

비선호 분리 적용 콘텐츠 추천 방안

윤주영* · 이길흥**

Contents Recommendation Scheme Applying Non-preference Separately

Yoon Joo-young · Lee Kil-hung

〈Abstract〉

In this paper, we propose a recommendation system based on the latent factor model using matrix factorization, which is one of the most commonly used collaborative filtering algorithms for recommendation systems. In particular, by introducing the concept of creating a list of recommended content and a list of non-preferred recommended content, and removing the non-preferred recommended content from the list of recommended content, we propose a method to ultimately increase the satisfaction. The experiment confirmed that using a separate list of non-preferred content to find non-preferred content increased precision by 135%, accuracy by 149%, and F1 score by 72% compared to using the existing recommendation list. In addition, assuming that users do not view non-preferred content through the proposed algorithm, the average evaluation score of a specific user used in the experiment increased by about 35%, from 2.55 to 3.44, thereby increasing user satisfaction. It has been confirmed that this algorithm is more effective than the algorithms used in existing recommendation systems.

Key Words : Contents Recommendation, Collaborative Filtering, Matrix Factorization, Latent Factor Model

I. 서론

개인화 콘텐츠 추천시스템은 사용자들의 데이터를 수집하고 분석함으로써 기업이 더욱 정확한 의사 결정을 내릴 수 있도록 도울 수 있다. 이를 통해 시장 동향 파악, 상품 개발, 마케팅 전략 등 다양한 영역에서 데이터 기반의 최적화를 할 수 있다. 이러한 이유

로 최근 개인화 추천시스템이 큰 관심을 받고 있다. 개인화 콘텐츠 추천시스템은 사용자의 선호와 관심사에 기반하여 맞춤형 콘텐츠를 제공하여, 정보 검색 시간을 단축하고 고객 충성도를 높여 구매 및 사용량을 증가시킨다. 또한, 추천시스템을 통해 사용자들의 선호도와 행동 데이터를 분석하여 디지털 마케팅 전략을 개선하는 데 활용할 수 있을 정도로 그 중요성이 대두되고 있다.

사용자의 만족도를 높이는 것은 다음과같이 중요

* 서울과학기술대학교 컴퓨터공학과 박사과정

** 서울과학기술대학교 컴퓨터공학과 교수(교신저자)

하다. 기존 사용자의 유지율, 신규 사용자 유치 등을 긍정적인 방향으로 개선 시킬 수 있다. 제품이나 서비스를 구매하여 사업의 목표를 달성하기 위한 전환율을 높여 매출을 높이는 쪽으로 개선할 수 있다. 궁극적으로 사업의 경쟁률을 높여 고객의 충성도를 강화하는 등의 사업의 운영 및 성공에 중요한 요소이다. 현재까지 진행되어 온 추천시스템에 관한 연구들은 추천 알고리즘의 정확성이나 추천하는 알고리즘의 속도 개선 등 추천 알고리즘의 추천 성능 개선을 위한 연구가 대부분이다. 하지만 사용자가 비선호하는 콘텐츠를 찾아내서 사용자의 만족도를 높이는 등의 연구는 거의 없었다.

본 논문에서는 비선호 콘텐츠를 제외하는 형태로 사용자의 만족도 높일 수 있는 프로세스를 제안하고자 한다. 추천 모델 알고리즘에서 가장 많이 사용하고 있는 잠재 요인 협업 필터링(Latent Factor Collaborative Filtering) 알고리즘[1-3] 중의 하나인 행렬 분해(Matrix Factorization) 알고리즘을 이용하여 추천 콘텐츠 목록을 생성한다. 동일한 알고리즘을 활용하여 비선호 추천 콘텐츠 목록을 생성하고, 이 목록을 사용하여 추천 콘텐츠 목록을 개선함으로써 사용자 만족도 향상 여부를 확인하기 위한 실증 분석을 진행하는 것이다. 논문의 구성은 다음과 같다. 1장에서는 연구의 배경 및 목적을 설명하고, 2장에서는 추천시스템의 배경 이론 및 관련 연구, 제안 모델을 소개한다. 3장에서는 실험 데이터 소개와 실험 방법, 성능 평가 내용을 설명하고, 4장에서는 실험 결과에 대해서 분석한다. 마지막으로 5장에서는 실험 결과에 대해 해석하고, 이번 연구의 시사점과 추후 연구 방향에 대해서 언급한다.

II. 콘텐츠 추천 시스템 고찰

2.1 배경 이론

2.1.1 추천 시스템(Recommend System)

추천 알고리즘은 크게 인구통계학적 필터링(Demographic Filtering), 콘텐츠 기반 필터링(Content-based Filtering), 협업 필터링으로 분류할 수 있다[4]. 최근에는 여러 추천 알고리즘을 조합하여 사용하는 하이브리드 추천 시스템(Hybrid Recommend Systems)[5] 과 딥러닝 기반 추천 시스템(Deep Learning-based Recommend System) 등이 개발되었다.

인구통계학적 필터링은 비슷한 인구통계학적 특성을 바탕으로 비슷한 특성을 가진 사람들이 선호도가 비슷할 것으로 가정하는 것으로, 인종, 성별, 나이 등으로 사용자 집단을 분류하는 것이다. 콘텐츠 기반 필터링은 콘텐츠의 특성을 바탕으로 비슷한 특성을 좋아하는 사람들이 선호도가 비슷할 것으로 가정하는 것으로, 예를 들면 영화의 장르, 감독, 배우 등으로 사용자 집단을 분류하는 것이다.

