Abstract

Formulas for finding a weighted least-squares fit to a vector-to-vector transformation are provided in two cases: (1) when the mapping is available as a continuous analytical function on a known domain, and (2) when the mapping function is available only approximately through knowledge of a finite collection of input and output sample vector pairs. The formulas in each case are very similar to each other, and also similar to more familiar linear regression formulas described in elementary linear-algebra textbooks.

Continuous and discrete formulas for multi-linear(affine) regression

R.M. Brannon

July 24, 2011

1 Introduction

A vector transformation $\underset{\sim}{\boldsymbol{y}} = \underset{\sim}{\boldsymbol{y}}(\underset{\sim}{\boldsymbol{x}})$ takes a vector $\underset{\sim}{\boldsymbol{x}}$ as input and returns a vector $\underset{\sim}{\boldsymbol{y}}$ as output. Figure 1 shows a square domain, Ω, in gray. For each $\underset{\sim}{\boldsymbol{x}} \in \Omega$, the mapping function may be applied to generate a transformed shape. The left side of Fig. 1 shows, in blue, the result of applying a nonlinear transformation to the square domain Ω. The nonlinearity is evident since the deformed (i.e., transformed) shape is distorted into curves. An affine transformation is the generalization of the affine scalar equation $y = mx + b$; specifically, an affine transformation can always be written in the form $\underset{\sim}{\boldsymbol{y}} = \underset{\approx}{\boldsymbol{M}} \cdot \underset{\sim}{\boldsymbol{x}} + \underset{\sim}{\boldsymbol{b}}$, for some second-order tensor $\underset{\approx}{\boldsymbol{M}}$ and vector $\underset{\sim}{\boldsymbol{b}}$. An affine transformation will always transform a square domain into a parallelogram. In particular, the right-hand side of Fig. 1 depicts a least-squares "best-fit" of the nonlinear transformation. The goal of this document is to illustrate how to find the best affine fit to a nonlinear vector-to-vector transformation. The formulas will be very similar to those for fitting nonlinear scalar funtions $y(x)$ to a straight line.

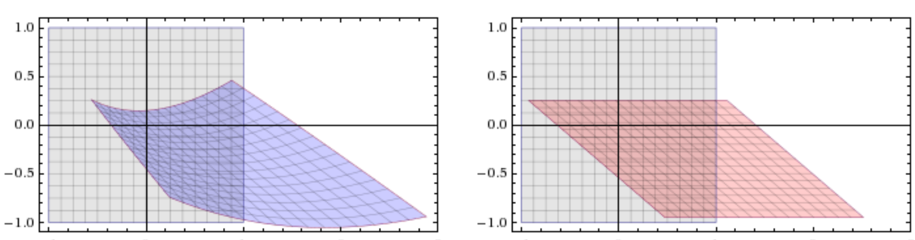

Figure 1: Left: continuous data. Right: continuous affine fit

In numerical work, an analytic continuous mapping function $\underset{\sim}{\boldsymbol{y}}(\underset{\sim}{\boldsymbol{x}})$ might not be available. Instead, a discrete approximate sampling of the mapping might be the only information available to approximately characterize the transformation. In other words, numerical methods often describe a mapping transformation only by knowing how, as illustrated in Fig. 2(a), a collection of discrete points in the initial domain transform. Below, we will demonstrate that finding the best fit to this type of discrete data is very similar to the continuous case. The formulas will be very similar to classical linear regression for fitting to scalar (x,y) data pairs.

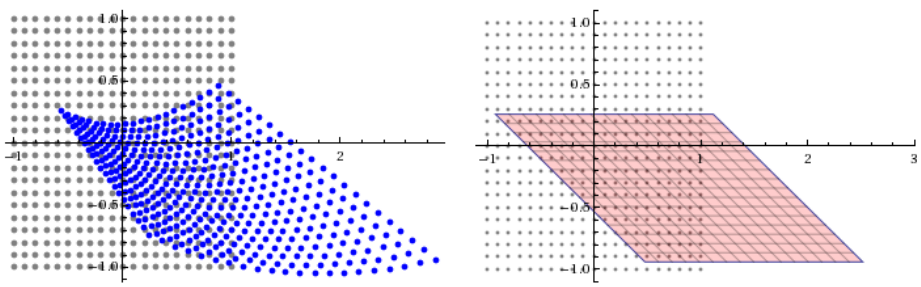

Figure 2: Left: Discrete data. Right: continuous affine fit.

When finding an approximate affine fit to data, continuous or discrete, you might wish for the fit to be better near some points of interest. This is accomplished by specifying a weighting function $w(\underset{\sim}{\boldsymbol{x}})$ or, in the discrete case, by giving weights w_i for each discrete input-output vector pair. The effect of weighting is illustrated in Fig. 3.

