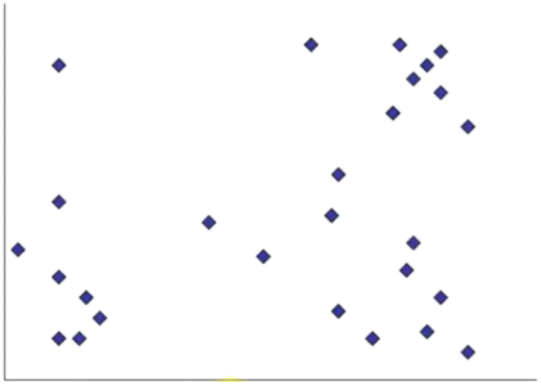
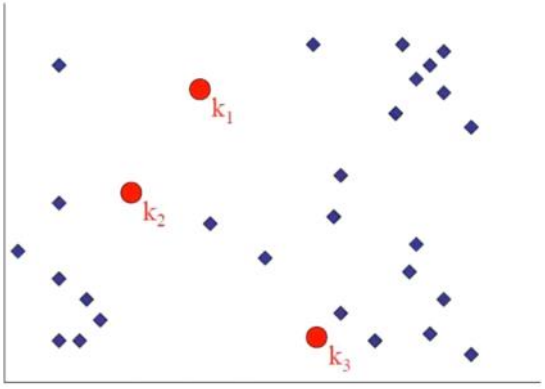
Kmeans(배포용)

2019년 1월 24일 목요일 오후 5:43

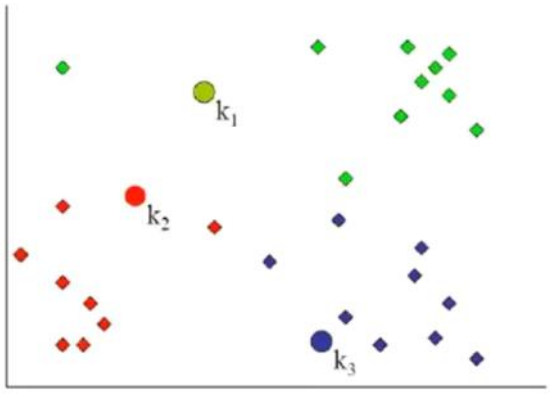
1. K-means 군집화의 실행단계
   a. 나누고자 하는 클러스터의 개수를 설정(분석가가 직접 설정)
      그 갯수만큼 임의의 초기점 생성
      예를 들어 개체 1000개 중에서 클러스터 4개를 찾는다면
      n(개체) = 1000
      k(클러스터) = 4
      몇 개의 클러스터(k)로 나눌지 분석자가 미리 결정해야 함.
   b. initial point를 군집 중심점으로 생각하고 군집 구분 실시
   c. 군집별로 새로운 군집 중심점을 계산
   d. 새로운 군집 중심점을 기준으로 군집 구분 실시
   e. c~d번 과정을 반복(더 이상 군집 구분의 변화가 없을 때까지)
      i. 다음과 같이 X, Y로 분포되어 있는 데이터들을 유사한 3개의 집단으로 군집화



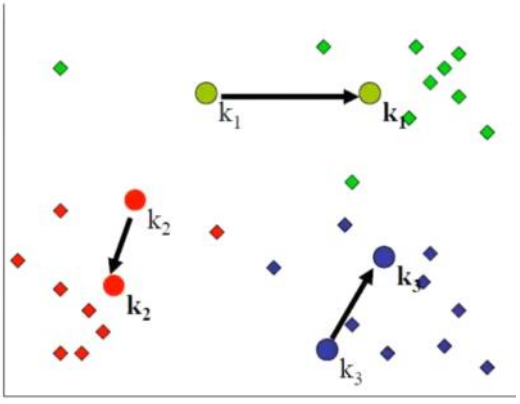
      ii. 우선 임의로 3개의 군집 중심점(임시)을 설정



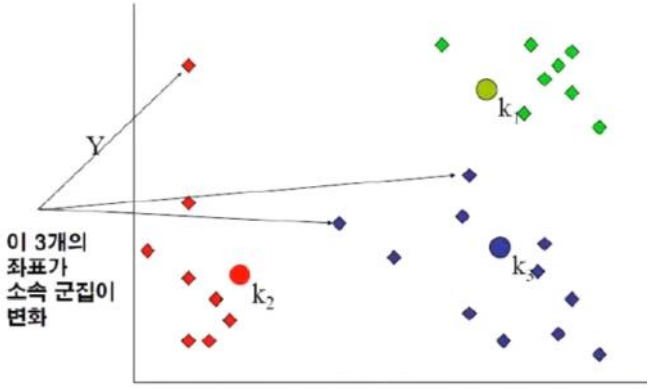
      iii. 임시로 설정된 세 좌표를 기준으로 군집화 수행



iv. 각 군집별 중심점을 계산하여, 새로운 중심점 설정



v. 새로운 중심점을 기준으로 군집화 수행 -> 일부 좌표의 소속 군집이 변화



vi. 다시 새로운 군집 중심정 설정(이 후 앞의 내용을 계속해서 반속)

2. K-means clustering 방법의 장/단점

a. 장점

적용이 쉽고 간단하다.

b. 단점

군집의 개수 K는 사전에 설정하여야 한다.

