

## 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
<b>1 Introduction</b>	<b>1</b>
1.1 Ambiguous Metrology . . . . .	1
1.2 Indirect Network Measurement . . . . .	3
1.3 Scope of this work . . . . .	7
<b>I A Practitioner’s Guide to Network Recovery</b>	<b>10</b>
<b>2 Metrology as matrices</b>	<b>11</b>
2.1 Observation and feature “spaces” . . . . .	12
2.2 Models & linear operators . . . . .	13
2.3 Measurement quantification & error . . . . .	16
2.3.1 Additive Smoothing . . . . .	16
2.3.2 Conditional Probabilities & Contingencies . . . . .	18
2.4 Distance vs. Incidence . . . . .	19
2.4.1 Kernels & distances . . . . .	20
2.4.2 Incidence structures & dependency . . . . .	22
2.4.3 Conclusion . . . . .	24
<b>3 Incidence through vector representation</b>	<b>25</b>
3.1 Graphs as incidence structures . . . . .	26
3.1.1 Embedding incidences in vector space . . . . .	28
3.1.2 Inner products on $B$ . . . . .	31
3.1.3 Edge Metrology, Edge Vectors . . . . .	32

3.2	Generalized node occurrence vectors . . . . .	33
3.2.1	Hyperedges as vectors of node occurrence . . . . .	34
3.2.2	Inner product on Hyperedges . . . . .	34
3.2.3	Combining Occurrence & Dependence . . . . .	34
<b>4</b>	<b>Roads to Network Recovery</b>	<b>35</b>
4.1	Organizing Recovery Methods . . . . .	35
4.2	Counting, Projection, and Co-Occurrence . . . . .	35
4.2.1	Cosine Similarity . . . . .	35
4.2.2	Hyperbolic Projection . . . . .	36
4.2.3	Transformations of Counts . . . . .	36
4.3	Resource and Information Flow . . . . .	36
4.3.1	Resource Projection . . . . .	36
4.3.2	Doubly-stochastic and eOT . . . . .	36
4.3.3	HSS . . . . .	36
4.4	Inverse Problems & PGM Structure . . . . .	36
4.4.1	Chow Liu . . . . .	36
4.4.2	Shrinkage & GLASSO . . . . .	37
4.4.3	stability selection . . . . .	37
4.5	A Path Forward . . . . .	38
4.5.1	Tracing Information Loss Paths . . . . .	38
4.5.2	Something New . . . . .	39

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

<b>5</b>	<b>Desire Paths and Spanning Forests</b>	<b>41</b>
5.1	The Gambit of the Inner Product . . . . .	42
5.1.1	Gambit of the Group . . . . .	42
5.1.2	Inner-Product Projections and “Clique Bias” . . . . .	43
5.2	Networks as Desire Path Density Estimates . . . . .	46
5.2.1	Subgraph Distributions . . . . .	47
5.2.2	Graph Unions as Desire Paths . . . . .	49
5.3	Tree & Forest Distributions . . . . .	49
5.3.1	Random Walk Activations . . . . .	50
5.3.2	Dependencies as Trees . . . . .	51
5.3.3	Matrix Tree and Forest Theorems . . . . .	51
5.4	Generative Model Specification . . . . .	52
<b>6</b>	<b>Forest Pursuit: Approximate Recovery in Near-linear Time</b>	<b>53</b>
6.1	Sparse Dictionary Learning . . . . .	53
6.1.1	Problem Specification . . . . .	53
6.1.2	Matching Pursuit . . . . .	53
6.1.3	Space of Spanning Forests . . . . .	53

6.2	Forest Pursuit: Approximate Recovery in Near-linear Time . . . . .	53
6.2.1	Uncertainty Estimation . . . . .	54
6.2.2	Approximate Complexity . . . . .	54
6.3	Simulation Study . . . . .	54
6.3.1	Method . . . . .	54
6.3.2	Results - Scoring . . . . .	54
6.3.3	Results - Performance . . . . .	57
6.4	Discussion . . . . .	57
6.4.1	Interaction Probability . . . . .	57
<b>7</b>	<b>LFA: Latent Forest Allocation</b>	<b>58</b>
7.1	Radom Spanning Trees . . . . .	58
7.2	Bayesian Estimation by Gibbs Sampling . . . . .	58
7.3	Simulation Study . . . . .	58
7.3.1	Score Improvement . . . . .	58
7.3.2	Odds of Individual Edge Improvement . . . . .	58
<b>III</b>	<b>Applications &amp; Extentions</b>	<b>60</b>
<b>8</b>	<b>Qualitative Application of Relationship Recovery</b>	<b>61</b>
8.1	Network Science Collaboration Network . . . . .	61
8.2	Les Miserables Character Network . . . . .	62
8.2.1	Backboning . . . . .	62
8.2.2	Character Importance Estimation . . . . .	62
<b>9</b>	<b>Recovery from Partial Orders</b>	<b>66</b>
9.1	Technical Language Processing . . . . .	66
9.2	Verbal Fluency Animal Network . . . . .	66
9.2.1	Edge Connective Efficiency and Diversity . . . . .	66
9.2.2	Thresholded Structure Preservation . . . . .	66
9.2.3	Forest Pursuit as Preprocessing . . . . .	66

## 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	Hyperedge Relation Observational Model	33
4.1	Relating Graphs and Hypergraph/bipartite structures as adjoint operators	38
5.1	Gram matrix as sum of observation outer products	44
5.2	Inner-product projections as sums of cliques illustrating “clique bias”.	45
5.3	Edge Measurements with true (tree) dependencies known	51
6.1	Comparison of MENDR recovery scores	54
6.2	Comparison of MENDR Recovery Scores by Graph Type	55
6.3	Partial Residuals (regression on $E[MCC]$ )	56
6.4	Runtime Scaling (Forest-Pursuit vs GLASSO)	56
6.5	Partial Residuals (regression on computation time)	57
7.1	Change in Expected MCC (EFM vs FP)	59
7.2	Logistic Regression Coef. (EFM - FP) vs. (Ground Truth)	59
8.1	134 Network scientists from [NEWMAN;BOCCALETI;SNEPPEN], connected by co-authorship	61
8.2	Max. likelihood tree dependency structure to explain co-authorships	62
8.3	Forest Pursuit estimate of NetSci collaborator dependency relationships	63
8.4	.....	63
8.5	.....	64
8.6	.....	64
8.7	.....	65
9.1	.....	67
9.2	.....	67
9.3	.....	68

9.4 Comparison of backboning/dependency recovery methods tested vs. Forest Pursuit . . . . .	68
9.5 When only retaining the top 2% of edge strengths, blah . . . . .	69
9.6 We might prefer to drop low-certainty/rare nodes from a preserved central structure. . . . .	70

# 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. [18] 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 [19], or simply ignoring measurement error in network reconstruction entirely[11].

## 1.1 Ambiguous Metrology

Networks in the “wild” rarely exist of and by themselves. 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*[27], 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.[\[24\]](#)

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.

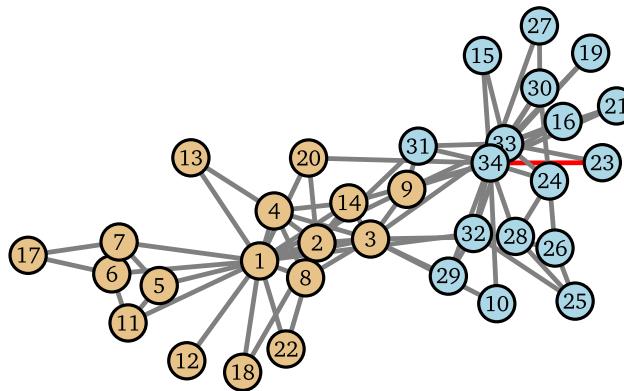


Figure 1.1: Zachary’s Karate Club, with ambiguously extant edge 78 highlighted.

*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-occurrences. 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.

$$\begin{aligned} \{ & \quad d \quad h \quad e \quad \} = x_1 \\ \{ & g \quad c \quad e \quad h \quad \} = x_2 \\ \{ & f \quad e \quad a \quad h \quad \} = x_3 \\ \{ & i \quad j \quad f \quad b \quad \} = x_4 \end{aligned}$$

Figure 1.2

These are naturally represented as “bipartite” networks, having separate entities 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. [25] 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”.

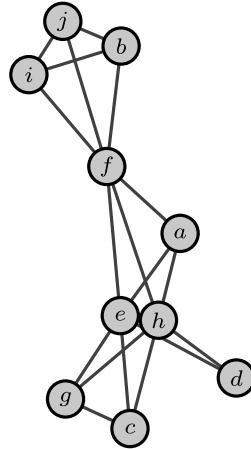


Figure 1.3

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 interaction between colleagues, perhaps 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 strengthened. 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

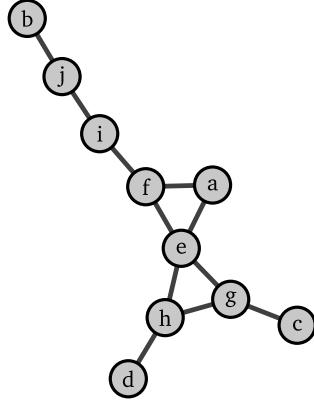


Figure 1.4: graph of mutual collaboration relationships i.e. the “ground truth” social network

goes wrong when we use co-occurrence/activation data to estimate the dependency network, especially when we wish to use it for metrics like centrality, shortest path distances, and community belonging?

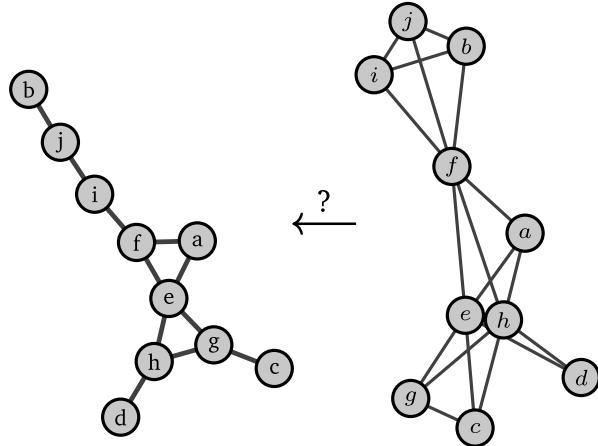


Figure 1.5: Recovering underlying dependency networks from node-cooccurrences.

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 tries to find a subset of edges in this hairball that still captures its core topology in a way that’s easier to visualize.[15, 21] 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, by framing network analysis as a problem of *inference*, as suggested by Peel et al. [1]. 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 either unspoken sources of epistemic and aleatory error, or high computation costs (or both). To begin addressing 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) is critical, 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 dependency 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 parallelizable 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 tech-

nical 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.  $x \in \mathbb{S} \in \{\mathbb{R}, \mathbb{Z}, \mathbb{N}, \dots\}$ . 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  $X : \mathbb{S}^{m \times n}$ .<sup>1</sup>

---

<sup>1</sup>Not all observations are scalar, but they can become so. If individual measurements 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

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

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

$$x = X(i, j) \quad X : I \times J \rightarrow \mathbb{S} \quad (2.1)$$

i.e. the measured value of the  $j$ th independent variable from the  $i$ th observation.<sup>2</sup> 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

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

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*):

$$\mathbf{x}'_j = X(\cdot, j), \quad \mathbf{x}' : I \rightarrow \mathbb{S}$$

---

<sup>2</sup>This notation is adapted from the sparse linear algebraic treatment of graphs in Kepner and Gilbert [20] and Kepner et al. [14].

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.<sup>3</sup>

To illustrate this formalism in a relevant domain, let’s take another look at co-citation networks. For  $m$  papers we might be aware of  $n$  total 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  $i^{\text{th}}$  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  $\mathbf{x}'_i : \mathbb{B}^{1 \times m}$ .

## 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

---

<sup>3</sup>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.

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 denoising 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<sup>4</sup>, and applying it to a model with  $l$  parameters is called the *forward map*:

$$\mathbf{x} = F\mathbf{p} \quad F : \mathbb{R}^l \rightarrow \mathbb{R}^n$$

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*:

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

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*:  $\hat{\mathbf{p}} = \min_{\mathbf{p}} F\mathbf{p} - \mathbf{x}$ .

---

<sup>4</sup>in the finite-dimensional case

Setting the gradient to 0 yields 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 variable, like citation count or a specific 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

In binary data, such as what we have been considering, it is common to model observables as so-called “Bernoulli trials”: events with two possible outcomes (on, off; yes, no; true, false), and one outcome has probability  $p$ . These can be thought of as weighted coin-flips: “heads” with probability  $p$ , and “tails”  $1 - p$ . If  $k$  trials are performed (the “exposure”), we say the number of successes  $s$  (the “count”) is distributed as a binomial distribution  $s \sim \text{Bin}(p, k)$ . The empirical estimate for the success probability is  $\hat{p} = \frac{s}{k}$ .

Note that this naturally resembles marginal sums on our design matrix  $X$ , if we treat columns (or rows!) as an array of samples from independent Bernoulli trials:  $\hat{p}_j = \frac{\sum_{i \in I} X(i, j)}{m}$ . Many probability estimates involving repeated measurements of binary variables (not simply the row/column variables) have this sort of  $\frac{\text{count}}{\text{exposure}}$  structure, as will become useful in later sections.

However, if we are “measuring” a probability, we run into issues when we need to quantify our uncertainty about it. For instance, an event might be quite rare, but if in our specific sample we *never* see it, we still do not generally accept a probability of zero.

### 2.3.1 Additive Smoothing

One approach to dealing with this involves adding *pseudocounts* that smooth out our estimates for count/exposure, from which we get the name “additive smoothing”.

ing".[CITE?]

$$\hat{p} = \frac{s + \alpha}{k + 2\alpha}$$

Adding 1 success and 1 failure ( $\alpha = 1$ ) as pseudocounts to our observations is called *Laplace's Rule of Succession*, or simply “Laplace smoothing,”<sup>5</sup> while adding  $\alpha = 0.5$  successes and failures is called using *Jeffrey’s Prior*. It’s so-called because this pseudocount turns out to be a special case of selecting a Bayesian prior on the binomial probability (a.k.a. a *Beta-Binomial* distribution)  $p \sim \text{Beta}(\alpha, \beta)$ , such that the posterior distribution after our success/failure counts is  $\text{Beta}(s + \alpha, k - s + \beta)$ , which has the mean:

$$E[p | s, k] = \frac{s + \alpha}{k + \alpha + \beta} \quad (2.2)$$

This exactly recovers additive smoothing with a Jeffrey’s prior for  $\alpha = \beta = 0.5$ .<sup>6</sup> This generalization allows us to be more flexible, and specify our prior expectations on counts or exposure with more precision. Such models provide both an estimate of the aleatory uncertainty (via the posterior distribution), and a form of “shrinkage” that prevents sampling noise from unduly affecting parameter estimates (via the prior distribution). Despite being a simple foundation, this treatment of “counts” and “exposure” can be built upon in many ways.

---

<sup>5</sup>derived when Laplace desired estimates of probability for unobserved phenomena, such as the sun (not) rising tomorrow.

<sup>6</sup>A useful comparison of the two priors (1, 0.5) is to ask, given all of the trials we have seen so far, whether we believe we are near the “end” or “middle” of an average run of trials. For  $\alpha = 1$ , we believe nearly all evidence has been collected, but for  $\alpha = 0.5$ , only half of expected evidence has been observed.

