

计算方法

Computational Methods

Lecture 2: 插值——如何从有限个观察值恢复整体？

Interpolation: Reconstruction, Redundancy, and Approximation

刘景钺

南京大学 · 计算机学院

课程信息

计算方法 Fall 2026

授课教师 刘景铎 liu@nju.edu.cn

上课 周四 5-6 节 · 1-18 周 · 仙 I-319

答疑 周四下午 4 点

助教 周宇恒 · 侯哲

作业 nm_nju_2026@163.com

课程主页 TCS Wiki · Fall 2026

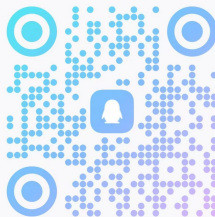
QQ 群 1054896504 加入时注明姓名、专业、
学号

教材: Timothy Sauer, 《数值分析》(原书第 2 版)。 最新公告以
课程主页为准。



计算方法 Fall 2026

群号: 1054896504



扫一扫二维码, 加入群聊



回顾：从求根到插值

上一讲： 方程求根

$$f(x) = 0$$

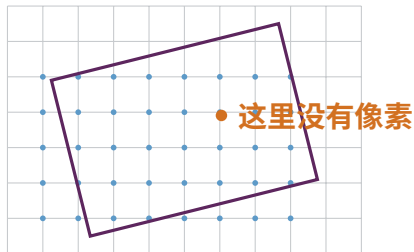
- 二分法：全局保证；
- 不动点迭代：构造收缩映射 / 局部动力学；
- 根的敏感性与牛顿法。

这一讲： 插值 (Interpolation)

$$(x_i, y_i) \quad i = 1, \dots, n$$

只看到一些**不完整的观察值**，
怎样恢复**完整对象**？

插值：像素之间有什么？



数字图像只给出网格上的值：

$$I(i, j).$$

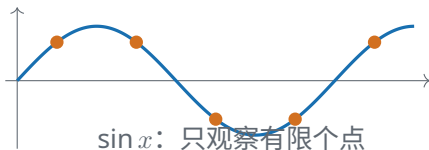
旋转后却需要

$$I(117.32, 48.71).$$

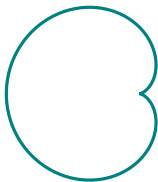
问题：这些离散样本背后，
应该假设存在什么连续对象？

一次旋转 360° 与连续做 360 次 1° 旋转，数值上会一样吗？

函数值之间

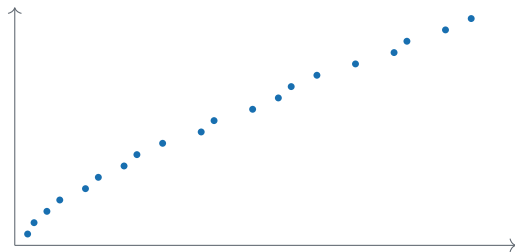


曲线与形状



心形线 (Cardioid)

离散统计量之间



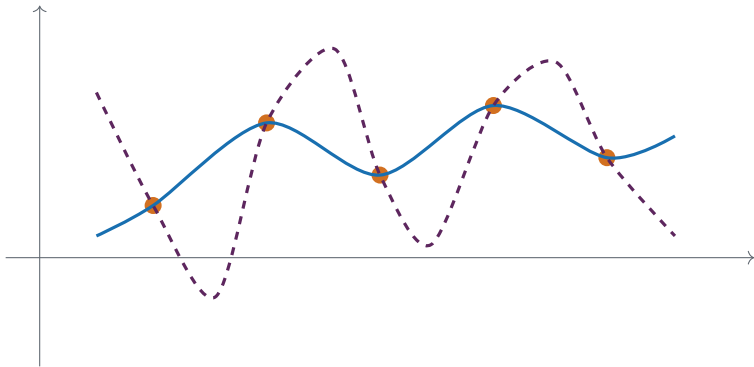
素数计数函数 $\pi(x)$

换坐标后，结构可能更简单

$$y = x^3 \implies \log y = 3 \log x.$$

双对数 (log-log) 坐标下变成直线。

有限个观察值不能唯一确定函数



同样一组有限数据点，可以对应**无穷多个**不同的函数。

插值之前，必须先限定未知函数所属的**模型类**。

先区分两件事：精确重建与数值逼近

精确重建

Algebraic Reconstruction

先假设未知对象真的就是

$$p \in \mathcal{P}_{<n}.$$

问题是：有限个取值能否**唯一决定** p ?

