

Document clustering

Hyunjoong Kim

soy.lovit@gmail.com

github.com/lovit

Clustering

- 군집화는 데이터에서 비슷한 객체들을 하나의 그룹으로 묶습니다.
 - 각 객체들이 어떤 군집으로 할당되어야 하는지에 대한 정답 정보 (y) 가 없기 때문에 unsupervised 알고리즘으로 분류됩니다.
 - 군집화 방법들은 각 객체들의 유사도(거리) 정보를 이용합니다.
유사한 객체를 하나의 군집으로 묶습니다.

Clustering

- 좋은 군집에 대한 기준은 다양하지만 공통적으로
“군집 내 객체들은 비슷하며, 군집 간 객체들은 이질적”임을 추구합니다.

Clustering

- 객체의 표현과 거리를 이용하는 방식에 따라 다양한 방법이 있습니다.
 - centroids models
 - connectivity models
 - density models
 - graph based models
 - ...

Clustering

- 그 전에 근본적으로 데이터의 representation 과 dissimilarity measure 가 잘 정의되어야 합니다.

k-means

k -means clustering

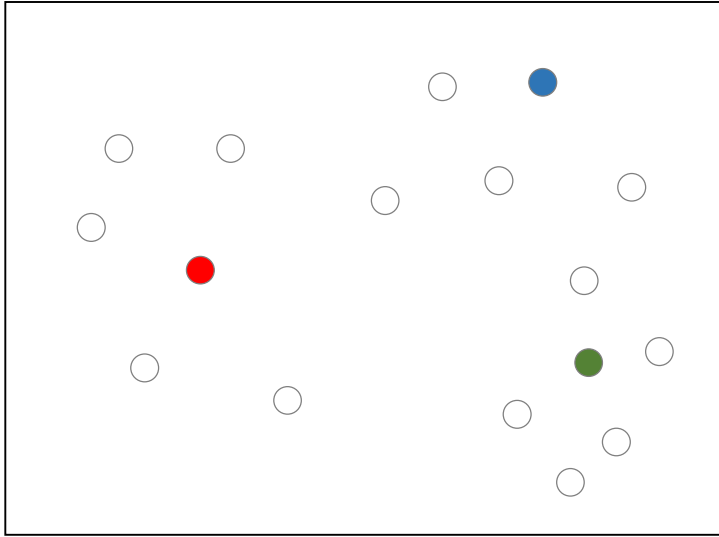
- 유사도

- n 개의 데이터 X 에 대하여 두 데이터 x_i, x_j 간에 정의되는 임의의 거리 (x_i, x_j)
 - 유클리디언, 코사인 등 벡터에서 정의되는 모든 거리 척도

- 그룹화의 방식

- 그룹의 개수는 k 개라고 가정
- 각 그룹을 centroid vector (평균 벡터)로 표현한 뒤, 이를 업데이트

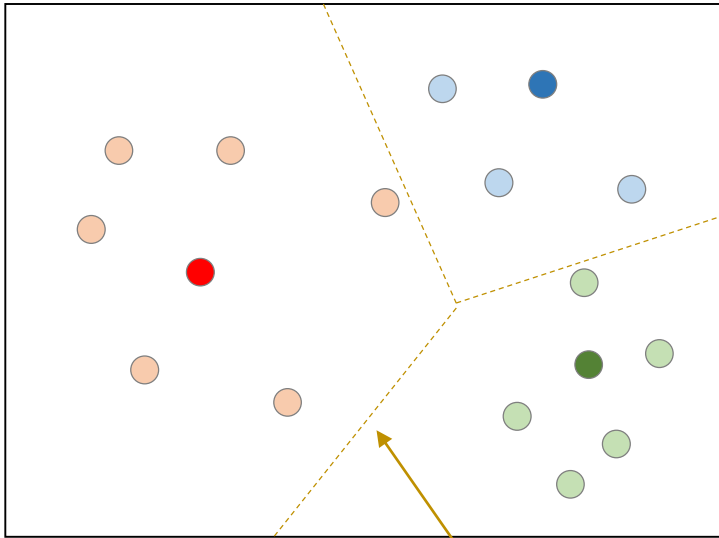
k -means clustering



1. Initialize

$k=3$ 이라 가정하면 3개의 점을 임의로 선택

k -means clustering



1. Initialize

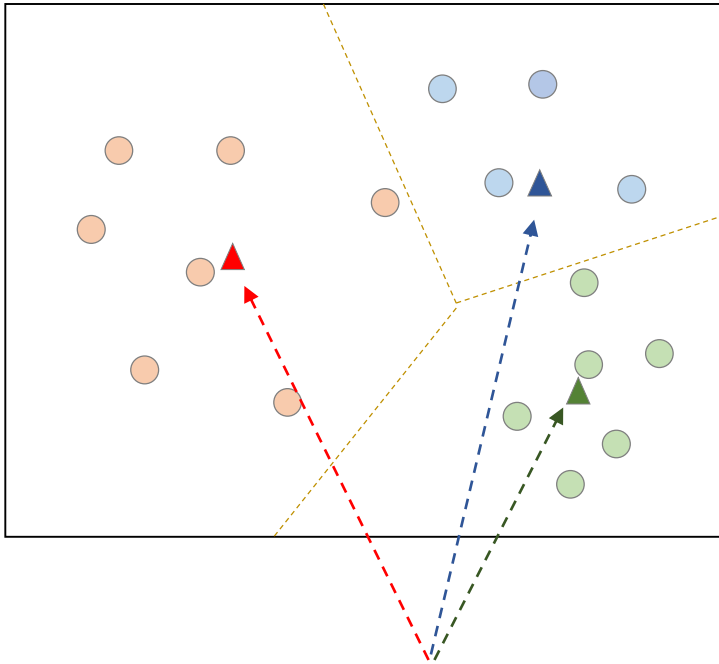
$k=3$ 이라 가정하면 3개의 점을 임의로 선택

2. Assign (epoch=0)

모든 점을 k 개의 centroid 중 가장 가까운 점의 색깔(label)로 할당

k 개의 centroids에 의하여 분할된 공간의 경계면으로, Voronoi partition, Voronoi diagram이라 부름

k -means clustering



데이터에는 존재하지 않는 가상의 centroids

1. Initialize

$k=3$ 이라 가정하면 3개의 점을 임의로 선택

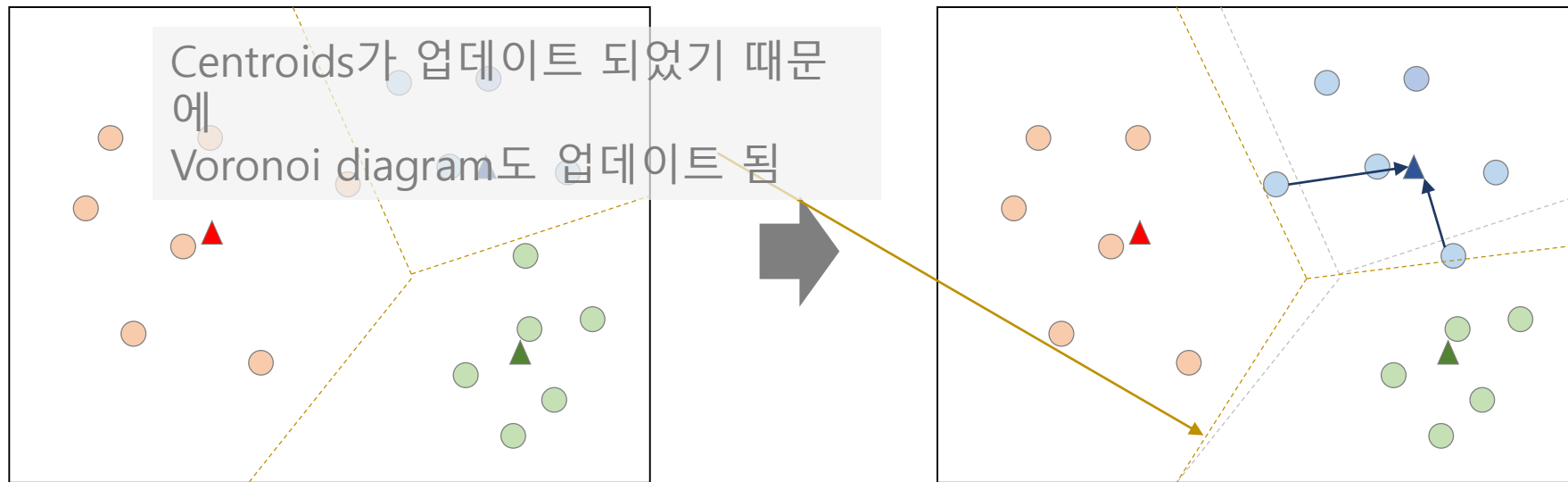
2. Assign (epoch=0)

모든 점을 k 개의 centroid 중 가장 가까운 점의 색깔(label)로 할당

3. Update centroid (epoch=0)

같은 색깔(label) 점들의 평균값을 가상의 centroids로 설정

k -means clustering



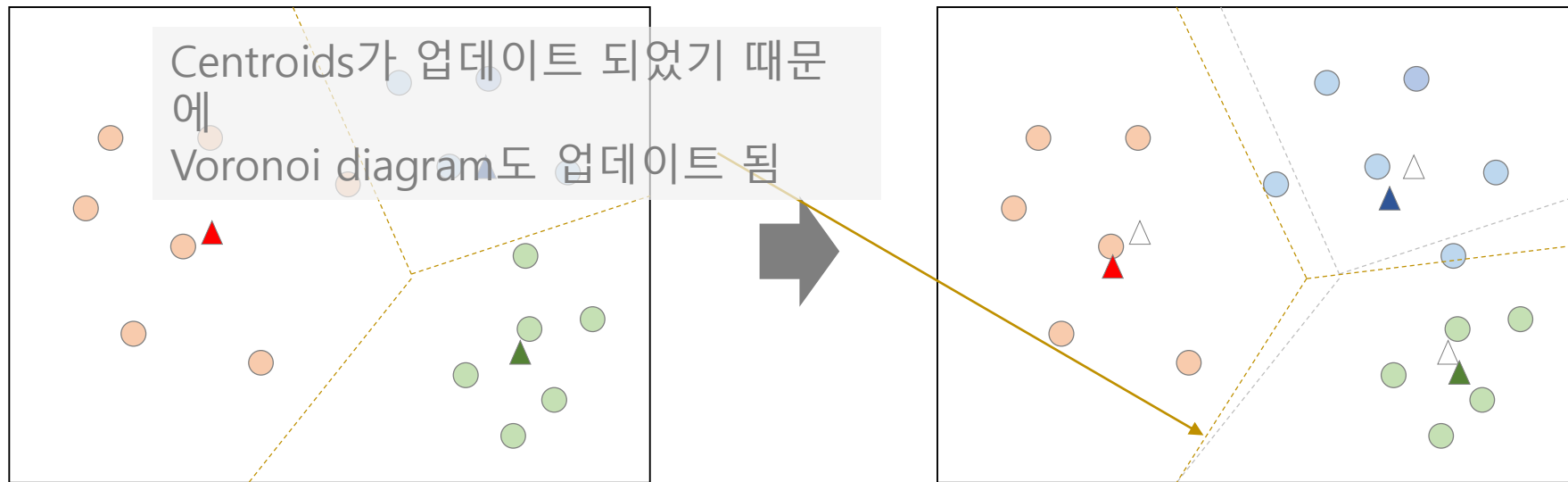
1. Initialize

$k=3$ 이라 가정하면 3개의 점을 임의로 선택

2. Assign (epoch=1)

모든 점을 업데이트 된 centroids 중 가장 가까운 점으로 할당

k -means clustering



1. Initialize

$k=3$ 이라 가정하면 3개의 점을 임의로 선택

2. Assign (epoch=1)

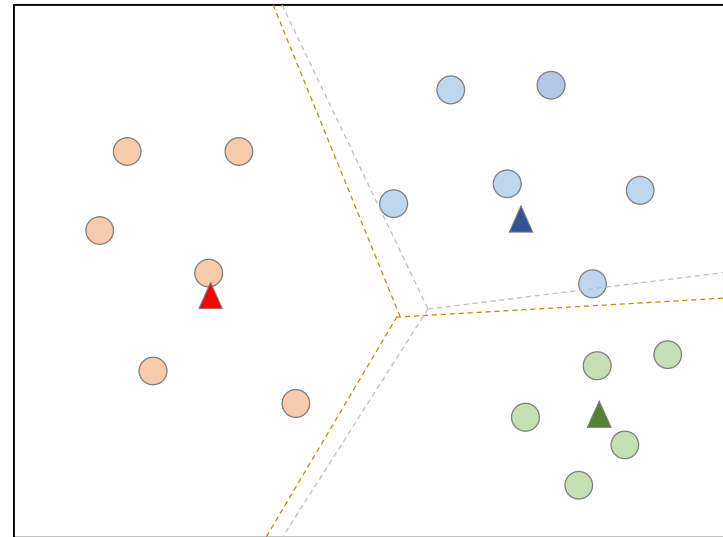
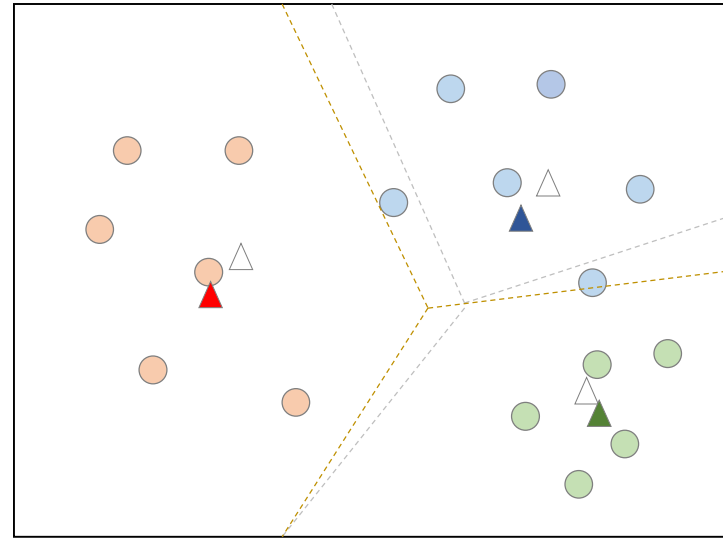
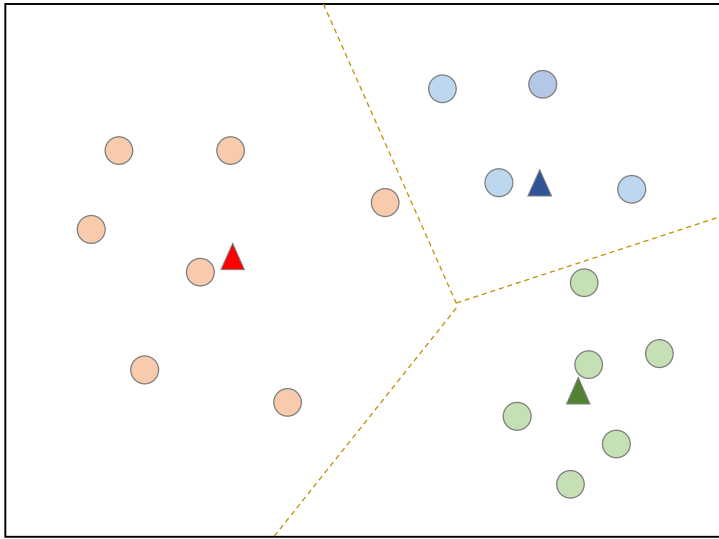
모든 점을 업데이트 된 centroids 중 가장 가까운 점으로 할당

