

# Hierarchical Methods For Multi Partially Overlapped Social Network De-anonymization

Yucheng Zhang 517021910431

## Abstract

In multi social networks de-anonymization problem, different social networks often have very different user numbers. Structure-based algorithms often performs not well enough since they only make use of the topological information. In this paper, we first improve the two graph matching solver by taking each node's features into consideration. Then we come up with the hierarchical matching methods for multi social network de-anonymization. Social networks with large amounts of users will be treated as central part, and the matching for small social networks will depend on the central part. Experiments on generated graphs show the advantage of our algorithm.

## 1 Introduction

With the development of technology, we live in an era with the explosion of information. When the information of social network users is released, only removing ID(such as user's real name or ID number) is not enough. De-anonymization algorithm can get user's information by matching the anonymized network with a sanitized one. However, in the situation of multi social network de-anonymization, directly using structure-based two-graph matching solver for all pairs of multi-social networks have several practical problems:

1. In real social network platforms, there may not exist one platform that can cover all the users.

2. Different social networks are partially overlapped, which is to say different social networks often have very different user numbers and the set of common users is unsure.
3. Social network is not only about the connections between users. Each user has its own inherent characteristics (e.g. preference for a certain topic; behavioral habits). Only make use of the topological structure(e.g. adjacent matrix) to do multi social network de-anonymization is a waste of known information and will lead to low accuracy.

Based on the above consideration, we first improve the two graph matching solver by attaching each node with a feature vector. These feature vectors represents each node’s inherent characteristics rather than topological structure. In real world, these features include things like each user’s preference and behavioral habits. For example, whether or not the user is interested in a specific sport, whether or not the user prefers a specific singer and whether the user is inclined to use the APP during a certain time of the day? Features like these are very common for a social network so that they are available. And we believe that the users’ behavior is consistent to some extent. By this we mean for the same user in two different social networks, the two feature vectors collected from these two networks have a large possibility to be similar.

Then, we come up with the hierarchical multi social networks de-anonymization methods. The top two or three networks will chosen as center. We hope that the union of nodes from central graphs will cover all potential nodes with a high probability. The graphs in central part will match one another. For the rest graphs, we call them peripheral graphs. Peripheral graphs will first match all central graphs, then match other peripheral graphs via central graphs. Experiments on generated data show a good performance of our methods.

Prior work [1] is the first to take the partially overlapped situation into consideration, but it:

1. Only considers two social networks. Our work is about multi social networks.
2. Does not take nodes’ features into consideration.
3. Just proof the conditions for correct matching, and does not give a specific algorithm.

Prior work[4][5] also focus on the setting of partially overlapped social networks, but they do not take nodes' features into consideration neither and they are seeded de-anonymization while our work is without seeds.

For the following sections, we first set up the model of feature-aware de-anonymization and analyse the conditions for correct matching in section 2; in section 3 we will show our hierarchical multi graphs matching methods; and experiments are in section 4. In section 5 there is the conclusion for this paper and acknowledgement is in section 6.

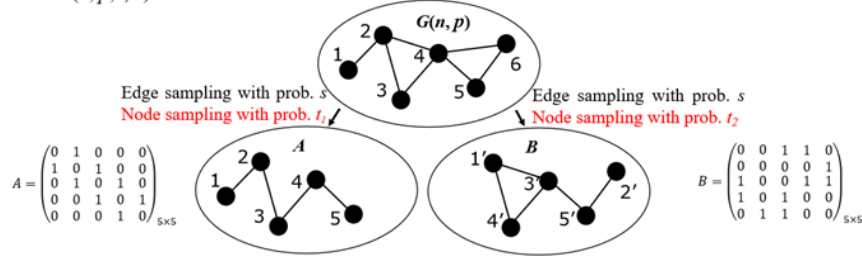
## 2 Feature-aware Two Graphs Matching

### 2.1 Problem Statement

We suppose the real underlying social network is the Erdos-Renyi Graph  $G(n, p)$  with  $n$  representing the total number of users(nodes) and  $p$  representing the possibility of an edge existing between two random nodes. We posit the node sampling probability is  $t_i$  for the  $i$ th graph and the edge sampling probability is  $s$  for all graphs.

