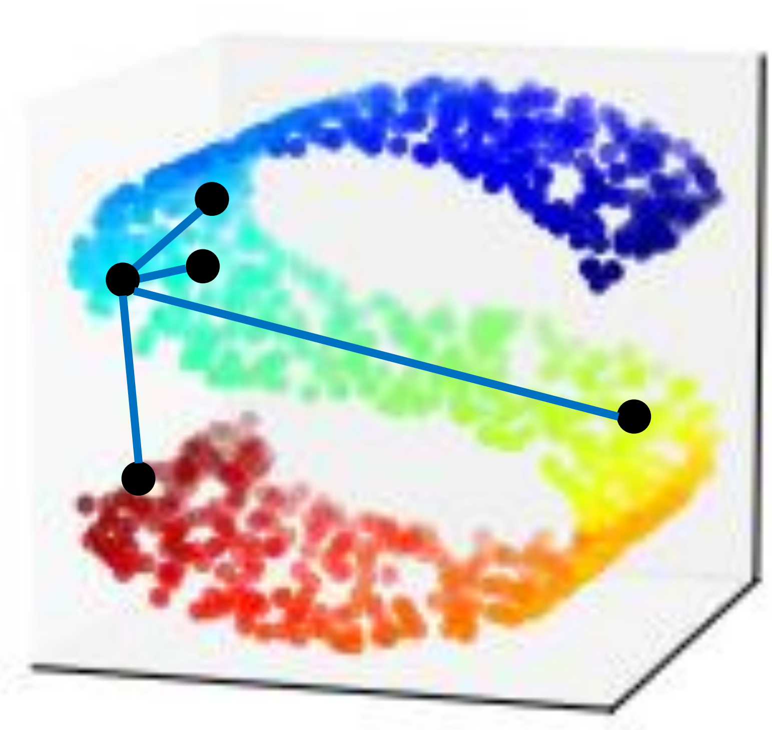
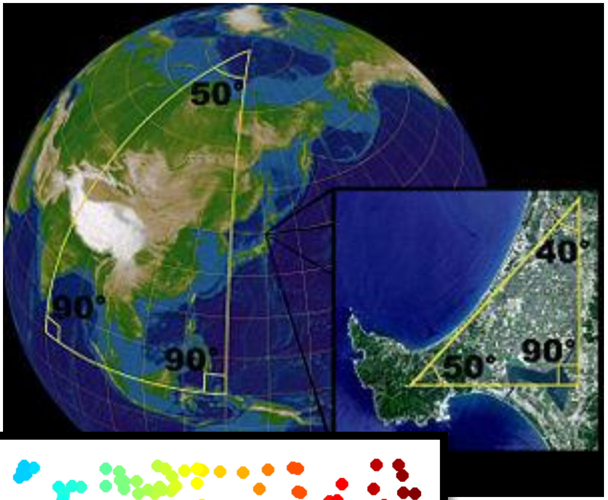
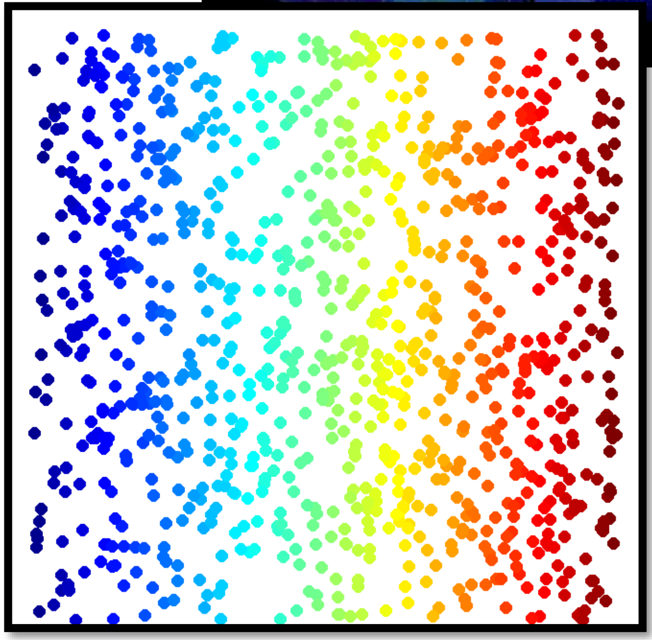
Unsupervised Learning: Neighbor Embedding

