Discrete Least Squares Approximation: A Review

Carson Slater

December 15 2022

1 Introduction

1.1 Background

What methods exists to approximate a line to a set of numerical data points? Given the problem of a set of generated data, which methods are better suited to describing the data using a function? We will outline three different types of non-piecewise methods to describe a set of data using a function, and explore two of these methods in depth. In this review, we will outline Lagrange Interpolation, Linear Least Squares Approximation, and Polynomial Least Squares Approximation.

1.2 The Problem

Consider n data points, $\{(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)\}$. Assume there exists a function $f(x)$ such that it perfectly describes the relationship between each x_i and y_i $\forall$ $i = 0, 1, 2, ..., n$. Our aim is to find a function such that we minimize the error between $f(x)$ and what we are using to approximate it. Consider Theorem 1:

Theorem 1 (Weierstrass Approximation Theorem)**.** *Suppose f is defined and continuous on $[a, b]$. For each $\epsilon > 0$, $\exists$ a polynomial $P(x)$ such that*

$$|f(x) - P(x)| < \epsilon, \forall\, x \in [a, b].$$

Assuming $f(x)$ exists, we want to minimize $|f(x) - P(x)|$. To try to represent $f(x)$, we can use any n^{th} degree polynomial, but what we find is that some are better suited for select circumstances.

2 Interpolation

In this review we will only cover Lagrange Interpolation, although there are different types of interpolation, some of which are piecewise. Consider the case where there are two points, which will from now on be called *nodes*, $\{(x_0, y_0), (x_1, y_1)\}$. Burden, et al. [1] define the Lagrange Polynomials for each point as

$$L_0(x) = \frac{x - x_1}{x_0 - x_1} \quad \text{and} \quad L_1(x) = \frac{x - x_0}{x_1 - x_0}.$$

This gives us the **Lagrange interpolating polynomial** of degree 1 through (x_0, y_0), and (x_1, y_1) as

$$P_1(x) = L_0(x)f(x_0) + L_1(x)f(x_1) = \frac{x - x_1}{x_0 - x_1}f(x_0) + \frac{x - x_0}{x_1 - x_0}f(x_1).$$

This principle of n^{th} degree Lagrange Interpolating Polynomials for $n+1$ nodes has been generalized. To do this, for each $k = 0, 1, 2, ..., n, n+1$, a function $L_{n,k}(x)$ such that $L_{n,k}(x_i) = 0$ when $i \neq k$ and $L_{n,k}(x_k) = 1$. To satisfy these conditions, we define a **Lagrange Interpolation Basis** as

$$L_{n,k}(x) = \frac{(x - x_0)(x - x_1)...(x - x_{k-1})(x - x_{k+1})...(x - x_n)}{(x_k - x_0)(x_k - x_1)...(x_k - x_{k-1})(x_k - x_{k+1})...(x_k - x_n)} = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i}.$$

This means that when x_k is the input value for the function, the value at x_i will be $1 \cdot f(x_i) = y_i$. Extending this to the concept of n^{th} degree Lagrange interpolating polynomial gives us

$$P_n(x) = L_{n,0}(x)f(x_0) + L_{n,1}f(x_1) + ... + L_{n,n}f(x_n) = \sum_{i=0}^{n} f(x_i)L_{n,i}.$$

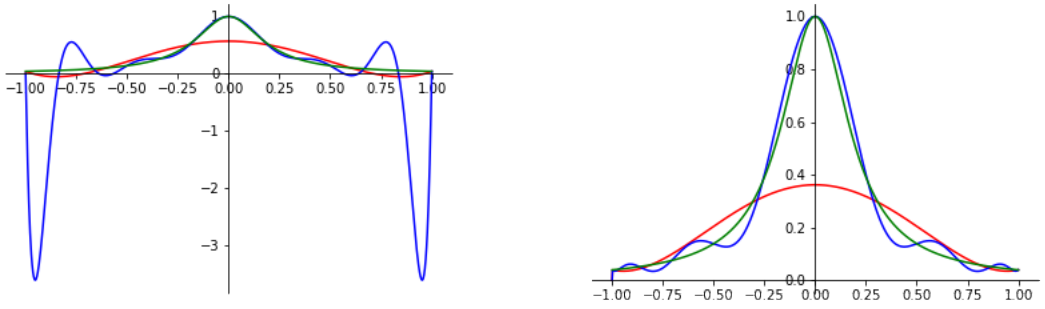
There are a few caveats to note for Lagrange interpolation. The first is Theorem 2.