### 2.3.2 Conditional Probabilities & Contingencies

In dependency/structure recovery, since our goal involves estimating (at least) pairwise relationships, the independence assumption required to estimate node occurrences as Beta-Binomial is clearly violated.<sup>7</sup>

However, it's common to estimate how similar two random variables  $A, B$  are, e.g. if samples of each correspond to columns of binary  $X$ . For instance, the joint probabilities  $P(A \cap B)$  answer "how often does A happen with B, out of all data?"

Conditional probabilities  $P(A | B) = \frac{P(A \cap B)}{P(B)}$  measure how often A occurs given B happened. Once again, we can estimate the base probabilities  $P(A)$  and  $P(B)$  with methods like Equation 2.2 for each marginal sums  $X(\cdot, A)$  or  $X(\cdot, B)$ , but the joint and conditional probabilities can instead be estimated using matrix multiplication using the Gram matrix, discussed below. It encodes pair-wise co-occurrence counts, such that  $G(i, i') : \mathbb{Z}^{n \times n}$  has the co-occurrence count for node  $i$  with  $i'$ .

The co-occurrence probability  $P(A \cap B)$  for each pair can also be approximated with the beta-binomial scheme mentioned above, but care must be taken not to confuse this with the edge strength connecting two nodes. First, nodes that rarely activate (low node probability) may nonetheless reliably connect to others when they do occur (high edge probability). In fact, without direct observation of edges, we are not able to estimate their count, or their exposure, which can be a source of systemic error from *epistemic uncertainty*. We don't know when edges are used,

---

<sup>7</sup>In fact, a recent method from [16] models probabilistic binary observations, *with dependencies*, by generalizing the mechanics overviewed here to a fully multivariate Bernoulli distribution, capable of including 3rd- and higher-order interactions, not just pairwise.

directly, and we also don't have a reliable way to estimate the opportunities each edge had to activate (their exposure), either. This is especially true when we wish to know whether an edge even *can* be traversed, i.e. the edge *support*. Support, as used in this sense, is the set of inputs for which we expect a non-zero output. Intuitively, this idea captures the sense that we might care more about *whether* an edge/dependency exists, not *how important* it is. For that, we have to re-assess our simple model: even if we could count the number of times an edge might have been traversed, how do we estimate the opportunities it had to be available for traversal (it's "exposure")?

Assuming this kind of epistemic uncertainty can be adequately addressed through modeling—attempts at which will be discussed in more detail in Chapter 4—conditional probability/contingency tables will again be useful for validation. When comparing estimated edge probability to some known "true" edge existence (if we have that), we can count the number of correct predictions, as well as type I (false positive) and type II (false negative) errors. We can do this at *every probability/weight threshold value*, as well, and we will return to ways to aggregate all of these values into useful scoring metrics in Section 6.3.

## 2.4 Distance vs. Incidence

As we have already seen, operations from linear algebra make many counting and combinatoric tasks easier, while unifying disparate concepts to a common set of mechanics. In addition to having a map from integer indices to sets of interest, these design matrices/vectors are implicitly assumed to have entries that exist in a field

$F = (\mathbb{S}, \oplus, \otimes)$ . equipped with operators analogous to addition ( $\oplus$ ) and multiplication ( $\otimes$ ).<sup>8</sup> With this, we are able to define generalized inner products that take pairs vectors in a vector space  $\mathbf{x} \in V$ , such that  $\langle \cdot, \cdot \rangle_F : \mathbb{S}^n \times \mathbb{S}^n \rightarrow \mathbb{S}$ .

$$\langle \mathbf{x}_a, \mathbf{x}_b \rangle_F = \bigoplus_{j=1}^n \mathbf{x}_a(j) \otimes \mathbf{x}_b(j)$$

We can use this to perform “contractions” along any matching dimensions of matrices as well, since the sum index is well-defined.

$$X \in \mathbb{S}^{m \times n} \quad Y \in \mathbb{S}^{n \times m}$$

$$Z(i, j) = X \oplus, \otimes Y = \bigoplus_{j=1}^n X(i, j) \otimes Y(j, k) = XY$$

For ease-of-use, we will assume the standard field for any given set  $(\mathbb{S}, +, \times)$  if not specified otherwise, which recovers standard inner products  $\langle \cdot, \cdot \rangle$ . However, Kepner et al. [14] illustrates the usefulness of various fields (or semirings). They allow linear-algebraic representation of many graph operations, such as shortest paths through inner products over  $(\mathbb{R} \cup -\inf, \min, +)$ . This works because discrete/boolean edge weights will not accumulate extra strength beyond 1 under contraction over observations.

#### 2.4.1 Kernels & distances

As alluded to in the previous section, co-occurrence have a deep connection to a Gram matrix, which is a matrix of all pairwise inner products over a set of vectors.

---

<sup>8</sup>Or, more generally, a semiring if inverse operations for  $\oplus, \otimes$  don't exist.

$$X^T X = G(j, j') = \langle \mathbf{x}'_j, \mathbf{x}'_{j'} \rangle = \sum_{i=1}^m X(i, j) X(i, j') \quad (2.3)$$

Matrices that can be decomposed into another matrix and its transpose are symmetric, and positive semidefinite (PSD), making every gram matrix PSD. They are directly related to (squared) euclidean distances through the polarization identity[CITE?].<sup>9</sup>

$$d^2(j, j') = \|\mathbf{x}'_j - \mathbf{x}'_{j'}\|^2 = G(j, j) - 2G(j, j') + G(j', j') \quad (2.4)$$

In our example from before, the gram matrix will have entries showing the number of papers shared by two authors (or total papers by each, on the diagonal). This is because an inner product between two author (column) vectors will add 1 for each paper in the sum only if it has both authors in common. This is called a *bipartite projection*[CITE] into the authors “mode”, and is illustrated visually in Figure 1.3.

Due to [CITE Shoenberg/Mercer], we can generalize Equation 2.4 such that *any* function “kernel” function  $\kappa(x, y)$  that creates PSD matrix  $K(j, j') \in \mathbb{S}^{n \times n}$ . It says that such a PSD matrix can always be decomposed into a form  $K = R^T R$  for any matrix  $R(i, j) \in \mathbb{S}^{m \times n}$ , thus letting us use the polarization identity to create arbitrary

---

**Important:** these definitions are all using the  $\mathbf{x}'$  notation to indicate that these measurements are almost exclusively being done in the *data space*, i.e. on column vectors. While most definitions work on distances in terms of the measurements/objects/data, for *inverse problems* (like network recovery, structure learning, etc.) they are more often applied in terms of the features (here, the nodes. This can be seen in statistics as well, where covariance and correlation matrices (which are related to the gram and distance matrix definitions above), are defined as relationships between features/dimensions, not individual samples.

distance metrics. on  $\mathbb{S}^n$  [9]<sup>10</sup>

$$d_K(j, j') = \frac{1}{2} (K(j, j) + K(j', j')) - K(j, j') \quad (2.5)$$

This ability to create valid distance measures from arbitrary kernel functions is the core of a vast area of machine learning and statistics that employs the so-called *kernel trick*. [CITE?] Different kernels yield different properties useful for distinguishing points having specific properties. One class of kernels are normalized to the range  $[0, 1]$ , such that we ensure that equality along any one dimension is given a weight of  $\tilde{K}(j, j) = 1$ . Such a kernel matrix can be derived from any other kernel, and is often combined with a logarithmic similarity measure  $\tilde{\kappa}(x, y) = \log s(x, y)$ .

$$\tilde{K}(j, j') = \frac{K(j, j')}{\sqrt{K(j, j)^2 K(j', j')^2}} \quad (2.6)$$

$$d_{\tilde{K}}(j, j') = -\log \tilde{K}(j, j')$$

Since this is equivalent to applying Equation 2.5 to  $s$  directly. This normalization should be familiar as the way cosine similarities and correlation matrices are made as well (also having 1s along their diagonal), and illustrates how non-metric similarities can be potentially made into (pseudo)metrics.

#### 2.4.2 Incidence structures & dependency

Rather than how “close” or “far” to points are in vector space, which is described with the kernels and distances above, whether something “touches”—or, is incident

---

<sup>10</sup>Distance metric, here, means that  $d(x, y)$  satisfies the triangle inequality for all  $x, y$ .

to—something else is usually described abstractly as an *incidence structure*. This is an abstract way to describe how things “touch”, such as when a set of points lie on a line/plane, or nodes touch an edge. We say an incidence structure is a triple of sets called (for historical reasons) *points*  $\mathcal{P}$ , *lines*  $\mathcal{L}$ , and *flags*  $\mathcal{I}$ .[\[22\]](#)

$$(\mathcal{P}, \mathcal{L}, \mathcal{I}) \quad \mathcal{I} \subseteq \mathcal{P} \times \mathcal{L}$$

Representing these as matrices will be further explored in Section [3.1](#). But, the discrete nature of these incidence sets makes it clear that estimating the size and elements of  $\mathcal{I}$ , is a very different question from estimating the similarity/distance between two entities in a vector space.

In statistics, such discrete structures usually arise when we are concerned with direct dependence from indirect. Take as an example a set of masses connected together by springs. If we shake one mass, all masses will also shake some amount, depending on the spring constants of the springs each mass is “transmitted” force through, and the losses due to friction or air resistance. While the amount of movement over time depends on how “close” in this spring network two masses are, the movement itself can only be transmitted through springs that the masses are *incident* to. Movement could be modeled through similarity/distance measurements like correlation, since none of the masses are independent (all move when any do), but incidence in terms of spring force transmission is modeled in terms of conditional (in)dependence. If we hold all but two masses still, and moving one doesn’t move the other, then we know they are conditionally independent: no spring connects them!

This idea gets formalized as probabilistic graphical models, which are networks that define “incidence” between two variables as conditional dependence. Letting the random variables in the column-space of  $X$  be  $A, B$  and the remaining columns be  $C = X(\cdot, J \setminus \{A, B\})$ , then

$$P(A \cap B | C) = P(A | C)P(B | C) \implies (A \perp B | C) \implies (A, B) \notin \mathcal{I} \quad (2.7)$$

for a set of incidences  $\mathcal{I}$  defining a PGM that was sampled as  $X$ .

[SPRING GRAPHIC?]

#### 2.4.3 Conclusion

## Chapter 3: Incidence through vector representation

To provide a sufficiently rigorous foundation for network recovery from binary occurrence, we will need a rigorous way to represent networks and occurrences that lends itself to building structured ways both connect to each other. We build on the incidence structure and matrix product formalism from the previous chapter, introducing various ways to build graphs as incidence structures that have direct representations as matrices. This can be extended to representing occurrences as matrices of *hyperedge vectors*. This view allows us to interpret different representations of graphs (or hypergraphs) as connected by simple matrix operations.

Traditionally[3, 14], we might say a graph on nodes (or, “vertices”)  $v \in V = \{1, \dots, n\}$  and “edges”  $E$  is a tuple:

$$G = (V, E) \quad \text{s.t.} \quad E \subseteq V \times V$$

In incidence geometry terms, this would be similar to making two duplicate sets of the same nodes, and defining a graph as the set of incidences between nodes. The *adjacency matrix*  $A$  of  $G$ , degree matrix  $D$ , and graph/discrete Laplacian  $L$  are then

defined as:<sup>1</sup>

$$A(u, v) = \mathbf{1}_E((u, v)) \quad A : V \times V \rightarrow \mathbb{B}$$

$$D(u, v) = \text{diag}(\sum_V A(u, v)) \quad D : V \times V \rightarrow \mathbb{N}$$

$$L(u, v) = D(u, v) - A(u, v) \quad L : V \times V \rightarrow \mathbb{Z}$$

However, if edges and their recovery is so important to us, defining them explicitly as nodes-node incidences can be problematic when we wish to estimate edge existence (or not), given noisy pairs of node co-occurrences. Additionally, we have to be very careful to distinguish whether our graph is *(un)directed, weighted, simple*, etc., and hope that the edge set has been filtered to a subset of  $N \times N$  for each case. Instead, we propose a less ambiguous framework for vectorizing graphs, based on their underlying incidence structure.

### 3.1 Graphs as incidence structures

Instead, we give edges their own set of identifiers,  $e \in E = \{1, \dots, \omega\}$ . Now, define graphs as incidence structures between nodes and edges such that every edge is incident to either two nodes, or none:

$$G = (V, E, \mathcal{I}) \quad s.t. \quad \mathcal{I} \subseteq E \times V \tag{3.1}$$

Variations on graphs can often be conveniently defined as constraints on  $\mathcal{I}$ :

- Self loops can be prohibited by only allowing unique flags for a given relation<sup>2</sup>

---

<sup>1</sup>The *indicator function*  $\mathbf{1}_A(x)$  is 1 for values of  $x$  in the set  $A$ , and 0 otherwise.

<sup>2</sup>never two flags with the same pair, i.e.  $\mathcal{I}$  is a set, not a multiset.

- Multigraphs are similarly described by whether we allow pairs of vertices to appear with more than one edge<sup>3</sup>

Together, these constraints define “simple” graphs. Similarly, we can equip Equation 3.1 with a function  $B$  that allows  $\mathcal{I}$  to encode information about the specific kinds of incidence relations under discussion. We give  $B$  a range of possible flag values  $S$ :

$$G = (V, E, \mathcal{I}, B) \quad s.t. \quad \mathcal{I} \subseteq E \times V \quad B : \mathcal{I} \rightarrow S \quad (3.2)$$

- Undirected, unweighted graphs only need single elements: “incidence exists” i.e.  $S = \{1\}$
- Directed graphs can use two elements e.g. a “spin” for  $S = \{-1, 1\}$
- Weighted, undirected graphs are supported on positive scalars e.g.  $S = \mathbb{R}^+$
- Weighted, directed graphs are supported on any scalar e.g.  $S = \mathbb{R}_{\neq 0}$

If the “absence” of incidence needs to be modeled explicitly, a “null” stand-in (0, False) can be added to each  $S$ , which is useful for representing each structure as arrays for use with linear algebra (i.e.  $\{0, 1\}$ ,  $\{-1, 0, -1\}$ ,  $\mathbb{R}_0^+$ , and  $\mathbb{R}$ , respectively). By doing so, we can also place an exact limit on the maximum possible size of  $\omega = \|E\|$  in the simple graph case, and indicate edges by their unique ID, such that  $\mathcal{I} = E \times V$  is no longer a subset relation for  $E = \{1, \dots, \binom{n}{2}\}$ . Instead of existence in  $\mathcal{I}$ , we explicitly use incidence relation  $S$  to tell us whether each possible edge “exists” or not, simplifying our graph definition further<sup>4</sup>:

---

<sup>3</sup>in the set of flags containing nodes  $u$  or  $v$ , only one  $e$  may be incident to both of them.

<sup>4</sup>if we allow multi-edges , then

$$G = (V, E, B) \quad s.t. \quad B : E \times V \rightarrow S$$

$$v \in V = \{1, \dots, n\} \tag{3.3}$$

$$e \in E = \left\{ 1, \dots, \binom{n}{2} \right\}$$

The representation of  $B$  in Equation 3.3 bears a remarkable similarity to our original description of design matrices in Equation 2.1. In fact, as a matrix,  $B(e, v)$  is called the *incidence* matrix: every row has two non-zero entries, with every column containing a number of non-zero entries equal to that corresponding node's degree in  $G$ . Traditionally, we use an *oriented* incidence matrix, such that each row has exactly one positive (non-zero) value, and one negative (non-zero) value.<sup>5</sup> Even for undirected graphs, the selection of *which entry* is positive or negative is left to be ambiguous, since much of the math used later is symmetric w.r.t. direction<sup>6</sup>.

