R에서 협업 필터링과 개인화 요인을 이용한 개인화 영화 추천 시스템

심대수*, 김철환*,박진수**, 박두순*
*순천향대학교 컴퓨터소프트웨어공학과
**순천향대학교 웰니스코칭 서비스연구센터
e-mail : tlaeotn123@naver.com

A Personalized Movie Recommendation System using Collaborative Filtering and Personal Propensity in R

Dae-Soo Sim*, Chul-Hwan Kim*, Jin-Soo Park**,
Doo-Soon Park*
*Dept. of Computer Software Engineering, SoonchungHyang University
**Wellness Coaching Service Research Center(C-ITRC)

요 약

인터넷의 보급과 동시에 데이터의 누적으로 생성된 수많은 빅 데이터의 활용을 통해 수 없이 많은 개인에 대한 분석과 추천이 가능해졌다. 그중 영화는 현대인의 문화로 자리 잡으며 수많은 데이터의 누적이 이루어 졌으며 계속해서 누적되어가고 있다. 이런 누적된 데이터를 통해서 개인에게 맞는 영화를 추천하는 협업필터링 시스템을 R을 통해 분석하고 Cold Start 문제를 개인화 요인으로서 보안하여 보다 신뢰성 높은 추천 시스템을 제안 한다.

1. 서론

정보화 시대에 데이터 누적은 자연스러운 일이다. 이러한 데이터의 누적으로 인해 흔히 빅 데이터라고 불리는 거대한 데이터 속 사용자가 원하는 정보를 찾기란 여간 쉬운 일이 아니다. 그 중 현대의 문화로 자리 잡은 영화의 데이터 또한 빅 데이터로 누적되었으며, 영화는 다른 콘텐츠들에 비해 지금까지 누적된 영화 콘텐츠뿐만 아닌 앞으로 누적될 콘텐츠 또한 무궁무진하다. 이[1]러한 영화의 장르와 종류가 날이 갈수록 다양해지고 있으며 영화시장의 성장률은 끝없이 증가한다. 2006년부터 2015년까지의 영화시장의 성장률은 (표 1)과 같다.

(표 1) 2006 - 2015 미국의 콘텐츠 시장 규모 변화[1]

(단위: 백만 달러)

구분	2006	2007	2008	2009	2010p	2011	2012	2013	2014	2015	2011-2015 CAGR(%)
영화	35,318	36,057	35,207	35,015	35,300	36,842	38,853	41,151	43,448	45,686	5.4
애니메이션[a]	5,737	5,664	4,857	5,532	6,621	6,116	6,443	6,751	7,114	7,476	2.5
방송	155,008	158,842	160,865	154,329	164,450	170,255	183,985	189,794	203,081	210,848	4.8
게임	9,342	11,823	14,720	13,746	13,607	14,135	14,631	15,116	15,810	17,014	5.3
음악	11,728	10,615	8,667	7,524	6,599	6,432	6,342	6,323	6,373	6,476	-0.4
출판	113,274	112,061	102,385	88,808	87,238	87,269	88,144	89,660	91,440	93,614	1.4
만화	695	700	683	685	635	646	651	662	671	665	0.9
광고	199,195	198,957	187,878	160,848	169,537	173,083	185,131	189,892	201,676	207,851	4.2
지식정보	112,163	118,790	117,744	110,178	113,527	118,783	125,282	132,422	140,404	149,113	6.0
캐릭터[b]	63,570	63,200	94,040	83,150	83,070	82,900	82,650	82,490	82,320	82,980	0.0
전체	455,520	469,713	461,737	429,912	443,144	458,842	484,835	503,901	531,718	555,277	4.6

"본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음" (IITP-2017-2014-0-00720-002)

이처럼 빅 데이터로 남아버린 영화와, 계속해서 생성되는 영화들 속에서 사용자는 자신의 요구사항에 맞는 영화를 찾기란 쉬운 일이 아니다. 그에 따라 '왓챠'(www.watcha.net), '로튼 토마토'(www.rottentomatoes.com)와 같이 많은 영화추천 시스템이 개발되어 왔으며 '넥플릭스'처럼 영화추천을 통해 떠오른 서비스들도 있다. 이처럼 빅 데이터가 일상이 된 현대 사회에서는 데이터 분석이 필수적이며 피할수 없는 과제가 되었다. 또한 이에 따른 많은 알고리즘과 끝없는 연구가 진행되고 있다.

