A Tutorial on Relevance Vector Machine

楊善詠

June 9, 2006

1　前言

這篇文章的內容主要在介紹 Relevence Vector Machine (RVM) 的基本概念與做法。由於 RVM 使用機率的方法來克服 Support Vector Machine (SVM) 的缺點，因此我也會一併介紹一些重要的機率概念。

我會假設這篇文章的讀者對機器學習有最基本的知識，並且稍微了解 SVM 的原理。

為了避免混淆，在所有的數學式中，一般的小寫斜體表示純量，如 w_i, t_i 等；小寫粗體表示向量，如 $\mathbf{x}, \mathbf{w}, \boldsymbol{\alpha}$ 等；而大寫正體或大寫希臘字母表示矩陣，如 A, Φ, Σ 等。此外，大寫的 $P(\cdot)$ 表示離散的機率分佈函數，而小寫的 $p(\cdot)$ 則是連續的機率分佈函數。

2　簡介

Supervised learning 意指我們要解決如下的問題：給定一群向量 $\{\mathbf{x}_i\}_{i=1}^N$ 與對應的目標 $\{t_i\}_{i=1}^N$作為輸入，我們想要找出 $\mathbf{x}_i$ 與 t_i 之間的對應關係，讓我們能夠在遇到一個新的向量 $\mathbf{x}_*$ 時，能夠預測出它所對應的目標 t_* 。這邊的 t_i 可能是類別標籤（分類：classification），或是任意實數（回歸：regression）。

如果使用 SVM 解這類問題，會導出 $\mathbf{x}$ 與 t 的對映關係符合以下的函數：

$$t = y(\mathbf{x}; \mathbf{w}) = \sum_{i=1}^{N} w_i K(\mathbf{x}, \mathbf{x}_i) + w_0 \tag{1}$$

其中 $K(\mathbf{x}, \mathbf{x}_i)$ 是我們選用的 kernel function，而 w_i 則代表不同的權重。只有在 $\mathbf{x}_i$ 是屬於 support vector 之一時，w_i 才會是零以外的值。

實作顯示 SVM 的表現良好，因此 SVM 被運用在許多地方。然而 SVM 並非沒有缺點，以下是 SVM 較為人垢病之處：

- 雖然 support vector 的數量會明顯少於 training instance 的個數，但依然會隨著 training instance 的數量線性成長。一方面可能造成過度調適 (overfitting) 的問題，另一方面則浪費計算時間。實作上經常需要多一步降低 support vector 數量的處理動作。
- 無法得到機率式的預測。一般人會比較偏好機率式的預測，因為機率式的預測能夠給人確定程度的資訊。就和氣象預報不會單純預測天氣為晴天或雨天，而會預測降雨機率的道理一般。
- SVM 的使用者必須給定一個誤差參數 C，這個參數對結果有很大的影響。不幸的是，大部分的情況下，使用者都必須猜過各種可能值，才能找最好的結果。
- Kernel function $K(\mathbf{x}, \mathbf{x}_i)$ 必須符合 Mercer's condition。

理論上 SVM 對雜訊是很敏感的，因為它求解時限制所有的 training instance 一定能被完美切開。雖然使用誤差參數 C 可以放寬這個限制，但也造成使用者不知道如何決定 C 的兩難問題。面對這個問題，最常見的解決方法就是引入機率的模型來解釋雜訊，不但可以脫離這個兩難困境，還得到了機率式預測的好處，這正是 RVM 的核心概念。

3 先講點機率吧！

3.1 貝式定理 (Bayes' Theorem)

