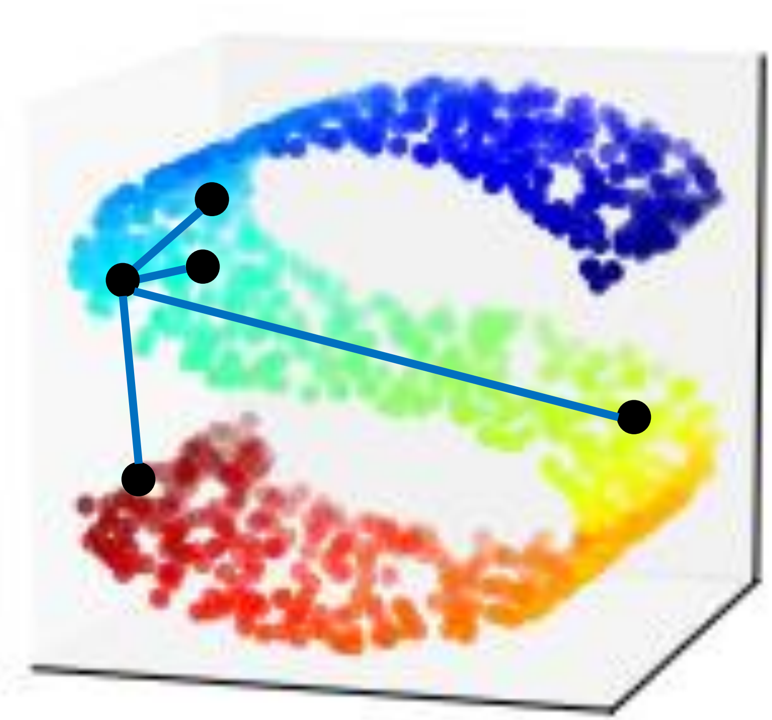
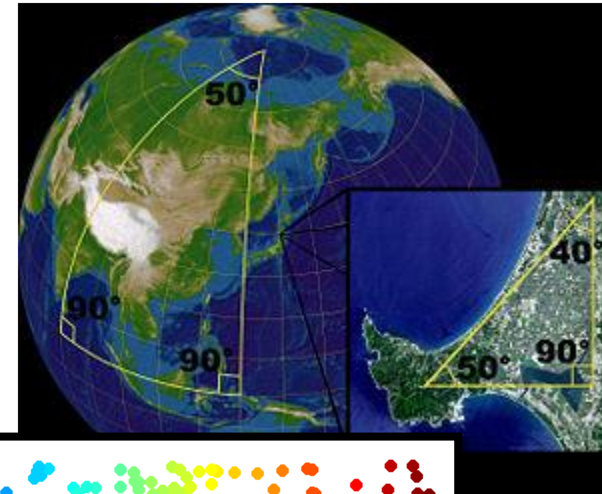
Unsupervised Learning: Neighbor Embedding

