Linear regression through Optimization

Chris Cornwell

September 11, 2025

Outline

Outline

Generalizing Linear regression to multiple variables

Measuring the fitness of linear models

Extending the linear regression procedure

If you have data $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_P, y_P)$ where, for some $N > 1$ the input $\mathbf{x}_i$ is a vector in $\mathbb{R}^N$, can you still do linear regression?

Extending the linear regression procedure

If you have data $\left(\mathbf{x}_1, y_1\right), \left(\mathbf{x}_2, y_2\right), \ldots, \left(\mathbf{x}_P, y_P\right)$ where, for some $N > 1$ the input $\mathbf{x}_i$ is a vector in $\mathbb{R}^N$, can you still do linear regression?

For each i with $1 \leq i \leq P$, say that $\mathbf{x}_i = \begin{bmatrix} x_{1,i} \\ x_{2,i} \\ \vdots \\ x_{N,i} \end{bmatrix}$. Define a $P \times \left(N + 1\right)$ matrix A so that the i^{th} row is $\left[1, x_{1,i}, x_{2,i}, \ldots, x_{N,i}\right]$.

Extending the linear regression procedure

If you have data $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_P, y_P)$ where, for some $N > 1$ the input $\mathbf{x}_i$ is a vector in $\mathbb{R}^N$, can you still do linear regression?

For each i with $1 \leq i \leq P$, say that $\mathbf{x}_i = \begin{bmatrix} x_{1,i} \\ x_{2,i} \\ \vdots \\ x_{N,i} \end{bmatrix}$. Define a $P \times (N+1)$ matrix A so that the i^{th} row is $[1, x_{1,i}, x_{2,i}, \ldots, x_{N,i}]$.

In other words, set $\tilde{\mathbf{x}}_i$ equal to the vector in $\mathbb{R}^{N+1}$ with 1 as its first component and $\mathbf{x}_i$ for the remaining N components. Then

$$A = \begin{bmatrix} — & \tilde{\mathbf{x}}_1^T & — \\ — & \tilde{\mathbf{x}}_2^T & — \\ & \vdots & \\ — & \tilde{\mathbf{x}}_P^T & — \end{bmatrix}.$$

Extending the linear regression procedure

A solution $\tilde{\mathbf{w}}^\star = [b^\star, w_1^\star, \ldots, w_N^\star]^T$ to the normal equations $A^T A \tilde{\mathbf{w}} = A^T \mathbf{y}$ (with A as above) gives coefficients for a linear model for the data.[1]

[1] The matrix $A^T A$ is $(N+1) \times (N+1)$ and $A^T \mathbf{y}$ is a vector in $\mathbb{R}^{N+1}$.

[2] Generalization of a plane in $\mathbb{R}^3$.

Extending the linear regression procedure

A solution $\tilde{\mathbf{w}}^\star = \left[b^\star, w_1^\star, \ldots, w_N^\star\right]^T$ to the normal equations $A^T A\tilde{\mathbf{w}} = A^T\mathbf{y}$ (with A as above) gives coefficients for a linear model for the data.[1]

Write $\mathbf{w}^\star \in \mathbb{R}^N$ for the non-constant coefficient vector $\left[w_1^\star, w_2^\star, \ldots, w_N^\star\right]^T$. The affine linear model on the data, in N variables, is

$$f_{\tilde{\mathbf{w}}^\star}(\mathbf{x}) = b^\star + w_1^\star x_1 + \ldots + w_N^\star x_N = b^\star + \mathbf{x}^T\mathbf{w}^\star.$$

[1]The matrix $A^T A$ is $(N+1) \times (N+1)$ and $A^T\mathbf{y}$ is a vector in $\mathbb{R}^{N+1}$.
[2]Generalization of a plane in $\mathbb{R}^3$.

