

웹과 모바일 환경을 연동한 추천시스템

A Recommendation System using the Web and Mobile Environment

이세일¹, 이상용²

¹ 충남 공주시 공주대학교 컴퓨터공학과

E-mail: lsilhr@kongju.ac.kr

² 충남 공주시 공주대학교 컴퓨터공학부(교신저자)

E-mail: sylee@kongju.ac.kr

요 약

정보의 홍수 속에서 살고 있는 사용자들은 좀 더 편리하게 정확한 서비스를 추천 받기를 원하고 있으며, 이러한 욕구들에 부응하여 추천시스템은 꾸준히 발전하고 있다. 유비쿼터스 컴퓨팅 환경에서는 사용자 환경과 사용자 자신의 컨텍스트 정보를 이용하여 추론하며, 그 결과를 사용자에게 서비스하고 있다. 그러나 실시간으로 획득된 컨텍스트 정보가 사용자에게 양질의 서비스를 제공하기 위하여 부족하거나 컨텍스트 정보를 모델링하는 방법에 문제가 있을 때에는 서비스의 질이 낮아질 수 있다.

본 논문에서는 이러한 문제점을 해결하기 위하여 모바일 환경에서 실시간으로 센서를 통하여 획득될 수 있는 컨텍스트 정보를 수집하고, 수집된 컨텍스트 정보는 하위수준의 정보를 상위수준의 정보로 모델링한다. 모델링된 컨텍스트 정보는 다시 정량화 단계 후, 웹 환경의 사용자 평가 정보와 결합하여 서비스를 추천한다. 이를 통하여 시스템은 유비쿼터스 환경에서 추천을 위한 양질의 컨텍스트 정보 부족 문제를 해결하였으며, 사용자에게 적합한 서비스를 제공할 수 있었다.

Key Words : 컨텍스트 정보, 추천시스템, 사용자 평가 정보, 유비쿼터스 컴퓨팅, 정량화

서 론

오늘날 우리는 대중매체들의 발달로 인하여 정보의 홍수 속에서 살고 있다. 이 다양한 정보들은 때때로 우리에게 노력과 시간을 필요로 하게 한다. 이러한 문제점을 해결하고자 추천시스템이 도입되었으며, 사용자들의 눈높이가 높아지면서 추천시스템은 더욱더 발전하여 왔다.

웹상의 추천시스템은 사용자들의 평가 정보나 사용자의 로그 정보 및 이력 정보를 이용하여 추천하고 있다. 그러나 이러한 방식의 문제점은 사용자들로부터 평가되어진 정보들의 부족과 방대한 정보들로 인하여 시스템에 무리가 따른다[1].

유비쿼터스 컴퓨팅 환경에서는 사용자 상태나 주변 환경의 컨텍스트 정보를 묵시적으로 얻을 수 있기 때문에 사용자들로부터 명시적으로 평가 값을 받지 않는다는 장점을 가지고 있다. 또한 초기 평가 정보의 부족 문제를 해결할 수 있고, 방대한 컨텍스트 정보들 중 필요한 정보들만 저장하고 나머지는 제거하기 때문에 많은 정보들의 축적 문제를 해결할 수 있다[2]. 그러나 사용자에게 양질의 서비스를 제공하기 위한 실시간 컨텍스트 정보가 부족하거나 컨텍스트 정보를 잘못된 방법으로 모델링할 때에는 서비스의 질이 낮아질 수 있다.

본 논문에서는 위와 같은 문제점을 해결하기 위

하여 모바일 환경에서 실시간으로 센서를 통하여 획득할 수 있는 컨텍스트 정보를 수집하고, 수집된 컨텍스트 정보는 하위수준의 정보를 상위수준의 정보로 모델링한다. 모델링된 컨텍스트 정보는 다시 정량화 단계 과정 후, 웹 환경의 사용자 평가 정보와 결합하여 서비스를 추천한다. 이를 통하여 이 시스템은 유비쿼터스 환경에서 추천을 위한 양질의 컨텍스트 정보 부족 문제를 해결하였으며, 사용자에게 적합한 서비스를 제공할 수 있었다.

