Classify My Social Contacts into Circles
Stanford University
CS224W Fall 2014

Amer Hammudi
(SUNet ID: ahammudi)
ahammudi@stanford.edu

Darren Koh
(SUNet: dtkoh)
dtkoh@stanford.edu

Jia Li
(SUNet: jli14)
jli14@stanford.edu

1 Abstract

Manually maintaining the contact list of family, friends, business, and so on is a classical problem of contact list management. The soaring popularity and population of users in social media made managing contact list is a very tedious and time consuming task. In this paper, we proposed a topology-based approach to identify leaders then build social circles around those leaders. Our result is comparable to previous publications. We could apply this work to targeting content for friends list and targeting ads for business list.

2 Introduction

Using social media for our personal and business purposes has becoming an essential accessories in our daily life. For individual professional network (such as LinkedIn) you want to group recruiters for job hunting, group Alumni to keep up with mutual interest and etc. In the business aspect, imagine you have to manually group your shoppers in Electronic Shoppers, Apparel Shoppers and etc so that every week you can target them with specific promotions and sales in their category. It is very important for us to come out with a sophisticated way to automatically grouping contact list to remove the time and impactful of managing contact list.

In this paper, we use the Facebook Dataset [1], one of the largest social networking website. We are interested in the network structure and users profiles within an Ego network [2]

3 Previous Work

We reviewed some relevant literatures done by domain expert. Jure [2][3] demonstrates how to apply probabilistic method to model the circles as latent variables.

The authors attempt to find social circles in the Ego-network using probabilistic methodology that models the circles as latent variables. The model takes input $G(V, E)$ and solves for a set of K circles and maps each node in the Ego-Network to one or more circles.

At input they know the properties of all the nodes in the ego network. They first encode a similarity vector between any two nodes in the ego network $\phi(x, y)$. Each circle $\mathcal{C}_k$, $k \in [1, K]$ is going to be associated with a vector θ_k which denotes the information of the circle itself, this vector is learned from the empirical data. The likelihood that a pair of contacts belongs to a circle is related to the inner product of the encoded vector of the pair of contacts and the circle's descriptor vector $< \phi(x, y), \theta_k >$. The unsupervised machine learning model is used to optimize the log-likelihood of Equation 2 [2]

with θ_k and α_k the parameters of the model. They solve for the number of circles K, and for each circle $\mathcal{C}_k$, $k \in [1, K]$, its information vector θ_k, and the list of contacts $\{x \in \mathcal{C}_k\}$ identified belonging to the circle.

The similarity vector $\phi(x, y)$ could be built from a function $\sigma(x, y)$ which is a 0-1 vector that describes which common properties x and y share, or from the a certain function based on u as well $\sigma(x, u)$ and $\sigma(y, u)$. They also proposed compressed versions of $\sigma'(x, y)$, still producing a 0-1 vector for the common properties, but with a less dimension. The authors propose the tree structure to encode features from nodes (x, y) into $\sigma(x, y)$ in a clean way.

They ran the prediction over three datasets from Facebook, Google+, and Twitter, for the original and compressed version of the similarity-scores. They evaluate the performance using the Balanced Error Rate and F-1 Score. For Facebook data, using the original version of the similarity-score greatly improves the metrics relative to the compressed version. However it makes little difference for the other two datasets. One reason is that Facebook stores more user's information compared to Google+ and Twitter. As for the number of circles K, although the algorithm is designed for an arbitrary number of circles, empirically the optimized K is usually no more than 10.

Another similar work done by Yu and Lin [5] the authors proposed detecting social circles in ego networks using edge-based clustering algorithm. Their approach can be summarized in 3 steps.

1. Construct missing edges, using closness measure the authors precited and created missing edges

2. Find similarity of adjacent edges and cluster edges using single-linkage hierarchical clustering (using Jaccard Similarity for topological and Cosine Similarity for profile vectors).

3. Finally each circle is given a label by abstracting their common properties.

Comparing this algorithms performance (accuracy and speed) against Link- Community [6], Low-Rank Embedding [5], and Probabilistic Model [1] Base on the dataset they used [4], they achieve 1-BER score of 0.76 (26.7% and 20.6% higher than Link Community and Low-Rank Embedding), and F-measure of 0.52 (205.9% and 188.9% higher than Link Community and Low-Rank Embedding). In terms of speed, they outperform Probabilistic Model [1] by approximately 10 magnitude (1000 vertices in 1,000 seconds vs 10,000 seconds in Probabilistic Model), but slightly higher than Link-Community and Low-Rank Embedding.