협업 필터링 방식은 사용자 데이터의 특성을 바탕으로 비슷한 특성을 가진 사람들이 선호도가 비슷할 것으로 가정하는 것으로, 예를 들면 영화의 점수를 비슷하게 평가하는 사용자 집단을 분류하거나, A 상품을 구매한 사람들이 B 상품을 구매하는지에 대한 사용자와 아이템 간의 유사성을 계산하여 사용자 집단을 분류하는 것이다. 하이브리드 추천시스템은 여러 추천 알고리즘을 조합하여 개인화된 추천을 제공하는 방식이다. 일반적으로 협업 필터링과 콘텐츠 기반 필터링 등을 결합하여 사용한다. 딥러닝 기반 추천 시스템은 복잡한 패턴과 의미를 추출하는데 신경망 모델을 사용하는 방법이며, 사용자와 아이템 간의 상호 작용 패턴을 학습하고, 개인화된 추천을 제공한다.

2.1.2 협업 필터링(Collaborative Filtering)

최근에 가장 많이 사용되는 알고리즘은 협업 필터링[1-3]이며, 사용자와 아이템 간의 상호 상관 관계를 분석하여 새로운 사용자와 아이템 간 관계를 찾아 주는 것으로 사용자의 과거 경험 및 행동 방식에 의존한다[2]. 이 방식의 장점으로는 분야별 상세한 관련 지식이 필요하지 않다는 점이다. 하지만 새로운 사용자 또는 아이템이 등장 시 관련 데이터가 없기 때문에 추천을 할 수 없는 문제인 콜드 스타트 문제(Cold Start Problem)[6]가 나타난다.

협업 필터링은 최근접 이웃 방법(Nearest Neighborhood Methods)과 잠재 요인 모델(Latent Factor Models)로 구분할 수 있다. 최근접 이웃 방법의 경우에 사용자 기반 협업 필터링(User-based Nearest Neighbor)과 아이템 기반 협업 필터링(Item-based Nearest Neighbor)으로 나뉘어진다.

사용자 기반 협업 필터링의 경우 비슷한 취향을 가진 사용자들 기반으로 추천을 수행하는 것으로, 사용자들 간의 유사성을 계산하기 위해 사용자들의 평가 행렬을 분석한다. 예를 들어, 사용자 A와 비슷한 취향을 가진 사용자 B를 찾아 사용자 A가 평가하지 않은 항목 중에서 B가 좋은 평가를 한 항목을 추천할 수 있다. 일반적으로, 유사성 매트릭스(예, 코사인 유사도)를 사용한다.

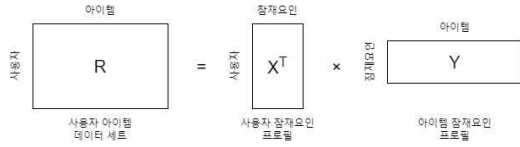
아이템 기반 협업 필터링의 경우 사용자들이 과거에 평가한 아이템들의 유사성을 계산하는 추천을 수행하는 것으로, 아이템 간의 유사성을 계산하기 위해 아이템들의 평가 행렬을 분석한다. 예를 들어, 사용자 A가 아이템 X를 좋아했을 때, 유사한 다른 아이템 Y를 찾아 A가 평가하지 않은 항목 중에서 Y를 추천할 수 있다. 본 논문에서 사용할 알고리즘인 행렬 분해(Matrix Factorization) 방법은 잠재 요인 모델(Latent Factor Models)[7]에 해당한다.

2.1.3 잠재 요인 협업 필터링(Latent Factor Collaborative Filtering)

잠재 요인 협업 필터링은 사용자와 아이템 간 상호 작용 패턴을 분석하여 유사한 성향을 보인 사용자나 아이템을 찾아 추천을 제공한다. 행렬 분해를 통해 얻은 잠재 요인 벡터를 이용하여 사용자와 아이템 간의 유사도를 측정하고, 유사한 사용자나 아이템을 기반으로 추천을 수행하는 방식으로 동작한다. 또한, 잠재 요인 협업 필터링은 사용자의 특성과 취향에 따라 개인화된 추천을 제공할 수 있는 장점이 있으며, 또한 새로운 사용자나 아이템에 대한 추천이 가능하여 콜드 스타트 문제(Cold Start Problem)[6]를 완화할 수 있으나 데이터의 부족으로 인하여 추천의 정확도가 떨어질 수 있다. 그리고, 사용자와 아이템 간 상호작용 행렬은 매우 크고 희소한 특성[8-10]을 가지는 경우가 많다. 행렬 분해를 통해 잠재 요인을 추출하기 위해서는 충분한 밀도의 상호작용 데이터가 필요하지만, 실제 데이터에서는 대부분의 셀이 비어 있을 수 있다. 이러한 희소성은 추천의 정확도를 저하할 수 있다. 행렬 분해는 일반적으로 복잡한 계산작업이다. 큰 규모의 상호작용 행렬을 계산하기 위해서는 시간과 자원이 더 많이 필요하다. 따라서 대규모 데이터 세트에 대한 행렬 분해는 높은 계산 비용이 요구된다. 그리고, 잠재 요인을 추출하는 과정에서 데이터의 차원을 축소하므로, 추출된 잠재 요인들의 의미를 해석이 상대적으로 어려울 수 있다는 단점이 있다.

2.1.4 행렬 분해 (Matrix Factorization)

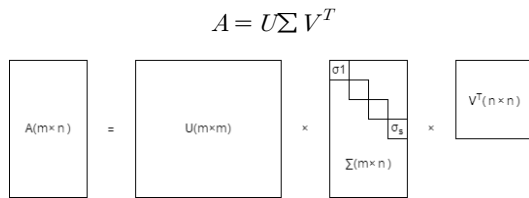
본 논문에서 사용한 행렬 분해 방법은 사용자와 아이템의 잠재적인 요인을 추출하고 이를 기반으로 추천을 수행하는 방법이다. 일반적으로 사용자와 아이템의 상호작용 행렬은 매우 크고 희소한(Sparse) 특성을 가지기 때문에, 행렬 분해를 통해 데이터의 차원을 축소하여 더욱 효율적으로 추천을 가능하게 한다.