본 논문의 구성은 2장에서 관련 연구, 3장에서는 실시간 컨텍스트 정보와 사용자 평가 정보를 이용한 추천 시스템, 4장에서는 실험 및 평가한 내용을 기술하였고 5장에서는 결론에 대하여 기술한다.

2. 관련 연구

2.1 컨텍스트 인식

컨텍스트(Context)는 유비쿼터스 컴퓨팅 환경에서 가장 많이 사용하는 단어 중 하나이다. 컨텍스트는 연구자들의 정의에 따라 조금씩 다르지만, 사용자와 주변의 환경에 대한 정보를 말한다. 컨텍스트를 분야별로 정리하면 다음과 같다. 사용자의 상황, 컴퓨팅 시스템 상황, 컴퓨터와 사용자 상호 이

력 상황, 물리적 환경 상황 그리고 기타 상황 등으로 나눌 수 있다[3].

사용자가 처하고 있는 상황에 따라 컨텍스트를 얻어내고, 그 컨텍스트를 이용하여 사용자에게 알맞은 서비스를 제공하는 과정을 컨텍스트 인식 기술(Context-Awareness Technology)이라고 한다. 컨텍스트 인식 서비스 미들웨어들은 센서들로부터 컨텍스트를 얻어내어 높은 수준으로 추론하여 사용자에게 알맞은 서비스를 제공한다. 잘 알려진 구조로는 SOCAM[4], Context-Toolkit[5], CoBrA[6] 등이 있다.

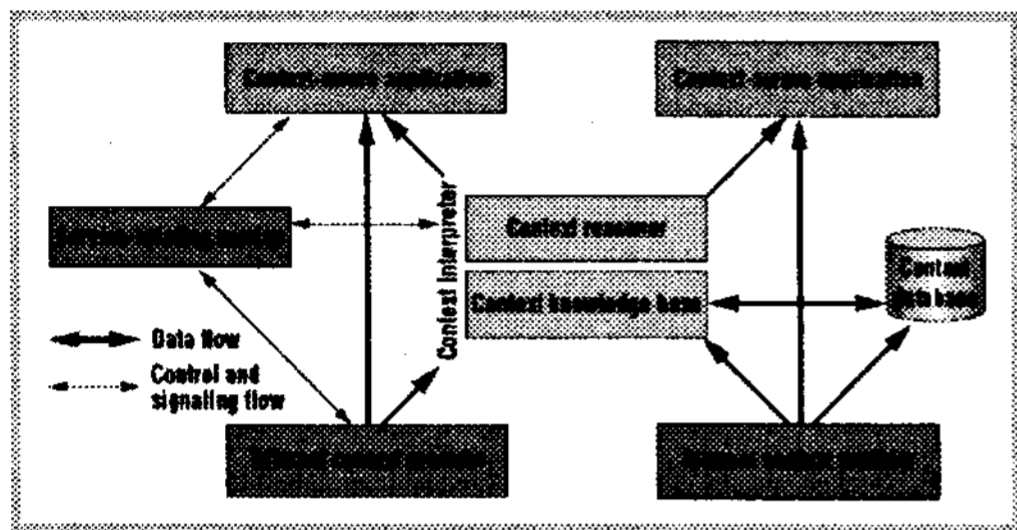


그림 1. SOCAM의 구조

SOCAM은 온톨로지 기반의 컨텍스트 모델을 사용하고, 컨텍스트 인식 모바일 서비스를 구축하고, OSGI에서 구동된다. 그리고 OWL 기반의 온톨로지를 사용한다.

2.2 추천시스템

추천시스템(Recommender Systems)은 자동화 기술을 이용한 정보 필터링 방법으로, 방대한 양의 정보 중에서 사용자들에게 알맞은 서비스만을 제공한다. 추천시스템들 중 가장 많이 이용되는 방법은 협력적 필터링(Collaborative Filtering) 기법으로, 다수 사용자와 다수 항목을 이용하여 이웃들과 상관관계를 이용하여 비슷한 성향을 가진 사용자를 기초로 추천하는 방법이다[7]. 그러나 초기 평가 문제(Early Rate Problem), 자료의 희소성 문제(Sparsity Problem) 그리고 확장성 문제(Scalability Problem)를 가지고 있다. 이러한 문제점을 해결하기 위하여 k-means 클러스터링 알고리즘[8], 협력적 필터링과 내용 기반 방법의 결합[9] 그리고 베이지안 기법[10] 등 다양한 방법으로 개선하려고 노력하고 있다.

