논문 2010-3-7

장르 기반 Collaborative Filtering 영화 추천

Genre-based Collaborative Filtering Movie Recommendation

황기태*

Kitae Hwang

요 약 Collaborative Filtering(CF) 기법에 기반을 둔 다양한 영화 추천 방법들이 제안 되어 왔다. CF는 영화를 본 사람들이 직접 영화에 대해 평가한 점수를 기반으로 같은 성향을 가진 이웃 그룹을 결정하고, 새로운 영화에 대해 그 영화를 이미 본 이웃의 점수를 기반으로 사용자의 새로운 영화에 대한 선호도 값을 예측하는 방법이다. 본 논문에서는 사용자에 따라 영화 장르에 대한 선호도 정보를 CF의 예측 값에 반영하는 새로운 방법을 제안한다. 이 방법은 CF를 기반으로 하는 모든 종류의 추천 방법에 결합하여 사용할 수 있다. 본 논문에서는 기존의 CF알고리즘에 장르기반 알고리즘을 결합한 CF-Genre의 성능과 기존의 CF 알고리즘의 성능을 측정 비교하였다. 성능 평가의 결과 CF-Genre가 기존 CF의 예측 성능을 3.3% 정도 개선하였다.

Abstract There have been proposed several movie recommendation algorithms based on Collaborative Filtering(CF). CF decides neighbors whose ratings are the most similar to each other and it predicts how well users will like new movies, based on ratings from neighbors. This paper proposes a new method to improve the result predicted by CF based on genres of the movies seen by users. The proposed method can be combined to the most of all existing CF algorithms. In this paper, a performance evaluation has been conducted between an existing simple CF algorithm and CF-Genre that is the proposed genre-based method added to the CF algorithm. The result shows that CF-Genre improves 3.3% in prediction performance over existing CF algorithms.

Key Words : Movie recommendation, Movie genre, Collaborative filtering

Ⅰ. 서 론

최근에 영화, 뮤직, 비디오, 스포츠, 드라마, 다큐멘트리 등 TV 디지털 콘텐츠가 매우 다양하며 한 해 무려 31,000,000 시간에 해당하는 디지털 콘텐츠가 쏟아져 나오고 있다. 이렇게 엄청한 양의 디지털 콘텐츠로부터 사용자는 자신이 보고자 하는 프로그램을 찾느라고 많은 시간을 낭비하게 된다.

추천(Recommendation)은 이러한 많은 상품으로부터 사용자가 가장 선호할 만한 상품을 자동으로 소개하는방법으로서, 1990년대부터 Minnesota 대학을 중심으로 하여 많은 연구와 발전을 이루어 왔다[1,2]. 대표적인 추천 기법은 Collaborative Filtering(CF)으로서 여러 사람들이 매긴 상품에 대한 점수를 기반으로 동일한 선호도를 가진 사람들인 이웃(neighbor)을 발견하고, 추천 받고자 하는 상품에 대한 이웃들의 평가 점수를 기반으로 추천 점수를 계산하는 방법이다. 또 다른 방법은 Content-based Filtering으로서 상품의 콘텐츠를 기반으로 사람들이 관심 있어 하는 콘텐츠 특성과 상품의 콘텐트 특성을 표현한 문서 내용을 비교하여 상품을 추천하는 방식이다. 이 방식은 모든 상품에 대해 콘텐츠 설명이 있어야만 추천이 가능하므로 CF에 비해 많이 사용되지는 않는다[3,4].

본 논문은 영화 추천을 위해 CF의 성능을 개선하는데 초점을 두었다. 영화는 추천의 전통적인 연구 대상이며

*정회원, 한성대학교 컴퓨터공학과
접수일자 2010.05.03 수정일자 2010.6.14

영화 추천 방법은 다른 비슷한 디지털 매체에도 동일하게 적용할 수 있기 때문에 추천의 기본이 되어 왔다.

영화 추천을 위해 제안된 대부분의 CF 알고리즘들은 주로 이웃을 결정하거나 상관 계수(correlation coefficients)를 계산하는 과정에 변형을 가함으로서 영화에 대한 사용자의 선호도의 예측 성능을 높이는데 주력하였다[3,5,6]. 그러나 본 논문에서는 영화 장르에 대한 사용자의 특별한 선호도가 있을 것이라는 직관으로 사용자가 본 영화의 장르에 통계를 내어 장르에 대한 선호도를 CF 알고리즘의 최종 예측 값에 결합하는 방식을 제안하였다.

본 연구팀은 이미 사전 분석 연구[7]를 통해 영화 장르가 예측성에 영향을 줄 수 있음을 어느 정도 확인하였으며, 본 논문에서는 장르에 대한 선호도를 계산하는 알고리즘을 만들고 이를 CF 알고리즘의 최종 예측 값을 수정하도록 함으로써 예측 성능을 향상시켰다. Minnesota 대학에서 오랫동안 축적한 영화 추천에 사용된 MovieLens 데이터베이스[4,6]를 이용하여 3,883편의 영화에 6,040명이 점수를 매긴 값을 분석하여 장르에 대한 사람들의 선호정보를 분석하고 이 데이터를 통해 이를 바탕으로 장르 기반 추천 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 CF 기반의 기존 영화 추천 연구에 대해 소개하고, 3장에서는 영화 장르에 따른 선호도를 분석하며 4장에서는 제안한 방법을 소개하고 성능 평가를 실시한 결과를 보이며 5장에서 결론을 맺는다.

