# 插值与拟合

## 一维插值

### 1.基本概念

已知未知函数在 $n+1$ 个互不相同的观测点 $x_0 < x_1 < \cdots < x_n$ 处的函数值:

$$y_i = f_(x_i), \quad i = 0, 1, \ldots, n.$$

寻求一个近似函数 $\phi(x)$，使之满足

$$\phi(x_i) = y_i, \quad i = 1, 2, \ldots, n.$$

若插值函数为代数多项式，则称为多项式插值

记

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

且满足

$$P_n(x_k) = y_k, \quad k = 0, 1, \ldots, n.$$

称 $P_n(x)$ 为 n 次插值多项式.

### 2.待定系数法

记

$$X = \begin{bmatrix} x_0^n & \cdots & x_0 & 1 \\ x_1^n & \cdots & x_1 & 1 \\ \vdots & & \vdots & \vdots \\ x_n^n & \cdots & x_n & 1 \end{bmatrix}, \quad A = [a_n \quad a_{n-1} \quad \cdots \quad a], \quad Y = [y_0 \quad y_1 \quad \cdots \quad y_n],$$

解方程组

$$AX = Y$$

即可得出 $P_n(x)$ 的各项系数.

### 3.拉格朗日插值

构造

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \quad i = 0, 1, \ldots, n.$$

从而 $l_i(x)$ 为 n 次多项式，且满足

$$l_i(x_j) = \begin{cases} 0, & j \neq i, \\ 1, & j = i. \end{cases}$$

令

$$L_n(x) = \sum_{i=0}^{n} y_i l_i(x)$$

称为 n 次拉格朗日插值多项式.

## 4.牛顿插值

称

$$N_n(x) = f(x_0) + (x - x_0) f[x_0,\ x_1] + \cdots + (x - x_0)(x - x_1) \cdots (x - x_{n-1}) f[x_0,\ x_1,\ \cdots,\ x_n]$$

为 n 次牛顿插值多项式.
其中

$$f[x_0,\ x_1,\ \cdots,\ x_k] = \frac{f[x_0,\ x_1,\ \cdots,\ x_{k-1}] - f[x_0,\ x_1,\ \cdots,\ x_k]}{x_0 - x_k}$$

为 $f(x)$ 关于点 $x_0,\ x_1,\ \ldots,\ x_k$ 的 k 阶差商.

## 5.分段线性插值

将每个小区间以线性函数替换，
记为

$$I_n(x) = \sum_{i=0}^{n} y_i l_i(x),$$

满足

$$I_n(x) = y_i，且\ I_n(x)\ 在每个小区间\ [x_i,\ x_{i+1}]\ 上为线性函数$$

$l_i(x)$ 为插值基函数，其表达式为

$$l_0(x)=\begin{cases}\dfrac{x-x_1}{x_0-x_1}, & x_0\le x\le x_1,\\ 0, & \text{其他}.\end{cases}$$

$$l_i(x)=\begin{cases}\dfrac{x-x_{i-1}}{x_i-x_{i-1}}, & x_{i-1}\le x\le x_i,\\ \dfrac{x-x_{i+1}}{x_i-x_{i+1}}, & x_i\le x\le x_{i+1},\\ 0, & \text{其他}.\end{cases}$$

$$l_n(x)=\begin{cases}\dfrac{x-x_{n-1}}{x_n-x_{n-1}}, & x_{n-1}\le x\le x_n,\\ 0, & \text{其他}.\end{cases}$$

且满足

$$l_i(x_j)=\begin{cases}0, & j\ne i,\\ 1, & j=i.\end{cases}$$

## 6.样条插值

给定区间 $[a,\ b]$ 的一个分划

$$a=x_0<x_1<\cdots<x_{n-1}<x_n=b$$

若函数 $S(x)$ 满足

(1) 在每个区间 $[x_i,\ x_{i-1}](i=0,\ 1,\ \ldots,\ n-1)$ 上 $S(x)$ 为 $k$ 次多项式；
(2) $S(x)$ 在 $[a,\ b]$ 上具有 $k-1$ 阶连续导数.

则称 $S(x)$ 为关于该分划的 $k$ 次样条函数.

# 二维插值

## 1.基本概念

已知 $xOy$ 平面上 $m\times n$ 个互不相同的节点

$$(x_i,\ y_j),\quad i=1,2,\ldots,m,\ j=1,2,\ldots,n$$

处的函数值

$$z_{ij},\quad i=1,2,\ldots,m,\ j=1,2,\ldots,n,$$

求一个近似函数，使其通过全部已知节点，即

$$f(x_i,\ y_j)=z_{ij},\quad i=1,2,\ldots,m,\ j=1,2,\ldots,n.$$

通常取插值区间

$$a = x_1 < \cdots < x_m = b,\ c < y_1 < \cdots < y_n = d$$

为 $xOy$ 平面上的一矩形区域.

