vGraph: A Generative Model for Joint Community Detection and Node Representation Learning

Fan-Yun Sun, Meng Qu, Jordan Hoffman, Chin-Wei Huang, Jian Tang

Mila, CIFAR

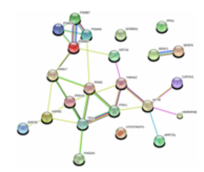
sunfanyun@gmail.com

Background

- **Graphs are ubiquitous with a variety of applications.**

Social Network

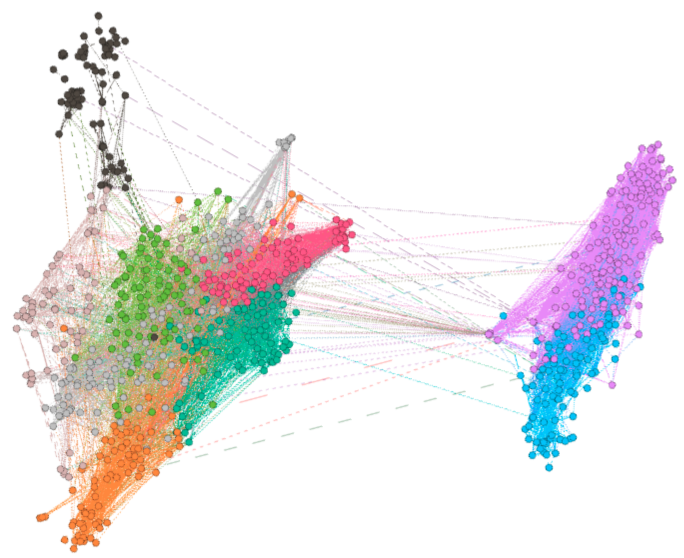
Airline Network



Protein Interaction Network

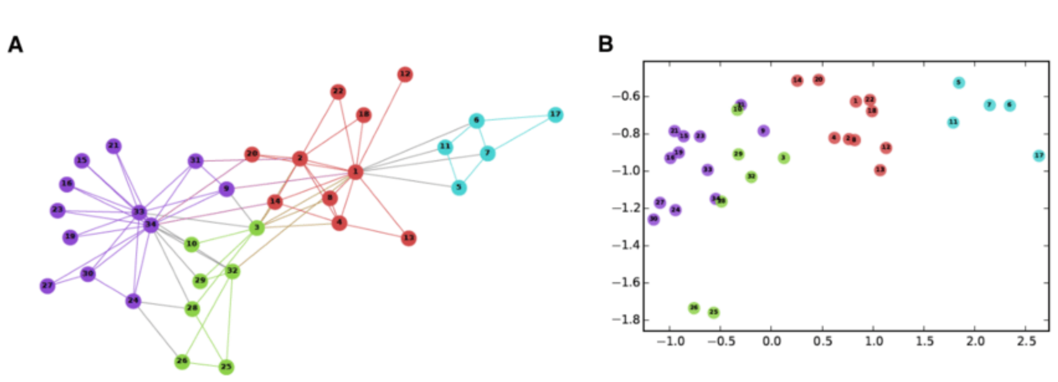
Community Detection

- **Consider a graph $G = (V, E)$:**
  - $V = \{v_1, \ldots, v_V\}$ is a set of vertices.
  - $E = \{e_{ij}\}$ is the set of edges.
- **Learn community assignment of all nodes. Community assignment of node v_i can be denoted as $F(v_i) \subseteq \{1, ..., K\}$.**



Node Representation Learning

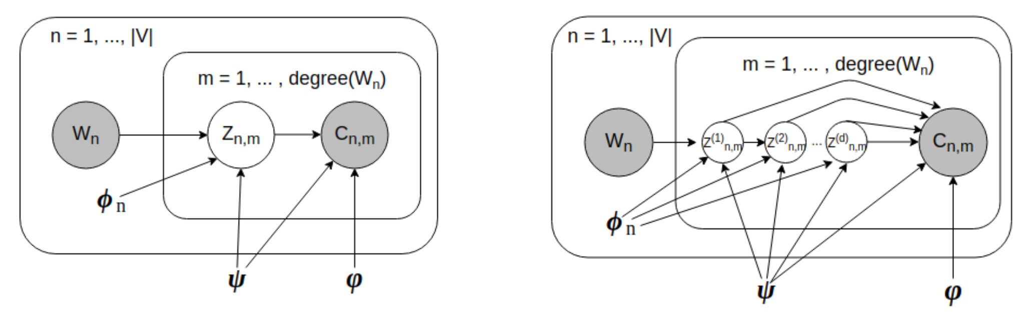
- **Consider a graph $G = (V, E)$:**
  - $V = \{v_1, \ldots, v_V\}$ is a set of vertices.
  - $E = \{e_{ij}\}$ is the set of edges.
- **Learn a node embedding $\phi_i \in \mathbb{R}^d$ for each $v_i \in V$ where d is predetermined dimension.**



https://ai.googleblog.com/2019/06/innovations-in-graph-representation.html

vGraph: a probabilistic generative model.

