Outer Product Convolutional DeepCF Using Side Information

윤성식(Seongsik Yoon) 국민대학교 AI빅데이터융합경영학과
이성규(Seongkyu Lee) 국민대학교 AI빅데이터융합경영학과
김보현(Bohyeon Kim) 국민대학교 AI빅데이터융합경영학과

ABSTRACT

추천 시스템은 개인의 Item에 대한 선호를 추론해 개인이 더 선호할 것 같은 Item을 필터링해 제공하는 알고리즘이다. 그 중에서도 협업 필터링(Collaborative Filtering)은 다양한 논문에서 차용되고 발전되는 만큼 성능 면이나 활용 면에서 성공적인 기법이다.

본 연구에서는 이 같은 Collaborative Filtering을 기반으로 한 DeepCF 아키텍처를 발전시켜 소비자의 Item 선호 정보와 소비자의 부가정보, Item의 정보 등을 폭넓게 추천 시스템에 반영할 수 있는 Outer Product Convolutional DeepCF with Side Information을 제안한다.

먼저, 기존 DeepCF의 Aggregation Function인 Concatenation이나 Element-wise와 다르게 소비자와 아이템 간의 상관관계를 충분히 반영할 수 있도록 Outer Product 방식을 이용해 2D Interaction map을 만든다. 또한 데이터의 Side Information을 Embedding 한 값을 활용해 하나의 Feature Map을 생성한다. 최종적으로는 만들어진 Feature map들을 Convolutional Layer를 통과시켜 예측에 활용한다.

도서 데이터를 이용해 제안 모델의 성능과 기존 DeepCF 모델의 성능을 비교 실험하였으며, 제안 모델이 근소하게 앞서는 것을 확인하였다. 이로써 여러 데이터에 대해 텍스트 정보와 이미지 정보를 폭넓게 활용할 수 있는 방향을 제시했다.

1. 서론

빅데이터 시대가 도래함에 따라 소비자들은 본인이 선택한 제품만이 아니라 다양한 종류의 제품을 빠르게 경험해 볼 수 있게 되었다. 소비자들은 인터넷 검색을 통해 많은 정보를 접할 수 있지만, 반대로 정보 과다 현상으로 선택이 어려워지는 경우도 존재한다. 이러한 정보 과다 현상을 어떻게 해결할 것인지가 매우 중요한 연구주제로 대두되고 있는데, 기업들은 시대 상황에 발맞추어 이를 해결하기 위해 소비자의 의사결정을 도와주는 추천 서비스들을 앞다투어 도입하고 있다.**[1]** 추천 시스템의 알고리즘 중 Collaborative Filtering을 성공적으로 사용한 대표적인 비즈니스 모델로는 OTT 서비스 기업인 Netflix와 온라인 쇼핑 사이트인 Amazon이 있다. 두 기업 모두 고객이 직접 남긴 평가와 상품에 관련되어 남긴 모든 정보를 바탕으로 각 고객에 맞는 제품을 추천해 주는데, 해당 추천이 비교적 정확하여 호평을 받았다.**[2]**

이렇게 Collaborative Filtering을 활용한 선례를 바탕으로 하여 본 논문에서는 Deep Neural Network를 기반으로 한 Collaborative 방식을 발전시킨 형태인 DeepCF Architecture에 대해 살펴보고 이를 더욱 발전시킬 방안을 모색한다.

DeepCF 추천 시스템에서 사용되는 데이터는 각 Item에 대한 사용자의 선호를 나타내는 피드백 데이터이다. 그중에서도 암시적 피드백(Implicit Feedback)을 사용하는데, 이는 평점이나 만족도 등과 같이 Item에 대한 User의 직접적인 평가를 다루는 명시적 피드백(Explicit Feedback) 과는 달리 구매 이력, 방문 기록 등 선호를 간접적으로 나타내는 데이터를 의미한다.**[3]**.