<그림 1> 잠재 요인 협업 필터링 행렬 분해

2.1.5 특잇값 분해 (Singular Value Decomposition, SVD)

SVD라고도 부르는 특잇값 분해 알고리즘은 $m \times n$ 행렬 A에 대해서, 아래와 같이 3개 행렬의 곱으로 분해(Decomposition)해 가는 과정을 말한다.



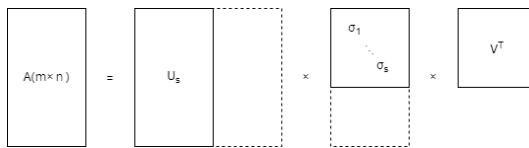
<그림 2> Full SVD

3개의 행렬은 다음 조건을 만족해야 한다.

- U : $m \times m$ 직교 행렬 ($AA^T = U(\Sigma\Sigma^T)U^T$)
- Σ : $m \times n$ 직사각 대각 행렬
- V : $n \times n$ 직교 행렬 ($A^TA = V(\Sigma^T\Sigma)V^T$)

2.1.6 특잇값 분해의 변형

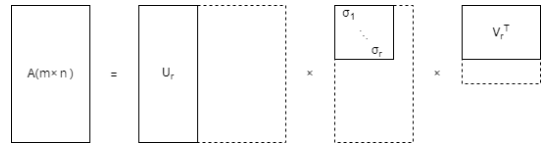
(1) Thin SVD : Economy SVD라고도 하는 Thin SVD는 Σ 를 $s \times s$ 로 줄여서 만든다.



<그림 3> Thin SVD

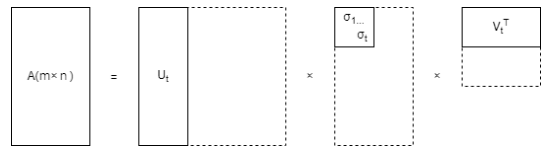
(2) Compact SVD : 원래 행렬의 압축을 위해 가장

중요한 특잇값과 벡터만 유지한다. 특잇값 중 0을 제거해 r 개의 특잇값만 남겨서 만든다.



<그림 4> Compact SVD

(3) Truncated SVD : 모든 특잇값을 유지하는 Thin SVD와 Compact SVD와 달리 top-k 특잇값과 벡터만 유지한다. 덜 중요한 특잇값과 벡터를 버림으로써 추가 압축 및 차원 감소를 할 수 있다. Compact SVD의 r 개의 특잇값 중 t 개의 특잇값을 사용해 만든다.



<그림 5> Truncated SVD

Full SVD, Thin SVD, Compact SVD 등과는 달리 Truncated SVD는 top-k 특잇값과 벡터만 압축했기 때문에 행렬 A를 완벽히 원복할 수 없지만, 복잡한 계산을 하지 않고도 행렬 A를 근사할 수 있는 장점이 있다. 대규모 희소행렬 작업 시 유용하며, 잠재 요인 협업 필터링에서 가장 많이 사용된다.

2.2 관련 연구

추천시스템 또는 협업 필터링이라고 부르는 것은 흥미로운 콘텐츠를 찾는 좋은 방법이 비슷한 관심사를 가진 다른 사람들을 찾고, 비슷한 사용자가 좋아하는 것을 추천하는 것이라는 가정에 기반한다[11].

Netflix Prize 대회에서 입증된 바와 같이, 행렬 분

해 모델은 제품 추천 목록을 생성할 때 암시적 피드백(Implicit Feedback), 시간 효과(Temporal Effects) 및 신뢰 수준(Confidence Levels) 등의 추가 정보를 결합할 수 있기 때문에, 기존의 최근접 이웃 기술보다 우수하다[2].

하지만, 이러한 전통적인 추천시스템은 신규 소비자 혹은 상품에 대한 추천이 불가능하거나, 소비자 및 상품에 대한 콘텐츠 정보를 쉽게 다루지 못하는 문제점이 존재한다. 전통적인 추천시스템의 한계점을 극복하기 위해, 최근 몇 년간 딥러닝 기반의 다양한 추천시스템들이 연구되고 있다. 소비자 및 상품의 다양한 콘텐츠 정보와 대용량의 데이터로부터 중요한 특성을 효과적으로 모형화하며 전통적인 추천시스템보다 더 높은 성능을 달성했다. 하지만 기존 딥러닝 기반의 추천시스템은 복잡한 모델 구조로 인해 과적합 문제를 야기시킨다[12].

협업 필터링 추천 알고리즘의 최근접 이웃 기반 알고리즘 등을 활용한 추천 시스템 연구는 추천 알고리즘의 정확도 측면에서는 높은 성능을 보였으나 사용자의 만족도에는 부정적인 영향을 미칠 수 있다는 주장이 있으며[13], 추천시스템을 평가할 때, 우연성(Serendipity) 또는 다양성(Coverage) 등의 지표를 통해 만족도를 향상할 수 있다는 것이 증명되었다[14].

우연히 발견한 좋은 분위기의 카페나 예기치 않았던 장소에서 우연히 만난 친구처럼 '뜻밖의 기쁨, 우연한 발견'을 뜻하는 '세렌디피티(Serendipity)'가 적용되면 서비스의 사용에 즐거움을 증진할 수 있다고 보았다[15]. 추천 시스템의 정확성, 다양성, 우연성이 만족도에 미치는 영향을 검증한 결과 세 가지 요인 모두 만족도에 유의미한 영향을 미치는 것으로 나타났다[16].

이런 분석 방법 이외에, 사용자의 감성 분석에 기반한 콘텐츠 추천 방법도 있다. 트위터 특성에 맞는 효과적인 감성 분석을 위해 사용자 키워드별 감성 목록을 만들고, 이를 ACO 알고리즘을 적용하여 사용자 별 시간에 따른 키워드의 감성 변화를 파악하였다.

특정 사용자가 긍정적이고, 호감을 느끼는 단어가 무엇인지, 또는 부정적이거나 꺼리는 단어가 무엇인지 파악이 가능하기 때문에 특정 사용자에게 프로파일링이 가능해지고 이를 이용하여 사용자별 맞춤 콘텐츠 추천이 가능하여지도록 해준다[17].

