

Recommender System Using Wide & Deep Technique With Channel Attention

지윤혁 20172860	국민대학교 AI빅데이터융합경영학과
백찬진 20172834	국민대학교 AI빅데이터융합경영학과
장영수 20172854	국민대학교 AI빅데이터융합경영학과
조기흠 20172856	국민대학교 AI빅데이터융합경영학과

<국문 초록>

IT기술의 발달과 함께 온라인 시장의 규모가 커지고 다룰 수 있는 상품의 수가 많아졌다. 이에 따라 정보 과부하 속에서 소비자가 구매의사결정을 할 때 드는 비용이 증가했다. 추천시스템은 사용자의 선호를 추론하고 그에 맞는 아이템을 필터링하여 제공하는 정보 필터링 시스템이다. 추천시스템을 사용한다면 위와 같은 문제를 어느정도 해결할 수 있다. 딥러닝 기술의 발전이후 추천시스템에도 딥러닝 기술이 도입되었고 사용자와 아이템의 복잡한 관계를 비선형적으로 표현할 수 있게 되면서 많은 성능 개선이 이뤄졌다. 본 연구에서는 이런 딥러닝 기반의 추천시스템인 ONCF를 기반으로 성능 개선을 위해 부가정보를 활용하는 방안을 제안한다. 이때 부가정보를 활용하기 위해 구글에서 제안한 Wide & Deep 모델을 사용한다. Wide & Deep 모델은 실수형 변수와 범주형 변수를 모두 사용하는 모델로 부가정보 활용에 있어 모든 정형화된 데이터를 사용할 수 있다는 장점이 있다. 또한 부가정보를 활용하여 모델의 채널을 확장하게 된다. 이후 확장된 채널간 중요도를 학습에 반영할 수 있게 Channel Attention 기법을 사용하였다. 이를 통해 최종적으로 부가정보간의 중요도를 고려하여 부가정보를 보다 효과적으로 사용할 수 있는 모델을 제안하고자 한다. 실험결과 제안 모델과 기존의 ONCF 성능 비교 결과 부가정보를 활용한 제안 모델이 성능이 ONCF 대비 더 높은 성능을 보였다. 결론적으로 기존 ONCF에 Wide & Deep과 Channel Attention 기법을 통해 부가정보를 보다 효과적으로 활용하여 모델의 성능 향상에 기여하였다.

1. 서론

인터넷의 발달로 온라인 시장의 규모가 커지면서 다루는 상품의 수가 많아졌다. 소비자가

다양한 상품에 접근하기 쉬워졌다는 긍정적인 점도 있지만 한편으로는 많은 상품속에서 소비자가 어느 제품을 구매해야 되는지 고민하는

비용 또한 증가했다. 이로 인해 정보 과부하 속에서 바쁜 현대인의 구매의사 결정을 도와주는 개인화된 추천 시스템이 중요성이 높아지고 있다. 추천 시스템은 소비자뿐만 아니라 기업에게도 매우 중요한 시스템이다. 다양한 상품 중 특정 제품을 구매할 가능성이 높은 소비자에게 그 제품을 추천함으로써 소비자의 구매를 유도하여 매출을 늘릴 수 있다.

추천 시스템은 새로운 기술이 아닌 머신러닝, 딥러닝 기술을 이용한 정보 필터링 기술이다. 추천 시스템은 소비자의 정보와 아이템의 정보 간의 연관성을 찾아 특정 소비자의 선호에 맞는 아이템을 추천한다. 초기의 추천 시스템은 협업 필터링(CF)과 내용 기반 필터링(CBF) 방식의 두 가지였으나, 이후에는 협업 필터링 방식을 기반으로 발전했다. 또한 협업 필터링 방식의 문제점인 Cold-Start 문제점을 보완하기 위해 협업 필터링과 내용 기반 필터링을 혼합해서 사용하는 하이브리드 방식의 시도도 진행되어 왔다. 'Netflix-Prize' 이후 추천시스템은 행렬분해 기법을 기반으로 발전해왔다. 행렬분해 기법은 M 명의 소비자와 N 개의 아이템 열을 가진 평점 행렬 R ($M \times N$)을 사용자 잠재 행렬 P ($M \times k$)와 아이템 잠재 행렬 Q ($N \times k$)으로 분해하여 2개의 잠재 행렬을 곱한 $\hat{R}(P \times Q^T)$ 행렬을 통해 기존 평점행렬을 예측하여 사용자가 아직 시청하지 않았거나 구매하지 않은 아이템에 대한 평점을 예측하는 방법이다. 행렬분해 방법은 우수한 성능을 보여줬지만 소비자와 아이템 사이의 복잡한 관계를 선형적인 관계로만 모델링 한다는 단점을 가지고 있었다. 이후 딥러닝 기술이 발전하면서 딥러닝 기반의 추천 시스템에 대한 연구가 활발히 진행되었다. 사용자와 아이템의 복잡한 관

