

# t-SNE

---

발표자: 변준현

# Index

---

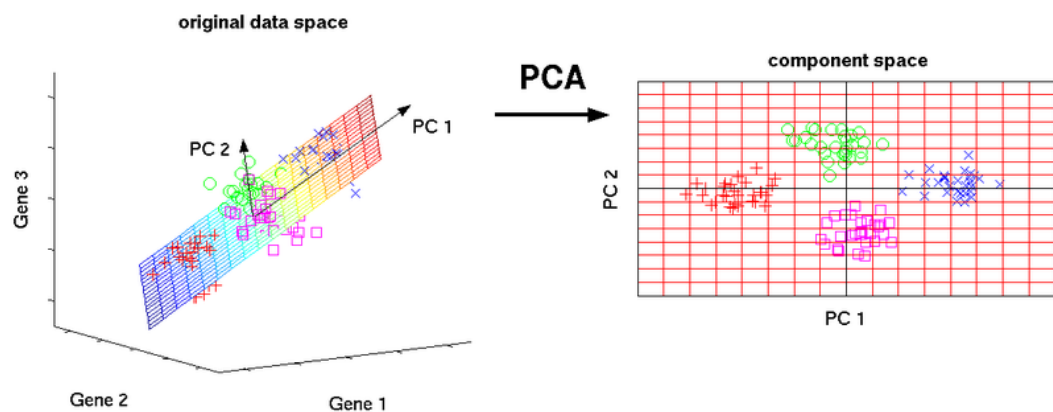
## 목차

- Embedding
  - Embedding
- SNE
  - SNE
  - SNE 한계점
- t-SNE
  - t-SNE
  - t-SNE 성능평가
  - t-SNE 한계점

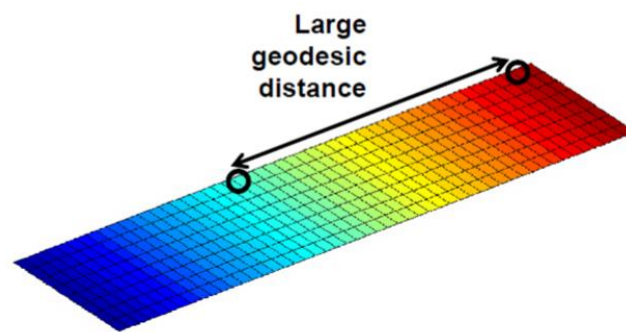
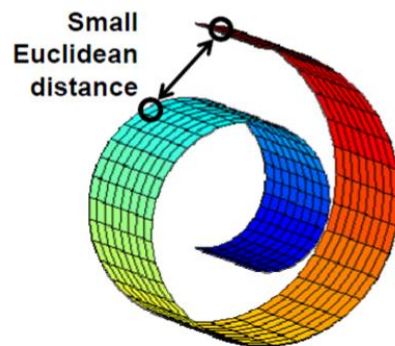
# Embedding

## Embedding이란?

고차원 데이터를 저차원으로 맵핑시키는 방법론 -> 차원축소를 의미한다.



<PCA>



<Isomap>

# SNE (Stochastic Neighbor Embedding)

---

## SNE(Stochastic Neighbor Embedding)

각 관측치에 유클리디안 거리를 구하고 유클리디안 거리에 확률을 부여한다.

Embedding을 이용해서 차원을 축소 시킨 후 축소된 데이터에 대한 유클리디안 거리 확률을 구한다.

기존 데이터의 거리에 대한 확률과 차원축소된 데이터의 거리에 대한 확률이 같아지도록 하는 것이 SNE의 사상이다.

SNE는 시각화에 좋은 성능을 띠며, 데이터는 정규분포를 따른다.

