真阳性 + 真阴性)/总数。
- **错误率 (Error Rate)**：表示不正确分类的比例，等于 1− 准确度。

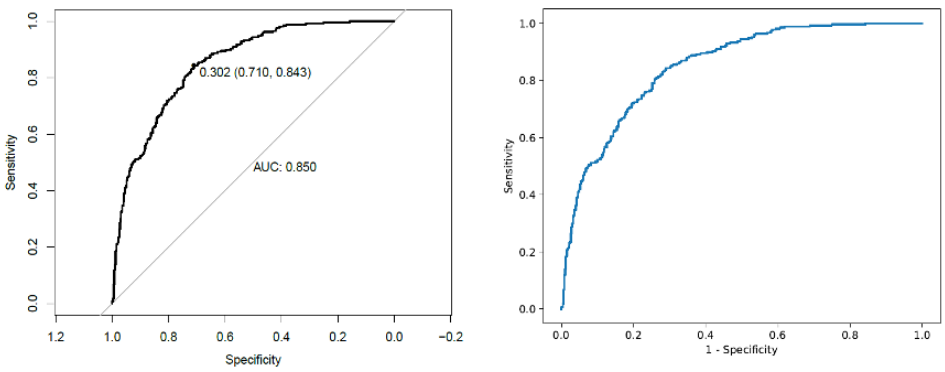
- **精度 (Precision)**：也称为查准率，表示真阳性在所有预测为阳性例子中的比例，计算公式为：真阳性/(真阳性 + 假阳性)。
- **召回 (Recall)**：也称为查全率，表示真阳性与阳性总数的比例，计算公式为：真阳性/(真阳性 + 假阴性)
- **灵敏度 (Sensitivity)**：也称为真阳性比率，度量了阳性样本被正确分类的比例，和召回的含义相同，计算公式为：真阳性/(真阳性 + 假阴性)。
- **特异性 (Specificity)**：也称为真阴性比率，度量了阴性样本被正确分类的比例，计算公式为：真阴性/(真阴性 + 假阳性)。
- **Kappa 统计量**：用来描述预测值和真实值之间一致的概率，可以消除一些因为完全偶然猜对的影响。0.6 以上表示效果不错，0.8 以上表示效果很好。
- **F1 分数**：可以看作是模型精度和回的一种调和平均，它的最大值是 1，最小值是 0，其值越高越好。

在这些指标中

- **准确度 (或者错误率) 最常用**，但无法衡量不同错误类型的代价，因此通常与精度和召回、灵敏度和特异性这两对指标结合使用。
- **精度和召回在搜索领域使用较广泛**。如在网络上搜索某个问题，精度高说明在搜索出来的结果中我们真正想要的内容的比例很高；而召回强调的是搜索出来的内容中要尽可能全地覆盖想要的内容。
- 灵敏度和特异性在医疗领域使用较广泛。灵敏度实际上就是召回，代表不放过任何疾病，比较适合用来筛查。

针对模型评估，通常使用两类误差来评估

- 训练误差：指模型在训练集上表现出的误差。**主流的模式是先计算混淆矩阵，然后观察各类评价指标**。
  - 在具体的分析问题中，通常希望**灵敏度与特异性** (或者精度与召回) **能同时达到高位**。但在具体的模型中，这一对指标往往是此消彼长的。
  - ROC 曲线：通过调节阈值的方式得到的**一条关于灵敏度和特异性的曲线**。每个阈值对应曲线上的一个点，研究者通常选取使得该点下方矩形面积最大的点作为最优阈值。
  - AUC：ROC 曲线下的面积



  - 当某个模型的 AUC 很大，说明分类效果很好。当 ROC 曲线接近 45 度对角线，说明该模型分类效果很差。
  - 如果模型 A 的 ROC 曲线完全在模型 B 的外侧，说明**相同的灵敏度下模型 A 的特异性更好、相同特异性下模型 A 的灵敏度**更好，因此具有全方位的优势。
- 泛化误差：指模型在任意一个测试数据样本上表现出的误差的期望。

