4회차 과제 답안_05조(5늘의 스터디)

1. K-평균 군집화 모델의 평가지표인 Rand지수와 실루엣 계수에 대해 설명해보시오

$$\text{Rand Index} = \frac{a+b}{{}_{N}C_{2}}$$

Rand지수

- **Rand 지수**는 퍼뮤테이션을 무시했을 때 같은 데이터에 대한 두 군집화 결과의 유사성을 측정하는 방법이다.
- (* a = 실제 군집 p에서도 같은 군집에 속하고 모델링에 따른 군집화 결과 c에서도 같은 군집에 속한 쌍의 수, b = p에서도 다른 군집에 속하고, c에서도 다른 군집에 속한 쌍의 수, n = 전체 샘플 개수)
- **특징:**
  1. 결과를 해석하기 쉽다.
  2. 군집 구조에 대한 어떠한 가정도 필요로 하지 않는다.
  3. 실제 레이블 값이 알려졌을 때 모든 군집화 기법을 평가하는 데 적용할 수 있다.
  4. 0~1 사이의 값을 가지며, 클수록 결과가 비슷하다.
  5. 유사도를 과대평가하는 단점을 가진다.

Rand지수는 가능한 모든 데이터 쌍의 개수에 대해 정답인 데이터 쌍의 개수의 비율로 정의한다.

```
from scipy.special import comb
rand_index = a_plus_b / comb(incidence.shape[0], 2)
rand_index
```

```
0.6
```

실루엣 계수

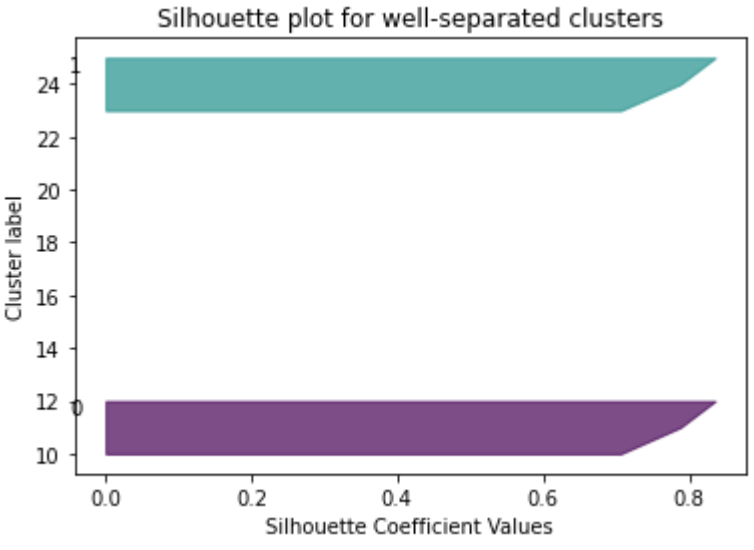
- 실제 레이블을 모를 때 유용하게 사용할 수 있다.

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

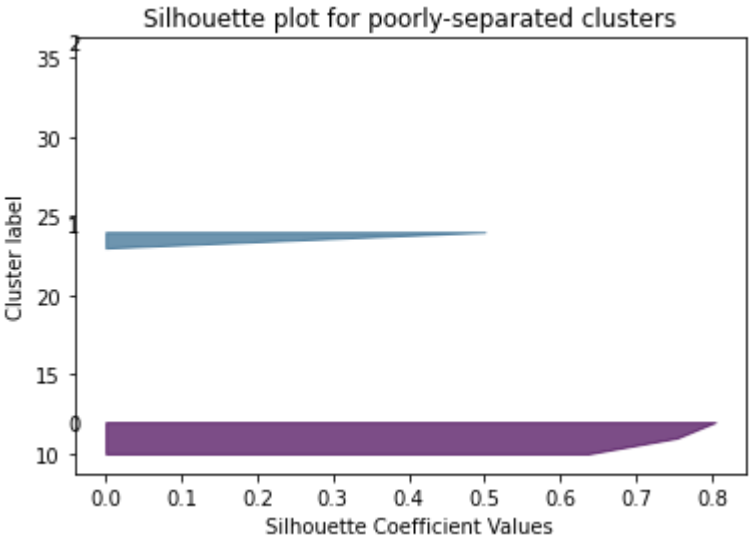
- 실루엣 계수는 -1~1사이의 값을 지닌다.
- 만일 데이터 i에 대해 같은 군집의 데이터가 다른 군집의 데이터보다 더 가깝다면 그 데이터의 실루엣 계수는 양수가 된다.
- 잘못된 군집화에서는 실루엣 계수가 음수인 데이터가 많아지므로 평균 실루엣 계수가 작아진다.

