Explore Deep Graph Generation

Xiuye Gu
xiuyegu@stanford.edu

CS 224W
Fall 2019

Abstract

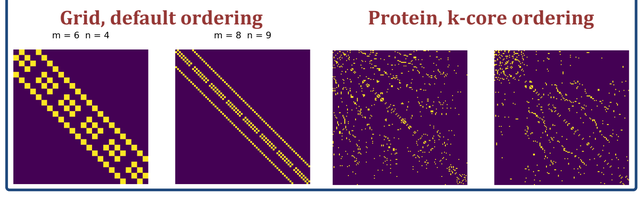
We explore deep graph generation from two directions:
1) we use CNN GANs to model the whole adjacency matrix directly after sorting the nodes;
2) we built upon the very recent Graph Recurrent Attention Networks (GRANs), proposed a graph completeness judger network and improved its attention mechanism.
The first direction works on small grid graphs to some extent but fails to work on large graphs and we analyze its failure.
For the second direction, experiments on the Grid and Protein datasets show that our improved version outperforms the original approach and the completeness judger is effective.

Motivation

- *"What I cannot create, I do not understand."* -- Richard Feynman.
- Twofold advantages: 1) discover new realistic graphs and benefit downstream applications, e.g., drug design, protein study; 2) better null models for network analysis.
- Deep generative models enjoy success in CV and NLP.
- 3 categories: autoregressive, VAEs, GANs.

Dataset

- **Grid:** 100 standard 2D grid graphs. $100 \leq |V| \leq 400$.
- **Protein:** 918 protein graphs. Two nodes (amino acids) are connected if < 6 Angstroms away. $100 \leq |V| \leq 500$.
- Plot of adjacency matrices after sorting nodes:

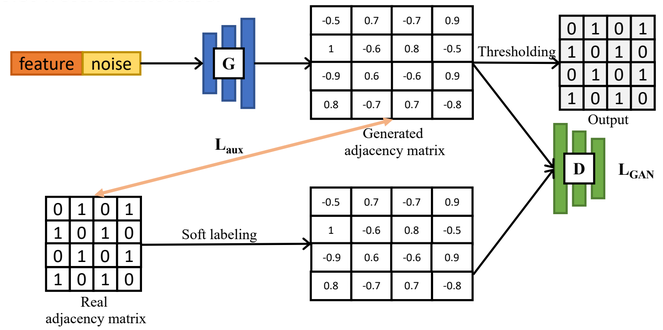


Future work

There is still a long way to go on deep generative models for graphs. It's less successful than generative models on images and texts/audios. Generation on graphs are much harder due to complex topological structure.
Because of the inherent limitations of auto-regressive models, though our first GAN approach failed, for future work we think combining the advantages of the two approaches is worthwhile, for example, GANs with a GNN generator and a GNN discriminator.

First direction: CNN GANs on adjacency matrices

- Net work architecture:



Feature & noise: input. 1D vector. We use one-hot feature vector.
G: generator network, adapt from DCGAN.
D: discriminator network, adapt from PatchGAN.

- Adjacency matrices have obvious patterns after sorting the nodes using some node orderings.
- So we propose to adopt CNN GANs to generate adjacency matrices, like image generation.
- Generative adversarial network: a discriminator learns to distinguish real and fake samples and a generator learns to fool the discriminator. A two-player minimax game:

$$\min_D V_{LSGAN}(D) = \frac{1}{2}\mathbb{E}_{x\sim p_{data}(x)}[(1-D(x))^2] + \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}[D(G(z))^2],$$
$$\min_G V_{LSGAN}(G) = \mathbb{E}_{z\sim p_z(z)}[(1-D(G(z)))^2].$$

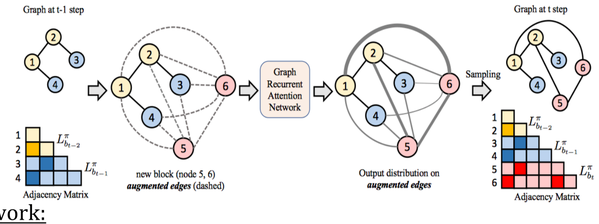
- Cannot generate exact binary outputs. Adopt soft labeling before sending real to D:

$$S(x) = \begin{cases} (1-\alpha)+\alpha\epsilon, & \text{if } x == 1 \\ -((1-\alpha)+\alpha\epsilon), & \text{otherwise (x == 0)} \end{cases}$$

- Inspired by InfoGAN, adopt L_{aux} to increase mutual information and address mode collapse.

Second direction: Improving GRAN

- GRAN review:
  - Generate one block of nodes at a time.
  - Use a GNN with attention to predict edge distributions.
  - Each step doesn't depend on previous hidden states.



- Graph completeness judger network:
  - GRAN doesn't know when to stop, relies on sampling the graph size based on the training data statistics (a multinomial distribution).
  - We propose to train a completeness judger network (3-layer GCN + Global max pooling + 2-layer MLP) to score if the input graph is sampled from the given graph distribution.
  - Then compute the score for all subgraphs, and use the scores to form a categorical distribution for graph sampling.
- Improving attention mechanism:
  - Original a_{ij}^r and m_{ij}^r use element-wise multiplication -- problematic. Attention doesn't utilize graph structure. We improve it and adopt multi-head attention.

