Best approximations and least-squares best-fitting polynomials

February 16, 2025

In engineering applications, we often encounter situations where we need to approximate data using a set of basis functions or measurements. Whether in signal processing, control systems, circuit analysis, or machine learning, engineers frequently model complex systems using a set of known vectors and seek the best approximation to observed data.

For instance, in electrical and computer engineering, least squares arises in:

1. Signal reconstruction, where we approximate a signal using a subset of basis functions.
2. System identification, where we fit a mathematical model to input-output data.
3. Error correction, where we find the best estimate of transmitted data in noisy communication channels.
4. Linear regression, which is used in machine learning and data-driven modeling of physical systems.

When the given vectors do not span the space of possible solutions, often no exact solution exists. Instead, we seek the best approximation—the linear combination that minimizes the error. The least squares method provides a systematic approach to finding this best approximation, making it a fundamental tool in many engineering disciplines.

In this section, we introduce least squares by examining a simple example where an exact solution does not exist and show how to determine the closest possible approximation.

Given a collection of n m-dimensional vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbf{R}^m$, we seek to approximate a given m-dimensional target vector $\mathbf{y} \in \mathbf{R}^m$ as a linear combination of the given vectors. That is, we seek to find scalars $a_1, \ldots, a_n$ such that

$$a_1\mathbf{v}_1 + \cdots + a_n\mathbf{v}_n$$

comes as close to, or best approximates, $\mathbf{y}$ as much as possible. By "best approximates," we will mean the linear combination that is to $\mathbf{y}$ with respect to the 2-norm (also known as the Euclidean norm).

Example

For example, we may ask what linear combination of

$$\begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} \text{ and } \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

This problem can be written as a linear system

$$\begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix}.$$

To determine whether an exact solution exists, we form the augmented matrix and perform Gaussian elimination (using partial pivoting):

$$\left(\begin{array}{cc|c} 2 & 1 & 1 \\ 1 & 1 & -1 \\ 0 & 1 & 3 \end{array}\right) \sim \left(\begin{array}{cc|c} 2 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & -3 \end{array}\right),$$

and because the rank of the coefficient matrix (the first two columns) is two but the rank of the augmented matrix is three, the system is inconsistent, so no exact solution exists.

One may never-the-less still ask: What linear combination of these two vectors is closest to the target? Recall that the span of a collection of vectors is by definition all linear combinations, so we want to find the vector in span $\left\{ \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}$ that is closest to $\mathbf{y}$.

Remember that $0.5 \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} + 0.2 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1.2 \\ 0.7 \\ 0.2 \end{pmatrix}$ is equivalent to multiplying $\begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.2 \end{pmatrix} = \begin{pmatrix} 1.2 \\ 0.7 \\ 0.2 \end{pmatrix}$. Thus, considering all linear combinations of these two vectors $a_1 \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} + a_2 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ is equivalent to considering all products $\begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$ for all vectors $\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \in \mathbf{R}^2$.

Recall that the *range* of this matrix $\begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}$ is all products of this matrix by a vector in the domain $\mathbf{R}^2$, so we are equivalently asking: What vector in the range of this matrix $\begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}$ is closest to the target vector $\begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix}$.

To define *closest*, we mean what vector $\mathbf{a}$ minimizes the distance between $\begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \mathbf{a}$ and $\begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix}$, or, in other words, what vector $\mathbf{a}$ minimizes $\left\| \begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \mathbf{a} - \begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix} \right\|_2$?

If we define the matrix

$$V = (\mathbf{v}_1 \cdots \mathbf{v}_n),$$

asking which linear combination of the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ is closest to an m-dimensional target vector $\mathbf{y}$ is equivalent to asking: What vector $\mathbf{a} \in \mathbf{R}^n$ brings $V\mathbf{a}$ closest to $\mathbf{y}$, or what $\mathbf{a}$ minimizes $\|V\mathbf{a} - \mathbf{y}\|_2$?

Remark

The range of a $3 \times n$ matrix A, which consists of all possible linear combinations of its columns, can take on different geometric forms depending on its rank:

1. Zero-dimensional ($\text{rank}(A) = 0$): If all entries of A are zero, its range consists only of the zero vector $\mathbf{0}_3$.
2. One-dimensional ($\text{rank}(A) = 1$): If all columns of A are scalar multiples of each other, its range is a line through the origin.
3. Two-dimensional ($\text{rank}(A) = 2$): If two of the columns of A form a basis for the span (that is, a basis of the range), then the range is a plane through the origin. scalar multiples of each other, its range is a line through the origin.
4. Three-dimensional ($\text{rank}(A) = 3$): If there are at least three linearly independent columns, the range is all of $\mathbf{R}^3$.

Now, identify any line or plane in the room you are in and let that represent the range of a $3 \times n$ matrix. Next, choose any point off that line or plane. The closest point within the line or plane to your chosen is the one where the difference vector between the chosen point and the range is perpendicular to every possible direction within the range. In other words, the shortest distance is achieved when the error vector is orthogonal to the subspace.
This insight forms the foundation of least squares approximation: when an exact solution does not exist, the best approximate solution is the projection of the given point onto the subspace defined by the range of the matrix.

