

Introduction to seoulsubway

KM Son

Department of Statistics, SKKU

April 17, 2019

1. 현황 및 분석목적

1.1 현황

1. 현재 지하철 승하차 통계량은 공공데이터포털을 통하여 제공된다.
2. 공공데이터 포털을 통하여 최단경로 OPEN API가 제공되고 있지만 여러가지 문제가 존재한다.
 - 을지로입구와 종각역을 동일한 역으로 인식하여 환승노선 구축 시 문제
 - 분기점 또는 지선의 경우 제공되지 않음.
 - topis를 기반으로 한 버스/지하철이 결합된 알고리즘에 따른 경로산출 문제

해결방안

수도권 지하철 네트워크를 구축하여 환승시간, 지하철 위치정보 등 공공데이터를 기반으로 한 최단경로 정의

모든 지하철 이용객들은 가장 시간이 적게 걸리는 이동 경로를 통하여 이동한다. 라는 가정을 두어 이동 간 최소 소요시간을 기준으로 최단경로를 정의한다.

1.2 분석목적

- 이용객 개별 승하차데이터를 통하여 인구흐름을 파악하고 싶다.
- 역사 내 혼잡도를 앞에서의 가정을 통하여 정의한 최단경로와 결합하여 추정한다.

2. 최단경로 함수

2.1 지하철 네트워크 구축

1. 위치정보를 이용하여 1-9호선, 경의중앙선, 분당선 등 총 22개의 수도권 지하철 노선이 포함된 지하철 네트워크를 구축하였다.
2. 지하철 노선 내 환승가능역을 통하여 다른 노선으로의 환승을 고려하였다.

2.2 최소 소요시간

1. Haversine 수식을 이용하여 역 간 소요시간을 계산한다.

우선, $\Theta = \frac{d}{r}$ 을 정의한다. 여기서 d 두 지점과의 거리이며, r 은 지구의 반지름이다. 지구의 반지름은 알려진 값이므로 다음 단계를 통하여 d 를 구한다.

$$\text{hav}(\Theta) = \text{hav}(\psi_2 - \psi_1) + \cos(\psi_1)\cos(\psi_2)\text{hav}(\lambda_2 - \lambda_1)$$

여기서 ψ_1 와 ψ_2 는 두 점의 경도, 그리고 λ_1 와 λ_2 는 두 점의 위도이다. 따라서 다음 수식을 이용하여 d 를 계산할 수 있다.

$$d = 2r \arcsin(\sqrt{\text{hav}(\Theta) = \text{hav}(\psi_2 - \psi_1) + \cos(\psi_1)\cos(\psi_2)\text{hav}(\lambda_2 - \lambda_1)})$$

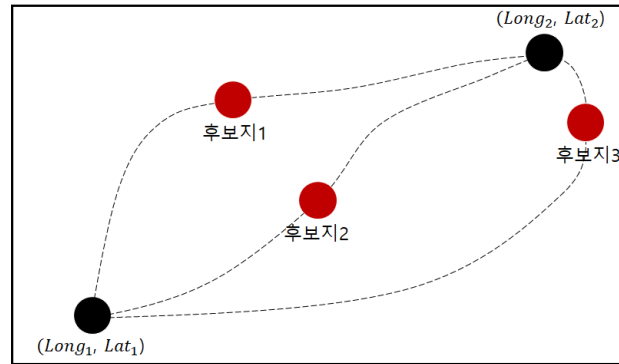
위 수식을 통하여 지구가 둥글다는 가정 하의 두 지점 간 최단거리를 측정한 후, 지하철의 평균표정속도는 34km 이다. [1] 을 참고한 역간 소요시간을 계산하였다.