본 논문에서는 R의 recommenderlab 라이브러리를 이용하여 데이터를 분석하며 데이터 MovieLens에서 배포하는 100,000명 사용자들의 평가 데이터를 이용해 추천 시스템을 개발한다. 또한, 학습데이터와, 테스트데이터로 데이터를 분할하여 추천시스템의 신뢰성을 MAE를 통해 측정한다. 그리고 데이터 희박성으로 발생하는 Cold Start 문제를MovieLens에서 제공하는 사용자의 관심 장르를 이용하여 신뢰성을 높여보고자 한다.

2. 영화 추천 시스템의 구성

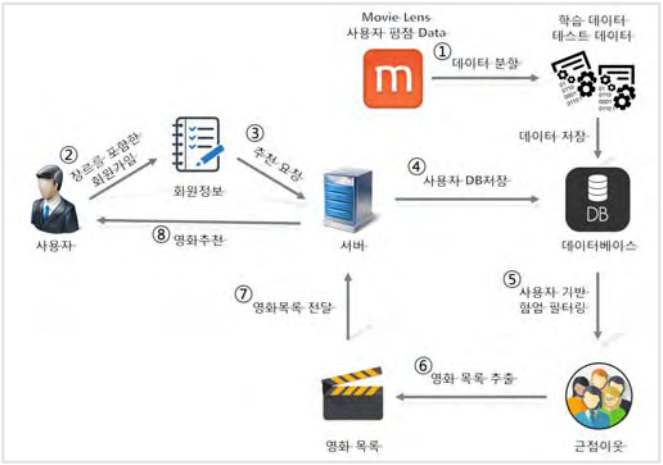
협업 필터링 기반 상품 추천 과정은 크게 입력 데이터 구성, 이웃집단 탐색, 추천 상품 결정 단계로 나눠볼 수 있으며, 이러한 과정을 자세히 살펴보면 다음과 같다.

(1) 입력 데이터 구성(Data Representation): 협업필터링 기반 상품추천시스템에서의 입력 데이터는 보통 n개의 상품에 대하여 m명 고객의 구매 트랜잭션의 집합으로 구성되며, 보통 n x m의 고객-상품 행렬 R로 표현될 수 있다.

(2) 이웃 집단 탐색(Neighborhood Formation): 고객간의 유사도를 계산하여 이웃 집단을 탐색하는 과정이다. 두 고객의 유사도를 측정하는 방법으로써 코사인(Cosine)을 이용한다.

(3) 추천 상품 결정(Generation of Recommendation): 상품 추천을 위한 마지막 단계로서 설정된 이웃 집단으로부터 상위 N개의 추천 상품 목록을 이끌어 내는 단계이다.[2].

따라서 본 논문에서 구현한 추천 시스템의 시나리오는 (그림1)과 같다.



(그림 1) 추천 시스템 시나리오

(그림 1)의 추천 시나리오는 다음과 같다.

① Movie Lens의 사용자 평점 데이터를 이용하기 학습데이터와, 검증데이터로 나눈 후, 서버 DB에 저장한다.

② 사용자의 평점 데이터와, 개인화 요소를 저장 및 사용하기 위해서 회원가입을 하지 않은 사용자의 경우 회원가입을 거친다.

③ 회원가입이 된 회원의 경우 영화의 추천을 요구한다.

④ 회원가입을 마친 사용자의 데이터를 서버 데이터베이스에 저장한다.

⑤ DB에 누적되어있는 사용자의 영화 평점 데이터가 충분할 시 협업 필터링을 이용하여 최 근접이웃을 구성한다. 여기서 협업필터링이란, 사용자들의 선호도와 관심 표현을 바탕으로 선호도, 관심도가 비슷한 사용자들을 식별해 내는 방법으로 과거에 이용한 콘텐츠가 비슷하다면 사용자 간에 유사한 성향을 가지고 있다고 판단하고 그 근거를 토대로 추천하는 방식이다[3]. 협업필터링을 통한 추천 방식에는 크게 사용자 기반(User-based)과, 아이템 기반(Item-based)방식이 있으나, 본 논문에서는 사용자 기반(User-based) 협업필터링을 사용하였다. 최 근접이웃을 구성하기 위해 사용자의 유사도를 측정하는 방식으로는 가장 널리 알려진 방법인 (그림 2) 코사인 유사도(Cosine Similarity)를 이용하여 두 사용자(A,B) 간의 유사도를 측정한다.

