# Support Vector Machine with SMO

李振維
Enoch Lee

## Abstract

## 1. Introduction

Support vector machine(通稱 SVM) 是一種分類、回歸的方法，是在機器學習 (Machine learning) 的一種監督式學習的方法，可以同時運用於線性及非線性的演算法，在實際的應用方面有數字辨識、人臉辨識... 等等。

- 監督式學習 Supervised: 在放入樣本給機器做學習的時候會一併告訴機器當次的輸出的結果應該為何。
  - 例：一台機器需要幫我們分類進來的水果是蘋果還是橘子，帶給訓練樣本的時候，我們會跟他說這一次進去的水果應該要幫我們分到哪一類。
- 非監督式學習 Non-Supervised: 在放入樣本給機器做學習的時候不提供當次的結果，由機器幫我們自動分類。
  - 例：一台機器要幫我們分類，但是給訓練樣本的時候不告訴它當次的輸出結果應該為何，而是跟機器說我們要分幾類，機器會依照特徵來將樣本分成要的類別數量。

### 1.1. Support Vector Machine

SVM 的基本概念很簡單，就是在一個平面上畫上一條線來分左右邊，來達到將輸入的資料數據做分類，而 SVM 的重點就是要找到那條線，而 SVM 有趣的地方就是有很多條線都可以將我們的資料分得好好的 (如圖 1)，而就是要找出哪一條線是 SVM 要的

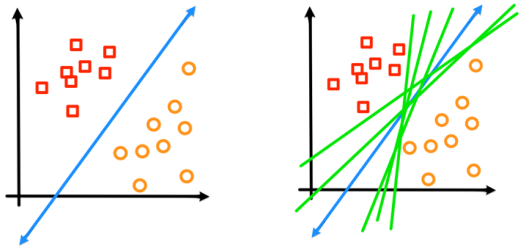


图 1. 搜尋最優分類線

也就是希望 SVM 的線能幫我們分類的越準確越好，一般來說輸入值都會有些微的誤差，而輸入數據如果有誤差的話 SVM 還是能夠歸類在對的一邊，也就是說這條線能夠離與他最近的數據之間的間距越遠越好，而在這些數據上面都會有一條與理想中 SVM 的線平行的線，而 SVM 的線與這些線之間的間距稱它為“Margin”。

1)，我們要怎麼知道哪一條線是我們的 SVM 要的?

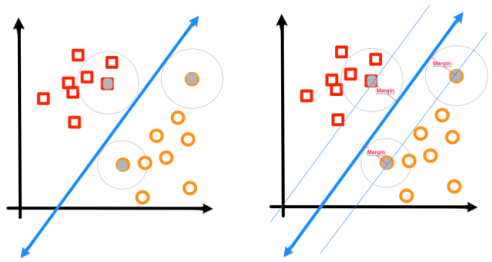


图 2. 點誤差及 Margin

所以 SVM 所要求線的條件就是有越大的間距越好，所以當有多種可以選擇時 (MarginA、MarginB) 會選擇有最大 Margin 的線，而這條線的 Margin 是由離他最近的單個或是多個數據點所限制出來的 (如圖 3)，而這樣的數據點也有專屬於它的名稱為“Support

vector”，這也就是這樣的方式叫做 SVM 的原因。

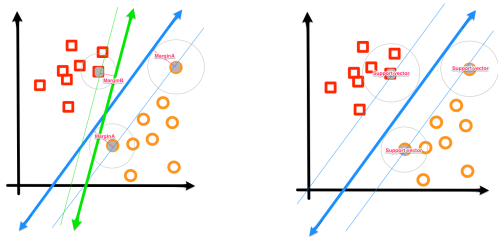


圖 3. 選擇最大 Margin

### 1.2. In-depth Knowledge

在前面介紹 SVM 的時候有說到要求的是那條最佳的分割線，先來說一下那條線是要用什麼樣的公式表示好了，中間那條線的公式為

$$f(x) = W^T X - b = 0$$

在更新 SVM 找出最佳的線時，透過變動 $W$ 來讓線做旋轉，透過變動 b 來讓現做水平的偏移:

- b 是 SVM 線的偏權值 (Bais)
- W 是 SVM 線的法向量，在神經網路也稱作這個為權重

而 SVM 是屬於二分法，因此將資料劃分成 1/-1 這兩類，因次就多出了兩條輔助線

$$f(x) = W^T X - b = 1$$
$$f(x) = W^T X - b = -1$$

接著講為什麼要找的最大的 Marge 的公式，首先確認大家都知道 $W^T X - b$ 這個當 $W$、$X$ 為多維坐標 (矩陣) 時，就代表此方程式的圖為超平面，再來是 $W$ 為該超平面的法向量，也就是垂直平面於的向量。首先證明 $W$ 為超平面的法向量：來看一下超平面 $W^T X - b = 0$，該平面上面有兩個點，分別為 $X_1$ 和 $X_2$，而這兩點都符合超平面 $W^T X - b = 0$，因此帶入後得知 $W^T X_1 - b = 0$ 和 $W^T X_2 - b = 0$，把兩式相減後可以得到 $W^T (X_1 - X_2) = 0$，可以發現 $(X_1$ - $X_2)$ 就是在超平面上的向量，而它跟內積為零更證明了是垂直於超平面的法向量。

再來已經確定為超平面的法向量，那麼當在平面外有一個點 $X$ 到超平面任一點 $X_1$ 的向量對做投影的距離，就是該點到超平面的距離 (如圖 4)。

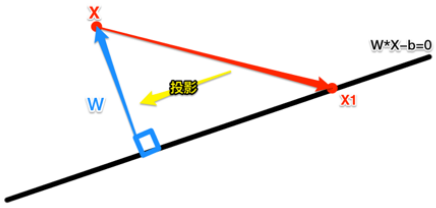



圖 4. 點到線的投影距離

$$\text{向量投影公式} = \frac{\vec{A} * \vec{B}}{\|A\|} = \frac{\vec{W} * (X - X_1)}{\|W\|} = \frac{\left|\vec{W} * X - b\right|}{\|W\|}$$

又希望我們的超平面可以將所有的點 $X_i$ 都分的正確，意思就是說 output 出來的值 $W^T X_i - b$ 要跟預期的值 $y_i$ 同號，因此 $y_i(W^T X_2 - b) > 0$ 所以就得到要求的極值以及極值的限制條件。