Theorem 2 (Uniqueness for A Given $P_n(x)$)**.** *Given $n+1$ unique values, $x_0, x_1, ..., x_n, x_{n+1}$, and function f such that it exists on these nodes, the interpolating polynomial of degree n, $P_n(x)$ is unique.*

The second is that this polynomial only interpolates for points on $[x_0, x_n]$, and is not the best approximation for any $x \notin [x_0, x_n]$. Since Lagrange interpolation and also **Newton's Divided Differences** method of interpolation aim at giving the lowest degree polynomial to fit the data points, one of the drawbacks is that when there are many data points, there are more nodes, and as n increases, the degree of $P_n(x)$ also increases, which can lead to oscillatory behavior. Consider the example of Runge's function,

$$f(x) = \frac{1}{1 + 25x^2}.$$

When choosing $n+1$ nodes, we can see from Figure 1 that the degree 5 interpolation is more stable that the degree 12 polynomial, although the degree 12 polynomial possess less error. This is a major drawback for interpolation, and can lead to problems as n, the number of nodes increases. This problem can be approached either using piecewise methods, or can be tackled using a different method altogether, such as regression.



(a)

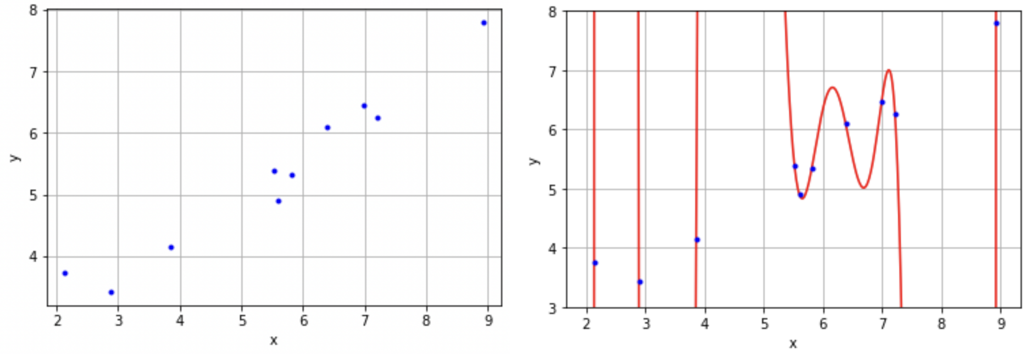
(b)

Figure 1: In both figures, we have Runge's function on $[-1, 1]$ plotted in green, $P_5(x)$ in red and $P_{12}(x)$ in blue. (a) Interpolation polynomials for Runge's function using evenly-spaced points, and (b) Interpolation using *Clenshaw-Curtis* points, which are beyond the scope of this review. Polynomials were generated using Newton's Divided Differences, but yield the same polynomial as Lagrange interpolation because of Theorem 2. Plots were generated using Python's `Matplotlib` package.

3 Discrete Least Squares Approximation

3.1 Use Criterion Contrasted with Interpolation

Consider the simulated case in Figure 2. Given that both variables were drawn from a at random, this would be a situation where the nodes for interpolation cannot be chosen strategically. If we were to try to approximate the $y = f(x)$ that perfectly describes this relationship, we will find that interpolation would not be the optimal choice to approximate $f(x)$.



(a)

(b)

Figure 2: (a) Plot of simulated data with positive correlation. (b) Plot of $P_9(x)$, the Lagrange interpolating polynomial for the simulated data.

