데이터 마이닝 최종보고서

데이터 전처리를 통한 Bert4Rec 성능 개선 : 학습 용량 증가의 해결책

120240287 장래영

120240288 홍문기

Abstract

최근 몇 년간 BERT4Rec 와 SASRec 와 같은 순차적 추천 모델은 트랜스포머 아키텍처를 활용하여 사용자 선호도를 예측하는 데 있어 상당한 성능 향상을 보여주었다. 그러나 BERT4Rec 의 높은 계산 복잡도와 큰 모델 크기는 대규모 데이터셋을 효율적으로 처리하는 데 어려움을 초래한다는 한계점이 존재한다. 본 연구에서는 입력 시퀀스 크기를 줄이면서도 예측 정확성을 유지하는 새로운 데이터 전처리 기법을 제안한다. 제안된 방법은 특정 구간 내 항목의 타임스탬프를 평균화하여 입력 데이터를 압축하고 모델의 학습 효율성을 높인다.

MovieLens 1M 데이터셋을 사용하여 BERT4Rec 모델과 전처리된 버전을 비교 평가한 결과, 제안된 방법이 학습 시간을 크게 줄이면서도 추천 성능(Hit Ratio@K 및 NDCG@K 지표)을 향상시킴을 확인하였다. 본 연구는 데이터 전처리를 통해 모델 성능을 향상시킬 수 있다는 결과를 제공하며, 더 확장 가능하고 효율적인 순차적 추천 시스템을 구현하기 위한 실질적인 지침을 제시한다.

1. 서론

최근 몇 년간 추천 시스템 분야는 딥러닝 모델의 적용을 통해 상당한 발전을 이루었다. 그 중에서도 순차적 추천 시스템은 시간에 따른 사용자의 행동 이력을 분석하여 사용자 선호도를 예측하는 능력 덕분에 많은 주목을 받고 있다. 이러한 시스템은 사용자 행동의 순서를 고려함으로써 더 정확하고 개인화된 추천을 제공할 수 있다.

더욱이, 순차적 추천 시스템은 여러 산업에서 다양한 활용도를 보이고 있다. 예를 들어, 전자 상거래에서는 사용자의 구매 이력을 분석하여 다음에 구매할 가능성이 높은 상품을 추천하는 데 사용된다. 이를 통해 매출 증대와 고객 만족도를 동시에 높일 수 있다. 스트리밍 서비스에서는 사용자가 시청한 콘텐츠의 순서를 분석하여 다음에 시청할 만한 영상을 추천하는 데 도움을 준다. 이는 사용자 유지율을 높이는 데 중요한 역할을 한다.

순차적 추천시스템의 대표적인 모델로는 그림(1) SASRec(Self-Attentive Sequential Recommendation)과 그림(2) Bert4Rec(Bidirectional Encoder Representations from Transformers for Sequential Recommendation)이 있다. 이 모델들은 Transformer Architecture 를 활용하여 시퀀스 내 항목 간의 의존성을 포착함으로써 추천의 정확성을 높인다.

먼저 그림(1) SASRec 은 Transformer 의 디코더 구조를 기반으로, 한 단방향 셀프 어텐션 메커니즘을 활용하여 현재 항목과 그 이전 항목들 간의 관계를 집중적으로 분석한다.

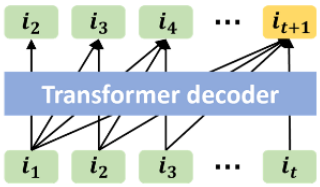

그림 1. SASRec 구조

반면, 그림(2) BERT4Rec 은 트랜스포머의 인코더 구조를 활용한 양방향 셀프 어텐션 메커니즘을 통해 항목과 시퀀스 내 모든 다른 항목들, 즉 과거와 미래의 항목들 간의 상호 의존성을 고려한다.

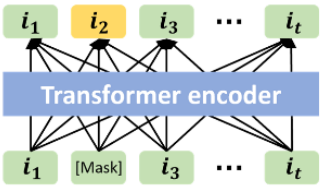

그림 2. BERT4Rec 구조.

2. 문제 정의

본 연구에서 다루는 BERT4Rec 의 한계점은 다음과 같다. 첫번째, BERT4Rec 는 양방향 셀프 어텐션 메커니즘을 사용하여 시퀀스 내 모든 항목 간의 관계를 학습하므로, 모델의 파라미터 수와 계산 비용이 크게 증가한다. 두번째, 대규모 데이터셋을 처리하는 데 있어 BERT4Rec 의 학습 시간과 필요한 컴퓨팅 자원이 증가한다. 이러한 문제는 실시간 추천 시스템에서 사용자 경험을 저하시키고 시스템의 효율성을 떨어뜨릴 수 있기 때문에 BERT4Rec 의 성능을 유지하면서 계산 효율성을 높일 수 있는 방법이 필요하다.

