

Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions

이은경

목차

1. Abstract
2. Introduction
3. BACKGROUND
4. Objectives
5. GNN basic

1. Abstract

1) 개요

1. Abstract



Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions

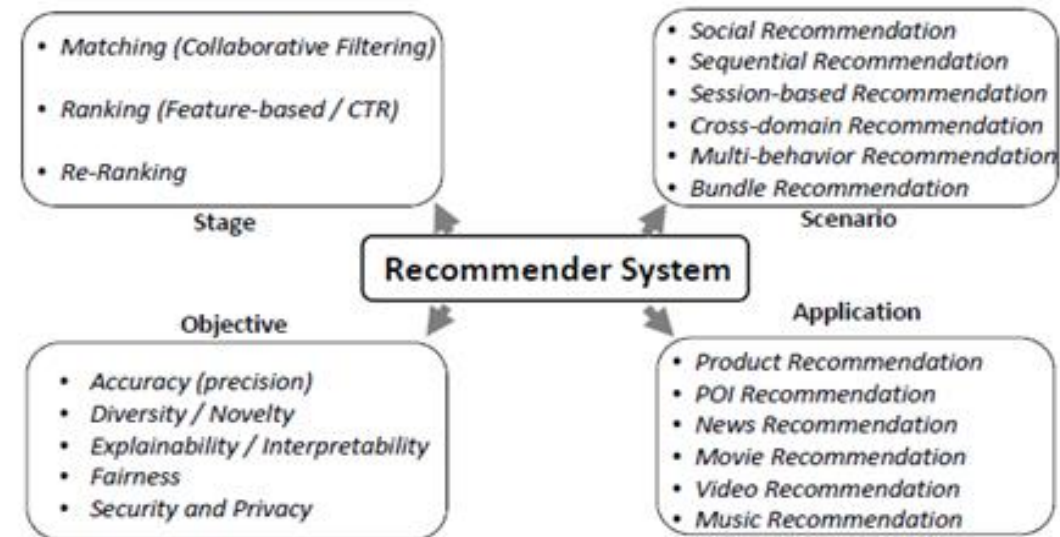
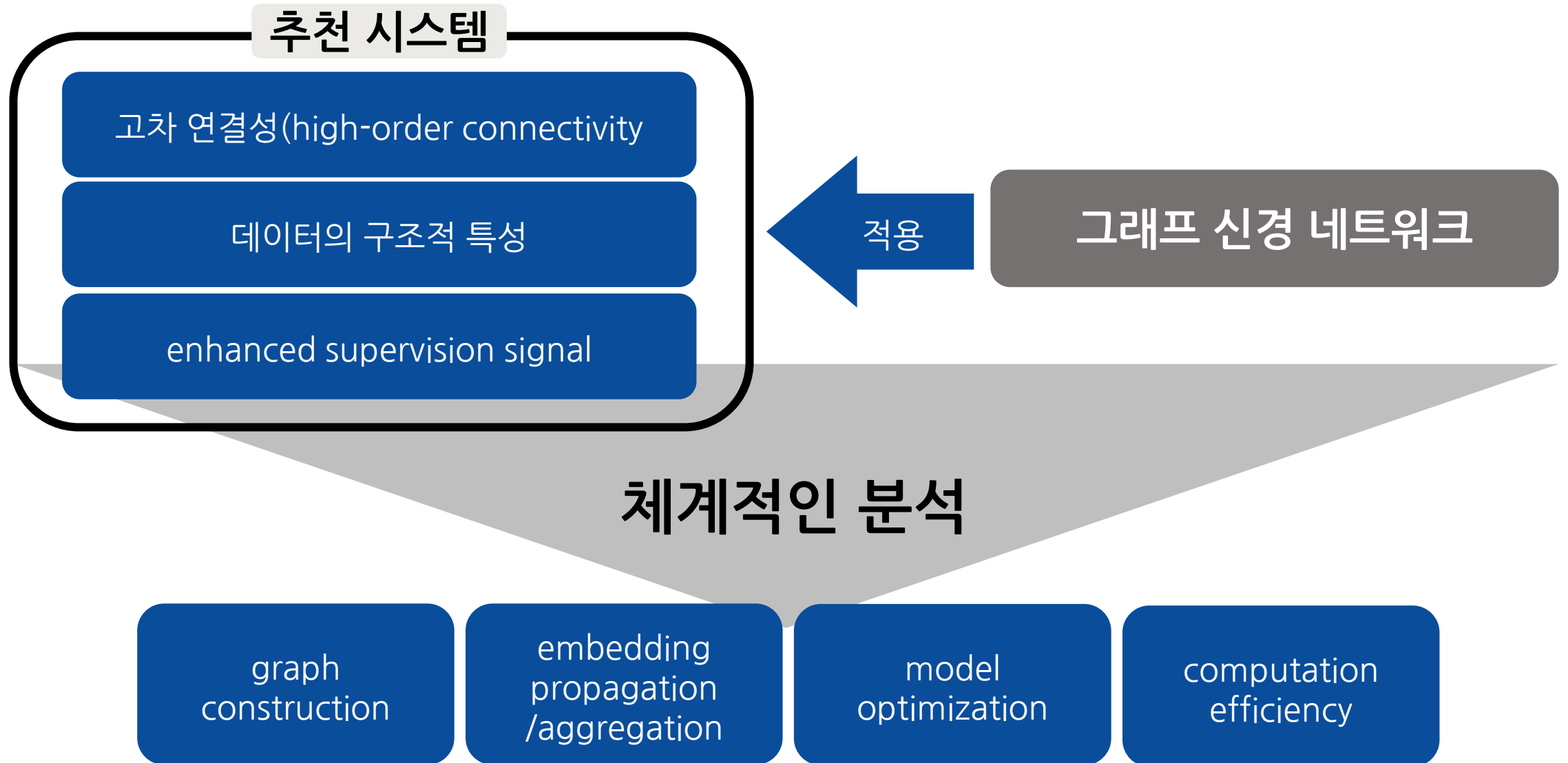


Fig. 1. An illustration of typical recommender systems (stages, scenarios, objectives, and applications)

1) 개요

1. Abstract



2. Introduction

1) 정의

Recommender Systems

2. Introduction



2) 역사

2. Introduction

Recommender Systems의 3단계

Shallow models

- 상호 작용의 similarity를 직접 계산해 collaborative filtering (CF) 효과 포착
- matrix factorization (MF)이나 factorization machine와 같은 모델 기반 CF 방법 제안
- 단점 : 복잡한 사용자 행동 또는 데이터 입력과 같은 문제



Neural models

- Neural collaborative filtering (NCF)은 MFP(다층 퍼셉트론)의 inner product를 확장하여 capacity를 개선(DeepFM)
- 단점 : 예측 및 훈련 패러다임이 관측된 데이터의 high-order structural information을 무시하기 때문에 이러한 방법은 여전히 제한적

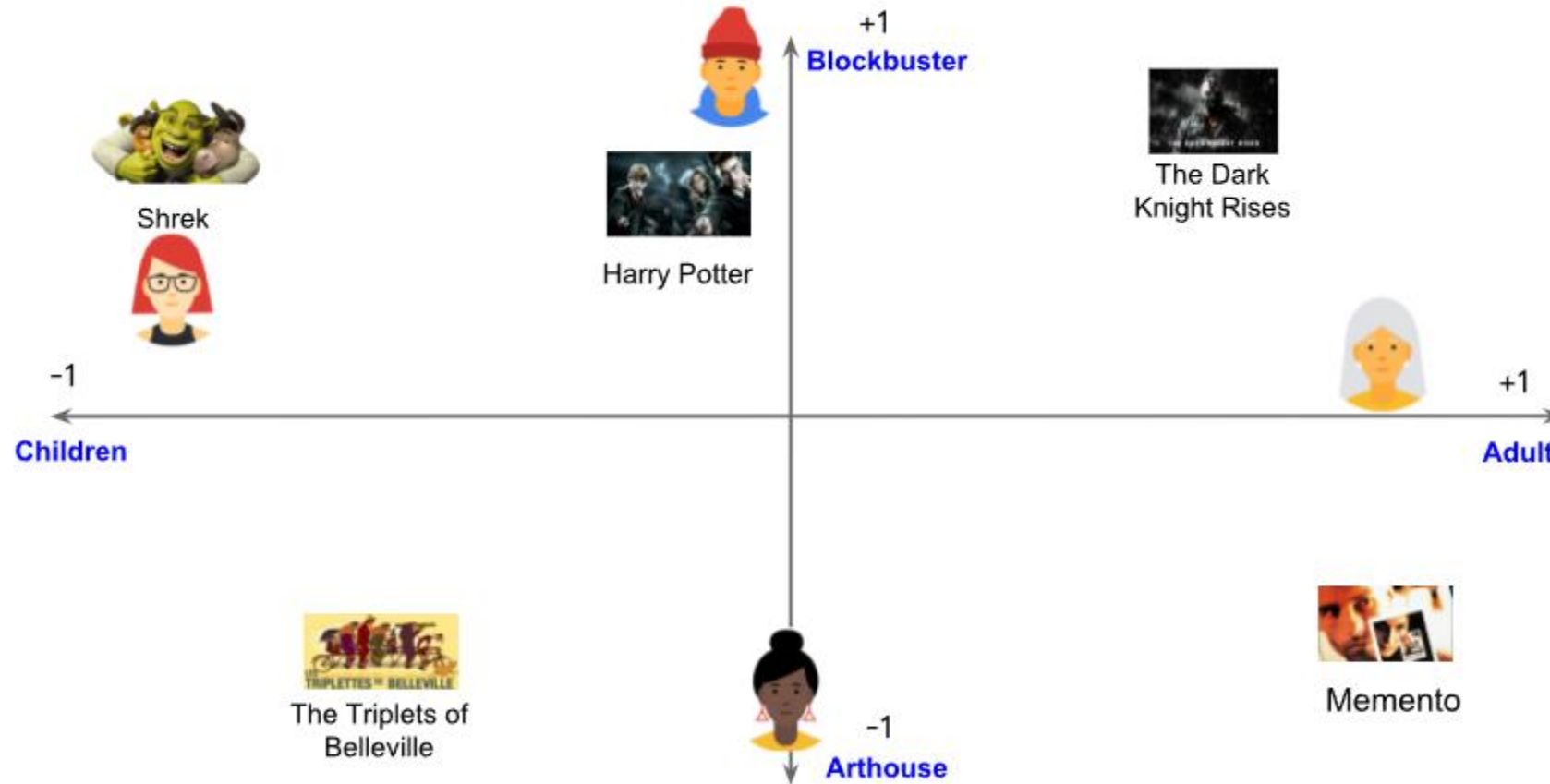


GNN based models

- 반복적으로 임베딩된 neighborhood를 집계하기 위해 neighborhood embedding를 채택하여 propagation layer를 쌓음
- 장점 : 각 노드는 기존 방법처럼 first-order neighbors만 액세스하는 것이 아니라 high-order neighbors information 접근 가능

