

이동경로 유사도 분석을 통한 매칭 서비스

:기초조사서

컴퓨터공학과/ 2017103950/ 고병후

요 약

스마트 폰은 대다수의 현대인들의 일상에서 늘 함께한다. 이런 Mobile Device는 그 특성상 여러 사용자 정보를 수집할 수 있다. 본 프로젝트는 Android운영체제를 갖는 Mobile Device를 대상으로 사용자의 여러 데이터를 수집하여 유저의 이동경로 패턴을 파악하는 LBS(Location Based Service)를 구현한다. 대부분의 사람들은 출퇴근, 등하교 등 일상적이고 반복적인 이동패턴을 갖는다. 이러한 패턴을 스마트폰을 통해 수집한 gps좌표, 가속도 벡터, wi-fi접속 기록 등의 데이터를 이용하여 분석하고 개인별로 일반화한다. 이를 통해 이동패턴, 이동수단(도보, 버스 등), 평균 이동시각을 분석해 유사한 패턴을 보이는 타인을 매칭하여 추천한다.

1. 서론

1.1. 연구배경

최근 스마트폰은 현대인의 삶에서 필수적인 도구가 되었다. 일상에서 늘 함께하는 Mobile Device는 여러 사용자 data를 남기며 이를 분석하면 유의미한 결과를 확보할 수 있다. 이렇게 Mobile Device를 통해 수집한 정보를 바탕으로 한 여러 서비스들이 개발되어 제공되고 있는데 본 프로젝트는 특히 위치정보와 관련된 서비스인 LBS(Location Based Service)에 집중한다.

기존에 사용자 위치기반 서비스는 실시간으로 방향과 위치좌표를 파악하여 여러 서비스를 제공하는 형식을 취하고는 했다. 하지만 사용자의 위치 및 이동의 특성을 나타내는 데이터는 위치와 방향 말고도 다양하게 존재한다. 예를 들어 와이파이의 연결 여부는 사용자가 해당 위치에 정지하여 시간을 보냈을 가능성을 시사하고 가속도 벡터의 상하방향 반복적인 운동은 사용자가 이동시에 도보를 이용했는지, 지하철과 버스 등 대중교통을 이용했는지를 알려준다.

본 프로젝트는 이러한 위치, 방향, 가속도 벡터, wi-fi연결 시간 등의 여러 데이터를 지속적으로 수집하여 이를 가공하고 분석하여 유의미한 정보인 '개인의 이동패턴'을 도출하는 것을 목적으로 한다. 다수의 사람들은 일상속에서 출퇴근, 등하교와 같은 반복적이고 일반화할 수 있는 이동 패턴을 갖기에 위에서 언급한 데이터들을 통해 개인별로 대표적인 이동패턴을 도출할 수 있다. 또한 각자의 이동패턴을 도출한다면 이것이 유사한 유저들을 상호 추천하여 동행이 가능한 서로를

인지할 수 있다. 이러한 이동패턴이 유사한 유저들간 연결은 car sharing, 택시비 절감 등의 효과로 이어질 수 있다.

1.2. 연구목표

스마트폰을 이용해 gps, 방향, 흔들림(가속도벡터), wi-fi연결여부 등의 데이터를 지속적으로 수집하고 저장한다. 이를 중앙에서 처리하여 사용자별 대표 이동패턴을 도출하고 해당 패턴이 높은 유사도를 보이는 유저를 상호 추천하는 서비스를 구현한다.

스마트폰이 일상에서 유저의 여러 행동 및 상황을 반영한 데이터를 보유하기에 이를 수집, 가공, 분석하여 사용자의 이동패턴을 파악한다. 이동패턴이 유사한 유저를 매칭하여 동행이 가능하게 해 이동에 있어 car sharing, 택시비절감 등 상호 도움이 되도록 함을 목표로 한다. 구체적인 목표는 아래와 같다.

