

## 실내 위치기반 서비스를 위한 사용자 관심지점 탐사 기법과 POI추천 시스템의 구현

김범수\*, 이연\*, 김경배\*\*, 배해영\*

### The Development of Users' Interesting Points Analyses Method and POI Recommendation System for Indoor Location Based Services

Beom-Su Kim\*, Yeon Lee\*, Gyeong-Bae Kim\*\*, Hae-Young Bae\*

#### 요약

최근 실내 위치기반서비스를 위한 다양한 측위 기술의 발전으로 실내에서도 사용자의 위치추정이 가능해 짐에 따라 다양한 형태의 실내 위치기반 서비스가 개발되고 있다. 이에 쇼핑몰이나 백화점 등의 대규모 상업 공간 같은 복잡한 실내 공간에서 사용자에게 가장 적합한 위치나 매장을 추천하는 개인화된 POI 추천 시스템의 개발이 필요하게 되었다. POI 추천을 위해서는 사용자의 이동성과 대규모 상업공간의 공간성을 고려한 사용자 관심지점 탐사 기법의 연구가 필요하다. 이에 본 논문에서는 실내 위치기반 서비스의 POI 추천 시스템의 구현과 사용자들의 이동 데이터로부터 다양한 관심지점을 고려하기 위해 사용자가 일정 시간 동안 머무른 지점을 Stay point라 정의하고 실내공간에서 Stay point를 탐색하는 알고리즘을 제안하였다. 또한 제안된 알고리즘을 이용하여 탐색한 Stay point로부터 방문패턴을 탐사하여 POI 추천 시스템을 구현하였다. 구현된 시스템은 사용자의 모든 이동 로그를 이용한 패턴탐사보다 데이터양을 획기적으로 줄임으로써 빠른 패턴탐사와 메모리 사용량을 줄일 수 있었다.

▶ Keyword : 실내 위치기반 서비스, POI 추천 시스템, Stay point

#### Abstract

Recently, as location-determination of indoor users is available with the development of variety of localization techniques for indoor location-based service, diverse indoor location based services

• 제1저자 : 김범수 • 교신저자 : 김경배 • 책임저자 : 배해영  
• 투고일 : 2012. 02. 23, 심사일 : 2012. 03. 14, 게재확정일 : 2012. 04. 12.  
\* 인하대학교 컴퓨터정보공학과(Dept. of Computer & Information Engineering, Inha University)  
\*\* 서원대학교 컴퓨터교육과(Dept. of Computer Education, Seowon University)

are proposed. Accordingly, it is necessary to develop individualized POI recommendation service for recommending most interested points of large-scale commercial spaces such as shopping malls and departments. For POI recommendation, it is necessary to study the method for exploring location which users are interested in location with considering user's mobility in large-scale commercial spaces. In this paper, we proposed POI recommendation system with the definition of users' as 'Stay point' in order to consider users' various interest locations. By using the proposed algorithm, we analysis users' Stay points, then mining the users' visiting pattern to finished the proposed. POI Recommendation System. The proposed system decreased data more dramatically than that of using user's entire mobility data and usage of memory.

▶ Keyword : Indoor location based service ,POI recomandation system, Stay point

## I. 서 론

그동안 GPS 기술의 발전과 휴대용 기기의 보급으로 다양한 위치기반 서비스가 발전되어 왔다. 최근에는 스마트폰 기술의 발전으로 개인 사용자가 언제 어디서나 인터넷에 연결할 수 있는 환경과 스마트폰에 탑재된 고성능 GPS를 통해 다양한 위치기반 서비스를 활용할 수 있게 되었다. 전통적인 위치기반 서비스들은 GPS 정보를 이용한 실외 위치기반 서비스가 대부분이었다. 그러나 최근 다양한 실내 측위 기술의 발전으로 실내에서도 사용자의 위치추적이 가능해 짐에 따라 다양한 형태의 실내 위치기반 서비스가 개발되고 있다. 이에 쇼핑몰이나 백화점 등의 대규모 상업 공간 같은 복잡한 실내 공간에서 사용자에게 가장 적합한 위치나 매장을 추천하는 개인화된 POI 추천 시스템의 개발이 필요하게 되었다.

POI 추천을 위해서는 비슷한 성향을 가진 사용자들의 관심지점을 찾아야 한다. 사용자들의 관심지점을 찾기 위해서는 사용자들의 구매 이력이나 이동 로그를 분석하여 빈번하게 발생하는 패턴을 찾아야 한다. 연구[1]은 오프라인 매장 추천을 위해 고객의 구매목록을 분석하여 사용자의 관심지점을 찾아 추천하고 있다. 그 밖에 연구[2]와 연구[3]은 상품추천을 위해 사용자의 구매 이력 및 프로파일을 사용하고 있다. 하지만 이런 사용자의 구매 이력을 분석하는 방법은 사용자가 관심 있게 방문 지점이나 매장이라 할지라도 실제 구매가 발생하지 않으면 기록이 남지 않아 관심지점 탐사에서 제외되는 문제점과 대규모 상업 공간의 공간성을 고려할 수 없는 문제점이 있다. 반면 사용자의 위치데이터를 이용한 패턴탐사 방법[4]은 사용자의 모든 방문지점을 고려할 수 있지만, 관심지점까지 이동하는데 거쳐 간 경로들과 실제 일정시간 머물지 않고 지

나친 지점까지 패턴에 탐사되는 문제점과 방대한 이동 로그를 분석해야 하는 문제점이 있다. 이러한 문제를 해결하고 사용자의 다양한 관심지점을 고려하기 위해서는 사용자가 최소 입계시간 동안 머무른 지점을 관심지점으로 고려할 필요성이 있다.

연구[5]에서는 사용자가 특정 입계시간 동안 머문 지점을 Stay point로 정의하고 이를 탐사하기 위해 사용자들의 모든 GPS 기록을 수집하여 사용자가 특정 건물 안으로 들어갔을 때 사라진 GPS 신호와 나왔을 때 다시 나타난 신호의 시간차를 이용하여 도착 시간과 떠난 시간을 구분하고 계산하여 Stay point를 탐색하고 있다.