Manifold Learning

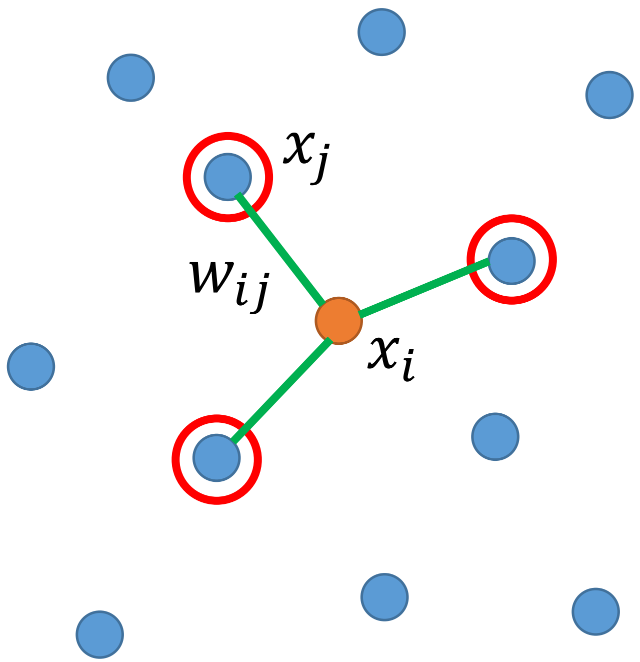


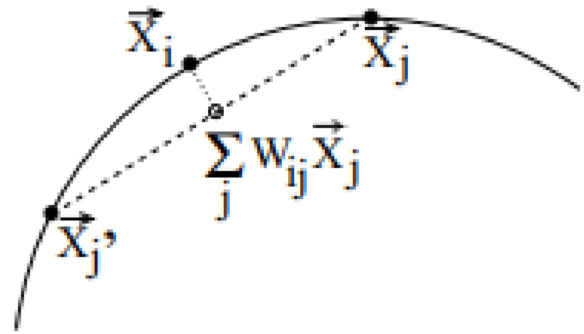




Suitable for clustering or following supervised learning

Locally Linear Embedding (LLE)





Approximates each x_i by linear combination of its K nearest neighbors $\{x_j : j \in \mathcal{N}_i\}$

Find a set of w_{ij} minimizing

$$\sum_i \left\| x_i - \sum_{j \in \mathcal{N}_i} w_{ij} x_j \right\|^2$$

w_{ij} represents the relation between x_i and x_j

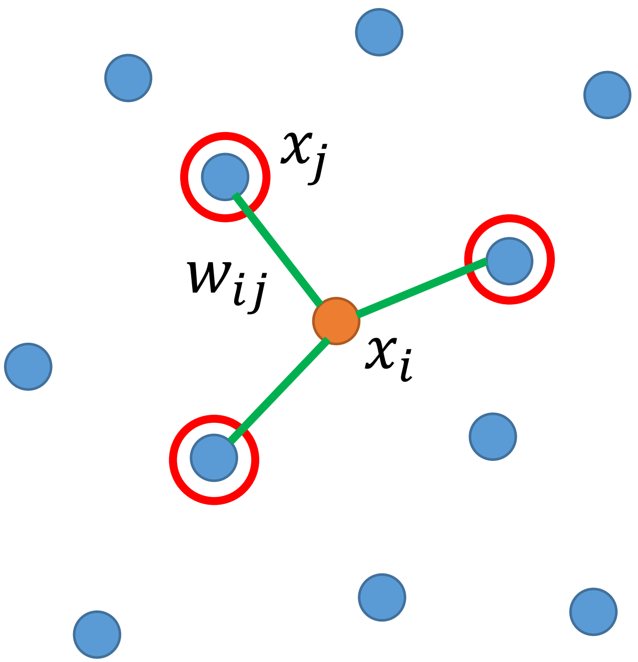
Then find the dimension reduction results z^i and z^j based on w_{ij}