4 Data Collection and Processing

The goal of our project is to use network analysis technique to group a person's contacts into different circles or lists, so it is important for us to find a social network dataset that includes ground truth circles for comparison purposes. The dataset we used for this paper is an undirected Facebook graph provided by Stanford Large Network Dataset Collection [1]. This dataset provides ground truth of 10 person's social network (Ego Networks) with their circles and individual profiles. To name some of the ego network, they are **{0, 107, 686, 1684, 3980, and etc}**. Table 1 shows the Dataset Statistics (compute using SNAP [4]).

Our algorithm and approach (Section 6) required the data to be split into two parts, network topology and node's profiles (node's features) for processing. The first part is, we partition the graph into 10 Ego networks and

each ego network contains n number of circles. In the second part, we collect all node's features in an adjacent matrices for our next part of algorithm which incorporate similarity measure to further improve the accuracy.

Table 1: Dataset Statistics

Ego Networks	10
Total Circles	193
Total Nodes	4039
Undirected Edges	88,234
Zero Deg Nodes	0
Max WCC Size	1 (4039 nodes)
Max SCC Size	1 (4039 nodes)
Total Triads	1,612,010

5 Dataset Analysis

5.1 Graph Property

Our approach of constructing circles uses individuals in an ego net who we consider leaders and from circles using these leaders friends. The power low degree distribution of one ego network shows that several individuals are as we expect the "popular ones" who we can use to build social circles. Please see Figure 1 and Figure 2

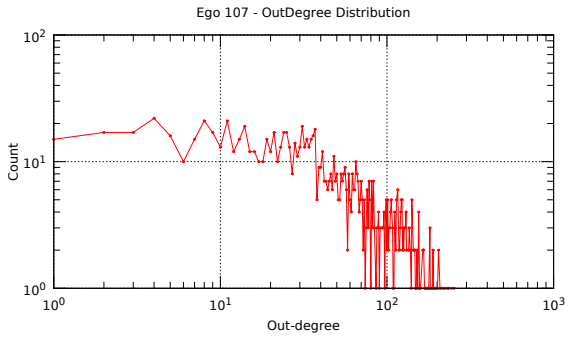

Figure 1: Ego 107 log-log Out-Degree Distribution Plot

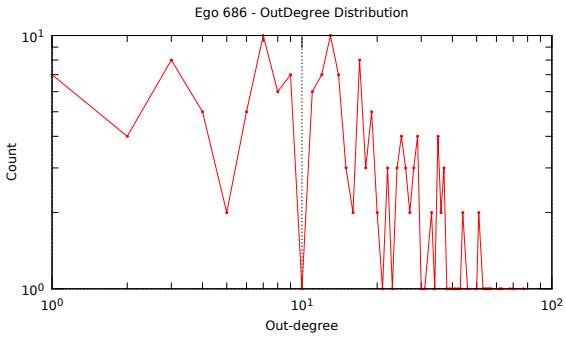

Figure 2: Ego 686 log-log Out-Degree Distribution Plot

6 Algorithm and Approach

Using the ground truth circles we extracted the leader of each community by using the highest score of the product, $d_u \cdot c_u$ where d_u is the degree density of node u within the circle and c_u is the clustering coefficient of node u using node u friends that belong to the relevant circle. using the leaders we formed new circles using the leaders friends and compared our result to published results. Our analysis suggests that forming circles using leader friends produces results comparable to published results[1][5]. Accordingly, our approach to form social circles in ego network is composed of

1. Define a method to extract leaders within an ego network

2. Use leader friends to form social circles

6.1 Finding leaders in an ego network

To find leaders in ego network we considered the following parameters.

- b_u is the betweenness score of node u in the ego network.
- c_u is the clustering coefficient of node u in the ego network.
- d_u is the degree density of node u in the ego network.

- s_u is the textual similarity between node u and the ego using ego and node u feature vectors.

