AFRL-RI-RS-TR-2020-100

UNIVERSALLY USEFUL PRIMITIVES FOR ALIGNING NETWORKS ACROSS TIME AND SPACE

UNIVERSITY OF MASSACHUSETTS AMHERST

JUNE 2020

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2020-100 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
NANCY A. ROBERTS
Work Unit Manager

/ S /
TIMOTHY A. FARRELL
Deputy Chief, Information Intelligence
Systems and Analysis Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
JUNE 2020	FINAL TECHNICAL REPORT	DEC 2017 – JAN 2020

4. TITLE AND SUBTITLE

UNIVERSALLY USEFUL PRIMITIVES FOR ALIGNING NETWORKS ACROSS TIME AND SPACE

5a. CONTRACT NUMBER
FA8750-18-2-0035

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62702E

6. AUTHOR(S)

Vincent Lyzinski
Daniel L. Sussman
Carey E. Priebe
Youngser Park

5d. PROJECT NUMBER
MAXX

5e. TASK NUMBER
00

5f. WORK UNIT NUMBER01
01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Massachusetts Amherst
Lederle Graduate Research Tower
North Pleasant St.
Amherst MA 01003-9306

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIEA
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2020-100

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Their team has, in the course of working on the DARPA Modeling Adversarial Activity (MAA) program, further developed a fast, flexible suite of graph matching tools designed to robustly align large networks in the presence of noise, paying special heed to developing methods for multiplex matching; developed the theory and methodology behind Graph Matching Matched Filters which provide a principled, scalable method for discovering noisy subgraphs in a larger background graph; provided an open source R code-base, denoted iGraphMatch, for implementing our graph matching and graph matching matched filters methods and their competitors at scale; further developed the theory of vertex nomination, developing the analogues of the classical statistical concepts of consistency and Bayes optimality in the context of vertex nomination; developed a novel concept of adversarial contamination and data-adaptive regularization in the context of vertex nomination; developed a suite of flexible vertex nomination algorithms designed to be implemented on large, noisy networks; produced illustrative simulations and data analyses on MAA provided data and on externally provided real data sources.

15. SUBJECT TERMS
Graph analytics, subgraph matching, vertex nomination, adversarial analytics, scalable algorithms, large graph networks

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					NANCY A. ROBERTS
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (*Include area code*)
U	U	U	UU	38	**N/A**

Contents

List of Figures

List of Tables

1 Summary

Our team has, in the course of working on the DARPA Modeling Adversarial Activity (MAA) program, further developed a fast, flexible suite of graph matching tools designed to robustly align large networks in the presence of noise, paying special heed to developing methods for multiplex matching; developed the theory and methodology behind Graph Matching Matched Filters which provide a principled, scalable method for discovering noisy subgraphs in a larger background graph; provided an open source `R` code-base, denoted `iGraphMatch`, for implementing our graph matching and graph matching matched filters methods and their competitors at scale; further developed the theory of vertex nomination, developing the analogues of the classical statistical concepts of consistency and Bayes optimality in the context of vertex nomination; developed a novel concept of adversarial contamination and data-adaptive regularization in the context of vertex nomination; developed a suite of flexible vertex nomination algorithms designed to be implemented on large, noisy networks; produced illustrative simulations and data analyses on MAA provided data and on externally provided real data sources.

The cooperative agreement period was 12/20/2017 - 1/31/2020. In addition to prime PI Lyzinski, and subcontract PIs Sussman, Park and Priebe, Dr. Jesus Arroyo (Postdoc at Johns Hopkins University), and Prof. Donniell Fishkind (Johns Hopkins University), Dr. Keith Levin (Postdoc at University of Michigan) worked on key aspects of theory and methods development and helped implement our algorithms in `R`. Graduate students involved include Konstantinos Pantazis (UMass→University of Maryland; funded by MAA directly), Heather Patsolic (JHU, PhD dissertation defended March 2020), Joshua Agterberg (JHU), Ao Sun (JHU), Lingyao Meng (JHU), Ji Ah Lee (UMass; funded by MAA directly), Al Fahad Al-Qadhi (UMD), Ayushi Saxena (UMD), Fei Feng (Boston University), Zihuan Qiao (Boston University), Wang (Esther) Qian (Boston University), Christy Lin (Boston University), and Benjamin Draves (Boston University).

2 Introduction

From social networks (where, for example, graphs represent the social interactions between users) to neuroscience brain networks (where graphs represent interactions between brain regions), graphs are an increasingly popular data modality across the social and lab sciences for modeling the complex dependency relationships between data objects. Similar to other big data scenarios, very large networks often span multiple modalities [1]; e.g., in social networks users can interact across different platforms (Twitter, Facebook, Instagram, etc.). Moreover, these networks are often collected only piecewise; e.g., a social network collected over a particular geographic region [2] rather than over the entire user base. Global graph analytics often require matching (i.e., identifying vertices representing the same data object) and merging graphs across modalities, and/or they require combining the piecewise collected networks into a coherent world graph. Template (or subgraph) detection presents an important example of such a global analytic and is an important inference task for detecting adversarial activities occurring within the network data. Moreover, the adversarial subgraph signal is often split across the data modalities or the piecewise-collected networks, and working in any one single modality or without the fully constructed world graph will result in a poor template recovery performance [3]. Successful template detection then requires a coherent pipeline for combining the information contained in the network modalities and piecewise

collected network samples before applying analytics for template detection.

In order to match and merge networks in Technical Area 2 (TA2) for MAA, we further developed our flexible, scalable graph matching approach paying special heed to overcoming the real data pathologies inherent to the MAA provided data. Significant advancements made during MAA include the following: modifying our core graph matching approach to incorporate prior information in the form of hard [4] and soft [5] seeded vertices; developing novel data-adaptive regularization techniques to match across networks with incommensurate topologies [6]; developing theory and methods for a novel approach for matching unipartite and bipartite networks [7]; developing novel metrics and statistics for assessing the feasibility and difficulty of matching network pairs [8, 9]; developing a novel theoretical framework through which graph matching and graph matchability can be understood via Maximum Likelihood Estimation, a key statistical result that broadens the impact of our graph matching work to new statistically-focused communities; and developing entirely novel, state-of-the-art graph matching approaches based on multiscale diffusion maps [10]. These efforts manifested in illuminating experiments and data analyses on MAA provided TA2 data, in which we were able to contribute fundamental understandings to the program about when network topology alone was sufficient to align vertices across channel graphs, and what underlying characteristics led vertices to be identifiable across channels. Some highlights of our MAA efforts are contained in Section 3.1.5.

Another important TA2 inference task is vertex nomination (VN), in which vertices of interest in one network are used to query a second network in order to discover latent vertices of interest in the second network. In [11], we developed the key statistical concepts of consistency and Bayes optimality in the context of VN. The work culminated in proving that universally consistent (i.e., vertex nomination methods consistent against all underlying graph distributions) do not exist. Beyond the immediate implications on applying VN in streaming or evolving network environments, this work led to the novel understanding of adversarial contamination in vertex nomination we developed in [12]. In [12], we developed an effective spectral-embedding based vertex nomination scheme; a simple model for adversarial network contamination that demonstrably negatively impacts the performance of our vertex nomination scheme; and network regularization methods to successfully mitigate the impact of the contamination. This work provides key practical insights when applying VN on noisy or contaminated networks, and lays the foundation for future work developing more robust VN methods. In [13], we developed a key MAA vertex nomination approach, dubbed VNviaSGM, in which seeded information is used to localize the search for vertices of interest across graphs before the localized networks are graph matched to provide the VN nomination lists. These approaches are flexible and designed to run at scale, and we successfully implemented them on MAA data, Microsoft Bing entity-transition graph data, Drosophila larva connectomes (nominating homologous neurons across hemispheres), cybersecurity data (Los Alamos National Labs data where we seek to pair IP addresses across computer network pairs; joint with colleagues at Naval Surface Warfare Center), and social network data (identifying users across networks), among other applications.

In order to uncover hidden templates in the large background graph in Technical Area 3 (TA3), we leveraged the insights gleaned from our TA2 efforts to develop the theory and methods underlying the Graph Matching Matched Filters approach [14]. Using the template as a matched filter, we efficiently query the background graph via graph matching to find analogous (to the template) structures in the background. Extending this methodology to MAA data necessitated lifting our graph matching and matched filters frameworks to the multiplex setting [3], where we could ef-

ficiently leverage the signal contained across multiple network channels. The resulting multiplex matched filter approach was our key TA3 analytic, and proved an invaluable tool for discovering latent, embedded templates in the presence of significant background noise. Some highlights of our MAA efforts are contained in Section 3.2.2.

In order to implement our MAA analytics, we created the open source `R` software package `iGraphMatch` (available for download at https://github.com/dpmcsuss/iGraphMatch). Paying special heed to scalability in limited resource environments, the code base allows for scalable implementations of our TA2 and TA3 analytics on large networks ($O(10^4)$ for TA2 and $O(10^5)$ for TA3). We also implemented our algorithms in `Python`, providing additional scalable alternative to our core `iGraphMatch` package.

3 Methods, Assumptions, and Procedures

Herein, we provide more detail for the methods and algorithms outlined in the Introduction. Note that in what follows, we assume graphs are undirected, and to each n-vertex graph G, we associate the adjacency matrix $A \in \mathbb{R}^{n\times n}$, satisfying

$$A[i,j] = \begin{cases} w & \text{if there is an edge with weight w between vertices i and j in } G \\ 0 & \text{if there is no edge between vertices i and j in } G. \end{cases}$$

Lifting our methodology to directed graphs is immediate, though is omitted for ease of exposition. Note that we will often use the graph G and its adjacency matrix A interchangeably in the sequel.

Notation: The following table collects some commonly used notation appearing in the sequel.

Table 1. Commonly used notation

Symbol	Description
$[n]$	For $n \in \mathbb{Z}_{>0}$, this denotes $\{1,2,3,\cdots,n\}$
$\binom{S}{2}$	For a set S, this represents the set $\{\{u,v\}$ s.t. $u,v \in S\}$
$\mathscr{G}_n$	For $n \in \mathbb{Z}_{>0}$, the set of labeled, undirected graphs on n vertices
$\mathscr{M}_n^c$	For $n,c \in \mathbb{Z}_{>0}$, the set of labeled, undirected c-channel multiplex graphs on n vertices
$V(g)$	For graph $g \in \mathscr{G}_n$, V_g denotes the set of vertices of g
$E(g)$	For graph $g \in \mathscr{G}_n$, E_g denotes the set of edges of g
$g[S]$	For a set $S \subset V(g)$, this denotes the induced subgraph of g on vertices in S
$\mathbf{0}_n$	The $n \times n$ matrix of all 0's
Π_n	For $n \in \mathbb{Z}_{>0}$, the set of $n \times n$ permutation matrices
$\mathscr{D}_n$	For $n \in \mathbb{Z}_{>0}$, the set of $n \times n$ doubly stochastic matrices

3.1 Graph Matching

Key contributions to the program: Using the `FAQ` algorithm of [15] as a starting point, we built a suite of flexible graph matching procedures designed to handle the data pathologies inherent to MAA (and other outside) data sets. We also furthered the theoretical understanding of graph matching writ large, publishing fundamental papers providing practically useful theoretical insights into graph matching and graph matchability.

Submitted/Published Graph Matching papers with MAA funding acknowledged include:

1. Arroyo, J., Priebe, C.E., and Lyzinski, V. "Graph matching between bipartite and unipartite networks: to collapse, or not to collapse, that is the question," *arXiv preprint arXiv:2002.01648*, 2020.

2. Arroyo, J. , Sussman, D. L. , Priebe, C. E. , and Lyzinski, V., "Maximum likelihood estimation and graph matching in errorfully observed networks," *arXiv preprint arXiv:1812.10519*, 2018.

3. Fang, F., Sussman, D. L. , and Lyzinski, V., "Tractable graph matching via soft seeding," *arXiv preprint arXiv:1807.09299*, 2018.

4. Fishkind, D.E., Adali, S., Patsolic, H. G., Meng, L., Lyzinski, V. and Priebe, C.E., "Seeded graph matching," *Pattern Recognition*, **87**, 2019, pp. 203 – 215.

5. Fishkind, D. E., Athreya, A., Meng, L., Lyzinski, V., and Priebe, C. E., "On a complete and sufficient statistic for the correlated Bernoulli random graph model," *arXiv preprint arXiv:2002.09976*, 2020.

