Dimensionality reduction & clustering

Lecture 17

Changho Suh

October 5, 2021

Outline

t-distributed Stochastic Neighbor Embedding (t-SNE)

1. Emphasize the main role of t-SNE.

2. Investigate the key idea of t-SNE.

3. Study how t-SNE works in detail.

4. Discuss the performance of t-SNE.

Main role of t-SNE

Recall: PCA is a linear technique.

t-SNE: A **non-linear** technique like kernel PCA

Mostly used for *data visualization*

In particular used for *visualizing clusters* of instances

t-SNE in words

A technique that tries to keep:

(i) similar examples close and

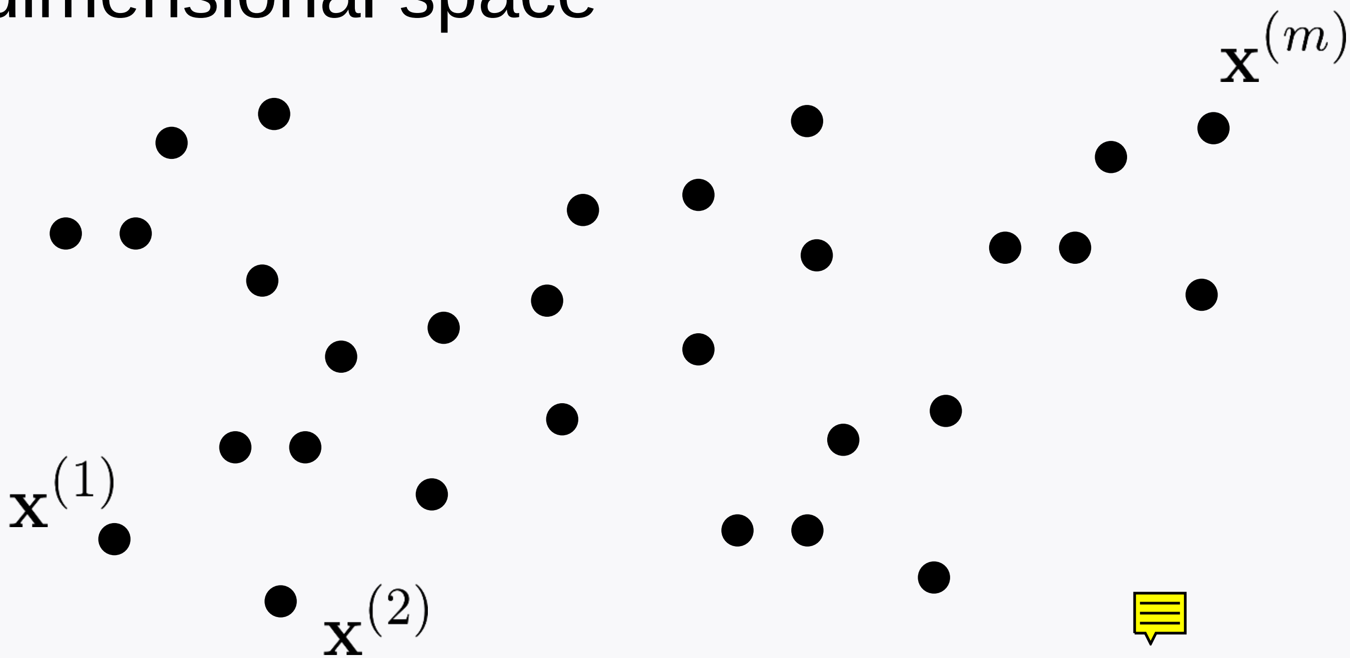
(ii) dissimilar examples apart.

t-SNE in words

original space	**reduced** space
high dimensional space	low dimensional space
similar examples	much closer
dissimilar examples	more far apart

