Impact of Clustering on the Performance of Network De-anonymization

Carla-Fabiana Chiasserini
Dipartimento di Elettronica e Telecomunicazioni
Politecnico di Torino, Italy
chiasserini@polito.it

Michele Garetto
Dipartimento di Informatica
Universita' di Torino, Italy
michele.garetto@unito.it

Emilio Leonardi
Dipartimento di Elettronica e Telecomunicazioni
Politecnico di Torino, Italy
leonardi@polito.it

ABSTRACT

Recently, graph matching algorithms have been successfully applied to the problem of network de-anonymization, in which nodes (users) participating in more than one social network are identified only by means of the structure of their links to other members. This procedure exploits an initial set of seed nodes large enough to trigger a percolation process which correctly matches almost all other nodes across the different social networks. Our main contribution is to show the crucial role played by clustering, which is a ubiquitous feature of realistic social network graphs (and many other systems). Clustering has both the effect of making matching algorithms more vulnerable to errors, and the potential to dramatically reduce the number of seeds needed to trigger percolation, thanks to a wave-like propagation effect. We demonstrate these facts by considering a fairly general class of random geometric graphs with variable clustering level, and showing how clever algorithms can achieve surprisingly good performance while containing matching errors.

Categories and Subject Descriptors

G.3 [**Mathematics of Computing**]: Probability and Statistics—*Probabilistic algorithms*; G.2.2 [**Discrete Mathematics**]: Graph Theory; H.1 [**Information Systems**]: Models and Principles

Keywords

Graph matching; bootstrap percolation; social networks; de-anonymization; privacy

1. INTRODUCTION

The advent of online social networks, and their massive worldwide penetration, can be well considered as one of the most influential changes brought by information and communication technologies into our lives during the last decade, with profound impact on all aspects of economy, society and culture. The extraordinary capitalization of the companies running these (typically free) online services can be explained by the huge amount of valuable information that can be extracted from the traces of activities performed by billions of users. Such information allows, for example, to build user profiles that can be effectively used for targeted advertisements, marketing and social surveys, and many other profitable business run by service providers and third parties. Privacy concerns raised by the collection, analysis and distribution of personal data, exposed more or less consciously by active users, have been recently hotly debated in the media. User privacy is especially threatened when data collected from different systems is combined together to construct richer and more accurate user profiles.

In this work we are specifically concerned with the problem of identifying users participating in different online social networks We emphasize that this problem can be perceived by people in totally different ways. Some users would prefer to hide any Personal Identifiable Information (PII) while using a service, and they see any attempt to correlate accounts created in different systems as a severe violation of their privacy. Other users instead are more than happy to merge or link together their various accounts, as this turns out to be convenient to the user itself. For example, the increasing practice of 'social logins' allow users to use existing accounts on social networks to directly sign into other services (different applications, websites, public Wi-Fi hotspots).

In our work, we are specifically interested in privacy issues, and consider the case of an 'attacker' trying to identify users belonging to two different social networks (without their consent). Recently, security experts have made the dramatic discovery that user privacy cannot be guaranteed when traces of communication activities are made available after applying the simple anonymization procedure which replaces real ID's by random labels [1].

A standard way to formalize the user identification problem is the following: each communication system (e.g., a given social network) generates (from the traces of user activities) a 'contact graph' in which nodes represent anonymized users, and edges denote who has come in contact with whom. The attacker then runs a *graph matching* algorithm on the contact graphs generated by different systems, which in the hardest case can make use only of the topologies of these graphs, without any additional side information [2]. The majority of algorithms proposed so far to achieve this goal are facilitated by an initial set of already matched nodes (called seeds). This is actually a realistic case, since, as explained above, some users explicitly link their accounts in different systems 'for free'. Many proposed matching strategies, based on heuristic algorithms, work by progressively expanding the set of already matched nodes, trying to identify all of the other nodes [1, 3, 4]. In particular, in their seminal paper Narayanan and Shmatikov [1] were able to identify a large fraction of users having account on both Twitter and Flickr (with only 12% error ratio).

COSN'15, November 2–3, 2015, Palo Alto, California, USA.
 ISBN 978-1-4503-3951-3/15/11 ...$15.00.
DOI: http://dx.doi.org/10.1145/2817946.2817953.

Significant progress has also been made towards theoretical understanding of the feasibility of network de-anonymization (in the first place), and of the asymptotic performance of graph matching algorithms applied to large systems. Recent analytical work has adopted the following convenient probabilistic generation model for two contact graphs $\mathcal{G}_1$ and $\mathcal{G}_2$: we consider the (inaccessible) 'ground-truth' graph $\mathcal{G}_\mathrm{T}$ representing true social relationships among people, and then assume that $\mathcal{G}_1$ is obtained by independently sampling each edge of $\mathcal{G}_\mathrm{T}$ with probability s (similarly, and independently, $\mathcal{G}_2$). Specifically, when the social network $\mathcal{G}_\mathrm{T}$ is modeled as an Erdös–Rényi random graph, it has been shown in [5] that, under mild conditions, users participating in two different social networks can be successfully matched by an attacker with unlimited computation power, even without seeds. Still in the case of Erdös–Rényi random graphs, in [6] authors have proposed a practical identification algorithm based on bootstrap percolation, [7] showing an interesting phase transition phenomenon in the number of seeds that are required for network de-anonymization. The results in [6] have been recently extended to the more realistic case in which contact graphs are scale-free (power-law) random graphs. In particular, by modeling them as Chung-Lu graphs, [8] and [9] have independently shown that a much smaller set of seeds is sufficient to trigger the percolation-based matching process originally studied in Erdös–Rényi graphs.

While previous work has captured the impact of power-law degree distribution on percolation graph matching, another essential feature of real social networks, namely, clustering, has not been investigated so far. Interestingly, in [6] authors attempted to apply their basic algorithm also to highly clustered random geometric graphs, observing almost total failure (error rates above 50%). This preliminary finding has been the starting point of our work. In this paper we consider a fairly general model of random geometric graphs that allows us to incorporate various levels of clustering in the underlying social network, without concurrently generating a scale-free structure. By so doing, we separate the (unkown) impact of clustering from the (known) impact of power law degree, going back to the original case of Erdös–Rényi graphs and exploring a totally different, 'orthogonal' direction. Our main findings are as follows:

(i) Clustered networks can be indeed largely prone to matching errors when we naively apply the method proposed in [6]. Such errors can be mitigated and asymptotically eliminated by an improved matching algorithm still based on bootstrap percolation;

(ii) Once errors are eliminated, clustering turns out to have a surprising beneficial effect on the performance of graph matching, thanks to a wave-like propagation phenomenon that allows to progressively identify all nodes starting from a very small, *compact* set of seeds;

(iii) In contrast with previous results derived for Erdös–Rényi [6] and Chung-Lu graphs [8], we show that the minimum number of seeds required for network de-anonymization can increase with the average node degree of the graph.

Our results are qualitatively validated via experiments with synthetic and real social network graphs. We emphasize that, although we focus on network de-anonymization, the results derived in this work have much broader applicability, since graph matching problems are arising in many different domains, ranging from computer graphics to bioinformatics.

2. NOTATION AND PRELIMINARIES

Without loss of generality, we assume that $\mathcal{G}_\mathrm{T}(\mathcal{V},\mathcal{E})$, $\mathcal{G}_1(\mathcal{V}_1,\mathcal{E}_1)$ and $\mathcal{G}_2(\mathcal{V}_2,\mathcal{E}_2)$ have the same set of nodes (or vertices)

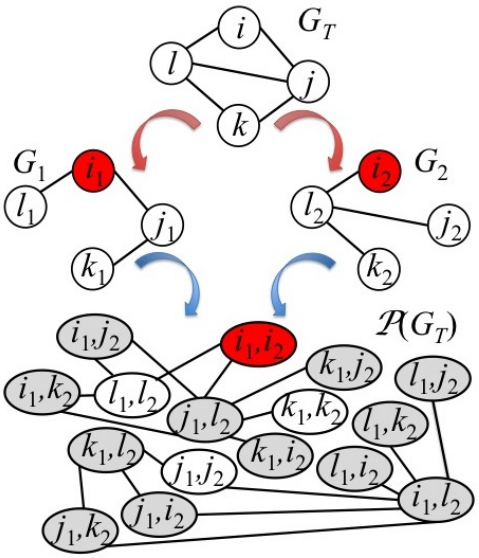

Figure 1: An example of $\mathcal{G}_1$ and $\mathcal{G}_2$ obtained from $\mathcal{G}_\mathrm{T}$ by independent edge sampling, and of the pairs graph $\mathcal{P}(\mathcal{G}_\mathrm{T})$. There is a single seed, highlighted in red. In $\mathcal{P}(\mathcal{G}_\mathrm{T})$, good pairs are highlighted in white and bad pairs in grey.

with cardinality n, i.e., $\mathcal{V}_1=\mathcal{V}_2=\mathcal{V}$ [1]. Similarly to previous work [4, 5, 6, 8, 9] we assume that edges in $\mathcal{G}_1$ and $\mathcal{G}_2$ are obtained by independently sampling each edge of $\mathcal{G}_\mathrm{T}$ with probability s. Specifically, each edge in $\mathcal{G}_\mathrm{T}$ is assumed to be (independently) sampled twice, the first time to determine its presence in $\mathcal{E}_1$, the second time to determine its presence in $\mathcal{E}_2$. This model is a reasonable approximation of real systems which permits obtaining fundamental analytical insights.

To match $\mathcal{G}_1$ and $\mathcal{G}_2$, we build the pairs graph $\mathcal{P}(\boldsymbol{\mathcal{V}},\boldsymbol{\mathcal{E}})$, with $\boldsymbol{\mathcal{V}}\subseteq\mathcal{V}_1\times\mathcal{V}_2$ and $\boldsymbol{\mathcal{E}}\subseteq\mathcal{E}_1\times\mathcal{E}_2$. In $\mathcal{P}(\boldsymbol{\mathcal{V}},\boldsymbol{\mathcal{E}})$ there exists an edge between $[i_1,j_2]$ and $[k_1,l_2]$ iff edge $(i_1,k_1)\in\mathcal{E}_1$ and edge $(j_2,l_2)\in\mathcal{E}_2$. We will slightly abuse the notation and denote the pair graph associated to a generic ground-truth graph $\mathcal{G}_\mathrm{T}$ simply as $\mathcal{P}(\mathcal{G}_\mathrm{T})$. Fig. 1 shows the pairs graph built from a toy example.

We will refer to pairs $[i_1,i_2]\in\mathcal{P}(\mathcal{G}_\mathrm{T})$, whose vertices correspond to the same vertex $i\in\mathcal{G}_\mathrm{T}$, as good pairs, and to all others (e.g., $[i_1,j_2]$) as bad pairs. Also, we will refer to two pairs such as $[i_1,j_2]$ and $[i_1,l_2]$, or $[i_1,j_2]$ and $[k_1,j_2]$, as conflicting. Finally, two adjacent pairs on $\mathcal{P}(\mathcal{G}_\mathrm{T})$ will be referred to as neighbors. The seeds set will be denoted by $\mathcal{A}_0(n)\subset\mathcal{V}$, with cardinality a_0.

