클러스터링

유사도(Similarity)

- 항목간의 유사한 정도를 수치로 나타낸 것
- 분류나 예측에서 필요
  - 메일이 스팸에 가까운지 아니면 정상 메일에 가까운지
- 추천(음악, 책 등)
  - 두 아이템 또는 사람이 서로 얼마나 가까운지
- 머신러닝에서는 샘플들간의 유사도(similarity) 또는 거리(distance)를 측정하는 것은 필수적

유사도와 거리

- 유사도 결과에 따라 데이터 분석 결과가 달라짐
- 분석 경험과 도메인에 대한 이해 필요함
- 최적의 분석 결과가 나오도록 유사도를 변경해 가면서 반복 수행 필요함
- 유사도 s(similarity)
  - $0 \leq s \leq 1$ (1에 가까울수록 유사도 높음)

  ** cf) 코사인 유사도 : -1 ~ 1
- 유사도의 상대 개념으로 거리(distance) 사용
  - 유사도와 거리의 관계 : $d = 1 - s$

유사도 예

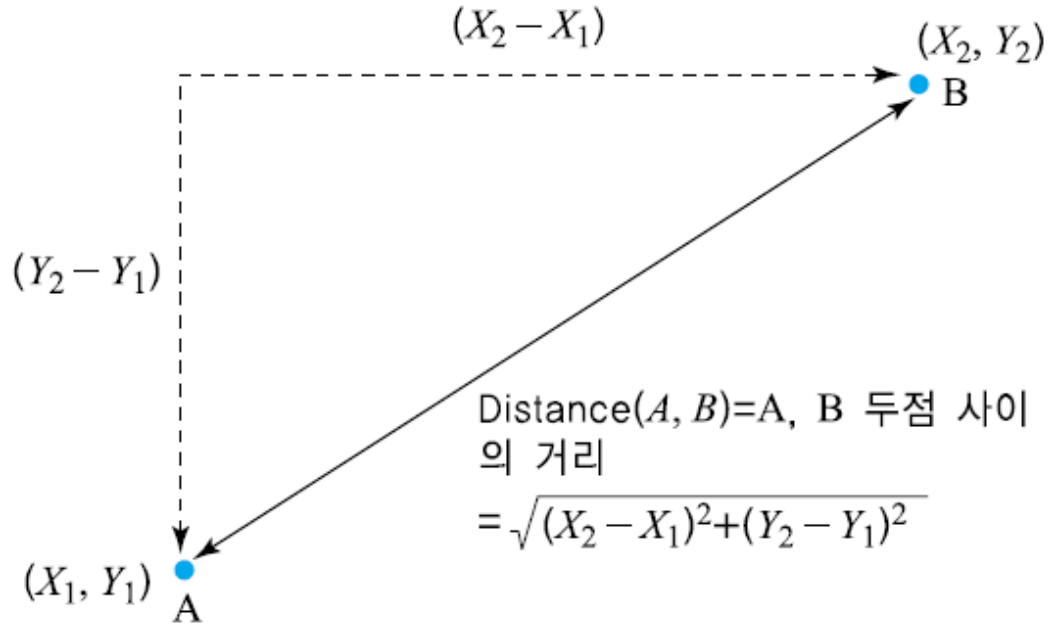
구분	키	몸무게	나이
A	174cm	70kg	21세
B	170cm	61kg	27세
C	162cm	73kg	29세

- **가까운 정도**
  - **키 : A, B**
  - **몸무게 : A, C**
  - **나이 : B, C**
- **샘플 특성의 성격과 단위가 달라서 주어진 그대로는 유사도 측정 곤란**
- **성격과 분포가 다른 데이터가 주어진 경우 거리 계산 방법**

  ➔ 표준 스케일링 방법

공간 거리

- **유클리디안(Euclidian) 거리**
  - 샘플이 다차원 공간상의 점(point)라 가정
  - 피타고라스 정리



- **n차원 :** $\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$

공간 거리

- 가정
  - 키, 몸무게, 나이의 분포 : 정규분포
  - 키, 몸무게, 나이의 표준편차 : 4cm, 3kg, 2세
- 특성 각각을 정규화(표준편차로 나누기)하여 거리를 계산

$$d_{(A,B)} = \sqrt{((174-170)/4)^2 + ((70-61)/3)^2 + ((21-27)/2)^2} = \sqrt{19}$$

$$d_{(A,C)} = \sqrt{((174-162)/4)^2 + ((70-73)/3)^2 + ((21-29)/2)^2} = \sqrt{26}$$

