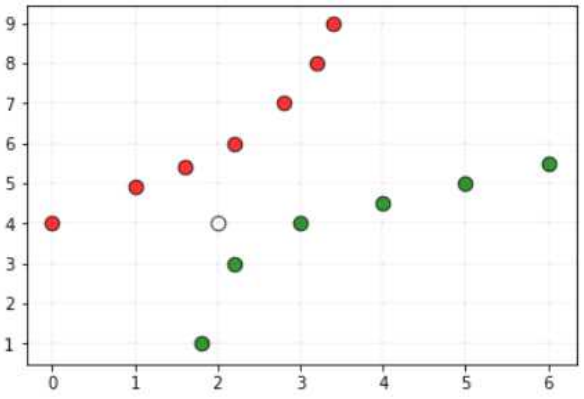
자료구조 (Homework 1)
2D-Array를 이용한 Weighted KNN 알고리즘 구현

제출기한 : 2021.4.16.(금), 5p.m. (늦은 제출은 받지 않음)

1. Weighted KNN 알고리즘 설명

KNN(K-Nearest Neighbor)은 어떤 새로운 데이터의 class를 예측하기 위해 기존에 가진 데이터들 가운데 현재 데이터와 가장 가까운 K개의 데이터를 뽑아, 그 데이터들의 class중 최빈(즉 가장 많이 출현한) class를 새로운 데이터의 class라고 예측하는 분류 알고리즘이다. 이 때 분류(Classification)란, 기계학습(Machine Learning)중 지도학습(Supervised Learning)의 일종으로, 기존에 존재하는 데이터들의 class를 기반으로 새로운 데이터의 class를 예측하는 task이다. 예를 들어, 어떤 메일이 스팸 메일인지 아닌지(2개의 class : 스팸o, 스팸x), 이미지 속 동물이 어떤 종류의 동물인지(class : 개, 고양이, 말 등)를 구분 짓는 것 등이 분류 문제에 속한다.

표준 KNN 알고리즘은 K개의 데이터 간의 거리에 따른 중요도는 고려하지 않고 단지 가장 많이 등장하는 class로 분류를 진행한다. 여기에 확장되어 Weighted KNN은 데이터 거리의 가중치를 통해 예측하려는 데이터에 더 가까울수록 분류에 영향을 주기 위함이다. 따라서 본 과제에서는 L2 norm의 크기에 반비례하는 가중치를 적용하는 Weighted KNN을 구현한다.



다음은 Weighted KNN(K=5) 알고리즘을 사용하여 두 개의 class(고등학생, 유치원생)를 분류하는 예시이다. 데이터의 features는 <키(cm), 몸무게(kg), 발 사이즈(mm)>로 세 가지이고, 총 10개의 데이터가 주어졌을 때 새로운 데이터의 class를 예측한다.

- 주어진 데이터

Sample #	Height (cm)	Weight (kg)	Foot size (mm)	Class
0	171.5	67.3	264.2	고등학생
1	163.2	51.4	233.2	고등학생
2	105.1	33.2	190.4	유치원생
3	183.3	78.7	271.2	고등학생
4	108.3	32.1	184.5	유치원생
5	159.6	49.2	225.7	고등학생
6	112.4	41.5	204.3	유치원생
7	98.8	29.9	177.2	유치원생
8	182.3	82.8	281.3	고등학생
9	101.1	36.3	192.5	유치원생

- class를 예측하고자 하는 새로운 데이터

Sample #	Height (cm)	Weight (kg)	Foot size (mm)	class
test	161.2	53.3	224.2	?

이때, test 데이터의 class를 예측하기 위해, 주어진 데이터들과 test 데이터 사이의 거리를 측정한다. 거리는 L2 distance(Euclidean distance)로 측정하며 측정 후에 test와 가장 가까운 K(=5)개의 데이터만 추출한다. test 데이터와 주어진 데이터들 사이의 거리는 아래와 같다.

