사용자 경향에 기반한 동적 추천 기법 : 영화 추천 시스템을 중심으로

(Dynamic Recommender on User Taste Tendency Model : Focusing on Movie Recommender System)

이 수 정 † 이 형 동 †† 김 형 주 †††
(Soojung Lee) (Hyungdong Lee) (Hyoungjoo Kim)

요 약 대부분의 추천 시스템에서는 개인의 선호 정보를 바탕으로 한 내용-기반 추천 기법과 다른 사람들로부터의 추천을 기반으로 한 사회적 추천 기법을 사용한다. 이들 두 기법은 각각 장단점을 갖고 있으며, 서로 경쟁 관계에 있다기보다 상호 보완적인 성격을 갖고 있다. 이에 두 기법의 적절한 조합이 전체 추천 시스템의 질을 결정하는 관건이 된다. 본 논문에서는 사용자 개인 마다 각 기법에 대한 만족도와 의존도가 다름을 밝히고, 이러한 각 개인의 경향에 따라 여러 추천 기법의 결과를 개인별로 조합해 주는 기법을 제안하였다. 각 개인의 경향을 나타내는 척도로 충성도, 다양도, 전문가도 등의 척도를 정의하여 사용하였으며, 이 원리에 의해 동작하는 조합 엔진의 결과는 최고 40%, 평균 23%의 coverage 개선 효과를 나타내었다.

키워드 : 정보 추천, 내용-기반 추천, 사회적 추천, 협업 기반 추천, 인구-통계 추천, 조합 추천 기법

Abstract Many recommender systems are based on Content-based Filtering and Social Filtering. Both methods have their own advantages and disadvantages, and they complement each other rather than compete. So incorporating of both methods can make the better system and combination technique controls the quality of the entire recommender system. In this paper, we presented each user has his own tendency to decide which is the better recommendation for himself among the various recommendation results, and suggested the personalized combination technique. To represent user tendency, we defined and used loyalty, diversity and pioneerity and showed by experiments that our combination technique is useful. This combination technique improved the average coverage 23% and for the ceiling 40%.

Key words : information filtering, content-based filtering, social filtering, collaborative filtering, demographic filtering, combination filtering

1. 서 론

정보의 범람 시대에 사는 현대인은 피곤하다. 너무나 많이 제공된 정보 중에서 자신에게 필요한 정보를 골라내는 것도 어려운 일이거니와, 열람한 정보가 자신에게 가장 알맞은 정보였는지 확신할 수도 없다. 정보 제공자 입장에서도 이러한 정보 과다의 현실은 같은 무게의 부담이 된다. 자신이 생산한 정보를 홍보할 효과적인 수단을 찾기가 힘든 것이다.

이러한 상황에서, 사용자 개인별로 사용자가 찾는 "바로 그" 정보를 찾아내고, 찾아낸 정보를 그 필요에 가장 부합하는 순서대로 순위화 해주는 개인화 추천 시스템은 훌륭한 해답이 될 수 있다. 실제로, 아마존(amazon, [1])을 비롯한 수많은 온라인 상점에서 고객 개인별로 상품을 추천해주는 추천 시스템이 사용되고 있으며, 고객과 판매자 모두에게 좋은 반응을 얻고 있다.

추천 시스템을 구성하기 위해서 크게 두 가지 기법이 많이 사용되고 있는데, 내용-기반 추천 방식(content-based filtering, CBF)과 협업 추천 방식(collaborative

· 본 연구는 정보통신부의 대학 IT연구센터(ITRC)와 BK-21 정보기술 사업단의 지원을 받아 수행되었습니다.
† 비 회 원 : 서울대학교 전기컴퓨터공학부
sjlee@oopsla.snu.ac.kr
†† 학생회원 : 서울대학교 전기컴퓨터공학부
hdlee@oopsla.snu.ac.kr
††† 종신회원 : 서울대학교 전기컴퓨터공학부 교수
hjk@oopsla.snu.ac.kr
논문접수 : 2003년 2월 18일
심사완료 : 2003년 10월 4일

filtering, CF)이 그것이다. CBF 시스템은 사용자의 선호 정보를 제공받아 그와 가장 유사한 아이템을 검색해 주는 기법이다. 반면에 CF 시스템은 추천할 대상 아이템에 대한 사용자의 평가를 입력받아 분석한 후, 비슷한 사용자끼리 묶어서 좋아할만한 아이템을 서로 추천하게 하는 방식으로 동작한다. 사회적인 관점에서 볼 때, CBF 기법은 사용자 개인의 내부에서 기인하는 추천이고, CF 기법은 사용자 외부의 다른 사람으로부터 오는 추천이라고 할 수도 있겠다. 자기 자신 내부의 목소리에 귀를 기울일 것인가, 외부의 다른 사람들 소리에 귀를 기울이고 따를 것인가의 문제는 많은 소소한 일 처리의 판단 기준이 되는 동시에, 각 개인에게 있어 가치관의 중요한 특성이기도 하다. CF와 같이 사용자 외부의 다른 사람으로부터 오는 추천 기법을 통틀어 사회적 추천 기법(Social filtering)이라고 부른다. CF 이외의 사회적 추천 기법에는 나이, 성별, 전문가 그룹 등의 인구-통계치를 이용하는 Demographic 추천이 있다[2,3].

정보 검색(Information Retrieval, IR) 기술에 기반을 두고 있는 CBF 기법은 공학적인 관점에서 볼 때 매우 상식적인 접근이라 할 만하나, 몇 가지 큰 단점을 갖고 있다. 주요하게는 사용자를 자신이 입력한 좁은 프로필 안에 가두게 될 염려가 있으며[3], 기법 자체가 추천 아이템의 도메인(그것이 책이냐 영화냐 음반이냐 등의)에 지나치게 종속적인 점이 문제이다. 또한 사용자는 되도록이면 적게 입력하고 양질의 추천을 얻으려는 성향이 있어서, 사용자의 온전한 선호 정보 프로필을 얻는 것도 쉬운 일은 아니다.

CF 기법은 다음과 같은 점에서 성공적이라 평가되어 왔다. 첫째, CF 기법은 대상 아이템이 상세한 설명 정보를 갖고 있지 않거나, 컴퓨터가 프로세싱하기 어려운 분야의 것이라 해도 문제없이 잘 동작한다. 사용자로부터의 상세한 선호 정보가 없어도 마찬가지로 잘 동작한다. 또한, 사용자의 선호 정보와 아이템 사이의 유사성을 계산하기 위한 복잡한 과정을 전혀 수행하지 않고도 양질의 추천을 제공해 준다는 것도 상당히 매력적이다. 이러한 이유로, CF 기법은 여러 추천 시스템에서 단독으로도 빈번히 이용되어 왔으며, 또 많은 경우에 있어서 충분히 훌륭한 성과를 거두기도 했다[4]. 그러나 CF 기법 역시 만만치 않은 문제점을 안고 있다. 첫 번째 문제는 CF 기법이 제대로 동작하려면, 대상 아이템에 대한 각 사용자의 평가가 어느 정도 풍부하게 축적되어 있어야 한다는데 있다. 대부분의 사용자는 아이템에 대해 평가를 입력하는데 대단히 인색하며, 설령 즐겨한다고 하더라도 시스템이 필요로 하는 만큼의 입력은 상당히 부담스러운 분량이어서, 사용자 개인이 감당하기가 매우 힘들다. 둘째 문제점은 시스템에 새로운 아이템이 추가되었을 때, 그 아이템은 사용자가 그에 대한 평가를 입력하기 전까지 어느 누구에게도 추천될 가능성이 원천 봉쇄되어 있다는 점이다. 시스템에 새로운 사용자가 추가되었을 때도 이와 유사한 문제가 발생한다. 또, CF 기법으로는 별로 만족스러운 추천 결과를 제공받을 수 없다고 호소하는 사용자들이 상당수 존재한다는 점도 아쉬운 부분이다. 이는 다른 사용자들과 어떤 뚜렷한 유사 경향이 나타나지 않는 사용자 층이 존재하기 때문에 생기는 문제점이다.

