temp

Jun Ha Chun

Seoul National University

temp

Jun Ha Chun

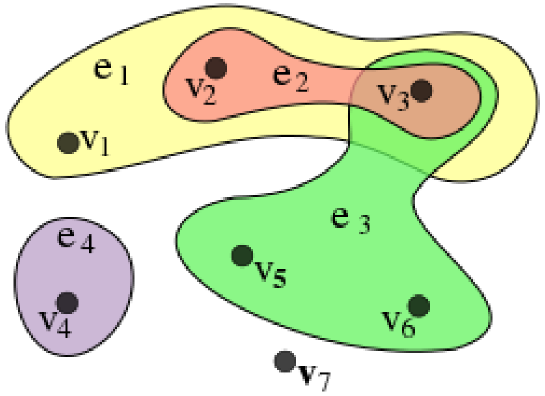
Seoul National University

HyperGCN:
A New Method of Training Graph Convolutional Networks on Hypergraphs

Jun Ha Chun

Seoul National University

Introduction - Hypergraph



- Generalization of a graph
  - Hyperedge can join any number of nodes
  - Examples
    - Co-citation
    - Co-authorship
    - 3D point cloud
  - Tasks (like graphs)
    - SSL
    - Combinatorial optimization
- Normalized Hypergraph Cut

$$\operatorname*{argmin}_{\emptyset \neq S \subset V} c(S) := \operatorname{vol} \partial S \left(\frac{1}{\operatorname{vol} S} + \frac{1}{\operatorname{vol} S^c} \right)$$

- Relaxation

$$\operatorname*{argmin}_{f \in \mathbb{R}^{|V|}} \frac{1}{2} \sum_{e \in E} \sum_{\{u,v\} \subseteq e} \frac{w(e)}{\delta(e)} \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}} \right)^2 = f^T \Delta f.$$

$$\text{subject to} \sum_{v \in V} f^2(v) = 1, \ \sum_{v \in V} f(v) \sqrt{d(v)} = 0.$$

$$\Theta = D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2} \text{ and } \Delta = I - \Theta$$

- Hypergraph convolution(HGNN)
  - $\Delta = I - D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}$

$$\mathbf{g} \star \mathbf{x} = \mathbf{\Phi}((\mathbf{\Phi}^\top \mathbf{g}) \odot (\mathbf{\Phi}^\top \mathbf{x})) = \mathbf{\Phi} g(\mathbf{\Lambda}) \mathbf{\Phi}^\top \mathbf{x} \approx \sum_{k=0}^{K} \theta_k T_k(\tilde{\mathbf{\Delta}}) \mathbf{x},$$

$$\approx \theta_0 \mathbf{x} - \theta_1 \mathbf{D}_v^{-1/2} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^\top \mathbf{D}_v^{-1/2} \mathbf{x},$$

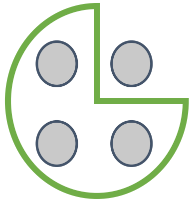
$$\approx \frac{1}{2} \theta \mathbf{D}_v^{-1/2} \mathbf{H} (\mathbf{W} + \mathbf{I}) \mathbf{D}_e^{-1} \mathbf{H}^\top \mathbf{D}_v^{-1/2} \mathbf{x}$$

$$\approx \theta \mathbf{D}_v^{-1/2} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^\top \mathbf{D}_v^{-1/2} \mathbf{x},$$

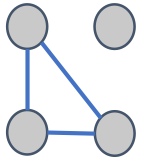
$$\mathbf{Y} = \mathbf{D}_v^{-1/2} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^\top \mathbf{D}_v^{-1/2} \mathbf{X} \mathbf{\Theta},$$

Backgrounds - Hypergraph

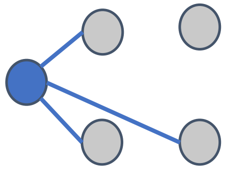
- Hypergraph approximation
  - Hypergraph to graph
  - Clique expansion
    - Hyperedge to Clique
      - $O(s^2)$ edges
  - Star expansion
    - Hyperedge to
      - a new vertex
      - edges connecting the new vertex to each vertex in the hyperedge
      - $O(s)$ edges



Hyperedge

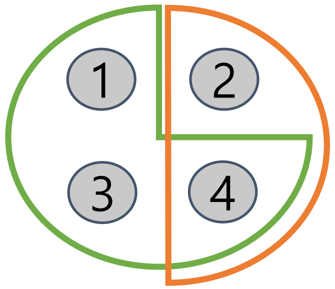


Clique ex.



