

데이터 마이닝 기법을 이용한 사용자 상황 추론 User's Context Reasoning using Data Mining Techniques

이재식^a, 이진천^b

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

^b 아주대학교 대학원 경영학과 박사과정
경기도 수원시 영통구 원천동 산 5번지
Tel: 031-219-2308, Fax: 031-219-1616, E-mail: giny777@empal.com

Abstract

The context-awareness has become the one of core technologies and the indispensable function for application services in ubiquitous computing environment. In this research, we incorporated the capability of context-awareness in a music recommendation system. Our proposed system consists of such components as Intention Module, Mood Module and Recommendation Module. Among these modules, the Intention Module infers whether a user wants to listen to the music or not from the environmental context information. We built the Intention Module using data mining techniques such as decision tree, support vector machine and case-based reasoning. The results showed that the case-based reasoning model outperformed the other models and its accuracy was 84.1%.

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 [18]. 유비쿼터스 컴퓨팅 환경에서는 사용자가 컴퓨터에게 직접 자신이 원하는 작업에 대한 명령을 하지 않아도, 컴퓨터가 알아서 조용히(Calm) 사용자가 원하는 작업을 해주어야 한다. 이것이 가능하기 위해서 컴퓨터는 사용자가 어떤 상황에 있는지를 인식할 필요가 있는데, 이러한 유비쿼터스 컴퓨팅의 특성을 상황인식(Context-Awareness)이라고 한다.

본 연구에서는 Context-Awareness의 개념을 상품 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 [10]. 만일 고객이 웹을 통하여 온라인 서점에 접속한다면, 그가 살펴보는 서적들의 정보를 이용하여 적절한 서적의 추천이 가능할 것이다. 하지만, 유비쿼터스 컴퓨팅 환경에서는 고객이 적극적으로 자신이 어떠한 정보를 원한다고 내비치기 전에, 컴퓨터가 먼저 고객의 의중을 파악하여 그에게 적절한 상품을 추천하여야 할 것이다.

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 구상하는 Context-Awareness에 의한 음악 추천 시스템의 전체적인 구조를 보여준다. 제 4절에서는 연구에 사용한 데이터의 소개 및 각 데이터 마이닝 기법의 적용과정을 설명한다. 제 5절에서 실험결과를 제시하고 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[12]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[14]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 사용자에 대한 정보, 사회적 환경과 믿음과 같은 추상적인 개념을 말한다. Dey와 Abowd[4]는 Context를 ‘어떤 객체가 처한 정황(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. 하지만, 상황을 이렇게 사례를 이용하여 정의하는 방법은 현실 문제에 적용하기가 어렵다. 정의에 포함되어 있지 않은 어떤 정보가 접수되면 그것을 이용하여 특정 상황이 발생했는지 아닌지를 판정할 수가 없기 때문이다.

Context-Awareness가 가능한 응용 프로그램들은, Context에 대한 정보를 감지함으로써, 사용자에게 Context에 대한 정보를 제공할 수 있고, 환경의 변화에 따라 프로그램 작동의 제어를 수정할 수도 있게 된다 [11]. Context-Awareness에 대한 초기 연구로는 Olivetti Research Lab에서 수행한 Active Badge System을 들 수 있다 [6, 16, 17]. 이 시스템은 사용자가 부착한 Badge를 통하여 그가 있는 곳을 인식하여 그 장소에 알맞게 응용 소프트웨어를 적절하게 조정하여 제공한다. 제록스의 Palo Alto Research Center에서도 유비쿼터스 컴퓨팅 실험과 더불어 Context-Awareness 모바일 컴퓨팅에 관한 초기의 연구가 수행되었다 [13]. Kofod-Petersen과 Aamodt[7]는 사용자가 처한 정황에 대한 정보를 사례기반 추론(CBR: Case-Based Reasoning) 기법의 사례로 포착하여 Context-Awareness 모바일 시스템에 적용하였다. Kumar 등[8]은 다중 CBR 접근방법을 제안하였다. 그들은 사용자에게 대한 상황을 인식하는 User Context CBR과 제품에 대한 상황을 인식하는 Product Context CBR을 활용하여 전자 상거래 상에서 적합한 제품 추천을 지원 하는 시스템을 개발하였다.

