S^3GC: Scalable Self-Supervised Graph Clustering

Devvrit*[1], Aditya Sinha†[2], Inderjit Dhillon*‡, Prateek Jain†

*University of Texas at Austin
†Google Research, India
‡Google

October, 2022

[1]work done while the author was an intern at Google Research
[2]Now at University of Illinois, Urbana-Champaign

Graph Data

$\rightarrow$ Graphs are a commonplace data structures.

$\rightarrow$ Usually nodes are often equipped with vector embeddings from different sources.

$\rightarrow$ Example: ogbn-arxiv - a citation graph where nodes can be equipped with embeddings of the title/content of the papers.

Problem Statement

Problem

Given a graph G with node set V, edges E, and node attributes $X \in \mathbb{R}^{n \times d}$, aim is to cluster the nodes into k clusters.

This problem of graph clustering with side information has been extensively studied in the literature [1]

Challenges

Existing solutions suffer from one of the following 3 limitations:

1. The clustering algorithm is highly reliant on either the graph structure or the node attributes.
2. The algorithm doesn't explicitly promote clusterability.
3. The method isn't scalable to realistic web-scale datasets.

S^3GC: Scalable Self-Supervised Graph Clustering

Consists of 3 components:

1. **A scalable encoder** that captures both node as well as graph structure information.
2. **A contrastive loss function** that ensures that the embedding of a node is close to "near-by" nodes, thus learning clusterable embeddings.
3. **Random Walk Sampler** to scalably generate positive samples from more than 1-hop neighbourhood.

Encoder

Given an adjacency matrix $A \in \mathbb{R}^{n \times n}$, define k-hop Diffusion matrix as

$$S_k = \sum_{i=0}^{k} A^i \tag{1}$$

Then, the encoder is defined as:

$$\overline{X} = \left(PReLU(AX\theta) + PReLU(S_k X\theta') + \mathcal{I} \right) \tag{2}$$

where $PReLU$ is the activation function, θ and θ' are learnable parameters, and $\mathcal{I}$ is a learnable $\mathbb{R}^{n \times d}$ matrix.

$\rightarrow \mathcal{I}$ supports in learning graph structure information. While $PReLU(AX\theta)$ and $PReLU(S_k X\theta')$ are 1-layer Graph Convolutional Neural Network that capture node attributes information.

Random Walk Sampler

$\rightarrow$ Given a pivot node u, the random walk sampler outputs all the distinct points seen in a random walk of length l started at u.

$\rightarrow$ These points act as "positive samples" to the node u, later used by contrastive loss. Negative samples are generated by picking r distinct random nodes in the graph.

Contrastive Loss

We use a SimCLR style loss function where positive and negative nodes are generated by the random walk sampler. Let p_u^+ be the set of positive nodes and p_u^- be the set of negative nodes. Then, the loss for node u is calculated as:

$$SimCLR\_Loss = \frac{\sum_{v \in p_u^+} exp(sim(u, v))}{\sum_{v \in p_u^+} exp(sim(u, v)) + \sum_{v' \in p_u^-} exp(sim(u, v'))} \quad (3)$$

where *sim* is some similarity function, for example inner product $sim(u, v) = \frac{u^T v}{\|u\|\|v\|}$