계를 비선형적 구조를 가진 딥러닝 모델을 통해 전보다 잘 파악할 수 있는 모델들이 등장했다. 대표적으로 ONCF(Outer Product-based Neural Collaborative Filtering) 모델은 사용자 잠재벡터와 아이템 잠재벡터의 외적을 통해 나온 2차원 행렬을 CNN모델(합성곱 신경망)을 통해 상관 관계를 파악하려고 하였다. 하지만 사용자 ID와 아이템 ID, 평점 데이터만을 사용하여 사용자의 직업, 나이, 성별, 아이템의 카테고리, 가격 등의 부가정보를 충분히 활용하지 못하는 점이 단점이었다. 부가정보를 충분히 활용한다면 사용자와 아이템간의 복잡한 관계를 보다 잘 파악할 수 있다. 따라서 본 연구에서는 기존 ONCF 모델을 확장하여 사용자와 아이템의 부가정보를 활용할 수 있는 추천 시스템을 제안하고자 한다. 사용자와 아이템의 상호 관계를 Outer-Product를 통해 파악한 ONCF 모델에 부가정보를 추가하여 기존의 ONCF보다 더 뛰어난 성능의 추천 시스템을 제안하고자 한다.

2. 선행연구

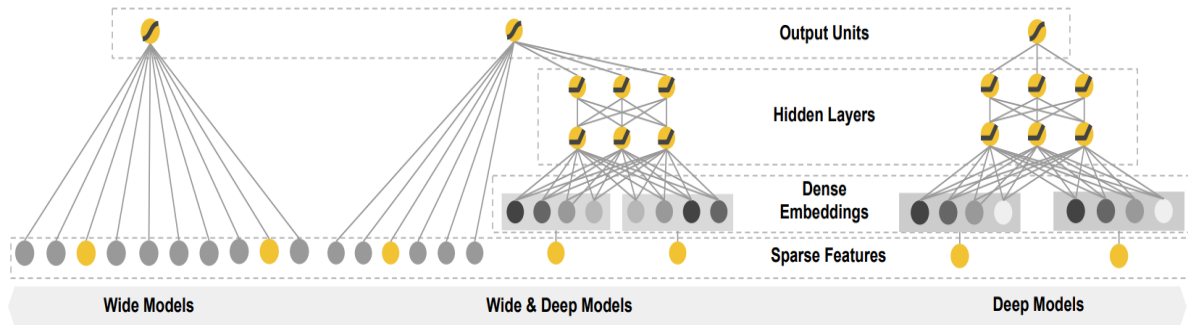
2-1 Wide & Deep

2016년 구글에서 발표한 Wide & Deep 모델은 실제 구글 플레이 스토어에서 앱 추천 시스템으로 사용되는 모델이다. Wide & Deep 논문에서 저자는 Wide 모델은 단순 선형 구조로 암기화(Memorization)에 특화되고 Deep 모델은 일반화(Generalization)에 특화되어 이 둘을 결합하는 방식을 사용한다고 말한다. Wide 모델은 암기에 특화되어 사용자의 선호를 잘 기억해 개인의 선호에 맞는 제품을 추천하지만, 반대로 지나치게 개인의 선호만을 고려하여 한

아이템이 추천된다는 단점이 있다고 말한다. 이와 반대로 Deep 모델은 일반화에 특화되어 다양한 아이템을 추천한다는 장점이 있지만 지나친 일반화로 인해 사용자의 선호와 관련 없는 아이템에 대해서도 추천이 이루어진다는 단점이 있다고 한다. Wide & Deep 모델은 Wide와 Deep 모델의 동시 학습을 진행함으로써 두 모델이 갖는 장점을 통해 서로의 단점을 보완한 모델을 제안한다. 이러한 점에서 Wide & Deep 모델은 추천 시스템에서 중요한 평가 요소인 관련성과 다양성의 측면을 살린 모델이라고 할 수 있다.