LLE

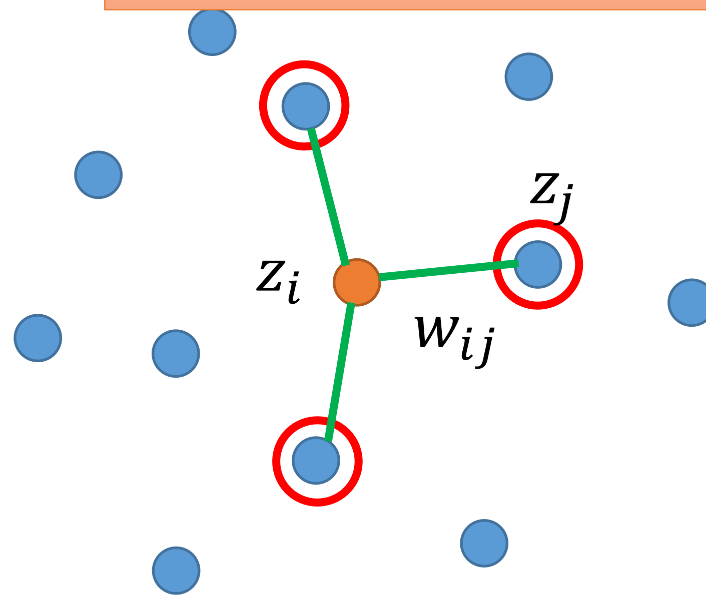
Find a set of z_i minimizing

$$\sum_i \left\| z_i - \sum_{j \in \mathcal{N}_i} w_{ij} z_j \right\|^2$$

Keep w_{ij} unchanged

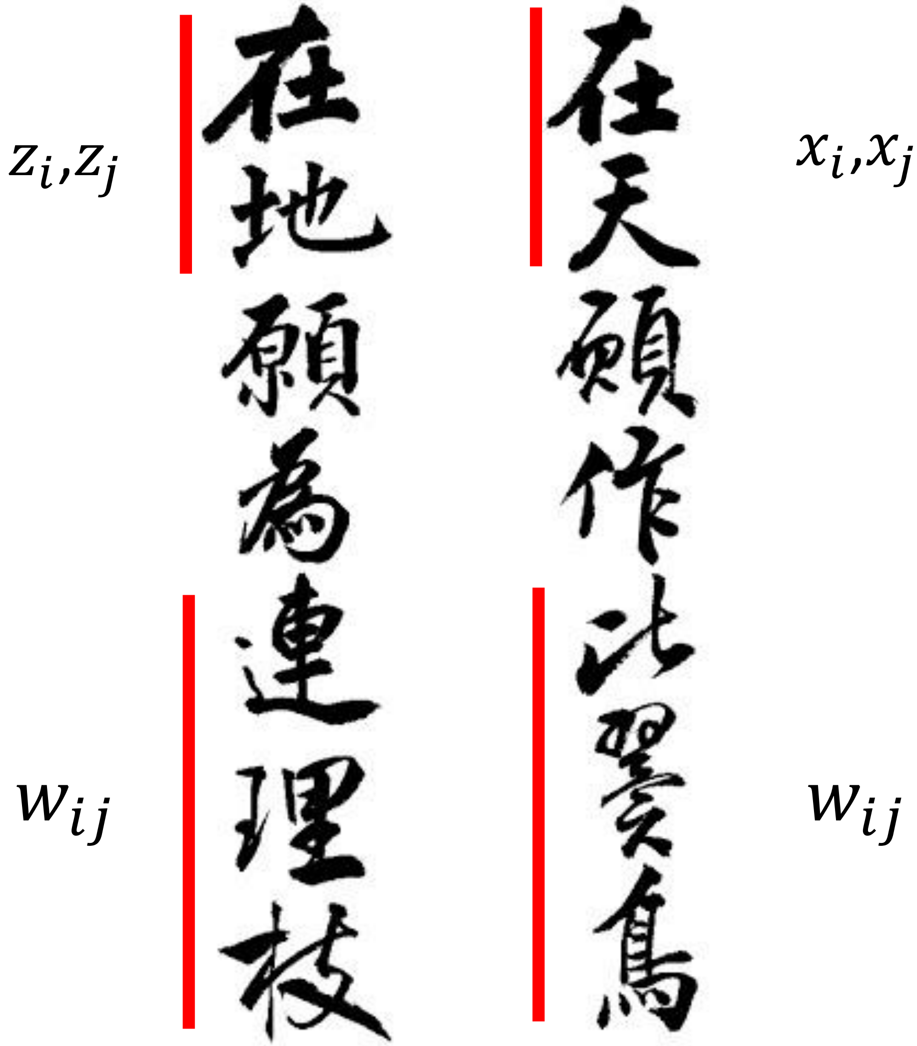


Original Space



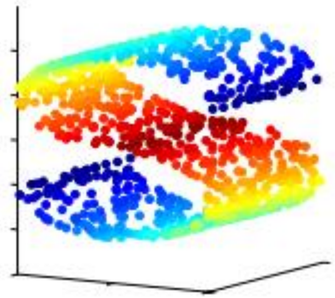
New (Low-dim) Space

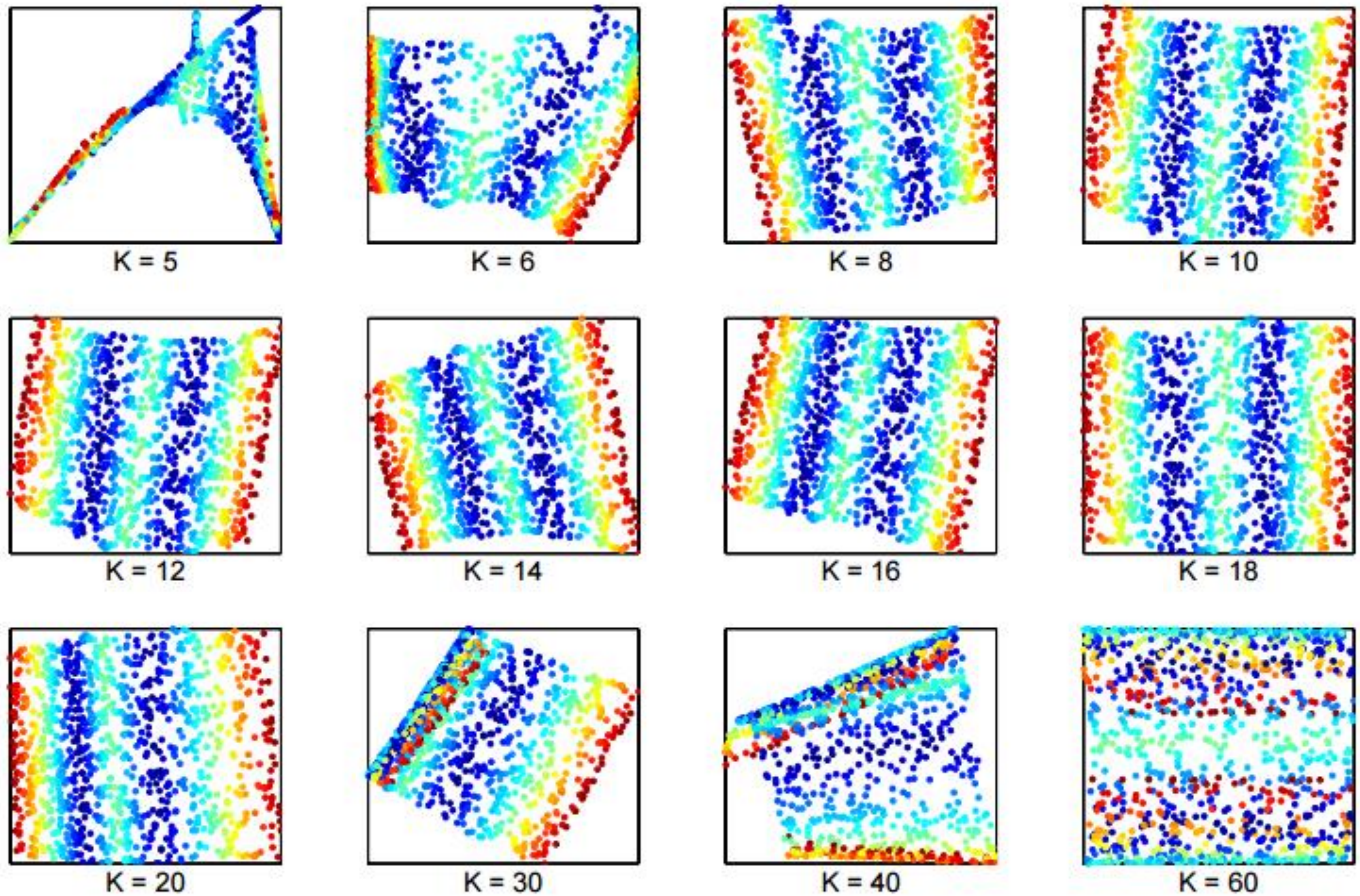
LLE



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, 2003

Laplacian Eigenmaps

- Graph-based approach

Distance on manifold approximated by distance on graph

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

Laplacian Eigenmaps

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

- *Dimension Reduction*: If $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are close in a high density region, $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are close to each other.

