Mining of Real-world Hypergraphs: Concepts, Patterns, and Generators

Geon Lee, Jaemin Yoo, Kijung Shin

CIKM'22

Mining of Real-world Hypergraphs: Concepts, Patterns, and Generators

- 0. Introduction
- 1. Static Structural Patterns
- 2. Dynamic Structural Patterns
- 3. Generative Models

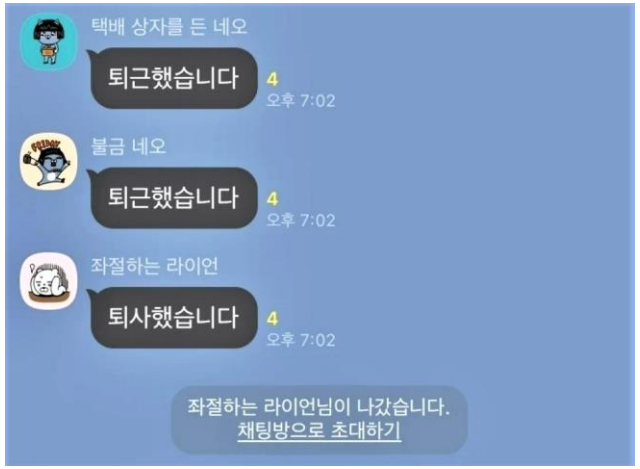
Mining of Real-world Hypergraphs: Concepts, Patterns, and Generators

- 0. Introduction — 전반적인 개념 소개 및 설명
- 1. Static Structural Patterns — 구조적 패턴을 통한 Hypergraph 분석
- ~~2. Dynamic Structural Patterns~~ — ~~구조적 패턴을 통한 Time-evolving Hypergraph 분석~~
- 3. Generative Models — Static, ~~Dynamic~~ Hypergraph 생성 모델

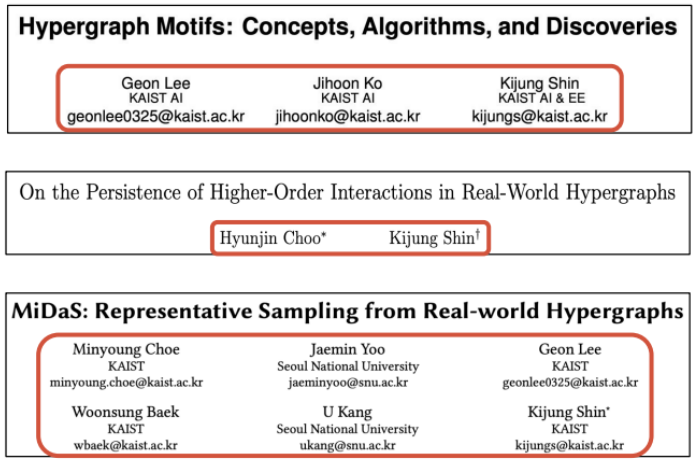
Hypergraph?

Hypergraph

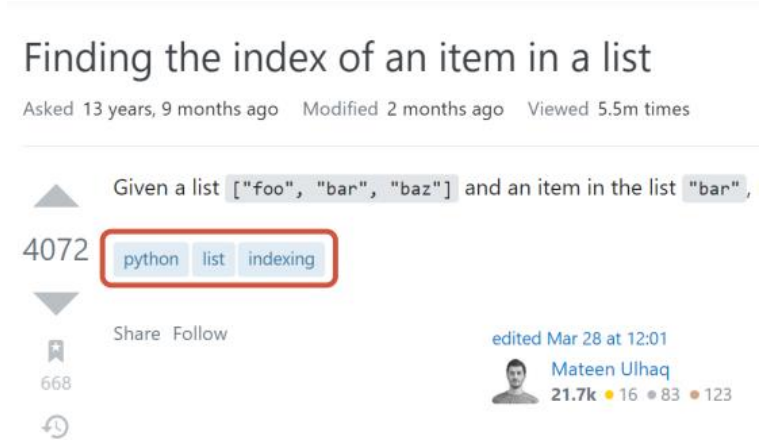
그룹 간 상호작용



같은 채팅방에 존재하는 사람들 (그룹)



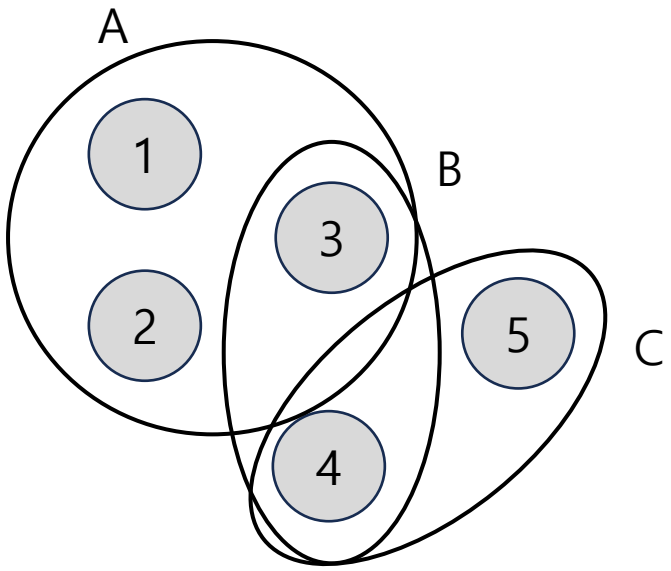
같이 논문을 쓴 사람들 (그룹)



한 게시물의 태그들 (그룹)