It is noteworthy to observe that when strategic nodes cannot be chosen, given a high enough n, P_n will demonstrate behavior that not ideal to approximate $f(x)$, due to the high degree nature of the polynomials. This is where discrete least squares approximation comes in. There exists methods for linear and polynomial approximates, and both of these methods can be better approximations for the $f(x)$ that is assumed to exactly describe the relationship between x and y. Although Lagrange interpolation can exactly describe the relationship at each data point that already exists, the approximation is likely to deviate in between nodes, whereas least squares can better approximate the relationship between nodes, because of the lower degree polynomials that are most commonly used. Put bluntly, a method is needed such that a lower degree $P(x)$ can be used to approximated $f(x)$.

3.2 Linear Least Squares Approximation

Continuing with the case where we have n data points randomly selected. Instead of finding a function that approximates exactly for the data points, an alternative approach is to find a line that best approximates over the whole domain, and does not diverge outside of $[x_0, x_n]$.

3.2.1 Error Minimization

Reframing the solution as such enables us to consider that we do not have to necessarily agree with each data point in our approximation. Instead, we can minimize the error of a low-degree polynomial with weighted coefficients. For linear least squares, we consider degree 1 polynomial, $P(x) = y$. We describe y as

$$y = \beta_0 + \beta_1 x.$$

To find the optimal weights (β_0, β_1) we need to find the weights that minimize the error between $P(x)$ and the data points. What we will find is that this endeavor is comparable to minimizing the Euclidean distance (ℓ_2 norm) between $P(x_i)$ and each $y_i \in \{(x_i, y_i) : i = 1, ..., n\}$.

3.2.2 Least Squares Minimization

Burden, et al. [1] highlight that finding the distance optimal line between $P(x)$ and y_i can be done many ways using different norms. For example, one could use the ℓ_∞ norm, minimizing

$$E_\infty(\beta_0, \beta_1) = \max_{1 \leq i \leq n} \{|y_i - (\beta_1 x_i + \beta_0)|\}.$$

The issue is that such a line more heavily considers the maximum deviation, and could overfit for that one observation. Another option is minimizing the distance between the line and all of points. Trying to use the ℓ_1 norm, we could minimize the **absolute deviation**, being

$$E_1(\beta_0, \beta_1) = \sum_{i=1}^{n} |y_i - (\beta_1 x_i + \beta_0)|$$

by finding β_0 and β_1 such that

$$0 = \frac{\partial}{\partial \beta_0} \sum_{i=1}^{n} |y_i - (\beta_1 x_i + \beta_0)| \quad \text{and} \quad 0 = \frac{\partial}{\partial \beta_1} \sum_{i=1}^{n} |y_i - (\beta_1 x_i + \beta_0)|.$$

This is much more considerate of all the other points than the **minimax** function derived from the ℓ_∞ norm. There is still one big problem, which is that the absolute deviation is not differentiable at 0 with respect to β_1. Hence, we finally consider the Euclidean norm (ℓ_2) to use as the distance we minimize between $P(x)$ and y_i. We can minimize

$$E_2(\beta_0, \beta_1) = \sum_{i=1}^{n} [y_i - (\beta_1 x_i + \beta_0)]^2$$

in the same manner as the absolute deviation without having to worrying about its differentiability at 0 with respect to β_1. The optimal β_0 and β_1 can be found using the following equalities,

$$0 = \frac{\partial}{\partial \beta_0} \sum_{i=1}^{n} [y_i - (\beta_1 x_i + \beta_0)]^2 \quad \text{and} \quad 0 = \frac{\partial}{\partial \beta_1} \sum_{i=1}^{n} [y_i - (\beta_1 x_i + \beta_0)]^2,$$

giving us

$$0 = 2\sum_{i=1}^{n} [y_i - (\beta_1 x_i + \beta_0)](-1) \quad \text{and} \quad 0 = 2\sum_{i=1}^{n} [y_i - (\beta_1 x_i + \beta_0)(x_i)].$$

Then we can split the sums up, giving us the **normal equations**, written as

$$\beta_0 \cdot n + \beta_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \quad \text{and} \quad \beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i. \tag{1}$$

