

시간을 고려한 모바일 사용자의 유용한 행동패턴 추출

이승철* 김응모*

*성균관대학교 전자전기컴퓨터공학과
e-mail: {eddie, umkim}@ece.skku.ac.kr

Efficient Mining of User Behavior Patterns by Temporal Access

SeungCheol Lee* and UngMo Kim*

*Dept. of Electrical and Computer Engineering, Sungkyunkwan University

요 약

유비쿼터스 컴퓨팅은 일상생활 속에 편재해 있는 PDA 또는 모바일 폰 등의 무선 단말기를 이용하여 사용자가 언제, 어디서나 유용한 서비스를 받을 수 있는 환경을 제공한다. 이는 대용량 데이터베이스에 저장된 지능형 멀티 모바일 에이전트의 통신 데이터를 분석하여 모바일 유저의 위치에 따른 요청된 유용한 서비스정보를 추출할 수 있게 되었으며, 이를 통한 효율적인 사용자 서비스는 물론 광고 등의 새로운 이익 창출로 이어져왔다. 그러나 기존 위치정보만을 이용한 서비스정보의 추출은 단순히 통계적인 빈발 행동패턴만을 추출하여 시간에 따른 사용자의 서비스 요청에 능동적으로 대처할 수 없을 뿐만 아니라 원치 않는 서비스정보를 제공하는 문제점을 야기 시켰다. 이 논문에서는 시간을 고려한 모바일 사용자의 유용한 행동패턴 추출을 위한 효율적인 마이닝 기법의 시간대별 모바일 사용자 행동패턴 및 메모리 적재에 용이한 새로운 콤팩트한 데이터 구조를 제안한다. 이는 사용자의 동적인 움직임에 따른 실시간적 서비스를 가능하게 하며, 더 나아가 유비쿼터스 컴퓨팅 환경에서 중요한 이슈인 데이터의 메모리 적재가 용이 할 뿐만 아니라 접근속도의 향상 및 메모리 사용이 적다는 이점이 있다.

1. 서 론

유비쿼터스 컴퓨팅 기술의 급속한 발전은 모바일 에이전트간의 자유로운 통신을 가능케 하였으며, 사용자는 일상생활 속에 편재해 있는 PDA 또는 모바일 폰 등의 무선 단말기를 이용하여 언제, 어디서나 유용한 서비스를 받을 수 있게 되었다[1]. 이는 곧 인간 삶에 질적인 양상을 가져왔으며, 사용자를 컴퓨팅의 직접적인 운영으로부터 해방시키는 지능형 환경의 결함으로 이어졌다.

최근에는 대용량 데이터베이스에 저장된 광대한 정보를 분류, 유용한 정보를 추출하는 연구가 활발히 이뤄져 왔다. 특히, 사용자에게 유용한 정보를 어떻게 효율적으로 제공할 것인가의 문제는 유비쿼터스 컴퓨팅 기술 중 중요한 문제로 대두되고 있다. 지능형 멀티 모바일 에이전트간의 통신은 반구조적인 데이터, 즉 계층적인 데이터 구조를 가지는 상황인식데이터를 이용하여 사용자의 위치에 따른 서비스된 행동패턴을 추출할 수 있게 되었다. 그러나 종래의 위치정보와 서비스 정보만을 고려한 사용자 행동패턴 추출방법은 단순히 통계적인 사용자의 빈발 행동패턴만을 추출하였다. 따라서, 사용자의 동적인 변화, 즉 시간에 따른 사용자의 서비스 요청에 능동적으로 대처할 수 없는 문제점이 발견되었다. 또한 반구조적 데이터 처리를 위한 효율적이며, 유비쿼터스 컴퓨팅 환경의 메모리적재에 용이한 새로운 데이터구조의 필요성이 절실히 제기 되어왔다.

본 논문에서는 종래의 문제점을 해결하기 위하여 모바일 사용자의 시간대별 행동패턴을 트리 구조로 변환한 후 빈발 트리 패스를 추출하는 새로운 마이닝 기법의 시간대별 모바일

사용자 행동패턴 및 반구조적 데이터 처리와 유비쿼터스 컴퓨팅 환경의 메모리적재에 용이한 트리기반의 새로운 데이터 구조를 제안한다. 이는 아직 도입단계인 트리 기반 마이닝에 진일보한 방식을 제공하며 유비쿼터스 컴퓨팅 환경에서 상황인식 데이터를 대상으로 효율적이며 확장성이 용이한 상황인식 처리를 가능하게 한다. 또한 의사결정시스템, 웹 서비스의 매치메이킹 등의 발전을 도모하여 현재 대량으로 컴퓨팅 환경에 존재하는 반구조적 데이터들에 대한 효율적이고 정확성이 높은 마이닝기법에 응용될 수 있는 기반 기술이 된다. 또한 지능형 유비쿼터스 전용 미들웨어에 적용하여 서비스디스커버리 및 사용자의 요구사항이나 주변상황정보를 이용한 솔루션 제공에 있어서 보다 향상된 정확성을 제공하며 최적의 서비스 제공을 가능하게 한다.

