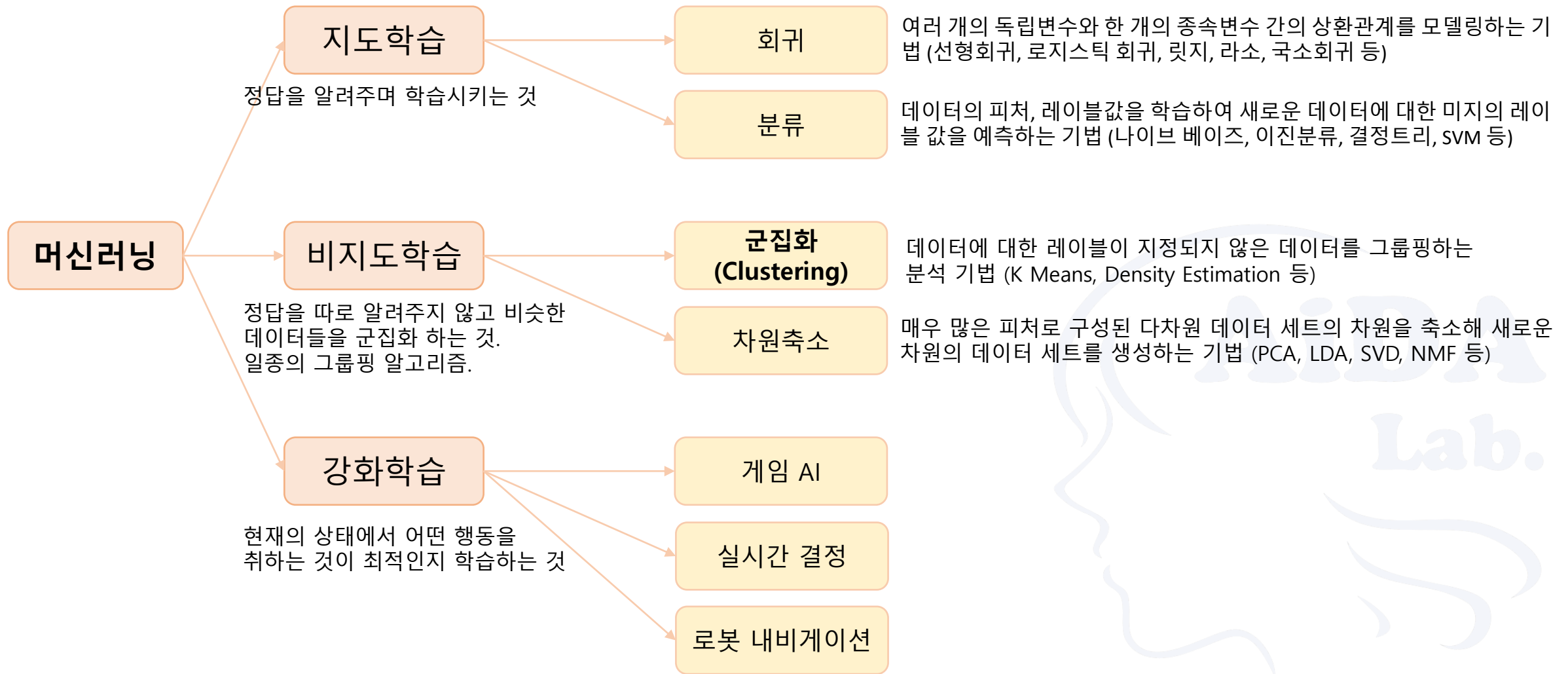


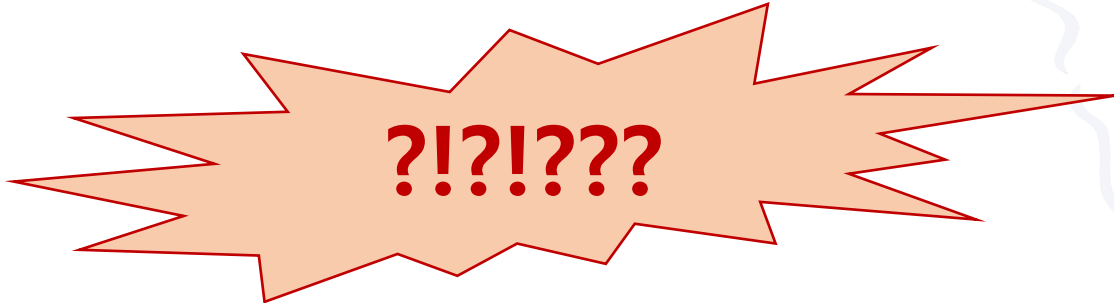
AI 고급 : 자연어 처리 과정(정규)

12. 텍스트 유사도-군집화





- 영수증, 이메일, 여행 일정, 회의록 등과 같은 서류 더미의 내용을 요약해 달라는 요청을 받는다면 무엇을 할 수 있는가?
 - 각 문서를 읽고 각 문서에 가장 관련이 있는 용어/구를 강조 표시한 다음 이 문서들을 모두 한 무더기로 정렬. 한 무더기가 너무 커지면 두 개로 분할..
 - 모든 문서를 검토하고 그룹화하면 각 파일을 더 면밀히 검토 가능
 - 각 무더기의 주요 구나 단어를 사용해 요약을 작성, 고유 명을 파일 주제로..



?!?!???

- 앞의 예시는 실제로 많은 분야에서 이루어지고 있는 작업임
- 이러한 정렬 작업의 핵심은 **두 문서를 비교하여 유사성(Similarity, ≡ 유사도)을 판단할 수 있는 능력**에 달려있음
 - 서로 비슷한 문서는 그룹화
 - 그룹을 통해 전반적인 주제, 토픽, 말뭉치 내부의 패턴 등을 광범위하게 파악
 - 패턴은 이산적(그룹이 전혀 겹치지 않을 때...) 또는 불분명(유사성이 크고 문서를 구별하기 어려울 때...)할 수 있음
 - 어떤 경우라도 결과 그룹은 모두 문서 내용의 모델을 나타냄

- 이와 같은 작업에 비지도 학습을 잘 통합한다면...
- 텍스트에 대한 비지도 학습
 - 종종 말뭉치들은 분류할 수 있게 준비된 레이블에 맞춰 사전에 태깅을 한 채로 전달되지 않음 → 기존의 방식으로서는 제대로 분류되지 않는 경우가 많음
 - 비지도 방식 접근법은 다양한 경우에 있어서 탐색적 텍스트 분석에 크게 유용할 수 있음

- 군집화 알고리즘

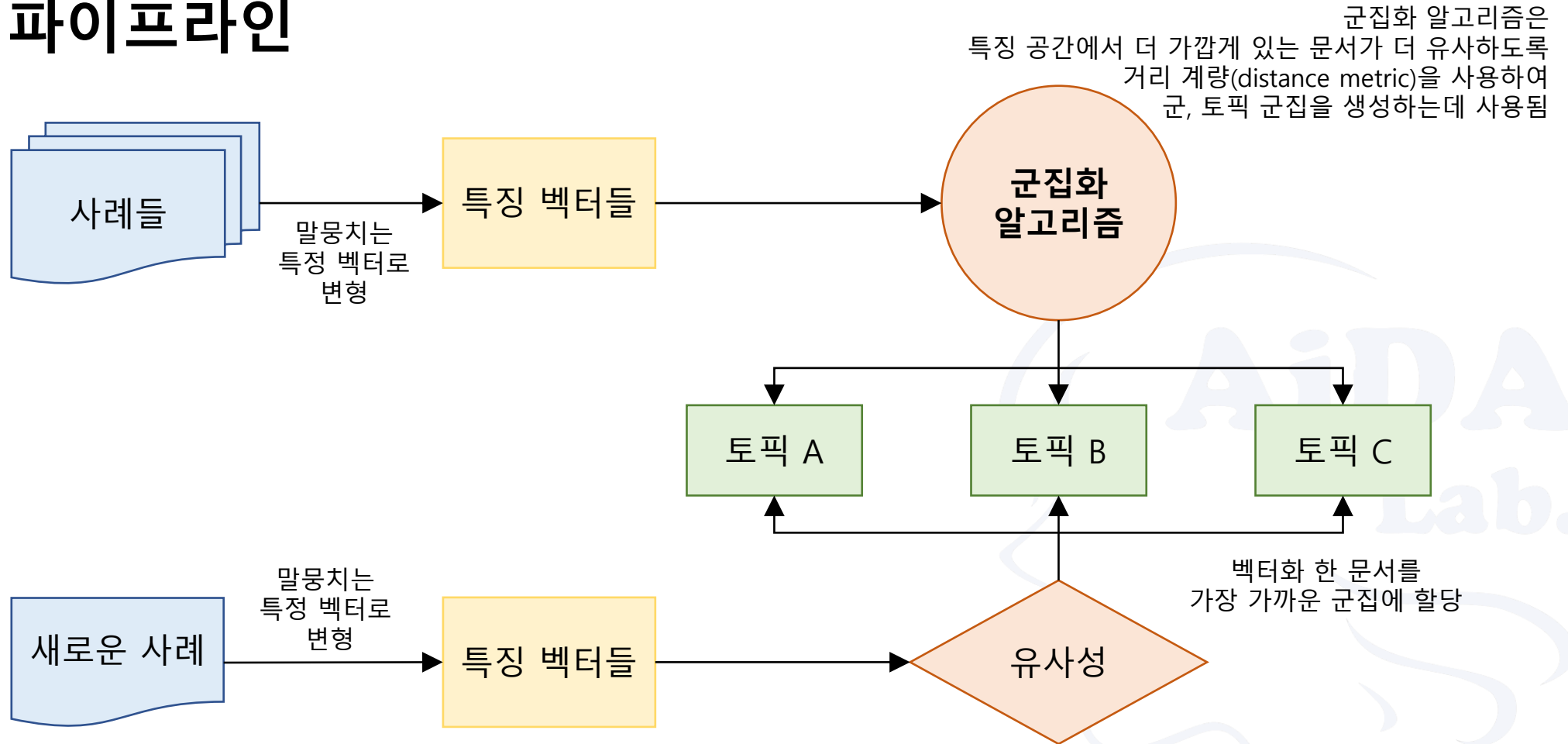
- 알고리즘의 목표

- 주어진 사례(데이터)를
 - 의미가 다른 그룹으로 구성하는 기능을 사용하여
 - 레이블이 지정되지 않은 데이터의 잠재 구조나 테마를 발견하는 것

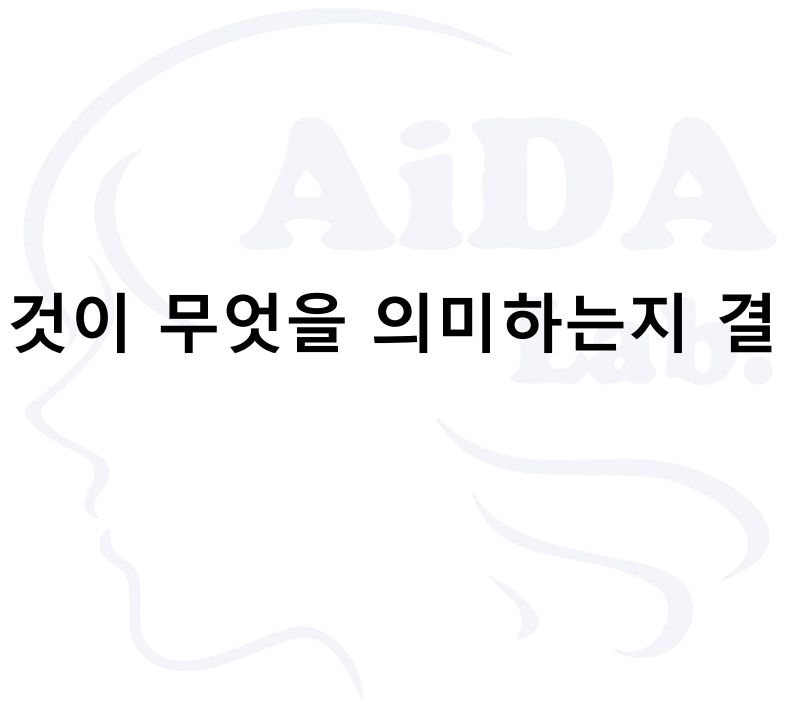
- 텍스트 데이터라면

- 각 사례에 해당하는 것: 단일 문서 또는 발화(말, utterance)
 - 특징에 해당하는 것: 토큰, 어휘, 구조, 메타데이터 등

• 군집화 파이프라인



- 문서의 많은 특징은 단어와 구에서부터 문법과 구조에 이르기까지 모든 분야에서 유사성을 나타낼 수 있다.
- 효과적인 군집화를 위해서는
 - 말뭉치의 두 문서가 유사하거나 유사하지 않다는 것이 무엇을 의미하는지 결정해야 함



• 공간화 유사도 결정 방법(Spatializing Similarity)

문자열 정합 (String Matching, 문자열 일치)	거리 계량 (Distance Metrics)	상대적 정합 (Relation Matching, 상대적 일치)	그 밖의 정합 (Other Matching)
편집거리(Edit Distance) - 레벤슈타인(Levenstein) - 스미스-워터만(Smith-Waterman)	- 유클리드(Euclidean) - 맨해튼(Manhattan) - 민코프스키(Minkowski)	집합 기반 - 다이스(Dice) - 타니모토(Tanimoto), 즉 자카드(Jaccard) - 공통 이웃(Common Neighbors) - 아다르 가중(Adar Weighted)	- 수치 거리(Numeric Distance) - 부울 상등(Boolean Equality) - 퍼지 정합(Fuzzy Matching, 퍼지 매칭) - 영역 특정(Domain Specific)
정렬(Alignment) - 자로-원클러(Jaro-Winkler) - 연성 TFIDF(Soft-TFIDF) - 몬지-엘칸(Monge-Elkan)	텍스트 분석학(Text Analytics) - 자카드(Jaccard) - TFIDF - 코사인 유사도(Cosine Similarity)	총계(Aggregate, 총량) - 평균값(Average Values) - 최댓값/최솟값(Max/Min Value) - 중간값(Mediancs, 중위수) - 빈도(Frequency) 또는 최빈값(Mode)	가제트(Gazettes) - 어휘 정합(Lexical Matching) - 이름 있는 개체(Named Entities) 즉, 개체명 인식(Named Entity Recognition, NER)
음운(Phonetic) - 사운드텍스(Soundex) - 변환(Translation)			

- 거리 계량의 종류

- 유클리드 거리

- 맨해튼 거리

- 자카드 거리

- 민코프스키 거리

- 유클리드 거리와 맨해튼 거리를 일반화한 것

- 정규화된 벡터 공간에서 두 점 사이의 거리를 정의함



• 자카드 거리(Jaccard Distance)

- 유한집합 간의 유사성을 집합들의 교집합과 합집합의 몫으로 정의
- 두 문서 A, B에서 양쪽 모두에 나타나는 고유단어의 수를 A, B에 나타나는 고유 단어의 총 수로 나누어 측정
- 값 0은 두 문서가 공통점이 없음, 값 1은 두 문서가 동일한 문서임을 의미
- 0~1 사이의 값은 상대적 유사성을 표현



$A \cap B$



$A \cup B$

• 마할라노비스 거리

참고: https://gaussian37.github.io/ml-concept-mahalanobis_distance/

- 특정 지점이 점의 분포로부터 얼마나 많은 표준편차를 벗어났는지 측정할 수 있게 다차원으로 일반화한 계량 방법. 분포와 관련해 좌표이동, 재조정 효과가 있음