在推導 RVM 之前，我們先來複習一下基本的機率常識，後面就會大量使用這些式子。

假設 A 是個連續的隨機變數，那麼它的機率分佈函數 $p(A)$ 會符合以下的性質：

$$\int_{\Theta} p(A)dA = 1$$

注意這邊的積分是定積分，其中 Θ 表示 A 的值域。在這篇文章中的積分符號皆表示定積分，因此我會省略不寫 Θ。

又設 A, B 皆為連續的隨機變數，那麼 $p(A, B)$ 也會符合以下的性質：

$$\int p(A, B)dA = p(B)$$

這兩個式子看起來很簡單也很直覺，但它們是導出下列式子的基礎，繼續往下吧！現在考慮機率的老朋友－－貝式定理：

$$p(A|B) = \frac{p(A,B)}{p(B)} = \frac{p(B|A)p(A)}{p(B)}$$

注意 $p(B)$ 可以寫成 $\int p(A,B)dA$，因此可以得到下列的式子：

$$p(A|B) = \frac{p(A,B)}{\int p(A,B)dA} = \frac{p(B|A)p(A)}{\int p(B|A)p(A)dA} \tag{2}$$

3.2 馬可夫性質 (Markov Property)

假設我們有兩顆骰子 A 與 B。A 有五面寫 B，一面寫 A；B 則有五面寫 A，一面寫 B。而遊戲規則是一次只能丟一顆骰子，得到的字母則代表下一次要丟哪一顆骰子。

設每次丟骰子的動作是彼此獨立的，若我們第一次丟 A，則得到 A 的機率為 $1/6$，得到 B 的機率則為 $5/6$。因此，第二次丟骰子時，得到 A 與 B 的機率分別為：

$$\begin{aligned}
P(T_2 = A) &= P(T_1 = A) \cdot \frac{1}{6} + P(T_1 = B) \cdot \frac{5}{6} \\
&= \frac{26}{36} \\
P(T_2 = B) &= P(T_1 = A) \cdot \frac{5}{6} + P(T_1 = B) \cdot \frac{1}{6} \\
&= \frac{10}{36}
\end{aligned}$$

其中 T_1 與 T_2 分別代表第一次與第二次丟骰子的結果。

若我們丟了許多次，會發現第 n 次得到的結果只和第 $n-1$ 次得到的結果有關。寫成條件機率就會長這樣：

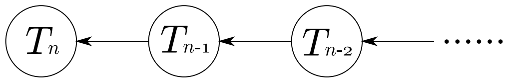
$$\begin{aligned}
P(T_n = A|T_{n-1} = A) &= \frac{1}{6} \\
P(T_n = A|T_{n-1} = B) &= \frac{5}{6} \\
P(T_n = B|T_{n-1} = A) &= \frac{5}{6} \\
P(T_n = B|T_{n-1} = B) &= \frac{1}{6}
\end{aligned}$$

一旦 T_{n-1} 已知，就會決定 T_n 的機率分佈。值得注意的是，T_n 的機率分佈和 T_{n-1} 有關，而 T_{n-1} 的機率分佈又和 T_{n-2} 有關，因此 T_n 和 T_{n-2} 並不算互相獨立的事件。但它們之

間又沒有直接的相依性，而有以下的性質：

$$P(T_n|T_{n-1}) = P(T_n|T_{n-1}, T_{n-2}) = P(T_n|T_{n-1}, T_{n-2}, T_{n-3}, \dots) \tag{3}$$

這個特性，我們稱為馬可夫性質（Markov Property）。



相依性關係示意圖

在馬可夫性質下，我們可以導出下列的式子：

$$\begin{aligned}
P(T_n|T_{n-2}) &= \frac{P(T_n, T_{n-2})}{P(T_{n-2})} \\
&= \sum_{T_{n-1}} \frac{P(T_n, T_{n-1}, T_{n-2})}{P(T_{n-2})} \\
&= \sum_{T_{n-1}} \frac{P(T_n, T_{n-1}, T_{n-2})}{P(T_{n-1}, T_{n-2})} \cdot \frac{P(T_{n-1}, T_{n-2})}{P(T_{n-2})} \\
&= \sum_{T_{n-1}} P(T_n|T_{n-1}, T_{n-2}) P(T_{n-1}|T_{n-2}) \\
&= \sum_{T_{n-1}} P(T_n|T_{n-1}) P(T_{n-1}|T_{n-2}) && \text{(馬可夫性質)}
\end{aligned}$$

