



Paper Review

Neural Graph Collaborative Filtering (NGCF)

Xiang Wanf, Xiangnan Han, Meng Wang, Feli Feng,
Tat-Seng Chua

2019 (SIGIR)

Simplifying and Powering Graph Convolution Network for Recommendation (LightGCN)

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong
Zhang, Meng Wang

2020 (SIGIR)

HTET ARKAR
School of Computer Science and Engineering
Chung-Ang University

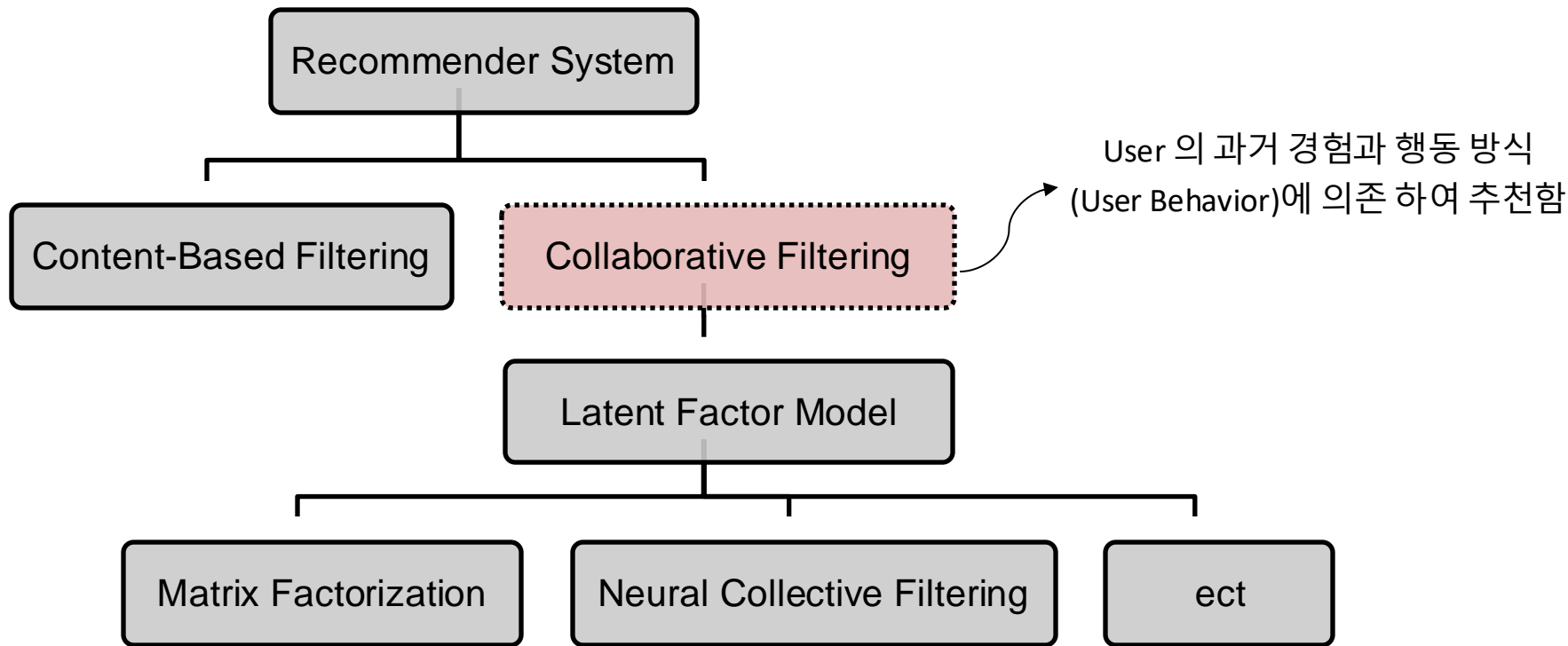
2024-08-06

□ Part I

- Introduction
- Collaborative Filtering
- Proposed Method I (NGCF)

□ Part II

- Problem
- Ablation Study
- Proposed Method II (LightGCN)
- Experiment
- Conclusion



□ What is Collaborative Filtering (협업 필터링)?

