

監督式學習參考手冊

Afshine AMIDI 和 Shervine AMIDI

December 15, 2019

翻譯: *kevingo*. 審閱: 詹志傑.

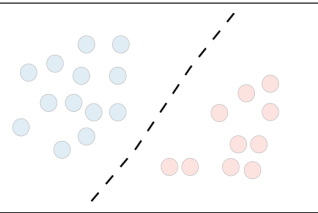
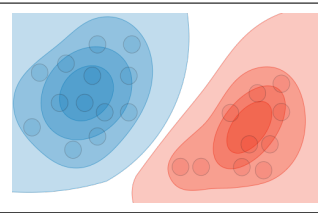
監督式學習介紹

給定一組資料點 $\{x^{(1)}, \dots, x^{(m)}\}$ ，以及對應的一組輸出 $\{y^{(1)}, \dots, y^{(m)}\}$ ，我們希望建立一個分類器，用來學習如何從 x 來預測 y

□ **預測的種類** – 根據預測的種類不同，我們將預測模型分為底下幾種：

| | 迴歸 | 分類器 |
|----|------|---------------------------|
| 結果 | 連續 | 類別 |
| 範例 | 線性迴歸 | 邏輯迴歸, 支援向量機(SVM), 單純貝式分類器 |

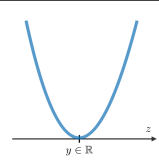
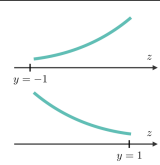
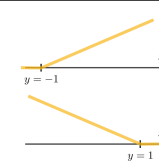
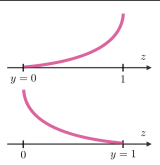
□ **模型種類** – 不同種類的模型歸納如下表：

| | 判別模型 | 生成模型 |
|------|---|---|
| 目標 | 直接估計 $P(y x)$ | 先估計 $P(x y)$ ，然後推論出 $P(y x)$ |
| 學到什麼 | 決策分界線 | 資料的機率分佈 |
| 示意圖 |  |  |
| 範例 | 迴歸, 支援向量機(SVM) | 高斯判別分析(GDA), 單純貝氏(Naive Bayes) |

符號及一般概念

□ **假設** – 我們使用 h_θ 來代表所選擇的模型，對於給定的輸入資料 $x^{(i)}$ ，模型預測的輸出是 $h_\theta(x^{(i)})$

□ **損失函數** – 損失函數是一個函數 $L: (z, y) \in \mathbb{R} \times Y \mapsto L(z, y) \in \mathbb{R}$ ，目的在於計算預測值 z 和實際值 y 之間的差距。底下是一些常見的損失函數：

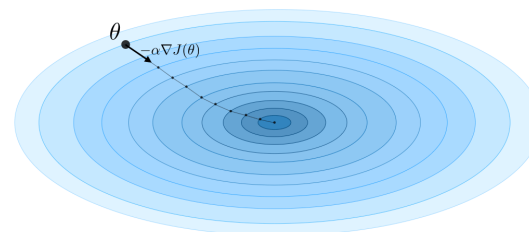
| 最小平方方法 | Logistic 損失函數 | Hinge 損失函數 | 交叉熵 |
|---|---|---|---|
| $\frac{1}{2}(y - z)^2$ | $\log(1 + \exp(-yz))$ | $\max(0, 1 - yz)$ | $-\left[y \log(z) + (1 - y) \log(1 - z)\right]$ |
|  |  |  |  |
| 線性迴歸 | 邏輯迴歸 | 支援向量機(SVM) | 神經網路 |