2-1) Collaborative Filtering

3. Background



2-1) Collaborative Filtering

3. Background

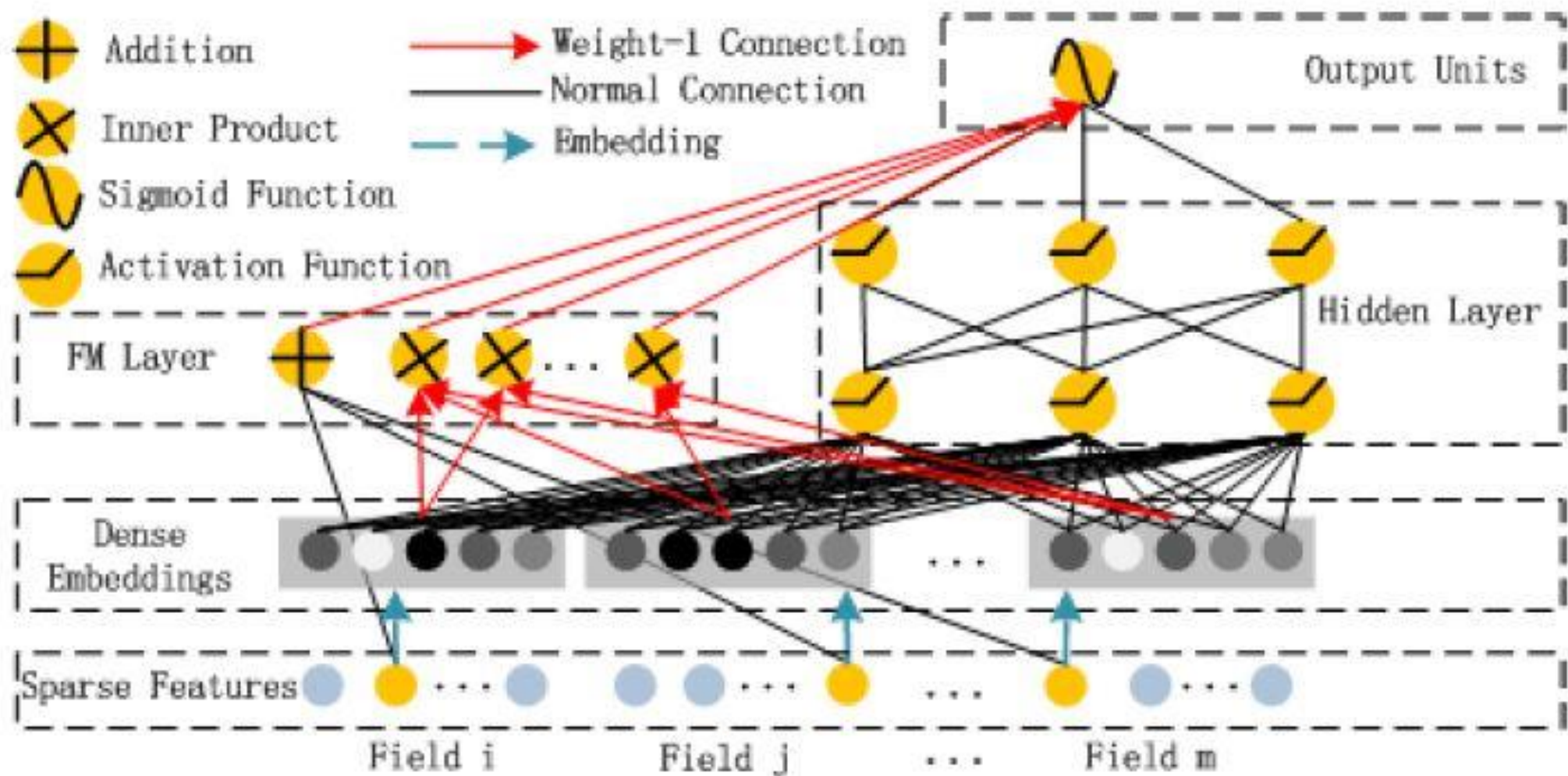
					
	Harry Potter	The Triplets of Belleville	Shrek	The Dark Knight Rises	Memento
	✓		✓	✓	
		✓			✓
	✓	✓	✓		
				✓	✓

 \approx

	.9	-1	1	1	-.9
	-.2	-.8	-1	.9	1
1	.1	.88	-1.08	0.9	1.09
-1	0	-0.9	1.0	-1.0	-1.0
.2	-1	0.38	0.6	1.2	-0.7
.1	1	-0.11	-0.9	-0.9	1.0
		-0.8	0.9	-1.0	-1.18
		0.91			

2-2) DeepFM

3. Background



3) Challenges

2. Introduction

입력은 노드와 엣지를 사용하여 그래프에 맞게 신중하고 적절하게 구성

특정 task을 위해, 그래프 신경망의 구성요소는 기존 연구가 다양한 선택 사항을 다른 장점과 단점으로 설명한 전파 및 집계 방법을 포함하여 adaptively하게 설계

최적화 목표, 손실 함수, 데이터 샘플링 등을 포함한 GNN 기반 모델의 최적화는 작업 요구 사항과 일치

계산 비용에 엄격한 제한이 있고 GNN의 inner propagation 연산이 여러 계산을 도입하기 때문에 추천 시스템에서 그래프 신경망의 효율적인 배치가 필요

3. Background

1) Overview

3. Background

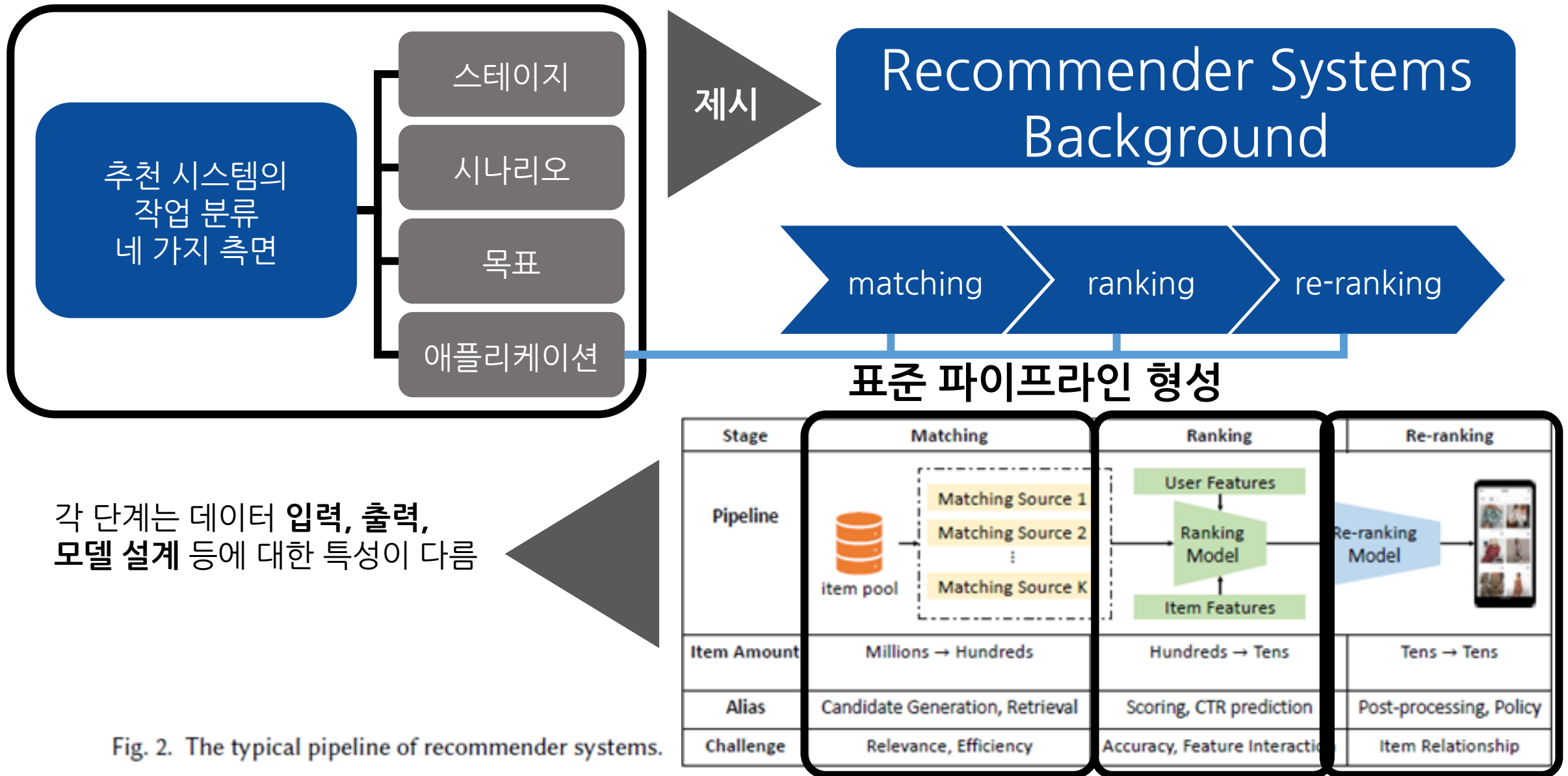
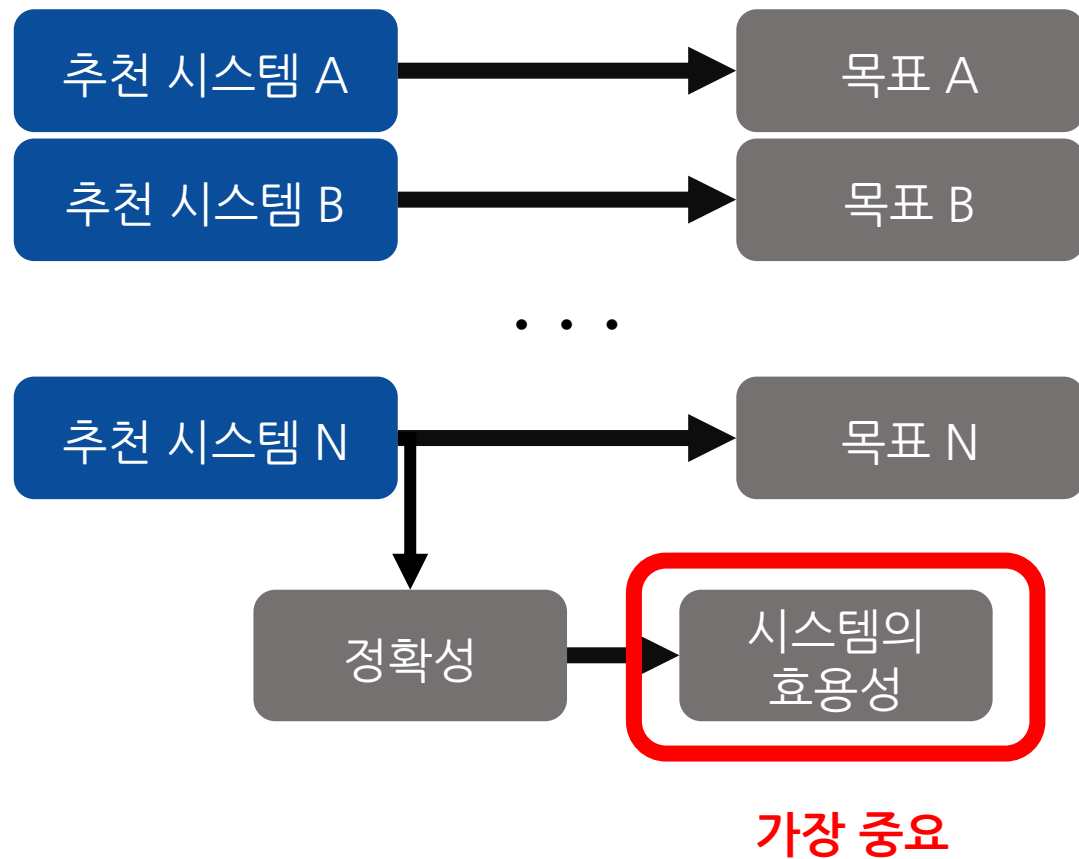
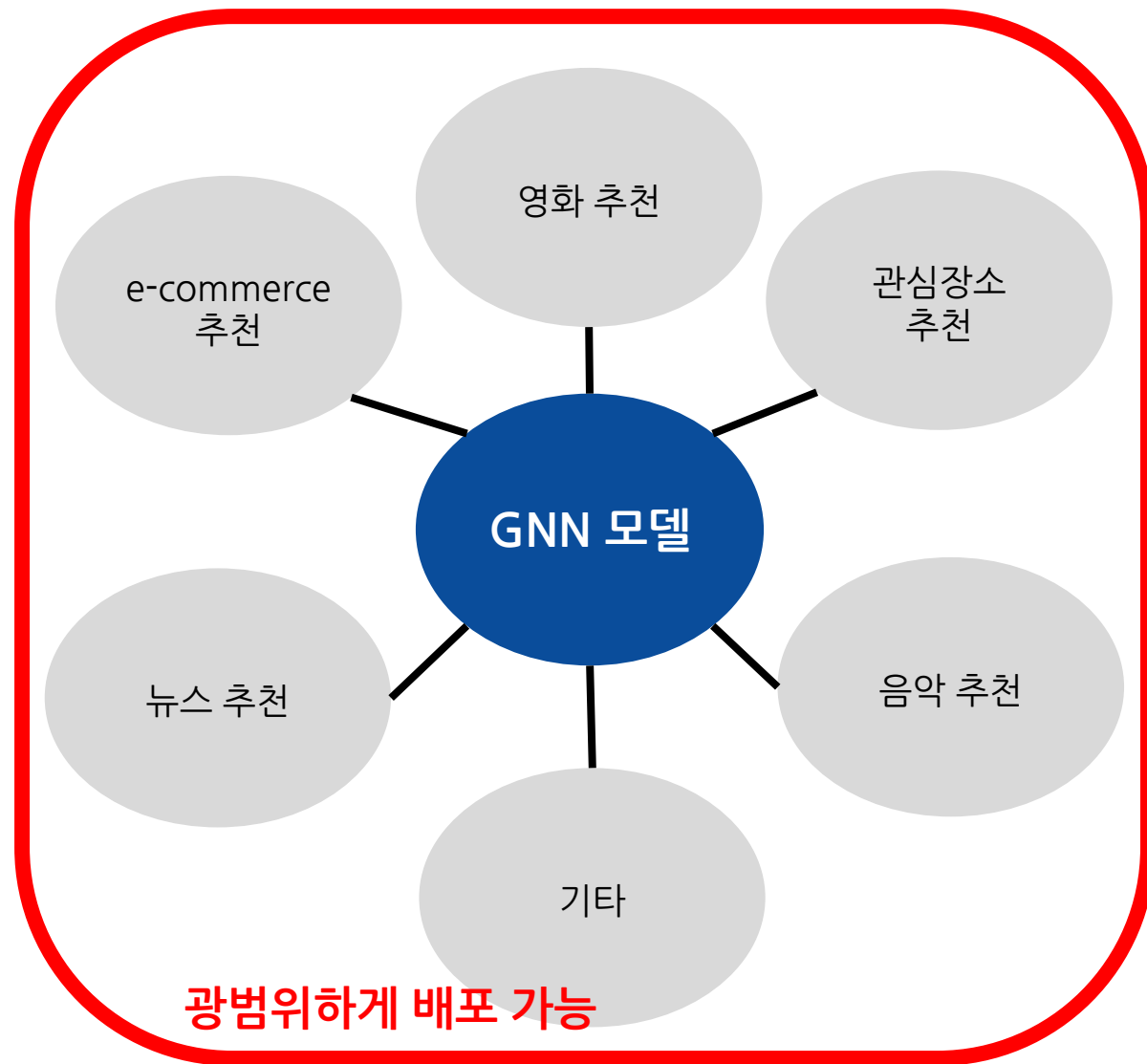