1. 내가 좋아하는 감독, 장르, 키워드의 영화를 찾아본다



Content-Based Filtering

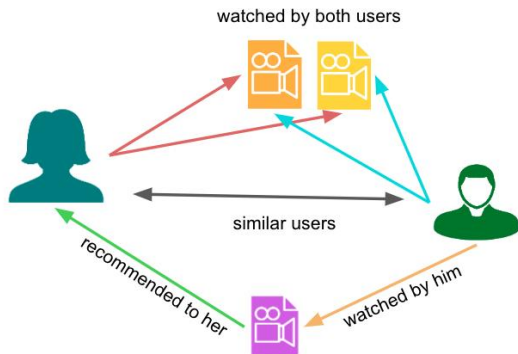
2. 나랑 성향이 비슷한 친구들이 본 영화를 찾아본다



Collaborative Filtering

□ Characteristics of Collaborative Filtering

- 가정 : 나와 비슷한 취향의 사람들이 좋아하는 것은 나도 좋아할 가능성이 높다
- 핵심 포인트 : "많은 사용자들"로부터 얻은 취향 정보를 활용
 - 사용자의 취향 정보 = 집단 지성
 - 축적된 사용자들의 집단 지성을 기반으로 추천



☐ **Types of Collaborative Filtering**

- Memory Based Approach

- Model Based Approach

- ☐ Non-Parametric Approach

- ☐ Matrix Factorization (행렬 분해) based Algorithm

- ☐ Deep Learning

□ Matrix Factorization

- 유저-아이템 상호작용의 잠재 요인(latent factor)을 고려하여 유저에게 적합한 아이템을 추천
- Collaborative Signal 을 latent factor 간의 곱셈을 선형으로 결합하는 내적(저차원 공간)을 통해 나타남
- 복잡한 구조의 signal 를 알아내기 어려움
- 새로운 User 가 나타나면 저차원 공간에 이를 표현하기가 어려움



Neural Collaborative Filtering

Collaborative Signal : patterns and information derived from the collective behavior and interactions of a group of users

□ Neural Collaborative Filtering

- Deep Neural Network 를 사용해 Collaborative Signal을 학습
- Non-linear 한 요소를 표현할 수 있음
- 유저-아이템 상호작용을 나타내기에 아직 부족함



Neural **Graph** Collaborative Filtering

□ NGCF 등장 이유

- 일반적으로 Collaborative Filtering Model 은 두 개의 주요 요소로 구성
 - Embedding : 유저와 아이템을 벡터로 변환하는 과정
 - Interaction Modeling : Embedding 을 기반으로 historical interaction(구매 혹은 클릭)을 재구성
- 기존의 CF model 은 user-item interaction 을 명시적으로 사용하지 않았음
 - 유저와 아이템 각각의 feature만을 embedding에 사용

□ Neural Graph Collaborative Filtering - NGCF

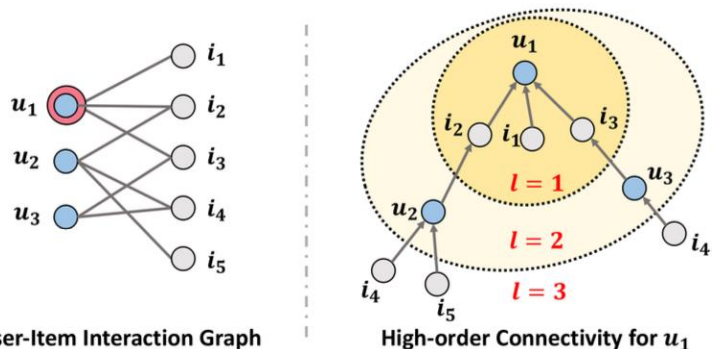


Figure 1: An illustration of the user-item interaction graph and the high-order connectivity. The node u_1 is the target user to provide recommendations for.

□ User-Item Interaction Graph

■ 유저가 아이템을 선택 => 끝

□ High-order Connectivity

■ 유저와 아이템간 관계를 그래프적으로 표현

■ Sequential 한 관계 (High-order)

□ Neural Graph Collaborative Filtering - NGCF

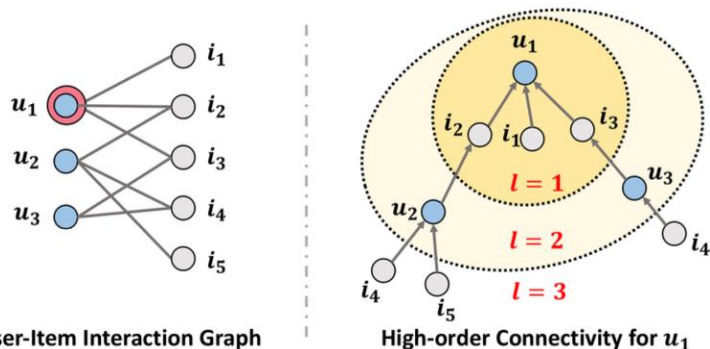


Figure 1: An illustration of the user-item interaction graph and the high-order connectivity. The node u_1 is the target user to provide recommendations for.

