Social Network De-anonymization with Overlapping Communities: Analysis, Algorithm and Experiments

Xinyu Wu,[1] Zhongzhao Hu,[1] Xinzhe Fu,[1] Luoyi Fu,[1] Xinbing Wang,[1] Songwu Lu.[2]

[1]Shanghai Jiao Tong University, China
[2]University of California, Los Angeles

April 18, 2018

Contents

Outline

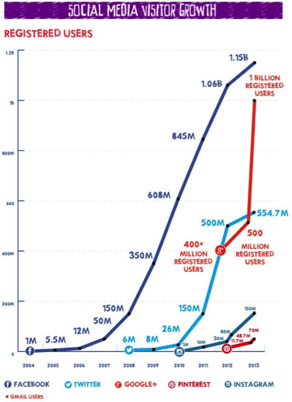
1. Introduction
2. Problem Formulation
3. Analytical Aspect
4. Algorithmic Aspect
5. Experimental Aspect
6. Conclusion

Social Networks

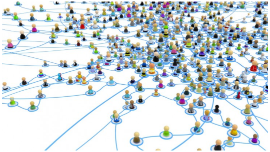
- We are in many social networks nowadays.

Booming Social Networks

- Social networks explode these days.



More Social Networks



Larger Social Networks

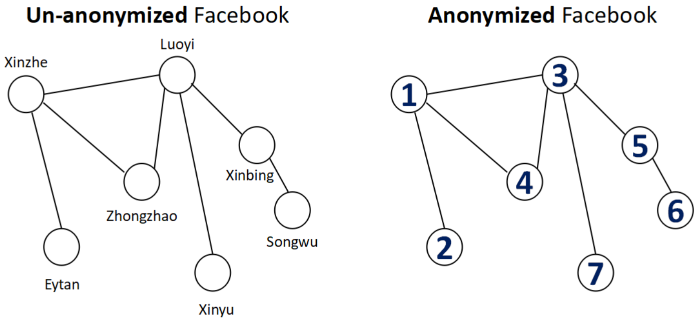
Privacy Exposed to Public

- Private information becomes more often released to public.

- It gives opportunities for adversaries to identify users.
- How to protect ?

Anonymize Yourself !

- **Anonymization :** Removing Personal Identifiers.
  - IDs, Names, Records, Institutes...

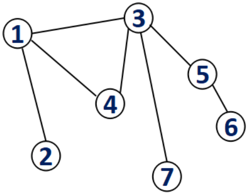


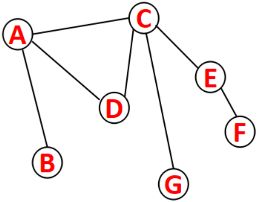
- Is it safe ?

A Toy Example

- IF : Another **identical** un-anonymized networks ?

Anonymized Facebook : **Un-Anonymized** Linkedin :



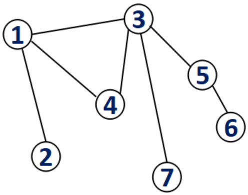


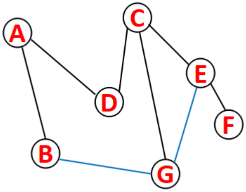
- It is trivial to identify all users in Facebook.
- It is **NOT** safe.

A Toy Example

- Social networks on different platforms are often different.
  - Friends may/may not be connected in social networks.

Anonymized Facebook : **Un-Anonymized** Linkedin :

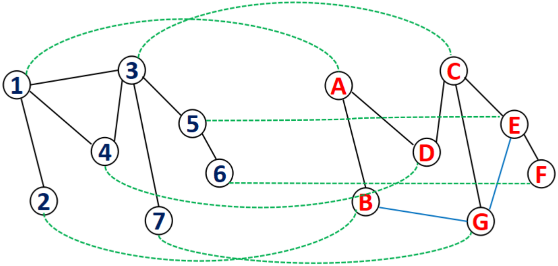




- Can we identify users in Facebook now ?

Social Network De-anonymization

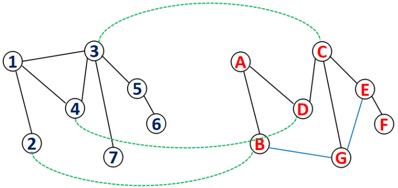
- **De-anonymization** is a way to identify users in an anonymized network by another un-anonymized network.
- We need to find a mapping from un-anonymized networks to anonymized networks.