3.2.3 Reframing with Linear Algebra

Consider reframing the problem using linear algebra. We have known values on a domain, $\{x_i : 1 \leq i \leq n\}$, and a function $P(x_i)$. This function is a linear transformation to obtain the corresponding outputs, $\{y_i : 1 \leq i \leq n\}$. We are approximating $f(x)$ with $P(x)$. Using linear least squares, the 1st degree case of discrete least squares minimization, with weights $\{\beta_0, \beta_1\}$. We can represent this system of equations $y_i = \beta_0 + \beta_1 x_i$ with matrices similarly to the elementary linear algebra case, $\mathbf{A}\vec{x} = \vec{b}$, where $\mathbf{A}$ is a $n \times 2$ (we will see in the general case it is $n \times m$) matrix with a column vector of 1's and a column vector of x_i's, $\vec{x}$ is the 2×1 vector of unknown weights $\{\beta_0, \beta_1\}$, and $\vec{b}$ is the $n \times 1$ vector of y_i's. It can be written as such:

$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}, \; x_i \neq x_j \; \forall \; i \neq j.$$

This is an overdetermined system, in so far as there are more rows than columns in $\mathbf{A}$. each x_i is distinct, and meaning that each row is linearly independent. Usually, the solution to $\mathbf{A}\vec{x} = \vec{b}$ is $\vec{x} = \mathbf{A}^{-1}\vec{b}$, but because $\mathbf{A}$ in this case is not symmetric, it is singular. So, we solve the system by multiplying both sides by $\mathbf{A}^T$. This yields the following:

$$\mathbf{A}^T\mathbf{A}\vec{x} = \mathbf{A}^T\vec{b}.$$

with $\mathbf{A}^T\mathbf{A}$ being a symmetric matrix. For the sake of future simplicity, we will change notation. We will now be representing $\mathbf{A}$ with $\mathbf{X}$, $\vec{x}$ with $\vec{\beta}$, and $\vec{b}$ with $\vec{y}$ respectively. In our problem, our unknowns are the β_i's, and we are given $\{x_i, y_i : 1 \leq i \leq n\}$. Hence, we actually have $\mathbf{X}^T\mathbf{X}\vec{\beta} = \mathbf{X}^T\vec{y}$ with the new notation. These are also the **normal equations**. In the 1^{st} degree case for linear least squares, we have it that

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_1 & x_2 & x_3 & \cdots & x_n \end{bmatrix} \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_1 & x_2 & x_3 & \cdots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix},$$

simplifies to

$$\begin{bmatrix} n & \sum\limits_{i=1}^{n} x_i \\ \sum\limits_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \sum\limits_{i=1}^{n} y_i \\ \sum\limits_{i=1}^{n} y_i x_i \end{bmatrix} \iff \begin{array}{c} \beta_0 \cdot n + \beta_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \\ \beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} y_i x_i. \end{array}$$

Hence, $\mathbf{X}^T\mathbf{X}\vec{\beta} = \mathbf{X}^T\vec{y}$ is in fact the normal equations, for the $k = 1$ degree discrete least squares approximation case, as the results are comparable to the set of equations in 1.

3.3 Polynomial Least Squares Approximation

3.3.1 Error Minimization With the ℓ_2 Norm

Much like Linear Least Squares, Polynomial Least Squares considers the following equation:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_k x^k.$$

Finding the optimal weights $\vec{\beta} = (\beta_0, \beta_1, \beta_2, ..., \beta_k)^T$ can be done similarly to Linear Least Squares, by minimizing the Euclidean distance between y_i and $P(x_i)$, except now the degree of $P(x)$ is now $k > 1$. Now, we have generally,

$$E_2(\beta_0, \beta_1, \beta_2, ..., \beta_k) = \sum_{i=1}^{n} [y_i - (\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + ... + \beta_k x_i^k)]^2,$$

alternatively written as,

$$E_2 = \sum_{i=1}^{n} [y_i - P_n(x_i)]^2 = \sum_{i=1}^{n} y_i^2 - 2\sum_{i=1}^{n} P_n(x_i)y_i + \sum_{i=1}^{n} (P_n(x_i))^2.$$

