

# 스마트 홈 환경에서 데이터 마이닝 기법을 이용한 지능형 서비스 추론 모델

준회원 강명석\*, 정회원 김학배\*\*

## Intelligent Service Reasoning Model Using Data Mining In Smart Home Environments

Myung-seok Kang\* Associate Member, Hag-bae Kim\*\* Regular Member

### 요약

본 논문에서는 스마트 홈 환경에서 데이터 마이닝 기법을 이용하여 사용자에게 상황에 적합한 서비스를 추론하는 모델을 제안한다. 의사결정트리 알고리즘들 중에 하나인 C4.5 알고리즘을 기반으로 서비스 추론에 쓰이는 서비스 트리를 생성하고, 정량적 특성 규칙과 정량적 판별 규칙을 이용하는 정량적 가중치 산정 알고리즘을 통해 사용자에게 제공될 서비스를 추론한다. 또한 시뮬레이션을 통해 그 성능을 검증하였다.

**Key Words** : ISR (Intelligent Service Reasoning), Data Mining, Smart Home

### ABSTRACT

In this paper, we propose a Intelligent Service Reasoning (ISR) model using data mining in smart home environments. Our model creates a service tree used for service reasoning on the basis of C4.5 algorithm, one of decision tree algorithms, and reasons service that will be offered to users through quantitative weight estimation algorithm that uses quantitative characteristic rule and quantitative discriminant rule. The effectiveness in the performance of the developed model is validated through a smart home-network simulation.

### I. 서론

홈오토메이션은 방문자 확인 통화와 문 개방 및 기초적인 수준의 무인 경비 기능을 제공하는 비디오 폰을 장착한 주택 수준으로 최근까지 주택건설 업체 중심으로 확산되어 왔다. 최근에는 좀 더 확장된 개념으로 네트워크화 된 가전기기 및 장치를 인터넷이나 휴대폰을 통해 감시 또는 제어하는 기본적인 홈 네트워크 시스템을 구축하고 있다. 이러한 홈 네트워크 시스템은 전력선 통신, 블루투스, ZigBee, WirelessLan, UWB (Ultra Wide Band),

IEEE1394 등 홈 네트워크를 총괄하면서 외부 인터넷 망과 연계되는 홈 게이트웨이 또는 홈 서버가 점차 보급되고 있다.

향후에는 홈 네트워크 시스템에 사용자와 시스템 간 상호 반응하며 서비스를 제공하는 지능형 서비스가 제공되어야 한다. 이러한 가정환경을 스마트 홈 환경이라고 한다. 이러한 환경에 따른 사용자 중심의 능동적 서비스를 제공하기 위해 홈 네트워크 시스템에 Markov Chain, Neural network, Bayesian 등 다양한 알고리즘을 통한 예측 및 추론 방법들이 적용이 시도되고 있다. 앞으로 홈 네트워크 시스템

\* 연세대학교 전기전자공학과 디지털정보처리 연구실(mskang@yonsei.ac.kr)

\*\* 연세대학교 전기전자공학과 디지털정보처리 연구실(hbkim@yonsei.ac.kr)

논문번호 : KICS2007-04-187, 접수일자 : 2007년 4월 23일, 최종논문접수일자 : 2007년 11월 27일

에는 점차 다양하고 복잡한 기기들이 연결되며, 사용자의 기대도 또한 증대할 것으로 보인다. 이러한 변화들에 반응하기 위해서는 가정환경에 맞게 낮은 시스템 성능에서도 많은 서비스 데이터들을 신속하게 처리하여 즉각적으로 사용자에게 반응할 수 있는 모델이 필요하다. 따라서 본 논문에서는 이미 비즈니스 분야에서 대량의 데이터에서 사용자가 원하는 지식을 찾고 이를 분석하여 다음 값을 예측하는 데이터 마이닝 기법<sup>[1-3]</sup>을 홈 네트워크에 적용한다면 이러한 변화에 적용할 수 있을 것으로 판단된다. 따라서 본 연구에서는 이러한 스마트 홈 환경에서 데이터 마이닝 기법을 이용하여 서비스 추천하여 사용자에게 지능형 서비스를 제공하는 모델을 제시하는 것이다.

## II. ISR 모델

ISR 모델은 스마트 홈 환경에서 데이터 마이닝 기법 중 의사결정트리 알고리즘인 C4.5를 기반으로 사용자의 생활 패턴을 학습하여 서비스 모드 또는 단일 서비스를 추천하여 그에 따른 기기를 제공하는 모델이다. 본 모델에서는 그림 1과 같이 DB영역의 데이터를 처리하는 Service Manager에서 센싱된 사용자의 생활 데이터를 context 데이터로 변환하는 context 분석 모듈(context analyst)과 context 데이터를 이용하여 서비스를 추천하는 서비스 제공 모듈(service provider)로 구분되는 2단계의 절차를 통해 서비스되며 단계별 의미는 아래와 같다.

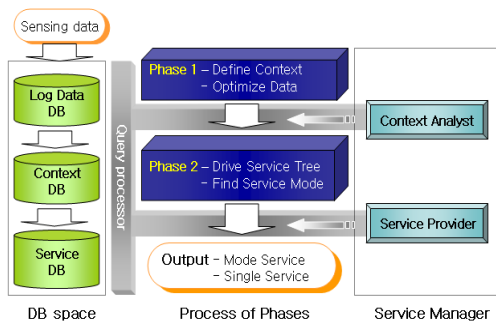


그림 1. ISR 모델

### 2.1 Context 분석 모듈(Context analyst)

사용자의 입력 데이터, 즉 사용자가 특정 기기를 사용한 로그데이터가 로그 데이터베이스에 기록이 된다. 이러한 로그 데이터를 지능형 서비스 추천

(ISR) 모델에서 처리하기 위해 전처리 과정을 거치는데 이것은 Service Manager의 Context Analyst가 수행하게 된다. 전처리 과정에서는 context 정의와 데이터 최적화 과정을 거친다. 첫 번째 과정인 context 정의는 로그 데이터를 사전에 정의된 Context 데이터베이스내 테이블별 속성 및 속성 범주에 따라 데이터를 변환하며, 각각 하나의 context를 생성하게 된다. 두 번째 과정인 데이터 최적화는 로그 데이터를 Context로 변환하기 전에 로그 데이터 중 불필요한 데이터는 삭제(데이터 중복배제), 데이터 변화 등의 과정을 거치게 된다. 이러한 전처리 과정을 통해 서비스 추천에 필요한 데이터만을 추출하여 처리하기 때문에 시스템에 불필요한 자원 낭비를 줄임으로서 시스템 효율성을 증대하게 된다.

### 2.2 서비스 제공 모듈(Service Provider)

서비스 제공(Service Provider) 모듈에서는 서비스 트리를 생성하고, 생성된 트리의 각 노드별 가중치를 산출하여 최적의 서비스 모드를 선택한다. 선택된 서비스 모드는 그 값을 과거 사용자의 서비스 기록과 비교하여 정확성을 확인 후 최종적으로 사용자에게 지능형 서비스를 제공하게 된다. 각 단계별 과정은 다음과 같다.