### 3.1.1 Embedding incidences in vector space

A formalism for graphs that starts with incidence matrices would benefit from a *canonical* oriented incidence matrix, rather than the family that is ambiguous w.r.t. edge orientation. To start, we can be more precise by selecting each row(edge) vector, and partitioning it into two: one for each non-zero column (node) that edge is incident to. Every incidence can be represented individually as standard basis vector  $e$  in the feature space of  $B$ , scaled by the corresponding value of  $S$ .

Let  $V_e$  be the set of nodes with (non-zero) incidence to edge  $e$ . Then the incidence

---

<sup>5</sup>In fact, this would make  $B \wedge^*(v, e)$  equivalent to a *graphical matroid*, another common formalism that generalizes graph-like structures to vector space representations.

<sup>6</sup>though not always!

vectors are:

$$\delta_e(v) = B(e, v)\mathbf{e}_v \quad \forall v \in V_e \quad (3.4)$$

And the (unoriented) incidence matrix vectors are recovered as sums over the incidence vectors for each edge:

$$\mathbf{b}_e = \sum_{v \in V_e} \delta_e(v) \quad (3.5)$$

A traditional approach might then define undirected graphs as equivalent, in some sense, to multidigraphs, where each edge is really two directed edges, in opposing directions. This does allow the matrix  $B$  to have the correct range for its entries in this formalism (the directed graph range,  $S = \{-1, 0, 1\}$ ), and the edge identity based on sums would hold. However, the resulting set of incidences would have twice the number of edges than our combinatoric limit for simple graphs, and prevent the more elegant definition of graph types through the set  $\mathbf{S}$ . Plus, it would necessitate averaging of weights over different edge ID's to arrive at a single undirected “edge weight”, and many other implementation details that make keeping track of specifics difficult for practitioners.

Instead, we would like a canonical oriented distance matrix, which can be derived from the vectorized incidences in the undirected range of  $B$  (the standard basis vectors). Without loss of generality, let  $u_e, v_e \in V_e$  be nodes such that  $u < v$ .<sup>7</sup> Using this, we can unambiguously define a *partition*  $B(e, \cdot) = B(e, u_e) + B(e, v_e)$  between incidences on  $e$ , along with a new derived incidence,  $B_o$ , which has oriented rows

---

<sup>7</sup>the inequality is strict because self-loops are not allowed.

like:

$$B_o(e, \cdot) = \mathbf{b}_e^o = \delta_e(u) - \delta_e(v)$$

In other words, while the unoriented incidence matrix is the “foundational” representation for graphs in our formalism, the (canonical) oriented one can be derived, even if negative incidence values are not in  $\mathbb{S}$ .<sup>8</sup>

But, now that we have removed the information on “which nodes an edge connects” from our definition of edges (since every edge is a scalar ID), how do we construct  $V_e$  without a circular dependency on  $B$  to find non-zero entries? Because of our unique identification of edges up to the combinatoric limit, we can still actually provide a unique ordering of the nodes in  $V_e$ , without searching over the entirety of  $B$ ’s domain. Using an identity from Angeletti et al. [8], we have a closed-form equation both to retrieve the IDs of nodes  $u, v$  (given an edge  $e$ ), and an edge  $e$  (given two nodes  $u, v$ ), for any simple graph with  $n$  vertices.

$$\begin{aligned} u_n(e) &= n - 2 - \left\lfloor \frac{\sqrt{-8e + 4n(n-1) - 7} - 1}{2} \right\rfloor \\ v_n(e) &= e + u_n(e) + 1 - \frac{1}{2} \left( n(n-1) + (n - u_n(e))^2 - n + u_n(e) \right) \\ e_n(u, v) &= \frac{1}{2} \left( n(n-1) - ((n-u)^2 - n + u) \right) + v - u - 1 \end{aligned} \tag{3.6}$$

Our ease-of-calculation lets us drop some of the excess notation and refer to our (un)oriented incidence matrices in terms of the incidences of each edge on their  $u$  or

---

<sup>8</sup>This works as long as we are in at least a ring, since semirings in general do not need to define additive inverse operations. In this case we would limit ourselves to the oriented incidence.

$v$ , directly.

$$B = B_u + B_e \quad B_o \equiv B_u - B_v$$

### 3.1.2 Inner products on $B$

With all of this background, the other representations of graphs can seen as derivations from the canonical incidence matrices. The Laplacian, which is usually introduced either in terms of adjacency/degree, or as the gram matrix for oriented edge vectors, is also related to the gram matrix between all pairs of incidences on  $(u, v)$ . The other identities are simply consequences of the polarization identity:

$$\begin{aligned} L &= B_o^T B_o \\ &= \|B_u - B_v\|^2 \\ &= 2\|B_u\|^2 + 2\|B_v\|^2 - \|B_u + B_v\|^2 \\ &= 2D - B^T B = D - A \end{aligned} \tag{3.7}$$

The Laplacian is often used in defining random-walks and markov chains, such that the degree of each node should be normalized to 1, which can be accomplished either by row- or column-normalizing it:  $L^{rw} = D^{-1}L$  or  $LD^{-1}$ , respectively. If this degree-normalization is desired without de-symmetrizing  $L$ , we can still perform an operation similar to normed kernels in Equation 2.6, called the symmetric normalized Laplacian.

$$\tilde{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = \frac{L(u, v)}{\sqrt{L(u, u)L(v, v)}} \tag{3.8}$$

### 3.1.3 Edge Metrology, Edge Vectors

Implicitly in the use of  $B$  with design matrix mechanics from the previous chapter is a treatment of edges as “observations” (the data space), and nodes as features. If an edge *is* an observation, then unfortunately we cannot really quantify uncertainty over repeated measurements of edges with the simple mechanics from Section 2.3.1 (because that edge *is* that corresponding observation, and IDs are not duplicated).

So far we have seen two ways of representing the entire graph in matrix form: Incidence matrix  $B$  (or  $B_o$ ), and the inner-product matrices derived from it ( $L, A$ ). Since we can recover node IDs from edge IDs by Equation 3.6, we can use a single vector to represent an entire graph structure by its edges alone. Then a data dimension for vectors in “edge space” can once again represent observations, with nodes implied by Equation 3.6. This is either done by contracting  $B$  along the nodes (columns) dimension, or by *unrolling* the upper triangle of  $L$  or  $A$  according to Equation 3.6.<sup>9</sup> If each vector represents a value of  $B$  associated with a corresponding edge, then  $m$  of these vectors would be equivalent to  $m$  observations of  $\binom{n}{2}$  edges on the same set of

---

<sup>9</sup>Equation 3.9 uses an averaging operation to accomplish the contraction, but any reduction over the two nodes shared by an edge would accomplish the same, especially since we rarely see separate values for edge weight per-node, the way incidences can.

$n$  nodes. Formally, for the  $i$ th observed structure on  $n$  nodes:

$$R(i, e) = \min(\{B_i(e, u_n(e)), B_i(e, v_n(e))\})$$

$$\text{s.t. } R : I \times E \rightarrow \mathbb{S}$$

(3.9)

$$i \in I = \{1, \dots, m\} \quad e \in E = \{1, \dots, \omega\}$$

$$n = \frac{1}{2}(1 + \sqrt{8\omega + 1})$$

This representation formalizes what practitioners call “edgelists” into a data structure that can unambiguously distinguish directed, undirected, and weighted graphs. In addition, it allows for repeated measurements of edges over the same set of nodes, while flexibly growing when new nodes arrive.<sup>10</sup>

### 3.2 Generalized node occurrence vectors

TODO

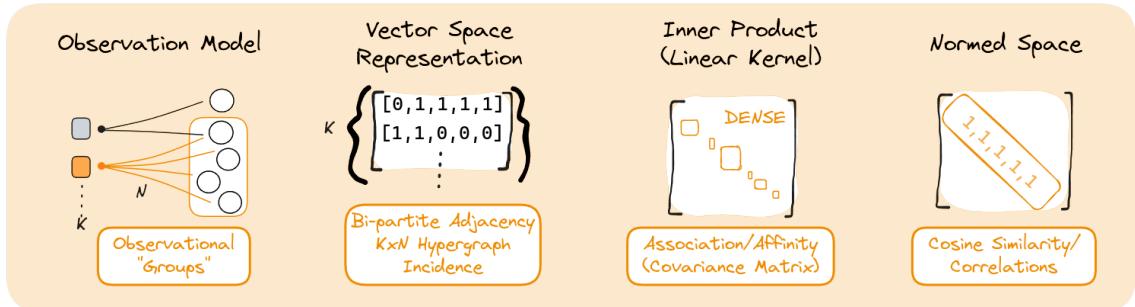


Figure 3.1: Hyperedge Relation Observational Model

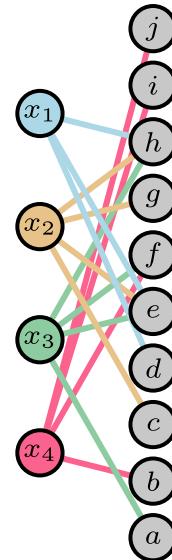
---

<sup>10</sup>For instance, say observations are stored as sparse entries via  $R$ , and a new node arrives. First, the participating nodes can be recovered in a vectorized manner through Equation 3.6. Then, a new node id increases  $n$ , followed by reassignment of the edge IDs with  $e_n(u, v)$ .

### 3.2.1 Hyperedges as vectors of node occurrence

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

(a)



(b) Bipartite representation of node “activation” data

### 3.2.2 Inner product on Hyperedges

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

Leads to correlation/covariance, etc.

### 3.2.3 Combining Occurrence & Dependence

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

## Chapter 4: Roads to Network Recovery

### 4.1 Organizing Recovery Methods

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

- Observing Nodes vs Edges
- Embeddings and Inner Products
- Preprocessing

### 4.2 Counting, Projection, and Co-Occurrence

Basic Co-occurrence, and friends. Gram is uncentered covariance.

All estimate some kind of rate

#### 4.2.1 Cosine Similarity

- Ochiai and uncentered correlation
- angles don't care about length
- geometric mean on cond prob, actually
- that means pseudocounts can make sense

#### 4.2.2 Hyperbolic Projection

- reweights by degree in original BP
- mention other ways

#### 4.2.3 Transformations of Counts

Come from contingency tables, like CS.

- odds-ratio
- Yule's Q,Y
- Mutual Info

### 4.3 Resource and Information Flow

#### Backboning

Intuition that the most important edges are on geodesics for many paths

##### 4.3.1 Resource Projection

##### 4.3.2 Doubly-stochastic and eOT

##### 4.3.3 HSS

### 4.4 Inverse Problems & PGM Structure

##### 4.4.1 Chow Liu

Global Tree structure from MI

#### 4.4.2 Shrinkage & GLASSO

#### 4.4.3 stability selection

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

## 4.5 A Path Forward

### 4.5.1 Tracing Information Loss Paths

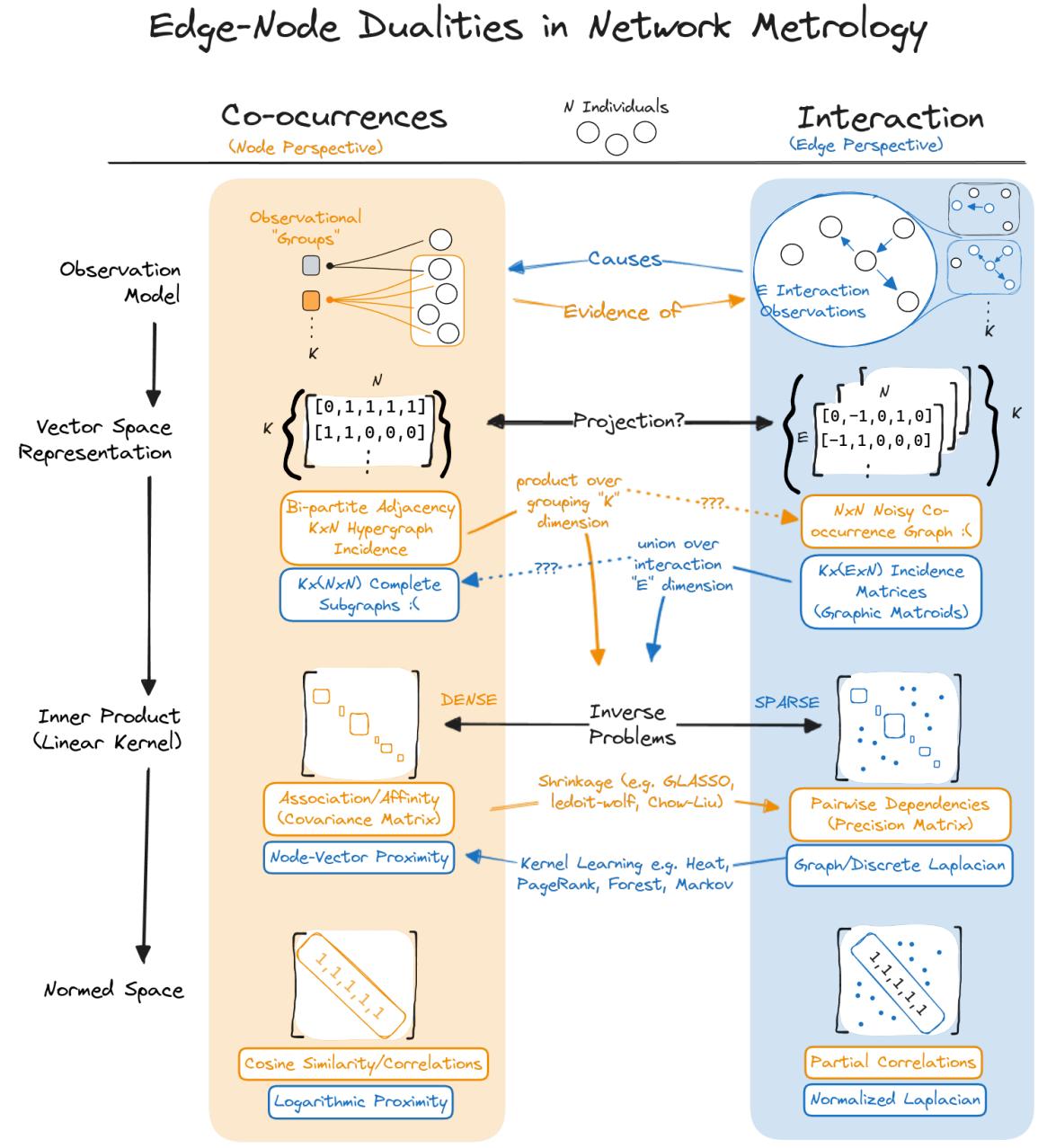


Figure 4.1: Relating Graphs and Hypergraph/bipartite structures as adjoint operators

## 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

### 4.5.2 Something New

Sorting algorithms... *none address all three!*

i.e. MOTIVATES FOREST PURSUIT

## Part II

Nonparametric Network Recovery With Random Spanning Forests

## Chapter 5: Desire Paths and 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 generated, presuming the existence of a dependency graph that *causes* activation patterns will give us useful ways to narrow down the generative specification.

