Scale robust community prediction with node embedding based negative sampling

신동혁, 최현준, 이현지

Interpretation of datasets

- **Problem Scenario**
  - ○ Base dataset(=paper author.txt) has community information, not only link information.
  - ○ Test dataset(=query public, query private) also has community information.
  - ○ Base dataset only provides "positive communities" which we have confidence that they exist.

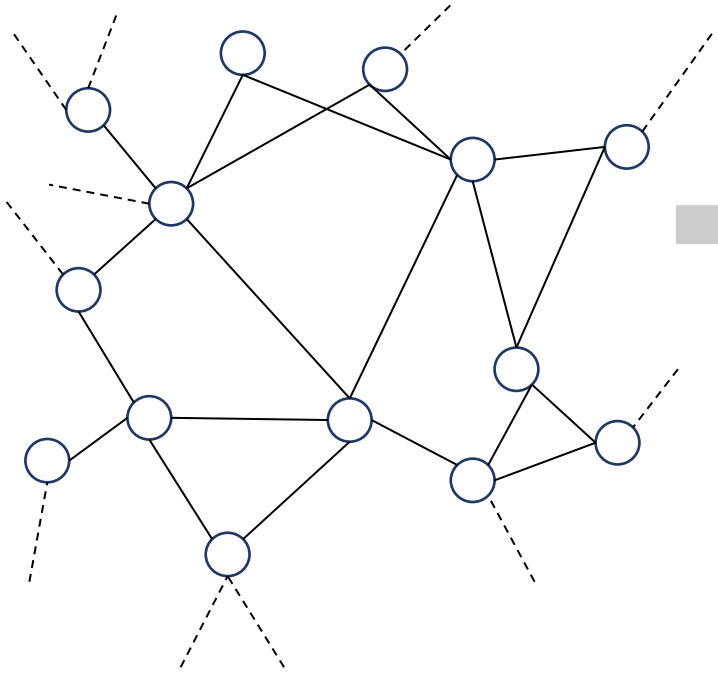
- **Assumptions**
  - ○ Base dataset graph is already formed.
  - ○ Community will be formed based on similar nodes.
  - ○ Community is complete graph
  - ○ Base and test dataset graphs are unweighted graphs

- **Classification vs. Clustering in Community Prediction**
  - ○ Classification accuracy >> Clustering accuracy

→ COMMUNITY PREDICTION task using Classification

Total Flow

Step 1.
Construct Network graph



Step 2.
Sampling Negative Community

Negative Communities

2362 30008
39072 54714
...
29068 9855 32756 55586 36631
...

Step 3.
Data encoding & Training Classifier

Positive Communities

4512 6350 32031
12354 25979 28348 33994 40349
47410
...
...

Step 1. Construct Network graph

Input

paper_author.txt

...
4512 6350 32031
12354 25979 28348 33994 40349 47410
...
...

- Base dataset which contain (positive) communities

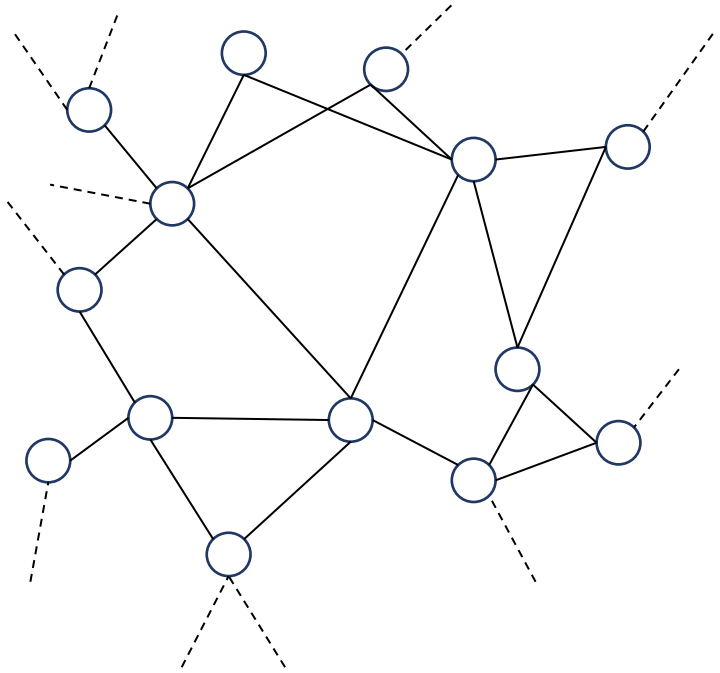
Split the communities

paper_author_links

4512 6350
4512 32031
6350 32031
12354 25979
12354 28348
12354 33994
...
40349 47410
...