하지만 이런 GPS 신호를 이용한 방법은 실내 환경에서 사용자가 특정한 지점에 도착한 시간과 떠난 시간을 구분 지을 수 없어 적용하기 어려운 문제점이 있다. 또한, 대규모 상업 공간 내에서 LBS를 이용하는 사용자들은 끊임없이 이동하는 특성을 가지고 있고 이런 사용자의 모든 이동기록을 저장하는 것은 LBS 서버에 많은 비용을 요구한다.

따라서 본 논문에서는 실내 공간에서 사용자의 다양한 관심지점을 탐색하기 위해 사용자가 입계시간 동안 머문 특정 지점을 Stay point로 정의하고 실내 공간에서 사용자의 Stay point 탐사를 위해 Current Table과 Update Table을 이용한 Stay point 탐사 방법을 제안한다. 제안한 방법은 Table 형태의 버퍼를 이용하여 사용자의 모든 이동데이터를 저장할 필요가 없으며 저장된 로그를 분석하는 기존의 방법과 달리 위치기반 서비스를 사용 중인 사용자의 Stay point를 실시간으로 탐사 가능하여 다양한 서비스에 적용할 수 있는 장점이 있다.

본 논문에서는 실내 환경에서 사용자의 Stay point 탐색 방법을 제안하고 제안된 알고리즘으로 수집된 사용자의 Stay point로부터 방문패턴을 탐사하여 실내 위치기반 서비스의

POI 추천 시스템을 구현하였다.

본 논문의 구성은 2장에서는 관련 연구로 실내의 검용 위치기반 서비스의 소개와 위치기반 서비스에서 이동패턴을 탐사하는 다양한 연구 및 패턴 탐사 알고리즘을 살펴본다. 본문에서는 실내 환경에서 Stay point를 탐색하는 알고리즘과 반발 패턴을 탐색하는 방법을 소개한다. 결론에서는 제안된 알고리즘을 이용하여 구현된 POI 추천 시스템의 성능을 평가하였다.

## II. 관련 연구

본 장에서는 본 논문에서 제안하고 있는 POI 추천 시스템이 구현된 모바일 기반 실내외 검용 위치기반 서비스를 소개하고 Stay point와 사용자의 다양한 위치정보로부터 유용한 이동패턴을 탐색하는 선행연구 및 위치 데이터로부터 이동패턴 탐사를 위해 사용되는 대표적인 연관규칙 탐사 알고리즘인 Apriori 알고리즘과 FP-Growth 알고리즘에 대해 살펴본다.

### 2.1 Indoor & Outdoor LBS System

본 논문에서 제안하고 있는 POI 추천 시스템이 구현된 그림 1의 모바일 기반 실내외 검용 위치기반 서비스는 스마트폰 및 모바일 환경에서 실내 환경과 실외 환경의 검용 위치기반 서비스 지원을 위해 개발된 서비스로 Indoor & Outdoor LBS System은 Indoor Position Determination 시스템과 연계하여 공간연산을 통해 코엑스와 같은 복잡한 상업구역에서의 실내위치 내비게이션 서비스, 친구 찾기, POI(Point of interest) 추천 등의 다양한 위치기반 서비스를 제공하고 있다.

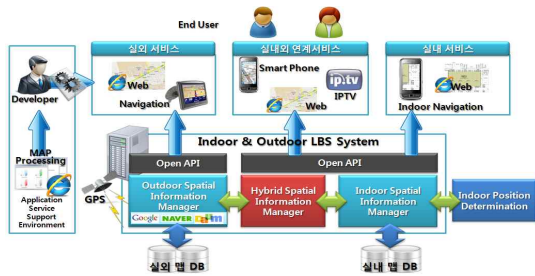


그림 1. 모바일 기반 실내외 검용 위치기반 서비스 구성도  
Fig. 1. System Architecture

### 2.2 Stay point의 정의

Stay point는 사용자가 잠시 동안 머물렀던 지점으로써 GPS 로그에서 해당 포인트를 특별한 의미가 있는 장소로 정

의하고 있다.[5] Sonia Khetarpaul 등은 GPS기록으로 부터 GPS 장치를 가진 사용자가 이동 중 어떤 목적을 가지고 특정 상점이나 빌딩 안으로 들어가면서 GPS 신호가 사라진 시간부터 빌딩 밖으로 나왔을 때 신호가 다시 나타난 시간의 차를 계산하여 해당 포인트에 임계시간 이상 머물렀을 때 이를 Stay point로 탐사하고 있다.

### 2.3 이동패턴 탐사 방법의 연구

끊임없이 변하는 사용자의 위치정보는 그 데이터의 양이 매우 많고 다양하다. 위치기반 서비스를 위해서는 이런 사용자의 위치정보로부터 불필요한 위치 데이터를 제거하고 대량의 데이터로부터 빠르게 유용한 사용자의 이동패턴을 탐색하는 연구가 필요하다. 이러한 이동패턴에 대한 연구는 기계학습이나 마이닝 기법을 이용한 다양한 방법으로 연구되고 있다.

기존의 연구에서 Sonia Khetarpaul 등은 Stay point와 지도도를 통해 GPS 데이터로부터 사용자의 관심지점을 마이닝 하는 방법을 제안하고 있다.[5] Quannan L 등은 트리형태의 자료구조를 사용하여 사용자들의 GPS 데이터로부터 위치데이터 간의 유사성을 탐색하는 HGSM(Hierarchical graph modeling user location history)기법을 제안하고 있다.[6]

그 밖에 RSOM과 베이저안 네트워크 등의 신경망 알고리즘을 이용하여 유저의 이동패턴을 분석하는 연구[7]과 시공간 데이터에 대한 이동패턴 탐색을 위한 STMPE(Spatio-Temporal Moving Pattern Extracting)등과 전통적인 마이닝 알고리즘인 Apriori 와 FP-Tree를 응용한 다양한 탐사 기법이 연구되고 있다.[6]

