Week #12

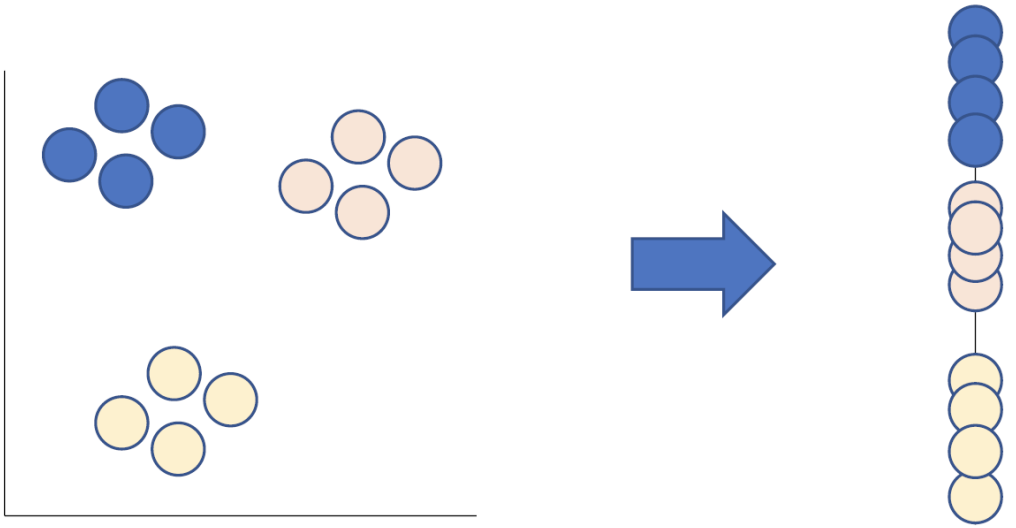
t-SNE & 자연어 처리: Pre-trained model 시대

May 24, 2023

Sources: Source: 파이썬 딥러닝 파이토치 (이경택,방성수, 안상준 지음), 정보문화사. Materials on the Internet including the Wikipedia/YouTube This book was modified and provided by Hyun-Chul Kim, Ph.D.

t-SNE (t-Stochastic Neighbor Embedding)

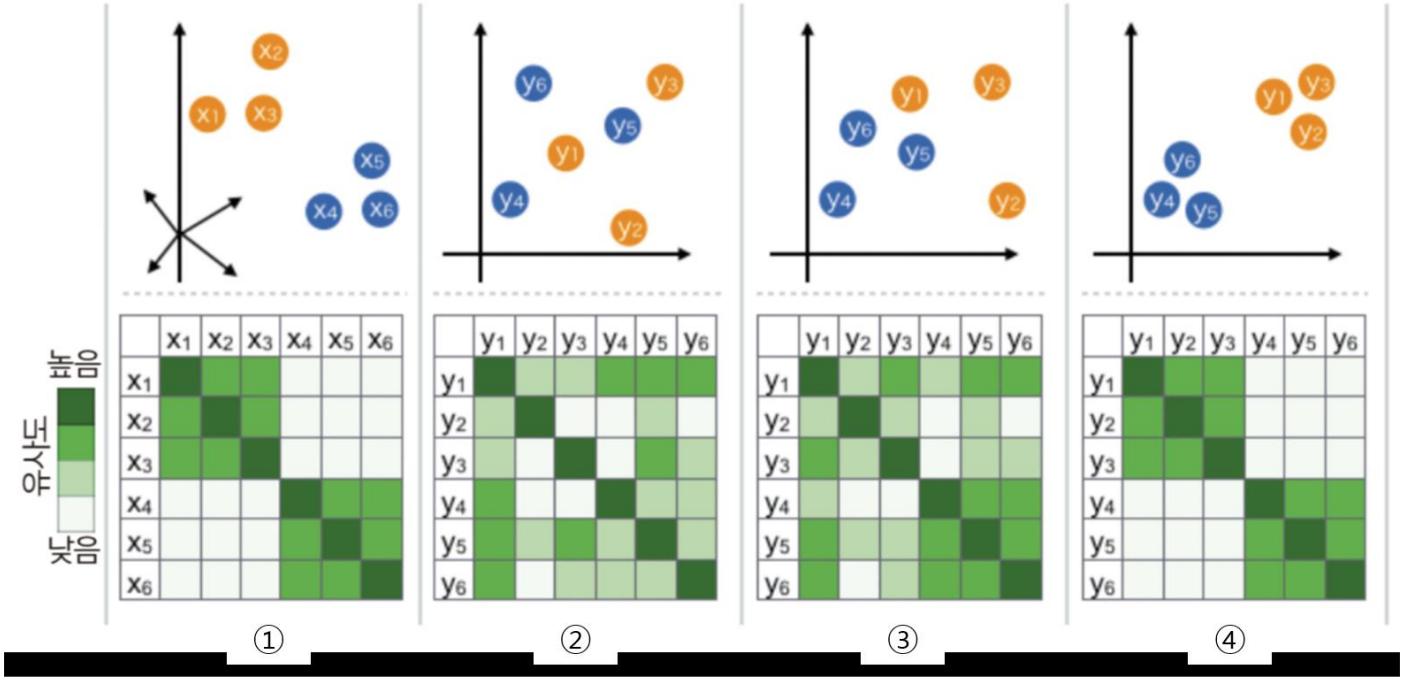
- t-SNE
  - 높은 차원의 복잡한 데이터를 2차원에 차원 축소하는 방법입니다. 낮은 차원 공간의 시각화에 주로 사용하며 차원 축소할 때는 비슷한 구조끼리 데이터를 정리한 상태이므로 데이터 구조를 이해하는 데 도움을 줍니다.
  - t-SNE는 **비선형적인 방법**의 차원 축소 방법이고 특히 고차원의 데이터 셋을 시각화하는 것에 성능이 좋습니다.



