PRINCIPAL COMPONENT ANALYSIS

中文教學文件

姜任遠 文宗麟

一、前言

由於在網路上已至少有一分相當完整的 PCA 中文教學文件，經過仔細檢視之後，我們認為沒有必要重造車輪（reinvent the wheel），因此將該教學文件附上，而將心力集中於 PPCA 及 PCA 應用實例兩篇 Siggraph paper 的寫作上。

本教學文件架構如下，第二部分為 PPCA 教學部分，第三部分為 PCA 應用實例，試以兩篇 Siggraph paper 示範之。

二、PPCA 部分

==Introduction==

PCA [Principal Component Analysis] 是一種統計學上常用來降低問題維度的方法，而 PPCA (Probabilistic PCA) 這個概念主要是來自於 Michael E. Tipping 以及 Christopher M. Bishop 1998 年的論文，主要的概念是將 PCA 用機率的方式來表示。
本篇 tutorial 前半部先補充一些相關數學概念，大部分都和機率統計有關，接下來再來說明 PPCA 的運理。簡單的來說，他們想要做的事情就是 build a probabilistic model associated with PCA，這樣的好處除了可以除了 PCA with missing data 的情形，也能藉由 mix 多組 PCA 來表示更複雜的資料(Mixture PPCA)。我們稍後在文件中會指出詳細的用法。

Part 1
==Background Probability and Statistics==
這部分提到的數學都是想要了解 **PPCA** 所必需的，在此只做簡單提示，詳見參考資料。

Gaussian Distribution
Likelihood Function
Maximum Likelihood Estimation
Mixture Model
Expectation Maximization

Part2

Probabilistic Principal Component Analysis

Question1

基本上 PPCA 就是想要 build a probabilistic model associated with PCA。我們要怎麼樣達到這樣的目的呢?這個 probabilistic mode 怎麼無中生有出來?

Answer1

首先我們先假設有一個"Parametric probability distribution model" which contains a number of adaptive parameters. 我們要做的事就是用手頭上現有的觀察到的 sample data 來決定這些 parameters 最適當的值，例如一堆點已經確定是依據 Gaussian distribution 來撒的，我們不知道這個 Gaussian distribution 的 parameters, 這是一個 parametric probability distribution function，但我們能用手的上的 sample data 來 evaluate 最適合的 parameters 例如平均值及標準差。

Question2

怎麼樣得到這個 parametric probability distribution model?

Answer2

由 Latent Variable Model 以及 Factor Analysis 得來。

既然你都問了，我就順便告訴你這兩個東西是什麼。為什麼要提 latent variable model 能，全都是因為我們想用"機率"來表達 PCA。假設我們有一些已知 variables 組成的 distribution function 但想用更小數目的 latent variable 來表達原來這個 distribution function，這時就要用到 latent variable model。

那什麼又是 factor analysis 呢? 它是其中一個最常見的 latent variable model：

$$\mathbf{t} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} + \boldsymbol{\epsilon}.$$

d-dimensional observation vector $\mathbf{t}$

q-dimensional vector of latent (or unobserved) variables $\mathbf{x}$ $\quad q < d$

($\mathbf{x}$ as common factors; $\boldsymbol{\epsilon}$ as special factors or noise)

依照慣例 latent variable

$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$,

$\epsilon \sim \mathcal{N}(\mathbf{0}, \Psi)$

我們在此做一個假設，

$$\epsilon \sim \mathcal{N}(0,\ \sigma^2\mathbf{I})$$

則 observed data

$$\mathbf{t} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C}) \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$$ 。

我們知道 Gaussian distribution 也能夠寫成 probability distribution function 的形式，因

此在 Factor Analysis 中，我們就能用低維度的 latent variable 來表示高維度的 data 機率分佈。當然這其中包含了許多參數，像 $\mathbf{W}$ 以及 σ^2。這就是我們所謂的 parametric probability distribution mode。

在此我們得到 observed variable vector $\mathbf{t}$ 的 parametric probability distribution function 爲：

$$p(\mathrm{t}; \mathrm{W}, \sigma^2) = (2\pi)^{-d/2}|\mathrm{C}|^{-1/2} \exp\left\{-\frac{1}{2}(\mathrm{t}-\mu)^{\mathrm{T}}\mathrm{C}^{-1}(\mathrm{t}-\mu)\right\} \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$$

Question3

咦? 那 Factor Analysis 和 PCA 有什麼關係呢?? 怎麼突然扯到它?