<그림 1>은 Wide & Deep 모델 구조이다. 그림의 왼쪽은 Wide 모델이고 오른쪽은 Deep 모델이다. 그리고 가운데는 해당 논문에서 제안하는 Wide & Deep 모델을 표현한다. Wide 모델에서는 cross-product transformation 기법이 들어간다. cross-product의 수식은 다음과 같이 표현된다.

$$\varphi_k(x) = \prod_{i=1}^d x_i^{c_{ki}} \quad c_{ki} \in \{0, 1\}$$



<그림 1> Wide & Deep Framework

출처: Wide & Deep Learning for Recommender Systems (Heng-Tze Cheng, 2016)

$$\mathbf{a}^l = f(W^{l-1}\mathbf{a}^{l-1} + \mathbf{b}^{l-1}) \quad (2)$$

의 식(2)는 Deep 모델을 표현한 식이다. 해당

Cross-Product는 범주형 변수의 조합을 통해서 새롭게 표현된다. A , B , C 앱이 있다고 가정된 상황에서 사용자가 A 앱을 설치했고 C 앱을 봤다면 Cross-Product를 통해 $(A, C) = (1, 1)$ 로 표현될 것이고 이는 A 와 C 앱의 관계를 표현하는 것이다. 만약 만약 A 라는 앱이 여행 관련 앱이고 C 앱이 숙소 예약 관련 앱이라면 사용자는 여행을 좋아한다는 특성을 알 수 있게 된다. 이렇게 범주형 변수들은 cross-product를 거쳐 Wide 모델의 입력 값으로 들어간다. 식(1)은 Wide 모델 구조를 수식으로 표현한 것이다.

$$\mathbf{y} = \mathbf{W}^T \mathbf{x} + \mathbf{b} \quad (1)$$

Wide 모델의 경우 위의 식과 같이 기본적인 선형구조로 표현된다. Deep 모델의 경우는 실수형 변수와 임베딩된 범주형 변수를 모두 입력 값으로 사용한다. 즉, 범주형 변수와 실수형 변수를 모두 사용할 수 있다.

논문에서는 실수형 변수와 임베딩된 범주형 변수를 Concatenate 하여 3개의 ReLU 활성화함수를 통과시킨다. Wide와 Deep 모델은 최종

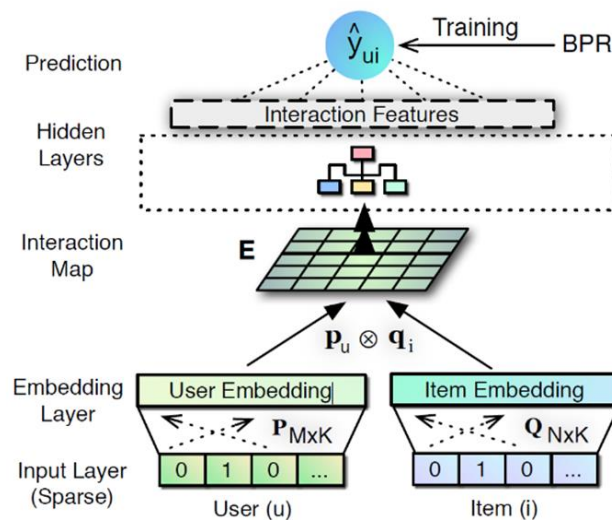
결과 값을 계산하기 위해 Wide 부분과 Deep 부분의 결과 값을 가중합을 하고 같은 손실 함수에 들어간다. 즉, 같은 손실값을 공유하며 역전파 학습이 진행된다고 할 수 있다. 해당 논문의 저자는 이러한 동시 학습법이 기존의 앙상블 기법과는 다르며 동시 학습법을 통해 두 모델은 서로를 고려하며 파라미터들을 동시에 최적화한다고 말한다.

Wide & Deep 모델은 암기화와 일반화를 모두 수행할 수 있는 모델이며 범주형 변수와 실수형 변수를 모두 사용하는 모델이다. 따라서 본 연구에서는 Wide & Deep의 이러한 장점을 활용하여 사용자와 아이템의 부가정보를 Wide & Deep 모델 각각을 통해 학습시킬 계획이다. 이후 학습을 통해 나온 결과를 ONCF의 사용자, 아이템 잠재벡터와 결합하여 ONCF

모델에 부가정보를 포함할 수 있는 모델을 제안하고자 한다.

