Conditional Structure Generation through Graph Variational Generative Adversarial Nets

Carl Yang, Peiye Zhuang, Wenhan Shi, Alan Luu, Pan Li (corresponding: jiyang3@illinois.edu)

University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Problem Formulation

Conditional Structure Generation: Given a set of graphs with semantic contexts, learn to generate graphs of meaningful structures with related contexts.

Motivations
1) Decompose massive networks into small subnetworks with clear structures and contexts
2) Map network structures and context semantics in embedding spaces
3) Flexibly generate network structures under given semantics

Challenges and Requirements

Flexible context-structure conditioning: Learn a single representation from a set of graphs with variable sizes.

Permutation-invariant graph generation: Capture unique graph representations regardless of node ordering.

Technical Contributions

Contribution 1: A novel GCN-VAE framework with flexible conditioning function.

\[ p(A|Z) = \prod_{i=1}^{n} \prod_{j=1}^{n} p(A_{ij}|z_i, z_j) \] \[ = \sigma(f(z_i)^T f(z_j)) \]

Learn with sets of graphs
Graph-level conditioning Permutation-invariant encoding

Figure: Use a single distribution to jointly model all nodes.

Contribution 2: A novel GCN-based graph discriminator to enable permutation invariance

\[ \mathcal{L}_{gen} = \log D(A) + \log(1 - D(G(Z_i))) + \log(1 - D(G(Z_j))) \]

Overall Architecture of CondGen: GCN-VAE-GAN.

Figure: End-to-end learnable graph structure representations.

Experimental Evaluations

Datasets: We created two benchmark datasets, i.e., a set of author citation networks from DBLP and a set of gene interaction networks from TCGA.

Baseline: We carefully adapt three state-of-the-art graph generation methods, i.e., GVAE, NetGAN and GraphRNN, by concatenating the condition vectors to both the node features of the input graph and the output of the last encoding layer.

Protocols: We evaluate both tasks of mimicking similar seen graphs and creating novel unseen graphs, through visual inspection and graph property comparison (statistics we use include LCC (size of largest connected component), TC (triangle count), CPL (characteristic path length), MD (maximum node degree) and GINI (gini index), measuring different properties of graphs).

Table: Performance evaluation over compared algorithms regarding several important graph statistical properties. The Real rows include the values of real graphs, while the rest are the absolute values of differences between graphs generated by each algorithm and the real graphs. Therefore, smaller values indicate higher similarities to the real graphs, thus better overall performance. We conduct paired t-test between each baseline and CondGen(S), scores with * indicate p < 0.05 and ** indicate p < 0.01, respectively.

More results: For more experimental results on runtimes, visual inspections, and training details, please refer to our paper and supplementary materials at http://jiyang3.web.engr.illinois.edu/.