$$S = \frac{1}{2}\sum_{i,j} w_{i,j} \left\| \boldsymbol{z}_i - \boldsymbol{z}_j \right\|^2$$

Any problem? How about $\boldsymbol{z}_i = \boldsymbol{z}_j = \boldsymbol{0}$?

Giving some constraints to $\boldsymbol{z}$:

If the dim of $\boldsymbol{z}$ is m, $Span(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots \boldsymbol{z}_N) = \mathbb{R}^m$

Spectral clustering: clustering on $\boldsymbol{z}$

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

Solve Laplacian Eigenmaps

$$S = \frac{1}{2}\sum_{1\le i,j\le N} w_{i,j}\left\|\mathbf{z}_i - \mathbf{z}_j\right\|^2 = \frac{1}{2}\sum_{1\le i,j\le N} w_{i,j}\left(\|\mathbf{z}_i\|^2 - 2\mathbf{z}_i^T\mathbf{z}_j + \left\|\mathbf{z}_j\right\|^2\right)$$

$$= \frac{1}{2}\sum_{1\le i,j\le N} w_{i,j} Trace\left(\mathbf{z}_i\mathbf{z}_i^T - 2\mathbf{z}_j\mathbf{z}_i^T + \mathbf{z}_j\mathbf{z}_j^T\right)$$