사용자 편의성 등의 측면을 고려한 UI/UX 개선을 통한 추천 방법도 있다. 추천 시스템의 궁극적인 목적이 사용자들에게 즐거움을 선사해 주는 것이라 할 때, 추천 성능의 개선만으론 그 목적을 달성하기 어렵다. 하지만 애플의 아이폰 성공 사례를 보듯이, 기능 외에도 사용자 경험(User Experience) 관점에서 제공된 뛰어난 사용자 인터페이스(User Interface)를 제공하여 긍정적인 사용자 경험을 창출하는 것이 중요 과제가 되었다[18].

이러한 선행연구들에도 불구하고 기존의 추천 알고리즘의 정확도 등의 성능 개선 이외 우연성 또는 다양성 등의 지표를 통해 만족도를 높이고 있는데, 사용자가 비선호하는 콘텐츠를 추천하지 않음으로써, 사용자의 만족도를 높이는 방법에 대한 연구는 국내외 문헌을 찾아본 결과 부재한 것으로 확인했다.

2.3 제안 모델

이전의 많은 연구를 통해 현재 가장 많이 사용되고 있는 잠재 요인 협업 필터링 알고리즘 중의 하나인 행렬 분해 모델을 이용한다. 영화 데이터를 이용하여 실험을 진행하며, 사용자가 입력한 평가 점수를 사용자가 입력한 순서에 맞춰 하나씩 데이터 세트에 추가하여 추천 콘텐츠 목록을 만들고, 사용자가 입력한 평가 점수를 0.5는 5, 1은 4.5, 1.5는 4, 2는 3.5, 2.5는 3, 3은 2.5, 3.5는 2, 4는 1.5, 4.5는 1, 5는 0.5 이런 식으로 역변환하여 입력하여 비선호 추천 콘텐츠 추천 목록을 생성한다.

실험군의 경우에는 다음 입력으로 예정된 사용자 평가 점수가 0.5에서 2.5 사이라면 비선호하는 것으로 보

고 비선호 추천 콘텐츠 목록에서 양수로 예측되었는지 확인하며, 3에서 5 사이라면 선호하는 것으로 보고 추천 콘텐츠 목록에서 양수로 예측되었는지 확인한다.

대조군의 경우에는 다음 입력으로 예정된 사용자 평가 점수가 0.5에서 2.5 사이라면 비선호하는 것으로 보고 추천 콘텐츠 목록에서 음수로 예측되었는지 확인하며, 3에서 5 사이라면 선호하는 것으로 보고 추천 콘텐츠 목록에서 양수로 예측되었는지 확인한다.

III. 비선호 분리 적용 콘텐츠 추천 방안

3.1 실험 데이터

실험에 사용한 데이터 세트는 Movie Lens 영화 데이터 세트이다[19]. 이 데이터 세트는 671명의 사용자가 100,004건의 영화 순위를 매긴 데이터 세트와 9,125건의 영화정보 데이터 세트를 포함한다.

데이터 세트에는 영화와 관련된 정보가 들어있는 movies.csv, 사용자들의 영화에 대한 평가 정보가 들어있는 ratings.csv, 각 영화의 Tag 정보가 들어있는 tag.csv 등 세 가지 파일이 있다. 실험에서는 movies.csv 와 ratings.csv 파일을 사용하여 실험을 진행했다.

	userid	movied	rating	timestamp
0	1	31	2.5	1260759144
1	1	1029	3.0	1260759179
2	1	1061	3.0	1260759182
3	1	1129	2.0	1260759185
4	1	1172	4.0	1260759205
...
99999	671	6268	2.5	1065579370
100000	671	6269	4.0	1065149201
100001	671	6365	4.0	1070940363
100002	671	6385	2.5	1070979663
100003	671	6565	3.5	1074784724

100004 rows × 4 columns

<그림 6> ratings.csv 데이터 세트

항목별 데이터는 아래와 같다.

- **userId** : 사용자별로 부여된 ID 값
- **movied** : 영화별로 부여된 ID 값
- **rating** : 사용자의 평가 점수(0.5부터 5점까지 있으며, 0.5점씩 증가한다.)
- **timestamp** : 데이터가 입력된 시간

movied	title	genres
0	1 Toy Story (1995)	Adventure Animation Children Comedy Fantasy
1	2 Jumanji (1995)	Adventure Children Fantasy
2	3 Grumpier Old Men (1995)	Comedy Romance
3	4 Waiting to Exhale (1995)	Comedy Drama Romance
4	5 Father of the Bride Part II (1995)	Comedy
...
9120	162672 Mohenjo Daro (2016)	Adventure Drama Romance
9121	163056 Shin Godzilla (2016)	Action Adventure Fantasy Sci-Fi
9122	163949 The Beatles: Eight Days a Week - The Touring Y...	Documentary
9123	164977 The Gay Desperado (1936)	Comedy
9124	164979 Women of '69, Unboxed	Documentary

9125 rows × 3 columns

<그림 7> movies.csv 데이터 세트

항목별 데이터는 아래와 같다.

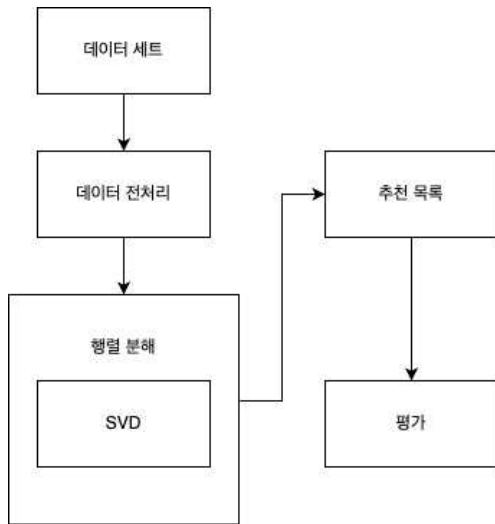
- **movied** : 영화별로 부여된 ID 값
- **title** : 영화 제목
- **genres** : 영화의 장르

3.2 실험 방법

실험을 위해 행렬 분해 알고리즘을 이용한 코드를 작성하였다. 준비된 데이터 세트를 입력하면, 데이터 전처리를 하고 이후 행렬 분해를 통해 추천 목록을 만든다. 이렇게 나온 추천 목록을 가지고 평가하는 형태로 연구를 진행하였다.