Demographic 기법은 마케팅 자료 따위로는 많이 이용되어 왔으나, 공학적인 추천 기법으로서는 사용하기가 애매한 점이 있어 사실상 거의 외면되어 온 것이 사실이다. Demographic 기법에 기반한 추천에는 같은 연령대로부터 온 추천, 같은 성별로부터 온 추천, 전문가 그룹에서 온 추천, 영화의 박스 오피스와 같은 불특정 다수로부터의 추천 등이 있을 수 있다[2].

위에서 살펴본 바와 같이 내용 기반 추천 기법과 사회적 추천 기법은 서로 상이한 관점에서 나온 기법이며, 서로 각기 다른 장단점을 갖고 있다. 사회적 기법이 여러 도메인에서 훌륭한 성과를 거두어 온 것이 사실이기는 하지만, 이들은 씨실과 날실이 그러하듯 서로 다른 각도에서 추천 결과를 제공하고 있으며, 우열 관계가 아닌 상호 보완적인 관계에 있다고 보는 것이 타당하다.

본 논문은 자신의 선호 정보에 기반한 내용-기반 추천 방식과 다른 사람들로부터의 사회적 추천 방식에 대한 만족도와 의존도가 개인별로 각각 다름에 착안하여, 사용자 개인별로 내용-기반 추천과 사회적 추천을 적절하게 조합하여 제공하는 시스템을 제안하였다.

2. 관련 연구(related work)

CBF 기법과 CF 기법을 결합하려는 시도는 전부터 있었으며, 상당 부분 연구가 진척되어 있다. 이러한 결합 시스템은 그 성격상 크게 조합 시스템(combination system)과 혼성 시스템(hybrid system)으로 나누어진다.

2.1 조합 시스템 (combination system)

조합 기법 연구란, 각각 다른 기법의 결과가 일단 독립적으로 산출된 뒤, 특정한 기준에 의해 각 결과가 다시 순위화 되는 기법을 의미한다. 간단한 초기의 시도로서, CBF 추천 결과 리스트와 CF 추천 결과 리스트를 뽑은 뒤 이 두 가지 리스트를 교대로 번갈아가며 섞은 연구가 있었다. 이 시스템이 낸 조합 결과는 각각의 결과보다 훨씬 나아진 정확도를 보였다[5]. 여기서 더 나아간 조합 연구로서, 두 기법의 결과 리스트의 평균을 구하여 평균값을 기준으로 가중치를 정해 두 결과를 섞는 연구도 있었다. 공학적으로, CBF의 결과 리스트와 CF의 결과 리스트는 이질적인 것이었고, 이러한 두 리

스트를 섞는 것에 대한 기준을 정하기가 애매하였기 때문에 순수한 의미의 조합 연구는 이것 이외에 한동안 눈에 띄는 연구가 이루어지지 않았다. 2000년에 이르러 보다 진보된 조합 시스템이 등장하였으며, 논문 [6]에서 저자는 사용자마다 그리고 아이템마다 각 기법에 대한 의존도가 다름을 발견하였다. 이것에 착안, 저자는 사용자와 아이템들 각 쌍(pair)에 각 기법에 대한 가중치를 부여하여 이 가중치에 따라 두 기법을 조합하는 시스템을 제안하였다. 이것은 본 논문에서 제안하는 시스템과 비슷한 궤도에 있는 첫 작업이라 할 수 있다.

2.2 혼성 시스템(hybrid system)

독립적인 각 기법으로부터 일단 결과를 얻어낸 후 얻은 그 결과를 조합하는 과정을 거치는 조합 시스템(combination system)과 달리, 혼성 시스템(Hybrid system)은 추천 단계에서부터 각 기법을 혼합하여 실행시키는 시스템이다. 이 시스템은 기본적으로 CF기법에 기반을 두고 있으며, 다른 기법들은 CF 기법의 단점을 보완하기 위한 목적으로 사용되었다.

혼성 시스템의 전통적인 대표 시스템으로 스탠포드에서 나온 Fab.(1997)를 들 수 있다[7]. 이 시스템은 웹 페이지를 추천해주는 시스템으로서, 내용-기반 추천을 제공하기 위해 사용자로부터 프로필을 받아 저장하는 동시에, 사용자 피드백을 받아 프로필을 갱신하였다. 또한 유사한 사용자들을 묶어내기 위해서 프로필을 분석하였다. 그리하여, 사용자가 적고 찾으려는 토픽이 많은 경우에는 주로 내용-기반 추천 기법이 사용되도록 하고, 반대로 사용자가 많고 찾으려는 토픽이 적을 때에는 주로 협업-기반 추천 기법이 동작하도록 되어 있다.

GroupLens는 필터봇(filterbot)이라는 에이전트를 두어, 사용자의 내용-기반의 프로필을 반영하게 하였다. 즉, 필터봇은 사용자의 선호 정보대로 행동하는 시스템의 또다른 사용자인 것처럼 취급되며, 이런 기작으로 CF 엔진을 돌려 GroupLens의 최종 추천 시스템이 더욱 잘 동작하게 할 수 있었다. 이러한 측면에서, GroupLens 역시 혼성 시스템의 일종이라 할 수도 있다[8].

[2]에서 Pazzani는 드물게 Demographic 추천 기법(DF)의 결과에 주목하였고 CF와 CBF, DF, CF+CBF의 결합 기법을 실험을 통하여 비교하였다. 그는 CBF, DF에서 얻은 정보를 사용자의 content-profile 행렬로 만들어서 CF 추천 기법을 동작시킬 때 입력으로 사용하는 방법을 제안하였다.

2.3 조합 시스템과 혼성 시스템의 비교

앞서 지적했듯이, 조합 기법은 공학적으로 그 조합 기준을 정하기가 어려운 면이 있다. 때문에 많은 연구자들은 혼성 시스템의 연구에 더 많은 관심을 보여 왔으며, 실제로 이들 시스템은 많은 도메인에서 훌륭한 성능을 보여 주었다. 그러나 혼성 시스템은 CF 기법이 잘 동작하지 않는 범위에서 CBF 기법을 이용하여 빈 공간을 메운 것일 뿐, 본질적으로 CF 기법과 크게 다르지 않으며 CF가 갖고 있는 문제점을 상당히 안고 있다. 또한, 사용자 정보가 충분하지 않아서 CF 엔진을 제대로 돌릴 수 없는 경우에는 섬세하게 설계된 순수 조합 기법 시스템보다 성능이 떨어진다. 많은 웹 사이트들에서 대량의 자체적인 사용자 정보를 보유하지 못하고 다른 곳에서 사용자 정보를 수입해야 하는 현실을 비추어 볼 때, 혼성 시스템 연구 외에, 이미 존재하는 여러 결과 리스트를 결합시켜주는 조합 기법에 대한 보다 깊은 연구가 절실히 필요하다. 본 논문에서는 이러한 측면에서, 순수 조합 기법 시스템을 제안하고 있다.

