Linear Regression

Ngoc Hoang Luong

University of Information Technology (UIT), VNU-HCM

April 6, 2023

Motivation

- Example: we want to predict the salary of NBA players in terms of certain variables: team, height, weight, position, years of experience, number of 2pts, number of 3pts, number of blocks, etc.
- We have information about some current NBA players:

Player	Height	Weight	Yrs Expr	2 Points	3 Points	Salary
1	...	...	...	...	...	...
2	...	...	...	...	...	...
3	...	...	...	...	...	...
...	...	...	...	...	...	...

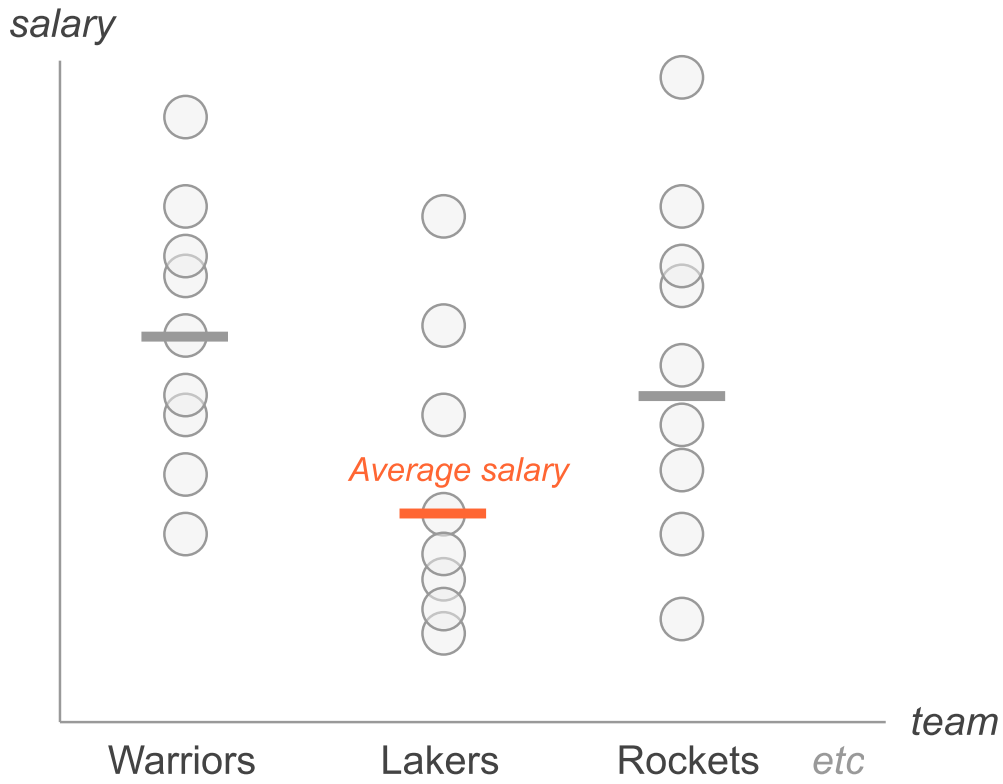
- $\mathbf{x}_i$ denotes the vector of measurements of player i's statistics (height, weight, etc.). And, y_i denotes the salary of the i-th player.
- We assume the existence of some **unknown** function $f : \mathcal{X} \to y$ that determines the ideal salary.
- We seek a model ($\hat{f} : \mathcal{X} \to y$), which we select from some set of candidate functions $h_1, h_2, \ldots, h_m$, that best approximates f.

The Intuition of Regression

- For simplicity, let's assume no inflation. We want to predict the salary of a new player.
- **Scenario 1**: We have no information on this new player. How would we predict the salary y_0?
- We can guess the salary using the historical average salary $\bar{y}$ of NBA players: $\hat{y}_0 = \bar{y}$.
- We use $\bar{y}$ as the typical score (i.e., a measure of center) as a plausible guess for y_0.
- We could also use the median of existing salaries, to disregard outliers.

The Intuition of Regression

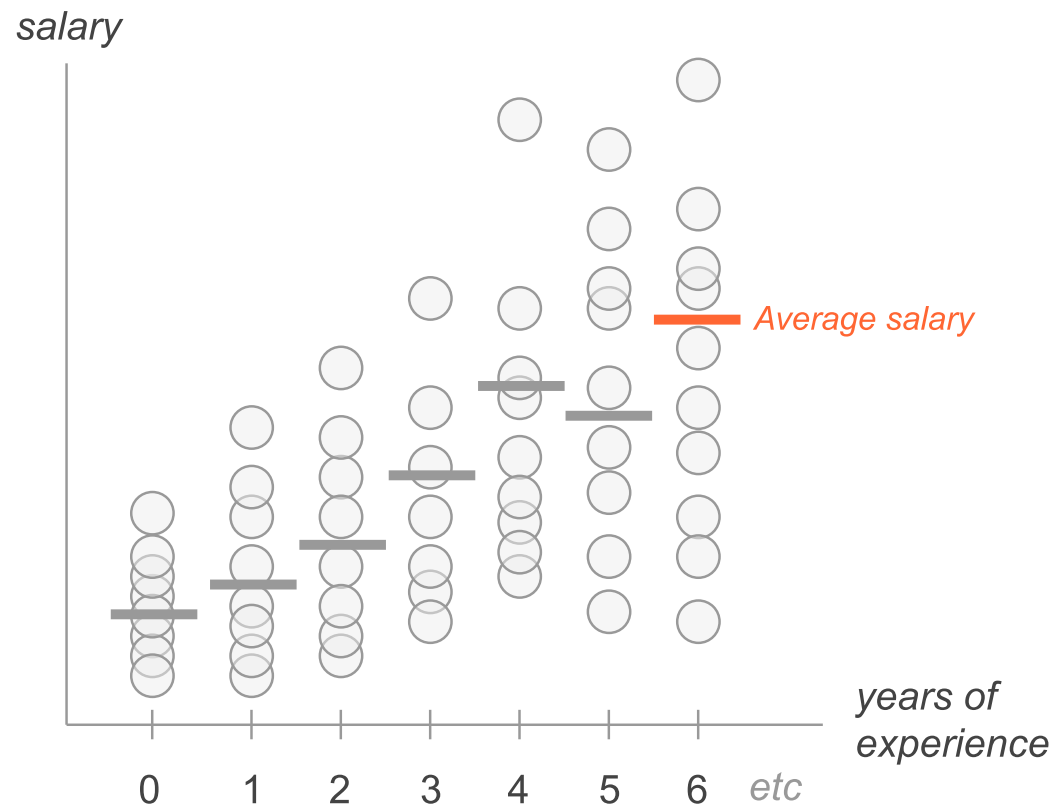
- **Scenario 2**: We know the new player will join LA Lakers. We can use this information to have a more educated guess for y_0.