To determine whether two vectors are perpendicular (or orthogonal), we check whether their dot product is zero. This property is key to finding the best approximation in least squares problems. We seek a vector $\mathbf{a}$ such that the difference between $V\mathbf{a}$ (the closest point in the range of V) and the target vector $\mathbf{y}$ is perpendicular to the entire range of V. That is, the error vector

$$V\mathbf{a} - \mathbf{y}$$

must be orthogonal to every vector in the range of V.

Since every vector in the range of V can be expressed as $V\mathbf{u}$ for some $\mathbf{u} \in \mathbf{R}^n$, this condition translates into the following equation:

$$(V\mathbf{a} - \mathbf{y}) \cdot (V\mathbf{u}) = 0$$

for every vector $\mathbf{u}$ in the domain $\mathbf{R}^n$.

This expresses the fundamental ideal behind least squares: the error vector must be orthogonal to the subspace spanned by the columns of V, ensuring that $V\mathbf{a}$ is the closest possible approximation to $\mathbf{y}$.

Remark

This, however, does not yet help us to find $\mathbf{a}$. However, you may have been taught the "transpose" of a matrix. The purpose of the transpose is that if a matrix $A : \mathscr{U} \to \mathscr{V}$, then $A^\top : \mathscr{V} \to \mathscr{U}$, and we have the property that $(A\mathbf{u}) \cdot \mathbf{v} = \mathbf{u} \cdot (A^\top \mathbf{v})$ for all vectors $\mathbf{u} \in \mathscr{U}$ and all vectors $\mathbf{v} \in \mathscr{V}$. Similarly, if $B : \mathscr{V} \to \mathscr{U}$, then $B^\top : \mathscr{U} \to \mathscr{V}$, and we have the property that $\mathbf{u} \cdot (B\mathbf{v}) = (B^\top \mathbf{u}) \cdot \mathbf{v}$, also for all vectors $\mathbf{u} \in \mathscr{U}$ and all vectors $\mathbf{v} \in \mathscr{V}$.

We note that $(V\mathbf{a} - \mathbf{y}) \cdot (V\mathbf{u})$ is the dot product of a vector on the left and $V\mathbf{u}$ on the right, so we can move the matrix multiplication to the left-hand side by using the transpose instead:

$$(V\mathbf{a} - \mathbf{y}) \cdot (V\mathbf{u}) = (V^\top (V\mathbf{a} - \mathbf{y})) \cdot \mathbf{u} = 0,$$

and this must be true for all vectors $\mathbf{u} \in \mathbf{R}^n$. The only vector that is perpendicular to all vectors in $\mathbf{u} \in \mathbf{R}^n$ is the zero vector $\mathbf{0}_n$, and thus, the left-hand side of the dot product must be this zero vector. Therefore,

$$V^\top (V\mathbf{a} - \mathbf{y}) = \mathbf{0}_n.$$

Recall that one of the properties of a matrix is that it is linear, $A(\mathbf{u}_1 + \mathbf{u}_2) = A\mathbf{u}_1 + A\mathbf{u}_2$, so in this case, we have:

$$V^\top V\mathbf{a} - V^\top \mathbf{y} = \mathbf{0}_n.$$

Moving the one term to the right-hand side, we now have:

$$V^\top V\mathbf{a} = V^\top \mathbf{y}.$$

Notice that if V is $m \times n$ and that $\mathbf{y}$ is m-dimensional. Thus, $V^\top V$ is now $n \times n$ and $V^\top \mathbf{y}$ is n-dimensional. Thus, the matrix is square, so you may think that there are either no solutions, exactly one solution, or infinitiely many solutions; however, in this case, because both the left- and right-hand sides are restricted to the range of $V^\top$, it follows that there is either one solution or infinitely many solutions: there can never be no solutions. In other words, there is always one vector in the range that is "closest" to the given target vector $\mathbf{y}$.

Example

Returning to our example, we asked what linear combination of $\begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ best approximates $\mathbf{y} = \begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix}$?

We now define $V = \begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}$ and solve the system of linear equations defined by $V^\top V\mathbf{a} = V^\top \mathbf{y}$, or the system described by the augmented matrix $(V^\top V | V^\top \mathbf{y})$. Now, $V^\top V = \begin{pmatrix} 5 & 3 \\ 3 & 3 \end{pmatrix}$ and $V^\top \mathbf{y} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$, and thus, we solve

$$\left(\begin{array}{cc|c} 5 & 3 & 1 \\ 3 & 3 & 3 \end{array}\right) \sim \left(\begin{array}{cc|c} 5 & 3 & 1 \\ 0 & 1.2 & 2.4 \end{array}\right),$$

which has the unique solution $\mathbf{a} = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$.

Consequently, the best approximation the target vector is

$$-\begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} + 2\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

and

$$\left\| \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} - \begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix} \right\| = \sqrt{6} \approx 2.4495.$$

To observe that this is the best approximation, note that

$$\left\| V \begin{pmatrix} -1.1 \\ 2 \end{pmatrix} - \mathbf{y} \right\|_2 = \left\| V \begin{pmatrix} -0.9 \\ 2 \end{pmatrix} - \mathbf{y} \right\|_2 \approx 2.4597$$

and

$$\left\| V \begin{pmatrix} -1 \\ 2.1 \end{pmatrix} - \mathbf{y} \right\|_2 = \left\| V \begin{pmatrix} -1 \\ 1.9 \end{pmatrix} - \mathbf{y} \right\|_2 \approx 2.4556,$$

all four of which are further away from the target $\mathbf{y}$.

