Linear Regression

Chris Cornwell

September 4, 2025

Outline

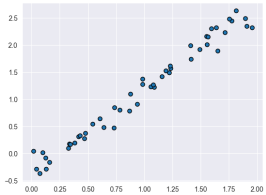
Outline

The goal

- Setting: have points in the plane, P of them. Say the points are $(x_1, y_1), (x_2, y_2), \ldots, (x_P, y_P)$.
- **Goal:** *Model* them as "approximately" coming from a line (or, being "noisy" samples from line), finding "best fit" line. This line is also called the **least squares regression** (LSR) line.

The goal

- Setting: have points in the plane, P of them. Say the points are $(x_1, y_1), (x_2, y_2), \ldots, (x_P, y_P)$.
- **Goal:** *Model* them as "approximately" coming from a line (or, being "noisy" samples from line), finding "best fit" line. This line is also called the **least squares regression** (LSR) line.
- **Running example:** A simulated data set, `'Example1.csv'`, with $P = 50$ points, is available here; these points are displayed in the plot below.



Outline

Finding the LSR line

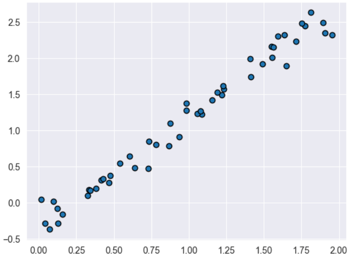


Figure 1: Our running example

How do we find the LSR line?

Finding the LSR line

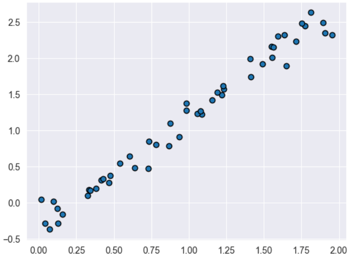


Figure 1: Our running example

How do we find the LSR line?

NumPy will do this: if x, y are the arrays containing the *x*- and *y*-coordinates, the slope and intercept for LSR line are given by:

```
np.polyfit(x,y,1)
```

Finding the LSR line

But, how? What is procedure to find the slope, intercept?

[1] Will use $\tilde{\mathbf{w}}$ for this vector, for the rest of these slides.

Finding the LSR line

But, how? What is procedure to find the slope, intercept?

- If a slope w and intercept b existed so that $(x_1, y_1), \ldots, (x_{50}, y_{50})$ *were* points on $y = wx + b$, then

  $$y_i = wx_i + b$$

  would hold for all $1 \leq i \leq 50$.

[1] Will use $\tilde{\mathbf{w}}$ for this vector, for the rest of these slides.

But, how? What is procedure to find the slope, intercept?

- If a slope w and intercept b existed so that $(x_1, y_1), \ldots, (x_{50}, y_{50})$ *were* points on $y = wx + b$, then

$$y_i = wx_i + b$$

would hold for all $1 \leq i \leq 50$.

1. Write those 50 equations as a matrix equation. Setting:

$$A = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_{50} \end{bmatrix}; \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{50} \end{bmatrix},$$

and writing[1] $\tilde{\mathbf{w}} = \begin{bmatrix} b \\ w \end{bmatrix}$, the matrix equation is $A\tilde{\mathbf{w}} = \mathbf{y}$.

[1] Will use $\tilde{\mathbf{w}}$ for this vector, for the rest of these slides.

Finding the LSR line

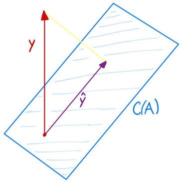
Now the equation $A\tilde{\mathbf{w}} = \mathbf{y}$ does not have a solution (those points are *not* on a line).

Finding the LSR line

Now the equation $A\tilde{\mathbf{w}} = \mathbf{y}$ does not have a solution (those points are *not* on a line).

Considering points as approx. on a line: (*noise in y-coordinate direction*)

- Find a $\hat{\mathbf{y}}$ *as close to* $\mathbf{y}$ *as possible* so that $A\tilde{\mathbf{w}} = \hat{\mathbf{y}}$ has a solution. For each i, making a (hopefully small) change $y_i \rightsquigarrow \hat{y}_i$ to get $\hat{\mathbf{y}}$.



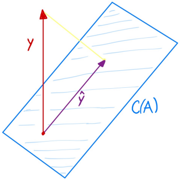
Finding the LSR line

Now the equation $A\tilde{\mathbf{w}} = \mathbf{y}$ does not have a solution (those points are *not* on a line).

Considering points as approx. on a line: (*noise in y-coordinate direction*)

- Find a $\hat{\mathbf{y}}$ *as close to* $\mathbf{y}$ *as possible* so that $A\tilde{\mathbf{w}} = \hat{\mathbf{y}}$ has a solution. For each i, making a (hopefully small) change $y_i \rightsquigarrow \hat{y}_i$ to get $\hat{\mathbf{y}}$.

