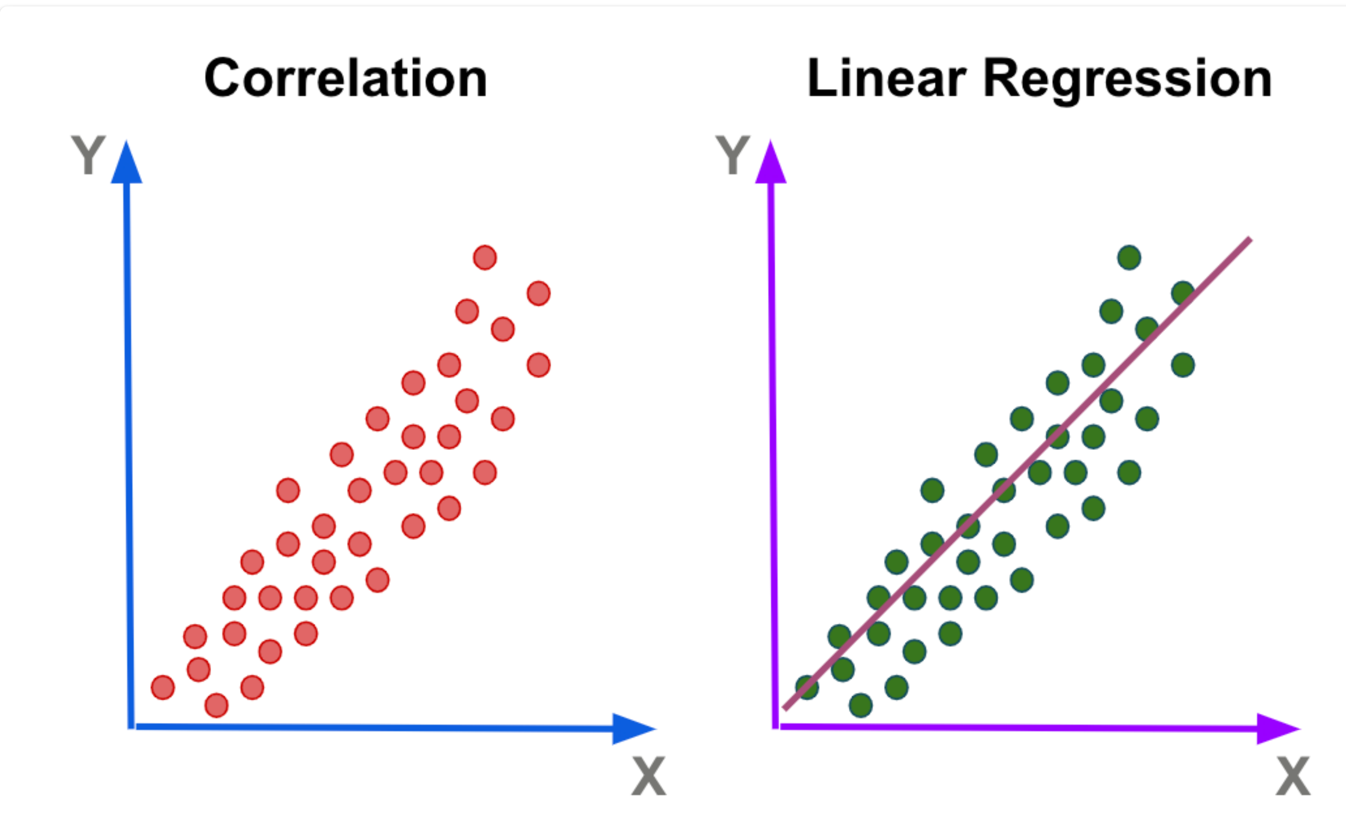
Noting Paradigm

- x - Plain text: Scalar.
- $\mathbf{x}$ - Bold-Face lowercase: Vector of scalars.
  - e.g., $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_D \end{bmatrix}$, where $\mathbf{x} \in \mathbb{R}^D$
- $\mathbf{X}$ - Bold-Face uppercase: Set of vectors.
  - e.g., $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, where $\mathbf{X} \subset \mathbb{R}^D$ and $|\mathbf{X}| = N$.