- Instead of using the salaries of all players, we focus on the salaries of Laker's players: $y_0 = \text{avg}(\text{Laker's Salaries})$.

The Intuition of Regression

- **Scenario 3**: If we know this new player has 6 years of experience, we look at the average salaries of players with the same experience.



- In all examples, the predicted salary is a conditional mean:

$$\hat{y}_0 = \text{avg}(y_i | \mathbf{x}_i = \mathbf{x}_0)$$

The Intuition of Regression

- The prediction is a conditional mean:

$$\hat{y}_0 = \texttt{avg}(y_i | \mathbf{x}_i = \mathbf{x}_0)$$

- But this strategy only works if we have data points $\mathbf{x}_i$ match the query point $\mathbf{x}_0$.
- The core idea of regression: Obtaining prediction $\hat{y}_0$ using quantities of the form $\texttt{avg}(y_i | \mathbf{x}_i = \mathbf{x}_0)$, which can be formalized as:

$$\mathbb{E}(y_i | x_{i1}^*, x_{i2}^*, \ldots, x_{ip}^*) \longrightarrow \hat{y}$$

  where x_{ij}^* is the i-th measurement of the j-th variable.
- The **regression function**: a conditional expectation.

The Linear Regression Model

- In a regression model, we use one or more features X to predict the response Y.
- A linear regression model tells us how to combine the features into linear way to approximate the response.
- In the univariate case, we have a linear equation:

$$\hat{Y} = b_0 + b_1 X$$

- For a given individual i, we have:

$$\hat{y}_i = b_0 + b_1 x_i$$

or:

$$\hat{\mathbf{y}} = b_0 + b_1 \mathbf{x}$$

The Linear Regression Model

- We can add an auxiliary constant feature in the form of a vector of 1's and use the matrix notations:

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{b}$$

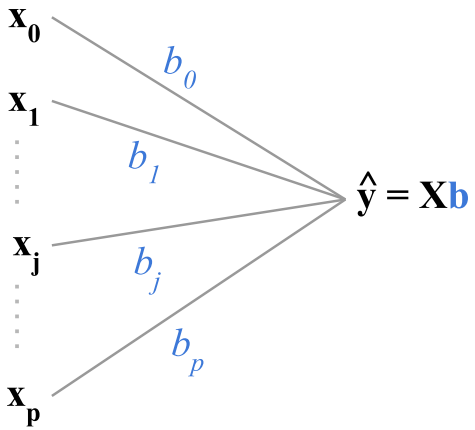
where:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}; \quad \hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

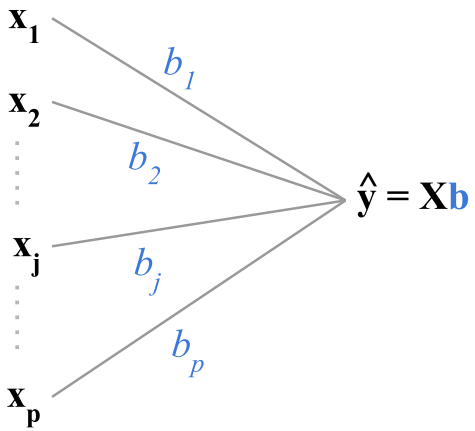
- In the multivariate case, when $p > 1$, we have:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \dots & x_{np} \end{bmatrix}; \quad \hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_p \end{bmatrix}$$

The Linear Regression Model



- If the predictors and the response are mean-centered, we can ignore the constant term x_0:



- How to obtain the vector of coefficients $\mathbf{b}$?

The Error Measure

- We want the predictions $\hat{y}_i$ to be as close as possible to y_i.
- To measure how close $\hat{y}_i$ and y_i, the most common choise is the squared distance:

$$d^2(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2 = (\hat{y}_i - y_i)^2 = (\mathbf{b}^\top \mathbf{x}_i - y_i)^2$$

- To measure the overall error, we can use:
  - The sum of squared errors (SSE):

$$\texttt{SSE} = \sum_{i=1}^{n} d^2(y_i, \hat{y}_i)$$

  - The mean squared error (MSE):

$$\texttt{MSE} = \frac{1}{n} \sum_{i=1}^{n} d^2(y_i, \hat{y}_i)$$

The Error Measure

- Let $e_i = (y_i - \hat{y}_i)$
- We have:

$$\begin{aligned}
\texttt{MSE} &= \frac{1}{n}\sum_{i=1}^{n} e_i^2 = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{b}^\top \mathbf{x}_i - y_i)^2 = \frac{1}{n}(\mathbf{Xb} - \mathbf{y})^\top(\mathbf{Xb} - \mathbf{y}) \\
&= \frac{1}{n}\|\mathbf{Xb} - \mathbf{y}\|^2 = \frac{1}{n}\|\hat{\mathbf{y}} - \mathbf{y}\|^2 \\
&= \frac{1}{n}\|e\|^2 = \frac{1}{n}(\hat{\mathbf{y}} - \mathbf{y})^\top(\hat{\mathbf{y}} - \mathbf{y})
\end{aligned}$$

- The MSE is proportional to the squared norm of the residual vector $\mathbf{e} = \hat{\mathbf{y}} - \mathbf{y}$

The Least Squares Algorithm

- In (ordinary least squares regression, we minimize the mean of squared errors (MSE).
- We compute the gradient of MSE wrt $\mathbf{b}$.

$$\begin{aligned}\nabla \mathtt{MSE}(\mathbf{b}) &= \frac{\partial}{\partial \mathbf{b}} \mathtt{MSE}(\mathbf{b}) \\ &= \frac{\partial}{\partial \mathbf{b}} \left(\frac{1}{n} \mathbf{b}^\top \mathbf{X}^\top \mathbf{X} \mathbf{b} - \frac{2}{n} \mathbf{b}^\top \mathbf{X}^\top \mathbf{y} + \frac{1}{n} \mathbf{y}^\top \mathbf{y} \right) \\ &= \frac{2}{n} \mathbf{X}^\top \mathbf{X} \mathbf{b} - \frac{2}{n} \mathbf{X}^\top \mathbf{y}\end{aligned}$$

- Equating to zero we have the **Normal Equations**:

$$\mathbf{X}^\top \mathbf{X} \mathbf{b} = \mathbf{X}^\top \mathbf{y}$$

- This is a system of n equations with $p+1$ unknowns (including the constant term b_0).