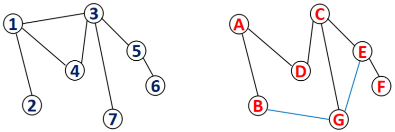
- $1 \leftrightarrow A$
- $2 \leftrightarrow B$
- $3 \leftrightarrow C$
- $4 \leftrightarrow D$
- $5 \leftrightarrow E$
- $6 \leftrightarrow F$
- $7 \leftrightarrow G$

Different Versions of De-anonymization

- **Seeded** De-anonymization : There are pre-mappings.

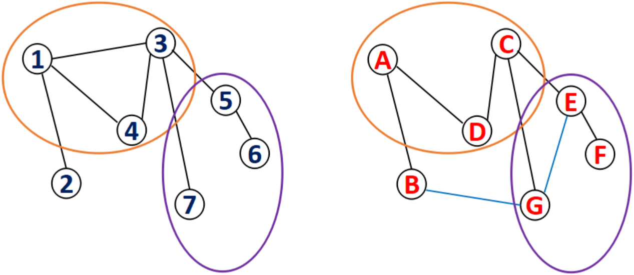


- **Seedless** De-anonymization : No pre-mappings.



Different Versions of De-anonymization

- De-anonymization with **Communities** :
  - Social cliques.



Related Work

- **Pioneering Works :**
  - A. Narayanan and V. Shmatikov, “De-anonymizing social networks”, in IEEE Symposium on Security and Privacy, pp. 173 – 187, 2009. (Seeded)
  - P. Pedarsani and M. Grossglauser, “On the privacy of anonymized networks” in Proc. ACM SIGKDD, pp. 1235 – 1243, 2011. (Seedless)
- **De-anonymization with Communities :**
  - E. Onaran, G. Siddharth and E. Erkip, “Optimal de-anonymization in random graphs with community structure”, arXiv preprint arXiv :1602.01409, 2016.
  - X. Fu, Z. Hu, Z. Xu, L. Fu and X. Wang, “De-anonymization of Networks with Communities : When Quantifications Meet Algorithms”, IEEE Globecom, 2017.

Our Contributions

In this work, we

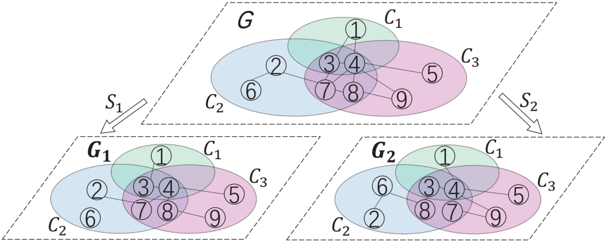
- study the effect of overlapping communities on seedless de-anonymization ;
- target at minimizing the expected de-anonymization error initially ;
- provide a systematic study for the above setting, including model, theory, algorithm, and experiments on real data.

Outline

Problem Formulation

- How to build the model ?
- **Observation :**
  - Connection $\rightarrow$ Friends.
  - Friends $\not\rightarrow$ Connection.
- **Characterization :**
  - Connection : Social Networks (Exposed).
  - Friends : Relationship Networks (Underlying).
- **Modeling :**
  - Social Network partially presents Relationship Network ;
  - Social network : a sampling of Relationship Network.

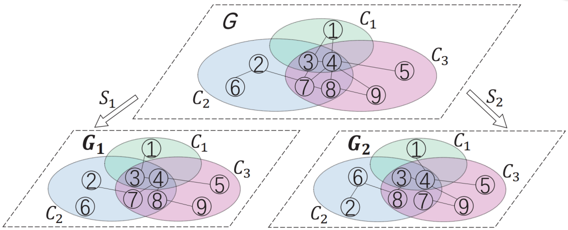
Problem Formulation



- $G(V,E)$: The Underlying Relationship Networks.
- $G_1(V,E_1)$: The Anonymized Networks.
- $G_2(V,E_2)$: The Un-anonymized Networks.

- Parameters : $\theta = \{\{p\}_{ij}, s_1, s_2\}$.

