한국빅데이터학회지
제2권 제1호, 2017, pp. 41-48

무비렌즈 데이터를 이용한 하이브리드 추천 시스템에 대한 실증 연구

An Empirical Study on Hybrid Recommendation System Using Movie Lens Data

김동욱[1] · 김성근[2] · 강주영[1†]

아주대학교 경영대학 e-비즈니스학과[1], 경영정보학과[2]

요 약

최근 추천 시스템의 인기와 함께 추천 시스템의 알고리즘의 성능에 대한 평가가 중요해 졌다. 본 연구는 영화 데이터에서 다양한 알고리즘 중 어떤 알고리즘의 효과적인지 판단하기 위하여 모델링과 RMSE를 통한 모델 검증을 하였다. 본 연구의 데이터는 무비렌즈의 평가 데이터 10만건을 활용하여 피어슨 상관계수를 활용한 사용자 기반 협업 필터링, 코사인 상관계수를 활용한 아이템 기반 협업 필터링 그리고 특이 값 분해를 활용한 아이템 기반 협업 필터링 모델을 만들었다. 세가지 추천 모델로 평점을 예측한 결과 사용자 기반 협업 필터링보다 아이템 기반 협업 필터링의 정확도가 월등히 높은 것을 확인했고, 행렬 분해를 사용했을 때 더 정확한 추천을 할 수 있었다.

■ 중심어 : 무비렌즈, 하이브리드 추천, 추천 시스템

Abstract

Recently, the popularity of the recommendation system and the evaluation of the performance of the algorithm of the recommendation system have become important. In this study, we used modeling and RMSE to verify the effectiveness of various algorithms in movie data. The data of this study is based on user-based collaborative filtering using Pearson correlation coefficient, item-based collaborative filtering using cosine correlation coefficient, and item-based collaborative filtering model using singular value decomposition. As a result of evaluating the scores with three recommendation models, we found that item-based collaborative filtering accuracy is much higher than user-based collaborative filtering, and it is found that matrix recommendation is better when using matrix decomposition.

■ Keyword : Movie Lens Data, Recommendation System, Hybrid Recommendation System

Ⅰ. 서 론

미국의 비디오 대여 업체 넷플릭스(Netflix)는 2006년 10월 넷플릭스 프라이즈(Netflix Prize)라는 알고리즘 대회를 개최했다[4]. 이 대회의 목적은 기존 넷플릭스 알고리즘인 cinematch의 RMSE(평균 제곱근 오차, Root Mean Square Error) 값을 10%를 개선시키는 것이었다. 이 대회를 위

2017년 01월 18일 접수; 2017년 2월 06일 게재 확정.
† 교신저자 jykang@ajou.ac.kr

해 넷플릭스는 1995년부터 2005년 동안의 비디오 대여 평점 데이터를 제공했다. 이 대회의 상금은 약 10억 원 정도로 당시의 많은 통계학자, 물리학자들이 도전했다. 기존의 Cinematch 알고리즘은 RMSE 값이 0.9514로 상당히 낮은 수치를 갖고 있었는데 이를 개선하기 위해서는 RMSE가 0.8563보다 작은 알고리즘을 만들어야 했다. 하지만 아무 사전 지식이 없는 참여자들이 Cinemath를 따라잡는데 겨우 1주일이 걸렸고 8.26%를 달성하는데 고작 10개월이 걸렸다. 때문에 참여자들은 곧 10%를 달성할 것이라고 예상 했으나 RMSE값을 0.8616 낮추는데 1년이 더 걸리고 당시 상위권에 위치한 팀이었던 Bellkor와 BigChaos가 연합해서 2009년 6월에서야 10%를 달성했다[3]. 이러한 과정에서 많은 경영인과 연구자들이 추천 시스템에 대해서 관심을 갖기 시작했다.

