

CS537

Numerical Analysis

Lecture

Least Squares and Curve Fitting

Professor Jun Zhang

Department of Computer Science

University of Kentucky

Lexington, KY 40206-0046

November 13, 2008

Method of Least Squares

Computer aided data collections have produced tremendous amount of data that are impossible to understand without some sort of post-processing

Given a set of data

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

If we assume that the data form a linear functional relation, we can write the function as

$$y = a x + b$$

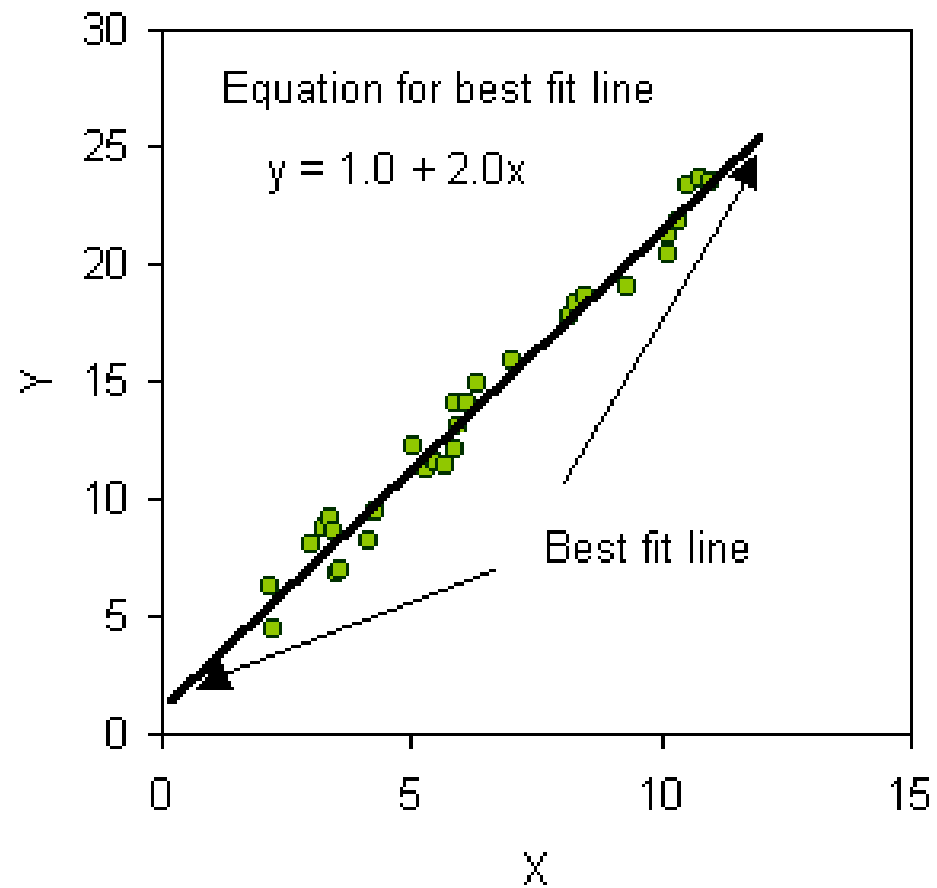
With the coefficients a and b to be determined

For each pairs (x_i, y_i) we can request

$$y_i = a x_i + b$$

For $i = 0, 1, \dots, m$. Note that, if $m > 1$, we have more than two linear equations to determine just two unknowns

Least Squares Fit



Many sample points, which may be inaccurate. Not good for interpolation.

Over-Determined System

In general, we have more equations than unknowns. There is no *exact solution* to the problem. However, we could determine a solution that minimizes the total error

