

# 论文阅读报告

## 一、论文基本情况

论文名	GraphGAN: Graph Representation Learning with Generative Adversarial Nets				
作者	Hongwei Wang	Jia Wang	Jialin Wang	Miao Zhao	Weinan Zhang
作者机构	1. Shanghai Jiao Tong University 2. Microsoft Research Asia 3. The Hong Kong Polytechnic University 4. Huazhong University of Science and Technology				
发表年份	2017 年 11 月 22 日				
来源（刊物名、会议名）	会议：AAAI				
论文可信力	期刊	EI/SCI		影响因子	
		Jcr 分区		CCF 分类	
	会议	CCF 分类	A	领域影响力	顶级 AI 会议
	引用次数				
阅读人	孙裕道	阅读日期		2018 年 12 月 25 日	

## 二、论文主要工作

### 1. 论文概述

该论文针对图网络表示学习中生成式模型和判别式模型的这两大类的应用的分析。提出一种新的结合了生成式模型和判别式模型的思想的网络表示学习的框架，实验结果 GraphGAN 的实验结果明显优于 Deepwalk、Node2vec、LINE 和 Struct2vec 的结果。

### 2. 论文提出的方法

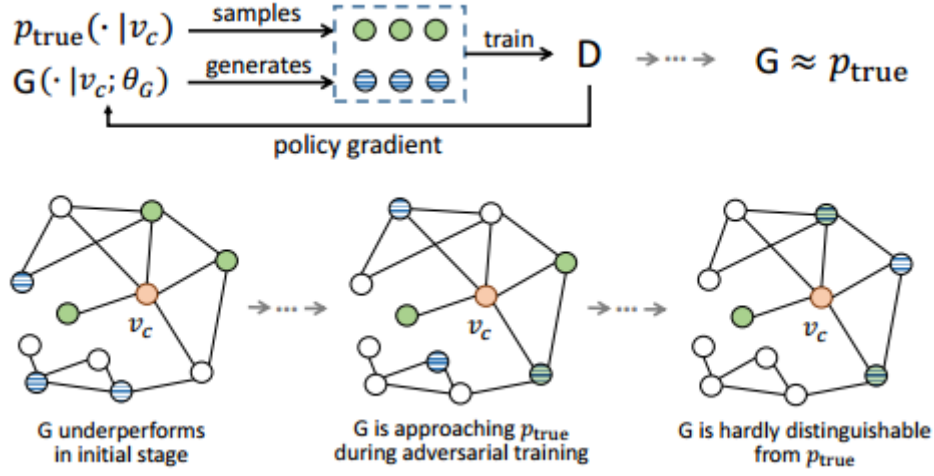
本文通过设计生成器  $G$  对于给定节点  $v$  拟合它与图中其他节点的潜在的真实的连通性分布，其目的是由生成策略生成节点能够欺骗过判别器  $D$ 。同时与之对应的判别器  $D$  的目的就是判别选取的节点是从真实分布中选取的还是由生成器  $G$  生成的节点。通过这种对抗的思想，经过反复迭代同时提高两个模型的性能。

#### (1) Graph Generative Adversarial

利用生成对抗网络中的这个 minimax-game 式子刻画了生成器和判别器的这种对抗关系，其对抗损失为：

$$\min_{\theta_G} \max_{\theta_D} V(G, D) = \sum_{c=1}^V \left( \mathbb{E}_{v \sim p_{\text{true}}(\cdot | v_c)} [\log D(v, v_c; \theta_D)] + \mathbb{E}_{v \sim G(\cdot | v_c; \theta_G)} [\log (1 - D(v, v_c; \theta_D))] \right).$$

其训练过程如下：



## (2) Discriminator Optimization

本篇论文设计的判别器是 sigmoid 函数：

$$D(v, v_c) = \sigma(\mathbf{d}_v^\top \mathbf{d}_{v_c}) = \frac{1}{1 + \exp(-\mathbf{d}_v^\top \mathbf{d}_{v_c})}$$

