JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 26, No. 9: 1293~1304, Sep. 2022

온라인 쇼핑몰 데이터를 이용한 개인화 추천 모델 성능 비교 분석

오재동[1] · 오하영[2*]

A Comparative Analysis of Personalized Recommended Model Performance Using Online Shopping Mall Data

Jaedong Oh[1] · Ha-young Oh[2*]

[1]Graduate Student, Aritificial Intelligence Convergence, Sungkyunkwan University, Seoul, 03063 Korea
[2*]Associate Professor, College of Computing and Informatics, Sungkyunkwan University, Seoul, 03063 Korea

요 약

개인화 추천시스템은 각 개인의 관심사나 선호도를 분석하여 이에 맞는 정보나 제품을 추천해주는 것을 의미한다. 이러한 개인화 추천을 통해 소비자들은 본인에게 필요한 제품들을 보다 빠르게 접함으로써 정보 탐색에 소모하는 시간을 단축할 수 있으며, 기업들은 소비자들의 필요에 맞는 적절한 제품을 추천해줌으로써 기업 이윤을 증가시킬 수 있다. 본 연구에서는 대표적인 개인화 추천 기법들인 협업 필터링, 행렬 요인화, 딥러닝을 사용하여 소비자에게 제품을 추천해준다. 이를 위해 원데이터 (Raw data)인 쇼핑몰 상품 구매 후기 데이터세트를 추천시스템의 입력으로 전달하기 위한 형태로 전처리하고, 전처리한 데이터세트를 다각도로 분석해본다. 또한, 각각의 모델들이 추천한 결과에 대해 검증 및 성능 비교를 수행하고 최적의 성능을 보이는 모델을 탐색하여 이후 해당 쇼핑몰에서 추천시스템 구축 시 어떤 모델을 사용하는 것이 좋을지를 제시한다.

ABSTRACT

The personalization recommendation system means analyzing each individual's interests or preferences and recommending information or products accordingly. These personalized recommendations can reduce the time consumers spend searching for information by accessing the products they need more quickly, and companies can increase corporate profits by recommending appropriate products that meet their needs. In this study, products are recommended to consumers using collaborative filtering, matrix factorization, and deep learning, which are representative personalization recommendation techniques. To this end, the data set after purchasing shopping mall products, which is raw data, is pre-processed in the form of transmitting the data set to the input of the recommended system, and the pre-processed data set is analyzed from various angles. In addition, each model performs verification and performance comparison on the recommended results, and explores the model with optimal performance, suggesting which model should be used when building the recommendation system at the mall.

키워드 : 추천시스템, 협업 필터링, 행렬 분해, 딥러닝

Keywords : Recommendation System, Collaborative Filtering, Matrix Factorization, Deep Learning

Received 29 July 2022, **Revised** 21 August 2022, **Accepted** 5 September 2022

*** Corresponding Author** Ha-young Oh (E-mail: hyoh79@gmail.com, Tel:+82-2-583-8585)
Associate professor, College of Computing and Informatics, Sungkyunkwan University, Seoul, 03063 Korea

Open Access http://doi.org/10.6109/jkiice.2022.26.9.1293 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

1.1. 개요

오늘날 우리들은 정보 통신 기술의 발전과 인터넷을 통해 이전보다 훨씬 쉽고 빠르게 수많은 정보들을 접할 수 있다. 이러한 발전과 함께, 사용자들이 상품을 구매하는 패턴 또한 빠르게 변화하고 있다[1]. 불과 몇 년 전만 해도, 소비자들은 대부분 매장에 직접 방문하여 물건을 보고 체험한 뒤 물건을 구매하였으나 최근에는 스마트폰, 태블릿 PC 등 각종 스마트 기기의 발달로 인터넷 쇼핑이 생활화되어 언제 어디서나 간편하게 물건을 구매할 수 있게 되었다.

인터넷 쇼핑을 통해 소비자들은 편리함을 느끼기도 하지만, 다양한 플랫폼들의 등장과 정보의 범람으로 인해 자신이 원하는 물건을 구매하는데 어려움을 느낀다[2]. 이것은 고객들이 느끼는 주관적 탐색 비용의 증가로 이어지며, 향후 해당 서비스를 꾸준히 사용하지 않을 수도 있다. 판매자 입장에서 개개인의 선호도를 고려해 적절한 아이템을 추천해주고 이를 구매로 연결하는 것은 소비자의 탐색 비용을 줄여줄 수 있고, 이윤 창출 효과도 얻을 수 있다. 이러한 이유들로, 요즘 추천시스템의 중요성이 대두되고 있다.

개인화 추천은 각 개인의 관심사나 선호도를 분석하여 이에 맞는 정보나 제품을 추천해주는 것을 의미한다[3]. 최근 사용자에 대한 보다 자세한 정보를 수집할 수 있게 되면서 개인화 추천이 점점 더 널리 쓰이고 있다. 개인화된 추천시스템은 사용자 데이터, 구매, 등급 및 다른 사용자와의 관계를 좀 더 상세하게 분석함으로써 각 사용자에게 맞춤형 추천을 제공한다. 이 중 가장 인기 있는 유형의 개인화된 추천시스템은 콘텐츠 기반과 협업 필터링 방식이다[4]. 본 논문에서는 시중에서 서비스되는 온라인 쇼핑몰의 데이터를 활용하여, 협업 필터링, 행렬 분해, 딥러닝 기반 추천시스템을 구축하고 사용자별 맞춤형 추천을 제공한다.

본 논문의 구성은 다음과 같다. 제2장에서는 추천시스템의 종류, 유사도 계산 지표, 한계에 대해 살펴본다. 제3장에서는 본 논문에서 사용한 데이터세트에 대한 개요, 데이터 전처리 과정, 전처리 이후 분석 및 추천시스템 구현 기술을 자세히 설명한다. 제4장에서는 제3장에서 구축한 추천시스템들에 대해 RMSE 값을 바탕으로 성능평가를 진행하고, 가장 적합한 모델을 식별하여 해당 모델의 추천 성능을 평가한다. 제5장에서는 결론 및 이후 나아갈 방향에 대해 언급한다.

