# Introduction to seoulsubway

*KM Son, JA Bang*
*Department of Statistics, SKKU*

*April 30, 2019*

## 0. Contact.

- KM_Son email : kinh4k@@g.skku.edu
- JA_Bang email : juna0033@@g.skku.edu

## 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 한정한 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 1`은 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 1`과 `Figure 2`를 이용하여 `d`를 계산한다.

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

Figure 1: fomula of Haversine

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

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

Figure 2: distance with Haversine formula

이를 통하여 두 지점 간 최단거리를 측정한 후, 지하철의 평균표정속도는 `34km`이다. [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 3`와 같이, 공간 내 포함된 환승역을 선택하여 도착역으로의 이동경로를 정의한다.

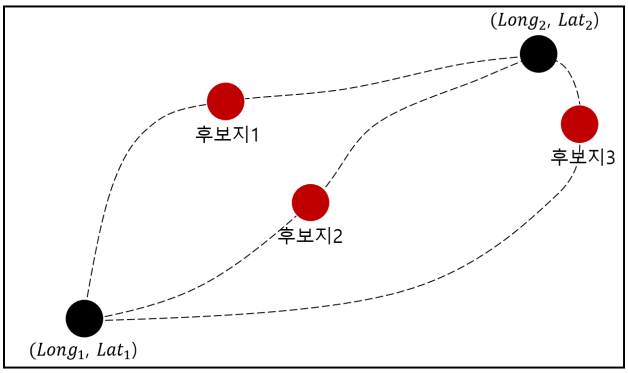


Figure 3: 공간제약기법(한번 환승)

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

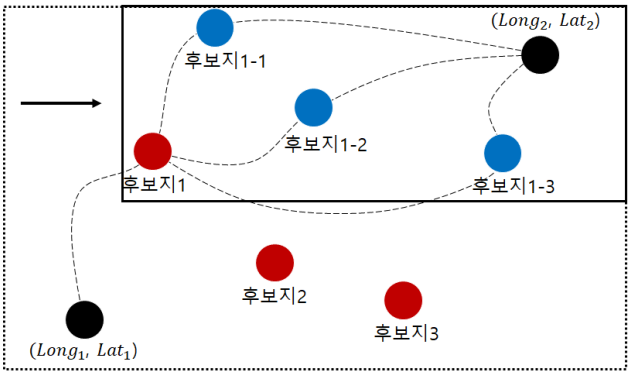


Figure 4: 공간제약기법(두번 환승)

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

- 종로3가역에서 혜화역으로의 최단경로 함수 결과

```
shortestpath(depart = "종로3가", arrival = "혜화")
```

```
## $Info
##    Depart Line Count Time Arrive
## 1 종로3가    1     2  2.3 동대문
## 2  동대문    4     1 1.38   혜화
```

```
##
## $Count
## [1] 3
##
## $Time
## [1] 9.38
##
## $Path1
##    Code     Name Line ExCode      lat     long Transfer   Dist Time
## 1:  153 종로3가     1     130 37.57161 126.9918      3|5 889.34 1.57
## 2:  154 종로5가     1     129 37.57093 127.0018        0 698.84 1.23
## 3:  155  동대문     1     128 37.57142 127.0097        4 604.84 1.07
##
## $Path2
##    Code   Name Line ExCode      lat     long Transfer    Dist Time
## 1:  421 동대문     4     421 37.57142 127.0097        1 1400.90 2.48
## 2:  420   혜화     4     420 37.58234 127.0018        0  783.28 1.38
```

최단경로 함수의 결과는 크게 Info, Count, Time, 그리고 Path로 구성된다. Info는 이동경로에 대한 요약 정보이다. Count와 Time은 경유역 수와 소요시간에 대한 정보이며, Path는 경유하는 지하철 역에 대한 정보를 제공한다.

- 종로3가역에서 혜화역으로의 최단경로 시각화


Figure 5: Shortest Path from Jongno 3-ga to Hyehwa on Seoul Map

위 최단경로 함수 결과 중 Path를 지도 위 시각화하여 이동 경로 중 경유하는 역들과 환승 역을 확인할 수 있다.

## 3. 활용 데이터 소개

### 3.1 이용객 개별 승하차 데이터 소개

- 데이터는 서울교통공사을 통해 제공받음.

2018년 벚꽃축제는 2018년 4월 5일부터 13일까지 총 8일동안 진행되었다. 따라서 2018년 4월 7일의 이용객 개별 승하차 데이터를 이용하였다. 데이터의 구조는 다음과 같다.