How t-SNE works

high dimensional space

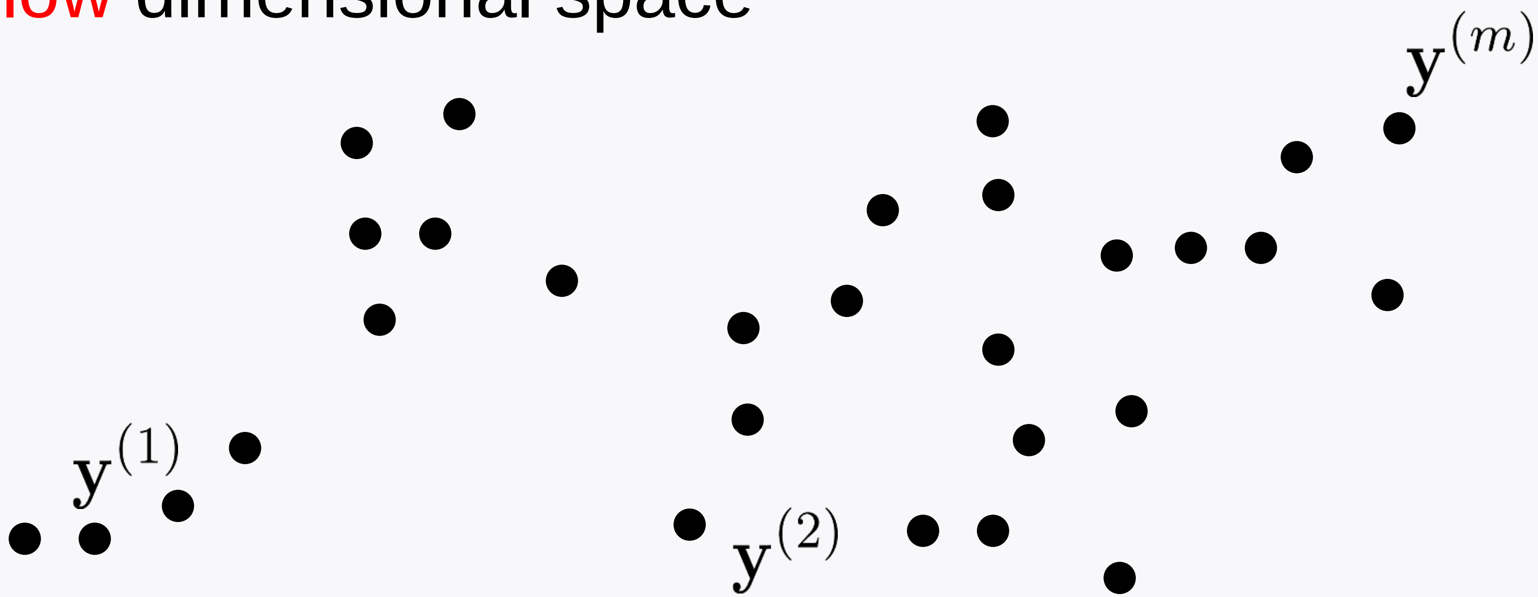


1. Quantify pairwise similarities: $p_{ij} = \dfrac{\exp\left(-\frac{\|\mathbf{x}^{(i)}-\mathbf{x}^{(j)}\|^2}{2\sigma^2}\right)}{\sum_{k\neq\ell}\exp\left(-\frac{\|\mathbf{x}^{(k)}-\mathbf{x}^{(\ell)}\|^2}{2\sigma^2}\right)}$

Note: Can be viewed as probability distribution.

How t-SNE works

low dimensional space



2. Define pairwise similarities: $$q_{ij} = \frac{\left(1 + \|\mathbf{y}^{(i)} - \mathbf{y}^{(j)}\|^2\right)^{-1}}{\sum_{k \neq \ell} \left(1 + \|\mathbf{y}^{(k)} - \mathbf{y}^{(\ell)}\|^2\right)^{-1}}$$

Note: Can also be viewed as probability distribution

Based on a student t-distribution

How t-SNE works

$$p_{ij} = \frac{\exp\left(-\frac{\|\mathbf{x}^{(i)}-\mathbf{x}^{(j)}\|^2}{2\sigma^2}\right)}{\sum_{k\neq\ell}\exp\left(-\frac{\|\mathbf{x}^{(k)}-\mathbf{x}^{(\ell)}\|^2}{2\sigma^2}\right)} \qquad q_{ij} = \frac{\left(1+\|\mathbf{y}^{(i)}-\mathbf{y}^{(j)}\|^2\right)^{-1}}{\sum_{k\neq\ell}\left(1+\|\mathbf{y}^{(k)}-\mathbf{y}^{(\ell)}\|^2\right)^{-1}}$$

3. Find $\{\mathbf{y}^{(i)}\}_{i=1}^m$ such that the two distributions are as similar as much possible:

$$\min_{\{\mathbf{y}^{(i)}\}_{i=1}^m} \mathsf{KL}(p_{ij}\|q_{ij})$$

Kullaback-Leibler divergence
(very similar to cross entropy)