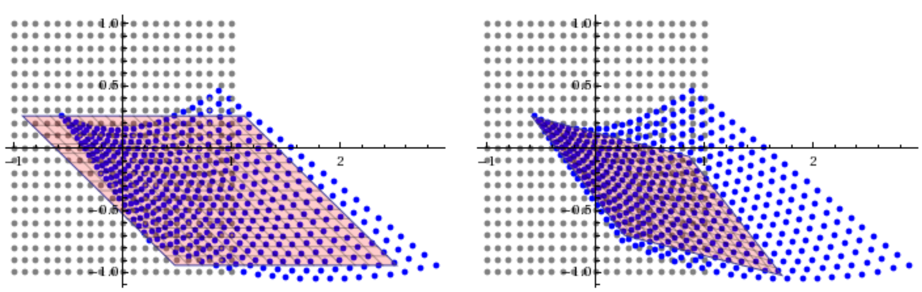

Figure 3: Left: Equally weighted fit. Right: A fit that is weighted to improve the approximation near the upper-left cusp on the mapped region (with the price being, of course, a poorer fit away from that cusp).

2 Problem Statement

Suppose that you have a nonlinear vector-to-vector function, $\underset{\sim}{\boldsymbol{y}} = \underset{\sim}{\boldsymbol{y}}(\underset{\sim}{\boldsymbol{x}})$, and you seek a constant tensor $\underset{\approx}{\boldsymbol{M}}$ and vector $\underset{\sim}{\boldsymbol{b}}$ such that the affine transformation, $\underset{\sim}{\boldsymbol{y}} = \underset{\approx}{\boldsymbol{M}}\cdot\underset{\sim}{\boldsymbol{x}}+\underset{\sim}{\boldsymbol{b}}$ is a "best fit" to the nonlinear function $\underset{\sim}{\boldsymbol{y}}(\underset{\sim}{\boldsymbol{x}})$ over a specified region in space, $\underset{\sim}{\boldsymbol{x}} \in \Omega$. Below, we provide an algorithm corresponding to minimizing the weighted mean square residual, $R^2 = R^2(\underset{\approx}{\boldsymbol{M}}, \underset{\sim}{\boldsymbol{b}})$, defined by

$$R^2 = \frac{\int_\Omega \underset{\sim}{\boldsymbol{r}} \cdot \underset{\sim}{\boldsymbol{r}}\; w dV}{\int_\Omega w dV} \tag{1}$$

in which dV is the volume element for integration over the input domain $\underset{\sim}{\boldsymbol{x}} \in \Omega$, $w = w(\underset{\sim}{\boldsymbol{x}})$ is a scalar-valued weighting function (e.g., it could be the mass density if you want to have a mass-weighted residual in a mechanics problem), and $\underset{\sim}{\boldsymbol{r}} = \underset{\sim}{\boldsymbol{r}}(\underset{\sim}{\boldsymbol{x}}; \underset{\approx}{\boldsymbol{M}}, \underset{\sim}{\boldsymbol{b}})$ is the local residual vector field defined by

$$\underset{\sim}{\boldsymbol{r}}(\underset{\sim}{\boldsymbol{x}}; \underset{\approx}{\boldsymbol{M}}, \underset{\sim}{\boldsymbol{b}}) = \underset{\sim}{\boldsymbol{y}}(\underset{\sim}{\boldsymbol{x}}) - \left(\underset{\approx}{\boldsymbol{M}} \cdot \underset{\sim}{\boldsymbol{x}} + \underset{\sim}{\boldsymbol{b}}\right). \tag{2}$$

Note that the local residual vector $\underset{\sim}{\boldsymbol{r}}$ depends on the location $\underset{\sim}{\boldsymbol{x}}$ and on the unknown (sought) tensor $\underset{\approx}{\boldsymbol{M}}$ and vector $\underset{\sim}{\boldsymbol{b}}$. Thus, after integration over $\underset{\sim}{\boldsymbol{x}} \in \Omega$, the resulting weighted mean square residual, R^2 depends only on the unknowns, and may therefore be minimized in by setting each partial derivative of R^2 with respect to the components of $\underset{\approx}{\boldsymbol{M}}$ and $\underset{\sim}{\boldsymbol{b}}$ equal to zero. This provides a set of equations solvable for these unknowns. Rather than showing this tedious analysis, this document provides only the algorithm for the final answer.

