Principal Component Analysis with Missing Data

葉冠麟

June 16, 2006

1 PCA review

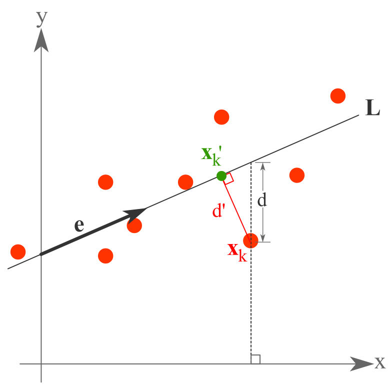

圖 1: 我們試著以一條直線 $\boldsymbol{L}$ 來表示空間中的點, 以線性迴歸的方式, 我們會累加 d 取最小值, 然而我們應該累加的是 d', 而這正是 PCA 的處理方式。

「我們要如何用一個點來代表二維空間中散佈的點?」這是一個簡單的數學問題, 許多人可以憑著直覺正確地回答:「取平均值 (mean)。」我們再進一步地問, 如果是一條線呢?「如何用一條線來代表二維空間中散佈的點?」用國中數學來思考這個問題, 我們會很直覺地聯想到線性規劃。線性規劃的精神在於「找一條直線 $\boldsymbol{L}$, 使得每一個點到 $\boldsymbol{L}$ 垂直於 x **軸** (**或** y **軸**)的距離 d, 累加起來最小。」我們國中的時候很滿足這樣的答案, 但顯然這樣的解是有問題的, 爲什麼計算的不是垂直於**直線** $\boldsymbol{L}$ 的距離 d'? 當開始有這樣的疑慮產生, 就是引入 PCA 最佳的時機。因爲 PCA 正是以 Least Squared Error 的角度來求解。

1.1 Principal Component Analysis (PCA)
(這部份的數學推導全引自 Pattern Classification, Wiley)

PCA的使用並不被限制在二維的空間, 我們可以把問題提升到 m 維空間。我們依然在空間中灑點, 這些點代表的是我們的資料, 分別是 $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, ..., $\boldsymbol{x}_n$, 皆爲 m 維的向量 (vector)。

我們先問個簡單的問題,「我們如何找到一個 m 維的向量, 足以代表灑在空間中的 n 個點?」「代表」這個字詞其實用得不很準確, 比較嚴謹的問法應該是, 我們如何找到 m 維的向量 $\boldsymbol{x}_0$, 使得

$$J_0(\boldsymbol{x}_0) = \sum_{k=1}^{n} \|\boldsymbol{x}_0 - \boldsymbol{x}_k\|^2, \tag{1}$$

最小。其中 $J_0(\boldsymbol{x}_0)$ 是我們的 squared-error criterion function 。定義 $\boldsymbol{m}$ 爲 n 筆資料的平均值, 即

$$\boldsymbol{m} = \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{x}_k, \tag{2}$$

則

$$\begin{aligned}
J_0(\boldsymbol{x}_0) &= \sum_{k=1}^{n} \|(\boldsymbol{x}_0 - \boldsymbol{m}) - (\boldsymbol{x}_k - \boldsymbol{m})\|^2 \\
&= \sum_{k=1}^{n} \|\boldsymbol{x}_0 - \boldsymbol{m}\|^2 - 2\sum_{k=1}^{n}(\boldsymbol{x}_0 - \boldsymbol{m})^t(\boldsymbol{x}_k - \boldsymbol{m}) + \sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2 \\
&= \sum_{k=1}^{n} \|\boldsymbol{x}_0 - \boldsymbol{m}\|^2 - 2(\boldsymbol{x}_0 - \boldsymbol{m})^t\sum_{k=1}^{n}(\boldsymbol{x}_k - \boldsymbol{m}) + \sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2 \\
&= \sum_{k=1}^{n} \|\boldsymbol{x}_0 - \boldsymbol{m}\|^2 + \underbrace{\sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2}_{independent of \boldsymbol{x}_0}.
\end{aligned} \tag{3}$$