DeepCF Architecture에서는 Implicit Data를 통해 User와 Item의 상호작용 정보를 나타낼 수 있는데, 해당 부분에서 DeepCF Architecture를 발전시킬 수 있는 첫 번째 가능성을 발견할 수 있다. Implicit Data만을 사용하면 Cold start problem에 직면할 수 있는데 이를 User나 Item에 대한 Side Information을 활용하여 초기 표현을 개선함으로써 해소할 수 있다. 두 번째로 DeepCF 논문의 Improvements단락**[4]**에서도 볼 수 있듯 해당 아키텍처에서 사용한 Aggregation Function인 Element-wise Product 방식과 Concatenation 방식은 각각의 Embedding Vector 간의 Correlation을 충분히 고려하지 못하기 때문에 이를 개선할 수 있는 가능성이 보인다.

따라서 본 논문에서는 해당 두 가지 가능성을 구체화하여 실험한다.

2. 관련 연구

2.1. DeepCF

기존 추천 시스템은 User와 Item의 의미적 차이가 크기 때문에 초기 표현 공간에서 사용자가 선호할 아이템을 예측하는 것에 어려움을 겪었다. 이를 해결하기 위해 DeepCF에서는 새로운 방식을 제시한다. 보통의 추천 시스템에서 사용자의 Ratings

정보와 아이템의 Ratings 정보는 하나의 층을 통과시켜 결과값을 도출해 내는 것이 일반적이다. 하지만 이러한 방법이 의미적인 차이를 좁힐 수 없기 때문에, User의 Ratings 값과 Item의 Ratings 값을 따로 MLP Layer들을 통과시켜 각각 User의 Latent Factor와 Item의 Latent Factor를 얻는다. 이를 Element-wise 방식으로 결합시켜 최종 Predictive Vector를 얻게 된다. 이 또한 의미적인 차이를 좁힐 수 없지만, User의 Ratings와 Item의 Ratings를 각자 학습을 시켜 여러 Layer를 통과시키기 때문에 더 Dense 한 결과를 얻을 수 있다. 이 방식을 Representation Learning이라고 한다.

Representation Learning 방식을 보완하기 위해, 사용자의 Ratings와 아이템의 Ratings를 선형함수를 통해 Latent Factor를 먼저 도출한 뒤, 이를 바로 Concatenation 시키고 MLP Layer에 통과시켜 의미적인 차이를 좁혀 Predictive Vector를 도출한다. 이러한 방식을 Matching Function Learning이라고 하며, 이 두 방식을 최종 Concatenation 시키고 Fusion 해 Sigmoid 함수를 적용하여 최종 Matching Score를 얻는다.

이러한 방식으로 학습이 진행되는 DeepCF의 한계점은 Representation Learning에서 Element-wise를 사용하고 Latent factor를 활용함에 있어 Concatenation을 적용하기 때문에 표현력의 한계가 존재한다는 점이다. 또한 Implicit Data만을 사용하기 때문에 새로운 Item이나 user 정보가 추가되면 추천에 한계가 생기는 Cold Start problem에 직면할 수 있다.**[4]** 이와 같은 DeepCF의 2가지 한계점을 보완하기 위해 ONCF와 Deep Contextual Modeling을 참조했다.

