

동적 추천 기능이 있는 음악 추천 서버 구현¹⁾

김형일, 박준태, 김종일, 김경섭, 김용욱, 김준태
동국대학교 컴퓨터공학과

e-mail:{hikim, juntae, kji1204, kyskks, yukim, jkim}@dongguk.edu

Implementation of a Music Recommendation Server with Dynamic Recommendation Capability

Hyungil Kim, Juntae Park, Jongil Kim, Kyungsup Kim, Yonguk Kim, Juntae Kim
Dept. of Computer Engineering, Dongguk University

요 약

본 논문에서는 동적 추천 기능이 있는 자바 기반 음악 추천 서버를 소개한다. 본 논문에서 소개하는 추천 서버는 각 음악 사이의 유사도를 그래프로 저장하는 그래프 기반 협동적 여과 방식을 사용하며, 독립된 서버로 구성되어 클라이언트와 정해진 프로토콜에 따라 통신하도록 구현되었다. 또한 이 추천 서버는 특정 사용자나 특정 음악의 정적인(static) 성향뿐 아니라 시간에 따라 달라지는 동적인(dynamic) 성향에 맞는 추천도 가능하도록 설계되었다. 정적 성향이란 어떤 사용자가 가지고 있는 음악에 대한 기본적인 취향을 나타내고, 동적 성향이란 특정한 상황이나 분위기에 따라 유동적으로 변하는 성향을 의미한다. 본 논문에서 소개하는 추천 서버는 정적 성향과 동적 성향에 대하여 각각 추천할 수 있는 기능을 가지고 있으며, 이러한 기능을 바탕으로 실제 사용자들의 음악 다운로드 데이터를 이용하여 추천 데모 사이트를 구축하였다.

1. 서론

추천 시스템은 사용자의 선호도를 분석하고, 아이템들에 대한 사용자의 선호도를 예측하여 책, 영화, 음악, 새로운 기사, 웹 페이지 등과 같은 아이템들을 추천한다. 여러 가지 추천 방법 중 가장 성공적으로 적용되어온 추천 방법은 협동적 여과(Collaborative Filtering, CF)를 기반으로 한 추천이다. 협동적 여과 기반 추천 시스템은 아이템들에 대한 각 사용자들의 평가 정보를 이용한다. 가장 일반적인 추천 방식은 사용자들 사이의 평가 정보를 비교하여 유사 사용자를 추출하고, 아이템에 대한 유사 사용자의 선호도를 기반으로 특정 아이템에 대한 사용자의 선호도를 예측하는 것이다. 협동적 여과 기법은 아이템의 내용 정보를 필요로 하지 않기 때문에 내용을 분석하기 어려운 음악이나 영화 같은 아이템을 추천할 수 있다 [2][5][7].

본 논문에서는 이러한 추천 시스템의 하나로서 개인별 취향에 맞는 음악을 추천하는 음악 추천 서버를 소개한다. 본 논문에서 소개하는 추천 서버는 협동적 여과 방법을 기반으로 하되, 확장성(scalability)이 개선된 그래프 기반 추천 방식을 사용한다. 그래프 기반 추천 방식은 각 음악 사이의 유사도를 그래프로 저장하고 관리함으로써 추천 계산 시간이 사용자 수나 음악 수에 비례하여 늘어나지 않는 추천 방법이다.

또한 이 추천 서버는 특정 사용자나 특정 음악의 정적인(static) 성향뿐 아니라 시간에 따라 달라지는 동적인(dynamic) 성향에 맞는 추천도 가능하도록 설계되었다. 정적 성향이란 어떤 사용자가 가지고 있는 음악에 대한 기본적인 취향을 나타내고, 동적 성향이란 특정한 상황이나 분위기에 따라 유동적으로 변하는 성향을 의미한다. 예

를 들어 과거부터 계속 주로 발라드 풍의 곡을 즐겨 듣는 사람이 있다면 그것이 그 사람의 정적인 성향으로서 발라드 풍의 곡을 추천해야 할 것이다. 그러나 경우에 따라서는 이러한 정적인 취향과 다른 곡들을 듣고 싶을 때도 있을 수 있다. 예컨대 평소 발라드를 듣던 사람이 신나는 일이 있어 흥겨운 댄스 뮤직을 듣고 싶어진다거나, 댄스 장르의 노래를 즐겨 듣던 사람이 늦은 밤 분위기 있는 발라드 곡을 듣고 싶은 경우도 있는데, 이렇게 상황에 맞추어 시간에 따라 변하는 것이 동적 성향이다.

