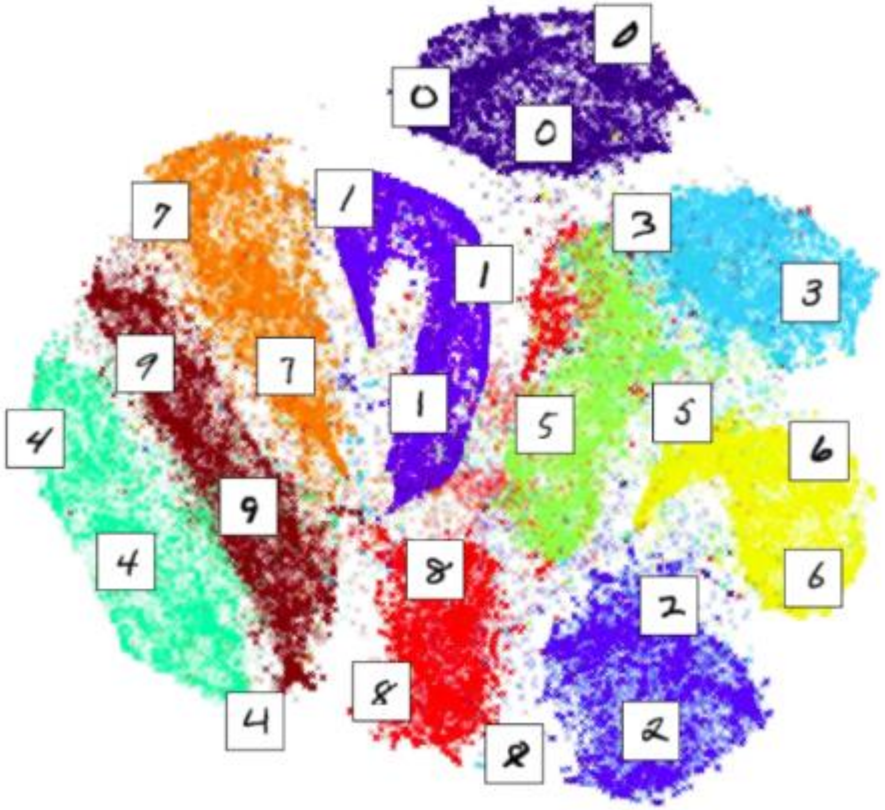


Dimensionality Reduction

t-SNE

Maaten and Hinton (2008)

DSBA 강필성 교수님 강의 참고

작성자: 구병모

로컬 선형 임베딩(Local Linear Embedding) 이란?

