



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

석사학위 청구논문

지도교수 박 창 규

패션제품 추천을 위한
협업 딥러닝 시스템 개발

2019년 2월

건국대학교 대학원
유기나노시스템공학과
이 광 한

패션제품 추천을 위한
협업 딥러닝 시스템 개발

Development of Collaborative Deep Learning
System for Fashion Recommendation

이 논문을 공학 석사학위 청구논문으로 제출합니다

2018년 10월

건국대학교 대학원
유기나노시스템공학과
이 광 한

이광한의 공학 석사학위 청구논문을 인준함

심사위원장 _____ 조 재 환 (인)

심사위원 _____ 고 준 석 (인)

심사위원 _____ 박 창 규 (인)

2018년 12월

건국대학교 대학원

목 차

표 목 차	iv
그림목차	v
국문초록	vii
제1장 서론	1
제1절 연구 배경 및 목적	1
제2장 이론적 배경 및 선행연구	3
제1절 이론적 배경	3
1. 콘텐츠 기반 필터링	3
2. 협업필터링	3
3. 인공신경망	8
4. Convolutional Neural Network	12
제2절 선행연구	14
제3장 제안시스템	16
제1절 Multilayer Perceptron	16
제2절 Visual based Generalized Matrix Factorization	18
제3절 Hybrid Visual based Recommender System	20
제4장 실험 및 결과	22
제1절 데이터 세트	22
제2절 학습 및 결과	24
1. Latent factor	25

2. Number of Negatives	28
3. Multilayer Perceptron Architecture	30
제3절 평가 방법 및 토의	31
 제5장 결론 및 논의	 35
제1절 요약 및 결론	35
제2절 향후 연구 방향	38
참고문헌	39
ABSTRACT	45

표 목 차

<표 4-1> Tradesy.com 데이터세트	22
<표 4-2> facotr 개수에 따른 HR@10과 NDCG@10의 변화	26
<표 4-3> Positive당 Negative 개수에 따른 HR@10과 NDCG@10의 변화	29
<표 4-4> MLP 구조 변화에 따른 HR@10 과 NDCG@10	30
<표 4-5> All items와 Cold items 세팅에서 각 모델의 성능 비교	33

그 립 목 차

<그림 1-1> 추천시스템 서비스 적용사례	2
<그림 2-1> 인공신경망의 구조	9
<그림 2-2> Neural Network에서의 Drop out 기법	11
<그림 2-3> Convolutional Neural Network Architecture	12
<그림 3-1> Multilayer Perceptron	16
<그림 3-2> Visual based Generalized Matrix Factorization 구조	18
<그림 3-3> Hybrid Visual based Recommender System 구조	20
<그림 4-1> facotr 개수에 따른 HR@10의 변화	25
<그림 4-2> facotr 개수에 따른 NDCG@10의 변화	26
<그림 4-3> Positive당 Negative 개수에 따른 HR@10 변화	28
<그림 4-4> Positive당 Negative 개수에 따른 NDCG@10 변화	28

국문초록

패션제품 추천을 위한 협업딥러닝 시스템 개발

본 논문에서는 매일 쏟아지는 패션 상품들 중에서 사용자의 취향을 고려한 패션 상품을 효과적으로 추천해주기 위해 기존의 추천시스템과 딥러닝을 결합한 하이브리드형 추천시스템을 개발하였다. 딥러닝은 최근 이미지, 영상, 자연어처리, 음성인식 등의 다양한 분야에서 뛰어난 성능을 보이는 기술이며 추천시스템 도메인에서 사용자와 상품사이의 상호작용을 효과적으로 모델링하기 위해 딥러닝을 접목하려는 연구가 많아지고 있다. 의류 패션시장의 SPA 브랜드 같은 경우 매주 새로운 의류상품이 쏟아지고 있고 이에 따라 추천시스템은 사용자에게 취향을 고려한 상품을 추천해줘야 하지만 상품에 대한 사용자의 평가가 존재해야 추천을 해줄 수 있는 협업필터링 추천시스템 특성 때문에 ‘Cold start’ 라는 문제가 발생하게 된다. 본 연구에서는 사용자가 의류 상품을 구매할 때 시각적 특징을 고려하여 구매한다고 가정하고 의류 상품의 이미지 데이터를 사용함으로써 이 ‘Cold start’ 문제에 대처했다. 또한 이 모델을 다시 딥러닝 모델인 Multi Layer Perceptron과 하이브리드 하여 성능을 더욱 향상시킴에 따라 패션시장에서 사용할 수 있는 하이브리드형 딥러닝 추천시스템을 제안한다.

추천시스템 모델을 학습시킬 데이터세트는 중고의류사이트에서 크롤링한 ‘Tradesy.com’ 를 사용하였고 상품에 대해 20회이상 평가한 사용자만을 추출하여 만들었다. 데이터세트는 8,355명의 사용자와 258,943의 상품과 492,350의 상호작용으로 이루어져 있다. 또한 제안한 모델이 ‘Cold start’ 문제를 잘 해결하는지 보이기 위해 ‘All items’와 ‘Cold items’ 세팅으로 나누고 데이터 집합은 각 사용자가 평가한 1개의 데이터를 테스트집합으로 사용하고 나머지를 훈련집합으로 사용해 실험을 진행하였다. 모델은 개발하기 위해 프로그래밍 언어 중 하나인 파이썬과 딥러닝 모델을 학

습하기 위해 딥러닝 프레임워크 중 하나인 케라스를 사용하였고 cpu, memory, gpu 각각 intel® Core™ i5-8600 @3.10Ghz, 32GB, 1080Ti 에서 실험을 진행하였다. 본 연구에서 제안하는 Visual Generalized Matrix Factorization의 성능을 평가하기 위해 Random, Itempop, Matrix Factorization, Generalized Matrix Factorization의 모델들과 비교를 하였다. 각 모델들을 비교하기 위한 평가지표로는 Hit Ratio(HR) 와 Normalized Discounted Cumulative Gain(NDCG)를 사용하였으며 제안한 VGMF 모델이 ‘All items’ 세팅에서 GMF와 MF에 비해 각각 13.7%, 33.5%의 성능 향상을(NDCG) 보였고 ‘Cold items’ 세팅에서는 각각 18.7%, 71.0%의 성능향상을(NDCG) 보였다. 또한 VGMF와 MLP를 하이브리드 하여 만든 모델의 경우 VGMF에 비해 ‘All items’, ‘Cold items’ 세팅에서 각각 8.8%, 7.9%의 성능향상을(NDCG) 보였다. 이를 통해 패션 제품을 추천할 때 상품의 이미지를 활용할경우 모델의 전체적인 성능이 올라감을 알 수 있고 또한 상품에 대한 평가정보가 없어서 발생하는 ‘Cold start’ 일 때 상품의 이미지를 사용하면 이 문제에 대해 기존의 모델들에 비해 잘 대처할 수 있음을 알 수 있다.

이처럼 본 연구에서는 최근 활발하게 연구가 이뤄지고 있는 딥러닝을 추천시스템에 결합하여 사용자의 취향을 고려한 패션제품 추천시스템 모델을 제안하였다. 비록 상품의 메타데이터 중 이미지정보만을 사용하여 성능을 높이려 했지만 추후 연구에서 사용자의 리뷰데이터, 상품 클릭 횟수, 사용자의 나이, 사용자의 성별 등의 다양한 메타데이터들을 추천시스템 모델에 결합 한다면 사용자에게 더욱 개인화된 추천을 해줄 수 있을 것이다

제 1 장 서론

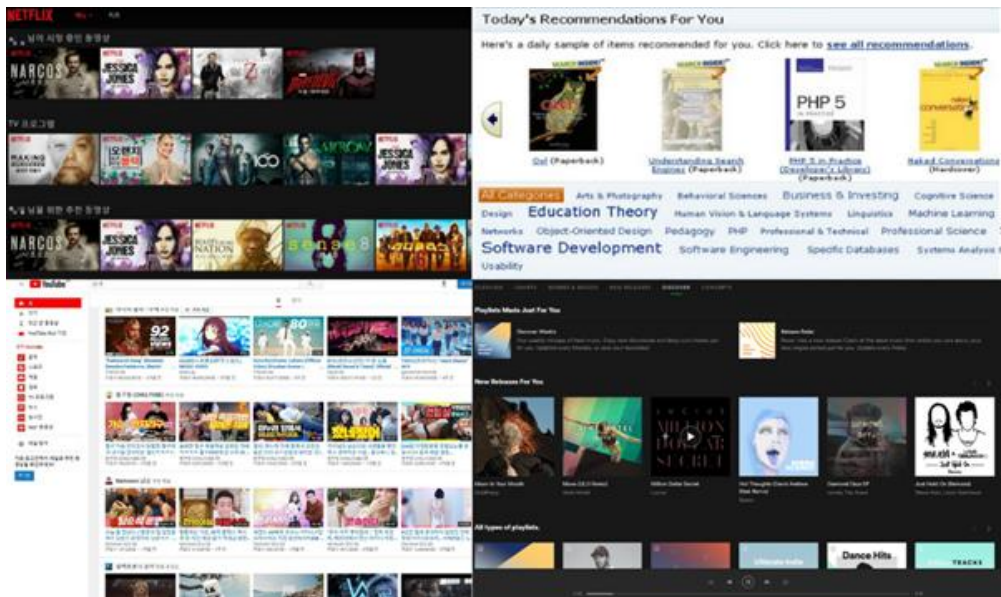
제1절 연구의 배경 및 목적

오늘날 인터넷의 발달과 스마트기기의 대중화로 인해 시간과 공간 제약 없이 온라인 서비스를 실시간으로 이용할 수 있게 되었다. 또한 다양한 형식의 많은 양의 데이터가 빠른 속도로 생성 되고 사용자가 효율적으로 처리할 수 없는 ‘정보과부하(information overload)’가 발생하였다[1]. 이런 ‘정보 과부하’로 인해 사용자가 웹에서 관련 콘텐츠를 찾기가 더 어려워지고 중요한 정보를 놓치는 경우가 생기고 있다[2]. 이에 따라 사용자는 데이터를 효율적으로 필터링하고 가장 관련성이 높은 콘텐츠를 제공해줄 수 있는 추천시스템이 필요하게 되었다.

추천시스템은 정보 과부하 문제에 대처하고 사용자에게 유용한 정보를 제공하는 어플리케이션이다[3]. 추천시스템은 사용자가 선호하는 콘텐츠의 순위 목록을 생성해주는 기술로서 추천 모델로는 크게 협업필터링, 콘텐츠 기반과 하이브리드 추천시스템으로 분류되고[4] 추천모델로부터 생성되는 추천목록은 일반적으로 사용자 선호도, 아이템 특징, 사용자와 아이템의 상호작용, 시간 및 공간데이터와 같은 정보를 기반으로 생성된다[5]. 이러한 추천시스템을 이용하여 ‘정보 과부하’를 처리한 다양한 성공사례가 있다. 예를들어 Amazon.com[6]의 제품 추천시스템, Netflix[7]의 영화 추천시스템, 뉴스기사 [8], 트위터 [9] 등이 있다. 최근에는 이러한 추천시스템에 딥러닝을 결합하여 추천성능을 높이려는 연구가 활발하다. 딥러닝은 이미지, 영상, 자연어처리, 음성인식 등의 다양한 분야에서 뛰어난 성능을 보이는 기술이며 입력데이터의 복잡한 상호작용을 효과적으로 처리하여 원하는 값을 예측할 수 있다.