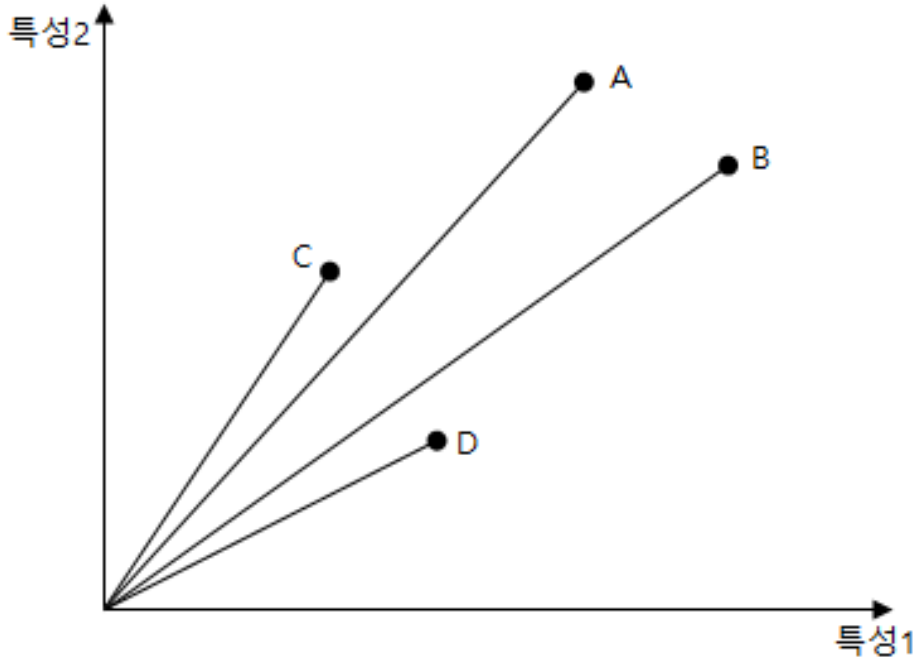
$$d_{(B,C)} = \sqrt{((170-162)/4)^2 + ((61-73)/3)^2 + ((27-29)/2)^2} = \sqrt{21}$$

- 결과
  - A, B가 가장 가깝고, A, C가 가장 멀다
- 일반적 항목간의 유사도
  - 관심을 어디에 두느냐에 따라 달라질 수 있다 (신체조건, 취향<영화, 음식 등> 등)

코사인 유사도

- 공간 상의 두 점이 만드는 각도 (방향성)
  - 각도가 적을수록 서로 가깝다
  - 즉, 가르키는 방향이 비슷하면 서로 유사하다고 본다

** Euclidian distance : 공간 상의 기하학적인 절대적 거리



- A, C가 가깝고,
- B, D가 가깝다

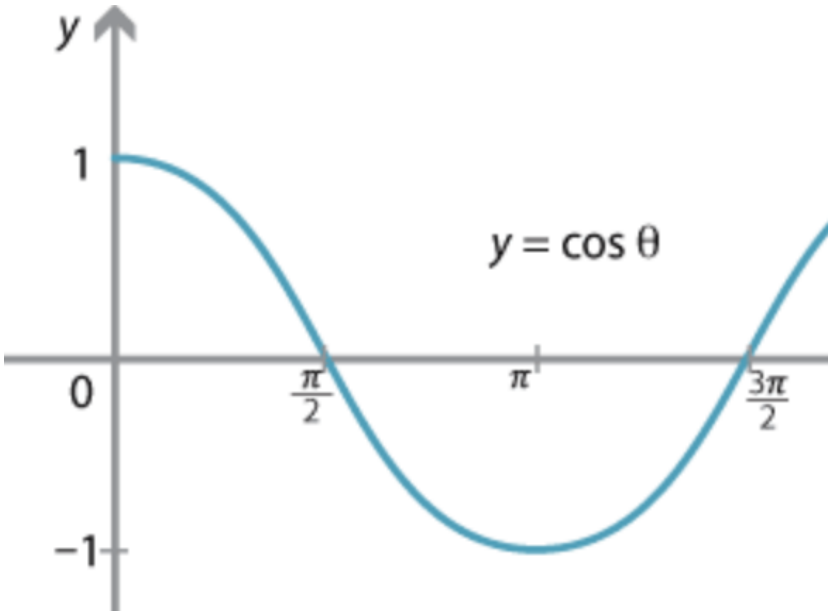
코사인 유사도

- 두 명의 아빠와 두 아들
  - 얼굴 크기 : 어른들이 서로 유사, 어린이들이 서로 유사
  - 얼굴 생김새 : 각 아빠와 아들이 서로 닮았다
- 크기 보다 모양(방향성)이 유사한 정도 평가 : 코사인 유사도
- 코사인 유사도 정의
  - 두 점 x, y 사이의 각도의 cos() 값
  - 계산 : 두 벡터의 내적(inner product)를 절대값의 곱으로 나누어 구한다

$$s_{\cos}(x,y) = \frac{X \bullet Y}{|X||Y|}$$

코사인 유사도

- 두 점의 cos() 값 : -1 ~ 1
  - 정확히 같은 방향 : cos(0) = 1
  - 서로 직각 방향 : cos(90) = 0
  - 서로 반대 방향 : cos(180) = -1