Extending the linear regression procedure

A solution $\tilde{\mathbf{w}}^\star = \left[b^\star, w_1^\star, \ldots, w_N^\star\right]^T$ to the normal equations $A^T A\tilde{\mathbf{w}} = A^T\mathbf{y}$ (with A as above) gives coefficients for a linear model for the data.[1]

Write $\mathbf{w}^\star \in \mathbb{R}^N$ for the non-constant coefficient vector $\left[w_1^\star, w_2^\star, \ldots, w_N^\star\right]^T$. The affine linear model on the data, in N variables, is

$$f_{\tilde{\mathbf{w}}^\star}(\mathbf{x}) = b^\star + w_1^\star x_1 + \ldots + w_N^\star x_N = b^\star + \mathbf{x}^T\mathbf{w}^\star.$$

(1) $f_{\tilde{\mathbf{w}}^\star}(\mathbf{x})$ is affine linear, meaning the difference in two function values is a dot product on the difference of the input. Specifically, $f_{\tilde{\mathbf{w}}^\star}(\mathbf{x}) - f_{\tilde{\mathbf{w}}^\star}(\mathbf{x}') = (\mathbf{x} - \mathbf{x}')^T\mathbf{w}^\star$.

[1] The matrix A^TA is $(N+1)\times(N+1)$ and $A^T\mathbf{y}$ is a vector in $\mathbb{R}^{N+1}$.

[2] Generalization of a plane in $\mathbb{R}^3$.

Extending the linear regression procedure

A solution $\tilde{\mathbf{w}}^\star = \left[b^\star, w_1^\star, \ldots, w_N^\star\right]^T$ to the normal equations $A^T A\tilde{\mathbf{w}} = A^T\mathbf{y}$ (with A as above) gives coefficients for a linear model for the data.[1]

Write $\mathbf{w}^\star \in \mathbb{R}^N$ for the non-constant coefficient vector $\left[w_1^\star, w_2^\star, \ldots, w_N^\star\right]^T$. The affine linear model on the data, in N variables, is

$$f_{\tilde{\mathbf{w}}^\star}(\mathbf{x}) = b^\star + w_1^\star x_1 + \ldots + w_N^\star x_N = b^\star + \mathbf{x}^T\mathbf{w}^\star.$$

(1) $f_{\tilde{\mathbf{w}}^\star}(\mathbf{x})$ is affine linear, meaning the difference in two function values is a dot product on the difference of the input. Specifically, $f_{\tilde{\mathbf{w}}^\star}(\mathbf{x}) - f_{\tilde{\mathbf{w}}^\star}(\mathbf{x}') = (\mathbf{x} - \mathbf{x}')^T\mathbf{w}^\star$.

(2) The graph of $f_{\tilde{\mathbf{w}}^\star}(\mathbf{x})$ is a hyperplane[2] in $\mathbb{R}^{N+1}$ with normal vector $\left[1, -w_1^\star, \ldots, -w_N^\star\right]^T$.

[1] The matrix $A^T A$ is $(N+1) \times (N+1)$ and $A^T\mathbf{y}$ is a vector in $\mathbb{R}^{N+1}$.
[2] Generalization of a plane in $\mathbb{R}^3$.

Example

A concrete example: using the data in `'Advertising.csv'`, found in the DataSets folder of the course site.

This 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).

Example

A concrete example: using the data in `'Advertising.csv'`, found in the DataSets folder of the course site.
This 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).

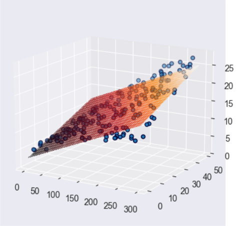
To perform linear regression, with independent variables `'TV'`, `'Radio'`, and `'Newspaper'` and setting y equal to `'Sales'`, one may follow the procedure above. The matrix A described is 200×4 ($N = 3$).

Example

A concrete example: using the data in `'Advertising.csv'`, found in the DataSets folder of the course site.

This 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).

To perform linear regression, with independent variables `'TV'`, `'Radio'`, and `'Newspaper'` and setting y equal to `'Sales'`, one may follow the procedure above. The matrix A described is 200×4 ($N = 3$). The resulting coefficients are (approximately)



3D projection of Advertising model

$$[b^\star, w_1^\star, w_2^\star, w_3^\star] = [2.9389, 0.0458, 0.1885, -0.001].$$

Outline

Generalizing Linear regression to multiple variables

Measuring the fitness of linear models

“Noise” and linear regression model

Underlying assumption: the (input) variables $x_1, \ldots, x_N$ are not random, but there is natural variability in the “y-direction” (value of $f_{\tilde{\mathbf{w}}}$), represented with a random variable.[3]

[3]Note that, if you instead consider the random variable to be in another direction (swapping some x_j with y) then there is a *different* linear regression.

