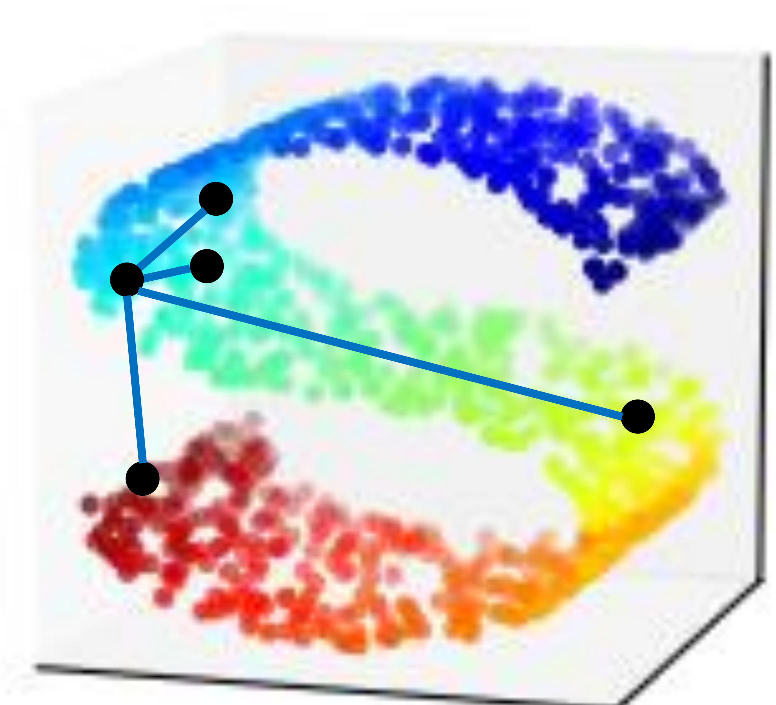
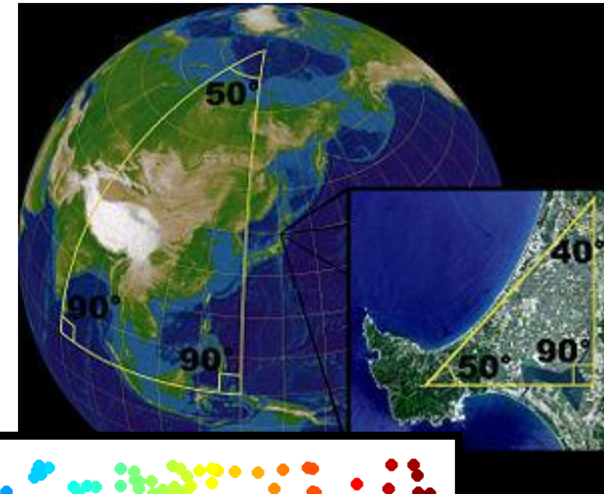
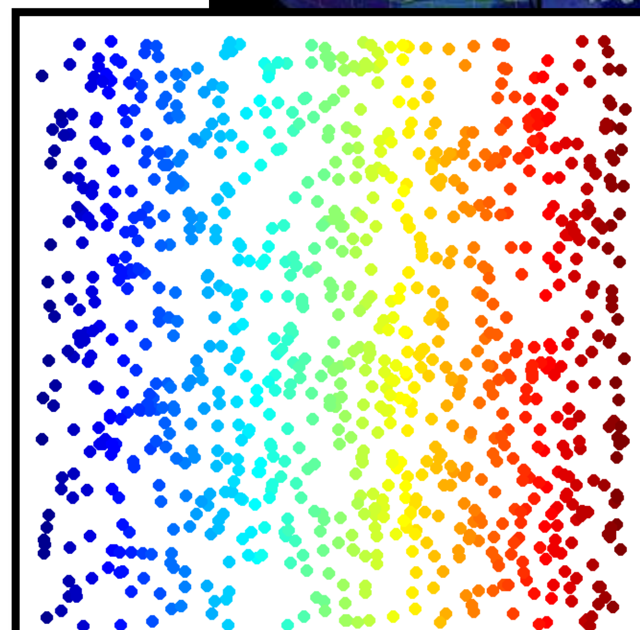
Unsupervised Learning: Neighbor Embedding

