计算物理 B

Uphi.

University of Science and Technology of China
yuhongfei@mail.ustc.edu.cn

Saturday 21st June, 2025

目录

1 蒙特卡洛方法 1

1.1 基础概念 . . . 1

1.1.1 投针实验 . . . 1

1.1.2 概率统计基础 . . . 1

1.2 伪随机数 . . . 2

1.3 随机抽样方法 . . . 2

1.3.1 离散随机变量的直接抽样 . . . 2

1.3.2 连续变量的直接抽样 . . . 2

1.3.3 变换抽样法 . . . 2

1.3.4 舍选法 . . . 3

1.3.5 复合抽样法 . . . 3

1.3.6 特殊近似抽样法 . . . 3

1.3.7 高维随机向量的抽样 . . . 3

1.4 减小方差的技巧 . . . 3

1.4.1 分层抽样法 . . . 3

1.4.2 重要抽样法 . . . 3

1.4.3 其他方法 . . . 4

1.5 随机游走 . . . 4

1.5.1 Metropolis 方法 . . . 4

1.6 蒙特卡洛应用 . . . 4

1.6.1 中子运输 . . . 4

1.6.2 定积分求解 . . . 4

1.6.3 统计物理 . . . 4

1.6.4 量子力学 . . . 4

2 有限差分法 5

2.1 基础概念 . . . 5

2.2 常微分方程的求解 . . . 5

2.2.1 欧拉法 . . . 5

2.2.2 梯形法 . . . 5

2.2.3 龙格-库塔法 . . . 6

2.3 偏微分方程的求解 . . . 6

2.4 差分方程组 . . . 7

2.4.1 直接求解 . . . 7

2.4.2 直接迭代 . . . 7
2.4.3 高斯-赛德尔迭代 . . . 7
2.4.4 超松弛迭代 . . . 7
2.4.5 Hockney 直接法 . . . 7

3 有限元素法 9
3.1 基础概念 . . . 9
3.2 有限元素法流程 . . . 9
3.3 有限元与有限差分的比较 . . . 9

4 分子动力学 11
4.1 基础概念 . . . 11
4.1.1 周期性边界条件 . . . 11
4.1.2 分子力场与截断距离 . . . 11
4.1.3 时间步长 . . . 12
4.1.4 约束动力学 . . . 12
4.1.5 SHAKE 算法 . . . 13
4.2 微正则系综和正则系综的模拟流程 . . . 13
4.2.1 Verlet 算法 . . . 13
4.3 分子动力学的优劣 . . . 13
4.4 分子动力学和蒙特卡洛方法的比较 . . . 13
4.5 分子动力学的局限性 . . . 14

5 高性能计算 15
5.1 并行计算与串行计算 . . . 15
5.2 MPI . . . 16
5.3 两种分子动力学并行算法 . . . 16

6 计算机代数 17

7 机器学习 19
7.1 基础概念 . . . 19
7.2 监督学习 . . . 19
7.2.1 线性回归 . . . 19
7.2.2 逻辑回归 . . . 21
7.2.3 决策树 . . . 21
7.2.4 集成学习 . . . 21
7.2.5 神经网络 . . . 22
7.3 非监督学习 . . . 22
7.3.1 聚类 . . . 22
7.3.2 降维 . . . 23

1 蒙特卡洛方法

1.1 基础概念

1.1.1 投针实验

间隔为 d 的平行线，投长度为 l 的针．针的自由度：重心（中心）到最近的平行线的距离 x，与平行线法线的夹角 θ，则当：

$$x \leqslant \frac{l}{2}\cos\theta.$$

时针与平行线相交．对于独立随机变量 x,θ，这个概率可以写为：

$$P = \frac{\displaystyle\int_0^{\frac{\pi}{2}}\int_0^{\frac{l}{2}\cos\theta} \mathrm{d}x\mathrm{d}\theta}{\displaystyle\int_0^{\frac{\pi}{2}}\int_0^{\frac{d}{2}} \mathrm{d}x\mathrm{d}\theta} = \frac{2l}{\pi d}.$$

这样一来在概率中就出现了 π.