https://gaussian37.github.io/ml-concept-t_sne/

t-SNE

- t-SNE 학습과정
  - 아래 그림에서 x_i는 기존 데이터에 해당하며 고차원에 분포되어 있고 y_i는 t-SNE를 통하여 저차원으로 매핑된 데이터로 볼 수 있습니다. 예시에서 기존 데이터는 3차원이고 저차원은 2차원으로 사용되었습니다



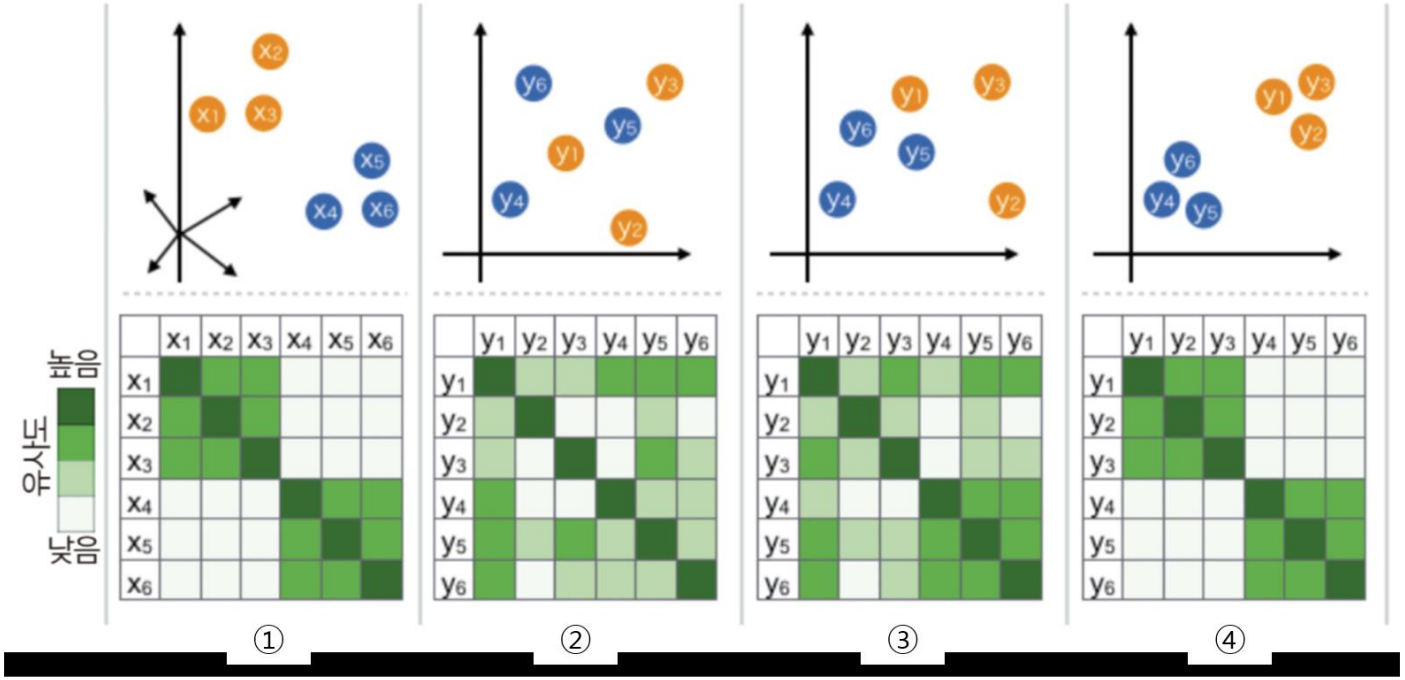
https://gaussian37.github.io/ml-concept-t_sne/

