

언플러그드 인공지능 수학 수업 2

차형준

October 17, 2021

1 언플러그드 인공지능 수학 강의 지도안2

1.1 k평균 알고리즘을 이용한 언플러그드 수학 수업

1.1.1 이론적 배경: k-평균 군집(k-mean clustering)

- 주어진 데이터를 k개의 군집으로 묶는 알고리즘.
- 비지도학습의 한 종류.
- 데이터 레이블(데이터를 분류하는 값)이 정해지지 않은 경우에 분류하는 용도로 사용한다.
- 알고리즘 단계
 1. k개만큼 군집 수(군집의 중심)를 초깃값으로 정한다.
 2. 각 점을 가까운 초깃값에 배정해 군집을 만들어 나간다.
 3. 점이 군집에 할당될 때마다 군집의 평균을 재계산하여 군집의 중심을 갱신한다.
 4. 2~3의 과정을 반복한다.
- k값을 선정하는 방법: 엘보우, 실루엣, 덴드로그램
- 이상값에 민감하고, 군집의 모양이 복잡한 경우 분류가 어렵다는 단점이 있다.

1.1.2 수업 개요

- 단원명: 분류와 예측
- 성취 기준
 - [12인수03-01] 인공지능을 이용하여 텍스트를 분류하는 수학적 방법을 이해한다.
 - [12인수03-02] 인공지능을 이용하여 이미지를 분류하는 수학적 방법을 이해한다.
- 이론적 기반: 비지도 학습, 군집화 알고리즘, K-평균 알고리즘
- 수업 형태: 모둠 수업, 활동 수업, 개별 활동 수업으로도 진행 가능.