사용자에게 추천을 해주는 모델을 학습시킬 때는 사용자가 상품에 남긴 평가정보가 필요하다. 하지만 의류 패션시장에서 SPA 브랜드의 경우 매 주 새로운 상품들이 쏟아져 나오기 때문에 상품에대한 사용자의 평가가

없는게 대부분이다. 이렇게 새로운 상품에 대한 정보가 부족할 때 발생하는 문제를 ‘Cold start’라고 한다. 본 연구에서는 의류 패션시장에서 사용자에게 상품을 추천할 때 발생하는 ‘Cold start’ 문제를 해결하기 위한 추천시스템 모델을 개발하고자 한다.



<그림 1-1> 추천시스템 서비스 적용사례

제2장 이론적 배경 및 선행연구

제1절 이론적 배경

1. 콘텐츠 기반 필터링

콘텐츠 기반 필터링은 유사한 특징을 가진 항목들이 사용자에게 의해 유사한 방식으로 평가될 것이라는 가정하에 개발되었다[10]. 콘텐츠 기반 필터링은 아이템의 정보를 이용하여 추천을 하기 때문에 사용자 선호도 정보가 없어서 발생하는 콜드스타트(Cold Start)문제를 해결할 수 있다. 하지만 비슷한 특징을 가진 상품들만을 추천해주기에 참신한 아이템을 추천해주지 못한다는 단점을 가진다.

2. 협업필터링

협업필터링은 비슷한 취향을 가진 사용자들은 비슷한 상품을 선호할 것이라는 가정에 기반하는 방법이다. 이러한 협업필터링에는 크게 메모리 기반 필터링과 모델기반 필터링이 있는데 메모리 기반 필터링에는 유저기반 필터링, 아이템기반 필터링이 있고 모델기반 필터링에는 크게 SVD, Matrix Factorization 이 있다

메모리 기반 협업필터링에는 유저 기반 협업필터링과 아이템 기반 협업필터링이 있다. 둘의 차이는 사용자들 간의 이웃을 형성하여 추천하느냐 상품 간의 이웃을 형성하여 추천하느냐이다. 메모리 기반 협업필터링은 높은 유사도를 가지는 이웃들은 같은 상품에 비슷한 평가를 내릴 것이라는 가정을 가지고 있다. 이웃을 형성하는 방법은 사용자나 상품간의 유사도를 구하고 상위 n 명의 최종적으로 평가에 기여할 이웃을 선정하여 최근접 이웃의 평가에 기반해 추천을 한다.

이웃들간의 유사도는 크게 ‘유클리디안 거리 유사도’, ‘코사인 유사도’, ‘피어슨 유사도’를 이용하여 구할 수 있는데 일반적으로 ‘코사인 유사도’와 ‘피어슨 유사도’를 사용한다.

$$cosine_sim(u,v) = \frac{\sum_{i \in I_{uv}} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2} \cdot \sqrt{\sum_{i \in I_v} r_{vi}^2}} \quad (1)$$

$$pearson_sim(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \mu_u) \cdot (r_{vi} - \mu_v)}{\sqrt{\sum_{i \in I_u} (r_{ui} - \mu_u)^2} \cdot \sqrt{\sum_{i \in I_v} (r_{vi} - \mu_v)^2}} \quad (2)$$

‘코사인 유사도’와 ‘피어슨 유사도’는 -1 에서 1사이의 값을 가지고 1에 가까울수록 비슷한 취향을 가지는 이웃이라고 할 수 있다. ‘피어슨 유사도’는 ‘코사인 유사도’와 다르게 평가값에 평균을 빼서 계산하기 때문에 사용자의 점수편향에 영향을 덜 받는다는 장점이 있다. 위와 같은 유사도를 이용하여 최근접 이웃을 형성할 때 각 사용자가 평가한 최소한의 상품수를 정해주지 않으면 극단적으로 하나의 같은 상품을 평가한 사용자간의 코사인 유사도를 구하면 평가점수와 상관없이 유사도는 무조건 0이 나오게 된다. 이러한 점을 방지하기 위해서 이웃간의 평가한 상품의 수에 비례해서 유사도를 구해줄 수 있는데 식은 다음 (3)과 같다.

$$pearson_baseline_shrunk_sim(u,v) = \frac{|I_{uv}| - 1}{|I_{uv}| - 1 + shrinkage} \cdot \hat{p}_{uv} \quad (3)$$

하나의 상품만을 평가한 사용자들간의 유사도를 구하게되면 피어슨유사도에 ‘shrinkage’ 계수를 나눈 아주 작은 유사도를 가지게 된다. 사용자들간의 평가한 상품 수에 비례해서 유사도는 증가하게 된다. 유사도를 구한 뒤 ‘KNN(K-Nearest-Neighborhood)’ 기법을 통해 최종적으로 사용자나 상품의 평가점수를 예측할 수 있다.

유저 기반 협업필터링은 사용자와 비슷한 취향을 가진, 즉 높은 유사도를 가진 사용자들의 피드백을 가중평균하여 예측하는 방법이다. 어떤 사용자

가 평가하지 않은 상품에 대한 평가점수를 예측한다고 할 때 다른 사용자들이 상품에 남긴 평가점수를 평균해서 예측하는 방법이 있다. 하지만 이 방법은 사용자들 간의 다른 취향을 전혀 고려하지 않은 방법이기에 예측 성능이 아주 낮다. 하지만 ‘KNN(K-nearest-neighborhood)’ 기법을 사용하여 유사도를 통해 최근접 이웃을 형성하고 최근접 이웃 N명의 평가점수를 가중평균하면 사용자들 간의 유사도를 가중치로 평가점수를 예측했기에 일반적인 평균방법에 비해 성능이 잘나온다.

$$\hat{r}_{ui} = \frac{\sum sim(u,v) \cdot r_{vi}}{sim(u,v)} \quad (4)$$

하지만 이 방법은 또 다른 편향을 가지고 있다. 각 사용자들은 어느 상품에 대해 평가를 내릴 때 다른 사용자들에 비해 점수를 높게 줄 수도 있고 더 낮게 줄 수도 있다. 즉 점수를 주는 기준이 다르기 때문에 한 사용자에게는 높은 점수일지라도 다른 사용자에게는 낮은 점수일 수 있어서 정확하지 않은 예측점수를 구하게 된다. 이와 같은 경우에는 ‘KNN’ 기법을 변형시킨 아래의 식 (5)와 같은 ‘KNN-mean-centering’ 을 사용한다. 이 기법은 최근접 이웃들의 평가점수에 각 사용자의 상품 평균점수를 빼주어 각 사용자가 평가 점수에 가지는 기준을 고려하여 점수를 예측할 수 있게 한다.

$$\hat{r}_{ui} = \mu_u \frac{\sum sim(u,v) \cdot (r_{vi} - \mu_v)}{sim(u,v)} \quad (5)$$

아이템 기반 협업필터링은 사용자 기반 협업필터링과 비슷하고 아이템 간의 유사도를 구하여 최근접 이웃을 형성하여 각 상품들이 매겨진 평가점수를 유사도에 따라 가중평균하여 구하고자 하는 상품의 예측점수를 구

할 수 있다. 이 또한 평가점수에 대한 편향이 존재하기 때문에 ‘KNN’ 기법을 변형시킨 ‘KNN-mean-centering’ 을 사용하여 구하고자 하는 상품에 대한 점수를 예측할 수 있다.

메모리기반 필터링에는 세가지 단점이 있다.(하이브리드 뉴럴넷 논문참고) 첫번째는 ‘Data Sparsity’ 이다. 이것은 아직 평가하지 않은 많은 수의 상품들과 관련이 있다[6]. 사용자와 상품의 가능한 상호작용 수에 비해서 실제로 존재하는 평가 수는 아주 적다. 그래서 이미 평가된 적은 데이터로 평가되지 않은 많은 부분들을 채워야 하기 때문에 모델을 만들 때 어려움이 있다. 두번째는 ‘Cold Start’ 이다. 이 문제는 새로운 상품이나 새로운 사용자가 생겼을 때 발생한다[11]. 새로운 사용자가 추가됐을 때 사용자가 상품에 평가한 기록이 있어야 이 사용자에게 올바른 추천을 해줄 수 있다. 또한 새로운 상품이 추가됐을 때 상품이 사용자에게 최소한의 평가를 받아야 다른 사용자에게 평가를 해줄 수 가있다. 이처럼 새로운 사용자나 새로운 상품이 시스템에 추가됐을 때 최소한의 평가가 존재해야 추천을 해줄 수가 있는데 새로운 사용자나 상품은 아무런 정보가 없기 때문에 ‘Cold Start’ 문제가 발생한다. 세번째는 Scalability이다. 사용자수와 상품의 수가 늘어날수록 그에 따라 연산 시간이 기하급수적으로 늘어난다[12]. 사용자 기반 협업필터링이나 상품 기반 협업필터링의 경우 사용자들 간의 상품간의 유사도를 각각 구해서 가중평균을 해야 하는데 사용자수와 상품수가 늘어날수록 연산 해야 하는 양이 많아지기 때문에 메모리 기반 협업필터링의 경우 대규모 추천시스템에서 ‘Scalability’라는 단점을 가지고 있다.

모델기반 필터링은 반면에 데이터에서 패턴을 찾고 예측등급과 평가등급의 차이가 최소화되도록 모델을 훈련시킨다. 모델을 훈련시키기 위하여 세 가지 예측 방법이 있는데 등급 예측[10,13,14], 클래스 예측[15][16], 랭킹 예측[17] 방법이 있다. 이러한 모델기반 방법에는 다양한 방법이 있지만 그 중에 행렬분해모델인 SVD, Matrix Factorization이 주로 쓰인다. SVD는 사용자가 상품에 평가한 정보가 있을 경우, 행렬을 분해하였을 때 잠재적 특성을 잘 정의한다[18]. 하지만 SVD는 사용자가 상품에 대해 평

가한 정보가 없을 경우 행렬을 분해 할 수 없고 대신 이 결측치를 평균점수로 대체했을 경우 편향이 발생하기 때문에 모델성능이 좋지 못하다. 하지만 Matrix Factorization은 평가 정보만을 이용하여 결측치를 예측하기에 이 문제를 해결할 수 있다. 추천 시스템에서 Matrix Factorization은 모든 사용자가 상품에 부여한 점수로 구성된 평점행렬 R 을 K 차원의 잠재적 특성 행렬(Latent Feature Matrix) P 와 Q 로 분해한다.

$$R \approx PQ^T \quad (6)$$

그리고 이 잠재적 특성 행렬을 이용해 평점행렬 R 에 근사하는 \hat{R} 을 구하고 결측치(Missing Values)를 예측한다.

