多项式插值与函数最佳逼近

Lagrange 插值

很熟悉了啊.

Th 4.1

插值多项式唯一.

n 次 Lagrange 插值多项式, 记为 $L_n(x)$, 形式为

$$L_n(x)=\sum_{k=0}^{n} f(x_k)\prod_{i\neq 0,i\neq k}^{n}\frac{x-x_i}{x_k-x_i}.$$

Th 4.2

给出 n 个互异点, 存在唯一的次数不超过 n 的多项式 $L_n(x_i)$ 满足 $L_n(x_i)=f(x_i),(i=0,\cdots,n)$.

称 $R_n(x)=f(x)-L_n(x)$ 为插值多项式的余项.

Th 4.3

对余项的估计, 很像 Taylor 公式中的 Lagrange 余项.

$$R_n(x)=f(x)-L_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x-x_i).$$

一些说明:

1. $\xi\in\left(\min\limits_{1\leqslant i\leqslant n}\{x_i\},\max\limits_{1\leqslant i\leqslant n}\{x_i\}\right)$.
2. 当 $f(x)$ 本身是一个次数不超过 n 的多项式的时候, 余项等于零.

差商, Newton 插值

先说差商. 考虑 Lagrange 插值中, 增加或减少节点对插值多项式的影响.

设 $L_{k-1}(x)$ 是 $x_0,x_1,\cdots,x_{k-1}$ 为插值节点的多项式, L_k 是 $x_0,x_1,\cdots,x_k$ 为插值节点的多项式.

令

$$g(x)=L_k(x)-L_{k-1}(x),$$

那么 $g(x)$ 是一个次数不超过 k 的多项式, 且对 $i=0,1,\cdots,k-1$ 都有 $g(x_i)=0$.

则可以设

$$g(x)=a_k(x-x_0)(x-x_1)\cdots(x-x_{k-1}).$$

其中 a_k 是一个常数.

(但是我有一个疑问, 如果插值的第 k 个点是 $(x_k,0)$ 怎么办)

接着就有

$$L_k(x)=a_0+a_1(x-x_0)+a_2(x-x_0)(x-x_1)+\cdots+a_k(x-x_0)(x-x_1)\cdots(x-x_{k-1}).$$

其中

$$a_k=\sum_{m=0}^{k}\frac{f(x_m)}{\prod\limits_{i=0,i\neq m}^{k}(x_m-x_i)}.$$

Def 4.2

设

$$f[x_i,x_j]=\frac{f(x_j)-f(x_i)}{x_j-x_i}$$

为 $f(x)$ 关于点 x_i,x_j 的 1 阶差商.

设

$$f[x_i,x_j,x_k]=\frac{f[x_j,x_k]-f[x_i,x_j]}{x_k-x_i}$$

为 $f(x)$ 关于点 x_i,x_j,x_k 的 2 阶差商.

类似地, 可以定义 k 阶差商为

$$f\left[x_0,x_1,\cdots,x_{k-1},x_k\right]=\frac{f\left[x_1,x_2,\cdots,x_{k-1},x_k\right]-f\left[x_0,x_1,\cdots,x_{k-2},x_{k-1}\right]}{x_k-x_0}.$$

约定 0 阶差商即为函数值.

性质:

1. $f[x_0,x_1,\cdots,x_k]=\sum\limits_{m=0}^{k}\frac{f(x_m)}{\prod\limits_{i=0,i\neq m}^{k}(x_m-x_i)}$.
2. k 阶差商的值与点的顺序无关, 这一点从上面的表达式也可以看出来.

接着就是 Newton 插值多项式了.

$$\begin{aligned}L_n(x)=&f(x_0)+f[x_0,x_1](x-x_0)+f[x_0,x_1,x_2](x-x_0)(x-x_1)\\&+\cdots+f[x_0,x_1,\cdots,x_n](x-x_0)(x-x_1)\cdots(x-x_{n-1}).\end{aligned}$$

差分, 等距节点插值

在等距条件下, 差商会有更简单的表示, 也就是差分. 设等距节点 $x_k=x_0+kh(k=0,1,\cdots,n)$, h 称为步长.

这里定义一些线性算子:

1. 向前差分算子: $\Delta f_k=f_{k+1}-f_k$,
2. 向后差分算子: $\nabla f_k=f_k-f_{k-1}$,
3. 中心差分算子: $\delta f_k=f_{k+1/2}-f_{k-1/2}$,

4. 位移算子: $Ef_k = f_{k+1}$.

