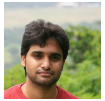
HyperGraph Convolutional Network (HyperGCN)

To Appear as a Poster in Neural Information Processing Systems, 2019

HyperGraph Convolutional Network (HyperGCN)

To Appear as a Poster in Neural Information Processing Systems, 2019

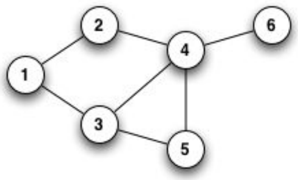




Joint work with Madhav Prateek Vikram Prof. Anand Louis Prof. Partha Talukdar

Motivation

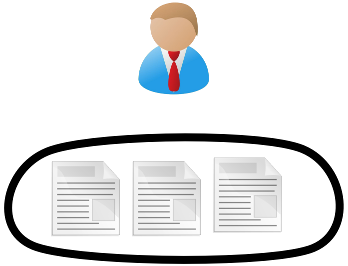
networks have complex relationships



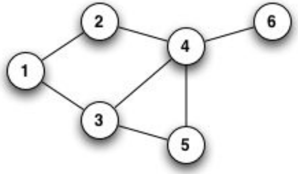
simple graphs do not capture such relationships

Motivation

networks have complex relationships



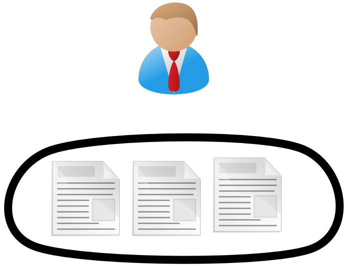
co-authorship



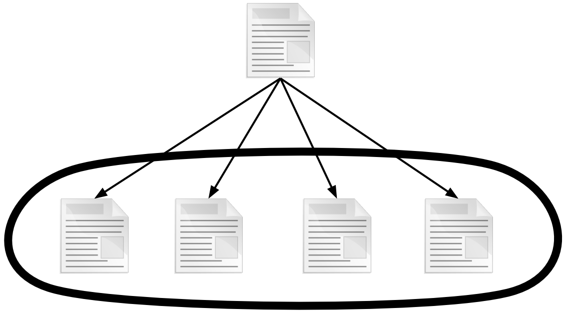
simple graphs do not capture such relationships

Motivation

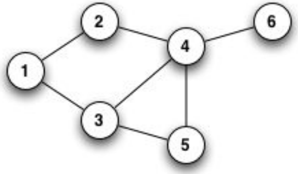
networks have complex relationships



co-authorship



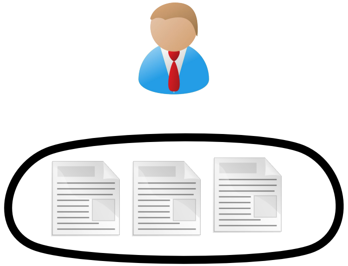
co-citation



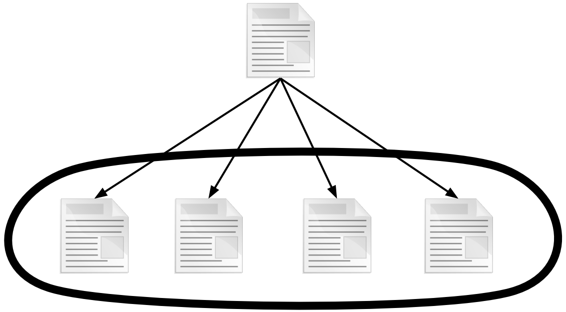
simple graphs do not capture such relationships

Motivation

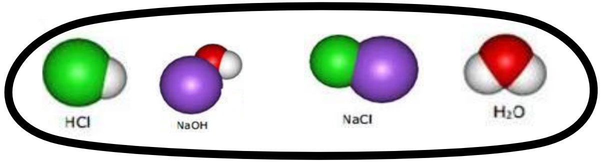
networks have complex relationships



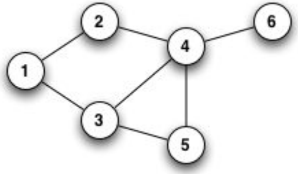
co-authorship



co-citation

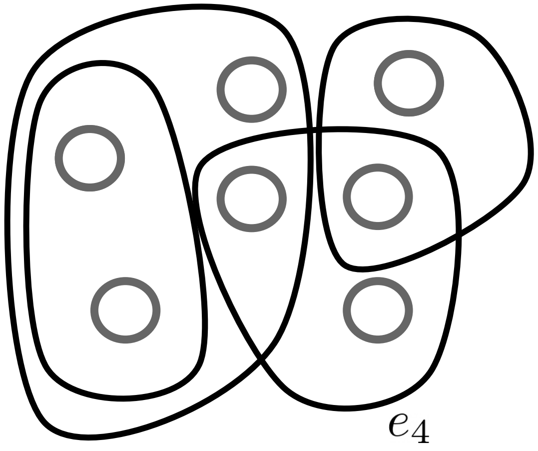


chemical reaction



simple graphs do not capture such relationships

Hypergraph

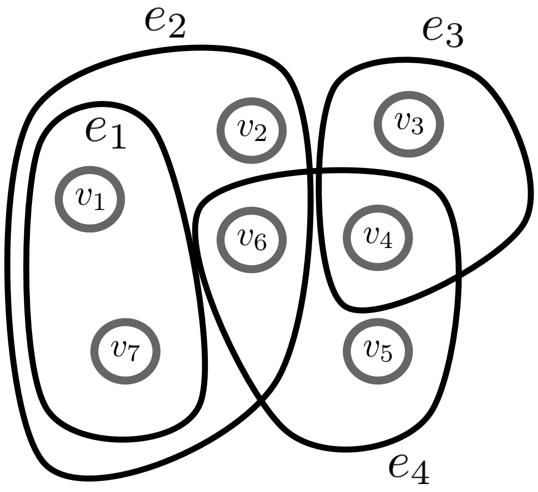


$$\mathcal{H} = (V, E)$$

$$E \subseteq 2^V$$

an edge can connect any number of vertices

Hypergraph



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

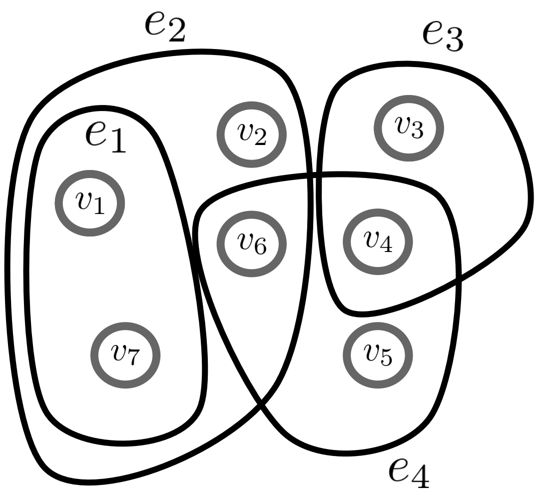
$$E \subseteq 2^V$$

an edge can connect any number of vertices

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$$

$$E = \{e_1, e_2, e_3, e_4\}$$

Hypergraph



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

$$E \subseteq 2^V$$

an edge can connect any number of vertices

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$$

$$E = \{e_1, e_2, e_3, e_4\}$$

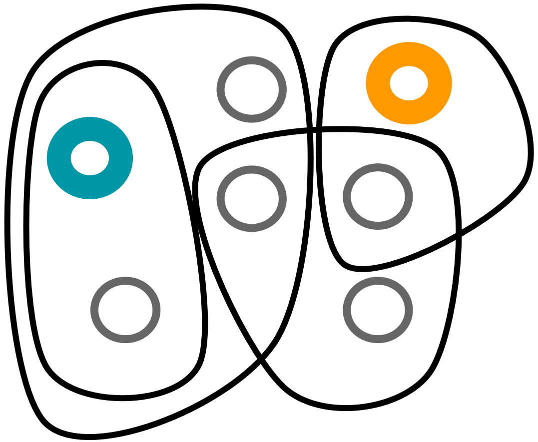
$$e_1 = \{v_1, v_7\}$$

$$e_2 = \{v_1, v_2, v_6, v_7\}$$

$$e_3 = \{v_3, v_4\}$$

$$e_4 = \{v_4, v_5, v_6\}$$

Hypergraph-based Semi-Supervised Learning (SSL)