넷플릭스 프라이즈(Netflix Prize) 이후로 추천 시스템의 중요성이 강조되어 왔다. 특히, 마케팅 활동에 대한 인식이 가치 기반 마케팅 에서 알고리즘 기반의 마케팅 활동으로 혁명적으로 변화했다. 이러한 추세는 최근 4차 산업혁명의 중심인 알고리즘 기반 AI와 맞물려 더욱 발전할 것으로 기대된다.

이미 해외의 다양한 기업들은 추천 시스템을 적극적으로 활용하고 있다. 미국 최대의 온라인 서점인 아마존(Amazon)의 경우, A9라는 고유의 추천 시스템을 갖고 있다. 아마존의 약 35%의 매출이 추천 상품에서 발생할 정도로 추천 시스템을 통해 많은 매출을 올리고 있다[10]. 또한 구글의 유튜브(Youtube)의 경우 개인화된 추천 시스템(Personalized recommendation)과 협업 필터링(Collaborative filtering)을 통해 개인의 취향을 고려한 비디오 보여줌으로써 세계 최대의 온라인 비디오 스트리밍 사이트로 발돋움 했다[7].

하지만 국내의 경우 추천 시스템의 도입이 미비한 실정이다. 영화, 쇼핑 분야에서 추천 시스템을 어느 정도 도입을 하고 있지만 사용자 또는 물건의 특성을 기반으로 하는 추천 알고리즘이 아닌 제품의 신규성, 사용자 과거 클릭에 기반한 단순 알고리즘이 대부분이다[1, 2, 6]. 앞서 살펴본 해외의 사례와 같이 알고리즘 기반 추천 시스템이 경영성과에 큰 영향을 미칠 수 있다는 점에서 추천 시스템 관련 연구 및 도입이 절실한 실정이다.

이에 본 연구는 지금까지 개발된 다양한 추천 알고리즘에 대해서 설명하고, 이를 활용한 모델링 및 평가, 비교를 통해 효과적인 추천 알고리즘을 만드는 방법에 대해서 제언하고자 한다.

본 연구의 순서는 다음과 같다. 우선 제Ⅱ장에서 다양한 추천 시스템에 대해서 설명하고 추천 시스템의 연구 동향을 설명한 뒤 제Ⅲ장에서는 본 연구에서 모델링할 모델 방법들에 대해서 설명할 것이다. 그리고 제Ⅳ장에서 실제 무비렌즈(Movielense)의 영화 리뷰 데이터를 활용하여 세 가지 모델을 만들어 본 뒤 RMSE 값을 통해 모델을 비교하고 추천 시스템을 만들 때 고려해야 될 요소에 대해서 논할 계획이다. 마지막으로 5장에서 결론 및 토의를 통해 연구의 의미와 한계에 대해서 생각해볼 것이다.

Ⅱ. 문헌연구

2.1 추천 시스템

추천 시스템(Recommender Systems)은 다양한 정보와 제품들 속에서 사용자가 관심 가질 만한 제품을 추천해주는 시스템이다[8]. 추천 시스템의 종류는 추천을 하는 방식에 따라서 다양하게 구분할 수 있다. 사용자들 간의 평가 또는 구매정보를 기반으로 추천하는 사용자 기반 협력 필터링(User-based collaborative filtering), 물건 또는 콘텐츠와 관련된 정보를 기반으로 추천을 해주는 아이템 기반 필터링(Item-based collaborative filtering) 그리고 개인에 대한 지식을

기반으로 추천하는 Knowledge-based collaborative filtering이 있다. 마지막으로 이러한 다양한 추천 알고리즘을 함께 사용하는 하이브리드 추천 시스템(hybrid recommenders)이 있다[8].

이러한 추천 시스템들의 가장 기본적인 가정은 사용자들의 의견이 선택되고 통합되어서 실질적인 사용자의 선호를 기반으로 합리적인 예측을 할 수 있다는 것이다. 이러한 가정은 사용자들이 몇몇 물품에 대하여 품질 또는 선호에 대한 내용 간의 공통점이 있다는 것을 암암리에 전제하고 있다[14].

