Generalization of least square method (廣義最小平方法)

加減 Plus & Minus

2020.2

1 機器學習情境回顧

給定已知 $N+1$ 筆，$M+K$ 維"連續變數"資料集， $\mathcal{D} := \left\{ \left(\underbrace{x_1^{(n)}, x_2^{(n)}, ...x_M^{(n)}}_{\vec{x}^{(n)}}, \underbrace{y_1^{(n)}, y_2^{(n)}, ..., y_K^{(n)}}_{\vec{y}^{(n)}} \right) \right\}_{n=1}^{N+1}$

- 切割資料集

$$\forall\ \sigma \in \mathcal{S}_{N+1}, \mathcal{D} = \mathcal{D}_\sigma^{\text{train}} \cup \mathcal{D}_\sigma^{\text{valid}},$$

  – 本文只考慮 Leave-One-Out：$|\mathcal{D}_\sigma^{\text{train}}| = N, |\mathcal{D}_\sigma^{\text{valid}}| = 1$ 即 $\sigma \in S_{N+1}^{Loo}, |S_{N+1}^{Loo}| = C_N^{N+1} = N+1$

- 只根據 $\mathcal{D}_\sigma^{\text{train}}$，可建構迴歸模型 $F_\sigma : \mathbb{R}^M \longrightarrow \mathbb{R}^K$，並同時使用 $\mathcal{D}_\sigma^{\text{train}}, \mathcal{D}_\sigma^{\text{valid}}$ 來評估建模成效

$$\text{終極目標: } \forall \sigma \in \mathcal{S}_{N+1}, \forall n = 1, 2, ..., N+1 \quad \underbrace{F_\sigma(\vec{x}^{(n)})}_{\text{predicted}} \approx \underbrace{\vec{y}^{(n)}}_{\text{target}}$$

- 根據多維向量定義，可以把單一模型 $F : \mathbb{R}^M \longrightarrow \mathbb{R}^K$ 問題，想成獨立K個模型 $f : \mathbb{R}^M \longrightarrow \mathbb{R}$

$$F_\sigma(\vec{x}^{(n)}) = \begin{bmatrix} f_1(\vec{x}^{(n)}) \\ f_2(\vec{x}^{(n)}) \\ \cdots \\ f_K(\vec{x}^{(n)}) \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_K \end{bmatrix} = \vec{y}^{(n)}$$

  – 於是只需研究 $f(\vec{x}^{(n)}) \approx y^{(n)} \in \mathbb{R}$ 迴歸問題如何建模 !!

  – 損失函數(Loss function)越小越好概念:

$$f(\vec{x}^{(n)}) \approx y^{(n)} \Longrightarrow \left(f(\vec{x}^{(n)}) - y^{(n)} \right)^2 \approx 0$$

$$\underset{\text{overall data}}{\Longrightarrow} \underbrace{\frac{1}{N} \sum_{n=1}^{N} \left(f(\vec{x}^{(n)}) - y^{(n)} \right)^2}_{\text{training\_loss}(\mathcal{D}_\sigma^{\text{train}})} \approx 0, \underbrace{\left(f(\vec{x}^{(N+1)}) - y^{(N+1)} \right)^2}_{\text{validation\_loss}(\mathcal{D}_\sigma^{\text{valid}})} \approx 0$$

  – 交叉驗證(cross validation)最大誤差越小概念:

$$\underset{\text{overall } \sigma}{\Longrightarrow} \max_\sigma \text{training\_loss}(\mathcal{D}_\sigma^{\text{train}}) \approx 0, \max_\sigma \text{validation\_loss}(\mathcal{D}_\sigma^{\text{valid}}) \approx 0$$