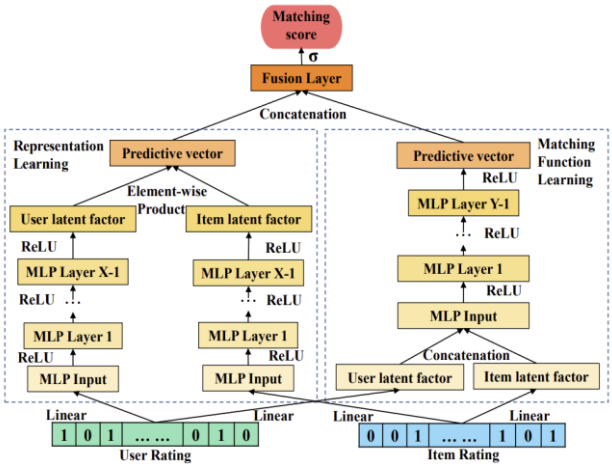

그림1. DeepCF Architecture

2.2. ONCF

ONCF(Outer product-based Neural Collaborative Filtering)는 기존의 NCF를 보완하여 Input 단계의 Sparse 한 User Ratings와 Item Ratings를 Dense 벡터로 각각 매핑하여 Embedding 값으로 반환하고, 이 두 Embedding Vector를 외적(Outer Product) 하여 얻은 2차원 구조의 Feature map(2D Interaction map)에 CNN을 사용한 구조다. Collaborative Filtering을 수행하기 위한 새로운 다층 신경망 구조를 제안하여, 외적을 사용하여 Concatenation 하거나 Element-wise을 하였을 때보다 Embedding 차원 간의 더 많은 상관관계를 포착할 수 있으며, 다층의 CNN을 통해 Embedding 차원 간의 고차 상관관계를 학습하는데 용이하다.**[5]**

외적을 통해 나온 Interaction map의 대각 성분은 Matrix Factorization(MF)의 내적과 합성곱 방식의 결과를 포함한다. 대각선 이외의 성분은 MF와 NCF에서 다루지 않는 Different Embedding Dimension Correlation을 다뤄 더 많은 Signal 정보를 담는다. 그리고 Interaction map의 Correlations을 이미지의 Local feature로 여겨 CNN Hidden layer를 적용하기 때문에 더 많은 유용한 Signal 정보를 담을 수 있다.

따라서 본 논문에서는 단순한 연결이나 합성곱 방식보다 Embedding 차원 간의 더 많은 상관관계를 포착할 수 있는 Outer Product 방식을 사용하여 2D Interaction map을 도출하여 연구에 적용했다.

2.3. Deep Contextual Modeling

기존의 추천 시스템 연구는 Explicit data나 Implicit data의 user와 item 간 선호를 바탕으로 학습이 진행되었다. 하지만 데이터가 Sparse 해서 정보가 충분하지 않기 때문에 Cold start problem을 겪고 있었다. 해당 논문은 다양한 보조 데이터들이 생성되고 사용할 수 있게 됨에 따라 이 같은 문제점을 해결하기 위해 보조 데이터(Auxiliary Data)를 사용할 수 있는 모델을 제시하였다. Side Information을 사용함으로써 얻는 장점은 두 가지이다. 첫 번째로, Side Information은 새로운 표현을 설계하고 추천 시스템을 통합시키는데 사용할 수 있다. 영화의 맥락과 같은 고유한 속성을 Side Information의 새로운 표현으로 혼합하여 순위와 등급을 예측하는 작업의 정확도를 높일 수 있다. 두 번째로, Side Information을 추천 시스템에 통합해 성능을 향상시키면서 해석성을 보존하거나 제한된 양의 데이터를 사용하면서 발생하는 문제를 해결할 수 있다는 점이다.

Deep Contextual Modeling에서는 MLP Layer를 거친 User Vector와 Item Vector와 함께, Side Information을 Embedding 한 Context Vector가 Concatenation되어 MLP Layer를 통과하게 된다. 이때, GMF Layer에는 User Vector와 Item Vector가 Element-wise를 거친 후 통과되게 되고, Context Vector는 MLP Layer를 거쳐 만들어진 MLP User Vector와 MLP Item Vector와 함께 Concatenation되어 MLP Layer를 통과해 값을 도출한다. 이때 GMF Layer를 거쳐 나온 값을 함께 Concatenation 하고 NeuMF Layer를 통과시켜 최종 Matching Score를 얻게 된다. 이로써 User와 Item 간 상관관계와 맥락 정보가 포함된 Prediction을 도출할 수 있게 된다.**[6]**

해당 논문의 Side Information을 활용하는 방식을 참고하여, 본 논문에서는 Data set에서 파악할 수 있는 전반적인 Side Information의 종류를 활용할 수 있도록 한다. 다양한 Side Information을 활용할 수 있는 일반화된 구조를 제안함에 따라 앞서 언급한 Cold start problem을 해결하고자 한다.

