LLE & Isomap Tutorial

Machine Learning for Graphics, Vision and Multimedia, 2006
Communication and Multimedia Laboratory
CSIE, National Taiwan University

許平 vivace@cmlab.csie.ntu.edu.tw
謝昌熹 isaddo@cmlab.csie.ntu.edu.tw

Dimensionality Reduction

高維度的資料往往很難描述、計算，一個常用的方法是假設這些資料並非真的存在於這麼高的維度上，也就是說，可以用一個較低維度的非線性流形 (non-linear manifold) 來模擬這些資料。流形(Manifold)，一般可以認爲是局部具有歐氏空間性質的空間。而實際上歐氏空間就是流形最簡單的實例。像地球表面這樣的球面是一個稍爲複雜的例子。一般的流形可以通過把許多平直的片折彎並粘連而成。

如果這個 manifold 的維度夠低，我們就可以在這個低維度的空間上視覺化原先的資料。降維的方法可以概括分成以下三種：

1. 線性方法(Linear methods)

   Principal component analysis (PCA)

   Singular value decomposition (SVD)

   Factor analysis (FA)

2. 非線性對應(Non-linear mappings)

   Generative topographic mapping (GTM)

   Gaussian process latent variable models (GPLVM)

   Neural network methods

3. 逼近法(Proximity)

   Multidimensional scaling (MDS)

   Isomap

   Locally linear embeddings (LLE)

本 Tutorial 著重在第三類的方法，這些方法通常透過計算相似矩陣(similarity matrix)或距離矩陣(distance matrix)來找到各資料之間的關係。

MDS

Multidimensional scaling (MDS)是一群尋找物件之間相關性的方法的統稱，例如，給予一個 Matrix D 描述各個城市之間的距離，MDS 會嘗試找到一個最好的配置方法來描述這些城市。

通常 MDS 會定義一個 stress value，若在 m 維空間上有兩物體 X_1, X_2，座標分別爲 $(x_{11}, x_{12}, x_{13} ... x_{1m})(x_{21}, x_{22}, x_{23} ... x_{2m})$，最簡單的方法是定義尤拉空間的距離當作兩物體的距離 d

$$f(d_{i,j}) = (\Sigma(x_{ia} - x_{ja})^2)^{1/2}$$

若今天有 N 個點就可以利用這個 d 來定義 stress

$$S = \sum_{i=1}^{N} \sum_{j=1}^{N} (D_{i,j} - f(d_{i,j}))$$

MDS 想要找到描述各物體之間的最好方法，在這個方法下所描述的距離，就是 $D_{i,j}$。至於如何找到這個描述，不同的 MDS 使用的方法都不盡相同。

以 MMDS 爲例，它解一個 eigenvalue problem，輸出一組座標系，基本上這跟 PCA 是殊途同歸。

MMDS

1. start with distances $d_{i,j}$
2. define $A = a_{ij} = -\frac{1}{2}(d_{i,j})^2$
3. put $B = a_{ij} - a_{i*} - a_{*j} + a_{**}$, where $a_{i*} = \frac{1}{n}\sum_j a_{ij}$, $a_{*j} = \frac{1}{n}\sum_i a_{ij}$, $a_{**} = \frac{1}{n^2}\Sigma\Sigma a_{ij}$
4. find the eigenvalues Λ and the associated eigenvectors Γ of B
5. the coordinate matrix is $X = \Gamma\Lambda^{\frac{1}{2}}$

Isomap

Step 1

Isomap 的 input 是許多高維度的 data，並把它們當作一個 graph，只要兩個 vertex 是鄰居，就會有一條 edge 連結，至於鄰居的判定方法可以是 K-nearest neighbors 或是用直接距離再取 threshold 都可以。