6. Fishkind, D. E., Meng, L., Sun, A., Priebe, C. E., and Lyzinski, V., "Alignment strength and correlation for graphs," *Pattern Recognition Letters*, **125**, 2019, pp. 295–302.

7. Li, L. and Sussman, D. L., "Graph matching via multi-scale heat diffusion," In *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, California, 2019, pp. 1157–1162.

8. Lyzinski, V., and Sussman, D. L., "Matchability of heterogeneous networks pairs," *Information and Inference: A Journal of the IMA*, 2020, iaz031.

One of the core inference tasks we studied within this program is the problem of graph matching. Formally, given two graphs

$$A, B \in \mathscr{G}_n := \{n - \text{vertex labeled graphs}\},$$

the Graph Matching Problem, abbreviate GMP, seeks to solve (in its simplest form)

$$\min_{P \in \Pi_n} \|AP - PB\|_F^2 \text{ or equivalently } \max_{P \in \Pi_n} \text{trace}(APBP^T), \tag{1}$$

where,

$$\Pi_n = \{\text{matrices P in } \{0,1\}^{n \times n} \text{ satisfying } P\vec{1} = \vec{1}^T P = \vec{1}\}z$$

denotes the set of $n \times n$ permutation matrices. The graph matching problem has a rich history in the literature [16, 17] as a tool for de-anonymizing networks, identifying corresponding vertices across networks, and uncovering significant structure across networks.

The GMP is a notoriously difficult combinatorial optimization problem, with the more general formulation (where the graphs are loopy, weighted and directed) equivalent to the NP-hard Quadratic Assignment Problem (QAP) [18, 15]. Our core approach, `FAQ` [15], for approximately solving the GMP is to relax the combinatorial search space of permutation matrices Π_n to the doubly stochastic matrices

$$\mathscr{D}_n = \{\text{matrices D in } [0,1]^{n\times n} \text{ satisfying } P\vec{1} = \vec{1}^T P = \vec{1}\}$$

and then employ the Frank-Wolfe algorithm [19] to solve the relaxed problem

$$\max_{P\in\mathscr{D}_n} \text{trace}(APBP^T),$$

before finally projecting the final doubly stochastic solution to the permutations. Our efforts in this program focused on extending `FAQ` to handle situations when the graphs were of differing orders [14, 4]; when the graphs were multiplex (i.e., each graph is comprised of multiple different channels) [3]; when the graphs had a priori known seeded vertices [4]; and when the graphs were topologically incommensurate [6].

Moving beyond `FAQ`, we developed novel matching machinery for aligning networks at multiple scales [10]. This allows for the simultaneous emphasis of local structure (edge structure and neighborhoods) and longer-range structures such as small communities. The algorithm leverages tools from graph signal processing to transform the adjacency matrices into "heat graphs" that that can capture structures at multiple scales. By employing our multiplex graph matching framework, along with some modifications to improve the robustness of the procedure, we demonstrate that combining the heat graphs with the adjacency matrix can substantially improve performance, especially when few seeds are available.

Moreover, we wrote numerous papers furthering the theoretical understanding of the GMP, both from a methodological/algorithmic perspective [8, 6, 9] and from a statistical perspective [20]. Highlights include the first graph matching paper in the literature [8] to formally look at the dual contributions of inter and intra-graph correlation to both matching feasibility and algorithmic runtime, both of which play significant roles in MAA, and establishing a novel connection between the GMP and Maximum Likelihood Estimation in a very general family of random graphs [20].

Below we outline our contributions to each of these areas, culminating in our work on MAA provided data in Section 3.1.5.

3.1.1 Matching graphs of different orders

In [14, 4], we develop novel, principled heuristics for matching graphs of different orders (i.e., with differing sized vertex sets). This is an essential step for our MAA efforts; indeed, in TA2 the channels to be merged are often different orders, and in TA3 the Graph Matching Matched Filter approach relies on matching the smaller template to the larger background graph.

Our methodology for matching graphs of different orders relies on a combination of centering and padding the underlying networks. Consider $A \in \mathscr{G}_m$ and $B \in \mathscr{G}_n$ with $m \leq n$ wlog. At one extreme (naive padding), we append $n-m$ isolated vertices to A, corresponding to matching B

and $A \oplus 0_{n-m}$; where 0_{n-m} is the $n-m \times n-m$ matrix of all 0's. This effectively only rewards common edges across the matching and does not penalize/reward any other structure. As a result, this matches A to the best fitting subgraph of B. At the other extreme (centered padding), we first set the non-edges in the adjacency matrices of both graphs (before padding) to -1 (as opposed to 0 in the uncentered regime), and then append $n-m$ isolated vertices to A. This effectively matches $(2B-(\mathbb{1}\mathbb{1}^T-I_n))$ and $(2A-(\mathbb{1}\mathbb{1}^T-I_m)) \oplus 0_{n-m}$. This equally rewards common edges and common non-edges across the matching, and matches A to the best fitting **induced** subgraph of B. Choosing differing weights for the edges/non-edges (beyond ± 1) in A and B lets us run the gamut between fitting to an induced subgraph and a subgraph; and we can adaptively choose these weights to optimize performance and modeling considerations (see Section 3.1.4). For example, we can define the weighted adjacency matrices $\breve{A} \in \mathbb{R}^{n \times n}$ and $\breve{B} \in \mathbb{R}^{n \times n}$ via

$$\breve{A}(u,v) = \begin{cases} 1 & \text{if u,v} \in V(A), \text{ and } \{\text{u,v}\} \in E(A); \\ -1 & \text{if u,v} \in V(A), \text{ and } \{\text{u,v}\} \notin E(A); \\ 0 & \text{if u or v} \in [n] \setminus [m]; \end{cases} \tag{2}$$

$$\breve{B}(u,v) = \begin{cases} 1 & \text{if u,v} \in V(B), \text{ and } \{\text{u,v}\} \in E(B); \\ -w & \text{if u,v} \in V(B), \text{ and } \{\text{u,v}\} \notin E(B); \end{cases}$$

where we vary w from 0 to 1. Note that $w=0$ yields Naive Padding, and $w=1$ yields Centered Padding. Multiplex approaches as described in the next section allow for even more flexibility in these weighting schemes.

3.1.2 Multiplex FAQ

One of the main contributions of our efforts on this program was lifting the `FAQ` algorithm to the multiplex setting [3]. Before giving details, we will first define precisely one notion of multiplex networks we will employ.

Definition 1. *The c-tuple $\mathbf{G} = (G_1, G_2, \ldots, G_c)$ is an n-vertex multiplex network if for each $i = 1,2,\ldots,c$, we have that $G_i \in \mathscr{G}_{n_i} = \{n_i\text{-vertex labeled graphs}\}$, and the vertex sets $(V_i = V(G_i))_{i=1}^c$ further satisfy the following:*
i. For each $i \in [c]$, we have that $V(G_i) \subseteq [n]$;
ii. $\bigcap_{i=1}^{c} V(G_i) \neq \emptyset$ and $\bigcup_{i=1}^{c} V(G_i) = [n]$;
iii. The layers are a priori *node aligned; i.e., vertices sharing the same label across layers correspond to the same entity in the network.*

Note that each vertex $v \in [n]$ need not appear in each channel $i \in [c]$, however, we do require that at least one vertex appears simultaneously in all channels. We will denote the set of c-layer, n-vertex multiplex networks via $\mathscr{M}_n^c$.

To lift the monoplex GMP formulation to the multiplex setting, we consider $\mathbf{H} \in \mathscr{M}_m^c$ and $\mathbf{G} \in \mathscr{M}_n^c$ with $m \leq n$. We employ the centered padding scheme to each channel of each graph

yielding the weighted adjacency matrices $\widehat{A}_i \in \mathbb{R}^{m\times m}$ and $\widehat{B}_i \in \mathbb{R}^{n\times n}$ defined via

$$\widehat{A}_i(u,v) = \begin{cases} 1 & \text{if u,v} \in V(H_i), \{u,v\} \in E(H_i); \\ -1 & \text{if u,v} \in V(H_i), \{u,v\} \notin E(H_i); \\ 0 & \text{if u or v} \in [m] \setminus V(H_i); \end{cases} \tag{3}$$

$$\widehat{B}_i(u,v) = \begin{cases} 1 & \text{if u,v} \in V(G_i), \{u,v\} \in E(G_i); \\ -1 & \text{if u,v} \in V(G_i), \{u,v\} \notin E(G_i); \\ 0 & \text{if u or v} \in [n] \setminus V(G_i); \end{cases}$$

Denoting $\widehat{\mathbf{A}} = (\widehat{A}_1, \cdots, \widehat{A}_c)$ and $\widehat{\mathbf{B}} = (\widehat{B}_1, \cdots, \widehat{B}_c)$, the *Centered Multiplex Graph Matching Problem* (cMGMP) is then defined as finding an element $P \in \Pi_n$ in

$$\underset{P\in\Pi_n}{\operatorname{argmin}} \sum_{i=1}^{c} \lambda_i \|(\widehat{A}_i \oplus \mathbf{0}_{n-m})P - P\widehat{B}_i\|_F^2. \tag{4}$$

The λ_i control the signal strength in each channel; although in most experiments on real and synthetic MAA (and outside) data, we found $\lambda_i \equiv 1$ was sufficient.

We extend the `FAQ` algorithm to this multiplex setting as outlined in Algorithm 1.

Algorithm 1 Multiplex FAQ [3]

Input: Multiplex graphs $\mathbf{H} \in \mathscr{M}_m^c$ and $\mathbf{G} \in \mathscr{M}_n$; weights λ_i; tolerance $\varepsilon \in \mathbb{R} > 0$; initialization $P^{(0)}$

Pad $\mathbf{H}$ and $\mathbf{G}$ via the centered padding scheme yielding $\widehat{A}$ and $\widehat{B}$

while $\|P^{(t)} - P^{(t-1)}\|_F > \varepsilon$ **do**

i. $P^{(t)} \leftarrow P^{(t-1)}$

ii. $\nabla(P^{(t)}) \leftarrow \sum_{i=1}^{c} \lambda_i \left((\hat{A}_i \oplus \mathbf{0}_{n-m})^\top P^{(t)} \hat{B}_i + (\hat{A}_i \oplus \mathbf{0}_{n-m}) P^{(t)} \hat{B}_i^\top \right)$;

iii. $Q^{(t)} \leftarrow \max_{Q\in\mathscr{D}_n} \operatorname{trace}\left[\nabla(P^{(t)})^\top Q\right]$;

iv. $\alpha^* \leftarrow \max_{\beta\in[0,1]} \sum_{i=1}^{c} \lambda_i \operatorname{trace}((\hat{A}_i \oplus \mathbf{0}_{n-m}) Q_\alpha^{(t)} \hat{B}_i (Q_\alpha^{(t)})^\top)$, where $Q_\alpha^{(t)} = \alpha P^{(t)} + (1-\alpha)Q^{(t)}$;

v. $P^{(t)} \leftarrow \alpha^* P^{(t)} + (1-\alpha^*)Q^{(t)}$;

end while

$P^* \leftarrow \max_{P\in\Pi_n} \operatorname{trace}\left(P^\top P^{(\text{final})}\right)$;

Output: P^* matching multiplex graphs A and B;

Multiplex `FAQ` is one of our core analytics, and is a key piece to our Multiplex Graph Matching Matched Filters approach for template discovery [14]; this, in turn, formed the foundation for our TA3 efforts under the program (see Section 3.2 for detail).

3.1.3 Seeded Graph Matching

In [4, 5], we adapted the `FAQ` algorithm to incorporate a priori known information about the alignment across graphs in the form of *seeds*. Effectively, we consider two types of seeds,

i. **Hard seeds:** Hard seeds are a priori known exact 1–to–1 matches across graphs.

ii. **Soft seeds:** A soft seeded vertex v in A has an a priori known distribution over possible matches in B; practically, this translates to vertices in A with a list of candidate matches in B [5].

