ANGEL: efficient, and effective, node-centric community discovery in static and dynamic networks

By Rossetti et al

EGOR DMITRIEV, Utrecht Univeristy, The Netherlands

Our approach is primarily designed for social networks analysis and belongs to a well-known subfamily of Community Discovery approaches often identified by the keywords bottom-up and node-centric

1 GOALS

- we propose ANGEL , an algorithm that aims to lower the computational complexity of previous solutions while ensuring the identification of high-quality overlapping partitions

2 PRELIMINARIES

- ...

3 CHALLENGES

- complex networks researchers agree that it is not possible to provide a single and unique formalization that covers all the possible characteristics a community partition may satisfy

4 PREVIOUS WORK / CITATIONS

- (Coscia et al. 2012): where the authors propose DEMON an approach whose main goal was to identify local communities by capturing individual nodes perspectives on their neighbourhoods and using them to build mesoscale ones
-
- **This Work:**
  - Introduces a Label Propagation algorithm
    * Least complex kind of algorithm
    * Gives good quality results
  - In contrast to DEMON it focuses on lowering the time complexity while at the same time increasing the partition quality
  - Properties:
    * It produces a deterministic output
    * Allows for a parallel implementation

5 DEFINITIONS

- During each iteration, the label of v is updated to the majority label of its neighbours. As the labels propagate, densely connected groups of nodes quickly reach a consensus on a unique label

6 OUTLINE / STRUCTURE

- Node Labeling $O(n + m)$ (Raghavan et al. 2007)
  - Initialize the labels at all nodes in the network. For a given node x, $C_x(0) = x$
  - Set $t = 1$
  - Arrange the nodes in the network in a random order and set it to X

- For each $x \in X$ chosen in that specific order, let $C_x(t) = f\left(C_{x_{\mathrm{ia}}}(t), \ldots, C_{x_{i\ \mathrm{m}}}(t), C_{x_{i(m+1)}}(t-1), \ldots\right.$
  f here returns the label occurring with the highest frequency among neighbors and ties are broken uniformly randomly.
- If every node has a label that the maximum number of their neighbors have, then stop the algorithm. Else, set $t = t + 1$ and go to (3)

- Community Matching:
  - Don't make use of the Jaccard similarity – a widely adopted strategy to address this kind of approaches
  - Each node has multiple labels
    - The ratio of nodes in it that already belongs to y w.r.t. the size of x:
      - Ratio is greater than (or equal to) a given threshold, the merge is applied and the node label updated
  - We assume that each node at time t carries three sets of labels
    - The identifiers of the communities it currently belongs to t
    - The identifiers of the communities it was part of at $t1$
    - he identifiers of the communities it will be associated to at $t + 1$
  - Event detection:
    - Birth (B): a community born at time t if there are no network substructures at t − 1 that can be matched with it
    - Merge: two or more communities at time t merge iff they are matched to the same network substructure at t + 1
    - Split (S): a community at t splits if it is matched to multiple network substructures at t + 1
    - Continue (C): a community at t remains the same at t + 1;
    - Death (D): a community dies at t if it is not matched with any network substructure at t + 1.

7 EVALUATION

- $\Psi(\mathcal{A}, \mathcal{B}) = \mathcal{A} \cap \mathcal{B} - (\mathcal{A} - \mathcal{B})$: Quality metric to relate discovered events A to ground truth ones B

8 CODE

- ...

9 RESOURCES

- ...