Important: These numbers are "nice" in order to demonstrate that this works. In reality, the coefficients of the linear combination will not necessarily be so "nice."

Example

As a second example, find the linear combination of

$$\begin{pmatrix} 1 \\ 1 \\ 0 \\ -2 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \\ 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ -1 \\ -3 \\ 2 \end{pmatrix}$$

that best approximates the target vector

$$\mathbf{y} = \begin{pmatrix} -4 \\ -1 \\ 6 \\ 3 \end{pmatrix}.$$

Let

$$V = \begin{pmatrix} 1 & -1 & 2 \\ 1 & 1 & -1 \\ 0 & 2 & -3 \\ -2 & 1 & 2 \end{pmatrix}$$

and then calculate $V^\top V\mathbf{a} = V^\top \mathbf{y}$ or

$$\left(\begin{array}{rrr|r} 6 & -2 & -3 & -11 \\ -2 & 7 & -7 & 18 \\ -3 & -7 & 18 & -19 \end{array}\right) \sim \left(\begin{array}{rrr|r} 6 & -2 & -3 & -11 \\ 0 & -8 & 16.5 & -24.5 \\ 0 & 0 & 5.0625 & -5.0625 \end{array}\right),$$

so $\mathbf{a} = \begin{pmatrix} -2 \\ 1 \\ -1 \end{pmatrix}$, and so the best linear combination of those three vectors that equals the given target vector is

$$-2\begin{pmatrix} 1 \\ 1 \\ 0 \\ -2 \end{pmatrix} + \begin{pmatrix} -1 \\ 1 \\ 2 \\ 1 \end{pmatrix} - \begin{pmatrix} 2 \\ -1 \\ -3 \\ 2 \end{pmatrix} = \begin{pmatrix} -5 \\ 0 \\ 5 \\ 3 \end{pmatrix} \approx \mathbf{y} = \begin{pmatrix} -4 \\ -1 \\ 6 \\ 3 \end{pmatrix}.$$

You will see that this is indeed close to the target vector on the right-hand side. Once again, this example was specifically chosen to have the numbers work out "nicely" so that you can observe that this does appear to be the best approximation.

Using least-squares to find best-fitting linear and quadratic polynomials

Suppose we have m points $(x_1, y_1), \ldots, (x_m, y_m)$, and we'd like to find a linear polynomial that passes as closely to these points. Thus, what we are asking is can we find coefficients a_1 and a_0 such that

$$\begin{gathered} a_1 x_1 + a_0 \approx y_1 \\ a_1 x_2 + a_0 \approx y_2 \\ a_1 x_3 + a_0 \approx y_3 \\ \vdots \\ a_1 x_m + a_0 \approx y_m \end{gathered}$$

If we rewrite this as vectors, we arrive at

$$\begin{pmatrix} a_1 x_1 + a_0 \\ a_1 x_2 + a_0 \\ a_1 x_3 + a_0 \\ \vdots \\ a_1 x_m + a_0 \end{pmatrix} \approx \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{pmatrix}.$$

Notice, we can split the left-hand vector into a linear combination: If we rewrite this as vectors, we arrive at

$$a_1 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_m \end{pmatrix} + a_0 \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \approx \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{pmatrix}.$$

You will note, therefore, that we are asking what is the best approximation of the target vector $\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$ as a linear combination of the vectors $\begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$ and $\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$.

Similarly, if we are attempting to find the best quadratic polynomial that passes through these m points, we are asking what linear combination of three vectors best approximates the target vector $\mathbf{y}$:

$$\begin{pmatrix} a_2 x_1^2 + a_1 x_1 + a_0 \\ a_2 x_2^2 + a_1 x_2 + a_0 \\ a_2 x_3^2 + a_1 x_3 + a_0 \\ \vdots \\ a_2 x_m^2 + a_1 x_m + a_0 \end{pmatrix} = a_2 \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_3^2 \\ \vdots \\ x_m^2 \end{pmatrix} + a_1 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_m \end{pmatrix} + a_0 \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \approx \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{pmatrix}.$$

You will note, again, that we are asking what is the best approximation of the target vector $\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$ as a linear combination of the three vectors $\begin{pmatrix} x_1^2 \\ \vdots \\ x_m^2 \end{pmatrix}$, $\begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$ and $\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$.

However, we already know how to find a best approximation, and so we define the two "tall" Vandermonde matrices,

$$V_1 = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ \vdots & \\ x_m & 1 \end{pmatrix} \quad \text{and} \quad V_2 = \begin{pmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \\ \vdots & & \\ x_m^2 & x_m & 1 \end{pmatrix},$$

where the index is the degree of the polynomial. Finding the least-squares best-fitting line is equivalent to solving $V_1^\top V_1 \begin{pmatrix} a_1 \\ a_0 \end{pmatrix} = V_1^\top \mathbf{y}$, and finding the best-fitting quadratic polynomial is equivalent to solving $V_2^\top V_2 \begin{pmatrix} a_2 \\ a_1 \\ a_0 \end{pmatrix} = V_2^\top \mathbf{y}$.