In the presence of s hard seeds, the Seeded Graph Matching (SGMP) problem aims to solve

$$\min_{P \in \Pi_{n-s}} \|A(I_s \oplus P) - (I_s \oplus P)B\|_F^2, \tag{5}$$

where, Π_{n-s} denotes the set of $(n-s) \times (n-s)$ permutation matrices, and $\oplus$ denotes the direct sum of matrices. Note that decomposing A and B via

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \text{ and } B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \tag{6}$$

where $A_{11}, B_{11} \in \mathbb{R}^{s \times s}$, $A_{12} B_{12} \in \mathbb{R}^{(n-s) \times s}$ $A_{21}, B_{21} \in \mathbb{R}^{s \times (n-s)}$, and $A_{22}, B_{22} \in \mathbb{R}^{(n-s) \times (n-s)}$, the SGMP is equivalent to

$$\max_{P \in \Pi_{n-s}} f(P) := \max_{P \in \Pi_{n-s}} \left[\text{trace}(P^T A_{21} (B_{21})^T) + \text{trace}(P^T A_{12}^T B_{12}) + \text{trace}(A_{22}^T P B_{22} P^T) \right].$$

In practice, we have found that incorporating even a few seeds via our `SGM` (seeded `FAQ`) algorithm leads to dramatic performance improvement in matching accuracy [4]. See Figure 1 for an example of this in the context of correlated Stochastic Blockmodel graphs [4]. In the figure, we see that dramatic performance improvements are possible (a higher match ratio means that the algorithm is recovering more of the latent correspondence across the graphs; the strength of this correspondence is captured by ρ) with even few seeds.

In contrast to the hard seed framework above, the information contained in soft seeds is used not to constrain the search space, but rather to initialize the `FAQ` algorithm with the optimization feasible region otherwise the unconstrained Π_n. Soft seeds can be thought of as providing a soft matching prior, and our GM procedure provides a posterior update of this prior. In the case where there is a true but unknown latent alignment between the networks, we show in [5] that if prior information provided by the soft seeding (our soft-seeded `FAQ` approach can handle very general prior structure here) is known about the true correspondence, this can be leveraged to initialize our `FAQ` algorithm near the truth. Moreover, under mild model conditions, the FAQ algorithm will, with high probability, converge in two steps to the true latent alignment. The theoretical insights offer practical guidance, as they confirm an oft observed phenomena: if `FAQ` will provide a good alignment, it will often do so in relatively few Frank-Wolfe steps. This leads to insights on early algorithm termination which are especially useful in scalable environments.

3.1.4 Matching Across Heterogeneous Topologies

In the MAA program, our task in TA2 was to match and merge multiple disparate network data sources into a single coherent background network. These individual sources often came in the form of differing network topologies across a pair of unipartite networks. This heterogeneity often

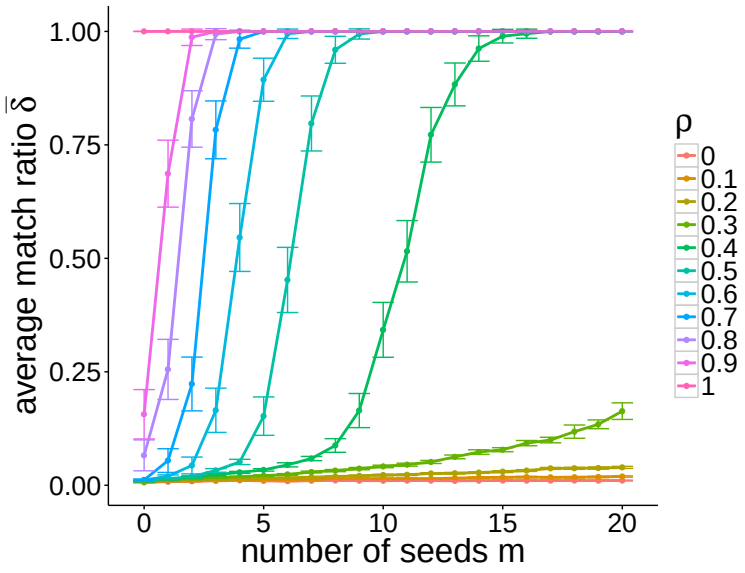


Figure 1. Effect of seeds on graph matching: Average match ratio $\overline{\delta} \pm 2$s.e. as a function of the number of seeds m, for different correlation values ρ (here ρ represents the edge-wise correlation across the pair of networks), when matching across a pair of correlated Stochastic Blockmodel graphs with $n = 300$ vertices.

has a deleterious effect on GM algorithmic performance, in that the true correspondence between the node-sets (denoted P_0) is often masked from optimization-based GM algorithms; i.e.,

$$P_0 \notin \text{argmax}_{P \in \Pi_n} \text{trace}(APBP^T).$$

In [6], we initiate the study of the impact of such heterogeneity on subsequent matching performance, and proposed spectral centering methods for ameliorating this effect. By preprocessing the networks via Universal Singular Value Thresholding (USVT) centering [21], we demonstrate, both in practical real data settings and in illustrative simulations, that we can overcome heterogeneous structure both within and across networks to recover graph matchability; i.e., we transform $A \mapsto \tilde{A}$ and $B \mapsto \tilde{B}$ so that P_0 will be a solution of the transformed GMP.

$$\{P_0\} = \text{argmax}_{P \in \Pi_n} \text{trace}(\tilde{A}P\tilde{B}P^T).$$

This *matchability* is essential, as when graphs are not matchable—i.e., the true node correspondence cannot be recovered in the face of noise—paired graph inference methodologies that utilize the across graph correspondence (for example, TA3 matched filters) cannot gainfully be employed. Non-matchability limits analysis to methods which rely on graph statistics which are invariant to relabeling of the vertices, which can be useful but lack the full power of methods which employ known or estimated vertex correspondences.

Another source of topological incommensurability across networks arises when one network is unipartite and one network is bipartite. This naturally occurs when, for example, we are matching a social network (unipartite) to transaction/purchasing networks (bipartite). Moreover, merging the channels across the multiple modalities as required in the MAA data requires such a matching (unipartite↔bipartite) As part of our MAA efforts [7], we developed a novel approach to matching the vertices across unipartite and bipartite networks. Often in the literature, when one network is

bipartite, it is collapsed or projected into a unipartite graph [22, 23], and graph matching proceeds as in the classical setting. This potentially leads to noisy edge estimates and loss of information. We formulate the graph matching problem between a bipartite and a unipartite graph using an undirected graphical model, and develop a method for graph matching based on alternating between a graph matching step via our `FAQ/SGM` algorithm and a graphical model estimation step via the Graphical Lasso. The model and optimization procedure were implemented in R using the `iGraphMatch` package (see Section 3.5). From a theoretical and practical (real-data) perspective, we explore both the situations in which collapsing is an admissible step in the alignment pipeline and the situations in which collapsing will lead to irrevocable performance loss. In both simulated and real data scenarios, our method performs well both in terms of vertex matching accuracy and graphical model estimation (and often better than either GM with collapsing or the Graphical Lasso alone).

3.1.5 GM in MAA for Task TA2

While we employed the above methods on numerous MAA provided data sets, the following two examples highlight some of the key contributions of our approach to the program .
Case Study 1: The above work was applied on MAA data towards answering the question of how gainfully we can match graphs leveraging only topological information. One of our key findings was that often, true alignment is not GM optimal (using classical GM objective function), but that by transforming the data suitably, we could lessen the impact of matching across different topologies and partially recover matchability across graphs. As an example, demonstrate this on the multiplex social network from [24]. The network contains 3 aligned channels representing user activity in FriendFeed, Twitter and Youtube (where the Youtube and Twitter channels were generated via FriendFeed which aggregates user information across these platforms). In total, there are 6,407 unique vertices across the three channels, with the channel specific networks satisfying:

channel	vertices	edges
FriendFeed	5,540	31,921
Twitter	5,702	42,327
YouTube	663	614

This data was provided to MAA by performers at PNNL, who also provided a template for TA3.

Using channels 1 and 2 as our test case, we considered matching the two channels with the `FAQ` algorithm initialized at the true alignment across channels, which we will refer to as P_T. If the algorithm terminates at a different alignment than P_T, this implies that P_T is not graph matching optimal. We consider mitigating potential heterogeneous degree structure across graphs by first centering the graphs via low-rank estimates of the graphs adjacency structure. To wit, if A (resp., B) is the adjacency matrix of channel 1 (resp., channel 2), we consider

$$\widetilde{A} = A - \widehat{\mathbb{E}(A)} \text{ for Channel 1}$$
$$\widetilde{B} = B - \widehat{\mathbb{E}(B)} \text{ for Channel 1;}$$

where $\widehat{\mathbb{E}(A)}$ is the best rank d_x estimate of $\mathbb{E}(A)$ and $\widehat{\mathbb{E}(B)}$ is the best rank d_y estimate of $\mathbb{E}(B)$. In Figure 2, we plot the number of vertices corrected recovered when matching $\widetilde{A}$ and $\widetilde{B}$ using `FAQ`

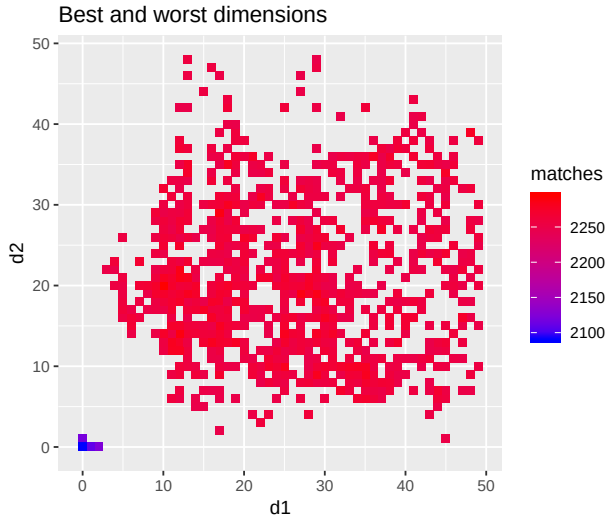


Figure 2. Effect of centering on graph matching in Friendfeed derived multilayer network: We plot the number of vertices corrected recovered when matching $\widetilde{A}$ and $\widetilde{B}$ using `FAQ` initialized at P_T over a range of d_x and d_y. Darker red colors correspond to more vertices matched correctly (correct according to P_T), while darker blue colors correspond to less vertices matched correctly (correct, again, according to P_T). Here, we only plot the best and worst pairs of (d_x, d_y) to visually aid the reader.

initialized at P_T over a range of d_x and d_y. Darker red colors correspond to more vertices matched correctly (correct according to P_T), while darker blue colors correspond to less vertices matched correctly (correct, again, according to P_T). Here, we only plot the best and worst pairs of (d_x, d_y) to visually aid the reader. We see that while centering can significantly increase the number of matchable vertices across channels, there is no centering regime under which P_T is GM optimal.

In Figure 3, we plot the relative improvement in GM objective function value for the solution obtained via running `FAQ` initialized at P_T versus the GM objective function evaluated at P_T. To interpret this plot, the solution P' corresponding to the point at roughly (2125, 0.250) matches $\approx$2125 vertices correctly (according to P_T) but satisfies

$$\frac{GM(P')}{GM(P_T)} \approx 1.250.$$

Color indicates the centering dimension used in channel 1 (L) and channel 2 (R). We see the general trend that the larger the objective function improvement, the fewer vertices correctly matched. We also see the general trend that centering with $d_x \approx d_y \approx 20$ achieves optimal recovery of P_T. We are presently exploring different mitigation strategies for recovering the latent alignment.

