Support Vector Regression

R94922044 黃子桓

1 導論

Support Vector Machine 除了分類 (classification) 問題外, 也可用來處理回歸 (regression) 的問題。 所謂回歸指的是每個實體 (instance) 所對應的標籤 (label) 是連續的實數, 而非離散的相異類別 (在 SVM 裡常以整數來表示)。 處理回歸問題的 SVM, 稱為 Support Vector Regression。

2 基本想法

如同 SVM, SVR 的目標為尋找空間中的最適平面 (hyperplane)。 和 SVM 不同的是, SVM 找的是能將資料一分為二的平面, 而 SVR 所找的則是能準確預測資料分佈的平面。 假設訓練資料 (training data) 表示為 $(x_1, y_1), \ldots, (x_l, y_l) \in \mathbb{R}^d \times \mathbb{R}$, 其中 x 表示輸入的「特徵(attributes)」, y 表示該特徵所對應的回歸值 (相當於 SVM 中的目標類別, target class)。 令 $f(x) = w \cdot x + b,\ w \in \mathbb{R}^d,\ b \in \mathbb{R}$, 如果對每個 instance x_i 而言, $f(x_i)$ 和 y_i 的差值都很小, 則我們知道這樣的 $f(x)$ 能從 x 準確地預測 y, 這個 w 即是 SVR 所要找的平面。 用數學語言來表達, 可將 SVR 改寫成下面問題 (請對照 SVM 一起看):

$$\begin{aligned} &\text{minimize } \frac{1}{2}\|w\|^2 \\ &\text{subject to } \|y_i - (w \cdot x_i - b)\| \leq \varepsilon \end{aligned} \tag{1}$$

其中 $\varepsilon \geq 0$, 用來表示 SVR 預測值與實際值最大的差距, 而此演算法也因此而得名, 稱為 ε-SVR。 式 (1) 和 SVM 相對照, 可發現十分相像, 不同者為 SVM 考慮的

是預測類別 (predicted class) 和實際類別 (actual class) 需同號 (表示預測正確), 而 SVR 考慮的是預測值和實際值的差需小於 ε。

在 ε 合理的情形下 (例如給定一個大得過份的 ε 就是不合理), 如果從式 (1) 能求出解, 這種情形稱為 feasible。 然而大多數的應用中, 因為有雜訊、誤差等等各種因素, 通常不會是 feasible 的情形, 因此我們要加入額外的項, 以容許某些 instances 落在 ε 之外:

$$\begin{aligned} &\text{minimize } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ &\text{subject to } \begin{cases} y_i - w \cdot x_i - b \le \varepsilon + \xi_i \\ w \cdot x_i + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \end{aligned} \tag{2}$$

在式 (2) 中, 每個 training instance 都有其對應的 ξ 及 ξ^*, 用來決定該 training instance 是否可以落在 ε 的範圍之外。 而 C 的作用則如同在 SVM 裡一般, 用來調整訓練模型 (training model) 是否過份或不足調適資料 (overfitting 或 underfitting)。 當問題定義清楚後, 下一步就是來仔細想想, 該怎麼解決這個問題了!

3 ε-SVR

3.1 Dual Problem

在上一章最後我們看到 ε-SVR 問題的其本定義, 如果大家還記得 SVM 的求解過程的話, 應該不難猜到我們打算用什麼神兵利器來處理 SVR。 沒錯, Lagrange multiplier 是我們的好朋友 :-) 利用 Lagrange multiplier, 我們可以將式 (2)

寫成 Lagrange function:

$$
\begin{aligned}
L = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) - \sum_{i=1}^{l}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\
- \sum_{i=1}^{l}\alpha_i(\varepsilon + \xi_i - y_i + w\cdot x_i + b) \\
- \sum_{i=1}^{l}\alpha_i^*(\varepsilon + \xi_i^* + y_i - w\cdot x_i - b)
\end{aligned}
\tag{3}
$$

因此求解式 (2) 即相當於求解

$$
\begin{aligned}
&\min_{w,b,\xi}\max_{\alpha,\eta} L \\
&\text{subject to } \alpha,\eta \geq 0
\end{aligned}
\tag{4}
$$

由於這個問題是一 convex optimization problem, 因此 min 和 max 可交換。 我們先考慮 min 的情形, 求極值的一般做法即是求所有變數偏微分為零值之處:

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0 \tag{5}$$

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)x_i = 0 \tag{6}$$

$$\frac{\partial L}{\partial \xi_i^{(*)}} = C - \alpha_i^{(*)} - \eta_i^{(*)} = 0 \tag{7}$$

將式 (5), (6), (7) 改寫成 $\eta_i^{(*)} = C - \alpha_i^{(*)}$ 及 $w = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)x_i$ 代入式 (4) 中, 可將 w, b, ξ 皆消去只留下 α, η:

$$
\begin{aligned}
&\max_{\alpha,\eta} -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)x_i\cdot x_j - \varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) \\
&\text{subject to } \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C]
\end{aligned}
\tag{8}
$$