We now briefly describe the Percolation Graph Matching (PGM) algorithm originally proposed in [6]. The PGM algorithm maintains an integer counter (initialized to zero) for any pair of $\mathcal{P}(\mathcal{G}_\mathrm{T})$ that may still be matched. It exploits a set $\mathcal{A}_t$, indexed by time step t, which is initialized (for $t=0$) with the seed pairs. At any given time $t\geq 0$, the PGM algorithm extracts at random one pair from $\mathcal{A}_t$ matching the corresponding nodes, and increases by one the counter associated to each of its neighbor pair in $\mathcal{P}(\mathcal{G}_\mathrm{T})$. Then the algorithm adds to $\mathcal{A}_{t+1}$ all pairs whose counter has reached r at time t with the exception of those pairs that are in conflict with either any of the already matched pairs or any of the pairs in $\mathcal{A}_t$. The algorithms stops when $\mathcal{A}_t=\emptyset$. It is straightforward to see that PGM takes at most n steps to terminate.

[1] This assumption can be easily removed by considering that only the intersection of vertices belonging to $\mathcal{G}_1$ and $\mathcal{G}_2$ has to be de-anonymized.

Table 1: Main system parameters

Symbol	Definition
$\mathcal{G}_T$	ground-truth graph
$\mathcal{G}_1$ and $\mathcal{G}_2$	contact graphs
$\mathcal{V}$, $\mathcal{V}_1$ and $\mathcal{V}_2$	set of vertices of $\mathcal{G}_T$, $\mathcal{G}_1$ and $\mathcal{G}_2$
$\mathcal{E}$, $\mathcal{E}_1$ and $\mathcal{E}_2$	set of edges of $\mathcal{G}_T$, $\mathcal{G}_1$ and $\mathcal{G}_2$
s	edge sampling probability
$\mathcal{P}(\mathcal{G}_T)$	pair graph
$\hat{\mathcal{P}}(\mathcal{G}_T)$	imperfect pair graph
$\mathcal{A}_0(n)$	seed set
$\mathcal{H}$	k-dimensional network domain
$p_{ij} = K(n)\min\left(1, \left(\frac{C(n)}{d_{ij}}\right)^{\beta}\right)$	edge (i,j) probability
$D(n)$	average degree of vertices

In the case where $\mathcal{G}_T$ is an Erdös–Rényi random graph, previous work [6] has established the following lower bound on the number of seeds that are needed to correctly match almost all nodes without errors. Table 1 summarizes the main parameters of the system.

Critical seed set size for Erdös-Rényi graphs [6]. *Let $\mathcal{G}_T$ be an Erdös-Rényi random graph $G(m,p)$. Let $r \geq 4$. Denote by a_c the critical seed set size:*

$$a_c = \left(1 - \frac{1}{r}\right)\left(\frac{(r-1)!}{m(ps^2)^r}\right)^{\frac{1}{r-1}}. \quad (1)$$

For $m^{-1} \ll ps^2 \leq s^2 m^{-\frac{3.5}{r}}$, we have that, if $a_o/a_c \to a > 1$, the PGM algorithm matches w.h.p. a number of good pairs equal to $m - o(m)$ (i.e., all vertex pairs except for a negligible fraction) with no errors.

Critical seed set size for random graphs bounded by Erdös-Rényi graphs. Let $\mathcal{H}(\mathcal{V}, \mathcal{E}_H)$ and $\mathcal{K}(\mathcal{V}, \mathcal{E}_K)$ be two random graphs insisting on the same set of vertices $\mathcal{V}$, where $\mathcal{E}_H \subseteq \mathcal{E}_K$. We define the following partial order relationship: $\mathcal{H}(\mathcal{V}, \mathcal{E}_H) \leq_{st} \mathcal{K}(\mathcal{V}, \mathcal{E}_K)$. Given that, we introduce the following extended results (in part borrowed from our previous work [8]):

Theorem 1. *Consider $\mathcal{G}_T$ satisfying: $G(m, p_{\min}) \leq_{st} \mathcal{G}_T \leq_{st} G(m, p_{\max})$ with $p_{\min} \leq p_{\max}$. Applying the PGM algorithm to $\mathcal{P}(\mathcal{G}_T)$ guarantees that $m - o(m)$ good pairs are matched with no errors w.h.p., provided that:*

1. *$m \to \infty$;*
2. *$p_{\min} = \Theta(p_{\max})$ and $p_{\min} \gg m^{-1}$;*
3. *$p_{\max} \leq m^{-\frac{3.5}{r}}$;*
4. *$\liminf_{m\to\infty} a_o/a_c > 1$, with a_c computed from (1) by setting $p = p_{\min}$.*

Also, under conditions 1)-4), the PGM successfully matches w.h.p. $m - o(m)$ correct pairs (with no errors) also in any subgraph $\mathcal{G}'_T$ of $\mathcal{G}_T$ that comprises a finite fraction of vertices of $\mathcal{G}_T$ and all the edges between the selected vertices. The proof can be found in our technical report [10].

Corollary *1. Under the same conditions as in Theorem 1, the PGM algorithm can be successfully applied to an imperfect pairs graph $\hat{\mathcal{P}}(\mathcal{G}_T) \subset \mathcal{P}(\mathcal{G}_T)$ comprising a finite fraction of the pairs in $\mathcal{P}(\mathcal{G}_T)$ and satisfying the following constraint: a bad pair $[i_1, j_2] \in \mathcal{P}(\mathcal{G}_T)$ is included in $\hat{\mathcal{P}}(\mathcal{G}_T)$ only if either $[i_1, i_2]$ or $[j_1, j_2]$ are also in $\hat{\mathcal{P}}(\mathcal{G}_T)$.*

The above results provide basic building blocks to perform the asymptotic analysis of the number of seeds that are sufficient to de-anonymize clustered networks described by the model presented next.

3. CLUSTERED NETWORK MODEL

To incorporate different degrees of clustering in the ground-truth social network $\mathcal{G}_T$, we have adopted the following geometric random graph model, which guarantees a large degree of flexibility, while inheriting the main features of the small-world graphs. We assume that nodes are located in a k-dimensional space corresponding to the hyper-cube[2] $\mathcal{H} = [0,1]^k \subset \mathbb{R}^k$, where the k dimensions could correspond to different attributes of the users. We consider n nodes independently and uniformly distributed over $\mathcal{H}$. Notice that the node density in the space is n. Given any two vertices $i, j \in \mathcal{V}$, with $i \neq j$, edge (i,j) exists in $\mathcal{G}_T$ with probability p_{ij} that depends only on the Euclidean distance d_{ij} between i and j (independently of everything else). We consider the following generic law for p_{ij}:

$$p_{ij} = K(n) f(d_{ij}). \quad (2)$$

In (2), f is a non-increasing function of the distance, and $K(n)$ is a normalization constant introduced to impose a desired average node degree $D(n)$, which is assumed to be the same for all nodes. It is customary in random graph models representing realistic systems to assume that the average node degree is not constant, but it increases with n due to network densification. Also, although a common choice is to assume $D(n) = \Theta(\log n)$, in our model we consider more in general $D(n) = \Omega(\log n)$.

Since we are interested in the order-sense asymptotic performance of network de-anonymization as n grows large, we further characterize the shape of function f as follows. We assume that $f(d)$ equals 1 for all distances $0 < d < C(n)$, where $C(n)$ is a parameter of the model (possibly scaling with n). Note that this implies that $K(n)$ must be less than or equal to 1 to obtain a proper probability function. For distances larger than $C(n)$, we assume that f decays according to a power-law with exponent β, with $\beta > 0$. In summary,

$$f(d_{ij}) = \min\left\{1, \left(\frac{C(n)}{d_{ij}}\right)^{\beta}\right\}. \quad (3)$$

The above characterization of the shape of $f(d)$ is fairly general and allows accounting for different levels of node clustering. In particular, our random-graph model degenerates into a standard Erdös–Rényi graph when $C(n)$ approaches 1, with arbitrary β. For $\beta \to \infty$, instead, edges can be established only between nodes whose distance is smaller than or equal to $C(n)$.

The average node degree is:

$$D(n) = \Theta\left(nK(n)\left(C^k(n) + C^{\beta}(n)\int_{C(n)}^{1} \rho^{k-1-\beta}\, d\rho\right)\right).$$

From the above equation it follows that for $\beta > k$ the dominant fraction of the neighbors of a given node lie at distance $\Theta(C(n))$ from it, while for $\beta < k$ only a marginal fraction of the neighbors of a node lie at distance $o(1)$ from it. Since we are interested in graphs with significant node clustering (so as to mimic real-world social networks), we restrict the analysis in this paper to the case $\beta > k$. In this case, the average node degree is:

$$D(n) = \Theta(nK(n)C^k(n)). \quad (4)$$

[2]To avoid border effects, we assume wrap-around conditions (i.e., a torus topology).

Since by construction $K(n) \leq 1$, the average node degree is constrained to be $O(nC^k(n))$. Moreover, given that we assume $D(n) = \Omega(\log n)$, we have $C(n) = \Omega\left(\left(\frac{\log n}{n}\right)^{1/k}\right)$.

The clustering coefficient turns out to be $\Theta(K(n))$, as direct consequence of the fact that almost all neighbors of a node lie at distance $\Theta(C(n))$ from it. We remark that the clustering coefficient of the networks generated according to our model is always much larger than in an Erdös–Rényi graph having the same average node degree (recall that in $G(n,p)$ the clustering coefficient is p). To see this, we observe that in our model the ratio between the clustering coefficient $\Theta(K(n))$ and the graph density [3] is $\Theta(1/C^k(n))$. Since in general $C^k(n) = o(1)$, our graph model exhibits a high level of clustering. In the following, we will slightly abuse the language and refer to groups of vertices lying in sub-regions of side $\Theta(C(n))$ as *clusters* (not to be confused with the clustering coefficient).

In essence, in our model, which has been chosen in light of its flexibility, $K(n)$ and $C(n)$ provide the two knobs that allow us to directly control the clustering coefficient of the graph and the average node degree (or the graph density). We will see next that these are indeed the crucial parameters affecting the asymptotic performance of the proposed graph matching algorithms.

4. OVERVIEW AND MAIN RESULTS

In our analysis we have to distinguish two cases:
1) $K(n) = o((nC^k(n))^{-\gamma})$, for some $\gamma > 0$, which will be referred to as *sparse clusters* case;
2) $K(n) = \omega((nC^k(n)^{-\gamma}))$ for any $\gamma > 0$, which will be referred to as *dense clusters* case.
In the first case the clustering coefficient goes to zero "relatively" fast as the number of nodes within a cluster goes to infinity (i.e., when $nC^k(n) \to \infty$). In the second case, the clustering coefficient is either bounded away from zero or decreases very slowly. It comprises the particularly relevant sub-case in which $K(n) = \Theta(1)$.