정의: 고차원의 공간에서 인접해 있는 데이터들 사이의 선형적 구조를 보존하면서 저차원으로 임베딩하는 방법론

장점

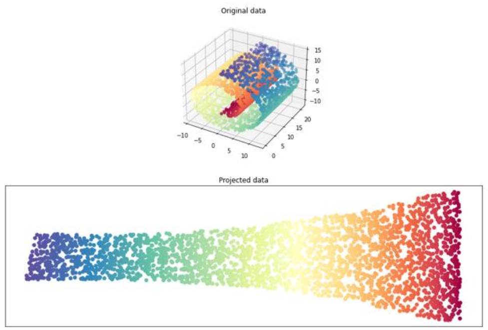
1. 사용하기 간단하다

2. 최적화가 국소최소점으로 가지 않는다

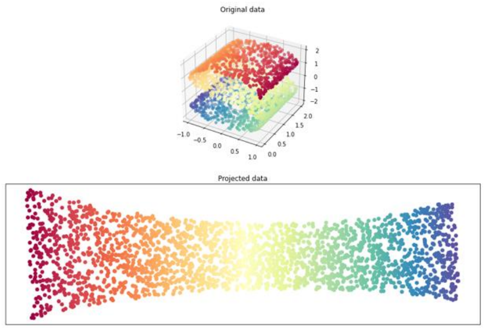
3. 비선형 임베딩이 가능하다

4. 고차원의 데이터를 저차원의 데이터로 매핑 가능하다

✓ Swiss Roll



✓ S-Curve



LLE 알고리즘

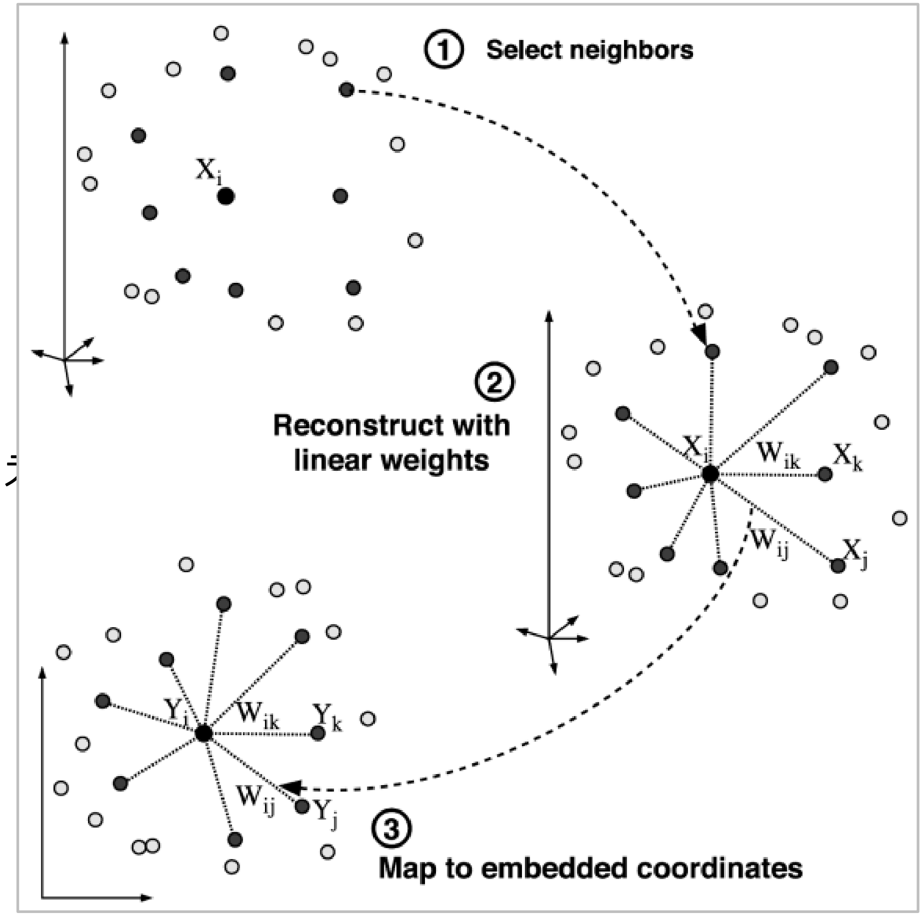
Step 1. 각 데이터 포인트 점에서 k개의 이웃을 구한다. ← K-nearest neighbor 방법

Step 2. 현재의 데이터를 나머지 k개의 데이터의 가중치의 합을 뺄 때 최소가 되는 가중치 매트릭스를 구한다.

$$E(\mathbf{W}) = \sum_i \left| \mathbf{x}_i - \sum_j \mathbf{W}_{ij}\mathbf{x}_j \right|^2$$

$$s.t.\ \mathbf{W}_{ij} = 0 \text{ if } \mathbf{x}_j \text{ does not belong to the neighbor of } \mathbf{x}_i$$

$$\sum_j \mathbf{W}_{ij} = 1 \text{ for all } i$$



Step 3. 앞서 구한 가중치를 최대한 보장하며 차원을 축소합니다.
이때 차원 축소된 후의 점을 Y로 표현하며 차원 축소된 Y_j 와의 값

$$\Phi(\mathbf{W}) = \sum_i \left| \mathbf{y}_i - \sum_j \mathbf{W}_{ij}\mathbf{y}_j \right|^2$$

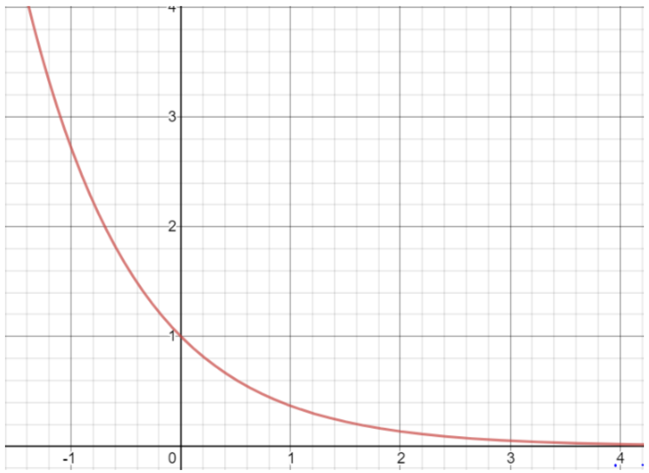
Stochastic Neighbor Embedding (SNE)