# 2.网格节点插值

## 分片线性插值

对于

$$x_i \le x \le x_{i+1},\ y_j \le y \le y_{j+1},$$

有插值函数

$$f(x,\ y) = z_{ij} + (z_{i+1,j} - z_{ij})\frac{x - x_i}{x_{i+1} - x_i} + (z_{i+1,j+1} - z_{i+1,j})\frac{y - y_j}{y_{j+1} - y_j}$$

## 双线性插值

对于

$$x_i \le x \le x_{i+1},\ y_j \le y \le y_{j+1},$$

有插值函数

$$f(x,\ y) = Axy + Bx + Cy + D$$

其系数通过四个顶点值构成的方程

$$\begin{gathered} f(x_i,\ y_j) = z_{ij}, \quad f(x_{i+1},\ y_j) = z_{i+1,j}, \\ f(x_{i+1},\ y_{j+1}) = z_{i+1,j+1}, \quad f(x_i,\ y_{j+1}) = z_{i,j+1}, \end{gathered}$$

解出.

# 3.散乱数据插值

已知在 $\Omega = [a,\ b] \times [c,\ d]$ 内散乱分布 N 个观测点 $(x_k,\ y_k),\ k = 1, 2, \ldots, N$ 及其观测值 $z_k (k = 1, 2, \ldots, N)$ ，寻求在 $\Omega$ 上的二元函数，使得

$$f(x_k,\ y_k) = z_k, \quad k = 1, 2, \ldots, N.$$

采用 Shepard方法，
首先计算任意观测点 $(x_k,\ y_k)$ 离插值点 $(x,\ y)$ 的距离

$$r_k = \sqrt{(x - x_k)^2 + (y - y_k)^2}, \quad k = 1, 2, \ldots, N,$$

接着定义第 k 个观测值对 $(x,\ y)$ 点函数值的影响权值

$$w_k(x,\ y) = \frac{1}{r_k^2} \Big/ \sum_{i=1}^{k} \frac{1}{r_i^2}, \quad k = 1, 2, \ldots, N,$$

最后得出插值函数

$$f(x, y) = \begin{cases} z_k, & r_k = 0, \\ \sum\limits_{k=1}^{N} w_k(x.\ y) z_k, & r_k \neq 0. \end{cases}$$

# 用Python求解插值问题

## 1.一维插值求解

使用 `scipy.interpolate` 模块的 `interp1d()` 函数和 `lagrange()` 函数,
例如:

```
import numpy as np
from scipy.interpolate import interp1d
from scipy.interpolate import lagrange

a = np.loadtxt('data.txt')
x0 = a[0]
y0 = a[1]
x = np.linspace(0, 15, 151)
yx1 = interp1d(x0, y0)  # 分段线性插值
y1 = yx1(x)  # 计算插值点的函数值
p2 = lagrange(x0, y0)  # 拉格朗日插值
y2 = np.polyval(p2, x)  # 计算拉格朗日多项式的值
yx3 = interp1d(x0, y0, 'cubic')  # 三阶样条插值
y3 = yx3(3)
```

## 2.二维插值求解

使用 `scipy.interpolate` 模块的 `interp2d()` 函数和 `griddata()` 函数,
例如:
**网格节点插值:**

```
import numpy as np
from scipy.interpolate import interp2d

z = np.loadtxt('data.txt')
```

```python
x = np.arange(0, 1500, 100)
y = np.arange(1200, -100, -100)
f1 = interp2d(x, y, z)  #双线性插值
xn1 = np.linsapce(0, 1400, 141)  # 计算插值点的函数值
yn1 = np.linsapce(0, 1200, 121)
zn1 = f1(xn1, yn1)
f2 = interp2d(x, y, z, 'cubic')  # 双三阶样条插值
xn2 = np.linsapce(100, 500, 5)
yn2 = np.linsapce(100, 400, 4)
zn2 = f2(xn2, yn2)
```

**二维散乱点插值:**

```python
import numpy as np
from scipy.interpolate import griddata

a = np.loadtxt('data.txt')
x = a[0]
y = a[1]
z = a[2]
# 将x和y数组垂直堆叠，转置得到xy数组，其中每一行是一个数据点的x和y坐标
xy = np.vstack([x, y]).T
# 生成均匀分布的x坐标值和y坐标值，覆盖了x和y数组中最小值和最大值之间的范围
xn = np.linspace(x.min(), x.max(), 100)
yn = np.linspace(y.min(), y.max(), 200)
# 创建了一个网格，用于构建插值的目标点坐标
xng, yng = np.meshgrid(xn, yn)
# 对散乱点进行插值，方法为三阶样条插值
zn = griddata(xy, z, (xng, yng), method='cubic')
```

# 最小二乘拟合

## 1.基本概念

已知平面上 n 个点 $(x_i,\ y_i)(i=1,2,\dots,n)$ ，要寻求一个函数

$$f(x)=(x,\ a_1,\ a_2,\ \dots,\ a_m)$$

使

$$J(a_1,\ a_2,\ \dots,\ a_m)=\sum_{i=1}^{n}\delta_i^2$$

达到最小，其中

$$\delta_i = f(x_i) - y_i, \quad i = 1, 2, \ldots, n,$$

称为拟合函数 $f(x)$ 在 $x_i$ 点处的残差.

