콜드 스타트 문제 개선을 위한 GAN 기반 Sparse Matrix 증강 모델

GAN-based Sparse Matrix Augmentation for Alleviating Cold-start Problem

요 약

추천 시스템에서 새로운 사용자나 제품, 서비스가 많아지거나, 데이터가 적은 상황에서 콜드 스타트 문제(Cold-start Problem)가 발생할 수 있다. 이를 해결하기 위한 방법 중 하나로 부족한 데이터를 보완하기 위한 데이터 증강 기법이 있다. 본 연구에서는 데이터 증강을 위해 Sparse Matrix 데이터에 적합한 GAN 기반 데이터 증강 모델을 제안한다. 실제 데이터와 유사한 데이터를 생성할 수 있도록 학습하여, 데이터가 부족한 상황에서도 추천 시스템의 정확도를 도출할 수 있도록 한다. 제안하는 모델의 효과를 검증하기 위하여 MovieLens 100K 데이터셋을 활용하여 비교를 통해 평가한다. 제안하는 GAN 기반 모델은 실제 데이터와 유사도 분석 결과 기존 GAN보다 9.1% 더 높은 유사성을 보였고, 행렬 내 행 간 유사도 분석 결과 71% 이상 행 간 유사도가 감소하여 GAN의 mode collapse 문제를 개선하였다. 제안한 추천 시스템 결과는 Matrix Factorization에서 약 24.6%, Light Graph Convolutional Network에서 약 33.2% 성능 개선을 보였다. 본 결과를 통해 제안하는 모델이 다양성 높은 데이터를 생성하며, 추천 시스템의 성능을 향상시킴으로써 Sparse Matrix 형태의 데이터 증강에 적합하다는 것을 확인하였다.

1. 서 론

추천 시스템은 특정 사용자의 이전 구매 이력, 평가, 또는 행동 패턴을 분석하여 해당 사용자에게 적합한 제품, 서비스를 예측하여 제공해 주는 시스템을 의미한다[1]. 주로 콘텐츠 기반 필터링(Content-based Filtering)과 협업 필터링(Collaborative Filtering)으로 구분된다[1].

추천 시스템에서는 사용자에게 적합한 아이템을 예측하여 제공하기 위하여 아이템과 사용자의 상호작용에 의존한다. 하지만, 이러한 상호작용을 구성하기 위한 데이터 수가 적을 경우, 추천 시스템의 성능은 급격히 떨어지게 된다. 반면, 데이터가 많을 경우 사용자와 아이템 간의 많은 상호작용을 통해 추천 시스템은 좋은 성능을 보일 수 있다. 이처럼 데이터 수가 적을 경우 적절한 추천 결과를 도출하지 못하는 상황을 콜드 스타트 문제(Cold-start Problem)라고 한다[2].

콜드 스타트 문제를 해결하기 위해 여러 방법이 소개되고 있다[2]. 그중 하나로 데이터 증강 기법이 있다. 이 방법은 실제 데이터와 유사한 가상 데이터를 생성함으로써, 데이터 부족으로 인한 콜드 스타트 문제를 해결할 수 있다. 데이터 증강을 위한 여러 기법이 존재하지만, 잠재성이 높은 GAN(Generative Adversarial Network) 기법을 추천 시스템 분야에 적용하여 더욱 적절한 데이터 증강을 도출할 수 있다[3].

추천 시스템을 구성하는 데이터는 행렬 형태로 표현할 수 있다. 사용자와 아이템 간의 상호관계의 예로서, 평점 데이터를 행렬로 표현하면, 그 행렬은 빈 공간이 많은 Sparse 특징을 가지게 된다. 이는 다수의 사용자와 아이템 간에 서로 상호관계를 가지는 데이터 수가 적기 때문이다.

본 연구에서는 Sparse Matrix 데이터 환경에서 GAN을 통해 데이터 증강을 진행할 때 발생하는 문제를 개선하고 실제 데이터와 유사한 가상 데이터를 생성하기 위한 GAN 구조를 제시하여 추천 시스템의 성능을 개선하고자 한다. GAN의 Generator를 2개로 설계하여 각각 학습한 뒤, 이를 결합하여 실제 데이터의 특징을 비슷하게 나타내는 데이터를 생성하고, 공개된 데이터셋을 이용하여 추천 시스템에서 효과를 검증하고 가능성을 고찰한다.