1.1.2 概率统计基础

随机变量：关联性、期望、方差、线性组合

概率密度函数、概率分布函数及其性质

常用随机分布：均匀分布、指数分布、伯努利分布、二项分布、泊松分布

大数法则 进行足够多的随机试验，其随机变量的均值收敛于其期望值.

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n} f(x_i) = \frac{1}{b-a}\int_a^b f(x)\mathrm{d}x.$$

可以用于求解积分.

中心极限定理 无论随机变量分布如何，它的若干个独立随机变量抽样值之和总是满足高斯分布. 设 $I_n = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} f(x_i)$, $I = \dfrac{1}{b-a}\displaystyle\int_a^b f(x)\mathrm{d}x$，则有 $I_n \sim N\left(I, \dfrac{\sigma^2}{n}\right)$，其中 σ^2 是分布 f

的方差.

蒙特卡洛的实现依赖于大数法则与中心极限定理，前者说明可以通过随机抽样进行蒙卡方法，后者用来评估蒙卡方法的精度：

$$|I_n - I| < \lambda \frac{\sigma}{\sqrt{n}}.$$

即蒙卡的精度依赖于随机变量样本自身的标准差与随机试验的次数.

1.2 伪随机数

随机数分为真随机数、准随机数、伪随机数.

伪随机数 伪随机数序列由**种子**确定，不是真正的随机，但有类似随机数的统计特征：**均匀性、独立性**. 蒙特卡洛研究一般采用伪随机数代替真正的随机数.

伪随机数产生方法 平方取中法（$x_{n+1} = [10^{-r} x_n^2](\mod 10^{2r})$，$\xi = x_n/10^{2r}$）、线性同余法（$x_{n+1} = (ax_n + c)(\mod m)$，$\xi = x_n/m$）、程序语言自带.

伪随机数的统计检验 构造卡方统计量 $\chi^2 = \sum_{i=1}^{k} \dfrac{\left(N_i - \dfrac{N}{k}\right)^2}{\dfrac{N}{k}}$ 进行卡方检验.

1.3 随机抽样方法

1.3.1 离散随机变量的直接抽样

对于离散随机变量 $P(X = x_i) = p_i$，将 $[0,1]$ 划分为 n 个区间，第 i 个区间长度为 p_i，且 $\sum_{i}^{n} p_i = 1$，利用伪随机数 $\xi \in [0,1]$ 生成随机样本 x_i.

1.3.2 连续变量的直接抽样

反函数法 对于分布函数 $F(x)$，随机变量 $F^{-1}(\xi)$ 满足分布 x，其中 $\xi \in [0,1]$ 是伪随机数.

1.3.3 变换抽样法

对于随机变量 X, Y，分布密度函数满足：

$$f(x) = \phi(y) \left| \frac{\mathrm{d}y}{\mathrm{d}x} \right|.$$

若是二维随机变量 $(x,y),(u,v)$，则有：

$$f(x,y)=\phi(u,v)\left|\frac{\partial(u,v)}{\partial(x,y)}\right|.$$

例：标准正态分布 考虑 $[0,1]$ 上均匀分布 (u,v)，做变换：

$$x=\sqrt{-2\ln u}\cos(2\pi v),\quad y=\sqrt{-2\ln u}\sin(2\pi v).$$

则 (x,y) 是相互独立的正态分布样本.

1.3.4 舍选法

第一类舍选法 对于区间 $[a,b]$ 的函数 $f(x)$，存在上确界 L. 则生成 $[a,b]\times[0,L]$ 的随机数 (ξ_1,ξ_2)，若点在 $f(x)$ “下面”，则接受 ξ_1. 舍选效率 $E=\dfrac{\int_a^b f(x)\mathrm{d}x}{L(b-a)}$.