Social Network De-anonymization



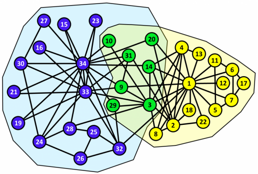
Definition (Social Network De-anonymization)

Given $G_1 = (V, E_1)$, $G_2 = (V, E_2)$, and $\boldsymbol{\theta} = \{\{p_{ij}\}, s_1, s_2\}$, the goal is to construct a mapping π that is closest to the correct mapping π_0.

$$\pi_0 = \{(1,1), (2,6), (3,3), (4,4), (5,5), (6,2), (7,8), (8,7), (9,9)\}$$

Overlapping Communities

- **Overlapping Stochastic Block Model (OSBM)**
  - Overlapping communities.
  - Higher overlapping, Higher connection possibility.



A simple version of OSBM :

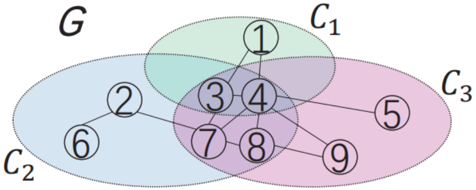
$$P((i,j) \in E) \triangleq p_{ij} = \frac{1}{1 + ae^{-x_{ij}}}.$$

- x : number of common communities of user i and j.
- a : the density parameter.

Overlapping Communities

$$P((i,j) \in E) \triangleq p_{ij} = \frac{1}{1 + ae^{-x_{ij}}}$$

Example :



- $P((1,4) \in E) = p_{14} = \frac{1}{1+ae^{-1}}$
- $P((2,5) \in E) = p_{25} = \frac{1}{1+a}$
- $P((3,4) \in E) = p_{34} = \frac{1}{1+ae^{-3}}$

Outline

1. Introduction
2. Problem Formulation
3. Analytical Aspect
4. Algorithmic Aspect
5. Experimental Aspect
6. Conclusion

Minimization of Expected Error

- **Goal :** minimizing the expected de-anonymization error.
- De-anonymization Error :
  - A mapping $\pi \leftrightarrow$ A permutation matrix Π_0

$$\pi = \{(1,2),(2,1),(3,3)\} \leftrightarrow \Pi = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

  - $d(\Pi, \Pi_0) = \frac{1}{2}||\Pi - \Pi_0||_F^2$ is the number of error mappings.
- Expected :
  - Minimizing $\mathbf{E}_{\Pi_0}\{d(\Pi, \Pi_0)\}$,
    - Expectation over different ground-truth Π_0.
  - **Minimum Mean Square Error (MMSE)**

Minimum Mean Square Error (MMSE)

- We intend to find Π as a minimizer of the expected de-anonymization error.

MMSE Estimator

Given G_1, G_2 and θ, the MMSE estimator is an estimation of Π_0 minimizing the number of mistakenly matched nodes in expectation, which is

$$\hat{\Pi} = \arg\min_{\Pi \in \Pi^n} \mathbf{E}_{\Pi_0}\{d(\Pi, \Pi_0)\}$$
$$= \arg\min_{\Pi \in \Pi^n} \sum_{\Pi_0 \in \Pi^n} ||\Pi - \Pi_0||_F^2 Pr(\Pi_0 | G_1, G_2, \theta),$$

where Π^n is the set of $n \times n$ permutation matrices.

Minimum Mean Square Error (MMSE)

Theorem 1

Given G_1, G_2 and θ, the MMSE estimator can be equivalently reformed as

$$\hat{\Pi} = \arg\max_{\Pi \in \Pi^n} \sum_{\Pi_0 \in \Pi^n} ||\Pi - \Pi_0||_F^2 ||\mathbf{W} \circ (\Pi_0 \mathbf{A} - \mathbf{B}\Pi_0)||_F^2,$$

where $\circ$ means the Hadamard product, $\mathbf{W}$ satisfies that $\mathbf{W}(i,j) = \sqrt{w_{ij}}$ and $w_{ij} = \log\left(\frac{1 - p_{\boldsymbol{c_i c_j}}(s_1 + s_2 - s_1 s_2)}{p_{\boldsymbol{c_i c_j}}(1 - s_1)(1 - s_2)}\right)$.

- But, **Is it easy to solve** ?
- It is NP-hard.

Transformation of MMSE

- Transform and simplify the original problem.
- $\hat{\Pi} = \arg\max_{\Pi \in \Pi^n} \sum_{\Pi_0 \in \Pi^n} ||\Pi - \Pi_0||_F^2 ||\mathbf{W} \circ (\Pi_0 \mathbf{A} - \mathbf{B}\Pi_0)||_F^2$.