이와 같은 추천 문제에서 가장 중요한 문제는 제품에 대한 사용자의 평가를 담은 평가 행렬(Rating matrix)에서 비어 있는 요소(Element)를 예측하는 것이다[9]. 이를 위해 아이디어들이 제안되었다. 대표적으로 이웃 탐색 기법(Neighborhood Method)과 행렬 분해(Matrix factorization)이 있다. 이웃 탐색 기법의 경우 지역적인 구조(Local structure)를 파악하여 추천 문제를 해결하겠다는 아이디어다. 주로 유사도를 계산하여 추천을 하는 방식이다. 반면, 행렬 분해의 경우 평가 행렬의 전체적인 구조(Global structure)를 기반으로 추천을 하는 방식이다. 가장 대표적인 예시가 특이 값 분해(SVD) 알고리즘이 있다. 실제 넷플릭스의 고유 알고리즘이었던 Cinematch를 이기는데 가장 기본적인 아이디어가 바로 행렬 분해였다. 희박성(Sparsity)이 매우 높은 평점 행렬에서 유의미한 성분이 있는 행렬로 차원 축소시켜서 이를 기반으로 추천하는 아이디어다.

2.2 사용자 기반 협업필터링

사용자 기반의 협업 필터링은 K-NN 협업 필터링으로도 불린다. 이 방법은 현재 사용자와 유사한 제품을 구매하거나 유사한 평가를 매긴 사용자의 선호나 취향이 유사할 것이라는 가정을 하고 추천을 한다. 그러므로 사용자 간의 유사도를 계산하고 이를 바탕으로 현재 사용자의 평점을 예측한다. 일반적으로 사용자 기반 협업 필터링에서 피어슨 상관계수(Pearson correlation)를 통한 유사도가 가장 효과적이다[5]. 하지만 경우에 따라서 제한된 피어슨 상관계수(Contrainted Pearson correlation), 스피어만 상관계수(Spearman rank correlation), 코사인 유사도(Cosine similarity) 등을 사용하기도 한다.

사용자 협업 필터링을 활용한 대표적인 사례는 무비렌즈로도 유명한 그룹렌즈(GroupLense)[1], 음악 추천 시스템으로 유명한 링고(Ringo)[2]와 비디오 추천으로 잘 알려진 벨 코어(BellCore) 비디오[16] 등이 있다.

2.3 아이템 기반 협업필터링

이러한 추천 시스템들의 가장 기본적인 가정은 사용자들의 의견이 선택되고 통합되어서 실질적인 사용자의 선호를 기반으로 합리적인 예측을 할 수 있다는 것이다. 이러한 가정은 사용자들이 몇몇 물품에 대하여 품질 또는 선호에 대한 내용 간의 공통점이 있다는 것을 암암리에 전제하고 있다[14].

2.4 하이브리드 추천 시스템

앞서 살펴본 다양한 알고리즘은 각각의 장점과 단점이 존재한다. 이러한 한계를 하이브리드 추천 시스템을 통해 극복하고자 했다. 넷플릭스 프라이즈의 우승 팀의 경우 약 100여 개의 알고리즘을 앙상블 했다고 한다. 이처럼 하이브리드 방식은 추천 시스템의 성능을 보다 향상시키기 위해 사용하는 경우가 많다.

$$p_{u,i} = g_1(i)p_{u,i}^{(1)} + \cdots + g_n(i)p_{u,i}^{(n)} \quad (1)$$

1) https://grouplens.org/datasets/movielens/.

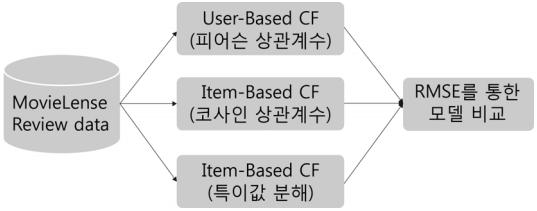
2) http://jolomo.net/ringo.html.

특히, 넷플릭스를 통해 제안된 feature-weighted linear stacking이 대표적인 예시다. 식 (1)은 사용자 u의 제품 I에 대한 평점을 여러 알고리즘 모델의 결과 값과 평점의 수 또는 장르와 같은 메타 변수의 선형 결합으로 표현한다[15].

