2019sp-cs-8501-Deep2Read Scribe Notes: GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders

Scribe: Arshdeep Sekhon

June 1, 2019

1 Motivation

Most graph generation methods follow a sequential approach to graph generation. In contrast, GraphVAE generates a probabilistic fully connected graph of a predefined maximum size directly at once. This avoids problems associated with non differentiable discrete decision making and ordering associated with sequential graph generation.

2 Method

The main idea is to map graphs to a continuous vector space, and output a probabilistic fully-connected graph from this representation. This output graph is compared to the true graph using a standard graph matching algorithm. Specifically, they use a VAE to map a graph to a latent 'z' and decode to a probabilistic graph defined by (A,E,F), where A denotes adjacency matrix, E denotes a tensor describing edge types of the graph, and F denotes features of the nodes.

Variational AutoEncoder Objective is:

$$L(\phi, \theta;) = E_{q_\phi(z|G)}[-logp_\theta(G|z)] + KL(q_\phi(z|G)||p(z)) \tag{1}$$

The first term of L, the reconstruction loss, enforces high similarity of sampled generated graphs to the input graph G. The second term, KL-divergence, matches a simpler p(z) (Gaussian distribution) to $q_\phi(z|G)$.

2.1 Reconstruction Loss using Graph Matching

Approximate Graph Matching is used to construct reconstruction loss for graph generation. Because nodes are unordered, the correspondence between predicted graph nodes and true graph nodes is done using approximate graph matching.

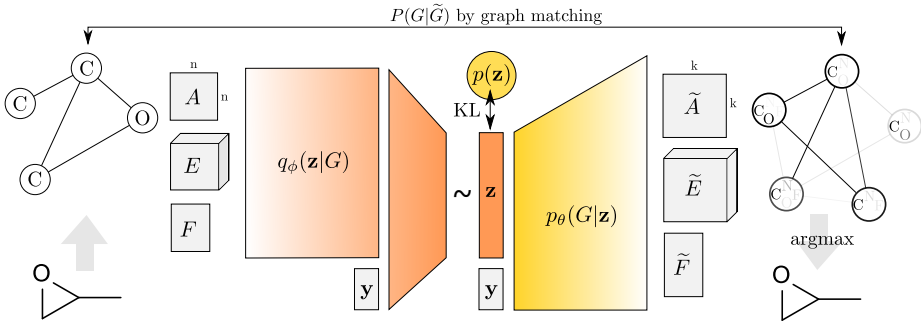

Figure 1: VAE to generate probabilistic graph: Encoder $q_\phi(z|G)$ maps input graph G to latent continuous vector representation z. Decoder $p_\theta(G|z)$ decodes z to a graph $\hat{G}$ defined by (A,E,F), where A denotes adjacency matrix, E denotes a tensor describing edge types of the graph, and F denotes features of the nodes.

This gives a binary assignment matrix $X \in \{0,1\}^{k \times n}$ where $X_{a,i} = 1$ if node $a \in \tilde{G}$ is assigned to $i \in G$ and $X_{a,i} = 0$ otherwise.

$$-log\ p(G|z) = -log\ p(A'|z) - log\ p(F|z) - log\ p(E|z) \tag{2}$$

where $A' = XAX^T$ whereas the predicted node attribute matrix and slices of edge attribute matrix are transferred to the input graph as $F' = X^T\tilde{F}$. The maximum likelihood estimates are:

$$log\ p(A'|z) = 1/k_a A'_{a,a} log\tilde{A_{a,a}} + (1-A'_{a,a}) log(1-\tilde{A_{a,a}}) + 1/k(k-1)_a A'_{a,b} log\tilde{A_{a,b}} + (1-A'_{a,b}) log(1-\tilde{A_{a,b}}) \tag{3}$$

$$log(p(F|z)) = 1/n_i log(F_{i,}^T)\tilde{F}'_{i,} \tag{4}$$

$$log(p(E|z)) = 1/(||A||_1 - n)_{i \neq j} log E_{i,j,}^T \tilde{E}'_{i,j,} \tag{5}$$

3 Evaluation

Evaluation is done on QM9 and Zinc datasets evaluated on mean test-time reconstruction log-likelihood, mean test-time evidence lower bound (ELBO), and decoding quality metrics.

		$\log p_\theta(G\|\mathbf{z})$	ELBO	Valid	Accurate	Unique	Novel
Cond.	Ours $c = 20$	-0.578	-0.722	0.565	0.467	0.314	0.598
	Ours $c = 40$	-0.504	-0.617	0.511	0.416	0.484	0.635
	Ours $c = 60$	-0.492	-0.585	0.520	0.406	0.583	0.613
	Ours $c = 80$	-0.475	-0.557	0.458	0.353	0.666	0.661
Unconditional	Ours $c = 20$	-0.660	-0.916	0.485	0.485	0.457	0.575
	Ours $c = 40$	-0.537	-0.744	0.542	0.542	0.618	0.617
	Ours $c = 60$	-0.486	-0.656	0.517	0.517	0.695	0.570
	Ours $c = 80$	-0.482	-0.628	0.557	0.557	0.760	0.616
	NoGM $c = 80$	-2.388	-2.553	0.810	0.810	0.241	0.610
	CVAE $c = 60$	–	–	0.103	0.103	0.675	0.900
	GVAE $c = 20$	–	–	0.602	0.602	0.093	0.809

Figure 2: QM9 Dataset Evaluation

Noise	$k = 15$	$k = 20$	$k = 25$	$k = 30$	$k = 35$	$k = 40$
$\epsilon_{A,E,F} = 0$	99.55	99.52	99.45	99.4	99.47	99.46
$\epsilon_A = 0.4$	90.95	89.55	86.64	87.25	87.07	86.78
$\epsilon_A = 0.8$	82.14	81.01	79.62	79.67	79.07	78.69
$\epsilon_E = 0.4$	97.11	96.42	95.65	95.90	95.69	95.69
$\epsilon_E = 0.8$	92.03	90.76	89.76	89.70	88.34	89.40
$\epsilon_F = 0.4$	98.32	98.23	97.64	98.28	98.24	97.90
$\epsilon_F = 0.8$	97.26	97.00	96.60	96.91	96.56	97.17

Figure 3: Zinc Dataset Evaluation

4 Conclusion

A VAE based approach to generate probabilistic graphs in one go, isntead of a sequential node and edge adding. The training is dependent on the robustness of graph matching used for constructing reconstruction loss. This method fails to account for complex edge dependencies.