"Noise" and linear regression model

Underlying assumption: the (input) variables $x_1, \ldots, x_N$ are not random, but there is natural variability in the "y-direction" (value of $f_{\tilde{\mathbf{w}}}$), represented with a random variable.[3]
So, there is a b and $\mathbf{w} = [w_1, \ldots, w_N]^T$, so that the relationship between $\mathbf{x}$ and y is

$$y = b + \mathbf{x}^T\mathbf{w} + \varepsilon,$$

where ε is a random variable with mean (first moment) equal to 0.

[3]Note that, if you instead consider the random variable to be in another direction (swapping some x_j with y) then there is a *different* linear regression.

“Noise” and linear regression model

Underlying assumption: the (input) variables $x_1, \ldots, x_N$ are not random, but there is natural variability in the “y-direction” (value of $f_{\tilde{\mathbf{w}}}$), represented with a random variable.[3]
So, there is a b and $\mathbf{w} = \left[w_1, \ldots, w_N\right]^T$, so that the relationship between $\mathbf{x}$ and y is

$$y = b + \mathbf{x}^T\mathbf{w} + \varepsilon,$$

where ε is a random variable with mean (first moment) equal to 0.

- Critically, the distribution of ε should be independent of $\mathbf{x}$.

[3]Note that, if you instead consider the random variable to be in another direction (swapping some x_j with y) then there is a *different* linear regression.

“Noise” and linear regression model

Underlying assumption: the (input) variables $x_1, \ldots, x_N$ are not random, but there is natural variability in the “y-direction” (value of $f_{\tilde{\mathbf{w}}}$), represented with a random variable.[3]

So, there is a b and $\mathbf{w} = \left[w_1, \ldots, w_N\right]^T$, so that the relationship between $\mathbf{x}$ and y is

$$y = b + \mathbf{x}^T\mathbf{w} + \varepsilon,$$

where ε is a random variable with mean (first moment) equal to 0.

- Critically, the distribution of ε should be independent of $\mathbf{x}$.
- A common consideration: ε normally distributed.

[3] Note that, if you instead consider the random variable to be in another direction (swapping some x_j with y) then there is a *different* linear regression.

Least Squares loss function

On given data $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_P, y_P)$, a common measure for how well a linear regression model fits is the **Mean Squared Error**. If the intercept and $\mathbf{w}$ vector for the regression line are $b^\star$ and $\mathbf{w}^\star$ then we have $\hat{y}_i = b^\star + \mathbf{x}_i^T \mathbf{w}^\star$ and the Mean Squared Error is

$$\frac{1}{P}\sum_{i=1}^{P}\left(\hat{y}_i - y_i\right)^2 = \frac{1}{P}\sum_{i=1}^{P}\left(b^\star + \mathbf{x}_i^T \mathbf{w}^\star - y_i\right)^2.$$

Least Squares loss function

On given data $\left(\mathbf{x}_1, y_1\right), \left(\mathbf{x}_2, y_2\right), \ldots, \left(\mathbf{x}_P, y_P\right)$, a common measure for how well a linear regression model fits is the **Mean Squared Error**. If the intercept and $\mathbf{w}$ vector for the regression line are $b^\star$ and $\mathbf{w}^\star$ then we have $\hat{y}_i = b^\star + \mathbf{x}_i^T \mathbf{w}^\star$ and the Mean Squared Error is

$$\frac{1}{P}\sum_{i=1}^{P}\left(\hat{y}_i - y_i\right)^2 = \frac{1}{P}\sum_{i=1}^{P}\left(b^\star + \mathbf{x}_i^T\mathbf{w}^\star - y_i\right)^2.$$

It is helpful to think of this as a function on an arbitrary choice of parameters $\tilde{\mathbf{w}} = \left(b, \mathbf{w}\right)$. Dropping the division by P, this is the **Least Squares loss** function:

Least Squares loss function

On given data $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_P, y_P)$, a common measure for how well a linear regression model fits is the **Mean Squared Error**. If the intercept and $\mathbf{w}$ vector for the regression line are $b^\star$ and $\mathbf{w}^\star$ then we have $\hat{y}_i = b^\star + \mathbf{x}_i^T \mathbf{w}^\star$ and the Mean Squared Error is

$$\frac{1}{P} \sum_{i=1}^{P} (\hat{y}_i - y_i)^2 = \frac{1}{P} \sum_{i=1}^{P} \left(b^\star + \mathbf{x}_i^T \mathbf{w}^\star - y_i\right)^2 .$$

It is helpful to think of this as a function on an arbitrary choice of parameters $\tilde{\mathbf{w}} = (b, \mathbf{w})$. Dropping the division by P, this is the **Least Squares loss** function: if $f_{\tilde{\mathbf{w}}}(\mathbf{x}) = b + \mathbf{x}^T \mathbf{w}$ is the linear model for parameters $\tilde{\mathbf{w}} = (b, w_1, \ldots, w_N)$ then this loss function is

$$g(\tilde{\mathbf{w}}) = \sum_{i=1}^{P} \left(f_{\tilde{\mathbf{w}}}(\mathbf{x}_i) - y_i\right)^2 .$$

Minimizing the Least Squares loss

For fixed data $\left\{\left(\mathbf{x}_i, y_i\right)\right\}_{i=1}^{P}$, how do we solve the problem

$$\underset{\tilde{\mathbf{w}}}{\text{minimize}}\ g\left(\tilde{\mathbf{w}}\right)?$$

Minimizing the Least Squares loss

For fixed data $\left\{\left(\mathbf{x}_i, y_i\right)\right\}_{i=1}^{P}$, how do we solve the problem

$$\underset{\tilde{\mathbf{w}}}{\text{minimize}}\; g\left(\tilde{\mathbf{w}}\right)?$$

We can use methods from calculus to determine the minimum (and the minimizer, to set equal to $\tilde{\mathbf{w}}$). Specifically, the first order condition is useful – that we want a value of $\tilde{\mathbf{w}}$ where all partial derivatives of $g\left(\tilde{\mathbf{w}}\right)$ are equal to zero.

Minimizing the Least Squares loss

For fixed data $\left\{\left(\mathbf{x}_i, y_i\right)\right\}_{i=1}^{P}$, how do we solve the problem

$$\underset{\tilde{\mathbf{w}}}{\text{minimize}}\; g\left(\tilde{\mathbf{w}}\right)?$$

We can use methods from calculus to determine the minimum (and the minimizer, to set equal to $\tilde{\mathbf{w}}$). Specifically, the first order condition is useful – that we want a value of $\tilde{\mathbf{w}}$ where all partial derivatives of $g\left(\tilde{\mathbf{w}}\right)$ are equal to zero.

- Note, what are the variables for the partial derivative? They are the parameters in $\tilde{\mathbf{w}}$, namely $b, w_1, w_2, \ldots, w_N$.

Minimizing the Least Squares loss

For fixed data $\left\{\left(\mathbf{x}_i, y_i\right)\right\}_{i=1}^{P}$, how do we solve the problem

$$\underset{\tilde{\mathbf{w}}}{\text{minimize}}\ g\left(\tilde{\mathbf{w}}\right)?$$

We can use methods from calculus to determine the minimum (and the minimizer, to set equal to $\tilde{\mathbf{w}}$). Specifically, the first order condition is useful – that we want a value of $\tilde{\mathbf{w}}$ where all partial derivatives of $g\left(\tilde{\mathbf{w}}\right)$ are equal to zero.

- Note, what are the variables for the partial derivative? They are the parameters in $\tilde{\mathbf{w}}$, namely $b, w_1, w_2, \ldots, w_N$.
- There will be the input (independent variables) $\mathbf{x}$ to the linear function; but g is a function of the *parameters*.