2. Done by solving $A^T A\tilde{\mathbf{w}} = A^T\mathbf{y}$ (normal equations). If solution is $\tilde{\mathbf{w}}^\star = \begin{bmatrix} b^\star \\ w^\star \end{bmatrix}$, then $\hat{\mathbf{y}} = A\tilde{\mathbf{w}}^\star$.



Normal equation

Why does solving $A^T A\tilde{\mathbf{w}} = A^T\mathbf{y}$ give the right thing?

Normal equation

Why does solving $A^T A\tilde{\mathbf{w}} = A^T\mathbf{y}$ give the right thing?

Related to orthogonal vectors in $\mathbb{R}^P$ (in the example, $\mathbb{R}^{50}$).

- $\mathbf{y} = \mathbf{z}_1 + \mathbf{z}_2$, for some $\mathbf{z}_1$ in null space of A^T, $\mathbf{z}_2$ in column space of A. (*Note: Null space of A^T orthogonal to column space of A.*)

Normal equation

Why does solving $A^T A\tilde{\mathbf{w}} = A^T\mathbf{y}$ give the right thing?

Related to orthogonal vectors in $\mathbb{R}^P$ (in the example, $\mathbb{R}^{50}$).

- $\mathbf{y} = \mathbf{z}_1 + \mathbf{z}_2$, for some $\mathbf{z}_1$ in null space of A^T, $\mathbf{z}_2$ in column space of A. (*Note: Null space of A^T orthogonal to column space of A.*)
- $\mathbf{z}_2$ is closest to $\mathbf{y}$ (in col. space), since $\mathbf{z}_1$ is orthogonal to column space: $\mathbf{z}_2 = \hat{\mathbf{y}}$.

Normal equation

Why does solving $A^T A\tilde{\mathbf{w}} = A^T\mathbf{y}$ give the right thing?

Related to orthogonal vectors in $\mathbb{R}^P$ (in the example, $\mathbb{R}^{50}$).

- $\mathbf{y} = \mathbf{z}_1 + \mathbf{z}_2$, for some $\mathbf{z}_1$ in null space of A^T, $\mathbf{z}_2$ in column space of A. (*Note: Null space of A^T orthogonal to column space of A.*)
- $\mathbf{z}_2$ is closest to $\mathbf{y}$ (in col. space), since $\mathbf{z}_1$ is orthogonal to column space: $\mathbf{z}_2 = \hat{\mathbf{y}}$.
- As $\mathbf{z}_2$ in column space, $\exists\ \tilde{\mathbf{w}}^\star$ so that $A\tilde{\mathbf{w}}^\star = \mathbf{z}_2 = \hat{\mathbf{y}}$.

Normal equation

Why does solving $A^T A\tilde{\mathbf{w}} = A^T\mathbf{y}$ give the right thing?

Related to orthogonal vectors in $\mathbb{R}^P$ (in the example, $\mathbb{R}^{50}$).

- $\mathbf{y} = \mathbf{z}_1 + \mathbf{z}_2$, for some $\mathbf{z}_1$ in null space of A^T, $\mathbf{z}_2$ in column space of A. (*Note: Null space of A^T orthogonal to column space of A.*)
- $\mathbf{z}_2$ is closest to $\mathbf{y}$ (in col. space), since $\mathbf{z}_1$ is orthogonal to column space: $\mathbf{z}_2 = \hat{\mathbf{y}}$.
- As $\mathbf{z}_2$ in column space, $\exists\ \tilde{\mathbf{w}}^\star$ so that $A\tilde{\mathbf{w}}^\star = \mathbf{z}_2 = \hat{\mathbf{y}}$. But then,

$$A^T(A\tilde{\mathbf{w}}^\star) = A^T\mathbf{z}_2 = A^T(\mathbf{y} - \mathbf{z}_1) = A^T\mathbf{y}.$$

So the $\tilde{\mathbf{w}}^\star$ that you want must be a solution to the normal equations.

Normal equations:

$$(A^T A)\tilde{\mathbf{w}} = A^T \mathbf{y}.$$

Finding the LSR line

Normal equations:

$$(A^T A)\tilde{\mathbf{w}} = A^T \mathbf{y}.$$

Note:

- $A^T A$ is 2×2 matrix, $A^T \mathbf{y} \in \mathbb{R}^2$, and $A^T A$ is invertible as long as there exists $x_i \neq x_j$.

Finding the LSR line

Normal equations:

$$(A^T A)\tilde{\mathbf{w}} = A^T \mathbf{y}.$$

Note:

- $A^T A$ is 2×2 matrix, $A^T \mathbf{y} \in \mathbb{R}^2$, and $A^T A$ is invertible as long as there exists $x_i \neq x_j$.

3. Pretty quick to find solution to $(A^T A)\tilde{\mathbf{w}} = A^T \mathbf{y}$.

Finding the LSR line

Normal equations:

$$(A^T A)\tilde{\mathbf{w}} = A^T \mathbf{y}.$$

Note:

- $A^T A$ is 2×2 matrix, $A^T \mathbf{y} \in \mathbb{R}^2$, and $A^T A$ is invertible as long as there exists $x_i \neq x_j$.