향후의 응용 프로그램들이 Context-Awareness 기능이 가능한 지능적 유비쿼터스 응용 프로그램이 되기 위해서 중요한 것들 중 하나는 사용자의 상황(User's Context)을 파악하는 프로그램이 되어야 한다는 것이다. 즉, 사용자의 의향을 파악해야 한다. 예를 들어, 어떤 남성이 백화점에 있다고 하자. 기존의 Context-Awareness 기능에 의해서, 장소, 주체, 시간 등의 Context를 정의할 수 있는 요소들이 파악될 것이고, 이에 따라 그에게 필요하다고 판단되는 남성복이나 넥타이 상품들이 추천될

것이다. 하지만, 이것은 표면적인 Context 정보에 의하여 수행된 불필요한 추천에 그칠 수가 있다. 만일 그가 여자친구에게 줄 선물을 사러 온 것이라면, 그에게는 여자 용품에 대한 추천이 수행되어야 하기 때문이다. 즉, 그의 과거 구매패턴 또는 그가 백화점에서 돌아다니는 매장의 종류를 분석하여 그의 의향을 파악할 필요가 있는 것이다. 이러한 연구주제를 상황 추론(Context Reasoning)이라고 부른다.

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 [9]. Decision Tree는 가지를 분리하는 방법, 즉 엔트로피(Entropy) 또는 이득률(Gain Ratio) 등에 따라 여러 종류의 알고리즘이 있다.

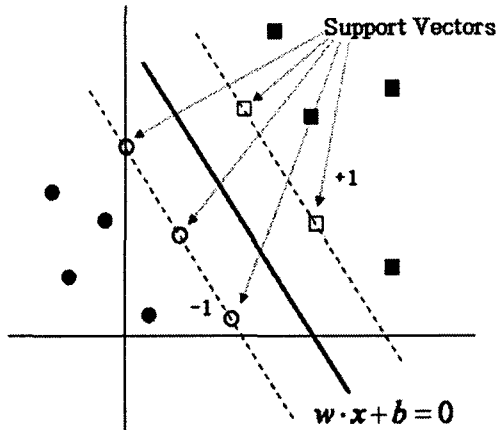
2.3. Support Vector Machine

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 [15].

두 개의 집단 $y_i \in \{-1, +1\}$ 로 분리된 입력값 $x_i \in R^n$, $i = 1, \dots, N$ 이 있다고 하자. Vapnik[15]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. <그림 2-1>에 나타난 바와 같이 학습자료 중에서 경계면 상의 점들을 Support Vector라고 한다.



<그림 2-1> 분리경계면과 Support Vectors

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.

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

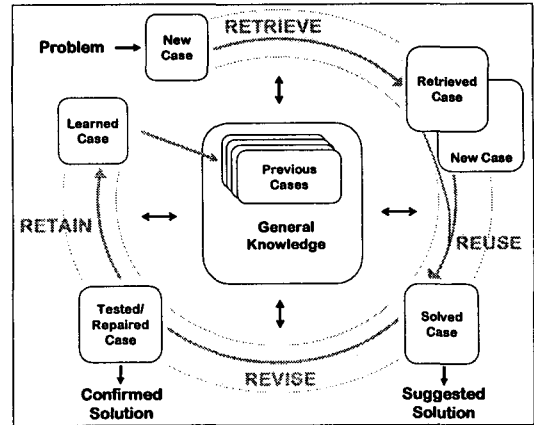
식 (2-2)에서 $K(\cdot)$ 는 커널함수이다. 커널함수는 이론적으로는 가능하나 실제로는 풀기 힘든 매핑 함수를 대신하여 데이터를 고차원 공간으로 이동시켜 특징 공간 내에 선형으로 분리 가능한 입력자료 집합을 만들어 주는 역할을 수행한다. 커널함수의 예로는 다항식 커널(Polynomial Kernel)과 RBF(Radial Basis Function) 커널을 들 수 있다.