본 논문의 구성은 다음과 같다. 2절에서는 관련된 연구분야에 대해서 기술하고, 전반적인 시스템 구성에 대해서 3절에서 설명하며, 새로운 데이터 구조와 데이터 마이닝 기법을 위한 문제정의는 4절에서 기술한다. 5절에서는 제안된 새로운 데이터 구조와 시간대별 모바일 사용자 행동패턴 기법에 대해서 설명하며, 6절에서는 성능평가 및 실험결과를 기술한다, 7 절에서는 전체적인 고찰 및 발전된 연구를 위한 향후 과제를 제시함으로써 결론을 맺는다.

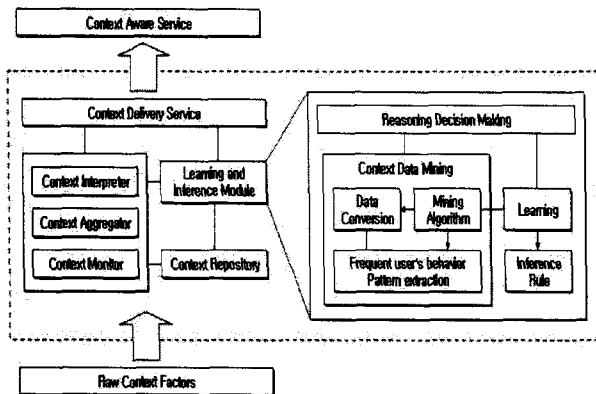
2. 관련연구

시간 데이터로부터 의미있는 지식을 탐사하기 위한 시간 데이터마이닝 기법에 대한 여러 연구가 진행되었다[1,2,3,4,5]. 시간 데이터마이닝은 방대한 양의 누적된 서로 상이한 시간

데이터로부터 이전에 잘 알려지지 않았지만, 묵시적이고 잠재적으로 유용한 시간 지식을 추출하는 기술로 정의된다. 현재까지 시간 데이터로부터 시간 지식을 탐사하기 위한 시간 데이터마이닝은 크게 시간규칙 탐사, 시퀀스마이닝, 경향분석 등으로 분류된다. 시간규칙탐사 기법은 기존 데이터마이닝 기법에서 정의한 규칙을 확장하여 시간관계 및 상호연관 관계를 포함한 시간규칙을 찾는 기법으로, 여기에 속한 기법으로는 주기적으로 반복되는 연관규칙[7]을 탐사하는 주기적 연관 규칙 탐사, 달력으로 표현된 시간 패턴을 가지는 연관규칙을 탐사하는 달력 연관규칙 탐사[1,2] 등이 있다. 이러한 연구들은 시간 데이터를 대상으로 수행되었으며 시간에 따른 상황 데이터 처리는 거의 고려하지 않았다. 이 논문에서는 시간적 데이터를 이용한 보다 효율적인 데이터 추출 방법과 유비쿼터스 컴퓨팅 기술을 시간적 데이터와 연계하여 시간적 데이터의 응용범위를 확대한다.

3. 시스템 구성

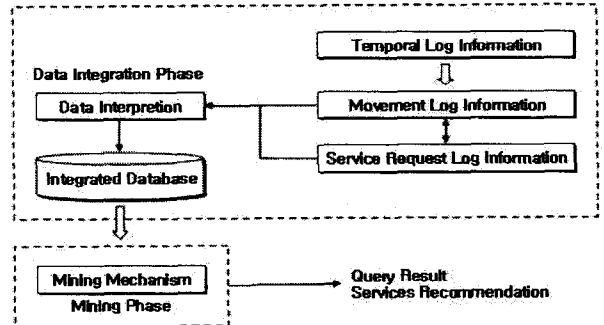
본 논문에서 제안한 새로운 데이터 구조와 데이터 마이닝 기법의 시스템 구성은 [그림 1]을 통해 나타내고 있다. [그림 1]은 J. paik et al.이 제안한 미들웨어 CALM[6]을 기반으로 실제 마이닝 모듈과의 연관관계를 나타내고 있다. 이는 미들웨어의 Learning 모듈과 Inference 모듈과 연계하여 미들웨어의 상황인식 의사결정 시스템에 정확하고, 확장성이 높은 의사결정을 할 수 있도록 한다. 또한 사용자가 언제 어디서나 원하는 정보를 미들웨어에 연결된 마이닝 모듈을 이용하여 추출 후 미들웨어에 연결된 지능형 멀티 모바일 에이전트를 통해 효율적인 정보 제공을 한다. 마이닝 모듈은 본 논문에서 제안하는 데이터 구조와 마이닝 기법을 사용하며, 이를 통해 모바일 사용자의 시간에 따른 유용한 행동패턴 정보를 제공할 수 있다.