- 텍스트 분석을 할 때, 하나의 글에서
  - 같은 단어가 얼마나 같이 자주 등장하는 지
  - 반대 성향의 단어가 얼마나 자주 등장하는 지

  반영하고자 할 경우에도 사용

자카드(Jaccard) 유사도

- 비슷한 취향의 사람을 찾을 때 사용 – 영화, 도서, 음악 추천 등
- 영화 보는 취향에 따른 유사도 측정
- 지난 1년 동안 국내에 개봉된 영화가 500편
  - A와 B가 본 영화 중 겹치는 영화가 5편, 5/500 = 0.01
  - A와 C가 본 영화 중 겹치는 영화가 10편, 10/500 = 0.02
  - 즉, 0.01 < 0.020이므로 A와 C가 더 가깝다고 할 수 있음
  - 위와 같은 계산 방법이 적절한가?

자카드(Jaccard) 유사도

- 어떤 두 항목이 겹치는 부분의 절대량만을 보지 않고, 두 항목의 공통 부분이 얼마나 많은지를 고려하여 이에 대한 상대적인 값을 유사도로 사용해야 함

$$S_{Jaccard}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

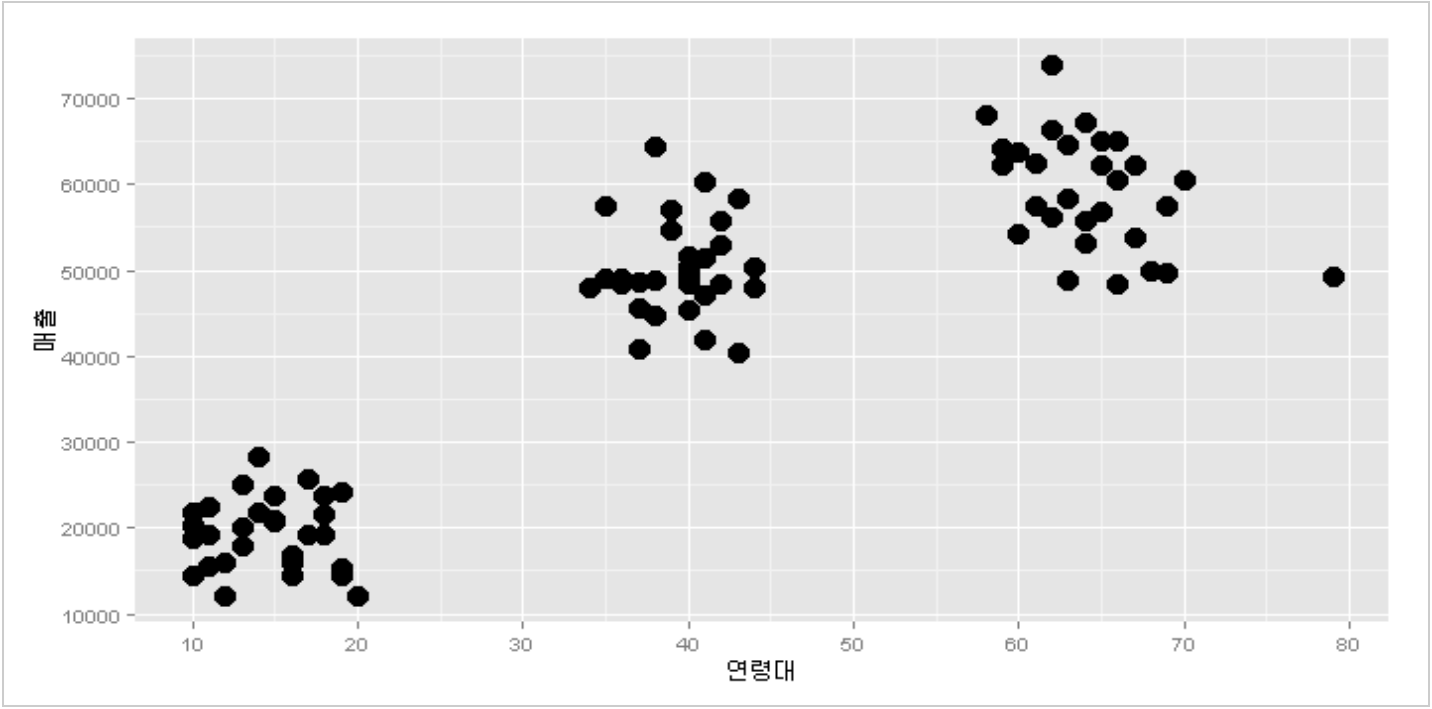
  - A, B, C가 각각 지난해 본 영화의 총 개수가 20편, 50편, 200편
  - J(A,B) = 5 / (20+50-5) = 0.076
  - J(A,C) = 10 / (20+200-10) = 0.047
  - 즉, 0.076 > 0.047이므로 A와 B가 더 가깝다고 할 수 있음