2.4 사례기반 추론

사례기반추론(CBR: Case-Based Reasoning)의 기본 사상은 유사한 문제는 유사한 해법을 가진다는 것과, 한번 발생한 문제는 자주 발생할 수 있다는 것이다. 따라서 과거에 현재의 문제와 유사한 문제가 존재하였고 그것이 어떻게 해결됐는지를 안다면, 과거의 경험을 바탕으로 현재 문제의 해결책을 추론할 수 있다는 것이다. CBR은 과거에 축적된 정보만 있으면 어떤 문제라도 해결책을 제시할 수 있으므로 복잡하거나 비구조화된 문제 해결에도 적용이 가능하며, 새로운 사례의 저장으로 문제해결을 위한 지식기반을 지속적으로 갱신할 수 있다는 장점이 있다.

CBR 시스템의 문제해결 과정은 <그림 2-2>에서와 같이 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.

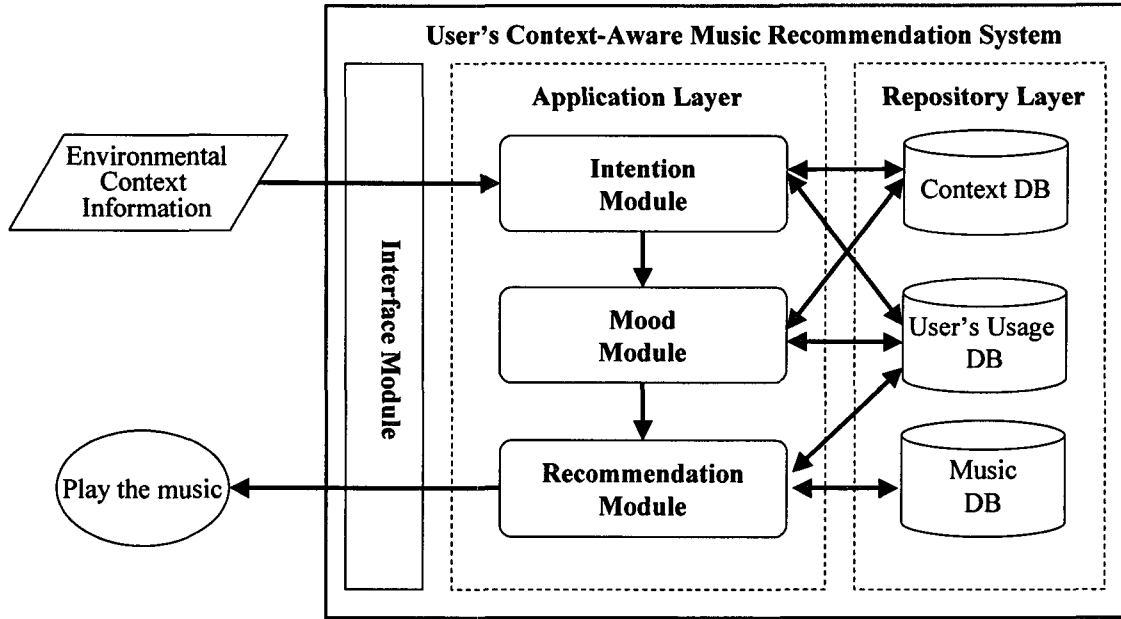


<그림 2-2> CBR 시스템의 문제 해결 과정

3. 사용자 상황인식 음악추천시스템

이 절에서는 본 연구에서 제안하는 외부환경정보 (Environmental Context)에 따른 사용자 상황인식 음악추천시스템(U_CaM_ReS: User's Context-Aware Music Recommendation System)의 구조 및 구성 모듈에 대해 설명한다. U_CaM_ReS는 음악취취의향 인식모듈, 사용자무드 인식모듈, 그리고 음악추천 모듈로 구성되며, 전체적인 구조는 <그림 3-1>과 같다. 본 연구에서는 U_CaM_ReS를 구성하는 세 개의 모듈 중 음악 취취의향 인식모듈의 실제 구현을 통해 그 유용성을 확인하고자 한다.