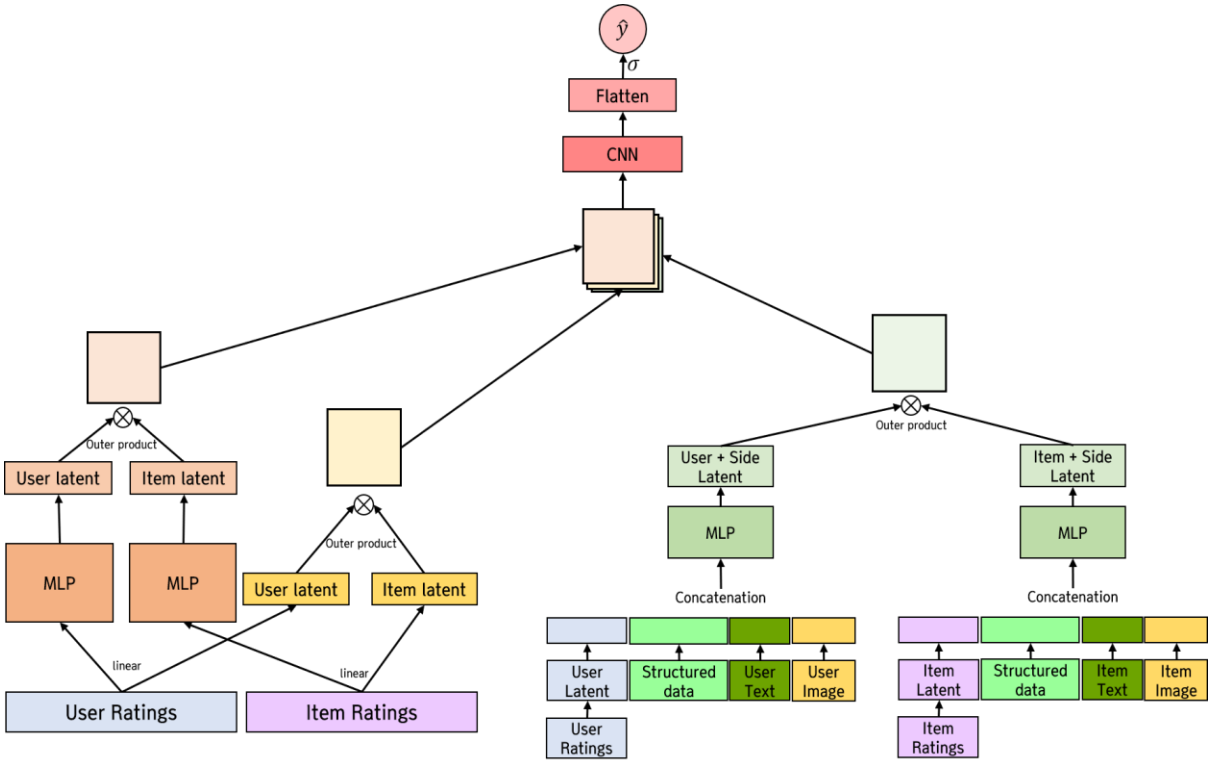

그림2. Proposal Architecture

3. 모델 구조

기존 DeepCF의 Element-wise와Concatenation 방식을 Outer Product로 변환하여 Embedding 간의 상관관계를 고려할 수 있는 2D Interaction map을 도출하는 것이 본 논문의 첫 번째 목적이다. Representation Learning 파트에서 User Ratings와 Item Ratings를 선형변환한 후 MLP Layer를 거쳐 User Latent와 Item Latent를 도출해 Element-wise 방식이 아닌 Outer Product를 사용하여 2D Interaction map을 만든다. 또한 기존 모델에서는 Matching Function Learning 부분에서 User와 Item Latent factor를 Concatenation 한 뒤 MLP Layer를 통과시켰다. 이를 수정하여 User Ratings와 Item Ratings를 선형변환한 뒤, User Latent와 Item Latent를 생성한 후 Outer Product를 진행해 도출한 2D Interaction map을 채널로 활용한다.