Ⅱ. 관련 연구

1. Collaborative Filtering(CF)

영화 추천 알고리즘은 일반적으로 흔히 People-to-people filtering으로 불리는 Collaborative filtering(CF) 방법에 근간을 두고 있다[4].

CF는 임의의 사용자 U에 대해 영화 M에 대한 선호도 P(1~5 사이의 값)를 예측하는 과정으로서, 영화를 본 많은 사람들로부터 각 영화에 대해 평가한 값 R(Rating)이 필요하다. 일반적으로 R 값은 1에서 5까지의 값을 주로 사용하며 5는 가장 재미있었음을 의미한다. CF의 첫 단계는 U 와 영화 M을 본 다른 사람들 사이의 영화 선호도의 유사성을 판단하기 위한 상관계수(correlation coefficients)를 구하는 단계이다. 이 값을 구하기 위해서 U 뿐만 아니라 영화 M을 본 다른 사람들이 이미 여러 영화에 대한 만족도를 표시하는 R 값을 매긴 기초 데이터가 필요하다. 상관계수는 -1에서 1사이의 실수 값으로 1에 가까울수록 U와 해당 사용자 사이의 영화에 대한 취향이 비슷함을 의미한다.

두 번째 단계는 이들 상관 계수로부터 유사성이 높은 사용자의 그룹인 이웃 N(neighbor)을 선정하는 단계이다. N의 크기가 크면 동질성이 다소 떨어지게 되어 예측성에 악영향을 미치게 되고 예측 알고리즘 수행에 따른 처리 시간이 너무 길어지게 되는 단점이 있다. 따라서 N의 크기를 적절히 설정하여야 한다[4].

세 번째 단계는 N으로부터 사용자 U가 영화 M에 대해 어느 정도 만족할 것인지를 예측치를 구하는 과정이다.

그림 1은 CF의 전체 과정을 나타낸다[1]. 우선 그림의 최상위의 보이는 2차원 행렬 값은 임의의 사용자 U의 영화 M에 대한 선호도를 예측하기 위하여 U 이외의 사용자들이 각 영화를 평가한 값 R로 구성된다.

그림 1에서 식 1-1은 피어슨 상관계수를 나타내는 수식이다. 사용자 U와 그 밖의 모든 사용자와의 상관계수를 구하여 U의 이웃 N을 선별한다. 식에서 X는 사용자 U에 해당한다. 즉, μx는 U의 전체 Rating에 대한 평균을, σx는 U의 Rating에 대한 표준 편차를 의미한다. 다음으로 Y는 사용자 U를 제외한 모든 사용자이다.

식1-1 하단의 ④는 식1-1을 적용한 결과로서 사용자 U와 사용자 U1~U3사이의 상관계수이다. 사용자 U와 유사한 성향을 지닌 이웃을 결정하기 위하여 상관계수가 일정한 기준 이상을 갖는 사용자를 선별한다.

식1-2는 예측 값을 얻어내는 수식이다. μx는 사용자 U의 평균값이고 Y_m은 사용자 U 이외의 다른 사용자들이 임의의 영화 M에 준 Rating 값이다. μy는 타 사용자의 Rating에 대한 평균이고 ρxy는 사용자 U와 타사용자 사이의 상관계수 값이다[1]. 식1-2로 계산한 결과는 바로 사용자 U가 영화 M을 보았을 때 만족도를 예상한 값이다. 이 값 역시 1-5사이의 값으로 5일 때 가장 높은 만족도를 나타낸다.

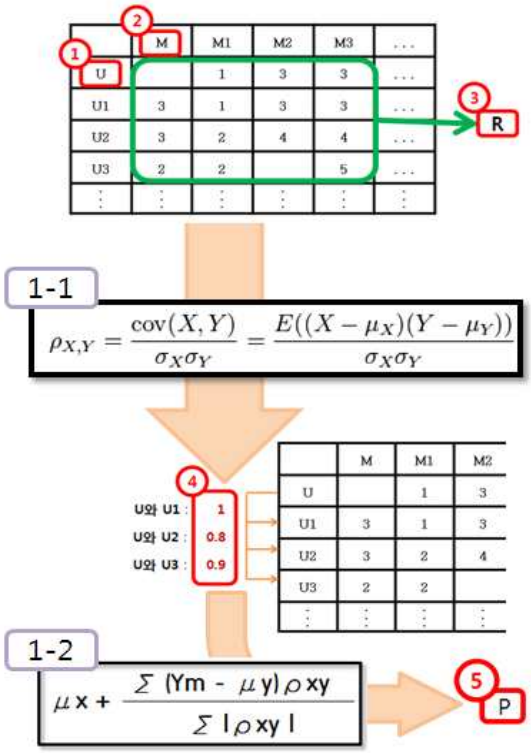

그림 1. CF 알고리즘 수행 예
Fig. 1. Execution Example of CF Algorithm

2. CF 관련 연구

CF 연구는 1992년 시작된 미국의 MIT와 미네소타 대학에서 공동으로 네트워크 상의 뉴스 그룹 사이에 교환되는 뉴스를 읽고자 하는 사용자에게 적합한 뉴스를 추천하는 Collaborative Filtering 시스템인 GroupLens를 만듦으로써 본격화되었다. 이후 GoupeLens 연구팀은 CF 알고리즘을 계속 개선하여 왔다[1,2].