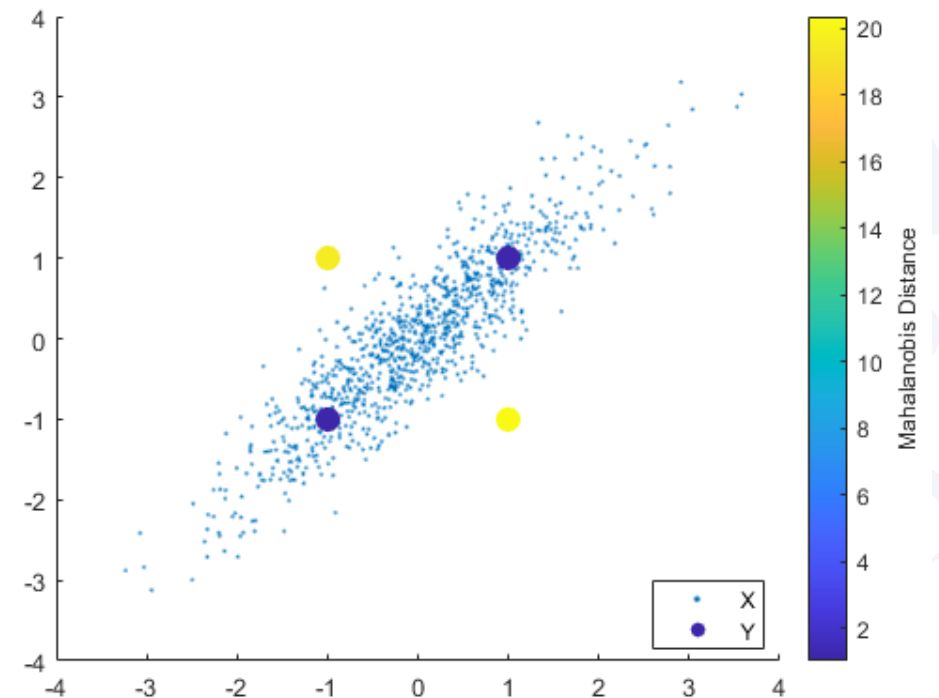
- 표본 점과 분포 사이의 거리를 측정한 값

- 마할라노비스 거리 = $\left((x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) \right)^{0.5}$

마할라노비스 거리는 x 에서 μ_i 까지의 거리이며, 더 정확히 말하면 x 에서 정규분포 $N(\mu_i, \Sigma)$ 까지의 거리가 됨

→ 기존의 유클리디안 거리에 공분산 계산이 추가된 것으로 볼 수 있음

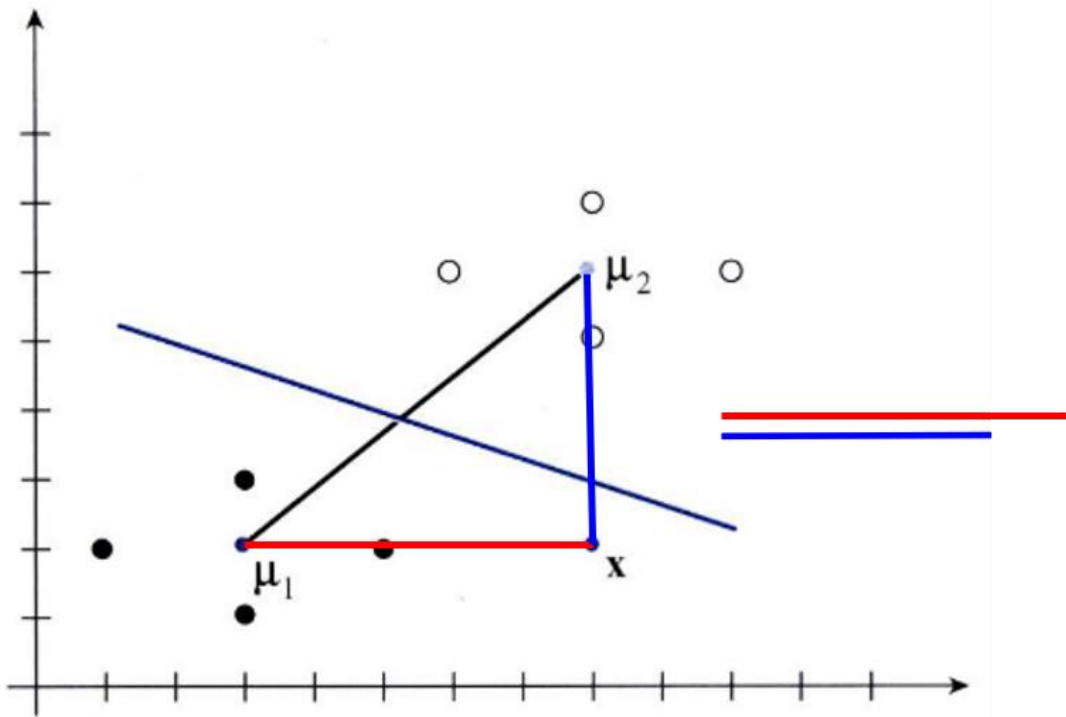
유클리디안 거리 : $\left((x - \mu_i)^T (x - \mu_i) \right)^{0.5}$



공분산 값

$$\Sigma_1 = \begin{pmatrix} ((1-3)^2 + (3-3)^2 + (5-3)^2 + (3-3)^2)/(4-1) & ((1-3)(2-2) + (3-3)(1-2) + (5-3)(2-2) + (3-3)(3-2))/(4-1) \\ ((2-2)(1-3) + (1-2)(3-3) + (2-2)(5-3) + (3-2)(3-3))/(4-1) & ((2-2)^2 + (1-2)^2 + (2-2)^2 + (3-2)^2)/(4-1) \end{pmatrix}$$

$$\Sigma_2 = \begin{pmatrix} ((6-8)^2 + (8-8)^2 + (10-8)^2 + (8-8)^2)/(4-1) & ((6-8)(6-6) + (8-8)(5-6) + (10-8)(6-6) + (8-8)(7-6))/(4-1) \\ ((6-6)(6-8) + (5-6)(8-8) + (6-6)(10-8) + (7-6)(8-8))/(4-1) & ((6-6)^2 + (5-6)^2 + (6-6)^2 + (7-6)^2)/(4-1) \end{pmatrix}$$



$$\Sigma = \begin{pmatrix} 8/3 & 0 \\ 0 & 2/3 \end{pmatrix}$$

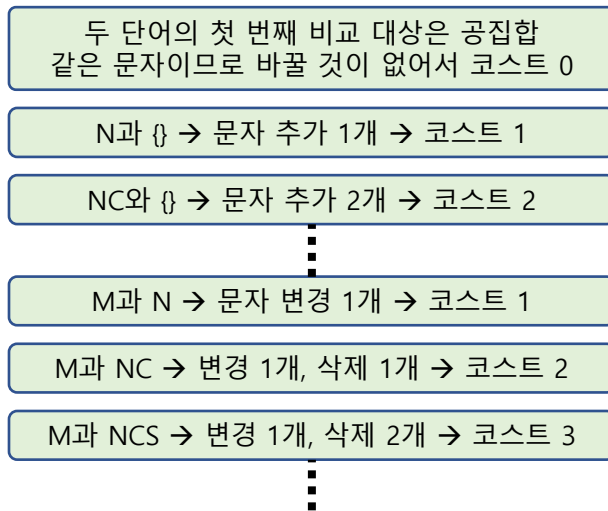
가우시안 분포와 분별함수에 의해 계산됨 (상세 내용은 패스)

$$d(x, \mu_1) = \left((8-3 \quad 2-2) \begin{pmatrix} 3/8 & 0 \\ 0 & 3/2 \end{pmatrix} \begin{pmatrix} 8-3 \\ 2-2 \end{pmatrix} \right)^{1/2} = 3.062$$

$$d(x, \mu_2) = \left((8-8 \quad 2-6) \begin{pmatrix} 3/8 & 0 \\ 0 & 3/2 \end{pmatrix} \begin{pmatrix} 8-8 \\ 2-6 \end{pmatrix} \right)^{1/2} = 4.899$$

• 편집거리(Edit Distance)

- 두 문자열의 유사도를 판단하는 알고리즘(주요 응용 대상은 자연어 처리 분야)
- 어떤 문자열을 삽입, 삭제, 변경을 몇 번이나 해서 바꿀 수 있는지 계산하여 그 최소값을 구한 후, 그 값을 유사도 판단의 척도로 다룸
- 예) 문자열 MICROSOFT와 NC소프트를 비교해보자.



	{}	N	C	S	O	F	T
{}	0	1	2	3	4	5	6
M	1						
I	2						
C	3						
R	4						
O	5						
S	6						
O	7						
F	8						
T	9						

	{}	N	C	S	O	F	T
{}	0	1	2	3	4	5	6
M	1	1	2	3	4	5	6
I	2	2	2	3	4	5	6
C	3	3	2	3	4	5	6
R	4						
O	5						
S	6						
O	7						
F	8						
T	9						

- 모든 문자를 직접 비교, 계산하지 않고 이전에 비교했던 값이 있을 경우, 그 때의 편집거리를 가지고 와서 사용할 수 있음
 - MICROSOFT와 NCISOFT의 마지막 T끼리 비교했을 때, T는 일치하므로 그 전까지 비교했던 MICROSOFT와 NCISOFT의 편집거리를 가져오면 됨
 - MICRO와 NCS를 비교할 경우, O와 S는 서로 다르므로 MICR과 NC를 비교했을 때의 편집거리에 1을 증가시키면 됨
- 따라서
 - $A[i] = B[j]$ 일 때의 편집거리 $D(i, j) = D(i - 1, j - 1)$
 - $A[i] \neq B[j]$ 일 때의 편집거리 $D(i, j) = \min(D(i - 1, j) + 1, D(i, j - 1) + 1, D(i - 1, j - 1) + 1)$

참고: <https://hsp1116.tistory.com/41>

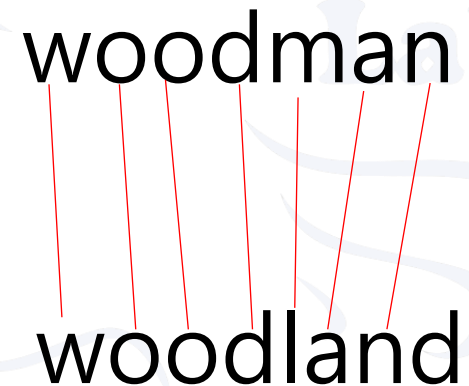
- 레벤슈테인 거리(Levenshtein Distance)

- 두 시퀀스의 차이를 측정하기 위한 문자열 메트릭
- 두 단어 사이의 레벤슈타인 거리는 한 단어를 다른 단어로 변경하는 데 필요한 최소 단일 문자 편집 (삽입, 삭제 또는 대체) 회수(코스트)
- 편집거리로 지칭되기도 함

$$\text{lev}(a, b) = \begin{cases} |a| & \text{if } |b| = 0, \\ |b| & \text{if } |a| = 0, \\ \text{lev}(\text{tail}(a), \text{tail}(b)) & \text{if } a[0] = b[0] \\ 1 + \min \begin{cases} \text{lev}(\text{tail}(a), b) \\ \text{lev}(a, \text{tail}(b)) \\ \text{lev}(\text{tail}(a), \text{tail}(b)) \end{cases} & \text{otherwise,} \end{cases}$$

woodman : woodland
하나의 삽입과 하나의 대체에 대한 페널티 포함

woodman
woodland



- 해밍 거리

- 곱집합 위에 정의되는 거리 함수
- 같은 길이의 두 문자열에서, 같은 위치에서 서로 다른 값이 몇 개인지 센다.

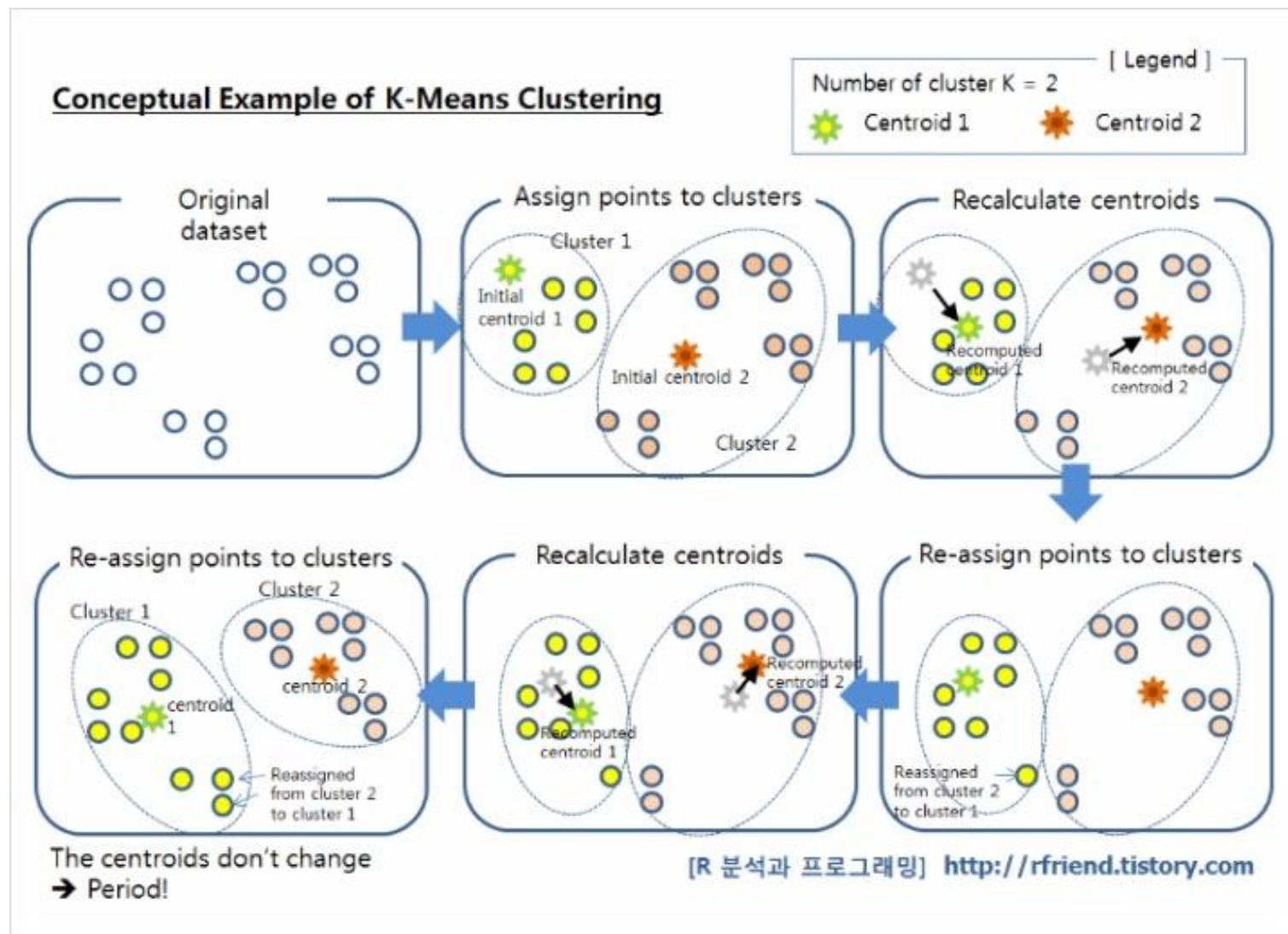


- 두 문서 간의 유사성을 정량화할 수 있으면 유사한 문서 그룹을 찾기 위하여 비지도 학습을 적용할 수 있음
- 비지도 학습에서의 두 가지 주요 접근법
 - 부분 군집화(Partitive Clustering)
 - 계층적 군집화(Hierarchical Clustering)



- k-Means Clustering (k 평균 군집화)
 - 군집화에서 가장 일반적으로 사용되는 알고리즘
 - 군집의 중심점이라는 특정한 임의의 지점을 선택해 해당 중심에 가장 가까운 포인트들을 선택하는 기법
 - 임의로 선택한 군집 수 k 및 벡터화된 사례(instance)들을 중심(centroid)에 근접하도록 군집(Clusters)들로 분할한 후, 이를 계산하여 군집 내부의 제곱합을 최소화 함

부분 군집화: k-Means Clustering



• 기본적인 알고리즘

(실제 사용 시에는 주어진 문제/데이터에 적합하게 초기화하여 생성한 임의의 위치를 적용)

1. 군집화의 기준이 되는 중심점을 구성하려는 군집의 개수만큼 임의의 위치에 배치
2. 각 데이터는 가장 가까운 곳에 위치한 중심점에 소속되어 군집을 형성
3. 데이터들의 소속이 결정되면 군집의 중심점을 소속된 데이터 군집의 평균 중심으로 이동
4. 전체 데이터 중 기존 중심점보다 더 가까운 중심점이 생긴 데이터는 더 가까운 중심점으로 소속 변경 → 군집 조정
5. 조정된 군집에서 중심점을 다시 소속된 데이터의 평균 중심으로 이동
6. 이 과정을 반복. 중심점을 이동했는데 소속이 변경되는 데이터가 없으면 종료.