2 模型假設 = 參數化 + 線性組合假設

- $f(\vec{x}^{(n)}) \overset{\text{引入模型假設}}{\Longrightarrow} f(\vec{w}, \vec{x}^{(n)}) := \sum_{b\in\mathcal{B}} w_b \cdot g_b(\vec{x}^{(n)}) \in Span\Big\{g_b\Big\}_{b\in\mathcal{B}} =: Span\, g_{\mathcal{B}}$ (linear combinations of given basis $g_{\mathcal{B}}$)
- 常見 basis $g_{\mathcal{B}}, (M = 1)$
  - Simple Intepolation:
  $$g_{\mathcal{B}} := \Big\{1, x_1, x_1^2, ...x_1^{N-1}\Big\}$$
  - Special Functions:
  $$g_{\mathcal{B}} := \text{ Hermites , Chebyshevs , Legendres , Laguerres , Bessels ...}$$
  - Fourier Series:
  $$g_{\mathcal{B}} := \Big\{e^{ikx_1}\Big\}_{k\in\mathbb{Z}}$$
  - ODE(微分方程)
  $$g_{\mathcal{B}} := \Big\{e^{\lambda x_1}\Big\}_{\lambda\in\text{eigenvalues}}$$
- 常見 basis $g_{\mathcal{B}}, (M \geq 1)$:
  - Linear Regression:
  $$g_{\mathcal{B}} := \Big\{1, x_1, x_2, ...x_M\Big\}$$
  - Response Surface Methodology:
  $$g_{\mathcal{B}} := \Big\{1, \underbrace{x_1, x_2, .., x_M}_{\text{first-order } M \text{ terms}}, \underbrace{x_1^2, ..., x_M^2, x_1x_2, x_1x_3, ...x_{M-1}x_M}_{\text{second-order } M^2 \text{ terms (features interaction)}}\Big\}$$
  $$\Longrightarrow$$
  - DIY or data transform by domain knowledge ...

3 核心推導

- 計算 $\vec{w}^*$ 使得 training_loss($\vec{w}^*, \mathcal{D}_\sigma^{\text{train}}$) 最小，則必須滿足 first-order optimality condition

$$\frac{\partial}{\partial w_b}\left[\frac{1}{2}\sum_{n=1}^{N}\left(f(\overbrace{w_b, w_{-b}}^{\vec{w}}, \vec{x}^{(n)}) - y^{(n)}\right)^2\right] = \sum_{n=1}^{N}\left[\frac{1}{2}\cdot\frac{\partial}{\partial w_b}\left(f(\overbrace{w_b, w_{-b}}^{\vec{w}}, \vec{x}^{(n)}) - y^{(n)}\right)^2\right] = 0$$

$$\Longrightarrow \sum_{n=1}^{N}\frac{1}{2}\times 2\left(f(\vec{w}, \vec{x}^{(n)}) - y^{(n)}\right)\cdot g_b(\vec{x}^{(n)}) = \sum_{n=1}^{N}\left(\sum_{b'\in\mathcal{B}} w_{b'} g_{b'}(\vec{x}^{(n)}) - y^{(n)}\right)\cdot g_b(\vec{x}^{(n)}) = 0$$

$$\Longrightarrow \bigwedge_{b\in\mathcal{B}}\left\{\sum_{b'\in\mathcal{B}}\sum_{n=1}^{N} g_{b'}(\vec{x}^{(n)}) g_b(\vec{x}^{(n)}) w_{b'} = \sum_{n=1}^{N} y^{(n)}\cdot g_b(\vec{x}^{(n)})\right\}$$

$$\equiv \underbrace{\left[\sum_{n=1}^{N}\phi^{(n)}{\phi^{(n)}}^T\right]\vec{w} = \left[\sum_{n=1}^{N} y^{(n)}\phi^{(n)}\right]}_{\text{Matrix}(|\mathcal{B}|\times|\mathcal{B}|)\text{-Vector}(|\mathcal{B}|\times 1)\text{ Multiplication}} \qquad \text{where } \phi^{(n)} := [g_b(\vec{x}^{(n)})]_{b\in\mathcal{B}} \text{ is column vector (also called kernel !!)}$$

- analytic optimal solution :

$$\vec{w}^* = \left[\sum_{n=1}^{N}\phi^{(n)}{\phi^{(n)}}^T\right]^{-1}\left[\sum_{n=1}^{N} y^{(n)}\phi^{(n)}\right]$$

- 使用 Sherman-Morrison formula 高效率計算反矩陣

$$\begin{cases} A_1^{-1} = \left(\phi^{(1)}{\phi^{(1)}}^T\right)^{-1} & n = 1 \\ A_{n+1}^{-1} = \left[A_n + \phi^{(n+1)}{\phi^{(n+1)}}^T\right]^{-1} = A_n^{-1} - \frac{A_n^{-1}\phi^{(n+1)}{\phi^{(n+1)}}^T A_n^{-1}}{1+{\phi^{(n+1)}}^T A_n^{-1}\phi^{(n+1)}} & n \geq 2 \end{cases}$$