Machine Learning with Graphs (MLG)

HW1: Learning to Identify High Betweenness Nodes

Deadline: 2021.03.19 (Fri.) 23:59

Submission: Code (.py/.ipynb) and Report (PDF)

HW1: Learning to Identify High BC Nodes

- Implement the paper that uses NN-based learning to identify high BC nodes
  - "*Learning to Identify High Betweenness Centrality Nodes from Scratch: A Novel Graph Neural Network Approach*" ACM CIKM 2019 https://arxiv.org/abs/1905.10418
  - Any package can be used, e.g., PyTorch, Tensorflow, Keras NetworkX, DGL, and PyTorch Geometric
  - Highly recommend **PyTorch Geometric**
    - A geometric deep learning extension library for PyTorch
    - https://github.com/rusty1s/pytorch_geometric
    - https://www.google.com/search?q=pytorch%20geometric
- HW1 is for you to get familiar with NN implementations and graph/network analysis packages

HW1: Learning to Identify High BC Nodes

- **2 test datasets**: ~5K synthetic graph, ~1M YouTube graph
  - Ground-truth ranking, i.e., high BC nodes, is provided
- Evaluation metrics (defined in the paper)
  - Recommendation metric: Top-N% Accuracy (N=1, 5, 10)
  - Ranking metric: Kendall tau distance
  - Wall-clock running time (in seconds)
- Reference code (official implementation in Tensorflow)
  - https://github.com/FFrankyy/DrBC
  - Note: You can NOT copy & paste the code. We will check the code plagiarism!
- You can follow all of the settings mentioned in the paper
  - E.g., Hyperparameter settings in Table 2

You implementation is not required to exactly follow the paper, i.e., you can do any you feel reasonable to improve the method

HW1: Learning to Identify High BC Nodes

- Compared methods
  - RK https://github.com/ecrc/BeBeCA
  - k-BC https://github.com/ecrc/BeBeCA
  - KADABRA https://github.com/natema/kadabra
- Required reproducibility
  - Table 3, 4, 5, 6, 7, 8, and 9 in the paper
- Bonus: use DrBC to find high-closeness and high-CC nodes
- HW1 Submission via Moodle
  - Deadline: **23:59, March 19 (Friday), 2021**
  - Submit your code: **.py** or **.ipynb** (preferred)
  - Submit a report (PDF): ≥10 pages (you cannot include code in report)
- Content in the report
  - Implementation details (describe how you implement)
  - Reproduced experimental results, along with analysis and insights
- Learning to learn by yourself. Good Luck!

Final Remark

- **What if I cannot successfully reproduce DrBC by following the detailed settings of the paper?**
  - Need not to totally follow the paper!!
    (The paper is not always correct)
  - You can modify DrBC to make the model converged and generate accurate results
    - Write the details of what you modify in the report
  - Figure out what are the potential problems
    - Write the possible reasons in the report
- **What if I totally cannot understand DrBC algorithm?**
  - You should come up with a supervised learning method (not necessary NN models) to train and predict BC values
    - Follow the experimental settings of DrBC
    - Write the details/results in the report

But you may receive a lower score, it depends on you method