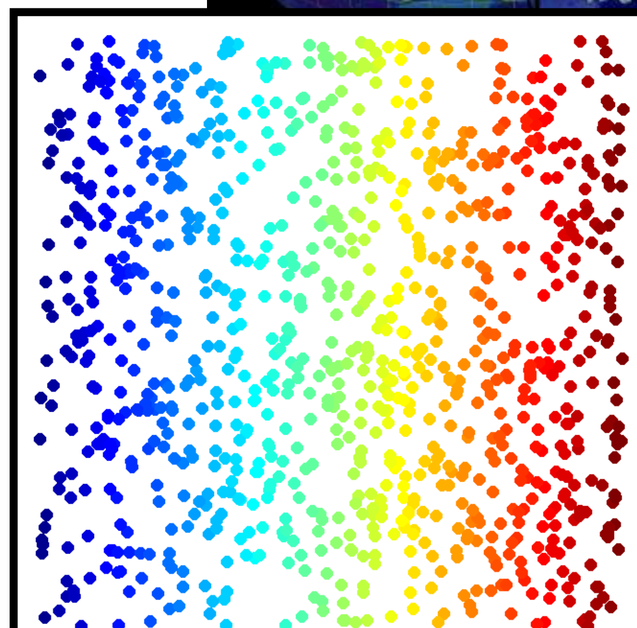
憑藉原來空間中每個點與鄰居之間的關係來做降維

Manifold Learning

高維空間的低維空間







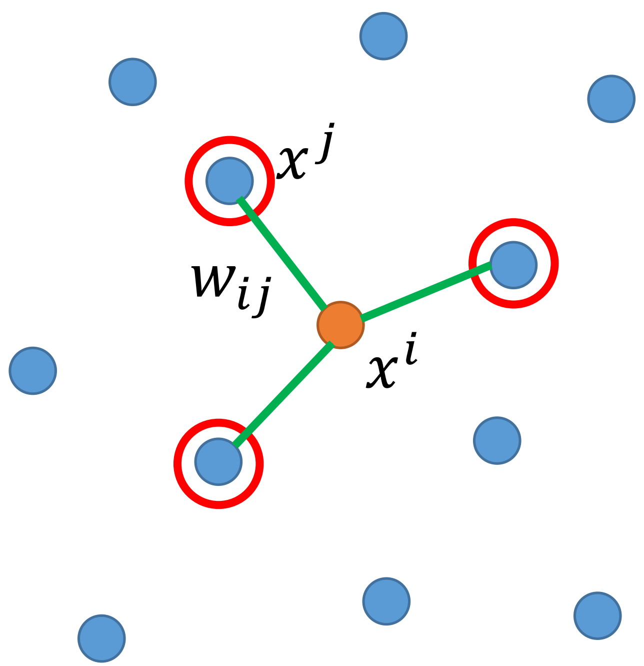
Euclidean Distance結果是錯的

應該要先降維到二維平面在用euclidean

Suitable for clustering or following supervised learning

照理說藍色的點距離黃色較近，然而距離紅色之euclidean distance較小

Locally Linear Embedding (LLE)



