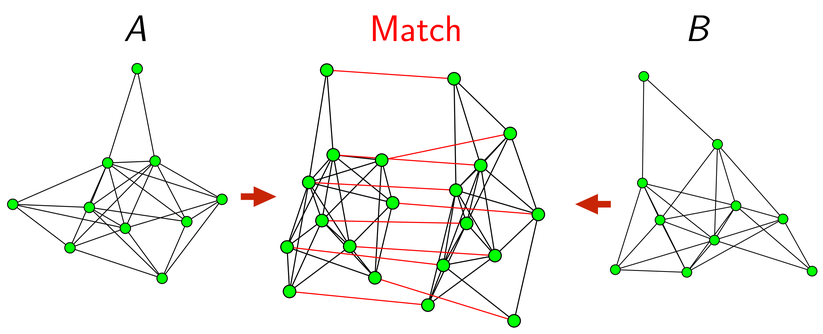
Graph Matching the Matchable Nodes When Some Nodes are Unmatchable

Daniel L. Sussman*, **Vince Lyzinski**†

*Boston University, Math/Stat; †Johns Hopkins University, Applied Math/Stat

What is graph matching (GM)?

- Given two graphs with **overlapping node sets**, match nodes which correspond to each other across graph using network structure.
- Ie. find $P^* = \text{argmin}_P \|A - PBP^T\|_F^2$
  - with $A, B \in \{0,1\}^{n\times n}$, P a permutation matrix.



Challenges

- Graphs with only **partially** overlapping node sets.
- Correctly identifying "matchable"/core nodes.
- Graphs with different numbers of nodes.
- ... and many more.

Algorithms

- Even in ideal settings GM is related to the very hard Quadratic Assignment Problem.
  - Has been researched for many years, see 30, 40, 50 Years of GM review papers.

We use employ an approximate algorithm:

1. Relax to doubly stochastic matrices.
2. Initialize at baricenter.
   - Incorporate seeds—nodes with known correspondence.
3. Gradient ascent maintaining seeds.
   - Each step solves a Linear Assignment Problem.

Note: Seeds are critical as algorithms only find to local minima.

Definition (Correlated Erdős-Rényi)

- We say $A, B \sim \text{CorrER}(\Lambda, R)$ for $\Lambda, R \in [0,1]$ if
  - $A, B \in \{0,1\}^{n\times n}$ are random adjacency matrices
  - with (A_{ij}, B_{ij}) independent for all $i < j$ and
  - $A_{ij}, B_{ij} \sim \text{Bern}(\Lambda_{ij})$ and $\text{corr}(A_{ij}, B_{ij}) = R_{ij}$.

NB: Non-identically distributed graphs require new techniques.

Core+Junk Setting

Very common that the node sets of the two graphs only partially overlap. We model this setting by assuming

- The node set $[n] = \mathcal{C} \cup \mathcal{J} = [n_c] \cup ([n] \setminus [n_c])$.
- For core nodes $u, v \in \mathcal{C}$ it holds that $R_{uv} > 0$.
- For any other pair of nodes, $R_{uv} = 0$.

Theorem (Core Matching)

Let $(A, B) \sim \text{CorrER}(\Lambda, R)$ have n_c core and n_j junk nodes with homogeneous junk. For

$$0 < \epsilon = \min_{i,j\in[n_c]} 2R_{ij}\Lambda_{ij}(1-\Lambda_{ij}),$$

if $\epsilon^2 n_c > C n_j \log(n_j)$, then

$$\mathbb{P}(\textit{any core nodes mis-matched by Oracle}) = e^{-O(\epsilon^2 n_c)}.$$