다음, 본 논문의 두 번째 목표는 Side Information을 수용할 수 있는 모델을 만드는 것이다. Side Information이 입력되는 파트는 User Latent에 User와 관련된 Structured Data, Text Data, Image Data 3종류의 Side Information을 Concatenation 하고, Item Latent에도 동일하게 Item과 관련된 Structured Data, Text Data, Image Data 3종류의 Side Information을 Concatenation 한다.**[7]** 다음 각각 MLP 층을 거친 뒤 도출된 Latent들을 Outer Product 하여 2D Interaction map을 만든 후, 이를 채널로 활용한다.

앞선 과정들을 거치면 총 3개의 Channel이 도출되는데, 이를 단순하게 Concatenation 시키는 것이 아닌, Channel로 활용하고자 이들을 Stack 한 후 CNN 층을 통과시키고 Flatten 하여 최종적으로 $\hat{y}$을 도출한다.

4. 실험 및 결과

4.1. 실험 및 평가 방법

본 연구에서 제안한 Model Architecture를 실험하기 위해 사용한 데이터 셋은 Kaggle Book Data이다. 데이터 셋은 User-ID, ISBN(Item), Ratings으로 이루어져 있고, 이에 대응되는 Side Information은 Book-Title, Book-Author, Year-of-Publication, Publisher, Image-URL-L로 이루어져 있으며, 이미지 데이터를 사용하기 위해 약 10만 건의 데이터를 크롤링 하여 사용하였다. 본 실험 환경에서 사용할 수 있는 용량이 한정되어 구매 건수가 10건 이상인 User의 정보만을 활용하였으며, 결과적으로 User는 총 9,571명, Item은 약 100,083건의 데이터를 가지고 실험을 진행하였다. Train 데이터의 개수는 약 197,303개, Test 데이터의 개수는 User의 수와 동일한 9,571개이다. Side Information으로 활용한 정보는 User의 경우, User rating은 Linear layer를 통과시켜 Latent vector 화 했고, Structured Data인 Location 데이터를 Embedding 하여 사용했다. Item의 경우 Item rating은 User rating과 동일하게 Linear layer를 통과시켰고, 책 제목, 저자, 출판사와 같은 Text Data는 FastText 모델을 활용해 Embedding 했으며 책 표지 이미지 데이터는 얕은 CNN layer를 사용해 Embedding 했다.