3. 실시간 컨텍스트 정보와 사용자 평가 정보를 이용한 추천 시스템

외부로부터 수집된 실시간 컨텍스트 정보들은 즉시 사용할 수 없다. 이러한 컨텍스트는 가공 처리된 후 다음 단계로 이동되어 응용 서비스에 사용될 컨텍스트만 수집된다. 수집된 컨텍스트 정보는 추천에 사용될 수 있는 추론에 사용될 수 있도록 가공되며, 마지막 단계인 서비스 추천 단계로 이동하여 사용자에게 서비스를 추천한다.

그림 2는 실시간으로 얻을 수 있는 컨텍스트 정보와 웹으로부터 수집된 사용자 평가 정보를 결합하여 서비스를 추천하는 시스템 구성도이다. 평가에 사용

할 수 있는 컨텍스트 정보가 부족할 경우에는 서비스의 질이 떨어지는 것을 보완하고자 웹에서 사용자들이 명시적으로 평가한 평가값을 결합한다. 이 값들을 이용하여 추론하게 되면 평가의 질을 개선할 수 있다.

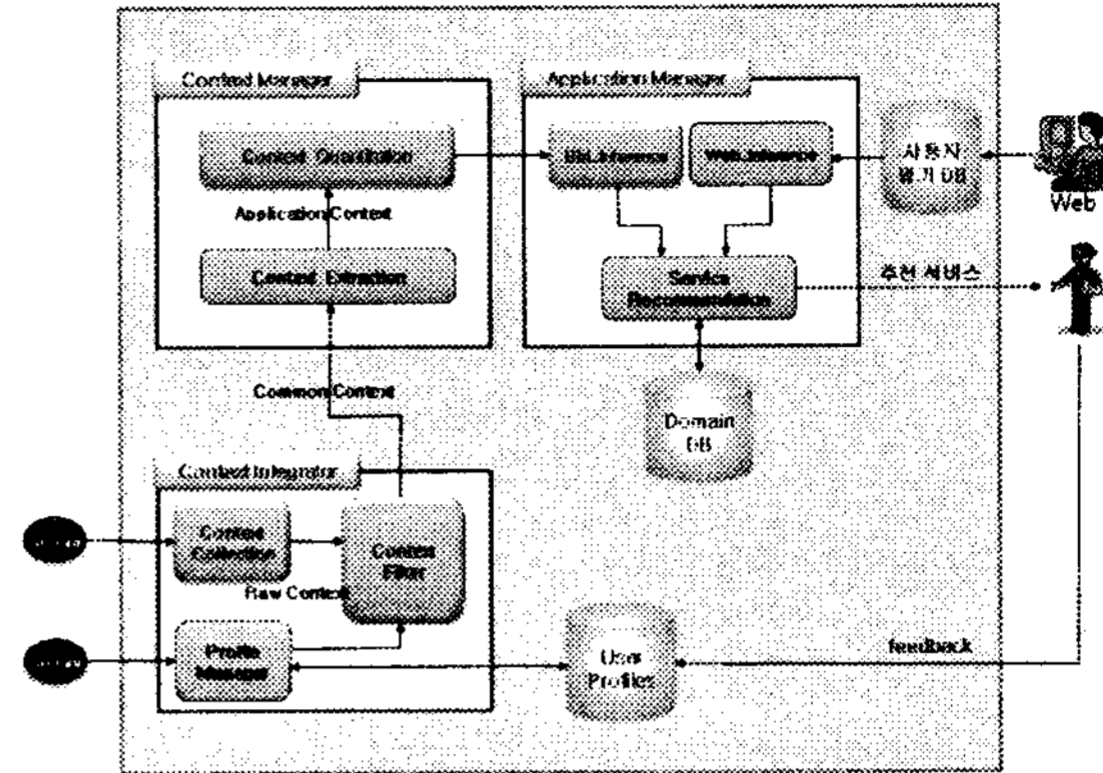


그림 2. 시스템 구성도

위 시스템의 작동 순서는 아래와 같다.