Ringo 음악 추천 시스템[3] 은 초기 GroupLens 방식을 수정하여 상관 계수가 일정 경계치보다 큰 값 들에 대해서만 이웃을 설정하는 방식을 사용하였으며, Bellcore의 비디오 추천 시스템[5]은 피어슨 상관 계수 계산 방식에 가중치를 주는 방식으로 최고의 이웃을 선택하려는 시도를 하였다. 영화를 보는 사용자의 특성 프로파일을 이용하여 CF와 하이브리드 형태의 알고리즘으로 충분한 Rating 정보가 없는 상황에서도 추천할 수 있고 추천의 예측성도 높이는 방법을 제안하였다.

영화/비디오의 추천 시스템인 MovieLens 시스템은 사용자 사이의 상관 계수를 계산할 때 다양한 가중치 방법을 제안하여 사용자 사이의 상관성을 보다 높이고자 하였으며, 이웃을 결정할 때 다양한 지수를 이용하여 최고의 이웃을 선정하도록 실험하여 성능을 조금 개선하기도 하였다[4,6].

추천은 Amazon, CDNOW, Drugstore, eBay, MovieFinder, Reel, Netfilx, 등 전자상거래에도 많이 이용되었으며[8,9,10], MovieLens 팀은 People-to-People 방식에 사용자와 영화 아이템 사이의 상관 관계를 분석하여 반영하는 Item-to-Item 방식을 다소 결합하여 영화 추천 성능의 개선을 시도하였다[4,11].

2000년대 초반까지는 CF의 알고리즘 성능을 개선하려는 많은 시도가 있었지만 추천 성능은 추천 값과 사용자의 실제 점수와의 평균 오차(MAE)가 0.7 정도 근처에서 적은 수준으로 개선되는 정도에 머물렀으며, 2000년도 후반에는 추천 응용 시스템 개발에 관한 연구가 주를 이룬다[12].

III. 영화 장르에 따른 선호도 분석

1. 선호도 분석의 목적

직관에 따르면 영화 장르에 대한 각 사용자의 선호도가 어느 정도 있는 것으로 판단된다. CF 알고리즘은 개인의 성향을 직접적으로 반영하기보다는 집단적인 성향을 파악하여 개인의 성향이 반영되도록 한다. 본 논문은 개인의 성향을 직접적으로 반영하여 영화에 대한 선호도를 결정하기 위해 영화 장르에 대해 개인의 선호도를 분석한다. 분석된 결과를 토대로 CF 알고리즘에서 예측한 영화에 대한 선호도 값에 장르기반의 가중치를 주는 방법을 제안한다.

선호도 분석은 사전 연구[7]를 통해 기초 분석을 진행하였으며 이 절에서 기초분석의 내용을 요약한다.

2. 선호도 분석에 사용된 데이타베이스

영화 장르에 대한 사용자의 선호도 분석을 위해 우리는 MovieLens에서 지난 10년 동안 만들어 데이터베이스를 사용하였다. 이 데이터베이스는 크게 다음과 같다.

(1) Movie DB

총 3,883개의 영화에 대해 영화 ID, 타이틀, 장르의 정보를 가진 DB로서 표 1과 같다.

표 1. Movie DB의 구성
Table 1. Attributes of Movie DB

Field 속성	의미
MovieID	각 Movie 마다 존재하는 ID로서 0부터 싲가하는 숫자가 부여된다.
Title	Movie의 제목으로서 문자열로 구성된다.
Genres	Movie의 해당 Genres로서 하나 이상의 Genres로 구성된다. 총 18개의 Genres가 존재하며 하나 이상의 Genres는 ‘,’로 구분된다.

(2) User DB

총 6,040명의 사용자에 대해 사용자 ID, 성별, 나이, 직업 등의 정보를 가진 DB로서 표 2와 같다.

표 2. User DB의 구성
Table 2. Attributes of User DB

Field 속성	의미
UserID	각 User 마다 존재하는 ID로서 1~6040사이의 숫자가 부여된다.
Gender	User의 성별로서 남성의 경우 ‘M’, 여성의 경우 ‘F’가 부여된다.
Age	User의 연령대로서 7개의 숫자로 구분된다. 각 숫자는 연령대의 대표값을 나타낸다.
Occupation	User의 직업으로서 21개군으로 구분된다. 각 직업군은 0~20까지의 숫자가 부여된다.
Zip-code	User의 주소를 의미한다.