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u \quad (7)$$

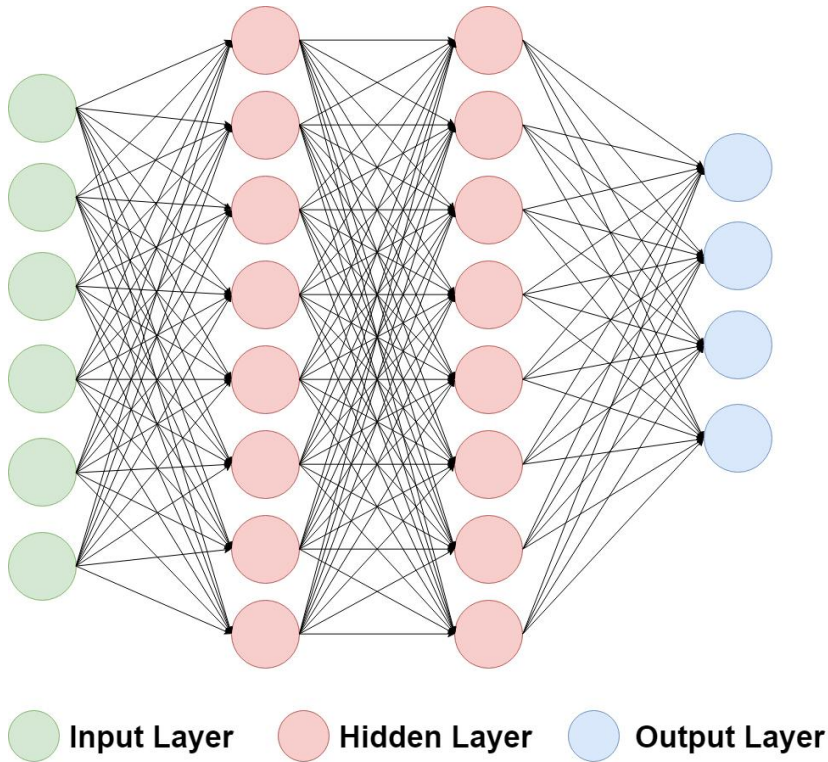
평점행렬의 사용자 u 가 상품 I 에 평가한 점수는 위의 식 (7)처럼 사용자 선호도 요인벡터(user preference factor vector)와 상품 선호도 요인벡터(item preference factor vector)의 곱 형태로 나타내고 평균점수, 상품편향, 사용자 편향을 더하여 예측할 수 있다. 실제평점과 예측평점의 차이의 제곱을 최소화 하도록 하는 파라미터를 학습하여 평점행렬 R 을 근사할 수 있고 결측치를 예측할 수 있다. 손실함수를 최소화 하도록 하는 파라미터는 Stochastic Gradient Descent 와 Alternating Least Squares의 방법으로 구할 수 있다.

하지만 위의 협업필터링 방법들은 사용자가 상품에 평가한 정보가 충분하지 않다면 추천을 해주기 어려운 콜드스타트(Cold Start) 문제를 가지고 있다 협업필터링의 메모리기반 추천시스템의 경우 유사도를 가중치로 사용하기에 사용자에게 설득력 있는 추천을 해주지만 새로운 상품에 대한

평가가 이루어지지 않았을 때 ‘Cold Start’ 문제가 발생하고 데이터가 많아질수록 연산하는데 시간이 오래걸린다. 모델기반 방법중 Matrix Factorization은 사용자와 상품의 잠재적 특성을 학습시켜서 선호도를 예측하는 모델로 뛰어난 성능을 보인다[7]. 하지만 Matrix Factorization은 선형모델이기 때문에 사용자와 상품사이의 복잡한 상호작용을 학습하기에 한계가 있다. 이 논문에서는 Matrix Factorization을 비선형으로 표현하여 사용자와 상품사이의 복잡한 상호작용을 학습하여 기존의 Matrix Factorization에 비해 모델의 성능을 향상시켰다.

3. 인공신경망

인간의 뇌 회로를 모방한 인공신경망은 아주 긴 역사를 가지고 있다. 인공신경망은 입력층, 은닉층 과 출력층으로 이루어져 있고 하나이상의 은닉층이 있는 경우 다층신경망이라 한다. 인공신경망은 복잡한 비선형 관계를 포착하고 모델링 할 수 있다.



<그림 2-1> 인공신경망의 구조

하지만 은닉층이 깊어질수록 많은 계산량이 필요해지고 기울기 소실 (gradient vanishing)과 과적합(Overfitting) 문제가 발생한다. 이에 다양한 해결방법이 있다.

gradient vanishing 은 은닉층이 깊어질수록 발생하는 문제이다. 활성화 함수란 신경망에서 입력신호의 총합을 출력신호로 변환하는 함수를 나타내고 비선형함수를 사용한다. 그래서 기존의 신경망에서는 주로 sigmoid 활성화함수를 사용하였다.

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (8)$$

sigmoid 함수는 출력값이 0에서 1사이로 나오게되고 많이 사용됐지만

레이어를 깊게 쌓을수록 그래디언트 값의 곱이 0에 가까워지기 때문에 학습이 느리게 진행되거나 전혀 학습이 되지 않는 경우가 발생했는데 이 문제점이 바로 gradient vanishing 이다. 이 문제점을 극복하기 위해 tanh 라는 활성화 함수를 사용하게 되었다.

$$\tanh(x) = 2\sigma(2x) - 1 \quad (9)$$

tanh 함수는 출력값이 -1에서 1사이로 나오고 sigmoid 함수에 비해 gradient vanishing 문제가 완화되긴 했지만 여전히 문제점을 가지고 있다.

$$f(x) = \max(0, x) \quad (10)$$

하지만 최근 신경망에서 ReLu 라는 활성화 함수를 사용하게 되며 이러한 gradient vanishing 문제를 어느정도 해결했다. ReLu 함수는 0아래의 음수인 경우 그래디언트가 전파되지 않지만 sigmoid 와 tanh 와 다르게 히든 레이어가 깊어질수록 그래디언트 값이 작아지지 않기 때문에 그래디언트가 전파되지 않을 일이 없고 수렴속도도 훨씬 빠르다.

위와 같은 이유로 현재 신경망의 레이어를 구성할 때 활성화 함수로는 특정한 경우를 제외하면 ReLu 함수를 사용해서 레이어가 깊더라도 gradient vanishing 문제를 해결하여 보다 복잡한 모델을 설계할 수 있게 되었다.

‘Overfitting’이란 모델이 학습데이터에만 지나치게 학습되어 성능이 오히려 떨어지는 경우를 말한다. 학습모델은 미래에 어떤 데이터가 들어오더라도 올바르게 판단해야 하는데 ‘Overfitting’이 발생하면 과거에 학습시킨 데이터만 올바르게 판단하고 미래의 데이터에 대해서는 올바르게 판단하

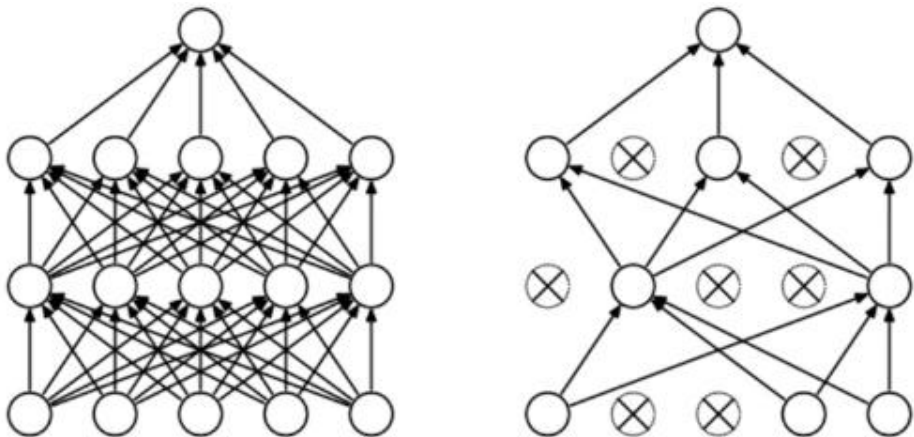
지 못하기 때문에 이 문제를 방지하기 위해 다양한 방법을 사용해야 한다..

첫번째로는 모델의 크기를 줄이는 방법이 있다. 신경망에서 학습할 파라미터의 수가 많을 경우 그 개수를 줄여서 ‘Overfitting’을 방지할 수 있다.

두번째는 ‘Early stopping’이다. ‘Overfitting’이 일어나기전, 학습모델이 복잡해지기 전에 학습을 멈춰서 ‘Overfitting’을 방지하는 방법이다.

세번째는 ‘Regularization’이다. ‘Regularization’은 학습될 파라미터의 크기를 줄이는 방법이다. 기존 손실함수에 모든 학습 파라미터의 제곱을 더하는 L2 Regularization이 있는데 이 방법은 큰 값을 가지는 파라미터에 제약을 주어 파라미터 값들이 가능한 널리 퍼지도록 하는 효과를 낸다. 또한 손실함수에 학습 파라미터의 절대값을 더하는 L1 Regularization 방법은 파라미터들을 거의 0에 가깝게 만드는 특성을 가진다. 이러한 Regularization 기법은 학습 모델이 학습데이터에만 맞춰서 너무 복잡한 모델이 되지 않도록 모델의 크기를 조절해서 미래에 들어오는 데이터에 대해서도 잘 예측하게 해준다.

네번째는 ‘Drop out’ 이다.



<그림 2-2> Neural Network에서의 Drop out 기법

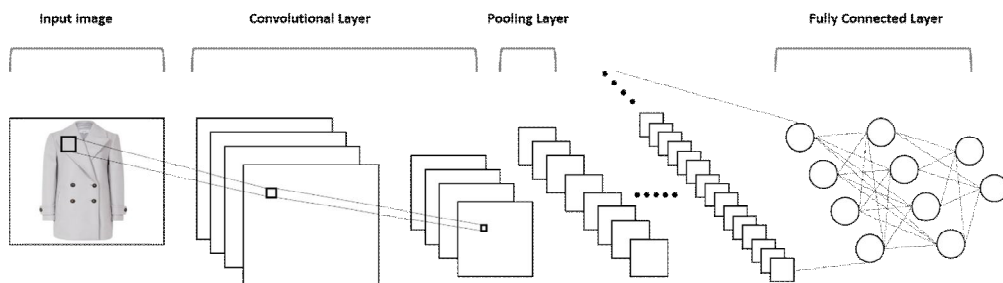
Drop out은 그림 2-2와 같이 일부의 뉴런을 사용하지 않고 학습하는 방

법이다[19]. 모든 파라미터들을 사용해서 모델을 학습시키는 것이 아니라 일부의 파라미터만을 사용하여 학습시키는 방법으로 좋은 성능을 내는 것으로 알려져 있다.

4. Convolutional Neural Network

Convolutional Neural Network는 1989년 “Backpropagation applied to handwritten zip code recognition”에서 처음으로 발표 되었다[20]. CNN은 다양한 분야에서 사용되는 신경망이지만 특히 이미지 인식에서 두각을 나타낸다. 이미지 데이터를 평범한 인공신경망에 입력하였을 때 파라미터의 개수가 너무 많아 학습을 시키기가 어렵다. 이러한 문제점을 해결하기 위해 만들어진 신경망이 바로 CNN이다.

CNN은 이미지 상에서 공간적으로 인접한 픽셀들에 대한 상관관계를 비선형 필터를 적용해 주요한 특징들만 추출하고 이미지 크기를 줄인다.



<그림 2-3> Convolutional Neural Network Architecture

이 논문에서는 2012년 “ImageNet Large Scale Visual Recognition Challenge” 을 우승한 AlexNet 구조를 ‘시각적 특징’ 추출기로 사용하였다. 추출기로 사용하기 위하여 120만장의 이미지를 사전 학습시켰다. 사전 학습시킨 AlexNet 모델은 5개의 Convolutional Layer와 3개의 Fully Connected Layer로 구성되어 있다[21]. Convolutional Layer에서는 필터를 적용해 feature map을 만든다. 그 다음 Pooling Layer 에서 feature

map의 크기를 줄인 후 얻은 feature가 Fully Connected Layer의 입력으로 들어가 결과를 낸다. 이때 이미지가 두번째 Fully Connected Layer를 거치고 난 후의 벡터 값을 '시각적 특징'으로 사용하였다.

제2절 선행연구

다양한 협업필터링 중에 Matrix Factorization은 가장 인기있는 모델이다[22][23]. 이 Matrix Factorization의 성능을 향상시키기 위해 연구가 활발하다. 상품의 토픽모델과 MF를 결합하거나[23], 이웃 근접 모델과 MF를 결합한 모델[24], MF를 Neural Network로 일반화시켜 표현력을 높이고 MLP와 Hybrid 하여 비선형적으로 표현한 신경망 모델[25]이 있다. 또한 최근에는 암시적 피드백을 MF나 다양한 모델로 해석하려는 연구가 많아지고 있다. 전통적인 Matrix Factorization 방법은 암시적 피드백에서 비관측 데이터를 해석할 때 어려움이 있었지만 최근 point-wise 와 pairwise 방법을 통해 Matrix factorization에 암시적 피드백을 성공적으로 적용시켰다[26].