Ⅲ. 연구 방법

본 연구는 추천 시스템에서 중요한 특징을 알아보기 위해 다양한 알고리즘에 기반하여 추천 시스템을 만들고 모델의 RMSE 값을 통해 정확도와 관련된 요소를 살펴보고자 한다. 연구 모델은 다음과 같다.

무비렌즈의 평가 데이터 10만 건을 활용하여 피어슨 상관계수를 활용한 사용자 기반 협업 필터링, 코사인 상관계수를 활용한 아이템 기반 협업 필터링 그리고 특이 값 분해를 활용한 아이템 기반 협업 필터링 모델을 만들 것이다. 만들어진 모델은 RMSE 값의 비교를 통해 평가를 하고 어떤 점이 이러한 차이를 만들었는지 논할 것이다.



〈그림 1〉 연구모형

3.1 UBCF-Pearson Correlation Model

평가행렬(Rating matrix) R에 대하여 $r_{u,i}$는 서비스 이용자 u의 영화 i에 대한 평점을 나타낸다. 이때, 예측행렬(Prediction matrix) P에 대하여 $p_{u,i}$는 서비스 이용자 u의 영화 i에 대한 예측 평점을 나타낸다.

$$p_{u,i} = \bar{r}_u + \frac{\sum_{u' \in N} s(u,u')(r_{u',i} - \bar{r}_{u'})}{\sum_{u' \in N} |s(u,u')|} \quad (2)$$

식 (2)는 $p_{u,i}$를 나타내는 모델이다. 사용자 u의 영화 i에 대한 값을 예측하기 위해 다른 사용자 u'의 평점 성향과 그 사람이 영화 i에 대하여 매긴 평점에 대한 편차를 활용한 유사도에 대한 가중 평균으로 계산된다. $S(u,u')$은 피어슨 상관계수를 나타낸다.

3.2 IBCF-Cosine Correlation Model

$$p_{u,i} = \frac{\sum_{j \in S} s(i,j)(r_{u,j} - b_{u,i})}{\sum_{j \in S} |s(i,j)} + b_{u,i} \quad (3)$$

식 (3)은 IBCF의 평가행렬 P을 나타내는 모델이다. 이 모델에 사용되는 $s(i,j)$값은 코사인 유사도를 나타낸다. 또한 $b_{u,i}$는 베이스라인(baseline) 예측 값으로써 다음과 같은 식 (4)를 통해 구할 수 있다.

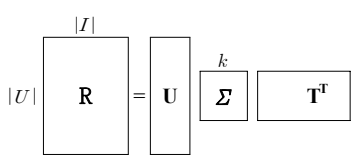
$$b_{u,i} = \mu + b_u + b_i \quad (4)$$

$$b_u = \frac{1}{|I_u| + \beta_u} \sum_{i \in I_u} (r_{u,i} - \mu) \quad (5)$$

$$b_i = \frac{1}{|U_i| + \beta_u} \sum_{i \in I_u} (r_{u,i} - b_u - \mu)$$

식 (5)의 경우 각각 사용자 기준과 영화 기준으로 베이스라인 예측 값을 계산한다. 특히 위 식의 베타계수는 감쇠계수(Damping coefficient)로서 평점의 영향력이 시간에 따라 약해지는 정도를 반영해준다. Funk[12]는 감쇠 계수의 값이 25일 때, 베이스라인 값이 가장 유용함을 발견했다. 그러므로 본 연구에서도 감쇠 계수가 25인 베이스라인 예측값을 사용하겠다.

3.3 IBCF-SVD Model



〈그림 2〉 SVD 모델

U는 사용자 k개의 주제에 대한 선호를 나타내는 행렬이고 Σ의 경우 특이 값 행렬로써 각 선호에 대한 가중치를 담은 행렬이다. 이 두 행렬의 요소 값을 활용하여 다음과 같은 예측행렬 P의 예측 값을 계산한다.

$$p_{u,i} = \sum_{f} u_f \sigma_f i_f \qquad (6)$$

특이 값 분해와 같은 행렬 분해의 경우 전체 평가 행렬의 구조적인 성격을 반영하기 때문에 추천 시스템에 적용 할 경우 예측력을 향상시킬 수 있다.