첫번째, Mobile Device의 back ground service로 필요한 여러 데이터들을 지속적으로 수집하여 중앙 server에 저장한다. 이를 가공, 정제하고 처리하여 유의미한 정보인 이동패턴을 도출한다. 이 패턴을 바탕으로 이동경로, 이동시간, 이동수단의 유사도를 통해 비슷한 특성을 가진 유저를 상호 추천한다.

두번째, 사용자에게 서비스에서 파악한 이동경로 패턴을 GUI를 통해 게시한다.

세번째, 서버단에서 사용자의 실제 이동경로들과 이를 일반화한 대표이동경로를 지도위에서 point와 이를 연결한 line형식으로 출력이 가능하게 한다. 또한 사용자별 유사도 도출 근거를 수치화해서 그래프로 표기 가능하게 한다.

2. 관련연구

2.1. Android & Android studio

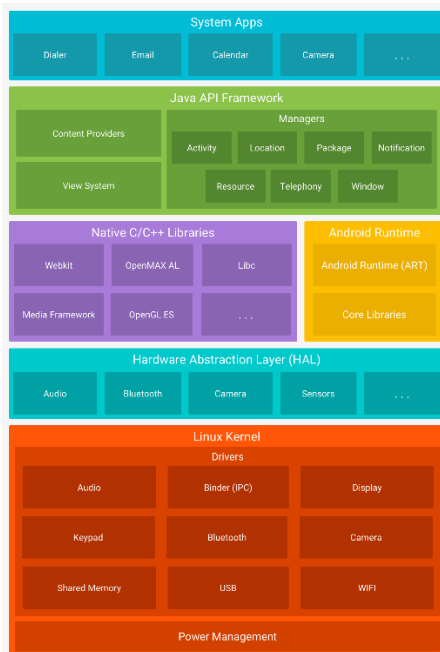


그림 1. Android software stack

해당 서비스를 구현하기 위해 사용자 데이터를 수집할 Mobile Device 기기가 필요한데 본 프로젝트에서는 이를 Android 운영체제를 사용중인 스마트 폰으로 한정한다.

Android란 리눅스 커널을 이용한 운영체제이다. 주로 Mobile Device를 대상으로 한다. VM(virtual machine)인 Android RunTime을 통해 application을 별도의 process에서 수행하도록 한다. Android는 현재 다수의 스마트폰에서 사용되고 있다. 운영체제의 아키텍처는 그림1과 같고 각 layer의 역할은 아래와 같다.

1) Linux kernel

Android platform의 기반이다. 스레딩 및 하위 메모리 관리 등 기본기능에 사용된다. Hardware 추상화, 메모리 관리, 드라이버 관리 등을 수행한다.

2) Hardware Abstraction Layer(HAL)

Hardware 추상화 계층으로 각 Hardware의 표준 인터페이스를 정의하여 하위 드라이버 구현을 고려하지 않아도 되게 한다.

3) Native Libraries & Android Runtime

저급 수준의 일반적인 library가 위치한다. 컴퓨팅 파워가 약한 Mobile Device 특성상 native 코드로 구현되어 컴파일 되어있다.

Runtime는 VM으로 컴파일된 application을 os에 맞추어진 형태로 바꾼다.

4) Java API Framework

Android Runtime을 추상화한 계층이다. 개발된 application이 하위 계층과 소통할 수 있도록 지원한다.

5) System Apps (Application)

실제로 개발된 application이다.

이러한 Android os위에서의 클라이언트 앱 구현을 위해 Android Studio를 사용한다.

Android Studio란 Android application개발을 위한 공식IDE(Integrated Development Environment)이다. IntelliJ IDEA를 이용한 코드 편집기를 사용하고 있다. 프로젝트 작성시에 빌드파일은 Gradle Scripts에 표기되며 각 모듈에는 manifest, java, res가 포함된다. 빌드를 위해 Gradle 빌드 시스템을 사용한다. 또한 여러 SDK(Software Development Kit)와 편리한 API를 지원하여 개발이 용이하고 쉽다. 또한 시뮬레이터를 이용해 가상의 스마트폰에서 실험이 가능하다. 이러한 이유로 본 프로젝트에서는 클라이언트 앱 개발을 위해 이를 사용한다.

