

De-anonymizing clustered social networks by percolation graph matching

Original

De-anonymizing clustered social networks by percolation graph matching / Chiasserini, Carla Fabiana; Garetto, M.; Leonardi, Emilio. - In: ACM TRANSACTIONS ON KNOWLEDGE DISCOVERY FROM DATA. - ISSN 1556-4681. - STAMPA. - 12:2(2018), pp. 1-39. [10.1145/3127876]

Availability:

This version is available at: 11583/2676472 since: 2018-02-07T10:18:27Z

Publisher:

ACM

Published

DOI:10.1145/3127876

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

ACM postprint/Author's Accepted Manuscript, con Copyr. autore

© Chiasserini, Carla Fabiana; Garetto, M.; Leonardi, Emilio 2018. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in ACM TRANSACTIONS ON KNOWLEDGE DISCOVERY FROM DATA, <http://dx.doi.org/10.1145/3127876>.

(Article begins on next page)

De-anonymizing clustered social networks by percolation graph matching

CARLA-FABIANA CHIASSERINI, Politecnico di Torino
 MICHELE GARETTO, Università di Torino
 EMILIO LEONARDI, Politecnico di Torino

On-line social networks offer the opportunity to collect a huge amount of valuable information about billions of users. The analysis of this data by service providers and unintended third parties are posing serious threats to user privacy. In particular, recent work has shown that users participating in more than one on-line social network can be identified based only on the structure of their links to other users. An effective tool to de-anonymize social network users is represented by graph matching algorithms. Indeed, by exploiting a sufficiently large set of seed nodes, a percolation process can correctly match almost all nodes across the different social networks. In this paper, we show the crucial role of clustering, which is a relevant feature of social network graphs (and many other systems). Clustering has both the effect of making matching algorithms more prone to errors, and the potential to greatly reduce the number of seeds needed to trigger percolation. We show these facts by considering a fairly general class of random geometric graphs with variable clustering level. We assume that seeds can be identified in particular sub-regions of the network graph, while no a-priori knowledge about the location of the other nodes is required. Under these conditions, we show how clever algorithms can achieve surprisingly good performance while limiting the number of matching errors.

CCS Concepts: •Mathematics of computing → Random graphs; Probabilistic algorithms; •Networks → Network privacy and anonymity; On-line social networks;

Additional Key Words and Phrases: Graph matching, bootstrap percolation, de-anonymization

1. INTRODUCTION

On-line social networks have recently emerged as one of the most influential innovations brought by information and communication technologies, with an enormous impact on social and economic aspects. Due to their popularity, the companies running these on-line services can acquire a huge amount of valuable information that can be extracted from the traces of activities performed by users. Such information can be exploited to construct user profiles, which may serve for targeted advertisements as well as marketing and social surveys. In this scenario, user privacy is clearly at stake. In particular, accurate user profiles can be obtained when users are members of different social networks and data extracted from different systems are combined together.

This paper is specifically concerned with the case of an ‘attacker’ trying to identify users belonging to different on-line 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 [Narayanan and Shmatikov 2009]. Indeed, the traces of user activities over a social network can be represented by a ‘contact graph’ in which nodes represent anonymized users, and edges denote who has come in contact with whom. Then, an attacker can execute a *graph-matching* algorithm on the contact graphs generated by different systems, and identify which labels correspond to the same user. In the hardest case, this is feasible by using only the topologies of the contact graphs [Pedarsani et al. 2013]. The majority of algorithms proposed so far to achieve this goal, however, exploit an initial set of already matched nodes (called seeds) [Narayanan and

Author’s addresses: C.F. Chiasserini and E. Leonardi, Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Italy; M. Garetto, Dipartimento di Informatica, Università di Torino, Italy.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© YYYY ACM. 1556-4681/YYYY/01-ARTA \$15.00

DOI: 0000001.0000001

Shmatikov 2009; Peng et al. 2014; Korula and Lattanzi 2014; Kazemi et al. 2015a]. This is actually a realistic case, since some users explicitly link their accounts in different systems ‘for free’.

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. Specifically, when the social network is modeled as an Erdős–Rényi random graph, it has been shown in [Pedarsani and Grossglauser 2011] 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 Erdős–Rényi contact graphs, in [Yartseva and Grossglauser 2013] the authors have proposed an identification algorithm, named PGM, based on bootstrap percolation [Janson et al. 2012], and they have determined the critical seed set size required to successfully trigger the de-anonymization process. Using a similar approach, more recently the work in [Chiasserini et al. 2016; Bringmann et al. 2014] has derived the critical seed set size for network de-anonymization when the contact graph exhibits a power-law degree distribution.

Another essential feature of real social networks, namely, clustering, has not been investigated so far. Interestingly, in [Yartseva and Grossglauser 2013] authors attempted to apply PGM 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 (unknown) impact of clustering from the (known) impact of power law degree, going back to the original case of Erdős–Rényi graphs and moving along a totally different, ‘orthogonal’ direction.

The main contributions of this work can be summarized as follows.

- Networks characterized by dense clusters may be largely prone to matching errors when we naively apply the method proposed in [Yartseva and Grossglauser 2013]. Such errors can be mitigated and asymptotically eliminated by an improved matching algorithm still based on bootstrap percolation.
- Once errors are removed, 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.
- In contrast with previous results derived for Erdős–Rényi [Yartseva and Grossglauser 2013] and power-law graphs [Chiasserini et al. 2016], we show that the 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, we do not confine our work exclusively to this problem. Indeed, the results derived here have much broader applicability, since graph matching is a general problem arising in many different domains, ranging from computer graphics (e.g., [Egozi et al. 2013]) to bioinformatics (e.g., [Singh et al. 2008]).

2. RELATED WORK

Many proposed matching strategies for network de-anonymization are based on heuristic algorithms and work by progressively expanding the set of already matched nodes, trying to identify all of the other nodes [Narayanan and Shmatikov 2009; Peng et al. 2014; Korula and Lattanzi 2014; Kazemi et al. 2015a]. In particular, in their seminal paper Narayanan and Shmatikov [Narayanan and Shmatikov 2009] were able to identify a large fraction of users having account on both Twitter and Flickr (with only 12% error ratio). Algorithms based on supervised learning have been proposed to de-anonymize social networks by exploiting semantic information (e.g., name, location and image of users) [Nunes et al. 2012; Abel et al. 2010]. Structure-based similarity (e.g., neighborhood structure) has been recognized to be the most important feature in the graph-matching process [Henderson et al. 2011; Backstrom et al. 2007].

Theoretical studies on the asymptotic performance of graph matching algorithms applied to large systems have appeared in [Pedarsani and Grossglauser 2011; Yartseva and Grossglauser 2013; Kazemi et al. 2015a; Chiasserini et al. 2016; Bringmann et al. 2014; Onaran et al. 2016]. Specifically, [Yartseva and Grossglauser 2013; Kazemi et al. 2015a] have addressed the case where users are members of two different social networks and the networks are modeled as Erdős–Rényi random graphs. In [Yartseva and Grossglauser 2013; Kazemi et al. 2015a] the authors propose practical identification algorithms based on bootstrap percolation [Janson et al. 2012], which exploit an initial seed set. It is worth mentioning that [Yartseva and Grossglauser 2013] shows an interesting phase transition phenomenon in the number of seeds that are required for network de-anonymization. The algorithm in [Kazemi et al. 2015a] instead can deal with only partial overlapping between the nodes in the two social network graphs and with moderate errors in the initial seed set.

While the above works all exploit the availability of a seed set, [Pedarsani and Grossglauser 2011; Kazemi et al. 2015b; Onaran et al. 2016] show that an attacker with unlimited computation power can perform de-anonymization even without seeds. In addition, [Kazemi et al. 2015b] considers the case where there is only partial overlap between the node sets of the two network graphs while [Onaran et al. 2016] generalizes the case of [Pedarsani and Grossglauser 2011] to graphs with community structure. The approach first proposed in [Pedarsani and Grossglauser 2011], which relies on the graphs structural information, has been also exploited [Ji et al. 2016] to quantify the full or partial de-anonymizability of general and real-world graphs.

The results in [Yartseva and Grossglauser 2013], related to a practical identification algorithm based on bootstrap percolation, have been recently extended to a more realistic case in which contact graphs are scale-free (power-law) random graphs. In particular, by modeling them as Chung-Lu graphs, [Chiasserini et al. 2016] and [Bringmann et al. 2014] have independently shown that a much smaller set of seeds is sufficient to trigger the percolation-based matching process originally studied in Erdős–Rényi graphs.

Finally, an early version of this work has appeared in the conference paper [Chiasserini et al. 2015b]. Also, it is worth mentioning that the identification problem is general and spans over different application fields [Leordeanu and Hebert 2005; Melnik et al. 2002; Motahari et al. 2013].

3. NOTATION AND PRELIMINARIES

The network de-anonymization problem under study can be formulated as follows. Consider two social networks, represented by the graphs $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2(\mathcal{V}_2, \mathcal{E}_2)$, respectively. Both graphs are considered to be sub-graphs of an inaccessible ground-truth graph, $\mathcal{G}_T(\mathcal{V}, \mathcal{E})$, representing the underlying true social relationships among people.

Without loss of generality, we assume that $\mathcal{G}_T(\mathcal{V}, \mathcal{E})$, $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2(\mathcal{V}_2, \mathcal{E}_2)$ have the same set of nodes (or vertices) with cardinality n , i.e., $\mathcal{V}_1 = \mathcal{V}_2 = \mathcal{V}$. This assumption can be easily released by seeking to match only the intersection of vertices belonging to \mathcal{G}_1 and \mathcal{G}_2 (see [Kazemi et al. 2015a] for an analysis with no-coincident node sets in Erdős–Rényi graphs). Similarly to previous works [Korula and Lattanzi 2014; Pedarsani and Grossglauser 2011; Yartseva and Grossglauser 2013; Chiasserini et al. 2016; Bringmann et al. 2014] we assume that edges in \mathcal{G}_1 and \mathcal{G}_2 are obtained by independently sampling each edge of \mathcal{G}_T with probability¹ s . Specifically, each edge in \mathcal{G}_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, first-step approximation of real systems and permits obtaining fundamental analytical insights [Yartseva and Grossglauser 2013; Bringmann et al. 2014; Kazemi et al. 2015a; Kazemi et al. 2015b]. Moreover, by looking at temporal snapshots of an email network, authors in [Pedarsani and Grossglauser 2011; Ji et al. 2014] have experimentally found that the above assumption of independent edge sampling is largely acceptable in their scenario.

¹Two different sampling probabilities s_1 and s_2 , respectively for \mathcal{G}_1 and \mathcal{G}_2 , could be considered as well. In Figure 16, we show some results for $s_1 = 0.75$ and $s_2 = 0.5$.

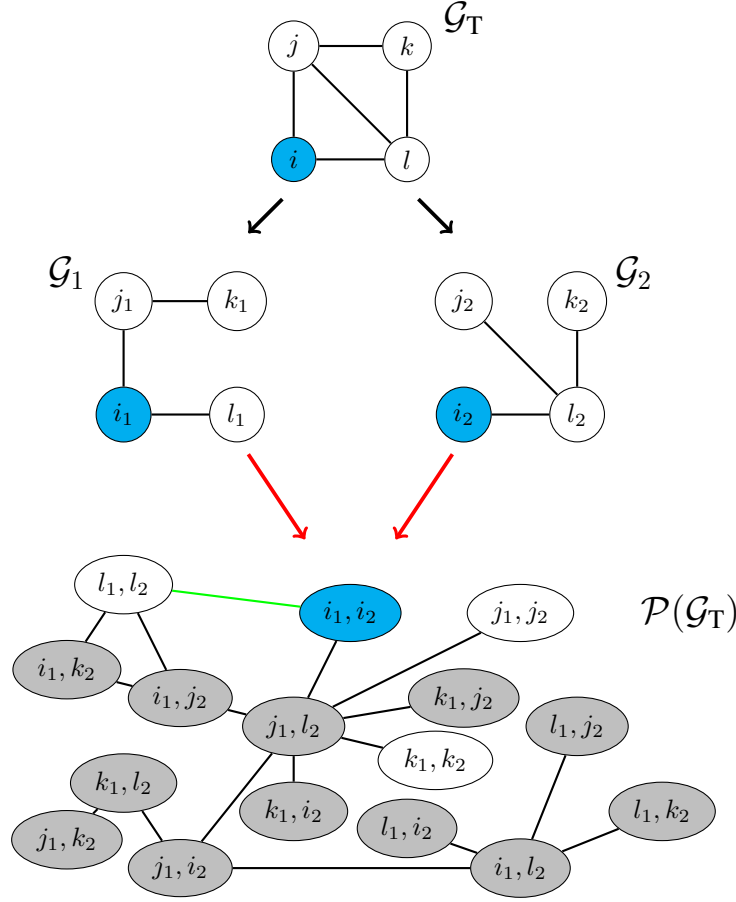


Fig. 1. An example of \mathcal{G}_1 and \mathcal{G}_2 obtained from \mathcal{G}_T by independent edge sampling, and of the pairs graph $\mathcal{P}(\mathcal{G}_T)$. There is a single seed, highlighted in blue. In $\mathcal{P}(\mathcal{G}_T)$, good pairs are highlighted in white and bad pairs in grey.

To match \mathcal{G}_1 and \mathcal{G}_2 , we build the pairs graph, $\mathcal{P}(\mathcal{V}, \mathcal{E})$, as the tensor product of \mathcal{G}_1 and \mathcal{G}_2 . Note that, by definition of tensor product, $\mathcal{V} = \mathcal{V}_1 \times \mathcal{V}_2$ and in $\mathcal{P}(\mathcal{V}, \mathcal{E})$ 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}_T simply as $\mathcal{P}(\mathcal{G}_T)$. Fig. 2 shows the pairs graph built from a toy example.

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

We now briefly describe the Percolation Graph Matching (PGM) algorithm originally proposed in [Yartseva and Grossglauser 2013]. The PGM algorithm maintains an integer counter (initialized to zero) for any pair of $\mathcal{P}(\mathcal{G}_T)$ that may still be matched. It exploits a set \mathcal{A}_t , indexed by time step t , which is initialized at $t = 0$ 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 $\mathcal{P}(\mathcal{G}_T)$. Then the algorithm adds to \mathcal{A}_{t+1} all pairs whose counter has reached a threshold 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 algorithm stops when $\mathcal{A}_t = \emptyset$. It is straightforward to see that PGM takes at most n steps to terminate.

Critical seed set size for Erdős-Rényi graphs [Yartseva and Grossglauser 2013].

In the case where \mathcal{G}_T is an Erdős-Rényi random graph², previous work [Yartseva and Grossglauser 2013] has obtained the following result on the critical seed set size a_c . We recall that a_c is such that: if $a_0/a_c < 1$, then only $o(n)$ nodes can be matched, while if $a_0/a_c < 1 + \delta$, for some $\delta > 0$, almost all nodes can be correctly identified.

LEMMA 3.1. Let \mathcal{G}_T be an Erdős-Rényi random graph $G(n, p)$. Let $r \geq 4$. Denote by a_c the critical seed set size:

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

For $n^{-1} \ll ps^2 \leq s^2 n^{-\frac{3.5}{r}}$, we have that, if $a_0/a_c < 1 + \delta$, the PGM algorithm matches w.h.p. a number of good pairs equal to $n - o(n)$ (i.e., all vertex pairs except for a negligible fraction) with no errors.

In our analysis we assume that s is a positive finite constant (it does not scale with n), thus we omit it whenever we report asymptotic expressions in order sense. Our approach could be extended also to the case of vanishing s , but this is out of the scope of this work.

Critical seed set size for random graphs bounded by Erdős-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: $\mathcal{H}(\mathcal{V}, \mathcal{E}_H) \leq_{st} \mathcal{K}(\mathcal{V}, \mathcal{E}_K)$.

Then, consider a vertex property \mathcal{R} satisfied by a subset of vertices, and denote with $\mathcal{R}(\mathcal{H}) \subseteq \mathcal{V}$ the set of vertices of \mathcal{H} that satisfy property \mathcal{R} . We say that \mathcal{R} is *monotonically increasing with respect to the graph ordering relation* “ \leq_{st} ” if $\mathcal{R}(\mathcal{H}) \subseteq \mathcal{R}(\mathcal{K})$ whenever $\mathcal{H} \leq_{st} \mathcal{K}$.

