

데이터 마이닝 기법을 이용한 상황 추론 추천시스템

A Recommendation System based on Context Reasoning by Data Mining Techniques

이재식[Ⓐ], 이진천[Ⓑ]

[Ⓐ] 아주대학교 경영대학 e-비즈니스학부
경기도 수원시 영통구 원천동 산 5번지
Tel: 031-219-2719, Fax: 031-219-1616, E-mail: leejsk@ajou.ac.kr

[Ⓑ] 유비쿼터스 컨버전스 연구소
경기도 수원시 영통구 원천동 산 5번지
Tel: 031-219-2308, Fax: 031-219-1616, E-mail: giny777@empal.com

Abstract

본 연구에서는 상황 추론의 기능을 추천 시스템에 접목하였다. 연구의 대상 영역은 음악 추천 분야인데, 본 연구에서 제안하는 시스템은 세 개의 모듈, 즉 Intention Module, Mood Module 그리고 Recommendation Module로 구성되어 있다. Intention Module은 사용자가 음악을 청취할 의향이 있는지 없는지를 외부 환경의 상황 데이터를 이용하여 추론한다. Mood Module은 사용자의 상황에 적합한 음악의 장르를 추론한다. 마지막으로 Recommendation Module은 사용자에게 선정된 장르의 음악을 추천한다.

Keywords:

Context-Awareness; Context Reasoning; Data Mining; Case-Based Reasoning; Decision Tree; Support Vector Machine

1. 서론

유비쿼터스 컴퓨팅이란 물리적 환경 전반에 걸쳐 존재하는 컴퓨터들을 사용자에게는 보이지 않게 효과적으로 감추면서, 항상 사용 가능하도록 함으로써 컴퓨터의 활용을 증진시키는 방법을 말한다 [11]. 상황인식 (Context-Awareness)은 유비쿼터스 컴퓨팅의 중요한 특성 중의 하나이다.

향후의 응용 프로그램들이 Context-Awareness가 가능한 지능적 유비쿼터스 응용 프로그램이 되기 위해서 중요한 것들 중 하나는 사용자의 상황을 파악하는 Context Reasoning 기능이 있어야 한다는 것이다. 본 연구에서는 Context Reasoning의 개념을 추천시스템(Recommendation System)에 접목시키고자 한다. 추천이란 '대상 고객과 유사한 집단의 의견을 활용하여, 그 고객이 대량의 선택가능 집합으로부터 관심 있는 내용을 효과적으로 파악할 수 있도록

도와주는 과정'이라고 정의된다 [7].

본 연구는 다음과 같이 구성되어 있다. 제 2절에서는 Context-Awareness에 관련된 기존 연구들을 살펴보고, 본 연구에서 사용할 데이터 마이닝 기법인 의사결정나무(Decision Tree), Support Vector Machine 그리고 사례기반추론(Case-Based Reasoning)에 대해 간략히 기술한다. 제 3절에서는 본 연구에서 제안하는 음악추천시스템인 M³의 전체적인 구조를 보여준다. 제 4절에서는 연구에 사용한 데이터를 소개하고, M³의 구현 과정을 설명한다. 제 5절에서 M³의 성능을 평가한 후에 제 6절에서 향후 연구 방향과 함께 결론을 내린다.

2. 관련 연구

2.1 상황인식(Context-Awareness)

Context-Awareness(또는 Context-Aware)란 응용 프로그램이 작동되고 있는 환경에 대한 정보를 활용하여 그와 관련된 정보와 서비스를 사용자에게 제공하고자 하는 것이다 [3]. Context-Awareness이라는 용어는 Schilit와 Theimer[8]에 의해서 처음 소개되었다. 그들은 상황이 발생한 장소, 주위의 사람들과 객체들 그리고 그 객체들에 일어난 변화들에 대한 여러 종류의 예를 제시함으로써 Context를 정의하였다.