t-SNE

- t-SNE 학습과정

① 모든 i,j 쌍에 대하여 x_i,x_j의 유사도를 가우시안 분포를 이용하여 나타냅니다.

② x_i 와 같은 개수의 점 y_i를 낮은 차원 공간에 무작위로 배치하고, 모든 i,j쌍에 관하여 y_i,y_j의 유사도를 t-SNE를 이용하여 나타냅니다



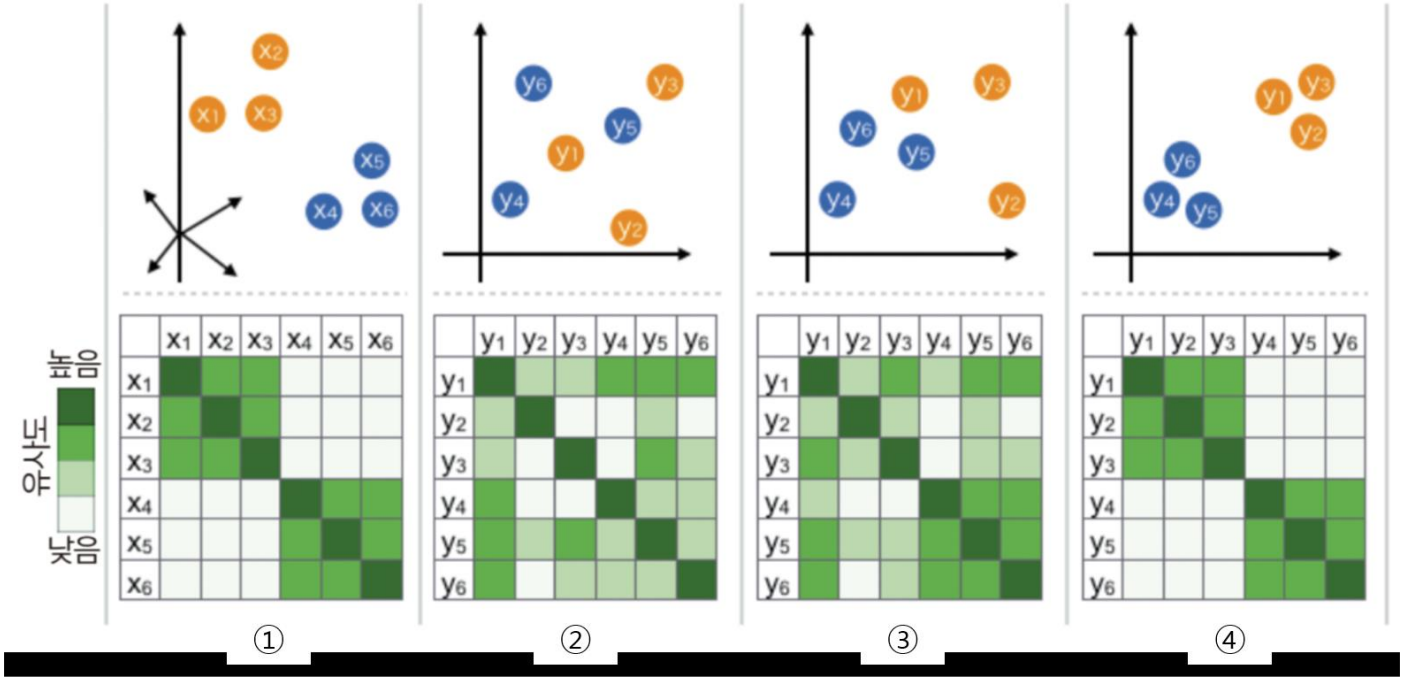
https://gaussian37.github.io/ml-concept-t_sne/

t-SNE

- t-SNE 학습과정

  ③ 앞의 ①, ②에서 정의한 유사도 분포가 가능하면 같아지도록 데이터 포인트 y_i를 갱신합니다.