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)}$$

<기존 데이터에 대한 확률>

$$q_{j|i} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)}$$

<차원축소된 데이터에 대한 확률>

# SNE (Stochastic Neighbor Embedding)

---

## SNE (Stochastic Neighbor Embedding)

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)}$$

<기존 데이터에 대한 확률>

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

<정규분포의 확률밀도함수>

# SNE (Stochastic Neighbor Embedding)

---

## Kullback-Leibler divergence

SNE의 비용함수이고 두 확률분포의 차이를 계산하는데 사용이 된다.

$p_{ij}$  와  $q_{ij}$ 가 같아지도록 하는 것이 목적이기 때문에 0에 근접한 수를 찾는 것을 목적으로 한다.

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

# SNE (Stochastic Neighbor Embedding)

---

## Gradient Descent

SNE의 식 자체가 굉장히 복잡하기 때문에 수치최적화 기법을 이용해서 해를 구한다.

$$\frac{\delta C}{\delta y_i} = 2 \sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j)$$

# SNE (Stochastic Neighbor Embedding)

---

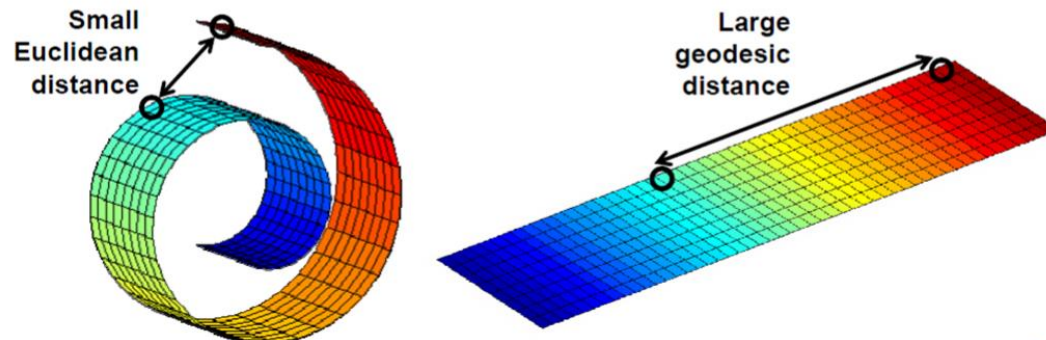
## SNE (Stochastic Neighbor Embedding) 의 한계점

- 비용함수 최적화 문제

목적함수가 굉장히 복잡해서 최적화가 어렵다.

- Crowding problem

고차원에서 저차원으로 차원축소 시킬 경우 거리가 멀고 가까운 개념이 붕괴되는 경우가 있다.





# t-SNE (t-Stochastic Neighbor Embedding)

---

## t-SNE(t-Stochastic Neighbor Embedding)

- SNE의 대칭 버전인 비용함수를 사용하며 더 간단하다.  
비용함수를 더 간단 명료하게 해준다.
- 저차원 공간에서 두 점 사이의 유사성을 계산하기 위해 가우시안 분포가 아닌 Student t분포를 사용한다.  
crowding 문제를 어느정도 방지해준다.

# t-SNE (t-Stochastic Neighbor Embedding)

---

## t-SNE(t-Stochastic Neighbor Embedding)의 비용함수

수학적 테크닉을 이용해서 식의 복잡도를 최소화 시킨다.

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j).$$

# t-SNE (t-Stochastic Neighbor Embedding)

---

t-SNE(t-Stochastic Neighbor Embedding)의 비용함수

$$\frac{\delta C}{\delta y_i} = 2 \sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j)$$