Context는 여러 학자들에 의해 다양하게 정의되었다. Schmidt 등[9]은 Context를 물리적 환경, 인간적 요소 그리고 시간 등의 세 개의 차원을 사용하여 정의하였다. Benerecetti 등[2]은 상황을 물리적 상황과 문화적 상황으로 분류하였다. 물리적 상황이란 환경을 구성하는 속성들을 말하는 것이고, 문화적 상황이란 사용자에게 대한 정보, 사회적 환경과 믿음과 같은 추상적인 개념을 말한다. Dey와

Abowd[4]는 Context를 ‘어떤 객체가 처한 상황(Situation)의 특징을 표현하는 정보들’이라고 정의하였다. 여기서 객체란 사람, 장소, 물리적 요소 또는 사용자와 응용 프로그램간의 상호작용과 관련되어 있다고 생각되는 컴퓨팅 요소를 말한다. 물론 사용자와 응용 프로그램 자체도 포함된다. Dey[5]는 Context를 ‘장소’, ‘주체’, ‘시간’ 그리고 ‘행동’ 등의 네 가지로 판정하는 분류 시스템을 제시하였다.

2.2. Decision Tree

Decision Tree는 대상이 되는 집단을 몇 개의 소집단으로 구분하여 분류 및 예측을 수행하는 기법이다. 이것은 나무의 깊이, 최종노드 안에 포함되는 사례의 개수, 가지의 분리방법, 그리고 가지치기 등과 같은 기준에 의해 생성된다 [6]. Decision Tree는 가지를 분리하는 방법, 즉 엔트로피(Entropy) 또는 이득률(Gain Ratio) 등에 따라 여러 종류의 알고리즘이 있다.

2.3. Support Vector Machine

Support Vector Machine(SVM)은 통계적 학습 이론(Statistical Learning Theory)에 기반을 둔 기법이다. SVM은 두 집단으로 구분된 입력값을 가지는 학습 데이터를 분류할 때, 기준이 되는 분리경계면 (Separating Hyperplane)을 특수한 학습 알고리즘을 이용하여 찾는다 [10].

두 개의 집단 $y_i \in \{-1, +1\}$ 로 분리된 입력값 $x_i \in R^n$, $i = 1, \dots, N$ 이 있다고 하자. Vapnik[10]이 제안한 공식에 따르면 SVM은 식 (2-1)을 만족한다.

$$\begin{cases} w \cdot x_i + b \geq +1, & \text{if } y_i = +1 \\ w \cdot x_i + b \leq -1, & \text{if } y_i = -1 \end{cases} \quad \text{식 (2-1)}$$

식 (2-1)은 분리경계면 $w \cdot x + b = 0$ 을 사이에 두고 양쪽에 두 개의 평행한 경계면 (Bounding Hyperplane)을 만들게 되는데, 그 폭은 $\frac{2}{\|w\|^2}$ 가 된다.

여기서 w 는 가중치 벡터를 나타내고 b 는 편차를 나타낸다. 학습자료 중에서 이 경계면 상의 점들을 Support Vector라고 한다.

이 Support Vector들로 이루어진 경계면들이 입력값들을 최대마진(Maximal Margin)으로 분류한다. 이 문제는 최적화 문제로 변형되어 풀게 되는데, 최종적으로 식 (2-2)를 얻게 된다.

$$\text{sgn}\left(\sum_{i=1}^n \alpha_i y_i K(x_i, x_j) + b\right) \quad \text{식 (2-2)}$$

식 (2-2)에서 $K(\cdot)$ 는 커널(Kernel) 함수이다. 커널 함수는 이론적으로는 가능하나 실제로는 풀기 힘든 매핑 함수를 대신하여 데이터를 고차원 공간으로 이동시켜 특징 공간 내에 선형으로 분리 가능한 입력자료 집합을 만들어 주는 역할을 수행한다. 커널함수의 예로는 다항식 커널(Polynomial Kernel)과 RBF(Radial Basis Function) 커널 등을 들 수 있다.