8.0 Regression

8.0.0 Why Regression?



Problem Setup

Given

- A set of inputs:
  - $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, where, for each input:
    - $\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \cdots \\ x_{iD} \end{bmatrix} \in \mathbb{R}^D$
- A set of corresponding ground-truth outputs:
  - $\mathbf{y} = \{y_1, y_2, \cdots, y_N\}$, where, for each output:
    - $y_i \in \mathbb{R}$
- A labelling relation:
  - $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$
  Goal
- Other than the given input-output relations, we want to know:
  - How the output goes when an unseen input is given.
  - This is the original purpose of regression.
- Therefore, we want to learn a mapping
  - $f(\mathbf{x}) : \mathbb{R}^D \mapsto \mathbb{R}$, where
    - if we input an unseen vector $\mathbf{x} \notin \mathbf{X}$,
    - it could output a scalar $y \notin \mathbf{y}$.
  - such that we could make a prediction over an unseen data based on the given input-output pairs.

Parametric/Nonparametric Regression

- Parametric Regression 参数性回归
  - Assume a functional form for $f(x)$.
- Nonparametric Regression 非参数性回归
  - Does not assume a functional form for $f(x)$.

To sum up

- From given relations, learn a function that
  - Takes a vector as an input
  - Output a real number that
    - "Fits" the given pattern

8.0.1 Definition

What is Regression?

- Regression aims at modelling the dependence of:
  - a response Y,
  - on a covariate X.
- That is, to predict the value of one or more continuous target variables y given the value of input vector x.

The regression model is described by

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

- where the dependence of an estimated response y on a covariate $\mathbf{x}$ is captured via:
  - $p(y|\mathbf{x})$,
  - i.e., a conditional probability distribution.

Conditional Mean of a regression function

- Considering the Mean Squared Error, we find the MMSE estimate:
  - $\mathcal{E}(f) = \mathbb{E}(y - f(\mathbf{x}))^2$
    - $= \int\int \cdots \int (y - f(\mathbf{x}))^2 \cdot p(\mathbf{x}, y)\ d\mathbf{x}dy$
    - $= \int\int \cdots \int (y - f(\mathbf{x}))^2 \cdot p(\mathbf{x}) \cdot p(y|\mathbf{x})d\mathbf{x}y$
    - $= \int \cdots \int p(\mathbf{x}) \cdot \int \left[\left(y - f(\mathbf{x}) \right)^2 \cdot p(y|\mathbf{x})dy \right] d\mathbf{x}$

      (Every dimension is integrated)
  - Therefore, we need to minimize:
    - $\int \left(y - f(x) \right)^2 \cdot p(y|\mathbf{x})dy$
      - $\implies \dfrac{\partial}{\partial f(\mathbf{x})} \int \left[\left(y - f(x) \right)^2 \cdot p(y|\mathbf{x})dy \right] = 0$
      - $\implies f(x) = \int y \cdot p(y|\mathbf{x})\ dy = \mathbb{E}[y|\mathbf{x}]$
- That is to say,
  - $f(x)$, our estimation on a given input $\mathbf{x}$ we wanted is the Conditional Mean of y given covariate $\mathbf{x}$.

8.1 Linear Regression

8.1.0 Affine Function

(Additional) What is an affine function (仿射函数)?

- A function that:
  - Takes a vector input, and
  - Outputs a scalar.

- i.e., $f : \mathbb{R}^N \mapsto \mathbb{R}$ is a general form of an affine function.
- More generally, an "affine transformation" (仿射变换) denotes:
  - $\mathbb{R}^n \mapsto \mathbb{R}^m$
    - Turning an n-d vector to an m-d one.
  - $\mathbf{x} \mapsto A\mathbf{x} + b$ is a more general description of an affine transformation, where
    - A is an $m \times n$ matrix, and
    - b is an m-d vector.
  - When $m = 1$, the affine transformation denotes an affine function.

8.1.1 What it looks like

Focusing on a specific sample

Given an input vector $\mathbf{x}$:

- We give an estimation $\hat{y} = f(\mathbf{x})$, where $f(\mathbf{x})$ is the conditional mean.
- In linear regression, this conditional mean $f(\mathbf{x})$ is an affine function of $\mathbf{x}$.
  - For each input vector $\mathbf{x} \in \mathbf{X}$, we design $M + 1$ operations, described in basic functions.
  - Each operation (i.e., basic functions) takes $\mathbf{x}$ as an input, and outputs a scalar.
  - We produces the linear combination of all the scalar outputs with learnable weights.