군집 설정에 Outlier(극단적 좌표)의 영향이 크게 작용한다.

3. 클러스터링 응용 사례

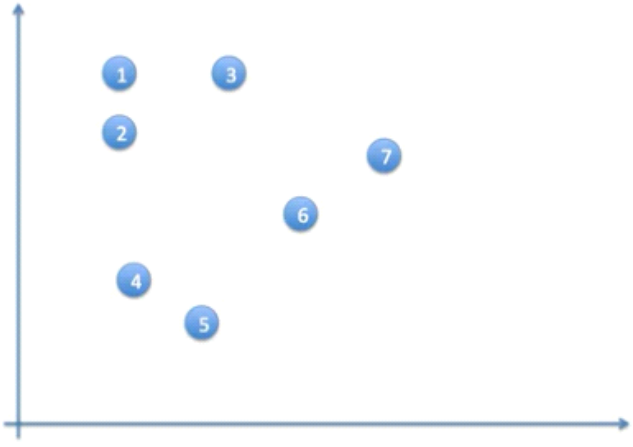
1. 페이스북 광고 : 특정 소비자 그룹에 적합한 광고 선택
2. 센서 빅데이터 그룹화
3. 이미지 분류 작업
4. 네트워크 유해 트래픽 감지

5. 언제 어디서 범죄가 발행할지 예측하여 예측 결과에 따라 순찰차 배치
6. 설문조사에 따른 소비자의 성향 분류
7. 뉴스, 문서 검색 결과의 주제별 분류
8. 공시지가 유사가격 권역 설정

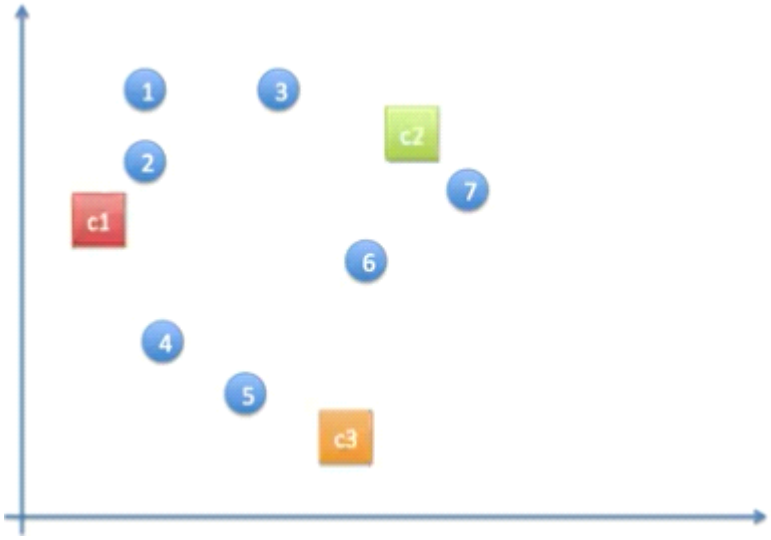
K-mean steps

1. **prepare data**
2. **decide how many clusters you need**
3. **choose initial center of cluster(centroid)**
   a. **randomly select centroid**
   b. **manually assign centroid**
   c. **kmean++**
4. **assign data point to nearest cluster**
5. **move centroid to the center of its cluster**
6. **repeat step 4 and step 5**
   until ther is no assigned cluster change