2.4 사례기반 추론

사례기반추론(CBR: Case-Based Reasoning)은 과거의 경험을 바탕으로 과거 문제와 유사한 현재 문제의 해결책을 추론하는 기법이다. CBR 시스템의 문제해결 과정은 검색(Retrieve), 재사용(Reuse), 수정(Revise), 유지(Retain)의 4 단계를 거친다 [1]. 검색은 주어진 문제와 가장 유사한 과거사례를 사례베이스로부터 찾는 단계이고, 재사용은 주어진 문제의 해결을 위해 찾아진 유사 사례들의 해법을 재사용하는 단계이다. 수정은 필요에 따라 유사 사례의 해법을 주어진 문제에 적합한 형태로 조정하는 단계이며, 마지막 단계인 유지는 새롭게 해결된 문제와 해법을 향후 문제해결을 위해 사례베이스에 저장하는 것이다.

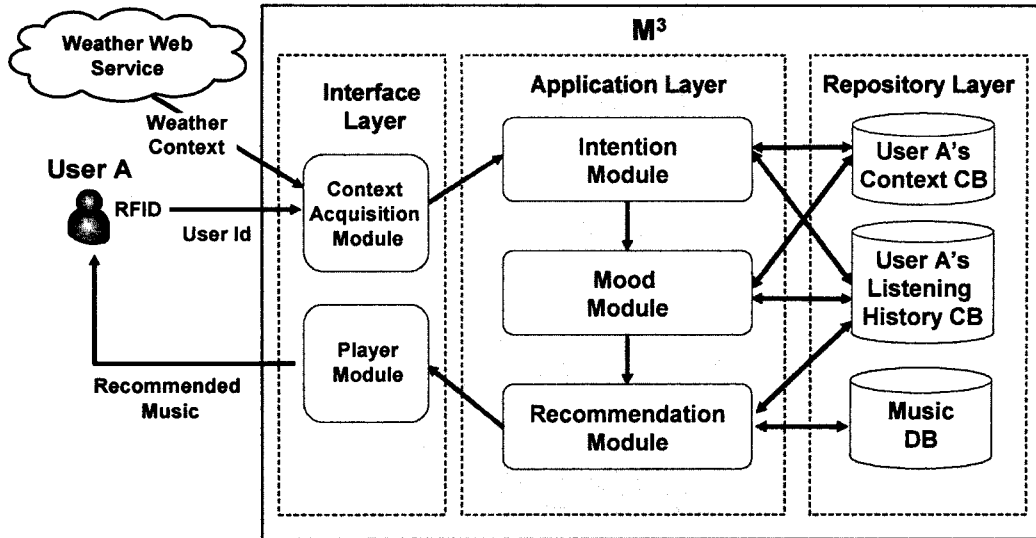
3. 상황 추론 음악추천시스템: M³

이 절에서는 본 연구에서 제안하는 상황 추론 음악추천시스템인 M³(Music for My Mood)의 구조 및 구성 모듈에 대해 설명한다.

M³는 <그림 3-1>과 같이 Interface Layer, Application Layer 그리고 Repository Layer의 3개의 Layer로 구성되어 있다. Interface Layer의 역할은 사용자의 인식, Context Data의 수집 그리고 추천된 음악을 사용자에게 전달하는 것이다. Repository Layer는 관련 사례와 데이터를 저장하고 관리한다. Context CB(Case Base)에는 과거의 Context Case가 저장되어 있다. Listening History CB에는 과거에 사용자가 청취했던 음악의 사례들이 저장되어 있다. Music DB에는 음악의 음원이 저장되어 있다. Application Layer는 사용자의 의도를 추론하여 음악을 추천하는 역할을 담당하는데, Intention Module, Mood Module, Recommendation Module로 구성되어 있고 각각의 기능은 다음과 같다.

1) Intention Module

Intention Module은 사용자가 음악을 듣고 싶어하는지 아닌지에 대한 사용자의 의도를 추론하는 모듈로서, 이를 위해 사용자가 과거에 어떤 상황 (Environmental Context)에서 노래를 들었는지에 대한 사례를 사용한다.



<그림 3-1> M³의 구조

2) Mood Module

Mood Module은 사용자가 어떤 분위기의 음악을 듣고 싶어하는지를 추론하는 모듈로서, 이를 위해 사용자가 과거에 어떤 상황에서 어떤 장르의 음악을 들었는지에 대한 사례를 사용한다.

3) Recommendation Module

음악추천 모듈은 사용자의 무드에 적합한 노래 리스트를 구성하여 사용자에게 추천하는 모듈이다.