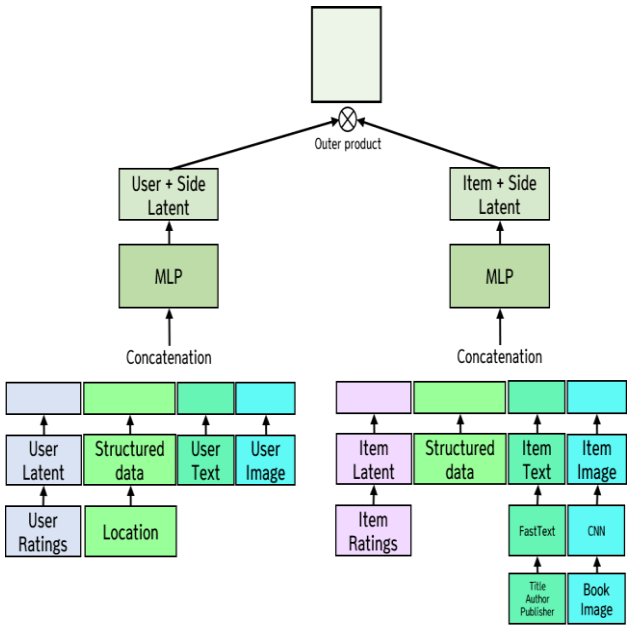

그림3. 데이터 사용 그림 세부사항

4.2. 실험 결과

실험은 총 두 가지로 진행되었다. 첫 번째로, 기존 DeepCF 모델에 비해 앞서 제시한 Outer Product와 CNN을 활용한 모델의 성능을 비교하는 실험을 진행하였다. 두 번째로, 기존 DeepCF 모델과 실험 1에서 구축한 모델에 Side Information까지 포함시킨 최종 모델의 성능을 비교하는 실험을 진행하였다.

성능 평가 지표는 기존 참조 논문과 같이 순위 정보를 포함하지 않는 Hit Rate와, 순위 정보를 포함하는 NDCG를 사용하였다.

Experiment	Metrics	Fusion Model
DeepCF	HR	0.4728
	NDCG	0.2725
Experiment 1 (with Outer Product & CNN)	HR	0.4816 (0.0088 ↑)
	NDCG	0.2840 (0.0115 ↑)
Experiment 2 (with Side Information)	HR	0.4836 (0.0108 ↑)
	NDCG	0.2995 (0.0270 ↑)

표1. 실험 성능 지표

표 1에서 실험 1의 결과를 보면, 기존 DeepCF보다 제안 모델의 성능이 Hit Ratio 지표를 사용하여 확인했을 때, 0.4728에서 0.4816로 약 1.8% 상승함을 확인했다. 또한 NDCG 지표를 사용하면 0.2725에서 0.2840로 약 4.2% 상승함을 확인했다.

실험 2의 결과는 기존 DeepCF 보다 제안 모델에 Side Information을 추가한 모델이 HR 지표 기준 0.0108 상승한 0.4836로 약 2.2% 상승을 확인했고, NDCG 지표 기준 0.0270 상승한 0.2995로 약 9.9% 상승을 확인했다. 두 실험 모두 기존 DeepCF 모델보다 향상된 성능을 보이고 있어 유의미한 결과를 얻었다고 할 수 있다.

Hyper Parameter Tuning은 Negative Sampling, Learning Rate, Optimizer, Embedding Size를 변경하는 실험을 진행하였다. Negative Sampling을 큰 배수로 설정하면 입력 데이터 셋의 양이 많게 증가해 실행에 제한이 있기 때문에 본 논문의 실험 환경인 Google Colab Pro에서 최대치로 수용할 수 있는 만큼으로 데이터 셋의 양을 늘리기로 결정했다. Negative Sampling을 조절했을 때 실험은 3배수, 4배수, 5배수로 진행하였고 4배수가 가장 좋은 성능을 보여 이를 채택했다. Learning Rate를 조절해 본 결과, 성능의 차이가 극명하게 나 0.01, 0.001, 0.0001, 0.00005, 0.00001로 실험을 진행했고 0.00001이 가장 좋은 성능을 보였다. 0.0001, 0.00005는 비슷한 결과를 보였지만, 0.00001이 근소하게 더 좋은 성능을 보여 이를 채택했다. Batch Size는 256으로 고정했다. Optimizer는 RMSProp, AdaGrad, SGD, Adam으로 실험을 진행하였다. 실험 결과, SGD는 다른 Optimizer 들에 비해 확연히 낮은 성능을 보였고 RMSProp이나 AdaGrad는 Adam과 비슷한 성능을 보였으나 Adam이 더 좋은 성능을 나타냈기 때문에 Adam을 채택했다. User Latent Embedding Size와 Item Latent Embedding Size를 128, 256, 512, 1024로 조정하여 실험을 진행한 결과 256이 가장 좋은 성능을 보였다.