给定一个线性无关的函数系

$$\{\varphi_k(x) \mid k = 1, 2, \ldots, m\}$$

若拟合函数以

$$f(x) = \sum_{k=1}^{m} a_k \varphi_k(x)$$

的形式出现，则称之为线性最小二乘拟合，否则称之为非线性最小二乘拟合.

记

$$R = \begin{bmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \cdots & \varphi_m(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \cdots & \varphi_m(x_2) \\ \vdots & \vdots & & \vdots \\ \varphi_1(x_n) & \varphi_2(x_n) & \cdots & \varphi_m(x_n) \end{bmatrix}, \quad A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

称

$$R^T R A = R^T Y$$

为线性最小二乘拟合的正规方程组，其解矩阵 A 为拟合函数 $f(x)$ 的系数.
在其有解的情况下，可化为

$$RA = Y.$$

# 2.Python求解最小二乘拟合

## 线性最小二乘拟合

根据所给线性方程组，调用 `A = np.linalg.pinv(R) @ Y` 即可求出系数矩阵.
还可以调用 `p = polyfit(x, y, n)` 实现 n 次多项式拟合，

若在最小二乘意义下解约束线性方程组

$$Cx = d,$$
$$s.t. \begin{cases} Ax \le b, \\ Aeq \cdot x = beq, \\ lb \le x \le ub, \end{cases}$$

即化为求解数学规划问题

$$min\ \frac{1}{2}||\ Cx-d\ ||_2^2,$$
$$s,t.\begin{cases}Ax\le b,\\ Aeq\cdot x=beq,\\ lb\le x\le ub,\end{cases}$$

其中

$$||\ Cx-d\ ||_2$$

表示 L2范数，即向量的欧几里得距离.

例如：用给定数据拟合函数 $y=ae^x+blnx$.

```
import numpy as np
import cvxpy as cp

a = np.loadtxt('data.txt')
x0 = a[0]
y0 = a[1]
# 无约束条件的求解
R = np.vstack([np.exp(x0), np.log(x0)]).T
A = np.linalg.pinv(R) @ y0
# 约束条件 a >= 0, b >= 0, a + b <= 1下的求解
t = cp.Variable(2, pos=True)
obj = Minimize(cp.sum_quares(R @ t - y0))
con = [sum(t) <= 1]
prob = cp.Problem(obj, con)
prob.solve(solver='CVXOPY')
```

## 非线性最小二乘拟合

使用 `scipy.optimize` 模块中的 `curve_fit` 函数，
调用格式为 `popt, pcov = curve_fit(func, xdata, ydata)`，
其中 func 为拟合的函数，xdata、ydata为自变量和因变量的观测值，返回值popt是拟合的参数，pcov是参数的协方差矩阵.

例如：用给定数据拟合 $y=ke^{mt}$.

```
import numpy as np
from scipy.optimize import curve_fit
```

```python
a = np.loadtxt('data.txt')
t0 = a[0]
y0 = a[1]
y = lambda t, k: k * np.exp(m * t)
p, pcov = curve_fit(y, t0, y0)
```

# 函数逼近

## 1.基本概念

已知连续函数

$$y(x),\quad x \in [\,a,\ b\,]$$

选取函数

$$\{r_k(x) \mid k = 1, 2, \ldots, m\}$$

构造 $f(x)$，即

$$f(x) = a_1 r_1(x) + a_2 r_2(x) + \cdots + a_m r_m(x),$$

使

$$J(a_1,\ a_2,\ \ldots,\ a_m) = \int_a^b [f(x) - y(x)]^2 dx$$

达到最小.
利用极值必要条件，有

$$\begin{bmatrix} (r_1,\ r_1) & \cdots & (r_1,\ r_m) \\ \vdots & & \vdots \\ (r_m,\ r_1) & \cdots & (r_m,\ r_m) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} (y,\ r_1) \\ \vdots \\ (y,\ r_m) \end{bmatrix},$$

其中

$$(g,\ h) = \int_a^b g(x) h(x) dx,$$

当上述方程组的稀疏矩阵非奇异时，有唯一解.
一般选取 $r_k(x)$ 为正交多项式，使

$$\int_a^b r_i(x) r_j(x) dx = 0,\ (i \neq j),$$

从而化简计算.

## 2.Python求解逼近函数

例如：求 $f(x) = cosx,\ x \in [-\frac{\pi}{2},\ \frac{\pi}{2}]$ 在 $H = Span[1, x^2, x^4]$ 中的最佳平方逼近多项式.

```
import sympy as sp

x = sp.var('x')
base = sp.Matrix([1, x ** 2, x ** 4])
y1 = base @ (base.T)
y2 = sp.cos(x) * base
r1 = sp.integrate(y1, (x, -sp.pi / 2, sp.pi /2))
r2 = sp.integrate(y2, (x, -sp.pi / 2, sp.pi /2))
a = r1.inv() @ r2
xs = a.n(4)
print('系数的符号解为\n', a)
print('系数的小数显示为', xs)
```