We note the sanitized network as  $G_1(V_1, E_1, A)$  with node set  $V_1$ , edge set  $E_1$  and adjacent matrix  $A \in \mathbb{R}^{n_1 \times n_1} (|V_1| = n_1 \text{ and } E(n_1) = nt_1)$ ; the auxiliary network as  $G_2$  with with node set  $V_1$ , edge set  $E_1$  and adjacent matrix  $B \in \mathbb{R}^{n_2 \times n_2} (|V_2| = n_2 \text{ and } E(n_2) = nt_2)$ . The  $i$ th node in  $A$  has a feature vector  $f_i^A$  for  $i$  in  $\{1, 2, \dots, n_1\}$ , while the  $j$ th node in  $B$  has a feature vector  $f_j^B$  for  $j$  in  $\{1, 2, \dots, n_2\}$ . All feature vectors have  $d$  dimensions. Besides we get a distance matrix  $D$ , where  $D_{ij} = \|f_i^A - f_j^B\|_2^2$  depicts the distance between  $i$ th nodes's feature vector in  $A$  and  $j$ th node's feature vector in  $B$ .

The  $G(n, p; t, s)$  network model:



We posit the matching matrix is  $P \in \{0, 1\}^{n_1 \times n_2}$ , where  $P_{ij} = 1$  means the  $i$ th node in  $G_1$  is the  $j$ th node in  $G_2$ . And we assume the ground truth matching

matrix is  $P^*$ . The goal is to find a certain  $P$  to match nodes in  $V_2$  to nodes in  $V_1$  with the highest accuracy:

$$\min_{P \in \{0,1\}^{n_1 \times n_2}} \|P - P^*\|_2^2$$

## 2.2 The Model for Feature Vectors

### 2.2.1 Binomial Model

In binomial model, the  $k$ th dimension of feature vectors obeys a binomial distribution:  $P(f_{ik}^A = 1) = P(f_{jk}^B = 1) = p_k$  and  $P(f_{ik}^A = 0) = P(f_{jk}^B = 0) = 1 - p_k$ . This is like we ask each user  $d$  "whether or not" questions and the user answers "yes" or "no" honestly.

If  $(i, j)$  is a correct match, we suppose that  $f_i^A = f_j^B$  so that  $D_{ij} = 0$ .

If  $(i, j)$  is a wrong match, then  $P(f_{ik}^A = f_{jk}^B) = p_k^2 + (1 - p_k)^2$  and  $P(f_{ik}^A \neq f_{jk}^B) = 2p_k(1 - p_k)$ . For the convenience of analysis, we posit that  $p_i = \bar{p}$  for  $i \in \{1, 2, \dots, k\}$ . So that  $D_{ij}$  obeys a binomial distribution  $B(d, 2\bar{p}(1 - \bar{p}))$ .

### 2.2.2 Gaussian Model

In Gaussian model, the  $k$ th dimension of feature vectors obeys a Gaussian distribution  $N(\mu_k, \sigma_k^2)$ .

If  $(i, j)$  is a wrong match,  $f_{ik}^A - f_{jk}^B$  obeys a Gaussian distribution  $N(0, 2\sigma_k^2)$ . If  $(i, j)$  is a correct match, we suppose that  $f_{ik}^A - f_{jk}^B$  obeys a Gaussian distribution  $N(0, 2\sigma_k'^2)$  with  $\sigma_k' \ll \sigma_k$ . For the convenience of analysis, we posit that  $\sigma_i = \sigma_{big}$  for  $i \in \{1, 2, \dots, k\}$  and  $\sigma_i' = \sigma_{small}$  for  $i \in \{1, 2, \dots, k\}$  with  $\sigma_{big} \gg \sigma_{small}$ .

So if  $(i, j)$  is a wrong match,  $D_{ij}$  obeys a generalized  $\chi^2(d)$  distribution, which means  $E(D_{ij}) = d\sigma_{big}^2$  and  $Var(D_{ij}) = 2d\sigma_{big}^4$ ; if  $(i, j)$  is a correct match,  $D_{ij}$  obeys a generalized  $\chi^2(d)$  distribution, which means  $E(D_{ij}) = d\sigma_{small}^2$  and  $Var(D_{ij}) = 2d\sigma_{small}^4$ .