#### 2.2.1 서비스 트리 생성

Context Analyst에서 처리된 context 데이터베이스의 각각의 속성별, context별 count정보에 따른 서비스 노드를 결정하며, 결정된 서비스 노드에 따라 서비스 트리를 형성한다. 만들어진 서비스 트리의 중간 트리에는 잎 노드를 결정하는데 필요한 속성들이 들어가며 각 속성 값에 따라 지식노드로 분기되며 최종적으로 분기된 노드, 즉 잎 노드(leaf node)에는 중간 노드의 속성과 속성 값에 따라 서비스 모드가 표현된다. 이러한 서비스 트리를 형성하는 방법은 서비스 트리 생성 알고리즘에 따른다.

#### 2.2.2 노드별 정량적 가중치 산출

서비스 트리 생성 알고리즘에 따라 생성된 트리에서 잎 노드에 있는 서비스 모드(service mode)를 결정하기 위해 루트 노드를 포함한 모든 중간 노드에는 각 속성에 따라 정량적 가중치(weight) 값을 부과하여 각 노드별 가중치 값을 조합하여(곱하여) 최종 가중치 값을 잎 노드에 부여한다. 잎 노드에 부여된 가중치 중 가장 큰 값을 갖는 잎 노드의 서비스 모드를 선택한다. 만약, 최종 가중치 값이 임

계값(limited value), 사용자의 의도가 서비스 모드 인지를 확인하는 기준 가중치 값, 보다 작을 경우에는 서비스 모드가 아닌 싱글 모드(single mode)로 context 데이터베이스에서 단일기기(single device)를 선택하여 사용자에게 제공한다. 정량적 가중치 값을 산출하는 방법은 정량적 특성화 규칙과 정량적 판별 규칙을 조합한 정량적 가중치 산출 규칙에 따른다.

2.2.3 서비스 확인

선택된 서비스 모드는 바로 사용자에게 제공되는 것이 아니라 간단한 확인 절차를 거치게 된다. 해당 서비스 모드에 포함된 기기 목록과 Context 데이터 베이스내 최근까지 사용자가 선택한 기기 정보와 비교하여 일치한다면 선택된 서비스 모드에 따른 기기들을 제공하며, 그렇지 않다면 다시 서비스 트리를 생성하는 과정으로 돌아간다.

2.3 데이터 스키마

ISR 모델에 사용되는 context 데이터베이스는 사용자 테이블, 시간 테이블 그리고 기기 테이블로 구성된 데이터 스키마를 가지고 있다. 각각의 테이블 들은 계층적 구조(그림 2)를 가지고 있다. 각 테이블 별 속성들을 살펴보면 다음과 같다.

2.3.1 사용자 테이블(user table)

사용자 테이블은 사용자를 분류하는 3가지 속성, 즉 나이, 성별, 그리고 이름으로 구분된다. 나이는 범주형데이터로표현하기위해나이에따라노인(old), 성인(adult), 미성년(young)으로 속성 값이 분류된다.

2.3.2 시간 테이블(time table)

시간 테이블은 연도, 월, 요일, 날짜 그리고 시간을 구분된다. 연도, 월, 날짜는 해당 숫자를 그대로 표현하며, 요일 및 시간은 범주형데이터로표현한다. 요일은 주중(weekday)와 주말(weekend)로 구분하며, 시간은 사람이 활동하는 시간대(06~24시)는 1T부터 6T까지 3시간씩 구분하였고, 사람이 활동하지 않는 시간대(00~06시)는 0T로 표현한다.

2.3.3 기기 테이블(device table)

기기 테이블은 각각의 기기들이 설치된 장소(location of device)와 기기의 이름으로 속성이 구분된다. 설치 장소는 일반적인 30평형대 아파트를 기준으로 안방(R1), 작은방(R2), 거실(R3), 주방(CI), 화장실(TO), 현관(FR)으로 구분하였다. 기기에는 가

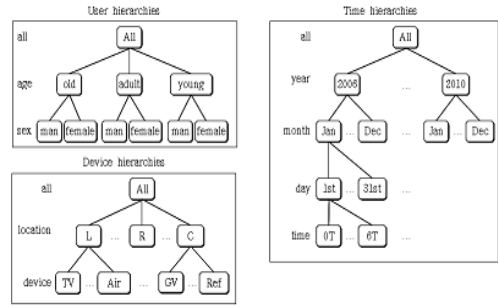


그림 2. 사용자, 기기, 시간 계층적 데이터 구조

정 내 사용하는 일반적인 가전기기들을 표현했으며, 그 종류는 TV(TV), 에어컨(AI), 컴퓨터(CO), 냉장고(RE), DVD플레이어(DV), 전등(LI), 출입문(DR), 커튼(CT), 가스밸브(GV)로 구분하였다. 설치 장소 별 설치된 기기는 계층적 구조를 갖는다. 예를 들어, 거실에는 전등, DVD플레이어, 커튼이 있으며 주방에는 냉장고, 전등, 가스밸브가 설치되어 있다. 따라서 거실에 있는 전등을 표현할 때는 “R3LI”로 표현하며, 주방에 있는 전등을 표현할 때는 “CILI”로 표현한다. 또한, 거실에는 냉장고가 없기 때문에 “R3RE”란 표현은 할 수 없다.

III. ISR 모델의 알고리즘

ISR 모델에서 서비스 제공 모듈(Service provider)에 사용되는 알고리즘, 즉 속성들의 특성에 따라 서비스 트리를 생성하는 알고리즘과 각 콘텍스트에 대한 가중치를 선정하는 알고리즘에 대해 자세히 알아본다.

3.1 서비스 트리 생성 알고리즘

서비스 트리란 사용자의 여러 가지 상황, 즉 사용자가 사용한 기기와 사용 시간등을 고려하여 사용자가 원하는 서비스를 찾아 해당 서비스 기기들을 제공해주는 의사결정트리이다. 의사결정트리는 흐름도와 유사한 트리구조로 중간노드(intermediate node)에는 속성에 대한 검사를 표시하고, 가지(arc)는 검사의 결과를 나타내며, 잎(leaf) 노드는 클래스나 클래스의 분포를 나타낸다. 본 논문에서는 누가 사용하는 지인 사용자 속성과 사용자가 사용한 기기 속성 그리고 기기를 사용한 시간 속성들이 중간노드가 되며, 잎 노드는 그에 따른 서비스 모드가 해당된다. 사용자가 원하는 서비스 모드를 분류하기 위해서는 데이터들의 속성 값들을 의사결정 트리

검사한다. 검사는 루트에서부터 해당 샘플에 대한 클래스의 예측을 가지는 잎 노드까지의 경로를 따라 진행한다. 의사결정트리는 분류 규칙 형식으로 쉽게 변환될 수 있다. 그림 3은 서비스 트리를 생성하는 알고리즘을 간략히 정리한 것이다. 노드에 적용할 속성을 선택하는 방법은 각 속성들의 정보 이득(Information Gain)이란 척도를 이용한다. 이 척도는 속성 선택 척도(attribute selection measure) 또는 분기 적합성 척도(measure of the goodness of split)라고 한다. 가장 큰 정보이득(또는 최대 엔트로피 감소)을 가지는 속성이 현재 노드에 대한 검사 속성으로 선택된다. 이 속성은 분할된 결과로 표본들을 분류하는데 요구되는 정보를 최소화하는데 이러한 분할들이 가지는 무작위성(randomness) 또는 불순도(impurity)를 최소화하도록 한다. 이와 같은 접근법은 객체들을 분류하는데 필요한 예상 검사 횟수를 최소화하여 단순한 트리가 발견되도록 한다.

