문자 수준 컨볼루션 뉴럴 네트워크를 사용한 추천 시스템을 위한 행렬 분해 모델

Matrix Factorization Model for Recommendation System Using Character-Level Convolutional Neural Network

저자
(Authors)

손동희, 심규석
Donghee Son, Kyuseok Shim

출처
(Source)

한국정보과학회 학술발표논문집 , 2017.06, 242-244 (3 pages)

발행처
(Publisher)

한국정보과학회
KOREA INFORMATION SCIENCE SOCIETY

URL

http://www.dbpia.co.kr/Article/NODE07207195

APA Style

손동희, 심규석 (2017). 문자 수준 컨볼루션 뉴럴 네트워크를 사용한 추천 시스템을 위한 행렬 분해 모델. 한국정보과학회 학술발표논문집, 242-244.

문자 수준 컨볼루션 뉴럴 네트워크를 사용한 추천 시스템을 위한 행렬 분해 모델

손동희, 심규석
서울대학교 전기·정보공학부
[dhson, shim]@kdd.snu.ac.kr

Matrix Factorization Model for Recommendation System Using Character-Level Convolutional Neural Network

Donghee Son, Kyuseok Shim
Dept. of Electric and Computer Engineering, Seoul National University

요 약

행렬 분해 모델은 추천시스템에서 사용하는 모델 중 한가지로써, 완전하지 않은 사용자-제품 평점 행렬을 추정하는 모델이다. 하지만 최근 사용자와 제품의 수가 많아지면서, 데이터의 희소성 문제가 심각 해졌고, 이로 인해 정확한 추정을 하는데 많은 어려움이 있다. 최근, 데이터의 희소성 문제를 해결하기 위해 제품과 관련된 텍스트 데이터를 이용하는 알고리즘들이 제시되었다. 하지만 기존연구는 텍스트 데이터를 단어수준으로 고려하기 때문에 같은 어근, 접두사, 접미사를 사용하는 유의어들을 비교적 많은 수의 파라미터를 이용하여 표현하는 등의 문제가 발생할 수 있다. 본 연구에서는 단어 수준으로 텍스트를 고려하는 행렬 분해 모델의 문제점을 해결하기 위해 텍스트 데이터를 문자 수준으로 고려하는 행렬 분해 모델을 제안하며, 실제 데이터를 이용하여 모델의 성능을 검증하였다.

1. 서론

추천시스템은 매출을 최대화하기 위하여 사용자에게 구매할 가능성이 높은 제품을 추천해준다. 추천시스템에서 사용하는 추천 방법 중 한가지인 협업 필터링 방법은 사용자와 제품의 관계를 파악하여, 사용자에게 제품을 추천해주는 방법이다. 행렬 분해 모델은 협업 필터링을 모델링해주는 기술 중 하나이다. 하지만 최근 사용자의 수와 제품의 수가 점점 많아지면서, 데이터의 희소성 문제가 심해지고 있고, 이로 인해 행렬 분해 모델을 이용하여 정확한 추천을 하는 것이 어려워지고 있다.

최근 데이터의 희소성 문제를 해결하기 위해 리뷰와 같은 제품과 관련된 텍스트 데이터를 이용하는 행렬 분해 모델 연구[1,2,3]가 많이 진행되어 왔다. 특히, [1]에서는 컨볼루션 뉴럴 네트워크를 이용해 텍스트 데이터를 고려하는 행렬 분해 모델을 제시하였다.

하지만 [1]에서는 텍스트 데이터를 단어수준으로 고려하기 때문에 같은 어근, 접두사, 접미사를 가지고 있는 비슷한 뜻의 단어를 비교적 많은 숫자의 파라미터를 이용하여 표현하고, 오타와 같이 원래 단어에서 형태가 조금 바뀌는 경우에 대해서는 효율적으로 반영하지 못한다.