① Context Integrator Module은 실시간 컨텍스트 정보와 사용자 정보를 입력 받는다. 가공되지 않은 컨텍스트 정보는 1차로 평가하여 가공된 Common Context를 생성한다.

② Context Manager Module은 Common Context를 응용 서비스에 사용할 수 있는 컨텍스트만 다시 필터링하며, 필터링된 컨텍스트 정보를 Application Context라 한다. Application Context는 추론에 사용할 수 있도록 정량화를 사용한다.

③ Application Manager Module 단계에서는 실시간 컨텍스트 정보와 웹상에서 명시적으로 평가된 사용자 평가 정보를 각각 협력적 필터링하여 평균을 구한 후, 사용자에게 서비스를 추천한다.

3.1 Context Integrator Module

외부의 센서로부터 실시간으로 얻게 된 컨텍스트 정보는 Raw Context 정보로 가공되지 않는 상태를 유지하고 있다. 이 정보는 Pure P2P 방식을 사용하여 다른 사용자들로부터 얻어낸 정보들이다. 이 Raw Context 정보는 Context Filter에 의해 분류 작업을 사용하여 불필요한 컨텍스트 정보를 필터링하여 1차적으로 가공되어 진다. 가공된 컨텍스트 정보는 Common Context라 불리며 Context Manager Module로 이동하게 된다.

Profile Manager는 사용자의 위치 컨텍스트 정보, 상황 컨텍스트 정보, 환경 컨텍스트 정보의 확인이 가능하다. 그리고 User Profiles DB를 읽어 선호도 파악과 사용자와의 동행인 정보를 얻을 수 있다. 동행인 정보는 전화번호의 주소록 그룹명을 확인하여 나와 주변 인물의 관계로 파악할 수 있다.

본 논문에서는 멀티영화관에서 사용자의 컨텍스트 정보와 주변 사용자들의 컨텍스트 정보를 이용하여 영화를 추천하는 시스템을 개발한다.

3.2 Context Manager Module

Context Integrator Module에서 생성된 Common Context는 응용 서비스에 사용하기에 적당하지 않다. Context Extraction에서는 응용에 필요한 컨텍스트 정보만을 수집하여 관리하며, Application Context라 불리는 2차로 가공된 컨텍스트 정보를 생산한다. 여기에 필요한 컨텍스트 정보는 표 1과 같다.

표 1. 서비스에 필요한 컨텍스트 정보

종 류	내 용
ID	사용자 전화번호
나이	나이
동행인	친구, 애인, 등으로 분류
성별	남/여
장르	영화장르
제목	영화제목
장소	서비스 장소
시간	서비스 시간

Context Quantification에서는 상대적인 정량화 방법을 사용하여 Application Context 정보를 Application Manager에서 사용할 수 있도록 수치화한다. 상대적 정량화는 절대적으로 분류되어 적절하지 못한 그룹에 분류되는 방법을 상대적인 방법으로 분류하여 서비스 받을 사용자가 적절한 그룹에 분류되게 하는 과정이다. 일반적으로 10대, 20대, 30대... 등으로 분류해 놓고 서비스를 받고자하는 사용자의 나이가 어떤 그룹에 속할 것인가를 결정한다. 결정된 정보는 알맞은 그룹에 편입되어 추천을 하는데 도움을 주고 있다. 그러나 다음과 같은 문제점이 있다. 예를 들어 39세의 나이가 일반적으로 30대에 적용되어 사용되지만, 30대 보다는 40대의 경향에 더 가깝다고 할 수 있다. 이러한 이유 때문에 상대적 분류 방법을 사용하여 그룹화하는 것이 정확성을 높일 수가 있다. 표 2는 상대적 분류 방법으로 분류하는 방법을 보여 주고 있다.

또한 서비스 받을 사용자와 친밀도가 높을수록 우수한 값을 부여하고 서비스 받을 사용자와 친밀도가 떨어지면 작은 수치를 부여한다.