First, we will investigate the common assumption that pairwise co-occurrences can serve as proxies for measuring relatedness, and how this “gambit of the group” is, in fact, a strong bias toward dense, clique-filled recovered networks. Because we wish to model our node activations as being *caused* by other nodes (that they depend on), we draw a connection to a class of models for *spreading*, or, *diffusive processes*. We outline how random-walks are related to these diffusive models of graph traversal, enabled by an investigation of the graph’s “regularized laplacian” from Avrachenkov et al. [12]. Then we use the implicit causal dependency tree structure of each observation, together with the Matrix Forest Theorem [17, 23] to more generally define our generative node activation model. This leads to a generative model for binary activation data as rooted random spanning forests on the underlying dependency graph.

## 5.1 The Gambit of the Inner Product

As we saw repeatedly in Chapter 4, networks are regularly assumed to arise from co-occurrences, whether directly as counts or weighted in some way. This assumption can be a significant a source of bias in the measurement of edges. *Why* a flat count of co-occurrence leads to “hairballs” and bias for dense clusters can be related to the use of inner products on node activation vectors.

### 5.1.1 Gambit of the Group

It seems reasonable, when interactions are unobserved, to assume some evidence for all possible interactions is implied by co-occurrence. However, similar to the use of uniform priors in other types of inference, if we don’t have a good reason to employ a fully-connected co-occurrence prior on interaction dependencies, we are adding systematic bias to our inference. Whether co-occurrence observations can be used to infer interaction networks directly was discussed at length in Whitehead and Dufault [26], where Whitehead and Dufault call this the *gambit of the group*.

So, consciously or unconsciously, many ethnologists studying social organization make what might be called the “gambit of the group”: they assume that animals which are clustered [...] are interacting with one another and then use membership of the same cluster [...] to define association.

This work was rediscovered in the context of measuring assortativity for social networks,<sup>1</sup> where the author of Fisher et al. [13] advises that “group-based methods”

---

<sup>1</sup>Assortativity is, roughly, the correlation between node degree and the degrees of its neighbors.

can introduce sampling bias to the calculation of assortativity, namely, systematic overestimation when the sample count is low.

The general problems with failing to specify a model of what “edges” actually *are* get analyzed in more depth in Peel et al. [1]. They include a warning not to naively use correlational measures with a threshold, since even simple 3-node systems will easily yield false positives edges. Still, it would be helpful for practitioners to have a more explicit mental model of *why* co-occurrence-based models yield systematic bias,

### 5.1.2 Inner-Product Projections and “Clique Bias”

Underlying correlation and co-occurrence models for edge strength is a reliance on inner products between node occurrence vectors. They all use gram matrices (or centered/scaled versions of them), which were brought up in Section 2.4. The matrix multiplication performed represents inner products between all pairs of feature vectors. For  $X(i,j) \in \mathbb{B}$ , these inner products sum together the times in each observation that two nodes were activated together.

However, another (equivalent) way to view matrix multiplication is as a sum of outer products

$$G(j,j') = X^T X = \sum_{i=1}^m X(i,j)X(i,j') = \sum_{i=1}^m \mathbf{x}_i \mathbf{x}_i^T$$

Those outer products of binary vectors create  $m \times m$  matrices that have a 1 in every  $i,j$  entry where nodes  $i,j$  both occurred, shown in Figure 5.1. Each outer product is effectively operating as a  $D_i + A_i$  with degrees normalized to 1. If the off-diagonals can be seen as adjacency matrices, they would strictly represent a clique on nodes

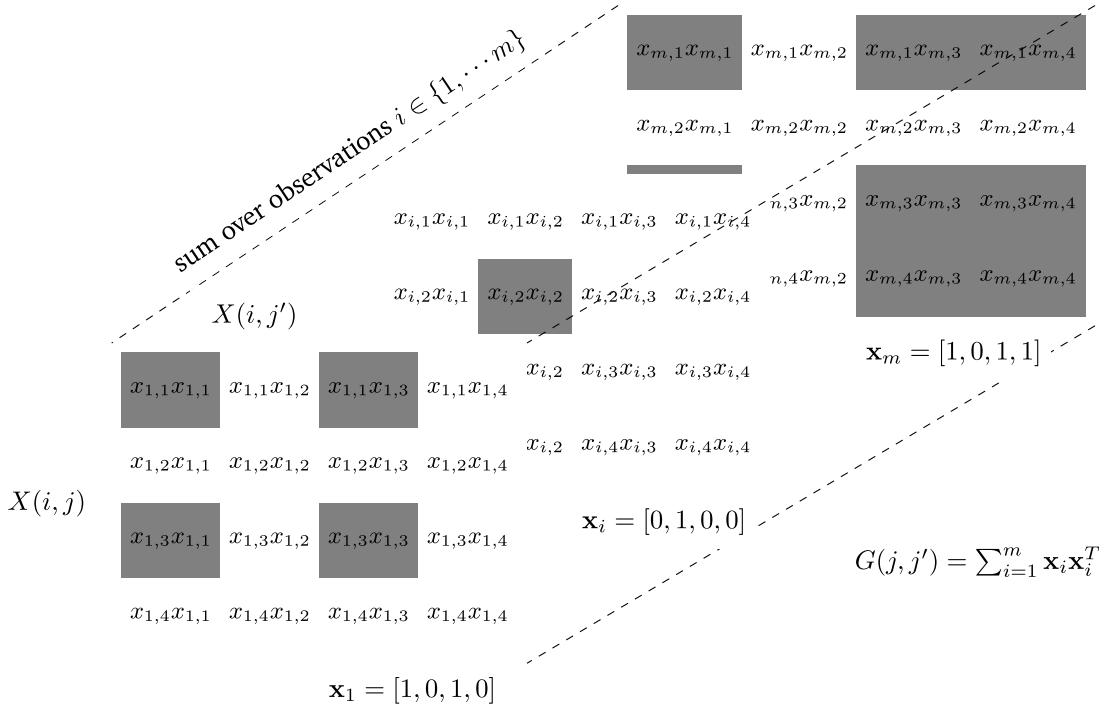


Figure 5.1: Gram matrix as sum of observation outer products

activated in the  $i$ th observation. In this sense, any method that involves transforming or re-weighting a gram matrix, is implicitly believing that the  $k$ th observation was a *complete graph*. This is illustrated in Figure 5.2.

For many modeling scenarios, this paradigm allows practitioners to make a more straight-forward intuition-check: do clique observations *make sense* here? When a list of authors for a paper is finished, does that imply that all authors mutually interacted with all others directly to equally arrive at the decision to publish? This would be similar to assuming the authors might simultaneously enter the same room, look at a number of others (who all look exclusively at each other, as well), and *all at once* decide to publish together. In our introduction, we described a more likely scenario we could expect from an observer on the ground: a researcher asks a colleague or two to collaborate, who might know a couple more with relevant expertise, and so

$$\begin{aligned}
 \{d, h, e\} &= x_1 \\
 \{g, c, e, h\} &= x_2 \\
 \{f, e, a, h\} &= x_3 \\
 \{i, j, f, b\} &= x_4
 \end{aligned}$$

(a)

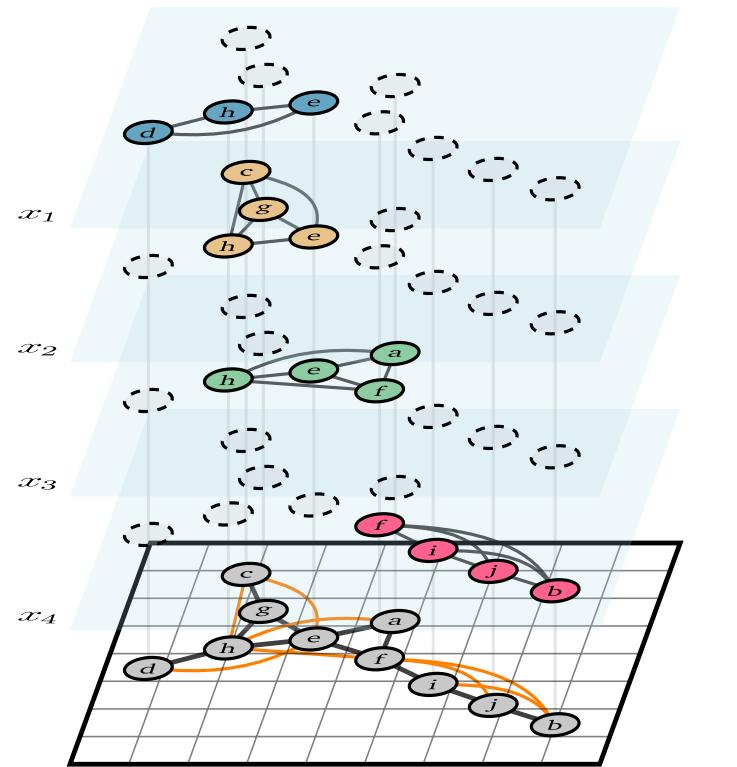


Figure 5.2: Inner-product projections as sums of cliques illustrating “clique bias”.

on.

## 5.2 Networks as Desire Path Density Estimates

Unfortunately, methods based on inner-product thresholding are still incredibly common, in no small part due to how *easy* it is to create them from occurrence data, regardless of this “clique-bias”. The ability to map an operation onto every observation, e.g. in parallel, and then reduce all the observations into an aggregate edge estimate is a highly desirable algorithmic trait. This may be a reason so many projection and backboning techniques attempt two re-weight (but retain) the same basic structure, time and again.

What we need is a way to maintain the ease-of-use of inner-product network creation:

- map an operation onto each observation
- reduce to an aggregate edge guess over all observations

but with a more domain-appropriate operator at the observation level.

Let’s start with replacements for the clique assumption. There are many non-clique classes of graphs we might believe local interactions occur on: path-graphs, trees, or any number of graphs that reflect the topology or mechanism of local interactions in our domain of interest. Authors have proposed classes of graphs that mirror human perception of set shapes [RELATIVE NEIGHBORHOOD]<sup>2</sup>, or graphs whose modeled

---

<sup>2</sup>e.g. for dependencies based on perception, such as human decision making tendencies, or causes based on color names.

dependencies are strictly planar [planar maximally filtered graphs]<sup>3</sup>. Alternatively, the interactions might be scale free, small-world, trees, or samples from stochastic block models.[CITE] In any case, these assumptions provide an explicit way to describe the set of *possible* interaction graphs we believe our individual observations are sampled from.

### 5.2.1 Subgraph Distributions

Let's use the notation from Equation 3.9, such that each observation of nodes  $\mathbf{x}_i$  is implicitly derived from a set of activated edges  $\mathbf{r}_i$ . To start, our current belief about what overall structure the whole network can take is  $G^* = (V, E, B^*)$ , where  $B^*$  might always return 1 to start out (the complete graph). From this, we construct a  $B_i$  for each observation. Each  $\mathbf{x}_i$  will induce a subgraph  $g_i = G^*[V_i]$ , where  $V_i = \{v \in \mathcal{V} | X(i, v) = 1\}$ , while our domain knowledge takes the form of a constraint on edges, which will induce a family of subgraphs again. The “family of subgraphs,” in general, can't directly use the edge constraint to create  $B_i(e, v) = \mathbf{1}_{V_i}(v)\mathbf{1}_{E_i}(e)$ , because edges  $E_i$  are not directly observed. Instead, we model them as inducing a family  $\mathcal{C}_i$  belonging to the relevant class of graphs  $\mathcal{C}$  on  $V$ , i.e.

$$\mathcal{C}_i = \{(V, E, B_i) \in \mathcal{C} | B_i(e, v) = \mathbf{1}_{V_i}(v)\mathbf{1}_{E_i}(e) \forall v \in V_i\}$$

$$E_i \in \{\mathcal{E} \subseteq \mathbf{P}(\mathbf{E}) | g_i[\mathcal{E}] \in \mathcal{C}\}$$

In other words, the edges that gave rise to the node activations in a given observation must belong to a graph that in turn belongs to our class.

---

<sup>3</sup>e.g. when interactions are limited to planar dependencies, like inferring ancient geographic borders.

Assuming we can define an associated measure for each  $\mu_i(c)$  on  $\mathcal{C}_i$ , we are able to define a probability distribution over subgraphs in the class.<sup>4</sup> Using notation similar to distributions over spanning trees in Zmigrod et al. [4]:

$$p_i(c) = \frac{\mu_{\mathcal{C}}(c | \mathcal{V})}{Z}$$

$$Z = \sum_{c \in \mathcal{C}^V} \mu_{\mathcal{C}}(c | \mathcal{V})$$

To further constrain the problem, let us assume that node activation is noiseless: any activated nodes were truly activated, and unactivated nodes were truly inactive (no false negative or false positive activations).<sup>5</sup> If a set of nodes  $V_i \subseteq V$  were observed in the  $i$ th observation, this assumption narrows the distribution to only the subgraphs  $c \in \mathcal{C}^{V_i}$ . This can be represented using the vectorization in Equation 3.9, due to the one-to-one correspondence established, so that

$$\mathbf{r}_i | \mathbf{x}_i \sim p_{\mathcal{C}}(c | V_i) \tag{5.1}$$

So we may not have an exact vector, but we have established a way to specify a family of edge vectors that could be responsible, along with a mechanism to sample them when a partition function is available or able to be approximated. Distributions over graphs are nontrivial, with spanning trees being a major exception. However,

---

<sup>4</sup>This is certainly not a trivial assumption, and might either be ill-defined or require techniques like the Gumbel trick[5] to approximate, unless the partition function  $Z$  has a closed form, or  $\mu$  is already a probability measure on some  $\sigma$ -algebra over  $\mathcal{C}$ . Closed-form  $Z$  values on  $\mathcal{C}$  are known for a handful of graph classes, such as trees.

<sup>5</sup>Hidden nodes (unobserved nodes beyond the  $n$ ) are outside the scope of this work, though could potentially be implied for certain structures e.g. when the metric is known to be tree-like. Sonthalia and Gilbert [7] use a greedy embedding that minimizes distortion to estimate the need for added *Steiner* nodes.

one way this might be accomplished is through random geometric graphs[CITE], or possibly geometric spanners like Urqharts[CITE] and Relative Neighborhood [CITE] graphs on a random “sprinkling” of points.

Also, the choice of a clique (implied by the inner product) is a trivial application of Equation 5.1, since that is equivalent to selecting the class of cliques on  $V_i$  nodes. This has only one element, making the complete graph the only possible selection, with probability 1.

### 5.2.2 Graph Unions as Desire Paths

- given a sample/MLE/MAP from each row’s distribution
- Use beta-bernoulli to estimate each edge’s parameters, pairwise indep.
- graphs don’t really exist, so we won’t act like they do
- approximate a network of paved sidewalks as a network of desire paths
- the desirepaths that are “ahead” in foot traffic are the ones we expect to be “paved” irl.
- once traveled, more likely to be traveled again -> arcsine distribution

## 5.3 Tree & Forest Distributions

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.