In the *sparse clusters* case the density of edges within a cluster is sufficiently small that the PGM algorithm can be safely applied within it without incurring matching errors. We therefore apply the following de-anonymization procedure. We start from a set of seeds which are assumed to lie in a small sub-region of $\mathcal{H}$ of size $\Theta(C(n))$ (i.e., within a cluster). Then, using the PGM algorithm, we run a first 'trigger phase' in which we correctly match almost all nodes located sufficiently close (within a fixed distance) to the seeds. The identification procedure then goes on through a second phase in which nodes located in 'expanding rings' around the initial seeds are progressively identified through a sequence of steps (representing a discretized version of a wave-like expansion). Note that, in this second phase, we do not apply PGM any more, but a simpler direct strategy, matching at each step those pairs having a sufficiently large number of neighbor pairs matched at previous steps. Fig. 2 illustrates graphically this idea.

In the *dense clusters* case, de-anonymization is more complex, due to the high clustering coefficient (note that for large values of $K(n)$ the graph can have many cliques or quasi-cliques of nodes). In particular, if we tried to match the nodes using only the local structure of a cluster (as in the sparse clusters case) we would initially incur an intolerable amount of errors disrupting the entire identification process. It follows that, to guarantee that almost no errors are made, we have to ignore all edges whose length is too short (in particular, shorter than a properly defined threshold $\omega(C(n))$), and identify the nodes on the only basis of the 'fingerprint' provided by longer edges. More specifically, we devise a different 'trigger phase' which starts from two sub-regions of $\mathcal{H}$ of side $h(n) = \Theta(C(n))$, which are sufficiently far from each other (i.e., separated by a minimum distance $\omega(C(n))$, see Fig. 3). We assume that a suitable number of seeds are initially selected within each of these two sub-regions. To identify all of the other nodes in the sub-regions, we modify the PGM algorithm so that only the edges between nodes belonging to different sub-regions are used. After that, similarly to the *sparse clusters* case, we enter a second phase which progressively expands the set of matched nodes. This time we exploit the fact that, in the dense cluster regime, the distance between two nodes in $\mathcal{H}$ can be estimated quite precisely. Thus, given a sub-region where nodes have already been matched, we can select a set of compact nodes that are sufficiently far from the matched sub-region, and re-apply the modified PGM algorithm. This procedure can be iterated until almost all nodes throughout the network are correctly identified.

[3] Given a generic graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, the graph density is defined as $\frac{2|\mathcal{E}|}{|\mathcal{V}|(|\mathcal{V}|-1)}$. It can be interpreted as the probability that an edge exists between two randomly selected nodes of the graph.

Table 2: Minimum seed set size to achieve percolation

Scenario	Minimum seed set size
$K(n) = \omega((nC^k(n))^{-\gamma}), \forall \gamma > 0$	$O((nC^k(n))^{\varepsilon}), \forall \varepsilon > 0$
$K(n) = o((nC^k(n))^{-\gamma})$, with $\gamma > 0$	$\Theta\left(\frac{\log nC^k(n)}{K(n)}\right)$

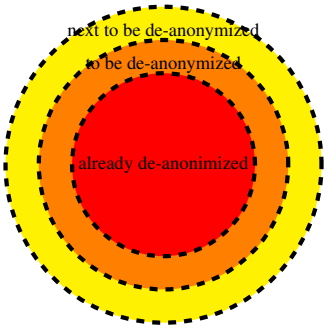

Figure 2: The de-anonymization procedure for $K(n) = o((nC^k(n))^{-\gamma})$.

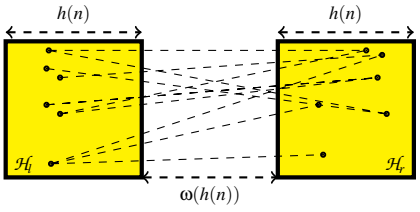

Figure 3: Bipartite graph construction for $K(n) = \omega((nC^k(n)^{-\gamma}))$.

In Table 2, we summarize our main results for the minimum seed set size that is required for successful network de-anonymization, assuming that seeds are selected within suitable clusters of $\mathcal{H}$. Observe that the minimum number of seeds depends on both $K(n)$ and $C(n)$, whereas it is independent of β. Specifically, in the *dense*

cluster case (first raw of the table), the minimum number of seeds can be simply expressed in terms of the average number of nodes falling within a cluster ($nC^k(n)$). Indeed, a seed set whose size is equal to $(nC^k(n))^\varepsilon$, for any $\varepsilon > 0$, is enough to guarantee an almost complete successful network de-anonymization. In the relevant case in which $C(n) = \Theta(\frac{\log n}{n})^{1/k}$ (i.e., when the average degree of the graph $D(n) = \Theta(\log n)$), the above expression reduces to $(\log n)^\varepsilon$, with arbitrarily small $\varepsilon > 0$. This result readily reveals the strong beneficial impact of clustering on network de-anonymization. Somehow surprisingly, the minimum seed set size increases when we increase the average degree of the nodes (i.e., for increasing $C(n)$). This is in sharp contrast with previous results derived for Erdös–Rényi and Chung-Lu graphs [6, 8]. The intuition behind this result is that, by increasing $C(n)$, we increase the cluster size, making the problem of identifying nodes within a cluster intrinsically more challenging. In the *sparse clusters* case (second raw of the table), our de-anonymization techniques become less effective, and the minimum seed set size turns out to be roughly inversely proportional to $K(n)$.

5. SPARSE CLUSTERS

In this case, we assume $K(n) = o\left((nC^k(n))^{-\gamma}\right)$, for some $\gamma > 0$, and a set of seeds $\mathcal{A}_0$ ($|\mathcal{A}_0| = a_0$) whose maximum mutual distance is $d_s = O(C(n))$.

As first phase, we show how nodes in $\mathcal{H}$ lying sufficiently close to the seeds can be identified. To this end, we start by defining two sub-regions, $\mathcal{H}_{\text{in}} \subset \mathcal{H}$ and $\mathcal{H}_{\text{out}} \subset \mathcal{H}$. Intuitively, $\mathcal{H}_{\text{in}}$ ($\mathcal{H}_{\text{out}}$) can be seen as the set of points whose distance from any seed vertex is higher (lower) than a given threshold. More formally, denote by $\mathbf{x}$ a generic point in $\mathcal{H}$ and by $\mathbf{x}_\sigma$ the position in $\mathcal{H}$ of a generic seed vertex σ. Then, given two positive constants α and δ, s.t. $\delta \leq 1$ and $\alpha(1+\delta) \leq 1$, we define:

$$\mathcal{H}_{\text{in}}(\alpha,\delta) = \left\{\mathbf{x} \text{ s.t. } \max_{\sigma\in\mathcal{A}_0} \|\mathbf{x}-\mathbf{x}_\sigma\| \leq f^{-1}((1+\delta)\alpha)\right\}$$

$$\mathcal{H}_{\text{out}}(\alpha,\delta) = \left\{\mathbf{x} \text{ s.t. } \min_{\sigma\in\mathcal{A}_0} \|\mathbf{x}-\mathbf{x}_\sigma\| > f^{-1}((1-\delta)\alpha)\right\}$$

where f is the non-increasing function defined in Section 3. The two sub-regions are depicted in Fig. 4. Note that, by construction, the area $|\mathcal{H}_{\text{in}}| = \Theta(C^k(n))$.

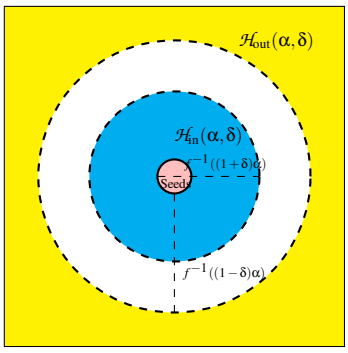

Figure 4: $\mathcal{H}_{\text{in}}(\alpha,\delta)$ and $\mathcal{H}_{\text{out}}(\alpha,\delta)$.

The theorem below proves that, given graph $\mathcal{G}_1$ ($\mathcal{G}_2$), it is possible to correctly distinguish nodes in $\mathcal{H}_{\text{in}}(\alpha,\delta)$ from nodes in $\mathcal{H}_{\text{out}}(\alpha,\delta)$ by counting the number of their neighboring seeds.

Theorem 2. *Given a node $i \in \mathcal{G}_1$ ($i \in \mathcal{G}_2$), let S_i be the number of seeds that are neighbors of i on $\mathcal{G}_1$ ($\mathcal{G}_2$). We say that node i is accepted if $S_i > \alpha s K(n) a_0$. If $d_s = O(C(n))$ and $a_0 = \Omega\left(\frac{\log(nC^k(n))}{K(n)}\right)$, then for an arbitrary $\delta > 0$, the above procedure accepts all nodes located in $\mathcal{H}_{in}(\alpha,\delta)$, while it excludes all nodes located in $\mathcal{H}_{out}(\alpha,\delta)$.*

PROOF. See Appendix A. □

Note that, in the above statement, $sK(n)$ is the probability that a node in $\mathcal{G}_1$ ($\mathcal{G}_2$) is connected with a seed node at distance $C(n)$ or less. Thus, $\alpha s K(n) a_0$ provides a suitable threshold on the number of connections between a node and the a_0 seed vertices.

Next, we denote by $\mathcal{N}^1(\alpha)$ and $\mathcal{N}^2(\alpha)$, respectively, the set of nodes from $\mathcal{G}_1$ and $\mathcal{G}_2$ that are classified as located in $\mathcal{H}_{\text{in}}(\alpha,\delta)$. By construction, we have $|\mathcal{N}^1(\alpha)| = \Theta(nC^k(n))$ and $|\mathcal{N}^2(\alpha)| = \Theta(nC^k(n))$. We build the pairs graph $\mathcal{P}(\mathcal{N})$ that is induced by the nodes of $\mathcal{G}_1$ and $\mathcal{G}_2$ that belong to, respectively, $\mathcal{N}^1(\alpha)$ and $\mathcal{N}^2(\alpha)$. While doing this, we can guarantee that a bad pair $[i_1, j_2]$ is included in $\mathcal{P}(\mathcal{N})$ only if either $[i_1, i_2]$ or $[j_1, j_2]$ are also included in $\mathcal{P}(\mathcal{N})$. This is accomplished as follows. We apply the previous classification procedure twice, using two different values α_1 and α_2, with $\alpha_1 > \alpha_2$, chosen in such a way that $\mathcal{H}_{\text{out}}(\alpha_1,\delta) \subseteq \mathcal{H}_{\text{in}}(\alpha_2,\delta)$. Then we insert in $\mathcal{P}(\mathcal{N})$ all pairs whose constituent nodes have been selected by at least one of the classification procedures, adding the constraint that at least one of the nodes must have been selected by both. Since, by construction, no good pair $[i_1, i_2]$ exists s.t. i_1 falls in $\mathcal{H}_{\text{in}}(\alpha_1,\delta)$ and i_2 in $\mathcal{H}_{\text{out}}(\alpha_2,\delta)$ (or vice versa), the above condition is ensured.

