Discovering social circles in a directed social network using node, structure and edge features

Group 4: Joyce Yang, Muneeb Shahid

Outline

- Introduction
- Problem Definition
- Proposed Method
- Experiments
- Conclusion and Future Work

Introduction

- Social Circles Discovery Problem
- Applications of Social Circles Discovery
- Motivation for our idea

Problem Definition

- Input: User's Signed Social Network (directed)
- Determine best number of active centers
- Discover best set of active centers
- Form social circles around each of the selected active center
- Difference with traditional community detection problem

Node-Edge K-Means Clustering

- Social circle detection
  - Node features
  - Network structures
  - Link feature

K-Means Clustering

P1: existence of link from active center to node
P2: existence of link from node to active center
P3: profile similarities
P4_1: strength of ties between active center and node
P4_2: strength of ties between node and active center
P5_1: trust level that active center have to node
P5_2: trust level that node have to active center

Algorithm 1: Assign social circles for each node

Input: A chromosome consists of k active centers $X = x_1, x_2, ...x_k$, and set of users $U = (V - X)$ to be clustered.
Output: set of predicted circles $\{\hat{C}_1, \hat{C}_2, ..., \hat{C}_k\}$
while $U \neq \emptyset$ **do**
  pick up a node $u \in U$ **initialize:** $max_s = 0,\ AC = -1$
  for $i = 1$ *to* k **do**
    $x_i \in X$
    initialize: $p_1 = 0,\ p_2 = 0,\ p_3 = 0,\ p_{4_1} = 0,\ p_{4_2} = 0,\ p_{5_1} = 0,\ p_{5_2} = 0$
    Compute values of $p_1, p_2, p_3, p_{4_1}, p_{4_2}, p_{5_1}, p_{5_2}$ between u and the active centre x_i
    if $max_s \leq p_3 + p_{4_1} + p_{4_2} + p_{5_1} + p_{5_2}$ **then**
      $max_s = p_3 + p_{4_1} + p_{4_2} + p_{5_1} + p_{5_2}$
      AC = x_i
    end
  end
  if $AC \neq -1$ **then**
    Add node u in the cluster of AC
  end
end

K-Means Clustering

- Profile similarities

$$prof\_sim(x, j) = \frac{\sum_{i=1}^{p} \delta_i(x, j)}{(p)},$$

$$\delta_i(x, j) = \frac{1}{\sqrt{\sum_{i=1}^{p} (x_i - j_i)^2}},$$

Algorithm 1: Assign social circles for each node

Input: A chromosome consists of k active centers $X = x_1, x_2, ...x_k$, and set of users $U = (V - X)$ to be clustered.

Output: set of predicted circles $\{\hat{C}_1, \hat{C}_2, ..., \hat{C}_k\}$

```
while U ≠ ∅ do
    pick up a node u ∈ U initialize: max_s = 0, AC = −1
    for i = 1 to k do
        x_i ∈ X
        initialize: p_1 = 0, p_2 = 0, p_3 = 0, p_4_1 = 0, p_4_2 =
                    0, p_5_1 = 0, p_5_2 = 0
        Compute values of p_1, p_2, p_3, p_4_1, p_4_2, p_5_1, p_5_2
          between u and the active centre x_i
        if max_s ≤ p_3 + p_4_1 + p_4_2 + p_5_1 + p_5_2 then
            max_s = p_3 + p_4_1 + p_4_2 + p_5_1 + p_5_2
            AC = x_i
        end
    end
    if AC ≠ −1 then
        Add node u in the cluster of AC
    end
end
```

K-Means Clustering

- Strength of Ties

$$str(x,u) = \frac{1}{deg_{out}(x) + deg_{in}(u) - 1}$$

Algorithm 1: Assign social circles for each node

Input: A chromosome consists of k active centers $X = x_1, x_2, ...x_k$, and set of users $U = (V - X)$ to be clustered.