期望求得：$\underset{w,b}{max} \frac{\left|\vec{W} * X - b\right|}{\|W\|}$
限制條件：$y_i(W^T X_2 - b) > 0$

要求的點 $X$ 要是 Support Vector，因此該點會在 $W^T X_2 - b = 1/-1$ 上面，所以 $\left|\vec{W} * X - b\right|$ 一定為 1，而且因為點 $X$ 為最接近線的 Support Vector，所已其他的點帶進來後一定會 $\geq 1/-1$，就可以得到:

期望求得：$\underset{w,b}{max} \frac{1}{\|W\|}$
限制條件：$y_i(W^T X_2 - b) \geq 1$

因為期望求得的 Margin 只是其中一邊 Support Vector 到 SVM 線的長度，所以要將其變兩倍 $\frac{2}{\|W\|}$。到這邊已經將期望部分規則大致出來了，但是因為求 MIN 比 MAX 來得簡單，因此希望求得的部分可以將 MAX 轉化成 MIN，這邊把它上下顛倒，就可以將求 Max 換成求 MIN，再來去除比較複雜的 $\sqrt{}$，因為範數的關係，我們的 $W$ 上會出現 $\sqrt{}$，因此將它平方來達到去除根號 (將期望求得平方並不會影響到要求的 W)，做完以上動作後即可得到:

期望求得：$\underset{w,b}{min} \frac{\|W\|^2}{2}$
限制條件：$y_i(W^T X_2 - b) \geq 1$

算式到這邊已經可以用 QP(quadratic programming) 的方式去解，舉例來說：有三個點 X1=[2,1] X2=[1,0] X3=[3,0]，y 分別是-1 ，-1，1，未知的 W=[w1,w2]，將三個點帶入限制式可以得到下面三個式子:

$$\begin{aligned} -2w_1 - 1w_2 + b &\geq 1 \\ -1w_1 - 0w_2 + b &\geq 1 \\ 3w_1 + 0w_2 - b &\geq 1 \end{aligned}$$

透過聯立解及交集後，可以得到的 W=[1,-1]、b=2，但是 QP 方法實在是效率太低了，當為正定時，用橢圓法可在多項式時間內解二次規劃問題。當為非正定時，二次規劃問題是 NP 困難的（NP-Hard）。即使 Q 只存在一個負特徵值時，二次規劃問題也是 NP 困難的，可以試想當資料有幾萬筆、每筆的特徵又有幾千個，這樣的做法是非常非常消耗效能的。

## 2. Slack Variable

SVM 在不斷修正變化的過程中... 線可能會不斷地偏移！甚至變得無法分出正確的選擇，因為有可能在輸入值的時候輸入錯誤！導致我們給的 y 給錯，因而造成許多不可挽回的錯誤，畢竟人有失足馬有亂蹄，因此，為了避免這種情況發生，將太誇張、差太多的值給剔除在調整的計算中。

### 2.1. Significance

所以在這過程中，可能會有偏差過大或是錯誤的數據出現 (如圖5)，導致計算出來的線並不是最理想的線，因次需在這邊加入 1. 鬆弛變數 ($\xi$)：這個變數代表的意思類似容錯率的概念，因此這個數值沒有負數，而這個鬆弛變數是為每一個點所客製化的，因此他不會是同一個數。2. 懲罰因子 (C)：這個因子代表的意思是你有多重視離群點，所以當我們懲罰因子越大就代表我們越重視離群點。這個概念有點像是當我的數據出現錯誤時，要用多大力道去將分割線打到能越接近數據正確的地方。這樣的話就能夠透過加入來適時的容錯，也藉由這兩個來幫助找到最理想的線 原本的

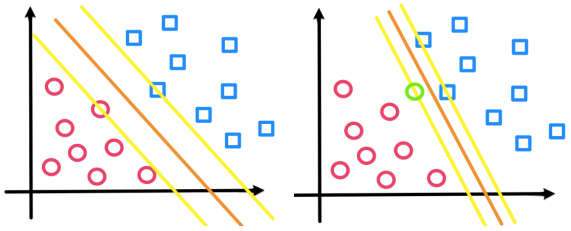


圖 5. 偏差過大的點導致 SVM 線不理想

SVM 因為沒有加入鬆弛變數及懲罰因子的部分，所以他所要求的值變得非常的硬性、非常直接並且強烈的劃分出來，因此在沒有加入這兩項時的 SVM 有一個稱呼”硬間隔”分類；在加入這兩項後，因為對錯誤的數據或離群點有了兼容得特性出現，因此不會再這麼的硬性劃分，也就變成了”軟間隔”分類。

### 2.2. Variety

已經說明了什麼是鬆弛變數及懲罰因子，了解他們是做什麼用以及對我們的 SVM 有什麼樣的幫助，接著將鬆弛變數加入式子中，上面有說過鬆弛變數是為每個點所客製的所以代表每個點都有自己的一個鬆弛變數，而這個鬆弛變數的目的就是要將原本錯誤的部分分到正確的上面，因此就將鬆弛變數加入限制式中，因為加入了鬆弛變數的關係而導致我們將每一個值都是正確的，但是這樣會變成無限制的擴充，因為只有加在限制式中，將限制式放寬了，但是要求的極小值卻沒有變，所以接下來就要再求極小值那邊做變化，在這邊將所有的鬆弛變數相加後再給他一個懲罰因子，因為每加入一個鬆弛變數，就必須支付一次代價。

期望求得：$\min_{w,b} \frac{\|W\|^2}{2} + C\sum_i^n \xi_i$
限制條件：$y_n\left(W^T X_n - b\right) \geq 1 - \xi_n, \xi_n \geq 0$

這樣加入到極值部分其實要想到剛剛一直說到的幾件事：

1. 鬆弛變數 ($\xi$) 是客製化的變數，假如 A 點是被正確的劃分的話，A 點鬆弛變數就是 0；假如 A 點沒有被正確劃分且算出來的結果是-1 的話，A 點的鬆弛變數就是-2。

2. 懲罰因子 (C) 會與“相加後的鬆弛變數”相乘並放在極小值的求解地方是為了來表示有多重視離群值。

上面兩件事可以知道當越重視離群值的話 C 就會越大，代表說當他如果有離群值的話，會影響到整個最小值的部份，有可能導致原本是可以很正確的分出，卻因為幾個離群值而導致偏了方向。