<비대칭 SNE>

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)$$

<대칭 SNE>

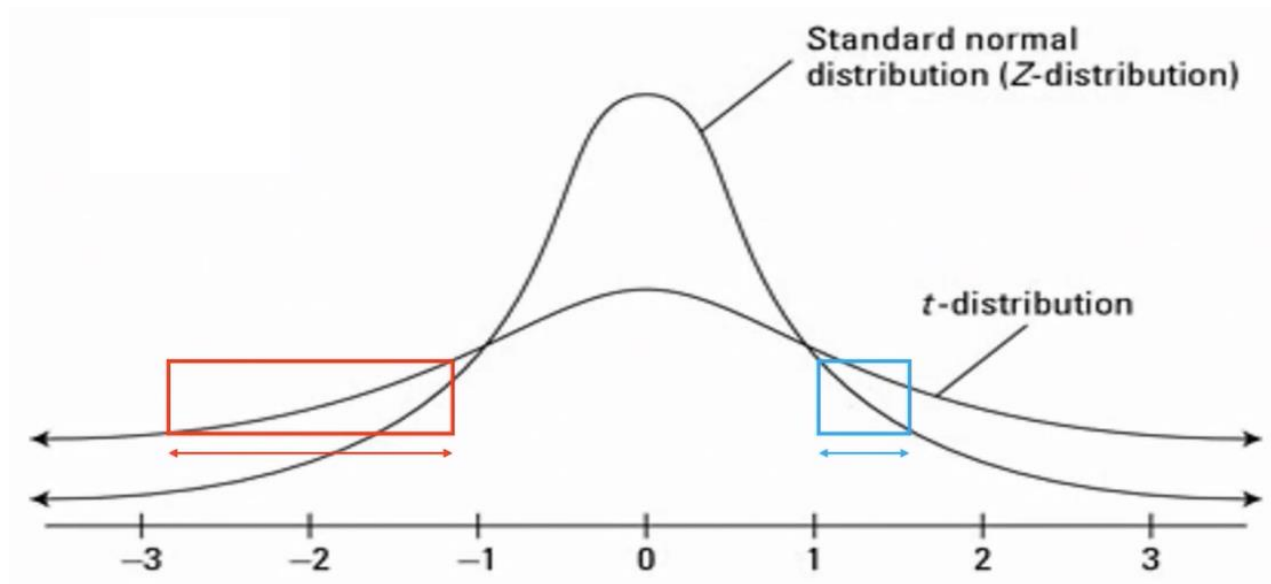
# t-SNE (t-Stochastic Neighbor Embedding)

