EE2211 Lecture 5: Least Squares and Linear Regression

Vincent Y. F. Tan

Department of Electrical and Computer Engineering, NUS

EE2211 Spring 2023

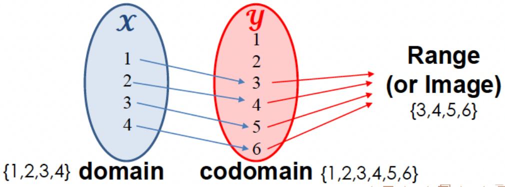
Acknowledgements to

Xinchao, Helen, Thomas, Kar Ann, Chen Khong, Robby, and Haizhou

Some Basic Mathematical Notions : Sets

- A set S is an unordered collection of objects.
- $S = \{1, 2, 3, 4, 5, 6\}$ is the possible outcomes of the toss of a die.
- $S = [a, b] = \{x \in \mathbb{R} : a \leq x \leq b\}$ is the set of all numbers from a to b inclusive.
- $S = (a, b] = \{x \in \mathbb{R} : a < x \leq b\}$ is the set of all numbers from a to b, excluding a including b.
- $\mathbb{R}$ is the set of all real numbers.
- $\mathbb{R}^d$ is the set of all real vectors of length d

- A function f is a map from a set X to another set Y. We write this as
$$f : X \to Y.$$
- For example, the function $f : \mathbb{R} \to [0, \infty)$ could be given by the recipe $f(x) = x^2$.
- The set of inputs is called the domain; the set of possible outputs is called the codomain; the set $\{f(x) : x \in X\}$ is called the range (or image).
- For example $f : \{1, 2, 3, 4\} \to \{1, 2, 3, 4, 5, 6\}$ given by the recipe $f(x) = x + 2$ has codomain $\{1, 2, 3, 4, 5, 6\}$ and range $\{3, 4, 5, 6\}$.



Linear Functions

A function $f : \mathbb{R}^d \to \mathbb{R}$ is linear if it satisfies

- (Homogeneity) For any vector $\mathbf{x} \in \mathbb{R}^d$ and scalar $a \in \mathbb{R}$,

$$f(a\,\mathbf{x}) = a\,f(\mathbf{x})$$

- (Additivity) For any two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

$$f(\mathbf{x}+\mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$$

Note that a linear function $f : \mathbb{R}^d \to \mathbb{R}$ must pass through the origin, i.e., $f(\mathbf{0}) = 0$ where $\mathbf{0} \in \mathbb{R}^d$ is the zero vector in d dimensions. Why?

Linear Functions : Exercises I

Exercise: Show that if a function $f : \mathbb{R}^d \to \mathbb{R}$ is linear, then for any two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and two scalars $a, b \in \mathbb{R}$, then

$$f(a\,\mathbf{x} + b\,\mathbf{y}) = a\,f(\mathbf{x}) + b\,f(\mathbf{y}).$$

Show also that if we have n vectors $\mathbf{x}_i \in \mathbb{R}^d, i = 1, \ldots, n$ and n scalars $a_i \in \mathbb{R}, i = 1, \ldots, n$, a linear function satisfies

$$f\left(\sum_{i=1}^{n} a_i\,\mathbf{x}_i\right) = \sum_{i=1}^{n} a_i\,f(\mathbf{x}_i).$$

Exercise: Let $f : \mathbb{R} \to \mathbb{R}$ be the absolute value function

$$f(x) = |x|.$$

Is f linear?

Linear Functions : Exercises II

Exercise: Fix a vector $\mathbf{a} \in \mathbb{R}^d$ and define the function $f : \mathbb{R}^d \to \mathbb{R}$ as

$$f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x} = \sum_{i=1}^{d} a_i x_i.$$

This is called the inner product function. Show that f is linear.

Exercise: Fix a matrix $\mathbf{A} \in \mathbb{R}^{m \times d}$ and define the function $f : \mathbb{R}^d \to \mathbb{R}^m$ as

$$f(\mathbf{x}) = \mathbf{A}\mathbf{x}.$$

This is the regular matrix multiplication. Show that f is linear.

Affine Functions

- An affine function f is a linear function plus possibly a constant.
- More precisely, a function $f : \mathbb{R}^d \to \mathbb{R}$ is affine if it can be expressed as

$$f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x} + b$$

for some vector $\mathbf{a} \in \mathbb{R}^d$ and some scalar $b \in \mathbb{R}$.
- The scalar b is called the bias or offset.

Example: The following function $f : \mathbb{R}^2 \to \mathbb{R}$ is affine. Why?

$$f(\mathbf{x}) = f(x_1, x_2) = -x_1 + 3x_2 + 7.$$

Exercise: Is a linear function affine? Is an affine function linear?

