

웹 서비스 QoS 랭킹을 위한 그래프 컨볼루션 기반 협력 필터링

최정환¹, 류덕산²
¹연세대학교, ²전북대학교

jeongwhan.choi@yonsei.ac.kr, duksan.ryu@jbnu.ac.kr

Graph Convolution-based Collaborative Filtering for Web Service QoS Ranking

Jeongwhan Choi¹, Duksan Ryu²
¹Yonsei University, ²Jeonbuk National University

요약

Quality of Service (QoS)에 기반한 웹 서비스 추천은 사용자의 개인화 추천을 위해 중요하다. 웹 서비스 추천을 위한 Collaborative Filtering 연구가 활발하게 이루어지고 있다. 그래프 컨볼루션 기반 협업 필터링은 전파된 이웃 정보를 활용하여 사용자와 서비스 정보를 업데이트 할 수 있으며, 웹 서비스의 QoS 랭킹에 적용된 사례가 없다. 본 연구의 목표는 그래프 구조의 고차 연결성을 활용하는 것이 QoS 추천 성능 향상에 효과적인지 확인하는 것이다. 이를 위해 QoS 데이터를 사용자-서비스 그래프로 구성하여 그래프 컨볼루션 기반 협력 필터링을 이용한다. 실험 결과 QoS 데이터셋에서 그래프 컨볼루션 기반 협력 필터링을 이용하는 것이 협력 필터링의 성능을 향상시키며 warm-start와 cold-start 환경에서 모두 더 높은 추천 성능을 보인다. 이처럼 그래프 구조를 사용하는 것이 QoS 랭킹 기반 추천을 향상시키므로 보다 효과적인 웹서비스의 개인화 추천을 기대할 수 있다.

1 서론

웹 서비스는 프로그래밍 방식의 메소드 호출을 통해 기계간 상호 작용을 지원하도록 설계된 재사용 가능한 웹 구성 요소이다 [1]. ProgrammableWeb은 공개 웹 서비스가 20,525개 있다고 보고하였고 클라우드 컴퓨팅의 발전으로 웹 서비스의 가용성은 가속화되고 있다 [2]. 웹 서비스 중 많은 수가 사용자에게 유사한 기능을 제공한다. QoS (Quality-of-Service)는 웹 서비스의 품질 속성을 나타내며 웹 서비스들을 구별하는 중요한 기준으로 인식된다. QoS 예측을 위해서 서비스 추천 [3], 선택 [4], 발견 [5] 등이 제안되었다.

QoS에 기반한 웹 서비스 선택 및 추천 방법은 매우 인기가 있다. 개인화된 QoS 인식 웹 서비스 추천은 서비스 사용자가 항상 비슷한 QoS 선호도 또는 사용 경험을 가진 사람들로 부터 최상의 추천을 해준다. 지금까지 협업 필터링 (Collaborative Filtering) 기반의 방법은 QoS 인식 웹 서비스 추천에 널리 사용되었다. 협업 필터링 방법의 목표는 주어진 사용자-서비스 행렬에서 누락된 QoS 값 또는 등급에 대한 더 나은 근사치를 얻는 것이다 [6-8]. 하지만 높은 예측 정확도가 만족스러운 추천 결과로 이어지지 않는다. 서비스 사용자는 QoS 값 보다 사용 가능한 웹 서비스의 등급 (rating)에 관심을 가질 가능성이 더 높다.

예를 들어 사용자 u_i 이 두 개의 웹 서비스 s_i 및 s_j 를 호출 한 후 응답시간에 대해 관찰된 QoS 값은 각각 0.4, 0.5 이다.

표 1: 협업 필터링 알고리즘들에 대한 비교

협업 필터링 알고리즘	이웃 정보	고차 연결 정보
MF 기반	△	X
그래프 컨볼루션 기반	O	O

모델 M_1, M_2 가 예측한 s_i 및 s_j 의 QoS 등급이 각각 0.3, 0.6, 그리고 0.5, 0.45라고 가정하면 M_2 가 M_1 보다 우수하다. 하지만 M_2 에 따라 u_i 과 유사한 사용자에게 s_j 를 추천하는 것은 부적절하다. 즉 웹 서비스 추천의 품질 순위가 사용자의 선택에 영향을 미친다. 따라서 QoS 등급을 기반으로 하는 개인화된 웹 서비스 추천은 잠재적으로 가치가 있다 [9].

한편 추천 도메인에서의 협업 필터링은 Graph Convolutional Network (GCN)의 적용으로 추천 성능의 진일보한 도약을 이루었다 [10-12]. 표 1와 같이 그래프 컨볼루션 기반 협업 필터링은 그래프의 고차 연결성을 활용할 수 있다. 하지만 Matrix Factorization (MF) 기반 협업 필터링은 이웃정보를 명시적으로 사용하지 않기 때문에 고차 연결을 고려하지 못한다. 우리가 아는 한 그래프의 고차 연결성을 활용하는 협업 필터링을 사용하여 랭킹 기반 웹 서비스 추천을 하는 연구는 없다. 따라서 본 논문에서는 QoS 등급 측면에서 개인화된 웹 서비스 추천을 실현하기 위해 그래프 컨볼루션 기반 협업 필터링 모델들의 적용 가능성을 제시한다. 협업 필터링은 유사한 사용자가 웹 서비스에 대해 유사한 선호도를 보일 것이라고

가정한다. 그래프 컨볼루션 기반 협업 필터링을 통해 Collaborative 신호를 추출하는 자연스러운 방법을 적용하여 웹 서비스 추천을 적용한다. 그래프 컨볼루션 기반 협업 필터링은 고차 연결성을 활용할 수 없는 협업 필터링 모델들 보다 더 뛰어난 추천 성능을 보이는 것을 실험을 통해 보여주고 희소한 환경에서도 추천 성능이 뛰어난 것을 보인다. 본 논문을 통해 그래프 컨볼루션 기반 협업 필터링 방법이 QoS 기반 랭킹 태스크에서 유망한 연구 방향임을 제시한다.