Ⅳ. 분석 결과

4.1 데이터 수집

데이터는 movielense에서 제공하는 데이터를 활용하였다. 데이터는 평가 정보, 영화의 장르와 관련된 정보, 서비스 이용자의 인구통계학적인 정보가 있다. 평가 정보의 경우 서비스 이용자 943명이 시청한 1682개의 영화에 대한 평점 정보와 로그 데이터가 있다. 데이터의 크기는 10만 건이다. 영화 정보의 경우 1682개의 영화가 어떤 장르에 속하는지 장르가 태깅(Tagging)되어있다. 마지막으로 사용자 정보는 943명에 대하여 나이, 성별, 직업, 사는 지역과 관련된 정보가 있다.

본 연구는 UBCF와 IBCF 모델을 만들 것이기 때문에 사용자의 평점 정보와 영화 콘텐츠와 관련된 데이터를 기반으로 모델링 할 것이다.

4.2 UBCF-Pearson Correlation Model 결과

우선 피어슨 상관계수를 활용한 사용자 기반 협업 필터링을 위해 평가행렬을 활용하여 상관계수 행렬을 만들었다. 모든 값에 대한 적절한 예측을 위해 상관계수를 구할 수 없는 쌍은 상관계수 행렬의 전체 평균으로 대체시켰다.

	A	B	C	D	E	F	G	H	I	J
1	1	2	3	4	5	6	7	8	9	10
2	0	0.160841	0.11278	0.5	0.420809	0.287159	0.258137	0.692086	-0.10206	-0.09234
3	0.160841	0	0.06742	0.148522	0.327327	0.446269	0.643675	0.585491	0.242536	0.668145
4	0.11278	0.06742	0	-0.2626	0.131549	-0.10911	0.064803	0.291937	0.131549	0.311086
5	0.5	0.148522	-0.2626	0	1	-0.58132	-0.26663	0.642938	0.131549	-0.30151
6	0.420809	0.327327	0.131549	1	0	0.241817	0.17563	0.5374	0.57735	0.087343
7	0.287159	0.446269	-0.10911	-0.58132	0.241817	0	0.237555	0.687745	0.132353	0.273987
8	0.258137	0.643675	0.064803	-0.26663	0.17563	0.237555	0	0.327028	-0.22564	0.299751
9	0.692086	0.585491	0.291937	0.642938	0.5374	0.687745	0.327028	0	0	0.455825
10	-0.10206	0.242536	0.131549	0.131549	0.57735	0.132353	-0.22564	0	0	0.329956

〈그림 3〉 상관계수 행렬(일부)

이를 활용해 UBCF-Pearson correlation Model을 활용하여 예측 행렬 P를 얻을 수 있었다.

	A	B	C	D	E	F	G	H	I	J
1	1	2	3	4	5	6	7	8	9	10
2	2.421194	1.392311	1.229559	1.642804	1.220198	1.053635	2.260511	1.595595	1.974272	1.248907
3	2.288362	1.396208	1.257519	1.721193	1.292463	1.155474	2.289814	1.784324	1.950446	1.265138
4	2.294681	1.789816	1.6652	1.824147	1.68892	1.591864	2.198598	1.927348	2.051644	1.612984
5	3.752052	3.092202	3.099797	3.195092	3.145662	3.085783	3.666048	3.246894	3.390061	3.226461
6	1.795142	0.691906	0.597284	0.953349	0.572691	0.464718	1.709219	0.896948	1.259294	0.668907
7	2.344984	1.421905	1.322588	1.748302	1.3256	1.191471	2.25664	1.690614	2.080486	1.392382
8	3.042341	2.127466	1.97293	2.403489	1.995785	1.794344	2.959136	2.414701	2.733631	1.943138
9	2.634857	1.577423	1.526848	1.836867	1.51892	1.411821	2.331283	1.87921	2.229963	1.572298
10	3.66818	3.015215	2.928979	3.137458	2.921592	2.831113	3.355819	3.114863	3.190191	2.866697

〈그림 4〉 예측 행렬 P(일부)

모델의 평가를 위해 RMSE를 계산한 결과 2.323241가 나왔다.

