단어/문서 시각화

김현중 (soy.lovit@gmail.com)

Embedding?

- **임베딩은 X라는 공간의 데이터에서 원하는 정보를 잘 저장하며, Y라는 새로운 공간으로 보내는 f: X → Y 함수**
  - (예시) 10,000개 단어로 이뤄진 문서 (1만차원)들의 유사도를 잘 보존하여 2차원으로 보내는 것
  - **어떤 정보를 보존할 것이냐**에 따라서 다양한 임베딩 방법이 존재

- **(벡터) 시각화는 고차원으로 표현되는 객체**(단어/문서/어떤 것이든)**를 2차원의 저차원 백터로 표현하는 것**
  - 임베딩을 흔히 차원축소라고 부르는 이유

Multidimensional Scaling (MDS)

- **Δ 공간 벡터 간의 거리 $\delta_{i,j}$를 Euclidean distance 으로 보존하는 저차원 x를 학습**

  - 거리 행렬 Δ의 각 포인트간 거리를 가장 잘 저장하는 새로운 공간 x를 학습

minimizes $\sum_{i<j}\left(\left|x_i - x_j\right| - \delta_{i,j}\right)^2$

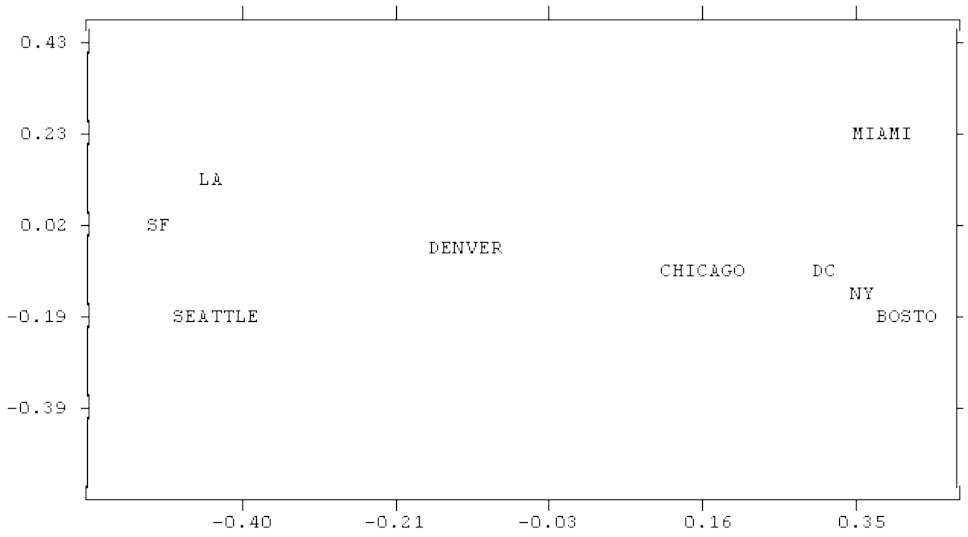
where $\Delta = \begin{pmatrix} \delta_{1,1} & \cdots & \delta_{1,n} \\ \vdots & \ddots & \vdots \\ \delta_{n,1} & \cdots & \delta_{n,n} \end{pmatrix}$

* Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, *29*(1), 1-27.

Multidimensional Scaling (MDS)

- 도시간 거리를 행렬로 만든 뒤, row를 x_i로 이용하면 실제 지도의 거리가 복원됨

		1 BOST	2 NY	3 DC	4 MIAM	5 CHIC	6 SEAT	7 SF	8 LA	9 DENV
1	BOSTON	0	206	429	1504	963	2976	3095	2979	1949
2	NY	206	0	233	1308	802	2815	2934	2786	1771
3	DC	429	233	0	1075	671	2684	2799	2631	1616
4	MIAMI	1504	1308	1075	0	1329	3273	3053	2687	2037
5	CHICAGO	963	802	671	1329	0	2013	2142	2054	996
6	SEATTLE	2976	2815	2684	3273	2013	0	808	1131	1307
7	SF	3095	2934	2799	3053	2142	808	0	379	1235
8	LA	2979	2786	2631	2687	2054	1131	379	0	1059
9	DENVER	1949	1771	1616	2037	996	1307	1235	1059	0



