2020 IEEE International Conference on Big Data (**IEEE BigData 2020**)

Hypergraph Attention Isomorphism Network by Learning Line Graph Expansion

Sambaran Bandyopadhyay, Kishalay Das, M. N. Murty

IBM Research AI and Indian Institute of Science
Bangalore, India

Email: samb.bandyo@gmail.com

November 27, 2020

Higher Order Associations in Real-World Networks

- Co-authorship, Co-citation, Chemical Reaction, Email Communication relationships are complex and go beyond pairwise associations. A more flexible and natural way to model such complex relationships is using Hyper-Graphs.

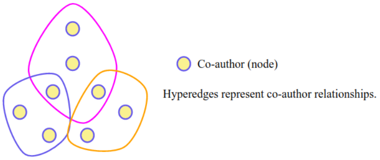


Figure:

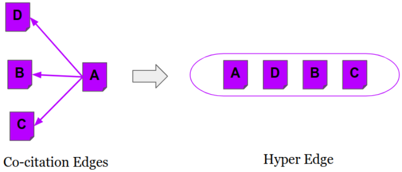
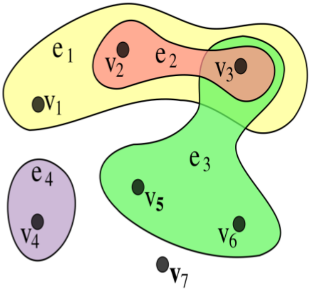


Figure:

Figure: Co-authorship and Co-citation Relationship.

Hyper-Graphs

- Hypergraphs are introduced to model such complex relationships among the real world entities in a graph structure.
- In a hypergraph, each edge may connect more than two nodes. So an hyperedge is essentially denoted by a subset of nodes, rather than just a pair.



$$\mathcal{H} = (V, E)$$
$$E \subseteq 2^V$$

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$$
$$E = \{e_1, e_2, e_3, e_4\}$$

$$e_1 = \{v_1, v_2, v_3\}$$
$$e_2 = \{v_2, v_3\}$$
$$e_3 = \{v_3, v_5, v_6\}$$
$$e_4 = \{v_4\}$$

Image Source:Wikipedia

Problem Formulation

- A hypergraph can be denoted as $\mathcal{H} = (V, E)$, where V is the set of hypernodes with E is the set of hyperedges.

Problem Formulation

- A hypergraph can be denoted as $\mathcal{H} = (V, E)$, where V is the set of hypernodes with E is the set of hyperedges.
- Each node $v \in V$ is associated with a F dimensional feature vector $x_v \in \mathbb{R}^F$ and this forms a feature matrix $X \in \mathbb{R}^{|V| \times F}$.

Problem Formulation

- A hypergraph can be denoted as $\mathcal{H} = (V, E)$, where V is the set of hypernodes with E is the set of hyperedges.
- Each node $v \in V$ is associated with a F dimensional feature vector $x_v \in \mathbb{R}^F$ and this forms a feature matrix $X \in \mathbb{R}^{|V| \times F}$.
- We have a training set $V^s \subset V$ where for each hypernode $v \in V^s$, we know the label $l_v \in \mathcal{L}$, where $\mathcal{L}$ is the set of labels.

Problem Formulation

- A hypergraph can be denoted as $\mathcal{H} = (V, E)$, where V is the set of hypernodes with E is the set of hyperedges.
- Each node $v \in V$ is associated with a F dimensional feature vector $x_v \in \mathbb{R}^F$ and this forms a feature matrix $X \in \mathbb{R}^{|V| \times F}$.
- We have a training set $V^s \subset V$ where for each hypernode $v \in V^s$, we know the label $l_v \in \mathcal{L}$, where $\mathcal{L}$ is the set of labels.
- The incidence matrix of the hypergraph : $H \in \mathbb{R}^{|V| \times |E|}$

