Introduction to LDA

在開始講 LDA (Linear Discriminant Analysis) LDA 在 pattern recognition 上的應用。Pattern recognition 在 learning phase 時，會有一些 training samples。那些 training samples 其實就是告訴我們一種對應關係，譬如說判斷一個人是不是超過 180cm，我們收到的 training samples 可能就是(188cm, yes), (176cm, no)…….。這時我們就用那些 samples 訓練(training)出一組參數，那組參數可以決定出一個函式，使的那個函式可以符合 training samples 的對應關係(ex. F(188) = yes)。以上例子的 feature 只有一個維度，那就是身高。假設我們要建立的關係是一張 100x100 大小的人臉照片，對應到那個人是誰的時候，我們的 feature space 維度很不巧就是 10000 個維度，在那麼大的維度下，不管是 training 還是做 application 都是很沒有效率的。想像一下如果 10000 中他們的對應關係其實是一條 curve，那其實我們可以把 10000 維降到一維也可以建立出他們的對應關係，因此在那麼大維度的空間去 training 是很沒有效率的事，這時我們就需要把資料降維。最常用的降維方式有兩種，分別是 PCA 和 LDA。

以下簡介一下傳統 PCA 跟 LDA 各別的特性 ：

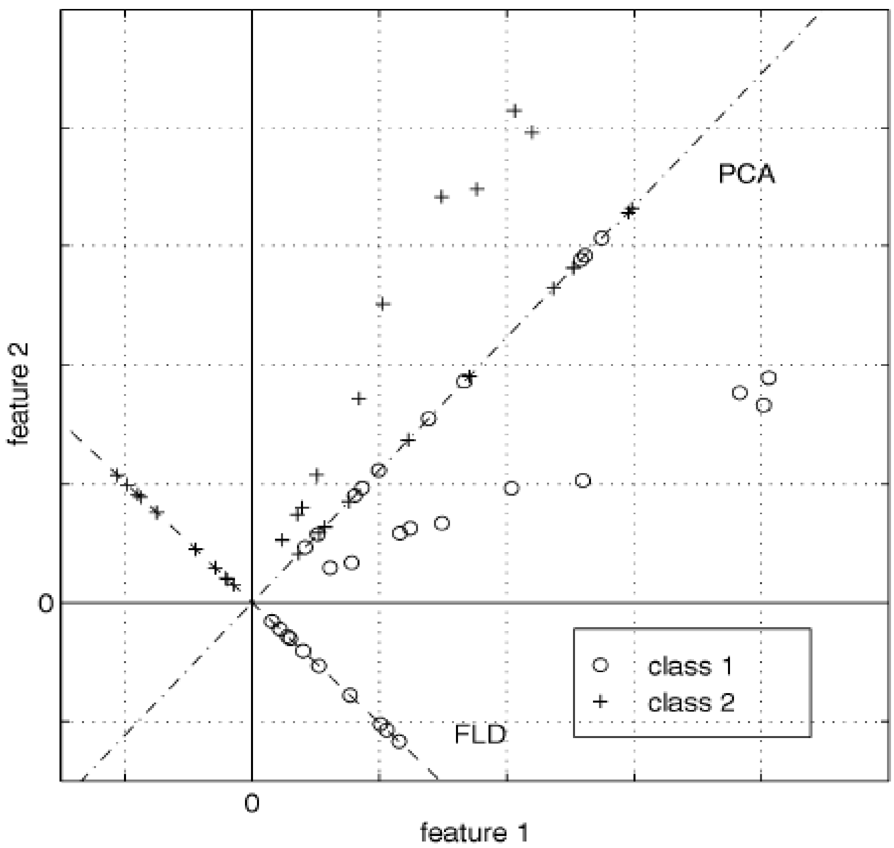
PCA

- 擁有 Euclidean space vector norm 最小的特性。簡單來說，就是當我們用 PCA 把資料降維後，用傳統的 vector norm 下來看，誤差會是最小的。