본 연구가 기여하는 바는 아래와 같다:

- Web QoS 추천을 위한 그래프 컨볼루션 기반 협업 필터링 프레임워크를 설계하고 그래프 구조의 고차 연결성을 활용하는 것이 효과적임을 보인다.
- 벤치마크 데이터셋에서 warm-start 환경 뿐만 아니라 두 종류의 cold-start 환경에서의 실험을 보이며 그래프 컨볼루션 기반 협업 필터링의 필요성을 보인다.

2 배경지식 및 관련연구

2.1 QoS 웹 서비스

응답 시간 및 처리량과 같은 QoS 속성은 사용자에 따라 다른 값을 가진다 [13]. 예를 들어 영상을 제공하는 웹 서비스와 같은 특정 서비스 기반 애플리케이션을 고려할 때, 사용자 종속 QoS 데이터는 주로 사용자의 호출 환경에 의해 결정된다 [14]. 따라서 두 사용자의 과거 QoS 데이터가 유사하면 호출 환경이 유사하여 향후 유사한 QoS를 경험할 가능성이 높다는 것이 일반적이다 [14]. 이러한 관점에서 본질적으로 사용자와 서비스 간의 유사성을 모델링하여 작동하는 협업 필터링 접근 방식 [15, 16]이 QoS 예측에 적합해진다 [13, 14]. 지금까지 협업 필터링 기반의 QoS 예측 모델은 클라우드 컴퓨팅 기반 애플리케이션 [17] 및 멀티미디어 서비스 기반 애플리케이션 [18]과 같은 많은 서비스 기반 애플리케이션을 위한 효율적인 솔루션을 제공하는 데 큰 진전을 이루었다. QoS에 기반한 웹 서비스 추천에서는 협업 필터링을 사용하는 연구들이 진행되어왔다 [3, 9, 14]. 사용자-항목 이분 그래프에 랜덤 워크 (Random walk)를 사용하여 Web QoS 값을 예측하는 프레임워크를 제안한 연구들이 있다 [6, 19, 20]. 선호도 전파를 사용하는 LMF-PP [6, 20]는 사용자와 서비스의 위치 정보를 기반으로 사용자와 서비스를 그룹화한 다음 사용자와 항목 간의 이분 그래프를 모델링하는 선호도 전파를 채택하였다. 이 그래프에서 랜덤 워크를 구성하여 유사성 계산을 통해 모든 사용자에게 대한 유사성 정보를 얻을 수 있었다. 이처럼 랜덤 워크 프로세스를 사용하여 사용자와 유사한 이웃 사용자들을 식별하여 더 나은 웹 서비스 QoS 예측을 수행한다.

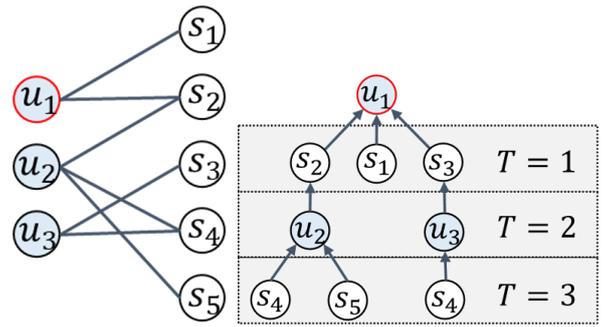


그림 1: 이분 그래프로 표현되는 사용자-서비스 행렬과 고차 연결의 예시

최근까지도 그래프 구조를 사용하여 QoS 값을 예측하지만 그래프 컨볼루션 네트워크를 적용한 협업 필터링 알고리즘 모델의 사례는 없다 [21, 22]. 따라서 본 연구는 QoS 웹 서비스 추천을 위한 그래프 컨볼루션 네트워크의 적용 가능성을 연구한다.

2.2 협업 필터링

협업 필터링은 Matrix Factorization 기반 협업 필터링과 그래프 컨볼루션 기반 협업 필터링으로 분류된다.

2.2.1 Matrix Factorization 기반 협업 필터링

초기 협업 필터링 모델은 행렬 분해에 초점을 맞추었다. 행렬 분해 모델은 두 개의 하위 행렬인 사용자 및 항목 행렬의 행렬 곱으로 인터랙션 행렬 $R_{u,v}$ 를 근사화하는 것을 목표로 한다. Singular Value Decomposition (SVD)는 처음에 특징 행렬을 학습하기 위해 제안되었다. 이후 많은 다른 행렬 분해 방법이 제안되었다. BPR-MF [23]는 사용자가 한 항목을 다른 항목보다 선호하는지 평가하기 위해 제시된 기법이며, 각 사용자와 아이템의 ID를 임베딩으로 초기화하고 내적 곱으로 상호 작용을 예측한다. 최근 몇 년 동안 많은 딥 러닝 접근 방식이 개발되었으며, 그 중 일부는 비선형 신경 아키텍처를 사용하여 고전적인 행렬 분해 알고리즘을 확장한다. [24]. NeuMF [24]와 같은 신경망 기반 협업 필터링은 비선형 신경망을 사용한다.

2.2.2 그래프 컨볼루션 기반 협업 필터링

다른 관련 연구는 사용자-서비스(아이템) 그래프 구조를 활용하는 것이다 (그림 1(a) 참고). 사용자-서비스 상호작용 그래프의 관점에서 개별 상호작용의 기록은 사용자의 1차 연결성과 동일하다. 따라서 사용자-서비스 그래프 구조에서 고차 연결성을 마이닝하는 것은 자연스러운 확장이다. 예를 들어,

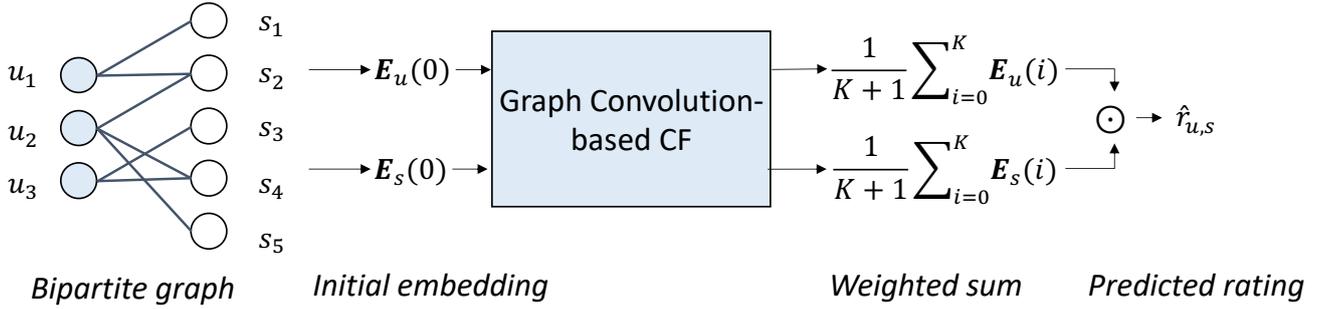


그림 2: 그래프 컨볼루션 기반 협업 필터링의 프레임워크. 파란색 박스는 NGCF, LightGCN, LT-OCF으로 구성할 수 있다.

