Graph Convolutional Matrix Completion

2022.4.25

박진혁

Data Mining
Quality Analytics

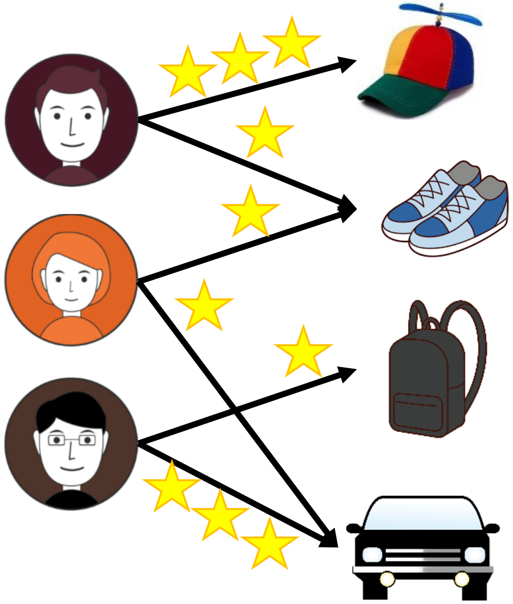
Content

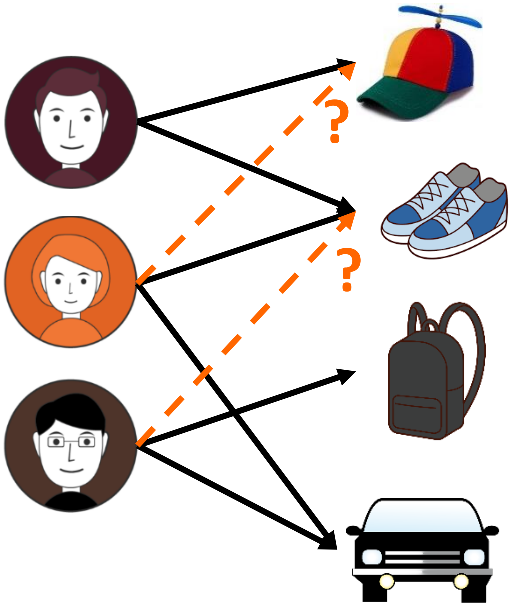
Background

❖ What is a recommender system?

➢ 사용자의 선호도 및 과거 행동을 바탕으로 사용자에 맞는 관심사를 제공하는 분야

1. Content-based Filtering
2. Collaborative Filtering
3. Hybrid Filtering





Introduction

❖ 2022년 4월 25일 기준 680회 인용

❖ Matrix Factorization에 graph구조를 투영시킨 방법론

Graph Convolutional Matrix Completion

Rianne van den Berg
University of Amsterdam

Thomas N. Kipf
University of Amsterdam

Max Welling
University of Amsterdam, CIFAR[1]

Abstract

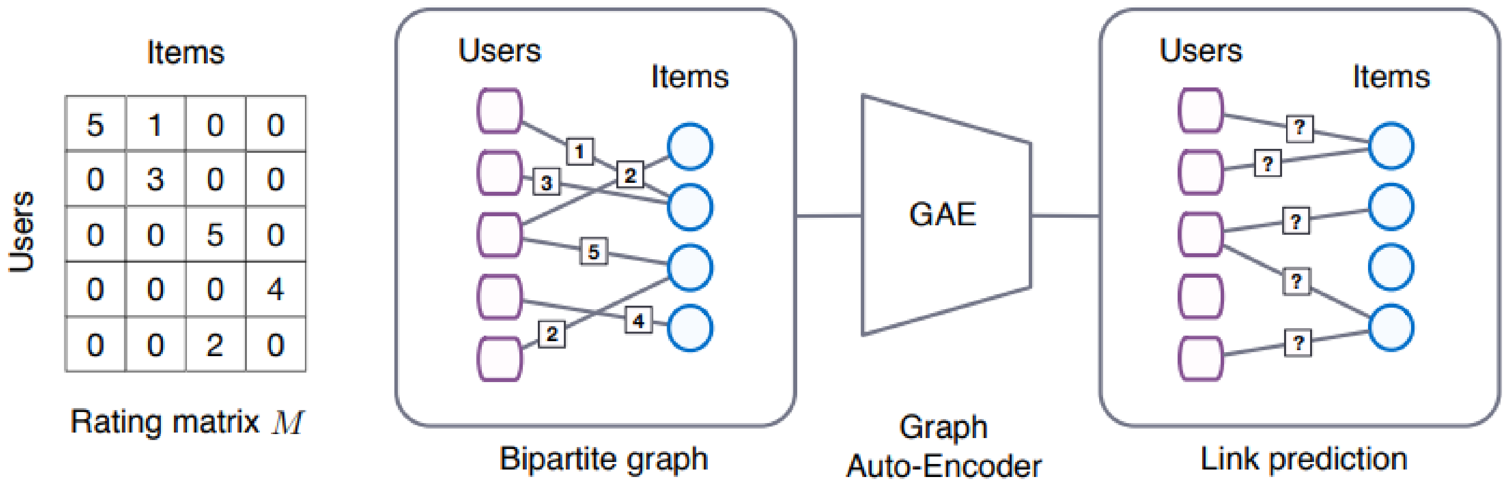
We consider matrix completion for recommender systems from the point of view of link prediction on graphs. Interaction data such as movie ratings can be represented by a bipartite user-item graph with labeled edges denoting observed ratings. Building on recent progress in deep learning on graph-structured data, we propose a graph auto-encoder framework based on differentiable message passing on the bipartite interaction graph. Our model shows competitive performance on standard collaborative filtering benchmarks. In settings where complimentary feature information or structured data such as a social network is available, our framework outperforms recent state-of-the-art methods.