알고리즘이 종료 될
때까지 2, 3을 반복

3. Update centroid (epoch=1)

색깔이 바뀐 점이 있기 때문에 Centroid를 다시 업데이트

k -means clustering



1. Initialize

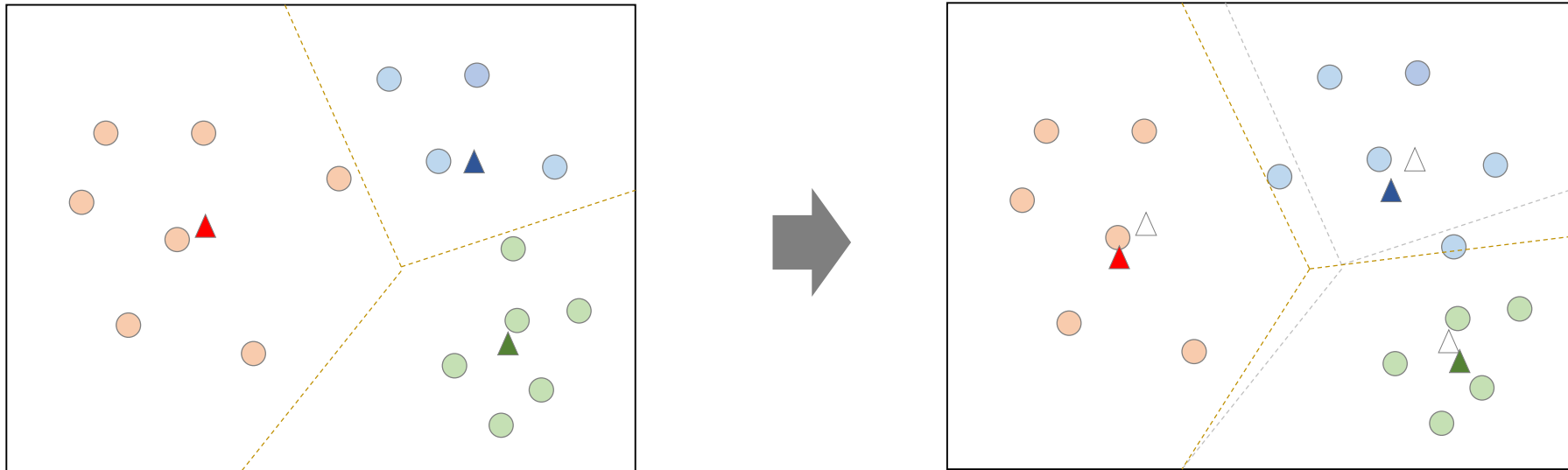
$k=3$ 이라 가정하면 3개의 점을 임의로 선택

2. Assign (epoch=2)

모든 점을 가장 가까운 centroids로 할당하여도 색깔이 변하지 않으므로 알고리즘 종료

Centroids based clustering

- 우리에게 익숙한 k -means 은 local optimal 을 찾는 heuristic 입니다 [1].
 - Lloyd k -means 는 수렴할 때까지 반복적으로 centroids를 업데이트 합니다.



k -means

- (Lloyd) k -means 는 빠릅니다.
 - 계산복잡도가 작습니다. $O(n * i * k)$
 - pairwise distance 를 요구하지 않기 때문에 대량의 데이터에 적합하며,
 - row 단위로 학습하기 때문에 mini-batch / 분산환경의 구현이 쉽습니다.
- 대량의 문서 집합의 군집화에는 가장 현실적인 방법입니다.

k -means

- 문서 군집화는 거리 척도가 중요합니다.
 - 일반적으로 k -means 는 Euclidean distance 를 이용합니다.
 - 문서 간 유사도는 두 문서의 공통된 단어 유무가 제일 중요한 정보입니다.
 - Euclidean distance 는 이를 고려하지 않습니다.
 - Sparse vector 형식으로 문서를 표현할 경우에는 Jaccard, Pearson, Cosine 을 쓸 수 있지만, Euclidean 만은 쓰지 말아야 합니다 [1].
 - Jaccard, Pearson, Cosine 모두 문서 벡터의 방향성에 관련된 척도입니다.

Spherical k -means

- Cosine distance 를 이용하는 k -means 를 Spherical k -means 라 합니다 [1].
 - Distance measure 외의 학습 방법은 Lloyd 와 같습니다만,
 - 이 차이로 결과는 확연히 다릅니다.
- scikit-learn 에는 metric 이 Euclidean 으로 고정되어 있습니다.

Spherical k -means

- Spherical 과 Euclidean 의 차이는 centroids normalize 방식입니다.
 - Lloyd 에서는 $c_j = \frac{1}{n_j} \sum_{x_i \in C_j} x_i$ 로 정의합니다.
 - x_i 의 norm 의 영향을 받습니다.
 - centroids 에 2-norm 크기 성분이 있습니다.
 - Spherical 에서는 $c_j = \frac{\sum_{x_i \in C_j} x_i}{\left| \sum_{x_i \in C_j} x_i \right|}$, 2-norm normalize 를 합니다.

Spherical k -means

- Spherical 과 Euclidean 의 차이는 centroids normalize 방식입니다.

- Spherical 에서는 $c_j = \frac{\sum_{x_i \in C_j} x_i}{|\sum_{x_i \in C_j} x_i|}$, 2-norm normalize 를 합니다.

- Centroids 에 방향 성분만 남습니다.

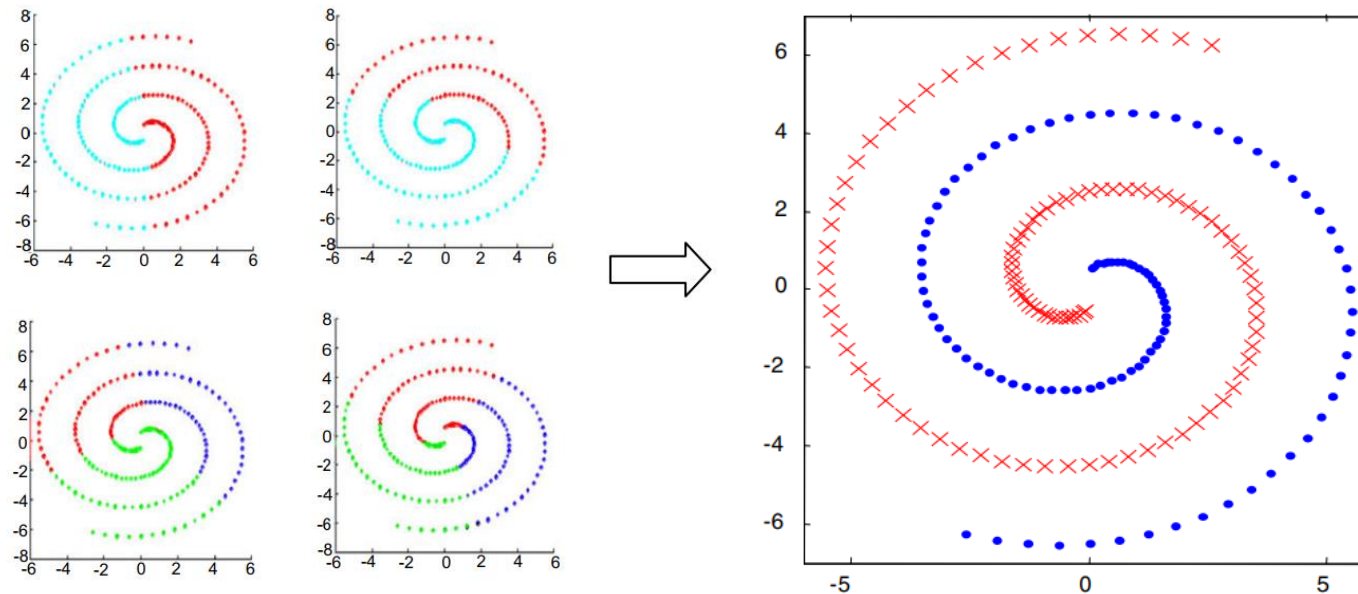
- $|x - c|_2^2 = \langle x, x \rangle - 2 \langle x, c \rangle + \langle c, c \rangle$ 이기 때문에 x, c 가 unit vector 이면 Euclidean 과 Cosine distance 의 거리 순서는 동일합니다.

Spherical k -means

- k -means 계열 알고리즘들은 알려진 단점들이 있습니다.
 1. 군집의 모양은 centroid 를 중심으로 한 구형을 가정합니다
(Voronoi diagram)
 2. Initial points 에 따라 군집의 모양이 달라질 수 있습니다.
 3. 적절한 군집의 개수는 사용자가 정의해야 합니다.
 4. 노이즈 데이터에 민감할 수 있습니다.

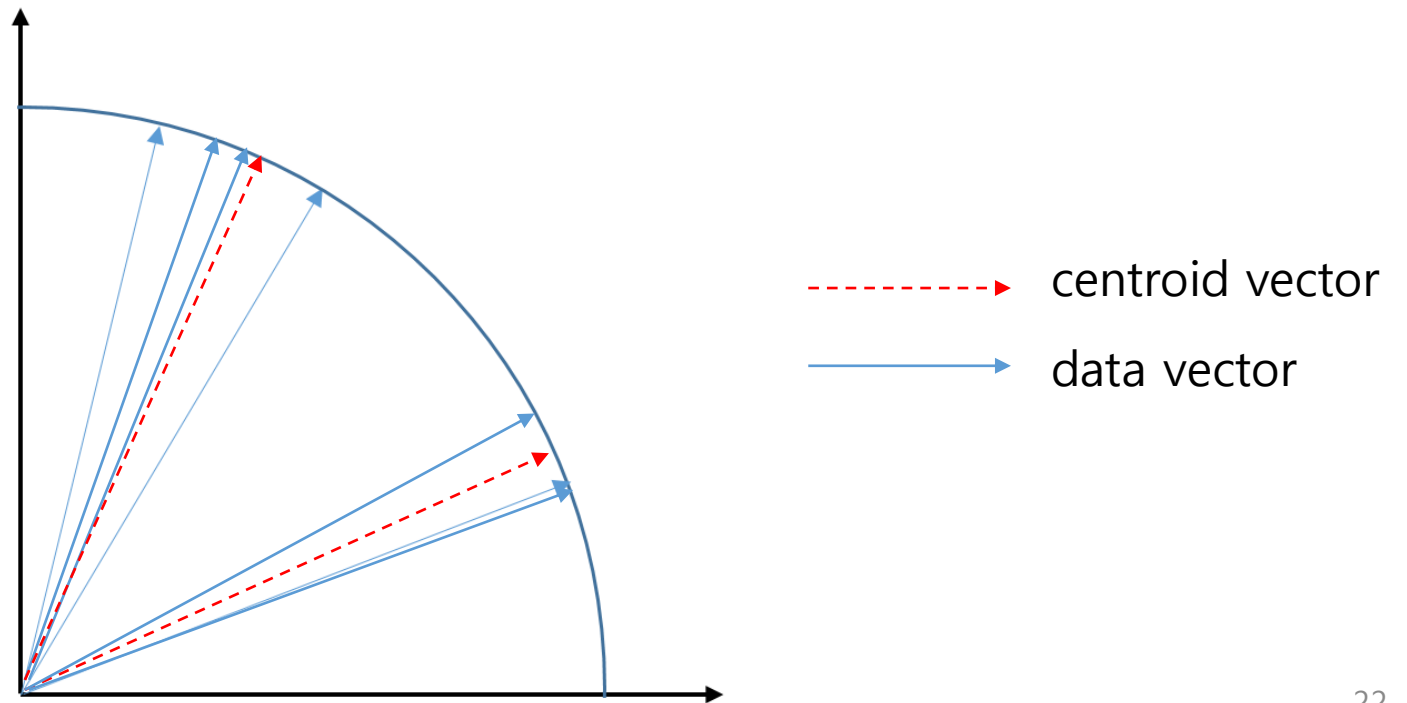
Limitations 1. Ball-shape

- k -means 는 Voronoi diagram 을 만듭니다.
 - 군집의 모습이 구형이 아닌 경우에 취약합니다.
 - k -means ensembles 은 이를 해결할 수 있습니다 [1]



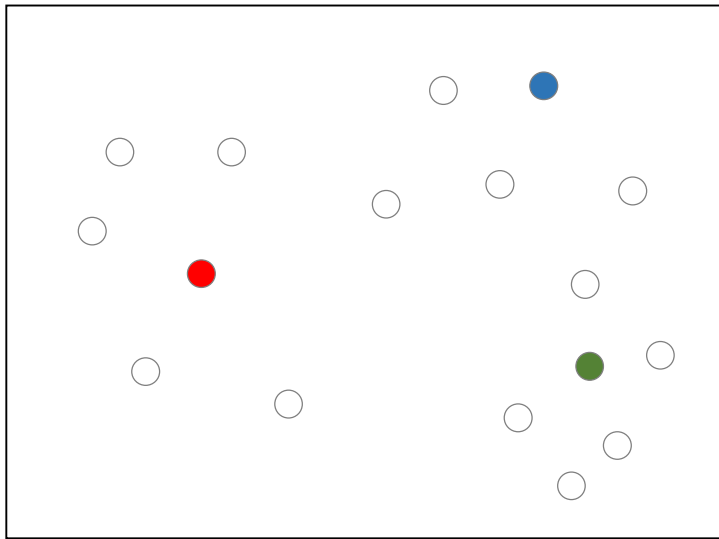
Solutions 1. Ball-shape

- 하지만 term frequency 로 문서를 표현하며, Cosine distance 를 이용하면, k -means 가 적용되기 어려운 공간이 아닙니다.
 - 각도 기준으로의 partitioning 입니다.

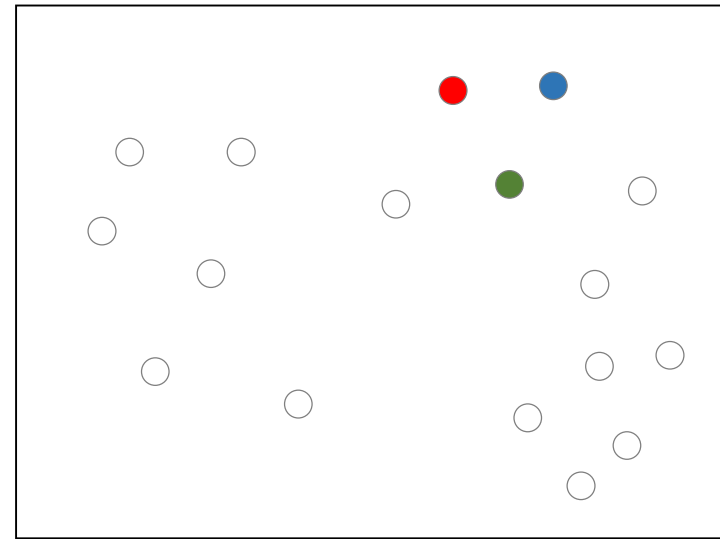


Limitations 2. Unstable initial points

- k -means 는 initial points 만 잘 뽑아도 수렴이 빠르다고 알려져 있으며,
 - 사실, 최악의 initialization 만 아니면 수렴은 원래 빠릅니다.
 - 특정 지역에 initial points 가 모여있지만 않아도 됩니다.



Ideal initialization



Worst initialization

Limitations 2. Unstable initial points

- k -means++^[1] 은 가장 널리 알려진 initialization method 입니다.
 - 지금의 initial point c_{t-1} 에서 먼 점을 우선적으로 선택합니다.

