NANJING UNIVERSITY · 计算机学院

计算方法

Computational Methods

Lecture 2：插值——有限观测如何决定整体?

Interpolation: Reconstruction, Redundancy, and Approximation

刘景铖

南京大学 · 计算机学院

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：

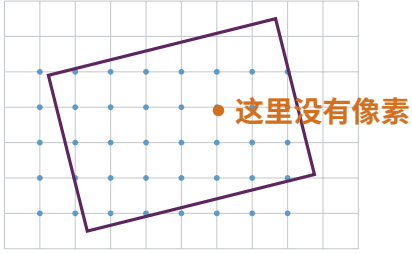
插值（Interpolation）

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

只看到一些**不完整的观测值**，

怎样推理出**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

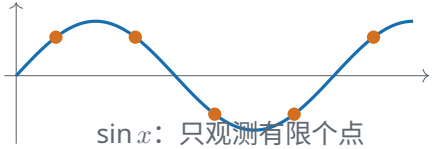
$$I(117.32, 48.71).$$

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？

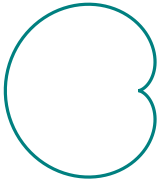
插值可以在很多地方出现

函数值之间



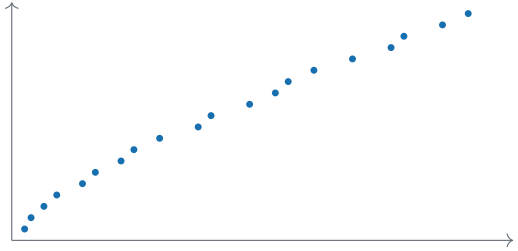
$\sin x$：只观测有限个点

曲线与形状



心形线（Cardioid）

离散统计量之间



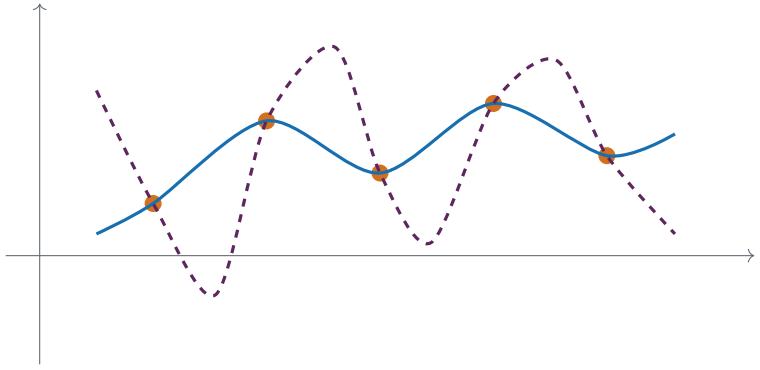
素数计数函数 $\pi(x)$

换坐标后，结构可能更简单

$$y = x^3 \quad \Longrightarrow \quad \log y = 3 \log x.$$

双对数（log–log）坐标下变成直线。

有限观测并不能唯一决定函数



同样的有限数据点，可以通过无穷多个函数。

插值之前，必须先选择一个**结构假设 / 表示类**。

先区分两件事：精确重建 vs. 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做。

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：采样点之间，P 离 f 有多远?

连续性只排除跳变；导数有界 / 更高阶光滑性才给出更强的定量误差控制。

本讲先把**精确重建**讲透；最后再回到**数值逼近**，这时才真正需要误差与正则性。

为什么多项式同时适合这两个问题？

作为精确模型：有限维、代数透明

$$p(x) = a_0 + a_1 x + \cdots + a_n x^n.$$

- $n+1$ 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

- Taylor 展开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；
- 物理电路、神经网络当然也能表示函数；多项式是最透明的起点。

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，两种坐标；插值就是换坐标

次数 $< n$ 的多项式是一个 n 维对象。

系数坐标

$$\mathbf{a} = (a_0, \dots, a_{n-1}),$$

$$p(x) = \sum_{j=0}^{n-1} a_j x^j.$$

$\Longleftrightarrow$

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，坐标变换就是 Vandermonde 线性系统：

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

Lagrange：直接选择一组**适合“取值”测量的基**，不必先求系数。

Lagrange 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

$$\boxed{P(x_i) = y_i \qquad i = 1, \dots, n.}$$

先构造存在性，再证明唯一性。

Lagrange 基函数：只把一个观测值“打开”