Introduction

❖ Matrix Completion문제를 link prediction on graph관점으로 접근

❖ Matrix Completion를 해결하기 위한 Graph Convolutional Matrix Completion(GCMC)제안

❖ GCMC는 auto-encoder형태로 구성



<Overall architecture>

Introduction

❖ What is a matrix completion?

- 행렬에서 값이 없는 항목들의 값을 채우는 문제
- R: 비어있는 곳이 없는 원래 데이터
- $\hat{R}$: 원래 데이터로부터 비어있는 곳을 복구한 데이터
- r_{ui}: 사용자u가 아이템i를 얼마나 좋아하는를 나타내는 지표
- *: 아직 사용자가 평가하지 않은 데이터

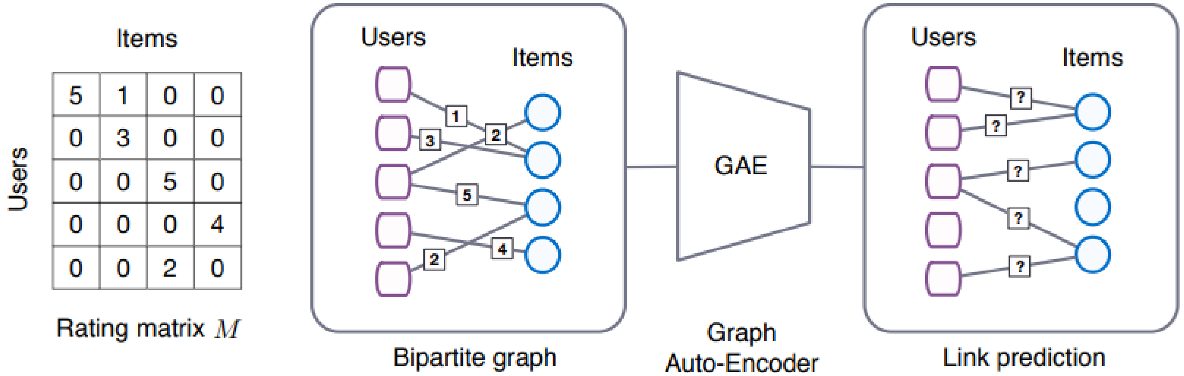
$$
\begin{array}{l}
\begin{array}{cccccccc} movie.1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\
\left\{\begin{array}{l} user1 \\ user2 \\ user3 \\ user4 \\ user5 \\ user6 \\ user7 \\ user8 \end{array}\right.
\begin{bmatrix}
3 & 5 & * & 4 & 1 & * & * & 2 \\
* & 3 & 5 & 1 & 2 & * & * & 3 \\
4 & 1 & * & 4 & 1 & * & 3 & 2 \\
5 & 2 & * & * & 2 & 3 & * & * \\
* & 2 & 4 & 2 & * & * & 1 & 2 \\
5 & * & * & 5 & 4 & * & * & 4 \\
1 & * & 5 & 2 & 3 & 1 & 5 & 3 \\
* & 3 & 2 & 1 & 4 & * & * & *
\end{bmatrix}
\end{array}
$$

$$\min_{\hat{R}} \left\| \hat{R} - \mathrm{R} \right\|_F^2$$

Proposed method

❖ Matrix completion as link prediction in bipartite graphs

- Matrix M은 $N_u * N_v$
- Matrix에서 0은 관측되지 않은 값
  - 목표: 관측되지 않은 값 예측
- w: User/item 집합 ,ε: Edge 집합 ,R: Rating 집합



$$G = (w, \varepsilon, R)$$
$$w = u \cup v,$$
$$(u_i, r, v_i) \in \varepsilon,$$
$$r \in [1, \ldots, R] = R$$