接着就可以得到:

1. $\Delta^n f_k = (E-I)^n f_k = \sum\limits_{j=0}^{n}(-1)^j C_n^j E^{n-j} f_k = \sum\limits_{j=0}^{n}(-1)^j C_n^j f_{n+k-j}$.
2. $\nabla^n f_k = (I-E^{-1})^n f_k = \sum\limits_{j=0}^{n}(-1)^{n-j} C_n^j E^{j-n} f_k = \sum\limits_{j=0}^{n}(-1)^{n-j} C_n^j f_{j+k-n}$.
3. $f_{n+k} = E^n f_k = (I+\Delta)^n f_k = \sum\limits_{j=0}^{n} C_n^j \Delta^j f_k$.
4. $f\left[x_k, x_{k+1}, \cdots, x_{k+m}\right] = \frac{1}{m!}\frac{1}{h^m}\Delta^m f_k$.
5. $f\left[x_k, x_{k-1}, \cdots, x_{k-m}\right] = \frac{1}{m!}\frac{1}{h^m}\nabla^m f_k$.

Hermite 插值多项式

之前的插值都是只有函数值的情形.

Def 4.3

给定闭区间 $[a,b]$ 中的 $n+1$ 个互异点 x_i 处的函数值以及直到 m_i 阶导数值, 设 $m=\sum_{i=0}^{n}(m_i+1)-1$, 可以得到一个次数不超过 m 的多项式 $H_m(x)$, 称为 $f(x)$ 的 m 次 Hermite 插值多项式.

Th 4.4

自然 m 次多项式 $H_m(x)$ 是唯一的.

Th 4.5

余项 $R_m(x) = f(x) - H_m(x) = \frac{f^{(m+1)}(\xi)}{(m+1)!}\prod\limits_{i=0}^{n}(x-x_i)^{m_i+1}$.

Newton 型 Hermite 插值多项式

补充定义差商:

$f[x_0,x_0] = f'(x_0)$ 以及 $f[x_0,x_0,x_1] = \frac{f[x_0,x_1]-f[x_0,x_0]}{x_1-x_0}$, 那么差值结果如下:

$$
\begin{aligned}
H_m(x) = & f\left(x_0\right) + f\left[x_0, x_0\right]\left(x-x_0\right) + \cdots + f[\underbrace{x_0, \cdots, x_0}_{m_0+1}](x-x_0)^{m_0} + \\
& f[\underbrace{x_0, \cdots, x_0}_{m_0+1}, x_1](x-x_0)^{m_0+1} + \cdots + \\
& f[\underbrace{x_0, \cdots, x_0}_{m_0+1}, \underbrace{x_1, \cdots, x_1}_{m_1+1}](x-x_0)^{m_0+1}(x-x_1)^{m_1} + \cdots + \\
& f[\underbrace{x_0, \cdots, x_0}_{m_0+1}, \cdots, \underbrace{x_{n-1}, \cdots, x_{n-1}}_{m_{n-1}+1}, x_n](x-x_0)^{m_0+1}\cdots(x-x_{n-1})^{m_{n-1}+1} + \cdots + \\
& f[\underbrace{x_0, \cdots, x_0}_{m_0+1}, \cdots, \underbrace{x_{n-1}, \cdots, x_{n-1}}_{m_{n-1}+1}, \underbrace{x_n, \cdots, x_n}_{m_n+1}](x-x_0)^{m_0+1}\cdots(x-x_{n-1})^{m_{n-1}+1}(x-x_n)^{m_n},
\end{aligned}
$$

高次插值的缺点以及分段低次插值

(先挖坑)

三次样条插值

Def 4.5

给定闭区间 $[a,b]$ 上的 $n+1$ 个插值点, 如果函数 $S(x)$ 满足:

1. $S(x_i)=y_i$,
2. $S(x)$ 在每一个小区间上都是三次多项式,
3. $S(x)\in C^2[a,b]$,

就称 $S(x)$ 是 $f(x)$ 的三次样条插值.

首先判断一下需要多少个条件, 每一个小区间需要确定 4 个参数, 也就是一共需要 $4n$ 个.

再计算一下条件数. 由于 $S(x)$ 是 $C^2[a,b]$ 的, 这就意味着对于小区间之间的分界点, 两侧的函数值, 一阶导数和二阶导数的极限要相等. 加上插值条件, 这里一共 $(n+1)+3(n-1)=4n-2$ 个.

这自然是还要补充两个条件, 通常有如下三种, 分别被称为第一型, 第二型和第三型:

1. 两端点的导数,
2. 两端点的二阶导数,
3. 周期边界, 也就是当 $f(x_0)=f(x_n)$ 时, $S'\left(x_0+0\right)=S'\left(x_n-0\right), S''\left(x_0+0\right)=S''\left(x_n-0\right)$.