□ **代價函數** – 代價函數 J 通常用來評估一個模型的表現，它可以透過損失函數 L 來定義：

$$J(\theta) = \sum_{i=1}^m L(h_\theta(x^{(i)}), y^{(i)})$$

□ **梯度下降** – 使用 $\alpha \in \mathbb{R}$ 表示學習速率，我們透過學習速率和代價函數來使用梯度下降的方法找出網路參數更新的方法可以表示為：

$$\theta \leftarrow \theta - \alpha \nabla J(\theta)$$



注意：隨機梯度下降法 (SGD) 使用每一個訓練資料來更新參數。而批次梯度下降法則是透過一個批次的訓練資料來更新參數。

□ **概似估計** – 在給定參數 θ 的條件下，一個模型 $L(\theta)$ 的概似估計的目的是透過最大概似估計法來找到最佳的參數。實務上，我們會使用對數概似估計函數 (log-likelihood) $\ell(\theta) = \log(L(\theta))$ ，會比較容易最佳化。如下：

$$\theta^{\text{opt}} = \arg \max_{\theta} L(\theta)$$

□ **牛頓演算法** – 牛頓演算法是一個數值方法，目的在於找到一個 θ ，讓 $\ell'(\theta) = 0$ 。其更新的規則為：

$$\theta \leftarrow \theta - \frac{\ell'(\theta)}{\ell''(\theta)}$$

注意：多維度正規化的方法，或又被稱之為牛頓-拉弗森 (Newton-Raphson) 演算法，是透過以下的規則更新：

$$\theta \leftarrow \theta - \left(\nabla_{\theta}^2 \ell(\theta)\right)^{-1} \nabla_{\theta} \ell(\theta)$$

線性迴歸

我們假設 $y|x; \theta \sim \mathcal{N}(\mu, \sigma^2)$

□ **正規方程法** – 我們使用 X 代表矩陣，讓代價函數最小的 θ 值有一個封閉解，如下：

$$\theta = (X^T X)^{-1} X^T y$$

□ **最小均方演算法 (LMS)** – 我們使用 α 表示學習速率，針對 m 個訓練資料，透過最小均方演算法的更新規則，或是叫做 Widrow-Hoff 學習法如下：

$$\forall j, \quad \theta_j \leftarrow \theta_j + \alpha \sum_{i=1}^m [y^{(i)} - h_{\theta}(x^{(i)})] x_j^{(i)}$$

注意：這個更新的規則是梯度上升的一種特例

□ **LWR** – 局部加權迴歸，又稱為 LWR，是線性迴歸的變形，通過 $w^{(i)}(x)$ 對其成本函數中的每個訓練樣本進行加權，其中參數 $\tau \in \mathbb{R}$ 定義為：

$$w^{(i)}(x) = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$$

分類與邏輯迴歸

□ **Sigmoid 函數** – Sigmoid 函數 g ，也可以稱為邏輯函數定義如下：

$$\forall z \in \mathbb{R}, \quad g(z) = \frac{1}{1 + e^{-z}} \in [0, 1]$$

□ **邏輯迴歸** – 我們假設 $y|x; \theta \sim \text{Bernoulli}(\phi)$ ，請參考以下：

$$\phi = p(y = 1|x; \theta) = \frac{1}{1 + \exp(-\theta^T x)} = g(\theta^T x)$$

注意：對於這種情況的邏輯迴歸，並沒有一個封閉解

□ **Softmax 迴歸** – Softmax 迴歸又稱做多分類邏輯迴歸，目的是用在超過兩個以上的分類時的迴歸使用。按照慣例，我們設定 $\theta_K = 0$ ，讓每一個類別的 Bernoulli 參數 ϕ_i 等同於：

$$\phi_i = \frac{\exp(\theta_i^T x)}{\sum_{j=1}^K \exp(\theta_j^T x)}$$

廣義線性模型

□ **指數族分佈** – 一個分佈如果可以透過自然參數(或稱之為正準參數或連結函數) η 、充分統計量 $T(y)$ 和對數區分函數(log-partition function) $a(\eta)$ 來表示時，我們就稱這個分佈是屬於指數族分佈。該分佈可以表示如下：

