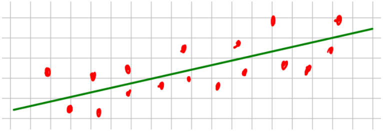
Mathematical Foundations of Computing

Serhii Denysov

February 2026

Let's start with the example - Linear Regression

- "Bad" function f, calculated at $x_1, x_2, \ldots, x_n$ (n points).
- So, we have $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.
  In ML they call it samples, observations etc.
- $y_i \approx f(x_i)$ – values of f, **measured with some error** (noise).
- Assume the process $f(x)$ is essentially linear.
- So, we build "good" $g(x) = c_0 + c_1 x$ – also linear.



Linear regression

To find c_0 and c_1, we need to construct and minimize an **error function**. What intuitive idea of an error function we already know?

Linear regression

To find c_0 and c_1, we need to construct and minimize an **error function**. What intuitive idea of an error function we already know?

- Same value at nodes: $g(x_i) = f(x_i)\ \forall x_i$. But – it makes sense if $y_i = f(x_i)$ are precise!
- Smallest difference idea:

$$\Phi(c_0, c_1) = \sum_{i=1}^{n} (g(x_i) - y_i)^2 = \sum_{i=1}^{n} (c_0 + c_1 x_i - y_i)^2 \to \min_{[a,b]}$$

- We *minimize sum of squared differences* – so it's called **Least Squares Approximation**
- Why **not** $\Phi(c_0, c_1) = \sum_{i=1}^{n} |g(x_i) - y_i| \to \min_{[a,b]}$?

Linear regression

To find c_0 and c_1, we need to construct and minimize an **error function**. What intuitive idea of an error function we already know?

- Same value at nodes: $g(x_i) = f(x_i)\ \forall x_i$. But – it makes sense if $y_i = f(x_i)$ are precise!

- Smallest difference idea:

$$\Phi(c_0, c_1) = \sum_{i=1}^{n} (g(x_i) - y_i)^2 = \sum_{i=1}^{n} (c_0 + c_1 x_i - y_i)^2 \to \min_{[a,b]}$$

- We *minimize sum of squared differences* – so it's called **Least Squares Approximation**

- Why **not** $\Phi(c_0, c_1) = \sum_{i=1}^{n} |g(x_i) - y_i| \to \min_{[a,b]}$?
  Even better, but hard to deal with!

Least squares algorithm

To minimize, set derivatives equal to zero:

$$\frac{\partial \Phi}{\partial c_0} = 0, \quad \frac{\partial \Phi}{\partial c_1} = 0$$

Derivatives:

$$\begin{aligned}
\frac{\partial \Phi}{\partial c_0} &= 2(c_0 + c_1 x_1 - y_1) + \cdots + 2(c_0 + c_1 x_n - y_n) \\
\frac{\partial \Phi}{\partial c_1} &= 2(c_0 + c_1 x_1 - y_1)x_1 + \cdots + 2(c_0 + c_1 x_n - y_n)x_n
\end{aligned}$$

System of Linear Equations

The SLE:

$$2c_0(x_1 + \cdots + x_n) + 2c_1(x_1^2 + \cdots + x_n^2) = 2(y_1x_1 + \cdots + y_nx_n)$$

$$c_0 n + c_1(x_1 + \cdots + x_n) = y_1 + \cdots + y_n$$

Matrix form:

$$Ac = y, \ c = (c_0, c_1)^T$$

$$A = \begin{pmatrix} x_1 + \cdots + x_n & x_1^2 + \cdots + x_n^2 \\ n & x_1 + \cdots + x_n \end{pmatrix}, \quad y = \begin{pmatrix} y_1x_1 + \cdots + y_nx_n \\ y_1 + \cdots + y_n \end{pmatrix}$$

Let's practice!

Let’s raise the level of abstraction

- Again, having ”bad” $f(x)$, we construct a ”good” $g(x)$:

$$g(x) \approx f(x) \quad \forall x \in [a, b]$$

- We need to specify “$\approx$” mathematically.
  What we have for the **distance** between objects?

Let's raise the level of abstraction

- Again, having "bad" $f(x)$, we construct a "good" $g(x)$:

$$g(x) \approx f(x) \quad \forall x \in [a, b]$$

- We need to specify "$\approx$" mathematically.
  What we have for the **distance** between objects? **Norm** of difference!

For now think $1D$.