### 2.3 Algorithm

The two graph matching problem can be approximated in the Koopmans-Beckmann's Quadratic Assignment Problem form in the first step:

$$\begin{aligned} P = & \arg \min_{P \in \{0,1\}^{n_1 \times n_2}} \|A - PBP^T\|_F^2 + \lambda \|P \circ D\|_F^2 \\ \text{s.t. } & P\mathbf{1}_{n_2} \leq \mathbf{1}_{n_1} \text{ and } P^T\mathbf{1}_{n_1} = \mathbf{1}_{n_2} \end{aligned}$$

where  $\lambda$  is a parameter controlling the weight of feature vector matching loss and  $\circ$  is the Hadamard product for two matrices. Although the two graphs may be partially overlapped, in the first step we match all nodes in the smaller graph to the other graph.

To remove the wrongly-matched nodes in the first step, we add one more step:

$$\forall i, j, \text{ if } P_{ij} = 1 \text{ and } D_{ij} > \text{threshold, set } P_{ij} = 0.$$

Then we use gradient descent method to solve it. Through the process of gradient descent, we may get continuous-value matching matrix. And the sum of a certain column may not equal to 1, which may violate the requirement for assignment problem. So after each step of gradient descent, we use Hungarian algorithm to get the discrete matching matrix.

---

**Algorithm 1:** Gradient-based Two-graph Matching

---

**input** : adjacent matrix  $A_{m \times m}$  and  $B_{n \times n}$ , feature distance matrix  $D_{m \times n}$ .  
**output** : identify matrix  $P^*$ .

```

1  $P \leftarrow \frac{1_{m \times n}}{\min\{m, n\}}$ ;  $\lambda \leftarrow 1$ ;  $\alpha \leftarrow 0.1$ ;  $\beta \leftarrow 0.01$ ;  $threshold \leftarrow 0$ ;
2  $loss = \|A - PBP^T\|_F^2 + \lambda \|P \circ D\|_F^2$ ;
3 while  $loss \geq \beta$  do
4    $G \leftarrow 4APBP^T - 2\lambda \|P \circ D\|_F \circ D$ ;
5    $Q = \arg \max_{Q \in \{0,1\}^{m \times n}} tr(Q^T G)$ ;
6    $P = \alpha P + (1 - \alpha)Q$ ;
7    $loss = \|A - PBP^T\|_F^2 + \lambda \|P \circ D\|_F^2$ ;
8  $P^* = \arg \max_{Q \in \{0,1\}^{m \times n}} tr(Q^T P)$ ;
9 For all  $P_{ij}^*$ , if  $D_{ij} > threshold$ , set  $P_{ij}^* = 0$ ;
10 return  $P^*$ ;

```

---

## 2.4 Conditions for Correct Matching

For the following part, we give an theoretical analysis of the conditions for correct matching if we follow the above loss function. Without loss of generality, we posit that  $t_1 > t_2$  and  $n_1 > n_2$ .

**Theorem 1.** *We suppose that  $\bar{p} = 0.5, s = 1$  and  $d \sim \theta(1)$ , then if  $p, t_1$  and  $t_2$  satisfy the following condition, we will get a correct matching.*

$$\frac{pt_1}{18} \max\left\{\frac{2}{1-t_1}, nt_1t_2\right\} \sim \ln n + \omega(1)$$

Before the proof, we first give some definitions. Like in [2], we divide the nodes into three parts, correctly matching nodes  $V_c$ , mismatched nodes  $V_m$  and unmatched nodes  $V_u$ . And  $E(V_u) = n(t_1 - t_2)$ .

The node pairs can be parted into three categories:

1.  $E_{c,k}$  is the correctly matched pairs where both nodes are correctly matched.
2.  $E_{m,k}$  is the mismatched node pairs where one of them is a mismatched node and the other is either mismatched or correctly matched.

3.  $E_u$  is the node pairs where at least one node of whom is unmatched.

And from the perspective of mathematical expectation, we have:

$$\begin{aligned} |E_{c,k}| &= \binom{nt_2-k}{2} \\ |E_{m,k}| &= \binom{k}{2} + \binom{k}{1} \binom{nt_2-k}{1} \\ |E_u| &= \binom{n(t_1-t_2)}{2} + \binom{n(t_1-t_2)}{1} \binom{nt_2}{2} \end{aligned}$$

We note  $\Delta_k^1 = \|A - PBP^T\|_F^2$  and  $\Delta_k^2 = \lambda \|P \circ D\|_F^2$  as the loss while there are exactly  $k$  wrongly matched nodes for  $k \in \{k_{min}, k_{min} + 1, \dots, [nt_2]\}$  where  $k_{min} = n(t_1 - t_2)$ . And we note that  $\Delta_k = \Delta_k^1 + \Delta_k^2$ .