Output: set of predicted circles $\{\hat{C}_1, \hat{C}_2, ..., \hat{C}_k\}$

```
while U ≠ ∅ do
    pick up a node u ∈ U initialize: max_s = 0, AC = −1
    for i = 1 to k do
        x_i ∈ X
        initialize: p_1 = 0, p_2 = 0, p_3 = 0, p_4_1 = 0, p_4_2 =
                    0, p_5_1 = 0, p_5_2 = 0
        Compute values of p_1, p_2, p_3, p_4_1, p_4_2, p_5_1, p_5_2
          between u and the active centre x_i
        if max_s ≤ p_3 + p_4_1 + p_4_2 + p_5_1 + p_5_2 then
            max_s = p_3 + p_4_1 + p_4_2 + p_5_1 + p_5_2
            AC = x_i
        end
    end
    if AC ≠ −1 then
        Add node u in the cluster of AC
    end
end
```

Objective Functions

- $deg\_cen_k^C(x)$ and $deg\_cen_k^R(x)$ means the average connectivity coefficient of x with the members of $C^{[k]}$ and $R^{[k]}$ respectively.
- $prof\_sim_k^C(x)$ and $prof\_sim_k^R(x)$ means the average profile similarity of x with the members of $C^{[k]}$ and $R^{[k]}$ respectively.
- $str_k^C(x)$ and $str_k^R(x)$ means the average strength of ties of x with the members of $C^{[k]}$ and $R^{[k]}$ respectively.
- $trust_k^C(x)$ and $trust_k^R(x)$ means the average trust level between x and members of $C^{[k]}$ and $R^{[k]}$ respectively.

Algorithm 2: Calculate fitness value for a set of k active centers

Input: Two N x k matrices $[x_{ij}]_{Nxk}$ and $[\hat{C_{ij}}]_{Nxk}$, N: number of population, k: number of circles

Output: Best possible set of k seeds x_1,x_2,...,x_k for Ego network segmentation

for 1 *to N* **do**

  initialize: $fitness = 0$

  pick up the i^{th} row from $[x_{ij}]_{Nxk}$ and $[\hat{C_{ij}}]_{Nxk}$ **for** 1 *to k* **do**

    initialize: $obj = 0$

    $Obj(j) = [deg\_cen_j^C(x_i) - deg\_cen_j^R(x_i) + prof\_sim_j^C(x_i) - prof\_sim_j^R(x_i) + str_j^C(x_i) - str_j^R(x_i) + trust_j^C(x_i) - trust_j^R(x_i)]$

  end

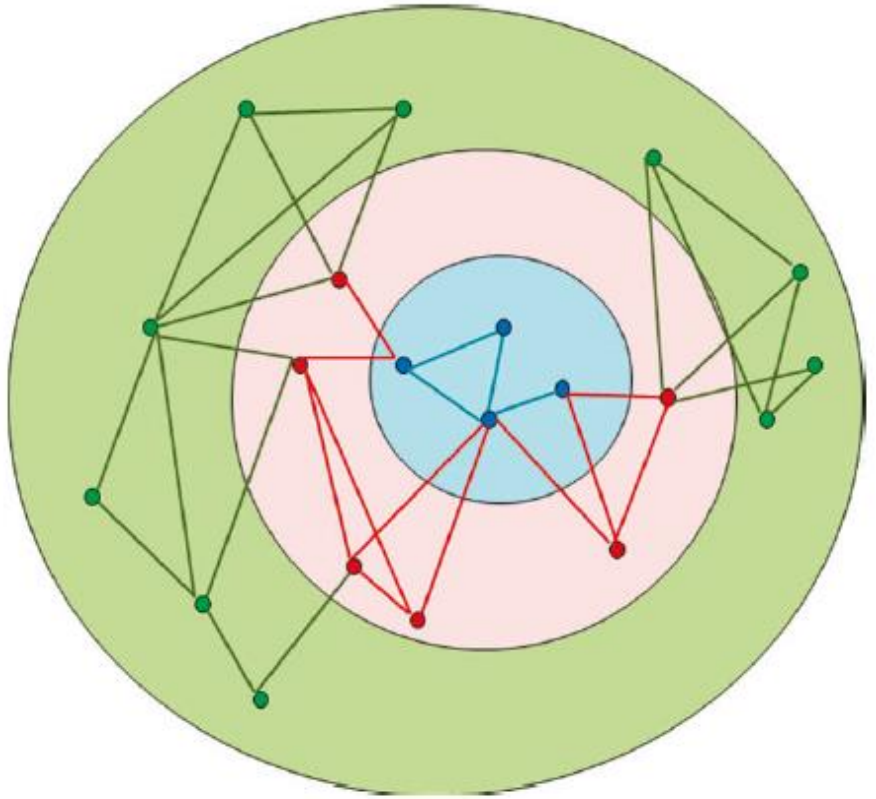
  $fitness = f(X_i) = \frac{\sum_{j=1}^{k} Obj(j)}{k}$

