ICDM 2021
IEEE International Conference on Data Mining
7 – 10 DECEMBER 2021
AUCKLAND NEW ZEALAND

Accurate Graph-Based PU Learning without Class Prior

Jaemin Yoo[1,*], **Junghun Kim**[1,*], **Hoyoung Yoon**[1,*], **Geonsoo Kim**[2], **Changwon Jang**[2], and **U Kang**[1]

* Equal contribution

[1] Seoul National University

[2] NCSOFT

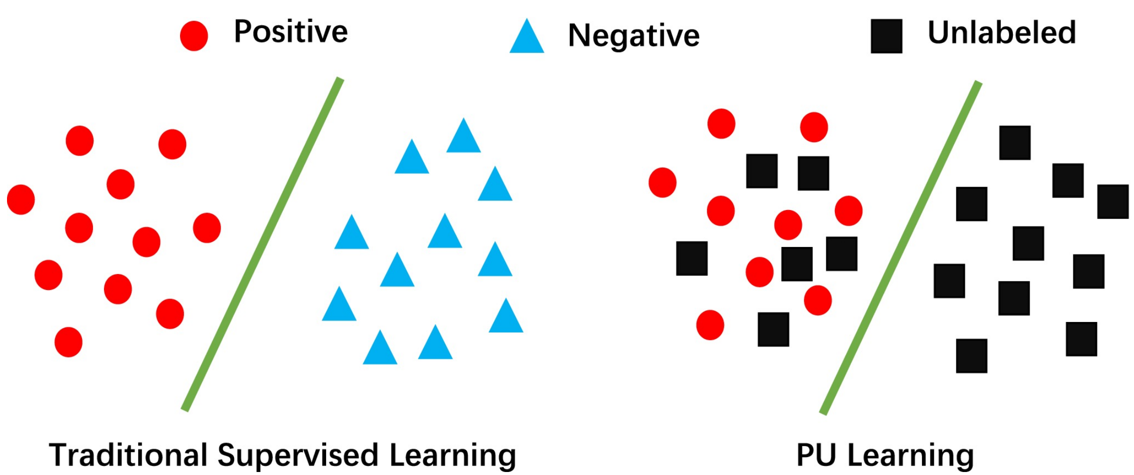
ICDM 2021

Outline

- **<u>Introduction</u>**
- Proposed Method
- Experiments
- Conclusion

PU Learning (1)

- **Positive-unlabeled (PU) learning**
  - Binary classification with limited observations
  - Negative examples are **unseen** during training
    - There are only positive and unlabeled examples

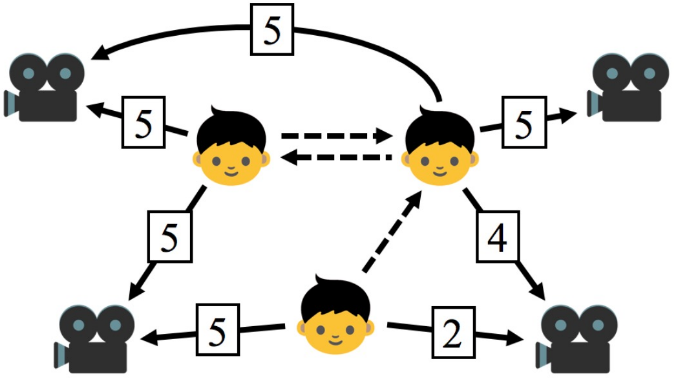


PU Learning (2)

- PU learning is **common** in the real world
  - Detecting review manipulation
  - Detecting bot accounts in a social network
- Consider detecting review manipulation:
  - We detected 100 reviews among 1000 ones
  - Are the remaining 900 reviews all normal?
  - They should be treated **unlabeled**, not **negative**

Graphs

- Many datasets are represented as **graphs**
  - Graphs allow us to understand the relationships
- PU learning is common also in graph data
  - Social networks, streaming services, …