Hypergraph

그룹 간 상호작용



채팅방 A 참가자: {1, 2, 3}

채팅방 B 참가자: {3, 4}

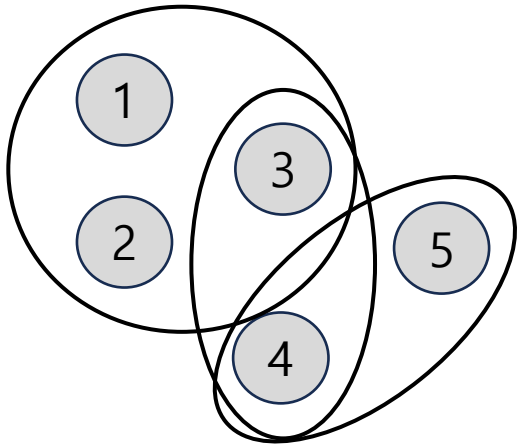
채팅방 C 참가자: {4, 5}

채팅방: Hyperedge (Group)

사용자: Node

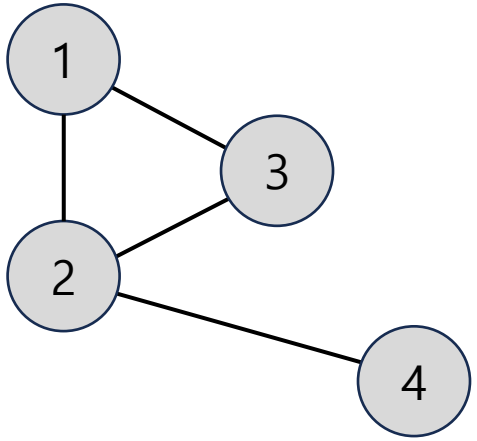
Hypergraph

일반적인 그래프와 비교



Node

Hyperedge (N개의 Node를 연결)

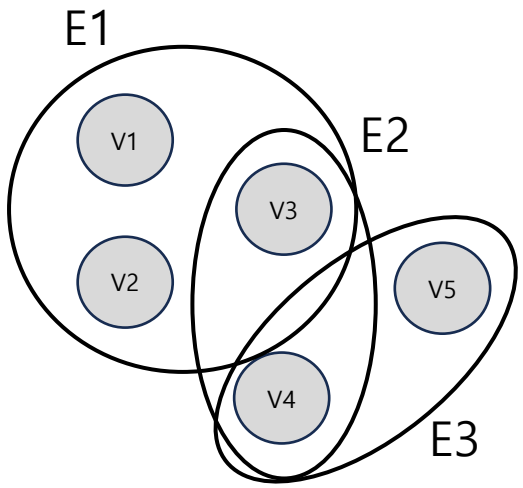


Node

Edge (2개의 Node를 연결)

Hypergraph

Hypergraph의 표현

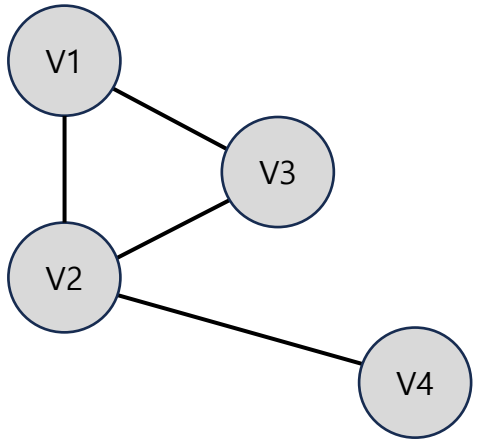


Node (|V| = 5)

Hyperedge (|E| = 3)

	E1	E2	E3
V1	1	0	0
V2	1	0	0
V3	1	1	0
V4	0	1	1
V5	0	0	1

Incidence matrix
|V| * |E|



Node (|V| = 4)

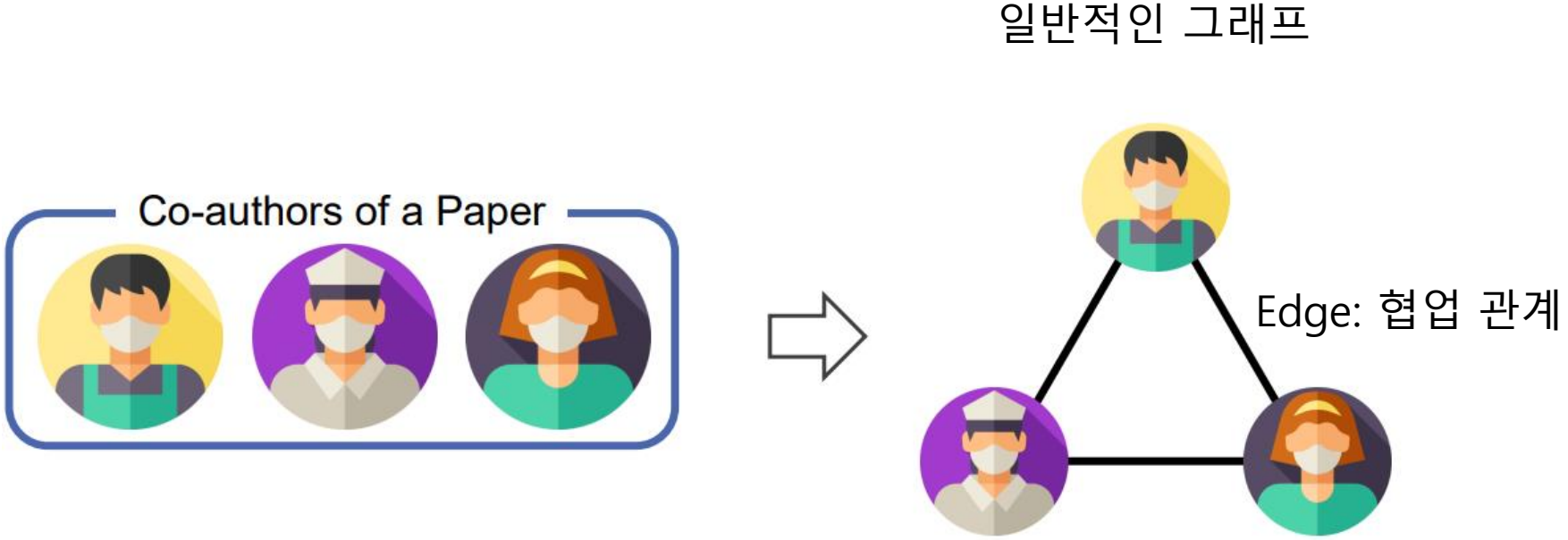
Edge (|E| = 4)