w_{ij} represents the relation between x^i and x^j

Find a set of w_{ij} minimizing

利用鄰居linear combination出Xi

$$\sum_i \left\| x^i - \sum_j w_{ij} x^j \right\|_2$$

將xi降維至zi，將xj降維至zj

利用minimize找出wij

Then find the dimension reduction results z^i and z^j based on w_{ij}

先找出wij後固定住，找出zi以及zj

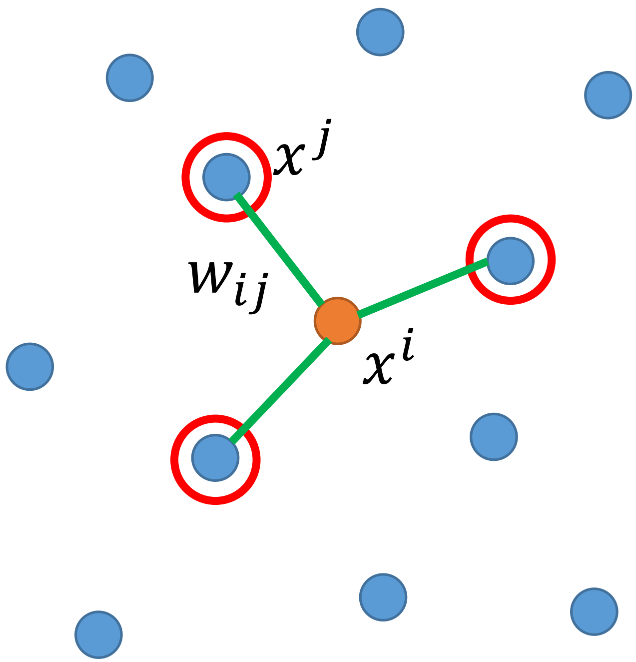
LLE

找出一組zi,zj, 可以利用wij以及zj組合出zi

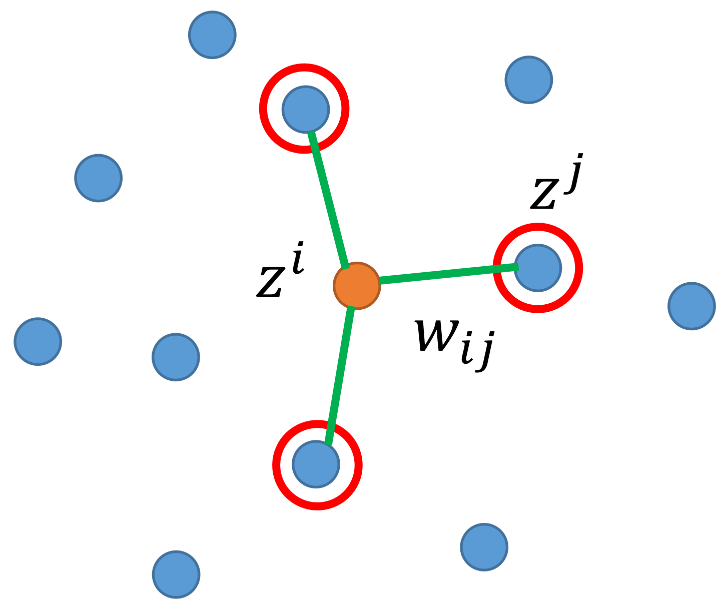
Find a set of z^i minimizing

$$\sum_i \left\| z^i - \sum_j w_{ij} z^j \right\|_2$$

Keep w_{ij} unchanged



Original Space



New (Low-dim) Space

LLE

原來data的domain

z^i, z^j

x^i, x^j

w_{ij}

zi,zj的關係

w_{ij}

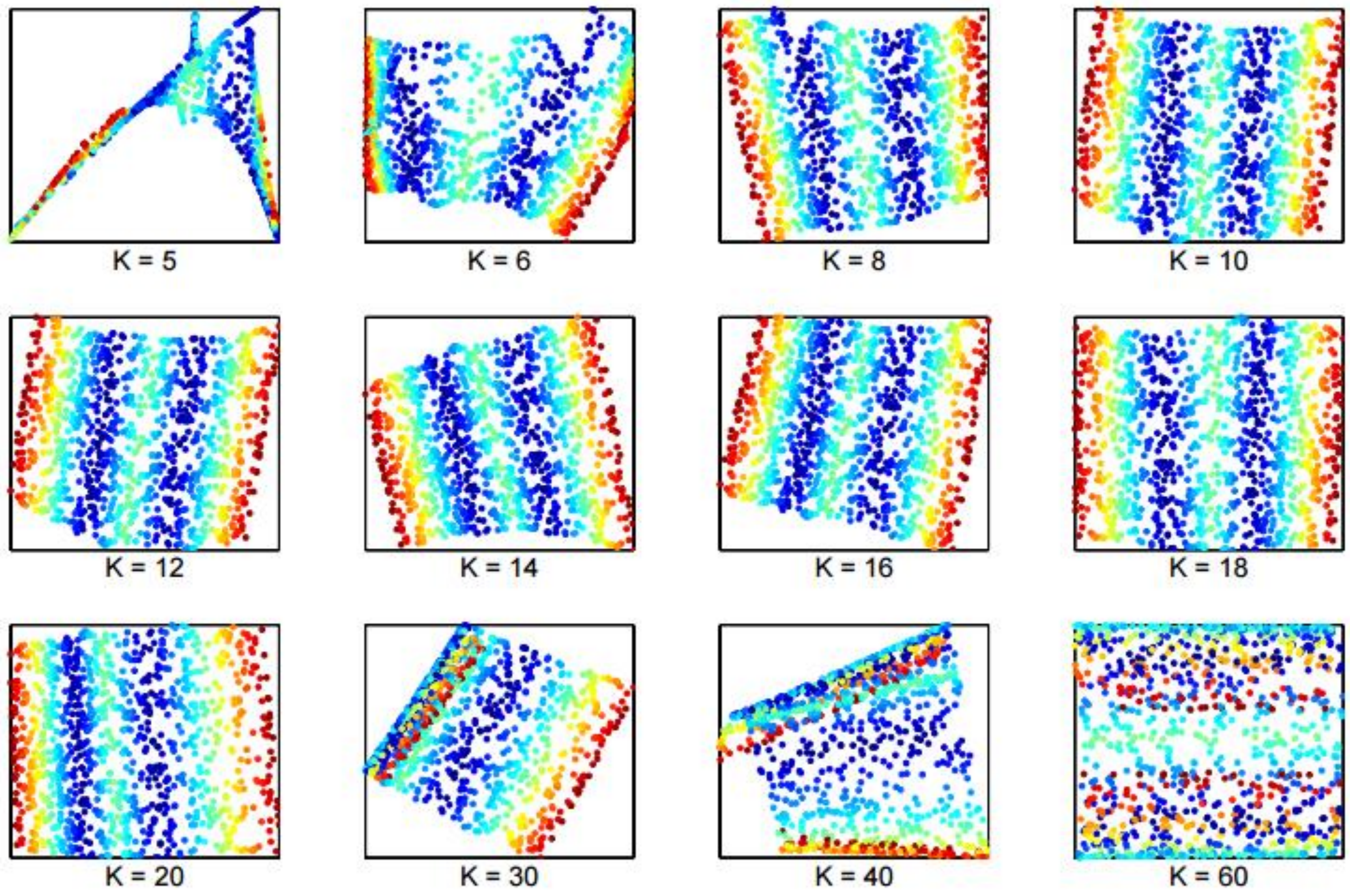
xi,xj的關係

Source of image:
http://feetsprint.blogspot.tw/2016/02/blog-post_29.html

LLE

Lawrence K. Saul, Sam T. Roweis, "Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds", JMLR, 2013

選 K 個鄰居，做manifold learning(展開)



上例為S型的manifold

Laplacian Eigenmaps

- Graph-based approach

衡量兩筆data間的距離用euclidean distance未必是好的量測

用graph中有沒有相連來定義距離做降維

Distance defined by graph approximate the distance on manifold