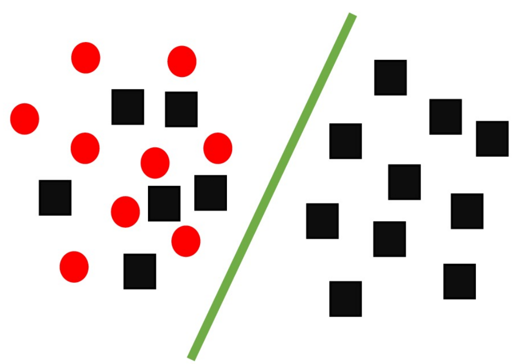


Problem Definition

- **Graph-based PU learning**
  - **Given**
    - Undirected graph $G = (\mathcal{V}, \mathcal{E})$
      - $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges, resp.
    - Feature matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$
      - d is the number of features
    - Set $\mathcal{P} \subset \mathcal{V}$ of positive nodes
      - The remaining nodes $\mathcal{U} = \mathcal{V} \setminus \mathcal{P}$ are unlabeled
  - **Classify**
    - Each node $u \in \mathcal{U}$ into positive or negative

Class Prior (1)

- Existing models require the **class prior** π_{p}
  - The ratio of positive nodes among unlabeled ones
- π_{p} provides **rich information** to PU learning
  - Assume that $|\mathcal{U}| = 16$ and $\pi_{\mathrm{p}} = 0.38$
    - Then, we know that exactly 6 nodes in $\mathcal{U}$ are positive



$\pi_{\mathrm{p}}|\mathcal{U}| = 0.38 \times 16 \approx 6$

We can draw a boundary after selecting the 6 nodes closest to $\mathcal{P}$

Class Prior (2)

- However, π_{p} is **not available** in most cases
  - π_{p} requires additional domain knowledge:
    - *The ratio of manipulated reviews*
    - *The ratio of bot accounts in a social network*

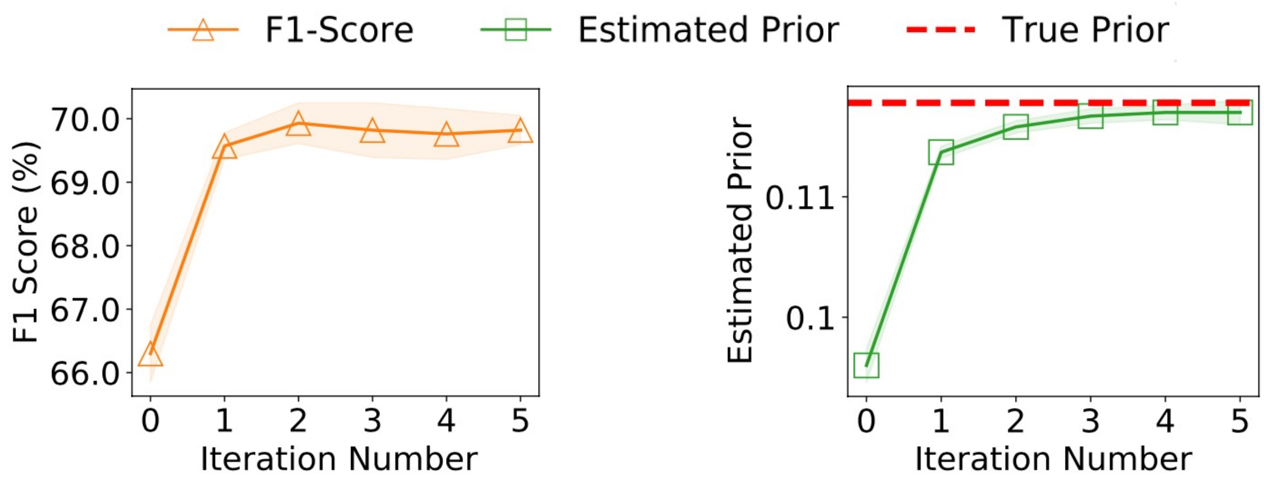
Q1. How can we solve PU learning without π_{p}?
Q2. How can we estimate π_{p} only from given data?

