기초 이론부터 실무 실습까지

머신 러닝 익히기

programmers

Part 07. K-means Clustering

정 정 민

programmers

Chapter 16. K-means Clustering

1. K-means Clustering 이란?
2. K-means Clustering 과정
3. 엘보우 방법 (Elbow Method)
4. 실루엣 계수 (silhouette coefficient)

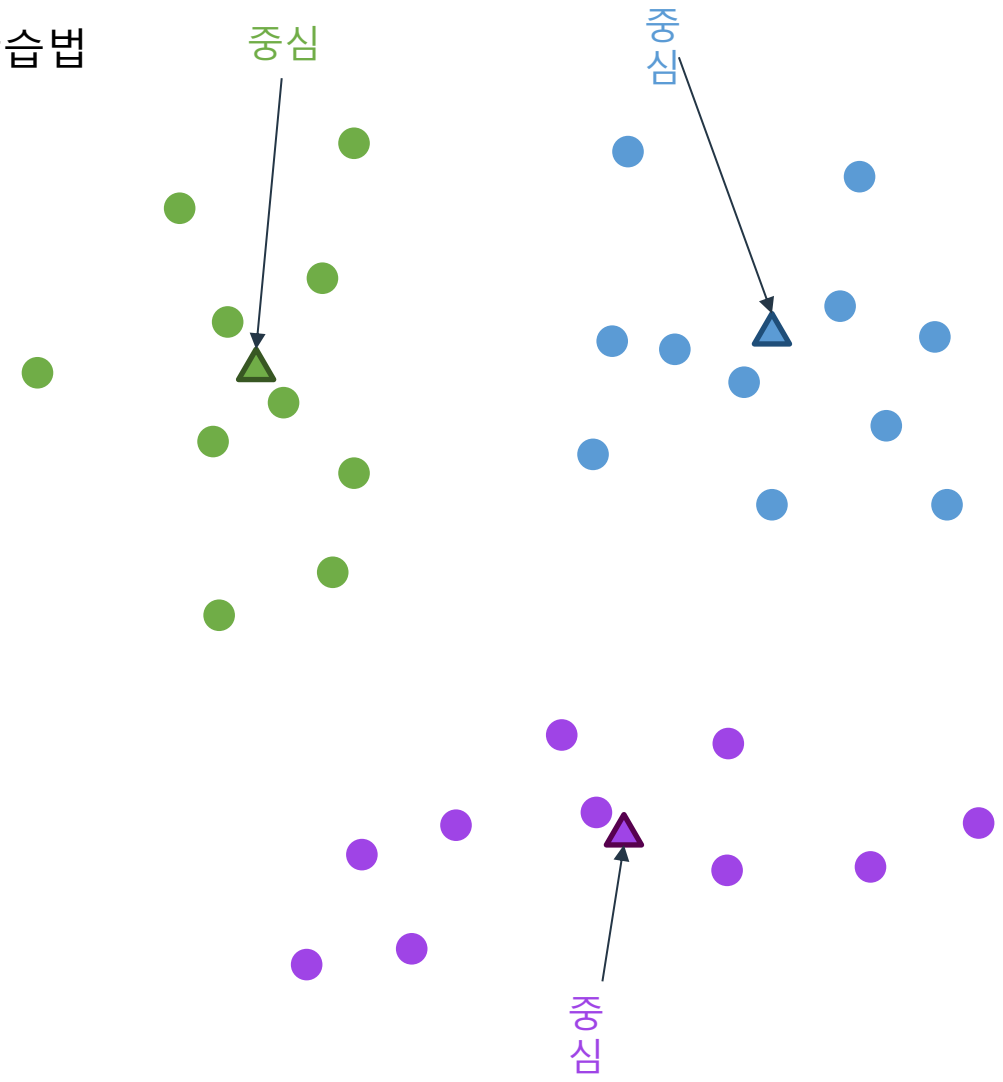
K-means Clustering 이란?

K-means Clustering

- ‘K-평균 군집화’라고 부르며
- 전체 데이터를 K개의 덩어리(클러스터)로 나누는 비지도 학습법

- 방법이 간단하며 효과적이고
- 결과 해석이 쉬워
- 많은 분야에서 사용됨!