4.3 IBCF-Cosine Correlation Model 결과

다음은 코사인 상관계수를 활용한 아이템 기반 협업 필터링이다. 이를 위해 영화와 관련된 행렬을 활용하여 코사인 유사도를 통해 다음과 같은 코사인 유사도 행렬을 만들었다.

	A	B	C	D	E	F	G	H	I	J
1	1	2	3	4	5	6	7	8	9	10
2		0.160841	0.11278	0.5	0.420809	0.287159	0.258137	0.692086	-0.10206	-0.09234
3	0.160841		0.06742	0.148522	0.327327	0.446269	0.643675	0.585491	0.242536	0.668145
4	0.11278	0.06742		-0.2626	0.131549	-0.10911	0.064803	0.291937	0.131549	0.311086
5	0.5	0.148522	-0.2626		1	-0.58132	-0.26663	0.642938	0.131549	-0.30151
6	0.420809	0.327327	0.131549	1		0.241817	0.17563	0.5374	0.57735	0.087343
7	0.287159	0.446269	-0.10911	-0.58132	0.241817		0.237555	0.687745	0.132353	0.273987
8	0.258137	0.643675	0.064803	-0.26663	0.17563	0.237555		0.327028	-0.22564	0.299751
9	0.692086	0.585491	0.291937	0.642938	0.5374	0.687745	0.327028		0	0.455825
10	-0.10206	0.242536	0.131549	0.131549	0.57735	0.132353	-0.22564	0		0.329956
11	-0.09234	0.668145	0.311086	-0.30151	0.087343	0.273987	0.299751	0.455825	0.329956	

〈그림 5〉 코사인 유사도 행렬(일부)

이를 활용해 IBCF-Cosine correlation Model을 활용하여 다음과 같은 예측 행렬 P를 얻을 수 있다.

	A	B	C	D	E	F	G	H	I
1	3.905131	3.389732	3.280046	3.590498	3.446954	3.66522	3.842306	3.959616	3.910358
2	3.959613	3.444214	3.334528	3.64498	3.501436	3.719702	3.896788	4.014099	3.96484
3	3.330044	2.814645	2.704959	3.015411	2.871867	3.090133	3.267219	3.38453	3.335271
4	4.225005	3.709607	3.59992	3.910373	3.766828	3.985095	4.16218	4.279491	4.230232
5	3.257839	2.742441	2.632754	2.943207	2.799663	3.017929	3.195015	3.312325	3.263067
6	3.925533	3.410134	3.300448	3.610901	3.467356	3.685622	3.862708	3.980019	3.93076
7	4.241435	3.726037	3.61635	3.926803	3.783258	4.001525	4.178611	4.295921	4.246662
8	4.018827	3.503429	3.393742	3.704195	3.56065	3.778917	3.956003	4.073313	4.024054
9	4.179192	3.663794	3.554107	3.86456	3.721015	3.939282	4.116368	4.233678	4.184419
10	4.427188	3.91179	3.802103	4.112556	3.969011	4.187278	4.364364	4.481674	4.432415

〈그림 6〉 예측 행렬 P(일부)

위의 예측 행렬의 평가를 위해 RMSE를 계산한 결과 0.9447755를 얻었다.

4.4 IBCF-SVD Model 결과

마지막으로 특이 값 분해를 활용한 아이템 기반 협업 필터링 모델을 만들었다. 특이 값 분해의 경우 오픈소스 R을 활용하여 계산하였다. R의 recommenderlab을 활용하여 특이 값 분해를 하였고 이를 통한 예측 행렬을 구하였다.

