NetVec: A Scalable Hypergraph Embedding System

Sepideh Maleki[1], Donya Saless[2], Dennis P Wall[3], and Keshav Pingali[1]

[1]The University of Texas at Austin, [2]University of Tehran, [3]Stanford University

Motivation: Prediction tasks on hypergraphs.

- Hypergraphs arise in many application domains.
- Many important problems can be formulated as classification and prediction problems for hyperedges.

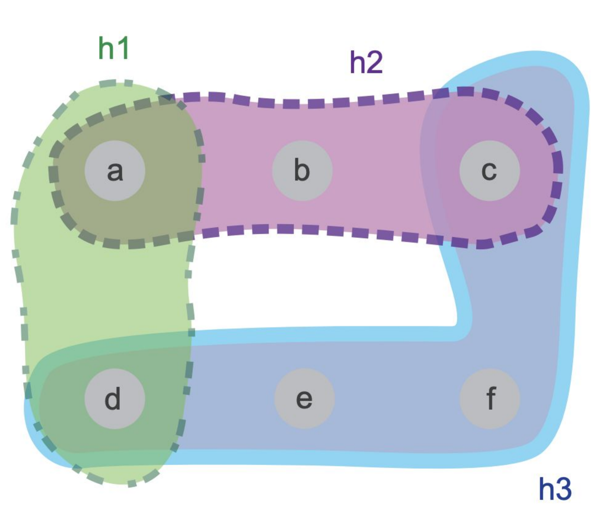
Our work:

- NetVec, a novel hierarchical framework for scalable, unsupervised embeddings of hyperedges and nodes.
- NetVec can be coupled with any existing network embedding algorithm to reduce the amount of compute time.

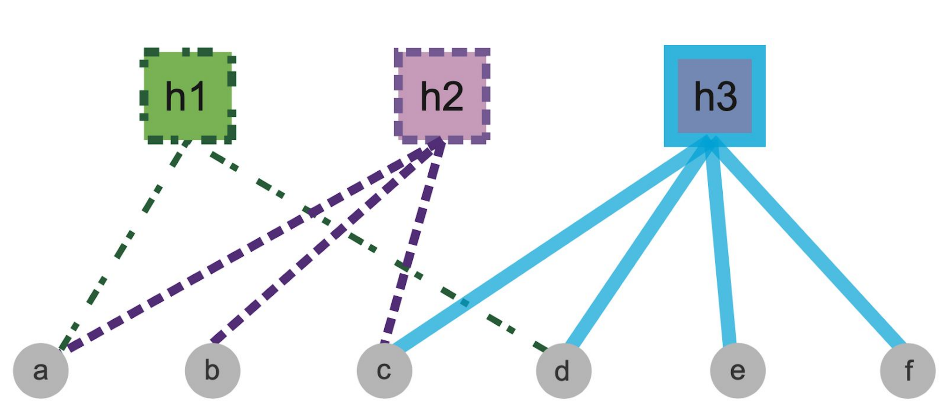
Background

Previous Approaches	vs. NetVec
• Hyperedges are not represented explicitly • Takes days to find embedding of hypergraphs with millions of nodes and hyperedges	• Hyperedges are represented explicitly • Finds embedding of hypergraphs with millions of nodes and hyperedges in just a few minutes

A hypergraph with 3 hyperedges and 6 nodes.