(3) Rating DB

총 6,040명의 사용자가 총 3,883영화에 대해 자신이 본 영화에 대해 1-5까지의 점수를 매긴 DB로서 표 3과 같다.

표 3. Rating DB의 구성
Table 3. Attributes of Rating DB

Field 속성	의미
UserID	각 User 마다 존재하는 ID로서 1~6040사이의 숫자가 부여된다.
MovieID	각 Movie 마다 존재하는 ID로서 0부터 시작하는 숫자가 부여된다.
Rating	각 User가 관람한 Movie에 부여한 점수로서 1~5사이의 숫자가 부여된다. 5가 가장 높은 점수이다.
Timestamp	User가 Movie에 Rating을 부여한 시점으로서 9자리의 숫자로 부여된다.

3. 분석

본 논문에서는 각 사용자에 대해서 영화 장르와의 취향을 결정하기 위해 GroupLens 데이터베이스를 대상으로 다음과 같은 절차로 분석하였다.

(1) 영화의 장르 벡터 결정

현재의 영화의 장르는 일반적으로 18개의 장르로 구분되며 표 4와 같다.

표 4. 영화의 18 장르
Table 4. 18 Movie Genres

No.	장르	No.	장르
G1	Action	G10	Film_Noir
G2	Adventure	G11	Horror
G3	Animation	G12	Musical
G4	Children	G13	Mystery
G5	Comedy	G14	Romance
G6	Crime	G15	Sci-Fi
G7	Documentary	G16	Thriller
G8	Drama	G17	War
G9	Fantasy	G18	Western

모든 영화의 장르는 다음의 18차원의 벡터 로 표시할 수 있으며 영화 m의 장르 벡터는 다음과 같이 표시한다.

$$GV_m = (g_1, g_2, .., g_{18})$$

여기서 m은 영화를 의미하며, g_i는 영화 m이 가진 i번째 장르 특징을 가지고 있는지 판단하는 상수로서 0 또는 1의 값이다. 1인 경우 장르 특징을 가지고 0인 경우 장르의 특징을 가지지 않는다. 예를 들어 영화 'Speed'의 장르는 G1(Action), G14(Romance), G16(Thriller)에 해당하므로 다음과 같다.

$$GV_{Speed} = (1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0)$$

(2) 사용자의 영화 장르에 대한 취향 벡터 결정

사용자가 본 영화를 바탕으로 각 사용자에 대해 영화의 취향을 장르로 표시할 수 있다. 이 벡터 역시 18차원으로 표시 가능하다. 사용자 u의 취향 벡터는 다음과 같이 얻을 수 있다.

$$PV_u = \sum_{m \in movies\ of\ seen\ by\ u} (GV_m \cdot R_{um})$$

여기서, R_{um} 은 영화 m에 대한 사용자 u의 Rating 값이다.

예를 들어 사용자 u가 시청한 영화가 표 5와 같을 때 u의 취향 벡터를 구하면 다음과 같다.

$$\begin{aligned} PV_u = & (0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0) \times 2 \\ & + (1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0) \times 4 \\ & + (0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0) \times 5 \\ & = (4,0,2,2,2,9,0,0,0,0,0,0,0,0,0,9,0,0) \end{aligned}$$

표 5. 장르 예
Table 5. Example of Genres.

영화	장르	Rating
Toy Story	G3(Animation), G4(Children),G5(Comedy)	2
Heat	G1(Action),G6(Crime), G16(Thriller)	4
Seven	G6(Crime),G16(Thriller)	5

(3) 장르에 대한 일반적인 선호도 분석

장르에 대한 일반적인 선호도가 있는지를 분석한다. R_{um} 은 영화 m 에 대한 사용자 u의 Rating 이고, $\overline{R_u}$ 는 사용자 u가 시청한 모든 영화에 대한 Rating의 평균이라고 하면, $R_{um} > \overline{R_u}$ 는 사용자 u는 영화 m을 재미있게 시청 하였고, 그 외의 경우는 그렇지 않다고 판단한다. 이러한 전제 하에 모든 사용자의 Rating의 평균과 시청한 모든 영화의 Rating을 토대로 장르에 대한 일반적인 선호도를 분석한 결과 표 6과 같다. 표에서 Fun 열의 Y는 "재미있다"이며 N는 "재미없다"이고 Y(%)는 "재미있다"의 백분율 값이다.