2-2 NCF, ONCF

ONCF모델은 기존 NCF모델을 한계점을 보완하여 등장한 모델이다. 아래의 <그림 2>는 ONCF모델 구조를 나타낸 것이다. <그림 2>에서 Embedding Layer 이전까지의 과정은 기존 NCF모델과 동일하다. 먼저 사용자ID와 아이템ID를 원핫인코딩(One-hot Encoding)을 통해 사용자와 아이템의 수만큼의 크기를 갖는 벡터로 변환한다. 위 벡터는 대부분의 성분이 0으로 이루어진 희소행렬이다. 이러한 희소행렬을 임베딩(Embedding)을 통해 사용자 잠재 벡터 p_u 와 아이템 잠재 벡터 q_i 로 변환한다..



<그림 2> ONCF Framework

출처: Outer Product-based Neural Collaborative Filtering(Xiangnan He. 2018)

$$p_u = P^T v_u^U \quad P \in R^{M \times k}$$

$$q_i = Q^T v_i^I \quad Q \in R^{N \times k} \quad (3)$$

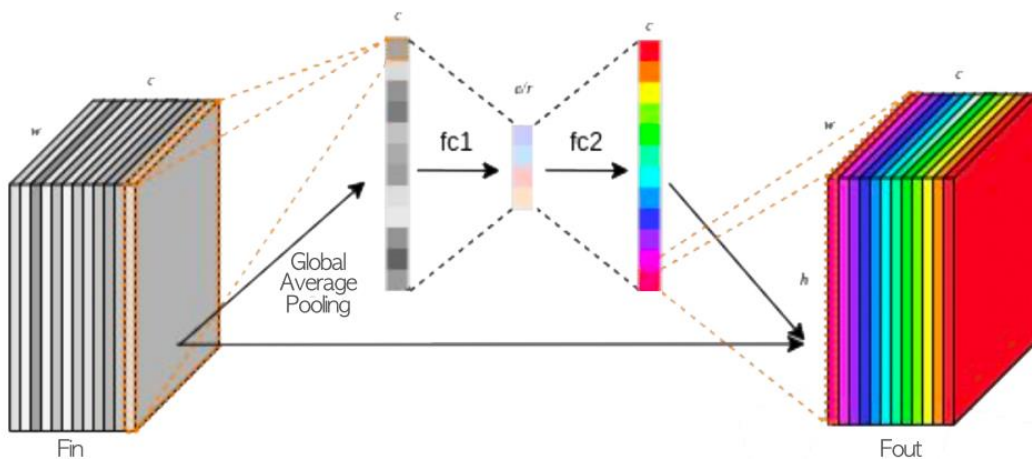
식 (3)은 사용자 잠재 벡터 p_u 와 아이템 잠재 벡터 q_i 를 구하는 연산을 식으로 나타낸 것이다. 임베딩 층에서 위와 같은 연산을 통해 얻은 사용자 잠재 벡터와 아이템 잠재 벡터의 외적을 통해 $(k \times k)$ 크기의 2차원 행렬을 구한다. 해당 논문에서는 이 2차원 행렬을 interaction map이라고 지칭한다. 해당 논문에서는 2차원의 interaction map을 이미지를 처리하는데 우수한 성능을 보이는 CNN모델에 통과시켜 결과 값을 구한다. 사용자와 아이템 잠재 벡터간 내적이 아닌 외적을 했을 때의 장점은 기존에 고려하지 못했던 임베딩 차원 간의 교차 상관 관계를 고려할 수 있다는 점이다. 기존NCF에서는 interaction map의 대각 성분만을 사용했다면 ONCF는 대각 성분을 포함한 나머지 성분 또한 고려하여 여러 차원간 관계를 파악할 수 있다고 주장한다. 또한 CNN모델을 사용하여 기존의 MLP 방식보다 파라미터의 수가 줄

어들어 연산량이 줄어든다는 장점이 있다.

본 연구에서는 기존 ONCF모델의 사용자와 아이템의 잠재 벡터에 부가정보가 포함된 벡터를 결합하여 새롭게 생성되는 2차원 interaction map에 더 많은 정보가 포함시켜, 추천 시스템의 성능 향상을 기대할 수 있을 것으로 판단한다. 이에 ONCF모델에 부가정보를 활용하여 사용자와 아이템의 상관관계를 파악하는데 다양한 정보가 포함될 수 있는 모델을 제안하고자 한다.

