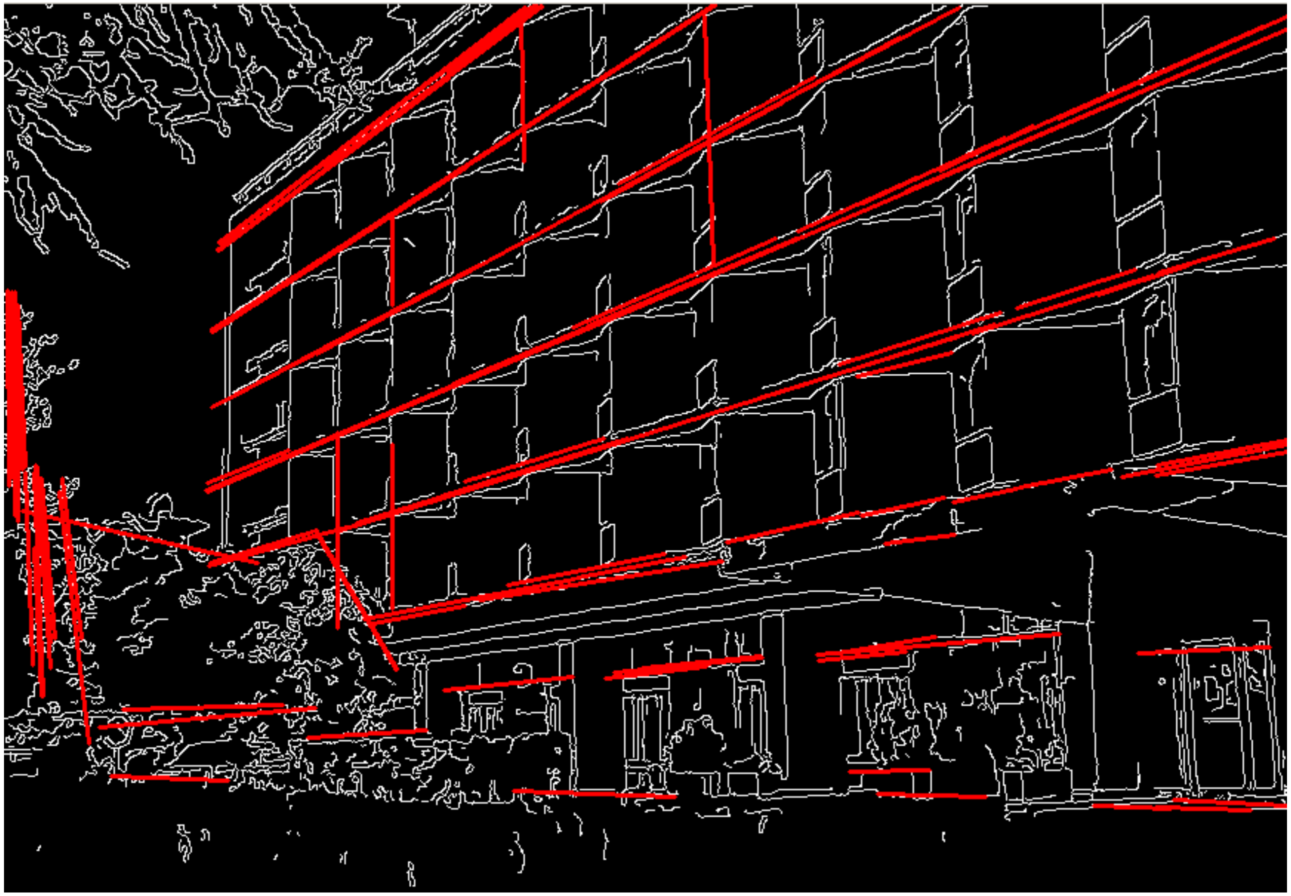
Fitting a line

D.A. Forsyth

University of Illinois at Urbana Champaign

Fitting

- Form a higher level, more compact representation of the image features
- Model case:
  - fit a line



Fitting: Challenges

Case study: Line detection

- **Noise** in the measured feature locations
- **Extraneous data:** clutter (outliers), multiple lines
- **Missing data:** occlusions

DON’T DO THIS!