사용자의 관심지점 추천을 위한 이동패턴 탐사에서 기계학습 기법을 이용한 방법은 이동패턴 탐사를 위한 모델링 과정이 필요하며 이동 경로 예측을 위한 마르코프 모델은 매우 결정적이 때문에 유연성이 떨어지는 단점이 있다. 마이닝 기법을 이용한 이동패턴 분석 방법들은 수집된 데이터를 모두 분석하여 특정한 패턴을 발견함으로써 예측에 유리한 기계학습 기법보다 신뢰성 있는 분석이 가능하다. 하지만 전통적인 마이닝 기법을 이용한 STMPE는 수집된 데이터가 적으면 그 신뢰성이 떨어지고 반면 신뢰성을 높이기 위해 많은 데이터를 분석하게 되면 성능이 저하되는 문제점을 가지고 있다. HGSM기법은 이동패턴 분석을 위해 Stay point만을 사용하기 때문에 패턴탐색에 필요한 데이터가 적으며 사용자가 관심 있게 머무른 지점의 시간적 특성의 반영이 가능하다는 장점이 있다. 하지만 GPS 신호의 On/Off를 이용한 방법은 실내 환경에서 적용할 수 없는 문제점이 있다. 따라서 본 논문에서는

실내 환경에서 Stay point를 탐사하는 방법을 제안하고 Stay point를 이용하여 사용자의 관심지점을 탐사함으로써 이동패턴 분석에 필요한 데이터를 줄이고 시간적 속성을 반영하고자 한다.

## 2.4 이동패턴 연관규칙 탐사 알고리즘

비슷한 이동패턴을 보이는 사용자들 간의 관심지점을 발견하기 위해서는 같은 성향의 사용자들이 빈번하게 방문한 지점에 대한 연관규칙 탐사가 필요하다. 연관규칙 탐사에 사용되는 대표적인 알고리즘으로 Apriori 알고리즘과 FP-Growth 알고리즘이 있다. Apriori 알고리즘과 FP-Growth 알고리즘은 연관규칙을 탐색하는 마이닝 기법으로 이동패턴 탐사에도 사용되고 있다.

### 2.4.1 Apriori 알고리즘

Apriori 알고리즘[7]은 이진연관규칙에 대한 빈발 항목집합을 찾아내는데 유용한 알고리즘이다. Apriori 알고리즘은 n번째 항목집합이 n+1번째 항목집합을 발견하기 위해 사용되는 레벨단위로 진행되는 반복 접근법을 사용한다. Apriori 알고리즘은 각 노드에서 생성된 후보 항목집합 중에서 빈발항목 집합을 찾아내기 위한 후보 집합 생성을 위해 매번 데이터 베이스를 스캔해야 하는 문제점이 있다. Apriori 알고리즘은 빈발항목집합을 발견하기 위해 레벨단위로 반복 접근한다.[10]

Apriori 알고리즘은 최초 1-빈발항목집합을 찾는다. 이 집합을 L1으로 나타내며, L1은 2-빈발항목집합인 L2를 찾는데 사용되며 L2는 3-빈발항목집합을 찾는 데 이용되는 식으로 더 이상 k-번째 빈발항목집합이 없을 때까지 진행된다. 이러한 Apriori 알고리즘의 특성은 모든 공집합이 아닌 빈발항목집합의 부분집합은 반드시 빈번하다는 것이다. 항목집합 I가 최소 지지도 임계치(minimum support threshold)인  $\min\_sup$ 를 만족하지 못하면 I는 빈번하지 않으므로  $P(I) < \min\_sup$ 가 된다. 항목 A가 항목집합 I에 추가되면 결과 항목집합은 IUA가 되며 이것 또한 I보다 더 빈번할 수 없다. 그러므로 IUA는 빈번하지 않으므로  $P(IUA) < \min\_sup$ 가 된다.

### 2.4.2 FP-Growth 알고리즘

Apriori 알고리즘은 이진 연관규칙에 대한 빈발항목집합을 찾아내는데 유용한 알고리즘이지만 대량의 후보 집합을 생성하여야 하며 이런 후보 집합을 생성하기 위해서는 빈번한 데이터베이스 스캔이 필요하다. FP-Growth는 이런 후보 집합의 생성 없이 연관규칙을 탐색하는 방법을 제안하고 있다.

FP-Growth는 빈발 항목을 가지는 데이터베이스를 FP-Tree로 압축하여 사용한다. FP-Tree 생성 시에는 전체 데이터베이스를 두 번 탐색하게 된다.[10]

첫 번째 탐색에서는 빈발아이템 리스트를 찾아낸다. 빈발아이템 리스트는 최소지지도보다 높은 지지도를 갖는 아이템들의 리스트이며 지지도의 내림차순으로 정렬되어 진다. 두 번째 탐색에서는 빈발아이템 리스트의 순서에 따라 트랜잭션을 검색하며 재귀적 호출을 통해 FP-Tree를 생성하게 된다.

FP-Tree 생성 후 연관규칙 탐색은 FP-Tree를 마이닝 하는 문제로 변화된다. FP-Tree 트리를 탐색하여 조건부 패턴 베이스를 만들고 조건부 패턴 베이스를 기반으로 하여 조건부 FP-Tree를 찾는데 최소지지도에 따라 그 이상으로 반복되어 나오는 것만 선택하게 된다. 마지막으로 이를 기반으로 빈발 패턴을 찾아내게 된다.

## III. Stay point 탐색 알고리즘과 방문패턴 탐색

사용자에게 POI를 추천하기 위해서는 사용자들의 이동기록이나 구매 이력을 분석하여 비슷한 성향을 보이는 사용자들 간의 패턴을 탐사하여야 한다. 하지만 사용자의 구매 이력만을 이용한 연관규칙 탐색은 사용자가 관심이 있는 상품이나 매장임에도 직접적인 구매가 일어나지 않으면 기록이 남지 않아 선호 지점 판단에서 제외되는 문제점이 있으며 사용자의 모든 위치데이터를 저장하여 이동기록으로 부터 선호 지점을 탐색하는 방법은 방대한 데이터를 분석하여야 하는 문제점과 관심지점까지 이동하는 데 지나온 경로가 빈발패턴 탐사에 포함되는 문제점이 있다. 따라서 실내위치기반 서비스에서 사용자의 다양한 관심지점을 고려하기 위해서는 사용자가 일정 시간 동안 관심 있게 머무른 위치를 선호하는 지점으로 하여 해당 위치들에 대해서만 연관규칙을 탐색할 필요성이 있다. 이에 본 논문에서는 사용자가 일정 시간 관심 있게 머무른 지점을 Stay point라 정의하고 실내 공간에서 사용자의 Stay point를 탐색하는 알고리즘을 제안한다.