S를 s개의 샘플 데이터를 가지는 집합이라고 하자. 클래스 레이블(class label)속성은 m개의 상이한 클래스  $C_i(1, 2, \dots, m)$  를 정의하는 m개의 상이한 값을 갖는다고 가정하자. 샘플 S에 있는 클래스  $C_i$ 의 샘플 개수를  $S_i$ 라고 하자. 주어진 샘플을 분류하는데 요구되는 기대 정보량(expected information)은 다음과 같다.

$$I(s_1, s_2, \dots, s_m) = -\sum_{i=1}^m p_i \log_2(p_i) \quad (3.1)$$

여기서  $P_i$ 는 임의의 샘플이 클래스  $C_i$ 에 속할 확률이며  $S_i/S$ 으로 계산된다. 정보가 비트로 부호화되어 있기 때문에 밑수가 2인 로그 함수를 사용한다.

속성 A는 v개의 상이한 값  $\{a_1, a_2, \dots, a_v\}$  을 갖는다고 하자. 속성 A는 S를 v개의 부분집합

$\{S_1, S_2, \dots, S_v\}$  로 분할하는데 사용될 수 있다. 여기에서  $S_j$ 는 A의 값  $a_{ij}$ 를 갖는 S의 샘플들을 포함한다. 만약 A가 검사 속성으로 선택되었다면, 이 부분집합들은 집합 S를 포함하는 노드에서 파생되는 가지들에 대응된다. 부분집합  $S_j$ 에 있는 클래스 C의 샘플 수를  $S_{ij}$ 라고 하자. A에 의해 부분집합으로 분할하는 경우의 엔트로피(entropy) 또는 기대 정보량(expected information)은 아래 식으로 얻어진다.

$$E(A) = \sum_{j=1}^v \frac{S_{1j} + \dots + S_{mj}}{s} I(s_{1j}, \dots, s_{mj}) \quad (3.2)$$

$\frac{S_{1j} + \dots + S_{mj}}{s}$  항은 j번째 부분집합(즉, A의

$a_j$ 의 값을 갖는)의 가중치로 사용되고, 이는 S의 부분집합의 샘플 수를 전체 샘플 수로 나눈 값이다. 엔트로피 값이 작으면 작을수록 분할된 부분집합의 순수도(purity)는 증가한다. 주어진 부분집합  $S_j$ 에 대해서,

$$I(s_{1j} + \dots + s_{mj}) = -\sum_{i=1}^m p_{ij} \log_2(p_{ij}) \quad (3.3)$$

이 되고 여기서  $p_{ij} = \frac{S_{ij}}{|S_j|}$  으로,  $S_j$ 의 샘플이 클래스  $S_j$ 에 속하는 확률이다. A에 대한 분기로 얻게 되는 부호 정보량(encoding information)은 다음과 같다.

$$Gain(A) = I(s_j, s_2, \dots, s_m) - E(A) \quad (3.4)$$

즉 Gain(A)은 속성 A의 값을 알고 있음으로 인해서 비롯되는 엔트로피의 기대 감소량이다. 이와 같은 방법으로 각 속성의 정보 이득을 계산하여 최대 정보량을 갖는 속성이 주어진 집합 S에 대한 검사 속성으로 선택된다. 새 노드가 생성되어 속성 이름이 부여되고, 속성의 각 값들에 따라서 가지가 생성되어, 샘플들은 따라서 분할된다.

```

| Algorithm : Generate service tree
| Input : samples(training data from context DB),
          attribute_list(list of context and non-context attributes name)
| Output : Service tree
| Method :
(1) create a node N;
(2) if samples are all of the same class, C then
(3)   return N as leaf node labeled with the class C;
(4) if attribute_list is empty then
(5)   return N as a leaf node labeled with the most common class in
      samples; // majority voting
(6) select test_attribute, the attribute among attribute_list with the highest
      information gain;
(7) label node N with test_attribute;
(8) for each known value ai of test_attribute // partition the samples
(9)   grow a branch from node N for the condition test_attribute = ai;
(10)  let si is the set of samples in samples for which
      test_attribute = ai; // a partition
(11)  if si is empty then // si=attribute_list.test_attribute
(12)   attach a leaf labeled with the most common class in samples;
(13)  else attach the node returned by Generate_service_tree
    
```

그림 4. 서비스 트리 생성 알고리즘

표 1. 훈련 데이터 테이블(training data)

RID	User	Time	Device	Service_mode	
1	FD	1T	R1	LI	WU
2	FD	1T	R3	CE	WU
3	MO	1T	CI	RE	CO
4	MO	1T	CI	GV	CO
5	MO	1T	CI	GB	CO
6	MO	1T	CI	FA	CO
7	FD	1T	FR	DO	GO
8	SO	3T	FR	DO	GO
9	MO	3T	CI	RE	CO
10	MO	3T	CI	GV	CO
11	MO	3T	CI	GB	CO
12	MO	3T	CI	FA	CO
13	MO	3T	R3	TV	MV
14	MO	3T	R3	DV	MV
15	MO	3T	R3	CE	MV
16	MO	4T	CI	RE	CO
17	MO	4T	CI	GV	CO
18	MO	4T	CI	GB	CO
19	MO	4T	CI	FA	CO
20	FD	6T	R3	CE	SL
21	FD	6T	R1	LI	SL

이러한 서비스 트리 생성 알고리즘에 서비스 트리를 생성하는 과정을 살펴보자. 표 1은 context 데이터베이스에서 추출한 데이터 튜플들로 구성된 훈련 데이터를 나타낸 것이다. 클래스 레이블 속성 service\_mode는 {WU(기상모드), CO(조리모드), GO(외출모드), MO(영화모드), SL(취침모드)} 5 개의 값을 취한다. 따라서 5개의 상이한 클래스가 있다(m=5). 클래스 C1→WU, C2→CO, C3→GO에 대응된다고 하자. WU 클래스에 2개, CO 클래스에 12개, GO 클래스에 2개, MO 클래스에 3개, SL 클래스에 2개의 샘플이 존재한다. 각 속성에 대한 정보 이득을 계산하기 위해, 먼저 식3.4을 사용하여 주어진 샘플을 분류하는데 필요한 기대 정보량을 계산한다.

$$I(s_1, s_2, s_3, s_4, s_5) = -\frac{2}{21} \log_2 \frac{2}{21} - \frac{12}{21} \log_2 \frac{12}{21} - \frac{2}{21} \log_2 \frac{2}{21} - \frac{3}{21} \log_2 \frac{3}{21} - \frac{2}{21} \log_2 \frac{2}{21} = 1.831631 \quad (3.5)$$

다음에 각 속성의 엔트로피를 계산한다. 속성 user의 엔트로피를 계산하면, user의 각 값에 대한 {WU, CO, GO, MO, SL}의 각 분포에 대한 기대 정보량을 계산한다.