[그림 1] 미들웨어 플랫폼과 데이터 마이닝 모듈의 연관 구조

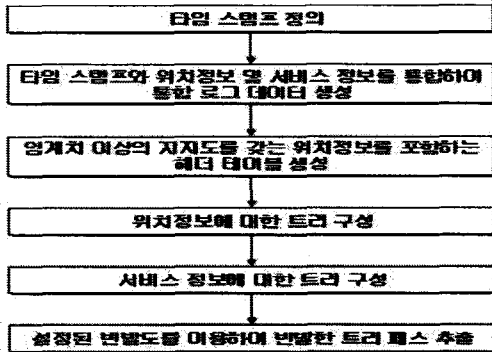
[그림 2]는 세부적인 데이터 마이닝 모듈의 시스템 구성을 나타내고 있다. 데이터 마이닝 모듈에서의 데이터 마이닝은 두 단계를 거친다. 첫째, 데이터 통합 단계이다. 이

단계에서는 위치에 따른 빈발하게 요구된 서비스 정보를 추출한다. 또한 시간에 따른 위치정보와 서비스정보를 고려하여 실시간 정보 처리에 용이하도록 한다. 둘째, 데이터 마이닝 단계에서는 본 논문에서 제안하는 데이터구조 및 마이닝 기법을 이용하여 효율적이며 유용한 사용자의 행동패턴을 추출 한다.



[그림 2] 데이터 마이닝 모듈의 세부 구성도

[그림 3]은 본 논문에서 제안된 시간에 따른 모바일 사용자의 빈발한 행동패턴을 추출하는 과정의 흐름도이다. 시간에 따른 모바일 사용자의 행동패턴을 추출하기 위한 마이닝 과정을 수행하기 전에 전처리 과정으로 타임 스탬프를 정의한다. 이는 원하는 시간대별로 원하는 정보를 추출하기 위한 것으로, 하루를 일정 시간 단위로 구분하여 정의하였다. 타임 스탬프가 정의되면, 정의된 시간대별로 위치정보 및 서비스 정보를 통합하여 하나의 통합된 로드 데이터를 생성한다. 이때, 통합 로그 데이터는 사용자 ID, 타임 스탬프 및 행동패턴으로 구성되며, 행동패턴은 다시 위치정보와 그에 따른 서비스 정보로 구성된다. 서비스 정보는 관리자 또는 사용자에 의해 미리 설정된 값을 부여하며, 시퀀스한 정보를 갖는다. 통합 로그 데이터는 통합 데이터베이스에 저장된다. 이후, 통합 데이터베이스에 저장된 통합 로그 데이터를 콤팩트한 트리 구조로 변환하는 과정이 수행된다. 통합 로그 데이터의 각 행동패턴으로부터 헤더 테이블을 추출하며, 추출되어지는 정보는 임계치 이상의 지지도를 갖는 위치정보를 추출함으로써 생성된다. 헤더 테이블이 생성되면, 이를 기반으로 하여 위치정보에 대한 트리를 구성한다. 위치정보에 대한 트리는 루트에 널(Null) 값을 부여한 후, 각 사용자 ID의 타임 스탬프별로 행동패턴에 포함된 위치정보를 각기 노드로 표시 및 연결함으로써 생성된다. 서비스 정보에 대한 트리 역시, 루트에 널(Null) 값을 부여한 후, 통합 로그 데이터의 행동패턴에 포함된 서비스 정보들로 노드를 구성 및 연결함으로써 생성된다. 트리 구성이 완성되면, 생성된 트리 중 빈발한 패스를 추출한다. 이때, 빈발한 패스를 추출하는 기준은 관리자 또는 사용자에 의해 미리 설정된 지지도를 기준으로 하며, 타임 스탬프의 빈발도를 만족하는 트리 패스만을 추출한다. 즉, 시간대별 지지도를 만족하는 트리 패스를 추출한다.