	V1	V2	V3	V4
V1	0	1	1	0
V2	1	0	1	1
V3	1	1	0	0
V4	0	1	0	0

Adjacency matrix
|V| * |V|

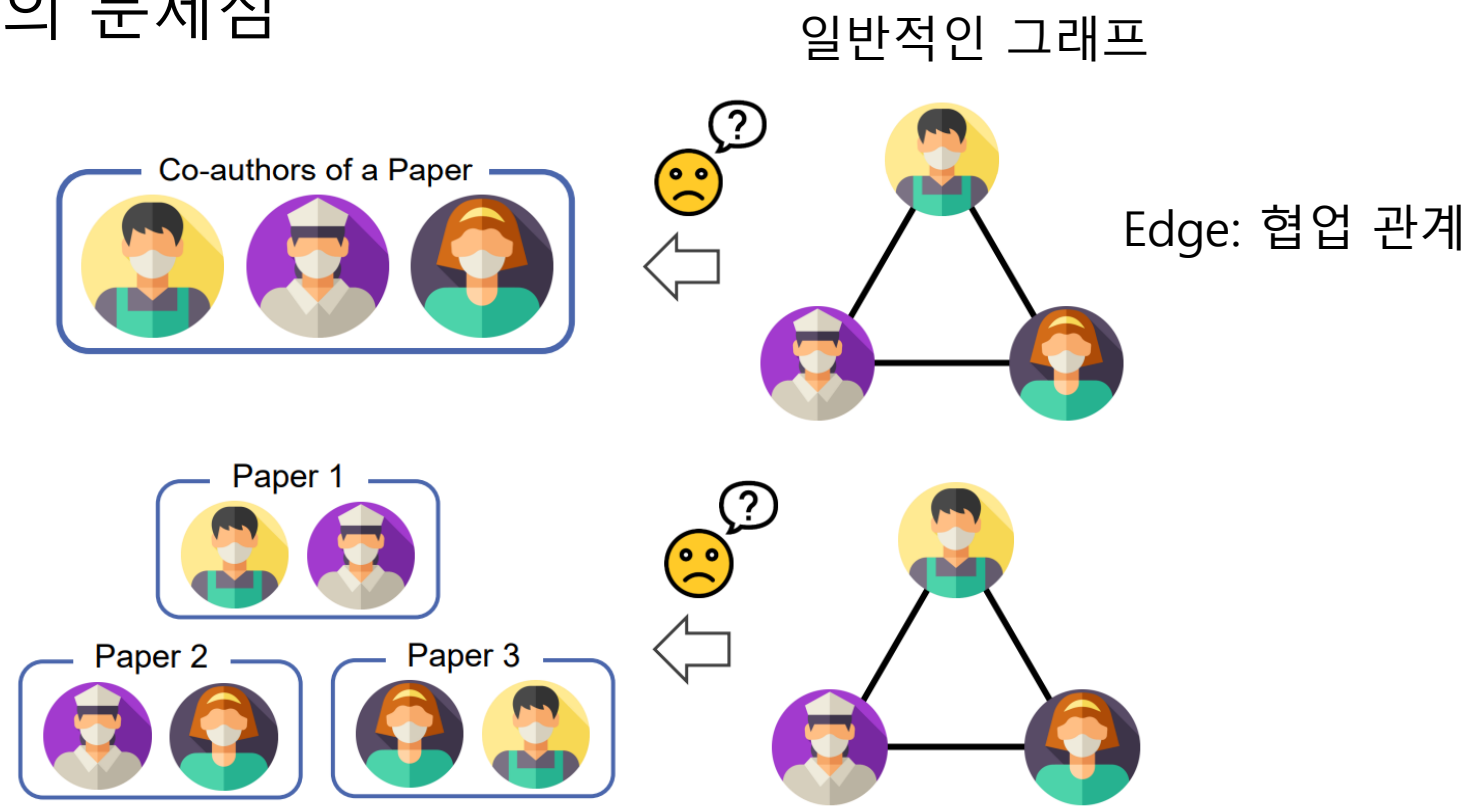
그래서 왜 필요한가?

일반적인 그래프의 문제점



그래서 왜 필요한가?

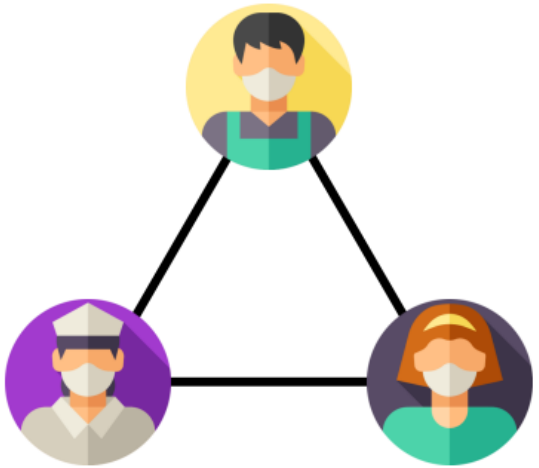
일반적인 그래프의 문제점

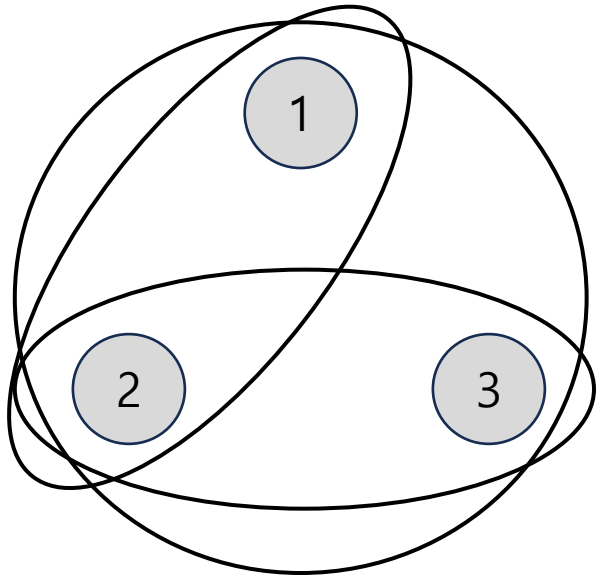


Q. 3명이 함께 논문을 쓴 적이 있나? -> ??

그래서 왜 필요한가?