## 3. Lagrange Multiplier

Lagrange Multiplier 又稱拉格朗日乘數法，其目的是在一個函式且有約束條件下要求極值 (最大/最小值)。此方法會引入一個以上新的未知數，而稱這個未知數為拉格朗因子。看一下他要怎麼表示：

有一函式：$f(x,y)$，在一限制式：$g(x,y)=C$ 下要求極值，可以將其轉換成 Lagrange Multiplier 的函式如下：

$$L(x,y,\alpha)=f(x,)+\alpha\{g(x,y)-C\}$$

當 Lagrange Multiplier 的函式達到極值時，函式 $L(x,y,\alpha)$ 會等於 $f(x,)$，因為這時 $\alpha\{g(x,y)-C\}$ 等於 0。而 $f(x,)$ 極值可以透過幾個方式來求出來，首先在極值時，函式及限制式會有一點交會點且他們會有相同的切線斜率，因此可以利用這個特性來使用偏導數為 0 的方式來求出，以上面的例子來說：

$$\begin{aligned}\frac{\partial L(x,y,a)}{\partial x} &= 0\\ \frac{\partial L(x,y,a)}{\partial y} &= 0\end{aligned}$$

透過這兩個加上原本的限制式 $g(x,y)=C$ 就能求出來，也可以求出極值。

而當有多個限制式時，就會增加多個 Lagrange 因子來幫助求極值，以上面的例子來說，當限制式擴充為兩個，分別為：$g(x,y)=C$、$h(x,y)=d$，而 Lagrange Multiplier 的函式就會變成如下 ($\alpha$ 及 $\beta$ 為我們的 Lagrange 因子)：

$$L(x,y,\alpha)=f(x,)+\alpha\{g(x,y)-C\}+\beta\{h(x,y)-d\}$$

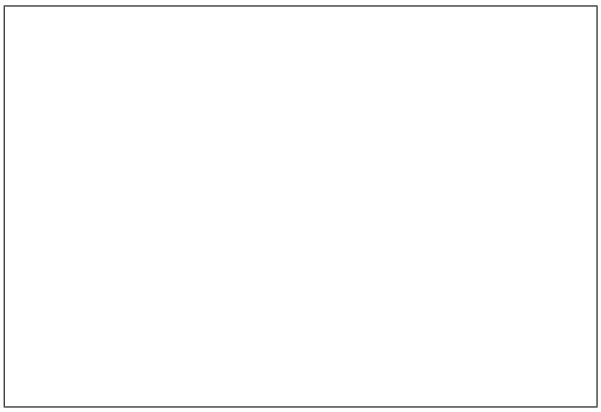
### 3.1. Solving Lagrange Multiplier

先來講 Lagrange Multiplier 的幾何意義，首先假設空間平面中有一個條線 $f(x,)$ 並且在該線有一個限制圓 $g(x,y)=C$ 時，要求這之間的極值 (最大/最小值)，可以看到途中交界的部分 (如圖6)。

图 6. 線與點的交會

在這有兩個交會點，而這交會點在曲線及平面上都有一個共同的切線斜率，而這兩點的斜率是成比例的，這比例 $\nabla f(x,y)=\alpha*\nabla g(x,y)$ 中的 $\alpha$ 就是 Lagrange Multiplier，而 $\alpha$ 可以用下面三個方程式去解。

1. $\frac{\partial f(x,y)}{\partial x}=\alpha\frac{\partial g(x,y)}{\partial x}$(兩邊對做 $X$ 偏微分)
2. $\frac{\partial f(x,y)}{\partial y}=\alpha\frac{\partial g(x,y)}{\partial y}$(兩邊對做 $Y$ 偏微分)
3. $g(x,y)=C$(原本的限制式)



然而上面所說的是一般的 Lagrange Multiplier 的解法，而 SVM 上面要使用的 Lagrange 不太一樣，由於 SVM 的限制式不為等號，因此無法使用上面的解法來解，而要靠 Lagrange Multiplier 的變種 Karush—Kuhn—Tucker 來做到。

### 3.2. Karush—Kuhn—Tucker

Karush-Kuhn-Tucker conditions 一般又稱做 KKT 條件，一般會隨著拉格朗日乘子一起出現，他是只要滿足再一些有規則條件下，一個非線性問題有最優化解法的一個充分必要條件，簡單的來講 KKT 條件就是拉個朗日乘子條件的變種，為了在非線性規劃中用一些條件限制來找出極值，而這些條件限制式有等號、不等號時可以利用 KKT 來解，而 KKT 的條件如下

1. Lagrange 因子 ($\alpha$) 必須 $\geq 0$
2. 偏導數 $=0$
3. 當為最佳解時，帶有 Lagrange 因子的方程式等於 0，且 Lagrange 因子為 MAX
4. 限制式 ($g(x)$) 必須為 $\leq 0$
5. KKT 互補鬆弛 (Complementary slackness)，$\alpha * g(x)=0$

### 3.3. SVM with Lagrange KKT

介紹完了 KKT 後就能和 SVM 做結合來解出要的解，加入 KKT 後的樣子如下：

期望求得：$\underset{w,b}{min}\frac{\|W\|^2}{2}+C\sum_i^n \xi_i$

限制條件：$y_n\left(W^T X_n-b\right)\geq 1-\xi_n, \xi_n\geq 0$

$$L\left(x,b,a\right)=\frac{\|W\|^2}{2}+C\sum_{i=1}^n \xi_i-\sum_{i=1}^n a_i\left[y_i\left(W*X_i-b\right)-1+\xi_i\right]-\sum_{i=1}^n \beta_i\xi_i$$

然後將 Lagrange 對 $W,b,\xi_n$ 做偏導數 =0，可以得到下面三個

1. $\frac{\partial L}{\partial x}=W-\sum_{i=1}^n a_i y_i x_i=0$
2. $\frac{\partial L}{\partial b}=\sum_{i=1}^n a_i y_i=0$
3. $\frac{\partial L}{\partial \xi_n}=C-\alpha_i-\beta_i=0$

將 $L\left(x,b,a\right)$ 展開並將 $\beta_i$ 用 $C-\alpha_i=\beta_i$ 替換後得到如下：

$$\begin{array}{lll}\frac{\|W\|^2}{2} & + & C\sum_{i=1}^n \xi_i \\ \sum_{i=1}^n a_i y_i W^T X_i+b\sum_{i=1}^n a_i y_i+\sum_{i=1}^n a_i & - & \\ \sum_{i=1}^n a_i\xi_i-C\sum_{i=1}^n \xi_i+\sum_{i=1}^n a_i\xi_i & & \end{array}$$

