Exact Matching of Random Graphs with Constant Correlation

Cheng Mao (Georgia Tech)

Mark Rudelson (University of Michigan)

Konstantin Tikhomirov (Georgia Tech)

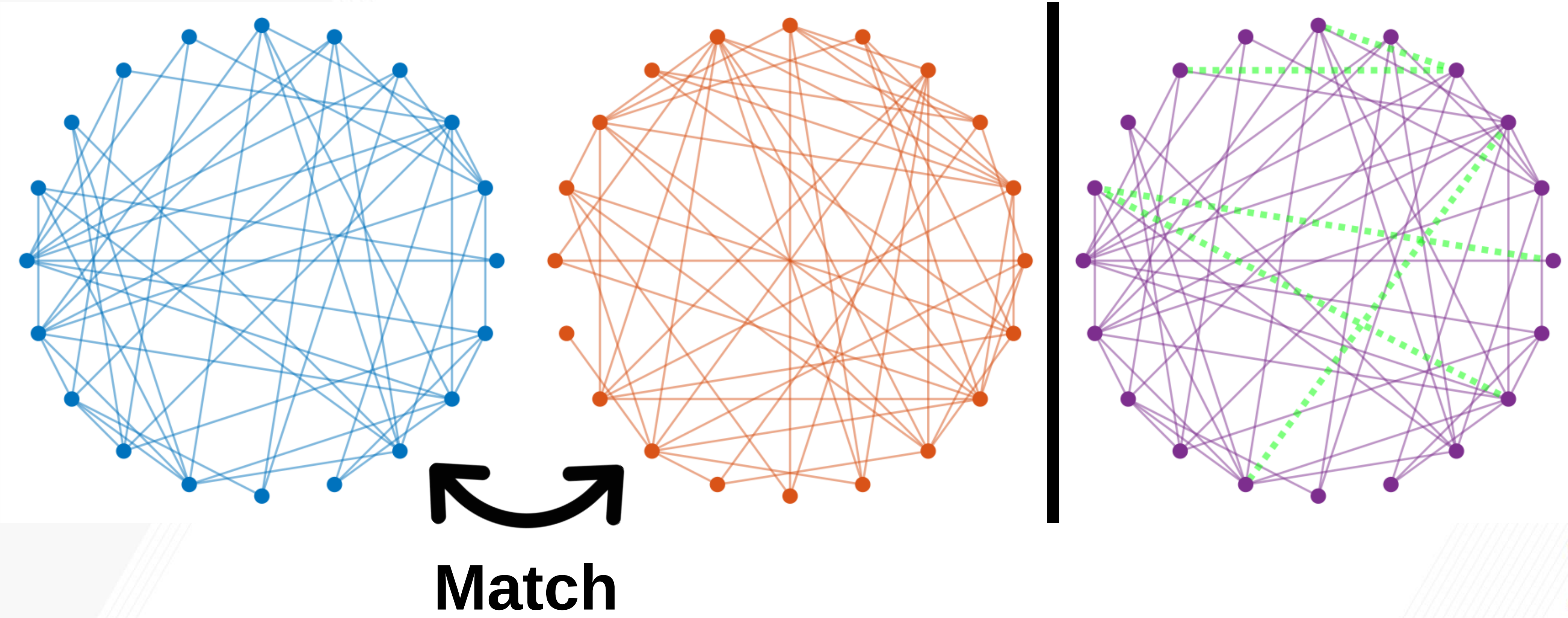
Simons Institute, Berkeley, CA

October 15, 2021

1. Introduction

Graph matching, a.k.a. network alignment

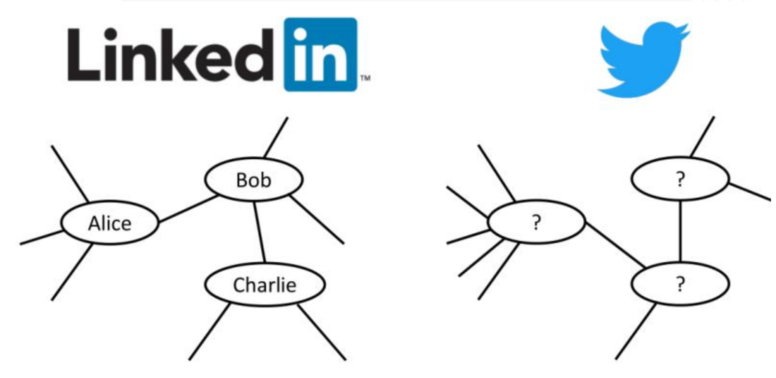