- If community consists of more than 3 nodes(=authors), Split into links which are combination of nodes

Output



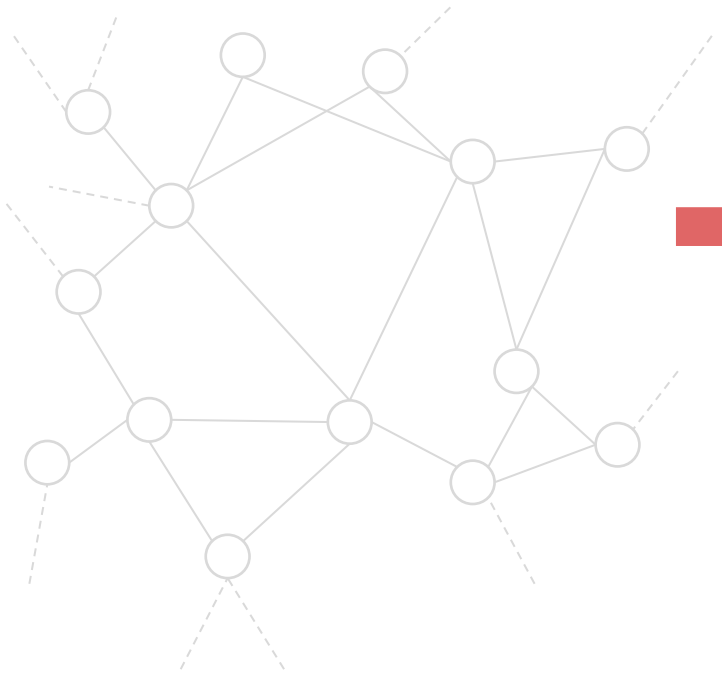
- Network Graph

Total Flow

Step 1.
Construct Network graph

Step 2.
Sampling Negative Community

Step 3.
Data encoding & Training Classifier



Negative Communities

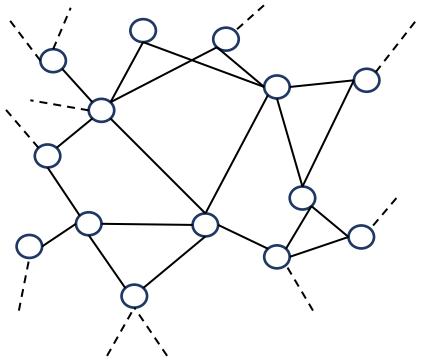
2362 30008
39072 54714
...
29068 9855 32756 55586 36631
...

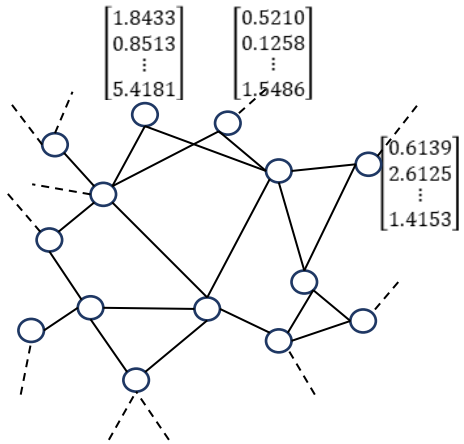
Positive Communities

4512 6350 32031
12354 25979 28348 33994 40349
47410
...
...

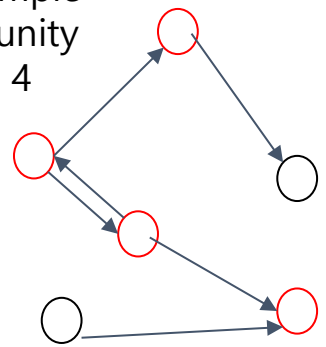
Step 2. Sampling Negative Community

Input → node2vec → Negative Sampling → **Output**





(ex) sample community of size 4



Negative Community Samples

2362 30008
39072 54714
…
29068 9855 32756 55586 36631
…

- Network Graph

- Find each node's embedding vector by using node2vec method

For link(=community of 2),
1. Just pick one from directed graph

For community of more than 3,
1. Make a directed graph with lowest similarity of each node connected
2. Pick the bidirectional pairs from the graph
3. Draw a community based on the pairs