Then, we can expand $P_n(x)$:

$$E_2 = \sum_{i=1}^{n} y_i^2 - 2\sum_{j=0}^{k} \beta_j \left(\sum_{i=1}^{n} y_i x_i^k \right) + \sum_{j=0}^{k} \sum_{\ell=0}^{k} \beta_j \beta_\ell \left(\sum_{i=1}^{n} x_i^{j+\ell} \right).$$

Now we can minimize this error by finding the partial derivatives with respect to β_j and solving the system of equations that satisfies:

$$0 = \frac{\partial E_2}{\partial \beta_j} = -2\sum_{i=1}^{n} y_i x_i^k + 2\sum_{\ell=0}^{k} \beta_\ell \sum_{i=1}^{n} x_i^{j+\ell},$$

which implies

$$\sum_{\ell=0}^{k} \beta_\ell \sum_{i=1}^{n} x_i^{j+\ell} = \sum_{i=1}^{n} y_i x_i^j$$

for each $j = 0, 1, ..., k$. Now we have $k+1$ **normal equations** for $k+1$ unknowns, β_ℓ.

3.3.2 Reframing with Linear Algebra

Similarly to the degree $k = 1$ case, we can describe the normal equations using matrices, following the same formula, $\mathbf{X}^T\mathbf{X}\vec{\beta} = \mathbf{X}^T\vec{y}$. Expanded, it would look like,

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_1 & x_2 & x_3 & \cdots & x_n \\ x_1^2 & x_2^2 & x_3^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^k & x_2^k & x_3^k & \cdots & x_n^k \end{bmatrix} \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^k \\ 1 & x_2 & x_2^2 & \cdots & x_2^k \\ 1 & x_3 & x_3^2 & \cdots & x_3^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^k \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_1 & x_2 & x_3 & \cdots & x_n \\ x_1^2 & x_2^2 & x_3^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^k & x_2^k & x_3^k & \cdots & x_n^k \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}$$

and would equate to,

$$\begin{bmatrix} n & \sum\limits_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \cdots & \sum\limits_{i=1}^{n} x_i^k \\ \sum\limits_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 & \cdots & \sum_{i=1}^{n} x_i^{k+1} \\ \sum\limits_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^4 & \cdots & \sum_{i=1}^{n} x_i^{k+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum\limits_{i=1}^{n} x_i^k & \sum_{i=1}^{n} x_i^{k+1} & \sum_{i=1}^{n} x_i^{k+2} & \cdots & \sum_{i=1}^{n} x_i^{2k} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} = \begin{bmatrix} \sum\limits_{i=1}^{n} y_i \\ \sum\limits_{i=1}^{n} y_i x_i \\ \sum\limits_{i=1}^{n} y_i x_i^2 \\ \vdots \\ \sum\limits_{i=1}^{n} y_i x_i^k \end{bmatrix}. \tag{2}$$

Where $\mathbf{S} = \mathbf{X}^T\mathbf{X}$ is clearly symmetric.

4 Uniqueness of the Solution to the Normal Equations

How can one know that the solution is to the normal equations is unique? Does there exist multiple, or even infinitely many solutions? Here we will show that there exists a unique solution to the normal equations given a select set of conditions. Before we begin the proof let us introduce one theorem and a couple definitions.

Theorem 3 (Symmetric Eigenvalue Decomposition)**.** *We can decompose any symmetric matrix $\boldsymbol{S}$ in $\mathbf{R}^n$ with the symmetric eigenvalue decomposition (SED):*

$$\boldsymbol{S} = \sum_{i=1}^{n} \lambda_i u_i u_j = \boldsymbol{U}^T \Lambda \boldsymbol{U},\ \Lambda = \boldsymbol{diag}(\lambda_1, ..., \lambda_n)$$

where $\boldsymbol{U} \coloneqq [\vec{u_1}, ..., \vec{u_n}]$ is orthogonal (i.e. $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_n$) and contains the eigenvectors of $\boldsymbol{S}$, while the diagonal matrix Λ contains the eigenvalues of $\boldsymbol{S}$. [2]