$$p(y; \eta) = b(y) \exp(\eta^T T(y) - a(\eta))$$

注意：我們經常讓 $T(y) = y$ ，同時， $\exp(-a(\eta))$ 可以看成是一個正規化的參數，目的在於讓機率總和為一。

底下是最常見的指數分佈：

| 分佈 | η | $T(y)$ | $a(\eta)$ | $b(y)$ |
|----------------|--|--------|--|---|
| 白努利(Bernoulli) | $\log\left(\frac{\phi}{1-\phi}\right)$ | y | $\log(1 + \exp(\eta))$ | 1 |
| 高斯(Gaussian) | μ | y | $\frac{\eta^2}{2}$ | $\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right)$ |
| 卜瓦松(Poisson) | $\log(\lambda)$ | y | e^{η} | $\frac{1}{y!}$ |
| 幾何(Geometric) | $\log(1 - \phi)$ | y | $\log\left(\frac{e^{\eta}}{1 - e^{\eta}}\right)$ | 1 |

□ **廣義線性模型的假設** – 廣義線性模型 (GLM) 的目的在於，給定 $x \in \mathbb{R}^{n+1}$ ，要預測隨機變數 y ，同時它依賴底下三個假設：

$$(1) \quad y|x; \theta \sim \text{ExpFamily}(\eta) \quad (2) \quad h_{\theta}(x) = E[y|x; \theta] \quad (3) \quad \eta = \theta^T x$$

注意：最小平方法和邏輯迴歸是廣義線性模型的一種特例

支援向量機

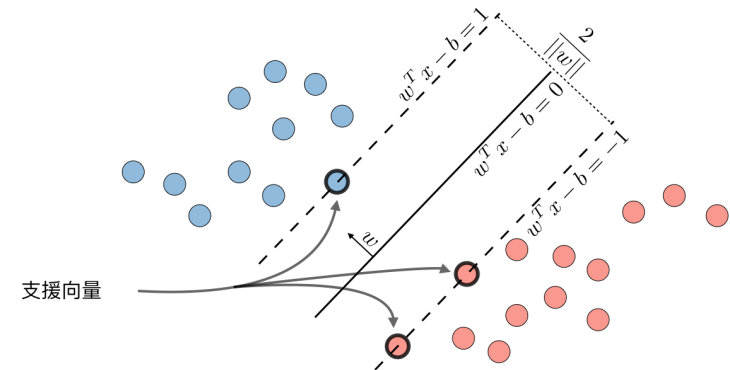
支援向量機的目的在於找到一條決策邊界和資料樣本之間最大化最小距離的線

□ **最佳的邊界分類器** – 最佳的邊界分類器可以表示為：

$$h(x) = \text{sign}(w^T x - b)$$

其中， $(w, b) \in \mathbb{R}^n \times \mathbb{R}$ 是底下最佳化問題的答案：

$$\min \frac{1}{2} \|w\|^2 \quad \text{使得} \quad y^{(i)}(w^T x^{(i)} - b) \geq 1$$



注意：該條直線定義為 $w^T x - b = 0$

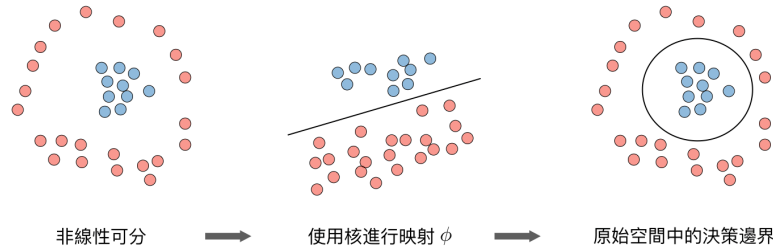
□ **Hinge 損失函數** – Hinge 損失函數用在支援向量機上，定義如下：

$$L(z, y) = [1 - yz]_+ = \max(0, 1 - yz)$$

□ **核(函數)** – 給定特徵轉換 ϕ ，我們定義核(函數) K 為：

$$K(x, z) = \phi(x)^T \phi(z)$$

實務上， $K(x, z) = \exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right)$ 定義的核(函數) K ，一般稱作高斯核(函數)。這種核(函數)經常被使用