The Least Squares Algorithm

$$\mathbf{X}^\top \mathbf{X}\mathbf{b} = \mathbf{X}^\top \mathbf{y}$$

- If $\mathbf{X}^\top \mathbf{X}$ is invertible, then the vector of regression coefficients $\mathbf{b}$:

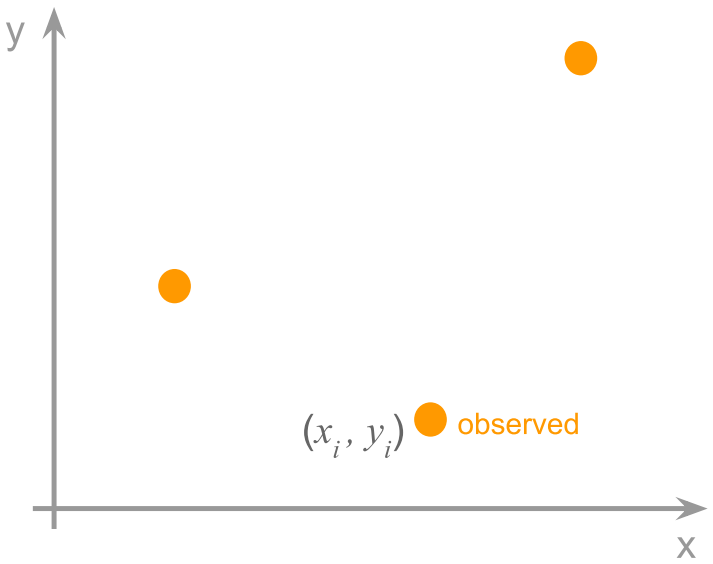
$$\mathbf{b} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

- We can compute the response vector:

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{b} = \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} = \mathbf{H}\mathbf{y}$$

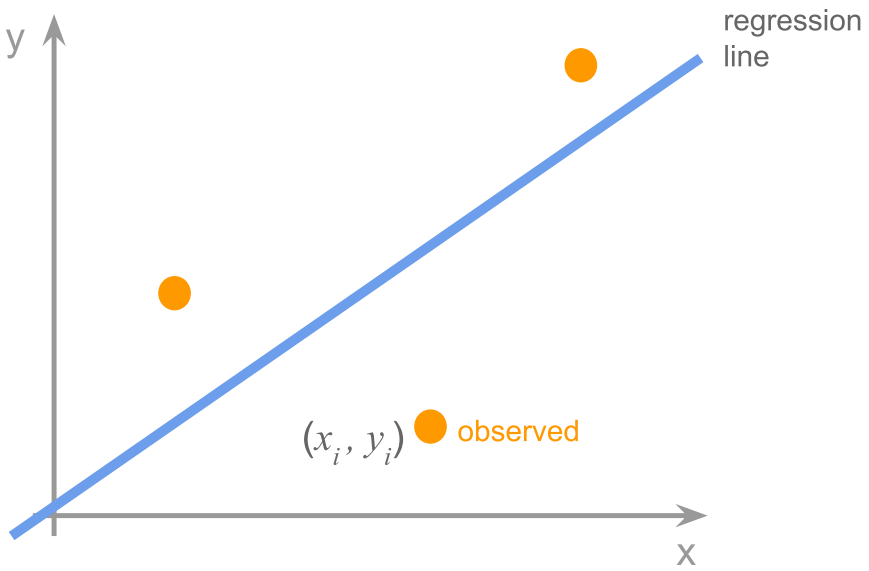
- The hat matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top$ is an **orthogonal projector**:
  - It is symmetric.
  - It is idempotent.
  - Its eigenvalues are either 0 or 1.

Geometries of OLS - Rows Perspective



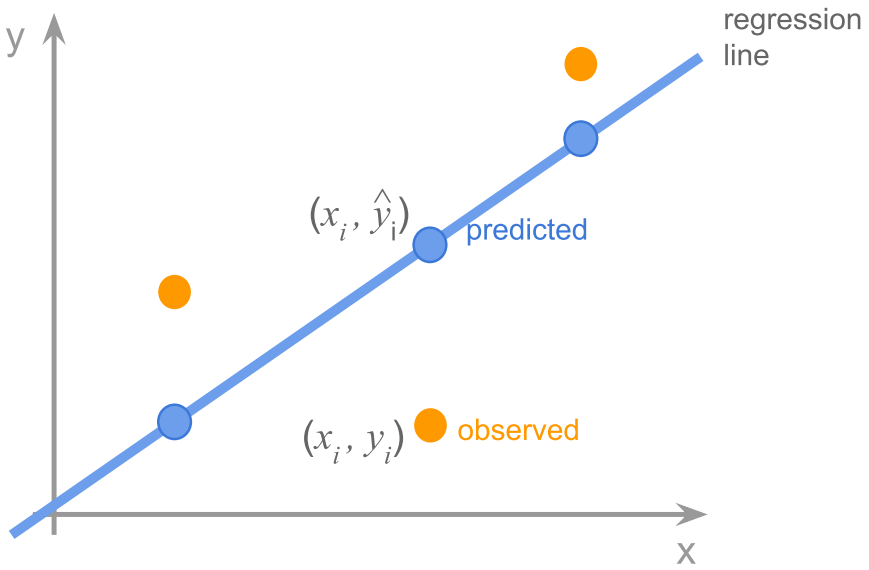
- Assume we have the response Y and one predictor X (i.e., $p = 1$).

Geometries of OLS - Rows Perspective



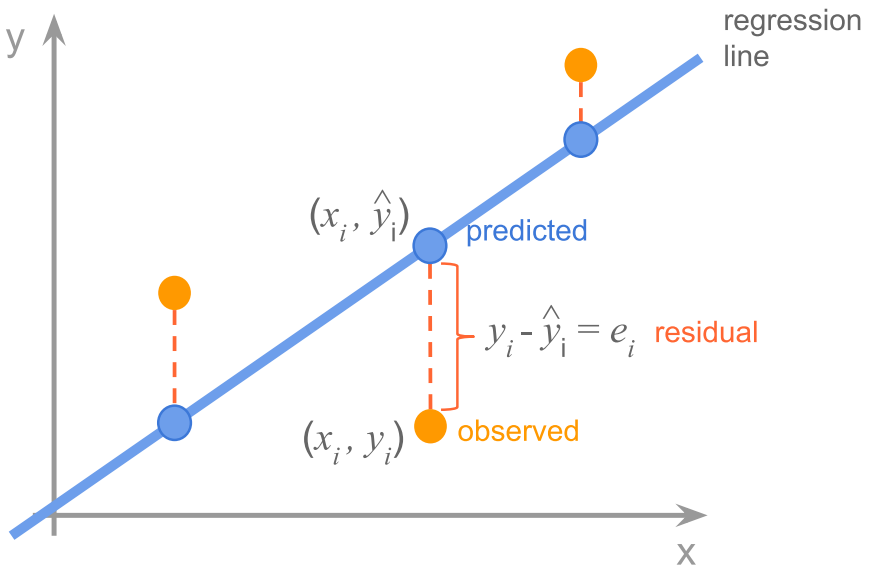
- We predict y_i by linear combining the inputs $\hat{y}_i = b_0 + b_1 x_i$. In 2D, the fitted model is a line.
- In 3D, the fitted model would be a plan. In higher dimensions, it would be a hyperplane.