[그림 3] 시스템 흐름도

4. 문제 정의

이 절에서는 모바일 사용자의 시간에 따른 동적인 움직임을 고려하여 새롭게 문제 정의를 한 후, 이를 기반으로 통합 데이터베이스에서의 타임 스탬프 및 통합 로그 데이터 집합을 정의한다.

4.1. 시간을 고려한 문제 정의

통합 데이터베이스에 시간에 따른 위치 정보 및 서비스된 정보를 저장한다. 이때 두 집합으로 저장되며, 집합 L과 S는 각각 타임 스탬프를 고려하여 위치 정보 집합과 서비스된 정보 집합으로 나타낸다. 각 구성요소 L과 S에 속한 l, s 는 순서쌍 $f = (l, s)$ 로 표현한다. 모든 순서쌍의 집합은 F로 나타내며 다음과 같이 표시한다.

$$F = L \times S = \{all\ pairs\ of\ (l : s)^t \mid l \in L\ and\ s \in S\}$$

두 구성요소 l^t 와 s^t 는 f 의 처음과 두 번째 구성요소에 해당되며, 만약 $l_1^t = l_2^t$ 와 $s_1^t = s_2^t$ 가 같다면, 두 개의 순서쌍 $(l_1, s_1)^t$ 와 $(l_2, s_2)^t$ 도 같다.

$$T = \sum_{i=1}^n \left(\sum_{j=1}^n f_j \right)^{t_i}$$

는 타임 스탬프 t 에서 f_j 는 위치

정보와 요청된 서비스 정보의 집합을 나타내며, 순서쌍 f 의 길이가 n 인 시퀀스를 표현하고 있다. 타임 스탬프 t_i 동안 동적인 움직임을 갖더라도 행동패턴의 시간적 순서에 따라 정렬된다. 그 이유는 모바일 사용자가 한 번에 다른 일을 처리할 수 없는 점에 착안하여 각각의 행동 패턴은 유니크하지는 않지만 동일 시간대에는 유니크한 순서를 갖는다. T에서 순서쌍 f_j 를 고려하여 시간대별 모바일 사용자 행동패턴 $T_p = \langle (l_1, l_2, \dots, l_n) \rangle$ 을 정의 할 수 있다. 통합 데이터베이스 $D = \{T_p^1, T_p^2, \dots, T_p^n\}$ 는 n 개의 행동패턴을 가지며, 행동패턴 T_p 의 지지도는 다음과 같이 정의 할 수 있다.

$$Sup(T_p) = \frac{|\{T_p^i \cup T_p^j \mid T_p^i \subset T_p, T_p^j \subset T_p\ and\ 1 \leq i, j \leq n\}|}{n}$$

접근 패턴 $Sup(T_p)$ 는 빈발 시간대별 모바일 접근 패턴으로써 n 은 전체 통합 데이터베이스의 통합 로그 데이터를 나타내며, T_p^i, T_p^j 는 각각 시간에 따른 위치 정보와 서비스 정보로 표현된다. 전체 통합 데이터베이스의 통합 로그 데이터에서 시간대별로 나타나는 위치에 따른 서비스 정보를 추출하여 사용자 또는 관리자가 미리 설정한 임계치 이상일 경우 빈발하다고 정의한다.

4.2. 타임 스탬프 및 통합 로그 데이터 집합

본 하부 절에서는 타임 스탬프를 관리자나 사용자로 하여금 미리 정의하게 하여, 원하는 시간대별로 서비스를 제공하거나 받을 수 있도록 한다. [표 1]은 타임 스탬프 T의 시간간격을 4시간 단위로 구분하여 미리 정의 하였으며, 총 6개의 타임 스탬프를 갖는다.

[표 1] 미리 정의된 타임 스탬프

타임 스탬프	시간 간격
T1	06:00 ~ 10:00
T2	10:00 ~ 14:00
T3	14:00 ~ 18:00
T4	18:00 ~ 22:00
T5	22:00 ~ 02:00
T6	02:00 ~ 06:00