For user="FD";

$$s_{11}=2, s_{21}=0, s_{31}=1, s_{41}=0, s_{51}=2$$

$$I(s_{11}, s_{21}, s_{31}, s_{41}, s_{51}) = 1.52193$$

For user="MD";

$$s_{12}=0, s_{22}=12, s_{32}=0, s_{42}=3, s_{52}=0$$

$$I(s_{12}, s_{22}, s_{32}, s_{42}, s_{52}) = 0.721928$$

For user="SO";

$$s_{13}=0, s_{23}=0, s_{33}=1, s_{43}=0, s_{53}=0$$

$$I(s_{13}, s_{23}, s_{33}, s_{43}, s_{53}) = 0$$

식 3.2를 사용하여, 샘플들이 user에 따라 분할되는 경우 주어진 샘플을 분류하는데 욕구되는 기대 정보량은 다음과 같다.

$$E(user) = \frac{5}{21} I(s_{11}, s_{21}, s_{31}, s_{41}, s_{51}) + \frac{15}{21} I(s_{12}, s_{22}, s_{32}, s_{42}, s_{52}) + \frac{1}{21} I(s_{13}, s_{23}, s_{33}, s_{43}, s_{53}) = 1.878027 \quad (3.6)$$

따라서 분할로 인한 정보 이득은 다음과 같다.

$$Gain(user) = I(s_1, s_2, s_3, s_4, s_5) - E(user) = 0.953603 \quad (3.7)$$

같은 방법으로, 시간과 기기에 대한 정보 이득을 계산하면, Gain(time)=0.9234, Gain(device)= 0.8972 을 얻을 수 있다. 이 중 사용자 속성이 가장 높은 정보 이득을 가지므로 사용자 속성이 첫 번째 검사 속성이 된다. 새 노드가 생성되고 user라는 레이블을 갖게 되며 가지들이 속성의 각 값들에 대해서 생성된다. 그리고 샘플들은 그림 4와 같이 분할된다. 남은 속성인 시간과 기기를 위와 같은 방법으로 다음 노드에 들어갈 검사 속성을 선택한다.

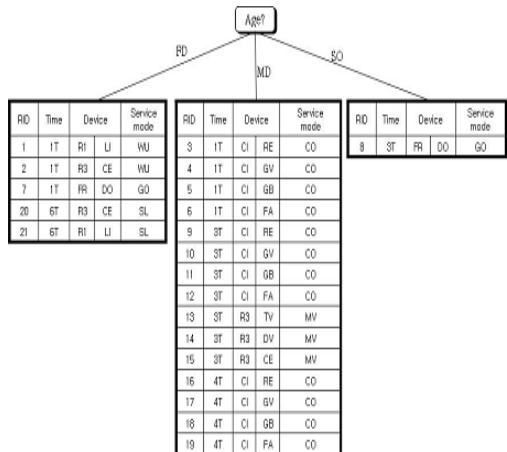


그림 4. Age 속성에 따라 1차 분류된 서비스 트리

### 3.2 정량적 가중치 산출 알고리즘

로그 데이터베이스의 데이터들은 전처리과정을 통해 간결하고 서비스 추론에 필요한 데이터, 즉 context data로 변환되어 context 데이터베이스에 저장된다. context 데이터베이스에는 사용자가 어떤 기기를 언제 사용했다는 기록이 저장되어 있는데, 이러한 일련의 데이터를 구분하는 각각의 속성들을 context로 본다. 앞서 살펴본 서비스 트리가 만들어진 후 어떤 서비스를 제공할지 트리의 노드에 따라 결정해야 한다. 즉, 트리의 잎 노드에 있는 서비스 모드를 결정해야 한다. 이렇게 서비스 모드를 결정하기 위해 트리의 중간 노드들의 가중치를 척도로 사용하여 가장 큰 가중치를 갖는 잎 노드가 선정된다. 가중치를 산출하기 위해 정량적 가중치 산출(CQW : Compute the Quantitative Weight) 알고리즘을 사용한다. CQW 알고리즘은 정량적 특성화 규칙과 정량적 판별 규칙을 종합한 정량적 서술 규칙으로 정의된다.

#### 3.2.1 정량적 특성화 규칙

정량적 특성화 규칙에서 각 논리합(disjunct)이나 일반화 테이블의 각 튜플이 가지는 특이성을 나타내기 위한 흥미도를 나타내는 척도로 t-가중치(t-weight)를 사용하는데, 이 척도는 다음과 같이 정의된다. 특성화하고자 하는 대상의 클래스를 목표 클래스(target class)라 하고,  $q_a$  를 이 목표 클래스의 일반화된 튜플이라고 하자. 그러면  $q_a$  의 t-가중치는 초기 작업 테이블에서  $q_a$  에 대응하는 목표클래스의 튜플의 비율을 의미한다. 이 정의는 식 3.8과 같이 표현된다.

$$t - weight = count(q_a) / \sum_{i=1}^n count(q_i) \quad (3.8)$$

이때 n은 일반화 테이블에서 목표 클래스의 튜플의 수이고  $q_1, \dots, q_n$  은 일반화 릴레이션의 목표 클래스의 튜플들을 나타내고  $q_a$  는  $q_1, \dots, q_n$  의 하나이다. 물론 t-가중치의 범위는 [0.00, 1.00] 또는 [0%, 100%]이다. 정량 특성화 규칙(quantitative characteristic rule)은 ①목표 클래스를 만족하는 논리합에 t-가중치가 추가된 논리식의 형태로 표현하거나, 또는 ②목표 클래스에 속하는 각 튜플의 count 값을 그에 해당하는 t-가중치 값으로 대체하

여 만들어지는 관계 테이블이나 교차표의 형태로 표현될 수 있다.

정량 특성화 규칙에서 각 논리합들은 조건을 나타낸다. 일반적으로 이러한 조건들의 논리합은 목표 클래스의 필요조건을 구성하게 되는 데 이 조건들은 목표 클래스의 모든 경우를 근거로 유도되기 때문이다. 즉 목표 클래스의 모든 튜플은 이러한 조건을 만족시켜야만 한다. 그러나 같은 조건을 만족시키는 튜플이 다른 클래스에 속할 수도 있기 때문에 이 규칙이 목표 클래스의 충분조건이 되지 않는 것이다. 그러므로 이 규칙은 식 3.9과 같은 형식을 가진다.

$$\forall X, target\_class(X) \Rightarrow condition_1(X)[t:w_1] \vee \dots \vee condition_m[t:w_m] \quad (3.9)$$

이 규칙은 만약 X가 target\_class에 있다면, X가  $condition_i$  를 만족시킬 확률은  $w_i$  이며,

$w_i$  는 {1,2,...,m}에 속하는 i번째 조건 또는 논리합의 t-가중치를 나타낸다.

예를 들어, 표 1에서 사용자와 서비스 모드 속성의 count를 표시한 표 2를 논리규칙 형식으로 변환할 수 있다. service\_mode를 목표 클래스 속성으로 하면, 이에 대응되는 논리식 형식의 특성화 규칙은 식 3.10과 같다.

$$\begin{aligned} \forall X, user(X) = FD \Rightarrow \\ (service\_mode(x) = WU[t:100.00\%]) \vee (service\_mode(x) = CO)[t:0.00\%] \\ \vee (service\_mode(x) = GO[t:50.00\%]) \vee (service\_mode(x) = MO)[t:0.00\%] \\ \vee (service\_mode(x) = SL[t:100.00\%]) \end{aligned} \quad (3.10)$$

처음의 t-가중치 값인 100.00%는 “(WU, FD)”의 count 항목 값에 2을 “(all\_mode, FD)”에 해당하는 count 항목 값인 2로 나누어 계산한 것이다. 다른 목표 클래스들에 대한 정량 특성화 규칙 또한 같은 방법으로 산출된다.