Manifold Learning

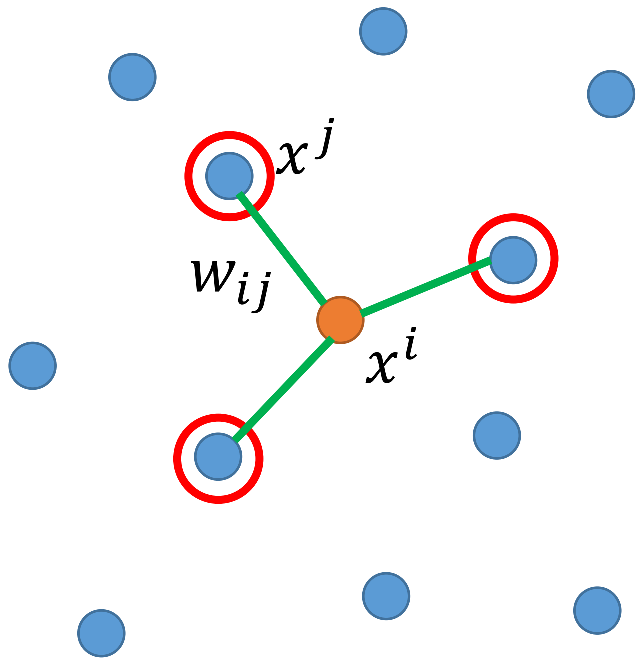






Suitable for clustering or following supervised learning

Locally Linear Embedding (LLE)



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

Find a set of w_{ij} minimizing

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

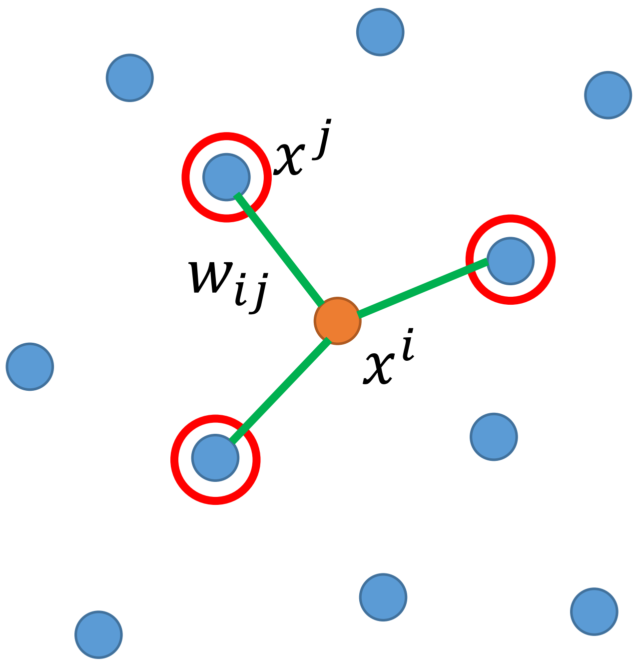
Then find the dimension reduction results z^i and z^j based on w_{ij}