3. 사례(motivating example)

이 절에서는 본 논문의 동기가 된 사례를 들어, CBF 추천 결과와 CF 추천 결과가 결합되어야 하며, 이들 리스트들을 조합할 때, 개인별로 CF 추천 결과와 CBF 추천 결과의 조합 비율이 달라야 한다는 것을 보이고자 한다. 이를 위해 도메인으로서 영화를 택해서 이야기하기로 하겠다.

여기 두 명의 고등학생이 있다. 한 명은 '갑'이라는 여고 2년생이며, 한 명은 고등학교 2학년 남학생으로, 각각의 프로필은 표 1에 주어져 있다.

이들은 오는 주말에 영화 구경을 가려하고 있으며, 주말에 관람할 수 있는 영화 중에서 추천된 영화의 목록은 표 2와 같다.

표 1 두 명의 학생에 대한 프로필

이름	갑	을
나이	s여고 2학년	s고 2학년
좋아하는 영화 배우	소피 마르소	소피 마르소
좋아하는 영화 감독	알프레도 히치콕	알프레도 히치콕

표 2 상영 중인 영화 목록

	영화 이름	비고
1	현기증	알프레도 히치콕 감독의 영화
2	구름 저편에	소피 마르소의 영화
3	센과 치히로의 모험	'갑'과 비슷한 학생 그룹이 가장 좋다고 평가한 영화
4	다이 하드	'을'과 비슷한 학생 그룹이 가장 좋다고 평가한 영화
5	반지의 제왕	요즘 가장 많은 관객을 동원하고 있는 영화
6	시민 케인	요즘 평론가 그룹에서 가장 좋은 평가를 받은 영화
7	화산고	요즘 고등학교 2학년 학생에게 가장 인기 좋은 영화

표 3 '갑'과 '을'이 선호한 영화

순위	'갑'이 선호한 영화	'을'이 선호한 영화
1	헌기증 (CBF 결과)	화산고 (Demographic 결과)
2	구름 저편에 (CBF 결과)	다이 하드 (CF 결과)
3	센과 치히로의 모험 (CF 결과)	반지의 제왕 (Demographic 결과)
4	시민 케인 (Demographic 결과)	구름 저편에 (CBF 결과)
5	반지의 제왕 (Demographic 결과)	센과 치히로의 모험
6	다이 하드	헌기증 (CBF 결과)
7	화산고 (Demographic 결과)	시민 케인 (Demographic 결과)

표 2에서 1,2는 '갑'과 '을'의 프로필의 선호 감독-배우 정보 때문에 추천된 CBF 추천의 결과이다. 3,4는 CF 추천의 결과이며, 5,6,7은 Demographic 추천 기법의 결과이다.

이 때, '갑'과 '을'의 실제 영화 선호도는 표 3과 같았다.

표 3을 보면 '갑'과 '을'이 입력한 선호 정보가 같았음에도 불구하고 영화를 선택하는 경향이 다르다는 것을 알 수 있다. 대체로 '갑'은 자신의 선호 정보에 충실한 경향을 보였고, '을'은 자신이 입력한 선호 정보에 따라 영화를 관람하는지의 여부가 뚜렷하지 않음을 알 수 있다. 이렇듯, 개인 별로 자신의 프로필에 기인한 CBF의 추천 결과를 취하는지, CF나 Demographic 과 같은 사회적 추천 결과를 취하는지를 결정하는 것은 사용자의 개인적 경향에 달려있음을 관찰할 수 있다. 우리는 관찰된 이 개인적 경향을 앞으로 사용자 경향(user tendency)이라 부르기로 한다.

4. 사용자 경향(user tendency)과 취향 공간 (taste space)

4.1 배경(background)

영화나 음악, 도서 등 문화적 현상에 대한 각 개인의 취향이 분포한 공간을 취향 공간(taste space)이라 부른다[9]. 비슷한 취향을 공유한 사람들의 경우 취향 공간의 특정한 부분을 공유하게 된다. 마이크로소프트사(社)의 파이어플라이(FilreFly)나 아마존(Amazon)의 추천 엔진은 대표적인 상용 CF 엔진으로서, 사용자들의 취향 공간에 기반한 관계 기술(Relational Technology : R- Tech)을 적용한 대표적 엔진들이다[9]. 취향 공간이라는 용어나 이에 대한 연구는 전통적으로 심리학이나 경영학의 마케팅 분야에 속해 있었으나, 인터넷의 보급과 함께 찾아온 온라인 커뮤니티 시대의 도래에 발맞추어 유럽 등지에서 취향 공간을 도입한 연구가 공학적으로 또한 나타나고 있다[9-12]. 취향 공간은 여러 인자로 분할-분석되어 수학적인 모델로 정립된 바는 없으나, 대개 (사용자, 선호 아이템)의 순서쌍으로 구성되며 '관계 연구(relationship technique)'에 시각적인 직관을 제공한다.

데이타마이닝에서 충성도(loyalty)와 다양도(diversity)라는 용어가 존재한다. 마이닝에서 충성도란 '어느 고객이 특정 사이트에 얼마나 고정적으로 들르는가' 하는 척도를 나타내며, 다양도는 '어느 고객이 선택하는 상품의 범위'를 나타내는 척도로 쓰인다. 마이닝을 이용한 통상적인 CRM의 목적은 고객의 충성도를 굳건히 하고, 고객의 상품 선택 다양도를 높임으로서 매출을 높이는 데 있다고 말할 수 있다[13]. 본 논문에서 정의할 '충성도(loyalty)'와 '다양도(diversity)'는 이들 개념과는 다르지만 의미상으로 유사성이 있음을 밝혀둔다.

4.2 사용자 경향과 취향 공간

3절에서 여러 가지 추천 기법이 제공한 추천 결과 중에 특정 추천 결과를 선호하는 개인별 경향을 사용자 경향이라 한다고 하였다. '갑'의 영화 관람은 자신의 선호 정보에 비교적 충실한 경향을 갖고 있으며, '을'의 경우 자신이 명시한 선호 정보에 비교적 충실하지 않은 경향을 갖고 있다고 말할 수 있을 것이다. 다른 관점에서, '갑'의 경우 외부에서 다른 사람들에 의해 제공되는 추천에 대해 수용성이 낮은 경향을 갖고 있으며 '을'의 경우는 높은 수용성을 갖고 있다고 말할 수도 있다. 또, '갑'보다 '을'의 취향이 보다 다른 사람들과 일치되어 있다고 말할 수 있으며, '갑'은 '을'에 비해 독특한 취향을 가졌다고 말할 수도 있다. 또한 '갑'은 '을'보다 평론가들이 제공한 추천에 긍정적인 태도를 보이고 있음도 관찰할 수 있다.

앞에서 기술한 여러 가지 경향들을 다음의 세 가지 인자를 정의하여 기술할 것을 제안한다.

A. 충성도(loyalty) : 사용자가 자신의 선호 취향에 얼마나 충실한지 나타내는 척도

B. 다양도(diversity) : 사용자의 선호 취향이 얼마나 다양한지 나타내는 척도. 다른 사람들로부터의 다양한 추천을 받아들이는 정도를 나타내기도 한다.

C. 전문가도(pioneerity) : 사용자가 자신의 선호 취향에 대해 얼마나 탐험적인 태도를 갖고 있는지를 나타내는 척도. 전문가적이냐 아마추어적이냐 하는 정도를

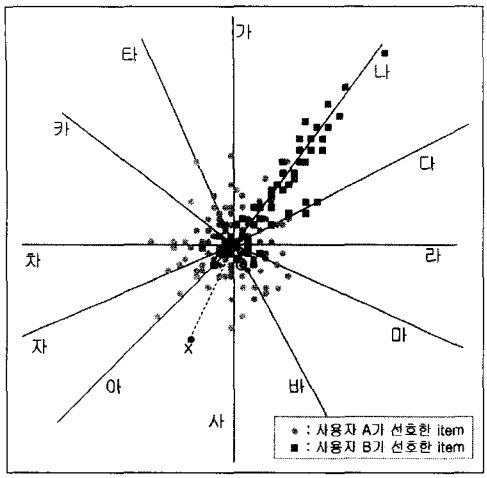


그림 1-1

나타낸다.

