Pick and Choose: A GNN-based Imbalanced Learning Approach for Fraud Detection

Yang Liu¹; Xiang Ao¹*; Zidi Qin¹; Jianfeng Chi²; Jinghua Feng²; Hao Yang²; Qing He¹

柳 阳¹; 敖 翔¹*; 秦紫笛¹; 池剑锋²; 冯景华²; 杨 浩²; 何 清¹

¹ Institute of Computing Technology, Chinese Academy of Sciences
² Alibaba Group

* denotes corresponding author.
➢ Background and Motivation

➢ Method – PC-GNN

➢ Experiment

➢ Conclusion and Future Work
➢ Background and Motivation

➢ Method – PC-GNN

➢ Experiment

➢ Conclusion and Future Work
Fraud

- Opinion fraud (fake/spam review)
- Financial fraud (fraudster/defaulter)

Online Review Sites

Friend or Paid Reviewer

Rival or Enemy

Images from wikihow, https://www.wikihow.com/Spot-a-Fake-Review-on-Amazon
Fraud

- Opinion fraud (fake/spam review)
- Financial fraud (fraudster/defaulter)

Images from https://hrdailyadvisor.blr.com/2020/04/02/qa-identity-theft-benefits-more-relevant-than-ever/
Fraud Detection

- A set of processes and analyses that allow businesses to identify and prevent unauthorized financial activity.

Graph-based Fraud Detection

- Relational data could be modeled as a graph
- Examples: product reviews

- **R-U-R** Reviews posted by the same **User**
- **R-S-R** Reviews under the same product with the same **Star** rating
- **R-T-R** Reviews under the same product in the same **Month**
Background

Graph-based Fraud Detection

- Relational data could be modeled as a graph
- Examples: financial scenario

- **U-T-U** User Trades to another
- **U-D-U** Users log in the same Device
- **U-F-U** User transfer Fund to another
- **U-S-U** Users have Social relationships
Background

Class imbalance problem

- Only a small fraction of samples belong to the fraud class
- The trained model is easily biased to the majority class

Over-sample the minority class
- e.g. SMOTE, ADASYN, etc.

Under-sample the majority class
- e.g. TU, TRUST, etc.
Motivation

Imbalanced Learning on Graphs

Challenges:

➢ Camouflage:
  • redundant links between fraudsters and benign users
  • lack necessary links among fraudsters

➢ Message Aggregation:
  • Most neighbors belong to the majority class
  • The prediction would be biased
➢ Background and Motivation

➢ Method – PC-GNN

➢ Experiment

➢ Conclusion and Future Work
Method

➢ Pick nodes from the whole graph

Label Frequency

LF(●) = 3
LF(○) = 6

Sampling Probability

\[ P(\bullet) = \frac{1}{3} \]
\[ P(\circ) = \frac{1}{6} \]

\[ P(v) \propto \frac{||\hat{A}(\cdot, v)||^2}{\text{LF}(\text{C}(v))} \]
Choose neighbors for the minority class

- Over-sample neighbors of the minority class
  \[ N_r(t)(v) = \{ u \in V | C(u) = C(v) \text{ and } D_r(t)(v, u) < \rho_+ \} \]

- Under-sample neighbors of both classes
  \[ \overline{N_r(t)}(v) = \{ u \in V | A_r(v, u) > 0 \text{ and } D_r(t)(v, u) < \rho_- \} \]

- For minority targets:
  \[ N_r(t)(v) = \overline{N_r(t)}(v) \cup \overline{N_r(t)}(v) \]

- For majority targets:
  \[ N_r(t)(v) = \overline{N_r(t)}(v) \]

Method

\[ D_r(t)(v, u) = \| D_r(t)\left(h_{u,r}\right) - D_r(t)\left(h_{v,r}\right)\|_1 \]

\[ D_r(t)\left(h_{u,r}\right) = \sigma \left(U_r(t)h_{v,r}\right) \]
PC-GNN: Pick and Choose Graph Neural Network