接著用 $\sum_{i=1}^n a_i y_i=0$ 將消去不必要的參數，經過整理後如下：

$$\frac{\|W\|^2}{2}-\sum_{i=1}^n a_i y_i W^T X_i+\sum_{i=1}^n a_i$$

得到比較簡單的式子後，我們將前面 $W$ 的範數展開後將其中一個 $W$ 用 $W=\sum_{i=1}^n a_i y_i x_i$ 替換且將第二項的 $W$ 往前提後可得到如下：

$$\frac{1}{2}W\sum_{i=1}^n a_i y_i X_i-W\sum_{i=1}^n a_i y_i X_i+\sum_{i=1}^n a_i$$

可以發現第一項及第一項可以合併成一項：

$$-\frac{1}{2}W\sum_{i=1}^n a_i y_i X_i+\sum_{i=1}^n a_i$$

再將最後一個 $W$ 用 $W=\sum_{i=1}^n a_i y_i x_i$ 替換並且整理後如下：

$$\sum_{i=1}^n a_i-\frac{1}{2}\sum_{j=1}^n a_j y_j X_j\sum_{i=1}^n a_i y_i X_i$$

接著利用乘法運算 ((x1+x2+x3+x4+...)(x1+x2+x3+x4+...) = x1x1+x1x2+x1x3+x2x1+x2x2+x2x3+ ⋯) 原理將第一項整理後如下：

$$\sum_{i=1}^n a_i-\frac{1}{2}\sum_{i,j=1}^n a_i a_j y_i y_j {X_i}^T X_j$$

因此將原本要求的函式替換成只需要求出最大的 $\alpha$ 就可以得到，而 $W$ 及 $\mathbb{S}\,\alpha$ 之間這樣的關係就是對偶關係，透過轉換利用另外的變數去更動而造成另外一邊的更動，而最後要找出的結果也跟另外一邊有關，這樣的關係就是對偶關係。

$$\underset{a}{Max}\sum_{i=1}^n a_i-\frac{1}{2}\sum_{i,j=1}^n a_i a_j y_i y_j {X_i}^T X_j$$

## 4. Kernel

核函數是為了將低維特徵映射到高維，當樣本在低維空間是無法做線性分割的，所以將它映射到高維後會產生差異，透過這樣的差異特徵在高維空間中能做線性的分割，

### 4.1. definition

在空間中任一點 $X$ 可以找出一個函數 $\varphi$ 可以來將該點 $X$ 映射到高維空間，但是由於映射後的 $\varphi\varphi\left(X\right)^T\varphi\left(Y\right)$ 內積的成本太高、效能較差，因此找出核函數 $K\left(X,Y\right)=\varphi\left(X\right)^T\varphi\left(Y\right)$，這就是核函數的由來。

以幾何面來說，在二維平面中這樣散佈是無法用線性的方式去分割，但是當透過核函數的一些方法映射到三維空間 (如圖7) 後，就可以成功分割了。

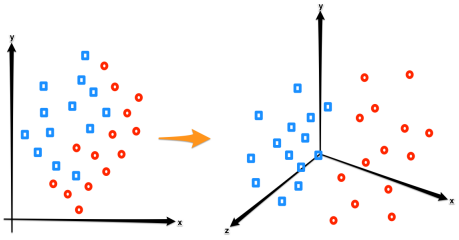


图 7. 二維轉換三維

### 4.2. Using and Kind

在做數據分割時，常常會遇到無法分割的情況，這樣的狀況有幾種可能：

1. 那些數據是無用的數據，導致沒辦法區別
2. 數據的特徵值太少，挑出來的數據太相似
3. 維度太低，導致數據無法分割

可以看到 2 跟 3 好像是一樣的，其實他代表的意思有點不同，維度的部分可以由多個特徵所組合而成，有可能兩個特徵組合成一個維度，就好比說月跟日兩個特徵組合成一個星座這樣的維度。

基本上在做分析的時候都會經過一個挑選數據 (由專家) 的部分，挑選出哪些是要來分析的數據、這些數據應該要怎麼去設定他的值，因為做了這樣的步驟，所以基本上 1 的機率就相對較小；至於數據的特徵值太少無非就是增加特徵值的數量，不然就是使用核函數將數據映射到高維空間讓他們分離，前者部分因為要重新去評估挑選數據部分而導致過程會增加 loading，後者則是需要挑選到適合的方式將數據映射分割。

因此在 SVM 上面融入核函數的概念，把他加入後函數就變成

$$\underset{a}{Max} \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i,j=1}^{n} a_i a_j y_i y_j K_{i,j}$$

核函數的種類有很多，一般來說較常使用的是高斯核 (RBF) 以及線性核 (Linear) 函數，這邊就介紹幾種核函數:

1. 線性核函數
2. 多項式核函數
3. 高斯 (徑向基) 核函數
4. sigmoid
5. 傅立葉核函數

上述列出來的這些是一般比較常見的核函數應用，不過這些之中最常用的還是線性核函數跟高斯 (徑向基) 核函數，至於要怎麼挑選適合的核函數通常有幾種:

1. 在分析數據之前，讓專家去評估這樣的數據是用於哪些方式
2. 用不同的核函數下去跑，看哪個是最接近所要的數據

### 4.3. Mercer’s Condition

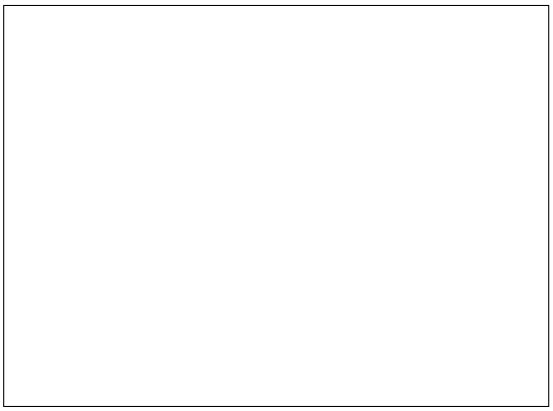
當然並不是所有的方法都能當作核函數來使用，因此只要是符合 Mercer’s Condition 的都可以使用核函數來做映射，所以可以找出自己核函數，而 Mercer’s Condition 如下:

1. 對稱性: $K(X_i, X_j) = K(X_j, X_i)$
2. 正半定性: 若由 $a_{ij} = K(X_i, X_j)$ 組成的矩陣K為正半定矩陣，則對任意非零實數組成的向量 Z 有 Z*K*Z 向量內容皆大於等於 0 的關係