- Given two unlabeled graphs A and B on n vertices
- Match their vertices to maximally align their edges:



Applications

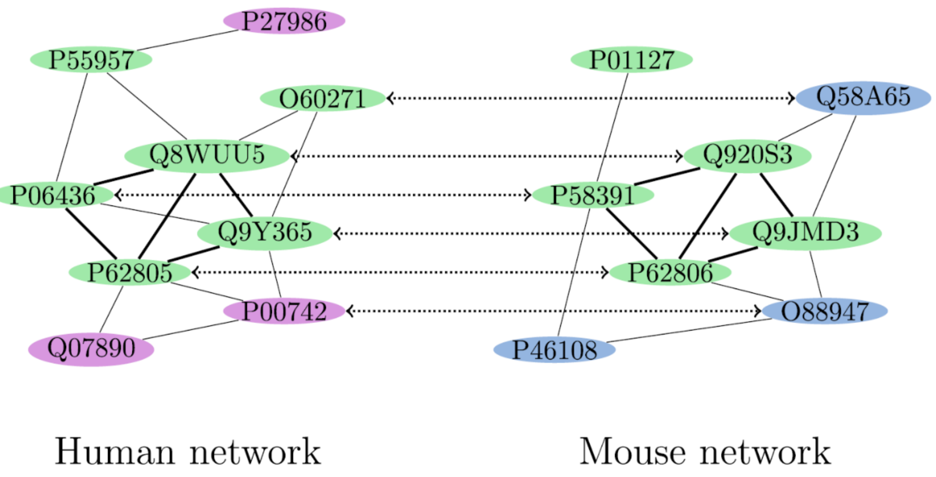
Social Networks

- [Narayanan, Shmatikov 2008, 2009]



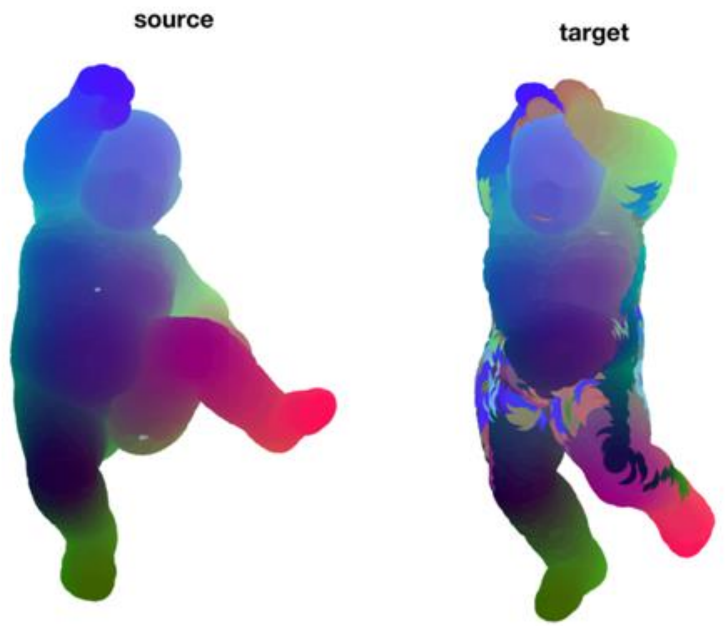
Computational Biology

- [Singh, Xu, Berger 2008; Kazemi et al. 2016]



