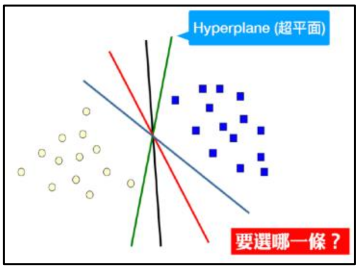
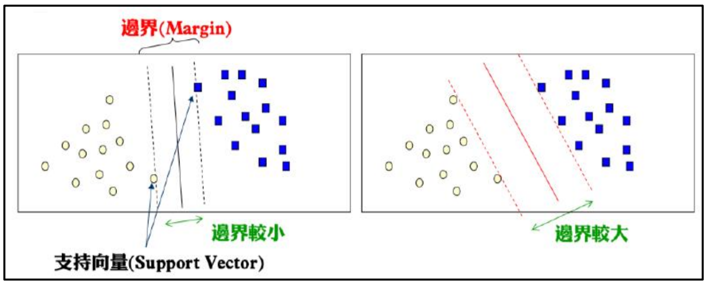
SVM (支持向量機)

原理:

一種監督式的學習方法，用統計風險最小化的原則來估計一個分類的超平面(hyperplane)，其基礎的概念非常簡單，就是找到一個決策邊界(decision boundary)讓兩類之間的邊界(margins)最大化，使其可以完美區隔開來。



- 想找到分界線，必須先找出距離另一組資料最近的邊緣資料點，因為分界線由這些邊緣資料點所「支撐」，所以他們被稱為支持向量(Support Vectors)。
- 選擇讓邊界較大者的分隔線，因為其推論錯誤率優於邊界較小者，因為邊界如果太小，任何些微的變動都會引起顯著的影響。

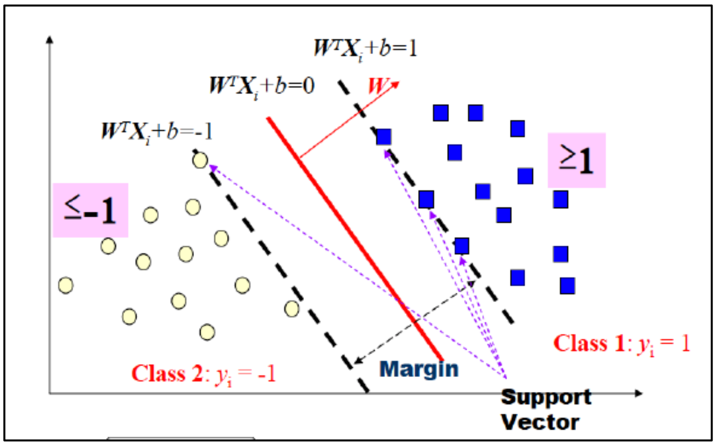


數學式:

假設一個二元線性分類問題，期望能找到一條具有最大 Margin 的 Hyperplane 之線性方程式: $W^TX_i+b=0$，能將 $y_i=-1$ 及 $y_i=1$ 的訓練資料分別落在此直線的兩側。

- 訓練資料表示成(X_i, y_i), $i=1,\cdots,N$
- X_i 為 n 維空間中的一個樣本，以向量方式表示 $X_i=(x_{i1}, x_{i2}, \cdots, x_{in})$
- $y_i\in\{-1,1\}$表示類別，為單一數值
- W、X_i: 向量；b、y_i: 單一數值
- 未知參數: W、b；已知參數: X_i、y_i

由於法向量 W 和截距 b 皆會變動。因此希望找出相對應的法向量 W 和截距 b，所構成的直線 $W^TX_i+b=0$ 可使得 Margin 最大。



- 圖示中兩條平行線的距離為：$\frac{2}{\|W\|}$。

  $W^TX_i+b = 1 \Rightarrow w_1x_1+w_2x_2+(b-1) = 0$

  $W^TX_i+b = -1 \Rightarrow w_1x_1+w_2x_2+(b+1) = 0$

  $$\frac{|(b-1)-(b+1)|}{\sqrt{w_1^2+w_2^2}} = \frac{2}{\sqrt{w_1^2+w_2^2}} = \frac{2}{\sqrt{W^TW}} = \frac{2}{\|W\|}$$

