Two Dimensional Euclidean Regression

Yaakov (J) Stein

June 1983

Abstract

Linear regression is used to find the line that best fits a set of N data points $\{(x_n, y_n)\}_{n=1}^{N}$, where the criterion is minimization of the sum of the squares of the distances of the data points from the line. In the conventional scenario the x coordinates are considered to be precisely known, and the distance of a data point to the line is taken to be the *vertical* distance. A trivial swap of independent and dependent variables enables changing the criterion to the sum of the squares of the *horizontal* distances. In certain applications, for example mapping problems where we wish to approximate points on a map by a straight line, the natural criterion to be minimized is the Euclidean distance of the data points from the approximation line, that is the distance from the data point to the closest point on the line. This corresponds to the perpendicular distance from the line rather than vertical or horizontal, which can not be found using standard linear regression techniques. In this paper an explicit formula for the Euclidean regression line is presented.

1 Classical 'one dimensional' linear regression

In this paper we assume that we are given N two dimensional data points $\{(x_n, y_n)\}_{n=1}^{N}$ and are required to find a straight line that best fits these data points. If all the data points are truly colinear, the expression 'best fits' is uniquely defined, but when they are not, one must pick the meaning best suited for the application.

In the classical linear regression problem, we assume that we have samples of a function $y(x)$, where x is the independent variable. Being an independent variable, we take the values of the x_n to be under our control, and thus perfectly known. On the other hand, the y_n values are assumed to have been experimentally measured, and thus suffer from various noise effects, including measurement error, quantization error, modeling error, etc.

Although the data points (x, y) exist in two dimensional space, we call this problem 'one dimensional' as there is a single independent variable x. The true y values, being derived from x, lie along a one dimensional line, at some angle to the x axis.

The classical (one dimensional) linear regression problem is to find the line $\hat{y} = mx + b$ that best approximates the experimental data. It is natural to define the approximation error of a data point (x_n, y_n) to be the difference between the approximated y and the measured value $\varepsilon_n = \hat{y}_n - y_n = (mx_n + b) - y_n$, as depicted in Figure 1. The line that 'best

fits' the data is the one that minimizes the sum of the squares of these approximation errors. The slope m and y-intercept b are given in equations (1) and (2), where $< f >\equiv \frac{1}{N}\sum_{n=1}^{N} f_n$.

$$m = \frac{\sigma_{xy}}{\sigma_x^2} = \frac{< xy > - < x >< y >}{< x^2 > - < x >^2} \quad (1)$$

$$b = \langle y \rangle - m \langle x \rangle \quad (2)$$

Of course, this method completely breaks down when the line is precisely vertical, since then $x = k$ and the assumed functional form $y = mx + b$ doesn't hold. However, it also breaks down in practice when that line has large but finite slope. To see why, consider that although x is an independent variable, and thus theoretically perfectly determined, in practice it too has some experimental uncertainty. When in an experiment we set a real x value in order to recover the y, we can not force it to take exactly the value we want. Until now we have assumed that our ability to set x results in errors that are insignificant as compared with those resulting from the measurement process for y. But when the line is of large slope, ignoring small 'setting' errors effectively creates large 'measurement' errors, as can be seen in Figure 2. This destabilizes the process, causing the slope estimation to be highly sensitive to slight inaccuracies in x.

We could have avoided this situation were we to have exploited a different type of linear regression problem. In this type of problem the roles of x and y are swapped, making y the independent variable and x the dependent one. The line is described by $x = m'y + b'$, where m' and b' are given by equations (3) and (4).

$$m' = \frac{\sigma_{xy}}{\sigma_y^2} = \frac{< xy > - < x >< y >}{< y^2 > - < y >^2} \quad (3)$$

$$b' = \langle x \rangle - m' \langle y \rangle \quad (4)$$

Of course, when m' is small it is still true that small changes in m' result in large changes in $m \sim \frac{1}{m'}$; but the solution is more stable as the error criterion depends on errors in x, as can be seen in Figure 3. If the line is not truly vertical, and we wish to compare the slope thus obtained to the previous one, we compute the reciprocal of m'.

$$m = \frac{\sigma_{xy}}{\sigma_x^2} \qquad m'' = \frac{\sigma_y^2}{\sigma_{xy}}$$