LDA

- 擁有降維後能將各群 data 分開的特性，但是以 vector norm 來看，誤差不保證是最小的。

所以確切來說，如果要做資料壓縮，那麼 PCA 一定是最佳的選擇。如果是要做 pattern classification，LDA 顯然就是比較好的選擇。以下的圖示分別表示同一組的 data，用 PCA 來降維跟用 LDA 來降維的結果。



上圖 o 跟＋是兩個不同的 class，用 PCA 求出降維後的 basis，會使所有 data 投影到那 basis 產生的 error (Euclidean distance)最小。另一方面，Fisher Linear Discriminant (LDA) 所產生的 basis 就不一樣了，你可以看得出來 o 跟＋被投影到 LDA basis 上時，有明顯被區分成兩群的情況，我們可以在投影過後的 space 中，決定出一個點把兩群資料分開！而 PCA 投影過後的 data 沒有辦法決定一個點把兩群資料分開。

LDA 的推導

以下將一步一步地推導 LDA。

Case 1：將兩類資料投影到一維空間

我們的目的是要找到一個 vector **w**, 把資料投影到 **w** 上面去，得到新的 coordinate y。

$$y = w^t x$$

以 LDA 的精神來看，是希望能將同一類的資料投影得越近越好，不同類的資料 map 得越遠越好。爲了描述這個概念，我們需要用一些量化的值去表示，首先是每個 class data 的平均植(mean)。

$$m_i = \frac{1}{n_i}\sum_{x\in Di} x$$

而投影過後的平均值是：

$$\tilde{m}_i = \frac{1}{n_i}\sum_{y\in Yi} y = \frac{1}{n_i}\sum_{x\in Di} w^t x = \frac{1}{n_i} w^t \sum_{x\in Di} x = w^t m_i$$

ni 是第 i 類的資料個數。Di 是第 i 類資料的集合。Yi 是投影後第 i 類資料的集合。所以以上可以看出投影過後的每個 class data 的平均值是原來在高維度空間的平均值投影。

再來我們可以定出投影後兩類資料的平均距離了

$$\left|\tilde{m}_1 - \tilde{m}_2\right| = \left|w^t(m_1 - m_2)\right|$$

我們也可以定出兩類資料在投影過後，分散的度量(scatter)

$$\tilde{s}_i^{\,2} = \sum_{y\in Yi}(y - \tilde{m}_i)^2$$

再來就依照 LDA 的精神，投影過後的兩類資料越分開越好，就代表他們投影過後的平均值差越多越好。投影過後的同類資料越集中越好，就是投影過後的分散程度越小越好。我們可以因此得到以下的函式：

$$J(w) = \frac{\left|\tilde{m}_1 - \tilde{m}_2\right|^2}{\tilde{s}_1^{\,2} + \tilde{s}_2^{\,2}}$$

當決定一個投影 basis w，我們可以求出一個 J(w)的值，我們希望分母越小越好，分子越大越好，我們很直覺地可以聯想到，求極值的方式不就是 lagrange method 嗎？沒錯，當找出一個 w 可使 J(w)出現最大值，那個 w 就是我們要的 basis。但是以上的 J(w)右邊的形式是間接跟 w 有關係，我們再做一下推導使得右邊的式子能跑出 w 這個 term 出來。

我們定義 scatter matrics Si，用來描述投影前的各類資料分散情況。

$$S_i = \sum_{x \in Di} (x - m_i)(x - m_i)^t$$

原來的分母每個 term 可以寫成 Si 跟 w 個組合：

$$s_i^{\,2} = \sum_{x \in Di} (w^t x - w^t m_i)^2 = \sum_{x \in Di} w^t (x - m_i)(x - m_i)^t w$$
$$= w^t S_i w$$