其中 $\sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2$ 與 $\boldsymbol{x}_0$ 無關, 所以略去不看, 此時若要讓 $J_0(\boldsymbol{x}_0)$ 最小, 顯然 $\boldsymbol{x}_0 = \boldsymbol{m}$。這結果很符合直覺, 我們的確可以用平均值來代替整組資料。

我們現在把問題變得有趣些, 我們試著從 $\boldsymbol{x}_0$ 拉出一條直線 $\boldsymbol{L}$, 目標依然是使 squared-error 最小。如圖 1所示, 每一點 $\boldsymbol{x}_k$ 都能在直線 $\boldsymbol{L}$ 上找到相對應的 $\boldsymbol{x}'_k$, 若是直線 $\boldsymbol{L}$ 沿著單位向量 $\boldsymbol{e}$ 走, 則

$$\boldsymbol{x}'_k = \boldsymbol{m} + a_k\boldsymbol{e}, \tag{4}$$

其中 a_k 爲縮放項。我們重新定義 squared-error criterion function ,

$$J_1(a_1,\ldots,a_n,\boldsymbol{e}) = \sum_{k=1}^{n} \|(\boldsymbol{x}'_k - \boldsymbol{x}_k)\|^2 = \sum_{k=1}^{n} \|((\boldsymbol{m} + a_k\boldsymbol{e}) - \boldsymbol{x}_k)\|^2, \tag{5}$$

在 J_1 中未知的變數有a_1, a_2, $\ldots$, a_k, 以及 $\boldsymbol{e}$, 我們先假設$\boldsymbol{e}$已知, 然後看看 a_k 必須等於多少才能使得 J_1 最小。

$$\begin{aligned} J_1(a_1,\ldots,a_n,\boldsymbol{e}) &= \sum_{k=1}^{n} \|((\boldsymbol{m} + a_k\boldsymbol{e}) - \boldsymbol{x}_k)\|^2 = \sum_{k=1}^{n} \|(a_k\boldsymbol{e} - (\boldsymbol{x}_k - \boldsymbol{m}))\|^2 \\ &= \sum_{k=1}^{n} a_k^2\|\boldsymbol{e}\|^2 - 2\sum_{k=1}^{n} a_k\boldsymbol{e}^t(\boldsymbol{x}_k - \boldsymbol{m}) + \sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2. \end{aligned} \tag{6}$$

因爲 $\sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2$ 與 a_k 無關, 且 $\boldsymbol{e}$爲單位向量, 即 $\|\boldsymbol{e}\|^2 = 1$, 我們可以另外定義

$$J'_1(a_1,\ldots,a_n,\boldsymbol{e}) = \sum_{k=1}^{n} a_k^2 - 2\sum_{k=1}^{n} a_k\boldsymbol{e}^t(\boldsymbol{x}_k - \boldsymbol{m}), \tag{7}$$

將 J'_1 分別對 a_k 做偏微分, 我們可以得到

$$a_k = \boldsymbol{e}^t(\boldsymbol{x}_k - \boldsymbol{m}). \tag{8}$$

這是一個很有趣的結果, 因爲這代表如果你已經知道 $\boldsymbol{e}$, 將空間中任一點$\boldsymbol{x}_k$投射到直線 $\boldsymbol{L}$ 上, 只需要將原座標爲移後與 $\boldsymbol{e}^t$ 做內積, 就可以得到空間轉換後的新座標 $\boldsymbol{x}'_k = a_k$ 。也就是說, 不論是原來存在的點, 或是之後才加入的, 都可以做相同的空間轉換。不過, 到底 $\boldsymbol{e}$ 是什麼? 既然我們已經知道 a_k 該等於多少, 我們先將 J_1 改寫成較爲精簡的模式

$$\begin{aligned} J_1(\boldsymbol{e}) &= \sum_{k=1}^{n} a_k^2\|\boldsymbol{e}\|^2 - 2\sum_{k=1}^{n} a_k^2 + \sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2 \\ &= -\sum_{k=1}^{n} [\boldsymbol{e}^t(\boldsymbol{x}_k - m)]^2 + \sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2 \\ &= -\sum_{k=1}^{n} \boldsymbol{e}^t(\boldsymbol{x}_k - \boldsymbol{m})(\boldsymbol{x}_k - \boldsymbol{m})^t\boldsymbol{e} + \sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2 \\ &= -\boldsymbol{e}^t\boldsymbol{S}\boldsymbol{e} + \sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2. \end{aligned} \tag{9}$$