Computer Vision

- [Lähner et al. 2016; Fan, M., Wu, Xu 2020]



Deterministic formulation

- **Noiseless: graph isomorphism problem**
  - Computational complexity not settled [Babai 2016]
- **Noisy:** Given adjacency matrices $A, B \in \mathbf{R}^{n \times n}$, solve

$$\max_{\pi} \sum_{i=1}^{n} A_{\pi(i)\pi(j)} B_{ij}$$

where $\pi: [n] \to [n]$ is a permutation/matching

  - The **quadratic assignment problem** is NP-hard

2. Model and Result

Correlated Erdős–Rényi graph model [Pedarsani-Grossglauser 11]

- A and B are marginally $G(n,p)$ graphs
- Ground-truth matching π^*
- Define

$$\delta := \mathbb{P}\{B_{ij} = 0 \mid A_{\pi^*(i)\pi^*(j)} = 1\}$$

$$\mathbb{E}[A_{\pi^*(i)\pi^*(j)} B_{ij}] = p(1-\delta)$$

so $\delta \in (0,1)$ is the **noise level** and $1-\delta$ is the **correlation**

- Given (A,B), aim to recover π^* exactly

When is exact recovery possible?

- Connectivity threshold for $A, B \sim G(n, p)$:

$$np \geq (1 + \epsilon) \log n$$

- Intersection of the two graphs $A_{\pi^*} \wedge B \sim G(n, p(1 - \delta))$:

$$np(1 - \delta) \geq (1 + \epsilon) \log n$$

- If $np = 1.1 \log n$, then δ needs to be small constant.

Georgia Tech
CREATING THE NEXT

Selected results for exact recovery

	Condition	Time
[Cullina, Kiyavash 16] [Wu, Xu, Yu 21]	$np(1-\delta) \geq (1+\epsilon)\log n\,, \qquad p \ll 1-\alpha$	exp
[Barak et al. 18]	$1-\delta \geq (\log n)^{-o(1)}, \qquad n^{o(1)} \leq np \leq n^{1-\epsilon}$	quasi-poly
[Ding, Ma, Wu, Xu 18] [Fan, M., Wu, Xu 19]	$\delta \leq (\log n)^{-C}, \qquad np \geq (\log n)^{C}$	poly
[Ding, Ma, Wu, Xu 18] [M., R., T. 21]	$\delta \leq (\log\log n)^{-C}, \qquad np \geq (\log n)^{C}$	poly
This Work	$\delta \leq \delta_0(\epsilon), \qquad (1+\epsilon)\log n \leq np \leq n^{o(1)}$	poly

3. Algorithm and Analysis

Matching via vertex signatures

- Associate each vertex i of A with a **signature** f_i^A
- Do the same for B
- Match vertex i of A and vertex j of B if and only if f_i^A is "close" to f_j^B

Naïve example:

- How about $f_i^A = \deg_i^A$, the degree of i in A ?
- **Issue:** the n degrees for each graph are in

$$(np - C\sqrt{np}, np + C\sqrt{np}\,)$$

Some methods in the literature

- [Ding, Ma, Wu, Xu 18]: same problem, vanishing noise

  Signature: Degree profile, i.e., neighbors' degrees

- [Mossel, Xu 18]: seeded version, constant noise

  Signature: Number of r-neighbors in a seed set

- [Ganassali, Massoulié, Lelarge 20, 21]: partial matching, constant noise

  Signature: Local trees of depth $O(\log n)$

Lesson: Use degree statistics & explore large neighborhoods

Main theorem

- Observe A and B with latent matching π^* (= identity WLOG)
- Average degree: $(1+\epsilon)\log n \le np \le n^{\frac{1}{C\log\log n}}$
- Noise level: $\delta \le \delta_0 \wedge (\epsilon/4)$, $\delta_0 > 0$ small constant
- A new $n^{2+o(1)}$-time algorithm recovers π^* exactly with probability $1 - n^{-\epsilon/10}$