전처리 과정에서 정의된 타임 스탬프에서 시간대별 모바일 사용자의 위치정보와 요청된 서비스 정보를 통합하여 하나의 통합된 로그 데이터를 생성을 [표 2]에서 보여주고 있다. 통합된 로그 데이터는 사용자 ID, 타임스탬프 및 행동패턴으로 구성되며, 이 중 타임 스탬프는 전처리 과정에서 미리 정의된 타임 스탬프를 사용하고, 행동패턴은 위치 정보에 따른 요청된 서비스 정보를 포함하도록 한다. 구체적으로, [표 2]에서, 사용자 ID 100번의 T1이라는 시간대에 <A;1;B;2;C;5, D;8> 정보가 축적되어 있는데, 여기서 A는 위치정보(l)를 나타내고 숫자 1은 서비스 정보(s)를 나타낸다. 서비스 정보는 관리자 또는 사용자가 사전에 임의로 설정한 값이다. 이때, 각각의 위치정보에 따른 요청된 서비스 정보는 시퀀스한 정보를 가지며, 이는 모바일 사용자가 한 번에 다른 일을 처리할 수 없는 점에 착안하여 각각의 행동 패턴은 유니크하지는 않지만 동일 시간대에는 유니크한 순서를 갖는다. 이와 같이 전처리 과정에서 정의된 타임 스탬프를 이용하여 정의된 시간대별 위치정보와 요청된 서비스 정보를 통합하여 하나의 통합된 로그 데이터를 생성할 수 있으며, 하나의 통합된 로그 데이터를 통합 데이터베이스에 저장함으로써 정보의 개별화로 인한 빠른 접근성의 한계와 검색 시간의 지연과 같은 문제점을 해결할 수 있는 장점을 가진다.

[표 2] 통합 데이터베이스에 저장된 로그 데이터 집합

유저 ID	타임 스탬프	행동 패턴
100	T1	< A:1, B:2, C:5, D:8 >
	T2	< A:1, D:1, E:2 >
	T3	< A:2, B:2, E:1 >
	T4	< C:4, F:2 >
	T5	< C:2, B:2 >
	T6	< A:2, B:2, E:1 >
101	T1	< A:1, B:3, C:5, D:8 >
	T2	< B:1, D:1 >
	T3	< C:1, E:3 >
	T4	< C:4, E:3 >
	T5	< B:2, D:1 >
	T6	< A:2, B:2, E:1 >
102	T1	< A:3, B:6, C:8, D:7 >
	T2	< B:2, C:6, D:7 >
	T3	< A:2, B:2, E:1 >
	T4	< E:3, G:7 >
	T5	< E:1, D:2 >
	T6	< A:1, D:1, E:2 >

5. 효율적인 마이닝 기법: 시간대별 모바일 사용자 행동패턴

이 절에서는 트리 기반의 새로운 데이터 구조를 제안하며, 구성된 트리 구조를 통해 효율적인 시간대별 모바일 사용자의 행동패턴을 추출한다.

5.1. 트리 기반의 데이터 구조를 위한 헤더 테이블 구성

통합 데이터베이스에 축적된 정보를 바탕으로 최소 지지도(Min_Sup) 이상을 만족하는 위치정보를 추출한 후, 이를 바탕으로 트리 기반의 데이터 구조를 구성시킨다. 먼저 [표 3] (a)에 도시한 바와 같이, [표 2]에서 발생하는 각 위치정보를 시간에 상관없이 추출 및 카운트한 정보를 생성하고, 각 위치정보 중 지지도, 즉 카운트가 미리 설정된 임계치 이상인 위치정보만을 추출하여 [표 3] (b)와 같은 헤더 테이블을 생성한다. 이 때 [표 3] (b)는 지지도가 2 이상인 위치정보를 추출하였다. 이 경우, F와 G는 지지도가 1로 빈발하지 않은 것으로 판단하여 제외 시킨다. 이와 같이 지지도 이상의 위치정보만을 추출하는 것은 빈발하지 않은 위치정보에서는 서비스 정보 역시 빈발하지 않기 때문이다. 이 때, 통합 데이터베이스 스캔이 한 번 이뤄진다.

[표 3] 트리 구성을 위한 헤더 테이블

A	9
B	11
C	8
D	9
E	10
F	1
G	1

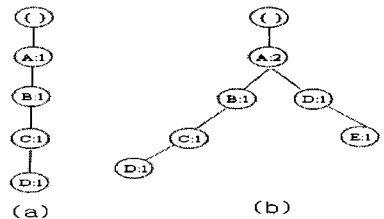
(a)

위치 정보	지지도
A	9
B	11
C	8
D	9
E	10

(b)