일반적인 그래프의 문제점

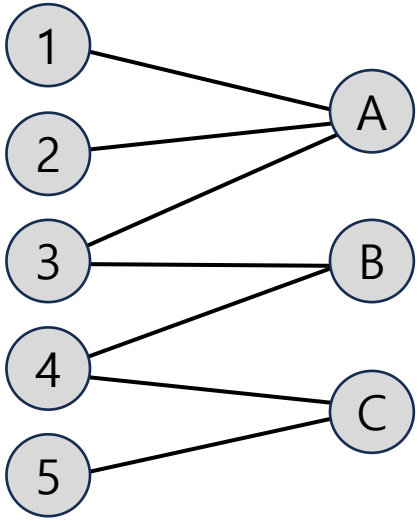




Q. 3명이 함께 논문을 쓴 적이 있나? -> 있다!

Hypergraph

Bipartite graph

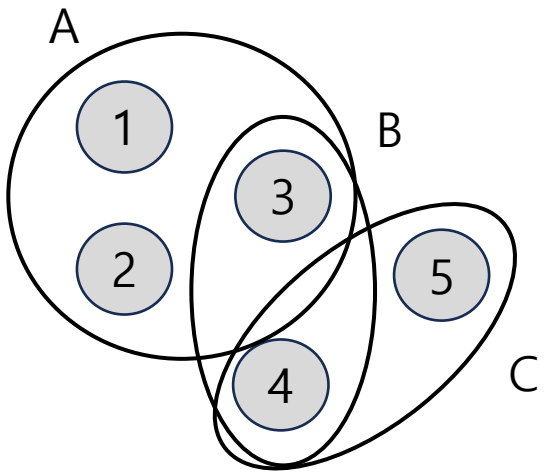


Node (논문)

Node (저자)

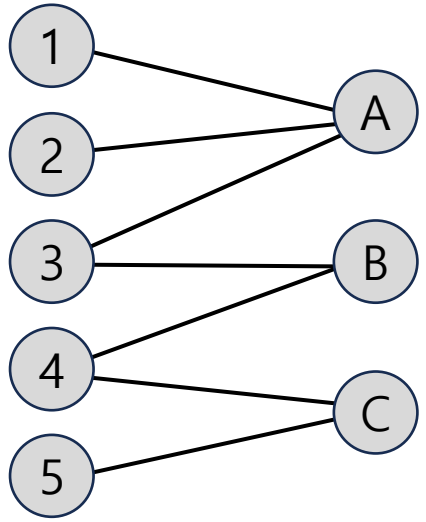
Hypergraph

Hypergraph vs. Bipartite graph



Node (저자)

Hyperedge (논문)



Node (저자)

Node (논문)

Structural Patterns

“What do real-world hypergraphs look like?”

“Given a static hypergraph, how can we analyze its structure?”

-> Structural patterns

그래프를 설명하거나 분석하기 위해 필요

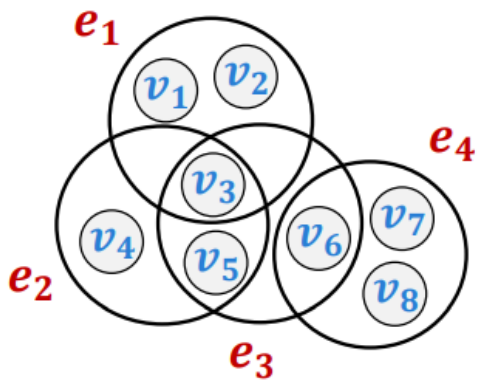
(이 튜토리얼에서는 Generative model이 얼마나 현실적인 Hypergraph를 생성해내는지를 평가하기 위해서 사용됨)

Structural Patterns

Background

Degree of a node: 노드가 포함된 하이퍼엣지의 수

Size of a hyperedge: 하이퍼엣지에 포함된 노드의 수



Hypergraph

Degree of v_5 is 2.

Size of e_2 is 3.