这是一个代数问题：不需要连续性，甚至可以在有限域上讨论。

数值逼近

Numerical Approximation

真实函数 f 未必是多项式；我们用 P 去近似它：

$$P(x_i) = f(x_i).$$

问题是：在采样点之间， P 与 f 相差多大？

连续性只能排除跳变；要得到更强的定量误差界，还需要导数有界或更高阶的光滑性。

本讲先把**精确重建**讲清楚；最后再回到**数值逼近**，这时才真正需要讨论误差与光滑性。

为什么多项式同时适合这两个问题？

作为精确模型：有限维，代数结构清楚

$$p(x) = a_0 + a_1x + \cdots + a_nx^n.$$

- ▶ $n + 1$ 个系数 = $n + 1$ 个自由度；
- ▶ 加、乘、求导、积分都封闭且易算；
- ▶ 在复数域中

$$p(x) = a_n \prod_{i=1}^n (x - r_i),$$

实系数时非实根成共轭对出现。

作为逼近模型：表达能力强

- ▶ 泰勒展开 (Taylor expansion) 给出局部多项式近似；
- ▶ Weierstrass：闭区间上的连续函数可被多项式任意一致逼近；
- ▶ 电路和神经网络当然也可以用来表示函数；多项式则是最容易分析的起点。

注意：Weierstrass 只保证“好的逼近多项式存在”，**并不保证**任意节点上的高次插值会收敛。

同一个多项式，两套坐标：插值就是换坐标

次数 $< n$ 的多项式构成一个 n 维空间。

系数坐标

$$\mathbf{a} = (a_0, \dots, a_{n-1}),$$
$$p(x) = \sum_{j=0}^{n-1} a_j x^j.$$



取值坐标

$$\mathbf{y} = (y_1, \dots, y_n),$$
$$y_i = p(x_i), \quad x_i \text{ 两两不同}.$$

若使用单项式基，这个坐标变换可以写成一个 Vandermonde 线性方程组：

$$\mathbf{y} = V\mathbf{a}, \quad V_{ij} = x_i^{j-1}, \quad y_i = \sum_{j=0}^{n-1} a_j x_i^j.$$

拉格朗日插值：直接选择一组与取值操作相适应的基，不必先求出单项式系数。

拉格朗日插值：存在且唯一

给定 n 个横坐标互不相同的数据点

$$(x_1, y_1), \dots, (x_n, y_n),$$

存在唯一一个次数至多 $n - 1$ 的多项式 P ，满足

$$P(x_i) = y_i \quad i = 1, \dots, n.$$

先构造存在性，再证明唯一性。

拉格朗日基函数：只在一个节点取 1

想找 L_k 使得

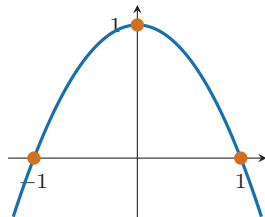
$$L_k(x_k) = 1, \quad L_k(x_j) = 0 \quad (j \neq k).$$

先让它在其他节点自动为零：

$$L_k(x) = A_k \prod_{j \neq k} (x - x_j).$$

归一化得到

$$L_k(x) = \prod_{j \neq k} \frac{x - x_j}{x_k - x_j}.$$



例：节点 $-1, 0, 1$ 时，
 $L_2(x) = 1 - x^2$ 。

这些基函数就是取值坐标的“单位向量”