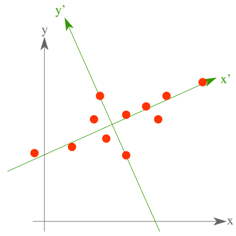

圖 2: 深灰色的座標表示原座標軸, 綠色的座標是轉換後的結果。

其中, $\boldsymbol{S} = \sum_{k=1}^{n} \boldsymbol{e}^t(\boldsymbol{x}_k - \boldsymbol{m})(\boldsymbol{x}_k - \boldsymbol{m})^t\boldsymbol{e}$, 我們將之稱爲 scatter matrix , 他的樣子幾乎同等於 covariance matrix , 只是少除了分母 $n-1$ 。推導到這一步, 我們似乎只要做個微分就能求出 $\boldsymbol{e}$, 但有一點要特別注意, $\boldsymbol{e}$ 是單位向量, $\|\boldsymbol{e}\|^2 = 1$ 的限制必須在求解時一併考慮進去。所以我們引入 Largrange Multipliers 的方法來求 $\boldsymbol{e}$, 使得 $\boldsymbol{e}^t\boldsymbol{S}\boldsymbol{e}$ 最大 (爲要使 J_1 最小)。令

$$u = \boldsymbol{e}^t\boldsymbol{S}\boldsymbol{e} - \lambda(\boldsymbol{e}^t\boldsymbol{e} - 1) \tag{10}$$

並將 u 對 $\boldsymbol{e}$ 做偏微分,

$$\frac{\partial u}{\partial \boldsymbol{e}} = 2\boldsymbol{S}\boldsymbol{e} - 2\lambda\boldsymbol{e}, \tag{11}$$

令其值爲 0 , 則得

$$\boldsymbol{S}\boldsymbol{e} = \lambda\boldsymbol{e}. \tag{12}$$

這是典型的 eigen decomposition 的問題, 已超出本文討論的範圍, 我們只要知道, 原來我們所要找的 $\boldsymbol{e}$ 其實就是空間中所有點形成的 covariance matrix 經過 eigen decomposition 所求得的 eigen vector 。

「要如何表示空間中的點?」我們從點討論到線, 問題依然可以提升到更高的維度, 但原則就只有一個 — 對 covariance matrix 做 eigen decomposition 。理論上 m 維的空間我們就可以找到 m 個 eigen vector , 我們若是對各個向量作投影, 就可以把點轉換到另一個比較符合分布情形的座標空間, 如圖 2所示。

1.2 降維

我們目前對 PCA 的認識是建立在 Least-Squared Error 上, 也就是說, 若是我們把點投射至 PCA 找到的向量上, 以這些向量爲新的座標軸, 則我們可以發現投影點

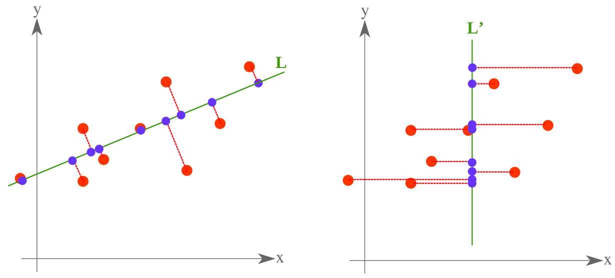

圖 3: 相較於 $\boldsymbol{L'}$, $\boldsymbol{L}$ 會是我們比較希望看到的投影方式, 因爲經過座標轉換後, 右圖的投影點有重疊, 使得我們無法單從一個維度辨別在原座標系統中不同位置的兩個點。

都會緊臨座標軸, 以致於我們原本要計較的 error , 也就是新座標軸中的座標, 值都不大。但是這樣的座標轉換對我們而言有什麼意義?