클러스터링 개요

- 성격이 비슷한 항목들을 그룹으로 묶는 작업
- 군집화(Clustering) : 대표적인 비지도 학습
- 분류, 회귀 등 정답을 예측하는 지도학습과 달리, 비지도 학습은 정답이 없이 데이터로부터 중요한 의미를 찾아내는 머신러닝
- 특성의 수를 줄이는 주성분분석(PCA)도 비지도 학습

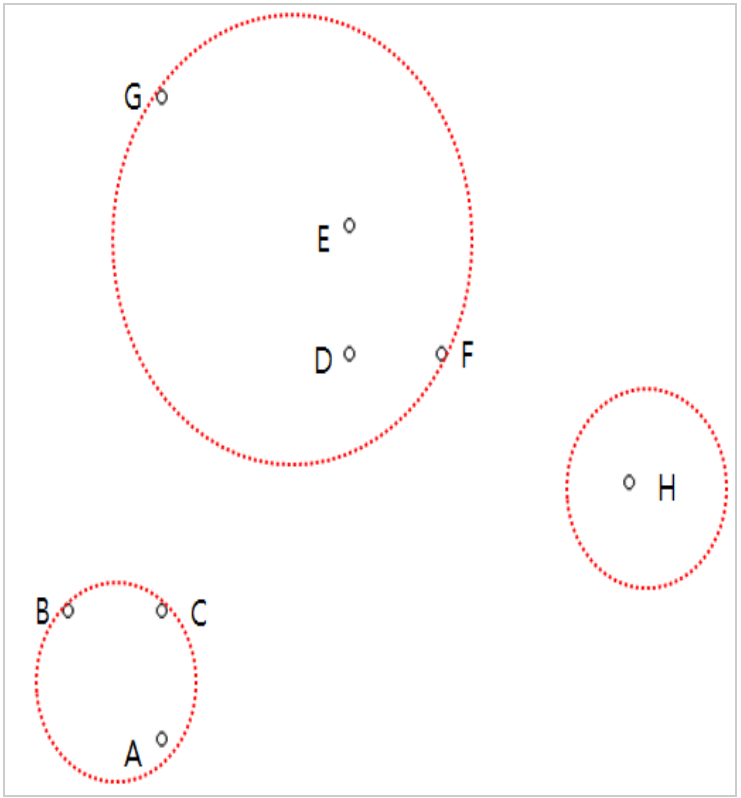
클러스터링 예

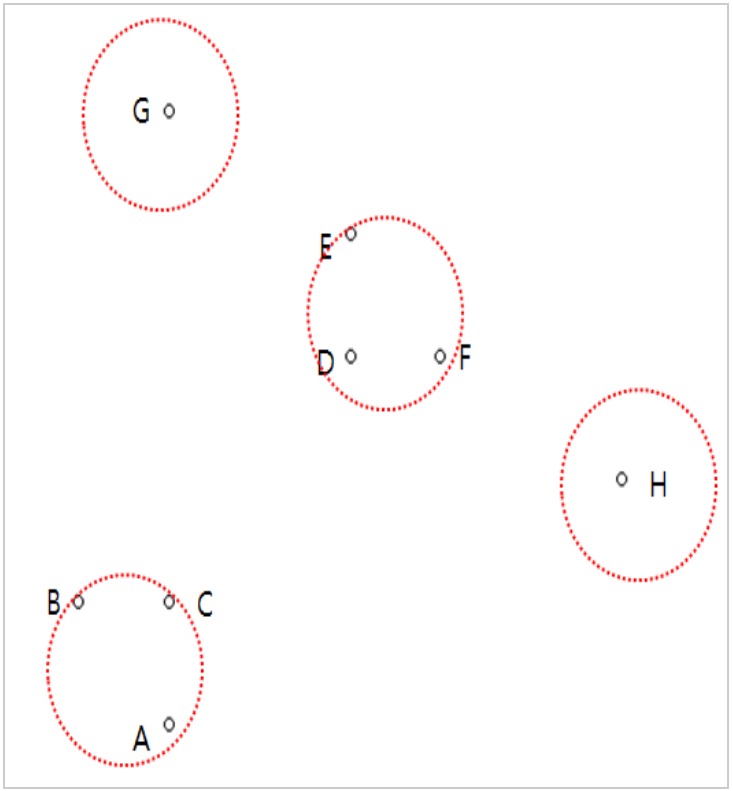
- 어느 문구점에서 한달간 구매한 사람을 **매출액 기준**으로 정리
- 고객의 타입에 따른 마케팅 차원에서
  - **몇 개의 그룹**으로 나누는 것이 **타당**할까?



