vGraph: A Generative Model for Joint Community Detection and Node Representation Learning

A Preprint

Fan-Yun Sun[1,2], **Meng Qu**[2], **Jordan Hoffmann**[2,3], **Chin-Wei Huang**[2,4], **Jian Tang**[2,5,6]
[1]National Taiwan University,
[2]Mila-Quebec Institute for Learning Algorithms, Canada
[3]Harvard University, USA
[4]Element AI, Canada
[5]HEC Montreal, Canada
[6]CIFAR AI Research Chair
b04902045@ntu.edu.tw

Accepted by NIPS 2019

焦乙竹 2019.10.17

Introduction

- Task
  - Node embedding (node classification)
  - Community detection
- Why do together
  - node representations can be used as good features for community detection (e.g., through K-means).
  - community membership can provide good contexts for learning node embeddings.

Introduction

- Model —— vGraph
  - A probabilistic generative model
  - Learn community membership and node representation collaboratively
- Assumption
  - Each node can be represented as a mixture of communities
  - Each community is defined as a multinomial distribution over nodes

Introduction

- Model —— vGraph
  - 给定节点u，首先从分布$p(z|u)$分配社区z.
  - 给定分配的社区z，根据社区分布$p(v|z)$获得另一个节点v来生成边(u,v).
  - 分布$p(z|u)$和$p(v|z)$通过节点和社区的embedding来参数化
  - 节点表示和社区以互惠互利的方式进行交互

Introduction

- *Contributions*
  - 同时学习节点表示和社区发现的框架
  - 设计了一种反向传播推理算法，使用变分推理来最大化数据似然性的下限
  - 在变分推理例程的目标函数中添加了平滑度正则化项，以确保相邻节点的社区成员相似

Problem Definition

this paper, we study jointly solving these two tasks. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represent a graph, where $\mathcal{V} = \{v_1, \ldots, v_V\}$ is a set of vertices and $\mathcal{E} = \{e_{ij}\}$ is the set of edges. Traditional graph embedding aims to learn a node embedding $\boldsymbol{\phi}_i \in \mathbb{R}^d$ for each $v_i \in \mathcal{V}$ where d is predetermined. Community detection aims to extract the community membership $\mathcal{F}$ for each node. Suppose there are K communities on the graph $\mathcal{G}$, we can denote the community assignment of node v_i as $\mathcal{F}(v_i) \subseteq \{1, ..., K\}$. We aim to jointly learn node embeddings ϕ and community affiliation of vertices $\mathcal{F}$.

Node	w
Neighbor node	c
Community	z

Methodology

- Assumption
  - Each node can belong to multiple communities.
  - For each node, different neighbors will be generated depending on the community context.
- The linked neighbor c is generated based on the assignment z

$$p(c|w) = \sum_z p(c|z)p(z|w).$$

$c \sim p(c|z)$ $\quad$ $z \sim p(z|w)$

Methodology

Node	w	Embedding of node i in p(z|w)	$\boldsymbol{\phi}_i$
Neighbor node	c	Embedding of node i in p(c|z)	$\boldsymbol{\varphi}_i$
Community	z	Embedding of the j-th community	$\boldsymbol{\psi}_j$

- vGraph parameterizes p(z|w) and p(c|z) by node and community embeddings.
- The prior distribution

$$p_{\boldsymbol{\phi},\boldsymbol{\psi}}(z=j|w) = \frac{\exp(\boldsymbol{\phi}_w^T \boldsymbol{\psi}_j)}{\sum_{i=1}^{K} \exp(\boldsymbol{\phi}_w^T \boldsymbol{\psi}_i)},$$

- The node d ity

$$p_{\boldsymbol{\psi},\boldsymbol{\varphi}}(c|z=j) = \frac{\exp(\boldsymbol{\psi}_j^T \boldsymbol{\varphi}_c)}{\sum_{c' \in \mathcal{V}} \exp(\boldsymbol{\psi}_j^T \boldsymbol{\varphi}_{c'})}.$$

- Negative sa
-

$$\log \sigma(\boldsymbol{\varphi}_c^T \cdot \boldsymbol{\psi}_j) + \sum_{i=1}^{K} E_{v \sim P_n(v)}[\log \sigma(-\boldsymbol{\varphi}_v^T \cdot \boldsymbol{\psi}_j)],$$

Methodology

Node	w	Embedding of node i in p(z|w)	ϕ_i
Neighbor node	c	Embedding of node i in p(c|z)	φ_i
Community	z	Embedding of the j-th community	ψ_j