	ITEM-1	ITEM-2	ITEM-3	ITEM-4
1	3.817771	3.129018	3.008694	3.914779
2	3.913732	3.36048	3.082873	3.700735
3	3.090347	2.590224	2.64935	3.037521
4	4.665214	4.082354	3.658694	4.24239
5	3.325304	2.594609	2.751653	3.42165
6	3.427377	2.865054	2.748976	3.550167
7	4.320995	3.492085	3.150271	4.258281
8	4.101341	3.434953	3.068788	4.017533
9	4.202311	3.744177	3.286454	3.847792
10	4.347687	3.626878	3.447632	3.897778
11	3.777385	3.177145	3.176891	3.280763
12	4.447174	3.969008	3.558052	3.832777
13	3.14772	3.339321	2.133467	4.14918
14	3.96576	3.580653	3.271983	3.775615
15	3.305546	3.037011	2.571454	3.070255
16	4.360105	3.757624	3.339291	4.306003
17	3.173078	2.590664	2.463449	3.219207

〈그림 7〉 예측 행렬 P(일부)

위 모델의 평가를 위해 RMSE를 계산한 결과 0.8199582를 얻었다. 앞서 살펴본 RMSE들에 비해 차이가 많이 남을 확인할 수 있었다.

4.5 세 가지 모델 성능 평가

앞서 살펴본 모델의 RMSE 값을 표 형태로 정리해 보았다.

〈표 1〉 RMSE 비교

추천 알고리즘	RMSE
UBCF-피어슨	2.323241
IBCF-코사인	0.9447755
IBCF-SVD	0.8199582

RMSE는 오차의 양을 나타냄으로 낮을수록 예측의 정확도가 높다는 것을 의미한다. UBCF-Pearson correlation Model의 RMSE값이 가장 큰 것을 확인할 수 있다. 즉, 본 연구의 데이터에서는 사용자 기반 협업 필터링 방식의 추천보다 아이템 기반 추천방식의 성능이 더 높은 것을 확인 할 수 있다. 특히 Neighborhood Method인 코사인보다 Matrix factorization의 일종인 특이 값 분해를 적용한 모델의 RMSE가 낮은 것을 확인할 수 있다. 이는 지엽적인 구조보다 전체적인 구조를 통해 예측할 경우 예측의 정확도가 월등히 높아짐을 나타낸다. 나아가 하이브리드 추천 시스템을 활용하여 지금 살펴본 알고리즘을 앙상블하면 더 개선된 RMSE 값을 얻을 수 있을 것으로 기대된다.

V. 결 론

본 연구는 영화 추천 시스템에서 어떤 알고리즘이 더 효과적인지 실증적으로 살펴보았다는 점에서 의의가 있다. 본 연구의 데이터에서는 세 가지 모델 중 IBCF-SVD 모델의 성능이 가장 우수하였다. UBCF의 경우 방대한 사용자 정보를 필요로 하기 때문에 서비스 초기에는 구현하기 어렵다는 단점이 있지만 IBCF의 경우 상대적으로 적은 데이터를 활용해 추천 시스템을 구현할 수 있다는 장점이 있다. 또한 아직 평가되지 않은 제품에 대한 추천도 가능하다는 이점도 갖는다. 본 연구의 결과처럼 IBCF가 경우에 따라서 더 높은 성능을 발휘할 수도 있다

기업들은 추천 시스템을 개선하기 위해 아이

템 정보와 사용자정보를 모두 활용하는 하이브리드 방식을 도입하고 있지만, 대다수의 기업이 방대한 정보를 갖고 서비스를 시작하기는 어렵다. 따라서 서비스의 초기에는 아이템기반 추천 시스템을 통해 추천하되 이후 사용자 데이터를 수집하여 하이브리드 방식을 구현할 수 있는 토대를 만든다면 보다 나은 추천 시스템을 단계적으로 만들 수 있을 것이라 생각된다. 또한 본 연구는 추천 시스템을 구현할 때, 특이 값 분해를 활용하는 것이 추천 시스템의 성능개선에 유효하게 작동한다는 것을 확인했다는 데에 의의를 갖는다.