Given that, we present the following results, which complement Theorem 2 and Corollary 3 in [Chiasserini et al. 2016]:

Theorem 1. Consider a subgraph $\mathcal{G}'_T \subseteq \mathcal{G}_T$, which comprises a subset of vertices of \mathcal{G}_T , whose number is denoted by m , and all the edges between the selected vertices. Assume that $G(m, p_{\min}) \leq_{st} \mathcal{G}'_T \leq_{st} G(m, p_{\max})$ 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) $m \rightarrow \infty$;
- (2) $p_{\min} = \Theta(p_{\max})$ and $p_{\min} \gg m^{-1}$;
- (3) $p_{\max} \leq m^{-\frac{3.5}{r}}$, with $r \geq 4$;
- (4) $\liminf_{m \rightarrow \infty} a_0/a_c > 1$, with a_c computed from (1) by setting $p = p_{\min}$.

PROOF. The proof can be found in Appendix C. \square

Corollary 1. Under the same conditions as in Theorem 1, the PGM algorithm can be successfully applied to a pairs graph $\hat{\mathcal{P}} \subset \mathcal{P}(\mathcal{G}'_T)$ 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 $\hat{\mathcal{P}}$ only if either $[i_1, i_2]$ or $[j_1, j_2]$ are also in $\hat{\mathcal{P}}$.

PROOF. The proof of can be found in [Chiasserini et al. 2016], for completeness a sketch is also reported in Appendix D. \square

²Given a positive integer n and a probability value $0 \leq p \leq 1$, the Erdős-Rényi graph $G(n, p)$ is defined as the undirected graph on n vertices whose edges are chosen as follows. For all pairs of vertices v, w there is an edge (v, w) with probability p .

The above results provide the 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.

4. 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.

We assume that nodes are located in a k -dimensional space corresponding to the hyper-cube³ $\mathcal{H} = [0, 1]^k \subset \mathbb{R}^k$, 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 . 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 nodes⁴. 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 [Leskovec et al. 2007]. Also, although a common choice is to assume $D(n) = \Theta(\log n)$, in our model we consider more general conditions: $D(n) = \Omega(\log n)$ and $D(n) = O(n^{1/2-\delta})$ with $0 < \delta < 1/2$. Note that, since $D(n) \rightarrow \infty$ as n grows large, the graph is connected with high probability.

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. Considering that the average distance between neighboring nodes is equal⁵ to $n^{-1/k}$, define $C(n) = \Omega(n^{-1/k})$. 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, in order 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,

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

thus $p_{ij} = K(n) \min \{ 1, (C(n)/d_{ij})^\beta \}$. 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 Erdős–Rényi graph when either $\beta \rightarrow 0$, with arbitrary $C(n)$, or $C(n)$ approaches 1, with arbitrary β . For $\beta \rightarrow \infty$, 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

³To avoid border effects, we assume wrap-around conditions (i.e., a torus topology).

⁴Note that the node degree distribution is a sum of Bernoulli functions, conditioned on the node position.

⁵We remark that $n^{-1/k}$ is the inverse of the k -th root of the node density in the region \mathcal{H} .

a node lie at distance $o(1)$ from it. Thus, we can write:

$$D(n) = \begin{cases} \Theta(nK(n)C^k(n)) & \beta > k \\ \Theta\left(nK(n)C^k(n) \log \frac{1}{C(n)}\right) & \beta = k \\ \Theta(nK(n)C^\beta(n)) & \beta < k \end{cases} \quad (4)$$

Next, we compute the scaling order of the clustering coefficient⁶ χ . As shown in Appendix B, we have:

$$\chi = \begin{cases} \Theta(K(n)) & \beta > k \\ \Theta\left(\frac{K(n)}{\log^2[1/C(n)]}\right) & \beta = k \\ \Theta(K(n)C(n)^{2(k-\beta)}) & \frac{2k}{3} < \beta < k \\ \Theta(K(n)C(n)^\beta \log[1/C(n)]) & \beta = \frac{2k}{3} \\ \Theta(K(n)C(n)^\beta) & \beta < \frac{2k}{3} \end{cases} \quad (5)$$

Looking at χ , we note that in all cases the clustering coefficient of the graph is upper-bounded by $K(n)$ (actually it equals $K(n)$ for $\beta > k$). In essence, in our model $K(n)$, $C(n)$ and β provide the three knobs that allow us to directly control the both the average node degree and clustering coefficient of the graph. We underline that many real-world networks exhibit a fairly large clustering coefficient, as it occurs in our model when $\beta > k$.

We remark that, unlike specifically tailored graph models such as stochastic block-models, our geometric random graphs do not directly capture the community-based structure exhibited by social networks. However, they successfully represent the clustering effect, which is the main feature investigated in this paper.

Note: In the following, we will slightly abuse the language and define as *clusters* (not to be confused with the clustering coefficient) sub-regions including nodes whose maximum mutual distance is $\Theta(C(n))$.

5. OVERVIEW AND MAIN RESULTS

Our goal is to characterize the seed set necessary for de-anonymization. To this end, we identify two different regimes depending on $K(n)$, which provides the graph density of clusters⁷:

- 1) $K(n) = o([nC^k(n)]^{-\gamma})$, for some $0 < \gamma < 1$, which will be referred to as *low-density cluster case*;
- 2) $K(n) = \omega([nC^k(n)]^{-\gamma})$ for any $\gamma > 0$, which will be referred to as *high-density cluster case*.

In the first case, the graph density within a cluster goes to zero “relatively” fast as the number of nodes within a cluster, $nC^k(n)$, goes to infinity. In the second case, the graph density within a cluster either is bounded away from zero or asymptotically decreases very slowly. It comprises the particularly relevant sub-case in which $K(n) = \Theta(1)$. Within each of the above cases, different operational sub-regimes can be identified based on the value of β/k and $B(n) = nK(n)C^k(n)$. Note that $B(n)$ can be interpreted as the average number of neighbors of a given node within a cluster centred at the considered node.

In order to analyze the performance achievable in different cases, we will consider three different matching strategies. Note that for all strategies we will assume that seeds can be identified in particular sub-regions of \mathcal{H} , while no knowledge on the location of the other nodes is required to carry on the node identification process.

(i) The simplest approach consists in applying PGM directly to the original pairs graph by using seeds in an opportunely defined sub-region of \mathcal{H} , and by selecting a proper threshold r . We point out

⁶Roughly speaking, the clustering coefficient is the probability that two neighbors of a node are neighbors of each other.

⁷Given 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.

that, in this case, r may scale to infinity as n grows large, as shown in Section 6. Such an approach will be used when the product $B(n) \cdot K(n)$ is small, for either low or high-density clusters (see Sections 7.2, 7.3, 7.4 and 8.2). Indeed, in these cases, the edge density within a cluster is small enough to safely apply the PGM algorithm without incurring matching errors.

(ii) The second approach implies that initially only a small sub-region of \mathcal{H} of size $\Theta(C(n))$ is considered. Then the PGM algorithm can be applied to this sub-region⁸, provided that a sufficiently large seed set is available therein and an opportune threshold r is selected. At the end of this first ‘trigger phase’ almost all nodes located in the considered region are correctly identified. The set of matched pairs is then iteratively expanded, using as seed set the good pairs identified at the previous stage (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 neighboring pairs matched at the previous steps. Fig. 2 depicts this approach. This matching procedure will be used to de-anonymize nodes pairs in the case of low-density clusters when $B(n) \cdot K(n)$ is sufficiently large (Sec. 7.1).

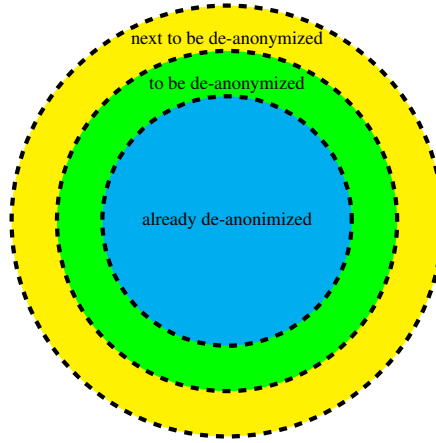


Fig. 2. Graphical representation of matching strategy based on wave-like expansion.

(iii) A more complex de-anonymization procedure is required for high-density clusters and large values of $B(n)$, i.e., when the graph may have many cliques or quasi-cliques of nodes (Sec. 8.1). Indeed, in this case, if we try to identify nodes using only the local structure of a cluster (as in the previous cases), an intolerable amount of matching errors may occur disrupting the entire identification process. In order to prevent this, all edges whose length is too short should be ignored, and nodes should be identified only on the basis of the ‘fingerprint’ provided by their longer edges. Thus, we first devise a ‘trigger phase’ using two sub-regions of \mathcal{H} of size $\Theta(C(n))$, which are sufficiently far from each other, i.e., they are separated by a minimum distance $\omega(C(n))$. Fig. 3 illustrates the two sub-regions in the case in which both are square shaped with generic side length $h(n)$. We assume that a suitable number of seeds is available within each of these sub-regions. To identify all of the other nodes therein, we modify the PGM algorithm so that only the edges between nodes belonging to different sub-regions are used. We then exploit the fact that, in the high-density cluster regime, the distance between two nodes in \mathcal{H} can be estimated quite precisely. Therefore we can select a set of compact nodes that are sufficiently far from a matched sub-region, and re-apply the direct strategy. The procedure can be iterated until almost all nodes throughout the network are correctly identified.

⁸Note that no a-priori knowledge on the position of the nodes (other than seeds) is required. Indeed, it is needed only the relative distance of the nodes from the seeds, which can be estimated as shown Sec. 7.1.

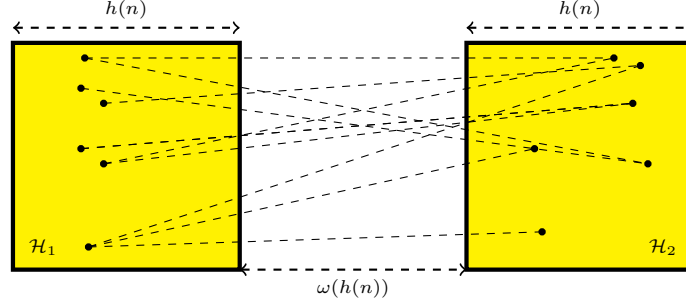
Fig. 3. Graphical representation of trigger phase based on two separated sub-regions of side $h(n)$.

Table I. Main parameters

n	Number of nodes	$\mathcal{H}(n)$	Hyper-cube where nodes are located
k	No. of dimensions of the hyper-cube $\mathcal{H}(n)$	β	Decaying factor of the connectivity probability with distance
$C(n)$	Size of a cluster	$K(n)$	Probability of two nodes being connected within a cluster
$B(n)$	Average no. of neighbors of a node within a cluster centered at the tagged node	$h(n)$	Size of a subregion of \mathcal{H}
$\mathcal{G}_T(n)$	Ground-truth graph	$\mathcal{G}_1, \mathcal{G}_2$	Graphs obtained from \mathcal{G}_T by independent edge sampling
s	Edge sampling probability	$g(n)$	Distance between two subregions of \mathcal{H}
$\mathcal{P}(\mathcal{G})$	Pairs graph induced by nodes in \mathcal{G}	$\mathcal{G}(n, p)$	Erdős-Rényi graph with n nodes and edge probability p
\mathcal{A}_0	Seed set	a_o	Seed set cardinality
r	Threshold value on no. of edges	μ	Average no. of good pairs neighbor of a tagged pair

Our main findings on the seed set size required for successful network de-anonymization are summarized below. The main used notations are reported in Table I.

- The required number of seeds heavily depends on all network parameters: $K(n)$, $B(n)$ (or $C(n)$) and β/k .
- For *high-density clusters* and $\beta > k/2$, the number of necessary seeds can be simply expressed in terms of the average number of nodes falling within a cluster, $[nC^k(n)]$. In particular, a seed set whose size is equal to $[nC^k(n)]^\epsilon$, for any $\epsilon > 0$, is enough to guarantee an almost complete successful network de-anonymization.
- In the relevant case in which $K(n) = \Theta(1)$ and $B(n) = \Theta(1)$ (i.e., $nC^k(n) = \Theta(\log n)$ and the average degree of the graph $D(n) = \Theta(\log n)$), the above expression reduces to $(\log n)^\epsilon$, with arbitrarily small $\epsilon > 0$. This result strikingly shows the beneficial impact of clustering on network de-anonymization. To grasp this fact, consider that, in the case of a $G(n, p)$ ground-truth graph with average degree $D(n)$, the PGM algorithm requires $\Theta\left(\frac{n}{D(n)^{r/(r-1)}}\right)$ seeds, which, in the case of $\log n$ degree, is just a poly-log factor less than n .
- Somehow surprisingly, the required seed set size increases when we increase the average degree of the nodes while keeping $K(n)$ constant (i.e., when we increase $C(n)$ and consequently $B(n)$). This is in sharp contrast with previous results derived for Erdős-Rényi and Chung-Lu graphs [Yartseva and Grossglauser 2013; Chiasserini et al. 2016]. The intuition behind this result is that, when clusters are very highly connected, it becomes very hard to distinguish nodes within the

same cluster. Therefore, by increasing the cluster size, we make the identification intrinsically more challenging.

- In the *low-density cluster* case, our de-anonymization techniques become less effective, and the required seed set size turns out to be roughly inversely proportional to $K(n)$.
- In both the *high-density cluster* and *low-density cluster* cases, for a fixed value of average node degree and $\beta > k/2$, the required seed set size increases as β decreases. Observe that the clustering coefficient decreases as we decrease β , while keeping the average degree constant.
- For *low-density clusters*, a fixed value of average node degree and $\beta < k/2$, by decreasing β the graph tends to a $G(n, p)$, thus our results tend to those derived in [Yartseva and Grossglauser 2013] for $G(n, p)$ graphs.

6. RELATIONSHIP BETWEEN PGM THRESHOLD AND MATCHING DYNAMICS

Let us consider a bad pair $[i_1, j_2]$ and that vertices i and j are placed at \mathbf{x}_i and \mathbf{x}_j , respectively. We want to investigate the number of good pairs that are neighbors of $[i_1, j_2]$ on the pairs graph $\mathcal{P}(\mathcal{G}_T)$.

Let $[l_1, l_2]$ be a generic good pair and $X_{[i_1, j_2], [l_1, l_2]}$ be the indicator function associated to the event that $\{[i_1, j_2]$ and $[l_1, l_2]$ are neighbors on $\mathcal{P}(\mathcal{G}_T)\}$. By using (2) and recalling that i) \mathcal{G}_1 and \mathcal{G}_2 have been obtained via independent edge sampling with probability s , and ii) nodes are uniformly distributed over \mathcal{H} , we have:

$$\begin{aligned} \mathbb{E}[X_{[i_1, j_2], [l_1, l_2]}] &= \int_{\mathcal{H}} \mathbb{E}[X_{[i_1, j_2], [l_1, l_2]} \mid \mathbf{x}_l] d\mathbf{x}_l = \\ &= s^2 K(n)^2 \int_{\mathcal{H}} f(\|\mathbf{x}_l - \mathbf{x}_i\|) f(\|\mathbf{x}_l - \mathbf{x}_j\|) d\mathbf{x}_l. \end{aligned} \quad (6)$$

Since (6) holds for any good pair, the average number of good pairs that are neighbors of $[i_1, j_2]$ is given by:

$$\begin{aligned} \mu &= \mathbb{E} \left[\sum_l X_{[i_1, j_2], [l_1, l_2]} \right] = (n-2) \mathbb{E}[X_{[i_1, j_2], [l_1, l_2]}] \geq (n-2) s^2 K(n)^2 \int_{\mathcal{H}} f^2(\|\mathbf{x}\|) d\mathbf{x} \\ &= \begin{cases} \Theta(B(n)K(n)) & \beta > \frac{k}{2} \\ \Theta\left(B(n)K(n) \log \frac{1}{C(n)}\right) & \beta = \frac{k}{2} \\ \Theta(nC^{2\beta}(n)K^2(n)) & \beta < \frac{k}{2} \end{cases} \end{aligned} \quad (7)$$

where in the order sense expressions we have neglected the constant factor s^2 . Then we can apply standard concentration inequalities (see App. A) to bound the probability that the number of good pairs that are neighbors of the bad pair $[i_1, j_2]$ exceeds a threshold r , i.e.,

$$\mathbb{P}\left(\sum_l X_{[i_1, j_2], [l_1, l_2]} \geq r\right) \leq e^{-\frac{r}{2} \log \frac{r}{\mu}} \quad (\text{for } e^2 \mu < r).$$

If we consider jointly all possible wrong pairs, we have:

$$\begin{aligned} \mathbb{P}(\text{there exists a wrong pair with at least } r \text{ neighboring good pairs}) &\leq n^2 e^{-\frac{r}{2} \log \frac{r}{\mu}} \quad (\text{for } e^2 \mu < r) \\ &= \exp\left(2 \log n - \frac{r}{2} \log \frac{r}{\mu}\right) \end{aligned}$$

which goes to zero as $n \rightarrow \infty$ as long as $2 \log n - \frac{r}{2} \log \frac{r}{\mu} \rightarrow \infty$. Such a condition is satisfied when:

$$\mu = o(\log^{1-\xi} n) \wedge r = \Theta\left(\frac{\log n}{\log \log n}\right) \quad \text{for some } 0 < \xi < 1 \quad (8)$$

or,

$$\mu = o(n^{-\xi}) \quad \wedge \quad r = \Theta(1) \quad \text{for some } 0 < \xi < 1. \quad (9)$$

Observe that the condition $\mu = o(\log^{1-\xi} n)$ includes the case $\mu = o(n^{-\xi})$; in other words, the condition on μ in (9) is a sub-case of that in (8). When either (8) or (9) hold, bad pairs can be ignored during the whole percolation process since w.h.p none of them will reach the threshold r at any stage of the algorithm. Indeed, by induction, it can be proved that only good pairs are matched, thus only good pairs contribute to the increase of the pairs marks. The following theorem therefore holds. It states that if PGM can match almost all good pairs in $\mathcal{G}_0 \subseteq \mathcal{G}_T$, it can also match them when it is applied to the whole graph \mathcal{G}_T . Although intuitive, the result is not trivial to prove.