표 2. user별 service\_mode count 테이블

user	service_mode					
	WU	CO	GO	MV	SL	all_mode
FD	2	0	1	0	2	5
MO	0	12	0	3	0	15
SO	0	0	1	0	0	1
all_user	2	12	2	3	2	21

3.2.2 정량적 판별 규칙

정량적 판별 규칙은 비교 서술의 목표와 대조 클래스의 판별 특징을 나타내는 것으로 각 일반화된 튜플에 대해서 통계적 흥미 척도인 d-가중치(d-weight)를 사용하여 정의된다.

$q_a$ 를 일반화된 튜플이고  $C_j$ 를 목표 클래스이며,  $q_a$ 가 목표 클래스의 일부 튜플들을 만족한다면 클래스 비교를 하고자 하는 것이므로  $q_a$ 는 대조 클래스의 일부 튜플들을 포함할 수도 있다는 것을 유의해야 한다.  $q_a$ 의 d-가중치는  $q_a$ 를 만족하는 초기 목표 클래스의 튜플 수  $q_a$ 를 만족하는 목표 클래스와 대조 클래스 작업 릴레이션에 있는 모든 튜플의 수로 나눈 것이다. 형식적으로  $C_j$ 의  $q_a$ 에 대한 d-가중치는 다음과 같이 정의된다.

$$t-weight = count(q_a \in C_j) / \sum_{i=1}^m count(q_a \in C_i) \tag{3.11}$$

여기서 m은 목표 클래스와 대조 클래스의 총 튜플 수이고,  $C_j$ 는 중의  $\{C_1, \dots, C_m\}$  하나이며,

$count(q_a \in C_i)$  는  $q_a$ 를 만족하는 클래스  $C_i$  튜플의 수이다. d-가중치 값의 범위는 [0.00, 1.00] 또는 [0%, 100%]이 된다. 목표 클래스에서 d-가중치가 높다는 것은 이 일반화된 튜플의 개념이 목표 클래스에 주로 기인한다는 것이고 낮은 d-가중치는 이 튜플의 개념이 대조 클래스에서 더 많이 기인한다는 것을 의미한다. 주어진 비교서술의 목표 클래스에 대한 정량적 판별 규칙은 식 3.12와 같은 형식을 가진다.

$$\forall X, target\_class(X) \Rightarrow condition(X)[d:d\_weight] \tag{3.12}$$

여기서 조건은 서술에 대한 일반화된 튜플로 구성되므로 화살표 방향이 왼쪽에서 오른쪽으로 가는 클래스 특성화로 얻어진 식 3.7과는 다르다. 표 1를 오전시간(1,3T)과 오후시간(4,6T)으로 구분하여 service\_mode의 count 분포를 나타내면 표 3과 같다.

표 3. 시간대별 사용자에 따른 서비스 모드 count 테이블

time	user	service_mode					
		WU	CO	GO	MV	SL	all_mode
1T, 3T	FD	2	0	1	0	0	3
	MD	0	4	0	0	0	4
	SO	0	0	0	0	0	0
	all_user	2	4	1	0	0	7
4T, 6T	FD	0	0	0	0	2	2
	MD	0	8	0	3	0	11
	SO	0	0	1	0	0	1
	all_user	0	8	1	3	2	14

오후 데이터를 목표 클래스(target\_class)라고 하고 오전 데이터를 대조 클래스(compare\_class)라고 하자. 여기서 사용자가 어머니(MD)이면서 서비스 모드가 조리모드(CO)일 경우의 오후 데이터에 대한 정량적 판별 규칙은 식 3.13과 같은 형식으로 표현된다.

$$\forall X, time(X)=4Tand6T \Leftarrow user=MD \wedge service\_mode=CO [d:66.66\%] \tag{3.13}$$

위 식에서 d-가중치 값은 사용자가 어머니이며 서비스 모드가 조리 모드일 때 오후 데이터의 count 값인 8을 모든 시간에 대한 count 값인 12로 나눈 값이다. 즉, 사용자가 어머니이고 서비스 모드가 조리이면 그 시간대가 오후 시간일 확률을 의미하는 것이다.

3.2.3 정량적 가중치 산출 알고리즘

동일한 클래스에 대한 정량적 특성화 규칙과 정량적 판별 규칙은 대응되는 특성화 규칙과 판별 규칙, 즉 t-가중치와 d-가중치를 종합하여 정량적 가중치 산출 (Compute the Quantitative Weight) 알고리즘으로 표현할 수 있다.

$$\forall X, target\_class(X) \Rightarrow condition_i(X)[t:w_i] \vee \dots \vee condition_m[t:w_m] \tag{3.14}$$

식 3.14는 만약 target\_class에 있다면, i는 {1, ..., m}중 하나일 때, X가 condition\_i 를 만족할 확률은 t-가중치 즉  $w_i$  라는 것을 나타낸다. 정량적 판별 규칙은 목표 클래스와 대조 클래스에 나타나는 특성에 대한 정량적 척도를 제시하기 때문

에 목표 클래스에 대한 충분조건을 제공한다. 이 규칙은 식 3.15와 같은 형식을 가진다.

$$\forall X, target\_class(X) \leftarrow condition_1(X)[d:w_1] \vee \dots \vee condition_m[d:w_m] \quad (3.15)$$

이 규칙은  $i$ 가  $\{1, 2, \dots, m\}$ 의 하나인 경우에 대해서  $X$ 가  $condition_i$ 를 만족하면  $target\_class$ 에 속할 확률이 ( $d$ -가중치의 값)라는 것이다.

량적 특성화 규칙과 정량적 판별 규칙은 다음과 같이 정량적 서술 규칙의 형태로 통합될 수 있다.

(1) 각 조건에 대해서 그에 연관된  $t$ -가중치와  $d$ -가중치를 표시하고,

(2) 주어진 클래스와 조건들에 대해서 양방향 화살표를 사용한다. 즉, 정량적 가중치 산출 알고리즘은 식 3.16과 같은 형식을 가진다.

$$\forall X, target\_class(X) \leftrightarrow condition_1(X)[t:w_1, d:w'_1] \vee \dots \vee condition_m(X)[t:w_m, d:w'_m] \quad (3.16)$$

이 식은 1부터  $m$ 까지 중의  $i$ 에서 만약  $X$ 가  $target\_class$ 에 있다면  $X$ 가 조건  $i$ 를 만족할 확률은

$w_i$  이고, 만약  $X$ 가 조건  $condition_i$ 를 만족하면  $X$ 가  $target\_class$ 에 속할 확률이  $w'_i$ 라는 것이다.

표 4는 표 1에서 사용자별 서비스 모드에 따른 count 값과 각각의 속성에 대한 정량적 특성 및 정량적 비교 규칙을 적용하여 산출한 교차표이다. 예를 들어 목표 클래스 CO(조리 모드)에 대한 정량적 가중치 산출 알고리즘은 식 3.17와 같다.