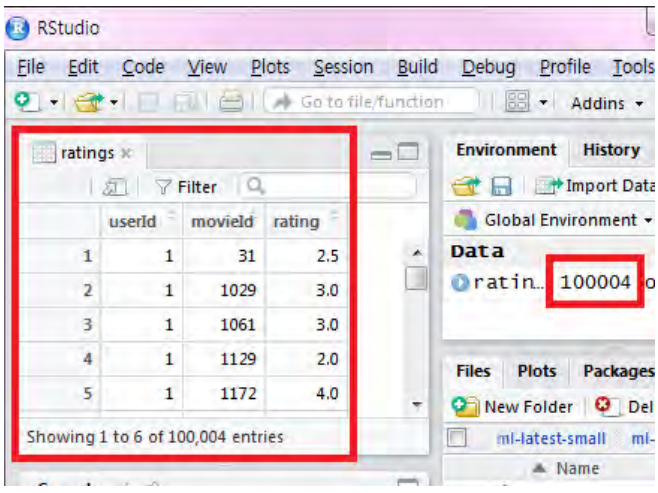
$$sim(A,B) = \frac{\sum_{i \in I_{AB}} R_{A,i} \; R_{B,i}}{\sqrt{\sum_{i \in I_{AB}} R_{A,i}^{\;2}} \sqrt{\sum_{i \in I_{AB}} R_{B,i}^{\;2}}}$$

(그림 2) Cosine Similarity

여기서 R은 n x m의 고객-상품 행렬이며, I_{AB} 는 A 사용자, B사용자가 공통으로 평가한 행렬, $R_{A,i}$와 $R_{B,i}$는 사용자 A와 B가 공통으로 평가한 I행렬의 각 아이템 i의 평가치를 뜻한다.[4]. 즉 A,B가 공통으로 평가한 두 아이템평가행렬 벡터의 사이각을 구하여 최소가 되는 최 근접이웃을 추출한다.

⑥ 최 근접이웃이 추천한 영화목록을 추출하여 사용자에게 전달하여 영화 목록을 추천한다.

또한 추천 시스템을 만들기 전 Movie Lens 의 100,000명 사용자 데이터를 기반으로 학습데이터(60,000), 테스트데이터(40,000)로 나누어 실제적인 추천시스템의 신뢰도를 측정해 본다. Movie Lens의 사용자 데이터는 무료로 배포되고 있으며 받은 데이터 자료는 (그림 3)과 같다.



(그림 3) Movie Lens의 사용자 평점 데이터

(그림 3)의 데이터를 이용하여 사용자의 예측 평가치와, 테스트데이터의 평가치를 비교하여 성능을 측정하는데 이때 측정 지표로는 (그림 4)와 같이 MAE(Mean Absolute Error)를 사용한다.

$$MAE = \frac{\sum_{i=1}^{q} |\text{실제고객평가치 } i - \text{예측된 평가치 } i|}{q}$$

(그림 4) Mean Absolute Error

q는 사용자가 평가한 아이템의 개수이며, 실제 고객 평가치i는 고객이 평가한 i번째 아이템의 평가치를 의미한다. 같은 의미로 예측된 평가치i는 사용자를 제외한 근접이웃의 평가치의 평균으로서 사용자의 평가치를 예측한 것이다.

따라서 MAE는 실제 값과, 예측 평가치의 오차의 합을 q로 나눈 예측치 오차의 평균으로 예측한 평가치와 실제 고객의 평가치의 오차를 나타내는 지표이다.

3. 영화 추천 시스템의 구현

본 논문에서 구현한 영화 추천 시스템은 기본적으로 R Studio에서 지원하는 'recommenderlab' 라이브러리를 이용한다. 따라서 패키지의 설치가 필요한데 R Studio의 Consol창에서 (그림 5)와 같은 명령어로 패키지를 인스톨한다.

```
Console ~/
'help.start()' for an HTML browser interface to help
.
Type 'q()' to quit R.

> install.packages("recommenderlab")
Installing package into 'C:/Users/SDS/Documents/R/wi
n-library/3.4'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/con
trib/3.4/recommenderlab_0.2-2.zip'
Content type 'application/zip' length 1428312 bytes
```

