

# Introduction to seoulsubway

*KM Son*

*Department of Statistics, SKKU*

*April 21, 2019*

## 1. 현황 및 분석목적

### 1.1 현황

1. 지하철 관련 데이터로써 공공데이터 포털 내 시간대별 승하차 인원 통계만 제공된다.
  - 통계량으로 정보가 제공되므로 분석이 제한적임.
2. 서울특별시 대중교통 환승 버스 지하철 환승경로 조회정보를 통해 최단경로 OPEN API가 제공되지만 다음과 같은 문제가 존재한다.
  - 가까운 위치의 두 역을 동일한 역으로 인식하는 문제. ex) 을지로입구역과 종각역
  - 분기점 또는 지선의 경우 경로가 제공되지 않음.
  - 서울시 교통정보 시스템을 기반으로 하여, 버스/지하철이 결합된 알고리즘 구조이다. 따라서 지하철로 한정된 경로산출 시 문제가 존재함.

### 1.2 해결방안

수도권 지하철 네트워크를 구축하여 환승시간, 지하철 위치정보 등 공공데이터를 활용한 지하철 최단경로 알고리즘을 제작한다.

### 1.3 분석 전 가정수립

모든 지하철 이용객들은 가장 시간이 적게 걸리는 이동 경로를 통하여 이동한다. 라는 가정 하에 최단경로를 정의한다.

### 1.4 분석목적

1. 이용객 개별 승하차데이터를 이용하여 수도권 내 인구흐름을 추정한다.
2. 석촌호수에서 열리는 벚꽃축제를 중심으로 이벤트 발생에 따른 인구흐름의 차이가 존재하는지 확인한다.

## 2. 최단경로 함수

### 2.1 지하철 네트워크 구축

첫번째로, 역별 위치정보를 이용하여 1-9호선, 경의중앙선, 분당선 등이 포함된 수도권 지하철 네트워크 구축하였다. 이 과정에서 지선 또는 분기점의 경우 독립된 노선으로 고려하였다. 이를 통하여 총 22개의 수도권 지하철 노선이 포함된 네트워크를 구축하였고 리스트형식의 데이터 구조를 정의하였다. 이때, 환승변수인 **Transfer**를 통하여 다른 노선으로의 환승을 고려하였다. 아래 Table 1은 지하철 6호선 네트워크 구조의 예시이다. 특히, 디지털미디어시티 역과 같이 2개 이상의 노선과 환승이 가능한 경우를 고려하기 위하여 환승변수에 구분자 “|”를 두었다.

Table 1: 지하철 네트워크 데이터 구조 예시 (6호선)

Code	Name	Line	ExCode	lat	long	Transfer	Dist	Time
2611	응암	6	610	37.59860	126.9156	6_A	0.00	0.00
2617	새절	6	616	37.59115	126.9136	0	847.71	1.50
2618	증산	6	617	37.58388	126.9096	0	882.51	1.56
2619	디지털미디어시티	6	618	37.57665	126.9010	A K	1109.77	1.96
2620	월드컵경기장	6	619	37.56953	126.8993	0	805.78	1.42
2621	마포구청	6	620	37.56352	126.9033	0	758.97	1.34

## 2.2 최소 소요시간

1. Haversine 수식을 이용하여 역 간 소요시간을 계산한다.

우선,  $\Theta = \frac{d}{r}$  을 정의한다. 여기서  $d$  두 지점 간 거리이며,  $r$ 은 지구 반지름이다. 지구의 반지름은 알려진 값으로 Figure 1과 Figure 2를 이용하여  $d$ 를 계산한다.

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

Figure 1: fomula of Haversine

여기서  $\psi_1$  와  $\psi_2$ 는 두 점의 경도,  $\lambda_1$  와  $\lambda_2$  는 두 점의 위도이다.

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

Figure 2: distance with Haversine formula

이를 통하여 두 지점 간 최단거리를 측정한 후, 지하철의 평균표정속도는 34km이다. [1]을 참고한 역간 소요시간을 계산하였다.

2. 공공데이터 포털 내 제공되는 환승역, 환승거리 및 소요시간 정보(서울교통공사 17.10 기준)를 참고하여 환승 시 소요시간을 반영하였다. 하지만 모든 환승역에 대한 정보가 제공되지는 않기에 제공되지 않는 환승역에 대하여 이세중. 환승 소요시간이 평균 2분 21초임을 고려해... [2]을 참고하였다. 또한 일반적인 경우 환승 시 지하철 대기시간이 추가적으로 소요된다. 따라서 대기시간으로 2분을 추가하였다.
3. 지하철 정차시간을 고려하기 위하여 이동 역당 30초를 추가하였다.