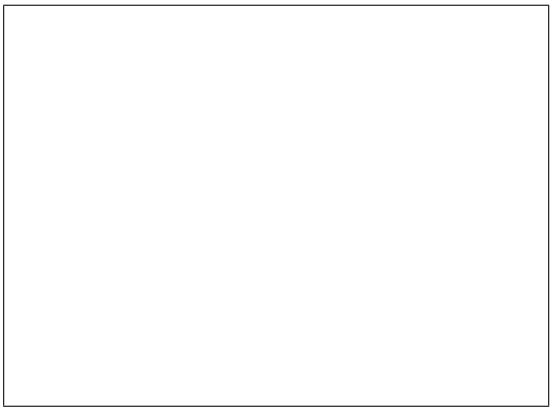
### 4.4. How to do high-dimensional conversion

核函數可以幫我們把原本的低維無法分割的特徵映射到高維來分割的概念已經知道，這邊舉兩個例子讓更能體會如何映射到高維。

1. 多項式（二維　三維）:



2. 高斯（二維　無限維）:



上面的兩個例子可以看到透過這樣的核函數將原本是二維向量轉換成多維向量，透過這樣的瞭解，應該可以很清楚地知道為什麼可以透過核函數來做到提高維度的部分了，SVM 也是透過核函數的轉換，讓原本可能在二維平面中無法分割的數據，轉換成在高維中可以做線性分割。

## 5. Sequential Minimal Optimization

SVM 原本採用 QP(quadratic programming) 來解出全域最佳解, 因為 QP 方式是採用二次規劃的方式找出所有特徵中最佳的結果, 但是意味著效能消耗十分龐大, 而現在來介紹 Sequential Minimal Optimization(之後簡稱 SMO) 這個 SVM 優化方法。

### 5.1. Introduction

SMO 是採用類似梯度下降的方式透過修正 $\alpha$ 及對偶性質來更新權重, 不一樣的是由於一些限制, 因此他是採兩個特徵來修正, 而這個限制來自於先前提到用 Lagrange 對 $b$ 做偏微分時的限制式 $\sum_{i=1}^{n} a_i y_i = 0$, 因為此限制式導致無法只更新一個 $\alpha$。

接著我們來看一個關係, 前面已經知道 $0 \geq \alpha \geq C$ 的關係, 這邊把他切成三個部分來看, 然後透過先前提到的 KKT 互補鬆弛 (Complementary slackness) 得知

$$a_i * \left[-y_i \left(W^T X_i - b\right) + 1 - \xi\right] = 0$$
$$-\beta_i * \xi = 0 \quad \Rightarrow (a_i - C) * \xi \quad = \quad 0$$

1. $a_i = 0$ 時, $y_i \left(W^T X - b\right) \geq 1$
   由互補條件的 $\beta$ 可以知道, $a_i = 0$ 時鬆弛變數 ($\xi$)=0, 鬆弛變數 ($\xi$) 帶入互補鬆弛的第一條限制式裡即可得知上述結果。

2. $a_i = C$ 時, $y_i \left(W^T X - b\right) \leq 1$
   由互補條件的 $\beta$ 可以知道, $a_i = C$ 時鬆弛變數 ($\xi$) 為 >0 的任一數, 再互補條件的條件兩邊將消去可以得知

   $$y_i \left(W^T X_i - b\right) + \xi = 1 \quad , \quad \xi \geq 0$$

   當鬆弛變數 ($\xi$) 最小 =0 時,$y_i \left(W^T X - b\right) = 1$, 所以當鬆弛變數 ($\xi$) 越大 $y_i \left(W^T X - b\right)$ 越小, 因此可以得知上述結果。

3. $0 < a_i < C$ 時, $y_i \left(W^T X - b\right) = 1$
   由互補條件的 $\beta$ 可以知道, $0<a_i < C$ 時鬆弛變數 ($\xi$)=0, 再互補條件的第一條限制式將鬆弛變數 ($\xi$) 帶入並為 0 可以知道, $y_i \left(W^T X - b\right) + 1 = 0$, 因此可以得知上述結果

上面可以知道 SMO 是一次更新兩個資料點, 並且也知道透過 KKT 條件的變化下得出的最佳解的 Lagrange 因子對應的限制式符合的範圍, 接下來會利用前面的部分延續推導出 SMO 的式子, 並且教如何使用最後推出的公式以及整個 SMO 流程。

### 5.2. Derivation

$$\underset{a}{Max} \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i,j=1}^{n} a_i a_j y_i y_j K_{ij}$$

這是目前加入 Lagrange 後的期望求得式子, 前面有說過求 Min 比求 Max 來的好求, 因此將它轉換成找出 Min 的 $\alpha$, 在這邊將這個式子乘上一個-1 就可以將原本最佳結果是求 Max 的 $\alpha$ 改成求 Min 的 $\alpha$

$$\underset{a}{Min} \quad \frac{1}{2} \sum_{i,j=1}^{n} a_i a_j y_i y_j K_{ij} - \sum_{i=1}^{n} a_i$$

接著將一次要更新的兩個值從求解問題的式子中抽取出來出來, 抽取 $\alpha_1$ 及 $\alpha_2$ 來做代表, 但因為拆解的工程浩大, 因此先將前半部的 $\sum$ 其中的 i 拆出來

$$= \frac{1}{2} \sum_{i=1}^{2} \left(\sum_{j=1}^{n} a_i a_j y_i y_j K_{ij}\right) +$$
$$\frac{1}{2} \sum_{i=3}^{n} \left(\sum_{j=1}^{n} a_i a_j y_i y_j K_{ij}\right) - \sum_{i=1}^{n} a_i$$

接著將括弧內的 Sigma 也一樣拆開

$$\frac{1}{2} \sum_{i=1}^{2} \left(\sum_{j=1}^{2} a_i a_j y_i y_j K_{ij} + \sum_{j=3}^{n} a_i a_j y_i y_j K_{ij}\right) +$$
$$\frac{1}{2} \sum_{i=3}^{n} \left(\sum_{j=1}^{2} a_i a_j y_i y_j K_{ij} + \sum_{j=3}^{n} a_i a_j y_i y_j K_{ij}\right) -$$
$$\sum_{i=1}^{n} a_i$$

