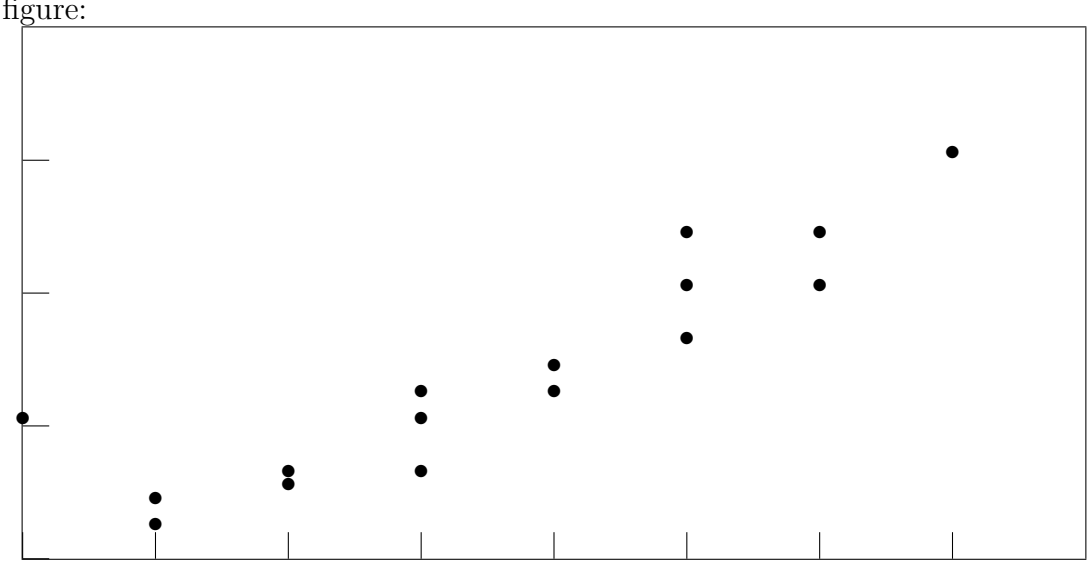
Project #1: Basis Functions

1 Time Series

A quantity evolving in time is one of the most common ways to represent data available in public data sets. Let $i, m, n \in \mathbb{N}$, a *time series* consisting of n time points and m variables is a totally ordered set $T := \{(t_i, \mathbf{x}_i)\}$, such that $\mathbf{x}_i \in \mathbb{R}^m$ for $i \in [0, n]$.

Suppose that we obtain a data set from an experiment as shown in the following figure:



It is clear that there is no function (why?) which produces the given data.

Suppose we want to approximate a function $f(t)$ with a sum of a given set of functions multiplied by a constant, that is,

$$f(t) = c_1 u_1(t) + c_2 u_2(t) + \cdots + c_k u_k(t) + \cdots + c_n u_n(t) = \sum_{k=1}^{n} c_k u_k(t).$$

We call $u_k(t)$ a **basis function**. It is possible to determine optimal constants c_k by minimizing the error function:

$$J(c_1, \cdots, c_n) = \min_{c_1, \cdots, c_n} \int_\Omega \left(\sum_{k=1}^{n} c_k u_k(t) - f(t) \right)^2 dt.$$

We say that $J(c_1, \cdots, c_n)$ is optimal in a *least square error sense*. Here, Ω is some subset of $\mathbb{R}$, and can be taken as integrating over $[a, b] \subset \mathbb{R}$ for the purposes of this discussion.