LLE

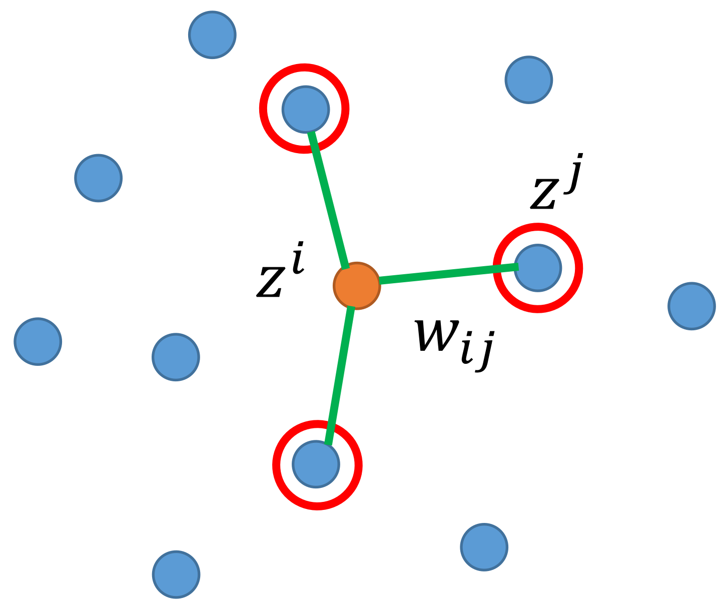
Find a set of z^i minimizing

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

Keep w_{ij} unchanged

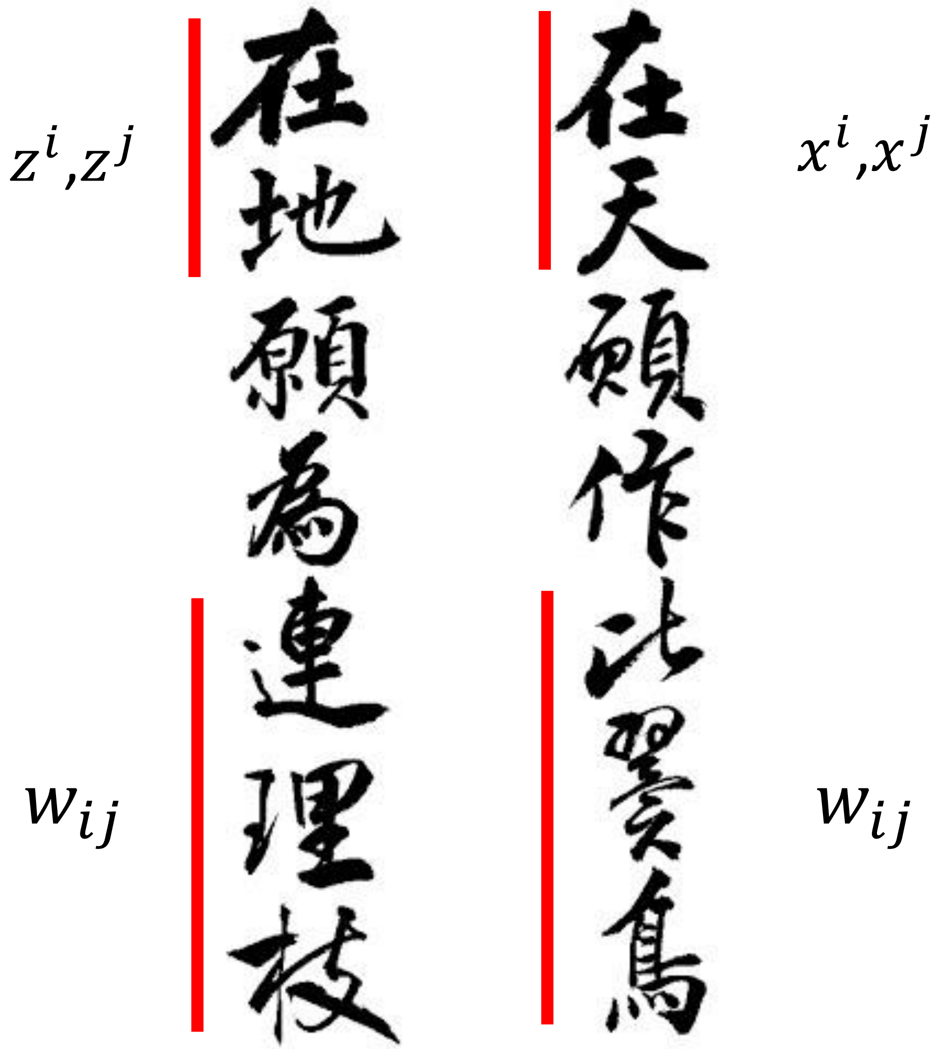


Original Space



New (Low-dim) Space

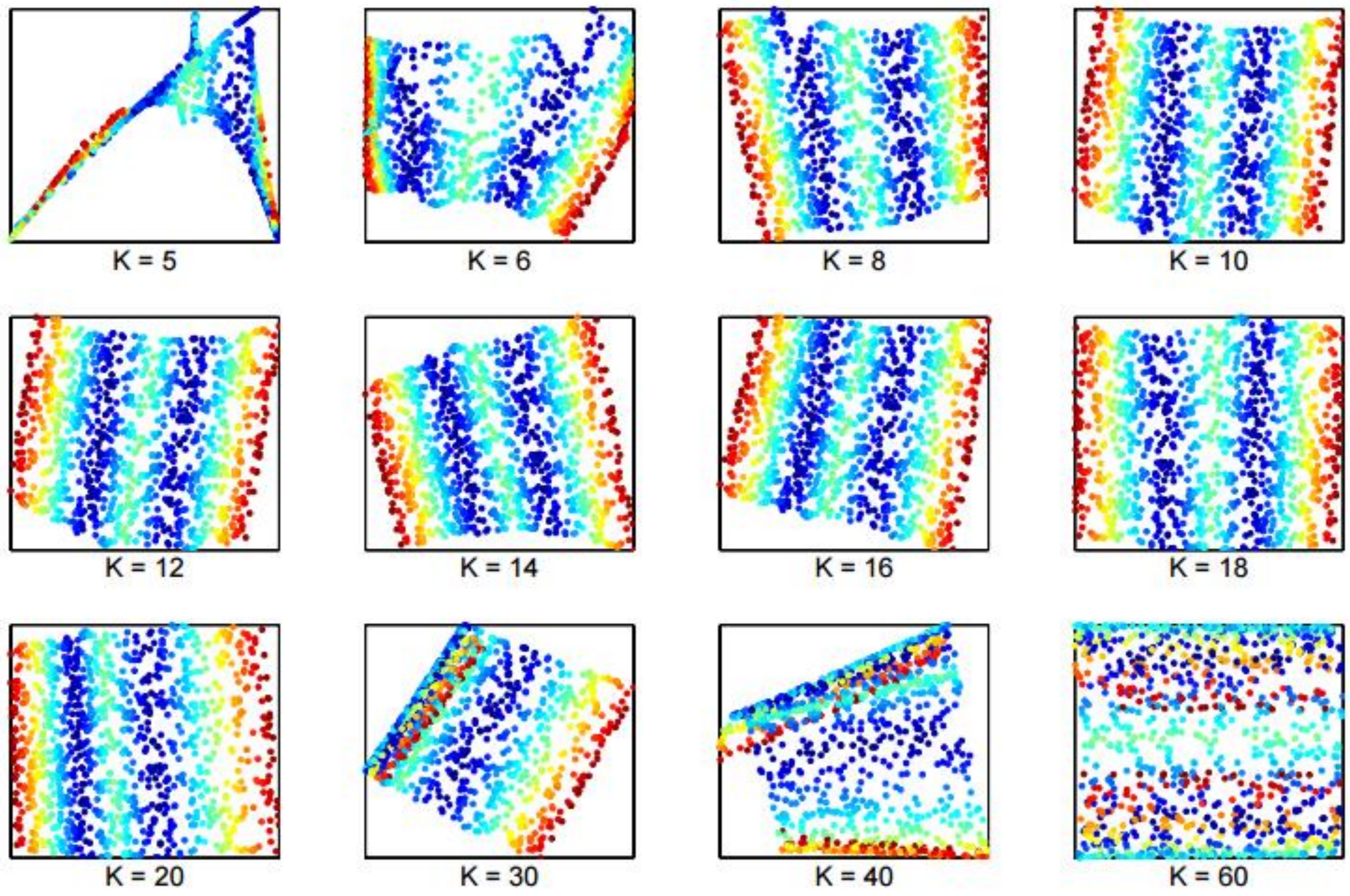
LLE



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, 2013



Laplacian Eigenmaps

- Graph-based approach

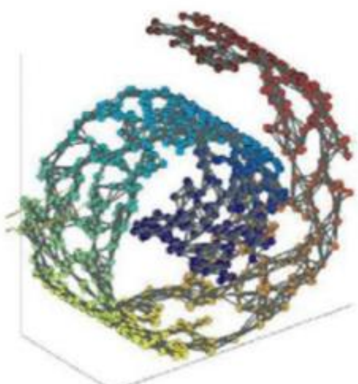