## 2.3 함수 제작

### 아이디어

1. 출발역과 도착역의 노선이 다른 경우, 두 지점을 기준으로 공간을 제약한다. 이때 환승을 위하여 우회하는 경우를 고려해야 한다. 따라서 위경도를 기준으로 두 지점이 포함된 조금 더 넓은 공간으로 제약한다.
2. 환승을 한번 하는 경우, Figure 3와 같이, 공간 내 포함된 환승역을 선택하여 도착역으로의 이동경로를 정의한다.

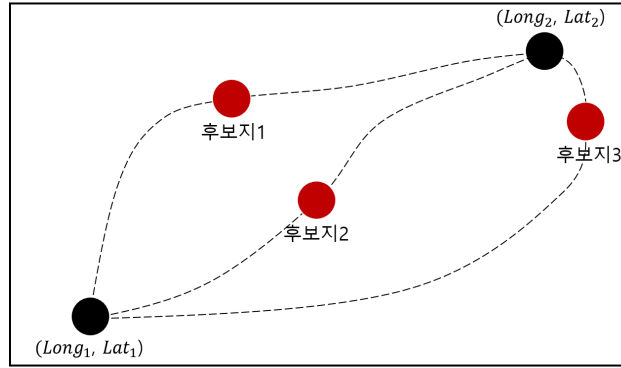


Figure 3: 공간제약기법(한번 환승)

3. 환승을 두번 하는 경우 Figure 4와 같이, 출발역과 노선이 같은 환승역을 선택하여 첫번째 환승 후보지로 정의한다. 다음으로 첫번째 환승 후보지와 도착역을 중심으로 동일한 방식을 적용한다.

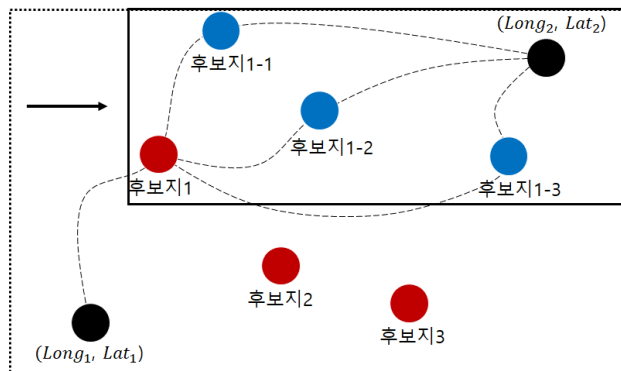


Figure 4: 공간제약기법(두번 환승)

4. 또한 알고리즘의 효율성을 높이기 위하여 다음과 같은 기준으로 최단 경로를 선정하였다.
- 출발역과 도착역의 노선이 공통되지 않은 경우, 처음 2번까지의 환승경로를 계산한다. 이는 공간을 제약한 상태이므로 처음 2번의 환승경로가 불필요하게 우회하는 경우는 제외한다.
  - 앞 단계에서 경로가 정의되지 않으면, 순차적으로 3번, 4번의 환승을 고려한 이동경로를 정의한다.

## 2.4 최단경로 함수 예시

- 서울역에서 혜화역으로 이동 시 최단경로 함수 예시

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

```
## $Info
##   Depart Line Count Time Arrive
## 1   서울     4     6 9.98   혜화
##
## $Count
## [1] 6
##
## $Time
## [1] 9.98
##
```

```

## $Path
##      Code      Name Line ExCode      lat      long Transfer      Dist
## 18  426      서울      4    426 37.55465 126.9726    1|A|K2  661.12
## 17  425      회현      4    425 37.55851 126.9782      0  764.32
## 16  424      명동      4    424 37.56099 126.9863      0  702.55
## 15  423      충무로     4    423 37.56124 126.9943      3 1277.33
## 14  422 동대문역사문화공원  4    422 37.56514 127.0079    2|5  718.09
## 13  421      동대문     4    421 37.57142 127.0097      1 1400.90
## 12  420      혜화      4    420 37.58234 127.0018      0  783.28
##      Time
## 18 1.17
## 17 1.35
## 16 1.24
## 15 2.26
## 14 1.27
## 13 2.48
## 12 1.38

```

## 참고

[1] : 정은혜. 서울 지하철...“1호선이 가장 느리다.”

[2] : 이세중. 환승 소요시간이 평균 2분 21초임을 고려해...