Suppose the linear equation is given as

$$y = a x + b$$

If the point (x_i, y_i) is on the straight line defined by the function, we have

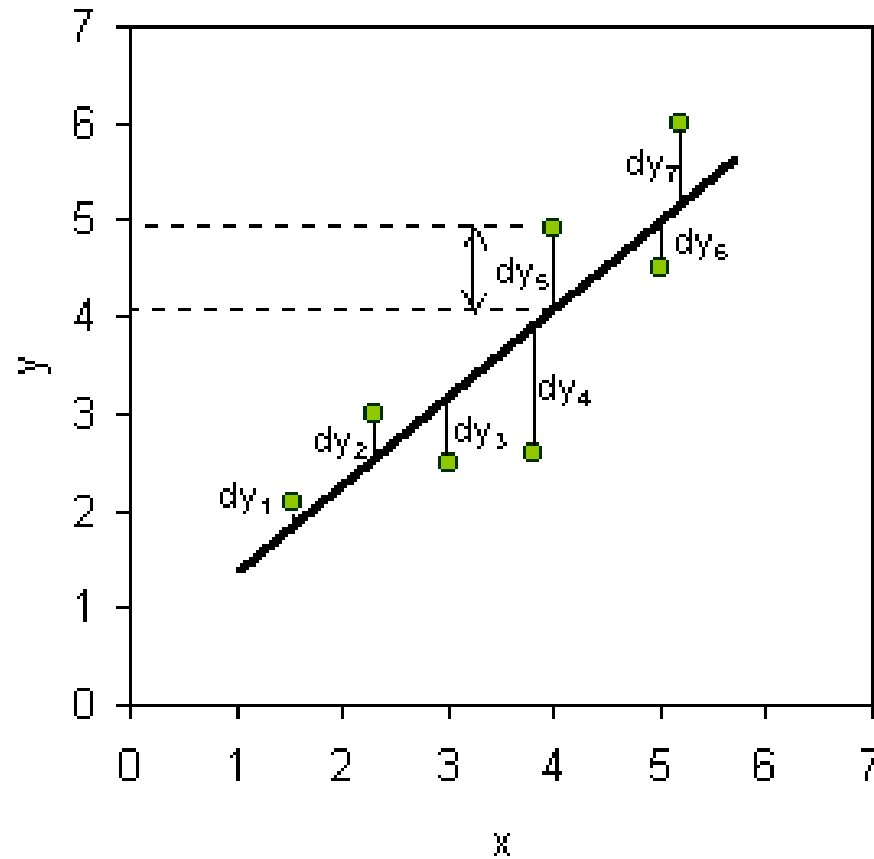
$$y_i - (a x_i + b) = 0$$

In most cases, a point (x_i, y_i) is not on the line, we have

$$r_i = y_i - (a x_i + b) \neq 0$$

For $i = 0, 1, 2, \dots, m$. $|r_i|$ is the curve fitting error

Minimize the Errors



Most sample points are not on the curve. Hopefully the total distance between the sample points and the fitting curve is minimized.

Minimizing Total Error

One can reason that if the sum of the errors is minimized, the data should fit the line as best as it can

The total error can be represented as

$$\sum_{k=0}^m |r_i| = \sum_{k=0}^m |ax_k + b - y_k|$$

We can minimize the above functional to select the coefficients ***a*** and ***b***. This can be solved by the techniques of linear programming

This is an l_1 approximation (1-norm approximation)

The shortcoming of this formulation is that the function of the total error is not differentiable. Many tools in calculus may not be used

Method of Least Squares

An alternative is to minimize a different error function

$$\phi(a, b) = \sum_{k=0}^m (a x_k + b - y_k)^2$$

which is continuously differentiable

This is also called an l_2 approximation. It is a special case of the l_p approximation with the l_p norm is defined as

$$\|\mathbf{x}\|_p = \left\{ \sum_{i=1}^n |x_i|^p \right\}^{1/p} \quad (1 \leq p < \infty)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ is an n dimensional vector

From statistical considerations, if the errors follow a normal probability distribution, the minimization of ϕ produces a best estimate of a and b

How to Compute a Minimum

We use a technique in calculus to determine the extreme point of a function

$$\frac{\partial \phi}{\partial a} = 0 \quad \frac{\partial \phi}{\partial b} = 0$$

This gives us two equations

$$\sum_{k=0}^m 2(a x_k + b - y_k) x_k = 0$$
$$\sum_{k=0}^m 2(a x_k + b - y_k) = 0$$