표 6. 장르에 대한 일반적인 선호도 분석
Table 6. Analysis of General Preferences of Users

Fun	G1	G2	G3	G4	G5
Y	131,670	67,174	25,255	35,371	186,634
N	125,787	66,779	18,038	36,815	169,946
합	257,457	133,953	43,293	72,186	356,580
Y(%)	51.14	50.15	58.34	49.00	52.34

Fun	G6	G7	G8	G9	G10
Y	46,464	5,299	213,642	17,711	12,825
N	33,077	2,611	140,887	18,590	5,436
합	79,541	7,910	354,529	36,301	18,261
Y(%)	58.42	66.99	60.26	48.79	70.23

Fun	*G11*	G12	G13	G14	G15
Y	*33,180*	23,406	22,805	79,874	78,904
N	*43,206*	18,127	17,373	67,649	78,390
합	*76,336*	41,533	40,178	147,523	157,294
Y(%)	*43.44*	56.36	56.76	54.14	54.14

Fun	G16	G17	G18
Y	102,679	44,550	11,549
N	87,001	23,977	9,134
합	189,680	68,527	20,683
Y(%)	54.13	65.01	55.84

이 결과에 따르면 공포 영화에 대해서는 재미있다는 반응보다 재미없다는 반응이 상대적으로 월등히 많다. 이 결과가 주는 의미는 공포 영화에 대한 매니아를 제외한 많은 사람들은 공포 영화를 좋아하지 않는다는 것을 의미한다.

Ⅳ. 장르 기반 선호도 가중치 알고리즘

1. 장르 기반 선호도 가중치

CF 알고리즘은 한 영화에 대해 한 사용자가 어떤 평가를 할 것인지 미리 예측하는 알고리즘이며 그 결과 값은 1-5 사이의 예측 값이다. 장르 기반 선호도 가중치는 CF 알고리즘에 대한 예측값을 보정하는 값으로서 예측하고자 하는 사용자 u에 대해 다음의 과정으로 계산한다.

(1) 사용자의 u의 장르별 Rating 합계 벡터

이 값은 다음과 같은 수식으로 표시되며, 사용자 u가 본 모든 영화에 대한 장르 벡트를 합한 값이다.

$$\begin{aligned} PV_u &= \sum_{m \in movies\ of\ seen\ by\ u} (GV_m \cdot R_{um}) \\ &= (r_1, r_2, ..., r_{18}) \end{aligned}$$

(2) 사용자 u의 장르별 Rating Count 벡터

PV_u 벡터를 구성할 때 사용된 정보로서 각 장르의 Rating 값을 계산할 때 장르마다 더해진 횟수를 나타 내며 다음과 같이 표현된다.

$CV_u = (c_1, c_2, ..., c_{18})$,
여기서 c_i는 i번째 장르의 $Rating$ 합한 횟수

(3) 사용자 u의 장르별 Rating 평균 벡터

사용자 u의 장르별 Rating 평균 벡터 EV_u는 다음과 같은 수식으로 표시된다.

$$EV_u = (r_1/c_1, r_2/c_2, ..., r_{18}/c_{18}) = (EV_{u1}, EV_{u2}, ..., EV_{u18})$$

즉, 사용자 u의 장르별 Rating 평균 벡터는 PV_u 벡터의 각 원소 값을 CV_u 벡터의 각 원소 값으로 나눈 값로 구성된다.

(4) 사용자 u의 Rating 평균 값

u가 본 영화의 점수 평균값을 다음과 같이 구한다.

$$E_u = \sum_{i=1}^{18} (r_i/c_i)/18$$

(5) 사용자 u의 영화 m에 대한 선호도 가중치, W_m

영화 m의 장르에 대해 사용자 u가 선호할 가중치 값을 계산한다.

$$W_{um} = \sum_{i=1}^{18} (EV_{ui} - E_m) \times g_{mi}$$
여기서 g_{mi}는 영화 m의 GV_m의 i번째 원소

GV_m의 i번째 원소인 g_{mi}는 영화 m이 장르 i에 해당하면 1이고 아니면 0인 값이다.

가중치 값에 대한 이해를 돕기 위해 예를 들어보자. 사용자 u가 G1, G3, G5의 장르를 가지는 영화 1과 G3, G5, G6의 장르를 갖는 영화 2를 시청하였고, 1번 영화에 4점을, 2번 영화에 3 점을 주었다면, PV_u, CV_u, E_u는 표 7, 8, 9와 같다.

표 7. 사용자 u의 장르별 Rating 합계 벡터
Table 7. Rating Summation Vector of U

G1	G2	G3	G4	G5	G6	G7	G8	G9
4	0	7	0	7	3	0	0	0
G10	G11	G12	G13	G14	G15	G16	G17	G18
0	0	0	0	0	0	0	0	0

표 8. 사용자 u의 장르별 Rating Count 벡터
Table 8. Rating Count Vector of Each Genre of U

G1	G2	G3	G4	G5	G6	G7	G8	G9
1	0	2	0	2	1	0	0	0
G10	G11	G12	G13	G14	G15	G16	G17	G18
0	0	0	0	0	0	0	0	0

표 9. 사용자 u의 장르별 Rating 평균 벡터
Table 9. Rating Average Vector of U

G1	G2	G3	G4	G5	G6	G7	G8	G9
4.0	0	3.5	0	3.5	3.0	0	0	0
G10	G11	G12	G13	G14	G15	G16	G17	G18
0	0	0	0	0	0	0	0	0

이제 가중치를 구하는 마지막 과정을 살펴보자. 이를 위해 표 10과 같은 장르별 Rating 평균 벡터가 있다고 가정한다. 그리고 이들의 평균은 3.6이 되며 이 값이 바로 만일 영화 1이 장르 G7과 G11로만 이루어진 경우 W_1은 다음과 같다.