利用連鎖率將括弧外面的 Sigma 乘進去

$$\frac{1}{2} \sum_{i=1}^{2} \sum_{j=1}^{2} a_i a_j y_i y_j K_{ij} +$$
$$\frac{1}{2} \sum_{i=1}^{2} \sum_{j=3}^{n} a_i a_j y_i y_j K_{ij} +$$
$$\frac{1}{2} \sum_{i=3}^{n} \sum_{j=1}^{2} a_i a_j y_i y_j K_{ij} +$$
$$\frac{1}{2} \sum_{i=3}^{n} \sum_{j=3}^{n} a_i a_j y_i y_j K_{ij} - \sum_{i=1}^{n} a_i$$

可以看到中間的兩項是一樣的, 因為不論是 i 還是 j, 乘開後其實是一樣的東西, 因此將它合併成一項來表示即可

$$\frac{1}{2} \sum_{i=1}^{2} \sum_{j=1}^{2} a_i a_j y_i y_j K_{ij} + \sum_{i=1}^{2} \sum_{j=3}^{n} a_i a_j y_i y_j K_{ij} +$$
$$\frac{1}{2} \sum_{i=3}^{n} \sum_{j=3}^{n} a_i a_j y_i y_j K_{ij} - \sum_{i=1}^{n} a_i$$

再來將前半部 $\sum 1\ 2$ 的展開來並且將後面的也抽取 $\alpha_1$ 及 $\alpha_2$

$$\frac{1}{2} a_1^2 y_1^2 K_{11} + \frac{1}{2} a_2^2 y_2^2 K_{22} + a_1 a_2 y_1 y_2 K_{12} +$$
$$a_1^2 y_1^2 \sum_{j=3}^{n} a_j y_j K_{1j} + a_2^2 y_2^2 \sum_{j=3}^{n} a_j y_j K_{2j} +$$
$$\frac{1}{2} \sum_{i=3}^{n} \sum_{j=3}^{n} a_i a_j y_i y_j K_{ij} - a_1 - a_2 - \sum_{i=3}^{n} a_i$$

到這裡先整理一下，這裡 y 平方必定為 1，所以可以忽略，並且將沒有包含 $\alpha_1$ 或 $\alpha_2$ 的歸類成 $\Upsilon$ 符號做代表

$$\frac{1}{2}a_1^2K_{11}+\frac{1}{2}a_2^2K_{22}+a_1a_2y_1y_2K_{12}+a_1y_1\sum_{j=3}^{n}a_jy_jK_{1j}+a_2y_2\sum_{j=3}^{n}a_jy_jK_{2j}-a_1-a_2+\Upsilon$$

然後為了還是能夠一次只更新一個值，所以將 $\alpha_1$ 替換成 $\alpha_2$，且透過前面推導可以知道 $\alpha_2$ 更新後 $\alpha_1$ 也會跟著一起被更新，也就不用擔心前面所說到會被固定住的問題，因此透過下面的式子來代替

$$a_1^{old}y_1+a_2^{old}y_2 = a_1^{new}y_1+a_2^{new}y_2 = -\sum_{i=3}^{n}a_iy_i$$

將式子的 $\alpha_2$ 移項後兩邊同乘 $y_1$

$$a_1^{new} = -y_1\sum_{i=3}^{n}a_iy_i-a_2^{new}y_1y_2$$

接著將多餘的 $\sum$ 用 I 取代，因此替換 $\alpha_1$ 的式子變成如下

$$a_1^{new} = I-a_2^{new}y_1y_2$$

將 $\alpha_1$ 用替換的式子替換後

$$\frac{1}{2}\left(I-a_2^{new}y_2y_1\right)^2K_{11}+\frac{1}{2}a_2^{new2}K_{22}+\left(I-a_2^{new}y_2y_1\right)a_2^{new}y_1y_2K_{12}+\left(I-a_2^{new}y_2y_1\right)y_1\sum_{j=3}^{n}a_jy_jK_{1j}+a_2^{new}y_2\sum_{j=3}^{n}a_jy_jK_{2j}-\left(I-a_2^{new}y_2y_1\right)-a_2^{new}+\Upsilon$$

再來因為要求的是 $a_2$，因此透過 Lagrange 的方式對 $a_2$ 做偏導數並等於 0，並且將 y 平方的部分為 1 給忽略掉來簡化式子

$$-y_2y_1IK_{11}+a_2^{new}K_{11}+a_2^{new}K_{22}+y_1y_2IK_{12}-2a_2^{new}K_{12}-y_2\sum_{j=3}^{n}a_jy_jK_{1j}+y_2\sum_{j=3}^{n}a_jy_jK_{2j}+y_2y_1-1=0$$

再來將有包含 $a_2$ 的項放等號左邊其餘放右邊

$$a_2^{new}K_{11}+a_2^{new}K_{22}-2a_2^{new}K_{12}=y_2\sum_{j=3}^{n}a_jy_jK_{1j}-y_2\sum_{j=3}^{n}a_jy_jK_{2j}-y_2y_1-y_1y_2IK_{12}+y_1y_2IK_{11}+1$$

右式的 $\sum$ 這邊用 $f(x_i)=\sum_{j=1}^{n}a_jy_j{X_j}^TX_i-b=\sum_{j=1}^{n}a_jy_jK_{ji}-b$ 來定義它後簡化，並且將多餘的 $\alpha_1^{old}$ 及 $\alpha_2^{old}$ 提出

$$a_2^{new}K_{11}+a_2^{new}K_{22}-2a_2^{new}K_{12}=y_2\left[f(x_1)-a_1^{old}y_1K_{11}-a_2^{old}y_2K_{12}\right]-y_2\left[f(x_2)-a_1^{old}y_1K_{21}-a_2^{old}y_2K_{22}\right]-y_2y_1-y_1y_2IK_{12}+y_1y_2IK_{11}+1$$

接著將右式的 I 用 $a_1^{old}+a_2^{old}y_2y_1 = -y_1\sum_{i=3}^{n}a_iy_i=I$ 更換後

$$a_2^{new}\left(K_{11}+K_{22}-2K_{12}\right)=y_2\left[f(x_1)-a_1^{old}y_1K_{11}-a_2^{old}y_2K_{12}\right]-y_2\left[f(x_2)-a_1^{old}y_1K_{21}-a_2^{old}y_2K_{22}\right]-y_2y_1-y_1y_2\left[a_1^{old}+a_2^{old}y_2y_1\right]K_{12}+y_1y_2\left[a_1^{old}+a_2^{old}y_2y_1\right]K_{11}+1$$