Georgia Tech
CREATING THE NEXT

Step 1: Partition trees

Georgia Tech
CREATING THE NEXT

Partition tree: Structure

- Fix graph A and vertex $i \in \{1, \ldots, n\}$
- $S(i, r)$: r-sphere of i in graph distance
- Construct a **complete binary tree** of depth $m = C \log \log n$

$$T = \{T_\sigma^r : \sigma \in \{-1, +1\}^r,\ r = 1, \ldots, m\}$$

Nodes $T_\sigma^r,\ \sigma \in \{-1, +1\}^r$ form a **partition** of $S(i, r)$

Partition tree: Definition

- $T^0 = \{i\}$
- for $r = 0, \dots, m-1$
  - for $\sigma \in \{-1, +1\}^r$
    - $T^{r+1}_{(\sigma,+1)} = \{j \in N(T^r_\sigma) \cap S(i, r+1) : \deg(j) \geq np\}$
    - $T^{r+1}_{(\sigma,-1)} = \{j \in N(T^r_\sigma) \cap S(i, r+1) : \deg(j) < np\}$

$N(S)$ is the set of neighbors of vertices in S

Overlap between children of a vertex in two graphs

- For a **typical** vertex i
- $|S(i,1)| \approx np$
- $|T^1_{\pm 1}| \approx np/2$
- $\left|T^1_{\pm 1}(i,A) \cap T^1_{\pm 1}(i,B)\right| \approx (np/2) \cdot (1 - \kappa(\delta))$

$\kappa(\delta) \to 0$ as $\delta \to 0$

Overlap between leaves in two graphs

- For a **typical** vertex i, whose m-neighborhood is a tree
- $|S(i,m)| \approx (np)^m$
- $|T_\sigma^m| \approx (np/2)^m$
- $|T_\sigma^m(i,A) \cap T_\sigma^m(i,B)| \approx (np/2)^m \cdot \left(1-\kappa(\delta)\right)^m$

How many typical vertices?

- If $\log n \leq np \leq n^{\frac{1}{C' \log \log n}}$ and $m = C \log \log n$
- With probability $1 - n^{-10}$
- $n - n^{1-c}$ typical vertices whose m-neighborhood are trees

Conclusion

- If $\log n \le np \le n^{\frac{1}{C' \log\log n}}$
- With probability $1 - n^{-10}$, for $n - n^{1-c}$ typical vertices $i \neq j$
- Leaves of partition trees at i in A and i in B have overlap

$$|T_\sigma^m(i,A) \cap T_\sigma^m(i,B)| > (np/2)^m \cdot \big(1 - \kappa(\delta)\big)^m$$

- Leaves of partition trees at i in A and j in B have tiny overlap

Step 2: Vertex signatures

Vertex signature: Definition

- Graph A, vertex i
- Define **signature** $f_i^A \in \mathbf{R}^{2^m}$: For leaf T_σ^m,
- $(f_i^A)_\sigma = \sum_j [\deg(j) - np - 1]$ for $j \in N(T_\sigma^m) \cap S(i, m+1)$

Entrywise difference between vertex signatures

- Recall $|T_\sigma^m(i,A) \cap T_\sigma^m(i,B)| \approx (np/2)^m \cdot \big(1-\kappa(\delta)\big)^m$
- Entrywise difference between signatures: For $i \neq j$,

$$\frac{(f_i^A - f_i^B)_\sigma^2}{\text{variance}} \leq 1 - \big(1-2\kappa(\delta)\big)^m \leq 1 - \frac{1}{\sqrt{\log n}}$$

$$\frac{(f_i^A - f_j^B)_\sigma^2}{\text{variance}} \approx 1$$

Sparsified ℓ_2 difference between vertex signatures

- Sparsification: Take uniform random $I \subset \{-1,+1\}^m$ of size

$$|I| = \mathrm{polylog}(n) \ll 2^m = \mathrm{length}(f_i^A)$$

- Match i and j if and only if

$$\frac{1}{|I|}\sum_{\sigma \in I} \frac{(f_i^A - f_j^B)_\sigma^2}{\text{variance}} \leq 1 - \frac{1}{\sqrt{\log n}}$$