1. Pick
\[ P(v) \propto \frac{||\hat{A}(., v)||^2}{\text{LF}(C(v))} \]

2. Choose
\[ \mathcal{N}^{(t)}_r(v) = \left\{ u \in \mathcal{V} | C(u) = C(v) \text{ and } \mathcal{D}^{(t)}_r(v, u) < \rho_+ \right\} \]
\[ \mathcal{N}^{(t)}_c(v) = \left\{ u \in \mathcal{V} | A_r(v, u) > 0 \text{ and } \mathcal{D}^{(t)}_r(v, u) < \rho_- \right\} \]

3. Aggregate
\[ h_v^{(t-1)} = \text{ReLU} \left( W^{(t)}_1 (h_v^{(t-1)} \oplus h^{(t)}_v \oplus \cdots \oplus h^{(t)}_v) \right) \]
\[ h_v^{(t)} = \text{ReLU} \left( W^{(t)}_2 (h_v^{(t-1)} \oplus \text{AGG}^{(t)}_r (h_v^{(t-1)}, u \in \mathcal{N}^{(t)}_r(v))) \right) \]
Method

PC-GNN: Training

- Training the distance function

\[ D_r^{(t)}(v, u) = \left\| D_r^{(t)}(h_{v,r}^{(t)}) - D_r^{(t)}(h_{u,r}^{(t)}) \right\|_1 \]

\[ p_{v,r}^{(t)} = D_r^{(t)}(h_{v,r}^{(t)}) \]

\[ L_{\text{dist}} = - \sum_{l=1}^L \sum_{r=1}^R \sum_{v \in V} \left[ y_v \log p_{v,r}^{(t)} + (1 - y_v) \log (1 - p_{v,r}^{(t)}) \right] \]

- Training GNN framework

\[ h_{v,r}^{(t)} = \text{ReLU} \left( W_r^{(t)} \left( h_{v,r}^{(t-1)} \oplus \text{AGG}_r^{(t)} \{ h_{u,r}^{(t-1)}, u \in N_r^{(t)}(v) \} \right) \right) \]

\[ h_v^{(t)} = \text{ReLU} \left( W^{(t)} \left( h_v^{(t-1)} \oplus h_{v,1}^{(t)} \oplus \cdots \oplus h_{v,R}^{(t)} \right) \right) \]

\[ p_v = \text{MLP} \left( h_v^{(L)} \right) \]

\[ L_{\text{gnn}} = - \sum_{v \in V} \left[ y_v \log p_v + (1 - y_v) \log (1 - p_v) \right] \]

- Overall loss function

\[ L = L_{\text{gnn}} + \alpha L_{\text{dist}} \]
Background and Motivation

Method – PC-GNN

Experiment

- RQ1: Does PC-GNN outperform the state-of-the-art methods for graph-based anomaly detection?
- RQ2: How do the key components benefit the prediction?
- RQ3: What is the performance with respect to different training parameters?
- RQ4: If the proposed modules are applied to other GNN models, will it bring performance improvement?

Conclusion and Future Work
Public benchmark - Opinion fraud detection
- **YelpChi**: hotel and restaurant reviews on Yelp
- **Amazon**: product reviews under the Musical Instrument category

Real-world dataset - Financial fraud detection
- Provided by Alibaba Group
- **M7** collects users from 2018/07/01 to 2018/07/31
- **M9** collects users from 2018/09/01 to 2018/09/30

Train/Valid/Test:
- 40%/20%/40%
Experiment

➢ Compared methods

• GCN, GAT: traditional GNNs
• DR-GCN: dual-regularized GCN for imbalanced classification
• GraphSAGE, GraphSAINT: sampling-based GNNs
• GraphConsis, CARE-GNN: SOTA for graph-based fraud detection
• PC-GNN\_p, PC-GNN\_c: Model with Pick and Choose removed for ablation study

➢ Metrics

• F1-macro: macro average of F1-score of each class
• AUC: Area Under the ROC Curve
• GMean: Geometric Mean of True Positive Rate (TPR) and True Negative Rate (TNR)
RQ1: Does PC-GNN outperform the state-of-the-art methods for graph-based anomaly detection?