其實 PCA 名氣會這麼大, 是因爲它可以拿來做資料壓縮。我們之前一直在說的「空間中的點」, 其實就只是一個向量, 他可以是一筆資料。一個簡單的想法, 一個 100 維的向量若是可以精簡成 10 維, 就可以將儲存空間壓縮成原來的 1/10 。問題就在於怎麼降維才能儘可能地保留原來資料的特性。簡單的說, 維度降下來以後, 我們希望原本分開的點仍然要分開, 就如圖 3所示, 左側圖示那樣的投影才是理想的投影。

「爲什麼右側的投影結果會比較差?」圖 3右側中的投影之所以會重疊, 直覺的想法, 是因爲座標軸之間有相關, 以致於拿掉其中一個軸, 就會使得座標點無法單獨被定義。不過這在 PCA 是不成問題的, 因爲 PCA 各軸之間是相互獨立的。我們底下就來證明這件事。

首先, 給定空間中散佈的 n 個 m 維的點, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$, 其平均值

$$\boldsymbol{m} = \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{x}_k. \tag{13}$$

令 $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 - \boldsymbol{m} & \boldsymbol{x}_2 - \boldsymbol{m} & \ldots & \boldsymbol{x}_n - \boldsymbol{m} \end{bmatrix}^T$。則 covariance matrix

$$\boldsymbol{S} = \frac{\boldsymbol{X}\boldsymbol{X}^T}{n-1}a. \tag{14}$$

我們接著對 $\boldsymbol{S}$ 做 eigen decomposition , 得到 m 個 eigen vector $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_m$, 以及相對應的 m 個 eigen value $\lambda_1, \lambda_2, \ldots, \lambda_m$。令 $E = \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \ldots & \boldsymbol{e}_m \end{bmatrix}$, 則

$$
\begin{aligned}
\boldsymbol{S}\boldsymbol{E} &= \begin{bmatrix} \boldsymbol{S}\boldsymbol{e}_1 & \boldsymbol{S}\boldsymbol{e}_2 & \dots & \boldsymbol{S}\boldsymbol{e}_m \end{bmatrix} \\
&= \begin{bmatrix} \lambda_1\boldsymbol{e}_1 & \lambda_2\boldsymbol{e}_2 & \dots & \lambda_m\boldsymbol{e}_m \end{bmatrix} \\
&= \begin{bmatrix} \lambda_1\boldsymbol{e}_{11} & \lambda_2\boldsymbol{e}_{21} & \dots & \lambda_m\boldsymbol{e}_{m1} \\ \lambda_1\boldsymbol{e}_{12} & \lambda_2\boldsymbol{e}_{22} & \dots & \lambda_m\boldsymbol{e}_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1\boldsymbol{e}_{1m} & \lambda_2\boldsymbol{e}_{2m} & \dots & \lambda_m\boldsymbol{e}_{mm} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{e}_{11} & \boldsymbol{e}_{21} & \dots & \boldsymbol{e}_{m1} \\ a\boldsymbol{e}_{12} & \boldsymbol{e}_{22} & \dots & \boldsymbol{e}_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{e}_{1m} & \boldsymbol{e}_{2m} & \dots & \boldsymbol{e}_{mm} \end{bmatrix} \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_m \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \dots & \boldsymbol{e}_m \end{bmatrix} \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_m \end{bmatrix} \\
&= \boldsymbol{E}\boldsymbol{D} \\
\Rightarrow \boldsymbol{S} &= \boldsymbol{E}\boldsymbol{D}\boldsymbol{E}^{-1}
\end{aligned}
\tag{15}
$$

其中

$$
\boldsymbol{D} = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_m \end{bmatrix} \tag{16}
$$

我們現在把空間中的點投射到 eigen vector 所張出來的空間, 則每個 $\boldsymbol{x}_k$ 都會得到新的座標 $\boldsymbol{y}_k$, 令$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \dots & \boldsymbol{y}_n \end{bmatrix}^T$ 則

$$
\boldsymbol{Y} = \boldsymbol{E}^T\boldsymbol{X}, \tag{17}
$$

而新座標所形成的 covariance matrix

$$
\begin{aligned}
\boldsymbol{S}_Y &= \frac{\boldsymbol{Y}\boldsymbol{Y}^T}{n-1} \\
&= \frac{1}{n-1}\boldsymbol{E}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{E} \\
&= \boldsymbol{E}^T(\frac{\boldsymbol{X}\boldsymbol{X}^T}{n-1})\boldsymbol{E} \\
&= \boldsymbol{E}^T\boldsymbol{S}\boldsymbol{E} \\
&= \boldsymbol{E}^T\boldsymbol{E}\boldsymbol{D}\boldsymbol{E}^{-1}\boldsymbol{E} = \boldsymbol{D} \qquad (18)
\end{aligned}
$$