#### 2.4.1 The Evaluation of $P(\Delta_k \leq \Delta_{k_{min}})$

For a given  $k$ , we first evaluate the possibility  $P(\Delta_k \leq \Delta_{k_{min}})$ .

As shown in [2],  $\Delta_k^1 - \Delta_{k_{min}}^1 = X_k^2 - X_k^1$  where  $X_k^2$  is the number of wrongly matched edges in and  $X_k^1$  is the number of edges is correctly matched but only sampled in either  $G_1$  or  $G_2$ .  $E(X_k^2) = |E_{m,k} - E_{m,k_{min}}|2ps(1-s)$ ,  $E(X_k^1) = |E_{m,k} - E_{m,k_{min}}|2ps$ .

We note that  $\Delta_k^2 - \Delta_{k_{min}}^2 = X_k^3$ .  $E(X_k^3) = 2d(k - k_{min})\bar{p}(1 - \bar{p})$ .

**Lemma 1.**  $P(X_k^2 - X_k^1 + X_k^3 \leq 0) \leq P(X_k^2 \leq b) + P(X_k^1 \geq a) + P(X_k^3 \leq c)$  for any positive constants  $a, b, c$  satisfying  $b - a + c = 0$ .

*Proof.*

$$\begin{aligned} &P(X_k^2 - X_k^1 + X_k^3 \leq 0) \\ &= P(X_k^1 \geq a \geq X_k^2 + X_k^3) + P(X_k^2 + X_k^3 \leq X_k^1 \leq a) + P(X_k^1 \geq X_k^2 + X_k^3 \geq a) \\ &\leq 2P(X_k^1 \geq a \geq X_k^2 + X_k^3) + P(X_k^1 \leq a \cap X_k^2 + X_k^3 \leq a) + P(X_k^1 \geq a \cap X_k^2 + X_k^3 \geq a) \\ &= P(X_k^1 \geq a) + p(X_k^2 + X_k^3 \leq a) \\ &= P(X_k^1 \geq a) + p(X_k^2 + X_k^3 \leq b + c) \end{aligned}$$

For the second term:

$$\begin{aligned} &p(X_k^2 + X_k^3 \leq b + c) \\ &\leq P(X_k^2 \leq b \cap X_k^3 \leq c) + P(X_k^2 \geq b \cap X_k^3 \leq c \cap X_k^2 + X_k^3 \leq b + c) + P(X_k^2 \leq b \cap X_k^3 \geq c \cap X_k^2 + X_k^3 \leq b + c) \\ &\leq 2P(X_k^2 \leq b \cap X_k^3 \leq c) + P(X_k^2 \geq b \cap X_k^3 \leq c) + P(X_k^2 \leq b \cap X_k^3 \geq c) \\ &\leq P(X_k^2 \leq b) + P(X_k^3 \leq c) \end{aligned}$$

Combine the above two parts together, we finish the proof.  $\square$

**Lemma 2.** For a binomial random variable  $X \in B(n, p)$  and  $\mu = np$ , then:

$$P(X > \mu + r) \leq \exp\left\{-\frac{r^2}{2(\mu + \frac{r}{3})}\right\}, r \geq 0$$

$$P(X < \mu - r) \leq \exp\left\{-\frac{r^2}{2\mu}\right\}, r \geq 0$$

*Proof.* This is a basic property of Chernoff bound for binomial random variables[3]. Here we leave out the detailed proof.  $\square$

We try to find  $a, b, c, r_1, r_2, r_3$  satisfying:

$$\begin{aligned} b - a + c &= 0 \\ a &= \mu_1 + r_1 \\ b &= \mu_2 - r_2 \\ c &= \mu_3 - r_3 \\ r_1 &= r_2 = r_3 \end{aligned}$$

By solving the above equation set, we get  $r_1 = r_2 = r_3 = \frac{-\mu_1 + \mu_2 + \mu_3}{3}$ .

