ABSTRACT

Title of Dissertation: MEASURING NETWORK DEPENDENCIES FROM NODE ACTIVATIONS
Rachael T.B. Sexton
Doctor of Philosophy,

Dissertation Directed by: Professor Mark D. Fuge
Department of Mechanical Engineering

My abstract for this dissertation.

MEASURING NETWORK DEPENDENCIES FROM NODE ACTIVATIONS

by

Rachael T.B. Sexton

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

Advisory Committee:

Professor Mark D. Fuge, Chair/Advisor

Professor Jordan L. Boyd-Graber

Professor Maria K. Cameron

Professor Michelle Girvan

Professor Vincent P. Lyzinski

Preface

Foreward

Acknowledgements

Table of Contents

Preface i

Foreward ii

Acknowledgements iii

Table of Contents iii

List of Tables vii

List of Figures viii

1 Introduction 1
- 1.1 Ambiguous Metrology . . . 1
- 1.2 Indirect Network Measurement . . . 3
- 1.3 Scope of this work . . . 7

I A Practitioner's Guide to Network Recovery 10

2 Metrology as matrices 11
- 2.1 Observation and feature "spaces" . . . 12
- 2.2 Models & linear operators . . . 14
- 2.3 Measurement quantification & error . . . 16
- 2.4 Proximity vs. Incidence . . . 16
  - 2.4.1 Kernels & distances . . . 16
  - 2.4.2 Incidence structures & dependency . . . 17
  - 2.4.3 Implications for networks . . . 17

3 Incidence in Vector Space 19
- 3.1 Dependence as Graph . . . 19
  - 3.1.1 Edges as Vectors of Nodes . . . 20
  - 3.1.2 Inner Product on Edges . . . 20
  - 3.1.3 From Edge Observations to Node Activations . . . 20
- 3.2 Occurence as Hypergraph . . . 21
  - 3.2.1 Hyperedges as Vectors of Nodes . . . 21

3.2.2 Inner product on Hyperedges . . . 21
3.3 Combining Occurence & Dependence . . . 21

4 Roads to Network Recovery **22**
4.1 Choosing a structure recovery method . . . 22
4.2 Organizing Recovery Methods . . . 22
4.2.1 Observing Nodes vs Edges . . . 24
4.2.2 Embeddings, Inner Products, & Preprocessing . . . 24
4.3 Tracing Information Loss Paths . . . 24
4.3.1 Table of Existing Approaches . . . 24
4.3.2 A Path Forward . . . 24

II Nonparametric Network Recovery With Random Spanning Forests **25**

5 Generative Random Spanning Forests **26**
5.1 Node Activation by Diffusive Processes . . . 27
5.1.1 Random Walk Activations . . . 27
5.1.2 Dependencies as Trees . . . 28
5.1.3 Matrix Tree and Forest Theorems . . . 28
5.2 Generative Model Specification . . . 29

6 Forest Pursuit: Approximate Recovery in Near-linear Time **30**
6.1 Sparse Dictionary Learning . . . 30
6.1.1 Problem Specification . . . 30
6.1.2 Matching Pursuit . . . 30
6.1.3 Space of Spanning Forests . . . 30
6.2 Forest Pursuit: Approximate Recovery in Near-linear Time . . . 30
6.2.1 Uncertainty Estimation . . . 31
6.2.2 Approximate Complexity . . . 31
6.3 Simulation Study . . . 31
6.3.1 Method . . . 31
6.3.2 Results - Scoring . . . 31
6.3.3 Results - Performance . . . 31
6.4 Discussion . . . 34
6.4.1 Interaction Probability . . . 34

7 LFA: Latent Forest Allocation **35**
7.1 Radom Spanning Trees . . . 35
7.2 Bayesian Estimation by Gibbs Sampling . . . 35
7.3 Simulation Study . . . 35
7.3.1 Score Improvement . . . 35
7.3.2 Odds of Individual Edge Improvement . . . 35

III Applications & Extentions **37**

8 Qualitative Application of Relationship Recovery **38**
8.1 Network Science Collaboration Network 38
8.2 Les Miserables Character Network 39
8.2.1 Backboning 39
8.2.2 Character Importance Estimation 39
8.3 Verbal Fluency Animal Network 39
8.3.1 Edge Connective Effiency and Diversity 39
8.3.2 Thresholded Structure Preservation 39
8.3.3 Forest Pursuit as Preprocessing 44

9 Recovery from Partial Orders **47**

List of Tables

List of Figures

1.1 Zachary's Karate Club, with ambiguously extant edge 78 highlighted. 2
1.2 . . . 3
1.3 . . . 4
1.4 graph of mutual collaboration relationships i.e. the "ground truth" social network . . . 6
1.5 Recovering underlying dependency networks from node-cooccurrences. 6

3.1 Edge Relation Observational Model . . . 19
3.2 Hyperedge Relation Observational Model . . . 21

4.1 Relating Graphs and Hypergraph/bipartite structures as adjoint operators 23

6.1 Comparison of MENDR recovery scores: **FP**: Forest Pursuit **GL**: GLASSO **CS**: Cosine Similarity **HYP**: Hyperbolic Projection **eOT**: Entropic Optimal Transport (Doubly Stochastic) **HSS**: High-Salience Skeleton **RP**: Resource Projection . . . 31
6.2 Comparison of MENDR Recovery Scores by Graph Type . . . 32
6.3 . . . 33
6.4 . . . 33
6.5 . . . 34

7.1 . . . 36
7.2 logistic regression coefficients for true edges via difference in EFM and FP scores. L2-regularization for overfit prevention was chosen with 5-fold cross validation, each time. . . . 36

8.1 134 Network scientists from [NEWMAN;BOCCALETTI;SNEPPEN], connected by co-authorship . . . 38
8.2 Max. likelihood tree dependency structure to explain co-authorships 39
8.3 Forest Pursuit estimate of NetSci collaborator dependency relationships 40
8.4 . . . 40
8.5 . . . 41
8.6 . . . 41
8.7 . . . 42
8.8 . . . 42
8.9 . . . 43
8.10 . . . 43

8.11 Comparison of backboning/dependency recovery methods tested vs. Forest Pursuit 44
8.12 When only retaining the top 2% of edge strengths, blah 45
8.13 We might prefer to drop low-certainty/rare nodes from a preserved central structure. 46

Chapter 1: Introduction

A wide variety of fields show consistent interest in inferring latent network structure from observed interactions, from human cognition and social infection networks, to marketing, traffic, finance, and many others. [9] However, an increasing number of authors are noting a lack of agreement in how to approach the metrology of this problem. This includes rampant disconnects between the theoretical and methodological network analysis sub-communities[1], treatment of error as purely aleatory, rather than epistemic [10], or simply ignoring measurement error in network reconstruction entirely[6].

1.1 Ambiguous Metrology

Networks in the "wild" rarely exist of and by themsleves. Rather, they are a model of interaction or relation *between* things that were observed. One of the most beloved examples of a network, the famed *Zachary's Karate Club*[14], is in fact reported as a list of pairwise interactions: every time a club member interacted with another (outside of the club), Zachary recorded it as two integers (the IDs of the members). The final list of pairs can be *interpreted* as an "edge list", which can be modeled with a network: a simple graph. This was famously used to show natural community

structure that nicely matches the group separation that eventually took place when the club split into two.[12]

Note, however, that we could have just as easily taken note of the instigating student for each interaction (i.e. which student initiated conversation, or invited the other to socialize, etc.). If that relational asymmetry is available, our "edges" are now *directed*, and we might be able to ask questions about the rates that certain students are asked vs. do the asking, and what that implies about group cohesion. Additionally, the time span is assumed to be "for the duration of observation" (did the students *ever* interact), but if observation time was significantly longer, say, multiple years, we might question the credulity of treating a social interaction 2 years ago as equally important to an interaction immediately preceding the split. This is now a "dynamic" graph; or, if we only measure relative to the time of separation, at the very least a "weighted" one.

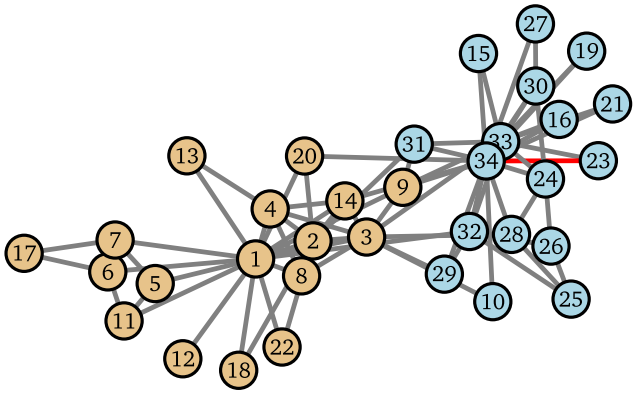


Figure 1.1: Zachary's Karate Club, with ambiguously extant edge 78 highlighted.

We do not know if any of these are true. In fact, as illustrated in Figure 1.1, we do not know if the network being described from the original edge data even has

77 or 78 edges, due to ambiguous reporting in the original work. Lacking a precise definition of what the graph's components (i.e. it's edges) are, *as measurable entities*, means we cannot estimate the measurement error in the graph.

1.2 Indirect Network Measurement

While the karate club graph has unquantified edge uncertainty derived from ambiguous edge measurements, we are fortunate that we *have edge measurements*. Regardless of how the data was collected, it is de facto reported as a list of pairs. In many cases, we simply do not have such luxury. Instead, our edges are only measured *indirectly*, and instead we are left with lists of node co-ocurrences. Networks connecting movies as being "similar" might be derived from data that lists sets of movies watched by each user; networks of disease spread pathways might be implied from patient infection records; famously, we might build a network of collaboration strength between academic authors by mining datasets of the papers they co-author together.

Such networks are derived from what we will call *node activation* data, i.e., records of what entities happened "together", whether contemporaneously, or in some other context or artifact.

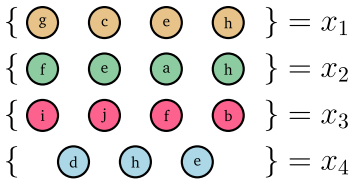


Figure 1.2

These are naturally represented as "bipartite" networks, having separate entites for,

say, “papers” and “authors”, and connecting them with edges (paper 1 is “connected” to its authors E,H,C, etc.). But analysts are typically seeking the collaboration network connecting authors (or papers) themselves! Networks of relationships in this situation are not directly observed, but which *if recovered* could provide estimates for community structure, importances of individual authors (e.g. as controlling flow of information), and the “distances” that separate authors from each other, in their respective domains. [13] Common practice assumes that co-authorship in any paper is sufficient evidence of at least some level of social “acquaintance”, where more papers shared means more “connected”.

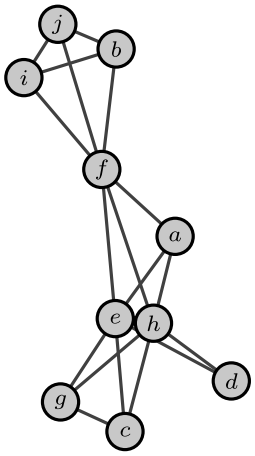


Figure 1.3

Thus our social collaboration network is borne out of indirect measurements: author connection is implied through “occasions when co-authorship occurred”. However, authors of papers may recall times that others were added, not by their choice, but by someone else already involved. In fact, the final author list of most papers is reasonably a result of individuals choosing to invite others, not a unanimous, simultaneous decision by all members. Let’s imagine we wished to study the social

network of collaboration more directly: if we had the luxury of being in situ as, say, a sociologist performing an academic ethnography, we might have been more strict with our definition of "connection". If the goal is a meaningful social network reflecting the strength of interraction between colleages, perhaps the we prefer our edges represent "mutual willingness to collaborate". Edge "measurement", then, could involve records of events that show willingness to seek or participate in collaboration event, such as:

- *author (g) asked (e), (h), and (c) to co-author a paper, all of whom agreed*
- *(i) asked (f) and (j), but (j) wanted to add (b)'s expertise before writing one of the sections*

and so on. Each time two colleagues had an opportunity to work together *and it was seized upon* we might conclude that evidence of their relationship strengthed. With data like this, we could be more confident in claiming our collaboration network can serve as "ground truth," as far as empirically confirmed collaborations go. However, even if the underlying "activations" are identical, our new, directly measured graph looks very different.