앞에서 기술한 내용을 취향 공간(taste space) 상에서 관찰하면 사용자 경향을 파악하기가 쉽다. 다음을 보자.

그림 1-1은 2차원 취향 공간을 나타내며(흔히 취향 공간이라 불리는 좌표계는 이 2차원 취향 평면을 이르는 말이다.) 원점에서 뻗어나간 '가' ~ '타'까지의 각 축은 유사한 특정 성질을 가진 아이템의 그룹을 나타낸다. A의 경우, 어느 쪽에도 편향되지 않은 고르고 다양한 취향을 가진 사용자를 나타낸다. 이러한 사용자는 그 선호 취향이 자못 다양하며 외부에서 추천되는 추천에 대해 수용성이 높은 특성을 나타낸다. 반면 B의 경우, 주로 '나' 그룹의 아이템만을 선호하는 경향이 있는 매우 편향된 취향의 사용자임을 나타낸다. 즉, A는 다양도가 높은 사용자를, B는 다양도가 낮은 사용자를 나타낸다.

그림 1-1에서 원점에서 떨어진 거리를 나타내는 선분 OX의 의미에 주목하자. 원점 O에 가까울수록 보다 많은 사람에게 알려져 있는 아이템임을 나타낸다. OX가 클수록 X는 사람들에게 많이 알려지지 않은 아이템이며, 원점에서 멀리 떨어진 X를 알고 즐기는 사람은 높은 전문가도를 가진 사람이라고 판단할 수 있다. 이 경우, B는 '나' 그룹에 대해서만은 전문가적 면모가 짙다고 할 수도 있겠다.

그림 1-2, 그림 1-3은 그림 1-1의 2차원 공간에 z축을 더한 3차원 공간이다. 이 때 x-y 평면은 그림 1-1의 2차원 취향 평면을 나타내며, z축은 시간 혹은 아이템 소비 횟수를 나타낸다. 즉, z축까지 합쳐진 이 3차원 공간의 점 분포도를 관찰함으로서, 사용자의 취향과, 시간에 따른 실재 아이템 소비 패턴을 알 수 있다.(혹은 다음 번 조사부터의 취향 혹은 다음 번 조사했을 때부터의 실재 소비 패턴이라고 해석해도 무리는 없다.)

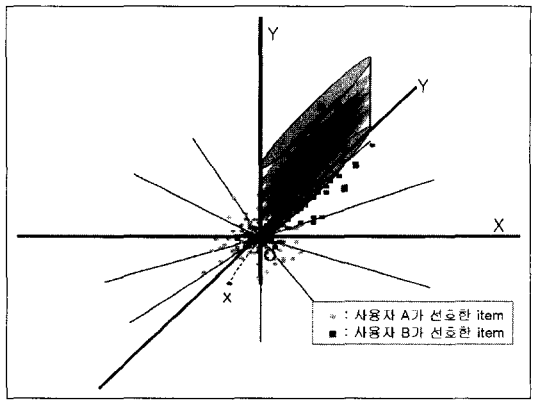


그림 1-2

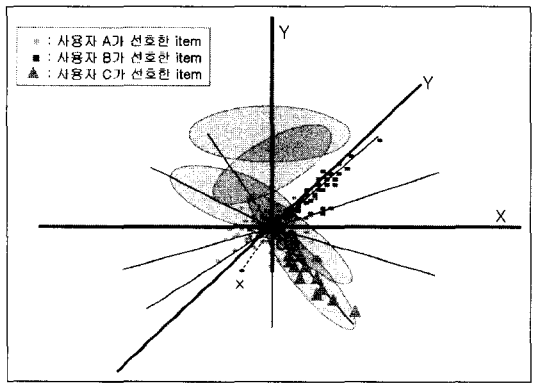


그림 1-3

그림 1-2의 B는 실제 소비하는 아이템이 자신의 프로필과 잘 일치하는 사용자를 나타낸다. 즉, 이 사용자는 자신의 프로필에 충실한 소비 패턴을 보이고 있으며, 이는 이 사용자의 프로필에 대한 충성도가 높음을 나타낸다. 반면, 그림 1-3의 C는 실제 소비하는 아이템이 자신의 프로필과 잘 일치하지 않는다. 이 사용자의 경우 아이템의 소비행태를 프로필만 가지고서는 점치기가 힘들며, 자신의 선호 취향 프로필에 대한 충성도가 낮다고 할 수 있다.

앞서 정의한 3차원 공간에 극좌표와 원기둥 좌표를 적용하여 다소 정량적인 접근을 시도할 수도 있다. 원점으로부터의 거리와 x축으로부터 측정한 회전 각도를 이용하여 평면을 기술하는 좌표를 평면 극좌표라고 한다. 그림 2-1은 임의의 한 개인 '병'의 취향을 평면에 나타낸 것이다. 그림에서 어둡게 나타낸 부분이 '병'의 선호 취향을 나타낸다. 앞서의 설명에 비추어, 그림 2-1의 평면의 각 θ들의 합이 클수록 다양도가 높은 사용자이며, 어둡게 나타낸 취향정보들이 원점으로부터 먼 거리까지

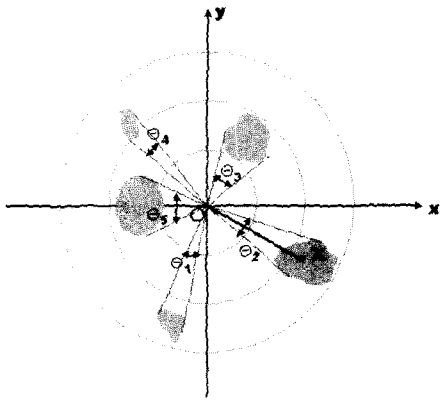


그림 2-1

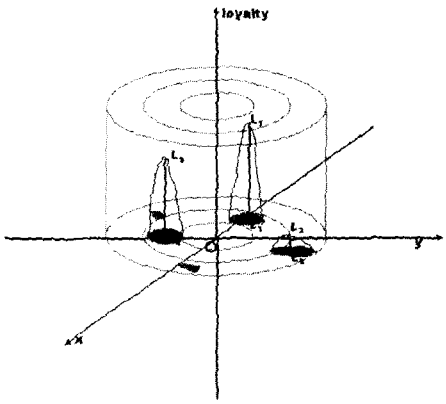


그림 2-2

진출해있는 취향이 보다 높은 전문가도를 가진다고 말할 수 있다. 즉, 다음과 같이 기술할 수 있다.

$$\text{Diversity} = \sum \Theta n$$

$$\text{Pioneerity} = \oint \text{OX} \,/\, \text{S}$$

그림 2-2는 그림 2-1의 평면에 한 차원을 더 하여 만든 3차원 공간 좌표이다. 위에서 설명한 맥락에서 볼 때, 추가된 z축은 충성도를 나타낸다. 즉,

$$\text{loyalty} = \overline{LL'}$$

그림 2-3은 원기둥 좌표계를 나타낸다. 원기둥 좌표계는 평면 극좌표에 높이를 나타내는 z축을 더해서 만든 3차원 좌표계로서 $P(\theta, r, z)$ 가 공간 내의 모든 점을 기술한다. 또한 θ, r, z는 상호 간에 직교 성질을 갖는다.

