

지식 그래프를 활용한 추천 서비스

방 학 5 주 차

이번 주차에서 리뷰할 논문

Knowledge Graph Convolutional Networks for Recommender Systems
(2019, Wang et al.)

인터넷 기술의 발전

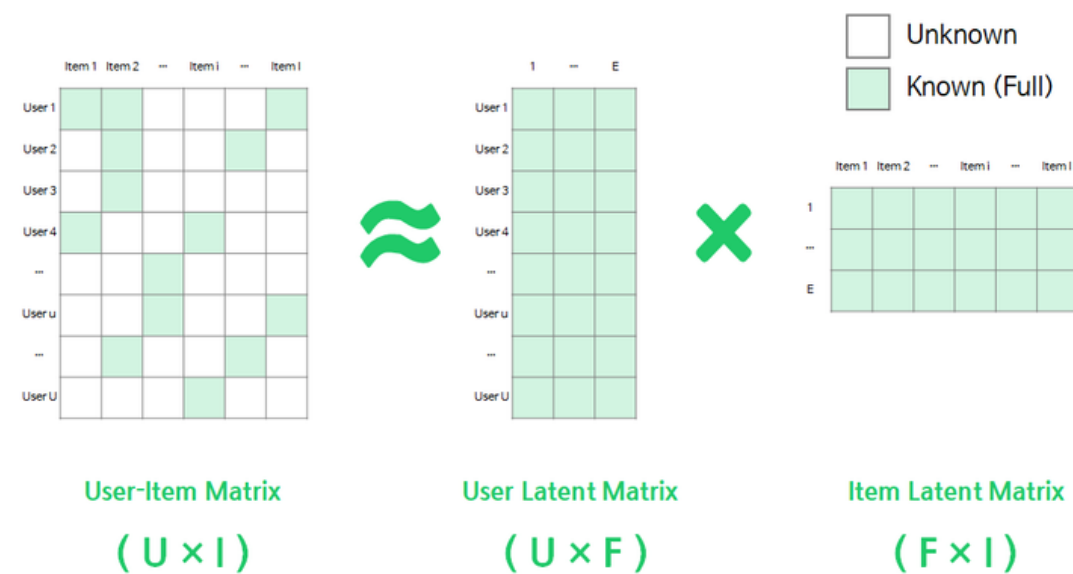
방대한 양의 정보가 장점이자 단점이 됨

**정보 과부화의 영향을 완화하기 위해 사용자의 개인화된 관심사에 맞는 소수의 항목을 검색하고
추천하는 추천 시스템이 제안됨**

전통적인 추천 기법

협업 필터링을 이용한 추천

사용자와 항목에 ID 기반 representation vector를 할당하고 inner product나 신경망 같은 연산을 통해 상호작용을 모델링함