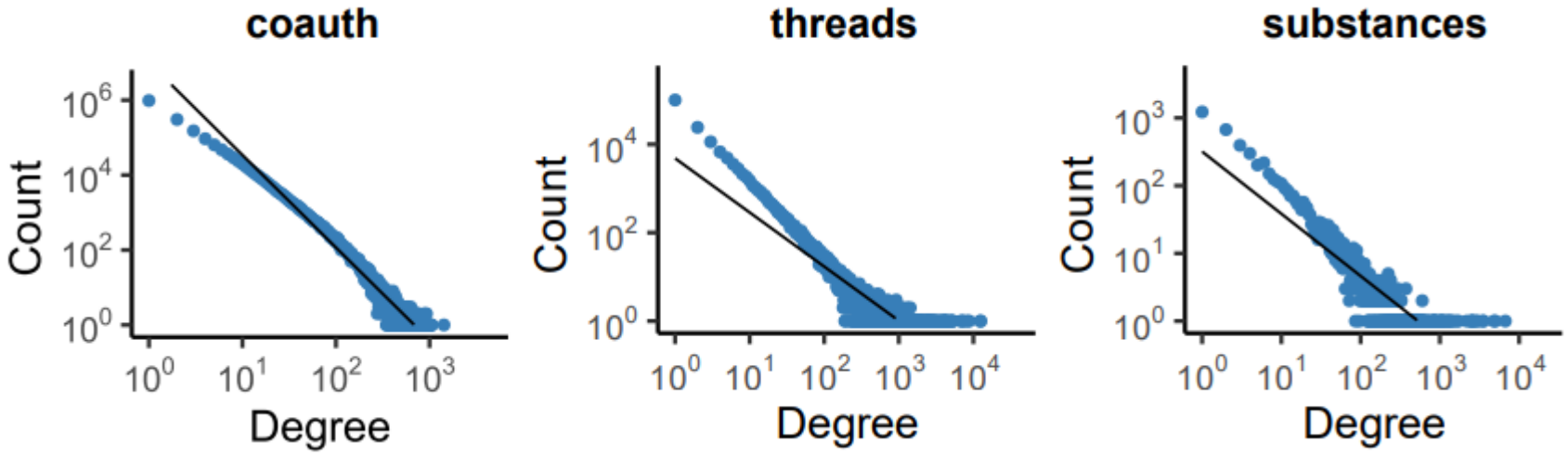
Example

Structural Patterns

차수 분포

각 사람이 얼마나 많은 그룹에 속하는지

Heavy-tailed, 즉 많은 그룹에 속하는 사람일수록 더 많은 그룹에 속하도록 편향됨

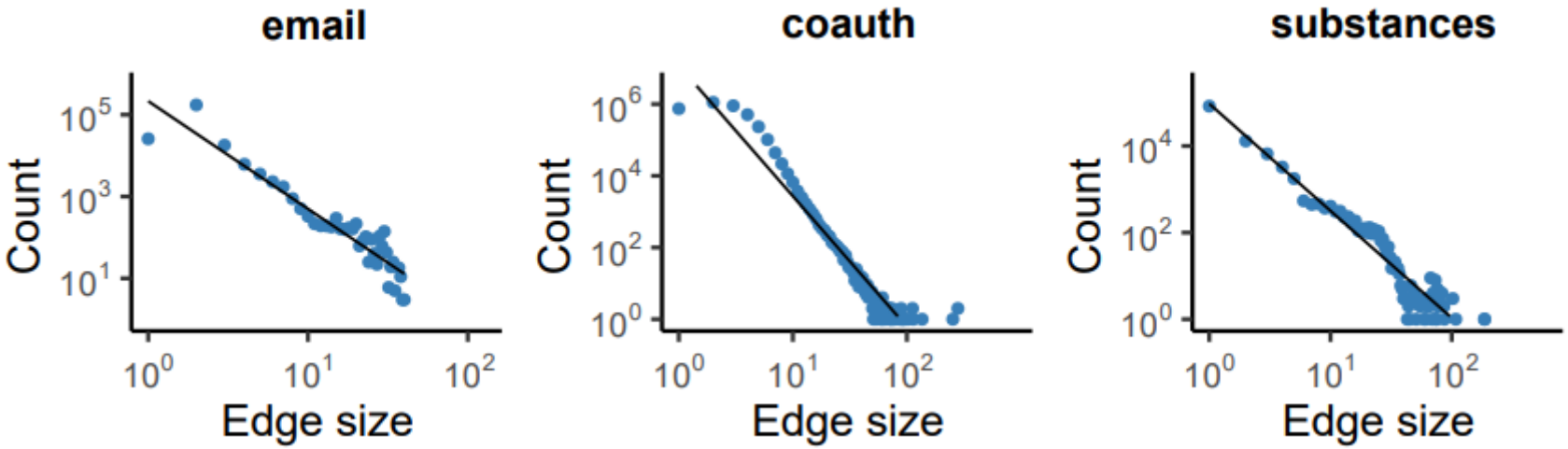


Structural Patterns

Hyperedge Size 분포

각 그룹에 얼마나 많은 사람이 속하는지

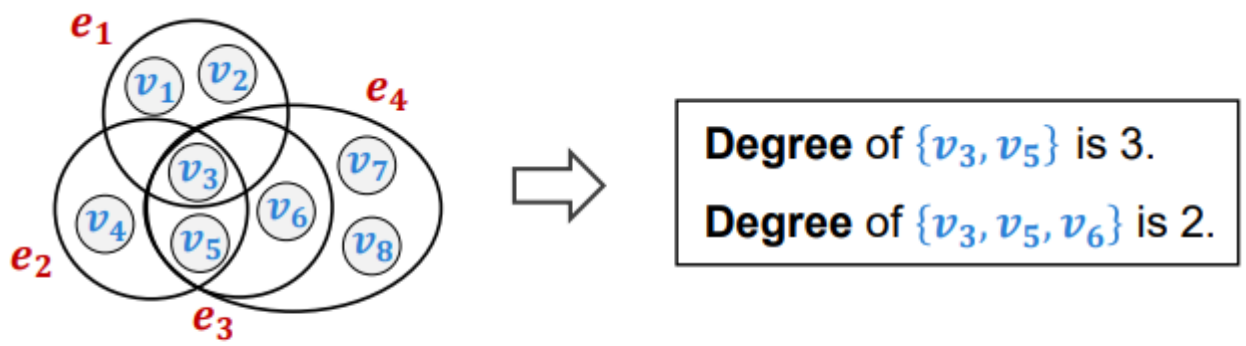
Heavy-tailed, 즉 많은 사람이 속하는 그룹일수록 더 많은 사람이 속하도록 편향됨