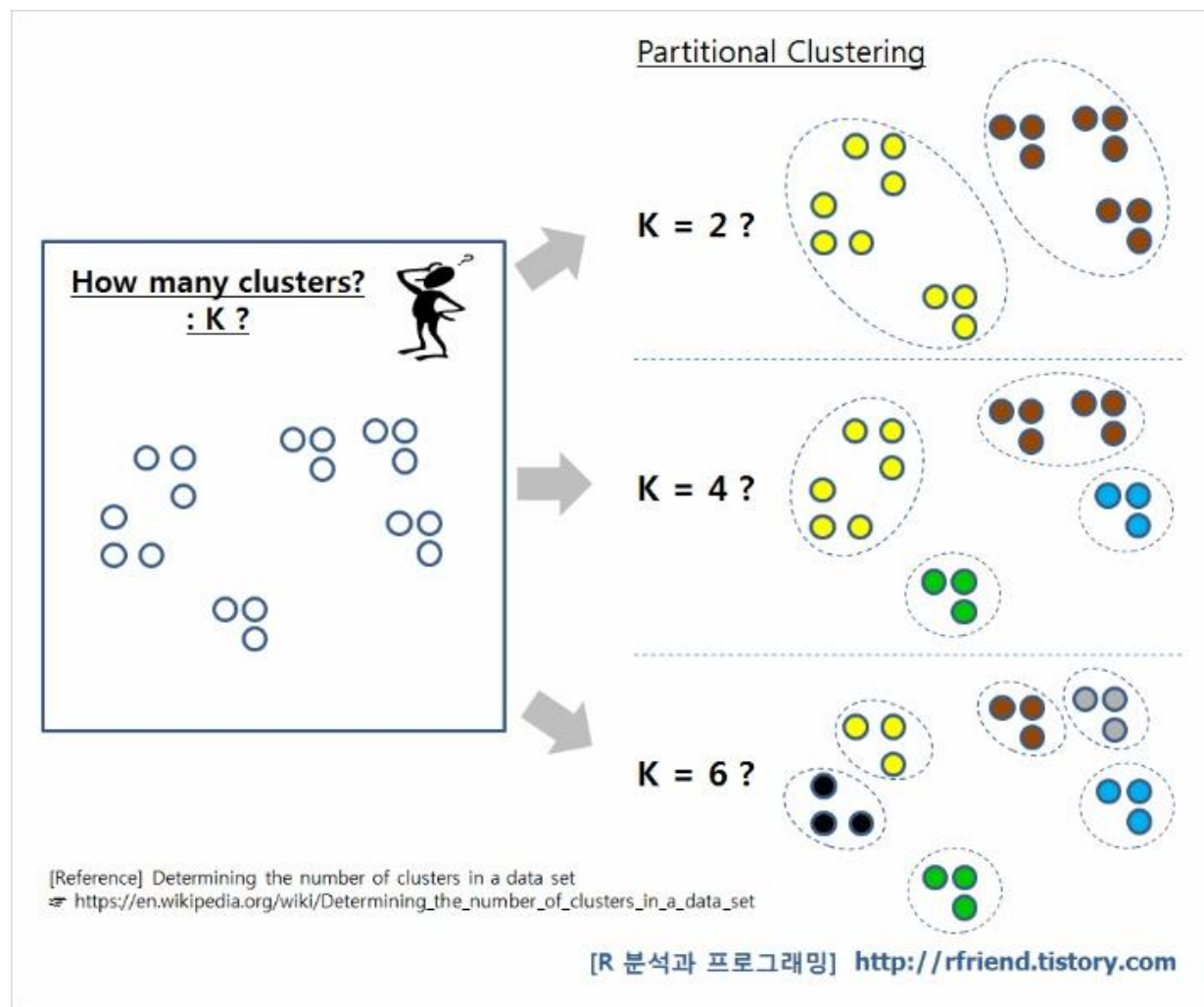
- **k-평균 군집화의 장점**

- 가장 많이 사용됨
- 알고리즘이 쉽고 간결함

- **k-평균 군집화의 단점**

- 거리 기반 알고리즘이므로 속성의 개수가 매우 많을 경우 군집화의 정확도가 떨어짐
→ 개선 방안으로 PCA를 이용한 차원감소 등을 고려할 수 있음
- 반복을 수행하는데, 반복 횟수가 많을 경우 수행 시간이 매우 느려짐
- 몇 개의 군집(Cluster)을 선택해야 할지 적절한 기준, 가이드가 없음

부분 군집화: k-Means Clustering



부분 군집화: k-Means Clustering



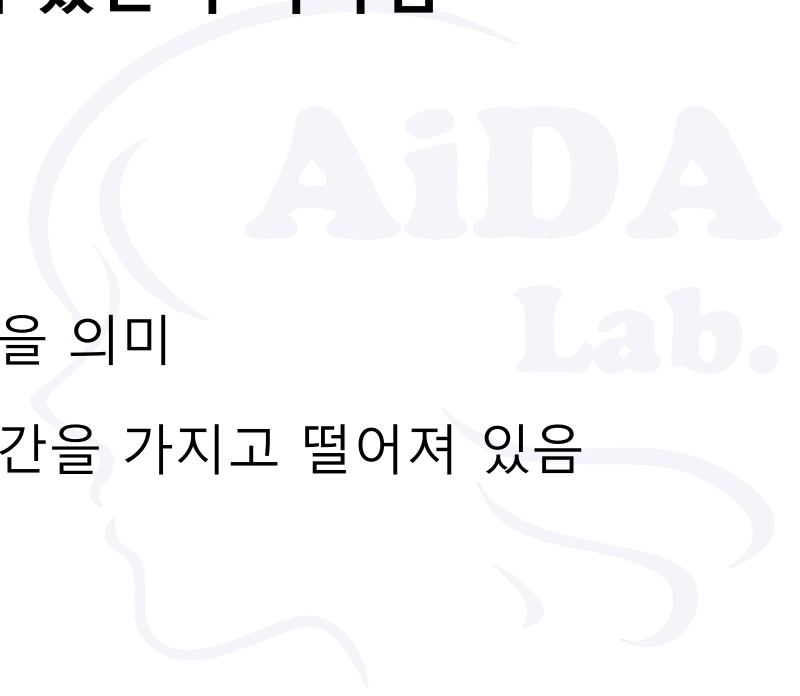
- 실습



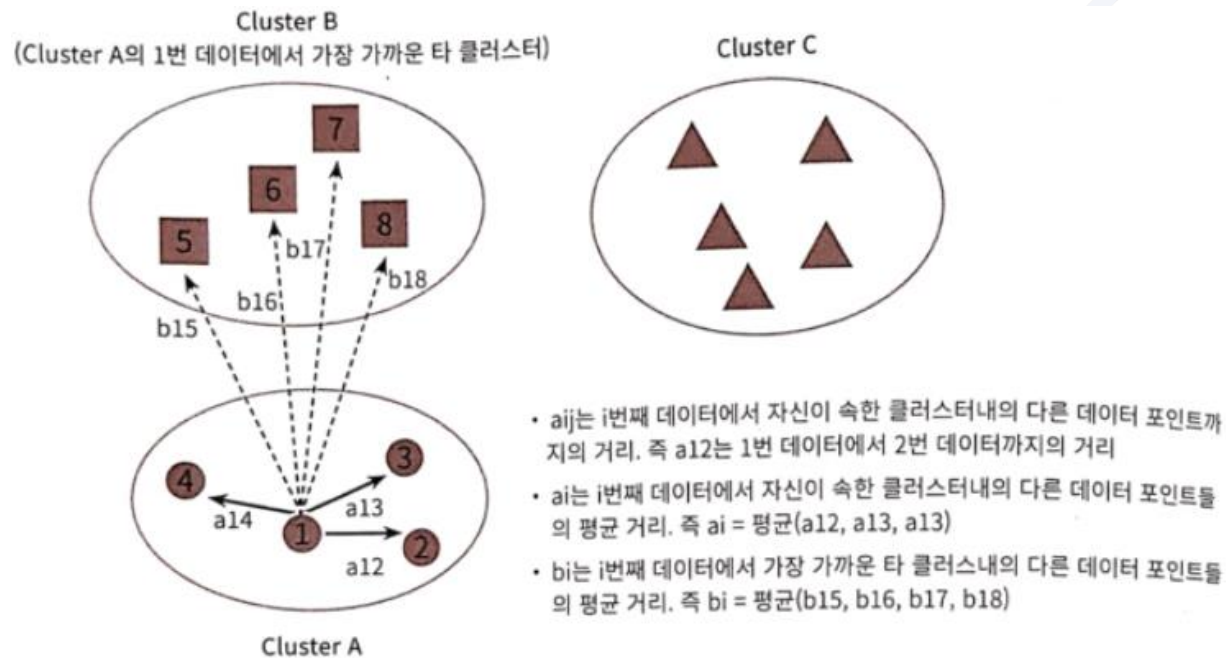
- 군집 평가(Cluster Evaluation)

- 지도학습, 또는 붓꽃 데이터 세트의 실습의 경우, 결과 레이블이 제공되지만 일반적인 데이터 군집화 세트는 타겟 레이블이 없음
- 군집화는 분류와 비슷해 보이지만 성격이 많이 다른 알고리즘
- 따라서 군집화가 효율적으로 잘 되었는지 평가할 수 있는 지표가 요구됨

- 실루엣 분석(Silhouette Analysis)
 - 군집화 평가 방법의 하나
 - 각 군집 간의 거리가 얼마나 효율적으로 분리되어 있는지 나타냄
 - 효율적으로 잘 분리되었다는 것은...
 - 다른 군집과의 거리는 충분히 떨어져 있고
 - 동일 군집끼리의 데이터는 서로 가깝게 잘 뭉쳐져 있음을 의미
 - 군집화가 잘 될수록 개별 군집은 비슷한 정도의 여유공간을 가지고 떨어져 있음



- 실루엣 계수(Silhouette Coefficient)를 기반으로 함
 - 개별 데이터가 가지는 군집화 지표
 - 실루엣 계수: 해당 데이터가 같은 군집 내의 데이터와 얼마나 가깝게 군집화 되어 있고, 다른 군집에 있는 데이터와는 얼마나 멀리 분리되어 있는지를 나타내는 지표



- 특정 데이터 포인트의 실루엣 계수 값: $a(i)$ 와 $b(i)$ 를 기반으로 계산함
 - $a(i)$: 해당 데이터 포인트와 같은 군집 내에 있는 다른 데이터 포인트와의 거리를 평균한 값
 - $b(i)$: 해당 데이터 포인트가 속하지 않은 군집 중 가장 가까운 군집과의 평균 거리
 - $b(i) - a(i)$: 두 군집 간의 거리가 얼마나 떨어져 있는가의 값
 - 정규화를 위해서 $\max(a(i), b(i))$ 값으로 나눠주면 $\rightarrow i$ 번째 데이터 포인트의 실루엣 계수

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

- 실루엣 계수는 -1~1 사이의 값을 가지며
 - 1에 가까울수록 근처의 군집과 더 멀리 떨어져 있음
 - 0에 가까울수록 근처의 군집과 가까워짐
 - (마이너스)값은 아예 다른 군집에 데이터 포인트가 할당되었음을 의미

- 좋은 군집화 조건

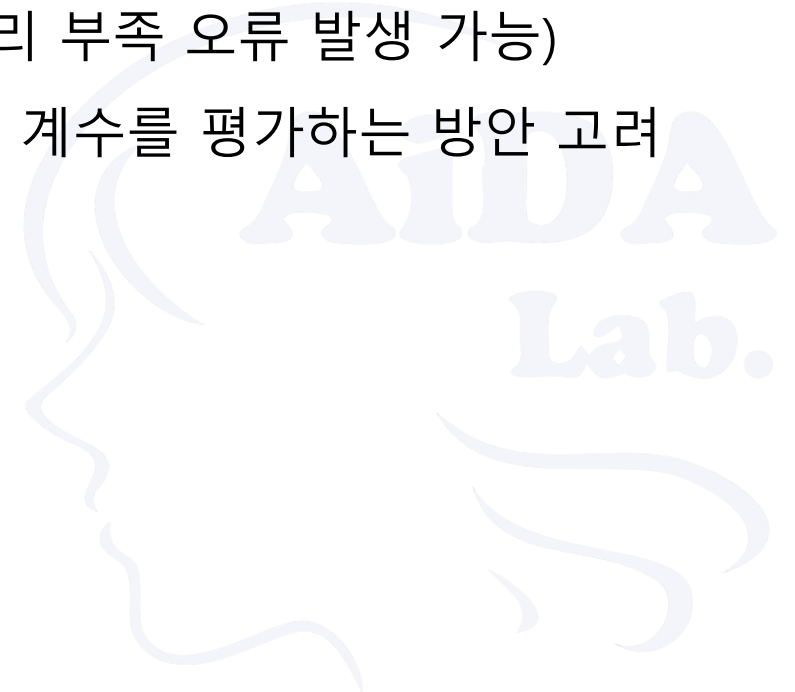
- 전체 실루엣 계수의 평균값은 0~1 사이의 값을 가지며 1에 가까울수록 좋음
- 전체 실루엣 계수의 평균값과 개별 군집의 평균값의 편차가 크지 않아야 함
 - 개별 군집의 실루엣 계수 평균값이 전체 실루엣 계수의 평균값에서 크게 벗어나지 않는 것이 중요함
 - 만약 전체 실루엣 계수의 평균값은 높지만, 특정 군집의 실루엣 계수 평균값만 유난히 높고, 다른 군집의 실루엣 계수 평균값은 낮다면 좋은 군집화 조건이라고 할 수 없음

- 실습



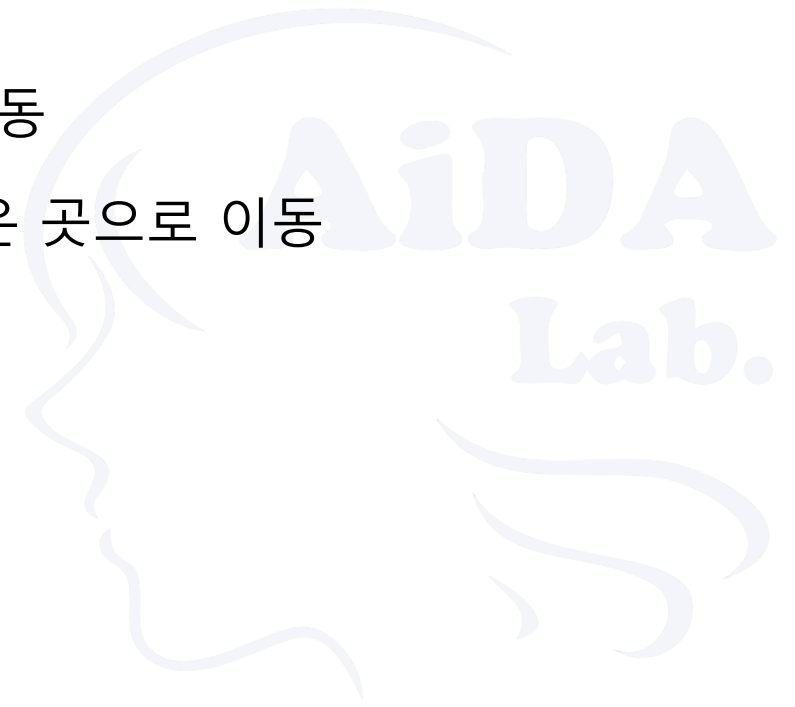
- 군집별 평균 실루엣 계수의 시각화를 통한 군집 개수 최적화 방법
 - 전체 데이터 평균 실루엣 계수 값이 높다고 해서 반드시 최적의 군집 계수로 군집화가 잘 되었다고 볼 수는 없음
 - 특정 군집 내의 실루엣 계수 값만 너무 높고, 다른 군집은 내부 데이터끼리의 거리가 너무 떨어져 있어서 실루엣 계수 값이 낮아져도 평균값은 높아질 수 있음
 - 개별 군집별로 적당히 분리된 거리를 유지하면서 군집 내의 데이터가 서로 뭉쳐있는 경우에 k-평균의 적절한 군집 개수가 설정되었다고 판단할 수 있음

