소셜 데이터를 이용한 협업필터링 추천 시스템 성능 개선 연구

주종민*, 양형정‡ , 김남훈*, 박성현**, 이건우**
*전남대학교 전자컴퓨터공학대학원
**전남대학교 전자컴퓨터공학부
e-mail:168798@jnu.ac.kr

A Study on improvement of performance of collaborative filtering recommendation system using social data

Jong-Min Joo*, · Hyung-Jeong Yang‡ · Nam-Hun Kim*, · Sung-Hyun Park** · Gun-Woo Lee**
*Dept. of Computer Science, Chonnam National University
**Dept. of Electronics and Communication Engineering, Chonnam National University

요 약

다양한 소셜 네트워크 서비스가 발달되고 많은 사람들이 소셜 미디어에 참여하면서 방대한 양의 정보가 발생하고 있다. 따라서 원하는 정보를 선별하고 가공하는 연구도 활발히 진행되고 있다. 협업필터링은 이러한 정보를 토대로 사용자에게 맞춤형 아이템을 추천해주는 알고리즘이다. 하지만 정확한 추천을 위해서는 매우 방대한 양의 정보가 필요하다. 또한 협업필터링에는 초기에는 제대로 추천이 이루어지지 않는 콜드스타터 문제가 있다. 이러한 문제를 해결하기 위해 본 논문에서는 소셜 네트워크 서비스 중의 하나인 트위터 데이터를 활용하여 협업필터링 추천 시스템의 성능을 높이고자 한다. 협업필터링의 평점에 특정 아이템 관련 트윗을 수집해서 긍정/부정을 측정하여 가중치를 부여한다. RMSE 평가 방법을 통한 실험 결과, 소셜 미디어의 긍부정 영향력을 측정하여 적용했을 때가 기존의 협업필터링 방식에 비해 약 5.5%의 성능 향상을 확인하였다.

1. 서론

협업 필터링은 특정 사용자와 비슷한 특성을 가진 사용자들의 선호 목록을 기반으로 아이템을 추천해주는 방식이며 추천 시스템 중 가장 많이 사용되는 기법 중 하나이다. 기존 추천 시스템들이 아이템의 연관성만을 고려해 아이템의 특성을 기술하는데 한계가 있지만 협업필터링은 이러한 기술적 한계를 극복하는 장점이 있다[1].

협업 필터링의 두 가지 유형인 유저 기반 또는 아이템 기반 모두 유사 목록을 찾아내고 사용자에게 이를 기반으로 아이템을 추천해준다. 따라서 정확한 추천을 위해서는 사용자간의 유사성을 측정하고, 아이템 사이의 유사성을 측정하여 추천에 이용할 수 있도록 매우 많은 양의 정보가 필요하다는 단점이 있다[2].

[3]은 사용자와 아이템의 수는 많지만 사용자가 평점을 부여한 아이템의 수가 희박할 때 추천 성능이 낮아지는 데이터의 희박성 문제와 최근접 이웃을 찾기 위해 연산이 기하급수적으로 늘어나는 데이터 확장성 문제를 극복하는 방법으로 장르 정보를 이용한 협업필터링 추천시스템을 개발하였다. 장르별 협업 필터링은 아이템을 최종적으로 추천하기 전에 아이템의 상위 범주인 장르에 대한 정보를 활용하는 방법이다. 즉, 아이템에 대한 평가치 정보를 아이템의 상위 개념인 장르에 대한 정보로 집적함으로써, 협업 필터링의 대상이 되는 사용자×아이템 매트릭스의 차원을 축소시켜 희박성을 완화시키고, 계산양이 줄어들어 확장성 문제를 해결해 협업필터링의 성능이 향상되었다.

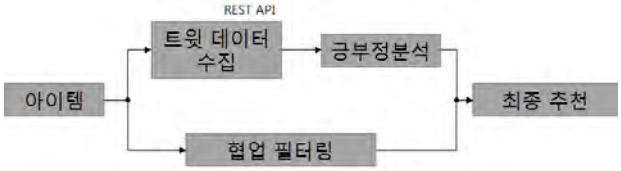
[4]는 온톨로지를 이용한 협업필터링의 성능 개선에 대한 연구를 수행하였다. 온톨로지는 사용자와 아이템간의 관계를 계층적으로 표현하는 방법이다. 협업필터링에서 데이터가 부족한 경우 성능이 많이 낮아지기 때문에 온톨로지를 기반으로 사용자의 아이템에 대한 선호도를 측정하여 데이터를 확장하였다. 따라서 데이터의 희박성 문제를 개선하여 협업필터링의 성능이 향상되었다.