- Compared with state-of-the-art CARE-GNN\textsuperscript{[CIKM'20]}
  - AUC improvement 3.6\%~5.2\%
  - GMean improvement 0.6\%~3.7\%

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>Metric</th>
<th>YelpChi</th>
<th>Amazon</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>F1-macro</td>
<td>AUC</td>
<td>GMean</td>
</tr>
<tr>
<td>GCN</td>
<td></td>
<td>0.5620±0.0067</td>
<td>0.5983±0.0049</td>
<td>0.4365±0.0262</td>
</tr>
<tr>
<td>GAT</td>
<td></td>
<td>0.4879±0.0230</td>
<td>0.5715±0.0029</td>
<td>0.1659±0.0789</td>
</tr>
<tr>
<td>DR-GCN</td>
<td></td>
<td>0.5523±0.0231</td>
<td>0.5921±0.0195</td>
<td>0.4038±0.0742</td>
</tr>
<tr>
<td>Baselines</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GraphSAGE</td>
<td></td>
<td>0.4405±0.1066</td>
<td>0.5439±0.0025</td>
<td>0.2589±0.1864</td>
</tr>
<tr>
<td>GraphSAINT</td>
<td></td>
<td>0.5960±0.0038</td>
<td>0.6999±0.0029</td>
<td>0.5908±0.0298</td>
</tr>
<tr>
<td>GraphCons</td>
<td></td>
<td>0.5870±0.0200</td>
<td>0.6983±0.0302</td>
<td>0.5857±0.0385</td>
</tr>
<tr>
<td>CARE-GNN</td>
<td></td>
<td>0.6332±0.0094</td>
<td>0.7619±0.0292</td>
<td>0.6791±0.0359</td>
</tr>
<tr>
<td>Ablation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC-GNN_L</td>
<td></td>
<td>0.5136±0.0147</td>
<td>0.7844±0.0013</td>
<td>0.2336±0.0356</td>
</tr>
<tr>
<td>PC-GNN_C</td>
<td></td>
<td>0.6634±0.0558</td>
<td>0.7847±0.0021</td>
<td>0.6258±0.0378</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td>0.6300±0.0230</td>
<td>0.7987±0.0014</td>
<td>0.7160±0.0130</td>
</tr>
</tbody>
</table>
Compared with state-of-the-art CARE-GNN\textsuperscript{[CIKM'20]}

- AUC improvement 2.6%~3.5%
- GMean improvement 28.4%~31.9%

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>F1-macro</th>
<th>AUC</th>
<th>GMean</th>
<th>F1-macro</th>
<th>AUC</th>
<th>GMean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baselines</td>
<td>GCN</td>
<td>0.3108±0.0256</td>
<td>0.6107±0.0041</td>
<td>0.5456±0.0159</td>
<td>0.3016±0.0574</td>
<td>0.5790±0.0040</td>
<td>0.5241±0.0422</td>
</tr>
<tr>
<td></td>
<td>GAT</td>
<td>0.2746±0.0168</td>
<td>0.6083±0.0149</td>
<td>0.5016±0.0168</td>
<td>0.2698±0.0069</td>
<td>0.5647±0.0069</td>
<td>0.4354±0.0346</td>
</tr>
<tr>
<td></td>
<td>DR-GCN</td>
<td>0.3070±0.0232</td>
<td>0.7195±0.0208</td>
<td>0.5647±0.0403</td>
<td>0.5055±0.0012</td>
<td>0.6637±0.0236</td>
<td>0.3106±0.0417</td>
</tr>
<tr>
<td>Baselines</td>
<td>GraphSAGE</td>
<td>0.5186±0.0030</td>
<td>0.6790±0.0029</td>
<td>0.1605±0.0132</td>
<td>0.5020±0.0026</td>
<td>0.6342±0.0040</td>
<td>0.0525±0.0362</td>
</tr>
<tr>
<td></td>
<td>GraphSAINT</td>
<td>0.5149±0.0036</td>
<td>0.6915±0.0068</td>
<td>0.2547±0.0459</td>
<td>0.5018±0.0019</td>
<td>0.6587±0.0049</td>
<td>0.1864±0.0354</td>
</tr>
<tr>
<td></td>
<td>GraphConv</td>
<td>0.5236±0.0087</td>
<td>0.6826±0.0049</td>
<td>0.2734±0.0548</td>
<td>0.5124±0.0043</td>
<td>0.6743±0.0076</td>
<td>0.2302±0.0467</td>
</tr>
<tr>
<td></td>
<td>CARE-GNN</td>
<td>0.5578±0.0015</td>
<td>0.7836±0.0020</td>
<td>0.3451±0.0098</td>
<td>0.5361±0.0035</td>
<td>0.7579±0.0060</td>
<td>0.2908±0.0294</td>
</tr>
<tr>
<td>Ablation</td>
<td>PC-GNN\textsubscript{p}</td>
<td>0.4979±0.0000</td>
<td>0.7434±0.0042</td>
<td>0.0000±0.0000</td>
<td>0.4982±0.0000</td>
<td>0.6575±0.0069</td>
<td>0.0000±0.0000</td>
</tr>
<tr>
<td></td>
<td>PC-GNN\textsubscript{c}</td>
<td>0.5753±0.0017</td>
<td>0.8132±0.0031</td>
<td>0.4362±0.0254</td>
<td>0.5353±0.0028</td>
<td>0.7668±0.0038</td>
<td>0.3138±0.0387</td>
</tr>
<tr>
<td>Ours</td>
<td>PC-GNN</td>
<td>0.5749±0.0044</td>
<td>0.8192±0.0032</td>
<td>0.6645±0.0422</td>
<td>0.5370±0.0021</td>
<td>0.7847±0.0019</td>
<td>0.5740±0.0391</td>
</tr>
</tbody>
</table>