We then apply the PGM algorithm on $\mathcal{P}(\mathcal{N})$. Our goal is now to verify that the conditions in Theorem 1 hold so that, applying Corollary 1, we can claim that all good pairs in $\mathcal{P}(\mathcal{N})$ can be matched without errors. To this end, let us define $m = \Theta(nC^k(n))$, which in order sense equals the number of nodes in $\mathcal{N}^1(\alpha)$ and $\mathcal{N}^2(\alpha)$. Then note that $p_{\min} = \Theta(p_{\max})$, $p_{\max} = K(n)$ and $K(n) = o(m^{-\gamma})$. Thus, for a sufficiently large r, $p_{\max} \ll m^{-\frac{3.5}{r}}$. Furthermore, since by assumption $nC^k(n)K(n) = \Omega(\log n)$, it follows $p_{\min} \gg m^{-1}$. At last, it is easy to see that $a_o/a_c \to \infty$. Indeed, from (1), $a_c = O(1/K(n))$ while, by assumption (see Theorem 2), $a_0 = \Omega\left(\frac{\log(nC^k(n))}{K(n)}\right)$. In conclusion, we have that all good pairs whose nodes fall in $\mathcal{H}_{\text{in}}(\alpha_1,\delta)$ can be correctly matched.

To further expand the set of identified pairs, we pursuit the following simple approach. Starting from the pairs already matched in the first phase, which act as seeds, we consider a larger region that includes the previous one. By properly setting a threshold r, we can match all pairs in this larger region having at least r neighbors among the seeds. So doing, we successfully match w.h.p. all good pairs in the region with no errors. More formally, the following theorem allows us to claim that our approach can be successfully employed.

Theorem 3. *Consider a circular region $\mathcal{D}(0,\rho)$ centered at 0, of radius ρ, with $\rho \geq C(n)$. Given that all (or almost all) nodes lying within $\mathcal{D}(0,\rho)$ have been correctly identified, it is possible to correctly identify (almost) all nodes in $\mathcal{D}(0,\rho_1) \setminus \mathcal{D}(0,\rho)$ with probability $1-o(n^{-1})$, for $\rho_1 = \rho + C(n)/2$, when $K(n) = o((nC^k(n))^{-\gamma})$ for some $\gamma > 0$. In addition, none of the bad pairs formed by nodes in $\mathcal{H} - \mathcal{D}(0,\rho)$ will be identified, again with probability $1-o(n^{-1})$. This is done by setting threshold $r = \frac{n}{2}|\mathcal{D}(0,\rho) \cap \mathcal{D}(\mathbf{x},C(n))|\frac{K(n)}{2}$, with $|\mathbf{x}| = \rho_1$, and identifying as good pairs those in $\mathcal{H} \setminus \mathcal{D}(0,\rho)$ that have at least r neighbors among good pairs in $\mathcal{D}(0,\rho)$.*

PROOF. The proof is based on the application of standard concentration results, namely, Chernoff bound and inequalities in [11, p. 16] (also reported in B for convenience). The detailed proof is given in [10]. □

Almost all good pairs can be matched w.h.p. by iterating the matching procedure of Theorem 3 a number of steps $\Theta(1/C(n))$. Indeed, each time the PGM algorithm successfully matches all good pairs whose constituent nodes lie within distance $C(n)/2$ from the set of previously matched pairs. Note that Theorem 3 also guarantees that, jointly over all steps, no bad pair is matched w.h.p.

6. DENSE CLUSTERS

The case $K(n) = \omega((nC^k(n)^{-\gamma}))$, for any $\gamma > 0$, is significantly different from the previous case since the de-anonymization algorithm must disregard all edges whose length is too short (shorter than a properly defined threshold $\omega(C(n))$) so as to avoid errors (i.e., matching bad pairs). The approach we propose to address this case relies on some results that we introduce next, in an more abstract sense, considering the case in which $\mathcal{G}_T$ is a bipartite graph. Then we apply such results to our clustered social network model, and derive the minimum seed set size that is required to trigger the identification process in this case.

6.1 Results on bipartite graphs

Let $\mathcal{G}_T$ be a $m_l \times m_r$ bipartite graph. Let $\mathcal{M}_l$ denote the set of vertices on the left hand side (LHS), with $|\mathcal{M}_l| = m_l$, and $\mathcal{M}_r$ the set of vertices on the right hand side (RHS), with $|\mathcal{M}_r| = m_r$. We assume that for any pair of vertices $i \in \mathcal{M}_l$ and $j \in \mathcal{M}_r$ an edge (i, j) exists in the graph with probability p_{ij}, with $p_{\min} \leq p_{ij} \leq p_{\max}$ and $p_{\max} = \eta p_{\min}$ for some constant $\eta > 1$. Our goal is to identify a minimum number of seeds a_0 located in either side of the graph, i.e., with $a_0 = |\mathcal{A}_0^l|$ in $\mathcal{M}_l$ and $a_0 = |\mathcal{A}_0^r|$ in $\mathcal{M}_r$, such that vertices in $\mathcal{M}_l$ and $\mathcal{M}_r$ can be correctly matched.

Let us first consider the case where $m_l = m_r = m$, for which the theorem below holds.

Theorem 4. *Assume that $\mathcal{G}_T$ is an $m \times m$ bipartite graph and that two sets of seeds, $\mathcal{A}_0^l$ and $\mathcal{A}_0^r$, both of cardinality a_0, are available on, respectively, the LHS and the RHS of the graph. Then the PGM algorithm with threshold $r \geq 4$ correctly identifies $m - o(m)$ good pairs w.h.p. on the RHS and the LHS of graph $\mathcal{P}(\mathcal{G}_T)$, with no errors, under the same 4 conditions listed in Theorem 1.*

PROOF. See Appendix C. □

Theorem 4 can be extended to the general case where $m_l \neq m_r$, as stated in the corollary below.

Corollary 2. *Assume that $\mathcal{G}_T$ is an $m_l \times m_r$ bipartite graph and define $m = \min(m_l, m_r)$. Under the same assumptions of Theorem 4, the PGM algorithm with threshold $r \geq 4$ successfully identifies w.h.p. $m - o(m)$ good pairs on both the LHS and the RHS of $\mathcal{P}(\mathcal{G}_T)$, with no errors. Furthermore, the PGM algorithm can be successfully applied to an imperfect pairs graph $\hat{\mathcal{P}}(\mathcal{G}_T) \subset \mathcal{P}(\mathcal{G}_T)$ comprising a finite fraction of pairs on both the LHS and the RHS of $\mathcal{P}(\mathcal{G}_T)$ and satisfying the following constraint: a bad pair $[i_1, j_2] \in \mathcal{P}(\mathcal{G}_T)$ is included in $\hat{\mathcal{P}}(\mathcal{G}_T)$ only if either $[i_1, i_2]$ or $[j_1, j_2]$ are also in $\hat{\mathcal{P}}(\mathcal{G}_T)$.*

PROOF. The assertion can be proved by following the same arguments as in Theorem 4 and applying Corollary 1. □

Finally, we prove the following result, which shows that all good pairs can be matched with no errors w.h.p.

Theorem 5. *Consider that $\mathcal{G}_T$ is an $m_l \times m_r$ bipartite graph with $m_l = \omega(\sqrt{m_r})$ and that a seed set $\mathcal{A}_0^l$ is available on the LHS of the graph, with $|A_0^l| = a_0 = \Theta(m_l)$. With probability larger than $1 - e^{-\frac{m_l}{\sqrt{m_r}}}$, all the m_r good pairs on the RHS can be successfully identified with no errors, provided that:*

1. $\frac{1}{\sqrt{m_r}} \ll p_{\min} \leq p_{\max} \ll 1$
2. $p_{\min} = \Theta(p_{\max})$
3. *a matching algorithm is used on $\mathcal{P}(\mathcal{G}_T)$ that matches all pairs on the RHS that have at least r adjacent seeds on the LHS, with $r = a_0 \frac{p_{\min}}{2}$.*

The same result holds in case of imperfect pairs graph comprising a finite fraction of all possible pairs on the RHS.

PROOF. Without loss of generality, we assume $a_0 \geq cm_r$ for some $c > 0$. The proof is obtained by applying the inequalities reported in Appendix B and [11, p. 16]. First, observe that, given a good pair $[j_1, j_2]$ on the RHS of the pairs graph, the number of its adjacent seeds on the LHS is $E[N_g] \geq a_0 p_{\min} = 2r$. Thus, by applying inequality (8) and union bound, we have:

$$P(\text{all good pairs on the RHS have at least } r \text{ adjacent seeds}) \geq 1 - m_r e^{-cm_l p_{\min} H(\frac{1}{2})} \geq 1 - e^{-\frac{m_l}{\sqrt{m_r}}}$$

which imply that all good pairs on the RHS are successfully matched since $m_l = \omega(\sqrt{m_r})$. Similarly, considering a bad pair $[j_1, k_2]$ on the RHS, the number of its adjacent seeds on the LHS is $E[N_b] \leq cm_r(p_{\max})^2 \ll r$. Thus, by applying inequality (10) and union bound, we have:

$$P(\text{all bad pairs on the RHS have less than } r \text{ adjacent seeds}) \geq 1 - m_r^2 e^{-cm_l \frac{p_{\min}}{4} \log\left(\frac{p_{\min}}{(p_{\max})^2}\right)} \geq 1 - e^{-\frac{m_l}{\sqrt{m_r}}}.$$

□

6.2 The de-anonymization procedure

We now outline how our proposed matching algorithm for the *dense clusters* case works. First, we consider two hyper-cubic regions, $\mathcal{H}_l \subset \mathcal{H}$ and $\mathcal{H}_r \subset \mathcal{H}$, whose side is $h(n) = \Omega(C(n))$ and whose distance is $g(n) = \omega(C(n))$ (see Fig. 3). Note that by construction, given two vertices $i \in \mathcal{H}_l$ and $j \in \mathcal{H}_r$, $p_{\min} = K(n)f(g(n) + \sqrt{k}h(n)) \leq p_{ij} \leq K(n)f(g(n)) = p_{\max}$. Let us assume $p_{\max} = \eta p_{\min}$ for some constant $\eta > 1$.

We then extract vertices in $\mathcal{H}_l$ and $\mathcal{H}_r$ from the rest of vertices so that we can focus on the bipartite graph induced by the nodes in the two sub-regions, along with the edges between them. To this end, we assume that two sufficiently large sets of seeds are available in $\mathcal{H}_l$ and $\mathcal{H}_r$ so that Theorem 2 can be applied. In this regard, observe that we can use the same procedure as in Section 5, to make sure that a bad pair $[i_1, j_2]$ is included in the pair graph only if either $[i_1, i_2]$ or $[j_1, j_2]$ are also included in it. We can then apply Corollary 2.