Now that we get:

$$\begin{aligned} &P(X_k^1 \geq \mu_1 + r_1) + P(X_k^2 \leq \mu_2 - r_2) + P(X_k^3 \leq \mu_3 - r_3) \\ &\leq \exp\left\{-\frac{r_1^2}{2(\mu_1 + \frac{r_1}{3})}\right\} + \exp\left\{-\frac{r_2^2}{2\mu_2}\right\} + \exp\left\{-\frac{r_3^2}{2\mu_3}\right\} \\ &\leq 3\exp\left\{-\frac{r_1^2}{2(\mu_1 + \mu_2 + \mu_3)}\right\} \\ &= 3\exp\left\{-\frac{(-\mu_1 + \mu_2 + \mu_3)^2}{18(\mu_1 + \mu_2 + \mu_3)}\right\} \end{aligned}$$

Remember that

$$\begin{aligned} \mu_1 &= ((nt_2 - \frac{1}{2})(k - k_{min}) - \frac{k^2}{2} + \frac{k_{min}^2}{2})(2ps) \\ \mu_2 &= ((nt_2 - \frac{1}{2})(k - k_{min}) - \frac{k^2}{2} + \frac{k_{min}^2}{2})2ps(1 - s) \\ \mu_3 &= 2(k - k_{min})d\bar{p}(1 - \bar{p}) \end{aligned}$$

And  $E(k_{min}) = nt_2(1 - t_1)$ .

We further get

$$\begin{aligned} &P(X_k^1 \geq \mu_1 + r_1) + P(X_k^2 \leq \mu_2 - r_2) + P(X_k^3 \leq \mu_3 - r_3) \\ &\leq 3\exp\left\{-\frac{(k - nt_2 + nt_1t_2)}{18} \frac{[(nt_2 + nt_1t_2 - k - 1)(-ps^2) + 2d\bar{p}(1 - \bar{p})]^2}{(nt_2 + nt_1t_2 - k - 1)ps(2 - s) + 2d\bar{p}(1 - \bar{p})}\right\} \end{aligned}$$



### 2.4.2 Bound the Error For ALL $k$

We suppose that  $\bar{p} = 0.5, s = 1$  and  $d \sim \theta(1)$ , then

$$\begin{aligned}
& E(S) \\
& \leq 3 \sum_{k=k_{min}+1}^{nt_2} n^k \exp\left\{-\frac{(k-nt_2+nt_1t_2)(nt_2+nt_1t_2-k-1)p}{18}\right\} \\
& = 3 \sum_{k=k_{min}+1}^{nt_2} \exp\left\{k \ln n - \frac{(k-nt_2+nt_1t_2)(nt_2+nt_1t_2-k-1)p}{18}\right\} \\
& \leq 3 \sum_{k=k_{min}+1}^{nt_2} \exp\left\{k(\ln n - \frac{(k-nt_2+nt_1t_2)(nt_2+nt_1t_2-k-1)p}{18k})\right\} \\
& \leq 3 \sum_{k=k_{min}+1}^{nt_2} \exp\left\{k(\ln n - \frac{pt_1}{18} \max\{\frac{2}{1-t_1}, nt_1t_2\})\right\}
\end{aligned}$$

So if  $p, t_1$  and  $t_2$  satisfy the following condition, we will get a correct matching.

$$\frac{pt_1}{18} \max\{\frac{2}{1-t_1}, nt_1t_2\} \sim \ln n + \omega(1)$$

## 3 Hierarchical Multi Social Network De-anonymization

We suppose the real underlying social network is the Erdos-Renyi Graph  $G(n, p)$  with  $n$  representing the total number of users(nodes) and  $p$  representing the possibility of an edge existing between two random nodes. There are  $N$  graphs needs to be matched. We note them as  $G_i(V_i, E_i, A_i)$  with node set  $V_i$ , edge set  $E_i$  and adjacent matrix  $A_i \in \mathbb{R}^{|V_i| \times |V_i|}$ . Each node in every graph has a  $d$  dimension feature vector. For two graphs  $G_i$  and  $G_j$ , we use  $P^{ij}$  to express its matching matrix.

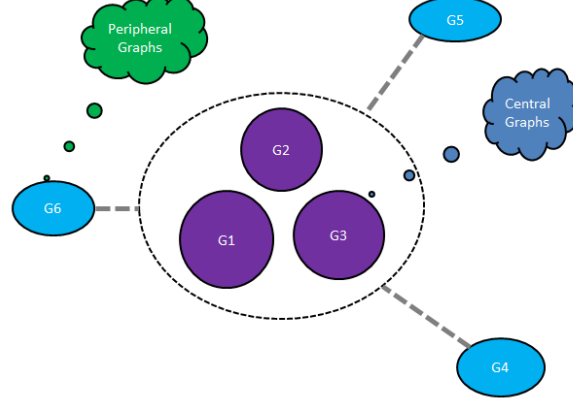
Now we partition total graphs into two parts:

### 1. central graphs

This part contains two or three graphs which has the most nodes. Only graphs whose node sampling possibility is larger than a certain threshold can be selected into central part. We hope the union of nodes in central graphs will cover as more nodes as possible while using the minimum number of graphs. According to possibility, if the threshold is 80%, two graphs will cover about 96% of total nodes; if the threshold is 70%, three graphs will cover about 97.3% of total nodes.