They are called the *normal equations* and can be written explicitly as

$$\left(\sum_{k=0}^m x_k^2 \right) a + \left(\sum_{k=0}^m x_k \right) b = \sum_{k=0}^m y_k x_k$$
$$\left(\sum_{k=0}^m x_k \right) a + (m + 1) b = \sum_{k=0}^m y_k$$

which can be solved for a and b

Least Squares Solution

The solution of the previous two-by-two linear system can be solved as

$$a = \frac{1}{d} \left[(m+1) \sum_{k=0}^m x_k y_k - \left(\sum_{k=0}^m x_k \right) \left(\sum_{k=0}^m y_k \right) \right]$$
$$b = \frac{1}{d} \left[\left(\sum_{k=0}^m x_k^2 \right) \left(\sum_{k=0}^m y_k \right) - \left(\sum_{k=0}^m x_k \right) \left(\sum_{k=0}^m x_k y_k \right) \right]$$

Where

$$d = (m+1) \sum_{k=0}^m x_k^2 - \left(\sum_{k=0}^m x_k \right)^2$$

Lots of simple computations

Linear Example

<i>x</i>	1.0	2.0	2.5	3.0
<i>y</i>	3.7	4.1	4.3	5.0

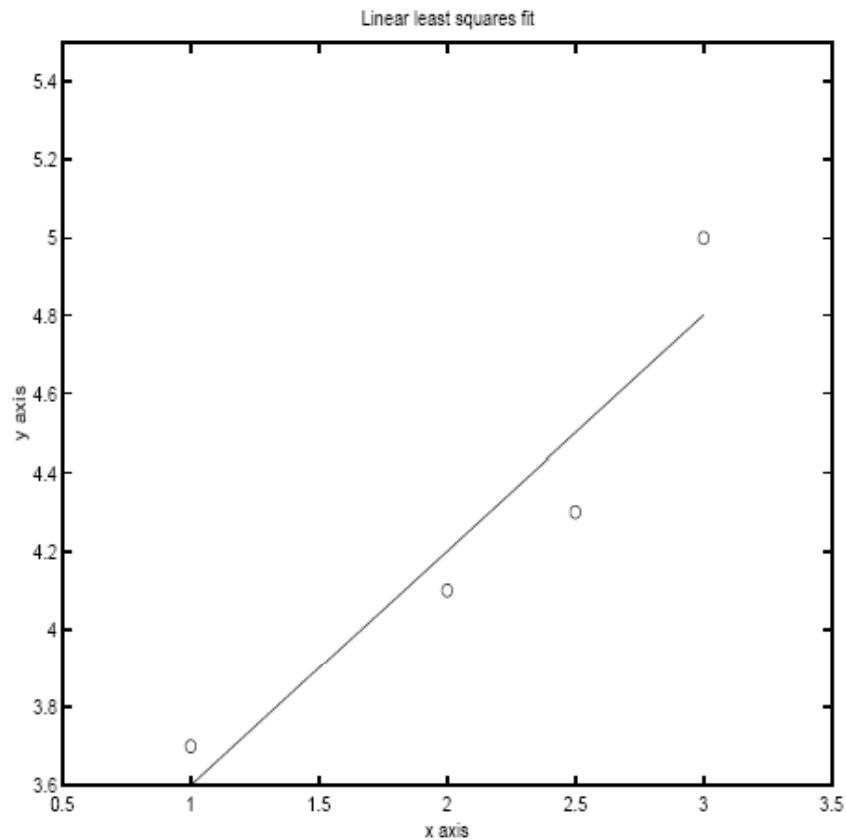
Least Squares Fit

The data in the previous slide lead to

$$20.25a + 8.5b = 37.65$$

$$8.5a + 4.0b = 17.1$$

which can be solved for $a = 0.6$ and $b = 3.0$

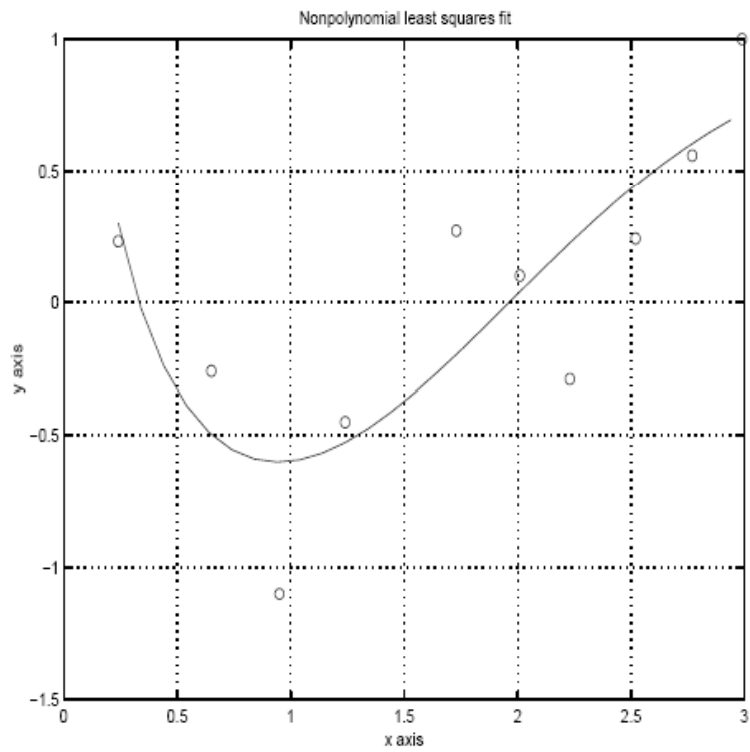


Nonpolynomial Example

We can fit a table by a nonpolynomial function

x	0.24	0.56	0.95	1.24	...	2.77	2.99
y	0.23	-0.26	-1.10	-0.45	...	0.56	1.00

$$a \ln x + b \cos x + ce^x$$



Basis Functions

A general least squares fitting can be written as

$$y = \sum_{j=0}^n c_j g_j(x)$$

In which the functions $g_0(x), g_1(x), \dots, g_n(x)$ are called basis functions. They are known and kept fixed

Given a set of data (x_k, y_k) , we want to find the values of c_j to minimize the total error as

$$\phi(c_0, c_1, \dots, c_n) = \sum_{k=0}^m \left[\sum_{j=0}^n c_j g_j(x_k) - y_k \right]^2$$

We again set the partial derivatives to be zero

$$\frac{\partial \phi}{\partial c_i} = 0 \quad (0 \leq i \leq n)$$

