인공지능을 위한
머신러닝 알고리즘

8. 비지도 학습

Tacademy

CONTENTS

학습 목표

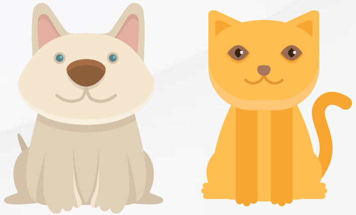
- 클러스터링과 비지도 학습의 관계를 이해할 수 있다.
- K-means 알고리즘의 클러스터링 과정을 이해할 수 있다.
- 데이터 포인트들 사이 거리 측정 알고리즘과 사용법을 이해할 수 있다.

1. 클러스터링

클러스터링이란?

cluster

- 클러스터링은 데이터에서 '클러스터(Clusters)'라는 '비슷한 그룹'을 찾는 기법을 뜻함
- 클러스터링은 서로 생김새가 비슷한 데이터끼리 하나의 클러스터로 묶고, 생김새가 매우 다른 데이터끼리 다른 클러스터로 분류 (類類相從, 가재는 게 편 등..)



클러스터링 = **비지도 학습 (Unsupervised Learning)**

- 데이터의 그룹을 묶을 수 있는 어떠한 사전 정보도 주어지지 않기 때문
- 사전 정보 (예> 레이블)이 주어지면 지도 학습임
- 이러한 이유 때문에, 클러스터링과 비지도 학습은 동의어로 여겨지기도 함

1. 클러스터링

일상 속 클러스터링의 예

예제 1 : **"small", "medium", "large"** 티셔츠를 만들기 위해서 사람들을 비슷한 크기로 그룹을 지음

- 각 사람 크기에 맞는 옷을 만들려면 너무 많은 비용이 듦
- 한 사이즈로 통일하기에는 맞지 않는 경우가 많음

예제 2 : 마케팅을 하기 위해서 고객들을 비슷한 정도에 따라 여러 분류로 나눔

- 고객의 유형에 따라 마케팅 전략을 세움

Tacademy

일상 속 클러스터링의 예

예제 3 : 문서가 많을 때, 내용의 비슷한 정도에 따라서 하나의 파일로 묶음

- 주제에 따라서 계층적 구조를 띄기도 함

- 클러스터링은 일상생활 속에서 가장 많이 사용되는 데이터 마이닝 기법 중 하나

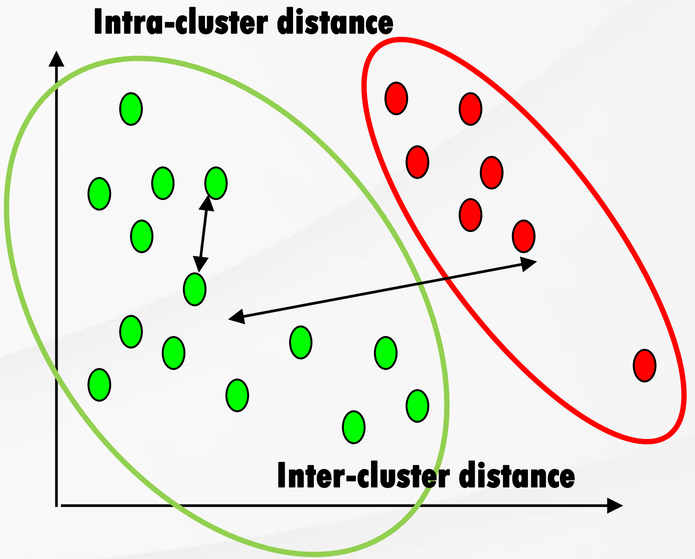
- 인터넷을 통한 온라인 문서들의 급속한 증가로 인해서 문서 분류가 중요한 이슈가 됨

클러스터링 이슈

- 어떻게 그룹을 나눌 것인가?
- 데이터들의 비슷한 정도(distance, similarity)를 어떻게 측정할 것인가?
- 클러스터링이 잘 되었는지 어떻게 평가할 것인가?

Inter-clusters distance → 최대화

Intra-clusters distance → 최소화



"클러스터링 결과의 질은 알고리즘, 거리 측정 방법 등에 따라 좌우됨"

2. K-means 클러스터링

K-means 클러스터링이란?