데이터 분석을 위해 사용자를 1명 선택했고, 선택한 사용자는 1번이다. 영화 평가 점수는 0.5(불만족), 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5(만족)점까지 점수 분포가 고르게 분포되어 있고, 평가 점수 평균도 2.95점이다. 평가한 영화 수도 20개로 적당하여 선택했다. 사용자 1번의 데이터 세트는 <그림 9>에서 확인할 수 있다.

모든 사용자 평가 데이터에는 Timestamp라는 평

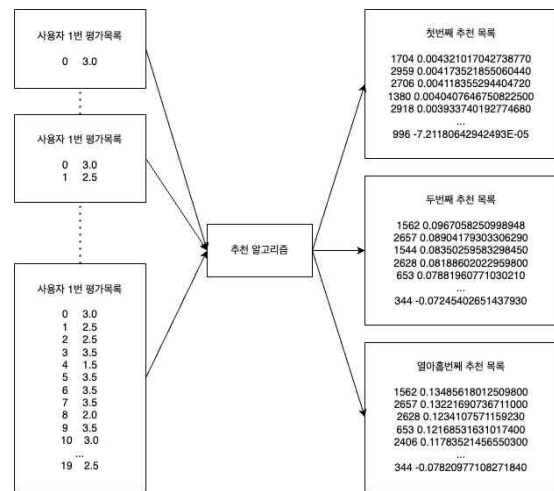


<그림 8> 프로그램 전체 흐름도

userId	movield	rating	timestamp
0	1	31	3.0
1	1	1029	2.5
2	1	1061	2.5
3	1	1129	3.5
4	1	1172	1.5
5	1	1263	3.5
6	1	1287	3.5
7	1	1293	3.5
8	1	1339	2.0
9	1	1343	3.5
10	1	1371	3.0
11	1	1405	4.5
12	1	1953	1.5
13	1	2105	1.5
14	1	2150	2.5
15	1	2193	3.5
16	1	2294	3.5
17	1	2455	3.0
18	1	2968	4.5
19	1	3671	2.5

<그림 9> 사용자 1번 데이터 세트

가를 진행한 시간이 포함되어 있다. 그래서 시간 흐름에 따라 순차적으로 어떤 영화에 평가했는지 알 수가 있었다. 그 흐름에 맞춰 사용자 1번의 평가를 1편씩, 20편까지 누적으로 1편씩 추가하면서 총 19번의 추천 콘텐츠 목록을 예측하였다.



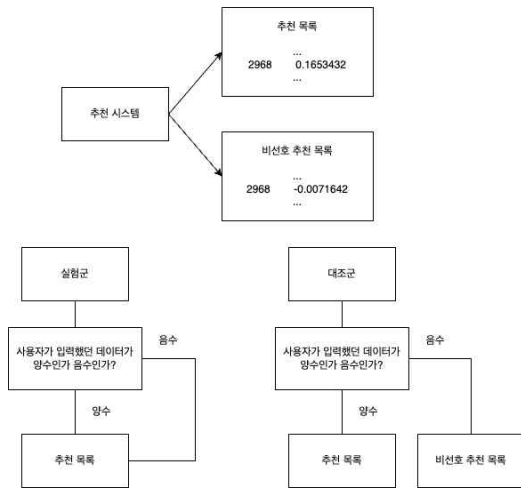
<그림 10> 추천 목록을 예측한 방법

그리고, 비선호 추천 목록을 만들기 위해 모든 사용자 평가 데이터를 다음과 같이 변경하였다.

<표 1> 데이터 변환 시트

원본 평가 점수	변환 평가 점수
0.5	5
1	4.5
1.5	4
2	3.5
2.5	3
3	2.5
3.5	2
4	1.5
4.5	1
5	0.5

이렇게 변환한 데이터 세트는 추천 목록을 뽑아낸 방식과 같은 방식으로 행렬 분해 알고리즘을 이용하여 비선호 추천 목록을 뽑아내었다. 행렬 분해에는 특잇값 분해(SVD : Singular Value Decomposition)를 사용하였다. 데이터 효율을 위해 Full SVD가 아닌 Truncated SVD를 이용하여 행렬 분해를 진행하였다. 실험에는 특잇값을 12로 하여 Truncated SVD를 수행하였다.



<그림 11> 추천 목록 A, B를 이용한 평가 방법

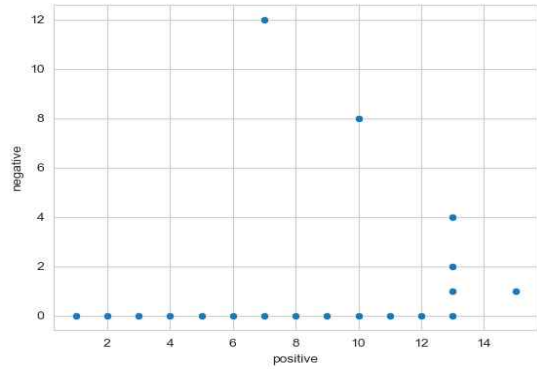
이후 추천 목록과 비선호 추천 목록을 사용자 1번의 시간 흐름에 따른 평가 데이터와 예측한 추천 콘텐츠를 1대 1로 비교하는 방식으로 평가를 진행했다. 비교는 행렬 분해 알고리즘으로 만든 추천 목록의 추천 및 비추천 항목의 추천 목록 A와 같은 행렬 분해 알고리즘으로 만든 추천 목록과 사용자의 평가 데이터를 뒤집은 데이터 세트로 만든 비선호 추천 목록 B를 가지고 비교하였다.