RQ2: How do the key components benefit the prediction?
➢ RQ3: What is the performance with respect to different training parameters?

➢ RQ4: If the proposed modules are applied to other GNN models, will it bring performance improvement?

<table>
<thead>
<tr>
<th>Dataset</th>
<th>F1-macro</th>
<th>AUC</th>
<th>GMean</th>
<th>F1-macro</th>
<th>AUC</th>
<th>GMean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN</td>
<td>0.4645±0.0033</td>
<td>0.5983±0.0049</td>
<td>0.0483±0.0037</td>
<td>0.5297±0.0539</td>
<td>0.8369±0.0125</td>
<td>0.1904±0.1716</td>
</tr>
<tr>
<td>GCN(T)</td>
<td>0.5620±0.0067</td>
<td>0.5983±0.0049</td>
<td>0.4365±0.0262</td>
<td>0.6486±0.0694</td>
<td>0.8369±0.0125</td>
<td>0.5718±0.1951</td>
</tr>
<tr>
<td>GCN(P)</td>
<td>0.5540±0.0275</td>
<td>0.6420±0.0643</td>
<td>0.5114±0.0107</td>
<td>0.7138±0.0086</td>
<td>0.8773±0.0027</td>
<td>0.8232±0.0374</td>
</tr>
<tr>
<td>GraphSAGE</td>
<td>0.4603±0.0000</td>
<td>0.5439±0.0025</td>
<td>0.0000±0.0000</td>
<td>0.4751±0.0000</td>
<td>0.7589±0.0046</td>
<td>0.0000±0.0000</td>
</tr>
<tr>
<td>GraphSAGE(T)</td>
<td>0.4405±0.1066</td>
<td>0.5439±0.0025</td>
<td>0.2589±0.1864</td>
<td>0.6416±0.0079</td>
<td>0.7589±0.0046</td>
<td>0.5949±0.0349</td>
</tr>
<tr>
<td>GraphSAGE(P)</td>
<td>0.6178±0.0286</td>
<td>0.7765±0.0025</td>
<td>0.6952±0.0176</td>
<td>0.5831±0.0227</td>
<td>0.7627±0.0097</td>
<td>0.6993±0.0081</td>
</tr>
</tbody>
</table>
➢ Background and Motivation

➢ Method – PC-GNN

➢ Experiment

➢ Conclusion and Future Work
Conclusion and Future work

➢ Conclusion
  • We propose a GNN-based imbalanced learning method named PC-GNN to solve the class imbalance problem in graph-based fraud detection
  • Experiments on two benchmark opinion fraud datasets and two real-world financial fraud datasets demonstrate the effectiveness of the proposed framework.

➢ Future Work
  • Graph structure learning for imbalanced graph data
Thanks for listening!

If you have any question, feel free to contact us at
liuyang17z@ict.ac.cn
aoxiang@ict.ac.cn

Paper and slides are available at
https://ponderly.github.io/