1.2. 관련 사례

넷플릭스의 추천시스템 알고리즘은 일반적인 영화의 그룹으로 배열된 영화들 자체 정보, 고객의 등급, 대여된 영화 및 현재 대기 행렬, 모든 Netflix 사용자의 합산 등급 요소들을 고려한다[5]. 아마존의 추천시스템은 고객의 쇼핑 선호도를 지능적으로 분석하고 예측해 추천 상품 목록을 제공한다[6]. 국내 기업 중 당근마켓의 추천시스템은 다양한 사용자 정보를 활용하고, 실시간으로 제품을 빠르게 추천하기 위해 딥러닝 추천시스템 기반으로 추천 엔진을 개발했다[7]. Choudhury et al[8]은

Table. 1 Comparison table of the previous work [9-11]

title	published	characteristic	dataset	limitations
Multimodal trust based recommender system with machine learning approaches for movie recommendation	2021	- trust and resemblance - trust based filter - check that the user is a cold user	- input of trust based filter model is producer-consumer trust matrix	- Diagonal cells as a consumer can't rely upon its own rating producing power
Deep Neural Networks for YouTube Recommendations[9]	2016	- three major perspectives: scale, freshness, noise - two-stage approach: one for candidate generation and one for ranking	- YouTube video datas	- Very large cardinality ID spaces are truncated by including only the top N after sorting based on their frequency in clicked impressions
Deep Learning Recommendation Model for Personalization and Recommendation Systems[10]	2019	- using the embedding tables to map a categorical feature to a dense representation - continuous & categorical features	- Generating a vector of random numbers using either a uniform or normal distributions - The Criteo AI Labs Ad Kaggle and Terabyte datasets (each data contains 13 continuous and 26 categorical features)	- The results in training times up to several weeks or more - The size of embeddings makes it prohibitive to use data parallelism
Matrix Factorization Techniques for Recommender Systems[11]	2009	- latent factor model - stochastic gradient descent - inputs with varying confidence levels - temporal dynamics	- Netflix Prize data	- more items and users slow down learning - SGD techniques cannot be parallelized

협업 필터링 기반 추천시스템의 문제점인 콜드 스타트, 데이터 희소성, 악의적인 공격을 해결하기 위해 trust and resemblance between the user 기법을 제안한다. 추천시스템 관련 추가적인 정보는 아래 표 1에서 확인할 수 있다.

Ⅱ. 이론적 배경

2.1. 추천시스템 종류

2.1.1. 협업 필터링

협업 필터링에는 사용자 기반 필터링 방법과 아이템 기반 필터링 방법이 있다. 사용자 기반 필터링 방법은 많은 사용자들로부터 얻은 선호도 정보에 따라 사용자들의 제품에 대한 평가 패턴이 비슷한 소비자들을 찾고, 해당 집합의 소비 패턴을 수치화하여 결과를 예측한다[3, 12]. 페이스북, 링크드인 등의 SNS 친구 추천 서비스가 해당 방법을 사용한다.

아이템 기반 필터링 방법은 사용자들 대신 평가된 아이템이 파라미터로 사용된다[13]. 해당 기법은 대부분의 사람들이 과거에 자신이 좋아했던 상품과 비슷한 상품이면 좋아하는 경향이 있고, 반대로 싫어했던 상품과 비슷한 상품이면 싫어하는 경향이 있다는 점에서 착안했다. 해당 기법은 사용자 간의 유사도를 전혀 고려하지 않기 때문에, 전혀 선호도가 비슷하지 않은 사용자들의 평가를 기반으로 추천하는 경우 상품들의 상관관계 정확도가 떨어진다는 단점이 있다.

2.1.2. 행렬 요인화 (Matrix Factorization, MF)

행렬 요인화 (Matrix Factorization, MF)는 평가 데이터로 구성된 하나의 행렬을 2개의 행렬인 사용자 잠재요인 행렬, 아이템 잠재 요인 행렬로 분해하는 방법이다[3, 11]. 행렬 요인화 방식은 모델 기반 추천 알고리즘으로, 데이터로부터 추천을 위한 모델을 구성한 후 해당 모델만 저장하고, 이후 저장한 모델을 사용해 추천한다. 행렬 요인화 알고리즘은 다음 그림 1과 같이 나타낼 수 있다.

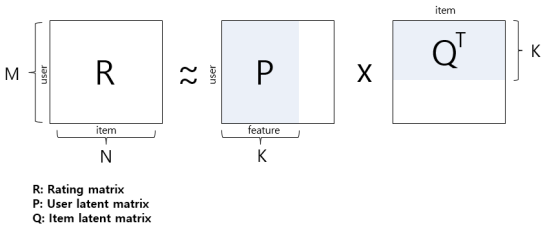

Fig. 1 Matrix Factorization

위 그림에서 R은 M명의 사용자가 N개의 아이템에 대해 평가한 데이터를 포함하고 있는 2차원 행렬로, 행렬의 각 원소는 해당 사용자의 해당 아이템에 대한 평점이다. 이 행렬은 사용자가 실제로 평가한 아이템에 대한 평점만 가지고 있어 많은 원소가 비어있는 NULL 값을 가진다.

행렬 R을 사용자 행렬(P)와 아이템 행렬(Q)로 쪼개어 분석하는 것이 MF 방식이며, P는 (M x K) 차원을 가지고, Q는 (N x K) 차원을 갖는다. P x Q^T인 R'은 R의 예측치이며, R'이 최대한 R에 가까운 값을 가지도록 하는 P와 Q을 구하면서 최적의 추천 값을 찾는다. P는 각 사용자의 특성을 나타내는 K개 요인의 값으로 이루어진 행렬, Q는 각 아이템의 특성을 나타내는 K개의 요인 값으로 이루어진 행렬로 볼 수 있으며, 여기서 K를 잠재요인이라고 부른다.

2.1.3. 내용(콘텐츠) 기반 필터링

내용 기반 필터링은 콘텐츠의 특성과 사용자의 선호도를 비교해 추천해주는 방식이다[3, 14, 15]. 해당 기법은 콘텐츠를 설명하는 카테고리를 자세한 요소들로 분리한다. 이후 사용자가 기존에 좋아한 콘텐츠가 어떤 특징을 가지는지 확인한다. 이를 통해 사용자가 선호할만한 제품을 추천해 줄 수 있다. 해당 기법은 사용자 행동 정보가 많이 필요하지 않은 대신, 다양한 분야의 항목을 추천하기 어렵다. 예를 들어 음악과 영화의 경우 얻을 수 있는 정보가 서로 달라 음악 데이터를 바탕으로 영화를 추천하기 어렵다.

2.1.4. 지식 기반 필터링

지식 기반 필터링은 와인, 커피 등과 같이 보다 전문적인 지식을 요구하는 분야에서 사용되며, 해당 분야에 대한 전문가를 필요로 한다. 지식 기반 필터링의 주요 장점은 cold-start 문제가 없다는 것으로, 아파트나 자동차와 같이 제품을 너무 자주 구매하지 않는 복잡한 영역에 매우 적합하다.