Basis Functions (II)

The partial derivatives are for $0 \leq i \leq n$

$$\frac{\partial \phi}{\partial c_i} = \sum_{k=0}^m 2 \left[\sum_{j=0}^n c_j g_j(x_k) - y_k \right] g_i(x_k)$$

Setting them simultaneously zero, we have

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

This is the normal equation, which is a system of linear equations with unknowns c_0, c_1, \dots, c_n

The coefficients of the linear system are

$$a_{ij} = \sum_{k=0}^m g_i(x_k) g_j(x_k)$$

The coefficient matrix is nonsingular if the basis function $g_0(x), g_1(x), \dots, g_n(x)$ are linearly independent. The basis functions should be appropriate for the problem in question and make the resulting coefficient matrix well conditioned

Orthonormal Basis Functions

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

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

We are looking for a particular $g(x) \in G$ 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 $0 \leq i \leq n$, easier or more difficult to solve

Choose Basis Functions

We say a basis $\{g_0, g_1, \dots, g_n\}$ 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

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

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 choose some basis functions so that the coefficient matrix is easy to solve, not necessarily as an identity matrix

Polynomial Basis Functions

Consider G as the space of all polynomials of degree $\leq n$. We naturally choose

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

Any polynomial in G can be represented as

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

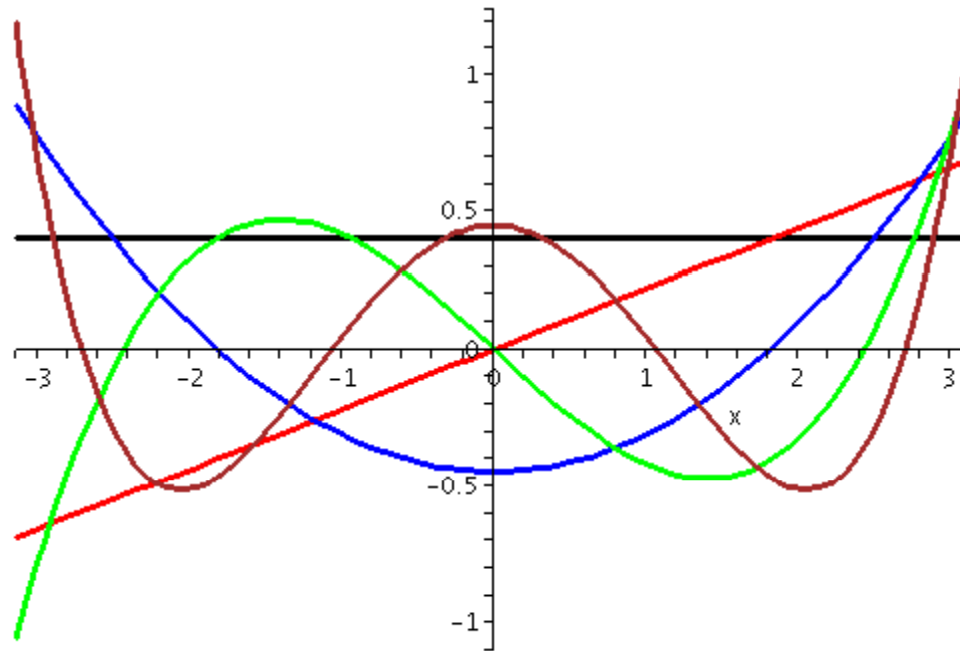
The simple basis is, however, not very good, since they are too much alike