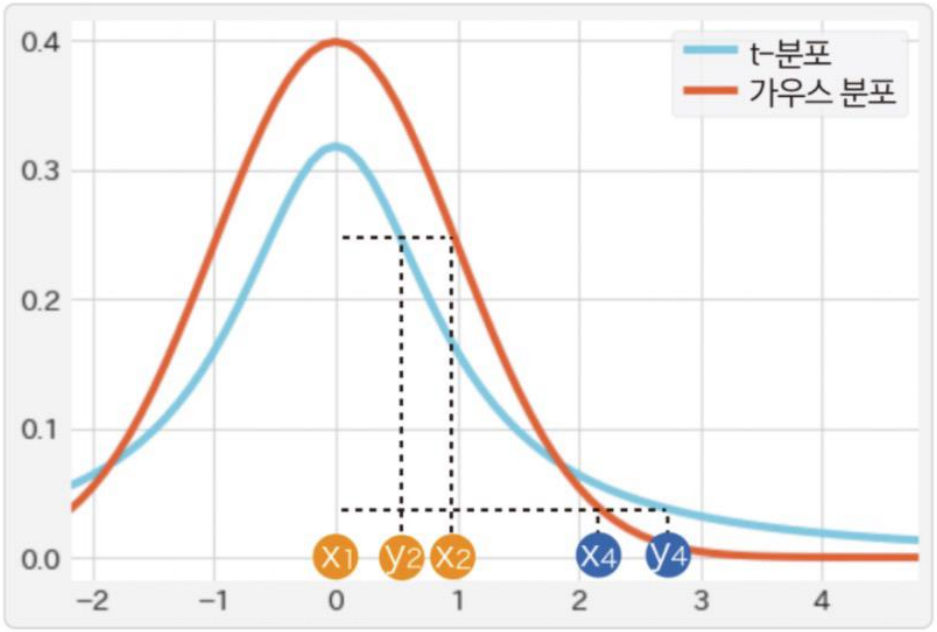
  ④ 수렴 조건까지 과정 ③을 반복합니다.



https://gaussian37.github.io/ml-concept-t_sne/

t-SNE

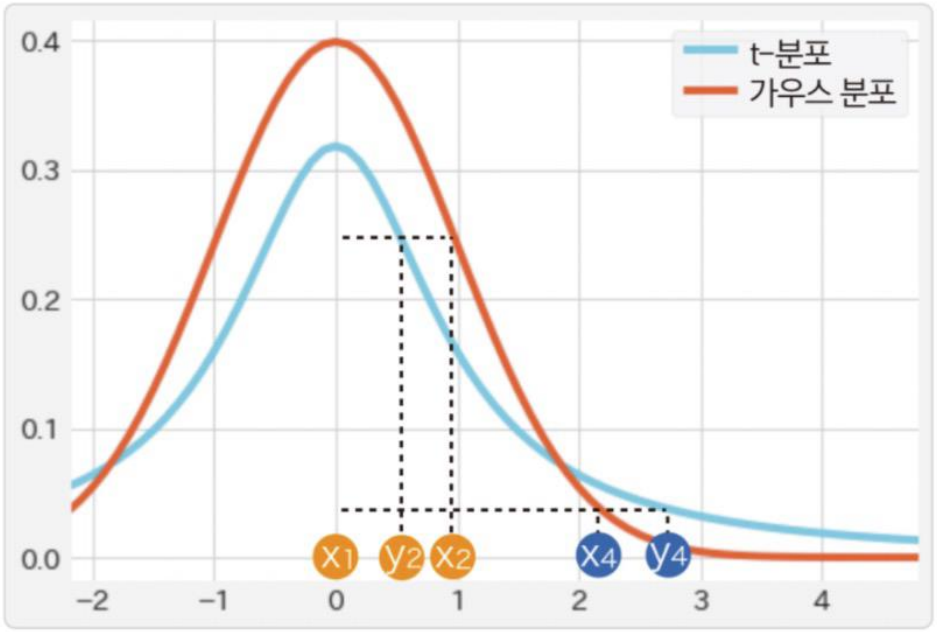
- t-SNE 학습과정
  - 이전 알고리즘에서 ①, ②의 유사도는 데이터 포인트들이 얼마나 비슷한 지 나타냅니다. 단순히 데이터 사이의 거리를 이용하는 것이 아니라 확률 분포를 이용합니다.
  - 아래 그래프는 가로축으로 거리, 세로축으로 유사도를 설정하여 t-분포와 가우시안 분포를 비교한 것입니다.