M³의 음악추천 과정은 다음과 같다.

- 제 1 단계 (Context 감지): 사용자를 인식한 후에 외부의 상황정보를 입력 받는다. 본 연구에서는 Web Service로부터 입력 받는 계절, 월, 요일, 날씨, 기온 등의 속성을 사용한다.
- 제 2 단계 (청취의도 추론): 외부 상황정보를 기반으로 Intention Module은 사용자가 현재 음악을 청취할 의도가 있는지를 추론한다. 만약 추론결과가 사용자가 음악을 듣고 싶어한다고 판단되면 다음 단계인 사용자무드 추론 단계가 수행되고, 음악을 듣고 싶어하지 않는다고 판단되면 프로세스가 종료된다.
- 제 3 단계 (사용자 무드 추론): Mood Module은 인터페이스 모듈을 통해 입력 받은 정보를 기반으로 사용자가 어떤 분위기의 음악을 듣고 싶어하는 상황인지를 추론한다.
- 제 4 단계 (음악 추천): 제 3 단계의 추론결과를 기반으로 사용자가 선호할만한 음악들을 추천한다.

4. M³의 구현

4.1 사용 데이터

본 연구에서 사용한 데이터는 사용자의 음악 청취이력 데이터와 일기 데이터이다. 사용자의 음악 청취이력 데이터는 웹 상에서 Streaming Music Service를 제공하는 한 회사로부터 수집한 것으로서, 어떤 특정 사용자인 A가 6개월 동안, 즉 171일 동안 들었던 노래들에 대한 데이터를 담고 있다. 일기 데이터는 기상청으로부터 수집 되었으며, 일자별로 날씨, 최저기온, 최고기온, 평균기온 등에 대한 데이터를 담고 있다. 수집된 데이터는 세 개의 데이터 집합, 즉 Training Data Set (65.0%), Validation Data Set (17.5%), Test Data Set (17.5%)으로 나누어 각각 모델의 구축, 최적화 및 성능 평가에 사용하였다.

4.2 Intention Module의 구축

Intention Module을 구축하기 위해서, 수집된 데이터로부터 사용자 A가 노래를 들었던 날을 'Yes', 듣지 않은 날을 'No'로 하는 목표속성 'WantMusic'을 생성하였다. 사용 데이터에서의 WantMusic 속성 값의 분포는 <표 4-1>과 같다.

<표 4-1> WantMusic 속성값 분포

WantMusic	날짜 수	비율
Yes	144	84.2%
No	27	15.8%

Intention Module의 구축에는 7개의 입력 속성과 1개의 목표 속성이 사용되었는데, 사용된 입력속성은

<표 4-2>에서 보는 바와 같이 4개의 범주형 속성과 3개의 수치형 속성으로 구성되어 있다.

Decision Tree를 이용한 Intention Model을 모델 구축에는 C5.0 알고리즘을 사용하였다 [9]. 본 연구에서는 최적의 Decision Tree 모델을 찾기 위해 가지치기 엄격도를 45%에서 70% 사이로 변화시켰으며, 최종노드에 포함되는 최소 사례의 개수는 3으로 하였다. C5.0 Decision Tree 모델은 SPSS사의 Clementine8.1을 사용하여 구축하였다.

<표 4-2> Intention Module 구축을 위해 사용된 입력속성

속성 명	속성 설명	속성 타입
계절	봄 ~ 겨울	범주형
월	1월 ~ 12월	범주형
요일구분	월 ~ 일	범주형
날씨	맑음, 구름 조금, 구름 많음, 흐림, 비, 눈	범주형
평균기온	해당 일의 평균기온 (°C)	수치형
최고기온	해당 일의 최고기온 (°C)	수치형
최저기온	해당 일의 최저기온 (°C)	수치형

SVM 모델 구축을 위한 커널 함수로는 RBF(Radial Basis Function)를 사용하였다. RBF 커널 함수는 두 개의 파라미터 C 와 γ 의 설정이 필요하다. 본 연구에서는 최적의 SVM 모델을 찾기 위해 C 를 1, 5, 10, 25, 50, 100, 1000으로 변경해 가면서 그리고 각각의 C 값에 따라 γ 를 0.5, 1, 2, 5, 10, 50으로 변경해 가면서 모델을 구축하였다. SVM 모델은 LIBSVM2.82를 사용하여 구축하였다.