2.1.5. 딥러닝

딥러닝 기반 추천 기술은 보통 신경망의 입력으로 다양한 사용자와 아이템의 특징값을, 출력으로 제품에 대한 개별 사용자의 선호도를 사용하며, 이미지 분야에서

좀 더 좋은 성능을 보인다.

2.1.6. Hybrid

Hybrid 추천 기술은 하나의 추천 알고리즘만 적용하는 것이 아닌 두 가지 이상의 기술을 혼합해서 추천하는 방식이다. 추천시스템은 대부분 한 가지 기술만 사용하지 않고, 여러 가지 기술을 혼합해서 제품을 추천한다.

2.1.7. GNN (Graphic Neural Network)

GNN은 그래프에 적용하는 신경망 구조로, 예측하고자 하는 값들을 그래프 점 사이의 관계를 통해 표현한다[3, 16]. 주로 연결 관계와 이웃들의 상태를 이용하여 각 점의 상태를 학습해 갱신하고, 마지막 상태를 통해 예측을 수행한다. GNN은 컴퓨터 비전, 자연어처리, 스마트 교통 시스템 등 다양한 분야에서 활용된다.

2.2. 추천시스템 유사도 지표

2.2.1. 상관계수

평가 자료가 연속 값일 때 사용하는 유사도로, 사이의 관계를 통해 표현한다. 한 변수의 변화에 따른 다른 변수의 변화 정도와 방향을 예측하는 분석 기법이다. 상관계수는 아래 식 (1)과 같다.

$$r = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2 \times \sum (y_i - \overline{y})^2}} \quad (1)$$

위 수식에서 x, y는 유사도 계산 대상이 되는 두 사용자이고, x_i, y_i는 두 사용자가 공통으로 평가한 아이템 중에서 i번째 아이템에 대한 두 사용자의 평점 값이다. 상관계수는 -1(완전 반대) ~ 1(완전 일치)까지의 값을 가진다.

2.2.2. 코사인 유사도

각 제품을 하나의 차원으로, 사용자의 평점 값을 좌표로 설정하고, 각 사용자의 평점 값을 벡터로 하여 벡터 간의 각도(코사인 값)를 구하는 기법이다. 상관계수와 마찬가지로 평가 자료가 연속 값일 때 사용하며, 사용자 간의 평점 값이 유사할수록 각도가 작다는 것을 이용해 유사도를 측정한다. 코사인 유사도의 계산식을 일반화하면 다음 식 (2)와 같다.

$$\cos(\theta) = \frac{A \cdot B}{||A|| \, ||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (2)$$

위 수식에서 A, B는 상관계수의 x_i, y_i 값과 의미하는 바가 일치하며, 마찬가지로 –1(완전 불일치) ~ 1(완전 일치)의 값을 가진다.

2.2.3. 타니모토 계수 (Tanimoto Coefficient)

데이터가 이진값(binary)을 갖는 경우에 사용하는 지표로, 타니모토 계수는 아래 식 (3)과 같다.

$$Simi\ larity(x, y) = \frac{c}{a + b + c} \quad (3)$$

위 수식에서 a는 사용자 x가 구매한 제품의 수를, b는 사용자 y가 구매한 제품의 수를, c는 사용자 x, y가 공동으로 구매한 제품의 수를 의미한다. 타니모토 계수는 0 (완전 불일치) ~ 1(완전 일치)의 값을 가지며, 이진수 데이터에 대해 좋은 성능을 보인다.

2.3. 추천시스템의 한계

2.3.1. Cold-Start Problem

시장에 기존에 없던 서비스나 제품이 새롭게 등장하는 경우 발생하는 문제다[17]. 이전에 참고할 만한 자료나 데이터가 없어 추천시스템 구축에 한계가 있으며, 구축했다 하더라도 좋은 정확도를 보이기는 어렵다.

2.3.2. Privacy preserving Recommendation System

개인 사용자에게 맞는 제품을 추천해주기 위해서는 사용자 정보를 수집하는 것이 중요하다. 하지만 최근 애플의 개인정보 보호법 강화 등으로 인해 이러한 정보를 수집하기 쉽지 않다[18].

2.3.3. Long-term and short-term user preference

개인 또는 그룹의 단·장기 관심사가 다를 수 있으며, 추천받고 싶은 아이템이 당장 필요한 것인지, 몇 년 후에 필요한 것인지 구분하기 어렵다[19].

2.3.4. The Long Tail Model

Long-tail 그래프는 시장에서 평점 또는 제품의 인기 분포를 보여준다. 추천시스템은 이들 중 인기가 높은 제품 10%에 대해서만 집중적으로 추천해주는 경향이 있다[20]. 특정 제품들만 계속 추천하는 경우 다른 제품들은 외면받는 문제가 발생한다.

Ⅲ. 연구 방법

본 연구는 Fun&C가 제공한 강아지대통령 사이트의 데이터를 대상으로 한다[21]. 해당 데이터세트의 사용자 후기, 상품 정보, 견종별 구매 상품 등을 활용하여 최적의 사료 추천 모델을 제시하는 것을 목표로 한다.

3.1. 데이터 설명

본 연구의 연구 대상은 강아지대통령 사이트에서 수집한 데이터세트이다. 이 데이터세트는 크게 구매 후기 테이블, 주문 테이블, 상품 테이블로 구성되어 있으며, 테이블 별로 중요 칼럼들은 아래 표 2에 정리하였다.

Table. 2 Data table

Table name	Type	Columns
goods_review	review	order_item_sno, goodsno, points, contents, dogname, m_no, pet_age
order	order	ordno, m_no, deliveryno, delivery, ... etc
order_item	order	sno, ordno, goodsno, ccode, goodsnm, ... etc
items	Item	goodsno, goodsnm, origin_no, ... etc
item_category	Item	sno, goodsno, category
category	category	sno, category, catnm

goods_review 테이블은 구매 후기 테이블로, 주문한 상품에 대한 고객의 후기, 평점, 주문 번호, 반려동물 타입, 고객 번호 등으로 구성되어 있다. order_item 테이블은 주문 번호, 상품 쿠폰 번호, 주문 상품 타입, 주문 상품명 등으로 구성되어 있다. order 테이블은 주문 번호, 고객 번호, 운송장 번호, 배송비 등으로 이루어져 있다. items 테이블은 상품번호, 상품명, 원산지 정보, 상품 크기 표, 상품 타입으로 구성되어 있다. item_category 테이블은 카테고리와 상품 정보를 연결하는 역할을 한다. category 테이블은 카테고리 번호, 카테고리 이름 등으로 구성되어 있다.