- It is intractable to maximize the log-likelihood of the observed edges for large graphs…
- To optimize the following evidence lower bound (ELBO)

$$\mathcal{L} = E_{z\sim q(z|c,w)}[\log p_{\psi,\varphi}(c|z)] - \underline{\mathrm{KL}(q(z|c,w)||p_{\phi,\psi}(z|w))}$$

- Parametrize the variational distribution $q(z|c, w)$ (the community membership of the edge (w, c)) with a neural network

$$q_{\phi,\psi}(z = j|w, c) = \frac{\exp((\phi_w \odot \phi_c)^T \psi_j)}{\sum_{i=1}^{K} \exp((\phi_w \odot \phi_c)^T \psi_i)}.$$

变分推断

`variational inference`就是用来计算`posterior distribution`的。

core idea

`variational inference`的核心思想包含两步：

- 假设分布 $q(z;\lambda)$ (这个分布是我们搞得定的，搞不定的就没意义了)
- 通过改变分布的参数 λ , 使 $q(z;\lambda)$ 靠近 $p(z|x)$

总结称一句话就是，**为真实的后验分布引入了一个参数话的模型**。即：用一个简单的分布 $q(z;\lambda)$ 拟合复杂的分布 $p(z|x)$ 。

这种策略将计算 $p(z|x)$ 的问题转化成优化问题了

$$\lambda^* = \arg\min_{\lambda} divergence(p(z|x), q(z;\lambda))$$

收敛后，就可以用 $q(z;\lambda)$ 来代替 $p(z|x)$ 了

Methodology

Node	w	Embedding of node i in p(z|w)	ϕ_i
Neighbor node	c	Embedding of node i in p(c|z)	φ_i
Community	z	Embedding of the j-th community	ψ_j

- It is intractable to maximize the log-likelihood of the observed edges…
- To optimize the following evidence lower bound (ELBO)

$$\mathcal{L} = E_{z\sim q(z|c,w)}[\log p_{\psi,\varphi}(c|z)] - \mathrm{KL}(q(z|c,w)||p_{\phi,\psi}(z|w))$$

- Parametrize the variational distribution $q(z|c, w)$ (the community membership of the edge (w, c)) with a neural network

$$q_{\phi,\psi}(z=j|w,c) = \frac{\exp((\phi_w \odot \phi_c)^T \psi_j)}{\sum_{i=1}^{K} \exp((\phi_w \odot \phi_c)^T \psi_i)}.$$

Methodology

Node	w	Embedding of node i in p(z|w)	ϕ_i
Neighbor node	c	Embedding of node i in p(c|z)	φ_i
Community	z	Embedding of the j-th community	ψ_j

- Approximate the community membership distribution of each node w, i.e., p(z|w)

$$p(z|w) = \sum_c p(z,c|w) = \sum_c p(z|w,c)p(c|w) \approx \frac{1}{|N(w)|} \sum_{c \in N(w)} q(z|w,c),$$

- To infer non-overlapping communities

$$\mathcal{F}(w) = \{\arg\max_k q(z=k|w,c)\}_{c \in N(w)}.$$

Complexity. Here we show the complexity of vGraph. Sampling an edge takes constant time, thus calculating Eq. (4) takes $\mathcal{O}(d(M+1))$ time, where M is the number of negative samples and d is the dimension of embeddings (the node embeddings and community embeddings have the same dimension). To calculate Eq. (6), it takes $\mathcal{O}(dK)$ time where K is the number of communities. Thus, an iteration with one sample takes $\mathcal{O}(\max(dM, dK))$ time. In practice the number of updates required is proportional to the number of edges $\mathcal{O}(|\mathcal{E}|)$, thus the overall time complexity of vGraph is $\mathcal{O}(|\mathcal{E}| d \max(M, K))$.

Methodology

Node	w	Embedding of node i in p(z|w)	ϕ_i
Neighbor node	c	Embedding of node i in p(c|z)	φ_i
Community	z	Embedding of the j-th community	ψ_j

- To optimize the lower bound w.r.t. the parameters in the variational distribution and the generative parameters...
  - If z is continuous, the reparameterization trick [15] can be used.
  - If z is discrete, we can use Gumbel-Softmax reparametrization