There is a simple strategy for fitting lines, known as *least squares*. *Do not use this* - I am showing it only because it is traditional, and because the form of the derivation is repeated for other problems. This procedure has a long tradition and a substantial bias. You have likely seen this idea, but may not know why you should not use it. Represent a line as $y = ax + b$. There are N tokens, and the i'th token is at $\mathbf{x}_i = (x_{1,i}, x_{2,i})$. Now choose the line that best predicts the measured y coordinate for each measured x coordinate, and so minimizes

Really, just DON'T

should not use it. Represent a line as $y = ax + b$. There are N tokens, and the i'th token is at $\mathbf{x}_i = (x_{1,i}, x_{2,i})$. Now choose the line that best predicts the measured y coordinate for each measured x coordinate, and so minimizes

$$(1/2)\sum_i (x_{2,i} - ax_{1,i} - b)^2.$$

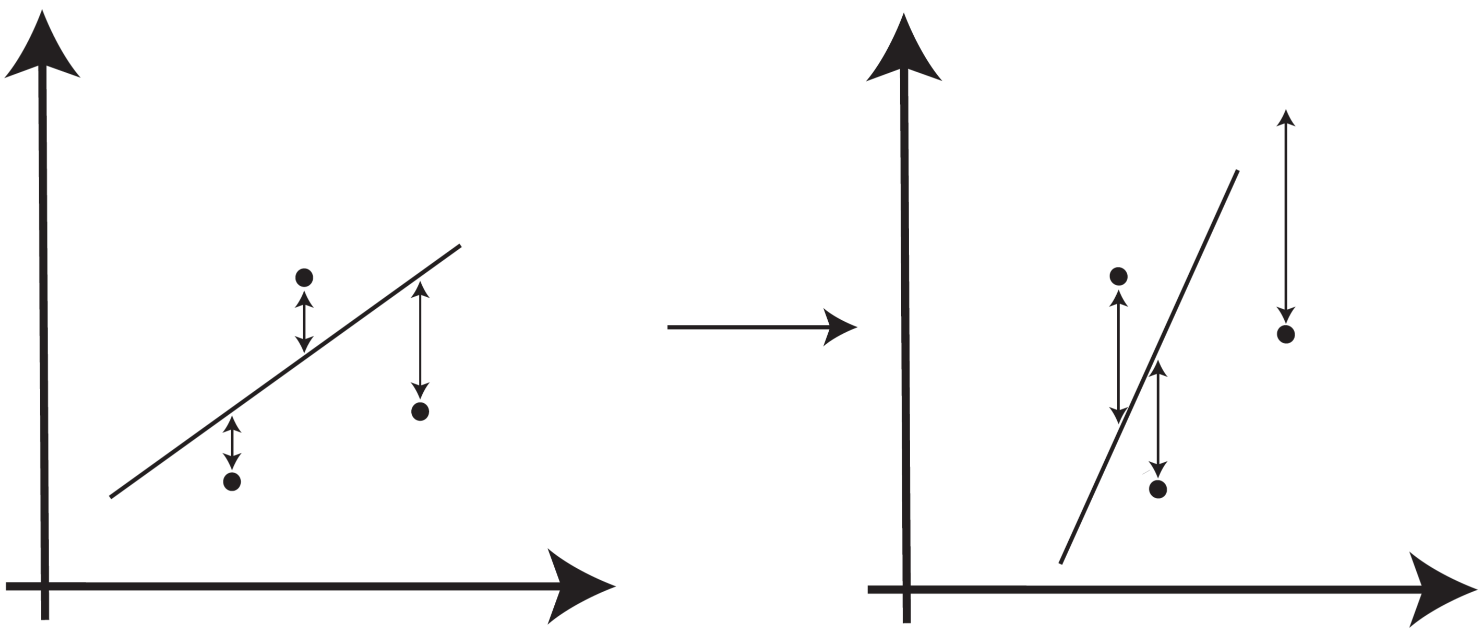
The term $(1/2)$ very often appears in quadratic problems like this to absorb the 2 that appears when you differentiate. A certain looseness in notation is traditional (you might not see either), because the 2 appears in front of an expression that is equal to zero. Adopting the notation