1. Select a point c_0 randomly
2. Select next point c_t with prob. $\frac{d(c_{t-1}, c_t)^2}{\sum_i d(c_{t-1}, c_i)^2}$
3. Repeat step 2 until choosing k points

[1] Arthur, D., & Vassilvitskii, S. (2007, January). k -means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 1027-1035). Society for Industrial and Applied Mathematics.

Limitations 2. Unstable initial points

- k -means++^[1] 은 가장 널리 알려진 initialization method 입니다.
 - scikit-learn 에 구현되어 있습니다.

[sklearn.cluster](#).KMeans ¶

```
class sklearn.cluster.KMeans(n_clusters=8, init='k-means++', n_init=10, max_iter=300, tol=0.0001,  
precompute_distances='auto', verbose=0, random_state=None, copy_x=True, n_jobs=1, algorithm='auto')
```

[\[source\]](#)

Limitations 2. Unstable initial points

- k -means++^[1] 은 고차원 데이터에는 의미가 없습니다.
 - 고차원 데이터에서는 가까운 점의 거리는 $d(x_i, x_j) \cong 0$ 이지만,
 - 조금만 멀어져도 대부분의 점들 간 거리의 값이 비슷합니다.
 - k -means++ 은 저차원 데이터에 적합합니다.
- 특히 sparse data + Cosine 에서는 대부분의 거리가 1 입니다.
 - 비싼 random sampling

[1] Arthur, D., & Vassilvitskii, S. (2007, January). k -means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 1027-1035). Society for Industrial and Applied Mathematics.

Limitations 2. Unstable initial points

- 하루의 뉴스 데이터 (30,091 개)의 pairwise distance 예시
 - 고차원의 데이터에서는 비슷하다는 것 외의 거리는 의미가 없습니다.

Cosine distance	Percentage
0.00 ~ 0.10	0.31%
0.10 ~ 0.20	0.31%
0.20 ~ 0.30	0.32%
0.30 ~ 0.40	0.19%
0.40 ~ 0.50	0.30%
0.50 ~ 0.60	0.32%
0.60 ~ 0.70	0.54%
0.70 ~ 0.80	2.11%
0.80 ~ 0.90	10.22%
0.90 ~ 1.00	85.39%

Limitations 2. Unstable initial points

- 고차원 벡터에서는 가깝다라는 의미는 있지만, 멀다라는 의미가 없습니다.

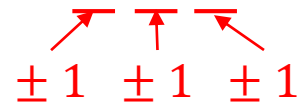
Assume that $|x-y|^2 = 1$ and x, y are integer vector

1차원 $X = [1], y = [0] \text{ or } [2]$

2차원 $X = [1, 0], y = [1, 1], [1, -1], [0, 0], [2, 0]$

3차원 $X = [1, 0, 0], y = [1, 0, 0]$

$\pm 1 \quad \pm 1 \quad \pm 1$



Beat “ k -means++”

- Initial points 는 거의 비슷한 점만 아니면 충분히 괜찮으며, k 개의 points 를 선택하기 위하여 $n * k$ 번의 계산은 불필요합니다.
- Term frequency representation 의 특징을 이용하면 널리 퍼진 initial points 를 빠르게 찾을 수 있습니다.

Beat " k -means++"

1. $\alpha * k$ 개의 후보를 random sampling $D_{init} \subset D$
2. D_{init} 에서 한 개의 점 c_i 을 임의로 선택.
3. D_{init} 에서 c_i 와 Cosine similarity 가 t_{init} 보다 큰 점을 D_{init} 에서 삭제
4. D_{init} 이 공집합이 아니면 k 개의 점을 뽑을 때까지 step 2 – 3 반복
5. D_{init} 이 공집합이며, 현재까지 선택한 점의 개수, k_0 가 $k_0 < k$ 이면
 $k - k_0$ 개의 점을 $D - D_{init}$ 에서 임의로 추출

Beat " k -means++"

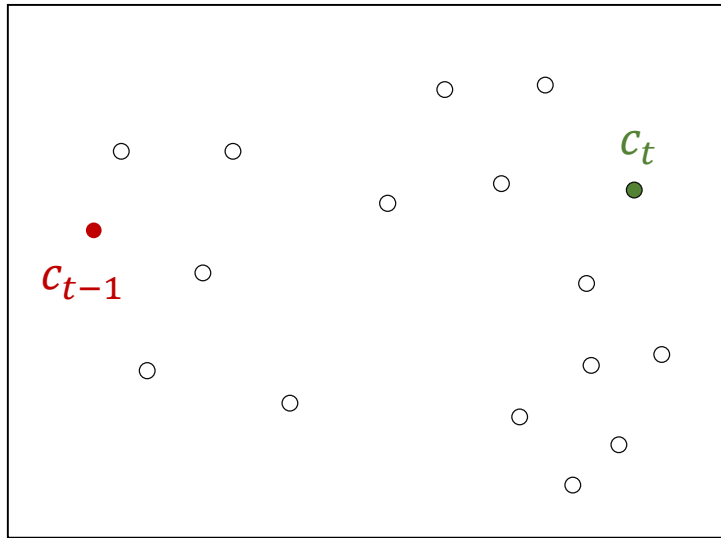
- Sparse vector 로 표현된 문서간 cosine 은 0.5 넘기가 어렵습니다.

$$\rightarrow t_{init} = 0.5$$

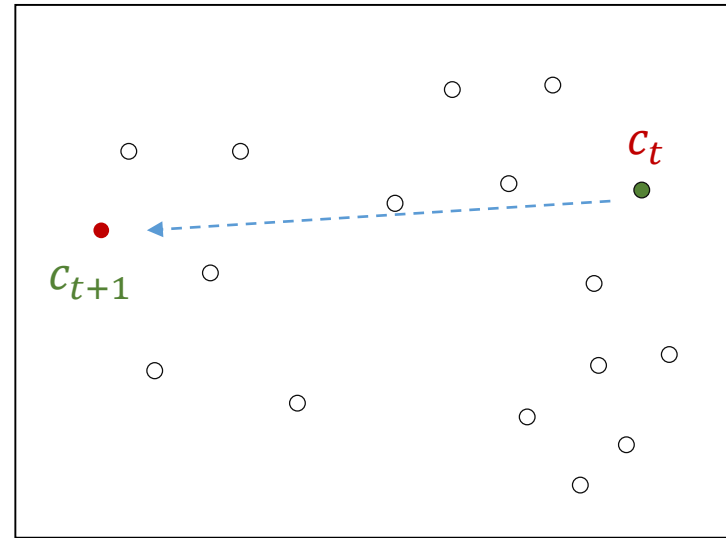
- D_{init} 안에서 $\cos(c_i, D_{init})$ 을 계산하기 때문에 최대 $\alpha * k^2$ 계산합니다.
 - $\alpha * k^2 \ll n * k$

Beat "k-means++"

- k -means++ 은 중복된 점이 추출될 위험이 있습니다.
- $p(c_t|c_{t-1}) = \frac{d(c_{t-1}, c_t)^2}{\sum_{t'} d(c_{t-1}, c_{t'})^2}$ 는 c_{t-1} 와 가까운 점을 다음 round 에 선택하지 않도록 할 뿐, p round 이후에도 선택되는 걸 방지하진 못합니다.



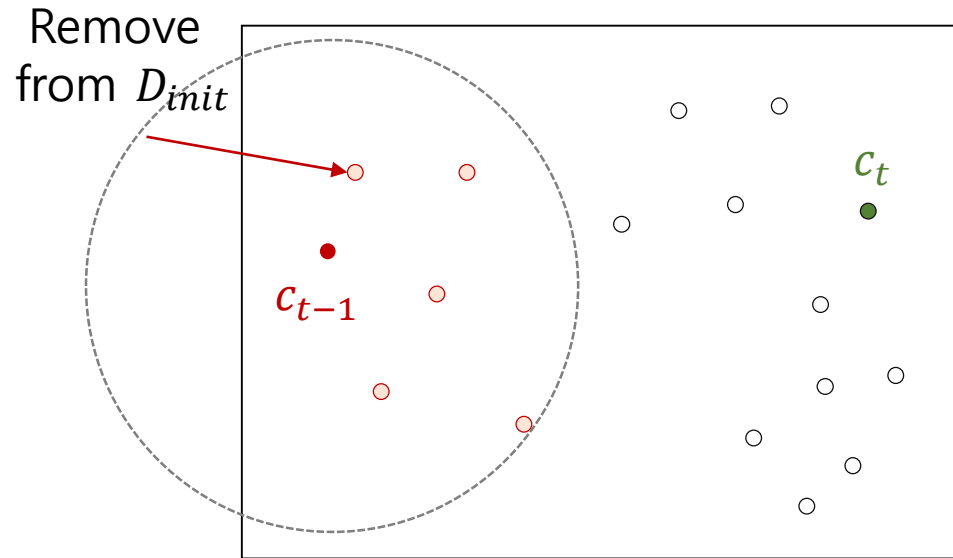
c_{t-1} 에서 먼 점 중 하나는 c_t



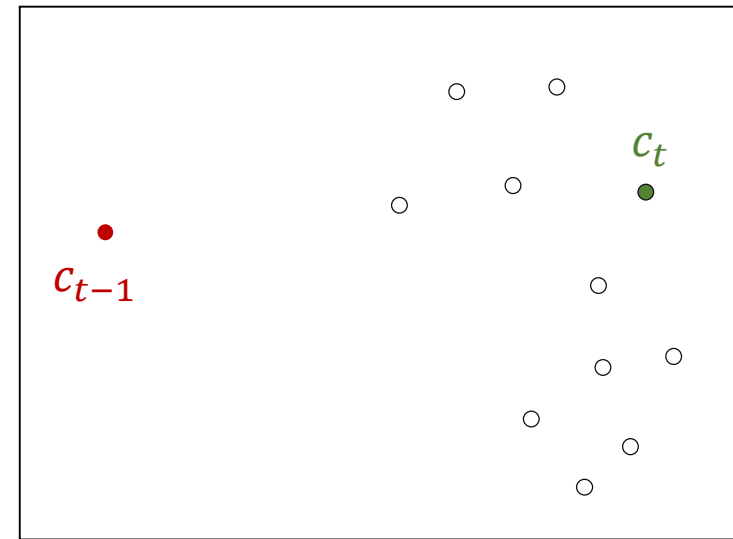
c_t 에서 먼 점은 $c_{t-1} = c_{t+1}$
일 수 있음

Beat " k -means++"

- D_{init} 에서 한 번 선택된 점 주위의 다른 점들을 모두 제거하면 중복되거나, 근처의 점이 선택될 위험이 적습니다.



c_{t-1} 에서 먼 점 중 하나는 c_t



$c_{t-1} \cong c_{t+1}$ 인 점은 선택되지 않음

Limitations 3. Defining k

- k -means 는 군집의 개수를 사용자가 직접 정의하여야 합니다.
- Silhouette 은 군집화 품질의 척도로, 적절한 k 를 정하는데 이용됩니다.

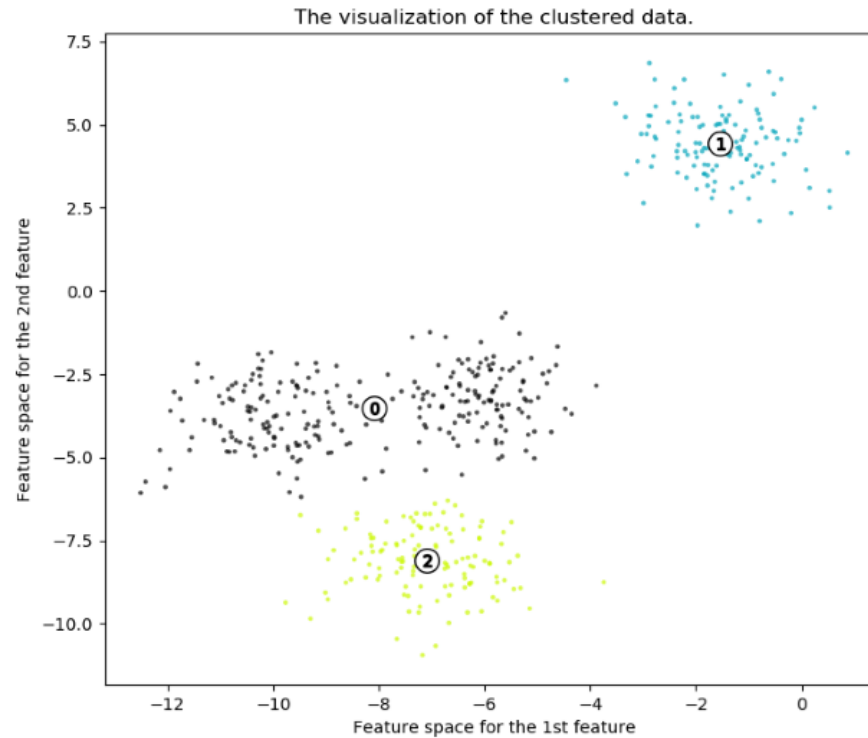
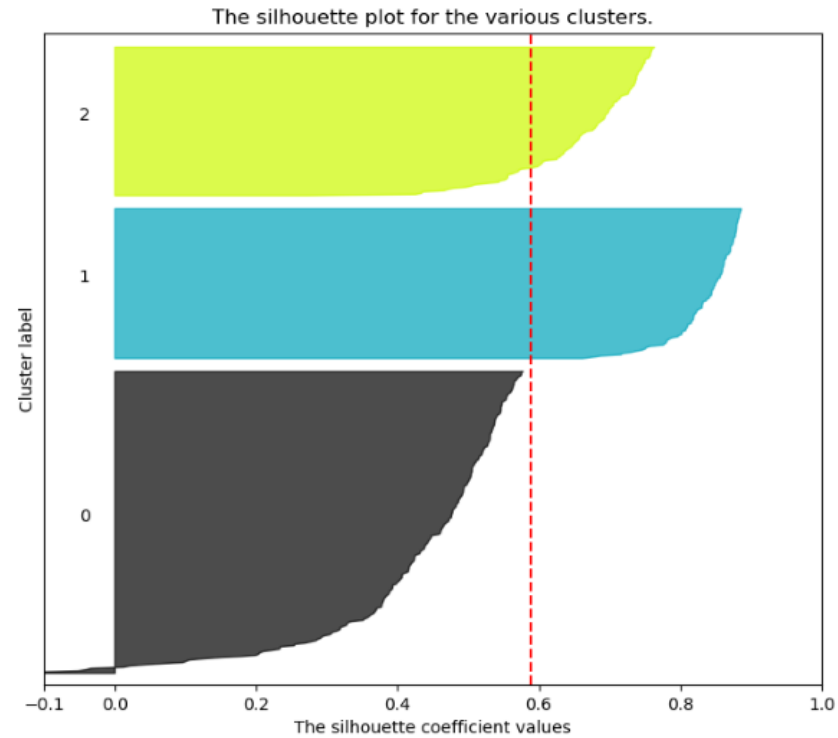
$$s(x) = \frac{b - a}{\max(a, b)}, \quad s(X) = \text{mean}(s(x))$$