In addition to providing the algorithm for this continuous fitting exercise, we will also provide very similar formulas for fitting discrete mapping data. Unlike the continuous fitting problem described above, which presumed that the mapping function $\underset{\sim}{\boldsymbol{y}}(\underset{\sim}{\boldsymbol{x}})$ was known, the discrete problem presumes that you have only discrete values of the mapping function's output, $\{\underset{\sim}{\boldsymbol{y}}_1, \underset{\sim}{\boldsymbol{y}}_2, ...\underset{\sim}{\boldsymbol{y}}_N\}$ corresponding to a finite discrete set of inputs $\{\underset{\sim}{\boldsymbol{x}}_1, \underset{\sim}{\boldsymbol{x}}_2, ..., \underset{\sim}{\boldsymbol{x}}_N\}$. Rather than having a continuous weighting function $w(\underset{\sim}{\boldsymbol{x}})$, the discrete multi-linear regression problem presumes that you have a table of weight values $\{w_1, w_2, ..., w_N\}$ for each discrete point. For the discrete problem, the mean square residual is defined similarly to that in Eq. (1), except that the integral is replaced by summation

$$R^2 = \frac{\sum_{i=1}^N \underset{\sim}{\boldsymbol{r}}_i \cdot \underset{\sim}{\boldsymbol{r}}_i\, w_i}{\sum_{i=1}^N w_i}. \tag{3}$$

Here, analogous to Eq. (2), the discrete residual vectors $\{\underset{\sim}{\boldsymbol{r}}_1, \underset{\sim}{\boldsymbol{r}}_2, ..., \underset{\sim}{\boldsymbol{r}}_N\}$ are defined by

$$\underset{\sim}{\boldsymbol{r}}_i(\underset{\approx}{\boldsymbol{M}}, \underset{\sim}{\boldsymbol{b}}) = \underset{\sim}{\boldsymbol{y}}_i - \left(\underset{\approx}{\boldsymbol{M}} \cdot \underset{\sim}{\boldsymbol{x}}_i + \underset{\sim}{\boldsymbol{b}}\right). \tag{4}$$

3 Algorithm

To find the desired tensor $\underset{\approx}{\boldsymbol{M}}$ and vector $\underset{\sim}{\boldsymbol{b}}$, the calculations shown below should be performed in sequence (using the continuous or discrete formula as appropriate).

Begin by evaluating the "weight" of the domain:

$$W = \int_{\Omega} w dV \qquad = \sum_{i=1}^{N} w_i \tag{5}$$

Next, evaluate the first moment vectors $\underset{\sim}{\overline{\boldsymbol{x}}}$ and $\underset{\sim}{\overline{\boldsymbol{y}}}$, which would be called the centers of mass in mechanics:

$$\underset{\sim}{\overline{\boldsymbol{x}}} = \frac{1}{W}\int_{\Omega} \underset{\sim}{\boldsymbol{x}}\, w dV \qquad = \frac{1}{W}\sum_{i=1}^{N} \underset{\sim}{\boldsymbol{x}}_i w_i \tag{6}$$

$$\underset{\sim}{\overline{\boldsymbol{y}}} = \frac{1}{W}\int_{\Omega} \underset{\sim}{\boldsymbol{y}}\, w dV \qquad = \frac{1}{W}\sum_{i=1}^{N} \underset{\sim}{\boldsymbol{y}}_i w_i \tag{7}$$

Evaluate two "helper" tensors:

$$\underset{\approx}{\boldsymbol{g}} = \left(\frac{1}{W}\int_{\Omega} \underset{\sim}{\boldsymbol{y}}\underset{\sim}{\boldsymbol{x}}\, w dV\right) - \underset{\sim}{\overline{\boldsymbol{y}}}\,\underset{\sim}{\overline{\boldsymbol{x}}} \qquad = \left(\frac{1}{W}\sum_{i=1}^{N} \underset{\sim}{\boldsymbol{y}}_i \underset{\sim}{\boldsymbol{x}}_i w_i\right) - \underset{\sim}{\overline{\boldsymbol{y}}}\,\underset{\sim}{\overline{\boldsymbol{x}}} \tag{8}$$

$$\underset{\approx}{\boldsymbol{G}} = \left(\frac{1}{W}\int_{\Omega} \underset{\sim}{\boldsymbol{x}}\underset{\sim}{\boldsymbol{x}}\, w dV\right) - \underset{\sim}{\overline{\boldsymbol{x}}}\,\underset{\sim}{\overline{\boldsymbol{x}}} \qquad = \left(\frac{1}{W}\sum_{i=1}^{N} \underset{\sim}{\boldsymbol{x}}_i \underset{\sim}{\boldsymbol{x}}_i w_i\right) - \underset{\sim}{\overline{\boldsymbol{x}}}\,\underset{\sim}{\overline{\boldsymbol{x}}} \tag{9}$$