Linear and Affine Functions

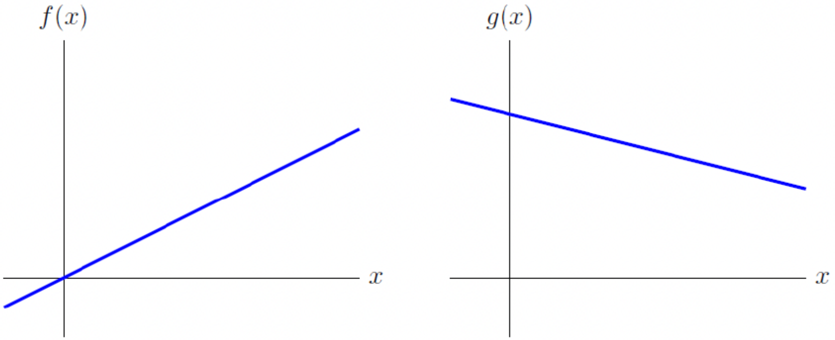


Figure 2.1 *Left.* The function f is linear. *Right.* The function g is affine, but not linear.

Local and Global Extrema

- Consider a function $f : [a,b] \to \mathbb{R}$.
- The function f has a local minimum at $c \in \mathbb{R}$ if

$$f(x) \geq f(c)$$

  for all x in an open neighborhood of c.
- The function f has a global minimum at $c \in \mathbb{R}$ if

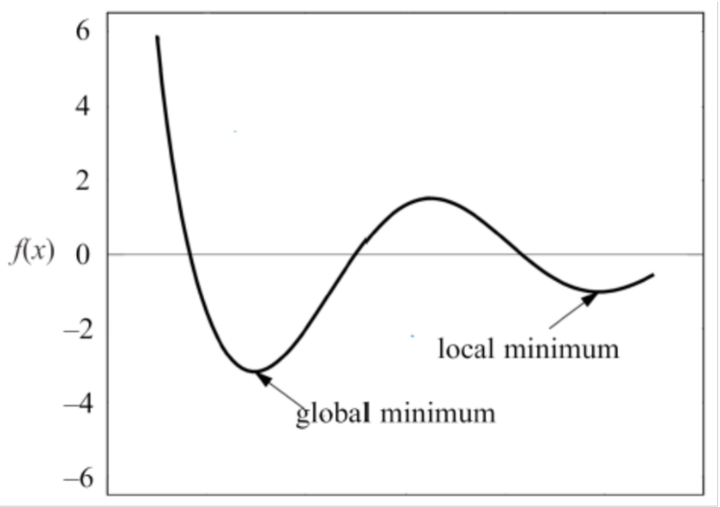
$$f(x) \geq f(c)$$

  for all $x \in [a,b]$.

Exercise: If c is a local minimum of f, is it a global minimum? If c is a global minimum of f, is it a local minimum?

Exercise: How would you define local maximum and global maximum?

Local and Global Extrema



$\min$ and $\arg\min$

- For a function $f : X \to Y$, the minimum $\min_{x\in X} f(x)$ returns the smallest value among all elements in the set $\{f(x) : x \in X\}$.
- For a function $f : X \to Y$, the argmin $x^* = \arg\min_{x\in X} f(x)$ returns the value of $x \in X$ that minimizes $f(x)$, i.e.,

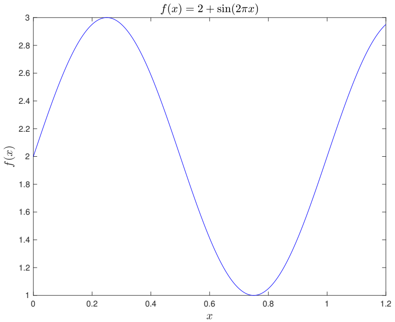
$$f(x^*) = \min_{x\in X} f(x)$$

- $\arg\min$ returns a value from the domain of the function X and $\min$ returns from the range (codomain) Y of the function.
- Let $X = \{0, 1\}$ and $f(0) = \pi$ and $f(1) = e$. Then

$$\arg\min_{x\in X} f(x) = 1 \qquad \min_{x\in X} f(x) = e,$$

and

$$\arg\max_{x\in X} f(x) = 0 \qquad \max_{x\in X} f(x) = \pi.$$



- Let $f : X = [0, 1.2] \to \mathbb{R}$ be defined as $f(x) = 2 + \sin(2\pi x)$ (see plot above). Then

$$\arg\min_{x \in X} f(x) = 3/4 \qquad \min_{x \in X} f(x) = +1$$

- Note that $f(3/4) = +1$.