W_1 = (3.7-3.6) + (3.0-3.6) = -0.5

다른 경우로 만일 영화 2가 장르 G1, G8, G17로 이루어진 경우 W_2는 다음과 같이 계산된다.

W_2 = (3.5-3.6) + (4.2-3.6) + (4.0-3.6) = 0.9

표 10. 사용자 u의 장르별 Rating 평균 벡터에 가정치
Table 10. Assumption of Rating Average Vector of U

G1	G2	G3	G4	G5	G6	G7	G8	G9
4.5	3.2	2.8	4.2	3.6	3.5	3.7	4.2	3.9
G10	G11	G12	G13	G14	G15	G16	G17	G18
4.1	3.0	0.0	4.1	2.5	3.4	3.7	4.0	3.1

2. 장르 기반 선호도 가중치를 반영한 추천

본 논문에서는 장르를 기반으로 사용자 u의 영화에 대한 특성을 파악하고 새로운 영화 m에 대한 선호도의 예측 값을 계산할 때 영화 m의 장르가 사용자 u가 선호하는 것인지를 계산한 가중치, W_m을 반영하는 추천 방법을 제안한다.

이 방법은 CF 알고리즘을 근본적으로 바꾸는 것이 아니라 CF 알고리즘의 마지막 단계에서 나오는 예측치를 수정하는 방법이다. 그러므로 이 알고리즘을 CF-Genre라고 명명한다. CF-Genre에서의 영화 m에 대한 최종 예측치는 다음과 같이 계산한다.

CF-Genre의 영화 m에 대한 최종 예측치 =
영화 m에 대한 CF 알고리즘의 예측치 - W_m

3. 장르 기반 추천 알고리즘의 성능 평가

(1) 성능 평가 모델

장르기반 추천 알고리즘은 CF 알고리즘의 결과를 보정하는 방법이므로 지금까지 연구된 모든 CF 관련 알고리즘에 적용될 수 있다. 그러므로 본 논문에서 제안하는 장르 기반 추천 알고리즘의 성능을 평가하기 위해서는 일반적인 단순한 CF 추천 시스템과 여기에 장르 기반 알고리즘을 결합한 추천 시스템의 성능을 비교하여 장르 기반 알고리즘의 추가로 인해 성능 향상이 발생하였는지를 평가한다. 비교의 기본이 되는 CF 기반의 추천 알고리즘을 CF-A라고 명명한다. CF-Genre는 CF-A에 장르 기반 방법을 결합시킨 것이다.

본 성능 평가를 위해 미네소타 대학의 MovieLenz 데이터베이스를 사용하였으며, 이 데이터베이스에 담긴 모든 영화에 대해 각각 GV_m 벡터를 생성하였다. 6040명의 사용자 수를 랜덤하게 300명을 추출하고 각 사용자에 대해 추천시 300개 영화를 랜덤하게 추출하여 실험하였다. 이웃을 정하기 위해 사용하는 correlation-threshould를 0.5로 하였으며 0.5 이상의 상관 계수를 가진 다른 사용자를 이웃으로 분류하였다. 성능 지수는 사용자가 대답한 실제 점수와 예측 값 사이의 오차 평균인 MAE로 하였다. MAE는 다음과 같으며 많은 연구에서 추천 알고리즘의 성능평가 기준으로 사용되고 있다.

MAE(Mean Absolute Error)
$1/n\sum_{i=1,n}|f_i-y_i|$, where f_i is prediction and y_i is true value.

(2) CF-A의 성능 평가

CF-A의 성능을 실험하는 이유는 CF-A와 CF-genre와의 성능 비교의 기준이 되기에 충분한 것인지를 보이기 위함이다.

CF-A 알고리즘의 실험 결과는 그림 2와 같다. 실험에서 이웃의 개수(Neighbor Number)를 제한하였다. 이웃의 개수가 많으면 많을수록 알고리즘의 계산 시간이 증가한다. 기존의 연구에 따르면 이웃의 개수는 20개 정도가 적합하다는 결과를 보이고 있지만 본 논문에서는 다양한 이웃의 개수에 따라 어떤 변화가 발생하는지 실험해 보았다.

CF-A의 MAE 값의 평균값을 계산하면 약 0.73이다. CF를 변형한 많은 알고리즘은 MAE 값이 대체로 0.7을 기준으로 조금씩 향상되는 면이 있다. 그러나 많은 연구들의 결과는 0.7을 기준으로 크게 향상되는 것은 아니다. 그림 2의 실험 결과는 CF-A가 보통의 CF 성능에 준하도록 작성되었음을 보이고 있다. 이 결과는 CF-A가 기존의 CF 알고리즘의 성능에 유사한 것으로 CF-Genre와 비교를 위해 충분한 대상 기준이 됨을 보여준다.