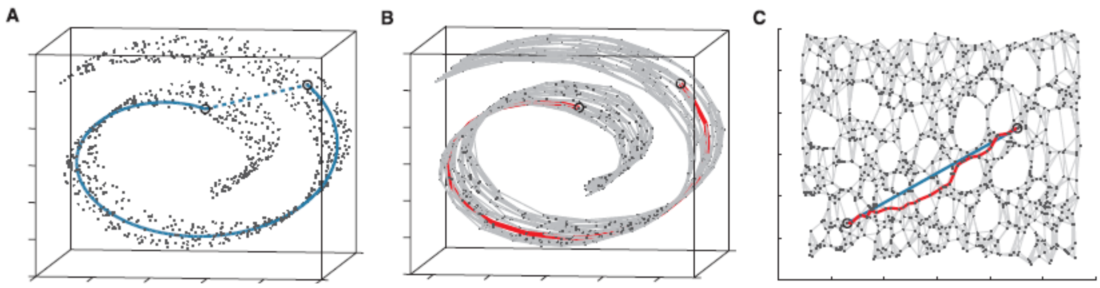
Step 2

接著，利用 Floyd's Algorithm 算出每個 vertex 之間的 shortest path distances。

Step 3

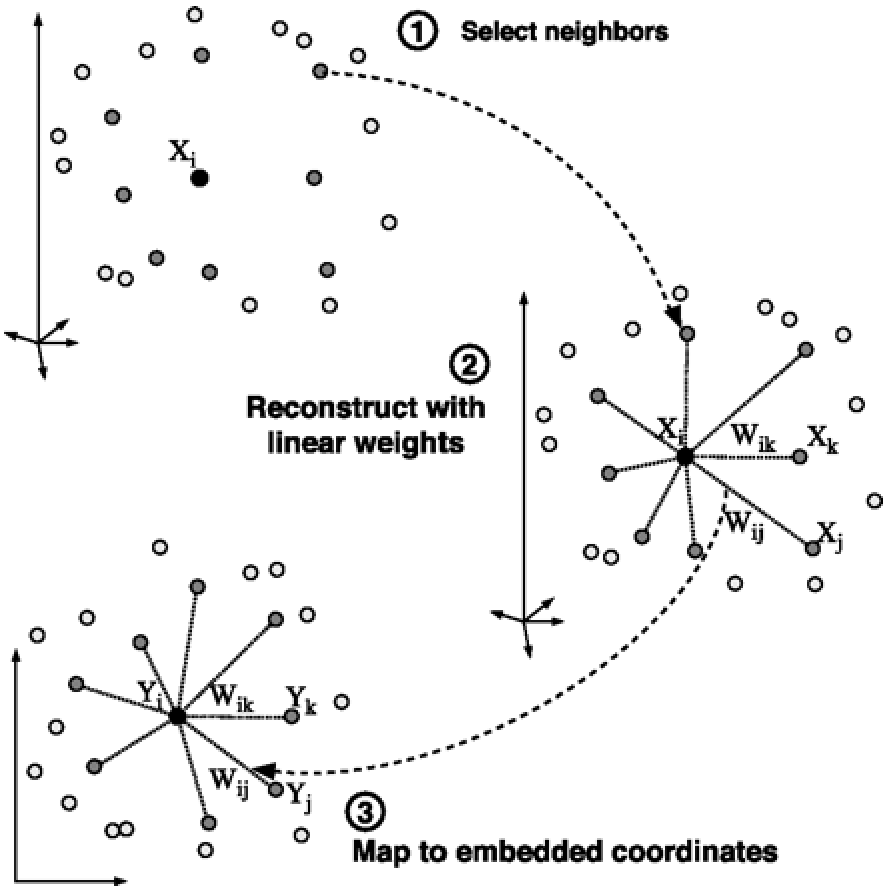
最後，把 Step 2 當中的結果當作 MMDS 的 input，就可以得到一個座標軸，利用這個座標軸描述出的 data 就是一個低維度的 manifold。

Isomap 的演算法雖然簡單，但確實解決了 PCA 或其他 linear methods 在 non-linear manifold 上遇到的問題，透過"neighbor"的定義，加強了各 data 之間的連結性，而不是只以絕對距離當作衡量的方法。



Locally linear embeddings (LLE)

LLE 演算法使用幾何上的直覺，假設有 N 個 data $\vec{X}_i$ ，$i = 1 \sim N$ ，每一個都是 D 維，這些 data 是從某一個 manifold 上取出來的。只要 data 數目夠多，雖然我們並不知道這個 manifold 長什麼樣子，但是我們可以預期在一個局部的範圍內，每個 data 跟他的鄰居可以形成一塊 patch，而這個 patch 會非常接近原本的 manifold。



