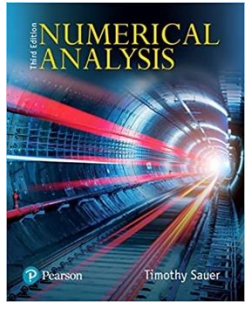
计算方法

刘景铖

计算机软件新技术国家重点实验室
南京大学

课程基本信息

- 教师：刘景铖
- Email：liu@nju.edu.cn
- Office hour：周二 16:00-18:00，计算机系516
- 课程主页：https://tcs.nju.edu.cn/wiki
- 数值分析 （Numerical Analysis） （原书第2版） Timothy Sauer, 机械工业出版社



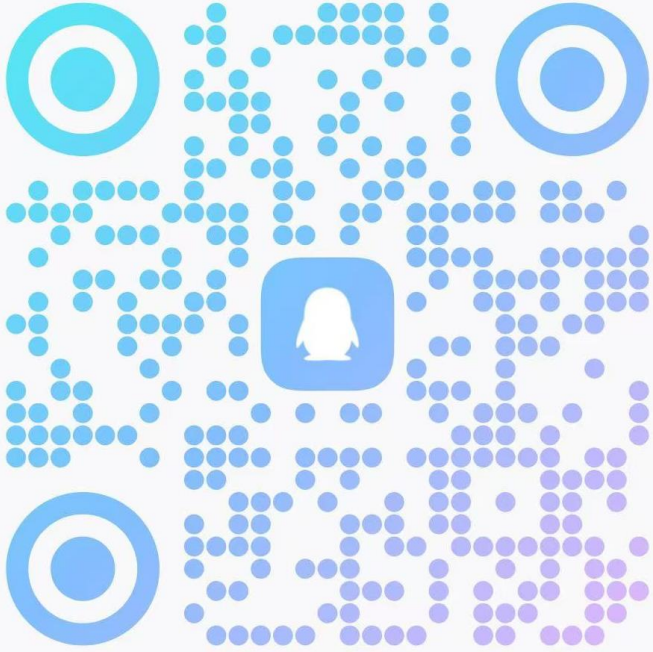
参考：

- Numerical Algorithms: Methods for Computer Vision, Machine Learning, and Graphics. Justin Solomon. CRC Press
- Lx=b,拉普拉斯方程求解以及它的应用（Lx=b, Laplacian Solver and Their Algorithmic Applications） Nisheeth K. Vishnoi

课程QQ群

计算方法 2026

群号: 522312428

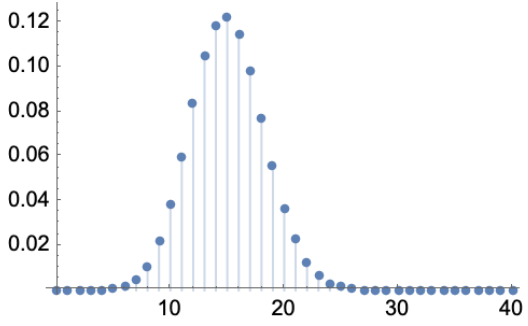


- 作业每两周发布一次
- 第一次作业今天将会发布
- 作业Email: nm_nju_2026@163.com
- QQ群可以讨论课程相关的问题
- 作业可以小组讨论（<=3人）
  - 清晰标明合作者，及各自的贡献
  - 使用了的参考资料必须注明
  - 但写作必须由自己独立完成，不可照抄
  - 类似地，别人提供的想法也必须注明
- 分辨合作与作弊之间的区别
  - 参与讨论之前，请先花时间自己进行思考
  - 避免参与你不能提供贡献的讨论
  - 作弊不仅让你丧失一次学习的机会
  - 还会影响你以后的自信心
- 对学术不诚信的行为零容忍
- 迟交作业需要提前预约

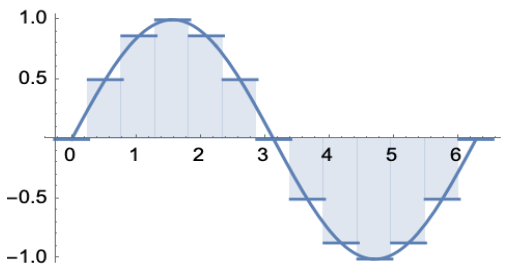
对作弊的认定参照http://www.acm.org/publications/policies/plagiarism_policy

回顾

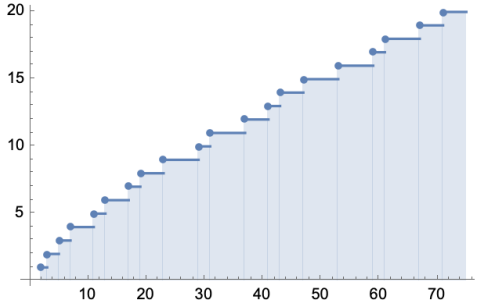
- **方程求根**：给定f，求x使得$f(x) = 0$
  - 连续函数：**二分法**
  - 1-Lipschitz: 转换为不动点迭代方程后使用不动点迭代法
  - **根的敏感性**
- 这节课：
  - **插**值(Interpolation)：给定一些不完整的观察值，想要推理出全景图像