LLE와의 관계

- LLE(로컬 선형 임베딩)과 유사하게 이웃이 되는 인스턴스를 기준으로 Embedding
- LLE는 이산적(deterministic) / SNE는 확률적(Stochastic)
- LLE는 k개의 인스턴스(이웃) 확정한 뒤 다른 인스턴스 고려 X / SNE는 전부 고려 BUT 거리에 따라 확률 부여
- 즉 확률적으로 지역성을 결정

SNE 특징

$$p_{j|i} = \frac{e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma_i^2}}}{\sum_{k\neq i} e^{-\frac{||\mathbf{x}_i - \mathbf{x}_k||^2}{2\sigma_i^2}}} \qquad q_{j|i} = \frac{e^{-||\mathbf{y}_i - \mathbf{y}_j||^2}}{\sum_{k\neq i} e^{-||\mathbf{y}_i - \mathbf{y}_k||^2}}$$



- p는 원 데이터의 차원에서 인스턴스 i가 인스턴스 j를 이웃으로 선택할 확률
- q는 축소된 데이터의 차원에서 인스턴스 i가 인스턴스 j를 이웃으로 선택할 확률
- 거리가 멀어질수록 낮은 확률 값 할당하기 위해서 → $y = e^x$ 사용
- p와 q는 비슷하게 생김 BUT 다른 점 有 → p 구하는 식에는 σ 존재
- σ = 거리에 따른 확률 값 차이를 얼마나 줄 것인지를 조정
- σ↑ : 멀리 있는 인스턴스의 확률 상대적으로 큼 / σ↓ : 멀리 있는 인스턴스의 확률 상대적으로 작음
- $Perplexity(P_i) = 2^{H(P_i)}$ $H(P_i) = \sum_j p_{j|i} \log_2 p_{j|i}$ → radius(σ) 값을 적절하게 선택해서 엔트로피 값 결정

➔ 실제로는 σ값에 거의 상관없이 강건(Robust), default 값 쓰면 됨

Cost Function

$$Cost = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} log \frac{p_{j|i}}{q_{j|i}}$$

- 목표: 두 확률 분포 p와 q 동일하게 만들기
- 목표 달성 위해 위 식 Kullback-Leibler divergence (KLD) 사용해서 두 값 비교 → 두 값의 엔트로피를 비교
- 두 확률 분포 p와 q 완전히 동일해진다면 Cost Function은 0
- 방법: Cost Function을 최소로 만드는 y값 구하기 → Gradient Descent (굉장히 복잡 BUT 결과는 간단)
- 결과: $\frac{\partial C}{\partial \mathbf{y}_i} = 2 \sum_j (\mathbf{y}_j - \mathbf{y}_i)(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})$

Symmetric SNE

- SNE에서는 $p_{i|j} \neq p_{j|i}$

$$p_{ij} = \frac{e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma_i^2}}}{\sum_{k \neq l} e^{-\frac{||\mathbf{x}_k - \mathbf{x}_l||^2}{2\sigma_i^2}}} \qquad p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \qquad \sum_j p_{ij} > \frac{1}{2n}$$

- Symmetric SNE에서는 $p_{i|j} = p_{j|i}$ → 조건부 확률을 pairwise 하게 만들어줌

$$Cost = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}}$$

$$\frac{\partial C}{\partial \mathbf{y}_i} = 4 \sum_j (\mathbf{y}_j - \mathbf{y}_i)(p_{ij} - q_{ij})$$