這是在離散隨機變數的情況下，而在連續的情況則為：

$$p(T_n|T_{n-1}) = \int p(T_n|T_{n-1}) p(T_{n-1}|T_{n-2}) dT_{n-1} \tag{4}$$

3.3 簡單的機率式預測

機率式預測的目的很單純：已知一件發生過的事件 t，我們想知道：在 t 已發生的條件下，未發生事件 t_* 所發生的機率。也就是說，我們要求的是 $P(t_*|t)$。

舉個具體一點的例子：假設我們丟了十次硬幣，其中七次是正面，三次是反面，那麼下次得到正面的機率是多少？在這個例子中，已發生事件 t 就是我們丟了十次硬幣，七次是正面而三次是反面；未發生事件 t_* 則是下次丟硬幣而得到正面。

在解決這類的問題時，我們通常會假設 t 與 t_* 的發生機率都遵循某個既定的原則 θ。我們知道這個原則的「大方向」（運作的原理），但不知道這個原則的「小細節」（參數

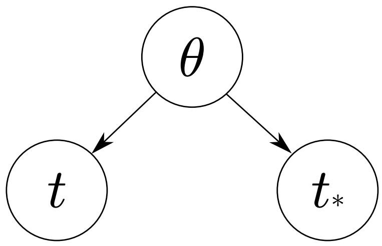
的值）。在丟硬幣的例子中，我們會假設每次丟硬幣都是獨立事件，正面與反面的機率分別是 θ(正) 與 θ(反)。雖然我們知道這個原則，卻不知道 θ 這個機率分佈是什麼。

當然，一旦 θ 決定好，丟硬幣時的機率分佈就決定了，也就是說，雖然 t 與 t_* 之前是相關的，但它們沒有直接的相依性，而遵守馬可夫性質：

$$P(t_*|\theta) = P(t_*|\theta, t)$$

所以我們得到

$$P(t_*|t) = \int P(t_*|\theta)P(\theta|t)d\theta$$



θ 與 t 之間的關係

不幸的是，一般情況下這個定積分是很難解的，因此我們退而求其次地找近似解。最簡單的方法就是用 delta function 去近似 $P(\theta|t)$：

$$\begin{aligned}\int P(t_*|\theta)P(\theta|t)d\theta &\approx \int P(t_*|\theta)\delta(\theta - \hat{\theta})d\theta \\ &= P(t_*|\hat{\theta})\end{aligned}$$

這麼一來就簡單多了。但 $\hat{\theta}$ 的值應該是多少呢？我們要找的 $\hat{\theta}$ 應該要符合「$\delta(\theta - \hat{\theta})$ 與 $P(\theta|t)$ 愈像愈好」的條件。最合理的作法，就是取在 $P(\theta|t)$ 的最大值處：

$$\hat{\theta} = \arg\max_{\theta} P(\theta|t)$$
$$P(t_*|t) \approx P(t_*|\hat{\theta})$$

這個方法，稱之為 Maximum a Posteriori (MAP)。

還有另一種常見情況是 $P(\theta|t)$ 不容易求，這時候我們會用貝式定理把它拆開：

$$P(\theta|t) = \frac{P(t|\theta)P(\theta)}{P(t)}$$

我們的目標是 θ，因此不看分母 $P(t)$。又假設 θ 出現各種可能值的機率都是相同的，亦即 $P(\theta)$ 為常數，因此 $\hat{\theta}$ 就是：

$$\hat{\theta} = \arg\max_{\theta} P(t|\theta)$$

這個方法就是常見的 Maximum Likelihood(ML)。

回到丟硬幣的問題。若 θ 代表以下的機率分佈：

$$\theta(\text{正}) = k,\ \theta(\text{反}) = 1 - k$$

可得

$$P(t|\theta) = k^7(1-k)^3$$

若套用 Maximum Likelihood 的方法，我們找 k 使得 $P(t|\theta)$ 最大，只要把該式微分就可得：

$$\arg\max_{k} k^7(1-k)^3 = 0.7$$
$$P(t_*|\hat{\theta}) = \begin{cases} 0.7, & \text{若 } t_* = \text{正} \\ 0.3, & \text{若 } t_* = \text{反} \end{cases}$$