$$= Trace\left(\sum_{i=1}^{N}\mathbf{z}_i d_i \mathbf{z}_i^T - \sum_{1\le i,j\le N}\mathbf{z}_j w_{i,j}\mathbf{z}_i^T\right)$$

$$= Trace(\mathbf{\Psi}^T(\mathbf{D} - \mathbf{W})\mathbf{\Psi})$$

L*: *Graph Laplacian

$$d_i = \sum_{j=1}^{N}\frac{w_{i,j} + w_{j,i}}{2}$$

$$\mathbf{D} = diag(d_1, \ldots, d_N)$$

$\mathbf{W} = [w_{i,j}]$（第i行第j列）

$$\mathbf{\Psi} = [\mathbf{z}_1\ \mathbf{z}_2\ \ldots \mathbf{z}_N]^{\mathbf{T}}$$

Optimization problem:

minimize $\mathrm{Trace}(\mathbf{\Psi}^T \mathbf{L}\mathbf{\Psi})$

subject to $\mathbf{\Psi}^T\mathbf{D}\mathbf{\Psi} = \mathbf{I}_m$

variables $\mathbf{\Psi} \in \mathbb{R}^{N\times m}$

$$Span(\mathbf{z}_1, \mathbf{z}_2, \ldots \mathbf{z}_N) = \mathbb{R}^m$$
$$\Leftrightarrow Rank(\mathbf{\Psi}) = m$$