클러스터의 수, k

- **적정한 군집의 수(k)를 먼저 찾아야 함**

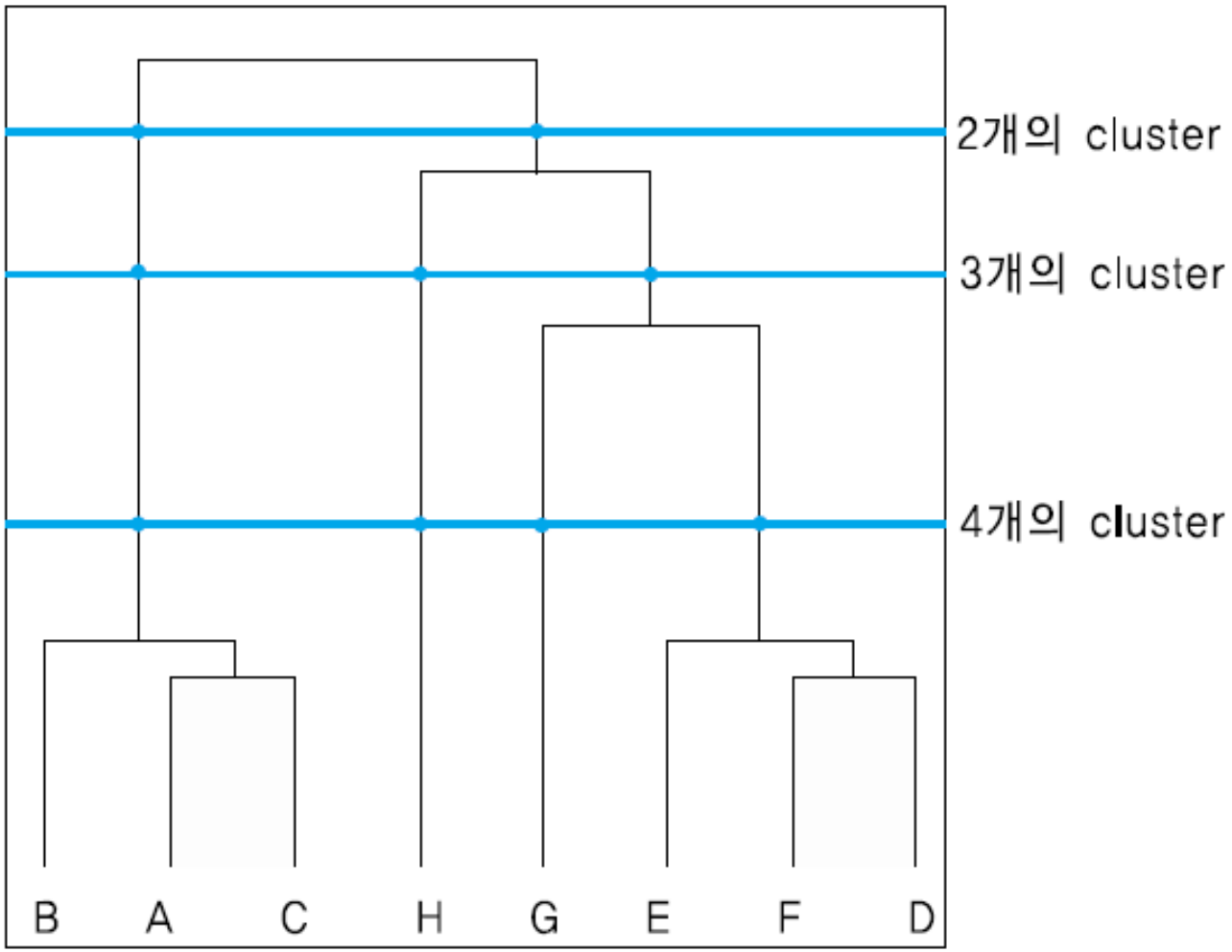




클러스터링 조건

- 조건
  - 같은 그룹 내의 항목들은 서로 속성이 비슷함 (유사도가 큼)
  - 다른 그룹에 속한 항목들과는 속성이 서로 다름 (유사도가 작음)
- 비정상 패턴(이상치) 식별에도 사용