Fundamentally, the network in Figure 1.4 shows which relationships the authors *depend* on to accomplish their publishing activity. When causal relations between nodes are being modeled as edges, we call such a graph a *dependency network*. We will investigate this idea further later on, but ultimately, if a network of dependencies is desired (or implied, based on analysis needs), then the critical problem remaining is *how do we recover dependency networks from node activations?* Additionally, what goes wrong when we use co-occurence/activation data to estimate the dependency

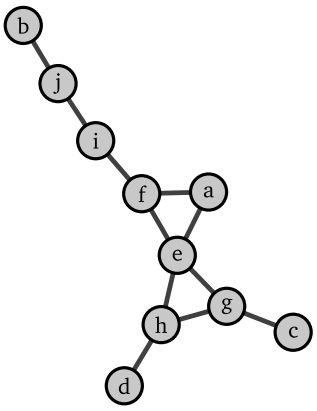


Figure 1.4: graph of mutual collaboration relationships i.e. the “ground truth” social network

network, especially when we wish to use it for metrics like centrality, shortest path distances, and community belonging?

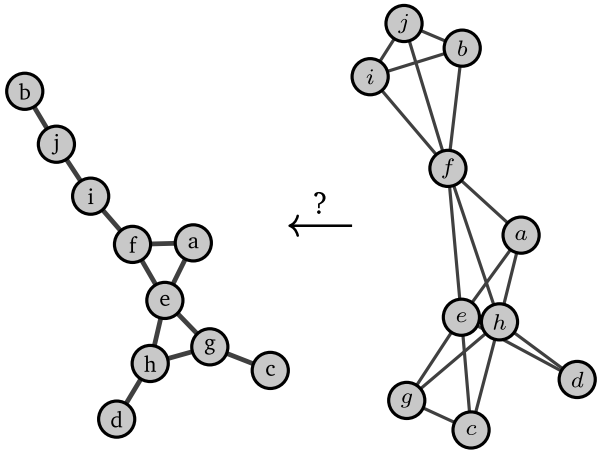


Figure 1.5: Recovering underlying dependency networks from node-cooccurrences.

Even more practically, networks created directly from bipartite-style data are notorious for quickly becoming far too dense for useful analysis, earning them the (not-so-)loving moniker “hairballs”. Network “backboning,” as it has come to be called [CITE] tries to find a subset of edges in this hairball that still captures it’s core topology in a way that’s easier to visualize. Meanwhile, underlying networks of dependencies that *cause* node activation patterns can provide this: they are almost

always more sparse than their hairballs. Accessing the dependency *backbone* in a principled way is difficult, but doing so in a rapid, scalable manner is critical for practitioners to be able to make use of it to trim their hairballs.

1.3 Scope of this work

The purpose of this thesis is to provide a solid foundation for basic edge metrology when our data consists of binary node activations. We give special focus to binary activations that occur due to spreading processes, such as random walks or cascades on an underlying carrier graph. Recovering the carrier, or, "dependency" network from node activations is of great interest to the network backboning and causal modeling communities, but often involves wither unspoken sources of epistemic and alleatoric error, or high computation costs (or both). To begin adressing these issues, we present a guide to current practices,pitfalls, and how common statistical tools apply to the network recovery problem: a *Practitioner's Guide to Network Recovery*. We cover what "measurement" means in our context, and specifically the ways we encode observations, operations, and uncertainties numerically Clarifying what different versions of what "relation" means (whether proximity or incidence), since network structure is intended to encode such relations as mathematical objects, despite common ambiguities and confusion around what practitioners intend on communicating through them. Then we use this structure to present a cohesive framework for selecting a useful network recovery technique, based on the available data and where in the data processing pipeline is acceptable to admit either extra

modeling assumptions or information loss.

Next, building on a gap found in the first part, we present a novel method, *Forest Pursuit*, to extract depencency networks when we know a *spreading process* causes node activation (e.g. paper co-authorship caused by collaboration requests). We create a new reference dataset to enable community benchmarking of network recovery techniques, and use it show greatly improved accuracy over many other widely-used methods. Forest Pursuit in its simplest form scales linearly with the size of active-node sets, being trivially parallelizeable and streamable over dataset size, and agnostic to network size overall. We then expand our analysis to re-imagine Forest Pursuit as a Bayesian probabilistic model, *Latent Forest Allocation*, which has an easily-implemented Expectation Maximization scheme for posterior estimation. This significantly improves upon the accuracy results of Forest Pursuit, at the cost of some speed and scalability, giving analysts multiple options to adapt to their needs.

Last, we apply Forest Pursuit to several qualitative case-studies, including a scientific collaboration network, and the verbal fluency "animals" network recovery problem, which dramatically change interpretation under use of our method. We investigate its use as a low-cost preprocessor for other methods of network recovery,like GLASSO, improving their stability and interpretability. Finally we discuss the special case when node activations are reported as an ordered set, where accounting for cascade-like effects becomes crucial to balance false positive and false-negative edge prediction. Along with application of this idea to knowledge-graph creation from technical language in the form maintenance work-order data, we discuss more broadly the future needs of network recovery, specifically in the context of embeddings and

gradient-based machine learning toolkits.

Part I

A Practitioner's Guide to Network Recovery

Chapter 2: Metrology as matrices

Where metrology is concerned, the actual unit of observation and how it is encoded for us is critical to how analysts may proceed with quantifying, modeling, and measuring uncertainty around observed phenomena. Experiment and observation tends to be organized as inputs and outputs, or, independent variables and dependent variables, specifically. Independent variables are observed, multiple times ("observations"), and changes in outcome for each can be compared to the varying values associated with the independent variable input ("features"). For generality, say a practitioner records their measurements as scalar values, i.e. $x \in \mathbb{S} \in \{\mathbb{R}, \mathbb{Z}, \mathbb{N}, \cdots\}$. The structure most often used to record scalar values of n independent/input variable features over the course of m observations is called a design matrix $X : \mathbb{S}^{m \times n}$.[1]

[1]Not all observations are scalar, but they can become so. If individual measurments are higher-dimensional (e.g. images are 2D), X is a tensor, which can be transformed through unrolling or embedding into a lower dimensional representation before proceeding. There are other techniques for dealing with e.g. categorical data, such as one-hot encoding (where the features are binary for each possible category, with boolean entries for each observation).

2.1 Observation and feature "spaces"

If we index a set of observations and features, respectively, as

$$i \in I = \{1, \cdots, m\}, \quad j \in J = \{1, \cdots, n\}, \qquad I, J : \mathbb{N}$$

then the design matrix can map the index of an observation and a feature to the corresponding measurement.

$$x = X(i, j) \qquad X : I \times J \to \mathbb{S}$$

i.e. the measured value of the *j*th independent variable from the *i*th observation. In this scheme, an "observation" is a single row vector of features in $\mathbb{S}^{n \times 1}$ (or simply $\mathbb{S}^n$), such that each observation encodes a position in the space defined by the features, i.e. the *feature space*, and extracting a specific observation vector i from the entire matrix can be denoted as

$$\mathbf{x}_i = X(i, \cdot), \quad \mathbf{x} : J \to \mathbb{S}$$

Similarly, every "feature" is associated with a single column vector in $\mathbb{S}^{1 \times m}$, which can likewise be interpreted as a position in the space of observations (the *data space*):

$$\mathbf{x}^*_j = X(\cdot, j), \quad \mathbf{x}^* : I \to \mathbb{S}$$

Note that this definition could be swapped without loss of generality. In other words, $\mathbf{x}$ and $\mathbf{x}^*$ being in row and column spaces is somewhat arbitrary, having more to do with the logistics of experiment design and data collection. We could have measured our feature vectors one-at-a-time, measuring their values over an entire "population", in effect treating that as the independent variable set.[2]

To illustrate this formalism in a relevant domain, let's take another classic network example from academia: co-citation networks. [DRAW BIPARTITE] Lists of co-authors on publications is often reported as "network" data, and subjected to network analysis techniques. For m papers we might be aware of a total of n authors. For a given paper, we are able to see which authors are involved, and we say those authors "activated" for that paper. It makes sense that our observations are individual papers, while the features might be the set of possible authors. However, we are not given information about which author was invited by which other one, or when each author signed on. In other words, the measured values are strictly boolean, and we can structure our dataset as a design matrix $X : \mathbb{B}^{m \times n}$. We can then think of the i^{th} paper as being represented by a vector $\mathbf{x}_i : \mathbb{B}^n$, and proceed using it in our various statistical models. If we desired to analyze the set of authors, say, in order to determine their relative neighborhoods or latent author communities, we could equally use the feature vectors for each paper, this time represented in a vector $\mathbf{x}_j^* : \mathbb{B}^{1 \times m}$.

[2] In fact, vectors are often said to be in the column-space of a matrix, especially when using them as transformations in physics or deep learning layers. We generally follow a one-observation-per-row rule, unless otherwise stated.

2.2 Models & linear operators

Another powerful tool an analyst has is *modeling* the observation process. This is relevant when the observed data is hypothesized to be generated by a process we can represent mathematically, but we do not know the parameter values to best represent the observations (or the observations are "noisy" and we want to find a "best" parameters that account for this noise). This is applicable to much of scientific inquiry, though one common use-case is the de-blurring of observed images (or de-noising of signals), since we might have a model for how blurring "operated" on the original image to give us the blurred one. We call this "blurring" a *linear operator* if it can be represented as a matrix[3], and applying it to a model with l parameters is called the *forward map*:

$$\mathbf{x} = F\mathbf{p} \qquad F : \mathbb{R}^l \to \mathbb{R}^n$$

where P is the space of possible parameter vectors, i.e. the *model space*. The forward map takes a modeled vector and predicts a location in data space.

Of critical importance, then, is our ability to recover some model parameters from our observed data, e.g. if our images were blurred through convolution with a blurring kernel, then we are interested in *deconvolution*. If F is invertible, the most direct solution might be to apply the operator to the data, as the *adjoint map*:

$$\mathbf{p} = F^H\mathbf{x} \qquad F^H : \mathbb{R}^n \to \mathbb{R}^l$$

[3] in the finite-dimensional case

which removes the effect of F from the data $\mathbf{x}$ to recover the desired model $\mathbf{p}$.

Trivially we might have an orthogonal matrix F, so $F^H = F^{-1}$ is available directly. In practice, other approaches are used to minimize the *residual*: $\hat{\mathbf{p}} = \min_{\mathbf{p}} F\mathbf{p} - \mathbf{x}$. Setting the gradient to 0 yeilds the normal equation, such that

$$\hat{\mathbf{p}} = (F^T F)^{-1} F^T \mathbf{x}$$

This should be familiar to readers as equivalent to solving ordinary least-squares (OLS). However, in that case it is more often shown as having the *design matrix* X in place of the operator F.

This is a critical distinction to make: OLS as a "supervised" learning method treats some of the observed data we represented as a design matrix previously as a target to be modeled, $y = X(\cdot, j)$, and the rest maps parameters into data space, $F = X(\cdot, J/j)$. With this paradigm, only the target is being "modeled" and the rest of the data is used to create the operator. In the citation network example, it would be equivalent to trying to predict one author's participation in every paper, *given* every other author's participation in them.

For simplicity, most work in the supervised setting treats the reduced data matrix as X, opting to treat y as a separate *dependent variable*. However, our setting will remain *unsupervised*, since no single target variable is of specific interest—all observations are "data". In this, we more closely align with the deconvolution literature, such that we are seeking a model and an operation on it that will produce the observed behavior in an "optimal" way.

2.3 Measurement quantification & error

- marginal sums
- rule of succession
- type I and II Error?
- Epistemic and Alleatoric Uncertainty?

Ultimately we are not great at specifying what “being related” actually means…

2.4 Proximity vs. Incidence

2.4.1 Kernels & distances

Importantly for the use of linear algebra, these values assigned for each feature are assumed to exist in a field (or, more generally, a semiring) R, equipped with operators analogous to addition ($\oplus$) and multiplication ($\otimes$) that allow for values to be aggregated through an inner product. The matrix of all pairs of inner-products found by matrix multiplication (contracting over the feature space) is given by:

$$G(\mathbf{x}_i, \mathbf{x}_j, R) = G_{ij} = \bigoplus_{k=1}^{n} x_{ik} \otimes x_{kj}$$

such that real-valued entries and a traditional “plus-times” inner product recovers the Gram matrix $G_{ij} = \sum_{k=1}^{n} x_{ik} x_{kj}$, or simply $G = X^T X$.

How “close” or “far away” things are…. Avrachenkov et al.

Important: these measurements often assume distance is defined in terms of the

measurements/objects/data, but for *inverse problems*, structure learning, etc., they are more often applied in terms of the features/operators.

Example with doc-term matrices

The inner product between two papers will yield a "true" only if two papers share at least one author in common. This is called a *bipartite projection*[CITE], specifically the "papers" projection.

