딥 오토인코더 기반의 개인화 추천 시스템
- 사용자의 내재적 특성 고려

Deep AutoEncoder based Personalized Recommendation System : Considering user's intrinsic characteristics

저자
(Authors)

제현우, 김준우, 이문용
Hyunwoo Je, Junwoo Kim, Mun Y. Yi

출처
(Source)

한국정보과학회 학술발표논문집 , 2017.06, 773-775 (3 pages)

발행처
(Publisher)

한국정보과학회
KOREA INFORMATION SCIENCE SOCIETY

URL

http://www.dbpia.co.kr/Article/NODE07207377

APA Style

제현우, 김준우, 이문용 (2017). 딥 오토인코더 기반의 개인화 추천 시스템. 한국정보과학회 학술발표논문집, 773-775.

딥 오토인코더 기반의 개인화 추천 시스템: 사용자의 내재적 특성 고려

제현우 ○, 김준우, 이문용

한국과학기술원 지식서비스공학과

jhw4824@kaist.ac.kr, junu@kaist.ac.kr, munyi@kaist.ac.kr

Deep AutoEncoder based Personalized Recommendation System: Considering user's intrinsic characteristics

Hyunwoo Je ○, Junwoo Kim, Mun Y. Yi

Department of Knowledge Service Engineering, KAIST

요 약

사용자에게 특정 상품을 추천하는 대표적인 방법론은 협업 필터링이다. 그러나 해당 방법은 콜드 스타트 문제가 발생할 때, 상당히 취약하다는 한계가 있다. 그렇기에 해당 문제를 해소하기 위해, 리뷰와 같은 자연어 정보와 사용자의 고유 특성을 나타내는 추가 정보들을 이용할 필요가 있다. 본 연구에서는 이러한 추가 정보를 효과적으로 사용하기 위해 Stacked Denoising AutoEncoder를 기반으로 하는 추천 시스템을 제안한다. 사용자 기반의 추가 정보를 효과적으로 사용하기 위해 두 개의 독립된 오토인코더가 공유된 hidden layer를 갖도록 모델을 설계하였다. 실험 결과, 제안하는 모델이 비교 모델인 SVR, Matrix Factorization, 그리고 오토인코더보다 더 나은 성능을 나타내었다. 그렇기에, 본 연구에서 제안하는 모델의 타당성을 알아볼 수 있었고, 오토인코더와 각종 추가 정보를 결합한 연구가 계속 필요함을 시사하고 있다.

1. 서 론

소비자 개개인의 취향이 다르기 때문에 소비자들에게 상품을 추천하는 것은 쉬운 일이 아니다. 고객의 취향은 겉으로 드러나지 않는다. 그렇기에 소비자들의 상품 구매 패턴을 분석함으로써 앞으로 고객이 구매할 확률이 높은 상품을 추출하는 일이 중요해지고 있다. 이러한 일을 하기 위해, 과거에는 전문 상품 큐레이터가 있었다. 요즘에는 하루에도 수많은 새로운 상품들이 등장하기 때문에 큐레이터가 일일이 고객에 맞는 개인화된 추천을 제공하기가 거의 불가능하다. 그렇기에 자동으로 추천 업무를 해주는 추천 시스템의 필요성이 증대되고 있다. 추천 시스템은 학계뿐만 아니라, 산업계에도 활용되어 큰 영향을 미치고 있는데 대표적인 기업으로는 아마존과 넷플릭스가 있다[1]. 이러한 기업들은 추천 시스템을 활용하여 그들의 매출을 유의하게 증가시켰다는 보고가 있다.

추천 시스템에는 크게 2가지 방법론이 있다. 첫째는 memory based 방식이며, 다음으론 model based 방식이다. Memory based 방식은 특정 유저와 비슷한 유저를 찾아 추천을 제공하려는 생각이다. 즉, 비슷한 유저들이 구매한 상품을 해당 유저도 구매할 가능성이 높다는 가정을 하고 있다[2]. 유사도를 계산할 때, 유저뿐만 아니라 상품을 기준으로 하는 방식도 가능하다[3]. Model based 방식은 사전에 정의된 매개변수를 최적화하는 과정을 통해 모델을 만드는 접근이다. 대표적인 방법은 Matrix Factorization(MF)이 있다[4]. MF는 사용자와 상품의 latent feature를 평점 정보가 있는 행렬을 활용해 벡터 형식으로 추출한다.

