Spring 2023

SWCON253: Machine Learning

Lecture 14
Unsupervised Learning: Clustering

Jinwoo Choi
Assistant Professor
CSE, Kyung Hee University

Announcement

강의 관련 설문조사: https://app.sli.do/event/5yHHy2F7WHpENeReUrcgs4
(Event code: 3175445)

Contents

References

- “기계학습” by 오일석, "패턴인식" by 오일석

1. Unsupervised Learning

1. Three Types of Learning
2. Three Tasks of Unsupervised Learning
3. Use of Prior Knowledge

Three Types of Learning

- 지도 학습 (Supervised Learning)
  - 모든 훈련 샘플이 레이블 정보를 가짐
- 비지도 학습 (Unsupervised Learning)
  - 모든 훈련 샘플이 레이블 정보를 가지지 않음
- 준지도 학습 (Semi-Supervised Learning)
  - 레이블을 가진 샘플과 가지지 않은 샘플이 섞여 있음
    - ★ e.g.) Active Learning: 성능을 높이는데 효과적인 샘플에 대해 labeling 을 수행

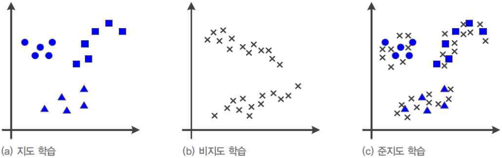


그림 6-1 기계 학습의 유형(속이 찬 샘플은 레이블이 있고, x 표시된 샘플은 레이블이 없음)

Three General Tasks of Unsupervised Learning

- **군집화 (Clustering)**
  - 유사한 샘플을 모아 같은 그룹으로 묶는 일
  - 응용 예) 맞춤 광고, 영상 분할, 유전자 데이터 분석, SNS 실시간 검색어 분석하여 사람들의 관심 파악 등
- **밀도 추정 (Density Estimation)**
  - 데이터로부터 확률분포를 추정하는 일
  - 응용 예) 분류, 생성 모델 구축 등
- **공간 변환 (Feature Space Conversion)**
  - 원래 특징 공간을 저차원 또는 고차원 공간으로 변환하는 일
  - 응용 예) 데이터 압축, 특징 추출(표현 학습), 데이터 가시화 등

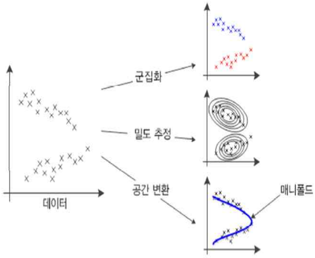


그림 6-2 비지도 학습의 군집화, 밀도 추정, 공간 변환 과업이 발견하는 정보

- 상기 Task들은 서로 밀접하게 연관되어 있음
- 비지도 학습을 위해서는 데이터에 내재한 구조를 잘 파악하여 새로운 정보를 발견해야 함

Use of Prior Knowledge

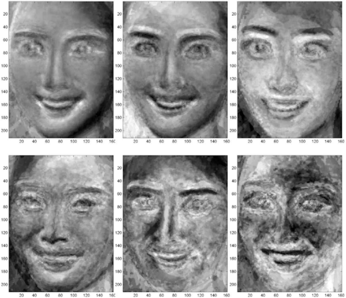
- 기계 학습이 사용하는 두 종류의 지식
  - 훈련 집합
  - 사전 지식 (prior knowledge)
    - ★ e.g., weight penalty in regularization, prior probability of MAP decision
    - ★ e.g., manifold hypothesis, smoothness hypothesis

- (참고) 자주 사용되는 사전 지식
  - 매니폴드 가정(Manifold Hypothesis)
    - ★ 데이터 집합은 특징 공간에서 하나 또는 여러 개의 매니폴드를 구성하며, 모든 샘플은 매니폴드 근처에 있다.
      - 매니폴드: **Low-dimensional structure** that is hypothesized to underlie high-dimensional data.
      - 매니폴드는 데이터의 **essential structure** 라고 할 수 있다.