a : mean distance between a sample and all other points in the same class

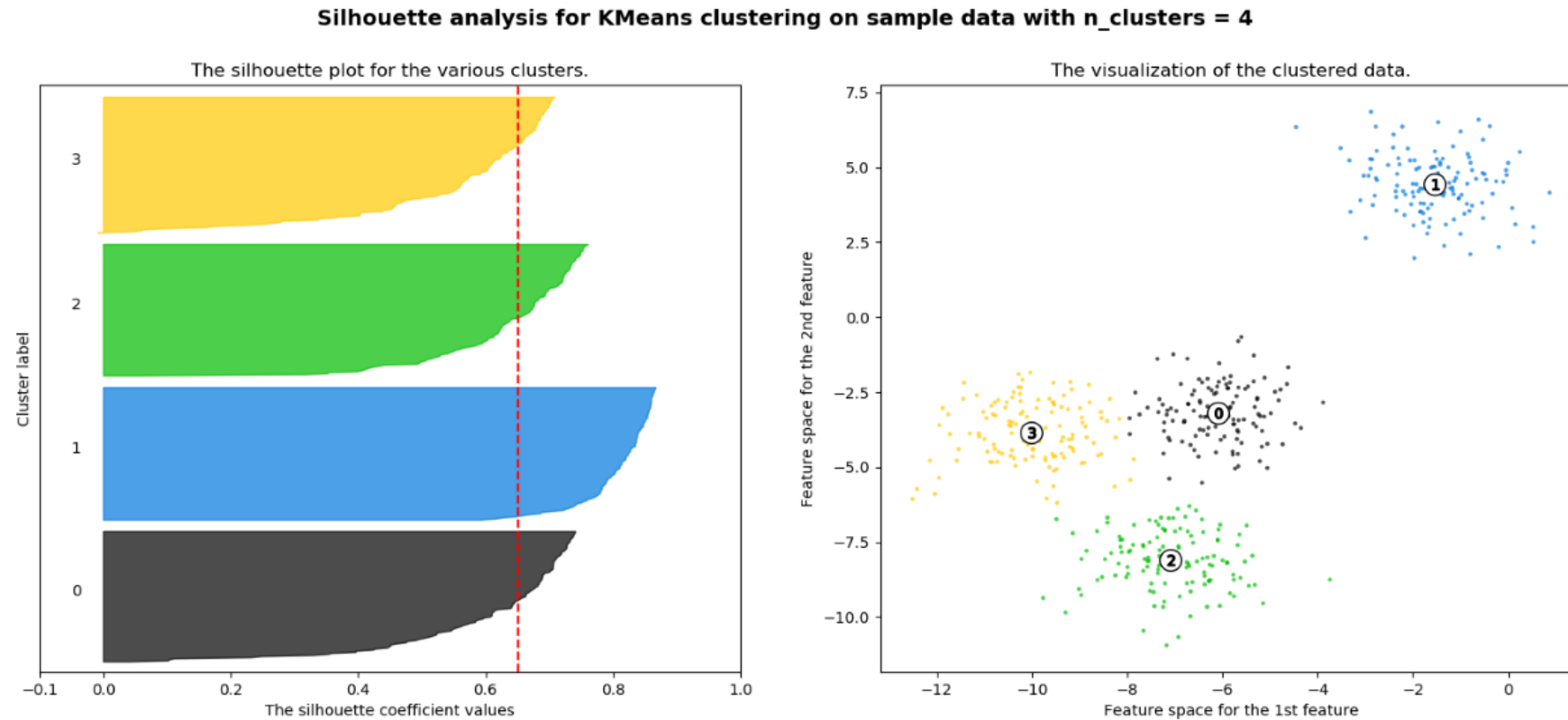
b : mean distance between a sample and all other points in the next nearest class

Limitations 3. Defining k

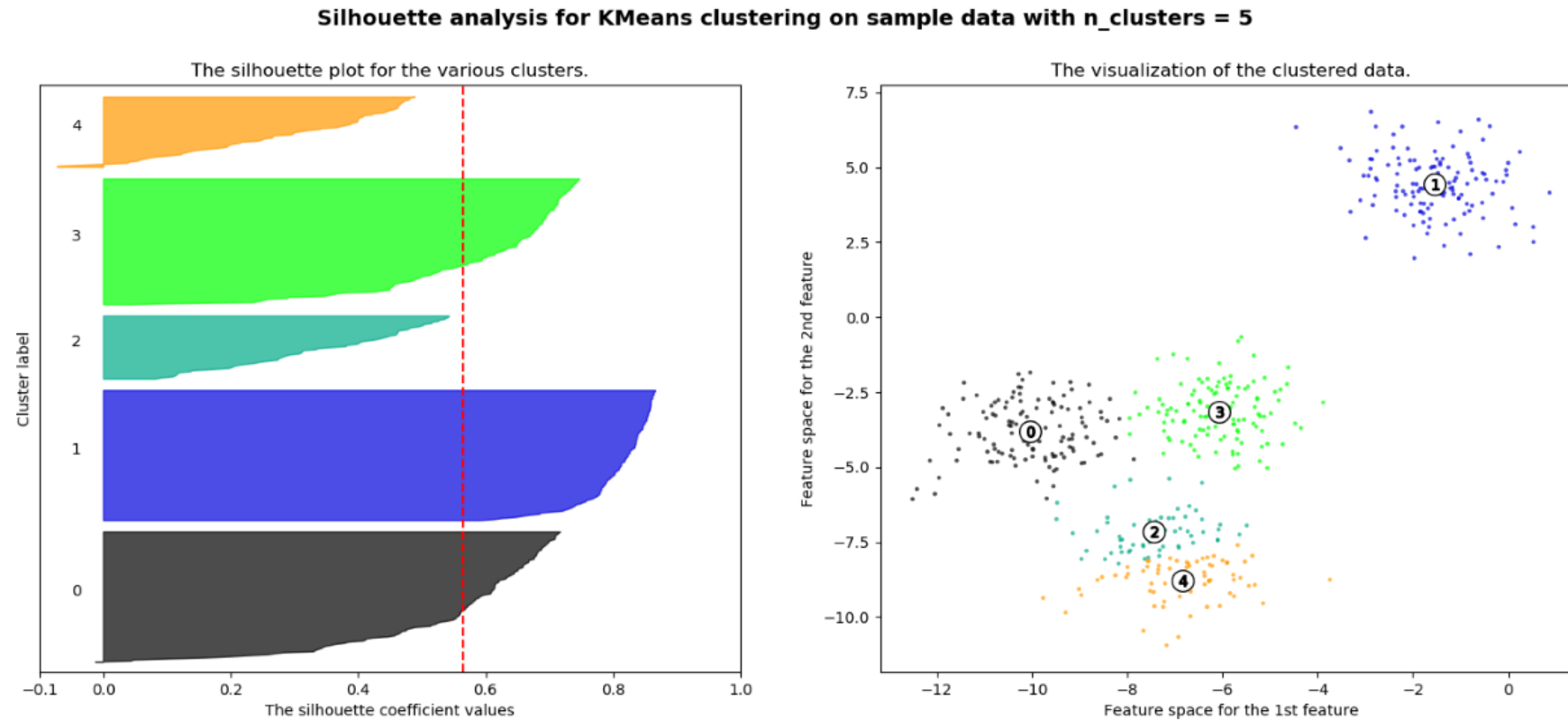
Silhouette analysis for KMeans clustering on sample data with $n_clusters = 3$



Limitations 3. Defining k



Limitations 3. Defining k



Limitations 3. Defining k

- Silhouette 은 사후 평가 방법입니다.
 - 적절한 k 를 찾았더라도, 그 군집이 최선이라고는 말하지 못합니다.
 - 모든 k 에 대하여 테스트 할 비용도 만만치 않습니다.

Limitations 3. Defining k

- Silhouette 은 model fitness measure 입니다.
 - 높은 값이 “우리가 예상하는” 좋은 군집화 결과를 의미하지는 않습니다.
 - 비슷하게, LDA 의 perplexity 역시 수리적으로는 설명력이 있지만, 현실적이지 않은 measure 라는 주장도 있습니다 [1].

Limitations 3. Defining k

- 고차원 데이터는 잘 작동하지 않을 수 있습니다.
 - 각 군집의 크기가 크다면, 멀리 떨어진 (=거리가 무의미한) 점들이 존재합니다.

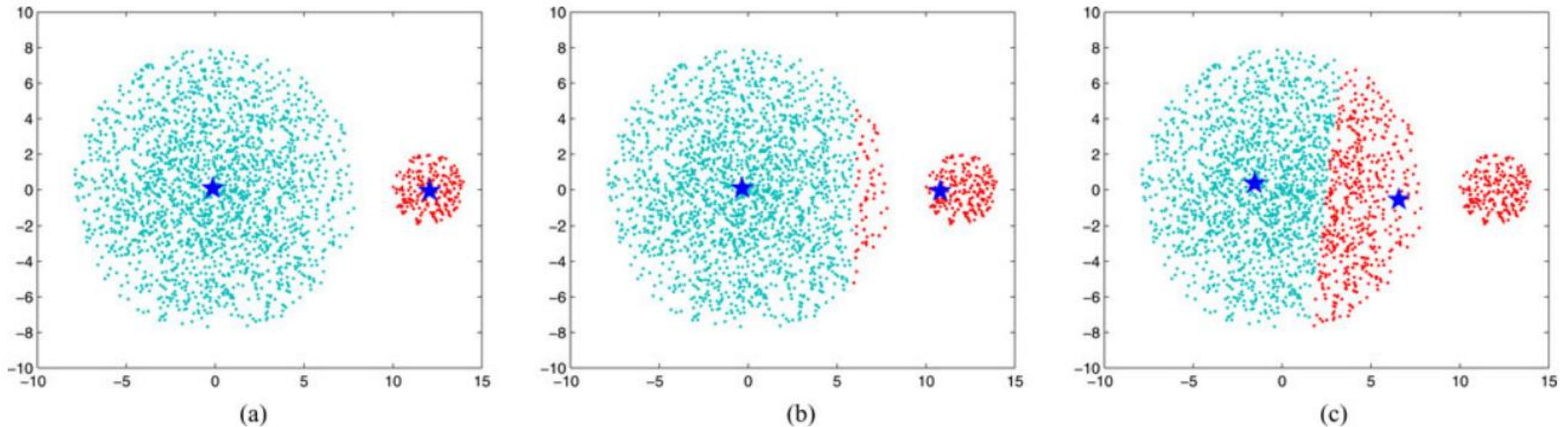
$$s(x) = \frac{b - a}{\max(a, b)}, \quad s(X) = \text{mean}(s(x))$$

a : mean distance between a sample and all other points in the same class

b : mean distance between a sample and all other points in the next nearest class

Limitations 3. Uniform effect

- Imbalanced class data 일 때, class distribution 이 잘 반영되지 않고, 모든 군집의 크기가 균일해지는 현상 [1]



Limitations 3. Uniform effect

- k -means type 은 몇 번의 반복으로 거의 수렴합니다.
 - 그렇다면 “repeat until converged” 동안의 학습은 더 좋은 걸까요?
 - Uniform effect 가 일어나는 과정일 수도 있습니다.

Solutions 3. Defining k

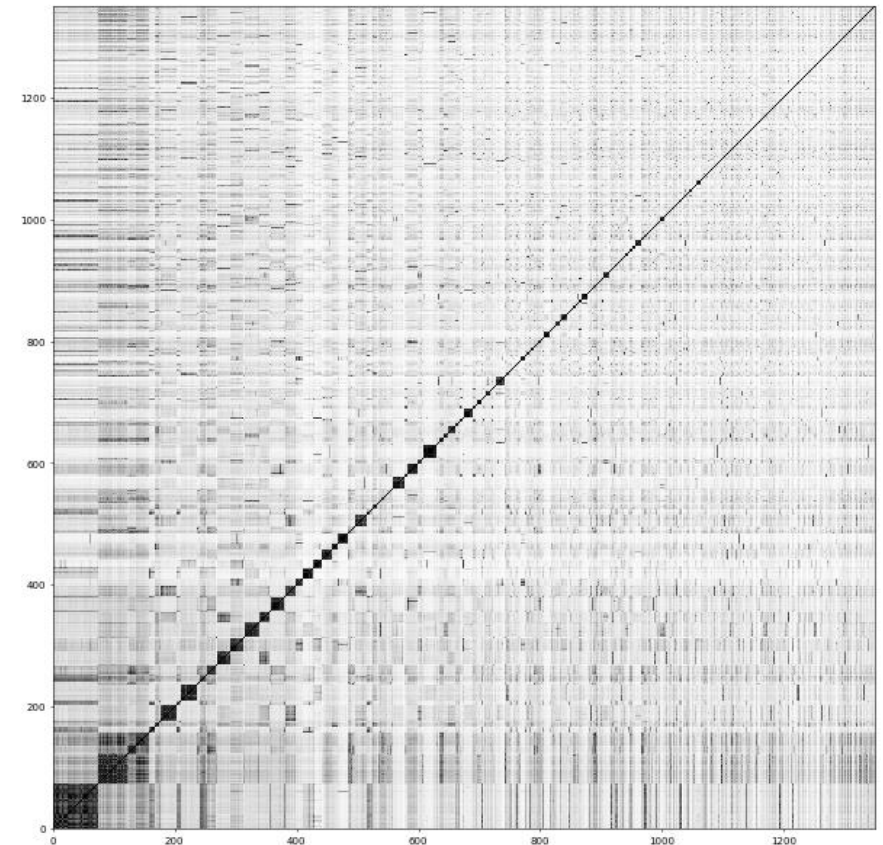
- 현실적인 미봉책은 예상하는 군집의 개수보다 크게 k 설정한 뒤, 후처리로 비슷한 군집을 병합합니다.
 - 학습 후, 하나의 군집이 여러개로 나뉘어 졌는지를 확인하기는 쉽습니다.
 - 하나의 군집에서 잘못된 점을 찾는 것이 더 어려우며, 그 점들의 후처리도 어렵습니다.

Solutions 3. Defining k

- 현실적인 미봉책은 예상하는 군집의 개수보다 크게 k 설정한 뒤, 후처리로 비슷한 군집을 병합합니다.
 - 실제 군집의 개수보다 k 가 크면 major 군집들이 여러 개로 나뉘어집니다.
 - k 가 작으면 minor 군집이 찢어질 가능성이 높습니다.
 - 이 때 centroids 는 major 편입니다.