이러한 두 가지 방법론에서 다양한 추가 정보를 사용해 추천 시스템 성능을 높이려는 시도 역시 존재한다. 추가 정보를 사용하는 이유는 평점 행렬에서는 평점 정보의 절대적인 부족으로 콜드 스타트 문제가 발생하기 때문이다. 그렇기에 해당 문제를 완화하기 위해서 리뷰에 있는 자연어 정보를 사용하거나, trust network와 같은 그래프 구조를 사용하여 추천 성능을 높이려고 시도하고 있다[5].

컴퓨터의 연산 능력과 데이터가 기하급수적으로 증대되고 있는 이때, 추천 시스템에도 딥러닝의 역할이 중요해지고 있다. CNN과 DBN을 활용한 추천 시스템 연구를 넘어, 요즘에는 기존의 협업 필터링에서 단점으로 알려진 콜드 스타트 문제를 완화하기 위해, 오토인코더를 사용한 추천을 시도하고 있다. 또한, 오토인코더의 구조

* 본 연구는 산업통상자원부 및 한국산업기술평가관리원의 산업핵심기술개발사업(지식서비스)의 일환으로 수행하였음. [10052955, 현장 전문가의 경험지식 획득 및 활용을 위한 경험지식플랫폼 개발 연구]

를 확장하여 멀티 레이어를 가진 오토인코더를 사용하기도 하며, 인풋 노드에 노이즈를 삽입함으로써 더욱 안정적인 hidden representation을 추출하는 방식을 사용하기도 한다[6].

본 연구에서는 Stacked Denoising AutoEncoder (SDAE)를 사용한 추천 시스템을 제안한다. 기존의 콜드스타트 문제를 완화시키기 위해 리뷰에 있는 자연어와 사용자의 고유한 내재적 특성을 나타내는 정보를 추가 정보로 사용하였다. 또한, 두 가지 이질적인 정보를 통합하기 위해서 hidden layer가 공유되고 있는 구조를 사용하였다. 선행 연구에 따르면 해당 구조는 trust network를 통합시키기 위한 적절한 구조임이 밝혀졌다[7]. 그렇기에 본 연구는 해당 구조를 활용하여 사용자의 내재적 특성을 나타내는 추가 정보를 통합시킴으로써 추천 성능을 향상 시킬 수 있음을 보였다.

2. Stacked Denoising AutoEncoder

오토인코더는 데이터 압축 알고리즘 중 하나로써 기존에는 주로 영상처리 분야의 연구 주제였지만, 최근에는 추천 시스템 분야에서도 많이 연구되고 있는 알고리즘이다.

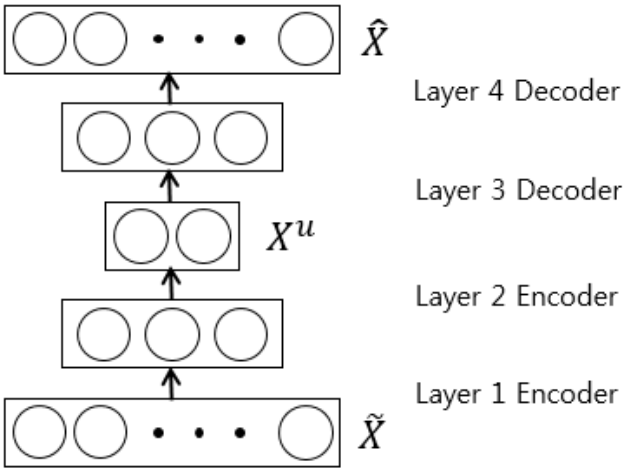

그림 1 SDAE

오토인코더는 인코더, 디코더로 구성되는데, 목적은 오토인코더를 거친 아웃풋을 인풋과 동일하게 만드는 것이다. 이때, hidden layer에서 추출되는 low dimension의 feature 벡터는 인풋 데이터의 특징을 잘 추출한 정보가 된다. 즉, 해당 압축된 정보를 사용한다면 적절한 디코더가 존재할 때, 원래 데이터로 복구할 수 있다는 것이다.

$$\text{Loss} = ||\tilde{X} - \hat{X}||_2^2 + \lambda \sum_l^L (||W_l||_2^2 + ||b_l||_2^2) \quad (1)$$