第二类舍选法 按照 $g(x)$ 采样，接受概率为 $A(x)=\dfrac{f(x)}{Mg(x)}$. 当 $g(x)$ 为均匀分布时退化为第一类舍选法.

1.3.5 复合抽样法

1.3.6 特殊近似抽样法

1.3.7 高维随机向量的抽样

1.4 减小方差的技巧

1.4.1 分层抽样法

在被积函数极小值附近，对积分贡献小，权重小，可以相对少抽样；在被积函数极大值附近，对积分贡献大，权重大，需要相对多抽样.

1.4.2 重要抽样法

对于剧烈变化的函数 $f(x)$，可以通过偏倚分布函数 $g(x)$ 辅助计算.

$$I=\int_a^b\frac{f(x)}{g(x)}g(x)\mathrm{d}x.$$

可以直接通过抽取 $g(x)$ 的抽样 η_i [1]来估算积分值：

$$I = \frac{1}{N}\sum_{i=1}^{N}\frac{f(\eta_i)}{g(\eta_i)}.$$

统计方差 $V\left(\frac{f}{g}\right) < V(g)$.

1.4.3 其他方法

1.5 随机游走

1.5.1 Metropolis 方法

生成满足 $f(x)$ 的随机游走序列 $\{x_n\}$ 的过程如下：

1. 参数：初始点 x_0，δ；
2. 对于第 n 个随机数 x_n，试探位置 $x_{\text{try}} = x_n + \eta$，其中 $\eta \in [-\delta, \delta]$；
3. 计算 $r = \dfrac{f(x_{\text{try}})}{f(x_n)}$，若 $r \geqslant \xi$，则 $x_{n+1} = x_{\text{try}}$，否则继续尝试新的 x_{try}，$\xi \in [0,1]$ 是随机数.

1.6 蒙特卡洛应用

1.6.1 中子运输

1.6.2 定积分求解

期望值估计法（平均值法）、掷点法

1.6.3 统计物理

1.6.4 量子力学

[1] Recall：直接抽样，抽取 $\xi \in [0,1]$，$\eta = G^{-1}(\xi)$.

2 有限差分法

2.1 基础概念

2.2 常微分方程的求解

考虑微分方程：

$$\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}x} = f(x,y), \\ y(x_0) = y_0. \end{cases}$$

在区间 $[a,b]$ 内生成子区间 $\{[x_n, x_{n+1}]\}$，每一个点 x_n 对应 y_n.

2.2.1 欧拉法

直接利用向前差分代替微分：

$$\frac{y_{n+1} - y_n}{h} = f(x_n, y_n).$$

即为：

$$y_{n+1} = y_n + hf(x_n, y_n).$$

2.2.2 梯形法

将向前差分和向后差分平均得到差分，即为梯形法：

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})].$$

此时一般用迭代求解：

$$\begin{cases} y_{n+1}^{(0)} = y_n + hf(x_n, y_n), \\ y_{n+1}^{(k+1)} = y_n + \dfrac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(k)})]. \end{cases}$$

2.2.3 龙格-库塔法

考虑对某几个点斜率 k_i 加权平均：

$$y_{n+1} = y_n + h\sum_N R_i k_i.$$

其中 R_i 为归一化系数，该递推被称为 N 阶龙格-库塔. （二阶）龙格-库塔引入了新的参数 R_1, R_2, a, b 满足：

$$k_1 = f(x_n, y_n), \quad k_2 = f(x_n + ah, y_n + bhk_1).$$

对 k_2 和 $y(x_{n+1})$ 泰勒展开，比较系数可以求出二阶龙格库塔的截断误差为 $|y_{n+1} - y(x_n + 1)| \sim O(x^3)$，系数为 $R_1 = 0,\ R_2 = 1,\ a = b = \dfrac{1}{2}$，方程递推公式为：

$$\begin{cases} y_{n+1} = y_n + hk_2, \\ k_1 = f(x_n, y_n), \\ k_2 = f\left(x + \dfrac{h}{2}, y_n + \dfrac{hk_1}{2}\right). \end{cases}$$

2.3 偏微分方程的求解

考虑偏微分方程：

$$\begin{cases} L\phi = q, \\ \phi\big|_G + g_1(s)\dfrac{\partial \phi}{\partial \mathbf{n}}\Big|_G = g_2(s). \end{cases}$$

其中 L 为微分算子，G 为边界，s 为边界参数.