3.2. 데이터 전처리

협업 필터링, 행렬 분해 등의 다양한 방식의 추천시스템을 구현하기 위해 온라인 쇼핑몰 데이터에 대해 전처리 작업을 진행하였다. 데이터 전처리 과정을 알고리즘으로 나타내면 다음과 같다.

Algorithm 1 Data Preprocessing

```
USE TABLE gd_goods_review
DROP NULL
IF goods NOT EXISTS gd_goods (TABLE):
   DROP datas (raw)
DROP DUPLICATES (mno, goodsno)
IF NUM(ratings) < 5:   # 5: Kfold Num
   DROP datas  (raw)
CREATE TABLE dog_rating
   COLUMNS mno, goodsno, point
CREATE TABLE dog_user
   COLUMNS mno, pet_age,  dogname
```

추천시스템의 입력으로 전달할 데이터를 만들기 위해, 우리는 items 테이블의 goodsno 칼럼과 goodsnm 칼럼을 사용해 새로운 테이블 dog_item을 생성하였다. 또한, goods_review 테이블에는 items 테이블에 존재하지 않는 상품들에 대한 정보가 존재하였고, 이러한 데이터들은 무결성을 해치기 때문에 모두 삭제했다.

goods_review 테이블 중 중복 데이터를 제거하고, 제품을 5개 이하로 구매한 고객들은 전체 데이터세트에서 제외하였다. 이후 해당 데이터세트의 고객 번호, 상품번호, 평점 칼럼을 이용하여 새로운 데이터세트 dog_rating을 생성하였다. 마지막으로 사이트를 이용하는 사용자의 애완동물 정보가 있는 데이터세트를 만들기 위해 고객 번호, 반려동물 나이, 반려동물 종류 칼럼을 사용하여 dog_user 테이블을 생성하였다.

3.3. 데이터 분석

goods_review 테이블의 반려동물 타입 데이터를 이용해, 강아지대통령 사이트를 가장 많이 이용하는 반려동물 종류 상위 10개를 막대그래프로 시각화하면 다음 그림 2와 같다. 말티즈를 기르는 사용자가 사이트를 가

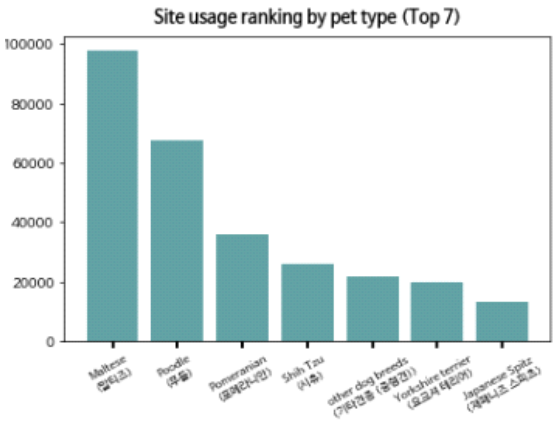

Fig. 2 Site usage ranking by pet type

장 많이 이용하고, 그다음으로 푸들, 포메라니안, 시츄를 기르는 사용자 순으로 많이 이용하는 것을 확인할 수 있다.

goods_review 테이블의 카테고리 데이터를 이용해, 강아지 대통령 사이트에서 가장 많이 판매되는 상품 종류 상위 7개를 막대그래프로 시각화하면 다음 그림 3과 같다. 껌 종류가 가장 많이 판매되고, 저키/트릿, 사사미, 건식사료 순으로 많이 판매되는 것을 확인할 수 있다.

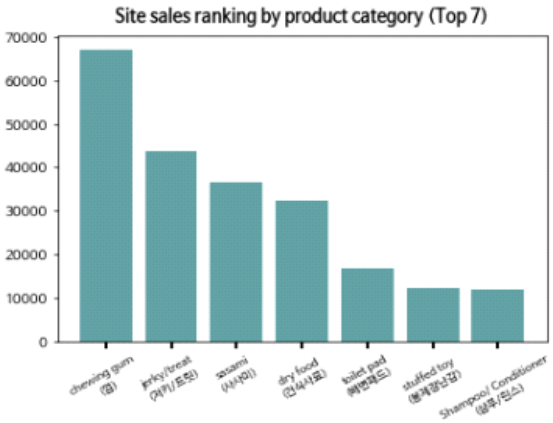

Fig. 3 Site sales ranking by product category

3.4. 추천시스템 구축하기

3.4.1. 협업 필터링 기법을 이용한 제품 추천

기본적인 협업 필터링 알고리즘은 이웃을 전체 사용자로 하며, 현재 사용자와 취향이 비슷한 사용자 그룹을 따로 선정하지 않고 모든 사용자의 평점을 가지고 예측을 수행한다. 해당 알고리즘으로 사용자별 아이템 평점 데이터세트 (dog_data)에 대한 예측치를 계산한 결과, RMSE 값 0.956을 얻었다.

사용자 중에서 유사도가 높은 사용자(이웃)를 선정해서 그 사람들의 평점만 가지고 예측을 수행할 수도 있다. 이웃을 정하는 기준은 이웃의 크기를 미리 정해놓고 추천 대상 사용자와 가장 유사한 K명을 선택하는 'K Nearest Neighbors (KNN)' 방법과 이웃의 크기 대신 유사도의 기준을 정해놓고 기준을 만족하는 사용자들을 이웃으로 정하는 Thresh holding 방법이 있다. 본 논문에서는 KNN 방법으로 이웃을 선정하였고, 이웃 수를 30으로 선정하여 예측치를 계산한 결과 RMSE 값 0.954를 얻었다.

추천의 정확도를 최대로 높여주는 최적의 이웃 크기를 찾기 위해 이웃 크기를 10~200까지 10단위로 변화시키면서 가장 좋은 성능을 보이는 K 값을 찾아보았다. K에 따른 RMSE 값 변화를 그래프로 나타내면 다음 그림 4와 같다.

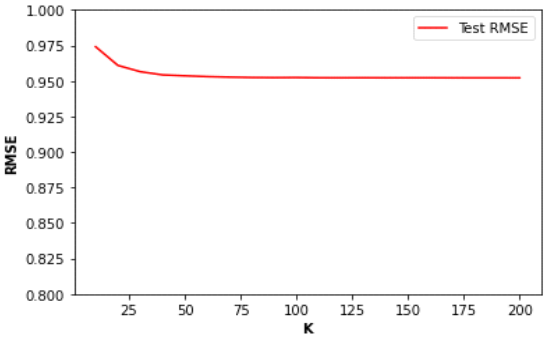

Fig. 4 RMSE values according to change in K values (Collaborative Filtering)

그래프에 의하면, K=90으로 설정할 때 RMSE 값이 0.9139로 가장 좋다. 이후 우리는 K값을 80~100까지 1단위로 변화시키며 가장 좋은 성능을 보이는 K를 탐색해보았다. 해당 구간에서 RMSE 값은 0.9141 ~ 0.9139로 유지되었으며, 이에 최적의 K 값을 90으로 설정한다.