- 오른쪽 그림은 K=3인 경우의 클러스터링
- 삼각형(△)은 클러스터의 중심점을 표시
  - 중심점은 클러스터 안에 포함된 데이터의 평균값



K-means Clustering 과정

강의 실습 코드 : 링크

K-means 알고리즘의 4 단계

- K-means 군집화 기법을 푸는 유명한 두 알고리즘은 아래와 같음
  - 로이드 (Lloyd) 알고리즘
  - 엘칸 (Elkan) 알고리즘

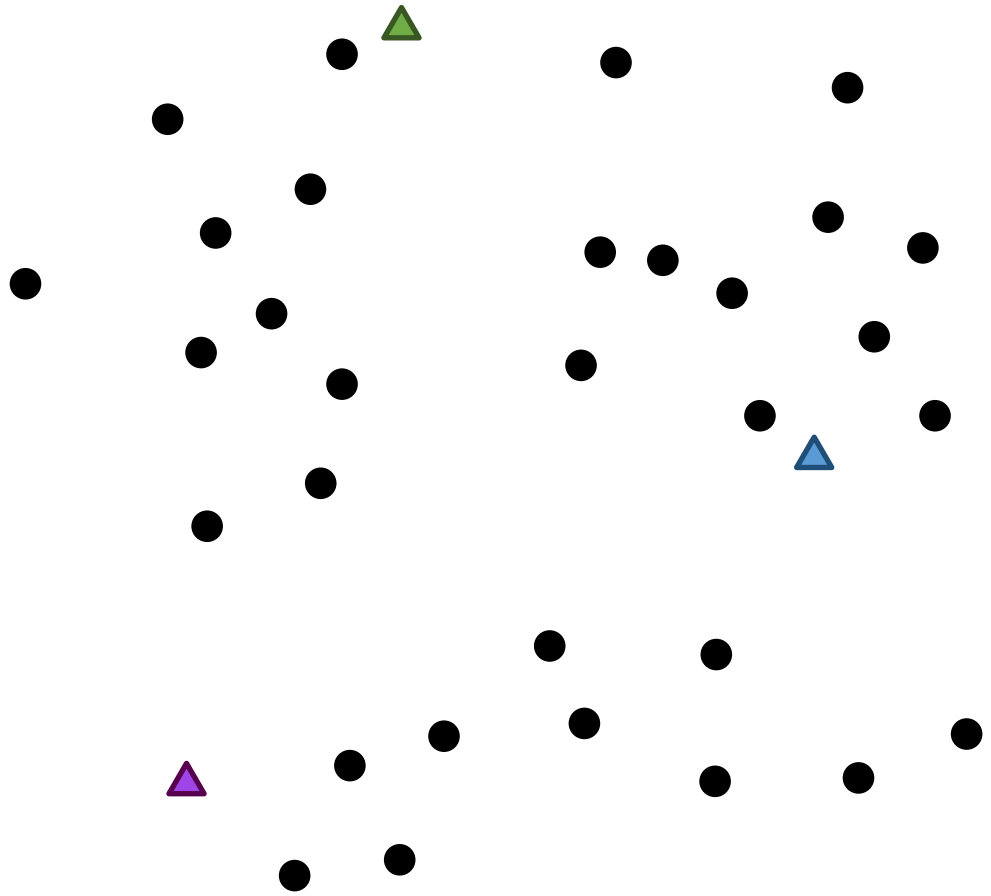
- 로이드 알고리즘이 가장 기본적인 방법
- **로이드 알고리즘**은 아래 4가지 단계로 구성됨
  - 초기화
  - 할당
  - 업데이트
  - 반복

- **엘칸 알고리즘**의 경우
  - 데이터 포인트와 클러스터 중심 거리를 계산하는 과정에 삼각 부등식을 사용
  - $|a + b| \leq |a| + |b|$

(1) 초기화

- K개의 클러스터 중심점을 임의로 선택

- 초기 위치는 최종 결과에 큰 영향을 미칠 수 있음
- k-means++ 초기화 방법을 많이 사용
  - 초기 중심점의 위치를 멀리 떨어지게 설정
  - 임의의 랜덤 위치보다 좋은 결과를 보여줌