Problem Formulation

- A hypergraph can be denoted as $\mathcal{H} = (V, E)$, where V is the set of hypernodes with E is the set of hyperedges.
- Each node $v \in V$ is associated with a F dimensional feature vector $x_v \in \mathbb{R}^F$ and this forms a feature matrix $X \in \mathbb{R}^{|V| \times F}$.
- We have a training set $V^s \subset V$ where for each hypernode $v \in V^s$, we know the label $l_v \in \mathcal{L}$, where $\mathcal{L}$ is the set of labels.
- The incidence matrix of the hypergraph : $H \in \mathbb{R}^{|V| \times |E|}$
- Hypernode diagonal degree matrix of $\mathcal{H}$: $D_V \in \mathbb{R}^{|V| \times |V|}$, where $D_V(i, i) = \sum_{j=1}^{|E|} H_{i,j}$.

Problem Formulation

- A hypergraph can be denoted as $\mathcal{H} = (V, E)$, where V is the set of hypernodes with E is the set of hyperedges.
- Each node $v \in V$ is associated with a F dimensional feature vector $x_v \in \mathbb{R}^F$ and this forms a feature matrix $X \in \mathbb{R}^{|V| \times F}$.
- We have a training set $V^s \subset V$ where for each hypernode $v \in V^s$, we know the label $l_v \in \mathcal{L}$, where $\mathcal{L}$ is the set of labels.
- The incidence matrix of the hypergraph : $H \in \mathbb{R}^{|V| \times |E|}$
- Hypernode diagonal degree matrix of $\mathcal{H}$: $D_V \in \mathbb{R}^{|V| \times |V|}$, where $D_V(i,i) = \sum_{j=1}^{|E|} H_{i,j}$.
- Hyperedge diagonal degree matrix of the hypergraph : $D_E \in \mathbb{R}^{|E| \times |E|}$, where $D_E(j,j) = \sum_{i=1}^{|V|} H_{i,j}$.

Problem Formulation

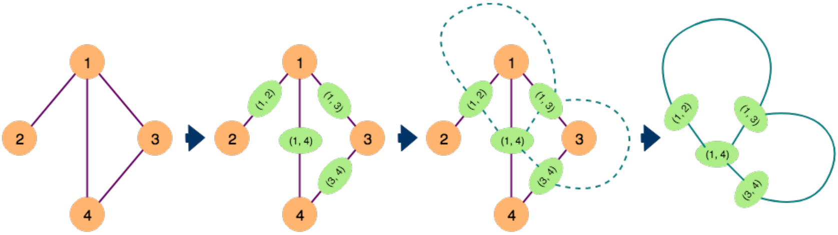
- **Goal** :
  - Develop a novel graph neural network to operate on hypergraphs, even with varying hyperedge sizes and generate hypernode embeddings.
  - We want to train the network on semi-supervised hypernode classification:
    learning a function $f : V \mapsto \mathcal{L}$ which can output label of each unlabelled hypernode $v \in V^u = V \setminus V^s$.

Problem Formulation

- **Goal** :
  - Develop a novel graph neural network to operate on hypergraphs, even with varying hyperedge sizes and generate hypernode embeddings.
  - We want to train the network on semi-supervised hypernode classification:
    learning a function $f : V \mapsto \mathcal{L}$ which can output label of each unlabelled hypernode $v \in V^u = V \setminus V^s$.
- **We proposed HAIN, a novel GNN algorithm for hypergraphs exploiting the concept of line graph in hypergraph.**

Line Graph of a Simple Graph

- Given a simple undirected graph $G = (V, E)$, the line graph $L(G)$ is the graph such that each node of $L(G)$ is an edge in G and two nodes of $L(G)$ are neighbors if and only if their corresponding edges in G share a common endpoint vertex.



(a) The Graph G (b) Vertices in L(G) added (c) Edges of L(G) added (d) The line graph L(G)

Hypergraph **A**ttention **I**somorphism **N**etwork

- However, computation of line graph of a hypergraph needs $O(|E|^2|V|)$ time - **not scalable**!