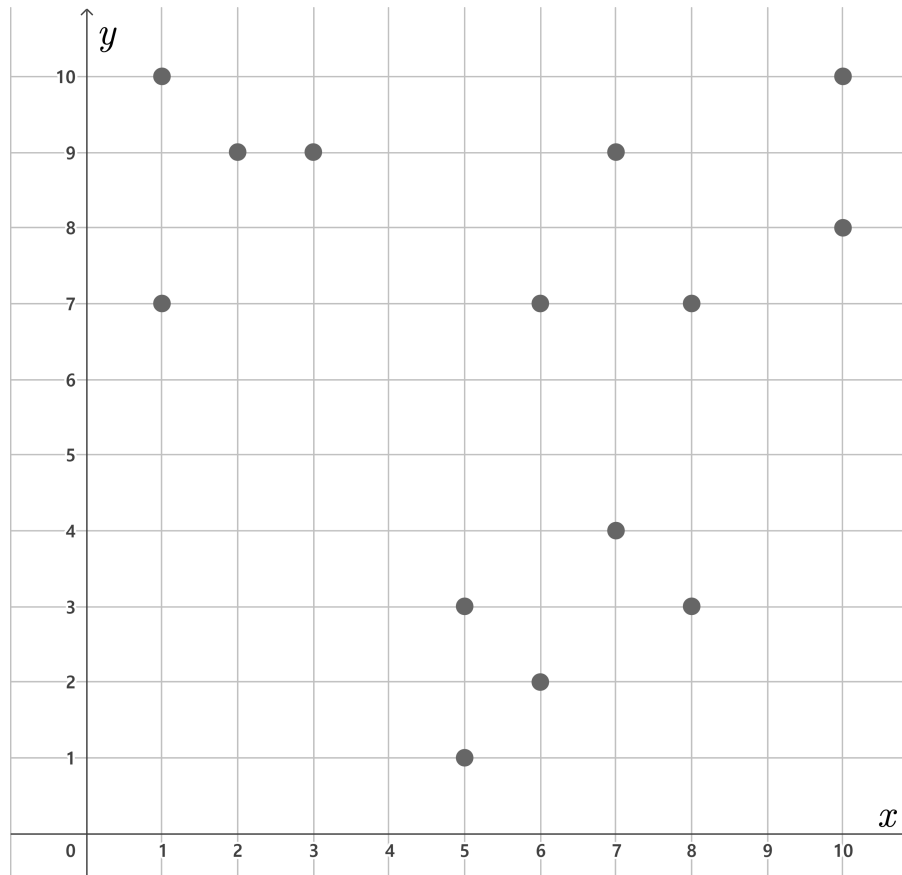
1.1.3 수업흐름

1. K평균 군집화가 필요한 상황 소개하기: 이미지 식별, 범죄예방을 위한 범죄유형분석, 왓챠 추천 영화.
2. 조별활동1: 주어진 그래프의 영역(군집)을 나누고 그 군집을 대표하는 점과 영역의 경계를 정해보기.
3. 조별발표: 그래프의 영역을 나눈 방법과 대표 숫자 그리고 경계 설정 이유를 설명해보기.
4. 발문1: 새로운 데이터를 주고 어떤 군집으로 들어가는지 체크하기.
5. 조별활동2: 새로운 데이터 분류하기. 분류했다면 어떤 근거로? 분류하지 못했다면 어떤 문제가 있었는지 정리하기.
6. 조별발표(자리에서 발표): 각각 분류한 결과와 그 근거를 이야기하기.
7. 발견하기: 데이터의 중심을 계산하는 방법과 분류하는 방법 안내하기.
8. 조별활동3: 각자의 주어진 중심 계산법 중 하나를 택해 군집의 중심 계산하기.
9. 관찰하기: 주어진 코드를 실행한 실제 군집 분류 결과 확인하기.
10. 토의하기: K-평균 알고리즘의 한계는? 해결 방법은?
11. 배운 내용 정리 및 마무리

1.1.4 평가방법

- 자기평가, 동료평가, 교사관찰평가
 1. 조별 활동 참여도
 2. 발문에 대한 토의 적극성
- 교사평가
 1. 발표 적극성
 2. 군집을 설정한 이유를 합리적으로 설정하였는지
 3. 군집의 중심 계산 방법에 대한 이해도

2 언플러그드 인공지능 수학 활동 학습지 2



1. 위의 주어진 점들을 여러 개의 군집으로 나누어보자.
2. 위와 같이 나눈 이유는 무엇인가요?
3. 분류한 군집마다 대표하는 점을 설정하고 그 점으로 설정한 이유를 적어봅시다.
4. 선생님이 불러준 새로운 점 적고 분류해봅시다.

5. 그렇게 분류한 이유는 무엇인가요?

6. 새로운 점에 대해서 직접 분류한 군집의 중심을 다음의 제시된 방법 중 두 가지 이상을 계산해봅시다.

- 최단연결법: 점과 군집 사이의 최단거리
- 최장연결법: 점과 군집 사이의 최장거리
- 중심연결법: 군집의 x 값과 y 값의 평균인 점과 점 사이 거리
- 평균연결법: 점과 군집 안의 모든 점과의 거리를 구해 평균을 구한 거리
- 와드연결법: 새로운 점을 포함해 오차제곱합(분산 \times 개수) 구하기

7. 위의 선택한 방법으로 새로운 점을 분류해봅시다.

```

[49]: # 실제 시뮬레이션

from sklearn.cluster import KMeans
import numpy as np
import matplotlib.pyplot as plt

# 주어진 데이터를 변수에 저장
data = np.array([[1,10],[2,9],[1,7],[3,9],[6,2],[7,4],[5,3],[8,3],[10,10],
[10,8],[8,7],[7,9],[5,1],[6,7]])
data_T = data.T

# 엘보우 기법을 이용해 군집 수 결정하기
wcss = []
for i in range(1,10):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit_transform(data)
    wcss.append(kmeans.inertia_)
plt.figure()
plt.plot(range(1,10),wcss)
plt.title('The Elbow Method')
plt.xlabel('k: Number of clusters')
plt.ylabel('WCSS')
plt.show()

# 군집 수가 3개일 때 분류하기
k = 3
kmeans = KMeans(n_clusters=k)
y_kmeans = kmeans.fit_predict(data)
data_T = np.vstack([data_T, y_kmeans])

# 군집 별 평균점을 포함해 그래프 그리기
kmeans_pred = KMeans(n_clusters=k, random_state=10).fit(data)
data_cluster_centers = kmeans_pred.cluster_centers_.T
plt.figure()
for i in range(k):
    plt.scatter(data_T[0],data_T[1], c=data_T[2], cmap=plt.cm.Set2_r)

x, y = np.meshgrid(np.arange(0,10.2,0.01), np.arange(0,10.2,0.01))
xy = np.c_[x.ravel(),y.ravel()]
z = kmeans_pred.predict(xy).reshape(x.shape)
plt.contourf(x, y, z, cmap=plt.cm.Set2_r, alpha=0.3)
plt.
    ↳scatter(data_cluster_centers[0],data_cluster_centers[1],c='darkred',marker='x')
plt.title('k-Means Clustering')
plt.show()

```