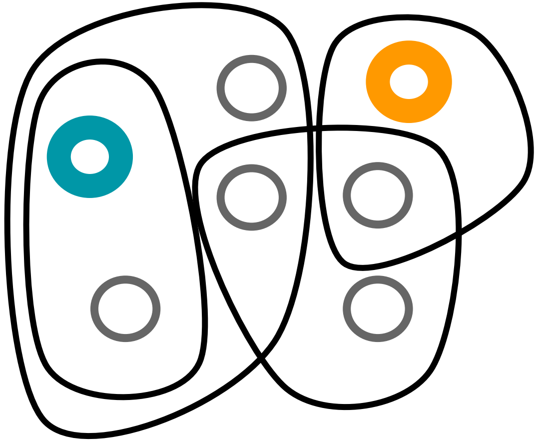


$$\mathcal{H} = (V, E)$$

$$E \subseteq 2^V$$

use labelled and unlabelled data for training

Hypergraph-based Semi-Supervised Learning (SSL)



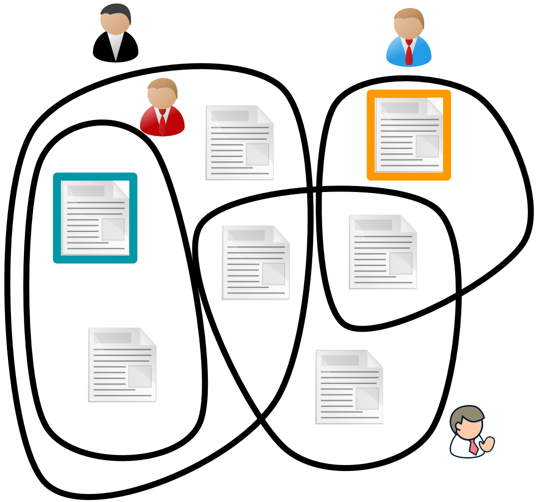
$$\mathcal{H} = (V, E)$$

$$E \subseteq 2^V$$

use labelled and unlabelled data for training

expensive cheap

Hypergraph-based Semi-Supervised Learning (SSL)



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

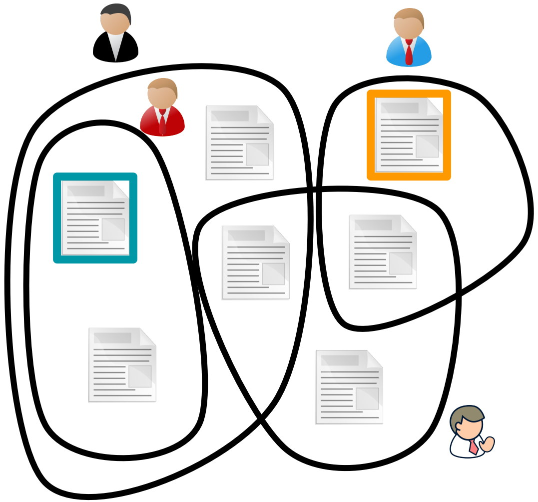
$$E \subseteq 2^V$$

use labelled and unlabelled data for training

expensive cheap

e.g. document classification in co-authorship

Hypergraph-based Semi-Supervised Learning (SSL)



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

$$E \subseteq 2^V$$

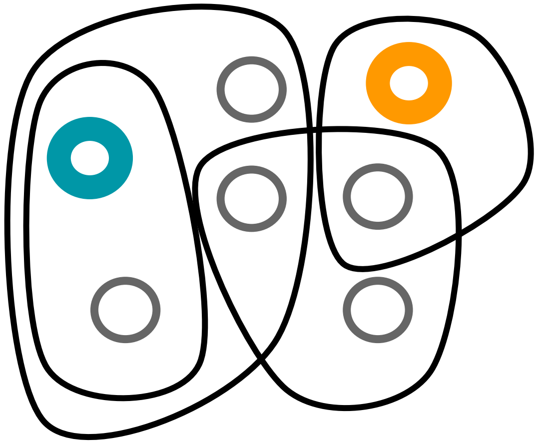
use labelled and unlabelled data for training

expensive cheap

e.g. document classification in co-authorship

Learn $f : \left\{x_1, \cdots, x_n\right\} \rightarrow \left\{y_1, \cdots, y_c\right\}$

Related Work



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

$$E \subseteq 2^V$$

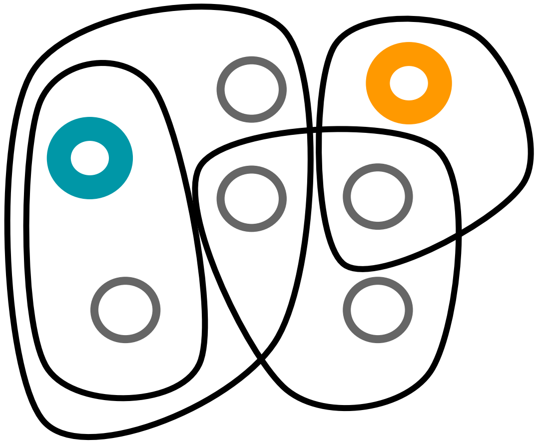
explicit regularisation

$$\underbrace{\mathcal{L}}\ = \underbrace{\mathcal{L}_S}_{\text{supervised}} + \underbrace{\lambda \cdot Q(\mathcal{H}, f)}_{\text{unsupervised}}$$

- Zhou et al. NIPS'06
- Hein et al. NIPS'13
- Anand Louis. STOC'15
- Chan and Liang. COCOON'18

Learn $f : \{x_1, \cdots, x_n\} \rightarrow \{y_1, \cdots, y_c\}$

Related Work



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

$$E \subseteq 2^V$$

explicit regularisation

$$\mathcal{L} = \mathcal{L}_S + \lambda \cdot Q(\mathcal{H}, f)$$

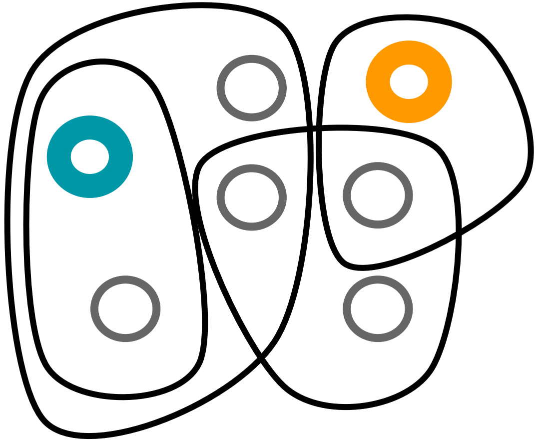
supervised — unsupervised

- Zhou et al. NIPS'06
- Hein et al. NIPS'13
- Anand Louis. STOC'15
- Chan and Liang. COCOON'18

❌ **hyperedges encode similarity**

Learn $f : \{x_1, \cdots, x_n\} \to \{y_1, \cdots, y_c\}$

Related Work



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

$$E \subseteq 2^V$$

explicit regularisation

$$\mathcal{L} = \underbrace{\mathcal{L}_S}_{\text{supervised}} + \underbrace{\lambda \cdot Q(\mathcal{H}, f)}_{\text{unsupervised}}$$

- Zhou et al. NIPS'06
- Hein et al. NIPS'13
- Anand Louis. STOC'15
- Chan and Liang. COCOON'18

❌ hyperedges encode similarity

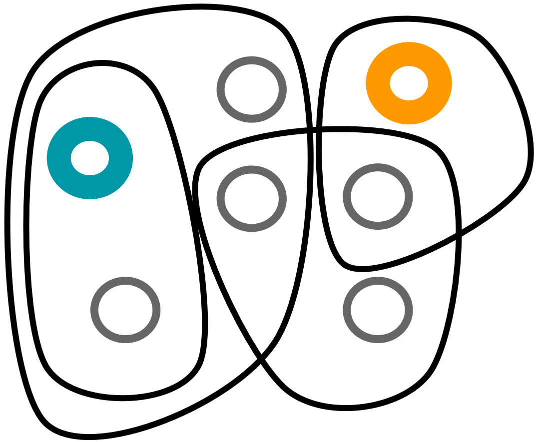
Learn $f : \{x_1, \cdots, x_n\} \rightarrow \{y_1, \cdots, y_c\}$