본 장에서는 3.1절에서 실내 공간에서 Stay point를 탐색하기 위한 알고리즘을 설명하고 3.2절에서는 Stay point를 이용한 방문패턴을 탐색하는 방법을 설명하였다.

### 3.1 Stay point 탐색 알고리즘

제안된 알고리즘은 사용자가 최소 임계시간 이상 특정 지점에 머물러 있을 때 이를 Stay point라 정의하며 이는 사용

자 관심도가 높은 지점으로 정의할 수 있다.

이동 중인 사용자의 Stay point를 측정하기 위해서 사용자가 특정 위치에 도착한 시간과 떠난 시간의 계산을 통해 임계시간 이상 머문 Stay point를 탐색할 수 있다. 하지만 건물 안으로 들어갈 때 사라지고 건물 밖에서 다시 수신된 GPS 신호를 이용한 연구[5][6]과 다르게 실내 공간에서는 사용자가 특정 지점에 도착한 시간과 떠난 시간을 구분 지을 수 없다. 따라서 실내 공간에서는 일정 간격으로 계속 사용자의 위치를 확인하여 반복적으로 나타나는 특정 지점의 시간을 계산하여 Stay point를 탐색하여야 한다.

본 연구에서는 표 1과 같은 Current Table과 표 2와 같은 Update Table을 두어 사용자의 위치정보를 관리한다. Current Table은 현재 사용자가 위치하고 있는 지점의 POI와 측정된 시간을 저장하는 테이블이다. Update Table은 Current Table이전의 사용자 위치정보 즉 과거의 위치정보를 기록하기 위한 테이블이다. c.time은 현재 사용자의 시간 정보이며 u.time은 과거의 시간 정보이다. stayTime은 Current Table의 c.time과 Update Table의 u.time의 시간차를 계산하여 사용자가 해당 지점에 머무른 시간을 의미한다.

그림 2는 Stay point 탐색 알고리즘으로 라인 1에서는 일정한 시간 간격으로 기록된 사용자의 Current Table의 POI와 이전에 기록된 Update Table의 POI가 같은지 비교한다. 라인 2에서는 POI정보가 같을 경우 현재 시간 c.time 과 과거 시간 u.time의 시간차를 계산하여 stayTime에 누적시켜 저장한다. 라인 3에서는 누적된 stayTime이 임계시간 이상일 경우 현재 poi를 Stay point로 저장한다. 라인 7에서는 라인 1의 두 POI가 같지 않을 경우 Update Table의 정보를 Current Table로 교체하고 stayTime을 0으로 초기화시킨다.

표 1. 커런트 테이블  
Table 1. Current Table

| Current Table |                 |     |
|---------------|-----------------|-----|
| uid           | c.time          | poi |
| user001       | 2012/1/21/01:10 | A   |

표 2 업데이트 테이블  
Table 2 Update Table

| Update Table |                 |     |          |
|--------------|-----------------|-----|----------|
| uid          | u.time          | poi | stayTime |
| user001      | 2012/1/21/01:09 | A   | 01:00    |

Algorithm : StayPoint\_detection

```

1: If(Current Table.POI and equal Update Table.POI) then
2:   stayTime = stayTime +(c.time - u.time)
3:   If(stayTime is greater than the threshold time) then
4:     Current POI is recorded in the staypoint database
5:   End If
6: Else
7:   Replace the Current Table Update Table and Staytime
   initialized to 0
8: End If
    
```

그림 2. StayPoint 탐색 알고리즘  
Fig. 2. StayPoint detection Algorithm

표 3. 사용자들의 위치데이터 기록  
Table 3. User location trajectory

| usr \ seq | seq |   |   |     |   |   |   |    |  |  |
|-----------|-----|---|---|-----|---|---|---|----|--|--|
|           | 1   | 2 | 3 | ... | 7 | 8 | 9 | 10 |  |  |
| user001   | A   | A | A | ... | C | K | C | C  |  |  |
| user002   | I   | J | C |     | D | D | A | C  |  |  |
| user003   | K   | K | B |     | A | A | A | B  |  |  |
| ⋮         | ⋮   | ⋮ | ⋮ |     | ⋮ | ⋮ | ⋮ | ⋮  |  |  |
| user007   | A   | A | A |     | J | J | A | D  |  |  |
| user008   | J   | A | A |     | K | B | B | B  |  |  |

일정한 시간 간격으로 수집된 사용자들의 방문 이력이 표 3과 같을 때 표 4는 사용자들의 Stay point를 탐색하는 과정을 보여주고 있다. 이때 사용자들의 위치는 2분 간격으로 수집하며 Stay point가 되기 위한 임계시간은 4분 이상으로 가정 하였다.

표 4의 seq1은 최초 사용자들의 위치가 측정된 1:10을 기준으로 Update Table을 생성하고 이후 사용자들의 두 번째 위치를 측정한 01:12분의 기록으로 Current Table을 구성하였다. user001, user003, user007을 보면 과거 poi기록과 현재 poi기록이 같음으로 stay time을 계산하여 seq2의 Update Table과 같이 stayTime을 갱신한다. 같은 방법으로 seq2에서 현재 위치의 사용자정보와 과거 위치의 사용자정보를 비교한다.

이때 user001 과 user007은 이전 stayTime과 현재 계산된 StayTime의 합이 임계시간 이상이기 때문에 Stay point로 데이터베이스에 저장한다. seq3 에서는 user008에 새로운 Stay point가 나타남을 볼 수 있다. 데이터베이스에 저장된 Stay point는 표 5와 같다. 같은 과정의 반복으로 모든 방문 이력에 대한 Stay point를 저장한다.