這好像很符合一般人的直覺。但必須要注意的是，因為我們沒有任何對 θ 的先決條件(prior)，所以會導出得正面的機率最可能為 0.7。若我們有對 θ 做出任何先決條件（比如說，得正面的機率應該很接近 0.5），不論是過程和結果都會大不相同。這也是 MAP 和 ML 最大的不同，ML 只是 MAP 的一個特例罷了。

4 Revelance Vector Regression

如果前面的部分都能讓你接受， RVM 的概念就很簡單了。馬上就來介紹如何使用 RVM 來解回歸問題。

設 $\{\mathbf{x}_i\}_{i=1}^{N}$ 是 training data 中的特徵值（feature 或 attribute），$\mathbf{t} = [t_1, t_2, \ldots, t_N]^T$ 則是目標值，RVM 與 SVM 同樣假設它們之間的關係符合函數 (1)，不同的是，t_i 還加上誤差的影響，因此我們必須用機率來表達它：

$$y(\mathbf{x};\mathbf{w}) = \sum_{i=1}^{N} w_i K(\mathbf{x}, \mathbf{x}_i) + w_0$$
$$p(t_i) = \mathcal{N}(t_i | y(\mathbf{x}_i;\mathbf{w}), \sigma^2)$$

其中 $\mathcal{N}(\cdot)$ 代表 normal distribution density function，定義如下：

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{x^2}{2\sigma^2})$$

我們可以合理地假設 $\{t_i\}_{i=1}^N$ 是彼次獨立的隨機變數，因此在已知 $\{w_i\}_{i=0}^N$ 與 σ^2 的條件下，$\mathbf{t}$ 的機率分佈如下：

$$\begin{aligned} p(\mathbf{t}|\mathbf{w},\sigma^2) &= \prod_{i=1}^N \mathcal{N}(t_i|y(\mathbf{x}_i;\mathbf{w}),\sigma^2) \\ &= (2\pi\sigma^2)^{-\frac{N}{2}}\exp(-\frac{||\mathbf{t}-\Phi\mathbf{w}||^2}{2\sigma^2}) \end{aligned}$$

其中 $\mathbf{w}$ 是由 w_i 組成的向量，Φ 則是由各特徵向量代入 kernel function 所得的 design matrix,

$$\begin{aligned} \mathbf{w} &= [w_0, w_1, w_2, \ldots, w_N]^T \\ \Phi &= \begin{bmatrix} 1 & K(\mathbf{x}_1,\mathbf{x}_1) & K(\mathbf{x}_1,\mathbf{x}_2) & \ldots & K(\mathbf{x}_1,\mathbf{x}_N) \\ 1 & K(\mathbf{x}_2,\mathbf{x}_1) & K(\mathbf{x}_2,\mathbf{x}_2) & \ldots & K(\mathbf{x}_2,\mathbf{x}_N) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & K(\mathbf{x}_N,\mathbf{x}_1) & K(\mathbf{x}_N,\mathbf{x}_2) & \ldots & K(\mathbf{x}_N,\mathbf{x}_N) \end{bmatrix} \end{aligned}$$

由前面的機率式預測，我們所求的條件機率如下：

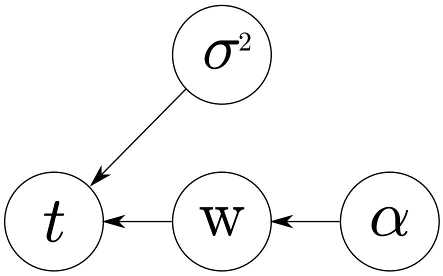
$$\begin{aligned} p(t_*|\mathbf{t}) &= \int p(t_*|\mathbf{w},\sigma^2)p(\mathbf{w},\sigma^2|\mathbf{t})d\mathbf{w}d\sigma^2 \\ &= \int p(t_*|\mathbf{w},\sigma^2)\frac{p(\mathbf{t}|\mathbf{w},\sigma^2)p(\mathbf{w},\sigma^2)}{p(\mathbf{t})}d\mathbf{w}d\sigma^2 \end{aligned}$$