Our focus: Implicit regularisation

$$f_{Neural}\Big(\mathcal{H}, X\Big) = ?$$

$$\mathcal{L} = \mathcal{L}_S$$

Related Work



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

$$E \subseteq 2^V$$

explicit regularisation

$$\mathcal{L} = \mathcal{L}_S + \lambda \cdot Q(\mathcal{H}, f)$$

supervised   unsupervised

- Zhou et al. NIPS'06
- Hein et al. NIPS'13
- Anand Louis. STOC'15
- Chan and Liang. COCOON'18

✗ hyperedges encode similarity

Learn $f : \{x_1, \cdots, x_n\} \to \{y_1, \cdots, y_c\}$

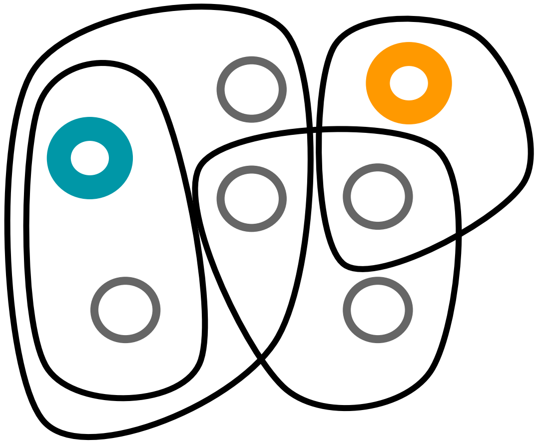
Our focus: Implicit regularisation

$$f_{Neural}\Big(\mathcal{H}, X\Big) = ?$$

$$\mathcal{L} = \mathcal{L}_S$$

✓ hyperedges need not encode similarity

Related Work



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

$$E \subseteq 2^V$$

explicit regularisation

$$\mathcal{L} = \mathcal{L}_S + \lambda \cdot Q(\mathcal{H}, f)$$

supervised unsupervised

- Zhou et al. NIPS’06
- Hein et al. NIPS’13
- Anand Louis. STOC’15
- Chan and Liang. COCOON’18

✗ **hyperedges encode similarity**

Learn $f : \{x_1, \cdots, x_n\} \to \{y_1, \cdots, y_c\}$

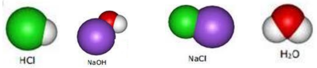
Our focus: Implicit regularisation

$$f_{Neural}\Big(\mathcal{H}, X\Big) = \,?$$

$$\mathcal{L} = \mathcal{L}_S$$

✓ **hyperedges need not encode similarity**

e.g.



Hypergraph total variation [Hein et al. NeurIPS'13]

$$\mathcal{L} = \mathcal{L}_S + \lambda \cdot Q(\mathcal{H}, f)$$

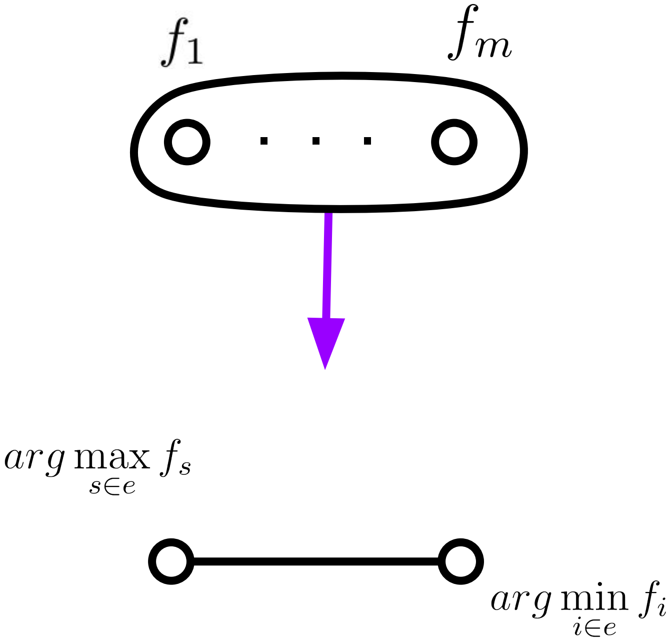
graphs: $Q\Big(\mathcal{G}, f \Big) = \sum_{\{u,v\} \in E} \Big(f_u - f_v \Big)^2$

Hypergraph total variation [Hein et al. NeurIPS'13]

$$\mathcal{L} \quad = \quad \mathcal{L}_S \quad + \quad \lambda \cdot Q(\mathcal{H}, f)$$

hypergraphs: $Q\big(\mathcal{H}, f\ \big) = \sum_{e \in E} \Big(\max_{s \in e} f_s - \min_{i \in e} f_i \Big)^2$

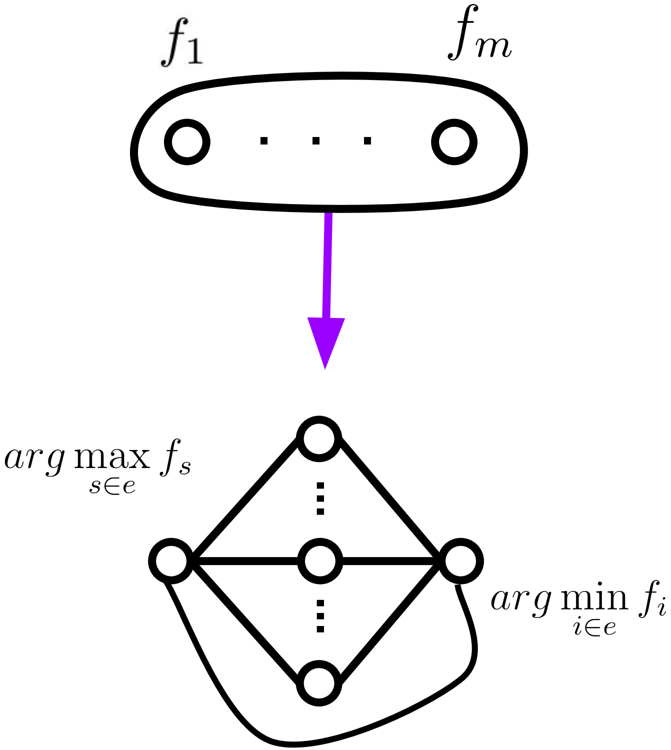
Hypergraph total variation [Hein et al. NeurIPS'13]



$$\mathcal{L} = \mathcal{L}_S + \lambda \cdot Q(\mathcal{H}, f)$$

hypergraphs: $Q\left(\mathcal{H}, f\right) = \sum_{e \in E} \left(\max_{s \in e} f_s - \min_{i \in e} f_i \right)^2$

Hypergraph total variation [Hein et al. NIPS 13]