표 4. Stay point 탐색 과정  
Table 4. Stay point detection process

| Seq 1        |                 |     |           | Current Table |                 |     |
|--------------|-----------------|-----|-----------|---------------|-----------------|-----|
| Update Table |                 |     |           | uid           | c.time          | poi |
| uid          | u.time          | poi | stay time | user001       | 2012/1/21/01:12 | A   |
| user001      | 2012/1/21/01:10 | A   | 00:00     | user002       | 2012/1/21/01:12 | J   |
| user002      | 2012/1/21/01:10 | I   | 00:00     | user003       | 2012/1/21/01:12 | K   |
| user003      | 2012/1/21/01:10 | K   | 00:00     | ⋮             |                 |     |
| ⋮            |                 |     |           | user007       | 2012/1/21/01:12 | A   |
| user007      | 2012/1/21/01:10 | A   | 00:00     | user008       | 2012/1/21/01:12 | A   |
| user008      | 2012/1/21/01:10 | J   | 00:00     |               |                 |     |

| Seq 2        |                 |     |           | Current Table |                 |     |
|--------------|-----------------|-----|-----------|---------------|-----------------|-----|
| Update Table |                 |     |           | uid           | c.time          | poi |
| uid          | u.time          | poi | stay time | user001       | 2012/1/21/01:14 | A   |
| user001      | 2012/1/21/01:12 | A   | 00:02     | user002       | 2012/1/21/01:14 | C   |
| user002      | 2012/1/21/01:12 | J   | 00:00     | user003       | 2012/1/21/01:14 | B   |
| user003      | 2012/1/21/01:12 | K   | 00:02     | ⋮             |                 |     |
| ⋮            |                 |     |           | user007       | 2012/1/21/01:14 | A   |
| user007      | 2012/1/21/01:12 | A   | 00:02     | user008       | 2012/1/21/01:14 | A   |
| user008      | 2012/1/21/01:12 | A   | 00:00     |               |                 |     |

| Seq 3        |                 |     |           | Current Table |                 |     |
|--------------|-----------------|-----|-----------|---------------|-----------------|-----|
| Update Table |                 |     |           | uid           | c.time          | poi |
| uid          | u.time          | poi | stay time | user001       | 2012/1/21/01:16 | B   |
| user001      | 2012/1/21/01:14 | A   | 00:04     | user002       | 2012/1/21/01:16 | C   |
| user002      | 2012/1/21/01:14 | C   | 00:00     | user003       | 2012/1/21/01:16 | K   |
| user003      | 2012/1/21/01:14 | B   | 00:00     | ⋮             |                 |     |
| ⋮            |                 |     |           | user007       | 2012/1/21/01:16 | I   |
| user007      | 2012/1/21/01:14 | A   | 00:04     | user008       | 2012/1/21/01:16 | D   |
| user008      | 2012/1/21/01:14 | A   | 00:02     |               |                 |     |

표 5. 발견된 Stay point 로그  
Table 5. Detected stay point log

| uid     | poi | stayTime |
|---------|-----|----------|
| user001 | A   | 00:04    |
| user007 | A   | 00:04    |
| user008 | A   | 00:04    |
| ...     |     |          |

3.2. FP-Growth 이용한 방문패턴 탐색

비슷한 성향의 이동패턴을 보이는 사용자들 간의 관심지점을 탐색하기 위해서는 발견된 Stay point로부터 사용자들이 빈번하게 방문하는 지점의 패턴을 탐색하여야 한다.

본 논문에서 구현된 시스템에서는 사용자들의 방문 기록에 대한 빈발패턴 탐색을 위해 FP-Growth 알고리즘을 이용하였다. FP-Growth는 Apriori 와 달리 대량의 후보 집합 생성이 필요 없어 데이터베이스 스캔을 줄일 수 있는 장점이 있다.

우선 데이터베이스를 1회 스캔하여 1-항목 집합과 지지도를 유도한다. 최소 지지도는 2로 가정하고 빈발항목 집합은

지지도를 기준으로 내림차순 정렬한다. 이렇게 생성된 1항목 집합을 기준으로 FP-Tree를 생성한다. FP-Tree의 생성은 우선 루트를 NULL로 생성하고 데이터베이스를 스캔한다. 트리의 가지는 표 6의 지도도에 따라 내림차순 정렬된 1-항목 집합의 순서로 생성된다.

먼저 생성된 노드가 부모노드가 되고 다음 생성된 노드가 자식 노드가 되어 트리를 구성한다. 하나의 Stay point에 링크가 추가될 경우 공통노드의 카운터를 증가한 후 다음 순서의 Stay point 노드를 생성하고 링크로 연결한다. 그림 3은 완성된 FP-Tree를 보여주고 있다. FP-Tree의 연관규칙은 상향식으로 생성되며 따라서 빈발패턴을 마이닝 하는 문제는 FP-Tree를 마이닝 하는 문제로 변환된다. FP-Tree의 마이닝은 길이 1인 초기의 접미부 패턴에서 시작하여 조건부 패턴 베이스를 생성하고 조건부 FP-Tree를 생성한다. 그리고 생성된 트리를 재귀적으로 탐색하여 마이닝을 수행한다. 패턴의 증가는 접미부 패턴과 조건부 FP-Tree로부터 생성된 빈발 패턴을 결합하여 얻어진다.

표 6. 1-항목집합 과 내림차순 정렬된 1-항목집합  
Table 6. A set of candidate entries And A set of candidate entries sorted descending

| Stay point | 지지도 | Stay point | 지지도 |
|------------|-----|------------|-----|
| A          | 6   | B          | 7   |
| B          | 7   | A          | 6   |
| C          | 6   | C          | 6   |
| D          | 2   | D          | 2   |
| E          | 2   | E          | 2   |

| 사용자 ID  | Stay point | 사용자 ID  | Stay point |
|---------|------------|---------|------------|
| user001 | A,B,E      | user001 | B,A,E      |
| user002 | B,D        | user002 | B,D        |
| user003 | B,C        | user003 | B,C        |
| user004 | A,B,D      | user004 | B,A,D      |
| user005 | AC         | user005 | AC         |
| user006 | B,C        | user006 | B,C        |
| user007 | AC         | user007 | AC         |
| user008 | A,B,C,E    | user008 | B,A,C,E    |
| user009 | A,B,C      | user009 | B,A,C      |

<1-항목집합> <내림차순 정렬된 1-항목집합>

그림 3에서 Stay point E를 보면 두 개의 노드에서 발견 된다. E를 접두부로 하면 두 개의 접두부 경로 (B A E:1)와 (B A C E:1)이 되며 이 접두부 경로들은 조건부 패턴 베이스를 구성한다. E의 조건부 FP-Tree의 단일 경로는 (B A:2)가 된다. C가 포함되지 못하는 이유는 C의 지지도가 1로서 최소 지지도 보다 작기 때문이다. 이 단일 경로의 모든 조합 (B E:2) (A E:2) (B A E:2)가 빈발패턴으로 생성된다. 표 7은 위와 같은 방법으로 마이닝 된 빈발 패턴을 보여준다.