□ User-Item Interaction Graph

- 사용자가 아이템을 선택 => 끝

□ High-order Connectivity

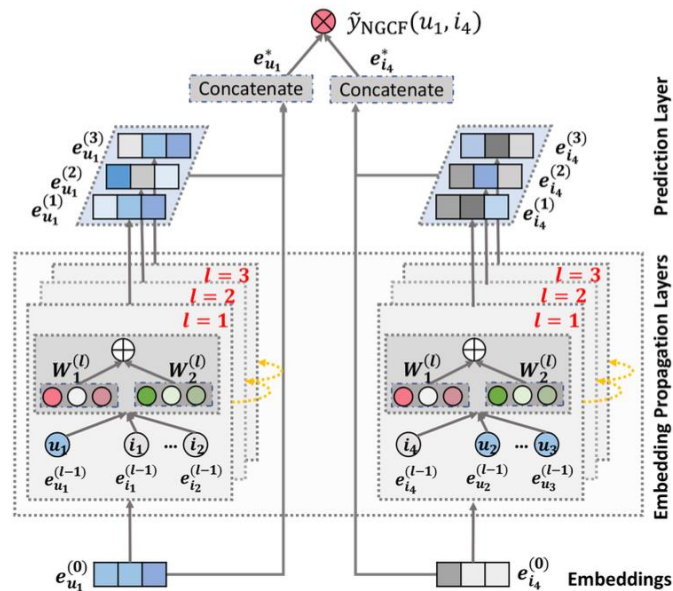
- $u_1 \leq i_2 \leq u_2$
(u_1 과 u_2 간 유사성 존재)
- $u_1 \leq i_2 \leq u_2 \leq i_4$
(u_1 은 i_4 사용할 가능성 존재)
- u_1 은 i_5 보다 i_4 를 선호할 것
(u_1 와 유사한 또 다른 유저 u_3 도 i_4 를 사용)

□ High-order Connectivity

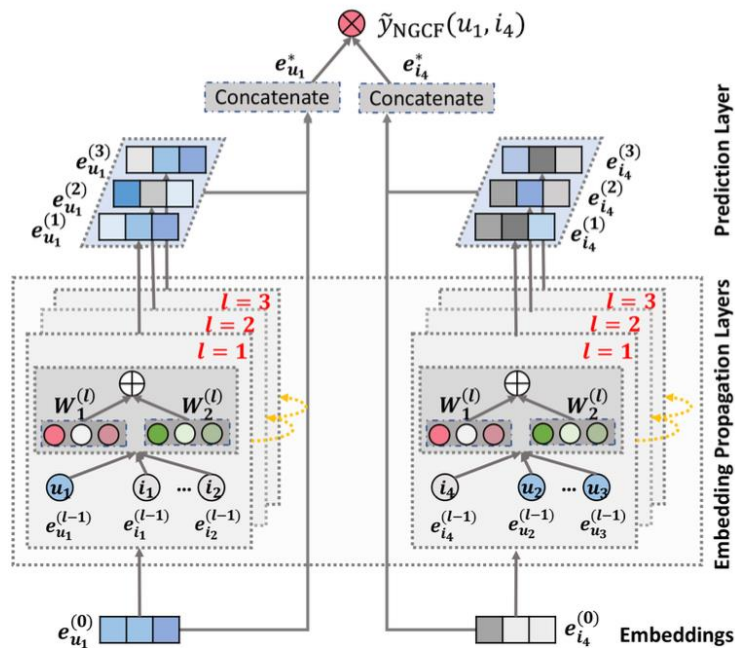
■ GNN (Graph Neural Network) 착용

- Graph 로 Embedding 을 전파할 수 있음
- 정보의 흐름을 embedding space 에 명시적으로 반영
- Embedding propagation layer 를 이용