其中， $\mathbf{d}_v$  和  $\mathbf{d}_{v_c}$  是通过训练得到的  $v$  节点和  $v_c$  的表示向量 embedding。具体的参数优化按照如下的梯度上升策略进行：

$$\nabla_{\theta_D} V(G, D) = \begin{cases} \nabla_{\theta_D} \log D(v, v_c), & \text{if } v \sim p_{\text{true}}; \\ \nabla_{\theta_D} (1 - \log D(v, v_c)), & \text{if } v \sim G \end{cases}$$

## (3) Generator Optimization

生成器的优化要采用 policy gradient 策略进行为：

$$\begin{aligned} & \nabla_{\theta_G} V(G, D) \\ &= \nabla_{\theta_G} \sum_{c=1}^V \mathbb{E}_{v \sim G(\cdot | v_c)} [\log (1 - D(v, v_c))] \\ &= \sum_{c=1}^V \sum_{i=1}^N \nabla_{\theta_G} G(v_i | v_c) \log (1 - D(v_i, v_c)) \\ &= \sum_{c=1}^V \sum_{i=1}^N G(v_i | v_c) \nabla_{\theta_G} \log G(v_i | v_c) \log (1 - D(v_i, v_c)) \\ &= \sum_{c=1}^V \mathbb{E}_{v \sim G(\cdot | v_c)} [\nabla_{\theta_G} \log G(v | v_c) \log (1 - D(v, v_c))]. \end{aligned}$$

#### (4) Graph Softmax for Generator

(4.1)本论文提出的生成器要满足三个性质：

a.正则化:生成器必须是一个有效的概率分布函数

b.图结构敏感:生成器在去拟合图的真实连通性分布的时候要注意保留图的结构特征，例如，对于图中的两个节点，它们的连通性概率要随着节点之间的最短路径的长度的增长而减小。

c.计算高效性:区别于对图中所有节点的 softmax,新的 softmax 函数的计算应该只涉及到图中的少量节点。

(4.2) 对于选定节点与其邻居节点间的关联性概率:

$$p_c(v_i|v) = \frac{\exp(\mathbf{g}_{v_i}^\top \mathbf{g}_v)}{\sum_{v_j \in \mathcal{N}_c(v)} \exp(\mathbf{g}_{v_j}^\top \mathbf{g}_v)},$$

其中 $\mathbf{g}_v$ 和 $\mathbf{g}_{v_c}$ 是节点 $v$ 和 $v_c$ 的表示向量 *embedding*。根据关联性概率设计了如下的生成器：

$$G(v|v_c) \triangleq \left( \prod_{j=1}^m p_c(v_{r_j}|v_{r_{j-1}}) \right) \cdot p_c(v_{r_m-1}|v_{r_m})$$

(4.3)生成器的在线生成策略算法如下图：

---

#### Algorithm 1 Online generating strategy for the generator

---

**Require:** BFS-tree  $T_c$ , representation vectors  $\{\mathbf{g}_i\}_{i \in \mathcal{V}}$

**Ensure:** generated sample  $v_{gen}$

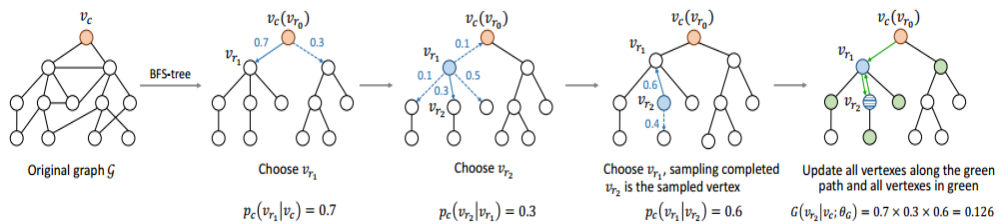
```

1:  $v_{pre} \leftarrow v_c, v_{cur} \leftarrow v_c;$ 
2: while true do
3:   Randomly select  $v_i$  proportionally to  $p_c(v_i|v_{cur})$  in Eq. (6);
4:   if  $v_i = v_{pre}$  then
5:      $v_{gen} \leftarrow v_{cur};$ 
6:     return  $v_{gen}$ 
7:   else
8:      $v_{pre} \leftarrow v_{cur}, v_{cur} \leftarrow v_i;$ 
9:   end if
10: end while