Fig. 2. The typical pipeline of recommender systems.

1) Overview

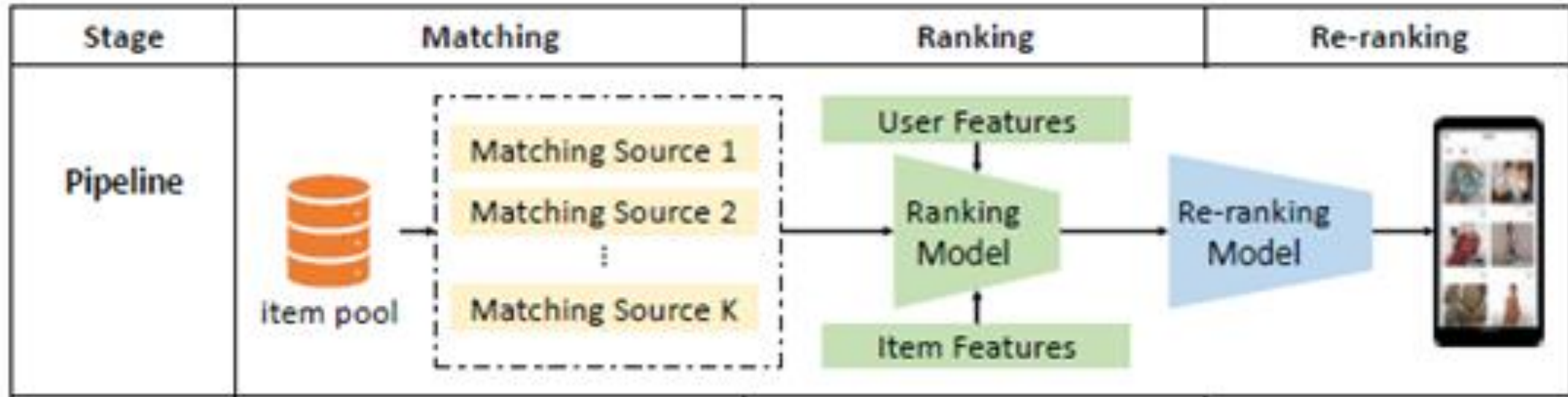


3. Background



2) stages

3. Background



item pool :
사용할 수 있는 모든 item.

1. Social Recommendation

- 다른 사용자와 상호 작용하며 개인의 행동은 개인적, 사회적 요인에 의해 좌우됨

2. Sequential Recommendation

- 시간이 지남에 따라 많은 수의 상호작용 행동을 생성 > 사용자의 과거 행동을 바탕으로 예측

3. Session-based recommendation.

- 제한된 저장 리소스로 인해 익명 사용자의 단편 데이터만 제공받을 수 있어 이를 바탕으로 예측

4. Bundle Recommendation

- 독립적인 item을 추천하는 것이 아닌 사용자가 소비할 item의 조합을 recommendation

5. Cross-Domain Recommendation

- 여러 도메인에 걸쳐 multi-modal 정보와 상호 작용함에 따라 cold start 및 data sparsity problems를 완화 가능

6. Multi-behavior Recommendation

- 한 가지 유형의 동작 대신 여러 유형의 동작에서 추천 시스템과 상호 작용

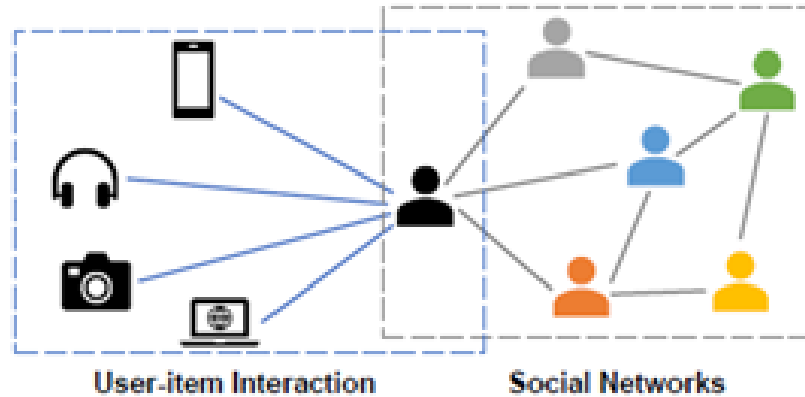


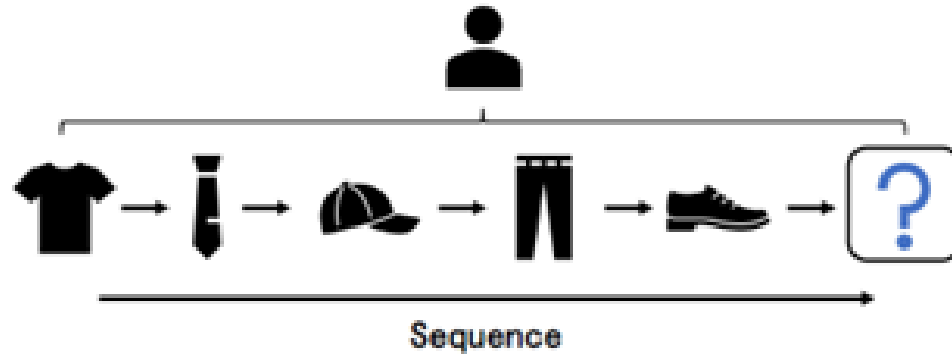
Fig. 3. An illustration of social recommendation. User interactions are affected by both their own preferences and the social factor.

SNS 등으로 다른 사용자와 상호 작용하는 능력으로 인해 개인의 행동은 개인적, 사회적 요인에 의해 좌우됨.

사회적 영향 : 사용자의 행동은 친구들이 무엇을 하거나 생각하는지에 의해 영향을 받음. 또한, 자신과 유사한 선호를 가진 다른 사람들과 사회적 관계를 구축하는 경향이 있음.

사회적 관계는 종종 final performance를 향상시키기 위해 추천 시스템에 통합되는데, 이를 social recommendation 이라 함.

Sequential Recommendation



3. Background

Fig. 4. An illustration of sequential recommendation. Given a user's historical sequence, recommender system aims to predict the next item.

목표

사용자의 과거 item x_1, x_2, \dots, x_n sequence가 주어지면 정보를 추출해 다음 item x_{n+1} 을 예측.

- collaborative filtering[56]과 같이 일반적으로 사용되는 방법은 각 사용자 행동을 사용하여 훈련(단일 item에 대한 user's preference를 직접 모델링)
- sequential recommendation 은 user's historical behavior sequence를 사용하여 timestamp-aware sequence 패턴을 학습하여 사용자가 관심을 가질 수 있는 다음 item을 추천.

주요과제

- 1) 시퀀스 길이가 길면 단기, 장기, 동적 관심사를 동시에 모델링하기 어려워짐.
- 2) 모델링한 시퀀스 이외에도 여러 시퀀스에서 item이 발생 가능하므로 서로 다른 시퀀스 간의 collaborative signals는 더 나은 representation learning을 위해 준비되어야 함.



Fig. 5. An illustration of session-based recommendation. Given an anonymous short session, recommender system aims to predict the next item.

- 실제 상황에서는 제한된 저장 리소스로 인해 장기간 동안 사용자 ID의 행동을 추적할 수 없거나 필요하지 않음. > 사용자 프로필과 장기간의 과거 상호 작용은 사용 불가. 익명 사용자의 단편 데이터만 제공.
- 기존의 recommendation(예: collaborative)은 Session-based recommendation 에서 성능이 저하될 수 있음.
- 익명의 behavioral session data로 다음 item을 예측하는 session-based recommendation(SBR)과 유사.
- sequence recommendation와는 달리, 각 세션에서 사용자의 행동은 세션 기반 특성만 보여주기 때문에 동일한 해당 세션이외의 세션에서는 독립적 처리 필요.

Bundle Recommendation

3. Background

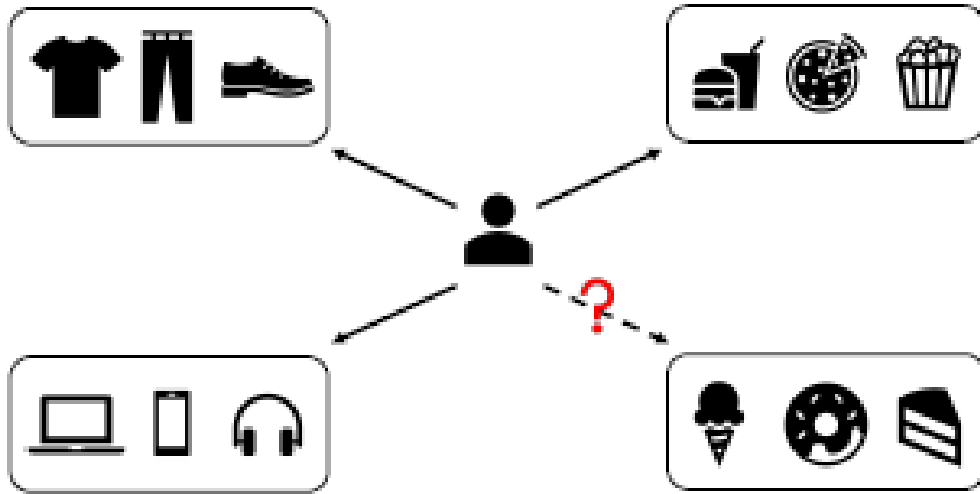


Fig. 6. An illustration of bundle recommendation.