Derivatives

- For a multivariable function $f : \mathbb{R}^d \to \mathbb{R}$, its gradient vector or derivative at $\mathbf{x} \in \mathbb{R}^d$ is the column vector

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \frac{\mathrm{d}f}{\mathrm{d}\mathbf{x}} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \cdots & \frac{\partial f}{\partial x_d} \end{bmatrix}^\top$$

- Recall that $\frac{\partial f}{\partial x_i}$ is the partial derivative of f with respect to the scalar variable x_i.
- For example, if $f(x_1, x_2) = 2x_1^2 + 5x_1x_2 + 3x_2^3$, then

$$\frac{\partial f}{\partial x_1} = 4x_1 + 5x_2 \quad \text{and} \quad \frac{\partial f}{\partial x_2} = 5x_1 + 9x_2^2.$$

Important Derivatives

- There are only two derivatives for functions $f : \mathbb{R}^d \to \mathbb{R}$ that take vectors to scalars you need to know for now.
- For a fixed vector $\mathbf{a} \in \mathbb{R}^d$, consider $f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x}$ (known as the inner or dot product). Then
$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \mathbf{a}.$$
- For a fixed matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$, consider $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}\mathbf{x}$ (known as the quadratic form). Then
$$\nabla_{\mathbf{x}} f(\mathbf{x}) = (\mathbf{A} + \mathbf{A}^\top)\mathbf{x}.$$
- In most applications, $\mathbf{A}$ is a symmetric matrix (i.e., $\mathbf{A} = \mathbf{A}^\top$) so
$$\nabla_{\mathbf{x}} f(\mathbf{x}) = 2\mathbf{A}\mathbf{x}.$$
- This generalizes the basic fact that if $f(x) = ax^2$, then $\frac{\mathrm{d}f}{\mathrm{d}x} = 2ax$.

Important Derivatives

Exercise: Show from the definition of $f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x}$ that

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \mathbf{a}.$$

Exercise: Show from the definition of $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}\mathbf{x}$ that

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = (\mathbf{A} + \mathbf{A}^\top)\mathbf{x}.$$

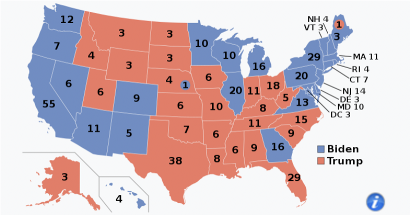
You'll be forgiven for having to exhibit a substantial amount of meticulous bookkeeping here.

Advice: It is difficult to remember a lot of derivative formulae of complicated multivariate functions. Usually, one consults the Matrix Cookbook

https://www2.imm.dtu.dk/pubdb/edoc/imm3274.pdf

Motivation for Linear Regression

- When I first taught this module in the Fall of 2020, we were in the midst of Covid sans vaccines, but there was another important global event.
- It was the 2020 United States presidential election, pitting the incumbent Republican Donald J. Trump against Democrat challenger Joseph R. Biden.



- Could we have used historical trends to predict who will win and by how much?

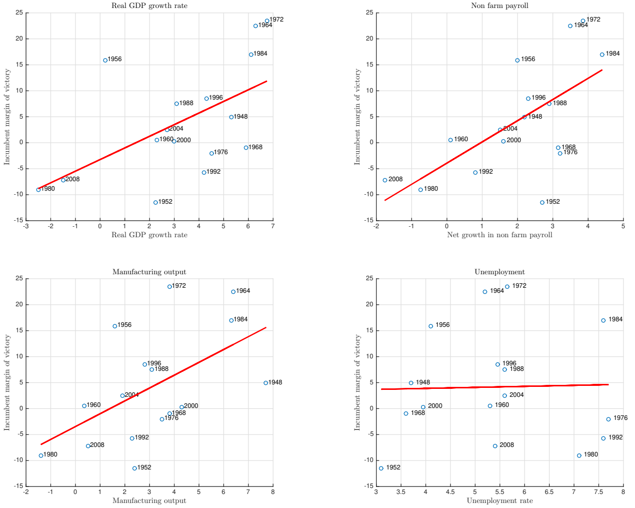
Motivation for Linear Regression

- Consider four economic indicators:
  (a) Real GDP growth rate x_1;
  (b) Change in non-farm payrolls x_2;
  (c) ISM (Institute of Supply Management) manufacturing index x_3;
  (d) Unemployment rate x_4.
- Which factor is the most important for determining the incumbent's winning margin?
- Data obtained from Nate Silver's blog at the New York Times.
- Data is of the form

$$\mathbf{x}_i = \begin{bmatrix} x_{i,1} \\ x_{i,2} \\ x_{i,3} \\ x_{i,4} \end{bmatrix} \quad \text{and} \quad y_t \quad \text{for} \quad i \in \{1948, 1952, \ldots, 2008\}$$

where $x_{i,1}$ is the real GDP growth rate in election in year i (etc.) and y_i is the incumbent's winning margin.