- 為了要求得距離$\frac{2}{\|W\|}$最大化，就類似於將 f(W)=$\frac{\|W\|^2}{2}$最小化。

  原始最佳化問題如下：

  Min. $\frac{\|W\|^2}{2}$

  Subject to $y_i\,(W^TX_i+b) \geq +1$, i = 1 … N

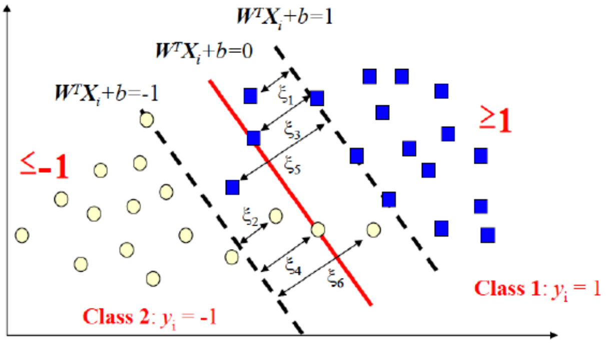
  可用 Karush-Kuhn-Tucker (KKT) Conditions Method 求出最好的 W* 和 b*（直接將 Lagrangian 函式偏微分該參數，再轉換成二次式(Quadratic Programming)去求解)。

- 一旦找到分割超平面的參數，當有一側試資料 X_{new} 時，就可以被歸類如下：

  $$f(X) = \mathrm{sgn}(W^* \times X_{new} + b) = \mathrm{sgn}(\Sigma\, u_i y_i X_i \times X_{new} + b)$$

  若 f(X)=1，則測試資料 X_{new} 為正類別；否則為負類別。

Soft Margin SVM



實的資料不太可能有可以完美分類的案子，因此我們在 training 的時候可以容忍一些 data 落到邊界之內(下圖範例)，此類的 SVM 稱為 soft-margin SVM。
如圖所示，理論上如存在一個理想的超平面，y_i (W^TX_i+b)應該要>=1，但圖中有些落在邊界區的點 y_i $(W^TX_i+b)<1$，但因為可以容忍誤差的關係，所以只要在式子右邊減去一個值，就可以達到<=1 的情形。在公式內加入 slack variable(ξ_i)就可以容忍這筆資料可以落在 margin 內的參數，值愈大容忍的範圍愈大。

但因為這個 slack variable 是用在限制式內，所以在目標函數也需要考慮進去，因此將所有點的 slack variable 相加並且給予一個權重/懲罰參數(C，要為正值的實數)，用來調整 slack variable 在目標函數的重要性。

$$\min_{w,\xi_i}\frac{1}{2}w^Tw+C\sum_{i=1}^{n}\xi_i$$

$$subject\ to\quad y_i(w^T\boldsymbol{x}_i+b)\geq 1-\xi_i,\xi_i\geq 0,\forall i=1,\ldots,n$$

因為未知參數在限制式內，無法直接去解此最佳化問題，因此在解「 有條件的最佳化問題」時，有時需要把原本的問題轉換成對偶問題(Dual Problem)後，會比較好解。因此我們用 Lagrangian dual function (因為有兩種限制式所以 Lagrangian parameters 為 $\alpha_i>=0$ 和 $\beta_i>=0$)將原最佳化問題轉換成 Lagrangian 函式:

$$J=L(\boldsymbol{w},b,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\xi})=\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}+C\sum_{i=1}^{n}\xi_i-\sum_{i=1}^{n}\alpha_i(y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)-1+\xi_i)-\sum_{i=1}^{n}\beta_i\xi_i$$

根據 Karush-Kuhn and Tucker (KKT) condition 的理論，如果我們要得到參數的最佳解，則直接將 Lagrangian 函式偏微分該參數

$$\frac{\partial L(\boldsymbol{w},b,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\xi})}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i$$

$$\frac{\partial L(\boldsymbol{w},b,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\xi})}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\frac{\partial L(\boldsymbol{w},b,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\xi})}{\partial \xi_i} = 0 \Rightarrow C - \alpha_i - \beta_i = 0, \forall i = 1,\ldots,n$$

根據 Lagrangian 求得的解，將最佳化問題做二次式(Quadratic Programming)轉換

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$$

$$subject\ to \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

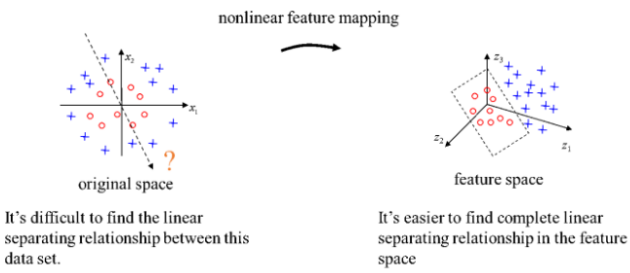
$$0 \le \alpha_i \le C, \forall i = 1,\ldots,n$$