因为

$$L_k(x_i) = \mathbf{1}\{i = k\},$$

所以令

$$P(x) = \sum_{k=1}^n y_k L_k(x)$$

就有

$$P(x_i) = \sum_{k=1}^n y_k L_k(x_i) = y_i.$$

每个 L_k 在第 k 个节点取 1，在其他节点取 0。

唯一性: n 个点已经用完所有自由度

假设 P, Q 都次数至多 $n - 1$, 而且都通过给定的 n 个点。
令

$$H(x) = P(x) - Q(x).$$

那么

$$H(x_1) = \cdots = H(x_n) = 0.$$

但一个**非零**的次数至多 $n - 1$ 的多项式, 最多有 $n - 1$ 个不同的根。
因此 $H \equiv 0$, 即

$$\boxed{P \equiv Q.}$$

一分钟练习：什么时候次数会更低？

用拉格朗日插值找出通过

$$(1, 1), \quad (2, 1)$$

的次数至多 1 的多项式。

一分钟练习：什么时候次数会更低？

用拉格朗日插值找出通过

$$(1, 1), \quad (2, 1)$$

的次数至多 1 的多项式。

$$L_1(x) = \frac{x-2}{1-2} = 2-x, \quad L_2(x) = \frac{x-1}{2-1} = x-1.$$

因此

$$P(x) = L_1(x) + L_2(x) = \boxed{1}.$$

“次数至多 $n-1$ ” 并不意味着实际次数一定达到 $n-1$ 。

计算复杂性：公式写出来以后，表示仍然重要

直接用

$$P(x) = \sum_{k=1}^n y_k L_k(x)$$

在一个新点求值：每个 $L_k(x)$ 含 $O(n)$ 个因子，朴素实现需要

$$O(n^2)$$

次算术运算。

表示方式	构造或预处理	之后单点求值
直接拉格朗日公式	—	$O(n^2)$
牛顿差商形式	$O(n^2)$	$O(n)$
拉格朗日重心形式	$O(n^2)$	$O(n)$

同一个数学对象，换一种表示，就可能得到不同的计算复杂性与数值行为。

同一个取值映射，三种计算任务

设 $\mathcal{P}_{< n}$ 是次数 $< n$ 的多项式空间。如果已知 m 个互不相同位置上的取值：

$$p \mapsto (p(\alpha_1), \dots, p(\alpha_m)).$$

$$m < n$$

信息不足，存在歧义
+ 随机化



秘密分享

$$m = n$$

刚好是坐标
唯一恢复



插值

$$m > n$$

加入冗余
容忍缺失或错误



纠错码

少给、刚好给够、额外多给不是三个彼此无关的技巧，而是同一个取值映射对应的三种信息结构。

11 位科学家希望保护一个高度机密的档案：

当且仅当至少 6 人在场，才能打开档案。

如果只用普通锁和钥匙：

$$\binom{11}{5} = 462 \quad \text{把锁}, \quad \text{每人需要} \binom{10}{5} = 252 \quad \text{把钥匙}.$$

人数继续增加时，这种组合构造会迅速爆炸。

Shamir (1979): **用一个 5 次多项式代替几百把锁。**

有限域上的多项式

为简单起见，先取 q 为素数：

$$\mathbb{F}_q = \mathbb{Z}/q\mathbb{Z}.$$

加、减、乘、除都在模 q 的意义下进行（除数非零）：

$$p(x) = a_0 + a_1x + \cdots + a_kx^k, \quad a_i, x \in \mathbb{F}_q.$$

系数表示

$$(a_0, \dots, a_k)$$

取值表示

$$(p(x_0), \dots, p(x_k))$$

只要横坐标互不相同，次数 $\leq k$ 的多项式仍由 $k+1$ 个点唯一决定。

有限域上的算术本身是精确的；后面所说的“错误”来自通信或存储过程中符号发生改变，而不是浮点舍入误差。

Shamir 秘密分享：把秘密放在 $p(0)$

在有限域 $\mathbb{F}_q$ 中，令秘密为 $s \in \mathbb{F}_q$ ，随机选择

$$a_1, \dots, a_5 \in \mathbb{F}_q,$$

并定义

$$p(x) = s + a_1x + \dots + a_5x^5.$$

第 i 个人得到一份份额：

$$(x_i, p(x_i)), \quad x_i \neq 0,$$

且所有 x_i 互不相同。

任意 6 人拿到 6 个点，就可以用拉格朗日插值恢复 p ，进而得到

$$s = p(0).$$

少于 6 份：为什么秘密仍然完全隐藏？

已经看到的 5 份份额

$$(x_1, y_1), \dots, (x_5, y_5)$$

这 5 个点固定不动。

任取一个候选秘密 $s' \in \mathbb{F}_q$ ，加入第 6 个点 $(0, s')$ 。

这 6 个点唯一确定一个次数至多 5 的多项式：

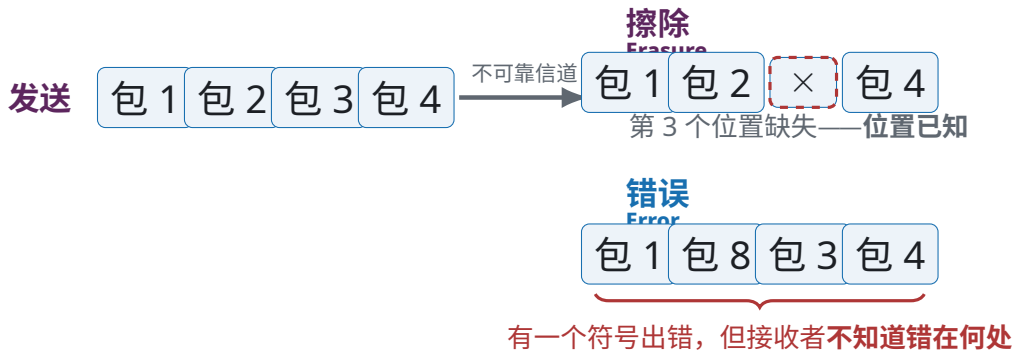
$$5 \text{ 份份额} + (0, s') \implies \boxed{\text{唯一的 } p_{s'}(x)}.$$