It follows that the execution of the PGM algorithm ensures that almost all of the good pairs in either the LHS or the RHS of the pairs graph are correctly de-anonymized. Without lack of generality, we assume that almost all pairs on LHS are de-anonymized, i.e., $m_l < m_r$, and that a non-negligible fraction of the good pairs on the RHS have still to be identified. Then the rest of good pairs on the RHS can be matched by applying Theorem 5.

To further expand the set of matched nodes, we first show how it is possible to estimate (at least in order sense) the length of the edges between two nodes, again by exploiting the dense structure of the clusters.

Proposition 1. *Given two nodes in region $\mathcal{H}$, it is possible to estimate with arbitrary precision their mutual distance d as far as $d \ll C(n)\left(nK^2(n)C^k(n)\right)^{\frac{1}{\beta}}$.*

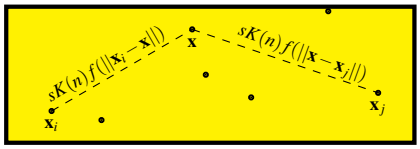

Figure 5: Computation of $\mathbb{E}[N_{ij}]$.

PROOF. Let us consider two nodes i and j on $\mathcal{G}_1$ ($\mathcal{G}_2$) whose mutual distance is d_{ij}. Let N_{ij} be the variable that represents the number of their common neighbors. By construction (see Fig. 5), we have:

$$\begin{aligned}\mathbb{E}[N_{ij}] &= (n-2)s^2K^2(n)\int_{\mathcal{H}} f(||\mathbf{x}-\mathbf{x}_i||)f(||\mathbf{x}-\mathbf{x}_j||)\mathrm{d}\mathbf{x} \\ &= \Theta(nC^k(n)K^2(n)f(d_{ij}))\,.\end{aligned}$$

Observe that $\mathbb{E}[N_{ij}]$ is continuous and strictly decreasing with d_{ij}, and thus invertible. Now, applying Chernoff bound we can show that for any $0<\delta<1$

$$\mathbb{P}\left(\frac{|N_{ij}-\mathbb{E}[N_{ij}]|}{\mathbb{E}[N_{ij}]}>\delta\right)\leq e^{-c(\delta)\mathbb{E}[N_{ij}]}$$

for a proper constant $c(\delta)>0$. Since $\mathbb{E}[N_{ij}]\to\infty$ as long as $d \ll C(n)\left(nK^2(n)C^k(n)\right)^{\frac{1}{\beta}}$, the assertion follows. □

We can therefore use the number of common neighbors between two given nodes as an estimator of their distance. We then set two thresholds, $d_L=\Theta(C(n)\log(n^{1/k}C(n)))$ and $d_H=\lambda d_L$ (with $\lambda>1$), and we leverage the above result to correctly classify the edges going out of previously matched nodes into three categories: edges that are shorter than d_L, edges that are longer than d_H and edges of length comprised between d_L and d_H. In particular, we are interested in the latter, for which the following result holds.

Proposition 2. *Assume $K(n)=\omega((nC^k(n))^{-\gamma})$, $\forall\gamma>0$. Consider a set comprising a finite fraction of the nodes in $\mathcal{G}_1$ ($\mathcal{G}_2$) lying in a region of side $\Theta(C(n))$, and the edges incident to them. For an arbitrarily selected $\delta>0$, w.h.p (i.e., with a probability larger than $1-[C(n)]^k$) we can select all edges whose length d is $(1+\delta)d_L\leq d\leq(1-\delta)d_H$. Furthermore, no edges whose length $d<(1-\delta)d_L$ and $d>(1+\delta)d_H$ are selected.*

The proof follows the same lines as in the proof in Appendix A (see [10] for further details).

At this point, we consider a bipartite graph whose LHS is still represented by $\mathcal{H}_l$, and whose RHS is given by the nodes that are connected with those in $\mathcal{H}_l$ through edges of length comprised between d_L and d_H. We can therefore apply Theorem 5 and match w.h.p. all good pairs on the RHS, with no errors. The procedure is then iterated so as to successfully de-anonymize the entire network. Note that, at every step we apply the following proposition to extract a group of matched nodes whose mutual distance is $\Theta(C(n))$.

Proposition 3. *Assume $K(n)=\omega((nC^k(n))^{-\gamma})$ $\forall\gamma>0$. Given a node i, we can set a threshold $d_T=\Theta(C(n))$ and select all nodes in $\mathcal{G}_1$ ($\mathcal{G}_2$) whose estimated distance from i is less than d_T. So doing, for an arbitrarily selected $\delta>0$, we successfully select with a probability larger than $1-[C(n)]^k$ all nodes whose real distance is $d\leq(1-\delta)d_T$. Furthermore, no nodes whose distance from i is $d>(1+\delta)d_T$ are selected by our algorithm.*

The proof is similar to that of Proposition 2 (see also [10]).

6.3 Minimum seed set size

To explicitly derive the minimum seed set size, we need to further specify $h(n)$ and $g(n)$, which are to be carefully selected so as to minimize the resulting critical size a_c in Theorem 4 and Corollary 2.

Starting from the result provided by Theorem 4, a_c can be written as:

$$\begin{aligned}a_c &= \left(1-\frac{1}{r}\right)\left(\frac{(r-1)!}{m(p_{\min}s^2)^r}\right)^{\frac{1}{r-1}} \\ &\leq \left(\frac{r-1}{(m(p_{\min}s^2)^{\frac{1}{r-1}}p_{\min}s^2}\right)\leq\frac{r-1}{p_{\min}s^2}\,. \qquad (5)\end{aligned}$$

The above expression can be minimized by maximizing $p_{\min}$, i.e., by minimizing $g(n)$ (recall that $p_{\min}=K(n)f(g(n)+\sqrt{kh(n)})$). However, $g(n)$ and $h(n)$ must also be selected in such a way that condition 1) of Theorem 4 is met. Additionally, as mentioned, it must be ensured that $h(n)=\Omega(C(n))$. At last, by standard concentration results, m_l and m_r turn out to be both $\Theta(nh^k(n))$ provided that $h(n)\geq(\log n/n)^{1/k}$.

Previous considerations suggest to fix $h(n)=\Theta(C(n))\geq(\log n/n)^{1/k}$ (i.e., the minimum possible value in order sense), which corresponds to having $m=\Theta(nC^k(n))$ (recall that $m=\min(m_l,m_r)$). We then derive $g(n)$ by forcing $p_{max}\approx m^{-\frac{\alpha}{r}}$, with $3.5<\alpha<4$ and $r\geq4$. Note that condition 1) of Theorem 4 is met since $p_{\max}$ and $p_{\min}$ are both $\Theta(m^{-\frac{\alpha}{r}})$. Hence, we have $p_{\max}=\Theta((nC^k(n))^{-\frac{\alpha}{r}})$ and $g(n)=\Theta(n^{\frac{\alpha}{\beta r}}[C(n)]^{1+\frac{\alpha k}{\beta r}}[K(n)]^{\frac{1}{\beta}}))$.

Given the above expression for p_{max}, considering that $p_{max}=\eta p_{min}$ and using (5), the minimum seed set size can be made as small as $a_c=O([nC^k(n)]^{\varepsilon})$, for any $\varepsilon>0$, by choosing $r>\frac{4}{\varepsilon}$. Finally, we remark that the obtained a_c is in order sense greater than the minimum number of seeds needed to apply Theorem 2 while selecting nodes in regions $\mathcal{H}_l$ an $\mathcal{H}_r$, thus the whole construction is consistent.

7. EXPERIMENTAL VALIDATION

Although our results hold asymptotically as $n\to\infty$, we can expect to qualitatively observe the main effects predicted by the analysis also in finite-size graphs. We will first investigate the performance of graph matching algorithms in synthetic graphs generated according to our model of clustered networks, and then apply them to a real social network graph.

7.1 Synthetic graphs

In this section we consider bi-dimensional graphs having $n=10,000$, the sampling probability $s=0.8$ and, unless otherwise specified, the average node degree in the ground-truth graph $D(n)=30$.

Fig. 6 reports the average number of correctly matched nodes across 1,000 runs of the PGM algorithm (using $r=5$) in various cases, as function of the number of seeds. In each run, seeds are either chosen uniformly at random among all nodes (label 'uniform seeds'), or as a compact set around one randomly chosen seed (label 'compact seeds'). In our model of clustered graphs, we have fixed $\beta=3$ (the decay exponent of the edge probability beyond $C(n)$), and we consider either $K(n)=0.05$ or $K(n)=0.2$. As reference, in the plot we also show the phase transition occurring (at about 600 seeds) when $\mathcal{G}_T$ is a $G(n,p)$ graph having the same average node degree. The plot confirms the wave-like nature of the identification process as predicted by our analysis, namely: i) clustered networks (larger $K(n)$) can be matched starting from a much

smaller seed set as compared to $G(n,p)$; ii) such huge reduction requires seeds to be selected within a small sub-region of $\mathcal{H}$.

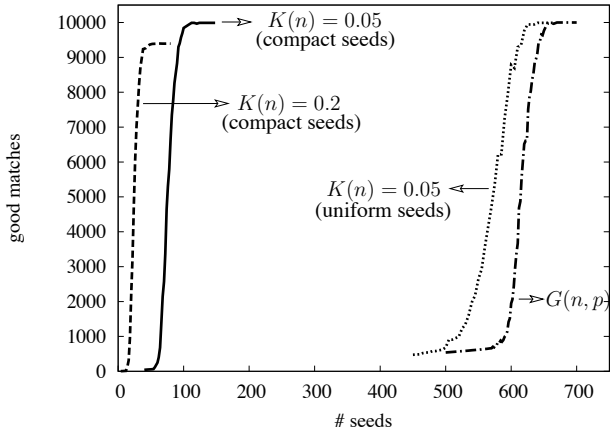

Figure 6: Comparison of PGM performance (with $r = 5$) in different networks with $n = 10,000$. Number of good matches (averaged over 1,000 runs) as a function of the number of seeds, chosen either uniform or compact.

What the plot in Fig. 6 does not clearly show (except for a rough estimate based on the maximum number of correctly matched nodes) is the error ratio incurred by the PGM algorithm, which is expected to become larger and larger as we increase the level of clustering in the network. This phenomenon is confirmed by Fig. 7, which reports the average error ratio (bad matches over all matches) incurred by PGM as a function of $K(n)$, starting from a compact set of seeds. In Fig. 7 we have considered also different values of β. The little circle denotes the operating point already considered for the left-most curve in Fig. 6 ($K(n) = 0.2$), having an error ratio of about 5%. The plot reveals that the error ratio increases dramatically when $K(n)$ tends to 1, confirming that PGM cannot be safely applied in highly clustered networks. The effect of β is more intriguing: smaller β's produce fewer errors since generated network graphs tend to become more similar to $G(n,p)$, where PGM is known to generate very few errors. As side-effect, smaller values of β tend to slightly increase the percolation threshold (not shown in the plot). For example, for $K(n) = 0.4$, the critical number of seeds (estimated from simulations) corresponding to β = 2.2,2.5,3,4 are equal to 11,15,24,45, respectively.