협업 필터링은 사용자-제품 평가 행렬을 기반으로 제품을 추천해주는데, 사용자에 따라 평가 경향이 다르기에 이를 고려하면 좀 더 정확한 추천을 해 줄 수 있다. 평가 경향을 고려하기 위해, 예측치를 계산하는 수식을 다음 식 (4)와 같이 바꾸어 보았다. 식 (4)에서 a는 사용자, u는 이웃 이용자, n은 이웃 사용자 수, $P_{a,i}$는 아이템 i에 대한 사용자 a의 예상 평점, $w_{a,u}$는 사용자 a와 u의 유사도, $r_{u,i}$는 아이템 i에 대한 사용자 u의 평점을 의미한다. 또한 r_a는 사용자 a의 전체 평점 평균을, r_u는 사용자 u의 전체 평점 평균을 의미한다.

$$P_{a,i} = \overline{r_a} + \frac{\sum_{u=1}^{n} w_{a,u} \times (r_{u,i} - \overline{r_u})}{\sum_{u=1}^{n} w_{a,u}} \quad (4)$$

사용자의 평점 평균을 구하고, 각 아이템의 평점을 각 사용자 평균에서의 차이(평점 편차)로 변환한다. 평점 편차를 사용해 해당 사용자의 해당 아이템의 편차 예측값을 구하고, 이를 현재 사용자의 평균에 더해서 실제 예측값을 구할 수 있다. 사용자의 평가 경향을 고려해

계산한 예측치의 RMSE 값은 0.754로, 이전보다 훨씬 개선된 것을 확인할 수 있다.

협업 필터링의 정확성을 좀 더 높이기 위해, 본 연구에서는 신뢰도 가중 기법을 적용하였다. 공통으로 평가한 제품의 수가 3개 이상, 해당 제품을 평가한 사용자가 10명 이상인 사용자와 제품에 대해 예측을 진행하였고, 더 나아가 예측값이 1보다 작은 경우 1로, 5보다 큰 경우 5로 조정한 후 RMSE 값을 계산하여 최종적으로 RMSE 값 0.739를 얻었다. 다른 모델들과의 비교를 위해, 해당 모델을 CF_tuned라고 부르며, 해당 모델에 대한 알고리즘은 다음과 같다.

Algorithm 2 Collaborative Filtering

```
READ Data: Data includes point, mno, goodsno
SPLIT Data INTO Train, Val, Test dataset
REPEAT FOR Kfold_num:
  CREATE Rating Matrix
  CALCULATE User Similarity
  SET Neighbor Size
  IF Goodsno IN Rating Matrix:
    Prediction = DOT(User Similarity, Item Ratings) /
          SUM(User Similarity) + 0.0001
    Prediction += MEAN(User Ratings)
RECOMMEND Items
```

앞서 언급한 협업 필터링 기법들은 모두 사용자 기반 CF, UBCF에 해당한다. 아이템 기반 CF(IBCF)도 적용하여 RMSE 값을 계산해보았을 때, RMSE 값은 0.812가 나왔다. 해당 모델도 이후 다른 모델들과의 비교를 위해 IBCF_tuned라고 부르기로 한다.

3.4.2. 행렬 분해를 이용한 제품 추천

행렬 분해의 핵심은 주어진 사용자, 제품의 관계를 가장 잘 설명하는 P, Q 행렬을 분해하는 것이다. 잠재 요인의 개수인 K를 정하고, 주어진 K에 따라 P(M x K)와 Q(N x K) 행렬을 만들고 임의의 수로 초기화한다. 분해한 P, Q 행렬을 이용하여 예측 평점 R'을 구하고, 예측 평점과 실제 평점을 비교하여 구한 오차를 줄이기 위해 P, Q 값을 수정한다.

행렬 분해 기법을 사용자별 아이템 평점 데이터세트(dog_data)에 적용하고, 잠재 요인의 수를 20으로 설정한 후 SGD를 이용한 오차 업데이트를 100번 진행했을 때 RMSE 값으로 0.775를 얻었다. 행렬 분해 작동 알고리즘에 대해 나타내면 다음과 같다.

Algorithm 3 Matrix Factorization

```
READ Data: Data includes point, mno, goodsno
CREATE Rating Matrix
MAP User ID -> User Idx, Item ID -> Item Idx
SET num of Latent Factor
DIVIDE Rating Matrix INTO User Matrix, Item Matrix
GET User bias, Item bias
REPEAT FOR Kfold_num:
  REPEAT FOR Iterations:
    Predictions = DOT(User Matrix, Item Matrix.T)
      + User bias + Item bias + Rating mean
    UPDATE User bias, Item bias, User Matrix,
        Item Matrix USING SGD
RECOMMEND Items
```

잠재 요인의 수인 K를 크게 설정하는 경우 다양한 패턴을 학습할 수 있기 때문에 정확도가 증가하는 반면, 지나치게 크게 설정하는 경우 과적합이 발생한다. 본 연구에서는 최적의 K 값을 찾기 위해 K를 20부터 300까지 20씩 증가시켜가며 RMSE 값을 비교하고, 다음 그림 5에 나타내었다. 대부분의 RMSE 값은 0.76 ~ 0.78 사이에 분포하고 있으며, K=80 에서 RMSE=0.776으로 가장 낮은 값을 보인다. 이에 우리는 최적의 K값을 80으로 설정한다.

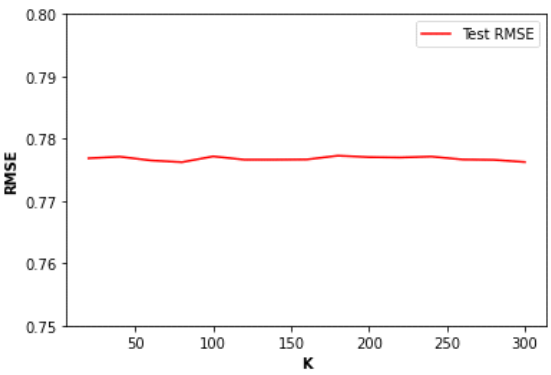

Fig. 5 RMSE values according to change in K values (Matrix Factorization)

최적의 학습률과 정규화 계수를 구하기 위해 학습률은 0.001 ~ 0.01 사이의 값을 0.001 간격으로 비교하고, 정규화 계수는 0.1 ~ 0.5 사이의 값을 0.1 간격으로 비교해 보았다. 그 결과 학습률은 0.008, 정규화 계수는 0.1

로 설정할 때 RMSE 값이 0.758로 가장 낮은 값을 갖는 것을 확인하였다. 학습률을 0.008, 정규화 계수를 0.1로 설정하여 학습 횟수별 RMSE 값을 시각화하면 다음 그림 6과 같다.