본 논문에서는 협업 필터링과 트윗 데이터를 활용하여 향상된 추천 시스템을 제안한다. 협업 필터링에서 사용된 아이템 목록을 트위터에서 크롤링하여 트윗 데이터를 수집하고, 수집된 데이터로 긍부정 분석을 실시하여 아이템에 대한 SNS (Social Network Service)상의 선호도를 분석한다. 긍부정의 결과를 협업필터링의 평점에 적용하여 성능을 향상시켰다. 기존 협업필터링의 결과와 본 논문에서 제시하는 긍부정 분석을 적용한 결과를 RMSE(Root Mean Square Error)로 측정하였을 때 긍부정을 적용한 값이 기존의 협업필터링과 비교하여 약 5.5%가량 낮아졌다.

‡ 교신저자

2. 본론

본 논문에서는 협업필터링에 트윗 데이터의 긍부정 분석을 결합한 추천 시스템을 제안한다. <그림 1> 은 본 논문에서 제안하는 추천 시스템 구성도이다.



<그림 1> 긍부정 분석을 이용한 협업필터링 시스템 구성도

본 논문에서 제안하는 SNS의 긍부정 분석을 이용한 추천시스템은 해당 아이템에 대해 관련 트윗을 수집하고 수집된 트윗의 긍부정 분석을 통해 아이템에 대한 선호도를 측정하여 기존의 협업필터링 시스템에 적용하게 된다.

2.1 협업필터링

협업필터링에는 두 가지 방식이 있다. 사용자 기반 협업필터링은 사용자별로 선호도를 조사하여 사용자간 유사도를 계산한다. 유사도가 높은 사용자를 기반으로 아이템을 추천하게 된다. 사용자 기반 협업 필터링의 경우 신규 사용자가 주어졌을 경우 신규 사용자에 대한 유사성을 판단할 데이터가 존재하지 않아 추천을 해줄 수 없다는 단점이 있다. 이러한 문제점을 해결하기 위해 아이템 기반 협업 필터링이 있다. 이 방법은 아이템간의 유사도를 측정하여, 유저가 어떤 아이템을 선호하면 유사한 다른 아이템을 추천해주는 방법이다[5].

협업 필터링에서 항목간의 유사도를 구할 때 $k-nn$ (k-nearest neighbors) 알고리즘을 사용한다. $k-nn$ 알고리즘은 가장 유사한 k개의 데이터를 이용해서 새로운 데이터를 예측하는 방법이다[6]. 유사한 데이터를 구할 때 다양한 유사 계수를 구하는 공식이 있으며 본 논문에서는 EuclideanDistance, loglikelihood, pearsonCorrelation, TanimotoCoefficient 4가지를 사용한다. 최선의 k값을 선택하는 것은 데이터에 의존적이다. k값이 커질수록 잡음에 강하지만 데이터의 구조를 파악하는데 어려움이 있다[7].

$$w_r = r_i + \mu \circ \frac{\sum_{x}^{n} e_x}{n} \qquad \text{<수식 1>}$$

<표 1> <수식 1>의 변수 설명

변수	설명
w_r	긍부정이 적용된 평점 값
r_i	아이템 i에 대한 평점 값
n	아이템 i에 대해 크롤링된 트윗 개수
e_x	트윗을 긍부정 분석한 값
μ	가중치 값

2.2 트윗 데이터 긍부정 분석

협업필터링에 긍부정 값을 적용하기 위해 트위터의 REST API[8]를 통해 해당 아이템에 대해 트윗을 크롤링한다. 하나의 아이템에 대해 모아진 트윗들을 긍부정 분석하면 그 아이템에 대한 선호도를 파악할 수 있다. 선호도를 긍부정 값으로 수치화하여 사용자가 아이템에 부여한 평점을 계산할 때 가중치로 적용한다. 긍부정값을 협업필터링에 적용시키는 알고리즘은 <수식 1>과 같다.