- 1) 음악취취의향 인식모듈 (Intention Module)
음악취취의향 인식모듈은 사용자가 음악을 듣고 싶어하는 상황인지 아닌지를 추론하는 모듈로, 이를 위해 사용자가 과거에 어떤 상황 (Environmental Context)에서 노래를 들었는지에 대한 이력 데이터를 사용한다.
- 2) 사용자무드 인식모듈 (Mood Module)
사용자무드 인식모듈은 사용자가 어떤 분위기의 음악을 듣고 싶어하는지를 추론하는 모듈로, 이를 위해 사용자가 과거에 어떤 상황에서 어떤 장르의 음악을 들었는지에 대한 이력 데이터를 사용한다.
- 3) 음악추천 모듈 (Recommendation Module)
음악추천 모듈은 사용자의 무드에 적합한 노래 리스트를 구성하는 모듈이다. 사용자 무드에 적합한 노래 리스트의 구성을 위해서는 단순 매핑 규칙 (Mapping Rule) 또는 모델에 의한 추천이 사용된다.



<그림 3-1> U_CaM_ReS의 구조

U_CaM_ReS에 의한 사용자 상황인식 음악추천 과정은 다음과 같다.

- 단계 1: 센서로부터 감지된 외부의 상황정보 및 사용자정보를 인터페이스 모듈을 통해 입력 받는다. 외부 상황정보로 계절, 월, 요일, 날씨, 기온 등 다양한 환경 속성들이 사용될 수 있다.
- 단계 2: 인터페이스 모듈을 통해 입력 받은 외부 상황정보를 기반으로 Intention Module은 사용자가 현재 음악을 듣고 싶어하는 상황인지 아닌지를 추론한다. 만약 추론결과가 사용자가 음악을 듣고 싶어하는 상황이면 다음 단계인 사용자무드 예측단계가 수행되고, 음악을 듣고 싶어하지 않는 상황이면 프로세스가 종료 된다.
- 단계 3: Mood Module은 인터페이스 모듈을 통해 입력 받은 정보를 기반으로 사용자가 어떤 분위기의 음악을 듣고 싶어하는 상황인지를 추론한다. 예를 들면, 사용자가 빠르고 경쾌한 댄스음악을 듣고 싶어하는 상황인지, 조용한 발라드음악을 듣고 싶어하는 상황인지를 추론하는 것이다.
- 단계 4: 단계 3에서의 추론결과를 기반으로 사용자가 선호할 노래들을 추천 한다.

4. 데이터 마이닝 기법을 이용한 Intention Module의 구현

4.1 사용 데이터

본 연구에서 사용한 데이터는 사용자 데이터와 일기 데이터이다. 사용자 데이터는 웹 상에서 스트리밍 음악 서비스(Streaming Music Service)를 제공하는 한 회사로부터 수집한 것으로서, 어떤 특정 사용자가 2005년 1월부터 6개월 동안 들었던 노래들에 대한 데이터를 담고 있다. 일기 데이터는 기상청으로부터 수집 되었으며, 일자별로 날씨, 최저기온, 최고기온, 평균기온에 대한 데이터를 담고 있다. 사용자 데이터에는 일자별로 청취한 노래와 해당 노래의 장르에 대한 데이터가 들어있으며, 총 14,373건의 데이터가 수집되었다.

4.2 목표속성의 정의 및 데이터 전처리

Intention Module을 구축하기 위해서, 수집된 데이터로부터 사용자가 노래를 들었던 날을 'Yes', 노래를 듣지 않은 날을 'No'로 하는 목표속성 'MusicOK'를 생성하였다. 그 결과, <표 4-1>과 같이 총 193일 중 144일은 노래를 들은 날로 MusicOK의 값이 'Yes'로 분류되었으며, 49일은 노래를 듣지 않은 날로 MusicOK의 값이 'No'로 분류되었다.

<표 4-1> 목표속성값 분포

MusicOK	사례 수	비율
Yes	144	74.6%
No	49	25.4%

본 연구에서는 수집된 사용자 데이터와 일기 데이터로부터 전처리 작업을 통해 모델 설계와 평가에 사용할 실험데이터를 구성하였다. 실험데이터에는 7개의 입력 속성과 1개의 목표속성이

있는데, 7개의 입력속성은 4개의 범주형 속성과 3개의 수치형 속성으로 구성되어 있다. <표 4-2>는 Intention Module 구축을 위해 사용된 입력속성이다.