위에서 설명한 바와 같이, 우리가 정의한 취향 공간 상의 다양도, 전문가도, 충성도는 각각 3차원 원기둥 좌표의 θ, r, z에 해당한다. 여기에서, 이들 3가지 척도가 3차원 취향 공간 상에서 의미적으로 직교(orthogonal)하며 완전(complete)한 성질을 가짐을 직관적으로 시사받을 수 있다.

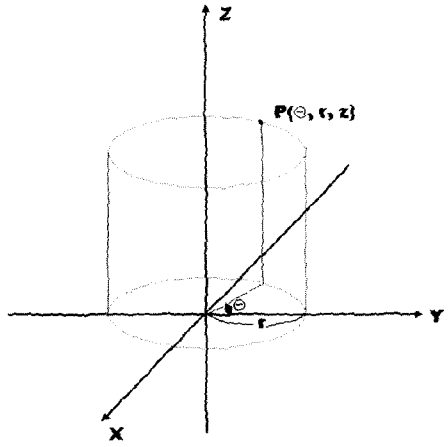


그림 2-3

5. 시스템 구조

본 논문이 제안할 조합 추천 기법의 효용성을 보이기 위하여, 영화 추천의 도메인 위에서 시스템을 구축하였다. 본 논문이 주장하는 조합 기법은 원칙적으로 도메인에 제약을 받지 않으나, 현실적으로 가장 쉽게 양질의 자료 및 사용자 정보를 구할 수 있는 영화 도메인에서 동작하는 시스템을 구축하였다.

본 논문의 시스템을 상위에서 조망한 그림이 그림 3에 있다. 본 시스템 Moviecon은 그림과 같이 Analyzer 모듈, CBF 모듈, CF 모듈, Demographic 모듈, Fusion 모듈로 구성되어 있다. 5.1절에서 5.5절에 걸쳐 본 시스템의 각 부분을 설명한다.

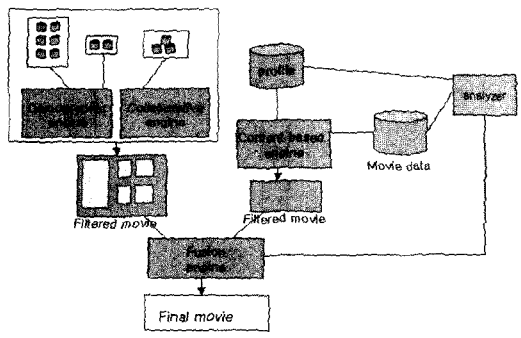


그림 3 시스템 구조

5.1 Analyzer 모듈

Analyzer 모듈은 입력된 각 사용자의 정보를 분석하여 각 사용자의 사용자 경향(user tendency)을 구하여 데이타베이스의 user_tendency 테이블에 저장한다. 여기서 구해진 사용자 경향은 뒤에서 Fusion 엔진이 각

추천 결과를 조합하는 비율을 개인별로 정하는 역할을 한다. Analyzer는 세 가지 부분으로 이루어져 있다.

- loyalty analyzer : 사용자의 선호 정보와 사용자가 평가한 영화 정보를 이용하여 사용자의 loyalty를 구한다. 이 loyalty 값은 사용자가 자신이 선호한다고 입력한 선호 정보 프로필에 얼마나 충실한가를 나타내는 값으로서, 사용자의 프로필과 사용자가 실제로 좋다고 평가한 영화 사이의 구성요소 유사성을 측정하여 구한다.

loyalty = Σ similarity(사용자 선호 정보(배우, 감독, 장르), 사용자가 좋다고 평가한 영화 집합)

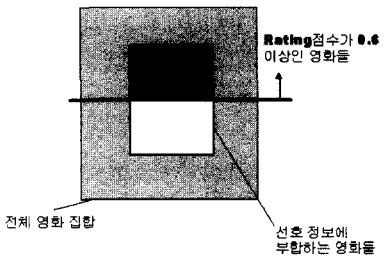


그림 4

그림으로 충성도의 의미를 살피면 그림 4와 같다. 본 시스템에서 영화에 대한 평가는 (0.0, 0.2, 0.4, 0.6, 0.8, 1.0)으로 입력받았다. 이 때, 0.5 이상의 점수는 호(好)감정을 0.5 이하의 점수는 오(惡)감정을 나타낸다고 전제했었다. 그러므로 사용자 i의 충성도($loyalty_i$)값은 다음과 같이 구해진다.

$$loyalty_i = \frac{\blacksquare}{\blacksquare} = \frac{\sum_{r\geq 0.6} r_{i,j} \cdot p_{i,j}}{\sum_j p_{i,j}}$$

r_{ij}는 사용자 i가 영화 j에 준 평가 점수($0.0 \leq r_{i,j} \leq 1.0$),

p_{ij}는 사용자 i가 영화 j의 구성 요소(감독, 배우, 장르)에 준 전체 점수

감독 충성도($loyalty_{dirct}$), 배우 충성도($loyalty_{actor}$), 장르 충성도($loyalty_{genre}$)는 각각 다음과 같다.

$$loyalty_{dirct} = \frac{\sum_{r\geq 0.6} r_{i,j} \cdot p_{i,j(director)}}{\sum_j p_{i,j}}$$ ($p_{ij(director)}$는 사용자 i가 영화 j의 감독에 준 점수)

$$loyalty_{actor} = \frac{\sum_{r\geq 0.6} r_{i,j} \cdot p_{i,j(actor)}}{\sum_j p_{i,j}}$$ ($p_{ij(actor)}$는 사용자 i가 영화 j의 배우들에 준 점수의 합)

$$loyalty_{genre} = \frac{\sum_{r\geq 0.6} r_{i,j} \cdot p_{i,j(\geq nre)}}{\sum_j p_{i,j}}$$ ($p_{ij(genre)}$는 사용자 i가 영화 j의 장르들에 준 점수의 합)[1]

어떤 사용자의 감독 충성도 $loyalty_{dirct}$ 값이 크면 이 사용자는 자신이 입력한 선호 감독의 영화라면 충실히 관람했다는 뜻이 된다. 즉, 감독이 영화를 고를 때 중요한 비중을 차지했다는 의미이고, 이는 바꾸어 말하면 감독을 보고 영화를 고르는 경향이 있다는 뜻도 된다. 그러므로 $loyalty_{dirct}$이 큰 사용자에 대해서는 감독 CBF의 결과를 중요하게 반영해 주어야 할 것이다. 배우 충성도나 감독 충성도에 대해서도 마찬가지 논리가 성립된다.

- **diversity analyzer** : 다양도는 사용자의 선호 취향이 얼마나 다양한지를 나타내는 척도이자 외부에 대한 수용성의 척도가 되기도 한다. 사용자 i의 다양도(diversityi)는 다음과 같이 정의된다.

$$diversity_{dirct} = \frac{\text{선호하는 } director\text{수} - \text{싫어하는 } director\text{수}}{\text{전체 } director\text{수}}$$

$$diversity_{actor} = \frac{\text{선호하는 } actor\text{수} - \text{싫어하는 } actor\text{수}}{\text{전체 } actor\text{수}}$$

$$diversity_{genre} = \frac{\text{선호하는} \geq nre\text{수} - \text{싫어하는} \geq nre\text{수}}{\text{전체} \geq nre\text{수}}$$

이 때 사용자 i의 전체 다양도($diversity_i$)는 세 다양도의 가중 평균으로 구해진다.