- **Uniform approximation:**

$$|f(x) - g(x)| < \varepsilon, \quad \forall x \in [a, b] \subseteq D$$

- L_2 **approximation:**

$$||f - g||_{L_2} = \sqrt{\int_a^b (f(x) - g(x))^2 \, dx} < \varepsilon$$

- **So, we need** to find such a function g from some class of functions, that it's distance to f $(\|f - g\|)$ is minimal!

More details

To fully describe the problem, we need to be more specific.

Let g be a linear combination of some simple (basis) functions:

$$g(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \cdots + c_n\varphi_n(x)$$

So we search for the optimal weights (coefficients):

$$c = (c_0, c_1, \ldots, c_n)$$

. to minimize distance between g and f.

So our problem:

$$\|g - f\| \to \min_c$$

We need a norm $\Rightarrow$ we need an inner product

- Many convenient norms induced by a **inner product** (dot product).

- For vectors $u, v \in \mathbb{R}^m$:

$$\langle u, v\rangle = u^T v, \qquad \|u\|_2 = \sqrt{\langle u, u\rangle}$$

- For functions on $[a, b]$ (L_2 case):

$$\langle u, v\rangle = \int_a^b u(x)v(x)\,dx, \qquad \|u\|_{L_2} = \sqrt{\langle u, u\rangle}$$

- How to solve “minimize the norm” problem?

Least squares in inner product form

- We choose an inner product $\langle \cdot, \cdot \rangle$.
  Then the norm is:
  $$\|u\|^2 = \langle u, u \rangle$$

- Our approximation is
  $$g(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \cdots + c_n\varphi_n(x)$$

- Least squares objective:
  $$\Phi(c) = \|f - g\|^2 = \langle f - g,\, f - g \rangle \to \min_c$$

Now: expand $\Phi(c)$ and set partial derivatives to zero.

Expanding the objective $\Phi(c) = \langle f - g, f - g \rangle$

Start:

$$\Phi(c) = \langle f - g, f - g \rangle$$

Expanding the objective $\Phi(c) = \langle f - g, f - g\rangle$

Start:

$$\Phi(c) = \langle f - g, f - g\rangle$$

Expand by bilinearity / symmetry:

$$\langle f - g, f - g\rangle = \langle f, f\rangle - 2\langle f, g\rangle + \langle g, g\rangle$$

Expanding the objective $\Phi(c) = \langle f - g, f - g \rangle$

Start:

$$\Phi(c) = \langle f - g, f - g \rangle$$

Expand by bilinearity / symmetry:

$$\langle f - g, f - g \rangle = \langle f, f \rangle - 2\langle f, g \rangle + \langle g, g \rangle$$

Now substitute $g = \sum_{j=0}^{n} c_j \varphi_j$.

- Middle term:

$$\langle f, g \rangle = \left\langle f, \sum_{j=0}^{n} c_j \varphi_j \right\rangle = \sum_{j=0}^{n} c_j \langle f, \varphi_j \rangle$$

- Last term:

$$\langle g, g \rangle = \left\langle \sum_{i=0}^{n} c_i \varphi_i, \sum_{j=0}^{n} c_j \varphi_j \right\rangle = \sum_{i=0}^{n} c_i \sum_{j=0}^{n} c_j \langle \varphi_i, \varphi_j \rangle$$

So $\Phi(c)$ is a quadratic function of c

Combine everything:

$$\Phi(c) = \langle f, f \rangle - 2 \sum_{j=0}^{n} c_j \langle f, \varphi_j \rangle + \sum_{i=0}^{n} c_i \sum_{j=0}^{n} c_j \langle \varphi_i, \varphi_j \rangle$$

So $\Phi(c)$ is a quadratic function of c

Combine everything:

$$\Phi(c) = \langle f, f \rangle - 2 \sum_{j=0}^{n} c_j \langle f, \varphi_j \rangle + \sum_{i=0}^{n} c_i \sum_{j=0}^{n} c_j \langle \varphi_i, \varphi_j \rangle$$

Important: $\langle f, f \rangle$ does not depend on c, so it disappears after differentiation.

Next: compute $\dfrac{\partial \Phi}{\partial c_k}$ for $k = 0, \ldots, n$.