- However, computation of line graph of a hypergraph needs $O(|E|^2|V|)$ time - **not scalable**!
- So, our solution HAIN uses a simple trick to avoid the explicit computation of the line graph.

Hypergraph Attention Isomorphism Network

- However, computation of line graph of a hypergraph needs $O(|E|^2|V|)$ time - **not scalable**!
- So, our solution HAIN uses a simple trick to avoid the explicit computation of the line graph.
- HAIN also learns the edge importance through an attention mechanism.

- Adjacency matrix of the line graph of the given hypergraph:

$$A_L = D_E^{-1} H^T D_V^{-1} H \ \in \mathbb{R}^{|E| \times |E|}$$

- Adjacency matrix of the line graph of the given hypergraph:

$$A_L = D_E^{-1} H^T D_V^{-1} H \ \in \mathbb{R}^{|E| \times |E|}$$

- Updating the hypernode features:

$$X^{l+1} = \sigma(H(A_L H^T X^l + \epsilon^l H^T X^l) W^l) = \sigma(H(A_L X_L^l + \epsilon^l X_L^l) W^l)$$

- Adjacency matrix of the line graph of the given hypergraph:

$$A_L = D_E^{-1} H^T D_V^{-1} H \ \in \mathbb{R}^{|E| \times |E|}$$

- Updating the hypernode features:

$$X^{l+1} = \sigma(H(A_L H^T X^l + \epsilon^l H^T X^l) W^l) = \sigma(H(A_L X_L^l + \epsilon^l X_L^l) W^l)$$

- Introducing the self-attention layer:

$$X^{l+1} = \sigma\Big(H\big(\underbrace{A_L \mathcal{D}(\sigma_{att}(X_L^l \theta_{att}^l))}_{\text{Scaled adjacency matrix}} X_L^l \ + \ \epsilon^l X_L^l\big) W^l\Big) \ \in \mathbb{R}^{|V| \times F^{l+1}}$$

- Adjacency matrix of the line graph of the given hypergraph:

$$A_L = D_E^{-1} H^T D_V^{-1} H \ \in \mathbb{R}^{|E| \times |E|}$$

- Updating the hypernode features:

$$X^{l+1} = \sigma(H(A_L H^T X^l + \epsilon^l H^T X^l) W^l) = \sigma(H(A_L X_L^l + \epsilon^l X_L^l) W^l)$$

- Introducing the self-attention layer:

$$X^{l+1} = \sigma\Big(H\big(\underbrace{A_L \mathcal{D}(\sigma_{att}(X_L^l \theta_{att}^l))}_{\text{Scaled adjacency matrix}} X_L^l \ + \ \epsilon^l X_L^l\big) W^l\Big) \ \in \mathbb{R}^{|V| \times F^{l+1}}$$

- Final update rule of lth layer of HAIN:

$$\boxed{X^{l+1} = \sigma\Big(H\Big(D_E^{-1} H^T D_V^{-1} H \mathcal{D}\big(\sigma_{att}(H^T X^l \theta_{att}^l)\big) H^T X^l \ + \ \epsilon^l H^T X^l\Big) W^l\Big)}$$

- Adjacency matrix of the line graph of the given hypergraph:

$$A_L = D_E^{-1} H^T D_V^{-1} H \ \in \mathbb{R}^{|E| \times |E|}$$

- Updating the hypernode features:

$$X^{l+1} = \sigma(H(A_L H^T X^l + \epsilon^l H^T X^l) W^l) = \sigma(H(A_L X_L^l + \epsilon^l X_L^l) W^l)$$

- Introducing the self-attention layer:

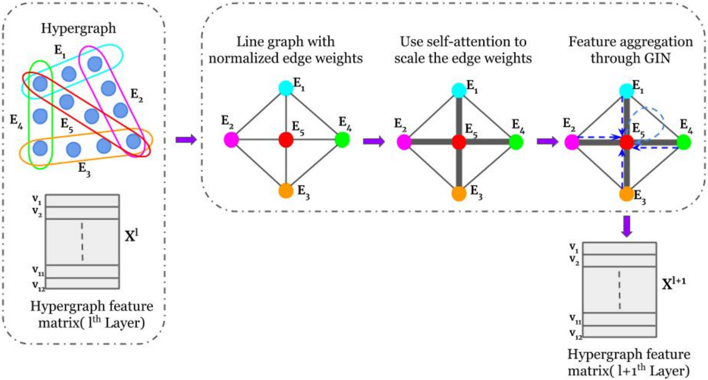
$$X^{l+1} = \sigma\Big(H\big(\underbrace{A_L \mathcal{D}(\sigma_{att}(X_L^l \theta_{att}^l))}_{\text{Scaled adjacency matrix}} X_L^l \ + \ \epsilon^l X_L^l\big) W^l\Big) \ \in \mathbb{R}^{|V| \times F^{l+1}}$$

- Final update rule of lth layer of HAIN:

$$\boxed{X^{l+1} = \sigma\Big(H\Big(D_E^{-1} H^T D_V^{-1} H \mathcal{D}\big(\sigma_{att}(H^T X^l \theta_{att}^l)\big) H^T X^l \ + \ \epsilon^l H^T X^l\Big) W^l\Big)}$$

- Each layer of HAIN takes $O(|V||E|)$ time.

Hypergraph Attention Isomorphism Network



Experimental Setup

	Cora (co-citation)	Citeseer (co-citation)	Pubmed (co-citation)	DBLP (Co-authorship)
No of hypernodes	2708	3312	19717	43413
No of hyperedges	1579	1079	7963	22535
Avg. hyperedge size	3.0 ± 1.1	3.2 ± 2.0	4.3 ± 5.7	4.7 ± 6.1
No of features	1433	3703	500	1425
No of Classes	7	6	3	6

Table: Co-citation and co-authorship hypergraph datasets used in this work.

Model Ablation Study:

- Star-GIN
- Clique-GIN
- Static-HAIN

Hypernode Classification

Method	Cora (co-citation)	Citeseer (co-citation)	Pubmed (co-citation)	DBLP (Co-authorship)
CI	35.60 ± 0.8	29.63 ± 0.3	47.04 ± 0.8	45.19 ± 0.9
MLP	57.86 ± 1.8	58.88 ± 1.7	69.30 ± 1.6	62.23 ± 2.0
MLP+HLR	63.02 ± 1.8	62.25 ± 1.6	69.82 ± 1.5	69.58 ± 2.1
HGNN	67.59 ± 1.8	62.60 ± 1.6	70.59 ± 1.5	74.35 ± 2.1
DHGNN	78.8 ± 1.25	63.45 ± 1.17	71.3 ± 1.33	74.65 ± 1.85
1-HyperGCN	65.55 ± 2.1	61.13 ± 1.9	69.92 ± 1.5	66.13 ± 2.4
FastHyperGCN	67.57 ± 1.8	62.58 ± 1.7	70.52 ± 1.6	72.66 ± 2.1
HyperGCN	67.63 ± 1.7	62.65 ± 1.6	74.44 ± 1.6	75.91 ± 2.0
Star-GIN	75.19 ± 1.41	64.17 ± 0.73	76.91 ± 0.67	76.71 ± 0.85
Clique-GIN	76.38 ± 1.24	64.23 ± 0.95	74.59 ± 0.83	77.23 ± 0.97
Static-HAIN	77.17 ± 1.17	66.51 ± 0.83	76.25 ± 0.77	79.13 ± 0.95
HAIN	**80.15 ± 1.71**	**68.89 ± 0.90**	**79.60 ± 0.67**	**81.69 ± 0.70**

Table: Results of hypernode classification (accuracy with standard deviation in %).