Sample #	Height (cm)	Weight (kg)	Foot size (mm)	L2 distance	class
0	171.5	67.3	264.2	**(3) 43.61**	***고등학생***
1	163.2	51.4	233.2	**(2) 9.41**	***고등학생***
2	105.1	33.2	190.4	68.51	유치원생
3	183.3	78.7	271.2	**(5) 57.81**	***고등학생***
4	108.3	32.1	184.5	69.45	유치원생
5	159.6	49.2	225.7	**(1) 4.64**	***고등학생***
6	112.4	41.5	204.3	**(4) 54.00**	***유치원생***
7	98.8	29.9	177.2	81.54	유치원생
8	182.3	82.8	281.3	67.64	고등학생
9	101.1	36.3	192.5	70.04	유치원생

K(=5)개의 데이터를 아래와 같이 추출한 후 weight를 구하고 prediction을 진행한다.

[(Sample #, L2, class)]
= [(5, 4.64, 고등학생), (1, 9.41, 고등학생), (0, 43.61, 고등학생), (6, 54.00, 유치원생), (3, 57.81, 고등학생)]

→ L2 distance의 역수를 취해준 후 추출된 K개의 데이터 내에서 각 class별로 해당 weight를 합한다.

[(5, 0.215, 고등학생), (1, 0.106, 고등학생), (0, 0.022, 고등학생), (6, 0.018, 유치원생), (3, 0.017, 고등학생)]

고등학생 : 0.215 + 0.106 + 0.022 + 0.017 = **0.36**
유치원생 : **0.018**

이 때, 거리와 이에 반비례하는 가중치(weight)는 아래와 같이 계산한다.

(test 데이터와 i번째 데이터 사이의 거리 = $\sqrt{(161.2-키_i)^2+(53.3-몸무게_i)^2+(224.2-발사이즈_i)^2}$)

(test 데이터와 i번째 데이터의 가중치 = $\frac{1}{\sqrt{(161.2-키_i)^2+(53.3-몸무게_i)^2+(224.2-발사이즈_i)^2}}$)

∴ 고등학생 class의 weight의 합이 0.36이므로 유치원생의 weight의 합보다 더 크다. 따라서 test는 '고등학생' class임을 예측할 수 있다.

2. 문제 설명

- weighted_knn_학번.c 한 가지 파일만 작성한다.

- 데이터가 저장된 data.txt 파일을 읽어 2D-Array 자료구조에 저장한 뒤, test.txt 파일을 읽어 파일에 있는 데이터들의 class를 예측하여 output.txt 파일로 출력한다.

- data.txt 파일의 포맷은 다음과 같다.

<데이터 개수> <데이터의 feature 개수>
<첫번째 데이터의 첫번째 feature값> ... <첫번째 데이터의 마지막 feature값> <첫번째 데이터의 class>
<두번째 데이터의 첫번째 feature값> ... <두번째 데이터의 마지막 feature값> <두번째 데이터의 class>
...
<마지막 데이터의 첫번째 feature값> ... <마지막 데이터의 마지막 feature값> <마지막 데이터의 class>

이때, 데이터 개수, 데이터의 feature 개수, 데이터의 class는 int형이고, 데이터의 feature 값들은 float형이다.

- test.txt 파일의 포맷은 다음과 같다.

<test 데이터 개수>
<첫번째 test 데이터의 첫번째 feature값> ... <첫번째 test 데이터의 마지막 feature값>
<두번째 test 데이터의 첫번째 feature값> ... <두번째 test 데이터의 마지막 feature값>
...
<마지막 test 데이터의 첫번째 feature값> ... <마지막 test 데이터의 마지막 feature값>

test 데이터 개수의 자료형은 int, 데이터의 feature값은 float형이다. (test 데이터의 feature 개수는 data.txt와 동일하다)

- output.txt 파일의 포맷은 다음과 같다.

<첫번째 test 데이터의 class>
<두번째 test 데이터의 class>
...
<마지막 test 데이터의 class>

- weighted_knn_학번.c, data.txt, test.txt, output.txt 파일들은 전부 같은 디렉토리에 있어야 한다.

- K값은 5, 거리 척도는 L2 distance를 사용하여 예측한다.

- 채점은 다른 테스트 파일을 통해 진행되지만, 파일의 포맷은 같다.

3. 제출

- weighted_knn_학번.c 파일을 압축하여 이름_학번.zip 파일을 사이버캠퍼스에 제출한다.