표 2: 논문에서 사용하는 표기

Notation	Description
\mathbf{R}	사용자-서비스 행렬
\mathbf{A}	인접 행렬
$\tilde{\mathbf{A}}$	symmetric normalized 인접 행렬
$\mathbf{E}_u(i), \mathbf{E}_s(i)$	i 번째 레이어의 사용자 u 와 웹 서비스 s 의 임베딩
K	전체 레이어 개수
α	ReLU 활성화 함수
$\hat{r}_{u,s}$	웹 서비스 s 에 대한 사용자 u 의 예측된 레이팅
\odot	내적 계산
D	임베딩 벡터의 차원 크기
\mathbf{W}	학습 가능한 가중치 행렬
γ	응답 시간에 대한 positive 임계값
w_i	각 레이어 i 에서의 가중치 값

사용자의 2차 연결성은 동일한 항목과 함께 상호 작용한 유사한 사용자로 구성된다. 다행히 다양한 기계 학습 영역에서 그래프 구조 데이터를 모델링하기 위한 GCN (Graph Convolutional Network)의 개발 및 성공으로 최근 그래프 컨볼루션 기반 협업 필터링 [10, 10–12] 모델들이 인기가 많았다. GNN은 여러 홉(hop)에 대한 이웃 모델링을 통해 임베딩을 학습하도록 한다 (그림 1(b) 참고). NGCF [10]는 사용자-아이템 간의 협업 신호를 포착하기 위해 상호 작용 인코더를 제안했다. 이때 비선형 활성화 함수와 학습 가능한 가중치 행렬이 포함된 GNN을 사용하여 높은 고차 연결성에 대한 협업 신호를 사용한다. LightGCN [11]은 NGCF의 비선형성과 가중치를 제거하여 성능과 효율성을 확보한다. LT-OCF [12]는 확산 방정식에서 영감을 받아 선형 GCN을 상미분방정식 컨셉으로 재설계하고 추천 시스템의 벤치마크 데이터셋에서 최고의 성능을 보인다.

3 연구 방법

그래프 기반 협업 필터링 방법에 사용되는 표기는 표 2와 같다. 본 연구의 그래프 기반 협업 필터링 프레임워크는 그림 2와 같으며 다음과 같은 순서로 구성된다:

1. 사용자-서비스 행렬 \mathbf{R} 으로부터 초기 사용자 임베딩 $\mathbf{E}_u(0)$ 와 서비스 임베딩 $\mathbf{E}_s(0)$ 을 구성한다.
2. 두 임베딩은 그래프 컨볼루션 기반 CF 모델에 입력되어 전파된 임베딩을 업데이트 한다. 즉 초기 임베딩을 학습하여 사용자와 서비스 임베딩 $\mathbf{E}_u(K), \mathbf{E}_s(K)$ 을 출력한다.
3. 각 레이어에서 나온 임베딩들의 가중치 합을 사용하여 최종 예측 레이어에서 레이팅 $\hat{r}_{u,s}$ 을 예측한다.

다음 하위 목차에서는 그래프 컨볼루션 레이어, 예측 레이어, 그리고 훈련 방법을 소개한다.

3.1 그래프 컨볼루션 레이어

사용자-서비스 행렬이 \mathbf{R} 일 때 인접행렬 \mathbf{A} 는 다음과 같다.

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{R} \\ \mathbf{R}^T & 0 \end{bmatrix}. \quad (1)$$

0번째 레이어의 임베딩 행렬을 $\mathbf{E}(0)$ 이라 하면 LightGCN과 LT-OCF의 선형 그래프 컨볼루션의 행렬을 다음과 같이 나타낼 수 있다.

$$\mathbf{E}_u(t+1) = \tilde{\mathbf{A}}\mathbf{E}_s(t), \quad (2)$$

$$\mathbf{E}_s(t+1) = \tilde{\mathbf{A}}\mathbf{E}_u(t). \quad (3)$$

$\tilde{\mathbf{A}}$ 는 \mathbf{A} 의 symmetric normalized 행렬이며 $\tilde{\mathbf{A}}$ 와 임베딩 행렬과의 행렬 곱을 통해 전파가 수행된다. 따라서 $\mathbf{E}_s(1)$ 은 임베딩 행렬 $\mathbf{E}_s(0)$ 이 이웃 정보로부터 전파되어 업데이트된 것이다. 비선형 그래프 컨볼루션을 사용하는 NGCF는 다음과 같

은 전파를 진행한다.

$$\mathbf{E}_u(t+1) = \alpha(\tilde{\mathbf{A}}\mathbf{E}_s(t)\mathbf{W}), \quad (4)$$

$$\mathbf{E}_s(t+1) = \alpha(\tilde{\mathbf{A}}\mathbf{E}_u(t)\mathbf{W}). \quad (5)$$

α 는 ReLU 활성화 함수이고 \mathbf{W} 는 학습 가능한 가중치 행렬이다. 최종 임베딩 행렬은 각 레이어에서의 임베딩들에 대한 가중치 합으로 생성된다 (그림 7 참고). 모델 예측은 사용자 및 서비스의 최종 임베딩에 대한 내적을 계산하여 $r_{(u,s)}$ 을 예측한다. 협업 필터링 분야에서 일반적으로 사용하는 BPR (Bayesian Personalized Ranking) [23] 손실 함수로 초기 임베딩 $\mathbf{E}(0)$ 을 학습한다.