Similarly, if our goal is to determine a network of "whether two authors ever coauthored", we could perform a bipartite projection using the boolean inner product in the observation space i.e. the "authors" projection. It is this second projection, for determining a structure between features embedded into the "observation" space, that we are primarily concerned with in this work, since it is the view that most closely resembles the concept of covariance or correlation between independent variables (features) in statistics more generally.

2.4.2 Incidence structures & dependency

foundational model of graph theory and incidence structures more broadly. More to come, but get the terminology down.

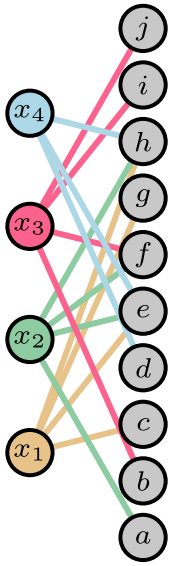
- Spring example, road example, etc.
- partial correlations

2.4.3 Implications for networks

Usually dependencies are taken as causing or enabling proximity. E.g. shortest paths, vs. edges.

$$
\begin{array}{c}
 & \begin{array}{cccccccccc} a & b & c & d & e & f & g & h & i & j \end{array} \\
\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \end{array} &
\left[\begin{array}{cccccccccc}
0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
 & & & & \vdots & & & & &
\end{array}\right]
\end{array}
$$

(a)



(b) Bipartite representation of node "activation" data

- Discuss Complex Systems and their representation.

The approach taken by researchers/investigators…do they assume a level of interchangeability between the two kinds of "relation"? Do they define Or do they

Chapter 3: Incidence in Vector Space

From [3], and linalg book, and hypergraph incidence model

For each, describe the meaning of

1. Observation model (and connection to node activation data)
2. Vector space representation
3. Inner product (linear kernel)
4. Induced Norms

3.1 Dependence as Graph

Initial notation (set-based) and outline of section

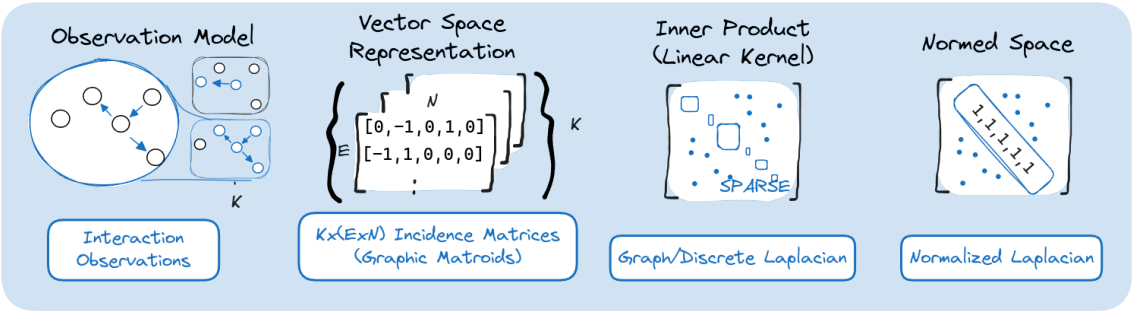


Figure 3.1: Edge Relation Observational Model

3.1.1 Edges as Vectors of Nodes

3.1.2 Inner Product on Edges

Laplacian as inner product on incidence observations. Associated objects (degree vector, o)

Rescaling to achieve normaalization.

Use to define kernels (and application e.g. soft-cosine measure)

3.1.3 From Edge Observations to Node Activations

Strictly speaking, we can't say that nodes are directly observed in this space… edges are. Collections of nodes are measured two-at-a-time (one-per-edge being traversed).

Another way to approach is to view inner products as a sum of outer products. A each edge uniquely corresponds to 2 nodes (in a simple graph). Use triangle unfolding for closed form bijection.

Unrolling 3D tensor of subgraphs along eads to a secondary representation of graphs as an *edgelist*, having binary activation vectors on edges rather than nodes. Then each observation in this model is necessarily a set of activated edges. The non-zero (visited) nodes are found using the incidence matrix as an operator.

3.2 Occurence as Hypergraph

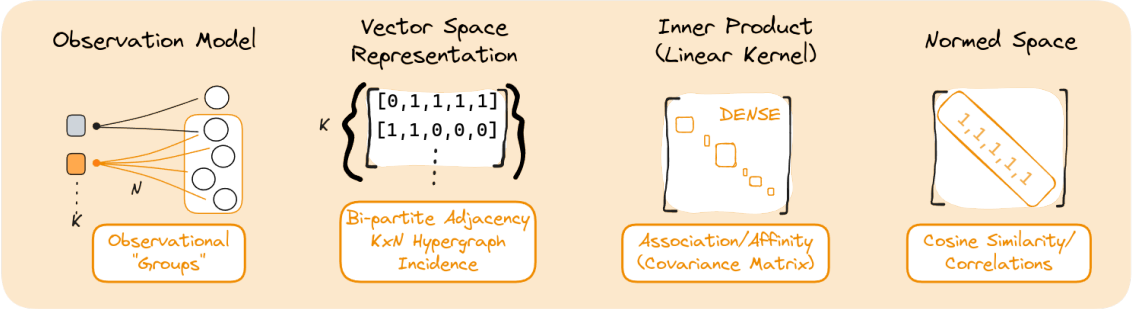


Figure 3.2: Hyperedge Relation Observational Model

3.2.1 Hyperedges as Vectors of Nodes

3.2.2 Inner product on Hyperedges

Roundabout way of describing binary/occurrence data. Inner product is co-occurrences.

Leads to correlation/covariance, etc.

3.3 Combining Occurence & Dependence

- soft cosine
- kernels on graphs (incl. coscia euclidean)
- Retrieving one from the other is hard.

Chapter 4: Roads to Network Recovery

4.1 Choosing a structure recovery method

Takeaway: a way to organize existing algorithms, AND highlight unique set of problems we set out to solve

4.2 Organizing Recovery Methods

i.e. Network Recovery as an Inverse Problem, and what information is had at each point.

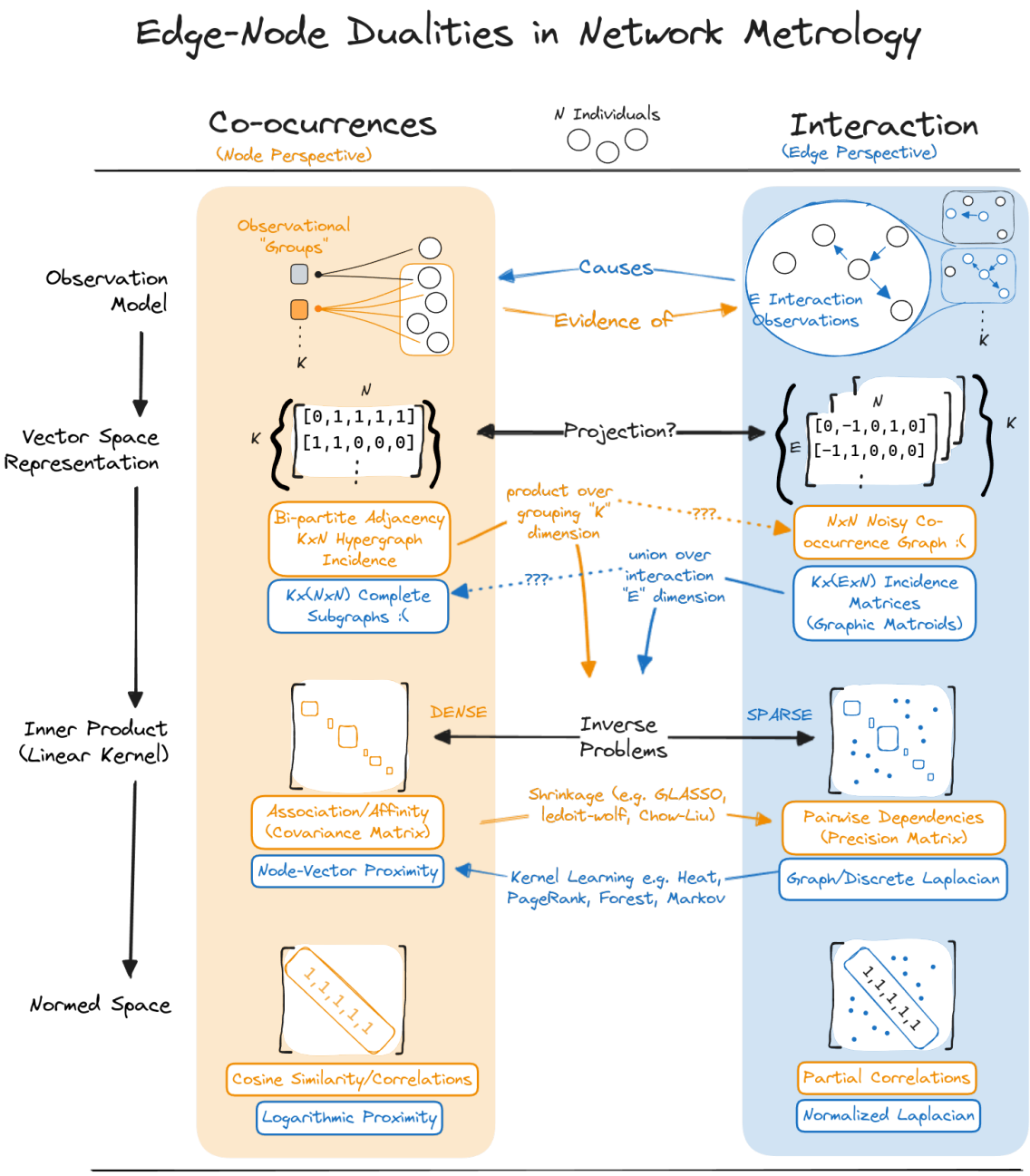


Figure 4.1: Relating Graphs and Hypergraph/bipartite structures as adjoint operators

4.2.1 Observing Nodes vs Edges

4.2.2 Embeddings, Inner Products, & Preprocessing

4.3 Tracing Information Loss Paths

4.3.1 Table of Existing Approaches

- Observation-level loss (starting with the inner product or kernel)
- Non-generative model loss (no projection of data into model space)
- no uncertainty quantification

4.3.2 A Path Forward

Sorting algorithms… *none address all three!*

i.e. MOTIVATES FOREST PURSUIT

Part II

Nonparametric Network Recovery With Random Spanning Forests

Chapter 5: Generative Random Spanning Forests

Addressing gaps discussed in the previous section to reach a generative model for network recovery requires careful attention to the generation mechanism for node activations. While there are many ways we might imagine bipartite data to be be generated, presuming the existence of a dependency graph that *causes* activation patterns will give us useful ways to narrow down the generative specification. The dependency graph gives us all of the ways that nodes' state can affect the state of others, i.e. the neighbor set of each node. This immediately leads us to model our node activations as resulting from *spreading*, or, *diffusive processes*.

In this chapter we outline how the random-walks are is related to these diffusive models of graph traversal, enabled by an investigation of the graph's "regularized laplacian" from Avrachenkov et al. [7]. Then we use the implicit causal dependency tree structure of each observation, together with the Matrix Forest Theorem [8, 11] to more generally define our generative node activation model: namely, as samples from the space of rooted random spanning forests on the dependency graph.

5.1 Node Activation by Diffusive Processes

The class of diffusive processes we focus on "spread" from one node to another. If a node is activated, it is able to activate other nodes it is connected to, directly encoding our need for the graph edges to represent that nodes "depend" on others to be activated. In this case, a node activates when another node it depends on spreads their state to it. These single-cause activations are equivalent to imagining a random-walk on the dependency graph, where visiting a node activates it.

5.1.1 Random Walk Activations

Random walks are regularly employed to model spreading and diffusive processes on networks. If a network consists of locations, states, agents, etc. as "nodes", and relationships between nodes as "edges", then random walks consist of a stochastic process that "visits" nodes by randomly "walking" between them along connecting edges. Epidemiological models, cognitive search in semantic networks, stock price influences, website traffic routing, social and information cascades, and many other domains also involving complex systems, have used the statistical framework of random walks to describe, alter, and predict their behaviors. [CITE…lots?]

When network structure is known, the dynamics of random-walks are used to capture the network structure via sampling [LITTLEBALLOFFUR, etc], estimate node importance's[PAGERANK], or predict phase-changes in node states (e.g. infected vs. uninfected)[SIR I think] In our case, Since we have been encoding the activations as binary activation vectors, the "jump" information is lost—activations are "emitted"

for observation only upon the random walker's initial visit. [CITE INVITE] In many cases, however, the existence of relationships is not known already, and analysts might *assume* their data was generated by random-walk-like processes, and want to use that knowledge to estimate the underlying structure of the relationships between nodes.