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

속성명	속성 설명	속성 타입
계절	봄, 여름, 가을, 겨울	범주형
월	1월~12월	범주형
요일구분	1=월, 2=화, 수, 목 3=금, 4=토/일	범주형
날씨	맑음, 구름조금, 구름많음, 흐림, 비, 눈	범주형
평균기온	해당 일의 평균기온 (단위: °C)	수치형
최고기온	해당 일의 최고기온 (단위: °C)	수치형
최저기온	해당 일의 최저기온 (단위: °C)	수치형

4.3 모델 설계

Decision Tree를 이용한 Intention Model을 모델 구축할 때에, 본 연구에서는 가지의 분리기준으로 이득률(Gain Ratio)을 사용하는 C5.0 알고리즘[9]을 사용하였다. Decision Tree 모델의 과잉적합을 방지하기 위해서는 적절한 가지치기와 최종노드(Leaf node)에 포함될 사례의 개수에 대한 적절한 설정이 필요하다. 본 연구에서는 최적의 Decision Tree 모델을 찾기 위해 가지치기 엄격도를 45%에서 70% 사이로 변화시켰으며, 최종노드에 포함되는 최소 사례의 개수는 3으로 하였다. 가지치기 엄격도의 경우, 값을 크게 설정할수록 Decision Tree는 단순하게 되고 값이 작을수록 복잡한 형태를 가지게 된다. C5.0 Decision Tree 모델은 SPSS사의 Clementine8.1을 사용하여 구축하였다.

SVM 모델 구축을 위한 커널 함수(Kernel Function)로는 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.

CBR을 이용한 Intention Model을 구축하기 위해서는 현재의 상황과 유사한 과거상황을 찾는 데 사용되는 유사도 함수(Similarity Function)의 정의가 선행되어야 한다. 식 (4-1)은 CBR 모델에서 두 사례간의 유사도를 측정하기 위해 사용되는 일반적인 형태의 유사도 측정 함수로, 본 연구에서도 이 함수를 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.

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

where

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

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

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. 가중치벡터의 각 속성별 가중치는 0과 1사이의 실수값을 갖도록

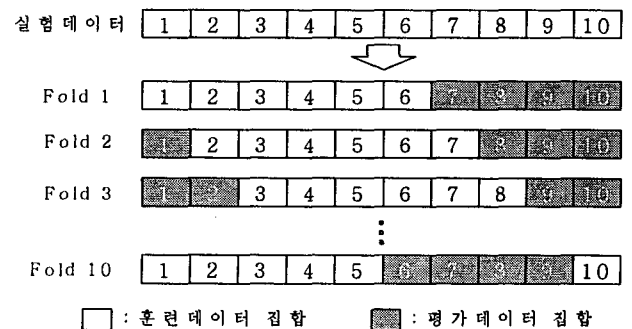
설정하였다. CBR 모델은 Microsoft사의 Visual Basic 6.0을 사용하여 개발하였다.

<표 4-3> 범주형 속성간의 유사도 점수 부여 기준

속성명	유사도 점수 부여 기준						
계절	신규사례 / 과거사례	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		
	유사도	조건					
1	$N_i = C_i$						
0.5	Distance between N_i and $C_i = 1$ Month						
0.2	Distance between N_i and $C_i = 2$ Months						
0	Otherwise						
요일구분	신규사례 / 과거사례	1	2	3	4		
	1	1	0.5	0	0		
	2	0.5	1	0	0		
	3	0	0	1	0.5		
	4	0	0	0.5	1		
날씨	신규사례 / 과거사례	맑음	구름 조금	구름 많음	흐림	비	눈
	맑음	1	0.5	0.2	0	0	0
	구름 조금	0.5	1	0.5	0.2	0	0
	구름 많음	0.2	0.5	1	0.5	0	0
	흐림	0	0.2	0.5	1	0.5	0.5
	비	0	0	0	0.5	1	0.5
	눈	0	0	0	0.5	0.5	1