所以，对**每一个** s' ，都恰好有一组

$$(a_1, \dots, a_5)$$

与已经看到的份额相容。

由于随机系数 $(a_1, \dots, a_5)$ 是均匀选择的，因此
无论真实秘密取什么值，这 5 份份额的联合分布都完全一样。



TCP/IP 常用**重传**来处理丢包；但重传的代价可能很高，也可能带来很大的延迟，甚至根本无法重传。

能不能预先加入冗余，让接收端直接恢复原来的消息？

擦除： $n + k$ 个取值就够了

想发送 n 个符号

$$a_0, \dots, a_{n-1} \in \mathbb{F}_q,$$

信道最多擦除 k 个位置。

应该发送几个符号？

令

$$p(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1},$$

并发送

$$\boxed{p(\alpha_1), \dots, p(\alpha_{n+k})},$$

其中 α_i 两两不同。

擦除 k 个以后仍至少剩 n 个点；用拉格朗日插值恢复 p ，再读出

$$(a_0, \dots, a_{n-1}).$$

$n + k$ 足够。稍后用一个普适下界证明：它也恰好是最优的。

距离：离散世界里的“稳定裕量”

对 $s, t \in \mathbb{F}_q^m$ ，定义汉明距离（Hamming distance）

$$d(s, t) = \#\{i : s_i \neq t_i\},$$

$$d_{\min}(C) = \min_{u \neq v} d(C(u), C(v)).$$

第一讲：连续扰动

看输入误差会被放大多少。

关注的是局部敏感性与条件数。

这一讲：离散扰动

主动把不同码字分得足够开。

关注的是不同码字之间的分离度。

鲁棒性的一种普遍策略：让不同合法对象之间留出足够的“安全裕量”。

最小距离告诉我们能承受多少故障

k 个擦除

删掉任意 k 个坐标后，不同码字仍必须可区分：

$$d_{\min} \geq k + 1.$$

擦除的位置是已知的。

k 个错误

同一个接收词不能同时距离两个合法码字都不超过 k ：

$$d_{\min} \geq 2k + 1.$$

错误的位置未知，因此还必须排除“解码歧义”。

若某个接收词 r 同时满足 $d(r, C(u)) \leq k$ 与 $d(r, C(v)) \leq k$ ，则由三角不等式

$$d(C(u), C(v)) \leq 2k.$$

所以要唯一纠正任意 k 个错误，必须让不同码字之间满足 $d_{\min} \geq 2k + 1$ 。

Singleton 界：距离与冗余的普遍上界

考虑任意单射编码

$$C : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m,$$

最小距离为 $d = d_{\min}(C)$ 。

从每个码字删去固定的 $d - 1$ 个坐标。若两个删后的码字相同，那么原来至多只在这 $d - 1$ 个位置不同，与最小距离 d 矛盾。

因此删坐标后的映射仍然单射：

$$q^n \leq q^{m-d+1}, \quad \implies \quad \boxed{d_{\min} \leq m - n + 1.}$$

这不是 Reed-Solomon 码特有的性质；任何要编码 q^n 条消息的 q 元编码都必须满足这个上界。

所以最少需要多少冗余？

把上面的距离要求代入 Singleton 界：

$$k \text{ 个擦除} \Rightarrow d_{\min} \geq k + 1 \Rightarrow \boxed{m \geq n + k},$$

$$k \text{ 个错误} \Rightarrow d_{\min} \geq 2k + 1 \Rightarrow \boxed{m \geq n + 2k}.$$