Solve Laplacian Eigenmaps

$$\boldsymbol{\Phi} = \boldsymbol{D}^{1/2}\boldsymbol{\Psi}$$

Optimization problem:

$minimize\ \mathrm{Trace}(\boldsymbol{\Psi}^T \boldsymbol{L} \boldsymbol{\Psi})$
$subject\ to\ \boldsymbol{\Psi}^T \boldsymbol{D} \boldsymbol{\Psi} = \boldsymbol{I}_m$
$variables\ \boldsymbol{\Psi} \in \mathbb{R}^{N\times m}$

Optimization problem:

$minimize\ \mathrm{Trace}(\boldsymbol{\Phi}^T \boldsymbol{D}^{-1/2} \boldsymbol{L} \boldsymbol{D}^{-1/2} \boldsymbol{\Phi})$
$subject\ to\ \boldsymbol{\Phi}^T \boldsymbol{\Phi} = \boldsymbol{I}_m$
$variables\ \boldsymbol{\Phi} \in \mathbb{R}^{N\times m}$

$$\boldsymbol{\Psi}_{opt} = [\boldsymbol{\psi}_N\ \boldsymbol{\psi}_{N-1}\ \dots \boldsymbol{\psi}_{N-m+1}]$$
$$\boldsymbol{\psi}_i = \boldsymbol{D}^{-1/2}\boldsymbol{\phi}_i$$
$$\boldsymbol{D}^{-1}\boldsymbol{L}\boldsymbol{\psi}_i = \lambda_i \boldsymbol{\psi}_i$$

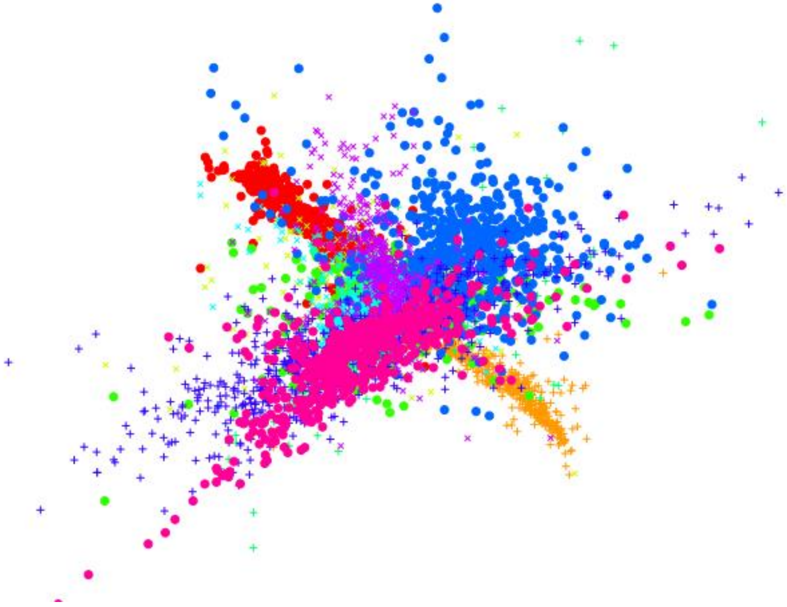
$$\boldsymbol{\Phi}_{opt} = [\boldsymbol{\phi}_N\ \boldsymbol{\phi}_{N-1}\ \dots \boldsymbol{\phi}_{N-m+1}]$$
$$\boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}\boldsymbol{\phi}_i = \lambda_i \boldsymbol{\phi}_i$$