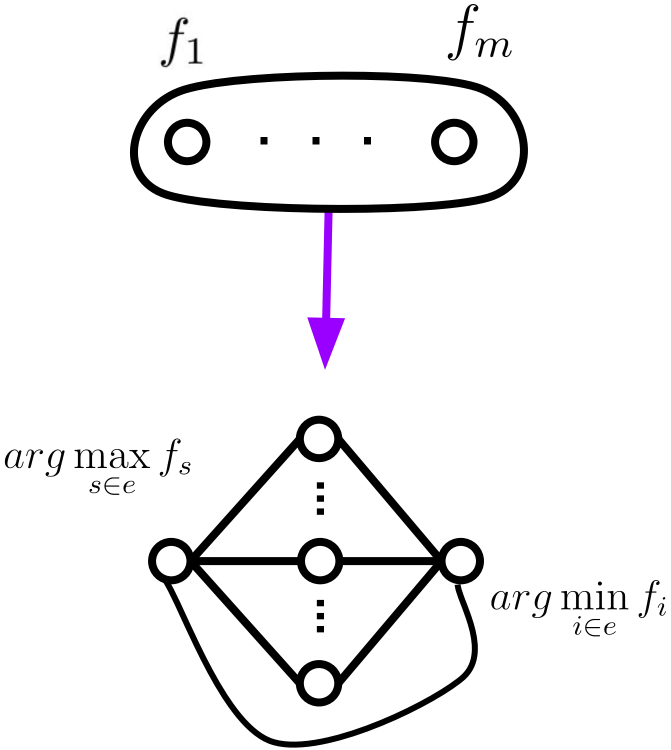
$$\mathcal{L} = \mathcal{L}_S + \lambda \cdot Q(\mathcal{H}, f)$$

hypergraphs: $Q(\mathcal{H}, f) = \sum_{e \in E} \left(\max_{s \in e} f_s - \min_{i \in e} f_i \right)^2$

$$+ \sum_{e \in E} \sum_{m \in e} \left[\left(\max_{s \in e} f_s - f_m \right)^2 + \left(f_m - \min_{i \in e} f_i \right)^2 \right]$$

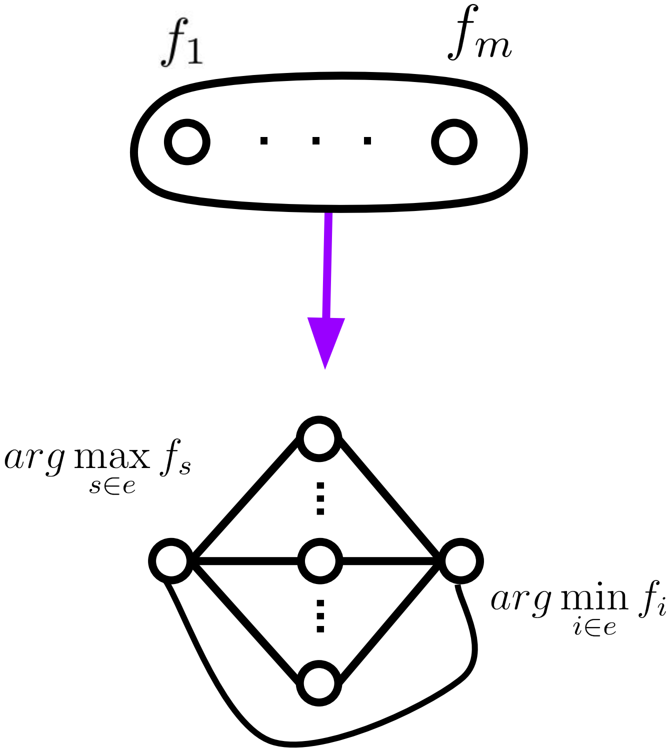
[Chan and Liang, COCOON 18]

Graph-based Hypergraph Modelling

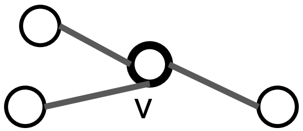


Graph neural network [Kipf and Welling, ICLR'16]

Graph-based Hypergraph Modelling

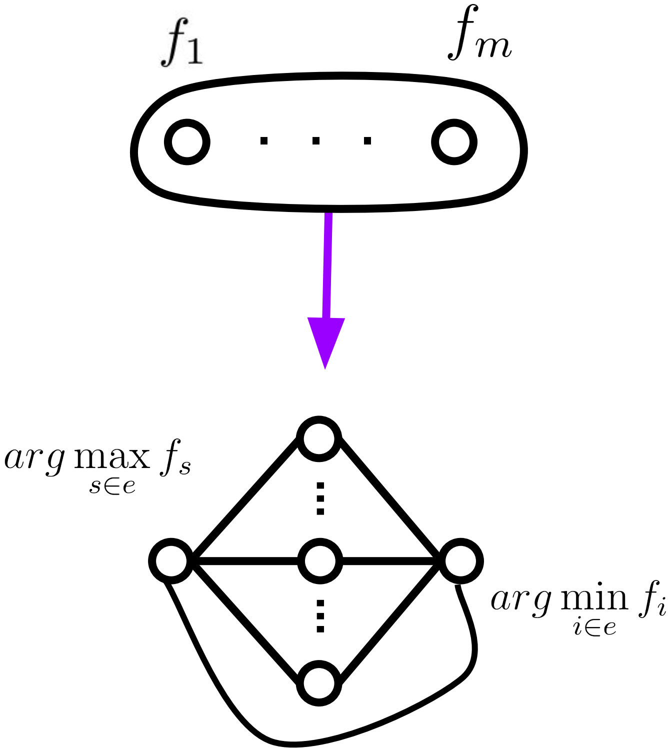


Graph neural network [Kipf and Welling, ICLR'16]

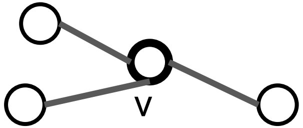


$$h_v^{\{l\}} = \sigma\left(\frac{W^{\{l\}}}{|\mathcal{N}_v|} \sum_{\{u,v\} \in \mathcal{N}_v} h_u^{\{l-1\}} + b \right)$$

Graph-based Hypergraph Modelling



Graph neural network [Kipf and Welling, ICLR'16]

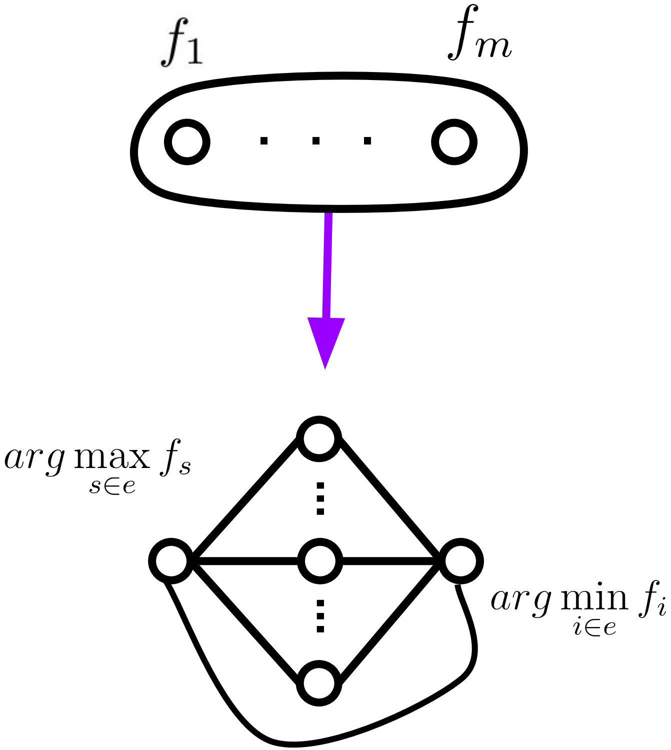


$$h_v^{\{l\}} = \sigma\left(\frac{W^{\{l\}}}{|\mathcal{N}_v|} \sum_{\{u,v\} \in \mathcal{N}_v} h_u^{\{l-1\}} + b \right)$$

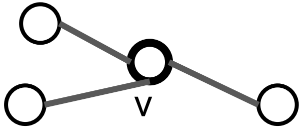
representation in current layer

representation in previous layer

Graph-based Hypergraph Modelling



Graph neural network [Kipf and Welling, ICLR'16]



$$h_v^{\{l\}} = \sigma\left(\frac{W^{\{l\}}}{|\mathcal{N}_v|} \sum_{\{u,v\} \in \mathcal{N}_v} h_u^{\{l-1\}} + b\right)$$