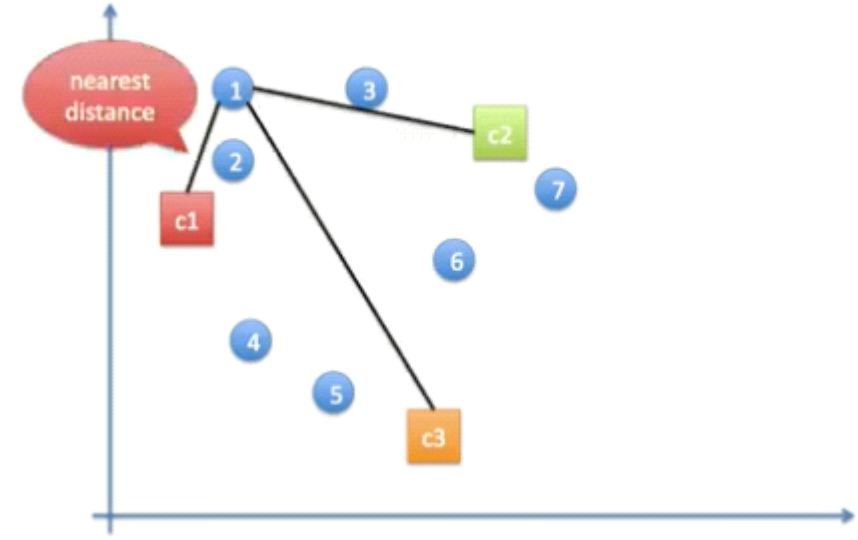
prepare data



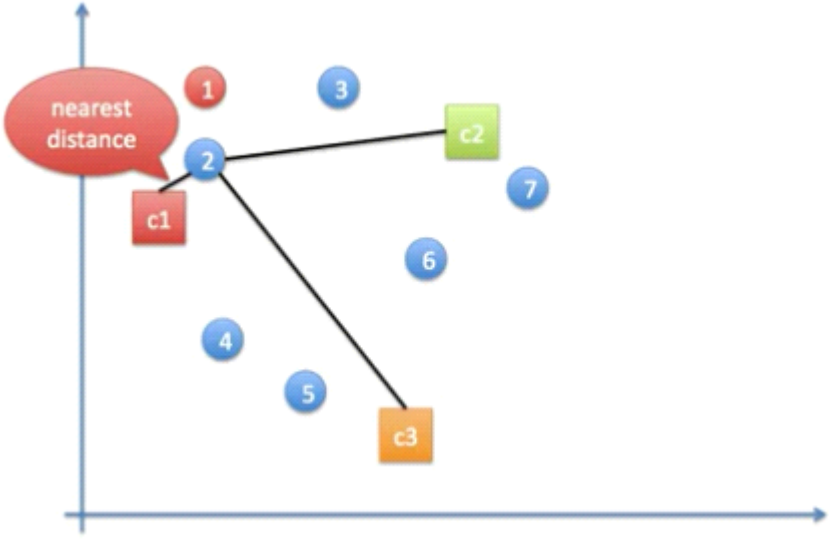
we want three clusters(k=3)



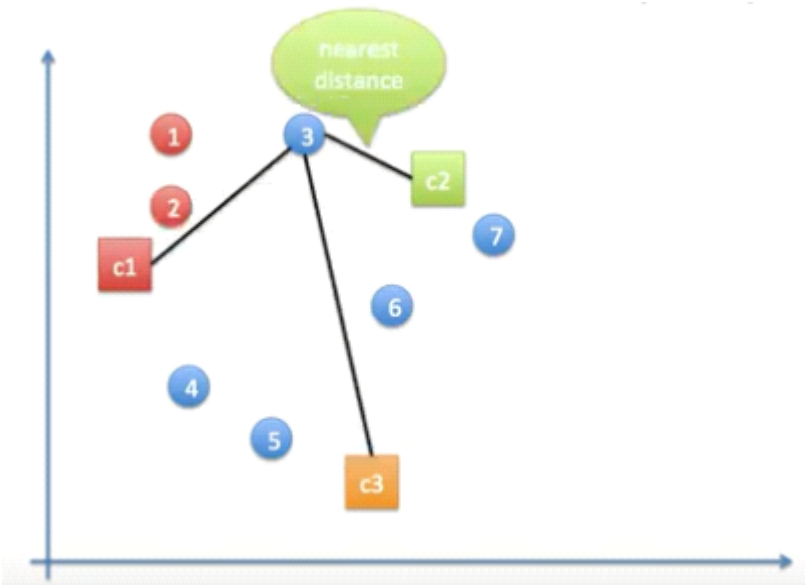
we want three clusters(k=3)



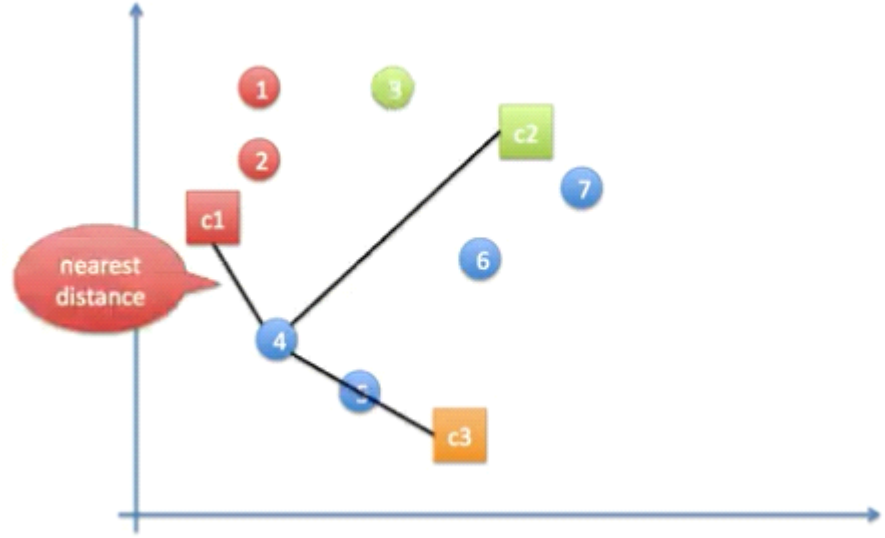
we want three clusters(k=3)