$$\boldsymbol{\Psi} = \boldsymbol{D}^{-1/2}\boldsymbol{\Phi}$$

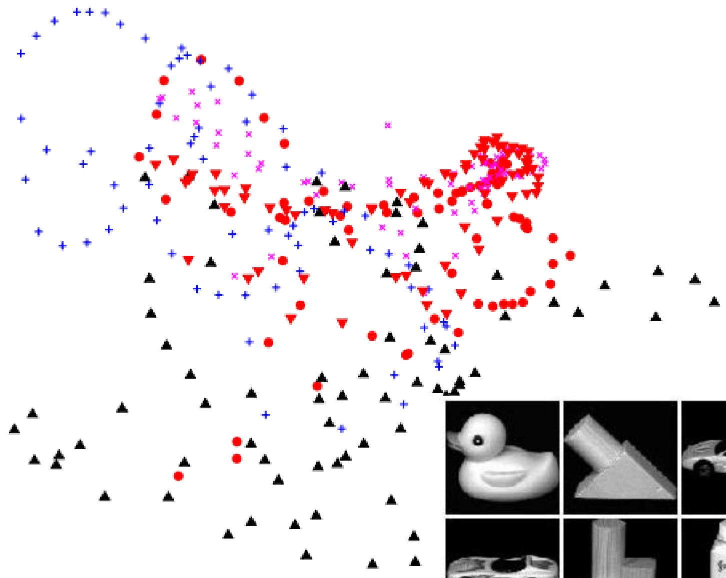
(Eigenvectors with smallest eigenvalues)

T-distributed Stochastic Neighbor Embedding (t-SNE)

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



LLE on MNIST



LLE on COIL-20

SNE

x ⟶ z

Compute similarity between all pairs of x: $S(x_i, x_j)$

Compute similarity between all pairs of z: $S(z_i, z_j)$

$$P(x_j|x_i) = \frac{S(x_i, x_j)}{\sum_{k \neq i} S(x_i, x_k)}$$

$$Q(z_j|z_i) = \frac{S(z_i, z_j)}{\sum_{k \neq i} S(z_i, z_k)}$$

$$S(x_i, x_j) = exp\left(-\left\|x_i - x_j\right\|^2\right)$$

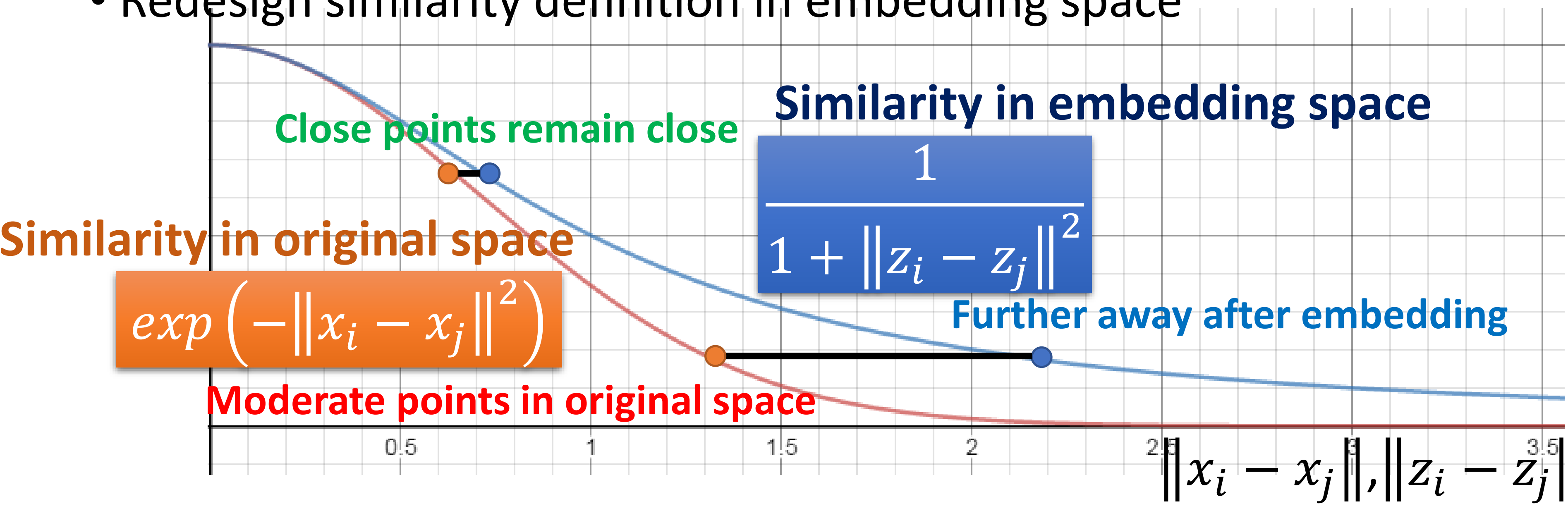
Find a set of z making the two distributions as close as possible

$$L = \sum_i KL\big(P(* \,|x_i)||Q(* \,|z_i)\big) = \sum_i \sum_j P(x_j|x_i) log \frac{P(x_j|x_i)}{Q(z_j|z_i)}$$