- 기존 추천 시스템은 주로 독립적인 item을 추천
- 번들은 상품 홍보에 중요한 마케팅 전략인 상품들의 집합.
- 목표 : 사용자가 소비할 item의 조합을 추천
- 온라인 플랫폼(예: Spotify의 음악 재생 목록, IKEA의 가구 제품군 등)에서 자주 사용. fashion outfits[83] 및 drug packages[209]에도 사용

최근 유통계에서 핫한 키워드

맥주와 기저귀

기저귀 심부름 나온 아빠가 맥주를 함께 구매하는 경우가 많다고 해요. 그래서 미국 월마트는 이 둘을 함께 진열해 시너지를 냈다고 합니다.

산업통상자원부

Cross-Domain Recommendation

3. Background

- 많은 사용자가 multi-modal 정보를 생성함에 따라 CDR(cross-domain recommendation)은 cold start 및 data sparsity problems를 완화하는 유망한 방법임이 입증
- 크게 single-target CDR(STCDR)과 dual-target CDR(DTCDR)의 두 가지로 구분.
- CDR 방법은 source domain에서 target domain으로 정보를 한 방향으로 전송.
- DTCDR은 다중 대상 CDR(MTCDR)로 확장할 수 있는 소스 도메인과 대상 도메인 모두의 정보 상호 활용을 강조.

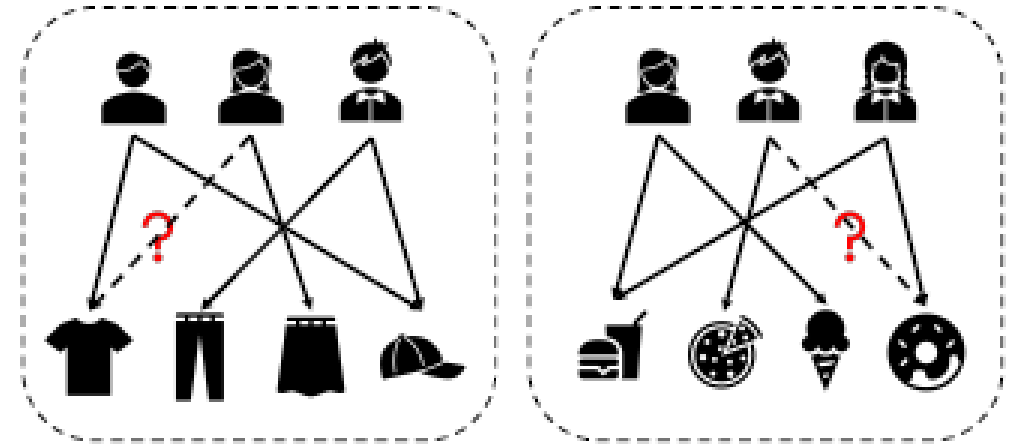


Fig. 7. An illustration of cross-domain recommendation.

Multi-behavior Recommendation

3. Background

- 사용자는 한 가지 유형의 동작 대신 여러 유형의 동작으로 상호 작용.
- 제품을 구매하기 전 클릭, 카트에 추가, 공유 등의 경우가 많음.
- 궁극적인 목표는 사용자가 구매할 제품을 추천하는 것.
- 사용자의 click, sharing, and other behaviors에 비해 구매 행태는 매우 희박(sparse).
- 각 사용자 u 와 item v 에 대해 서로 다른 유형의 행동 $\{y_{\{1\}}, y_{\{2\}}, \dots, y_{\{K\}}\}$ 이 있다고 가정. i 번째 동작의 경우, 사용자가 관찰된 동작을 가지고 있으면 $y_i = 1$, 그렇지 않으면 $y_i = 0$
- 주요 challenges
 - 1) 행동마다 target behavior에 미치는 영향이 다름.
 - 2) item에 대한 다양한 유형의 행동으로부터 포괄적인 representations 학습이 어려움.



Fig. 8. An illustration of multi-behavior recommendation.

4. Objectives

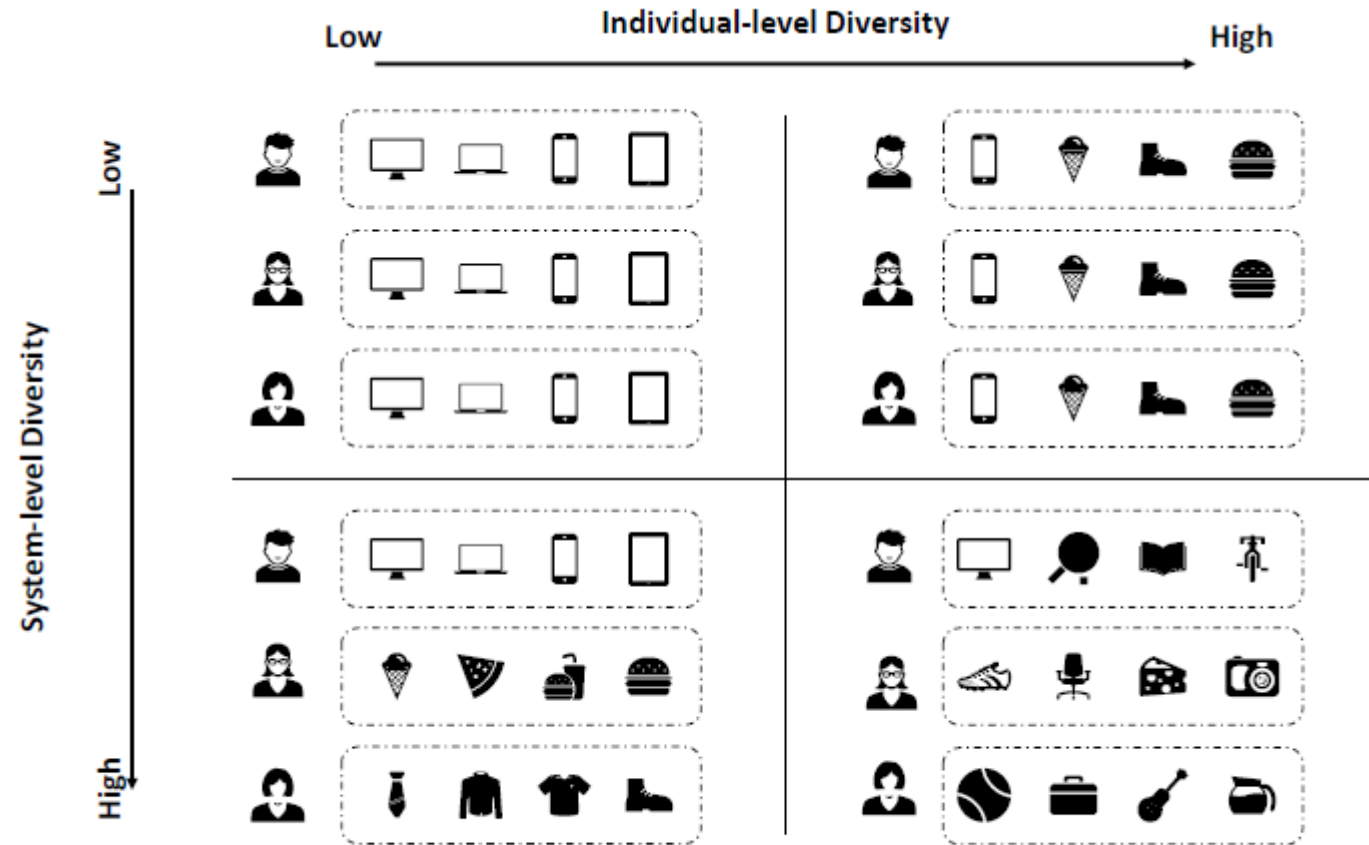


Fig. 9. Illustration of individual-level diversity and system-level diversity.

- 추천 시스템에서는 개인 수준의 다양성과 시스템 수준의 다양성이라는 두 가지 유형이 고려됨.

개인수준

- 유사한 item 반복시 시스템 탐색을 꺼리게 됨
- 각 사용자에게 대한 recommendation item의 차이를 측정

시스템수준

- 서로 다른 사용자의 추천 결과를 비교해 서로가 유사하지 않아야 함.
 - 시스템 수준의 다양성이 낮음 : long-tail item은 무시한 채 모든 사용자에게 인기 item 추천
- 시스템 수준의 다양성을 long-tail recommendation 이라고도 함.

공통 주요 challenges

- 사용자별 item별 signal strength가 다름.
 - 각 사용자마다 dominant topics와 disadvantaged topics가 존재.
- accuracy-diversity dilemma : accuracy와 모순되어 초래.

- 정확한 추천 결과 뿐만 아니라 사용자에게 item을 추천하는 방법과 이유에 대한 설명을 생성하는 것
- 설명 가능성을 높이면 사용자의 perceived transparency[131], 지속성(persuasiveness)[142] 및 신뢰도(trustworthiness)[75]를 향상시킬 수 있으며 시스템 디버그 & 재정립 가능.
- 선행연구 핵심점

1) 본질적인 설명 가능한 모델을 설계하기 위해 노력

- 예) Explicit Factor Models[202], Hidden Factor 및 Topic Model[106], TriRank [54]과 같은 투명한 논리로 모델을 설계함으로써 recommendation 결과의 설명 가능성 보장

2) Compromise : post-hoc separate models

- "black box" 추천 시스템에 의해 생성된 결과를 설명하기 위해 post-hoc separate models을 설계. Description Mining [119]

공통 주요 challenges

- 1) explainable information 표현시 그래프 구조 item 속성이 필요하므로 GNN의 없이는 어려움.
- 2) reasoning
recommendations은 지식 그래프의 external knowledge에 의존.

- 추천 시스템은 데이터와 알고리즘에 의해 편향될 수 있어, 공정성에 대한 우려 존재
- 공정성의 범주

1) user fairness

- 특정 사용자 또는 인구 통계 그룹 간에 알고리즘 편향이 없음을 보증.

2) item fairness

- 서로 다른 항목에 대한 공정한 노출 또는 서로 다른 항목 간의 인기 편향이 없음을 나타냄. > 다양성에서 해결 가능

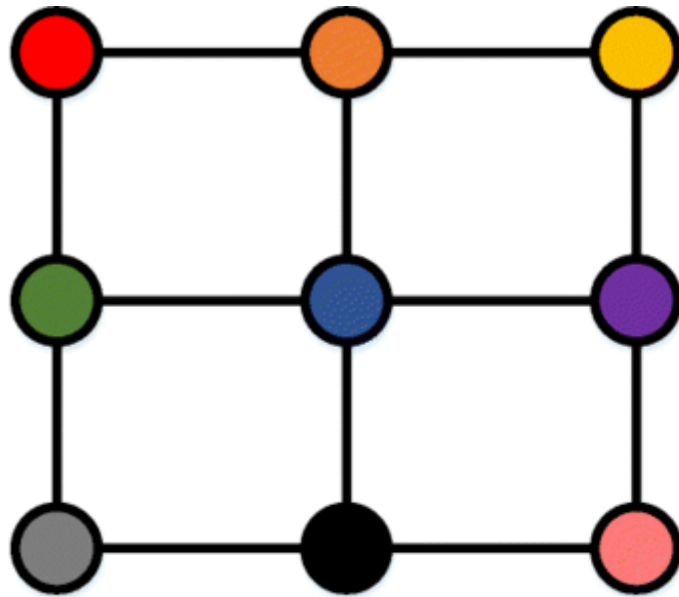
공정성 향상 방법

- 직접 recommendation의 편차 줄이는 것
- 후처리 방법에서 불공정을 완화하기 위한 item의 순위를 매김

5. GNN Basic

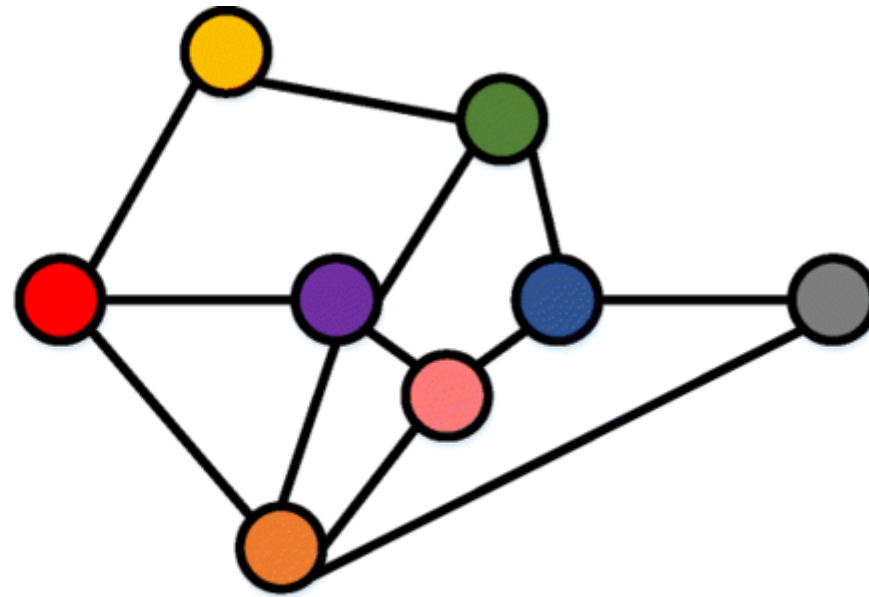
5. Graph Neural Networks

5. GNN basic



CNN

In Euclidean Space



GNN

In Non-Euclidean Space

- 컨볼루션 신경망(CNN)과 그래프 표현 학습(GRL)의 발전으로 GNN 주목
- 이미지, 텍스트와 같은 일반적 Euclidean 데이터에 CNN적용시 localized features을 추출하는 데 매우 효과적.