협업 필터링이 가진 단점

Sparsity 문제와 Cold Start 문제

Sparsity Problem : 데이터가 sparse 하면 추천에 참고할 근거가 부족해서 정확도가 떨어짐

Cold Start Problem : 한번도 평점을 남겨 본 적이 없으면 추천이 불가능함

추가 연구 : Knowledge Graph 개념의 적용

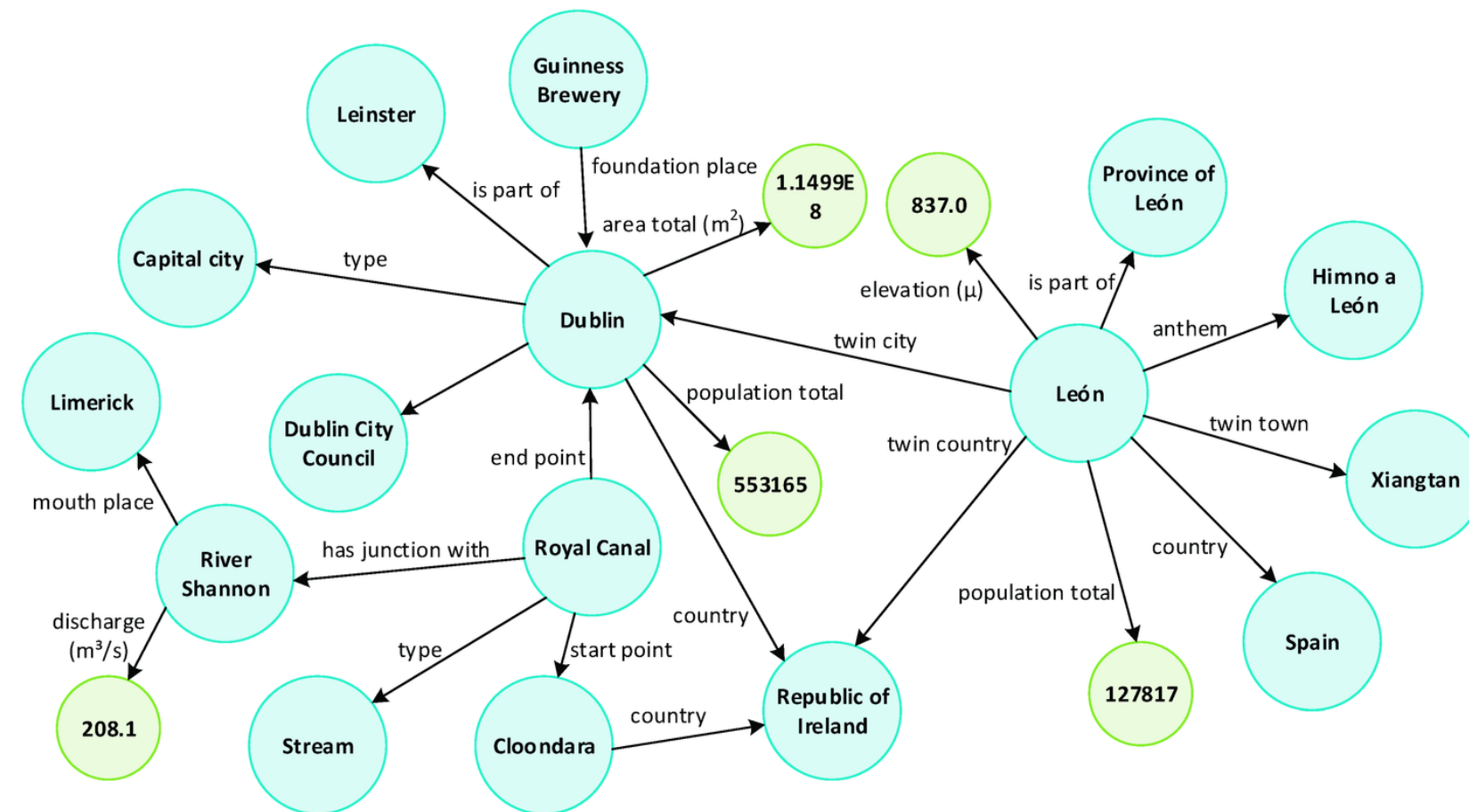
속성이 서로 연결되어 지식 그래프를 형성함

지식 그래프 기반의 추천 시스템이 갖는 장점

- (1) 항목 간 충분한 의미적 연관성이 잠재적 연관성을 탐색하고 결과의 정확도를 높이는데 도움이 됨
- (2) 지식 그래프의 다양한 관계 유형은 관심사의 확장과 추천의 다양성을 높이는데 도움이 됨
- (3) 사용자의 과거 이력과 추천 항목을 연결하여 추천 시스템에 설명력을 부여함