5.3.3 线性判别分析（LDA）

通过线性投影将数据降到一维，使得在一维空间中也能够很好地将数据进行分类。

5.3.4 Logistic 回归:

一般线性回归 (General Linear Regression) 要求**因变量 y 是连续变量**且误差项要服从正态分布。**如果因变量 y 是分类变量，线性回归就不再适用了。对于二分类变量的问题，**研究者可以使用 Logistic 回归模型来分析处理。

- 自变量：各影响因素的线性组合：$\beta_0 + \beta_1 x$
- 因变量：某事件发生的概率
- 连接函数：对自变量线性组合施加一个函数变换，使得值域范围在 $[0, 1]$：

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{1}$$

  - 其中 $f(x) = \dfrac{1}{1 + e^{-x}}$ 是 Logistic 函数，进而这种形式的回归被称为 Logistic 回归

- 对（1）作对数变换：

$$\ln \frac{p}{1 - p} = \beta_0 + \beta_1 x$$

  经过对数变换后转换为求解线性回归问题

  - 其中 $\dfrac{p}{1 - p}$ 成为事件发生比/优势 (Odds)
  - 回归系数可以解释为对数优势之比 (Odds Ratio) 的贡献

Logistic 回归的优点在于

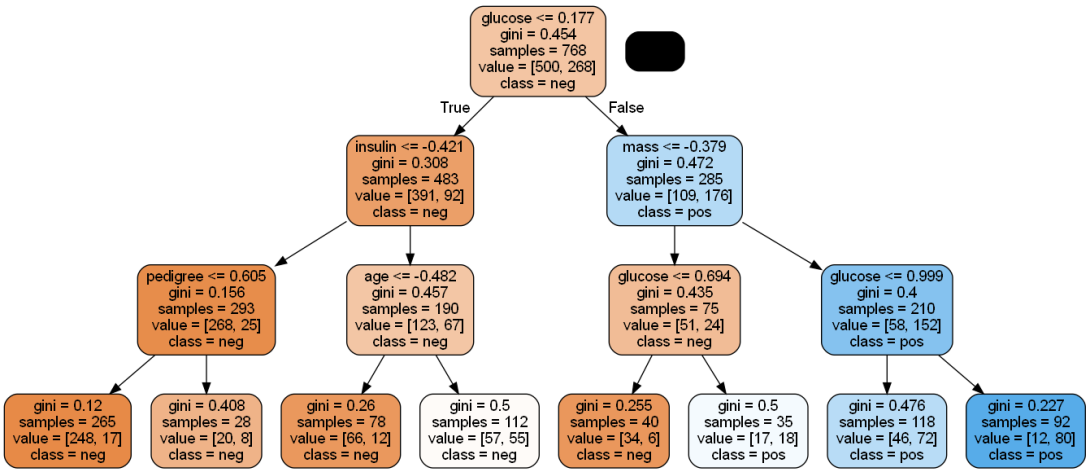
- Logistic 回归的计算性能很好，在业界使用得非常广泛
- 与其他机器学习算法不同的地方在于它的统计性能和可解释性也非常好
- 模型的自变量之间假设了线性关系，因此**回归系数可以理解成权数**，应用到具体的业务中时非常方便。

5.3.5 决策树:

决策树是一种常用的分类模型，用树结构的方式来描述一个分类的过程，树存在很多分支的节点，每个节点都是一个逻辑判断。**逐级判断的专家系统和决策树的逻辑存在本质的不同**：决策树的关键在于**基于训练集数据学习划分规则，递归地建立树状模型**，这种思路也是机器学习的特点所在。

决策树模型的算法中最关键的步骤是选择最优的划分属性，常用的方法有“信息增益”、“增益率”、“基尼指数”等。

- Breiman 提出的 CART 是最常用的决策树模型之一，**使用基尼指数**来选择划分属性
- Quinlan 基于**信息增益准则**提出了 ID3 算法，掀起了决策树研究的热潮
- 其他研究者很快创造了 ID4、ID5 等算法
- Quinlan 将后续自己的算法命名为 C4.0，之后的修改版 C4.5 成为最热门的算法之一，后续的商业化版本称为 C5.0