그 중 point-wise 방법은 비관측된 피드백이 사용자가 어느정도 선호하지 않는다는 가정에서 시작하여 사용자와 아이템의 연관성 점수를 예측하고 그에 따라 상품을 순위화 한다. 그 예로 긍정적인 것과 비관측된 피드백에 서로 다른 '신뢰 수준'을 연관시키고 그 결과 가중치 행렬을 인수분해 하거나[27] 관측되지 않은 피드백을 음의 인스턴스로 샘플링하고 유사한 가중치 행렬을 인수분해 했다[28].

pairwise 방법은 관측된 피드백이 비관측된 피드백보다 조금 더 선호할 것이라는 가정에 기반 한다. 이 방법은 실제 점수와 예측 점수의 차이를 최소화하는 방식이 아닌 관측된 피드백의 순위와 비 관측된 피드백의 순위의 차이를 최대화 하도록 최적화 시킨다[25]. 그 예로 BPR(Bayesian Personalized Ranking) 프레임 워크를 제안하여 BPRMF가 다양한 경쟁 모델을 능가한다는 것을 실험적으로 보여주었다[29].

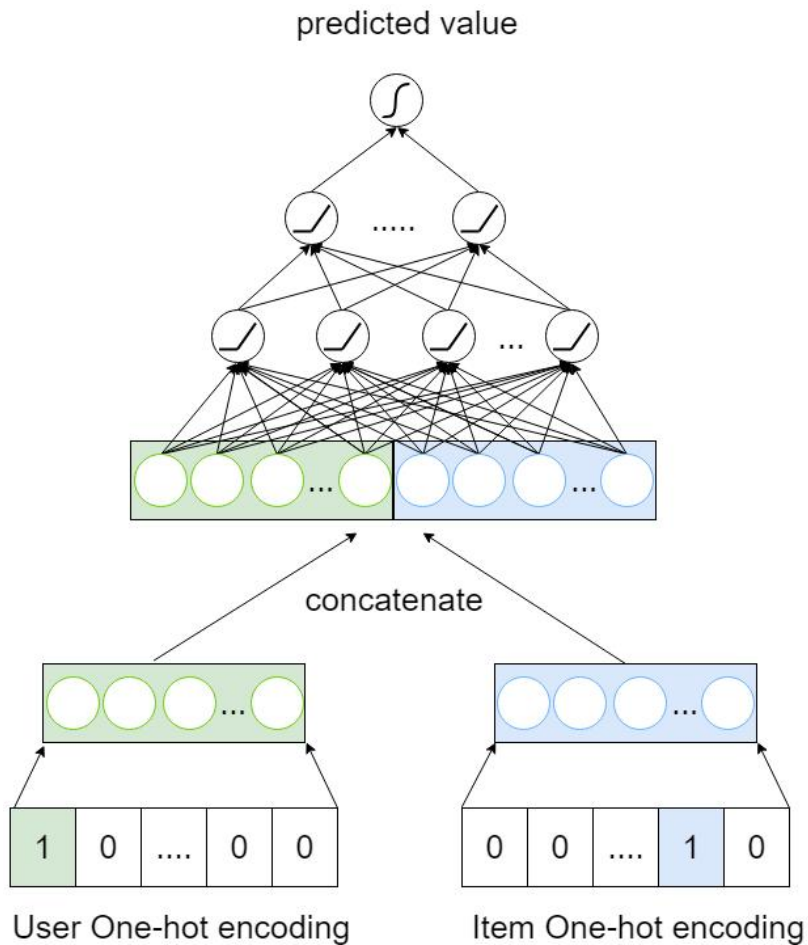
NCF모델[25]에서는 point-wise 와 pairwise 두 방법 모두를 사용하였으며, 출력값을 0에서 1사이의 값으로 나오게 하여 확률적으로 설명가능한 신경망 모델을 만들었고 기존의 Matrix Factorization이 사용자와 상품 사이의 복잡한 상호작용을 선형모델로 표현할 수 없다는 한계를 신경망

모델로 극복하였다. 하지만 새로운 상품이 들어왔을 때 ‘Cold Start’ 문제가 발생하는데 위와 같은 모델들은 이러한 점을 해결해주지 못한다는 단점을 가진다.

요약하면 [22,23,24,30] 연구에서 사용된 MF는 선형모델로서 사용자와 상품사이의 포착하지 못한다. 또한 [25]연구에서 MF가 가지는 한계를 극복하기 위해 MF를 신경망으로 일반화시켜 비선형모델을 만들었지만 이러한 협업필터링들은 근본적으로 ‘Cold Start’ 문제를 가진다. 이 논문에서는 이러한 ‘Cold Start’ 문제를 해결하기 위해 초기에 상품에 대한 평가정보가 아닌 상품에 대한 메타정보를 협업필터링 모델과 결합하였고 ‘Cold Start’ 뿐만 아니라 대용량 데이터세트에 대해서도 사용자와 상품사이의 관계를 잘 파악하는 모델을 만들기 위해 MLP 또한 결합한 모델을 제안한다.

제3장 제안시스템

제1절 Multilayer Perceptron(MLP)



<그림 3-1> Multilayer Perceptron 구조

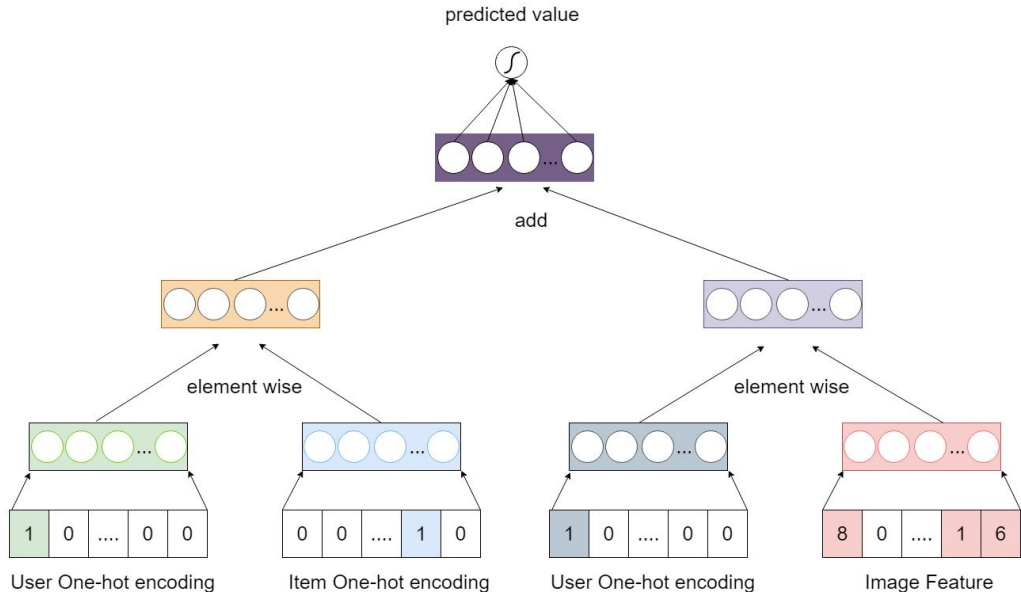
Matrix Factorization에서는 사용자와 상품의 latent factor를 내적하여 평가점수를 근사하였다. 하지만 두개의 latent factor를 concatenate 하는

방식으로 이 두 latent factor 사이의 관계를 학습 시킬 수 있다. 이러한 디자인은 딥러닝 모델로 넓게 적용되어져 왔다. 사용자와 상품 각각의 latent factor를 concatenate 한 후 은닉층을 쌓아 다층 신경망 모델로서 두 latent factor 사이의 복잡한 상호작용을 학습 시킬 수 있다. MLP 모델은 다음과 같이 정의할 수 있다.

$$\begin{aligned}
 z_1 &= \phi_1(p_u, q_i) = \begin{bmatrix} p_u \\ q_i \end{bmatrix}, \\
 \phi_2(z_1) &= a_2(W_2^T z_1 + b_2), \\
 &\vdots \\
 \phi_L(z_{L-1}) &= a_L(W_L^T z_{L-1} + b_L), \\
 \hat{y}_{ui} &= \sigma(h^T \phi_L(z_{L-1}))
 \end{aligned} \tag{11}$$

W_x, b_x, a_x 는 각 x번째 레이어에서 가중치 행렬, 바이어스 벡터, 활성화 함수이다. concatenate 하는 두개의 latent vector는 평균이 0이고 분산이 0.1인 가우시안 분포로 초기화 하였고 은닉층에서의 가중치 행렬은 He initializer[31]로 초기화하였고 ReLu 활성화 함수를 사용하였다. 또한 마지막 출력층에서의 가중치행렬은 Xavier initializer[32]로 초기화를 하였고 sigmoid 활성화 함수를 사용하여 출력값을 0에서 1사이의 값이 나오도록 하였다. 또한 은닉층의 개수와 각 층의 뉴런수는 실험을 통하여 결정하였고 Embedding vector 크기가 32 이고 은닉층은 1개이고 은닉층의 뉴런수는 16개일 때 성능이 가장 우수했다. MLP는 p_u 와 q_i 사이의 관계를 비선형적으로 학습을 시키기 때문에 기존의 Matrix Factorization이나 Neural Matrix Factorization에서 내적하는 방법보다 성능이 좋다. 하지만 이 논문의 데이터에서는 상품에 대한 평가정보가 부족해서 MLP모델이 더욱 복잡한 모델을 만들어 내지 못했다.

제2절 Visual based Generalized Matrix Factorization



<그림 3-2> Visual based Generalized Matrix Factorization 구조

Matrix Factorization 사용자 latent factor p_u 와 상품 latent factor q_i 를 내적하여 평가점수를 근사할 수 있다. 이를 Neural Matrix Factorization으로 일반화 시킬 수 있는데 다음과 같다.

$$\phi_1(p_u, q_i, vp_u, vq_i) = p_u \odot q_i + vp_u \odot vq_i, \quad (12)$$

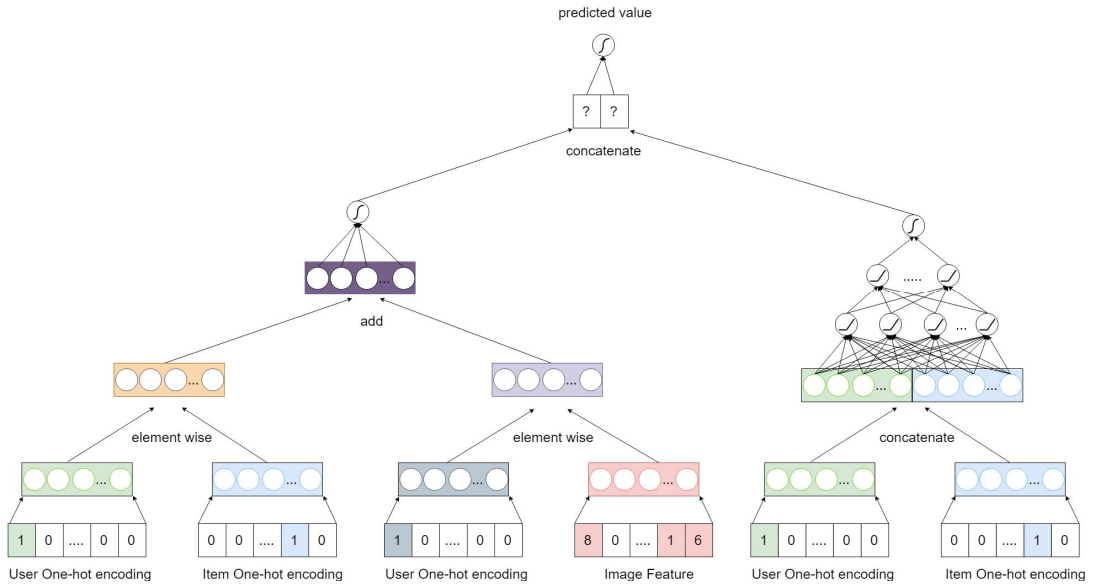
$$\hat{y}_{ui} = a_{out}(h^T(p_u \odot q_i + vp_u \odot vq_i))$$

h 는 출력층에서의 가중치 행렬이다. 만약 가중치 행렬의 모든 요소 값이 1 이라면 Matrix Factorization과 같은 식이 된다. 하지만 Generalized Matrix Factorization으로 일반화 시킨 위의 식에서 h 가중치 행렬은 학습을 통해서 요소 값이 다양한 값을 가지게 되고 사용자와 상품의 latent factor에 서로 다른 가중치를 줄 수 있다. 또한 a 의 활성화 함수를 사용하

기 때문에 선형성을 가지는 Matrix Factorization과 달리 비선형성을 가지게 된다. 이때 활성화 함수는 sigmoid 함수를 사용하고 Generalized Matrix Factorization은 기존의 Matrix Factorization에 비해 모델의 표현력이 높다.