Motivation for Linear Regression



Scatter plots of incumbent's victory margin against various economic factors

Linear Regression

- Linear regression is a linear approach for modelling the relationship between a scalar response y and one or more explanatory variables (or attributes, or features) $\mathbf{x}$.
- We have a dataset $\{(\mathbf{x}_i, y_i) : i = 1, \ldots, m\}$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ are the feature vector and target of the i-th sample respectively.
- Without the offset, we can form the design matrix and the target vector

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_m^\top \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \ldots & x_{1,d} \\ x_{2,1} & x_{2,2} & \ldots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \ldots & x_{m,d} \end{bmatrix} \in \mathbb{R}^{m \times d} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \in \mathbb{R}^m$$

- We wish to find $\mathbf{w} \in \mathbb{R}^d$ satisfying (or approximately satisfying) the linear system

$$\mathbf{X}\mathbf{w} = \mathbf{y}.$$

Linear Regression (With Offset)

- m: size of the dataset
- d: dimension/length of each feature vector (input)
- y_i: scalar or real-valued target/output (e.g., height, exam marks)

Goal:

- Design a function/model/regressor $f_{\mathbf{w},b}$ as a linear combination of the features in $\mathbf{x}$, i.e.,

$$f_{\mathbf{w},b}(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b,$$

  where $\mathbf{w} \in \mathbb{R}^d$, the unknown, is the d-dimensional weight vector and b is the bias or offset.
- The notation $f_{\mathbf{w},b}$ means that the model is parametrized by two quantities $\mathbf{w}$ and b.
- Note that the model can also be more compactly written as

$$f_{\mathbf{w},b}(\mathbf{x}) = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix}^\top \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}.$$

Objective (Loss) Function in Linear Regression

- We wish to minimize the error e_i between the prediction $f_{\mathbf{w},b}(\mathbf{x}_i)$ and the target, where

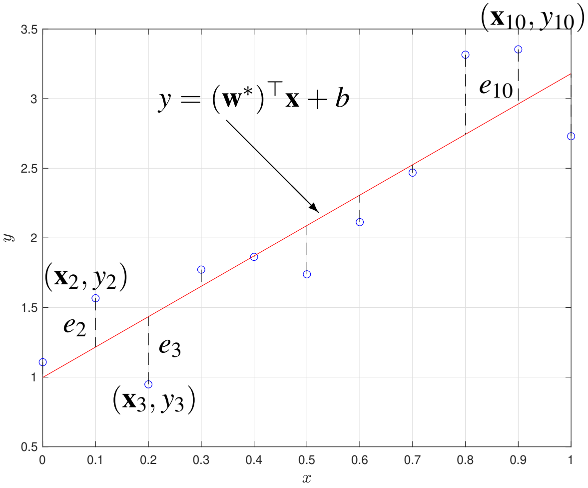
$$e_i = f_{\mathbf{w},b}(\mathbf{x}_i) - y_i$$

- We average the square of the errors over all training samples. This defines the objective or loss function

$$\mathrm{Loss}(\mathbf{w}, b) = \frac{1}{m} \sum_{i=1}^{m} (f_{\mathbf{w},b}(\mathbf{x}_i) - y_i)^2$$

- $\mathrm{Loss}(\mathbf{w}, b)$ is known as the (squared or ℓ_2) loss or objective function
- $(f_{\mathbf{w},b}(\mathbf{x}_i) - y_i)^2$ is also called the per-sample loss or objective function and is a measure of the difference or penalty between the prediction $f_{\mathbf{w},b}(\mathbf{x}_i)$ and the target y_i

Objective (Loss) Function in Linear Regression



Linear Regression: Minimize the sum of squares of the errors e_i, i.e. $\sum_{i=1}^{11} e_i^2$.
Note that here $\mathbf{x}$ is a scalar, but in general $\mathbf{x}$ can be a vector

Objective (Loss) Function in Linear Regression

- Define $\overline{\mathbf{w}} \in \mathbb{R}^{d+1}$ as the $(d+1)$-dimensional vector that concatenates b and $\mathbf{w}$, i.e.,

$$\overline{\mathbf{w}} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}.$$

- Similarly, define $\bar{\mathbf{x}}_i \in \mathbb{R}^{d+1}$ as the $(d+1)$-dimensional vector that concatenates 1 and $\mathbf{x}_i$

$$\overline{\mathbf{x}}_i = \begin{bmatrix} 1 \\ \mathbf{x}_i \end{bmatrix} = \begin{bmatrix} 1 \\ x_{i,1} \\ x_{i,2} \\ \vdots \\ x_{i,d} \end{bmatrix}.$$