Answer3

你仔細看 FA 的數學表示式

$$\mathrm{t} = \mathbf{W}\mathbf{x} + \mu + \epsilon. ,$$

x 代表低維變數，t 代表高維的資料，我們主要是想用低維的”factor”來表示高維的資料。你在想想 PCA 的數學表示式是什麼呢?當 PCA 做 dimension reduction 的時候，也就是找 principal component 的時候

$$\mathrm{x}_n = \mathrm{W}^{\mathrm{T}}(\mathrm{t}_n - \bar{\mathrm{t}}) ,$$

在做 data reconstruction 的時候

$$\hat{\mathbf{t}}_n = \mathbf{W}\mathbf{x}_n + \bar{\mathbf{t}} 。$$

所以基本上，factor analysis 是可視做 PCA 的 data reconstruction 加上一個 noise 的 term，以及導入機率(probability distribution) 的概念。

Question4:

那要怎麼解這個表示 PCA 的 parametric probability distribution model 呢?

Answer4:

我們用的方法叫做 Maximum Likelihood。什麼是 likelihood 呢?在前面我們已經說過，它有點像是 conditional probability 的相反。Likelihood 就是 P(A|B) consider B given A fixed。一般用 ML 來解的問題都能表達成一個 likelihood function 的形式：

$$\mathrm{lik}(\theta) = f_D(x_1, \ldots, x_n \mid \theta)$$

Given a probability distribution *D*, sample x and distributional parameter θ

θ 就是我們要解的參數。已知 observed data x 的情況下，我們想知道在怎樣的 θ 能讓 function 達到最大值，這樣用 ML 解出來的 θ 又稱作 maximum likelihood estimator。

回頭看我們已知的 parametric probability function：

$$p(\mathbf{t}\,;\mathbf{W},\sigma^2) = (2\pi)^{-d/2}|\mathbf{C}|^{-1/2}\exp\left\{-\frac{1}{2}(\mathbf{t}-\mu)^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{t}-\mu)\right\} \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$$

在知道 observed data 的狀況，我們要求 $\mathbf{W}$ 以及 σ^2。

如果要用 ML，首先我們先得到 likelihood function 為

$$\mathcal{L}(\mathbf{W},\sigma^2) = -\frac{N}{2}\left\{d\ln(2\pi) + \ln|\mathbf{C}| + \mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{S}\right)\right\} \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$$

這只是把上列 function 經過一個取 log 的動作使它變成 strictly increasing。因為 ML 使用的概念就像是微分求 function 的極值，所以我們必需要 strictly increasing function。
詳細 ML 解法請見論文。
所得出來的 MLE(maximum likelihood estimator)，也就是 parametric probability function 的參數為：

$$\mathbf{W}_{\mathrm{ML}} = \mathbf{U}_q(\mathbf{\Lambda}_q - \sigma^2\mathbf{I})^{1/2}\mathbf{R},$$

$$\sigma^2_{\mathrm{ML}} = \frac{1}{d-q}\sum_{j=q+1}^{d}\lambda_j$$

Question5
到底和 PCA 有何關聯?
Anwser5
從 Question3 我們已經可以看出 factor analysis 和 PCA 的相似之處，由 FA 所導得的 parametric probability function 求解後，發現參數 $\mathbf{W}_{\mathrm{ML}}$ 的 columns，就是求 principal component 時 sample 的 covariance matrix 的 eigenvector 經過 scale 和 rotate 得來。
So, from now on, PCA can be expressed in terms of a probability density model---PPCA.