You will note that the coefficients correspond to the order of the columns of V_1 and V_2.

These products can be easily calculated:

$$V_1^\top V_1 = \begin{pmatrix} \sum_{k=1}^m x_k^2 & \sum_{k=1}^m x_k \\ \sum_{k=1}^m x_k & m \end{pmatrix}, \quad V_1^\top \mathbf{y} = \begin{pmatrix} \sum_{k=1}^m x_k y_k \\ \sum_{k=1}^m y_k \end{pmatrix}$$

and

$$V_2^\top V_2 = \begin{pmatrix} \sum_{k=1}^m x_k^4 & \sum_{k=1}^m x_k^3 & \sum_{k=1}^m x_k^2 \\ \sum_{k=1}^m x_k^3 & \sum_{k=1}^m x_k^2 & \sum_{k=1}^m x_k \\ \sum_{k=1}^m x_k^2 & \sum_{k=1}^m x_k & m \end{pmatrix}, \quad V_2^\top \mathbf{y} = \begin{pmatrix} \sum_{k=1}^m x_k^2 y_k \\ \sum_{k=1}^m x_k y_k \\ \sum_{k=1}^m y_k \end{pmatrix}.$$

Remark

These matrices have a very specific *structural property* where each the entries are constant along the anti-diagonals (from top-right to bottom-left); for example,

$$\begin{pmatrix} a & b & c & d \\ b & c & d & e \\ c & d & e & f \\ d & e & f & g \end{pmatrix}.$$

These matrices are called “Hankel” matrices.

Example

Find the best-fitting least-squares linear and quadratic polynomial that passes through the points

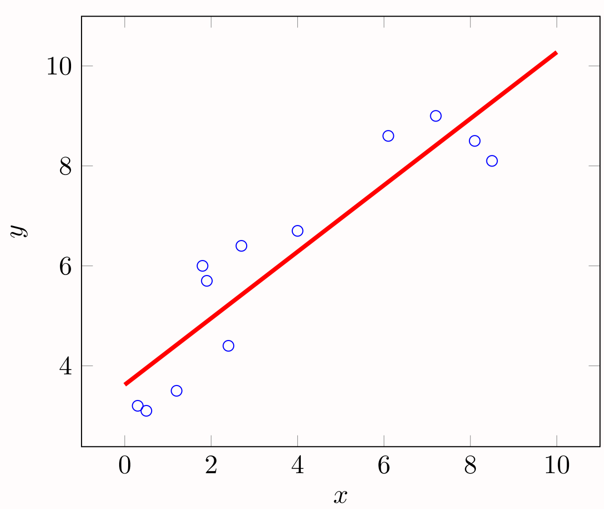
$$(0.3, 3.2), (0.5, 3.1), (1.2, 3.5), (1.8, 6.0), (1.9, 5.7), (2.4, 4.4),$$
$$(2.7, 6.4), (4.0, 6.7), (6.1, 8.6), (7.2, 9.0), (8.1, 8.5), (8.5, 8.1).$$

First, we define the matrices

$$V_1 = \begin{pmatrix} 0.3 & 1 \\ 0.5 & 1 \\ 1.2 & 1 \\ 1.8 & 1 \\ 1.9 & 1 \\ 2.4 & 1 \\ 2.7 & 1 \\ 4.0 & 1 \\ 6.1 & 1 \\ 7.2 & 1 \\ 8.1 & 1 \\ 8.5 & 1 \end{pmatrix}, \quad V_2 = \begin{pmatrix} 0.09 & 0.3 & 1 \\ 0.25 & 0.5 & 1 \\ 1.44 & 1.2 & 1 \\ 3.24 & 1.8 & 1 \\ 3.61 & 1.9 & 1 \\ 5.76 & 2.4 & 1 \\ 7.29 & 2.7 & 1 \\ 16.00 & 4.0 & 1 \\ 37.21 & 6.1 & 1 \\ 51.84 & 7.2 & 1 \\ 65.61 & 8.1 & 1 \\ 72.25 & 8.5 & 1 \end{pmatrix}.$$

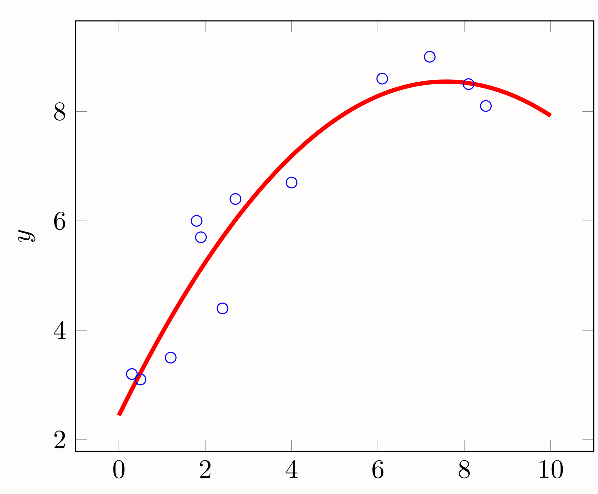
We then calculate $V_1^\top V_1 = \begin{pmatrix} 264.59 & 44.7 \\ 44.7 & 12 \end{pmatrix}$ and $V_1 \top \mathbf{y} = \begin{pmatrix} 33.794 \\ 7.32 \end{pmatrix}$, which if we solve $V_1^\top V \begin{pmatrix} a_1 \\ a_0 \end{pmatrix} = V_1^\top \mathbf{y}$ to get that $\begin{pmatrix} a_1 \\ a_0 \end{pmatrix} \approx \begin{pmatrix} 0.665460199321999 \\ 3.621160757525552 \end{pmatrix}$, and thus, the best-fitting least-squares line passing through this data is shown here:

