L_2 project polynomials along a path

Oliver K. Ernst

August 20, 2018

Contents

1 Polynomial in one variable **1**

2 General formula **2**

3 Polynomial in arbitrary number of variables **2**

4 Essentials of cubic interpolation in 1D **4**

5 Cubic interpolation in arbitrary dimension **4**

1 Polynomial in one variable

Let the polynomial (in 1D) be:

$$f(x) = \sum_{l=0}^{K} \alpha_l x^l \tag{1}$$

Let the data be of the form a trajectory $(x(t), y(t))$ for $t \in [0, T]$.

We want to minimize:

$$\int_C ds \ (f(x) - y)^2 = \int_{t=0}^{t=T} dt \, |s'(t)| \, (f(x(t)) - y(t))^2 \tag{2}$$

along the curve of data points. Here:

$$\begin{aligned} s'(t) &= \frac{dx(t)}{dt} \\ |s'(t)| &= \sqrt{\left(\frac{dx(t)}{dt}\right)^2} \end{aligned} \tag{3}$$

But we don't have $(x(t), y(t))$ - instead, we only have discrete data points: (x_i, y_i) for sample indexes $i = 1, \ldots, N$, with spacing Δt. The integral becomes:

$$\int_{t=0}^{t=T} dt \, |s'(t)| \, (f(x(t)) - y(t))^2 \approx \Delta t \sum_{i=1}^{N-1} |s'_i| \, (f(x_i) - y_i)^2 \tag{4}$$

where

$$|s'_i| \approx \frac{\sqrt{(x_{i+1} - x_i)^2}}{\Delta t} \tag{5}$$

gives

$$\sum_{i=1}^{N-1} \left(f(x_i) - y_i\right)^2 \sqrt{\left(x_{i+1} - x_i\right)^2} = \sum_{i=1}^{N-1} \left(\sum_{l=0}^{K} \alpha_l x_i^l - y_i \right)^2 \sqrt{\left(x_{i+1} - x_i\right)^2} \tag{6}$$

Minimizing with respect to some param α_k gives:

$$2 \sum_{i=1}^{N-1} \left(\sum_{l=0}^{K} \alpha_l x_i^l - y_i \right) x_i^k \sqrt{\left(x_{i+1} - x_i\right)^2} = 0 \tag{7}$$

Rewrite this in the form:

$$\vec{a}_k^{\intercal} \cdot \vec{\alpha} = b_k \tag{8}$$

where $\vec{\alpha}^{\intercal} = (\alpha_0, \dots, \alpha_K)$ of length $K + 1$ and the constant term is:

$$b_k = \sum_{i=1}^{N-1} y_i x_i^k \sqrt{\left(x_{i+1} - x_i\right)^2} \tag{9}$$

and the vector elements are:

$$a_{kl} = \sum_{i=1}^{N-1} x_i^{l+k} \sqrt{\left(x_{i+1} - x_i\right)^2} \tag{10}$$

Differentiating with respect to each of the $K + 1$ parameters, we can form a $K + 1 \times K + 1$ matrix $\boldsymbol{A}$ and vector $\vec{b}$:

$$\boldsymbol{A}\vec{\alpha} = \vec{b} \tag{11}$$

where the elements of $\boldsymbol{A}$ are a_{kl} and the elements of $\vec{b}$ are b_k (indexed $0, \dots, K$).

2 General formula

Let the polynomial be generally d dimensional. Let these be denoted by $x^{[1]}, \dots, x^{[d]}$.

The general formula is as follows: for each parameter $\alpha_{<k>}$, let the corresponding x term in the polynomial be $\hat{x}_{<k>}$ as some product of the $x^{[j]}$ raised to some powers. Let evaluating it at point x_i indexed by i with corresponding value y_i be denoted by $(\hat{x}_{<k>})_i$.

$$b_k = \sum_{i=1}^{N-1} y_i \times (\hat{x}_{<k>})_i \times \sqrt{\sum_{j=1}^{d} (x_{i+1}^{[j]} - x_i^{[j]})^2} \tag{12}$$

and the elements of the matrix as

$$a_{kl} = \sum_{i=1}^{N-1} (\hat{x}_{<k>})_i \times (\hat{x}_{<l>})_i \times \sqrt{\sum_{j=1}^{d} (x_{i+1}^{[j]} - x_i^{[j]})^2} \tag{13}$$

where the elements of the $N \times N$ matrix $\boldsymbol{A}$ are $a_{k\lambda}$, and the N elements of $\vec{b}$ are b_k.

3 Polynomial in arbitrary number of variables

We now consider forming a general d-dimensional polynomial up to cutoff order K. How many terms are in the polynomial?

- Of order 0 terms, there is only 1.

- Of order 1 terms, there are d terms: $x^{[1]}, \dots, x^{[d]}$.
- Of order 2 terms, there d pure terms $(x^{[1]})^2, \dots, (x^{[[]}d])^2$, and $\binom{d}{2}$ mixing terms: $(x^{[[]}1]x^{[[]}2]), \dots$. This gives a total of $d + \binom{d}{2} = \binom{d}{1} + \binom{d}{2}$ terms.
- Of order $l \geq 1$ terms, there are $\sum_{p=1}^{l} \binom{d}{p}$ terms.