지식 그래프 기반의 추천 시스템의 한계

높은 차원과 이질성



연구자들이 한계를 극복한 방법

KGE 방식의 전처리

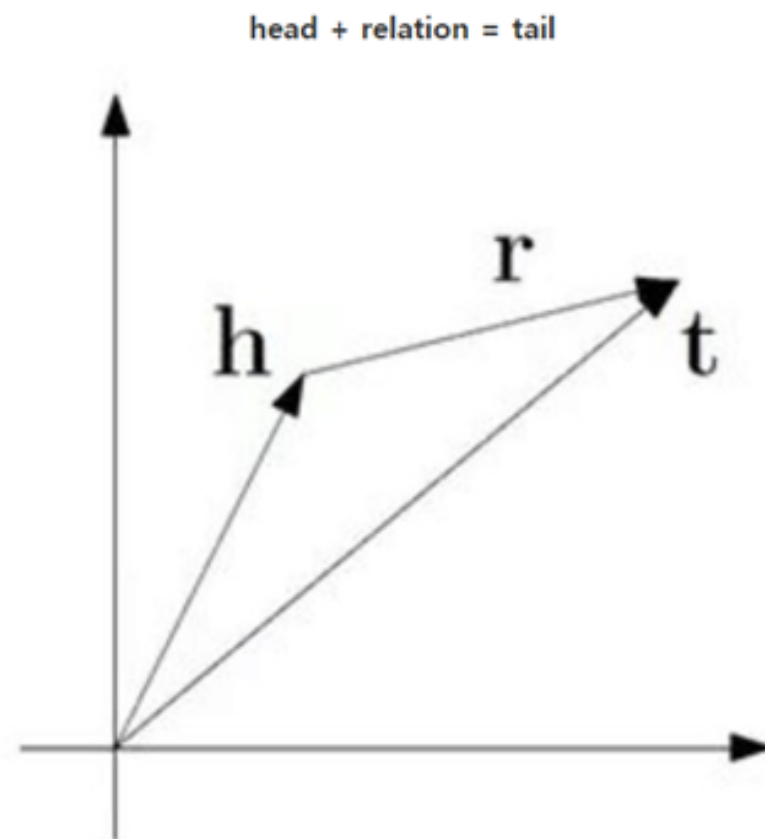
엔티티와 관계를 저차원 Representation Vector에 매핑하기

KGE 방식의 전처리가 갖는 한계

- (1) 일반적인 KGE 방법론은 엄격한 의미적 연관성을 모델링하는데 중점을 두게 됨
- (2) 추천 시스템보다는 지식 그래프의 완성이나 링크 예측에 더 적합함

KGE 방식의 전처리가 갖는 한계

예시 1) TransE



$$f_r(h, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2^2$$

KGE 방식의 전처리가 갖는 한계

예시 2) TransR

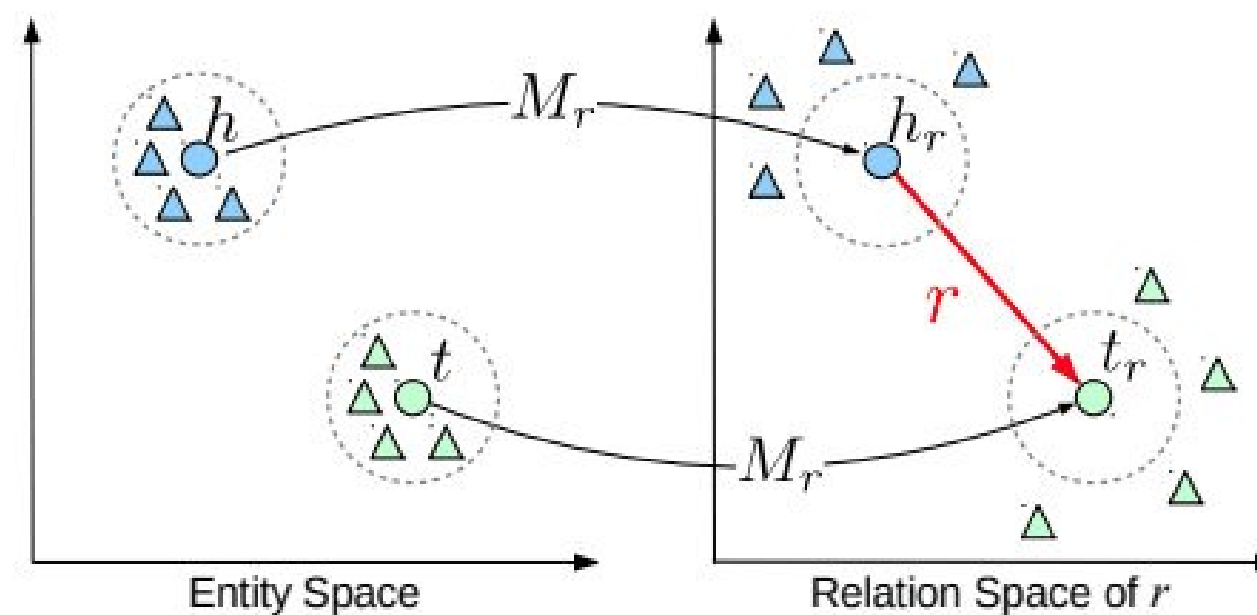


Figure 1: Simple illustration of TransR.