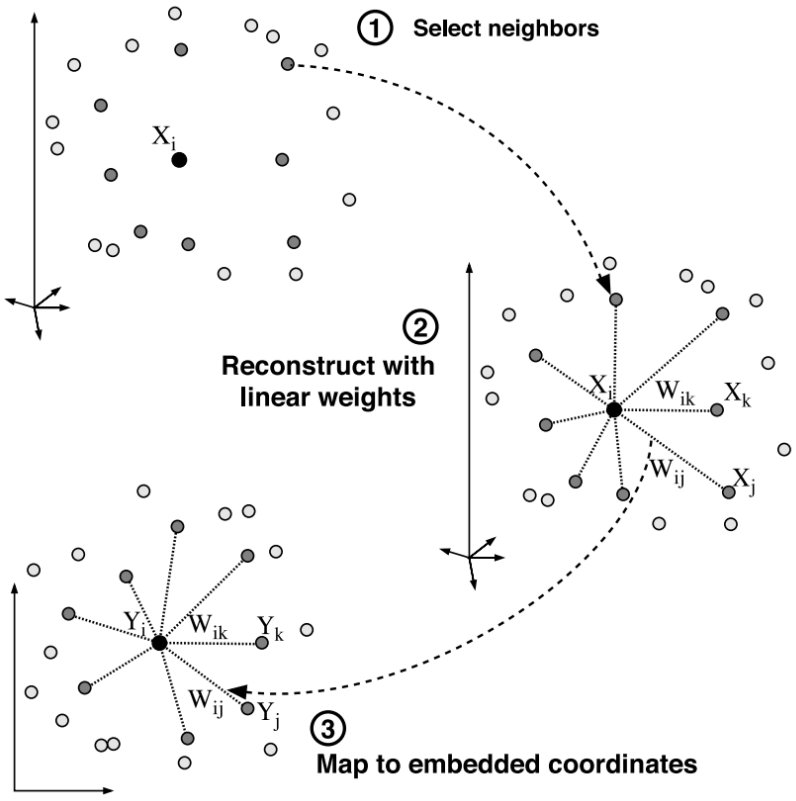
Locally Linear Embedding (LLE)

- **LLE는 한 점 x_i의 local geometry, 주위 k개의 점들과의 구조를 보존하는 새로운 공간 벡터 y_i를 학습하며, 세 단계의 학습 단계로 이뤄짐**

  - 1단계: x_i와 가까운 k개의 이웃을 선택
  - 2단계: 본래 공간에서의 이웃간의 구조학습

    minimizes $\varepsilon(W) = \sum_i \left| x_i - \sum_j w_{ij} x_j \right|^2$
  - 3단계: W를 보존하는 y_i학습

