GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks

이상용 / 2020-04-03

Computational Data Science LAB

GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks

Computational Data Science LAB	
목차	1. Introduction 2. GraphLIME 3. Experiments
논의사항 및 결정사항	
관련문서	Huang, Q., Yamada, M., Tian, Y., Singh, D., Yin, D., & Chang, Y. (2020). GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks. *arXiv preprint arXiv:2001.06216.*

CONTENTS

1. Introduction
2. GraphLIME
3. Experiments

01 | Introduction

- Graph neural networks (GNN)은 그래프 구조화된 데이터에 대해 우수한 성능과 일반화 능력을 보임
- GNN은 특정한 노드에 대한 node representation과 그래프의 structural information을 고려하여 학습하는 특성으로 인해, GNN 모델에 대한 해석을 하는 것은 어려움
- 최근에 제안된 GNNexplainer는 쓸모 있는 features를 찾기는 하지만, 해당 노드를 잘 설명하는 graph structure를 찾는 것에 더 초점을 둠
- 본 논문은 그래프에 대한 local interpretable model explanation을 제공하는 GraphLIME 을 제안

02 | GraphLIME

Formulation of GraphLIME Explainer

- ✓ 그래프에서 v_i 는 각각의 노드, $\boldsymbol{x}_i$는 i번째 노드의 feature vector
- ✓ $\boldsymbol{X}_n = \{\boldsymbol{x}_i | v_i \in \mathcal{S}_N\}$: 샘플링된 N-hop neighbor information matrix
- ✓ 해석하고자 하는 기 훈련된 모델 $f: \mathbb{R}^d \rightarrow \mathbb{R}$ 이며, $y_i = f(\boldsymbol{x}_i)$는 instance $\boldsymbol{x}_i$가 특정한 class에 속할 확률
- ✓ Explanation model $g \in G$이며, G는 $interpretable\ model\ class$(linear regression, decision tree, …)
- ✓ 본 논문에서의 G는 '해석가능한 커널 기반의 비선형 변수선택 알고리즘' 인 **Hilbert-Schmidt Independence Criterion Lasso (HSIC Lasso)**를 사용
- ✓ 목적

$$\xi(v) = argmin_{g \in G} g(f, \boldsymbol{X}_n)$$

$\xi(v)$는 **노드 v를 설명할 수 있는 feature set**

02 | GraphLIME

Nonlinear Explanation Model: HSIC LASSO

- HSIC Lasso는 "**supervised**" feature-wise **kernelized nonlinear feature selection method**

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \frac{1}{2}\|\bar{\boldsymbol{L}} - \sum_{k=1}^{d}\beta_k \bar{\boldsymbol{K}}^{(k)}\|_F^2 + \rho\|\boldsymbol{\beta}\|_1, \quad \text{s.t.} \quad \beta_1,\ldots,\beta_d \geq 0,$$

- $||\cdot||_F$: Frobenius norm (Euclidean norm)
- $\bar{\boldsymbol{L}} = \boldsymbol{HLH}/\left|\left|\boldsymbol{HLH}\right|\right|_F$: normalized centered Gram matrix, $\boldsymbol{L}_{ij} = L(y_i, y_j)$ ($G = V^T V$)
- $\boldsymbol{H} = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$: centering matrix (1. Symmetic : $A = A^T$ 2. Idempotent : $A^2 = A$)
- $\bar{\boldsymbol{K}}^{(k)} = \boldsymbol{HK}^{(k)}\boldsymbol{H}/\left|\left|\boldsymbol{HK}^{(k)}\boldsymbol{H}\right|\right|_F$: **k-th feature의** normalized centered Gram matrix, $[\boldsymbol{K}^{(k)}]_{ij} = K(x_i^{(k)}, x_j^{(k)})$

$$L(y_i, y_j) = \exp\left(-\frac{\|y_i - y_j\|_2^2}{2\sigma_y^2}\right), \quad K(\boldsymbol{x}_i^{(k)}, \boldsymbol{x}_j^{(k)}) = \exp\left(-\frac{(\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_j^{(k)})^2}{2\sigma_x^2}\right)$$

i, j 쌍에 대해 모든 k번째 변수들의 유사도 — i, j 쌍에 대해 f로 예측한 결과들의 유사도

02 | GraphLIME

Interpretation of HSIC Lasso

- HSIC Lasso는 *minimum redundancy maximum relevancy* (mRMR)의 특성을 가짐
  - ✓ X들과 중복성이 적으면서, Y를 잘 예측하는 변수를 선택하는 알고리즘

$$\frac{1}{2}\|\bar{\boldsymbol{L}} - \sum_{k=1}^{d} \beta_k \bar{\boldsymbol{K}}^{(k)}\|_F^2 = \frac{1}{2}\sum_{k,m=1}^{d} \beta_k \beta_m \mathrm{NHSIC}(\boldsymbol{f}_k, \boldsymbol{f}_m) - \sum_{k=1}^{d} \beta_k \mathrm{NHSIC}(\boldsymbol{f}_k, \boldsymbol{y}) + \frac{1}{2}$$