연구자들이 한계를 극복한 방법 2

지식 그래프 구조를 활용하기 위해 그래프 알고리즘을 직접 설계하기

모델 1) PER (Personalized Entity Recommendation)

정보 네트워크 안에 존재하는 다양한 관계를 각 사용자에게 다르게 조합하여 추천 품질을 향상함

사용자와 아이템 사이의 연결성을 다양한 유형의 경로로 표현하고, 전역적/개인화된 추천 모델을 정의한 후 베이지안 순위 최적화 기법을 사용하여 제안된 모델을 추정함

연구자들이 한계를 극복한 방법 2

지식 그래프 구조를 활용하기 위해 그래프 알고리즘을 직접 설계하기

모델 2) FMG (FM with Group lasso)

메타그래프의 도입과 MF-FM 결합을 통해 엔티티 추천 문제를 해결함

다양한 관계를 메타그래프로 표현하고, 메타그래프 간 유사도를 통해 사용자와 아이템 간의 잠재적 특성을 학습함. 메타그래프 기반 특성들을 Group lasso를 통해 정규화된 FM으로 선택하여 최종적인 추천 점수를 예측함

연구자들이 한계를 극복한 방법 2

지식 그래프 구조를 활용하기 위해 그래프 알고리즘을 직접 설계하기

모델 3) RippleNet

Preference Propagation을 통해 선호도를 파악함

임베딩 기반 방법과 경로 기반 방법을 결합하여 자동적이면서 반복적으로 사용자의 계층적 잠재 관심사를 발견하고, 선호도를 확장함

이 논문의 목적

지식 그래프 기반 추천 시스템의 문제점을 찾기

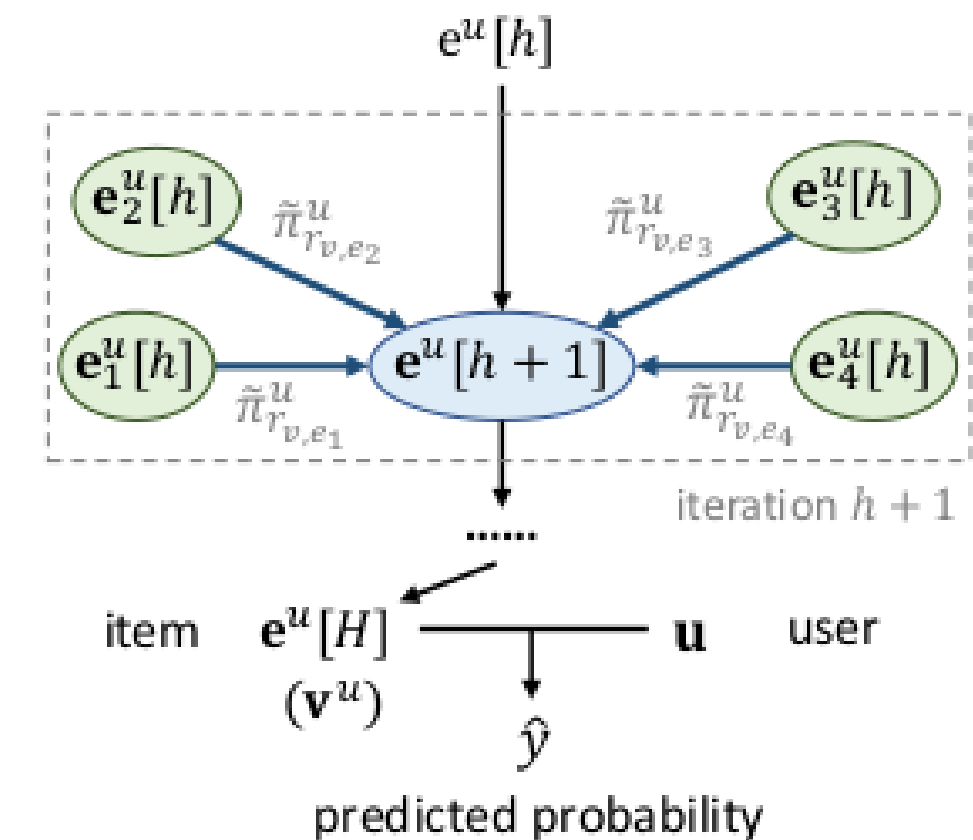
지식 그래프에서 고차 구조와 의미 정보를 모두 자동으로 가져올 수 있도록 설계하는 것이 목표임