Then the final answer for the affine mapping tensor $\underset{\approx}{\boldsymbol{M}}$ is

$$\underset{\approx}{\boldsymbol{M}} = \underset{\approx}{\boldsymbol{g}} \cdot \underset{\approx}{\boldsymbol{G}}^{-1} \tag{10}$$

and the final answer for the "intercept" is

$$\underset{\sim}{\boldsymbol{b}} = \underset{\sim}{\overline{\boldsymbol{y}}} - \underset{\approx}{\boldsymbol{M}} \cdot \underset{\sim}{\overline{\boldsymbol{x}}} \tag{11}$$

Note that the affine mapping, $\underset{\sim}{\boldsymbol{y}} = \underset{\approx}{\boldsymbol{M}} \cdot \underset{\sim}{\boldsymbol{x}} + \underset{\sim}{\boldsymbol{b}}$ may be written more intuitively in the "point-slope" form,

$$\underset{\sim}{\boldsymbol{y}} - \underset{\sim}{\overline{\boldsymbol{y}}} = \underset{\approx}{\boldsymbol{M}} \cdot (\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\overline{\boldsymbol{x}}}) \tag{12}$$

4 Example

Let $\underset{\sim}{\boldsymbol{e}}_1$ and $\underset{\sim}{\boldsymbol{e}}_2$ denote base vectors in a 2D space. Consider a function $\underset{\sim}{\boldsymbol{y}} = \underset{\sim}{\boldsymbol{y}}(\underset{\sim}{\boldsymbol{x}})$ that takes a vector,

$$\underset{\sim}{\boldsymbol{x}} = x_1 \underset{\sim}{\boldsymbol{e}}_1 + \underset{\sim}{\boldsymbol{x}}_2 \underset{\sim}{\boldsymbol{e}}_2 \tag{13}$$

as input, and returns a vector

$$\underset{\sim}{\boldsymbol{y}} = y_1 \underset{\sim}{\boldsymbol{e}}_1 + y_2 \underset{\sim}{\boldsymbol{e}}_2 \tag{14}$$

as output. In particular, consider the mapping function that was used in the graphics of Figs. 1, 2, and 3:

$$\begin{aligned} y_1 &= \frac{1}{50}\left(2{x_1}^2 + 5x_1(3x_2 - 7) + 3(x_2 - 17)x_2 + 38\right) && (15) \\ y_2 &= \frac{1}{50}\left(-2{x_1}^2 - 5x_1(x_2 - 6) + 10\left({x_2}^2 - 2\right)\right) && (16) \end{aligned}$$

Carrying out the above steps for a uniform weighting function (viz. $w(\underset{\sim}{\boldsymbol{x}}) = 1$ or $w_i = 1$) gives the following results:

$$\begin{aligned} W &= 4.\bigstar 0 \bigstar 0 \bigstar 0 \bigstar 0 && (17) \\ \{\underset{\sim}{\overline{\boldsymbol{x}}}\} &= \{0.\bigstar 0 \bigstar, 0.0 \bigstar 0 \bigstar 0\} && (18) \\ \{\underset{\sim}{\overline{\boldsymbol{y}}}\} &= \{0.\bigstar 9 \bigstar 6 \bigstar 7, -0.\bigstar 4 \bigstar 3 \bigstar 3\} && (19) \\ \left[\underset{\approx}{\boldsymbol{g}}\right] &= \{\{-0.\bigstar 5 \bigstar 6 \bigstar 7, -0.\bigstar 7 \bigstar\}, \{0.\bigstar 2, 0\}\} && (20) \\ \left[\underset{\approx}{\boldsymbol{G}}\right] &= \{\{0.\bigstar 6 \bigstar 6 \bigstar 7, 0\}, \{0, 0.\bigstar 6 \bigstar 6 \bigstar 7\}\} && (21) \\ \left[\underset{\approx}{\boldsymbol{M}}\right] &= \{\{-0.\bigstar, -1.\bigstar 2\}, \{0.\bigstar, 0\}\} && (22) \end{aligned}$$

Here, some of the digits have been replaced with "★" to allow this problem to be given as a homework assignment in a university setting. Data files (x.dat, y.dat, and w.dat) containing the discrete $\underset{\sim}{\boldsymbol{x}}$, $\underset{\sim}{\boldsymbol{y}}$, and w values may be found in an online compressed archive file (2dAffineRegression.zip) at the University of Utah CSM website. The archive also contains a data file of the non-uniform weights (wstar.dat) that were used to generate the results in Fig. 3.