- Negative Community Samples

Total Flow

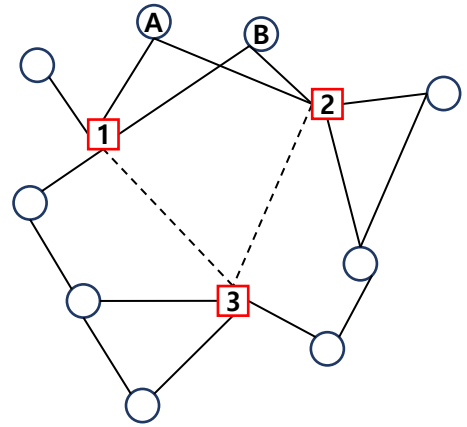
Step 1.
Construct Network graph

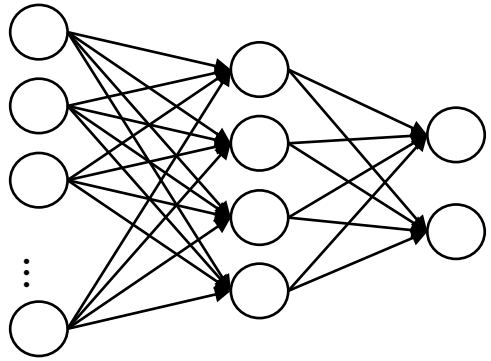
Step 2.
Sampling Negative Community

Negative Communities

2362 30008
39072 54714
...
29068 9855 32756 55586 36631
...

Step 3.
Data encoding & Training Classifier





Positive Communities

4512 6350 32031
12354 25979 28348 33994 40349
47410
...
...

Step 3. Data encoding & Training Classifier

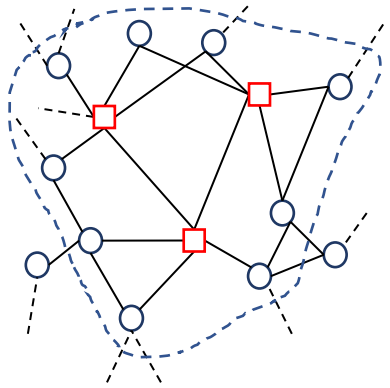
Input → Extract subgraphs → Subgraph labeling → Data encoding → Classification → **Output**

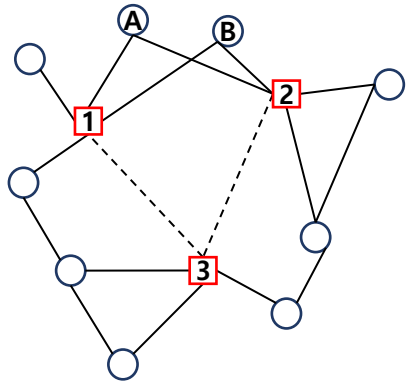
Positive Communities

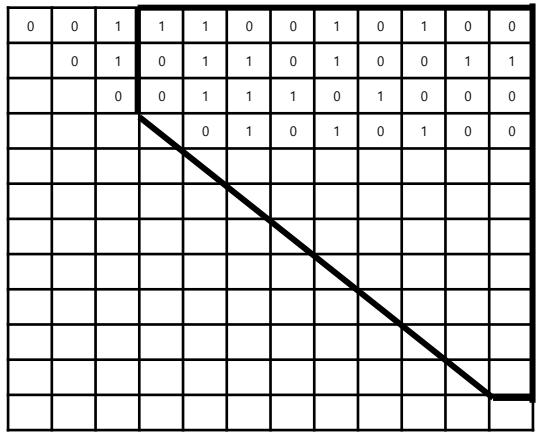
4512 6350 32031
12354 25979 28348 33994 40349 47410
...
...

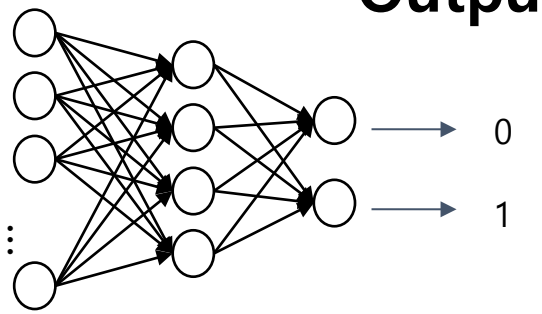
Negative Communities

2362 30008
39072 54714
...
29068 9855 32756 55586 36631
...