Distance defined by graph approximate the distance on manifold

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

Laplacian Eigenmaps

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

- *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.



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

As a regularization term

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

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

Graph Laplacian

S evaluates how smooth your label is

$$L = D - W$$

Laplacian Eigenmaps

- *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.

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

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

Giving some constraints to z:

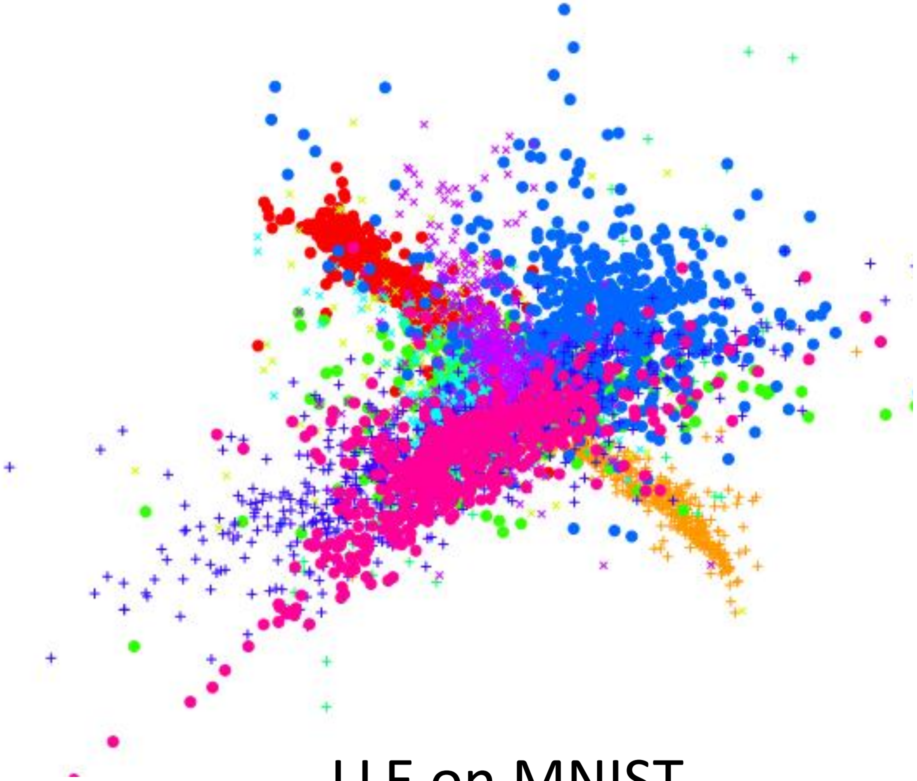
If the dim of z is M, Span{z^1, z^2, … z^N} = R^M