Example of manifold: PCA

Miss daegu

6 Eigen faces



Credit: http://jbhuang0604.blogspot.com/2013/04/miss-korea-2013-contestants-face.html

Use of Prior Knowledge

- 기계 학습이 사용하는 두 종류의 지식
  - 훈련 집합
  - 사전 지식 (prior knowledge)
    - ★ e.g., weight penalty in regularization, prior probability of MAP decision
    - ★ e.g., manifold hypothesis, smoothness hypothesis

- (참고) 자주 사용되는 사전 지식
  - 매니폴드 가정(Manifold Hypothesis)
    - ★ 데이터 집합은 특징 공간에서 하나 또는 여러 개의 매니폴드를 구성하며, 모든 샘플은 매니폴드 근처에 있다.
      - 매니폴드: low-dimensional structure that is hypothesized to underline high-dimensional data.
      - 매니폴드는 데이터의 essential structure 라고 할 수 있다.
  - 부드러움 가정(Smoothness Hypothesis)
    - ★ 획득된 샘플은 어떤 요인에 의해 특징 공간에서 위치가 변화한다. (예: 조명을 조금씩 변화하면서 영상을 획득)
    - ★ 이때 특징 공간에서의 샘플의 위치는 부드러운 곡면을 따라 변화한다.

- 비지도 학습과 준지도 학습은 사전 지식을 더 명시적으로 사용

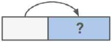
(참고) Self-Supervised Learning (자기지도 학습)

What is Self-Supervised Learning (SSL)?

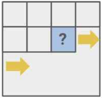
- A special type of *representation learning* that enables learning good data representation *from unlabelled dataset*. (constructing supervised learning tasks out of unsupervised datasets)
- The methods can be categorized as: ***Self-prediction*** and ***Contrastive learning***

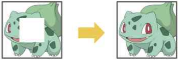
Self-prediction

- Given an individual data sample, *predict one part* of the sample given the other part. The part to be predicted pretends to be missing.



"Intra-sample" prediction





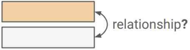
He, Kaiming, et al. "Masked autoencoders are scalable vision learners." Proceedings of the IEEE/CVF CVPR 2022

https://neurips.cc/media/neurips-2021/Slides/21895.pdf)

(참고) Self-Supervised Learning (con't)

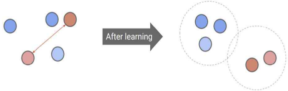
Contrastive learning

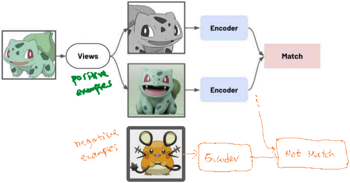
- Given multiple data samples, the task is to *predict the relationship* among them.



"Inter-sample" prediction

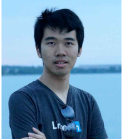
- Tries to learn such an *embedding space* in which similar sample pairs stay close to each other while dissimilar ones are far apart.





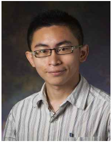
CVIU 2023

Learning Representational Invariances for Data-Efficient Action Recognition


Yuliang Zou[2]


Jinwoo Choi[1]

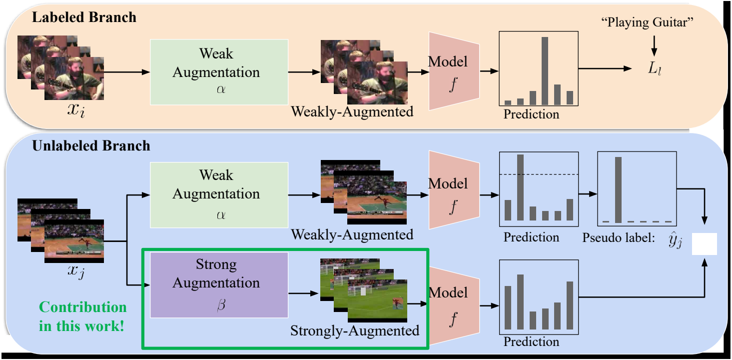

Qitong Wang[2]


Jia-Bin Huang[2]