Georgia Tech
CREATING THE NEXT

Conclusion

- If $\log n \leq np \leq n^{\frac{1}{C \log \log n}}$
- Noise $\delta \leq \delta_0$ small constant
- $n - n^{1-c}$ typical vertices i and j are matched correctly
- With probability $1 - n^{-10}$ obtain an almost exact matching $\hat{\pi}$

$$|\{i : \hat{\pi}(i) \neq \pi^*(i)\}| \leq 4n^{1-c}$$

Step 3: Refine to an exact matching

One-step refinement

- Given π_0 such that $|\{i : \pi_0(i) \neq \pi^*(i)\}| \leq \lambda\, n$
- Match $i = \pi_1(j)$ if
  - $N_A(i) \cap \pi_0\big(N_B(j)\big) \geq c\epsilon^2 np$
  - $N_A(i) \cap \pi_0\big(N_B(k)\big) < c\epsilon^2 np$ for all $k \neq j$
  - $N_A(k) \cap \pi_0\big(N_B(j)\big) < c\epsilon^2 np$ for all $k \neq i$
- Extend π_1 to a permutation on $\{1, \dots, n\}$

Georgia Tech
CREATING THE NEXT

Iterative refinement

- With probability $1 - n^{-\epsilon/10}$
- if $|\{i: \pi_0(i) \neq \pi^*(i)\}| \leq \lambda\, n$
- then $|\{i: \pi_1(i) \neq \pi^*(i)\}| \leq \lambda\, n/2$
- $|\{i: \pi_\ell(i) \neq \pi^*(i)\}| \leq \lambda\, n/2^\ell$, for $\ell = 1, 2, \ldots$
- $\pi_{\log_2(n)} = \pi^*$

Conclusion

- Average degree: $(1+\epsilon)\log n \le np \le n^{0.5-\epsilon}$
- Noise level: $\delta \le \epsilon/4$
- Starting from a data-dependent partial matching
- Recover π^* exactly with probability $1 - n^{-\epsilon/10}$

Main theorem

- Observe A and B with latent matching π^*
- Average degree: $(1+\epsilon)\log n \le np \le n^{\frac{1}{C\log\log n}}$
- Noise level: $\delta \le \delta_0 \wedge (\epsilon/4)$
- The $n^{2+o(1)}$-time algorithm recovers π^* exactly with probability $1 - n^{-\epsilon/10}$

4. Discussion

Future directions

- Theory of Erdős–Rényi graph matching
  - Dense graphs, global algorithms
  - Partial recovery, detection [Ganassali, Massoulié 20; Hall, Massoulié 20; Ganassali, Massoulié, Lelarge 21; Wu, Xu, Yu 20; M., Wu, Xu, Yu 21]
- Variations
  - Seeded version [Kazemi, Hassani, Grossglauser 15; Mossel, Xu 18; Yu, Xu, Lin 20]
  - Side information
- Other random graph matching models
  - Universality [Fan, M., Wu, Xu 19]
  - Preferential attachment [Korula, Lanttanzi 14; Racz, Sridhar 20]
  - Correlated stochastic block models [Onaran, Garp, Erkip 16; Racz, Sridhar 20]

Georgia Tech
CREATING THE NEXT

Thank you!

"Exact Matching of Random Graphs with Constant Correlation".
Cheng Mao, Mark Rudelson, Konstantin Tikhomirov.
arXiv preprint arXiv:2110.05000, 2021