Step1

我們可以利用各個鄰居的線性組合來重建每個 data，以用來描述這些 patch。重建時的 error 可以寫成 cost function

$$\varepsilon(W) = \sum_{i} \left| \vec{X}_i - \Sigma_j W_{ij} \vec{X}_j \right|^2$$

把每個 data 與其鄰居的重建結果相減，其中 W_{ij} 是矩陣 W 的一個元素，表示

第 j 個 data 對第 i 個 data 的影響力(weight)，爲了計算出W_{ij}，我們需要把 cost 最小化，並且符合了兩個條件

1. 只有$\vec{X}_i$的鄰居對$\vec{X}_i$有影響，也就是說如果 j 不是 i 的鄰居，則$W_{ij}=0$。

2. 這個 Weight 矩陣每列的總合爲 1，$\Sigma_j W_{ij} = 1$。

Step 2

這個問題是有固定解的。這裡我們先考慮其中一個 data $\vec{x}$，假設他有 K 個鄰居$\vec{\eta}_j$， $j = 1 \sim K$ ，我們可以把 cost 寫成

$$\varepsilon = \left|\vec{x} - \sum_j w_j \vec{\eta}_j\right|^2 = \left|\sum_j w_j \left(\vec{x} - \vec{\eta}_j\right)\right|^2$$

把整個絕對值展開的話，可以觀察到每一項都是 $w_i w_j$ 的形式，係數則是$(\vec{x}-\vec{\eta}_j)(\vec{x}-\vec{\eta}_k)$ ，也就是說我們可以寫成

$$\left|\sum_j w_j \left(\vec{x} - \vec{\eta}_j\right)\right|^2 = \sum_{jk} w_j w_k C_{jk}$$

其中$C_{jk} = (\vec{x}-\vec{\eta}_j)(\vec{x}-\vec{\eta}_k)$ ，作者把這個 C 矩陣稱爲 local covariance matrix。

根據拉格蘭日乘數法(註一)，如果把 *wj* 當作變數，$\Sigma_j W_{ij} = 1$ 當作限制(constrain function g)，想求 ε 的最小值。則 Lagrange 方程式對每個 *wj* 作偏微分要是 0，可以寫成

$$\frac{\partial}{\partial w_j}(\varepsilon + \lambda(g-1)) = 0$$

總共有 N 個偏微分的式子，因此這是一個 N 元 N 次方程組，整理過後可以得到。

$$w_j = \Sigma_k C_{jk}^{-1}(\vec{x} \cdot \vec{\eta}_k + \lambda)$$

Step 3

假設這些 data 會坐落或是接近一個平滑的非線性低維度的 manifold 上，什麼是好的估計模擬呢？就是能找到一個線性的轉換(包括平移、旋轉、伸縮)把原本 data 的鄰居從高維度的座標系轉換成這個 manifold 的座標系上。

根據這個概念，LLE 建立一個保留鄰居關係的轉換。在演算法的第三步，每個高維的 data X_i 會被對應到一個低維的 vector Y_i，而這些 Y_i 可以形成這個 manifold，且它的維度是 d。(d << D)

Y_i 的選擇必須要能夠最小化 cost function

$$\min_{Y} \quad \Phi(Y) \quad = \quad \sum_i \left| \vec{Y}_i - \sum\nolimits_j W_{ij} \vec{Y}_j \right|^2$$

如果把絕對值展開，會有三個 component，可以發現這是一個 $Y_i Y_j$ 的 quadratic form，對任意 a,b，$Y_a Y_b$ 的係數只會出現在(1) $i = a, j = b$ (2) $i = b, j = a$ 這兩種情況。對第一個 component，如果 $a = b$ 則係數是 1，其餘情況是 0，對第二個 component 會分別產生 W_{ab} 與 W_{ba} 係數，對第三個 component，會有 $W_{1a}W_{1b} + W_{2a}W_{2b} + \ldots$ 的係數，整理以上結果，可以把係數寫成

$$M_{ij} = \delta_{ij} - W_{ij} - W_{ji} + \sum_k W_{ki} W_{kj}$$

也就是說

$$\Phi(Y) = \sum_{ij} M_{ij} (\vec{Y}_i \cdot \vec{Y}_j)$$

根據 Rayleitz-Ritz theorem，如果這個 manifold 只有 d 維的自由度，那麼從第 d+1 之後的每個 eigenvector 都是對應到 eigenvalue 0，因此只要保留 M 矩陣的前 d 個 eigenvector，就可以形成我們想要的 manifold。