### 5.3.1 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

### 5.3.2 Dependencies as Trees

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

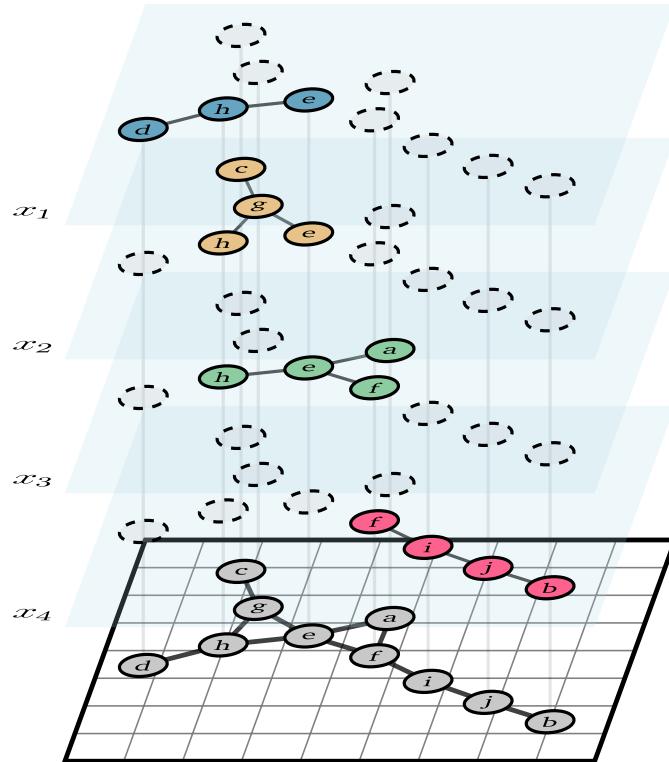


Figure 5.3: Edge Measurements with true (tree) dependencies known

[GRAPHIC 1 - my data]

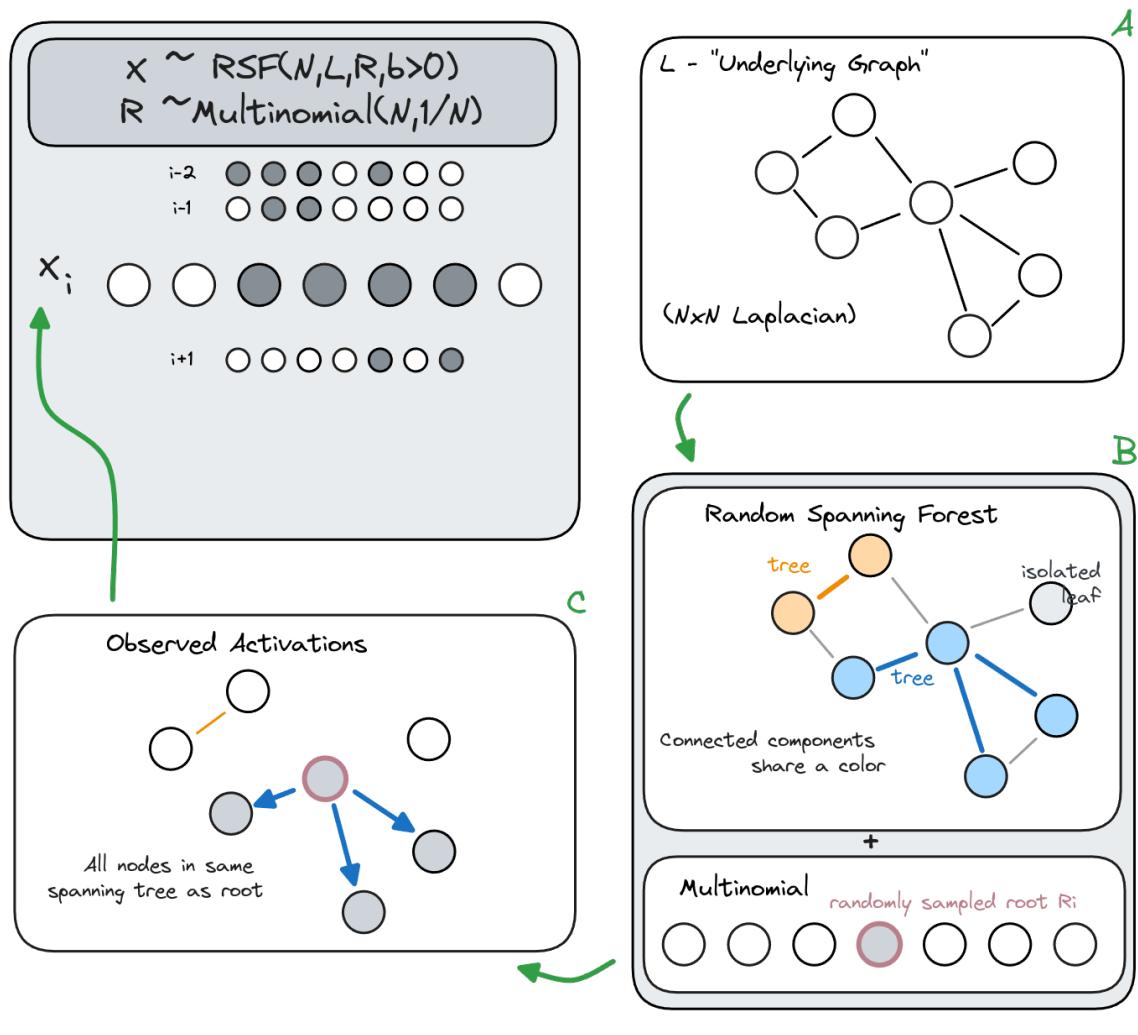
[GRAPHIC 2 - infection vector from meta node]

### 5.3.3 Matrix Tree and Forest Theorems

- one from kirchoff
- one from Chebotaryv

## 5.4 Generative Model Specification

### Random (Rooted) Spanning Forest (RSF) Observation Model



- 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

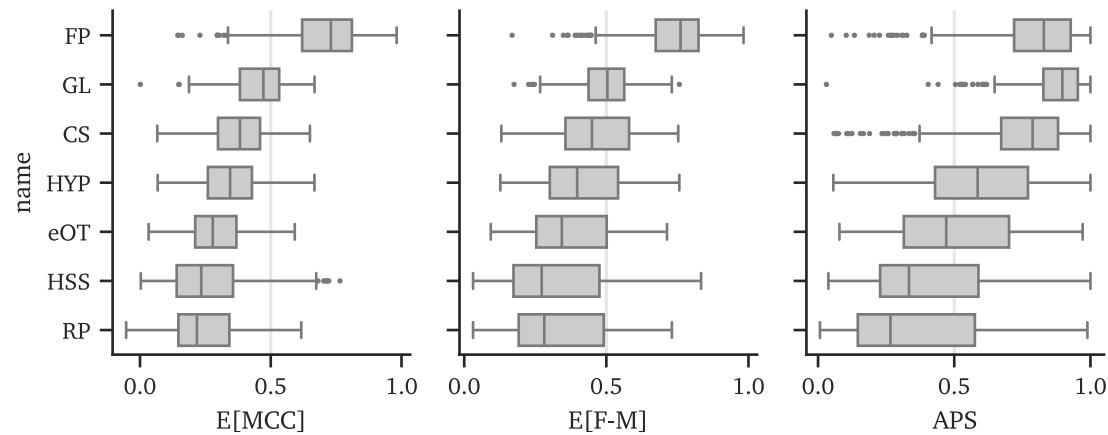


Figure 6.1: Comparison of MENDR recovery scores

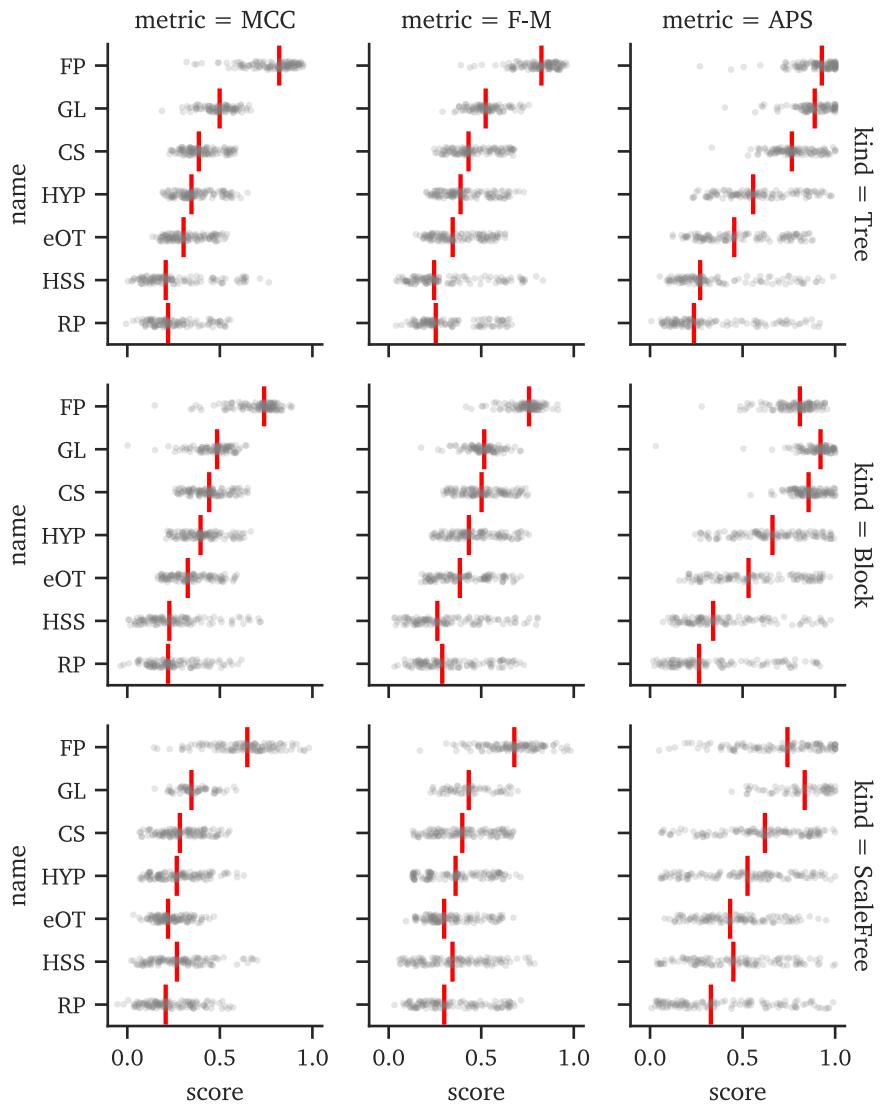


Figure 6.2: Comparison of MENDR Recovery Scores by Graph Type

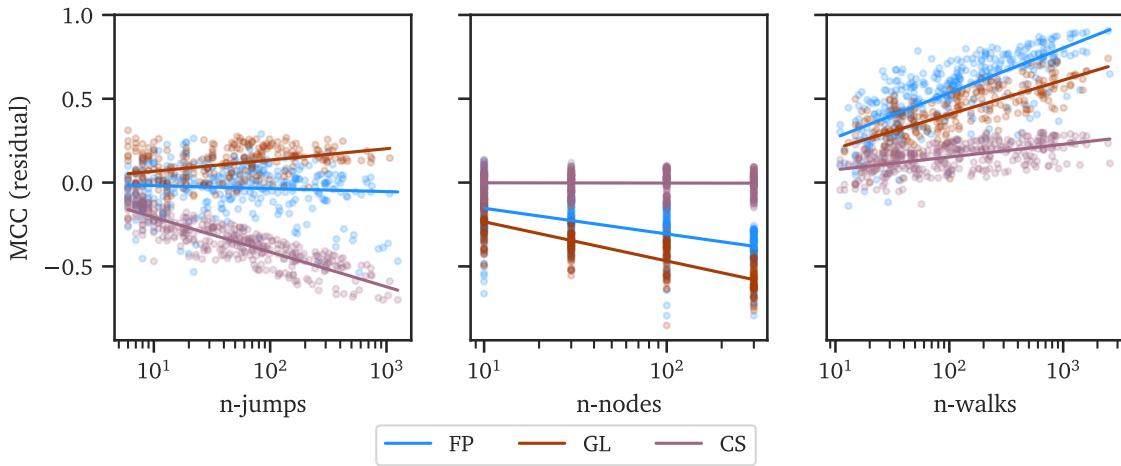


Figure 6.3: Partial Residuals (regression on  $E[MCC]$ )

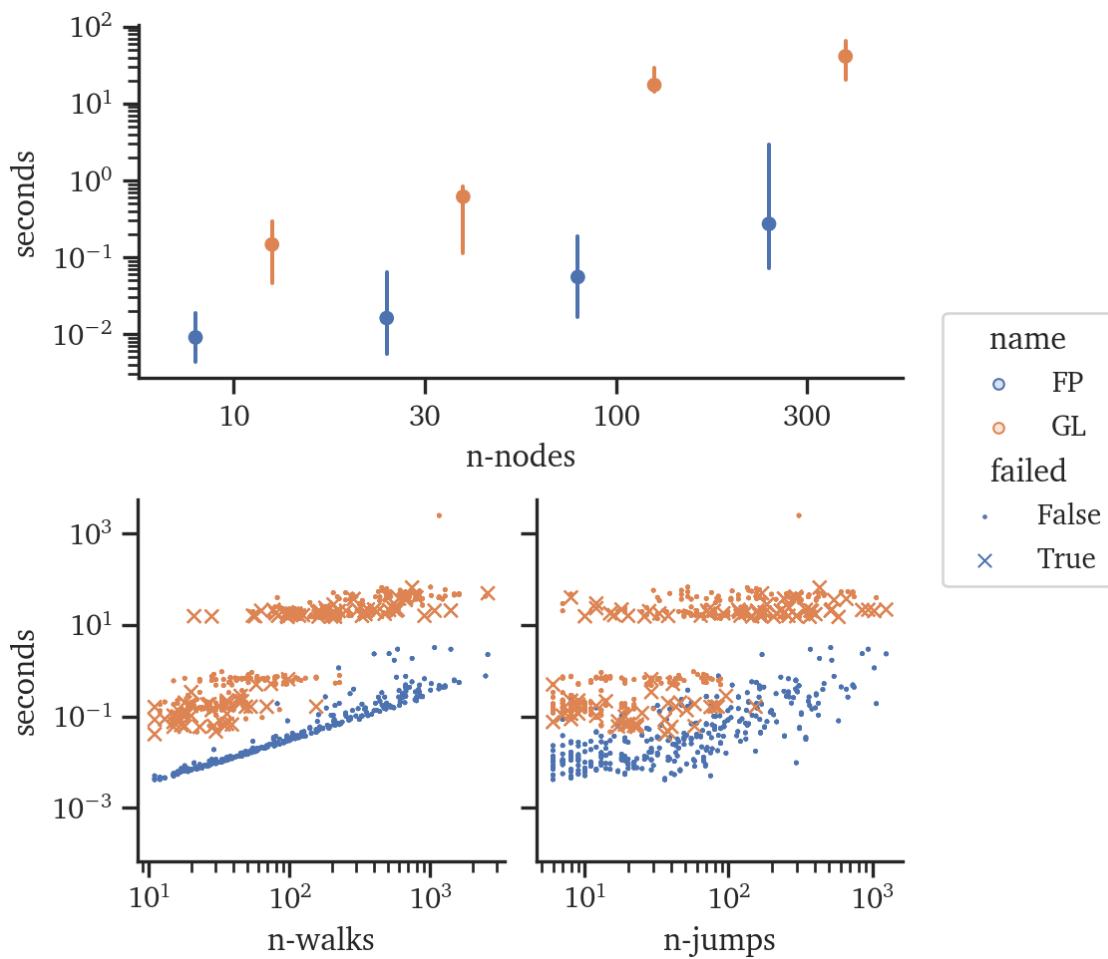


Figure 6.4: Runtime Scaling (Forest-Pursuit vs GLASSO)