5. Graph Neural Networks

5. GNN basic

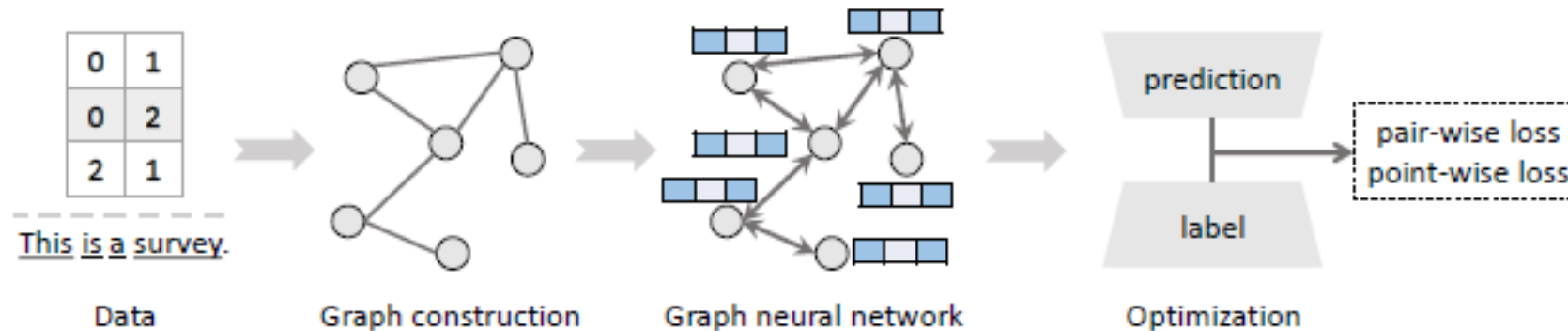


Fig. 10. The overall procedure of implementing a GNN model: construct graphs from data (e.g., table or text), design tailored GNN for generating representations, map representations to prediction results, and further define loss function with labels for optimization.

- GRL 측면 : 그래프의 복잡한 연결 구조를 나타내는 그래프 노드, 엣지 또는 하위 그래프에 대한 저차원 벡터를 생성하는 것 목표.
- CNN 측면 : 그래프와 같은 non-Euclidean 데이터의 경우, CNN은 operation objects(예: 그래프의 이미지 또는 노드의 픽셀)가 크기가 고정되지 않은 상황을 처리하기 위한 일반화가 필요.
- CNN과 GRL을 결합하여 구조 정보를 distill하고 high-level representations을 학습하기 위한 GNN 등장

1) Graph Construction.

2. Introduction

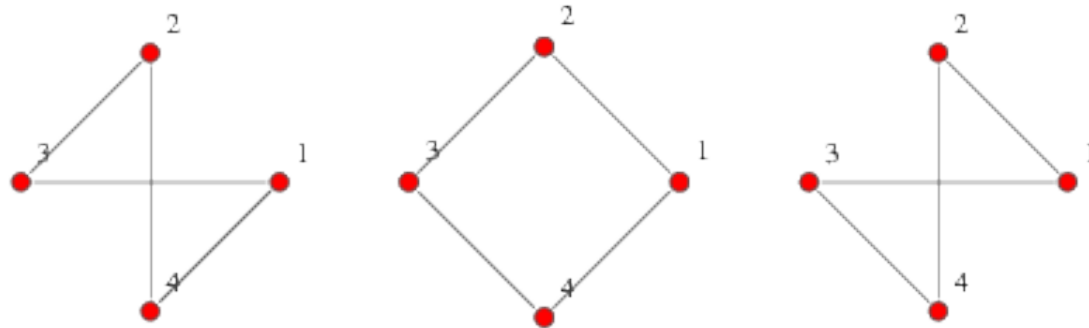


Table 1. Frequently used notations.

V/E	The set of graph nodes/edges
\mathcal{U}/\mathcal{I}	The set of users/items
\mathcal{N}_i	The neighborhood set of graph node i
\mathbf{h}_i^l	The embedding of graph node i in the l -th propagation layer
\mathbf{A}	The adjacency matrix of graph
\mathbf{W}^l	Learnable transformation matrix in the l -th propagation layer
$\delta(\cdot)$	Nonlinear activation function
\parallel	Concatenation operation
\odot	Hadamard product

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

- 그래프 : $\mathcal{G} (V, E)$. V : node, E : edge

- 최근 GNN 기반 모델의 3가지 범주

1) Homogeneous graph : 각 엣지가 두 개의 노드만 연결하고 노드와 엣지의 유형만 있는 그래프

2) Heterogeneous graph : 각 엣지가 두 개의 노드만 연결하고 여러 유형의 노드 또는 엣지가 있는 그래프

3) Hypergraph : 각 가장자리가 세 개 이상의 노드를 결합하는 그래프

- 이미지 및 텍스트와 같이 구조화되지 않은 데이터에는 노드와 엣지를 수동으로 정의 필요

2) Network Design

5. GNN basic

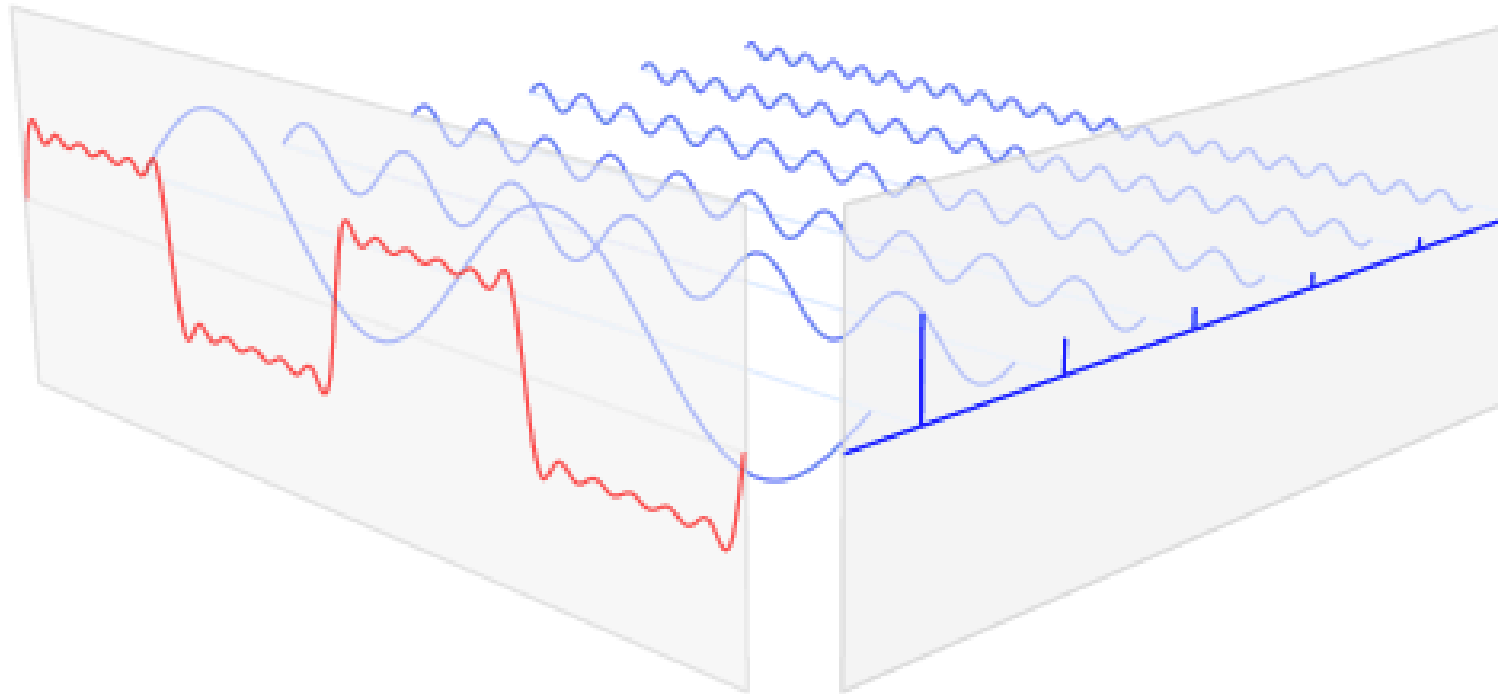
- 일반적으로 GNN 모델은 spectral 과 spatial 모델로 분류
- "information"은 임베딩(저차원 벡터)으로 표현. 주요하고(primary) 중추적인(pivotal) 작업은 이웃 임베딩을 aggregating하고 target embedding과 융합하여 layer by layer로 그래프 임베딩을 업데이트하는 것을 포함하여 구조적 연결에 따른 그래프에 embeddings를 propagate하는 것.

	spectral	spatial
공통점	그래프 node와 edge 사이의 고차 상관관계를 포착하기 위해 이웃 정보를 반복적으로 수집	
처리방법	그래프를 신호로 간주하고 spectral 영역에서 그래프 컨볼루션 처리	CNN과 같은 weighted aggregation를 통해 localized feature을 추출하기 위해 그래프 구조에 대한 컨볼루션을 직접 수행.
특징	<p>그래프 신호는 그래프에 정의된 Fourier transform에 의해 spectral 영역으로 먼저 변환된 다음 필터가 적용. 마지막으로 처리된 신호는 spatial domain으로 다시 변환</p> $g \star x = \mathcal{F}^{-1}(\mathcal{F}(g) \odot \mathcal{F}(x))$ <p>여기서 \mathcal{F}는 graph Fourier transform</p>	

2-1) spectral

2. Introduction

- 정의 : 신호/오디오/이미지/그래프를 time/spatial domain 이 아니라 frequency domain으로 변환해 분석. 특정 신호를 단순한 요소(wavelets, graphlets)의 (조)합으로 **분해** 하는 것 > 푸리에 변환