- 실루엣 계수를 통한 k-평균 군집 평가 방법
 - 직관적으로 이해하기 쉬움
 - 각 데이터 별로 다른 데이터와의 거리를 반복적으로 계산해야 함
 - 데이터 양이 늘어나면 수행시간이 크게 증가함(메모리 부족 오류 발생 가능)
 - 이 경우, 군집별로 임의의 데이터를 샘플링해 실루엣 계수를 평가하는 방안 고려



- 평균 이동(Mean Shift)

- 중심을 군집의 중심으로 지속적으로 움직이면서 군집화 수행
- k-평균과 유사하지만
 - k-평균: 중심에 소속된 데이터의 평균 거리 중심으로 이동
 - 평균 이동: 중심을 데이터가 모여 있는 밀도가 가장 높은 곳으로 이동

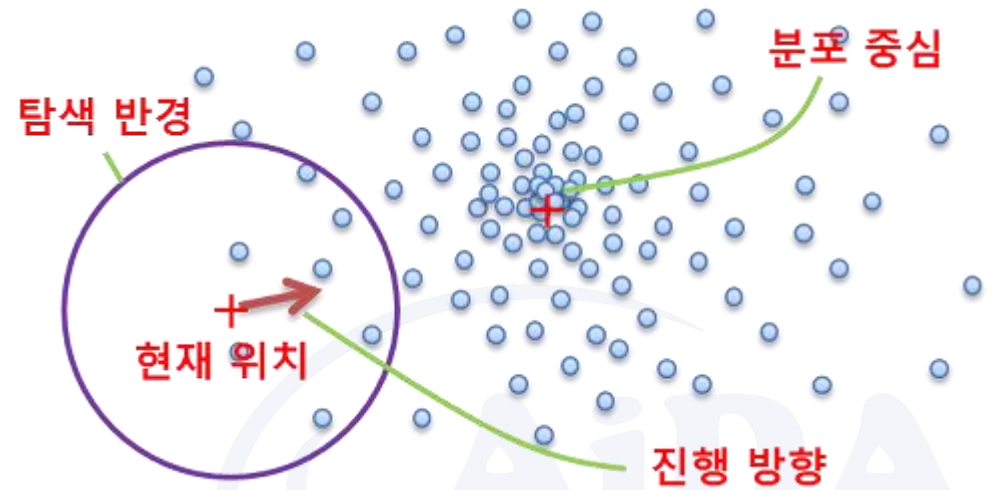


- 평균 이동 군집화는

- 데이터의 분포도를 이용해 군집 중심점을 찾음 (\because 군집 중심점은 데이터가 모여 있는 곳)
 - 확률 밀도 함수(Probability Density Function) 이용
- 가장 집중적으로 데이터가 모여 있어 확률 밀도 함수가 피크인 점을 군집 중심점으로 선정
 - 주어진 모델의 확률 밀도를 찾기 위해서 KDE(Kernel Density Estimation)을 많이 이용
- 특정 데이터를 반경 내의 데이터 분포 확률 밀도가 가장 높은 곳으로 이동하기 위해 주변 데이터와의 거리 값을 KDE 함수 값으로 입력,
그 반환 값을 현재 위치에서 업데이트하면서 이동
 - 이 방식을 전체 데이터에 반복적으로 적용하면서 데이터의 군집 중심점을 찾음

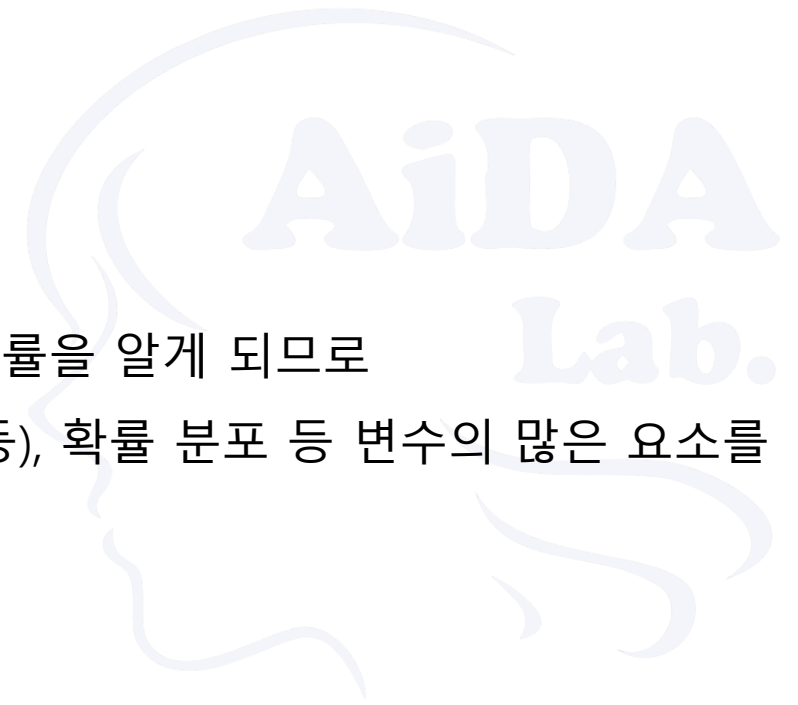
• 알고리즘

1. 개별 데이터의 특정 반경 내에 주변 데이터를 포함한 데이터 분포도를 KDE 기반의 Mean Shift 알고리즘으로 계산
2. KDE로 계산된 데이터 분포도가 높은 방향으로 데이터 이동
3. 모든 데이터를 1~2까지 수행하면서 데이터 이동, 개별 데이터들이 군집중심점으로 모임
4. 지정된 반복 횟수만큼 전체 데이터에 대해서 KDE 기반으로 데이터를 이동시키면서 군집화 수행
5. 개별 데이터들이 모인 중심점을 군집 중심점으로 설정



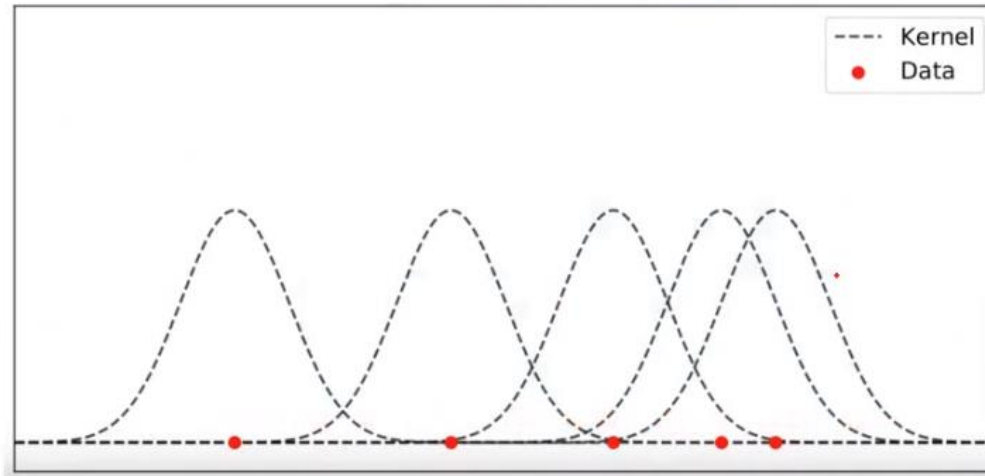
- KDE(Kernel Density Estimation)

- 커널 함수를 통해 어떤 변수의 확률 밀도 함수를 추정하는 대표적인 방법
- 관측된 데이터 각각에 커널 함수를 적용한 값을 모두 더한 뒤, 데이터 건수로 나눠 확률 밀도 함수(Probability Density Function, PDF)를 추정함
- PDF: 확률 변수의 분포를 나타내는 함수
 - 정규 분포 함수, 감마 분포, t-분포 등이 있음
 - PDF를 알면 특정 변수가 어떤 값을 갖게 될 지에 대한 확률을 알게 되므로
 - 이를 통해 변수의 특성(정규 분포의 경우 평균, 분산 등), 확률 분포 등 변수의 많은 요소를 알 수 있음
- 대표적인 커널 함수: 가우시안 분포 함수

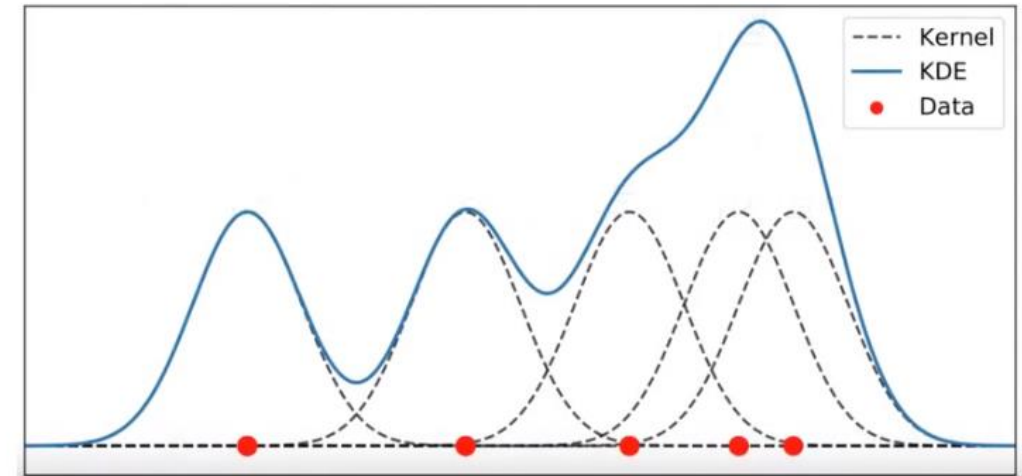


- 가우시안 함수의 적용

개별 관측 데이터에 가우시안 커널 함수 적용



가우시안 커널 함수 적용 후 합산



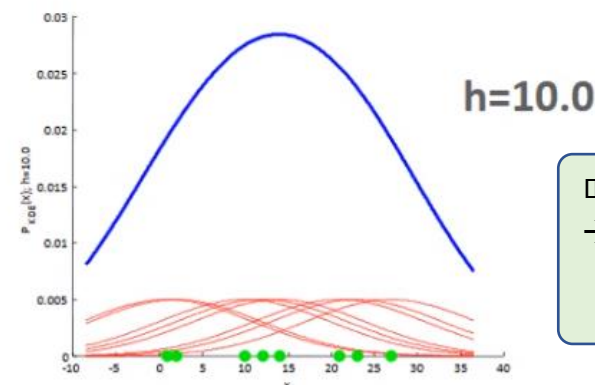
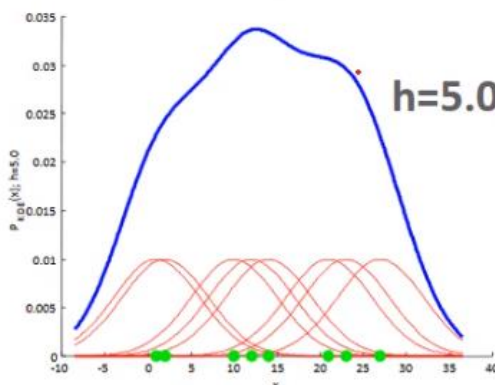
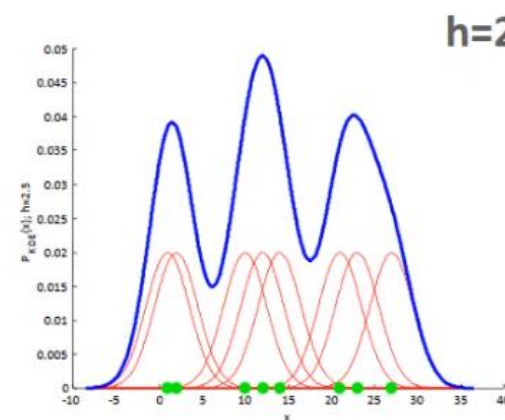
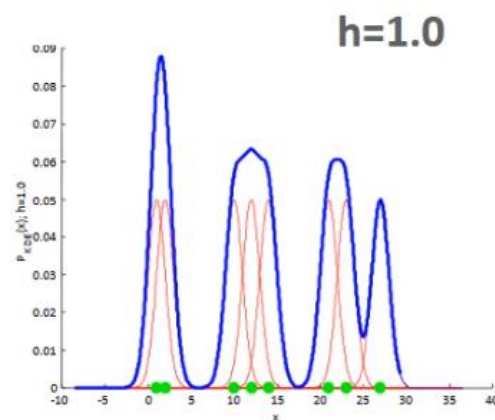
$$KDE = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$

K : 커널 함수,
 x : 확률 변수 값,
 x_i : 관측 값,
 h : 대역폭

부분 군집화: Mean Shift

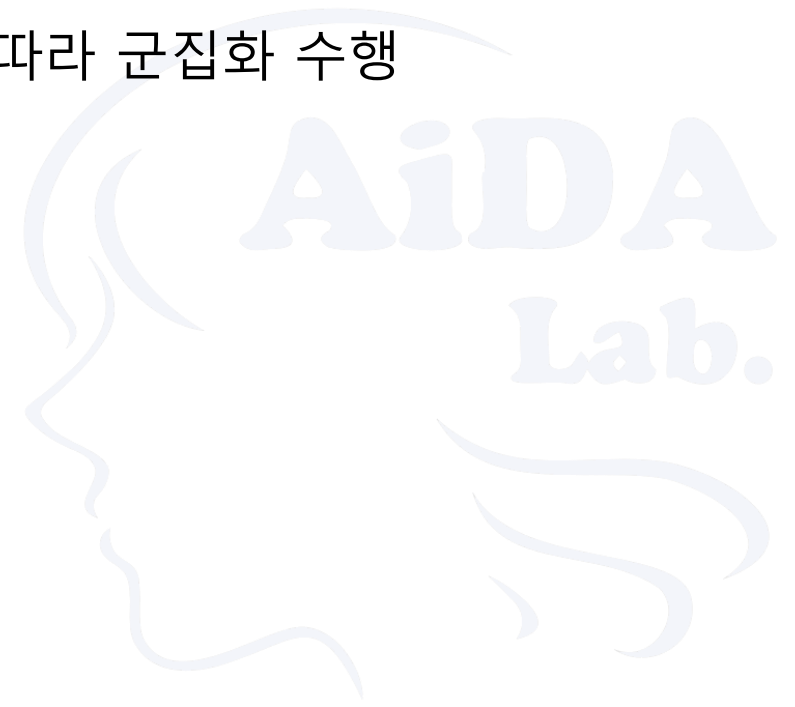
- 대역폭 h 는 KDE 형태를 부드러운(또는 뾰족한) 형태로 평활화(Smoothing)하는데 적용
- h 를 어떻게 설정하느냐에 따라 확률 밀도 추정 성능을 크게 좌우할 수 있음

작은 h 값: 좁고 뾰족한 KDE
→ 변동성이 큰 방식으로 확률
밀도 함수를 추정하므로
과적합 하기 쉬움



매우 큰 h 값: 과도하게 평활화된 KDE
→ 지나치게 단순화된 방식으로 확률
밀도 함수를 추정하므로 과소적합
하기 쉬움

- 일반적으로 평균 이동 군집화는
 - 대역폭이 클수록 평활화된 KDE로 인해 적은 수의 군집 중심점을 가짐
 - 대역폭이 작을수록 많은 수의 군집 중심점을 가짐
 - 군집의 개수를 지정하지 않으며 오직 대역폭의 크기에 따라 군집화 수행