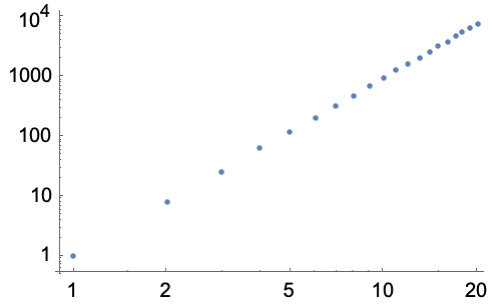
插值是为了什么？



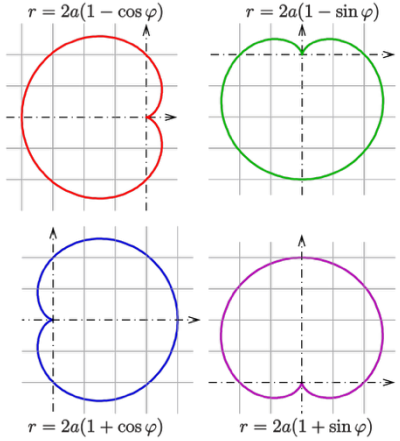
$\sin(x)$



素数的个数



Log-Log plot of $f(x) = x^3$



心形线(Cardioid)

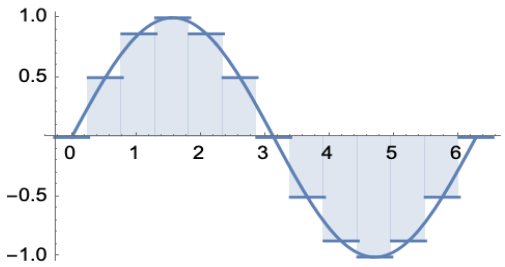
Source:Wikipedia CC-BY-SA 4.0

旋转照片

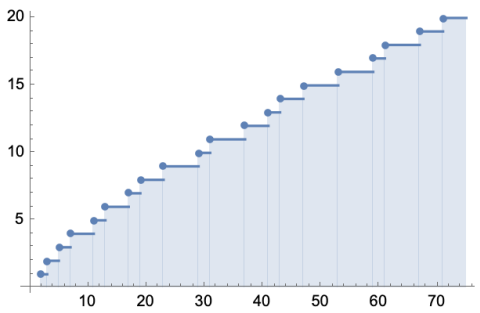
[旋转360度后照片一样吗？]

$$y' = \ln f(x) = 3 \ln x$$
$$x' = \ln x$$
$$y' = 3x'$$

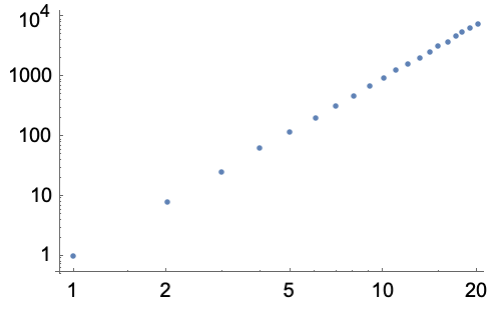
插值是为了什么？



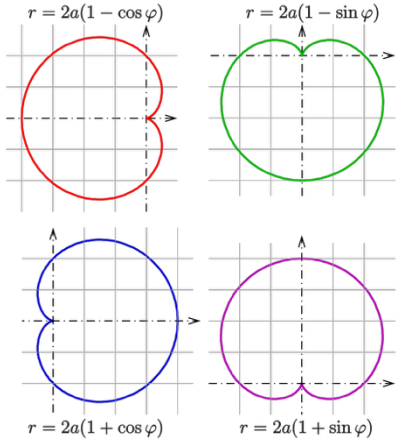
$\sin(x)$



素数的个数

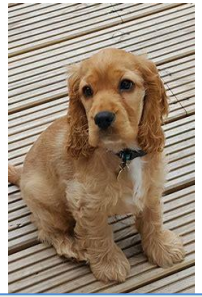


Log-Log plot of $f(x) = x^3$



心形线(Cardioid)

Source:Wikipedia CC-BY-SA 4.0



旋转照片

[旋转360度后照片一样吗？]

$$y' = \ln f(x) = 3 \ln x$$
$$x' = \ln x$$
$$y' = 3x'$$

为什么想要连续性？好看？

插值的连续性

- **回**顾一下不连续函数的**数值稳定性：**
- 给定函数f
- 输入： $x + \Delta x$
- 计算：$f(x)$

$$f(x + \Delta x) - f(x) \approx \Delta x \, f'(x)$$

在函数不连续性的点上，数值可能非常不稳定！

如果要更进一步的平滑处理，往往需要更高阶的导数

用什么插值？

- **多**项式定义：

$$p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n$$

- $a_0, a_1, \ldots, a_n$**称**为多项式的系数
- **若**$a_n \neq 0$, 则称n为多项式的次数
- **常用表达形式**（代数基本定理）：

$$p(x) = a_n(x - r_1)(x - r_2) \ldots (x - r_n)$$

- **一般来**说$r_1, r_2, \ldots, r_n$**可能是复数**，复数根总是成对出现的

- 为什么使用多项式？如果使用物理电路插值呢？神经网络呢？
  - 回顾Taylor展开，也是多项式
  - Taylor展开一般只关注一个点上的近似多项式
  - Weierstrass **定理**: 对于连续函数，**多**项式可以任意地逼近
  - 课外阅读：Stone-Weierstrass 定理和神经网络的Universal approximation