end

Objective Functions

- **Core area**: members of the circle

- **Residual area:**the immediate neighbor of any of the member of the circle and provide interaction information with members ofthe circle.



Objective Functions

- Degree Centrality

$$deg\_cen_x^c = \frac{(\#\ of\ Link_{in}^c(x)\ +\#\ of\ Link_{out}^c(x))}{(n-1)}$$

Algorithm 2: Calculate fitness value for a set of k active centers

Input: Two N x k matrices $[x_{ij}]_{Nxk}$ and $[\hat{C_{ij}}]_{Nxk}$, N: number of population, k: number of circles

Output: Best possible set of k seeds $x_1, x_2, \ldots, x_k$ for Ego network segmentation

for 1 *to* N **do**

  initialize: $fitness = 0$

  pick up the i^{th} row from $[x_{ij}]_{Nxk}$ and $[\hat{C_{ij}}]_{Nxk}$ **for** 1 *to* k **do**

    initialize: $obj = 0$

    $Obj(j) = [deg\_cen_j^C(x_i) - deg\_cen_j^R(x_i) + prof\_sim_j^C(x_i) - prof\_sim_j^R(x_i) + str_j^C(x_i) - str_j^R(x_i) + trust_j^C(x_i) - trust_j^R(x_i)]$

  end

  $fitness = f(X_i) = \frac{\sum_{j=1}^{k} Obj(j)}{k}$

end

Find best objective value

- Crossover and Mutation

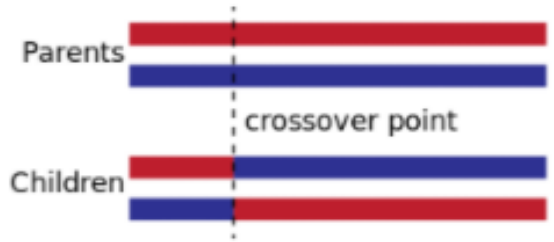
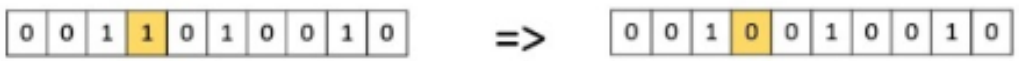


Figure 1: Example of Crossover



Algorithm 3: Find the best set of active centers for social circles discovery (ideal)

Consider population matrix $pop(X) = [X_i]_{Nx1} = [x_{ij}]_{Nxk}$ and fitness matrix $F = [f(X_i)]_{1xN}$ and generate an augmented matrix $Q = [X_i|f(X_i)]_{Nx(k+1)}$

Then sort matrix Q in descending order of fitness value $f(X_i)$

while *changes in highest fitness value during* 10 *consecutive iterations appears* **do**

  for 1 *to* N **do**

    1. Randomly select two parent chromosomes with relative high fitness value. For example, from top 10 of the descending Q.