這顯示了一個珍貴的情報: 當我們做完座標轉換, covariance matrix 會形成一個對角矩陣, 也就是說, 座標系統中任兩軸間的相關度是 0 !

不過這就代表我們可以隨便挑掉幾個軸來降維嗎? 當然不是! 我們希望取自相關度高的軸, 因爲這表示該軸對整組資料有比較高的代表性。這時候我們回頭去看一下矩陣 $\boldsymbol{D}$, 也就是投影過後的 covariance matrix , 對角線上擺得其實就是 eigen value , 也就是說, 要計較座標軸的 variance , 其實就只要看 eigen value! 所以事情變得再簡單不過了, 當我們做完 eigen decomposition , 我們看要把資料降成幾維, 假設是 k 維, 就挑相對應的 eigen value 最大的 k 個 eigen vector, 然後把資料點一一投影上去, 就得到新的座標值。

1.3 實作

假設我們有 n 筆 m 維的資料, 分別是 $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$, 我們的目標是把筆資料降維成 k 維。底下是實作的步驟。

1. **求出 covariance matrix。**
   令 $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \ldots & \boldsymbol{x}_n \end{bmatrix}^T$, $\boldsymbol{m} = \sum_{i=1}^{n} \boldsymbol{x_i}$ 。計算 $\boldsymbol{S} = \frac{1}{n-1}(\boldsymbol{X} - \boldsymbol{m})(\boldsymbol{X} - \boldsymbol{m})^t$ 。

2. **對 $\boldsymbol{S}$ 做 eigen decomposition 。**
   得到 m 個 eigen vector $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n$, 以及相對應的 eigen value $\lambda_1, \lambda_2, \ldots, \lambda_n$ 。

3. **選出前 k 大的 eigen value 所對應的 eigen vector , $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_k$。並投影資料點。**
   假設每一筆資料 $\boldsymbol{x}_i$ 投影後得到 $\boldsymbol{x}'_i = \begin{bmatrix} \boldsymbol{x}'_{i1} & \boldsymbol{x}'_{i2} & \ldots & \boldsymbol{x}'_{ik} \end{bmatrix}$, 則 $\boldsymbol{x}'_{ij} = \boldsymbol{e}_j^t(\boldsymbol{x}_i - \boldsymbol{m})$ 。

2 PCA with missing data

前面會花這麼多篇幅介紹 PCA , 是因爲若是不討論 PCA , 我們要探討的主題 PCA with missing data 就不完整。PCA with missing data 嚴格來說只多引入了一個步驟, 其餘的與 PCA 如出一轍。PCA with missing data 要多處理的一步, 就如同它名稱所揭示的, 就是資料遺失的問題。也就是說我們可能拿到如

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & * & * & \boldsymbol{x}_5 & * \end{bmatrix} \tag{19}$$

這樣的資料。因爲資料通常不容易收集, 我們不可能因爲資料不完整就全部作廢。但很明顯的, 其餘的處理不說, 光是 PCA 就會遇上麻煩, 因爲無法拿這樣的資料去算 covariance matrix 。所以 PCA with missing data 多出的一步就是要還原資料。最偷懶的方式, 就是直接把遺失的資料代換成現存資料的平均值, 不過當資料遺失過多, 平均值似乎也不能代表什麼。所以我們底下要介紹一個方法能比較正確的補回資料。