5. 실험 및 평가

본 연구에서는 모델의 예측성능 평가에 있어서 학습 데이터로부터의 영향을 최소화하고 신뢰성을 확보하기 위해 10-Fold 교차검증(Cross Validation)을 수행하였다. 10-Fold 교차검증을 위한 실험데이터의 구성은 <그림 5-1>의 방법을 사용하였다. 각각의 실험데이터 집합은 훈련데이터 집합(Training Data Set)과 평가데이터 집합(Test Data Set)으로 나누었으며, 그 비율은 6:4로 하였다. 훈련데이터 집합은 Decision Tree와 SVM의 경우에는 모델의 구축에, CBR의 경우에는 사례베이스로 사용되었다. 평가데이터 집합은 최종 선정된 최적모델의 일반화 성능을 평가하기 위한 목적으로 사용되었다.



<그림 5-1> 10-Fold 실험데이터 구성방법

<표 5-1>는 실험을 통해 얻은 최적의 Decision Tree 모델, SVM 모델 그리고 CBR 모델의 평가데이터 적중률이다.

<표 5-1> 각 모델의 평가데이터에 대한 적중률
단위: %

Fold 번호	Decision Tree	SVM	CBR
Fold 1	85.5	81.6	81.6
Fold 2	78.9	78.9	79.0
Fold 3	79.2	80.5	85.7
Fold 4	78.2	82.1	84.6
Fold 5	78.5	79.7	83.5
Fold 6	78.8	83.8	85.0
Fold 7	80.8	80.8	85.9
Fold 8	83.1	83.1	87.0
Fold 9	85.5	82.9	85.5
Fold 10	88.0	80.0	82.7
평균	81.7	81.3	84.1

<표 5-1>에서 보듯이, Decision Tree, SVM, CBR 모델의 평균 적중률은 각각 81.7%, 81.3%, 84.1%로서 CBR 모델이 가장 높은 성능을 보였다. 각 기법들간의 예측 성능이 통계적으로 차이가 있는지를 검증하기 위해 Paired Sample t-Test를 수행하였다. 그 결과 <표 5-2>와 같이 CBR 모델과 SVM 모델간의 t-값은 4.617로 유의수준 5%에서의 t-값인 1.833 보다 크게 나타났으며, CBR모델과 Decision Tree 모델간의 t-값은 1.728로 유의수준 10%에서의 t-값인 1.383보다 크게 나타났다. 이것은 CBR 모델이 유의수준 5%에서 SVM 모델보다, 그리고 유의수준 10%에서 Decision Tree 모델보다 예측 성능이 더 우수함을 의미한다. 한편, Decision Tree 모델과 SVM 모델간의 예측 성능은 통계적으로 차이가 없다고 나타났다.

<표 5-2> Paired Sample t-Test 분석 결과

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

(*: 유의 수준 5%; **: 유의수준 10%)

<표 5-3>은 CBR 모델의 10-Fold 평균 Confusion Matrix이다.

<표 5-3> CBR 모델의 Confusion Matrix
단위: 일

실제 \ 예측	Yes	No
	Yes	53.3
No	8	11.6

<표 5-3>의 데이터로 계산해보면, Yes를 Yes로 예측한 적중률인 민감도(Sensitivity)는 92.5%이고, No를 No로 예측한 적중률인 특이도(Specificity)는 59.2%이다. 제 4절의 <표 4-1>에서 목표속성이 Yes인 날의 비율이 74.6%이므로, 예측모델의 구축 없이 항상 Yes라고만 할 때의 적중률은 74.6%이고, 항상 No라고만 할 때의 적중률은 25.4%이다. 하지만 예측 모델을 구축함으로써 민감도를 92.5%로 향상시켰다. 실체가 No인데 Yes로 잘못 예측한 경우, 즉 1-특이도가 40.8%로 다소 높지만, 이 경우에는 U_CaM_Res의 두 번째 모듈인 Mood Module에서 음악 장르를 적절하게 선택함으로써 사용자의 불편함을 다소 해소시킬 수 있을 것이다.