표 2. 상대적 정량화의 분류 방법

색 인	분 류	가 중 치
1	~14	2
2	15~19~24	3
3	25~29~34	4
4	35~39~44	5
5	45~49~54	4
6	55~59~64	3
7	65~	2

예를 들어 39세의 서비스 받을 사용자와 43세의 일반 사용자가 있다고 한다. 일반적인 분류 방법은 30대와 40대의 전혀 다른 그룹에 속하지만 상대적 정량화 방법을 하면 표 2와 같이 같은 그룹에 속하므로 가중치는 가장 높은 5를 부여한다. 즉 같은 그룹에 속하는 사용자들은 높은 값을 부여하지만, 멀어질수록 낮은 값을 부여한다.

3.3 Application Manager Module

Ubi_Inference에서는 Context Quantification로부터 가공된 실시간 컨텍스트 정보를 이용하여 가중치를 부여한다. 이 값들은 사용자에게 적합한 서비스를 제공하기 위하여 협력적 필터링 방법에 적용한다. 우선 피어슨 상관 계수(Pearson Correlation Coefficient)를 이용하여 유사도를 구하고, 평가값을 구한다.

Web_Inference는 웹상의 사용자들로부터 자신이 본 영화를 평가하여 사용자 평가 DB에 저장한다. 이 저장된 정보도 피어슨 상관 계수를 이용하여 유사도와 평가값을 구한다.

웹으로 평가 받은 정보는 안정적이지만 서비스된 시간이 짧아 평가 자료가 부족하면, 평가 정보의 부족 문제로 잘못된 서비스를 추천하는 치명적인 약점을 가지고 있다. 또한 평가에 사용될 실시간 컨텍스트 정보의 부족도 부정확한 추천이 될 수 있다.

Service Recommendation에서는 이러한 문제점을 해결하고자 Ubi_Inference에서 실시간 컨텍스트 정보를 이용하여 협력적 필터링 방법을 구하고 Web_Inference도 사용자 평가 정보를 이용하여 협력적 필터링 방법을 적용한다. 구해진 값은 산술 평균을 구하고 그 값으로 예측한다. 예측된 결과값을 이용하여 도메인 DB에 있는 추천된 영화 정보를 사용자에게 제공한다.

협력적 필터링에 사용된 유사도식(식 1)과 예측식은(식 2) 같다.

$$W_{x,y} = \frac{\sum_{a=1}^n (r_{x,a} - \bar{r}_x)(r_{y,a} - \bar{r}_y)}{\sqrt{\sum_{a=1}^n (r_{x,a} - \bar{r}_x)^2} \sqrt{\sum_{a=1}^n (r_{y,a} - \bar{r}_y)^2}} \quad (1)$$

$W_{x,y}$ 는 사용자 x와 사용자 y의 가중치이다. $R_{x,a}$ 는 사용자 x가 컨텍스트 정보 y에 대하여 가중치가 부여된 값이며, \bar{r}_x 는 x의 선호도 평균값이다. a는 x와 y가 모두 선택한 컨텍스트 정보이며, n은 컨텍스트의 총 개수이다.

$$P_{x,b} = \frac{\sum_{y=1}^n w(x,y)(r_{y,b} - \bar{r}_y)}{\sum_{y=1}^n w(x,y)} \quad (2)$$

$P_{x,b}$ 는 사용자 x와 컨텍스트 정보 b에 대한 선호도를 예측한 값이다. $w(x,y)$ 는 사용자 x와 사용자 y의 유사도 가중치이고, $r_{y,b}$ 는 사용자 y가 컨텍스트 정보 b에 대하여 평가된 값이다. \bar{r}_y 는 사용자 y의 선호도 평균값이고, n은 결정된 이웃의 수이다.

4. 실험 및 평가

본 논문은 펜티엄 IV, 2.8Ghz, 512MB의 환경을 사용한다. 서버의 운영체제는 리눅스를 사용하고 DB는 Microsoft SQL Server를 사용한다. Web의 개발 환경은 톰캣 환경에 JAVA, JSP를 사용하며, PDA는 Windows Mobile 2003 Se이며, MS Visual Studio 2005, C#, .NET Framework 2.0을 이용하여

설계하고 실험하였다.