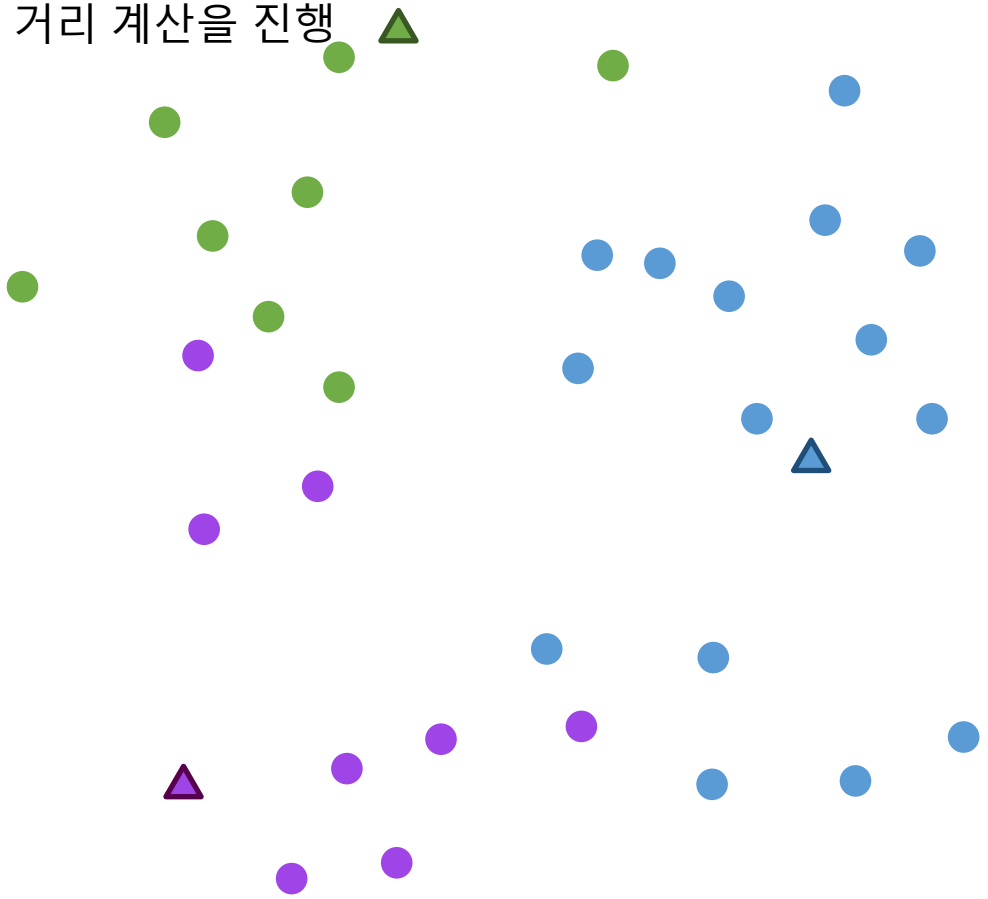


(2) 할당

- 각 데이터 포인트를 가장 가까운 클러스터 중심에 할당
- 일반적으로 유클리드 거리(Euclidean Distance)를 기반으로 거리 계산을 진행

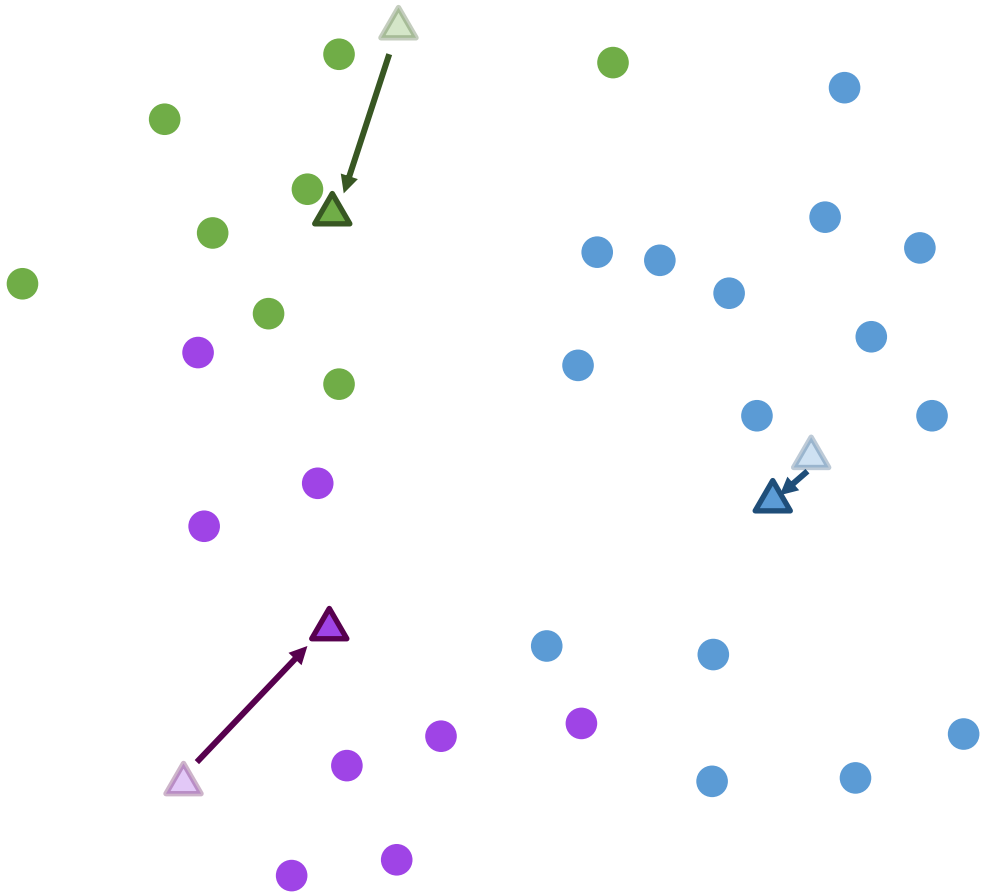
$$d(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + .. (a_n - b_n)^2}$$