Weighted-Edge Matching Problem (WEMP)

Given $G_1(V, E_1)$, $G_2(V, E_2)$ and weight matrix $\mathbf{W}$, the weight-edge matching problem is to find

$$\tilde{\Pi} = \arg\min_{\Pi \in \Pi^n} ||\mathbf{W} \circ (\Pi \mathbf{A} - \mathbf{B}\Pi)||_F^2$$

Validity of Transformation

$$\hat{\Pi} = \arg\max_{\Pi \in \Pi^n} \sum_{\Pi_0 \in \Pi^n} ||\Pi - \Pi_0||_F^2 ||\mathbf{W} \circ (\Pi_0 \mathbf{A} - \mathbf{B}\Pi_0)||_F^2$$

$$\Downarrow \; ?$$

$$\tilde{\Pi} = \arg\min_{\Pi \in \Pi^n} ||\mathbf{W} \circ (\Pi \mathbf{A} - \mathbf{B}\Pi)||_F^2$$

Valid ?

- In average case : valid based on Sequence Inequality.
- For a specific network : an approximation ratio with lower bound 0.5.

Outline

Algorithmic Aspect

After transforming to WEMP, there are 2 crucial issues :

- Why does optimizing WEMP work ?
  - The **advantage** of solving WEMP ?
- How can we solve it ?
  - The **mechanism** for solving WEMP ?

Optimality v.s. **Complexity**

Advantage of Solving WEMP

- **Aspect 1** : Advantage of WEMP
  - Under mild conditions, the optimal solution of WEMP $\tilde{\Pi}$ can make the error negligible.
  - Negligible : Relative Node Mapping Error (RNME) $\rightarrow 0$.

$$\mathrm{RNME} = \frac{||\tilde{\Pi} - \Pi_0||_F^2}{||\Pi_0||_F^2}$$

Notation : $||\mathbf{W} \circ (\Pi \mathbf{A} - \mathbf{B}\Pi)||_F^2 = ||\Pi \hat{\mathbf{A}} - \hat{\mathbf{B}}\Pi||_F^2$

Advantage of Solving WEMP

Theorem 2

Given G_1, G_2, θ, $\mathbf{W}$. Set

$$K = \min_{s,t,j}\{(p_{c_s c_j} + p_{c_t c_j})\min\{s_1, s_2\}\},$$

$$L = \max_{s,t,j}\{[(p_{c_s c_j} + p_{c_t c_j})\max\{s_1, s_2\}]^2\}.$$

If the following four conditions :

- $\frac{L}{K} = o(1)$;
- the minimizer of WEMP, $\tilde{\Pi}$, satisfies that $||\hat{\mathbf{A}} - \Pi_0 \hat{\mathbf{B}} \Pi_0^T||_F^2 / ||\hat{\mathbf{A}} - \tilde{\Pi}\hat{\mathbf{B}}\tilde{\Pi}^T||_F^2 = \Omega(1)$;
- $||\hat{\mathbf{A}} - \Pi_0 \hat{\mathbf{B}} \Pi_0^T||_F^2 = o(Kn^2)$;
- Π_0 and $\tilde{\Pi}$ keep invariant of the community representations,

hold, then the *RNME*, $||\tilde{\Pi} - \Pi_0||_F^2 / ||\Pi_0||_F^2$, can be upper bounded by the minimum value of WEMP, i.e., $||\hat{\mathbf{A}} - \tilde{\Pi}\hat{\mathbf{B}}\tilde{\Pi}^T||_F^2$, and as $n \to \infty$, *RNME* $\to 0$.

Advantage of Solving WEMP

- Why are the conditions **mild** ?
- Take the example of **OSBM**.
  - $a = \Omega(1)$.
  - $s = o(1)$ and $\hat{p} = 1 - o(1)$, then $\hat{p}\log(\frac{1-\hat{p}(2s-s^2)}{\hat{p}(1-s)^2}) = \hat{p}\log(1 + \frac{1-\hat{p}}{\hat{p}(1-s)^2}) \approx \frac{1-\hat{p}}{(1-s)^2} = o(1) = o(\min_{i,j} p_{\mathbf{c}_i\mathbf{c}_j})$, thus condition (iii) holds.
  - Meanwhile $s = o(1)$ makes condition (i) hold.
  - Easy to verify that condition (ii),(iv) hold.