https://gaussian37.github.io/ml-concept-t_sne/

t-SNE

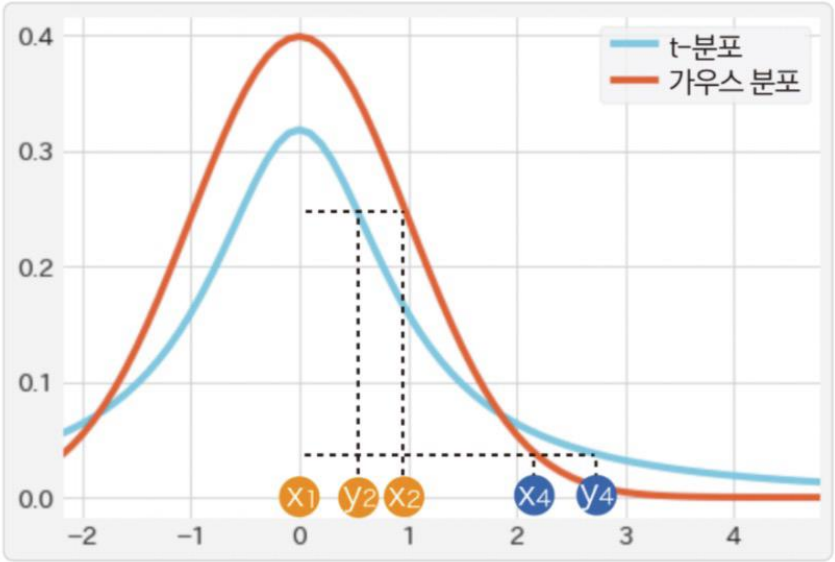
- t-SNE 학습과정
  - **데이터 사이의 거리가 가까울수록 유사도가 크고, 멀수록 유사도가 작아집니다.** 먼저 원본의 높은 차원 공간에서 정규 분포로 유사도를 계산하고 p_{ij}라는 분포로 나타냅니다. p_{ij}는 데이터 포인트 x_i, x_j 의 유사도를 나타냅니다.
  - 다음으로 x_i 에 대응하는 데이터 포인트 y_i 를 낮은 차원 공간에 무작위로 배치합니다. y_i에 관해서도 t-분포로 유사도를 나타내는 q_{ij}를 계산합니다



