CHAPTER 8

Approximation Theory

1. Introduction

We consider two types of problems in this chapter:

(1) Representing a function by a “simpler” function that is easier to evaluate.

(2) Fitting functions to given data and finding the “best” function in a certain class that can be used to represent the data.

2. Discrete Least Squares Approximation

Given a set of data points \( \{(x_i, y_i)\}_{i=1}^{m} \), we look for the best function \( f \) of some class of functions such that

\[
\sum_{i=1}^{m} [y_i - f(x_i)]^2
\]

is minimized.
In the figure on the previous page, we see the quadratic polynomial that “best” approximates our data points in the sense of least squares. Notice, as opposed to interpolation, there is no requirement that any of the data points actually lie on the graph of the approximating function. Instead, we choose the quadratic where the sum of the squares of the vertical distances from the points to the curve is minimized. This choice is made using the theory of max-min problems from calculus.

**Linear Least Squares**

This is illustrative and the easiest to work with. Suppose we have \( m \) data points \( \{(x_i, y_i)\}_{i=1}^{m} \) that appear to be somewhat linear, so we want to find the line \( y = a_1x + a_0 \) that best approximates the data points in the sense of least squares, i.e., we want to minimize the function

\[
E(a_0, a_1) = \sum_{i=1}^{m} [y_i - (a_1x_i + a_0)]^2.
\]

Set

\[
\frac{\partial E}{\partial a_0} = 2 \sum_{i=1}^{m} (y_i - a_1x_i - a_0)(-1) = 0
\]

\[
\frac{\partial E}{\partial a_1} = 2 \sum_{i=1}^{m} (y_i - a_1x_i - a_0)(-x_i) = 0
\]

We get the system of linear equations in \( a_0 \) and \( a_1 \):

\[
a_0m + a_1 \sum_{i=1}^{m} x_i = \sum_{i=1}^{m} y_i
\]

\[
a_0 \sum_{i=1}^{m} x_i + a_1 \sum_{i=1}^{m} x_i^2 = \sum_{i=1}^{m} x_i y_i
\]

We solve for \( a_0 \) and \( a_1 \) to get
6. Trigonometric Polynomial Approximation

Trigonometric functions are used to approximate functions that have periodic behavior, functions with the property that for some constant $T$,

$$f(x + T) = f(x) \quad \text{for all } x.$$  

We can generally also transform the problem so that $T = 2\pi$ and restrict the approximation to the interval $[-\pi, \pi]$.

**Definition** (Page 333). An integrable function $w$ is a weight function on the interval $I$ if $w(x) \geq 0$ for all $x \in I$, but $w \neq 0$ (not identically 0) on any subinterval of $I$.

The purpose of a weight function is to assign varying degrees of importance on certain portions of the interval. For example, if $w(x) = 1 - x$ on $I = [0, 1]$, more importance is given to the points near 0 than the points near 1.

**Definition** (Page 334). The set of functions $\{\phi_0, \phi_1, \ldots, \phi_n\}$ is said to be orthogonal for the interval $[a, b]$ with respect to the weight function $w$ if for some positive numbers $\alpha_0, \alpha_1, \ldots, \alpha_n$,

$$\int_a^b w(x) \phi_j(x) \phi_k(x) \, dx = \begin{cases} 
0, & \text{when } j \neq k \\
\alpha_k > 0, & \text{when } j = k.
\end{cases}$$

If, in addition, $\alpha_k = 1$ for each $k = 0, 1, \ldots, n$, the set is said to be orthonormal.
Continuous Least Squares for Orthogonal Functions

**Theorem** (Page 335). If \( \{\phi_0, \phi_1, \ldots, \phi_n\} \) is an orthogonal set of functions on an interval \([a, b]\) with respect to the weight function \( w \), then the least squares approximation to \( f \) on \([a, b]\) with respect to \( w \) is

\[
P(x) = \sum_{k=0}^{n} a_k \phi_k(x),
\]

where

\[
a_k = \frac{\int_{a}^{b} w(x) \phi_k(x) f(x) \, dx}{\int_{a}^{b} w(x)(\phi_k(x))^2 \, dx} = \frac{1}{\alpha_k} \int_{a}^{b} w(x) \phi_k(x) f(x) \, dx.
\]

For each positive integer \( n \), the set \( T_n \) of trigonometric polynomials of degree less than or equal to \( n \) is the set of all linear combinations of \( \{\phi_0, \phi_1, \ldots, \phi_{2n-1}\} \), where

\[
\phi_0(x) = \frac{1}{2},
\]

\[
\phi_k(x) = \cos kx, \quad \text{for each } k = 1, 2, \ldots, n, \text{ and}
\]

\[
\phi_{n+k}(x) = \sin kx, \quad \text{for each } k = 1, 2, \ldots, n.
\]

The set \( \{\phi_0, \phi_1, \ldots, \phi_{2n-1}\} \) is orthogonal on \([-\pi, \pi]\) with respect to the weight function \( w(x) \equiv 1 \).