Star ex.

- Hypergraph approximation
  - $\Delta = I - D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}$
  - $L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$



$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

$$HH^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix}$$

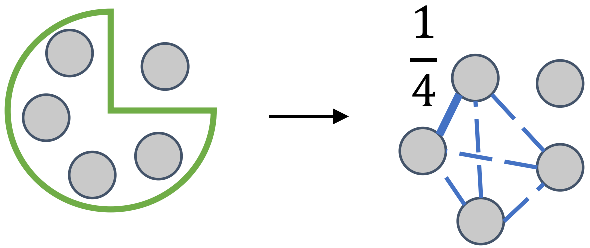
- Hypergraph Laplacian implies clique expansion
  - Inefficient when hyperedges are large

- Let's reduce order of edges added for each hyperedge
  - Explicit hyperedge expansion
  - Select all edge = Clique
  - Select some edges?
  - Select 1 edge?

HyperGCN

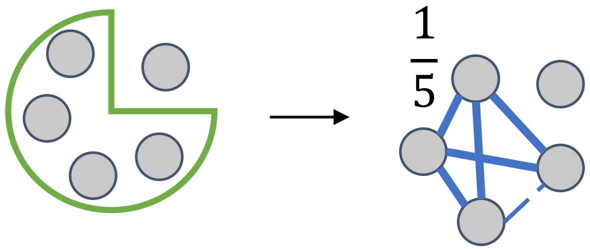
- What should we learn?
  - SSL node classification: nodes in hyperedge = similar
  - Smoothness regularizer: $\sum_{e \in E} \max_{i,j \in e} \left| h_i - h_j \right|^2$
  - Let's select with $\underset{i,j \in e}{argmax} \left| h_i - h_j \right|^2$
  - Edges with large difference should be "learned" more

- **1-HyperGCN** (select 1 edge)
  - $G_S = \{V, E_S\}, E_S = \left\{ \underset{i,j \in e}{argmax} \left| h_i - h_j \right|^2 : e \in E \right\} + self\ edges$
  - $w(\{i_e, j_e\}) = \frac{1}{|e|}$, $A_S = weighted\ adjacency\ matrix$
  - GCN step: $\sigma(\overline{A_S} X \Theta)$



- One is not enough
  - ignores nodes in $K_e = \{k \in e : k \neq i_e, k \neq j_e\}$
  - Enhance by using them as mediators

- **HyperGCN** (Select more)
  - $G_S = \{V, E_S\}, E_S = \left\{ \underset{i,j \in e}{argmax} \left| h_i - h_j \right|^2 : e \in E \right\}$
    $+ \{(k, l) : l \in \{i_e, j_e\}, k \in K_e,\ e \in E\} + self\ edges$
  - $w(\{i_e, j_e\}) = \frac{1}{2|e|-3}$, $A_S = weighted\ adjacency\ matrix$
  - GCN step: $\sigma(\overline{A_S} X \Theta)$



- G_S is recomputed every epoch
  - To learn different structures as parameter changes

- **FastHyperGCN**
  - Use initial features X to construct fixed Laplacian

Experiments

- Dataset: Co-citation/authorship hypergraph

	DBLP (co-authorship)	Pubmed (co-citation)	Cora (co-authorship)	Cora (co-citation)	Citeseer (co-citation)
# hypernodes, $\|V\|$	43413	19717	2708	2708	3312
# hyperedges, $\|E\|$	22535	7963	1072	1579	1079
avg.hyperedge size	**4.7 ± 6.1**	**4.3 ± 5.7**	**4.2 ± 4.1**	**3.0 ± 1.1**	**3.2 ± 2.0**
# features, d	1425	500	1433	1433	3703
# classes, q	6	3	7	7	6
label rate, $\|V_L\|/\|V\|$	0.040	0.008	0.052	0.052	0.042