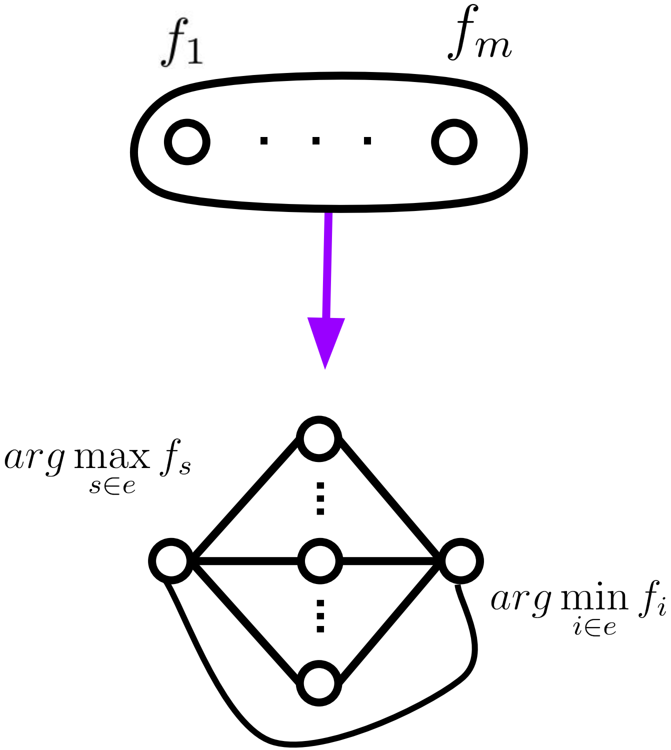
representation in current layer

non-linear activation

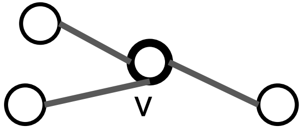
set of all neighbours of v

representation in previous layer

Graph-based Hypergraph Modelling



Graph neural network [Kipf and Welling, ICLR'16]



$$h_v^{\{l\}} = \sigma\left(\frac{W^{\{l\}}}{|\mathcal{N}_v|} \sum_{\{u,v\} \in \mathcal{N}_v} h_u^{\{l-1\}} + b\right)$$

weight in current layer

bias

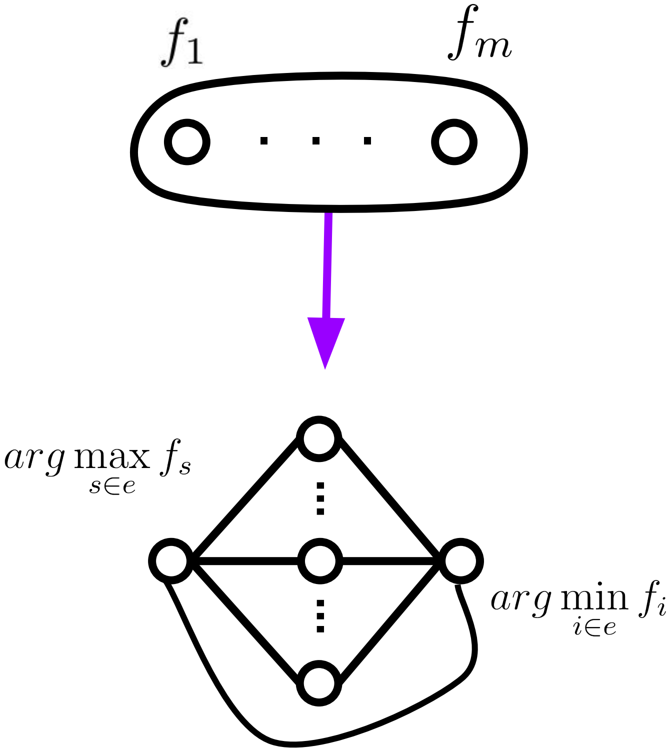
representation in current layer

non-linear activation

set of all neighbours of v

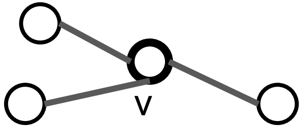
representation in previous layer

Graph-based Hypergraph Modelling



Graph neural network [Kipf and Welling, ICLR'16]

$$H^{\{l\}} = \sigma\left(A \cdot H^{\{l-1\}} \cdot W^{\{l\}} \right)$$



$$h_v^{\{l\}} = \sigma\left(\frac{W^{\{l\}}}{|\mathcal{N}_v|} \sum_{\{u,v\} \in \mathcal{N}_v} h_u^{\{l-1\}} + b \right)$$