Outline

- Introduction
- **<u>Proposed Method</u>**
- Experiments
- Conclusion

Overview

- We propose **GRAB** for accurate PU learning
  - It estimates the unknown prior π_{p} from data
  - **Idea 1.** Model the graph as a Markov network
  - **Idea 2.** Update an estimate $\hat{\pi}_{\mathrm{p}}$ through iterations



Objective Function

- Our goal is to minimize the following:

$$\mathcal{L}(\theta; \mathbf{X}, \mathbf{y}, \mathcal{P}, \mathcal{U}) = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} (-\log \hat{y}_i(+1)) + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}|\mathbf{X}, \mathbf{y})} \left[\frac{1}{|\mathcal{U}|} \sum_{j \in \mathcal{U}} (-\log \hat{y}_j(z_j)) \right],$$

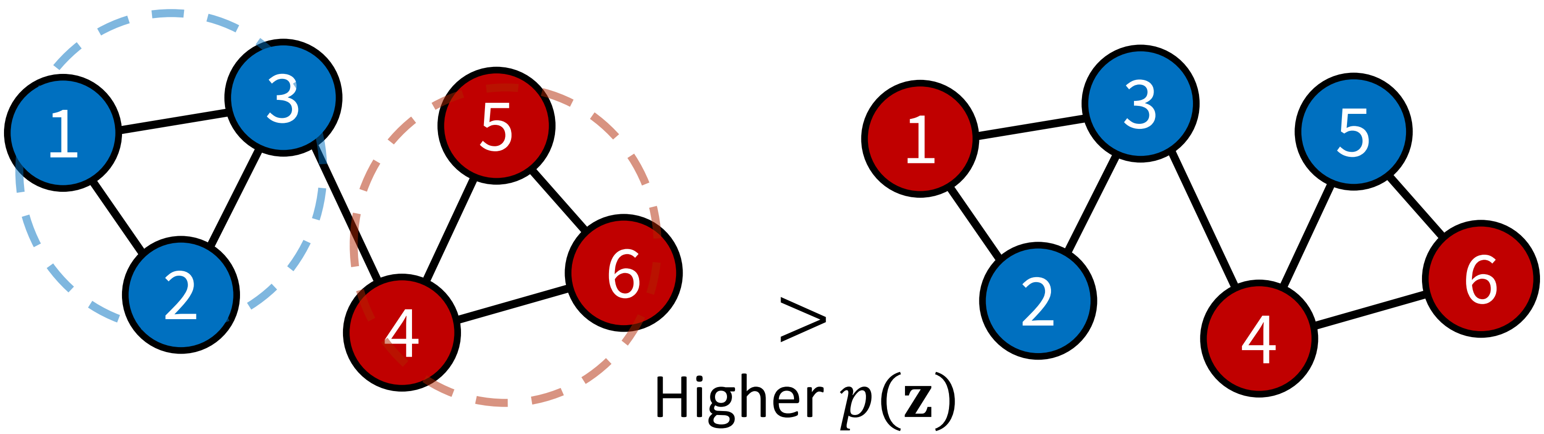
Positive part

Unlabeled part

  - $\hat{y}_i$ is the prediction of our classifier for each node i
  - We model each unlabeled node i as a variable Z_i
  - The **challenge** is to model $p(\mathbf{z}|\mathbf{X}, \mathbf{y})$ without prior π_p

Markov Network

- We model the graph G as a **Markov network**
  - Adjacent nodes are likely to have the same state
  - Large $p(\mathbf{z})$ if $\mathbf{z}$ follows the graph structure well