2-3 Channel Attention

Attention은 결과값을 도출하는데 특정 단어가 미치는 영향력을 수치화하여 이를 학습에 사용하는 기술이다. 이때, Attention은 중요한 단어에 더 높은 가중치를 곱하여 계산하도록 설정된다. Channel-Attention 기술은 위와 같은 Attention 방식을 이미지를 학습하기에 적합한 CNN모델에 적용한 방식이다.



<그림 3> Channel Attention framework

Channel Attention은 채널 간의 관계를 파악하고 특정 채널의 중요도를 강조한다. 위의

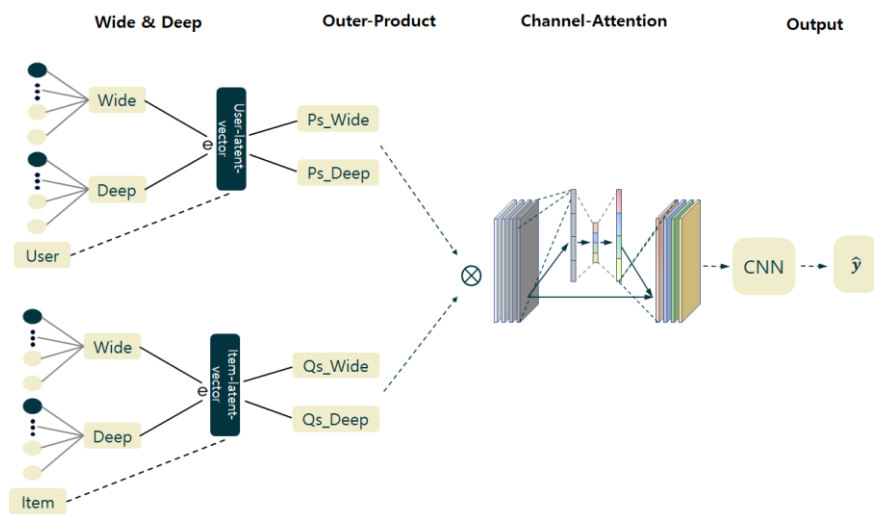
<그림 2>는 Channel Attention의 구조를 표현한 그림이다. 먼저 입력된 2차원의 이미지들은 Global average pooling을 통해 채널 크기의 벡터로 변환된다. 이후 Fully Connected layer를 통과해 해당 벡터들이 압축되어 활성화함수를 거친다. 이때 활성화함수로 보통 ReLU 함수나 Leaky ReLU 함수가 사용된다. 이후 압축된 벡터를 다시 Fully Connected Layer와 시그모이드 활성화함수에 통과시켜 0~1사이의 값을 갖는 벡터를 만든다. 이 벡터는 초기 채널이 Global Average Pooling을 통해 변환된 벡터의 크기와 같지만, 초기 채널이 가지고 있지 않았던 채널 간의 중요성을 벡터 내부에 포함한다는 점에서 차이가 있다. <그림3>은 앞서 설명한 Channel Attention을 그림으로 나타낸 것이다. 초기에 들어오는 채널이 Channel Attention의 과정을 거쳐 채널 간의 중요성을 포함한 벡터로 변환된다.

$$F_{out} = F_{in} \times \sigma_2(f_{c_2}(\sigma_1(f_{c_1}(GAP(F_{in})))))) \quad (4)$$

식 (4)은 위의 과정을 수식으로 표현한 것이다. 본 연구에서는 Wide & Deep 모델을 통해 부가정보를 학습시키고 이를 잠재 벡터와 element-wise한 4개의 벡터를 외적하여 4개의 interaction map을 생성한다. 기존의 ONCF는 하나의 interaction map을 사용하지만 본 연구에서 제안한 모델은 4개의 interaction map을 채널로 쌓기 때문에 이 과정에서 각 채널마다의 중요성을 파악하기 위해 Channel Attention 기법을 사용할 계획이다.