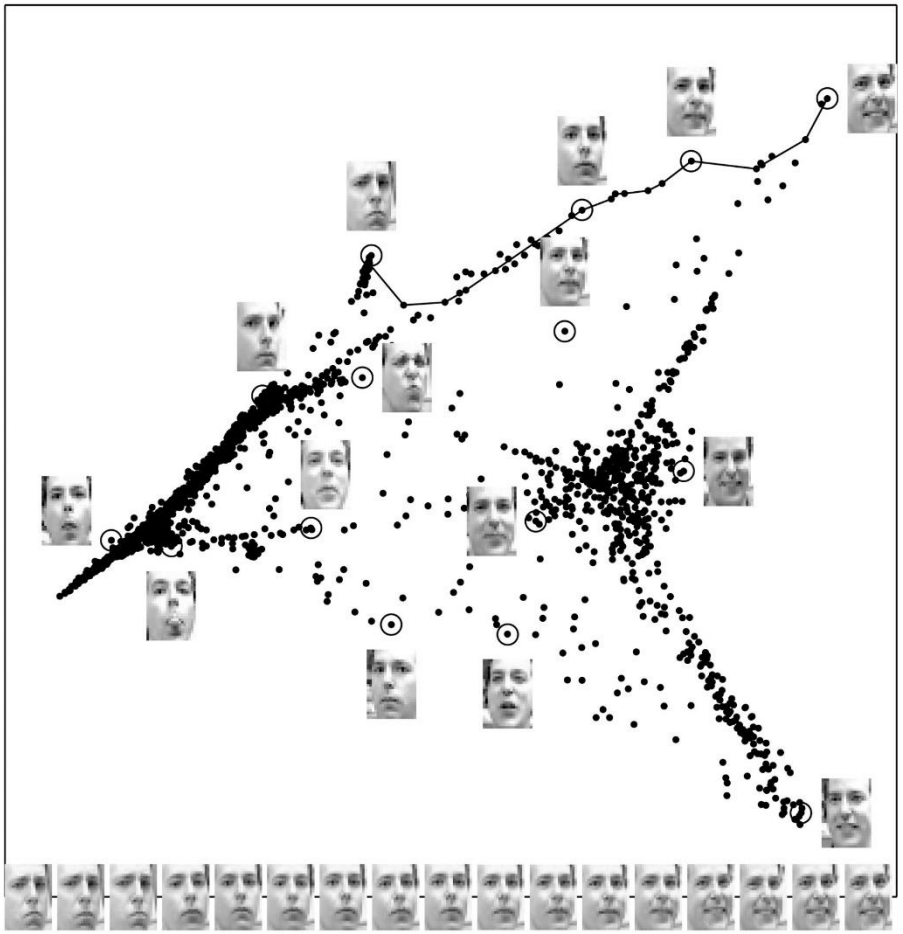
    $\varphi(Y) = \sum_i \left| y_i - \sum_j w_{ij} y_j \right|^2$



* Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*(5500)

Locally Linear Embedding (LLE)

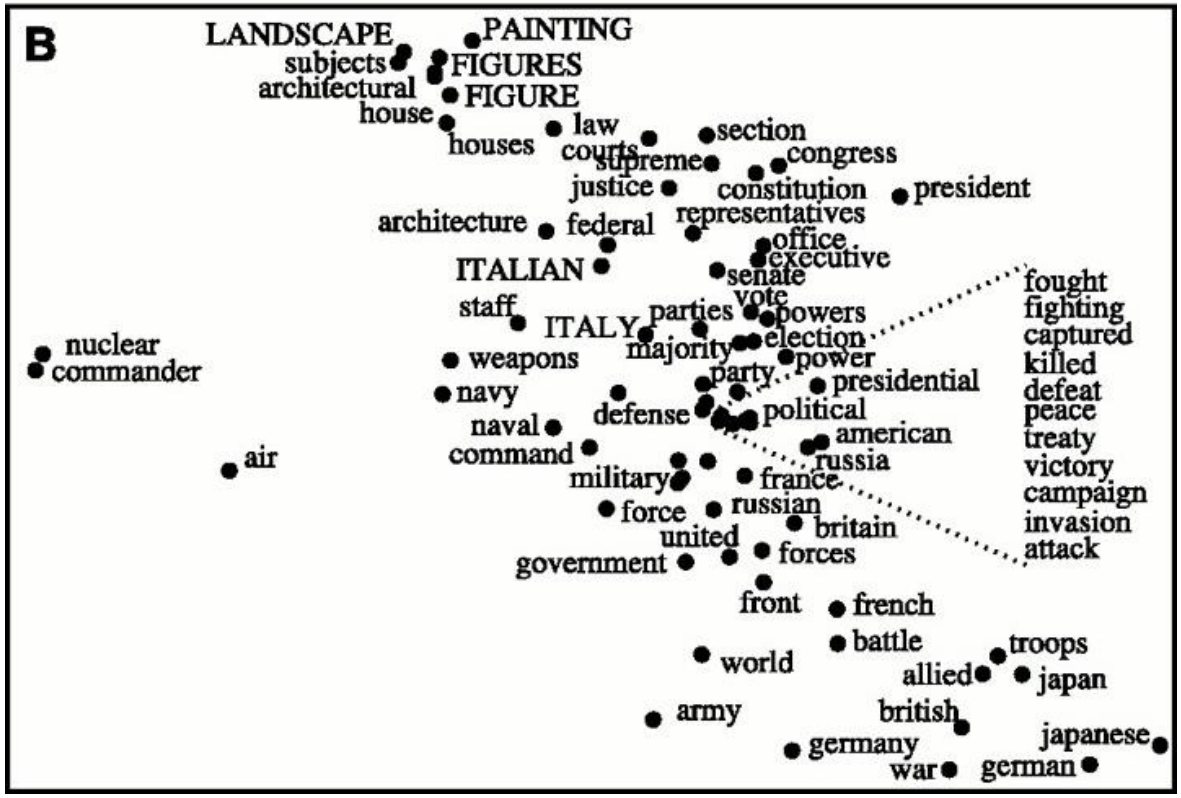
- **비슷한 점들간의 지역적 구조만을 보존해도 "어떤 흐름"이 학습됨**
  - 얼굴 이미지 데이터를 LLE로 시각화한 예시



* Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*(5500)

Locally Linear Embedding (LLE)

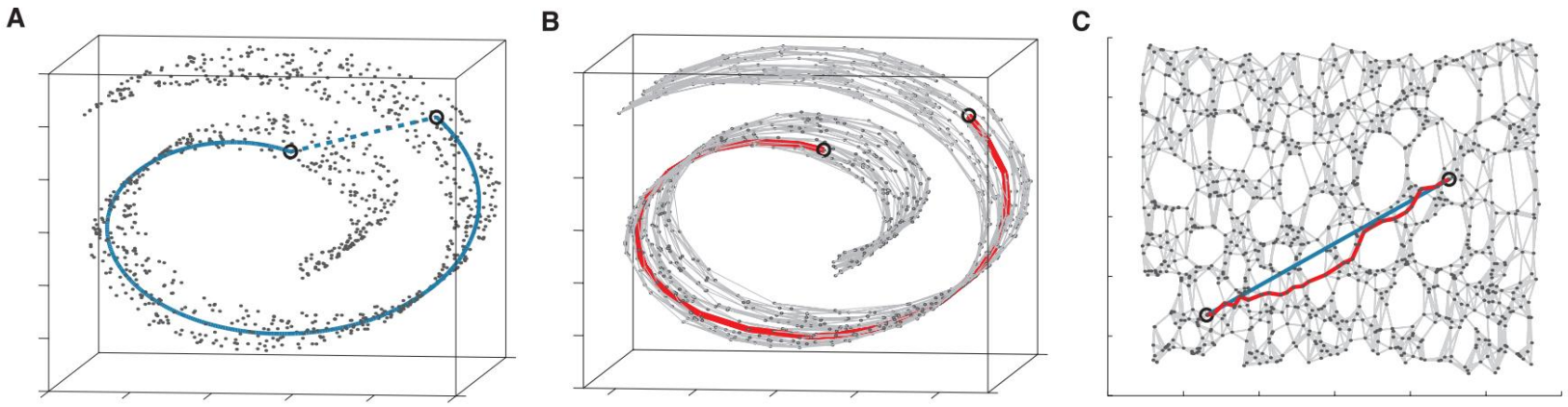
- **비슷한 점들간의 지역적 구조만을 보존해도 "어떤 흐름"이 학습됨**
  - Term-document matrix를 단어 기준으로 임베딩한 예시 (topic modeling)



* Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*(5500)

ISOMAP

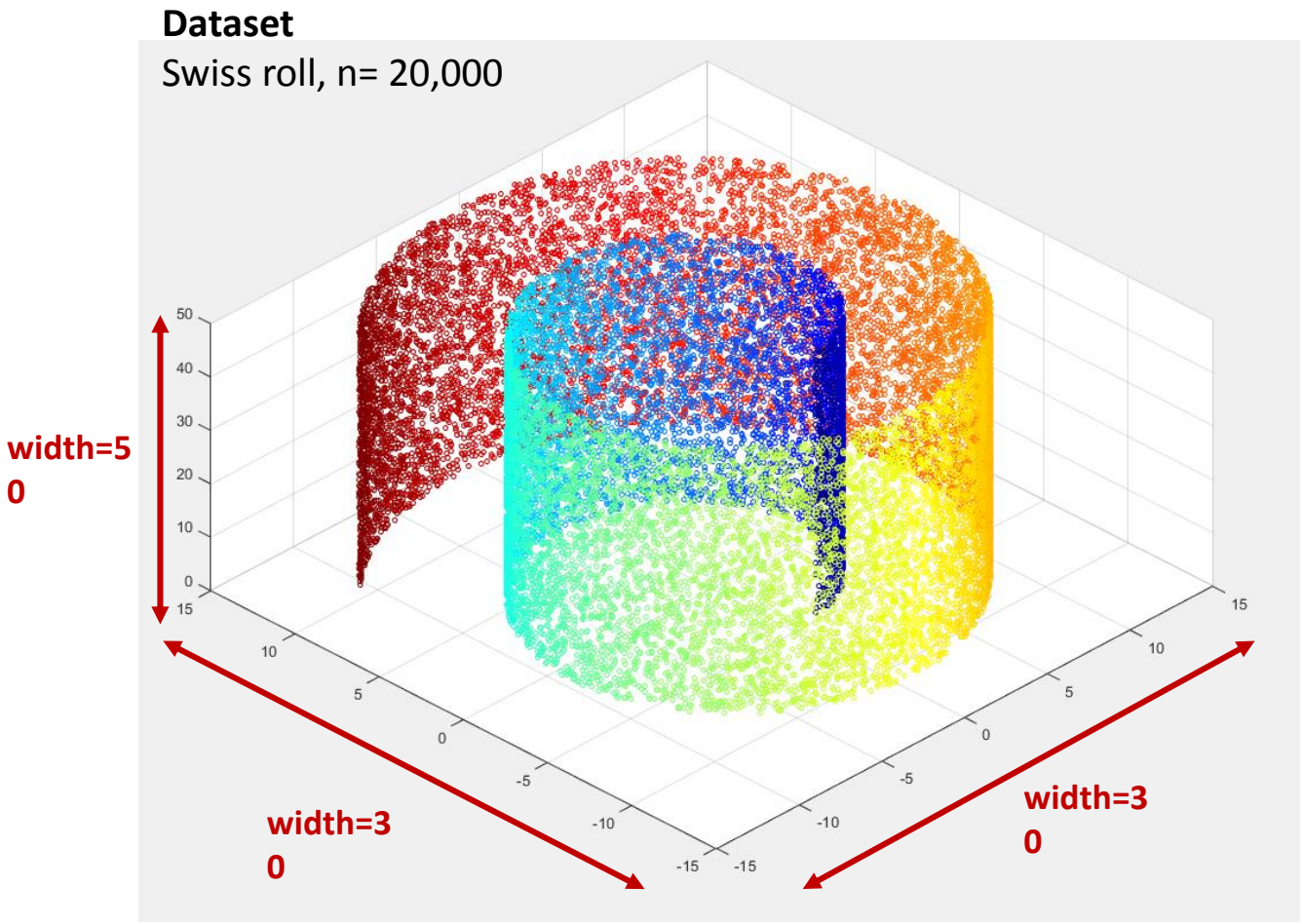
- ***k*** **nearest neighbor graph를 만든 뒤, 두 점간의 거리를 본래 공간 x에서의 Euclidean distance가 아닌, graph에서의 shortest path distance가 보존되는 새로운 공간 (C) 벡터를 학습**



* Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*,

ISOMAP

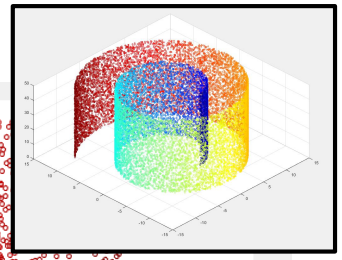
- **ISOMAP은 Swiss roll과 같이 어떤 구조를 지니는 데이터를 잘 시각화 한다고 알려짐**



* Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*,

ISOMAP

- **ISOMAP은 Swiss roll과 같이 어떤 구조를 지니는 데이터를 잘 시각화 한다고 알려짐**



* Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*,

t-Stochastic Neighbor Embedding (t-SNE)

- t-SNE는 최근 시각화 방법으로 가장 널리 쓰이고 있는 임베딩 알고리즘으로, x_i의 이웃간의 거리를 확률적으로 표현한 뒤, X에서의 확률적 거리 정보를 보존하는 y_i를 학습

Find y_i that minimizes $\sum p_{ij} * \log \frac{p_{ij}}{q_{ij}}$

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

$$q_{ij} = \frac{\left(1 + \left|y_i - y_j\right|^2\right)^{-1}}{\sum_{k \neq l} (1 + |y_k - y_l|^2)^{-1}}$$

* Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(2579-2605), 85.

t-Stochastic Neighbor Embedding (t-SNE)

- t-SNE는 X에서의 $p_{j|i}$가 큰 x_i, x_j가 q_{ij}도 크도록 q를 학습하는 것

  - X에서 nearest neighbor graph를 만든 뒤, 그대로 Y라는 공간으로 이동하는 것

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

  - LLE와 비슷하지만, 학습방법이 gradient descent (NN 학습방법)을 이용하고, k-NN을 찾는 것이 아니라는 점이 다름
    - nearest neighbor를 표현하는 방법이 k-NNG가 아니라 $p_{j|i}$

t-Stochastic Neighbor Embedding (t-SNE)

- 처음 제안된 t-SNE (Maaten & Hinton, 2008)는 계산 복잡도가 높아서 큰 데이터의 시각화에 사용되지 못함
- 이후 개선된 Barnes hut t-SNE (Maaten, 2014)이 제안되었으며, 대부분의 패키지는 이 알고리즘을 쓰고 있음

sklearn.manifold.TSNE

class sklearn.manifold. TSNE (*n_components=2, perplexity=30.0, early_exaggeration=4.0, learning_rate=1000.0, n_iter=1000, n_iter_without_progress=30, min_grad_norm=1e-07, metric='euclidean', init='random', verbose=0, random_state=None, method='barnes_hut', angle=0.5*) ¶ [source]

* L.J.P. van der Maaten and G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research* 9(Nov), 2008
** L.J.P. van der Maaten. Accelerating t-SNE using Tree-Based Algorithms. *Journal of Machine Learning Research* 15(Oct):3221-3245, 2014

t-Stochastic Neighbor Embedding (t-SNE)

- **x에서 고려하는 최인접이웃의 개수는 perplexity에 의하여 조절됨**
  - Perplexity가 클수록 더 많은 점을 고려하게 되며,
  - 적은 수의 데이터를 임베딩할 경우, 임베딩이 잘 되지 않으면 perplexity를 줄이면 됨