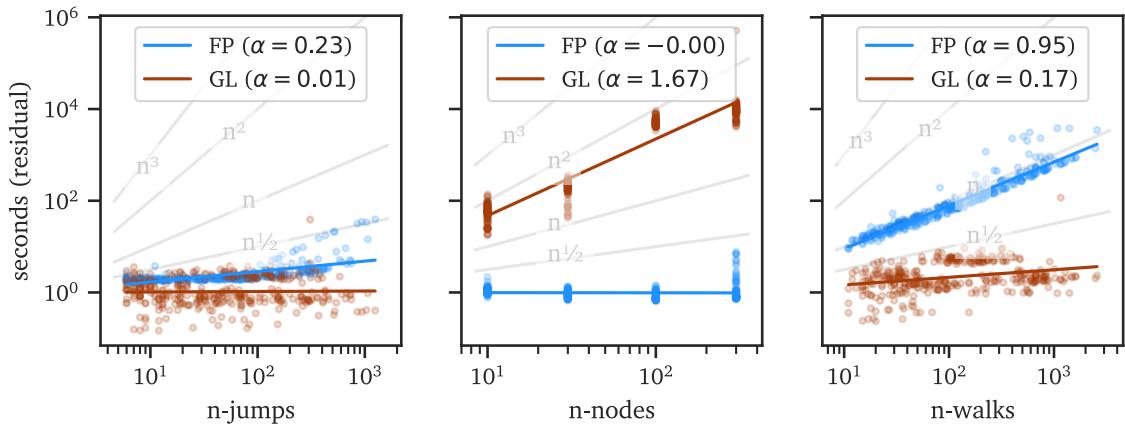
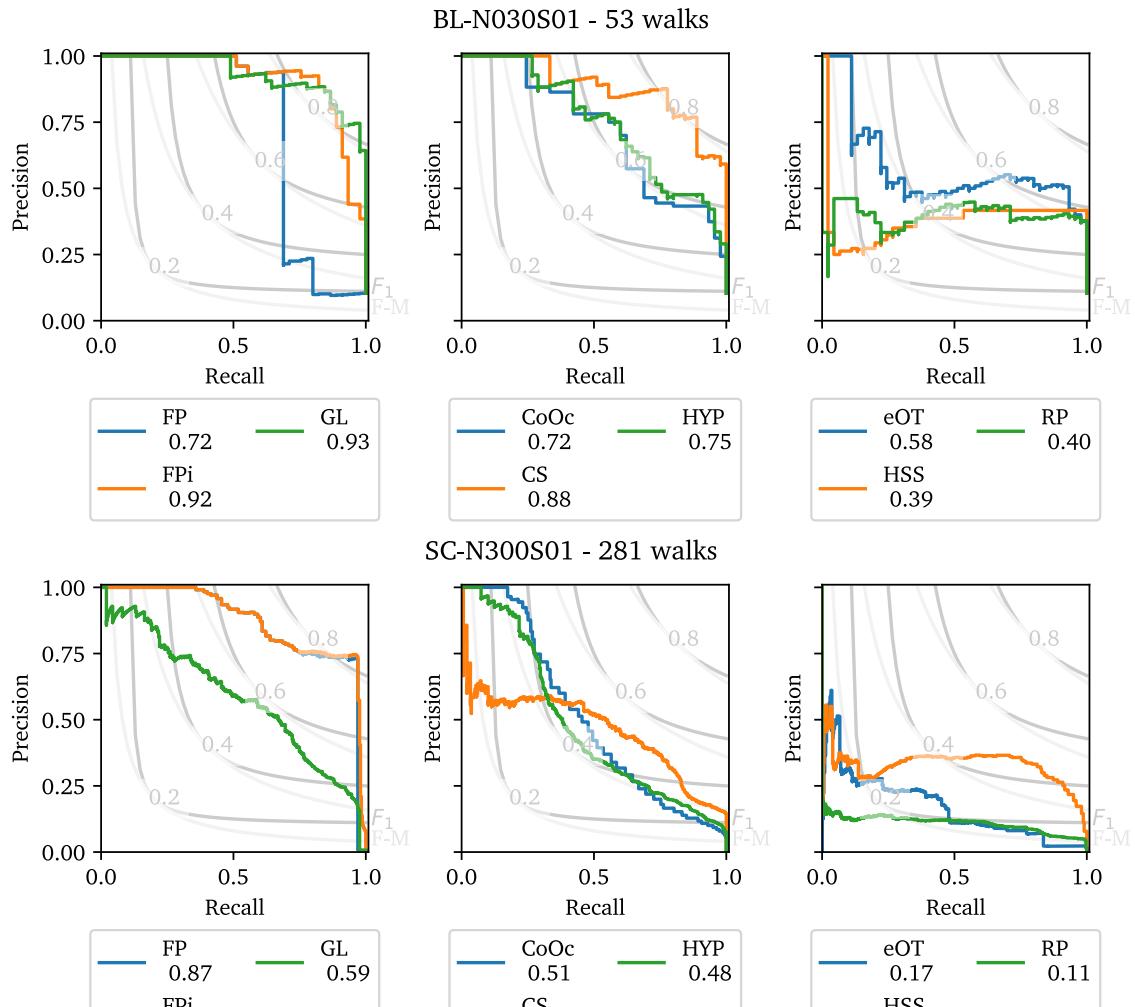


Figure 6.5: Partial Residuals (regression on computation time)

### 6.3.3 Results - Performance

## 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

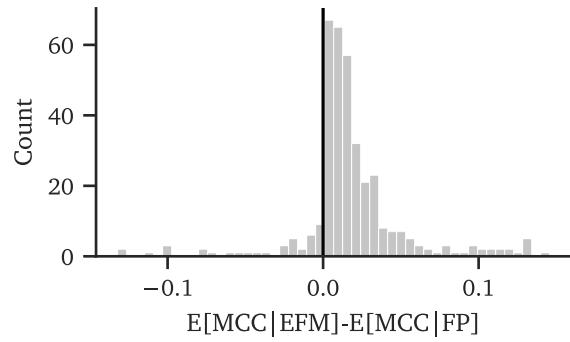


Figure 7.1: Change in Expected MCC (EFM vs FP)

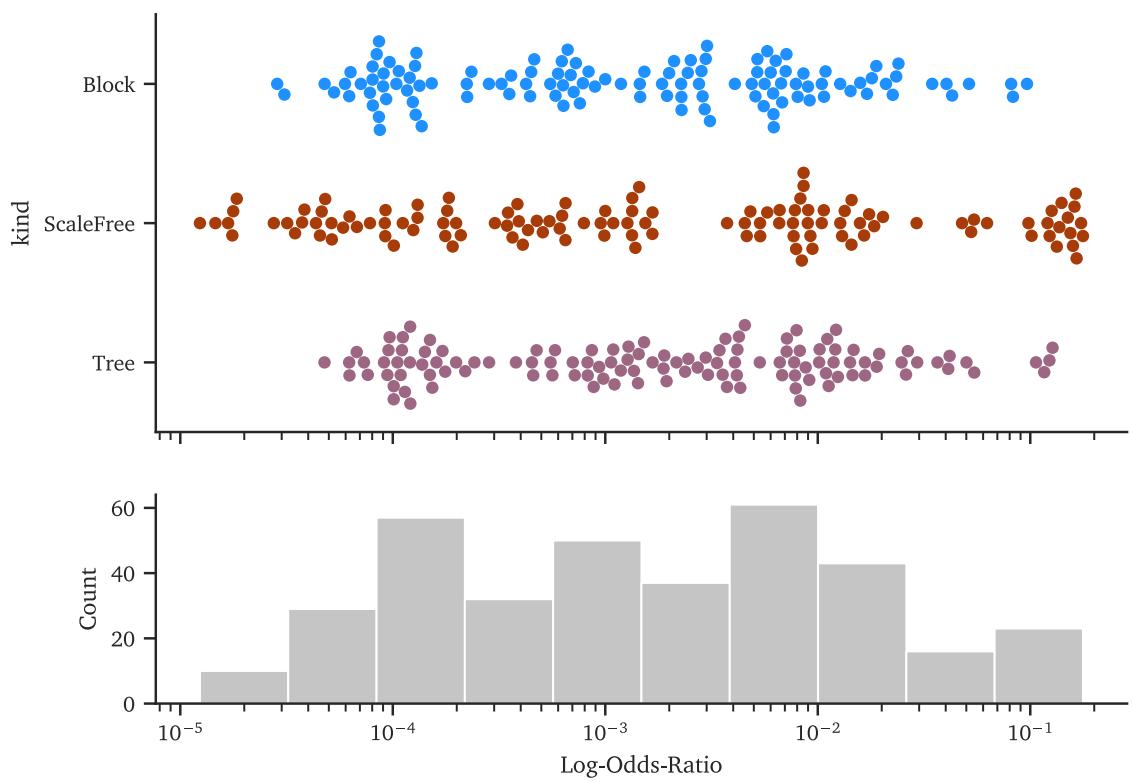


Figure 7.2: Logistic Regression Coef. (EFM - FP) vs. (Ground Truth)

## Part III

### Applications & Extentions

# Chapter 8: Qualitative Application of Relationship Recovery

## 8.1 Network Science Collaboration Network

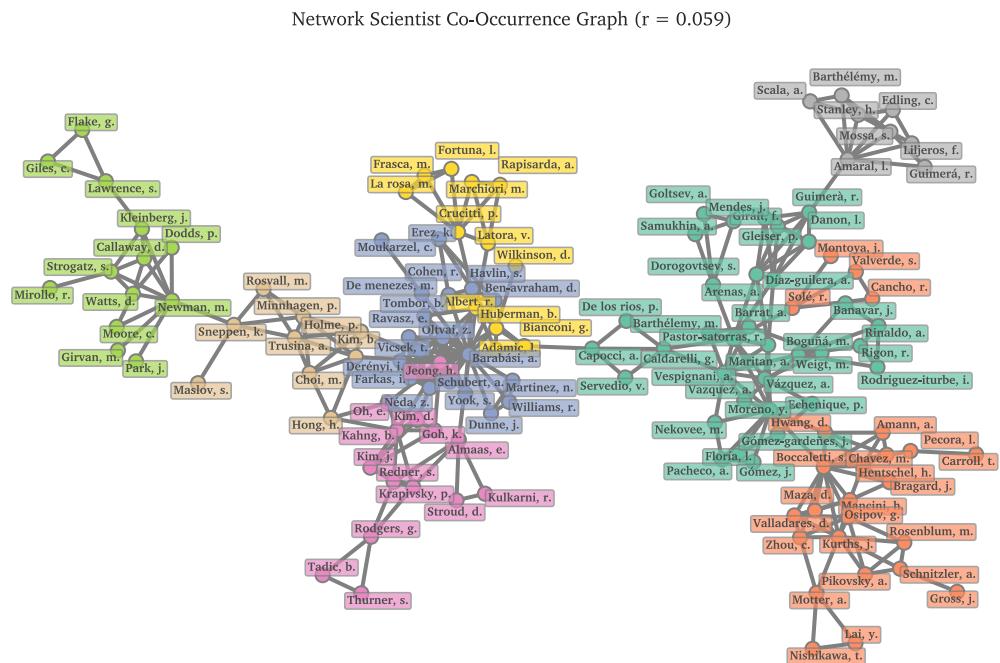


Figure 8.1: 134 Network scientists from [NEWMAN;BOCCALETTI;SNEPPEN], connected by co-authorship

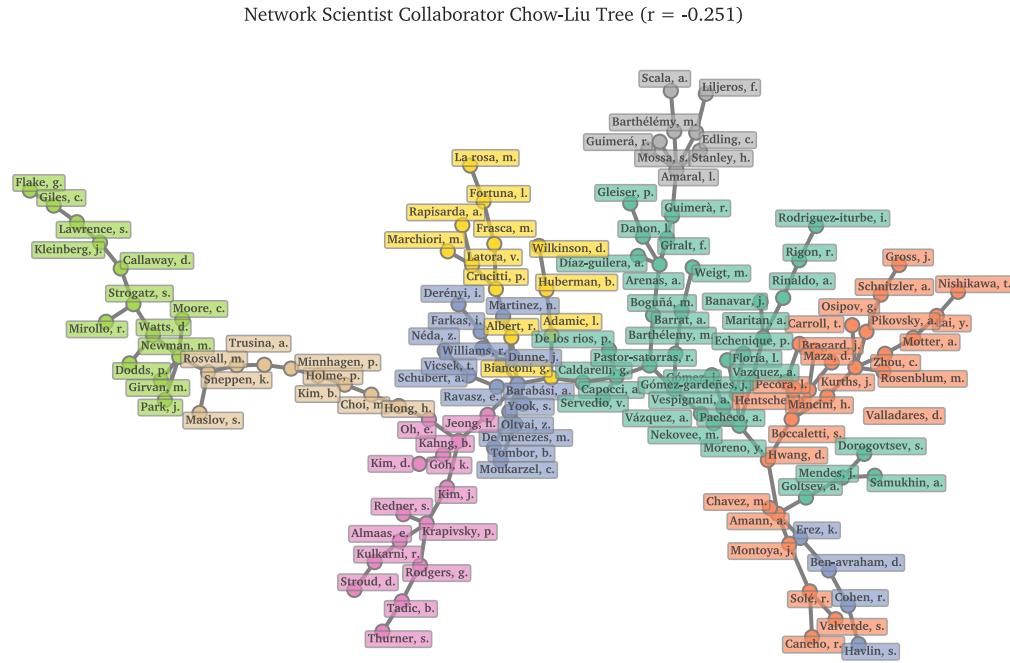


Figure 8.2: Max. likelihood tree dependency structure to explain co-authorships

## 8.2 Les Miserables Character Network

### 8.2.1 Backboning

### 8.2.2 Character Importance Estimation

Network Scientist Collaboration Network Estimate ( $r = -0.069$ )

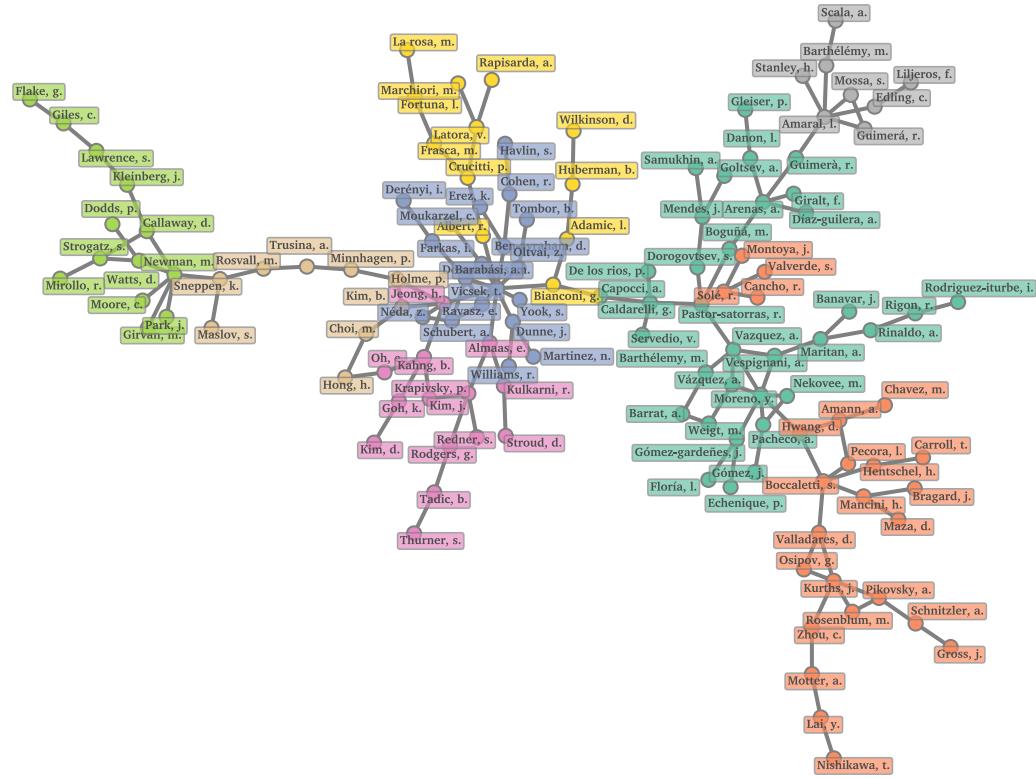


Figure 8.3: Forest Pursuit estimate of NetSci collaborator dependency relationships