2.2. GPS

GPS(Global Positioning System)는 위성항법시스템이다. 위성항법시스템은 위성을 통해 위치(Positioning) 및 항법(Navigation), 시각(Timing) 정보(PNT)를 제공해주는 시스템이다.

GPS는 미 국방성 (US Department of Defense) 에서 개발하여 운용중인 인공위성으로부터 수신한 Radio 신호를 이용하여 정지 및 운동하고 있는 물체의 위치를 결정하는 시스템이다. gps에 사용되는 인공위성은 연속적으로 신호를 송신하며 항상 4개의 인공위성이 어디에서도 보일 수 있도록 디자인되어 있다. 즉, gps란 위성에서 발사한 전파를 수신하여 관측점까지 소요 시간을 계산함으로써 관측점까지의 위치를 구하는 time navigation system이다.

위성항법시스템은 군사적 목적은 물론 스마트폰에 탑재될 정도로 작은 수신기만 있으면 사용자의 지리적 위치와 상관없이 신호를 이용할 수 있다. 또한 이동 중에도 실시간 사용이 가능하여 자동차, 선박 등의 길 안내뿐만 아니라 응급구조, 재해 예측 등 활용범위가 매우 넓다.

단, 실내에서는 오차범위가 커지며 기본적으로 약 10m의 오차범위를 감수해야 한다는 한계점이 있다. 본 연구에서는 사용자의 위치 값을 측정하기 위해 gps좌표를 이용한다.

2.3. 가속도센서

Android platform은 기기의 동작을 모니터링 할 수 있는 센서를 다수 지원하는데 본 연구에서는 동작센서를 사용한다. 그 중에서도 흔들림을 감지하기 위해 선형가속 센서를 이용한다.

동작 센서는 흔들기, 회전, 스윙등 기기에 가해지는 운동을 측정할 수 있다. 기기에 가해지는 동작은 사용자의 의도적 입력(스마트폰을 기우려 화면을 가로로 맞추는 등) 또는 기기가 있는 물리적 환경(기기를 가진 사용자가 자동차를 타고 이동하는 등)에 의해 생성된다.

Android studio에서 동작 센서는 각 SensorEvent에 대해 센서 값의 다차원적 배열을 반환한다. 예를 들어 단일 센서 이벤트에서 가속도계는 세 개의 좌표축에 대한 가속력 데이터를 반환하고 자이로스코프는 세 개의 좌표축에 대한 회전 속도 데이터를 반환한다. 이런 데이터 값은 다른 SensorEvent 매개변수와 함께 float 배열로 반환된다.

해당 프로젝트에서는 관련 Android studio API인 TYPE_ACCELEROMETER를 이용하여 x, y, z축의 가속도를 측정한다. 이를 통해 기기에 가해진 힘(흔들림)을 측정 가능하다.

2.4. Douglas-Peucker algorithm

다각형 근사에 사용되는 벡터 단순화 알고리즘이다. 객체의 경계선을 추출할 때 원래의 모습과 유사한 다각형으로 간략화가 가능하다. epsilon이라는 임의의 값을 설정하여 algorithm을 진행한다.

N개의 점과 그 점을 잇는 선분으로 구성된 도형이 있다면 시작 점과 끝점을 잇는다. 이 선분을 기준으로 가장 거리가 먼 점을 찾아 일정거리 이상의 값을 갖는다면 꼭짓점으로 추가해 선분을 분할한다. 이 과정을 분할된 각 선분에 재귀적으로 수행한다. 재귀의 깊이에 따라 정밀도가 달라진다. 선분에서 가장 멀리 떨어진 지점이 기준거리 이내라면 분할을 멈춘다. 단 다각형과 같이 닫힌 도형의 경우는 시작점과 끝점을 동일하게 두어 수행한다. 이 과정이 끝나면 다각형은 원본과 근사하지만 단순화된 도형으로 출력된다.