6.1.1 Detail explanation of parameters

- b_u The betweenness centrality is an indicator of a node's centrality in a network. A node with high centrality is an indication of the node importance and subsequently the node leadership within a community.

- c_u The the local clustering coefficient of node u, computed as follows. Let G be the sub graph of node u and node u friends then

$$c_u = \frac{2 \cdot E(G)}{|G|(|G|-1)} \tag{1}$$

  where $E(G)$ total number of edges in G and $|G|$ is the number of nodes in G. A high clustering score is a measure of node u importance and perhaps an indication of node u facilitating connectivity between his friends

- d_u The degree density of node u and is an indication of the popularity of node u in the ego network.

- s_u The measure of how similar node u to the ego and is not a discriminating attribute of node u importance within a community in the ego network or an indication of node u leadership attributes. Accordingly, our method to find network leaders is focused on the ego network topology. However, if we consider $s_u = \sigma(u, v)$ closeness measure between two nodes u, v this measure can be used in the algorithm to form the social circle using an identified leader and the leader fiends, more on that below.

6.2 Identifying Leaders

To identify the leaders in an ego network, we considered severals approaches, one of the approaches is supervised machine learning. Using nodes with highest score of $b_u \cdot c_u \cdot d_u$ as the leaders of the ground truth circles and the rest of nodes in the ego network as 0, we trained a logistic regression function and used it to identify the leaders in the ego network. For each node in the ego network we computed

$$\frac{1}{1 - \exp^{-X\theta}} \tag{2}$$

where θ is the trained weights and $X = \{1, b_u, c_u, d_u\}$. After analyzing the BER and F1 scores of the circles formed using the top 10 nodes, we found that the results are not satisfactory. In an attempt to understand the reasons for the poor result, we found that the data is not separable. Figure 2 shown plot on b_u, c_u, d_u.

We further used a quadratic kernel for the supervised learning model and the results did not improve. We think that the number of leaders identified relative to the number of nodes is too low and much more data is needed to make the supervised learning model effective.

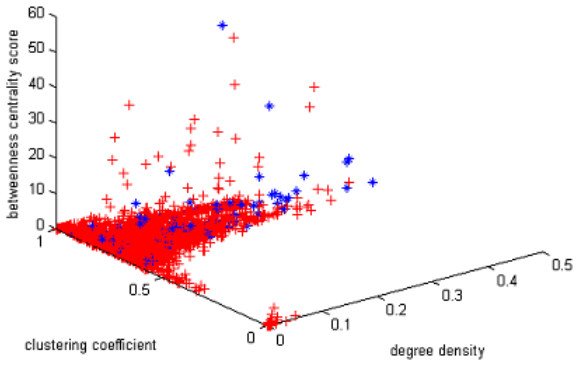

Figure 2

In view of the machine learning results, we simplified our approach of finding leaders in ego network by using the product of different combination of the three features defined above as

- $(b_u \cdot c_u \cdot d_u)$
- $(b_u \cdot c_u)$
- $(b_u \cdot d_u)$

- $(c_u \cdot d_u)$

The algorithm to find leaders is the following

1. $\forall u \in G$ compute u score using the parameters b_u, c_u, d_u. where G is the ego network
2. Sort scores by descending order $u_0, ..., u_n$ where u_0 is the highest score
3. Add u_0 to the set L of leaders, and let $C(u_0)$ be the set of nodes containing u_0 and u_0 friends.
4. for u_i , $i \in [1, n]$ and $Len(L) < 10$ add node u_i to L if $\frac{|C(u_i) \cap C(u_j)|}{min(|C(u_i)|, |C(u_j)|)} < \epsilon$, $\forall u_j \in L$

ϵ in the algorithm is a tuning parameter, based on our experiments, the best result is using $\epsilon = 60\%$. we decided to fix the number of circles to create from any ego network to 10, for future research the number of circles should vary based on certain heuristic.

Our best result achieved using the combination $b_u \cdot c_u$. results and scores are discussed in results section.

6.3 Use leader friends to form social circles

To form the social circle using the leader friends, our method uses conductance score to trim nodes at the edges of the community. The algorithm takes the set C_u that contains the leader u and the leader friends and order the nodes according to degree.