Mechanism for Solving WEMP

- **Aspect 2 :** Mechanism for WEMP
- Definitions :
  - **Community Representation (C_i)** : Communities $\{1, 2, 3, 4\}$, vertex i in $\{1, 3\}$, then $C_i = \{1, 0, 1, 0\}$.
  - **Community Representation Matrix (M)** :
    - The i^{th} row of **M** is C_i.

$$\text{If } \left\{ \begin{array}{c} 1 \rightarrow C_1 \\ 2 \rightarrow C_2 \\ 3 \rightarrow C_1, C_2 \end{array} \right\} \text{ then } M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

Mechanism for Solving WEMP

Formulating WEMP :

$$\begin{aligned} &\text{minimize} \quad \|\Pi\hat{\mathbf{A}} - \hat{\mathbf{B}}\Pi\|_F^2 \\ &\textbf{s.t.}\ \forall i \in V_1,\ \textstyle\sum_i \Pi_{ij} = 1 && (1) \\ &\forall j \in V_2,\ \textstyle\sum_j \Pi_{ij} = 1 && (2) \\ &\forall i, j,\ \Pi_{ij} \in \{0, 1\}, && (3) \end{aligned}$$

$$\forall i \in V_1, C_i = C_{\pi(i)}. \tag{4}$$

Embedding Eqn. (4) into the objective function we get

$$F_0(\Pi) = ||\Pi\hat{\mathbf{A}} - \hat{\mathbf{B}}\Pi||_F^2 + \mu||\Pi\mathbf{M} - \mathbf{M}||_F^2.$$

Idea of Algorithm Design

Problem Relaxation :

$$\Omega_0 = \{\Pi_{ij} \in \{0,1\} | \forall i,j, \sum_i \Pi_{ij} = 1 \ , \sum_j \Pi_{ij} = 1\};$$
$$\Omega = \{\Pi_{ij} \in [0,1] | \forall i,j, \sum_i \Pi_{ij} = 1 \ , \sum_j \Pi_{ij} = 1\}.$$

Convex-Concave Relaxation Method :

$$F(\Pi) = (1-\alpha)F_1(\Pi) + \alpha F_2(\Pi)$$

- F_1 is the convex relaxation of F.
- F_2 is the concave relaxation of F.
- α is an adjustable parameter from $[0, 1]$.

A simple way to obtain F_1 and F_2

Lemma 3

A way to get convex and concave relaxation is

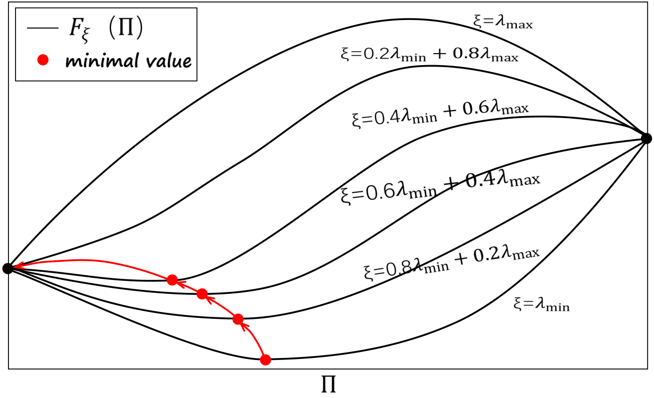
$$F_1(\Pi) = F_0(\Pi) + \frac{\lambda_{min}}{2}(n - ||\Pi||_F^2)$$
$$F_2(\Pi) = F_0(\Pi) + \frac{\lambda_{max}}{2}(n - ||\Pi||_F^2)$$

Therefore we form our new objective function in CCOM as

$$F_\xi(\Pi) = (1-\alpha)F_1(\Pi) + \alpha F_2(\Pi) = F_0(\Pi) + 2\xi(n - ||\Pi||_F^2),$$

where λ_{min} (λ_{max}) is the smallest (largest) eigenvalue of the Hessian matrix of $F_0(\Pi)$,and $\xi = (1-\alpha)\lambda_{min} + \alpha\lambda_{max}$, $\xi \in [\lambda_{min}, \lambda_{max}]$.

An illustration of Convex-Concave Method



Main Algorithm

- Main Algorithm

Algorithm 1: Convex-concave Based De-anonymization Algorithm (CBDA)

Input: Adjacent matrices $\mathbf{A}$ and $\mathbf{B}$; Community assignment matrix $\mathbf{M}$; Weight controlling parameter μ; Adjustable parameters δ, $\Delta\xi$.
Output: Estimated permutation matrix $\tilde{\mathbf{\Pi}}$.