- **평균 이동 군집화의 장점**

- 데이터 세트의 형태를 특정 형태로 가정하거나, 특정 분포도 기반의 모델로 가정하기 않기 때문에 유연한 군집화가 가능함
- 이상치의 영향력이 크지 않고 미리 군집의 개수를 정할 필요가 없음

- **평균 이동 군집화의 단점**

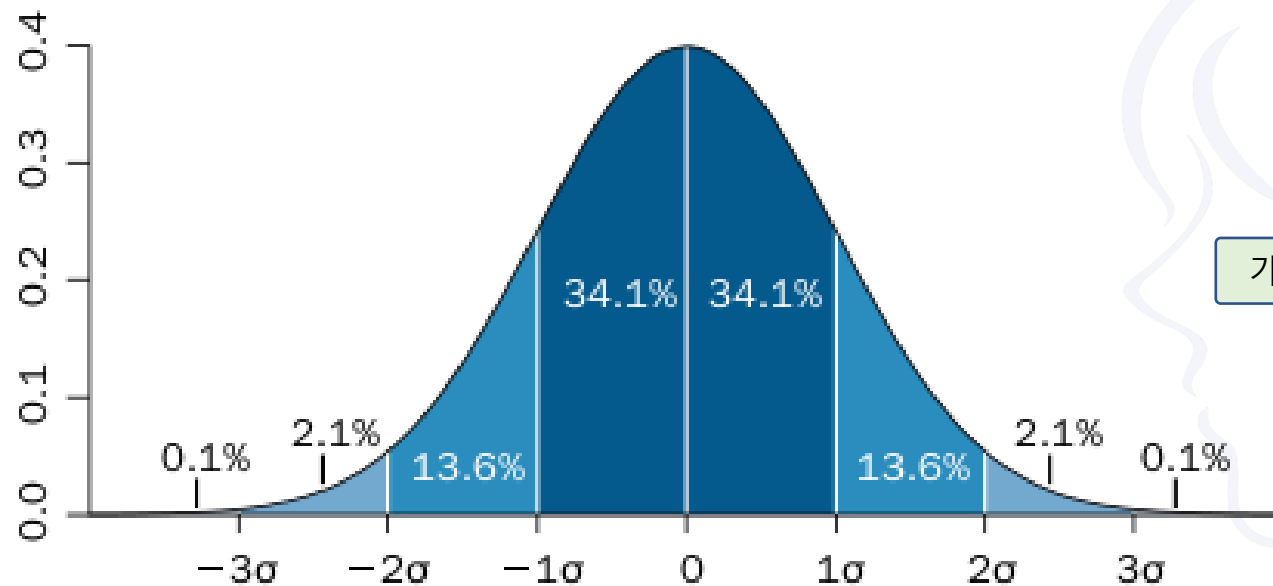
- 알고리즘의 수행 시간이 길다
- band-width의 크기에 따른 군집화 영향도가 매우 크다

- **이런 이유로 평균이동 군집화 기법은 분석 업무 기반의 데이터 세트보다는 컴퓨터 비전 영역에서 더 많이 사용됨**

- 영상, 이미지에서 특정 개체 구분, 움직임 추적에 뛰어남

- GMM (Gaussian Mixture Model)

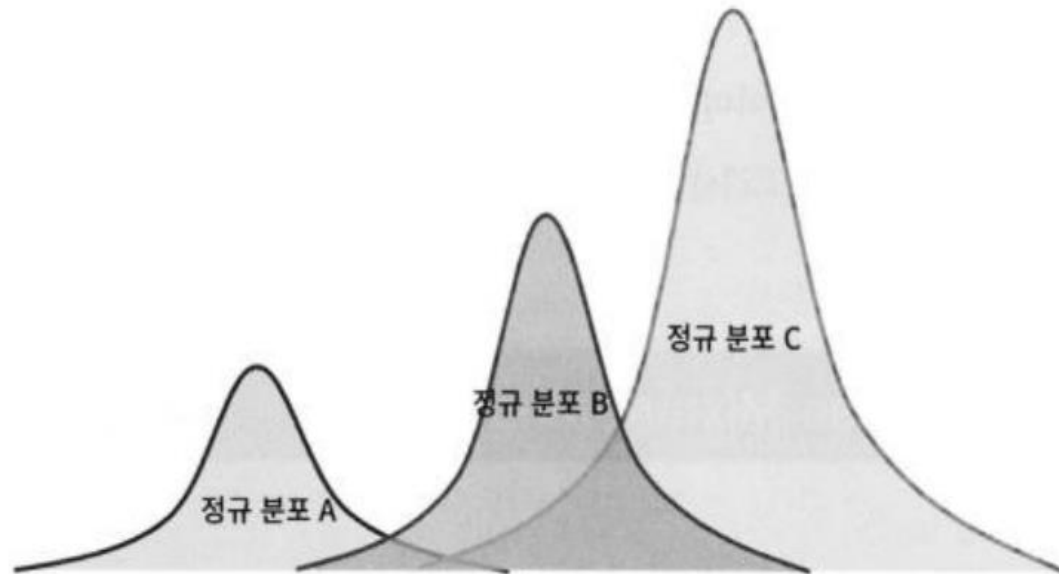
- 군집화를 적용하고자 하는 데이터가 여러 개의 가우시안 분포를 가진 데이터 집합들이 섞여서 생성된 것이라는 가정 하에 군집화를 수행하는 방법



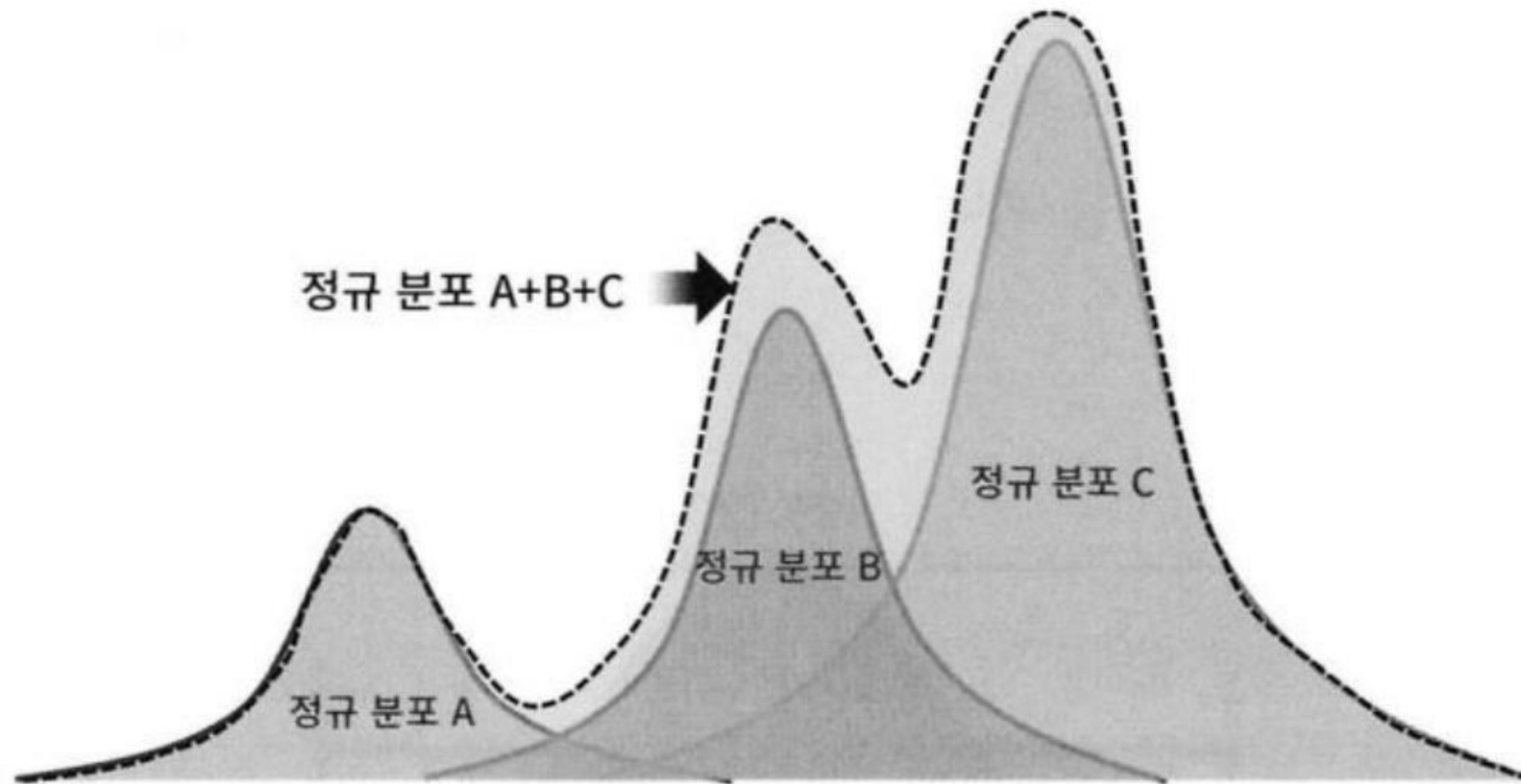
가우시안 확률분포(=정규분포)

- GMM(가우시안 혼합 모델)

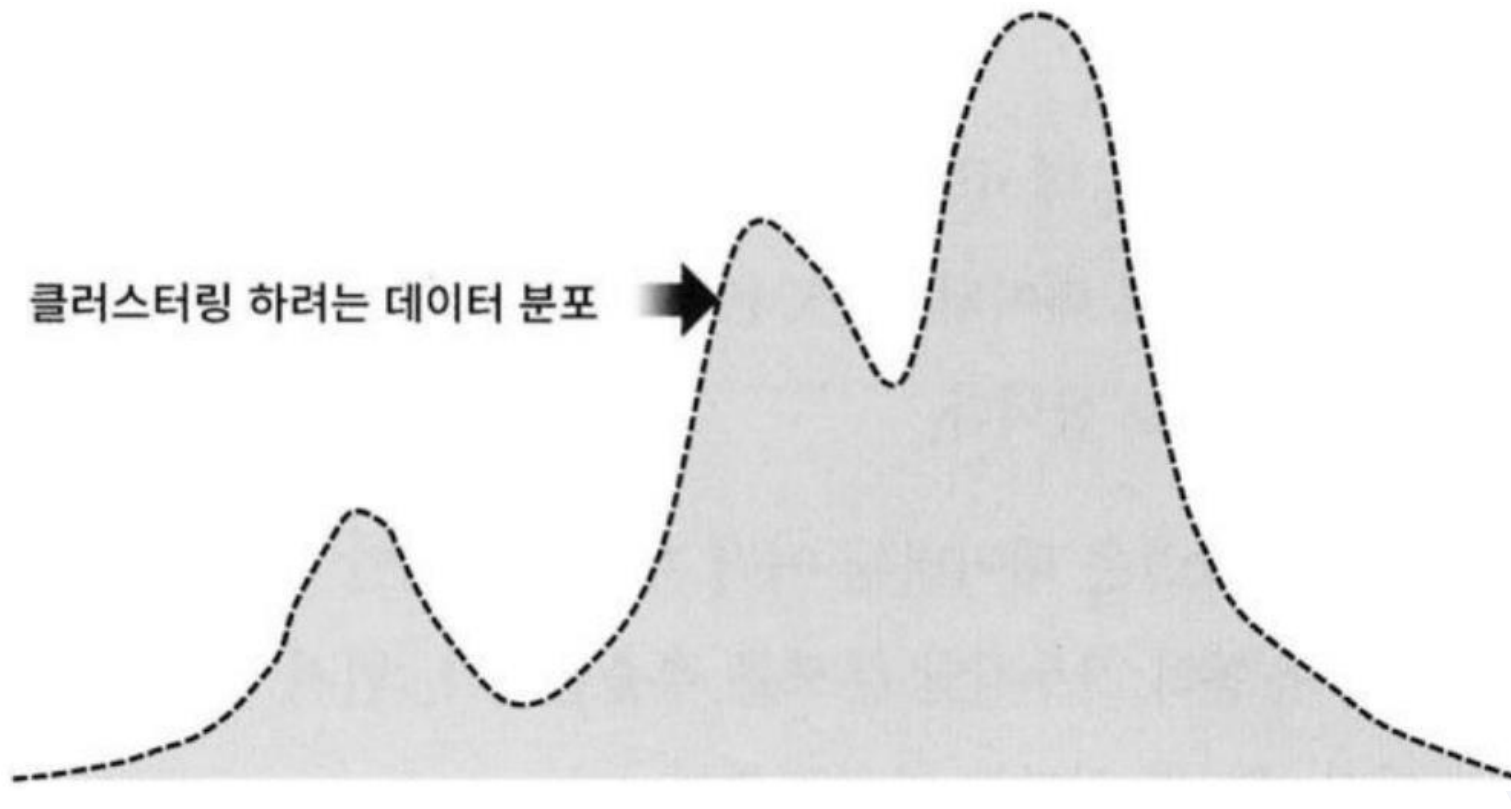
- 데이터를 여러 개의 가우시안 분포가 섞인 것으로 간주
- 섞인 데이터 분포에서 개별 유형의 가우시안 분포 추출
- 다음과 같이 3개의 가우시안 분포 A, B, C를 가진 데이터 셋이 있다고 가정



