Seed-Based and Unseeded Graph Deanonymization: Propagation-Based Alignment and Structural Heuristic Initialization

Apu Kumar Chakroborti

1. Introduction

Graph deanonymization refers to the task of recovering a hidden node mapping between two structurally related graphs. Given two graphs G_1 and G_2, representing two versions of the same underlying social or communication network, the objective is to infer a bijective (or partial) mapping $f : V(G_1) \to V(G_2)$. This problem is central to privacy analysis and re-identification research, as structural patterns often persist across anonymization procedures.

This report presents two strategies:

1. **Seed-Based Deanonymization** – where a small fraction of nodes are known to match across the two graphs. These "seed pairs" are used to propagate further mappings via a structural compatibility measure.

2. **Unseeded Deanonymization** – where no prior information is provided. Initial candidate alignments are estimated using structural heuristics (degree, clustering coefficient, PageRank) and refined using the same propagation mechanism as in the seeded case.

Both approaches share a common refinement stage: an iterative propagation algorithm that exploits neighbor consistency to expand the partial map until convergence. All algorithmic descriptions in this report correspond directly to the code in the provided implementation.

2. Seed-Based Deanonymization

2.1 Initial Seed Mapping

In the seeded model, we assume an initial partial mapping $M_0 = \{(u_i, v_i)\}$.

representing known ground-truth correspondences between G_1 and G_2. This seed plays the role of an anchor: if two nodes have mapped neighbors, then their likelihood of being mapped increases.

2.2 Score Computation

For an unmapped node $x \in V(G_1)$, the algorithm computes a score vector $S(x, y)$ for all $y \in V(G_2)$. using the function `match_scores`. For node x in G_1, we count the alignment-supported structural overlap:

- If a neighbor of x is already mapped to a neighbor of y in G_2, then $S(x, y)$ increases.
- Contributions from incoming and outgoing (or undirected) neighbors are normalized using $1/\sqrt{\deg(y)}$ or $1/\sqrt{\deg(x)}$.
- The formula used to compute the score for a node (x) in G_1 relative to all unmapped nodes (y) in G_2, where some common neighbors are already present: $\frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{\deg(x) \cdot \deg(y)}}$

Thus, score similarity grows when neighbors of x align well to neighbors of y.

2.3 Eccentricity Criterion

To avoid uncertain matches, we apply the eccentricity threshold θ, where $\text{ecc}(\mathbf{S}) = \frac{\max(\mathbf{S}) - \text{2nd-max}(\mathbf{S})}{\sigma(\mathbf{S})}$.

Only if the best score is significantly better than the rest do we accept the candidate.

2.4 Reciprocity Check

Before accepting a mapping $x \mapsto y$, the reverse direction must also agree, i.e., $y \mapsto \arg\max_{x' \in V(G_1)} S(y, x')$.

The mapping is added only if the best reverse-match returns the original node x. This eliminates asymmetric or unstable matches.

3. Unseeded Deanonymization

Unlike the seed-based case, we assume *no prior knowledge* of node correspondences. Therefore, an initial alignment must be estimated using graph structural features.

3.1 Structural Feature Extraction

For each node v in G_1 and G_2, the structural feature vector is $\phi(v) = \begin{bmatrix} \deg(v) \\ C(v) \\ \text{PR}(v) \end{bmatrix}$.

where:

- $\deg(v)$ is the node degree, $C(v)$ is the clustering coefficient, and $\text{PR}(v)$ is PageRank.

Because the features have different scales, each dimension is normalized: $\hat{\phi}(v) = \frac{\phi(v) - \mu}{\sigma}$

3.2 Similarity Matrix

Given the normalized feature matrices for both graphs, we compute:

$$\text{Cosine Similarity}(u, v) = \frac{\hat{\phi}(u) \cdot \hat{\phi}(v)}{\|\hat{\phi}(u)\|\|\hat{\phi}(v)\|},$$

yielding cosine similarity between each pair ($u \in G_1, v \in G_2$).

3.3 Hungarian Algorithm for Unique Initial Mapping

To avoid ambiguous greedy matching, we use the Hungarian algorithm (also known as the Kuhn–Munkres algorithm) for optimal one-to-one alignment. For each graph, we select the top-K highest-degree nodes and compute a cost matrix: $C_{ij} = -\text{Sim}(u_i, v_j)$. Hungarian assignment minimizes cost, equivalently maximizing similarity, giving a clean and globally optimal initial mapping: $M_0 = \{(u_i, v_{\pi(i)})\}_{i=1}^{K}$.

This mapping mimics a pseudo-seed set and is passed to the same propagation algorithm as used in the seeded scenario.

3.4 Propagation as Refinement

After Hungarian alignment, we invoke the full propagation mechanism described in Section 2. The initial mapping serves as an artificial seed set, and propagation expands the matches using neighbor-consistency and reciprocity checks.

This two-stage unseeded method:

1. Estimates seed mappings using structural heuristics.
2. Uses the same propagation procedure as the seed-based case.

In practice, this enables deanonymization even when no initial correspondences are provided.

4. Discussion and Observations

Seed-based propagation tends to outperform the unseeded method because:

- The initial mapping is more accurate.
- Real seeds constrain the matching landscape early.

In contrast, unseeded initialization depends heavily on structural similarity. If the two graphs differ significantly (noise, edge deletions, rewiring), the initial pairing quality decreases. However, the propagation step is effective in refining approximate seeds into more accurate full mappings when the two graphs retain sufficient structural correlation.

Both methods share the same convergence property: the mapping stops expanding when no new pairs satisfy the eccentricity and reciprocity criteria.