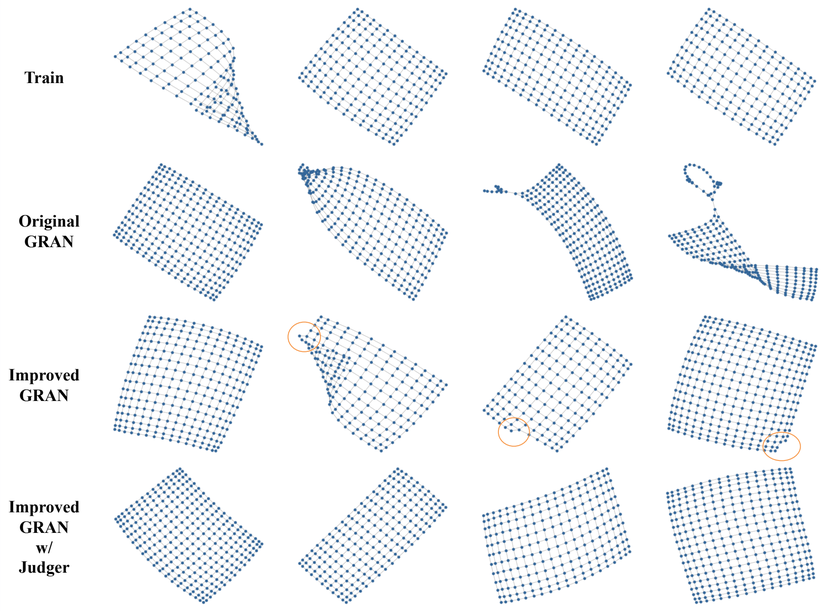
$$m_{ij}^r = f(h_i^r - h_j^r),$$
$$\tilde{h}_i^r = [h_i^r, x_i],$$
$$a_{ij}^r = \text{Sigmoid}\big(g(\tilde{h}_i^r - \tilde{h}_j^r)\big),$$
$$h_i^{r+1} = \text{GRU}(h_i^r, \sum_{j\in\mathcal{N}(i)} a_{ij}^r \otimes m_{ij}^r),$$

$$\Rightarrow$$

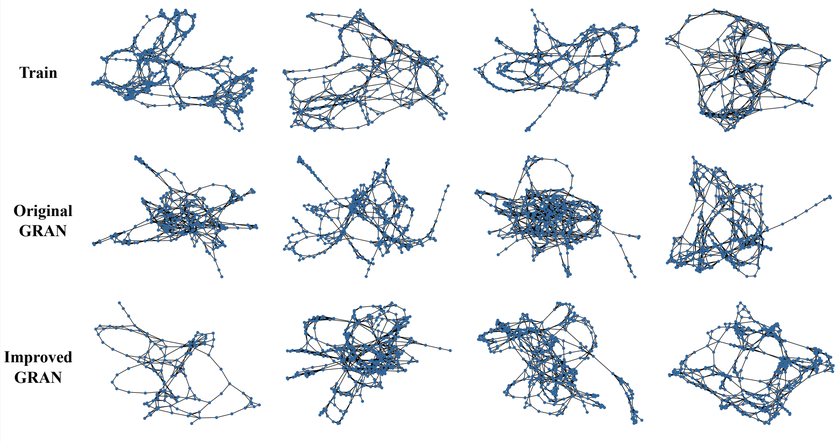
$$m_{ij}^{r,k} = f_k([h_i^r, h_j^r]),$$
$$\tilde{h}_i^r = [h_i^r, x_i],$$
$$a_{ij}^{r,k} = \text{Masked\_Softmax}\big(g_k([\tilde{h}_i^r, \tilde{h}_j^r])\big) = \frac{\exp\big(g_k([\tilde{h}_i^r, \tilde{h}_j^r])\big)}{\sum_{l\in\mathcal{N}_i}\exp\big(g_k([\tilde{h}_i^r, \tilde{h}_l^r])\big)},$$
$$h_i^{r+1} = \text{GRU}\Big(h_i^r, \|_{k=1}^K\big(\sum_{j\in\mathcal{N}(i)} a_{ij}^{r,k} m_{ij}^{r,k}\big)\Big).$$

Results

- Qualitative results on Grid



- Qualitative results on Protein



- Maximum mean discrepancy (MMD) over graph statistics

	Grid ($\lvert V\rvert_{max}$ = 361, $\lvert E\rvert_{max}$ = 684, $\lvert V\rvert_{avg} \approx 210$, $\lvert E\rvert_{avg} \approx 392$)				Protein ($\lvert V\rvert_{max}$ = 500, $\lvert E\rvert_{max}$ = 1575, $\lvert V\rvert_{avg} \approx 258$, $\lvert E\rvert_{avg} \approx 646$)			
	Deg.	Clus.	Orbit	Spec.	Deg.	Clus.	Orbit	Spec.
GRAN	$6.84e^{-4}$	**0**	$1.45e^{-3}$	$1.50e^{-2}$	$6.91e^{-3}$	$9.30e^{-2}$	$7.75e^{-2}$	$4.74e^{-3}$
Improved	$1.42e^{-4}$	**0**	$2.41e^{-4}$	**$8.48e^{-3}$**	**$6.57e^{-3}$**	**$7.44e^{-2}$**	**$5.92e^{-2}$**	**$4.71e^{-3}$**
w/ Judger	**$4.33e^{-6}$**	$2.71e^{-5}$	**$3.19e^{-6}$**	$1.35e^{-2}$	-	-	-	-

Deg.: degree distribution.
Clus.: clustering coefficients.
Orbit: the number of 4-node orbits.
Spec.: spectrum of graph Laplacian.
All metrics the smaller the better.