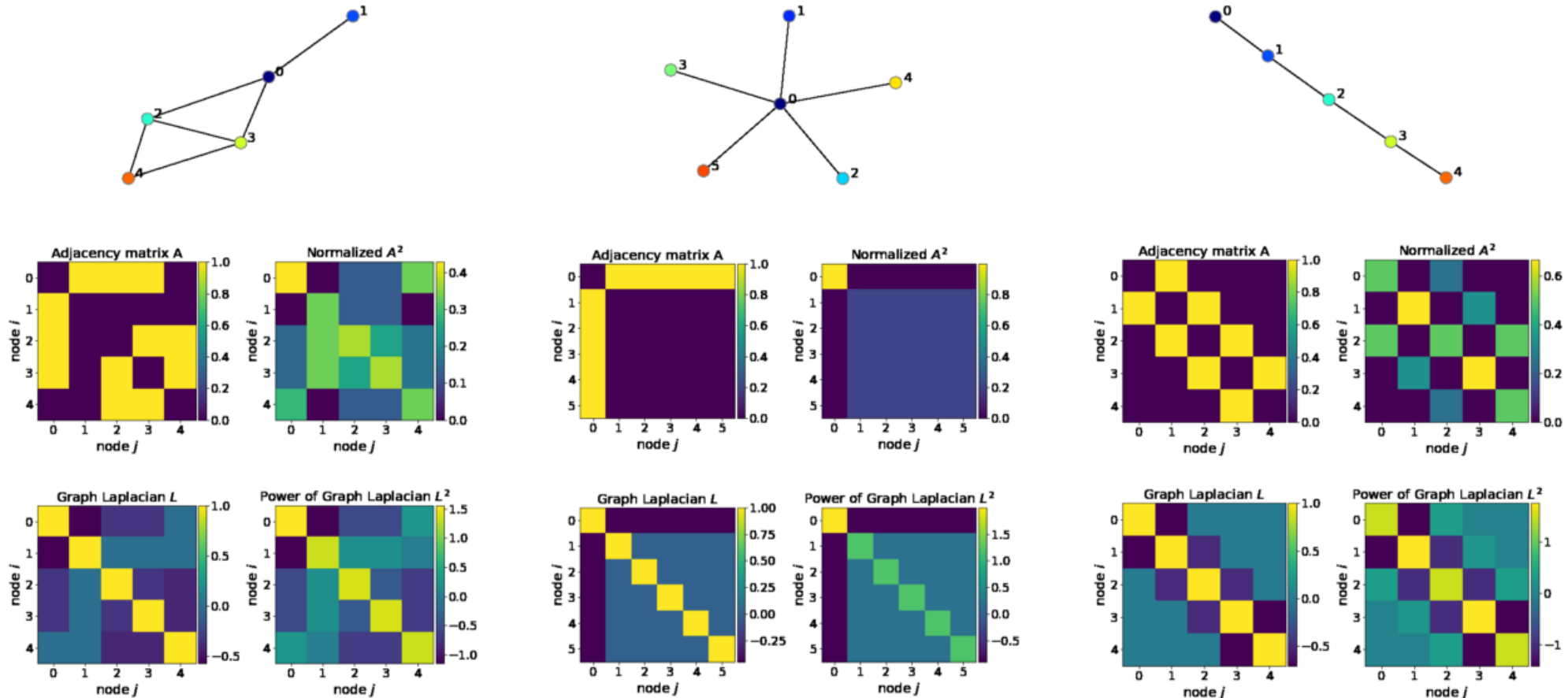
- **그래프 Laplacian L 의 고유 분해***eigen-decomposition* 하는 것.
- 그래프 **Laplacian L** 을 특별한 방법으로 정규화된 adjacency matrix A 로 생각할 수 있으며 고유 분해는 그래프를 구성하는 elementary orthogonal components을 찾는 방법.

2-1) spectral

2. Introduction

+ 그래프 Laplacian은 노드 i 에 힘을 가하면 그래프에서 "에너지"가 어떤 방향으로 얼마나 부드럽게 확산 되는지 보여줌

- 인접 행렬 A 는 A^n 가 노드 간의 n 홉 연결을 의미



2-1) spectral

2. Introduction

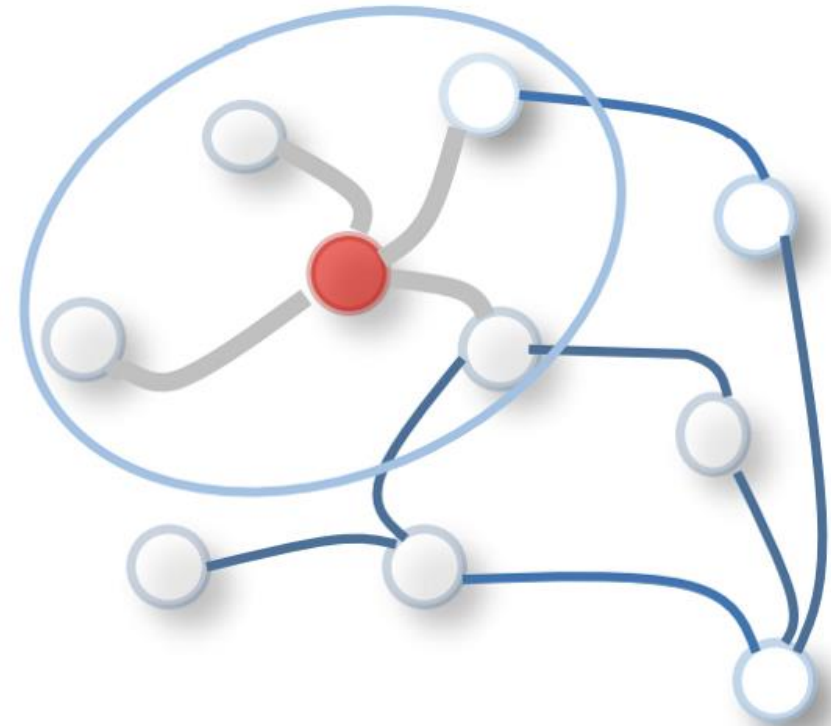
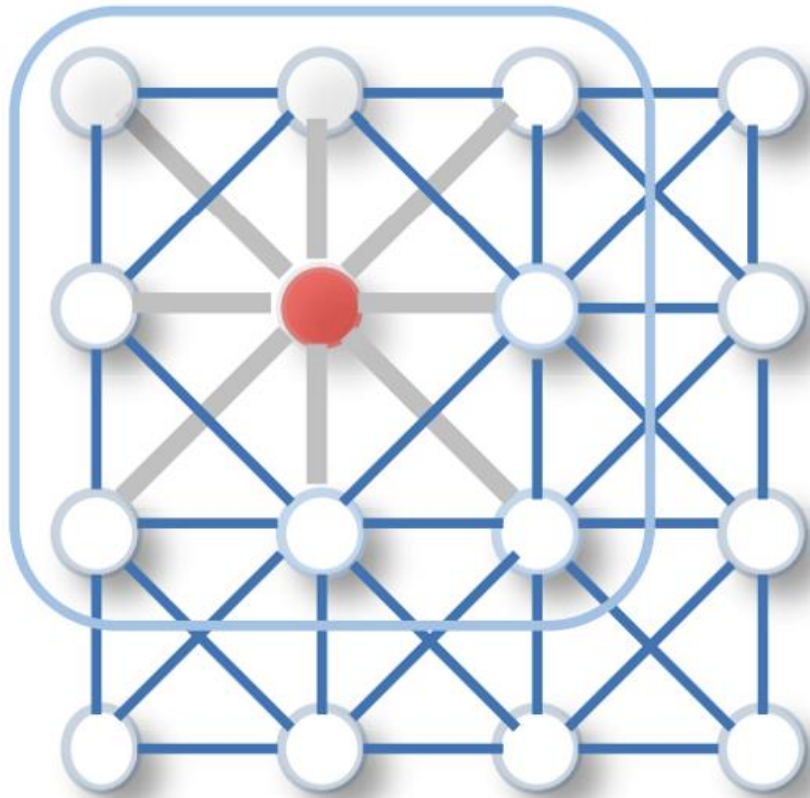
가정

1. Symmetric normalized Laplacian 가정
 2. A 는 $A = A^T$ 대칭행렬.
 3. 그래프는 방향이 없음.
- 2,3 조건이 만족되지 않으면 노드 차수가 정의되지 않으며 Laplacian을 계산하기 위해 몇 가지 가정 필요
- eigen-decomposition of the graph Laplacian.
- 고유 분해에 의해 찾을 수 있는 그래프 Laplacian L 의 고유 벡터 V 인 보다 일반적인 basis를 사용 $L = V\Lambda V^T$, 여기서 Λ 은 L 의 고유값
 - spectral graph convolution : 가장 작은 고유 값에 해당하는 몇 개의 고유 벡터를 사용

2-2) spatial

2. Introduction

- Spatial 컨볼루션은 노드에 대해 공간적으로 컨볼루션하는 방식에서 일반 컨볼루션과 유사
- 각 노드와 가깝게 연결된 **고정된 이웃 노드에서만** convolution 연산을 수행해 노드를 업데이트.



2-3) Inductive vs Transductive

2. Introduction

- Inductive : 전체가 아니라 자기 주변의 이웃만 보면 됨 > 단점 : 새로운 하위 그래프가 주어졌을 때 노드 임베딩시 “aligning” 필요
- Transductive : 전체를 다 보는 것으로, 새로운 노드가 추가될 때 처음부터 다시 학습해야 함

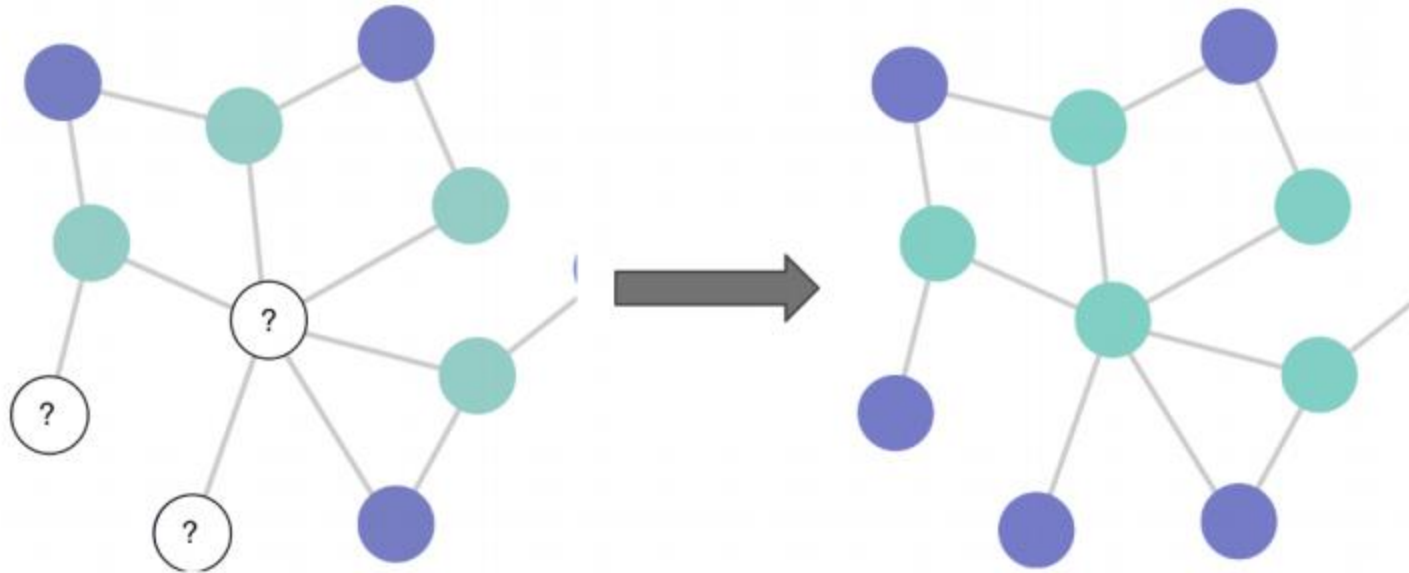


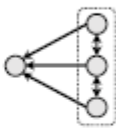




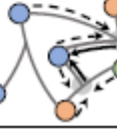


Figure 1. Node classification in transductive setting. At training time, the learning algorithm has access to all the nodes and edges including nodes for which labels are to be predicted.