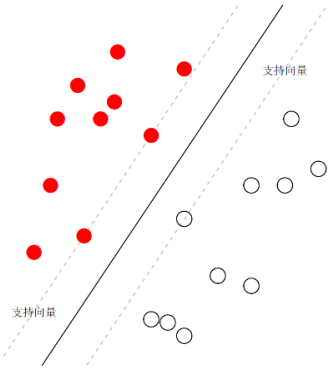


5.3.6 随机森林：

决策树的思路直观且算法简约，但学习过程不稳定，容易受随机误差影响等。如果需要对一个问题进行决策，一种有效的方法是采用“少数服从多数”的原则通过投票进行判断。

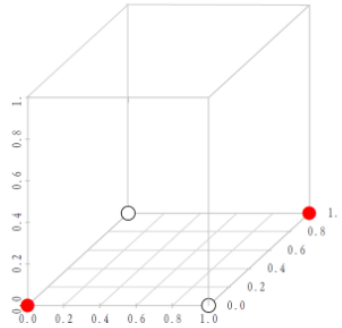
- 在机器学习中借鉴这一思路将多个学习器 (模型) 整合起来构成一个混合模型，称为集成学习 (Ensemble Learning)
- Breiman(2001) 提出的随机森林 (Random Forest) 就是集成学习的典型代表
- 随机森林是包含多个决策树的分类器，其输出的类别由个别树输出的类别的众数而定。模型中对 N 个样本有放回地随机抽取 N 个、对 M 个特征随机采样 m 个 ($m << M$)。对每一种情况都建立一个决策树模型。综合多个决策树，以票数多的结果为准。
- 随机森林算法的关键是**如何抽取使得构造出的决策树之间的相关性尽量小**
- **缺点是运算速度稍慢**，因为一座森林包含了很多棵树，相当于运行了大量的决策树算法，比较耗资源。**工程上可以使用并行的方式对算法进行处理**，可减少运算时间
- 随机森林算法还可以用来解决 Bagging 方法中各个模型之间相关性的影响 (因为每一个 Bootstrap 样本上的估计量相关性太强，方差为 $\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$，$B\uparrow$ 但第一项始终存在)：随机森林可以在树生长的过程中通过**随机选取输入变量的方式**降低 Bagging 估计量之间的相关性，而不会给方差带来很大的增长

5.3.7 支持向量机

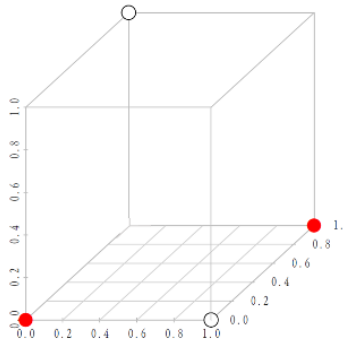


假设研究者可以在两类点间插入一块**可以旋转并改变宽度**的“木板”，当“木板”边缘遇到点时就会卡住。当**“木板”处于宽度最大的角度时**，可以认为其分类效果最好。

- 这个“木板”的边缘也称为**支持向量**
- 基于这种思路的分类方法也称为支持向量机 (Support Vector Machine)。
- **线性不可分问题**：假设用支持向量机将其分类，无论如何是没有办法插入一块“木板”的



- **通过升维/投影**让点变得稀疏且容易插入“木板”超平面，从而实现分类。在 SVM 中，升维的目的是便于分类，并不需要真的把数据转化成高维空间中的具体坐标。**在 SVM 方法里可以用核函数解决空间映射**的问题。