2.1 Wiberg's method

跟之前一樣, 假設我們有 n 筆 m 維的資料, 分別是 $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$, 令 $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \ldots & \boldsymbol{x}_n \end{bmatrix}^T \in \Re^{m \times n}$, 只是我們目前無法直接拿 $\boldsymbol{X}$ 來用, 因爲裡頭有許多的「洞」, 也就是遺失的資料。給個具體的例子, 我們很可能會需要處理像

$$\begin{bmatrix}
\boldsymbol{x}_1^{(1)} & \boldsymbol{x}_2^{(1)} & \boldsymbol{x}_3^{(1)} & \boldsymbol{x}_4^{(1)} & \boldsymbol{x}_5^{(1)} & \boldsymbol{x}_6^{(1)} & * & * & * & * & * & * \\
\boldsymbol{x}_1^{(2)} & \boldsymbol{x}_2^{(2)} & \boldsymbol{x}_3^{(2)} & \boldsymbol{x}_4^{(2)} & * & * & \boldsymbol{x}_7^{(2)} & \boldsymbol{x}_8^{(2)} & * & * & * & * \\
\boldsymbol{x}_1^{(3)} & \boldsymbol{x}_2^{(3)} & * & * & * & \boldsymbol{x}_6^{(3)} & * & \boldsymbol{x}_8^{(3)} & \boldsymbol{x}_9^{(3)} & \boldsymbol{x}_{10}^{(3)} & * & * \\
* & * & * & * & * & * & \boldsymbol{x}_7^{(4)} & \boldsymbol{x}_8^{(4)} & \boldsymbol{x}_9^{(4)} & \boldsymbol{x}_{10}^{(4)} & \boldsymbol{x}_{11}^{(4)} & \boldsymbol{x}_{12}^{(4)}
\end{bmatrix}$$

這樣坑坑洞洞的矩陣。我們的任務就是要找到一個矩陣 Y , 使得它可以很把洞填滿, 而且盡量近似於原來的資料。不過什麼叫近似原來的資料? 資料都遺失了, 我們怎麼

知道他原來長什麼樣子？ 我們目前唯一能要求的, 就是有的資料希望能夠保留下來。那沒有的資料呢？我們就必須使用類似 em 的方法來求解。

因爲資料是殘缺的, 我們沒有辦法得到資料的平均值, 爲了與之前確切的平均值作區分, 我們以 $\boldsymbol{\mu}$ 來表示, 代表平均值的 maximum likelihood approximation。於是, 我們的問題以比較正式的方式寫下, 就是要求 $\boldsymbol{Y} \in \Re^{m\times n}$, 使得

$$\Phi = \|\boldsymbol{X} - \boldsymbol{e}^T\boldsymbol{\mu} - \boldsymbol{Y}\| \tag{20}$$

最小, 其中 $\boldsymbol{e} = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^T$ 。在之前的討論中, 我們會先把平均值減去, 爲的是免去位移向量的動作, 不過現在平均值未知, 所以必須當作未知數一併求解。

SVD[1] 告訴我們, 任何一個 $m \times n$ 的矩陣 $\boldsymbol{M} \in \Re^{m\times n}$ 都可以分解成

$$\boldsymbol{M} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T, \tag{21}$$

其中 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 分別是 $m \times r$ 和 $n \times r$ 的正交矩陣, $\boldsymbol{S} = diag(\sigma_i)$ 是一個 $r \times r$ 的對角矩陣。當然, $\boldsymbol{Y}$ 也可以作 SVD 分解, 假設

$$\boldsymbol{Y} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{V}}^T, \tag{22}$$

我們另外令

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_1 & \dots & \boldsymbol{u}_m \end{bmatrix}^T = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{S}}^{\frac{1}{2}}, \tag{23}$$

$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{v}_1 & \dots & \boldsymbol{v}_n. \end{bmatrix}^T = \tilde{\boldsymbol{V}}\tilde{\boldsymbol{S}}^{\frac{1}{2}}, \tag{24}$$

$$\Rightarrow \boldsymbol{u}^T\boldsymbol{v} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{V}}^T = \boldsymbol{Y}. \tag{25}$$

所以我們就不需要去解整個 Y 矩陣, 取而代之的, 我們只要計算 $\boldsymbol{u} \in \Re^m$ 和 $\boldsymbol{v} \in \Re^n$ 兩個向量, 於是原來 $m \times n$ 的大矩陣, 只剩下 $m + n$ 個變數。