Geometries of OLS - Rows Perspective



- With a model, we obtain predicted value $\hat{y}_i$.
- Some predicted values are equal to the observed values.
- Some predicted values are higher than the observed values.
- Some predicted values are smaller than the observed values.

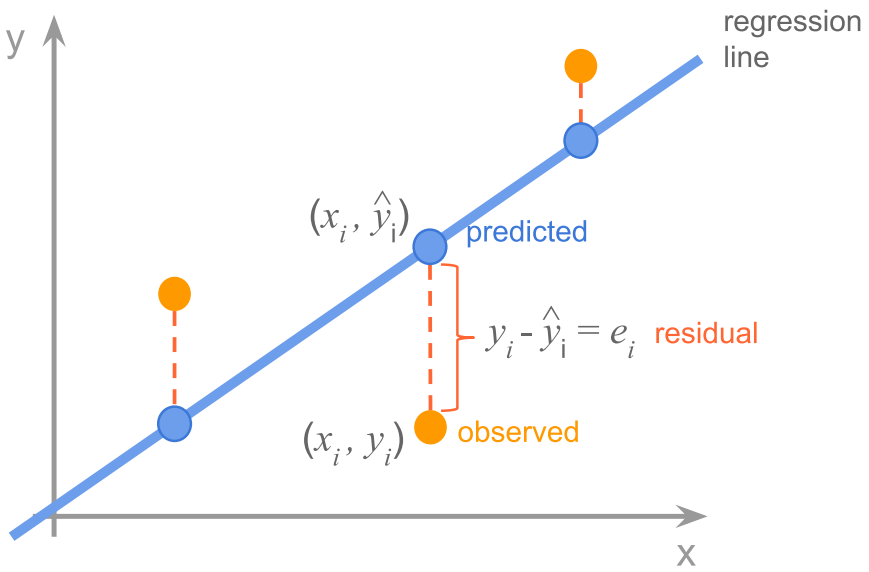
Geometries of OLS - Rows Perspective



- Given a set of data points, we want the line that minimizes the squares of the errors $e_i = \hat{y}_i - y_i$, which are known as the **residuals**.
- We want to find parameters $b_0, b_1, \ldots, b_p$ that minimize the squared norm of the vector of residuals.

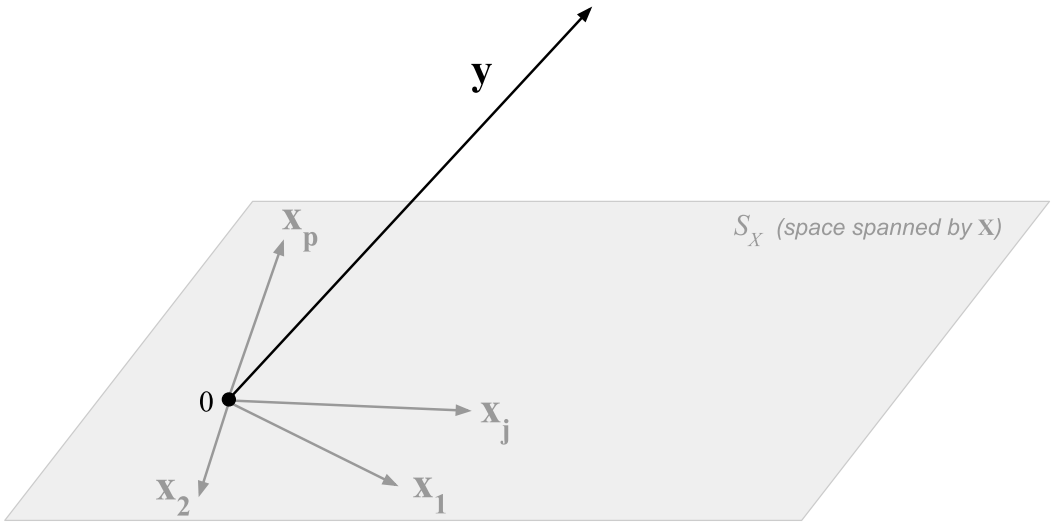
$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 = \sum_{i=1}^{n} (b_0 + b_1 x_i - y_i)^2$$

Geometries of OLS - Rows Perspective



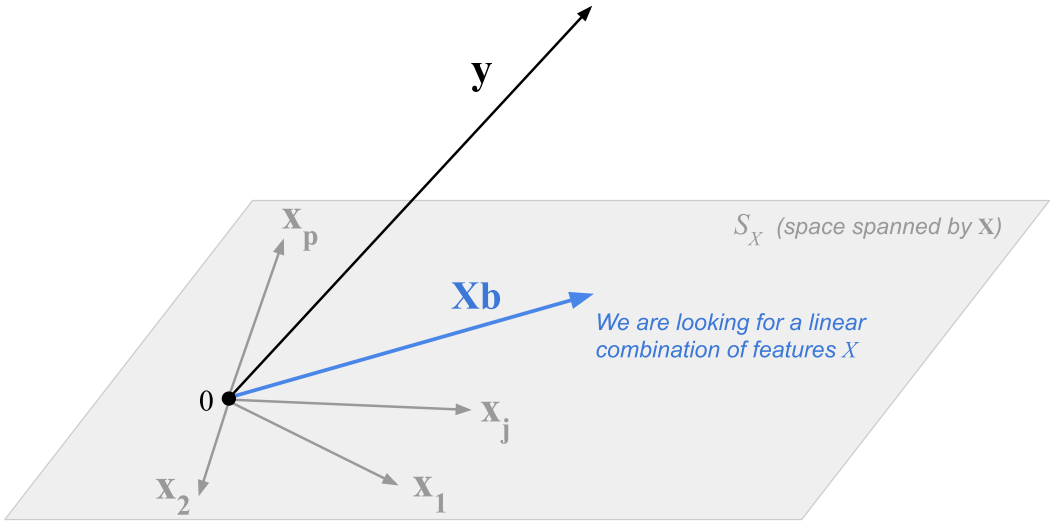
$$
\begin{aligned}
\|\mathbf{e}\|^2 &= \|\hat{\mathbf{y}} - \mathbf{y}\|^2 \\
&= \|\mathbf{Xb} - \mathbf{y}\|^2 \\
&= (\mathbf{Xb} - \mathbf{y})^\top (\mathbf{Xb} - \mathbf{y}) \\
&\propto \texttt{MSE}
\end{aligned}
$$