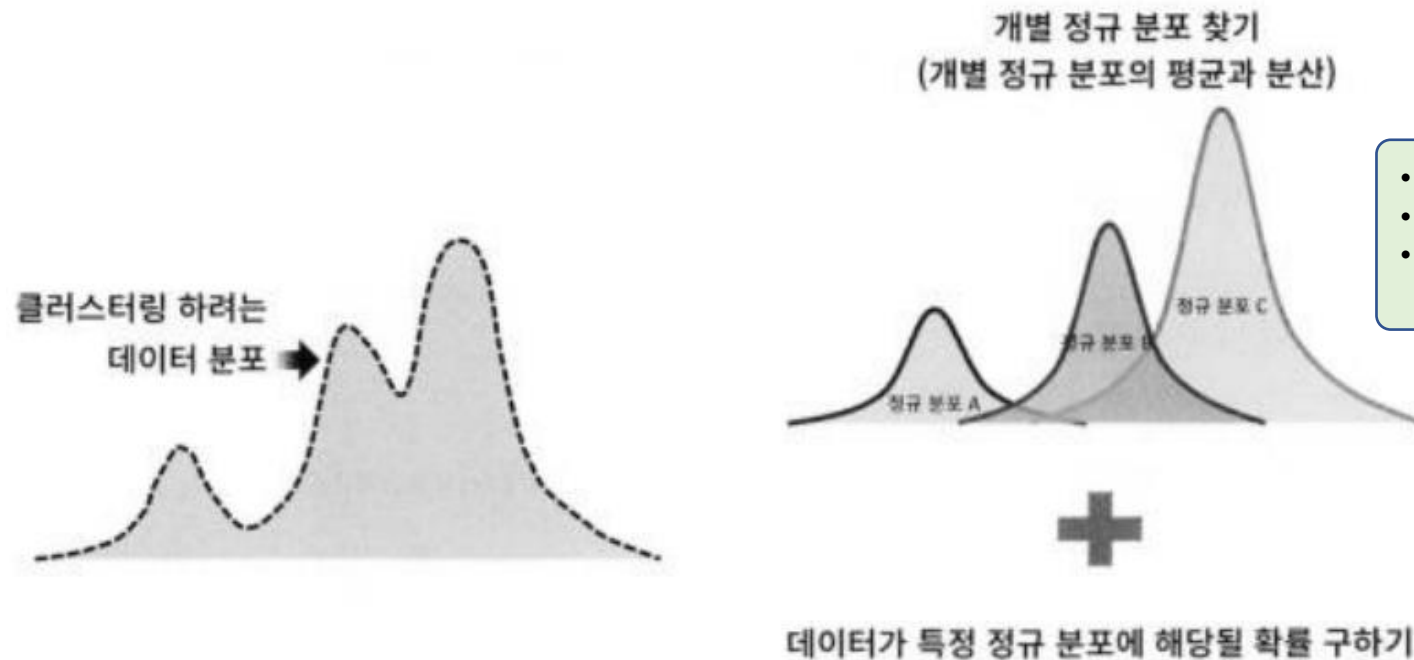
- 3개의 정규분포가 합쳐진 형태



- 클러스터링 하고자 하는 분포



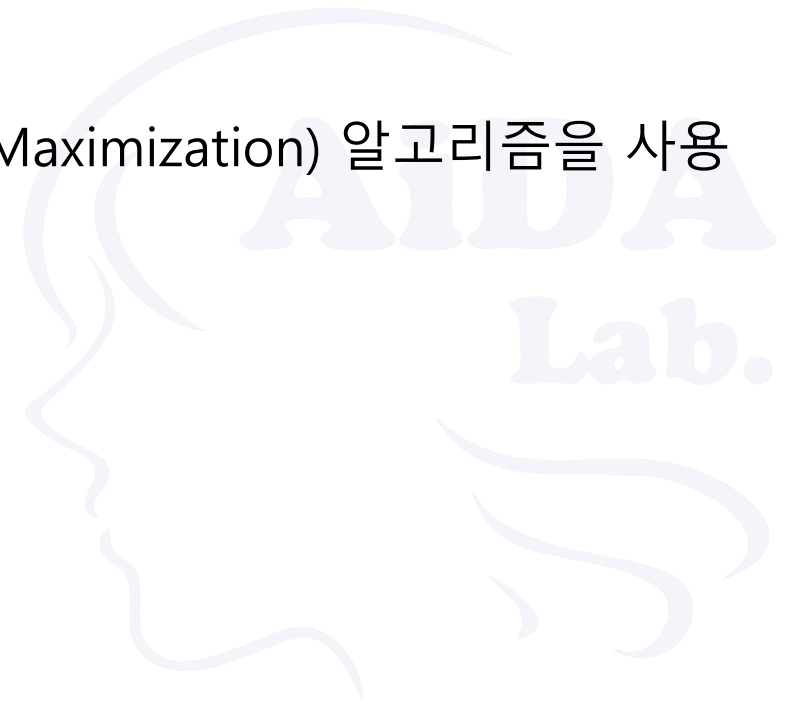
- 전체 데이터 셋은 서로 다른 정규 분포 형태를 가진 여러 가지 확률 분포 곡선으로 구성될 수 있다.
- 이러한 서로 다른 정규 분포에 기반에 군집화를 수행하는 것이 GMM 군집화 방식



- 만약 1000개의 데이터 셋이 있다면
- 이를 구성하는 여러 개의 정규 분포 곡선을 추출하고
- 개별 데이터가 이 중 어떤 정규분포에 속하는지 결정하는 방식

- 모수 추정

- GMM에서는 이러한 방식을 모수 추정이라고 하며 대표적으로 2가지를 추정함
 - 개별 정규 분포의 평균과 분산
 - 각 데이터가 어떤 정규 분포에 해당되는지의 확률
- GMM에서는 모수 추정을 위하여 EM(Expectation and Maximization) 알고리즘을 사용

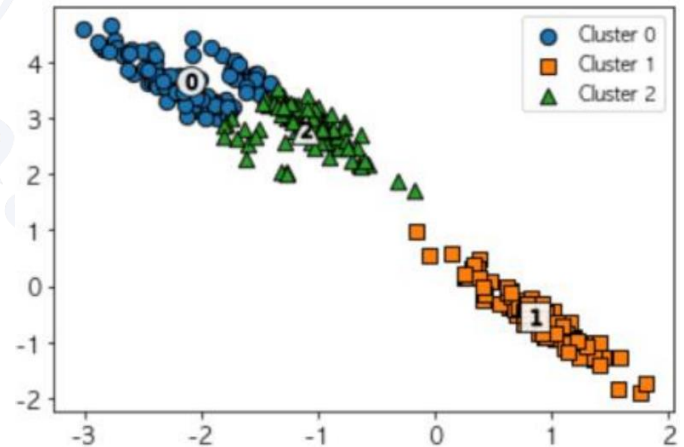
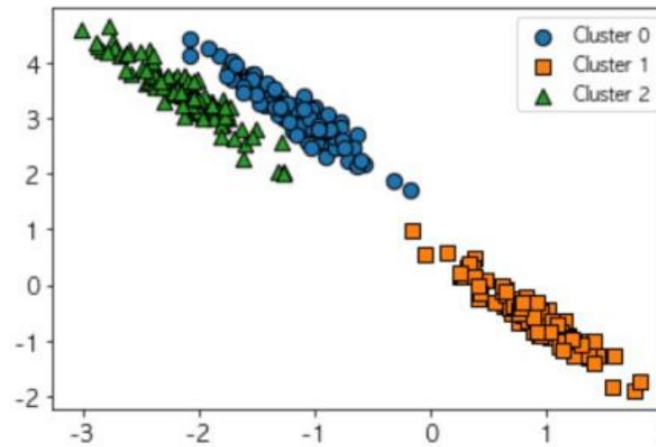
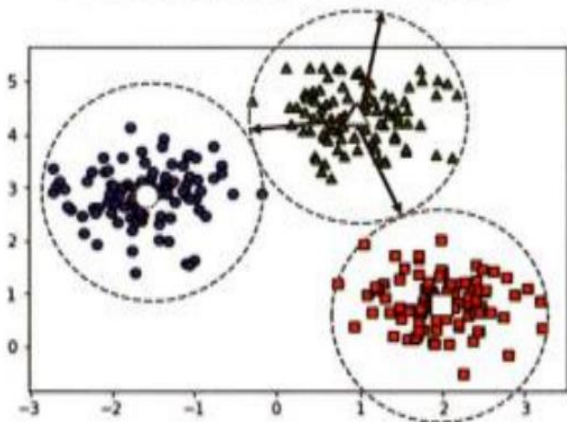


- k-평균과 GMM의 비교

- k-평균

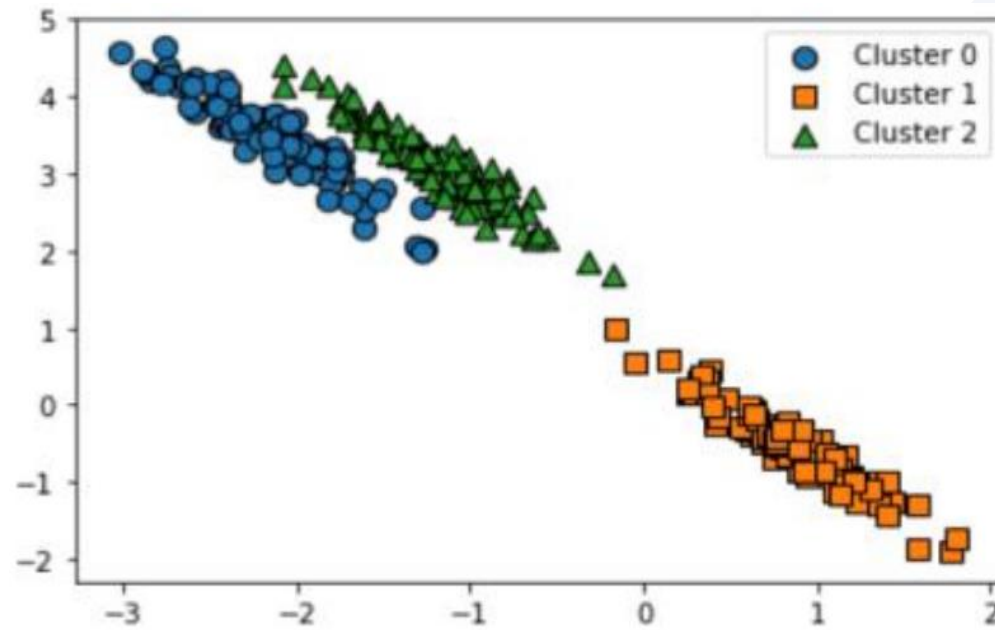
- 원형의 범위에서 군집화 수행 → 데이터 셋이 원형의 범위를 가질수록 k-평균의 군집화 효율 향상
- 데이터가 원형의 범위로 퍼져 있지 않은 경우, 특히 데이터가 길쭉한 타원형으로 늘어선 경우 → 군집화를 잘 수행하지 못함

Kmeans는 원형의 범위를 가지고 Clustering을 수행



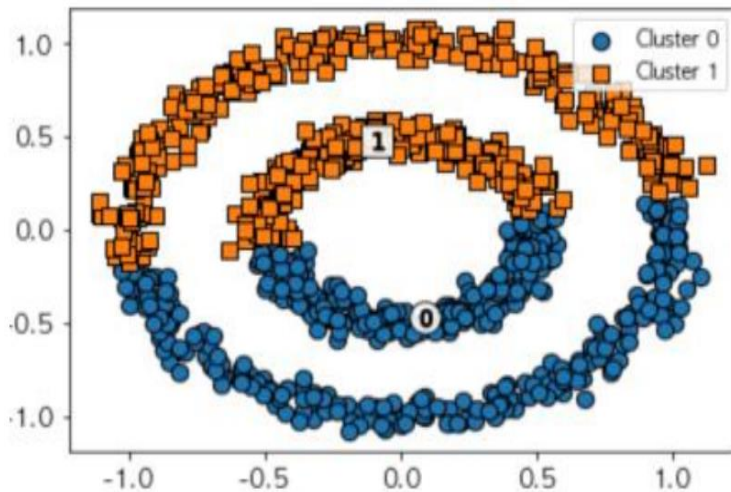
- GMM 군집화

- k-평균과 다르게 군집의 중심 좌표를 구할 수 없음
- k-평균보다 유연하게 다양한 데이터 셋에 잘 적용될 수 있음
- 군집화를 위한 수행 시간이 오래 걸림

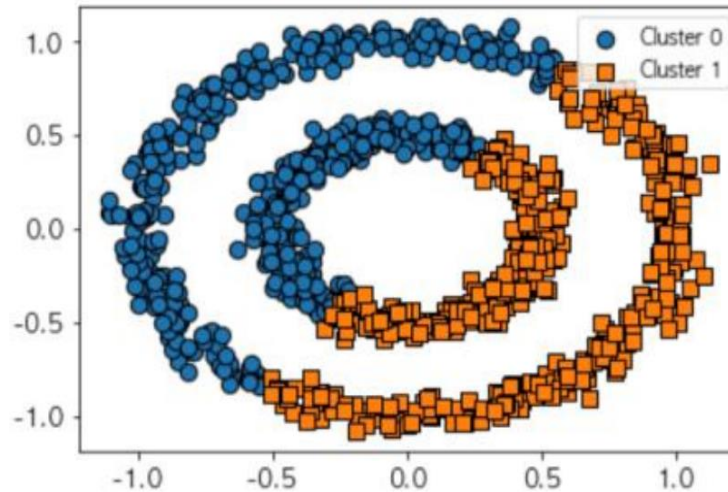


AiDA
Lab.

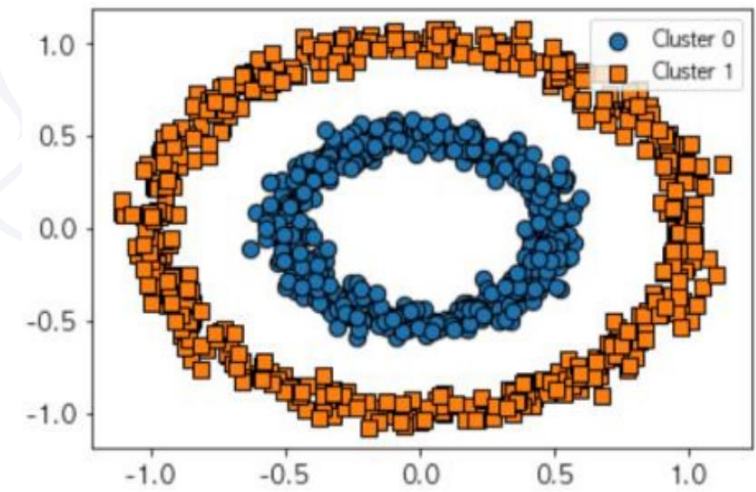
- DBSCAN (Density Based Spatial Clustering of Applications with Noise)
 - 밀도 기반 군집화의 대표적인 알고리즘
 - 간단하고 직관적임
 - 데이터의 분포가 기하학적으로 복잡한 데이터 셋에도 효과적인 군집화 가능



K-Means



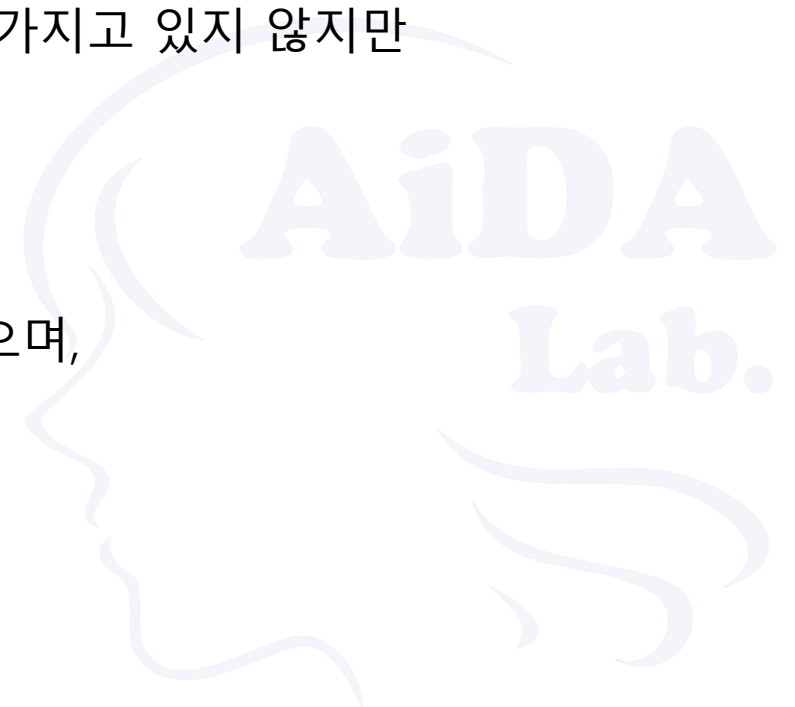
GMM



DBSCAN

- DBSCAN을 구성하는 가장 중요한 2가지 파라미터
 - 입실론 주변 영역(epsilon): 개별 데이터를 중심으로 입실론 반경을 가지는 원형의 영역
 - 최소 데이터 개수(min points): 개별 데이터의 입실론 주변 영역에 포함되는 타 데이터의 개수
- 입실론 주변 영역에 포함되는 최소 데이터 개수를 충족시키는지 여부에 따라 데이터 포인트 정의
 - 핵심 포인트 (Core Point)
 - 주변 영역 내에 최소 데이터 개수 이상의 타 데이터를 가지고 있을 경우, 해당 데이터를 핵심 포인트라고 함

- 이웃 포인트 (Neighbor Point)
 - 주변 영역 내에 위치한 타 데이터
- 경계 포인트 (Border Point)
 - 주변 영역 내에 최소 데이터 개수 이상의 이웃 포인트를 가지고 있지 않지만
 - 핵심 포인트를 이웃 포인트로 가지고 있는 데이터
- 잡음 포인트 (Noise Point)
 - 최소 데이터 개수 이상의 이웃 포인트를 가지고 있지 않으며,
 - 핵심 포인트도 이웃 포인트로 가지고 있지 않은 데이터



