A brief introduction to geometric fitting algorithms pt. 0

Alejandro Garnung Menéndez

28 January 2025

Contents

Least-squares

Least squares is one of our most reliable methods for geometric fitting; not only is it practical (explicit), but it is also mathematically elegant (convex, in mathematical terms).

The classic process of fitting a line $y = mx + b$ to data points, which are not guaranteed to fit perfectly, involves minimizing the following least-squares error function:

$$D = \sum_{i=1}^{n} \left(f(x_i) - y_i\right)^2 = \sum_{i=1}^{n} \left(mx_i + b - y_i\right)^2, \tag{1}$$

where $f(x) = mx + b$. This expression measures vertical distances, i.e., deviations in the direction of the y-axis, in a 2D axis representation.

Alternative approaches to least squares involve using norms other than L_2, for example, the L_1 norm:

$$D' = \sum_{i=1}^{n} \left|f(x_i) - y_i\right|,$$

Or the perpendicular distance:

$$D'' = \frac{|Ax_i + By_i + C|}{\sqrt{A^2 + B^2}}.$$

It is worth mentioning that minimizing distances orthogonal to the line (total least squares), instead of the vertical ones, is often more robust to outliers.

The extremality condition for the scalar function D (refer to Equation (1)) is:

$$\frac{\partial D}{\partial b} = \sum_{i=1}^{n} 2 \cdot (mx_i + b - y_i) = 0,$$

$$\frac{\partial D}{\partial m} = \sum_{i=1}^{n} 2x_i \cdot (mx_i + b - y_i) = 0.$$

For example, if $n = 2$ (to satisfy algebraically the necessary condition for a complete system with 2 unknowns), we obtain:

$$\begin{aligned} x_0(mx_0 + b - y_0) + x_1(mx_1 + b - y_1) &= 0, \\ (mx_0 + b - y_0) + (mx_1 + b - y_1) &= 0. \end{aligned}$$

In matrix form, the above system can be written as:

$$\begin{bmatrix} 1 & 1 \\ x_0 & x_1 \end{bmatrix} \cdot \begin{bmatrix} b + mx_0 - y_0 \\ b + mx_1 - y_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Expanding further:

$$\begin{bmatrix} 1 & 1 \\ x_0 & x_1 \end{bmatrix} \cdot \left(\begin{bmatrix} 1 & x_0 \\ 1 & x_1 \end{bmatrix} \cdot \begin{bmatrix} b \\ m \end{bmatrix} - \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

This can also be expressed compactly as:

$$A^T \cdot A \cdot x - A^T \cdot y = 0.$$

Here, A is the design matrix, $x = \begin{bmatrix} b \\ m \end{bmatrix}$, and $y = \begin{bmatrix} y_0 \\ y_1 \end{bmatrix}$.

We can now solve the system using Singular Value Decomposition (SVD) or the Moore-Penrose pseudoinverse. Using the pseudoinverse, the solution is given by:

$$x = (A^T A)^{-1} A^T y,$$

with the condition that $A^T A$ is invertible (which requires that the points are distinct).

Fitting polynomials

Generalizing the line equation (i.e. the first-degree polynomial) to a k-order polynomial:

$$y = a_0 + a_1 x + \cdots + a_k x^k \tag{2}$$

The remainder of the optimization problem is given by:

$$R^2 = \sum_{i=1}^{n} \left(y_i - \left(a_0 + a_1 x_i + \cdots + a_k x_i^k \right) \right)^2 \tag{3}$$

The partial derivatives set to zero, necessary to satisfy the optimality condition for finding the unknowns that minimize the residual, are:

$$\frac{\partial R^2}{\partial a_0} = -2 \sum_{i=1}^{n} \left(y_i - \left(a_0 + a_1 x_i + \cdots + a_k x_i^k \right) \right) = 0$$

$$\frac{\partial R^2}{\partial a_1} = -2\sum_{i=1}^{n}\left(y_i - \left(a_0 + a_1 x_i + \cdots + a_k x_i^k\right)\right) x_i = 0$$

$$\frac{\partial R^2}{\partial a_k} = -2\sum_{i=1}^{n}\left(y_i - \left(a_0 + a_1 x_i + \cdots + a_k x_i^k\right)\right) x_i^k = 0$$

This leads to a system of nonlinear equations (if $k > 1$):

$$a_0 n + a_1 \sum_i x_i + \cdots + a_k \sum_i x_i^k = \sum_i y_i$$

$$a_0 \sum_i x_i + a_1 \sum_i x_i^2 + \cdots + a_k \sum_i x_i^{k+1} = \sum_i x_i y_i$$

$$a_0 \sum_i x_i^k + a_1 \sum_i x_i^{k+1} + \cdots + a_k \sum_i x_i^{2k} = \sum_i x_i^k y_i$$

Rewritten in matrix form:

$$\begin{bmatrix} n & \sum x_i & \dots & \sum x_i^k \\ \sum x_i & \sum x_i^2 & \dots & \sum x_i^{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum x_i^k & \sum x_i^{k+1} & \dots & \sum x_i^{2k} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \vdots \\ \sum x_i^k y_i \end{bmatrix}$$

$$G \cdot \mathbf{a} = \mathbf{b}.$$

The first matrix, G, is the Gram matrix. These are denoted the "normal equations". It is straightforward to demonstrate that we can write the normal equations using the Vandermonde matrix, V:

$$G \cdot \mathbf{a} = \mathbf{b} \iff V^T V \mathbf{a} = V^T \mathbf{y},$$

i.e.:

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^k & x_2^k & \dots & x_n^k \end{bmatrix} \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^k \\ 1 & x_2 & x_2^2 & \dots & x_2^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^k & x_2^k & \dots & x_n^k \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

Thus, given n tuples (x_i, y_i), and fitting a polynomial of degree k with coefficients $a_0, \dots, a_k$, we have:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^k \\ 1 & x_2 & x_2^2 & \dots & x_2^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

Let $X \equiv V$, so $X \cdot \mathbf{a} = \mathbf{y}$.

By premultiplying by X^T, the problem can be solved numerically, or by directly inverting (if the problem is well-posed / the matrix is well-formed):

$$\mathbf{a} = (X^T X)^{-1} X^T \mathbf{y},$$

which is the least-squares solution for the Vandermonde polynomial.