2. peripheral graphs

The rest of graphs will be put into peripheral part.



When coming to the matching part, we partition it into three phases:

1. center-center matching

We first match all the graphs in central part using the two-graph matching solver mentioned in the front part. We will give a tolerance on consistency, which is to say, if three graphs  $G_1, G_2, G_3$  are selected into central parts. We do not require exactly  $P^{12} = P^{13}P^{23T}$  for the three-graph tuple.

2. center-peripheral matching

When matching graphs from peripheral part, we first map every one of them to every graph in the central part respectively. By doing so, we get all the center-peripheral matching pairs. For a peripheral graph  $G_i$ , we will get  $P^{1i}, P^{2i}, P^{3i}$  in this phase.

3. peripheral-peripheral matching

When it comes to peripheral-peripheral matching, rather than directly mapping them, we use all graphs in the center part as intermediates. That is to say, for two peripheral graphs  $G_i$  and  $G_j$ , we have  $P_{ij} = \sum_{\forall G_k \in \text{center}} P_{ik}^T P_{kj}$ . This is kind of like a 'voting' process: we let central graphs decide how to match peripheral graphs. Then we use Hungarian algorithm to turn  $P_{ij}$  into an assignment matrix.

---

**Algorithm 2:** Hierarchical Multi-graph Matching

---

**input** : adjacent matrices of all graphs; feature distance matrices  
of all center-center and center-peripheral graph pairs.  
**output** : matchings for all graph pairs.

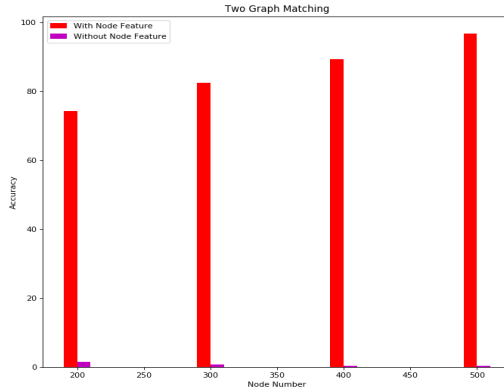
- 1 matching all center-center graph pairs using feature-aware  
two-graph matching solver;
- 2 matching all center-peripheral graph pairs using feature-aware  
two-graph matching solver;
- 3 matching all peripheral-peripheral graph pairs by:
- 4 (1) for peripheral graph  $G_a$  and  $G_b$ ,  $P^{ab} = \sum_{\forall G_k \in \text{center}} (P^{ka})^T P^{kb}$ ;
- 5 (2)  $P^{ab} = \arg \max_{Q \in \{0,1\}^{m \times n}} \text{tr}(Q^T P^{ab})$ ;
- 6 (3) For all  $P_{ij}^{ab}$ , if  $D_{ij}^{ab} > \text{threshold}$ , set  $P_{ij}^{ab} = 0$ ;

---

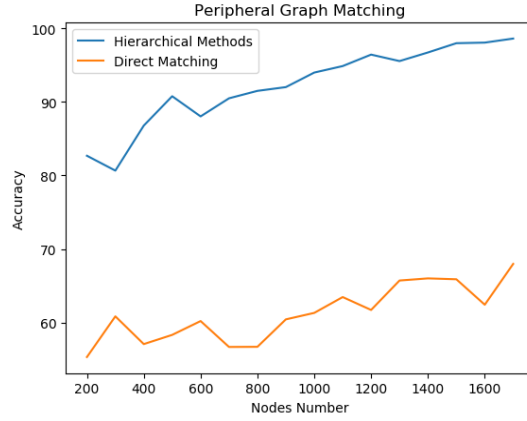
## 4 Experiments

The following experiments are conducted on generated data. For the following experiments, we set  $d = 7, s = 1, p = 0.05$ .