Methodology

- Inspired by existing spectral clustering studies [6], we augment our training objective with a smoothness regularization term that encourages the learned community distributions of linked nodes to be similar.
- $$\mathcal{L}_{reg} = \lambda \sum_{(w,c)\in\mathcal{E}} \alpha_{w,c} \cdot d(p(z|c), p(z|w))$$
- $$\alpha_{w,c} = \frac{|N(w) \cap N(c)|}{|N(w) \cup N(c)|},$$
- The entire loss function

$$\mathcal{L} = -E_{z\sim q_{\phi,\psi}(z|c,w)}[\log p_{\psi,\varphi}(c|z)] + \mathrm{KL}(q_{\phi,\psi}(z|c,w)||p_{\phi,\psi}(z|w)) + \mathcal{L}_{reg}$$

Experiments

- Datasets
  - For non-overlapping community detection and node classification, Citeseer, Cora, Cornell, Texas, Washington, and Wisconsin
  - For overlapping communtiy detection, Facebook, Youtube, Amazon, Dblp, Coauthor-CS. For Youtube, Amazon, and Dblp
- Evaluation Metric
  - For overlapping community detection, F1-Score and Jaccard Similarity
  - For non-overlapping community detection, Normalized Mutual Information (NMI) and Modularity
  - For node classification, Micro-F1 and Macro-F1

Experiments

- Comparative Methods
  - For overlapping community detection, we choose four competitive baselines: BigCLAM, CESNA, Circles, and SVI.
  - To evaluate node embedding and non-overlapping community detection, we compare our method with the five baselines: MF, DeepWalk, LINE, Node2vec, ComE.

Experiments

Table 2: Evaluation (in terms of F1-Score and Jaccard Similarity) on networks with overlapping ground-truth communities. NA means the task is not completed in 24 hours. In order to evaluate the effectiveness of smoothness regularization, we show the result of our model with (vGraph+) and without the regularization.

	F1-score						Jaccard					
Dataset	Bigclam	CESNA	Circles	SVI	vGraph	vGraph+	Bigclam	CESNA	Circles	SVI	vGraph	vGraph+
facebook0	**0.2948**	0.2806	0.2860	0.2810	0.2440	0.2606	**0.1846**	0.1725	0.1862	0.1760	0.1458	0.1594
facebook107	**0.3928**	0.3733	0.2467	0.2689	0.2817	0.3178	**0.2752**	0.2695	0.1547	0.1719	0.1827	0.2170
facebook1684	0.5041	**0.5121**	0.2894	0.3591	0.4232	0.4379	0.3801	**0.3871**	0.1871	0.2467	0.2917	0.3272
facebook1912	0.3493	0.3474	0.2617	0.2804	0.2579	**0.3750**	0.2412	0.2394	0.1672	0.2010	0.1855	**0.2796**
facebook3437	0.1986	0.2009	0.1009	0.1544	0.2087	**0.2267**	0.1148	0.1165	0.0545	0.0902	0.1201	**0.1328**
facebook348	0.4964	0.5375	0.5175	0.4607	**0.5539**	0.5314	0.3586	0.4001	0.3927	0.3360	**0.4099**	0.4050
facebook3980	0.3274	0.3574	0.3203	NA	**0.4450**	0.4150	0.2426	0.2645	0.2097	NA	**0.3376**	0.2933
facebook414	0.5886	0.6007	0.4843	0.3893	0.6471	**0.6693**	0.4713	0.4732	0.3418	0.2931	0.5184	**0.5587**
facebook686	0.3825	0.3900	0.5036	0.4639	0.4775	**0.5379**	0.2504	0.2534	0.3615	0.3394	0.3272	**0.3856**
facebook698	0.5423	0.5865	0.3515	0.4031	0.5396	**0.5950**	0.4192	0.4588	0.2255	0.3002	0.4356	**0.4771**
Youtube	0.4370	0.3840	0.3600	0.4140	0.5070	**0.5220**	0.2929	0.2416	0.2207	0.2867	0.3434	**0.3480**
Amazon	0.4640	0.4680	**0.5330**	0.4730	**0.5330**	0.5320	0.3505	0.3502	0.3671	0.3643	0.3689	**0.3693**
Dblp	0.2360	0.3590	NA	NA	0.3930	**0.3990**	0.1384	0.2226	NA	NA	0.2501	**0.2505**
Coauthor-CS	0.3830	0.4200	NA	0.4070	0.4980	**0.5020**	0.2409	0.2682	NA	0.2972	**0.3517**	0.3432

Experiments

Table 3: Evaluation (in terms of NMI and Modularity) on networks with non-overlapping ground-truth communities.