(註一)以二維方程式爲例，要找 $f(x, y)$ 極值的條件是：

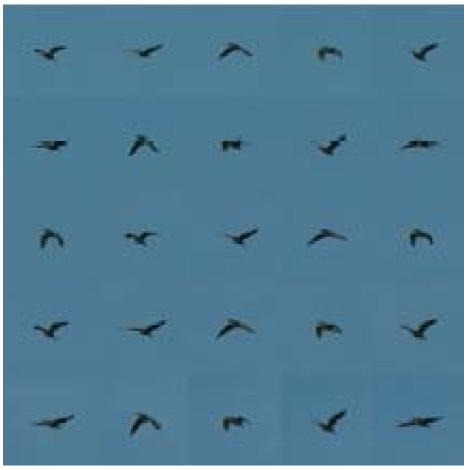
$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = 0$$

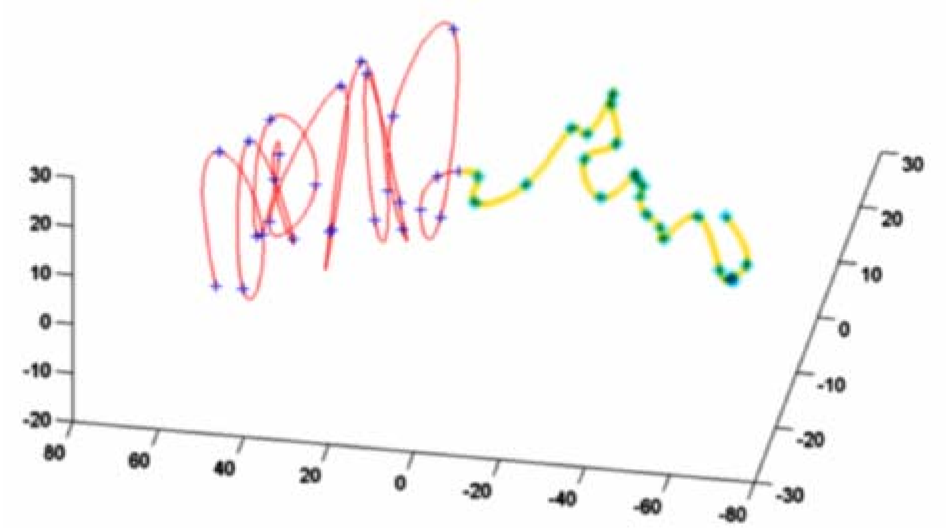
但是如果 $x.y$ 的範圍一開始就被另一個函數 $g(x, y) = 0$ 所限制，Lagrange 提出以 $f(x, y) + \lambda g(x, y)$ 對 x 和 y 的偏微分爲 0，來代替 $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = 0$ 作爲在 $g(x, y) = 0$ 上面尋找 $f(x, y)$ 極值的條件。式中引入的 λ 是一個待定的數，稱爲乘數，因爲是乘在 g 的前面而得名。

Isomap & LLE Applications

Image Spaces and Video Trajectories: Using Isomap to Explore Video Sequences
Robert Pless
ICCV 2003

在這一篇 paper 中作者把 video 中的每個 frame 當作 Isomap 的 input，很顯然的這些 input 都是高維的 data，經過 Isomap 的轉換之後，會變成一群較低維度的 data，最後再按照 video 中的順序連起來，就可以形成一個軌道，作者認爲這個軌道可以提供許多有用的資訊，例如在影片分段的例子中，把鳥的飛行影片轉換後，觀察軌道可以發現存在兩種不同的飛行模式，而正好可以對應到展翅與滑翔兩種動作。