Structural Patterns

Pair/Triple Degree의 분포

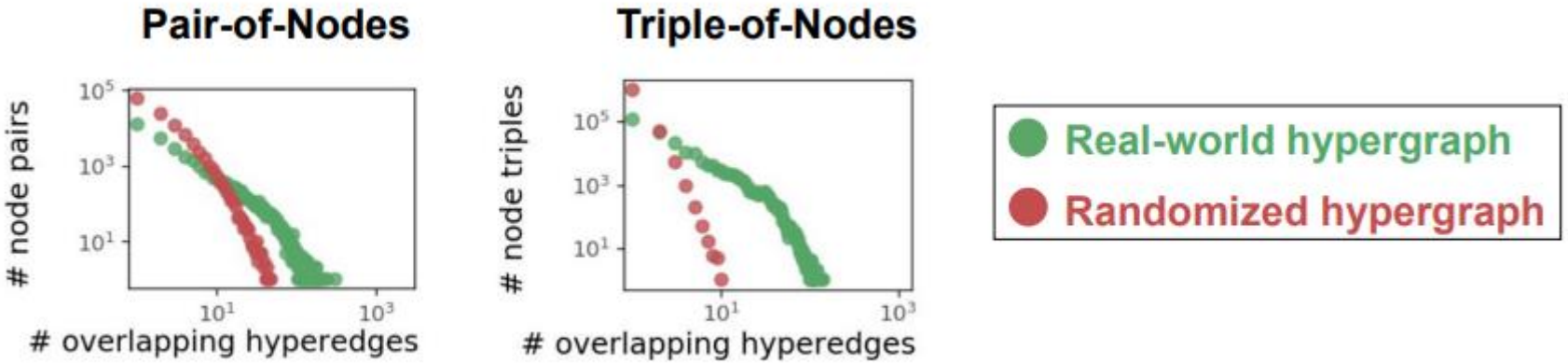
어떤 두/세 노드가 공통으로 overlap되는 하이퍼엣지의 수



Structural Patterns

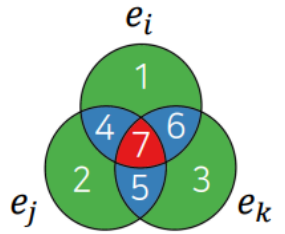
Pair/Triple Degree의 분포

랜덤 데이터에 비해 skew되고 heavy tailed됨



Structural Patterns

Advanced Patterns

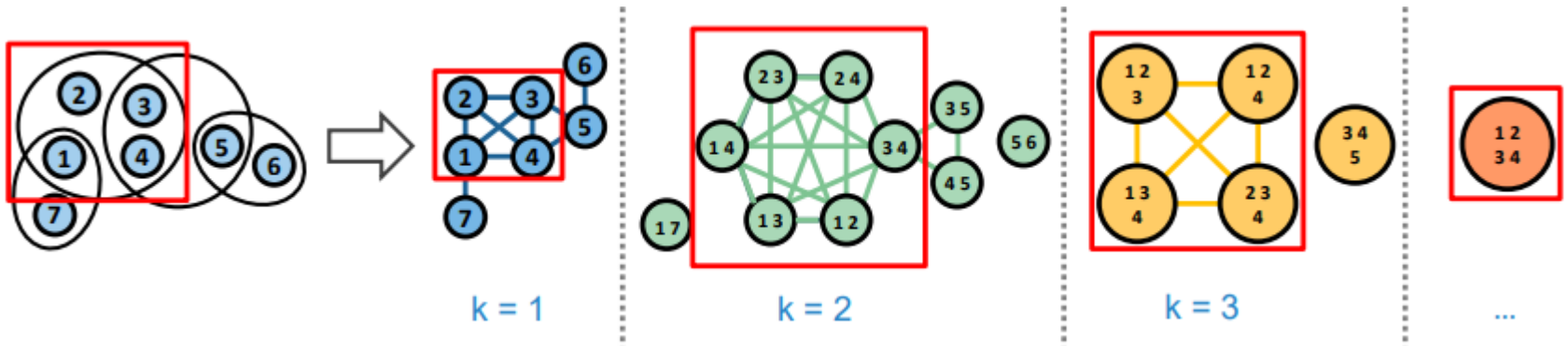


(1) $e_i \backslash e_j \backslash e_k$
(2) $e_j \backslash e_k \backslash e_i$
(3) $e_k \backslash e_i \backslash e_j$
(4) $e_i \cap e_j \backslash e_k$
(5) $e_j \cap e_k \backslash e_i$
(6) $e_k \cap e_i \backslash e_j$
(7) $e_i \cap e_j \cap e_k$

H-motifs: 3개의 하이퍼엣지 간 교집합 구조의 개수를 세서 Graphlet처럼 사용

Multi-level decomposition: 여러 개의 레벨로 나눠서, k레벨이라면 노드를 k개씩 묶은 대체 노드들의 하이퍼그래프로 만들어서 사용

등등 많은 것들이 있지만.. 시간 관계상 생략



Generative Models

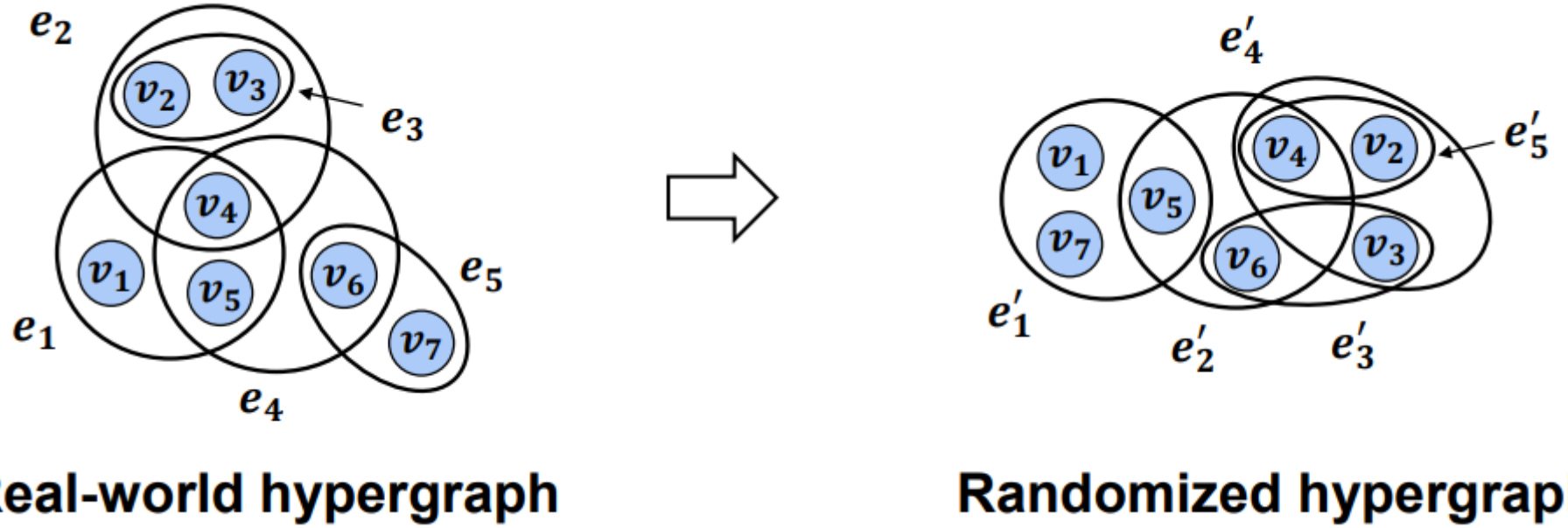
Real-world Hypergraph에 가까운 Hypergraph를 생성하는 것이 목표

