

오픈소스 SW 프로젝트 레포트

Traveling Santa 2018 - Prime Path

지도교수 나인섭

조선대학교 IT융합대학
20134893 박지종

오픈소스 SW 프로젝트 레포트

Traveling Santa 2018 - Prime Path

주제

- Traveling Salesman Problem의 변형인 Traveling Santa 2018 - Prime Path
- kaggle link: <https://www.kaggle.com/c/traveling-santa-2018-prime-paths>
- 좌표 데이터가 주어지면 가장 적은 Cost로 모든 좌표에 한번씩 방문하여 순회 할 수 있는 경로를 찾는 것이 목적.
- 북극(CityId = 0)에서 시작하고 끝나야함.
- 모든 도시를 정확히 한 번 방문해야 함.
- 두 지점간의 거리는 2D 유클리드 거리임.
- 정확히 10번째 단계마다 프라임 도시(CityId 값이 소수)에서 출발하지 못하면 다음 목적지까지 도달하는데 평소보다 10% 오래 걸림.

개발 환경

- 운영 체제
 - Apple MacOS Mojave 10.14.2
 - ▶ Apple MacBook Pro 13, 2017
 - ▶ Intel i5-7267U
 - ▶ Intel Iris Plus 650
 - Microsoft Windows 10 Pro 1803
 - ▶ AMD Ryzen7 1700
 - ▶ Nvidia GTX1070
- 개발언어
 - Jupyter Notebook (Python 3)
 - Python 3.7.2
- 라이브러리
 - Python
 - Pandas
 - Numpy
 - Seaborn
 - matplotlib
 - Cython(C-Extensions for Python)
 - PyConcorde ⁴⁾

필요성 및 기대효과

- 다수의 방문해야 하는 곳의 좌표를 입력하고, 가장 효율적인 경로를 찾는 것이 일차적 목표
- 중간중간 반드시 필요한 보급을 할 수 있는 다수의 포인트를 지정하고, 일정 거리마다 방문해야 함
- 다수의 경유지를 지정할 수 있는 내비게이션의 탐색 알고리즘으로 이용될 수 있음
- 발전 시켜 물류업에 적용하면 비용, 시간적으로 효율적인 경로를 예측 가능

기존 연구

1. 탐욕 알고리즘

- 미래를 생각하지 않고 각 단계에서 최선의 것을 선택하는 방법
- 각 단계의 가장 최선의 선택이 전체적으로도 최선이 되기를 바라는 알고리즘
- 경우에 따라서 좋은 결과를 내지 않음

2. Concorde TSP Solver ⁵⁾

- 외판원 순회 문제(Traveling Salesman Problem)와 이와 연관된 네트워크 최적화 문제를 위한 컴퓨터 코드
- ANSI C 프로그래밍 언어로 작성됨
- TSPLIB(<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>)의 인스턴스 110개 모두에 대한 최적의 솔루션을 얻는 데 사용됨(최대 85900개 도시)

제안 방법

실험 및 결과

고찰 일지

2019. 01. 07 월요일

이제 막 프로그래밍 언어 기초에서 발걸음을 뗀 수준이라 좀 더 접하기 쉬운 주제가 있지 않을까 싶어서 알고리즘 강의 때 다룬 적 있었던 TSP 문제를 골랐다.

Traveling Santa 2018 - Prime Path가 그것인데 역시 강의에서 잠깐 언급하고 지나가는 것과 실제로 구현하는 것은 굉장한 차이가 있다. 어떻게 해야 할 지 막막하다. 지금은 그저 다른 사람이 만들어 놓은 커널을 보면서 코드를 공부하는 수 밖에 없는 것 같다.

2019. 01. 08. 화요일

어떤 코드를 먼저 알아봐야 할까 고민하다가, 여러 커널 중 Starter code, beginner라는 태그, 그리고 알고리즘 강의때 종종 들었던 그리디 알고리즘이 붙어 있는 커널²⁾을 먼저 살펴봤다. 그리디 알고리즘은 이후의 사항을 고려하지 않고 각 단계에서 최상의 선택지를 골라가는 것이다. 하지만 이 알고리즘은 TSP에 대해 좋은 답을 도출해내지는 못한다. 그래서 그런지 이 커널의 코드를 실행해보면 결과물은 나오지만 솔루션 점수는 좋지 못하다. 심지어 마지막으로 방문한 도시에서 북극(CityId = 0)까지의 거리도 상당히 멀다. 그래서 이 커널은 잠시 접어두고 다른 커널을 찾아보았다.