→ **즉, 실루엣 계수가 클수록 좋은 군집화라고 할 수 있다.**

- 컨벡스한 군집에서 높게 나타나는 단점이 존재한다.



잘된 군집화 예시



잘못된 군집화 예시

2. K-최근접 이웃 모델과 K-평균 군집화 모델의 차이를 서술하고 각 모델이 활용될 수 있는 사례를 적어보시오.

차이점

- KNN은 지도학습 알고리즘이며, 분류나 회귀를 해결하기 위해 라벨이 있는 학습 데이터를 기반으로 예측을 수행한다.
- K-평균 군집화 모델은 비지도 학습 모델로, 데이터를 그룹화하거나 클러스터링을 하기 위해 라벨이 없는 데이터의 패턴을 기반으로 클러스터링을 수행한다.
- 둘은 모두 K개의 점을 지정하여 거리를 기반으로 구현되는 거리기반 분석 알고리즘이다.
- KNN은 이웃의 개수(K)에 따라 예측이나 분류 결과가 달라지지만, K-평균 군집화 모델은 초기 중심 위치에 따라 군집 결과가 달라진다.
- KNN은 예측 시에 새로운 데이터에 대해 인접한 이웃을 찾아 사용하지만, K-평균 군집화 모델은 주어진 데이터를 군집화하여 군집의 중심을 찾는다.

활용사례

- K-NN의 경우, 이메일 스팸 필터링, 영화 추천 시스템과 같은 예측에 활용될 수 있다.
- K-평균 군집화는 고객의 구매 이력이나 행동패턴을 기반으로 고객 세분화할 때 활용 가능하다.

3.계층적 군집화 모델에서 군집 사이의 거리를 정의하는 연결법들을 나열해보고 각 연결법들의 특징에 대해 서술해보시오.

단일 연결법: 두 군집 간 원소끼리의 거리를 모두 비교한 후 그중 **최소 거리**를 군집 간 거리로 정의하는 방법이다.

최장 연결법: 두 군집 간 원소끼리의 거리를 모두 비교한 후 그 중 **최대 거리**를 군집 간 거리로 정의하는 방법이다.

평균 연결법: 두 군집 간 원소끼리의 거리를 모두 비교한 후 그 **평균 거리**를 군집 간 거리로 정의하는 방법이다.

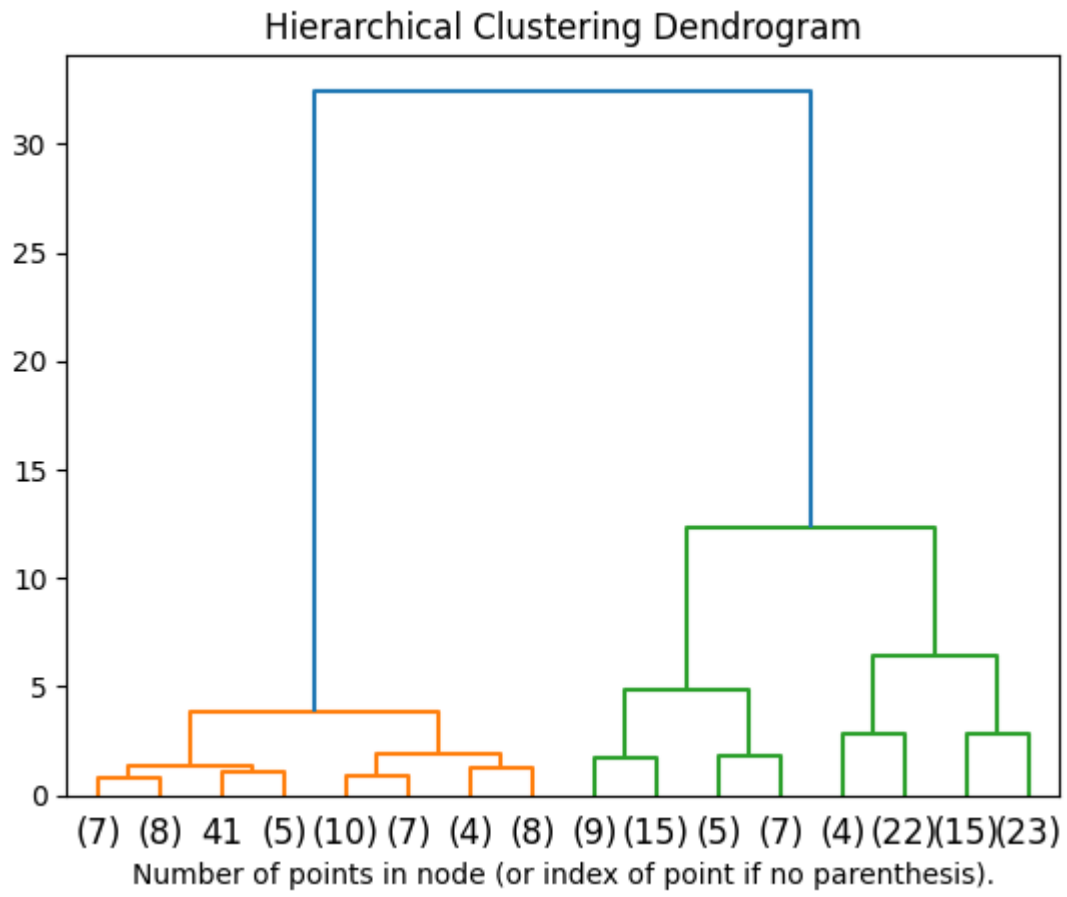
와드 연결법: 두 군집을 병합했을 때, **군집 내 분산의 증가분**을 두 군집 사이의 거리로 정의하는 방법이다.