본 논문에서는 단어수준 고려 모델의 문제를 해결하기 위하여 문자수준으로 텍스트데이터를 고려하는 행렬 분해 모델을 제시한다. 또한 실제 데이터를 이용한 실험을 진행하여 제시한 모델의 성능을 검증하였다.

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(NRF-2012M3C4A7033342). 또한 이 논문은 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2016R1D1A1A02937186).

2. 문제 정의

N명의 사용자와 M개의 제품에 대하여, 평점 행렬 $R \in \mathbb{R}^{N \times M}$을 $(R)_{i,j} = i$번째 사용자가 j번째 제품에 매긴 평점으로 정의한다. 하지만 실제 데이터의 평점 행렬 R은 불완전하기 때문에, 평점 행렬 R을 정확하게 추측하는 것이 행렬 분해 모델의 목표이다.

3. 배경 이론

3.1 행렬 분해법(Matrix Factorization)

행렬 분해법은 협업 필터링에서 사용하는 방법 중 하나로 평점 행렬 R을 추측하기 위한 방법이다. R의 계수가 k라고 가정을 한다면 R을 임의의 잠재 행렬 $U^T \in \mathbb{R}^{N \times k}$와 $V \in \mathbb{R}^{k \times M}$의 곱으로 나타낼 수 있고 R을 가장 잘 표현하는 잠재 행렬 U와 V를 찾아내는 것이 행렬 분해법이다.

잠재 행렬 U와 V의 각 i번째 열, j번째 열을 나타내는 k차원 벡터 u_i와 v_j는 i번째 사용자와 j번째 제품을 나타내는 특징 벡터로 이해할 수 있으며, i번째 사용자가, j번째 제품에 매긴 평점의 추정치 $\hat{r}_{ij}$ 는 u_i와 v_j 두 벡터의 내적을 통하여 계산이 가능하다.

행렬 분해법의 일반적인 방법은 손실 함수를 정의한 후, 경사 하강법 등의 방법을 이용하여 손실 함수를 최소화하는 U, V를 찾는 것이다. [4]에서는 확률 모델링을 이용하여 행렬 분해 알고리즘의 손실 함수를 다음과 같이 정의하였다.

$$L = \frac{1}{2}\sum_i^N \sum_j^M I_{ij}\left(r_{ij} - u_i^T v_j\right)^2 + \frac{\lambda_u}{2}\sum_i^N \|U_i\|_{Fro}^2 + \frac{\lambda_v}{2}\sum_j^M \|V_i\|_{Fro}^2$$

I_{ij} 는 지시함수로 r_{ij}값이 학습데이터에서 관측이 된다면 1 관측되지 않는다면 0을 가지는 함수이다. 그리고 $\|\cdot\|_{Fro}$는 프로베니우스 노름을 의미한다.

3.2 컨볼루션 뉴럴 네트워크

컨볼루션 뉴럴 네트워크는 이미지 분야[5]에서 주로 사용되는 인공 신경망 구조로, 입력 데이터의 각 부분마다 동일한

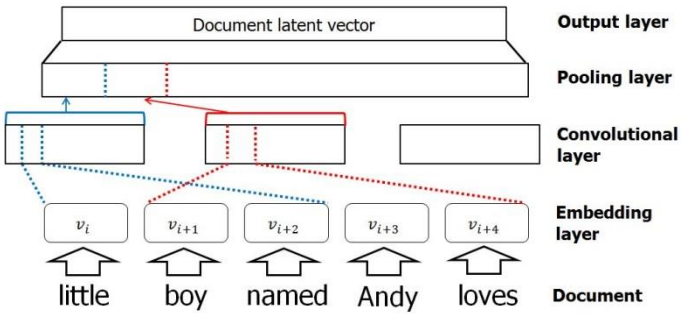

그림 1 ConvMF의 컨볼루션 뉴럴 네트워크 구조

연산을 진행하여 출력을 내보내는 구조이다. 컨볼루션 뉴럴 네트워크 구조는 텍스트 데이터를 사용하는 문장 분류[6]와 같은 일에도 응용 될 수 있다.