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

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

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

Orthogonal Polynomials



Orthonormal basis polynomials associated with the Legendre polynomials

Chebyshev Polynomials

The first few Chebyshev polynomials are

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

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

The Chebyshev polynomials can be generated recursively as

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

They can also be written as

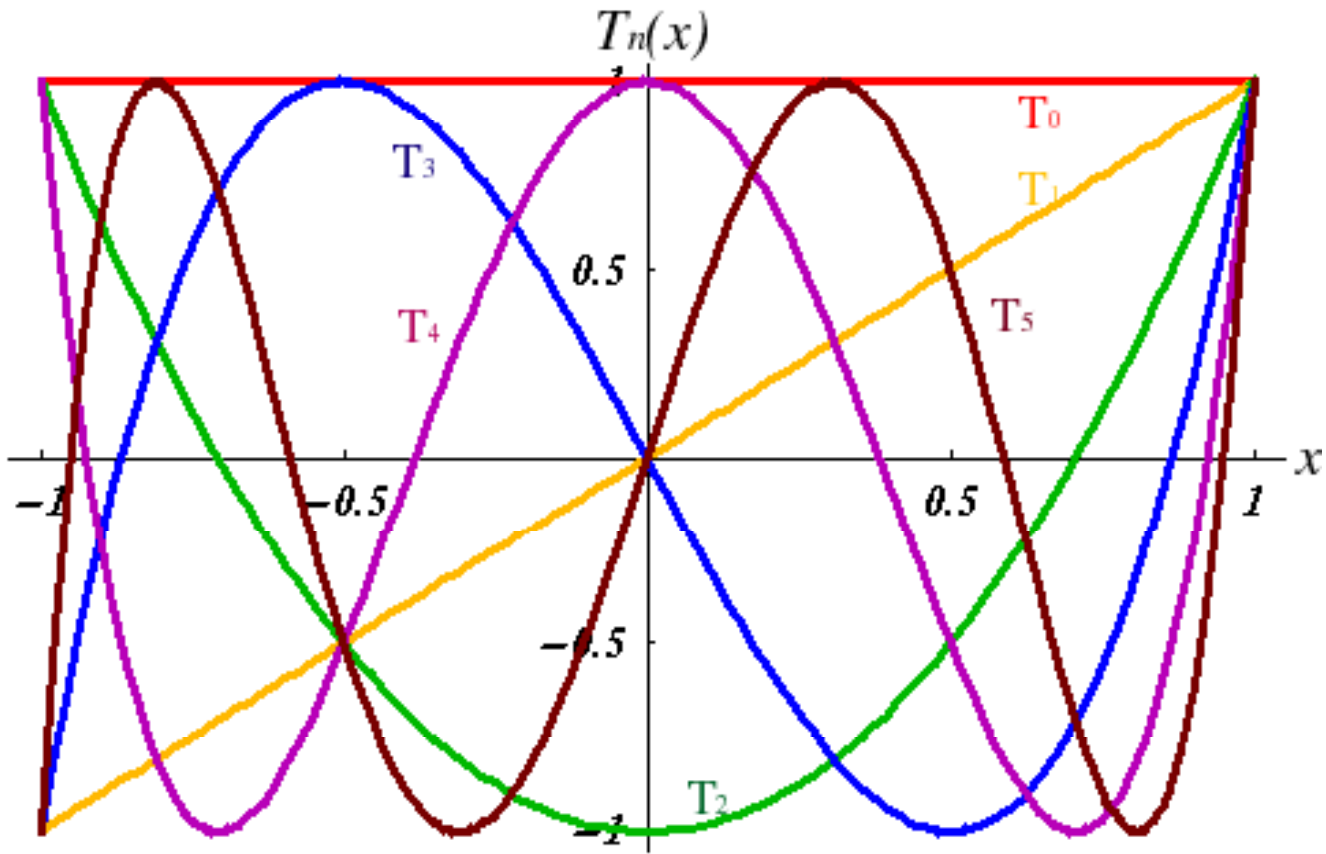
$$T_k(x) = \cos(k \arccos x)$$

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

Chebyshev Polynomials



The first few Chebyshev polynomials

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

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

The Chebyshev polynomials are defined on the interval $[-1,1]$, we would also like the abscissas $\{x_j\}$ to 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 $[a,b]$

Evaluating Chebyshev Polynomials

$$\begin{aligned}
 f(x) &= \sum_{j=0}^n c_j T_j(x) = \sum_{j=0}^n (w_j - 2xw_{j+1} + w_{j+2})T_j \\