하지만 새로운 상품이 들어왔을 때 이 상품에 대한 정보가 부족해서 상품의 latent factor 값을 제대로 학습시키지 못하게 되어 모델의 예측 성능이 나빠진다. 이렇게 새로운 상품에 대한 평가점수가 부족하여 ‘Cold Start’ 문제가 발생했을 때 상품의 메타데이터를 사용하면 모델의 성능을 높여줄 수 있다. 이 논문에서는 사용자가 의류상품을 구매할 때 시각적 정보를 바탕으로 구매한다고 가정하고 의류의 시각적인 메타정보인 시각적 특징을 사용하여 ‘Cold Start’ 문제를 해결하였다. 이때 pretrain된 CNN의 두번째 fully connected layer를 거친 4096길이의 벡터를 시각적 특징으로 사용하였다. 위의 식에서는 시각적 특징에 대한 사용자의 latent factor vp_u 와 시각적 특징의 latent factor vq_i 를 추가하여 새롭게 정의하였다. 결과적으로 기존의 Matrix Factorization과 Generalized Matrix Factorization에 비해 ‘Cold Start items’ 뿐만 아니라 ‘All items’ 세팅에 대해서도 성능이 향상됨을 보였다.

제3절 Hybrid Visual based Recommender System



<그림 3-3> Hybrid Visual based Recommender System 구조

MLP와 Visual Generalized Matrix Factorization은 각각 좋은 성능을 보이지만 두 모델을 hybrid 한다면 사용자와 상품 사이의 복잡한 관계를 더 잘 표현할 수 있다.

이 논문에서는 MLP와 Visual Generalized Matrix Factorization을 각각 pretrain 시키고 hybrid 하여 학습하였다. 한번에 두 모델을 학습하지 않고 각각 pretrain 하면 latent vector 사이즈를 자유롭게 조절하여 hybrid 시킬 수 있기 때문에 성능이 더 좋다는 장점이 있다. 두 모델을 hybrid 하는 식은 다음 (13)과 같이 표현할 수 있다.

$$\phi^{VGMF} = p_u^V \odot q_i^V, \quad (13)$$

$$\begin{aligned} \phi^{MLP} &= a_L(W_L^T(a_{L-1}(\dots a_2(W_2^T \begin{bmatrix} p_u^M \\ q_i^M \end{bmatrix} + b_2)\dots)) + b_L), \\ \hat{y}_{ui} &= \sigma(h^T \begin{bmatrix} \phi^{VGMF} \\ \phi^{MLP} \end{bmatrix}) \end{aligned}$$

pretrain된 각각의 모델에서 나온 출력값들을 concatenate하고 마지막 레이어를 거쳐갈 때 아래와 같이 각각의 모델 값에 가중치를 줘서 최종 hybrid 출력값을 (14)와 같이 낼 수 있다.

$$h \leftarrow \begin{bmatrix} \alpha h^{VGMF} \\ (1-\alpha)h^{MLP} \end{bmatrix} \quad (14)$$

α 는 하이퍼파라미터이고 이 논문에서는 0.5를 주었다. 결과적으로 실험 모델들 중 성능이 가장 뛰어났다. 아두이노 보드를 이용하여 상황과 맥락을 파악하고 분석할 수 있는 웨어러블 디바이스를 제작하기 앞서, 아두이노 회로 제작과 코드 개발에 대한 이해를 위해 선행 연구로서 압력센서와 자이로 센서를 이용하여 간단한 조작을 실행할 수 있는 기능을 가진 장갑형 input device를 제작하였다. 회로는 바느질이 가능한 전도성 실로 제작되었으며 따라서 2차원에서 벗어난 형태의 회로가 제작될 수 있었다. 개략적인 회로 배치는 아래 그림과 같다

제 4 장 실험 및 결과

제1절 데이터 세트

이 논문에서는 실험들을 통해 다음의 두 가지 research question에 대해 답한다.

RQ1. 새로운 상품에 대한 평가정보가 없을 때 발생하는 ‘Cold Start’ 문제를 상품의 메타데이터 이미지 데이터를 GMF 와 결합하여 모델을 만들었을 경우 ‘Cold Start’ 문제를 잘 해결하는지?

RQ2. 제안한 VGMF와 MLP를 결합한 HVRs는 VGMF에 비해 전체적인 성능이 향상되는지?

<표 4-1> Tradesy.com 데이터 세트

Dataset	Data Preprocessing	Users (명)	Items (개)	feedback
Tradesy	Before	33,864	326,393	655,409
	After	8,355	258,943	492,350

데이터 세트는 tradesy.com 데이터세트를 사용하였다. tradesy.com 데이터 세트는 중고의류거래 사이트 ‘tradesy’에서 수집한 정보를 바탕으로 하

고 유저의 구매기록과 상품에 대해 ‘좋아요’를 표시한 데이터를 포함하고 있다. 상품의 57% 가 1번만 판매되었고 94% 가 5번 미만 판매되었다. 이 데이터 세트는 상품에 대한 사용자의 충분한 평가 데이터가 없어서 발생하는 ‘Cold Start’ 문제를 가지고 있기 때문에 이 논문의 추천시스템 모델이 기존의 모델들에 비해 ‘Cold Start’ 문제를 얼마나 잘 해결하는지 보여줄 수 있다. 기존에 데이터세트와 다르게 상품을 20개미만 평가한 사용자들은 모두 제외하고 20개 이상 평가한 사용자들로 데이터를 재구성하였다.

사용자가 상품에 남긴 평가데이터는 explicit과 implicit으로 구분할 수 있다. explicit은 별점데이터 평점데이터와 같이 1에서 5점까지 상품에 대해 점수를 매기는 것이고 implicit 데이터는 페이지 방문횟수, 상품페이지 클릭횟수 등 직접적으로 사용자가 상품에 평가를 내린 것이 아닌 간접적인 평가데이터이다. tradesy.com 데이터세트는 이중 implicit 데이터이고 사용자가 구매했을 경우 1 그렇지 않을 경우 0 이다. 이때 explicit 과 다르게 1이라는 의미는 사용자가 상품을 좋아한다는 뜻이 아니다. 또한 평가점수가 0이라는 것도 사용자가 상품을 싫어한다는 의미가 아니다. 사용자가 상품을 구매했지만 맘에 들어 하지 않을 수 있고 구매하지 않은 상품은 아직 발견하지 못한 상품일 수 있다.

이 논문에서는 상품에 대한 평가가 부족하여 발생하는 ‘Cold Start’문제를 해결하기 위해 상품의 ‘시각적 특징’을 추가로 사용하여 모델의 성능을 높였다. ‘시각적 특징’은 상품의 사진에서 추출한 정보로 합성곱 신경망 구조를(vbpr참고) 이용하였다. 여기서 ‘시각적 특징’ 추출기로 사용된 합성곱 신경망 구조는 120만장의 이미지넷(ILSVR2010) 사진을 통해 사전학습된 모델로 ‘시각적 특징’으로 사용된 데이터는 이 모델의 두번째 완전 연결 계층을 거쳐 생성된 4096차원의 시각적 특징 벡터이다.

제2절 학습 및 결과

이 논문에서 학습하고자 하는 값은 parameter θ 이다. 식으로 표현하면 다음 식 (15)와 같다

$$\underset{\theta}{argmax} P(y|x;\theta) \quad (15)$$

관측된 x 라는 데이터와 parameter θ 를 통해서 실제 예측 값 y 를 구할 수 있는데 이때 y 가 나올 확률을 최대로 하는 θ 를 학습시켜서 정해주는게 이 실험의 목표이다. 일반적으로 예측값과 실제값을 차이를 최소화하기 위해서 square loss를 사용하지만 이 논문의 데이터는 0과 1로 이루어져 있기에 bionomial distribution이다. 따라서 아래와 같은 negative log-likelihood 를 손실함수로 사용할 수 있다.

$$p(positive, negative|P, Q, \Theta_f) = \prod_{(u,i) \in positive} \hat{y}_{ui} \prod_{(u,j) \in negative} (1 - \hat{y}_{uj}) \quad (16)$$

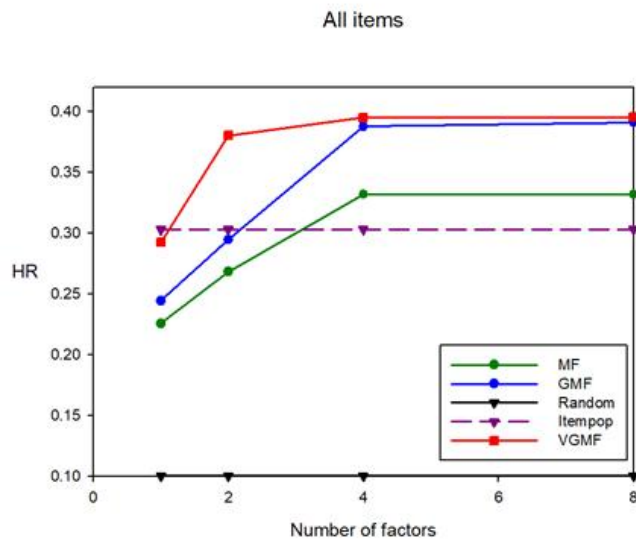
$$\begin{aligned} L &= - \sum_{(u,i) \in positive} \log \hat{y}_{ui} - \sum_{(u,i) \in negative} \log(1 - \hat{y}_{uj}) \\ &= - \sum_{(u,i) \in positive \cup negative} y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui}) \end{aligned} \quad (17)$$

(16)의 식은 평가점수가 1인지 0인지 구하기 위한 likelihood 이고 (17)의 식은 손실함수 식이다. 이 손실함수가 최소가 되도록 파라미터를 학습하는 것이 이 실험의 목표이다.