Next, we focus on the 'hard' case corresponding to the little square shown in Fig. 7, i.e., $K(n) = 0.8$, $\beta = 3$. This case corresponds to networks having highly dense clusters, where the performance of the original PGM algorithm is rather poor (error ratio about 50%). Fig. 8 shows the average number of nodes matched by different algorithms as a function of the number of seeds: thick lines correspond to good matches, whereas thin lines (with the same line style) refer to bad matches produced by the same algorithm. For sake of simplicity, network de-anonymization is performed by applying a simplified version of the algorithm proposed and analysed in Section 6. This simple algorithm consists in adopting PGM after having removed all graph edges shorter than $x \cdot C(n)$. In the following, we will call this algorithm 'filtered PGM' and we will label the corresponding curves in the plots by '$f =<x>$'. We stress that filtered PGM approaches the performance that can be obtained by the algorithm in Section 6.

Looking at Fig. 8, it is important to remark that in this scenario the performance of the various algorithms is highly sensitive to the location of the set of seeds (in each run we uniformly select one

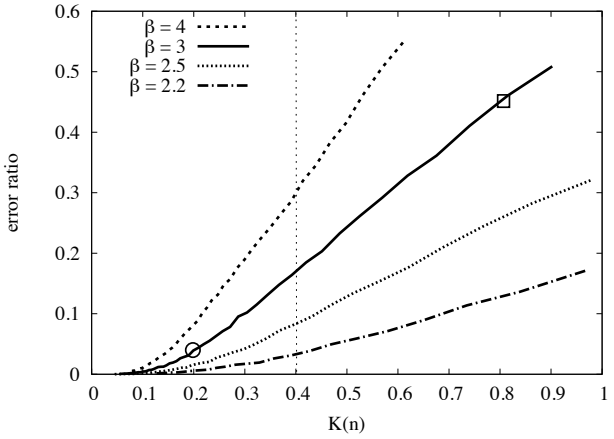

Figure 7: Error ratio of PGM as a function of $K(n)$ for different values of β, starting from compact seeds.

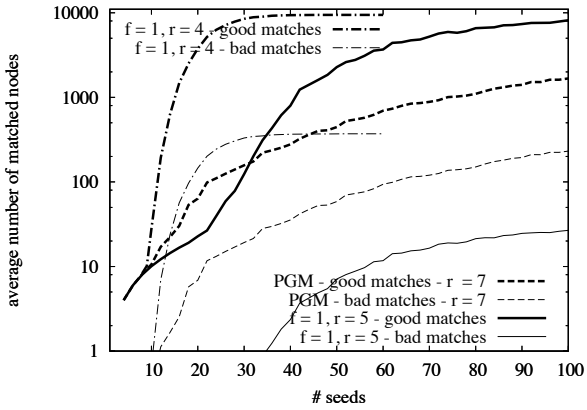

Figure 8: Average number of good and bad pairs matched by different Numbers close to the vertical line at $K(n) = 0.4$ denote corresponding estimates of the percolation thereshold derived via simulation.

seed among all nodes, and choose all of the other seeds among its neighbors). Since we average the results over 1,000 runs, this explains why all curves do not exhibit a sharp transition[4]. An average number of matched nodes equal to, say, 2,000, must be given the following probabilistic interpretation: about 1/5 of (uniformly chosen) initial locations allow us to match almost all nodes (10,000), while 4/5 of initial locations do not trigger the percolation effect.

Also, we note that the poor performance of standard PGM cannot be fixed by just increasing the threshold r: using $r = 7$, PGM still produces about 12% error ratio, while also requiring a disproportionally larger number of seeds (only about 2,000 nodes are matched on average starting from 100 seeds). Instead, filtered PGM, with $f = 1$ and $r = 4$, requires very few seeds to match almost all nodes, incurring about 3.7% error ratio. Using $f = 1$, $r = 5$, filtered PGM requires more seeds, but achieves as low as 0.3% error ratio.

Next, we fix r and increase the filtering factor f so as to diminish the number of errors while, however, reducing the average number of matched nodes (i.e., the probability to trigger percolation from a

[4]We verified that, if we instead fix the very first seed across all runs, a sharp transition appears. The transition threshold changes as we vary the initial seed (results not shown here).

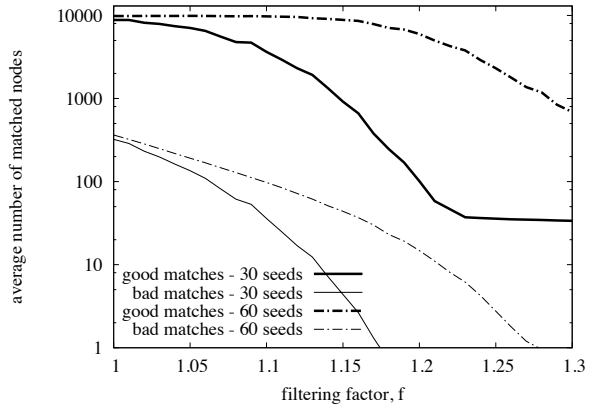

Figure 9: Effect of varying the filtering factor f for fixed $r = 4$ (scenario with $K(n) = 0.8$).

Table 3: Combinations of parameters achieving error ratio 3%, percolation probability 50%

average node degree	f	# seeds
36	1.1	22
45	1.2	24
53	1.3	28
64	1.4	32

given seed set). Fig. 9 illustrates this effect for $r = 4$, in the case of two different seed set sizes, 30 and 60. Having 60 seeds one could, for example, employ $f = 1.1$ obtaining very high chance of percolation (almost 100%) and small error ratio (around 1%).

Alternately, we can fix a desired error ratio and average number of matched nodes (i.e., the probability to trigger large-scale percolation), and look for the filtering factor and number of seeds that let us achieve the desired goals. Table 3 reports an example of this numerical exploration, in which we vary the average degree of the nodes in $\mathcal{G}_T$ corresponding to each examined scenario (the average degree can be increased, for fixed $K(n) = 0.8$, by increasing $C(n)$). The results in Table 3 validate, at least qualitatively, the counter-intuitive theoretical predictions in Table 2: as we increase $C(n)$ (and thus the average node degree), the seed set size necessary to achieve a desired matching performance increases as well.

7.2 Real social graph

We consider a real graph derived from the Slovak social network Pokec.The public data set, available at [12], is a directed graph with 1,632,803 vertices and 30,622,564 edges, where nodes are users of Pokec and directed edges represent friendships. Since the original graph contains too many vertices for our computational power, and since we would like to isolate the impact of clustering from the effect of long-tailed degree distributions, we considered only vertices having: i) in-degree larger than 20; ii) out-degree smaller than 200. We ended up with a reduced graph having $n = 133,573$ nodes, 5,449,236 edges, average (in or out) degree 40.8 and clustering coefficient 0.11. We use this graph as our ground-truth, and employ an edge sampling probability $s = 0.8$. Notice that we maintain the direct nature of the edges, since all considered algorithms immediately apply to direct networks as well [5].

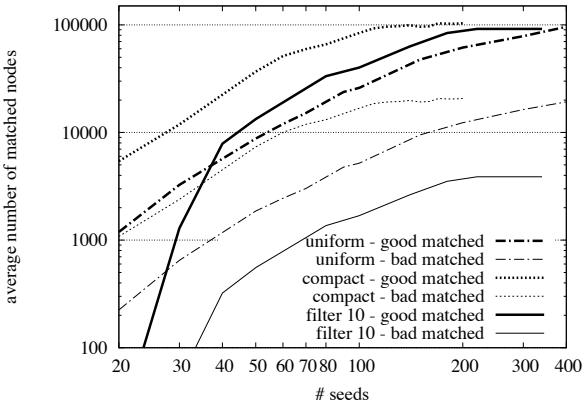

Figure 10: Performance of matching algorithms in a subset of the friendship graph of the social network Pokec.

Fig. 10 shows the performance of the different algorithms using threshold $r = 6$. As before, curves labelled 'uniform' refer to the PGM algorithm in which seeds are selected uniformly at random among the nodes. Curves labelled 'compact' refer to the PGM algorithm in which seeds are chosen among the closest neighbors of a uniformly selected node. Curves labelled 'filter 10' differ from the previous one in that the edges connecting each node to its nearest 10 neighbors are not used by the algorithm. We emphasize that a $G(n,p)$ having the same number of nodes and average degree would require $a_c = 5,783$ seeds, according to (1). In contrast, all considered algorithms require much fewer seeds to match almost all nodes, confirming that real social networks are much simpler to de-anonymize than $G(n,p)$. In particular, the uniform variant requires about 400 seeds to match on average 100,000 nodes, but incurs a quite large error ratio (about 17%). The compact variant reduces this number roughly by a factor 3, but produces the same error ratio. At last, the filtered variant requires a bit more seeds than the compact one, but it allows to lower down the error ratio to about 4%. The above results confirm the crucial performance improvement that can be obtained by jointly: i) starting from a compact set of seeds (to exploit the wave-propagation effect), ii) carefully discarding edges connecting nodes to their local clusters (to limit the errors).

8. CONCLUSIONS

We focused on the effect of node clustering on social network de-anonymization. We defined a flexible model of geometric random graphs that can incorporate different levels of clustering. Then we designed de-anonymization algorithms and analysed their performance by using bootstrap percolation. Our theoretical results highlight that clustering significantly helps to reduce the number of seeds required to trigger the identification process, and that our algorithms can correctly match almost all nodes while making errors negligible (asymptotically as the network grows large). Our findings were confirmed by numerical experiments on synthetic and real social graphs.

[5]In direct networks, counters of matchable pairs are incremented only by using outgoing edges from matched pairs.

APPENDIX

A. PROOF OF THEOREM 2

Without loss of generality, let us focus on $\mathcal{G}_1$ and let us consider a node $i \in \mathcal{H}_{\text{in}}(\alpha,\delta)$. By construction, the number of seeds that are neighbors of i on $\mathcal{G}_1$ is given by $S_i = \sum_{\sigma\in\mathcal{A}_0} X_{i\sigma} S^1_{i\sigma} \geq_{st} Y_i \geq_{st} Y$ where

$$Y_i = \text{Bin}(a_0, sK(n) f(\max_{\sigma\in\mathcal{A}_0} ||\mathbf{x}_i - \mathbf{x}_\sigma||))$$

and $Y = \text{Bin}(a_0, sK(n)(1+\delta)\alpha)$, with $\mathbb{E}[Y] = sK(n)(1+\delta)\alpha a_0$. Now, using the inequalities in Appendix B, we can bound:

$$P(Y_i < \alpha sK(n)a_0) \leq \exp\left(-\mathbb{E}[Y_i] H\Big(\frac{\alpha sK(n)a_0}{\mathbb{E}[Y_i]}\Big)\right) \leq \exp\left(-(1+\delta)\alpha sK(n)a_0 H\Big(\frac{1}{1+\delta}\Big)\right) \quad (6)$$

with $H(b) = 1-b+b\log b$.