Case Study 2: In [8], we present a novel statistic, dubbed alignment strength, for estimating the dual effect of within-network heterogeneity and across-network correlation on both graph matchability and the algorithmic difficulty of matching the graph pairs. To demonstrate the utility of alignment strength, we consider the 4-channel VAST 2017 challenge dataset [25]

To investigate how effectively we could identify matched nodes across networks, we first performed a set of experiments designed to deduce the alignment strength between the $\binom{4}{2}$ network pairs. We proceed by following the guidance laid out in [8] and defining alignment strength be-

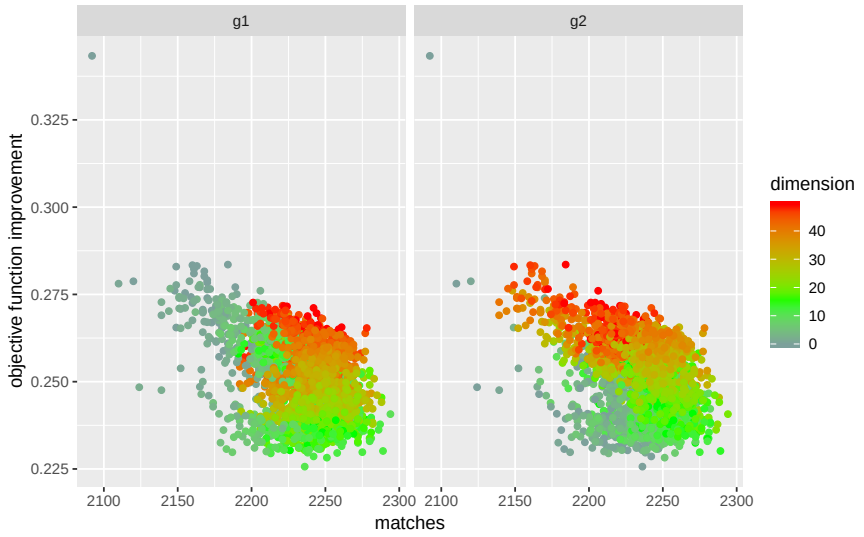


Figure 3. Effect on spectral centering dimension on GM in Friendfeed multilayer network: We plot the relative improvement in GM objective function value for the solution obtained via running `FAQ` initialized at P_T versus the GM objective function evaluated at P_T. Color indicates the centering dimension used in channel 1 (L) and channel 2 (R).

tween two graphs G and H (with true alignment P) via

$$\mathrm{str}(G,H,P) := 1 - \frac{\|G - PHP^T\|_1^2}{\frac{1}{n!}\sum_{Q\in\Pi(n)}\|G - QHQ^T\|_1^2},$$

where $\Pi(n)$ is the set of $n \times n$ permutation matrices. Theory and practice both have pointed to a larger value of $\mathrm{str}(G,H,P)$ indicating that the true alignment is easier to recover via our graph matching approach.

We computed the alignment strength between each of the $\binom{4}{2}$ channel pairs under each of three weightings: the unweighted case (where edge weights are ignored), the weighted case, and the Pass To Ranks (PTR) case (where we pass the weights to their relative ranks, a common robust analysis method). Results are summarized in Figure 4, column "corr." We see that the strength of the true alignment between the "calls" network and the "emails" network is significantly greater than 0 (i.e., our graph matching framework has hope of recovering this true alignment) and not for the other 5 network pairs (i.e., solutions to the traditional graph matching problem will most likely not recover this true alignment). Exploring this further, for each of the network pairs and each of the weightings considered, we compute the p-value for testing (below P is the true alignment and $\mathbf{Q}$ is a uniformly random alignment)

$$H_0 : \|G - \mathbf{Q}H\mathbf{Q}^T\|_1^2 = \|G - PHP^T\|_1^2 \text{ versus } H_1 : \|G - \mathbf{Q}H\mathbf{Q}^T\|_1^2 > \|G - PHP^T\|_1^2;$$

a p-value > 0.05 would indicate that not only is the alignment strength weak, the true alignment is essentially a chance alignment. The p-values are summarized in Figure 4, column "pval," with the estimated distributions of $\|G - \mathbf{Q}H\mathbf{Q}^T\|_1^2$ in relation to $\|G - PHP^T\|_1^2$ plotted in Figure 4. From the figure and the table, we see that the true alignment is essentially a chance alignment for each

```
# # A tibble: 18 x 4
# # Groups:   pair [?]
#    pair                    wtype        pval         corr
#    <chr>                   <fct>       <dbl>        <dbl>
#  1 calls vs emails         weighted    0         0.0427
#  2 calls vs emails         ptr         0         0.0342
#  3 calls vs emails         unweighted  0         0.0298
#  4 calls vs meetings       weighted    0.42      0.000000147
#  5 calls vs meetings       ptr         0.24      0.000000157
#  6 calls vs meetings       unweighted  0.41      0.000000158
#  7 calls vs purchases      weighted    0.73     -0.000000829
#  8 calls vs purchases      ptr         0.79     -0.000000569
#  9 calls vs purchases      unweighted  0.65     -0.000000375
# 10 emails vs meetings      weighted    0.79     -0.000000159
# 11 emails vs meetings      ptr         0.570    -0.000000125
# 12 emails vs meetings      unweighted  0.87     -0.000000245
# 13 emails vs purchases     weighted    1        -0.000000752
# 14 emails vs purchases     ptr         1        -0.000000645
# 15 emails vs purchases     unweighted  1        -0.00000118
# 16 meetings vs purchases   weighted    1        -0.000000135
# 17 meetings vs purchases   ptr         1        -0.000000500
# 18 meetings vs purchases   unweighted  1        -0.000000272
```

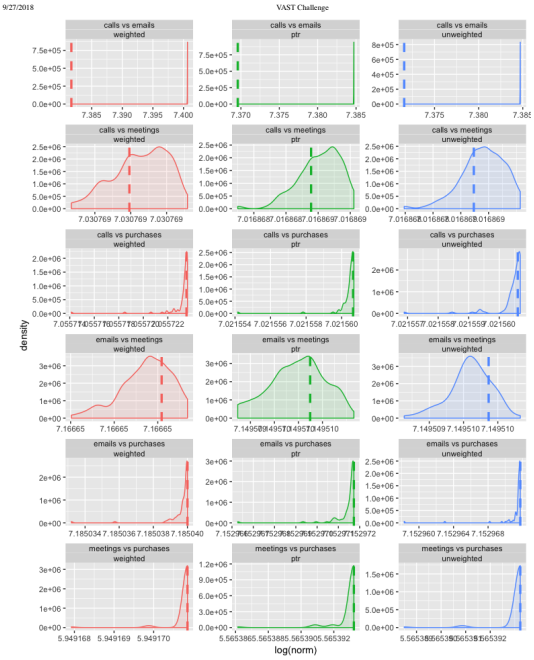


Figure 4. Alignment strength and matchability in VAST data example: Left: Table of summary statistics for measuring the alignment strength between the VAST network pairs. Right: Density plots for $\|G - \mathbf{Q}H\mathbf{Q}^T\|_1^2$, i.e., the pairwise 1-norms with randomly shuffled alignment. The dashed lines are the 1- norm with the true alignment P.

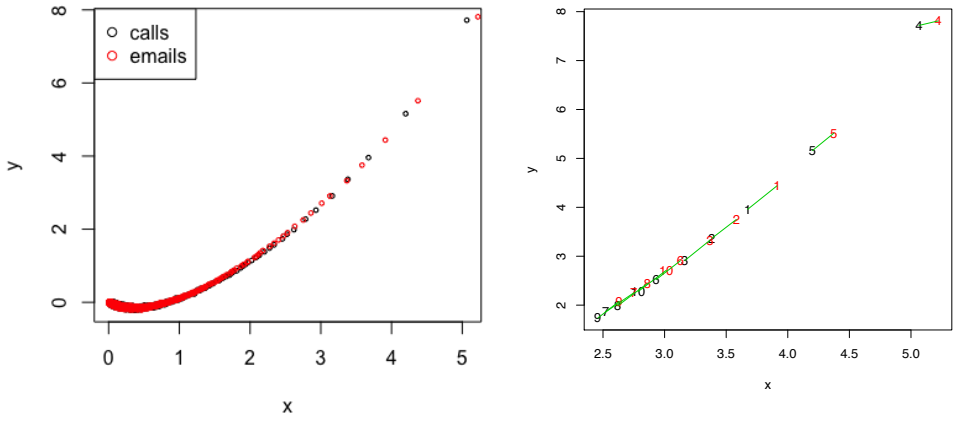


Figure 5. Spectral matching in VAST data example: Left: 2-d scatter plots of embeddings of the "calls" and "emails" graphs; Right: Zoom in of the tails of the 2-d scatter plots of embeddings of the "calls" and "emails" graphs. In both figures, matched points are connected via green dotted lines.

of the 6 pairs except the "calls-emails" pair. This is further strong indication that for these 5 pairs there is not significant signal in the local network topology for aligning the networks.

The relatively strong alignment strength between the "calls-emails" pair is born out in practice in Figure 5. Using Adjacency Spectral Embedding to embed the graphs into $\mathbb{R}^2$ (where the parameter 2 was selected via a principled model selection heuristic derived from [21, 26]), we further demonstrate strong statistical/structural similarity between matched nodes across the graphs.

3.2 Graph Matching Matched Filters

Key contributions to the program: We developed the novel Graph Matching Matched Filters (GMMF) approach to noisy template detection. We provided a flexible, scalable algorithmic suite for implementing GMMF in the multiplex network setting, and demonstrated the approaches excellent performance on both real and synthetic MAA (and outside) data sets. Moreover, we developed the underlying theory for this methodology, which offers key insights into the effectiveness (and drawbacks) of this approach.

Submitted/Published GMMF papers with MAA funding acknowledged include:

8 Pantazis, K., Sussman, D. L., Park, Y., Priebe, C. E., and Lyzinski, V., "Multiplex graph matching matched filters," *arXiv preprint arXiv:1908.02572*, 2019.

- Preliminary version accepted for publication at GTA3 3.0: The 3rd Workshop on Graph Techniques for Adversarial Activity Analytics, in conjunction with In Conjunction with the 2019 IEEE Big Data Conference, Los Angeles, CA.

9 Sussman, D. L., Park, Y., Priebe, C.E., and Lyzinski, V., "Matched filters for noisy induced subgraph detection," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

- Preliminary version accepted for publication at GTA3 2018: Workshop on Graph Techniques for Adversarial Activity Analytics, in conjunction with 11th ACM International Conference on Web Search and Data Mining, Marina Del Rey, CA (won the best paper award at the workshop).

The TA3 task in MAA is to find all instantiations of a provided template graph in a large, noisy background network. In the program, both template and background were multiplex networks, each composed of multiple communication and transaction channels representing disparate modes in the data collect process.

Our approach to template discovery, denoted the Graph Matching Matched Filters (GMMF) approach, uses the template as a matched filter with which we search the template, with the search mechanism provided by instantiations of the graph matching problem. The core approach, developed in the single channel setting in [14], proceeds as follows:

Input: Template $A \in \mathscr{G}_m$, background network $B \in \mathscr{G}_n$, padding regime, restarts N,

i. Pad A and B according to the naive or centered padding regime; denote the padded adjacency matrices via $\hat{A}, \hat{B} \in \mathbb{R}^{n \times n}$

ii. For each $i = 1, 2, \ldots, N$, generate an independent initialization $P_i^{(0)} \leftarrow \alpha J_n/n + (1-\alpha)P$ where $P \sim \text{Unif}(\Pi_n)$ and $\alpha \sim$Unif[0,1]; and set $P_i^* \leftarrow \texttt{FAQ}(\hat{A}, \hat{B}, P_i^{(0)})$;

iii. Rank the matchings $\{P_1^*, P_2^*, \cdots, P_N^*\}$ by increasing value of the graph matching objective function, $\text{trace}(\hat{A} P_i^* \hat{B}, (P_i^*)^T)$;

Output: Ranked list $(P_{(1)}^*, P_{(2)}^*, \cdots, P_{(N)}^*)$ of matchings, aligning template $A \in \mathscr{G}_m$ to background network $B \in \mathscr{G}_n$.

In the algorithm, multiple random restarts are utilized to combat the indefinite GM relaxation utilized in `FAQ` (i.e., $\min_{D \in \mathscr{D}_n} \text{trace}(ADBD^T)$). These restarts are trivially parallelizable, and our efficient implementation of GMMF in `iGraphMatch` can run each iteration for a $O(10^2)$ template matching to a $O(10^4)$ background in a few seconds on a standard laptop. Note that we were able to implement this on backgrounds of order $O(10^5)$, in line with our proposed computational milestone.

In [14], we also prove novel theoretical results which provide a deeper, and practically actionable, insight into the effectiveness of GMMF: in a very general network model where a noisy version of the template is embedded into the background, we found that

i. Under the naive padding scheme, there exists an adversarial background such that GMMF almost surely will not recover the noisy template;

ii. Under the centered padding scheme, oracle GMMF almost surely recovers the noisy template even in the presence of exponentially many background vertices;

iii. Under either padding regime, GMMF will almost surely fail to recover the noisy template if the template is of sub-logarithmic order compared to the background.

This theory is born out in numerous simulations and real data experiments, and we demonstrate that (leveraging few seeded vertices) GMMF can perfectly recover the noisy template instantiation.