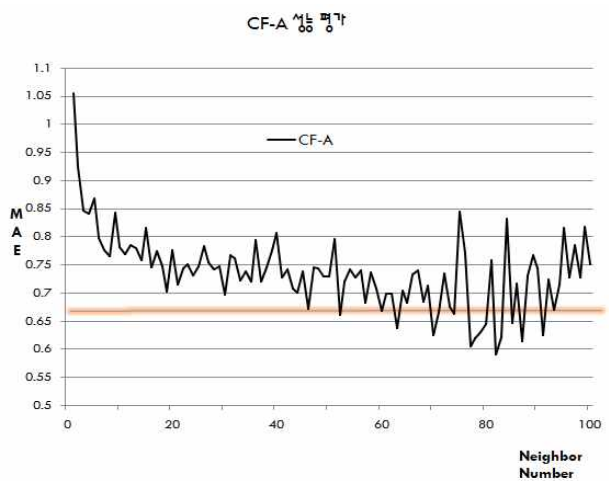

그림 2. CF-A의 성능 평가 결과
Fig. 2. Result of Performance Evaluation of CF-A

(3) CF-A와 CF-Genre의 성능 비교

그림 3은 CF-A와 CF-Genre의 두 방법에 대한 성능 비교를 보여 준다. 그림에서 CF-Genre 알고리즘이 CF-A의 성능을 개선하고 있음으로 볼 수 있다.

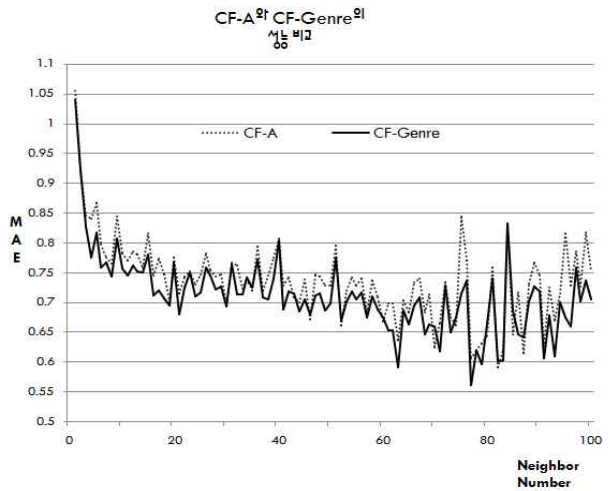

그림 3. CF-A와 CF-Genre의 성능 비교
Fig. 3. Performance Comparison of CF-A and CF-Genre

성능 개선 정도를 계산하기 위한 식은 다음과 같이 설정하였다.

$$\text{성능개선}(\%) = \frac{\overline{MAE}_{CF\text{-}A} - \overline{MAE}_{CF\text{-}Genre}}{\overline{MAE}_{CF\text{-}A}} \times 100$$

$\overline{MAE}_{CF\text{-}A}$ 는 이웃의 크기 0~100 개의 모든 경우에 대해 CF-A의 MAE 값의 평균값을 계산한 것으로 약 0.73 이며, $\overline{MAE}_{CF\text{-}Genre}$ 는 이웃의 모든 경우에 대해 CF-Genre의 MAE 값의 평균값을 계산한 것으로 0.711 이다. 성능 계산 식에 따르면 CF-Genre는 CF-A의 성능을 약 3.3% 개선한다. 3.3%의 성능은 결코 작은 수가 아니다. CF의 모든 알고리즘에 적용될 수 있기 때문이다.

(4) CF-Genre 알고리즘의 실행 시간 현실성

이론적으로 알고리즘의 성능 개선 정도가 높다 하더라고 실제 알고리즘 실행 시간이 비현실적이면 실세계에서 적용이 불가능하다. 본 논문에서는 CF-Genre 알고리즘의 실행 시간 성능을 측정하였다. 측정에 사용된 컴퓨터는 1GB의 메모리를 가진 펜티엄 PC로서 범용으로 사용되는 컴퓨터이며, 총 10회 실험의 수행 결과는 표 11과 같이 알고리즘의 수행 시간이 평균 698ms로서 1초가 걸리지 않았다. 그러므로 이 알고리즘은 인터넷 상의 웹에서 서비스하기에도 충분한 시간이다.

표 11. CF-Genre 알고리즘의 실행 시간
Table 11. Execution Time of CF-Genre Algorithm

실행횟수	이웃을 찾는 시간(ms)	예측 값을 계산하는 시간(ms)	시간(ms)
1	370	316	687
2	377	323	700
3	375	326	701
4	374	320	694
5	375	318	694
6	373	319	692
7	366	316	683
8	382	325	708
9	379	333	712
10	376	334	710
평균	375	323	698

(5) CF-Genre의 성능 개선의 의미

본 연구는 최고의 추천 알고리즘을 찾고자 하는 것이 아니다. 본 논문에서 제안하는 장르 기반 추천 알고리즘은 CF 알고리즘을 기반으로 하는 많은 알고리즘에 추가적으로 사용할 수 있기 때문에 본 성능 평가는 기본적인 CF 알고리즘에 대해 장르 기반 추천 방식을 더한 CF-Genre가 성능을 개선하느냐 하는 점을 평가하고자 함이다. 성능 개선의 정도가 3.3%이므로 단순 계산에 의하면 현존하는 최고의 CF 추천 알고리즘에 본 논문에서 제안한 장르 기반 추천 방식을 결합하면 3.3%의 성능을 여전히 개선할 수 있다는 것이다. 추후 연구를 통해 현존하는 모든 CF 알고리즘을 분석하고 본 논문의 장르 기반 가중치 알고리즘이 쉽게 적용될 수 있는지 확실히 검증할 필요가 있다.

