THE INFINITY MIRROR TEST FOR GRAPH GENERATORS

Satyaki Sikdar, Daniel Gonzalez, and Tim Weninger

SIAM Workshop on Network Science 2020
July 9–10 · Toronto

Summary

We propose a stress test for evaluating graph models. This test iteratively and repeatedly fits a model to itself, exaggerating models' implicit biases.

Graph Generators

Graph models extract meaningful features Θ from a graph G and are commonly evaluated by generating a new graph $\tilde{G}$. Early models, like the Erdös-Rényi and Watts-Strogatz models, depend on preset parameters to define the model. More recent approaches aim to learn Θ directly from G to generate $\tilde{G}$.

The Chung-Lu model, for example, generates $\tilde{G}$ by randomly rewiring edges based on G's degree sequence [2]. The Stochastic Block Model (SBM) divides the graph G into blocks and generates new graphs $\tilde{G}$ respecting the connectivity patterns within and across blocks [4]. Graph grammars extract a list of node or hyperedge replacement rules from G and generate $\tilde{G}$ by applying these grammar rewriting rules [1, 7]. Recently, graph neural network architectures have also gained popularity for graph modeling. For example, Graph Variational Auto-Encoders (GVAE) learn a latent node representation from G and then sample a new graph $\tilde{G}$ from the latent space [5]. NetGAN learns a Generative Adversarial Network (GAN) on both real and synthetic random walks over G and then builds $\tilde{G}$ from a set of plausible random walks [3].

Each of these models has its advantages, but each model may induce modeling biases that common evaluation metrics are unable to demonstrate. In the present work, we describe the Infinity Mirror Test to determine the robustness of various graph models. We characterize the robustness of a graph generator by its ability to repeatedly learn and regenerate the same model. The *infinity mirror test* is designed to that end.

Infinity Mirror Test

Named after a common toy, which uses parallel mirrors to produce infinitely-repeating reflections, this methodology trains a model $\mathcal{M}$ on an input graph G_0 and generates a new graph $\tilde{G}$ and repeats this process iteratively on $\tilde{G}$ for a total of k generations, obtaining a sequence of graphs $\langle G_1, G_2, \cdots, G_k \rangle$. This process is illustrated in Figure 1.

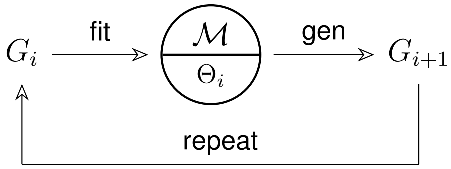

Figure 1: The idea behind the infinity mirror is to iteratively to fit our model $\mathcal{M}$ on G_i, use the fitted parameters Θ_i to generate the next graph G_{i+1}, and repeat with G_{i+1}.

The infinity mirror methodology produces a sequence of generated graphs, each graph prediction based on a model of the previous prediction. Like repeatedly compressing a JPEG image, we expect that graphs generated later in the sequence will eventually degenerate. However, much can be learned about hidden biases or assumptions in the model by examining the sequence of graphs before degeneracy occurs.

We illustrate some initial results in Figure 2 using three synthetic input graphs with easily-identifiable visual structure $G_0 = \{$a 10×10 grid graph, a 25-node ring of 4-cliques, and a synthetic graph with 3 well-defined communities$\}$. We consider the following graph models $\mathcal{M} = \{$Chung-Lu, degree-corrected SBM (DC-SBM), Hyperedge Replacement Grammars (HRG), Clustering-based Node Replacement Grammars (CNRG), GVAE, and NetGAN$\}$.

For each combination of input graph G_0 and generative model $\mathcal{M}$, we generate 50 independent chains of length $k = 20$ and compute the DeltaCON score comparing G_0 and G_k. We select the chain resulting in the median DeltaCON score, and illustrate G_1, G_5, and G_{20} (i.e., the 1st, 5th, and 20th generations) in Figure 2 if they exist.

Main Results and Discussions

These initial results show that CNRG can capture and maintain certain structures from the input graph: on the ring of cliques we see perfect replication, as well as on the first generation of community, with some marginal degradation as the number of repetitions increases. SBM performs similarly-well on such structured input. Neither model works particularly well on grids; CNRG introduces triangles, while SBM creates nodes of degree 1.