To find this minimum, derive with respect to each c_k (for now, consider taking an ordinary one-variable derviative with respect to the variable c_k), and set the equation equal to zero:

$$\begin{aligned}
\frac{\partial J}{\partial c_k} =& \frac{\partial}{\partial c_k}\left(\int_\Omega \left(\sum_{k=1}^{n} c_k u_k(t) - f(t)\right)^2 dt\right) \\
=& \int_\Omega \left(\frac{\partial}{\partial c_k}\left(\sum_{k=1}^{n} c_k u_k(t) - f(t)\right)^2 dt\right) \\
=& \int_\Omega \left(2u_k(t)\left(\sum_{k=1}^{n} c_k u_k(t) - f(t)\right) dt\right) \\
=& 2\int_\Omega c_1 u_1(t) u_k(t) dt + 2\int_\Omega c_2 u_2(t) u_k(t) dt + ...2\int_\Omega c_k u_k^2(t) dt... \\
&\qquad + 2\int_\Omega c_n u_k(t) u_n(t) dt - 2\int_\Omega u_k(t) f(t) dt. \\
=& 0
\end{aligned}$$

Let $(u_j, u_k) = \int_\Omega u_j(t) u_k(t) dt$. Rearranging the previous terms,

$$c_1\left(u_1(t), u_1(t)\right) + \cdots c_k\left(u_1(t), u_k(t)\right) + \cdots + c_n\left(u_1(t), u_n(t)\right) = \left(u_k(t), f(t)\right).$$

Thus, in matrix form,

$$\begin{bmatrix} (u_1, u_1) & \cdots & (u_1, u_n) \\ \vdots & \ddots & \vdots \\ (u_n, u_1) & \cdots & (u_n, u_n) \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} (u_1, f) \\ \vdots \\ (u_n, f) \end{bmatrix}.$$

The basis functions $u_k(t)$ can be arbitrary. If we select monomials, then $f(t) = c_0 + c_1 t + c_2 t^2 + \cdots + c_k t^k + \cdots + c_n t^n$. This results in traditional linear regression if the basis set is $\{c_0, c_1 t\}$, traditional quadratic regression if the basis set is $\{c_0, c_1 t, c_2 t^2\}$, Fourier series with basis $\{\cos(k\pi t/l)\}$, discrete Fourier transform follows with little effort and so on. Figure 1 depicts multiple regression cases using monomials as basis functions.

We can select functions such that $(u_i, u_j) = 0$, $\forall i \neq j$. Such functions are called orthogonal functions, e.g. $\sin(t)$, $\cos(t)$, etc. Thus,

$$c_i = \frac{(f, u_i)}{(u_i, u_i)}. \tag{1}$$

The consequences of this method are far reaching. When $\sin(t)$ and $\cos(t)$ are used, this method produces the Fourier series for a finite interval of length l. If we study this problem in the limit when $l \to \infty$, the Fourier transform is produced.