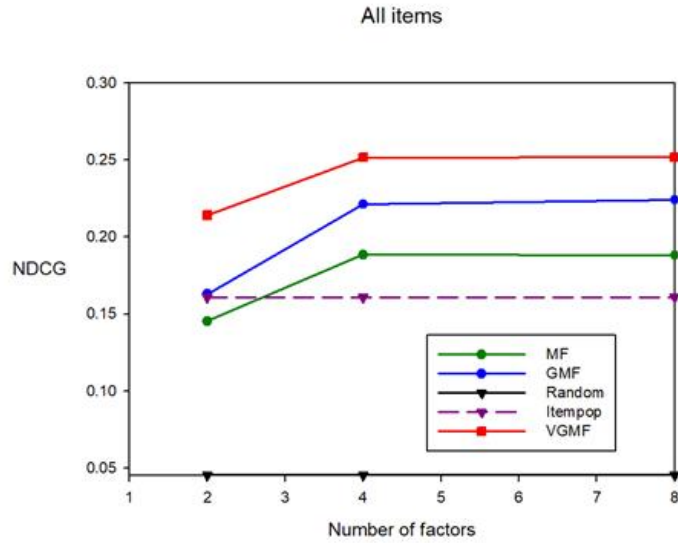
학습은 learning rate는 0.03, 0.01, 0.003, 0.001 에서 실험하였고 배치 사이즈는 2¹⁰, 2¹¹, 2¹², 2¹³에서 실험하였고 latent factor는 1,2,4,8에서 실험하였고 MLP레이어는 히든레이어 개수를 0,1,2개 각각 실험하였다. 임

베딩레이어의 파라미터는 가우시안 분포로 초기화 하였고 ReLu를 사용하는 히든레이어에서는 He normal 분포를 사용하였고, sigmoid를 사용하는 마지막 레이어에서는 Xavier 분포를 사용하였다

1. Latent factor



<그림 4-1> facotr 개수에 따른 HR@10의 변화



<그림 4-2> facotr 개수에 따른 NDCG@10의 변화

<표 4-2> facotr 개수에 따른 HR@10과 NDCG@10의 변화

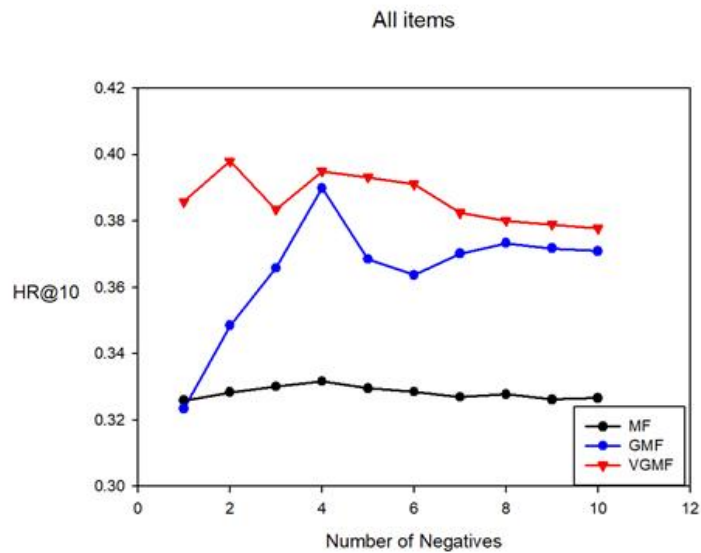
All items		Number of facotrs			
		1	2	4	8
Random	HR@10	0.1002	0.1002	0.1002	0.1002
	NDCG@10	0.0454	0.454	0.0454	0.0454
itempop	HR@10	0.3029	0.3029	0.3029	0.3029
	NDCG@10	0.1608	0.1608	0.1608	0.1608
MF	HR@10	0.2257	0.2682	0.3317	0.3318
	NDCG@10	0.1172	0.1452	0.1885	0.1881
GMF	HR@10	0.2443	0.2944	0.3877	0.3911
	NDCG@10	0.1317	0.1629	0.2213	0.2242
VGMF	HR@10	0.2922	0.3801	0.3950	0.3953
	NDCG@10	0.1607	0.2141	0.2516	0.2518

latent factor의 개수가 늘어날수록 사용자와 상품에 대해서 더 잘 표현해 줄 수 있다. 따라서 latent factor의 개수가 늘어날수록 모델의 성능은 좋아진다. 하지만 사용자와 상품에 대한 데이터가 얼마 없는 상태에서 많은 수의 latent factor를 고려하게 되면 ‘overfitting’ 문제가 발생한다. 즉 적

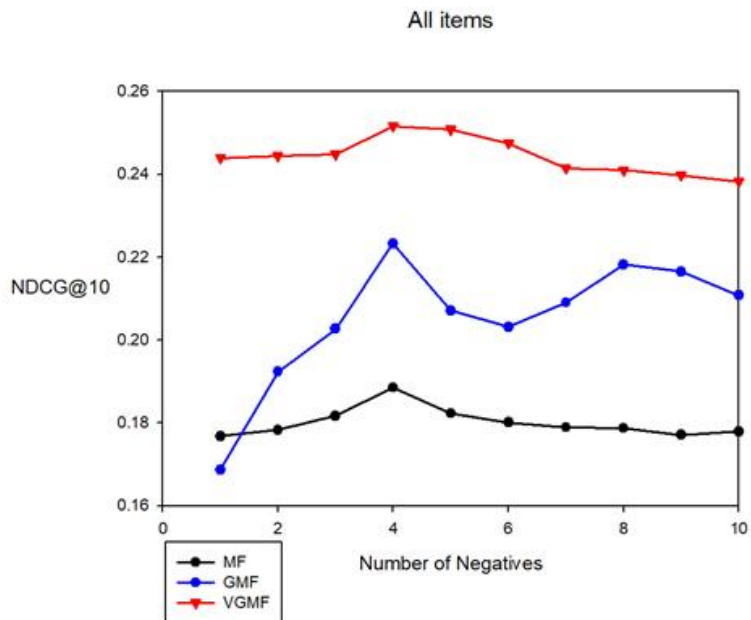
은 데이터로 많은 수의 latent factor를 설명하려 했기 때문이다. 그래서 학습데이터에서는 성능이 좋았지만 테스트데이터는 성능이 떨어지는 현상이 발생한다. latent factor의 개수가 1부터 4까지 늘어날수록 성능이 향상됨을 볼 수 있다. 하지만 4부터 8까지는 그 이상의 latent factor에서는 더 이상의 향상을 보이지 않고 비슷한 성능을 보이다 8이상의 latent factor에서는 오히려 성능을 떨어뜨렸다.

Random 모델이 가장 성능이 떨어지고 그 뒤를 Itempop이 잇는다. Matrix Factorization은 오늘날 연구에서 뛰어난 성능을 자랑하지만 선형 모델이기에 사용자와 상품사이의 복잡한 관계를 효과적으로 모델링할 수 없다. latent factor가 4까지 증가하는 동안 Matrix Factorization의 성능은 증가했고 8이상에서는 오히려 성능이 떨어졌으며 GMF 다음으로 성능이 좋았다. GMF 모델은 VGMF 다음으로 성능이 높게 나타났다. GMF 또한 latent factor의 개수가 4개일 때까지 성능이 증가했으며 8까지 성능이 비슷하다 8이상에서 성능이 떨어졌다. VGMF는 성능이 가장 뛰어났으며 latent factor의 개수가 1개일때도 Itempop모델과 성능이 비슷했다. 대체적으로 latent factor가 1 부터 4까지는 성능이 확연히 상승했지만 8이상부터는 성능이 오히려 떨어졌다. 적정 latent factor는 4로 정하고 모든 성능 평가를 실시했다

2. Number of Negatives



<그림 4-3> Positive당 Negative 개수에 따른 HR@10 변화



<그림 4-4> Positive당 Negative 개수에 따른 NDCG@10 변화

<표 4-3> Positive당 Negative 개수에 따른 HR@10과 NDCG@10의 변화

All items		Number of Negatives				
		1	2	3	4	5
MF	HR@10	0.3259	0.3284	0.3301	0.3317	0.3296
	NDCG@10	0.1768	0.1783	0.1817	0.1885	0.1823
GMF	HR@10	0.3234	0.3484	0.3658	0.3899	0.3685
	NDCG@10	0.1687	0.1923	0.2027	0.2233	0.2071
VGMF	HR@10	0.3858	0.3980	0.3835	0.3950	0.3932
	NDCG@10	0.2439	0.2444	0.2448	0.2516	0.2508

all items		Number of Negatives				
		6	7	8	9	10
MF	HR@10	0.3285	0.3269	0.3278	0.3262	0.3266
	NDCG@10	0.1801	0.1789	0.1787	0.1771	0.1779
GMF	HR@10	0.3637	0.3702	0.3733	0.3717	0.3709
	NDCG@10	0.2031	0.2090	0.2182	0.2165	0.2108
VGMF	HR@10	0.3911	0.3825	0.3801	0.3789	0.3777
	NDCG@10	0.2475	0.2415	0.2410	0.2398	0.2383

BPR 모델은 하나의 positive instace당 하나의 negative instance만 샘플링하여 모델의 성능을 평가했지만 [NCF] 논문에서는 하나의 positive instance당 negative instance의 비율을 조절하여 실험을 해보았다. 이 논문에서는 [25]의 연구와 같이 negative instance 샘플링 비율을 늘리면서 실험을 하였고 샘플링 비율이 늘어날수록 성능이 향상됨을 보였다. Matrix Factorization 모델은 샘플링 비율이 1부터 4까지 성능에 차이가 없다가 그 이상의 비율에선 성능이 오히려 떨어졌다. GMF와 VGMF의 경우 샘플링 비율이 각각 4일 때 성능이 좋았다.

결과적으로 negative instance의 비율이 1에서 4일 때 두드러지게 성능이 향상되었고 5이상에서는 대체적으로 성능이 더 떨어졌다. negative instance 샘플링 비율이 높아질수록 성능은 높아지지만 너무 높다면 성능은 떨어지고 적절한 샘플링 비율은 4로 볼 수 있다.

3. Multilayer Perceptron Architecture

MLP에서는 서로 복잡한 상호작용을 가지는 두개의 latent factor를 concatenate하고 Neural network를 통과시킴으로써 학습하였다. latent factor를 학습시키기 위해서 hidden layer와 latent factor의 길이에 따라서 실험하였고 결과는 아래와 같다.

<표 4-4> MLP 구조 변화에 따른 HR@10 과 NDCG@10

hidden layers	Architecture	Latent factor	HR@10	NDCG@10
0	layers (2-1)	2	0.2080	0.1130
0	layers (4-1)	4	0.2072	0.1129
0	layers (8-1)	8	0.2099	0.1145
0	layers (16-1)	16	0.2084	0.1121
0	layers (256-1)	256	0.2034	0.1091
1	layers (4-2-1)	4	0.2743	0.1252
1	layers (8-4-1)	8	0.2930	0.1631
1	layers (16-8-1)	16	0.3315	0.2097
1	layers (32-16-1)	32	0.3430	0.2249
1	layers (64-32-1)	64	0.3402	0.2248
1	layers (128-64-1)	128	0.3184	0.2081
2	layers (32-16-8-1)	32	0.2397	0.1181

hidden layer가 0개인 모델에서는 latent factor의 길이가 늘어남에 상관없이 HR과 NDCG 값에 큰변화가 없었다. hidden layer가 1개일 때 latent factor의 길이가 늘어날수록 성능이 향상되었고 32-16-1 구조에서 성능이 가장 좋았고 그 이상에서는 성능이 오히려 떨어졌다. hidden layer가 2개일 때부터는 오버피팅이 발생했다. MLP는 데이터사이의 복잡한 상호작용을 모델링 할 수 있지만 데이터가 충분하지 못할 때 성능이 잘나오지 못하는 단점이 있다. 즉 대부분이 'Cold Start'인 tradesy.com 데이터 세트 특성상 MLP 구조의 성능이 잘 나올 수 없지만 VGMF와 Hybrid 하게 될 경우 상품에 대해 평가정보가 많을 때와 적을 때 모두 latent factor 값을 잘 학습시켜주기 때문에 성능이 향상된다

제3절 평가 방법 및 토의

이 논문에서는 [22,29,33] 연구에서 사용된 leave-one-out 평가방식을 사용하였다. leave-one-out은 하나의 데이터만 테스트세트로 사용하고 나머지는 학습데이터로 사용하는 방법이다. tradesy.com데이터 세트의 각 사용자가 마지막으로 평가한 상품을 테스트세트로 사용하였고 나머지는 학습데이터로 사용하였다.