Objective (Loss) Function in Linear Regression

- We wish to find $\overline{\mathbf{w}}^* = [b^*, \mathbf{w}^*]^\top \in \mathbb{R}^{d+1}$ that minimizes

$$\overline{\mathbf{w}}^* = \operatorname*{arg\,min}_{\overline{\mathbf{w}}=[b,\mathbf{w}]^\top} \mathrm{Loss}(\mathbf{w}, b)$$

  where the ℓ_2 or squared loss is

$$\mathrm{Loss}(\mathbf{w}, b) = \frac{1}{m} \sum_{i=1}^{m} \left(f_{\mathbf{w},b}(\mathbf{x}_i) - y_i\right)^2$$

- The $1/m$ does not affect the solution so we can choose to include or exclude it.

Objective (Loss) Function in Linear Regression

- Note that
$$f_{\mathbf{w},b}(\mathbf{x}_i) - y_i = \begin{bmatrix} 1 \\ \mathbf{x}_i \end{bmatrix}^\top \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} - y_i = \overline{\mathbf{x}}_i^\top \overline{\mathbf{w}} - y_i,$$
so that
$$\sum_{i=1}^m (f_{\mathbf{w},b}(\mathbf{x}_i) - y_i)^2 = \sum_{i=1}^m \left(\overline{\mathbf{x}}_i^\top \overline{\mathbf{w}} - y_i\right)^2.$$
In other words,
$$\sum_{i=1}^m (f_{\mathbf{w},b}(\mathbf{x}_i) - y_i)^2 = (\mathbf{X}\overline{\mathbf{w}} - \mathbf{y})^\top (\mathbf{X}\overline{\mathbf{w}} - \mathbf{y})$$
- The design matrix is now the $m \times (d+1)$ matrix
$$\mathbf{X} = \begin{bmatrix} \overline{\mathbf{x}}_1^\top \\ \overline{\mathbf{x}}_2^\top \\ \vdots \\ \overline{\mathbf{x}}_m^\top \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{x}_1^\top \\ 1 & \mathbf{x}_2^\top \\ \vdots & \vdots \\ 1 & \mathbf{x}_m^\top \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,d} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{m,1} & x_{m,2} & \dots & x_{m,d} \end{bmatrix} \in \mathbb{R}^{m\times(d+1)}$$

Optimizing the Loss Function in Linear Regression

- The objective function is now simplified to

$$\begin{aligned} J(\overline{\mathbf{w}}) &= (\mathbf{X}\overline{\mathbf{w}} - \mathbf{y})^\top (\mathbf{X}\overline{\mathbf{w}} - \mathbf{y}) \\ &= \overline{\mathbf{w}}^\top \mathbf{X}^\top \mathbf{X}\overline{\mathbf{w}} - \overline{\mathbf{w}}^\top \mathbf{X}^\top \mathbf{y} - \mathbf{y}^\top \mathbf{X}\overline{\mathbf{w}} + \mathbf{y}^\top \mathbf{y} \\ &= \overline{\mathbf{w}}^\top \mathbf{X}^\top \mathbf{X}\overline{\mathbf{w}} - 2\overline{\mathbf{w}}^\top (\mathbf{X}^\top \mathbf{y}) + \mathbf{y}^\top \mathbf{y} \end{aligned}$$

  The terms in blue are the same. Why?
- Differentiating this w.r.t. $\overline{\mathbf{w}}$ (see the rules on slide 14),

$$\nabla_{\overline{\mathbf{w}}} J(\overline{\mathbf{w}}) = 2\mathbf{X}^\top \mathbf{X}\overline{\mathbf{w}} - 2\mathbf{X}^\top \mathbf{y}.$$

- Setting this to zero yields

$$2\mathbf{X}^\top \mathbf{X}\overline{\mathbf{w}}^* = 2\mathbf{X}^\top \mathbf{y}.$$

- If $\mathbf{X}$ has full column rank, $\mathbf{X}^\top \mathbf{X}$ is invertible and

$$\overline{\mathbf{w}}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}.$$

  This is the least squares solution. Is it a global or local minimum?