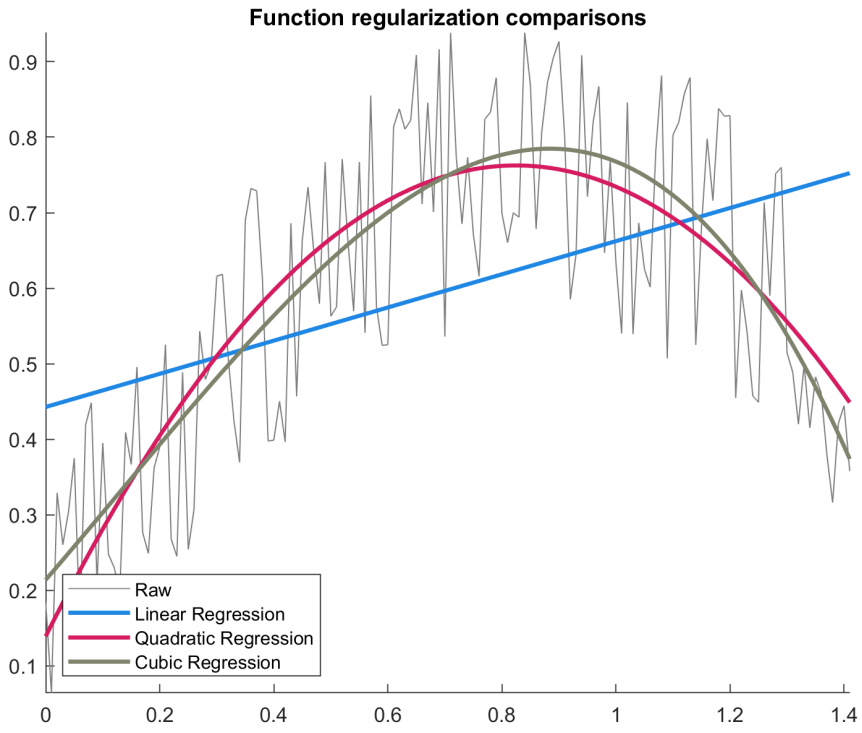


Figure 1: Multiple regression cases using monomials as basis functions

Example 1. Given $f(t) = t^2$, $t \in [0, 1]$, find c such that the function $g(t) = ct$ approximates $f(t)$ in a *least square error sense.*

$$c = \frac{(t^2, t)}{(t, t)} = \frac{\int_0^1 t^3 dt}{\int_0^1 t^2 dt} = \frac{3}{4}$$

Thus, $\frac{3}{4}t$ approximates t^2 optimally in a *least square error sense* in the interval $[0, 1]$.

These examples lead us to find functions approximately fitting the given data. Suppose that $\{(x_i, b_i) : i = 1, \cdots, n\}$ are given observation data from a physical model. Let $f(x) = a_1\phi_1(x) + \cdots + a_m\phi_m(x)$ be a function to represent the physical model. We are looking for coefficients $a_i, i = 1, \cdots, m$ such that

$$\min_{a_1, \cdots, a_m} \max_{1 \le i \le n} \{|f(x_i) - b_i|\}, \tag{1}$$

or

$$\min_{a_1, \cdots, a_m} \sum_{i=1}^{n} |f(x_i) - b_i|, \tag{2}$$

or

$$\min_{a_1, \cdots, a_m} \sum_{i=1}^{n} (f(x_i) - b_i)^2. \tag{3}$$

In this lab, we mainly study the problem of finding $a_1, \cdots, a_m$ satisfying (3) which is called *Least Square Problem.* The other cases lead to different approximate schemes and will not be covered here.

Let us first give a remark on choosing the model function f. Usually, people use their experiences and the pattern of observation data to choose $\phi_i, i = 1, \cdots, m$. For example, if the pattern of the observation data behaviors like a periodic function, we choose $\sin kx$ and/or $\cos kx$ as ϕ_i. We may choose exponential functions $e^{c_i x}$ as ϕ_i if the asymptotic behavior of the physical model has exponential decay or growth.

2 Project Assignment

In this assignment, you will create a smooth representation of a time series with noise using basis functions. Follow these steps to complete the task:

1. **Create a time series with noise.** Generate a function with added noise to simulate real-world data. This will serve as the starting point for approximation.

2. **Program an integral.** Implement numerical integration using methods such as the trapezoidal rule or Simpson's rule. This will be useful for computing inner products later.

3. **Create basis functions.** Define a set of basis functions, such as cosine functions ($\cos(nt)$) and polynomial-exponential functions ($x^n e^{nt}$). Choose a sufficient number of functions to approximate the original function accurately.

4. **Compute the inner product.** Write a function that takes two basis functions as inputs and computes their inner product using numerical integration. This will be used to construct the linear system.

5. **Formulate a linear system.** Construct a system of equations using the method outlined in the material. This involves computing inner products of basis functions and the target function.

6. **Solve for coefficients and approximate the function.** Solve the linear system to find the coefficients of the basis functions. Use these coefficients to form an approximation to the original noisy function.

3 Deliverables

To receive credit, you will submit:

1. A pdf file containing your handwritten work as well as screenshots of your code including output.

2. An image of your curve approximating the time series, contained in the pdf above.

The goal of this project is to reinforce concepts of function approximation, numerical integration, and solving linear systems. The approximation should produce a smooth representation of the initial noisy time series.