덴드로그램(Dendrogram)



적절한 클러스터의 수(k) 선택

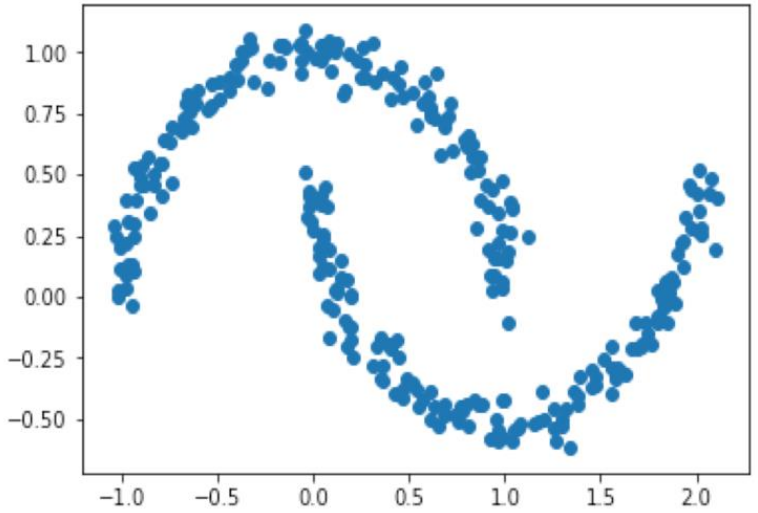
- 군집화에서 가장 어렵고 중요한 것
  - 예) 고객 집단 : 단골, 보통집단, 불만집단
    - k=3
- k값 선택 기준 : 각 클러스터 내의 항목들의 동질성
  - k = N : 동질성 최고 (N, 전체 샘플 수)
    - 너무 크면 항목들을 세밀하게 구분할 수 있으나 군집화 의미가 없어진다
  - k = 1 : 동질성 최악
    - 너무 작으면 여러 성격을 가진 항목들이 너무 섞여 있는 현상 발생