The pseudocode of the algorithm to form the circles is shown in Algorithm 1, the $C_u \setminus v$ is the set C_u less the node v. In the Conductance function, the definition of each parameters are

- *cutSet* is the set of nodes to be removed
- *v_cutSet* is the $\sum_{u \in cutSet} d_u$, where d_u is the degree of node u
- *keepSet* is the set of nodes to keep
- *v_keepSet* is $\sum_{u \in keepSet} d_u$
- *e_total* is the total edges in the sub graph containing the leader and the leader friends
- $E(G(cutSet))$ is number of edges in the sub graph containing the nodes in the *cutSet*

Algorithm 1 Forming Circles

```
1: set cutSet ← {}
2: set keepSet ← C_u
3: set e_total ← total # of edges in C_u
4: add node v with lowest degree to cutSet and remove v from keepSet
5: for node v ∈ keepSet do
6:     x ← Conductance(
           cutSet ∪ v, Volume(cutSet ∪ v),
           keepSet \ v, Volume(keepSet \ v),
           e_total)
7:     y ← Conductance(
           cutSet, Volume(cutSet),
           keepSet, Volume(keepSet),
           e_total)
8:     if x < y then
9:         cutSet ← cutSet + v
10:        keepSet ← keepSet \ v
11:    end if
12: end for

13: function CONDUCTANCE(cutSet, v_cutSet, keepSet, v_keepSet, e_total)
14:    subGraphCut ← E(G(cutSet))
15:    subGraphRemain ← E(G(keepSet))
16:    cut ← (e_total - SubGraphCut - SubGraphRemain)
17:    return cut ÷ min(v_cutSet, v_keepSet)
18: end function

19: function VOLUME(setA)
20:    vol ← 0
21:    for node u ∈ setA do
22:        vol ← vol+ out-degree of u
23:    end for
24:    return vol
25: end function
```

We considered using the measure $\sigma(u, v)$, which is the cosine measure of closeness of the node v textual attributes and the leader node u for adding node v to the *cutSet*, however we found that, using only conductance score is efficient and produced good result. We think it

is possible to improve the result by using the measure $\sigma(u, v)$ and stop the algorithm based on a second condition related to cosine measure of closeness between the node v considered and the leader node u.

7 Results and Analysis

We followed the above approach for Leader-Identification and Circle-Forming. To evaluate the results, we use two metrics: the Balanced Error Rate (BER) and the F1-Score.

BER for predicted circle C and ground-truth circle $\bar{C}$ is defined as

$$BER(C, \bar{C}) = \frac{1}{2}\left(\frac{|C \setminus \bar{C}|}{|C|} + \frac{|\neg C \setminus \neg \bar{C}|}{|\neg C|}\right) \quad (3)$$

where $\neg C$ is the complement of C in the vertices set V. The higher 1-BER is the better the prediction is. F1-Score is the harmonic mean of Precision and Recall,

$$F1 = \frac{2\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (4)$$

In order to clarify the difference between 1-BER and F1-scores, we rewrite them in terms of prediction True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN),

$$1 - BER = \frac{1}{2}\left(\frac{TP}{TP+FP} + \frac{TN}{FN+TN}\right) \quad (5)$$

$$F1 = \frac{2TP}{2TP+FP+FN} \quad (6)$$

We see that 1-BER takes account of True Negative (TN) whilst F1-score doesn't.

We have yet to define how to match a set of predicted circles $\mathcal{C}$ with the ground-truth set $\bar{\mathcal{C}}$. We find a one-to-one mapping $f : \mathcal{C} \to \bar{\mathcal{C}}$ such that f maximises

$$\sum_{C \in \mathcal{C}} \text{score}(C, f(C)) \quad (7)$$

where score is either 1-BER or F1. In the case that $|\mathcal{C}| >= |\bar{\mathcal{C}}|$, no two predicted circles will be mapped to the same ground-truth circle; whilst if $|\mathcal{C}| < |\bar{\mathcal{C}}|$ we no longer enforce this constraint.

We ran the algorithm in Section 6 and scored the predicted circles using 1-BER and F1-scores in Table 2.