Table 2: 2019년 04년 07일의 이용객 개별 승하차 데이터 예시

| ID | 운행일자 | 요일 | 승차호선명 | 승차역ID | 승차역명 | 하차호선명 | 하차역ID | 하차역명 | 카드구분 | 사용자구분 | 시간대 | 승객수 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| #1 | 20180407 | 토 | 3 | 323 | 약수 | 1 | 150 | 서울 | 후불 | 일반 | 12 | 1 |
| #2 | 20180407 | 토 | 2 | 202 | 을지로입구 | 4 | 420 | 혜화 | 선불 | 일반 | 21 | 1 |
| #3 | 20180407 | 토 | 2 | 202 | 을지로입구 | 2 | 241 | 이대 | 후불 | 일반 | 19 | 1 |
| #4 | 20180407 | 토 | 5 | 2547 | 광나루 | 5 | 2550 | 길동 | 선불 | 경로 | 19 | 1 |
| #5 | 20180407 | 토 | 2 | 202 | 을지로입구 | 2 | 231 | 신대방 | 선불 | 일반 | 20 | 1 |
| #6 | 20180407 | 토 | 2 | 225 | 방배 | 7 | 2729 | 건대입구 | 후불 | 일반 | 21 | 1 |

분석 시 ID, 운행일자, 요일, 승차호선명, 승차역명, 하차호선명, 하차역명, 시간대 그리고 승객수 변수만을 고려하며, 시간대는 하차 시간대를 의미한다. 이용객 개별 승하차 데이터는 1-8호선과 241개의 지하철 역으로 구성되며, 1호선의 경우는 서울교통공사 소속인 서울 - 청량리 구간만 포함한다.

### 3.2 시각화를 위한 데이터셋 정의

출발역 -> 도착역의 한정된 정보만 제공되는 데이터를 이용하여 분석목적에 적합한 형태로 재정의하는 과정이 필요하다. 최종적으로 사용할 데이터셋의 형태는 Table 3과 같으며, 시간대별 구간의 혼잡도와 위치정보를 포함한다.

Table 3: 최종 데이터셋 예시 (07시)

| time | from | from_lat | from_long | to | to_lat | to_long | sum_count |
|---|---|---|---|---|---|---|---|
| 07 | 가락시장 | 37.49 | 127.12 | 경찰병원 | 37.50 | 127.12 | 959 |
| 07 | 가락시장 | 37.49 | 127.12 | 문정 | 37.49 | 127.12 | 1149 |
| 07 | 가락시장 | 37.49 | 127.12 | 송파 | 37.50 | 127.11 | 1630 |
| 07 | 가락시장 | 37.49 | 127.12 | 수서 | 37.49 | 127.10 | 1253 |
| 07 | 가산디지털단지 | 37.48 | 126.88 | 구로 | 37.50 | 126.88 | 118 |
| 07 | 가산디지털단지 | 37.48 | 126.88 | 남구로 | 37.49 | 126.89 | 3309 |

다음은 위와 같은 형태의 데이터셋을 구축하기 위한 과정이다.

#### 3.2.1 최단경로 데이터베이스 제작

우선 이동 간 경유하는 모든 구간에 대한 정보를 포함한 새로운 데이터 구조를 정의한다. 예를 들어서 종로3가역에서 출발하여 혜화역에 도착하는 경우, 종로3가-종로5가, 종로5가-동대문 그리고 동대문에서 환승 후, 동대문-혜화로 3개의 구간을 경유할 것이다. 이와 같은 방식으로 모든 지하철 경로의 이동 정보를 포함한 데이터베이스를 제작하였다. 또한 시각화를 위하여 데이터베이스 내 지하철 역 위치정보를 포함하였다.

Table 4: 최단경로 데이터베이스 예시 (종로3가역->혜화역)

| set | from | from_lat | from_long | to | to_lat | to_long |
|---|---|---|---|---|---|---|
| 종로3가-혜화 | 종로3가 | 37.57 | 126.99 | 종로5가 | 37.57 | 127.00 |
| 종로3가-혜화 | 종로5가 | 37.57 | 127.00 | 동대문 | 37.57 | 127.01 |
| 종로3가-혜화 | 동대문 | 37.57 | 127.01 | 혜화 | 37.58 | 127.00 |

### 3.2.2 경로별 승하차 인원에 대한 데이터베이스

아래 Table 5은 이용객 개별 승하차 데이터를 이용한 모든 승차역-하차역 경로에 대한 이용객 수의 합계이다.

Table 5: 시간대별 경로 데이터베이스 예시 (종로3가역->혜화역)

| date | day | time | set | count |
|---|---|---|---|---|
| 20180407 | Sat | 07 | 종로3가-혜화 | 1 |
| 20180407 | Sat | 09 | 종로3가-혜화 | 4 |
| 20180407 | Sat | 10 | 종로3가-혜화 | 4 |
| 20180407 | Sat | 11 | 종로3가-혜화 | 7 |
| 20180407 | Sat | 12 | 종로3가-혜화 | 10 |
| 20180407 | Sat | 13 | 종로3가-혜화 | 13 |

이를 set 변수를 기준으로 최단경로 데이터베이스와 결합하여 최종적으로 사용될 시간대별 구간의 합을 구하였다. 여기서 구간이란 이웃한 두 역 사이로 정의되며, 지하철 네트워크 내에는 총 748개의 구간이 존재한다.

### 참고

[1] : 정은혜. 서울 지하철... "1호선이 가장 느리다."

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