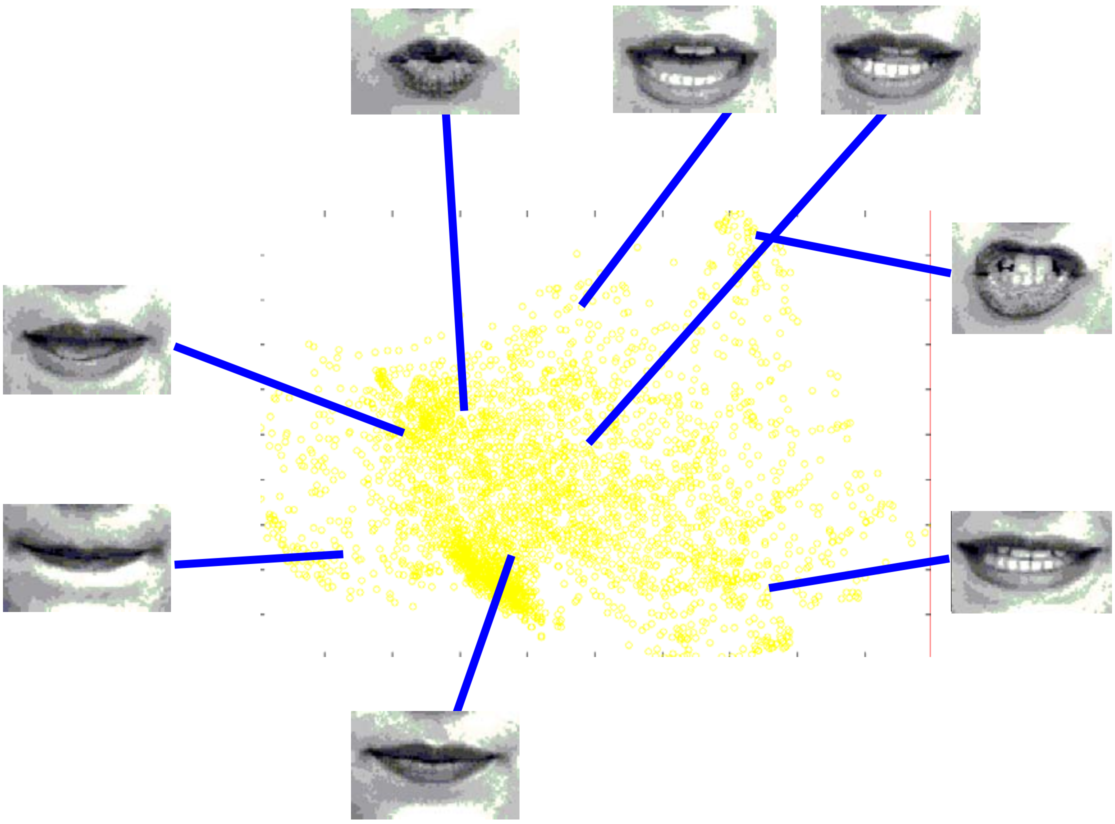


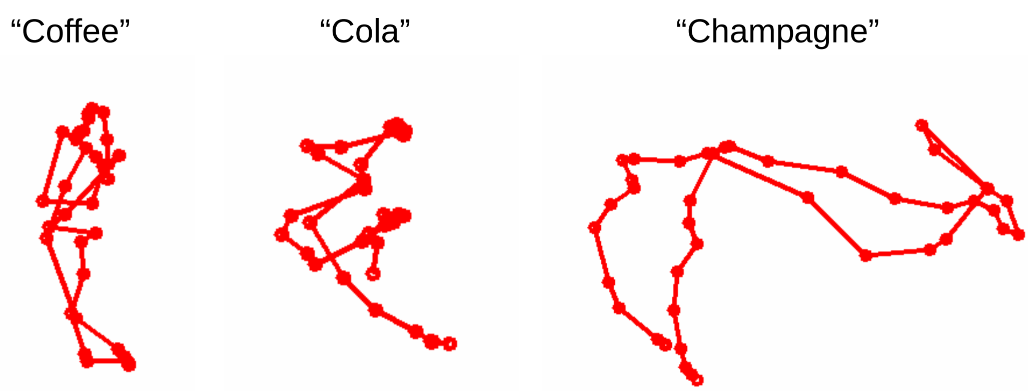
Representation Analysis and Synthesis of Lip Images Using Dimensionality Reduction