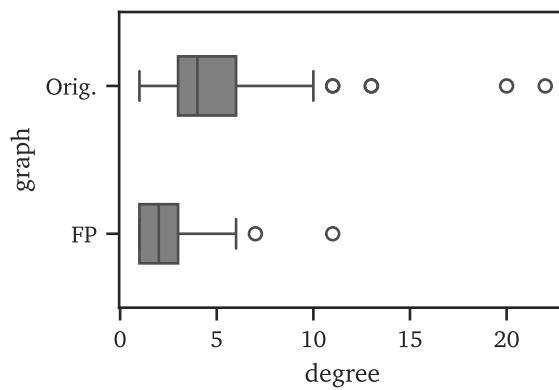


Figure 8.4

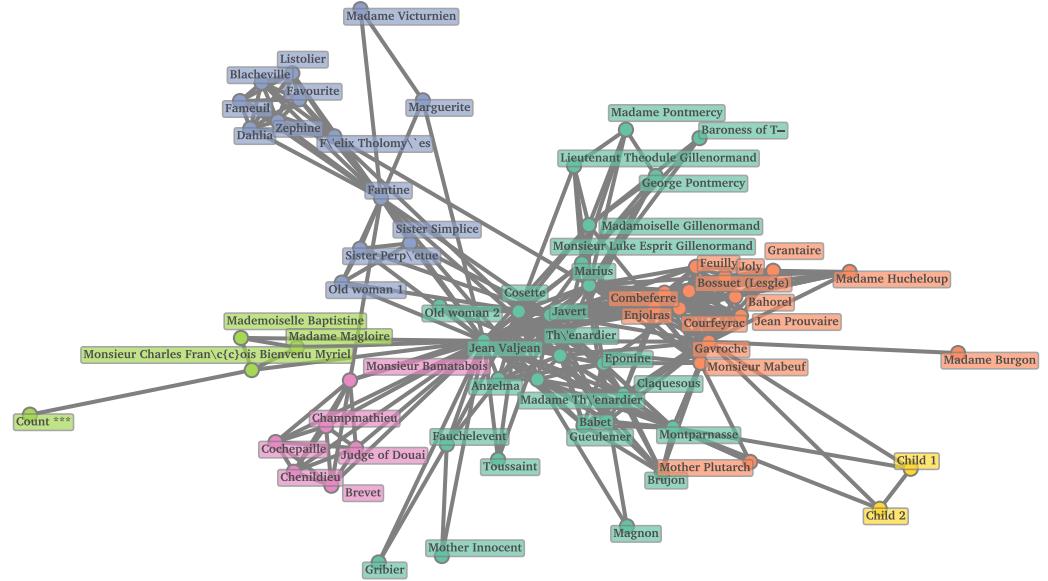


Figure 8.5

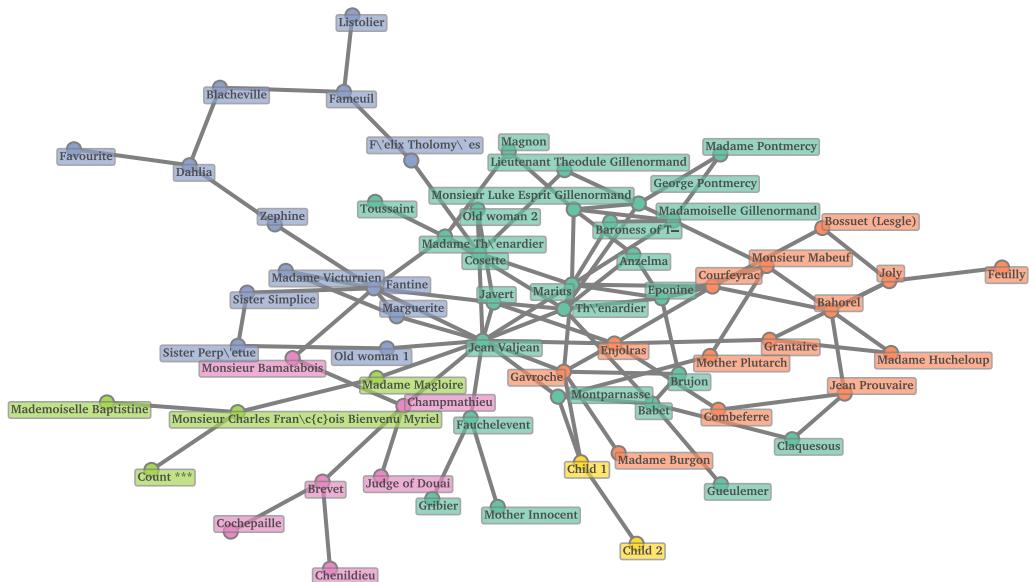


Figure 8.6

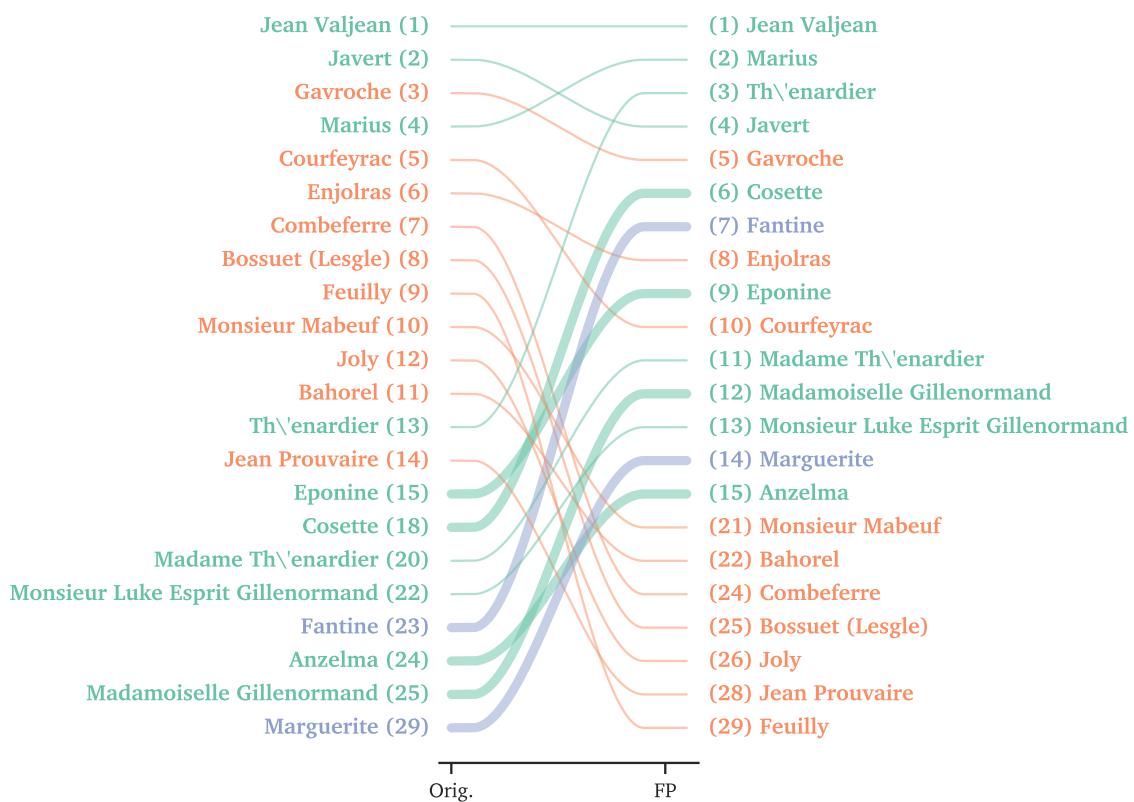


Figure 8.7

# Chapter 9: Recovery from Partial Orders

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

## 9.1 Technical Language Processing

*insert from [6, 10]*

## 9.2 Verbal Fluency Animal Network

### 9.2.1 Edge Connective Efficiency and Diversity

### 9.2.2 Thresholded Structure Preservation

Differences in structural preservation with increased thresholding.

### 9.2.3 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.

## Verbal Fluency Animals (DS-filtered) Co-Occurrence Graph ( $r = 0.33$ ) ( $\psi = 0.35$ )

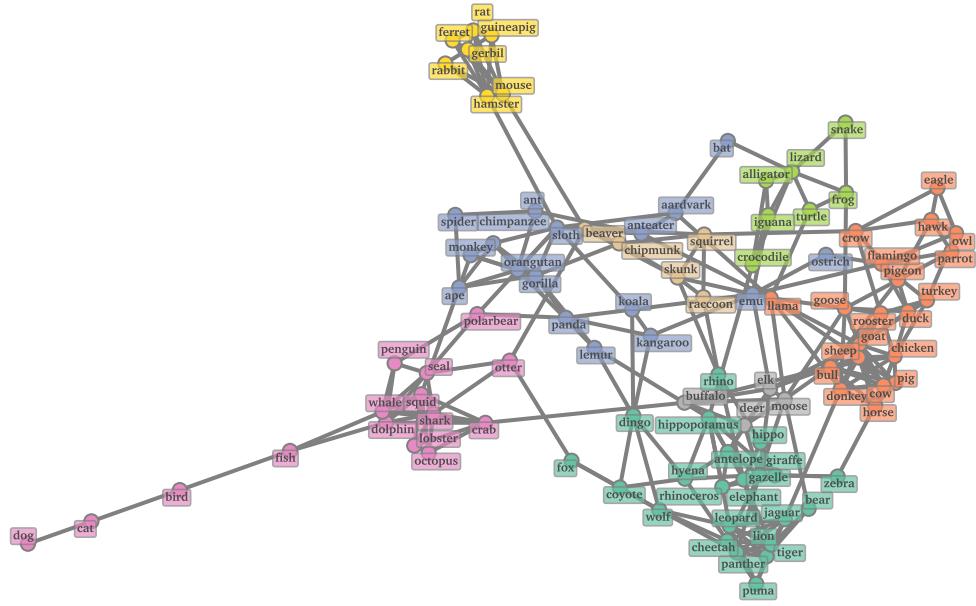


Figure 9.1

Verbal Fluency Animal Dependencies (Chow-Liu) Network ( $r = -0.13$ ) ( $\psi = 1.00$ )

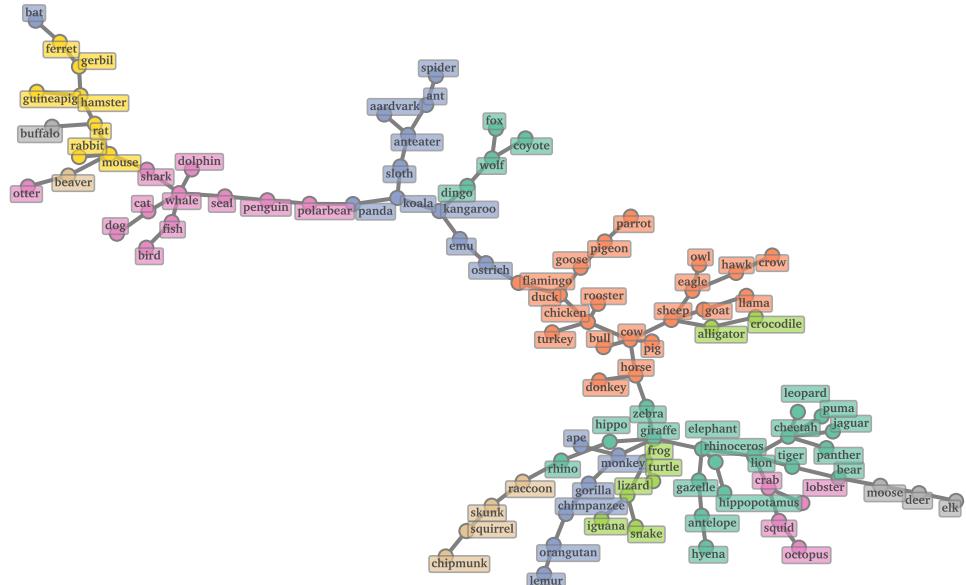


Figure 9.2

Verbal Fluency Animal Dependencies (GLASSO) Network ( $r = -0.02$ ) ( $\psi = 0.45$ )

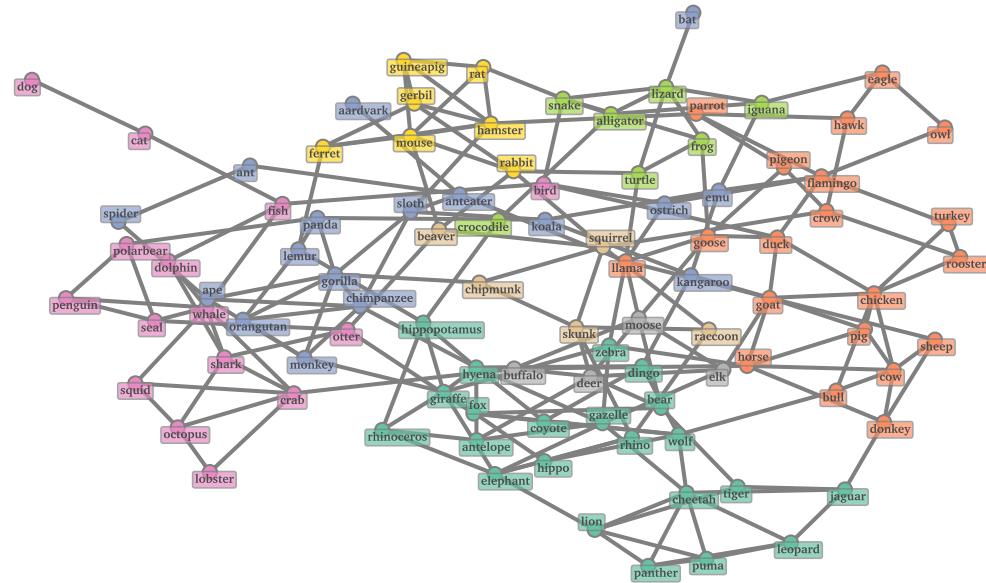


Figure 9.3

Verbal Fluency Animal Dependencies (FP) Network Estimate ( $r = -0.16$ ) ( $\psi = 0.84$ )