Collaborative signal 을 포착 가능

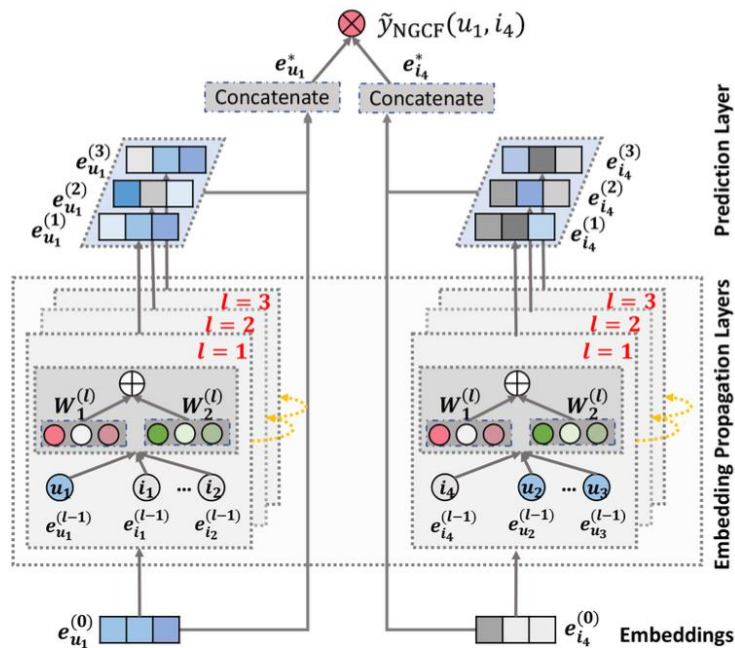


□ Architecture



1. Embedding Layer
2. Embedding Propagation Layer
3. Prediction Layer

□ Architecture



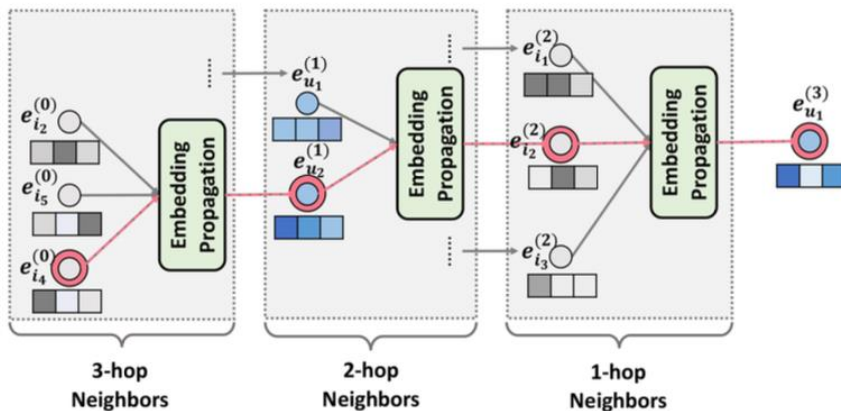
1. Embedding Layer

- 유저와 아이템간 상호작용이 반영되지 않은 유저, 아이템 각각의 Embedding

$$\mathbf{E} = [\underbrace{\mathbf{e}_{u_1}, \dots, \mathbf{e}_{u_N}}_{\text{users embeddings}}, \underbrace{\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_M}}_{\text{item embeddings}}]$$

$\mathbf{e}_u \in \mathbb{R}^d$ ($\mathbf{e}_i \in \mathbb{R}^d$) where d denotes the embedding size

□ Architecture



2. Embedding Propagation Layer

$$\mathbf{e}_u^{(k+1)} = \sigma \left(\mathbf{W}_1 \mathbf{e}_u^{(k)} + \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u| |\mathcal{N}_i|}} (\mathbf{W}_1 \mathbf{e}_i^{(k)} + \mathbf{W}_2 (\mathbf{e}_i^{(k)} \odot \mathbf{e}_u^{(k)})) \right),$$

$$\mathbf{e}_i^{(k+1)} = \sigma \left(\mathbf{W}_1 \mathbf{e}_i^{(k)} + \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u| |\mathcal{N}_i|}} (\mathbf{W}_1 \mathbf{e}_u^{(k)} + \mathbf{W}_2 (\mathbf{e}_u^{(k)} \odot \mathbf{e}_i^{(k)})) \right),$$

- $\mathbf{W}_1, \mathbf{W}_2$: feature transformation matrix
- $\sigma(\cdot)$: nonlinear activation function

□ Architecture

3. Prediction

■ Embedding propagation output : $\{e_u^{(1)}, \dots, e_u^{(L)}\}$

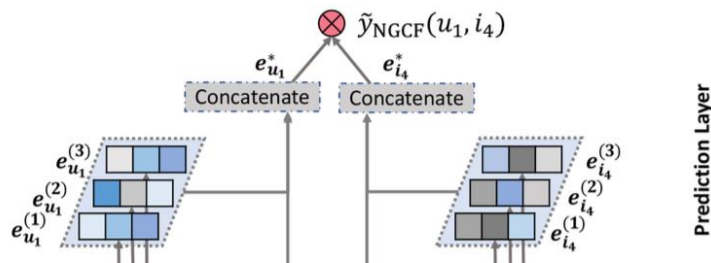
■ 각기 다른 연결에서 전달 받은 메시지를 강조하므로, 유저 선호에 대해 각기 다른 부분을 반영

□ Concatenation 을 통해 최종 유저/아이템에 대한 임베딩 구성

$$\mathbf{e}_u^* = \mathbf{e}_u^{(0)} \parallel \dots \parallel \mathbf{e}_u^{(L)}, \quad \mathbf{e}_i^* = \mathbf{e}_i^{(0)} \parallel \dots \parallel \mathbf{e}_i^{(L)},$$

■ 파라미터가 없어 연산이 단순

■ Prediction Layer output : $\hat{y}_{\text{NGCF}}(u, i) = \mathbf{e}_u^{*T} \mathbf{e}_i^*.$



Thank You!



HTET ARKAR (hak3601@cau.ac.kr)