본 논문에서 소개하는 추천 서버는 가장 최근에 들은 음악들에 대한 선호도를 바탕으로 추천을 수행하도록 하는 기능을 구현함으로써 추천 결과가 동적 성향을 따라갈 수 있도록 설계하였다. 이 추천 서버는 자바로 개발되었으며 독립된 서버로 구성되어 클라이언트와 정해진 프로토콜에 따라 통신하도록 구현되었다. 이 추천 서버를 이용하여 실제 사용자들의 음악 다운로드 데이터를 기반으로 정적인 추천과 동적인 추천을 수행하는 음악 추천 데모 사이트를 구축하였다.

2. 관련 연구

협동적 여과 기반 추천 시스템은 많은 사용자로부터의 아이템에 대한 평가를 이용하는 기법이다. 사용자 기반(user-based) 협동적 여과는 사용자들 사이의 평가 정보를 비교하여 유사 사용자들을 추출하고, 유사 사용자들의 선호도를 기반으로 사용자의 아이템 선호도를 예측한다. 사용자 유사도는 피어슨(Pearson) 상관관계나 벡터 유사도와 같은 통계적 기법에 의해 계산된다. 유사도가 계산되고 나면 식 (1)에서와 같이 아이템에 대해 유사한 사용자들의 평가에 대한 가중 평균을 계산하여 사용자에게 대한 아이템 선호도를 예측한다. 식 (1)에서 P_{ai} 는 아이템 i 에

1) 본 연구는 서울시 보유키술 사업화 지원 사업으로부터 연구비를 지원받았음.

대한 사용자 a 의 선호도 예측 값이고, $r_{u,i}$ 는 사용자 u 의 아이템 i 에 대한 평가 값이다. $S_{a,u}$ 는 사용자 a 와 사용자 u 사이의 유사도이고, n 은 유사 사용자들의 총 수이다.

$$P_{a,i} = \frac{\sum_{u=1}^n (s_{a,u} \cdot r_{u,i})}{\sum_{u=1}^n s_{a,u}} \quad (1)$$

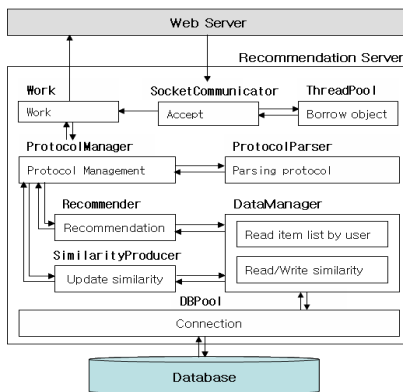
GroupLens[7]와 같은 협동적 여과 시스템에서는 사용자의 선호도를 예측하는데 상관관계(correlation)기반 기법이 사용되었고, 다양한 변형된 기법들이 추천 시스템의 정확도 향상을 위해 제안되었다. Breese 등[2]은 각 아이템에 얼마나 많은 사용자가 평가를 수행했느냐에 따라 서로 다른 가중치를 적용하는 방법 등을 제안하였고, Herlocker 등[5]은 다양한 유사도 계산 방식과 유사도 가중치 방법에 대한 실험을 수행한 바 있다. Billsus와 Pazzani[1], Sarwar 등[9]은 충분한 정보가 없을 때 사용자의 선호도 예측을 위해 속성 추출 기술을 적용하는 방법과, 사용자-아이템 선호도 행렬의 차원을 줄이는 방법을 제안하였다.