Definition 1 (Linear Independence of Column Vectors)**.** *A set of column vectors $\boldsymbol{X}$ are said to be linearly independent if the only solution to the equation*

$$c_1\vec{x_1} + c_2\vec{x_2} + ... + c_m\vec{x_m} = \vec{0}$$

is the solution $c_1 = c_2 = ... = c_m = 0$. [4]

Definition 2 (Positive Definite)**.** *A matrix is positive definite if is symmetric, and all its pivots are positive. [5]*

Note that if there exists a set of column vectors $\mathbf{X}$ as a matrix, each with n elements, where $n > m$, such as in the usual case of polynomial regression,

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^m \\ 1 & x_2 & x_2^2 & \cdots & x_2^m \\ 1 & x_3 & x_3^2 & \cdots & x_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^m \end{bmatrix},$$

then each of the column vectors are linearly independent. Consider the equation in Definition 1,

$$c_0 \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + c_1 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} + c_2 \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_3^2 \\ \vdots \\ x_n^2 \end{bmatrix} + ... + c_m \begin{bmatrix} x_1^m \\ x_2^m \\ x_3^m \\ \vdots \\ x_n^m \end{bmatrix} = \vec{0}$$

This is a system of equations with n m^{th} degree polynomials, implying there are n roots the polynomial $c_0 + c_1x_i + c_2x_i^2 + ... + c_mx_i^m = 0$ such that $i = 1, ..., n$. If $n > m$, then there would be n roots for an m^{th} degree polynomial, which is a contradiction. So the way for this equality to hold, given the x_i's are distinct would be if $c_0 = c_1 = c_2 = ... = c_m = \vec{0}$. [3]

Theorem 4 (Invertibility of $\mathbf{X}^T\mathbf{X}$)**.** *Let $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ be an $n \times m$ data matrix with columns containing $\{1, x_i, x_i^2, ..., x_i^k\}$ for all $i = 1, 2, ..., n$, with the columns being linearly independent of each other and $n > m$. Then $\boldsymbol{X}^T\boldsymbol{X}$ is symmetric positive definite, and hence is invertible.*

Proof. We know from equation 2 that $\mathbf{S} = \mathbf{X}^T\mathbf{X}$ is a symmetric matrix such that the x_i's composing it are unique. We also know for $\mathbf{X}$ is an $n \times m$ matrix such that $n > m$. From equation 2, we know that $\mathbf{S}$ is symmetric because

$$[s_{j\ell}] = \sum_{i=1}^{k} x_i^{j+\ell}.$$

We also know that all of the diagonal entries are strictly positive, as

$$[s_{jj}] = \sum_{i=1}^{n} x_i^{2j}.$$

So $\mathbf{S}$ is symmetric and positive definite. Also consider that there exists an eigenpair $(\lambda, \vec{x})$ such that $\mathbf{S}\vec{x} = \lambda\vec{x}$, where $\vec{x} \neq \vec{0}$. Without loss of generality, we assume $\vec{x}^T\vec{x} = 1$. Then

$$0 < \vec{x}^T\mathbf{S}\vec{x} = \vec{x}^T(\lambda\vec{x}) = \lambda\vec{x}^T\vec{x} = \lambda,$$

implying all eigenvalues are positive. Conversely, we know from Definition 2 that $\mathbf{S}$ is positive definite. From Theorem 3 we know that there exists an orthogonal matrix $\mathbf{U}$ such that $\mathbf{S} = \mathbf{U}^T\Lambda\mathbf{U}$ where $\Lambda = \mathbf{diag}(\lambda_1, ..., \lambda_n)$. If $\vec{x} \neq \vec{0}$, then $y := \mathbf{U}\vec{x} \neq \vec{0}$. Then

$$\vec{x}^T\mathbf{S}\vec{x} = \vec{x}^T(\mathbf{U}^T\Lambda\mathbf{U})\vec{x} = (\mathbf{U}\vec{x})^T\Lambda\mathbf{U}\vec{x} = y^Ty = \sum_{i=1}^{n}\lambda_i y_i^2 > 0,$$

because all eigenvalues are positive. So $\mathbf{S}$ is symmetric positive definite if and only if all its eigenvalues are positive. Because $\det(\mathbf{S}) = \prod_{i=1}^{n}\lambda_i$, then this would imply that