- useful tool for analysis of our data: reg laplacian
- interpretations

5.1.2 Dependencies as Trees

The whole graph isn't a tree….Every data point is.

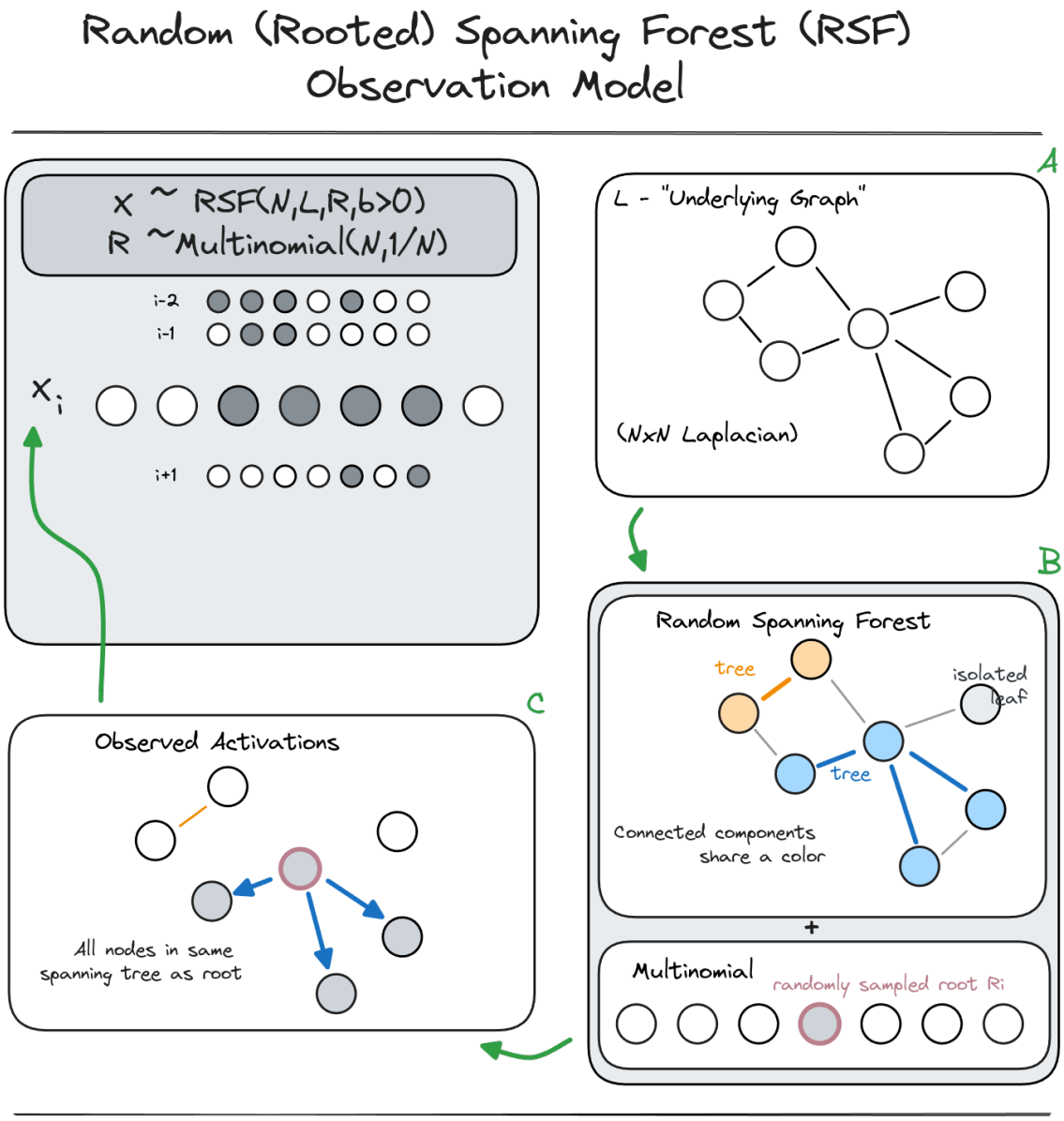
[GRAPHIC 1 - my data]

[GRAPHIC 2 - infection vector from meta node]

5.1.3 Matrix Tree and Forest Theorems

- one from kirchoff
- one from Chebotarv

5.2 Generative Model Specification



- hierarchical model - marginalize over the root node.

Chapter 6: Forest Pursuit: Approximate Recovery in Near-linear Time

filling the gap we saw in the literature

6.1 Sparse Dictionary Learning

6.1.1 Problem Specification

6.1.2 Matching Pursuit

6.1.3 Space of Spanning Forests

6.2 Forest Pursuit: Approximate Recovery in Near-linear Time

I.e. the PLOS paper (modified basis-pursuit via MSTs) ### Algorithm Summary

6.2.1 Uncertainty Estimation

6.2.2 Approximate Complexity

6.3 Simulation Study

6.3.1 Method

6.3.2 Results - Scoring

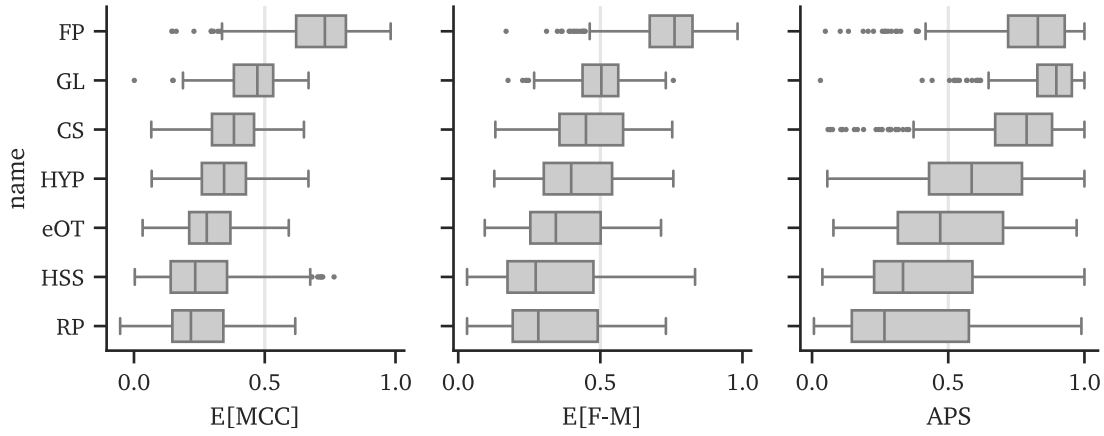


Figure 6.1: Comparison of MENDR recovery scores: **FP**: Forest Pursuit **GL**: GLASSO **CS**: Cosine Similarity **HYP**: Hyperbolic Projection **eOT**: Entropic Optimal Transport (Doubly Stochastic) **HSS**: High-Salience Skeleton **RP**: Resource Projection

6.3.3 Results - Performance

```
(np.float64(0.0694651520667083), np.float64(1085594.3113734804))
```

```
4.594236123462085 1632.4803075964253
```

```
8.436143086693106 355.6127449677926
```

```
8.374142176881948 3379.80886903671
```