손실 함수를 수식 1과 같이 정의한 후, 모델을 트레이닝 시킨 후의 아웃풋은 sparse하지 않은 dense 아웃풋 형태가 된다. 그 뿐만 아니라, 인풋에 노이즈를 넣거나 hidden layer를 추가함으로써 더 나은 추천 성능을 보인 연구들도 존재한다[6]. 또한, 인풋에 단순히 평점 정보 이외에 추가 정보까지 고려한 시도도 있었다[8]. 하지만 추천 성능의 향상은 거의 일어나지 않았다. 이는 단순히 추가 정보를 인풋 노드에 넣는 것이 무조건적인 성능 향상을 가져오지 않는다는 것을 시사한다. 이는 어떤 오토인코더 구조를 사용하느냐에 따라서, 추가 정보의 사용이 추천 성능에 미치는 영향이 달라진다는 것을 의미한다.

이러한 선행 연구들은 추천 성능을 향상시키기 위해 단순히 layer를 늘리거나 추가 정보를 인풋에 더하는 것 이외에, 어떻게 오토인코더 내부 구조를 모델링하는지도 중요하다는 것을 나타낸다.

3. 제안 기법

상용되고 있는 추천 시스템에는 일반적인 사용자, 상품 평점 행렬 정보 이외에 각 사용자, 또는 상품에 대한 추가적인 정보들이 함께 저장된다. 그렇기에 특정 사용자에게 맞춤화된 추천을 제공하기 위해서는 평점 이외의 정보를 어떻게 효과적으로 사용하는지에 대한 모델링이 중요하다.

본 연구는 이러한 점에 주목하여, 딥러닝의 일종인 SDAE를 사용해서, 평점과 사용자의 추가 정보를 인풋으로 사용하는 추천 모델을 제안한다.

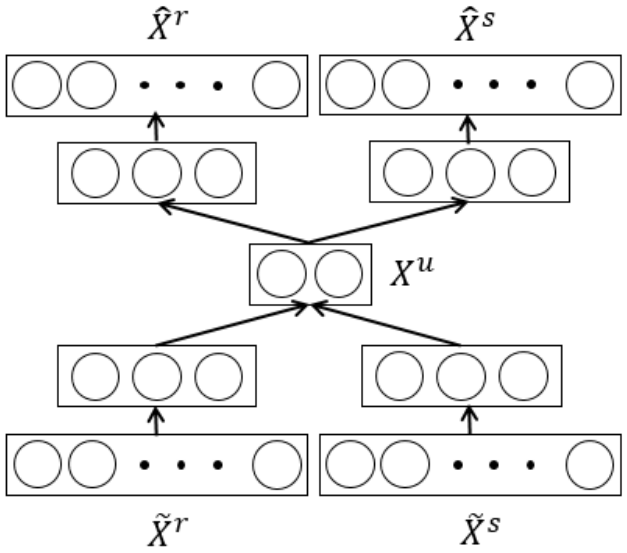

그림 2 사용자의 추가 정보를 고려한 SDAE

추가 정보와 기존의 평점 정보 사이의 내재된 상관성을 고려하기 위해서 공유된 hidden layer 구조를 사용하였다. 즉, hidden representation은 평점과 사용자 특성 정보라는 두 가지 정보를 통합하여 형성된다.

사용자의 내재적 특성을 추출하기 위해서, 각 사용자가 남긴 리뷰를 모은 후 불용어, 구두점을 제거하고 어

간 추출과 같은 전처리를 하였다. 해당 정보는 bag of words 형식으로 사용자의 어휘 사용 특성을 나타내는 추가 정보로 사용된다. 또한, 사용자 고유의 내재적 특성 정보를 나타내는 정보로는 사용자가 부여한 평균 평점과 cool, funny, useful의 득표 수, 그리고 리뷰를 남긴 횟수를 추가 정보로 고려하였다.

인풋 노드의 평점 부분($\tilde{X}^r$)에는 각 유저가 구매한 상품 평점에 대한 벡터 데이터를 넣었다. 유저가 상품을 구매한 경우에는 1-5까지의 자연수 값을 가지며, 구매하지 않은 경우에는 0으로 처리했다. 또한, 각각의 추가 정보는 정규화를 시킨 후 해당 인풋 노드($\tilde{X}^s$)에 넣었다.

$$\text{Loss} = ||\tilde{X}^r - \hat{X}^r||_2^2 + ||\tilde{X}^s - \hat{X}^s||_2^2 + \lambda \sum_l^L (||W_l^r||_2^2 + ||b_l^r||_2^2 + ||W_l^s||_2^2 + ||b_l^s||_2^2) \quad (2)$$

손실 함수는 수식 2와 같이 정의하였다. 손실 함수의 값을 최소화시키기 위해 adam optimizer를 사용하였다. 총 hidden layer의 개수는 3개이며, 각 layer에서의 노드의 수는 50, 25, 50개로 설정하였다. activation function으로 relu를 사용하였으며, 인풋 노드에 0.25의 확률로 dropout을 함으로써 노이즈를 부여했다.