Lagrange Interpolation 拉格朗日插值

- **任意**给定n**个不同的数据点**$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- 满足$p(x_i) = y_i, \forall i$**的**，次数为$n-1$次（或更低次）多项式有且仅有一个
- 存在性的构造证明：
  - 固定k，试找出$L_k(x_k) = 1, L_k(x_j) = 0, \forall j \neq k$

    $L_k(x) = A(x - x_1) \dots (x - x_{k-1})(x - x_{k+1}) \dots (x - x_n)$
  - **注意到**$L_k(x_j) = 0, \forall j \neq k$
  - 而$L_k(x_k) = A(x_k - x_1) \dots (x_k - x_{k-1})(x_k - x_{k+1}) \dots (x_k - x_n)$
  - 可令$\mathrm{A} = \frac{1}{(x_k - x_1) \dots (x_k - x_{k-1})(x_k - x_{k+1}) \dots (x_k - x_n)}$

Lagrange Interpolation

存在性的构造证明：

- 对任意k，找出$L_k(x_k)=1, L_k(x_j)=0, \forall j \neq k$

$$L_k(x)=A(x-x_1)\dots(x-x_{k-1})(x-x_{k+1})..(x-x_n)$$

- 把它们加起来：

$$P_{n-1}(x)=y_1L_1(x)+\cdots+y_nL_n(x)$$

- 则 $P_{n-1}(x_k)=0+\cdots+0+y_kL_k(x_k)+0+\cdots+0=y_k$
- **注意多**项式的次数**最多**为$n-1$**次**
- **思考**：是否可能小于$n-1$次？

- **练习**：尝试用Lagrange interpolation找出通过$(1,1),(2,1)$这两点**的多**项式，并化简

Lagrange Interpolation

唯一性证明：反证法。

假设有$P(x)$和$Q(x)$两个多项式，至多$n-1$次且都通过给定的n个点

- 考虑$H(x)=P(x)-Q(x)$，H的次数最多为$n-1$次
- $H(x_1)=H(x_2)=H(x_3)=\cdots=H(x_n)=0$
- **H有n个零点**
- **代数基本定理**说，$n-1$次多项式有且仅有$n-1$个零点，除非H是零多项式
- **H必**须恒等于零
- $P(x)$恒等于$Q(x)$

- 所以这样的多项式只有一个

Lagrange Interpolation

拉格朗日插值：

- 对任意k，找出$L_k(x_k) = 1, L_k(x_j) = 0, \forall j \neq k$

$$L_k(x) = A_k(x - x_1) \dots (x - x_{k-1})(x - x_{k+1}) \dots (x - x_n)$$

$$令 A_k = \frac{1}{(x_k - x_1) \dots (x_k - x_{k-1})(x_k - x_{k+1}) \dots (x_k - x_n)}$$

- 把它们加起来：

$$P_{n-1}(x) = y_1 L_1(x) + \cdots + y_n L_n(x)$$

- **思考**：计算复杂性？
  - 每给定一个新的x,需要多少步算术运算？
  - 每个$L_k(x)$ 需要$O(n)$，共$O(n)$个$L_k(x)$，所以需要$O(n^2)$次
  - **课外阅读**：加速至$O(n)$次运算:
    - 通过Newton basis（牛顿差商形式）
    - 或者Barycentric Lagrange interpolation（利用上除法）

Lagrange Interpolation Error Analysis

拉格朗日插值的误差分析

- 误差公式：对连续n次可导函数f，在$x_1, \ldots, x_n$上插值

$$f(x) - P(x) = \frac{(x - x_1)(x - x_2)\cdots(x - x_n)}{n!} f^{(n)}(c),$$

- **与Taylor展开的余**项相似，但是结论更强

证明思路：

- 不断地在newton basis下面应用罗尔定理
- 一阶可导函数的两个零点之间，存在一个导数为零的点
- 详见书本第三章

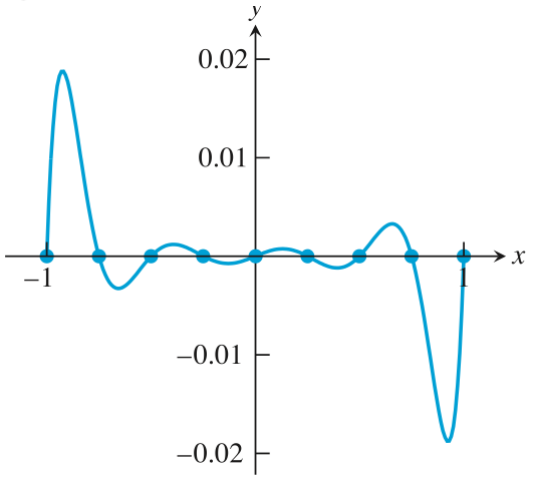
Lagrange Interpolation Error Analysis（选讲）

对连续n次可导函数f，在$x_1, \dots, x_n$上拉格朗日插值的误差

$$f(x) - P(x) = \frac{(x - x_1)(x - x_2)\cdots(x - x_n)}{n!} f^{(n)}(c),$$

证明：令$q(t)$为$x_1, \dots, x_n, x$上插值的多项式

- 则$q(t) = P(t) + \lambda \prod_{j=1}^{n}(t - x_j)$，其中$\lambda = \frac{f(x)-P(x)}{\prod_{j=1}^{n}(x-x_j)}$
- 考虑$\phi(t) \coloneqq f(t) - q(t)$，则$\phi(t)$在$n+1$个点处为零
- 由Rolle定理，$\phi'(t)$在n个点处为零
- 反复应用Rolle定理，$\phi^{(n)}(t)$在区间上某个点为零
- 在该点$\phi^{(n)}(c) = 0 \Rightarrow f^{(n)}(c) = q^{(n)}(c) = \lambda n!$
- 整理即得误差公式