For a general polynomial then of order $l = 0, \dots, K$, we have for the number of terms:

$$M = 1 + \sum_{l=1}^{K} \sum_{p=1}^{l} \binom{d}{p} \tag{14}$$

Let us try to write a general form. A general term of *any* order in the polynomial is:

$$\prod_{j=1}^{d} (x^{[j]})^{p_j} \tag{15}$$

Here, p_j is the power of the term in the j-th dimension.

A general term of order l in the polynomial is:

$$\left(\prod_{j=1}^{d-1} (x^{[j]})^{p_j} \right) (x^{[d]})^{l - \sum_{r=1}^{d-1} p_r} \tag{16}$$

Here, the last term enforces the constraint that the term is of order l. Note that this is irregardless of whether any power p_j is greater than l, leading to some other power being less than zero - we will deal with this momentarily.

To shorten notation, introduce:

$$P_{d-1} = \sum_{r=1}^{d-1} p_r \tag{17}$$

We are now in position to write the general form of the polynomial:

$$f(x^{[1]}, \dots, x^{[d]}) = \sum_{l=0}^{K} \sum_{p_1=0}^{l} \sum_{p_2=0}^{l-P_1} \cdots \sum_{p_{d-1}=0}^{l-P_{d-2}} \left(\prod_{j=1}^{d-1} (x^{[j]})^{p_j} \right) (x^{[d]})^{l-P_{d-1}} \alpha_{l,p_1,\dots,p_{d-1}} \tag{18}$$

Here, $\alpha_{l,p_1,\dots,p_{d-1}}$ are the coefficients to determine by L_2 projection.

Following the arguments in the 1D case, the discrete sum to minimize is:

$$\sum_{i=1}^{N-1} \left(f(x_i^{[1]}, \dots, x_i^{[d]}) - y_i \right)^2 \sqrt{\sum_{j=1}^{d} (x_{i+1}^{[j]} - x_i^{[j]})^2} \tag{19}$$

where we may substitute (18). Differentiating with respect to some $\alpha_{k,q_1,\dots,q_{d-1}}$ gives:

$$2 \sum_{i=1}^{N-1} \left(f(x_i^{[1]}, \dots, x_i^{[d]}) - y_i \right) \left(\prod_{j=1}^{d-1} (x_i^{[j]})^{q_j} \right) (x_i^{[d]})^{k-Q_{d-1}} \sqrt{\sum_{j=1}^{d} (x_{i+1}^{[j]} - x_i^{[j]})^2} = 0 \tag{20}$$

where analogous to P_{d-1} before: $Q_{d-1} = \sum_{r=1}^{d-1} q_r$.

Assign the indexes $(l, p_1, \dots, p_{d-1})$ representing the coefficients α in the polynomial some ordering: $(l, p_1, \dots, p_{d-1}) \to \lambda$. Note that each of these indexes obey: $0 \leq p_j \leq l$ as well as $P_{d-1} = \sum_{r=1}^{d-1} p_r \leq l$.

Then we may write this in the compact form as before:

$$\vec{a}_k^\intercal \cdot \vec{\alpha} = b_k \tag{21}$$

where the vectors are of length M, the constant term is:

$$b_k = \sum_{i=1}^{N-1} y_i \left(\prod_{j=1}^{d-1} (x_i^{[j]})^{q_j} \right) (x_i^{[d]})^{k-Q_{d-1}} \sqrt{\sum_{j=1}^{d} (x_{i+1}^{[j]} - x_i^{[j]})^2} \tag{22}$$

and the elements of the vector are

$$a_{k\lambda} = \sum_{i=1}^{N-1} \left(\prod_{j=1}^{d-1} (x_i^{[j]})^{p_j+q_j} \right) (x_i^{[d]})^{l+k-P_{d-1}-Q_{d-1}} \sqrt{\sum_{j=1}^{d} (x_{i+1}^{[j]} - x_i^{[j]})^2} \tag{23}$$

for the appropriate $(l, p_1, \ldots, p_{d-1}) \leftrightarrow \lambda$.

The solution is now given as above with $M \times M$ matrix $\boldsymbol{A}$ and vector $\vec{b}$ of length M.

4 Essentials of cubic interpolation in 1D

Summarizing the essentials of Paul Breeuwsma's website.

Consider being given 4 points: $(-1, p_0), (0, p_1), (1, p_2), (2, p_3)$. The goal is to estimate the value at a point $x \in [0, 1]$ by cubic splines. The formula is:

$$f(p_0, p_1, p_2, p_3, x) = \left(-\frac{1}{2}p_0 + \frac{3}{2}p_1 - \frac{3}{2}p_2 + \frac{1}{2}p_3 \right) x^3 + \left(p_0 - \frac{5}{2}p_1 + 2p_2 - \frac{1}{2}p_3 \right) x^2 + \left(-\frac{1}{2}p_0 + \frac{1}{2}p_2 \right) x + p_1 \tag{24}$$

5 Cubic interpolation in arbitrary dimension

1. Locate the neighborhood of 4 points in all d dimensions, for a total of 4^d points.
2. Draw lines along one dimension, for a total of 4^{d-1} lines, each of length 4 points.
3. Perform 1D interpolation along each line to obtain 4^{d-1} points.
4. Repeat 1.

The resulting $f(\boldsymbol{x})$ can be used in the L_2 projection above. For every data point, there are 4^d surrounding points. A well-determined problem is guaranteed if there are at least 4^d data points inside each voxel.