    2.**[CROSSOVER]:**

    Generate a random (integer) number *randc_pos* from the range [1,k], and exchange the alleles of chromosomes X_1 and X_2 at random position(*randc_pos*) to produce two new chromosomes X_1^{new} and X_2^{new}

    compare the fitness of X_1, X_2, X_1^{new}, X_2^{new} and feed the one with best fitness value to mutation

    3.**[MUTATION]:**

    Generate a random position *randm_pos* in the range [1,k] and *rand_id* in the range [1,n], then mutate the allele which is at *randm_pos* by *rand_id*.

    Compare the X^{new}'s fitness value with the Q_i 's fitness value.

    $Q_i \Leftarrow X^{new}$ if X^{new} has a better fitness value

  end

  Sort matrix Q in descending order of fitness value $f(X_i)$ again for next round.

end

Summary

Algorithm 1: Assign social circles for each node

Input: A chromosome consists of k active centers $X = x_1, x_2, ...x_k$, and set of users $U = (V - X)$ to be clustered.

Output: set of predicted circles $\{\hat{C}_1, \hat{C}_2, ..., \hat{C}_k\}$

while $U \neq \emptyset$ **do**
- pick up a node $u \in U$ **initialize:** $max_s = 0,\ AC = -1$
- **for** $i = 1$ *to* k **do**
  - $x_i \in X$
  - **initialize:** $p_1 = 0,\ p_2 = 0,\ p_3 = 0,\ p_{4_1} = 0,\ p_{4_2} = 0,\ p_{5_1} = 0,\ p_{5_2} = 0$
  - Compute values of $p_1, p_2, p_3, p_{4_1}, p_{4_2}, p_{5_1}, p_{5_2}$ between u and the active centre x_i
  - **if** $max_s \leq p_3 + p_{4_1} + p_{4_2} + p_{5_1} + p_{5_2}$ **then**
    - $max_s = p_3 + p_{4_1} + p_{4_2} + p_{5_1} + p_{5_2}$
    - AC = x_i
  - **end**
- **end**
- **if** $AC \neq -1$ **then**
  - Add node u in the cluster of AC
- **end**

end

Algorithm 2: Calculate fitness value for a set of k active centers

Input: Two N x k matrices $[x_{ij}]_{Nxk}$ and $[\hat{C_{ij}}]_{Nxk}$, N: number of population, k: number of circles

Output: Best possible set of k seeds $x_1, x_2, ..., x_k$ for Ego network segmentation

for 1 *to* N **do**
- **initialize:** $fitness = 0$
- pick up the i^{th} row from $[x_{ij}]_{Nxk}$ and $[\hat{C_{ij}}]_{Nxk}$ **for** 1 *to* k **do**
  - **initialize:** $obj = 0$
  - $Obj(j) = [deg\_cen_j^C(x_i) - deg\_cen_j^R(x_i) + prof\_sim_j^C(x_i) - prof\_sim_j^R(x_i) + str_j^C(x_i) - str_j^R(x_i) + trust_j^C(x_i) - trust_j^R(x_i)]$
  - **end**
- $fitness = f(X_i) = \frac{\sum_{j=1}^{k} Obj(j)}{k}$

end

Algorithm 3: Find the best set of active centers for social circles discovery (ideal)

Consider population matrix $pop(X) = [X_i]_{Nx1} = [x_{ij}]_{Nxk}$ and fitness matrix $F = [f(X_i)]_{1xN}$ and generate an augmented matrix $Q = [X_i | f(X_i)]_{Nx(k+1)}$