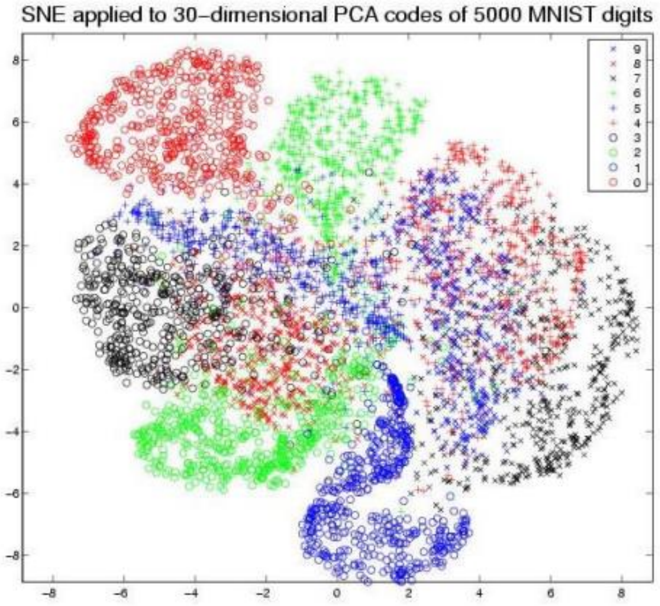
Crowding problem

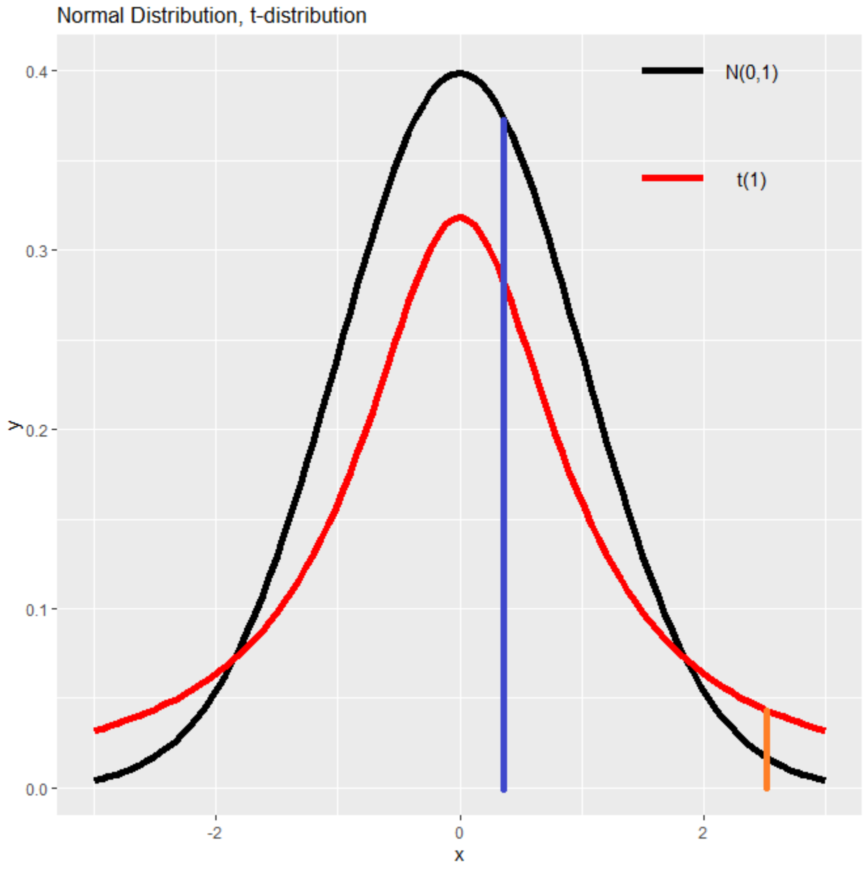
- 앞서 살펴본 SNE 기법들은 Crowding problem 발생

∵ 가우시안 분포 (정규 분포)를 적용해서 인스턴스 확률을 배정했기 때문

- 중심에서 멀어질수록 확률이 급격하게 감소하는 문제

- 해결 Idea! 가우시안 분포 → 스튜던트 t-분포 사용





t-SNE

- 자유도 1의 스튜던트 t-분포 사용
- T분포는 축소된 차원의 확률 분포인 q에만 적용

$$f(t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

$$\Gamma(n) = (n-1)!$$

- Optimization of t-SNE

$$p_{ij} = \frac{e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma_i^2}}}{\sum_{k\neq l} e^{-\frac{||\mathbf{x}_k - \mathbf{x}_l||^2}{2\sigma_i^2}}} \qquad q_{ji} = \frac{(1+||\mathbf{y}_i - \mathbf{y}_j||^2)^{-1}}{\sum_{k\neq l}(1+||\mathbf{y}_k - \mathbf{y}_l||^2)^{-1}}$$