5.2. 시간대별 모바일 접근 패턴 트리 구성

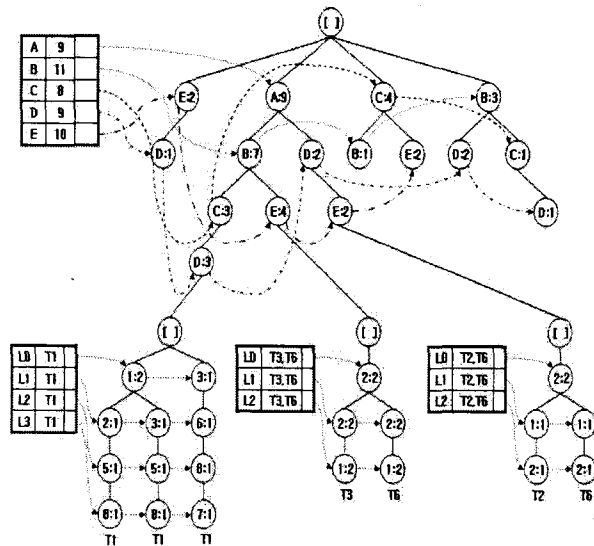
[표 3] (b)와 같이 빈발하는 위치정보를 기반으로 생성된 헤더 테이블을 바탕으로 트리를 구성한다. A, B, C, D 및 E는 지지도 이상을 가지는 위치 정보이며, 헤더 테이블에 정보가 입력되어있다. 트리를 구성하기 위해서 통합 데이터베이스를 마지막 스캔한다. 따라서 통합 데이터베이스를 스캔하는데 필요한 횟수는 단지 2번으로써, 스캔에 필요한 시간을 효과적으로 단축 할 수 있다. 통합 데이터베이스를 스캔 후 다음과 같은 몇 단계를 거쳐 시간대별 모바일 사용자 행동패턴 트리를 구성한다. I) 루트를 Null 값으로 생성한다. II) [표 2]의 통합 데이터베이스에서 유저 ID 100의 T1에서 행동패턴 중 위치 정보인 <A, B, C, D>를 트리로 구성한다. 이때, A,B,C 및 D는 시퀀스 정보로 순서를 갖는다. 이는 모바일 사용자가 동일 시간에 동일한 일을 동시에 할 수 없기 때문이다. [그림 4] (a)는 위의 과정을 통해 완성된 트리 가지이다. III) [그림 4] (b)와 같이 유저 ID 100의 T2에서의 위치정보 <A, D, E>에 대하여 동일한 방법으로 트리를 구성한다. 이때, A는 이미 트리에 생성되어 있으므로 II)에서 생성된 트리의 A에 카운트 1을 올려주고, D는 트리에 없는 정보이므로 새로 가지를 생성한다. IV) 상기와 같은 방법으로 유저 ID 100에 대해서 T6까지 트리를 구성한 후, 나머지 유저 ID 101과 102에 대하여도 동일한 방법으로 트리를 구성한다. 이때, 유저 ID 100의 T4에서의 위치정보 <C, F>와 같이 지지도에 의해 [표 3] (b)의 헤더 테이블에서 제거된 위치정보가 포함된 경우, 위치정보 C는 헤더 테이블에 포함된 위치 정보이므로, 트리에서 카운트를 올려주고, 위치정보 F는 헤더 테이블에 포함되지 않은 위치 정보이므로 노드를 생성하지 않는다.



[그림 4] 트리 구조로 변환하는 행동패턴 과정

이와 같이 위치정보에 대한 트리가 완성되면, 서비스 정보에 대한 트리를 생성한다. 서비스 정보 트리는 위치 정보의 단말노드에 링크를 연결시켜 생성한다. 연결된 링크를 통해 위치에 따른 서비스 정보의 파악이 이뤄질 수 있다. 위치 정보의 헤더 테이블처럼 서비스 정보 트리에서도 헤더 테이블이 존재한다. 서비스 정보의 헤더 테이블은 해당 트리의 패스에 대한 타임 스탬프 및 트리의 레벨이

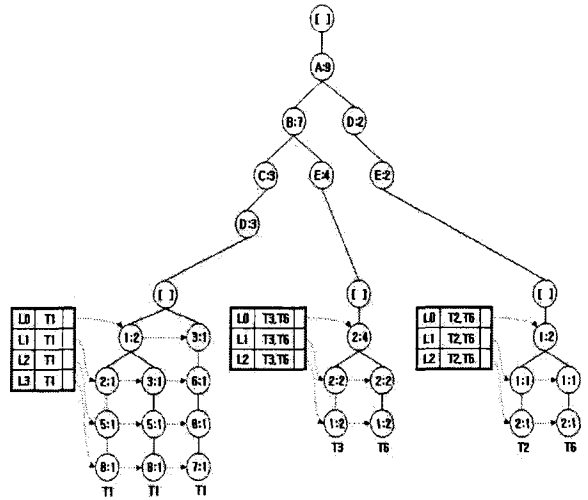
기록되어있다. V) 위치 정보의 단말노드의 링크를 통해 연결되는 서비스 정보 트리는 각 연결된 트리의 가지마다 루트를 Null값으로 생성한다. VI) 루트를 Null 값으로 생성하고, 통합 데이터베이스에서 유저 ID 100 T1 <A:9, B:7, C:3, D:3>의 행동패턴에 대하여 위치 정보 트리를 기반으로 하여 서비스 정보를 트리로 구성한다. 서비스 정보 트리 <1, 2, 5, 8>를 생성하고 카운트를 기록한다. [그림 5]는 통합 데이터베이스의 행동패턴 <A:1, B:2, C:5, D:8>, <A:1, B:3, C:5, D:8>, <A:3, B:6, C:8, D:7>, <A:2, B:2, E:1>, <A:2, B:2, E:1>, <A:2, D:1, E:2> 그리고 <A:2, D:1, E:2>의 시간대별 모바일 사용자 행동패턴 트리를 구성하였다. 트리의 (1:2, 2:1, 5:1, 8:1)에서 앞의 숫자는 서비스 정보를 나타내고, 뒤의 숫자는 카운트 정보를 나타내며, 서비스 정보의 헤더 테이블의 각 타임 스탬프와 서비스 정보 트리의 각 노드를 연결한다. VII) 이와 같은 방법으로 나머지 테이블 노드에 대한 서비스 정보의 트리를 생성한다.