6 人工智能

6.1 人工智能简史

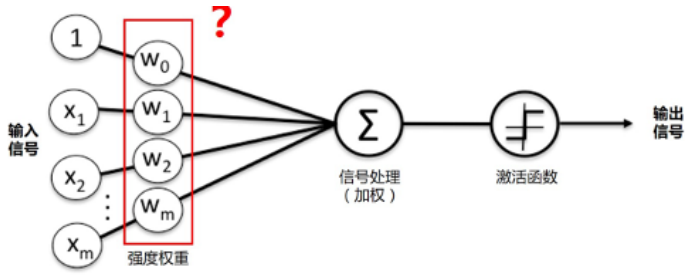
1. 派别：符号主义（人工智能源于数理逻辑）；联结主义（人工智能源于仿生学）；行为主义（人工智能源于控制论）
2. 时间线：1943 提出神经元数学模型，1956 提出 AI 概念，1957 感知机（联结主义），1960 末认为感知机存在局限，1970s 强调数理逻辑和演绎思维，1980s 专家系统，1990s 数据挖掘 + 神经网络 BP 算法，1998 SVM 热潮，2004$^{\text{th}}$ 深度学习，2007 NVIDIA CUDA GPU，2009 吴恩达 GPU 深度学习，2012 深度学习进步，2015AlphaGo

6.2 神经网络

1. 模仿神经元细胞构造一个数学模型：步骤如下

- 假设该神经元细胞包含 m 个树突，每个树突 i 接收一个信号源，对应一个输入变量 x_i

- 细胞核接收到这 m 个信号后需要汇总处理，比如使用最简单的形式——线性加权：$w_0 + \sum_{i=1}^{m} w_i x_i$
- 汇总信号只有超过某个阈值后才会被传递出去，在数学上，**研究者可以用激活函数来处理**
- **处理后的信息可以作为输入再传到另一个神经元结构，这样就可以构成复杂的神经网络。**



6.2.1 感知机

只使用单个的神经元模型也称为感知机 (Perceptron)，是形式最简单的神经网络。
对于感知机的学习：输入用公式来描述就是 $y = w_0 + w_1x_1 + w_2x_2 + ... + w_mx_m$。**感知机学习算法的主要步骤**如下：

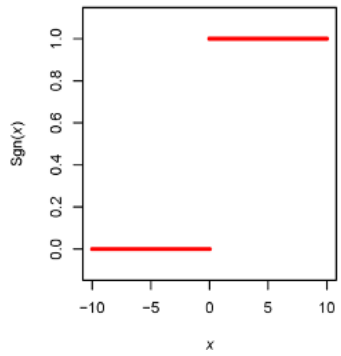
- 随机指定初始权重 (比如 $w_0 = 0, w_1 = 0$,$w_2 = 0$)，然后设定学习率 η 的值 (比如 $\eta = 0.1$，学习率设置得过大，有可能会越过最优值，如果设置得太小，又会迭代次数增加，影响计算性能。)
- 将初始权重代入 $y = w_0 + w_1x_1 + w_2x_2 + ... + w_mx_m$ 后可以计算当前输出 $\hat{y}$，然后对每一个样本点 (xi)，根据 $\hat{y}$ 与 y 的差异不断地调节权重值:

$$\begin{cases} w_i \leftarrow w_i + \Delta w_i \\ \Delta w_i = \eta(y - \hat{y})x_i \end{cases}$$

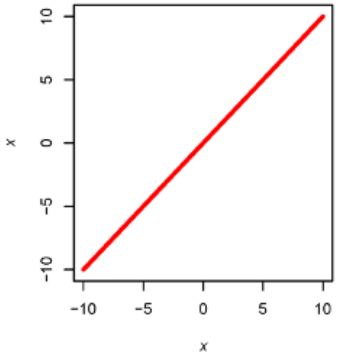
- 经过多轮迭代后，直到 $\hat{y} = y$，则感知机不再发生变化，输出最终的权重结果