2.5. Django

Django는 web application framework로 web service를 효율적으로 빠르게 개발하도록 해주는 software다. web application개발에 공통적으로 구현되어야 하는 인증, database의 정보 검색, cookie 관리 등을 쉽게 개발하도록 지원한다. 즉, Django는 다양한 기능을 web application framework라고 하는 재사용 가능한 모듈의 대규모 컬렉션으로 그룹화하여 작업을 더 쉽게 수행할 수 있도록 한다. 개발자는 Django를 사용하여 코드를 보다 효율적으로 구성하고 작성하며 개발 시간을 크게 단축할 수 있다. 특히 양식 처리, 사용자 인증, 관리자 페이지에 대한 편리한 지원이 많다.

MVC(Model-View-Controller)의 디자인 패턴을 지향한다. ORM(Object Relational Mapping)을 지원하여 SQL에 대한 사전지식이 부족하더라도 database를 다룰 수 있다.

위에서 언급한 이유들과 같이 prototype을 단시간 내에 개발하기에 유리하고 Python을 사용하기에 여러 library를 사용할 수 있다는 장점 때문에 본 프로젝트는 Django를 이용하여 구현한다.

3. 프로젝트 내용

3.1. 시나리오

본 프로젝트는 유저의 Mobile Device에서 data를 수집하고 이를 server로 전송하고 중앙에서 처리하여 결과를 유저의 기기로 전송하는 형태이다. 그렇기에 Mobile단과 server단이 필요하다.

각 단에서의 동작시나리오를 아래와 같이 기술한다.

3.1.1. Mobile app(User)단 시나리오

1) 회원가입 및 로그인

위치정보를 가진 사용자를 분류하고 인지하기 위해 회원가입과 로그인을 요구한다. 각 사용자는 고유의 id를 갖고 인증은 jwt(Json web token)을 통해 수행한다.

2) 스마트폰의 센서를 통한 유저 흔적(data) 수집

모바일 단에서 background program으로 지속적이고 규칙적으로 data를 수집한다. 수집되는 data의 종류와 수집 이벤트 발생 조건은 아래의 기술과 같다.

①GPS: 정해진 시간당 저장

②방향: GPS와 동시수집

③가속벡터: 특정수치 이상의 값일 때 추출 혹은 gps와 동시 수집

④Wi-fi 연결여부: 이벤트 발생시 수집

해당 data들을 수집 시간과 함께 모바일 단에 저장해 둔다.

3) 일정시간이 지나면 서버로 저장된 데이터 전송

일정 시간이 지나거나 특정 용량에 도달하면 server로 데이터들을 전송한다. HTTP를 이용한 통신

을 통해 구현한다.

4) 서버로부터 이동패턴이 유사한 동행인 추천

server가 해당 사용자의 대표이동패턴과 패턴의 유사도가 높은 유저를 추천하는 데이터를 전송하면 이를 수신하고 출력한다.

5) 대표 이동패턴을 지도로 출력 가능

3.1.2. Server단 시나리오

1) 유저로부터 데이터 수집

유저로부터 유저id, 시간, gps좌표, 방향, 가속도벡터, wi-fi연결여부의 데이터를 수신 받는다. 해당 데이터는 raw data로 처리 전까지 server에 쌓아 둔다.

2) 데이터 정제

raw data는 gps 오류, 의미가 적은 data등 많은 noise를 포함한다. 그렇기에 이러한 데이터들을 유의미한 data만 남기도록 정제한다. 해당 과정은 아래와 같이 filter를 통해 이루어진다.

①Baseline filter: gps값의 시간당 변화량이 특정 수치 이하일 때 이를 정지로 취급하여 대표 값만 놔두고 나머지 data를 제거한다.

②Velocity filter : GPS 좌표의 속도를 사용한 오류 검출을 한다.특정시간에 지나치게 빠른 속도 변화를 보이는 gps 값을 오류로 판단해 제거한다.

③Angle filter : GPS 좌표의 방향을 고려한 각도로 오류 검출한다. 인접한 시간에 지나치게 빠른 방향의 변화를 보이는 값을 제거한다.