想找 L_k 使得

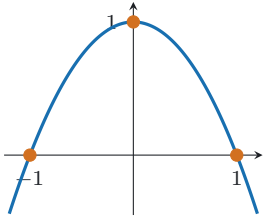
$$L_k(x_k) = 1, \qquad L_k(x_j) = 0 \quad (j \neq k).$$

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

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



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

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

就有

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

每个 L_k 负责“打开”一个观测值，其他位置全部关掉。

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 Lagrange 插值找出通过

$$(1,1), \qquad (2,1)$$

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 Lagrange 插值找出通过

$$(1,1), \qquad (2,1)$$

的次数至多 1 的多项式。

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

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

$$\boxed{O(n^2)}$$

次算术运算。

表示 / 算法	构造 / 预处理	之后单点求值
直接 Lagrange 公式	—	$O(n^2)$
Newton 形式（差商）	$O(n^2)$	$O(n)$
Lagrange 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。对某个观察者而言，若拿到 m 个互异取值：

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

$m < n$	$m = n$	$m > n$
信息不足，存在歧义	刚好是坐标	加入冗余
+ 随机化	唯一恢复	容忍缺失 / 错误
⇓	⇓	⇓
秘密分享	**插值**	**纠删码**

少给、刚好、多给不是三个技巧，而是同一个表示映射的三种信息结构。

背景 11 位科学家与 462 把锁

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：

$$\binom{11}{5} = 462 \quad \text{把锁,} \qquad \binom{10}{5} = 252 \quad \text{把钥匙/人.}$$

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：

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

所有加、减、乘、除都模 q 进行（除数非零）：

$$p(x) = a_0 + a_1 x + \cdots + a_k x^k, \qquad a_i, x \in \mathbb{F}_q.$$

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 $\le k$ 的多项式仍由 $k+1$ 个点唯一决定。

这里的有限域算术本身是精确的；后面的“错误”来自通信 / 存储符号被改坏，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

$$\boxed{p(x) = s + a_1 x + \cdots + a_5 x^5.}$$

给第 i 个人一份份额：

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

且所有 x_i 互不相同。

任意 6 人拿到 6 个点，Lagrange 插值恢复 p，进而恢复

$$\boxed{s = p(0).}$$

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 $(0, s')$。

6 个点唯一决定一个次数至多 5 的多项式：

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

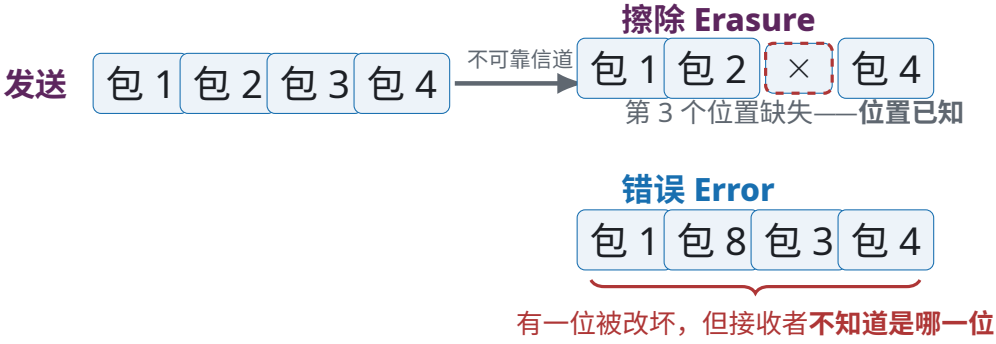
所以，对**每一个** s'，都恰好有一组

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

与已经看到的份额相容。

随机系数 $(a_1, \dots, a_5)$ 均匀选择，因此
无论真实秘密是哪一个，这 5 份份额的联合分布都完全一样。

背景 从重传到纠错：两种故障



TCP/IP 常用**重传**来处理丢包；但重传可能昂贵、延迟很高，甚至根本不可能。

能不能预先加入冗余，让接收端直接恢复原消息？

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

$$p(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1},$$

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；
Lagrange 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 Hamming 距离

$$\boxed{d(s,t) = \#\{i : s_i \neq t_i\}}, \qquad \boxed{d_{\min}(C) = \min_{u \neq v} d(C(u), C(v))}.$$

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

主动让不同合法表示之间留出足够大的间隔。

关注的是合法对象之间的分离度。

鲁棒性的一种普遍策略：给合法对象之间留出“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：

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

擦除的位置已经告诉了我们。

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：

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

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 bound：距离与冗余之间的普适极限

考虑任意单射编码

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

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：

$$q^n \le q^{m-d+1}, \qquad \Longrightarrow \qquad \boxed{d_{\min} \le m - n + 1.}$$

这不是 Reed–Solomon 的性质，而是任何装下 q^n 条消息的 q 元编码都逃不掉的限制。

所以最少需要多少冗余?