一通暴算最后可以得到:

$$\mu_i M_{i-1}+2M_i+\lambda_i M_{i+1}=d_i,\quad i=1,2,\cdots,n-1,$$

其中

$$\begin{aligned}
&M_i=S''(x_i),\\
&h_i=x_{i+1}-x_i,\\
&\mu_i=\frac{h_{i-1}}{h_{i-1}+h_i},\\
&\lambda_i=\frac{h_i}{h_{i-1}+h_i},\\
&d_i=6f[x_{i-1},x_i,x_{i+1}].
\end{aligned}$$

第一型

利用 $S'(x_0)=f'(x_0), S'(x_n)=f'(x_n)$ 可以得到

$$\begin{pmatrix}
2 & 1 & & & & \\
\mu_1 & 2 & \lambda_1 & & & \\
 & \mu_2 & 2 & \lambda_2 & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & \mu_{n-1} & 2 & \lambda_{n-1}\\
 & & & & 1 & 2
\end{pmatrix}
\begin{pmatrix} M_0\\ M_1\\ M_2\\ \vdots\\ M_{n-1}\\ M_n\end{pmatrix}
=
\begin{pmatrix} d_0\\ d_1\\ d_2\\ \vdots\\ d_{n-1}\\ d_n\end{pmatrix}.$$

第二型

利用 $M_0=f''(x_0),M_n=f''(x_n)$ 可以得到

$$\begin{pmatrix}2 & \lambda_1 & & & & \\ \mu_2 & 2 & \lambda_2 & & & \\ & \mu_3 & 2 & \lambda_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mu_{n-2} & 2 & \lambda_{n-2} \\ & & & & \mu_{n-1} & 2\end{pmatrix}\begin{pmatrix}M_1 \\ M_2 \\ M_3 \\ \vdots \\ M_{n-2} \\ M_{n-1}\end{pmatrix}=\begin{pmatrix}d_1-\mu_1 f''\left(x_0\right) \\ d_2 \\ d_3 \\ \vdots \\ d_{n-2} \\ d_{n-1}-\lambda_{n-1} f''(x_n)\end{pmatrix}.$$

第三型

利用 $S'\left(x_0\right)=S'\left(x_n\right),S''\left(x_0\right)=S''\left(x_n\right)$ 可以得到

$$\begin{pmatrix}2 & \lambda_1 & & & \mu_1 \\ \mu_2 & 2 & \lambda_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ \lambda_n & & & \mu_n & 2\end{pmatrix}\begin{pmatrix}M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ M_n\end{pmatrix}=\begin{pmatrix}d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n\end{pmatrix}.$$

这都是很好解的.

给出三次条样函数的误差估计.

Th 4.8

$$\left\|f^{(k)}-S^{(k)}\right\|_{\infty} \leqslant c_k\left\|f^{(4)}\right\|_{\infty}\left(\max_{0 \leqslant i \leqslant n-1}(x_{i+1}-x_i)\right)^{4-k}, k=0,1,2,$$

其中 $c_0=\frac{1}{16}, c_1=c_2=\frac{1}{2}$.

最佳一致逼近

Def 4.6

线性空间

Def 4.7

线性赋范空间

Def 4.8

距离

Def 4.9

最佳逼近元

Def 4.10

记 $M_n=\{p_n \mid p_n$ 为次数不超过 n 的多项式 $\}$, 则 $M_n\subset C[a,b]$. 设 $f\in C[a,b]$. 若 $\exists p_n\in M_n$, 使得对 $\forall q_n\in M_n$, 有

$$\|f-p_n\|_\infty \leqslant \|f-q_n\|_\infty,$$

则称 $p_n(x)$ 是 $f(x)$ 的 n 次最佳一致逼近多项式.

因为是一致逼近, 所以选择 L^∞ 范数.

Th 4.11

对于 $f\in C[a,b]$, 在 M_n 中一定存在最佳一致逼近多项式.

两个条件, 闭区间和连续函数.

Def 4.12

称满足 $x_0\in[a,b], |g(x_0)|=\|g\|_\infty$ 的点 x_0 为 $g(x)$ 在 $[a,b]$ 上的偏差点.

若 $g(x_0)=\|g\|_\infty$ 的 x_0 为正偏差点, 否则为负偏差点.

Lemma

如果 $f\in[a,b]$, $p_n(x)$ 是其 n 次最佳一致逼近多项式, 那么 $f-p_n$ 一定存在正负偏差点.

(why 提到了 切比雪夫多项式?)

最佳平方逼近

Def 4.13

内积空间

Def 4.14

正交

Lemma

Cauchy - Schwartz 不等式

Th 4.15

将 Th 4.10, 也就是最佳一致逼近中的 $\|\cdot\|_\infty$ 改为 $\|\cdot\|_2$ 就得到了最佳平方逼近. 泛函里学过, 最佳平方逼近是有唯一解的, 并且给出了计算方式.