Theorem 2. *Consider a subgraph of $\mathcal{G}_T(\mathcal{V}, \mathcal{E})$, denoted by $\mathcal{G}_0(\mathcal{V}_0, \mathcal{E}_0)$ with $\mathcal{V}_0 \subseteq \mathcal{V}$ and $\mathcal{E}_0 \subseteq \mathcal{E}$. Assume that, applying PGM to \mathcal{G}_0 , we can successfully match (almost) all good pairs in $\mathcal{P}(\mathcal{G}_0)$ using as seed set $\mathcal{A}_0 \in \mathcal{V}_0$. Then, whenever either (8) or (9) hold, applying PGM to \mathcal{G}_T using the same seed set \mathcal{A}_0 successfully matches at least (almost) all good pairs corresponding to the vertices in \mathcal{V}_0 .*

PROOF. See Appendix E. \square

Finally, we show the following important result. Theorem 3 identifies the conditions on the seed set size and on the relation between the average node degree and the PGM threshold, under which good pairs can be successfully matched in any subgraph $\mathcal{G}_0 \subseteq \mathcal{G}_T$.

Theorem 3. *Consider a subgraph of $\mathcal{G}_T(\mathcal{V}, \mathcal{E})$, denoted by $\mathcal{G}_0(\mathcal{V}_0, \mathcal{E}_0)$, as before. Assume that $G(m, p_{\min}) \leq_{st} \mathcal{G}_0$, with $G(m, p_{\min})$ being an Erdős-Rényi graph. Under (8), applying the PGM algorithm to $\mathcal{P}(\mathcal{G}_T)$ (with $r = \Theta(\frac{\log n}{\log \log n})$) guarantees that $m - o(m)$ good pairs are matched with no errors w.h.p., provided that:*

- (1) $m \rightarrow \infty$;
- (2) $mp_{\min} \gg r$;
- (3) $a_0 > \frac{r}{p_{\min} s^2} (1 + \delta)$ for any $\delta > 0$

PROOF. See Appendix F. \square

7. LOW-DENSITY CLUSTERS

In this case, we consider clusters with low graph density, i.e., with $K(n) = o([nC^k(n)]^{-\gamma})$, for some $0 < \gamma < 1$. Also, we assume that there exists a set of seeds \mathcal{A}_0 ($|\mathcal{A}_0| = a_0$) such that the maximum mutual distance between seed nodes is $d_s = O(C(n))$, i.e., that seeds are concentrated within a k -dimensional space of radius $C(n)$. Below, we further distinguish four cases, according to (i) the average number of neighbors that a node has within distance $C(n)$ and (ii) the relationship between β and k . For each of such cases, we derive the number of seeds a_0 that are necessary in order to successfully de-anonymize our social graph. Specifically, we obtain the following results:

- Case $B(n) = \Omega(\log n)$, $\beta \geq k/2$ (Sec. 7.1): $a_0 = \Omega\left(\frac{\log(nC^k(n))}{K(n)}\right)$. In this case each node within a cluster has got a number of neighbors that, although limited, is still significant. Thus we consider a small sub-region of \mathcal{H} of side $\Theta(C(n))$ and initially apply the PGM algorithm to this sub-region. In this way, by selecting an opportune threshold r , we are able to match almost all nodes within a cluster. The set of matched pairs is then iteratively expanded, using as seed set the good pairs identified at the previous stage and matching those pairs having a sufficiently large number of already matched neighboring pairs.
- Case $B(n) = o(\log n)$ and $B(n) = \omega(n^{-\xi}) \quad \forall \xi > 0$, $\beta > k/2$ (Sec. 7.2): $a_0 = \Theta\left(\frac{\log n}{K(n)C^{\beta}(n)h^{-\beta}(n)}\right)$. In this case, given any node, its number of neighbors within the cluster is

small. We therefore apply PGM directly to the whole pairs graph by selecting a proper threshold r . Indeed, the edge density within a cluster is so small that the PGM algorithm can be safely adopted without incurring a significant number of errors.

- Case $B(n) = o(n^{-\xi})$ for some $0 < \xi < 10$, $\beta > k/2$ (Sec. 7.3) and Case $\beta \leq k/2$ (Sec. 7.4): $a_0 = \Theta \left(\left[(nh^k(n))^{\frac{1}{r-1}} (h^{-\beta}(n)K(n)C^\beta(n))^{\frac{r}{r-1}} \right]^{-1} \right)$. Here the number of neighbors of a node within a cluster vanishes fast as n increases. The result is thus obtained by using the same methodology as before, i.e., applying PGM to the whole pairs graph.

Algorithm 1 De-anonymization algorithm used in Section 7.1

Require: $\mathcal{G}_1, \mathcal{G}_2, \mathcal{A}_0$ with $a_0 = \Omega(\log[nC^k(n)]/K(n))$

- 1: Choose $\alpha \geq 0, \delta > 0$
- 2: $\mathcal{N}^1(\alpha) \leftarrow \emptyset, \mathcal{N}^2(\alpha) \leftarrow \emptyset, \mathcal{M} \leftarrow \emptyset$
- 3: **for** $i \in \mathcal{G}_1$ **do**
- 4: **if** No. of seeds that are neighbor of i in $\mathcal{G}_1 > \alpha sK(n)a_0$ **then**
- 5: $\mathcal{N}^1(\alpha) \leftarrow \mathcal{N}^1(\alpha) \cup \{i\}$
- 6: **for** $i \in \mathcal{G}_2$ **do**
- 7: **if** No. of seeds that are neighbor of i in $\mathcal{G}_2 > \alpha sK(n)a_0$ **then**
- 8: $\mathcal{N}^2(\alpha) \leftarrow \mathcal{N}^2(\alpha) \cup \{i\}$
- 9: Build the pairs graph $\mathcal{P}(\mathcal{N})$ induced by nodes in $\mathcal{N}^1(\alpha)$ and $\mathcal{N}^2(\alpha)$
- 10: **Set r to a sufficiently large value**
- 11: Apply PGM to $\mathcal{P}(\mathcal{N})$ using r as above % Trigger phase
- 12: $\mathcal{M} \leftarrow \{\text{matched pairs}\}$
- 13: **while** $\mathcal{M} \neq \emptyset$ **do** % Wave-like expansion
- 14: $\mathcal{A}_0 \leftarrow \mathcal{A}_0 \cup \mathcal{M}$
- 15: Update r as in Theorem 5
- 16: Match all pairs with at least r already-matched neighboring pairs
- 17: $\mathcal{M} \leftarrow \{\text{newly matched pairs}\}$
- 18: **return** Set of matched pairs

7.1. Case $B(n) = \Omega(\log n), \beta \geq k/2$

We first focus on the case in which $B(n) = nK(n)C^k(n) = \Omega(\log n)$. Here we have $\mu = B(n)K(n)$ for $\beta > k/2$ and $\mu = B(n)K(n)\log \frac{1}{C(n)}$ for $\beta = k/2$. Thus, in general in both cases we cannot guarantee that μ meets at least one of the conditions in (8)–(9). We therefore proceed as summarized in Alg. 1.

Specifically, our steps are as follows. Since $K(n) = o(1)$, by construction $nC^k(n) \gg \log n$, i.e., $C(n) = \omega \left(\left[\frac{\log n}{n} \right]^{\frac{1}{k}} \right)$. As a first step, 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}_{in} (\mathcal{H}_{out}) can be seen as the set of points whose distance from any seed vertex is lower (higher) than a given threshold. More formally, denote by \mathbf{x} a generic point in \mathcal{H} and by \mathbf{x}_σ the position in \mathcal{H} of a generic seed vertex σ .

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

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

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

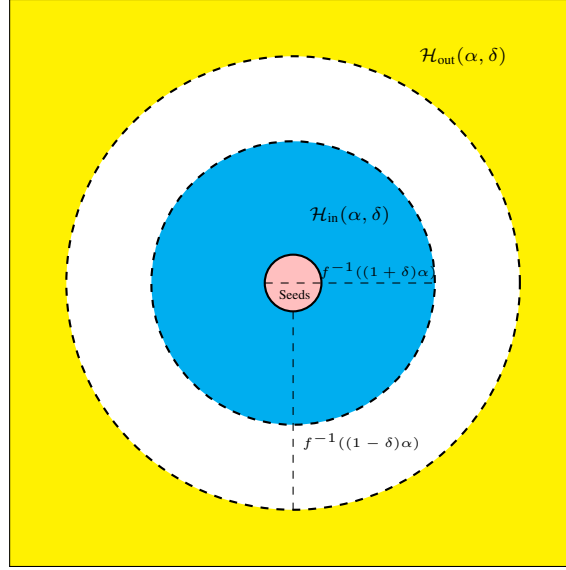


Fig. 4. Graphical representation of $\mathcal{H}_{\text{in}}(\alpha, \delta)$ and $\mathcal{H}_{\text{out}}(\alpha, \delta)$.

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. Note that the theorem holds under quite general conditions, specifically, for both low-density and high-density clusters.

Theorem 4. *Provided that $\frac{nC^k(n)}{\log(nC^k(n))} = \Omega\left(\frac{1}{K(n)}\right)$ and $nC^k(n) > \log n$, 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 tag a node i as “accepted” if $S_i > \alpha s K(n) a_0$. If $d_s = O(C(n))$ and $a_0 = \Theta\left(\frac{\log(nC^k(n))}{K(n)}\right)$, then for an arbitrary $\delta > 0$, the above procedure accepts all nodes located in $\mathcal{H}_{\text{in}}(\alpha, \delta)$, while it rejects all nodes located in $\mathcal{H}_{\text{out}}(\alpha, \delta)$.*

PROOF. See Appendix G. \square

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, $|\mathcal{N}^1(\alpha)| = \Theta(nC^k(n))$ and $|\mathcal{N}^2(\alpha)| = \Theta(nC^k(n))$. We build the pairs graph $\mathcal{P}(\mathcal{N})$ 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 make sure 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 $r = \Theta(1)$, $p_{\max} \ll m^{-\frac{3.5}{r}}$. Since by assumption $nK(n)C^k(n) = \Omega(\log n)$, we have $mK(n) = \Omega(\log n)$ and, hence, $mp_{\min} = \Omega(\log n)$. It follows that $p_{\min} \gg m^{-1}$. At last, it is easy to see that $a_o/a_c \rightarrow \infty$. Indeed, consider (1) where p is replaced with p_{\min} , and recall that $p_{\min} = \Theta(K(n))$ and $mp_{\min} = \Theta(B(n))$, then we have:

$$a_c = \Theta\left(\frac{1}{[B(n)]^{\frac{1}{r-1}} K(n)}\right).$$

I.e., $a_c = o(1/K(n))$ while, by assumption (see Theorem 4),

$$a_o = \Omega\left(\frac{\log(nC^k(n))}{K(n)}\right).$$

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 pursue 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 5. *Under the assumption $B(n) = \Omega(\log n)$, consider a circular region $\mathcal{D}(0, \rho) \subseteq \mathcal{H}$ centered at 0, of radius ρ , with $\rho \geq C(n)$. Assume that all (or almost all) good pairs whose constituent nodes lie within $\mathcal{D}(0, \rho)$ have been correctly matched. Then, it is possible to correctly match (almost) all good pairs whose constituent nodes are 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 $0 < \gamma < 1$. In addition, none of the bad pairs formed by nodes in $\mathcal{H} - \mathcal{D}(0, \rho)$ will be matched, 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 with vertices in $\mathcal{D}(0, \rho)$.*

PROOF. The proof is based on the application of standard concentration results, namely, Chernoff bound and inequalities in Appendix A. The detailed proof is given in [Chiasserini et al. 2015a]. \square

Almost all good pairs can be matched w.h.p. by iterating the matching procedure of Theorem 5 a number of steps equal to $\Theta(1/C(n))$. 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 5 also guarantees that, jointly over all steps, no bad pair is matched w.h.p.

7.2. Case $B(n) = o(\log n)$ and $B(n) = \omega(n^{-\xi}) \forall \xi > 0$, $\beta > k/2$

In this case, since in general $nC^k(n)$ is not guaranteed to be greater or equal to $\log n$, we cannot apply Theorem 4. However, given that $B(n) = nK(n)C^k(n) = o(\log n)$ and $K(n) = o([nC^k(n)]^{-\gamma})$ for some $0 < \gamma < 1$, it follows that $\mu = nC^k(n)K^2(n) = o(\log^{1-\gamma} n)$ (under the assumption $\beta > k/2$). Thus we can fix $r = \Theta\left(\frac{\log n}{\log \log n}\right)$ and apply a standard PGM on the whole graph \mathcal{G}_T . In this way, condition (8) holds and w.h.p. no wrong pairs are ever matched at any stage of the algorithm. However, we have to show that, by applying PGM, the process of matching good pairs successfully percolates over the all pairs graph. To this end, we need to apply Theorems 2 and 3, and, in particular, to show that the conditions in Theorem 3 hold for an opportunely selected subgraph of \mathcal{G}_T , \mathcal{G}_0 .

We then consider the subgraph \mathcal{G}_0 induced by the m vertices residing in a square box of side $h(n)$. Observe that, since the percolation process over good pairs is monotonically increasing, then the percolation process over \mathcal{G}_0 is faster than a percolation process over $G(m, p_{\min})$, where p_{\min} is the minimum connectivity probability among vertices in \mathcal{G}_0 .

Consider that $m = nh^k(n)$ and $f(d) = \Theta(C^\beta(n)h^{-\beta}(n))$, thus

$$mp_{\min} = nh^k(n)K(n)C^\beta(n)h^{-\beta}(n) \stackrel{(a)}{=} \Gamma(n)r$$

where $\Gamma(n)$ is an arbitrarily slow function such that (i) $\Gamma(n) \rightarrow \infty$ and (ii) (a) guarantees that the condition $mp_{\min} \gg r$ in Theorem 3 is satisfied. We then derive $h(n)$ as:

$$h(n) = \left(\frac{r\Gamma(n)}{nC^\beta(n)K(n)} \right)^{\frac{1}{k-\beta}} \quad (10)$$

where $C(n) \ll h(n) \ll 1$, by construction. Thus we can successfully apply Theorem 3 by choosing a number of seeds:

$$a_0 = \Theta\left(\frac{r}{p_{\min}}\right) = \Theta\left(\frac{\frac{\log n}{\log \log n}}{K(n)C^\beta(n)h^{-\beta}(n)}\right)$$

and obtain that PGM successfully matches all good pairs within a distance $\Theta(h(n))$ from the seed set.

To further expand the set of good pairs that are matched, conceptually we can apply again the PGM algorithm by employing all good pairs that have been previously matched as new seed set. Then all good pairs whose constituent nodes lie within a distance $\Theta(C(n))$ from this new seed set will be matched. Iterating the argument, we can match almost all good pairs in the graph. In practice, however, iterating PGM is not necessary as the matching procedure does never stop until almost all good pairs in the graph have been matched. This because PGM naturally uses previously matched pairs as adjoint seeds to match new pairs (when PGM matches a pair and places it in $\mathcal{Z}(t)$, it adds one mark to all its neighbors).