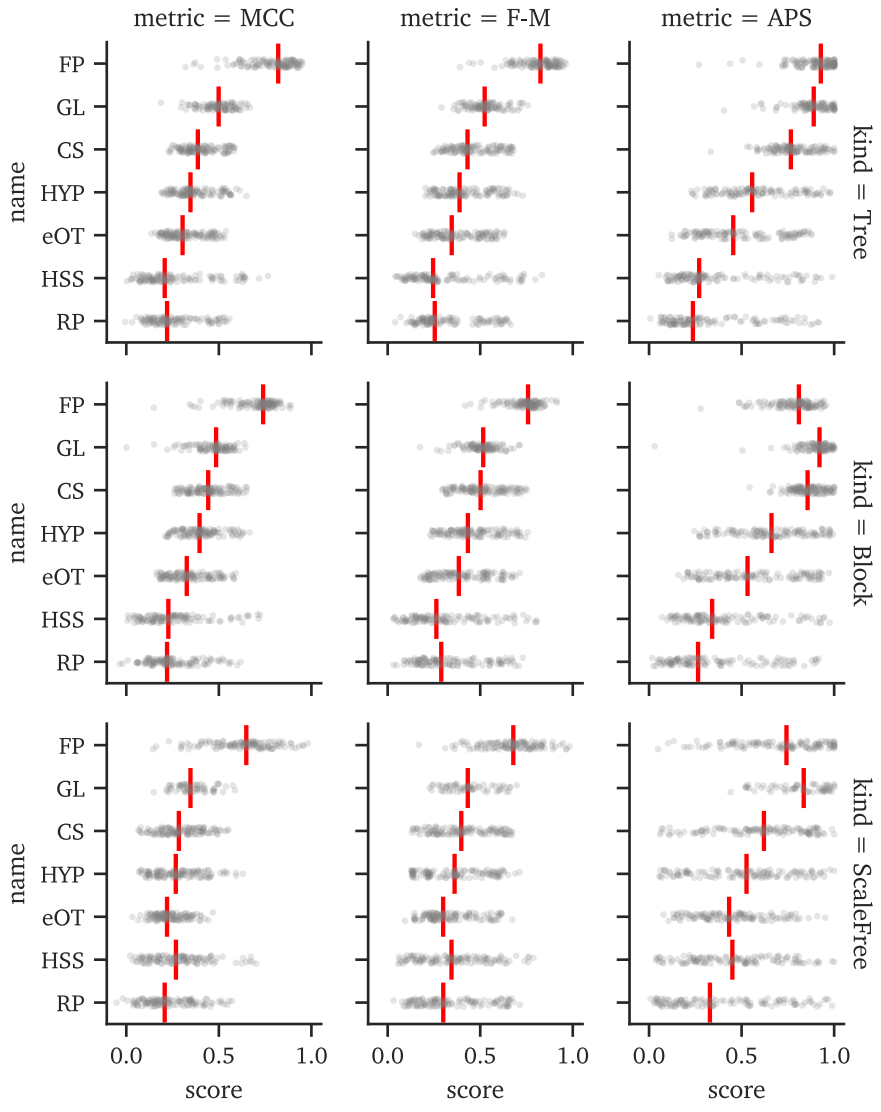


Figure 6.2: Comparison of MENDR Recovery Scores by Graph Type

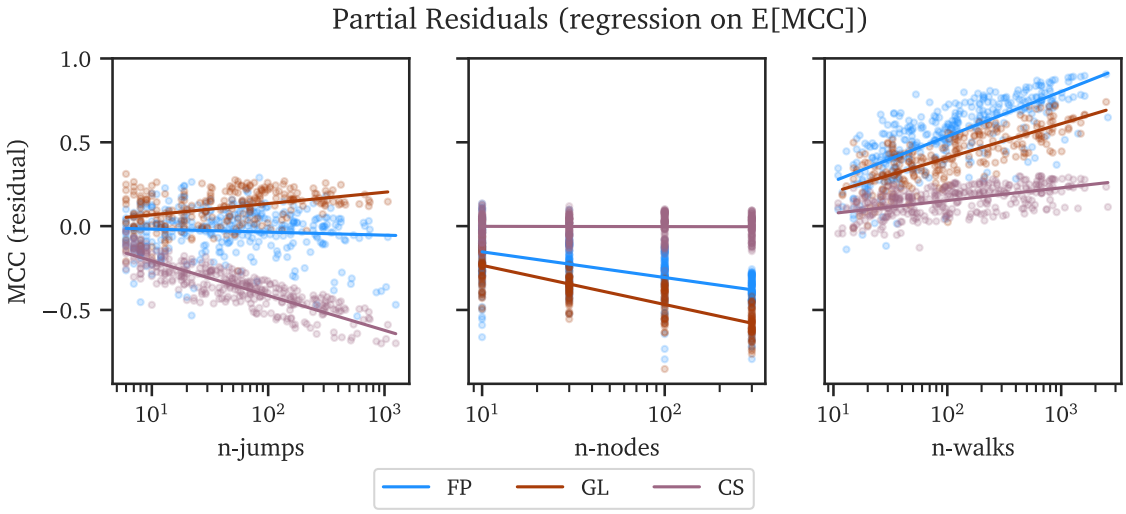


Figure 6.3

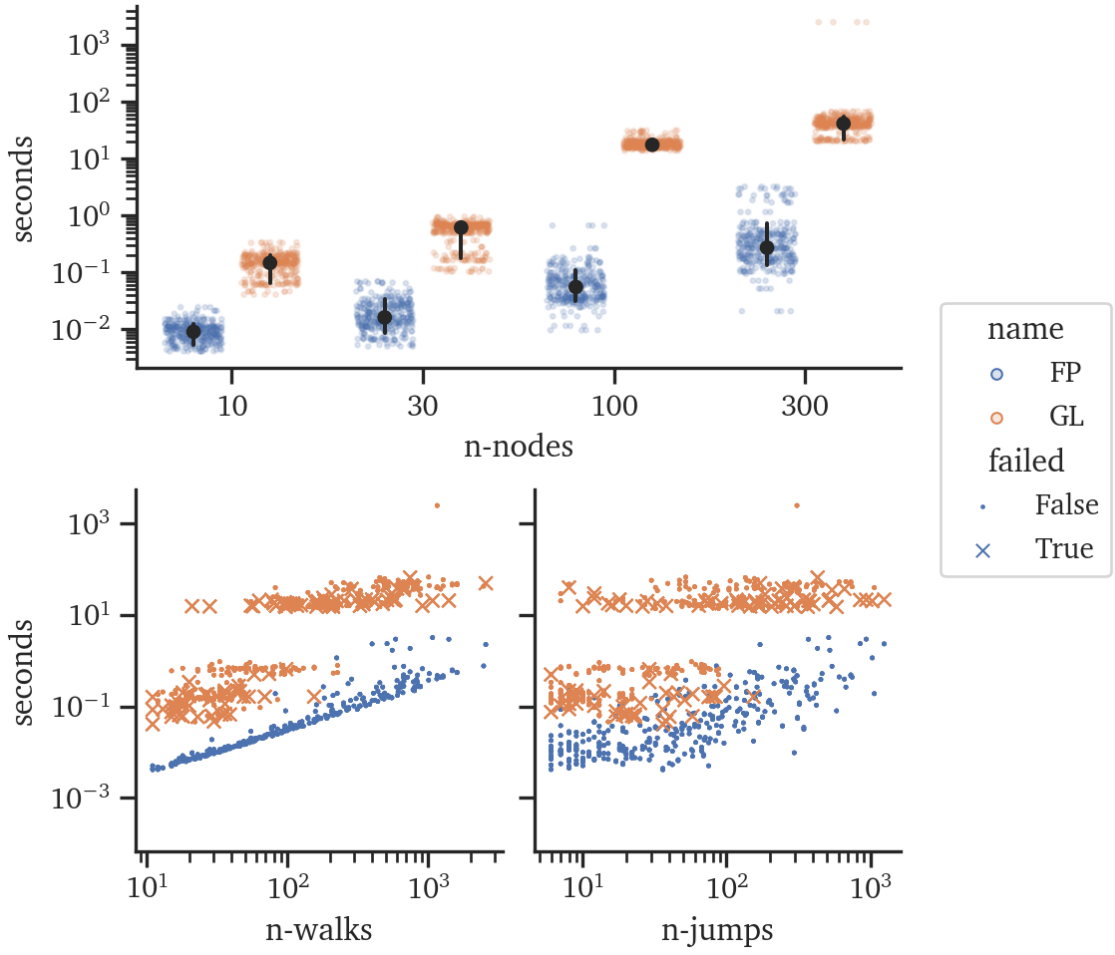


Figure 6.4

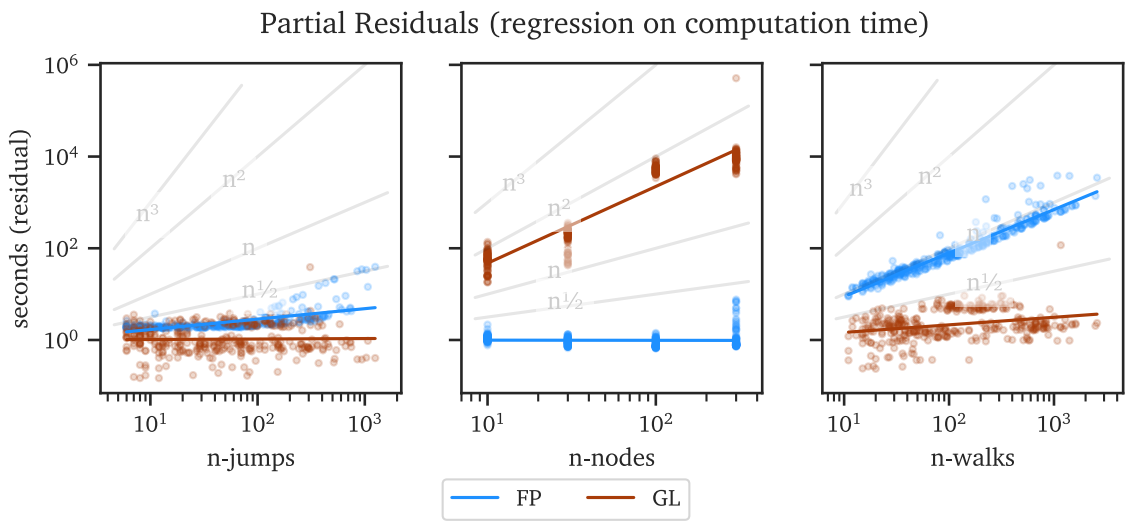
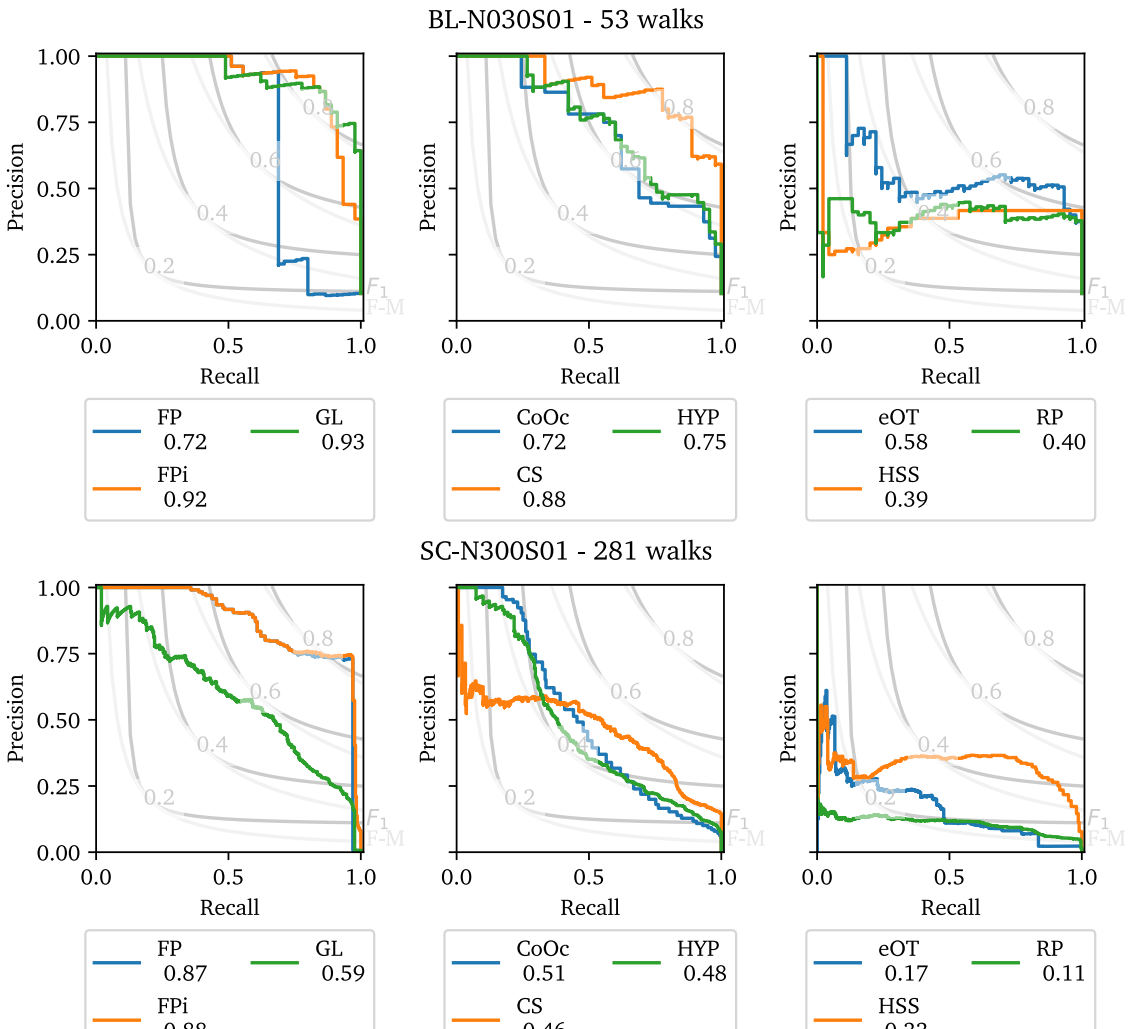


Figure 6.5

6.4 Discussion

6.4.1 Interaction Probability



Chapter 7: LFA: Latent Forest Allocation

7.1 Radom Spanning Trees

- Methods for sampling i.e. wilson's and Duan's (other? Energy paper?)
- Tree Likelihoods, other facts

7.2 Bayesian Estimation by Gibbs Sampling

- comparison with LDA
- Simplifying Assumptions (conditional prob IS prob for this)

I.e. the unwritten paper, modifying technique by Duan and Dunson [2] for RSF instead of RSTs

7.3 Simulation Study

7.3.1 Score Improvement

7.3.2 Odds of Individual Edge Improvement

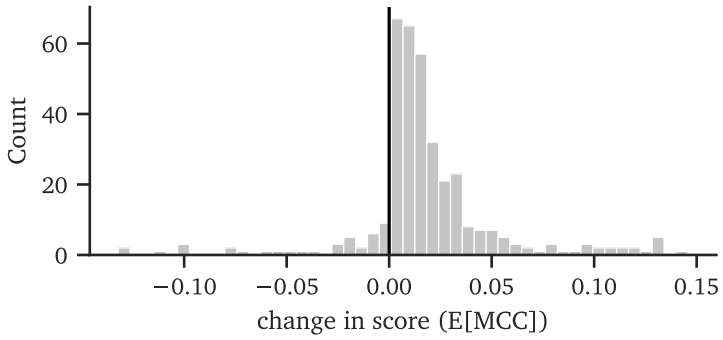


Figure 7.1

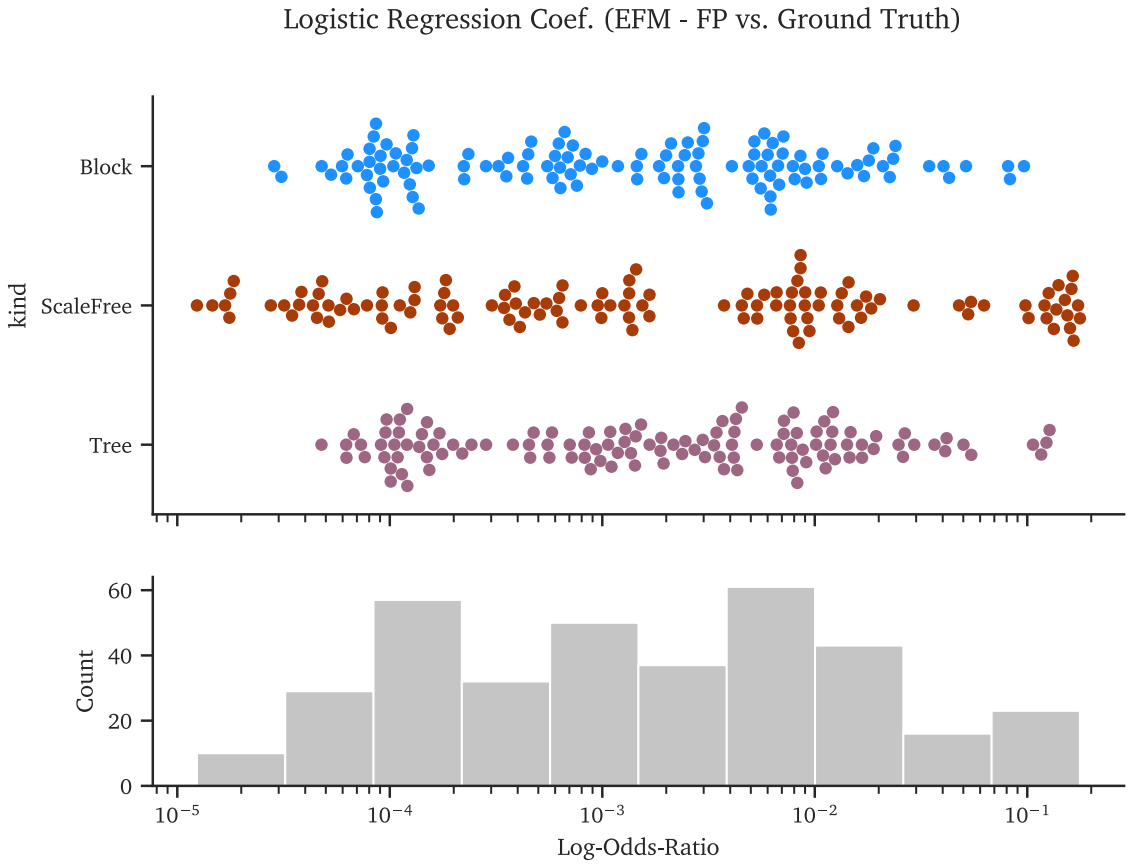


Figure 7.2: logistic regression coefficients for true edges via difference in EFM and FP scores. L2-regularization for overfit prevention was chosen with 5-fold cross validation, each time.