Least Squares : Training and Prediction

- In summary, given a dataset $(\mathbf{x}_i, y_i)$ for $i = 1, 2, \ldots, m$, form the design matrix and target vector

$$\mathbf{X} = \begin{bmatrix} \overline{\mathbf{x}}_1^\top \\ \overline{\mathbf{x}}_2^\top \\ \vdots \\ \overline{\mathbf{x}}_m^\top \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{x}_1^\top \\ 1 & \mathbf{x}_2^\top \\ \vdots & \vdots \\ 1 & \mathbf{x}_m^\top \end{bmatrix} \in \mathbb{R}^{m\times(d+1)} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \in \mathbb{R}^m$$

- Training/Learning:

$$\overline{\mathbf{w}}^* = \begin{bmatrix} b^* \\ \mathbf{w}^* \end{bmatrix} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}.$$

- Prediction/Testing: Given a new training sample $\mathbf{x}_{\text{new}}$,

$$\hat{y}_{\text{new}} = \begin{bmatrix} 1 \\ \mathbf{x}_{\text{new}} \end{bmatrix}^\top \overline{\mathbf{w}}^* = b^* + \mathbf{x}_{\text{new}}^\top \mathbf{w}^*.$$

Linear Regression: Example 1

- Dataset $(\mathbf{x}_i, y_i), i = 1, 2, 3, 4$ includes the samples

$$\mathbf{x}_1 = -7, \quad \mathbf{x}_2 = -5, \quad \mathbf{x}_3 = 1, \quad \mathbf{x}_4 = 5$$
$$y_1 = -6, \quad y_2 = -4, \quad y_3 = -1, \quad y_4 = 4$$

- Here, $m = 4$ and $d = 1$.
- Design matrix and target vector are

$$\mathbf{X} = \begin{bmatrix} 1 & -7 \\ 1 & -5 \\ 1 & 1 \\ 1 & 5 \end{bmatrix} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} -6 \\ -4 \\ -1 \\ 4 \end{bmatrix}$$

- The linear system $\mathbf{X}\overline{\mathbf{w}} = \mathbf{y}$ is overdetermined and there is no solution for $\overline{\mathbf{w}}$ because

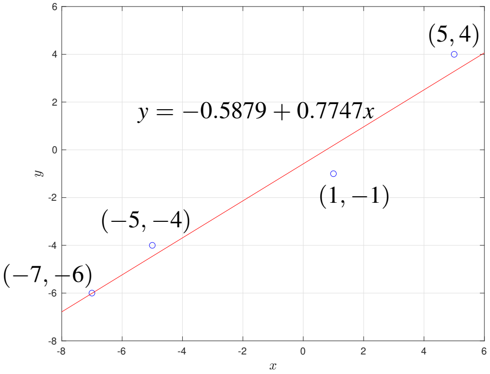
$$\operatorname{rank}(\mathbf{X}) < \operatorname{rank}(\tilde{\mathbf{X}}).$$

Linear Regression: Example 1 (Training)

- Using some numerical software, we can find

$$\overline{\mathbf{w}}^* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} = \begin{bmatrix} -0.5879 \\ 0.7747 \end{bmatrix}$$

- We can plot the points and the least squares line.

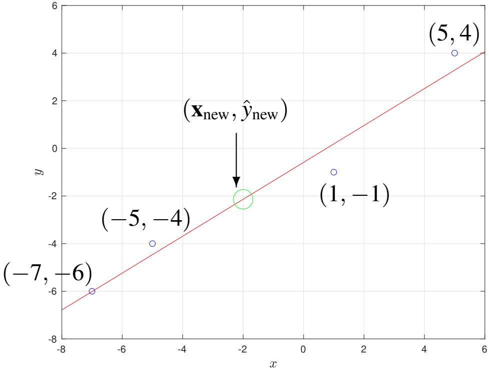


Linear Regression: Example 1 (Prediction)

- Suppose we want to predict the value of y_{new} when $\mathbf{x}_{\text{new}} = -2$. Then we plug $\mathbf{x}_{\text{new}} = -2$ into model to get

$$\hat{y}_{\text{new}} = \begin{bmatrix} 1 \\ \mathbf{x}_{\text{new}} \end{bmatrix}^{\top} \overline{\mathbf{w}}^* = 1 \times (-0.5879) + (-2) \times (0.7747) = -2.1374$$

- Pictorially,



Linear Regression: Example 2

- Now our feature vectors are

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{x}_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad \mathbf{x}_3 = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \quad \mathbf{x}_4 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

and targets are

$$y_1 = 1 \quad y_2 = 0 \quad y_3 = 2 \quad y_4 = -1.$$

- The design matrix and target vector are

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 3 \\ 1 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} 1 \\ 0 \\ 2 \\ -1 \end{bmatrix}.$$

- Note that $3 = \mathrm{rank}(\mathbf{X}) < \mathrm{rank}(\tilde{\mathbf{X}}) = 4$ so the overdetermined system does not have a solution.