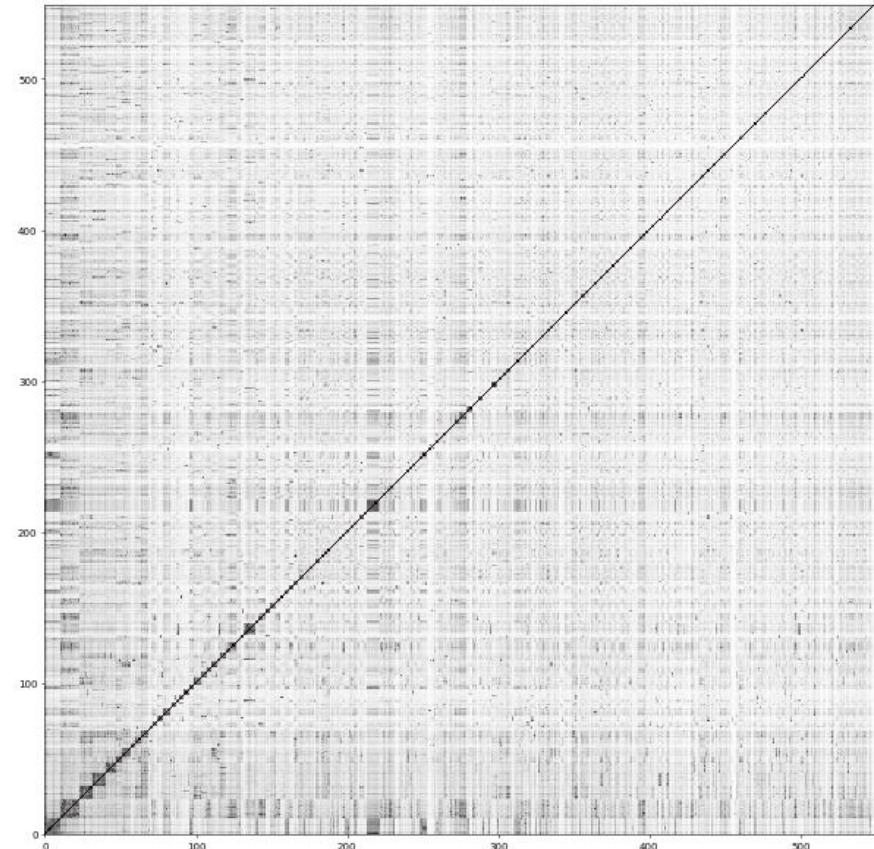
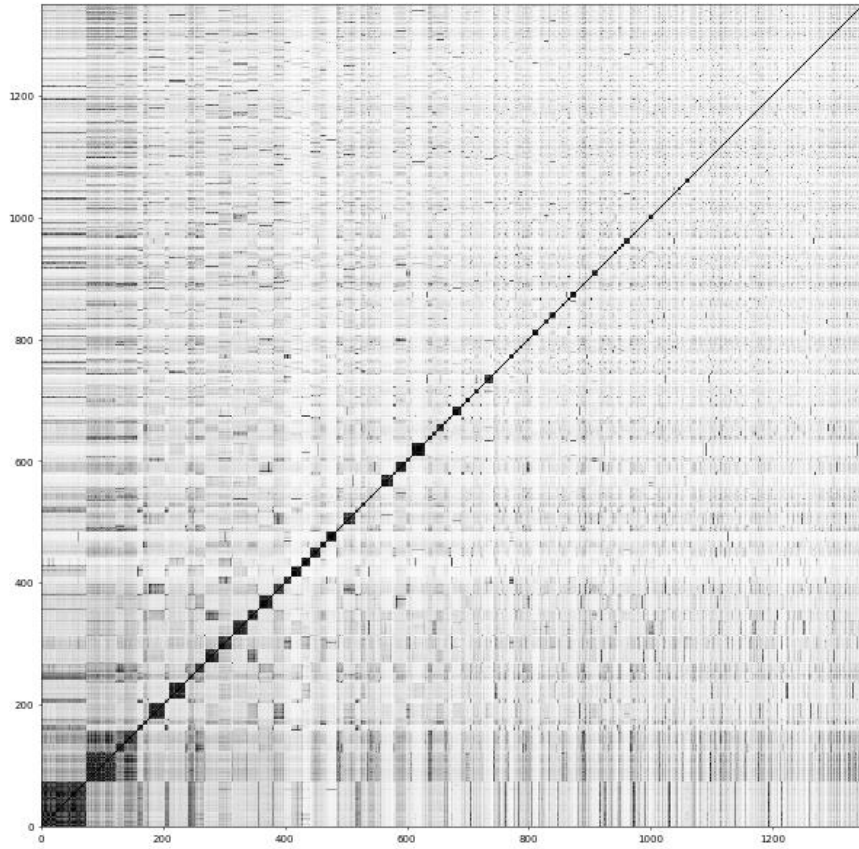
Solutions 3. Defining k

- Centroid vectors 의 pairwise distance matrix 는 군집화 결과의 직관적인 이해를 도와줍니다.
- Diagonal 만 진할수록, 각 cluster 는 separation 이 잘 이뤄진 것입니다.
- 진한 square 를 하나의 군집으로 묶을 수 있습니다.



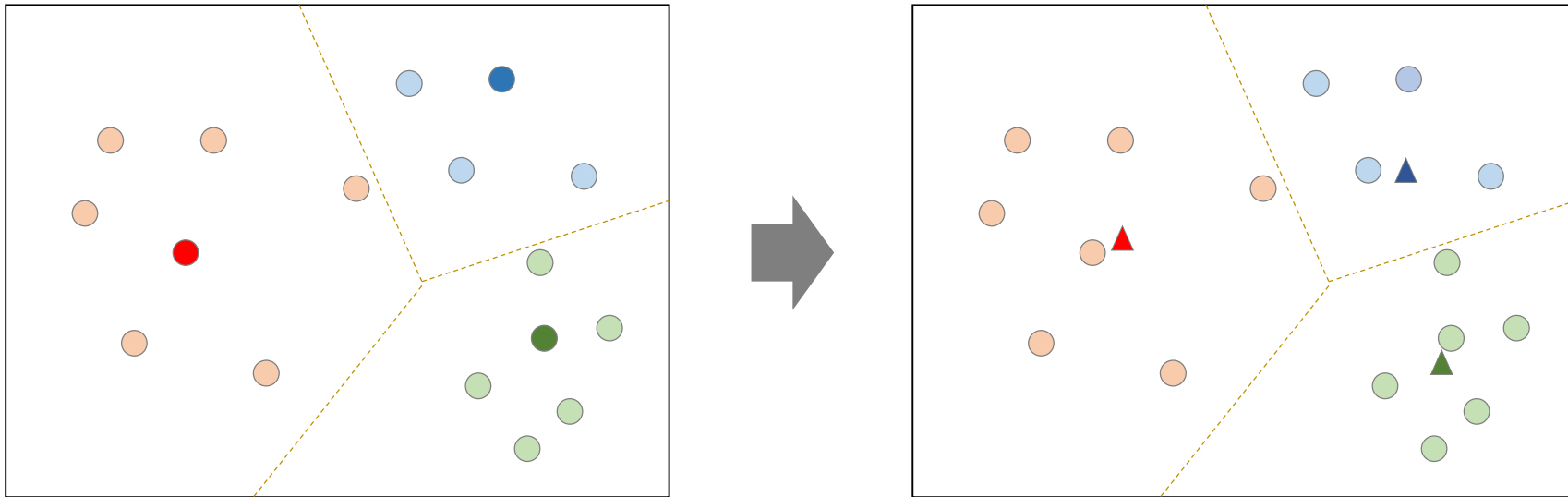
Solutions 3. Defining k

- term frequency vector 간의 cosine similarity $\geq t$ (eg 0.4) 를 하나의 군집으로 묶음으로서 손쉽게 후처리를 할 수 있습니다.



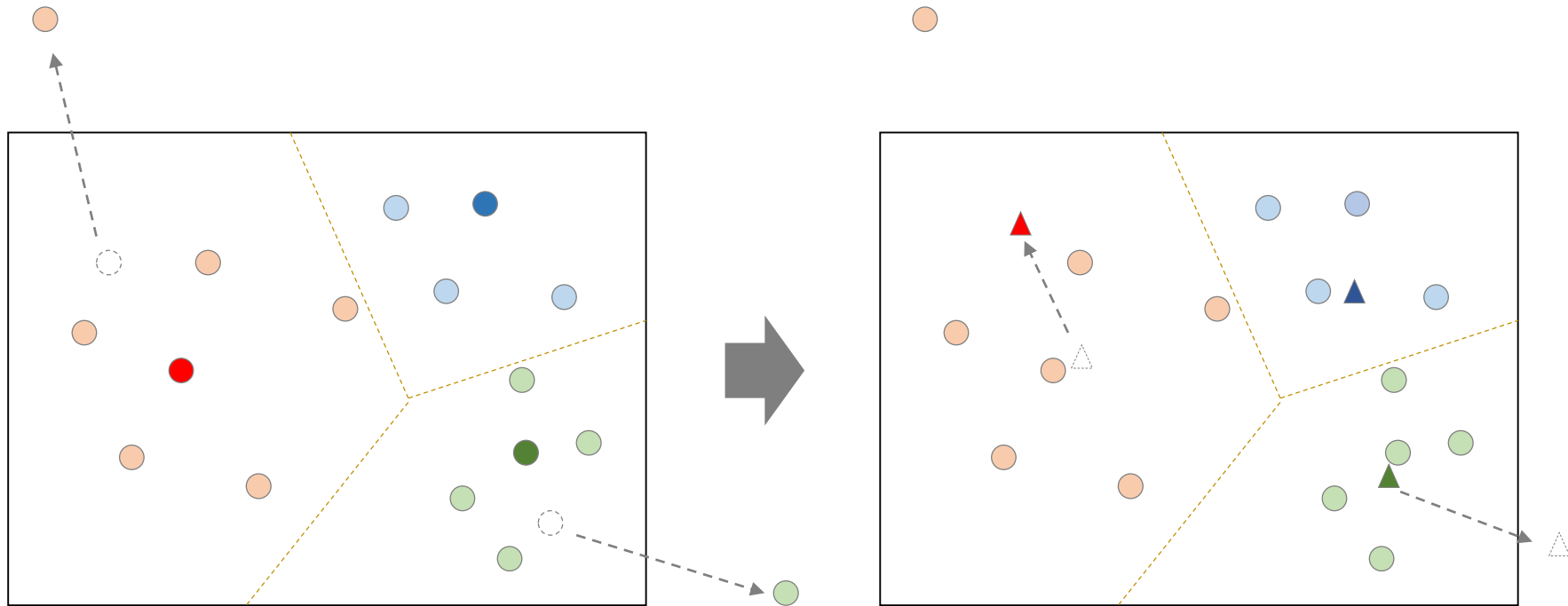
Limitations 4. Sensitive to noise points

- 모든 점을 반드시 한 개 이상의 군집으로 assign 하기 때문에, 일단 가장 가까운 군집에 할당되어 centroid 를 크게 움직입니다.



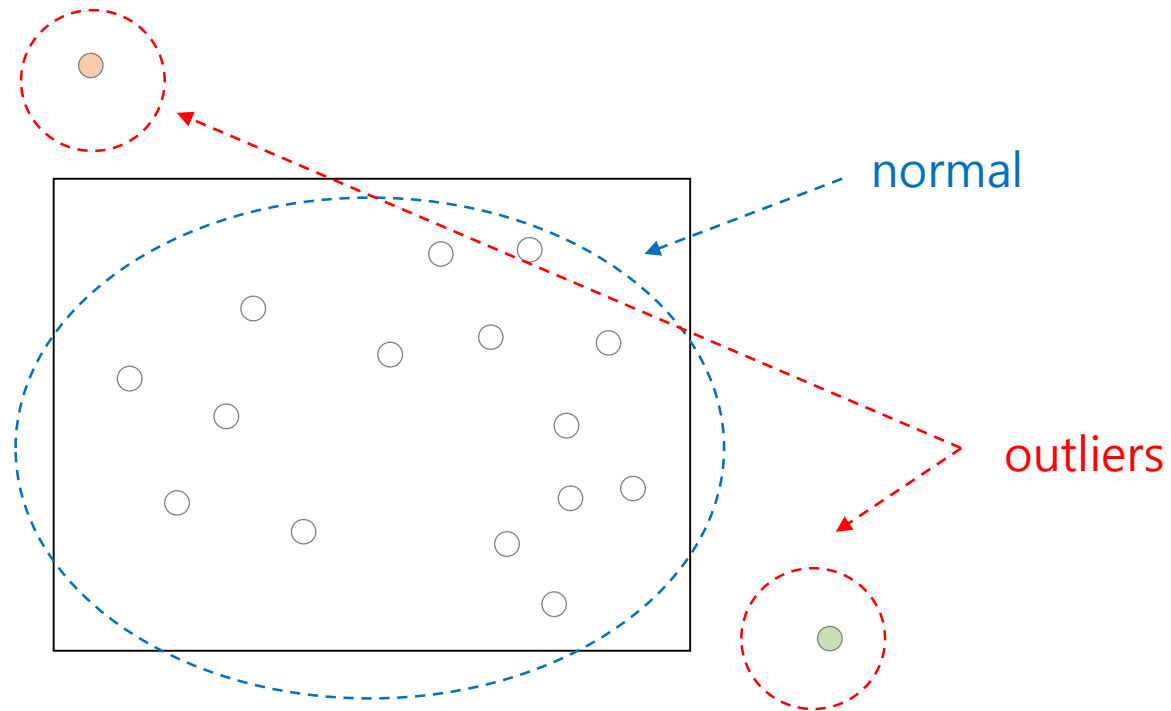
Limitations 4. Sensitive to noise points

- 모든 점을 반드시 한 개 이상의 군집으로 assign 하기 때문에, 일단 가장 가까운 군집에 할당되어 centroid 를 크게 움직입니다.



Solutions 4. Sensitive to noise points

- 데이터의 노이즈를 미리 제거하는 것이 좋습니다.
 - 텍스트 데이터에서는 길이가 극단적으로 짧거나 긴 문서들은 노이즈입니다.
 - Cosine distance 도 길이가 1, 2 처럼 지나치게 짧은 문서 간의 거리는 잘 정의되지 않습니다.



k-means

[sklearn.cluster](#) **KMeans** ¶

```
class sklearn.cluster. KMeans (n_clusters=8, init='k-means++', n_init=10, max_iter=300, tol=0.0001,  
precompute_distances='auto', verbose=0, random_state=None, copy_x=True, n_jobs=1, algorithm='auto')
```

[\[source\]](#)

- k -means 는 수렴 속도가 매우 빠릅니다. 절대 max_iter=300 으로 설정할 필요가 없습니다.
 - k -means 역시 근사알고리즘입니다. 반복을 많이 한다하여 더 좋은 결과가 나오지 않습니다.
 - 일반적으로 20 ~ 30 반복이면 거의 수렴합니다.

k -means

`sklearn.cluster.KMeans` ¶

```
class sklearn.cluster.KMeans(n_clusters=8, init='k-means++', n_init=10, max_iter=300, tol=0.0001,  
precompute_distances='auto', verbose=0, random_state=None, copy_x=True, n_jobs=1, algorithm='auto')
```

[\[source\]](#)

- Bag of words 를 이용할 때에는 k 를 크게, $n_init=1$ 로 설정합니다.
 - n_init 번 반복한 뒤 Silhouette 을 이용하여 가장 좋은 결과를 return 합니다.
 - 그러나 고차원 sparse data 에서는 Silhouette 이 제대로 작동하지 않습니다.

GMM / BGMM

Agglomerative clustering

DBSCAN

Gaussian Mixture Model (GMM)

- 유사도

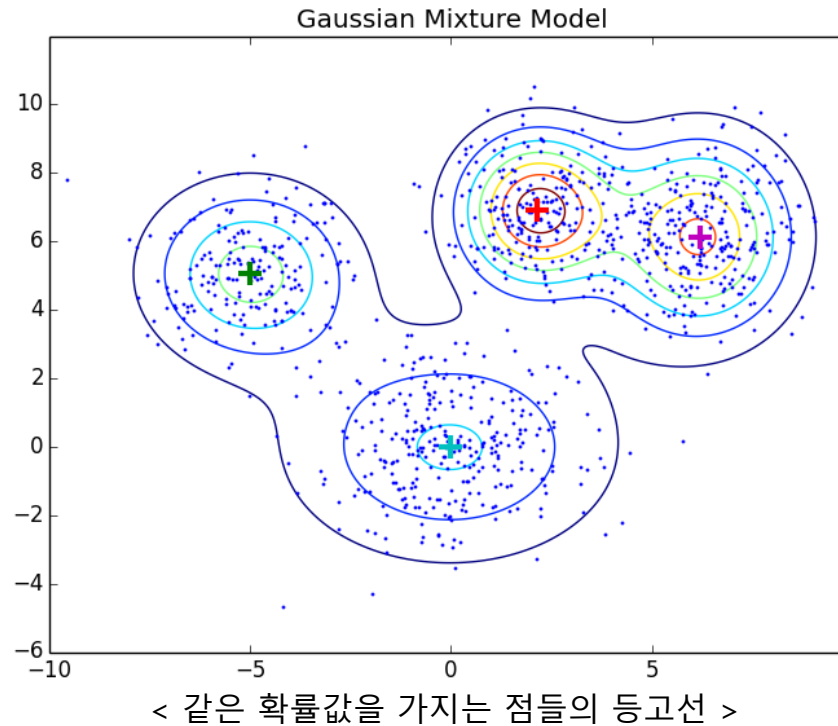
- n 개의 데이터 x 에 대하여 k 개의 Gaussian distribution의 확률값 $P(x_i | G_j)$
- 데이터의 분포 (밀도)를 고려한 k-means 라 생각할 수 있습니다.
- Gaussian 을 이용하기 때문에 Euclidean distance 에 대해서만 정의됩니다.