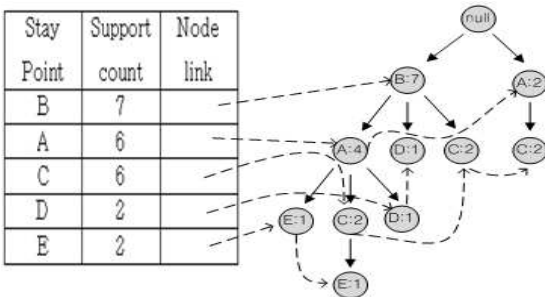


그림 3. 완성된 FP-Tree  
Fig. 3. Completed FP-Tree

표 7. 마이닝 된 빈발패턴  
Table 7. Frequent pattern

| Stay point | 조건부 패턴집합                    | 조건부 빈발패턴 트리         | 빈발패턴 방문지점                         |
|------------|-----------------------------|---------------------|-----------------------------------|
| E          | {B,A:1},<br>{B,A,C:1}       | {B:2,A:2}           | {B,E:2},<br>{A,E:2},<br>{B,A,E:2} |
| D          | {B,A:1},<br>{B:1}           | {B:2}               | {B,D:2}                           |
| C          | {B,A:2},<br>{B:2},<br>{A:2} | {B:4,A:2},<br>{A:2} | {B,C:4},<br>{A,C:4},<br>{B,A,C:2} |
| A          | {B:4}                       | {B:4}               | {B,A:4}                           |

### IV. 성능평가

본 장에서는 사용자의 이동 트랜잭션 데이터에서 Full point와 Stay point로부터 추출된 빈발패턴의 비교를 통해 Stay point를 이용한 빈발패턴 탐사가 Full point를 이용한 빈발패턴보다 적은 데이터를 가지고도 비슷한 패턴을 탐색할 수 있음을 실험하였다.

실험에 사용된 데이터는 Indoor & Outdoor LBS System의 평가를 위해 수집한 테스트 데이터를 이용하였으며 테스트와 데이터 수집은 6개월간 중국 중경 시에 있는 쇼핑센터에서 진행되었다. 데이터는 쇼핑몰 이용 고객들에게 간단한 설문 조사를 거친 후 테스트용 단말기를 지급하여 사용자가 쇼핑센터에 머무르는 동안의 위치 데이터를 수집하였다. 이렇게 수집된 데이터로부터 사용자들의 쇼핑 동선을 비슷하게 구성하여 빈발한 패턴의 출현을 높이기 위해 표 8과 같은 실험 군으로 분류 하여 데이터를 선별 하였다. 이때 쇼핑 목적을 의류, 화장품으로 답했던 참여자들의 데이터를 우선으로 선별하여 각각 그룹으로 분류 하였다. 그리고 데이터가 편중되는 것을 피하고 각 집단별 데이터 건수를 고르게 분포시키기 위하여 수집된 데이터로부터 추가적으로 성별과 비슷한 연령대의 데이터를 선별하여 각각의 집단을 2,000건의 데이터로 구성 하였다. 성별과 연령대가 비슷한 사용자의 데이터를 선별한 이유는 별도의 데이터 분석 없이 연령대와 성별이 선호하는 브랜드나 쇼핑 성향이 가장 비슷할 것으로 추정하기에 적합한 속성이기 때문이다. 이렇게 실험 데이터 집단을 구성한 이유는 적은 데이터 속에서 의미 있는 빈발패턴의 출현을 높이고자 함이다.

실험에 사용된 사용자들의 위치데이터는 시스템이 테스트

중인 쇼핑센터의 1층을 대상으로 하였으며 사용자 1인당 60분간 LBS 서버로부터 1분 간격으로 수집되며 Stay point의 임계시간은 3분으로 설정 하였다. 총 10,000건의 실험에 사용된 트랜잭션의 평균 길이, 최대길이, 아이템 수는 표 9와 같다.

표 8. 실험군  
Table 8. Experimental group

| 연령     | 성별  | 참여 수  |
|--------|-----|-------|
| 20대    | 남   | 2,000 |
|        | 여   | 2,000 |
| 30대    | 남   | 2,000 |
|        | 여   | 2,000 |
| 30대 이상 | 남/여 | 2,000 |

표 9. 트랜잭션 집합  
Table 9. Transaction Set

| 구분         | 평균 길이 | 최대 길이 | 아이템 수  |
|------------|-------|-------|--------|
| Full point | 55    | 60    | 10,000 |
| Stay point | 8     | 15    | 10,000 |

먼저 10,000건의 트랜잭션 데이터에서 Full point와 Stay point의 지도도별 발견된 빈발패턴은 그림 4와 같다. 그림 4의 그래프에서 사용자의 Stay point를 고려하여 탐색한 빈발패턴의 수를 보면 패턴의 지도도가 높아질수록 사용자의 모든 이동 기록으로 생성한 빈발패턴인 Full\_point 빈발패턴과 큰 차이를 보이지 않고 있음을 알 수 있다. Full\_point의 빈발 패턴 수는 가장 많은 패턴을 생성하지만, 지도도가 높아질수록 빈발패턴의 수가 급격히 저하되며 Stay\_point의 빈발 패턴 수와 비슷해짐을 확인 할 수 있다. 이는 쇼핑몰 내에서 사용자들이 특정 관심지점에 이르기까지 서로 다른 다양한 경로로 이동하며 거쳐 간 지점들이 빈발패턴에 포함되어 패턴수가 늘어났기 때문이다. 따라서 지도도가 높아짐에 따라 지도도가 낮은 빈발패턴이 가지치기 되어 감소함을 볼 수 있으나 관심지점은 사용자들 간의 지나온 경로에 상관없이 Full point나 Stay point 모두 같은 지점을 포함하고 있기 때문에 Stay\_point의 빈발패턴과 유사해 짐을 볼 수 있다.