CBR 모델 구축에 사용되는 유사도 함수 (Similarity Function)는 식 (4-1)을 사용하였다. 여기서, N_i 는 새로운 사례의 i 번째 속성값; C_i 는 과거사례의 i 번째 속성값; n 은 속성의 개수; $f(N_i, C_i)$ 는 두 속성값 N_i 과 C_i 사이의 거리함수; W_i 는 i 번째 속성에 대한 가중치이다.

$$Similarity(N, C) = \frac{\sum_{i=1}^n f(N_i, C_i) \times W_i}{\sum_{i=1}^n W_i} \quad \text{식 (4-1)}$$

식 (4-1)의 거리함수 $f(\cdot)$ 는 두 사례의 속성간 유사도를 측정하기 위한 것으로, 본 연구에서는 비교 대상 속성이 수치형일 경우 식 (4-2)를 사용하였다. 여기서, Max 는 사례베이스에 있는 모든 사례의 i 번째 속성값 중 최대값이고, Min 은 최소값이다.

$$f(N_i, C_i) = \begin{cases} 1-d & \text{if } 0 \leq d \leq 1 \\ 0 & \text{if } d > 1 \end{cases} \quad \text{식 (4-2)}$$

여기서,

$$d = \frac{|N_i - C_i|}{Max - Min}$$

비교 대상 속성이 범주형인 경우에는 <표 4-3>에 제시된 기준에 따라 유사도 점수를 부여하였다.

<표 4-3> 범주형 속성간의 유사도 점수 부여 기준

유사도 점수 부여 기준						
계	신규 사례 / 과거 사례	봄	여름	가을	겨울	
		봄	1	0.2	0.5	0.2
		여름	0.2	1	0.2	0
		가을	0.5	0.2	1	0.2
		겨울	0.2	0	0.2	1
월	유사도	조건				
	1	$N_i = C_i$				
	0.5	N_i 과 C_i 의 차이 = 1 개월				
	0.2	N_i 과 C_i 의 차이 = 2 개월				
	0	Otherwise				
...	...					
...	...					

CBR 모델의 예측성능은 해(Solution)를 생성하기 위해 참조되는 최근접 이웃의 수 k 와 유사도 산출에 사용되는 속성들의 가중치로부터 많은 영향을 받게 된다. 따라서, 최적의 CBR 모델을 구축하기 위해서 k 값을 1, 3, 5, 7로 조정해 가면서 실험을 수행하였고, 가중치벡터의 경우에는, 무작위 방식에 의해 100개의 가중치벡터를 생성한 후, 각각을 적용한 모델에서 훈련데이터 집합에 대해 가장 좋은 예측성능을 보인 가중치벡터를 최종 모델의 가중치벡터로 사용하였다. CBR 모델은 Microsoft사의 Visual Basic 6.0을 사용하여 개발하였다.

본 연구에서는 모델의 예측성능 평가에 있어서 Training Data Set으로부터의 영향을 최소화하고 신뢰성을 확보하기 위해 10-Fold 교차검증(Cross Validation)을 수행하였다. Decision Tree 모델, SVM 모델 그리고 CBR 모델의 Validation Data Set에 대한 10-Fold 평균 적중률은 각각 90.6%, 90.2%, 93.3%이다. 각 기법들간의 예측 성능이 통계적으로 차이가 있는지를 검증하기 위해 <표 4-4>와 같이 Paired Sample t-Test를 수행하였다. 그 결과, CBR 모델이 유의수준 5%에서 SVM 모델보다, 그리고 유의수준

10%에서 Decision Tree 모델보다 예측 성능이 더 우수하였다. 한편, Decision Tree 모델과 SVM 모델간의 예측 성능은 통계적으로 차이가 없다고 나타났다.

<표 4-4> Paired Sample t-Test 분석 결과

분석 모델	t-값
CBR-SVM	4.617*
CBR-Decision Tree	1.728**
Decision Tree-SVM	0.260

(*: 유의수준 5%; **: 유의수준 10%)

그러므로 Intention Module은 CBR 기법으로 구축되었다.