[그림 5] 시간대별 모바일 사용자 행동패턴 트리

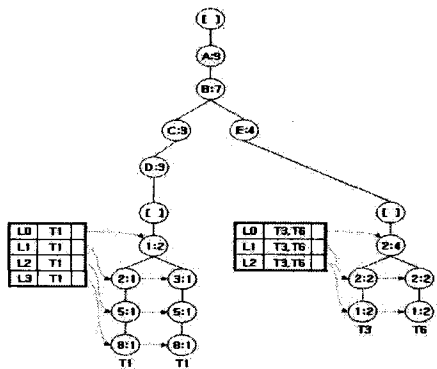
5.3. 시간대별 모바일 접근 패턴 마이닝

[그림 5]와 같이 위치 정보 및 서비스 정보에 대한 트리가 생성되면, 시간에 따른 모바일 사용자의 유용한 행동패턴을 추출하기 위하여 생성된 트리 중 빈발하는 싱글 패스를 추출한다. [그림 6] 및 [그림 7]은 시간대별 모바일 사용자의 행동패턴 트리로부터 빈발도가 2 이상인 트리 패스를 추출함으로써, 사용자의 시간에 따른 유용한 정보를 모두 찾아낼 수 있다. 그 예로, [그림 6]은 위치 정보 <A, B, C, D, E>에 대한 트리로서 위치 정보의 빈발도가 2 이상인 트리 패스만을 구성하였다. 위치 정보에 대한 빈발도를 검사한 후, 빈발한 위치 정보에 따른 서비스 정보의 빈발도를 검사한다.



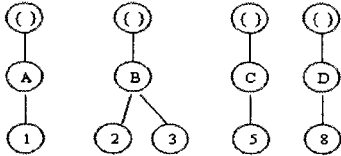
[그림 6] 위치 정보 <A, B, C, D, E>에 대한 트리 구성

[그림 6]에서 위치 정보 <A, B, C, D, E>에 대해서 3개의 서비스정보 트리가 생성되었으며, 7개의 트리 가지가 구성되었다. 기존의 통계학적 방법의 빈발도 체크에서는 [그림 6]의 가장 오른쪽에 해당하는 위치정보 <A, D, E>에 대해서 빈발하다고 본다. 그러나 시간 개념을 도입한 시간대별 모바일 사용자 행동패턴에서는 타임 스탬프 정보에 기준을 두어, 비록 같은 위치에서 같은 서비스를 제공 받았더라도 지지도 이하이므로 가치치기를 한다. [그림 7]은 타임 스탬프 기준으로 지지도 2 이상인 모든 트리 패스를 추출하였다. 타임 스탬프 T1에서 <A, B, C, D>의 위치에 따라 빈발하게 서비스된 정보는 <1, 2, 5, 8> 및 <1, 3, 5, 8>이 된다. 또한 T3의 <A, B, E>에서는 <2, 2, 1>에 대한 서비스가 빈발하고, T6에서는 위치 정보 <A, B, E>에 따른 서비스 정보 <2, 2, 1>이 빈발함을 알 수 있다. 이와 같은 방법으로 모든 시간대별 유용한 행동패턴을 구할 수 있으며, 모바일 사용자는 시간에 따른 올바른 서비스 정보를 제공 받을 수 있게 된다.