- SSL: Hypernode classification task
  - Clique expansion = expansion with mediators when $|e| = 2, 3$

Data	Method	DBLP co-authorship	Pubmed co-citation	Cora co-authorship	Cora co-citation	Citeseer co-citation
$\mathcal{H}$	CI	54.81 ± 0.9	52.96 ± 0.8	55.45 ± 0.6	64.40 ± 0.8	70.37 ± 0.3
$\mathbf{X}$	MLP	37.77 ± 2.0	30.70 ± 1.6	41.25 ± 1.9	42.14 ± 1.8	41.12 ± 1.7
$\mathcal{H}, \mathbf{X}$	MLP + HLR	30.42 ± 2.1	30.18 ± 1.5	34.87 ± 1.8	36.98 ± 1.8	37.75 ± 1.6
$\mathcal{H}, \mathbf{X}$	HGNN	25.65 ± 2.1	29.41 ± 1.5	31.90 ± 1.9	**32.41 ± 1.8**	**37.40 ± 1.6**
$\mathcal{H}, \mathbf{X}$	1-HyperGCN	33.87 ± 2.4	30.08 ± 1.5	36.22 ± 2.2	34.45 ± 2.1	38.87 ± 1.9
$\mathcal{H}, \mathbf{X}$	FastHyperGCN	27.34 ± 2.1	29.48 ± 1.6	32.54 ± 1.8	**32.43 ± 1.8**	**37.42 ± 1.7**
$\mathcal{H}, \mathbf{X}$	HyperGCN	**24.09 ± 2.0**	**25.56 ± 1.6**	**30.08 ± 1.8**	**32.37 ± 1.7**	**37.35 ± 1.6**

  - Robust to noisiness

Method	$\eta = 0.75$	$\eta = 0.70$	$\eta = 0.65$	$\eta = 0.60$	$\eta = 0.55$	$\eta = 0.50$	sDBLP
HGNN	**15.92 ± 2.4**	**24.89 ± 2.2**	**31.32 ± 1.9**	39.13 ± 1.78	42.23 ± 1.9	44.25 ± 1.8	45.27 ± 2.4
FastHyperGCN	28.86 ± 2.6	31.56 ± 2.7	33.78 ± 2.1	33.89 ± 2.0	**34.56 ± 2.2**	**35.65 ± 2.1**	41.79 ± 2.8
HyperGCN	22.44 ± 2.0	29.33 ± 2.2	33.41 ± 1.9	**33.67 ± 1.9**	35.05 ± 2.0	37.89 ± 1.9	**41.64 ± 2.6**

  - Training time

Model↓ Metric →	Training time	Density	DBLP	Pubmed
HGNN	170s	337	0.115s	0.019s
FastHyperGCN	**143s**	**352**	**0.035s**	**0.016s**

Table 1: average training time of an epoch (lower is better)

Experiments

- Combinatorial Optimization: K-subhypergraph problem
  - Maxmizing “density”
  - Greedy heuristic
    - MaxDegree: select k nodes with largest degree
    - RemoveMinDegree: remove all hyperedge including smallest degree node (repeat n-k times)

Dataset→ Approach↓	**Synthetic** test set	**DBLP** co-authorship	**Pubmed** co-citation	**Cora** co-authorship	**Cora** co-citation	**Citeseer** co-citation
MaxDegree	174 ± 50	4840	1306	194	544	507
RemoveMinDegree	147 ± 48	**7714**	**7963**	450	1369	843
MLP	174 ± 56	5580	1206	238	550	534
MLP + HLR	231 ± 46	5821	3462	297	952	764
HGNN	337 ± 49	6274	7865	437	**1408**	**969**
1-HyperGCN	207 ± 52	5624	1761	251	563	509
FastHyperGCN	352 ± 45	7342	7893	452	**1419**	**969**
HyperGCN	$\mathbf{359 \pm 49}$	**7720**	**7928**	**504**	**1431**	**971**
# hyperedges, $\lvert E \rvert$	500	22535	7963	1072	1579	1079

Thank you