각 아이템에 대해 트위터에서 크롤링을 수행하여 수집된 트윗마다 긍부정 분석을 실시한다. 긍부정 분석은 감정어 사전을 통해 트윗 문장을 긍정/부정으로 구분한다. 각 트윗별로 구한 긍부정 값에 평균을 구하고, 가중치를 조절하여 -1부터 1까지의 범위를 갖는 긍부정 값을 기존의 평점 값에 더해준다.

3. 실험 및 결과

3.1 데이터 집합

본 실험에 사용된 데이터는 추천시스템에 많이 사용되는 MovieLens와 Book-Crossing[9]이다. MovieLens는 1997년 미네소타 대학의 그룹렌즈 프로젝트에서 개발되었다. 실험에 참가한 사용자가 특정 영화에 대해서 평점을 매긴 데이터다. 본 논문에서는 MovieLens의 여러 데이터 중 약 100,000개의 데이터를 포함하는 'ml-100k'를 사용하였다. 데이터의 구성은 943명의 사용자가 1,682개의 영화에 대해 평점을 1~5점까지 정수 단위로 매겨 총 10만 개의 데이터로 이루어졌다. 각 사용자는 최소 20개 이상의 영화에 대해 평가를 하였다.

Book-Crossing은 2004년에 4주에 걸쳐 Book-Crossing community를 통해 책에 대해 평점을 매긴 데이터이다. 278,858명의 사용자가 271,379개의 책에 대해 평점을 1~10점까지 정수 단위로 매겨 총 1,149,780 개의 데이터로 이루어졌다. 사용자별로 평점을 부여한 평균 개수는 약 43개지만 그 편차가 크다. 평점을 부여한 개수가 너무 적을 경우 추천이 어렵기 때문에 실험에서는 15개 미만은 제외한 데이터를 생성하였다[10]. <표 2>는 실험에 사용된 데이터를 정리한 것이다.

<표 2> 실험에 사용된 데이터

	사용자 수	아이템 수	평점 범위
MovieLens	943	1682	1-5
Book-Crossing	4,975	136,939	1-10

3.2 긍부정 기반 추천 실험

기존의 협업필터링에서 긍부정을 적용했을 때 성능이 향상되는지 알아보기 위해 트위터에서 크롤링을 진행하였다. REST API로 각 데이터의 아이템에 대해 트윗 데이터를 4주 동안 수집하였다. 실험은 데이터를 80%/20%로 훈련/테스트 집합으로 나누었고, 협업필터링만으로 측정한 값과 긍부정 값을 적용했을 때를 RMSE(Root Mean Square Error)로 비교 분석하였다.

<표 3> MovieLens 데이터 셋으로 각 $k-nn$값, 각 유사도마다의 긍부정을 적용했을 때 RMSE 값

MovieLens		$k-nn$ 값				
		10	30	50	100	300
EuclideanDistance	기존	1.003119	1.010874	1.011445	1.015846	1.018456
	기존+긍부정	0.974211	0.976751	0.996892	1.012311	1.019482
	차이값	-0.028908	-0.034123	-0.014553	-0.003535	0.001026
loglikelihood	기존	1.052315	1.051842	1.043663	1.047801	1.049125
	기존+긍부정	1.008529	1.003804	0.979586	0.991710	1.010150
	차이값	-0.043786	-0.048038	-0.064077	-0.056091	-0.038975
pearsonCorrelation	기존	1.072990	1.071485	1.065019	1.069128	1.070485
	기존+긍부정	1.033882	1.024877	1.011311	1.023835	1.040610
	차이값	-0.039108	-0.046608	-0.053708	-0.045293	-0.029875
TanimotoCoefficient	기존	1.039801	1.034851	1.031343	1.038145	1.041804
	기존+긍부정	1.006919	0.990766	0.974349	0.988156	0.992256
	차이값	-0.032882	-0.044085	-0.056994	-0.049989	-0.049548