Derivative with respect to c_k

From:

$$\Phi(c) = \langle f, f \rangle - 2 \sum_{j=0}^{n} c_j \langle f, \varphi_j \rangle + \sum_{i=0}^{n} c_i \sum_{j=0}^{n} c_j \langle \varphi_i, \varphi_j \rangle$$

Derivative with respect to c_k

From:

$$\Phi(c) = \langle f, f \rangle - 2 \sum_{j=0}^{n} c_j \langle f, \varphi_j \rangle + \sum_{i=0}^{n} c_i \sum_{j=0}^{n} c_j \langle \varphi_i, \varphi_j \rangle$$

Derivative of the linear part:

$$\frac{\partial}{\partial c_k} \left(-2 \sum_{j=0}^{n} c_j \langle f, \varphi_j \rangle \right) = -2 \langle f, \varphi_k \rangle$$

Derivative with respect to c_k

From:

$$\Phi(c) = \langle f, f \rangle - 2 \sum_{j=0}^{n} c_j \langle f, \varphi_j \rangle + \sum_{i=0}^{n} c_i \sum_{j=0}^{n} c_j \langle \varphi_i, \varphi_j \rangle$$

Derivative of the linear part:

$$\frac{\partial}{\partial c_k} \left(-2 \sum_{j=0}^{n} c_j \langle f, \varphi_j \rangle \right) = -2 \langle f, \varphi_k \rangle$$

Derivative of the quadratic part:

$$\frac{\partial}{\partial c_k} \left(\sum_{i=0}^{n} c_i \sum_{j=0}^{n} c_j \langle \varphi_i, \varphi_j \rangle \right) = 2 \sum_{j=0}^{n} c_j \langle \varphi_k, \varphi_j \rangle$$

Derivative with respect to c_k

From:

$$\Phi(c) = \langle f, f\rangle - 2\sum_{j=0}^{n} c_j \langle f, \varphi_j\rangle + \sum_{i=0}^{n} c_i \sum_{j=0}^{n} c_j \langle \varphi_i, \varphi_j\rangle$$

Derivative of the linear part:

$$\frac{\partial}{\partial c_k}\left(-2\sum_{j=0}^{n} c_j \langle f, \varphi_j\rangle\right) = -2\langle f, \varphi_k\rangle$$

Derivative of the quadratic part:

$$\frac{\partial}{\partial c_k}\left(\sum_{i=0}^{n} c_i \sum_{j=0}^{n} c_j \langle \varphi_i, \varphi_j\rangle\right) = 2\sum_{j=0}^{n} c_j \langle \varphi_k, \varphi_j\rangle$$

So:

$$\frac{\partial \Phi}{\partial c_k} = -2\langle f, \varphi_k\rangle + 2\sum_{j=0}^{n} c_j \langle \varphi_k, \varphi_j\rangle$$

Setting derivatives to zero $\Rightarrow$ SLE

Optimality condition:

$$\frac{\partial \Phi}{\partial c_k} = 0, \qquad k = 0, 1, \ldots, n$$

Substitute the derivative:

$$-2\langle f, \varphi_k \rangle + 2 \sum_{j=0}^{n} c_j \langle \varphi_k, \varphi_j \rangle = 0$$

Setting derivatives to zero $\Rightarrow$ SLE

Optimality condition:

$$\frac{\partial \Phi}{\partial c_k} = 0, \qquad k = 0, 1, \ldots, n$$

Substitute the derivative:

$$-2\langle f, \varphi_k \rangle + 2\sum_{j=0}^{n} c_j \langle \varphi_k, \varphi_j \rangle = 0$$

Divide by 2:

$$\sum_{j=0}^{n} c_j \langle \varphi_k, \varphi_j \rangle = \langle f, \varphi_k \rangle, \qquad k = 0, \ldots, n$$

This is our system of linear equations for coefficients $c_0, \ldots, c_n$.

Matrix form: Gram matrix and right-hand side

Define the **Gram matrix** G:

$$G_{kj} = \langle \varphi_k, \varphi_j \rangle, \qquad k, j = 0, \dots, n$$

and the vector d:

$$d_k = \langle f, \varphi_k \rangle, \qquad k = 0, \dots, n$$

Then the SLE becomes:

$$Gc = d$$

- G is symmetric.
- If $\{\varphi_0, \dots, \varphi_n\}$ are linearly independent, then G is positive definite $\Rightarrow$ the solution is unique.

Can supernice Gram matrix turn a nightmare?

Very natural and intuitive setup:

$$\langle f, g \rangle = \int_{-1}^{1} f \cdot g \; dx$$

$$\varphi_i(x) = x^i, \; i = 0 \ldots n$$

$$g(x) = p_n(x) = c_0 + c_1 x + \cdots + c_n x^n$$

Gram matrix G becomes... Hilbert matrix!

Check it!