장르 다양도가 큰 사용자란 영화를 감상할 때 여러 장르의 영화를 다양하게 보는 사람인 동시에, 장르에 대한 수용성이 큰 사람을 가리킨다. 감독 다양도, 배우 다양도의 경우도 마찬가지이다. 그러나 감독이나 배우에 대해서는 그룹으로 분류해 놓은 dataset이 현재 없으므로 감독 다양도나 배우 다양도는 현재 시스템에서 적용되기 어렵다. 그러나 배우나 감독을 그룹화 해 놓은 자료가 있다면 추천을 위한 자료로 유용하게 쓰일 것이다.

- **pioneerity analyzer** : 전문가도는 사용자가 자신의 선호 취향에 대해서 얼마나 전문가적 태도를 견지하고 있는가에 대한 척도이다. 만약 어떤 사용자가 감독 충성도가 높다고 하여도, 그가 거의 모든 사람들이 다 아는 감독에 대해서만 알고 있는 것이었다면 이 사용자는 전문가도가 높다고 할 수 없다. 전문가도를 구하기 위해 먼저 awareness계수를 먼저 구해야 한다. 어떤 집단에서 awareness계수란 집단의 사용자에게 얼마나 알려져 있는지를 나타내는 척도이다. 즉 어떤 아이템 j에 대한 집단 S의 awareness 계수 a는 다음과 같이 구해진다.

$$a_j = \frac{\text{집단 } S\text{에서 } j\text{를 아는 구성원의 수}}{\text{집단 } S\text{의 전체 구성원 수}}$$

본 시스템에서는 전체 집단에 대해 이러한 조사를 하기 어려운 관계로 웹의 검색 엔진에서 검색되는 문서의 수로 awareness 계수 값을 대신하였다.

어떤 사용자 i가 n명의 감독을 선호한다면 이 사람의

1) 각 영화는 2개 이상의 장르에 속할 수 있다.

감독 전문가도는 다음과 같이 주어진다.

$pioneeritiy_{i(director)}$ = ($\sum$n명의 감독 중 awareness 계수가 가장 작은 하위 50% 감독들의 awareness)$^{-1}$

배우 전문가도도 마찬가지로 구하면 된다. 전체 전문가도는 이 두 가지 값의 가중평균값이다.[2)]

전문가도가 높은 사람은 대개 평론가의 추천 리스트를 따르는 경향이 있다는 가정을 세워서 시스템을 구성했다.

Analyzer는 구한 값을 데이타베이스의 user_tendency 테이블에 사용자id와 함께 저장한다. 이는 후에 Fusion엔진의 입력으로 쓰인다.

5.2 순수 CBF 엔진

CBF 엔진의 역할은 사용자가 입력한 선호 정보와 영화 정보를 비교하여 사용자 프로필에 기반하여 사용자가 좋아하리라 예측되는 영화를 추출하는 것이다. 본 시스템에서 사용자에게 배우, 감독, 장르의 리스트와 함께 라디오 버튼 박스를 제공하여 선호 정보로서 좋아하는 장르와 좋아하는 배우, 좋아하는 감독을 입력받았으며, 싫어하는 의사표시는 (-) 점수를 주어 특별히 표시하게 했다.

표 4 CBF 엔진을 위한 사용자 입력 정보

	set의 크기	rating scale
감독 선호 정보	총 166명의 감독	-2 : 특별히 싫어한다. 0 : 별 감정이 없다. +1 : 좋아한다. +2 : 매우 좋아한다.
배우 선호 정보	총 260명의 배우	
장르 선호 정보	총 21개의 장르	

CBF 엔진은 사용자의 선호 정보와 영화 정보를 입력으로 받아 각 영화의 CBF_score를 계산하여 반환해준다. 사용자 i가 영화 j에 대해서 받게 될 CBF_scorei,j는 다음과 같은 식으로 계산된다.

$CBF\text{-}score(감독)_{i,j}$ = $\sum_{for\ all\ director\ of\ movie\ j}$ (사용자 i가 영화 j의 감독에 준 선호 점수)

$CBF\text{-}score(배우)_{i,j}$ = $\sum_{for\ all\ actor\ \in\ movie\ j}$ (사용자 i가 영화 j에 출연하는 배우에게 준 선호 점수)

$CBF\text{-}score(장르)_{i,j}$ = $\sum_{for\ all\ genre\ which\ movie\ j\ belongs\ to}$ (사용자 i가 영화 j가 속하는 장르에 준 선호 점수)

$CBF\text{-}scorei,j$ = $CBF\text{-}score(감독)_{i,j}$ + $CBF\text{-}score(배우)_{i,j}$ + $CBF\text{-}score(장르)_{i,j}$

2) 모든 장르는 서로 동등하다고 보는 것이 자연스럽다. 그러므로 장르 전문가도는 정의하지 않았다.

5.3 순수 CF 엔진

CF 엔진은 사용자가 영화에 대해 평가한 평가 점수의 집합 위에서 동작한다. 이 사용자들이 평가 점수를 분석함으로서, CF 엔진은 취향이 비슷한 사용자들을 서로 묶을 수 있고, 비슷한 취향을 가진 사람들끼리 서로 좋아하는 영화를 추천하게 함으로써 추천 엔진이 동작한다.

본 시스템에서는 각 사용자 사이의 상관 계수 분석을 통하여 CF 추천이 이루어지는 알고리즘을 택하였다. 이는 가장 단순한 방법이나 효율적이지는 않다.

R={$r_{i,j}$}와 같은 행렬 R을 생각하자. $r_{i,j}$는 사용자 i가 영화 j에 대해서 평가한 평가 점수이다. 사용자 c가 있을 때, 이 행렬에 피어슨 상관계수 k_{ci}(사용자 c와 사용자 i 사이의 피어슨 상관 계수)를 적용함으로써 사용자 c와 비슷한 취향의 사용자들을 찾아낸다.

$$k_{ci}=\frac{\sum_{j\in I_{ci}}(r_{cj}-\overline{r_c})(r_{ij}-\overline{r_i})}{\sqrt{\sum_{j\in I_{ci}}(r_{cj}-\overline{r_c})^2}\sqrt{\sum_{j\in I_{ci}}(r_{ij}-\overline{r_i})^2}}$$

Ici은 두 사용자 c와 i 둘 다에 의해 평가된 영화들의 집합이며, $\overline{r_i}$와 $\overline{r_c}$ 는 각각 사용자 i와 c가 평가한 평가점수의 평균이다. 이 때 양의 상관 계수 뿐만 아니라 음의 상관계수 또한 고려된 식임을 주목하라. CF 모듈에 대한 예측치를 계산하는 식은 다음과 같다.

$$\widehat{r_{cj}^{CF}}=+\frac{\sum_{i\in U_{cj}}(r_{ij}-\overline{r_i})k_{ci}}{\sum_{i\in U_{cj}}|k_{ci}|}$$

U_{cj}는 영화 j의 평가에 참여한 사용자 i의 집합이다. 이 때 $k_{ci}>T_2$ (T_2는 유사한 그룹에 속하는 사용자이기 위한 상관 계수의 최소 크기)인 경우만 고려하는데, 이 때 T_2를 경계값이라 한다. 본 시스템에서는 CF 모듈을 동작시킬 때, T_2의 값을 입력으로 주게 되어 있다.

마지막으로 $\widehat{r_{cj}^{CF}} \geq T_1$ (T_1은 좋아할 것이라 믿어지는 최소 예측 수치) 을 만족시키는 영화 j 만이 사용자 c에게 추천되는 원리이다.