 &= \sum_{j=0}^n w_j T_j - 2x \sum_{j=0}^n w_{j+1} T_j + \sum_{j=0}^n w_{j+2} T_j \\
 &= \sum_{j=0}^n w_j T_j - 2x \sum_{j=1}^{n+1} w_j T_{j-1} + \sum_{j=2}^{n+2} w_j T_{j-2} \\
 &= \sum_{j=0}^n w_j T_j - 2x \sum_{j=1}^n w_j T_{j-1} + \sum_{j=2}^n w_j T_{j-2} \\
 &= w_0 T_0 + w_1 T_1 + \sum_{j=2}^n w_j T_j - 2xw_1 T_0 - 2x \sum_{j=2}^n w_j T_{j-1} + \sum_{j=2}^n w_j T_{j-2} \\
 &= w_0 + xw_1 - 2xw_1 + \sum_{j=2}^n w_j (T_j - 2xT_{j-1} + T_{j-2}) \\
 &= w_0 - xw_1
 \end{aligned}$$

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

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

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 $0 \leq i \leq n$

Algorithm (II)

5. Use an equation-solving routine to solve the normal equations for coefficients c_0, c_1, \dots, 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 $b = (b_i)_{0:n}$

$$a_{ij} = \sum_{k=0}^m T_i(z_k) T_j(z_k) \quad (0 \leq i, j \leq n)$$

$$b_i = \sum_{k=0}^m y_k T_i(z_k) \quad (0 \leq i \leq n)$$

Specific procedures are detailed in book

Polynomial Regression

Assume the data collected contain errors, the procedure for *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