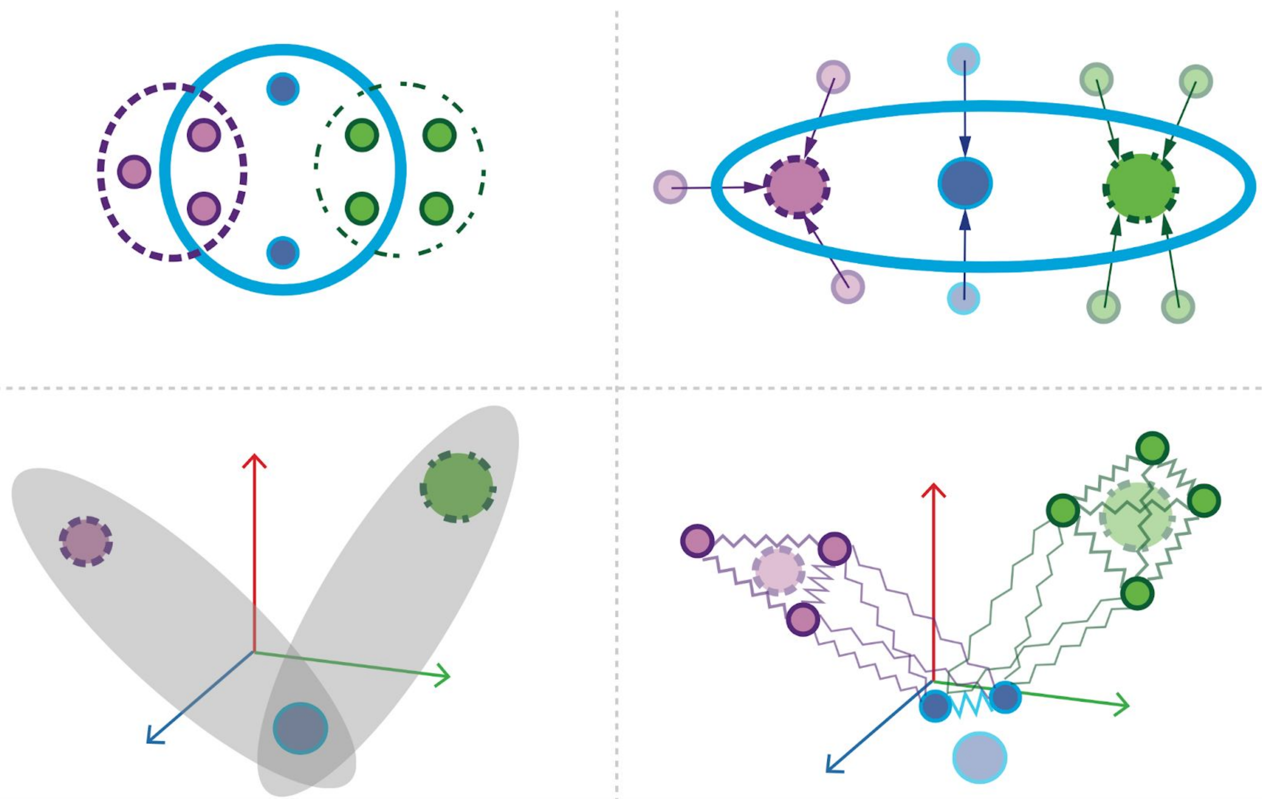


Star expansion of the hypergraph above,



NetVec: a scalable multi-level embedding framework

We developed **NetVec**, a parallel multi-level framework for constructing hypergraph embeddings. This framework consists of three phases: **(i)** Coarsening, **(ii)** Initial embedding, and **(iii)** Refinement.



Algorithm 1: Coarsening

Input: fineGraph $G = (V, E, \omega, W)$, neighborhood function $\mathcal{N}(u)$, depth K
Output: coarseGraph $G' = (V', E', \omega, W')$
for $i = 1$ to K **do**
  for $v \in V$ **do**
    AssignHyperedge($\mathcal{N}(v)$)
  end for
  for $e \in E$ **do**
    $M \leftarrow$ FindAssignedNodes(M)
    $N \leftarrow$ Merge(M)
    coarseGraph.addNode(N)
  end for
end for

(i) At each level of **coarsening**, *AssignHyperedge* assigns each node *v* in the current hypergraph to a hyperedge c(v) as defined below where ω(e) is the weight of hyperedge *e* and $\delta_e(v)$ is the weight of node *v* in hyperedge e.

$$c(v) = \underset{e \in E}{\operatorname{argmax}}\ \omega(e) \cdot \delta_e(v)$$

(ii) NetVec coarsens the hypergraph until it is small enough that *any* unsupervised embedding method can generate the embedding of the coarsest graph in just a few seconds.

(iii) The goal of ***refinement*** is to improve embedding obtained from the initial embedding algorithm. If a set of node *S* in a hypergraph H_{i-1} were merged to form a node n in the coarser hypergraph H_i, the embedding of n in H_i is assigned to all the nodes of *S* in H_{i-1} at the beginning of refinement.