Data Mining Quality Analytics

Proposed method

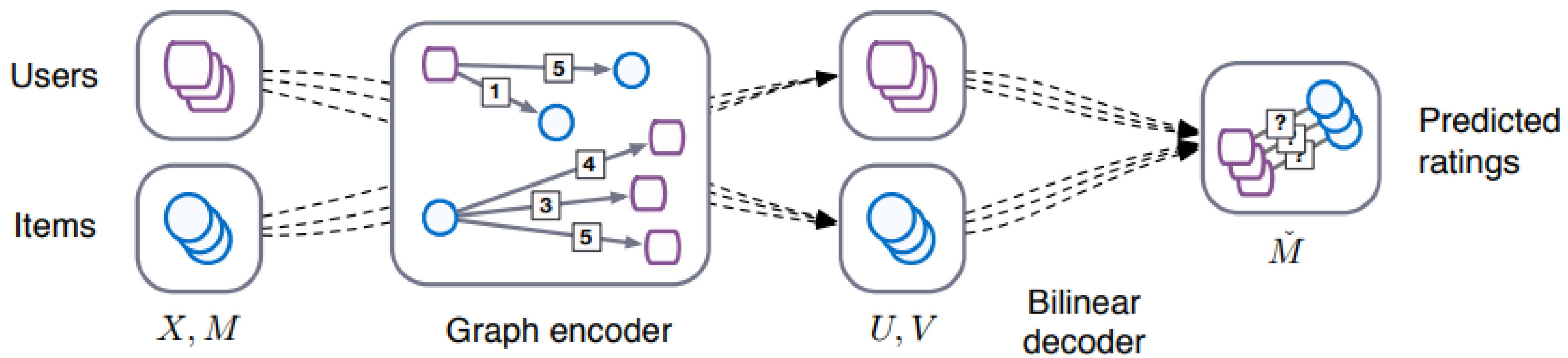
- Graph auto-encoder
  - Graph Encoder와 Pairwise Decoder로 구성
  - Graph Encoder
    - $Z = f(X, A), Z = f(X, M_1, \dots, M_R)$
  - Pairwise Decoder
    - $\hat{A} = g(Z)$, node embedding(z_i, z_j) 을 인풋으로 받아$\hat{A}_{ij}$를 예측
  - Z: node embedding matrix(N, E), X: feature matrix(N, D), A: graph adjacency matrix(N, N)
  - Decoder $g(U, V)$ → $\widehat{\mathrm{M}}$
  - 예측된 $\widehat{\mathrm{M}}$와 M 사이의 reconstruction error를 줄이는 것이 목표

Proposed method

❖ Graph convolutional encoder

➢ Weight sharing: Graph에서 location에 대한 효율적인 특징을 나타냄

➢ Processing channel: 평점 종류에 대한 처리과정 존재



<Overall architecture>

Proposed method

❖ Graph convolutional encoder

- Local graph convolution(Message passing)
  - $M_{j\rightarrow i,r} = \frac{1}{c_{ij}} W_r x_j$, C_{ij}: normalization constant
  - $|N_i|$: left normalization, $\sqrt{|N_i||N_j|}$: symmetric normalization
- Graph convolution layer
  - $h_i = \sigma[accum(\sum_{j\in N_{i,1}} \mu_{j\rightarrow i,1}, \ldots, \sum_{j\in N_{i,R}} \mu_{j\rightarrow i,R})]$
  - $u_i = \sigma(Wh_i)$
- Processing channel: 평점 종류에 대한 처리과정 존재

Proposed method

❖ Bilinear decoder

➢ User의 임베딩 벡터와 item의 임베딩 벡터를 내적 계산하는 방식

➢ $p(\hat{M}_{ij}) = \frac{e^{u_i^T Q_r v_j}}{\sum_{s \in R} e^{u_i^T Q_r v_j}}$, Softmax 함수에 의해 bilinear을 통해 확률 분포 생성

➢ $\hat{M}_{ij} = g(u_i, v_j) = \boldsymbol{E}_{p(A=\pi r^2 \hat{M}_{ij=r}}[r] = \sum_{r \in R} r * p(\hat{M}_{ij})$, 평점에 따라 확률이 가중 평균

Proposed method

❖ Model training

- Loss function
  - $L = \sum_{i,j;\Omega_{i,j}=1} \sum_{r=1}^{R} I\left[r = M_{ij}\right] logp(\hat{M}_{ij} = r)$
- Node drop
  - 특정 node에 대해 $p_{dropout}$확률로 밖으로 나가는 모든 message를 dropout하는 방식
- Mini-batching
  - User-item pair 총합에서 고정된 수의 contribution만을 추출하여 학습에 사용
  - 현재 batch에서 존재하지 않는 각 평점 class의 user/item 행을 제거할 수 있음