최근에는 Sarwar[10], Deshpande[4] 등에 의해 아이템 사이의 유사도를 이용하여 사용자의 증가에 따른 계산 복잡도 문제를 해결하면서 추천의 질을 높일 수 있는 아이템 기반(item-based) 협동적 여과 알고리즘이 제안되었고, Condliff[3], Popescul[8], Hoffman[6] 등에 의해 확률 이론을 바탕으로 하는 다양한 모델 기반(model-based) 추천 방법에 대한 연구도 수행되었다.

본 논문에서 소개하는 추천 서버는 협동적 여과 방법을 기반으로 하되, 확장성(scalability)이 개선된 그래프 기반 추천 방식을 사용한다. 그래프 기반 협동적 여과는 기본적으로는 아이템 기반(item-based) 협동적 여과 알고리즘에 가까우나, 선호도 정보를 누적하여 아이템들 사이의 연관관계를 그래프로 표현함으로써 오프라인에서 아이템들의 유사도 계산을 수행할 필요가 없으며, 대용량 데이터에서 실시간 추천이 가능한 장점이 있다.

3. 추천 서버의 구성과 기능

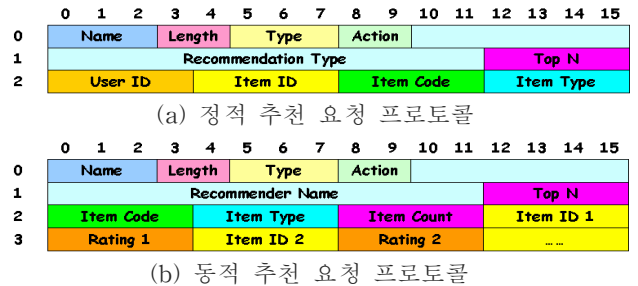
3.1 시스템 구성과 통신 프로토콜



[그림 1] 시스템 구성도

그림 1은 음악 추천 서버의 시스템 구성을 나타낸다. 웹 서버는 사용자의 필요에 따라 추천 서버와 통신하여 추천 요청을 하고 추천 결과를 받아 제공한다. 추천 서버는 추천 엔진, 소켓 통신기, 프로토콜 관리기(ProtocolManager), 데이터 관리기(DataManager), 데이터베이스 풀(DBPool), 추천기(Recommender), 유사도계산기(SimilarityProducer) 등으로 구성된다. 데이터베이스에는 사용자별 선호도 정보와 함께 추천 대상 아이템들 사이의 연관도가 가중치 그래프 형태로 저장되고 갱신된다.

본 논문에서 소개하는 추천 서버는 클라이언트와 정해진 통신 프로토콜에 따라 소켓 통신을 하도록 설계되었다. 그림 2에 정적 추천 요청 프로토콜과 동적 추천을 위한 가변 길이의 프로토콜 구조를 나타내었다.



[그림 2] 프로토콜 구조

통신 프로토콜은 크게 헤더와 몸통 부분으로 구성된다. 헤더의 첫 3바이트는 프로토콜 이름을 저장하며, 다음 2바이트는 프로토콜의 길이를 저장한다. 다음 3바이트는 메시지 타입을 명시하는 것으로서 해당 프로토콜이 요청을 위한 프로토콜('REQ')인지 응답을 위한 프로토콜('RES')인지를 나타낸다. 다음 2바이트는 작업 내용으로, 단일 아이템 기반 추천('RI'), 아이템 리스트 기반 추천('RL'), 사용자 기반 추천('RU') 혹은 선호도 갱신('PS') 등을 나타낸다.

프로토콜의 몸통 부분은 추천 방식(알고리즘)의 이름을 명시하는 Recommendation Type, 추천 받을 아이템 수를 의미하는 Top N, 추천 대상이 되는 사용자 ID, 아이템 ID, 추천 받을 대상을 나타내는 Item Code(예: 음악, 앨범, 또는 아티스트 추천), 추천 받을 아이템의 카테고리를 나타내는 Item Type(예: 팝, 발라드 등)으로 구성된다. 사용자 기반 추천을 수행하는 경우에는 사용자 ID가 명시되고, 단일 아이템 기반 추천을 하는 경우에는 아이템 ID가 명시된다. 선호도 갱신을 요청하는 경우에는 Item Type 부분에 갱신할 가중치 값인 Weight를 저장한다.