For \( f \in C[-\pi, \pi] \), the continuous least squares approximation by functions in \( T_n \) is

\[
S_n(x) = \frac{1}{2} a_0 + \sum_{k=1}^{2n-1} a_k \phi_k(x)
\]

where

\[
a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \phi_k(x) \, dx \quad \text{for each } k = 0, 1, \ldots, 2n - 1.
\]
Another way to express this is

\[ S_n(x) = \frac{a_0}{2} + a_n \cos nx + \sum_{k=1}^{n-1} (a_k \cos kx + b_n \sin kx) \]

where

\[ a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos kx \, dx \quad \text{for each } k = 0, 1, \ldots, n, \quad \text{and} \]

\[ b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin kx \, dx \quad \text{for each } k = 1, 2, \ldots, n - 1. \]

\[ \lim_{n \to \infty} S_n(x) \]

is called the Fourier series of \( f \). Fourier series are used to describe the solution of various ordinary and partial differential equations that occur in physical situations.

**Discrete Least Squares Approximation and Interpolation**

We now look at a discrete analog to Fourier series that is useful for large amounts of equally spaced points.

Suppose we have \( 2m \) data points \( \{ (x_j, y_j) \}_{j=0}^{2m-1} \) with the \( x_j \) equally partitioning \([-\pi, \pi]\), i.e.,

\[ x_j = -\pi + \left( \frac{j}{m} \right) \pi \quad \text{for each } j = 0, 1, \ldots, 2n - 1. \]

Although not in our set of points, \( x_{2m} = \pi \). Also, note that \( x_0 = -\pi \).

Our goal: to determine the trigonometric polynomial in \( \mathcal{T}_n \) that minimizes

\[ E(S_n) = \sum_{j=0}^{2n-1} [y_j - S_n(x_j)]^2, \]
i.e., we need to choose constants $a_0, a_1, \ldots, a_n, b_1, b_2, \ldots, b_{n-1}$ such that

$$E(S_n) = \sum_{j=0}^{2n-1} \left\{ y_j - \left[ \frac{a_0}{2} + a_n \cos nx + \sum_{k=1}^{n-1} (a_k \cos kn + b_n \sin kx) \right] \right\}^2$$

is minimized.

Finding the constants is simplified by the fact that $\{\phi_0, \phi_1, \ldots, \phi_{2n-1}\}$ is orthogonal with respect to summation over equally spaced points $\{(x_j, y_j)\}_{j=0}^{2m-1}$ in $[-\pi, \pi]$, i.e.,

for $k \neq l$, \quad $\sum_{j=0}^{2m-1} \phi_k(x_j)\phi_l(x_j) = 0$.

Why?

**Theorem.** If $r$ is not a multiple of $2m$, $r$ an integer,

$$\sum_{j=0}^{2m-1} \cos rx_j = 0 \quad \text{and} \quad \sum_{j=0}^{2m-1} \sin rx_j = 0,$$

and if $r$ is not a multiple of $m$,

$$\sum_{j=0}^{2m-1} (\cos rx_j)^2 = m \quad \text{and} \quad \sum_{j=0}^{2m-1} (\sin rx_j)^2 = m.$$

**Proof.**

**Recall.**:

(1) $\sum_{i=1}^{n} r^i = 1 + r + r^2 + \cdots + r^n = \frac{1 - r^{n+1}}{1 - r}$.

(2) $i^2 = -1$.

(3) $e^{i\pi} = \cos z + i \sin z$. 
NOTE.:

(1) \( e^{irx_j} = e^{i\pi (\pi + i\pi/m)} = e^{-ir\pi} \cdot e^{ir\pi/m} \).

(2) \( e^{2ir\pi} = \cos 2r\pi + i \sin 2r\pi \).

(3) \( \sin t_1 \cos t_2 = \frac{1}{2} \left[ \sin(t_1 - t_2) + \sin(t_1 + t_2) \right] \).

Then

\[
\sum_{j=0}^{2m-1} \cos rx_j + i \sum_{j=0}^{2m-1} \sin rx_j = \sum_{j=0}^{2m-1} e^{irx_j} = \sum_{j=0}^{2m-1} e^{i\pi(\pi + i\pi/m)} =
\]

\[
e^{-ir\pi} \sum_{j=0}^{2m-1} e^{ir\pi/m} \quad \{r \text{ not a multiple of } 2m\}
\]

\[
e^{-ir\pi} \cdot \frac{1 - e^{2ir\pi}}{1 - e^{ir\pi/m}} = e^{-ir\pi} \cdot \frac{1 - [1 + 0]}{1 - e^{ir\pi/m}} = 0
\]

\[
\implies \sum_{j=0}^{2m-1} \cos rx_j = 0 \quad \text{and} \quad \sum_{j=0}^{2m-1} \sin rx_j = 0.
\]