3.2.1 Multiplex GMMF

In order to leverage the signal across the different channels in the TA1 provided template networks, we lift our graph matching matched filter (GMMF) methodology to the multiplex setting in [3]. The multiplex GMMF (MGMMF) utilizes our existing FAQ/SGM graph matching methodology to find instantiations of a noisy induced subgraph template in a background multiplex network: i.e., given a multiplex template $\mathbf{H} \in \mathscr{M}_m^c$, MGMMF aims to find the best fitting subgraph(s) in a larger multiplex background network $\mathbf{G} \in \mathscr{M}_n^c$. The details of MGMMF are provided in Algorithm 2.

Moreover, as in [14], in [3] we establish novel theory underpinning the effectiveness of the MGMMF approach, paying special heed to the benefit of considering multiple channels. Our findings indicate that under general random graph models:

i. Multiple channels can amplify the weak signal present in each individual channel, and template recovery is possible in the presence of multiple weak-signal channels. As an example, we consider $n = m = 100$, and we let $\mathbf{G}, \mathbf{H} \in \mathscr{M}_{100}^c$ (for c ranging over $\{1, 2, \cdots, 10\}$). For $i \in [c]$, we have that $(G_i, H_i) \sim \text{ER}(100, 0.5, \rho)$ (i.e., each graph is marginally Erdős-Rényi(100,0.5) and edges across graphs are collectively independent except that for each $\{i, j\} \in \binom{V}{2}$ the random edge between i, j in G_1 has correlation ρ with random edge between i, j in G_2). Utilizing $s = 10$ seeded vertices, we match $\mathbf{G}$ and $\mathbf{H}$ using `MFAQ` (Algorithm 1). Results are summarized in Figure 6.

Algorithm 2 M-GMMF

Input: Multiplex graphs $\mathbf{H} \in \mathscr{M}_m^c$ and $\mathbf{G} \in \mathscr{M}_n^c$ with $m < n$; padding regime; tolerance $\varepsilon \in \mathbb{R} > 0$; restarts N

1. Pad $\mathbf{H}$ and $\mathbf{G}$ accordingly; in the naive (resp., centered) padding regime, the padded $\mathbf{H}$ is denoted via $\tilde{\mathbf{A}}$ (resp., $\hat{\mathbf{A}}$), and the padded $\mathbf{G}$ via $\tilde{\mathbf{B}}$ (resp., $\hat{\mathbf{B}}$);
for $k = 1, 2, \cdots, N$, **do**
2. $P^{(0)} \leftarrow \alpha J_n/n + (1-\alpha)P$ where $P \sim \text{Unif}(\Pi_n)$ and $\alpha \sim$Unif[0,1];
3. In the naive (resp., centered) padding regime, $P_k^* \leftarrow \texttt{MFAQ}(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, P^{(0)}, \varepsilon))$ (resp., $P_k^* \leftarrow \texttt{MFAQ}(\hat{\mathbf{A}}, \hat{\mathbf{B}}, P^{(0)}, \varepsilon))$;
end for
4. Rank the matchings $\{P_1^*, P_2^*, \cdots, P_N^*\}$ by increasing value of the multiplex graph matching objective function;
Output: Ranked list $(P_{(1)}^*, P_{(2)}^*, \cdots, P_{(N)}^*)$ of matchings, aligning multiplex template $\mathbf{H}$ to background $\mathbf{G}$.

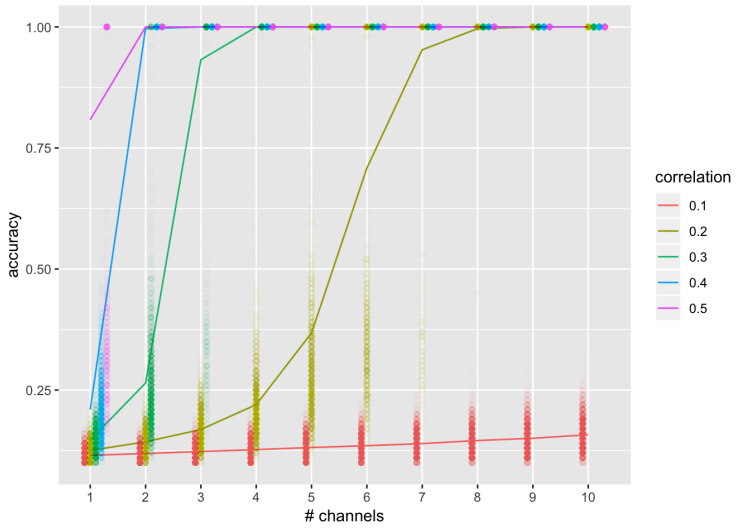


Figure 6. Correlation and matchability in multiplex graph matching: Considering $n = m = 100$, we let $\mathbf{G}, \mathbf{H} \in \mathscr{M}_{100}^c$ (for c ranging over $\{1, 2, \cdots, 10\}$). For $i \in [c]$, we have that $(G_i, H_i) \sim \text{ER}(100, 0.5, \rho)$. Utilizing $s = 10$ seeded vertices, we match $\mathbf{G}$ and $\mathbf{H}$ using `MFAQ` (Algorithm 1). In red (resp., olive, green, blue, purple) we plot the results for $\rho = 0.1$ (resp., $\rho = 0.2$, $\rho = 0.3$, $\rho = 0.4$, $\rho = 0.5$). The partially transparent points visualize the accuracy distribution and correspond to individual Monte Carlo replicates.

ii. Given enough channels with strong signal (i.e., positive correlation), the template and background remain matchable even in the presence of (potentially) multiple anti-correlated (i.e., noisy or adversarial) channels; See Figure 7.

3.2.2 MGMMF in MAA

Our MGMMF algorithm provides the backbone of our approach to the TA3 template finding problem, and we successfully employed MGMMF on numerous MAA-provided synthetic and real

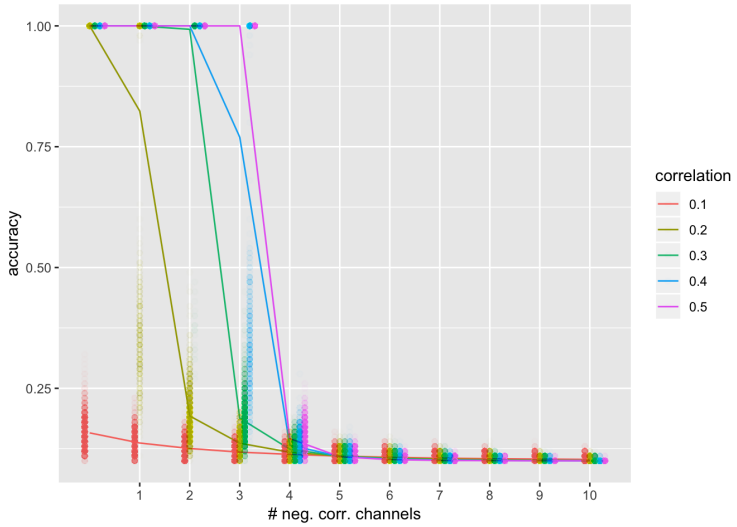


Figure 7. Effect of noisy channels on matchability in multiplex graph matching: We consider $n = m = 100$, and we let $\mathbf{G}, \mathbf{H} \in \mathscr{M}_{100}^{10}$ (i.e., $c = 10$), where for $i \in [10]$ we have that $(G_i, H_i) \sim \mathrm{ER}(100, 0.5, \rho)$. We consider ρ to take two possible values: $\rho = r$ for c_g channels, or $\rho = -r$ for $c_b = c - c_g$ channels, where r varies in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. We plot the matching accuracy (averaged over 2000 Monte Carlo replicates) obtained by `M-FAQ` (with 10 seeds) versus c_b. In red (resp., olive, green, blue, purple) we plot the results for $r = 0.1$ (resp., $r = 0.2$, $r = 0.3$, $r = 0.4$, $r = 0.5$). The partially transparent points visualize the accuracy distribution and correspond to individual Monte Carlo replicates.

Table 2. We provide the % of template edges present in the recovered background signal in the best random restart. For example, the best recovered background signal recovered 86.67% of the edges in template channel 1, and 85.07% of the edges in template channel 2, and 96.77% of the edges in template channel 3. Here, the best performer is the one that recovers the highest average % across the three channels (averaging the % within each channel across channels).

	% recovered in ch. 1	% rec'd in ch. 2	% rec'd in ch. 3
Centered	86.67	85.07	96.77
Naive	98.33	100	96.77

datasets. Herein, we highlight MAA applications to further underscore the good performance obtained by our approach.

Case Study 1: We consider the performance of our multiplex matched filter approach in detecting a hidden template in a multilayer social media network from [24] (considered for TA2 in Section 3.1.5. The background network contains 3 aligned channels representing user activity in FriendFeed, Twitter and Youtube (where the Youtube and Twitter channels were generated via FriendFeed which aggregates user information across these platforms). Given a 35 vertex multiplex template $\mathbf{H}$ created by Pacific Northwest National Laboratories for the DARPA MAA program, we ran our `M-GMMF` algorithm (Algorithm 2) to attempt to recover the template in $\mathbf{G}$; results are summarized below.

In our first experiment, we first considered running "cold-start" `M-GMMF`; that is, no prior information (in the form of seeds, hard or soft) is utilized in the algorithm. We consider padding

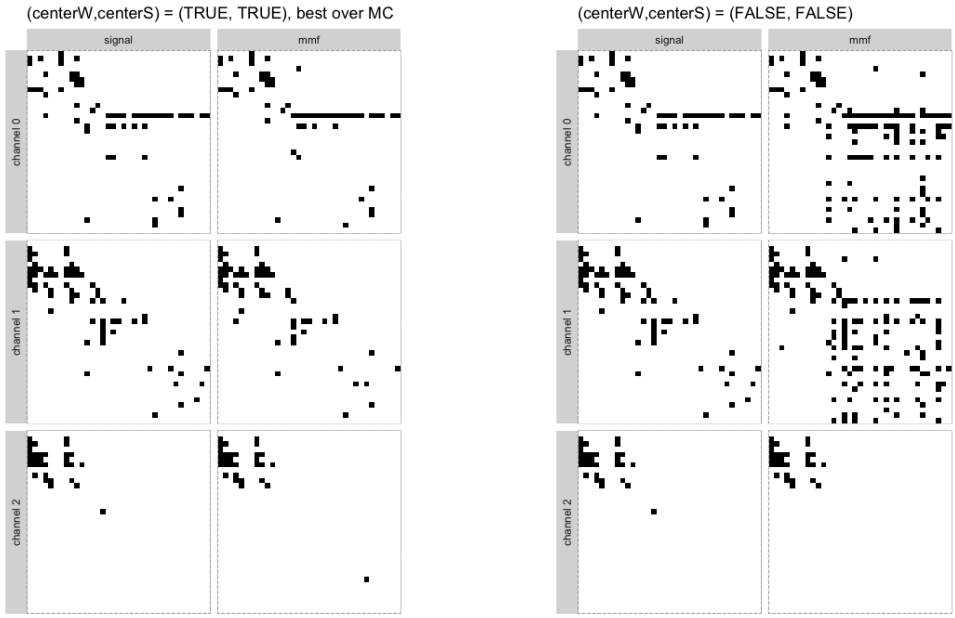


Figure 8. Multiplex matched filters in PNNL real-world data example: Signal recovered by the best performing random restart in M-GMMF using the Centered Padding regime (L) and Naive Padding scheme (R). As in Table 2, the best performer is the one that recovers the highest average % of the template edges across the three channels (averaging the % within each channel across channels). We plot the signal template across the three channels (in the left 3 panels) and the best recovered subgraphs in the background (in the right 3 panels).

the graph via the Centered and Naive Padding regimes and we ran M-GMMF with $N = 100$ random restarts. Numeric results are summarized in Table 2 (with the best recovered subgraph over the 100 restarts also plotted in Figure 8). Note here that the Centered Padding regime recovers most of the template edges (across the three channels) with minimal extra template edges in the recovered signal, while the Naive padding scheme recovers more template edges at the expense of extra template edges recovered.