使用等距采样插值时的误差函数

如何选点$x_1, \dots, x_n$，才能最小化误差？插值的表现可否媲美函数逼近？

- **采用更多靠近**边界的点

龙格现象

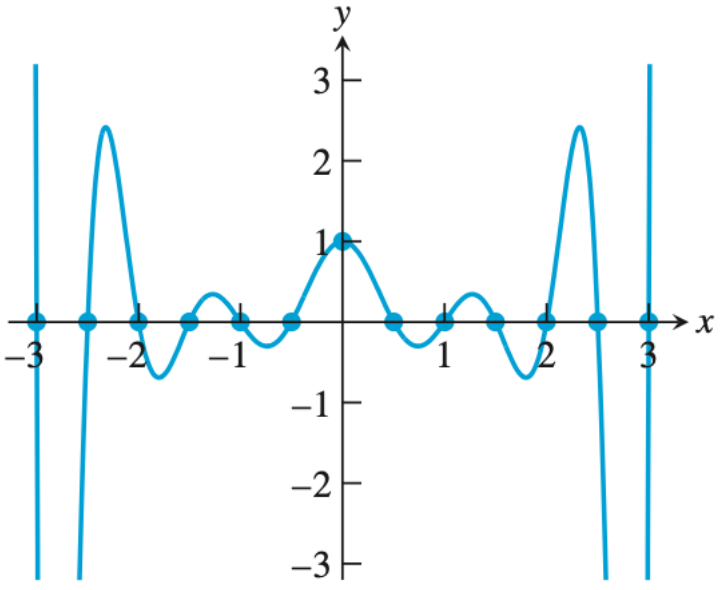


Figure 3.5 Interpolation of triangular bump function. The interpolating polynomial wiggles much more than the input data points.

高次多项式插值固然可以让得到的函数更加smooth，但是也更容易出现龙格现象

插值的应用 I：分享秘密

多项式插值的原理看似简单，但简单的原理也可以有非凡的应用

11个科学家参与了一个高度保密的项目。科学家们希望把档案锁起来，保密的要求是：当且仅当其中6位或者更多的科学家在场的时候，才能打开所有的锁（进而读取档案）。

问题：最少需要多少把锁？每个科学家最少需要同时携带多少把钥匙？

组合问题答案：$\binom{11}{5} = 462$把锁，每个科学家需要带$\binom{10}{5} = 252$把钥匙！

试想，如果有更多的科学家呢？

Adi Shamir在1979年提出了一个全新的解法：使用多项式插值！

插值的应用 I：分享秘密

多项式插值的原理看似简单，但是简单的原理也可以有非凡的应用

- 11个科学家参与了一个高度保密的项目。科学家们希望把档案锁起来，保密的要求是：当且仅其中6位或者更多的科学家在场的时候，才能打开所有的锁（进而读取档案）。
- 问题：最少需要多少把锁？每个科学家最少需要同时携带多少把钥匙？

- Adi Shamir在1979年提出了全新的解法：使用多项式插值！
  - **把密**码/密文藏进一个5次多项式p里面
  - **每个人手握多**项式的一个点值$(x_i, p(x_i))$，点值互不相同
  - **任意**6个人（或更多），就有了6个数据点
  - 通过拉格朗日插值，即可唯一地还原多项式p

注：实际的算法使用的是有限域而非实数域；
需要确保任意5个人的点值，在信息论意义下面，跟一个点值都没有是一样的。

有限域多项式

$$p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_k x^k \ (mod\ q)$$
$$p: \{0, \ldots, q-1\} \rightarrow \{0, \ldots, q-1\}$$
$$a_i \in \{0, \ldots, q-1\}$$

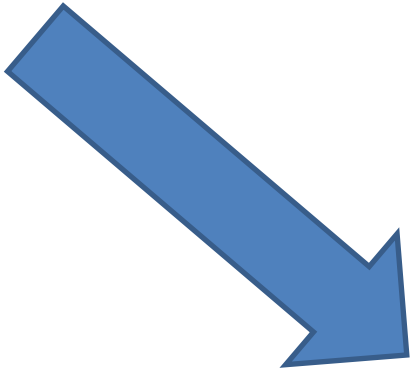
- **系数表示**
- **取**值表示: $p(1), p(2), \ldots, p(k), p(k+1)$
  - **插**值: Lagrange, **或解**线性方程组

$$L(x) := \sum_{j=0}^{k} y_j \prod_{0 \le m \le k, m \ne j} \frac{x - x_m}{x_j - x_m}.$$

  - 经过$k+1$**个点的次数**$\le k$ **的多**项式有且只有一个
  - **取模下的除法: 乘法的逆**

Error correcting code I: Erasure codes

Two types of errors
- Erasure errors: packet loss
- General errors: packet corruption



TCP/IP: resend packets, change data rate etc.
But this can be wasteful!
Can we reconstruct packets?