Question6
總結一下吧!
Answer6
Please allow me to speak in English…
This is how PPCA works: Given an observation sequence t (high dimension), the PPCA model estimates the latent variable sequence x (low dimension) and finds the optimal parameter set $\mathbf{W}$ σ^2 according to the maximum likelihood criterion.

Question7
解說了這麼久怎麼得到 PCA 的機率表示法，辛苦你了 Orz…

不過還是得說說有什麼應用吧…

Answer7

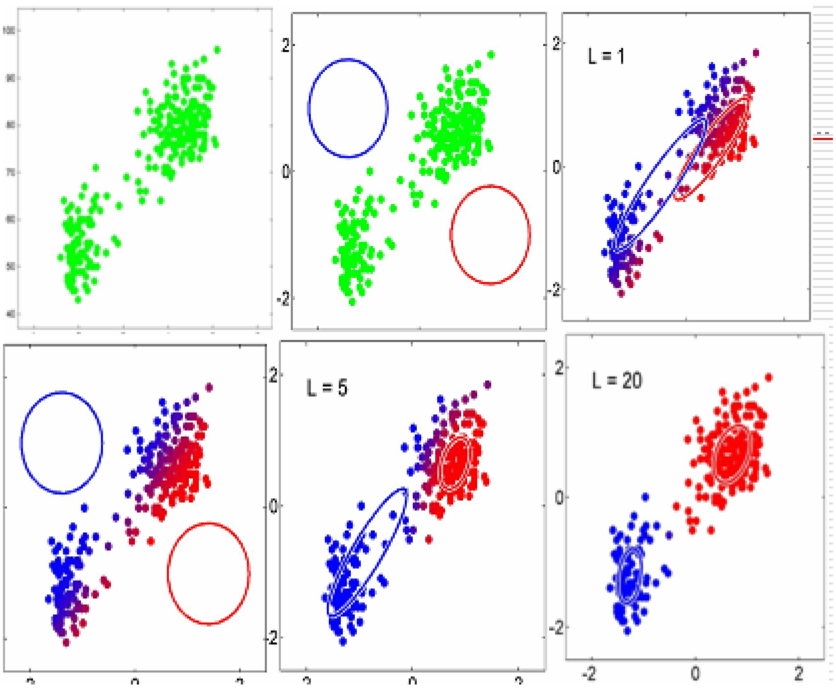
(汗…)

既然我們可以把 observed data 用機率的分佈方法來表達，我們就能組合多組 PPCA 來表示同一組 data---mixture PPCA，就像 mixture Gaussian function 一樣。

$$P(x_i) = (1-f)\mathcal{N}(x_i;\mu_1,\sigma) + f\mathcal{N}(x_i;\mu_2,\sigma)$$

以上是兩組 Gaussian function 的 mixture。

可用 iterative 的方法(Expectation Maximization)來找出 observed data 可以由那些 PPCA 的組合來組成：(詳見論文)



=Reference=

[1] Wikipedia

[2] Google

[3] “Probabilistic Principal Component Analysis”, Journal of the Royal Statistical Society, Series B, 61, Part 3, pp. 611-622.