3) 정보집합 생성

정제된 raw data를 조합하여 의미 있는 data인 combined data를 생성한다. 즉 이동경로 데이터를 생성한다. 이는 출발지, 목적지를 포함한 이동경로, 이동수단(보행, 대중교통)으로 구성된다.

4) 유저의 대표이동경로 파악

이동경로 data를 통해 해당 유저를 대표할 수 있는 이동경로를 도출한다. combined data로 complex data를 생성하는 과정이다. wi-fi연결이 된 시점의 위치, 일정시간동안 gps값의 유지 등을 이용해 정지된 위치를 파악한다. 이 정지된 위치를 바탕으로 사용자의 이동경로가 도출된다.

이때 꺾적의 단순화를 위해 Douglas-Peucker algorithm을 수행한다.

5) 대표이동패턴을 바탕으로 가장 유사한 유저를 추천

도출한 대표이동경로를 분석해 가장 유사도가 높은 타인을 찾아 매칭한다. 이로써 이동 패턴이 유사한 유저라는 meaning information이 생성된다.

유사도의 기준은 아래의 3가지 기준에 각각 가중치를 두어 계산한다.

①대표 이동패턴의 각 정지위치(출발지, 목적지)의 거리

②이동수단(보행, 자동차, 전철)의 일치,

③이동 시각의 유사함

6) 분석된 데이터를 지도api로 출력하여 시각화 가능

3.2. 아키텍처

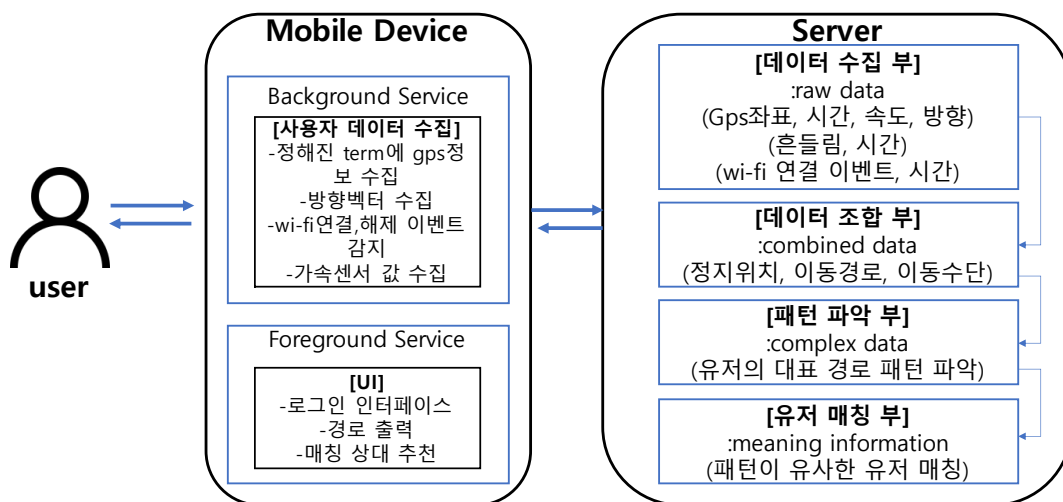


그림 2. Architecture overview

본 프로젝트의 전체적인 Architecture는 그림2와 같다. 유저단은 데이터를 수집하는 background service와 UI가 위치한 foreground service로 구성되며 server는 각 역할에 따른 모듈들로 구성된다.

3.3. 요구사항

3.3.1. Mobile app(User) 요구사항

- background에서 지속적으로 data를 수집하는 기능의 application이 필요하다.
- Mobile단에서 user와 그 data들을 관리할 RDBMS가 필요하다. 이곳에 데이터를 축적해 두었다가 특정 시간이 지나면 server로 이를 전송하고 모바일 단말기에서는 삭제한다.
- Server로부터 전달받은 User의 대표이동 경로를 지도에 표기할 수 있어야 한다. 이때 정지 위치 및 주요 동선지점을 pin으로 표시, 각 pin은 line으로 이어지는 구조로 표기되도록 한다.
- data수집시에 이상이 있거나 서버와의 접속이 원활히 진행되지 않는다면 유저, 서버단에 모두 이를 알릴 수 있는 push알림이 있어야 한다.