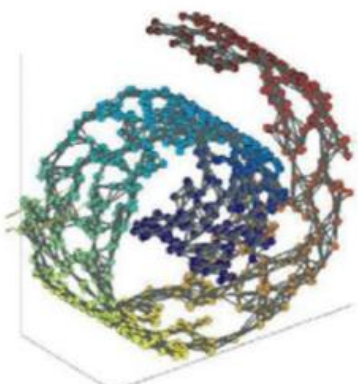
Construct the data points as a ***graph***

Laplacian Eigenmaps

$$w_{i,j} = \begin{cases} \text{similarity} & \text{If connected} \\ 0 & \text{otherwise} \end{cases}$$

semi–supervised

- *Review in semi-supervised learning*: If x^1 and x^2 are close in a high density region, $\hat{y}^1$ and $\hat{y}^2$ are probably the same.



$$L = \sum_{x^r} C\left(y^r, \hat{y}^r\right) \boxed{+\lambda S}$$

As a regularization term

$$S = \frac{1}{2} \sum_{i,j} w_{i,j} \left(y^i - y^j\right)^2 = \boldsymbol{y}^T L \boldsymbol{y}$$

在graph上相連的兩個點越近越好

L: (R+U) x (R+U) matrix

S evaluates how smooth your label is

Graph Laplacian

$$L = D - W$$

unsupervise的情況

Laplacian Eigenmaps

找出zi以及zj，能夠minimize S

- *Dimension Reduction*: If x^1 and x^2 are close in a high density region, z^1 and z^2 are close to each other.

兩個constrain:
1. minimize S
2. Span

$$S = \frac{1}{2}\sum_{i,j} w_{i,j}\left(z^i - z^j\right)^2$$

wij已知，找zi,zj

然而若zi,zj都是0即可minimize

Any problem? How about $z^i = z^j = \mathbf{0}$?