Spectral clustering: clustering on z

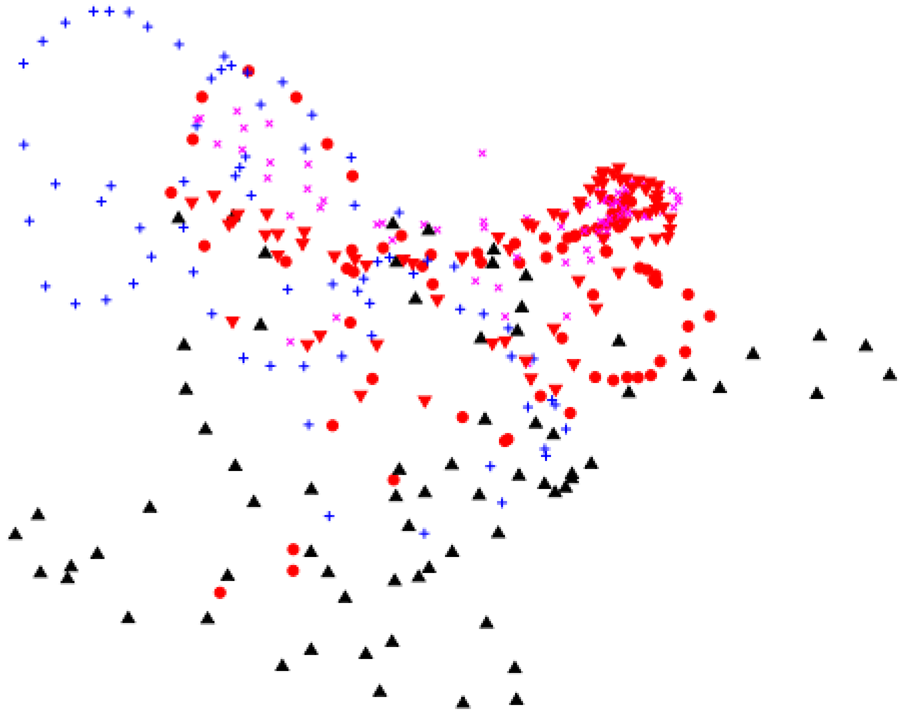
Belkin, M., Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems* . 2002