因此， $n + 2k$ 不再只是“看起来够用”的猜测，而是**信息论意义下的最优目标**。

问题：多项式取值能恰好达到这个极限吗？

先看最小例子：2 个符号，1 个错误

两个符号写成一条直线

$$p(x) = a_0 + a_1x.$$

在 4 个互不相同的位置发送取值

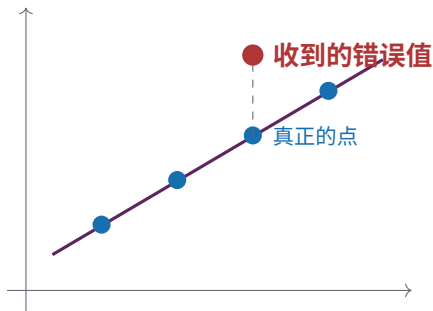
$$p(\alpha_1), p(\alpha_2), p(\alpha_3), p(\alpha_4),$$

其中最多 1 个收到的值被改坏。

4 个接收点中至少 3 个正确，
所以正确直线通过至少 3 个接收点。

所以正确直线唯一。

这就是 Reed-Solomon 纠错最简单的例子。



任何一条通过 3 个接收点的候选直线，至少经过 2 个正确点；
而两个正确点已经唯一决定原直线。

Reed-Solomon: 把直线例子推广到次数 $< n$

把消息编码成多项式

$$p(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1}.$$

选择两两不同的

$$\alpha_1, \dots, \alpha_N \in \mathbb{F}_q, \quad N = n + 2k,$$

并发送

$$C(p) = (p(\alpha_1), \dots, p(\alpha_N)).$$

最简单的做法是取 $q \geq N$ ，这样就有足够多互不相同的取值点。

上页是 $n = 2, k = 1$ ；现在的问题只剩下：

为什么任意次数 $< n$ 的两个消息多项式，都能保持足够大的距离？

为什么 Reed-Solomon 恰好达到 Singleton 界?

取两个不同消息多项式 $p \neq q$ 。两个码字相同的坐标满足

$$(p - q)(\alpha_i) = 0.$$

非零多项式 $p - q$ 的次数 $< n$ ，因此这样的坐标最多有 $n - 1$ 个。于是

$$d_{\min} \geq N - (n - 1) = 2k + 1.$$

另一方面，Singleton 界给出

$$d_{\min} \leq N - n + 1 = 2k + 1.$$

所以

$$d_{\min} = 2k + 1.$$

同一个关于根个数的事实，同时解释了拉格朗日插值的唯一性和 Reed-Solomon 码的距离。

唯一可解还不够：解码必须高效

收到 $N = n + 2k$ 个值后，朴素方法可以枚举

$$\binom{n + 2k}{n}$$

个 n 点子集：逐个插值，再检查它与多少个接收值一致。
一般参数下这是指数级的。

Berlekamp-Welch：用一个多项式表示错误位置，
最后把解码问题转化为一个**线性方程组**。

具体推导放在补充材料。

一次小结：两个事实，三种计算任务

事实 1：维数与唯一性

$$p \in \mathcal{P}_{<n}$$

n 个互异取值 $\implies$ 唯一决定 p .

因此：

- ▶ $m = n$ ：插值与精确重建；
- ▶ $m < n$ ：再加随机化 $\implies$ 秘密分享。

事实 2：根的个数与分离度

$$p \neq q, \quad \deg(p - q) < n \\ \implies p - q \text{ 最多有 } n - 1 \text{ 个根.}$$

因此：

- ▶ 多给一些取值，可以把不同码字分得更开；
- ▶ $m > n$ ：冗余 $\implies$ 纠删与纠错。

表示、信息、冗余本质上都在问：
有限个坐标究竟携带了多少关于整体的信息？

回到实数：代数上能恢复，不代表数值上逼近得好

在有限域里，只要横坐标互不相同，我们主要关心**能不能精确恢复**。

在 $\mathbb{R}$ 上，给定一个真实函数 f ，我们却还要问：

$P(x)$ 在采样点之间到底离 $f(x)$ 多远？

插值 (Interpolation) $\neq$ 逼近 (Approximation)

“在采样点上误差为 0” 并不意味着 “在整个区间上都很接近”。

拉格朗日插值误差公式

设 f 在包含 $x, x_1, \dots, x_n$ 的区间上 n 次连续可导, P 是在 $x_1, \dots, x_n$ 上的次数 $\leq n-1$ 插值多项式。

那么对每个 x , 存在某个 ξ_x 落在这些点所张成的最小区间内, 使得

$$f(x) - P(x) = \frac{f^{(n)}(\xi_x)}{n!} \prod_{j=1}^n (x - x_j).$$

它和泰勒余项 (Taylor remainder) 具有相同的结构: 都是 “高阶导数 $\times$ 几何因子”; 这里只是一个展开点被 n 个插值节点取代。