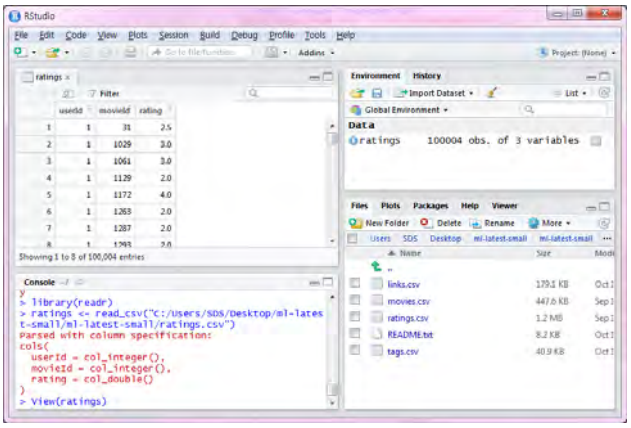
(그림 5) R Studio 개발환경 설정

기본적으로 'recommederlab' 라이브러리에서 제공되는 함수들을 이용하여 최 근접이웃 구성 및 추천 시스템을 개발하게 된다. 또한 누적된 Data 사용을 위해 전에 Movie Lens를 통해서 받아두었던 자료들을 Import해야 하는데 (그림 6)과 같은 방법을 통해 Movie Lens의 사용자 Data들을 명령어로 Import 한다.

```
Console ~/
y
> library(readr)
> ratings <- read_csv("C:/Users/SDS/Desktop/ml-lates
t-small/ml-latest-small/ratings.csv")
Parsed with column specification:
cols(
  userId = col_integer(),
  movieId = col_integer(),
  rating = col_double()
)
> View(ratings)
```

(그림 6) Movie Lens Data Import

Data를 Import 하게 되면R Studio에서 사용자의 편의를 위해 (그림 7) 과 같이 새로운 테이블이 생성되며 읽어들인 Data를 볼 수 있게 된다.



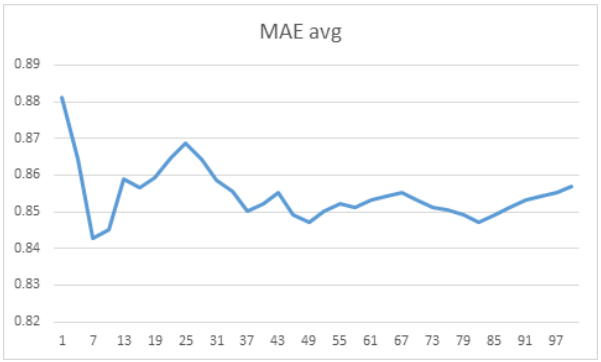
(그림 7) 읽어 들인 Data과 R Studio View

또한 학습데이터와 테스트데이터로 나누기 위해서 (그림 8)과 같은 명령어로 데이터를 분할한다.

```
Console ~/
t-small/ml-latest-small/ratings.csv")
Parsed with column specification:
cols(
  userId = col_integer(),
  movieId = col_integer(),
  rating = col_double()
)
> View(ratings)
> trainingData<- sample(100000,60000)
>
Values
  trainingDa… int [1:60000] 41924 14140 86…
```

(그림 8) Training Data를 샘플링 하는 과정

그 뒤 테스트 데이터에 대해서 최 근접 이웃을 구성하며 최 근접이웃의 수인 N의 값이 변동될 때마다 MAE(Mean Absolute Error)의 값이 변동됨을 알 수 있었다. 따라서 MAE의 값이 최소가 되는 최 근접이웃의 수 N은 높은 신뢰성을 위해서 중요하며 따라서 최적의 최 근접이웃의 수인 N을 찾기 위해 N값을 1부터 3씩 최대 100까지 증가시키며 측정한 결과는 (그림 9)와 같다.



(그림 9) 근접이웃 N에 따른 MAE의 변화

위와 같은 결과로서 최적으로 생각되는 최 근접이웃의 N은 7로 두어서 최 근접 이웃을 구성하였다.

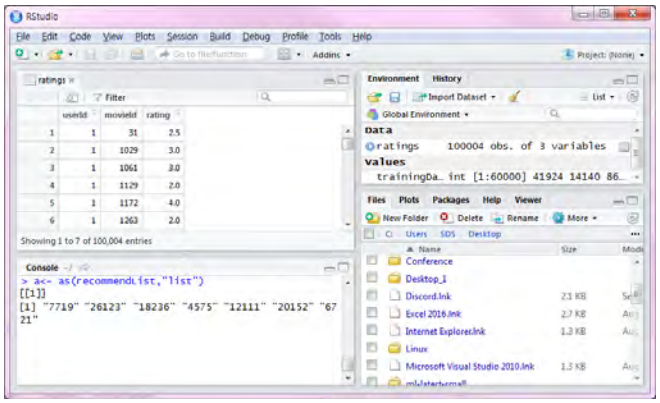
또한 사용자의 데이터 부족 시 Cold Start 문제를 보안하기 위해서 기존에는 다수의 사용자에게 평가가 좋은 영화를 모두에게 추천하였지만, 추가적으로 다수의 사용자에게 평가가 좋으며, 개인화 요소인 사용자가 선호하는 장르를 통해 추천한 테스트데이터 40,000건의 MAE의 평균 비교는 (표 2)와 같다.