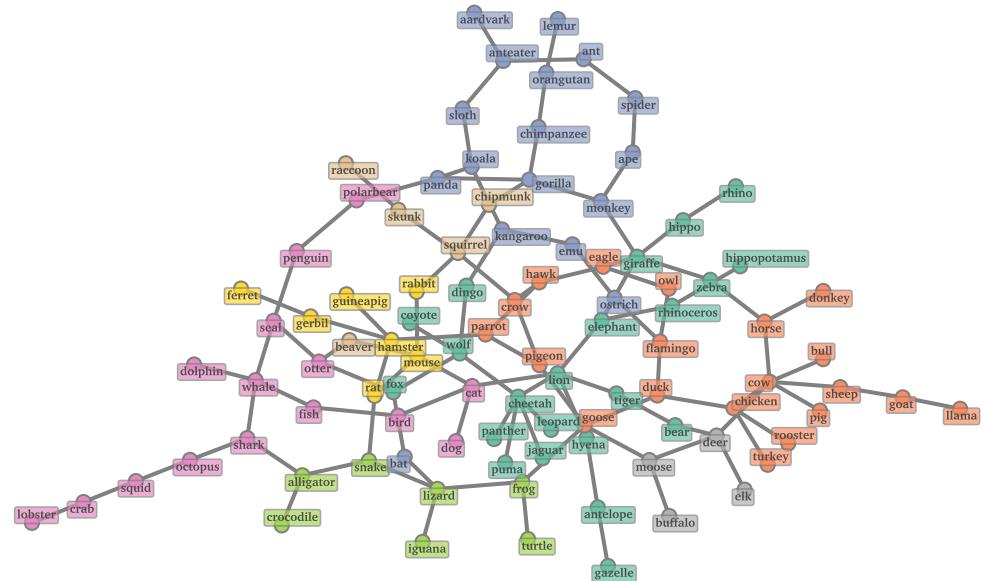
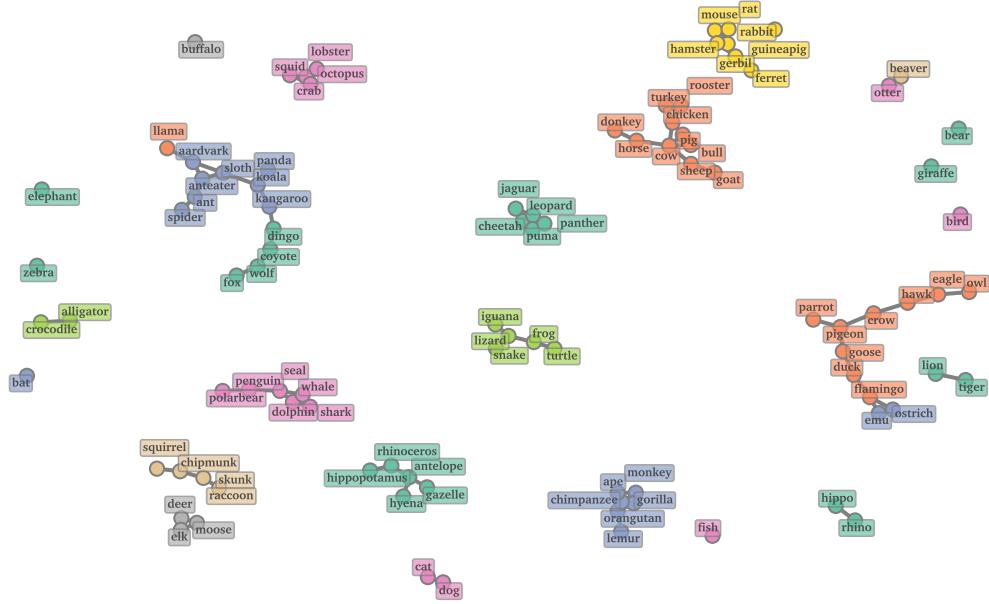


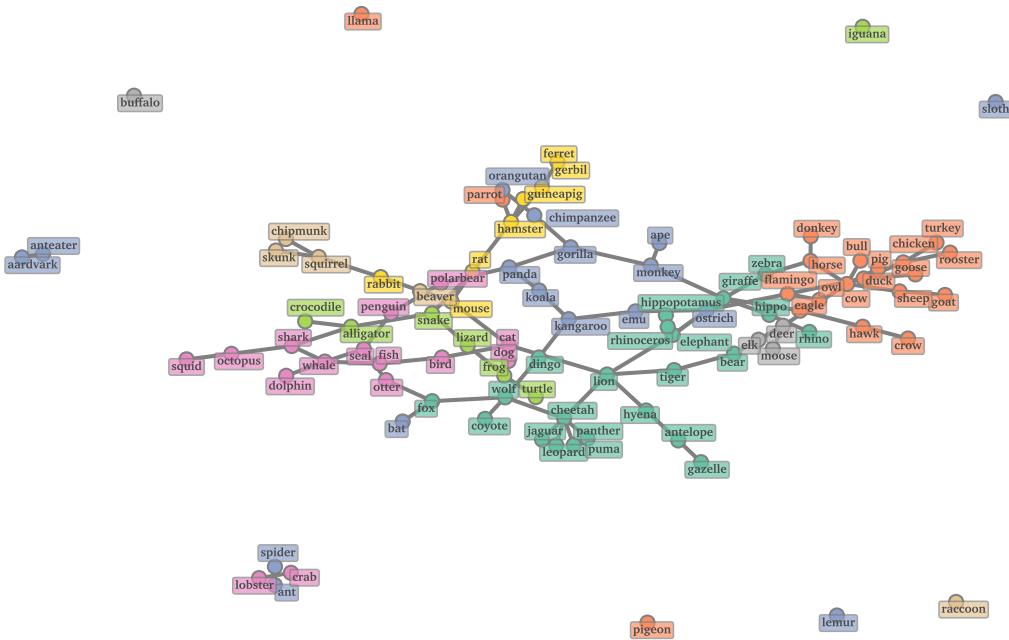
Figure 9.4: Comparison of backboning/dependency recovery methods tested vs. Forest Pursuit

Verbal Fluency Animals Co-Occurrence (DS 98%) Network ( $r = 0.36$ ) ( $\psi = 1.02$ )



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

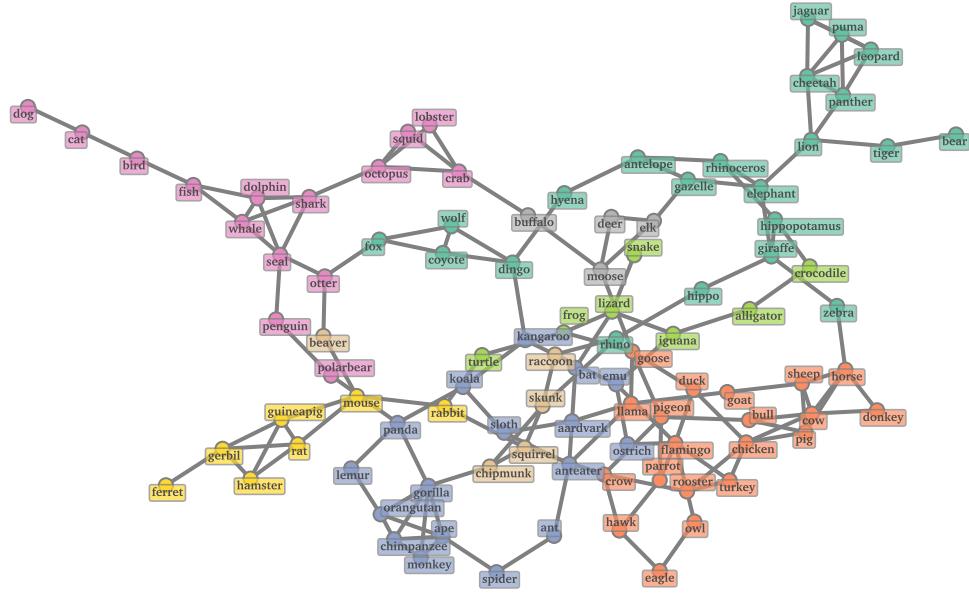
Verbal Fluency Animal Dependencies (FP 98%) Network ( $r = -0.11$ ) ( $\psi = 1.02$ )



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

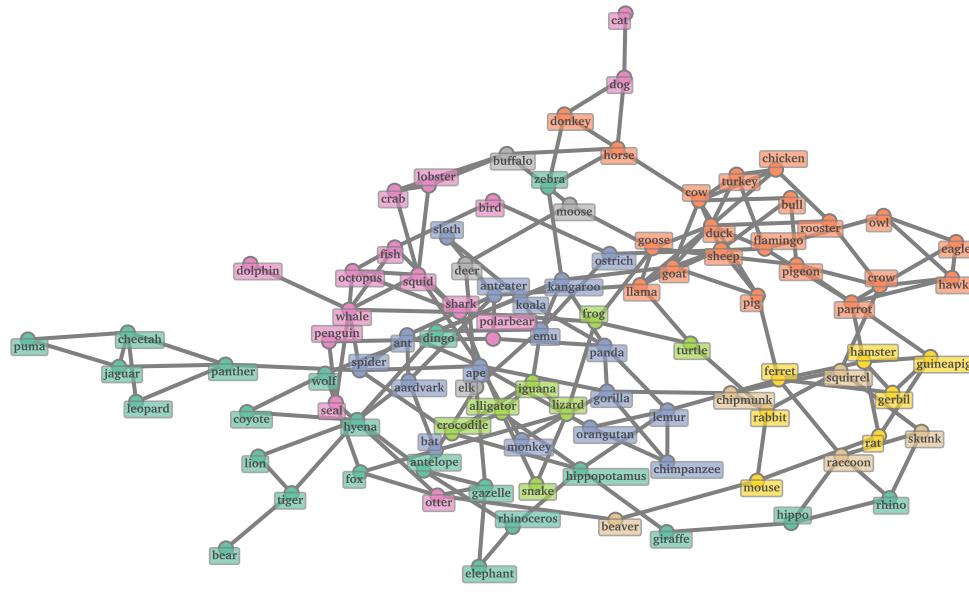
Figure 9.5: When only retaining the top 2% of edge strengths, blah

## Animal Dependencies (FP → DS) Network ( $r = 0.05$ ) ( $\psi = 0.61$ )



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

Animal Dependencies (FP → GLASSO) Network ( $r = 0.15$ ) ( $\psi = 0.61$ )



(b) Intact global structure with isolates

Figure 9.6: We might prefer to drop low-certainty/rare nodes from a preserved central structure.

## 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](https://doi.org/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](https://doi.org/10.48550/arXiv.2106.16120). arXiv: [2106.16120](https://arxiv.org/abs/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](https://doi.org/10.1137/20M1355896). Accessed: Feb. 1, 2023.
- [4] R. Zmigrod, T. Vieira, and R. Cotterell, “Efficient computation of expectations under spanning tree distributions,” *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 675–690, Jul. 8, 2021, ZSCC: 0000004, ISSN: 2307-387X. doi: [10.1162/tacl\\_a\\_00391](https://doi.org/10.1162/tacl_a_00391). Accessed: May 8, 2023.
- [5] M. B. Paulus, D. Choi, D. Tarlow, A. Krause, and C. J. Maddison, *Gradient estimation with stochastic softmax tricks*, 2020. doi: [10.48550/ARXIV.2006.08063](https://doi.org/10.48550/ARXIV.2006.08063).
- [6] 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](https://doi.org/10.1115/1.4045686).
- [7] R. Sonthalia and A. C. Gilbert, “Tree! i am no tree! i am a low dimensional hyperbolic embedding,” arXiv, arXiv:2005.03847, Oct. 22, 2020, type: article. doi: [10.48550/arXiv.2005.03847](https://doi.org/10.48550/arXiv.2005.03847). arXiv: [2005.03847](https://arxiv.org/abs/2005.03847).
- [8] M. Angeletti, J.-M. Bonny, and J. Koko, “Parallel Euclidean distance matrix computation on big datasets,” working paper or preprint, 2019. [Online]. Available: <https://hal.science/hal-02047514>.
- [9] K. Avrachenkov, P. Chebotarev, and D. Rubanov, “Similarities on graphs: Kernels versus proximity measures,” *European Journal of Combinatorics*, Special Issue in Memory of Michel Marie Deza, vol. 80, pp. 47–56, Aug. 1, 2019, ZSCC: 0000018, ISSN: 0195-6698. doi: [10.1016/j.ejc.2018.02.002](https://doi.org/10.1016/j.ejc.2018.02.002). Accessed: May 8, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0195669818300155>.

- [10] 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](https://doi.org/10.1115/detc2019-98429).
- [11] T. P. Peixoto, "Reconstructing networks with unknown and heterogeneous errors," *Physical Review X*, vol. 8, no. 4, p. 041011, Oct. 16, 2018, ZSCC: 0000099. doi: [10.1103/PhysRevX.8.041011](https://doi.org/10.1103/PhysRevX.8.041011). Accessed: Nov. 2, 2023.
- [12] 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](https://doi.org/10.1080/10556788.2016.1193176). Accessed: May 8, 2023.
- [13] D. N. Fisher, M. J. Silk, and D. W. Franks, "The perceived assortativity of social networks: Methodological problems and solutions," in *Trends in Social Network Analysis*. Springer International Publishing, 2017, pp. 1–19, ISBN: 9783319534206. doi: [10.1007/978-3-319-53420-6\\_1](https://doi.org/10.1007/978-3-319-53420-6_1).
- [14] J. Kepner et al., "Mathematical foundations of the GraphBLAS," in *2016 IEEE High Performance Extreme Computing Conference (HPEC)*, Sep. 2016, pp. 1–9. doi: [10.1109/HPEC.2016.7761646](https://doi.org/10.1109/HPEC.2016.7761646).
- [15] Z. Neal, "The backbone of bipartite projections: Inferring relationships from co-authorship, co-sponsorship, co-attendance and other co-behaviors," *Social Networks*, vol. 39, pp. 84–97, Oct. 1, 2014, ZSCC: 0000200, issn: 0378-8733. doi: [10.1016/j.socnet.2014.06.001](https://doi.org/10.1016/j.socnet.2014.06.001). Accessed: May 7, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0378873314000343>.
- [16] B. Dai, S. Ding, and G. Wahba, "Multivariate bernoulli distribution," *Bernoulli*, vol. 19, no. 4, Sep. 2013, issn: 1350-7265. doi: [10.3150/12-bejsp10](https://doi.org/10.3150/12-bejsp10).
- [17] 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](https://doi.org/10.48550/arXiv.1307.3810). arXiv: [1307.3810](https://arxiv.org/abs/1307.3810).
- [18] 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](https://doi.org/10.1145/2086737.2086741).
- [19] 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](https://doi.org/10.1016/j.socnet.2012.01.003).
- [20] J. Kepner and J. Gilbert, *Graph Algorithms in the Language of Linear Algebra*. Philadelphia: Society for Industrial and Applied Mathematics, Aug. 4, 2011, 375 pp., ISBN: 9780898719901.

- [21] P. B. Slater, “A two-stage algorithm for extracting the multiscale backbone of complex weighted networks,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 26, E66–E66, Jun. 30, 2009, ZSCC: 0000040. doi: [10.1073/pnas.0904725106](https://doi.org/10.1073/pnas.0904725106). Accessed: May 7, 2024. [Online]. Available: <https://www.pnas.org/doi/abs/10.1073/pnas.0904725106>.
- [22] G. E. Moorhouse, “Incidence geometry,” *University of Wyoming*, pp. 3–5, 2007.
- [23] 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](https://doi.org/10.48550/arXiv.math/0602070). arXiv: [math/0602070](https://arxiv.org/abs/math/0602070) [math].
- [24] 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](https://doi.org/10.1073/pnas.122653799).
- [25] M. E. J. Newman, “Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality,” *Physical Review E*, vol. 64, no. 1, p. 016132, Jun. 2001, issn: 1095-3787. doi: [10.1103/physreve.64.016132](https://doi.org/10.1103/physreve.64.016132).
- [26] H. Whitehead and S. Dufault, “Techniques for analyzing vertebrate social structure using identified individuals: Review and recommendations,” *Advances in the Study of Behavior*, vol. 28, no. 28, pp. 33–74, 1999.
- [27] 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](https://doi.org/10.1086/jar.33.4.3629752).