중심 연결법: 각 군집의 중심을 구한 후 **중심 사이의 거리**를 군집의 거리로 정의하는 방법이다.

4.교재 346페이지에 덴드로그램 시각화 예제를 실습해보면서 각 과정에 대해 설명해보시오.

```
# 실습에 필요한 패키지 불러오기
import matplotlib.pyplot as plt
import numpy as np
from scipy.cluster.hierarchy import dendrogram
from sklearn.cluster import AgglomerativeClustering
from sklearn.datasets import load_iris

# 데이터 불러오기
# 군집화는 비지도 학습 모델이므로 y는 정의하지 않음
x = load_iris().data

# 병합적 군집화 모델 생성
# distance_threshold = 0으로 전체 계층 구조를 계산
model = AgglomerativeClustering(distance_threshold=0, n_clusters=None)

# 군집화 모델 학습
model = model.fit(x)

# linkage_matrix 계산
counts = np.zeros(model.children_.shape[0])
n_samples = len(model.labels_)
for i, merge in enumerate(model.children_):
  current_count = 0
  for child_idx in merge:
    if child_idx < n_samples:
      current_count += 1
```

```
    else:
      current_count += counts[child_idx - n_samples]
  counts[i] += current_count
linkage_matrix = np.column_stack([model.children_, model.distances_, counts]).astype(float)

# 덴드로그램 시각화
# 위에서 3개 계층까지만 시각화
dendrogram(linkage_matrix, truncate_mode='level', p=3)
plt.title("Hierarchical Clustering Dendrogram")
plt.xlabel("Number of points in node (or index of point if no parenthesis).")
plt.show()
```



5. 이전 5장에서 릿지 회귀의 유일해는 $\hat{w}_{\text{Ridge}} = (X^T X + \alpha I)^{-1} X^T y$임을 알 수 있었다.이를 X의 특잇값 분해를 이용해 다시 나타내면, $X\hat{w}_{\text{Ridge}} = U\Sigma(\Sigma^2 + \alpha I)^{-1}\Sigma U^T y$이고, 규제가 없는 최소제곱법 모델의 해는 $X\hat{w} = UU^T y$이다. 이를 통해 PCA모델과 릿지회귀에는 어떠한 관련성이 있는지 설명해보이시오.

릿지 회귀	회귀 모델의 오버피팅을 방지하기 위해 규제를 사용하는 회귀 모델
PCA	고차원 데이터를 저차원 공간으로 투영하는 방법, 데이터의 주성분을 찾는 데 사용

릿지 회귀의 해는 $\hat{w}Ridge = (XTX + I)^{-1}XTy$다.

여기서 X는 입력 데이터 행렬, y는 타깃 데이터 행렬, I는 단위 행렬이다.
X의 특잇값 분해를 이용하면
$X = U\sum V$가 된다.