Then sort matrix Q in descending order of fitness value $f(X_i)$

while *changes in highest fitness value during* 10 *consecutive iterations appears* **do**
- **for** 1 *to* N **do**
  - 1. Randomly select two parent chromosomes with relative high fitness value. For example, from top 10 of the descending Q.
  - 2.**[CROSSOVER]:**
  - Generate a random (integer) number *randc_pos* from the range [1,k], and exchange the alleles of chromosomes X_1 and X_2 at random position(*randc_pos*) to produce two new chromosomes X_1^{new} and X_2^{new}
  - compare the fitness of X_1, X_2, X_1^{new}, X_2^{new} and feed the one with best fitness value to mutation
  - 3.**[MUTATION]:**
  - Generate a random position *randm_pos* in the range [1,k] and *rand_id* in the range [1,n], then mutate the allele which is at *randm_pos* by *rand_id*.
  - Compare the X^{new}'s fitness value with the Q_i 's fitness value.
  - $Q_i \Leftarrow X^{new}$ if X^{new} has a better fitness value
  - **end**
- Sort matrix Q in descending order of fitness value $f(X_i)$ again for next round.

end

Optimization

- Parallel computing through multiprocessing package in python

Calculating fitness value for new populations

Algorithm 3: Find the best set of active centers for social circles discovery (ideal)

Consider population matrix $pop(X) = [X_i]_{Nx1} = [x_{ij}]_{Nxk}$ and fitness matrix $F = [f(X_i)]_{1xN}$ and generate an augmented matrix $Q = [X_i|f(X_i)]_{Nx(k+1)}$

Then sort matrix Q in descending order of fitness value $f(X_i)$

while *changes in highest fitness value during* 10 *consecutive iterations appears* **do**

  for 1 *to* N **do**

    1. Randomly select two parent chromosomes with relative high fitness value. For example, from top 10 of the descending Q.

    2.[**CROSSOVER**]:

    Generate a random (integer) number *randc_pos* from the range [1,k], and exchange the alleles of chromosomes X_1 and X_2 at random position(*randc_pos*) to produce two new chromosomes X_1^{new} and X_2^{new}

    compare the fitness of X_1, X_2, X_1^{new}, X_2^{new} and feed the one with best fitness value to mutation

    3.[**MUTATION**]:

    Generate a random position *randm_pos* in the range [1,k] and *rand_id* in the range [1,n], then mutate the allele which is at *randm_pos* by *rand_id*.

    Compare the X^{new}'s fitness value with the Q_i 's fitness value.

    $Q_i \Leftarrow X^{new}$ if X^{new} has a better fitness value

  end

  Sort matrix Q in descending order of fitness value $f(X_i)$ again for next round.