실험 내용은 설문조사 방법을 하였다. 설문조사 인원은 남자 241명과 여자 186명으로 총 427명을 조사하여 관련성을 조사하였다. 서비스 추천에 사용된 컨텍스트 정보는 사용자 ID, 나이, 동행인, 성별, 장르, 제목, 장소 그리고 시간 등을 사용한다. 나이는 상대적인 분류방법을 이용하여 분류하였고, 동행인의 경우에는 누구와 동행했는가에 따라 영화의 장르가 결정되는 것을 알 수 있었다. 또한 남녀의 구분과 영화 보는 시간에 따라 영화의 취향이 변하는 것도 알 수 있었다.

정량화 단계 이후 사용되는 선호도 값은 1, 2, 3, 4, 5 단계로 표현하였으며, 가장 선호하지 않는 값은 1을 부여했으며, 선호도가 높으면 5를 부여하였다.

본 논문의 평가 방법으로는 평균 절대 오차 방법인 MAE(Mean Absolute Error)를 사용하였다. 사용자의 평가 정보와 예측된 예측값을 비교한다. 식 (3)은 실제 사용자와 예측 사용자와의 예측 값의 차이를 나타낸다. 사용자가 평가한 선호도를 $\{R_1, R_2, \dots, R_n\}$ 이라 하고, 예측된 선호도는 $\{V_1, V_2, \dots, V_n\}$ 이라 한다. N은 총 예측 회수이며, E는 평가된 값이며 값이 작을수록 좋다.

$$|E| = \frac{|R_1 - V_1| + |R_2 - V_2| + \dots + |R_n - V_n|}{N} \quad (3)$$

본 논문에서 제안한 시스템(CF_WC)을 실험한 결과 일반적인 협력적 필터링 방법을 이용한 방법(CF)과 실시간 컨텍스트 정보의 정량화 단계를 개선한 협력적 필터링(CFI) 보다 우수함을 보였다.