동적 성향에 기반한 추천을 위한 추천 요청 프로토콜의 몸통 부분은 가변 길이로써, 추천 받을 아이템 수인 Top N, Item Code, Item Type, 추천 기반이 될 아이템 수를 나타내는 Item Count, 각 아이템들의 ID와 평가값(rating)을 명시한다.

그림 3은 이러한 프로토콜에 따른 몇 가지 메시지의 예를 보인 것이다. 그림 3-a는 아이디가 2396005인 사용자에게 장르 제한 없이(Item Type=0) 음악(Item Code=1)을 10(TopN=10)곡 추천해달라고 요청하는 예이다. 그림 3-b는 아이디가 2396005인 사용자의 아이템 ID가 87130인 음악(Item Code=1)에 대한 평가값을 1만큼 갱신하도록 요청하는 예이다. 그림 3-c는 아이템 ID가 86771인 음악은

good(+1) 평가를 주고 아이템 ID가 84913인 음악은 bad(-1) 평가를 했을 때 장르 제한 없이 1곡 추천 해달라고 요청하는 예이다.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	IRP	43	REQ	RI											
1	GraphBased														
2	2396005	0	1	0											

(3.a) 사용자 기반 아이템 추천 요청 메시지

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	IRP	43	REQ	PS											
1	GraphBased														
2	2396005	87130	1	1											

(3.b) 선호도 갱신 요청 메시지

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	IRP	55	REQ	RL											
1	GraphBased														
2	1	0	2	86771											
3	1	84913	-1												

(3.c) 2개 아이템 기반 동적 추천 요청 메시지

[그림 3] 메시지의 예

3.2 추천 알고리즘

이 추천 서버에서 사용하는 그래프 기반 추천 기법은 다수 사용자들의 음악 구매 기록(download), 음악 실행 횟수(play count), 개인 음악 목록(play list) 등 암시적 선호 정보들로부터 음악 사이의 연관 관계를 가중치 그래프로 표현하여 저장하고, 이러한 그래프를 이용하여 추천 대상 사용자에 대한 음악을 추천하는 방법이다. 그래프 기반 추천 기법은 사용자의 암시적 선호 정보가 그래프 구조에 누적됨으로써 추천 실행 시간에 유사도 계산 등의 과정을 수행할 필요가 없어 추천 계산 시간이 사용자의 수나 아이템의 수에 비례하여 늘어나지 않는다.

그래프 모델을 사용하여 음악을 추천하는 과정은 크게 네 단계로 나눌 수 있다. 첫째, 각 음악 사이의 연관 관계를 그래프 모델로 표현하고 저장한다. 둘째, 추천 대상 사용자의 취향에 맞을 가능성이 있는 음악들을 저장된 그래프를 이용하여 찾아낸다. 셋째, 찾아진 음악들에 대하여 해당 사용자의 선호도 예측 값을 그래프 데이터를 이용하여 계산한다. 넷째, 산출된 예측 값을 이용하여 추천 대상 음악에 순위를 정하여 사용자에 적합한 음악을 추천한다.

3.3 정적 성향에 기반한 추천

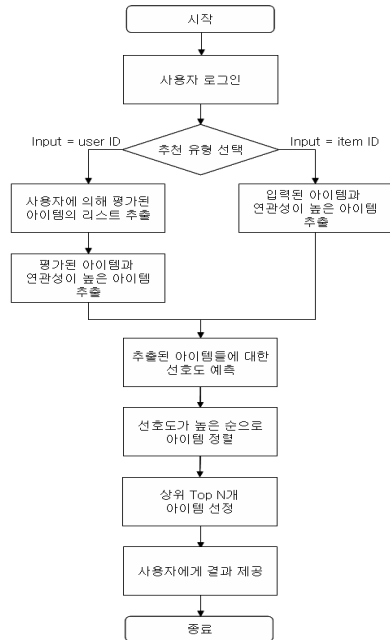
정적 성향에 기반한 추천에서 사용자는 로그인한 후, 크게 두 가지 종류의 추천을 요청할 수 있다. 하나는 사용자의 기존 선호도 정보를 이용한 추천 방식이고, 다른 하나는 임의의 아이템을 선택하여 그 아이템과 유사한 아이템을 추천 하는 방식이다.