Kernel SVM

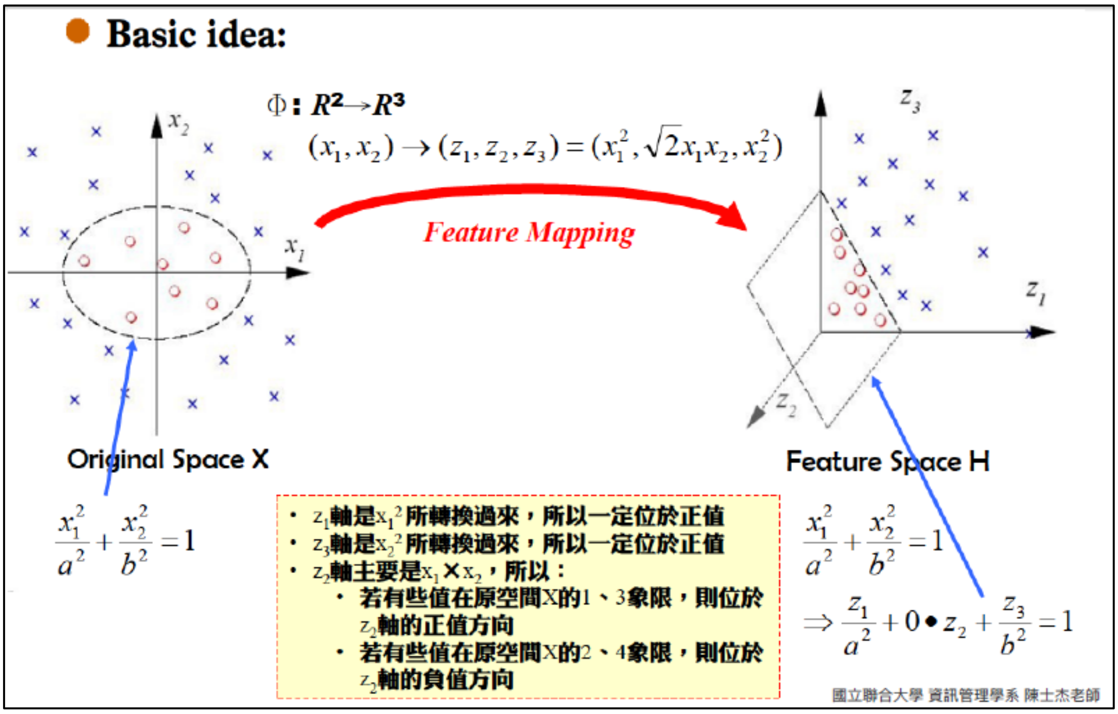
Kernel trick 在機器學習的角色就是希望當不同類別的資料在原始空間中無法被線性分類器區隔開來時，經由非線性投影後的資料能在更高維度的空間中可以更區隔開，同時卻是利用原始空間資料進行分析的方法。

- 特徵映射:

無法在原始空間(Rd)中適當的找到一個線性分類器將兩類區隔開，這時後此需要找到一個非線性投影(φ)將資料進行轉換到更高維度的空間，此時在高維度的空間中只需要一個線性分類器，hyperplane 就可以完美分類。而這個更高維度的空間則稱為 Hilbert space(H)。



● 特徵函數(φ):



特徵函數(φ)取得不易。因為不容易得知哪一種函數可讓原本無法線性區分的原始資料空間，映射成可線性區分之特徵空間。另外，特徵空間的維度會急遽增加，此時解決高維度最佳化問題會很耗時。因此需要 kernel 函數來輔助。

● Kernel Method

高維度空間內積

$$
\begin{aligned}
&\langle \Phi(x_1, x_2), \Phi(x'_1, x'_2) \rangle = \langle (z_1, z_2, z_3), (z'_1, z'_2, z'_3) \rangle \\
&= \langle (x_1^2, \sqrt{2}x_1x_2, x_2^2) \bullet (x'^2_1, \sqrt{2}x'_1x'_2, x'^2_2) \rangle \\
&= x_1^2x'^2_1 + 2x_1x_2x'_1x'_2 + x_2^2x'^2_2 = (x_1x'_1 + x_2x'_2)^2 = (\langle X, X' \rangle)^2 = \underset{\text{Kernel Function}}{k(X, X')}
\end{aligned}
$$