표 4. 사용자별 서비스 모드에 대한 t-가중치 및 d-가중치 테이블

service_mode	user											
	FD			MD			SO			allUser		
	count	t-weight	d-weight	count	t-weight	d-weight	count	t-weight	d-weight	count	t-weight	d-weight
WU	2	1.00	0.40	0	0.00	0.00	0	0.00	0.00	2	1.00	0.10
CO	0	0.00	0.00	12	1.00	0.80	0	0.00	0.00	12	1.00	0.57
GO	1	0.50	0.20	0	0.00	0.00	1	0.50	1.00	2	1.00	0.10
MV	0	0.00	0.00	3	1.00	0.20	0	0.00	0.00	3	1.00	0.14
SL	2	1.00	0.40	0	0.00	0.00	0	0.00	0.00	2	1.00	0.10
allmode	5	0.24	1.00	15	0.71	1.00	1	0.05	1.00	21	1.00	1.00

표 5. 사용자별 서비스 모드에 따른 정량적 가중치 (q-weight) 테이블

service_mode	user							
	FD		MD		SO		allUser	
	count	q-weight	count	q-weight	count	q-weight	count	q-weight
WU	2	0.40	0	0.00	0	0.00	2	0.10
CO	0	0.00	12	0.80	0	0.00	12	0.57
GO	1	0.10	0	0.00	1	0.50	2	0.10
MV	0	0.00	3	0.20	0	0.00	3	0.14
SL	2	0.40	0	0.00	0	0.00	2	0.10
allmode	5	0.24	15	0.71	1	0.05	21	1.00

$$\forall X, service\_mode(X) = 'GO' \leftrightarrow$$

$$user = FD[t:50\%d:20\%] \vee user = MD[t:0\%d:0\%] \vee user = SO[t:50\%d:100\%] \quad (3.17)$$

이것은 서비스 모드가 외출 모드(GO)일 경우는 그 사용자가 아버지일 확률은 0.5이고, 어머니일 확률은 0 그리고 아들일 확률은 0.5라는 것을 의미한다. 반면에 사용자가 아버지일 때 기상 모드(WU)일 확률은 0.4, 조리 모드(CO)일 확률은 0, 외출 모드일 확률은 0.2, 영화 모드일 확률은 0, 수면 모드일 확률은 0.4라는 것을 의미한다. t-weight 값과 d-weight 값의 곱으로 특성 조건이 일어날 확률을 산출할 수 있는데, 이것을 q-weight(quantitative weight)라고 한다. 예를 들어, 사용자가 아버지이고 서비스 모드가 외출 모드인 상황은 아래 식과 같이 t-weight 값 0.5와 d-weight 값 0.2를 곱하여 q-weight 0.1을 산출할 수 있다.

$$\forall X, service\_mode(X) = 'GO' \wedge user = FD[q:0.1] \quad (3.18)$$

정량적 가중치 산출 알고리즘으로 사용자와 서비스 모드간의 정량적 가중치 값을 계산하면 표 5와 같다.

#### IV. 시뮬레이션 결과 및 고찰

제안한 지능형 서비스 추론 모델이 실질적으로 예측정확도와 지연시간 면에서 어느 정도의 성능을 갖는지 평가를 수행한다. 가정 내 사용자의 생활 데이터를 이용하여 실험하였고, 성능 평가는 사용자별, 기기별, 시간별 정확도를 구분하여 측정한다. 마지막으로 서비스 모드별 정확도를 측정하여 스마트 홈 환경에서 모델 적용의 적합성을 검증한다.



### 4.1 시물레이션 환경

시물레이션에 사용된 환경은 30평형대의 일반 아파트에서 생활하는 가족 구성원의 생활을 데이터 추출하여 사용되었다. 그림 5는 본 시물레이션에 사용된 아파트를 단면으로 표현한 것이다. 기기와 기기가 설치된 장소는 3장에서 설명된 기기 테이블(device table)에 따라 구성되어 있으며, 기기는 고정 설치된 것으로 가정한다. 가족 구성원 및 시간도 3장에 명시된 사용자 테이블(user table) 명세에 따라 정의한다.

본 논문에서는 아버지, 어머니, 아들로 구성된 한 가정의 생활을 2007년 1월 3일부터 한 달간 기록하여 이를 본 모델의 데이터 스키마에 맞게 변환하여 데이터를 추출하였다. 실제 스마트 홈 환경아래 각 사용자의 움직임을 수집하는 센서와 각 기기별 제어 정보를 수집할 수 있는 홈 서버(게이트웨이)를 통해 데이터를 수집하여야 하나, 이러한 환경에서 하기에는 여러 제약 조건이 있어 실험자가 개별적으로 기록하여 데이터를 추출하였다. 추출된 데이터는 총 731개의 데이터로 이중 600개는 훈련 데이터(training data)로, 나머지 131개는 실험 데이터(test data)로 구분하여 시물레이션에 사용하였다. 이중 훈련 데이터 600개는 그림 6과 같이 사용자, 시간, 기기에 따른 데이터 분포를 보였다.

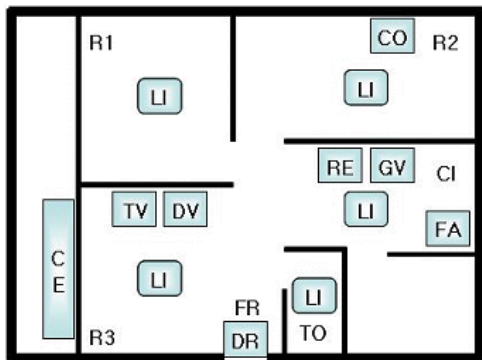


그림 5. 스마트 홈 환경 구성

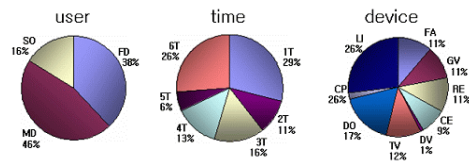


그림 6. 속성별 훈련데이터 구성도

### 4.2 시물레이션 결과

샘플 데이터 중 600개의 훈련 데이터를 이용하여 서비스 트리를 생성하기 위해 서비스 생성 알고리즘에 따라 각 속성별 정보 이득(Information Gain) 값을 계산하면 아래와 같은 결과를 얻을 수 있다. 여기서 사용자 테이블에서는 역할(u\_name) 속성에 따라, 기기 테이블에서는 각 기기명(d\_name)에 따라, 시간 테이블에서는 시간(time) 속성에 따라 서비스 트리를 생성하였다.

위 속성별 정보 이득 값 중 사용자 속성이 가장 큰 정보 이득 값을 가졌기 때문에, 루트 노드에 적용되고, 사용자의 속성 값, 즉 아버지(FD), 어머니(MD), 아들(SO)에 따라 1차 노드들에 대한 속성을 부여한다. 첫 번째 아버지와 연결된 1차 노드는 시간이, 어머니와 연결된다. 같은 방법으로 2차 노드까지 계산하면 그림 7과 같은 서비스 트리를 생성할 수 있다. 서비스 트리의 각 노드들과 서비스 모드와의 관계에 대한 정량적 가중치 값(q-weight score)을 정량적 가중치 산출 규칙에 따라 계산할 수 있다. 표 6는 시간에 따른 기기 및 사용자의 q-weight 값을, 표 7은 기기에 따른 사용자와 서비스 모드의 q-weight 값을, 표 8은 사용자에 따른 기기와 서비스 모드의 q-weight 값을 각각 계산한 테이블이다.