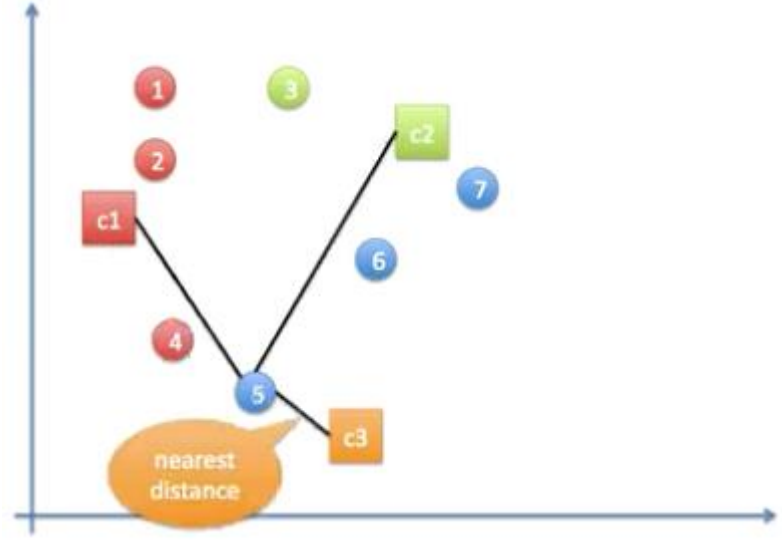
we want three clusters(k=3)



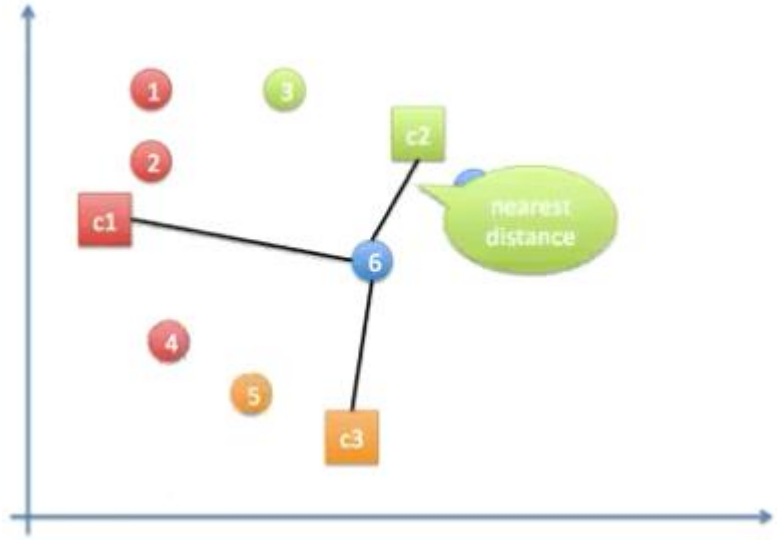
we want three clusters(k=3)



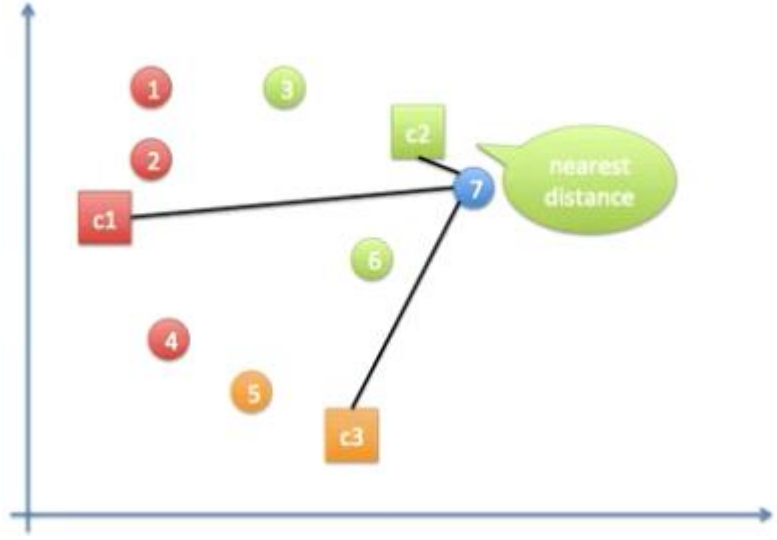
we want three clusters(k=3)