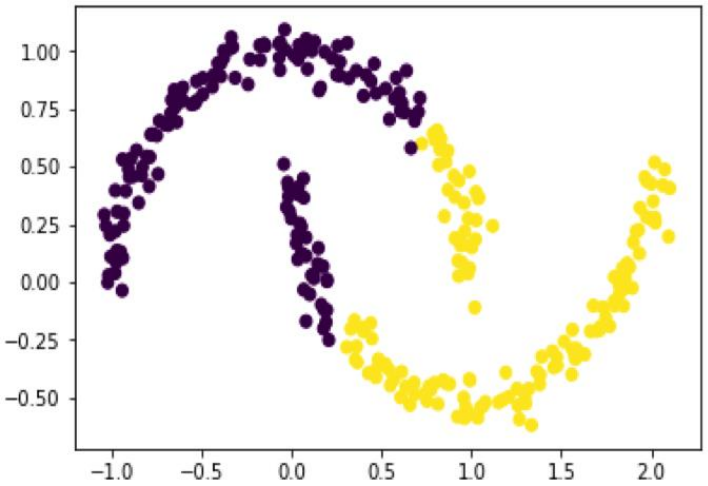
➔ 동질성이 충분히 만족되는 적절히 큰 k값 선택해야

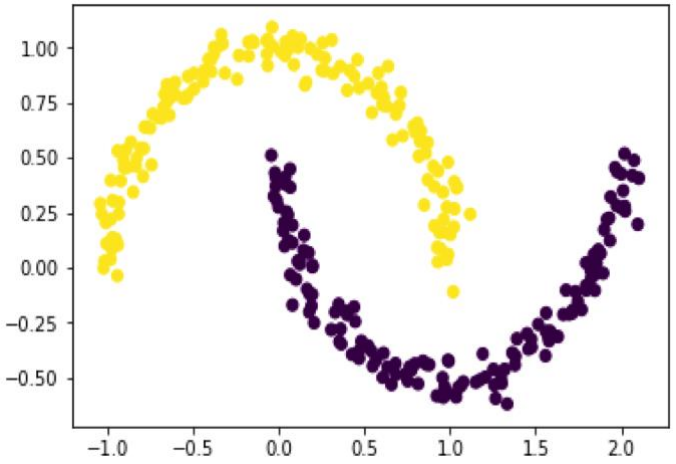
클러스터링 알고리즘 – k-means

- 특성 변수 공간 상의 임의의 k 개의 초기 지점을 클러스터 중점(cluster center)으로 정한다
- 각 클러스터 중점을 중심으로 거리가 가까운 항목을 선택하여 클러스터 공간을 나눔
- 각 클러스터에 포함된 항목들의 평균 위치를 구해 이를 새로운 클러스터 중점(centroid)으로 변경
- 새로 설정된 중점을 중심으로 경계를 다시 그림
  - 각 항목들이 소속된 클러스터가 바뀔 수 있음
- 변경된 항목들을 가지고 클러스터 중심을 다시 계산
- 더 이상 클러스터의 모양이 바뀌지 않을 때까지 반복 수행함
  - KMeans() 사용

Two Moons 데이터





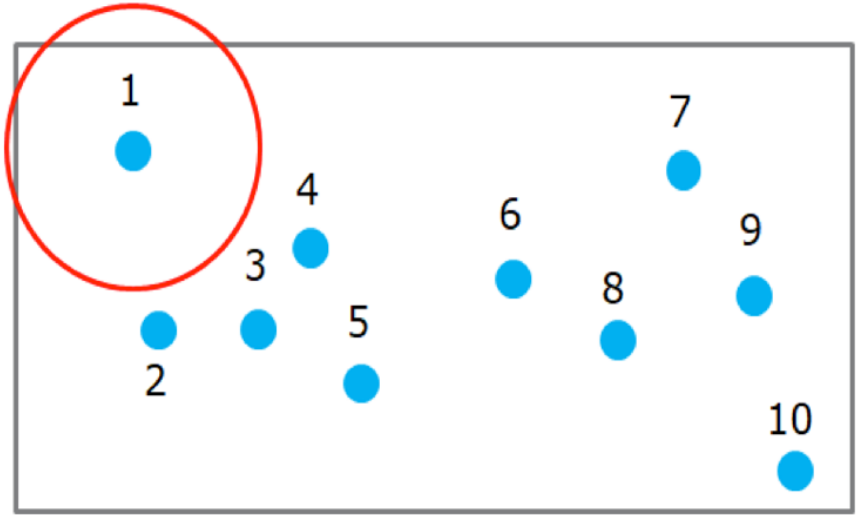


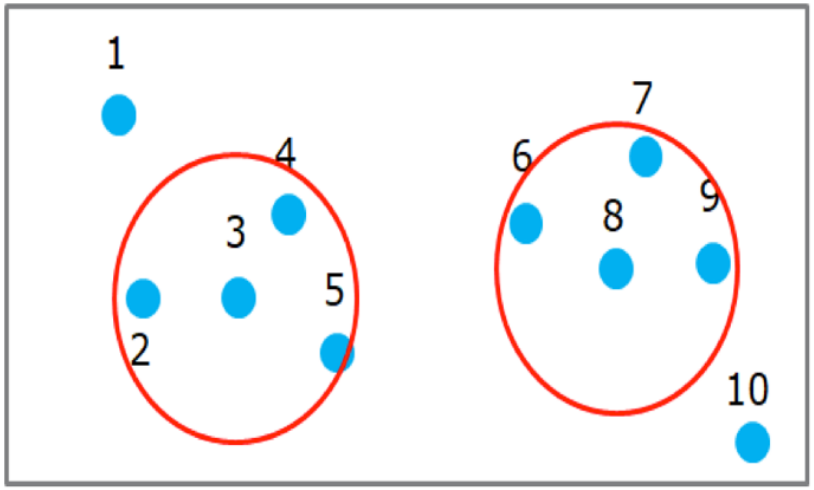
클러스터링 알고리즘 – DBSCAN

- 밀도 기반 클러스터링 알고리즘
- k-means처럼 단순히 거리만을 기준으로 군집화를 하는 것이 아니라 "가까이 있는 샘플들은 같은 군집에 속한다" 는 원칙으로 군집을 차례로 넓혀가는 방식이다.
- 샘플들의 몰려 있는 정도 즉, 밀도가 높은 부분을 중심으로 인접한 샘플들을 포함시켜 나간다.
- 한 점을 기준점으로 반경 r내에 점이 n개 이상 있으면 하나의 군집으로 인식하는 방식