Analysis on Training

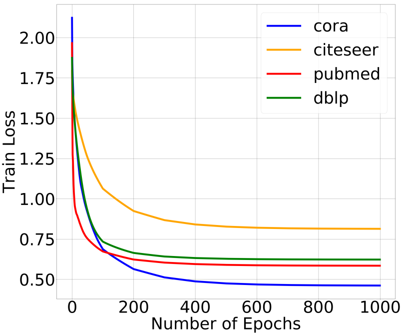


Figure:

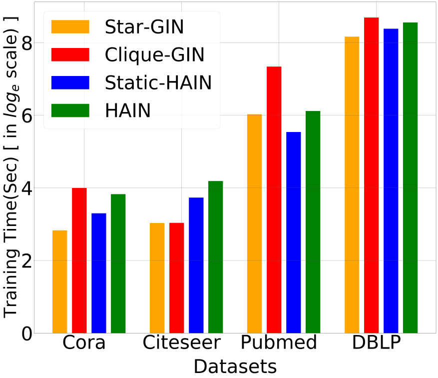


Figure:

Figure: (a) Shows HAIN training Loss over different epochs of the algorithm. (b) Training time of different approaches on four datasets.

Sensitivity Analysis

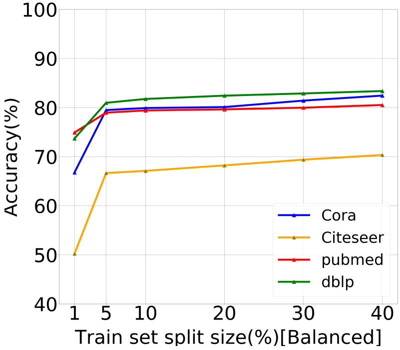


Figure:

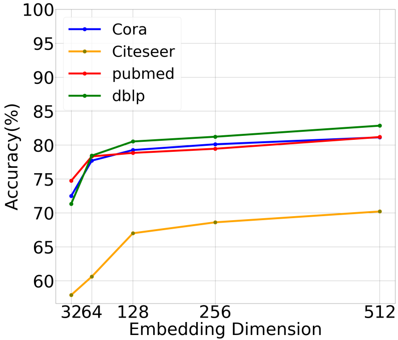


Figure:

Figure: (a) Shows node classification accuracy over different train-test sizes, (b) Shows node classification accuracy over different embedding dimensions.

Conclusions

- We proposed a **novel GNN algorithm**, referred as HAIN, which operates directly on **hypergraphs** and can handle varying edge sizes.

Conclusions

- We proposed a **novel GNN algorithm**, referred as HAIN, which operates directly on **hypergraphs** and can handle varying edge sizes.
- HAIN **implicitly** formed a **line graph** from the hypergraph to make it **computational efficient**.

Conclusions

- We proposed a **novel GNN algorithm**, referred as HAIN, which operates directly on **hypergraphs** and can handle varying edge sizes.
- HAIN **implicitly** formed a **line graph** from the hypergraph to make it **computational efficient**.
- HAIN uses an **attention mechanism** to learn the hyperedge relationships.

Conclusions

- We proposed a **novel GNN algorithm**, referred as HAIN, which operates directly on **hypergraphs** and can handle varying edge sizes.
- HAIN **implicitly** formed a **line graph** from the hypergraph to make it **computational efficient**.
- HAIN uses an **attention mechanism** to learn the hyperedge relationships.
- HAIN shows promising results for **hypernode classification** both in quality and computational time.

Conclusions

- We proposed a **novel GNN algorithm**, referred as HAIN, which operates directly on **hypergraphs** and can handle varying edge sizes.
- HAIN **implicitly** formed a **line graph** from the hypergraph to make it **computational efficient**.
- HAIN uses an **attention mechanism** to learn the hyperedge relationships.
- HAIN shows promising results for **hypernode classification** both in quality and computational time.
- **Model ablation study** shows the usefulness of each component of HAIN through experiments.

Thank You!

Any Questions?