2. 激活函数的选择

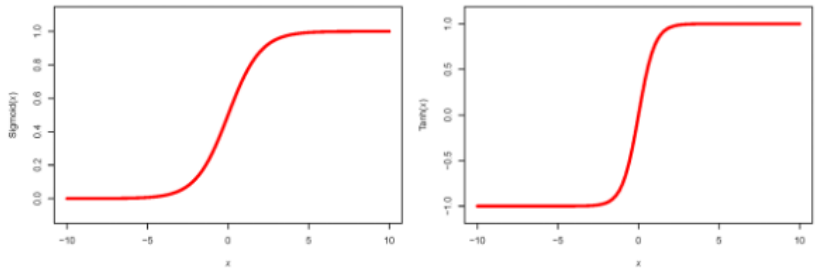
- **分段函数：**当信号大于某个阈值的时候输出 1，反之输出 0，如下图所示。该函数不连续、不光滑，数学处理上不是很方便，所以很少使用



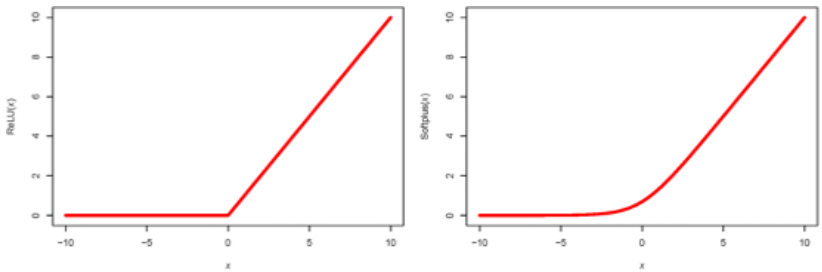
- **线性函数：**$f(x) = x$。虽然其数学性质很好，但是因为线性函数的线性组合还是线性函数，无论神经网络有多少层都仍然是一个线性组合，难以处理复杂的非线性问题



- **Sigmoid 函数和 Tanh 函数**：对数 S 型函数 $\text{Sigmoid}(x) = \dfrac{1}{1+e^{-x}}$ 和双曲正切函数 $\tanh(x) = \dfrac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$，但两者缺陷在于：当输入的绝对值非常大的时候会出现饱和 (Saturate) 现象，函数会变得变平，对输入的微小改变会变得不敏感



- **ReLU 函数和 Softplus 函数**：ReLU 更容易学习和优化，于是成了目前最广泛使用的激活函数，Softplus 是



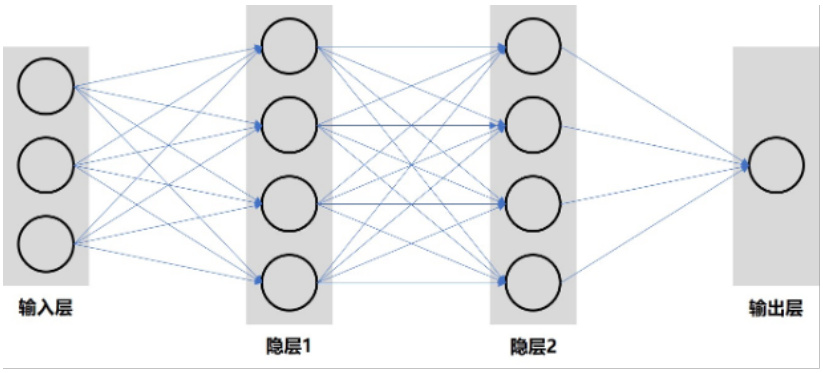
6.2.2 神经网络

感知机这种简单模型，也可以解决一些分类问题。这种基于迭代的学习思路是神经网络求解的基本思路。但感知机的学习算法无法解决线性不可分的问题，在实际的工作中，人们需要更复杂的神经网络模型来处理非线性的情况。

有了神经元细胞核的线性汇总和激活函数后，把多个人工神经元按照一定的层次结构连接起来，就得到了人工神经网络 (Artificial Neural Network，ANN)。

最简单的神经网络系统是前馈神经网络 (Feedforward Neural Network，FNN)：各神经元只接受前一级的输入，并输出到下一级，只有前馈 (Feedforward) 而无反馈 (Feedback)

除了输入层和输出层之外，**额外的中间层称为隐藏层**。严格来说，FNN 包含单层前馈神经网络和多层前馈神经网络，前者即感知机（只有一个神经元），**后者称为多层感知机 (MLP)**。多层感知机如下图所示：