클러스터링 알고리즘 – DBSCAN

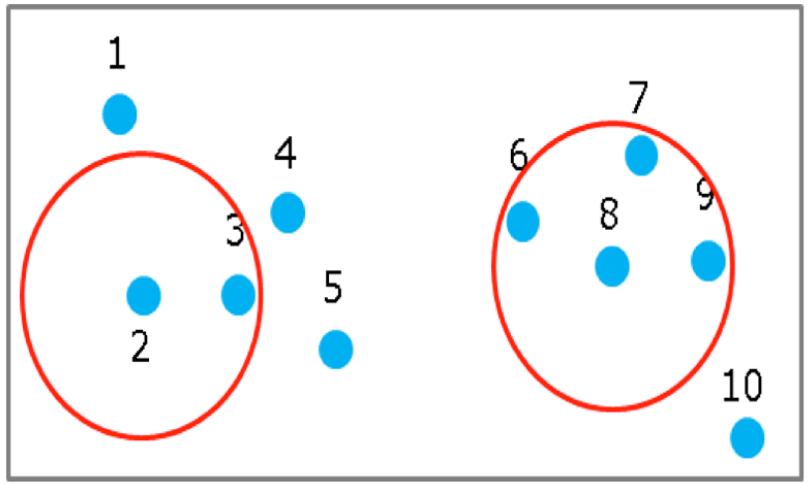
- 1번 데이터를 중심으로 보면 반지름 r인 원 안에 군집이 되기 위한 최소 기준인 (예를 들어 n=4라면) 샘플이 없다.
- 이 데이터는 **노이즈 데이터**(noise point)가 되며 클러스터에서 제외
- 3번과 8번 데이터를 중심으로 보면 원 안에 4개의 점이 있으며 이러한 데이터를 **코어 데이터**(core point)라고 한다.
  - 코어 데이터들은 스스로 클러스터를 형성할 수 있다.

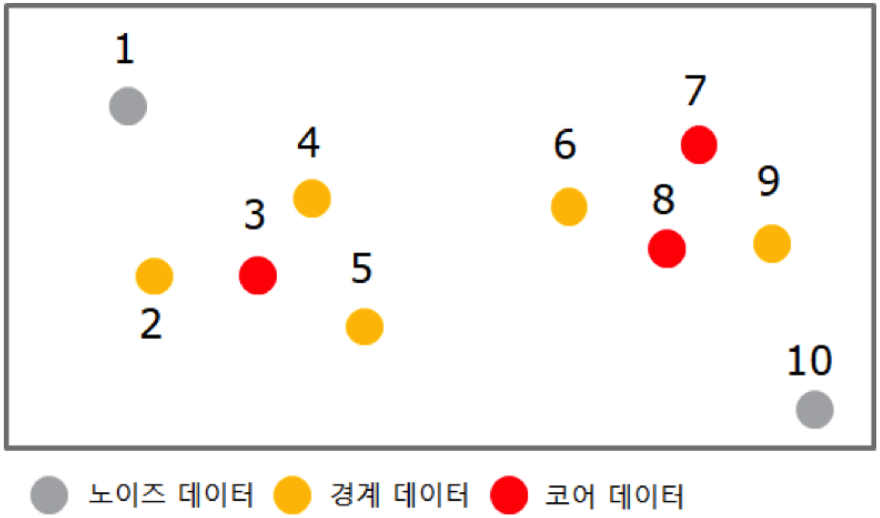




클러스터링 알고리즘 – DBSCAN

- 2번 데이터는 최소 기준인 4개의 데이터를 포함하지는 못하지만 코어 데이터인 3번을 포함한다. 이런 데이터를 경계 데이터(border point)라고 하며 인접한 군집에 포함시킨다.
- 정해진 반지름 r인 원을 이용해 코어 데이터, 경계 데이터, 노이즈 데이터들을 분류하면 아래와 같다.
- 두개의 클러스터와 한개의 노이즈를 구분했다.





Clustering (군집)

❖ **What is Clustering (군집)?**
- 각 객체의 유사성을 측정하여 유사성이 높은 집단으로 나눔
- 그룹에 대한 사전 정보 없음.
- 그룹의 개수나 특성에 대한 사전 정보가 주어지면 -> Classification (분류) 사용
- 군집의 개수나 구조에 대한 가정 없이 각 데이터 간의 거리를 기준으로 나눔

• **Similarity or Proximity (유사도)**
- 항목 간의 유사한 정도를 수치로 표현
- Euclid Distance (유클리드 거리), Manhattan Distance(맨하탄 거리), etc.
- 범주형 – Jaccard Distance (자카드 유사도)

❖ **What is Hierarchical/Agglomerative Clustering (계층적/응집형군집)?**
- 객체간의 유사도를 계산해 가장 가까운 것들부터 차례로 군집화
- Dendrogram 을 사용해 군집 형성 과정 파악
- 방법: Single, Complete, Average, Ward(군집간 정보 손실 최소화)