Iterative Optimization

- Then, we minimize $\mathcal{L}(\theta)$ through iterations:
  - **Initialization**
    - $f \leftarrow$ A graph convolutional network (GCN) classifier
  - **Iterative updates**
    - **Marginalization step**
      - $\hat{\mathbf{y}} \leftarrow$ Make a prediction from the current f
      - $\mathbf{B} \leftarrow$ Run graphical inference using $\hat{\mathbf{y}}$ as prior
    - **Update step**
      - $\mathcal{L} \leftarrow$ Make a new objective function from $\mathbf{B}$
      - $f \leftarrow$ Train a new classifier minimizing $\mathcal{L}(\cdot)$

Marginalization Step

- To approximate $p(\mathbf{z}|\mathbf{X}, \mathbf{y})$ by marginalization
  1. Make a prediction $\hat{\mathbf{y}}$ from the current classifier f
  2. Run **graphical inference** treating $\hat{\mathbf{y}}$ as priors
     - Specifically, we run loopy belief propagation (LBP)
     - It iteratively propagates the priors through the graph
  3. Get an approximate marginal b_i for each node i
  4. Return a **belief** matrix $\mathbf{B} \in \mathbb{R}^{|\mathcal{V}|\times 2}$ as a result

Update Step

- We train a new classifier f based on $\mathbf{B}$
  1. Make a new objective function $\hat{\mathcal{L}}$
  2. Train f to minimize the objective function
- The new objective function $\hat{\mathcal{L}}$ is defined as

$$\tilde{\mathcal{L}}(\theta; \mathbf{X}, \mathbf{y}, \mathbf{B}, \mathcal{P}, \mathcal{U}) =$$

$$\frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} l(\bar{y}_i, \hat{y}_i) + \frac{1}{|\mathcal{U}|}\sum_{j\in\mathcal{U}} l(b_j, \hat{y}_j),$$

l: Loss function
$\bar{y}_i$: One-hot label

  - b_j is used as an answer for each node $j \in \mathcal{U}$

Outline

- Introduction
- Previous Works
- **Experiments**
- Conclusion

Datasets

- We use five datasets from different domains
  - Four are public datasets used in previous works
  - MMORPG is a private dataset collected in this work
    - Classify each character into a **normal** user or a **bot**

Name	Nodes	Edges	Features	Pos.	Neg.
Cora[1]	2,708	5,278	1,433	818	1,890
Citeseer[1]	3,327	4,552	3,703	701	2,626
Cora-ML[2]	2,995	8,158	2,879	857	2,138
WikiCS[3]	11,701	215,603	300	2,679	9,022
MMORPG[4]	6,312	68,012	136	298	401

Experimental Setup

- **Evaluation metrics**
  - **F1 score:** The average of precision and recall
  - **Accuracy:** The ratio of correct predictions
- **Competitors**
  - Representation learning [KDD'16, IJCAI'18]
  - General PU learning [ICML'15, NIPS'17]
  - Graph-based PU learning [ICMEW'17, CIKM'19]

Classification Accuracy

- **Q1.** How accurate is GRAB in PU learning?
  - GRAB outperforms all other baselines
  - The prior π_p is given **only to the competitors**

Method	Cora		Citeseer		Cora-ML	
	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)
GCN+CE	23.0±1.9	84.3±0.2	25.8±2.2	89.6±0.2	26.8±3.0	85.8±0.4
GCN+PULP	40.2±1.7	86.1±0.2	37.1±3.1	90.3±0.4	38.3±1.3	86.4±0.2
GCN+URE	50.9±0.8	88.0±0.1	42.6±1.7	90.9±0.2	54.6±1.8	89.4±0.3
GCN+NRE	76.7±0.9	92.7±0.2	66.2±1.1	**93.2±0.2**	80.0±0.6	94.1±0.2
Node2Vec	58.1±1.5	87.1±0.4	32.7±2.2	88.4±0.5	62.3±1.9	89.1±0.6
ARGVA	62.3±9.4	89.2±1.9	17.9±29.	89.5±2.2	50.3±28.	88.7±3.1
LSDAN	63.5±4.1	89.4±1.0	47.0±19.	91.2±1.3	63.4±3.7	90.1±0.7
GRAB (ours)	**80.4±0.2**	**93.0±0.1**	**69.7±0.4**	92.9±0.1	**85.0±0.1**	**94.9±0.0**