3.2 예측 레이어

예측 레이어는 모든 K 레이어를 통해 전파한 후 최종 임베딩을 구성하기 위한 레이어이다. 최종 임베딩은 각 레이어의 임베딩들의 가중치 합을 사용한다.

$$\mathbf{E}_u(\text{final}) = \sum_{i=0}^K w_i \mathbf{E}_s(i), \quad (6)$$

$$\mathbf{E}_s(\text{final}) = \sum_{i=0}^K w_i \mathbf{E}_u(i),$$

w_i 는 각 레이어에서의 가중치를 의미한다. w 값이 모든 i 에서 동일하다면 모든 레이어에서의 임베딩 값의 평균을 사용하는 것이고 $w_0 = 0$ 이라면 초기 임베딩 $\mathbf{E}_u(0)$ 과 $\mathbf{E}_s(0)$ 을 사용하지 않는 것을 의미한다. 따라서 이 가중치 합은 마지막 레이어의 임베딩뿐만 아니라 이전 레이어의 임베딩을 사용하여 좋은 성능을 낼 수 있다. $\mathbf{E}_u(\text{final})$ 와 $\mathbf{E}_s(\text{final})$ 를 구한 후 서비스 s 에 대한 사용자 u 의 레이팅을 예측한다. 이를 위해 각 레이어에서 $\mathbf{E}_u(\text{final})$ 와 $\mathbf{E}_s(\text{final})$ 의 내적을 한다.

$$\hat{r}_{u,s} = \mathbf{E}_u(\text{final}) \odot \mathbf{E}_s(\text{final}), \quad (7)$$

\odot 는 내적을 의미하며 $\hat{r}_{u,s}$ 은 최종 레이팅 값이 된다.

3.3 훈련 방법

우리는 Bayesian Personalized Ranking (BPR) 손실 함수 [23]를 사용한다. BPR 손실 함수는 관찰된 항목의 예측이 관찰되지 않은 항목보다 더 높을 것이라는 전제하는 예측하는 pairwise 손실 함수이다.

$$L = - \sum_{u=1}^{|U|} \sum_{i \in N_u} \sum_{j \notin N_u} \ln(\sigma(\hat{r}_{ui} - \hat{r}_{uj})) + \lambda \|\Theta\|^2, \quad (8)$$

σ 는 sigmoid 함수이다. Θ 는 학습 가능한 파라미터이다. LightGCN, LT-OCF의 경우 초기 임베딩만 해당되는 반면에,

Algorithm 1: 그래프 컨볼루션 기반 협업 필터링의 학습 과정

Input: Web QoS graph \mathbf{A} , The number of total layers K , *model*

```

1 Initialize  $\mathbf{E}_u(0)$  and  $\mathbf{E}_s(0)$ ; while the BPR loss  $L$  is
  not converged do
2   for  $i \leftarrow 1$  to  $K$  do
3     if model = LightGCN or LT-OCF then
4        $\mathbf{E}_u(i), \mathbf{E}_s(i) = \text{Eq. 2, Eq. 3}$ ;
5     else if model = NGCF then
6        $\mathbf{E}_u(i), \mathbf{E}_s(i) = \text{Eq. 4, Eq. 5}$ ;
7    $\mathbf{E}_u(\text{final}), \mathbf{E}_s(\text{final}) = \text{Eq. 6}$ ;
8    $\hat{r}_{u,s} = \text{Eq. 7}$ ;
9   if model = LightGCN or LT-OCF then
10    Update  $\mathbf{E}_u(0)$  and  $\mathbf{E}_s(0)$  with BPR loss;
11  else if model = NGCF then
12    Update  $\mathbf{W}, \mathbf{E}_u(0)$  and  $\mathbf{E}_s(0)$  with BPR loss;
13 return  $\mathbf{E}_u(0)$  and  $\mathbf{E}_s(0)$ ;

```

NGCF에서 초기 임베딩 뿐만 아니라 다른 학습 가능한 가중치 행렬들도 포함된다. λ 는 L_2 정규화의 정도를 조절한다. 우리는 관찰된 사용자-서비스 인터랙션들을 positive 인스턴스로 사용하고 [11]에서 사용한 negative 인스턴스를 샘플링하는 전략을 사용한다.

그래프 컨볼루션 기반 협업 필터링의 훈련 알고리즘은 Algorithm 1에 있으며 BPR 손실이 수렴될 때까지 각 부분을 반복하여 훈련한다. NGCF는 초기 임베딩과 학습 가능한 가중치 행렬을 학습하는 반면에, LightGCN과 LT-OCF의 경우 초기 임베딩들만을 학습한다.

4 실험 설정

실험 설정 목차에서는 연구질문을 설정하고 데이터셋, 평가 척도, 비교 모델, 하이퍼파라미터를 소개한다.

4.1 연구질문

설정된 연구질문은 아래와 같다.

- RQ1:** 그래프 컨볼루션 기반 협업 필터링 알고리즘은 MF 기반 협업 필터링 알고리즘보다 우수한가?
- RQ2:** 그래프 컨볼루션 기반 협업 필터링 알고리즘은 데이터셋의 희소성에 따른 성능 변화가 있는가?

표 3: WSDream의 데이터셋 정보

Dataset	WSDream (Warm-start)	WSDream (Cold-start)	WSDream (Cold-start*)
# Users	338	338	338
# Services	5,824	5,824	5,824
γ	0.05	0.02	0.01
Core	10	2	2
Density	13.64%	5.36%	3.74%
# Interactions	57,727	8,490	1,036

4.2 실험환경

데이터 세트는 6:2:2의 비율로 분할되어 있다. 모든 실험은 다음과 같은 소프트웨어 및 하드웨어 환경에서 수행되었다: UBUNTU 18.04 LTS, PYTHON 3.9.6, NUMPY 1.20.3, SCIPY 1.7, MATPLOTLIB 3.3.1, PYTORCH 1.8.0, CUDA 11.2, and NVIDIA Driver 417.22, i9 CPU, and NVIDIA RTX TITAN.