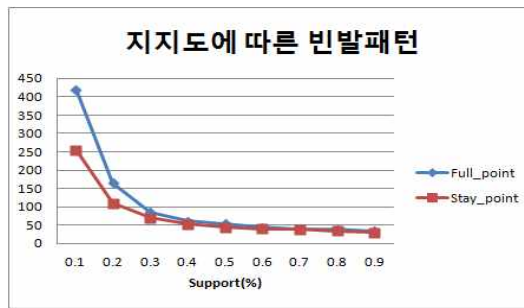


그림 4. 최소 지도도별 빈발패턴  
Fig. 4. Frequent patterns by min\_sup

표 10은 발견된 빈발패턴의 유사성을 살펴보기 위해 최소 지도도 0.9 구간에서 발견된 패턴 중 가장 빈번하게 나타난 POI F104에 대한 각각 트랜잭션 별 패턴을 보여주고 있다.

그림 5 쇼핑센터 지도에 표시된 POI F104는 화장품 전문 매장이다. 먼저 Full\_point에서 발견된 POI들의 실제 지도상 위치를 보면 F103, F102는 F104와 같은 화장품 전문 매장이며 F122는 두 매장 사이에 있는 통로의 POI임을 알 수 있다.

발견된 빈발 패턴 중 화장품 매장의 빈발패턴이 높았던 이유는 F103, F104, F102 모두 인접해 있었으며 같은 카테고리 상품의 취급하고 있었기 때문이다. 이렇게 인접한 지역에서 같은 업종의 매장이 빈번하게 탐사 된 이유는 사용자의 쇼핑 패턴이 단순히 물건을 구입하고 끝내는 것이 아니라 다양한 상품비교를 위해 여러 관심지점을 방문하는 것으로 보여 진다

이는 F122 통로는 F103에서 F104로 이동한 사용자들이 빈번하게 거쳐 간 지점으로 Full point를 이용한 패턴 탐사에서 경로로써 포함되었음을 알 수 있다. 반면 Stay point를 이용한 패턴 탐사에서는 같은 매장에 대한 패턴 (F103, F104)에 경로 패턴이 포함 되어 있지 않았으며 Full point를 고려한 빈발패턴

에 포함된 패턴 {F103, F104}, {F102, F104}, {F110, F104}가 모두 발견됨을 볼 수 있다. 이는 Stay point를 이용한 빈발패턴 탐사가 중간 경로를 포함하지 않으면서도 Full point를 이용한 빈발패턴 탐사와 유사한 결과를 나타냄을 보여준다.



표 10. 발견된 F104의 빈발항목 집합  
Table 10. Frequent item sets of F104

|            | 빈발항목  |
|------------|---|
| Full_point | {F103, F104}, {F103, F122, F104},<br>{F102, F104}, {F110, F104} |
| Stay_point | {F103, F104}, {F102, F104},<br>{F110, F104}                     |

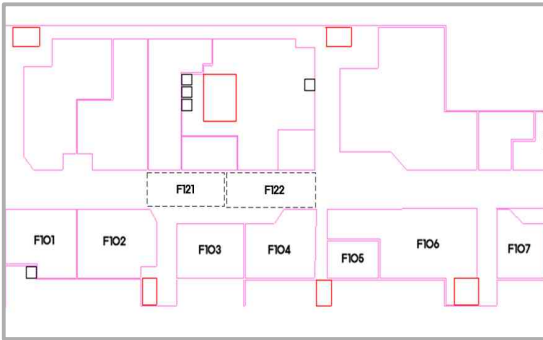


그림 5. 시스템이 구축된 쇼핑센터의 지도  
Fig. 5. Map of shopping centers

각 Stay\_point와 Full\_point 트랜잭션의 집합에서 최소 지지도 별 빈발패턴 탐색에 소요된 수행 시간은 그림 6과 같다. Stay\_point의 빈발패턴 탐색에 소요된 수행 시간은 각각 (4.2초, 5.6초, 6.8초) 이었으며 Full\_point 빈발패턴 탐색에 소요된 수행 시간은 각각 (8.2초, 10.7초, 14.5초) 로 Stay\_point 빈발패턴 탐색이 더 나은 성능을 보여주고 있다.

그림 7은 최소 지지도별 메모리 사용량을 보여주고 있다. 그림 6의 빈발패턴 탐색에 소요된 수행 시간 차이에 비해 트랜잭션별 메모리 소요량은 큰 차이를 보이고 있다. 이는 트랜잭션의 크기가 증가함에 따라 메모리 소요량이 급격히 증가하며 트랜잭션의 평균 길이와 빈발항목의 평균길이에 민감한 FP-Tree 특성[11] 때문이다.

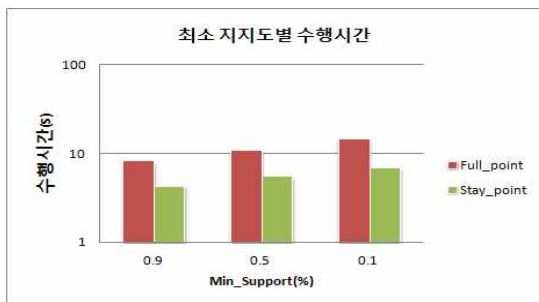


그림 6. 최소 지지도별 수행시간  
Fig. 6. Execution time

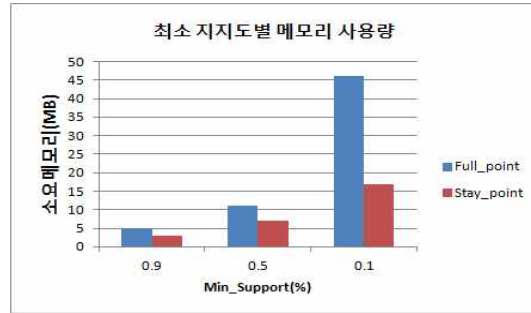


그림 7. 최소 지지도별 메모리 사용량  
Fig. 7. Memory Usage

표 11은 지지도별 Full\_point를 이용한 빈발패턴 탐사와 논문에서 제안한 알고리즘을 이용하여 탐색된 Stay\_point를 이용한 패턴탐사에 소요된 수행시간과 메모리, 발견된 패턴수의 비교를 보여주고 있다. 각 지지도별 빈발패턴 탐색에 소요된 수행시간은 Stay\_point를 이용한 탐색이 Full\_point를 이용한 패턴 탐색보다 약 50%정도의 빠른 수행 성능을 보여주었으며 메모리 사용량은 약 46%정도 적게 사용함을 볼 수 있었다. 반면 발견된 패턴의 수를 보면 지지도가 높아질수록 사용자의 전체 이동기록을 이용하여 발견한 빈발패턴과 Stay\_point만을 가지고 발견한 빈발패턴의 수가 비슷해지고 있음을 볼 수 있다. 전체적인 결과를 종합해 보면 지지도가 높아질수록 사용자의 관심지점을 탐사 하기위해 Stay\_point를 이용하는 것이 Full\_point를 이용하는 것보다 더 나은 성능을 보이면서도 비슷한 빈발패턴을 찾을 수 있음을 확인할 수 있었다. 이는 빈발패턴 탐색에 사용된 데이터의 레코드 수가 같더라도 전체 위치데이터를 이용한 Full\_point보다 사용자가 특정 지점에 위치한 데이터만을 이용한 Stay\_point 데이터의 평균길이가 더 적었기 때문이다.