컨볼루션 뉴럴 네트워크를 텍스트 데이터에 활용하는 경우 많은 경우에 단어를 기본단위로 이용하게 된다. 하지만 단어를 기본단위로 사용하게 되면 같은 어근, 접두사, 접미사를 사용하면서 비슷한 의미를 가지는 단어를 나타내기 위해서 많은 파라미터를 사용하게 된다. 또한 학습데이터에 나타나지 않거나 적게 나타난 단어들에 대해서는 효율적으로 처리하지 못하는 문제가 발생할 수 있다. 그렇기 때문에 오타가 있거나 소셜 네트워크 텍스트 데이터와 같이 형태가 많이 변화하는 텍스트 데이터의 경우 성능이 저하될 수 있다.

위와 같은 문제점을 해결하기 위해서 텍스트 데이터를 처리할 때 기본 단위를 단어로 하지 않고 문자 단위로 처리하는 컨볼루션 뉴럴 네트워크 구조[7],[8]가 제시 되었다.

4. 기존 연구

본 연구는 [1]에서 제시된 컨볼루션 뉴럴 네트워크와 행렬 분해 모델을 합친 ConvMF 모델을 기반으로 한다.

4.1.ConvMF의 컨볼루션 뉴럴 네트워크 구조

기존의 행렬분해 모델[4]과 ConvMF와의 가장 큰 차이점은 j번째 제품에 대한 잠재 벡터 v_j를 생성할 때 제품과 관련된 텍스트 데이터를 이용한다는 것이다. ConvMF는 컨볼루션 뉴럴 네트워크를 이용하여 텍스트 데이터를 잠재 벡터 v_j로 변환시킨다. ConvMF에서 사용하는 컨볼루션 뉴럴 네트워크의 구조는 그림 1과 같다.

임베딩 레이어(Embedding layer)

임베딩 레이어에서는 텍스트 데이터를 행렬로 바꿔준다. 텍스트 데이터를 단어들의 시퀀스라고 보고, 각 단어를 벡터로 대응 시킨다. 그 후에 모든 벡터들을 연결 한 것을 텍스트 행렬로 사용한다. 단어를 벡터로 대응시키는 과정에서 미리 학습된 Glove[9]와 같은 단어 임베딩 모델을 이용하여 단어를 벡터로 대응시키는 것도 가능하다. 텍스트 행렬 $D \in \mathbb{R}^{p \times l}$는 다음과 같이 나타날 수 있다.

$$D = [\cdots \quad w_{i-1} \quad w_i \quad w_{i+1} \quad \cdots]$$

p는 단어 임베딩 차원(w_i의 차원), l은 텍스트의 길이, w_i는 텍스트에서 i번째 단어를 대응시킨 벡터이다.

컨볼루션 레이어(Convolutional layer)

컨볼루션 레이어에서는 슬라이딩 윈도우를 이용하여 임베딩 레이어에서 구한 행렬로부터 텍스트 데이터의 특징 벡터를 뽑아낸다. j번째 채널에서 뽑아낸 특징 벡터는 다음과 같이 나타낼 수 있다.

$$c_i^j = f(W_c^j * D_{(:,i:(i+ws-1))} + b_c^j)$$

ws는 윈도우 사이즈로 몇 개의 단어를 고려하여 문맥적 특징을 뽑아낼지를 결정해주는 파라미터이다. $W_c^j \in \mathbb{R}^{p \times ws}$ 이며, j번째 채널에서 사용하는 윈도우 행렬이다. $f()$는 비선형 활성