$$y_i = P_N(x_i) + \varepsilon_i \quad (0 \leq i \leq m)$$

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*

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

can be computed to see how good the approximation is

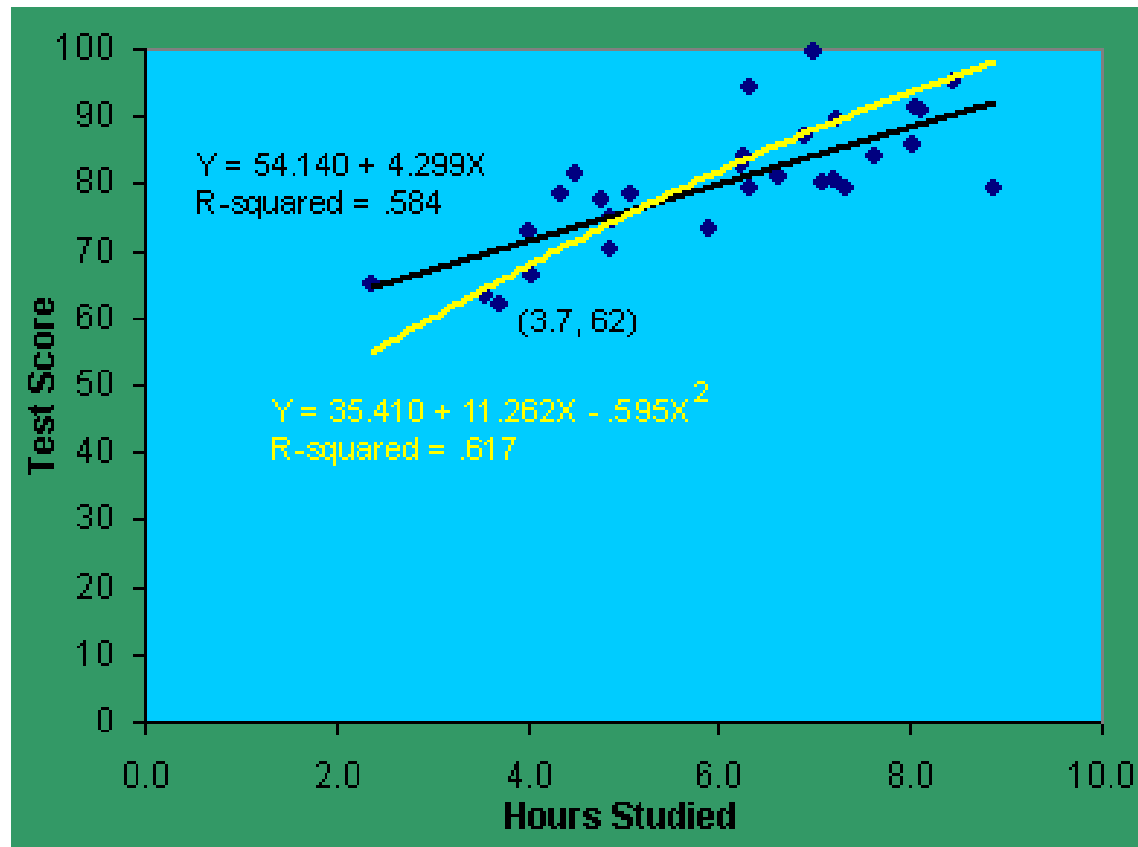
If the original data really represent a polynomial of degree N with noise, then

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

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

The drawback is that we need to compute p_0, p_1, \dots

An Example of Regression



A relationship between the hours studied and the test scores

Inner Product

Let two functions f and g whose domains contain $\{x_0, x_1, \dots, x_m\}$ 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

$$\alpha_n = \frac{\langle x q_n, q_n \rangle}{\langle q_n, q_n \rangle}$$

$$\beta_n = \frac{\langle x q_n, q_{n-1} \rangle}{\langle q_{n-1}, q_{n-1} \rangle}$$

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

Orthogonal Polynomials

We can check the orthogonality of polynomials as

$$\begin{aligned}\langle q_1, q_0 \rangle &= \langle x - \alpha_0, q_0 \rangle = \langle xq_0 - \alpha_0 q_0, q_0 \rangle \\ &= \langle xq_0, q_0 \rangle - \alpha_0 \langle q_0, q_0 \rangle = 0\end{aligned}$$

$$\begin{aligned}\langle q_2, q_1 \rangle &= \langle xq_1 - \alpha_1 q_1 - \beta_1 q_0, q_1 \rangle \\ &= \langle xq_1, q_1 \rangle - \alpha_1 \langle q_1, q_1 \rangle - \beta_1 \langle q_0, q_1 \rangle = 0\end{aligned}$$

The remaining part can be proved by using induction

Assume $\langle q_i, q_j \rangle = 0$ for $0 \leq i, j \leq m-1$

$$\begin{aligned}\langle q_i, q_m \rangle &= \langle q_i, xq_{m-1} - \alpha_{m-1} q_{m-1} - \beta_{m-2} q_{m-2} \rangle \\ &= \langle q_i, xq_{m-1} \rangle - \alpha_{m-1} \langle q_i, q_{m-1} \rangle - \beta_{m-2} \langle q_i, q_{m-2} \rangle \\ &= x \langle q_i, q_{m-1} \rangle = 0\end{aligned}$$

Well Defined?