Ego Id	1-BER	F1
0	0.63	0.32
107	0.60	0.49
1684	0.68	0.29
1912	0.69	0.38
3437	0.55	0.57
348	0.74	0.38
3980	0.63	0.28
414	0.83	0.49
686	0.76	0.47
698	0.77	0.35
Avg	0.70	0.40

Table 2: Evaluation metrics of the Leader-based Social Circles Identification Algorithm

As a comparison, [2] obtains 1-BER score 0.77, F1-Score 0.40 using Compressed Features (collapsed profile information by category); 1-BER score 0.84, F1-score 0.59 using Uncompressed Features (the original profile information of people). [5] obtains 1-BER score 0.76, F1-score 0.52.

Our approach works well and is largely comparable to [2] using Compressed Features or [5]. Our approach that first identifies a leader, then builds the circle of friends around the leader is therefore firmly validated.

Given that [2] draws on personal information attached to the network, and ours is based on the key topology indicators, default of running on larger data sets, we could interpret: the personal information attached to nodes and edges, at a coarse level, is already reflected in the topology of the network. Only when we look into the personal information at a finer granularity will we be able to improve the prediction.

7.1 Ground Truth Circles vs Predicted Circles

To visualize how the circles prediction perform, Figure 4 shown how our algorithm accurately

predicted (Yellow overlay on the left two graphs are correctly predicted nodes, Red nodes are ground truth, and Green nodes are predicted nodes) the circles (2 samples) for Ego-686 Network along with the "Heat Map" of the Clustering Coefficient (cu) and Betweenness Centrality (bu). While Figure 5 shows the weakness of our algorithm, which failed to predict most of the nodes in the circles (2 samples) for Ego-3980 Network.

Since our algorithm starts by picking leaders, the heat map highlights the algorithm strength and weakness. It is clear from the figures that for circles with low clustering and few nodes our algorithm fails to pick a leader to represent the community; accordingly our score for such circles is low while our score for dense circles that contain high centrality score nodes our algorithm performs much better.

Ego-686 Network is visually dense, has distinct connected components. Nodes in the core components have high betweenness centrality. Our algorithm successfully picks the nodes from these circles as leaders and forms the community using the leader friends. On the other hand, Ego-3980 Network is sparser. The ground truth circles lack high local clustering and centrality score nodes for our algorithm to pick as leaders, accordingly our algorithm produces low results and fails to predict such circles.

8 Conclusion and Future Work

Our approach to form social circles in an ego network using leaders was validated based on the comparable results to published methods that we achieved. It is possible to achieve better results, however, by improving on two weaknesses in our method

1. Allowing the number of circles to vary instead of fixing the number to 10 as our algorithm suggests. Our results were negatively affected due to this limitation. For example two of the ego networks in our data set contained less than 2 social circles while others contain more.

2. Incorporating person's information features to the circle forming algorithms to cut certain nodes and add nodes in the ego network that belong to the community but not friends with the leader found for this community.

We believe that since our approach is easily scalable, company like Facebook may find our approach useful in the sense that our method can be used to create initial circles to suggest to the ego as starting point after which the ego may refine to his/her liking.

9 Contribution

All team members {Amer Hammudi, Darren Koh, Jia Li} involved and contributed equally on all parts of this project. Followings are contribution of individual who involved more in depth on certain area.

- Amer Hammudi: Model and Algorithm
- Darren Koh: Data Processing/Analysis and Graph Plotting
- Jia Li: Literature research and Result Scoring

10 ACKNOWLEDGEMENTS

We would like to thank Himabindu (CS224W Head TA), Vikesh Khanna (CS224W TA), and Bruno Abrahao (Snap Group, Infolab @ Stanford) for their advice and input.