- 대안으로,
  - 코사인 유사도 (Cosine Similarity)
  - 맨해튼 거리 (Manhattan Distance)



(3) 업데이트

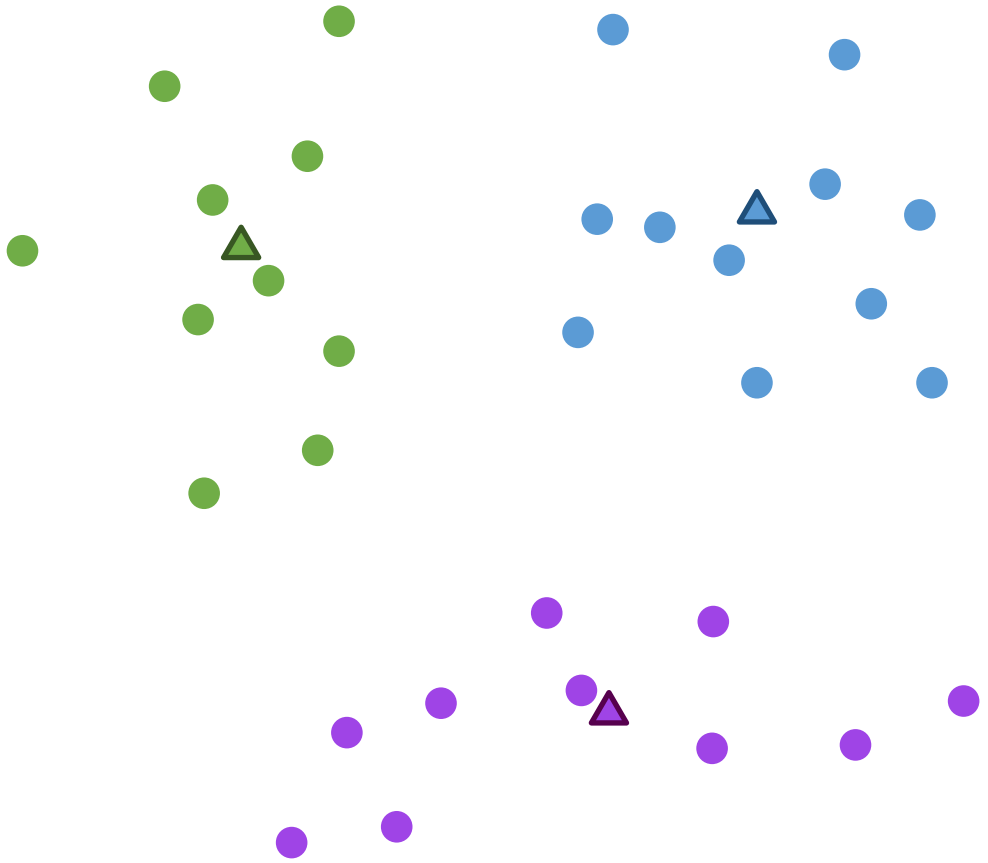
- 각 클러스터에 속한 데이터들의 평균점 위치로 클러스터 중심의 위치를 업데이트



(4) 반복

- 클러스터 중심의 변화가 미미할 때까지 할당 과정과 업데이트 단계를 반복

- 변화가 미미함의 정의는
  - 정말 위치의 변화가 없거나
  - 클러스터에 할당되는 데이터 포인트의 변화가 없거나
  - 동일한 데이터 포인트 할당 과정이 반복되거나
  - 지정된 횟수에 도달하거나
  - 등등