우리는 본 연구에서 데이터 전처리과정을 통해 Bert4Rec 모델이 가지고 있는 한계점을 극복한다.

2.1 제안 방법

본 연구에서는 앞서 언급한 대규모 데이터셋 처리에서의 학습시간 문제를 해결하기 위해 데이터 전처리 접근 방식을 제안한다.

입력 시퀀스를 구간별로 평균화하여 시퀀스 크기를 줄이는 방법을 도입한다. 이를 통해 모델의 입력 데이터 크기를 줄이고, 학습 용량을 단축하여 계산 효율성을 높인다.

Movielens 1M data 를 활용하여 연구를 진행하였기 때문에 시청이력을 바탕으로 구현 방식은 다음과 같다.
우선, 사용자의 시청 이력을 시간 순서대로 구성한 후, 데이터셋의 크기에 따라 데이터셋을 구간으로 나누어 각 구간의 타임스탬프 값을 평균값 및 중앙값으로 한다.

타임스탬프 값을 구간 평균으로 나누었으므로 전체를 input 으로 넣는 것보다 연산량을 줄일 수 있고, 사용자의 시간적 선호도 변화를 효과적으로 반영함과 더불어 모델이 패턴을 효율적으로 학습하도록 구축하였다.

본 연구에서 우리는 양방향 모델인 그림(2) BERT4Rec 의 input 을 1000, 5000, 10000 개의 구간평균 및 구간중앙값으로 나누는 전처리 작업을 진행하였다.
평균값의 경우 평균에서 크게 벗어난 이상치의 영향을 받기에, 성능에 변화가 있을 것이라 판단하였다. 따라서, 평균값으로 전처리 작업을 진행함과 더불어 추가적으로 구간 중앙값에 대한 전처리 작업도 진행하였다.

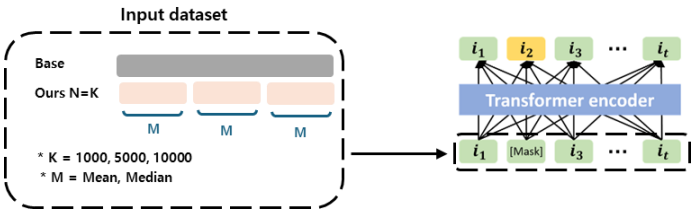

그림 3. 전처리 제안 BERT4Rec 구조

Bert4Rec 모델과 더불어 단방향 셀프 어텐션 메커니즘을 사용하는 그림(4)SASRec 모델에서의 성능도 확인하고자 Bert4Rec 과 동일한 전처리 작업을 진행하였다.

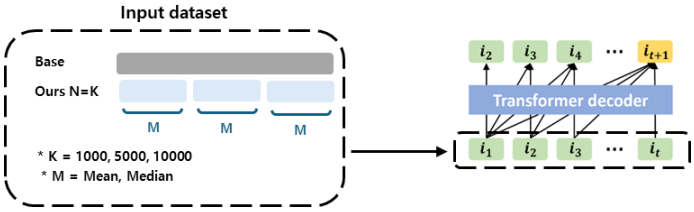

그림 4. 전처리 제안 SASRec 구조

2.2 기대 효과

본 연구의 기대효과는 다음과 같다.
첫째, 타임스탬프 값을 구간 평균으로 나누어 연산량을 줄임으로써 모델의 계산 효율성을 높일 수 있다.
둘째, 사용자의 시간적 선호도 변화를 효과적으로 반영하여 모델이 패턴을 효율적으로 학습할 수 있도록 구축하였다.

이러한 접근 방식은 BERT4Rec 의 구조적 복잡성을 해결하고, 대규모 데이터셋을 효율적으로 처리하여 사용자에게 보다 빠르고 정확한 추천을 제공하는 데 기여할 것으로 기대된다.

3. 실험

3.1 실험 환경

본 연구에서는 MovieLens 1M 데이터를 사용하여 실험을 진행하였다. 학습률(learning rate)은 0.001 로, 배치 크기(batch size)는 128 로 설정하였다. 모델의 레이어 수는 2, 헤드 수는 1 로 설정하였으며, 드롭아웃 레이트(dropout rate)는 0.5, 마스크 확률(mask_prob)은 0.15 로 설정하였다. 학습은 50 에포크(epoch)에 걸쳐 수행되었다.

본 연구에서는 평가 방식으로 Hit Ratio@K(HR@K)와 Normalized Discounted Cumulative Gain@K(NDCG@K)를 사용하였다.

$$HR@K = \frac{1}{|U|} \sum_{u \in U} I(\text{rank}(i_u) \leq K)$$

그림 5. Hit Ratio@K 수식

Hit Ratio 는 추천 시스템의 성능을 평가하기 위한 지표로, 상위 K 개의 추천 목록에서 적어도 하나의 아이템이 실제로 사용자가 선택한 아이템과 일치하는지를 측정한다.
HR@K 는 다음과 같이 정의된다.