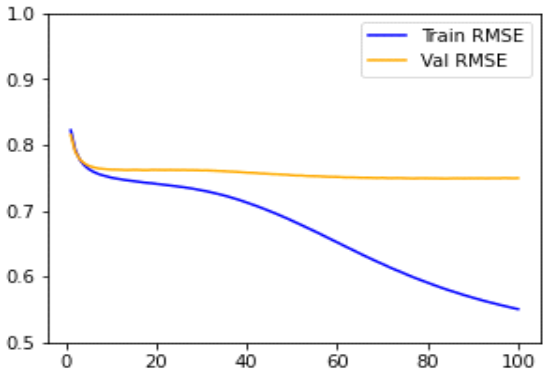

Fig. 6 RMSE values according to the number of epochs (Matrix Factorization)

다른 모델들과의 비교를 위해, 해당 모델을 MF_tuned라고 부르기로 한다. MF_tuned 모델을 사용하여 98380 사용자에게 제품을 추천한 결과는 그림 7에 나타내었다.

```
goodsno
42047          코리아펫쇼 코펫 VIP 초청장 1매
56228            로얄캐닌 미니 스타터 8.5kg
20801      아일오브독스 로얄젤리 샴푸 NO.20 1L
6112               럭셔리독 양고기 캔 100g
82285     캐니대 독 그레인프리 퓨어씨 연어 5.4kg
Name: goodsnm, dtype: object
```

Fig. 7 Top-5 Product Recommendations

3.4.3. 딥러닝을 이용한 제품 추천

본 연구에서는 행렬 요인화 모델을 신경망으로 변환하여 반려동물 사이트의 제품을 추천하도록 모델을 구축해보았다. 구축한 모델 구조는 그림 8과 같다.

생성한 모델을 학습하고, 학습 횟수별 RMSE 값을 시각화하면 그림 9와 같다.

딥러닝을 이용한 추천시스템을 구축하기 위해서는, 앞의 모델 구조에 은닉층을 추가해주면 된다. 은닉층이 많을수록, 다양한 패턴을 학습할 수 있어 보다 정확한 추천을 할 수 있다. 다음 그림 10은 본 연구에서 사용한 딥러닝 모델의 구조를 나타낸 것이다.

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 1)]          0           []
input_2 (InputLayer)            [(None, 1)]          0           []
embedding (Embedding)           (None, 1, 200)       326600      ['input_1[0][0]']
embedding_1 (Embedding)         (None, 1, 200)       1226400     ['input_2[0][0]']
dot (Dot)                       (None, 1, 1)         0           ['embedding[0][0]',
                                                                  'embedding_1[0][0]']
embedding_2 (Embedding)         (None, 1, 1)         1633        ['input_1[0][0]']
embedding_3 (Embedding)         (None, 1, 1)         6132        ['input_2[0][0]']
add (Add)                       (None, 1, 1)         0           ['dot[0][0]',
                                                                  'embedding_2[0][0]',
                                                                  'embedding_3[0][0]']
flatten (Flatten)               (None, 1)            0           ['add[0][0]']
==================================================================================================
Total params: 1,560,765
Trainable params: 1,560,765
Non-trainable params: 0
__________________________________________________________________________________________________
```

Fig. 8 Basic Neural-Network model structure

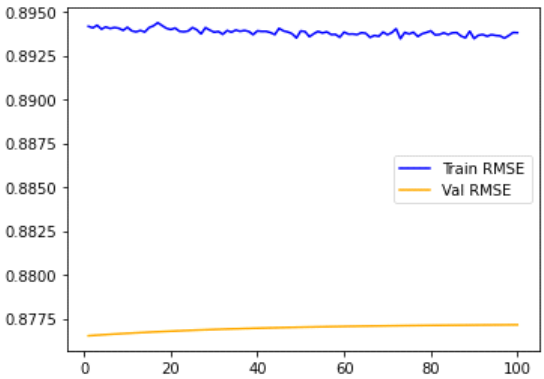

Fig. 9 Train & Test RMSE values according to number of epochs (basic NN)

```
Model: "model_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_3 (InputLayer)            [(None, 1)]          0           []
input_4 (InputLayer)            [(None, 1)]          0           []
embedding_4 (Embedding)         (None, 1, 200)       326600      ['input_3[0][0]']
embedding_5 (Embedding)         (None, 1, 200)       1226400     ['input_4[0][0]']
embedding_6 (Embedding)         (None, 1, 1)         1633        ['input_3[0][0]']
embedding_7 (Embedding)         (None, 1, 1)         6132        ['input_4[0][0]']
flatten_1 (Flatten)             (None, 200)          0           ['embedding_4[0][0]']
flatten_2 (Flatten)             (None, 200)          0           ['embedding_5[0][0]']
flatten_3 (Flatten)             (None, 1)            0           ['embedding_6[0][0]']
flatten_4 (Flatten)             (None, 1)            0           ['embedding_7[0][0]']
concatenate (Concatenate)       (None, 402)          0           ['flatten_1[0][0]',
                                                                  'flatten_2[0][0]',
                                                                  'flatten_3[0][0]',
                                                                  'flatten_4[0][0]']
dense (Dense)                   (None, 2048)         825344      ['concatenate[0][0]']
activation (Activation)         (None, 2048)         0           ['dense[0][0]']
dense_1 (Dense)                 (None, 256)          524544      ['activation[0][0]']
activation_1 (Activation)       (None, 256)          0           ['dense_1[0][0]']
dense_2 (Dense)                 (None, 1)            257         ['activation_1[0][0]']
==================================================================================================
Total params: 2,910,910
Trainable params: 2,910,910
Non-trainable params: 0
__________________________________________________________________________________________________
```

Fig. 10 Deep Neural-Network model structure

생성한 모델을 학습하고, 학습에 따른 RMSE 값을 시각화하면 다음 그림 11과 같다. 해당 모델의 RMSE 값을 은닉층이 없는 신경망과 비교했을 때, 확실히 값이 개선된 것을 볼 수 있다.

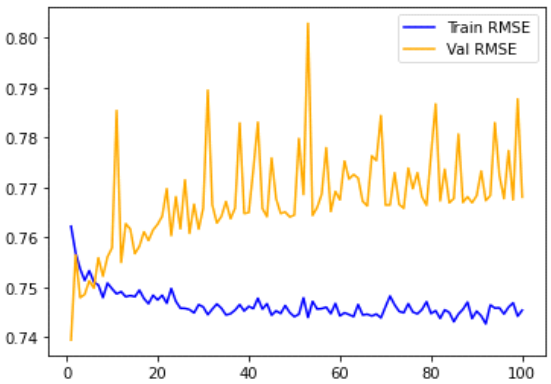

Fig. 11 Train & Test RMSE values according to number of epochs (Deep NN)

우리는 해당 모델을 다른 모델들과 비교하기 위해, DL_basic이라고 부르기로 한다. 해당 모델에 대한 알고리즘은 다음과 같다.

Algorithm 4 Deep Learning

READ Data: Data includes point, mno, goodsno
MAP User ID -> User Idx, Item ID -> Item Idx
SET Num of Latent Factor, Num of User Embedding,
  Num of Item Embedding
DEFINE Model(User, Items) -> Predictions
REPEAT FOR Kfold_num:
  REPEAT FOR Iterations:
    TRAIN Model
    VALIDATE Model
    Predictions = Model(User, Item)
RECOMMEND Items

DL_basic 모델을 사용하여 사용자번호가 98380인 사용자에게 제품을 추천한 결과는 아래 그림 12와 같다.