Geometries of OLS - Columns Perspective



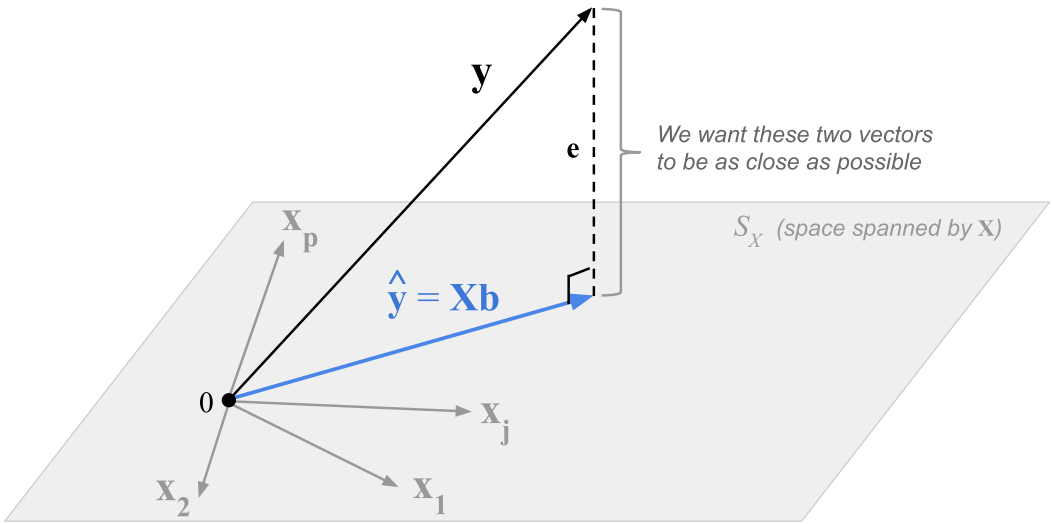
- Consider the variables in the n-dimensional spaces, both the response and the predictors.
- The X variables span some subspace $\mathbb{S}_X$. This subspace does not contain the response Y, unless Y is a linear combination of $X_1, X_2, \ldots, X_p$.

Geometries of OLS - Columns Perspective



- There are an infinite number of linear combinations of $X_1, X_2, \ldots, X_p$.
- We want a linear combination $\mathbf{Xb}$ that best approximates $\mathbf{y}$.

Geometries of OLS - Columns Perspective



- We want a mix of features $\hat{\mathbf{y}} = \mathbf{Xb}$ that is the closest approximation to $\mathbf{y}$.
- The difference between $\hat{\mathbf{y}}$ and $\mathbf{y}$ is: $\mathbf{e} = \hat{\mathbf{y}} - \mathbf{y}$.
- We want $\hat{\mathbf{y}}$ such that the squared norm $\|\mathbf{e}\|^2$ is as small as possible.

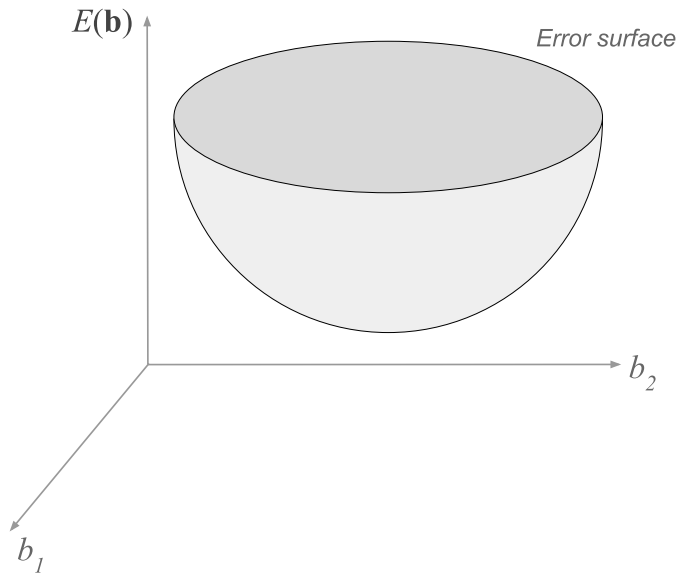
$$\min \|\mathbf{e}\|^2 = \min \|\hat{\mathbf{y}} - \mathbf{y}\|^2 \propto \min \mathtt{MSE}$$

Geometries of OLS - Parameters Perspective

- From the point of view of parameters $\mathbf{b}$, we can classify the order of each term in the Mean Squared Error (MSE):

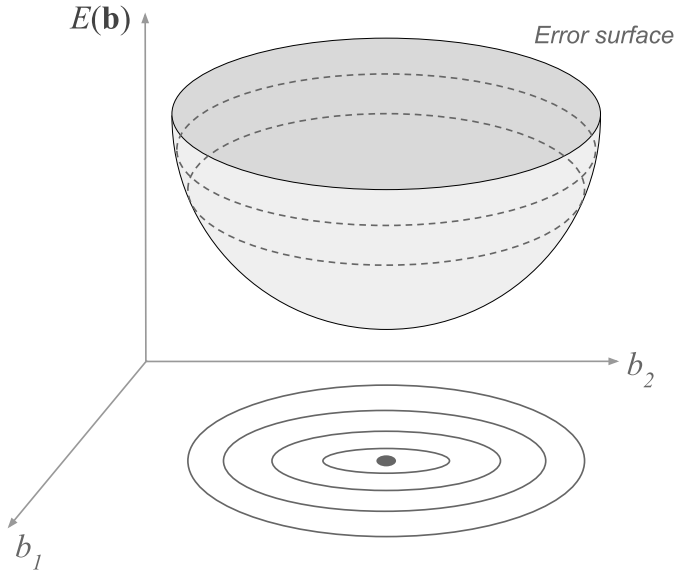
$$E(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}(\underbrace{\mathbf{b}^\top \mathbf{X}^\top \mathbf{X}\mathbf{b}}_{\text{Quadratic Form}} \underbrace{- 2\mathbf{b}^\top \mathbf{X}^\top \mathbf{y}}_{\text{Linear}} + \underbrace{\mathbf{y}^\top \mathbf{y}}_{\text{Constant}})$$

- Since $\mathbf{X}^\top \mathbf{X}$ is positive semidefinite, we know that $\mathbf{b}^\top \mathbf{X}^\top \mathbf{X}\mathbf{b} \geq 0$.
- Assume we have only two predictors X_1 and X_2, then $\mathbf{b} = (b_1, b_2)$. The `MSE` will be a paraboloid in (E, b_1, b_2) space.