Another example: RAID storage levels

- Imagine you have 4 disks, each of 4TB
- To tolerate 1 disk failure, a common practice is to use RAID 5
- You have 3 disks of available storage, and 1 disk for parity

- Say you lose B, you can rebuild $B = D \oplus A \oplus C$

- However, at current unrecoverable read error (URE) rate of 1e-14, the success rate of rebuilding is less than 40%
  - RAID 6: One common implementation is via Reed-Solomon codes
- Cloud storage providers are already using more general ECCs

Erasure codes

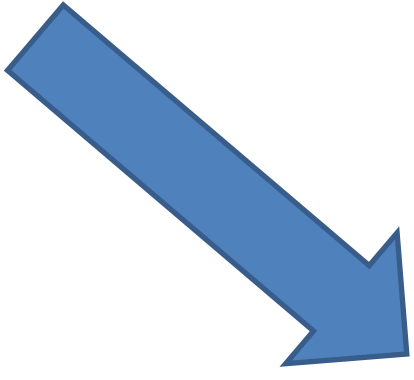
Imagine you want to send n packets through a lossy channel
Lossy channel: can lose up to k packets
Question: can you send $n + k$ packets and recover the message?
Hint: what do we know about polynomials?

Answer: Yes!
Any n points uniquely determine a degree $\leq n - 1$ polynomial.

Say we are given messages $a_0, a_1, \ldots, a_{n-1}$
Consider $p(x) = a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 \ (mod\ q)$
Send $p(1), p(2), \ldots, p(n+k)$

Any n packets allow us to recover all. This is optimal!

Error correcting code II: General errors



Imagine we want to send 2 packets
Channel: corrupt at most 1 packet, but can be any one

Reed-Solomon codes

Imagine we want to send 2 packets
Channel: corrupts at most 1 packet, but can be any one

2 points -> a line?
What if we send other points on the line as well?

Say we send 4 points in total.
How do we determine which one is corrupted?

At most 1 error -> there is a line passing through 3 points
If we found such a line, it must be the correct line:

- At most 1 error, so 2 out of the 3 points must be correct
- 2 points determine a unique line, which must be the correct line

Reed-Solomon codes

Imagine we want to send n packets
Channel: corrupts at most k packet, but can be any one

Reed-Solomon codes
Say we are given messages $a_0, a_1, \dots, a_{n-1}$
Consider $p(x) = a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 \ (mod\ q)$
Send $p(1), p(2), \dots, p(n+2k)$

At most k errors -> ∃ a poly. of deg $n-1$ passing through $n+k$ points
If we found such a poly., it must be the correct poly.:

- At most k error, so n out of the $n+k$ points must be correct
- There is a unique poly. of deg $n-1$ going through n points
- So, if our poly. goes through them, it's the correct one

Reed-Solomon codes

Many applications:

- Voyager I for colored photos
- HDFS-RAID in Facebook
- GFS II in Google
- Covid-19 Pool testing
- …

Q: Why mod q?

- Numerical instability or genuine error

Q: Choice of q? Need $q > n + 2k$.

- Not ideal, but OK in practice.
- Often times, errors are bursty. If a bit is corrupted, all nearby symbols are often unreliable

Efficient decoding:

- $\binom{n+2k}{n}$ interpolations to try? Brute-force would take exponential time in general.
- Berlekamp-Welch: one polynomial interpolation suffices – error locating polynomial

课外阅读：Berlekamp-Welch算法

Sent: $p(1), p(2), \ldots, p(n), p(n+1), \ldots, p(n+2k)$

Received: $r_1, r_2, \ldots, r_n, r_{n+1}, \ldots, r_{n+2k}$

Berlekamp Welch Theorem

There is a polynomial $Q(x)$ of deg. $n+k-1$, $E(x)$ of deg. k, such that $p(x) = \frac{Q(x)}{E(x)}$, and $\forall i, Q(i) = r_i E(i)$. In other words,

$$\forall i, p(i)E(i) = r_i E(i)$$

To understand this, let $Q(x) = b_{n+k-1}x^{n+k-1} + \cdots + b_0$, and $E(x) = x^k + e_{k-1}x^{k-1} + \cdots + e_0$.

Total number of unknowns $= n + 2k$.

For each received value r_i, there is a linear equation $Q(i) = r_i E(i)$.

Total number of linear equations $= n + 2k$.

课外阅读：Error locating polynomial

Sent: $p(1), p(2), \ldots, p(n), p(n+1), \ldots, p(n+2k)$
Received: $r_1, r_2, \ldots, r_n, r_{n+1}, \ldots, r_{n+2k}$

Ideally $p(i) = r_i$

Say the location of errors are $f_1, \ldots, f_k$
Consider the error locating polynomial $E(x) = (x - f_1) \ldots (x - f_k)$

Instead we ask that $p(i)E(i) = r_i E(i)$

Case 1: location i is an error, 0=0
Case 2: location i is correct, $p(i) = r_i$

Let $Q(x) = p(x)E(x)$. Then, the system of linear equations have a solution! (Consistency)
Uniqueness of $p(x)$: Say $Q(x), E(x)$ is one solution, and $Q'(x), E'(x)$ is another solution.
Notice that $Q(x)E'(x) = Q'(x)E(x)$, because they agree on n+2k points, and they are of degree at most n+2k.
So, $p(x) = \frac{Q(x)}{E(x)} = \frac{Q'(x)}{E'(x)}$ is unique.

