9/7/2021

AI 1104

- Analyse two SL models:
  - (i) Linear regression
  - (ii) Artificial neural network

Supervised Learning (SL): $D = \{(x_1, y_1), (x_2, y_2) \ldots (x_N, y_N)\}$ (Entire dataset)

$T = \{(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)\}$ (Training dataset)

$V = \{(x_{n+1}, y_{n+1}), \ldots (x_N, y_N)\}$ (Test/Val. dataset)

(x, y) ← x: input point, y: Label

model: $\hat{y} = f(x, \theta)$ → θ: model parameters, x: input

$\hat{y}$: predicted label

Loss: $R(\theta) = \sum_{i=1}^{n} R_i(\theta) = \sum_{i=1}^{n} d(y_i, \hat{y}_i) = \sum_{i=1}^{n} d(y_i, f(x_i; \theta))$

$R(\theta)$: cumulative loss over training samples

Goal: Find $\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}} R(\theta)$

$d: \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ → have $d(\cdot)$ be a valid distance; $\mathbb{R}_+$: set of non-negative reals

$\mathcal{Y}$: set of values that our label can take

(i) Linear regression: Assume $x, y \in \mathbb{R}$, $\hat{y} = \beta_0 + \beta_1 x = f(x; \theta)$ where $\theta = \{\beta_0, \beta_1\}$ ← linear model

$R_i(\theta) = (y_i - \hat{y}_i)^2 = (y_i - (\beta_0 + \beta_1 x_i))^2$

i is the sample index

$\Rightarrow R(\theta) = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2$ — (a) $\theta = \{\beta_0, \beta_1\}$

$= \| X\beta - y \|_2^2$ → X: matrix, β, y: vectors

Where $\underline{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$, $\underline{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$; $X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$

$\underline{\beta}$: 2×1 ; $\underline{y}$: $n \times 1$; X: $n \times 2$

Note: $\hat{\underline{y}} = X \cdot \underline{\beta}$

$$\theta^* = \beta^* = \underset{\beta \in \mathbb{H}}{\operatorname{argmin}} \; \| X \cdot \underline{\beta} - \underline{y} \|_2^2$$

$$\|\underline{x}\|_2 = \left(\sum_{i=1}^{m} x_i^2 \right)^{1/2}$$

$\|\cdot\|_2$: l_2 norm of vector $\underline{x}$

If $\underline{e} = \underline{y} - \hat{\underline{y}}$, then our goal is to min. the l_2 norm of $\underline{e}$.

$$R(\theta) = \| X \cdot \underline{\beta} - \underline{y} \|^2 = \underline{e}^T \underline{e} = (X \cdot \underline{\beta} - \underline{y})^T \cdot (X\underline{\beta} - \underline{y})$$

$$\Rightarrow R(\theta) = \underline{\beta}^T X^T \cdot X \underline{\beta} - 2 \cdot \underline{\beta}^T X^T \underline{y} + \underline{y}^T \underline{y}$$

Q: Let $\underline{a}, \underline{b} \in \mathbb{R}^n$, is $\underline{a}^T \underline{b} = \underline{b}^T \underline{a}$?

$$\nabla_\theta R(\theta) = 0 \iff \nabla_\beta \left(\underline{\beta}^T X^T X \underline{\beta} - 2 \underline{\beta}^T X^T \underline{y} + \underline{y}^T \underline{y} \right) = 0$$

$$2 \cdot X^T X \underline{\beta} - 2 X^T \underline{y} = 0 \quad \text{(verify offline)}$$

$$X^T X \underline{\beta} = X^T \underline{y} \quad -(1)$$

If $X^T X$ is a positive definite matrix ($\Rightarrow X^T X^{-1}$ exists),

then $\underline{\beta}^* = (X^T X)^{-1} \cdot X^T \underline{y}$ $-(2)$

$(X^T X)^{-1} \cdot X^T$: Moore Penrose inverse of X ; Pseudo-inverse of X.

We can arrive at (1) using scalar or regular differentiation operation.

Specifically $\frac{\partial}{\partial \beta_0} R(\theta)$; $\frac{\partial}{\partial \beta_1} R(\theta)$. Try this as an exercise.

Given: $\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_m x^m = f(x;\theta)$; $x, y \in \mathbb{R}$

Q: Find $\theta^* = \beta^*$; What is $d(y, \hat{y})$?

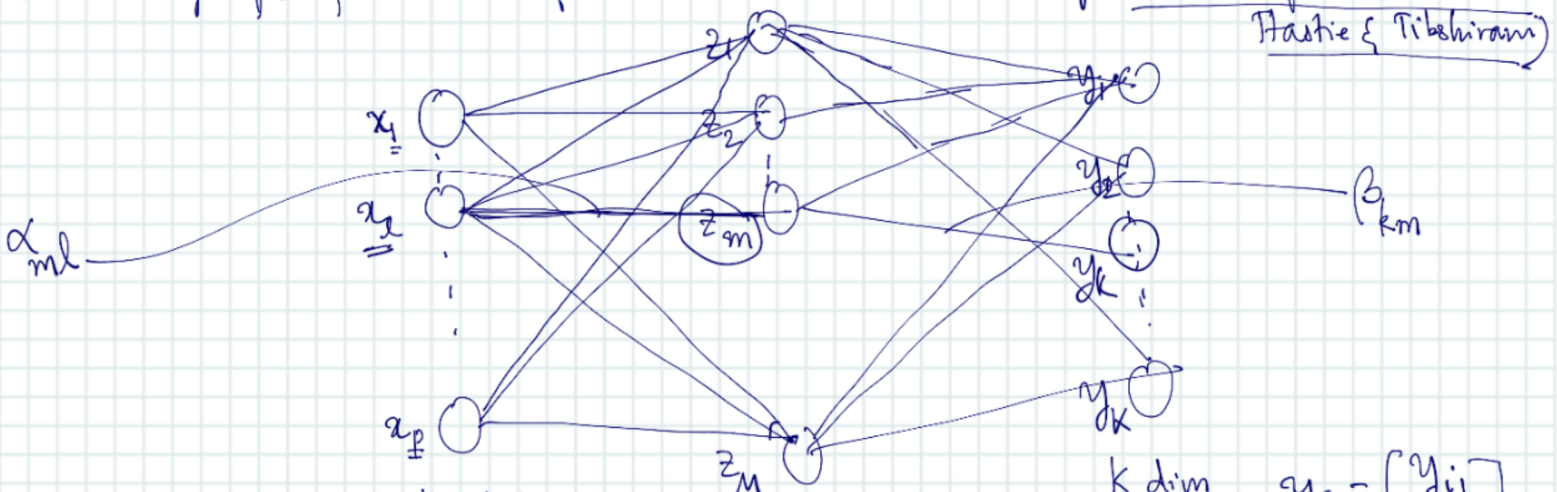
A: $\boxed{\beta^* = (X^T X)^{-1} \cdot X^T y}$ where $X = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^m \\ 1 & & & & \\ & & & & \\ 1 & & & & x_n^m \end{bmatrix}$ $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}$ $m \times 1$

Identical solution as before.

$y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$ $n \times m$

$n \times 1$

(ii) Multilayer Perceptron or Artificial Neural Network (Following Elements of Statistical Learning Hastie & Tibshirani)



p input

M hidden node

K dim output

$x_i = \begin{bmatrix} x_{i1} \\ \vdots \\ x_{ip} \end{bmatrix}$ ↓ data index

$y_i = \begin{bmatrix} y_{i1} \\ \vdots \\ y_{ik} \end{bmatrix}$