The above de-anonymization procedure is summarized in Alg. 2.

Algorithm 2 De-anonymization algorithm used in Section 7.2, 7.3, 7.4, 8.2

Require: $\mathcal{G}_1, \mathcal{G}_2$

- 1: **if** $\mu = o(n^{-\xi})$ **then** % As in Secs. 7.3-7.4
 - 2: **Set** r **to an arbitrary constant value**
 - 3: **else if** $\mu = o(\log^{1-\xi} n)$ **then** % As in Secs. 7.2-8.2
 - 4: $r \leftarrow \Theta(\log n / \log \log n)$
 - 5: Select a subregion of \mathcal{H} of size $h(n)$ % As in (10) for Sec. 7.3, (11) for Secs. 7.2-7.4
 - 6: % and (14) for Sec. 8.2
 - 7: $\mathcal{A}_0 \leftarrow \{\text{seeds in } \mathcal{H}\}$
 - 8: Apply PGM to $\mathcal{P}(\mathcal{G}_T)$ using r and \mathcal{A}_0 as above
 - 9: **return** Set of matched pairs
-

7.3. Case $B(n) = o(n^{-\xi})$ **for some** $0 < \xi < 1$, $\beta > k/2$

This case immediately implies that $\mu = nC^k(n)K^2(n) \ll B(n) = o(n^{-\xi})$. Thus condition (9) holds and, for an arbitrarily chosen $r = \Theta(1)$ we can again apply a standard PGM on the whole graph so as to ensure that w.h.p no wrong pairs are ever matched at any stage of the algorithm. The procedure reported in Alg. 2 therefore holds also for this case.

Specifically, as before we have to show that it is possible to apply Theorems 2 and 3 to an opportune subgraph $\mathcal{G}_0 \subseteq \mathcal{G}_T$. We then consider the subgraph \mathcal{G}_0 induced by the m vertices residing in a square box of side $h(n)$. By defining p_{\min} as before, we have again that the matching process over \mathcal{G}_0 is faster than the matching process over $G(m, p_{\min})$. However, since $r = \Theta(1)$, now we have:

$$m p_{\min} = n h^k(n) K(n) C^\beta(n) h^{-\beta}(n) = \Gamma(n)$$

with $\Gamma(n)$ being an arbitrarily slow function such that $\Gamma(n) \rightarrow \infty$ so that condition $p_{\min} > m^{-1}$ is automatically satisfied. Deriving $h(n)$, we obtain:

$$h(n) = \left(\frac{\Gamma(n)}{[n C^\beta(n) K(n)]} \right)^{\frac{1}{k-\beta}}. \quad (11)$$

Note that condition $p_{\min} \ll m^{\frac{3.5}{r}}$ can be easily met by selecting a sufficiently slow $\Gamma(n)$. Also, we have to impose:

$$a_0 = \Theta \left(\frac{1}{(m p_{\min}^r)^{\frac{1}{r-1}}} \right) = \Theta \left(\frac{1}{[n h^k(n)]^{\frac{1}{r-1}} [h^{-\beta}(n) K(n) C^\beta(n)]^{\frac{r}{r-1}}} \right) \quad (12)$$

so as to guarantee a successful matching over $G(m, p_{\min})$, hence \mathcal{G}_0 , and, by Theorem 2, over the corresponding subset in \mathcal{G}_T . Finally, a successful matching of good pairs on \mathcal{G}_T can be achieved following the same rationale described in Section 7.2.

7.4. Case $\beta \leq k/2$

Since we assume $D(n) = \Theta(n K(n) C^\beta(n)) = O(n^{1/2-\delta})$ with $\delta > 0$, necessarily we have $\mu = n C^{2\beta}(n) K^2(n) = O(n^{-2\delta})$. Thus, again we can properly fix a finite r and apply PGM to match almost all good pairs within a range $\Theta(h(n))$ (with $h(n)$ defined as in (11)) under the same condition on a_0 as in (12). (See Alg. 2 for a summary of the procedure to be applied.)

8. HIGH-DENSITY CLUSTERS

We now consider clusters with high graph density, i.e., with $K(n) = \omega([n C^k(n)]^{-\gamma}) \forall \gamma > 0$. We first focus on the case where $\beta > k/2$ and analyze the seed set size that is required for successful graph de-anonymization. Specifically, we consider the following two cases:

- Case $B(n) = \Omega(\log^{1-\xi} n)$, $\forall \xi > 0$ (Sec. 8.1): $a_0 = O([\max\{n C^k(n), \log n\}]^\epsilon)$, for any $\epsilon > 0$. In this case (large values of $B(n)$), the graph may have many cliques or quasi-cliques of nodes. Thus, matching nodes using only the local structure of a cluster, may lead to a high number of errors. For a successful graph de-anonymization we therefore match nodes only on the basis of the ‘fingerprint’ provided by their longer edges. In particular, we first trigger the matching procedure using two sub-regions of side $\Theta(C(n))$, which are sufficiently far from each other and include a suitable number of seeds. To identify the nodes therein, we apply a modified version of PGM algorithm, which considers only edges spanning between the two sub-regions. Then, by exploiting the fact that in the high-density cluster regime the distance between two nodes can be estimated quite precisely, we select a set of compact nodes that are sufficiently far from a matched sub-region, and re-apply the direct matching strategy with a properly chosen r . The procedure can be iterated until almost all nodes are matched.
- $B(n) = o(\log^{1-\xi} n)$, for some $0 < \xi < 1$ (Sec. 8.2): $a_0 = \left(\frac{\frac{\log n}{\log \log n}}{K(n) C^\beta(n) h^{-\beta}(n)} \right)$. Here the edge density within a cluster is not very large, thus we safely apply PGM directly on the original pairs graph, using seeds in an opportunely defined sub-region and by selecting a proper threshold r .

At last (Sec. 8.3), we note that $\beta \leq k/2$ does not represent a meaningful case for high-density clusters, since it cannot be matched with the above condition on $K(n)$.

8.1. Case $B(n) = \Omega(\log^{1-\xi} n)$, $\forall \xi > 0$, and $\beta > k/2$

First, observe that $B(n) = \Omega(\log^{1-\xi} n)$ (for any $\xi > 0$) implies $nC^k(n) = \Omega(\log^{1-\xi} n) \forall \xi > 0$, given that $K(n) \leq 1$. Second, since $\mu = B(n)K(n)$, and $K(n) = \omega([nC^k(n)]^{-\gamma})$, $\forall \gamma > 0$ (or, equivalently $K(n) = \omega(\log^{-\gamma} n)$, $\forall \gamma > 0$), μ does not meet conditions (8)–(9), and we cannot exploit the corresponding values of r to guarantee a successful matching process. Indeed, as mentioned in Section 5, this case is significantly different from low-density clusters, and the de-anonymization algorithm should 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).

In order to provide a clear outline of the de-anonymization procedure that we adopt for high-density clusters, we provide the pseudocode in Algorithm 3. We present the details and prove our analytical results in the following sections. In particular, since our approach relies on some results on bipartite graphs, we introduce them in Sec. 8.1.1. Then we apply such results to our clustered social network model (Sec. 8.1.2), and derive the seed set size that is required to trigger the identification process (Sec. 8.1.3).

Algorithm 3 De-anonymization algorithm used in Section 8.1

Require: $\mathcal{G}_1, \mathcal{G}_2, \mathcal{A}_0, \epsilon > 0$

- 1: Identify two squared regions, $\mathcal{H}_1, \mathcal{H}_2 \subset \mathcal{H}$, of side $h(n) = \Omega(C(n))$ and $g(n) = \omega(C(n))$ apart from each other, including $a_0 = O([\max\{nC^k(n), \log n\}]^\epsilon)$ seeds each
- 2: $r \leftarrow 4/\epsilon$
- 3: Build the pairs graph $\mathcal{P}(\mathcal{H}_{12})$ induced by nodes in \mathcal{H}_1 and \mathcal{H}_2
- 4: Apply PGM to $\mathcal{P}(\mathcal{H}_{12})$ using r as above and only inter-region edges % Trigger phase
- 5: **while** No. of newly matched pairs > 0 **do** % Expansion phase
- 6: $\mathcal{H}_2 \leftarrow$ compact set of nodes sufficiently apart from \mathcal{H}_1 and within $\Theta(C(n))$ from each other
- 7: Build the pairs graph $\mathcal{P}(\mathcal{H}_{12})$ induced by nodes in \mathcal{H}_1 and \mathcal{H}_2
- 8: Update r as in Theorem 7 and use matched nodes in \mathcal{H}_1 as seeds
- 9: Consider inter-region edges only and match all pairs in $\mathcal{P}(\mathcal{H}_{12})$, whose constituent nodes lie in \mathcal{H}_2 , with at least r already-matched neighboring pairs, whose constituent nodes lie in \mathcal{H}_1
- 10: **return** Set of matched pairs

8.1.1. Results on bipartite graphs. Let \mathcal{G}_T be an $m_1 \times m_2$ bipartite graph. Let \mathcal{M}_1 denote the set of vertices on the left hand side (LHS), with $|\mathcal{M}_1| = m_1$, and \mathcal{M}_2 the set of vertices on the right hand side (RHS), with $|\mathcal{M}_2| = m_2$. We assume that for any pair of vertices $i \in \mathcal{M}_1$ and $j \in \mathcal{M}_2$ 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 the required number of seeds a_0 located in either side of the graph, i.e., with $a_0 = |\mathcal{A}_0^l|$ in \mathcal{M}_1 and $a_0 = |\mathcal{A}_0^r|$ in \mathcal{M}_2 , such that vertices in \mathcal{M}_1 and \mathcal{M}_2 can be correctly matched.

Let us first consider the case where $m_1 = m_2 = m$, for which the theorem below holds.

Theorem 6. 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 $a_0 > a_c$, 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 four conditions listed in Theorem 1.

PROOF. See Appendix H. \square

Theorem 6 can be extended to the general case where $m_1 \neq m_2$, as stated in the corollary below.

Corollary 2. Assume that \mathcal{G}_T is an $m_1 \times m_2$ bipartite graph and define $m = \min(m_1, m_2)$. Under the same assumptions of Theorem 6, 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 a 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 6 and applying Corollary 1. \square

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

Theorem 7. Consider that \mathcal{G}_T is an $m_1 \times m_2$ bipartite graph with $m_1 = \omega(\sqrt{m_2})$ and that a seed set \mathcal{A}_0^l is available on the LHS of the graph, with $|A_0^l| = a_0 = \Theta(m_1)$. With probability larger than $1 - e^{-\frac{m_1}{\sqrt{m_2}}}$, all the m_2 good pairs on the RHS can be successfully identified with no errors, provided that:

- (1) $\frac{1}{\sqrt{m_2}} \ll p_{\min} \leq p_{\max} \ll 1$
- (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 pairs graph comprising a finite fraction of all possible pairs on the RHS.

PROOF. Without loss of generality, we assume $a_0 \geq cm_2$ for some $c > 0$. 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 concentration results in App. A and union bound, we have:

$$\begin{aligned} \mathbb{P}(\text{all good pairs on the RHS have at least } r \text{ adjacent seeds}) &\geq 1 - m_2 e^{-cm_1 p_{\min} H(\frac{1}{2})} \\ &\geq 1 - e^{-\frac{m_1}{\sqrt{m_2}}} \end{aligned}$$

which imply that all good pairs on the RHS are successfully matched since $m_1 = \omega(\sqrt{m_2})$. Similarly, considering a bad pair $[j_1, k_2]$ on the RHS, the number of its adjacent seeds on the LHS is $E[N_b] \leq cm_2 (p_{\max})^2 \ll r$. Thus, by applying concentration results in App. A and union bound, we have:

$$\begin{aligned} \mathbb{P}(\text{all bad pairs on the RHS have less than } r \text{ adjacent seeds}) &\geq 1 - m_2^2 e^{-cm_1 \frac{p_{\min}}{4} \log\left(\frac{p_{\min}}{(p_{\max})^2}\right)} \\ &\geq 1 - e^{-\frac{m_1}{\sqrt{m_2}}}. \end{aligned}$$

\square

8.1.2. The de-anonymization procedure. We now outline how our proposed matching algorithm for *high-density clusters* works. We start by focusing on single vertices in \mathcal{H} so as to identify regions in \mathcal{H} where PGM can be successfully applied. Specifically, we consider two hyper-cubic regions, $\mathcal{H}_1, \mathcal{H}_2 \subset \mathcal{H}$, 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 $i \in \mathcal{H}_1$ and $j \in \mathcal{H}_2$, $p_{\min} = K(n)f(g(n) + \sqrt{k}h(n)) \leq p_{ij} \leq K(n)f(g(n)) = p_{\max}$. Let us assume $p_{\max} = \eta p_{\min}$ for some constant $\eta > 1$.

We then extract vertices in \mathcal{H}_1 and \mathcal{H}_2 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}_1 and \mathcal{H}_2 so

that Theorem 4 given in Section 7.1 can be applied (or Corollary 4 given in Appendix E when $C^k(n) = o\left(\frac{\log n}{n}\right)$).

At this point, we consider the pairs graph originated from the above bipartite graph. In this regard, observe that we can use the same procedure as in Section 7, 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., $m_1 < m_2$, 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 7.

To further expand the set of good pairs that are matched, 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 high-density structure of the clusters.

Proposition 1. *Assume $nC^k(n)K^2(n) = \omega(1)$, given two nodes in region \mathcal{H} , it is possible to estimate with arbitrary precision their mutual distance d as far as*

$$d \ll C(n) (nC^k(n)K^2(n))^{\frac{1}{\beta}}.$$

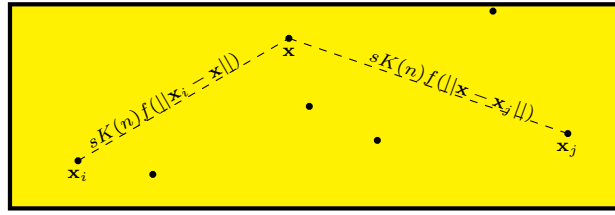


Fig. 5. Computation of $\mathbb{E}[N_{ij}]$.

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, we have:

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

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 $\mathbb{E}[N_{ij}] \rightarrow \infty$ as long as $d \ll C(n) (nK^2(n)C^k(n))^{\frac{1}{\beta}}$, the assertion follows. \square

We can therefore use the number of common neighbors between two given nodes as an estimator of their distance. We then set two thresholds,

$$\begin{aligned} d_L &= \Theta\left(\max\left\{C(n) \log[n^{\frac{1}{k}}C(n)], \log^{\frac{1}{k}} n\right\}\right) \\ d_H &= \lambda d_L \quad \text{with } \lambda > 1 \end{aligned}$$

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 $K(n) = \omega([nC^k(n)]^{-\gamma}) \forall \gamma > 0$, and $nC^k(n) = \Omega(\log^{1-\xi} n) \forall \xi > 0$. 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 the proof in Appendix G (see [Chiasserini et al. 2015a] for further details).

At this point, we consider a bipartite graph whose LHS is still represented by \mathcal{H}_1 , and whose RHS is given by the nodes that are connected with those in \mathcal{H}_1 through edges of length comprised between d_L and d_H . Again, we consider the pairs graph originated from them and apply Theorem 7 so as to 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 $K(n) = \omega([nC^k(n)]^{-\gamma}) \forall \gamma > 0$ and $nC^k(n) = \Omega(\log^{1-\xi} n)$, $\forall \xi > 0$. 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 [Chiasserini et al. 2015a]).

8.1.3. Seed set size. To explicitly derive the required 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 6 and Corollary 2.

First, recalling (1) and Theorem 1, we have:

$$a_c = \left(1 - \frac{1}{r}\right) \left(\frac{(r-1)!}{m(p_{\min} s^2)^r}\right)^{\frac{1}{r-1}} \leq \left(\frac{r-1}{(mp_{\min} s^2)^{\frac{1}{r-1}} p_{\min} s^2}\right) \leq \frac{r}{p_{\min} s^2}. \quad (13)$$

Thus, a_c can be minimized by maximizing p_{\min} , i.e., by minimizing $g(n)$ (recall that $p_{\min} = K(n)f(g(n) + \sqrt{k}h(n))$). However, $g(n)$ and $h(n)$ must also be selected in such a way that condition 1) of Theorem 6 is met. Additionally, as mentioned, it must be ensured that $h(n) = \Omega(C(n))$. At last, by standard concentration results, m_1 and m_2 turn out to be both $\Theta(nh^k(n))$ provided that $h(n) \geq (\log n/n)^{1/k}$.