The M-FAQ algorithmic primitive (Algorithm 1) used in our implementation of M-GMMF is most effective when it can leverage a priori available matching data in the form of *seeded* vertices. While hard seeds are costly and often unavailable in practice, there are many scalable procedures in the literature for automatically generating soft seed matches. Here, we use as a soft-seeding the output of [27], a filtering approach for finding all subgraphs of the background network homomorphic to the template. For each node in the template, the output of [27] produces a multi-set of candidate matches in the background, where each candidate match corresponds to a template copy contained in the background as a subgraph (not necessarily as an induced subgraph). We convert the candidate matches into probabilities by simply converting the multi-set to a count vector and normalizing the count vector to sum to 1. We then consider the normalized count vectors as rows of a stochastic matrix; this stochastic matrix provides M-FAQ with a soft-seeding which can be used to initialize the algorithm. Considering random restarts as perturbations (akin to Step 2 of Algorithm 2) of the soft-seeding (conditioned on retaining nonnegative entries), we ran M-GMMF using a generalization of the Centered Padding regime as in Eq. 2. Optimal performance in the

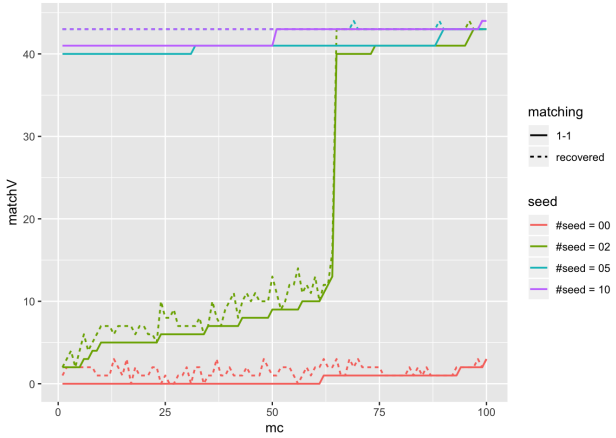


Figure 9. Seeding in M-GMMF in PNNL real-world data example (accuracy): Performance of M-GMMF on the PNNL V5 10K B0 (version 2) data set for the 100 random restarts. Note that the restarts have been ordered based on increasing value of matched template vertices

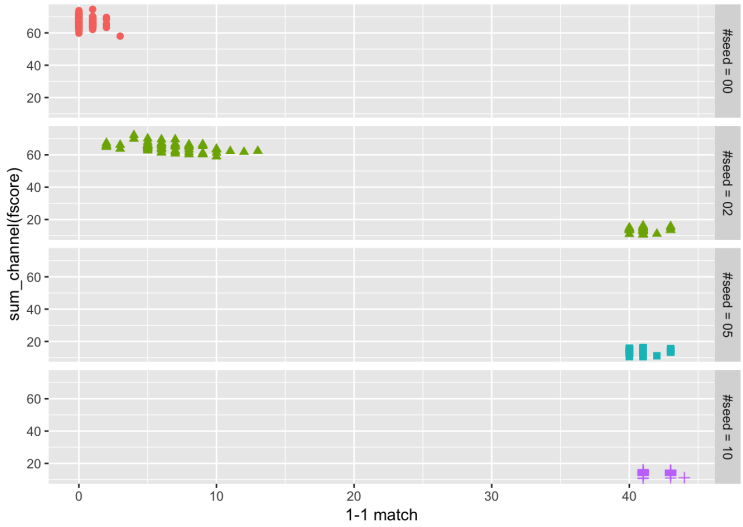


Figure 10. Seeding in M-GMMF in PNNL real-world data example (objective function): Objective function value obtained by M-GMMF on the PNNL V5 10K B0 (version 2) data set for the 100 random restarts. This data consists of 6 channels with $O(10K)$ vertices in the multiplex background graph. Note that the restarts have been ordered based on decreasing value of the obtained objective function.

present experiment was achieved with $w = 0.25$, in which case $N = 4000$ random restarts yielded an induced subgraph in the background that was **isomorphic to the template network**.

Case Study 2: We applied our M-GMMF code to the PNNL V5 10K B0 (version 2) data set. Using only structural information (no timestamps or demographic features), our M-GMMF framework perfectly recovered the embedded template using only 2 seeds. In the 100 random restarts, we have the following breakdowns (see Figure 9 for details; note that the restarts have been ordered based on increasing value of matched template vertices):

i. 0 seeds: No restart recovers more than a few of the embedded template vertices;

ii. 2 seeds: $\approx 40\%$ of the restarts recover at least 40/44 embedded template vertices; the best performing restart recovers all 44/44 embedded template vertices;

iii. 5 seeds: The best performing restart recovers all 44/44 embedded template vertices; all restarts recover at least 40/44 embedded template vertices exactly;

iv. 10 seeds: The best restart recovers all 44/44 embedded template vertices; all restarts recover at least 41/44 embedded template vertices exactly.

Moreover (see Figure 10), there is a objective function gap between restarts that demonstrate good versus bad template recovery. Note that the embedded template is not isomorphic to the signal template graph in this example, hence our objective function is never equal to 0 across all channels. Nevertheless we are able to recover all of the template vertices exactly.

3.3 Vertex Nomination

Key contributions to the program: We developed the theoretical framework for understanding consistency in the Vertex Nomination (VN) framework, and we prove that universally consistent VN schemes do not exist. This lack of universal consistency is the motivating result for a new statistical understanding of adversarial attacks in the context of vertex nomination. We developed a framework for adversarial attacks in VN, and proposed data-adaptive regularization procedures for countering the adversary. This work led to the development of scalable, robust VN procedures that can be used (for example) in MAA to uncover latent vertex and structural alignments across networks.

Submitted/Published Vertex Nomination papers with MAA funding acknowledged include:

10. Agterberg, J., Park, Y., Larson, J., White, C., Priebe, C. E., and Lyzinski, V., "Vertex nomination, consistent estimation, and adversarial modification," *arXiv preprint arXiv:1905.01776*, 2019.

11. Lyzinski, V., Levin, K., and Priebe, C.E., "On consistent vertex nomination schemes," *Journal of Machine Learning Research,* **20**, 69, 2019, pp. 1–39.

12. Patsolic, H. G., Park, Y., Lyzinski, V., and Priebe, C. E., "Vertex nomination via seeded graph matching," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, to appear, 2020.

Vertex nomination (VN) [28, 29, 30, 31] comprises a suite of algorithms that aim to efficiently query (often large) network data sets given limited training. Similar in spirit to popular network-based information retrieval (IR) procedures such as `PageRank` [32] and personalized recommender systems on graphs [33], in vertex nomination the goal is as follows: Given vertices of interest in a graph G_1, rank the vertices in a second graph G_2 based on how likely they are judged to be interesting; with (ideally) interesting vertices in G_2 concentrating at the top of the rank list; see Figure 11 for a visual representation of this VN framework. As an inference task, this formulation of vertex nomination is distinguished from other supervised network mining tasks by the combination of the generality of what defines vertices as interesting [34, 11] and the (often) limited nature of the available training data (i.e., known vertices of interest) in G_1.

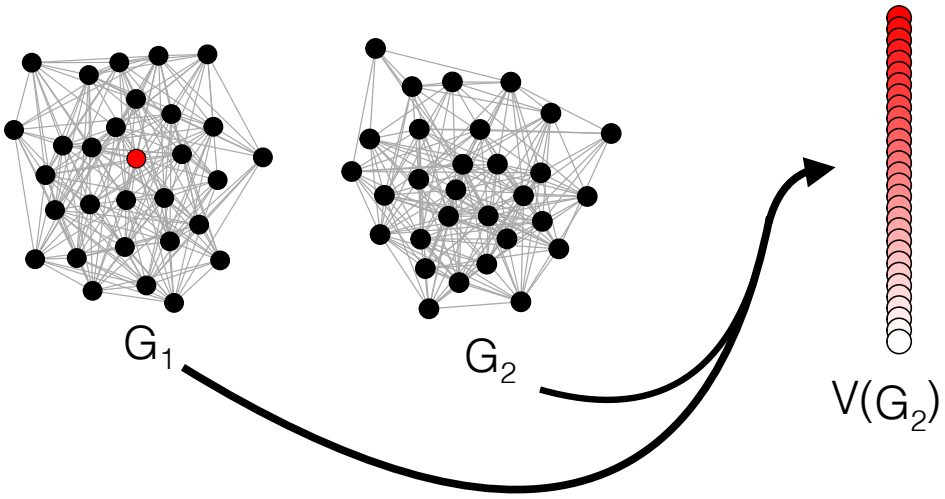


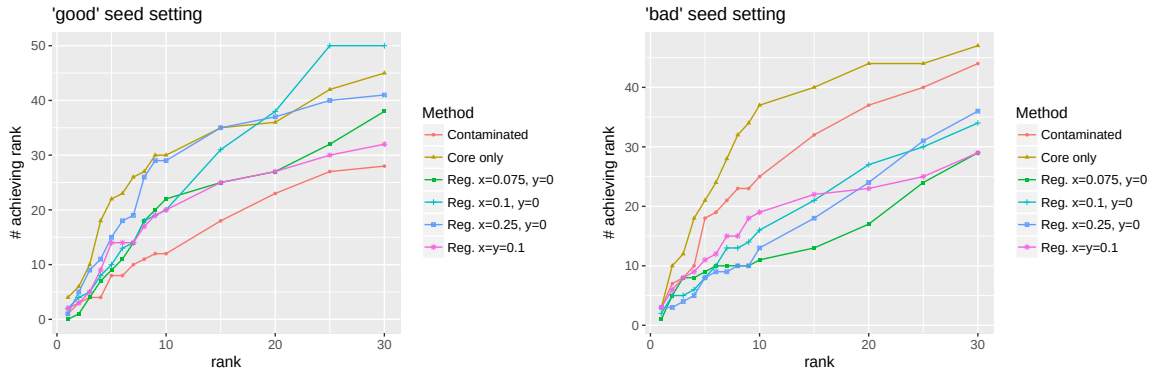
Figure 11. A visual representation of the Vertex Nomination framework: Given a vertex or vertices of interest v^* (colored red) in a graph $G_1 = (V_1, E_1)$, find the corresponding vertex/vertices of interest u^* (if it exists) in a second graph $G_2 = (V_2, E_2)$, ranking the vertices of G_2 into a nomination list so that u^* ideally appears at the top of the nomination list.

3.3.1 Consistency and Adversarial Noise in VN

Theoretical analysis of statistical learning methods, either supervised or unsupervised, often begins with the property of *consistency*: in the limit as the number of observations goes to infinity, does the learning procedure approach Bayes optimal performance. A procedure is dubbed *universally consistent* if it is asymptotically optimal regardless of the distribution of the observations. In the absence of universal consistency, a procedure designed for one class of distributions cannot be guaranteed to successfully adapt to a new setting or to be robust to adversarial data contamination. In the forty years since Stone's Theorem [35], the field has deduced universal consistency of a variety classification rules, providing the foundation for generally adaptive learning machines for the classification task.

In [11], we extend the concepts of consistency and Bayes optimality to the vertex nomination task. Moreover, in [11], we prove that, unlike in the classification setting, no universally consistent vertex nomination scheme exists. This lack of universal consistency is the motivating result for a new statistical understanding of adversarial attacks in the context of vertex nomination. A simple consequence of the lack of universal consistency is that for any vertex nomination rule, there are network sequences for which the rule is not consistent. An adversary can then be understood as a probabilistic mechanism designed to transform network sequences for which the rule is consistent into network sequences for which the rule is not consistent [12].

In [12], we consider a simple spectral-graph-embedding-based vertex nomination scheme and demonstrate the following in both synthetic and real data applications: A vertex nomination scheme that works effectively; a simple model for adversarial network contamination that demonstrably negatively impacts the performance of our vertex nomination scheme; and network regularization successfully mitigating the impact of the contamination (see Figure 12, panel a, for an example of this phenomena). This work provides crucial first steps towards an understanding of the complex interplay between universal consistency, adversarial network attack models, and adaptive regular-



a) # achieving rank $\leq x$ versus x b) # achieving rank $\leq x$ versus x

Figure 12. Adversarial contamination and regularization in vertex nomination: (Adapted from [12]) We plot the performance of VN $\circ$ GMM $\circ$ ASE for a "good" seed set (in a) and a "bad" seed set (in b) of size $s = 10$ in nominating across the High School friendship networks from [37]. The x-axis shows the ranks in the nomination list and the y-axis shows how many vertices $v \in G_1$, when viewed as the vertex of interest (v.o.i.), had their corresponding vertex of interest ranked in the top x by VN $\circ$ GMM $\circ$ ASE. The gold line represents performance in the uncontaminated network pair; the red line for performance when G_2 is adversarially contaminated (noise vertices added to G_2); the green, teal, blue and pink lines for performance under various levels of regularization (anomalous vertex trimming) to counteract the adversary.

ization. Together with industry and government colleagues, this work on vertex nomination has seen a number of impactful applications in noisy (e.g., DTI brain network graphs [36]), evolving (e.g., Bing entity transition networks), and adversarial data domains [12].