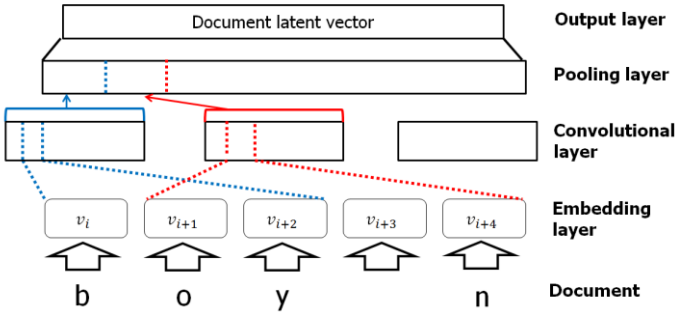

그림 2 제안하는 모델의 컨볼루션 뉴럴 네트워크 구조

함수로 ConvMF에서는 ReLU함수를 사용했다. b_c^j는 바이어스이고, $*$ 는 컨볼루션 연산을 의미한다.

최종적으로 j번째 채널에서 생성 특징 벡터는 c_i^j를 모두 연결한 벡터이다. j번째 채널에서 생성된 특징 벡터는 다음과 같이 나타낼 수 있다.

$$c^j = [c_1^j, c_2^j, \ldots \; c_i^j, \ldots, c_{l-ws+1}^j]$$

풀링 레이어(Pooling layer)

풀링 레이어에서는 각 채널에서 생성된 특징 벡터에서 가장 영향이 큰 요소만 뽑아내는 최대 pooling을 한다. 그리고 모든 채널에 대해서 뽑힌 특징 벡터를 모두 연결한 것을 텍스트 데이터의 특징 벡터로 사용한다. 문서의 특징 벡터는 아래와 같이 나타낼 수 있다.

$$d_f = [\max(c^1), \max(c^2), \ldots, \max(c^j), \ldots, \max(c^{n_c})]$$

n_c는 컨볼루션 뉴럴 네트워크에서 사용한 채널의 개수이다.

출력 레이어(Output layer)

출력 레이어에서는 문서의 특징 벡터를 제품의 특징 벡터로 변환해준다. d_f의 차원이 k 차원이 아니기 때문에 k 차원의 벡터로 변환 시켜준다. 결과적으로 변환된 아이템의 특징 벡터는 아래와 같이 나타난다.

$$s = tanh(W_{f2}\{tanh(W_{f1}d_f + b_{f1})\} + b_{f2})$$

$W_{f1} \in \mathbb{R}^{f \times n_c}, W_{f2} \in \mathbb{R}^{k \times f}$ 는 투사 행렬, $b_{f1} \in \mathbb{R}^f, b_{f2} \in \mathbb{R}^k$ 는 바이어스로 작용된다.

위와 같은 과정을 거치면 텍스트 데이터는 k 차원의 잠재 벡터로 변환된다. 입력으로 j번째 제품에 대한 텍스트 X_j가 들어갔을 때 최종 출력되는 벡터 s_j는 다음과 같이 쓸 수 있다.

$$s_j = CNN(W, X_j)$$

W는 컨볼루션 뉴럴 네트워크에서의 모든 웨이트와 바이어스를 의미한다.

4.2. ConvMF의 최적화 과정

ConvMF의 손실 함수는 다음과 같이 정의된다.

$$L(U,V,W) = \sum_i^N \sum_j^M \frac{I_{ij}}{2}(r_{ij} - u_i^T v_j)^2 + \frac{\lambda_u}{2}\sum_i^N \|u_i\|^2$$

$$+ \frac{\lambda_v}{2}\sum_j^M \|v_j - cnn(W, X_j)\|^2 + \frac{\lambda_W}{2}\sum_k \|w_k\|^2$$

u_i와 v_j의 경우 경사 하강법을 이용하여 위의 손실 함수를 최소화하는 방향으로 업데이트를 해주게 되고, W의 경우 컨볼루션 뉴럴 네트워크의 손실 함수를 최소화하는 방향으로 에러 역전파을 이용해 업데이트를 한다. 컨볼루션 뉴럴 네트워크의 손실 함수는 다음과 같이 정의된다.