- 데이터 포인터들의 집합 **D**를 다음과 같이 정의하자
  - $\{x_1, x_2, \ldots, x_n\}$, 여기서 $x_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$는 실수값을 갖는 벡터
  - $X \subseteq R^r$, r은 데이터 속성의 개수

- **K-means** 알고리즘은 주어진 데이터를 **k**개의 클러스터로 분류
  - 각 클러스터는 **Centroid**라고 불리는 중심점 **(Center)**를 가짐
  - K의 값은 프로그래머가 정할 수 있는 가변 값임

K-means 클러스터링이란?

- K가 주어졌을 때, 알고리즘은 다음의 과정을 거침

1. 데이터 포인트들 중에 무작위로 K개를 선택하여 **Centroid**로 정함
2. 각 데이터 포인트들을 K개의 **Centroid**로 할당함
3. 각 클러스터의 구성원들을 기반으로 **Centroid**를 다시 계산
4. 수렴 조건이 만족되지 않으면 **2**번으로 감

수렴 조건

01 다른 클러스터들로 재배치되는 데이터 포인터들이 존재하지 않음

02 Centroids가 변경되지 않음

03 Sum of Squared Error (SSE)가 최저 임계치에 도달한 경우

Tacademy

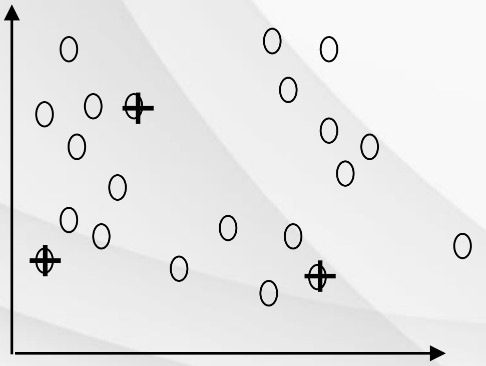
수렴 조건

$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2$$

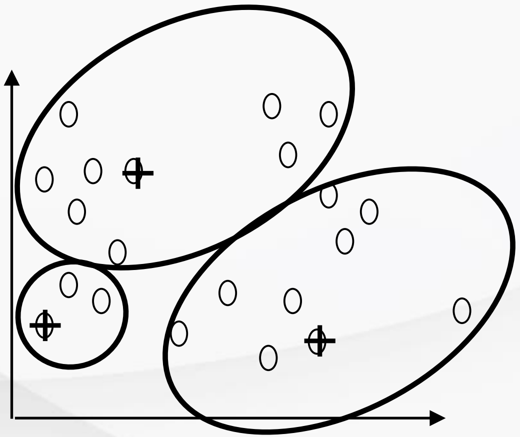
- C_j는 j번째 클러스터를 뜻함
- m_j는 클러스터 C_j의 centroid (클러스터 C_j에 있는 모든 데이터들의 평균 벡터)
- dist (x, m_j)는 데이터 포인트 x와 centroid m_j사이의 거리