3.3.2 VN via SGM

In [13], we introduce a core MAA VN procedure, denoted `VNmatch`. `VNmatch` is a graph matching-based VN heuristic in which seeded vertices in the proximity of the vertices of interest (v.o.i.) in G_1 are used to localize the two networks G_1 and G_2—effectively, the algorithm seeks to capture local neighborhoods of the vertices of interest in each graph—before using a soft matching version of `SGM` (see Algorithm 3) to identify corresponding v.o.i. across graphs. Formally, the algorithm proceeds as follows:

> **Input:** $A \in \mathscr{G}_n, B \in \mathscr{G}_m$; $S \subset V(A), S' \subset V(B)$ – the seed sets, with $S \leftrightarrow S'$; parameters ℓ and h satisfying $\ell \geq h$; vertex of interest x.

i. Compute $S_x = s \cap N_h(x)$, where $N_h(x)$ is the h-neighborhood of $x \in V(A)$, and denote the matching vertices in $V(B)$ via S'_x (so that $S_x \leftrightarrow S'_x$). If $|S_x| = 0$, terminate the algorithm.

ii. Compute $A_x = A[N_\ell(S_x)]$ and $B_x = B[N_\ell(S'_x)]$, where for a set $T \subset V(A)$, $A[T]$ is the induced subgraph of A with vertices in T.

iii. Use soft seeded graph matching (i.e., Algorithm 3) to match $A_x = (V_x, E_x)$ and $B_x = (V'_x, E'_x)$, yielding $p(\cdot,\cdot) : V_x \times V'_x \mapsto [0,1]$;

iv. Create nomination list ϕ_x for x by ranking the vertices in V'_x by decreasing value of $p(x,\cdot)$;

Algorithm 3 `SoftSGM`

Input: $G \in \mathscr{G}_n, G' \in \mathscr{G}_m$ with respective adjacency matrices A and B; number of seeds s (assumed to be first s vertices of G and G'); number of random restarts $R \in \mathbb{N}$; random initialization parameter $\gamma \in [0,1]$; stopping criterion ε;
Step 0: if $n \neq m$ set $A = (2A - (\mathbb{1}\mathbb{1}^T - I_n)) \oplus 0_{\min(0,m-n)}$ and $B = (2B - (\mathbb{1}\mathbb{1}^T - I_m)) \oplus 0_{\min(0,n-m)}$; Let $N = \max(n,m)$
for i=1:R **do**
Step 1: Generate Q_i Uniformly from the set of permutation matrices, Π_{N-s};
Step 2: Generate β_i Uniformly from $(0,\gamma)$ and set $P_i^{(0)} = \beta_i Q_i + (1-\beta_i)\frac{1}{N-s}(\mathbb{1}\mathbb{1}^T)$;
Step 3:
while $\|f(P^{(j)}) - f(P^{(j-1)})\|_F > \varepsilon$ **do**
Step a: Compute $\nabla f(P^{(j)}) = A_{21}B_{21}^T + A_{12}^T B_{12} + A_{22}P^{(j)}B_{22}^T + A_{22}^T P^{(j)} B_{22}$;
Step b: Compute $Q^{(j)} \in \arg\max\{\text{trace}(Q^T \nabla f(P^{(j)}))\}$ over $Q \in \mathscr{D}_{N-s}$ via the Hungarian Algorithm [38, 39];
Step c: Compute $\alpha^{(j)} = \arg\max\{f(\alpha P^{(j)} + (1-\alpha)Q^{(j)})\}$ over $\alpha \in [0,1]$;
Step d: Set $P^{(j+1)} = \alpha^{(j)}P^{(j)} + (1-\alpha^{(j)})Q^{(j)})\}$;
end while
Step 5: Compute $P_i \in \arg\max\{\text{trace}(Q^T P^{(\text{final})})\}$ over $Q \in \Pi_{N-s}$ via the Hungarian Algorithm, where $P^{(\text{final})}$ is output from the while loop;
end for
Step 6: Define p via $p(\ell,k) = \left[\sum_{i=1}^R \frac{1}{R} P_i\right]_{\ell,k}$;
Output: p

Output: ϕ_x

When applying this algorithm for multiple v.o.i., individual lists can be made for each v.o.i. and then combined via max ranking or average ranking heuristics.

The localization aspect of VNmatch is essential for scaling this algorithm to very large data sets. Combined with automated seed detection procedures (for example, seeding by mining available vertex attributes or features) this algorithm provides a flexible method for querying large graphs at scale. As such, we have seen successful implementation of this approach on Drosophila larva connectomes (nominating homologous neurons across hemispheres), cybersecurity data (LANL data where we seek to pair IP addresses across computer network pairs; joint with colleagues at NSWC), and social network data (identifying users across networks), among other applications.

3.4 Joint Graph Embeddings

Key contributions to the program: We developed theoretical and practical methods for embedding multiple networks simultaneously into a common Euclidean space. This enables the machinery of time-series analysis and classical multiscale statistical inference methods to be directly employed on the embedded networks.

Submitted/Published Joint Graph Embedding papers with MAA funding acknowledged include:

13. Levin, K., Athreya, A., Tang, M., Lyzinski, V., Park, Y., and Priebe, C. E., "A central limit theorem for an omnibus embedding of random dot product graphs and implications for multiscale network inference," *arXiv preprint arXiv:1705.09355*, 2019.

14. Wang, S., Arroyo, J., Vogelstein, J. T., and Priebe, C. E., "Joint embedding of graphs," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

In [40], we develop fundamental statistical tools for multi-graph inference. Building on our previous work in multiple network inference, this paper develops a multiple graph (n$\geq$2) embedding and hypothesis testing paradigm that allows for principled hypothesis testing for network differences to occur at both the network and vertex level. These tools are imminently MAA relevant, as they allow us to determine if graph distributions change across network pairs, and moreover which vertices across graphs are responsible for this change. The methodology has had successful application of these methods on complex real data sets provided by neuroscience colleague and colleagues at Microsoft Research.

In [41], we develop a novel optimization-based approach for jointly spectrally embedding networks. Given a set of graphs, the methods identifies linear subspaces spanned by rank one symmetric matrices and projects the adjacency matrices of the graphs into this subspace. As a bonus, the projection coefficients can be treated as features of the graphs. Through illustrative theory and methods development and experiments, the method is shown to produce parameter estimates (in a novel joint graph model) with small errors. Moreover, the extracted features enable state-of-the-art performance in classifying graphs across real (human brain graphs) and synthetic data examples.

3.5 iGraphMatch

The iGraphMatch R package serves as a practical complement to our theoretical developments for the graph matching problem. In particular, it aims at providing a centralized repository for state-of-art graph matching methodologies, as well as useful auxiliary tools and such as metric computation and random correlated graph pairs samplers. Although there exist some open source software and packages containing graph matching functionality, iGraphMatch package provides versatile options of working with both igraph objects and graphs in matrix form. Most algorithms have been adapted for matching graphs under a generalized setting: weighted directed graphs of different order, matching multiple graphs, and incorporating different forms of prior information including partial matches and similarity between nodes.

In general, the GM algorithms implemented in the iGraphMatch package can be divided into three groups. The first group of algorithms is composed of methods requires some relaxations to the objective function, such as FW (Frank-Wolfe), CONVEX, and PATH. These algorithms employ a gradient-descent-based methodology to optimize the relaxed objective functions over a continuous domain. The second group of algorithms consists of algorithms applying the idea of percolation to pairs of nodes. The guiding intuition is that a larger number of matched neighbors is an indicator of a more plausible match and hence the currently matched nodes spread information to neighboring nodes. The last group of graph matching algorithms use spectral properties of adjacency matrices to convert the GM problem into an eigenvector problem.

Functions for conducting graph matching have the same basic syntax:

```
graph_match_*(A, B, seeds, ***algorithm parameters***)
```

The first two arguments represent the adjacency matrices of two networks which are flexible in taking matrices or igraph objects or a list of graphs to match multi-layer graphs. The `seeds` argument contains prior information a partial correspondence if any is known. Other arguments vary with regard to different graph matching algorithms. Each graph matching function will output a list of graph matching results, including a data frame of the correspondence of matched pairs along with other auxiliary information.

Here we present a toy example showing the steps of conducting a graph matching analysis. We first sample a pair of correlated random graph from the correlated Erdős-Rényi graph model, then match two sampled graphs using the FW algorithm and finally evaluate the matching performance using the package.

First, install the iGraphMatch package from the GitHub repository:

```
R> devtools::install_github("dpmcsuss/iGraphMatch")
R> library(iGraphMatch)
```

Then sample a pair of correlated Erdős-Rényi graphs with 5 nodes, 0.5 edge probability, and Pearson correlation between the corresponding edges of two graphs of 0.5.

```
R> cgnp_pair <- sample_correlated_gnp_pair(n = 5, corr = 0.5, p = 0.5)
R> g1 <- cgnp_pair$graph1
R> g2 <- cgnp_pair$graph2
```

Note that in addition to `sample_correlated_gnp_pair` for sampling from the correlated Erdős-Rényi graphs, there are also functions for sampling from the stochastic block model, the random dot product model, and general independent edge models.

Suppose the first two pairs of nodes in the sampled graph be hard seeds, meaning they are the prior information on matches with certainty. In addition, we will include the pair of nodes (3,4) as soft seeds. This means there is some prior information suggesting (3,4) is a promising match but with uncertainty. We are going to fit soft seeds into the initialization of the non-seed part and let it evolve over iterations in the FW algorithm.

```
R> hard_seeds <- 1:5 <= 2
R> soft_seeds <- data.frame(seed_A = 3, seed_B = 4)
R> (start_bari <- init_start(start = "bari", nns = 3,
      ns = 2, soft_seeds = soft_seeds))
     [,1]   [,2]  [,3]
[1,]  0.0      1   0.0
[2,]  0.5      0   0.5
[3,]  0.5      0   0.5
```

Here we choose to initialize at the barycenter incorporating soft seeds, where the nodes with no prior information have equal chance of getting matched. Other options including starting at the random stochastic matrix and the result of convex relaxed Frank-Wolfe methodology. Then implement seeded graph matching using the FW methodology in `R` by using `graph_match_FW`.

```
R> (match_bari <- graph_match_FW(g1, g2, hard_seeds, start = start_bari))
```

```
$call
graph_match_FW(A = g1, B = g2, seeds = hard_seeds, start = start_bari)

$corr
 corr_A corr_B
1      1      1
2      2      2
3      3      4
4      4      3
5      5      5

$ns
[1] 2

$P
5 x 5 sparse Matrix of class "dgCMatrix"

[1,]  1 . . . .
[2,]  . 1 . . .
[3,]  . . . 1 .
[4,]  . . 1 . .
[5,]  . . . . 1

$D
5 x 5 sparse Matrix of class "dgCMatrix"

[1,]  1 . . . .
[2,]  . 1 . . .
[3,]  . . . 1 .
[4,]  . . 1 . .
[5,]  . . . . 1
```

After matching two graphs, the `match_report` function can be used to get a summary of the overall matching result. This includes commonly used measures including the number of matches, the number of correct matches, common edges, common non-edges, edge correctness and the objective function value.

```
R> match_report(match_bari, g1, g2, label = 1:5)

Call:
graph_match_FW(A = g1, B = g2, seeds = hard_seeds, start = start_bari)

# Matches: 3
# True Matches:  1
# Common Edges:  6
```