훈련 데이터를 통해 산출한 서비스 트리와 가중치 값을 이용하여 산출한 예측 값과 실험 데이터의 결과 값을 비교하여 예측 정확도를 측정하였다. 훈련 데이터 중 사용자별, 시간별, 기기별 정확도를 훈련데이터수를 100개에서 600개까지 100개씩 증가하여 측정하였다.

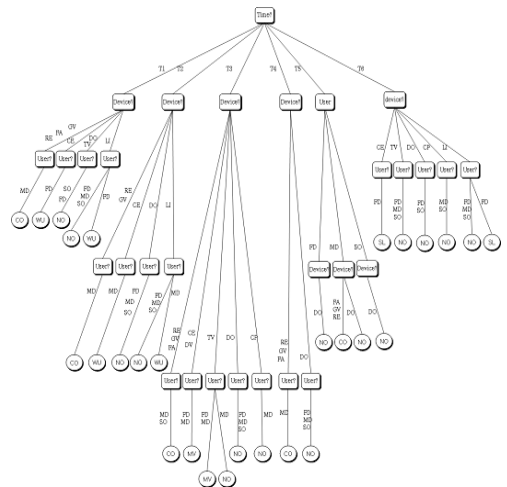


그림 7. 훈련 데이터의 서비스 트리

표 6. 시간에 따른 기기 및 사용자 q-가중치 테이블

	TIME					
	1T	2T	3T	4T	5T	6T
Device						
GV	0.03503	0.00213	0.06588	0.08214	0.09585	0.00000
FA	0.03503	0.00213	0.06588	0.08214	0.09585	0.00000
RE	0.03503	0.00213	0.06588	0.08214	0.09585	0.00000
CE	0.04902	0.00270	0.00523	0.00000	0.00000	0.06521
DV	0.00000	0.00000	0.05435	0.00000	0.00000	0.00000
TV	0.03351	0.00000	0.03088	0.00000	0.00000	0.11381
DO	0.02484	0.13580	0.00323	0.09418	0.00000	0.00488
CP	0.00000	0.00000	0.00815	0.00000	0.00000	0.04327
LI	0.09021	0.05425	0.00000	0.00000	0.22885	0.23365
all_device	0.28833	0.10987	0.15333	0.13187	0.08000	0.28000
User						
FD	0.26159	0.00235	0.02201	0.00048	0.03786	0.19837
MD	0.09825	0.02590	0.17800	0.20106	0.02873	0.02952
SO	0.00219	0.22516	0.00183	0.01612	0.00117	0.08268
all_user	0.28833	0.10987	0.15333	0.13187	0.08000	0.28000

표 7. 사용자에 따른 기기 및 서비스 모드 q-가중치 테이블

	User			
	FD	MD	SO	all_user
Device				
GV	0.00000	0.11054	0.00008	0.05299
FA	0.00000	0.11054	0.00008	0.05299
RE	0.00000	0.11054	0.00008	0.05299
CE	0.07372	0.00102	0.00000	0.03549
DV	0.00177	0.00323	0.00000	0.00417
TV	0.04299	0.01295	0.00095	0.04587
DO	0.04416	0.00433	0.07944	0.08179
CP	0.00000	0.00272	0.01563	0.00727
LI	0.08601	0.00708	0.04359	0.10288
all_device	0.16236	0.20043	0.06825	0.43103
Service				
CO	0.00000	0.69120	0.00084	0.33000
MV	0.02358	0.00873	0.00000	0.02500
NO	0.25968	0.05909	0.30091	0.48833
SL	0.20087	0.00000	0.00000	0.07667
WU	0.16048	0.00273	0.00000	0.08000
all_Service	0.38167	0.45833	0.16000	1.00000

표 8. 기기에 따른 사용자 및 서비스모드 q-가중치 테이블

	Device								
	GV	FA	RE	CE	DV	TV	DO	CP	LI
User									
FD	0.00000	0.00000	0.00000	0.18005	0.00854	0.01790	0.09179	0.00000	0.22818
MD	0.22344	0.22344	0.22344	0.00248	0.00645	0.03247	0.00900	0.00747	0.01961
SO	0.00016	0.00016	0.00016	0.00000	0.00000	0.00137	0.16512	0.04286	0.11462
all_user	0.11100	0.11100	0.11100	0.08867	0.00833	0.11500	0.17000	0.02000	0.27000
Device									
CO	0.33333	0.33333	0.33333	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
MV	0.00000	0.00000	0.00000	0.03205	0.33333	0.02415	0.00000	0.00000	0.00000
NO	0.00000	0.00000	0.00000	0.00000	0.00000	0.23280	0.34812	0.04096	0.27862
SL	0.00000	0.00000	0.00000	0.22115	0.00000	0.00000	0.00000	0.00000	0.07899
WU	0.00000	0.00000	0.00000	0.23077	0.00000	0.00000	0.00000	0.00000	0.07407
all_device	0.11100	0.11100	0.11100	0.08867	0.00833	0.11500	0.17000	0.02000	0.27000

4.2.1 사용자별 예측 정확도

그림 8은 사용자별 예측 정확도를 나타낸 것으로 훈련 데이터가 증가할수록 예측정확도는 증가하다, 400개 이상에서는 큰 차이를 없다. 또한 아버지나 어머니 보다 집안에서 활동이 적은 아들의 예측 정확도가 상대적으로 높게 나온 것은 그 비중(14%)이 작기 때문인 것으로 판단된다.

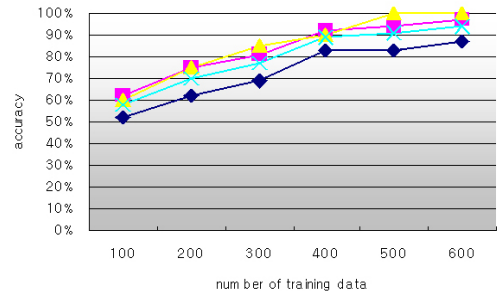


그림 8. 사용자별 예측 정확도

4.2.2 시간대별 예측 정확도

그림 9는 시간대별 예측 정확도를 나타낸 것으로 사용자 예측 정확도와 비슷하게 훈련데이터가 400개 까지 증가할수록 정확도가 올라가지만, 그 이상에선 큰 변화가 없었다. 1T, 2T, 3T, 6T 시간대에 비해 상대적으로 활동량이 적은 시간, 즉 데이터 수가 적은 4T와 5T 시간대에서는 전체 훈련 데이터의 크기에 따라 예측 정확도의 편차가 크게 나타났다.