所以現在我們要求的有三個向量, 分別是 $\boldsymbol{u} \in \Re^m, \boldsymbol{v} \in \Re^n$ 和 $\boldsymbol{\mu} \in \Re^m$, 而我們已知的只有部分的資料。當然, 遺失的資料已經超出我們掌控的範圍, 我們只能根據現存的資料來求解。所以我們的問題需要改寫成

$$\min \phi = \frac{1}{2}\sum_I (\boldsymbol{X}_{mn} - \boldsymbol{\mu}_n - \boldsymbol{u}_m^T\boldsymbol{v}_n)^2, \tag{26}$$

$$I = \{(m, n) : \boldsymbol{X}_{mn}\ is\ observed\}.$$

[1]詳情可以參考 `http://www.uwlax.edu/faculty/will/svd/index.html`

與 Φ 相比, 我們現在要對每一個在 $\boldsymbol{X}$ 中有值的 entry, 去累加他的誤差, 並設法使得誤差最小。附帶一提, 其中的 $\boldsymbol{u}_m^T\boldsymbol{v}_n$ 其實就是 $\boldsymbol{Y}_{mn}$ 。

既然我們只使用部分的資料, 且是各個 entry 獨立的計算誤差, 我們實在沒有必要維持原來矩陣的形式, 我們可以把有值的 entry 拉成一條向量, 舉個例子,

$$\begin{bmatrix} 1 & * & 2 \\ 3 & * & * \\ * & 4 & 5 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}.$$

顯而易見的, 若是 $\boldsymbol{Y}$ 中 有 p 個 entry 是有值的, 我們就可以得到一向量 $\boldsymbol{w} = \begin{bmatrix} \Omega_1 & \dots & \Omega_{|\Omega|} \end{bmatrix} \in \Re^p$, 其中$\Omega = \{\omega = X_{ij} | X_{ij} \textit{ is observed}\}$ 。當然, 經過這樣的調整, $\boldsymbol{\mu}$也必須改變, 定義

$$\hat{\boldsymbol{\mu}} = \begin{bmatrix} \hat{\boldsymbol{\mu}}_1 \\ \vdots \\ \hat{\boldsymbol{\mu}}_{|\Omega|} \end{bmatrix}, \hat{\boldsymbol{\mu}}_k = \boldsymbol{\mu}_i \quad \text{if } \Omega_k \in \boldsymbol{X}_i. \tag{27}$$

承接上一個例子, 若是該矩陣相對應的 $\boldsymbol{\mu}$ 爲平均值, 且假設 $\boldsymbol{\mu} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^T$, 則 $\hat{\boldsymbol{\mu}} = \begin{bmatrix} 1 & 1 & 2 & 3 & 3 \end{bmatrix}^T$ 。基本上, 我們目前所做的努力無非是想讓原來的 ϕ 在拿掉遺失的資料後, 仍可以維持原來的等式。所以我們也必須對 $\boldsymbol{v}$ 和 $\boldsymbol{u}$ 做點小修正, 定義

$$\hat{\boldsymbol{v}} = \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_n \end{bmatrix} \in \Re^{r\times 1}, \hat{\boldsymbol{u}} = \begin{bmatrix} \hat{\boldsymbol{u}}_1 \\ \dots \\ \hat{\boldsymbol{u}}_m \end{bmatrix} \in \Re^{(r+1)\times 1}, \hat{\boldsymbol{u}}_i = \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{\mu}_i \end{bmatrix} \tag{28}$$

, 最後我們引入兩個矩陣 $\boldsymbol{B} \in \Re^{|\Omega|\times rn}$ 和 $\boldsymbol{G} \in \Re^{|\Omega|\times(r+1)m}$, 使得

$$\phi = \frac{1}{2}\boldsymbol{f}^T\boldsymbol{f}. \tag{29}$$

$$\boldsymbol{f} = \boldsymbol{w} - \hat{\boldsymbol{\mu}} - \boldsymbol{B}\hat{v} = \boldsymbol{w} - \boldsymbol{G}\hat{u} \tag{30}$$