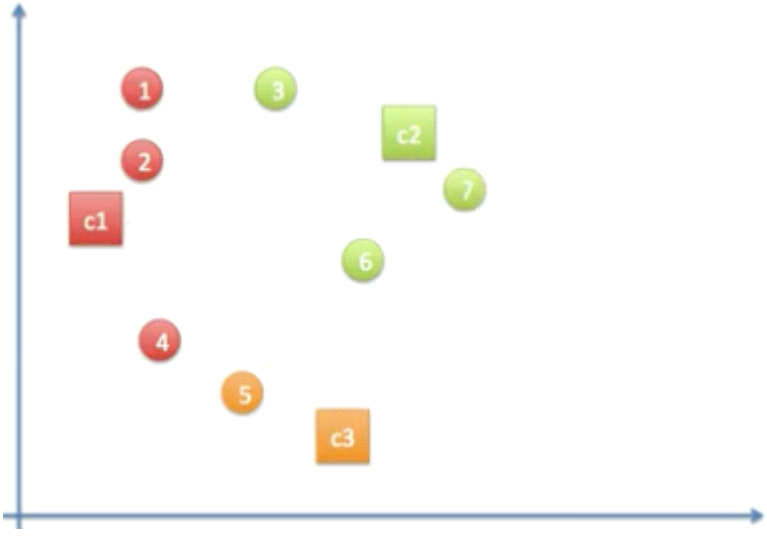
we want three clusters(k=3)



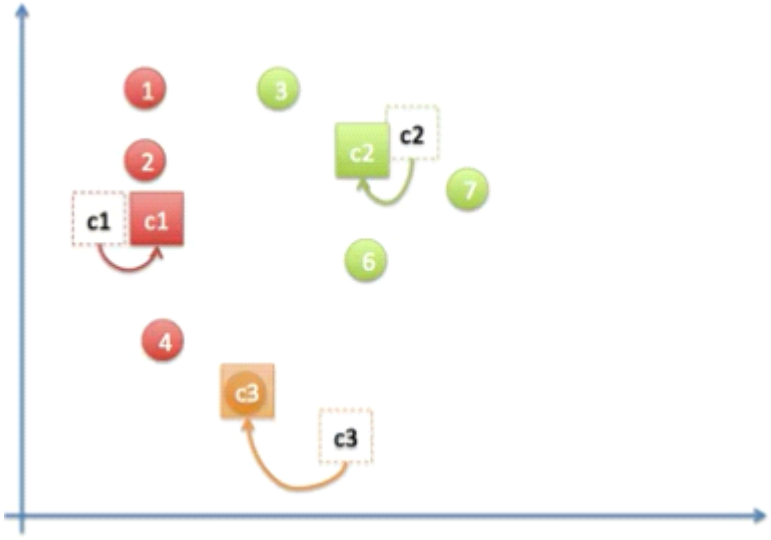
we want three clusters(k=3)



First assignment is done!