3. Pretty quick to find solution to $(A^T A)\tilde{\mathbf{w}} = A^T \mathbf{y}$.

So, three steps:

1. Write the P equations in matrix form. (get matrix A, vector $\mathbf{y}$)
2. Get matrix $A^T A$ and vector $A^T \mathbf{y}$ for normal equations.
3. Use a method to solve normal equations for $\tilde{\mathbf{w}}$.

Outline

Solving normal equation, in pseudocode

Procedure to carry out the steps:

1. Write the P equations in matrix form. (get matrix A, vector $\mathbf{y}$)
2. Get matrix $A^T A$ and vector $A^T \mathbf{y}$ for normal equations.
3. Use a method to solve normal equations for $\tilde{\mathbf{w}}$.

Solving normal equation, in pseudocode

Procedure to carry out the steps:

1. Write the P equations in matrix form. (get matrix A, vector $\mathbf{y}$)
2. Get matrix $A^T A$ and vector $A^T \mathbf{y}$ for normal equations.
3. Use a method to solve normal equations for $\tilde{\mathbf{w}}$.

Given $(x_1, y_1), (x_2, y_2), \ldots, (x_P, y_P)$, as a NumPy array (call it `D`, with shape `(P,2)`):

```
A ← [ [1, x1], ..., [1, xP] ] # 2-column matrix
y ← y—coordinate array
# next, get 2x2 matrix and 2-vector
Compute A.T times A; compute A.T times y
Solve normal eq'ns (e.g., using inverse)
return solution
```

Outline

Result on running example

For the data (linked to above) with 50 points, the LSR line comes out close to

$$y = 1.520275x - 0.33458.$$

($b^\star = -0.33458$ and $w^\star = 1.520275$)

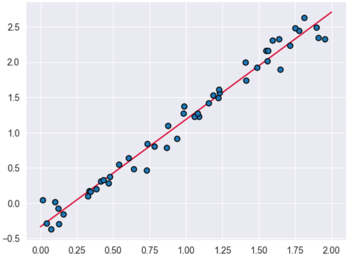
Result on running example

For the data (linked to above) with 50 points, the LSR line comes out close to

$$y = 1.520275x - 0.33458.$$

($b^\star = -0.33458$ and $w^\star = 1.520275$)

A plot of the line (in red), alongside the points, looks as follows.



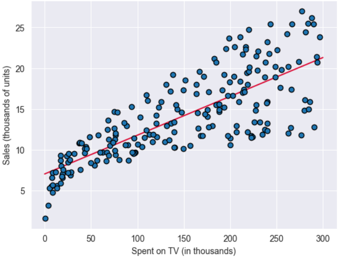
Another example, Advertising data

In the DataSets folder, the `'Advertising.csv'` file contains data on amounts spent (in thousands of dollars) on TV, Radio, and Newspaper advertising in 200 different markets, as well as the amounts sold in each market (in thousands of units).

Another example, Advertising data

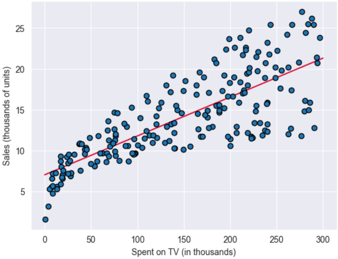
In the DataSets folder, the `'Advertising.csv'` file contains data on amounts spent (in thousands of dollars) on TV, Radio, and Newspaper advertising in 200 different markets, as well as the amounts sold in each market (in thousands of units).

We will look more at this data later. For now, plotted here are the columns (`'TV'`, `'Sales'`).



Another example, Advertising data

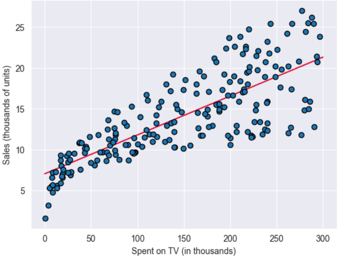
If you switch the role of x- and y-coordinates, you can still do linear regression; *i.e.*, for purpose of a thought experiment, predict the TV data as the *response*, instead of the Sales.



Another example, Advertising data

If you switch the role of x- and y-coordinates, you can still do linear regression; *i.e.*, for purpose of a thought experiment, predict the TV data as the *response*, instead of the Sales.

The LSR line for the data is then **not** the same line, if you switch the roles of TV and Sales in the algorithm to get $\tilde{\mathbf{w}}^\star$.



Another example, Advertising data

If you switch the role of x- and y-coordinates, you can still do linear regression; *i.e.*, for purpose of a thought experiment, predict the TV data as the *response*, instead of the Sales.

The LSR line for the data is then **not** the same line, if you switch the roles of TV and Sales in the algorithm to get $\tilde{\mathbf{w}}^\star$. (Use domain knowledge.)