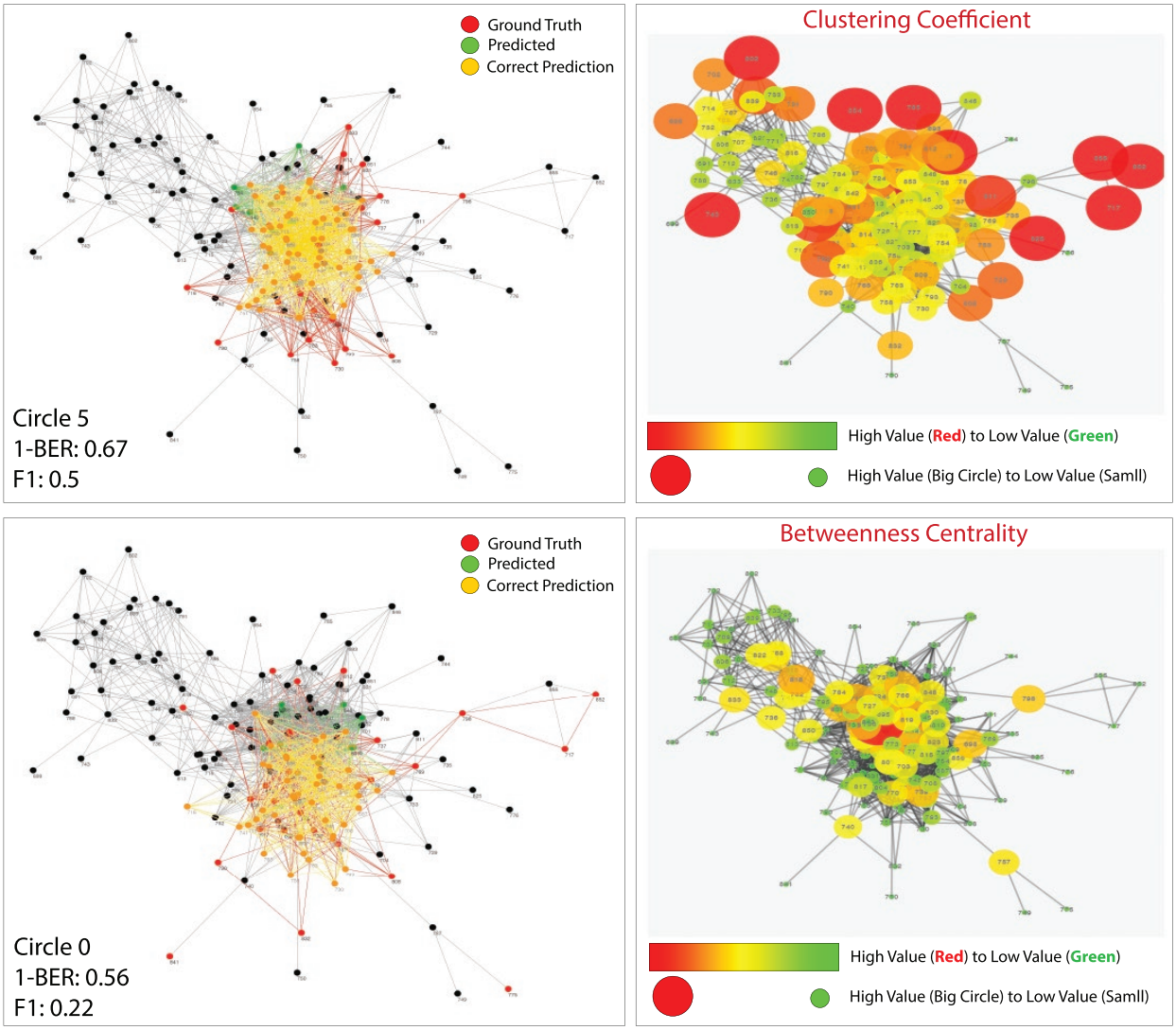

Figure 4: Ego 686, Left two graphs are Circle_5 and Circle_0. Right two graphs are network analysis (Clustering Coefficient and Betweenness Centrality "Heat Map") on the ego 686 graph