기존의 추천시스템에서는 RMSE 를 평가지표로 하여 RMSE가 가장 낮은 모델이 성능이 가장 좋은 모델이라고 평가를 하였다. 하지만 사용자가 상품에 매기는 평점은 많은 편향을 가지고 있기 때문에 실제점수를 예측을 잘하는 모델이 사용자를 만족하는 상품을 추천해준다고 보기 어렵다. 따라서 이 논문에서는 평가지표로 Hit Ratio(HR)와 Normalized Discounted Cumulative Gain(NDCG)[25, 34]를 사용하였다. 모든 상품중 무작위로 99개의 상품을 추출하여 각 사용자의 마지막 평가 상품과 더하여 100개의 추천 상품리스트를 만든 후 학습 한 모델이 사용자가 실제로 마지막으로 평가한 아이템이 높은 순위에 위치하는지 HR과 NDCG의 평가기법으로 나타내었다. HR@10, NDCG@10 을 기준으로 모델의 성능을 평가하였고 top-n의 변화에 따른 모델의 성능도 나타내었다. HR@10은 실제로 사용자가 평가한 상품이 top-10 내에 들어가면 1, 그렇지 않으면 0으로 평가하여 모든 사용자들의 HR의 값을 더한후 평균내었다. HR은 순서와 상관없이 top-10 내에 들어가기만 하면 1로 평가했지만 NDCG는 top-10내에서 더 높은순위에 위치할수록 높은 점수로 평가하였고, top-10내에 들지 못하면 0의 값을 주었다. 그리고 모든 사용자들의 NDCG값을 더한 후 평균을 내었다.

평가하고 서로 비교할 모델은 다음과 같다.

Random : 각 사용자를 평가할 100개의 상품 중에 무작위로 top-10 상품을 추출하여 추천 리스트를 만든 후 실제 평가한 상품이 리스트 안에 들어가 있는지 평가하였다.

itempop : 상품이 평가받은 횟수에 따라 인기를 정하였다. 각 사용자가 평가할 100개의 상품들을 인기도 순으로 나열하고 top-10 리스트 안에 실제로 사용자가 평가한 상품이 들어가 있는지 평가하였다.

Matrix Factorization : 기존에 추천시스템에서 자주 사용하는 선형모델이다. 평가행렬을 k 차원의 길이를 가지는 두 개의 latent factor로 분해하고 학습하여 각 사용자가 평가할 100개의 상품들의 평가점수를 근사하였다. 그 후 높은 순서대로 나열하고 사용자가 실제로 평가한 상품이 top-10 리스트에 들어가 있는지 평가하였다.

Multilayer Perceptorn : 사용자와 상품 각각의 latent factor를 concatenate하고 neural network에 통과시킨 모델이다. 사용자와 상품 사이의 복잡한 관계를 표현하는 모델을 만들 수 있다.

Generalized Matrix Factorization[25] : Matrix Factorization을 neural network로 일반화 시킨 모델이다. 마지막 레이어에서 각 latent factor에 가중치를 줘서 표현력이 풍부한 모델을 만들었다. 또한 활성화 함수로 비선형 함수인 sigmoid 함수를 사용했기에 비선형성을 가진다. 평가 방법은 Matrix Factorization과 같다.

Visual based Generalized Matrix Factorization : 이 논문에서 사용한 tradesy.com의 데이터세트와 같은 경우 대부분의 데이터가 상품에 대한 평가정보가 부족한 'Cold Start' 문제를 가진다. 상품에 대한 평가정보가 부족할 경우 평가정보 이외의 상품에 대한 메타데이터를 같이 사용하여 모델의 성능을 높여줄 수 있다. 이 논문에서는 의류상품을 사용자가 시각적인 정보를 고려하여 구매한다고 가정하고 의류 이미지데이터를 메타데이터로 사용하여 모델을 만들고 성능을 평가하였다.

hybrid visual based recommender system : 상품의 메타데이터, 이미지 데이터를 사용하여 만든 VGMF모델과 사용자와 상품사이의 복잡한 관계를 표현하는 MLP 모델을 hybrid 하여 만든 모델이다.

<표 4-5> All items와 Cold items 세팅에서 각 모델의 성능 비교

Metric	Setting	(a)	(b)	(c)	(d)	(e)	Improvement		
		MF	MLP	GMF	VGMF	VGMF + MLP	(d) vs (a)	(d) vs (c)	(e) vs (d)
HR@10	All items	0.3317	0.3430	0.3877	0.3950	0.4115	19.1%	1.9%	4.2%
	Cold items	0.2384	0.2861	0.3123	0.3203	0.3311	34.4%	2.6%	3.4%
NDCG@10	All items	0.1885	0.2249	0.2213	0.2516	0.2737	33.5%	13.7%	8.8%
	Cold items	0.1132	0.1619	0.1631	0.1936	0.2088	71.0%	18.7%	7.9%

위의 표를 보면 VGMF는 선형모델의 한계를 가진 MF에 비해 HR@10, NDCG@10 각각 19.1%, 33.5%의 성능 향상을 보였고 MF를 neural network로 일반화시켜서 비선형성을 추가해준 GMF에 비해서는 각각 1.9%, 13.7%의 성능향상을 보였다. 또한 사용자와 상품사이의 복잡한 상호작용을 표현할수 있는 모델인 MLP와 이 논문에서 제안한 VGMF를 Hybrid 한 모델은 VGMF 보다 각각 4.2%, 8.8 %의 성능향상을 보였다. 이 결과로 보아 상품의 메타데이터인 이미지 데이터를 활용할 경우 그렇지 않았을 경우에 비해 성능이 향상된다는 것을 알 수 있다.

이 논문에서는 사용자가 실제로 평가한 상품이 5번 미만 평가 받은 아이템 즉 ‘Cold Start’ 문제를 가지는 테스트 데이터세트를 임의로 만들었다. 위의 표를 보면 평가정보가 얼마 없는 상품들로 모델을 평가하였기 때문에 itempop 같은 경우 Random과 비슷한 성능을 보인다. ‘All items’ 세팅일 때 MF와 GMF에 비해 NDCG@10 각각 33.5%, 13.7%의 성능 향상을 보이던 VGMF는 ‘Cold items’ 세팅일 때 MF와 GMF에 비해 NDCG@10 각각 71.0%, 18.7%의 성능 향상을 보였다. MLP 모델을 VGMF와 결합할 경우 VGMF에 비해 모델의 성능이 NDCG@10 각각 8.8%, 7.9% 향상됨을 알 수 있었다.

MF는 선형모델의 한계를 가지고 있지만 [25]의 연구에서는 neural network로 MF를 일반화 하여 GMF를 만들고 선형모델의 한계를 극복하였다. 하지만 MF와 GMF의 경우 새로운 상품을 추천해줘야 할 경우 상품에 대한 평가정보가 없어서 발생하는 ‘Cold Start’ 문제를 겪게 되는데 이때 기존의 GMF에 상품의 메타데이터인 이미지 데이터를 활용하여 결합함으로써 ‘Cold Start’ 문제를 해결했을 뿐 아니라 모델의 전체적인 성능을 향상시켰다. 또한 서로 복잡한 상호작용을 가지는 사용자 latent factor와 상품 latent factor를 학습시킬 수 있는 MLP와 VGMF를 결합함으로써 VGMF보다 성능을 향상 시켰다.

제 5 장 결론 및 논의

제1절 요약 및 결론

오늘날 많은 양의 데이터가 빠른 속도로 생성 되고 사용자는 이러한 정보 속에서 원하는 정보를 찾기가 더 어려워졌다. 이런 ‘정보 과부하’로 매 주 매 계절 새로운 옷이 쏟아지는 의류 도메인에서 사용자는 상품을 구매할 때 자신의 취향을 고려한 상품을 찾기 어렵고 의류를 파는 기업입장에서는 일부분의 상품만을 사용자가 조회하기 때문에 인기 있는 상품을 제외한 대부분의 상품은 잘 팔리지 않는다. 추천시스템은 이러한 ‘정보 과부하’ 문제에 대처하고 사용자에게는 유용한 정보를 제공, 기업에게는 보다 많은 상품을 팔게 해주는 유용한 어플리케이션이다.

추천시스템은 크게 콘텐츠 기반 필터링과 협업필터링과 하이브리드 기반 추천시스템으로 분류된다. 콘텐츠 기반 필터링은 상품의 정보를 이용해 사용자에게 추천을 하기 때문에 ‘Cold Start’ 문제에 대처할 수 있지만 비슷한 상품만을 추천해주고 참신한 상품을 추천해주지 못한다는 단점이 있다. 협업필터링은 메모리 기반 협업필터링과 모델 기반 협업필터링으로 나눌 수 있는데 메모리 기반 협업필터링은 또 사용자 기반 협업필터링과 아이템 기반 협업필터링으로 나눌 수 있고 모델기반 협업필터링은 크게 SVD와 Matrix Factorization으로 나눌 수 있다. 메모리기반 협업필터링은 사용자에게 설명력 있는 추천을 해줄 수 있지만 대용량 데이터셋에서 사용자나 상품사이의 유사도를 구할 때 연산시간이 아주 오래 걸리고 ‘Cold Start’ 문제를 가진다. 모델기반 협업필터링의 Matrix Factorization은 사용자와 상품의 평가행렬을 k 차원 벡터로 분해하고 k 차원 벡터를 학습한후 평가행렬을 다시 근사함으로써 사용자의 상품에 대한 평가점수를 예측할 수 있으며 단일 모델중 뛰어난 성능을 보이지만 역시 ‘Cold Start’ 문제를 가진다. 하이브리드 기반 추천시스템은 협업필터링과 콘텐츠 기반 필터링을 앙상블한 모델로 ‘Cold Start’ 문제에 대처하고 참신한 상품을 추천해줄 수 있다.

[25] 연구에서는 기존에 뛰어난 성능을 보이는 Matrix Factorization 모델이 가지는 한계점을 Neural network를 사용하여 일반화된 Generalized Matrix Factorization로 극복하였다. Generalized Matrix Factorization은 기존의 Matrix Factorization에서 분해된 각각의 latent factor를 element-wise하고 Neural network를 통과시켜서 각각 latent factor에 가중치를 주어 모델의 표현력을 풍부하게 해주었고 비선형함수인 sigmoid 함수를 활성화함수로 사용함으로써 선형모델인 Matrix Factorization에 비해 비선형모델을 만들어서 성능을 높였다. 하지만 GMF의 경우에도 협업 필터링과 마찬가지로 ‘Cold Start’ 문제를 가지는데 특히 의류도메인 내에서 새로운 상품이 들어왔을 때 상품에 대한 정보가 부족하여 모델을 잘 학습시켜주지 못한다.