$$\overline{u} = \frac{\sum u_i}{N}$$

the line is given by the solution to the problem

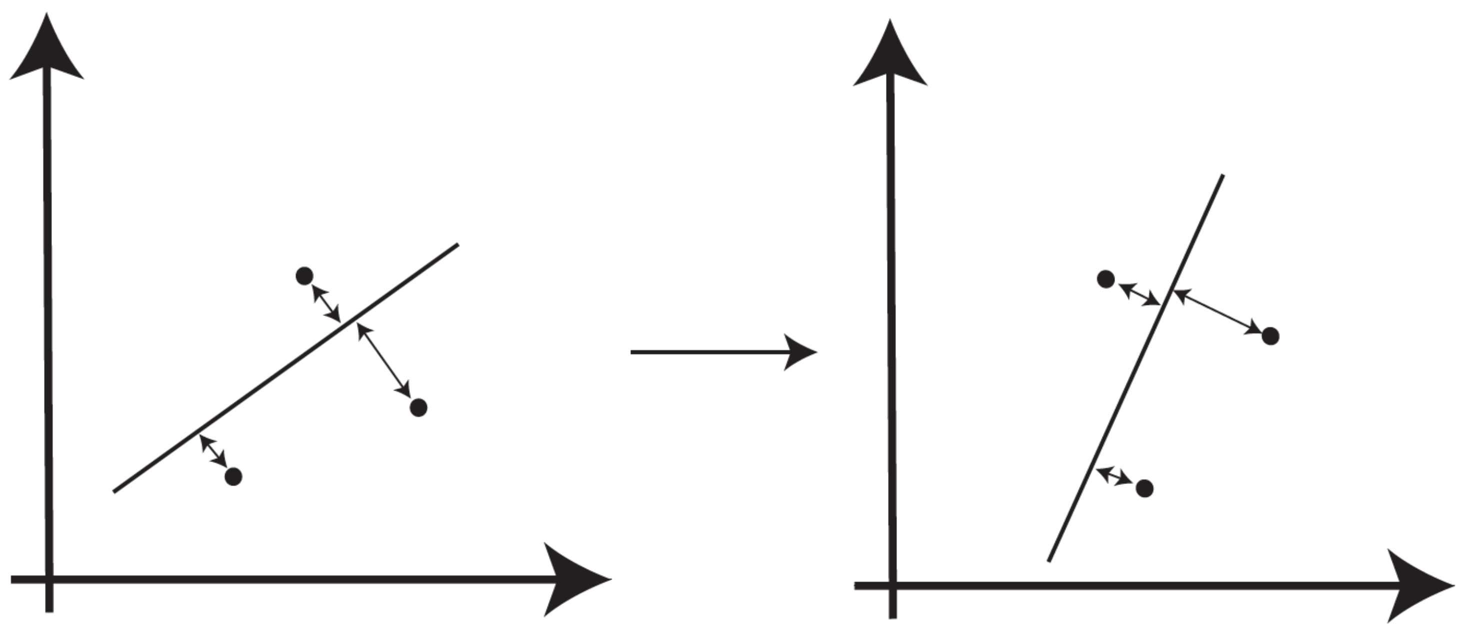
$$\begin{pmatrix} \overline{x_2^2} \\ \overline{x_2} \end{pmatrix} = \begin{pmatrix} \overline{x_1^2} & \overline{x_1} \\ \overline{x_1} & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}.$$

Because



Least squares error

Total least squares



Total least squares

Total least squares

Working with the actual distance between the token and the line (rather than the vertical distance) leads to a problem known as *total least squares*. Write $\mathbf{a} = (a_1, a_2)^T$ and represent a line as the collection of points where $\mathbf{a}^T\mathbf{x} + c = 0$. Require that $\mathbf{a}^T\mathbf{a} + c^2 \neq 0$, otherwise the equation is true for all points on the plane. Every line can be represented in this way.