2 Two dimensional Euclidean regression

In various applications, including mapping problems, one wishes to find the line that best approximates a two-dimensional set of data points, where the x and y values are on an equal standing, that is, neither x nor y are independent variables upon which the other depends. For example, we may have a set of points on a map that represent a straight road or border, and wish to draw that road or border in a single line-drawing operation. Since the x and y values have been obtained by digitizing aerial photographs or by collecting surveying data, they have experimental inaccuracies that would result in a jagged line were the individual points connected. Also, the line representation is a more compact one for storing in a mapping database, and a necessary one for automatic analysis of the map.

The most common approach to the two dimensional problem has been to exploit one dimensional linear regression. Assuming a mapping problem with x axis representing the west-east direction and the y axis representing the south-north direction, according to this approach one estimates a line $y = mx + b$ that best approximates the data points under the assumption that the west-east coordinate is without error, while the south-north coordinate is contaminated.

This approach completely breaks down for roads or borders running from south to north, and is unreliable for roads or borders close to that direction.

The common remedy is to check the m obtained from the linear regression. If it exceeds 1 in absolute value (i.e. is between 45 and 135 degrees), the linear regression parameters are recomputed under the assumption that the errors are in the west-east coordinate. In other words, instead of m we compute m''. Assuming that we have precomputed all the sums needed for determination of $< x >$, $< y >$, $< xy >$, $< x^2 >$, and $< y^2 >$, this new linear regression does not require returning to the data points, but only recombination of statistics already computed. Hence this operation can be performed after the fact and is not overly expensive from the computation point of view.

Although this technique is straightforward, it is inelegant and does not really capture the nature of the problem at hand. In the present case neither x nor y have special status, and the natural error criterion is the distance of data point from the approximation line, that is the Euclidean distance from the data point to the nearest point on the line. This corresponds to an error perpendicular to the approximation line, as depicted in Figure 4. Although it is conventional in such cases to represent the line in an axis neutral fashion $Ax + By + C = 0$, we will continue to use the usual form $y = mx + b$. When using this representation, the distance from the data point (x_n, y_n) and the line $y = mx + b$ is given by equation (5).

$$\varepsilon_n = \left| \frac{y_n - (mx_n + b)}{\sqrt{m^2 + 1}} \right| \tag{5}$$

The line which minimizes the sum of the squares of *these* errors ε_n is the two dimensional Euclidean regression line.

By a least squares calculation outlined in the appendix, we can derive the formula for the parameters of the Euclidean regression line. Representing the line being sought by $y = \hat{m}x + \hat{b}$, we find that

$$\hat{m} = \frac{-R \pm \sqrt{R^2 + 4}}{2} \tag{6}$$

where we have defined

$$R = \frac{\sigma_x^2 - \sigma_y^2}{\sigma_{xy}} \tag{7}$$

and $\hat{b}$ is defined analogously to equation 2.

Calling the two possible slopes $\hat{m}_\pm$, from the expression in equation 5 we find that

$$\hat{m}_+ \, \hat{m}_- = \frac{\left(-R + \sqrt{R^2 + 4}\right)\left(-R - \sqrt{R^2 + 4}\right)}{4} = -1$$

demonstrating that the solutions are perpendicular to one another, from which it is easy to see that one represents minimum error and the other maximum error. In general it is easy

to select the slope that minimizes the error, and it can be shown that the correct solution is the one for which the sign of $\hat{m}$ matches that of σ_{xy}.

3 Simulation and Verification

We have performed a simulation to compare the results of two dimensional Euclidean regression to standard linear regression. In this simulation, for each angle from 1 to 179 degrees (with the exception of 90°) we generated 10,000 different line segments with the corresponding slope but differing in starting point. All line segments were chosen to have the same length (independent of slope). We uniformly sampled these line segments to produce 50 clean data points, and then randomly displaced each of these points by adding independently distributed Gaussian noise to its x and y values. The standard deviation in both x and y directions was taken to be 2 percent of the length of the entire line segment.