Previous considerations suggest to fix:

$$h(n) = \Theta(\max\{C(n), (\log n/n)^{1/k}\}) \geq (\log n/n)^{1/k}$$

(i.e., the minimum possible value in order sense), which corresponds to having $m = \Theta(\max\{nC^k(n), \log n\})$ (recall that $m = \min(m_1, m_2)$). We then derive $g(n)$ by forcing $p_{\max} \approx m^{-\frac{\alpha}{r}}$, with $3.5 < \alpha < 4$ and $r \geq 4$. Note that condition 1) of Theorem 6 is met since p_{\max} and p_{\min} are both $\Theta(m^{-\frac{\alpha}{r}})$. Hence, we have:

$$\begin{aligned} p_{\max} &= \Theta([\max\{nC^k(n), \log n\}]^{-\frac{\alpha}{r}}) \\ g(n) &= \Theta\left(C(n)[\max\{nC^k(n), \log n\}]^{\frac{\alpha}{\beta}} [K(n)]^{\frac{1}{\beta}}\right) \end{aligned}$$

Given the above expression for p_{\max} , considering that $p_{\max} = \eta p_{\min}$ and using (13), the seed set size can be made as small as $a_0 = O([\max\{nC^k(n), \log n\}]^\epsilon)$, for any $\epsilon > 0$, by choosing

$r > \frac{4}{\epsilon}$. Finally, we remark that the obtained a_0 is in order sense greater than the number of seeds needed to apply Theorem 4 while selecting nodes in regions \mathcal{H}_1 and \mathcal{H}_2 , thus the whole construction is consistent.

8.2. Case $B(n) = o(\log^{1-\xi} n)$, for some $0 < \xi < 1$, and $\beta > k/2$

First, we observe that $B(n) = o(\log^{1-\xi} n)$ implies $nC^k(n) = o(\log^{1-\xi} n)$. On its turn, this implies $\beta < k$ so as to guarantee that $D(n) = \Omega(\log n)$. In this case, $\mu = o(\log^{1-\xi} n)$, thus we can select r as in (8) and apply the PGM directly on the pairs graph $\mathcal{P}(\mathcal{G}_T)$, without worrying about errors. The same procedure outlined in Alg. 2 holds in this case.

The idea is to show that PGM successfully matches almost all good pairs on an opportune bipartite subgraph of \mathcal{G}_T . To show this, we have to extend Theorem 3 to bipartite graphs, as done in Corollary 3.

Corollary 3. *Consider a subgraph of $\mathcal{G}_T(\mathcal{V}, \mathcal{E})$ denoted by $\mathcal{G}_0(\mathcal{V}_0, \mathcal{E}_0)$, which is an $m_1 \times m_2$ bipartite graph with $m_1 = \Theta(m_2)$. Assume that, for any pair of vertices $i \in \mathcal{M}_1$ and $j \in \mathcal{M}_2$, 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 $\eta > 1$.*

Under (8), applying the PGM algorithm to $\mathcal{P}(\mathcal{G}_T)$ (with $r = \Theta(\frac{\log n}{\log \log n})$) guarantees that $m - o(m)$ good pairs are matched with no errors, if a_0 randomly chosen seeds among vertices in \mathcal{M}_1 and in \mathcal{M}_2 are available, provided that

- (1) $m_1 \rightarrow \infty$;
- (2) $m_1 p_{\min} \gg r$;
- (3) $a_0 = \Theta(\frac{r}{p_{\min}})$.

PROOF. The proof follows the same lines as the proof of Theorem 3. \square

It follows that we can trigger percolation on \mathcal{G}_T , whenever we can find two boxes of side $h(n)$ separated by $g(n)$ so that the induced bipartite graph satisfies the assumptions of Corollary 3, i.e., we place a sufficiently large number of seeds a_0 in each of the boxes.

In particular, we select:

$$h(n) = g(n) \geq \max[C(n), \log n/n] \quad (14)$$

obtaining $m_1 = \Theta(nh^k(n))$, $m_2 = \Theta(m_1)$ and

$$\begin{aligned} p_{\min} &= \Theta(K(n)C^\beta(n)h^{-\beta}(n)) \\ p_{\max} &= \Theta(p_{\min}) \end{aligned}$$

Now, to satisfy condition 2 of Corollary 3, we must impose $m_1 p_{\min} \gg r$. By following the same steps as in Section 7.2, we obtain a condition on the number of seeds a_0 to be placed in each box:

$$a_0 = \left(\frac{\frac{\log n}{\log \log n}}{K(n)C^\beta(n)h^{-\beta}(n)} \right).$$

Note that Corollary 3 ensures that, after the execution of PGM, almost all nodes in the boxes of side $h(n)$ are correctly identified. Similarly to the procedure reported in Section 7.2, to ensure that we can further expand the set of identified nodes, we can invoke Theorems 2 and 7 by setting $h(n) = g(n)$.

8.3. Case $\beta \leq k/2$

This case is impossible. Indeed the assumptions i) $\beta \leq \frac{k}{2}$ and ii) $K(n) = \omega([nC^K(n)]^{-\gamma})$ for every $\gamma > 0$ contrast with the assumption that $D(n) = O(n^{\frac{1}{2}-\delta})$ for some $\delta > 0$. Indeed, by i) and ii) and given that $C(n) = \Omega(n^{-\frac{1}{k}})$, we have $D(n) = \Theta(nK(n)C^\beta(n)) = \Omega(nK(n)n^{-\frac{1}{2}}) = \omega(n^{\frac{1}{2}-\delta})$.

Table II. Main results on seed set size for sample cases of low-density and high-density clusters and for different conditions on $B(n)$ and β/k

Scenario	Conditions		
Low-density $K(n) = n^{-\alpha}, \alpha > 0$	Conditions		
	$B(n) = n^{\zeta-\alpha}$ $\beta \geq k/2$	$B(n) = \Theta(1)$ $k/2 < \beta < k$	$B(n) = o(n^{\zeta-\alpha})$ $(\zeta < \alpha < 1 - \beta \frac{1-\zeta}{k}) \wedge \beta < k$
	Required seed set size		
	$\Theta(n^\alpha \log n)$	$\omega\left(n^\alpha \left[\frac{\log n}{\log \log n}\right]^{\frac{k}{k-\beta}}\right)$	$n^{\frac{\alpha(1+r)-2}{r-1} + \frac{\beta(1-\zeta)}{k-\beta}}$
High-density $K(n) = 1$	Conditions		
	$B(n) = n^\zeta$ $\zeta > 0 \wedge \beta > k/2$	$B(n) = 1$ $k/2 < \beta < k$	$\beta \leq k/2$
	Required seed set size		
	$n^\epsilon, \forall \epsilon > 0$	$\omega\left(\frac{\log n}{\log \log n}\right)^{1+\frac{\beta}{k-\beta}}$	Impossible

9. EXPERIMENTAL VALIDATION

A summary of our analytical results obtained when $K(n) = n^{-\alpha}$ and $B(n) = n^{\zeta-\alpha}$ ($\alpha, \zeta \geq 0$) is reported in Table II. The table highlights the trend of the required seed set size, in some special cases that will be explored in our experimental validation. Although our results hold asymptotically as $n \rightarrow \infty$, 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, allowing us to assess the validity of our results for the different considered regimes. Next, we consider real social network graphs, exploring also variants and improvements of matching algorithms.

9.1. 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} .

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

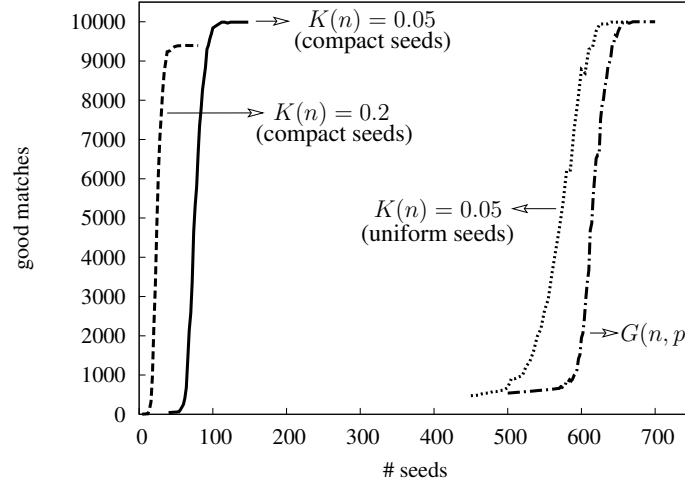


Fig. 6. Comparison of PGM performance (with $r = 5$) in different networks with $n = 10,000$. Number of good matches (averaged over 1,000 runs) as a function of the number of seeds, chosen either uniform or compact.

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 $\beta = 4, 3, 2.5, 2.2, 2.0, 1.8$, are respectively, 11, 15, 24, 45, 78, 138. Recall that the percolation threshold in a $G(n, p)$ with the same average node degree and the same value of r is about 600.

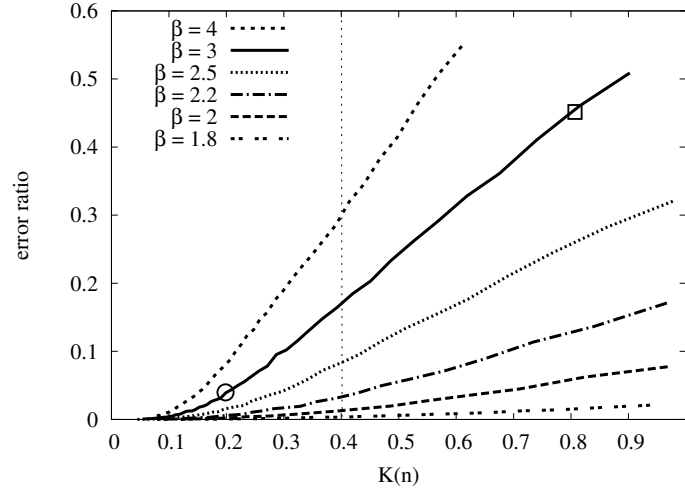


Fig. 7. Error ratio of PGM as a function of $K(n)$ for different values of β , starting from compact seeds.

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 high-density clusters, where the performance of the original PGM algorithm is rather poor (error ratio about 50%). Fig. 8 shows the average

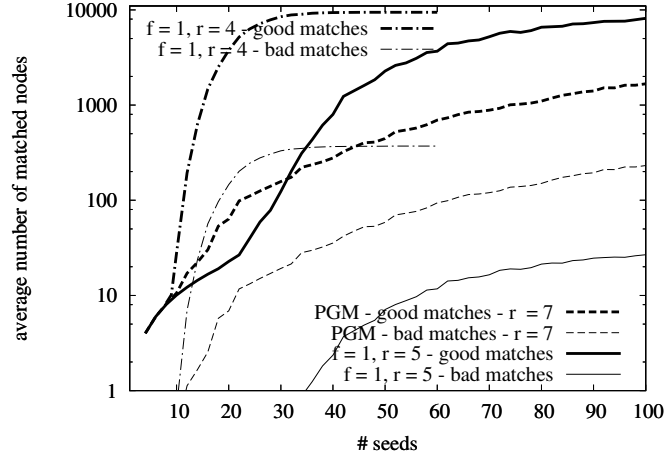


Fig. 8. Average number of good and bad pairs matched by different algorithms for $K(n) = 0.8$, $\beta = 3$, starting from compact seeds.

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 analyzed in Section 8. 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 ‘ $f = \langle x \rangle$ ’. We stress that filtered PGM approaches the performance that can be obtained by the algorithm in Section 8.

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 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 transition⁹. 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 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 III reports an example of this numerical exploration, in which

⁹We verified that, if we instead fix the very first seed across all runs, a sharp transition appears. However, the transition threshold changes as we vary the initial seed (results not shown here).

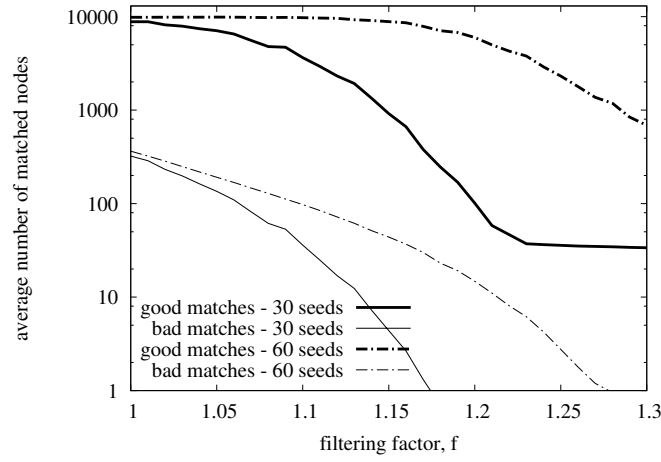


Fig. 9. Effect of varying the filtering factor f for fixed $r = 4$ (scenario with $K(n) = 0.8$).

Table III. 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

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 III validate, at least qualitatively, the counter-intuitive theoretical predictions in Table II: 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.

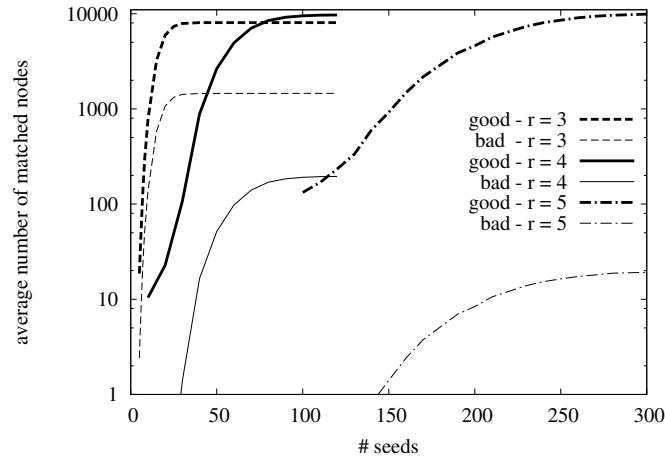


Fig. 10. Effect of varying r for fixed filtering factor $f = 1$ (scenario with $K(n) = 0.5$, $\beta = k = 4$).

At last, we considered a higher dimensional ground-truth graph with $k = 4$ dimensions. Here, we selected $\beta = k$, corresponding to the especially interesting case where neighbors of a node are equally distributed at all distance scale, a condition that allows efficient navigability by decentralized algorithm [Kleinberg 2000]. For this experiment, we chose $K(n) = 0.5$, $f = 1$. Fig. 10 shows the impact of using different thresholds $r = 3, 4, 5$. We clearly see a trade-off between critical number of seeds and error ratio: $r = 3$ requires few tens of seeds, but produces about 14% error ratio; $r = 5$ requires hundreds of seeds, but lowers the error ratio to 0.2%.

9.2. Real social graphs

We first experimented with a rather small ($n = 2539$) network representing adolescent friendship, created from a survey that took place in 1994/1995. In the survey, students of an American high-school were asked to list their best 10 friends (5 female and 5 male) [hea 1995]. Hence, students' answers are expected to describe the real ground-truth of their friendship relationship. From this data we generated an undirected network in which an edge between two students is present if at least one of them indicated the other in his/her list. We obtained a graph with two desirable properties for our purposes: i) a significant clustering coefficient (0.14); ii) a rather peaked degree distribution around the mean (equal to 8.23), so as to isolate the impact of clustering from that related to highly skewed distributions. Given the small size of the graph, we have used $s = 0.9$, $r = 3$. Fig. 11 shows the performance of two algorithms: the original PGM starting from compact seeds, and the filtered PGM in which we removed the edges connecting each node to its 'nearest' 4 neighbors, estimated using the number of common neighbors. As reference, we show also the performance of PGM in a $G(n, p)$ graph with the same n and average degree as our real-world graph (but negligible clustering). We observe that clustering helps PGM to percolate with fewer seeds as compared to the reference $G(n, p)$ graph, but producing 10% error ratio. By filtering out 4 neighbors for each node, we require more seeds, but we incur only 5% of errors.