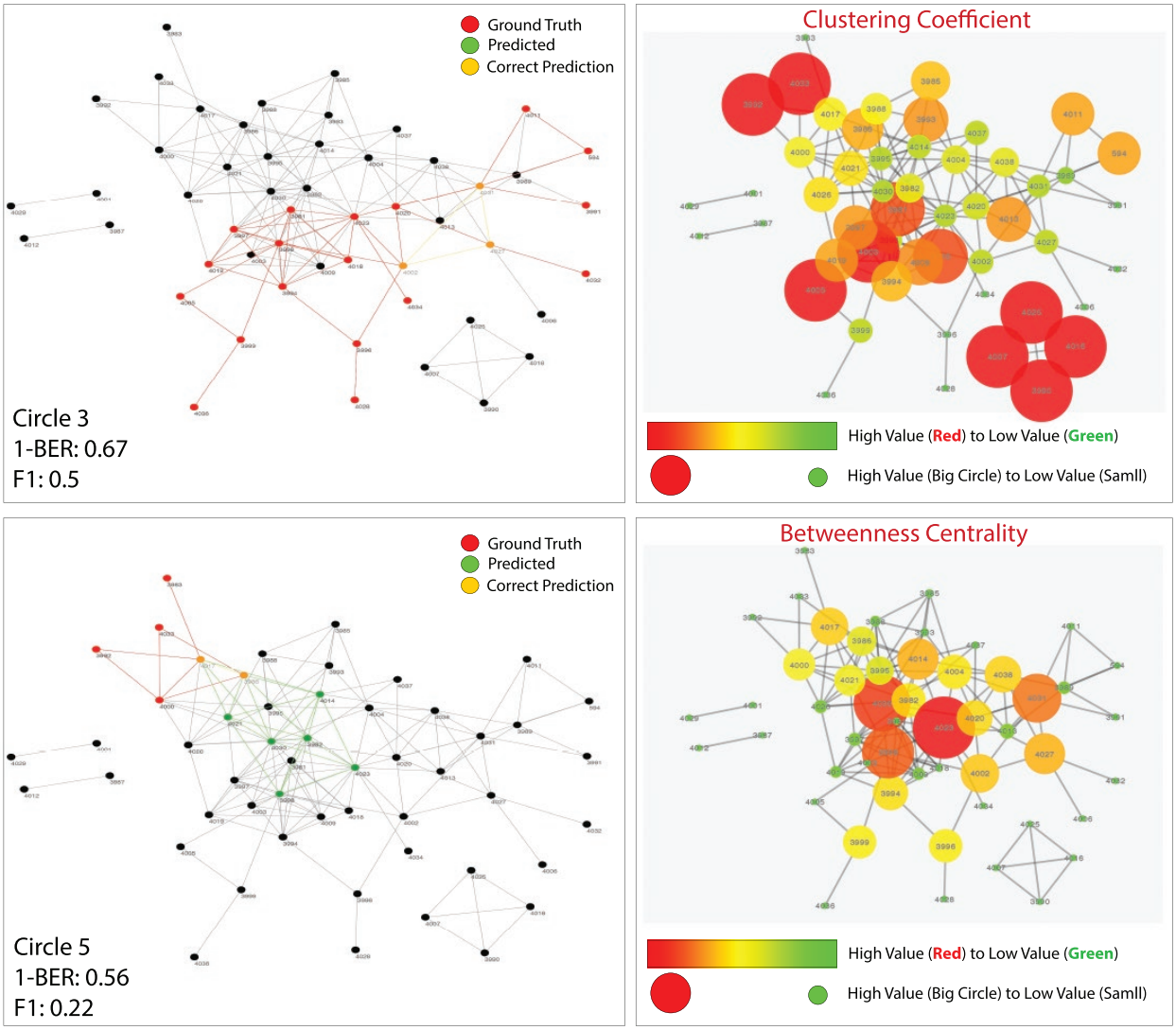

Figure 5: Ego 3980, Left two graphs are Circle_3 and Circle_5. Right two graphs are network analysis (Clustering Coefficient and Betweenness Centrality "Heat Map") on the ego 3980 graph

References

[1] Jure Leskovec and Andrej Krevl, "SNAP Datasets: Stanford Large Network Dataset Collection" Jun 2014

[2] Julian McAuley and Jure Leskovec, "Learning to discover social circles in ego networks", Advances in Neural Information Processing Systems 25 (NIPS 2012)

[3] Julian McAuley and Jure Leskovec, "Discovering Social Circles in Ego Networks", ACM Transactions on Knowledge Discovery from Data (TKDD), Vol 8, No 1, Feb 2012. doi: 10.1145/2556612

[4] Jure Leskovec and Rok Sosi, "Snap.py: SNAP for Python, a general purpose network analysis and graph mining tool in Python"

[5] Yu Wang and Lin Gao, "An Edge-based Clustering Algorithm to Detect Social Circles in Ego Networks", Journal of Computers, Vol 8, No 10 (2013), 2575-2582, Oct 2013. doi: 10.4304/jcp.8.10.2575-2582