V. 결 론

Collaborative Filtering(CF)은 여러 사람들이 매긴 영화에 대한 점수를 기반으로 동일한 선호도를 가진 사람들인 이웃(neighbor)을 발견하고, 추천 받고자 하는 영화에 대한 이웃들의 평가 점수를 기반으로 추천 점수를 계산하는 방법이다. 이 방법은 이웃을 얼마나 정확히 찾아내느냐가 성능을 결정한다. 또한 많은 연구를 통해 이웃을 결정하는 방법들이 제안되었지만 성능은 작은 범위에서 밖에 향상되지 못하였다.

본 논문에서는 영화의 장르를 기준으로 사용자의 취향을 분석하고 이 분석된 정보를 이용하여 CF 알고리즘의 예측 값을 보정하는 하이브리드 방법론을 제안하였다. MovieLens 데이터베이스를 활용하여 기존의 CF 알고리즘과 CF 알고리즘에 장르 기반 가중치를 적용한 CF-Genre 방법을 실험한 결과 예측치와 Rating 실제 값 사이의 평균 오차율을 3.3% 개선하였다. 이 방법은 어떠한 CF 알고리즘에도 적용이 가능하다는 큰 장점을 가진다.

참 고 문 헌

[1] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Rie, J., "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," *Proceedings of ACM CSCW94*

Conference on Computer Supported Cooperative Work, 1994.

[2] Kostan J., Miller B., Maltz D., Herlocker J., Gordon L., Riedl J., "GroupLens: Applying Collaborative Filtering to Usenet News" *Communications of the ACM*, Vol. 40, No. 3, pp. 77-87, 1997

[3] Marco Balabanovic and Yoav Shoham, "Fab: Content-based collaborative recommendation", *Communications of the ACM*, Vol, 40, No, 3, pp.66-72, 1997

[4] Good N.,Schafer J. B., Kostan J., Borchers A., Sarwar B., Herlocker J., and Riedl J., "Combining Collaborative Filtering with Personal Agents for Better Recommendations", *Conf. on the American Association of Artificial Intelligence.* pp. 439-446, 1999.

[5] Will Hill, Larry Stead, Mark Rosenstein, and George Furnas, "Recommending and evaluating choices in a virtual community of use". *Proc. of ACM CHI'95 Conf. in Human Factors in Computing Systems*, pp. 195-201, 1995.

[6] Herlocker J., Konstan J., Borchers A., Riedl J., "An Algorithm Framework for Performing Collaborative Filtering", *Proc. of ACM SIGIR'99, ACM Press*, 1999

[7] Jin Won Park, Min Cheul Shin, Sang Min Choi, Kitae Hwang, "Analysis for Genre-based Movie Recommendation", *Journal of Engineering Research, Hansung University*, 2009

[8] J. Ben Schafer, Joseph Konstan, Jhon Riedl, "Recommender Systems in E-Commerce," *GroupLens Research Project Department of Computer Science and Engineering University of Minnesota*, 1999.

[9] J. Ben Schaferm Joseph A. Konstan, John Riedl, "E-Commerce Recommendation Applications", *Journal of Data Mining and Knowledge Discovery*, Vol. 5, No, 1/2, pp.115-152, 2000

[10] Badrul Sarwar, George Karypis, Joseph Konstan, and John Rie, "Analysis of Recommendation Algorithms for E-Commerce," *The ACM E-Commerce 2000 Conference*, 2000.

[11] Badrul Sarwar, George Karypis, Joseph Konstan, and John Rie, "Item-based Collaborative Filtering Recommendation Algorithms," *Accepted for publication at the WWW10 Conference*, May, 2001.

[12] Ling. K., Beenen G., Ludford P., Wang X., Chang K., Li X., Cosley D., Frankowski D., Terveen L., Rashid A. M., Resnick P., Kraut R. "Using Social Psychology to Motivate Contributions to Online Communities", *Journal of Computer-Mediated Communication*, Vol. 10, No. 4, 2005

저자 소개

황 기 태(정회원)

- 서울대학교 컴퓨터공학과 학사
  서울대학교 컴퓨터공학과 석사
  서울대학교 컴퓨터공학과 박사
- 경력
  University of Callifornia, Irvine 방문교수

<주관심분야 : 모바일 & 유비쿼터스 시스템, 콘텐트 스트리밍>

※ 본 연구는 2010년 한성대학교 교내 연구비를 지원받아 수행되었음