생성 모델을 만드는 목적:

- 벤치마크 데이터 생성: 하이퍼그래프에서 동작하는 알고리즘의 벤치마크를 위한 데이터 생성
- 익명화: 민감한 정보를 담고 있는 Real-world 그래프를 유사한 성질을 가진 합성데이터로 변환

Generative Models

Configuration models: Node Degree와 Hyperedge sizes의 분포를 유지

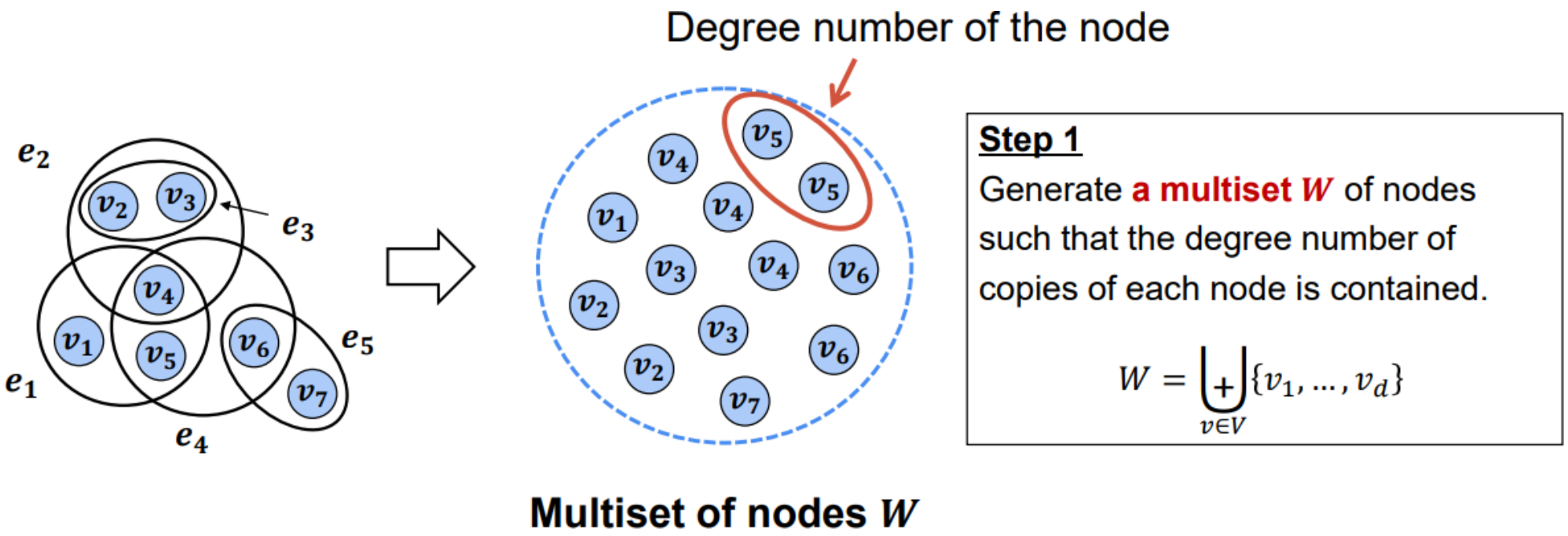


Real-world hypergraph **Randomized hypergraph**

Generative Models

Step 1. 멀티셋 생성

각 노드마다, 차수만큼 복제해 멀티셋에 넣음 -> 차수 유지



Step 1

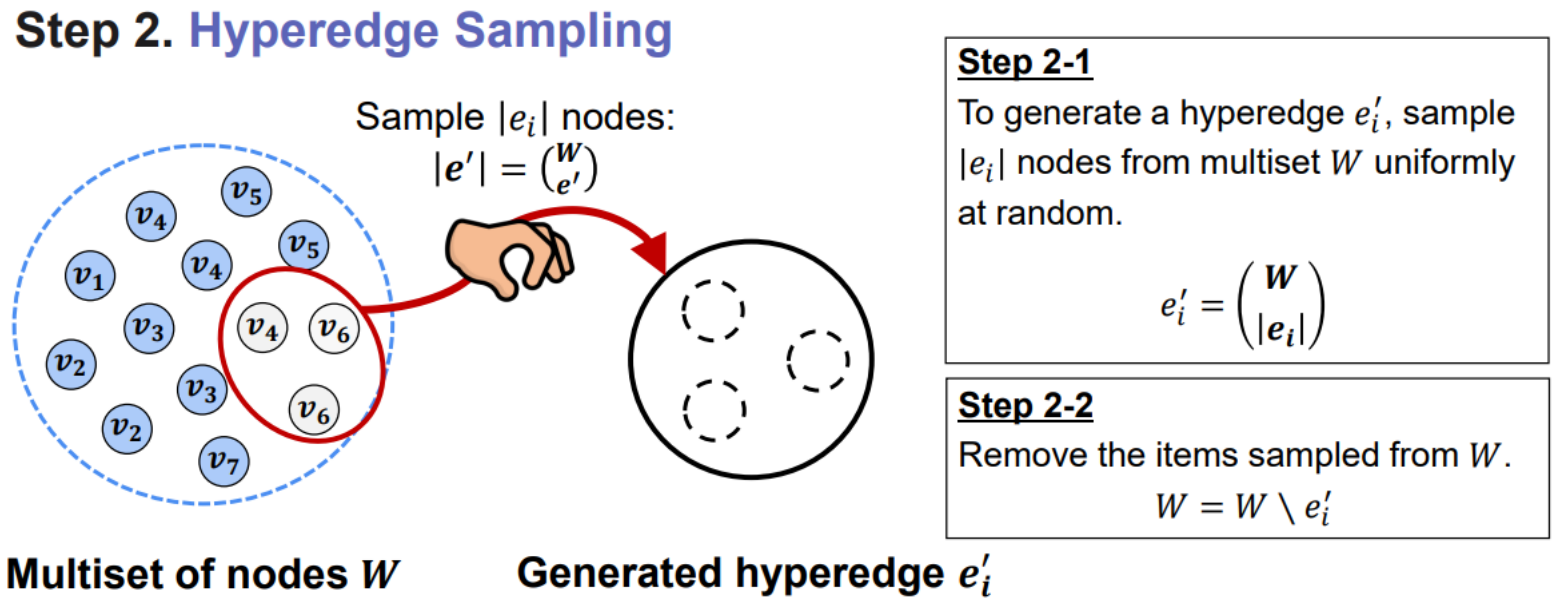
Generate **a multiset W** of nodes such that the degree number of copies of each node is contained.

$$W = \biguplus_{v \in V} \{v_1, \ldots, v_d\}$$