离散数据的最佳平方逼近

对于给定数据

Error: Unknown environment 'tabular'

如果选取基 $\varphi_k(x)=x^k$, 得到的 m 次最佳平方逼近多项式被称为 m 次最小二乘多项式.

记

$$
\boldsymbol{\varphi}_k = \begin{pmatrix} \varphi_k(x_1) \\ \varphi_k(x_2) \\ \vdots \\ \varphi_k(x_n) \end{pmatrix}, \quad k = 0, 1, \cdots, m, \quad \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix},
$$

那么 m 次最小二乘多项式的系数就是下面线性方程组的解

$$
\begin{pmatrix} (\boldsymbol{\varphi}_0, \boldsymbol{\varphi}_0) & (\boldsymbol{\varphi}_0, \boldsymbol{\varphi}_1) & \cdots & (\boldsymbol{\varphi}_0, \boldsymbol{\varphi}_m) \\ (\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_0) & (\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_1) & \cdots & (\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_m) \\ \vdots & \vdots & \ddots & \vdots \\ (\boldsymbol{\varphi}_m, \boldsymbol{\varphi}_0) & (\boldsymbol{\varphi}_m, \boldsymbol{\varphi}_1) & \cdots & (\boldsymbol{\varphi}_m, \boldsymbol{\varphi}_m) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_m \end{pmatrix} = \begin{pmatrix} (\boldsymbol{y}, \boldsymbol{\varphi}_0) \\ (\boldsymbol{y}, \boldsymbol{\varphi}_1) \\ \vdots \\ (\boldsymbol{y}, \boldsymbol{\varphi}_m) \end{pmatrix}.
$$

超定线性方程组的最小二乘解

对于线性方程组

$$
\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix},
$$

其中 $n < m$, 且系数矩阵的列向量线性无关, 这样的线性方程组称为超定方程组.

这样的方程组一般没有精确解.

记

$$
\boldsymbol{A}_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix}, \quad j = 1, 2 \cdots, n, \quad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix},
$$

那么上述方程组可被写成

$$
x_1 \boldsymbol{A}_1 + x_2 \boldsymbol{A}_2 + \cdots + x_3 \boldsymbol{A}_3 = \boldsymbol{b}.
$$

还是利用泛函的东西.

记 $\boldsymbol{M} = \operatorname{span}\{\boldsymbol{A}_1, \boldsymbol{A}_2 \cdots, \boldsymbol{A}_n\}$, 那么 $\boldsymbol{M}$ 是 $\mathbb{R}^m$ 的一个有限维子空间, 那么设

$$
\Phi(x_1, x_2, \cdots, x_n) = \left\| \boldsymbol{b} - \sum_{i=1}^{n} x_i \boldsymbol{A}_i \right\|^2,
$$

就一定会有 $(x_1^*, x_2^*, \cdots, x_n^*)$ 使得

$$
\Phi(x_1^*, x_2^*, \cdots, x_n^*) = \min_{x_i \in \mathbb{R}} \Phi(x_1, x_2, \cdots, x_n).
$$

根据之前的理论知道, $(x_1^*, x_2^*, \cdots, x_n^*)$ 其实就是下面方程的解

$$
\boldsymbol{A}^T \boldsymbol{A} \boldsymbol{x} = \boldsymbol{A}^T \boldsymbol{b}.
$$

连续函数的最佳平方逼近

$$\begin{pmatrix} (\boldsymbol{\varphi}_0, \boldsymbol{\varphi}_0) & (\boldsymbol{\varphi}_0, \boldsymbol{\varphi}_1) & \cdots & (\boldsymbol{\varphi}_0, \boldsymbol{\varphi}_m) \\ (\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_0) & (\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_1) & \cdots & (\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_m) \\ \vdots & \vdots & \ddots & \vdots \\ (\boldsymbol{\varphi}_m, \boldsymbol{\varphi}_0) & (\boldsymbol{\varphi}_m, \boldsymbol{\varphi}_1) & \cdots & (\boldsymbol{\varphi}_m, \boldsymbol{\varphi}_m) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_m \end{pmatrix} = \begin{pmatrix} (\boldsymbol{y}, \boldsymbol{\varphi}_0) \\ (\boldsymbol{y}, \boldsymbol{\varphi}_1) \\ \vdots \\ (\boldsymbol{y}, \boldsymbol{\varphi}_m) \end{pmatrix}.$$

依旧是成立的, 只需要注意内积变成函数的内积.