end

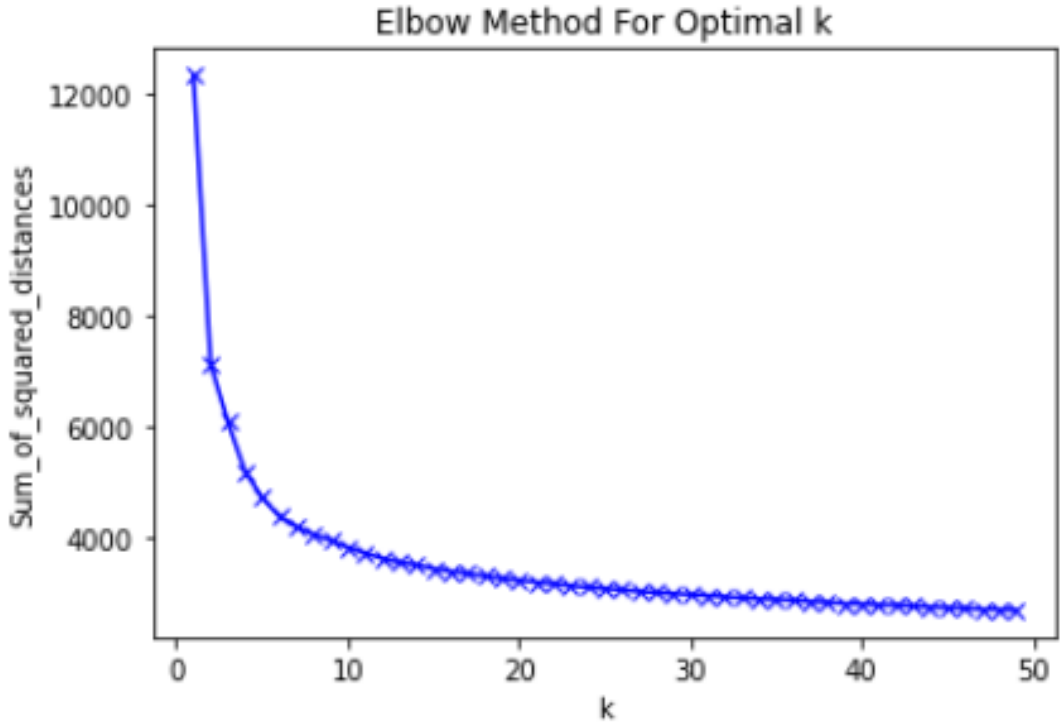
Experiments: Slashdot Dataset

Table 1: Dataset Statistics

Property	Value
No. of Nodes	82140
No. of Edges	549202
Avg. Clustering co-efficient	0.0588
Diameter	12
No. of features for each node	100

Configurable Parameters Choosen

- Recall:
  - **K** - number of active centers = 7
  - **N** - number of populations = 20
  - Relatively high fitness value in Algorithm = 10



Experiments: Evaluation Metrics

- Net Object Values of properties

Table 2: Net values of the properties

Property	No Trust	Trust
Net Degree Centrality for Social Circle	1.35	1.20
Net Degree Centrality for Residual	0.22	1.17
Net Strength of Ties for Social Circle	0.20	0.05
Net Strength of Ties for Residual	0.0008	0.02
Net Profile Similarity for Social Circle	0.06	0.06
Net Profile Similarity for Residual	0.009	0.05
Net Objective Value (overall difference)	1.38	0.07

Experiments: Evaluation Metrics

- Silhouette Coefficient Index

Table 3: Silhouette Coefficient index scores

Property	No Trust	Trust
Silhouette Coefficient Score	-0.20	-0.20

Experiments: Evaluation Metrics

- Davies Bouldin Index

Table 4: Davies Bouldin index scores

Property	No Trust	Trust
Davies Bouldin Score	6.80	6.03

Experiments: Evaluation Metrics

- Calinski Harabasz Index

Table 5: Calinski Harabasz index scores

Property	No Trust	Trust
Calinski Harabasz Score	36.6	57.02

Conclusion and Future work

- Lessons learned
- Future Work
  - Social behavior theory, such as " enemy of my enemy is my friend". Triad relationship
  - Determining different weightages for the features based on the given application
  - Graph summarization of the social circles for generating labels

Thanks! Questions?

Page of all resources

Table 2: Net values of the properties

Property	No Trust	Trust
Net Degree Centrality for Social Circle	1.35	1.20
Net Degree Centrality for Residual	0.22	1.17
Net Strength of Ties for Social Circle	0.20	0.05
Net Strength of Ties for Residual	0.0008	0.02
Net Profile Similarity for Social Circle	0.06	0.06
Net Profile Similarity for Residual	0.009	0.05
Net Objective Value (overall difference)	1.38	0.07

$$s = \frac{b - a}{max(a, b)}$$

$$R_{ij} = \frac{s_i - s_j}{d_{ij}}$$

$$DB = \frac{1}{K} \sum_{i=1}^{K} max(R_{ij})$$

$$s = \frac{tr(B_K)}{tr(W_K)} \frac{n_E - K}{k - 1}$$

$$W_K = \sum_{q=1}^{K} \sum_{x \epsilon C_q} (x - c_q)(x - c_q)^T$$

$$B_K = \sum_{q=1}^{K} (n_q)(c_q - c_E)(c_q - c_E)^T$$

Table 4: Davies Bouldin index scores

Property	No Trust	Trust
Davies Bouldin Score	6.80	6.03

Table 3: Silhouette Coefficient index scores

Property	No Trust	Trust
Silhouette Coefficient Score	-0.20	-0.20

Table 5: Calinski Harabasz index scores

Property	No Trust	Trust
Calinski Harabasz Score	36.6	57.02