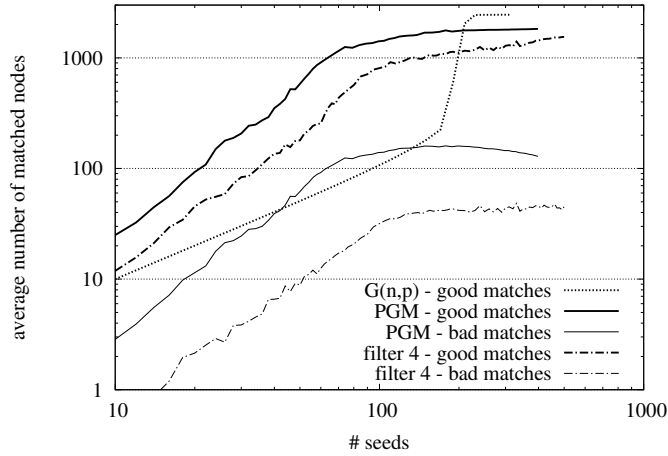


Fig. 11. Performance of matching algorithms in an adolescence friendship network of 2539 students.

We next considered a much larger graph derived from the Slovak social network Pokec. The public data set, available at [pok 2015], is a directed graph with 1,632,803 vertices, 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, 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 well¹⁰.

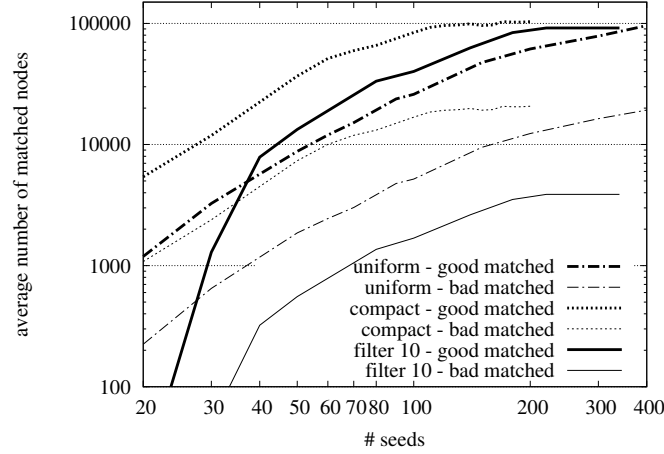


Fig. 12. Performance of matching algorithms in a subset of the friendship graph of the social network Pokec.

Fig. 12 shows the performance of the different algorithms using threshold $r = 6$. As before, curves labeled ‘uniform’ refer to the PGM algorithm in which seeds are selected uniformly at random among the nodes. Curves labeled ‘compact’ refer to the PGM algorithm in which seeds are chosen among the closest neighbors of a uniformly selected node. Curves labeled ‘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).

At last, we take as ground-truth graph an early snapshot of Facebook, first analyzed in [Viswanath et al. 2009], containing 63,731 nodes, with average node degree 25.64, maximum node degree 1,098. This snapshot exhibits a quite large clustering coefficient (14.8%), making careless matching algorithms particularly prone to errors. This is confirmed by the results in Fig. 13, obtained for $s = 0.75$. The PGM algorithm (curves labelled uniform) results into an intolerable error ratio with $r = 6$ (40%). Increasing the threshold to $r = 12$ brings down the error ratio of PGM to about 20%, but requires quite a large number of seeds (above 1000).

We then experimented with the ‘compact’ variant of PGM in which seeds are chosen among the closest neighbors of a uniformly selected node (as before, closeness between nodes is estimated by counting the number of common neighbors). Moreover, edges connecting each node to its nearest 30 neighbors are not used. As shown by the leftmost curves on Fig. 13, this variant of PGM reduces the number of seeds necessary to trigger wide-scale percolation by roughly an order of magnitude,

¹⁰In direct networks, counters of matchable pairs are incremented only by using outgoing edges from matched pairs.

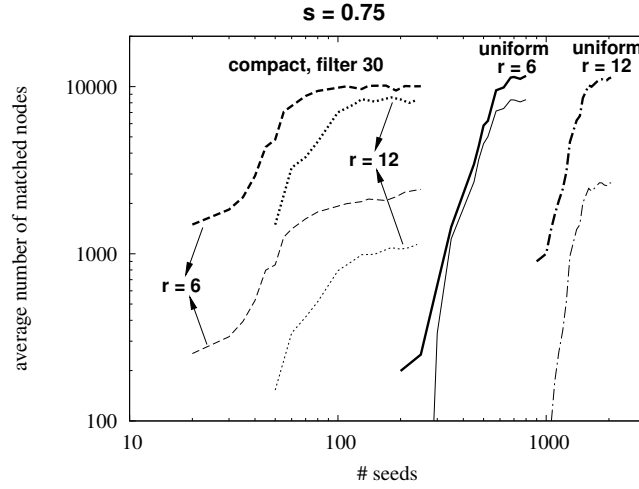


Fig. 13. Performance of matching algorithms on the Facebook graph, in the case of $s = 0.75$, for $r = 6, 12$.

confirming the great benefit achieved by starting from a compact set of seeds. Errors are also reduced, but even by filtering quite a large number of neighbors (30) the error ratio is still quite high, equal to about 20% (10%) with $r = 6$ ($r = 12$).

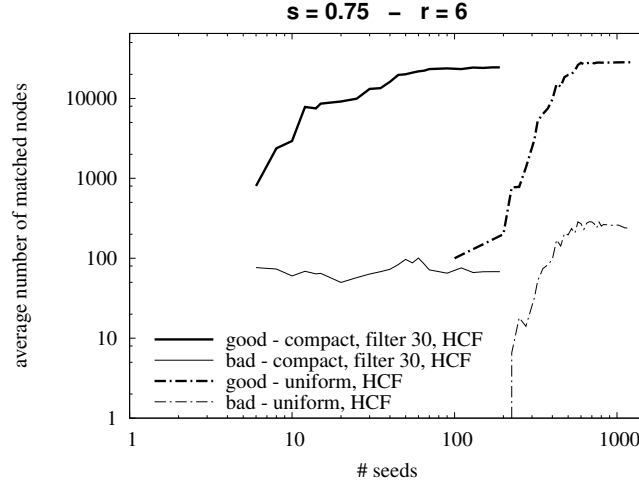


Fig. 14. Performance of HCF variants of matching algorithms on the Facebook graph, in the case of $s = 0.75$, $r = 6$.

A simple approach to reduce the error ratio, adopted in many proposed algorithms [Korula and Lattanzi 2014; Yartseva and Grossglauser 2013], is to select the next pair of nodes to match as the candidate pair with the highest mark count, rather than uniformly at random among the pairs with counter larger than or equal to r . This strategy is called *deferred matching variant* of PGM in [Yartseva and Grossglauser 2013], and a similar idea is exploited by the algorithm proposed in [Korula and Lattanzi 2014]. In the following, we will refer to this strategy as HCF (Highest Count First). Note that, while it is significantly difficult to precisely characterize the gain achievable by HCF with respect to the case in which a random candidate pair is selected, the scaling order of the critical number of seeds does not change, as argued in [Yartseva and Grossglauser 2013]. In Fig. 14 we compare the performance of different algorithms employing HCF, in the same scenario

considered in Fig. 13, with $r = 6$. We observed a dramatic reduction of the error ratio: the deferred PGM algorithm (curves labelled ‘uniform, HCF’) achieves an error ratio of about 1%, while its compact variant with filtering further reduces the error ratio to 0.3%. Moreover, similarly to what we obtained in Fig. 13, the compact variant requires just a few seeds (say a few tens) to achieve its maximum performance.

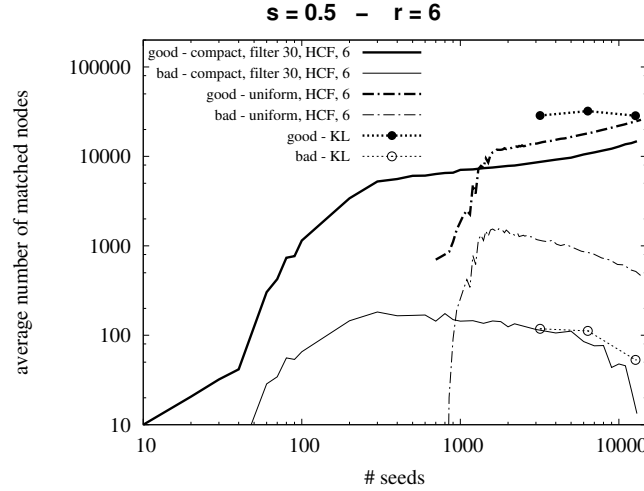


Fig. 15. Performance of HCF variants of matching algorithms on the Facebook graph, in the case of $s = 0.5$, $r = 6$.

Matching performance is particularly sensitive to the thinning probability s , which, however, does not affect the scaling order of the critical number of seeds. In Fig. 15 we show what happens on the Facebook graph when we reduce s from 0.75 to 0.5, again fixing $r = 6$. Besides requiring many more seeds (knee at about 1500 seeds), deferred PGM incurs significant errors (12% at the knee), which however tend to diminish as we further increase the number of seeds. The compact, filtered PGM requires much fewer seeds (knee at about 300 seeds), incurring an error ratio around 3% (at the knee). We also show the performance achieved by the algorithm proposed in [Korula and Lattanzi 2014], shown by dots labelled KL. In [Korula and Lattanzi 2014], the authors report results for a quite large number of seeds (several thousands of seeds), resulting into an error ratio comparable to the one obtained by our compact variant (around 1%). Note that, using fixed $r = 6$, our algorithm cannot match nodes with small degree, which explains why the KL algorithm is able to correctly match more nodes.

At last, in Fig. 16 we show the results of an experiment in which we have used different sampling probabilities for \mathcal{G}_1 and \mathcal{G}_2 . For a direct comparison with what we obtained in the case of equal sampling probabilities, we selected $s_1 = 0.75$, $s_2 = 0.5$. As expected, we get results lying between those obtained with equal $s = 0.75$ (Fig. 14) and those achieved for $s = 0.5$ (Fig. 15).

10. 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 analyzed 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.

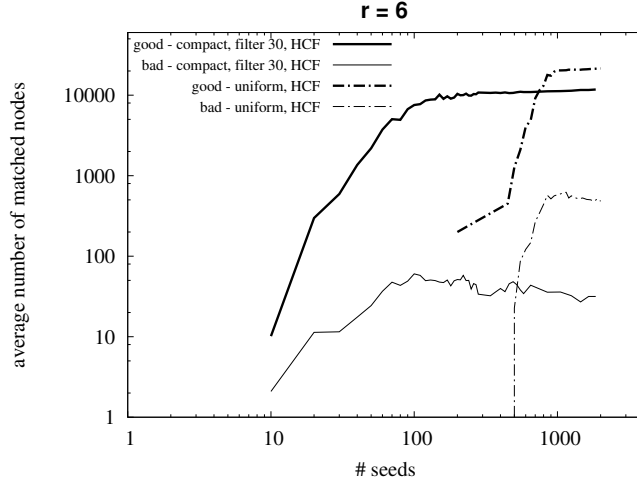


Fig. 16. Performance of HCF variants of matching algorithms on the Facebook graph, in the case of $s_1 = 0.75$, $s_2 = 0.5$, $r = 6$.

Our work can be extended along several directions, including the effect of erroneously selected seeds, more general graph models and partially overlapping node sets, as done in [Kazemi et al. 2015a; Kazemi et al. 2015b] for Erdős–Rényi graphs.

APPENDIX

A. DEVIATION BOUNDS FOR THE BINOMIAL DISTRIBUTION

In this paper we frequently use some classical deviation bounds for the binomial distribution (see e.g. Lemma 1.1 p. 16 in [Penrose 2003]), which we report here for the reader's convenience. Define the function $H(x) := 1 - x + x \log x$, for $x > 0$. Suppose $n \in \mathbb{N}$, $p \in (0, 1)$ and $0 < k < n$. Let $\mu = np$. If $k \geq \mu$ then

$$P(\text{Bin}(n, p) \geq k) \leq \exp \left(-\mu H \left(\frac{k}{\mu} \right) \right) \quad (15)$$

and if $k \leq \mu$ then

$$P(\text{Bin}(n, p) \leq k) \leq \exp \left(-\mu H \left(\frac{k}{\mu} \right) \right). \quad (16)$$

Finally, if $k \geq e^2 \mu$ then

$$P(\text{Bin}(n, p) \geq k) \leq \exp \left(- \left(\frac{k}{2} \right) \log \left(\frac{k}{\mu} \right) \right). \quad (17)$$

B. CLUSTERING COEFFICIENT

We focus on a single vertex, which, for the sake of simplicity, is assumed to be placed at the origin of region \mathcal{H} . Then, we compute the average number of its neighbor pairs that are connected, i.e.,

$$T(n) = \frac{(n-1)(n-2)}{2} K^3(n) \int \int f(\|x_1\|) f(\|x_2\|) \cdot f(\|x_2 - x_1\|) dx_1 dx_2.$$

The clustering coefficient is then given by $\frac{2T(n)}{D(n)(D(n)-1)}$.

$T(n)$ can be evaluated as follows¹¹:

$$T(n) \sim n^2 K^3(n) \int \int_{\|x_1\| < \|x_2\|} f(\|x_1\|) f(\|x_2\|) \cdot f(\|x_2 - x_1\|) dx_1 dx_2.$$

To compute (in order sense) the above integral, we partition the integral domain in two regions distinguishing the following two cases: i) $\|x_1 - x_2\| \geq \delta \|x_2\|$ for an arbitrarily chosen $\delta > 0$ and ii) the complementary case $\|x_1 - x_2\| < \delta \|x_2\|$. In the first case, by triangular inequality we have $\delta \|x_2\| \leq \|x_1 - x_2\| \leq \|x_2\| + \|x_1\| \leq 2\|x_2\|$, and thus $f(\|x_2 - x_1\|)$ is in order sense equal to $f(\|x_2\|)$ over the considered domain. As a consequence, for the integral over the first region, we get:

$$\begin{aligned} T_1(n) &\sim n^2 K^3(n) \int \int_{\substack{\|x_1\| < \|x_2\| \\ \|x_1 - x_2\| > \delta \|x_2\|}} f^2(\|x_2\|) f(\|x_1\|) dx_1 dx_2 \\ &\sim n^2 K^3(n) \int_0^1 \rho_2^{k-1} \max\left(1, \left(\frac{C(n)}{\rho_2}\right)^{-2\beta}\right) \int_0^{\rho_2} \rho_1^{k-1} \max\left(1, \left(\frac{C(n)}{\rho_1}\right)^{-\beta}\right) d\rho_1 d\rho_2. \end{aligned}$$

It is easy to see that a similar expression is obtained when the integral extends over the domain for which $\|x_2 - x_1\| < \delta \|x_2\|$. In particular, considering that in this case $f(\|x_1\|) \sim f(\|x_2\|)$ by construction, and denoting with $\rho_3 = \|x_1 - x_2\|$, we have:

$$T_2(n) \sim n^2 K^3(n) \int_0^1 \rho_2^{k-1} f^2(\|\rho_2\|) \int_0^{\delta \rho_2} f(\|\rho_3\|) d\rho_1 d\rho_3$$

In conclusion, we obtain:

$$T(n) = T_1(n) + T_2(n) = \begin{cases} \Theta(n^2 K^3(n) C(n)^{2k}) & \beta > \frac{2k}{3} \\ \Theta\left(n^2 K^3(n) C(n)^{2k} \log \frac{1}{C(n)}\right) & \beta = \frac{2k}{3} \\ \Theta(n^2 K^3(n) C(n)^{3\beta}) & \beta < \frac{2k}{3}. \end{cases}$$

Recalling (4), for $\beta > k$ the clustering coefficient turns out to be $\Theta(K(n))$. Similarly, for $\beta = k$, it is $\Theta\left(\frac{K(n)}{\log^2[1/C(n)]}\right)$. For $\frac{2k}{3} < \beta < k$, it is $\Theta(K(n) C(n)^{2(k-\beta)})$, while for $\beta = \frac{2k}{3}$ it is $\Theta(K(n) C(n)^\beta \log[1/C(n)])$. Finally, for $\beta < \frac{2k}{3}$, the clustering coefficient is $\Theta(K(n) C(n)^\beta)$.