The least-squares best-fitting linear polynomial of this data



Alternatively, to find the least-squares best-fitting quadratic polynomial, we calculate $V_2^\top V_2 = \begin{pmatrix} 13964.6999 & 1857.873 & 264.59 \\ 1857.873 & 264.59 & 44.7 \\ 264.59 & 44.7 & 12 \end{pmatrix}$ and $V_2 \top \mathbf{y} = \begin{pmatrix} 2154.796 \\ 337.94 \\ 73.2 \end{pmatrix}$, which if we solve $V_2^\top V \begin{pmatrix} a_2 \\ a_1 \\ a_0 \end{pmatrix} = V_2^\top \mathbf{y}$ to get that $\begin{pmatrix} a_2 \\ a_1 \\ a_0 \end{pmatrix} \approx \begin{pmatrix} -0.106255401076057 \\ 1.610419356536262 \\ 2.444030944461919 \end{pmatrix}$, and thus, the best-fitting least-squares quadratic polynomial passing through this data is shown here:

The least-squares best-fitting quadratic polynomial of this data



Now, what actually is being minimized? For each y_k, it is being approximated by either $a_1x_k + a_0$ or $a_2x_k^2 + a_1x_k + a_0$, respectively. Thus, as the 2-norm is the square root of the sum of the squares of the entries, we have that we are minimizing:

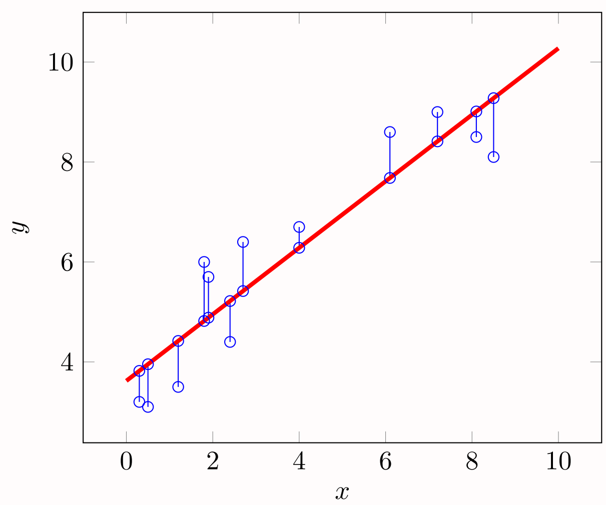
$$\sqrt{\sum_{k=1}^{m}((a_1x_k + a_2) - y_k)^2} \text{ or } \sqrt{\sum_{k=1}^{m}((a_2x_k^2 + a_1x_k + a_2) - y_k)^2},$$

again, respectively.

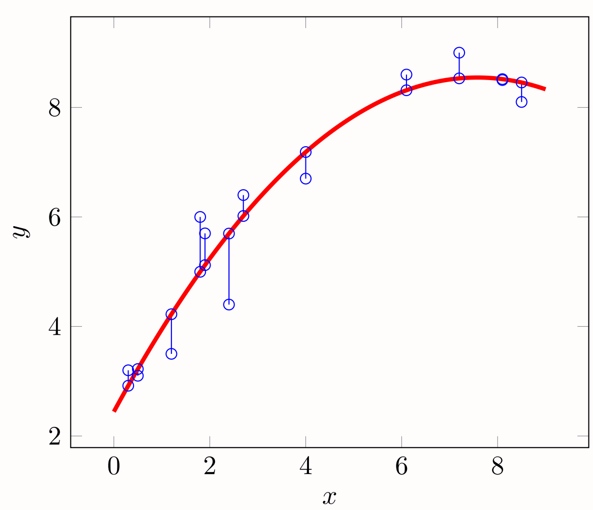
Example

In the previous example, the values being minimized are the sums of the squares of the differences between the actual y_k values and the corresponding point on the least-squares best-fitting polynomial. These errors being squared are highlighted in these two graphs.

The errors of the best-fitting linear polynomial



The errors of the best-fitting quadratic polynomial



Remark

Engineers often avoid using the 2-norm because it measures only the overall deviation of the entire vector, whereas they are typically more interested in the error per entry. Instead, they frequently use the root-mean-squared error (RMSE), which can be viewed as the *root of the average squared error*, in contrast to the 2-norm's *root of the total squared error*. To compute the RMSE, one first sums the squares of the error, but then divides by the number of entries to find the average, and only then takes the square root.