```
goodsno
492                브리더 크롬 철장 중
1436       미라클 순 트리트먼트 샴푸 750ml
3527         굿프랜드 슬라이스 사사미 30g
4000    오~사사미 웰빙 고구마 치킨 져키 400g
3814       미스터두크 하네스&리드줄 블루그린 중
Name: goodsnm, dtype: object
```

Fig. 12 Top-5 Product Recommendations

Ⅳ. 성능평가

추천 모델들을 만든 후에는 어떤 모델을 사용할 것인지 결정하고, 최적의 매개변수를 설정하기 위해 모델 중 일부나 매개변수 설정에 따른 추천 성능을 테스트한다.

4.1. 데이터 분할

모델을 평가하기 위해서는 데이터 중 일부로 모델을 만들고 나머지로 테스트해야 하며, 모델의 추천 결과와 사용자 선호도(평점)를 비교해야 한다. 이를 위해 테스트 데이터세트의 사용자별 데이터는 무시하고 학습 데이터세트의 데이터를 기반으로 추천 결과를 생성해야 한다. 본 연구에서는 학습과 테스트 데이터세트의 비율을 0.8: 0.2로 나눈다.

4.2. 가장 적합한 모델 식별

성능 지표는 다양한 모델 또는 매개변수를 비교하는 데 유용하게 사용된다. 같은 데이터에 여러 기법을 적용하고 성능 지표를 비교하면 가장 적합한 추천 방식을 선택할 수 있다. 다양한 평가 지표가 존재하며, 이중 어떤 평가 지표를 선택할지는 우리의 몫이다.

4.2.1. 모델 비교

서로 다른 모델을 비교하기 위해 먼저 모델을 정의해야 한다. 본 연구에서는 MF_tuned 모델을 기본으로 설정하고, 기본 모델과 모델들의 성능을 비교하기 위한 모델로 CF_tuned 모델, IBCF 모델, DL_basic 모델, 이웃 수를 설정하지 않은 협업 필터링 모델을 선정하였다.

각각의 모델들에 대해 RMSE 값을 시각화하면 다음 그림 13과 같다. 여러 모델 중 MF_tuned 모델의 RMSE 값이 가장 작은 것을 볼 수 있다.

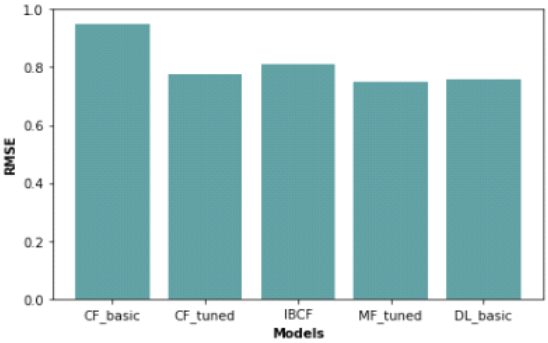

Fig. 13 Model performance comparison

4.2.2. 가장 적합한 모델 식별

4.2.1에서 계산한 모델별 평가 지표 값들을 사용하여 ROC 곡선을 나타내는 도표를 만들어 모델을 비교한 결과는 그림 14와 같다[22].

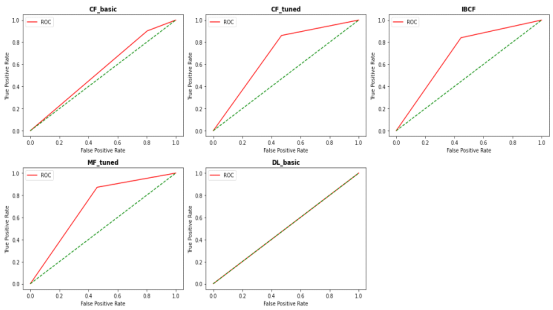

Fig. 14 ROC-Curve for each model

좋은 성능을 보이는 모델일수록 그래프가 가지는 면적이 커지게 되므로 AUC (Area Under the Curve), 즉 ROC 곡선 아래 면적으로 성능을 알 수 있다. 위 그래프를 보면 CF_basic, DL_basic 모델을 제외한 모델들은 비슷한 AUC 값을 가지는 걸 알 수 있다.

4.2.3 추천 결과 평가

추천 결과들을 긍정 평점(평점 값이 4점 이상인 데이터)들과 비교해 혼동 행렬을 계산하는 방법으로 추천 모델의 정확성을 측정할 수 있다.

MF_tuned 모델에 대해 사용자별로 추천할 아이템의 수를 10, 20, 30, … 50개로 늘려가며 혼동 행렬을 만들고, 아이템 수별 정밀도(precision), 재현율(recall), FPR (False Positive Rate), TPR(True Positive Rate)을 계산하여 다음 표 3에 나타내었다.

Table. 3 MF_tuned model performance assessment

num_items	asccuracy	precision	recall (TPR)	FPR
10	0.534	1.0	0.333	0.0
20	0.457	0.0	0.0	0.157
30	0.481	0.0	0.0	0.071
40	0.5	0.5	0.375	0.094
50	0.506	0.6	0.25	0.053

Ⅴ. 결 론

5.1. 연구 결과 토의

본 연구에서는 산업체 데이터의 상품 구매 후기 데이터를 활용하여 협업 필터링, 행렬 분해, 딥러닝을 이용한 제품 추천시스템을 구현하고, 모델들 각각의 성능을 비교하였다. 데이터를 제공해 준 산업체에서 추천시스템을 구축할 경우, 본 연구에서 가장 성능이 좋게 나온 MF_tuned 모델을 사용하는 것이 권장된다.

5.2. 개선 사항

딥러닝을 이용한 추천시스템의 경우, 입력으로 (제품번호, 사용자번호)를 전달받아 출력으로 입력값에 대한 예상 평점을 반환하도록 구현되었다. 해당 모델을 사용하여 특정 사용자에 대해 예상 평점이 가장 높은 제품 n개 추천을 진행할 경우, 많은 시간이 소요된다. 이후 연구에서는 딥러닝 모델을 사용한 제품 추천 시 속도를 개선하는 방법들을 추후 더 조사해 볼 예정이다.

5.3. 확장 가능성

본 연구에 사용된 데이터세트의 구매 후기 작성 일자를 분석하면 시간의 흐름에 따른 온라인 쇼핑몰 평점 평균 변화를 살펴볼 수 있다. 이와 같은 시계열 정보를 활용하여, 시간의 흐름에 따라 평가 경향이 변화된 사용자들을 분석하고, 개인화 맞춤 시스템을 보다 고도화할 수 있을 것으로 기대된다.

더 나아가, 본 논문에서 제안하는 모델 아키텍처를 다른 온라인 쇼핑몰에 적용하여 제품을 추천해보고, 추천 성능을 비교 분석해 볼 수 있을 것으로 기대된다.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2022 R1F1A1074696).

References

[1] J. Son, S. B. Kim, H. Kim, and S. Cho, “Review and Analysis of Recommender Systems,” *Journal of Korean Institute of Industrial Engineers*, vol. 41, no. 2. pp. 185 – 208, Apr. 2015.

[2] S. Y. Cho, J. E. Choi, K. H. Lee, and H. W. Kim, “An Online Review Mining Approach to a Recommendation System,” *Information Systems Review*, vol. 17, no. 3, pp. 95 – 111, Dec. 2015.

[3] I. Im, *Personalization Recommendation System Using Python*, 1st ed. Seoul: CRbooks, 2020.

[4] Introduction to recommender systems [Internet]. Available: https://thingsolver.com/introduction-to-recommender-systems/.

[5] How Netflix Works. electronics [Internet]. Available: https://electronics.howstuffworks.com/netflix2.htm.

[6] Amazon's Product Recommendation System In 2021: How Does The Algorithm Of The eCommerce Giant Work? [Internet]. Available: https://recostream.com/blog/amazon-recommendation-system.