我们常考虑 L 为斯杜-刘维尔算符：

$$L = -\nabla(p\nabla) + f.$$

其中 p, f 为函数，当 $p = 1, f = 0$ 时方程即为泊松方程.

我们考虑 $p = 1$ 时的方程：

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + f(x, y)\phi = q(x, y).$$

有限差分法得到上述方程离散化为：

$$\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} + (h^2 f_{i,j} - 4)\phi_{i,j} = h^2 q_{i,j}.$$

其中 i, j 是二维平面的格点.

2.4 差分方程组

2.4.1 直接求解

2.4.2 直接迭代

2.4.3 高斯-赛德尔迭代

2.4.4 超松弛迭代

2.4.5 Hockney 直接法

3 有限元素法

3.1 基础概念

基本思想 把解微分方程的问题转化为变分问题.

3.2 有限元素法流程

1. 将待解函数 φ 作用的坐标空间划分成网络，每个格点之间作为一个元素；
2. 网格上每个点的函数值 $\{\varphi_i\}$ 视为变分参数；
3. 元素内线性插值；
4. 元素内建立泛函，写成线性方程形式；
5. 总泛函（区域泛函求和）求极值，考虑边界条件；
6. 解线性方程组.

3.3 有限元与有限差分的比较

- 相似性：偏微分方程离散化、与步长有关、可以转化为线性方程组迭代求解；
- 不同：有限差分考虑微分方程的差分形式，有限元素考虑泛函极值；有限差分区域划分要求规则（矩形），有限元素区域划分灵活；有限差分边界条件特殊处理，有限元素用统一的观点处理边界条件（程序上更容易实现）；有限差分适用范围广，有限元素因为需要寻找泛函因此应用场景有限.

重点比较角度：基本方法、区域划分（有限元素划分灵活）、边界条件（有限元素易于程序化）、适用范围（有限差分更广泛）.

4 分子动力学

分子动力学方法 按照体系内部的内禀东西学规律来计算确定位形的转变，不存在随机因素.

4.1 基础概念

4.1.1 周期性边界条件

分子动力学往往用于研究大块物质在给定密度下的性质，但实际计算中很难在无穷大的系统中进行，因此引入边界条件，利用少量粒子的模拟预测体系的宏观性质.

元胞：分子动力学的体积元

1. 立方体：最简单且使用广泛，适用于气体和液体，不适用于晶体；
2. 六方柱：DNA 分子；
3. 截断正八面体和十二面体：球形分子，前者使用广泛，后者最省空间.

为了将分子动力学元胞内的模拟扩大到真实系统的模拟，通常采取周期性边界条件，可以消除引入元胞后的表面效应，构造出一个准无穷大的体积来更精确地代表宏观系统，让小体积元胞镶嵌在一个无穷大的物块之中.

最小像力约定 假设一个元胞中有 N 个粒子，在考虑不同粒子之间相互作用时，通常采用最小像力约定：在无穷重复的基本元胞中，每个粒子只同他所在的基本元胞的另外 $N-1$ 个粒子或其最临近的影像粒子发生相互作用.

简单来说，粒子间相互作用的力程被限制在以粒子为中心、与元胞尺度 L 形状相同的体积元中.

Lennard-Jones 势：

$$V = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right].$$

其中 $-\varepsilon$ 是最小位势，在 $2^{1/6}\sigma$ 处取得，$r=\sigma$ 处势能为零.