3.3.2. Server 요구사항

- float값으로 전송될 gps좌표계산을 컴퓨팅 파워에 따라 int로 변환해 연산한다.
- 간단한 data는 json으로 송수신 하지만 데이터의 크기에 따라 FTP를 고려해야 한다.
- 분석에 필요한 임계값 (gps값의 차이로 정지여부를 판단할 때 기준 등)들을 설정하기 위해 수집된 데이터를 matplotlib등을 통해 시각화해 확인해야 한다.
- 한 유저에 대해 여러 개의 이동 동선 데이터를 지도에 중첩하여 구분되게 표기해야 한다. 이를 위해 이동 동선당 다른 색을 부여한다.

4. 향후 일정

표 1.Schedule.

내용	3월			4월				5월					6월
	3w	4w	5w	1w	2w	3w	4w	1w	2w	3w	4w	5w	2w
기초조사서 작성													
데이터 수집 앱 개발													
데이터 수집 진행													
데이터 정제기능 개발													
중간보고서													
데이터 조합기능 개발													
유사한 유저 추천기능 개발													
최종보고서													
발표													

4월

-데이터 수집 application개발 및 수집대상 유저 5인에게 app배포.

-중간보고서 작성

5월

-서버단에서 데이터 정제, 데이터 분석, 추천알고리즘 구현

-추천 동행인과 예측 경로를 유저 인터페이스에서 출력

-최종 보고서 작성

6월

-발표 및 시연 준비

5. 결론 및 기대효과

본 프로젝트는 사용자가 일상속에서 갖는 반복적이고 일반적인 이동경로를 Mobile device를 통해 수집한 data를 이용해 파악하고 해당 패턴의 유사성을 보이는 타인을 매칭한다. 이 서비스를 통해 사용자는 본인과 동행이 가능한 사람을 찾을 수 있다. 이러한 매칭이 실제 동행으로 이어진다면 car sharing, 택시비절감 등의 이득을 얻을 수 있을 것이다.

또한 본 프로젝트는 확장성을 고려하여 각 기능별로 분리되어 설계되어 있다. 예를 들면 현재 유저의 이동수단을 파악함에 있어 가속도 벡터의 값을 정해진 상수로 크기를 비교하여 도보인지 교통수단에 탑승했는지를 파악한다. 또 유사한 이동경로 패턴을 갖는 유저를 탐색할 때 분기문을 이용하여 판단하도록 설계되어 있다. 이러한 기능들은 향후 보다 많은 데이터가 축적되면 deep learning algorithm을 적용하여 정확도를 높일 수 있다.

해당 서비스는 또한 실제 사용가능 성에 있어 유망하다. 현재 다수 존재하는 car sharing, 렌터카 사업 등에 이동경로가 비슷한 유저를 추천하는 본 시스템이 적용된다면 더욱 높은 효과를 기대할 수 있을 것이다.

6. 참고문헌

[1] Android 구조: <https://developer.android.com/guide/platform?hl=ko>

[2] Android studio 개요: <https://developer.android.com/studio/intro?hl=ko>

[3]GPS정의 (한국항공우주연구원): https://www.kari.re.kr/kor/sub03_08_02.do

[4]한진희 "GPS 수신기 설계 기술 및 동향"

[5]Android 동작센서: https://developer.android.com/guide/topics/sensors/sensors_motion?hl=ko

[6]Django 설명: <https://aws.amazon.com/ko/what-is/django/>

[7]양혜진 "모바일의 위치데이터를 활용한 사용자 조사 방법"

[8]김선용, 박범준, 정재진 "모바일장치의 GPS를 이용한 사용자 이동경로 분석 및 이동경로 추천 시스템"

[9] 양승국, 백혜정, 김제민, 전명중, 박영택 "개인 경로 학습을 위한 GPS 좌표 기반 이동 궤적 추출"