For each of the 10,000 sets of noisy data points we estimated the slope and y-intercept using both our method, and the conventional method of performing linear regression with selection of dependent and independent variables based on the angle exceeding 45°. We declared the Euclidean regression to be superior if the slope it predicted was closer to the true value than for that estimated by the conventional method. We then tallied the number of superior instantiations out of the 10,000 tested, and plotted this number versus the angle in Figure 5. As can be seen from the figure, the two dimensional Euclidean regression is clearly superior to the conventional method except for lines very close to the vertical or horizontal.

We have also used Euclidean regression in several real mapping problems. In these applications we need to couple the technique with a segmentation algorithm that terminates line segments at points where the slope changes significantly, and automatic removal of outliers. In the situations we have encountered there are typically a few tens of data points per segment, and the noise on the points is circularly distributed. Once the line segments are computed they are concatenated and overlaid as a graphic representation of the feature. We have verified by visual inspection that Euclidean regression produces a representation of significantly improved quality.

Appendix: Derivation of equation 5

The derivation proceeds along familiar lines. Using equation (5), the total error to be minimized is given by

$$\begin{aligned}\sum_{n=1}^{N} \varepsilon_n^2 &= \sum_{n=1}^{N} \frac{[y_n - (\hat{m}x_n + \hat{b})]^2}{\hat{m}^2 + 1} \\ &= \frac{1}{\hat{m}^2 + 1}\left[\sum_n y_n^2 + \hat{m}^2 \sum_n x_n^2 + N\hat{b}^2 + 2\hat{m}\hat{b}\sum_n x_n - 2\hat{m}\sum_n x_n y_n - 2\hat{b}\sum_n y_n \right]\end{aligned}$$

Differentiating according to b we find that

$$0 \;=\; \frac{d}{d\hat{b}}\sum_{n=1}^{N}\varepsilon_n^2 \;=\; \frac{1}{\hat{m}^2+1}\left(2\hat{m}\sum_n x_n \;-\; 2\sum_n y_n \;+\; 2N\hat{b}\right)$$

from which we easily deduce

$$\hat{b} \;=\; \frac{2\sum y - \;2\hat{m}\sum x}{2N} \;=\; < y > - \hat{m} < x >$$

as expected.

The differentiation by $\hat{m}$ is harder. Calling the contents of the square brackets Z,

$$0 \;=\; \frac{d}{d\hat{m}}\sum_{n=1}^{N}\varepsilon_n^2 \;=\; \frac{d}{d\hat{m}}\left(\frac{Z}{\hat{m}^2+1}\right) \;=\; \frac{\left(2\hat{m}Z - (\hat{m}^2+1)\frac{d}{d\hat{m}}Z\right)}{(\hat{m}^2+1)^2}$$

and for this to be zero, its numerator must be zero.

$$0 \;=\; 2\hat{m}Z - (\hat{m}^2+1)Z'$$

The required derivative of Z is

$$Z' \;=\; 2\hat{m}\sum x_n^2 - 2\sum x_n y_n + 2\hat{b}\sum x_n$$

and after collecting terms and dividing by N, we obtain the following expression.

$$2(\hat{m}^2-1)\left(b < x > - < xy >\right) + 2\hat{m}\left(\hat{b}^2 + < y > - < x^2 > -2\hat{b} < y >\right) \;=\; 0$$

Substituting $b = < y > - m < x >$, we get a quadratic equation for $\hat{m}$.

$$\begin{array}{rl} [< x >< y > - < xy >] & \hat{m}^2 + \\ \left[(< x >^2 - < x^2 >) - (< y >^2 - < y^2 >)\right] & \hat{m} - \\ [< x >< y > - < xy >] & = \;0 \end{array}$$

Before solving, it is useful to define

$$R \equiv \frac{(< x >^2 - < x^2 >) - (< y >^2 - < y^2 >)}{< x >< y > - < x >< y >}$$

and to rewrite the quadratic equation,

$$\hat{m}^2 + Rm - 1 \;=\; 0$$

from which equation 6 follows.

Figures

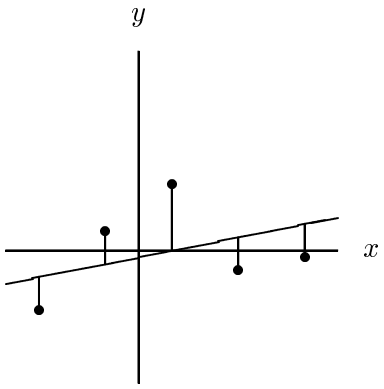

Figure 1: The classical linear regression problem. Here the approximation error is the vertical distance of the data point from the approximation line.