4.1.2 分子力场与截断距离

分子力场包括成键与非成键部分：

1. 键伸缩能：$E_s = \dfrac{1}{2}k_s(l-l_0)^2$；
2. 键角（弯折能，考虑谐振子模型）：$E_B = \dfrac{1}{2}k_b(\theta-\theta_0)^2$；

3. 二面角（扭转能）：$E_R = \sum_{n=0}^{N} \frac{V_n}{2}[1+\cos(n\omega-\gamma)]$，其中 V_n 为势垒高度，n 为多重度（键的极小值点个数），γ 为相因子，ω 为转角；
4. 范德华力：Lennard-Jones 势；
5. 静电相互作用：Coulomb 势.

截断距离 问题：分子相互作用计算复杂度 N^2?

解决方案：常考虑截断距离 $r_c < \frac{L}{2}$ 来降低计算量，只计算 $r < r_c$ 的相互作用，此时要求 $V(r_c)$ 很小.

新的问题：判断 r_c 仍需要 N^2?

解决方案：单次循环，建立邻居列表（判断距离 $r_n > r_c$），往往十倍时间步长更新一次（判断距离越大，邻居越多，更新频率越小）.

新的问题：计算 r_n 仍需要 N^2?

解决方案：元胞划分为 M^3 个边长为 l 的格子，$l > r_n$，在每个元胞所在格子及相邻格子（共 27 个格子）中搜索邻居.

新的问题：静电相互作用是长程力，范德华相互作用的短程力?

解决方案：分别设置两个阶段距离，双重截断. 实际操中常常用基团间的截断代替基于原子距离的截断.

新的问题：截断距离附近势能与力不连续?

解决方案：

1. Shifted Potential: 势能减去 $v_c = V(r_c)$，势能变得连续. 同时再加上 $(r-r_c)\frac{\mathrm{d}V}{\mathrm{d}r}\Big|_{r=r_c}$，在截断处势能导数为零，力连续.
2. Switch Function: 势能乘一个满足 $f(0)=1, f(r_c)=0$ 的函数.

4.1.3 时间步长

时间步长越长，误差越大，时间步长越小，弛豫时间越长.

实际应用中采样频率为信号最高频率的 5 ∼ 10 倍，以此保证采样后的数字信号完整保留原始信号的信息. 时间步长往往选为系统中最短运动周期的 $\frac{1}{10}$.

分子体系中最高频运动往往是含氢原子的化学键振动.

4.1.4 约束动力学

“固定”分子中的高频运动，同时不应影响低频运动，从而增大时间步长. 例如约束包含氢原子的化学键.

要求：高频明显高于其他运动的频率，且与其他运动耦合较弱.

4.1.5 SHAKE 算法

4.2 微正则系综和正则系综的模拟流程

微正则系综：E, N, V 守恒；
正则系统：T, N, V 守恒；
巨正则系综：T, μ, V 守恒（N 不守恒，因此往往不对其进行计算）

4.2.1 Verlet 算法

二步法差分计算牛顿方程的一种，一般由初始跳脚分为两种：

1. 给定初始位置 $\{\mathbf{r}_i^{(1)}\}$ 与速度 $\{\mathbf{v}_i^{(1)}\}$；
2. 位置递推：$\mathbf{r}_i^{(n+1)} = \mathbf{r}_i^{(n)} + h\mathbf{v}_i^{(n)} + \dfrac{h^2}{2m}\mathbf{F}_i^{(n)}$；
3. 速度递推：$\mathbf{v}_i^{(n+1)} = \mathbf{v}_i^{(n)} + \dfrac{h}{2m}(\mathbf{F}_i^{(n+1)} + \mathbf{F}_i^{(n)})$.

1. 给定初始位置 $\{\mathbf{r}_i^{(0)}, \mathbf{r}_i^{(1)}\}$；
2. 位置递推：$\mathbf{r}_i^{(n+1)} = 2\mathbf{r}_i^{(n)} - \mathbf{r}_i^{(n-1)} + \dfrac{h^2}{2m}\mathbf{F}_i^{(n)}$；
3. 速度递推：$\mathbf{v}_i^{(n)} = \dfrac{\mathbf{r}_i^{(n+1)} - \mathbf{r}_i^{n}}{2h}$.