4.3 데이터셋

실험에서 사용하는 웹 서비스 QoS 데이터셋은 WSDream [3]이며 응답시간 (response time)을 QoS 값으로 사용한다*. 실험에 사용되는 데이터셋은 표 3과 같고 테스트셋 비율은 20%이다. 데이터셋을 구성하기 위해 응답시간 기준으로 γ 가 특정 값 이상일 경우에 연결성이 있다고 가정한다. 예를 들어 γ 가 0.05s 이하의 경우 positive 인터랙션으로 간주하여 그래프를 구성한다. Warm-start 환경의 인터랙션 수는 57,727 개의 인터랙션이 있는 반면에, Cold-start 환경에 대한 데이터셋은 두 가지이며 각각 8,490과 1,036 개의 인터랙션 행렬이 있다. Cold-start*는 밀도 (density)가 3.74%인 경우를 의미하며 5.36%인 Cold-start 보다 극단적인 환경이다. Warm-start 환경의 경우 훈련 데이터셋과 테스트 데이터셋을 구성할 수 있도록 하나의 사용자가 10개 이상의 인터랙션이 있도록 구성하였다. Cold-start 환경의 경우 2개 이상의 인터랙션이 있도록 필터링 하였다.

4.4 평가척도

테스트셋에 있는 각각의 사용자에게 대해서, 인터랙션이 없는 서비스를 모두 negative 샘플로 여기고, 모델은 훈련 데이터셋에서 사용된 positive 샘플을 제외한 모든 샘플에 대한 사용자의 랭킹을 계산한다. 평가를 위해서 랭킹 기반 평가 척도인

Recall@k와 NDCG@k를 사용한다. Recall@k은 전체 서비스 중 추천된 서비스 k개가 속한 비율을 나타내며 다음과 같이 정의한다:

$$\text{Recall@k} = \frac{\text{rel}_k}{\min(k, \text{rel})}. \quad (9)$$

rel_k 는 상위 k개 결과의 관련 항목 수를 의미하고, rel 은 사용자와 관련된 총 항목 수를 의미한다. Recall@k는 사용자와 연관된 모든 서비스 중에서 모델이 추천한 시스템 k개가 얼마나 포함되는지의 비율을 의미한다. NDCG (Normalized Discounted Cumulative Gain)는 추천 항목 목록과 최적 순위 항목 목록 간의 차이를 평가하며 다음과 같이 정의한다:

$$\text{NDCG@k} = \frac{\text{DCG@k}}{\text{IDCG@k}}. \quad (10)$$

NDCG@k는 추천의 순서에 가중치를 두어서 성능을 평가하며 1에 가까울 수록 좋은 성능이다. DCG@k과 IDCG@k은 각각 예측 순위와 이상적인 순위의 상위 k개의 항목의 DCG(Discounted Cumulative Gain)이다. DCG@k은 다음과 같이 계산된다:

$$\text{DCG@k} = \sum_{i=1}^k \frac{2^{\text{rel}_i - 1}}{\log_2(i + 1)}. \quad (11)$$

여기서 rel_i 은 순위 위치에 있는 항목의 값이다. NDCG의 값은 0과 1 사이이며 값이 클수록 순위가 높고 1이 이상적인 순위를 나타낸다.

4.5 비교 모델

MF 기반 모델 두 가지와 그래프 컨볼루션 기반 협력 필터링 모델 세 가지를 사용한다. 각 모델에 대한 설명은 다음과 같다:

- **MF 기반 협업 필터링:** BPR-MF [23]은 암시적 피드백을 학습하도록 pair-wise 손실 함수를 최소화하는 고전적인 협업 필터링 알고리즘이다. BPR에서는 MF를 사용하여 사용자 및 항목의 임베딩을 초기화한다. NeuMF [24]는 상호 작용을 캡처하기 위해 사용자 및 항목 임베딩의 상호작용에 비선형 히든 레이어들을 사용하는 협업 필터링 알고리즘이다. 특히 각 히든 레이어의 차원이 동일하게 유지되는 두 개의 히든 레이어 아키텍처를 사용한다.
- **그래프 컨볼루션 기반 협업 필터링:** NGCF [10]는 잔차 예측을 수행하는 비선형 그래프 컨볼루션 기반 협업 필터링 알고리즘이다. LightGCN [11]은 특징 변환 및 비선형 활성화를 제거하여 그래프 컨볼루션 네트워크의 구조를 단순화하는 그래프 컨볼루션 기반 협업 필터링 알고리즘이다. LT-OCF [12]는 선형 그래프 컨볼루션 기반

*<http://wsdream.github.io/dataset>에서 WSDream 데이터셋 다운로드가 가능하다.

표 4: WSDream (Warm-start)에 대한 각 모델들의 Recall@k와 NDCG@k 추천 성능 비교. **볼드체**는 최고 성능, 밑줄은 두번째 최고 성능을 의미한다. *improvement*는 그래프 컨볼루션 기반 협업 필터링 알고리즘이 MF 기반 협업 필터링 알고리즘에 대해 향상된 성능 비율을 의미한다.

Model	WSDream (Warm-start)			
	Recall@20	NDCG@20	Recall@50	NDCG@50
BPR-MF	0.0729	0.2653	0.1680	0.2518
NeuMF	0.0697	0.2580	0.1530	0.2259
NGCF	0.2193	0.4592	0.4327	0.5152
LightGCN	0.2217	0.4661	0.4382	<u>0.5209</u>
LT-OCF	<u>0.2199</u>	0.4661	0.4382	0.5211
<i>improvement</i>	204.12%	75.69%	160.83%	130.68%

협업 필터링 알고리즘을 상미분방정식 개념으로 재설계한 협업 필터링 알고리즘이다.