- Network graph
- Training communities:
  1. Positive communities (paper author)
  1. Negative communities (output from prev set)

- Extract enclosing subgraph for a target communities

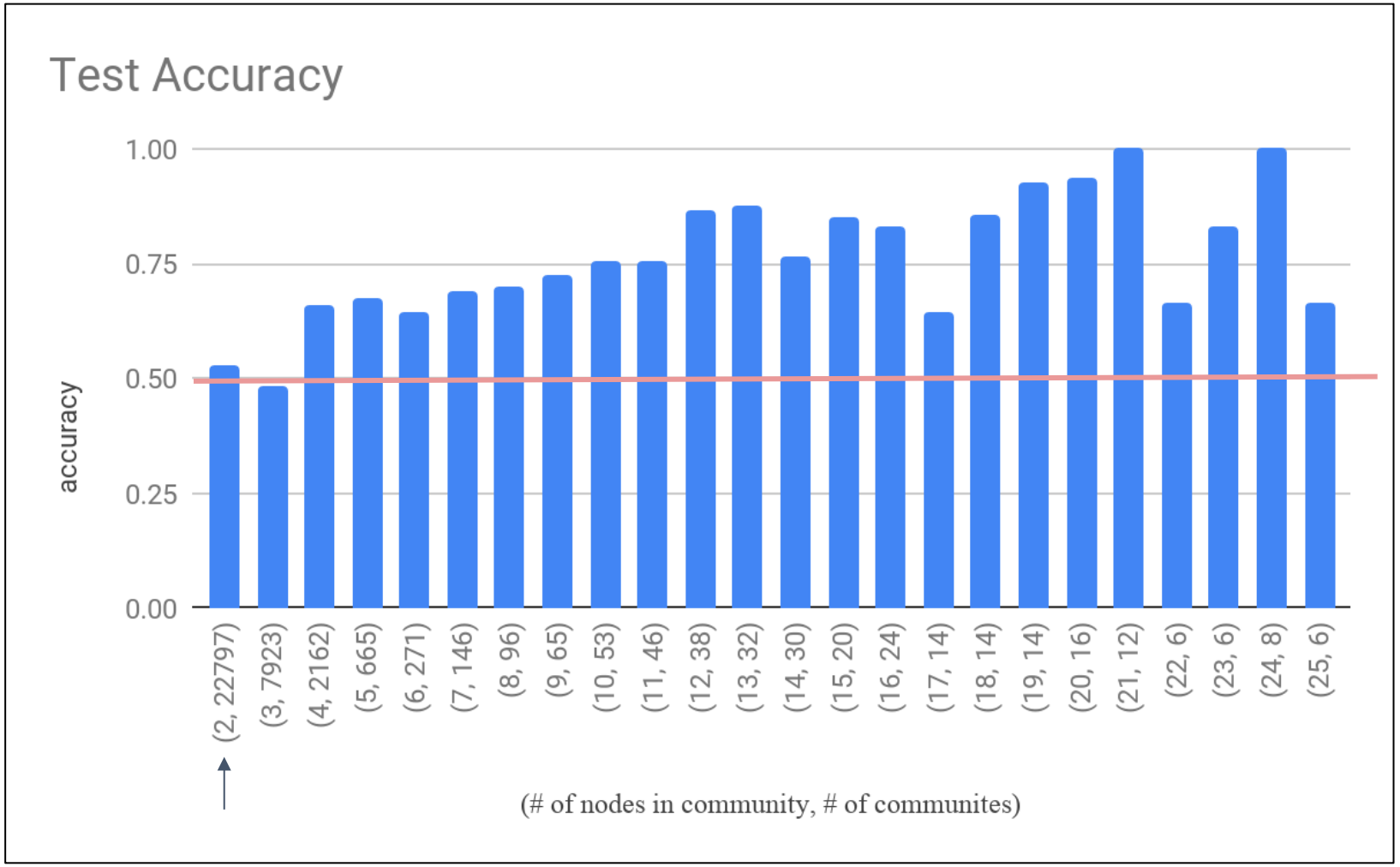
- Impose a vertex ordering based embedding vector

If $GM\big(d(A,1),d(A,2),d(A,3)\big) \geq GM\big(d(B,1),d(B,2),d(B,3)\big)$,
Assign $order\ A \geq order\ B$

- Construct adjacency matrix using the calculated vertex ordering

- Train the classifier

Results



TP	10939
FP	9605
TN	7370
FN	6086
Precision	0.53247
Recall	0.59747
F1 score	0.56310
Accuracy	0.5385

Contribution

Our team propose new method that …

1. Do negative sampling based on node embedding to perform classification task
2. Is robust to diverse scales of community
3. Use graph labeling which preserve features of community structure by node embedding
4. Is extensible to other classification models (GNN, MLP etc)

Thanks ☺

신동혁: tlsehdgur0@kaist.ac.kr

최현준: juneir@kaist.ac.kr

이현지: alee6868@kaist.ac.kr