[그림 7] 빈발한 시간대별 모바일 사용자 행동패턴

타임 스탬프 기준으로 임계치 이상인 모든 트리 패스를 추출한 후, 최종적으로 실제 모바일 사용자에게 제공할 정보를 마이닝 한다. [그림 8]은 타임 스탬프 T1에 대한 위치별 유용한 서비스 정보의 싱글 패스 트리를 모두 추출한 것이다. 이로써, T1이라는 시간에 어느 위치에 모바일 사용자가 있더라도 정확하고, 유용한 정보를 제공할 수 있다.



[그림 8] 타임스탬프 T1에 대한 위치별 유용한 서비스 싱글 패스 트리

위와 동일한 방법으로 모든 타임 스탬프의 위치별 유용한 서비스를 싱글 트리 패스로 추출한다.

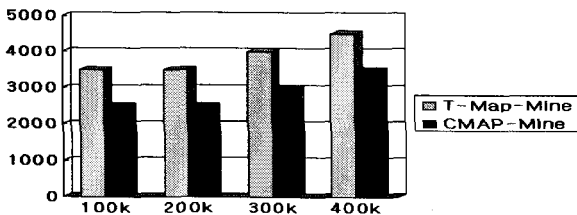
6. 성능평가

본 논문에서 수행된 모든 실험은 2.4GHz 펜티엄 4 1GB 메모리의 PC에서 수행되었으며, OS는 마이크로 소프트의 Windows XP를 사용하였다. [표 4]는 성능평가의 시뮬레이션 모델을 위한 파라미터이다.

[표 4] 시뮬레이션 모델을 위한 기본적인 파라미터

Parameter	Description	Default Value
U	The number of mobile Users	1000,000
E	The number of Mobility events	500
W	Nodes of Network W*W	5
N	The number of different services	100,000
Ua	User alive time units	20
Tc	The time units of transaction cut time	20
Hs	The number of popular services per node	10

성능평가는 기존 튜플기반(Apriori-Algorithm[7])에서 위치 정보를 기반으로 서비스 정보를 추출해내는 방법인 CMAP - Mine과 본 논문에서 제안한 시간대별 모바일 사용자 행동패턴 기법(이하 T-Map-Mine)을 비교, 분석 하였다.



[그림 9] |U|변화에 대한 빈발패턴 추출 속도

[그림 9]에서 보여주는 것처럼 기존 튜플 기반의 마이닝

기법에서 트리 기반의 마이닝 기법으로 변환한 후 빈발패턴 추출의 속도가 향상됨을 알 수 있다. 이는 여러 후보 항목을 만들어 내는 기존 방법에서 벗어나 하나의 트리를 구성한 후 모든 빈발 패턴 트리를 추출하기 때문이다.

7. 결론 및 향후 연구 과제

본 논문에서 제안한 시간대별 모바일 사용자 행동패턴은 시간에 따른 모바일 사용자의 빈발한 행동패턴 추출방법으로써, 모바일 에이전트 간의 통신인 반구조적인 데이터 형식에 용이한 트리 구조로 변환한 후 빈발한 트리 패스를 추출한다. 이는 종래의 위치 정보와 서비스 정보만을 이용한 사용자 행동패턴 추출에서 벗어나 보다 능동적이며, 실시간적인 서비스를 사용자에게 제공할 수 있게 되었다.

향후 연구에서는 실제 에이전트간의 데이터 언어인 ACL을 이용, 본 논문에서 제안한 마이닝 기법을 응용해야 할 것이다.

감사의 글

본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅 및 네트워크 원천기반 기술개발사업의 지원에 의한 것임.

참고 문헌

[1] C. Y Chang, M.S Chen, "Integrating web caching and web prefetching in client-side proxies", in: Proceedings of the ACM 11th International conference Information and knowledge Management, 2002.

[2] Dan Lim, "Ubiquitous mobile Computing:UMC's model and success", Educational Technology & society 2(4) 1999 ISSN 1436-4522.

[3] J. F. Roddick, K. Hornsby and M. Spiliopoulou, Temporal Spatial and Spatio-Temporal data mining and knowledge discovery research Bibliography. <http://www.cs.flinders.edu.au2000>.

[4] J. F. Roddick and M. Spiliopoulou, "Temporal data mining:survey and issues," Research Report ACRC-99-007, University of south Australia, 1999.

[5] J. L.Chen, Resource allocation for cellular data services using multiagent schemes, IEEE Trans. Syst. Man Cybern. 31 (6) (2001) 864-869.

[6] J. Paik, R. Dong and M. kim. "XML-Encoded Context Data Mining for Ubiquitous Middleware Platforms." GESTS International Transactions on Computer Science and Engineering.

[7] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules." The VLDB conference, Santiago, chile, September 1994.