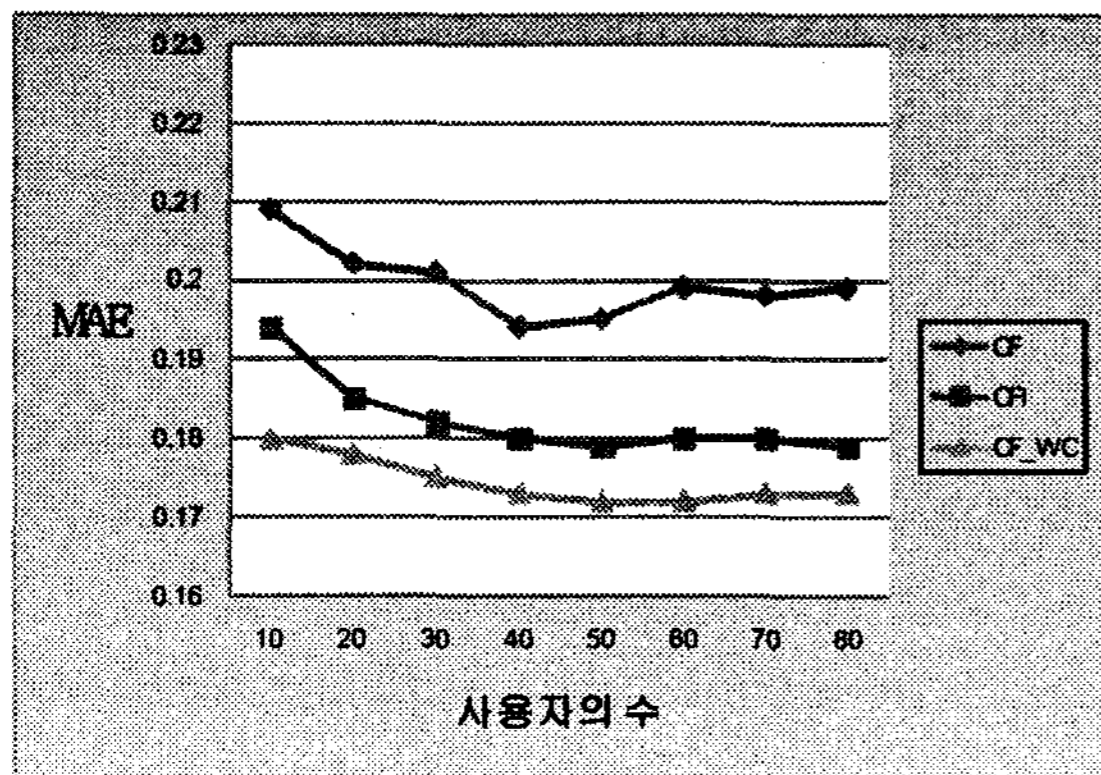


그림 3. 사용자 수에 따른 성능 평가

5. 결론

유비쿼터스 환경에서는 각종 센서 네트워크들을 이용하게 되면서 다양한 컨텍스트 정보들을 얻을 수 있다. 그러나 컨텍스트 정보들을 잘못된 방법으로 가공할 경우 추천의 질이 떨어지는 문제점을 가지고 있다. 이러한 문제점을 해결하기 위하여 센서로부터 들어온 컨텍스트 정보를 Context Filter에서 가공하여 불필요한 컨텍스트 정보는 필터링하며, Context Extraction에서는 응용에 사용될 수 있도록 2차적인

필터링 작업을 사용함으로써 서비스 추천에 사용할 수 있도록 컨텍스트 정보를 제공하였다. 두 번째 문제점은 사용자에게 알맞은 서비스를 추천하기 위한 컨텍스트 정보의 부족 문제이다. 이것을 해결하기 위하여 웹에서 사용자들로부터 실제로 입력된 사용자 평가 정보와 실시간 컨텍스트 정보를 가공하여 각각 협력적 필터링하였고, 예측된 결과값은 산술평균을 이용하였다.

본 시스템은 일반적인 협력적 필터링 방법보다 14.4%로 우수했고, 실시간 컨텍스트 정보의 정량화 단계를 개선한 협력적 필터링(CFI) 보다는 4.5% 우수한 것을 확인할 수 있었다.

참 고 문 헌

- [1] 김병만, 이경, "항목 속성과 평가 정보를 이용한 혼합 추천 방법", 정보과학회논문지, Vol.31, No.12, 2004.
- [2] 이세일, 이상용, "사용자 프로파일 정보와 컨텍스트 정보를 이용한 협력적 필터링", 한국퍼지 및 지능 시스템학회 논문지, vol.16, No.1, pp.336-339, 2006.
- [3] 윤효근, "모바일 P2P 환경에서 실시간 컨텍스트 인식 및 서비스를 위한 다중 에이전트 시스템", 박사학위 논문, 2006.
- [4] T. Gu, H. K. Pung, D. Q. Zhang. Towards an OSGi-Based Infrastructure for Context-Aware Applications in Smart Homes. IEEE Pervasive Computing, Vol. 3, Issue 4, 2004.
- [5] A.K.Dey and G.D.Abowd, "Towards a Better Understanding of Context and Context-Awareness", GVU Technical Report GIT-GVU-99-22. Submitted to the 1st International Symposium on Handheld and Ubiquitous Computing (HUC '99), June 1999
- [6] Chen, H. An Intelligent Broker Architecture for Pervasive Context-Aware Systems. PhD thesis, University of Maryland, Baltimore County, 2004.
- [7] 이형동, 김형주, "협업 필터링 추천시스템에서 취향 공간을 이용한 평가 예측 기법", 정보과학회논문지, Vol.34, No.5, 2007.
- [8] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, John T. Riedle, "Application of Dimensionality Reduction in Recommender System A Case Study", ACM WebKDD 2000 Web Mining for E-Commerce Workshop, 2000.
- [9] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, "Combining Collaborative Filtering with Personal Agents for Better Recommendations", In Proceedings of National Conference on Artificial Intelligence, pp.439-446, 1999.
- [10] K. Miyahara, M. J. Pazzani, "Collaborative Filtering with the Simple Bayesian Classifier", Proc. of the International Conference on Artificial Intelligence, pp.679-689, 2000.