- **Towards Combining Community Detection and Node Representation Learning:**
  - Each node can be represented as a mixture of communities.
  - Each community is defined as a multinomial distribution over nodes.
- **Generative Process:**
  - For node w, we first draw a community assignment $z \sim p(z|w)$, representing the social context of w during the generation process.
  - Then, the linked neighbor c is generated based on the assignment z through $c \sim p(c|z)$.
  - This generation process can be formulated as $p(c|w) = \sum_z p(c|z)p(z|w)$. For hierarchical vGraph, $p(c|w) = \sum_{\vec{z}} p(c|\vec{z})p(\vec{z}|w)$.



- **Variational Inference:**
  - Objective: maximize the log-likelihood of observed edges.
  - Define approximate posterior: $q(z|c, w)$.
  - Optimize the evidence lower bound (ELBO):
    - ψ, φ: two sets of node embeddings.
    - ϕ: community embeddings.

$$\mathcal{L} = E_{z\sim q(z|c,w)}[\log \boxed{p_{\psi,\varphi}(c|z)}] - \mathrm{KL}(q(z|c,w)\|\boxed{p_{\phi,\psi}(z|w)})$$

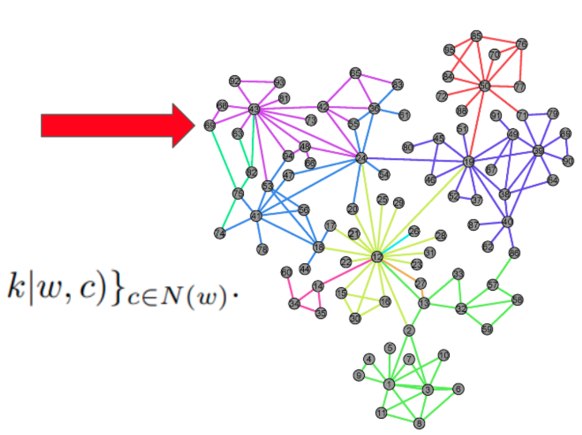
$\boxed{p_{\phi,\psi}(z = j|w)}$ **Softmax** of node embedding over community embeddings

$\boxed{p_{\psi,\varphi}(c|z = j)}$ **Softmax** of community embeddings over node embeddings

$q_{\phi,\psi}(z = j|w, c)$ **Softmax** of $\phi_w \odot \phi$ over community embeddings.

  - Infer overlapping communities:

$$\mathcal{F}(w) = \{\arg\max_k q(z = k|w, c)\}_{c \in N(w)}.$$



- **Community-smoothness Regularized Optimization:**
  - Two nodes are similar if they are connected and share similar neighbors.

$$\mathcal{L}_{reg} = \lambda \sum_{(w,c)\in\mathcal{E}} \alpha_{w,c} \cdot d(p(z|c), p(z|w)). \quad \alpha_{w,c} = \frac{|N(w) \cap N(c)|}{|N(w) \cup N(c)|}.$$

Experiment

- **Overlapping Community Detection:**
  - ⊡ Evaluation Metrics: F1-Score, Jaccard Similarity.

	F1-score						Jaccard					
Dataset	Bigclam	CESNA	Circles	SVI	vGraph	vGraph+	Bigclam	CESNA	Circles	SVI	vGraph	vGraph+
facebook0	**0.2948**	0.2806	0.2860	0.2810	0.2440	0.2606	**0.1846**	0.1725	0.1862	0.1760	0.1458	0.1594
facebook107	**0.3928**	0.3733	0.2467	0.2689	0.2817	0.3178	**0.2752**	0.2695	0.1547	0.1719	0.1827	0.2170
facebook1684	0.5041	**0.5121**	0.2894	0.3591	0.4232	0.4379	0.3801	**0.3871**	0.1871	0.2467	0.2917	0.3272
facebook1912	0.3493	0.3474	0.2617	0.2804	0.2579	**0.3750**	0.2412	0.2394	0.1672	0.2010	0.1855	**0.2796**
facebook3437	0.1986	0.2009	0.1009	0.1544	0.2087	**0.2267**	0.1148	0.1165	0.0545	0.0902	0.1201	**0.1328**
facebook348	0.4964	0.5375	0.5175	0.4607	**0.5539**	0.5314	0.3586	0.4001	0.3927	0.3360	**0.4099**	0.4050
facebook3980	0.3274	0.3574	0.3203	NA	**0.4450**	0.4150	0.2426	0.2645	0.2097	NA	**0.3376**	0.2933
facebook414	0.5886	0.6007	0.4843	0.3893	0.6471	**0.6693**	0.4713	0.4732	0.3418	0.2931	0.5184	**0.5587**
facebook686	0.3825	0.3900	0.5036	0.4639	0.4775	**0.5379**	0.2504	0.2534	0.3615	0.3394	0.3272	**0.3856**
facebook698	0.5423	0.5865	0.3515	0.4031	0.5396	**0.5950**	0.4192	0.4588	0.2255	0.3002	0.4356	**0.4771**
Youtube	0.4370	0.3840	0.3600	0.4140	0.5070	**0.5220**	0.2929	0.2416	0.2207	0.2867	0.3434	**0.3480**
Amazon	0.4640	0.4680	**0.5330**	0.4730	**0.5330**	0.5320	0.3505	0.3502	0.3671	0.3643	0.3689	**0.3693**
Dblp	0.2360	0.3590	NA	NA	0.3930	**0.3990**	0.1384	0.2226	NA	NA	0.2501	**0.2505**
Coauthor-CS	0.3830	0.4200	NA	0.4070	0.4980	**0.5020**	0.2409	0.2682	NA	0.2972	**0.3517**	0.3432

- **Non-overlapping Community Detection:**
  - ⊡ Evaluation Metrics: NMI, Modularity.