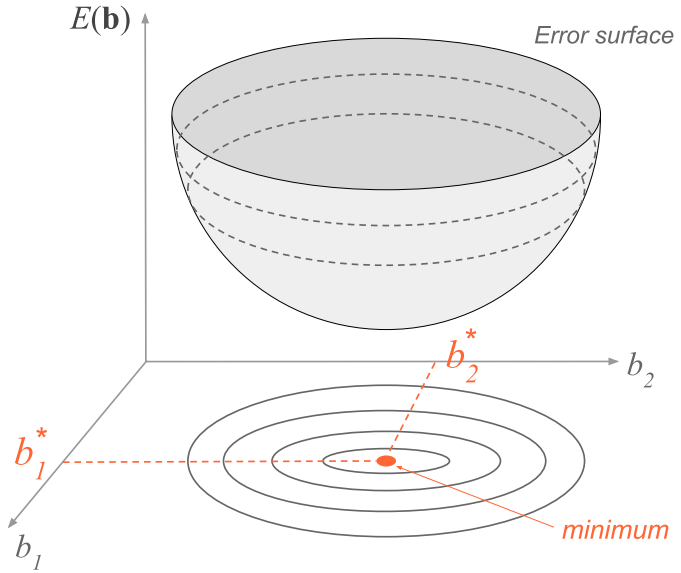
Geometries of OLS - Parameters Perspective

- Since $\mathbf{X}^\top\mathbf{X}$ is positive semidefinite, we know that $\mathbf{b}^\top\mathbf{X}^\top\mathbf{X}\mathbf{b} \geq 0$.
- Assume we have only two predictors X_1 and X_2, then $\mathbf{b} = (b_1, b_2)$. The `MSE` will be a paraboloid in (E, b_1, b_2) space.
- Imagine we get horizontal slices of the `MSE` surface. For each slice, we can project it onto the plane spanned by parameters b_1 and b_2. The resulting projections are like a topographic map, with error contours on this plane.



Geometries of OLS - Parameters Perspective

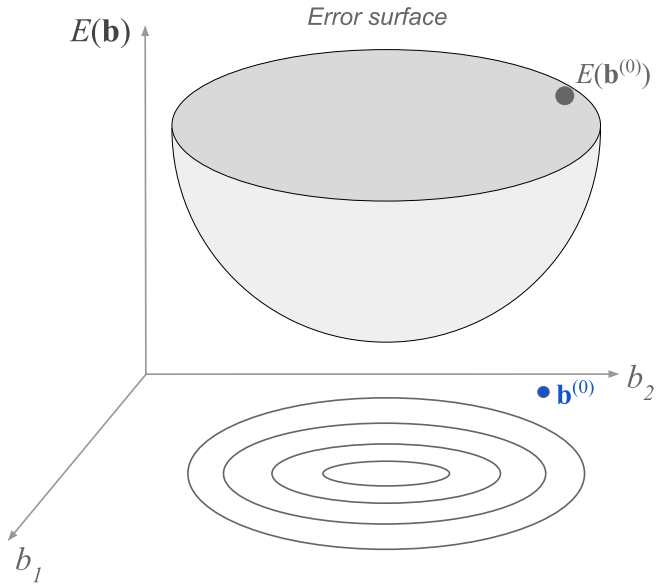
- Assume we have only two predictors X_1 and X_2, then $\mathbf{b} = (b_1, b_2)$. The `MSE` will be a paraboloid in (E, b_1, b_2) space.
- Imagine we get horizontal slices of the `MSE` surface. For each slice, we can project it onto the plane spanned by parameters b_1 and b_2. The resulting projections are like a topographic map, with error contours on this plane.



- The minimum of the error surface at point (b_1^*, b_2^*).
- Assuming $\mathbf{X}^\top\mathbf{X}$ invertible, $\mathbf{b}^* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$.

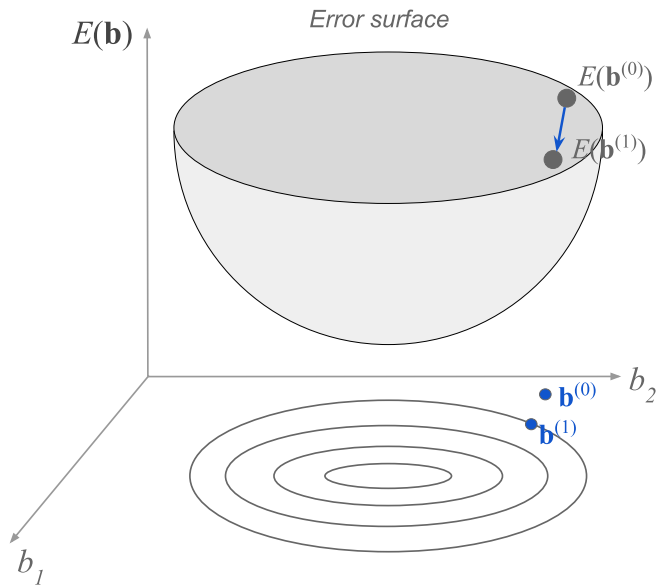
Idea of Gradient Descent

- Start with an random point $\mathbf{b}^{(0)} = (b_1^{(0)}, b_2^{(0)})$ of model parameters.
- Evaluate the error function at this point $E(\mathbf{b}^{(0)})$. This gives a location somewhere on the loss surface.



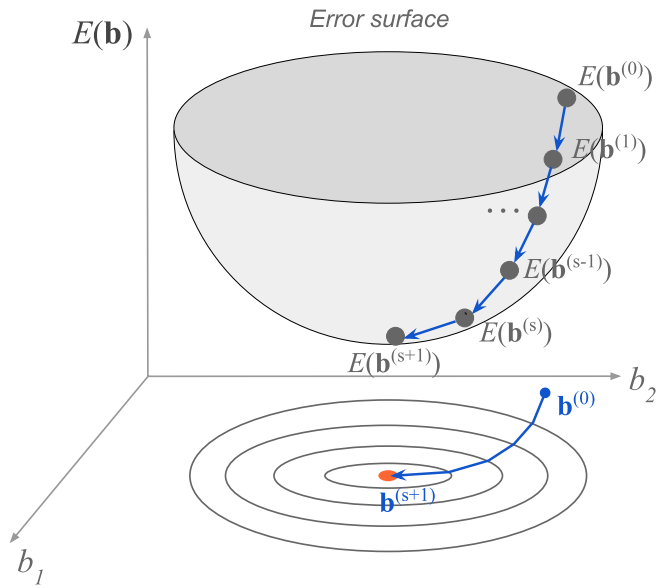
Idea of Gradient Descent

- Start with an random point $\mathbf{b}^{(0)} = (b_1^{(0)}, b_2^{(0)})$ of model parameters.
- Evaluate the error function at this point $E(\mathbf{b}^{(0)})$. This gives a location somewhere on the loss surface.
- We get a new vector $\mathbf{b}^{(1)}$ so that we "move down" the surface to obtain a new position $E(\mathbf{b}^{(1)})$.