把距离要求代入 Singleton：

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

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

现在 $n+2k$ 不再只是“看起来够用”的猜测，而是一个**信息论上的最优目标**。

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

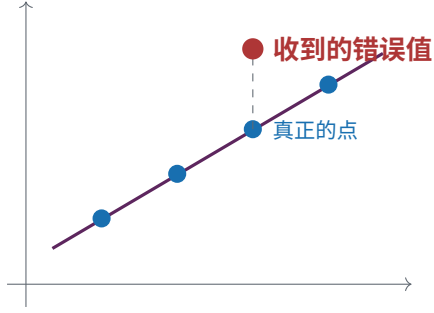
两个符号写成一条直线

$$p(x) = a_0 + a_1 x.$$

发送 4 个互异位置上的取值

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

其中最多 1 个收到的值被改坏。



任何一条通过 3 个接收点的直线，其中至少有 2 个点是真的；
而两个正确点已经唯一决定原直线。

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。

这就是一般 Reed–Solomon 纠错的最低维原型。

Reed–Solomon：把直线例子推广到次数 $< n$

消息写成

$$p(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}.$$

选择两两不同的

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

并发送

$$\boxed{C(p) = (p(\alpha_1), \ldots, p(\alpha_N)).}$$

最简单只需 $q \geq N$，保证有足够多互异取值点。

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 极限?

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

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

而 Singleton 同时给出

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

所以

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

同一个根计数事实：Lagrange 唯一性 $\longleftrightarrow$ Reed–Solomon 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，再检查与多少收到的点一致。
一般参数下这是指数级的。

Berlekamp–Welch：把“哪些位置错了”也编码成一个多项式，
最后转化为一个**线性方程组**。

具体推导放在补充材料。

一次小结：同两个事实，三种计算任务

事实 1：维数 / 唯一性

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

n 个互异取值 $\Longrightarrow$ 唯一决定p.

因此：

- $m = n$：插值 / 精确重建；
- $m < n$：再加随机化 $\Rightarrow$ 秘密分享。

事实 2：根计数 / 分离

$$p \neq q, \qquad \deg(p - q) < n$$

$\Longrightarrow$ $p - q$ 最多有$n - 1$ 个根.

因此：

- 多给取值可以制造码字间距；
- $m > n$：冗余 $\Rightarrow$ 纠删 / 纠错。

表示、信息、冗余本质上都在问：
有限坐标里究竟装了多少关于整体对象的信息？

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，只要横坐标不同，我们主要关心**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）

“在训练点上误差为 0”并不自动意味着“在所有位置都接近”。

Lagrange 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 $\le n-1$ 插值多项式。
那么对每个 x，存在某个 ξ_x 位于这些点的凸包中，使得

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

与 Taylor 余项是同一种“高阶导数 $\times$ 几何因子”的结构；这里只是一个展开点被 n 个插值节点取代。

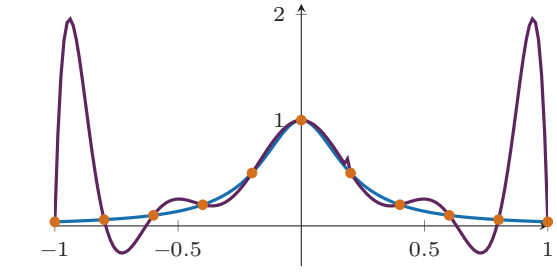
函数本身的难度

$f^{(n)}$

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。

增加次数意味着**更多自由度**，却不保证整体误差变小。

这才是 Runge phenomenon。

选点本身就是算法设计的一部分

误差公式给出

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

若我们可以自己选择两两不同的 $x_1,\dots,x_n$，自然会问

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

不同节点在代数上都能唯一插值；在数值上却远非等价。

这已经是一个极小极大（minimax）算法设计问题。

预告：最优节点不会等距，而会向区间两端更密集。

本讲真正区分了两种“插值问题”

精确重建：对象本来就在模型类中

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

n 个互异取值已经是完整坐标。

于是可以讨论：

- 刚好给够：Lagrange 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：模型只是近似真实函数

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

节点上完全正确，节点之间仍可能很差。

误差取决于两部分：

$$\underbrace{f^{(n)}}_{\text{函数的正则性}} \quad \text{和} \quad \underbrace{\prod_j (x - x_j)}_{\text{节点的几何}}.$$

唯一可恢复 $\neq$ **数值逼近得好**.