如果直接使用 Maximum Likelihood 的方法解 $\mathbf{w}$ 與 σ^2，結果通常使 $\mathbf{w}$ 中的元素大部分都不是 0。讓我們回想 SVM 的第一個缺點：使用了過多的 support vector 而導致 overfitting。在 RVM 中我們想要避免這個現像，因此我們為 $\mathbf{w}$ 加上先決條件：它們的機率分佈是落在 0 周圍的 normal distribution：

$$\begin{aligned} p(w_i|\alpha_i) &= \mathcal{N}(w_i|0,\alpha_i^{-1}) \\ \boldsymbol{\alpha} &= [\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_N]^T \\ p(\mathbf{w}|\boldsymbol{\alpha}) &= \prod_{i=0}^N \frac{\alpha_i}{\sqrt{2\pi}}\exp(-\frac{\alpha_i w_i^2}{2}) \end{aligned}$$

因此前面的機率預測改為：

$$p(t_*|\mathbf{t}) = \int p(t_*|\mathbf{w},\boldsymbol{\alpha},\sigma^2)p(\mathbf{w},\boldsymbol{\alpha},\sigma^2|\mathbf{t})d\mathbf{w}d\boldsymbol{\alpha}d\sigma^2 \tag{5}$$



RVM 的參數相依性示意圖

先整理一下式子。前面的 $p(t_*|\mathbf{w},\boldsymbol{\alpha},\sigma^2)$ 中，因為 t_* 只和 $\mathbf{w}$ 與 σ^2 直接相關，由馬可夫性質得到：

$$\begin{aligned}p(t_*|\mathbf{w},\boldsymbol{\alpha},\sigma^2) &= p(t_*|\mathbf{w},\sigma^2)\\ &= \mathcal{N}(t_*|y(\mathbf{x}_*;\mathbf{w}),\sigma^2)\end{aligned}$$

後面則用貝式定理拆開：

$$\begin{aligned}p(\mathbf{w},\boldsymbol{\alpha},\sigma^2|\mathbf{t}) &= \frac{p(\mathbf{w},\mathbf{t},\boldsymbol{\alpha},\sigma^2)}{p(\mathbf{t})}\\ &= \frac{p(\mathbf{w},\mathbf{t},\boldsymbol{\alpha},\sigma^2)}{p(\mathbf{t},\boldsymbol{\alpha},\sigma^2)}\frac{p(\mathbf{t},\boldsymbol{\alpha},\sigma^2)}{p(\mathbf{t})}\\ &= p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha},\sigma^2)p(\boldsymbol{\alpha},\sigma^2|\mathbf{t})\end{aligned}$$

$$\begin{aligned}p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha},\sigma^2) &= \frac{p(\mathbf{w},\mathbf{t},\boldsymbol{\alpha},\sigma^2)}{p(\mathbf{t},\boldsymbol{\alpha},\sigma^2)}\\ &= \frac{p(\mathbf{t}|\mathbf{w},\boldsymbol{\alpha},\sigma^2)p(\mathbf{w},\boldsymbol{\alpha},\sigma^2)}{p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2)p(\boldsymbol{\alpha},\sigma^2)}\\ &= \frac{p(\mathbf{t}|\mathbf{w},\boldsymbol{\alpha},\sigma^2)p(\mathbf{w}|\boldsymbol{\alpha},\sigma^2)p(\boldsymbol{\alpha},\sigma^2)}{p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2)p(\boldsymbol{\alpha},\sigma^2)}\\ &= \frac{p(\mathbf{t}|\mathbf{w},\sigma^2)p(\mathbf{w}|\boldsymbol{\alpha})}{p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2)}\end{aligned}$$

我們知道貝式定理也可以寫成定積分的形式：

$$\frac{p(\mathbf{t}|\mathbf{w},\sigma^2)p(\mathbf{w}|\boldsymbol{\alpha})}{p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2)} = \frac{p(\mathbf{t}|\mathbf{w},\sigma^2)p(\mathbf{w}|\boldsymbol{\alpha})}{\int p(\mathbf{t}|\mathbf{w},\sigma^2)p(\mathbf{w}|\boldsymbol{\alpha})d\mathbf{w}}$$