사용자 선호도를 이용한 추천 방식은 사용자의 아이템에 대한 반응 정보(예: 클릭, 다운로드, 스트리밍 서비스 등)를 활용하여 아이템을 추천하는 방식으로, 사용자가 과거에 선호한 아이템들을 기반으로 그래프 모델에서 연관성이 높은 아이템들을 추출하고 추출된 아이템들에 대한 선호도를 예측한다. 예측된 선호도 값에 따라 아이템을 정렬한 후, 상위 N개를 선정하여 사용자에게 아이템을 추천하게 된다.

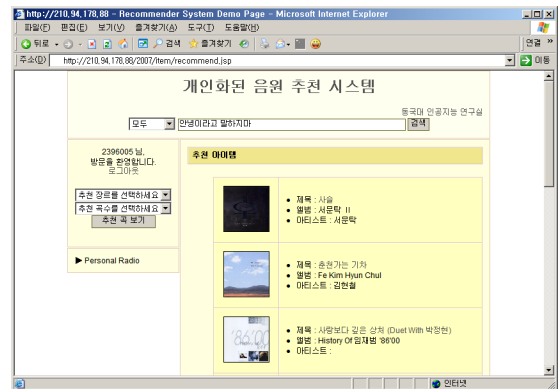
특정 아이템에 대한 유사 아이템 추천은 사용자의 선호도 정보를 활용하지 않고 사용자가 선택한 아이템만을

가지고 유사한 아이템을 추천하는 방식으로 추천 흐름은 사용자 선호도 방식과 동일하다.

그림 4는 정적인 추천 과정을 나타내며, 그림 5는 본 논문에서 소개한 추천 서버를 이용하여 정적인 추천을 하는 데모 화면이다.



[그림 4] 정적 추천 과정



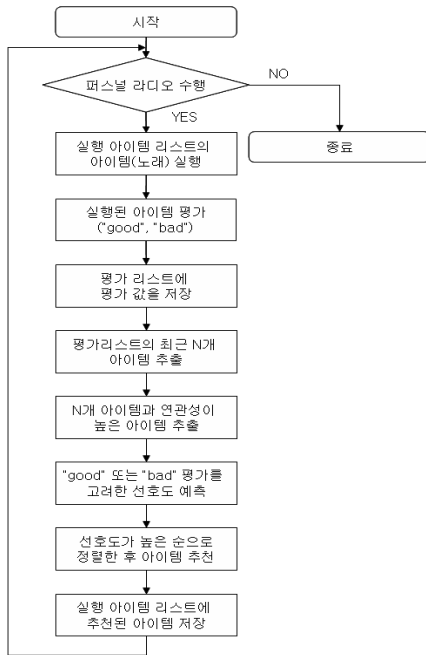
[그림 5] 정적 추천 데모 화면

3.4 동적 성향에 기반한 추천

동적 성향에 기반한 실시간 추천은 사용자가 최근에는 K개의 아이템을 기반으로 이루어진다. 본 논문에서는 추천 서버를 이용하여 사용자의 동적인(일시적인) 성향에 따라 연속적으로 선곡이 되어 플레이되는 퍼스널 라디오라는 데모를 구현하였다.

퍼스널 라디오는 사용자의 동적 성향을 따라가기 위하여 임시 선호도 평가 리스트와 실행 아이템 리스트를 유지한다. 실행 아이템 리스트는 앞으로 실행될 아이템을 임시로 저장하는 리스트로 현재 퍼스널 라디오를 사용하는 사용자의 동적 성향에 따라 추천된 아이템들을 담고 있다. 임시 선호도 평가 리스트는 실행된 아이템에 대한 사용자의 최근 평가 정보를 저장한 리스트로, 동적 추천에 사용된다.