4.6 하이퍼파라미터

실험에 필요한 몇 가지 공통 하이퍼파라미터가 있다. 모든 방법의 정규화 계수 λ 는 $\{1.0 \times 10^{-4}, 1.0 \times 10^{-3}, 1.0 \times 10^{-2}\}$ 을 고려한다. 임베딩 벡터의 차원은 $\{16, 32, 64, 128\}$ 의 범위 내에서 고려하고 $\mathcal{N}(0, 0.1)$ 의 정규 분포는 초기 임베딩을 설정하는 데 사용된다. 레이어 조합 계수는 $w_i = \frac{1}{1+K}$ 으로 평균 값을 사용한다. 여기서 K 는 레이어 조합의 수이다. K 의 개수는 $\{1, 2, 3, 4\}$ 를 고려한다. 우리는 NGCF와 동일한 early stop 알고리즘을 사용하고 learning rate는 $\{1.0 \times 10^{-4}, 1.0 \times 10^{-3}, 1.0 \times 10^{-2}\}$ 의 범위에서 실험을 진행한다. NGCF의 node dropout은 $\{0.0, 0.1, 0.2, 0.3, 0.4\}$ 의 범위에서 고려하고 LightGCN과 LT-OCF는 dropout을 사용하지 않는다.

5 실험 결과

5.1 그래프 컨볼루션 기반 협업 필터링 모델의 성능 (RQ1)

표 4의 WSDream (Warm-start) 데이터셋에서는 협업 필터링 모델들과 그래프 컨볼루션 기반 협업 필터링 모델들의 추천 성능 결과를 확인할 수 있다. BPR-MF와 NeuMF의 경우 그래프 컨볼루션 기반 협업 필터링 모델들보다 낮은 추천 성능을 보인다. 그 이유는 사용자와 서비스의 이웃 정보와 고차 연결성을 활용하지 못하여 절절한 임베딩으로 업데이트 하는데 한계가 있기 때문이다. 반면에 NGCF, LightGCN과 LT-OCF는 명시적인 방식으로 이웃 정보의 고차 연결을 탐색할 수 있다.

LightGCN과 LT-OCF의 최고 성능을 보이거나 두번째 최고 성능을 도달한다. Recall@20의 경우 LightGCN이 0.2217

로 LT-OCF보다 더 좋은 추천 성능을 보이지만 Recall@50과 NDCG@50의 기준으로는 LT-OCF가 더 나은 성능을 보인다. Recall@20 기준으로 LightGCN은 BPR-MF 보다 204.12% 향상되었으며 NDCG@20 기준으로 LT-OCF는 BPR-MF의 추천 성능보다 160.83%의 성능 향상을 보인다. 비선형 그래프 컨볼루션을 사용하는 NGCF는 선형 그래프 컨볼루션을 사용하는 LightGCN과 LT-OCF보다 낮은 추천 성능을 보인다. NDCG@20 기준으로 LightGCN과 LT-OCF는 NGCF 대비 1.50% 향상하는 것을 확인할 수 있다. 이를 통해 Web QoS 추천에서 그래프 컨볼루션의 선형성을 고려하는 것이 비선형성을 고려하는 것 보다 더 좋은 추천 성능을 보인다는 것을 알 수 있다.

5.2 Cold-start 환경에서의 성능 차이 (RQ2)

표 3와 같이 Cold-start 환경 [6]을 구성하도록 하기 위해 각각 5.36%와 3.74%의 밀도를 가진 WSDream(Cold-start)와 WSDream(Cold-start*) 데이터셋을 사용한다. 표 5과 6는 희소한 인터랙션을 가진 두 데이터셋에 대한 실험 결과이다.

5.2.1 WSDream(Cold-start)의 결과

표 5에서는 LightGCN과 LT-OCF가 가장 좋은 성능을 보인다. Recall@20과 NDCG@20에서는 LightGCN이 더 좋은 성능을 보인다. 반면에 Recall@50과 NDCG@20의 경우에는 LT-OCF가 더 높은 추천 성능을 보인다. 그래프 컨볼루션 기반 협업 필터링 알고리즘은 Warm-start 환경보다 Cold-start 환경에서 더 높은 성능 향상을 보인다. Recall@50 기준으로 Cold-start 데이터셋이 811.951% 향상을 보였다. 이는 Warm-start 환경에서의 160.83% 향상에 비하면 큰 차이의 성능 향상을 보인다. 이 환경에서는 동일하게 그래프 컨볼루

표 5: WSDream (Cold-start)에 대한 각 모델들의 Recall@k와 NDCG@k 추천 성능 비교

Model	WSDream (Cold-start)			
	Recall@20	NDCG@20	Recall@50	NDCG@50
BPR-MF	0.0469	0.0324	0.0979	0.0509
NeuMF	0.0946	0.0420	0.0931	0.0487
NGCF	0.3782	0.3334	0.6038	0.4109
LightGCN	0.6457	0.5549	<u>0.8845</u>	<u>0.6374</u>
LT-OCF	<u>0.6435</u>	<u>0.5545</u>	0.8928	0.6503
<i>improvement</i>	1276.76%	1612.65%	811.951%	1177.6%

표 6: WSDream (Cold-start*)에 대한 각 모델들의 Recall@k와 NDCG@k 추천 성능 비교

Model	WSDream (Cold-start*)			
	Recall@20	NDCG@20	Recall@50	NDCG@50
BPR-MF	0.1130	0.0459	0.2400	0.0792
NeuMF	0.0931	0.0487	0.2386	0.0790
NGCF	0.5067	0.3328	0.7205	0.3874
LightGCN	<u>0.8989</u>	<u>0.7187</u>	<u>0.9288</u>	<u>0.7316</u>
LT-OCF	0.9359	0.7316	0.9840	0.7476
<i>improvement</i>	728.23%	1493.9%	312.41%	843.94%

선 기반 협업 필터링이 더 나은 성능을 보인다. Recall@50 과 NDCG@50은 LT-OCF가 가장 좋은 성능을 보이는 반면에 Recall@20과 NDCG@20의 경우에는 LightGCN이 더 높은 성능을 보인다.

5.2.2 WSDream(Cold-start*)의 결과

표 6에서도 그래프 컨볼루션 기반 협업 필터링 알고리즘이 MF 기반 협업 필터링 알고리즘 보다 더 좋은 추천 성능을 보인다. LT-OCF는 모든 평가척도에서 가장 높은 추천 성능을 보이며 Recall@20 기준으로 LightGCN 보다 4.12% 향상되고 NGCF 보다 84.71% 향상된 것을 확인할 수 있다. 또한 BPR-MF의 경우 0.2400의 Recall@50 성능을 보인 반면에, LT-OCF는 312.41% 향상된 0.9840의 추천 성능을 보인다.

