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

List of Tables

List of Figures

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. [26] 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[3], treatment of error as purely aleatory, rather than epistemic [29], or simply ignoring measurement error in network reconstruction entirely[16].

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*[48], 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.[40]

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.

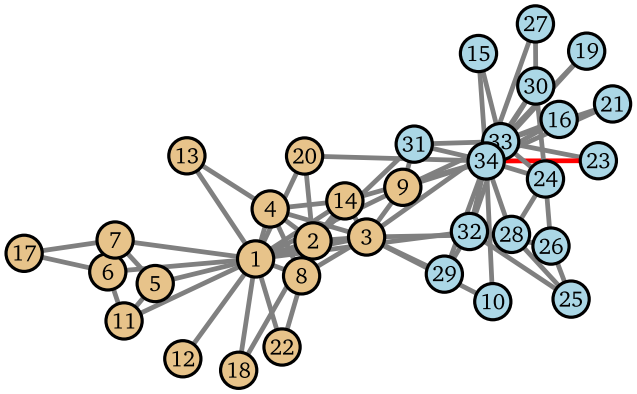


Figure 1.1: Zachary's Karate Club, with ambiguously extant edge 78 highlighted.

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.

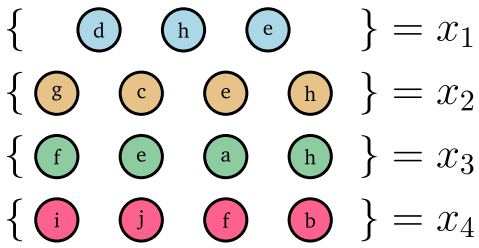


Figure 1.2

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. [41] 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".

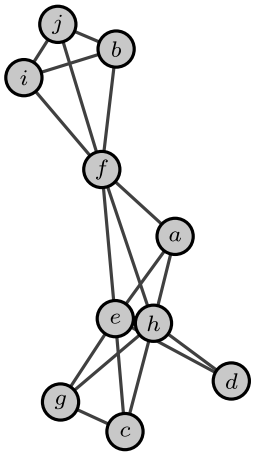


Figure 1.3

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

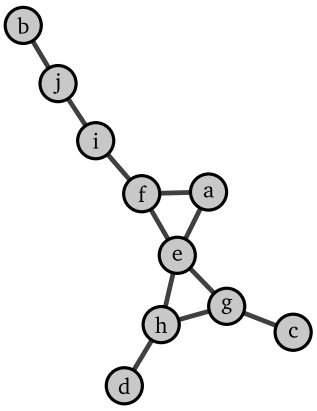


Figure 1.4: graph of mutual collaboration relationships i.e. the “ground truth” social network

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

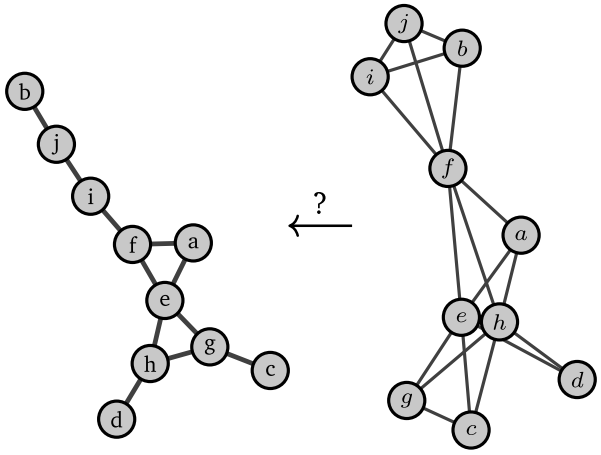


Figure 1.5: Recovering underlying dependency networks from node-cooccurrences.

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 it’s core topology in a way that’s easier to visualize.[22, 32] 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. [3]. 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. $x \in \mathbb{S} \in \{\mathbb{R}, \mathbb{Z}, \mathbb{N}, \cdots\}$. 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 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

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

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

$$x = X(i, j) \qquad X : I \times J \to \mathbb{S} \tag{2.1}$$

i.e. the measured value of the jth independent variable from the ith observation.[2] 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*):

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

[2] This notation is adapted from the sparse linear algebraic treatment of graphs in Kepner and Gilbert [30] and Kepner et al. [20].

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.[3]

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^{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 $\mathbf{x}'_j : \mathbb{B}^{1 \times m}$.

2.2 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

[3] 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 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[4], 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*:

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

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

[4] 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 $s \sim Bin(p,k)$. 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 smooth-

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,"[5] 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 Beta prior on the Bernoulli probability $p \sim \text{Beta}(\alpha, \beta)$, such that the posterior distribution for p after our observations is $\text{Beta}(s+\alpha, k-s+\beta)$, which has the mean:

$$E[p \mid s, k] = \frac{s+\alpha}{k-\alpha+\beta} \tag{2.2}$$

This exactly recovers additive smoothing with a Jeffrey's prior for $\alpha = \beta = 0.5$.[6] 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.

[5] derived when Laplace desired estimates of probability for unobserved phenomena, such as the sun (not) rising tomorrow.

[6] 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.[7]

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 $P(A \mid B) = \frac{P(A \cap B)}{P(B)}$ measure how often A occures 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,

[7] In fact, a recent method from [24] 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 interractions, 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 6.4.