The linear regression formula is given below:

$$\hat{y} = f(\mathbf{x}) = \Big[w_1\phi_1(\mathbf{x}) + w_2\phi_2(\mathbf{x}) + \cdots + w_M\phi_M(\mathbf{x})\Big] + w_0\phi_0(\mathbf{x})$$

- $= \sum_{j=1}^{M} w_j\phi_j(\mathbf{x}) + w_0\phi_0(\mathbf{x})$
- $= \begin{bmatrix} w_0 & w_1 & \cdots & w_M \end{bmatrix} \begin{bmatrix} \phi_0(\mathbf{x}) \\ \phi_1(\mathbf{x}) \\ \cdots \\ \phi_M(\mathbf{x}) \end{bmatrix}$
- $= \mathbf{w}^\top \phi(\mathbf{x})$

where,

- $M + 1$ is the number of operations.
- $\mathbf{w}$ is the weight vector:

- $\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \cdots \\ w_M \end{bmatrix}$.

- ϕ is the basic function vector:
  - $\phi = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \cdots \\ \phi_M \end{bmatrix}$.

Focusing all the test samples

In compact form, we have:

$$\hat{\mathbf{y}} = \mathbf{\Phi}^\top \mathbf{w}$$

Namely, the above compact form describes:

$$\begin{bmatrix} \hat{y_1} \\ \hat{y_2} \\ \vdots \\ \hat{y_N} \end{bmatrix} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_M(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_M(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_M(\mathbf{x}_N) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_M \end{bmatrix}$$

where,

- $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N \in \mathbf{X}$.
- $\mathbf{w} = \begin{bmatrix} w_1 \\ x_2 \\ \vdots \\ w_M \end{bmatrix} \in \mathbb{R}^{M+1}$ is the $M+1$-d weight vector over the designed $M+1$ operations, i.e., basic functions.
- $\mathbf{\Phi} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_0(\mathbf{x}_2) & \cdots & \phi_0(\mathbf{x}_N) \\ \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \cdots & \phi_1(\mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_M(\mathbf{x}_1) & \phi_M(\mathbf{x}_2) & \cdots & \phi_M(\mathbf{x}_N) \end{bmatrix}$ is the $M \times N$ design matrix.

8.1.2 Many Ways to Design Basic Functions ϕ

Polynomial Regression

- $\forall j \in [0, M],\ \phi_j(\mathbf{x}) = \mathbf{x}^j$.

Gaussian Basis Functions

- $\forall j \in [0, M],\ \phi_j(\mathbf{x}) = e^{-\dfrac{\|\mathbf{x}-\mu_\mathbf{j}\|^2}{2\sigma^2}}$

Spline Basis Functions

- Piecewise polynomials.

8.1.3 Many Ways to Learn Weights $\mathbf{w}$

Once the basic functions $\phi = \begin{bmatrix} \phi_0 \\ \phi_1 \\ \cdots \\ \phi_M \end{bmatrix}$ are decided, we would proceed to learn the weights $\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \cdots \\ w_M \end{bmatrix}$.

8.2 Learn $\mathbf{w}$ with Least Squares

8.2.1 Ordinary Least Squares

Given

- A set of inputs:
  - $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, where, for each input:
    - $\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \cdots \\ x_{iD} \end{bmatrix} \in \mathbb{R}^D$
- A vector of corresponding ground-truth outputs:
  - $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_N \end{bmatrix}$, where, for each output:
    - $y_i \in \mathbb{R}$
- A labelling relation:
  - $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$

- A set of designed basic functions:

- $\phi = \begin{bmatrix} \phi_0 \\ \phi_1 \\ \cdots \\ \phi_M \end{bmatrix}$.

Do

Learn a weight vector $\mathbf{w} \in \mathbb{R}^{M+1}$ which minimizes the sum squared error:

$$\mathcal{J}_{LS}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 = \frac{1}{2}\sum_{i=1}^{N}\left[y_i - \mathbf{w}^\top \phi(\mathbf{x}_i)\right]^2 = \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}^\top \mathbf{w}\|_2^2$$

Steps to find the optimal $\mathbf{w}$

To find such $\mathbf{w}$, we need to solve:

$$\frac{\partial}{\partial \mathbf{w}}\mathcal{J}_{LS}(\mathbf{w}) = \frac{\partial}{\partial \mathbf{w}}\left(\frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}^\top \mathbf{w}\|_2^2\right) = 0$$