표 11. Full\_point 와 Stay\_point의 성능비교  
Table 11. Performance comparison of Full\_point and Stay\_point

| 지지도별 발견된 빈발패턴 |     |     |     |
|---------------|-----|-----|-----|
| 구분            | 0.1 | 0.5 | 0.9 |
| Full point    | 420 | 55  | 35  |
| Stay point    | 255 | 45  | 30  |

| 지지도별 수행시간  |     |      |      |
|------------|-----|------|------|
| 구분         | 0.1 | 0.5  | 0.9  |
| Full point | 8.2 | 10.7 | 14.5 |
| Stay point | 4.2 | 5.6  | 6.8  |
|            | 51% | 52%  | 46%  |

| 지지도별 메모리 사용량 |     |     |     |
|--------------|-----|-----|-----|
| 구분           | 0.1 | 0.5 | 0.9 |
| Full point   | 5   | 11  | 48  |
| Stay point   | 2   | 7   | 17  |
|              | 40% | 63% | 35% |

**참고문헌**

- [1] Jae Kyeong Kim, Kyung Hee Chae, Min Yong Kim, "A Store Recommendation Procedure in Ubiquitous Market," Journal of Intelligence and Information Systems, Vol 13, No. 4, pp.45-63, Dec. 2007.
- [2] Jong-Hee Kim, Soon-Key Jung, "The Goods Recommendation System based on modified FP-Tree Algorithm" Journal of the Korea society of computer and information, pp.205-213, Jun. 2010.
- [3] Jin Byeong Woon, Cho Young Sung, Ryu Keun Ho, "Personalized e-Commerce Recommendation System using RFM method and Association Rules" Journal of the Korea society of computer and information, pp.227-235, Sept. 2010.
- [4] Vu. Thi Hong Nhan, Bum Ju Lee, Keun Ho Ryu, "Discover of Frequent Sequence Pattern in Moving Object Databases," The KIPS Transactions:PartD, Vol 15D, No. 2, pp.163-170, Apr. 2008.
- [5] Sonia Khetarpaul, Rashmi Chauhan, S K Gupta, L Venkata Subramaniam, Ullas Nambiar, "Mining GPS Data to Determine Interesting Locations," IIWeb '11 Proceedings of the 8th International Workshop on Information Integration on the Web: in conjunction with WWW 2011, No. 8, Mar. 2011.
- [6] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, Wei-Ying Ma, "Mining User Similarity Based on Location History," GIS '08 Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems, No. 34, Nov. 2008
- [7] Young Hwan Choi, Sang Yong Lee, "Users' Moving Patterns Analysis for Personalized Product Recommendation in Offline Shopping Malls," Journal of Korean Institute of Intelligent Systems, Vol 16, No. 2, pp.185-190, Apr. 2006.
- [8] Ji Woong Park, Dong Suk Hong, Dong Oh Kim, Ki Joon Han, "An Efficient Algorithm for Spatio-Temporal Moving Pattern Extraction," Journal of Korea Spatial Information System Society, Vol 8, No. 2, pp.1-124, Sept. 2006.
- [9] Rakesh Agrawal, Ramakrishnan Srikant, "Mining Sequential Patterns," Proc. of the Int. Conf. on Data Engineering, Taipei, Taiwan, 1995.
- [10] Jiwei Han and Micheline Kamber, "Data Mining Conceptions and Techniques, ELSEVIER Inc., pp.224-235 2006.
- [11] Hyung Bong Lee, "An implementation and Performance Characteristics of the FP-tree Association Rules Mining Algorithm," Proceedings of the Korea Information Processing Society Conference, Vol 13, No. 2, pp.337-340, Nov. 2006.

**저 자 소개**



**김 범 수**  
 2010 : 한서대학교 컴퓨터정보학과 이학사.  
 2012~현재 : 인하대학교 컴퓨터정보공학과 공학석사  
 관심분야: 공간 데이터베이스, 데이터마이닝, 데이터베이스 설계, 지리 정보 시스템  
 Email : bskim@dlab.inha.ac.kr



**이 연**  
 2006 : 중국 중경우전대학교 지리정보시스템학과 이학사  
 2008 : 인하대학교 컴퓨터정보공학과 공학석사  
 2008~현재 : 인하대학교 컴퓨터정보공학과 박사과정  
 관심분야: 공간 데이터베이스, 공간 데이터웨어하우스, 지리 정보 시스템, USN, 스트림 데이터 시스템  
 Email : leeyeon@dlab.inha.ac.kr



**김 경 배**

1992 : 인하대학교 전자계산공학과  
공학사

1994 : 인하대학교 대학원 전자  
계산공학과 공학 석사

2000 : 인하대학교 대학원 전자  
계산공학과 공학 박사

2000~ 2004 : 한국전자통신연구원  
선임연구원

2004년~현재 : 서원대학교 컴퓨터  
교육과 조교수

관심분야 이동실시간 데이터베이  
스, 스토리지 시스템,  
GIS, VOD

Email: gbkim@seowon.ac.kr



**배 해 영**

1974 : 인하대학교 응용물리학 공  
학사

1978 : 연세대학 전자계산학 공학  
석사

1989 : 숭실대학교 전자계산학 공학  
박사

2006~2009 : 인하대학교 정보통  
신대학 원장, 대학  
원장 역임

현재 : 인하대학교 컴퓨터정보공학  
과 교수, 지능형 GIS연구  
센터 센터장, 중국 중경우  
전대학교 명예교수

관심분야 데이터베이스, 공간 데이  
터베이스

Email : hybae@inha.ac.kr