Giving some constraints to z:

因此要加入一些constrain：Span{z1...zn}

If the dim of z is M, $\text{Span}\{z^1, z^2, \dots z^N\} = R^M$

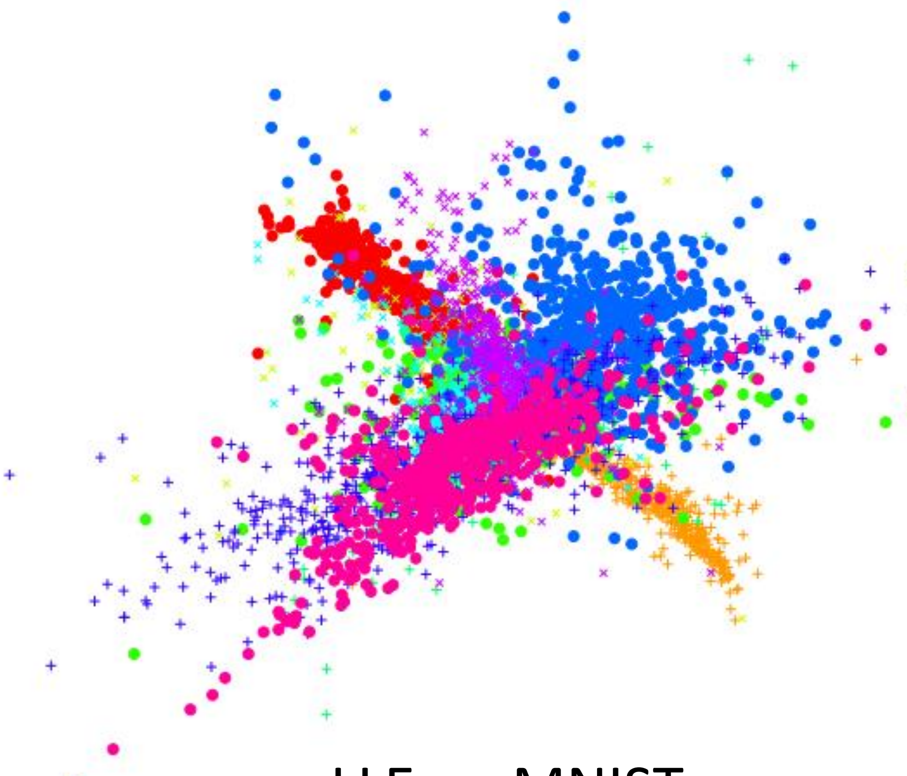
Spectral clustering: clustering on z

Belkin, M., Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems* . 2002

LLE,Laplacian只有說在原來的dimension中相近的點再降維後也要相近，但沒有說原來距離遠的東西再降維後如何
因此原本不像的東西也有可能被擺在一起

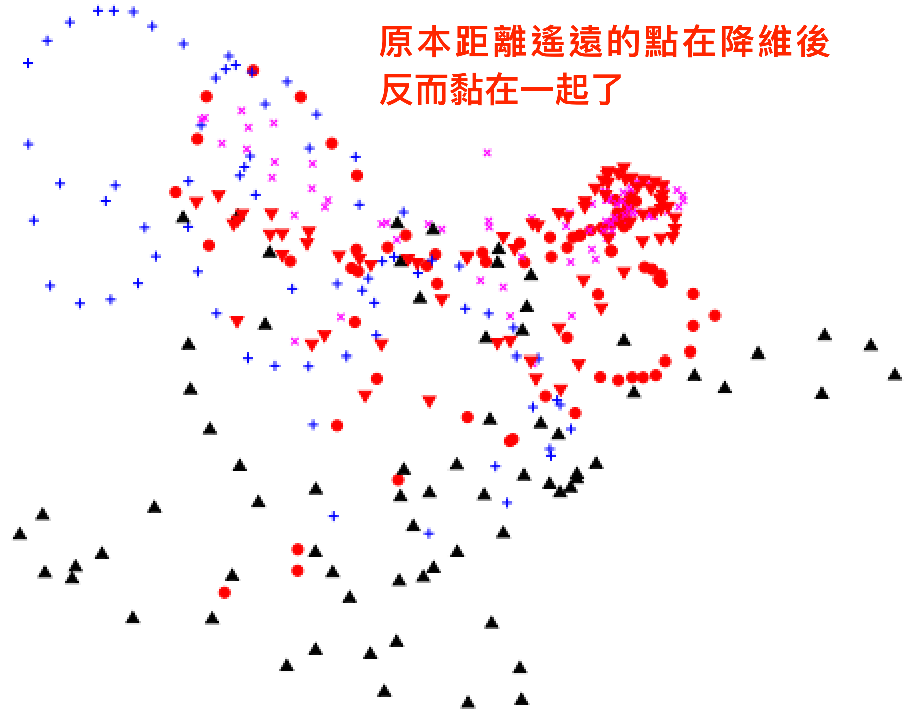
T-distributed Stochastic Neighbor Embedding (t-SNE)