이를 이용해서 릿지 회귀 해를 다시 표현 하면, $X\hat{w}Ridge = U\sum(\sum^2 + aI)^{-1}Uy$다.

여기서 a는 규제 강도이며 규제가 없는 최소 제곱법 모델의 해는 다음과 같이 표현할 수 있다.

$$\hat{u} = (XTX)^{-1}XTy$$
$$X\hat{u} = UUTy$$

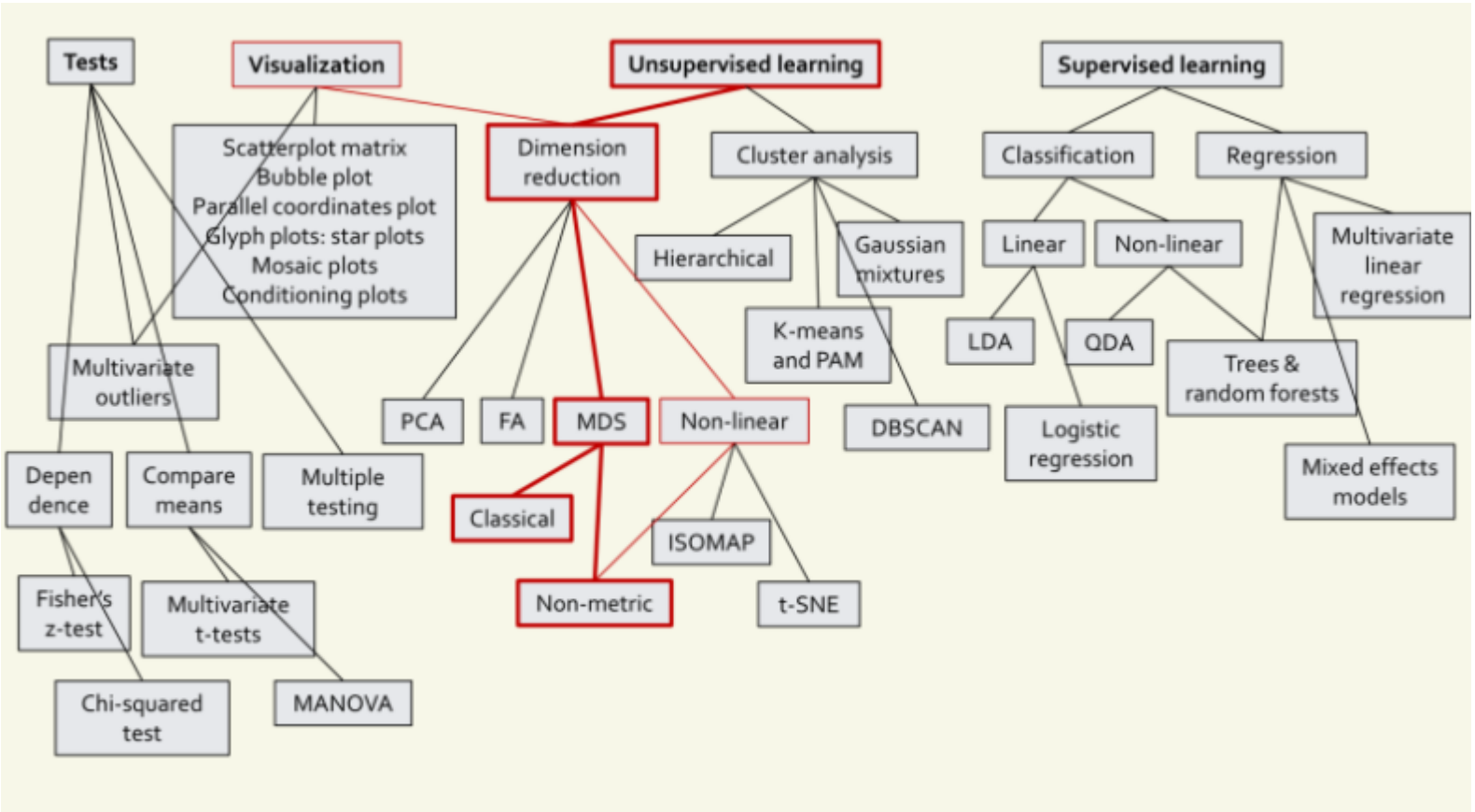
위의 두 표현을 비교하면 다음과 같은 관계를 알 수 있다.