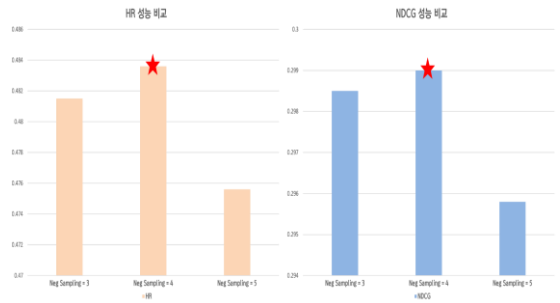

그림4. Negative Sampling 성능 비교

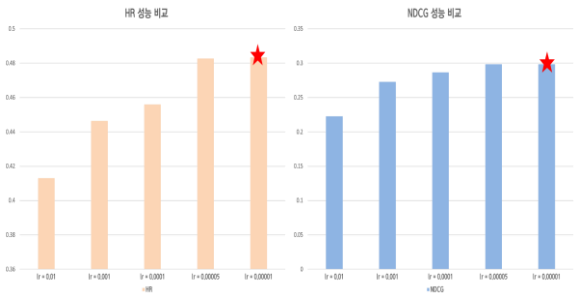

그림5. Learning Rate 성능 비교

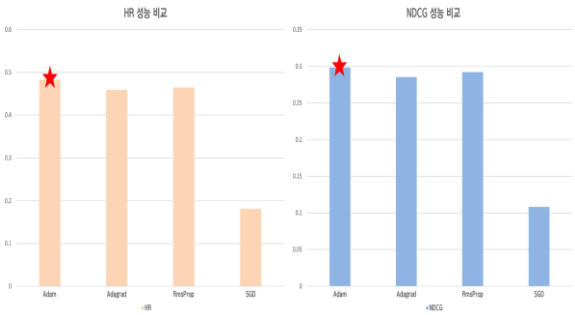

그림6. Optimizer 성능 비교

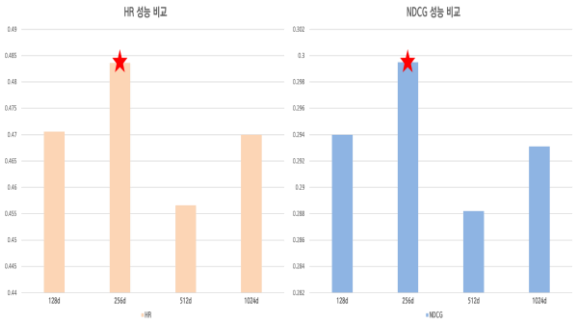

그림7. Latent Embedding Size 성능 비교

5. 결론

5.1 Summary

Collaborative Filtering 방법을 근간으로 한 DeepCF에는 두 가지 학습 방법을 사용해 복잡한 Matching Function을 학습할 수 있는 유연성을 가지고 있다는 장점과, 낮은 순위 관계를 학습할 수 있다는 장점이 있다. 하지만 Element-wise와 Concatenation 같은 제한된 aggregation function의 표현적 한계가 존재하고 Cold start problem이 존재한다.

위와 같은 문제를 개선하기 위해서 모델을 구축하고 실험을 진행 했을때, 실험 1의 결과 기존의 DeepCF 모델에서 사용된 Aggregation Function인 Element-wise와 Concatenation을 Outer Product로 변환하고 2D Interaction Map들을 활용해 CNN에 통과시킨 결과가 모델 성능에 좋은 영향을 끼친다는 사실을 발견할 수 있었으며, Hit Rate와 NDCG가 각각 약 1.8%, 4.2% 상승했다. 또한, User의 Side Information과 Item의 Side Information의 Embedding이 Concatenation된 값으로 2D Interaction Map을 만들어 함께 Channel로 활용한 결과, 기존 DeepCF에 비해 Hit Rate는 약 2.2%, NDCG는 약 9.9% 상승했다. 그 뿐만 아니라, 실험 1 보다도 실험 2의 결과가 0.4%, 5.4% 성능 향상을 보였다.