weight in current layer

bias

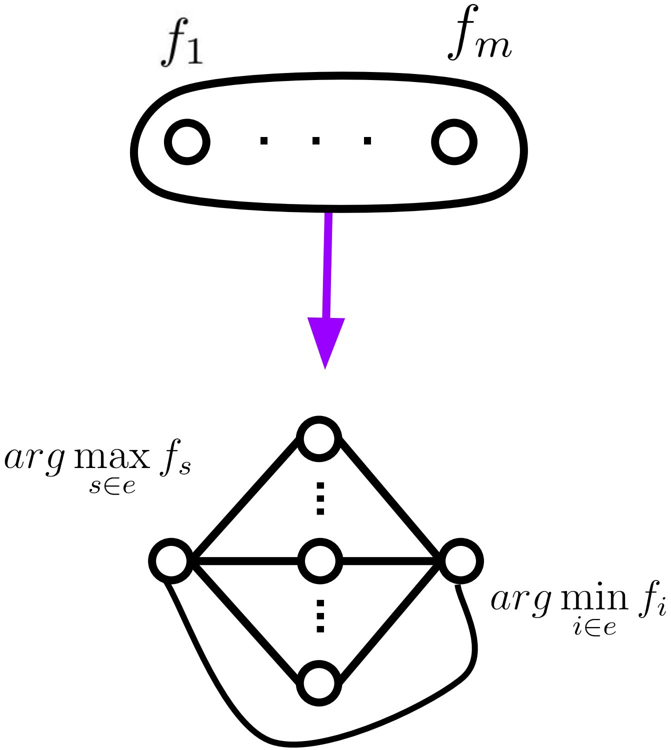
representation in current layer

non-linear activation

set of all neighbours of v

representation in previous layer

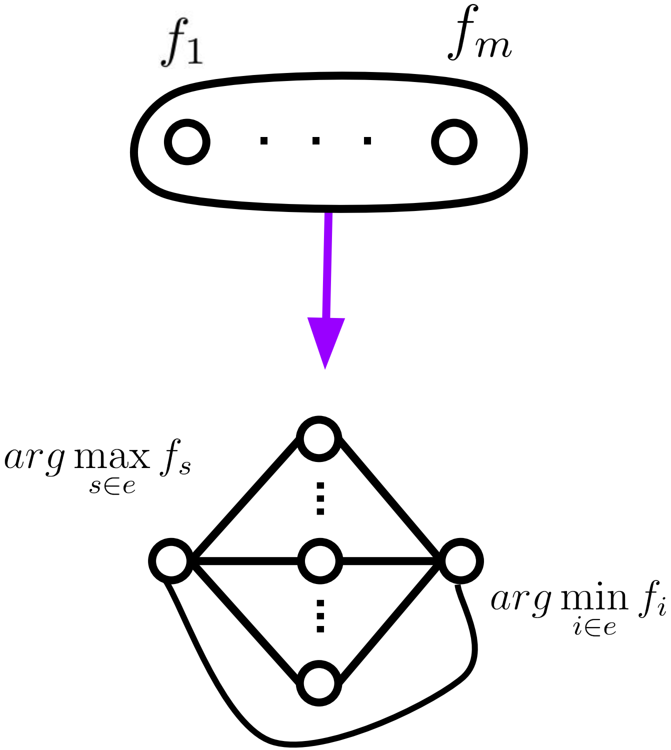
Hypergraph Convolutional Network



$$H^{\{l\}} = \sigma\left(A \cdot H^{\{l-1\}} \cdot W^{\{l\}}\right)$$

Set $f = H^{\{l-1\}} \cdot W^{\{l\}}$

Hypergraph Convolutional Network

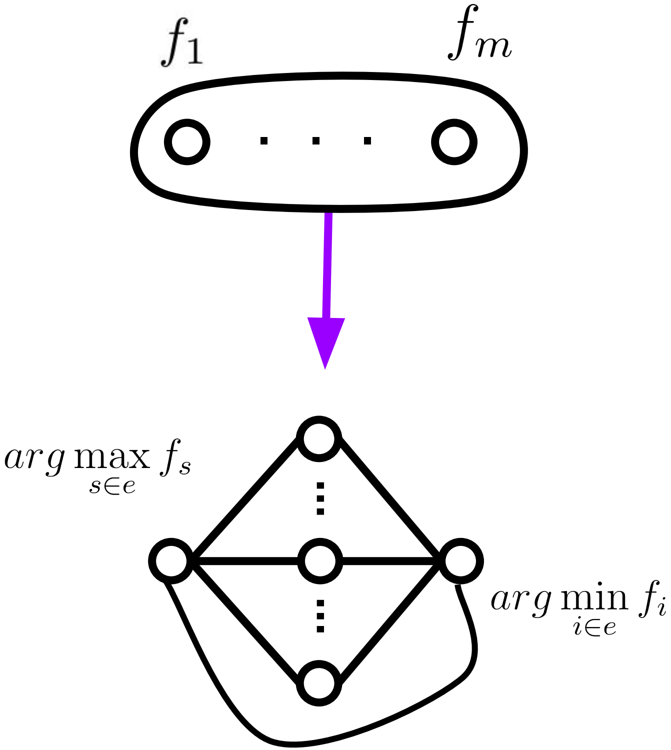


$$H^{\{l\}} = \sigma\left(\underset{n \times n}{A} \cdot \underset{n \times d_{l-1}}{H^{\{l-1\}}} \cdot \underset{d_{l-1} \times d_l}{W^{\{l\}}}\right)$$

Set $f = H^{\{l-1\}} \cdot W^{\{l\}}$

$\boxed{\textbf{parameters shared across input}}$

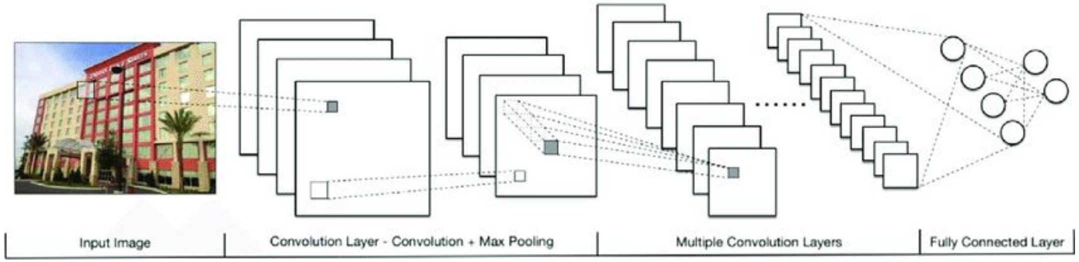
Hypergraph Convolutional Network



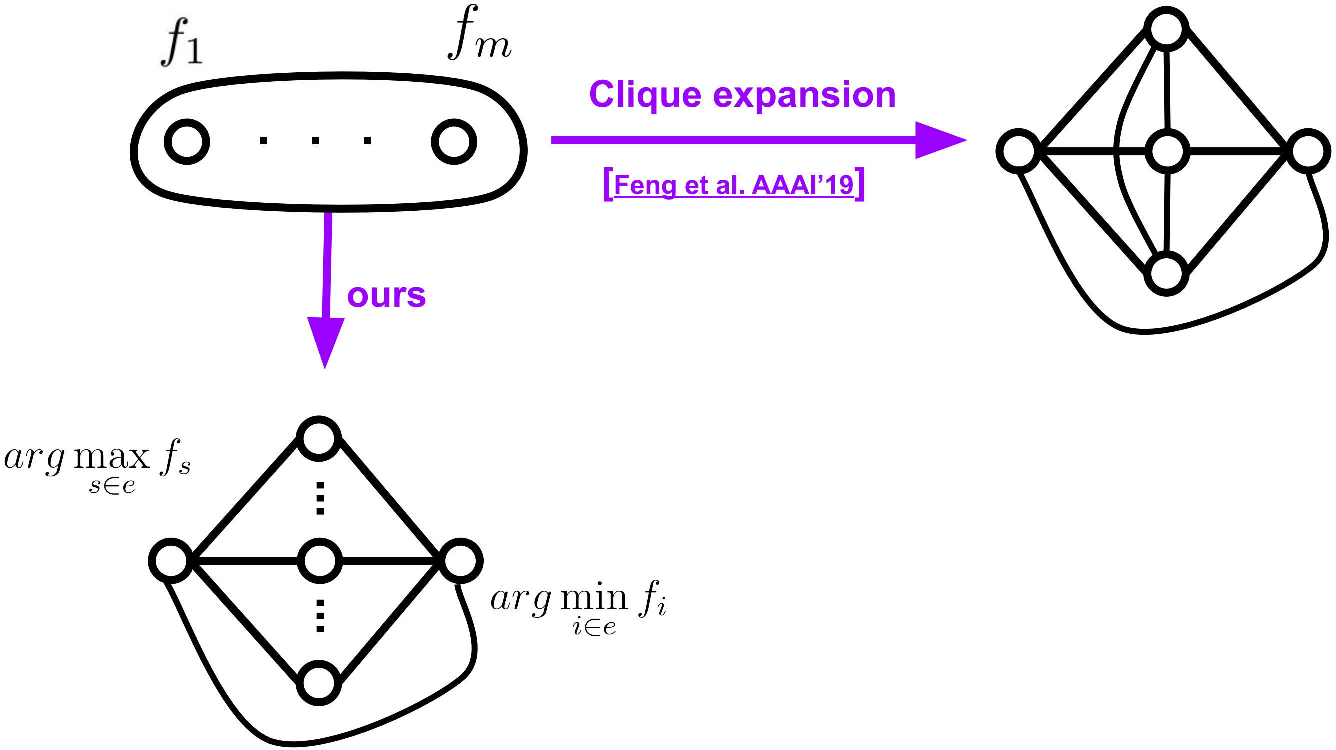
$$H^{\{l\}} = \sigma\Big(\underset{n \times n}{A} \cdot \underset{n \times d_{l-1}}{H^{\{l-1\}}} \cdot \underset{d_{l-1} \times d_l}{W^{\{l\}}} \Big)$$

