Orthonormal Basis Functions

given a set of basis functions $\{g_0, g_1, \ldots, g_n\}$, the set of all functions that are linear combinations of the basis functions are

$$\mathcal{G} = \{\ g : \ \text{such that} \ \ g(x) = \sum_{j=0}^{n} c_j\, g_j(x)\ \}$$

we are looking for a particular $g(x) \in \mathcal{G}$ such that the fitting total error is minimized

any $n+1$ functions that are linearly independent can be used as basis functions. Different choices of basis functions make the normal equation

$$\sum_{j=0}^{n} \left[\sum_{k=0}^{m} g_i(x_k)\, g_j(x_k) \right] c_j = \sum_{k=0}^{m} y_k\, g_i(x_k)$$

for $0 \le i \le n$, easier or more difficult to solve

Chosen Basis Functions

we say a basis $\{g_0, g_1, \ldots, g_n\}$ has the property of *orthonormality* if

$$\sum_{k=0}^{m} g_i(x_k)\, g_j(x_k) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

in this case, the normal equation is simplified as

$$c_j = \sum_{k=0}^{m} y_k\, g_j(x_k) \qquad (0 \leq j \leq n)$$

which can be evaluated straightforwardly

the *Gram-Schmidt procedure* can be used to orthonormalize a given basis in order to have the above property. This procedure may be expensive

we can also chose some basis functions so that the coefficient matrix is easy to solve, not necessarily as an identity matrix

Polynomial Basis Functions

consider $\mathcal{G}$ as the space of all polynomials of degree $\leq n$. We naturally chose

$$g_0(x) = 1,\ g_1(x) = x,\ \ldots,\ g_n(x) = x^n$$

any polynomial in $\mathcal{G}$ can be represented as

$$g(x) = \sum_{j=0}^{n} c_i\, g_j(x) = \sum_{j=0}^{n} c_j\, x^j$$

the simple basis is, however, not very good, since they are too much alike

assume we have the data restricted in the interval $[-1, 1]$ with

$$-1 = x_0 < x_1 < \cdots < \cdots < x_m = 1$$

we can define a set of *Chebyshev* polynomials that form a good basis

Chebyshev Polynomials

the first few Chebyshev polynomials are

$$T_0(x) = 1, \; T_1(x) = x, \; T_2(x) = 2x^2 - 1$$

$$T_3(x) = 4x^3 - 3x, \; T_4(x) = 8x^4 - 8x^2 + 1$$

the Chebyshev polynomials can be generated recursively as

$$T_j(x) = 2x\,T_{j-1}(x) - T_{j-2}(x) \qquad (j \geq 2)$$

they can also be written as

$$T_k(x) = \cos(k \; \arccos x)$$

a function can be represented as a linear combination of the Chebyshev polynomials

$$f(x) = \sum_{j=0}^{n} c_j\, T_j(x)$$

a function $f(x)$ written as a linear combination of the Chebyshev polynomials can be evaluated efficiently

Evaluating Chebyshev Polynomials

to evaluate $f(x)$ for any given x in

$$f(x) = \sum_{j=0}^{n} c_j T_j(x)$$

we use a backward recursion procedure

$$\begin{cases} w_{n+2} = w_{n+1} = 0 \\ w_j = c_j + 2x\, w_{j+1} - w_{j+2} \qquad (n \geq j \geq 0) \\ g(x) = w_0 - x w_1 \end{cases}$$

the Chebyshev polynomials are defined on the interval $[-1, 1]$, we would also like the abscissas $\{x_i\}$ lie in the interval $[-1, 1]$, i.e., $\min\{x_k\} = -1$ and $\max\{x_k\} = 1$. If they lie in a different interval $[a, b]$, we can use a transformation

$$x = \frac{1}{2}(b-a)z + \frac{1}{2}(a+b)$$

to map the interval $[-1, 1]$ onto $[a, b]$

Algorithm of Polynomial Fitting

1. find the smallest interval containing all x_k with $a = \min\{x_k\}$ and $b = \max\{x_k\}$

2. make a transformation to the interval $[-1, 1]$ using the map

$$z_k = \frac{2x_k - a - b}{b - a} \qquad (0 \le k \le m)$$

3. decide on the order of the polynomials, around 8 or 10

4. using the Chebyshev polynomials as a basis, generate the $(n+1) \times (n+1)$ normal equations

$$\sum_{j=0}^{n} \left[\sum_{k=0}^{m} T_i(z_k) T_j(z_k) \right] c_j = \sum_{k=0}^{m} y_k T_i(z_k)$$

for $0 \le i \le n$

Algorithm (II)