趋衡过程 增加或从系统中移出能量，直到系统具有所要求的能量. 往往在模拟过程中先模拟若干步，再通过对初始速度乘标度因子 β，回到第一步，再次反复，直到达到平衡，来接近平衡态. 标度因子：

$$\beta = \left[\frac{T^\star(N-1)}{16\sum_i v_i^2}\right]^{\frac{1}{2}}.$$

弛豫时间 达到平衡所需的时间.

遍历性假设 在足够长时间下，物理量的时间平均 = 系综平均. 我们可以利用遍历性假设求解各种物理量.

4.3 分子动力学的优劣

4.4 分子动力学和蒙特卡洛方法的比较

- 分子动力学是确定性模拟方法，通过数值求解粒子的运动方程来模拟整个系统的行为，适合于运动方程可以推导并可以积分的情况.
- 蒙特卡洛模拟是随机性模拟方法，通过不断产生随机数序列来模拟过程. 特点是不需要计算能量的梯度，适合于离散的状态空间采样，但需要仔细设计如何移动.

4.5 分子动力学的局限性

分子动力学的局限性：模拟准确性受限，计算量大，只能进行有限时间的模拟，涉及采样效率问题.

往往此时需要进行高性能计算.

5 高性能计算

5.1 并行计算与串行计算

- 串行计算：分为“指令”和数据两部分，在程序执行时“独立地申请和占有”内存空间，所有计算均局限于该饿你存空间；
- 并行计算：将进程相对独立地分配于不同的节点上，由各自独立的操作系统调度，享有独立的 CPU 和内存资源，进程间可以通过消息传递信息.

任务 并行程序所能处理的并发性最小单元.

进程 一个完成任务的实体. 任务被分配给进程. 与处理器不同，处理器时物理资源，进程时抽象化处理器的方式，往往将进程映射到处理器进行处理.

串行程序并行化

1. 将问题分解成任务；
2. 任务分配给进程；
3. 进程进行必要的数据访问、通信和同步；
4. 进程映射或绑定到处理器.

进程间通信 通信、同步、聚集.

通信 读/写操作系统提供的共享数据缓存区.

同步 多个进程之间相互等待的操作.

聚集 多个进程的局部结果综合起来，产生一个新的结果，存储在某个进程或所有进程的变量中.

并行效率 计算时间是分钟级，数据传输时间是秒级.

5.2 MPI

MPI 基于消息传递的并行程序的标准.

MPI 有两种模式：但程序多数据流模式 (SPMD) 与多程序多数据流模式 (MPMD).

MPI 流程：

1. 程序声明参数;
2. `MPI_init`
3. `MPI_Comm_rank, MPI_Comm_size`
4. `MPI_Finalize`

5.3 两种分子动力学并行算法

粒子分解 将粒子分配到不同的处理器. 较易实现，难以实现大规模并行，通信开销大.

域分解 每个处理器分担一个小盒子内的计算. 通信量低，并行效率高，但负载不均.

6 计算机代数

7 机器学习

7.1 基础概念

机器学习 利用已有数据进行学习，获取数据中的特征，构造或完善具有预言能力的模型，提高未来行动的效率、效果，以及准确性.

7.2 监督学习

7.2.1 线性回归

通过线性回归进行预测. 假设样本为 $\{(\mathbf{x}^{(n)}, y^{(n)})\}$[1]，假设函数 h 往往写为：

$$h(\tilde{\mathbf{x}}) = \mathbf{w}^\top \tilde{\mathbf{x}}.$$

其中 $\tilde{\mathbf{x}} = (1, \mathbf{x})$.

损失函数 度量单样本预测错误的程度，损失函数越小，模型越好.

代价函数 度量全部样本集的平均误差.

目标函数 最终要优化的函数. 优化方法：最小二乘法、梯度下降法.

