Graphite: Iterative Generative Modeling of Graphs

Aditya Grover, Aaron Zweig, Stefano Ermon
Computer Science Department
Stanford University
Graphs are ubiquitous

How do we learn representations of nodes in a graph?

Useful for several prediction tasks. E.g., friendship links on social networks (link prediction), living status of organisms in ecological networks (node classification)

Social, biological, information networks etc.
Latent Variable Model of a Graph

- Graphs are represented as adjacency matrices $A \in \{0,1\}^{n \times n}$.
- For every node $i$, we associate a latent vector representation $z_i \in \mathbb{R}^k$.

Example graph

Adjacency matrix

\[
A = \begin{bmatrix}
0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 \\
\end{bmatrix}
\]

Latent feature matrix

\[
Z = \begin{pmatrix}
\mathbf{z}_1^T \\
\mathbf{z}_2^T \\
\mathbf{z}_3^T \\
\mathbf{z}_4^T \\
\end{pmatrix}
\]
Graphite: A VAE for Graphs

latent matrix $Z \in \mathbb{R}^{n \times k}$

Decoder: Generate data $p_{\theta}(A | Z)$

adjacency matrix $A \in \{0,1\}^{n \times n}$
Graphite: A VAE for Graphs

Latent matrix $Z \in \mathbb{R}^{n \times k}$

Decoder: Generate data

$p_\theta(A | Z)$

Adjacency matrix $A \in \{0, 1\}^{n \times n}$

Encoder: Infer representations

$q_\phi(Z | A)$
Graphite: Learning & Inference

Given: Dataset of adjacency matrices, $D_A$

$p_\theta(A \mid Z)$  $q_\phi(Z \mid A)$
Graphite: Learning & Inference

Given: Dataset of adjacency matrices, $D_A$

Learning objective: $\max_{\theta, \phi} \text{ELBO}(\theta, \phi; D_A)$
Graphite: Learning & Inference

Given: Dataset of adjacency matrices, $D_A$

Learning objective: $\max_{\theta, \phi} \text{ELBO}(\theta, \phi; D_A)$

Test time use cases

Generative modeling tasks
- Density estimation, clustering nodes, compressing graphs etc.

Graph tasks
- Link Prediction: Denoise graph
- Semi-supervised node classification: Feed $z_i$ for labelled nodes to a classifier
Parameterizing Graph Autoencoders

$Z \xrightarrow{q_{\phi}(Z|A)} A$

Encoding $q_{\phi}(Z|A)$: Graph Neural Network (GNN)

Graphite: Iterative Generative Modeling of Graphs
Parameterizing Graph Autoencoders

Encoding $q_\phi(Z|A)$: Graph Neural Network (GNN)

Decoding $p_\theta(A|Z)$: Challenging to “upsample” graphs given latent representations
Decoding Graphs - MLP

\[ Z \in \mathbb{R}^{n \times k} \]

\[ p_\theta(A | Z) \]

\[ A \in \{0,1\}^{n \times n} \]

Option 1: Multi-layer Perceptrons (MLP)

Simonovsky et al., 2018

\[ O(n^2d + dk) \] total parameters for single hidden layer of width \( d \)
Decoding Graphs - RNN

\[ Z \in \mathbb{R}^{n \times k} \]

\[ p_\theta(A | Z) \]

\[ A \in \{0,1\}^{n \times n} \]

Option 2: Recurrent Neural Network (RNN)

You et al., 2018

Arbitrary ordering of nodes required for training e.g., BFS, DFS

Graphite: Iterative Generative Modeling of Graphs
Graphite – Decoding Graphs using GNN

Key idea
Learn the low-rank structure of adjacency matrix $A$ in the latent space $Z$

$Z \in \mathbb{R}^{n \times k}$

$p_\theta(A | Z)$

$A \in \{0,1\}^{n \times n}$
Graphite – Decoding Graphs using GNN

For fixed number of iterations:

**Step 1 (Low rank matrix reconstruction)**
Map $Z$ to an intermediate graph $\hat{A}$ via an inner product

$$\hat{A} \approx ZZ^T$$
Graphite – Decoding Graphs using GNN

For fixed number of iterations:

Step 1 (Low rank matrix reconstruction)
Map $Z$ to an intermediate graph $\tilde{A}$ via an inner product
$$\tilde{A} \approx ZZ^T$$

Step 2 (Progressive refinement)
Refine $Z$ by message passing over $\tilde{A}$ using a GNN
$$Z = \text{GNN}_\theta(\tilde{A})$$
Graphite – Decoding Graphs using GNN

For fixed number of iterations:

Step 1 (Low rank matrix reconstruction)
Map $Z$ to an intermediate graph $\hat{A}$ via an inner product

$$\hat{A} \approx ZZ^T$$

Step 2 (Progressive refinement)
Refine $Z$ by message passing over $\hat{A}$ using a GNN

$$Z = \text{GNN}_\theta(\hat{A})$$

Output step: Set $p_\theta(A | Z) = \text{Bernoulli}(\text{sigmoid}(ZZ^T))$
Graphite – Decoding Graphs using GNN

Unlike MLP, GNN decoder with single hidden layer of length $d$ has $O(dk)$ parameters.

- Unlike RNN, no arbitrary ordering of input nodes is required.

Decoding is also computationally efficient. See paper for details.
Empirical Results – Density Estimation

Baseline VGAE [Kipf et al., 2016]
GNN Encoder + Non-learned Inner Product Decoder. No iterative refinement.

![Bar chart showing negative log-likelihoods. Lower is better.]

Graphite: Iterative Generative Modeling of Graphs
Empirical Results – Link Prediction

AUC. Higher is better.

SpecCluster  DeepWalk  node2vec  VGAE  Graphite

Cora  Citeseer  Pubmed

Graphite: Iterative Generative Modeling of Graphs
Empirical Results – Semi-supervised Node Classification

Percentage accuracy. Higher is better.

Graphite: Iterative Generative Modeling of Graphs
**Summary**

**Graphite**: A latent variable generative model for graphs where both encoder and decoder are parameterized by graph neural networks.

- **Encoder** performs message passing on input graph
- **Decoder** iteratively refines inner product graphs

For more details, please visit us at Poster #7.

Code: [https://github.com/ermongroup/graphite](https://github.com/ermongroup/graphite)