3. 제안모델

본 연구에서 제안하는 추천시스템은 앞서 언급한 Wide & Deep과 ONCF의 장점을 참고하여 결합한 모델이다. Wide & Deep 모델을 사용함으로써 사용하는 부가정보의 유형에 관계없이 다양한 부가정보를 활용할 수 있다. 또, ONCF를 사용함으로써 사용하는 잠재 벡터가 가진 많은 정보를 담을 수 있다는 장점이 있다.



<그림 4> 제안 모델 framework

즉, Wide & Deep 모델을 통해 다양한 부가정보를 활용하고, 이를 ONCF에 적용하여 기존 모델 대비 성능 면에서 더 좋은 모델은 제안하고자 한다. <그림4>는 본 연구에서 제안한 모델의 구조이다. 먼저 ONCF에서 사용된 기존 사용자-아이템 잠재 벡터와 부가정보(Side information) 사이의 상관관계를 고려하기 위해, 부가정보를 벡터로 나타내는 과정이 필요하다. 사용자와 아이템 대한 부가정보를 각각 Wide & Deep 모델의 입력으로 사용하여 이를 사용자부가정보_Wide, 사용자부가정보_Deep, 아이템부가정보_Wide, 아이템부가정보_Deep으로 분리하여 추출하였다. 그리고 이를 간단히 아래의 식 (5)와 같이 표현하였다.

$$Ps_Wide = P \circ User_Side_Wide$$

$$Ps_Deep = P \circ User_Side_Deep$$

$$Qs_Wide = Q \circ Item_Side_Wide$$

$$Qs_Deep = Q \circ Item_Side_Deep \quad (5)$$

본 연구에서는 부가정보의 벡터를 추출하기 위하여 Wide & Deep 모델을 사용한다. P 와 Q 는 각각 사용자 잠재 벡터와 아이템 잠재 벡터를 의미하고 $User_Side_Wide$ 벡터는 사용자의 부가정보를 Wide 모델을 통해 학습시킨 벡터를 의미한다. 식 (5)의 Ps_Wide 와 Ps_Deep 벡터는 사용자 잠재 벡터와 사용자의 부가정보를 2가지 방법으로 학습시킨 벡터 각각을 element-wise 연산을 통해 새로운 벡터로 표현한 것이다. 아이템의 경우도 동일한 방식을 거쳐 새로운 잠재 벡터인 Qs_Wide 와 Qs_Deep 벡터를 구한다. 위의 과정을 거쳐 생성된 각각의 벡터들은 element-wise 연산으로 인해 기존 사용자-아이템에 대한 정보뿐만 아니라, 부가

정보에서 추출한 정보까지도 온전히 포함하고 있다. 즉, 기존의 잠재 벡터에 다양한 부가정보를 활용하여 데이터를 더 풍부하게 하였다고 할 수 있다. 본 연구에서는 이러한 잠재 벡터들 간의 외적을 통해 4가지 interaction map을 만들고, 이를 통해 사용자와 아이템 간의 관계를 파악하고자 한다.

아래의 식 (6)은 새로운 사용자 잠재 벡터와 아이템 잠재 벡터 간의 조합의 경우의 수에 맞춰 가능한 4가지 외적을 표현한 식이다.

$$Map\ 1 = Ps_Wide \otimes Qs_Wide$$

$$Map\ 2 = Ps_Wide \otimes Qs_Deep$$

$$Map\ 3 = Ps_Deep \otimes Qs_Wide$$

$$Map\ 4 = Ps_Deep \otimes Qs_Deep \quad (6)$$

사용자와 아이템의 조합을 고려해봤을 때 가능한 경우의 수는 4가지이다. 이를 통해 interaction map을 의미하는 4개의 2차원 행렬을 구할 수 있다. 본 연구에서는 4개의 map을 모두 사용하고자 한다.

기존의 ONCF 모델은 사용자 잠재 벡터와 아이템 잠재 벡터 간의 외적을 통해 나온 하나의 2차원 행렬인 interaction map을 CNN(합성곱 신경망 모델)을 통과시킨다. 본 연구에서는 식 (6)의 연산에 의한 결과 값인 4개의 2차원 행렬을 이미지의 채널로 간주하여 CNN 모델에 통과시킨다. 또한 이 과정에서 채널 간의 중요도가 학습에 반영될 수 있도록 Channel Attention 기법을 사용한다.