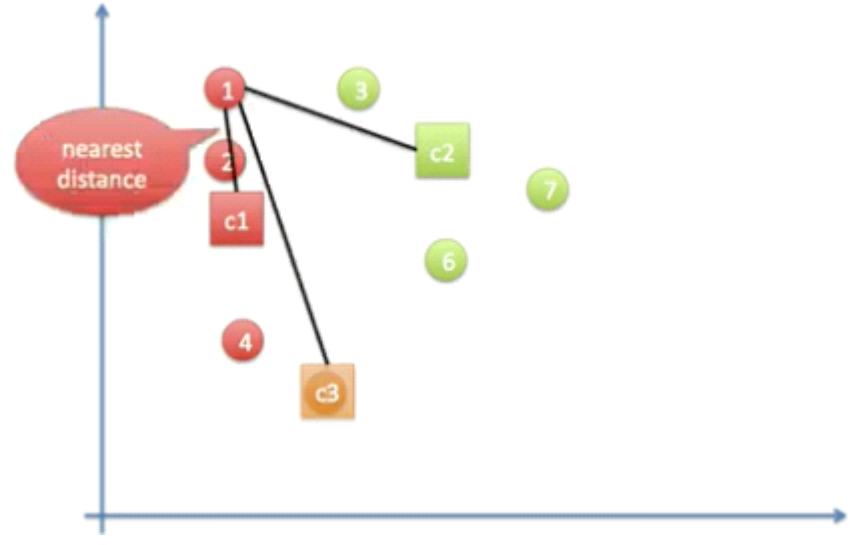


move centroid to the center of cluster

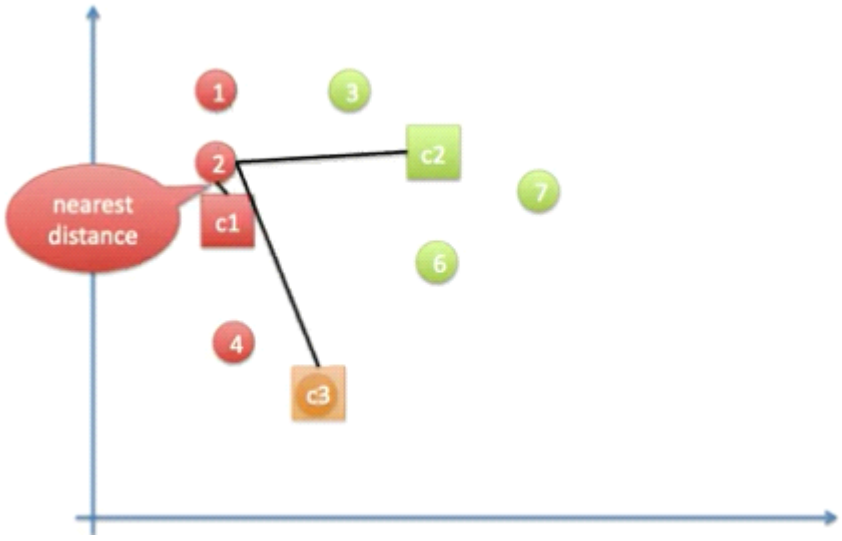


from each data point, assign cluster again using distance

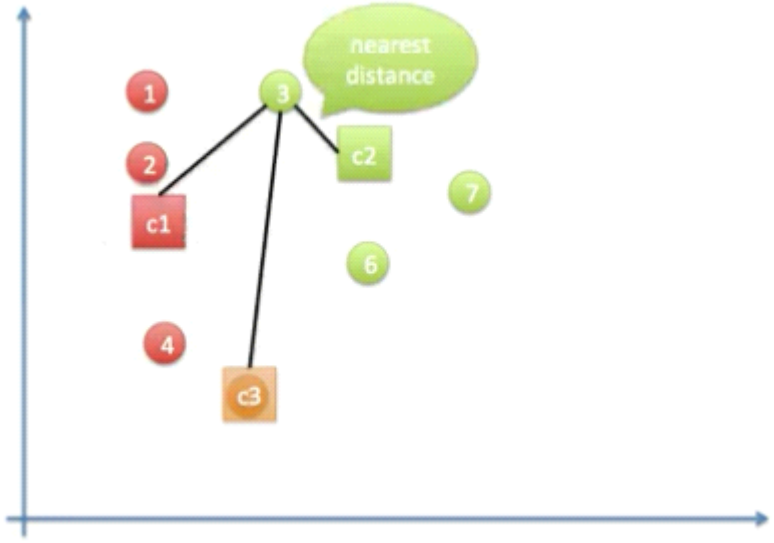
from each data point, assign cluster again using distance