Algorithm 2: Refinement

Input: Bipartite graph representation $H^* = (V^*, E^*, \omega, W)$ of hypergraph $H = (V, E, \omega, \delta)$, vector representation z_u for all $u \in (V^*)$, neighborhood function $\mathcal{N}(u)$, depth K, parameter ω
Output: Refined vector representation $h_u, \forall u \in (V^*)$
$z_u^0 \leftarrow z_u, \forall u \in (V^*)$
for $i = 1$ **to** K **do**
  for $u \in E$ **do**
    $\tilde{z}_u^i \leftarrow \sum_{v \in \mathcal{N}(u)} w_{uv} z_v^{i-1} / \sum_{v \in \mathcal{N}(u)} w_{uv}$
    $z_u^i \leftarrow (1-\omega) z_u^{i-1} + \omega \tilde{z}_u^i$
  end for
  for $u \in V$ **do**
    $\tilde{z}_u^i \leftarrow \sum_{v \in \mathcal{N}(u)} w_{vu} z_v^{i-1} / \sum_{v \in \mathcal{N}(u)} w_{vu}$
    $z_u^i \leftarrow (1-\omega) z_u^{i-1} + \omega \tilde{z}_u^i$
  end for
end for
$h_u \leftarrow z_u^k, \forall u \in (V^*)$

Abstractly, the refinement algorithm uses Jacobi over-relaxation to solve the linear system, Lz = 0 using the relaxation parameter ω, where L is the Laplacian matrix of H^*. It is defined as D - W where D is the diagonal matrix with diagonal elements $d_{ii} = \sum_j w_{ij}$, W is the weighted adjacency matrix of H^*.

Results (hyperedge prediction)

We compare NetVec with Hyper-SAGNN[1], the state of the art supervised hyperedge prediction framework and Node2Vec on 4 standard datasets listed below.

	Node Type			#V			#E
GPS	USER	LOCATION	ACTIVITY	146	70	5	1,436
MovieLens	USER	MOVIE	TAG	2,113	5,908	9,079	47,957
DRUG	USER	DRUG	REACTION	12	1,076	6,398	171,756
WORDNET	HEAD	RELATION	TAIL	40,504	18	40,551	145,966

Area Under Curve (AUC) scores for hyperedge prediction. Time is in seconds.

	GPS		MovieLens		Drug		WordNet	
	AUC	Time	AUC	Time	AUC	Time	AUC	Time
NetVec	**94.4**	10	**96.9**	20	**98.0**	900	**92.8**	950
Hyper-SAGNN	90.6	1800	90.8	11,160	95.9	39,540	87.7	82,800
Node2Vec	94.0	10	79.8	19	97.4	895	89.0	940

For hyperedge classification task, we compare NetVec with two other multi-level frameworks MILE[2], and GraphZoom[3] on large and smaller hypergraph datasets.

Results (hyperedge classification)

Data set	Nodes	Hyperedges	Edges	Classes
Cora	2,709	1,963	10,556	7
Citeseer	3,328	2,182	9,352	6
Corum	6,132	8,274	611,684	2
Chem2Bio	295,911	727,997	2,911,988	12
Amazon	358,017	135,268	10,155,376	2
Live Journal	1,423,948	1,195,945	35,684,736	2
Friendster	8,724,335	2,917,783	94,193,160	2

Hyperedge classification for small hypergraphs. Time in seconds. Node2Vec is the initial embedding algorithm.

	Cora		Citeseer		Corum		Chem2Bio	
	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
NetVec	76.5	**20**	**58.0**	**20**	93.4	**75**	**81.9**	**2,760**
MILE	**77.2**	**20**	57.0	25	91.7	144	-	-
GraphZoom	77.2	33	57.5	45	93.4	155	-	-
Node2Vec	76.0	35	57.1	45	**94.3**	165	71.8	9,025

Hyperedge classification for large hypergraphs. Time in seconds. Node2Vec is the initial embedding algorithm.

	Amazon		LiveJournal		Friendster	
	Accuracy	Time	Accuracy	Time	Accuracy	Time
NetVec	**81.4**	**120**	62.8	**480**	**55.7**	**430**
MILE	55.8	3,900	55.4	4,620	-	-
GraphZoom	63.1	79,140	**64.4**	262,860	-	-

References

[1] Ruochi Zhang, Yuesong Zou, and Jian Ma. 2020. Hyper-SAGNN: a self-attention based graph neural network for hypergraphs. In International Conference on Learning Representations (ICLR).

[2]Jiongqian Liang, Saket Gurukar, and Srinivasan Parthasarathy. 2020. MILE: A Multi-Level Framework for Scalable Graph Embedding

[3]Chenhui Deng, Zhiqiang Zhao, Yongyu Wang, Zhiru Zhang, and Zhuo Feng. 2020. GraphZoom: A multi-level spectral approach for accurate and scalable graph embedding.

Contact

Sepideh Maleki: Smaleki@cs.utexas.edu

TEXAS
The University of Texas at Austin