2. 공공데이터를 통하여 제공되는 환승역, 환승거리 및 소요시간 정보(서울교통공사 17.10 기준)를 참고 하여 환승 시 소요시간을 반영하였다. 하지만 모든 정보가 제공되지는 않기에 미입력 환승정보의 경우 이세 중. 환승 소요시간이 평균 2분 21초임을 고려해... [2] 을 참고하여 평균값을 입력하였다. 또한 보통의 경우 환승 후 지하철을 바로 탑승하지 못하고 다음 열차를 기다리기 위하여 대기하는 시간이 추가적으로 소요된다. 따라서 환승 후 대기시간으로 2분을 추가 부여하였다.
3. 마지막으로 지하철 이동 시, 이동 역당 정차시간 30초를 추가적으로 반영하였다.

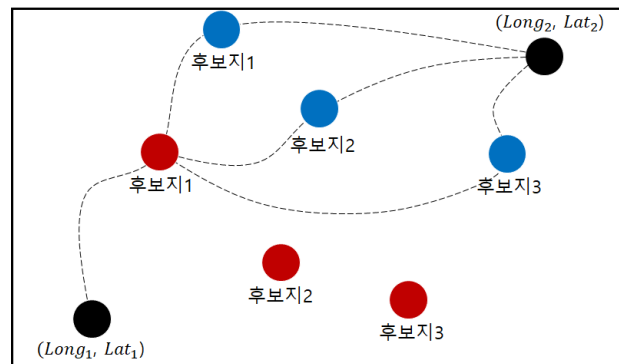
2.3 함수 제작

아이디어

1. 출발역과 도착역이 같은 노선이 아닌 경우, 두 지점을 기준으로 공간을 제약한다. 이때 환승을 위하여 우회하는 경우를 고려하기 위하여 위경도를 기준으로 두 지점이 포함된 조금 더 넓은 공간으로 제약한다.
2. 환승을 한번 하는 경우, 제약한 공간 내 포함된 출발역에서 도착역으로 환승이 가능한 지하철역을 선택하여 도착역으로 이동경로를 정의한다.



3. 환승을 두번 하는 경우, 제약한 공간 내 포함된 출발역과 노선이 같은 환승역을 선택하여 첫번째 환승 후보지로 정의한다. 다음으로 첫번째 환승 후보지와 도착역을 중심으로 공간을 제약하여 동일한 방식으로 이동경로를 정의한다.



4. 위와 같은 방식으로 함수를 정의한 후, 알고리즘의 효율성을 높이기 위하여 다음과 같은 기준으로 경로를 선정하였다.
 - 출발역과 도착역의 노선이 공통되지 않은 경우, 처음 2번까지의 환승경로를 계산한다.

이는, 공간을 제약하였으므로, 3번이상의 환승을 하는 경우가 2번의 환승을 하는 경우보다 가까운 경우를 어느 정도 고려한 결과이므로 2번까지의 환승을 우선적으로 고려하는 것이 문제가 되지않는다 판단하여 진행하였다.

- 만약 경로가 나오지 않는다면, 순차적으로 3번, 4번의 환승을 고려한 이동경로를 구한다.

2.4 최단경로 함수 예시

- 앞에서 정의한 기준을 통하여 아래와 같은 결과를 제공하는 최단경로 함수를 제작하였다.

```
seoulsubway::shortestpath(depart = "종로5가", arrival = "혜화")

## $Info
##   Depart Line Count Time Arrive
## 1 종로5가    1      1 1.07 동대문
## 2 동대문    4      1 1.38  혜화
##
## $Count
## [1] 2
##
## $Time
## [1] 8.15
##
## $Path1
##   Code   Name Line ExCode   lat   long Transfer   Dist Time
## 30  154 종로5가    1    129 37.57093 127.0018      0 698.84 1.23
## 29  155 동대문    1    128 37.57142 127.0097      4 604.84 1.07
##
## $Path2
##   Code   Name Line ExCode   lat   long Transfer   Dist Time
## 13  421 동대문    4    421 37.57142 127.0097      1 1400.90 2.48
## 12  420 혜화      4    420 37.58234 127.0018      0  783.28 1.38
```

참고

- [1] : 정은혜. 서울 지하철...“1호선이 가장 느리다.”
- [2] : 이세중. 환승 소요시간이 평균 2분 21초임을 고려해...