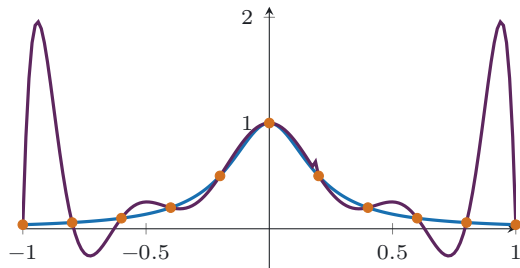
函数本身的难度

$$f^{(n)}$$

我们选择的采样点

$$\prod_{j=1}^n (x - x_j)$$

等距点为什么会出问题？Runge 现象



— $f(x) = 1/(1 + 25x^2)$ — 11 个等距点的次数 ≤ 10 插值

所有节点上都有

$$P(x_i) = f(x_i).$$

但是在端点附近， P 出现很大的振荡。

提高次数虽然增加了**自由度**，却不能保证整体误差变小。

这就是 Runge 现象。

选点本身就是算法设计的一部分

误差公式给出

$$|f(x) - P(x)| \leq \frac{\max_{\xi} |f^{(n)}(\xi)|}{n!} \left| \prod_{j=1}^n (x - x_j) \right|.$$

若我们可以自己选择两两不同的 $x_1, \dots, x_n$ ，自然会问

$$\min_{x_1, \dots, x_n \in [-1, 1]} \max_{x \in [-1, 1]} \left| \prod_{j=1}^n (x - x_j) \right|.$$

不同的节点在代数上都能唯一确定插值多项式；在数值上却远非等价。
这已经是一个极小极大（minimax）的算法设计问题。

预告：最优节点不是等距的，而会在区间两端更加密集。

本讲区分了两个不同的问题

精确重建：对象本来就属于模型类

$$p \in \mathcal{P}_{<n}.$$

n 个互异取值已经是完整坐标。

于是可以讨论：

- ▶ 刚好给够：拉格朗日插值；
- ▶ 故意少给：Shamir 秘密分享；
- ▶ 故意多给：Reed-Solomon 冗余与纠错。

数值逼近：多项式只是对真实函数的近似

$$P(x_i) = f(x_i).$$

节点上完全正确，节点之间仍可能很差。

误差取决于两部分：

$$\underbrace{f^{(n)}}_{\text{函数的光滑性}} \quad \text{和} \quad \underbrace{\prod_j (x - x_j)}_{\text{插值节点的位置}}$$

能唯一恢复，并不意味着**逼近得好**。 下一讲：怎样设计节点，让最坏误差尽量小？

下节课：Chebyshev 多项式与逼近

本讲留下的问题： 不同的插值节点都能唯一确定插值多项式，为什么数值表现却会相差很大？

Chebyshev 多项式与插值

我们将解这个极小极大（minimax）问题：

$$\min_{x_1, \dots, x_n \in [-1, 1]} \max_{x \in [-1, 1]} \left| \prod_{j=1}^n (x - x_j) \right|.$$

并进一步区分：

插值 与 函数逼近.

后续路线：函数逼近与正交多项式 → 最小二乘法与最佳平方逼近

Appendix

以下是进一步证明与连接，
不影响本讲主线，可按课堂时间选讲或课后阅读。

假设有 4 块磁盘：

A, B, C, D .

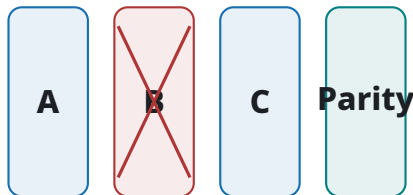
RAID 5 常用分布式奇偶校验 (distributed parity)，使单块磁盘故障后仍能恢复数据。抽象地写成

$$D = A \oplus B \oplus C.$$

若 B 丢失：

$$B = D \oplus A \oplus C.$$

更高等级的存储容错会使用更一般的**纠删码**（例如 **Reed-Solomon 码**），以容忍多处缺失或错误。



补充 背景：Reed-Solomon 在哪里出现？

Reed-Solomon 码以及更一般的纠删码，被广泛用于：

- ▶ 深空通信与数字图像传输；
- ▶ 分布式存储、RAID 和对象存储；
- ▶ 光盘、二维码等存储或通信系统；
- ▶ 混合检测（pooled testing）等利用“冗余观察”恢复信息的场景。

实际中的错误往往会成片出现，形成**突发错误（burst errors）**：相邻比特或符号可能一起受损。

因此工程实现通常还会配合交织（interleaving）、分块等技术，并采用更复杂的编码方案。

补充 Berlekamp-Welch: 错误位置对应一个“消失多项式”