Proposed method

❖ Vectorized implementation

$$\begin{bmatrix} U \\ V \end{bmatrix} = f(X, M_1, \dots, M_R) = \sigma(\begin{bmatrix} H_u \\ H_v \end{bmatrix} W^T$$

$$with \begin{bmatrix} H_u \\ H_v \end{bmatrix} = \sigma\left(\sum_{r=1}^{R} D^{-1}\,\mathcal{M}_r X \mathrm{W_r^T}\right),$$

$$and \mathcal{M}_r = \begin{bmatrix} 0 & M_r \\ M_r^T & 0 \end{bmatrix}$$

행렬	역할	Shape
$[\mathrm{UV}]$	Node embedding matrix	$\mathrm{N+M, E}$
$[H_u H_v]$	Hidden matrix	$\mathrm{N+M, E}$
W^T	Weight matrix	$\mathrm{E, E}$
D^{-1}	Normalization matrix	$\mathrm{N+M, N+M}$
A	Graph adjacency matrix	$\mathrm{N+M, N+M}$
X	Feature matrix	$\mathrm{N+M, D}$
$\mathrm{W_r^T}$	Edge-type specific weight matrix	$\mathrm{D, E}$

Proposed method

- Input feature representation and side information
  - Side information
    - Graph convolutional layer의 병목현상을 해결하기 위한 방법
    - $u_i = \sigma\left(Wh_i + W_2^f f_i\right)$, W_2^f: weight 행렬,
    - $f_i = \sigma\left(W_1^f x_i^f + b\right)$, W_1^f: weight 행렬, x_i^f: user/item feature vector

- Weight sharing
  - Ordinal weight sharing
    - $W_r = \sum_{s=1}^{r} T_s$
  - Weight sharing(선형 결합 형태)
    - $Q_r = \sum_{s=1}^{nb=1} a_{rs} P_s$

Experiments

❖ Dataset&Result

- ➢ 총 6개의 데이터 셋을 가지고 실험 진행
- ➢ MovieLens데이터는 데이터 크기에 따라 총 3가지로 구서
- ➢ Fixster, Douban, YahooMusic은 user와 item의 정보를 가지고 있음

Dataset	Users	Items	Features	Ratings	Density	Rating levels
Flixster	3,000	3,000	Users/Items	26,173	0.0029	0.5, 1, . . . , 5
Douban	3,000	3,000	Users	136,891	0.0152	1, 2, . . . , 5
YahooMusic	3,000	3,000	Items	5,335	0.0006	1, 2, . . . , 100
MovieLens 100K (ML-100K)	943	1,682	Users/Items	100,000	0.0630	1, 2, . . . , 5
MovieLens 1M (ML-1M)	6,040	3,706	—	1,000,209	0.0447	1, 2, . . . , 5
MovieLens 10M (ML-10M)	69,878	10,677	—	10,000,054	0.0134	0.5, 1, . . . , 5

Experiments

❖ Dataset&Result

- ➢ 모든 결과값은 RMSE score로 계산
- ➢ 모든 데이터 셋에서 해당 알고리즘이 가장 좋은 성능을 나타냄

Model	ML-100K + Feat
MC [3]	0.973
IMC [11, 31]	1.653
GMC [12]	0.996
GRALS [25]	0.945
sRGCNN [22]	0.929
GC-MC (Ours)	0.910
GC-MC+Feat	**0.905**

Model	Flixster	Douban	YahooMusic
GRALS	1.313/1.245	0.833	38.0
sRGCNN	1.179/0.926	0.801	22.4
GC-MC	**0.941/0.917**	**0.734**	**20.5**

Model	ML-1M	ML-10M
PMF [20]	0.883	–
I-RBM [26]	0.854	0.825
BiasMF [16]	0.845	0.803
NNMF [7]	0.843	–
LLORMA-Local [17]	0.833	0.782
I-AUTOREC [27]	0.831	0.782
CF-NADE [32]	**0.829**	**0.771**
GC-MC (Ours)	0.832	0.777

Conclusion

- ❖ Encoder decoder형태를 가지고 새로운 평점을 labeled edge 형태로 예측
- ❖ 본 논문에서 제안한 모델은 여러 benchmark모델의 성능을 능가함
- ❖ Matrix factorization의 적용에 있어서 한계가 존재함
- ❖ Matrix factorization의 추가적인 개선이 있다면 더 나은 추천 시스템이 가능

Data Mining Quality Analytics

감사합니다