from each data point, assign cluster again using distance

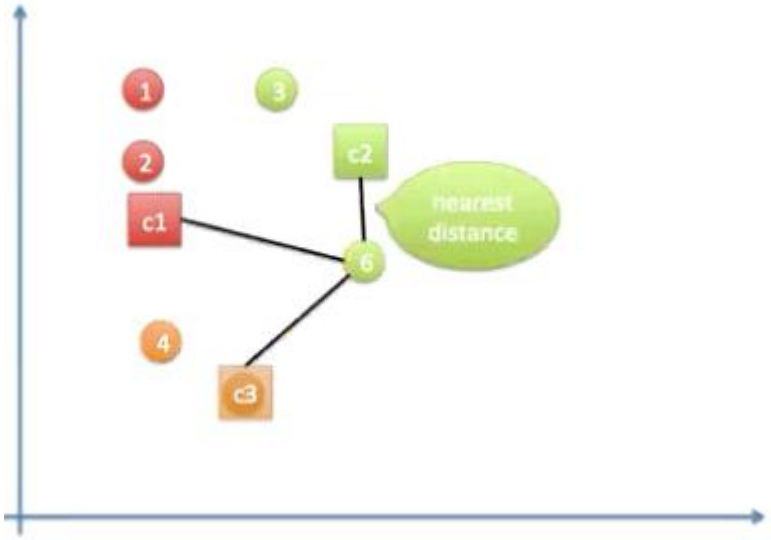


from each data point, assign cluster again using distance

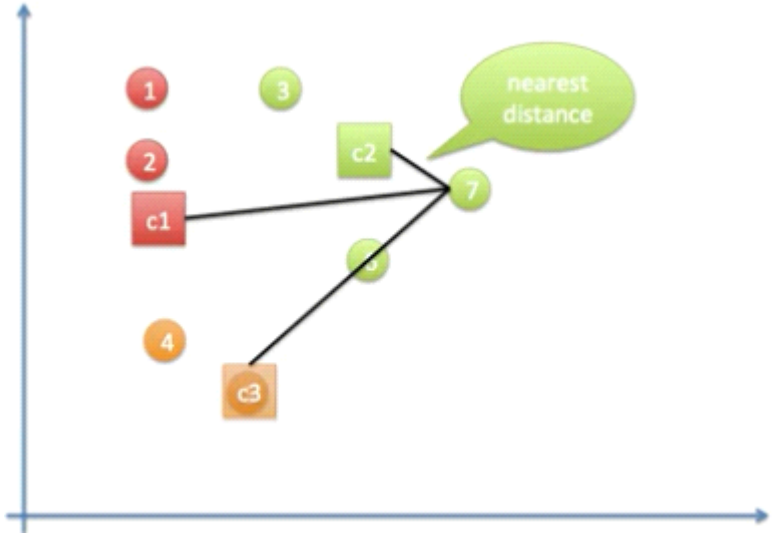


from each data point, assign cluster again using distance

from each data point, assign cluster again using distance

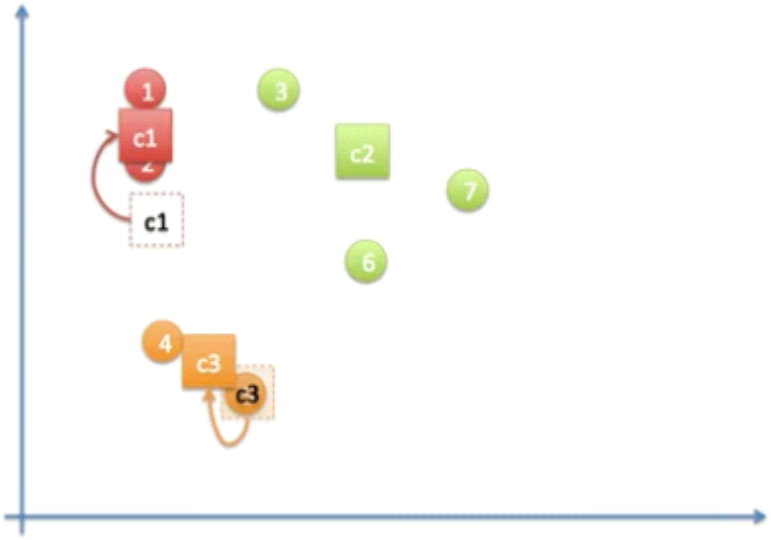


from each data point, assign cluster again using distance



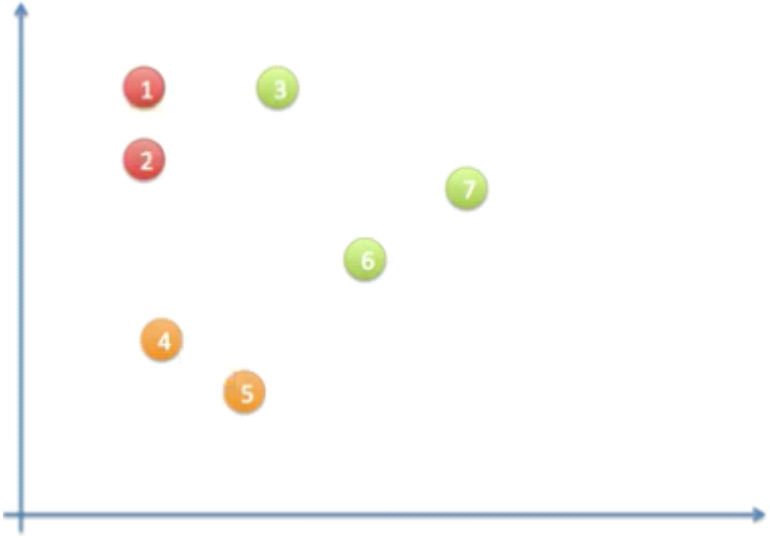
second iteration is done!

move centroid to the center of cluster



assign cluster again until there is no cluster assignment change or hit the maximum iteration count

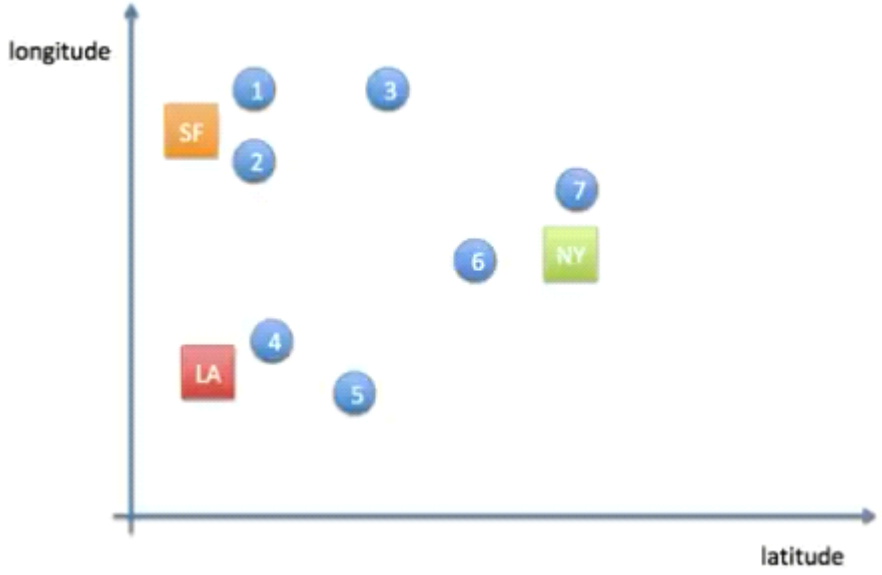
no cluster change, so k-mean clustering is done!