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： 不同插值节点在代数上都有效，为什么数值表现差异巨大？

Chebyshev 多项式与插值

我们将解这个极小极大（minimax）问题：

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

并正式区分：

插值 vs. 函数逼近.

后续路线：函数逼近与正交多项式 $\longrightarrow$ 最小二乘法与最佳平方逼近

Appendix

以下内容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。

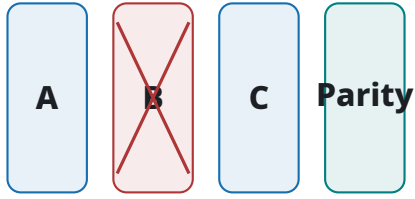
补充 背景：存储中的同一个问题——RAID 与冗余

假设有 4 块磁盘：

$$A,\ B,\ C,\ D.$$

RAID 5 的典型思路之一是用分布式校验（parity）使单盘故障后可重建，例如抽象地写成

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



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 / Reed–Solomon 型编码**，以容忍多处缺失或错误。

补充 背景：Reed–Solomon 在哪里出现?

历史上和工程中，Reed–Solomon / 更一般的纠删码被用于：

- 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 / 通信系统；
- 基于编码的组合式检测（pooling）等利用“冗余观测”恢复信息的场景。

真实错误往往具有**突发（burst）**结构：相邻比特 / 符号可能一起受损。所以工程实现通常还会配合交织（interleaving）、分块和更复杂的码设计。

补充 Berlekamp–Welch：真错误对应一个“消失多项式”

发送 $p(\alpha_1), \dots, p(\alpha_N)$，收到 $r_1, \dots, r_N$；实际有 $e \le k$ 个错误，位置为 $f_1, \dots, f_e$。
定义真正的错误定位多项式（error-locating polynomial）

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

令

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

那么对**每一个**收到的位置 i 都有

$$\boxed{Q_0(\alpha_i) = r_i E_0(\alpha_i)}.$$

- 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

$$\deg Q \le n + k - 1, \qquad \deg E \le k,$$

以及 $N = n + 2k$ 个**齐次线性方程**

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

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给出原来的 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

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

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：

$$\boxed{p = Q/E.}$$

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

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

但 $p \neq q$。
因此在长度 $N = n + 2k$ 的 Reed–Solomon 取值码中，这两个码字最多只在

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

个位置不同。
结合主线中的下界，可再次得到

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

补充 误差公式：为什么会出现 n 阶导数?

固定要估计的点 x_0。令

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

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

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

所以

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

整理即得误差公式。

补充 Polynomial interpolation and boosting

假设一个函数可以被一个已知低次数的多项式表示。
如果有人能够在很多输入点上以“不错的成功率”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取值码字；
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 $\Longrightarrow$ 把“多数时候正确”提升成“处处正确”。

补充 课外阅读：Permanent 的随机自归约

对矩阵 A, B，沿一条仿射直线看

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

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，若我们能在足够多随机 t 上可靠地计算 $g(t)$，就可以利用插值 / 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。

补充 如果不要求经过每个点：一致逼近

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）：

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

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

Weierstrass 定理告诉我们：对闭区间上的连续函数，随着次数增加，多项式可以在 ∞-范数下任意逼近。

补充 概率论回顾：Bernstein 构造要用什么？

随机变量 X 的期望与方差：

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

Chebyshev 不等式：对任意 $t > 0$，

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

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。

补充 Bernstein polynomial：用二项分布取样

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

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

即 K 是 n 次 Bernoulli(x) 试验中的成功次数。
那么

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

因此 K/n 会随 n 增大而集中在 x 附近。

如果 $K/n \approx x$，那么连续性暗示 $f(K/n) \approx f(x)$.

补充 Bernstein polynomial 真的是多项式

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。

补充 为什么 $B_n f$ 一致收敛到 f?

对任意 $\varepsilon > 0$，把事件分成两类：

$$|K/n - x| \le \varepsilon \qquad \text{和} \qquad |K/n - x| > \varepsilon.$$

记连续函数的模为

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

那么

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

由 Chebyshev，

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

补充 Bernstein 的定量误差

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

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

这里

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

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”。

补充 Beyond Polynomial：Stone–Weierstrass

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）也属于“一个函数类是否稠密”的同类问题；具体定理需要检查相应模型类的结构，不能简单把所有网络都直接套进 Stone–Weierstrass。