6.2.3 BP 算法

在 FNN 中，模型接受输入 x 提供的初始信息，再传入神经网络的每一层，最终产生 $\hat{y}$，称为前向传播 (FP)。在这个过程中，误差会反向传播 (BP)。

BP 算法**基于经典的链式求导法则**，计算出每个节点的梯度，**从而可以通过迭代求得各权重的值**，因此称为反向传播算法，简称 BP 算法。其基本思路如算法所示。

Algorithm 1: BP 算法

Input: 数据集 $\{x_k, y_k\}, 1 \le k \le n$；学习率 η；最大迭代次数 $iter.max$。

Output: 神经网络所有连接的权重 w_i。

随机初始化所有权重 w_i

for $j \leftarrow 1$ *to* $iter.max$ **do**

  for $k \leftarrow 1$ *to* n **do**

    (1) 基于前向传播计算当前样本的输出值 $\hat{y}_k$

    (2) 基于反向传播计算各神经元的梯度项（因为链式法则，在算法上可以共享部分计算，从而提升性能）

    (3) 通过学习率更新权重

    (4) 如果权重不再更新，终止程序

  end

end

6.3 深度学习

1. 原理：深度学习就是网络层次更深的神经网络，也可以看作是由许多简单函数复合而成的函数，当这些复合函数足够多时，**深度学习模型就可以表达非常复杂的变换**

深度学习模型的复杂层次导致未知参数数量激增，学习训练过程运算量巨大。幸而构成模型的基本元素通常为简单函数，因此非常适合并行计算，尤其是利用 GPU 进行计算。

2. 常见深度学习框架：Theano（LISA 实验室，基于 Python，基于 GPU 加速），Caffe（Berkeley 贾扬清，基于 C++，工业界），**Torch（NYU，Facebook，PyTorch）**，TensorFlow（Google，用户数最多），**MXNet（DMLC）**

7 附录：代码与运行效果

1. 用 Bootstrap 法实现**样本均值的 Bootstrap 分布**的代码如下：以一个正态分布的样本为例：

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from numpy.random import default_rng

# Generate a sample following normal distribution with mean 0 and std 1
rng = default_rng()
n=10000
normal_dist = rng.normal(0, 1, size = n)

# Generate the population distribution, the std of which = std/sqrt(n)
# This is to serve as a reference against Bootstrap distribution
normal_dist_ref = rng.normal(0, (1/np.sqrt(n)), size = n)

# Use Bootstrap method to estimate the mean of the population
bootstrap_mean1 = []#B=10
bootstrap_mean2 = []#B=20
bootstrap_mean3 = []#B=50
bootstrap_mean4 = []#B=100

def bootstrap_function(bootstrap_mean, B):
for i in range(B):
bootstrap_sample = rng.choice(normal_dist, size=n, replace=True)
bootstrap_mean.append(np.mean(bootstrap_sample))
return bootstrap_mean

bootstrap_mean1 = bootstrap_function(bootstrap_mean1,10)
bootstrap_mean2 = bootstrap_function(bootstrap_mean2,20)
bootstrap_mean3 = bootstrap_function(bootstrap_mean3,50)
bootstrap_mean4 = bootstrap_function(bootstrap_mean4,100)


# Obtain the default palette of Seaborn for line color
palette = sns.color_palette()

# Create a KDE plot of the bootstrap means
plt.figure(figsize=(8, 4))
sns.kdeplot(bootstrap_mean1, color=palette[0], label='B=10')
sns.kdeplot(bootstrap_mean2, color=palette[1], label='B=20')
```

```
sns.kdeplot(bootstrap_mean3, color=palette[2], label='B=50')
sns.kdeplot(bootstrap_mean4, color=palette[3], label='B=100')
sns.kdeplot(normal_dist_ref,color=palette[4], fill=True, label='Real Distribution')
plt.title("the Bootstrap distribution and real distribution of sample mean")
plt.xlabel("y.bar")
plt.ylabel("Density")
plt.legend()
plt.show()
```