추천 목록에서 Prediction 점수가 양수인 경우에는 추천, 음수인 경우에는 비추천으로 보고, 원본 평가 데이터가 0.5에서 2.5는 비선호 콘텐츠로 분류하고, 3에서 5까지는 선호 콘텐츠로 분류하여 성능평가를 진행하였다.

IV. 제안 방안 실험 및 분석

4.1 제안 방안 실험

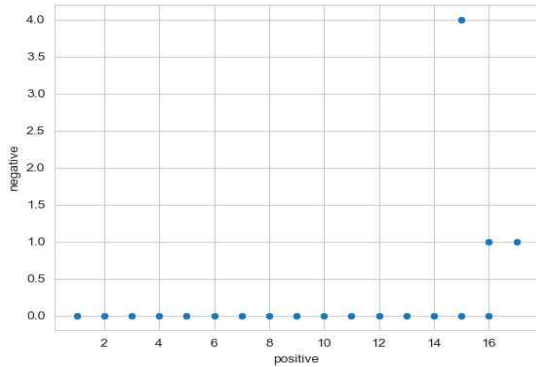
사용자 1번의 평가 점수 20개를 토대로 평가 점수 1번을 입력한 상태에서 추천 목록을 예측하고, 평가 점수 1, 2번을 입력한 상태에서 추천 목록을 예측하는 등 점진적으로 평가 점수 1번부터 19번까지 데이터를 누적적으로 입력하고 매번 추천 콘텐츠 목록을 만들었다. 추천 콘텐츠 목록에 1번을 입력한 상태에서 만들어진 추천 목록이라면, 추천 목록 2번에서 20번까지의 영화가 추천 콘텐츠 목록에 있는지 확인하는 형태로 추천 콘텐츠 데이터 세트를 만들었다. 아래 <그림 12>, <그림 13>은 해당 목록을 1번부터 19번까지 진행한 결과이다.



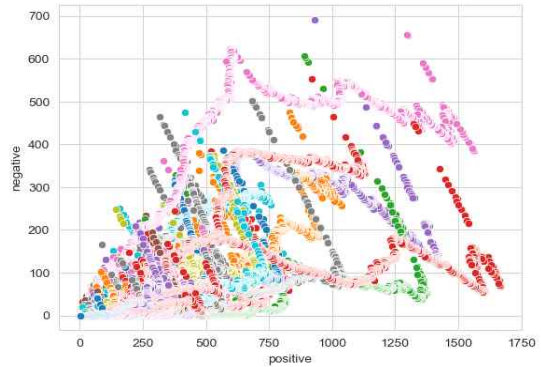
<그림 12> 사용자 1번의 추천 콘텐츠 산포도

추천 콘텐츠 데이터 세트를 확인한 결과 Positive한 추천 목록은 잘 찾으나, Negative한 비선호 추천 목록을 잘 찾아내지 못하는 것을 확인할 수 있었다. 이러한 흐름이 사용자 1번에만 국한된 것인지 확인을 하기 위해, 671명의 사용자 전체에 대한 데이터도 아래의 <그림 14>, <그림 15>을 통해 확인하였다.

<그림 14>, <그림 15>에서 확인한 바와 같이



<그림 13> 사용자 1번의 비선호 추천 콘텐츠 산포도

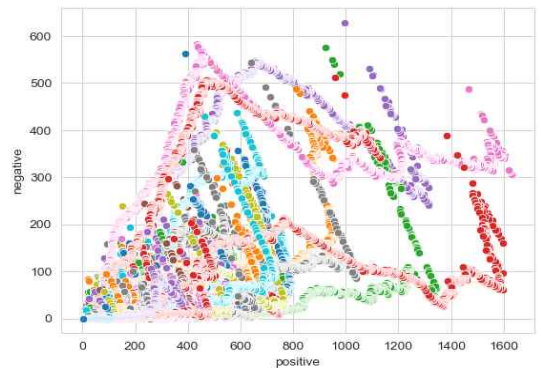


<그림 14> 671명 사용자의 추천 콘텐츠 산포도

Positive와 Negative를 고르게 예측하여 추천 콘텐츠 리스트를 생성하는 경우보다, Positive 한 결과만을 위주로 추천 콘텐츠 리스트가 생성하는 경우가 많은 것을 확인할 수 있었다. <그림 12>, <그림 13>의 사용자 특성이 다른 사용자들에게도 나타나는 것을 보았을 때, 알고리즘의 고유한 특성으로 보인다.

사용자 1번의 평가 점수를 입력 시간을 토대로 1개씩 누적으로 늘려가며 다음 추천 목록을 예측해 보았다. 예를 들자면, 1개의 평가점수를 입력 후 나머지 19개의 목록에 대한 추천 목록을 생성하고, 2개의 평가점수를 입력 후 나머지 18개의 목록에 대한 추천 목록을 생성하는 식으로 총 19번의 추천 목록을 생성해 보았다. 생성이 완료된 이후에는 오차 행렬 (Confusion Matrix)을 이용하여 정밀도(Precision), 재현율(Recall), 정확도(Accuracy), F1 점수(F1-score)에 대한 성능 평가를 진행하였다.