推導過程中我們可看出一些性質, 首先 w 可表示成 x 的某種線性組合; 另一方面, 我們從 $f(x) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)(x\cdot x_i) + b$ 可發現計算 $f(x)$ 的複雜度和 x 的維度無關, 只和 Support Vector ($\alpha_i - \alpha_i*$ 不為零的 instances) 的個數有關。 而式裡的 $x\cdot x_i$ 部份如同 SVM, 可用某個 kernel function 來取代, 將線性的 SVR 搖身一變成為非線性系統, 這部份將在後面略加介紹。

在此, 我們將不介紹如何求解式 (8), 因為我還沒看懂 ..., 我們先介紹如何來找 b 的值。

3.2 Computing b

根據 Karush-Kuhn-Tucker (KKT) conditions, SVR 的 dual problem 只有在下述條件成立時才有解:

$$\begin{aligned} &\alpha_i(\varepsilon + \xi_i - y_i + w \cdot x_i + b) = 0 \\ &\alpha_i^*(\varepsilon + \xi_i^* + y_i - w \cdot x_i - b) = 0 \end{aligned} \tag{9}$$

$$\begin{aligned} &(C - \alpha_i)\xi_i = 0 \\ &(C - \alpha_i^*)\xi_i^* = 0 \end{aligned} \tag{10}$$

$$\alpha_i \alpha_i^* = 0 \tag{11}$$

從上述條件我們可以發現, 僅當 $\alpha_i^{(*)} = C$ 時, $\xi_i^{(*)}$ 才有可能不為零, 因此 x_i 才有可能落於 ε 之外。 再來, $\alpha_i \alpha_i^* = 0$ 說明 α_i 和 α_i^* 不可能同時不為零。 因此我們可歸納出:

$$\varepsilon - y_i + w \cdot x_i + b \geq 0 \text{ and } \xi_i = 0 \qquad \text{if } \alpha_i < C \tag{12}$$

$$\varepsilon - y_i + w \cdot x_i + b \leq 0 \qquad \text{if } \alpha_i > 0 \tag{13}$$

我們可依樣畫葫蘆寫下 α_i^* 的情形。 結合上述條件, 我們可決定 b 的範圍:

$$\max\{-\varepsilon + y_i - w \cdot x_i | \alpha_i < C \text{ or } \alpha_i^* > 0\} \leq b \leq \max\{-\varepsilon + y_i - w \cdot x_i | \alpha_i > 0 \text{ or } \alpha_i^* < C\} \tag{14}$$

如果 b 是離散的, 測式該範圍內所有可能的值不失為一可行之法。 而若 b 非離散, 則需以其它方式求之, 恕筆者無力介紹。

3.3 Kernel Function

說到 SVR, 自然不能不提 kernel function。 前面提過, $x \cdot x_i$ 可改寫成一 kernel function $k(x, x')$, 令 $k(x, x') = \Phi(x) \cdot \Phi(x')$, 其中 $\Phi(x)$ 為一將 d 維資料投影到其它維資料的對應, 例如 $\mathbb{R}^2 \to \mathbb{R}^3$。 透過 kernel function, SVR 跨過線性的門檻, 威力大大提升。

然而並非所有雙變數的 function 都可拿來當作 kernel function, 之所以能成為 kernel function, 必須滿足上述的條件, 亦即存在 $\Phi(x)$ 使得 $k(x, x') = \Phi(x) \cdot \Phi(x')$。

一般言而先找 $\Phi(x)$ 再來找 $k(x,x')$ 較為容易, 但我們可利用 Mercer's Condition 來判斷一個雙變數 function 是否是 kernel function:

Theorem 1. *$k(x,x')$ is kernel function, if and only if for any $g(x)$ such that $\int g(x)^2dx$ is finite, then $\int k(x,x')g(x)g(x')dxdx' \geq 0$*

上面這段定理簡單的說就是, 如果我們找到某個 function $k(x,x')$, 想檢查 $k(x,x')$ 是不是 kernel function, 則我們必須證明對於「所有的」$g(x)$ 來說, 如果 $\int g(x)^2dx$ 是有限的, 則 $\int k(x,x')g(x)g(x')dxdx' \geq 0$。看起來並不容易, 不過我們會用一個例子來說明。

Example: $k(x,x') = (x \cdot x')^n$ 是 kernel function。

證明:

$$\begin{aligned}
&\int k(x,x')g(x)g(x')dxdx' = \int (x \cdot x')^n g(x)g(x')dxdx' \\
&= \int (x_1x'_1 + \ldots + x_dx'_d)^n g(x)g(x')dxdx' \\
&= \int \left[\sum_{r_1+\ldots+r_d=n} \frac{n}{r_1!r_2!\cdots r_d!} (x_1^{r_1} \ldots x_d^{r_d} x'^{r_1}_1 \ldots x'^{r_d}_d) \right] g(x)g(x')dxdx' \\
&= \sum_{r_1+\ldots+r_d=n} \frac{n}{r_1!r_2!\cdots r_d!} \left[\int (x_1^{r_1}x_2^{r_2} \ldots x_d^{r_d}) g(x)dx \right]^2 \geq 0
\end{aligned} \tag{15}$$

像這個例子算是最簡單的介紹, 其它 kernel function 要證明就不是那麼容易了。