注意：我們使用“核(函數)技巧”來計算代價函數時，不需要真正的知道映射函數 ϕ ，這個函數非常複雜。相反的，我們只需要知道 $K(x, z)$ 的值即可。

□ **Lagrangian** – 我們將Lagrangian $\mathcal{L}(w, b)$ 定義如下：

$$\mathcal{L}(w, b) = f(w) + \sum_{i=1}^l \beta_i h_i(w)$$

注意：係數 β_i 稱為Lagrange 乘數

生成學習

生成模型嘗試透過預估 $P(x|y)$ 來學習資料如何生成，而我們可以透過貝氏定理來預估 $P(y|x)$

高斯判別分析

□ **設定** – 高斯判別分析針對 y 、 $x|y=0$ 和 $x|y=1$ 進行以下假設：

$$y \sim \text{Bernoulli}(\phi)$$

$$x|y=0 \sim \mathcal{N}(\mu_0, \Sigma) \quad \text{和} \quad x|y=1 \sim \mathcal{N}(\mu_1, \Sigma)$$

□ **估計** – 底下的表格總結了我們在最大似估計時的估計值：

| $\hat{\phi}$ | $\hat{\mu}_j \quad (j=0,1)$ | $\hat{\Sigma}$ |
|--|---|---|
| $\frac{1}{m} \sum_{i=1}^m 1_{\{y^{(i)}=1\}}$ | $\frac{\sum_{i=1}^m 1_{\{y^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^m 1_{\{y^{(i)}=j\}}}$ | $\frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T$ |

單純貝氏

□ **假設** – 單純貝氏模型會假設每個資料點的特徵都是獨立的。

$$P(x|y) = P(x_1, x_2, \dots | y) = P(x_1|y)P(x_2|y)\dots = \prod_{i=1}^n P(x_i|y)$$

□ **解決方法** – 最大化對數概似估計來給出以下解答， $k \in \{0,1\}, l \in [1, L]$

$$P(y=k) = \frac{1}{m} \times \#\{j|y^{(j)}=k\} \quad \text{和} \quad P(x_i=l|y=k) = \frac{\#\{j|y^{(j)}=k \text{ 和 } x_i^{(j)}=l\}}{\#\{j|y^{(j)}=k\}}$$

注意：單純貝氏廣泛應用在文字分類和垃圾信件偵測上

基於樹狀結構的學習和整體學習

這些方法可以應用在迴歸或分類問題上

□ **CART** – 分類與迴歸樹(CART)，通常稱之為決策數，可以被表示為二元樹。它的優點是具有可解釋性。

□ **隨機森林** – 這是一個基於樹狀結構的方法，它使用大量經由隨機挑選的特徵所建構的決策樹。與單純的決策樹不同，它通常具有高度不可解釋性，但它的效能通常很好，所以是一個相當流行的演算法。

注意：隨機森林是一種整體學習方法

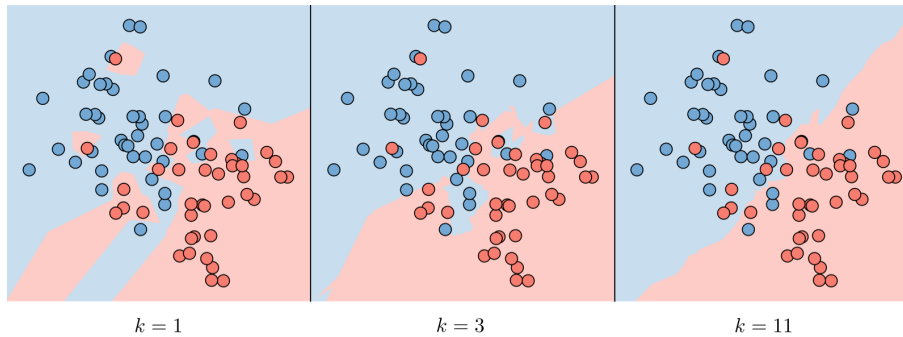
□ **增強學習(Boosting)** – 增強學習方法的概念是結合數個弱學習模型來變成強學習模型。主要的分類如下：

| 自適應增強 | 梯度增強 |
|---|------------------|
| - 在下一輪的提升步驟中，錯誤的部分會被賦予較高的權重 - "Adaboost" | - 弱學習器會負責訓練剩下的錯誤 |