其中 $p(\mathbf{t}|\mathbf{w},\sigma^2)$ 與 $p(\mathbf{w}|\boldsymbol{\alpha})$ 都是 Gaussian function 的乘積，因此對它定積分並不是問題。積分後化簡得到：

$$p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha},\sigma^2) = (2\pi)^{-\frac{N+1}{2}}|\Sigma|^{-\frac{1}{2}}\exp\{-\frac{(\mathbf{w}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{w}-\boldsymbol{\mu})}{2}\} \tag{6}$$

$$p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2) = (2\pi)^{-\frac{N}{2}}|\Omega|^{-\frac{1}{2}}\exp(-\frac{\mathbf{t}^T\Omega^{-1}\mathbf{t}}{2}) \tag{7}$$

其中：

$$\begin{aligned}
\Sigma &= (\sigma^{-2}\Phi^T\Phi + \mathrm{A})^{-1} \\
\boldsymbol{\mu} &= \sigma^{-2}\Sigma\Phi^T\mathbf{t} \\
\mathrm{A} &= \begin{bmatrix} \alpha_0 & 0 & 0 & \dots & 0 \\ 0 & \alpha_1 & 0 & \dots & 0 \\ 0 & 0 & \alpha_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \alpha_N \end{bmatrix} \\
\Omega &= \sigma^2 I + \Phi\mathrm{A}^{-1}\Phi^T
\end{aligned}$$

經過代換後的 (5) 如下：

$$p(t_*|\mathbf{t}) = \int p(t_*|\mathbf{w},\boldsymbol{\alpha},\sigma^2)p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha},\sigma^2)p(\boldsymbol{\alpha},\sigma^2|\mathbf{t})d\mathbf{w}d\boldsymbol{\alpha}d\sigma^2$$

現在我們可以找它的近似解了：

$$\begin{aligned}
(\boldsymbol{\alpha}_{\mathrm{MP}},\sigma^2_{\mathrm{MP}}) &= \arg\max_{\boldsymbol{\alpha},\sigma^2} p(\boldsymbol{\alpha},\sigma^2|\mathbf{t}) \\
p(t_*|\mathbf{t}) &= \int p(t_*|\mathbf{w},\boldsymbol{\alpha},\sigma^2)p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha},\sigma^2)p(\boldsymbol{\alpha},\sigma^2|\mathbf{t})d\mathbf{w}d\boldsymbol{\alpha}d\sigma^2 \\
&\approx \int p(t_*|\mathbf{w},\boldsymbol{\alpha},\sigma^2)p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha},\sigma^2)\delta(\boldsymbol{\alpha}-\boldsymbol{\alpha}_{\mathrm{MP}})\delta(\sigma^2-\sigma^2_{\mathrm{MP}})d\mathbf{w}d\boldsymbol{\alpha}d\sigma^2 \\
&= \int p(t_*|\mathbf{w},\boldsymbol{\alpha}_{\mathrm{MP}},\sigma^2_{\mathrm{MP}})p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha}_{\mathrm{MP}},\sigma^2_{\mathrm{MP}})dw
\end{aligned}$$

積分式中的兩項皆為 Gaussian function 的乘積，因此定積分後的結果為：

$$\begin{aligned}
p(t_*|\mathbf{t}) &= \mathcal{N}(t_*|y_*,\sigma^2_*) \\
y_* &= \boldsymbol{\mu}^T\boldsymbol{\phi}(\mathbf{x}_*) \\
\sigma^2_* &= \sigma^2_{\mathrm{MP}} + \boldsymbol{\phi}(\mathbf{x}_*)^T\Sigma\boldsymbol{\phi}(\mathbf{x}_*) \\
\boldsymbol{\phi}(\mathbf{x}_*) &= [1,\ K(\mathbf{x}_*,\mathbf{x}_1),\ K(\mathbf{x}_*,\mathbf{x}_2),\ \dots,\ K(\mathbf{x}_*,\mathbf{x}_N)]^T
\end{aligned}$$