Form the objective function $F_0(\mathbf{\Pi})$ and $F(\mathbf{\Pi})$.
$\xi \leftarrow 0$, $k \leftarrow 1$, $\mathbf{\Pi_1} \leftarrow \mathbf{1}_{n\times n}./n$. Set ξ_m, the upper limit of ξ.
while $\xi < \xi_m$ and $\mathbf{\Pi_k} \notin \Omega_0$ **do**
  while $k = 1$ or $|F(\mathbf{\Pi_{k+1}}) - F(\mathbf{\Pi_k})| \geq \delta$ **do**
    $\mathbf{X}^{\perp} \leftarrow \arg\min_{\mathbf{X}^{\perp}} \mathbf{tr}(\nabla_{\mathbf{\Pi_k}} F(\mathbf{\Pi_k})^T \mathbf{X}^{\perp})$, where $\mathbf{X}^{\perp} \in \Omega$.
    $\gamma_k \leftarrow \arg\min_{\gamma} F(\mathbf{\Pi_k} + \gamma(\mathbf{X}^{\perp} - \mathbf{\Pi_k}))$, where $\gamma_k \in [0, 1]$.
    $\mathbf{\Pi_{k+1}} \leftarrow \mathbf{\Pi_k} + \gamma_k(\mathbf{X}^{\perp} - \mathbf{\Pi_k})$, $k \leftarrow k + 1$.
  end while
  $\xi \leftarrow \xi + \Delta\xi$.
end while

Convergence Proof

Lemma 4

CBDA converges and the final output is a permutation matrix in the original feasible region Ω_0.

Proof sketch :

$$F_\xi(\Pi_{k+1}) \leq F_\xi(\Pi_k) + \gamma_k(F_\xi(\Pi^\xi) - F_\xi(\Pi_k)) + \gamma_k \Delta R_k.$$

$$F_\xi(\Pi_{k+1}) - F_\xi(\Pi^\xi)$$

$$\leq \prod_{i=1}^{k}(1-\gamma_i)\Delta\xi(||\Pi^{\xi-\Delta\xi}||_F^2 - ||\Pi^\xi||_F^2) + \sum_{i=1}^{k}\gamma_i \prod_{j=1}^{k-i}(1-\gamma_j)\Delta R_i.$$

Outline

Experimental Aspect

Synthetic Networks :

Notation	Definition	Range
N	Number of Nodes	{500, 1000, 1500, 2000}
s	Sampling Probability ($s_1 = s_2 = s$)	0.3-0.9
a	OSBM Parameter	{3, 5, 7, 9}
η	Community Ratio	{0.05, 0.1}
OL/NOL	Overlapping or Non-Overlapping	{OL, NOL}

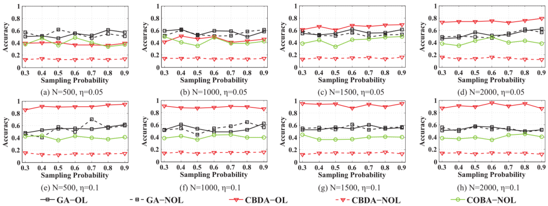


Fig. 2: Experiments on Synthetic Networks.

Experimental Aspect

Sampled Social Networks :

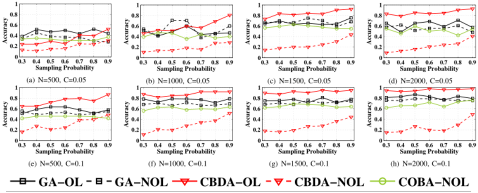
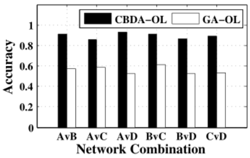
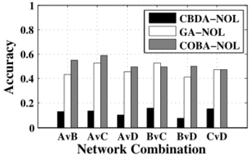


Fig. 8: Experiments on Sampled Real Social Networks.

Cross-Domain Networks :





Outline

Conclusion

- **Conclusion :**
  - De-anonymization can be achieved under mild conditions.
  - Overlapping communities benefits de-anonymization.
- **Future directions :**
  - Theoretical bounds for successful de-anonymization ;
  - Partial overlapping users ;
  - Multilevel network de-anonymization.

Thanks !