Recall that for $s \neq 0$, the line given by $s(\mathbf{a}, c)$ is the same as the line represented by $(\mathbf{a}, c)$. Recall that the perpendicular distance from a point $\mathbf{x}$ to a line $(\mathbf{a}, c)$ is given by

$$\text{abs}(\mathbf{a}^T\mathbf{x} + c) \qquad \text{if} \qquad \mathbf{a}^T\mathbf{a} = 1.$$

It is convenient to assume that each point has an associated weight w_i. In the case where you have no idea what the weights are, use $w_i = 1/N$. Notice that choosing $w_i = 1/N$ ensures that the fitting cost you compute does not depend on the number of points **exercises** . Recall

$$\overline{u} = \frac{\sum_i w_i u_i}{\sum_i w_i}$$

Minimizing the sum of perpendicular distances between points and lines requires minimizing

$$(1/2)\sum_i w_i(\mathbf{a}^T\mathbf{x}_i + c)^2,$$

subject to $\mathbf{a}^T\mathbf{a} = 1$. Introduce a Lagrange multiplier λ; at a solution

Total least squares

subject to $\mathbf{a}^T\mathbf{a} = 1$. Introduce a Lagrange multiplier λ; at a solution

$$\begin{pmatrix} \overline{x_1^2} & \overline{x_1 x_2} & \overline{x_1} \\ \overline{x_1 x_2} & \overline{x_2^2} & \overline{x_2} \\ \overline{x_1} & \overline{x_2} & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ c \end{pmatrix} = \lambda \begin{pmatrix} 2a_1 \\ 2a_2 \\ 0 \end{pmatrix}.$$

This means that

$$c = -a_1\overline{x_1} - a_2\overline{x_2}.$$

Substitute this back to get the eigenvalue problem

$$\begin{pmatrix} \overline{x_1^2} - \overline{x_1}\;\overline{x_1} & \overline{x_1 x_2} - \overline{x_1}\;\overline{x_2} \\ \overline{x_1 x_2} - \overline{x_1}\;\overline{x_2} & \overline{x_2^2} - \overline{x_2}\;\overline{x_2} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \mu \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}.$$

Because this is a 2D eigenvalue problem, two solutions up to scale can be obtained in closed form (for those who care, it's usually done numerically!). The scale is obtained from the constraint that $\mathbf{a}^T\mathbf{a} = 1$. The two solutions to this problem are lines at right angles; one maximizes the sum of squared distances and the other minimizes it (which is which is explored in exercises **exercises**).