4.3 Mood Module의 구축

Mood Module을 구축하기 위해서, 먼저 사용자의 무드를 세 개의 카테고리, 즉 'Slow Music', 'Fast Music', 그리고 'Any Music'으로 분류하였다. Slow Music은 부드럽고 조용하고 잔잔한 장르의 음악으로서 'Ballad'와 'R&B'가 여기에 속한다. Fast Music은 활기차고 템포가 빠르고 비트가 강한 장르의 음악으로서 'Rock/Metal'과 'Dance'가 여기에 속한다. Any Music은 사용자가 특별히 선호하는 장르가 없는 경우를 말한다.

어떤 날 'd'의 무드는 식 (4-3)에 의해서 계산되는 무드 점수 $M(d)$ 에 의하여 결정된다.

$$M(d) = \frac{\sum_i n_{id} \times G_i}{L_d}, \quad d = 1, \Lambda, K \text{ 식 (4-3)}$$

여기서 $i = \{Ballad, R \& B, Rock / Metal, Dance\}$

n_{id} : d 날에 들은 i 장르의 음악 개수.

G_i : i 장르의 점수.

$$G_i = \begin{cases} 0.1 & \text{if } i = 'Ballad' \\ 0.3 & \text{if } i = 'R \& B' \\ 0.7 & \text{if } i = 'Rock / Metal' \\ 0.9 & \text{if } i = 'Dance' \end{cases}$$

L_d : d 날에 들은 음악의 총 개수.

K : 날짜의 개수.

음악의 유형이 부드럽고 느리면 장르 점수 G_i 는 0에 가까워지고, 빠르면 1에 가까워진다. 사용자의 무드는 <표 4-5>에 제시된 조건에 의하여 결정된다. \bar{M} 는 모든 $M(d)$ 값들의 평균이다. 만일

$M(2007/08/14)$ 이 0.28이고 \bar{M} 가 0.5라면, 사용자의 무드는 Slow Music으로 판정된다.

<표 4-5> 사용자 무드의 결정

사용자 무드	조건
Slow Music	$M(d) - \bar{M} \leq -0.05$
Any Music	$-0.05 < M(d) - \bar{M} < 0.05$
Fast Music	$M(d) - \bar{M} \geq 0.05$

Mood Module도 CBR로 구현되었다. 최적의 CBR 모델을 Intention Module의 경우와 마찬가지로 방법으로도 출하였다. 그 결과, Validation Data Set에 대한 최적의 CBR 모델은 $k = 5$ 일 때이었고, 모델의 적중률은 63.3%이었다.

4.4 Recommendation Module의 구축

Mood Module에 의해서 사용자의 무드가 결정되면, Recommendation Module은 사용자의 무드에 적합한 음악을 추천하는데, 결정된 장르의 음악들 중에서 지난 1주일 동안에 가장 많이 들은 음악 15곡 (일반적으로 음악 CD 1장에 수록된 곡의 수)을 추천한다.

예를 들어, 사용자 A의 무드가 Slow Music으로 추론되었다면, Recommendation Module은 그녀가 지난 1주일 동안 들었던 음악들 중에서 Ballad와 R&B 15곡을 추천한다.

5. M³의 성능 평가

M³의 성능을 평가하기 위하여 비교 시스템 M³-C (=M³ minus Context reasoning)을 구축하였다. M³-C는 M³에서 Intention Module과 Mood Module, 즉 Context Reasoning 기능을 없앤 시스템이다. 두 시스템 모두 Microsoft Visual Basic 6.0으로 PC 상에서 구현되었다.

본 연구에서는 평가 측정치로 Accuracy(적중률)를 사용하였다. Accuracy는 식 (5-1)과 같이 계산하였는데, 추천된 음악들 중에서 실제로 청취한 음악의 비율의 평균을 말한다.

$$Accuracy = \left(\sum_{i=1}^N \frac{|R_i \cap L_i|}{|R_i|} \times 100 \right) / N \text{ 식 (5-1)}$$

여기서,

N : Test Data Set의 크기.