여기서 U 는 사용자 집합, i_u 는 사용자 u 가 선택한 아이템, rank(i_u)는 아이템 i_u 의 추천 목록 내 순위, $I(\cdot)$는 지시 함수로 조건이 참일 경우 1, 거짓일 경우 0 을 반환한다.

Normalized Discounted Cumulative Gain@K(NDCG@K)는 추천 목록의 품질을 평가하는 지표로, 아이템의 순위에 따라 가중치를 부여하여 추천 시스템의 순위 품질을 측정한다. NDCG@K 는 다음과 같이 정의된다:

$$DCG@K = \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

$$IDCG@K = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

$$NDCG@K = \frac{DCG@K}{IDCG@K}$$

그림 6. NDCG@K 수식

여기서 rel_i 은 순위 i 의 아이템에 대한 관련성 점수, REL 은 이상적인 순위 목록에서의 관련성 집합이다. DCG@K 는 실제 추천 목록에서의 누적 이득, IDCG@K 는 이상적인 추천 목록에서의 누적 이득이다. NDCG@K 는 DCG@K 를 IDCG@K 로 정규화한 값으로, 0 에서 1 사이의 값을 가지며,

1 에 가까울수록 더 높은 추천 품질을 의미한다.

3.2 Bart4Rec 과 SASRec 의 성능 비교

Model	Metric	Base	Ours@1000	Ours@5000	Ours@10000
Bert4Rec(Mean)	HR@10	0.543	0.577	0.597	0.581
	NDCG@10	0.309	0.327	0.344	0.333

표 1. Bert4Rec 모델 데이터 전처리(Mean)반영 성능

실험결과, 제안한 방법이 기존 모델(Base) 대비 성능이 향상된 것을 확인할 수 있었다. 구체적으로, HR@10 의 경우 Base 모델의 성능은 0.543 이었으나, 제안한 방법을 적용한 결과 Ours@1000 에서 0.577, Ours@5000 에서 0.597, Ours@10000 에서 0.581 로 향상되었다.

마찬가지로, NDCG@10 의 경우 Base 모델의 성능은 0.309 였으나, 제안한 방법을 적용한 결과 Ours@1000 에서 0.327, Ours@5000 에서 0.344, Ours@10000 에서 0.333 으로 개선되었다. 이러한 결과는 제안한 방법이 BERT4Rec 모델의 성능을 유지하거나 향상시킬 수 있음을 보여준다.

Model	Metric	Base	Ours@1000	Ours@5000	Ours@10000
Bert4Rec (Median)	HR@10	0.543	0.562	0.577	0.575
	NDCG@10	0.309	0.320	0.327	0.325

표 2. Bert4Rec 모델 데이터 전처리(Median)반영 성능

타임스탬프 값을 중앙값으로 계산했을 때, 성능이 향상되었으나 평균값으로 계산했을 때 보다는 성능이 다소 낮았다. 이는 MovieLens 데이터셋의 타임스탬프 값에 이상치(outliers)가 거의 없음을 시사한다.
평균값을 사용하면 데이터의 전반적인 분포를 더 잘 반영할 수 있으며, 이상치의 영향이 적기 때문에 모델이 더 일관되고 안정적인 패턴을 학습할 수 있다.

따라서 MovieLens 데이터셋에서는 평균값을 사용하는 것이 시간적 선호도 변화를 보다 효과적으로 반영하고, 추천 성능을 더욱 향상시키는 데 기여한 것으로 판단된다.

Model	Metric	Base	Ours@1000	Ours@5000	Ours@10000
SASRec (Mean)	HR@10	0.478	0.477	0.479	0.480
	NDCG@10	0.264	0.264	0.264	0.266

표 3. SASRec 모델 데이터 전처리(Mean)반영 성능

SASRec 모델에서 타임스탬프 값을 평균값으로 사용했을 때, 실험 결과 HR@10 및 NDCG@10 지표에서 성능이 크게 향상되지 않았다. 이는 SASRec 모델의 특성과 관련하여 설명할 수 있다. SASRec 모델은 Sequential Recommendation 을 위해 Self-Attention 기반의 구조를 사용하며, 주어진 시퀀스 데이터에서 아이템 간의 시간적 흐름을 중요하게 고려한다.

Model	Metric	Base	Ours@1000	Ours@5000	Ours@10000
Bert4Rec (Median)	HR@10	0.478	0.478	0.473	0.478
	NDCG@10	0.264	0.263	0.262	0.263

표 4. SASRec 모델 데이터 전처리(Median)반영 성능

SASRec 모델에서 타임스탬프 값을 중앙값으로 사용했을 때, 실험 결과 HR@10 및 NDCG@10 지표에서 평균값을 사용했을 때와 마찬가지로 성능이 크게 향상되지 않았다. 이는 앞서 설명한 SASRec 모델의 특성과 관련하여 설명할 수 있다.