Set $f = H^{\{l-1\}} \cdot W^{\{l\}}$

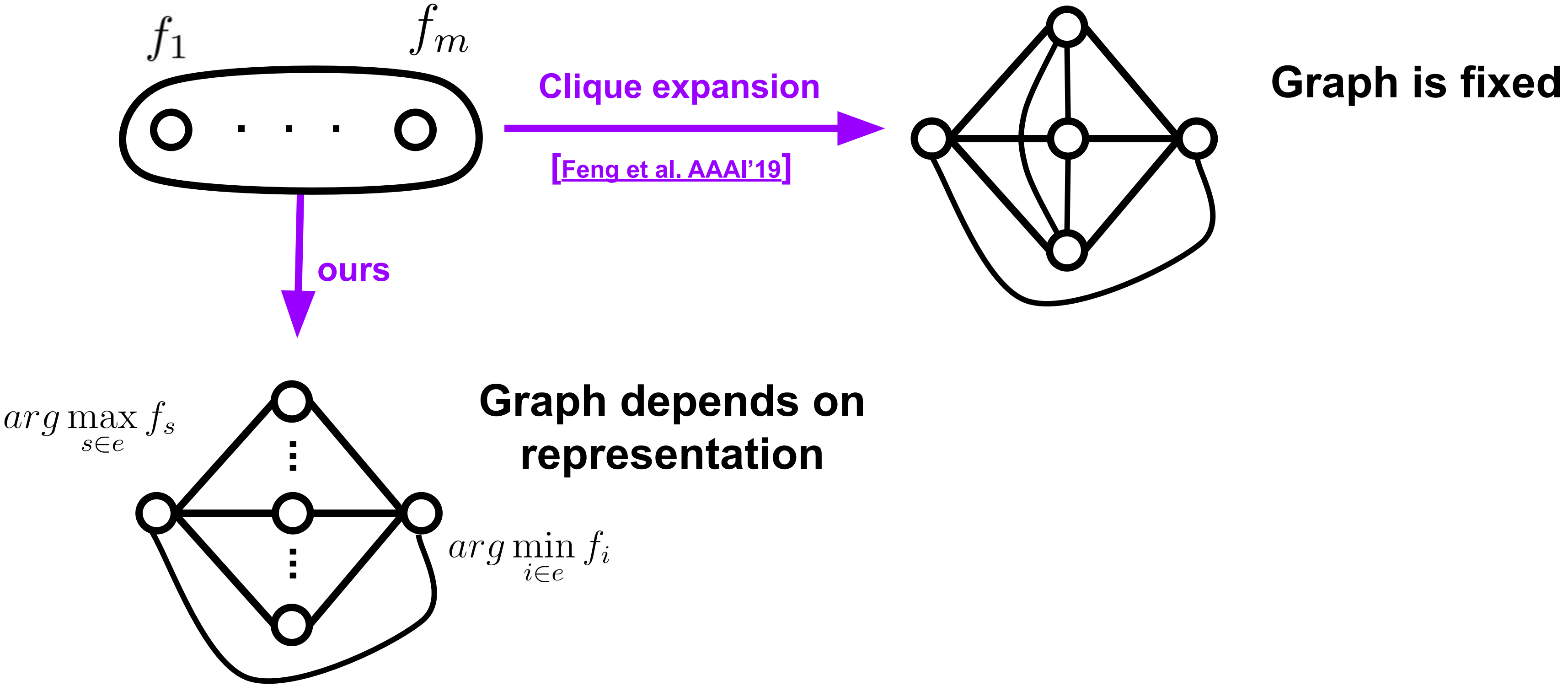
parameters shared across input



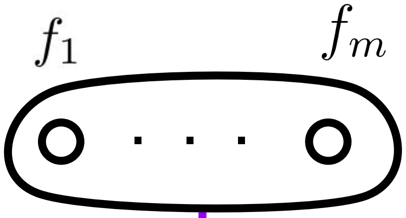
A simple, strong baseline



A simple, strong baseline

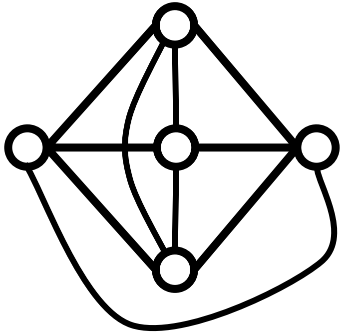


A simple, strong baseline



Clique expansion

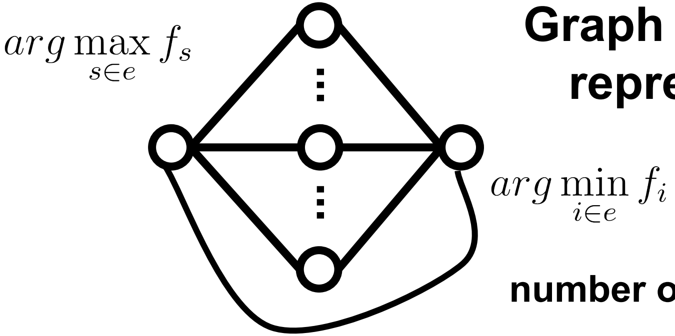
[Feng et al. AAAI'19]



Graph is fixed

number of edges is $^{m}C_{2}$

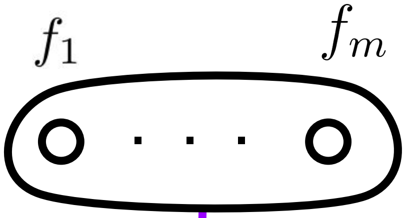
ours



Graph depends on representation

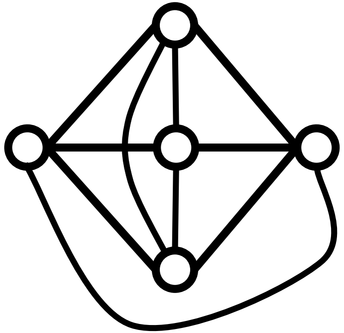
number of edges is 2m-3

A simple, strong baseline



Clique expansion

[Feng et al. AAAI'19]

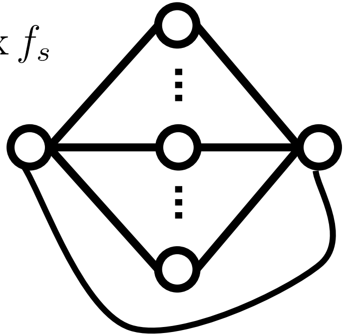


Graph is fixed

number of edges is $^mC_2$

ours

$\arg\max_{s \in e} f_s$



$\arg\min_{i \in e} f_i$

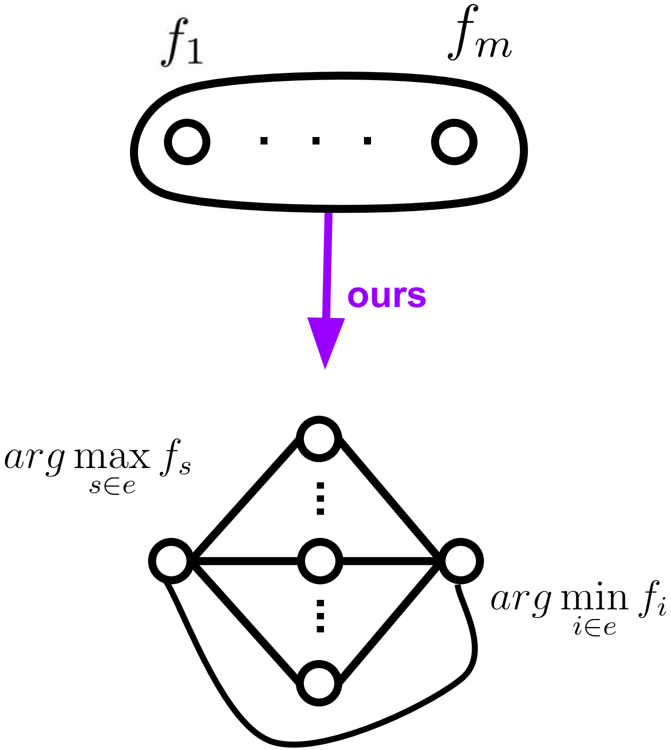
Graph depends on representation

number of edges is 2m-3

document classification on co-citation networks

	Cora	Citeseer
Avg. Hyperedge size	3.0 ± 1.1	3.2 ± 2.0
GCN on Clique Expansion	32.41 ± 1.8	37.40 ± 1.6
HyperGCN	32.37 ± 1.7	37.35 ± 1.6

FastHyperGCN



$$H^{\{l\}} = \sigma\Big(\underset{n \times n}{A} \cdot \underset{n \times d_{l-1}}{H^{\{l-1\}}} \cdot \underset{d_{l-1} \times d_l}{W^{\{l\}}} \Big)$$

HyperGCN

Set $f = H^{\{l-1\}} \cdot W^{\{l\}}$

FastHyperGCN

Set $f = H^{\{0\}} = X$

Experiments on large noisy hypergraphs

Test accuracy (lower is better) on co-authorship and co-citation datasets

	DBLP
Avg. Hyperedge size	8.5 ± 8.8
GCN on Clique Expansion	45.27 ± 2.4
HyperGCN	$\mathbf{41.64 \pm 2.6}$
FastHyperGCN	41.78 ± 2.8

Experiments on large noisy hypergraphs

Test accuracy (lower is better) on co-authorship and co-citation datasets

	DBLP
Avg. Hyperedge size	8.5 ± 8.8
GCN on Clique Expansion	45.27 ± 2.4
HyperGCN	$\mathbf{41.64 \pm 2.6}$
FastHyperGCN	41.78 ± 2.8

- Authors can co author documents from different topics

Experiments on large noisy hypergraphs

Test accuracy (lower is better) on co-authorship and co-citation datasets

	DBLP
Avg. Hyperedge size	8.5 ± 8.8
GCN on Clique Expansion	45.27 ± 2.4
HyperGCN	$\mathbf{41.64 \pm 2.6}$
FastHyperGCN	41.78 ± 2.8

- Authors can co author documents from different topics
- HyperGCN accumulates less noise than clique expansion