(표 2) MAE avg 비교 데이터

Non-Genre MAE avg	Genre MAE avg
1.4132	1.1523

이전의 평범한 방식보다 MAE가 0.16만큼 줄어든 것을 볼 수 있다. 즉 장르를 추가한 영화추천이 더 신뢰성이 높음을 알 수 있다.

(그림 10)는 임의의 사용자에 대한 최 근접 이웃(N = 7)의 추천 목록 결과이다.



(그림 10) 최 근접이웃 (N=7) 로부터 추출된 영화목록

위처럼 현재는 Movie Id를 통하여 콘솔에 출력되지만 후에 추가적으로 Movie Lens의 Data가 아닌 직접 데이터를 누적하고 관리하게 될 경우 Naver의 영화목록과 같은 정보를 크롤링(Crawling)하여 Id와는 별개로 영화의 정보를 DB로 구축하여 연동할 수 있다. 현재 Movie Lens에서도 Movie ID에 해당하는 영화의 제목이 있지만 부가적인 영화에 대한 정보는 가지고 있지 않기 때문에 나중에 좋은 추천시스템을 개발하기 위해서는 DB구축을 잘 해두어야 할 것이다.

최종적으로 Movie Lens의 movie id 와 movie title을 조합해보면 사용자에게 추천된 영화 목록은 (그림 11)과 같다.

userId	movieId	rating
1	31	2.5
1	1029	3
1	1061	3
1	1129	2
1	1172	4
1	1263	2
1	1287	2
1	1293	2
1	1339	3.5

movieId	title	genres
7719	Four Musketeers, The (1974)	Action|Adventure|Comedy|Romance
26122	Onibaba (1964)	Drama|Horror|War
18236	House of Flying Daggers (Shi mian mai fu)	Action|Drama|Romance
4575	Police Story (Ging chaat goo si) (1985)	Action|Comedy|Crime|Thriller
12111	Quigley Down Under (1990)	Adventure|Drama|Western

(그림 11) User1의 정보와 사용자에게 추천된 영화 목록.

userId	movieId	rating
2	10	4
2	17	5
2	39	5
2	47	4
2	50	4
2	52	3
2	62	3
2	110	4
2	144	3

movieId	title	genre
2171	Next Stop Wonderland (1998)	Comedy|Drama|Romance
8951	Vera Drake (2004)	Drama
73017	Sherlock Holmes (2009)	Action|Crime|Mystery|Thriller
34334	Stealth (2005)	Action|Adventure|Sci-Fi|Thriller
27773	Old Boy (2003)	Mystery|Thriller

(그림 12) User2의 정보와 사용자에게 추천된 영화 목록.

4. 결론

본 논문에서는 데이터의 누적으로 인한 수많은 영화 데이터 사이에서 사용자에게 적합한 영화를 추천하기 위해서 Movie Lens의 사용자 Data를 이용해 사용자 기반(User-based) 협업 필터링을 이용해서 최 근접 이웃을 구성하며, 최 근접 이웃의 수 N에 따라 신뢰도가 변경되는 것을 MAE(Mean Absolute Error)로 측정한 후 최적이 되는 N값을 찾아 최 근접 이웃의 영화를 추천하는 R Studio를 이용해 데이터를 분석, 추천하는 시스템을 만들어 보았다. 또한 추천 시스템에서 항상 일어나는 데이터 희박성 문제 중 Cold Start 문제를 해결하기 위해 데이터의 부족 시 사용자의 선호 장르를 추가하여 추천 시 더욱 신뢰성이 높아짐을 알 수 있었다. 추가적으로 Movie Lens의 Data가 아닌 직접 Data를 누적하기 위해서 DB구축을 잘 해야 할 것이며, 여전히 문제가 되고 있는 여러 추천시스템의 문제인 데이터 희박성(Sparsity), 확장성(Scalability), 투명성(Transparency)을 보안하기 위해 더욱 연구가 진행되어야 할 것이다.

참고문헌

[1] 해외 콘텐츠시장조사(미국), 한국콘텐츠진흥원, 2012.
[2] 김재경, “개인별 상품추천시스템, WebCF-PT: 웹마이닝과 상품계층도를 이용한 협업필터링”, 경영 정보학 연구 논문지, pp. 63-78, 2004.11.
[3] 김영아, 박두순, “협업 필터링 기반 드라마 추천 시스템”, 한국정보처리학회 추계학술대회 발표 논문집, 제주 한라대학교, pp. 1137-1138, 2013.11
[4] 이재식, “장르별 협업필터링을 이용한 영화추천 시스템의 성능 향상”, 한국 지능정보 시스템 학회 논문지, pp. 66-78, 2007.12.