Idea of Gradient Descent

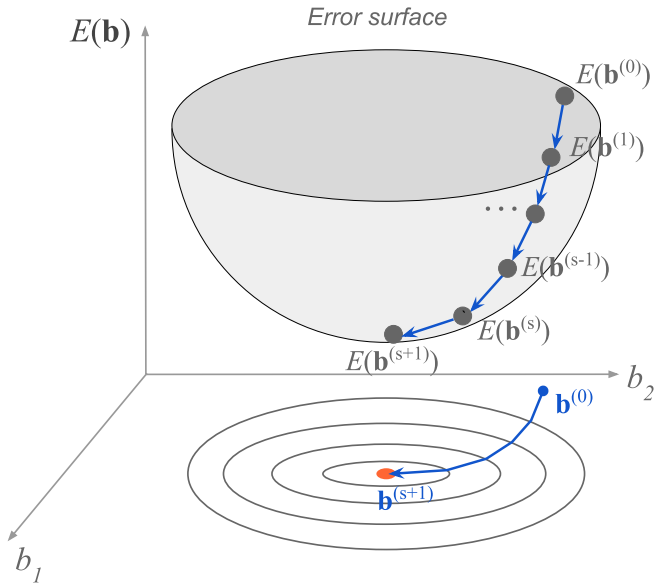
- Start with an random point $\mathbf{b}^{(0)} = (b_1^{(0)}, b_2^{(0)})$ of model parameters.
- Evaluate the error function at this point $E(\mathbf{b}^{(0)})$. This gives a location somewhere on the loss surface.
- We get a new vector $\mathbf{b}^{(1)}$ so that we "move down" the surface to obtain a new position $E(\mathbf{b}^{(1)})$.
- At each step s, we obtain a new vector $\mathbf{b}^{(s)}$ that yields an error $E(\mathbf{b}^{(s)})$ that is closer to the minimum of $E(\mathbf{b})$.



- Eventually, we should get very close to the minimum $\mathbf{b}^*$.

Idea of Gradient Descent

- Keep in mind that we don't see the surface.
- We only have local information at the current point $\mathbf{b}^{(s)}$ we evaluate the error function $E(\mathbf{b}^{(s)})$.



- Imagine we need to get to the bottom of a valley from the top of a mountain at night.
- We touch our surroundings to feel which direction the slope of the terrain goes down.
- We identify the direction that gives the steepest descent.

Moving Down an Error Surface

- "Moving down an error surface" means that we generate the new vector $\mathbf{b}^{(s+1)}$ from the current point $\mathbf{b}^{(s)}$ using the formula:

$$\mathbf{b}^{(s+1)} = \mathbf{b}^{(s)} + \alpha \mathbf{v}^{(s)}$$

- $\mathbf{v}^{(s)}$ is the vector indicating the direction we move at step s. We can consider $\mathbf{v}^{(s)}$ to be a unit vector.
- α the **step size**, indicating how far we move along direction $\mathbf{v}^{(s)}$.

$$\begin{aligned}
\mathbf{b}^{(1)} &= \mathbf{b}^{(0)} + \alpha \mathbf{v}^{(0)} \\
\mathbf{b}^{(2)} &= \mathbf{b}^{(1)} + \alpha \mathbf{v}^{(1)} \\
\vdots \qquad & \qquad \vdots \\
\mathbf{b}^{(s+1)} &= \mathbf{b}^{(s)} + \alpha \mathbf{v}^{(s)}
\end{aligned}$$

- We assume a *constant* step size α. More sophisticated versions of gradient descent allow *variable* step size.
- The direction $\mathbf{v}^{(s)}$ changes at each step s. How do we find $\mathbf{v}^{(s)}$?

The direction of $\mathbf{v}^{(s)}$

- What does it mean for $\mathbf{b}^{(s+1)}$ getting closer to the minimum?
- We want the error $E(\mathbf{b}^{(s+1)})$ is less than the error $E(\mathbf{b}^{(s)})$.
- The difference $\Delta E_{\mathbf{b}}$ should be as negative as possible

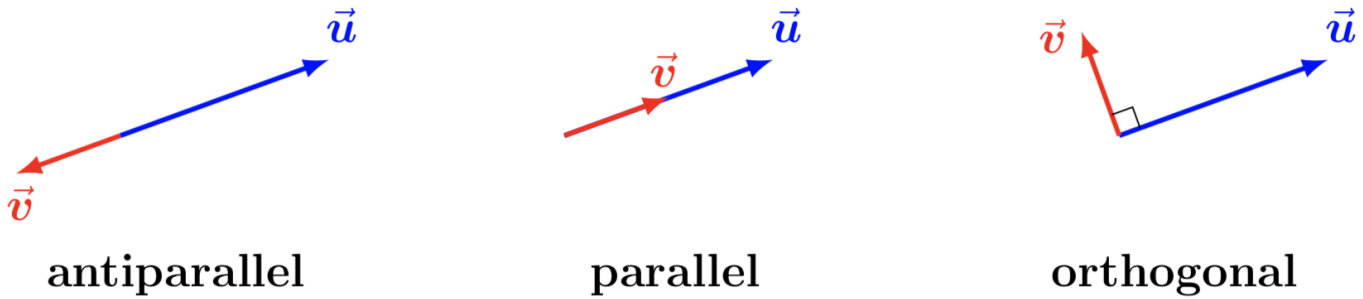
$$\Delta E_{\mathbf{b}} = E(\mathbf{b}^{(s+1)}) - E(\mathbf{b}^{(s)}) = E(\mathbf{b}^{(s)} + \alpha \mathbf{v}^{(s)}) - E(\mathbf{b}^{(s)})$$

- Apply Taylor series expansion of $E(\mathbf{b}^{(s)} + \alpha \mathbf{v}^{(s)})$, we get:

$$\begin{aligned}\Delta E_{\mathbf{b}} &= E(\mathbf{b}^{(s)}) + \nabla E(\mathbf{b}^{(s)})^{\top}(\alpha \mathbf{v}^{(s)}) + O(\alpha^2) - E(\mathbf{b}^{(s)}) \\ &= \alpha \nabla E(\mathbf{b}^{(s)})^{\top} \mathbf{v}^{(s)} + O(\alpha^2) \approx \alpha \nabla E(\mathbf{b}^{(s)})^{\top} \mathbf{v}^{(s)}\end{aligned}$$

- The last term involves the inner product between the gradient and a unit vector: $[\nabla E(\mathbf{b}^{(s)})]^{\top} \mathbf{v}^{(s)}$
- Denote $\mathbf{u} = \nabla E(\mathbf{b}^{(s)})$, we need to find $\mathbf{v}$ to make $\mathbf{u}^{\top}\mathbf{v}$ as negative as possible.