Experiments on large noisy hypergraphs

Test accuracy (lower is better) on co-authorship and co-citation datasets

	DBLP	Pubmed	Cora
Avg. Hyperedge size	8.5 ± 8.8	4.3 ± 5.7	4.2 ± 4.1
GCN on Clique Expansion	45.27 ± 2.4	29.41 ± 1.5	31.90 ± 1.9
HyperGCN	$\mathbf{41.64 \pm 2.6}$	$\mathbf{25.56 \pm 1.6}$	$\mathbf{30.08 \pm 1.8}$
FastHyperGCN	41.78 ± 2.8	29.48 ± 1.6	32.54 ± 1.8

- Authors can co author documents from different topics
- HyperGCN accumulates less noise than clique expansion

Experiments on large noisy hypergraphs

Test accuracy (lower is better) on co-authorship and co-citation datasets

	DBLP	Pubmed	Cora
Avg. Hyperedge size	8.5 ± 8.8	4.3 ± 5.7	4.2 ± 4.1
GCN on Clique Expansion	45.27 ± 2.4	29.41 ± 1.5	31.90 ± 1.9
HyperGCN	$\mathbf{41.64 \pm 2.6}$	$\mathbf{25.56 \pm 1.6}$	$\mathbf{30.08 \pm 1.8}$
FastHyperGCN	41.78 ± 2.8	29.48 ± 1.6	32.54 ± 1.8

- Authors can co author documents from different topics
- HyperGCN accumulates less noise than clique expansion

Average training time (lower is better) of an epoch

	DBLP	Pubmed
GCN on Clique Expansion	0.115s	0.019s
FastHyperGCN	**0.035s**	**0.016s**

What NeurIPS reviewers liked in the paper

- Bridges different fields
  Spectral hypergraph theory + graph neural networks

What NeurIPS reviewers liked in the paper

- Bridges different fields

  Spectral hypergraph theory + graph neural networks

- Reduces complexity from quadratic to linear

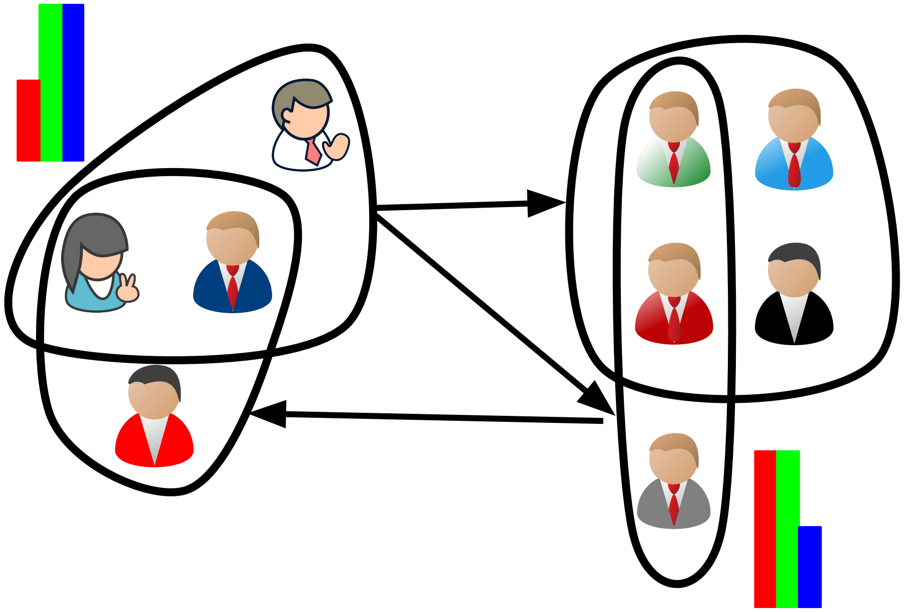
  ${}^{m}C_2$ to 2m-3

What NeurIPS reviewers liked in the paper

- Bridges different fields
  Spectral hypergraph theory + graph neural networks
- Reduces complexity from quadratic to linear
  ${}^{m}C_{2}$ to 2m-3
- Improves performance on large noisy hypergraphs
  lower error and training time

Limitations and Future Work

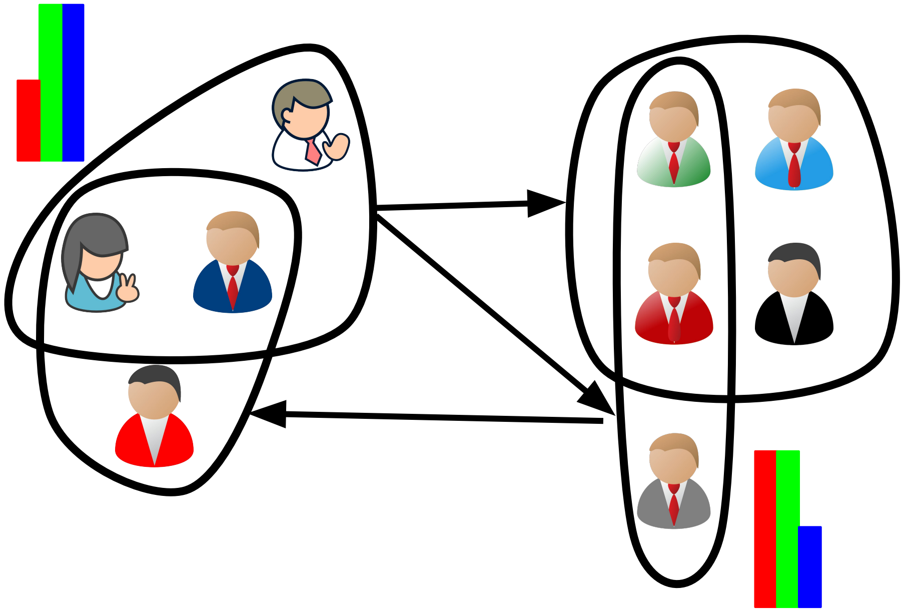
- **Soft Semi-supervised learning** (submitted to ICLR 2020)



Limitations and Future Work

- **Soft Semi-supervised learning** **(submitted to ICLR 2020)**

- **Unsupervised learning**

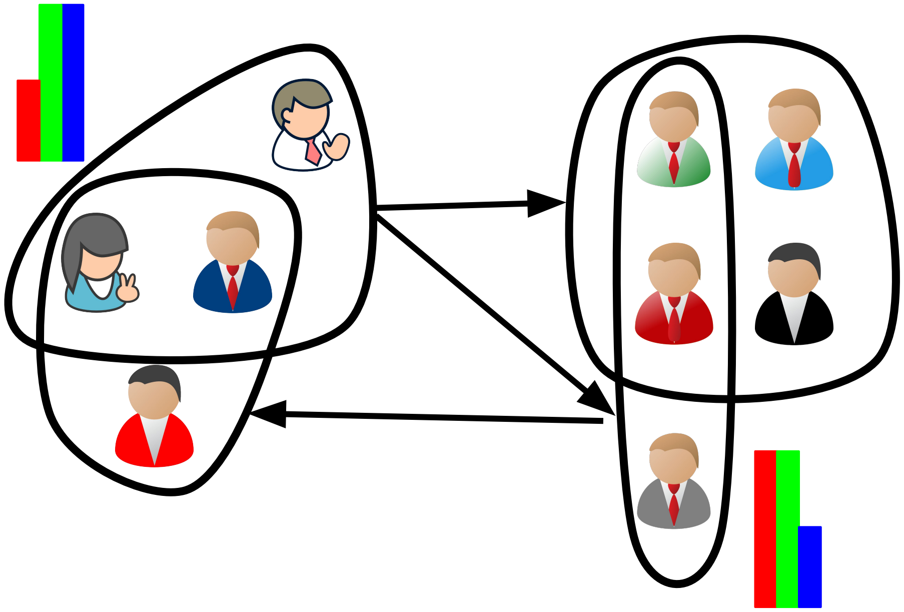


Limitations and Future Work

- **Soft Semi-supervised learning** (submitted to ICLR 2020)

- **Unsupervised learning**

❌ **Inherently transductive**
cannot handle unseen vertices at test time



Q & A