KYUNG HEE UNIVERSITY

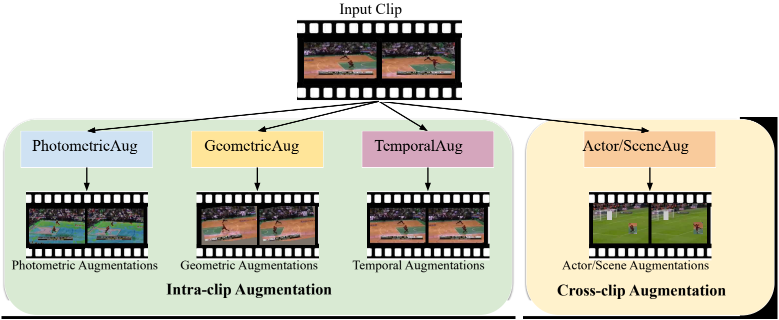
VIRGINIA TECH

FixMatch: semi-supervised learning with consistency regularization

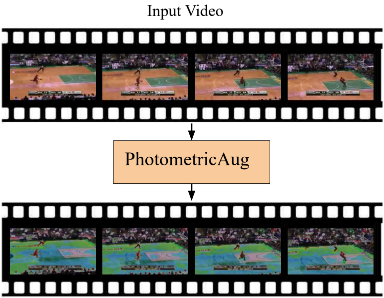


FixMatch: [Sohn et al. NeurIPS 2020]

Motivation: we inject *invariances to our model*

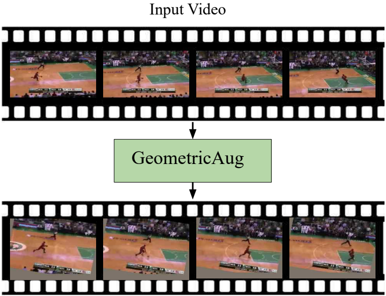


Photometric invariance



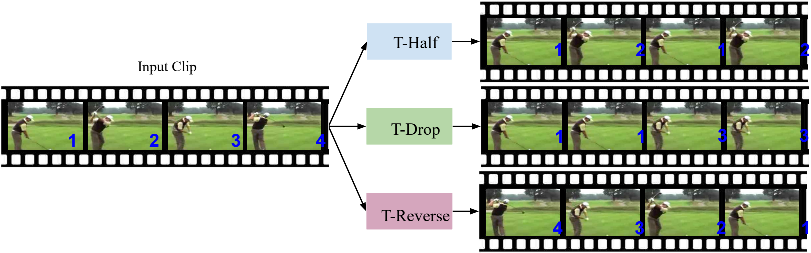
- Brightness
- Contrast
- Sharpness
- Invert color
- Color histogram equalize
- Solarize

Geometric invariance

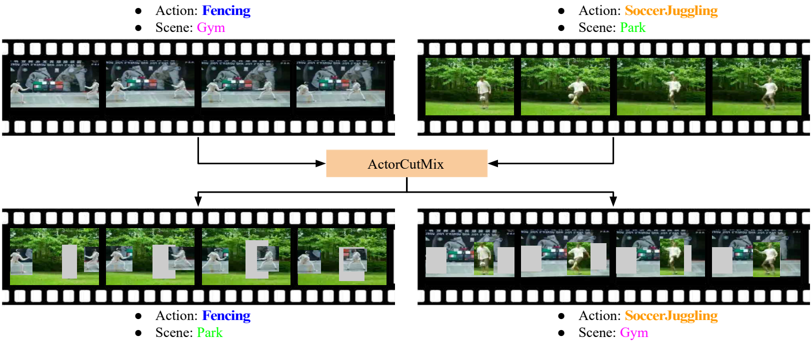


- Translation x/y
- Shear x/y
- Rotation
- Cut out

Temporal invariance



Scene invariance: ActorCutMix



2. *k*-Means Clustering

1. Clustering
2. *k*-Means Clustering
3. Multi-Start *k*-Means
4. *k*-Means vs. *k*-Medoids
5. Determining *k*

Clustering

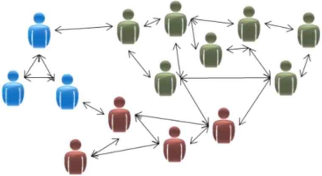
◆ 군집화 (Clustering) 문제

- 유사한 샘플을 모아 같은 그룹으로 묶는 일
- $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$에서 다음 식을 만족하는 군집 집합 $C = \{c_1, c_2, \cdots, c_k\}$를 찾아내는 작업 :

$$\left.\begin{array}{l} c_i \neq \emptyset, i = 1,2,\cdots,k \\ \bigcup_{i=1}^{k} c_i = \mathbb{X} \\ c_i \bigcap c_j = \emptyset, i \neq j \end{array}\right\}$$