Crowding Problem in SNE

- When intrinsic dimension > embedding dimension…

- Solution:
  - ➢ Close points ➔ Remain close
  - ➢ Moderate points ➔ Farther away
- Redesign similarity definition in embedding space



t-SNE

x ⟶ z

Compute similarity between all pairs of x: $S(x_i, x_j)$

Compute similarity between all pairs of z: $S'(z_i, z_j)$

$$P(x_j|x_i) = \frac{S(x_i, x_j)}{\sum_{k \neq i} S(x_i, x_k)}$$

$$Q(z_j|z_i) = \frac{S'(z_i, z_j)}{\sum_{k \neq i} S'(z_i, z_k)}$$

$$S(x_i, x_j) = exp\left(-\left\|x_i - x_j\right\|^2\right)$$

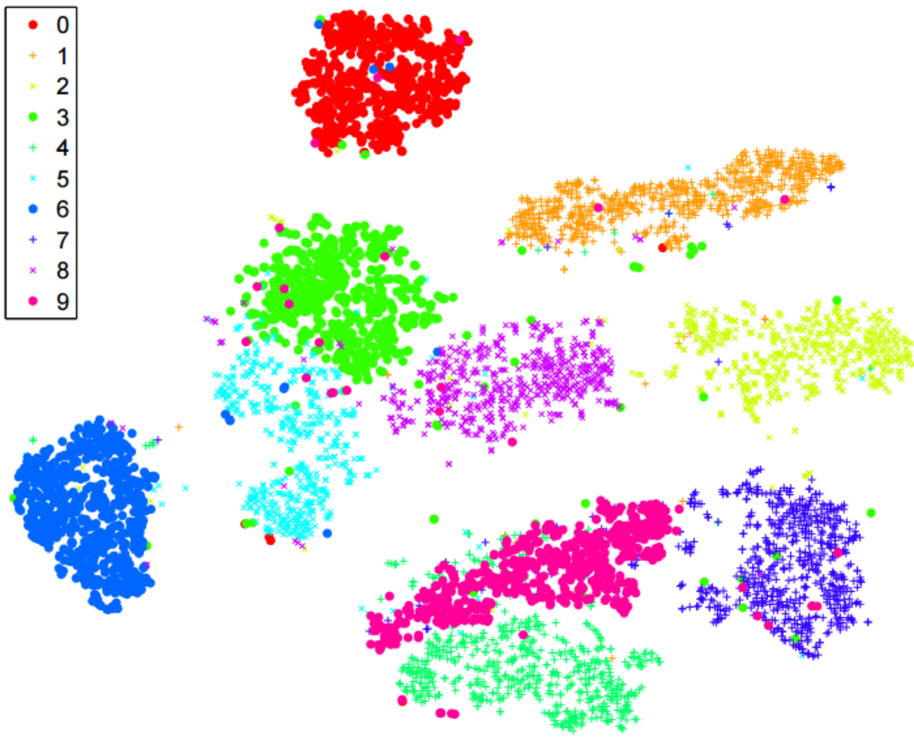
$$S'(z_i, z_j) = \frac{1}{1 + \left\|z_i - z_j\right\|^2}$$

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

$$L = \sum_i KL\big(P(* \,|x_i)||Q(* \,|z_i)\big) = \sum_i \sum_j P(x_j|x_i) log \frac{P(x_j|x_i)}{Q(z_j|z_i)}$$

t-SNE

- Good at visualization



t-SNE on MNIST

t-SNE on COIL-20

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