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

t-SNE

x $\longrightarrow$ z

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

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

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

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

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

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

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

Ignore σ for simplicity

t-SNE –Similarity Measure

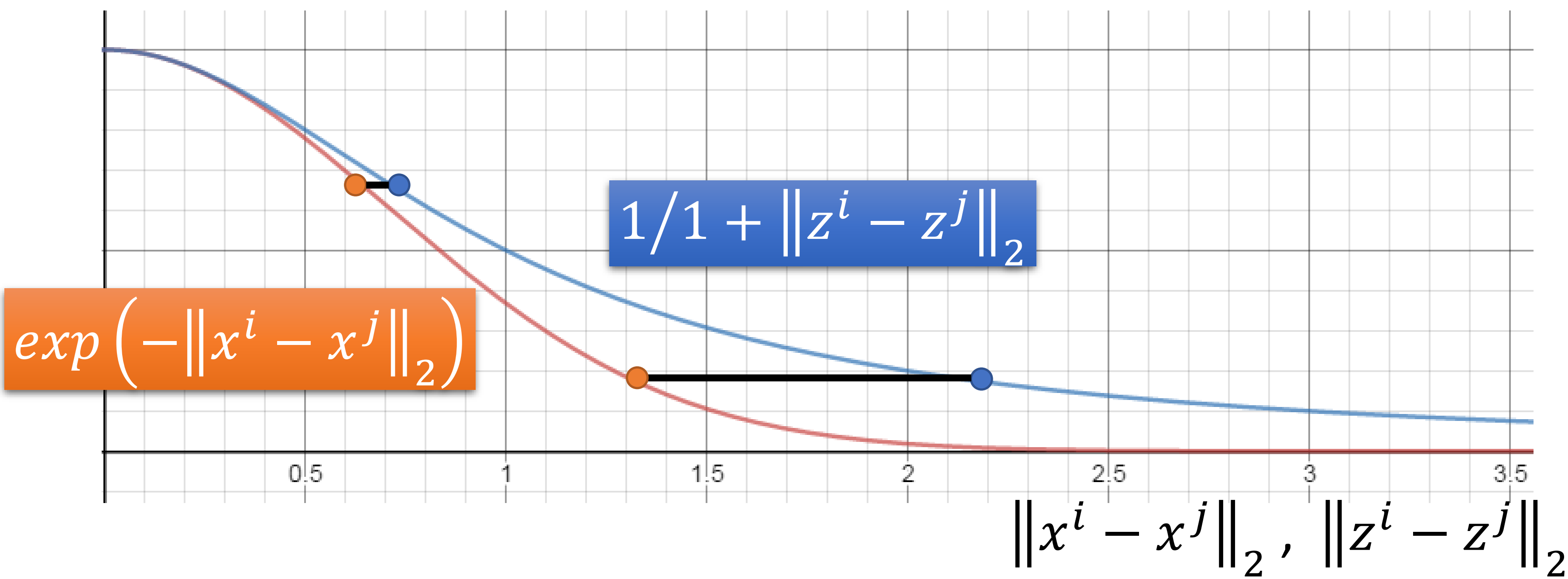
$$S(x^i, x^j)$$
$$= exp\left(-\left\|x^i - x^j\right\|_2\right)$$

SNE:

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

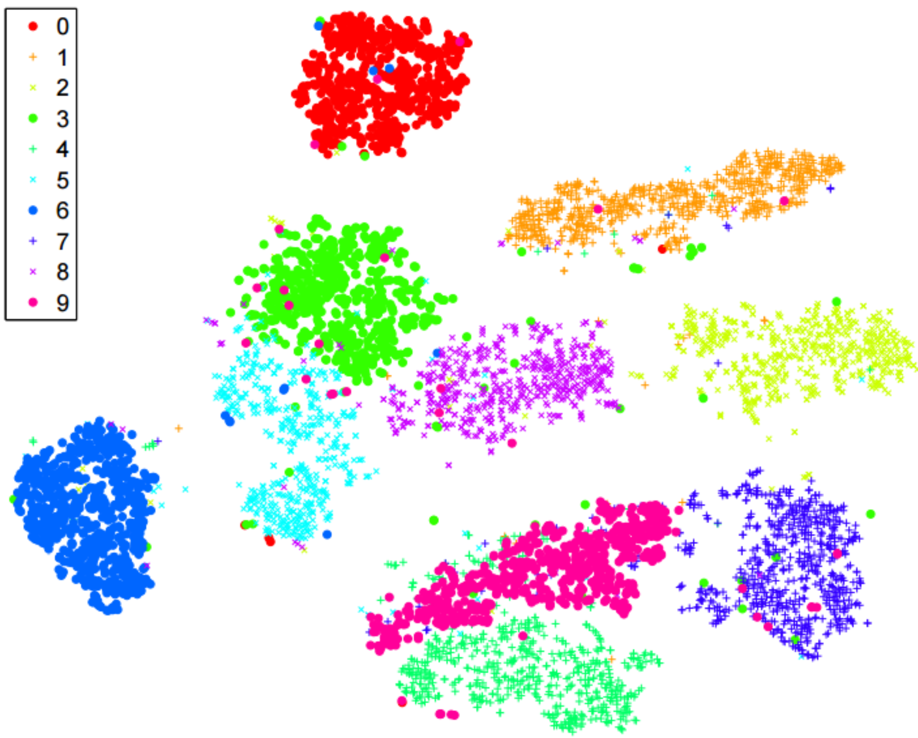
t-SNE:

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



t-SNE

- Good at visualization



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