By expanding the squared l-2 norm of the summed squared error, we get

- $\mathcal{J}_{LS}(\mathbf{w}) = \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}^\top \mathbf{w}\|_2^2$
  - $= \frac{1}{2}(\mathbf{y} - \mathbf{\Phi}^\top \mathbf{w})^\top(\mathbf{y} - \mathbf{\Phi}^\top \mathbf{w})$
  - $= \frac{1}{2}(\mathbf{y}^\top - \mathbf{w}^\top \mathbf{\Phi})(\mathbf{y} - \mathbf{\Phi}^\top \mathbf{w})$
  - $= \frac{1}{2}\left(\mathbf{y}^\top \mathbf{y} - \mathbf{y}^\top \mathbf{\Phi}^\top \mathbf{w} - \mathbf{w}^\top \mathbf{\Phi} \mathbf{y} + \mathbf{w}^\top \mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{w}\right)$

Then, we have,

- $\frac{\partial}{\partial \mathbf{w}}\mathcal{J}_{LS}(\mathbf{w})$
  - $= \frac{\partial}{\partial \mathbf{w}}(\frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}^\top \mathbf{w}\|_2^2)$
  - $= \frac{1}{2}(0 - y^\top \mathbf{\Phi}^\top - \mathbf{\Phi}\mathbf{y} + 2\mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{w})$ (The 2nd and 3rd term are same)
  - $= \frac{1}{2}(0 - 2\mathbf{\Phi}\mathbf{y} + 2\mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{w})$
  - $= \mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{w} - \mathbf{\Phi}\mathbf{y}$

Therefore, to find the optimal $\mathbf{w}$,

- $\frac{\partial}{\partial \mathbf{w}}\mathcal{J}_{LS}(\mathbf{w}) = 0$
  - $\implies \frac{\partial}{\partial \mathbf{w}}(\frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}^\top \mathbf{w}\|_2^2) = 0$

- $\implies \mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{w} - \mathbf{\Phi}\mathbf{y} = 0$
- ★ $\implies \mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{w} = \mathbf{\Phi}\mathbf{y}$

Thus, the Least-Squares estimate of weight vector $\mathbf{w}$ is given by:

$$\mathbf{w}_{LS} = \left(\mathbf{\Phi}\mathbf{\Phi}^\top\right)^{-1} \mathbf{\Phi}\mathbf{y} = \mathbf{\Phi}^\dagger \mathbf{y}$$

where $\mathbf{\Phi}^\dagger$ is the Moore-Penrose pseudo-inverse of $\mathbf{\Phi}$ (伪逆矩阵).

8.2.2 Probabilistic Least Squares with MLE

We know that, for linear regression, the estimation $f(x)$ is given by

$$\hat{y_k} = f = (\mathbf{x}_k) = \mathbf{w}^\top \phi(\mathbf{x}_k)$$

To get each output data sample, we add a Gaussian noise:

$$\hat{y_k} = \mathbf{w}^\top \phi(\mathbf{x}_k) + \epsilon_k$$

where,

$$\forall k = 1, \cdots, N, \ \epsilon_n \sim \mathcal{N}(0, \sigma^2 I)$$

In compact form, we have:

$$\hat{\mathbf{y}} = \mathbf{\Phi}^\top \mathbf{w} + \epsilon$$

Namely,

$$\begin{bmatrix} \hat{y_1} \\ \hat{y_2} \\ \vdots \\ \hat{y_N} \end{bmatrix} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_M(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_M(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_M(\mathbf{x}_N) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}$$

which allows us to model $\mathbf{y}$ as:

$$p(\mathbf{y}|\mathbf{\Phi}, \mathbf{w}) = \mathcal{N}(\mathbf{\Phi}^\top \mathbf{w}, \sigma^2 I)$$

Then, we get the log-likelihood:

$$L(\mathbf{\Phi}, \mathbf{w}) = \ln p(\mathbf{y}|\mathbf{\Phi}, \mathbf{w})$$

- $= \sum_{k=1}^{N} \ln p(y_k|\phi(\mathbf{x}_k), \mathbf{w})$
- $= \cdots$
- $= -\dfrac{N}{2} \ln \sigma^2 - \dfrac{N}{2} \ln 2\pi - \dfrac{\mathcal{J}_{LS}}{\sigma^2}$

Since MLE is given by

$$\mathbf{w}_{ML} = argmax_w \ln p(\mathbf{y}|\mathbf{\Phi}, \mathbf{w})$$

we could say that

$$\mathbf{w}_{WL} = \mathbf{w}_{LS}$$

yielding the Gaussian Noise Assumption.