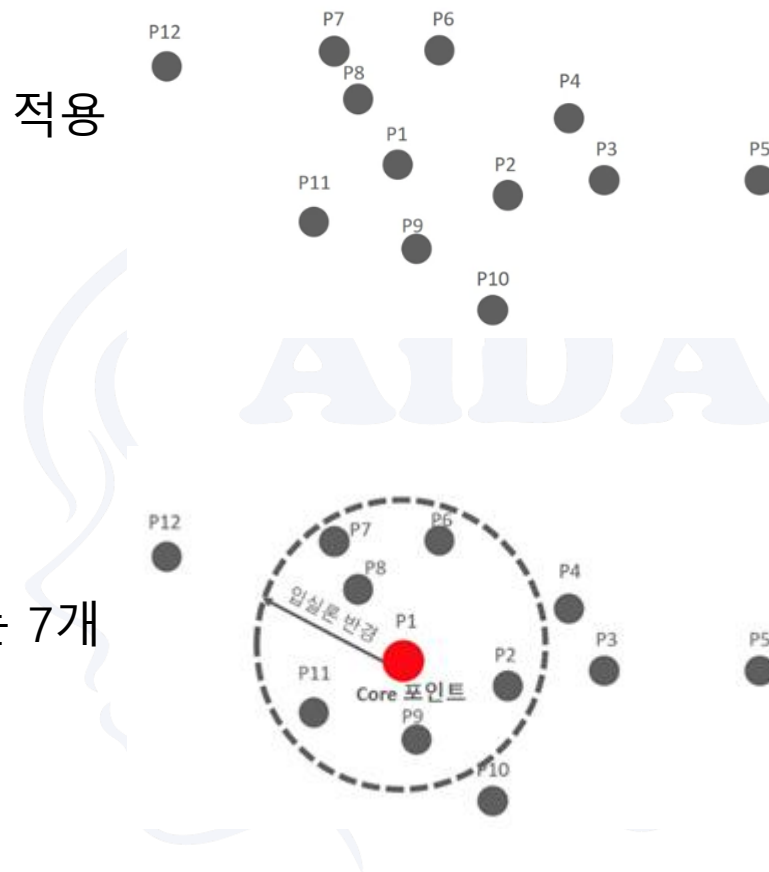
- DBSCAN 모델의 군집화 적용

- 1단계

- P1~P12까지 12개의 데이터세트에 대해 DBSCAN 군집화 적용
 - 특정 입실론 반경 내에 포함될 최소 데이터 세트는 6세트
 - 자기 자신의 데이터 포함

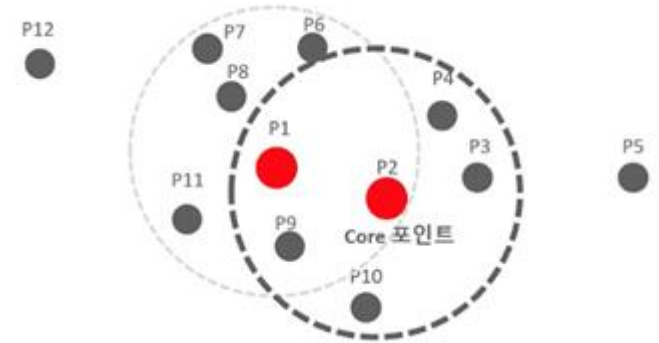
- 2단계

- P1 데이터를 기준으로 입실론 반경 내에 포함된 데이터는 7개
 - 자신은 P1, 이웃 데이터 P2, P6, P7, P8, P9, P11
 - 최소 데이터 5개를 만족하므로 P1데이터는 핵심포인트



- 3단계

- P2 데이터를 살펴보면 P2 역시 반경 내에 6개의 데이터 보유
- 자신은 P2, 이웃 데이터 P1, P3, P4, P9, P10(5개)
- P2도 최소 데이터 5개 이상을 만족하므로 핵심 포인트



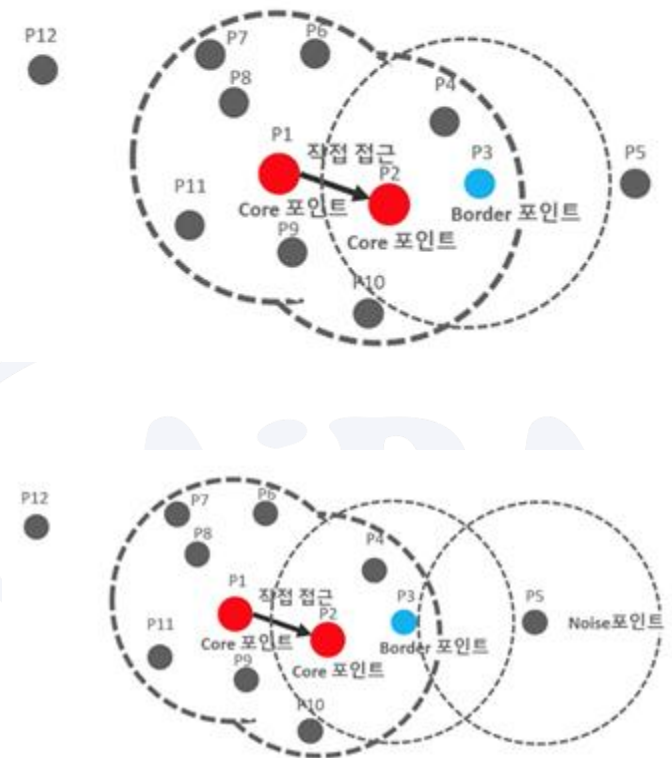
- 4단계

- 핵심 포인트 P1의 이웃 데이터 포인트 P2 역시 핵심 포인트일 경우, P1에서 P2로 연결하여 직접 접근 가능
- 특정 핵심 포인트에서 직접 접근이 가능한 다른 핵심 포인트를 서로 연결하면서 군집화 구성
- 이런 식으로 점차적으로 군집 영역을 확장해 나가는 것이 DBSCAN 군집화 방식



- 5단계

- P3 데이터의 경우 반경 내에 포함되는 이웃데이터는 P2, P4
이므로 군집으로 구분할 수 있는 핵심 포인트가 될 수 없음
- 그러나 이웃 데이터 중에 핵심 포인트인 P2가 있으므로 경계
포인트가 될 수 있음 → 경계 포인트는 군집의 외곽을 형성
- P5와 같이 반경 내에 최소 데이터를 가지고 있지도 않고, 핵
심 포인트 또한 이웃 데이터로 가지고 있지 않는 데이터를
잡음 포인트라고 함



- **DBSCAN: 입실론 주변 영역의 최소 데이터를 포함하는 밀도 기준을 충족시
키는 데이터인 핵심 포인트를 연결하면서 군집화를 구성하는 방식**

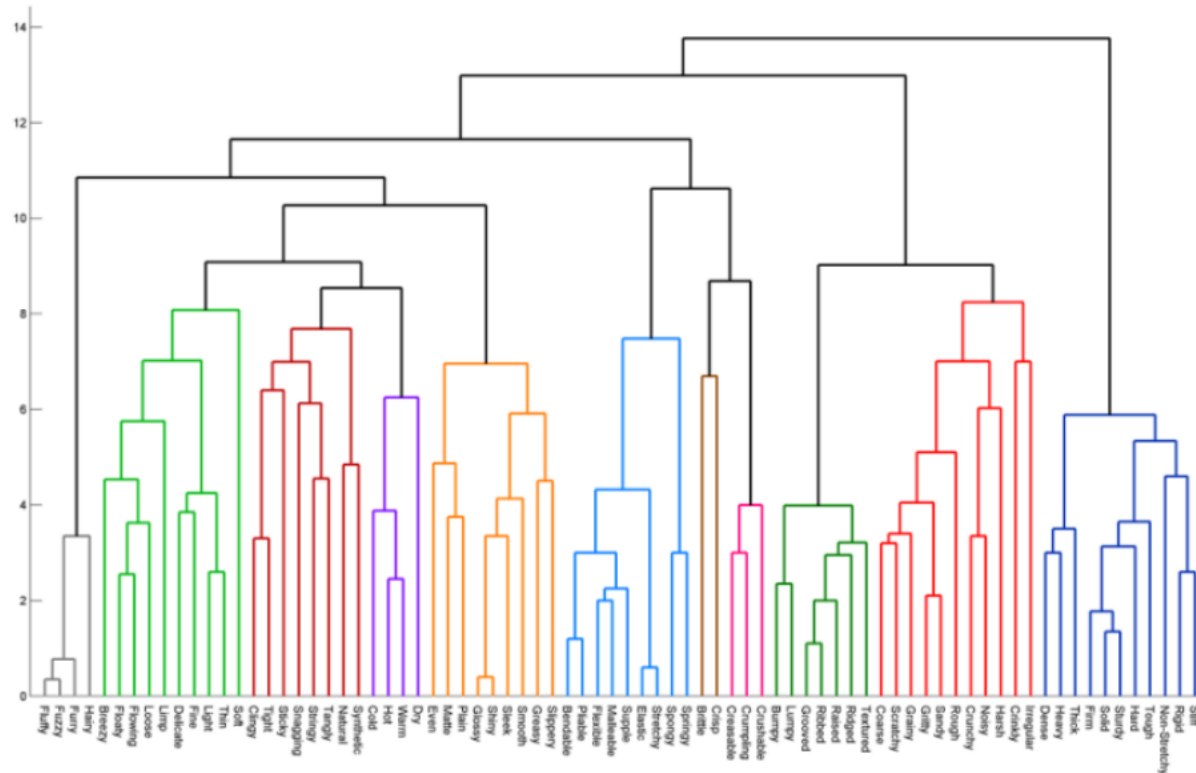
- 실습



- 계층적 군집화(Hierarchical Clustering, HC)

- 계층적 트리 모형을 이용해 개별 개체들을 순차적, 계층적으로 유사한 개체 및 그룹과 통합하여 군집화를 수행하는 알고리즘
- 개체들이 결합되는 순서를 나타내는 트리 형태의 구조인 덴드로그램(Dendrogram)을 사용
 - k-평균 군집화와 달리 군집 수를 사전에 결정하지 않아도 학습 수행 가능
- 덴드로그램을 생성한 후 적절한 수준에서 트리를 자르면 전체 데이터를 몇 개의 군집으로 나눌 수 있게 됨

- 덴드로그램(Dendrogram)

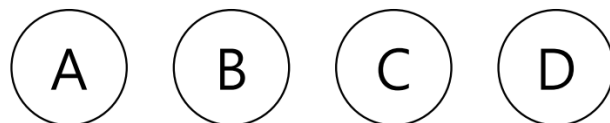


- 학습 과정

- HC를 수행하려면 모든 개체들 간 거리, 유사도가 사전에 계산되어 있어야 함

- 예시

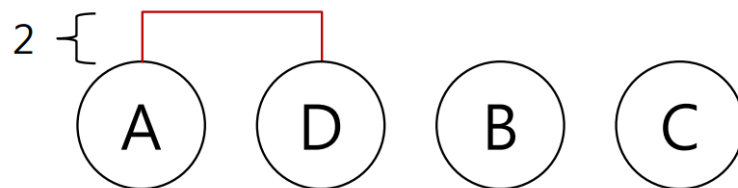
- 주어진 학습 데이터의 개체 수가 4 이고 거리 행렬이 다음과 같다면...



	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

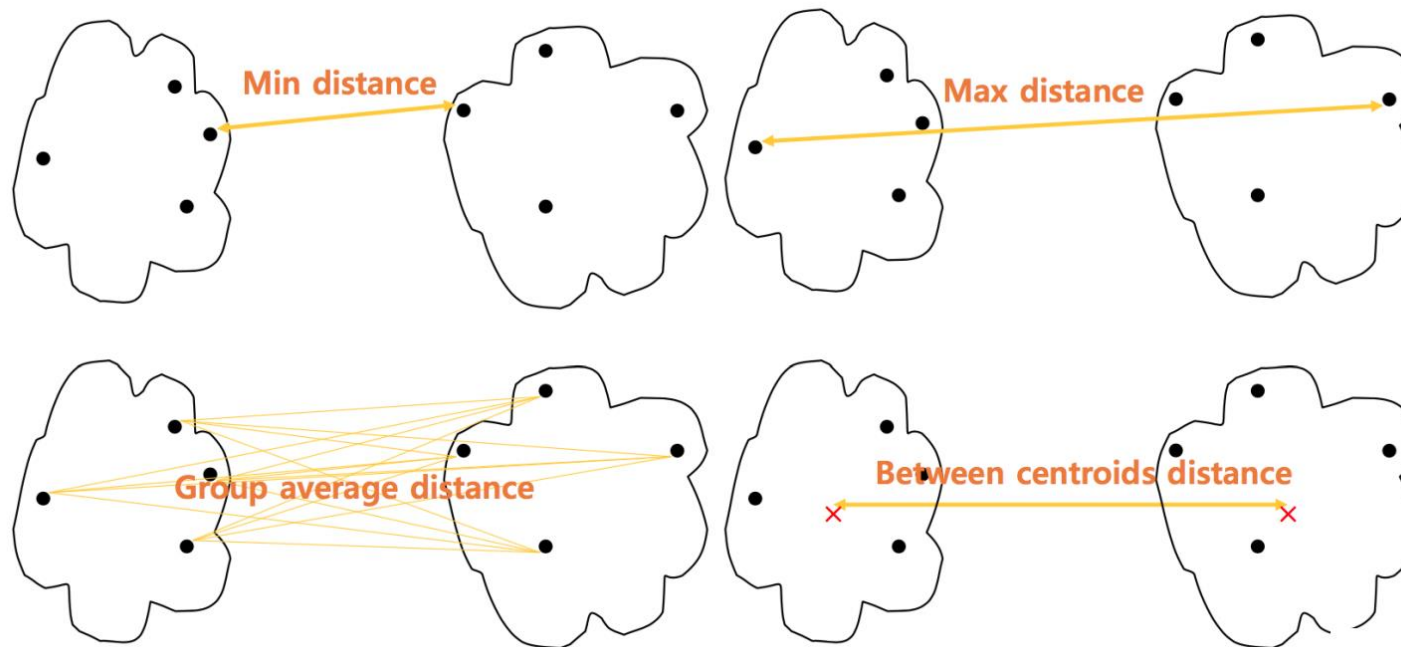
AiDA
Lab.

- 거리가 가까운 관측치들끼리 차례대로 군집으로 묶는다.
- 거리가 가장 짧은 것이 2이고 이에 해당하는 개체는 A와 D이므로 먼저 A와 D를 하나의 군집으로 엮는다.
- 왼쪽에 작성하는 덴드로그램의 높이는 관측치간 거리(2)가 되도록 한다.

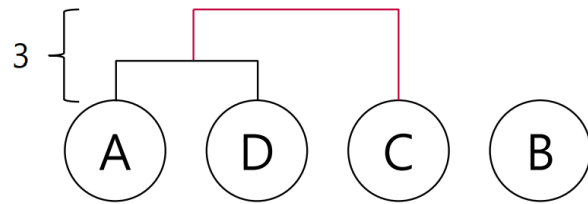


	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

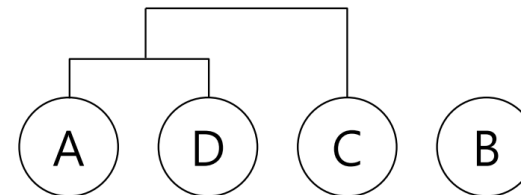
- A와 D를 한 군집으로 묶었으면 거리행렬을 바꿔 준다. 즉 개체-개체 거리를 군집-개체 거리로 계산한다.('AD'와 'B', 'AD'와 'C' 이렇게 거리를 구해야 한다)
- 군집-개체, 혹은 군집-군집 간 거리를 계산하는 방법은 여러가지가 있으며 적절한 것을 선택하도록 한다.



- 적절한 거리 계산 방법을 선택하여 거리를 계산했다면 거리 행렬을 업데이트한다.
- 업데이트 후 AD와 C가 가장 가까우므로 이 둘을 연결한다.