✓ Gradient:

$$\frac{\partial C}{\partial \mathbf{y}_i} = 4\sum_j (\mathbf{y}_j - \mathbf{y}_i)(p_{ij} - q_{ij})(1+||\mathbf{y}_i - \mathbf{y}_j||^2)^{-1}$$

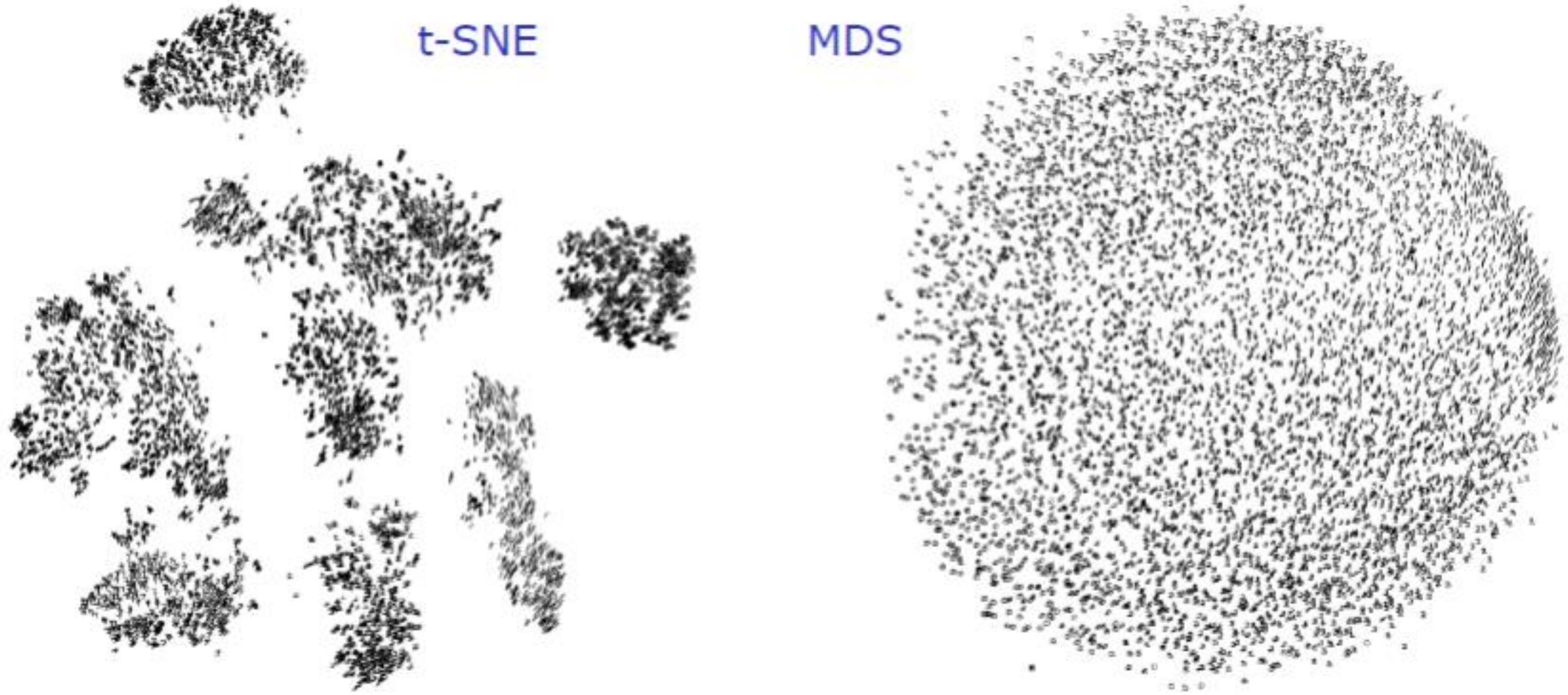
t-SNE 알고리즘

Algorithm 1: Simple version of t-Distributed Stochastic Neighbor Embedding.