Linear Regression: Example 2 (Training & Prediction)

- But we can check that $\mathbf{X}$ has full column rank and so the least squares solution exists and is given by

$$\overline{\mathbf{w}}^* = \begin{bmatrix} b^* \\ \mathbf{w}^* \end{bmatrix} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} = \begin{bmatrix} -0.7500 \\ 0.1786 \\ 0.9286 \end{bmatrix}$$

  This is the training or learning step.

- If we want to make predictions for $\mathbf{x}_{\text{new}} = [0, -1]^\top$, we use the model

$$\hat{y}_{\text{new}} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}^\top \overline{\mathbf{w}}^* = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}^\top \begin{bmatrix} -0.7500 \\ 0.1786 \\ 0.9286 \end{bmatrix} = -1.6786.$$

  This is the prediction step. [Python Demo]

Learning Vector-Valued Linear Functions

- Suppose we want to predict:
  1. Donald Trump's winning margin;
  2. The number of number of house seats won by Republicans;
  3. The number of incumbent governors that retain their governorships.
- Suppose there are h outputs we want to predict (above $h = 3$).
- Given a dataset $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m$ where $\mathbf{x}_i \in \mathbb{R}^d$ (column vector) and $\mathbf{y}_i \in \mathbb{R}^{1\times h}$ (row vector), the model to be used is

$$\underbrace{\begin{bmatrix} y_{1,1} & \dots & y_{1,h} \\ y_{2,1} & \dots & y_{2,h} \\ \vdots & \ddots & \vdots \\ y_{m,1} & \dots & y_{m,h} \end{bmatrix}}_{\mathbf{Y}\in\mathbb{R}^{m\times h}} = \underbrace{\begin{bmatrix} 1 & x_{1,1} & \dots & x_{1,d} \\ 1 & x_{2,1} & \dots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{m,1} & \dots & x_{m,d} \end{bmatrix}}_{\mathbf{X}\in\mathbb{R}^{m\times(d+1)}} \underbrace{\begin{bmatrix} b_1 & b_2 & \dots & b_h \\ w_{1,1} & w_{1,2} & \dots & w_{1,h} \\ \vdots & \vdots & \ddots & \vdots \\ w_{d,1} & w_{d,2} & \dots & w_{d,h} \end{bmatrix}}_{\overline{\mathbf{W}}\in\mathbb{R}^{(d+1)\times h}}$$

- When $h = 1$, this particularizes to standard linear regression.
- This is exactly h separate linear regression problems.

Learning Vector-Valued Linear Functions: Objective

- Our loss function is a generalization of the previous study

$$\mathrm{Loss}(\overline{\mathbf{W}}) = \mathrm{Loss}(\mathbf{W}, \mathbf{b}) = \sum_{k=1}^{h} (\mathbf{X}\overline{\mathbf{w}}_k - \mathbf{y}^{(k)})^\top (\mathbf{X}\overline{\mathbf{w}}_k - \mathbf{y}^{(k)})$$

  where for each $1 \le k \le h$,

$$\overline{\mathbf{w}}_k = \begin{bmatrix} b_k \\ w_{1,k} \\ \vdots \\ w_{d,k} \end{bmatrix} \in \mathbb{R}^{d+1} \quad \text{and} \quad \mathbf{y}^{(k)} = \begin{bmatrix} y_{1,k} \\ y_{2,k} \\ \vdots \\ y_{m,k} \end{bmatrix} \in \mathbb{R}^{m}$$

  are the k-th columns of $\mathbf{W}$ and $\mathbf{Y}$ respectively.

- We are aggregating or summing the contributions of the errors from each of the h prediction tasks.
- Our goal is to find

$$\overline{\mathbf{W}}^* = \arg\min_{\mathbf{W},\mathbf{b}} \mathrm{Loss}(\overline{\mathbf{W}}) \quad \text{where} \quad \overline{\mathbf{W}} = \begin{bmatrix} \mathbf{b}^\top \\ \mathbf{W} \end{bmatrix}.$$

Learning Vector-Valued Linear Functions: Training

- Objective:

$$\overline{\mathbf{W}}^* = \arg\min_{\mathbf{W},\mathbf{b}} \mathrm{Loss}(\overline{\mathbf{W}}) \quad \text{where} \quad \overline{\mathbf{W}} = \begin{bmatrix} \mathbf{b}^\top \\ \mathbf{W} \end{bmatrix}.$$

- By differentiating with respect to each column $\overline{\mathbf{w}}_k$ and setting the result to zero, we find that the least squares solution is

$$\overline{\mathbf{W}}^* = \begin{bmatrix} \overline{\mathbf{w}}_1^* & \overline{\mathbf{w}}_2^* & \ldots & \overline{\mathbf{w}}_h^* \end{bmatrix} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y} \in \mathbb{R}^{(d+1)\times h}.$$

  This will be an exercise in a tutorial.
- In this new setting, what condition does $\mathbf{X}$ have to satisfy for $\overline{\mathbf{W}}^*$ to exist?
- We need $(\mathbf{X}^\top\mathbf{X})^{-1}$ to exist, which means that $\mathbf{X}$ has to have full column rank.