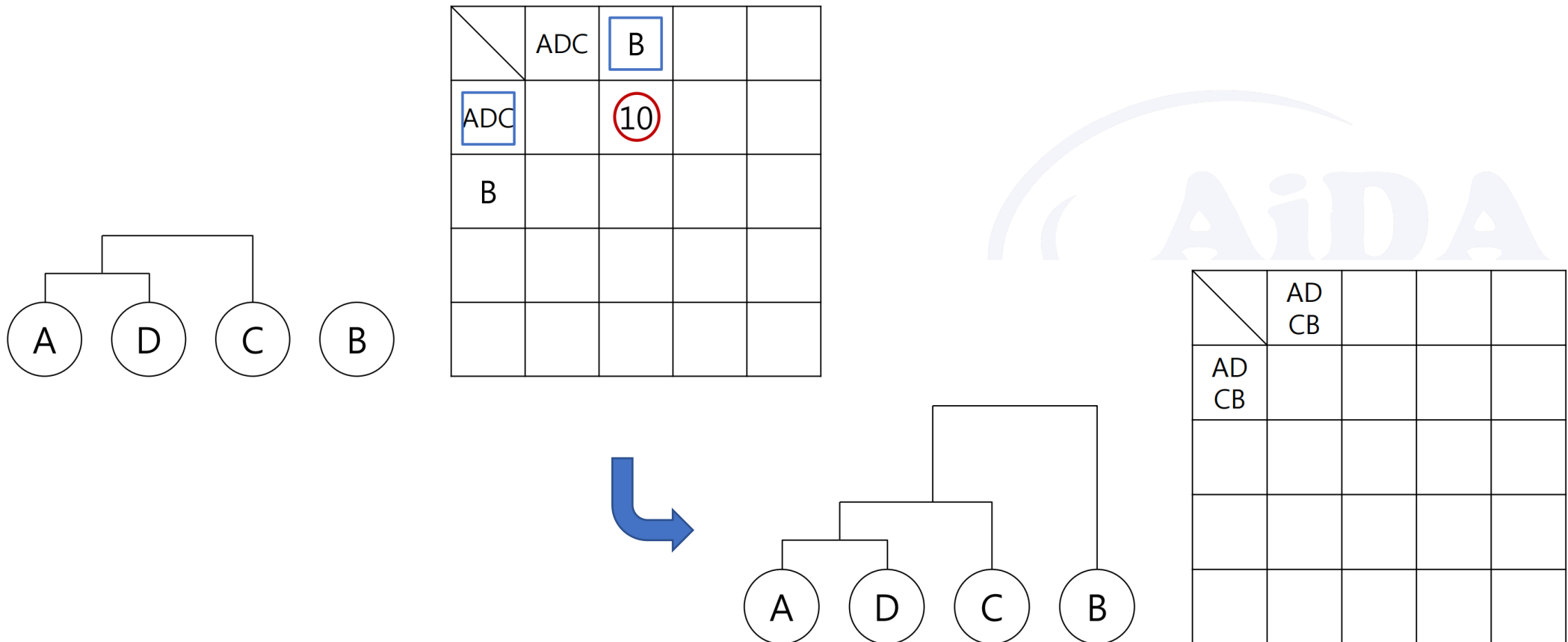


	AD	B	C	
AD		20	3	
B			10	
C				



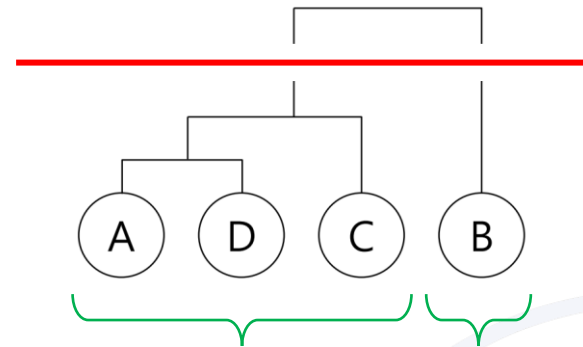
	ADC	B		
ADC		10		
B				

- 더 이상 분석 대상 관측치가 없으면 학습을 종료 한다.

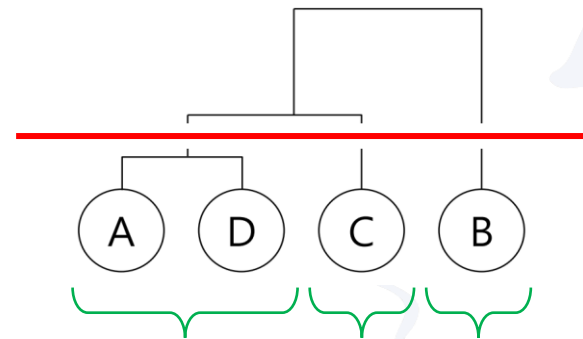


- 클러스터링 결과

- 덴드로그램의 최 상단을 잘라 주면
A, D, C와 B의 두 개의 군집으로 나뉘어짐



- 위에서 두번째 층을 자르면
A, D와 C, 그리고 B의 세 개의 군집으로 나뉘어짐



- 계층적 군집화는 간단하게 군집화를 수행할 수 있지만 k-평균보다 무겁다

HC의 계산 복잡성은 $O(n^3)$

- 토픽 모델링(Topic Modeling)

- 문서 집합에 숨어 있는 주제(토픽)를 찾아내기 위한 비지도 방식 머신러닝
- 주어진 여러 문서에서 주제(토픽)을 추출하여 각 문서에 어떤 주제가 있는지, 인사이트를 얻거나 각 문서를 주제별로 분류하기 위한 알고리즘
- 머신러닝 기반의 토픽 모델은 숨겨진 주제를 효과적으로 표현할 수 있는 중심 단어를 함축적으로 추출함
- 여기서 토픽이란 단어의 집합을 가리킴
- 한 단어가 다른 단어와 같이 출현하는 문서의 개수가 많을수록 그 두 단어는 같은 토픽 내에 그룹핑 될 가능성이 높아짐

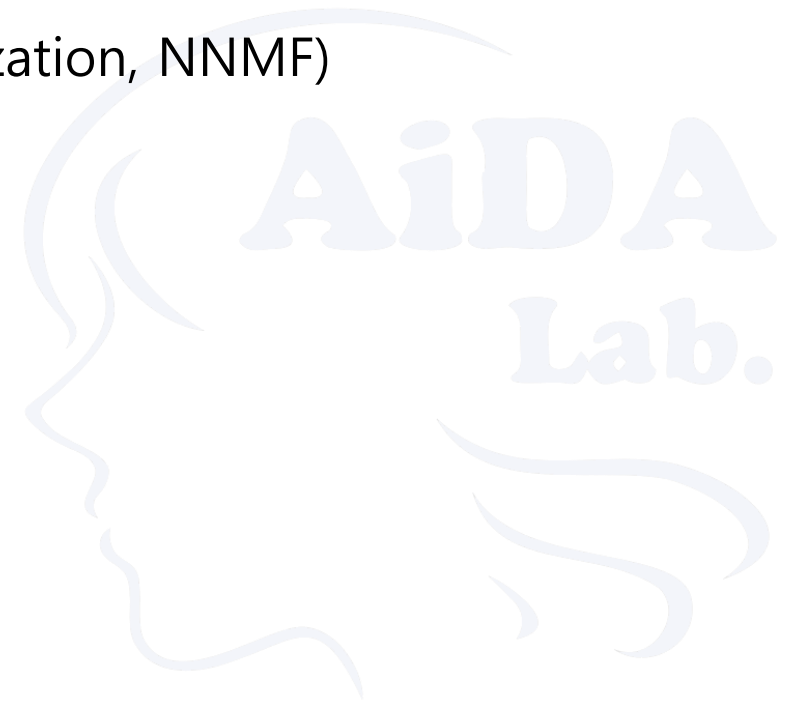
• 예시

토픽 1	토픽 2
개	고양이
산책	개다래

문서 1	문서 2	문서 3
토픽 1: 100% 토픽 2: 0%	토픽 1: 60% 토픽 2: 40%	토픽 1: 0% 토픽 2: 100%
개 개 산책	고양이 개다래 개 개 산책	고양이

- 문서 1은 토픽 1에서만 나오는 단어들로 구성되어 있다 → 문서 1은 100% 토픽 1로 구성
- 문서 2는 토픽 1에서만 나오는 단어인 개가 60%이고, 토픽 2에서만 나오는 단어인 고양이가 나머지 40% → 문서 2는 60% 토픽 1, 40% 토픽 2로 구성
- 문서 3은 고양이라는 토픽 2에 나오는 단어만으로 구성되어 있다 → 문서 3은 100% 토픽 2로 구성

- 머신러닝 기반의 토픽 모델링에서 자주 사용되는 기법
 - 잠재 디리클레 할당(Latent Dirichlet Allocation, LDA)
 - 잠재 의미 분석(Latent Semantic Analysis, LSA)
 - 음이 아닌 행렬 인수분해(Non-Negative Matrix Factorization, NNMF)

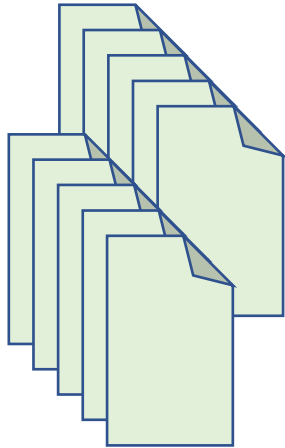


- **잠재 디리클레 할당(Latent Dirichlet Allocation, LDA)**

- 2003년 데이비드 블레이, 앤드류 응, 마이클 조던이 발표한 토픽 발견 기술
- 생성적 확률 모델군에 속하며, 토픽은 주어진 용어 집합이 발생할 확률로 표현됨 → 문서는 토픽들의 혼합체(Mixture)로 표현할 수 있음

- **LDA 모델의 특징**

- 주제가 구별될 필요가 없으며, 여러 주제에서 단어가 발생할 수 있음
→ 언어의 융통성을 다루는데 유용한 일종의 국소적인 애매함(Fuzziness)을 허용



한 말뭉치를 이루는 문서들이
여러 토픽을 구성한다

election stump run
president iraqi

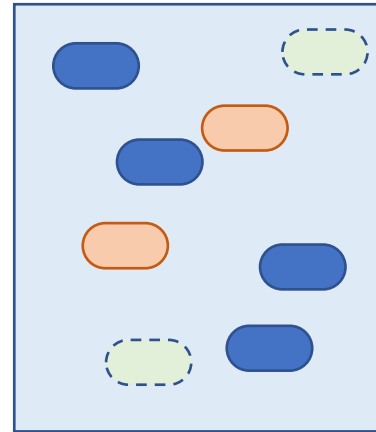
aioli fry pancake
kobe cream garlic

kobe nba record
draft finals tendon

note record pop
cello harmony trio

fair console game
unity tactical

1개 토픽은 여러 단어에
걸쳐 분포한다



단일 문서에
여러 토픽이 있다



문서 안에 있는 단어로
잠재 주제를 만들어서
문서 각각이
어떤 내용으로 구성되
었는지를 표현

• LDA의 알고리즘

1. 문서 각각의 단어에 무작위로 주제를 할당
2. 단어에 할당한 주제에서 문서 각각의 주제별 확률을 계산
3. 단어에 할당한 주제에서 주제별 단어의 확률을 계산
4. 2와 3을 곱해서 계산하는 확률로 문서 각각의 단어에 주제를 다시 할당
5. 과정 2~4를 완료 조건을 만족할 때까지 반복

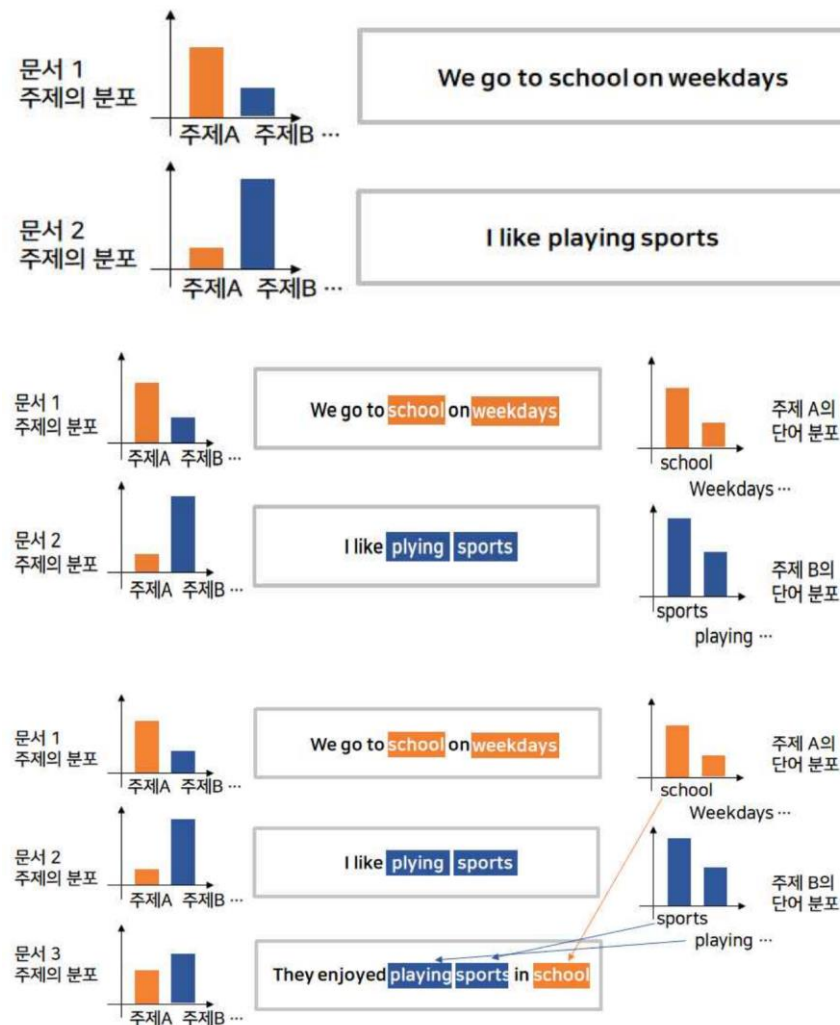
- 과정 4에서 계산한 확률에 따라 문서 각각의 단어에 주제를 할당하며
- 문서 각각의 주제 확률은 과정 2를 통해서 결정하므로
- 같은 문서 안에서 특정 주제를 선택하기가 쉽다
(즉 같은 문서의 단어는 같은 주제를 할당하는 경향이 있다.)

- 계산을 반복하면 문서에 특정 주제를 할당할 확률이 커짐
- 또한 특정 주제와 관련 있는 단어를 선택하기 쉬우므로 해당 주제를 대표하는 단어를 선택할 확률이 커짐

• 예제

- 5개의 영어 문장에 잠재 디리클레 할당 적용
- 주제의 개수: 2개

- We go to school on weekdays.
- I like playing sports.
- They enjoyed playing sports in school.
- Did she go there after school?
- He read the sports columns yesterday.



- 토픽 모델링의 활용 예

- 문서의 분류

- 토픽 모델링의 가장 기본적인 용도

- 검색

- 단순히 입력 키워드의 개수만을 가지고 검색 결과를 정하는 알고리즘(예, TF-IDF)은 입력 키워드가 포함되지 않은 문서는 검색 결과에서 제외하게 됨
 - 토픽 모델링은 입력된 키워드에 없는 단어만을 가진 문서라도 입력된 키워드와 연관성이 높다면 검색 결과의 상위에 포함시킬 수도 있음

- 주어진 문서들에 대한 인사이트
 - 주어진 문서의 집합에서 어떤 주제들이 나타나는지 관찰하여 새로운 인사이트를 얻을 수 있음
 - 비중이 높은 주제에 맞춘 콘텐츠의 작성 등
- 추천 시스템
 - 텍스트를 가지고 있는 아이템들의 경우, 토픽 모델링으로 토픽을 추출하고 각 아이템의 토픽 구성 벡터를 생성한 후, 사용자가 열람한 아이템들의 토픽 구성율을 구하고, 이에 기반하여 사용자가 열람하는 아이템들과 비슷한 구성의 토픽을 가진 아이템을 추천할 수 있음