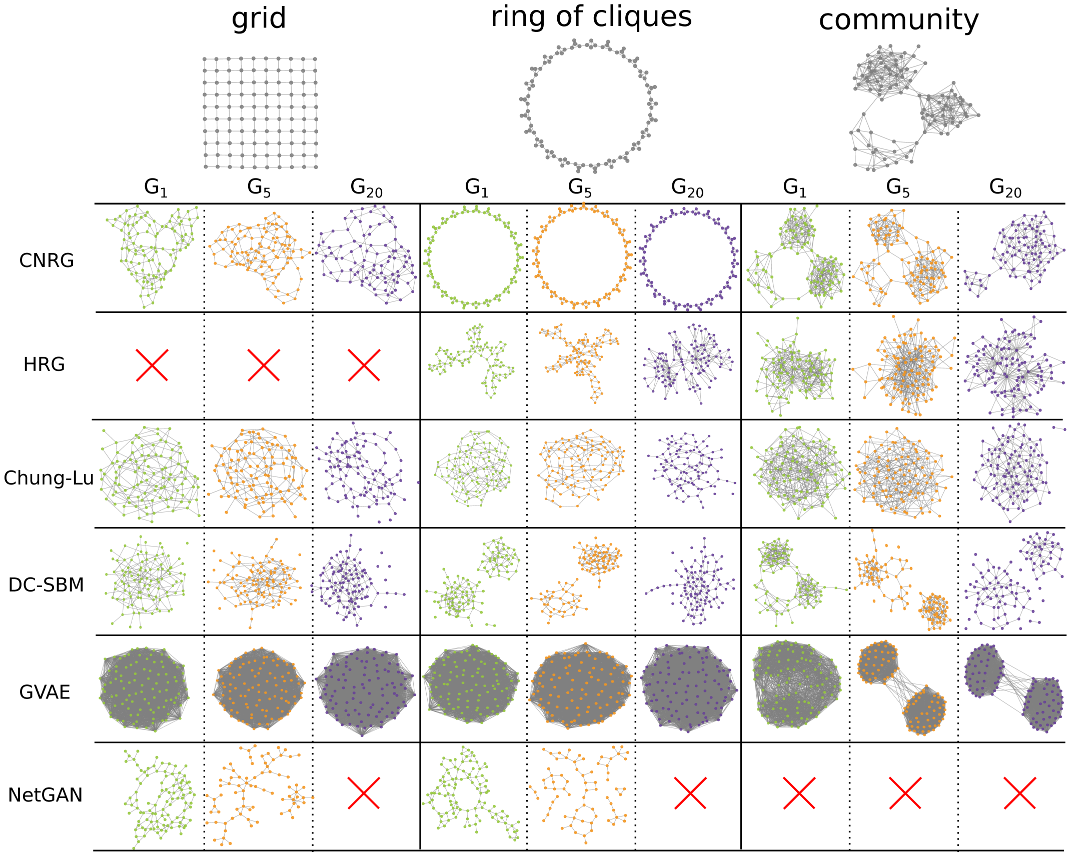

Figure 2: (Best viewed in color.) Plot showing the evolution of various graph models at generations 1, 5, and 20 (G_1, G_5, G_{20}) on three types of synthetic graphs. Red × indicates model fit failure.

HRG fails entirely for grids and does not appear to perform well on the other two graph types.

The Chung-Lu model mimics the degree distribution of the input as we would expect, but fails to capture the network structure. We can see long chains formed from the grid input by the last generation, and the output of the other two graph types do not resemble the original graph. We also see that Chung-Lu fails to preserve the connectivity of the graph, as shown by the disconnected components in the last generation of each column.

GraphVAE results in overly dense graphs regardless of the input, which obfuscates any topological structure the input graph had.

NetGAN tends to produce sparse, tree-like graphs. This becomes a problem down the line. The way that NetGAN computes its train/test/validation split relies on finding a spanning tree on the input graph; when the model is recursively applied to its increasingly tree-like outputs, it will eventually fail to find a satisfactory split, causing the model to fail to fit.

The infinity mirror test we proposed exposes biases in models that might otherwise go unnoticed. Future work will determine how best to use this methodology to analyze biases and errors in graph models.

References

[1] S. Aguinaga, D. Chiang, and T. Weninger. Learning hyperedge replacement grammars for graph generation. *IEEE TPAMI*, 2018.

[2] W. Aiello, F. Chung, and L. Lu. A random graph model for power law graphs. *Experimental Mathematics*, 10(1):53–66, 2001.

[3] A. Bojchevski, O. Shchur, D. Zügner, and S. Günnemann. Netgan: Generating graphs via random walks. *arXiv:1803.00816*, 2018.

[4] B. Karrer and M. E. Newman. Stochastic blockmodels and community structure in networks. *Physical review E*, 83(1):016107, 2011.

[5] T. N. Kipf and M. Welling. Variational graph auto-encoders. *arXiv:1611.07308*, 2016.

[6] D. Koutra, J. T. Vogelstein, and C. Faloutsos. Deltacon: A principled massive-graph similarity function. In *SDM*, 2013.

[7] S. Sikdar, J. Hibshman, and T. Weninger. Modeling graphs with vertex replacement grammars. In *IEEE ICDM*, 2019.