The direction of $\mathbf{v}^{(s)}$



- When $\mathbf{u}$ and $\mathbf{v}$ are parallel, then $\mathbf{u}^\top\mathbf{v} = \|\mathbf{u}\|$ (we assume $\mathbf{v}$ to be a unit vector).
- When $\mathbf{u}$ and $\mathbf{v}$ are antiparallel, then $\mathbf{u}^\top\mathbf{v} = -\|\mathbf{u}\|$.
- When $\mathbf{u}$ and $\mathbf{v}$ are orthogonal, then $\mathbf{u}^\top\mathbf{v} = 0$.
- In any case, we have that:

$$\mathbf{u}^\top\mathbf{v} \geq -\|\mathbf{u}\|$$

- The least we can get is $-\|\mathbf{u}\|$ when $\mathbf{v}$ is the opposite direction of $\mathbf{u} = \nabla E(\mathbf{b}^{(s)})$, i.e., the gradient of $E(\mathbf{b})$ at step s.

The direction of $\mathbf{v}^{(s)}$

$$\Delta E_{\mathbf{b}} = \alpha \nabla E(\mathbf{b}^{(s)})^{\top} \mathbf{v}^{(s)} + O(\alpha^2) \geq -\alpha \| E(\mathbf{b}^{(s)}) \|$$

- To make $\Delta E_{\mathbf{b}}$ as negative as possible, $\mathbf{v}^{(s)}$ should be parallel to the opposite direction of the gradient $\nabla E(\mathbf{b}^{(s)})$:

$$\mathbf{v}^{(s)} = -\frac{\nabla E(\mathbf{b}^{(s)})}{\| \nabla E(\mathbf{b}^{(s)}) \|}$$

- The norm division is to make $\mathbf{v}^{(s)}$ a unit vector. However, we don't need to implement this normalization because it can be absorbed into the step size α.
- **Gradient Descent**: We are descending in the direction opposite to the gradient of the error function.

GD for Linear Regression in Vector-Matrix Notation

- The error function $E(\mathbf{b})$:

$$E(\mathbf{b}) = \frac{1}{n}(\mathbf{y} - \mathbf{X}\mathbf{b})^\top(\mathbf{y} - \mathbf{X}\mathbf{b}) = \frac{1}{n}(\mathbf{b}\mathbf{X}^\top\mathbf{X}\mathbf{b} - 2\mathbf{b}^\top\mathbf{X}^\top\mathbf{y} + \mathbf{y}^\top\mathbf{y})$$

- The formula for its gradient $\nabla E(\mathbf{b})$ is:

$$\nabla E(\mathbf{b}) = \frac{1}{n}(2\mathbf{X}^\top\mathbf{X}\mathbf{b} - 2\mathbf{X}^\top\mathbf{y}) = \frac{2}{n}\mathbf{X}^\top(\mathbf{X}\mathbf{b} - \mathbf{X}^\top\mathbf{y})$$

- The algorithm:
  1. Initialize $\mathbf{b}^{(0)} = (b_0^{(0)}, b_1^{(0)}, \ldots, b_p^{(0)})$
  2. For $s = 0, 1, 2, \ldots$ do:
     - Update model parameters $\mathbf{b}$:

$$\mathbf{b}^{(s+1)} = \mathbf{b}^{(s)} - \alpha\nabla E(\mathbf{b}^{(s)}) = \mathbf{b}^{(s)} - \alpha\left[\frac{2}{n}\mathbf{X}^\top(\mathbf{X}\mathbf{b}^{(s)} - \mathbf{X}^\top\mathbf{y})\right]$$

     - When there is little change between $\mathbf{b}^{(k+1)}$ and $\mathbf{b}^{(k)}$ (for some k), we assume the algorithm **converged** and $\mathbf{b}^* = \mathbf{b}^{(k+1)}$.

GD for Linear Regression in Pointwise Notation

- The error function $E(\mathbf{b})$:

$$E(\mathbf{b}) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \mathbf{b}^\top \mathbf{x}_i\right)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$
$$= \frac{1}{n}\sum_{i=1}^{n}(y_i - b_0 x_{i0} - b_1 x_{i1} - \ldots - b_j x_{ij} - \ldots - b_p x_{ip})$$

- The partial derivate wrt b_j is:

$$\frac{\partial E(\mathbf{b})}{\partial b_j} = -\frac{2}{n}\sum_{i=1}^{n}(y_i - b_0 x_{i0} - b_1 x_{i1} - \ldots - b_j x_{ij} - \ldots - b_p xip)x_{ij}$$
$$= -\frac{2}{n}\sum_{i=1}^{n}\left(y_i - \mathbf{b}^\top \mathbf{x}_i\right)x_{ij}$$

GD for Linear Regression in Pointwise Notation

1. Initialize $\mathbf{b}^{(0)} = (b_0^{(0)}, b_1^{(0)}, \ldots, b_p^{(0)})$
2. For $s = 0, 1, 2, \ldots$ do:
   - Update model parameters $\mathbf{b}$:

$$b_j^{(s+1)} = b_j^{(s)} + \alpha \cdot \frac{\partial}{\partial b_j} E(\mathbf{b}^{(s)})$$
$$= b_j^{(s)} + \alpha \cdot \frac{2}{n} \sum_{i=1}^{n} \left(y_i - [\mathbf{b}^{(s)}]^\top \mathbf{x}_i \right) x_{ij}$$

     for all $j = 0, 1, \ldots, p$ simultaneously.
   - Store these elements into the vector:

$$\mathbf{b}^{(s+1)} = (b_0^{(s+1)}, b_1^{(s+1)}, \ldots, b_p^{(s+1)})$$

   - When there is little change between $\mathbf{b}^{(k+1)}$ and $\mathbf{b}^{(k)}$ (for some k), we assume the algorithm **converged** and $\mathbf{b}^* = \mathbf{b}^{(k+1)}$.