Overall Algorithm

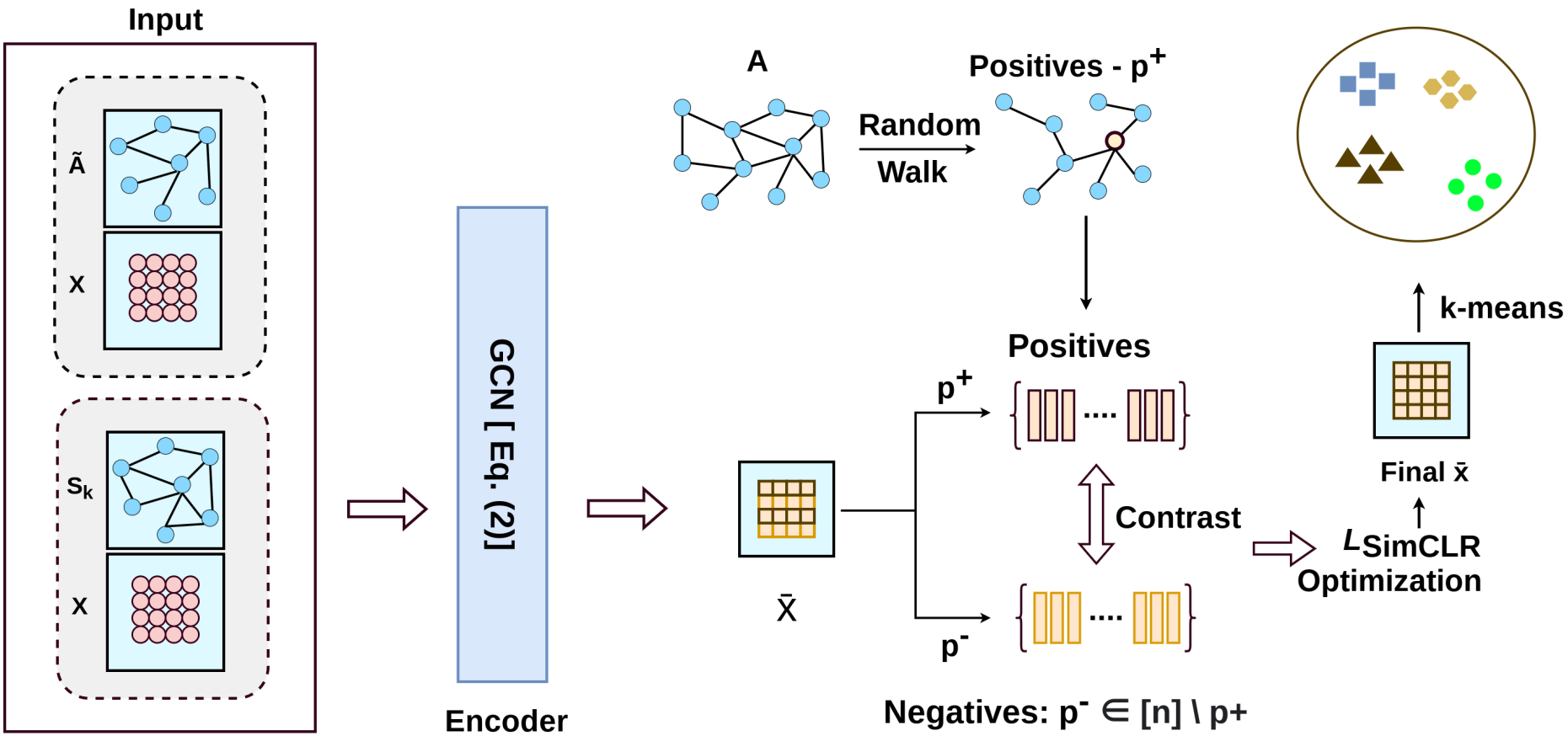

Figure: S^3GC Algorithn

Results on real world datasets

Table: **Comparison of clustering obtained by our method to several state-of-the-art methods.** We use the official implementations provided by the authors for all the methods. * denotes that the method ran Out of Memory (OOM) while trying to run the experiments on the hardware. – indicates that the method did not converge.

Dataset	Metric	Baseline									Ours
		k-means	MinCutPool	METIS	Node2vec	DGI	DMoN	GRACE	BGRL	MVGRL	**S^{3}GC**
Pubmed	NMI	0.314	0.214	0.297	0.288	0.322	0.257	0.308	0.315	**0.345**	0.333
ogbn-arxiv	NMI	0.216	0.380	0.345	0.406	0.412	0.356	*	0.321	*	**0.463**
Reddit	NMI	0.114	–	0.727	**0.792**	0.306	0.628	*	*	*	**0.807**
ogbn-products	NMI	0.273	0.430	0.468	0.489	0.467	0.428	*	*	*	**0.536**

Scaling up to 100M nodes and 1.6B edges

Table: Results of comparison of the embeddings generated by our method as compared to different scalable methods on ogbn-papers100M with 111M nodes and 1.6B edges.

Method	ogbn-papers100M				
	Accuracy	NMI	CS	F1	ARI
k-means	0.144	0.368	0.342	0.101	0.074
Node2vec	**0.175**	0.380	0.352	**0.099**	**0.112**
DGI	0.151	0.416	0.386	**0.111**	0.096
S^{3}GC (Ours)	**0.173**	**0.453**	**0.430**	**0.118**	**0.110**

Conclusion

1. We introduced S^3GC, a new method for scalable graph clustering with node feature side-information.
2. Our choice of the encoder, positive and negative node samples, and the loss function - makes our method scalable as well as generate clusterable embeddings.
3. S^3GC is able to scale to graphs with 100M nodes while still ensuring SOTA clustering performance.

Bibliography

[1] Yuchen Zhao and Philip S Yu. On graph stream clustering with side information. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 139–150. SIAM, 2013.