最後分母可以寫成下面的式子，Sw 是 S1+S2。

$$\tilde{s}_1^{\,2} + \tilde{s}_2^{\,2} = w^t (S_1 + S_2) w = w^t S_W w$$

分子則可以寫成下列形式：

$$(m_1 - m_2)^2 = (w^t m_1 - w^t m_2)^2 = w^t (m_1 - m_2)(m_1 - m_2)^t w$$
$$= w^t S_B w$$

原來的 J(w)將由上列的改寫可變成：

$$J(w) = \frac{w^t S_B w}{w^t S_W w}$$

要求一個 w 使 J(w)最大，我們可以用最熟悉的 Lagrange multiplier。以上的式子可以看出，w 有無限多解，因為當 w 乘上一個倍數，J(w)值都會是一樣的(分母分子相消)。因此我們限定 w 的長度，使得分母乘出來為 1。而那就當成是 Lagrange method 的條件，而目標就是讓分子最大。

$$c(w) = w^t S_B w - \lambda (w^t S_W w - 1)$$
$$\Rightarrow \frac{dc}{dw} = 2 S_B w - 2 \lambda S_W w = 0$$
$$\Rightarrow S_B w = \lambda S_W w$$

所以讓 J(w)最大的 w，就會符合下列的式子

$$S_B w = \lambda S_W w$$

這是一個 generalized eigenvalue problem。當 Sw 有 inverse，就可以讓上式成爲普通的 eigenvalue problem。

$$S_W^{-1} S_B w = \lambda w$$

但是 SBw 的方向是(m1-m2)，所以其實我們要的 w 就是以下的解，不需要解 eigenvalue problem。

$$w = S_W^{-1}(m_1 - m_2)$$

Case 2： 多種類資料投影到高維空間

現在我們做一些改變來符合多種類資料跟高維空間的需求。首先我們把 Sw 改成多種類資料的版本(class number > 2)，如果對應到 case 1，c = 2。

$$S_w = \sum_{i=1}^{c} S_i$$

再來把 SB 改成以下式子

$$S_B = \sum_{i=1}^{c} n_i (m_i - m)(m_i - m)^t, m = \frac{1}{n} \sum_x x$$

請注意 SB 跟 case 1 的不同。大體來講這還是描述了各類 data 之間的分散程度。當要投影到高維空間後，我們不再是求一個 vector basis w，而是要求一組 basis，所以多組 w 將會寫成一個 matrix W 來表示，裡面的 column vector 就是一個 basis。因此原來的分母跟分子將變成：

$$\tilde{S}_B = W^t S_B W, \tilde{S}_W = W^t S_W W$$

所以原來的 J(W)將變成

$$J(W)=\frac{\left|W^{t}S_{B}W\right|}{\left|W^{t}S_{w}W\right|}$$

注意一下 J(W)中的 W 是一個 matrix，代表一組 basis。分子分母因爲 W 是 matrix，所以必需加個 determinant 才能變成常數。那加了 determinant 真的符合 LDA 的精神嗎？答案是是的。因爲 determinant 出來是 eigen value 相乘，也就是 hyperellipsoidal scattering volume，可以想像是整體資料分散的體積。所以取 J(W)的最大値的確是符合 LDA 的精神。那要怎麼求出 W 呢？其實求 W 第 i 個 column vector，只要解以下的式子，取第 i 大的 eigenvalue 對應的 eigenvector 就是了。因此如果要投影到 k 維的 space，取前 k 大的 eigenvalues 對應的 eigenvectors 就可以了。

$$S_{B}w_{i}=\lambda S_{W}w_{i}$$

這是最基礎的 LDA 簡介，LDA 還會有各種變形，譬如說如果投影後的 data 仍然不能用 hyper plane 切開怎麼辦？這時就有 Kernel LDA 的出現。LDA 的計算仍然跟原始資料的維度有關，更有效率的 2DLDA 可以大大減少求得 LDA basis 的計算成本。那都是更進階的技巧。但整體而言，LDA 最重要的精神就是把高維的資料投影到低維空間中，並且讓他們投影過後能夠具備好分辨(分割)的特性。