Practical recipe: Normal equations (discrete least squares)

- Input data – points (x_i, y_i), $i = 1, \dots, m$.
- Function to find:

$$g(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \cdots + c_n\varphi_n(x)$$

- Examples:
  - $g(x) = c_0 + c_1 x$

    $(n = 1 \text{ and } \varphi_j(x) = x^j)$
  - $g(x) = c_0 + c_1 x + c_2 x^2$

    $(n = 2 \text{ and } \varphi_j(x) = x^j)$
  - $g(x) = c_0 + c_1 \sin(x) + c_2 \cos(x)$

    $(n = 2 \text{ and } \varphi_0(x) = 1, \varphi_1(x) = sin(x), \varphi_2(x) = cos(x))$

Normal equations

Ideally, we want to solve a system of linear equations (SLE):

$$\begin{cases} g(x_1) = y_1, \\ g(x_2) = y_2, \\ \dots \\ g(x_m) = y_m \end{cases}$$

Or if we write down all the details:

$$\begin{cases} c_0\varphi_0(x_1) + c_1\varphi_1(x_1) + \cdots + c_n\varphi_n(x_1) = y_1, \\ c_0\varphi_0(x_2) + c_1\varphi_1(x_2) + \cdots + c_n\varphi_n(x_2) = y_2, \\ \dots \\ c_0\varphi_0(x_m) + c_1\varphi_1(x_m) + \cdots + c_n\varphi_n(x_m) = y_m \end{cases}$$

Frame Title

The matrix of the SLE is

$$A = \begin{pmatrix} \varphi_0(x_1) & \varphi_1(x_1) & \dots & \varphi_n(x_1) \\ \varphi_0(x_2) & \varphi_1(x_2) & \dots & \varphi_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_0(x_m) & \varphi_1(x_m) & \dots & \varphi_n(x_m) \end{pmatrix}$$

- It has m rows and $n+1$ columns, usually not square – we can not solve it directly
- In practice often m is much bigger than n, like we have 100 points and three unknows.
- So instead of solving SLE, we search for coefficients c_i which will minimize

$$\Phi(c) = \|Ac - y\| \tag{1}$$

The theory above (with making derivatives zero etc.) gives us simple approach to solve (1):

- Build matrix $B = A^T A$. This matrix will be a square one.

- Build vector $z = A^T y$, where $y = (y_1, y_2, \dots, y_m)^T$

- Solve the system: $Bc = z$

- Resulting vector $c = (c_0, c_1, ..., c_n)$ will give the best least squares approximation (function $g(x)$).

Example 1: fit a line to 4 points

Data:

$$(0,0),\ (1,1),\ (2,0),\ (3,1)$$

$$g(x) = c_0 + c_1 x \quad \Rightarrow \quad \varphi_0(x) = 1,\ \varphi_1(x) = x$$

The initial SLE:

$$\begin{cases} c_0 + 0 \cdot c_1 = 0, \\ c_0 + 1 \cdot c_1 = 1, \\ c_0 + 2 \cdot c_1 = 0, \\ c_0 + 3 \cdot c_1 = 1 \end{cases}$$

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

$$y = (0,1,0,1)$$

Fit line to 4 points

$$A^T = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{pmatrix}$$

$$B = A^T A = \begin{pmatrix} 4 & 6 \\ 6 & 14 \end{pmatrix}$$

$$z = A^T y = (2, 4)$$

So, we need to solve

$$Bc = z$$

.

The solution is: $c = (\frac{1}{5}, \frac{1}{5})$

Example 2: fit with a sin

Data:

$$(0,0),\ (1,1),\ (2,0)$$

$$g(x) = c_0 + c_1 \sin(x) \quad \Rightarrow \quad \varphi_0(x) = 1,\ \varphi_1(x) = \sin(x)$$

The initial SLE:

$$\begin{cases} c_0 + \sin(0) \cdot c_1 = 0, \\ c_0 + \sin(1) \cdot c_1 = 1, \\ c_0 + \sin(2) \cdot c_1 = 0 \end{cases}$$

$$A = \begin{pmatrix} 1 & \sin(0) \\ 1 & \sin(1) \\ 1 & \sin(2) \end{pmatrix}$$

$$y = (0, 1, 0)$$

Fit with a sin

$$A^T = \begin{pmatrix} 1 & 1 & 1 \\ 0 & \sin(1) & \sin(2) \end{pmatrix}$$

$$B = A^T A = \begin{pmatrix} 3 & \sin(1) + \sin(2) \\ \sin(1) + \sin(2) & \sin(1)^2 + \sin(2)^2 \end{pmatrix}$$

$$z = A^T y = (1, \sin(1))$$

So, we again need to solve

$$Bc = z$$

All the same, only need to calculate values of sin.