Part III

Applications & Extentions

Chapter 8: Qualitative Application of Relationship Recovery

8.1 Network Science Collaboration Network

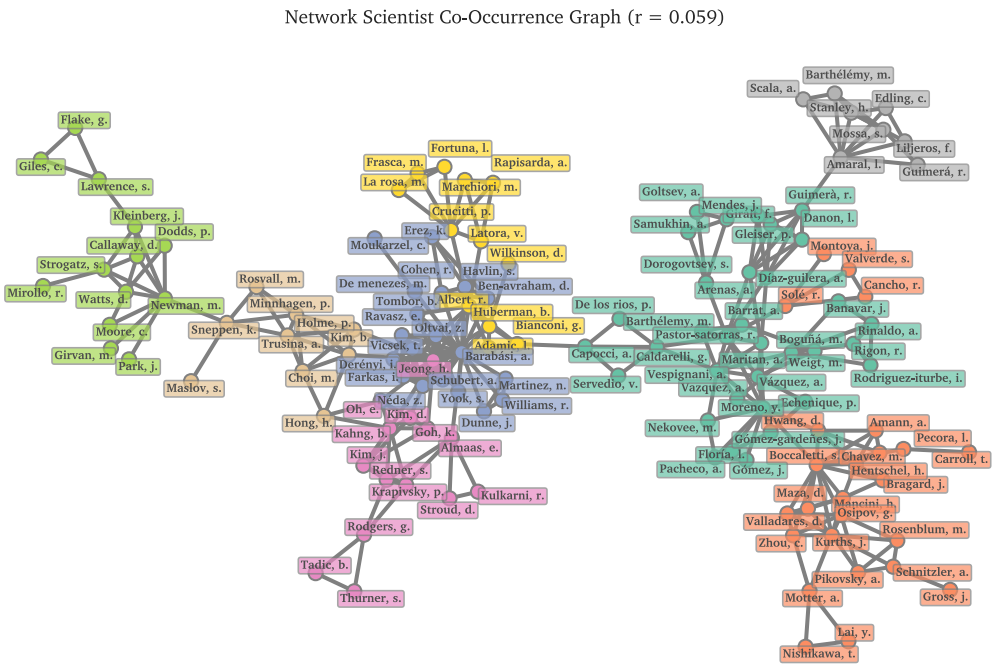


Figure 8.1: 134 Network scientists from [NEWMAN;BOCCALETTI;SNEPPEN], connected by co-authorship

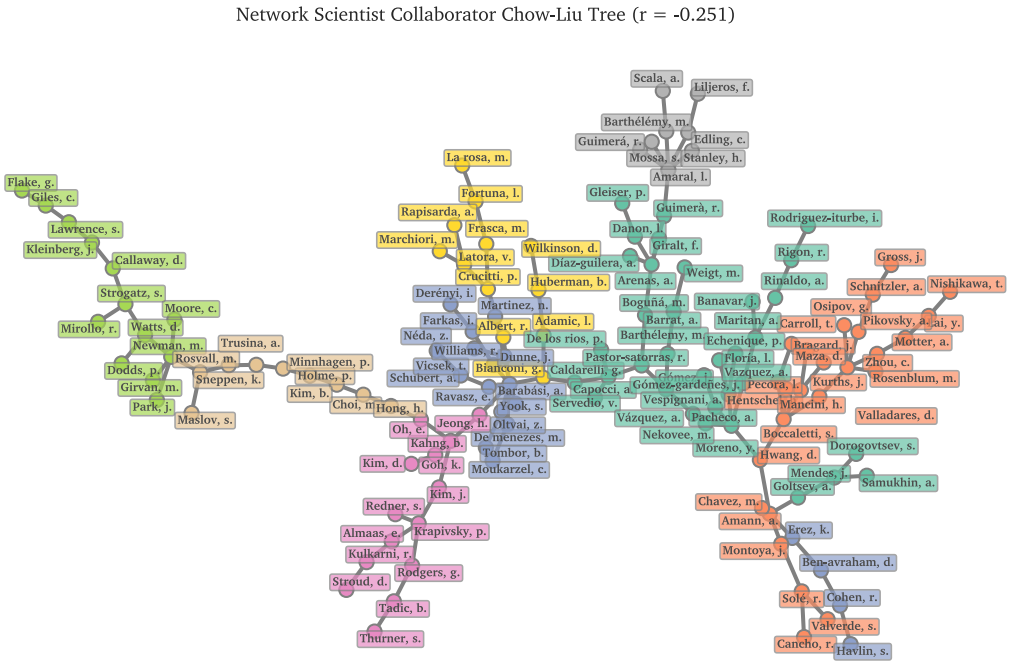


Figure 8.2: Max. likelihood tree dependency structure to explain co-authorships

8.2 Les Miserables Character Network

8.2.1 Backboning

8.2.2 Character Importance Estimation

8.3 Verbal Fluency Animal Network

8.3.1 Edge Connective Effiency and Diversity

8.3.2 Thresholded Structure Preservation

Differences in structural preservation with increased thresholding.

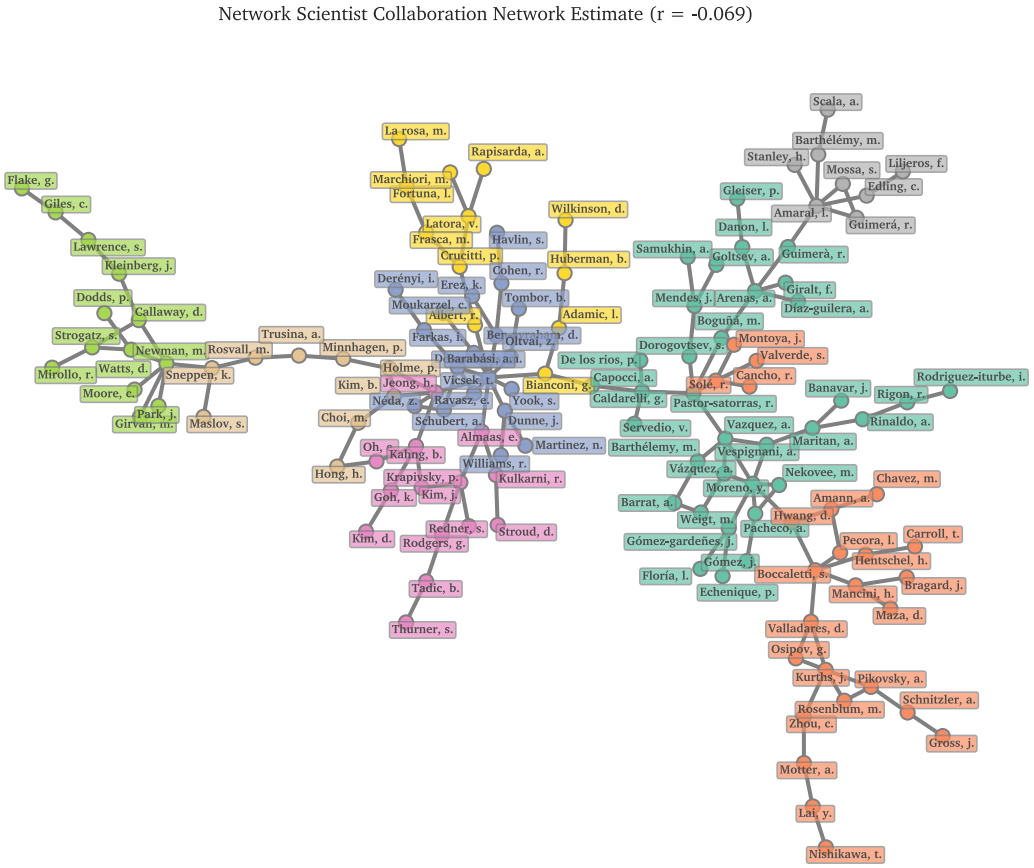


Figure 8.3: Forest Pursuit estimate of NetSci collaborator dependency relationships