2. 문제 정의

사용자가 어떤 아이템을 평가했는지를 사용자-아이템 행렬 $R : m \times n$로 표현할 수 있다. m은 사용자의 수, n은 아이템의 수이며, $R_{i,j} = x$는 사용자 i가 아이템 j에 x만큼의 평점을 내린 것을 의미한다. 즉, 행렬 내 값 $R_{i,j}$는 사용자와 아이템 특성 2개에 의해 영향을 받으며, 이는 사용자 별로 다른 평점을 부여하는 특성과 아이템의 만족도라고 할 수 있다. 기존의 GAN은 mode collapse 현상이 나타나는 문제점이 존재한다[4]. 이는 모델이 학습 데이터의 특정한 형태의 특징에만 집중하여, 다양한 형태의 데이터를 생성하지 못하는 현상이다[4]. 본 연구에서는 새로운 GAN 모델의 설계를 통해 사용자와 아이템의 특성을 반영하여 다양한 데이터를 생성할 수 있는 모델을 제안한다.

3. GAN 모델

GAN은 생성자(G : Generator)와 판별자(D : Discriminator) 두 신경망으로 이루어져 있다. 생성자는 노이즈로부터 실제 데이터와 유사한 가상 데이터를 생성하며, 판별자는 생성자가 생성한 가상 데이터와 실제 데이터를 구분하려고 하는 모델이다. 두 신경망은 서로 경쟁하면서 점차 더 나은 성능을 발휘하도록 학습된다[5]. 그 결과 실제 데이터와 구분할 수 없을 정도로 비슷한 가상 데이터를 만들어 낼 수 있게 된다. GAN의 학습을 위한 목적 함수는 아래와 같다[5].

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(Z)))]$$

위 식에서 x는 실제 데이터, z는 임의로 생성되는 noise, $G(i)$는 input(i)을 입력으로 받은 생성자가 생성한 데이터, $D(i)$는 i를 입력으로 받은 판별자가 실제 데이터라고 판정할 확률값을 의미한다.

본 연구에서는 GAN을 이용하여 가상 데이터를 생성하는 방안을 제안한다. 추천 시스템의 특성상 평점 행렬은 빈값이 많은 Sparse한 특징을 가지며, 이는 GAN이 주로 활용되는 컴퓨터 비전 분야와는 다른 특성이다[3,6]. Sparse한 특성을 반영하기 위해 2개의 생성자와 1개의 판별자로 구성된 GAN을 2개 사용한다. 제안하는 모델의 구조는 그림 1과 같다.

3.1 모델 구조

모델은 2가지의 GAN으로 구성되어 있다. 하나는 평점 행렬(r: Rating Matrix)을, 나머지 하나는 바이너리 행렬(b: Binary Matrix)을 각각 학습한다. 바이너리 행렬은 사용자가 아이템에 평점을 부여하였을 경우 1, 아닐 경우 0을 부여한 행렬이다.

기존의 GAN 구조가 1개의 생성자와 1개의 판별자를 가지는 것과 달리, 본 연구의 GAN 모델은 2개의 생성자와 1개의 판별자를 사용한다. 하나의 생성자는 사용자 시점의 데이터를 생성하고, 다른 생성자는 아이템 시점의 데이터를 생성한다. 이렇게 2개의 생성자를 통해 생성된 두 행렬은 그림 1에서 ⊙로 표현된 요소별 곱셈(element-wise multiplication)을 통해 사용자와 아이템의 특성을 반영한 하나의 결과 행렬을 생성하게 된다.