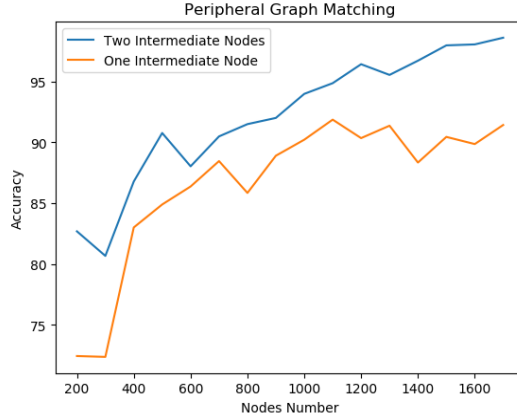
1. We first compare the feature-aware two-graph solver with the original method. For this experiment  $t_1 = 0.9$  and  $t_2 = 0.2$ . From the result we can see that without node features the accuracy is almost 0, while feature-aware method can achieve nearly 100% if nodes number is larger than a threshold.



2. There are two graphs in central part both with node sampling probability 0.9. There are many graphs in the peripheral part with node sampling probability 0.2. If directly matching peripheral graphs the accuracy is around 60%, while our hierarchical method can achieve an accuracy of over 90%. The accuracy shows the advantage of our methods compared with directly matching two nodes.

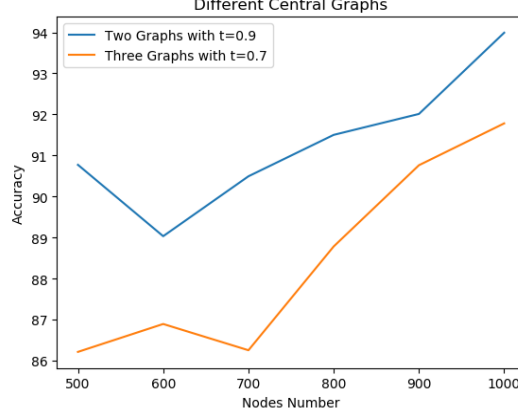


3. In this experiment we compare one central graph with two central graphs. The central graph has node sampling probability 0.9 while peripheral graphs are 0.2. Obviously with one more central graph, the performance is better. This is because one central graphs will cover more potential nodes.



4. In this experiment we compare two central graphs with node sampling probability 0.9 and three central graphs with node sampling probability 0.7.

Result shows two central graphs with node sampling probability 0.9 is better. This is because (1)it covers more potential nodes( $99\% \geq 97.3\%$ );(2)larger node sampling probability means the matching is more stable and more reliable for every central-peripheral matching.



## 5 Conclusions and Future Work

In this paper, our main contribution includes:

1. We come up with the feature-aware solver for partially overlapped graph matching problem, and we analyze the conditions for correct matching.
2. We come up with the hierarchical methods for multi partially overlapped social network de-anonymization problems. Experiments show the advantage of our methods.

Future work may includes:

1. Online Matching

Future work may take the change of each platform's user number into consideration. There may be new users for social networks. Small platforms may grow larger while large platform may decline.

2. All Platforms Too Small

In this work, we have an assumption that the top 2 or 3 platforms will cover most of the latent nodes. But what if the biggest platform only contains less than 50% of total nodes?

## 6 Acknowledgement

We are grateful to Professor Luoyi Fu who provides lots of help for us. Without her enthusiastic guidance, this work can not be done. We also want to show our gratitude to TA Jiapeng Zhang for his patient assistance.

## References

- [1] Ehsan Kazemi, Lyudmila Yartseva, Matthias Grossglauser: When can two unlabeled networks be aligned under partial overlap? Allerton 2015: 33-42
- [2] Zhang, Jiapeng; Fu, Luoyi; Wang, Xinbing; Lu, Songwu (2020), De-anonymization of social networks: the power of collectiveness, to appear in IEEE INFOCOM'20.
- [3] Pedarsani, Pedram; Grossglauser, Matthias (2011), On the Privacy of Anonymized Networks, Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'11), pp. 1235-1243.
- [4] Zhongzhao Hu, Luoyi Fu, Xiaoying Gan: De-anonymize social network under partial overlap. ACM TUR-C 2019: 16:1-16:5
- [5] Carla-Fabiana Chiasserini, Michele Garetto, Emilio Leonardi: De-anonymizing Clustered Social Networks by Percolation Graph Matching. ACM Trans. Knowl. Discov. Data 12(2): 21:1-21:39 (2018)