Learning Vector-Valued Linear Functions: Prediction

- Given a dataset $\{(\mathbf{x}_i, \mathbf{y}_i)\}$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $\mathbf{y}_i \in \mathbb{R}^{1\times h}$ and $1 \le i \le m$, we can use the above procedure to learn the least squares solution

$$\overline{\mathbf{W}}^* = \begin{bmatrix} \overline{\mathbf{w}}_1^* & \overline{\mathbf{w}}_2^* & \dots & \overline{\mathbf{w}}_h^* \end{bmatrix} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y} \in \mathbb{R}^{(d+1)\times h}.$$

- Given a new sample $\mathbf{x}_{\text{new}} \in \mathbb{R}^d$, the predictions are contained in the row vector

$$\hat{\mathbf{y}}_{\text{new}} = \begin{bmatrix} 1 \\ \mathbf{x}_{\text{new}} \end{bmatrix}^\top \overline{\mathbf{W}}^* \in \mathbb{R}^{1\times h}$$

Linear Regression: Example 3

- Now our feature vectors are

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{x}_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad \mathbf{x}_3 = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \quad \mathbf{x}_4 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

  and targets are

$$\mathbf{y}_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad \mathbf{y}_2 = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad \mathbf{y}_3 = \begin{bmatrix} 2 & -1 \end{bmatrix} \quad \mathbf{y}_4 = \begin{bmatrix} -1 & 3 \end{bmatrix}$$

- Here, $m = 4$, $d = 2$, $h = 2$.
- The design matrix and target matrix are

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 3 \\ 1 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{Y} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 2 & -1 \\ -1 & 3 \end{bmatrix}.$$

- Note that the first regression problem here (corresponding to the first components of each $\mathbf{y}_i$) is exactly the same as that in Linear Regression Example 2 on Slide 31.

Linear Regression: Example 4 (Training & Prediction)

- We have already checked that $\mathbf{X}$ has full column rank. Hence, the least squares solution is

$$\overline{\mathbf{W}}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y} = \begin{bmatrix} -0.7500 & 2.2500 \\ 0.1786 & 0.0357 \\ 0.9286 & 1.2143 \end{bmatrix} \in \mathbb{R}^{(d+1)\times h}.$$

- Now, someone gave us a new sample $\mathbf{x}_{\text{new}} = [0, -1]^\top$. The predicted output is

$$\hat{\mathbf{y}}_{\text{new}} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}^\top \overline{\mathbf{W}}^* = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}^\top \begin{bmatrix} -0.7500 & 2.2500 \\ 0.1786 & 0.0357 \\ 0.9286 & 1.2143 \end{bmatrix} = \begin{bmatrix} -1.6786 & 3.4643 \end{bmatrix}$$

  The first prediction -1.6786 corresponds to that in Linear Regression Example 2 on Slide 32.

- This is the **prediction** step. [Python Demo]

Summary

- (Learning/Training) Given a dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, the least squares solution (with offset) is

$$\overline{\mathbf{w}}^* = \begin{bmatrix} b^* \\ \mathbf{w}^* \end{bmatrix} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \in \mathbb{R}^{d+1}$$

where

$$\mathbf{X} = \begin{bmatrix} \overline{\mathbf{x}}_1^\top \\ \overline{\mathbf{x}}_2^\top \\ \vdots \\ \overline{\mathbf{x}}_m^\top \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{x}_1^\top \\ 1 & \mathbf{x}_2^\top \\ \vdots & \vdots \\ 1 & \mathbf{x}_m^\top \end{bmatrix} \in \mathbb{R}^{m \times (d+1)} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \in \mathbb{R}^m.$$

- (Prediction/Testing) Given a new feature vector (sample, example) $\mathbf{x}_{\text{new}}$, the prediction based on the least squares solution is

$$\hat{y}_{\text{new}} = \begin{bmatrix} 1 \\ \mathbf{x}_{\text{new}} \end{bmatrix}^\top \overline{\mathbf{w}}^* = b^* + \mathbf{x}_{\text{new}}^\top \mathbf{w}^*.$$