- Problem of the previous approaches
  - Similar data are close, but different data may collapse



LLE on MNIST

原本距離遙遠的點在降維後
反而黏在一起了



LLE on COIL-20

擅長visualization

t-SNE

高維的點做低微的視覺化

x → z

相似度計算可以與降維前不同的evaluation matrix

Compute similarity between all pairs of x: $S\left(x^i, x^j\right)$ 相似度

Compute similarity between all pairs of z: $S'\left(z^i, z^j\right)$

$$P\left(x^j|x^i\right) = \frac{S\left(x^i, x^j\right)}{\sum_{k \neq i} S\left(x^i, x^k\right)}$$

降維前相似度歸依化的分佈

xi與其他data point的相似度

$$Q\left(z^j|z^i\right) = \frac{S'\left(z^i, z^j\right)}{\sum_{k \neq i} S'\left(z^i, z^k\right)}$$

降維後相似度歸依化的分佈

Find a set of z making the two distributions as close as possible

KL divergence 計算降維前後distribution的相似度

$$L = \sum_i KL\left(P\left(* \,|x^i\right) || Q\left(* \,|z^i\right)\right)$$

知道xi，求zi

利用gradient descend更新參數

$$= \sum_i \sum_j P\left(x^j|x^i\right) log \frac{P\left(x^j|x^i\right)}{Q\left(z^j|z^i\right)}$$