三、PCA 應用實例部分

以下試以兩篇 Siggraph paper 來作 PCA 的應用實例。

A Morphable Model For The Synthesis of 3D Faces

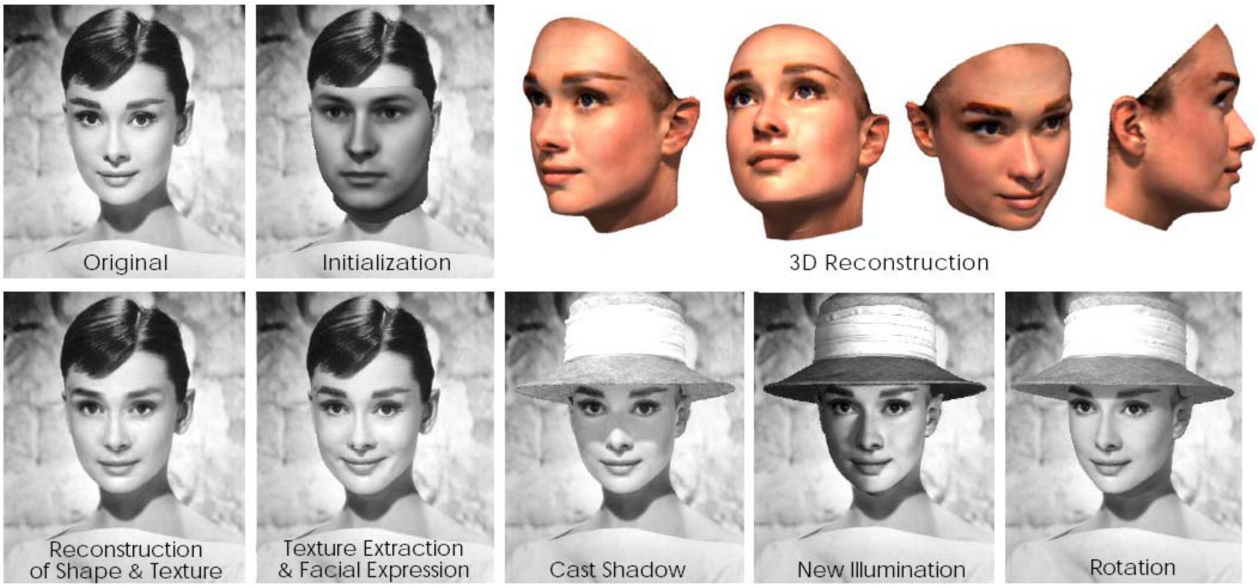
(published in Siggraph 1999)

[摘要]

簡單來說，這篇 paper 是希望藉由大量的 3D face database 作爲奧援，經由文中所提出的處理方法，達到以下兩件事情：

1) 給定一張 2D face image，產生一個 3D face model
2) 讓臉形、臉上的材質變化作到儘可能的自然

爲了能夠更快的了解上述功能，建議讀者在此可前往本 paper 官方網站上下載 demo 影片來看。



在影片中，系統被給定了一張奧黛莉赫本的 2D face image。系統便生成一個 3D face model，利用這個資訊，我們可以替她戴上一頂帽子，同時還能產生自然的打光。

至於上面提到的第二點，paper 作者則是以黑白照片轉換爲彩色照片來作示範，請注意膚質的轉換是相當自然的，而不只是單純的內插。

以下，本文作者將儘可能簡要的說明此系統之演算法，而將重點放在 PCA 是如何被運用此一系統上。細節部分，仍請參照該篇 paper。

[演算法]

[資料庫]

首先必須了解此系統有一資料庫在後方作爲奧援。總計掃描了 200 張年輕成人（男女各半）
的臉部。

而人的頭部大致上可以看作是一個 cylinder，因此，我們將此假想中的 cylinder 依高度、角度各切成 512 等分。

此時，給定高度 h，角度 ，我們可以得到：

1) Shape: 該點與 cylinder 中心軸之距離 r(h, φ)
2) Texture: 該點之色彩 R(h, φ) , G(h, φ),   B(h, φ)

由於每張臉都是作如此畫分，我們可以得到各點之間的 full correspondence。

[Morphable 3D Face Model]

經過資料採集之後，每張臉應該都有自己的 shape vector, texture vector: S_i, T_i。

將之平均之後可求得整個資料庫的 shape 與 texture 之 average vector：$\overline{S}$, $\overline{T}$

而 new face 可視爲資料庫已有 old face 的線性組合，所以各係數抽取出來就可以完整描述一張 new face：

$$\mathbf{S}_{mod} = \sum_{i=1}^{m} a_i \mathbf{S}_i \,, \quad \mathbf{T}_{mod} = \sum_{i=1}^{m} b_i \mathbf{T}_i \,, \quad \sum_{i=1}^{m} a_i = \sum_{i=1}^{m} b_i = 1.$$