	NMI						Modularity					
Dataset	MF	deepwalk	LINE	node2vec	ComE	vGraph	MF	deepwalk	LINE	node2vec	ComE	vGraph
cornell	0.0632	0.0789	0.0697	0.0712	0.0732	**0.0803**	0.4220	0.4055	0.2372	0.4573	0.5748	**0.5792**
texas	0.0562	0.0684	**0.1289**	0.0655	0.0772	0.0809	0.2835	0.3443	0.1921	0.3926	**0.4856**	0.4636
washington	0.0599	0.0752	**0.0910**	0.0538	0.0504	0.0649	0.3679	0.1841	0.1655	0.4311	0.4862	**0.5169**
wisconsin	0.0530	0.0759	0.0680	0.0749	0.0689	**0.0852**	0.3892	0.3384	0.1651	0.5338	0.5500	**0.5706**
cora	0.2673	0.3387	0.2202	0.3157	**0.3660**	0.3445	0.6711	0.6398	0.4832	0.5392	0.7010	**0.7358**
citeseer	0.0552	0.1190	0.0340	0.1592	**0.2499**	0.1030	0.6963	0.6819	0.4014	0.4657	0.7324	**0.7711**

- **Node Classification:**
  - ⊡ Evaluation Metrics: Micro-F1, Macro-F1.

	Macro-F1						Micro-F1					
Datasets	MF	DeepWalk	LINE	Node2Vec	ComE	vGraph	MF	DeepWalk	LINE	Node2Vec	ComE	vGraph
Cornell	13.05	22.69	21.78	20.70	19.86	**29.76**	15.25	33.05	23.73	24.58	25.42	**37.29**
Texas	8.74	21.32	16.33	14.95	15.46	**26.00**	14.03	40.35	27.19	25.44	33.33	**47.37**
Washington	15.88	18.45	13.99	21.23	15.80	**30.36**	15.94	34.06	25.36	28.99	33.33	**34.78**
Wisconsin	14.77	23.44	19.06	18.47	14.63	**29.91**	18.75	**38.75**	28.12	25.00	32.50	35.00
Cora	11.29	13.21	11.86	10.52	12.88	**16.23**	12.79	22.32	14.59	27.74	**28.04**	24.35
Citeseer	14.59	16.17	15.99	16.68	12.88	**17.88**	15.79	19.01	16.80	**20.82**	19.42	20.42

- **Visualization:**

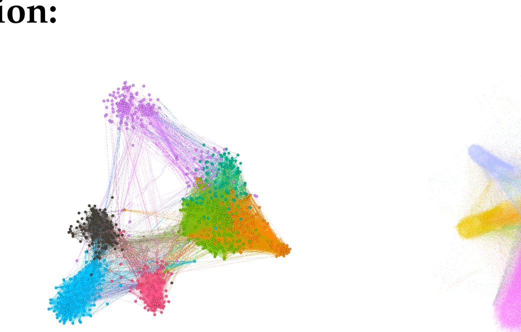


Figure 1: Result on the facebook107 dataset and Dblp-full dataset using vGraph. The coordinates of the nodes are determined by t-SNE of the node embeddings.

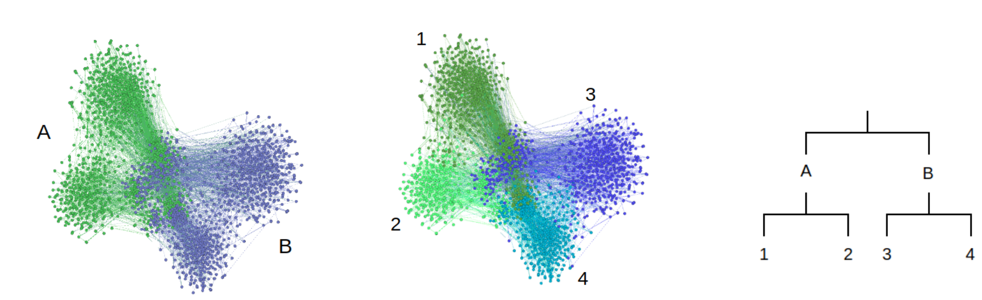


Figure 2: We visualize the result on a subset of Dblp dataset using two-level hierarchical vGraph. In the leftmost panel we visualize the first-tier communities. In the middle panel, we visualize the second-tier communities. In the rightmost panel we show the corresponding hierarchical tree structure.