https://gaussian37.github.io/ml-concept-t_sne/

t-SNE

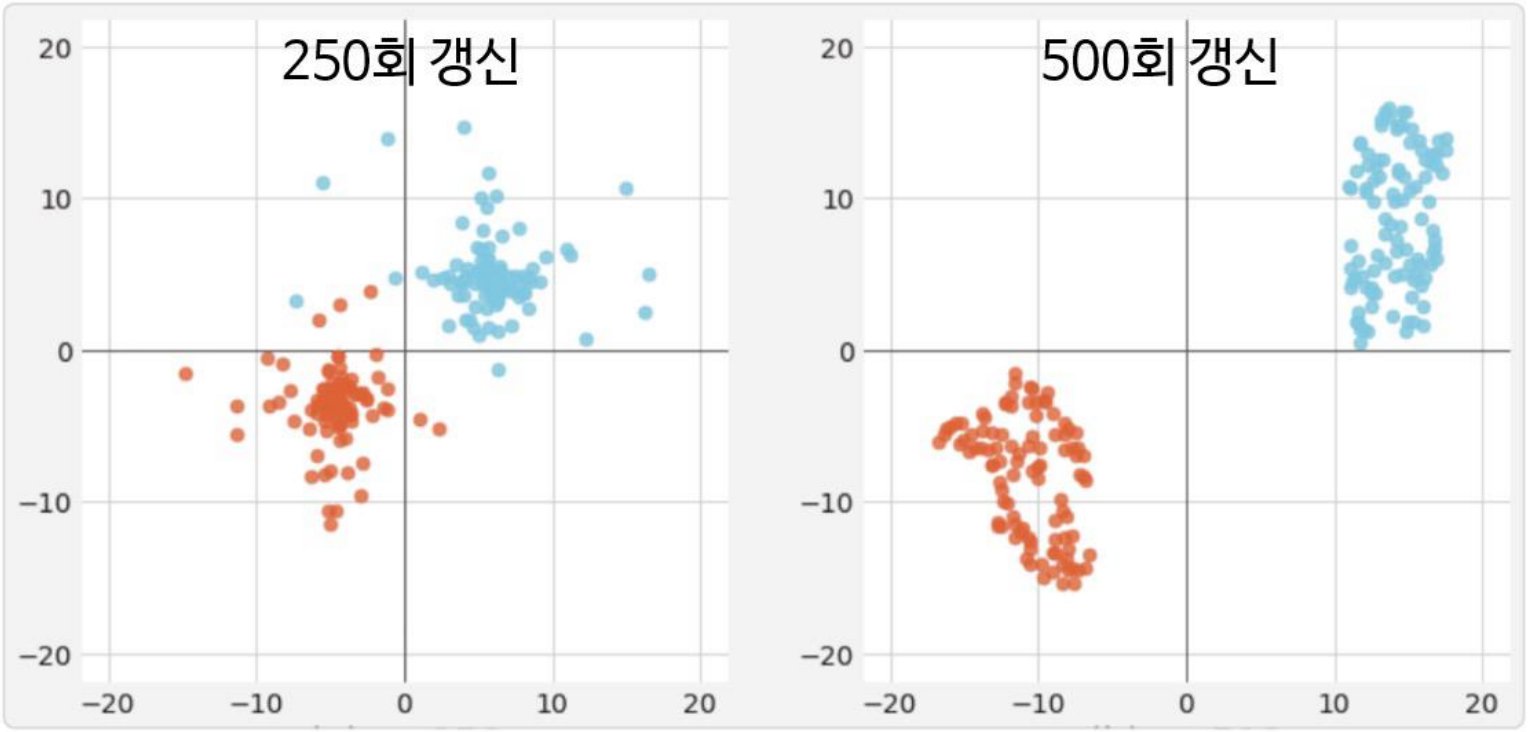
- t-SNE 학습과정
  - 여기서 p_{ij} 와 q_{ij} 를 계산하면 q_{ij} 를 p_{ij} 와 같은 분포가 되도록 데이터 포인트 y_i 를 갱신합니다. 이는 높은 차원 공간의 x_i 유사도 각각의 관계를 낮은 차원 공간의 y_i에서 재현하는 것입니다.
  - 이 때, 낮은 차원 공간에서 t-분포를 이용하므로, **유사도가 큰 상태**의 관계를 재현할 때는 낮은 차원 공간에서 **데이터 포인트를 더 가까이** 배치합니다. 반대로 **유사도가 작은 상태의 관계**를 재현할 때에는 **낮은 차원 공간에서 데이터 포인트를 더 멀리 배치**합니다.



https://gaussian37.github.io/ml-concept-t_sne/

t-SNE

- t-SNE 학습과정
  - 아래 그림은 t-SNE를 적용하였을 때, 데이터 포인트 y_i를 갱신하는 모습입니다. 왼쪽 그래프는 갱신 횟수가 250회이고 오른쪽 그래프는 갱신 횟수가 500회 입니다. **갱신 횟수가 늘수록 데이터 포인트의 차이를 명확하게 나타냅니다**.



https://gaussian37.github.io/ml-concept-t_sne/

왜 t-분포를 사용?

- t-SNE
  - 낮은 차원에 임베딩 할 때, 정규 분포를 사용하지 않고 t-분포를 사용합니다. 그 이유는 앞에서 다루었듯이 t-분포가 heavy-tailed distribution임을 이용하기 위해서 입니다. 즉, t-분포는 **일반적인 정규분포보다 끝단의 값이 두터운 분포**를 가집니다.
  - t-SNE가 전제하는 확률 분포는 정규 분포이지만 정규 분포는 꼬리가 두텁지 않아서 *i*번째 개체에서 적당히 떨어져 있는 이웃 *j*와 아주 많이 떨어져 있는 이웃 k가 선택될 **확률이 크게 차이가 나지 않게** 됩니다.
  - 또한 **높은 차원 공간에서는 분포의 중심에서 먼 부분의 데이터 비중이 높기 때문에** 데이터 일부분의 정보를 고차원에서 유지하기가 어렵습니다.
  - 이러한 문제로 인하여 구분을 좀 더 잘하기 위해 정규 분포보다 **꼬리가 두터운 t분포**를 사용합니다.

https://www.youtube.com/watch?v=NEaUSP4YerM