	NMI						Modularity					
Dataset	MF	deepwalk	LINE	node2vec	ComE	vGraph	MF	deepwalk	LINE	node2vec	ComE	vGraph
cornell	0.0632	0.0789	0.0697	0.0712	0.0732	**0.0803**	0.4220	0.4055	0.2372	0.4573	0.5748	**0.5792**
texas	0.0562	0.0684	**0.1289**	0.0655	0.0772	0.0809	0.2835	0.3443	0.1921	0.3926	**0.4856**	0.4636
washington	0.0599	0.0752	**0.0910**	0.0538	0.0504	0.0649	0.3679	0.1841	0.1655	0.4311	0.4862	**0.5169**
wisconsin	0.0530	0.0759	0.0680	0.0749	0.0689	**0.0852**	0.3892	0.3384	0.1651	0.5338	0.5500	**0.5706**
cora	0.2673	0.3387	0.2202	0.3157	**0.3660**	0.3445	0.6711	0.6398	0.4832	0.5392	0.7010	**0.7358**
citeseer	0.0552	0.1190	0.0340	0.1592	**0.2499**	0.1030	0.6963	0.6819	0.4014	0.4657	0.7324	**0.7711**

Table 4: Results of node classification on 6 datasets.

	Macro-F1						Micro-F1					
Datasets	MF	DeepWalk	LINE	Node2Vec	ComE	vGraph	MF	DeepWalk	LINE	Node2Vec	ComE	vGraph
Cornell	13.05	22.69	21.78	20.70	19.86	**29.76**	15.25	33.05	23.73	24.58	25.42	**37.29**
Texas	8.74	21.32	16.33	14.95	15.46	**26.00**	14.03	40.35	27.19	25.44	33.33	**47.37**
Washington	15.88	18.45	13.99	21.23	15.80	**30.36**	15.94	34.06	25.36	28.99	33.33	**34.78**
Wisconsin	14.77	23.44	19.06	18.47	14.63	**29.91**	18.75	**38.75**	28.12	25.00	32.50	35.00
Cora	11.29	13.21	11.86	10.52	12.88	**16.23**	12.79	22.32	14.59	27.74	**28.04**	24.35
Citeseer	14.59	16.17	15.99	16.68	12.88	**17.88**	15.79	19.01	16.80	**20.82**	19.42	20.42

Experiments

$$
\begin{aligned}
KL[q(z)||p(z|x)] &= E_{q(z)}[\log q(z)] - E_{q(z)}[\log p(z|x)] \\
&= E_{q(z)}[\log q(z)] - E_{q(z)}[\log p(z,x)] + E_{q(z)}[\log p(x)] \\
&= E_{q(z)}[\log q(z)] - E_{q(z)}[\log p(z,x)] + \log p(x)
\end{aligned}
$$

$$
\begin{aligned}
ELBO(q) &= \log p(x) - KL[q(z)||p(z|x)] \\
&= E_{q(z)}[\log p(z,x)] - E_{q(z)}[\log q(z)]
\end{aligned}
$$

$$
\begin{aligned}
ELBO(q) &= E_{q(z)}[\log p(z,x)] - E_{q(z)}[\log q(z)] \\
&= E_{q(z)}[\log(p(x|z)p(z))] - E_{q(z)}[\log q(z)] \\
&= E_{q(z)}[\log p(x|z)] + E_{q(z)}[\log p(z)] - E_{q(z)}[\log q(z)] \\
&= E_{q(z)}[\log p(x|z)] - KL[q(z)||p(z)]
\end{aligned}
$$

$$
\begin{aligned}
log(w|\alpha,\eta) &= log \int\int\sum_z p(\theta,\beta,z,w|\alpha,\eta)d\theta d\beta && (3) \\
&= log \int\int\sum_z \frac{p(\theta,\beta,z,w|\alpha,\eta)q(\beta,z,\theta|\lambda,\phi,\gamma)}{q(\beta,z,\theta|\lambda,\phi,\gamma)} d\theta d\beta && (4) \\
&= log\ E_q \frac{p(\theta,\beta,z,w|}{q(\beta,z,\theta|\lambda,} \\
&\geq E_q\ log \frac{p(\theta,\beta,z,w|\alpha,\eta)}{q(\beta,z,\theta|\lambda,\phi,\gamma)} && (6) \\
&= E_q\ logp(\theta,\beta,z,w|\alpha,\eta) - E_q\ logq(\beta,z,\theta|\lambda,\phi,\gamma) && (7)
\end{aligned}
$$

$$\mathcal{L} = E_{z\sim q(z|c,w)}[\log p_{\psi,\varphi}(c|z)] - \mathrm{KL}(q(z|c,w)||p_{\phi,\psi}(z|w)$$