Tacademy

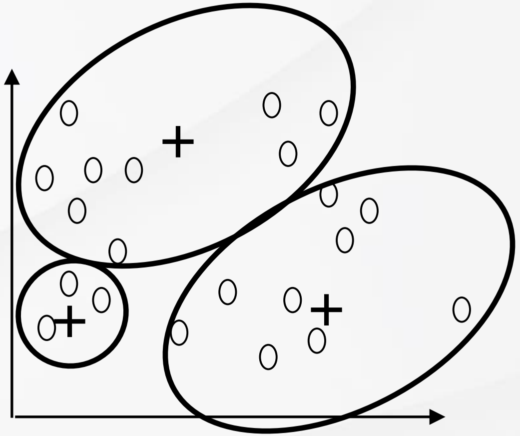
K-means의 예



1. K개의 중심을 무작위 선택



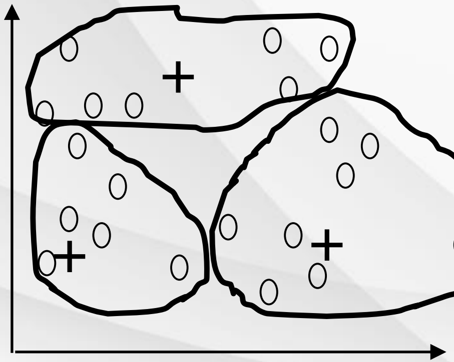
2. 클러스터 배정

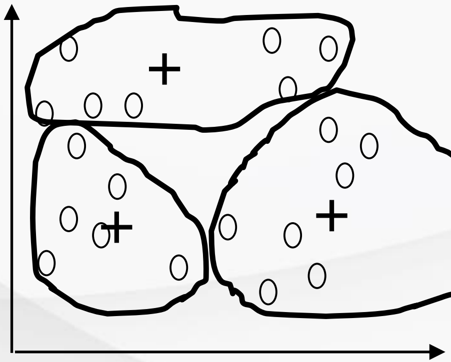


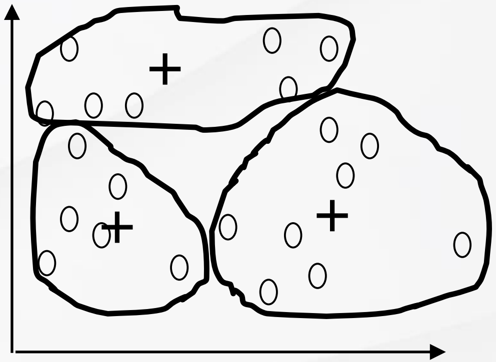
3. Centroid를 다시 계산

Tacademy

K-means의 예







4. 클러스터 재배정

5. Centroids 다시 계산

6. 클러스터 재배정

7. 클러스터링 종료

거리 측정 함수(Distance Function)의 예

- K-means 알고리즘은 데이터 집합에서 평균을 정의하고 계산할 수 있으면 사용할 수 있음
- 유클리디안 공간 (Euclidean Space)에서 클러스터의 평균은 다음과 같이 계산

$$\mathbf{m}_j = \frac{1}{|C_j|}\sum_{\mathbf{x}_i \in C_j} x_i$$

$|C_j|$는 클러스터 C_j에 존재하는 데이터 포인트 개수

- 하나의 데이터 포인터 x_i로부터 *centroid* $\mathbf{M}_j$ 까지의 거리는 다음과 같이 계산

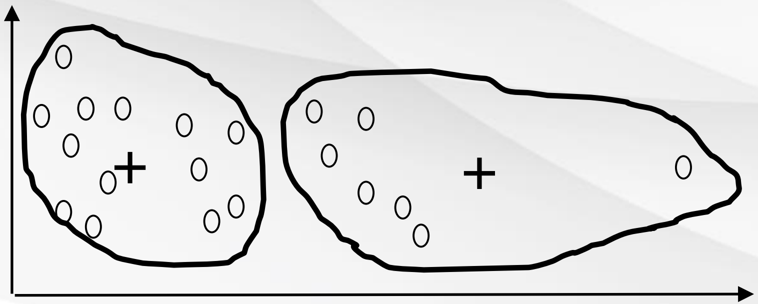
$$dist(x_i, m_j) = |x_i, mj|$$
$$= \sqrt{(x_{i1} - m_{ji})^2 + (x_{i2} - m_{j2})^2 + \ldots + (x_{ir} - m_{jr})^2}$$

K-means의 장점

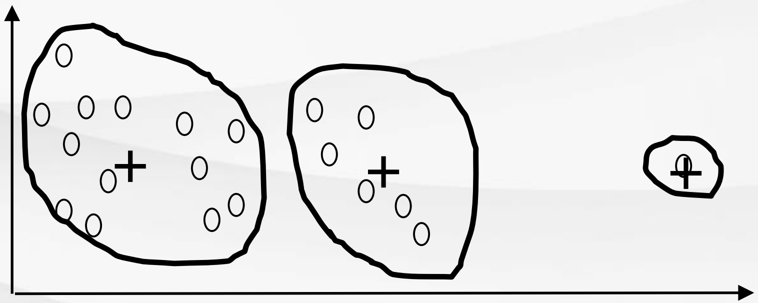
- 이해하고 구현하기 쉬움
- 효율적인 시간 복잡도를 가짐: O(tkn)
  - n은 데이터 포인터들의 개수
  - k는 클러스터의 개수
  - t는 클러스터 재배정 반복 횟수
- k와 t의 값이 작으므로 k-means 알고리즘은 데이터 개수에 따라 선형 복잡도를 갖는 알고리즘으로 볼 수 있음
- K-means는 가장 널리 사용되는 클러스터링 알고리즘
- Sum of Squared Error를 사용할 경우 지역적 최적화에서 종료될 수 있음, 전역적 최적점은 찾기가 어려움