6. 결론

본 논문에서는 Context-Awareness의 개념을 음악 추천시스템에 접목시키는 연구를 수행하였다. Context-Awareness 개념 중에서도, 사용자가 처한 환경인 장소라든가 시간 등을 파악하는 것이 아니라, 사용자가 어떤 생각을 가지고 있는가를 파악하는 User's Context Reasoning에 초점을 맞추어 연구를 수행하였다.

본 연구에서는 먼저 Context-Awareness에 기반한 음악 추천시스템의 전체 구조를 제시한 후에, 그 중에서 사용자의 Context를 추론하는 모듈만을 구현하여 유용성을 검증하였다. 계절, 월, 날씨 등의 데이터를 이용하여 사용자가 과연 음악을 들을 생각이 있는지 없는지를 예측하는 모델을 구축하였는데, 우리가 사용한 세 가지 데이터 마이닝 기법들, 즉 의사결정나무, Support Vector Machine, 사례기반추론 중에서 사례기반추론 기법이 가장 우수한 적중률인 84.1%를 보였다.

한계점과 함께 향후 연구과제는 다음과 같다. 첫째, 민감도에 비해서 특이도가 현저하게 낮다. 이것은 데이터의 양이 부족한 이유에서 기인한 면이 없지 않다. 그러므로 데이터의 보충도 필요하겠지만, 하이브리드 기법의 개발을 통하여 특이도 향상에 노력할 필요가 있다. 둘째, 사용자의 음악청취의향을 예측하기 위한 속성으로 우리가 선정한 계절, 월, 날씨 등에 대해서 그 선정의 타당성을 검토하지 못하였다. 향후 연구에서는 다양하게 속성의 종류를 보강한 후에 적절한 선정작업을 거쳐서 속성을 선정할 필요가 있다. 마지막으로, 후속 연구에서 사용자무드 인식모듈과 음악추천모듈을 구축하여 Context-Awareness에 기반한 음악 추천시스템을 완성할 예정이다.

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*, The 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]. Harper, R., Lamming, M. G. and Newman, W. M., "Locating Systems at Work: Implications for the Development of Active Badge Applications," *Interacting with Computers*, 4, pp. 343-363, 1992.
- [7]. Kofod-Petersen, A. and A. Aamodt, "Case-Based Situation Assessment in a Mobile Context-Aware Systems," *Workshop on Artificial Intelligence for Mobile Systems (AIMS2003)*, Seattle, October, 2003.
- [8]. Kumar, P., S. Gopalan and V. Sridhar, "Context enabled Multi-CBR based Recommendation Engine for E-Commerce," *Proceedings of IEEE International Conference on e-Business Engineering (ICEBE'05)*, pp. 237-244, 2005.
- [9]. Quinlan, R., *C5.0: An Information Tutorial*, RuleQuest, <http://www.rulequest.com/see5-unix.html>, 1998.
- [10]. Resnick, P. and H. R. Varian, "Recommender Systems," *Communications of the ACM*, Vol. 40, pp. 56-58, 1997.
- [11]. Salber, D., A. K. Dey, R. J. Orr and G. D. Abowd, "Designing For Ubiquitous Computing: A Case Study in Context Sensing," *GVU Technical Report GIT-GVU*, pp. 99-129, 1999.
- [12]. Schilit, B. N. and M. Theimer, "Disseminating Active Map Information to Mobile Hosts," *IEEE Network*, Vol. 8, No. 5, pp. 22-32, 1994.
- [13]. Schilit, B. N., N. L. Adams and R. Want, "Context-Aware Computing Applications," *Proceedings of the Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, December 1994.
- [14]. 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.
- [15]. Vapnik, V., *The Nature of Statistical Learning Theory*, Chapter 5, Springer-Verlag, New York, 1995.
- [16]. Want, R., A. Hopper, V. Falcao and J. Gibbons, "The Active Badge Location System," *ACM Transactions on Information Systems*, Vol. 10, No. 1, pp. 91-102, 1992.
- [17]. Ward, A., A. Jones and A. Hopper, "A New Location Technique for the Active Office," *IEEE Personal Communications*, Vol. 4, No. 5, pp. 42-47, 1997.
- [18]. Weiser, M., "Hot Topics: Ubiquitous Computing," *IEEE Computer*, October 1993.