```

---

生成器的在线生成策略示意图如下：



### (4.3) GraphGAN framework

---

#### Algorithm 2 GraphGAN framework

---

**Require:** dimension of embedding  $k$ , size of generating samples  $s$ , size of discriminating samples  $t$   
**Ensure:** generator  $G(v|v_c; \theta_G)$ , discriminator  $D(v, v_c; \theta_D)$

- 1: Initialize and pre-train  $G(v|v_c; \theta_G)$  and  $D(v, v_c; \theta_D)$ ;
- 2: Construct BFS-tree  $T_c$  for all  $v_c \in \mathcal{V}$ ;
- 3: **while** GraphGAN not converge **do**
- 4:   **for** G-steps **do**
- 5:      $G(v|v_c; \theta_G)$  generates  $s$  vertices for each vertex  $v_c$  according to Algorithm 1;
- 6:     Update  $\theta_G$  according to Eq. (4), (6) and (7);
- 7:   **end for**
- 8:   **for** D-steps **do**
- 9:     Sample  $t$  positive vertices from ground truth and  $t$  negative vertices from  $G(v|v_c; \theta_G)$  for each vertex  $v_c$ ;
- 10:     Update  $\theta_D$  according to Eq. (2) and (3);
- 11:   **end for**
- 12: **end while**
- 13: **return**  $G(v|v_c; \theta_G)$  and  $D(v, v_c; \theta_D)$

---

### 3. 论文仿真实验及结果

本论文在五个数据集上进行了实验，用作对比的 baseline 算法分别选取了 Deepwalk、Node2vec、LINE 和 Struc2vec。实验表明，在 Link prediction、Node classification 和 Recommendation 任务中，GraphGAN 的实验结果优于上述 baseline 算法。

Table 1: Accuracy and Macro-F1 on arXiv-AstroPh and arXiv-GrQc in link prediction.

Model	arXiv-AstroPh		arXiv-GrQc	
	Acc	Macro-F1	Acc	Macro-F1
DeepWalk	0.841	0.839	0.803	0.812
LINE	0.820	0.814	0.764	0.761
Node2vec	0.845	0.854	0.844	0.842
Struc2vec	0.821	0.810	0.780	0.776
GraphGAN	<b>0.855</b>	<b>0.859</b>	<b>0.849</b>	<b>0.853</b>

Table 2: Accuracy and Macro-F1 on BlogCatalog and Wikipedia in node classification.

Model	BlogCatalog		Wikipedia	
	Acc	Macro-F1	Acc	Macro-F1
DeepWalk	0.225	0.214	0.194	0.183
LINE	0.205	0.192	0.175	0.164
Node2vec	0.215	0.206	0.191	0.179
Struc2vec	0.228	0.216	0.211	0.190
GraphGAN	<b>0.232</b>	<b>0.221</b>	<b>0.213</b>	<b>0.194</b>

### 三、存在问题

#### 1. 自己对该论文不太理解的地方

- (1) 图网络结构中，给定节点其连通性 $P_{true}(. / v_c)$ 分布是怎样定义的？
- (2) 图网络表示学习中，一个图节点原始数据的格式是什么形式的，其特征数的个数可以多少？
- (3) 图网络表示学习中，两个节点对应的 embedding 分别为 $d_v^T$  和  $d_{v_c}$ , 则 $d_v^T d_{v_c}$ 的内积表示的含义是什么？
- (4) GraphGAN 的算法 2 的框架中，生成器具体生成了一个节点原始特征数据还是一个节点的 embedding 的形式，以及生成一个节点是直接与 root 节点相连吗，生成一个节点后的其他节点会有怎样的更新变化，图网络的拓扑会发生改变吗？

#### 2. 该论文方法本身的问题

- (1) 文章设计的生成器无法作用在有权图上，要针对有权图的图表示学习需要重新设计生成器，不具有一般性。
- (2) 可能是数据集选取的原因，GraphGAN 的实验效果作用于节点分类任务中相较于其他的 baseline 算法虽然有提升，但是效果确实还是不够好。

### 四、论文评价

#### 1. 论文亮点

创新在于结合了生成式模型和判别式模型的思想来进行图网络的表示学习

#### 2. 评价

这篇文章将最近非常火的生成对抗网络 GAN 的思想运用在了网络表示学习中，是一个很新奇的想法。