但是這邊要注意的是，一個好的 system 一定要能夠知道怎樣的臉是可能的怎樣是不可能的，換句話說，這個 a 跟 b 並不是隨便可以亂動，有一定範圍在那邊。

那麼，現在是 PCA 登場的時候了，這篇 paper 裡面 PCA 被使用了兩次。現在我們要講解的這個部分主要是在作 data compression:

首先我們先用 $\Delta S_i = S_i - \overline{S}$, $\Delta T_i = T_i - \overline{T}$ 計算出其 covariance matrix C_S, C_T，接著我們對 C_S, C_T 作 PCA，可以得到其 eigenvector s_i, s_t。

那麼，我們可以把原本的表示方法轉換成下列方式：

$$S_{model} = \overline{S} + \sum_{i=1}^{m-1} \alpha_i s_i \,, \; T_{model} = \overline{T} + \sum_{i=1}^{m-1} \beta_i t_i \,,$$

我們可以看到，原本是對每一個各別的 shape vector, texture vector 作 linear combination，現在則是對一個共用的座標軸（s_i, s_t 本身爲一組 orthogonal basis）

作 linear combination。資料量與計算量明顯少了許多。

[Matching Morphable Model to Images]

以下，講解本篇 paper 很重要一個應用，亦即把 3d 的 morphable model 自動化地 match 到一個 2d 的 image 上面去，以及 PCA 如何居中幫助此一功能的進行。

簡明扼要地說，使用者先把環境變數$\vec{\rho}$設好，再去找一組α_j, β_j,可以讓 I_{model} 跟 I_{input} 的距離最小。(分別對 RGB 而言)

但是這種把 3d model 投到一個平面上與 given image 作比較的事情其實不算是定義很嚴謹的一個問題，因爲其解的可能性太大了。

所以我們的α_j, β_j要受原本資料庫中的α_i, β_i限制，不能長出太奇怪的臉。

並且，爲了避免卡在 local minima，所以我們 induct 一個 coarse-to-fine 的 stategy，此時，PCA 再度派上用場：
首先先對 image 作 down sampling。此外，model 能夠調的部分，我們一開始固定在 first principal component 就好。在之後的 iteration 才慢慢把其他的 principal component 加上來。

The Space of Human Body Shapes

(published in Siggraph 2003)

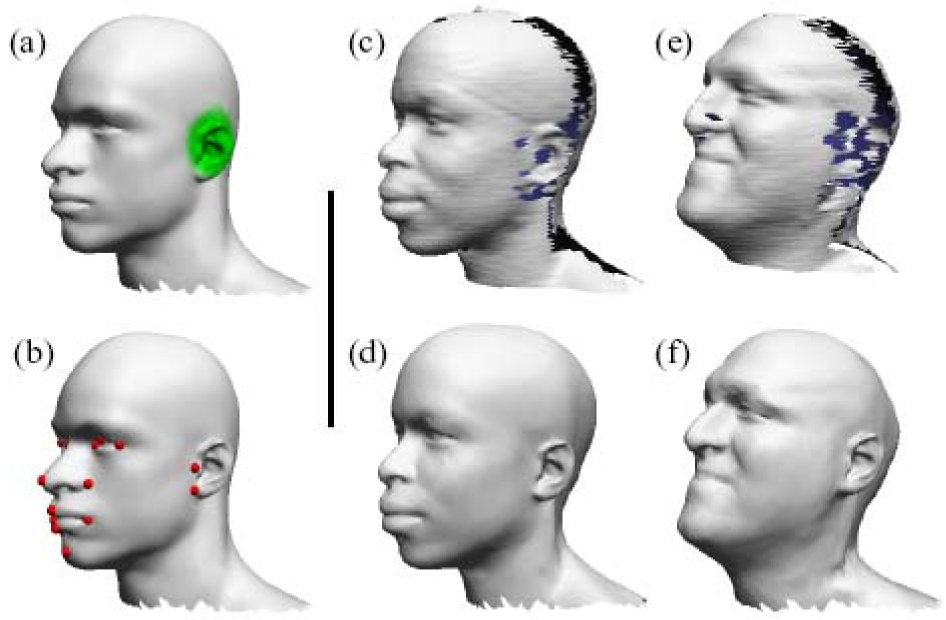
[摘要]