---

## t-SNE(t-Stochastic Neighbor Embedding)의 Student t-분포 사용

가우시안 분포의 경우 거리가 멀어졌을 때의 확률이 급격하게 낮아지는 경향이 있음

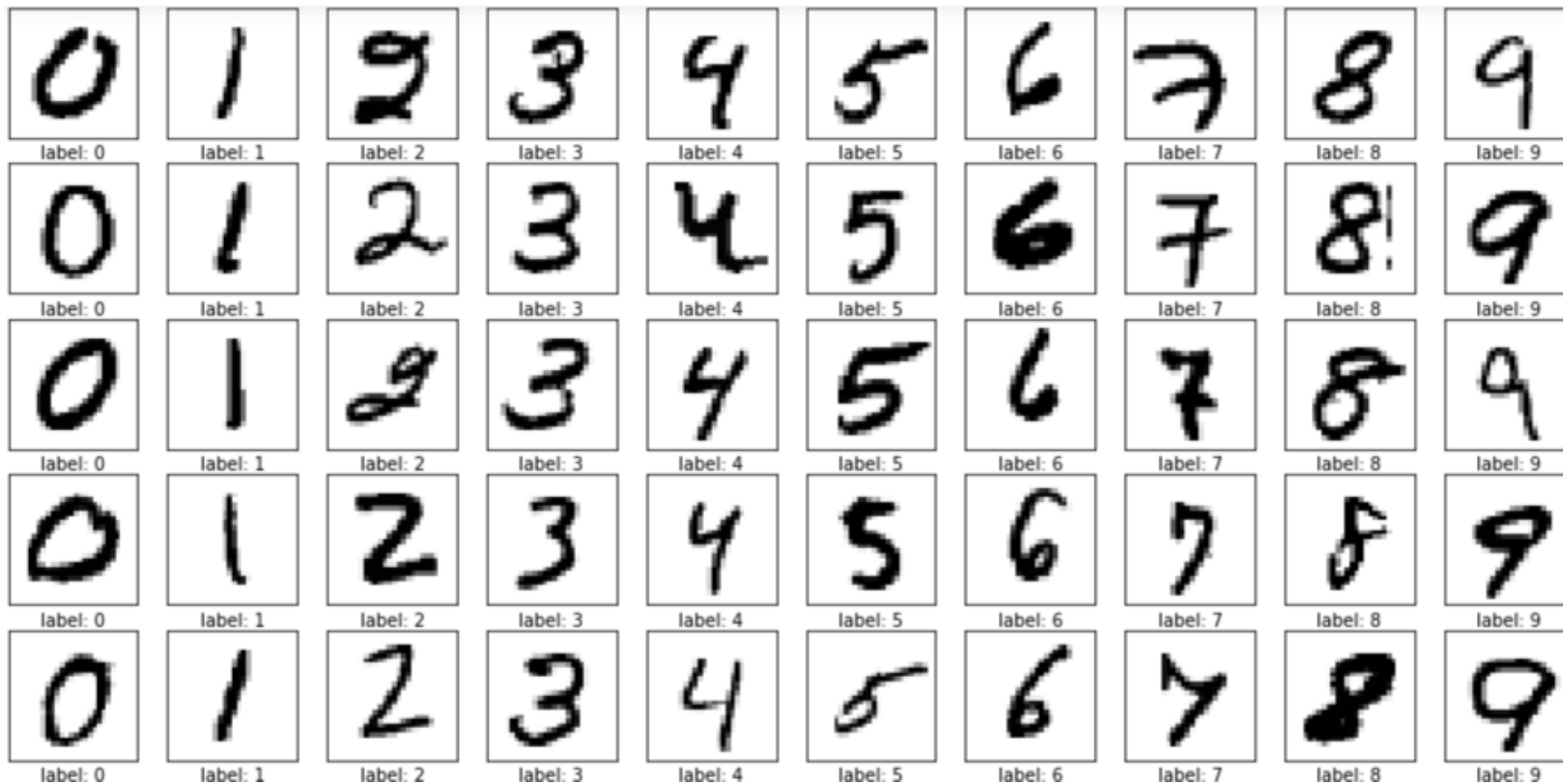
-> 가우시안 분포보다 봉우리가 낮고 꼬리가 두터운 t분포를 사용



# t-SNE의 평가

---

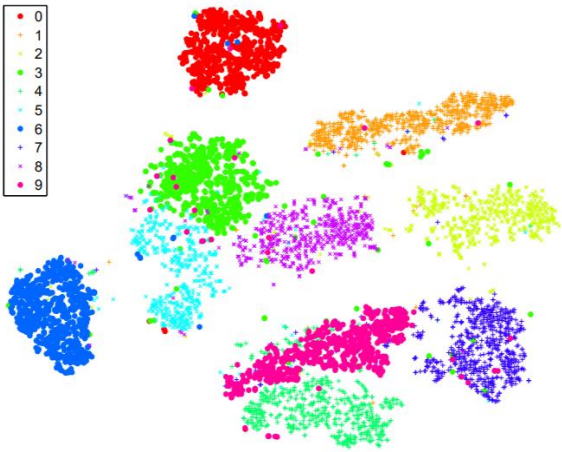
## MNIST 손글씨 데이터



# t-SNE의 평가

## MNIST 손글씨 데이터

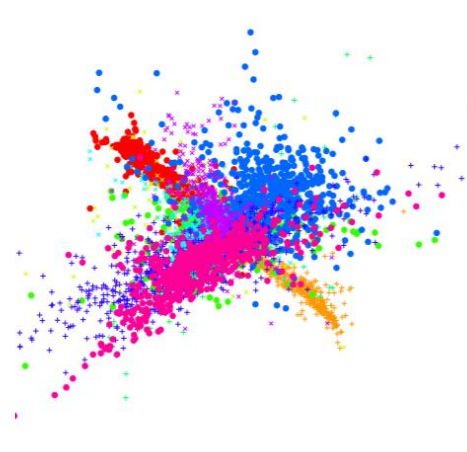
0  
1  
2  
3  
4  
5  
6  
7  
8  
9



<t-SNE>



<Isomap>



<LLE>



<Sammon mapping>

# t-SNE의 평가

---

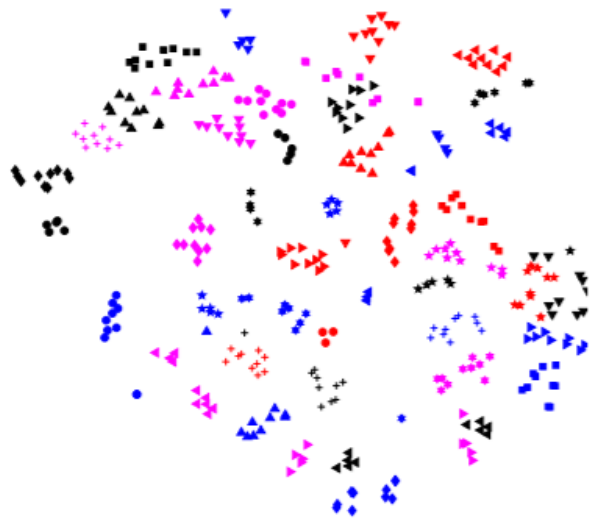
## Olivetti 얼굴 데이터



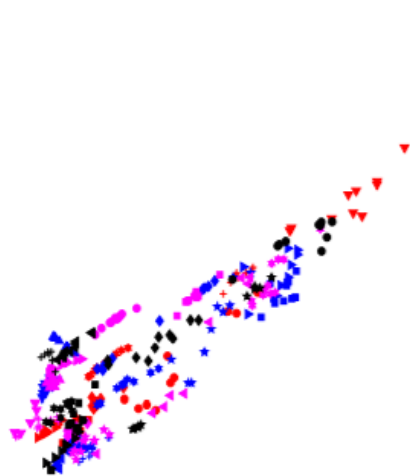