참 고 문 헌

[1] 안현철, 한인구, 김경재, "연관규칙기법과 분류모형을 결합한 상품 추천 시스템: G 인터넷 쇼핑몰의 사례", Information Systems Review, 제8권, 제1호, pp.181-201, 2006.

[2] 이재식, 박석두, "장르별 협업 필터링을 이용한 영화추천 시스템의 성능 향상", 한국지능정보시스템학회논문지, 제13권, 제4호, pp.65-78, 2007.

[3] Bell, R.M., Y. Koren, and C. Volinsky, *The BellKor solution to the Netflix prize*, 2007.

[4] Bennett, J. and L. Stan, "The netflix prize", Proceedings of KDD cup and workshop. 2007.

[5] Breese, J.S., D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering", In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence* (pp.43-52). Morgan Kaufmann Publishers Inc., 1998.

[6] Cho, Y.S. and R.K. Ho, "Personalized Recommendation System using FP-tree Mining based on RFM", *Journal of the Korea Society of Computer and Information*, Vol.17, No.2, pp.197-206, 2012.

[7] Davidson, J., B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, and D. Sampath, "The YouTube video recommendation system", In *Proceedings of the fourth ACM Conference on Recommender Systems* (pp.293-296), ACM, 2010.

[8] Ekstrand, M.D., J.T. Riedl, and J.A. Konstan, "Collaborative filtering recommender systems", *Foundations and Trends in Human-Computer Interaction*, Vol.4, No.2, pp.81-173, 2011.

[9] Koren, Y., R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems", *Computer*, Vol.42, No.8, pp.30-37, 2009.

[10] Linden, G., B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering", *IEEE Internet Computing*, Vol.7, No.1, pp.76-80, 2003.

[11] Resnick et al., "GroupLens: an open architecture for collaborative filtering of netnews", *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, ACM, 1994.

[12] Funk, S., "Netflix update: Try this at home", http://sifter.org/~simon/journal/20061211.html, Archived by WebCite at http://www.webcitation.org/5pVQphxrD, December 2006.

[13] Sarwar, B., G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms", In *Proceedings of the 10th international conference on World Wide Web* (pp.285-295). ACM, 2001.

[14] Schafer, J.B., D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems", In The adaptive web (pp.291-324). Springer Berlin Heidelberg, 2007.

[15] Sill, J., G. Takács, L. Mackey, and D. Lin, Feature-weighted linear stacking. arXiv preprint arXiv:

0911.0460, 2009.

[16] Shardanand, U. and P. Maes, “Social information filtering: Algorithms for automating “word of mouth”, in ACM CHI ’95, pp.210-217, ACM Press/Addison-Wesley Publishing Co., 1995.

[17] Hill, W., L. Stead, M. Rosenstein, and G. Furnas, “Recommending and evaluating choices in a virtual community of use”, in ACM CHI ’95, pp.194-201, ACM Press/Addison-Wesley Publishing Co., 1995.

저 자 소 개

김 동 욱(Dong-Wook Kim)

- 2012년~현재 : 아주대학교 경영대학 e-비즈니스학과 재학
- 관심분야 : 빅데이터 분석, 텍스트마이닝, 머신러닝

김 성 근(Sung-Geun Kim)

- 2009년 : 가톨릭대학교 심리학과 (문학사)
- 2016년~현재 : 아주대학교 경영정보학과 석사과정 재학 중
- 관심분야 : 인터넷 마케팅, 텍스트마이닝, 머신러닝

강 주 영(Juyoung Kang)

- 1995년 2월 : 포항공과대학교 컴퓨터공학과 (공학사)
- 1997년 : 서울대학교 컴퓨터공학 (석사)
- 2005년 : 한국과학기술원 경영공학 (박사)
- 2005년~현재 : 아주대학교 경영대학 e-비즈니스학과 교수
- 관심분야 : 텍스트 마이닝, 빅데이터 분석, 클라우드 컴퓨팅, 지능형 전자상거래