Fitting curved shapes is hard

Points close to line -> TLS behaves well

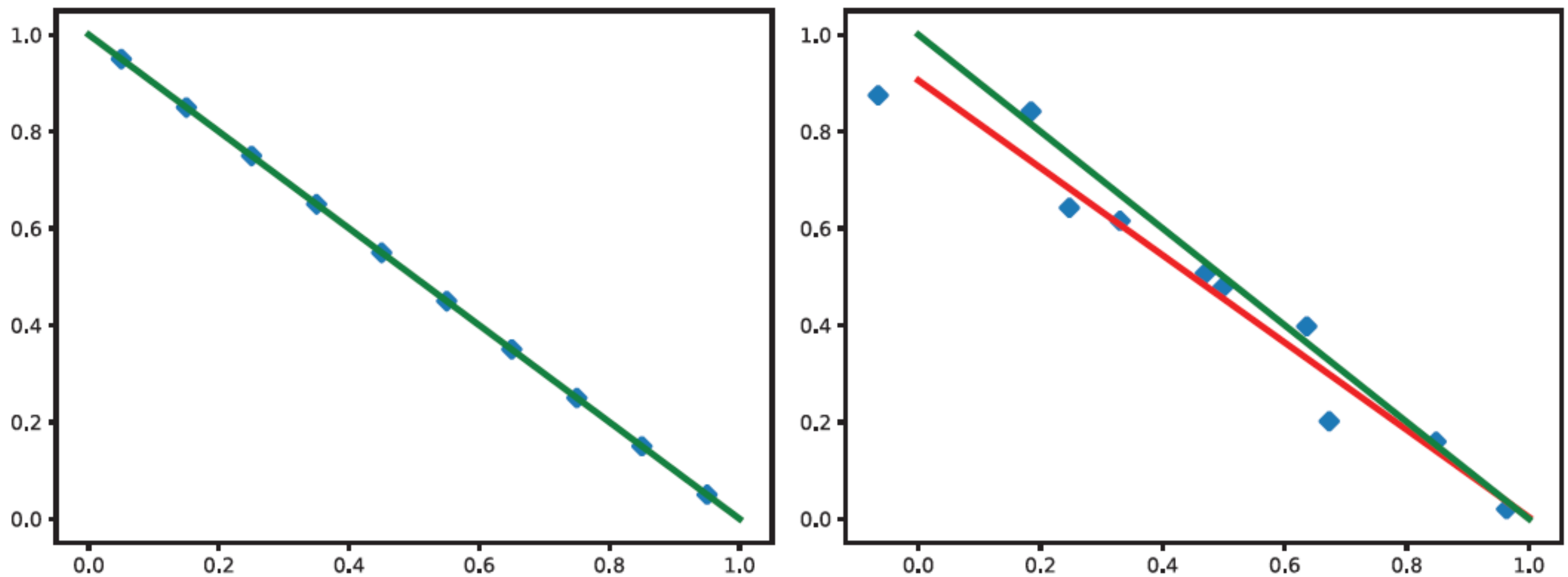


FIGURE 12.4: *Total least squares behaves well when all points lie fairly close to the line.* **Left**: *A line fitted with total least squares to a set of points that actually lie on the line.* **Right**: *The points have been perturbed by adding a small random term to both x and y coordinate, but the fitted line (red) is not that different from the original (green).*

Large errors – TLS behaves badly

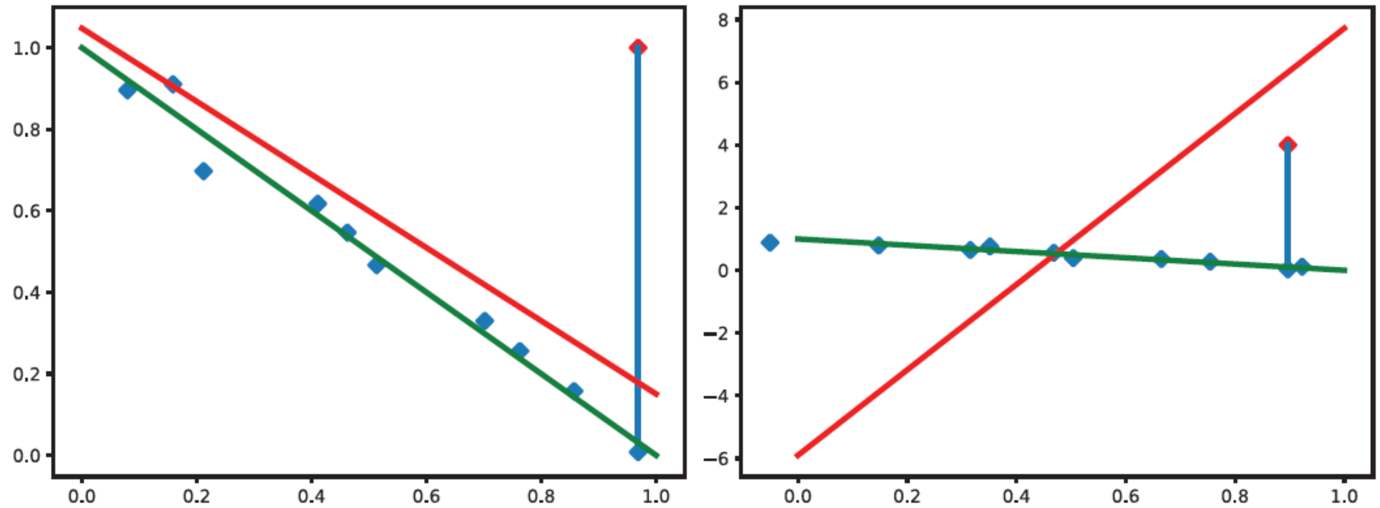


FIGURE 12.5: *Total least squares can misbehave when some points are significantly perturbed. Here the points are those of Figure 12.4, but one point has had its y coordinate replaced with a constant (new point in red, joined to its original position with a line).* **Left:** *The constant is 1, and the line produced by total least squares has shifted significantly.* **Right:** *The constant is 4, and now the fitted line is completely wrong. In each case, the original line is in green (and passes close to the points), and the new line is red.*

Things to think about...

14.1. Why does the family of lines given by $(u, v, 1)$ omit any line through the origin?

14.2. A plane is given by $\mathbf{a}^T\mathbf{x} + d = 0$; show the squared distance from a point $\mathbf{x}_i = (x_{1,i}, x_{2,i}, x_{i,3})$ to the plane is $(\mathbf{a}^T\mathbf{x} + d)^2$ if $\mathbf{a}^T\mathbf{a} = 1$.