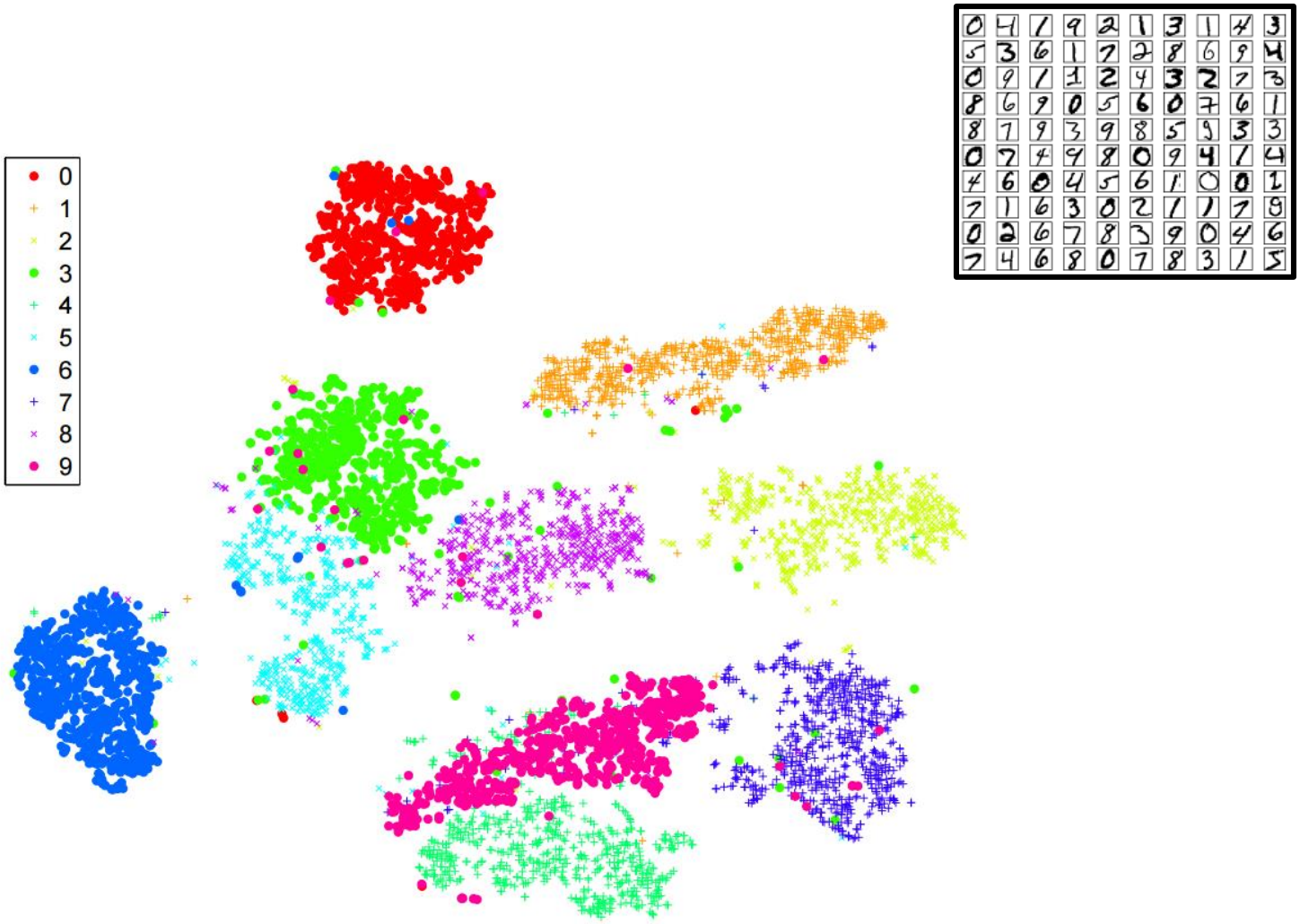
sklearn.manifold.TSNE

class sklearn.manifold. TSNE (*n_components=2, perplexity=30.0, early_exaggeration=4.0, learning_rate=1000.0, n_iter=1000, n_iter_without_progress=30, min_grad_norm=1e-07, metric='euclidean', init='random', verbose=0, random_state=None, method='barnes_hut', angle=0.5*) ¶ [source]