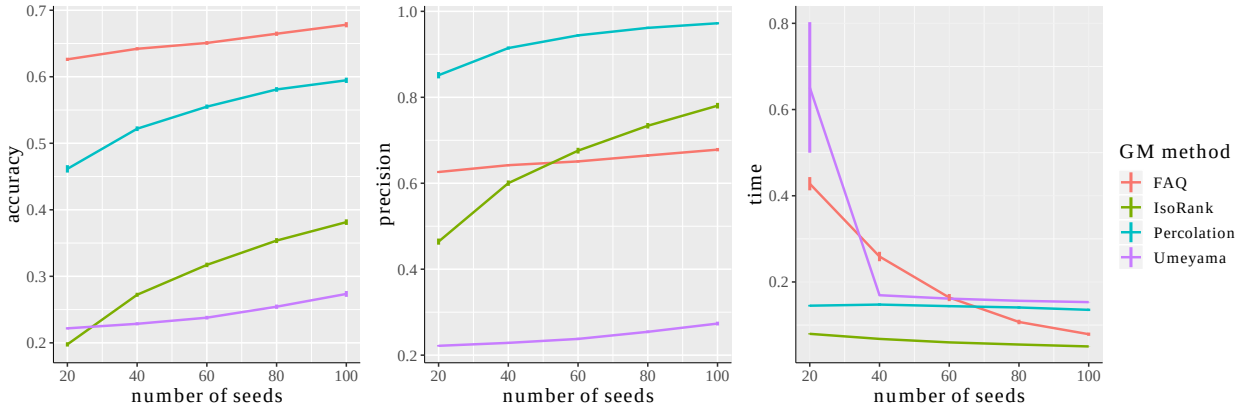


Figure 13. Comparison between different graph matching algorithms on the Enron data: We plot matching accuracy, precision and time of matching as a function of the number of seeds. There are 184 nodes in each graph and for each n_s we sampled $nmc = 500$ random seed sets.

```
# Common Non-edges:  15
Edge Correctness:  1
Objective Value:  2
```

The `best_matches` function has the functionality of finding the best matched vertices according to a given metric. As an example here, we apply the `best_matches` to rank the matches in the previous result using a row permutation statistic.

```
R> non_seeds <- !check_seeds(hard_seeds, n = 5, logical = TRUE)
R> best_matches(g1, g2, measure = "row_perm_stat", num = 3,
                  x = non_seeds, match_corr = match_bari$corr$corr_B)
```

```
  A_best B_best
1      3      4
2      5      5
3      4      3
```

To demonstrate the performance of different GM algorithms implemented in the package, we present the following experiment result on the Enron network dataset. The dataset consists of email messages between 184 employees of the Enron Corporation where each graph represents one week of emails and each edge indicates whether there is email sent from one employee to the other. The two networks are unweighted and directed with 488 and 482 edges in the two graphs, respectively.

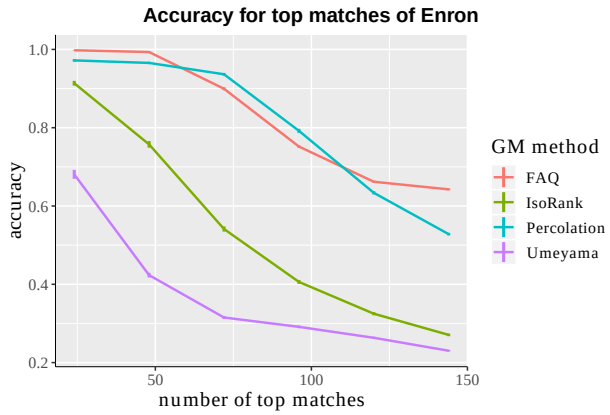


Figure 14. Graph matching accuracy for Enron with different number of top matches: Apply row permutation test to the matching result to get the accuracy with different number of top matches. Average of 500 Monte Carlo replicates with 40 seeds.

In Figure 13, the matching accuracy measures the absolute number of correct matches, the matching precision measures the fraction of correct matches in the matched set. In general, all the algorithms achieve better graph matching results within a smaller amount of time as the number of seeds increases. FW yields the largest quantity of correct matches, while PERCOLATION yields the highest quality matched set with significantly higher fraction of correct matches, and ISORANK is much faster than the other algorithms. On average, PERCOLATION appears to provide a superior balance between acquiring high quality matches and finding many matches while also being relatively fast.

Figure 14 shows the matching accuracy in the range of the top matches, where we vary the number of top matches between 24, 48, 72, 96, 120 and 144. With a bigger set of top matches, the matching accuracy associated with the matched set drops. Notice, that the top 50 matches using the FW algorithm recover the true correspondence between the subgraph almost perfectly, which also suggests a way of finding more reliable seeds.

4 Conclusions

Graph matching and graph template discovery are important emerging fields in the applied and theoretical statistics communities. Moreover, as graph-valued data becomes more common across the sciences, graph inference methods are becoming increasingly important to the broader scientific community. The DARPA MAA project enabled a deeper understanding of the possibilities and limitations of graph matching and template discovery in complex data environments, and the insights and methods developed therein should have a broad impact on myriad important academic/industrial/government programs that rely on these core analytics in their processing pipelines.

5 Bibliography

[1] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *Journal of complex networks*, vol. 2, no. 3, pp. 203–271, 2014.

[2] L. Li and W. M. Campbell, "Matching community structure across online social networks," *arXiv preprint arXiv:1608.01373*, 2016.

[3] K. Pantazis, D. L. Sussman, Y. Park, C. E. Priebe, and V. Lyzinski, "Multiplex graph matching matched filters," *arXiv preprint arXiv:1908.02572*, 2019.

[4] D. Fishkind, S. Adali, H. G. Patsolic, L. Meng, D. Singh, V. Lyzinski, and C. Priebe, "Seeded graph matching," *Pattern Recognition*, vol. 87, pp. 203 – 215, 2019.

[5] F. Fang, D. L. Sussman, and V. Lyzinski, "Tractable graph matching via soft seeding," *arXiv preprint arXiv:1807.09299*, 2018.

[6] V. Lyzinski and D. L. Sussman, "Matchability of heterogeneous networks pairs," *Information and Inference: A Journal of the IMA*, 01 2020. iaz031.

[7] J. Arroyo, C. E. Priebe, and V. Lyzinski, "Graph matching between bipartite and unipartite networks: to collapse, or not to collapse, that is the question," *arXiv preprint arXiv:2002.01648*, 2020.

[8] D. E. Fishkind, L. Meng, A. Sun, C. E. Priebe, and V. Lyzinski, "Alignment strength and correlation for graphs," *Pattern Recognition Letters*, vol. 125, pp. 295–302, 2019.

[9] D. E. Fishkind, A. Athreya, L. Meng, V. Lyzinski, and C. E. Priebe, "On a complete and sufficient statistic for the correlated bernoulli random graph model," *arXiv preprint arXiv:2002.09976*, 2020.

[10] L. Li and D. L. Sussman, "Graph matching via multi-scale heat diffusion," in *2019 IEEE International Conference on Big Data (Big Data)*, pp. 1157–1162, IEEE, 2019.

[11] V. Lyzinski, K. Levin, and C. E. Priebe, "On consistent vertex nomination schemes," *Journal of Machine Learning Research*, vol. 20, no. 69, pp. 1–39, 2019.

[12] J. Agterberg, Y. Park, J. Larson, C. White, C. E. Priebe, and V. Lyzinski, "Vertex nomination, consistent estimation, and adversarial modification," *arXiv preprint arXiv:1905.01776*, 2019.

[13] H. G. Patsolic, Y. Park, V. Lyzinski, and C. E. Priebe, "Vertex nomination via seeded graph matching," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. to appear, 2020.

[14] D. L. Sussman, Y. Park, C. E. Priebe, and V. Lyzinski, "Matched filters for noisy induced subgraph detection," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[15] J. T. Vogelstein, J. M. Conroy, V. Lyzinski, L. J. Podrazik, S. G. Kratzer, E. T. Harley, D. E. Fishkind, R. T. Vogelstein, and C. E. Priebe, "Fast Approximate Quadratic Programming for Graph Matching," *PLoS ONE*, vol. 10, no. 04, 2014.

[16] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 03, pp. 265–298, 2004.

[17] F. Emmert-Streib, M. Dehmer, and Y. Shi, "Fifty years of graph matching, network alignment and network comparison," *Information sciences*, vol. 346–347, pp. 180–197, 2016.

[18] E. L. Lawler, "The quadratic assignment problem," *Management science*, vol. 9, no. 4, pp. 586–599, 1963.

[19] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.

[20] J. Arroyo, D. L. Sussman, C. E. Priebe, and V. Lyzinski, "Maximum likelihood estimation and graph matching in errorfully observed networks," *arXiv preprint arXiv:1812.10519*, 2018.

[21] S. Chatterjee, "Matrix estimation by universal singular value thresholding," *The Annals of Statistics*, vol. 43, no. 1, pp. 177–214, 2014.

[22] K. A. Zweig and M. Kaufmann, "A systematic approach to the one-mode projection of bipartite graphs," *Social Network Analysis and Mining*, vol. 1, no. 3, pp. 187–218, 2011.

[23] T. Zhou, J. Ren, M. Medo, and Y. Zhang, "Bipartite network projection and personal recommendation," *Physical review E*, vol. 76, no. 4, p. 046115, 2007.

[24] M. Magnani and L. Rossi, "The ml-model for multi-layer social networks," in *2011 International Conference on Advances in Social Networks Analysis and Mining*, pp. 5–12, IEEE, 2011.

[25] M. A. Whiting, K. Cook, R. J. Crouser, J. Fallon, G. Grinstein, J. Haack, C. Henderson, K. Liggett, D. Staheli, J. Strasburg, *et al.*, "Vast challenge 2017: Mystery at the wildlife preserve," in *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 173–178, IEEE, 2017.

[26] M. Zhu and A. Ghodsi, "Automatic dimensionality selection from the scree plot via the use of profile likelihood," *Computational Statistics & Data Analysis*, vol. 51, no. 2, pp. 918–930, 2006.

[27] J. D. Moorman, Q. Chen, T. K. Tu, Z. M. Boyd, and A. L. Bertozzi, "Filtering methods for subgraph matching on multiplex networks," in *2018 IEEE International Conference on Big Data (Big Data)*, pp. 3980–3985, IEEE, 2018.

[28] D. Marchette, C. E. Priebe, and G. Coppersmith, "Vertex nomination via attributed random dot product graphs," in *Proceedings of the 57th ISI World Statistics Congress*, vol. 6, 2011.

[29] G. Coppersmith, "Vertex nomination," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 6, no. 2, pp. 144–153, 2014.

[30] S. Suwan, D. S. Lee, and C. E. Priebe, "Bayesian vertex nomination using content and context," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7, no. 6, pp. 400–416, 2015.

[31] D. E. Fishkind, V. Lyzinski, H. Pao, L. Chen, and C. E. Priebe, "Vertex nomination schemes for membership prediction," *The Annals of Applied Statistics*, vol. 9, no. 3, pp. 1510–1532, 2015.

[32] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web.," tech. rep., Stanford InfoLab, 1999.

[33] Z. Huang, W. Chung, and H. Chen, "A graph model for e-commerce recommender systems," *Journal of the American Society for information science and technology*, vol. 55, no. 3, pp. 259–274, 2004.

[34] P. Rastogi, V. Lyzinski, and B. Van Durme, "Vertex nomination on the cold start knowledge graph," *Human Language Technology Center of Excellence: Technical report*, 2017.

[35] C. Stone, "Consistent nonparametric regression," *Annals of Statistics*, vol. 5, pp. 595–645, 1977.

[36] J. Yoder, L. Chen, H. Pao, E. Bridgeford, K. Levin, D. E. Fishkind, C. E. Priebe, and V. Lyzinski, "Vertex nomination: The canonical sampling and the extended spectral nomination schemes," *Computational Statistics & Data Analysis*, vol. 145, p. 106916, 2020.

[37] R. Mastrandrea, J. Fournet, and A. Barrat, "Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys," *PloS one*, vol. 10, no. 9, p. e0136497, 2015.

[38] R. Jonker and T. Volgenant, "Improving the hungarian assignment algorithm," *Operations Research Letters*, vol. 5, no. 4, pp. 171–175, 1986.

[39] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistic Quarterly*, vol. 2, pp. 83–97, 1955.

[40] K. Levin, A. Athreya, M. Tang, V. Lyzinski, Y. Park, and C. E. Priebe, "A central limit theorem for an omnibus embedding of random dot product graphs and implications for multiscale network inference," *arXiv preprint arXiv:1705.09355*, 2019.

[41] S. Wang, J. Arroyo, J. T. Vogelstein, and C. E. Priebe, "Joint embedding of graphs," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

6 List of Acronyms

Acronym	Description
MAA	Modeling Adversarial Activity DARPA program
FAQ	The fast approximate quadratic assignment graph matching algorithm from [15]
GMP	Graph Matching Problem
SGMP	Seeded Graph Matching Problem
SGM	Seeded FAQ algorithm from [4]
VN	Vertex Nomination
GMMF	Graph Matching Matched Filters
M-GMMF	Multiplex Graph Matching Matched Filters