发送 $p(\alpha_1), \dots, p(\alpha_N)$, 收到 $r_1, \dots, r_N$; 实际有 $e \leq k$ 个错误, 错误位置记为 $f_1, \dots, f_e$ 。

定义错误定位多项式 (error-locator polynomial)

$$E_0(x) = \prod_{j=1}^e (x - \alpha_{f_j}), \quad \deg E_0 \leq k.$$

令

$$Q_0(x) = p(x)E_0(x), \quad \deg Q_0 \leq n + k - 1.$$

那么对**每一个**接收位置 i 都有

$$Q_0(\alpha_i) = r_i E_0(\alpha_i).$$

- ▶ 错误位置: $E_0(\alpha_i) = 0$, 错误值被自动“消掉”;
- ▶ 正确位置: $r_i = p(\alpha_i)$ 。

补充 Berlekamp-Welch: 把 Q, E 直接当作未知量

我们不知道错误位置, 因此也不知道 E_0 。直接寻找多项式 Q, E , 满足

$$\deg Q \leq n + k - 1, \quad \deg E \leq k,$$

以及 $N = n + 2k$ 个**齐次线性方程**

$$\boxed{Q(\alpha_i) = r_i E(\alpha_i)}, \quad i = 1, \dots, N.$$

未知系数一共有

$$(n + k) + (k + 1) = N + 1.$$

所以 N 个齐次方程一定存在一个非零解 (Q, E) 。

而 E 不可能恒为 0: 否则 Q 在 N 个点为零, 但 $\deg Q < N$, 只能也恒为 0。

补充 Berlekamp-Welch: 为什么任意非零解都能恢复原来的 p ?

取上一页任意非零解 (Q, E) 。至少有

$$N - k = n + k$$

个位置是正确的; 在这些位置 $r_i = p(\alpha_i)$, 于是

$$Q(\alpha_i) - p(\alpha_i)E(\alpha_i) = 0.$$

但

$$\deg(Q - pE) \leq n + k - 1.$$

一个次数至多 $n + k - 1$ 的多项式却有至少 $n + k$ 个不同零点, 因此

$$Q \equiv pE.$$

所以恢复出的商唯一:

$$p = Q/E.$$

补充 Reed-Solomon: 也可以直接构造达到距离上界的码字

取任意次数 $< n$ 的多项式 p 和任意非零 $c \in \mathbb{F}_q$, 令

$$q(x) = p(x) + c \prod_{j=1}^{n-1} (x - \alpha_j).$$

于是

$$p(\alpha_j) = q(\alpha_j), \quad j = 1, \dots, n-1,$$

但 $p \neq q$ 。

因此在长度 $N = n + 2k$ 的 Reed-Solomon 码中, 这两个码字恰好在

$$N - (n - 1) = 2k + 1$$

个位置不同。

结合主线中的下界, 可再次得到

$$d_{\min} = 2k + 1.$$

固定要估计的点 x 。令

$$\omega(t) = \prod_{j=1}^n (t - x_j), \quad \lambda = \frac{f(x) - P(x)}{\omega(x)},$$

并定义

$$\phi(t) = f(t) - P(t) - \lambda\omega(t).$$

那么

$$\phi(x_1) = \cdots = \phi(x_n) = \phi(x) = 0.$$

反复应用 Rolle 定理 n 次，得到某个 ξ 使

$$\phi^{(n)}(\xi) = 0.$$

而

$$P^{(n)} = 0, \quad \omega^{(n)} = n!,$$

所以

$$f^{(n)}(\xi) = \lambda n!.$$

整理即得误差公式。

假设函数本身是低次多项式，并且次数上界已知。
如果某个算法在很多输入点上都有不低的概率给出正确值，那么可以：

1. 查询足够多的点；
2. 把这些答案看成一个可能含有错误的码字；
3. 通过纠错恢复整个低次多项式；
4. 从而在所有点上正确计算它。

低次数表示 + 冗余查询 $\Rightarrow$ 把“多数时候正确”提升为“处处正确”。

对矩阵 A, B ，把它们之间的仿射直线参数化为

$$g(t) = \text{perm}(A + tB).$$

如果矩阵大小为 $n \times n$ ，那么 $g(t)$ 是次数至多 n 的一元多项式。

因此，若我们能在足够多个随机选取的 t 上可靠地计算 $g(t)$ ，就可以利用插值或纠错恢复整个 g ，进而得到目标值。

这是 TCS 中常见的**随机自归约 (random self-reducibility)** 思想：
把一个最坏输入嵌入一条低次多项式曲线，再从随机点恢复回来。