將它全部展開後，並將 y 平方的部分為 1 給忽略掉

$$a_2^{new}\left(K_{11}+K_{22}-2K_{12}\right)=y_2f(x_1)-a_1^{old}y_1y_2K_{11}-a_2^{old}K_{12}-y_2f(x_2)+a_1^{old}y_1y_2K_{21}+a_2^{old}K_{22}-y_2y_1-a_1^{old}y_1y_2K_{12}-a_2^{old}K_{12}+a_1^{old}y_1y_2K_{11}+a_2^{old}K_{11}+1$$

接著可以發現有相同的項目可以消去，消去相同項目後就剩下

$$a_2^{new}\left(K_{11}+K_{22}-2K_{12}\right)=y_2f(x_1)-a_2^{old}K_{12}-y_2f(x_2)+a_2^{old}K_{22}-y_2y_1-a_2^{old}K_{12}+a_2^{old}K_{11}+1$$

有 $a_2$ 的歸類並提出來且將 1 展開成 $y_2$ 平方，之後把右邊部分把 $y_2$ 也給提出來

$$a_2^{new}\left(K_{11}+K_{22}-2K_{12}\right)=a_2^{old}\left(K_{11}+K_{22}-2K_{12}\right)+y_2\left(f(x_1)-y_1-f(x_2)+y_2\right)$$

右式括弧裡面可以透過誤差值來替換掉，因為誤差值 $E_i=W^TX_i-b-y_i$，而由前面 Lagrange 偏導數部分將 W 替換後 $E_i=\sum_{j=1}^{n}a_jy_j{X_j}^TX_i-b-y_i$，接著可以發現前半段 $\sum$ 的部分就是先前定義的 $f(x_i)$，所以知道 $E_i=f(x_i)-y_i$，因此式子變成

$$a_2^{new}\left(K_{11}+K_{22}-2K_{12}\right)=a_2^{old}\left(K_{11}+K_{22}-2K_{12}\right)+y_2\left(E_1-E_2\right)$$

然後將右邊除了 $a_2^{new}$ 的部分給消去，因此最後更新 $a_2^{new}$ 的式子變成如下

$$a_2^{new}=a_2^{old}+\frac{y_2(E_1-E_2)}{K_{11}+K_{22}-2K_{12}}$$

到這邊已經可以將 $a_2^{new}$ 更新，但如果更新後直接去更新 $a_1^{new}$ 會問題，因為更新後有可能造成新的 a 會跳脫出原本的限制範圍內，因此在這邊要做一個判斷來確認 $a_2^{new}$ 更新過後不會導致 $a_1^{new}$ 在約束範圍外。

為了要確保更新之後的 $a_2^{new}$ 不會導致 $a_1^{new}$ 超出限制範圍，這邊用前面使用到的 $a_1^{new}y_1+a_2^{new}y_2=-\sum_{i=3}^{n}a_iy_i=\varpi$，這邊就有兩種狀況

1. 選定的兩個 $\alpha$ 點所對應的期望值 y 為同號
   因為要求 $0 \leq \alpha \leq C$，所以變動後的 $a_(1)^{new}$ 也要在範圍內，因此將 $a_(1)^{new}$ 帶入 $0 \leq a_(1)^{new} \leq C$，又 $a_(1)^{new} = \varpi - a_2^{new}$，可以把中間替換成 $0 \leq \varpi - a_2^{new} \leq C$，然後兩邊同時減去 $\varpi$ 並乘以 -1，$\varpi \geq a_2^{new} \geq \varpi - C$, 將 $\varpi$ 還原成 $a_1^{new} + a_2^{new}$，所以 $a_2^{new}$ 就變成

$$a_1^{new} + a_2^{new} \geq a_2^{new} \geq a_1^{new} + a_2^{new} - C$$

   又因 $a_2^{new}$ 也要符合 $0 \leq a_2^{new} \leq C$ 所以他左邊的最小值要找出兩者中最大的，右邊的最大值要找出兩這中最小的，才會讓 $a_1^{new}$ 落在限制範圍內

$$Max(0, a_1^{new} + a_2^{new} - C) \leq a_2^{new} \leq Min(C, a_1^{new} + a_2^{new})$$

2. 選定的兩個 $\alpha$ 點所對應的期望值 y 為異號
   因為要求 $0 \leq \alpha \leq C$，所以變動後的 $a_(1)^{new}$ 也要在範圍內，因此將 $a_(1)^{new}$ 帶入 $0 \leq a_(1)^{new} \leq C$，又 $a_(1)^{new} = \varpi + a_2^{new}$，可以把中間替換成 $0 \leq \varpi + a_2^{new} \leq C$，然後兩邊同時減去 $\varpi$，$-\varpi \leq a_2^{new} \leq C - \varpi$, 將 $\varpi$ 還原成 $a_1^{new} - a_2^{new}$，所以 $a_2^{new}$ 就變成

$$-a_1^{new} + a_2^{new} \geq a_2^{new} \geq C - a_1^{new} + a_2^{new}$$

   又因 $a_2^{new}$ 也要符合 $0 \leq a_2^{new} \leq C$ 所以他左邊的最小值要找出兩者中最大的，右邊的最大值要找出兩這中最小的，才會讓 $a_1^{new}$ 落在限制範圍內

$$Max(0, -a_1^{new} + a_2^{new}) \leq a_2^{new} \leq Min(C, C - a_1^{new} + a_2^{new})$$

由上述所知 $a_2^{new}$ 的要符合在某個範圍之間，如果超出就將更新值設定成極限值。

已經更新完其中一個 $a_2$ 也確定新的 $a_2^{new}$ 不會導致 $a_1^{new}$ 超出限制範圍，$a_1$ 的更新也相對 $a_2$ 簡單多，只要利用 $a_1^{old}y_1 + a_2^{old}y_2 = a_1^{new}y_1 + a_2^{new}y_2 = -\sum_{i=3}^{n} a_i y_i$ 中的 $a_1^{old}y_1 + a_2^{old}y_2 = a_1^{new}y_1 + a_2^{new}y_2$，將 $a_2^{new}y_2$ 移項後並同乘 y1，就可以知道新的 a1 的更新公式

$$a_1^{new} = a_1^{old} + y_1\left(a_2^{old} - a_2^{new}\right)$$

上面已經有新的 $a_1^{new}$ 及 $a_2^{new}$，因此可以這兩個 $\alpha$ 知道新的 $W$ 及 $b$，由前面 Lagrange 的限制公式也知道的 $W$ 的更新可以使用 $W = \sum_{i=1}^{n} a_i y_i x_i$，但如果因為更新兩個 $\alpha$ 而重跑一次迴圈會造成效能損耗太大，因此將新舊的 $W$ 列出且將更新的兩個值特別拉出

$$W^{new} = a_1^{new}y_1x_1 + a_2^{new}y_2x_2 + \sum_{i=3}^{n} a_i y_i x_i$$
$$W^{old} = a_1^{old}y_1x_1 + a_2^{old}y_2x_2 + \sum_{i=3}^{n} a_i y_i x_i$$