C. PROOF OF THEOREM 1

The theorem assumes $p_{\min} \gg m^{-1}$. In Theorem 2 in [Chiasserini et al. 2016], we considered the case $p_{\min} \gg \sqrt{m^{-3/r-1}}$, with $r \geq 4$. Here, instead, we consider the complementary case $p_{\min} = O(\sqrt{m^{-3/r-1}})$. With reference to PGM algorithm reported in Algorithm 4, we define:

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

Next, we define

$$T_{G_{p_{\min}}} = \min\{t \text{ s.t. } |\mathcal{A}_t(G(m, p_{\min}))| = t\}; \quad T_{G_{p_{\max}}} = \min\{t \text{ s.t. } |\mathcal{A}_t(G(m, p_{\max}))| = t\}$$

¹¹Given f and g , $f \sim g$ denotes that $f = \Theta(g)$

Algorithm 4 The PGM algorithm

```

1:  $\mathcal{A}_0 = \mathcal{B}_0 = \mathcal{A}_0(n)$ ,  $\mathcal{Z}_0 = \emptyset$ ,  $t = 0$ 
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 neighbor pairs of  $[*_1, *_2]$ 
     in  $\mathcal{M}(\mathcal{G}_T)$ .
5:   Let  $\Delta\mathcal{B}_t$  be the set of all neighbor 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  $\Delta\mathcal{A}_t$ :
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  $T = t$ ,  $\mathcal{Z}_T = \mathcal{A}_T$ 

```

By Lemma 3.1, we have that both $T_{G_{p_{\min}}}$ and $T_{G_{p_{\max}}}$ are equal to $m - o(m)$. Then inductively on t , $\forall t < \min(T_{G_{p_{\min}}}, T_{G_{p_{\max}}})$, w.h.p.:

$$|\mathcal{B}_t''(\mathcal{G}_T)| \leq |\mathcal{B}_t''((G(m, p_{\max})))| = \emptyset. \quad (18)$$

In (18), the inequality descends by monotonicity of sets \mathcal{B}_t'' with respect to “ \leq_{st} ”. The following equality descends from Corollary 1 in [Chiasserini et al. 2016] applied to $G_{p_{\max}}$. We remark that, under our assumption on p_{\min} and p_{\max} , we have $t_0 = T$ in Corollary 1 in [Chiasserini et al. 2016], along with:

$$|\mathcal{A}_t(\mathcal{G}_T)| \stackrel{(a)}{=} |\mathcal{B}_t'(\mathcal{G}_T)| \stackrel{(b)}{\geq} |\mathcal{B}_t'(G(m, p_{\min}))| \stackrel{(c)}{=} |\mathcal{A}_t(G(m, p_{\min}))| \stackrel{(d)}{>} t. \quad (19)$$

In (19), equality (a) is an immediate consequence of (18), inequality (b) holds by monotonicity of sets \mathcal{B}_t' with respect to “ \leq_{st} ”, while equality (c) descends from Lemma 3.1. Inequality (d) descends from the fact that we assume $t < T_{G_{p_{\min}}}$.

Thus, necessarily, $\mathcal{A}_T(\mathcal{G}_T) = T \geq \min(T_{G_{p_{\min}}}, T_{G_{p_{\max}}}) = m - o(m)$ and $\mathcal{B}_T''(\mathcal{G}_T) = \emptyset$.

D. PROOF OF COROLLARY 1

Essentially the scheme of Theorem 1 can be repeated to show that there exists $t_1 < T$ such that $\mathcal{B}_{t_1}(\hat{\mathcal{P}})$ comprises all good pairs in $\hat{\mathcal{P}}$ and no bad pairs. First observe that $\hat{\mathcal{P}}$ can always be transformed into $\mathcal{P}(\mathcal{G}_T'')$, being \mathcal{G}_T'' a proper subgraph of \mathcal{G}_T' (and therefore of \mathcal{G}_T), by adding and removing only bad pairs.

Second, from Theorem 1 we know that, denoted by \bar{N} the number of vertices in \mathcal{G}_T'' , for a $t_1 = \frac{(\bar{N})^{-3/r-\epsilon}}{(p_{\min}s)^2} = o(\bar{N})$, it holds: $\mathcal{B}_{t_1}'(\mathcal{P}(\mathcal{G}_T'')) = \bar{N}$ (equal, by construction, to the number of good pairs in $\hat{\mathcal{P}}$). Third, again from Theorem 1, it holds that $\mathcal{B}_{t_1}''(\mathcal{P}(\mathcal{G}_T'')) = \emptyset$.

Hence, if we prove that $\mathcal{B}_{t_1}''(\hat{\mathcal{P}}) = \emptyset$, we can conclude that $\mathcal{B}_{t_1}'(\hat{\mathcal{P}}) = \bar{N}$ since condition $\mathcal{B}_{t_1}''(\mathcal{P}(\mathcal{G}_T'')) = \emptyset$ necessarily implies $\mathcal{B}_t'(\mathcal{P}(\mathcal{G}_T'')) = \mathcal{B}_t'(\hat{\mathcal{P}})$ for every $t \leq t_1$. Indeed, by construction, the subgraphs of $\mathcal{P}(\mathcal{G}_T'')$ and $\hat{\mathcal{P}}$ induced by their good pairs are identical.

To prove that $\mathcal{B}_{t_1}''(\hat{\mathcal{P}}) = \emptyset$, we can upper-bound the number of marks collected at time t by every bad pair $[i_1, j_2] \in \hat{\mathcal{P}}$ with a binomial r.v. $\text{Bi}(t, p_{\max}^2 s^2)$ and, then, proceed exactly as in the proof of Corollary 1 in [Chiasserini et al. 2016] to show that $\mathbb{P}\{\mathcal{B}_{t_1}''(\hat{\mathcal{P}}) \neq \emptyset\} \rightarrow 0$. Given that, by

construction, a bad pair $[i_1, j_2]$ can be included in $\hat{\mathcal{P}}$ only if either $[i_1, i_2]$ or $[j_1, j_2]$ are also in $\hat{\mathcal{P}}$, then, none of the bad pairs in $\hat{\mathcal{P}}$ can be matched for $t > t_1$, because it will be necessarily blocked by a previously matched good pair.

E. PROOF OF THEOREM 2 AND COROLLARY 4

First we assume that PGM $\mathcal{G}_0(\mathcal{V}_0, \mathcal{E}_0)$ successfully matches all good pairs. Consider the evolution of PGM over $\mathcal{G}_T(\mathcal{V}, \mathcal{E})$ and $\mathcal{G}_0(\mathcal{V}_0, \mathcal{E}_0)$. Since (8) holds we can assume that no bad pairs reach the threshold at any stage of the process, and restrict our analysis to good pairs. With abuse of notation we say that a good pair belongs to \mathcal{V}_0 if the corresponding vertex belongs to \mathcal{V}_0 .

First, observe that if we assume that at time t $Z_{\mathcal{G}_T}(t) = Z_{\mathcal{G}_0}(t)$, necessarily $\mathcal{A}_{\mathcal{G}_0}(t) = \mathcal{A}_{\mathcal{G}_T}(t) \cap \mathcal{V}_0$ and, thus, $|\mathcal{A}_{\mathcal{G}_T}(t)| \geq |\mathcal{A}_{\mathcal{G}_0}(t)| > 0$. However, in general, we cannot assume $Z_{\mathcal{G}_T}(t) = Z_{\mathcal{G}_0}(t)$ at every t . Indeed, PGM operating over \mathcal{G}_T can select and match at some point some good pairs not belonging to \mathcal{V}_0 , which have already reached the threshold, while the PGM operating over \mathcal{V}_0 cannot. We conclude that in general $Z_{\mathcal{G}_T}(t) \neq Z_{\mathcal{G}_0}(t)$.

Let us consider $t \leq |\mathcal{V}_0|$. Since $|Z_{\mathcal{G}_T}(t)| \leq |Z_{\mathcal{G}_0}(t)| = t$ by construction, necessarily $Z_{\mathcal{G}_0}(t) \setminus Z_{\mathcal{G}_T}(t) \neq \emptyset$. Let $\tau = \min_{t' \leq t} z_{\mathcal{G}_0}(t') \notin Z_{\mathcal{G}_T}(t)$. Note that $z_{\mathcal{G}_0}(\tau)$ must be in $\mathcal{A}_{\mathcal{G}_0}(\tau - 1)$ and thus also in $\mathcal{A}_{\mathcal{G}_T}(t) \cup Z_{\mathcal{G}_T}(t)$. Indeed, since by construction $Z_{\mathcal{G}_0}(\tau - 1) \subset Z_{\mathcal{G}_T}(t)$, we have $Z_{\mathcal{G}_0}(\tau - 1) \cup \mathcal{A}_{\mathcal{G}_0}(\tau - 1) \subseteq Z_{\mathcal{G}_T}(t) \cup \mathcal{A}_{\mathcal{G}_T}(t)$. This implies that $z_{\mathcal{G}_0}(\tau) \in \mathcal{A}_{\mathcal{G}_T}(t)$, i.e., $\mathcal{A}_{\mathcal{G}_T}(t) \neq \emptyset$. By induction over t , we can then prove that $\mathcal{A}_{\mathcal{G}_T}(t) \neq \emptyset$ for any $t \leq |\mathcal{V}_0|$, hence $T_{\mathcal{G}_T} \geq |\mathcal{V}_0|$. Now, consider $t > |\mathcal{V}_0|$. Since $Z_{\mathcal{G}_0}(t) = Z_{\mathcal{G}_0}(T_{\mathcal{G}_0}) = \mathcal{V}_0$ by construction, we have that either $\mathcal{V}_0 \subseteq Z_{\mathcal{G}_T}(t)$ or, necessarily, $Z_{\mathcal{G}_0}(t) \setminus Z_{\mathcal{G}_T}(t) \neq \emptyset$. In the latter case, we can repeat the previous argument to show that $\mathcal{A}_{\mathcal{G}_T}(t) \neq \emptyset$. It follows that $\mathcal{V}_0 \subseteq Z_{\mathcal{G}_T}(T_{\mathcal{G}_T})$. The extension to the more general case in which PGM over $\mathcal{G}_0(\mathcal{V}_0, \mathcal{E}_0)$ matches almost all good pairs, can be proved along the same lines.

A more general version of the previous theorem is given below for the case where the threshold α is a function of n .

Corollary 4. Assume that a target distance $D(n) \gg C(n)$ can be found, with $D(n) \geq \left(\frac{\log n}{n}\right)^{\frac{1}{k}}$, satisfying the following condition:

$$nD^k(n) = \Omega\left(\frac{\log(nD^k(n))}{K(n)f(D(n))}\right).$$

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 tagged as “accepted” if $S_i > f(D(n))sK(n)a_0$. If $d_s = O(D(n))$ and $a_0 = \Theta\left(\frac{\log[nD^k(n)]}{K(n)f(D(n))}\right)$, then, for an arbitrary $\delta > 0$, the above procedure accepts all nodes located in $\mathcal{H}_{in}(f(D(n)), \delta)$, while it rejects all nodes located in $\mathcal{H}_{out}(f(D(n)), \delta)$.

PROOF. The proof follows exactly the same lines as the proof of Theorem 4. \square

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 shorter. Thus, $\alpha sK(n)a_0$ provides a suitable threshold for the number of connections between a node and the a_0 seed vertices.

F. PROOF OF THEOREM 3

Assume that $r = \frac{\log n}{\log \log n}$, i.e., (8) holds. This implies that we can disregard bad pairs and the graph de-anonymization process corresponds to a pure bootstrap percolation process on the subgraph of $\mathcal{P}(\mathcal{G}_T)$ induced by good pairs.

Also, it is straightforward to see that the percolation probability over a graph is monotonically increasing with respect to the graph ordering relation “ \leq_{st} ” (see Section 3 for the ordering relation definition). This because at every t we have $\mathcal{A}(t) = \mathcal{A}'(t) = \mathcal{B}'(t)$, in light of (8), and $\mathcal{B}'(t)$ is

obviously monotonic with respect to \leq_{st} . Thus, since by construction $G(m, p_{\min}) \leq_{st} \mathcal{G}_0$, if percolation successfully occurs on $G(m, p_{\min})$, then it takes place successfully also over \mathcal{G}_0 .

To prove that the matching process is successful on $G(m, p_{\min})$, we follow an approach similar to that in [Janson et al. 2012]. Consider the evolution of \mathcal{A}_t ; it is clear that percolation stops when $\mathcal{A}(t)$ becomes empty. Since $\mathcal{A}(t)$ includes the seed set, clearly it cannot be empty till $t = a_0$. Then, recalling that $a_0 > \frac{r}{p_{\min} s^2} (1 + \delta)$, we should consider $t > \frac{r}{p_{\min} s^2} (1 + \delta)$.

We first observe that the probability that a good pair has a number of matched neighbors greater than or equal to r at time t is $\text{Bin}(t, p_{\min} s^2)$. For $t_1 = \lceil \frac{r(1+\delta)}{p_{\min} s^2} \rceil$,

$$\mathbb{P}(\text{Bin}(t_1, p_{\min} s^2) \geq r) \geq 2c$$

with c being a suitable small positive constant. This holds because the expectation of the Binomial is by construction equal to $\lceil \frac{r(1+\delta)}{p_{\min} s^2} \rceil p_{\min} s^2 > r$. Thus, using concentration results in App. A, we can easily show that w.h.p. $\mathcal{A}(t_1) > cn$, hence $\mathcal{A}(t) > \mathcal{A}(t_1) > cn \geq t \forall t \in [t_1, cn]$. Now, consider $t \in [cn, n(1 - e^{-r})]$ and define $t_2 = \lfloor cn \rfloor$. Observe that, given our choice of r , we have $n(1 - e^{-r}) = n - o(n)$. Moreover, applying again inequalities in App. A, we have:

$$\mathbb{P}(\text{Bin}(t_2, p_{\min} s^2) < r) \leq \exp\left(-t_2 p_{\min} s^2 H\left(\frac{r}{t_2 p_{\min} s^2}\right)\right) \leq \exp\left(-t_2 p_{\min} s^2 \left[1 - \frac{\delta}{2}\right]\right)$$

where $H(b) = 1 - b + b \log b$. The above probability tends to 0 faster (in order sense) than e^{-r} . It follows that $\mathbb{E}[n - \mathcal{A}(t)] \leq \mathbb{E}[n - \mathcal{A}(t_2)] < e^{-t_2 p_{\min} s^2 (1-\delta)}$, thus, by Markov inequality $n - \mathcal{A}(t) \leq n - \mathcal{A}(t_2) < n e^{-r}$ w.h.p. In conclusion, w.h.p. the percolation process does not stop for $t \in [cn, n(1 - e^{-r})]$.

G. PROOF OF THEOREM 4

Without loss of generality, let us focus on \mathcal{G}_1 and 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 $S_i = \sum_{\sigma \in \mathcal{A}_0} X_{i\sigma} S_{i\sigma}^1 \geq_{st} Y_i \geq_{st} Y$ 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 inequalities in App. A, we can bound:

$$\mathbb{P}(Y_i < \alpha sK(n)a_0) \leq \exp\left(-\mathbb{E}[Y_i] H\left(\frac{\alpha sK(n)a_0}{\mathbb{E}[Y_i]}\right)\right) \leq \exp\left(-(1 + \delta)\alpha sK(n)a_0 H\left(\frac{1}{1 + \delta}\right)\right) \quad (20)$$

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:

$$\mathbb{P}(\text{all nodes in } \mathcal{H}_{\text{in}} \text{ are accepted} \mid N_{\text{in}}) \leq 1 - N_{\text{in}} \exp\left(-(1 + \delta)\alpha sK(n)a_0 H\left(\frac{1}{1 + \delta}\right)\right), \quad (21)$$

with (21) that tends to 1 if $\log N_{\text{in}} - (1 + \delta)\alpha sH\left(\frac{1}{1 + \delta}\right)K(n)a_0 \rightarrow -\infty$. 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 in App. A. As a consequence, $\mathbb{P}(\text{all vertices in } \mathcal{H}_{\text{in}} \text{ are accepted}) \rightarrow 1$ 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, in order to obtain 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 $\mathcal{H}_{\text{out}}^1 = \mathcal{H}^1(\alpha, \frac{e^2-1}{e^2}) \subset \mathcal{H}_{\text{out}}(\alpha, \delta)$ and $\mathcal{H}_{\text{out}}^0(\alpha, \delta) = \mathcal{H}_{\text{out}}(\alpha, \delta) \setminus \mathcal{H}_{\text{out}}^1$. Furthermore, we partition $\mathcal{H}_{\text{out}}^1$ into disjoint sub-regions, i.e., $\mathcal{H}_{\text{out}}^1 = \cup_{h \geq 1} \mathcal{H}_{\text{out}}^{1,h}$, with $\mathcal{H}_{\text{out}}^{1,h} = \mathcal{H}_{\text{out}}(\alpha, \frac{h^\beta e^2-1}{h^\beta e^2}) \setminus \mathcal{H}_{\text{out}}(\alpha, \frac{(h+1)^\beta e^2-1}{(h+1)^\beta e^2})$. Now, given a vertex i in $\mathcal{H}_{\text{out}}^0(\mathcal{H}_{\text{out}}^1)$, the number of its neighbor seeds S_i on \mathcal{G}_1 can be bounded from above by a $\text{Bin}(a_0, sK(n)(1-\delta)\alpha)$ ($\text{Bin}(a_0, \frac{sK(n)}{h^\beta e^2}\alpha)$). Furthermore, by elementary geometrical arguments, it can be shown that: i) $|\mathcal{H}_{\text{out}}^0| = \Theta(C^k(n))$, ii) $|\mathcal{H}_{\text{out}}^{1,h}| = \Theta(C^k(n))$ and iii) $\mathcal{H}_{\text{out}}^{1,h} = \Theta(h^{k-1}\mathcal{H}_{\text{out}}^{1,1})$.