* L.J.P. van der Maaten and G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research* 9(Nov), 2008
** L.J.P. van der Maaten. Accelerating t-SNE using Tree-Based Algorithms. *Journal of Machine Learning Research* 15(Oct):3221-3245, 2014

t-Stochastic Neighbor Embedding (t-SNE)

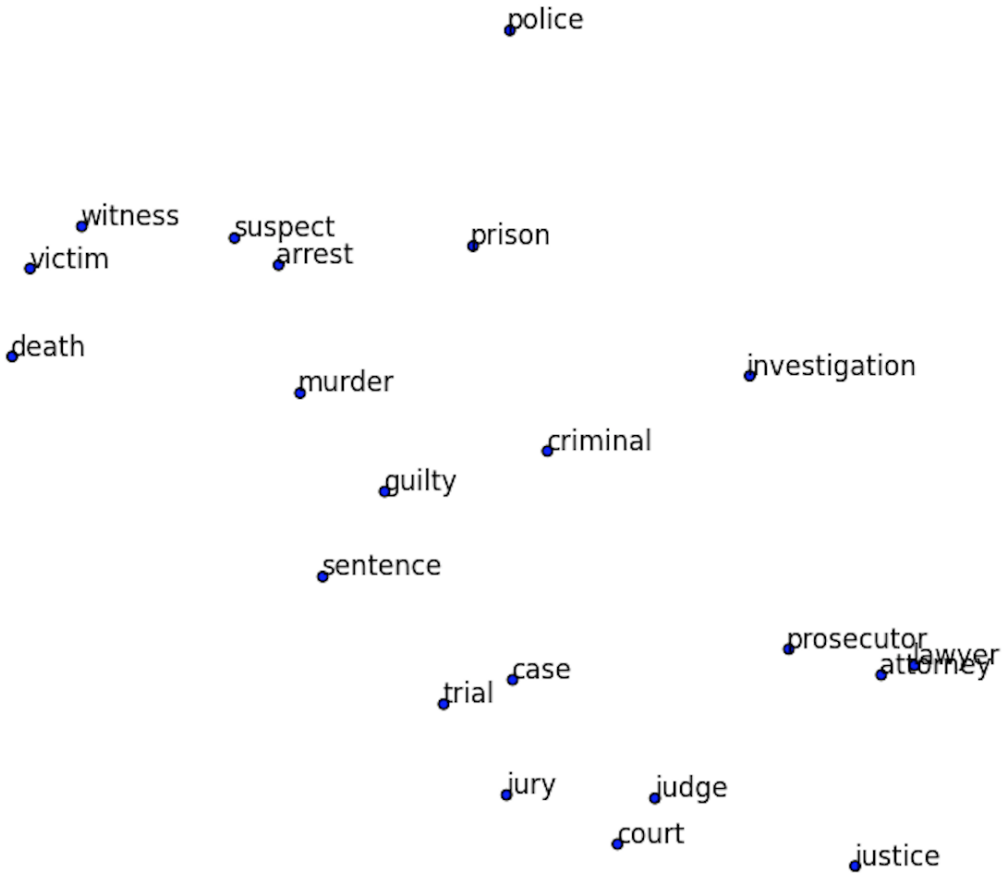
- 손글씨 숫자 데이터 (MNIST)의 시각화 예시



* Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(2579-2605), 85.

t-Stochastic Neighbor Embedding (t-SNE)

- 최근 Word2Vec과 같은 word embedding (고차원 벡터) 학습 결과의 시각화 방법으로 자주 이용됨



* http://nlp.yvespeirsman.be/blog/visualizing-word-embeddings-with-tsne/