Leader-driven community detection

By

Prajwal M P 192IT15
Bavya BalaKrishnan 192IT03

INTRODUCTION

- **Leader-driven community detection algorithms for community detection in large-scale complex networks.**
- Identify some particular nodes in the target network, called leader nodes, around which local communities can be computed.
- New way for evaluating performances of community detection algorithms

Transforming data clustering problems into a community detection problems

- Real world complex networks exhibit a level of organization, called communities.
- A community is defined as a connected subgraph whose nodes are much linked with one each other than with nodes outside the subgraph.
- Useful in computation distribution,huge graph visualization and large-scale graph compression
- Different types of community detection algorithms are there:
  - Disjoint communities detection
  - Overlapping communities detection
  - Local community identification

RELATED WORK

Community detection approaches

- **Group-based approaches**

Identifying groups of nodes that are highly connected or share some strong connection patterns.

1. High mutual connectivity(maximal clique or to a γ -quasi-clique)
2. High internal reachability(k-clique,k-core subgraph,k-community)

- **Network-based approaches**

  Considering the whole connection patterns in the network.

  Modularity

  Let P = {C 1 , . . . , C k } a partition of the node's set V of a graph. The modularity of the partition P is given by:

Assumptions in modularity optimization approaches

- The best partition of a graph is the one that maximize the modularity.
- If a network has a community structure, then it is possible to find a precise partition with maximal modularity.
- If a network has a community structure, then partitions inducing high modularity values are structurally similar.

Seed centric

- Leader-driven algorithms constitute a special case of seed centric approaches.
- Nodes of a network are classified into two categories: leaders and followers.
- Leaders represent communities.
- An assignment step is applied to assign followers nodes to most relevant communities.

Community evaluation approaches

- Evaluation on networks for which a ground-truth decomposition into communities is known.

  we can use classical external clustering evaluation indices to evaluate and compare community detection algorithms

  ### Adjusted Rand Index (ARI) :

  $P_i = \{P_{i_1}, \ldots, P_{i_l}\}$, $P_j = \{P_{j_1}, \ldots, P_{j_k}\}$ be two partitions of a set of nodes V .

$$\mathrm{ARI}(P_i, P_j) = \frac{\sum_{x=1}^{l}\sum_{y=1}^{k}\binom{|P_i^x \cap P_j^y|}{2} - t_3}{\frac{1}{2}(t_1 + t_2) - t_3}$$

where:

$$t_1 = \sum_{x=1}^{l}\binom{|P_i^x|}{2}, \quad t_2 = \sum_{y=1}^{k}\binom{|P_j^y|}{2}, \quad t_3 = \frac{2t_1t_2}{n(n-1)}$$

  Normalized Mutual Information (NMI)

  We seek to quantify how much we reduce the uncertainty of the clustering of randomly picked element from V in a partition Pj if we know Pi .

METHODOLOGY

Algorithm 2 LICOD algorithm

```
Require: G =< V, E > a connected graph
ℒ ← Ø {set of leaders}
for v ∈ V do
  if isLeader(v) then
    ℒ ← ℒ ∪ {v}
  end if
end for
𝒞 ← computeComumunitiesLeader(ℒ)
for v ∈ V do
  for c ∈ 𝒞 do
    M[v, c] ← membership(v, c) {see equation 6}
  end for
  P[v] = sortAndRank(M[v])
end for
repeat
  for v ∈ V do
    P*[v] ← rankAggregate_{x∈{v}∩Γ_G(v)} P[x]
    P[v] ← P*[v]
  end for
until Stabilization of P*[v]∀v
for v ∈ V do
  /* assigning v to communities */
  for c ∈ P[v] do
    if |M[v, c] − M[v, P[0]]| ≤ ϵ then
      COM(c) ← COM(c) ∪ {v}
    end if
  end for
end for
return 𝒞
```

Algorithm is implemented using the igraph graph analysis toolkit

Function is Leader () :

Based on nodes centralities

- Degree centrality
- Betweenness centrality

A node is identified as a leader if its centrality is greater or equal to $\sigma \in [0, 1]$ percent of its neighbors centralities.

Function computecommunitiesleaders

Two leaders are grouped in the same community if the ratio of common neighbors to the total number of neighbors is above a given threshold $\delta \in [0, 1]$.

Function membership(v, c)

$$membership(v, c) = \frac{1}{(min_{x \in COM(c)} SPath(v, x)) + 1}$$

Rank aggregation approaches

- Requirement:minimum number of pairwise disagreements
- Borda's method

$B_{L_k}(i) = \{\text{count}(j) | L_k(j) < L_k(i) \,\&\, j \in L_k\}$. The total Borda's score for an element is then: $B(i) = \sum_{t=1}^{k} B_{L_t}(i)$.

- Kemeny optimal aggregation

si is preferred to sj ,if the majority of rankers ranks si before sj

Community assignment

Threshold epsilon controls the degree of desired overlapping

Datasets

Dataset	# Nodes	# Edges	# Real communities
Zachary	34	78	2
Football	115	616	11
US Politics	100	411	2
Dolphin	62	159	2

- Centrality metrics = [Degree centrality (dc), Betweenness centrality (BC),Eigen Vector Centrality]
- Voting method = [Borda, Local Kemeny]
- $\sigma \in [0.5, 0.6, 0.7, 0.8, 0.9, 1.0]$
- $\delta \in [0.5, 0.6, 0.7, 0.8, 0.9, 1.0]$
- epsilon $\in [0.0, 0.1, 0.2]$

Task Driven Evaluation

- Evaluation in function of the topological features of computed communities.
- Task-driven evaluation.
  - Let T be a task where community detection can be applied.
  - Performance measure for T execution applying the community detection algorithm
  - Here we use data clustering as an evaluation task
  -

Table 3 Characteristics of used datasets

Dataset	Glass	Iris	Wine	Vehicle	Abalone
#Instances	214	150	178	846	772
#Attributes	10	4	13	18	8
#Classes	7	3	3	4	29