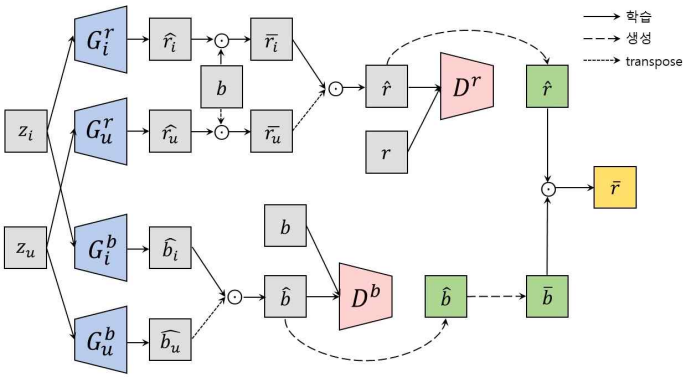

그림 1. 제안하는 모델의 구조

3.2 모델 학습

본 연구의 학습 과정을 크게 2가지 단계로 구분할 수 있다. 첫 번째는 평점 행렬의 학습이다. 위 그림에서 z_i와 z_u는 생성자의 입력 값이 되는 무작위적 노이즈이다. 여기서 두 노이즈 행렬의 크기는 전치 관계(Transpose)이다. r은 실제 평점 행렬을, b은 실제 바이너리 행렬을 의미한다. 노이즈 z_i를 입력받은 G_i^r는 가상 평점 행렬 $\hat{r_i}=G_i^r(z_i)$를, 노이즈 z_u를 입력받은 G_u^r은 가상 평점 행렬 $\hat{r_u}=G_u^r(z_u)$를 생성한다. 각각의 가상 평점 행렬에 바이너리 행렬을 요소별 곱셈한 두 값 $\bar{r_i}=\hat{r_i}\odot b$와 $\bar{r_u}=\hat{r_u}\odot b^T$를 요소별 곱셈을 통해 최종적인 가상 평점 행렬 $\hat{r}=\bar{r_i}\odot\bar{r_u}$을 도출한다. 판별자인 D^r에서는 실제 평점 행렬(r)과 가상 평점 행렬($\hat{r}$)을 구별한다.

다음 단계로는 바이너리 행렬의 학습이다. 평점 행렬과 마찬가지로 노이즈 z_i를 입력받은 G_i^b는 가상 평점 행렬 $\hat{b_i}=G_i^b(z_i)$를, 노이즈 z_u를 입력받은 G_u^b은 가상 평점 행렬 $\hat{b_u}=G_u^b(z_u)$를 생성한다. 이를 요소별 곱셈을 통해 최종적인 가상 바이너리 행렬 $\hat{b}=\bar{b_i}\odot\bar{b_u}$을 도출한 뒤, 판별자 D^b를 통해 실제 바이너리 행렬(b)과 가상 바이너리 행렬($\hat{b}$)을 구별하며 학습을 진행한다. 모델의 학습에는 Adam Optimizer를 이용한 gradient descent를 통해 학습한다. 한 번의 epoch에 2개의 생성자를 번갈아 가며 학습하고 그 후 판별자를 학습하는 방식을 이용한다.

바이너리 평점 행렬의 GAN 학습이 완료되면 실제 데이터와 유사한 Sparsity 구현을 위해 $\hat{b}$에서 상위 k개 값을 1로 바꾸고 나머지를 0으로 바꾼 $\bar{b}$를 도출한다. 이를 $\hat{r}$과 요소별 곱셈을 통해 최종적인 가상 평점 행렬 $\bar{r}$를 완성한다[6].

4. 실험 및 결과 분석

본 연구에서 제안하는 모델의 효과를 검증하기 위해 수행한 실험과 그 결과를 분석한다.

4.1 데이터셋 및 실험 환경

본 연구에서는 MovieLens 100K[7] 데이터셋을 사용하여 실험을 진행한다. MovieLens 데이터셋은 사용자가 영화에 대해 부여한 평점 정보와 영화 정보를 담고 있다. 표 1은 실험에 사용된 데이터셋의 기본 정보이다.

표 1. 실험 데이터 기본 정보

사용자 수	영화 수	평점 수	Sparsity
6,040	3,883	1,000,209	0.9574

MovieLens 데이터셋을 사용자와 영화를 각각 행과 열로 구성한 행렬 $R:6040\times3883$으로 전처리한다. 이렇게 구성된 행렬은 0.9574의 Sparsity를 가진 빈 공간이 많이 있는 특징을 가지고 있음을 확인할 수 있다.

GAN의 학습과정에서는 50번의 epoch 과정으로 학습을 진행하며, Learning rate는 판별자는 0.00006, 생성자는 0.00004이다. 생성자는 5개의 Layer(128, 512, 1024, 2048, output)이며, 판별자는 5개의 Layer(2048, 1024, 512, 128, 1)로 구성한다. 각각의 Layer 사이에 Batch Normalization과 Dropout(0.3)을 진행한다.

4.2 평가 환경

본 연구에서는 아래의 평가 환경 모델 내의 추천 시스템 구성을 통해 기존 데이터셋과 생성된 데이터셋의 결과 성능을 비교한다.

Matrix-Factorization (MF)[8]: 사용자와 아이템 간의 상호작용을 행렬로 모델링하여 추천을 수행하는 추천 시스템 기법이다. 사용자-아이템 평점 행렬을 낮은 차원의 사용자 잠재 요인 행렬과 아이템 잠재 요인 행렬의 내적으로 근사화하여, 이를 추천 결과로 활용한다.

Light Graph Convolutional Network (LightGCN)[9]: 사용자와 아이템 간의 그래프 구성을 통해 그래프 합성곱 네트워크를 이용해 추천을 수행하는 추천 시스템 기법이다. 사용자와 아이템을 노드로, 상호작용을 엣지로 나타내는 그래프를 통해 추천 결과를 도출한다.