성능 평가 결과 추천 콘텐츠 목록은 정밀도 평균 37%, 재현율 평균 85%, 정확도 평균 39%, F1 점수 평균 50%의 성능이 나타났다. 다른 수치에 비해 상대적으로 높은 재현율을 가지고 있지만, 정밀도와 정확도가 낮은 상황이다. 이 모델은 많은 양의 실제 양성(TP : True Positive) 예측값을 내고 있긴 하지만, 반대로 많은 양의 거짓양성(FP : False Positive) 예측값을



<그림 15> 671명 사용자의 비선호 추천 콘텐츠 산포도

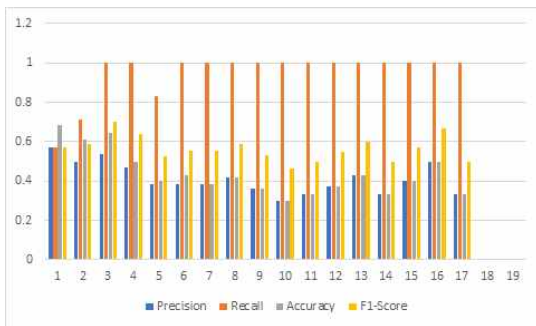
내고 있다는 것이다. 전반적인 모델 성능은 F1 점수를 보면 중간 정도로 평가할 수 있다. 모델의 성능 향상을 위해 추가적인 분석과 개선이 필요한 상황이다.

<표 2> 사용자 1번의 추천 콘텐츠 목록 성능 평가표

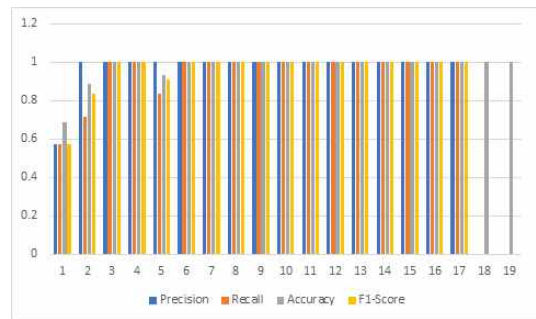
	FN	FP	TN	TP	Precision	Recall	Accuracy	F1-score
1	3	3	9	4	0.57	0.57	0.68	0.57
2	2	5	6	5	0.5	0.71	0.61	0.59
3	0	6	4	7	0.54	1	0.65	0.7
4	0	8	1	7	0.47	1	0.5	0.64
5	1	8	1	5	0.38	0.83	0.4	0.52
6	0	8	1	5	0.38	1	0.42	0.56
7	0	8	0	5	0.38	1	0.38	0.56

8	0	7	0	5	0.42	1	0.42	0.59
9	0	7	0	4	0.37	1	0.36	0.53
10	0	7	0	3	0.3	1	0.3	0.46
11	0	6	0	3	0.33	1	0.33	0.5
12	0	5	0	3	0.38	1	0.38	0.55
13	0	4	0	3	0.43	1	0.43	0.6
14	0	4	0	2	0.33	1	0.33	0.5
15	0	3	0	2	0.4	1	0.4	0.57
16	0	2	0	2	0.5	1	0.5	0.67
17	0	2	0	1	0.33	1	0.33	0.5
18	0	2	0	0	0	0	0	0
19	0	1	0	0	0	0	0	0
평균					0.37	0.85	0.39	0.51
표준 편차					0.15	0.32	0.18	0.19

9	0	0	7	4	1	1	1	1
10	0	0	7	3	1	1	1	1
11	0	0	6	3	1	1	1	1
12	0	0	5	3	1	1	1	1
13	0	0	4	3	1	1	1	1
14	0	0	4	2	1	1	1	1
15	0	0	3	2	1	1	1	1
16	0	0	2	2	1	1	1	1
17	0	0	2	1	1	1	1	1
18	0	0	2	0	0	0	1	0
19	0	0	1	0	0	0	1	0
평균					0.87	0.85	0.97	0.86
표준 편차					0.32	0.32	0.08	0.32



<그림 16> 사용자 1번의 추천 콘텐츠 성능 결과 그래프



<그림 17> 사용자 1번의 비선호 추천 콘텐츠 성능 결과 그래프

다음은 추천 콘텐츠 목록과 비선호 추천 콘텐츠 목록을 각각 만들어 비교한 성능 평가표 내용이다.

<표 3> 사용자 1번의 비선호 추천 콘텐츠 목록 성능 평가표

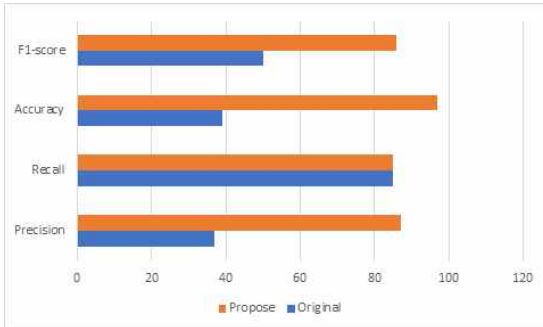
	FN	FP	TN	TP	Precision	Recall	Accuracy	F1-score
1	3	3	9	4	0.57	0.57	0.68	0.57
2	2	0	11	5	1	0.71	0.89	0.83
3	0	0	10	7	1	1	1	1
4	0	0	9	7	1	1	1	1
5	1	0	9	5	1	0.83	0.93	0.91
6	0	0	9	5	1	1	1	1
7	0	0	8	5	1	1	1	1
8	0	0	7	5	1	1	1	1

4.2 실험 결과 분석

추천 콘텐츠를 잘 분류하는 알고리즘의 특성을 이용하여 반대로 비선호 콘텐츠를 추천하게 했더니, 성능 평가 결과 정밀도 평균 87%, 재현율 평균 85%, 정확도 평균 97%, F1 점수 평균 86%의 성능이 나타났다. 높은 정밀도와 재현율을 가지고 있으며, 정확도 또한 매우 높게 나왔다. F1 점수도 높은 값을 가지고 있어 모델의 성능은 매우 좋다고 평가할 수 있다.

기존의 콘텐츠 추천 알고리즘에 비해 봐도 정밀도 135%, 정확도 149%, F1 점수 72% 등의 증가함을 알 수가 있다. 이러한 알고리즘을 통하여 사용자의 추천

콘텐츠 목록에서 비선호 콘텐츠를 제거하여 사용자가 보지 않게 되는 경우를 가정하면, 실제 사용자 1번의 평가 점수의 평균이 2.55점에서, 3.44점으로 추천 점수가 약 35% 향상됨을 확인할 수 있었다.