K-means의 단점

- 데이터의 평균 값이 정의될 수 있는 데이터에만 사용 가능
- K의 값은 프로그래머의 몫 → 가장 최적의 K의 값을 찾기 어려움
- 아웃라이어에 매우 민감함
    - 아웃라이어 데이터란 다른 데이터 포인트들과 매우 동떨어져 있는 데이터를 뜻함
    - 아웃라이어 데이터는 데이터 기록 과정 중 벌어지는 오류 또는 독특한 성격을 갖는 이종 데이터로 인해 발생할 수 있음



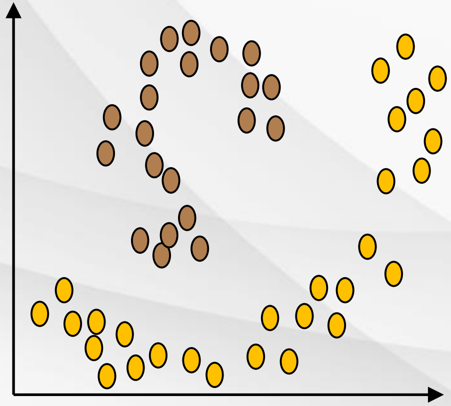
올바르지 못한 클러스터링



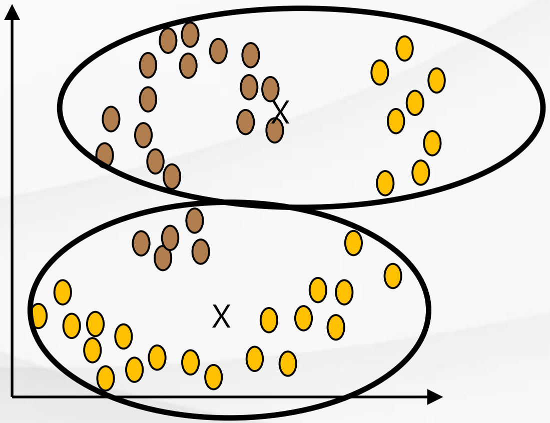
바람직한 클러스터링

K-means의 단점

K-means 알고리즘은 타원체 모양이 아닌 클러스터들을 찾는 문제에 적합하지 않음
(예> 고리 모양)



두 개의 고리 모양 클러스터



K-means 클러스터

K-means의 단점을 다루기 위한 방법

- 클러스터링 도중 다른 데이터 포인트보다 **Centroid**로부터 거리가 비이상적으로 먼 데이터 포인트를 제거해나감
  - 안전한 방법은 클러스터링 도중 아웃라이어가 발생하는지 모니터링 하다가 발생하면 지울지 말지 직접 결정
- 데이터로부터 무작위로 샘플링함. 샘플링은 전체 데이터 중 일부만 선택하기 때문에 아웃라이어가 선택될 확률은 낮음
  - 클러스터링이 종료되면, 샘플링 되지 않은 데이터 포인트들을 정해진 클러스터로 배정시킴

3. 거리 측정 함수들

3. 거리 측정 함수들

데이터의 특성이 수치값 (Numeric Value)을 가질 때

- 유클리디안 거리 측정 (**Euclidean Distance**)과 맨허튼 거리 측정 (**Manhattan Distance**)이 주로 사용됨

- 두 개의 데이터 포인트(x_i와 x_j)들의 거리를 측정할 때 다음과 같이 표기 $dist(x_i, x_j)$

- 두 측정 방법은 **Minkowski Distance**의 특별한 예

$$dist(x_i, x_j) = ((x_i, x_j)^h + (x_{i2}, x_{j2})^h + \cdots + (x_{ir}, x_{jr})^h)^{\frac{1}{h}}$$