If we consider jointly all nodes in $\mathcal{H}_{\text{in}}(\alpha,\delta)$ and we denote with N_{in} their number, we can bound the probability that every node in $\mathcal{H}_{\text{in}}(\alpha,\delta)$ is accepted with:

$$P(\text{all nodes in } \mathcal{H}_{\text{in}} \text{ are accepted} \mid N_{\text{in}}) \leq 1 - N_{\text{in}} \exp\left(-(1+\delta)\alpha sK(n)a_0 H\Big(\frac{1}{1+\delta}\Big)\right), \quad (7)$$

with (7) that tends to 1 if $\log N_{\text{in}} - (1+\delta)\alpha s H\Big(\frac{1}{1+\delta}\Big)K(n)a_0 \to -\infty$. This can be enforced by opportunely setting $a_0 = \Omega\left(\frac{\log N_{\text{in}}}{K(n)}\right)$. Since by construction $|\mathcal{H}_{\text{in}}| > C^k(n) \geq \frac{\log n}{n}$, we have w.h.p. $N_{\text{in}} \leq 2n|\mathcal{H}_{\text{in}}|$ by standard concentration results (see also [10, Lemma 2]). As consequence,

$$P(\text{all vertices in } \mathcal{H}_{\text{in}} \text{ are accepted}) \to 1$$

provided that $a_0 = \Omega\left(\frac{\log(nC^k(n))}{K(n)}\right)$. Then we focus on the nodes in $\mathcal{H}_{\text{out}}(\alpha,\delta)$ and we show that all those nodes are jointly rejected. Conceptually we repeat the same approach as before, however, the argument is made slightly more complex by the fact that, to achieve tight bounds on the probability that all nodes in $\mathcal{H}_{\text{out}}(\alpha,\delta)$ are jointly rejected, we need to partition $\mathcal{H}_{\text{out}}(\alpha,\delta)$ into smaller sub-regions containing nodes which lie at similar distance from the seeds.

Assuming $\delta < \frac{e^2-1}{e^2}$, we define $\mathcal{H}^1_{\text{out}} = \mathcal{H}^1(\alpha, \frac{e^2-1}{e^2}) \subset \mathcal{H}_{\text{out}}(\alpha,\delta)$ and $\mathcal{H}^0_{\text{out}}(\alpha,\delta) = \mathcal{H}_{\text{out}}(\alpha,\delta) \setminus \mathcal{H}^1_{\text{out}}$. Furthermore, we partition $\mathcal{H}^1_{\text{out}}$ into disjoint sub-regions, i.e., $\mathcal{H}^1_{\text{out}} = \cup_{h\geq 1} \mathcal{H}^{1,h}_{\text{out}}$, with

$$\mathcal{H}^{1,h}_{\text{out}} = \mathcal{H}_{\text{out}}(\frac{\alpha, h^\beta e^2 - 1}{h^\beta e^2}) \setminus \mathcal{H}_{\text{out}}(\alpha, \frac{(h+1)^\beta e^2 - 1}{(h+1)^\beta e^2})$$

Now, given a vertex i in $\mathcal{H}^0_{\text{out}}$ ($\mathcal{H}^{1,h}_{\text{out}}$), the number of its neighbor seeds S_i on $\mathcal{G}_1$ can be bounded from above by a $\text{Bin}(a_0, sK(n)(1-\delta)\alpha)$ $\left(\text{Bin}(a_0, \frac{sK(n)}{h^\beta e^2}\alpha)\right)$. Furthermore, by elementary geometrical arguments, it can be shown that: i) $|\mathcal{H}^0_{\text{out}}| = \Theta(C^k(n))$, ii) $|\mathcal{H}^{1,1}_{\text{out}}| = \Theta(C^k(n))$ and iii) $\mathcal{H}^{1,h}_{\text{out}} = \Theta(h^{k-1}\mathcal{H}^{1,1}_{\text{out}})$.

Denoted with N^0_{out} and $N^{1,h}_{\text{out}}$ the number of nodes in $\mathcal{H}^0_{\text{out}}$ and $\mathcal{H}^{1,h}_{\text{out}}$, respectively, by exploiting again the inequalities in [11, pag 16], w.h.p. we have:

$$P\left(\text{all nodes in } \mathcal{H}^0_{\text{out}} \text{ are rejected}\right) \leq 1 - N^0_{\text{out}} \exp\left(-(1-\delta)\alpha sK(n)a_0 H\Big(1-\delta\Big)\right) \to 1.$$

The above expression holds under the assumption that $a_0 = \Omega\left(\frac{\log(nC^k(n))}{K(n)}\right)$. Indeed, we remark that $N^0_{\text{out}} \leq 2n|\mathcal{H}^0_{\text{out}}| = \Theta(nC^k(n))$ w.h.p. At last,

$$P\left(\text{all nodes in } \mathcal{H}^1_{\text{out}} \text{ are rejected}\right) \leq 1 - \sum_{h=1}^{\infty} N^{1,h}_{\text{out}} \exp\left(-\frac{\alpha sK(n)a_0}{2}(\beta\log h + 2)\right).$$

For every h, $N^{1,h}_{\text{out}} \leq 2n|\mathcal{H}^{1,h}| = \Theta(nh^{k-1}C^k(n))$; also, the number of sub-regions of $\mathcal{H}^1_{\text{out}}$ is $O(n/C^k(n))$. Thus, w.h.p. we have that jointly on all h's, the number of nodes in these sub-regions can be bounded by $2n|\mathcal{H}^{1,h}|$. Under the assumption that $a_0 = \Omega\left(\frac{\log(nC^k(n))}{K(n)}\right)$, it can be easily shown that

$$P\left(\text{all nodes in } \mathcal{H}^1_{\text{out}} \text{ are rejected}\right) \to 1.$$

B. CONCENTRATION INEQUALITIES

For the reader's convenience, we report below the inequalities that can be found also in [11, p. 16].

Lemma 1. *Let $H(b) = 1-b+b\log b$ for $b > 0$. Suppose $n \in \mathbb{N}$ $p \in (0,1)$ and $0 \leq k \leq n$. Let $\mu = np$; if $k \leq \mu$, then:*

$$P(Bin(n,p) \leq k) \leq \exp\left(-\mu H\Big(\frac{k}{\mu}\Big)\right) \quad (8)$$

if $k > \mu$, then:

$$P(Bin(n,p) \geq k) \leq \exp\left(-\mu H\Big(\frac{k}{\mu}\Big)\right) \quad (9)$$

if $k > e^2\mu$, then

$$P(Bin(n,p) \geq k) \leq \exp\left(-\frac{k}{2}\log\frac{k}{\mu}\right). \quad (10)$$

Algorithm 1 The PGM algorithm

1: $\mathcal{A}_0 = \mathcal{B}_0 = \mathcal{A}_0(n)$, $\mathcal{Z}_0 = \emptyset$
2: **while** $\mathcal{A}_t \setminus \mathcal{Z}_t \neq \emptyset$ **do**
3: $t = t+1$
4: Randomly select a pair $[*_1, *_2] \in \mathcal{A}_{t-1} \setminus \mathcal{Z}_{t-1}$ and add one mark to all neighboring pairs of $[*_1, *_2]$ in $\mathcal{M}(\mathcal{G}_T)$.
5: Let $\Delta\mathcal{B}_t$ be the set of all neighboring pairs of $[*_1, *_2]$ in $\mathcal{M}(\mathcal{G}_T)$ whose mark counter has reached threshold r at time t.
6: Construct set $\Delta\mathcal{A}_t \subseteq \Delta\mathcal{B}_t$ as follows. Order the pairs in $\Delta\mathcal{B}_t$ in an arbitrary way, select them sequentially and test them for inclusion in ΔA_t:
7: **if** the selected pair in $\Delta\mathcal{B}_t$ has no conflicting pair in $\mathcal{A}_{t-1}$ or $\Delta\mathcal{A}_t$ **then**
8: Insert the pair in $\Delta\mathcal{A}_t$
9: **else**
10: Discard it
11: $\mathcal{Z}_t = \mathcal{Z}_{t-1} \cup [*_1, *_2]$, $\mathcal{B}_t = \mathcal{B}_{t-1} \cup \Delta\mathcal{B}_t$, $\mathcal{A}_t = \mathcal{A}_{t-1} \cup \Delta\mathcal{A}_t$
12: **return** $T = t$, $\mathcal{Z}_T = \mathcal{A}_T$

C. PROOF OF THEOREM 4

The following proof uses the PGM algorithm that has been introduced in [6] and here is reported for completeness in Alg. 1. The notation is briefly explained below; the reader may also refer to [10] for a detailed description of the PGM algorithm and associated notation.

With reference to PGM algorithm, we define:

- $\mathcal{B}_t(\mathcal{G}_\mathrm{T})$ as the set of pairs in $\mathcal{P}(\mathcal{G}_\mathrm{T})$ that at time step t have already collected a least r marks. It is composed of good pairs $\mathcal{B}'_t(\mathcal{G}_\mathrm{T})$ and bad pairs $\mathcal{B}''_t(\mathcal{G}_\mathrm{T})$;
- $\mathcal{A}_t(\mathcal{G}_\mathrm{T})$ as the set of matchable pairs at time t. Similarly to $\mathcal{B}_t(\mathcal{G}_\mathrm{T})$, it comprises good pairs $\mathcal{A}'_t(\mathcal{G}_\mathrm{T})$ and bad pairs $\mathcal{A}''_t(\mathcal{G}_\mathrm{T})$. In general, $\mathcal{A}_t(\mathcal{G}_\mathrm{T})$ and $\mathcal{B}_t(\mathcal{G}_\mathrm{T})$ do not coincide as $\mathcal{B}_t(\mathcal{G}_\mathrm{T})$ may include conflicting pairs that are not present in $\mathcal{A}_t(\mathcal{G}_\mathrm{T})$;
- $\mathcal{Z}_t(\mathcal{G}_\mathrm{T})$ as the set of pairs that have been matched up to time t. By construction, $|\mathcal{Z}_t(\mathcal{G}_\mathrm{T})| = t, \forall t$.

For the sake of readability, below we omit the dependency on the $\mathcal{G}_\mathrm{T}$.