<표 4> Book-Crossing 데이터 셋으로 각 $k-nn$값, 각 유사도마다의 긍부정을 적용했을 때 RMSE 값

Book-Crossing		$k-nn$ 값				
		10	30	50	100	300
EuclideanDistance	기존	1.262736	1.105563	1.293843	1.325151	1.345129
	기존+긍부정	1.145185	0.987745	1.145860	1.158472	1.168477
	차이값	-0.117551	-0.117818	-0.147983	-0.166679	-0.176652
loglikelihood	기존	1.516746	1.493425	1.542724	1.598451	1.614779
	기존+긍부정	1.425190	1.417449	1.487100	1.517402	1.558904
	차이값	-0.091556	-0.075976	-0.055624	-0.081049	-0.055875
pearsonCorrelation	기존	1.802640	1.715734	1.918116	1.985740	2.052168
	기존+긍부정	1.699856	1.654522	1.771255	1.851486	1.889922
	차이값	-0.102784	-0.061212	-0.146861	-0.134254	-0.162246
TanimotoCoefficient	기존	1.458507	1.438036	1.542318	1.584116	1.612588
	기존+긍부정	1.345185	1.320482	1.421585	1.509985	1.554115
	차이값	-0.113322	-0.117554	-0.120733	-0.074131	-0.058473

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_{r,i} - x_{t,i})^2}{n}}$$ <수식 2>

<표 5> RMSE 표현식

n	n개의 추천된 데이터
x_r	제안방법에 의해 추천된 아이템의 평점
x_t	정답 집합에 저장된 아이템의 평점

$k-nn$의 적절한 k 값을 찾기 위해 k 값을 10, 30, 50, 100, 500으로 실험하였고 최근접 이웃을 구할 때 사용되는 유사도 방식으로 본문에서 설명한 4가지 방식을 사용하여 <표 3>, <표 4>의 결과를 얻었다. MovieLens와 Book-Crossing 데이터 셋을 비교했을 때 MovieLens가 더 낮은 RMSE 값을 보인다. 이는 Book-Crossing 데이터 셋에서 희소성을 낮추기 위해 여러 가지 방법으로 데이터를 줄였지만 MovieLens와 비교했을 때 희소성이 높기 때문에 추천 성능이 낮아지기 때문이다.

MovieLens 데이터에서는 EuclideanDistance 에서 k가 30일 때, 그 외의 나머지 유사도 방식에서는 k가 50일 때 성능이 가장 많이 개선되었다. Book-Crossing 데이터에서는 모든 유사도 계산 방식에서 k가 30일 때 긍부정 적용 전후 모두 가장 낮은 RMSE 값을 보였지만 성능은 pearsonCorrelation, TanimotoCoefficient 유사도 방식에서 k가 50일 때 가장 많이 개선되었다.

<표 6>은 [3]의 온톨로지를 이용한 협업필터링과의 성능 차이를 보여주고 있다. 모든 k값에서 제안한 방법의 성능이 더 높았고 k가 10일 때 가장 많이 성능이 개선되었다.

<표 6> 온톨로지 방법과의 성능 비교

Book-Crossing		$k-nn$ 값		
		10	30	50
pearson-Correlation	협업필터링	1.8026	1.7157	1.9181
	제안한 방법 (협업필터링+긍부정)	1.6998	1.6545	1.7712
	협업필터링+온톨로지	2.1233	1.9428	1.9744
	제안한 방법과 온톨로지 방법의 차이값	-0.4235	-0.2883	-0.2032

3.3 특정 데이터 제거에 따른 비교 분석

Book-Crossing 데이터의 경우 평점이 0으로 설정된 값이 50% 이상을 차지한다. 따라서 전체 데이터를 사용할 경우와 평점이 0인 항목을 제거한 데이터를 비교 분석하였다[11].

<표 7> 평점이 0인 데이터를 삭제했을 시 RMSE 결과 값

Book-Crossing		$k-nn$ 값		
		10	30	50
EuclideanDistance	기존	3.2851	3.5858	3.9518
	0제거	1.1451	0.9877	1.1459
Loglikelihood	기존	3.7514	4.0021	4.2514
	0제거	1.4251	1.4174	1.4871
pearsonCorrelation	기존	4.1152	4.5911	5.0485
	0제거	1.6999	1.6545	1.7713
TanimotoCoefficient	기존	3.8521	4.1785	4.4815
	0제거	1.3451	1.3205	1.4216

<표 7>에서 보이는 것과 같이 평점이 0인 데이터를 제거하였을 때 훨씬 낮은 RMSE 값을 얻을 수 있었다. 이는 데이터의 희소성을 낮추어서 추천 시스템의 성능이 향상되었기 때문이다.