現在剩下的問題就是如何求 $\boldsymbol{\alpha}_{\mathrm{MP}}$ 及 σ^2_{MP} 了。簡單一點的方法是使用 Maximum-Likelihood：

$$p(\boldsymbol{\alpha}, \sigma^2|\mathbf{t}) \propto p(\mathbf{t}|\boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha})p(\sigma^2)$$
$$(\boldsymbol{\alpha}_{\mathrm{MP}}, \sigma^2_{\mathrm{MP}}) = \arg\max_{\boldsymbol{\alpha},\sigma^2} p(\mathbf{t}|\boldsymbol{\alpha}, \sigma^2)$$

不幸的是，在第 (7) 式中並沒有公式解可以求最大值，因此我們須要用數值方法求近似解。把 (7) 式對 $\boldsymbol{\alpha}$ 與 σ^2 偏微分後求等與零的解，可得：

$$\alpha_i^{\mathrm{new}} = \frac{\gamma_i}{\mu_i^2} \tag{8}$$
$$(\sigma^2)^{\mathrm{new}} = \frac{||\mathbf{t} - \Phi\boldsymbol{\mu}||^2}{N - \sum_{i=0}^{N}\gamma_i} \tag{9}$$
$$\gamma_i = 1 - \alpha_i\Sigma_{i,i} \tag{10}$$

其中 $\Sigma_{i,i}$ 是 Σ 中第 i 項在對角線上的元素。我們先給出 $\boldsymbol{\alpha}$ 與 σ^2 猜測值，然後由上式不斷更新，就能逼近 $\boldsymbol{\alpha}_{\mathrm{MP}}$ 及 σ^2_{MP}。

在足夠多的更新之後，大部分的 α_i 會趨近無限大，意即對應的 w_i 為 0。其它的 α_i 會穩定趨近有限值，與之對應的 $\mathbf{x}_i$ 就稱之為 **relevance vector**。

5 Relevance Vector Classification

現在我們討論二元分類的情況：目標值 $\{t_i\}_{i=0}^N$ 只可能為 0 或 1。這邊我們使用把回歸演算法應用到分類問題時常用的 sigmoid function：

$$P(t_i = 1|\mathbf{w}) = \sigma[y(\mathbf{x}_i; \mathbf{w})] = \frac{1}{1 + e^{-y(\mathbf{x}_i;\mathbf{w})}}$$

在每次觀測皆為獨立事件的前提下，得到觀測結果為 $\mathbf{t}$ 的機率為：

$$P(\mathbf{t}|\mathbf{w}) = \prod_{i=1}^{N} \sigma[y(\mathbf{x}_i; \mathbf{w})]^{t_i}\{1 - \sigma[y(\mathbf{x}_i; \mathbf{w})]\}^{1-t_i} \tag{11}$$

除了少一項雜訊變異量外，解法其本上和前面的回歸問題相同。然而這邊導出的 $P(\mathbf{t}|\mathbf{w})$ 並非 normal distribution，也無法直接求解定積分，因此我們套用 Laplace's method：

1. 假設 $\boldsymbol{\alpha}$ 已知，找出 $\mathbf{w}$ 發生最大值之處。亦即我們要找：

$$\begin{aligned}
\mathbf{w}_{\mathrm{MP}} &= \arg\max_{\mathbf{w}} p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha}) \\
&= \arg\max_{\mathbf{w}} \frac{P(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})}{p(\boldsymbol{\alpha},\mathbf{t})} \\
&= \arg\max_{\mathbf{w}} P(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha}) \\
&= \arg\max_{\mathbf{w}} \log\{P(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha})\}
\end{aligned}$$

取 log 後的 $P(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha})$ 如下：

$$\log\{P(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha})\} = \sum_{i=1}^{N}[t_i \log y_i + (1-t_i)\log(1-y_i)] - \frac{1}{2}\mathbf{w}^T \mathrm{A}\mathbf{w} \tag{12}$$