- 군집의 개수 k는 주어지는 경우와 자동으로 찾아야 하는 경우가 있음
- 군집화를 부류 발견 작업이라 부르기도 함

Organize computing clusters



Social network analysis

Market segmentation

Astronomical data analysis

Clustering

군집화(Clustering) 문제

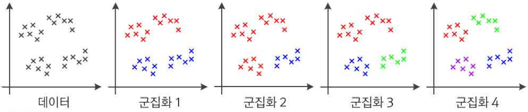
- 유사한 샘플을 모아 같은 그룹으로 묶는 일
- $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$에서 다음 식을 만족하는 군집 집합 $C = \{c_1, c_2, \cdots, c_k\}$를 찾아내는 작업 :

$$\left.\begin{array}{l} c_i \neq \emptyset, i = 1,2,\cdots,k \\ \bigcup_{i=1}^{k} c_i = \mathbb{X} \\ c_i \bigcap c_j = \emptyset, i \neq j \end{array}\right\}$$

- 군집의 개수 k는 주어지는 경우와 자동으로 찾아야 하는 경우가 있음
- 군집화를 부류 발견 작업이라 부르기도 함

군집화의 주관성

- *Which one is correct?*



k-Means Clustering (*k*-평균 군집화)

- ***k*-평균 군집화**
  - 원리는 단순하지만 구현이 쉽고 성능이 좋아 자주 쓰임
  - 군집 개수 *k*를 미리 알고 있어야 함

- **Algorithm (*k*=3 case)**

1) Randomly select *k* distinct points (***initial centers***)

2) ***Assign*** each data point to the *nearest center*.

3) Calculate the ***mean*** of each cluster (***new centers***)

4) Repeat 2) & 3) until the clustering did not change.

알고리즘 6-1 *k*-평균

입력: 훈련집합 $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, 군집의 개수 k

출력: 군집집합 $C = \{c_1, c_2, \cdots, c_k\}$

1 k개의 군집 중심 $Z = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\}$를 초기화한다. 1)
2 while (true)
3 　for (i=1 to n)
4 　　$\mathbf{x}_i$를 가장 가까운 군집 중심에 배정한다. 2)
5 　if (라인 3~4에서 이루어진 배정이 이전 루프에서의 배정과 같으면) break
6 　for (j=1 to k)
7 　　$\mathbf{z}_j$에 배정된 샘플의 평균으로 $\mathbf{z}_j$를 대치한다. 3)
8 for (j=1 to k)
9 　$\mathbf{z}_j$에 배정된 샘플을 c_j에 대입한다.

Figures are from StatQuest: https://www.youtube.com/watch?v=4b5d3muPQmA

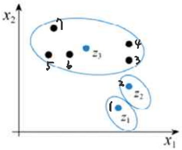
k-Means Clustering (cont'd)

Assignment Matrix (군집 배정 행렬)

- 각 샘플이 어떤 군집에 배정되었는 지를 나타내기 위해 사용
- $\mathbf{A} = [a_{ji}]_{\substack{j=1..k \\ i=1..n}} : (k \times n)$

★ $a_{ji} = 1$: i번째 샘플이 j번째 군집에 **배정된** 경우

★ $a_{ji} = 0$: i번째 샘플이 j번째 군집에 **배정되지 않은** 경우



(sample index)

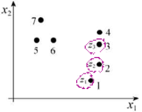
$$\begin{array}{rcl} & \begin{array}{ccccccc} i=1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} & \\ \mathbf{A} = & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} & \left.\begin{array}{l} k=1 \\ 2 \\ 3 \end{array}\right\} \text{cluster index} \end{array}$$

k-Means Clustering (cont'd)

예제 6-1 ***k*-평균의 동작**

[그림 6-5]는 훈련집합이 7개의 샘플을 가진 n=7인 예를 보여 준다. 좌표는 다음과 같다.

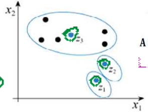
$$\mathbf{x}_1=\begin{pmatrix}18\\5\end{pmatrix},\ \mathbf{x}_2=\begin{pmatrix}20\\9\end{pmatrix},\ \mathbf{x}_3=\begin{pmatrix}20\\14\end{pmatrix},\ \mathbf{x}_4=\begin{pmatrix}20\\17\end{pmatrix},\ \mathbf{x}_5=\begin{pmatrix}5\\15\end{pmatrix},\ \mathbf{x}_6=\begin{pmatrix}9\\15\end{pmatrix},\ \mathbf{x}_7=\begin{pmatrix}6\\20\end{pmatrix}$$



1) 초기 군집 중심
↓
2) 샘플 배정 {A}
↓
3) 새 군집 중심

군집의 개수 k=3이라 하자. 맨 왼쪽 그림은 초기 군집 중심을 보여 준다. [알고리즘 6-1]의 라인 3~4는 7개 샘플을 아래와 같이 배정할 것이다.