$$\varepsilon(W) = \frac{\lambda_v}{2}\sum_j^M \left\| \left(v_j - cnn(W, X_j)\right) \right\|^2 + \frac{\lambda_W}{2}\sum_k \|w_k\|^2$$

5. 제안하는 모델

본 논문에서 제안하는 모델은 컨볼루션 뉴럴 네트워크를 이용해 잠재 벡터 v_j를 만드는 과정에서 텍스트 데이터를 문자 수준으로 고려하는 모델이다. 제안하는 모델에서 사용하는 컨

표 1 실험 데이터와 관련 통계치

Data	사용자 수	제품 수	평점 수	Density
MovieLens	6040	3667	997780	4.5049%
Amazon	426924	26965	583937	0.0051%

볼루션 뉴럴 네트워크의 구조는 그림 2와 같다. 제안하는 모델에서 임베딩 레이어를 제외한 컨볼루션 뉴럴 네트워크 구조와 최적화 방식은 ConvMF와 같은 방식으로 동작한다.

임베딩 레이어(Embedding layer)

임베딩 레이어에서는 텍스트 데이터를 행렬로 바꿔준다. ConvMF의 경우에는 텍스트 데이터를 단어의 시퀀스로 인식하였지만, 제안하는 모델에서는 문자의 시퀀스로 인식한다. 그 후, 각 문자를 벡터로 대응시키고 그 벡터들을 모두 연결시킨 것을 텍스트 행렬로 사용한다. 고려한 문자는 소문자 알파벳, 대문자 알파벳, 숫자와 기타 특수문자로 총 95개 이다. 텍스트 행렬 $D \in \mathbb{R}^{p \times l}$는 다음과 같이 나타날 수 있다.

$$D = [\cdots \quad w_{i-1} \quad w_i \quad w_{i+1} \quad \cdots]$$

p는 문자 임베딩 차원, l은 텍스트의 길이, w_i는 텍스트에서 i번째 문자를 대응시킨 벡터이다.

6. 실험

6.1. 실험 환경

실험에 사용한 머신은 Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz에 GeForce GTX980 GPU 가 장착되어있고, 메인메모리의 용량은 4GB인 머신이다. 코드는 python과 keras[1]를 통해 구현되었다.

6.2. 실험 데이터

실험에 사용한 데이터는 다음과 같다.

MovieLens[2] : 사용자가 영화에 대하여 평점을 매겨놓은 데이터이다. 텍스트 데이터로는 IMDB[3]의 해당 영화 줄거리(plot summary)를 사용했다.

Amazon[4] : 사용자가 instant video에 대하여 평점을 매겨놓은 데이터이다. 텍스트 데이터는 해당 상품에 대한 리뷰를 이용했다. 두 데이터 세트 모두 평점은 1,2,3,4,5점 중 하나이다. 두 데이터에 대한 간략한 수치는 표 1과 같다.

6.3.구현한 알고리즘

ProbMF[4] : 사용자-제품 평점 행렬만을 이용해, 최적화 과정을 통해 평점을 추정하는 알고리즘이다.

ConvMF[1] : 사용자-제품 평점 행렬과 제품의 텍스트 데이터를 컨볼루션 뉴럴 네트워크를 이용하여 고려하여 평점을 추정하는 알고리즘이다.

ConvMF+[1]: ConvMF에서 단어 임베딩 벡터를 미리 학습 되어있는 Glove[9]를 이용한 알고리즘이다.

CharMF : 본 논문에서 제안하는 알고리즘이다.