接著坐上下相減可以得到 $W^{new}$ 的更新公式

$$W^{new} = W^{old} + \left(a_1^{new} - a_1^{old}\right) y_1 x_1 + \left(a_2^{new} - a_2^{old}\right) y_2 x_2$$

再來是更新新的偏權值 b，可以使用原本的公式 $W^{newT}X_i - b^{new} = y_i$，再將 $b^{new}$ 及 $y_i$ 移項後

$$W^{newT}X_i - y_i = b^{new}$$

但由於一次使用兩個點更新，因此這邊更新出來的 $b^{new}$ 有可能出現兩種 $b_1^{new}$、$b_2^{new}$，因此要判斷要使用哪一個來作為 $b^{new}$，這邊採用的是先前提到的 KKT 互補鬆弛所推導出來 $\alpha$ 範圍來判定，由於前面推導可以得知 $0 < \alpha_i^{new} < C$ 的話該點就是我們的 Support Vector

1. 當任一點 $0 < \alpha_i^{new} < C$，$b^{new} = b_i^{new}$
2. 當沒有一點符合可用原來的 $b^{new} = b^{old}$ 或使用 $b^{new} = \frac{b_1^{new} + b_2^{new}}{2}$

### 5.3. Coding Process

## 6. Multi-Category SVM

在 Machine Learning 、類神經網路的世界中，很常用到的分類方式都是多分類，因為這樣的分法才能找出多種的類別，並且想想實際上應用的話就會發現，通常我們要做的分類並不會只有兩種，而是希望能夠有多種的分類，如果只有能夠分類兩種的話，不就很難去做應用了嗎? 例如一個數字辨識機就是需要分類出 0 9 的數字樣式，所以如果只能夠二分法的話在實際應用上面會不太夠活用，然而我們的 SVM 雖然可以透過核函數及鬆弛變數部分將我們的線做非線性的分法，但是他始終還是二分法。

### 6.1. Introduction

一般的 SVM 屬於二分法，因此需要靠一些方法結合來達到多分類的效果，由於方法有很多種，因此這邊

只介紹部分的方法來了解怎麼用二分法的 SVM 來做出多分類。其實 SVM 多分類的概念很簡單，就是將多個 SVM 結合幾來達到所要的多分類，而且這些方法都各有好處。

### 6.2. Kind

1. 一對多分類法 (one-versus-rest)
   在訓練多個 SVM 時，在訓練每個 SVM 的時候將目標群歸為一類 (預期結果為 1)，其餘的算另一類 (預期結果為 0) 來當作樣本去做訓練。當真正的資料要進行分類的時候，會將數據丟入每個分類器來做運算，再從結果中選出最大的就是該資料的歸屬。(圖8)

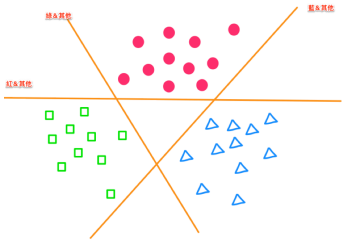


图 8. 一對多分類法

2. 一對一分類法 (one-versus-one)
   在訓練多個 SVM 時，將每個種類與其他的種類各別做一對一的分類，這樣就可以得到 n(n+1)/2 分類器。在真正的分類資料進行時，我們會有一個投票表，將數據丟進每個分類器去做分類，哪個分類器出來的類別，該票數就 +1，分類完畢後最高得票的類別就是給資料的歸數。(圖9)

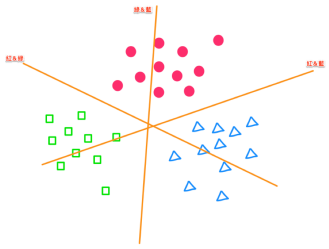


图 9. 一對多分類法

3. 層次分類法 (H-SVMs)
   在訓練多個 SVM 時，將種類分成一群一群的分類，透過每次的二分法將群數一堆一堆慢慢拆開來，每次分類時幾乎都會去掉一半的種類，透過這樣的方式快速去找出資料對應的種類是屬於哪一種。(圖10)

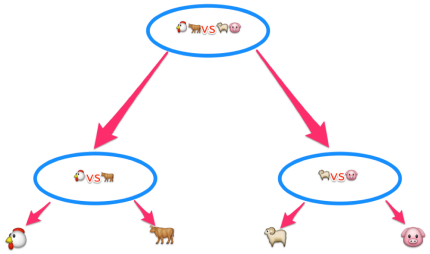


图 10. 一對多分類法

4. 有向循環圖分類法 (DAG-SVMs)
   在訓練多個 SVM 時，將每個種類與其他的種類各別做一對一的分類，這部分有點像是一對一分類法，但是不一樣的地方是他會有點像是瀑布的方式將資料倒下來讓它流至應該屬於的分類，所以我們會有一個由一個一個 SVM 分類器所組成金字塔型，而這個流向有方向性所以也稱為有向循環圖分類。(圖11)

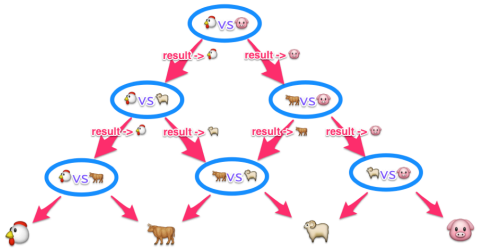


图 11. 一對多分類法

上面介紹了常用的幾種分類方式，可以知道要怎麼去將原本的 SVM 分類變得能夠多分類的部分，上面的順序是由最常見的方式排下來，對於相當有名的 libsvm 是使用上面的第二種”一對一分類法” 來寫裡面多分類的部分，所以看過上面的解釋後，也可以寫出一個 libsvm 的分類方式了。