다양한 채널을 가지는 interaction map을 Channel Attention을 통하여 학습한다면, 특정 채널의 중요도를 고려하여 학습이 가능하다는

장점이 존재한다. 이는 다양한 부가정보 중에서 중요한 부가정보를 선택적으로 학습하는 것이 가능하며, 암묵적이던 채널들 간의 상관관계를 명시적으로 표현하는 것이 가능하기 때문에 여러 채널로 확장한 모델에 Channel Attention을 적용하는 것이 효과적이라고 할 수 있다.

4. 실험결과

4.1. 실험 데이터 및 방법

본 연구에서는 제안한 모델의 성능을 기존 ONCF와 비교하기 위하여 MovieLens-1M을 축소된 데이터를 사용하였다. 실험을 위한 원본 데이터는 유저 6040명, 영화 3900편의 1만 개의 평가에 대한 데이터이다. 실험의 빠른 비교를 위하여 200명의 사용자를 랜덤으로 추출하였다. 해당 데이터를 딥러닝 기반의 추천 시스템에 학습시키기 위해서는 사용자가 시청하지 않은 영화 중 일부를 샘플링하여 학습에 사용하는 네거티브 샘플링(Negative Sampling)이 필수적이다.

네거티브 샘플의 수를 다양하게 하여 실험한 결과 네거티브 샘플(시청하지 않은 영화)과 포지티브 샘플(시청한 영화) 간의 비율을 10대 1로 하였을 때 가장 높은 성능이 나타났다. 따라서 이후 실험의 네거티브 샘플 수는 10으로 고정하여 실험을 진행하였다.

실험 시 모델의 학습도를 판단하고 최종 모델의 성능 평가를 위하여 데이터를 훈련용 데이터와 테스트용 데이터로 나누어 실험을 진행하였다. 사용자별로 시청한 하나의 데이터를 테스트 데이터로 사용하여 각 사용자별로 모든

영화에 대해 선호도에 대한 예측을 진행한 후, 예측값이 가장 높은 k 개를 추천하여 성능을 측정하였다.

성능 확인을 위한 지표는 HR과 AUC를 사용하였다. 첫번째로 HR은 전체 사용자 수 대비 적중한 사용자 수를 의미하는 지표이다. 이를 통해 이해하기 쉽고 직관적으로 모델의 성능을 이해할 수 있다. 두번째로 AUC를 평가지표로 사용함으로써 FPR과 TPR을 모두 고려하여 모델의 성능을 측정하는 것이 가능하다.

$$Hit Rate = \frac{\# of Hit User}{\# of User} \quad (7)$$

기존의 ONCF와 본 연구에서 제안한 모델의 성능을 비교하기 위하여 모델의 구조 외에 모든 부분을 통일하여 실험을 진행하였다.

기존 ONCF와 본 연구에서 제안한 모델 모두 딥러닝을 기반으로 한 모델이기 때문에 하이퍼파라미터를 설정하여 실험을 진행해야 한다. 다양한 하이퍼파라미터를 고려하여 실험을 진행한 후 본 실험의 목적에 맞게 하이퍼파라미터를 다음과 같이 설정하였다.

먼저, 학습 횟수는 조기종료와 같은 방식으로 설정하였다. 학습 진행 시 더이상 학습이 진행되지 않는 지점인 epoch 50을 비교 횟수로 설정하였다. 다음으로 본 실험의 손실함수는 MSELoss를 사용한다. MSELoss는 모델이 확실하게 결과를 맞출 때까지 계속해서 학습을 진행한다는 장점을 가지고 있기 때문에 보다 안정적인 학습이 가능하다. 옵티마이저는 사전 실험을 통하여 설정하였으며 가장 높은 성능을

보인 Adam을 사용하였으며 배치 크기는 512개로 설정하였다. 모든 실험은 실험의 구현을 위해 python 3.8 언어와 pytorch 1.11.0 버전을 사용하였다

4. 2. 실험 결과

본 연구에서는 세가지의 실험을 진행하였다. 첫번째로, 기본 ONCF모델에 부가정보를 사용하는 것이 성능 향상에 도움이 되는지를 검증하기 위하여 기본 ONCF와 Channel Attention을 사용하지 않고, Wide&Deep만을 사용하여 부가정보를 추가한 모델과의 성능을 비교한다. 두번째로 부가정보를 추가할 때, Channel Attention을 사용하지 않은 모델과 사용한 모델과의 비교를 통해 채널간의 중요도를 강조하는 것이 부가정보를 효과적으로 사용하는데 도움이 되는지를 검증하고자 하였다. 최종적으로 기본 ONCF와 본 연구에서 제안한 모델의 성능을 비교하여 Channel Attention 기법을 활용한 부가정보가 추천시스템의 성능을 향상시키는데 유의미한 기여를 하는지 확인하고자 하였다.