由於都做過normalization，因此在算KL不需要擔心不同evaluation之間scale不同的問題

Ignore σ for simplicity

transform以後距離相對關係不變

t-SNE –Similarity Measure

原本的evaluation

$$S(x^i, x^j)$$

$$= exp\left(-\left\|x^i - x^j\right\|_2\right)$$

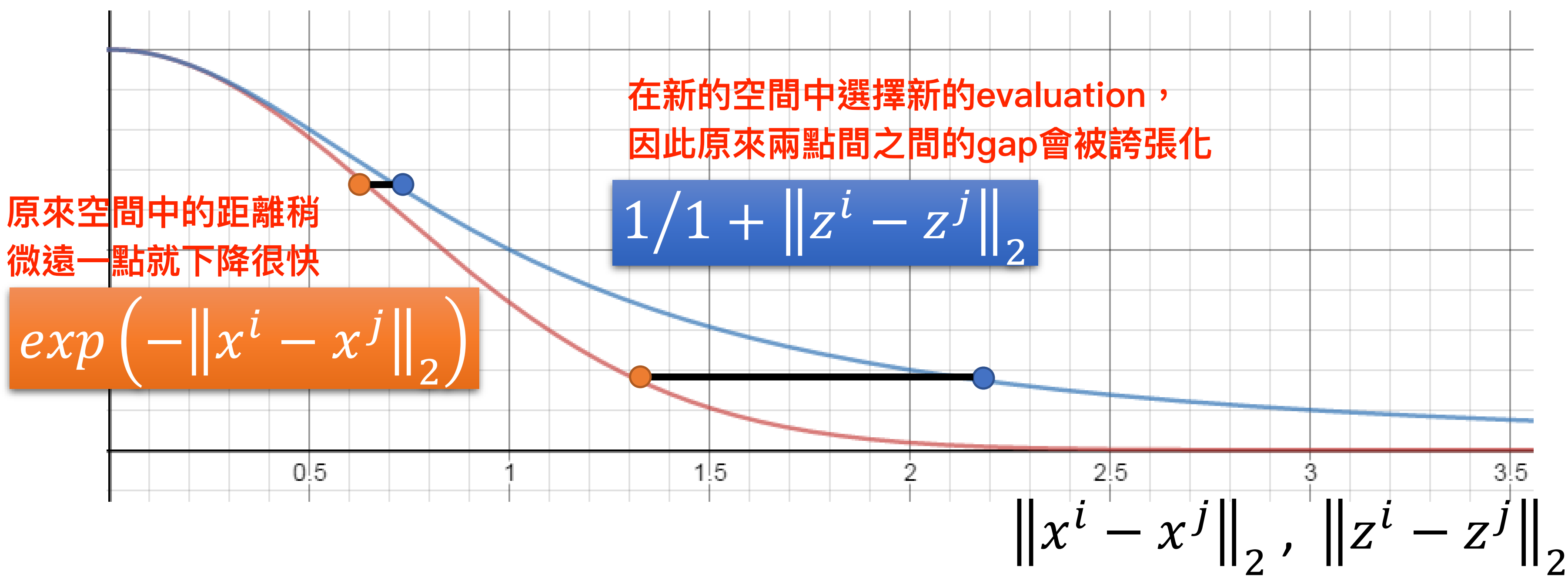
distance下降很快

SNE:

$$S'(z^i, z^j) = exp\left(-\left\|z^i - z^j\right\|_2\right)$$

t-SNE:

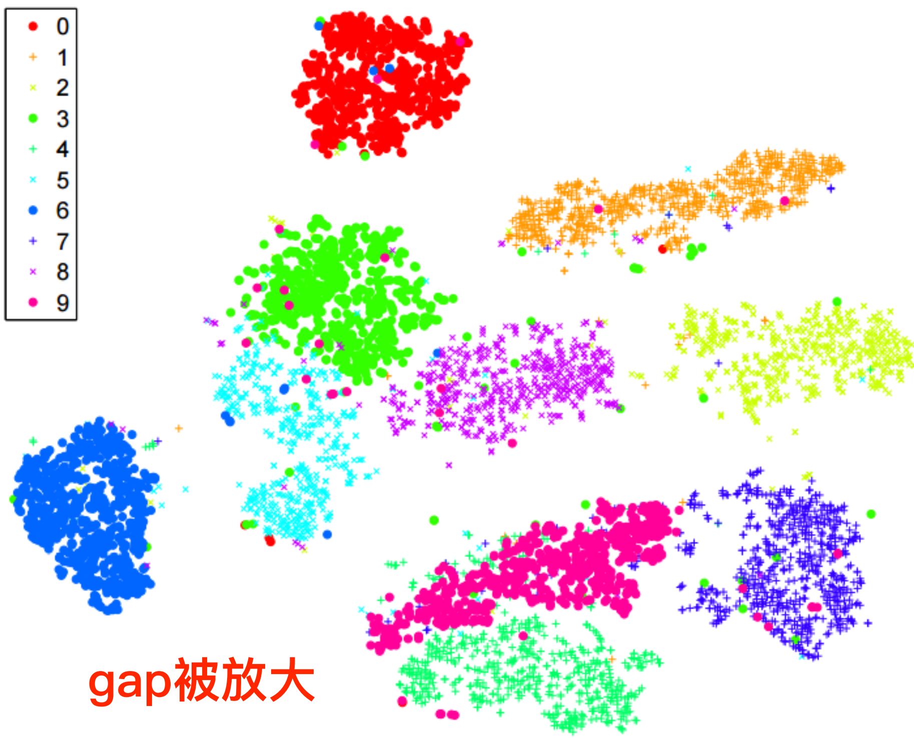
$$S'(z^i, z^j) = 1/1 + \left\|z^i - z^j\right\|_2$$



t-SNE

先做PCA降維後再做dimension reduction

- Good at visualization



gap被放大

t-SNE on MNIST

t-SNE on COIL-20

9

To learn more …

- Locally Linear Embedding (LLE): [Alpaydin, Chapter 6.11]
- Laplacian Eigenmaps: [Alpaydin, Chapter 6.12]
- t-SNE
    - Laurens van der Maaten, Geoffrey Hinton, “Visualizing Data using t-SNE”, JMLR, 2008
    - Excellent tutorial: https://github.com/oreillymedia/t-SNE-tutorial