Note that $Q(x), E(x)$ may not be unique themselves. Consider the case where there are no errors.

ECCs and distance properties

Hamming distance $d(s,t) \coloneqq \sum_i 1(s_i \neq t_i)$

Minimum distance of a code:= $\min_{m \neq m'} d(c(m), c(m'))$

To handle k erasure errors, need minimum distance $k+1$

To handle k general errors, need minimum distance $> 2k$

Theorem: The Reed-Solomon code mapping n messages to a codeword of $n+2k$ messages has minimum distance $2k+1$

Distance properties

Claim 1: The Reed-Solomon code mapping n messages to a codeword of $n + 2k$ messages has minimum distance **at most** $2k + 1$

It suffices to show that, $\exists m \neq m',\ d(c(m), c(m')) \leq 2k + 1$

Say we choose codeword $c_0, c_1, \ldots, c_{n-2}, c_{n-1}$, we interpolate to get $p_1(x)$
Choose another codeword $c_0, c_1, \ldots, c_{n-2}, e_{n-1}$, we interpolate to get $p_2(x)$
$p_1(x)$ agrees with $p_2(x)$ on at least $n - 1$ points
$p_1(x)$ and $p_2(x)$ encode messages $m \neq m'$ in their coefficients

Reed-Solomon code will send $c(m)$, $c(m')$ that agrees on at least $n - 1$ points
Therefore, $d(c(m), c(m')) \leq 2k + 1$

Side note: this is essentially the Singleton/Projection bound.

Distance properties

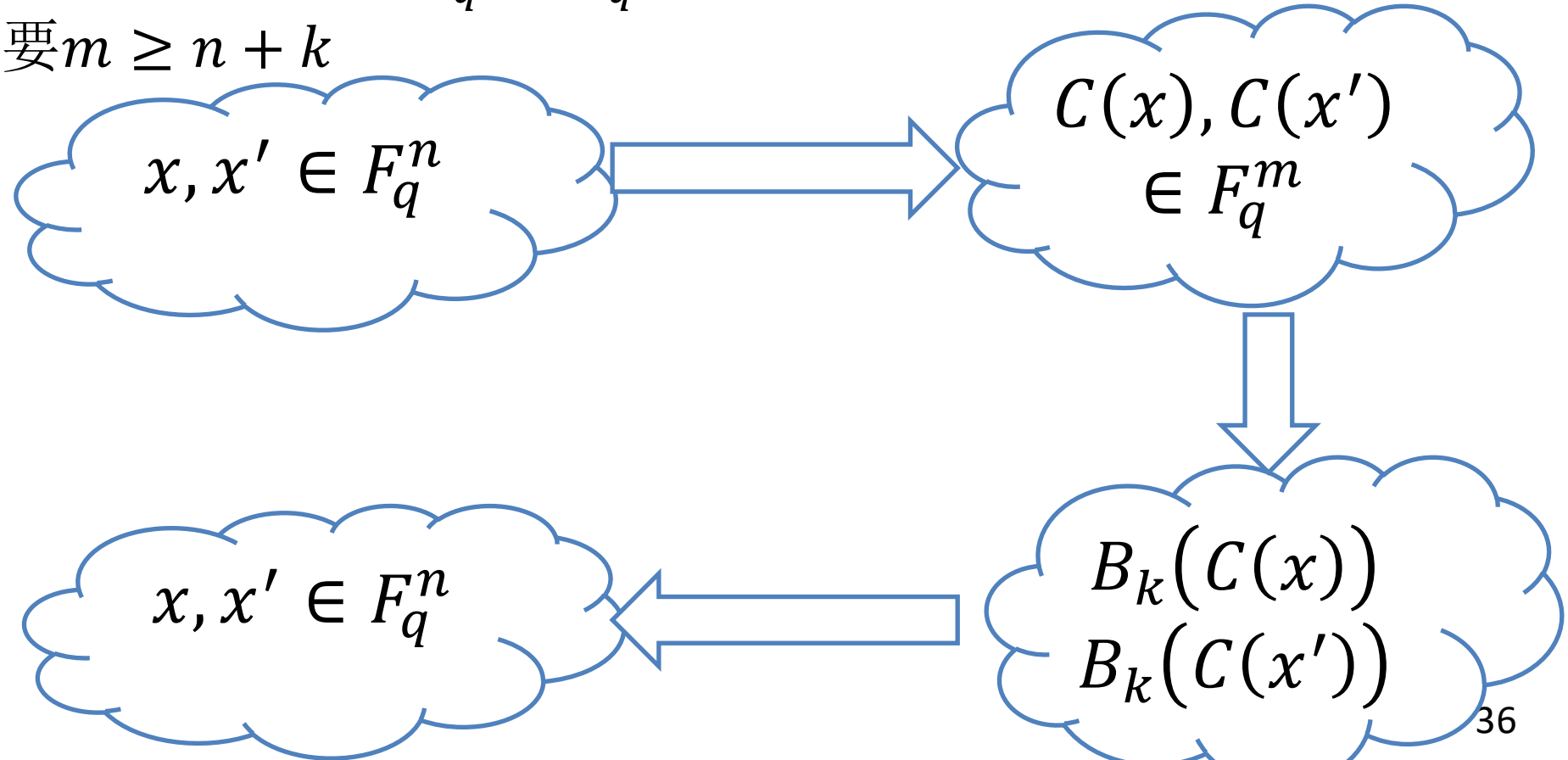
Claim 2: The Reed-Solomon code mapping n messages to a codeword of $n + 2k$ messages has minimum distance at least $2k + 1$

This essentially follows from decoding. If the minimum distance is smaller, then decoding will fail.