2) Network Design

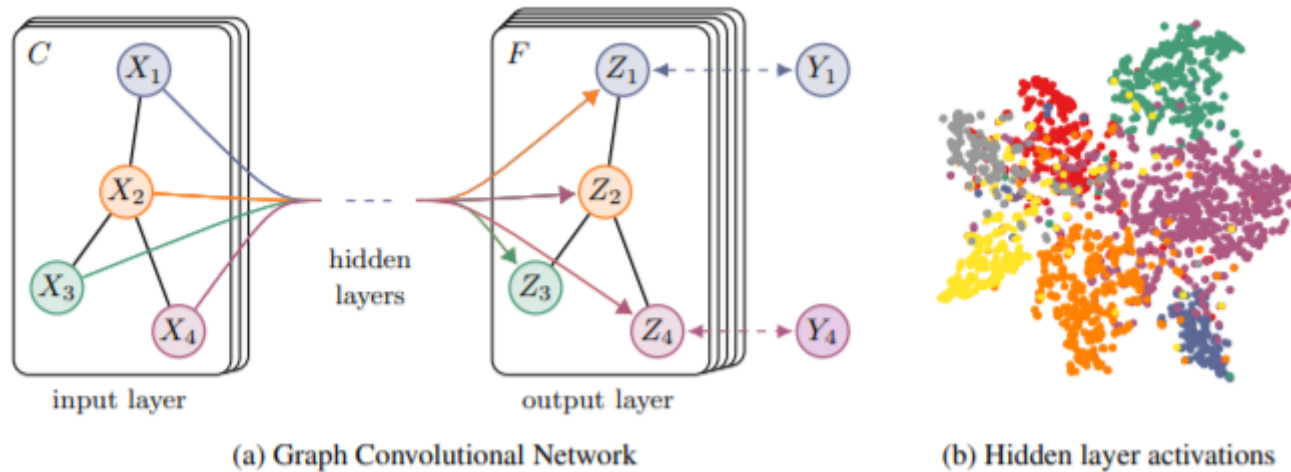
5. GNN basic

Model	Graph type of GNN	GNN category	Aggregate
GCN		Spectral	
GraphSAGE		Spatial	
GAT		Spatial	
HetGNN		Spatial	
HGNN		Spectral	

- 일반적으로 GNN 모델은 spectral 과 spatial 모델로 분류
- "information"은 임베딩(저차원 벡터)으로 표현. 주요하고(primary) 중추적인(pivotal) 작업은 이웃 임베딩을 aggregating하고 target embedding과 융합하여 layer by layer로 그래프 임베딩을 업데이트하는 것을 포함하여 구조적 연결에 따른 그래프에 embeddings를 propagate하는 것.

2) Network Design

5. GNN basic



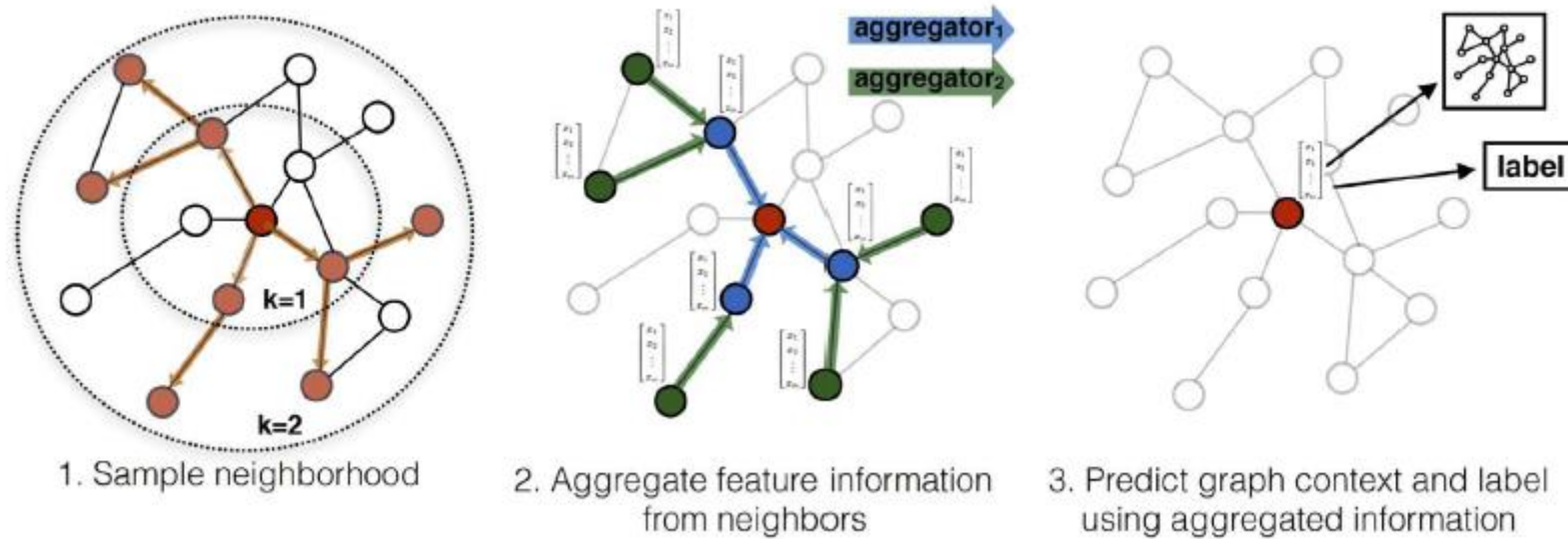
1. GCN : semi-supervised classification을 위해 그래프 컨볼루션과 신경망을 결합한 일반적인 spectral 모델.

$$H^{l+1} = \delta(\tilde{D}_2^{-\frac{1}{2}} \tilde{A} \tilde{D}_2^{-\frac{1}{2}} H^l W^l)$$

- $H^l \in R^{|V| \times D}$ 는 l 번째 convolution layer 의 graph nodes에 대한 embedding matrix.
- D 는 embedding dimension .
- $\tilde{A} \in R^{|V| \times |V|}$ 는 self-loop하는 graph의 adjacency matrix.
- node i 가 j 와 연결, $i=j$ 일 때 각 entry $\tilde{A}_{ij} = 1$. 반대는 $\tilde{A}_{ii} = 0$, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$

2) Network Design

5. GNN basic



2. GraphSAGE : 대상 노드의 이웃을 샘플링해 임베딩 집계하고 병합하는 spatial GNN 모델.

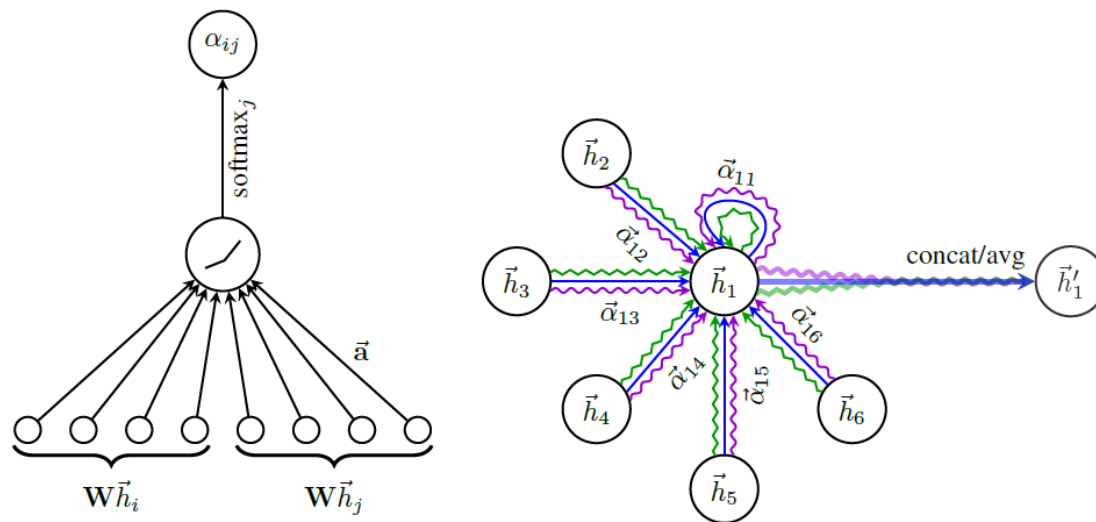
$$h_{\mathcal{N}_i}^l = \text{AGGREGATE}(\{h_j^l, \forall j \in \mathcal{N}_i\})$$

$$h_{\mathcal{N}_i}^{l+1} = \delta(W^l[h_i^l || h_{\mathcal{N}_i}^l])$$

- \mathcal{N}_i : target 노드 i 의 샘플링된 이웃
- Aggregate feature: MEAN, LSTM등 옵션 존재

2) Network Design

5. GNN basic



3. GAT(Graph Attention network) : 일반화 능력 부족과 inverse의 계산과 같은 문제를 해결하는 spatial GNN.

- GAT는 서로 다른 노드에 다른 가중치로 임베딩을 집계하기 위해 attention 메커니즘을 활용.

Propagation 공식 : $h_i^{l+1} = \delta(\sum_{j \in \mathcal{N}_i} \alpha_{ij} W^l h_j^l)$

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T [W^l h_i^l || W^l h_j^l]))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(\mathbf{a}^T [W^l h_i^l || W^l h_k^l]))}$$

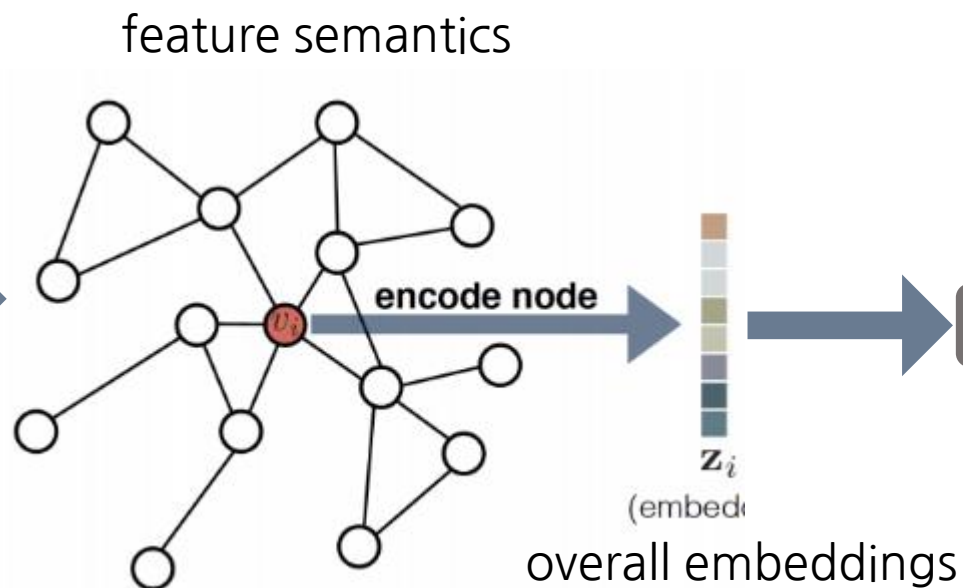
- α_{ij} : 노드 j 에서 노드 i 로의 propagation weight / \mathcal{N}_i 은 노드 i 자체를 포함한 노드 i 의 neighborhood set.
- attention 메커니즘은 학습 가능한 벡터 \mathbf{a} 에 의해 매개 변수화된 fully connected 레이어를 통해 구현.