\(r\) not a multiple of \(m\) \implies

\[
\sum_{j=0}^{2m-1} (\cos rx_j)^2 = \sum_{j=0}^{2m-1} \frac{1}{2} \left[ 1 + \cos 2rx_j \right] =
\]

\[
\frac{1}{2} \left[ \sum_{j=0}^{2m-1} 1 + \sum_{j=0}^{2m-1} \cos 2rx_j \right] = \frac{1}{2} \left[ 2m + 0 \right] = m.
\]

Similarly, \( \sum_{j=0}^{2m-1} (\sin rx_j)^2 = m \). \(\Box\)
Sample case:
\[
\sum_{j=0}^{2m-1} \phi_k(x_j)\phi_{n+l}(x_j) = \sum_{j=0}^{2m-1} (\cos kx_j)(\sin lx_j) = \\
\frac{1}{2} \left[ \sum_{j=0}^{2m-1} \sin(l-k)x_j + \sum_{j=0}^{2m-1} \sin(l+k)x_j \right] = \frac{1}{2}[0 + 0] = 0.
\]

Overall, we have

**Theorem.** The constants in the summation

\[
S_n(x) = \frac{a_0}{2} + a_n \cos nx + \sum_{k=1}^{n-1} (a_k \cos kx + b_k \sin kx)
\]

that minimize the least squares sum

\[
E(a_0, \ldots, a_n, b_1, \ldots, b_{n-1}) = \sum_{j=0}^{2m-1} [y_j - S_n(x_j)]^2
\]

are

\[
a_k = \frac{1}{m} \sum_{j=0}^{2m-1} y_j \cos kx_j, \quad \text{for each } k = 0, 1, \ldots, n,
\]

and

\[
b_k = \frac{1}{m} \sum_{j=0}^{2m-1} y_j \sin kx_j, \quad \text{for each } k = 1, 2, \ldots, n-1.
\]

**Maple.** See trigpoly.mw or trigpoly.pdf.
The interpolatory trigonometric polynomial on the $2m$ data points $\{(x_j, y_j)\}_{j=0}^{2m-1}$ with $x_j = -\pi + \left(\frac{j}{m}\right)\pi$ is the least squares polynomial from $T_n$ for this collection of points.

However, if we want the coefficients to have the same form with $n = m$ as the discrete least squares approximation, a modification is needed. Interpolation requires computing

$$\sum_{j=0}^{2m-1} (\cos mx_j)^2 = 2m$$

instead of

$$\sum_{j=0}^{2m-1} (\cos rx_j)^2 = m$$

for $r$ not a multiple of $m$.

We then replace $a_m$ with $\frac{a_m}{2}$, giving an interpolating polynomial of the form

$$S_m(x) = \frac{a_0 + a_n \cos mx}{2} + \sum_{k=1}^{m-1} (a_k \cos kx + b_k \sin kx)$$

where

$$a_k = \frac{1}{m} \sum_{j=0}^{2m-1} y_j \cos kx_j, \quad \text{for each } k = 0, 1, \ldots, m,$$

and

$$b_k = \frac{1}{m} \sum_{j=0}^{2m-1} y_j \sin kx_j, \quad \text{for each } k = 1, 2, \ldots, m - 1.$$
The fast Fourier transform (FFT) algorithm requires \(O(m \log_2 m)\) multiplications and \(O(m \log_2 m)\) additions.

Suppose the number of data points is a power of 2, i.e., \(m = 2^p\). Then, for \(x_j = -\pi + \left(\frac{j}{m}\right)\pi\) with \(j = 0, 1, \ldots, 2m - 1\), we have \(2m = 2^{p+1}\) points \(\{(x_j, z_j)\}_{j=0}^{2m-1}\) where the \(z_j\) may be complex (we usually use \(y_j\) if they are real).

The FFT procedure computes the complex coefficients \(c_k\) in the formula

\[
F(x) = \frac{1}{m} \sum_{k=0}^{2m-1} c_k e^{ikx},
\]

where

\[
c_k = \sum_{j=0}^{2m-1} y_je^{\pi ijk/m}, \quad \text{for each } k = 0, 1, \ldots, 2m - 1.
\]

Then

\[
\frac{1}{m}c_k e^{-i\pi k} = a_k + ib_k.
\]

**MAPLE.** In Maple, `FourierTransform` in the `Discrete Transforms Package` uses

\[
c_k = \frac{1}{\sqrt{m}} \sum_{j=0}^{2m-1} y_je^{-2\pi ijk/m}, \quad \text{for each } k = 0, 1, \ldots, 2m - 1,
\]

so

\[
a_k = (-1)^k \sqrt{\frac{2}{m}} \Re(c_k) \quad \text{and} \quad b_k = (-1)^k \sqrt{\frac{2}{m}} \Im(c_k).
\]

See `fft.mw` or `fft.pdf`.

---

8. APPROXIMATION THEORY