Data: data set $\mathcal{X} = \{x_1, x_2, ..., x_n\}$,
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, ..., y_n\}$.
begin
    compute pairwise affinities $p_{j|i}$ with perplexity $Perp$ (using Equation 1)
    set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$
    sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, ..., y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
    for t=*1* **to** T **do**
        compute low-dimensional affinities q_{ij} (using Equation 4)
        compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using Equation 5)
        set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t)\left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)}\right)$
    end
end

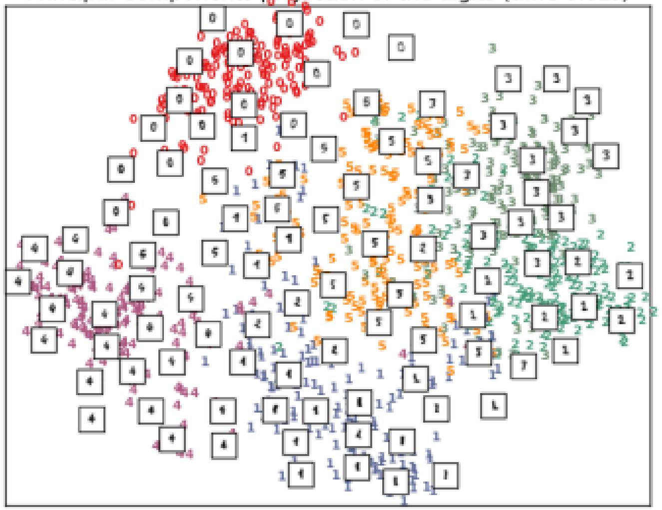
t-SNE vs MDS

- MNIST Dataset

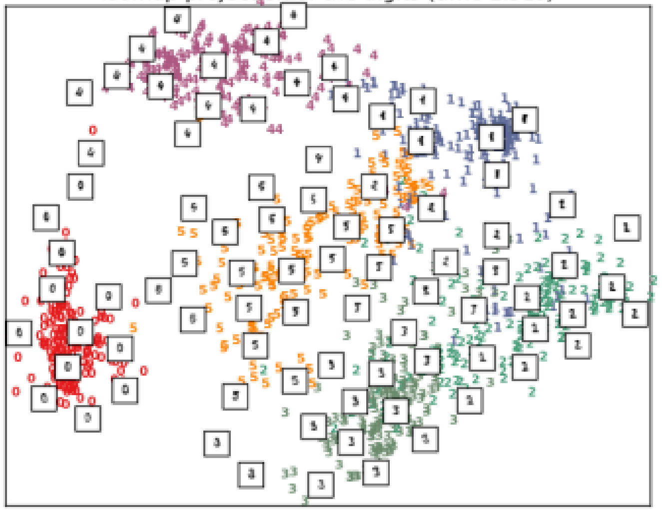


t-SNE Example

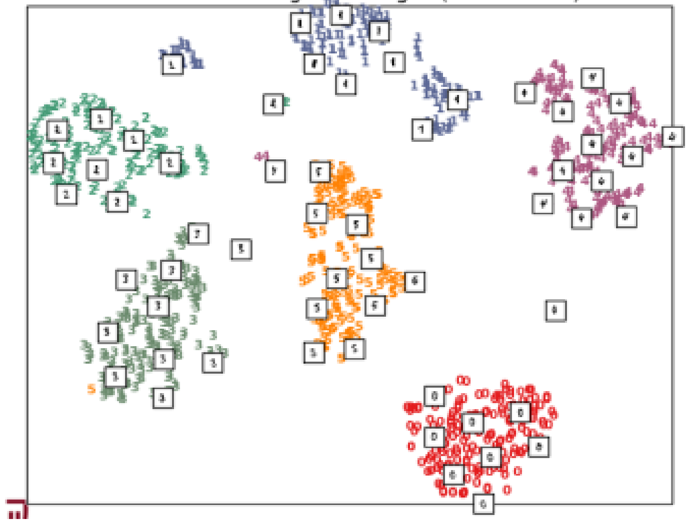
Principal Components projection of the digits (time 0.01s)



Isomap projection of the digits (time 1.51s)



t-SNE embedding of the digits (time 15.61s)



What’s next..

✓ 실습