‘Cold start’ 문제에 대처하기 위해 이 논문에서는 사용자가 의류상품을 구매할 때 시각적인 정보를 고려한다는 가정을 하고 상품의 메타 데이터인 이미지 데이터를 사용하여 만든 VGMF 모델의 성능을 평가하였다. 이때 이미지 데이터는 pretrain 한 Convolutional Neural Network의 두번째 Fully Connected Layer를 통과시켜 나온 4096길이의 벡터를 사용하였다. ‘Cold Start’ 문제에 대처하는 모델을 만들기 위해 모델 학습용 데이터 세트는 94%가 5번 미만 평가된 상품이 존재하는 tradesy.com을 사용하였다. 또한 실제로 각 사용자가 테스트 세트에서 평가한 상품이 5번 미만 평가된 상품이었을 경우 ‘cold items’로 지정하고 새로운 테스트 세트를 만들어서 VGMF가 ‘Cold Start’ 일때 다른 모델에 비해 높은 성능을 가지고 있음을 보였다. 평가방법은 leave-one-out으로 사용자가 마지막에 구매한 상품만 테스트세트로 사용하고 나머지는 학습용세트로 사용하는 방식이다. 평가지표로는 Hit Ratio(HR) 와 Normalized Discounted Cumulative Gain(NDCG)를 사용하였다. latent factor의 크기가 4, Positive와 negatives의 비율이 4:1일 때 VGMF 성능이 가장 높게 나왔다. 또한 All items과 Cold items 세팅일 때 GMF에 비해 각각 10.9%, 18.7%의 성능향상을 보임으로써 상품의 메타정보를 사용한 VGMF가 GMF에 비해 전체적인 성능이 좋을 뿐만 아니라 ‘Cold Start’ 문제에 더

잘 대처하는 모델임을 알 수 있었다. 또한 복잡한 상호작용을 가지는 두 개의 latent factor를 잘 학습시킬 수 있는 Multi Layer Perceptron과 VGMF를 Hybrid 하여 All items와 Cold items 세팅에서 성능이 각각 VGMF에 비해 8.8% , 7.9% 향상됨을 보였다. 최종성능으로 latent factor 길이가 4일 때와 Negative sampling 비율이 4일때 Hybrid Visual based Recommender System의 성능이 가장 뛰어났고 뒤이어 VGMF, GMF, MF, itempop, Random 순서로 성능이 좋았다.

기존의 선형모델로 한계를 가지는 MF와 MF를 변형시켜 만든 GMF는 'Cold start' 문제에 잘 대처하지 못했는데 GMF에 상품의 메타데이터를 Hybrid 함으로써 'Cold start' 문제를 해결 하였고 latent factor의 유사도를 기반으로 하는 MF와 GMF와 VGMF와 같은 협업필터링은 한계점을 가지는데 latent factor를 element-wise 하지않고 concatenate 하는 MLP 모델로 latent factor를 학습하여 다시한번 VGMF와 Hybrid 하여 최종적으로 'Cold start' 문제에 잘 대처하고 전체적인 성능이 높은 Hybrid Visual based Recommender System을 제안하였다

제2절 향후 연구 방향

이 논문에서는 기존의 tradesy.com 데이터세트에서 임의로 상품에 평가한 횟수가 20회 이상인 사용자만을 추출하여 새로운 데이터세트를 만들었다. 그렇기 때문에 사용자의 정보는 풍부하다는 장점이 있다. 하지만 real-world에서는 평가횟수가 부족한 사용자에게도 상품을 추천해주어야 하기 때문에 제안한 추천모델들을 적용하기 어렵다. 하지만 상품에 대한 평가정보가 부족할 때 상품의 메타데이터를 사용한 점과 동일하게 사용자의 나이, 성별, 취향 등의 개인정보를 사용하여 모델에 결합한다면 real-world에서 적용 가능한 모델을 만들 수 있다. 또한 이 논문에서는 상품의 이미지 만을 결합하여 모델을 만들었지만 고객의 리뷰데이터, 상품 클릭 횟수, 상품페이지에 머문 시간 등을 추가적으로 결합하는 모델을 만든다면 ‘Cold start’에 잘 대처하는 높은 성능을 가지는 추천시스템을 만들 수 있을 거라 예상된다.

참고문헌

서양문헌

학술지

- [1] Edmunds, A., & Morris, A. The problem of information overload in business organisations: A review of the literature. International Journal of Information Management, 20 (1), 17 - 28. (2000)
- [2] Vickery, A. ,& Vickery, B. C. Information science in theory and practice. De Gruyter. (2005).
- [3] Resnick, P., & Varian, H. R. Recommender systems. Communications of the ACM, 40 (3), 56 - 58. (1997).
- [4] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering 17, 6 (2005), 734 - 749.(2005)
- [5] Shuai Zhang, Lina Yao, and Aixin Sun. Deep Learning based Recommender System: A Survey and New Perspectives. ACM J. Comput. Cult. Herit. 1, 1, Article 35 (July 2017), 35 pages. (2017)
- [6] Linden, G., Smith, B., & York, J. Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7

- (1), 76 - 80. (2003).
- [7] Amatriain, X. (2013). Big & personal: data and models behind netflix recommendations. In Proceedings of the 2nd international workshop on big data, streams and heterogeneous source mining: Algorithms, systems, programming models and applications (pp. 1 - 6). ACM .
- [8] Nanas, N., Vavalis, M., & Houstis, E. Personalised news and scientific literature aggregation. Information Processing & Management, 46 (3), 268 - 283. (2010).
- [9] Gupta, P., Goel, A., Lin, J., Sharma, A., Wang, D., & Zadeh, R. (2013). Wtf: The who to follow service at twitter. In Proceedings of the 22nd international conference on world wide web . In WWW '13 (pp. 505 - 514). New York, NY, USA: ACM.
- [10] Wu, M.-L., Chang, C.-H., & Liu, R.-Z. Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices. Expert Systems with Applications, 41 (6), 2754 - 2761. (2014).
- [11] Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. Methods and metrics for cold-start recommendations. In Proceedings of the 25th annual international acm sigir conference on research and development in information retrieval . In SIGIR '02 (pp. 253 - 260). New York, NY, USA: ACM.(2002).
- [12] Papagelis, M. , Rousidis, I. , Plexousakis, D. , & Theoharopoulos, E. In M.-S. Hacid, N. V. Murray, Z. W. Ra 's, & S. Tsumoto

(Eds.), Incremental collaborative filtering for highly-scalable recommendation algorithms (pp. 553 - 561)). Berlin, Heidelberg: Springer Berlin Heidelberg.(2005).

- [13] Amini, M. , Nasiri, M., & Afzali, M. Proposing a new hybrid approach in movie recommender system. International Journal of Computer Science and Information Security, 12 (8), 40 - 45.(2014).
- [14] Ge, X., Liu, J., Qi, Q., & Chen, Z. A new prediction approach based on linear regression for collaborative filtering. In 2011 eighth international conference on fuzzy systems and knowledge discovery (FSKD): 4 (pp. 2586 - 2590). (2011).
- [15] Resnick, P., & Varian, H. R. Recommender systems. Communications of the ACM, 40 (3), 56 - 58. (1997).
- [16] Zhang, T., & Iyengar, V. S. Recommender systems using linear classifiers. Journal of Machine Learning Research, 2 , 313 - 334.(2002).
- [17] Musat, C.-C. , Liang, Y. , & Faltings, B. Recommendation using textual opinions. In Proceedings of the twenty-third international joint conference on artificial intelligence . In IJCAI '13 (pp. 2684 - 2690). AAAI Press.(2013).

- [18] Y. Koren, R. Bell and C. Volinssky, "Matrix Factorization Techniques for Recommender System," IEEE Computer, vol.42, no.8, pp.30–37, (2009).

- [19] Srivastava, N. , Hinton, G. , Krizhevsky, A. , Sutskever, I. , & Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15 (1), 1929 - 1958. (2014).

- [20] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4):541 - 551, (1989).

- [21] Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In NIPS, pp. 1106 - 1114, (2012).

- [22] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua. Fast matrix factorization for online recommendation with implicit feedback. In SIGIR, pages 549–558, (2016).

- [23] Y. Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In KDD, pages 426–434, (2008).

- [24] H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deeplearning for recommender systems. In KDD, pages 1235–1244, (2015).
- [25] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and Chua. Chua. Neural collaborative filtering. In WWW, (2017).
- [26] R. He and J. McAuley. Vbpr: Visual bayesian personalized ranking from implicit feedback. CoRR, (2015).
- [27] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In ICDM, pages 263–272, (2008).
- [28] Pan, R.; Zhou, Y.; Cao, B.; Liu, N. N.; Lukose, R.; Scholz, M.; and Yang, Q. One-class collaborative filtering. In ICDM. (2008)
- [29] Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. Bpr: Bayesian personalized ranking from implicit feedback. In UAI. (2009)
- [30] S. Rendle. Factorization machines. In ICDM, pages 995–1000, (2010).
- [31] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In

ICCV, (2015).

- [32] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In AISTATS, (2010).
- [33] I. Bayer, X. He, B. Kanagal, and S. Rendle. A generic coordinate descent framework for learning from implicit feedback. In WWW, (2017).
- [34] X. He, T. Chen, M.-Y. Kan, and X. Chen. TriRank: Review-aware explainable recommendation by modeling aspects. In CIKM, pages 1661–1670, (2015).

ABSTRACT

Development of a Collaborative Deep Learning System for Fashion Recommendation

Lee, Gwang Han

Department of Organic and Nano System Engineering

Graduate School of Konkuk University

In this paper, we developed a hybrid based recommender system that combines existing recommendation system with deeplearning to efficiently recommend fashion products considering user 's preference among newly emerging fashion products. Deep Learning is a technology that shows outperform in various fields such as image, video, natural language processing, speech recognition, etc., and there is an increasing number of studies to incorporate deep learning to effectively model the interaction between user and product in the recommender system domain . In the case of the SPA brand in the apparel fashion market, new apparel products are poured every week. Accordingly, the recommender system should recommend the product considering the preference to the user. However, since there is the evaluation of the user about the product, the user can learn the recommendation system model, which causes the problem of 'Cold Start'. In this study, suppose that the user considers the visual features of the apparel product, and coped with the 'Cold Start' problem by using the apparel product image data. In addition, this model is hybrid with Deep Learning model MultiLayer Perceptron to further improve the performance, thus suggesting a hybrid based deeplearning recommender system that can be used in the fashion

market.

The data set to learn the recommended system model was created by extracting only the users who evaluated 'Tradesy.com', which was crawled from the second-hand clothing site, and evaluated the product more than 20 times. The data set consists of 8,355 users and 258,943 products and 492,350 interactions. We also divide the proposed model into 'All items' and 'Cold items' settings to see if the 'Cold start' problem is solved well. The data set uses one data evaluated by each user as a test set and the rest as a training set. The experiment was carried out. To develop the model, we used Python, one of the programming languages, and Keras, one of the deeplearning frameworks, and experimented with intel® Core™ i5-8600 @ 3.10Ghz, 32GB, and 1080Ti, respectively, for cpu, memory and gpu. To evaluate the performance of the proposed Visual Generalized Matrix Factorization, we compared the models of Random, Itempop, Matrix Factorization, and Generalized Matrix Factorization. Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) were used as the evaluation metrics for comparing each model. The proposed VGMF model showed 10.9% and 33.5% improvement in performance in the 'All items' setting compared to GMF and MF (NDCG), and 18.7% and 71.0% performance improvement (NDCG) in the 'Cold items' setting respectively. In the case of the model made by hybrid of VGMF and MLP, the performance improvement (NDCG) was 8.8% and 7.9% in the 'All items' and 'Cold items' settings, respectively, compared to VGMF. If you use the image of the product when recommending the fashion product, you can see that the overall performance of the model is increased. Also, if you use the image of the product when the 'Cold start' occurs because there is no evaluation information about the product, It can be seen that it can cope well with existing models.

In this study, we proposed a fashion product recommender system model considering user 's preference by combining deeplearning which is actively studied recently with recommender system. Although we tried to improve the performance by using only image information among the metadata of the product, in future studies, if we combine various metadata such as the user's review data, the number of clicks of the product web page, the age of the user and the gender of the user into the recommender system model, Personalized recommendations will be made.

Keyword : Recommender System / Artificial Intelligence / Deep Learning