- 그룹화의 방식

- 데이터의 분포를 가장 잘 설명할 수 있는 k 개의 Gaussian 의 parameter 인 (μ, Σ) 를 학습합니다.

Gaussian Mixture Model (GMM)

- 데이터가 Centroids 를 중심으로 Gaussian 을 따른다고 가정합니다.
- 군집 사이에 밀도 차이가 있을 경우에 적합합니다.



Gaussian Mixture Model (GMM)

- scikit-learn 에 GMM 이 구현되어 있습니다.
 - k -means 처럼 `n_components` 를 사용자가 정의합니다.
 - 원형이 아닌 데이터의 분포를 학습하기 위해 분산행렬의 모양을 선택합니다.
 - `covariance_type` : {'full', 'tied', 'diag', 'spherical'}

sklearn.mixture.GaussianMixture

```
class sklearn.mixture. GaussianMixture (n_components=1, covariance_type='full', tol=0.001,  
reg_covar=1e-06, max_iter=100, n_init=1, init_params='kmeans', weights_init=None,  
means_init=None, precisions_init=None, random_state=None, warm_start=False, verbose=0,  
verbose_interval=10)
```

[\[source\]](#)

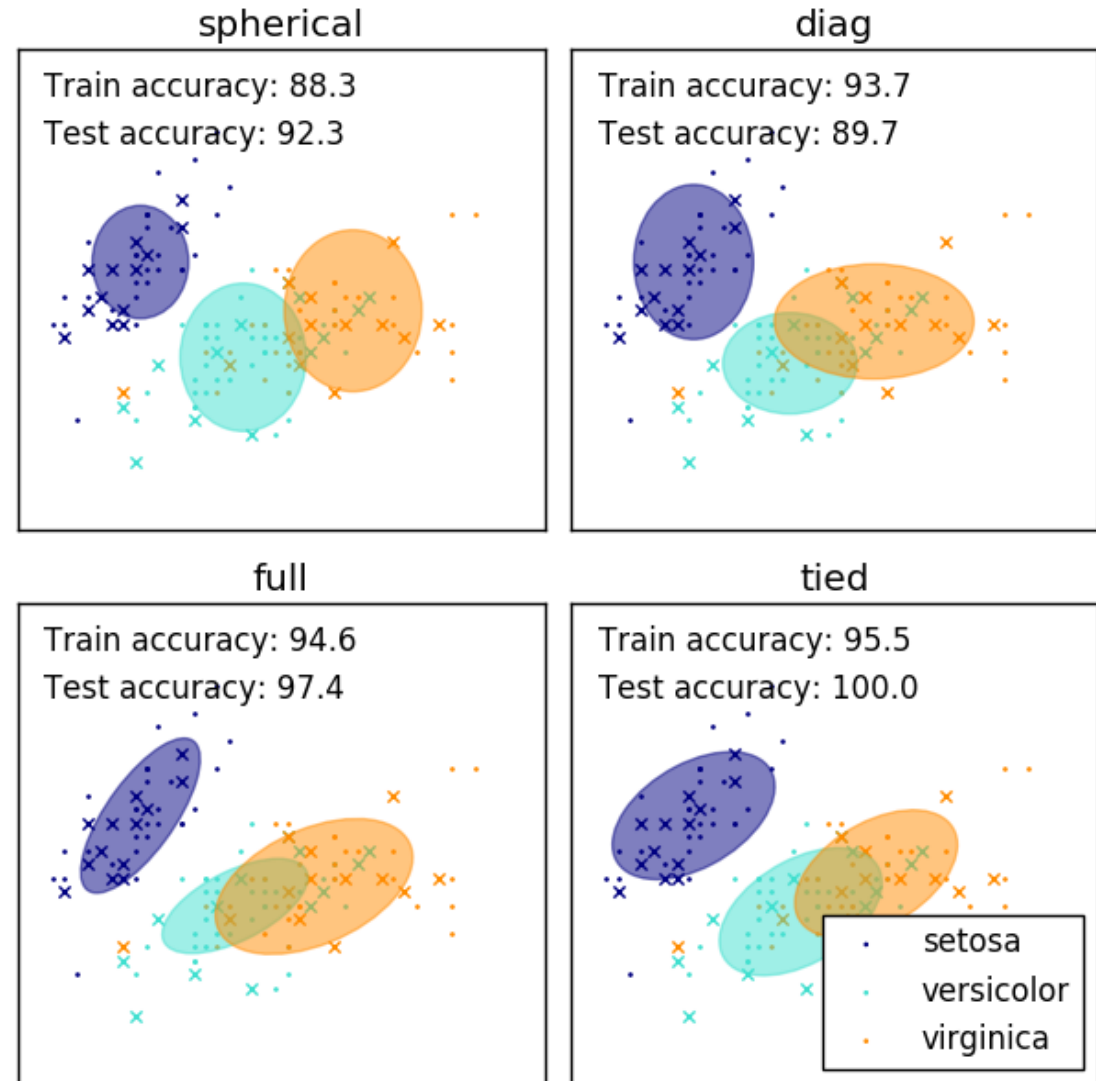
Gaussian Mixture Model (GMM)

'full': its own general covariance matrix,

'tied': share the same general covariance matrix,

'diag': its own diagonal covariance matrix,

'spherical': its own single variance



Gaussian Mixture Model (GMM)

- scikit-learn 에 GMM 이 구현되어 있습니다.
 - Parameter initialization 을 위하여 k -means 이용합니다.
 - k -means 의 기본 설정값이 지나치게 커서 초기화가 오래걸립니다.

sklearn.mixture.GaussianMixture

```
class sklearn.mixture. GaussianMixture (n_components=1, covariance_type='full', tol=0.001,  
reg_covar=1e-06, max_iter=100, n_init=1, init_params='kmeans', weights_init=None,  
means_init=None, precisions_init=None, random_state=None, warm_start=False, verbose=0,  
verbose_interval=10)
```

[\[source\]](#)

Gaussian Mixture Model (GMM)

- pip install 로 설치하는 패키지는 한 directory 에 저장됩니다.

(예시) .../anaconda/envs/YOURENVS/lib/python3.5/site-packages/sklearn

- k_means 파일을 직접 열어서 default argument 를 수정합니다.

.../sklearn/cluster/k_means_.py

```
def __init__(self, n_clusters=8, init='k-means++', n_init=10,  
             max_iter=300, tol=1e-4, precompute_distances='auto',  
             verbose=0, random_state=None, copy_x=True,  
             n_jobs=1, algorithm='auto'):  
  
    self.n_clusters = n_clusters  
    self.init = init  
    self.max_iter = max_iter  
    self.tol = tol
```

Bayesian Gaussian Mixture Model (BGMM)

- 유사도

- n 개의 데이터 x 에 대하여 모델이 학습하는 가장 적절한 k 개의 Gaussian distribution의 확률값 $P(x_i | G_j)$

- 그룹화의 방식

- 데이터의 분포를 잘 설명할 수 있는 k 개의 (평균 μ , 분산 Σ) 을 학습합니다.
- Dirichlet process 를 이용하여 가장 적절한 군집의 개수도 학습합니다.

Bayesian Gaussian Mixture Model (BGMM)

- `n_components` 를 설정할 수 있지만, 모델 스스로 값을 학습합니다.
- Covariance type 은 GMM 과 동일합니다.
- 군집의 구분이 잘 되는 데이터에서 Dirichlet process 가 잘 작동합니다.
 - 노이즈가 많은 데이터에는 잘 작동하지 않습니다.

`sklearn.mixture`.BayesianGaussianMixture

```
class sklearn.mixture. BayesianGaussianMixture (n_components=1, covariance_type='full',  
tol=0.001, reg_covar=1e-06, max_iter=100, n_init=1, init_params='kmeans',  
weight_concentration_prior_type='dirichlet_process', weight_concentration_prior=None,  
mean_precision_prior=None, mean_prior=None, degrees_of_freedom_prior=None,  
covariance_prior=None, random_state=None, warm_start=False, verbose=0, verbose_interval=10)
```

[\[source\]](#)

Hierarchical clustering

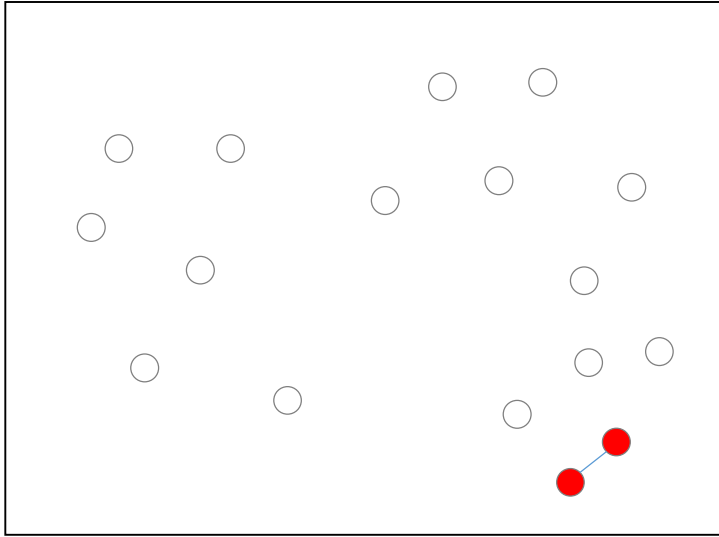
- 유사도

- 두 데이터 x_i, x_j 간에 정의되는 임의의 거리 $d(x_i, x_j)$
 - 그룹 간의 거리는 $d(C_i, C_j)$ 를 기반으로 정의 (min, max, average 등)
 - single linkage / complete linkage

- 그룹화의 방식

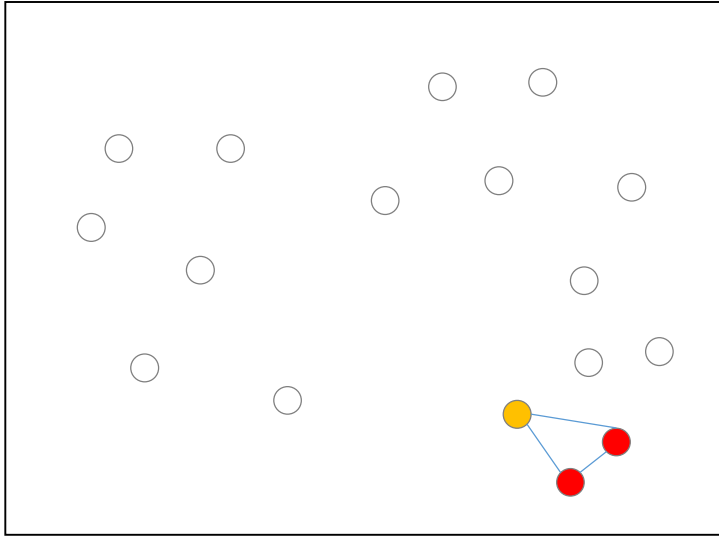
- 거리가 가장 가까운 두 집합을 하나의 집합으로 묶으며, 모든 집합이 하나의 집합이 될 때 까지 반복합니다.

Hierarchical clustering



Iter = 1
가장 가까운 두 점을 연결

Hierarchical clustering



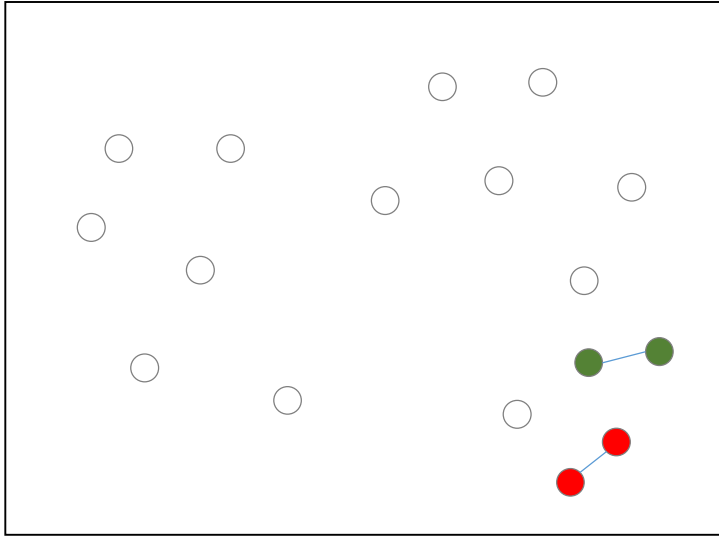
Iter = 1

가장 가까운 두 점을 연결

Iter = 2

$d(C_i, C_j)$ 를 $d(x_p, x_q)$ 의 평균으로 정의한다면
두 빨간색점들과의 거리 평균이 다른 점들보다
가까우므로 주황색 점이 연결
(average linkage)