엘보우 방법 (Elbow Method)

강의 실습 코드 : 링크

K의 중요성

- 앞서 예시에서 데이터가 잘 나뉘게 된 이유는 적절한 K를 선택했기때문

- 만약 K가 너무 작다면
  - 중요한 하위 그룹을 잘 포착하지 못할 수 있음
  - 같은 클러스터 안에 서로 상당히 다른 데이터가 공존할 수 있음
  - 유의미한 인사이트를 얻기 어려움

- 혹은 K가 너무 크다면
  - 과적합 문제
  - 해석의 어려움
  - 효율성 저하

- 적절한 K를 고르는 방법을 엘보우 방법(Elbow Method)이라고 함

엘보우 방법 (Elbow Method)

- 클러스터 수를 늘려가며 각각에 대한 클러스터링 성능을 측정해, 클러스터 수에 따른 성능 변화를 분석
- **클러스터 수(K)**는 일반적으로 극히 작은 값(1)에서부터 매우 큰 값까지 사용
  - 사용하는 데이터의 수에 따라 매우 큰 K 값은 상이할 수 있음
- **클러스터링 성능은 SSE(Sum of Squared Errors) 값**을 활용
  - SSE : 각 클러스터 내의 데이터 포인트와 클러스터 중심 간의 거리의 제곱 합
  - 즉, 데이터 포인트가 클러스터 중심에 얼마나 가까운지를 나타냄
- 그래프 상 SSE의 감소율이 급격히 줄어드는 지점이 최적 클러스터 수(K)로 간주
  - 이런 지점이 마치 팔꿈치 같다고 해서 엘보우(Elbow)라는 이름이 되었다고 하네요!

실루엣 계수(Silhouette Coefficient)

강의 실습 코드 : 링크

클러스터링의 성능 평가

- 지도 학습의 과정처럼 비지도 학습의 학습 성능 평가를 진행하는 과정은 매우 중요함
- 군집화도 성능 측정이 중요함
  - 그래야 K의 수와 같은 중요 하이퍼파라미터를 튜닝할 수 있음

- 하지만 군집화와 같은 비지도 학습은 정답이 존재하지 않는 경우가 많아 성능을 측정이 쉽지 않음

- 그럼에도 일반적으로 많이 사용하는 평가 척도는 아래와 같음
  - SSE (Sum of Squared Errors)
  - 실루엣 계수 (Silhouette Coefficient)

실루엣 계수 (Silhouette Coefficient)

- 클러스터 안의 응집도와 서로 다른 클러스터 간의 분리도를 동시에 고려해 군집화의 품질을 평가하는 방법
- 이 값은 -1에서 +1 사이의 값을 가지며, 높은 값은 좋은 클러스터링을 의미함

- **응집도 (Cohesion) : $a(i)$**
  - 특정 데이터 i에 대해, 동일한 클러스터 안에 들어있는 다른 데이터들과의 평균 거리
  - 클러스터 내부의 데이터가 얼마나 모여있는지를 나타냄
- **분리도 (Separation) : $b(i)$**
  - 특정 데이터 i에 대해, i 가 들어있는 클러스터 말고, 다른 클러스터 중 가장 가까운 클러스터 중심까지 거리
  - 다른 클러스터와 얼마나 떨어져 있는지를 나타냄

- **실루엣 계수 : $s(i)$**
  - $s(i) = \frac{b(i)-a(i)}{max(a(i),b(i))}$
  - 최대값 : $1 \rightarrow a(i)$가 거의 0에 근접해 $b(i)$만 남는 상황 : 제일 좋은 상황
  - 최소값 : $-1 \rightarrow b(i)$가 값이 작아지고 오히려 $a(i)$가 커지는 경우 : 제일 나쁜 상황

[숙제] 더 해보기!

강사가 먼저 해본 결과를 공유합니다 : 링크
꼭 스스로 먼저 해보고 강사의 결과와 비교해보세요!

- 실루엣 계수도 Elbow Method에 사용할 수 있는 방법론입니다.

- 진행했던 코드를 똑같이 돌리되,
- SSE의 결과로 나왔던 Elbow point와
- 실루엣 계수로 나오는 Elbow point가 같을지 코드로 확인하기!

- **주의점!**
  - 실루엣 계수는 K=1인 상황에서 계산할 수 없음

E.O.D