高維度空間距離

$$
\begin{aligned}
\|\Phi(\mathbf{X}) - \Phi(\mathbf{X}')\|^2 &= \|\Phi(\mathbf{X})\|^2 + \|\Phi(\mathbf{X}')\|^2 - 2\langle \Phi(\mathbf{X}), \Phi(\mathbf{X}') \rangle \\
&= \Phi(\mathbf{X})^T\Phi(\mathbf{X}) + \Phi(\mathbf{X}')^T\Phi(\mathbf{X}') - 2\Phi(\mathbf{X})^T\Phi(\mathbf{X}') \\
&= <\Phi(\mathbf{X}), \Phi(\mathbf{X})> + <\Phi(\mathbf{X}'), \Phi(\mathbf{X}')> - 2<\Phi(\mathbf{X}), \Phi(\mathbf{X}')> \\
&= k(\mathbf{X}, \mathbf{X}) + k(\mathbf{X}', \mathbf{X}') - 2k(\mathbf{X}, \mathbf{X}')
\end{aligned}
$$

高維度空間角度

$$
\begin{aligned}
&\langle \Phi(X), \Phi(X') \rangle = \|\Phi(X)\| \times \|\Phi(X')\| \cos\theta \\
&\Rightarrow \cos\theta = \frac{\langle \Phi(X), \Phi(X') \rangle}{\|\Phi(X)\| \times \|\Phi(X')\|} \\
&\qquad = \frac{\langle \Phi(X), \Phi(X') \rangle}{\sqrt{<\Phi(X), \Phi(X)>} \times \sqrt{<\Phi(X'), \Phi(X')>}} \\
&\qquad = \frac{k(X, X')}{\sqrt{k(X, X)} \times \sqrt{k(X', X')}}
\end{aligned}
$$

由上述推導得知，特徵空間的角度，可以由原空間相對應的點之內積、距離、角度，透過 Kernel Function 求得。同樣地，映射函數 φ 在此也顯得不重要了。因此 kernel method 特別之處在於，演算過程沒有明顯地用到特徵空間上資料的座標值，思想上用到了特徵空間，計算結果也映射到特徵空間，但實際的計算過程是在原始空間。該方法保有特徵空間運算的優勢，又可不用直接面對映射函數 φ 的尋找工作，以及在高維度計算的不方便性。

- Kernel trick **是一個在特徵空間中使用原始屬性集合來進行內積計算的方法。即** $k(X_i, X_j) = <\Phi(X_i), \Phi(X_j)>$
- **可用在非線性的支援向量機的問題上：**
  - **不用知道正確的映射函數。**
  - **使用核函數計算內積，比起使用轉換後的屬性集合來得容易。**
  - **因在原始的空間中進行計算，可避免維度過高所造成的問題。**

常用的 kernel 函數:

- **Linear Kernel (線性核函數)**
  - $k(X_i, X_j) = <X_i, X_j>$
- **Polynomial Kernel (多項式核函數)**
  - $k(X_i, X_j) = (<X_i, X_j>+1)^p$, $P \in Z^+$
- **Gaussian Kernel (高斯核函數)** Radial Basis Function kernel (RBF)
  - $k(X_i, X_j) = e^{[-\|X_i - X_j\|^2/(2\sigma^2)]}$

上述多項式 kernel 函數中有一個參數 P，而高斯 kernel 函數有一個參數 σ。若使用這些函數時，參數設定值不同，則所對應到的特徵空間也會不同。

SVM 的優缺點

- 優點:

SVM 對不同類型的數據集都有不錯的表現

- 缺點:

對資料預處理以及參數調節的要求很高

三個重要的參數:

1. 核函數 2. 核函數的參數（RBF 的 gamma 值）3. 正規化的參數（C）

Logistic Regression & SVM

- 實務上，linear logistic regression 和 linear SVM 常會產生相似的結果
- 但因 logistic regression 試圖最大化訓練數據集的條件概似(conditional likelihood)，所以較容易受離群值影響；而 SVM 主要在意那些非常接近決策邊界的點(支援向量)。
- logistic regression 的優點是一個簡單的模型，所以更容易實作，也很容易完成更新處理(有利利於串串流數據)

參考資料

1. 機器學習-支撐向量機(support vector machine, SVM)詳細推導 https://medium.com/@chih.sheng.huang821/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E6%94%AF%E6%92%90%E5%90%91%E9%87%8F%E6%A9%9F-support-vector-machine-svm-%E8%A9%B3%E7%B4%B0%E6%8E%A8%E5%B0%8E-c320098a3d2e
2. 機器學習: Kernel 函數 https://medium.com/@chih.sheng.huang821/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-kernel-%E5%87%BD%E6%95%B8-47c94095171
3. Class Handout, Lee, Chia Jung professor, MDM64001, School of Big Data Management, Soochow University