See the following slides on Hamming bound.

Hamming bound

考虑自纠错码$C: F_q^n \to F_q^m$. 要处理k个字符的erasure，那么需要$m \geq n + k$



Hamming bound

考虑自纠错码$C: F_q^n \to F_q^m$. 要处理k个字符的erasure，那么需要$m \geq n + k$

反证法：假设$m \leq n + k - 1$

- 考虑所有的codeword的集合: $\{C(x): x \in F_q^n\}$
- 去掉这些codeword的后面 k 位 （投影到前面$n-1$位）
- 剩下每个codeword的长度最多为$n-1$
- 由Pigeon-hole principle，一定存在两个codeword在前$n-1$位是相同的，记为x, x'
- 这意味着，存在$x, x' \in F_q^n$, $x \neq x'$，使得$C(x)$和 $C(x')$在去掉了最后的 k 位后相等
- 这与能处理k个字符的erasure的假设矛盾。

Hamming bound

考虑自纠错码$C: F_q^n \to F_q^m$. 要处理k个字符的corruption，那么需要$m \geq n + 2k$

反证法：假设$m \leq n + 2k - 1$

- 考虑所有的codeword的集合: $\{C(x): x \in F_q^n\}$
- 去掉这些codeword的后面 $2k$ 位 （投影到前面$n-1$位）
- 剩下每个codeword的长度最多为$n-1$
- 由Pigeon-hole principle，一定存在两个codeword在前$n-1$位是相同的，记为x, x'
- 这意味着，存在$x, x' \in F_q^n$, $x \neq x'$，使得$C(x)$和 $C(x')$在去掉了最后的 $2k$ 位后相等
- 因此，可以从$C(x)$修改最多k个字符，再从$C(x')$修改最多k个字符，将得到一样的信息
- 从接收者的角度看，将无法区分这是由$C(x)$修改最多k个字符得到的，还是从$C(x')$修改最多k个字符得到的
- 这与能处理k个字符的corruption矛盾

Polynomial interpolation and Boosting （选讲）

If you can compute a polynomial on sufficiently many points, with decent amount of confidence
And the polynomial has a known degree bound

- Then, you can compute the polynomial correctly, everywhere, by interpolation!

- Say you want to compute some function that can be “approximated”, or “represented” by a polynomial
- And someone promise to help by correctly computing an answer most of the time
- Then you can just encode the function using polynomials (or ECCs to be precise)
- Ask for enough answers
- And decode the function that you really wanted to compute

课外阅读: random self-reducibility of the PERMANENT

函数的一致性逼近（选讲）

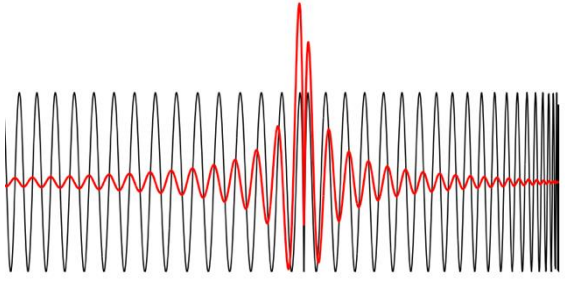
如果不要求多项式恰好经过每一个点，只要求尽可能地近似呢？

函数f的∞-范数 (∞-norm)：

$$\|f\|_\infty := \sup|f|$$

最佳一致逼近：找出多项式p，使得$\|f-p\|_\infty$为最小

回顾Weierstrass **定理的内容**：对于连续函数，**多**项式可以任意地逼近



一些概率论的回顾（选讲）

期望(Expectation)

$$\mathrm{E}[X] = \sum_{x} x \cdot \Pr[X = x]$$

方差(Variance)

$$Var[X] = E[(X - \mathrm{E}[X])^2]$$

Chebyshev不等式

$$\Pr[|X - \mathrm{E}[x]| \geq k] \leq \frac{Var[X]}{k^2}$$

Bernstein Polynomial（选讲）

给定：连续函数$f: [0,1] \to R$

目标：构造多项式p，使得在$x \in [0,1]$这一点上，$p(x)$尽可能地接近$f(x)$

设K是这样的一个随机变量，定义为n次独立的Bernoulli试验中成功的次数，每次成功的概率是x

Bernoulli试验：成功的概率是x，不成功的概率是$1-x$

K服从二项分布(Binomial distribution)，期望$\mathrm{E}[K] = nx$

方差$\mathrm{Var}[K] = nx(1-x)$

Bernstein Polynomial （选讲）

设K是这样的一个随机变量，定义为n次独立的Bernoulli试验中成功的次数，每次成功的概率是x，K服从二项分布(Binomial distribution)

- 期望$\mathrm{E}[K] = nx$
- 方差$\mathrm{Var}[K] = nx(1-x)$
- $\Pr[K = i] = \binom{n}{i} x^i (1-x)^{n-i}$