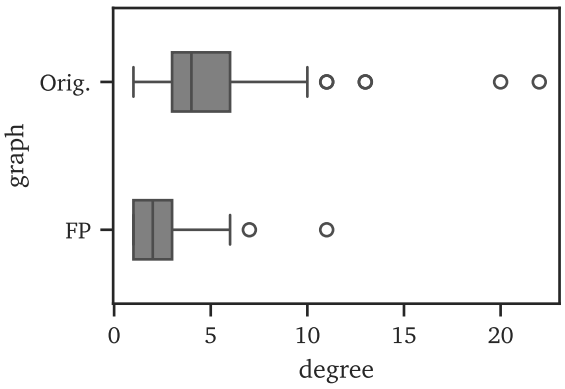


Figure 8.4

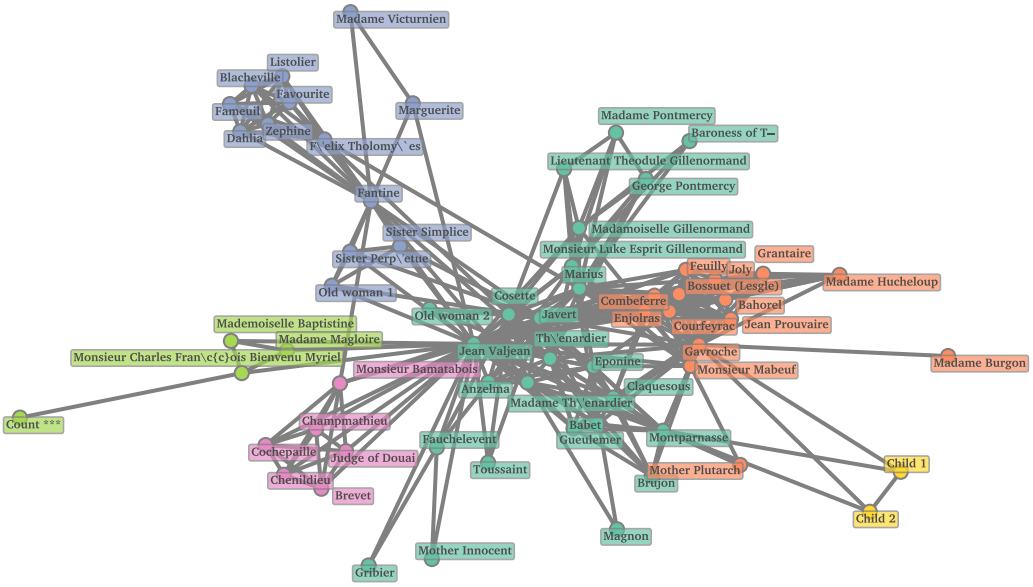


Figure 8.5

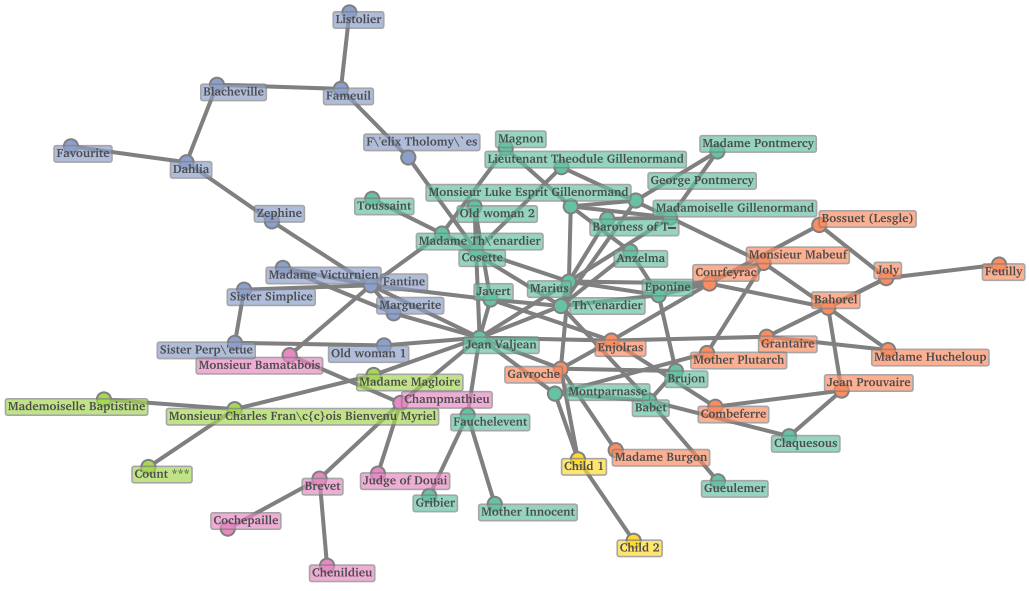


Figure 8.6

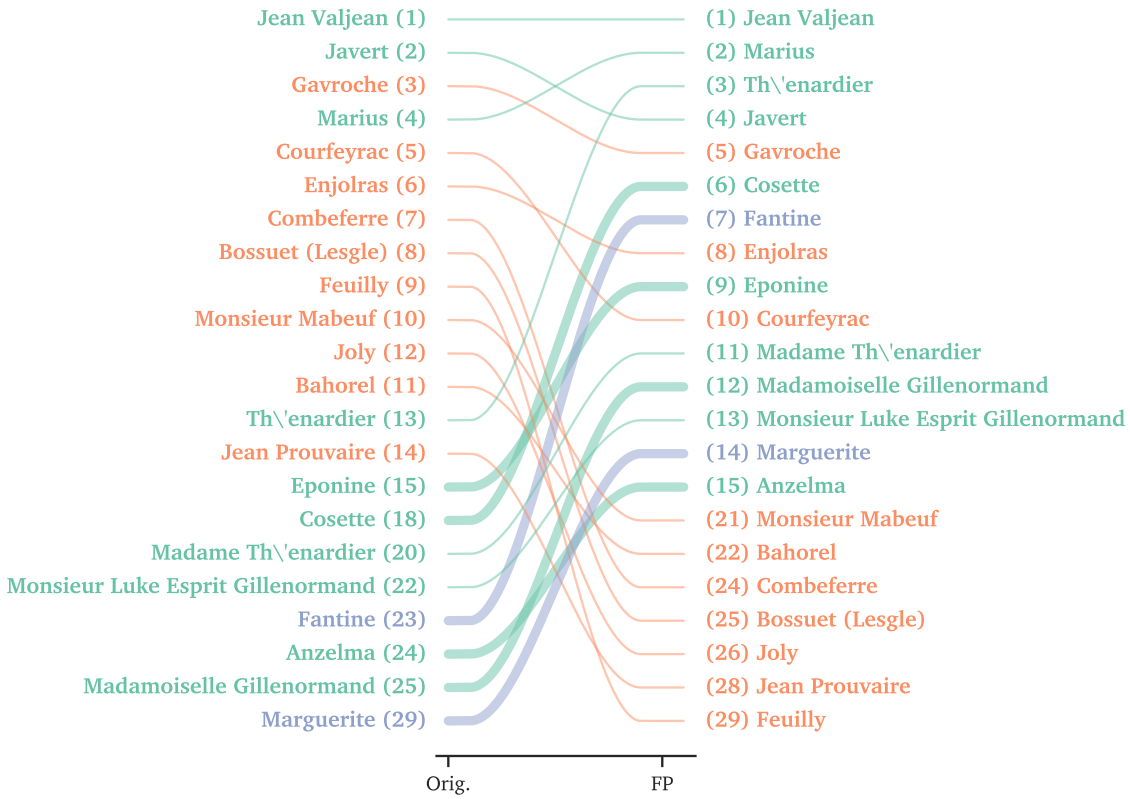


Figure 8.7

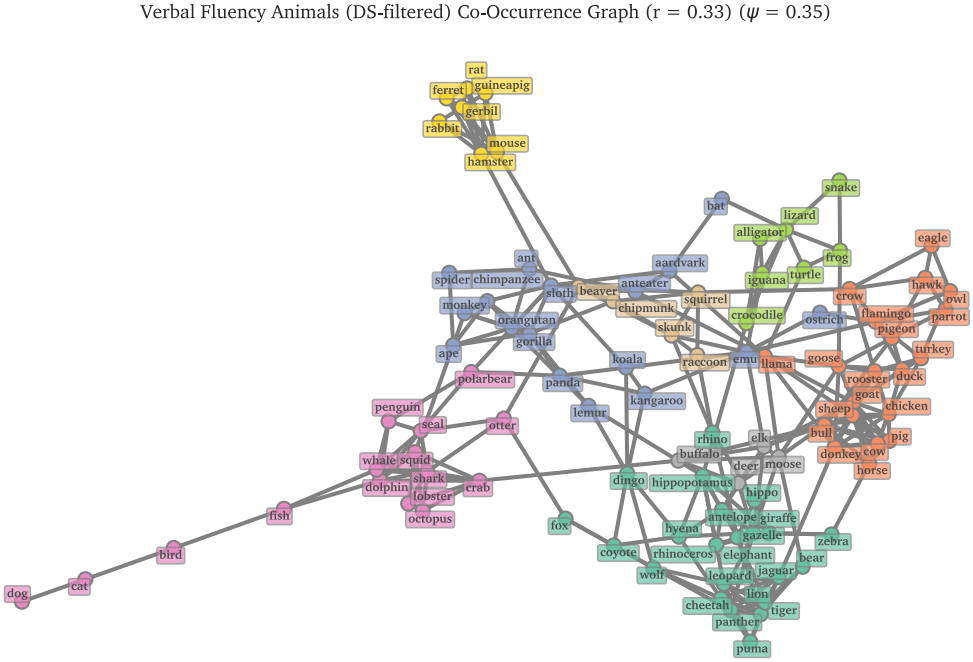


Figure 8.8

Verbal Fluency Animal Dependencies (Chow-Liu) Network (r = -0.13)(ψ = 1.00)

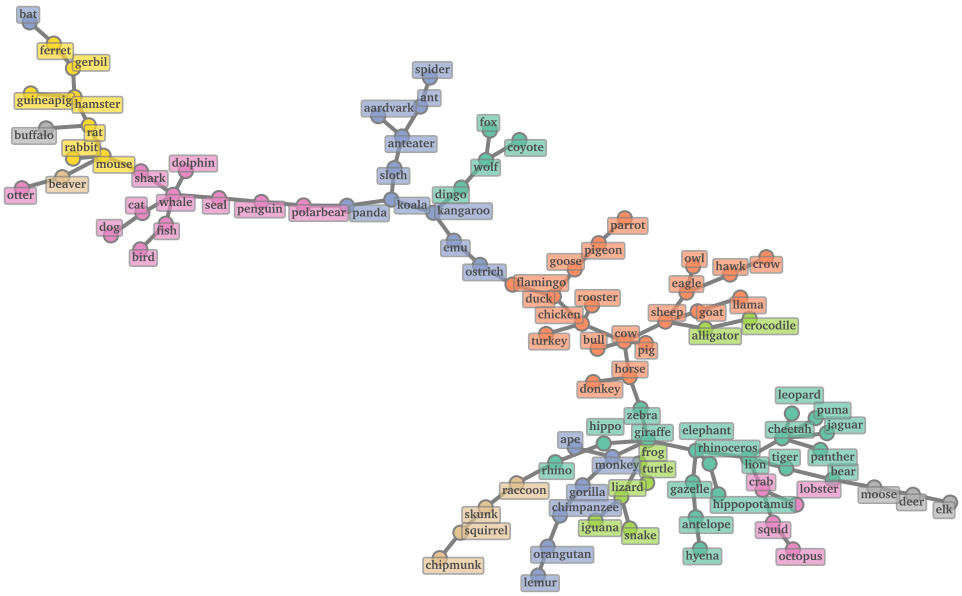


Figure 8.9

Verbal Fluency Animal Dependencies (GLASSO) Network (r = -0.02)(ψ = 0.45)

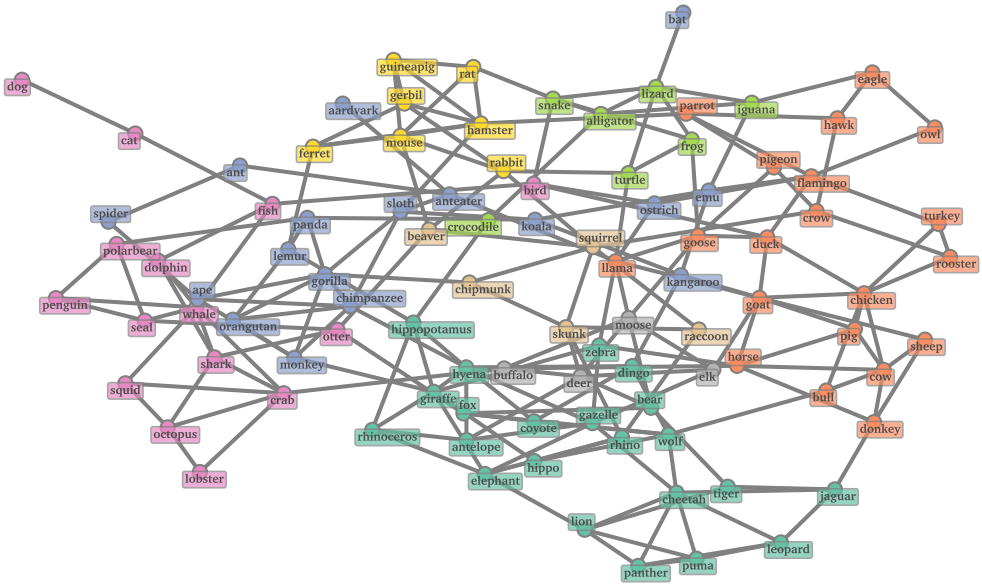


Figure 8.10

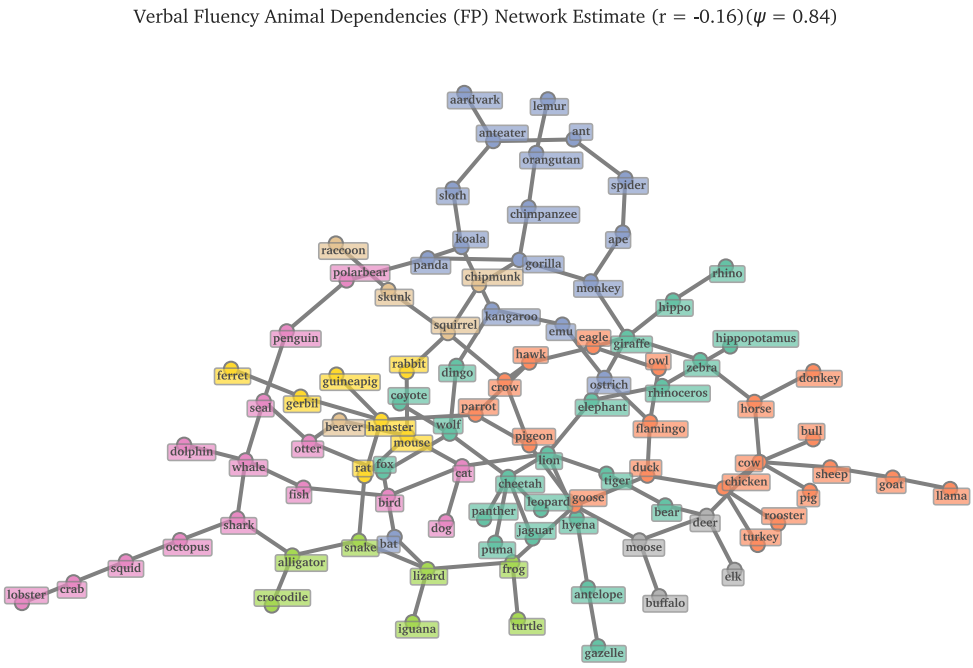


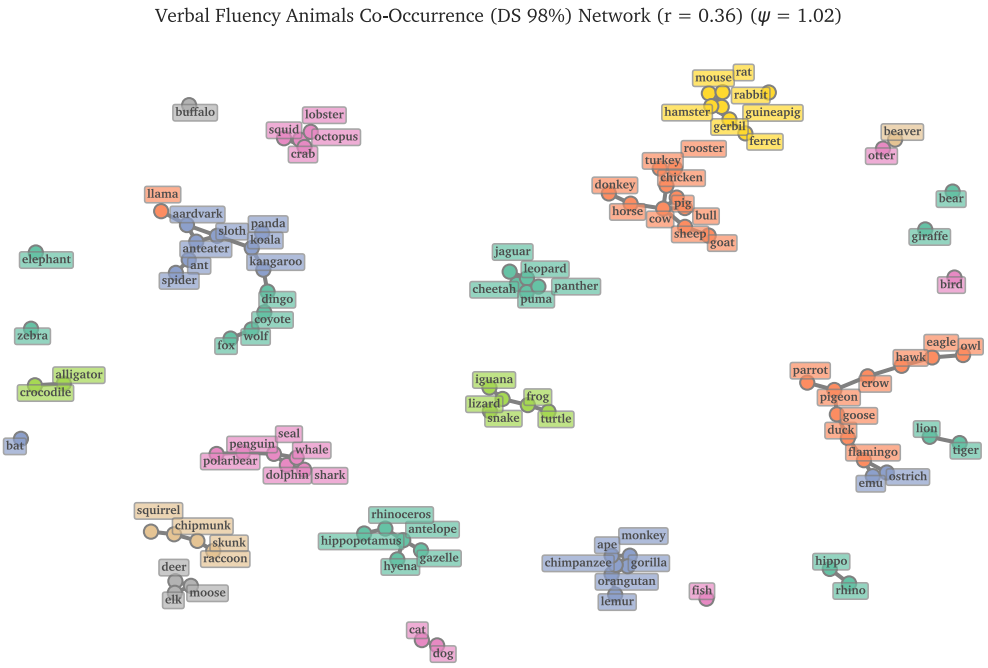
Figure 8.11: Comparison of backboning/dependency recovery methods tested vs. Forest Pursuit