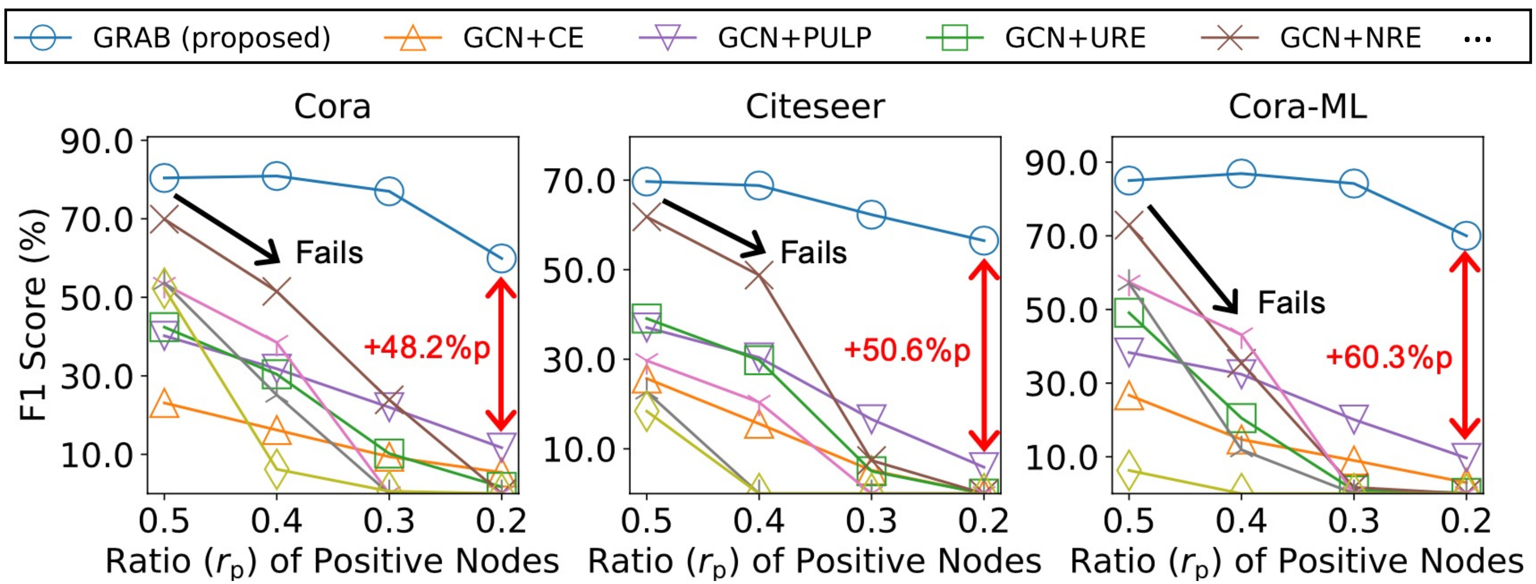
No Class Prior

- **Q2.** How well do competitors work without π_p?
  - The baselines show consistently lower accuracy
  - The improvement of GRAB is more significant

Method	Cora		Citeseer		Cora-ML	
	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)
GCN+CE	23.1±1.5	84.4±0.2	25.7±2.3	89.6±0.2	26.7±2.7	85.7±0.3
GCN+PULP	40.2±1.7	86.1±0.2	37.1±3.0	90.3±0.4	38.3±1.3	86.4±0.2
GCN+URE	42.4±1.5	86.8±0.2	39.1±2.0	90.6±0.2	49.1±3.6	88.6±0.5
GCN+NRE	70.1±1.6	91.4±0.3	61.8±1.9	92.8±0.2	72.9±2.0	92.5±0.4
Node2Vec	53.3±2.1	87.0±0.6	29.7±2.2	88.9±0.3	57.4±1.7	88.8±0.4
ARGVA	53.7±16.	88.1±2.4	22.7±30.	89.8±2.2	57.1±21.	89.7±2.5
LSDAN	52.3±3.9	87.5±0.6	18.4±25.	89.8±2.2	6.3±17.	83.9±1.6
GRAB (ours)	**80.4±0.2**	**93.0±0.1**	**69.7±0.4**	**92.9±0.1**	**85.0±0.1**	**94.9±0.0**