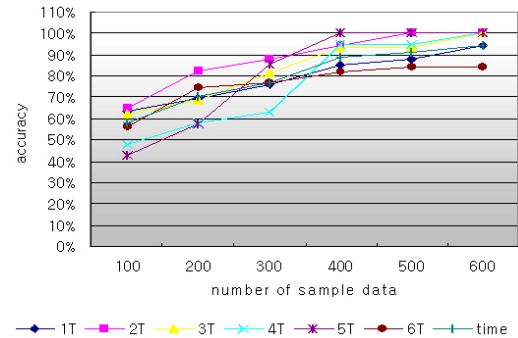


그림 9. 시간대별 예측 정확도

4.2.3 기기별 예측 정확도

그림 10은 기기별 예측 정확도를 나타낸 것으로 사용자 및 시간대별 예측 정확도와 유사하게 훈련

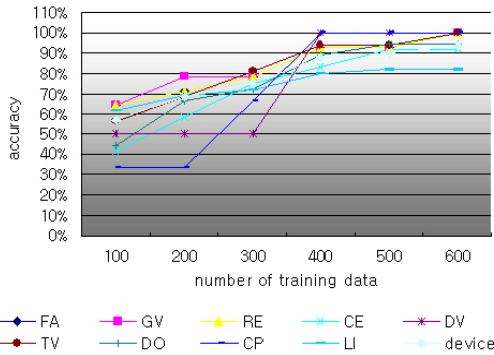


그림 10. 기기별 예측 정확도

데이터의 수가 400개 이전까지는 큰 폭으로 증가하다가, 400개 이상이 되면 예측 정확도의 큰 변화가 없었다. 또한, 전체 데이터 중 상대적으로 사용이 적은 즉 데이터 수가 적은 DVD(DV)와 컴퓨터(CP)는 훈련 데이터의 크기에 따라 예측 정확도의 편차가 컸다.

#### 4.2.4 서비스 모드별 정확도

그림 11은 서비스 모드별 예측 정확도를 나타낸 것으로 앞서 살펴본 속성들과 같이 대부분의 서비스 모드에서 훈련데이터의 수가 400개 이상이 되면 90% 이상의 예측 정확도를 보인다. 단, 취침모드(SL)의 경우 다른 서비스 모드에 비해 상대적으로 낮은 예측 정확도를 보이는데, 이것은 취침모드에 사용되는 전등(LI)이 취침모드보다 일반 모드(NO)에서 더 많이 사용되었기 때문으로 나온 결과다. 따라서, 서비스 테이블에서 취침모드의 조건을 수정이 필요하였다. 또한, 영화모드(MV)의 경우 예측 정확도의 편차가 크게 나타났는데, 이것은 실험 데이터에서 영화모드에 해당되는 데이터가 극히 작았기 때문이다.

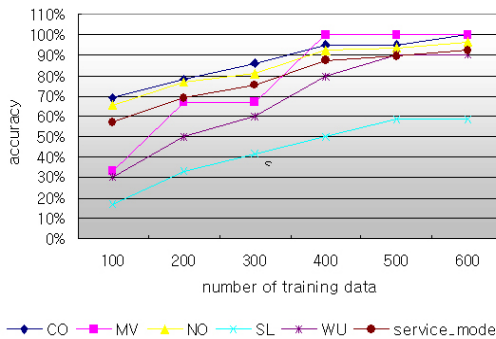


그림 11. 서비스 모드별 예측 정확도

지금까지 시뮬레이션을 통해 지능형서비스추론(ISR) 모델의 예측 정확도를 사용자, 시간대, 기기 그리고 서비스 모드별로 살펴보았다. 각각의 결과를 통해 본 모델은 학습데이터의 수가 400개 이상일 경우 90%내외의 예측 정확도를 가진다는 것을 확인할 수 있었다. 사용자와 시간대별 예측 정확도는 각 요소별로 비슷한 예측 정확도를 보였으나, 기기 및 서비스 모드별 예측 정확도에서 일부 요소들이 큰 차이점을 보였다. 기기의 DVD와 컴퓨터 그리고 서비스 모드의 영화모드(MV)가 그 경우이다. 이것은 시뮬레이션에 사용된 데이터의 수가 다른 요소에 비해 상대적으로 적기 때문에 예측 정확도에 민감하게 나타난 것으로 판단된다. 따라서, ISR 모델을 적용한 스마트 홈 환경에서는 초기 학습데이터의 부재로 모드별 서비스에는 제약이 많겠지만, 1달 이상의 학습데이터가 축적되면 90% 이상의 예측 정확도를 제공할 수 있을 것으로 판단된다.

## V. 결론

본 논문에서는 스마트 홈 환경에서 데이터 마이닝 기법을 이용하여 사용자가 원하는 서비스를 추론하여 제공하는 지능형 서비스 추론(ISR) 모델을 제시하였다. ISR 모델에서는 데이터 마이닝 기법 중 의사결정트리 방법인 C4.5 알고리즘을 기반으로 한 서비스 트리 생성 알고리즘을 통해 서비스 트리를 생성하고, 정량적 특성화 규칙과 정략적 판별 규칙을 기반으로 한 정략적 가중치 산출 알고리즘을 통해 서비스 추론에 사용되는 각 속성들 간의 정략적 가중치(q-weight)를 산출한다. 이러한 과정을 거쳐 사용자가 원하는 서비스를 추론하여 제공할 수 있다. 또한, 약 한 달간의 생활 데이터를 수집하여 ISR 모델에 적용하여 본 논문에 제안하는 ISR 모델의 예측 정확도를 확인하였다. 추론에 사용된 사용자, 시간대, 기기별 요소에 따라 예측 정확도를 산출한 결과, 훈련데이터의 수가 증가함에 따라 예측 정확도 또한 증가하면, 400개 이상의 훈련데이터의 경우 90% 이상의 예측 정확도를 보였다. 하지만, 기기와 서비스 모드별 일부 요소에서는 다른 요소에 비해 상대적으로 낮은 예측 정확도를 보이거나 훈련 데이터에 따라 큰 편차를 보였다. 이러한 결과는 훈련데이터의 수를 크게 하여 해결할 수 있을 것으로 판단되며, 또한 서비스 모드를 정의하는 서비스 테이블의 수정으로 보완될 수 있을 것으로 판단된다.

참 고 문 헌

- [1] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami, "Data Mining : A Performance Perspective", IEEE Transaction on Knowledge and Data Engineering, Vol.5, no.6, pp.914-925, December 1993.
- [2] J.R.Quinlan, "C4.5:Programs for Machine Learning", San Mateo, CA : Morgan kaufmann, 1993.
- [3] Kim, H. and Loh, W.-Y, "Classification trees with bivariate linear discriminant node models, Journal of computational and Graphical Statistics, pp.5112-5130, December 2003.

강 명 석 (Myung-seok Kang)

준회원



2001년 2월 원광대학교 컴퓨터 공학과 졸업  
2003년 2월 원광대학교 컴퓨터 공학과(석사)  
2003년 9월~현재 연세대학교 전기전자공학과(박사과정)  
<관심분야> 실시간 고장포용 시스템, 지능형 홈네트워크

김 학 배 (Hag-bae Kim)

정회원



1988년 2월 서울대학교 전자공학과 졸업  
1990년 2월 미국 미시간대학교 전기 및 컴퓨터공학과(석사)  
1994년 2월 미국 미시간대학교 전기 및 컴퓨터공학과(박사)  
1996~현재 연세대학교 전기전자공학과 교수

<관심분야> 실시간 시스템, 인터넷 웹서버 기술, 디지털시스템 고장포용 및 신뢰도 평가분야