Generative Models

Step 2. Hyperedge Sampling

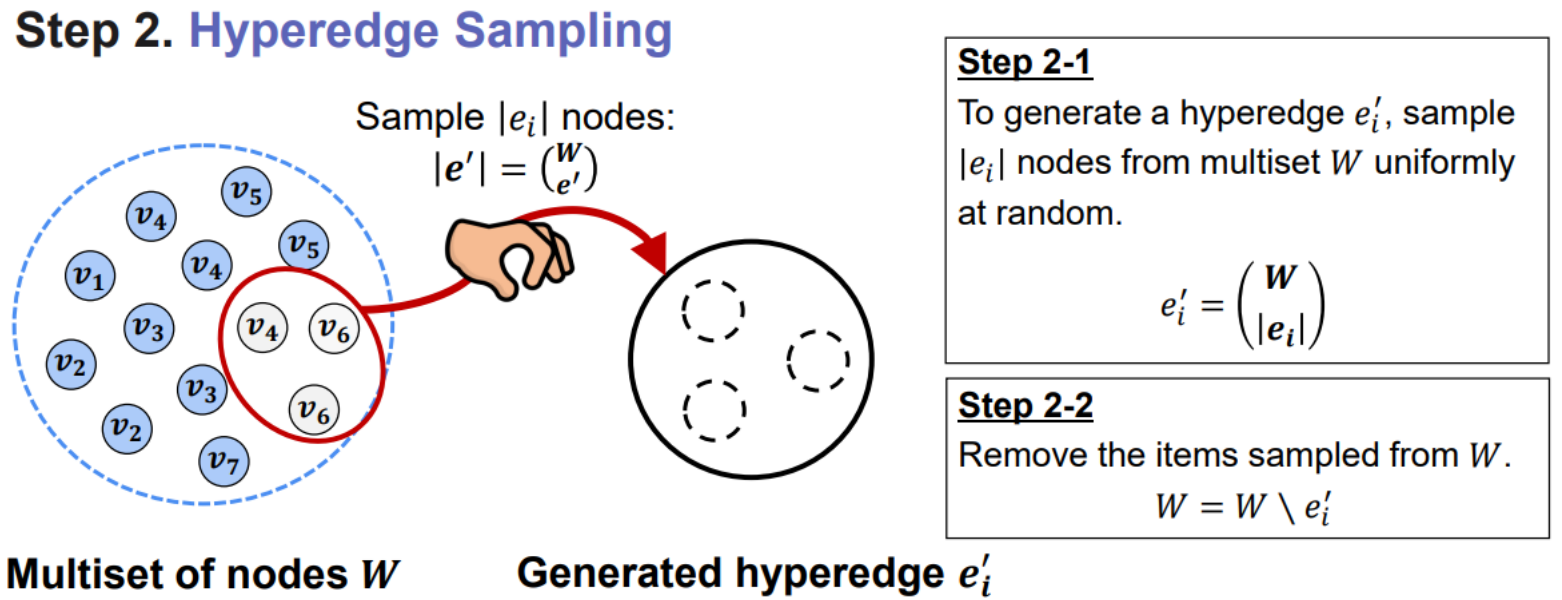
각 Hyperedge마다, Size만큼 샘플링해 가져옴 -> Hyperedge size 유지



Generative Models

Step 2. Hyperedge Sampling

각 Hyperedge마다, Size만큼 샘플링해 가져옴 -> Hyperedge size 유지



Mining of Real-world Hypergraphs: Concepts, Patterns, and Generators

- 0. Introduction — 전반적인 개념 소개 및 설명
- 1. Static Structural Patterns — 구조적 패턴을 통한 Hypergraph 분석
- ~~2. Dynamic Structural Patterns~~ — ~~구조적 패턴을 통한 Time-evolving Hypergraph 분석~~
- 3. Generative Models — Static, ~~Dynamic~~ Hypergraph 생성 모델