본 논문에서 제안한 Outer Product Convolutional DeepCF using Side Information는 여러 데이터를 수용할 수 있는 일반화에 초점을 두고 만들어졌으며, 더 다양한 Auxiliary Data를 사용하는 경우에 훨씬 좋은 성능을 보장할 수 있다. 또한 Outer Product를 활용하여 User와 Item 간의 Correlation을 보존해 학습시킬 수 있다는 장점이 존재한다.

5.2 한계점 및 향후 연구

본 논문에서 실험에 사용한 환경은 Google Colab Pro으로, 모델의 무게를 고려해 Convolutional Layer를 얕고 간단하게 쌓았다. 그렇기 때문에 더 유의미한 정보를 도출하는 데에 한계점이 있었을 것으로 추측한다. 따라서 Convolutional Layer를 더 깊게

쌓거나 Kernel Size, Stride를 조정하는 등의 방안을 적용했을 때, 성능 향상의 가능성이 존재한다. 또한, 총 Representation Learning의 2D Interaction Map, Matching Function Learning의 2D Interaction Map, 그리고 Side Information을 통해 만들어진 2D Interaction Map, 총 3가지의 2D Interaction Feature Map들을 Channel로 활용해 Input으로 사용되는 마지막 Convolutional Layer의 조정을 통해 성능을 향상시킬 수 있을 것으로 기대한다. 또한, 각 채널 간의 중요도가 CNN Layer으로는 고려되지 않기 때문에 모델이 무거워지더라도 CBAM(Convolutional Bloc Attention Module)과 같은 기술을 사용해서 Spatial Attention과 Channel Attention 기법**[8]**이 들어가면 각 채널 간의 중요도가 고려될 수 있어 더 좋은 성능을 낼 수 있을 것으로 예상된다.

본 논문에서는 도서 데이터를 사용하였으며, 데이터가 굉장히 Sparse 해 User에 비해 Item의 개수가 상당히 많아 학습하는 것에 어려움이 있었다. 또한 환경에서 모델이 학습할 수 있는 데이터의 양이 제한되어 있기 때문에 데이터 셋과 수집한 Side Information 전부를 사용하지 못했다. 따라서 구매 횟수가 10번 이상인 데이터만 사용하였으며, 데이터 셋 전체를 사용하지 못하고, Colab Pro의 환경 상 최대로 수용 가능한 범위의 데이터의 수만큼 임의로 잘라 사용하였다. 많은 양의 데이터를 수용할 수 있는 환경에서 실험을 진행한다면 더 나은 성능이 기대된다.

그리고 사용한 데이터 셋이 제안한 Side Information 구조에서 User와 Item에 관련된 모든 종류의 Side Information(Structured, Text, Image)을 활용함에 있어 Side Information의 다양성 부족에 따른 제약이 있었다. 모든 종류의 Side Information을 포용할 수 있는 데이터를 사용하여 학습을 시켜본다면 더 나은 성능을 기대할 수 있을 것으로 예상된다.

6. 참고 문헌

1. 박대우, 고인수 외 (2020), 빅데이터 기반 도서추천시스템 구축을 위한 아키텍처에 대한 연구
2. 김영준, 김영휘 외 (2015), 협업 필터링과 데이터마이닝을 통한 도서추천 시스템 제안
3. 최영제, 문현실 외 (2020), Application of Domain Knowledge in Transaction-based Recommender Systems through Word Embedding
4. Zhi-Hong Deng, Ling Huang 외 (2019), DeepCF : A Unified Framework of Representation Learning and Matching Function Learning in Recommender System
5. Xiangnan He, Xiaoyu Du 외 (2018), ONCF : Outer Product-based Neural Collaborative Filtering
6. Amit Livne (2021), Deep Recommender Systems Utilizing Side Information

7. Wei Niu, James Caverlee 외 (2018), Neural Personalized Ranking for Image Recommendation
8. Sanghyun Woo, Jongchan Park 외 (2018), CBAM: Convolutional Block Attention Module
9. 문현실, 임진혁 외 (2020), A Deep Learning Based Recommender System Using Visual Information