이 실험을 통해서 **RQ2**에 대한 데이터셋이 희소한 인터랙션을 가진 Cold-start 환경에서도 그래프 컨볼루션 기반 협업 필터링 알고리즘이 더 우수한 방법을 보이는 것을 알 수 있다. 따라서 Warm-start 환경과 Cold-start 환경 모두에서 고차 연결성을 고려하는 그래프 컨볼루션 기반 협업 필터링 알고리즘의 필요성을 보인다.

6 토의

6.1 민감도 분석: 레이어의 개수에 따른 성능 차이가 있는가?

레이어 개수에 따른 성능 차이는 표 7에서 확인할 수 있다. Warm-start 환경에서는 LightGCN의 레이어가 많을 수록 (고차 연결 정보를 고려할 수록) Recall@20과 NDCG@20의 추천 성능이 향상되는 것을 보인다. 이를 통해 Warm-start 환경에서는 그래프 구조의 고차 연결성을 고려하는 것이 중요하다는 것을 알 수 있다. 반면 Cold-start 환경에서는 레이어 수가 증가할 수록 성능이 하락하는 경향을 보인다. 추천 성능이 하락하는 이유는 전파가 반복되면서 임베딩에 오버스무딩 (over-smoothing) 현상 [25]이 일어나기 때문이다. 그래서 Cold-start 환경에서는 그래프 컨볼루션을 한 번만 사용하는 것이 최고의 추천 성능을 보인다. 더 희소한 인터랙션을 가진 Cold-start* 환경에서는 적절한 레이어 개수에서 최고의 추천 성능을 보인다. Recall@20 기준으로 $K = 3$ 일 때 가장 좋은 추천 성능을 보이며 NDCG@20의 경우 $K = 4$ 일 때 최고의 추천 성능을 보인다.

표 7: LightGCN의 레이어 개수에 따른 성능 변화 (볼드체: 최고 성능)

K	WSDream (Warm-start)		WSDream (Cold-start)		WSDream (Cold-start*)	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
1	0.1935	0.4013	0.6457	0.5549	0.8846	0.6977
2	0.2056	0.4388	0.6337	0.5417	0.8952	0.7200
3	0.2105	0.4554	0.6241	0.5332	0.8989	0.7187
4	0.2217	0.4661	0.6060	0.5269	0.8982	0.7244

표 8: WSDream (Warm-start) 데이터셋에서의 임베딩 벡터의 차원 크기 (D)에 따른 성능 변화

D	Recall@20	Recall@50
16	0.2109	0.4245
32	0.2186	0.4340
64	0.2217	0.4661
128	0.2193	0.4353

표 9: WSDream (Cold-start) 데이터셋에서의 임베딩 벡터의 차원 크기 (D)에 따른 성능 변화

D	Recall@20	Recall@50
16	0.5610	0.8277
32	0.5948	0.8676
64	0.6457	0.8845
128	0.6372	0.8862

6.2 민감도 분석: 임베딩 크기를 변경할 때 그래프 컨볼루션 기반 협업 필터링 알고리즘이 얼마나 정확하냐?

표 8와 9는 임베딩 크기에 따른 LightGCN의 성능 변화를 나타낸다. 표 8에서는 Warm-start 환경에서의 추천 성능 변화를 나타낸다. LightGCN은 임베딩 크기가 64 일 때 Recall@20과 Recall@50의 성능이 가장 높으며 임베딩 크기가 128일 때는 성능이 다시 하락하는 것을 알 수 있다. 표 9에서는 Cold-start 환경에서의 추천 성능 변화를 나타낸다. Warm-start 환경에서와 다른 점은 D 가 128일 때 Recall@50이 향상되는 것을 확인할 수 있다. 하지만 Recall@20에서는 여전히 D 가 64일 때 가장 높은 성능을 보이며 16일 때는 Recall@20이 0.5948로 7.88% 하락한다.

표 10: WSDream (Warm-start) 데이터셋에서의 훈련 복잡도 분석. Training Time은 한 에폭에 대한 초를, #Parameters는 모델의 파라미터 수를 나타낸다.

Model	Training Time (s)	#Parameters
BPR-MF	0.0375	105,728
Neu-MF	0.1087	358,465
NGCF	0.1662	122,112
LightGCN	0.0866	105,728
LT-OCF	0.1688	105,728

표 11: WSDream (Cold-start) 데이터셋에서의 훈련 복잡도 분석

Model	Training Time (s)	#Parameters
BPR-MF	0.0186	54,400
Neu-MF	0.0516	204,481
NGCF	0.0693	70,784
LightGCN	0.0254	54,400
LT-OCF	0.0661	54,400

6.3 훈련 복잡도 분석: 그래프 컨볼루션 기반 협업 필터링 알고리즘들의 경험적 학습 복잡도는 MF 기반 협업 필터링 알고리즘 보다 복잡하냐?

표 10과 11은 모든 협업 필터링 알고리즘에 대한 한 에폭당 걸리는 시간과 학습 가능한 파라미터 수를 나타낸다. 표 11에서는 BPR-MF는 한 에폭당 0.0186초의 시간이 걸리는 반면에 Neu-MF는 0.0516초의 시간이 걸린다. Neu-MF는 BPR-MF보다 학습할 파라미터들이 약 3.76 배 많고 학습 시간은 대략 약 3배 정도 더 필요하다. NGCF도 LightGCN 보다 3배 이상의 시간이 더 소요된다. 표 10에서 알 수 있듯이, NGCF는 학습 가능한 가중치 행렬이 존재하기 때문에 LightGCN 보다 1.15배 파라미터 수가 많다. LT-OCF의 경우에는 학습 가능

한 가중치 행렬이 존재하지 않지만, 상미분 방정식을 풀기 위한 적분기에서 대략 3배 정도의 계산량을 필요로 한다. 또한 LightGCN과 LT-OCF는 BPR-MF와 동일하게 초기 임베딩들만 학습하기 때문에 파라미터 수가 동일하다 그래서 모델 복잡도가 동일하다.