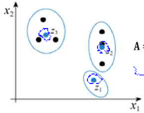
$$\{\mathbf{x}_1\}\text{은 } \mathbf{z}_1,\ \{\mathbf{x}_2\}\text{은 } \mathbf{z}_2,\ \{\mathbf{x}_3,\mathbf{x}_4,\mathbf{x}_5,\mathbf{x}_6,\mathbf{x}_7\}\text{은 } \mathbf{z}_3$$



$$\mathbf{A}=\begin{pmatrix}1&0&0&0&0&0&0\\0&1&0&0&0&0&0\\0&0&1&1&1&1&1\end{pmatrix}$$

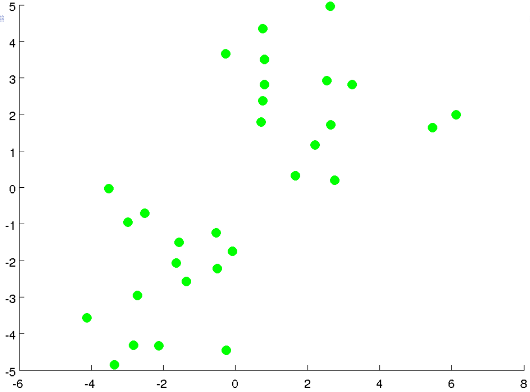
[그림 6-5]의 가운데 그림은 새로 계산한 군집 중심이다. $\mathbf{z}_1=(18,5)^{\mathrm{T}}$, $\mathbf{z}_2=(20,9)^{\mathrm{T}}$, $\mathbf{z}_3=(12,16.2)^{\mathrm{T}}$이고, 식 (6.2)에 대입하면 $J=244.8$이 된다. 이때 거리함수 dist로 식 (1.7)의 유클리디언 거리를 사용한다.

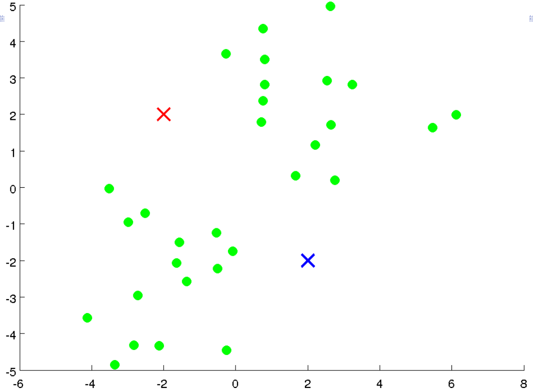
두 번째 루프를 실행하면 행렬 $\mathbf{A}$는 아래와 같이 바뀐다. 군집 중심은 $\mathbf{z}_1=(18,5)^{\mathrm{T}}$, $\mathbf{z}_2=(20,13.333)^{\mathrm{T}}$, $\mathbf{z}_3=(6.667,16.667)^{\mathrm{T}}$이다. 이것을 식 (6.2)에 대입하면 $J=58.0$이 된다. [그림 6-5]의 맨 오른쪽 그림은 두 번째 루프 수행 후의 상황이다.

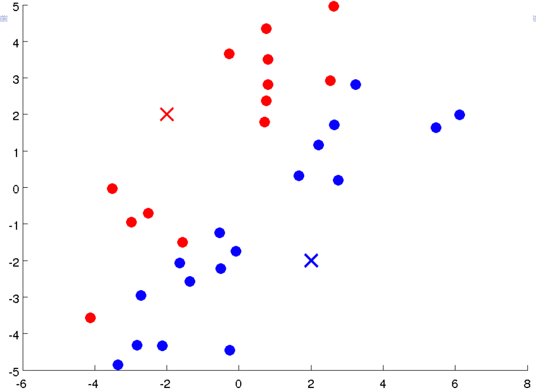


$$\mathbf{A}=\begin{pmatrix}1&0&0&0&0&0&0\\0&1&1&1&0&0&0\\0&0&0&0&1&1&1\end{pmatrix}$$