두번째로 찾은 커널은 Concorde⁵⁾라는 TSP Solver를 이용한 것³⁾이다. Concorde는 C로 작성된 TSP 종류의 문제를 풀 수 있는 코드인데, 이걸 파이썬에서 활용할 수 있도록 jvkersch가 github에 게시한 pyconcorde 라이브러리를 활용해 푼 것이다(Concorde 이용 전 모든 도시를 CityId의 순번대로 순차적으로 순회하는 모습도 보여준다). 우선 코드를 받아서 실행해보려, pyconcorde를 받아 설치하려 하지만 설치 진행부터 막힌다.

2019. 01. 09 수요일

pyconcorde를 설치하는 파이썬 스크립트 코드를 열어보니 파이썬을 위한 C언어 확장인 Cython이 필요했다. pip를 이용해 손쉽게 설치 가능했다. Cython 설치 후 다시 시도해보니 첫 시도보다 진척이 있었다. 하지만 이번에는 웹에서 Concorde와 QSOpt를 받아오는 것에서 문제가 생긴 것. 이 것도 무리없이 스크립트 코드를 통해 힌트를 얻어 다운로드 하고 요구하는 폴더에 넣어 해결 가능했다. 이제 제대로 되려나 싶었지만, Concorde를 이용한 솔루션 코드를 실행하려니 concorde.tsp 모듈을 import 하는 라인에서 tsp라는 이름을 concorde 모듈에서 찾을 수 없다는 에러를 표출한다. pyconcorde는 제대로 설치된 것 같지만 원판 concorde의 설치에 문제가 있는 것 같아 직접 가이드대로 직접 필요한 파일을 받아 빌드를 시도해 보았지만 계속해서 빌드에 실패한다. Xcode에 포함된 Clang 컴파일러가 linker command failed with exit code 1 메시지만 띄우고 있다. 원인은 다운로드 받은 QSOpt의 라이브러리가 아키텍처 버전이 맞지 않기 때문이었다. 다른 맞는 버전의 파일을 다시 받아 시도하니 정상적으로 빌드 되었고, 무사히 설치 되

었다. 하지만 여전히 파이썬 코드에서 `concorde.tsp` 모듈을 불러올 수 없다. 다른 방법을 강구해야겠다.

참고문헌

- 1) 캐글 컴페티션: Traveling Santa 2018 - Prime Paths (<https://www.kaggle.com/c/traveling-santa-2018-prime-paths>)
- 2) Theo Viel, Greedy Reindeer - Starter code (<https://www.kaggle.com/theoviel/greedy-reindeer-starter-code/>)
- 3) Andre Sionek, Object Oriented Santa's Route + Concorde Solver (<https://www.kaggle.com/andresionek/object-oriented-santa-s-route-concorde-solver>)
- 4) jvkersch, pyconcorde (<https://github.com/jvkersch/pyconcorde>)
- 5) Concorde TSP Solver Home (<http://www.math.uwaterloo.ca/tsp/concorde.html>)

느낀 점

git 주소: https://github.com/Scharnhorst93/Traveling_Santa_2018

2019. 01. 09 수요일

기나긴 방황을 마치려 17년 복학 이후로 성적이 전무한 필수 과목들부터 차근차근 수강 해왔고, 이게 교수님들께 어떻게 보였을지 모르지만 몇몇 교수님으로부터는 과찬까지 듣기도 했는데, 역시 그런 칭찬을 들을 정도는 아니구나 싶다. 그간 받은 평균을 상회하는 성적은 학부 1~2학년 수준에 불과하고 실전으로 들어가려면 한참 멀었고 공부해야 할 것은 차고 넘친다는 것을 새삼 느꼈다. 평소에도 고쳐야 한다고 생각하는 게으른 성격이 재차 적나라하게 느껴질 정도였다. 시간이 턱없이 부족하다. 멘토도 있으면 좋겠다. 이제야 프로그래밍 언어 첫 발걸음을 땀 나에겐 간단한 알고리즘 문제조차도 버겁다. 아는 것이 없으니 질문도 못하고, 이 개인 프로젝트를 진행 하는 것도 어떻게 갈피를 잡아야 할 지도 모르겠다.

그래도 희망이 없는 건 아닌 것 같다. 적어도 지금은 난해한 코드나 알고리즘이 아니면 패턴이나 어떻게 흘러가는 지 어느정도 보이게 된 것 같다. 아마 이번 동계 계절학기 동안 끝내지 못하고 낙제점을 받을지도 모르지만 일단은 끝까지 붙잡고 실전 노하우를 톱밥만큼이라도 건져야겠다는 생각이다.