與上一篇 "A Morphable Model For The Synthesis of 3D Faces" 相同的是，本 paper 企圖作出一個 morphable 的 model，只不過差別在於這次的對象是整個 human body 而不只是臉部。

當然最簡單的方法就是 3d scan。可是這很費時，有洞還得自己去補。再者，只告訴我們個人的體型，而不是一群人。最後，要作編輯也挺麻煩的。所以最佳的解決之道就是造一個 morphable model 出來。

而難度在哪裡呢？與上一篇相較的話，body 沒辦法單純用一個 cylinder 來表示，換句話說，model 之間要建立 vertex to vertex 的 correspondence 就比較難。（前者可依據高度、角度是否相同一一判斷）

建出這樣的 model 之後，能有許多的應用，比方說我們在作 3D scan 的時候，遇到像耳朵這樣的地方難免會產生破洞，我們就可以利用這個 morphable model 根據某些準則與目標 model 作 matching 將破洞補起來。



[演算法]

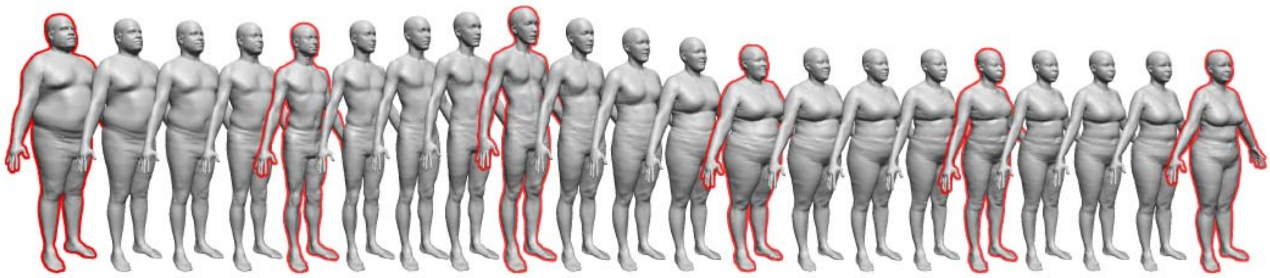
[Morphable Model 的建立]

最開始，有幾件事情是必須手動處理的。首先，我們的第一個 model 是由 artist 所繪製而成的。再來，我們必須手動設定此一 template 與其他 model 的少數 vertex 的一一對應關係，在本篇 paper 的術語中，稱之為 marker。

那麼接下來就可以來作對應了，我們可以把如何將一個 template 點對點地 match 到 target surface D 這件事情，reduce 成 minimize a set of error function。

[PCA 可資運用之處]

建立好 vertex to vertex 一一對應的關係之後，我們可用與上篇 paper 同樣的方式來作 data compression。



上圖中，紅色標起來的是 key model，其他是內插出來的。

比較特別的是說，這些 interpolated model 每個都有一些跟 origianl model 不一樣的獨特之處，所以說不完全只是"interpolate"

至於作法也很簡單，其實就是在經過 PCA 轉換的座標上做 Gaussian sampling。如此一來，將可造出無限多種原本不存在的 model，大部分有相當真實的外表，但卻有少部分與原來的 model 都不相同。

[Conclusion]

當然這兩篇 paper 不僅止於此，不過基於 tutorial 的本質，對本 paper 的介紹就僅限於與 PCA 較相關的部分。