6.4. 성능 측정 지표 및 세부사항

성능 측정 지표로는 평균 제곱근 오차(RMSE)를 사용했다. 실제 데이터에서 관측되는 값을 r_{ij}, 알고리즘을 이용하여 추정한 값을 $\hat{r}_{ij}$로 두었을 때, RMSE는 아래와 같이 나타난다.

$$\text{RMSE} = \sqrt{\frac{\sum_{i,j}^{N,M}(r_{ij}-\hat{r}_{ij})^2}{\#\ of\ ratings}}$$

전체 데이터의 80%를 학습 데이터로, 10%는 검증 데이터로,

표 2 실험 결과

	데이터			
	MovieLens		Amazon	
모델	RMSE	파라미터 수 ($\times 10^5$)	RMSE	파라미터 수 ($\times 10^5$)
ProbMF	0.8975	-	1.5939	-
ConvMF	0.8529	19.1	1.3426	19.1
ConvMF+	**0.8524**	19.1	1.3554	19.1
CharMF	0.8525	**3.1**	**1.3003**	**1.2**

10%는 시험 데이터로 하였다. 실험은 임의로 분할한 3개의 데이터세트들에 대하여 진행하였고, RMSE값은 3개의 시험 데이터들에 대해 측정한 평균값을 사용했다. 채널의 개수와 λ_u, λ_v 값은 [1]에서의 값과 똑같이 설정하였다.

6.5. 실험 결과

표 2를 통하여 모델과 데이터에 따른 실험 결과를 비교할 수 있다. MovieLens 데이터에 대해서는 ConvMF, ConvMF+, CharMF 3가지 모델 모두 비슷한 성능을 보이고 있다. 하지만 컨볼루션 뉴럴 네트워크에서 학습해야 하는 파라미터의 수가 ConvMF와 ConvMF+에 비해 CharMF가 훨씬 적은 것을 알 수 있다. Amazon 데이터에 대해서는 ConvMF와 ConvMF+보다 CharMF가 성능이 뛰어나고, 컨볼루션 뉴럴 네트워크에서 학습해야 하는 파라미터의 수도 적다는 것을 알 수 있다. 파라미터의 개수가 줄어든 가장 큰 이유는 단어를 단위로 임베딩 할 때보다 문자를 단위로 임베딩하면서 학습 해야하는 임베딩 행렬의 크기가 줄어들었기 때문이다.

7. 결론

본 논문에서는 문자 수준 컨볼루션 뉴럴 네트워크를 이용한 행렬 분해 모델을 제시하였고, 이를 단어 수준 컨볼루션 뉴럴 네트워크를 이용한 행렬 분해 모델과 성능을 비교하였다. 기존모델과 비교하여, 정확도 향상과 파라미터의 개수가 줄어드는 등의 효과를 실험적으로 확인하였다.

8. 참고문헌

[1] D. Kim et al. Convolutional matrix factorization for document context-aware recommendation. In *RecSys* 2016.

[2] S. Purushotham et al. Collaborative Topic Regression with Social Matrix Factorization for Recommendation Systems. In *ICML* 2012.

[3] H. Wang et al. Collaborative deep learning for recommender systems. In *SIGKDD* 2015.

[4] Salakhutdinov, Ruslan, and Andriy Mnih. "Probabilistic Matrix Factorization." In *Nips*. Vol. 1. No. 1. pp. 2-1, 2007.

[5] A. Krizhevsky et al. Imagenet classification with deep convolutional neural networks. In *Nips*. p. 1097-1105, 2012.

[6] Y.Kim. "Convolutional neural networks for sentence classification." In *arXiv* 2014.

[7] X. Zhang et al. Character-level convolutional networks for text classification. In *Nips,* 2015.

[8] Y. Shen et al. A latent semantic model with convolutional-pooling structure for information retrieval. In *CIKM* 2014.

[9] J. Pennington et al. Glove: Global Vectors for Word Representation. In *EMNLP*. Vol. 14. 2014.

[1] https://github.com/fchollet/keras
[2] https://grouplens.org/datasets/movielens/
[3] http://www.imdb.com/
[4] http://jmcauley.ucsd.edu/data/amazon/