Hierarchical clustering



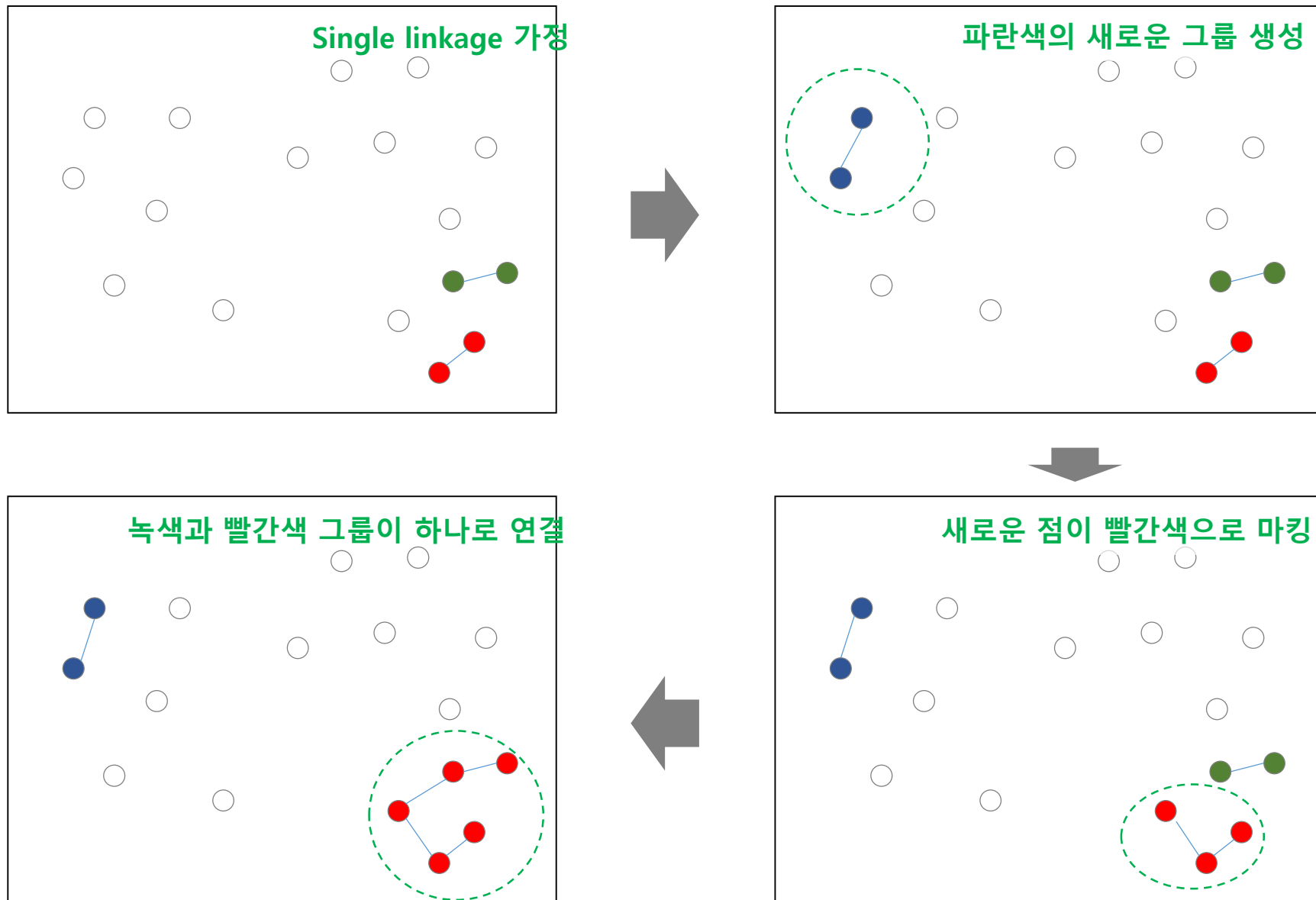
Iter = 1

가장 가까운 두 점을 연결

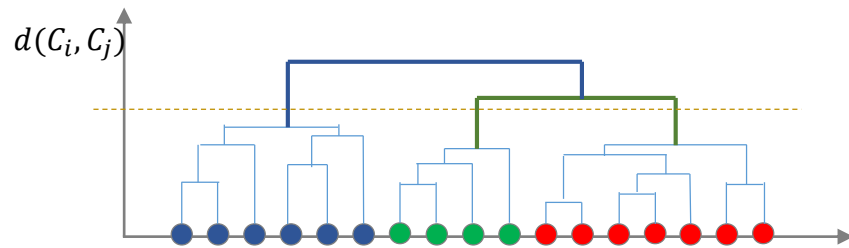
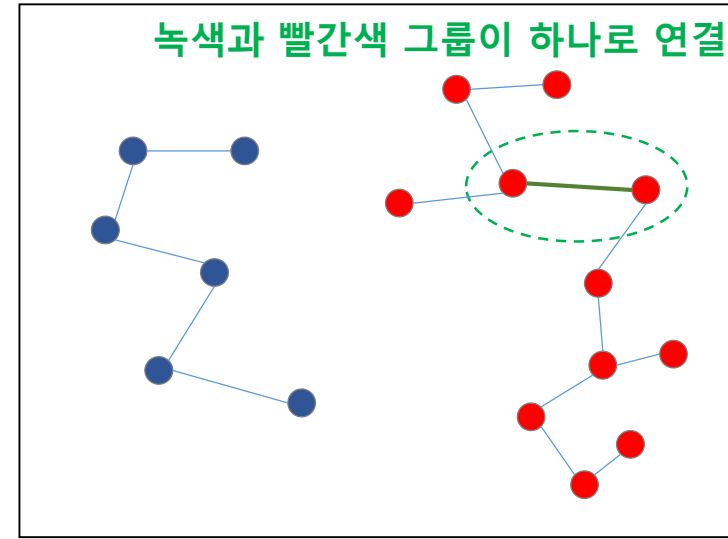
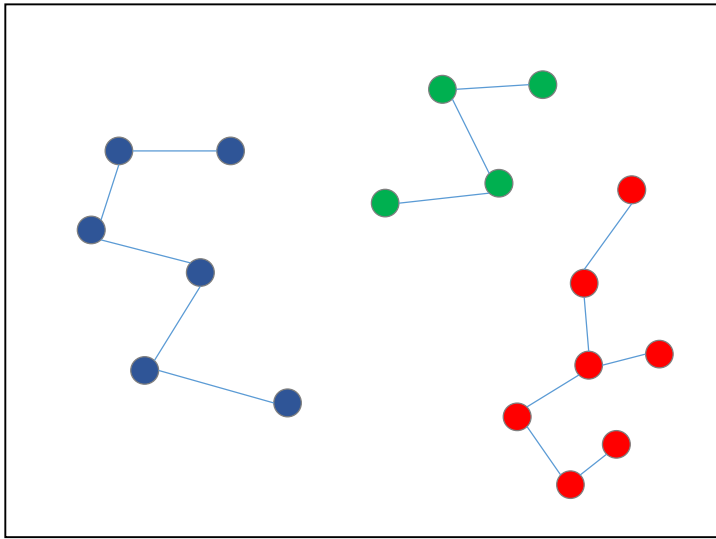
Iter = 2

$d(C_i, C_j)$ 를 $d(x_p, x_q)$ 의 min으로 정의한다면
녹색의 점이 하나로 연결
(single linkage)

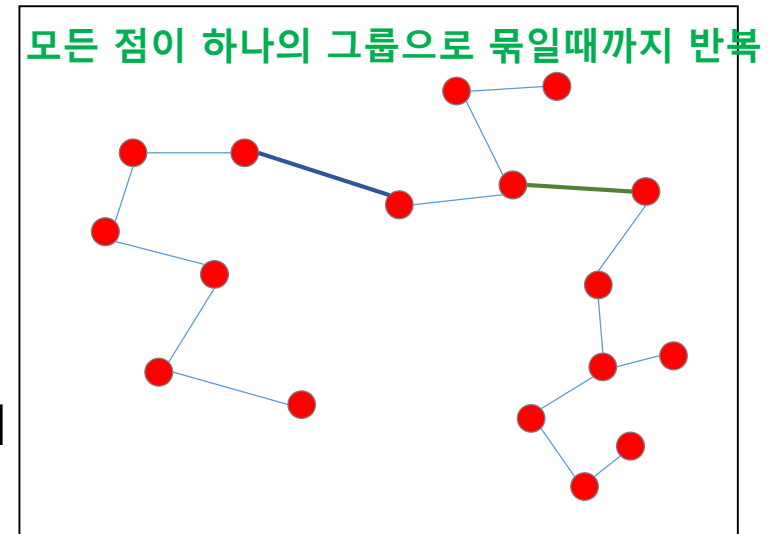
Hierarchical clustering



Hierarchical clustering

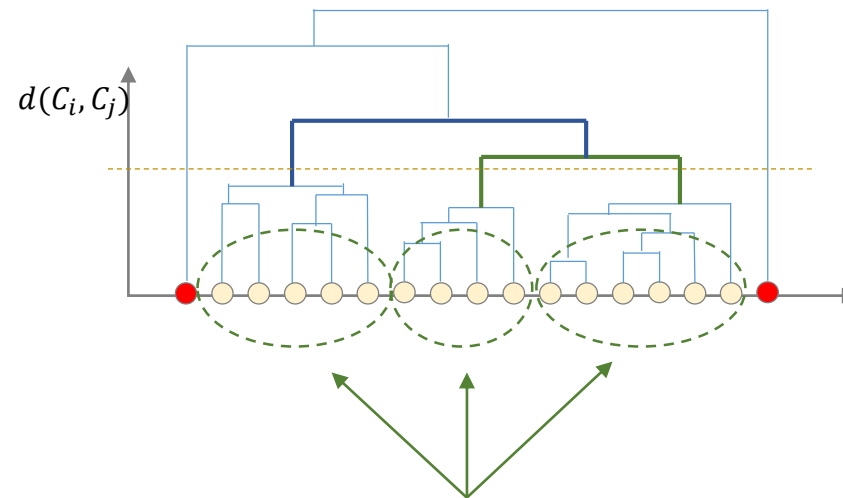


- Dendrogram은 링크가 생성되는 과정을 시각화한것
- 노란선의 distance로 cut한다는 것은 파란/녹색의 링크를 추가하지 않고 3개의 군집으로 묶겠다는 의미



Hierarchical clustering

- Outliers 의 영향을 덜받습니다.
 - Single linkage 는 가장 가까운 점들을 하나씩 이어나갑니다.
 - 마지막까지 다른 점들과 큰 군집으로 묶이지 않는 점들이 outliers 입니다.



다른 점들은 큰 3개의 그룹으로 묶이지만,
붉은색 점들은 마지막에 큰 군집으로 묶임

Hierarchical clustering

- 계산 비용이 비쌉니다.
 - 데이터의 개수가 N 개라고 할 때, 모든 점들간의 거리를 계산해야 하기 때문에 $O(N^2)$ 계산 공간과 비용이 필요합니다.
 - 상대적으로 k -means 보다 큰 계산 공간과 계산 시간을 필요로 합니다.

Hierarchical clustering

- 고차원 벡터에서 잘 작동하지 않습니다.
 - 고차원에서는 최인접이웃들의 거리 외에는 정보력이 없습니다.
 - average linkage 는 두 군집의 모든 점들 간의 거리의 평균을 군집 간의 거리로 이용하기 때문에 대부분의 군집 간 거리가 비슷합니다.
 - 고차원 데이터에서는 최초의 몇 단계 외에는 의미가 없습니다.

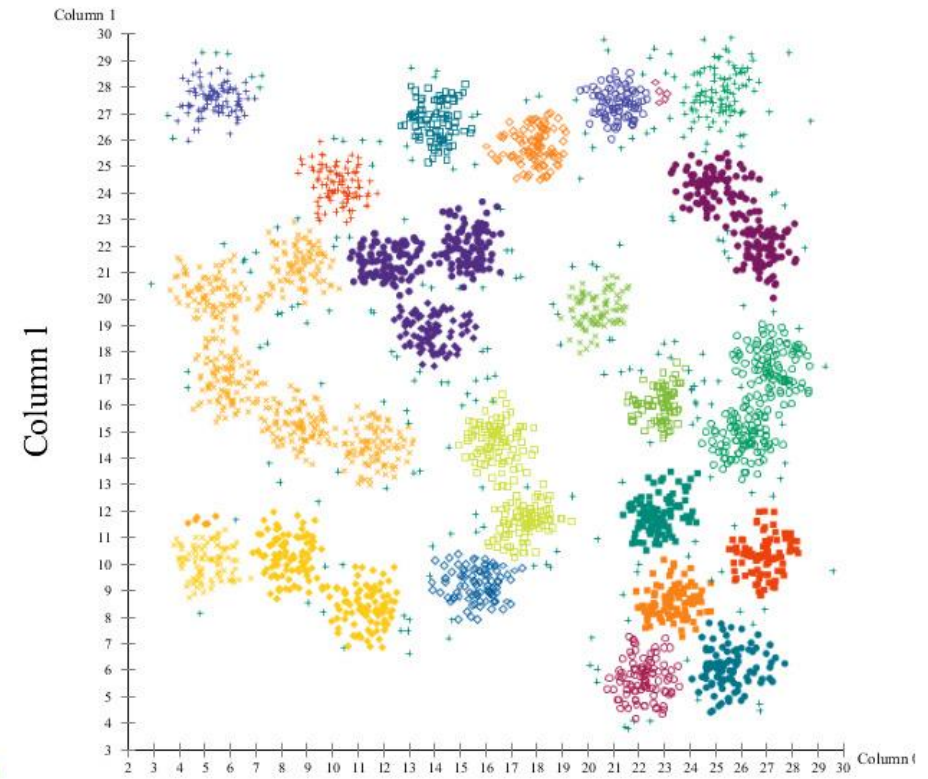
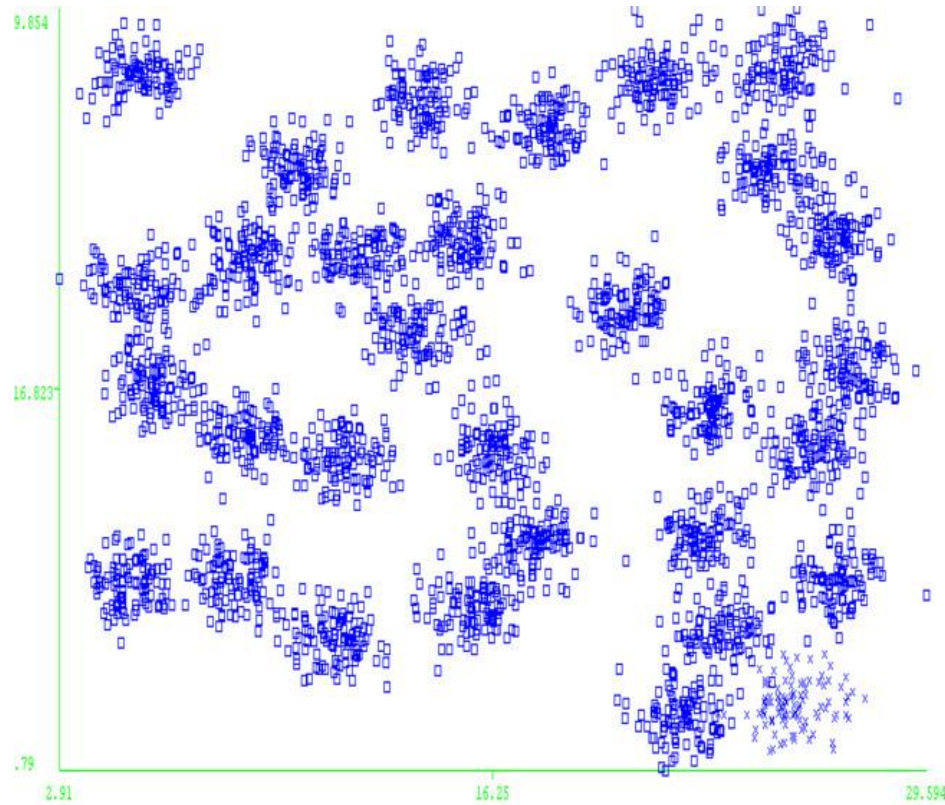
Hierarchical clustering

- Clustering ensemble 에서는 hierarchical clustering 을 이용합니다.
 - 고차원에서는 Euclidean 이나 Cosine 을 이용하여 데이터 간 거리를 정의하기 어렵습니다.
 - Clustering ensemble 은 여러 번의 클러스터링 결과를 이용하여 데이터 간의 유사도를 정의합니다. 잘 정의된 유사도가 주어진다면 hierarchical clustering 을 적용할 수 있습니다.

DBSCAN

- Density-Based Spatial Clustering of Applications with Noise
 - 모든 점이 반드시 그룹에 속하지 않는다고 가정합니다 (노이즈)
 - 유사도
 - n 개의 데이터 X 에 대하여 두 데이터 x_i, x_j 간에 정의되는 임의의 거리 $d(x_i, x_j)$
 - 그룹화의 방식
 - Threshold 이상의 밀도를 지닌 점들을 모두 이어나갑니다.