3.4 추천 개수에 따른 비교 분석

추천 시스템에서는 추천해주는 개수에 따라 성능이 달라진다. 추천의 개수를 10, 50, 100개로 변화시켜 Book-Crossing 데이터로 실험하였다[12]. <표 8>에서 일반적으로 추천의 개수가 적을 때 RMSE 값이 더 낮게 나왔다. 이는 추천을 많이 해줄수록 오히려 추천해주는 평점의 범위가 넓어져서 예측하기 어려워지기 때문이다.

<표 8> 추천 개수에 따른 RMSE 결과 값

Book-Crossing		$k-nn$ 값		
		10	30	50
EuclideanDistance	10	1.1555	0.9914	1.7025
	50	1.1451	0.9877	1.1459
	100	1.2251	1.0985	1.1185
Loglikelihood	10	1.3371	0.9718	1.0574
	50	1.4251	1.4174	1.4874
	100	1.4510	1.7051	1.8851
pearsonCorrelation	10	1.4085	1.2985	1.7215
	50	1.6999	1.6545	1.7713
	100	1.6821	1.7543	1.8854
TanimotoCoefficient	10	1.2158	1.1854	1.3011
	50	1.3451	1.3205	1.4216
	100	1.4157	1.3851	1.4251

4. 결론

본 논문에서는 트윗 데이터를 긍부정 분석한 값을 협업필터링에 적용하여 기존의 협업필터링의 성능을 향상시키는 방법을 제안하였다. 트윗데이터를 수집하여 긍부정 분석을 수행하였고, 이를 협업 필터링에 적용하여 긍부정을 적용하였을 때 RMSE 값이 약 5.5%정도 더 낮아진 것을 확인하였다.

향후 연구로는 트위터 데이터의 영향력을 분석하여 추천 시스템에 적용하고자 한다. 즉, 트위터상에서 분석된 긍부정 값을 그대로 적용하는 것보다 해당 트윗을 작성한 사용자의 영향력을 측정하여 긍부정 값에 가중치를 부여할 수 있다. 팔로워 수가 많고 리트윗이 많이 발생한 사용자라면 그만큼 트위터상에서 다른 사용자들에게 높은 영향력을 발휘하므로 가중치를 높게 설정하는 연구를 진행하고자 한다.

감사의 글

본 논문은 중소기업청에서 지원하는 2017년도 산학연협력 기술개발사업(No.C0493205)의 연구수행으로 인한 결과물임을 밝힙니다.
본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학 ICT연구센터육성 지원사업의 연구결과로 수행되었음 (IITP-2017-2016-0-00314)

참고문헌

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

[2] Cho, Bongkwan, and Jaeil Jung. "A Study on Intelligent Railway Level Crossing System for Accident Prevention." International Journal of Railway 3.3 (2010): 106-112.

[3] 이재식, and 박석두. "장르별 협업필터링을 이용한 영화 추천시스템의 성능향상." 한국지능정보시스템학회논문지 13.4 (2007): 65-78.

[4] Yu, Li. "Using ontology to enhance collaborative recommendation based on community." Web-Age Information Management, 2008. WAIM'08. The Ninth International Conference on. IEEE, 2008.

[5] Wang, Jun, Arjen P. De Vries, and Marcel JT Reinders. "Unifying user-based and item-based collaborative filtering approaches by similarity fusion." Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2006.

[6] Cover, Thomas, and Peter Hart. "Nearest neighbor pattern classification." IEEE transactions on information theory 13.1 (1967): 21-27.

[7] Billsus, Daniel, and Michael J. Pazzani. "Learning Collaborative Information Filters." Icml. Vol. 98. 1998.

[8] https://dev.twitter.com/rest/public

[9] https://grouplens.org/datasets/

[10] Shen, Lei, and Yiming Zhou. "A new user similarity measure for collaborative filtering algorithm." Computer Modeling and Simulation, 2010. ICCMS'10. Second International Conference on. Vol. 2. IEEE, 2010.

[11] Tashkandi, Araek, Lena Wiese, and Marcus Baum. "Comparative Evaluation for Recommender Systems for Book Recommendations." BTW (Workshops). 2017.

[12] Mehta, Bhaskar, Thomas Hofmann, and Wolfgang Nejdl. "Robust collaborative filtering." Proceedings of the 2007 ACM conference on Recommender systems. ACM, 2007.