8.3.3 Forest Pursuit as Preprocessing

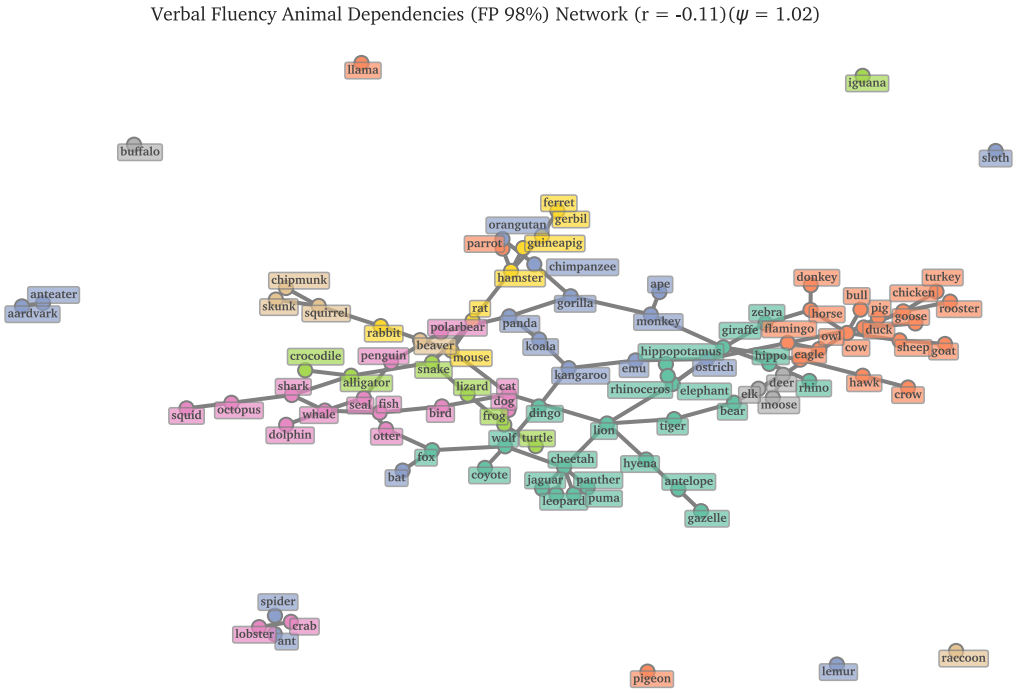
Differences in structural preservation with increased thresholding.

Retaining the top 2% of edges, co-occurrence retains local communities

at the cost of global structure.

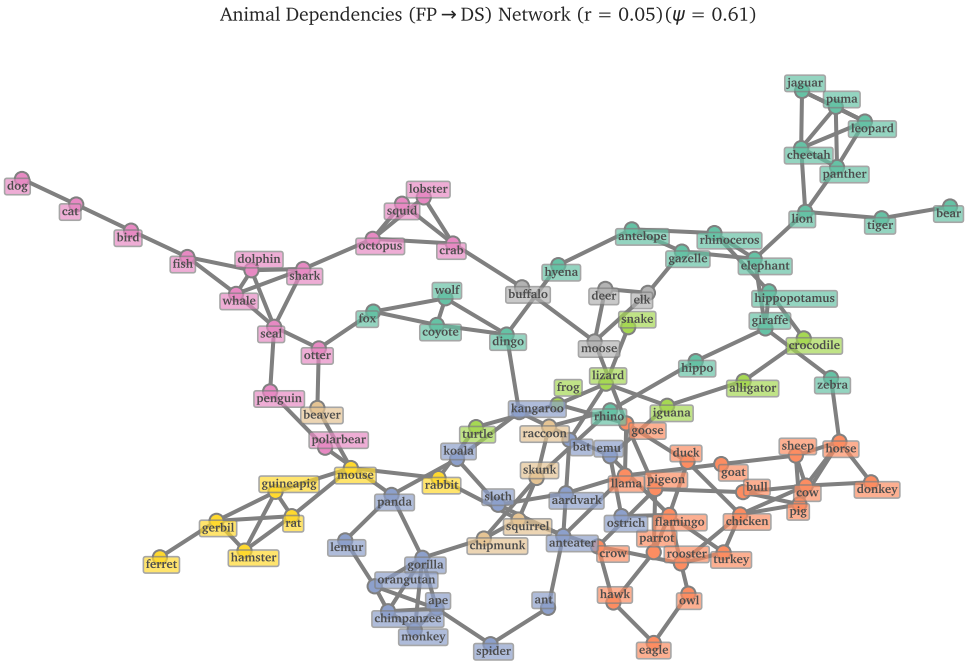


(a) co-occurrence methods will retain local communities at the cost of global structure

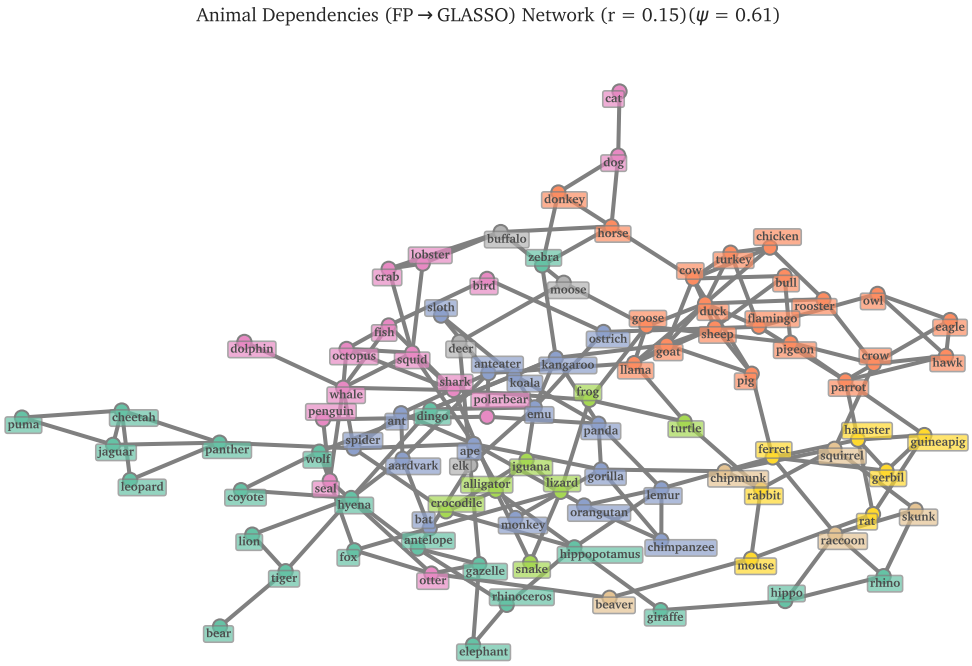


(b) dependency network drops rarer nodes from the preserved central structure at higher uncetainty cutoffs

Figure 8.12: When only retaining the top 2% of edge strengths, blah



(a) Islands of local structure remain (doubly-stochastic)



(b) Intact global structure with isolates

Figure 8.13: We might prefer to drop low-certainty/rare nodes from a preserved central structure.

Chapter 9: Recovery from Partial Orders

Like before, but with the added twist of *knowing* our nodes were activated with a particular partial order.

insert from [4, 5]

Bibliography

[1] L. Peel, T. P. Peixoto, and M. De Domenico, "Statistical inference links data and theory in network science," *Nature Communications*, vol. 13, no. 1, Nov. 2022, ISSN: 2041-1723. DOI: 10.1038/s41467-022-34267-9. [Online]. Available: https://www.nature.com/articles/s41467-022-34267-9.

[2] L. L. Duan and D. B. Dunson, "Bayesian spanning tree: Estimating the backbone of the dependence graph," arXiv, arXiv:2106.16120, Jun. 30, 2021, ZSCC: 0000001 type: article. DOI: 10.48550/arXiv.2106.16120. arXiv: 2106.16120.

[3] L. Torres, A. S. Blevins, D. Bassett, and T. Eliassi-Rad, "The why, how, and when of representations for complex systems," *SIAM Review*, vol. 63, no. 3, pp. 435–485, Jan. 2021, ISSN: 0036-1445. DOI: 10.1137/20M1355896. Accessed: Feb. 1, 2023.

[4] R. Sexton and M. Fuge, "Organizing tagged knowledge: Similarity measures and semantic fluency in structure mining," *Journal of Mechanical Design*, vol. 142, no. 3, Jan. 2020, ISSN: 1050-0472. DOI: 10.1115/1.4045686.

[5] R. Sexton and M. Fuge, "Using semantic fluency models improves network reconstruction accuracy of tacit engineering knowledge," in *Volume 2A: 45th*

Design Automation Conference, American Society of Mechanical Engineers, Aug. 2019. DOI: 10.1115/detc2019-98429.

[6] T. P. Peixoto, "Reconstructing networks with unknown and heterogeneous errors," *Physical Review X*, vol. 8, no. 4, p. 041 011, Oct. 16, 2018, ZSCC: 0000099. DOI: 10.1103/PhysRevX.8.041011. Accessed: Nov. 2, 2023.

[7] K. Avrachenkov, P. Chebotarev, and A. Mishenin, "Semi-supervised learning with regularized laplacian," *Optimization Methods and Software*, vol. 32, no. 2, pp. 222–236, Mar. 4, 2017, ZSCC: 0000013, ISSN: 1055-6788. DOI: 10.1080/10556788.2016.1193176. Accessed: May 8, 2023.

[8] O. Knill, "Counting rooted forests in a network," arXiv, arXiv:1307.3810, Jul. 18, 2013, ZSCC: 0000014 type: article. DOI: 10.48550/arXiv.1307.3810. arXiv: 1307.3810.

[9] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 5, no. 4, pp. 1–37, 2012, ISSN: 1556-472X. DOI: 10.1145/2086737.2086741.

[10] D. J. Wang, X. Shi, D. A. McFarland, and J. Leskovec, "Measurement error in network data: A re-classification," *Social Networks*, vol. 34, no. 4, pp. 396–409, Oct. 2012, ISSN: 0378-8733. DOI: 10.1016/j.socnet.2012.01.003.

[11] P. Chebotarev and E. Shamis, "The matrix-forest theorem and measuring relations in small social groups," arXiv, arXiv:math/0602070, Feb. 4, 2006,

ZSCC: 0000285 type: article. DOI: 10.48550/arXiv.math/0602070. arXiv: math/0602070 [math].

[12] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002, ISSN: 1091-6490. DOI: 10.1073/pnas.122653799.

[13] M. E. J. Newman, "Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality," *Physical Review E*, vol. 64, no. 1, p. 016 132, Jun. 2001, ISSN: 1095-3787. DOI: 10.1103/physreve.64.016132.

[14] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, Dec. 1977, ISSN: 2153-3806. DOI: 10.1086/jar.33.4.3629752.