Concretely, if V is the corresponding Vandermonde matrix for a linear polynomial and $\mathbf{a}$ is the vector of coefficients, then

$$\|V\mathbf{a} - \mathbf{y}\|_2 = \sqrt{\sum_{k=1}^{m}((a_1 x_k + a_0) - y_k)^2},$$

while the corresponding RMSE is

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{k=1}^{m}((a_1 x_k + a_0) - y_k)^2}.$$

This relationship can also be expressed succinctly as

$$\text{RMSE} = \sqrt{\frac{1}{n}\|V\mathbf{a} - \mathbf{y}\|_2^2} = \frac{1}{\sqrt{n}}\|V\mathbf{a} - \mathbf{y}\|_2.$$

Remark

Given these points $(x_1, y_1), \ldots, (x_m, y_m)$, in addition to finding the least-squares best-fitting linear and quadratic polynomials, you may ask: What is the best-fitting constant function a_0 passing through these points?

In this case, the tall Vandermonde matrix consists of only a single column: an $m \times 1$ matrix of all ones:

$$V_0 = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}.$$

When you now solve $V_0^\top V_0(a_0) = V_0^\top \mathbf{y}$, this simplifies to

$$(m)(a_0) = \left(\sum_{k=1}^{m} y_k\right),$$

and solving this, we get that

$$a_0 = \frac{1}{m}\sum_{k=0}^{m} y_k,$$

which may actually be what you would expect: the least-squares best-fitting constant function passing through m points is that constant function equal to the average of the y-values.

Note, the equation looks a little awkward, because we have a 1×1 matrix multiplied by a 1-dimensional vector equated to a 1-dimensional vector, but that is the same result you would get if you asked: What is the scalar multiple of a given vector $\mathbf{v}$ that best approximates a given vector $\mathbf{u}$? You also saw this in your first-year linear algebra course: it is, of course, the projection: $\text{proj}_{\mathbf{v}}(\mathbf{u}) = \frac{\mathbf{v}\cdot\mathbf{u}}{\mathbf{v}\cdot\mathbf{v}}\mathbf{v}$, so if we let $\mathbf{v} = \mathbf{1}_m$ (an m-dimensional vector of all ones), then

$$\begin{aligned}\text{proj}_{\mathbf{1}_m}(\mathbf{u}) &= \frac{\mathbf{1}_m \cdot \mathbf{u}}{\mathbf{1}_m \cdot \mathbf{1}_m}\mathbf{1}_m \\ &= \frac{\sum_{k=1}^{m} u_k}{m}\mathbf{1}_m \\ &= \left(\frac{1}{m}\sum_{k=1}^{m} u_k\right)\mathbf{1}_m,\end{aligned}$$

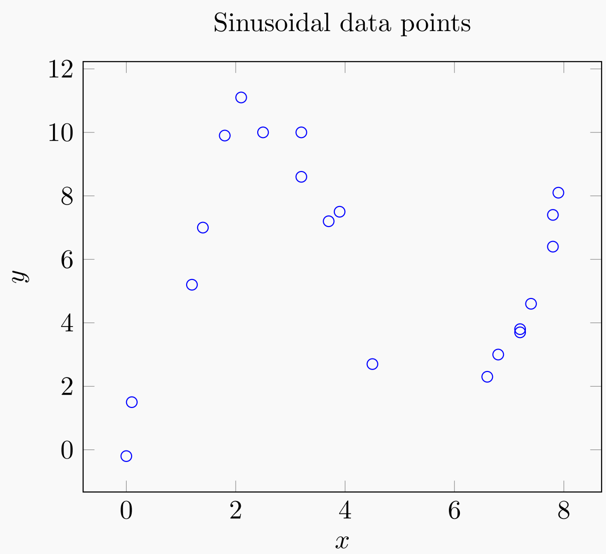
That is, the best approximation of $\mathbf{u}$ by a scalar multiple of $\mathbf{1}_m$ is that constant vector with all entries equal to the average of the entries in $\mathbf{u}$.

Remark

Beyond the scope of this course, this technique is not restricted to simply finding least-squares best-fitting polynomials. You can also find least-squares best-fitting linear combinations of any set of functions. For example, consider the following data:

$$(0.0, -0.2), (0.1, 1.5), (1.2, 5.2), (1.4, 7.0), (1.8, 9.9), (2.1, 11.1), (2.5, 10.0), (3.2, 8.6), (3.2, 10.0), (3.7, 7.2), (3.9, 7.5), (4.5, 2.7), (6.6, 2.3), (6.8, 3.0), (7.2, 3.8), (7.2, 3.7), (7.4, 4.6), (7.8, 6.4), (7.8, 7.4), (7.9, 8.1).$$

These data points are shown here:



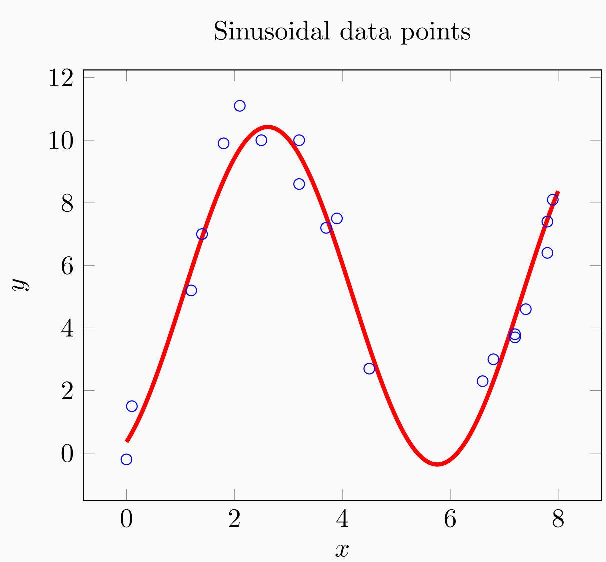
If you were aware that this was coming from a system that was sinusoidal with a period of 2π and offset by a constant, then you may ask: What is the best fitting curve of the form $y = a\sin(x) + b\cos(x) + c$ that fits this data? Like above, we are asking what coefficients approximate the y-values as closely as possible, so $a\sin(x_k) + b\cos(x_k) + c \approx y_k$. We create the corresponding matrix

$$V = \begin{pmatrix} \sin(x_1) & \cos(x_1) & 1 \\ \sin(x_2) & \cos(x_2) & 1 \\ \sin(x_3) & \cos(x_3) & 1 \\ \vdots & \vdots & \vdots \\ \sin(x_m) & \cos(x_m) & 1 \end{pmatrix},$$

and again solve $V^\top V\mathbf{a} = V^\top \mathbf{y}$. Thus, we have

$$V = \begin{pmatrix} 0 & 1 & 1 \\ 0.100 & 0.995 & 1 \\ 0.932 & 0.362 & 1 \\ 0.985 & 0.170 & 1 \\ \vdots & \vdots & \vdots \\ 0.999 & -0.046 & 1 \end{pmatrix}, V^\top V = \begin{pmatrix} 11.018 & 2.886 & 8.429 \\ 2.886 & 8.982 & 0.750 \\ 8.429 & 0.750 & 20 \end{pmatrix}, \text{ and } V^\top \mathbf{y} = \begin{pmatrix} 58.561 \\ -30.441 \\ 119.8 \end{pmatrix}.$$

Solving $V^\top V\mathbf{a} = V^\top \mathbf{y}$, we have $\begin{pmatrix} 2.690 \\ -4.674 \\ 5.031 \end{pmatrix}$, so the least-squares best-fitting linear combination is $2.690\sin(x) - 4.674\cos(x) + 5.031$, shown here:



Formulas for the least-squares best-fitting polynomials

When you are asked to program least-squares best-fitting polynomials in a computer, you are simply given a formula. We will show that those formulas come from solving the systems of two and three linear equations above, starting with

$$\left(\begin{array}{cc|c} \sum_{k=1}^{m} x_k^2 & \sum_{k=1}^{m} x_k & \sum_{k=1}^{m} x_k y_k \\ \sum_{k=1}^{m} x_k & m & \sum_{k=1}^{m} y_k \end{array}\right)$$

This, however, looks too confusing, so we will rewrite this by substituting:

- $S_x = \sum_{k=1}^{m} x_k$, the *sum* of the xs.
- $S_{x^2} = \sum_{k=1}^{m} x_k^2$, the *sum* of the xs squared.
- $S_y = \sum_{k=1}^{m} y_k$, the *sum* of the ys.
- $S_{xy} = \sum_{k=1}^{m} x_k y_k$, the *sum* of the products of the xs and ys.

Thus, we have:

$$\left(\begin{array}{cc|c} S_{x^2} & S_x & S_{xy} \\ S_x & m & S_y \end{array}\right)$$

Add $-\frac{S_x}{m}$ times Row 2 onto Row 1, we get:

$$\left(\begin{array}{cc|c} S_{x^2} - \frac{1}{m}S_x^2 & 0 & S_{xy} - \frac{1}{m}S_x S_y \\ S_x & m & S_y \end{array}\right)$$

It may not be obvious, but this is equivalent to a row-echelon form, so we may solve:

$$a_1 = \frac{S_{xy} - \frac{1}{m}S_x S_y}{S_{x^2} - \frac{1}{m}S_x^2},$$

and multiplying this by $1 = \frac{m}{m}$, we have

$$a_1 = \frac{mS_{xy} - S_x S_y}{mS_{x^2} - S_x^2}.$$

Having found this, the second equation says that:

$$a_1 S_x + m a_0 = S_y,$$

so solving this for a_0, we get

$$a_0 = \frac{1}{m}(S_y - a_1 S_x).$$

It is a little more tedious to do Gaussian elimination on the 3×3 system for least-squares best-fitting quadratic polynomials, but adding appropriate multiples of Row 3 onto Rows 2 and 1, as above, and then adding an appropriate multiple of Row 2 onto Row 1, we get that

$$\left(\begin{array}{ccc|c} S_{x^4} & S_{x^3} & S_{x^2} & S_{x^2y} \\ S_{x^3} & S_{x^2} & S_x & S_{xy} \\ S_{x^2} & S_x & m & S_y \end{array}\right) \sim$$

$$\left(\begin{array}{ccc|c} \begin{array}{c} -S_x^2 S_{x^4} + 2S_x S_{x^2} S_{x^3} \\ -S_{x^2}^3 + S_{x^2} S_{x^4} m - S_{x^3}^2 m \end{array} & 0 & 0 & \begin{array}{c} -S_x^2 S_{x^2y} + S_x S_{x^2} S_{xy} + S_x S_{x^3} S_y \\ -S_{x^2}^2 S_y + S_{x^2} S_{x^2y} m - S_{x^3} S_{xy} m \end{array} \\ S_{x^3} m - S_x S_{x^2} & S_{x^2} m - S_x^2 & 0 & S_{xy} m - S_x S_y \\ S_{x^2} & S_x & m & S_y \end{array}\right).$$