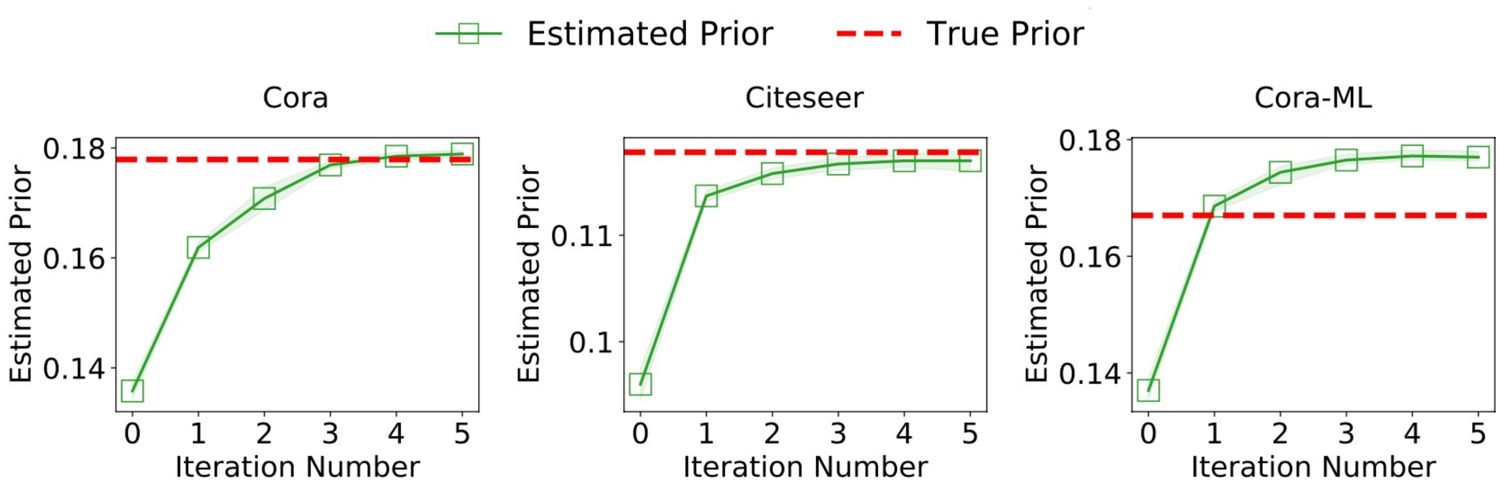
Fewer Observations

- **Q3.** Does GRAB work well with smaller r_{p}?
  - r_{p} refers to the ratio of **observed** positive nodes
  - GRAB works well with small r_{p} unlike competitors



Prior Estimation

- **Q4.** Is the unknown prior estimated well?
  - GRAB updates an estimation through iterations
  - GRAB finds the unknown π_{p} well from given data



Outline

- Introduction
- Previous Works
- Proposed Method
- **Conclusion**

Conclusion

- We propose **GRAB** for accurate PU learning
  - GRAB does not require the class prior as an input
- **Main ideas**
  - **Idea 1.** Model the graph as a Markov network
    - To design an objective function based on $p(\mathbf{z}|\mathbf{X}, \mathbf{y})$
  - **Idea 2.** Update an estimate $\hat{\pi}_{\mathrm{p}}$ through iterations
    - GRAB runs **marginalization** and **update** steps
- **Experiments**
  - GRAB consistently outperforms existing methods

Thank you!

Jaemin Yoo (jaeminyoo@snu.ac.kr)