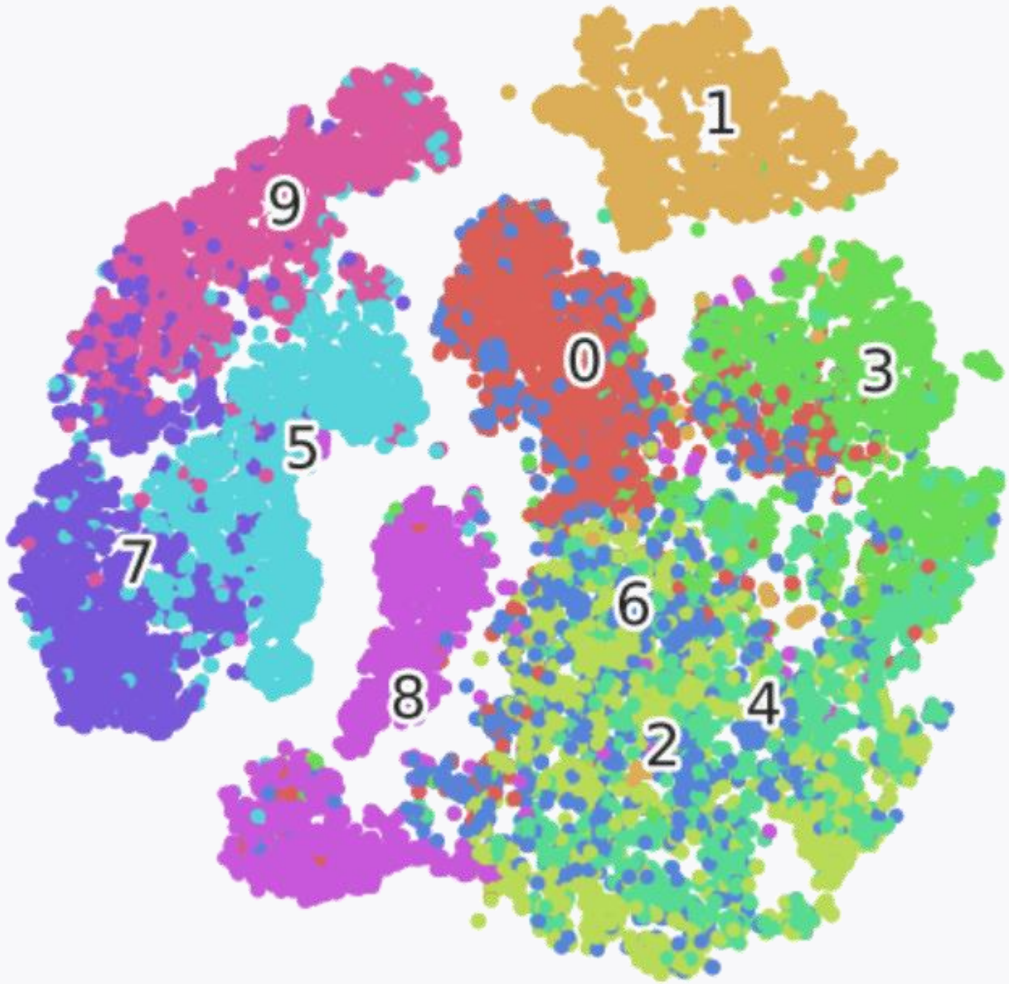
How to solve the optimization?

$$p_{ij} = \frac{\exp\left(-\frac{\|\mathbf{x}^{(i)}-\mathbf{x}^{(j)}\|^2}{2\sigma^2}\right)}{\sum_{k\neq\ell}\exp\left(-\frac{\|\mathbf{x}^{(k)}-\mathbf{x}^{(\ell)}\|^2}{2\sigma^2}\right)} \qquad q_{ij} = \frac{\left(1+\|\mathbf{y}^{(i)}-\mathbf{y}^{(j)}\|^2\right)^{-1}}{\sum_{k\neq\ell}\left(1+\|\mathbf{y}^{(k)}-\mathbf{y}^{(\ell)}\|^2\right)^{-1}}$$

$$\min_{\{\mathbf{y}^{(i)}\}_{i=1}^m} \underbrace{\mathsf{KL}(p_{ij}\|q_{ij})}_{\text{a complicated function of } \{\mathbf{y}^{(i)}\}_{i=1}^m}$$