Minimizing the Least Squares loss

For fixed data $\left\{\left(\mathbf{x}_i, y_i\right)\right\}_{i=1}^{P}$, how do we solve the problem

$$\underset{\tilde{\mathbf{w}}}{\text{minimize}}\; g\left(\tilde{\mathbf{w}}\right)?$$

We can use methods from calculus to determine the minimum (and the minimizer, to set equal to $\tilde{\mathbf{w}}$). Specifically, the first order condition is useful – that we want a value of $\tilde{\mathbf{w}}$ where all partial derivatives of $g\left(\tilde{\mathbf{w}}\right)$ are equal to zero.

- Note, what are the variables for the partial derivative? They are the parameters in $\tilde{\mathbf{w}}$, namely $b, w_1, w_2, \ldots, w_N$.
- There will be the input (independent variables) $\mathbf{x}$ to the linear function; but g is a function of the *parameters*.

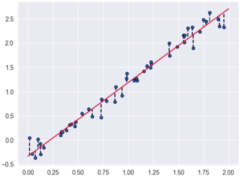
Next: a review of calculus minimization techniques.

Meaning of Least Squares loss

For $1 \le i \le P$, the quantity $\left|f_{b,\mathbf{w}}(\mathbf{x}_i) - y_i\right|$ is the vertical distance from $(\mathbf{x}_i, y_i)$ to the point predicted by the linear model, $(\mathbf{x}_i, f_{\tilde{\mathbf{w}}}(\mathbf{x}_i))$.

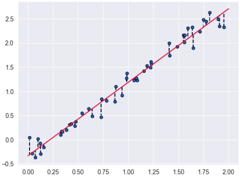
Meaning of Least Squares loss

For $1 \leq i \leq P$, the quantity $\left|f_{b,\mathbf{w}}(\mathbf{x}_i) - y_i\right|$ is the vertical distance from $(\mathbf{x}_i, y_i)$ to the point predicted by the linear model, $(\mathbf{x}_i, f_{\tilde{\mathbf{w}}}(\mathbf{x}_i))$.



Meaning of Least Squares loss

For $1 \leq i \leq P$, the quantity $\left|f_{b,\mathbf{w}}(\mathbf{x}_i) - y_i\right|$ is the vertical distance from $(\mathbf{x}_i, y_i)$ to the point predicted by the linear model, $(\mathbf{x}_i, f_{\tilde{\mathbf{w}}}(\mathbf{x}_i))$.

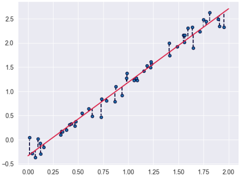


The length of the vector $\mathbf{y} - \hat{\mathbf{y}}$ (*i.e.*, the distance from $\mathbf{y}$ to the vector of predictions, in the column space of our feature matrix) is equal to

$$\sqrt{(f_{\tilde{\mathbf{w}}}(\mathbf{x}_1) - y_1)^2 + (f_{\tilde{\mathbf{w}}}(\mathbf{x}_2) - y_2)^2 + \ldots + (f_{\tilde{\mathbf{w}}}(\mathbf{x}_P) - y_P)^2} = \sqrt{g(\tilde{\mathbf{w}})}.$$

Meaning of Least Squares loss

For $1 \leq i \leq P$, the quantity $\left|f_{b,\mathbf{w}}(\mathbf{x}_i) - y_i\right|$ is the vertical distance from $(\mathbf{x}_i, y_i)$ to the point predicted by the linear model, $(\mathbf{x}_i, f_{\tilde{\mathbf{w}}}(\mathbf{x}_i))$.



The length of the vector $\mathbf{y} - \hat{\mathbf{y}}$ (*i.e.*, the distance from $\mathbf{y}$ to the vector of predictions, in the column space of our feature matrix) is equal to

$$\sqrt{(f_{\tilde{\mathbf{w}}}(\mathbf{x}_1) - y_1)^2 + (f_{\tilde{\mathbf{w}}}(\mathbf{x}_2) - y_2)^2 + \ldots + (f_{\tilde{\mathbf{w}}}(\mathbf{x}_P) - y_P)^2} = \sqrt{g(\tilde{\mathbf{w}})}.$$

We see that minimizing $g(\tilde{\mathbf{w}})$ is the same as minimizing that distance, giving the $\hat{\mathbf{y}}$ in the column space that makes $\mathbf{y} - \hat{\mathbf{y}}$ orthogonal to the column space.