DBSCAN

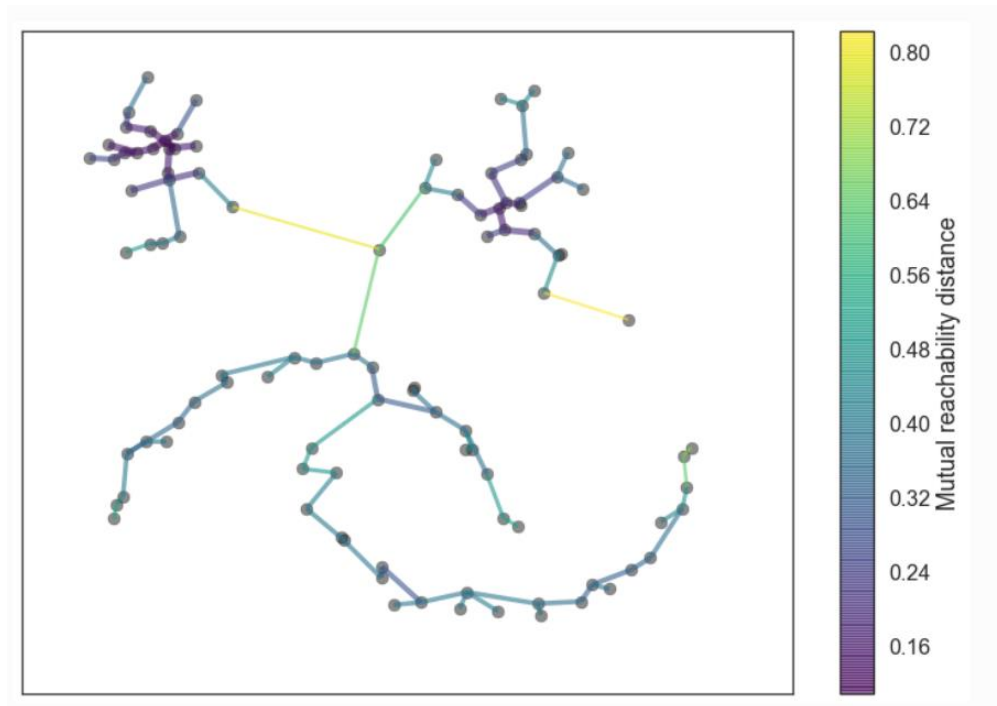


DBSCAN

- Parameters 에 민감합니다.
 - 군집을 결정하는 밀도값 threshold 에 의하여 데이터에서의 노이즈 비율이 예민하게 변합니다.
- 계산 비용이 큼니다.
 - DBSCAN 은 모든 점들간의 거리를 한 번 이상 계산해야하기 때문에 $O(N^2)$ 의 계산 비용을 필요로 합니다.

HDBSCAN

- 최근에 parameter 에 조금 덜 민감한 밀도 기반 방법이 제안되었습니다.
- 그러나 이 역시 (경험적으로) parameter 에 따라 군집의 모양이 달라집니다.
- 논문보다 documentation 의 설명을 참고하시면 좋습니다. *



Clustering labeling

Centroid based cluster labeling

- k -means 의 결과는 centroid vectors 와 labels 두 가지 입니다.
- A centroid vector 는 해당 군집에서의 term frequency proportion 과 비슷합니다.
 - L2 normalize 의 효과로, 희귀한 단어의 weight 는 더 줄어듭니다.
 - TF-IDF 와 같은 term – weighting 과정을 거친다면 해당 군집에서의 weighted term proportion 으로 해석할 수 있습니다.

Centroid based cluster labeling

- 다른 군집보다 상대적으로 자주 등장한 단어를 키워드로 정의합니다.
- 군집 C_i 에서의 단어 w_{ij} 의 키워드 점수, $s(w, c_i)$ 를 정의합니다.

$$s(w, c_i) = \frac{p_i(w)}{p_i(w) + p_{-i}(w)}$$

$p_i(w)$: C_i 에서 단어 w 의 비율

$p_{-i}(w)$: C_i 외에서 단어 w 의 비율

Centroid based cluster labeling

- 각 군집의 문서 개수가 다르기 때문에 p_i, p_{-i} 를 다음과 같이 정의합니다.
 $p_i / (p_{-i} + p_i)$ 은 discriminative power 를 나타냅니다.

- $$p_i(w) = \frac{DF(C_i)}{DF(C_i)} * c_{iw}$$

- $$p_{-i}(w) = \frac{1}{\sum_{j \neq i} DF(C_j)} \times \sum_{j \neq i} DF(C_j) * c_{jw}$$

Centroid based cluster labeling

- 한 군집의 키워드는 군집 내 여러 문서들에서 등장해야 합니다
 - Coverage 가 큰 단어는 centroids 의 값이 클 가능성이 높습니다.

1. for each cluster and its centroid c_i , select top k1 words order by c_{iw}
2. compute $s(w, c_i)$ and select top k2

Centroid based cluster labeling

- Cluster (or topic) labeling algorithms 들은 cluster label 과 matrix X 를 이용하는 경우가 많습니다 [1,2,3,4].
- 제안된 방법은 Term frequency matrix 를 이용하지 않고도 빠르게 cluster labeling 을 할 수 있습니다.
 - 정교한 labeling 이전에, 빠르게 cluster 의 경향을 해석하는 용도입니다.

[1] David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. Evaluating topic 220 models for digital libraries. In Proceedings of the 10th annual joint conference on Digital libraries, pages 215–224. ACM, 2010.

[2] Carson Sievert and Kenneth E Shirley. Ldavis: A method for visualizing and interpreting topics. 235 In Proceedings of the workshop on interactive language learning, visualization, and interfaces, pages 63–70, 2014.

[3] Justin Snyder, Rebecca Knowles, Mark Dredze, Matthew Gormley, and Travis Wolfe. Topic models and metadata for visualizing text corpora. Proceedings of the 2013 NAACL HLT Demonstration Session, pages 5–9, 2013.

[4] Adrian Kuhn, St'ephane Ducasse, and Tudor G'irba. Semantic clustering: Identifying topics in source code. Information and Software Technology

Performance: Clustering labeling

- IMDB reviews ($k=1,000$)
 - 2,514 개의 영화 리뷰가 포함되어 있기 때문에 $k=1,000$ 으로 학습합니다.
 - 5 개의 예시 군집들의 의미가 잘 파악됩니다.

영화 "타이타닉"	iceberg, zane, sinking, titanic, rose, winslet, camérons, 1997, leonardo, leo, ship, cameron, dicaprio, kate, tragedy, jack, disaster, james, romance, love, effects, special, story, people, best, ever, made
Marvle comics 의 heros (Avengers)	zemo, chadwick, boseman, bucky, panther, holland, cap, infinity, mcu, russo, civil, bvs, antman, winter, ultron, airport, avengers, marvel, captain, superheroes, soldier, stark, evans, america, iron, spiderman, downey, tony, superhero, heroes
Cover-field, District 9 등 외계인 관련 영화	skyline, jarrod, balfour, strause, invasion, independence, cloverfield, angeles, district, los, worlds, aliens, alien, la, budget, scifi, battle, cgi, day, effects, war, special, ending, bad, better, why, they, characters, their, people
살인자가 출연하는 공포 영화	gayheart, loretta, candyman, legends, urban, witt, campus, tara, reid, legend, alicia, englund, leto, rebecca, jared, scream, murders, slasher, helen, killer, student, college, students, teen, summer, cut, horror, final, sequel, scary
영화 "매트릭스 "	neo, morpheus, neos, oracle, trinity, zion, architect, hacker, reloaded, revolutions, wachowski, fishburne, machines, agent s, matrix, keanu, smith, reeves, agent, jesus, machine, computer, humans, fighting, fight, world, cool, real, special, effects

Performance: Clustering labeling

- “소나타”가 포함된 네이버 블로그 (k=500)
 - 소나타는 다의어이기 때문에 다양한 문맥에서 이용됩니다.
 - 각 군집의 레이블로부터 소나타의 문맥을 유추할 수 있습니다.

렌트카 광고	제주렌트카, 부산출발제주도, 제주신, 이끌림, 제주올레, 왕복항공, 불포함, 제주도렌트카, 064, 롯데호텔, 자유여행, 객실, 제주여행, 특가, 해비치, 제주시, 제주항, 티몬, 2박3일, 올레, 유류, 항공권, 조식, 제주도여행, 제주공항, 2인
중고차 매매	최고급형중고, 최고급, 프리미어, 프라임, 2011년식, YF소나타TOP, 2010년식, 풀옵션, 2011년, YF소나타PR, 1인, Y20, 2010년, 완전무사고, 판매완료, 군포, 검정색, YF쏘나타, 2011, 하이패스, 2010, 무사고, 등급, 파노라마, 허위매물
클래식 음악	금관악기, 아이엠, Tru, 트럼펫, 트럼, 나팔, 금관, 텔레만, Eb, 호른, 오보에, Tr, Concerto, 하이든, 협주곡, Ha, 악기, 연주하는, 오케, 오케스트라, 독주, 악장, 작곡가, 곡
아이비 “유혹의 소나타 ”	Song, 공부할, 부른, 노래, 가사, 부르는, 가수, 보컬, 목소리, 발라드, 명곡, 신나, 들으면, 듣기, 유혹의, 앨범,아이비, 제목
광염 소나타 및 일제강점기 소설들	백성수, 발가락, 현진, 이광수, 김유, 자연주의, 친일, 평양, 운수, 유미, 저지르, 야성, 탐미, 김동인, 복녀, 광염, 닦았다, 사실주의, 광기, 저지, 1920, 단편소설, 범죄, 감자, 동인, 한국문학

Summary

군집화

- k -means 는 centroids 를 중심으로 구형의 군집을 만듭니다.
 - Euclidean 을 이용할 경우 구 형태의 군집을 만듭니다.
 - Cosine 을 이용할 경우, 벡터의 각도를 기준으로 만들어진 partition 입니다.
- Hierarchical clustering, DBSCAN 은 복잡한 모양의 데이터용입니다.
 - Sparse vector + Cosine 의 공간은 복잡하지 않습니다.
 - 단순한 알고리즘이 빠르며 안정적입니다.

문서 군집화

- 고차원 벡터에서는 매우 가까운 거리만 의미를 지닙니다.
 - k -means 이용 시 k 가 지나치게 작을 경우 먼 문서들이 하나의 클러스터에 할당될 수 있기 때문에 불안정한(unstable) 학습이 될 수 있습니다.
 - 고차원 벡터의 경우 충분히 큰 k 로 군집화를 수행한 뒤, 동일한 의미를 지니는 군집들을 하나로 묶는 후처리 (post-processing) 방식을 추천합니다.

문서 군집화

- 불필요한 단어들을 제거하는 것은 군집화 알고리즘에 도움이 됩니다.
 - Document frequency (DF)가 지나치게 높거나 낮은 단어
 - 뉴스 문서에서 '기자'와 같은 단어나 '-는'과 같은 단어

Diagram illustrating the process of finding the minimum element in an array and swapping it with the first element.

Initial Array:

3	5	0	0	0	0	5
---	---	---	---	---	---	---

Step 1: Find the minimum element. The minimum element is 0, located at index 2.

Step 2: Swap the minimum element with the first element (index 0).

Resulting Array:

0	3	0	2	0	1	4
---	---	---	---	---	---	---

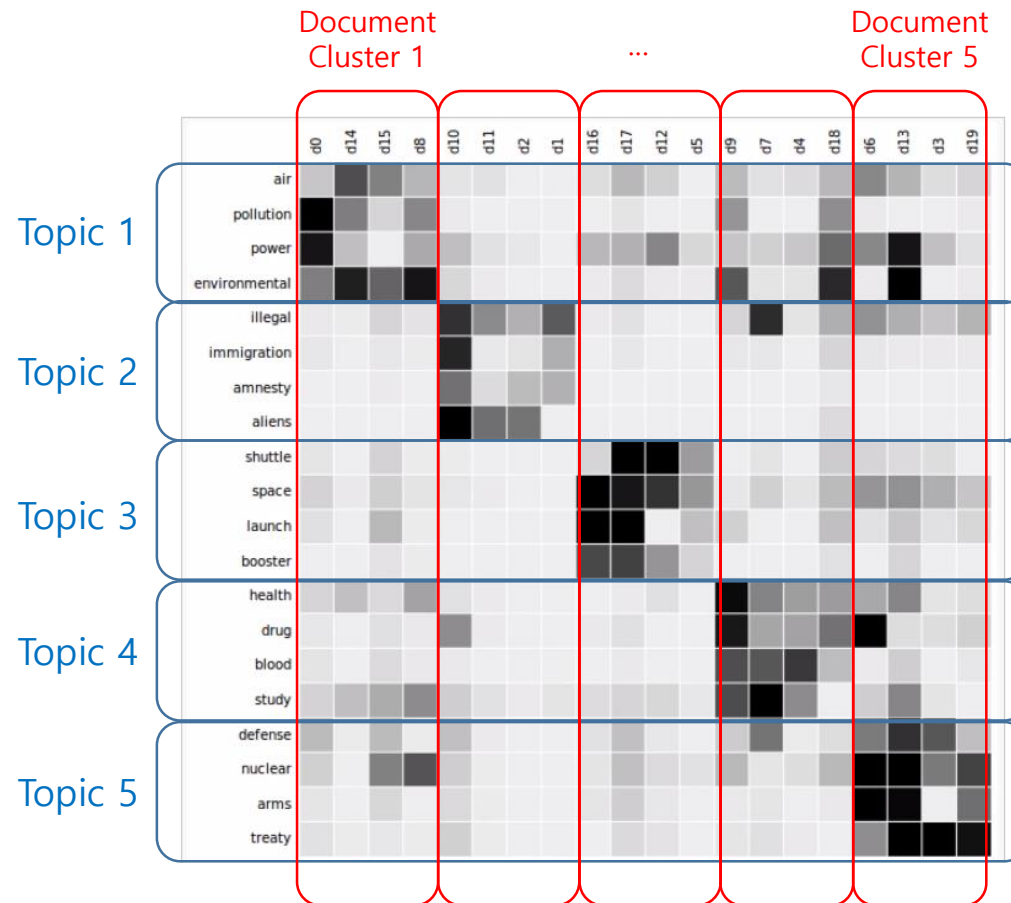
The diagram shows the array after the first swap. The minimum element (0) has been moved to the first position, and the element at the first position (3) has been moved to the position of the minimum element.

문서 군집화

- Topic modeling 에 이용되는 LDA 는 “단어 → 토픽” 으로의 군집화의 의미로 해석할 수 있습니다.
- 좋은 LDA 학습 결과를 위한 팁은 문서 군집화와 비슷합니다.
 - 기대하는 군집/토픽의 개수보다 더 많은 개수를 설정합니다.
 - 변별력이 없는 단어는 term document matrix 에서 제거한 뒤 학습합니다.
 - 어느 문서에나 등장하거나 (= DF가 매우 크거나)
 - 특정 문서에만 등장하는 단어 (= DF가 매우 작은)

문서 군집화

- 단어-문서 행렬이 뚜렷한 블록들로 구분이 된다면 군집화에 최적입니다.



Package

Spherical k -means

```
from soyclustering import SphericalKMeans

kmeans = SphericalKMeans(
    n_clusters=1000,
    init='similar_cut',
    max_iter=10,
    tol=0.0001,
    verbose=True
)

labels = kmeans.fit_predict(x)
```

Cluster labeling

```
from soyclustering import proportion_keywords

keywords = proportion_keywords(
    kmeans.cluster_centers_,
    labels,
    index2word=idx2vocab,
    topk=30,
    candidates_topk=100
)

keywords[0] # [(word, label_score), (word, label_score), ... ]
```