5.4 Demographic 모듈

이 모듈은 시스템의 전체 사용자가 전체 영화에 대해 평가한 평가점수가 저장된 리스트를 입력으로 받아 일반 사용자에 의한 영화 순위 리스트, 평론가 순위 리스트, 장르별 순위 리스트를 반환한다. 몇 개의 sql 스크립트로 이루어져 있다.

5.5 Fusion 엔진

본 시스템의 Fusion 엔진은 두 단계에 걸쳐 동작한다. CF 엔진의 결과를 제일 존중하되, 첫 단계에서는 CBF와 CF의 결과를 조합하며 두 번째 단계에서는 첫 번째 Fusion 결과 리스트와 Demographic 추천 결과를

조합하게 되어 있다.

이것의 의미는 다음과 같다. 첫 번째 단계는 사용자마다 다른 프로필에 대한 충성도를 반영하는 단계로서, 개인적 선호 취향과 사회적 추천 사이의 개인별 균형점을 찾아주는데 의의가 있다. 두 번째 단계는 사용자마다 개인별로 사회적 추천을 받아들이는 양상이 다른 데에 따른 것이다. 똑같이 남의 말에 귀를 잘 기울이는 사람들임에도 불구하고, 어떤 사람들은 비평가 그룹의 견해에 더 공감하기도 하고, 어떤 이는 대중 그룹의 견해에 더 공감하기도 하는 것이 그 예이다. 이 단계는 여러 가지 사회적 추천 결과 사이에서 개인별로 균형점을 잡아주고자 하는데 목적이 있다. 또한, 사용자 본인이나 유사 그룹의 사람들이 평소 멀리 하는 아이템 중에서도 재미있게 받아들일 수 있는 아이템이 있을 것이다. 평소 액션 영화를 잘 보지 않는 사람이라도 특출나게 빼어난 완성도를 지닌 액션 영화에는 어느 정도 호의적인 반응을 갖게 마련이다. 두 번째 단계에서는 사용자의 다양도와 전문가도에 따라 사용자마다 적당한 범위에서 프로필이나 유사 집단 외의 아이템도 추천될 수 있도록 하였다.

5.5.1 1단계 Fusion : CBF와 CF의 결합

이 단계에서는 사용자의 충성도에 따라 CF에 CBF를 섞는 비율을 달리하여 개인별 1차 Fusion 리스트를 제공한다. 충성도가 높은 사용자의 경우 CBF의 결과가 중요하게 반영되며 그렇지 않은 경우에는 CBF의 반영 비율을 미미하게 한다.

두 리스트를 조합하는 식은 다음과 같다.

Fusion List 1

= rating (Kcf CF, K_{cbf} Loyalty CBF)

$$= CF\_score + (K_{cbf} \times Loyalty_{genre} \times CBF\_score_{genre} + K_{cbf} \times Loyalty_{director} \times CBF\_score_{director} + K_{cbf} \times Loyalty_{actor} \times CBF\_score_{actor})$$

예를 들어 감독 충성도가 높은 사용자의 경우 좋아하는 감독이 만든 영화가 Fusion 리스트의 상위에 존재하게 된다. K_{cbf}의 경우 초기 테스트를 통하여 가장 적절한 값을 구하였다. 본 시스템의 경우 K_{cbf}값은 0.5이다.

5.5.2 2단계 Fusion : 1차 Fusion 결과 리스트와 Demographic 추천 결과의 조합

1단계의 조합만으로는 사용자를 좁은 범위의 아이템에 갇히게 할 우려가 있고, 의외의 아이템이 주는 감동을 사용자에게 주기 힘들다. CF가 사회적 기법이라고는 하나 비슷하나 사람들로부터의 추천이기 때문에 경우에 따라, 보수적인 그룹인 경우 내용-기반 추천과 같이 소수의 아이템 속에 갇혀버리는 결과를 초래할 가능성이 있다. 이를 위해 2단계에서는 인구-통계적 사회적 추천을 적극 이용하였다. 사용자의 수용 가능 범위에서, 선호 정보 밖의 아이템들도 추천될 수 있는 통로를 만들어 둔 것이라고 할 수 있는 이 단계는, "많은 사람들이 특별히 좋아하는 아이템에는 반드시 주목할 만한 가치가 있다."라는 전제를 받아들인 것이기도 하다. 사용자의 충성도가 낮고, 다양도가 높을 경우에는 Demographic 결과(DF)가 보다 많이 반영되도록 하였다. 사용자의 전문가도가 높을 경우에는 평론가 집단의 추천 결과를 중요하게 반영하고 그 반대의 경우에는 일반 사용자 집단의 추천 결과를 주요하게 반영하는 단계이다.

Fusion List 2

= Fusion List1 + (DF의 결과 반영분)

시스템 기획 당시, 본 추천 시스템에서 반영할 DF 결과는 장르별 차트와 평론가-일반 사용자별 차트였으나, 다수의 평론가가 작성한 일반성을 갖춘 평론가 차트를 확보하지 못한 관계로, 장르별 차트만을 반영하게 되었다. 다음은 DF의 장르별 차트 결과 반영분을 구하는 식을 나타낸다.

$(1-loyalty)^2$ $diversity^2$ [1+0.1(1-diversity) $cbf\text{-}score_{genre}$] chart-score

일단 장르별 차트를 반영함에 있어서, 사용자의 선호하는 장르의 차트가 더 많이 반영되어야 하는 것은 자명하다. 이를 위해서, 사용자의 장르 선호도가 참조되어야 하며, 이를 위해 일차적으로 $cbf\text{-}score_{genre}$ 인자가 곱해졌고, 다양도가 떨어질수록 자신의 선호 장르를 고집하는 경향은 커지므로 (1-diversity)값을 다시 곱하여 0.1 가중하여 더한 것이다(0.1은 실험적으로 가장 알맞은 값을 선택한 것이다.) 또, 충성도가 낮고 다양성이 높을수록 외부로부터의 수용성은 높아지므로, 이런 사용자의 경우 외부로부터의 추천인 DF 결과 값을 중요하게 반영해줄 필요가 있다. 앞에 곱해진 두 인자 $(1-loyalty)^2$와 $diversity^2$는 이러한 이유로 곱해진 것이며, 제곱 인수는 실험에 의해 결정된 것임을 밝혀둔다. 이렇게 하여, 사용자의 다양도가 높을 경우, 자신이 선호하는 장르 외의 영화들도 평균 점수가 높을 경우 상당 부분 반영되나, 다양도가 낮을 경우에는 자신이 선호하는 장르외의 영화 추천 비율을 낮추는 방식으로 동작하게 되어 있다.

6. 실 험

6.1 대상 도메인 설명

본 논문이 제안할 조합 추천 기법의 효용성을 보이기 위하여, 영화 추천의 도메인 위에서 시스템을 구축하였다. 본 논문이 주장하는 조합 기법은 원칙적으로 도메인에 제약을 받지 않으나, 현실적으로 가장 쉽게 양질의 자료 및 사용자 정보를 구할 수 있는 영화 도메인에서 동작하는 시스템을 구축하였다.

본 논문에서는 Eachmovie[14]에서 제공한 영화 자료 및 사용자 평가 정보와 한국의 최대 영화 데이타베이스 사이트 Films[15], 미국의 영화 데이타베이스 사이트 IMDB[16]의 자료를 이용하여 시스템을 구축하였다. 본래 최근의 영화 추천 연구는 대개 Eachmovie dataset과 IMDB의 자료를 바탕으로 이루어지지만, 한국 사람을 대상으로 실험이 수행되었으므로, 국문으로 된 Films 데이타를 적극 이용하였다.