其中 $y_i = \sigma\{y(\mathbf{x}_i;\mathbf{w})\}$。

一般而言，使用 Newton's method 可以很快地找到 $\mathbf{w}_{\mathrm{MP}}$。剛好在使用 Newton's method 時所計算的 Hessian 在第二步也會被用到，方法如下：

$$\begin{aligned}
\mathbf{g} =& \nabla_{\mathbf{w}} \log\{P(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha})\} \\
=& \Phi^T(\mathbf{t}-\mathbf{y}) - \mathrm{A}\mathbf{w} \\
\mathrm{H} =& \nabla_{\mathbf{w}}\nabla_{\mathbf{w}} \log\{P(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha})\} \\
=& (-\Phi^T\mathrm{B}\Phi - \mathrm{A})^{-1} \\
\Delta\mathbf{w} =& -\mathrm{H}^{-1}\mathbf{g} \\
\mathbf{w}_{\mathrm{MP}} :=& \mathbf{w}_{\mathrm{MP}} + \Delta\mathbf{w}
\end{aligned}$$

其中

$$\begin{aligned}
\mathbf{y} &= [y_1, y_2, \ldots, y_N]^T \\
\mathrm{B} &= \begin{bmatrix} y_1(1-y_1) & 0 & \ldots & 0 \\ 0 & y_2(1-y_2) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & y_N(1-y_N) \end{bmatrix}
\end{aligned}$$

2. Laplace's method 使用 Gaussian function 來近似 $p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha})$，而這個 Gaussian function 的中心位於 $\mathbf{w}_{\mathrm{MP}}$，變異矩陣 (covariance matrix) 則由 Hessian 而來：

$$\begin{aligned}
\Sigma &= (-\mathrm{H}\mid_{\mathbf{w}_{\mathrm{MP}}})^{-1} \\
&= (\Phi^T\mathrm{B}\Phi + \mathrm{A})^{-1}
\end{aligned}$$

3. 使用 $\mathbf{w}_{\mathrm{MP}}$ 代替 $\boldsymbol{\mu}$，並配合上一步的 Σ 更新 $\boldsymbol{\alpha}$，方法則和 (8) 相同。

6 實作細節

在每一次的更新過程中，$\boldsymbol{\alpha}$ 中大部分的值會趨近無限大，因此我們在每次更新過 $\boldsymbol{\alpha}$ 後，會把過大的 α_i 視為無限大而不再更新，對應的 w_i 與 μ_i 則為 0，這麼一來就可以省去不少矩陣運算時間。更重要的是，若我們不除去過大的 $\boldsymbol{\alpha}$，Σ^{-1} 會是一個誤差敏感 (ill-conditioned) 矩陣，由其反矩陣更新 $\boldsymbol{\alpha}$ 將變得毫無意義。

此外，作者在分析過 $\boldsymbol{\alpha}$ 的發散過程後發現，當 $\boldsymbol{\alpha}$ 已接近最後的答案時，發散的速度會變得極慢。因此他提出了另一個更新方法：

$$\alpha_i = \gamma_i \frac{\gamma_i - \Sigma_{i,i}}{\mu_i^2} \tag{13}$$

使用這個式子會讓 $\boldsymbol{\alpha}$ 快速發散，但若一開始就使用會產生過大的誤差，因此應該等 $\boldsymbol{\alpha}$ 呈現穩定成長時才使用這個更新方法。

最後，我們並非一定要求出 Σ 不可。因為

$$\Sigma^{-1} = \sigma^2 \Phi^T \Phi + A$$

是個對稱矩陣，我們可以用 Cholesky decomposition 進行分解：

$$\Sigma^{-1} = \mathrm{U}^T \mathrm{U}$$

其中 U 是個三角矩陣。接著再計算 U^{-1}，並由 U^{-1} 來取代會用到 Σ 的地方。一般來說這個方法會比較快。

7 結論

RVM 把機率模型引入 SVM 中，因此改善了 SVM 對雜訊處理的缺陷，並增加了機率式預測的優點。另一方面，與 SVM 比較測試的結果也指出 relevance vector 的數量遠少於 support vector。

然而 RVM 並非沒有缺點，最大的缺點是它的矩陣運算使用 $O(N^2)$ 的記憶體，但實務上 training data 的數量卻動輒數萬至數十萬，使得 RVM 難以用來解決一般性的問題。