세가지 실험의 결과는 <표1>과 같다. 실험 결과 Wide&Deep의 방식을 통해 부가정보를 단독으로 사용하는 것은 기본 ONCF와 비교하였을 때, 성능 향상에 영향을 주지 않음을 알 수 있다. 하지만 Channel Attention 기법을 사용하여 채널들에 중요도 정보를 추가하는 방식은 성능 향상에 도움을 주는 것을 알 수 있다.

Model	HR@K			AUC
	K = 10	K = 15	K = 20	
ONCF	0.55	0.63	0.68	0.7286
WD-ONCF	0.485	0.55	0.62	0.71591
WDCA-ONCF	0.575	0.655	0.705	0.79067

<표1>

즉, 부가정보와 채널 간의 중요도 모두 사용될 때에 모델 성능 향상에 기여한다.

5.결론 및 한계점

5.1. 결론

온라인 시장의 성장으로 인해 다룰 수 있는 상품의 수가 많아지면서 소비자가 구매의사결정을 할 때 드는 비용이 증가했다. 따라서 추천시스템에 부가정보를 활용하는 방식에 대한 연구가 다양한 분야에서 이루어져왔다.

본 연구는 부가정보 없이 협업 필터링 기법과 외적의 방식을 통해 좋은 성능을 냈던 ONCF에 부가 정보를 활용하여 추천시스템의 성능을 개선하였다는데 의의가 있다. 추가로 기존에 하나로 구성되었던 Interaction Map을 부가정보를 활용하여 여러 채널로 확장하였다는 것에 의의가 있다.

결론적으로 본 연구에서 제안한 방법론은 기존의 ONCF에 더욱 풍부한 부가 정보를 추가하여 추천시스템의 성능 향상에 기여하였고, Channel Attention의 방식을 사용하여 더욱 효과적으로 부가정보를 사용할 수 있음을 입증하였다.

또한, 사용자가 매기는 평점이 사용자와 아이템의 부가정보에 따라 영향을 받을 수 있음을 시사하였다. 따라서 더욱 다양한 부가정보를 활용한다면 더욱 좋은 성능을 낼 수 있을 것이라고 기대한다.

5.2. 한계점

위의 마지막 문단과 반대로 사용할 수 있는 부가정보의 수가 일정 수준 이상 되지 못한다면 부가정보가 성능 향상에 도움을 줄 수 있다고 확신할 수 없는 한계를 지닌다. 본 연구에서는 부가정보의 수가 다양한 MovieLense데이터를 사용하여 실험을 진행하였기 때문에 부가정보의 수에 따른 비교와 적용이 불가능하다. 따라서 향후 연구에서는 다양한 데이터에 대해 실험을 진행하고 부가정보의 수에 따른 실험을 진행할 예정이다. 추가로, 빠른 실험을 위해 데이터를 축소하는 과정에서 본래 데이터가 가지고 있던 정보가 매우 많이 손실되었다는 한계를 가진다. 샘플링한 데이터의 수를 다양하게 하여 실험을 진행한다면 더욱 좋은 성능과 결과를 얻을 수도 있을 것이라고 생각한다.

기존 ONCF에 Wide&Deep을 이용하여 부가정보를 추가한 모델의 경우 기존 ONCF에 비해 성능이 낮았던 것을 확인할 수 있었다. 이는 무조건 적으로 부가정보를 활용하는 것이 성능 향상에 기여하는 것이 아님을 의미한다. 본 연구에서 제안한 방식은 기존의 부가정보를 User_Wide, User_Deep, Item_Wide, Item_Deep으로 학습시켜 4가지의 채널로 이용한다. 해당 방식을 변형하여 여러가지 조합을 통해 학습을 진행한다면 다양한 경우를 고려하여 실험이 가능할 것이다. 예를 들어, User와 Item의 부가정보 중에서 Wide와 Deep중 하나의 부분만 사용하여 채널을 구성하는 방식으로 모델을 수정하여 성능을 비교해본다면, 성능 향상을 저해하는 요소를 발견할 수 있을 것이라고 생각한다.