[7] Recommend Deep Learning Personalization [Internet]. Available:https://medium.com/daangn/%EB%94%A5%EB%9F%AC%EB%8B%9D-%EA%B0%9C%EC%9D%B8%ED%99%94-%EC%B6%94%EC%B2%9C-1eda682c2e8c.

[8] S. S. Choudhury, S. N. Mohanty, and A. K. Jagadev, “Multimodal trust based recommender system with machine learning approaches for movie recommendation,” *International Journal of Information Technology*, vol. 13, no. 2, pp. 475 – 482, Jan. 2021.

[9] P. Covington, J. Adams, and E. Sargin, “Deep Neural Networks for YouTube Recommendations,” in *Proceedings of the 10th ACM conference on recommender systems*, New York: NY, USA, pp. 191-198, 2016.

[10] M. Naumov, D. Mudigere, H. M. Shi, J. Huang, N. Sundaraman, J. Park, X. Wang, U. Gupta, C. Wu, A. G. Azzolini, D. Dzhulgakov, A. Mallevich, I. Cherniavskii, Y. Lu, R. Krishnamoorthi, A. Yu, V. Kondratenko, S. Pereira, X. Chen, W. Chen, V. Rao, B. Jia, L. Xiong, and M. Smelyanskiy, “Deep Learning Recommendation Model for Personalization and Recommendation Systems,” *arXiv preprint arXiv:1906.00091*, May. 2019.

[11] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer*, vol. 42, no. 8, pp. 30-37, Aug. 2009.

[12] H. Wang, Z. Shen, S. Jiang, G. Sun, and R. J. Zhang, “User-based Collaborative Filtering Algorithm Design and Implementation,” in *Journal of Physics: Conference Series*, Changsha, China, vol. 1757, no. 1, p. 012168, 2021.

[13] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proceedings of the 10th international conference on World Wide Web*, New York: NY, USA. pp. 285 – 295, 2001.

[14] The Concept and Application of the Recommended Algorithm and the Patterns of Development [Internet]. Available: https://www.kocca.kr/trend/vol20/sub/s21.html.

[15] What Content-Based Filtering is and Why You Should Use It [Internet]. Available: https://www.upwork.com/resources/what-is-content-based-filtering#:~:text=Content%2Dbased%20filtering%20is%20a,them%20to%20a%20user%20profile.

[16] Recommendation system using knowledge graph [Internet]. Available:https://zzaebok.github.io/knowledge_graph/recommender_syste/KG_recommend/.

[17] The Cold Start Problem for Recommender System [Internet]. Available: https://medium.com/@markmilankovich/the-cold-start-problem-for-recommender-systems-89a76505a7.

[18] The Ethical and Privacy Issues of Recommendation Engines on Media Platforms [Internet]. Available: https://towardsdatascience.com/the-ethical-and-privacy-issues-of-recommendation-engines-on-media-platforms-9bea7bcb0abc.

[19] K. Sun, T. Qian, T. Chen, Y. Liang, Q. V.H. Nguyen, and H. Yin, “Where to Go Next: Modeling Long- and Short-Term User Preferences for Point-of-Interest Recommendation”, In *Proceedings of the AAAI Conference on Artificial Intelligence*, New York, USA. pp. 214-221, 2020.

[20] Recommender Systems: What Long-Tail tells ? [Internet]. Available: https://medium.com/@kyasar.mail/recommender-systems-what-long-tail-tells-91680f10a5b2.

[21] Homepage of the dogpresident [Internet]. Available: https://dogpre.com/.

[22] Draw your own ROC Curve and Precision-Recall Curve [Internet]. Available: https://yangoos57.github.io/ml/data_viz/roc_curve_and_pe_re_curve/.

오재동 (Jaedong Oh)

성균관대학교 인공지능융합학과 석사
※ 관심분야: 추천시스템, NLP, 챗봇

오하영 (Ha-young Oh)

Sungkyunkwan University Professor (2019~)
Ajou University Professor (2016~2019)
Soongsil University Professor (2013~2016)
Ph.D. in computer engineering at Seoul National University (2013)