其他非參數方法

□ **k-最近鄰** – k-最近鄰演算法，又稱之為k-NN，是一個非參數的方法，其中資料點的決定是透過訓練集中最近的k 個鄰居而決定。它可以用在分類和迴歸問題上。

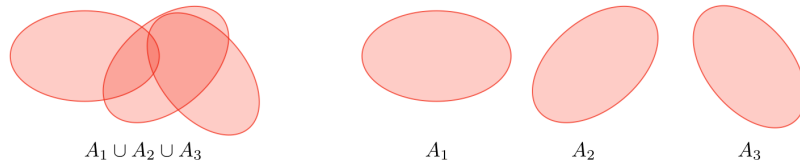
注意：參數k 的值越大，偏差越大。k 的值越小，變異越大。



學習理論

□ **聯集上界** – 令 A_1, \dots, A_k 為 k 個事件，我們有：

$$P(A_1 \cup \dots \cup A_k) \leq P(A_1) + \dots + P(A_k)$$



□ **霍夫丁不等式** – 令 Z_1, \dots, Z_m 為 m 個從參數 ϕ 的白努利分佈中抽出的獨立同分佈(iid) 的變數。令 $\hat{\phi}$ 為其樣本平均、固定 $\gamma > 0$ ，我們可以得到：

$$P(|\phi - \hat{\phi}| > \gamma) \leq 2 \exp(-2\gamma^2 m)$$

注意：這個不等式也被稱之為 *Chernoff* 界線

□ **訓練誤差** – 對於一個分類器 h ，我們定義訓練誤差為 $\hat{\epsilon}(h)$ ，也可以稱為經驗風險或經驗誤差。定義如下：

$$\hat{\epsilon}(h) = \frac{1}{m} \sum_{i=1}^m 1_{\{h(x^{(i)}) \neq y^{(i)}\}}$$

□ **可能近似正確(PAC)** – PAC 是一個框架，有許多學習理論都證明其有效性。它包含以下假設：

- 訓練和測試資料集具有相同的分佈
- 訓練資料集之間彼此獨立

□ **打散(Shattering)** – 給定一個集合 $S = \{x^{(1)}, \dots, x^{(d)}\}$ 以及一組分類器的集合 \mathcal{H} ，如果對於任何一組標籤 $\{y^{(1)}, \dots, y^{(d)}\}$ ， \mathcal{H} 都能打散 S ，定義如下：

$$\exists h \in \mathcal{H}, \quad \forall i \in [1, d], \quad h(x^{(i)}) = y^{(i)}$$

□ **上限定理** – 令 \mathcal{H} 是一個有限假設類別，使 $|\mathcal{H}| = k$ 且令 δ 和樣本大小 m 固定，結著，在機率至少為 $1 - \delta$ 的情況下，我們得到：

$$\epsilon(\hat{h}) \leq \left(\min_{h \in \mathcal{H}} \epsilon(h) \right) + 2 \sqrt{\frac{1}{2m} \log \left(\frac{2k}{\delta} \right)}$$

□ **VC 維度** – 一個有限假設類別的 Vapnik-Chervonenkis (VC) 維度 $VC(\mathcal{H})$ 指的是 \mathcal{H} 最多能夠打散的數量

注意： $\mathcal{H} = \{2 \text{ 維的線性分類器}\}$ 的 VC 維度為 3



□ **理論(Vapnik)** – 令 \mathcal{H} 已給定， $VC(\mathcal{H}) = d$ 且 m 是訓練資料級的數量，在機率至少為 $1 - \delta$ 的情況下，我們得到：

$$\epsilon(\hat{h}) \leq \left(\min_{h \in \mathcal{H}} \epsilon(h) \right) + O \left(\sqrt{\frac{d}{m} \log \left(\frac{m}{d} \right)} + \frac{1}{m} \log \left(\frac{1}{\delta} \right) \right)$$