4.3 평가 척도

생성된 데이터를 통한 콜드 스타트 문제 개선과 추천 시스템 성능 향상을 평가하기 위해서 MF 모델 성능 척도로 RMSE(Root Mean Square Error)를, LightGCN 모델 성능 척도로 NDCG(Normalized Discounted Cumulative Gain)과 HIT Rate를 사용한다. NDCG는 추천된 항목의 순위를 고려하여 성능을 측정하는 지표이고, HIT Rate는 K값 이상의 항목이 추천

결과에 포함되어 있는 비율을 의미한다. 두 척도를 통해, 각각 추천의 품질과, 정확한 추천의 빈도를 평가할 수 있다. RMSE, NDCG, 그리고 HIT는 다음 수식을 통해 계산된다.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}, \quad NDCG_{@K} = \frac{DCG_{@K}}{IDCG_{@K}}$$

$$HIT = \frac{Number\,of\,Hits}{Number\,of\,Users}$$

4.4 실험 결과

표 2. 기존 데이터와 생성된 데이터에 따른 성능 비교

	평가지표			Data 수
	MF	LightGCN		
	RMSE	NDCG@10	HIT@10	
Movie Lens	1.0782	0.3005	0.0704	1,000,209
Movie Lens +GAN	0.8130	0.4003	0.3230	2,078,904

실험 결과는 표 2와 같다. 기존 데이터는 1,000,209개이며, 제안하는 모델로 데이터 증강 결과 2,078,904개로, 약 108% 증강하였다. 평가 결과, 기존의 MovieLens 데이터를 통해 추천 시스템을 만든 것보다, 본 연구에서 제안한 GAN 모델을 통해 생성된 데이터와 함께 추천 시스템을 만든 것이 더 나은 성능을 보여준다. RMSE는 약 24.6%, NDCG@10은 약 33.2% 개선된 결과를 보인다. 특히 HIT@10은 3배 이상 크게 증가함을 확인하였다. 이는 실제 데이터와 유사한 데이터를 생성함으로써 콜드 스타트 문제를 해결할 수 있는 가능성을 내포한다.

표 3. GAN 별 실제 데이터와 생성 데이터 유사도 비교

	실제 데이터	기존 GAN	제안하는 GAN
실제 데이터와 유사도(MSE)	-	0.7593	0.6902
행 간 코사인 유사도 평균	0.1315	0.8164	0.2319

기존 GAN과 본 연구에서 제안하는 GAN의 결과를 비교하면 표 3과 같다. 행렬 데이터 간의 MSE 비교 결과, 제안하는 GAN의 생성 데이터가 기존 GAN 데이터보다 더 유사함을 확인할 수 있다. 또한, 행렬 내 행 간 코사인 유사도 평균 결과, 기존 GAN 데이터는 mode collapse 현상으로 인해 다양성 낮은 데이터들이 생성되는 반면, 제안하는 GAN은 이러한 문제점을 개선함을 확인하였다.

5. 결론 및 향후 과제

본 연구에서는 콜드 스타트 문제 개선을 위해 새로운 GAN 모델 기반의 데이터 증강 기법을 제안하였다. MovieLens 데이터셋을 통해 실험한 결과, 다양성이 높은 데이터들이 효과적으로 생성되며 추천 시스템의 성능이 개선됨을 확인하였다. 향후 실제 사용자가 판단하기에도 더 좋은 추천 성능을 보이는지 설문조사를 진행해 보고, 주요 추천 도메인에서의 성능을 확인하고자 한다.

참고문헌

[1] Ricci, Francesco, Lior Rokach, and Bracha Shapira. "Introduction to recommender systems handbook." Recommender systems handbook. Boston, MA: springer US, 2010. 1-35.

[2] Schein, Andrew I., et al. "Methods and metrics for cold-start recommendations." Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. 2002.

[3] Prosvetov A. V. "GAN for Recommendation System" Big Data and AI Conference. 2019

[4] Zhang, Zhaoyu, Mengyan Li, and Jun Yu. "On the convergence and mode collapse of GAN." SIGGRAPH Asia 2018 Technical Briefs. 2018. 1-4.

[5] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.

[6] Chae, Dong-Kyu, et al. "AR-CF: augmenting virtual users and items in collaborative filtering for addressing cold-start problems." Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020.

[7] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactivee Intelligent Systems.

[8] Ma, Hao, et al. "Sorec: social recommendation using probabilistic matrix factorization." Proceedings of the 17th ACM conference on Information and knowledge management. 2008.

[9] He, Xiangnan, et al. "Lightgcn: Simplifying and powering graph convolution network for recommendation." Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. 2020.