6.2 실험 방법

본 시스템의 궁극적인 Fusion 엔진의 결과를 순수 CBF 엔진의 결과와 순수 CF 엔진의 결과, CF와 CBF를 단순히 교대로 섞는 단순한 조합 기법의 성능과 비교하였다. 총 자료의 50%는 트레이닝 집합으로 50%는 실험 집합으로 이용하였다.

6.3 실험 평가 척도

추천 시스템을 위한 평가 척도는 여러 가지가 알려져 있으나, 대개 시스템의 전반적인 정확도를 측정하는 통계적 척도와 상위에 순위화 된 아이템이 얼마나 잘 추천되었는가를 판단하는 정성적인 척도로 나누어진다. 본 실험에서는 시스템의 전반적인 정확도를 보이기 위한 통계적 방법으로서 일반적으로 가장 많이 쓰이는 정규화 된 평균 절대 오차(NMAE)를 사용하였다. 그리고 상위의 영화들이 얼마나 사용자의 실제 평가와 맞게 추천되었는지를 알아보기 위한 coverage를 계산하여 성능 향상을 보였다. NMAE와 coverage는 각각 다음과 같이 정의된다.

$$NMAE = \frac{\sum_{(i,j)\in T} |r_{ij} - p_{ij}|}{Ns}$$

r_{ij}는 사용자 i가 영화 j에 대해 실제로 평가한 점수, p_{ij}는 추천 시스템이 예측한 점수,

n은 실험 대상 영화의 총 개수, s는 한 계급의 크기를 뜻한다. MAE의 정성적 의미는 사용자가 실제로 평가한 결과와 추천 시스템이 예측한 결과가 대체로 얼마나 차이가 나는가 하는 뜻을 담고 있다.

$$coverage_{0.6} = \frac{\text{사용자의실제평가에서0.6이상을받은영화들중실제로추천된영화갯수}}{\text{사용자의실제평가에서0.6이상을받은전체영화의갯수}}$$

coverage는 사용자가 실제로 재미있다고 평가한 영화가 얼마나 제대로 추천되었는지를 나타내는 척도이다. 즉, 순위의 상위부에서 얼마나 예측치가 잘 맞는지를 기술한다. 대부분의 사용자가 100편의 영화가 있을 때 10편이 못 되는 영화만을 관람한다는 사실을 생각해 볼 때, coverage는 실용적으로 유의한 지표이다. coverage가 높을수록 훌륭한 추천 시스템이다.

6.4 결 과

본 시스템이 제안한 fusion 알고리즘이 추천 결과를 향상시켰음을 볼 수 있다. coverage의 경우 상위의 결과일수록 본 fusion 알고리즘이 큰 성과를 거둠을 보여주는데 이는 영화를 조금만 보는 사람에게는 매우 귀중한 기능이라 할 수 있다.

추천 기법	NMAE	$coverage_{0.6}$	$coverage_{0.8}$
순수 CBF 기법	0.1472	31%	42%
순수 CF 기법	0.1110	63%	68%
interleave 조합	0.1012	66%	62%
fusion 1	0.0921	74%	81%
fusion 2	0.0902	76%	84%

7. 토론 및 결론

fusion엔진의 결과는 사용자에 따라서 성과의 정도가 많은 차이가 있었다. 대체로 충성도가 높고 다양도가 높은 사용자들에게 있어서 본 fusion 리스트의 만족도가 보다 좋게 나타났다. 그러나 충성도가 매우 낮은 사용자들에게 있어서 본 fusion엔진은 성능 향상을 크게 보이지 않았다. 또, 본 영화의 수가 많을수록, 긍정적으로 답한 영화가 많을수록 본 시스템의 결과가 좋게 나타나는 것을 관찰할 수 있었다. 이러한 측면에서 볼 때, 본 시스템은 동호회 등에서 쓰이면 좋은 반응을 얻을 것이라 생각된다.

결론적으로 사용자의 개인적 경향이 개인별로 바람직한 각 추천 기법을 결정한다는 전제는 틀리지 않음이 밝혀졌다. 사용자 경향에 기반하여 각 추천 기법을 동적으로 조합하는 기법은 한 쪽의 순수 추천 기법이나 기존의 조합 방법과 비교했을 때 좋은 성능 향상을 가져왔다.

또 중요한 점으로, 두 번째 fusion 단계의 도입으로 검색 엔진 평가에서 Novelty라 이야기하는 기대하지 않았던 유의미한 추천 결과를 바랄 수 있게 되었으며, CF의 결과가 조금 부정확한 경우에도 양호한 추천 성능을 보이는 것으로 나타났다. 이는 본 시스템의 2차에 걸친 조합 기법이 온라인 등의 제약 상황에서 CF 모듈의 동작이 여의치 않아도 양질의 추천 결과를 제공할 수 있음을 시사한다.

8. 감사의 글(Acknowledgements)

Dataset을 제공해 준 컴팩 리서치 센터와 쉽지 않은 결정을 내려 준 Films 측에 진심으로 감사를 표한다.

참 고 문 헌

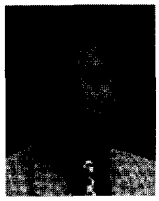
[1] Amazon, http://www.amazon.com

[2] M. J. Pazzani. A framework for Collaborative, Content-based and Demographic Filtering. Arti-

ficial Intelligence Review, 13(5-6):393-408, 1999.
[3] Douglas W. Oard, Gary Marchionini. Conceptual framework for text filtering, Technical Report CS-TR3643, 1996.
[4] P. Melville, R. J. Mooney and R. Nagarajan. Content-Boosted Collaborative Filtering. SIGIR 2001.
[5] Breese, Heckermen, and Kadie. Emperical analysis of predictive algorithms for collaborative filtering. Technical report, Microsoft Research, October 1998.
[6] M. Claypool, A.Gokhale, T. Minranda, P. Murnikov, D. Netes, and M. Sartin. Combining content-based and collaborative filters in an online newspaper. In ACM SIGIR WS on recommender Systems, 1999.
[7] M. Balabanovic and Y.Shoham. Fab: Content-based, collaborative recommendation. Communications of the Association of Computing Machinery, 40(3):66~72, 1997.
[8] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, John Riedl, GroupLens: An Open Architecture for Collaborative Filtering of Netnews., Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work.
[9] Kevin Kelly, New Rules for the New Economy, Viking, 1998.
[10] PEKING Annual Report 2002, http://www.interpeking.com
[11] C. Nightingale, A Mathmatical Model for Visual Taste when selecting from Repeating Patterns, 2001.
[12] Gabor Peli, Political Niches in the Voters' Space, 2001.
[13] Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers.
[14] EachMovie Dataset, http://research.compaq.com/SRC/eachmovie
[15] Films Database, http://films.hitel.net
[16] Internet Movie Database, http://www.imdb.com

이 수 정
2001년 서울대 물리학, 전산과학(학사) 2003년 서울대 컴퓨터공학부(석사). 관심분야는 데이타베이스, 정보검색

이 형 동
1997년 홍익대 컴퓨터공학과(학사). 1999년 서울대 컴퓨터공학과(석사). 1999년~현재 서울대 컴퓨터공학부 박사과정. 관심분야는 데이타베이스, 정보검색



김 형 주
1982년 서울대학교 전자계산학과(학사) 1985년 Univ. of Texas at Asution(석사) 1988년 5월~1988년 9월 Univ. of Texas at Austin. Post-Doc. 1988년 9월~1990년 12월 Georgia Institute of Technology(부교수). 1991년 1월~현재 서울대학교 컴퓨터공학부 교수