From this, we may deduce that:

$$a_2 = \frac{S_x^2 S_{x^2y} - S_x S_{x^2} S_{xy} - S_x S_{x^3} S_y + S_{x^2}^2 S_y - S_{x^2} S_{x^2y} m + S_{x^3} S_{xy} m}{S_x^2 S_{x^4} - 2S_x S_{x^2} S_{x^3} + S_{x^2}^3 - S_{x^2} S_{x^4} m + S_{x^3}^2 m},$$

and hence

$$a_1 = \frac{S_{xy} m - S_x S_y - a_2(S_{x^3} m - S_x S_{x^2})}{S_{x^2} m - S_x^2}, \text{ and}$$

$$a_0 = \frac{1}{m}(S_y - a_2 S_{x^2} - a_1 S_x).$$

Savitzky-Golay filters

The formulas in the previous section must be recomputed for every new set of points, and for large values of x, the corresponding condition number of the matrix will be large, so this will magnify even small errors, including errors from simple floating-point computations.

To avoid this, a common design is to consider only equally spaced x-values, where the period between samples is h, so $t_n = t_0 + nh$. Thus

$$\ldots, (t_{n-3}, y_{n-3}), (t_{n-2}, y_{n-2}), (t_{n-1}, y_{n-1}), (t_n, y_n)$$

are mapped to the non-positive integers

$$\ldots, (-3, y_{n-3}), (-2, y_{n-2}), (-1, y_{n-1}), (0, y_n)$$

This is performed by the simple linear mapping $s \leftarrow \frac{t-t_n}{h}$. Consequently, $t_n + \delta h$ is mapped onto the shifted and scaled value $s = \delta$.

Now, we need only find the solution when the Vandermonde matrix for $N + 1$ points is either

$$V_1 = \begin{pmatrix} 0 & 1 \\ -1 & 1 \\ -2 & 1 \\ -3 & 1 \\ \vdots & \vdots \\ -N & 1 \end{pmatrix} \text{ or } V_2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & -1 & 1 \\ 4 & -2 & 1 \\ 9 & -3 & 1 \\ \vdots & \vdots & \\ N^2 & -N & 1 \end{pmatrix}.$$

Now, as the condition numbers of these matrices is relatively small, and as they are integer matrices, is it actually quite straight-forward to calculate $\left(V_m^\top V_m\right)^{-1} V^\top$ directly, and thus, one can quickly find the coefficients of the least-squares best-fitting polynomial. For example, if $N = 7$, we have

$$\left(V_1^\top V_1\right)^{-1} V^\top = \frac{1}{336} \begin{pmatrix} 28 & 20 & 12 & 4 & -4 & -12 & -20 & -28 \\ 140 & 112 & 84 & 56 & 28 & 0 & -28 & -56 \end{pmatrix}$$

and

$$\left(V_2^\top V_2\right)^{-1} V^\top = \frac{1}{56448} \begin{pmatrix} 2352 & 336 & -1008 & -1680 & -1680 & -1008 & 336 & 2352 \\ 21168 & 5712 & -5040 & -11088 & -12432 & -9072 & -1008 & 11760 \\ 39984 & 21168 & 7056 & -2352 & -7056 & -7056 & -2352 & 7056 \end{pmatrix}$$

This means that the coefficients can be found in $O(mN)$ time where m is the degree of the polynomial being fitted.

Once we have that least-squares best-fitting linear or quadratic polynomial $a_1 s + a_0$ or $a_2 s^2 + a_1 s + a_0$, we can:

1. Approximate the signal $y(t)$ at time $t_n + \delta h$ by calculating $a_1 \delta + a_0$ or $(a_2 \delta + a_1)\delta + a_0$.
2. Approximate the derivative of the signal $y(t)$ at time $t_n + \delta h$ by calculating $\frac{a_1}{h}$ or $\frac{2a_2\delta + a_1}{h^2}$.
3. Approximate the second derivative of the signal $y(t)$ at time $t_n + \delta h$ by calculating $\frac{2a_2}{h^2}$ for the quadratic polynomial.
4. Approximate the integral $\int_{t_{n-1}}^{t_n} y(\tau)\mathrm{d}\tau$ with $h\left(a_0 - \frac{a_1}{2}\right)$ and $h\left(a_0 - \frac{a_1}{2} + \frac{a_2}{3}\right)$.
5. Approximate the integral $\int_{t_n}^{t_{n+1}} y(\tau)\mathrm{d}\tau$ with $h\left(a_0 + \frac{a_1}{2}\right)$ and $h\left(a_0 + \frac{a_1}{2} + \frac{a_2}{3}\right)$.