Introduction

KGCN

Graph Convolution Network에서 영감을 받음

엔티티 간 고차원의 구조적인 근접성을 포착하기 위해 제안됨



(b)

KCGN의 핵심 아이디어

엔티티의 표현을 계산할 때 이웃 인포메이션을 집계하고 통합함

장점

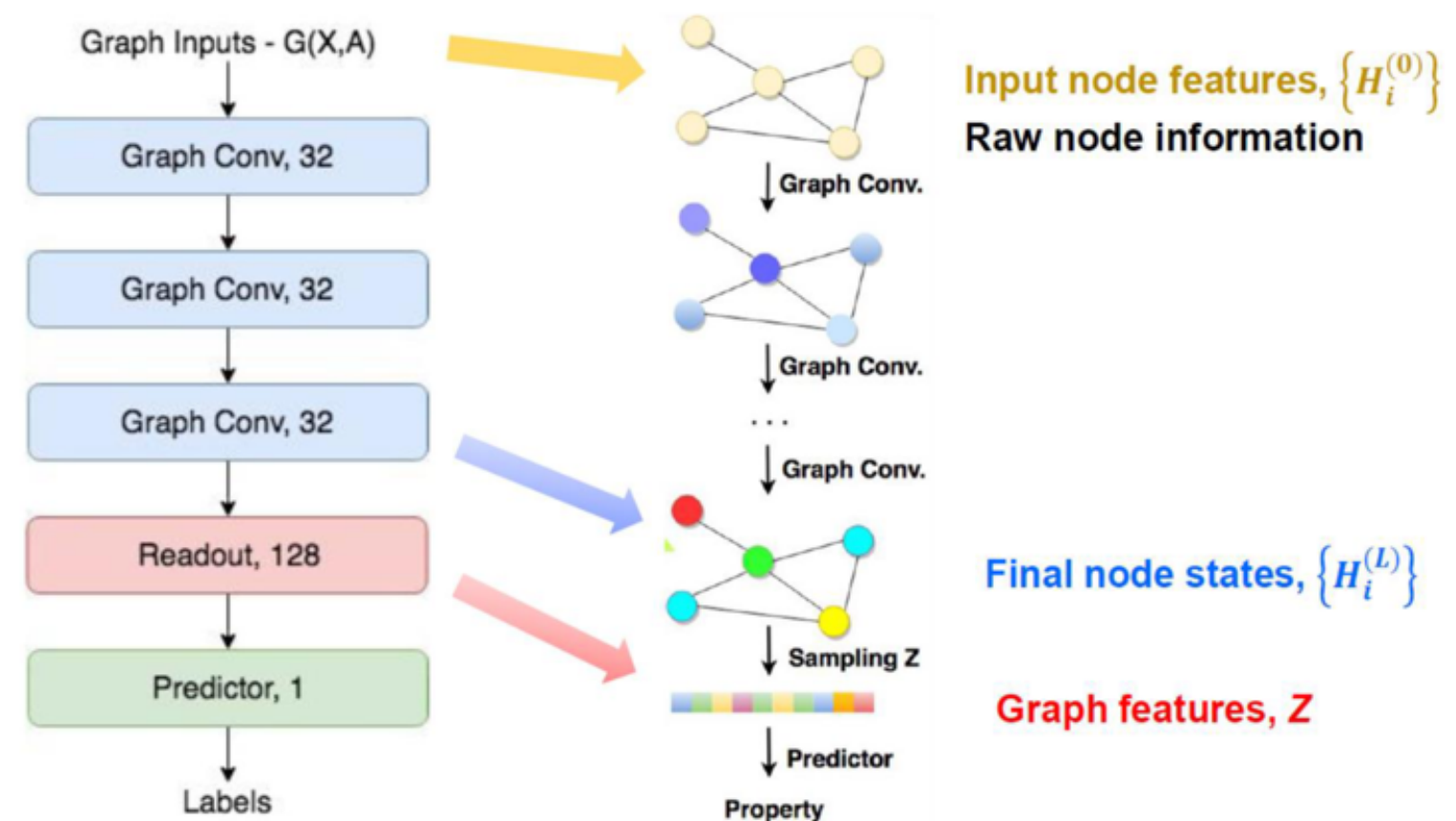
- (1) 이웃을 집계하면서 local 근접 구조가 성공적으로 capture되어 각 엔티티에 정의됨
- (2) 이웃이 관계와 다른 사용자를 기반으로 점수가 부여되면서 의미 정보와 개인 선호도를 모두 특징지을 수 있음