Bernstein多项式的一组基：$b_{n,i}(x) := \binom{n}{i} x^i (1-x)^{n-i}$

$\lim_{n\to\infty} \Pr\left[\left|\frac{K}{n} - x\right| > \delta\right] = 0$ 对所有x一致地成立

- 注：可从Chebyshev不等式证明

Bernstein Polynomial （选讲）

- 期望$\mathrm{E}[K] = nx$
- 方差$\mathrm{Var}[K] = nx(1-x)$
- $\Pr[K = i] = \binom{n}{i} x^i (1-x)^{n-i}$，Bernstein多项式的一组基

$\lim_{n\to\infty} \Pr\left[\left|\frac{K}{n} - x\right| > \delta\right] = 0$ 对所有x一致地成立

对于闭区间[0,1]，f是连续的意味着f也是一致连续的

因此$\lim_{n\to\infty} \Pr\left[\left|f(\frac{K}{n}) - f(x)\right| > \delta\right] = 0$对$x$一致地成立

进而$\lim_{n\to\infty} \mathrm{E}\left[\left|f(\frac{K}{n}) - f(x)\right|\right] = 0$ 对x一致地成立

Bernstein Polynomial （选讲）

- 期望$\mathrm{E}[K] = nx$
- 方差$\mathrm{Var}[K] = nx(1-x)$
- $\Pr[K = i] = \binom{n}{i} x^i (1-x)^{n-i}$，Bernstein多项式的一组基

$\lim_{n\to\infty} \Pr\left[\left|\frac{K}{n} - x\right| > \delta\right] = 0$ 对所有x一致地成立

对于闭区间$[0,1]$，f是连续的意味着f也是一致连续的

因此$\lim_{n\to\infty} \Pr\left[\left|f(\frac{K}{n}) - f(x)\right| > \delta\right] = 0$对$x$一致地成立

进而$\lim_{n\to\infty} \mathrm{E}\left[\left|f(\frac{K}{n}) - f(x)\right|\right] = 0$ 对x一致地成立

$$\mathrm{E}\left[f\left(\frac{K}{n}\right)\right] = \sum_{i=0} f\left(\frac{i}{n}\right) \Pr(K = i) = \sum_{i=0} f\left(\frac{i}{n}\right) b_{n,i}(x)$$

Bernstein Polynomial （选讲）

- 期望$\mathrm{E}[K] = nx$
- 方差$\mathrm{Var}[K] = nx(1-x)$
- $\Pr[K = i] = \binom{n}{i} x^i (1-x)^{n-i}$，Bernstein多项式的一组基

$\lim_{n\to\infty} \Pr\left[\left|\frac{K}{n} - x\right| > \delta\right] = 0$ 对所有x一致地成立因此

$\lim_{n\to\infty} \Pr\left[\left|f(\frac{K}{n}) - f(x)\right| > \delta\right] = 0$对x一致地成立

进而 $\lim_{n\to\infty} \mathrm{E}\left[\left|f(\frac{K}{n}) - f(x)\right|\right] = 0$ 对x一致地成立

$$\mathrm{E}\left[f\left(\frac{K}{n}\right)\right] = \sum_{i=0} f\left(\frac{i}{n}\right) \Pr(K = i) = \sum_{i=0} f\left(\frac{i}{n}\right) b_{n,i}(x)$$

- Bernstein多项式，一致逼近f
- 具体的误差亦可以使用Chebyshev不等式得到

Bernstein Polynomial （选讲）

误差的定量分析

- **注意到** $\left|f(x)-\mathrm{E}\left[f\left(\frac{K}{n}\right)\right]\right| \le \mathrm{E}\left[\left|f(x)-f\left(\frac{K}{n}\right)\right|\right]$
  - 使用了三角不等式$|a+b| \le |a|+|b|$
- **后者分开两项**

$$\mathrm{E}\left[\left|f(x)-f\left(\frac{K}{n}\right)\right|\right]$$
$$\le 2\Pr\left[\left|\frac{K}{n}-x\right|>\epsilon\right]\cdot\|f\|_\infty+\Pr\left[\left|\frac{K}{n}-x\right|\le\epsilon\right]\cdot\sup_{|x-y|\le\epsilon}|f(x)-f(y)|$$

回忆：$\mathrm{E}[X]=\sum_x x\cdot\Pr[X=x]$

对 $\Pr\left[\left|\frac{K}{n}-x\right|>\epsilon\right]$**使用Chebyshev不等式，即可得到：**

- $\forall x\in[0,1],\ \forall\epsilon>0,$

$$\left|f(x)-\mathrm{E}\left[f\left(\frac{K}{n}\right)\right]\right| \le \frac{\|f\|_\infty}{2n\epsilon^2}+\sup_{|x-y|\le\epsilon}|f(x)-f(y)|$$

Beyond Polynomial （选讲）

Weierstrass **定理**: 对于连续函数f，**多**项式可以任意地逼近

如果不是多项式呢？

推广（Stone-Weierstrass **定理**）: **考**虑连续函数的任意一个子**代数**(subalgebra) A，A能任意地逼近连续函数当且仅当A能分离点：$\forall x \neq y, \exists p \in A, s.t., p(x) \neq p(y)$

因此更一般的，如神经网络等，往往也可以通过Stone-Weierstrass **定理**证明其相关的逼近定理。

下节课

Chebyshev多项式与插值

函数逼近与正交多项式

最小二乘法与最佳平方逼近