R_i : i 번째 Test Case에 대하여 추천한 음악의 집합.

L_i : i 번째 Test Case에 대하여 실제로 청취한 음악의 집합.

<표 5-1>은 Test Data Set에 대한 두 시스템의 Accuracy 비교를 보여주고 있다.

<표 5-1> 두 시스템의 Accuracy 비교

시스템	Accuracy
M ³ -C	73.6%
M ³	89.8%

<표 5-1>에서 보는 바와 같이, M³의 Accuracy는 89.8%로서 73.6%인 M³-C보다 16.2% 포인트만큼 높다. 이는 M³가 Context Data를 이용하여 사용자의 의도추론과 무드에 대한 추론을 함으로써 추천의 정확도가 증가한 것이다.

6. 결론

본 논문에서는 Context-Awareness의 개념을 접목한 추천시스템의 프레임워크를 제시하는 연구를 수행하였다. Context-Awareness 개념 중에서도, 사용자가 처한 환경인 장소라든가 시간 등을 파악하는 것이 아니라, 사용자가 어떤 의도를 가지고 있는가를 파악하는 사용자의 Context Reasoning 또는 Intention Reasoning에 초점을 맞추어 연구를 수행하였다.

본 연구에서 제시한 프레임워크의 실용성을 입증하기 위하여, 음악추천시스템 M³를 구축하였다. M³는 Context Reasoning 기능이 없는 비교 시스템인 M³-C에 비해서 월등히 향상된 성능을 보여 주었다. M³는 유비쿼터스 컴퓨팅 환경에서 Smart Home Application의 하나로 사용될 수 있다.

본 연구에서는 Context Data로서 날씨 관련 데이터를 주로 사용하였다. 향후에는 특정 음악을 청취한 시간, 사용자의 사생활과 관련된 특정일 등 다양한 Context Data를 활용하는 연구를 수행할 필요가 있다.

Acknowledgement

본 연구는 21세기 프론티어 연구개발 사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스 컴퓨팅 및 네트워크원천기반기술개발사업의 지원에 의한 것임.

참고문헌

- [1]. Aamodt, A. and E. Plaza, "Case-based Reasoning: Fundamental Issues, Methodological Variations, and System Approaches," *Artificial Intelligence Communication*, Vol. 7, No. 1, pp. 39-59, 1994.
- [2]. Benerecetti, M., P. Bouquet and M. Bonifacio, "Distributed Context-Aware System," *Human-Computer Interaction*, Vol. 16, No. 2, pp. 213-228, 2000.
- [3]. Cuddy, S., M. Katchabaw and H. Lutfiyya, "Context-aware Service Selection based on Dynamic and Static Service Attributes," *Wireless and Mobile Computing, Networking and Communications*, IEEE International Conference, 2005.
- [4]. Dey, A. K. and G. D. Abowd, "Towards a Better Understanding of Context and Context-Awareness," *Proceedings of CHI 2000 Workshop on the What, Who, Where, When, Why, and How of Context-Awareness*, Hague, Netherlands, pp. 1-6, 2000.
- [5]. Dey, A. K., "Understanding and Using Context," *Personal and Ubiquitous Computing*, Vol. 5, No. 1, pp. 4-7, 2001.
- [6]. Quinlan, R., *C5.0: An Information Tutorial*, RuleQuest, <http://www.rulequest.com/see5-unix.html>, 1998.
- [7]. Resnick, P. and H. R. Varian, "Recommender Systems," *Communications of the ACM*, Vol. 40, pp. 56-58, 1997.
- [8]. Schilit, B. N. and M. Theimer, "Disseminating Active Map Information to Mobile Hosts," *IEEE Network*, Vol. 8, No. 5, pp. 22-32, 1994.
- [9]. Schmidt, A., M. Beigl and H. W. Gellersen, "There is More to Context than Location," *Computers and Graphics*, Vol. 23, No. 6, pp. 893-901, 1999.
- [10]. Vapnik, V., *The Nature of Statistical Learning Theory*, Chapter 5, Springer-Verlag, New York, 1995.
- [11]. Weiser, M., "Hot Topics: Ubiquitous Computing," *IEEE Computer*, October 1993.