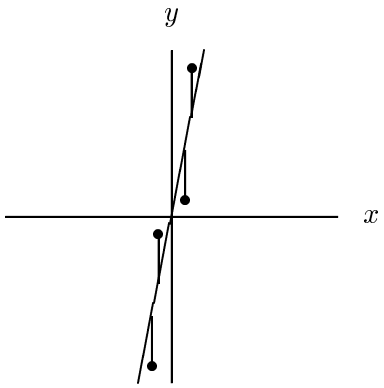

Figure 2: Classical linear regression when the slope is large. Small inaccuracies in x values are interpreted as large errors in y, making it difficult to recover m.

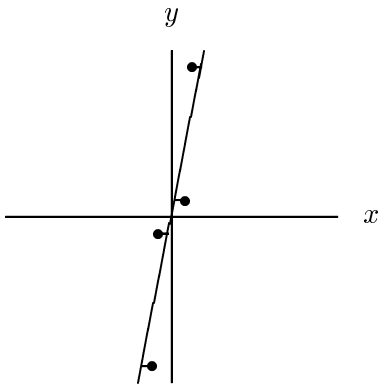

Figure 3: Linear regression with y as the independent variable. The data is the same as in the previous figure, but here the approximation error is the vertical distance of the data point from the approximation line.

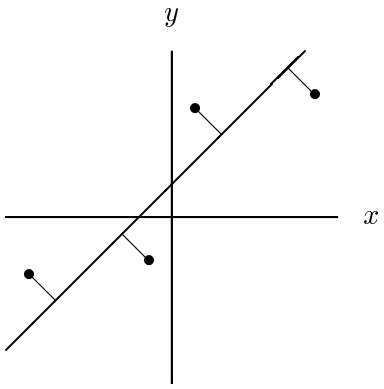

Figure 4: Euclidean regression. Here the approximation error is distance of the data point from the approximation line, i.e. the distance to the closest point on that line.

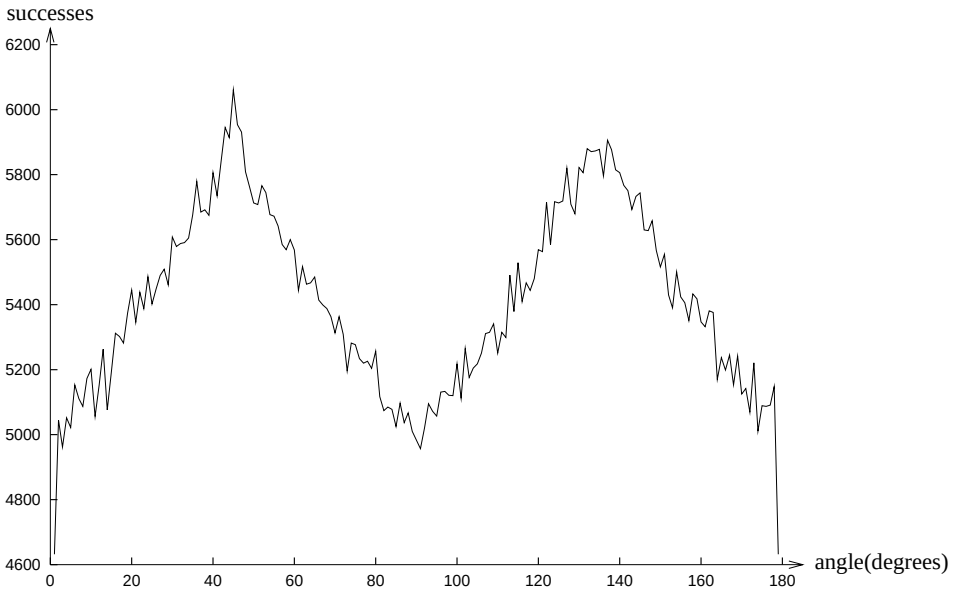

Figure 5: Simulation results.