For any two vertices $i \in \mathcal{M}_l$ and $j \in \mathcal{M}_r$, let X_{ij} be the Bernoulli random variable that represents the presence of an edge $(i,j) \in \mathcal{E}$. By construction, $Ber(p_{\min}) \leq_{st} X_{ij} \leq_{st} Ber(p_{\max})$. I.e., two variables $\underline{X}_{ij}$ and $\overline{X}_{ij}$, with distribution, respectively, $Ber(p_{\min})$ and $Ber(p_{\max})$, can be defined on the same probability space as X_{ij} such that $\underline{X}_{ij} \leq X_{ij} \leq \overline{X}_{ij}$ point-wise.

We consider the corresponding pairs graph $\mathcal{P}(\mathcal{G}_\mathrm{T})$, which is, by construction, composed of all the pairs of vertices residing in $\mathcal{M}_l$ and $\mathcal{M}_r$ and of the edges connecting pairs of vertices in $\mathcal{M}_l$ with pairs of vertices in $\mathcal{M}_r$. We denote by $\mathcal{P}_l$ and $\mathcal{P}_r$, respectively, the set of pairs of $\mathcal{P}(\mathcal{G}_\mathrm{T})$, whose vertices lie in $\mathcal{M}_l$ and $\mathcal{M}_r$. Observe that, given two good pairs $[i_1,i_2] \in \mathcal{P}_l$ and $[j_1,j_2] \in \mathcal{P}_r$, the presence of an edge in $\mathcal{P}(\mathcal{G}_\mathrm{T})$ is associated with the random variable:

$$Y_{[i_1,i_2],[j_1,j_2]} = X_{ij}X_{ij}S^1_{ij}S^2_{ij} = X^2_{ij}S^1_{ij}S^2_{ij}$$

where S^1_{ij} and S^2_{ij} are mutually independent $Ber(s)$ random variables, which are in turn independent of X_{ij}. By construction,

$$p_{\min}s^2 \leq \mathbb{E}[Y_{[i_1,i_2],[j_1,j_2]}] \leq p_{\max}s^2.$$

Instead, given two bad pairs $[i_1,k_2] \in \mathcal{P}_l$ and $[j_1,l_2] \in \mathcal{P}_r$, we have $Y_{[i_1,k_2],[j_1,l_2]} = X_{ij}X_{kl}S^1_{ij}S^2_{kl}$, with $p^2_{\min}s^2 \leq \mathbb{E}[Y_{[i_1,k_2],[j_1,l_2]}] \leq p^2_{\max}s^2$. Finally, if we consider one good pair and one bad pair (e.g., $[i_1,i_2] \in \mathcal{P}_l$ and $[j_1,k_2] \in \mathcal{P}_r$), we obtain $Y_{[i_1,i_2],[j_1,k_2]} = X_{ij}X_{ik}S^1_{ij}S^2_{ik}$, with $p^2_{\min}s^2 \leq \mathbb{E}[Y_{[i_1,i_2],[j_1,j_2]}] \leq p^2_{\max}s^2$.

Recall that we assume that two seed sets, $\mathcal{A}^l_0 \in \mathcal{P}_l$ and $\mathcal{A}^r_0 \in \mathcal{P}_r$ (with $|\mathcal{A}^l_0| = |\mathcal{A}^r_0|$), are available. On $\mathcal{P}(\mathcal{G}_\mathrm{T})$ we run the PGM algorithm [6], opportunely modified, as follows. At every time step t, we extract uniformly at random one pair $\mathbf{z}^l(t) = [z^l_1, z^l_2]_t \in \mathcal{A}^l_{t-1} \setminus \mathcal{Z}^l_{t-1}$ and $\mathbf{z}^r(t) = [z^r_1, z^r_2]_t \in \mathcal{A}^r_{t-1} \setminus \mathcal{Z}^r_{t-1}$, adding a mark to all the neighbor pairs in $\mathcal{P}_r$ and $\mathcal{P}_l$, respectively. In other words, matched pairs in $\mathcal{P}_l$ contribute to the mark of pairs in $\mathcal{P}_r$ and vice versa. Thus, for a generic node pair $[i_1,j_2] \in \mathcal{P}_r \setminus \mathcal{Z}^r_t$, marks are updated according to the iteration: $M^r_{[i_1,j_2]}(t) = M^r_{[i_1,j_1]}(t-1) + Y_{\mathbf{z}^l(t),[i_1,j_2]}$. Similarly, for $[i_1,j_2] \in \mathcal{P}_l$ marks are updated according to $M^l_{[i_1,j_2]}(t) = M^l_{[i_1,j_2]}(t-1) + Y_{[i_1,j_2],\mathbf{z}^r(t)}$. For the rest, the algorithm proceeds exactly as described in Section 2.

Now, it is important to observe that marks of pairs on the RHS of the graph evolve exactly as the marks of a coupled PGM that operates over a pairs graph $\mathcal{P}_R$ defined as follows. Denote the generic pair by $[*_1,*_2]$; then $\mathcal{P}_R$ is a graph insisting on the set of nodes $\mathcal{M}_r$ and in which the presence of edge $(\mathbf{z}^r(t),[*_1,*_2])$, for any $[*_1,*_2] \in \mathcal{P}_r \setminus \mathcal{Z}^r_t$, is dynamically unveiled at time t by observing variable $X_{z^l_1(t)*_1}X_{z^l_2(t)*_2}S^l_{z^l_1(t)*_1}S^r_{z^l_1(t)*_2}$. In other words, the edges originated from $\mathbf{z}^l(t)$ are replaced by the edges originated from $\mathbf{z}^r(t)$ and vice versa.

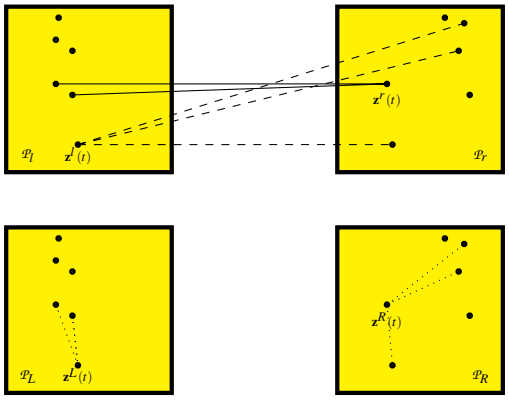

Figure 11: Graphical representation of the PGM evolution over coupled graphs.

Furthermore, we make the following observations.

(i) We assume that the sequence of matched pairs $\{\mathbf{z}^R_t\}_t \in \mathcal{P}^{(R)}$ exactly corresponds to the sequence of matched pairs $\{\mathbf{z}^r(t)\}_t \in \mathcal{P}_r$, i.e., $\mathbf{z}^r(t) = \mathbf{z}^R(t)$ at every t. This is made possible by the fact that given $\mathcal{Z}^r_{t-1} = \mathcal{Z}^R_{t-1}$, marks collected by every unmatched pair in the two graphs at time t exactly correspond.

(ii) Our construction is consistent since edges between pairs are unveiled only once, specifically at the time at which the first between the two edge endpoints in $\mathcal{P}_R$ is placed in $\mathcal{Z}^R_t = \mathcal{Z}^r_t$. Since then, the edge is replaced with an edge between two pairs that are both in $\mathcal{P}_R$, hence it will not be used again.

(iii) $\mathcal{P}_R$ is isomorphic to a pairs graph originated by a generalized Erdös–Rényi graph $\mathcal{G}^R_\mathrm{T}$, in which the presence of every edge $(\mathbf{z}^r(t),*)$ can be represented by a Bernoulli r.v. and the probability that the edge is added to the graph takes values in the range $[p_{\min}, p_{\max}]$ and is independent of other edges. Indeed, observe that the presence of an edge in $\mathcal{P}_R$ deterministically corresponds to the presence of the corresponding edge in $\mathcal{P}(\mathcal{G}_\mathrm{T})$. Furthermore, by construction, different edges in $\mathcal{P}_R$ correspond to different edges in $\mathcal{P}(\mathcal{G}_\mathrm{T})$.

The same observations hold when we consider the evolution of the marks of the pairs on the left hand side and a pairs graph $\mathcal{P}_L$, which is originated from a coupled generalized Erdös–Rényi graph $\mathcal{G}^L_\mathrm{T}$ with same properties as $\mathcal{G}^R_\mathrm{T}$.

Now, clearly

$$G(m,p_{\min}) \leq_{st} \mathcal{G}^R_\mathrm{T} \leq_{st} G(m,p_{\max})$$

and

$$G(m,p_{\min}) \leq_{st} \mathcal{G}^L_\mathrm{T} \leq_{st} G(m,p_{\max}),$$

i.e., $\mathcal{G}^R_\mathrm{T}$ ($\mathcal{G}^L_\mathrm{T}$) can be obtained by opportunely thinning a graph $G(m,p_{\max})$, while a graph $G(m,p_{\min})$ can be obtained by opportunely thinning $\mathcal{G}^R_\mathrm{T}$ ($\mathcal{G}^L_\mathrm{T}$). Then we invoke Theorem 1 to conclude our proof and show that our algorithm correctly percolates over $\mathcal{G}^R_\mathrm{T}$ and $\mathcal{G}^L_\mathrm{T}$ and, thus, over the bipartite graph $\mathcal{G}_\mathrm{T}$.

D. REFERENCES

[1] A. Narayanan, V. Shmatikov, "De-anonymizing social networks," *IEEE Symposium on Security and Privacy*, 2009.

[2] P. Pedarsani, D.-R. Figueiredo, M. Grossglauser, "A Bayesian method for matching two similar graphs without seeds," *IEEE Allerton* 2013.

[3] W. Peng, F. Li, X. Zou, J. Wu, "A two-stage deanonymization attack against anonymized social networks," *IEEE Trans. on Computers*, 63(2), 2014.

[4] N. Korula, S. Lattanzi, "An efficient reconciliation algorithm for social networks," *PVLDB*,2014.

[5] P. Pedarsani, M. Grossglauser, "On the privacy of anonymized networks," *SIGKDD*, 2011.

[6] L. Yartseva, M. Grossglauser, "On the performance of percolation graph matching," *COSN*, 2013.

[7] S. Janson, T. Luczak, T. Turova, T. Vallier, "Bootstrap percolation on the random graph $G_{n,p}$," *The Annals of Applied Probability,* 22(5), 2012.

[8] C.F. Chiasserini, M. Garetto, E.Leonardi, "De-anonymizing scale-free social networks by percolation graph matching," *INFOCOM*, 2015.

[9] K. Bringmann, T. Friedrich, A. Krohmer, "De-anonymization of heterogeneous random graphs in quasilinear time," *22nd Annual European Symposium on Algorithms, ESA'14.*

[10] C.F. Chiasserini, M. Garetto, E. Leonardi, "Impact of clustering on the performance of percolation-based graph matching," *Technical Report,* 2015, `http://arxiv.org/abs/1508.02017`.

[11] M. Penrose, *Random Geometric Graphs,* Oxford University Press, 2003.

[12] Pokec network dataset - KONECT, (website) `http://konect.uni-koblenz.de/networks/soc-pokec-relationships`