7 위협 요소

내부 유효성에 대한 위협은 비교 모델로 5개의 협업 필터링을 사용한 것이다. 향후 다양한 협업 필터링들을 추가 실험하여 비교할 계획이다. 구성 유효성에 대한 위협은 이분 그래프를 구성하기 위한 데이터 전처리 단계에 있다. 그래프 구성을 위해 임계값을 기준으로 더 낮은 응답 시간이 사용자와 서비스 간의 인터랙션이 있다고 간주한다. 인터랙션의 여부를 1과 0으로 구분하는 것은 웹 서비스 QoS 데이터를 풍부하게 활용하는데 한계가 있다. 이 한계점을 극복하기 위해 추후에 가중치가 포함된 이분 그래프를 사용하여 그래프 기반 협업 필터링을 설계할 계획이다.

8 결론

그래프 컨볼루션 기반 협업 필터링은 여러 홉에 대한 이웃 정보를 표현할 수 있는 장점이 있다. 본 연구는 그래프 구조를 활용하는 것이 QoS 랭킹 기반 추천에 적용 가능한지 확인하였다. QoS 데이터의 사용자-서비스 그래프를 사용하여 사용자에 대한 서비스 추천 프레임워크를 설계하였다. 또한 QoS에서의 희소성 문제에서 그래프 컨볼루션 기반 협업 필터링의 성능이 모델 기반 협업 필터링 모델들보다 성능이 증가하는 것을 보였다. 추후 QoS에 더 적합한 모델을 설계할 것이고 응답시간 뿐만 아니라 처리량과 같은 다양한 QoS 값을 사용할 계획이다.

Acknowledge

교신 저자는 류덕산. 본 연구는 원자력안전위원회의 재원으로 한국원자력안전재단의 지원을 받아 수행한 원자력안전연구사업(No. 2105030)과 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF-2022R1I1A3069233)의 결과임

참고 문헌

[1] L.-J. Zhang, J. Zhang, and H. Cai, "Services computing. 2007."

[2] Q. Duan, Y. Yan, and A. V. Vasilakos, "A survey on service-oriented network virtualization toward convergence of networking and cloud computing," *IEEE Transactions on Network and Service Management*, vol. 9, no. 4, pp. 373–392, 2012.

[3] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative web service qos prediction via neighborhood integrated matrix factorization," *IEEE Transactions on Services Computing*, vol. 6, no. 3, pp. 289–299, 2012.

[4] T. Yu, Y. Zhang, and K.-J. Lin, "Efficient algorithms for web services selection with end-to-end qos constraints," *ACM Transactions on the Web (TWEB)*, vol. 1, no. 1, pp. 6–es, 2007.

[5] R. Phalnikar and P. A. Khutade, "Survey of qos based web service discovery," in *2012 World Congress on Information and Communication Technologies*, pp. 657–661, IEEE, 2012.

[6] D. Ryu, K. Lee, and J. Baik, "Location-based web service qos prediction via preference propagation to address cold start problem," *IEEE Transactions on Services Computing*, vol. 14, no. 3, pp. 736–746, 2018.

[7] J. Choi, J. Lee, D. Ryu, S. Kim, and J. Baik, "Gainqos: A novel qos prediction model for edge computing," *Journal of Web Engineering*, vol. 21, no. 1, pp. 27–51, 2022.

[8] D. Wu, X. Luo, M. Shang, Y. He, G. Wang, and X. Wu, "A data-characteristic-aware latent factor model for web services qos prediction," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[9] M. Chen, Y. Ma, B. Hu, and L.-J. Zhang, "A ranking-oriented hybrid approach to qos-aware web service recommendation," in *2015 IEEE International Conference on Services Computing*, pp. 578–585, IEEE, 2015.

[10] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019.

[11] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph

- convolution network for recommendation,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 165–174, 2020.
- [12] J. Choi, J. Jeon, and N. Park, “Lt-ocf: Learnable-time ode-based collaborative filtering,” in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021.
- [13] S. H. Ghafouri, S. M. Hashemi, and P. C. K. Hung, “A survey on web service qos prediction methods,” *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 2439–2454, 2022.
- [14] Z. Zheng, L. Xiaoli, M. Tang, F. Xie, and M. R. Lyu, “Web service qos prediction via collaborative filtering: A survey,” *IEEE Transactions on Services Computing*, 2020.
- [15] H. Shin, S. Kim, J. Shin, and X. Xiao, “Privacy enhanced matrix factorization for recommendation with local differential privacy,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1770–1782, 2018.
- [16] Y. He, C. Wang, and C. Jiang, “Correlated matrix factorization for recommendation with implicit feedback,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 3, pp. 451–464, 2019.
- [17] Z. ur Rehman, O. K. Hussain, F. K. Hussain, E. J. Chang, and T. S. Dillon, “User-side qos forecasting and management of cloud services,” *World Wide Web*, vol. 18, pp. 1677–1716, 2015.
- [18] M. S. Hossain, “Qos in web service-based collaborative multimedia environment,” in *16th International Conference on Advanced Communication Technology*, pp. 881–884, 2014.
- [19] M. Tang, X. Dai, B. Cao, and J. Liu, “Wswalker: A random walk method for qos-aware web service recommendation,” in *2015 IEEE International Conference on Web Services*, pp. 591–598, 2015.
- [20] K. Lee, J. Park, and J. Baik, “Location-based web service qos prediction via preference propagation for improving cold start problem,” in *2015 IEEE International Conference on Web Services*, pp. 177–184, 2015.
- [21] T. E. Trueman, P. Narayanasamy, and J. Ashok Kumar, “A graph-based method for ranking of cloud service providers,” *The Journal of Supercomputing*, vol. 78, no. 5, pp. 7260–7277, 2022.
- [22] Z. Chang, D. Ding, and Y. Xia, “A graph-based qos prediction approach for web service recommendation,” *Applied Intelligence*, vol. 51, no. 10, pp. 6728–6742, 2021.
- [23] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “Bpr: Bayesian personalized ranking from implicit feedback,” in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, p. 452–461, 2009.
- [24] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-s. Chua, “Neural Collaborative Filtering,” in *TheWebConf (former WWW)*, 2017.
- [25] K. Oono and T. Suzuki, “Graph neural networks exponentially lose expressive power for node classification,” in *ICLR*, 2020.