Denoted with N_{out}^0 and $N_{\text{out}}^{1,h}$ the number of nodes in $\mathcal{H}_{\text{out}}^0$ and $\mathcal{H}_{\text{out}}^{1,h}$, respectively, by exploiting again inequalities in App. A, w.h.p. we have:

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

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

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

For every h , $N_{\text{out}}^{1,h} \leq 2n|\mathcal{H}_{\text{out}}^{1,h}| = \Theta(nh^{k-1}C^k(n))$; also, the number of sub-regions of $\mathcal{H}_{\text{out}}^1$ 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 $2n|\mathcal{H}_{\text{out}}^{1,h}|$. Under the assumption that $a_0 = \Omega\left(\frac{\log[nC^k(n)]}{K(n)}\right)$, it can be easily shown that $\mathbb{P}(\text{all nodes in } \mathcal{H}_{\text{out}}^1 \text{ are rejected}) \rightarrow 1$.

H. PROOF OF THEOREM 6

For any two vertices $i \in \mathcal{M}_1$ and $j \in \mathcal{M}_2$, let X_{ij} be the Bernoulli random variable that represents the presence of an edge $(i, j) \in \mathcal{E}$. By construction, $\text{Ber}(p_{\min}) \leq_{st} X_{ij} \leq_{st} \text{Ber}(p_{\max})$. I.e., two variables \underline{X}_{ij} and \bar{X}_{ij} , with distribution, respectively, $\text{Ber}(p_{\min})$ and $\text{Ber}(p_{\max})$, can be defined on the same probability space as X_{ij} such that $\underline{X}_{ij} \leq X_{ij} \leq \bar{X}_{ij}$ point-wise.

We consider the corresponding pairs graph $\mathcal{P}(\mathcal{G}_T)$, which is, by construction, composed of all the pairs of vertices residing in \mathcal{M}_1 and \mathcal{M}_2 and of the edges connecting pairs of vertices in \mathcal{M}_1 with pairs of vertices in \mathcal{M}_2 . We denote by \mathcal{P}_1 and \mathcal{P}_2 , respectively, the set of pairs of $\mathcal{P}(\mathcal{G}_T)$, whose vertices lie in \mathcal{M}_1 and \mathcal{M}_2 . Observe that, given two good pairs $[i_1, i_2] \in \mathcal{P}_1$ and $[j_1, j_2] \in \mathcal{P}_2$, the presence of an edge in $\mathcal{P}(\mathcal{G}_T)$ is associated with the random variable:

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

where S_{ij}^1 and S_{ij}^2 are mutually independent $\text{Ber}(s)$ r.v.'s, 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 $[i_1, k_2] \in \mathcal{P}_1$ and $[j_1, l_2] \in \mathcal{P}_2$, $Y_{[i_1, k_2], [j_1, l_2]} = X_{ij} X_{kl} S_{ij}^1 S_{kl}^2$, with $p_{\min}^2 s^2 \leq \mathbb{E}[Y_{[i_1, k_2], [j_1, l_2]}] \leq p_{\max}^2 s^2$. Finally, if we consider one good pair and one bad pair (e.g., $[i_1, i_2] \in \mathcal{P}_1$ and $[j_1, k_2] \in \mathcal{P}_2$), $Y_{[i_1, i_2], [j_1, k_2]} = X_{ij} X_{ik} S_{ij}^1 S_{ik}^2$, with $p_{\min}^2 s^2 \leq \mathbb{E}[Y_{[i_1, i_2], [j_1, k_2]}] \leq p_{\max}^2 s^2$.

Recall that we assume that two seed sets, $\mathcal{A}_0^l \in \mathcal{P}_1$ and $\mathcal{A}_0^r \in \mathcal{P}_2$ (with $|\mathcal{A}_0^l| = |\mathcal{A}_0^r|$), are available. On $\mathcal{P}(\mathcal{G}_T)$ we run the PGM algorithm [Yartseva and Grossglauser 2013], opportunely modi-

fied, as follows. At every time step t , we extract uniformly at random one pair $\mathbf{z}^l(t) = [z_1^l, z_2^l]_t \in \mathcal{A}_{t-1}^l \setminus \mathcal{Z}_{t-1}^l$ and $\mathbf{z}^r(t) = [z_1^r, z_2^r]_t \in \mathcal{A}_{t-1}^r \setminus \mathcal{Z}_{t-1}^r$, adding a mark to all the neighbor pairs in \mathcal{P}_2 and \mathcal{P}_1 , respectively. In other words, matched pairs in \mathcal{P}_1 contribute to the mark of pairs in \mathcal{P}_2 and vice versa. Thus, for a generic node pair $[i_1, j_2] \in \mathcal{P}_2 \setminus \mathcal{Z}_t^r$, marks are updated according to the iteration: $M_{[i_1, j_2]}^r(t) = M_{[i_1, j_2]}^r(t-1) + Y_{\mathbf{z}^l(t), [i_1, j_2]}$. Similarly, for $[i_1, j_2] \in \mathcal{P}_1$ marks are updated according to $M_{[i_1, j_2]}^l(t) = M_{[i_1, j_2]}^l(t-1) + Y_{[i_1, j_2], \mathbf{z}^r(t)}$. For the rest, the algorithm proceeds exactly as described in Section 3.

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}_2 and in which the presence of edge $(\mathbf{z}^r(t), [*_1, *_2])$, for any $[*_1, *_2] \in \mathcal{P}_2 \setminus \mathcal{Z}_t^r$, is dynamically unveiled at time t by observing variable $X_{z_1^l(t)*_1} X_{z_2^l(t)*_2} S_{z_1^l(t)*_1}^l S_{z_1^l(t)*_2}^r$. 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.

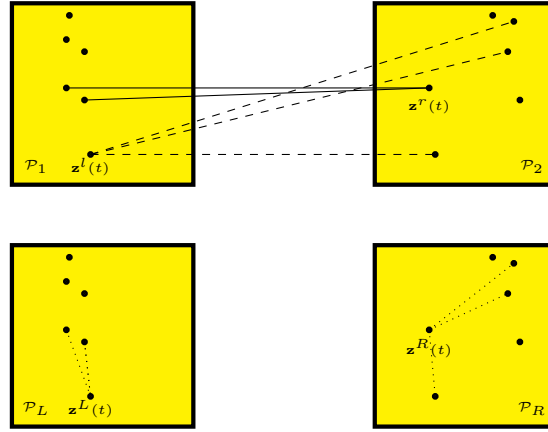


Fig. 17. Graphical representation of the PGM evolution over coupled graphs.

Furthermore, we make the following observations.

(i) We assume that the sequence of matched pairs $\{\mathbf{z}_t^R\}_t \in \mathcal{P}^{(R)}$ exactly corresponds to the sequence of matched pairs $\{\mathbf{z}^r(t)\}_t \in \mathcal{P}_2$, i.e., $\mathbf{z}^r(t) = \mathbf{z}^R(t)$ at every t . This is made possible by the fact that given $\mathcal{Z}_{t-1}^r = \mathcal{Z}_{t-1}^R$, 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 $\mathcal{Z}_t^R = \mathcal{Z}_t^r$. 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 Erdős–Rényi graph \mathcal{G}_T^R , 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 $\mathcal{P}(\mathcal{G}_T)$. Furthermore, by construction, different edges in \mathcal{P}_R correspond to different edges in $\mathcal{P}(\mathcal{G}_T)$.

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 Erdős–Rényi graph \mathcal{G}_T^L with same properties as \mathcal{G}_T^R .

Now, clearly $G(m, p_{\min}) \leq_{st} \mathcal{G}_T^R \leq_{st} G(m, p_{\max})$ and $G(m, p_{\min}) \leq_{st} \mathcal{G}_T^L \leq_{st} G(m, p_{\max})$, i.e., \mathcal{G}_T^R (\mathcal{G}_T^L) can be obtained by opportunistically thinning a graph $G(m, p_{\max})$, while a graph $G(m, p_{\min})$ can be obtained by opportunistically thinning \mathcal{G}_T^R (\mathcal{G}_T^L). Then we invoke Theorem 1 to conclude our proof and show that our algorithm correctly percolates over \mathcal{G}_T^R and \mathcal{G}_T^L and, thus, over the bipartite graph \mathcal{G}_T .

REFERENCES

1995. Add health public data set, wave I (online). (1995). <http://www.cpc.unc.edu/projects/addhealth>
2015. Pokec network dataset – KONECT. (May 2015). <http://konect.uni-koblenz.de/networks/soc-pokec-relationships>
- Fabian Abel, Nicola Henze, Eelco Herder, and Daniel Krause. 2010. Interweaving Public User Profiles on the Web. In *Proceedings of the 18th International Conference on User Modeling, Adaptation, and Personalization (UMAP'10)*. Springer-Verlag, Berlin, Heidelberg, 16–27. DOI: http://dx.doi.org/10.1007/978-3-642-13470-8_4
- Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. 2007. Wherefore Art Thou R3579x?: Anonymized Social Networks, Hidden Patterns, and Structural Steganography. In *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*. ACM, New York, NY, USA, 181–190. DOI: <http://dx.doi.org/10.1145/1242572.1242598>
- Karl Bringmann, Tobias Friedrich, and Anton Krophmer. 2014. *Algorithms - ESA 2014: 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, Chapter De-anonymization of Heterogeneous Random Graphs in Quasilinear Time, 197–208.
- Carla-Fabiana Chiasserini, Michele Garetto, and Emilio Leonardi. 2015a. *Companion technical report*. Technical Report. <https://www.dropbox.com/s/4ddvmgy0zkauj0d/TechRep.pdf?dl=0>
- Carla-Fabiana Chiasserini, Michele Garetto, and Emilio Leonardi. 2015b. Impact of Clustering on the Performance of Network De-anonymization. In *Proceedings of the 2015 ACM on Conference on Online Social Networks (COSN '15)*. ACM, New York, NY, USA, 83–94. DOI: <http://dx.doi.org/10.1145/2817946.2817953>
- C. F. Chiasserini, M. Garetto, and E. Leonardi. 2016. Social Network De-Anonymization Under Scale-Free User Relations. *IEEE/ACM Transactions on Networking* in press (2016).
- A. Egozi, Y. Keller, and H. Guterman. 2013. A Probabilistic Approach to Spectral Graph Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 1 (2013), 18–27. DOI: <http://dx.doi.org/10.1109/TPAMI.2012.51>
- Keith Henderson, Brian Gallagher, Lei Li, Leman Akoglu, Tina Eliassi-Rad, Hanghang Tong, and Christos Faloutsos. 2011. It's Who You Know: Graph Mining Using Recursive Structural Features. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*. ACM, New York, NY, USA, 663–671. DOI: <http://dx.doi.org/10.1145/2020408.2020512>
- Svante Janson, Tomasz Łuczak, Tatyana Turova, and Thomas Vallier. 2012. Bootstrap percolation on the random graph $G_{n,p}$. *Ann. Appl. Probab.* 22, 5 (10 2012), 1989–2047. DOI: <http://dx.doi.org/10.1214/11-AAP822>
- S. Ji, W. Li, N. Z. Gong, P. Mittal, and R. Beyah. 2016. Seed-Based De-Anonymizability Quantification of Social Networks. *IEEE Transactions on Information Forensics and Security* 11, 7 (July 2016), 1398–1411.
- Shouling Ji, Weiqing Li, Mudhakar Srivatsa, and Raheem Beyah. 2014. Structural Data De-anonymization: Quantification, Practice, and Implications. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14)*. ACM, New York, NY, USA, 1040–1053.
- Ehsan Kazemi, S. Hamed Hassani, and Matthias Grossglauser. 2015a. Growing a Graph Matching from a Handful of Seeds. *Proc. VLDB Endow.* 8, 10 (June 2015), 1010–1021. DOI: <http://dx.doi.org/10.14778/2794367.2794371>
- E. Kazemi, L. Yartseva, and M. Grossglauser. 2015b. When can two unlabeled networks be aligned under partial overlap?. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 33–42. DOI: <http://dx.doi.org/10.1109/ALLERTON.2015.7446983>
- Jon Kleinberg. 2000. The Small-world Phenomenon: An Algorithmic Perspective. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing (STOC '00)*. ACM, New York, NY, USA, 163–170. DOI: <http://dx.doi.org/10.1145/335305.335325>
- Nitish Korula and Silvio Lattanzi. 2014. An Efficient Reconciliation Algorithm for Social Networks. *Proc. VLDB Endow.* 7, 5 (Jan. 2014), 377–388. DOI: <http://dx.doi.org/10.14778/2732269.2732274>
- M. Leordeanu and M. Hebert. 2005. A spectral technique for correspondence problems using pairwise constraints. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, Vol. 2. 1482–1489 Vol. 2.
- Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2007. Graph Evolution: Densification and Shrinking Diameters. *ACM Trans. Knowl. Discov. Data* 1, 1, Article 2 (March 2007). DOI: <http://dx.doi.org/10.1145/1217299.1217301>
- S. Melnik, H. Garcia-Molina, and E. Rahm. 2002. Similarity flooding: a versatile graph matching algorithm and its application to schema matching. In *Data Engineering, 2002. Proceedings. 18th International Conference on*. 117–128.
- A. S. Motahari, G. Bresler, and D. N. C. Tse. 2013. Information Theory of DNA Shotgun Sequencing. *IEEE Transactions on Information Theory* 59, 10 (Oct 2013), 6273–6289.

- Arvind Narayanan and Vitaly Shmatikov. 2009. De-anonymizing Social Networks. In *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy (SP '09)*. IEEE Computer Society, Washington, DC, USA, 173–187. DOI : <http://dx.doi.org/10.1109/SP.2009.22>
- André Nunes, Pável Calado, and Bruno Martins. 2012. Resolving User Identities over Social Networks Through Supervised Learning and Rich Similarity Features. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC '12)*. ACM, New York, NY, USA, 728–729. DOI : <http://dx.doi.org/10.1145/2245276.2245413>
- Efe Onaran, Siddharth Garg, and Elza Erkip. 2016. Optimal De-Anonymization in Random Graphs with Community Structure. *CoRR* abs/1602.01409 (Mar 2016). <http://arxiv.org/abs/1602.01409>
- Pedram Pedarsani, Daniel Ratton Figueiredo, and Matthias Grossglauser. 2013. A Bayesian method for matching two similar graphs without seeds. In *Proceedings of the 2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton '13)*. IEEE Computer Society, Washington, DC, USA, 1598–1607. DOI : <http://dx.doi.org/10.1109/Allerton.2013.6736720>
- Pedram Pedarsani and Matthias Grossglauser. 2011. On the Privacy of Anonymized Networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*. ACM, New York, NY, USA, 1235–1243. DOI : <http://dx.doi.org/10.1145/2020408.2020596>
- Wei Peng, Feng Li, Xukai Zou, and Jie Wu. 2014. A Two-Stage Deanonymization Attack against Anonymized Social Networks. *IEEE Trans. Comput.* 63, 2 (2014), 290–303. DOI : <http://dx.doi.org/10.1109/TC.2012.202>
- Mathew Penrose. 2003. *Random geometric graphs*. Oxford University Press, Oxford, New York. <http://opac.inria.fr/record=b1100684> Rimpression : 2004.
- Rohit Singh, Jinbo Xu, and Bonnie Berger. 2008. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences* 105, 35 (2008), 12763–12768. DOI : <http://dx.doi.org/10.1073/pnas.0806627105>
- Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. 2009. On the Evolution of User Interaction in Facebook. In *Proceedings of the 2Nd ACM Workshop on Online Social Networks (WOSN '09)*. 37–42.
- Lyudmila Yartseva and Matthias Grossglauser. 2013. On the Performance of Percolation Graph Matching. In *Proceedings of the First ACM Conference on Online Social Networks (COSN '13)*. ACM, New York, NY, USA, 119–130. DOI : <http://dx.doi.org/10.1145/2512938.2512952>

Received February 2007; revised March 2009; accepted June 2009