# t-SNE의 평가

---

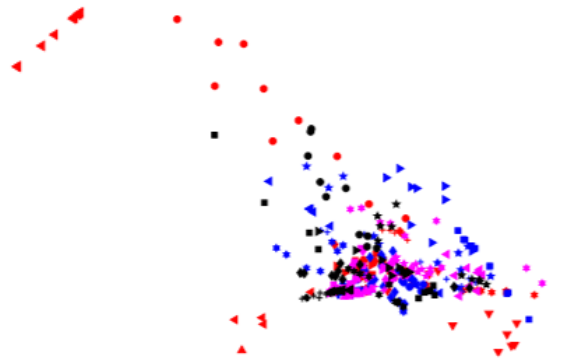
## Olivetti 얼굴 데이터



<t-SNE>



<Isomap>



<LLE>



<Sammon mapping>



# t-SNE의 한계점

---

## t-SNE의 한계점

- **t-SNE는 일반적인 경우에 성능이 불명확하다.**

t-SNE는 데이터 시각화를 목적으로 한 차원축소이기 때문에 일반적인 경우에는 성능이 불명확하다.

- **차원의 저주**

t-SNE는 유클리디안 거리를 사용했기 때문에 데이터를 가까이서 봤을 때 선형일 것이라는 가정을 하고 진행을 하는데 차원의 저주는 차원이 증가할 수록 위 가정이 위배되는 것을 의미한다. 해결 방법으로는 AutoEncoder를 이용해서 비선형 차원축소를 먼저 이용하고 t-SNE를 이용하는 방법이 있다.

- **비용함수가 convex하지 않다.**

t-SNE를 실행할 때마다 최적화된 파라미터가 달라질 수 있는 문제가 있다.

# Thank You

---