5. use an equation-solving routine to solve the normal equations for coefficients $c_0, c_1, \ldots c_n$ to obtain the function

$$f(x) = \sum_{j=0}^{n} c_j T_j(x)$$

6. transform the function back to the original variable as

$$f\left(\frac{2x - a - b}{b - a}\right)$$

the computational extensive part is to form the coefficient matrix of the normal equation $Ac = b$, let $A = (a_{ij})_{0:n \times 0:n}$ and $b = (b_i)_{0:n}$

$$\begin{aligned} a_{ij} &= \sum_{k=0}^{m} T_i(z_k) T_j(z_k) \qquad (0 \le i, j \le n) \\ b_i &= \sum_{k=0}^{m} y_k T_i(z_k) \qquad (0 \le i \le n) \end{aligned}$$

specific procedures are detailed in book

Polynomial Regression

assume the data collected contain errors, the procedure of *smoothing data* is to remove the experimental errors as much as possible

smoothing data is different from interpolation, since the latter assumes that the data are accurate

given a table of experimental data

x	x_0	x_1	$\cdots$	x_m
y	y_0	y_1	$\cdots$	y_m

we want to find a polynomial that represents the original data features

$$P_N(x) = \sum_{i=0}^{N} a_i \, x^i$$

we have

$$y_i = P_N(x_i) + \epsilon_i \qquad (0 \le i \le m)$$

where ϵ_i is the observational error in y_i

Polynomial Regression (II)

we can use the method of least squares through solving a system of normal equations to determine $P_n(x)$. A quantity call *variance*

$$\sigma_n^2 = \frac{1}{m-n} \sum_{i=0}^{m} [y_i - p_n(x_i)]^2 \qquad (m > n)$$

can be computed to see how good the approximation is

if the original data really represent a polynomial of degree N with noise, then

$$\sigma_0^2 > \sigma_1^2 > \cdots > \sigma_N^2 = \sigma_{N+1}^2 = \cdots = \sigma_{m-1}^2$$

we can compute $\sigma_0^2, \sigma_1^2, \ldots$ until we see for some N that $\sigma_N^2 \approx \sigma_{N+1}^2 \approx \sigma_{N+2}^2 \approx \cdots$, then we choose the polynomial P_N as the one representing the original data trend

the drawback is we need to compute $p_0, p_1, \ldots$

Inner Product

let two functions f and g whose domains contain $\{x_0, x_1, \ldots, x_m\}$ we define

$$\langle f, g \rangle = \sum_{i=0}^{m} f(x_i)\, g(x_i)$$

as the inner product of the functions f and g

an inner product $\langle \cdot, \cdot \rangle$ of two functions has the following properties

1. $\langle f, g \rangle = \langle g, f \rangle$

2. $\langle f, f \rangle > 0$ unless $f(x_i) = 0$ for all i

3. $\langle a\, f, g \rangle = a \langle f, g \rangle$ where a is a scalar

4. $\langle f, g + h \rangle = \langle f, g \rangle + \langle f, h \rangle$

Orthogonal Polynomials

a set of functions is *orthogonal* if $\langle f, g \rangle = 0$ for any two different functions in the set

we can generate a set of orthogonal functions as

$$\begin{cases} q_0(x) = 1 \\ q_1(x) = x - \alpha_0 \\ q_{n+1}(x) = xq_n(x) - \alpha_n q_n(x) - \beta_n q_{n-1}(x) \end{cases}$$

for $n \geq 1$, where

$$\begin{aligned} \alpha_n &= \frac{\langle x\, q_n, q_n \rangle}{\langle q_n, q_n \rangle} \\ \beta_n &= \frac{\langle x\, q_n, q_{n-1} \rangle}{\langle q_{n-1}, q_{n-1} \rangle} \end{aligned}$$

the polynomials $\{q_0, q_1, \ldots, q_{m-1}\}$ can span a linear space in which they are a basis

Representing a Function

a polynomial of degree $n(\leq m-1)$ in the spanned linear space can be represented as

$$p(x) = \sum_{i=0}^{n} a_i\, q_i(x)$$

if we form the inner product with respect to q_j on both sides

$$\langle p, q_j \rangle = \sum_{i=0}^{n} a_i\, \langle q_i, q_j \rangle$$

for $0 \leq j \leq n$ and using the fact that $\langle q_i, q_j \rangle = 0$ if $i \neq j$ (why?), we have

$$\langle p, q_j \rangle = a_j\, \langle q_j, q_j \rangle$$

Hence

$$a_j = \frac{\langle p, q_j \rangle}{\langle q_j, q_j \rangle}$$

for $j = 0, 1, \ldots, n$ are the needed coefficients