GCN이란

GNN 중 하나로 CNN의 특성 중 일부를 그래프에 적용하는 것을 목표로 함

CNN에서 따온 것

- (1) local feature를 학습
- (2) weight sharing를 적용



GCN의 분류

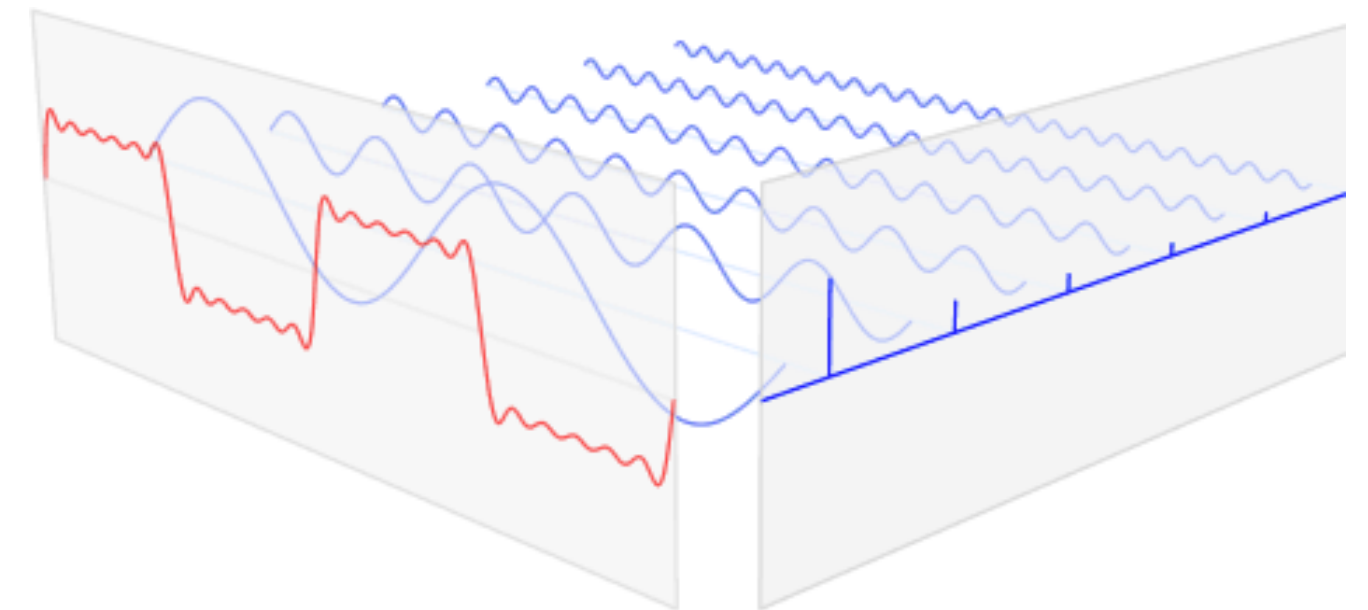
스펙트럼 방식과 비스펙트럼 방식

1) 스펙트럼 방식

특정한 신호를 여러 단순한 요소들의 합으로 분해하는 방식

대표적으로 푸리에 변환을 사용해서 분해함.