3) Model optimization

- 학습 표준 절차

5. GNN basic

설계된 네트워크 수행

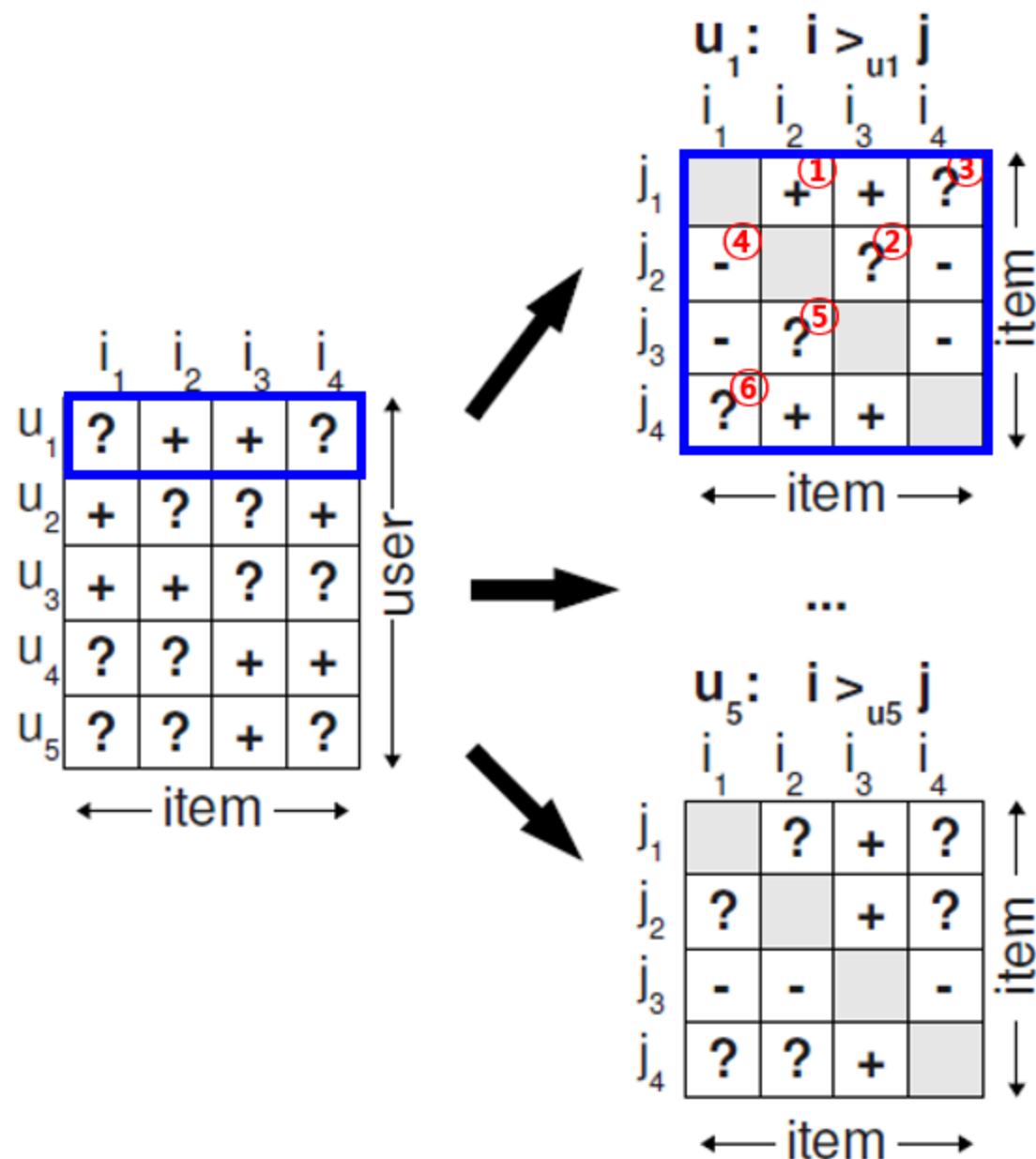
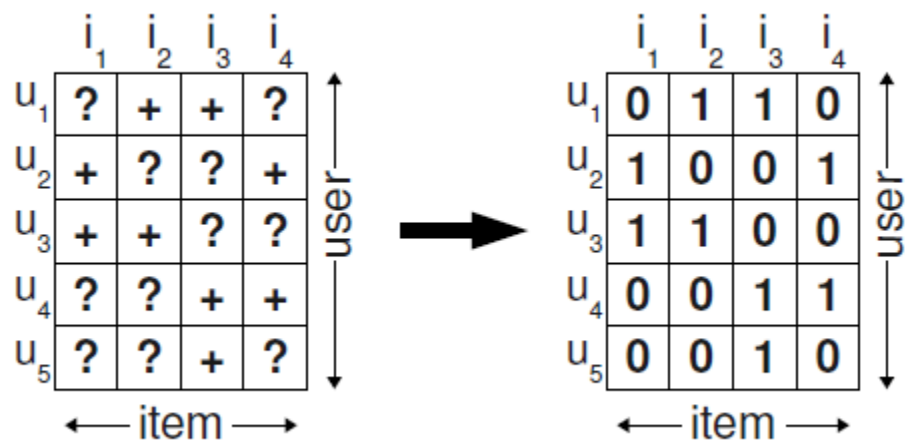


loss function을 formulate

학습에 optimizers 사용

3) Model optimization

- 특정 task에 대해 선택할 수 있는 몇 가지 유형의 mapping functions(예: MLP, inner product)와 loss function(e.g. pair-wise, point-wise)가 존재.
 - Pair-wise loss function 는 positive 와 negative 샘플 간의 구별 필요.
 - 일반적인 공식 BPR은 $\mathcal{L} = \sum_{p,n} -\ln \sigma(s(p) - s(n))$
 - σ : 시그모이드 함수. p 과 n 는 각각 positive 와 negative, $s(\cdot)$: samples을 measuring

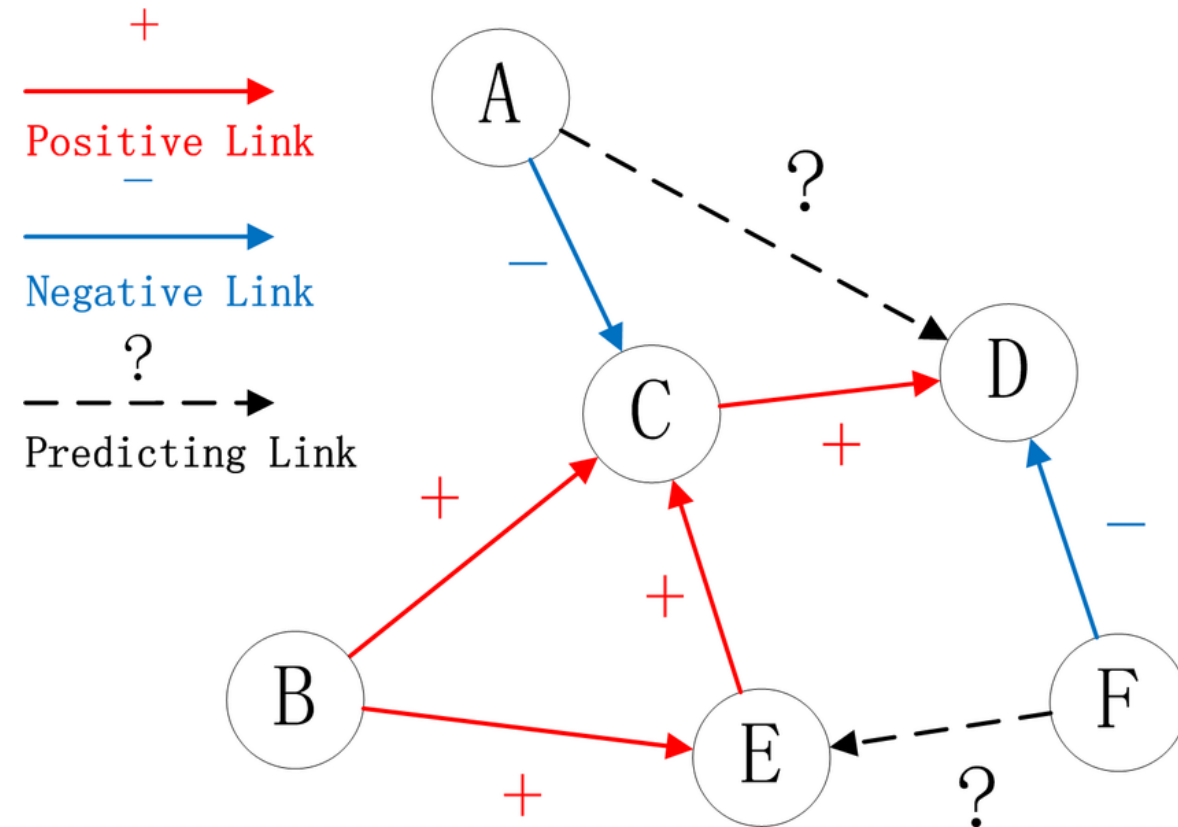


3) Model optimization

1. link prediction : 두 노드 i, j 사이에 엣지가 존재할 가능성 정의 필요.
 - 각 propagation layer의 node embeddings과의 similarity 계산. $s(i, j) = f(\{h_i^l\}, \{h_j^l\})$.
 - 각각 관찰된 positive와 무작위 선택한 negative 샘플로 구성된 데이터 $O = \{(i, j, k)\}$ 구성가능
 - 사용자 i 가 item j 와 상호 작용했지만 item k 와는 상호 작용하지 않았음을 나타냄.
 - k 는 이전에 상호 작용하지 않았던 다른 모든 item에서 샘플링.
 - pair-wise loss BPR이 선택되면 최적화 object

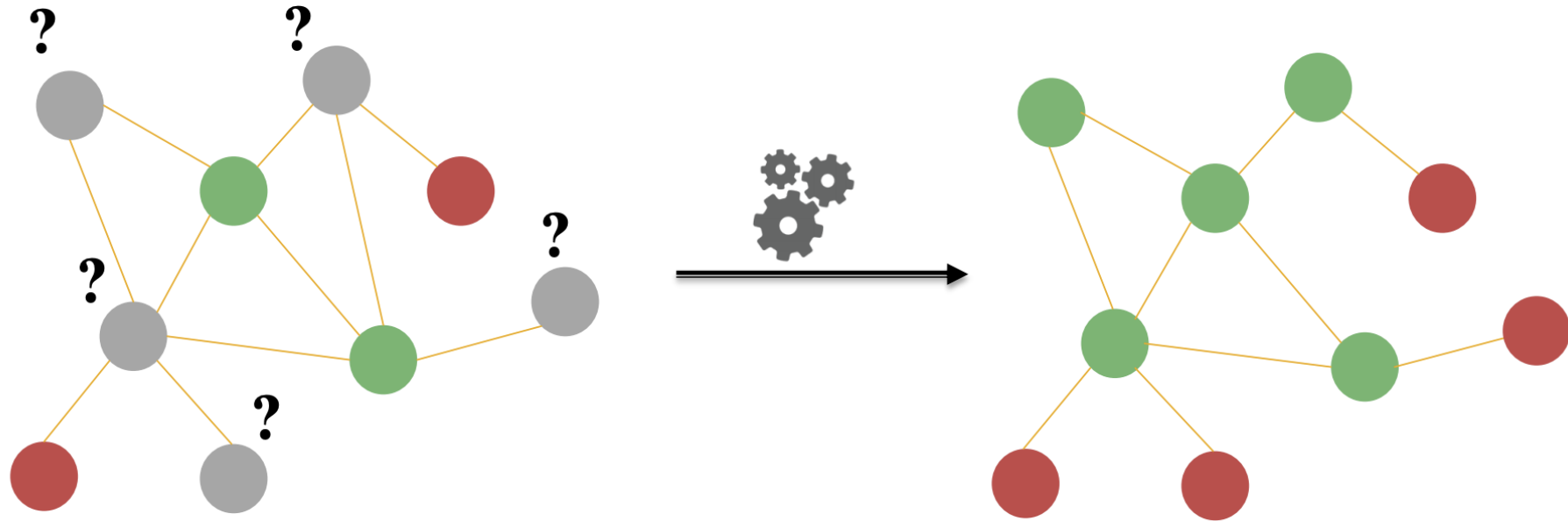
$$\mathcal{L} = \sum_{(i, j, k) \in O} -\ln \sigma(s(i, j) - s(i, k))$$

5. GNN basic



3) Model optimization

5. GNN basic



2. node classification : 노드 임베딩은 다음과 같은 클래스에 속하는 확률 분포로 변환 $p_i = f(\{h_i^l\})$

- 여기서 $p_i \in \mathbb{R}^{c \times 1}$ 은 분포, c 는 클래스 수. 데이터는 $O = \{(i, j, k)\}$. $y \in \mathbb{R}^{c \times 1}$ 와 i 는 class c 를 따르는 $y_{ic} = 1$, 그렇지 않으면 $y_{ic} = 0$
- 일반적으로 point-wise loss function는 classification task에서 자주 선택 e.g. cross-entropy loss $\mathcal{L} =$

$$-\sum_{(i,j,k) \in O} y_i^T \log p_i$$

3. Why are GNNs required for recommender systems

1. structural data
2. high-order connectivity
3. supervision signal

5. GNN basic

Q & A
감사합니다