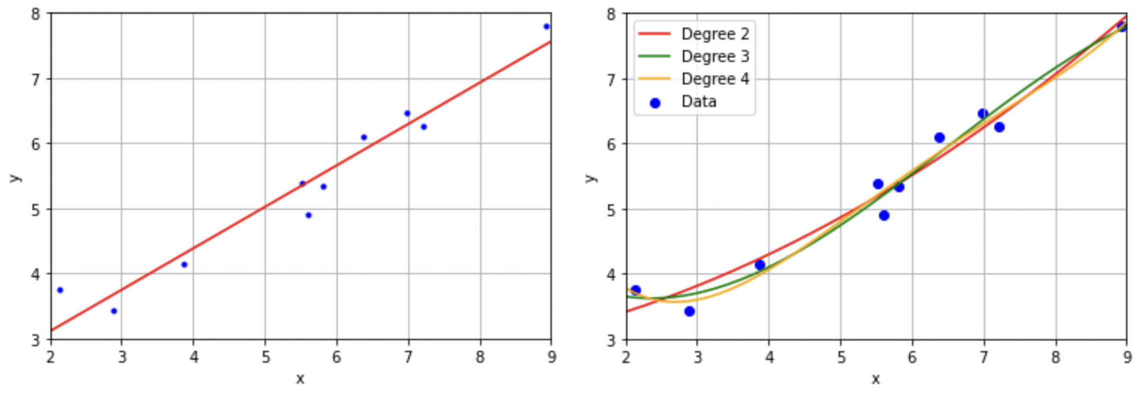
$$\det(A) \geq 0 \iff \mathbf{S}^{-1} \text{ exists.}$$

□

Now, by Theorem 4, we see that if $\vec{c} := \mathbf{X}^T y$, then the unique solution to the system $\mathbf{S}\vec{\beta} = \vec{c}$ exists, and is

$$\vec{\beta} = \mathbf{S}^{-1}\vec{c} \iff \vec{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\vec{y}.$$

Hence, the solution to the normal equations is unique. We also already know that it is the minimizer of $\sum_{i=1}^{n}[y_i - P_n(x_i)]^2$.



(a)

(b)

Figure 3: (a) The Linear Least Squares approximation for random data, the $n = 1$ degree case of Discrete Least Squares approximation. (b) Select Polynomial Least Squares approximations for the same random data.