(그래프의 Laplacian matrix를 eigen-decomposition 함)



GCN의 분류

스펙트럼 방식과 비스펙트럼 방식

2) 비스펙트럼 방식

원본 그래프에서 직접 작동하며, 노드의 공간적인 관계에 기반해서 convolution을 진행함

$$h_v = f(\Theta^T(\sum_{u \in N(v) \cup v} \bar{A}_{vu} X_u)), \forall v \in V$$

$$H^{(k)} = f(\sum_{i=1}^{k-1} \bar{A} H^{(i-1)} \Theta^{(k)})$$

Predict Function

유저 u 가 아이템 v 에 가진 잠재적 관심을 측정하기

상호작용 행렬 Y 와 지식 그래프 G 가 있을 때,

$$\mathcal{F}(u, v | \Theta, Y, \mathcal{G})$$

KGCN Layer 1

사용자와 관계 간의 점수를 계산하기

점수는 **personalized filter**가 됨

p_i 가 사용자 u 에게 r 관계가 중요한 정도, g 는 $R \times R \rightarrow R$ 인 함수일 때

$$\pi_r^u = g(u, r)$$

KGCN Layer 2

선형 결합으로 위상적 근접 구조를 특성화하기

위상적 근접 구조 : 집합 간의 근접성을 나타내는 것

엔티티 e , 엔티티 간 관계 $r_{v,e}$, v 와 붙어있는 entity 집합인 $N(v)$ 가 있을 때

$$\mathbf{v}_{N(v)}^u = \sum_{e \in N(v)} \tilde{\pi}_{r_{v,e}}^u \mathbf{e},$$

$$\tilde{\pi}_{r_{v,e}}^u = \frac{\exp(\pi_{r_{v,e}}^u)}{\sum_{e \in N(v)} \exp(\pi_{r_{v,e}}^u)}$$

(normalized된 점수)

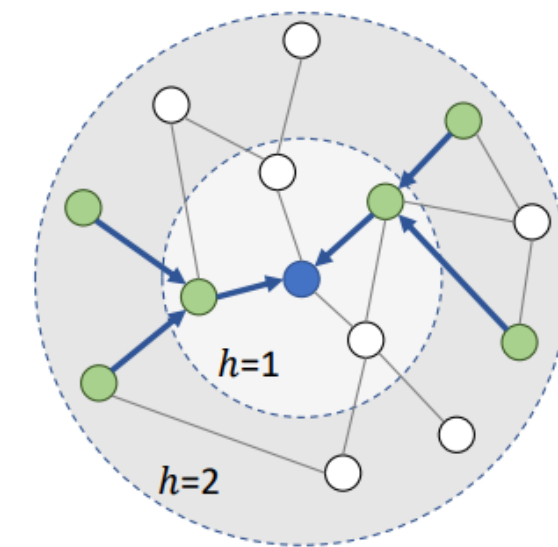
KGCN Layer 3

Receptive Field (엔티티 e 에 대한 neighborhood 표현)

실제 지식 그래프에서 크기는 엔티티에 따라 달라질 수 있으므로 균일하게 샘플링함

$N(v)$ 와 서로소인 집합 $S(v)$ 가 있을 때

$$v^u_{S(v)}$$



KGCN Layer 4

Aggregator

entity 표현 v 와 neighborhood 표현 $v_{S(v)}^u$ 을 하나의 벡터로 만들 때 사용함

Aggregator 1 : Sum aggregator

$$agg_{sum} = \sigma \left(\mathbf{W} \cdot (\mathbf{v} + \mathbf{v}_{S(v)}^u) + \mathbf{b} \right)$$

KGCN Layer 4

Aggregator

entity 표현 \mathbf{v} 와 neighborhood 표현 $\mathbf{v}_{S(v)}^u$ 을 하나의 벡터로 만들 때 사용함

Aggregator 2 : Concat aggregator

$$agg_{concat} = \sigma \left(\mathbf{W} \cdot concat(\mathbf{v}, \mathbf{v}_{S(v)}^u) + \mathbf{b} \right)$$

KGCN Layer 4

Aggregator

entity 표현 v 와 neighborhood 표현 $v_{S(v)}^u$ 을 하나의 벡터로 만들 때 사용함

Aggregator 3 : Neighbor aggregator

$$agg_{neighbor} = \sigma \left(\mathbf{W} \cdot v_{S(v)}^u + \mathbf{b} \right)$$