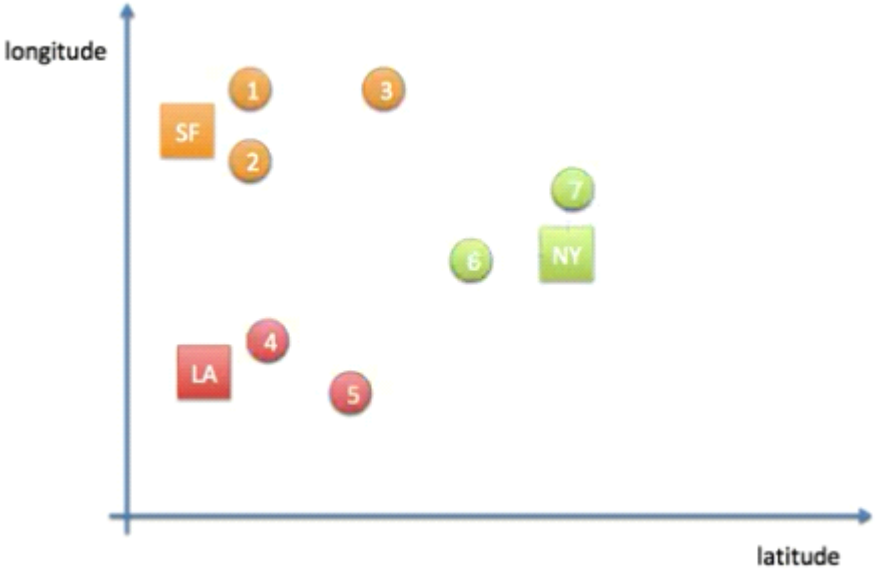
how to init centroid

1. randomly choose
2. manually assign init centroid
3. k-mean++

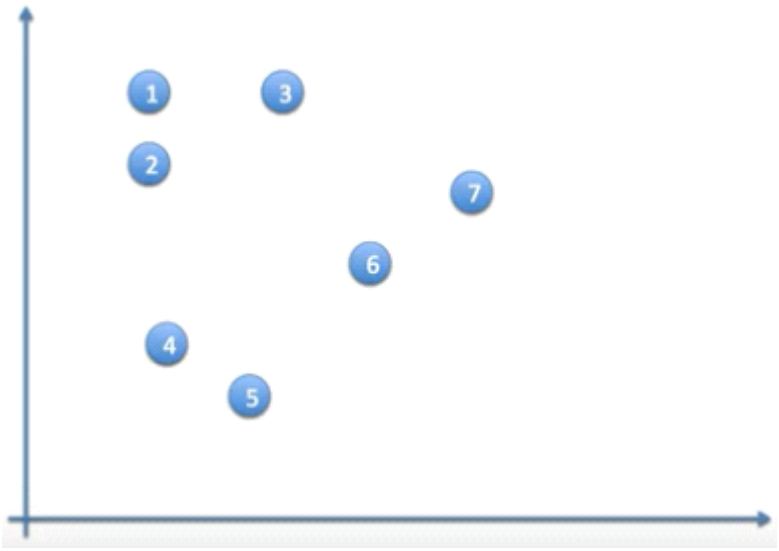
manually assign init centroid



manually assign init centroid

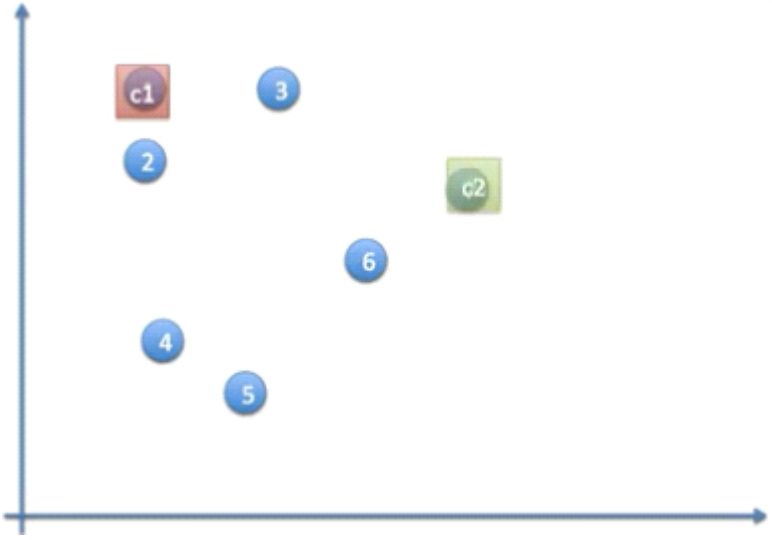


k-mean++ init centroid



k-mean++ init centroid

select farthest data point from first centroid as second centroid



k-mean++ init centroid

select farthest data point from centroid as next centroid