4. 실험 결과 및 분석

본 연구에서는 Yelp 데이터셋을 이용하였다. 해당 데이터셋에는 평점 정보 이외에 다양한 정보들이 포함되어 있지만, 본 연구에서 사용한 것은 평점, 리뷰, 그리고 사용자 특성 정보이다. 유저 개개인의 내재적 특성을 추출하기 위해서 최소한의 리뷰 개수와 리뷰에 있는 단어의 양을 각각 10, 300개으로 제한하였다. 해당 제한 조건을 만족시키는 데이터는 사용자 5510명, 상품 3904개, 평점 개수 8554개이다.

본 연구에서 제안하는 모델의 성능을 판단하는 측정치로 RMSE(Root Mean Square Error)를 사용하였다. 또한, 모델의 성능을 객관적으로 파악하기 위해 비교 모델로 SVR과 MF, 그리고 SDAE 방식을 선정하여 실험을 수행하였다. SVR에서는 두 가지 종류의 추가 정보를 모두 feature로 사용하였다. 5-fold cross validation을 수행한 결과, 본 연구의 실험 결과는 표 1과 같다.

Model	RMSE
SVR	1.166
Matrix Factorization	1.032
SDAE	0.991
Ours (사용자 어휘 사용 특성 정보 추가)	0.974
Ours (사용자 내재적 특성 정보 추가)	0.982

표 1 모델 성능 비교

5. 결론 및 향후 연구

본 연구는 협업 필터링에서 문제가 되어왔던 콜드 스타트 문제를 완화하기 위해서 추가 정보를 사용한 SDAE 모델을 제안하였다. 무조건적으로 추가 정보를 사용하는 것은 성능에 도움이 되지 않을뿐더러 불필요한 연산을 초래하는 문제가 발생한다. 그렇기에 적합한 추가 정보뿐만 아니라, 해당 정보를 활용할 수 있는 모델의 구조를 찾는 것이 중요하다. 그렇기에 본 연구는 공유된 hidden layer 구조를 활용함으로써 이질적인 두 정보를 통합하였다. 해당 모델은 여러 비교 모델과의 실험을 통해 성능의 우수성이 입증되었다.

향후, 두 가지 종류의 각기 다른 추가 정보를 오토인코더를 활용하여 동시에 효과적으로 통합하는 방법에 대해서도 연구할 필요가 있다. 또한, 다양한 데이터셋을 통해 제안하는 모델 성능의 보편성을 확인할 필요가 있다.

참 고 문 헌

[1] Linden, Greg, Brent Smith, and Jeremy York. "Amazon. com recommendations: Item-to-item collaborative filtering." IEEE Internet computing 7.1 (2003).

[2] Resnick, Paul, et al. "GroupLens: an open architecture for collaborative filtering of netnews." Proceedings of the 1994 ACM conference on Computer supported cooperative work. ACM, 1994.

[3] Sarwar, Badrul, et al. "Item-based collaborative filtering recommendation algorithms." Proceedings of the 10th international conference on World Wide Web. ACM, 2001.

[4] Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." Computer 42.8 (2009).

[5] Guo, Guibing, Jie Zhang, and Neil Yorke-Smith. "TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings." Aaai. 2015.

[6] Strub, Florian, and Jeremie Mary. "Collaborative Filtering with Stacked Denoising AutoEncoders and Sparse Inputs." NIPS Workshop on Machine Learning for eCommerce. 2015.

[7] Pan, Yiteng, Fazhi He, and Haiping Yu. "Trust-aware Top-N Recommender Systems with Correlative Denoising Autoencoder." arXiv preprint arXiv:1703.01760 (2017).

[8] Strub, Florian, Jérémie Mary, and Romaric Gaudel. "Hybrid Collaborative Filtering with Autoencoders." arXiv preprint arXiv:1603.00806 (2016).