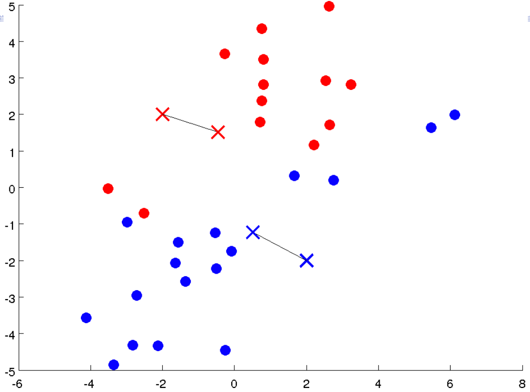
A: 군집 배정 정보를 나타내는 k*n 행렬
(i번째 샘플이 j번째 군집에 배정되었다면 a_{ji}는 1, 그렇지 않으면 0)

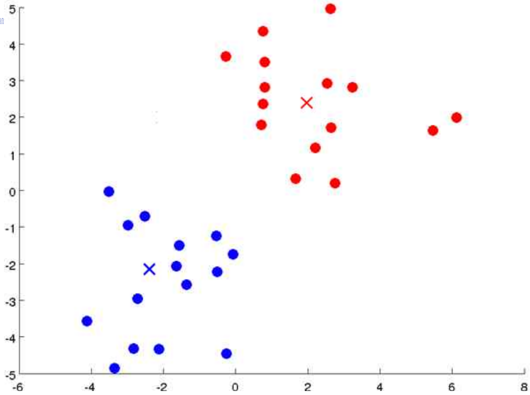

5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8


5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8



Slide credit: Andrew Ng

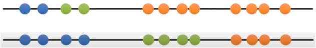

5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8


5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8

Multi-Start *k*-Means (다중시작 *k*-Means)

◆ **Problems of the (Naive) *k*-Means:**

- The clustering result is *very sensitive to the initial cluster* position (초기 군집 중심의 위치에 민감).
  - ★ Note that the result of the intuitive example is pretty terrible compared to what we did by eye.



◆ ***How to make this algorithm better?***

- 여러 개의 초기 군집 중심을 이용하자. (*Multi-Start*)
- 그런데, 어떤 결과가 좋은 결과인 지 어떻게 알지? (*Cost function*을 정의하여 최적화 문제로 만들자!)

Multi-Start *k*-Means (cont'd)

◆ Multi-Start *k*-Means Algorithm

- 서로 다른 초기 군집 중심을 가지고 여러 번 수행한 다음, 가장 좋은 품질의 해를 취함

알고리즘 6-2 다중 시작 k-평균

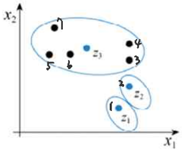
입력: 훈련집합 $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, 군집의 개수 k, 다중 시작 횟수 t
출력: 군집집합 $C = \{c_1, c_2, \cdots, c_k\}$

```
1  for (i=1 to t)
2      X에서 임의로 k개 샘플을 뽑는다.
3      라인 2에서 뽑은 샘플을 초기 군집 중심으로 삼고, [알고리즘 6-1]의 k-평균을 수행한다.
4      k-평균이 출력한 해를 가지고 식 (6.2)의 목적함숫값을 계산한다.
5  t개의 해 중 목적함숫값이 가장 작은 해를 최종해로 취한다.
```

Multi-Start *k*-Means (cont'd)

◆ *k*-Means as an ***Minimization*** Problem

$$\boxed{J(Z,\mathbf{A}) = \sum_{i=1}^{n}\sum_{j=1}^{k} a_{ji}\,dist(\mathbf{x}_i,\mathbf{z}_j)}$$



- Let us define the ***distance*** metric as:

$$dist(\mathbf{x}_i,\mathbf{z}_j) = \|\mathbf{x}_i - \mathbf{z}_j\|_2^2 = (\mathbf{x}_i - \mathbf{z}_j)^T(\mathbf{x}_i - \mathbf{z}_j)$$

then the *J became* the *Within-Cluster Sum of Squares* (**WCSS**).

$$\begin{array}{cc} & \begin{array}{ccccccc} i=1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \mathbf{A} = & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{array}{c} k=1 \\ 2 \\ 3 \end{array} \end{array}$$