변수끼리의 관련도 / 변수와 종속변수의 관련도

- $\boldsymbol{f}_k$는 k번째 feature vector이며, HSIC는 값이 클수록 두 변수가 강한 의존성
- $\mathrm{NHSIC}(\boldsymbol{f}_k, \boldsymbol{y}) = \mathrm{tr}(\bar{\boldsymbol{K}}^{(k)}\bar{\boldsymbol{L}})$, 만약 $\boldsymbol{f}_k$와 $\boldsymbol{y}$가 강한 의존성을 가지면 → 값이 커지고 대응되는 β_k또한 커짐 (minimize)
- $\mathrm{NHSIC}(\boldsymbol{y}, \boldsymbol{y}) = 1$
- $\mathrm{NHSIC}(\boldsymbol{f}_k, \boldsymbol{f}_m)$, 만약 $\boldsymbol{f}_k$와$\boldsymbol{f}_m$이 강한 의존성을 가지면 → 값이 커지고 β_k, β_m둘 중 하나는 zero (minimize)

- 이러한 특성으로 중요변수 선택을 할 수 있음

02 | GraphLIME

Interpretation of HSIC Lasso

- HSIC Lasso pseudocode

Algorithm 1 Locally nonlinear Explanation: GraphLIME

Input: GNN classifier f, Number of explanation features K
Input: the graph $\mathcal{G}$, the node $\boldsymbol{x}$ being explained
Output: K explanation features

f

1: $\boldsymbol{X}_n = N\_hop\_neighbor\_sample(x)$
2: $\boldsymbol{Z} \leftarrow \{\}$
3: **for all** $x_i \in \boldsymbol{X}_n$ **do**
4: $\quad y_i = f(\boldsymbol{x}_i)$
5: $\quad \boldsymbol{Z} \leftarrow \boldsymbol{Z} \cup (\boldsymbol{x}_i, y_i)$
6: **end for**

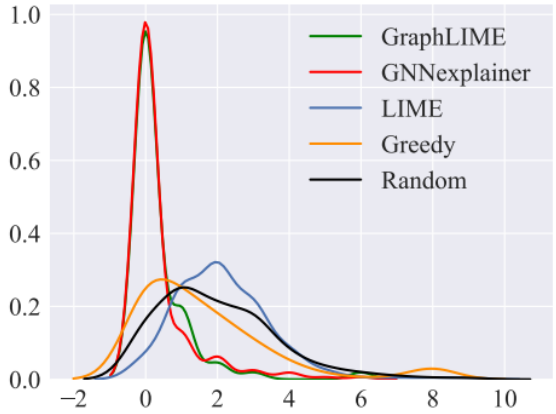
g

7: $\boldsymbol{\beta} \leftarrow$ HSIC Lasso$(\boldsymbol{Z})$ $\triangleright$ with x_i as features, y_i as label
8: Select top-K features as explanations based on $\boldsymbol{\beta}$

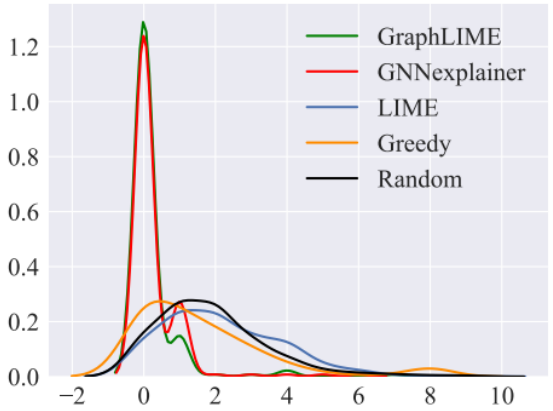
03 | Experiments

- Does the explanation framework filter useless features?

Datasets	Cora	Pubmed
# of Classes	7	3
# of Nodes	2708	19717
# of Features	1433	500
# of Links	5429	44338



(a) Distribution of noisy features on Cora



(b) Distribution of noisy features on Pubmed

- Do I trust this prediction?

Table 2: Average F1-Score (%) of trustworthiness for different explainers on Cora.

Method	K=10	K=15	K=20	K=25
Random	40.4±4.3	37.7±4.6	36.6±4.3	36.0±4.2
Greedy	71.7±2.9	71.6±3.2	71.5±3.3	71.5±3.2
GNNexplainer	77.6±3.1	71.5±2.9	77.1±1.4	76.5±2.5
LIME	89.3±1.8	89.7±1.7	89.8±1.1	89.8±1.1
GraphLIME	**95.3±0.5**	**95.6±0.7**	**95.4±0.7**	**95.4±0.6**

Table 3: Average F1-Score (%) of trustworthiness for different explainers on Pubmed.

Method	K=10	K=15	K=20	K=25
Random	21.9±2.6	16.9±2.1	15.3±1.7	14.6±1.6
Greedy	63.5±7.5	62.8±7.3	62.4±7.3	62.6±7.2
GNNexplainer	79.5±2.7	77.6±2.2	79.3±3.7	75.1±1.6
LIME	83.6±0.8	84.0±0.9	84.5±0.9	84.6±0.9
GraphLIME	**92.5±0.8**	**92.3±0.8**	**91.6±0.9**	**91.5±0.8**

Q&A

감사합니다.