Michal Aharon and Ron Kimmel

IJCV 2005

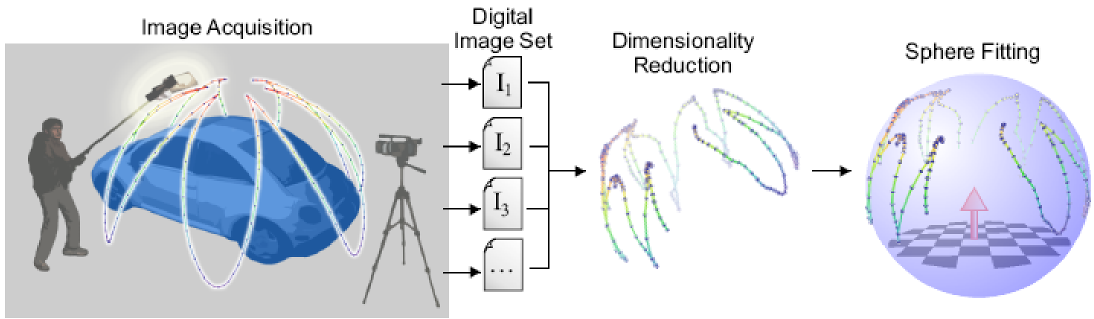
在這篇 paper 中作者把說話的嘴型當作 LLE 的 input，然後形成一個低維度的 manifold，接著找到每個單字在 manifold 上面形成的曲線，透過分析這些曲線可以達到讀唇語的效果。

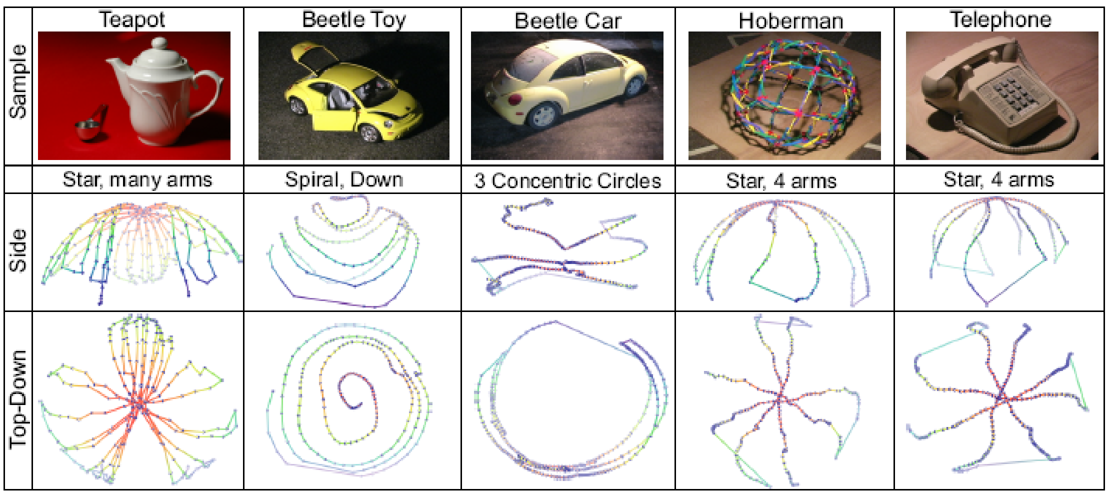




Light Waving: Estimating Light Positions From Photographs Alone
Holger Winnemöller, Ankit Mohan, Jack Tumblin, Bruce Gooch
Eurographics 2005

在這篇 paper 中作者拍攝同一個角度但不同光源的物體，將這些照片當作 Isomap 的 input，distance 的算法則是使用常見的 L2-norm，找到低維度的 manifold 之後，再把它 fit 到一個球表面上，藉此重建 light position，用來當作 rendering 或 relighting 的光源參考。





References

1. Stephen Borgatti, Multidimensional Scaling.

2. Joshua Tenenbaum, Vin de Silva, John Langford, A Global Geometric Framework for Nonlinear Dimensionality Reduction, Science, 2000.

3. Sam Roweis, Lawrence Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, Science, 2000.

4. Lawrence Saul, Sam Roweis, An Introduction to Locally Linear Embedding.

5. Lawrence Saul, Sam Roweis, Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds, Journal of Machine Learning Research, 2003.

6. Unsupervised Learning of Image Manifolds by Semidefinite Programming