> 💡 $X\hat{w}Ridge = U\sum(\sum^2 + aI)^{-1}Uy$는 a = 0일때, $X\hat{u} = UUTy$와 같다.
> a>0인 경우,
> $X\hat{w}Ridge$는 $X\hat{u}$와 비슷하지만, $X\hat{u}$보다 규제를 더 많이 받는다.

따라서, 릿지 회귀 모델과 PCA 모델은 아래와 같은 관계를 갖는다.

> 💡 a=0인 경우, 릿지 회귀 모델은 PCA 모델과 동일하다.
> a>0인 경우, 릿지 회귀 모델은 PCA 모델에 규제를 추가한 모델이다.

6. 차원 축소기법인 MDS, Isomap, LLE, t-SNE의 특징에 대해 서술해보시오.



https://velog.io/@swan9405/MDS-Multidimensional-Scaling

MDS

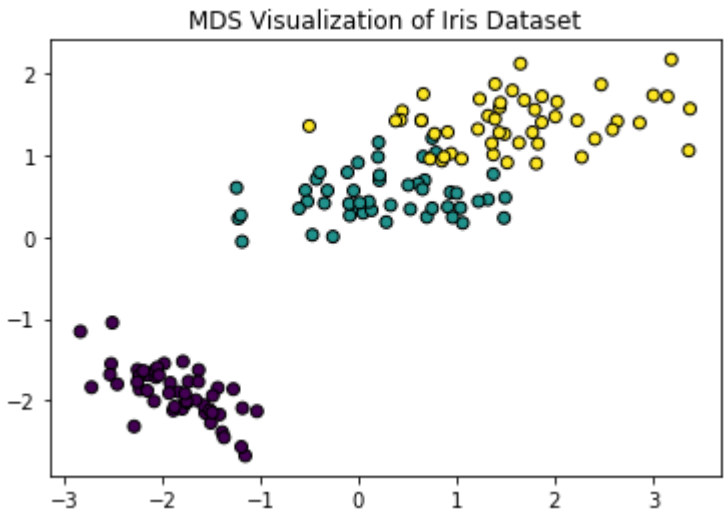
- 고차원에서의 데이터 샘플 간 거리가 저차원에서도 최대한 보존되도록 변환하는 선형 차원 축소 기법
- MDS는 데이터 간의 거리(유사도) 정보를 보존하면서 차원을 축소하여 시각화나 분석을 용이하게 한다는 장점이 있다.
- 단점은 모든 점들 간의 거리 정보의 중요도를 같게 본다는 것이다.

MDS 는 원 공간에서 모든 점들 간에 정의된 거리 **행렬 D**가 주어졌을 때, 임베딩 공간에서의 Euclidean distance 인 **|yi − yj|**와 거리 행렬 **δi**의 차이가 최소가 되는 임베딩 y를 학습한다.

학습 데이터가 원 공간의 벡터로 입력된다 하여도 x의 pairwise distance(관측값 쌍 간의 쌍별 거리) 를 계산함으로써 **D**를 만들 수 있다.

$$minimize \sum_{i<j} \left(|y_i - y_j| - \delta_{ij} \right)^2$$

```
from sklearn.manifold import MDS
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt

# 데이터 로드
iris = load_iris()
X = iris.data
y = iris.target

# MDS 모델 생성
mds = MDS(n_components=2, random_state=42)
X_mds = mds.fit_transform(X)

# 결과 시각화
plt.scatter(X_mds[:, 0], X_mds[:, 1], c=y, cmap='viridis', edgecolor='k')
plt.title('MDS Visualization of Iris Dataset')
plt.show()
```



그림에서 각 클래스(품종)는 색으로 구분되어 있음을 볼 수 있다.

MDS가 원본 데이터의 클래스 구조를 유지하면서 2차원으로 잘 표현하는지를 확인할 수 있다.