1. 시간적 흐름의 중요성 :
SASRec 모델은 주로 사용자의 과거 행동 패턴에 기반하여 추천을 수행한다. 타임스탬프를 평균값으로 계산할 경우,

이전 행동과 최근 행동의 차이가 상쇄되거나 모호해질 수 있다. 특히, 평균값은 시간적 선호도 변화를 세밀하게 반영하기 어려울 수 있다.

2. 모델 설계의 특성:
SASRec 모델은 타임스탬프 정보를 적절히 활용하여 사용자의 시간적 선호도를 학습하도록 설계되었다. 그러나 평균값을 사용하면 이러한 시간적 정보가 일관되게 반영되지 않을 수 있어, 모델의 성능 향상에 제한적인 영향을 미칠 수 있다.

따라서 SASRec 모델에서는 시간적 흐름을 더 정확하게 반영하고, 사용자의 시간적 선호도를 효과적으로 학습하기 위해 타임스탬프 값을 평균값보다 다른 방식으로 처리하는 것이 추천 시스템의 성능을 개선하는 데 도움이 될 수 있다.

4. 후속연구

데이터 전처리 과정으로 성능을 효과적으로 올렸으나, 모델링 과정에서 Convolution layer 를 추가하여 추가적인 성능 향상을 달성할 수 있도록 연구를 진행하고 있다.

Convolution layer 는 시퀀스 데이터에서 특정 시간 간격 내의 패턴을 감지하고 로컬 패턴을 학습하는데 뛰어난 특징을 가지고 있으므로, 글로벌 문맥 정보를 학습하는 셀프 어텐션 매커니즘을 보완하는데 효율적이다.
또한, 계산 효율성이 높아, 초기 단계에서 시퀀스 데이터의 차원을 줄이는데 효과적이다.

이러한 두가지 이유를 바탕으로 Convolution layer 를 추가한 모델링 연구를 진행하고 있으며, 이를 바탕으로 더 좋은 성능의 연구결과를 도출하여 해외 학술지에 논문을 투고하고자 한다.

5. 결론

본 연구에서는 BERT4Rec 의 성능을 향상시키기 위해 데이터 전처리 과정을 설계하고 평가하였다.
구체적으로는 Movielens 1M dataset 를 사용하여 1000, 5000, 10000 개로 타임스템프를 평균값 및 중앙값으로 압축하였으며, 순차적 추천시스템에서 대표적인 평가방식인 HR@10 과 NDCG@10 지표를 사용하여 성능을 평가하였다.

실험결과, 데이터 전처리 모델이 기존 모델에 비해 높은 HR@10 과 NDCG@10 결과를 달성하였다. 이는 데이터 전처리를 사용한 모델이 학습 시간을 크게 줄임과 더불어 추천 성능을 효과적으로 향상시켰음을 의미한다.

6. 참고문헌

(1) Zhankui He, Handong Zhao, Zhe Lin, ZhaowenWang, Ajinkya Kale, and Julian McAuley. Locallyconstrained self-attentive sequential recommendation.CIKM, 2021.

(2) Wang-Cheng Kang and Julian J. McAuley.Self-Attentive Sequential Recommendation. ICDM,2018.

(3) Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu,Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, andJi-Rong Wen. S3-Rec: Self-Supervised Learningfor Sequential Recommendation with MutualInformation Maximization. CIKM, 2020.

(4) Chen Ma, Peng Kang, and Xue Liu. Hierarchicalgating networks for sequential recommendation.KDD, 2019.

(5) Jiaxi Tang and Ke Wang. Personalized top-nsequential recommendation via convolutional sequenceembedding. WSDM, 2018.

(6) Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis,Joemon M Jose, and Xiangnan He. A simpleconvolutional generative network for next itemrecommendation. WSDM, 2019.

(7) Tim Donkers, Benedikt Loepp, and Jürgen Ziegler,Sequential User-based Recurrent Neural NetworkRecommendations. RecSys, 2017.

(8) Balázs Hidasi and Alexandros Karatzoglou.Recurrent Neural Networks with Top-k Gains forSession-based Recommendations., CIKM, 2018.

(9) Chao-Yuan Wu, Amr Ahmed, Alex Beutel,Alexander J. Smola, and How Jing. RecurrentRecommender Networks. WSDM, 2017.

(10) Shuqing Bian, Wayne Xin Zhao, Kun Zhou, JingCai, Yancheng He, Cunxiang Yin, and Ji-Rong Wen.Contrastive Curriculum Learning for SequentialUser Behavior Modeling via Data Augmentation.CIKM, 2021.