最小二乘法 (OLS) 求平方损失函数 $J(\mathbf{w}) = \frac{1}{2}(\mathbf{X}\mathbf{w}-\mathbf{y})^2$ 的最小值，得到 $\mathbf{w} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$，其中 $\mathbf{X} = (\tilde{x}^{(1)}, \ldots, \tilde{\mathbf{x}}^{(n)})^\top$.

批量梯度下降 (BGD) 用所有样本迭代更新 w_j ：

$$w_j = w_j - \alpha\frac{1}{n}\sum_{i=1}^{n}\left((h(\mathbf{x}^{(i)}) - y^{(i)})x_j^{(i)}\right).$$

其中 α 称为学习率，学习率应适中，过大会导致代价函数振荡，过低会导致收敛速度慢.

[1] 我们用上标 (i) 表示第 i 组样本，下表 j 表示向量第 j 个分量.

随机梯度下降 (SGD) 用一个样本更新 w_j：

$$w_j = w_j - \alpha(h(\mathbf{x}^{(i)}) - y^{(i)})x_j^{(i)}.$$

容易陷入局部极小.

小批量梯度下降 (MBGD) 只用一低昂批量的训练样本.

梯度下降与最小二乘的比较 梯度下降需要多次迭代；最小二乘直接计算，复杂度为 $O(n^3)$.

数据规范化 线性模型需要做数据的归一化/标准化，但集成学习模型不需要.

误差 训练误差指模型在训练集上的误差，也称为经验误差；泛化误差模型在测试集上的误差. 训练误差相比泛化误差小于、约等于、大于分别对应过拟合、拟合和欠拟合.

拟合不当的处理 欠拟合：添加新的特征、增加模型复杂度、减小正则化系数；过拟合：更多训练数据（减小噪声影响）、降维、正则化、集成学习.

正则化 L1 正则化（Lasso 回归）、L2 正则化（Ridge 回归）、弹性网络. L1 正则化和 L2 正则化分别对应在损失函数 $J(\mathbf{w})$ 后加上 $\lambda\sum_{j=1}^{n} w_j^k$，其中 k 对应 Lk 正则化.

L1 正则化能产生稀疏模型，用于特征选择；L2 正则化能降低参数范围的总和，防止过拟合.

回归的评价指标 均方误差 (MSE)、均方根误差 (RMSE)、平均绝对误差 (MAE). 对于误差的几种不同形式的求和：

$$\left(\frac{1}{n}\sum_{i=1}^{n}|y^{(i)} - h(\mathbf{x}^{(i)})|^k\right)^{\frac{1}{l}}.$$

三个评价指标分别对应 $k=2, l=1$; $k=2, l=2$; $k=1, l=1$.

MSE 连续可导，标与使用梯度下降法，是常用的损失函数. 但 MSE 对离群点敏感，MAE 对离群点不敏感.

此外还有回归平方和（SSR，预测值雨平均值的误差）、残差平方和（SSE，预测值和真实值的误差）、总离差平方和（SST，平均值和真实值的误差），以及决定系数 R 方：

$$R^2 = \frac{\text{SSR}}{\text{SST}}.$$

7.2.2 逻辑回归

线性回归解决回归问题，输出连续值；逻辑回归解决分类问题，输出离散值，其本质是在线性回归的基础上加上了一个 Sigmoid 函数（非线性映射），使其成为了一个分类算法.

7.2.3 决策树

将一个复杂的多类别分类问题转化为若干个简单的分类问题. 常见术语有：根节点、非叶子节点（测试条件）、分支（测试结果）、叶节点（分类标记）.

原理 从训练数据中不断决策划分，属于判别模型；归纳分类算法、监督学习算法、贪心算法、分割方法.

优点与不足 推理过程容易理解，计算简单，可解释性强，可判断属性变量的重要性，为减少变量数目提供参考；是一种“弱”学习方法，适合集成.

决策树超参数一般有树的高度、叶子的子集统计量等，常见的决策树多为“二叉树”.

决策树的三种基本类型 ID3、C4.5、CART.