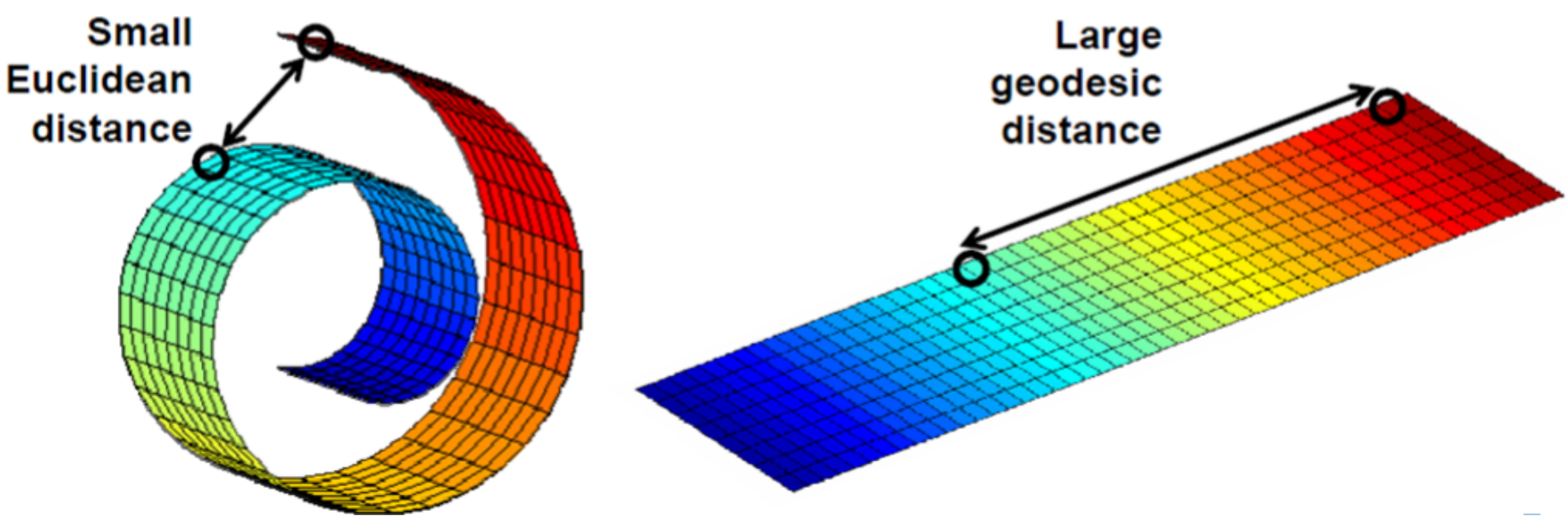
비슷한 샘플은 2차원 공간에서도 가깝게 위치하도록 MDS가 학습된다.

Metric MDS & Non-metric MDS?

Metric MDS와 Non-metric MDS는 다차원 스케일링(MDS)의 두 가지 주요 유형이다.

Metric MDS	Non-metric MDS
거리 보존	순서 보존
유클리드 거리 사용	등간 척도 사용
목적 함수는 원본 데이터간의 거리와 MDS에서 학습한 거리 간의 차이를 최소화하는 것	목적 함수는 순서 차이를 최소화하는 것이며, 이를 위해 Stress majorized by Shepard’s(Kruskal’s)solution(스크리블 튜니드 스트레스)을 최소화함
주로 stress라는 지표를 사용하여 평가	stress와 함께 자주 사용되는 평가지표로서, shepard다이어그램을 확인하여 순서관계를 시각적으로 평가

Isomap



- Isomap은 다차원 스케일링(MDS) 또는 주성분 분석(PCA)의 확장이자 두 방법론을 결합한 방법론으로 볼 수 있다.
- 두 점은 유클리디안 거리로는 가깝지만 실제 측지거리를 구할 경우 색깔이 나타내는 의미만큼 멀리 떨어져 위치함을 알 수 있다.

즉, Isomap 알고리즘은 두 데이터간의 실제 특징을 반영하는 거리 정보를 사용하는 효과적인 차원 축소를 추구한다.

LLE

- 각 샘플마다 최근접 이웃을 찾은 후 고정된 이웃 샘플로 해당 샘플을 근사하거나 재구성해 차원 축소 진행, 비선형 차원 축소 기법이다.

t-SNE

- t-SNE(Stochastic Neighbor Embedding)는 고차원 데이터를 저차원으로 차원 축소하는 비선형 기술
- 특히 고차원 데이터의 구조를 이해하거나 군집 간의 관계를 파악하는 데 유용하다.
- 가까운 이웃일수록 선택될 확률이 높다.
- 다른 샘플을 이웃으로 선택할 확률 분포가 저차원에서도 유지되도록 차원 축소 진행한다.