- Cf.) Since the total variance is constant, this is equivalent to *maximizing* the *Between-Cluster Sum of Squares* (BCSS).

k-Means vs. *k*-Medoids

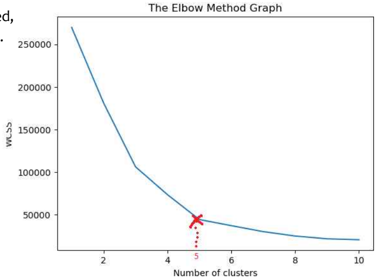
- ***k*-Means**
  - 각 군집에 속한 **샘플의 평균**으로 군집 중심을 갱신

- ***k*-Medoids**
  - 각 군집에 속한 **샘플의 대표**를 뽑아 군집 중심을 갱신 (*k*-Means에 비해 잡음에 둔감)
  - 대표 샘플의 예시: 다른 샘플까지의 거리합이 최소가 되는 샘플, 평균에 가장 가까운 샘플, …

Determining *k*

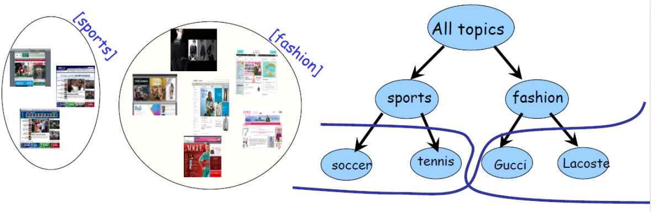
◆ *Elbow Method* for Determining the Optimal *k*

- Find the ***elbow*** (***knee***) of the *cost-vs-k* graphs, *visually*.
- More formal use: an explicit objective function is used, and depends on the particular optimization problem.
- An elbow may also be defined purely geometrically, in terms of the curvature or the second derivative.

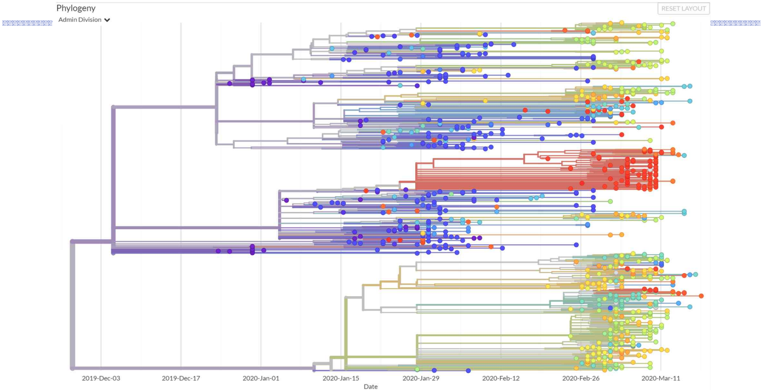


https://medium.com/@sametgirgin/k-means-clustering-model-in-6-steps-with-python-dfe95e5a5fac

Hierarchical Clustering



- A hierarchy might be more nature
- Different users might care about different levels of granularity or even prunings.



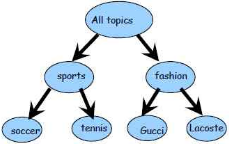
Genomic epidemiology of novel coronavirus https://nextstrain.org/ncov

Hierarchical Clustering

- **Top-down (divisive)**
  - Partition data into 2-groups (e.g., 2-means)
  - Recursively cluster each group

- **Bottom-up (agglomerative)**
  - Start with every point in its own cluster.
  - Repeatedly merge the "closest" two clusters
  - Different definitions of "closest" give different algorithms.

Bottom-up (agglomerative)



- Have a distance measure on pairs of objects.
- $d(x, y)$: Distance between x and y

- Single linkage: $\mathrm{dist}(A, B) = \min_{x \in A, x' \in B} d(x, x')$

- Complete linkage: $\mathrm{dist}(A, B) = \max_{x \in A, x' \in B} d(x, x')$

- Average linkage: $\mathrm{dist}(A, B) = \operatorname*{average}_{x \in A, x' \in B} d(x, x')$

- Ward's method $\mathrm{dist}(A, B) = \frac{|A|\,|B|}{|A|+|B|} \|\mathrm{mean}(A) - \mathrm{mean}(B)\|^2$

Things to remember

- Intro to unsupervised learning
- K-means algorithm
- Optimization objective
- Initialization and the number of clusters
- Hierarchical clustering