ID3 以信息增益为衡量标准，选择信息上最大的特征最为当前决策节点，实现对数据的归纳分类，偏向特征值多的特征；信息增益 = 信息熵-条件熵.

C4.5 以信息增益率选择属性，偏向特征值小的特征.

CART 用基尼指数选择属性，克服 C4.5 需要的巨大计算量，偏向于特征值较多的特征.

剪枝 去掉一些分支，降低过拟合的风险. 在每个节点划分前先估计，若不能提升泛化能力，则标记为叶节点，这种被称为“预剪枝”. 另一种是在已经生成的决策树上进行剪枝，被称为“后剪枝”，往往比预剪枝保留更多分枝，欠拟合风险小，泛化性能更优.

7.2.4 集成学习

利用多个较弱的统计模型集成预测.

集成学习方法 袋装法、提升法、堆叠法.

袋装法 将训练样本随机分成 M 份，分别训练再集成；往往需要足够大的数据集，数据集有限的情况下可以用经验重抽样法.

随机森林 袋装法的扩展，将样本随机分为多组，随机选择特征，构建决策树，最后随机森林平均得到预测结果.

增强算法

7.2.5 神经网络

分类 深度神经元网络 (DNN)、卷积神经元网络 (CNN)、循环神经元网络 (RNN). 分别用于深度学习、图像识别、自然语言处理.

常见模型 输入层 $\mathbf{x}$ -> 隐层 $\mathbf{b}$ -> 输出层 $\hat{\mathbf{y}}$. $\mathbf{b} = \mathbf{V}\mathbf{x}$, $\hat{\mathbf{y}} = \mathbf{W}\mathbf{b}$.

神经元函数 每个神经元独立权重 $z^{(i)} = \mathbf{x}^\top \cdot \mathbf{w}^{(i)}$，与非线性变换 $\sigma(z^{(i)})$.

反向传播 (BP) 基本步骤:

1. 将输入样本提供给输入层神经元;
2. 逐层将信号传递给隐层、输出层，产生输出层结果;
3. 计算输出层误差;
4. 将误差反向传播至隐层神经元;
5. 根据隐层神经元对连接权重和阈值进行调整.

BP 算法的优缺点 能够自适应、自主学习，拥有很强的非线性映射能力，推导过程严谨科学; 但收敛速度慢，隐层节点数没有明确准则，容易陷入局部极小值问题.

神经网络的优缺点 神经网络适用于表征学习，充分利用海量数据与算力; 但需要有标签的同种数据，数据需求量大，有些物理问题不是分类问题，难以解决.

7.3 非监督学习

7.3.1 聚类

聚类 基于研究对象属性的相似性对研究对象进行分组，使足内样本相似，组间样本有差异.

聚类尺度函数 几何距离、相似度、相关系数.

聚类算法 分层算法、分割算法、密度聚类

聚合聚类与分裂聚类 分别为自底向上合并相似对象和自顶向下分裂一个类的方法

7.3.2 降维

Why 降维? 高维数据增加了运算难度，使得学习方法泛化能力变弱，降维能增加数据的可读性. 降维可以减少冗余特征，加速算法运行，降低存储和计算成本.

主成分分析法 (PCA) 把数据变换到一个新的坐标系，使得任何数据投影的第一大方差在第一个坐标（第一主成分）上，第二大方差在第二个坐标（第二主成分）上.

基本步骤:

1. 数据标准化，计算数据的协方差矩阵 $\Sigma = \frac{1}{m}\mathbf{X}\mathbf{X}^\top$;
2. 计算协方差矩阵的特征值与特征向量;
3. 将特征值最大的 k 个特征向量组成特征向量矩阵 P;
4. 将数据转化到降维后的空间得到 $Y = PX$.

PCA 的优缺点 仅需要以方差衡量信息，不受数据集意外的因素影响，可以消除原始成分间的相互影响，计算方法简单，无参数限制；但不如原始样本特征解释性强，丢失的信息可能对数据处理有影响，k 难以确定.