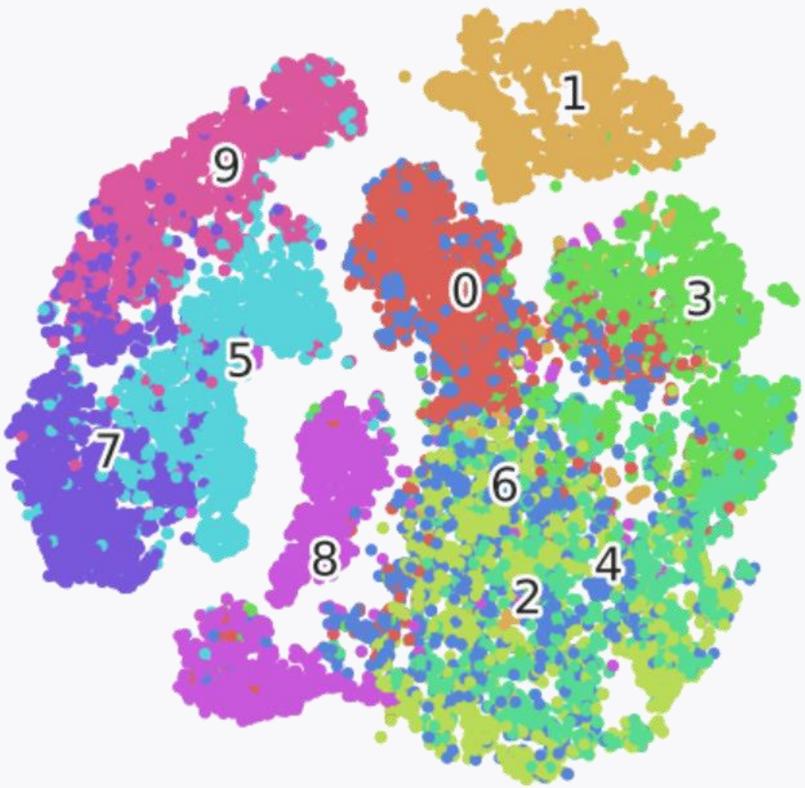
Idea: Just apply **gradient descent**.

MNIST example

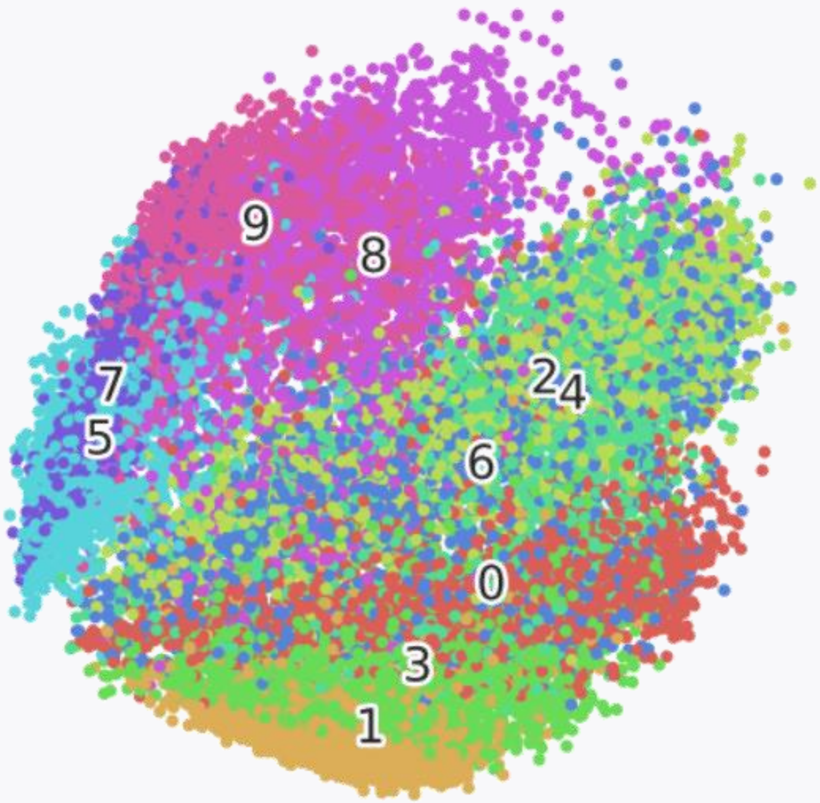


t-SNE vs PCA

t-SNE



PCA



Limitations of t-SNE

1. Specialized technique only for $d = 2, 3$

   Not clear as to how to generalize to arbitrary *d*.

2. Lose data structure significantly during projection.

3. Training instability:

   Different seeds → different results.

Hence: it is often preferred to use PCA or kernel PCA instead.

Look ahead

Will study clustering methods:

1. K-means

2. K-medoids

3. hierarchical clustering (agglomerative clustering)