<그림 18> 기존 알고리즘 대비 제안 추천 알고리즘 성능 비교

V. 결론 및 향후 연구 방안

본 연구에서는 추천 콘텐츠 목록을 이용하는 사용자의 만족도를 개선하기 위해 비선호 추천 콘텐츠 목록을 만들고, 비선호 콘텐츠를 제거하는 형태로 사용자의 콘텐츠 이용에 대한 만족도를 높일 수 있는지 실험하였다. 추천 콘텐츠 목록 생성에는 행렬 분해 알고리즘을 사용하였으며, 비선호 콘텐츠 목록을 만들기 위해 사용자가 입력한 평가 점수를 역변환하여 데이터 세트를 만들었다.

실험 결과 기존 방식의 추천 프로세스보다 제안한 추천 프로세스가 비선호 콘텐츠를 잘 예측해 내었으며, 결과적으로 사용자의 평가 점수가 향상되는 결과를 기대할 수 있다. 이번 실험에 사용한 사용자 1번이 입력한 평가 점수 항목을 예시로 들어 본다면, 기존에 사용자 평가 점수의 평균이 2.55점이었으나, 비선호 콘텐츠가 제거되었다고 가정했을 때, 사용자 평가 점수의 평균이 3.44로 약 35% 향상되었다. 이처럼 사용자가 선호하지 않는 콘텐츠를 발견하여 추천 콘

츠 목록에서 제거할 수 있다면 사용자의 이용 만족도가 높아지는 것을 확인하였다.

참고문헌

- [1] A. Das, M. Datar, A. Garg, S. Rajaram, "Google news personalization: Scalable online collaborative filtering," Proceedings of the 16th international conference on World Wide Web, 2007, pp.271-280.
- [2] Y. Koren, R. Bell, C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer, Vol.42, No.8, 2009, pp.30-37.
- [3] G. Linden, B. Smith, J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," IEEE Internet Computing, Vol.7, No.1, 2003, pp.76-80.
- [4] Montaner, M., López, B., & De La Rosa, J. L, "A taxonomy of recommender agents on the internet," Artificial intelligence review, Vol.19, No.4, 2003, pp.285-330.
- [5] M. Balabanović, Y. Shoham, "Fab: content-based, collaborative recommendation," Communications of the ACM, Vol.40, No.3, 1997, pp.66-72.
- [6] A. Schein, A. Popescul, L. Ungar, D. Pennock, "Methods and Metrics for Cold-Start Recommendations," Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2002, pp.253-260.
- [7] C.Lin, Y.He, R.Everson, S. Ruger, "Weakly Supervised Joint Sentiment-Topic Detection

from Text," IEEE Transactions on Knowledge and Data Engineering, Vol.24, 2012.

[8] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, John Riedl, "An Algorithmic Framework for Performing Collaborative Filtering," ACM, 1999, pp.230-237.

[9] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, John Riedl, "Explaining Collaborative Filtering Recommendations," ACM, 2000, pp.241-250.

[10] Paul Resnick, Neophytos Iaconou, Mitesh Suchak, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," ACM, 1994, pp.175-186.

[11] Resnick, P. and Varian, H, "Recommender systems," Communications of the ACM, Vol.40, No.3, 1997, pp.56-58.

[12] 이승연, "교차 합성곱을 이용한 딥러닝 기반의 추천 시스템 개발," 명지대학교 석사학위논문, 2020.

[13] McNee, S. M., Riedl, J., & Konstan, J. A, "Being accurate is not enough: how accuracy metrics have hurt recommender systems," CHI'06 extended abstracts on Human factors in computing systems, 2006, pp.1097-1101.

[14] Ge, M., Delgado-Battenfeld, C., & Jannach, D, "Beyond accuracy: evaluating recommender systems by coverage and serendipity," Proceedings of the fourth ACM conference on Recommender systems, 2010, pp.257-260.

[15] 백소담, 이원영, 채안병, 황은영, 김성우, "온라인 동영상 환경에서의 세렌디피티 요인에 관한 탐색," 한국 HCI 학회 논문지, 제12권, 제3호, 2017, pp.25-33.

[16] 권장욱, "맞춤형 추천시스템에서의 세렌디피티 효과에 관한 실증 연구," 관광레저연구, 제33권,

제1호, 2021, pp.357-376.

[17] 이정민, "소셜 미디어의 감성 분석에 기반한 콘텐츠 추천 방법에 관한 연구," 고려대학교 석사학위논문, 2016.

[18] 문창식, "영화 콘텐츠 추천을 위한 웹페이지 UI/UX 설계 및 구현," 한양대학교 석사학위논문, 2015.

[19] F. Maxwell Harper and Joseph A. Konstan, "The MovieLens Datasets: History and Context," ACM Transactions on Interactive Intelligent Systems (TiiS), Vol.5, No.4, 2015, p.19.

■ 저자소개 ■



윤주영
(Yoon Joo-young)

2023년 9월-현재
서울과학기술대학교
컴퓨터공학과(박사과정)
2023년 8월 서울과학기술대학교
빅데이터경영정보학과(공학석사)
관심분야 : 추천시스템
E-mail : wisewiz@kakao.com



이길흥
(Lee Kil-hung)

2000년 3월-현재
서울과학기술대학교 컴퓨터공학과
교수
1999년 8월 연세대학교 대학원
전기컴퓨터공학과(공학박사)
1991년 2월 연세대학교 대학원 전자공학과
(공학석사)
1989년 2월 연세대학교 전자공학과(공학사)
관심분야 : 컴퓨터네트워크, 초고속통신망,
통신망관리, 무선센서망,
인터넷서비스
E-mail : khlee@seoultech.ac.kr

논문접수일 : 2023년 8월 25일
수정접수일 : 2023년 9월 8일(1차)
2023년 9월 11일(2차)
게재확정일 : 2023년 9월 14일