

알고리즘 실습 보고서

- SLS -

전공 : 컴퓨터공학과

분반 : 05반

학번 : 201502085

이름 : 이규봉

1. 실행 환경

과제의 코드를 테스트한 환경은 아래와 같습니다. 행렬 관련 계산을 수월하게 하기 위해 numpy를 이용했습니다.

- Windows 10
- Pycharm 2019.02
- Python 3.7.4
- Numpy 1.16.4

2. 과제 설명

- DP와 Least Square Method를 이용한 Segmented least squares 문제 해결

3. 문제 해결 방법

- 문제를 해결하기 위해 자신이 사용한 방법, 아이디어에 대한 설명
(구현한 코드에 대한 설명 등)

행렬계산을 이용해 Least Square Method로 주어진 구간의 오차를 최소화하는 직선을 구합니다. 이 과정에서 역행렬이 존재하지 않는 케이스를 배제하기 위해, 의사 역행렬을 구하는 메서드인 `numpy.pinv`를 이용했습니다. 이 과정을 계속 반복하며 각 구간에서의 에러 값과 직선을 구성하는 기울기와 y 절편 값, 오차 값을 찾습니다. 그리고 DP로 주어진 인자값 까지의 `opt` 값을 반복해서 구하면, Cost of the optimal solution 값은 `opt[-1]` (마지막 값) 값이 됩니다. 그리고 `dividingPtAtJ`라는 리스트에도 역시 DP를 적용해 각 구간에서 어떻게 직선을 나누었는지에 대한 정보를 기록해놓았다가, `dividingPts`에 `pointCnt`와 함께 튜플의 원소로 주면, 해당 문제에서의 잘린 직선들의 구간들이 됩니다. 따라서, 이 직선들의 정보를 반복문에서 출력하면 됩니다.

4. 결과 화면

```
C:\Users\jopem\Anaconda3\python.exe C:/Algorithm-HW/Assign07/main.py
Cost of the optimal solution : 4.311779160739688

An optimal solution :
[Segment 0 - 5 : y = -0.018421052631579365 * x + 1.2986842105263157 // square error: 0.5907105263157895
[Segment 5 - 12 : y = 1.0485419630156412 * x + -2.9538051209103466 // square error: 1.7210686344238981

Process finished with exit code 0
```

5. 느낀점 및 고찰

Least Square Method를 반복해 적용하며 DP를 이용해 적당히 잘라야할 지점을 찾고 Optimal Solution 값을 구하는 것이 신기했습니다. numpy 사용방법을 배우는 등 으로 시간이 소요되어 문제 풀이에는 총 5 ~ 6 시간 정도 소요된 것 같습니다.