We need to show

$$\langle q_n, q_n \rangle \neq 0$$

If this is not the case, then

$$\langle q_n, q_n \rangle = 0 \quad \text{and} \quad \sum_{i=0}^m [q_n(x_i)]^2 = 0$$

This means that

$$q_n(x_i) = 0 \quad \text{for} \quad i = 0, 1, 2, \dots, m$$

It follows that it has **$m+1$** roots. If **n** is smaller than **m** , then we know the **q_n** is the zero polynomial, which is not true, since

$$q_0(x) = 1$$

$$q_1(x) = x - \alpha_0$$

$$q_2(x) = x^2 + (\text{lower order terms})$$

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

Solving Inconsistent Equations

A system of linear equations of the form

$$\sum_{j=0}^n a_{kj} x_j = b_k \quad (0 \leq k \leq m)$$

with $m > n$ is inconsistent, if there is no possible vector (x_0, x_1, \dots, x_n) to make the residual zero. There is no solution in the conventional sense satisfying this system

It is of some interest in applications to find the vector that minimizes the 2-norm residual

$$\phi(x_0, x_1, \dots, x_n) = \sum_{k=0}^m \left(\sum_{j=0}^n a_{kj} x_j - b_k \right)^2$$

We can take the partial derivatives with respect to x_i and set them equal to zero to obtain the normal equations

$$\sum_{j=0}^n \left(\sum_{k=0}^m a_{ki} a_{kj} \right) x_j = \sum_{k=0}^m b_k a_{ki}$$

for $0 \leq i \leq n$

Direct Factorizations

The normal equations obtained can be solved by Gaussian elimination and its solution of the original system in the least squares sense

If we write the original linear system as

$$Ax = b$$

It is also possible to directly factor the matrix A as

$$A = QR$$

where Q is an $(m + 1) \times (n + 1)$ orthogonal matrix satisfying $Q^T Q = \mathbf{1}$ and R is an upper triangular $(n + 1) \times (n + 1)$ matrix satisfying $r_{ii} > 0$ and $r_{ij} = 0$ for $j < i$. We then have

$$R x = Q^T b$$

which can be solved by a back substitution

Pseudo Inverse (II)

We can define the pseudo inverse of A as

$$A^+ = V \Sigma^{-1} U^T$$

And the “solution” of the rectangular linear system is defined to be

$$x = A^+ b = V \Sigma^{-1} U^T b$$

It can be shown that this definition of solution does minimize the residual norm of the original inconsistent system

Note that QR factorization and Singular Value Decomposition are more expensive to perform, in many cases, than solving the normal equations. In case that the matrix A is a sparse matrix, we may be able to solve it more efficiently using certain iterative methods

Forming the normal equation $A^T A$ is an option

Proof (I)

Consider a system of linear equations $Ax = b$, and A is an $m \times n$ matrix. The minimal solution of them system is $x = A^+ b = V \Sigma^{-1} U^T b$, with

$$A^+ = V \Sigma^{-1} U^T$$

Proof.

Let x be any n -vector. Define

$$y = V^T x \quad \text{and} \quad c = U^T b$$

Using the orthogonal properties, we have

$$\begin{aligned} \rho &= \inf_x \|Ax - b\| = \inf_x \|U \Sigma V^T x - b\| \\ &= \inf_x \|U^T (U \Sigma V^T x - b)\| \\ &= \inf_x \|\Sigma V^T x - U^T b\| = \inf_x \|\Sigma y - c\| \end{aligned}$$

Note that Σ is a diagonal matrix

Proof (2)

From previous page, we have

$$\|\Sigma y - c\|^2 = \sum_{i=1}^r (\sigma_i y_i - c_i)^2 + \sum_{i=r+1}^m c_i^2$$

To minimize the expression, we need to minimize the first terms on the right-hand side, by defining

$$y_i = \frac{c_i}{\sigma_i} \quad \text{for } 1 \leq i \leq r$$

Other components of y can remain unspecified.

We can set

$$y_i = 0 \quad \text{for } r+1 \leq i \leq m$$

So we set $y = \Sigma^{-1}c$, and

$$x = Vy = V\Sigma^{-1}c = V\Sigma^{-1}U^T b = A^+ b$$

Some Properties

There are some interesting properties for the pseudoinverse. They are called **Penrose Properties**.

$$A = AA^+ A$$

$$A^+ = A^+ AA^+$$

$$AA^+ = (AA^+)^T$$

$$A^+ A = (A^+ A)^T$$

Proof:

$$\begin{aligned} AA^+ A &= U\Sigma V^T (V\Sigma^{-1}U^T)U\Sigma V^T \\ &= U\Sigma\Sigma^{-1}\Sigma V^T = U\Sigma V^T = A \end{aligned}$$