1차 entity 표현

엔티티 자체와 바로 인접한 이웃에 의해 달라짐

각 엔티티의 초기 표현을 전파한 게 1차 엔티티 표현이 됨

이걸 또 전파하면 2차 엔티티 표현이 되고, 그 후로도 반복됨

Predict Function 계산하기

```
Algorithm 1: KGCN algorithm
Input: Interaction matrix  $Y$ ; knowledge graph  $\mathcal{G}(\mathcal{E}, \mathcal{R})$ ;
neighborhood sampling mapping  $\mathcal{S} : e \rightarrow 2^{\mathcal{E}}$ ; trainable
parameters:  $\{\mathbf{u}\}_{u \in \mathcal{U}}$ ,  $\{\mathbf{e}\}_{e \in \mathcal{E}}$ ,  $\{\mathbf{r}\}_{r \in \mathcal{R}}$ ,  $\{\mathbf{W}_i, \mathbf{b}_i\}_{i=1}^H$ ;
hyper-parameters:  $H, d, g(\cdot), f(\cdot), \sigma(\cdot), \text{agg}(\cdot)$ 
Output: Prediction function  $\mathcal{F}(u, v | \Theta, Y, \mathcal{G})$ 
1 while KGCN not converge do
2   for  $(u, v)$  in  $Y$  do
3      $\{\mathcal{M}[i]\}_{i=0}^H \leftarrow \text{Get-Receptive-Field}(v)$ ;
4      $\mathbf{e}^u[0] \leftarrow \mathbf{e}, \forall e \in \mathcal{M}[0]$ ;
5     for  $h = 1, \dots, H$  do
6       for  $e \in \mathcal{M}[h]$  do
7          $\mathbf{e}_{\mathcal{S}(e)}^u[h-1] \leftarrow \sum_{e' \in \mathcal{S}(e)} \tilde{\pi}_{r_{e,e'}}^u \mathbf{e}'^u[h-1]$ ;
8          $\mathbf{e}^u[h] \leftarrow \text{agg}(\mathbf{e}_{\mathcal{S}(e)}^u[h-1], \mathbf{e}^u[h-1])$ ;
9      $\mathbf{v}^u \leftarrow \mathbf{e}^u[H]$ ;
10    Calculate predicted probability  $\hat{y}_{uv} = f(\mathbf{u}, \mathbf{v}^u)$ ;
11    Update parameters by gradient descent;
12 return  $\mathcal{F}$ ;
13 Function Get-Receptive-Field( $v$ )
14    $\mathcal{M}[H] \leftarrow v$ ;
15   for  $h = H-1, \dots, 0$  do
16      $\mathcal{M}[h] \leftarrow \mathcal{M}[h+1]$ ;
17     for  $e \in \mathcal{M}[h+1]$  do
18        $\mathcal{M}[h] \leftarrow \mathcal{M}[h] \cup \mathcal{S}(e)$ ;
19   return  $\{\mathcal{M}[i]\}_{i=0}^H$ ;
```

$\hat{y}_{uv} = f(\mathbf{u}, \mathbf{v}^u)$

Loss Function

$$\mathcal{L} = \sum_{u \in \mathcal{U}} \left(\sum_{v: y_{uv}=1} \mathcal{J}(y_{uv}, \hat{y}_{uv}) - \sum_{i=1}^{T^u} \mathbb{E}_{v_i \sim P(v_i)} \mathcal{J}(y_{uv_i}, \hat{y}_{uv_i}) \right) + \lambda \|\mathcal{F}\|_2^2,$$

L : Cross-entropy loss

P : Negative Sampling distribution

T^u : user에 대한 negative sample 수 = 여기서는 상호작용을 한 sample 수

마지막 항 : L2 regularizer

방학 5주차

Conclusion

과제 1

EXPERIMENTS 읽고 정리하기

방학 5주차

Conclusion

과제 2

CONCLUSIONS AND FUTURE WORK 읽고 정리하기

가천대학교 이다

THANK YOU

KGCN

가천대학교 이다