插值要求

$$P(x_i) = f(x_i).$$

如果我们不要求精确经过指定点，而只希望**整体尽可能接近**，就进入函数逼近（Approximation）：

$$\|f\|_{\infty} := \sup_x |f(x)|,$$

$$\min_{\deg p \leq n} \|f - p\|_{\infty}.$$

Weierstrass 定理告诉我们：对闭区间上的连续函数和任意 $\varepsilon > 0$ ，都存在多项式 p 使 $\|f - p\|_{\infty} < \varepsilon$ 。

随机变量 X 的期望与方差：

$$\mathbb{E}[X] = \sum_x x \Pr[X = x], \quad \text{Var}(X) = \mathbb{E}[(X - \mathbb{E}X)^2].$$

Chebyshev 不等式：对任意 $t > 0$,

$$\Pr[|X - \mathbb{E}X| \geq t] \leq \frac{\text{Var}(X)}{t^2}.$$

下一步要用一个集中在 x 附近的随机变量，自动构造逼近 $f(x)$ 的多项式。

给定连续函数 $f : [0, 1] \rightarrow \mathbb{R}$ ，固定 $x \in [0, 1]$ 。

令

$$K \sim \text{Bin}(n, x),$$

即 K 是 n 次伯努利试验 (Bernoulli trials) 中的成功次数，每次成功的概率为 x 。

那么

$$\mathbb{E}[K] = nx, \quad \text{Var}(K) = nx(1 - x).$$

因此，随着 n 增大， K/n 会越来越集中在 x 附近。

如果 $K/n \approx x$ ，连续性就意味着 $f(K/n) \approx f(x)$ 。

因为

$$\Pr[K = i] = \binom{n}{i} x^i (1-x)^{n-i},$$

所以

$$\begin{aligned} B_n f(x) &:= \mathbb{E} \left[f\left(\frac{K}{n}\right) \right] \\ &= \sum_{i=0}^n f\left(\frac{i}{n}\right) \binom{n}{i} x^i (1-x)^{n-i}. \end{aligned}$$

定义 Bernstein 基函数

$$b_{n,i}(x) = \binom{n}{i} x^i (1-x)^{n-i}.$$

$B_n f$ 是次数至多 n 的多项式。

为什么 $B_n f$ 一致收敛到 f ?

对任意 $\varepsilon > 0$, 把事件分成两类:

$$|K/n - x| \leq \varepsilon \quad \text{和} \quad |K/n - x| > \varepsilon.$$

记 f 的连续模为

$$\omega_f(\varepsilon) = \sup_{|x-y| \leq \varepsilon} |f(x) - f(y)|.$$

那么

$$|f(x) - B_n f(x)| \leq \omega_f(\varepsilon) + 2 \|f\|_\infty \Pr[|K/n - x| > \varepsilon].$$

由 Chebyshev,

$$\Pr[|K/n - x| > \varepsilon] \leq \frac{x(1-x)}{n\varepsilon^2} \leq \frac{1}{4n\varepsilon^2}.$$

上一页直接给出，对任意 $x \in [0, 1]$ 、 $\varepsilon > 0$ ：

$$|f(x) - B_n f(x)| \leq \omega_f(\varepsilon) + \frac{\|f\|_\infty}{2n\varepsilon^2}.$$

这里

$$\omega_f(\varepsilon) = \sup_{|x-y| \leq \varepsilon} |f(x) - f(y)|.$$

概率集中控制 “采样点离 x 多远”；
一致连续性控制 “函数值因此变化多少”。

Weierstrass: 多项式在 $C([a, b])$ 中对 $\|\cdot\|_\infty$ 稠密。

更一般地, 在紧 Hausdorff 空间 X 上, 若实连续函数的子代数

$$A \subseteq C(X, \mathbb{R})$$

► 包含常数函数;

► 分离点: $x \neq y \Rightarrow \exists f \in A$ 使 $f(x) \neq f(y)$,

那么 A 在 $C(X, \mathbb{R})$ 中对 $\|\cdot\|_\infty$ 稠密。

神经网络的通用逼近 (Universal Approximation) 也可以看作“某个函数类是否稠密”的问题; 但具体定理取决于网络模型本身的结构, 不能把所有网络都直接套用 Stone-Weierstrass 定理。