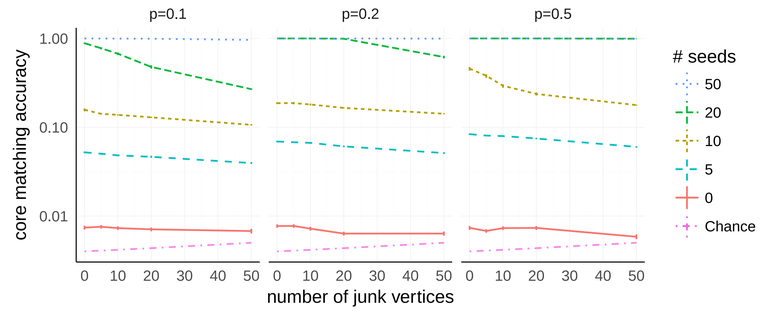


Figure: Core matching accuracy on the log-scale as a function of the number of junk nodes and the number of seeds, for $A, B \sim \text{CorrEr}(pJ, 0.5J_{n_c} \oplus 0_{n_j})$. We believe that performance for an oracle would be similar to the higher seeded setting due to our theory and results regarding soft seeding. Similar results for heterogeneous ER graphs.

Conclusions

- If junk is homogeneous then graph matching can handle up to $\tilde{o}(n_c/\log(n_c))$ junk nodes.
  - Non-homogeneous junk only allows $o(\sqrt{n_c})$ junk nodes.
- Early steps towards core identifications.
  - Theory and improved methods are future work.
- For different sized graphs improved padding allows for $\exp(o(n_c))$ junk nodes.
  - Oracle Centering by Λ allows for arbitrary heterogeneity in core and junk.
- Much work yet to do ....

Core Identification

- Let $\Delta_v(P) = \|(A - PBP^T)_{v\cdot}\|_1$.
- Compute permutation test statistic for each node

$$T(v, P^*) := \frac{\Delta_v(P^*) - \mathbb{E}_P\Delta_v(P)}{\sqrt{\text{Var}_P\Delta_v(P)}}.$$

  - Vertex-wise Mantel Test.
  - Avoids issues with raw measures, e.g. #errors.
- Rank nodes according to T.
  - Alternatively, use unsupervised clustering on T.

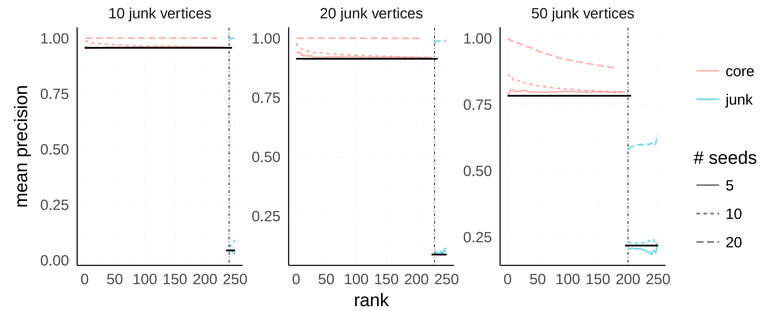


Figure: Mean precision for classifying nodes as core or junk using the permutation statistic. Each graph is generated as with $p = 0.2$ (see left). The horizontal lines indicate the performance of a random classifier for each of core and junk.

Padding Setting

- If the graphs are different sizes
  - $A - B$ doesn't make sense,
  - so we need to "pad" the smaller matrix A.

Padding Options

Naive Set $\tilde{A} = A \oplus 0_{n_j}$.

Improved Set $\tilde{A} = (2A - J_{n_c}) \oplus 0_{n_j}$ and $\tilde{B} = (2B - J_n)$.

Theorem (Different Sized Graphs)

- *Using the improved padding scheme, oracle graph matching will "work" if $R > 1/2 + \epsilon$ and $\log(n_j + n_c) = o(\epsilon^2 n_c)$ (under suitable variance conditions).*
- *Using the naive padding scheme, oracle graph matching can always fail if $n_c < n_j$ and $R < 1$ for suitably adversarial junk structure.*

Simulation results are similar to Core+Junk setting.

See arXiv submission 1705.02294 for details and references.