3. 거리 측정 함수들

데이터의 특성이 수치값 (Numeric Value)을 가질 때

$h = 2$인 경우 유클리디안 거리 측정

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ir} - x_{jr})^2}$$

$h = 1$인 경우 맨허튼 거리 측정

$$dist(\mathbf{x}_i, \mathbf{x}_j) = | x_{i1} - x_{j1} | + | x_{i2} - x_{j2} | + \ldots + | x_{ir} - x_{jr} |$$

가중치 적용 유클리디안 거리 측정

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + \ldots + w_r(x_{ir} - x_{jr})^2}$$

Tacademy

데이터의 특성이 수치값 (Numeric Value)을 가질 때

제곱 유클리디안(Squared Euclidean distance) 거리

멀리 떨어져 있는 데이터 포인터들에게 더 많은 가중치를 줄 경우

$$dist(\mathbf{x}_i, \mathbf{x}_j) = (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ir} - x_{jr})^2$$

Chebychev distance

데이터의 특성들 중 어느 하나라도 다를 경우, '다름'으로만 정의하고자 하는 경우

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \max(|x_{i1} - x_{j1}|, |x_{i2} - x_{j2}|, \ldots, |x_{ir} - x_{jr}|)$$

Tacademy

데이터의 특성이 이산값 (Binary Value)을 가질 때

- 이산적 특성: 두 개의 값 또는 상태를 가짐

  예 > 성별: 남자, 여자

- Confusion 행렬을 사용하여 거리 함수를 정의함

		데이터 포인트 i		
		1	0	
데이터 포인트 j	1	a	b	a+b
	0	c	D	c+d
		a+c	b+d	a+b+c+d

a : 두 개의 데이터 포인트들이 모두 1값을 갖는 속성의 개수

b : 데이터 포인트 j는 1값을 갖고 i는 0값을 갖는 속성의 개수

c : 데이터 포인트 j는 0값을 갖고 i는 1값을 갖는 속성의 개수

d : 두 개의 데이터 포인트들이 모두 0 값을 갖는 속성의 개수

데이터의 특성이 이산값 (Binary Value)을 가질 때

- 이산적 특성은 두 개의 상태 (0 또는 1)이 동일하게 중요할 때, 대칭적임 (동일한 가중치)
- 거리함수: 단순 계수 비교 (일치하지 않은 값의 비율)

$$dist(x_i, xj) = \frac{b+c}{a+b+c+d}$$

x_1	1	1	1	0	1	0	0
x_2	0	1	1	0	0	1	0

$$dist(x_i, xj) = \frac{2+1}{2+2+1+2} = 3/7$$

Tacademy

데이터의 특성이 이산값 (Binary Value)을 가질 때

- 이산적 특성은 두 개의 상태 (0 또는 1)이 비대칭적일 때
- 한 상태가 다른 상태보다 더 중요한 경우
- 일반적으로 상태 1이 더 중요한 경우에 사용됨 (빈도가 더 낮아 희귀한 상태)
- Jaccard 계수 측정이 사용됨

$$dist(x_i, xj) = \frac{b+c}{a+b+c}$$

x_1	1	1	1	0	1	0	0
x_2	0	1	1	0	0	1	0

$$dist(x_i, xj) = \frac{2+1}{2+2+1} = 3/5$$


학습정리

학습정리

지금까지 [비지도 학습]에 대해서 살펴보았습니다.

클러스터링

클러스터링은 데이터에서 '클러스터(Clusters)'라는 '비슷한 그룹'을 찾는 기법을 뜻함
데이터의 그룹을 묶을 수 있는 어떠한 사전 정보도 주어지지 않기 때문에
비지도 학습(Unsupervised Learning)과 동의어로 사용됨

K-means 클러스터링

데이터 포인트들을 무작위로 K개 선택하여 Centroid 계산 →
클러스터 배정과 Centroid 재계산 (수렴 조건이 만족될 때까지 반복)
아웃라이어에 약한점에도 불구하고 단순함과 효율성으로 인해 널리 사용됨

거리 측정 함수들

수치적 특성과 이산적 특성에 따라 다양한 거리 함수 사용
수치적 특성: 유클리디안, 제곱 유클리디안, 멘허튼, Chebychev 거리 측정
이산적 특성: 단순 계수 비교, jaccard 계수 비교