定義

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_{11} & \boldsymbol{B}_{12} & \dots & \boldsymbol{B}_{1n} \\ \boldsymbol{B}_{21} & \boldsymbol{B}_{22} & \dots & \boldsymbol{B}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{B}_{|\Omega|1} & \boldsymbol{B}_{|\Omega|2} & \dots & \boldsymbol{B}_{|\Omega|n} \end{bmatrix},$$

$$\boldsymbol{B}_{ij} = \begin{cases} \boldsymbol{u}_m & \text{if } \boldsymbol{w}_i = \boldsymbol{X}_{mn} \text{ and } j = n, \\ \mathbf{0} & \text{otherwise.} \end{cases} \in \Re^{|\boldsymbol{u}|} \tag{31}$$

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_{11} & \boldsymbol{G}_{12} & \dots & \boldsymbol{G}_{1m} \\ \boldsymbol{G}_{21} & \boldsymbol{G}_{22} & \dots & \boldsymbol{G}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{G}_{|\Omega|1} & \boldsymbol{G}_{|\Omega|2} & \dots & \boldsymbol{B}_{|\Omega|m} \end{bmatrix},$$

$$\boldsymbol{G}_{ij} = \begin{cases} \begin{bmatrix} \boldsymbol{v}_n & -1 \end{bmatrix} & \text{if } \boldsymbol{w}_i = \boldsymbol{X}_{mn} \text{ and } j = m, \\ \mathbf{0} & \text{otherwise.} \end{cases} \in \Re^{|\boldsymbol{v}+1|} \tag{32}$$

爲了求 ϕ 最小, 我們將 ϕ 分別對 $\hat{\boldsymbol{u}}$ 和 $\hat{\boldsymbol{v}}$ 做偏微分。令

$$\frac{\partial \phi}{\partial \hat{\boldsymbol{u}}} = \boldsymbol{B}^T \boldsymbol{B} \hat{\boldsymbol{v}} - \boldsymbol{B}^T (\boldsymbol{w} - \hat{\boldsymbol{\mu}}) = 0 \tag{33}$$

$$\frac{\partial \phi}{\partial \hat{\boldsymbol{v}}} = \boldsymbol{G}^T \boldsymbol{G} \hat{\boldsymbol{u}} - \boldsymbol{B}^T \boldsymbol{w} = 0 \tag{34}$$

$$\Rightarrow \hat{\boldsymbol{v}} = \boldsymbol{B}^+ (\boldsymbol{w} - \hat{\boldsymbol{\mu}}) \tag{35}$$

$$\hat{\boldsymbol{u}} = \boldsymbol{G}^+ \boldsymbol{w} \tag{36}$$

其中 $\boldsymbol{B}^+$ 和 $\boldsymbol{G}^+$ 分別是 $\boldsymbol{B}$ 和 $\boldsymbol{G}$ 的 pseudo-inverse 。

所以若是要實作, 我們必須先猜測一組 $\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}$ 和 $\hat{\boldsymbol{\mu}}$ 並分成兩邊, 一邊是 $\hat{\boldsymbol{u}}$ 和 $\hat{\boldsymbol{\mu}}$, 另一邊則單放 $\hat{\boldsymbol{v}}$ 。一開始先固定其中一邊, 假設是 $\hat{\boldsymbol{v}}$, 則我們可以兜出 $\boldsymbol{G}$, 所以可以求出 $\hat{\boldsymbol{u}}$。反之, 固定另一邊, 則可以求出 $\hat{\boldsymbol{v}}$。反覆相同的步驟, 直到收斂爲止。當我們求出了 $\hat{\boldsymbol{u}}$ 和 $\hat{\boldsymbol{v}}$, 資料重建的工作也就大功告成了, 之後就就可以進行 PCA 的正常程序, 所以就不贅述了。

2.2 實作

1. 起始 $\hat{\boldsymbol{v}}$。

2. 更新 $\hat{\boldsymbol{v}} = \boldsymbol{B}^{+}(\boldsymbol{w} - \hat{\boldsymbol{\mu}})$

3. 更新 $\hat{\boldsymbol{u}} = \boldsymbol{G}^{+}\boldsymbol{w}$

4. 若是收斂就停止, 否則回到步驟 2 。