5 Contrasting Methods

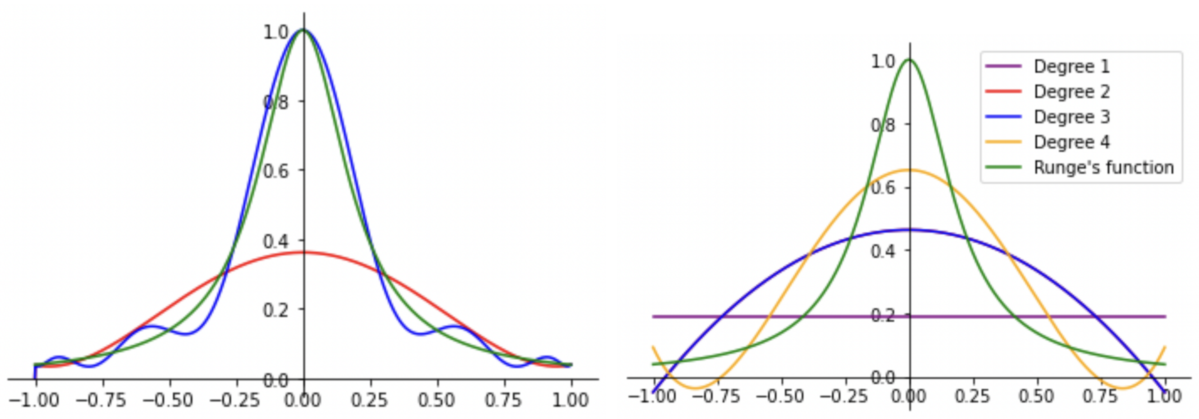
5.1 Random Data

We now return to the case with randomly generated data, with a sufficient number $n + 1$ points. Recall in Figure 2(b) how Lagrange interpolation was able to agree with each data point at node x_i, $i = 0, 1, 2, ..., n, n + 1$, but seemed to produce highly oscillatory results unless the nodes were close enough.

Consider Figure 3. Each Discrete Least Squares approximation behaves much better over the domain, despite not agreeing with each data point. Generally speaking, it is better to use Discrete Least Squares approximations for randomly generated data.

5.2 Selected Data

Suppose we can choose points $n + 1$ points, x_i, $i = 0, 1, 2, ..., n, n + 1$, from which we gather observations y_i, $i = 0, 1, 2, ..., n, n + 1$. Figure 3 1 shows what happens when $n + 1$ values are selected to create an n^{th} degree interpolation polynomial, $P_n(x)$. Figure 4 showcases a scenario where interpolation might possess an advantage for finding the true $f(x)$, which is known in this example but usually is unknown in practice. The visualizations contrasting interpolation and discrete least squares in Figure 4 do not necessarily do justice to polynomial least squares of higher degrees, but the figure serves to show where interpolation might be a good option for usage.



(a)

(b)

Figure 4: (a) This is the same plot as Figure 1(b), showing Runge's function, as well as $P_5(x)$ and $P_{12}(x)$, the 5th and 12th degree Lagrange interpolation polynomials for Runge's function using the *Clenshaw-Curtis* points. (b) The n^{th} degree Discrete Least Squares approximation for Runge's function using the same points. *Note that the degree 2 polynomial is covered by the degree 3 one.*

6 Conclusion

6.1 From Concept to Practice

What are some practical applications whereby one method might be better than another? Interpolation might shine brightest when the ability to choose the independent variable is possible. Examples of this include:

- Time series observations when observations can be measured at chosen times.
- Experiments where sensors can be placed a strategic locations.
- Approximating a complicated, known function, with a simpler one on a given domain.

Linear Least Squares would be very optimal when:

- There exists a strong correlation between the dependent and independent variable,
- it would be useful to know how to describe the relationship between the two with a simple weight, β_1, or
- there are sufficiently high data points, such that Lagrange interpolation would produce unsatisfactory results.

Lastly, Polynomial Least Squares would be very useful when:

- data visualization reveals the relationship between the dependent and independent variables are quadratic, cubic, or quartic, etc.,

- data analysis reveals that there are no outliers within the data, as this has a greater effect on model fitting than Linear Least Squares, or

- in real world practices such as studying isotopes of sediments, and the rise of a disease in a population.

6.2 Concluding Thoughts

Ultimately, both non-piecewise Interpolation and Discrete Least Squares methods are viable options for finding the $P(x)$ to approximate the true $f(x)$, whose existence is proven by Theorem 1. As demonstrated, there exists a unique, optimal solution for Discrete Least Squares equations; however, these models do not usually agree with every data point unlike Lagrange interpolation.

References

[1] Annette Burden, Richard Burden, and J. Faires. *Numerical Analysis, 10th ed.* 01 2016.

[2] G.C. Calafiore and L. El Ghaoui. *Optimization Models.* Control systems and optimization series. Cambridge University Press, October 2014.

[3] Peter Jantsch. Office hours, December 2022.

[4] Stephen Lovett. Lecture notes in linear algebra: Linear independence, basis and dimension, January 2021.

[5] Allan Pinkus. *Eigenvalues and eigenvectors*, page 127–153. Cambridge Tracts in Mathematics. Cambridge University Press, 2009.