퍼스널 라디오에는 현재 추천되어 실행되고 있는 음악에 평가를 수행할 수 있는 버튼(good button, bad button)들이 있다. 들던 곡이 마음에 들면 Good 버튼을 누르고, 마음에 들지 않으면 Bad 버튼을 눌러 현재의 음악 취향을 표현할 수 있다. 퍼스널 라디오는 최근 K개 곡에 대한 이러한 선호도를 바탕으로 다음 대기곡을 추천 받음으로써 사용자의 기본적인 취향과 현재의 선호 성향이 다르거나 선호 성향이 시간에 따라 변하여도 현재의 성향에 맞는 곡을 추천할 수 있다. 그림 6은 이러한 동적인 추천 과정을 나타내며, 그림 7은 퍼스널 라디오의 실행 화면을 나타낸다.



[그림 6] 동적 추천 과정



[그림 7] 퍼스널 라디오의 실행 화면

4. 결론

본 논문에서는 정적 추천과 동적 추천 기능이 있는 자바 기반 음악 추천 서버를 소개하였다. 본 논문에서 소개한 추천 서버는 각 음악 사이의 유사도를 그래프로 저장하는 그래프 기반 협동적 여과 방식을 사용하여 확장성이 개선된 그래프 기반 추천 방식이다. 그래프 기반 추천 방식은 각 음악 사이의 유사도를 그래프로 저장하고 관리함으로써 추천 계산 시간이 사용자 수나 음악 수에 비례하

여 늘어나지 않는 추천 방법이다.

추천 서버는 과거 선호도를 기반으로한 특정 사용자에 대한 추천과, 특정 음악에 대한 유사한 음악의 추천 등 정적인 추천을 수행할 수 있고, 또한 사용자가 가장 최근에 선호한 아이템들을 바탕으로 추천을 수행하도록 하는 기능을 구현하여 시간에 따라 달라지는 동적인 성향을 따라가는 동적 추천도 수행할 수 있다.

이 서버는 자바로 개발되었으며 독립된 서버로 구성되어 클라이언트와 정해진 프로토콜에 따라 통신하도록 구현되었다. 이 추천 서버를 이용하여 실제 사용자들의 음악 다운로드 데이터를 기반으로 정적인 추천을 수행하는 데모와 동적인 추천을 수행하는 퍼스널 라디오 등을 구축하였다. 향후 동적인 추천에 대해서는 사용자의 일시적인 취향이나 변화하는 성향에 추천 결과가 얼마나 잘 맞추어가는지를 다양한 실험을 통하여 분석할 필요가 있다.

참고문헌

- [1] D. Billsus and M. J. Pazzani, "Learning Collaborative Information Filters," *Proceedings of the 15th International Conference on Machine Learning*, 1998.
- [2] J. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, 1998.
- [3] M. Condliff, D. Lewis, D. Madigan, and C. Posse, "Bayesian Mixed Effects Models for Recommender Systems," *Proceedings of ACM SIGIR '99 Workshop on Recommender Systems*, 1999.
- [4] M. Deshpande and G. Karypis, "Item-Based Top-N Recommendation Algorithms," *ACM Transaction on Information Systems*, Vol.22, No.1, 2004.
- [5] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, "An Algorithmic Framework for Performing Collaborative Filtering," *Proceedings of ACM SIGIR '99*, 1999.
- [6] T. Hoffman, "Latent Semantic Models for Collaborative Filtering," *ACM Transaction on Information Systems*, Vol. 22, No. 1, 2004
- [7] J. Konstan, B. Millr, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl, "GroupLens: Applying Collaborative Filtering to Usenet News," *Communications of the ACM*, Vol.40, No.3, 1997.
- [8] A. Popescul, L. Ungar, D. Pennock, and S. Lawrence, "Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments," *Proceedings of 17th Conference on Uncertainty in Artificial Intelligence*, 2001.
- [9] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of Dimensionality Reduction in Recommender System--A Case Study," *Proceedings of the ACM WebKDD Workshop*, 2000.
- [10] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based Collaborative Filtering Recommendation Algorithms," *Proceedings of the 10th International WWW conference*, 2001.