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 ($\otimes$).[8] With this, we are able to define generalized inner products that take pairs vectors in a vector space $\mathbf{x} \in V$, such that $\langle \cdot, \cdot \rangle_F : \mathbb{S}^n \times \mathbb{S}^n \to \mathbb{S}$.

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

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. [20] 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 $(\mathbb{R} \cup -\infty, \min, +)$. 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.

[8] Or, more generally, a semiring if inverse operations for $\oplus, \otimes$ don't exist.

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

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?]:[9]

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

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

[9] **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 $\mathbb{S}^n$ [13][10]

$$d_K(j,j') = \tfrac{1}{2}\left(K(j,j) + K(j',j'))\right) - K(j,j') \tag{2.5}$$

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 $\tilde{\kappa}(x,y) = \log s(x,y)$.

$$\begin{aligned} \tilde{K}(j,j') &= \frac{K(j,j')}{\sqrt{K(j,j)^2 K(j',j')^2}} \\ d_{\tilde{K}}(j,j') &= -\log \tilde{K}(j,j') \end{aligned} \tag{2.6}$$

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

[10] 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* $\mathfrak{P}$, *lines* $\mathfrak{L}$, and *flags* $\mathfrak{I}$.[35]

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

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 $\mathfrak{I}$, 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

$$P(A \cap B \mid C) = P(A \mid C)P(B \mid C) \implies (A \perp B \mid C) \implies (A, B) \notin \mathfrak{I} \tag{2.7}$$

for a set of incidences $\mathfrak{I}$ 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[7, 20], we might say a graph on nodes (or, "vertices") $v \in V = \{1, \cdots, n\}$ 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:[1]

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

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

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

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, $e \in E = \{1, \cdots \omega\}$. Now, define graphs as incidence structures between nodes and edges such that every edge is incident to either two nodes, or none:

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

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

- Self loops can be prohibited by only allowing unique flags for a given relation[2]

[1] The *indicator function* $\mathbf{1}_A(x)$ is 1 for values of x in the set A, and 0 otherwise.
[2] never two flags with the same pair, i.e. $\mathscr{I}$ is a set, not a multiset.

- Multigraphs are similarly described by whether we allow pairs of vertices to appear with more than one edge[3]

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

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

- 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 $\mathscr{I} = E \times V$ is no longer a subset relation for $E = \{1, \cdots, \binom{n}{2}\}$. Instead of existence in $\mathscr{I}$, we explicitly use incidence relation S to tell us whether each possible edge "exists" or not, simplifying our graph definition further[4]:

[3] in the set of flags containing nodes u or v, only one e may be incident to both of them.
[4] if we allomulti-edges , then

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

$$v \in V = \{1, \cdots, n\} \tag{3.3}$$
$$e \in E = \left\{1, \cdots, \binom{n}{2}\right\}$$

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.[5] 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[6].

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 $\mathbf{e}$ 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

[5] In fact, this would make B^*(v,e) equivalent to a *graphical matroid*, another common formalism that generalizes graph-like structures to vector space representations.

[6] though not always!

vectors are:

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

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

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

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 $u < v$.[7] 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

[7] 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 $\mathbb{S}$.[8]

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. [12], 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.

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

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

[8] 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.

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

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 ajacency/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: $L^{\text{rw}} = D^{-1}L$ 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.

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

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 it's 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.[9] 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

[9]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 ith observed structure on n nodes:

$$
\begin{gathered}
R(i,e) = \min(\{B_i(e,u_n(e)), B_i(e,v_n(e))\}) \\
\text{s.t.} \quad R : I \times E \to \mathbb{S} \\
i \in I = \{1,\cdots,m\} \qquad e \in E = \{1,\cdots,\omega\} \\
n = \tfrac{1}{2}(1+\sqrt{8\omega+1})
\end{gathered}
\tag{3.9}
$$

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.[10]

3.2 Generalized node occurrence vectors

TODO

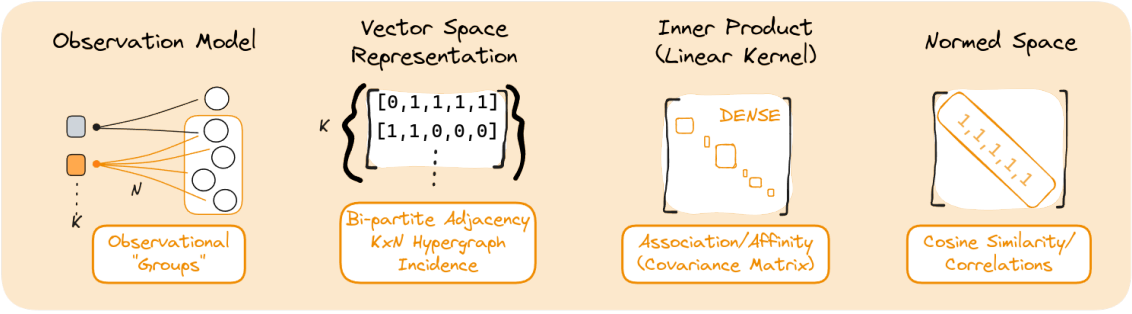


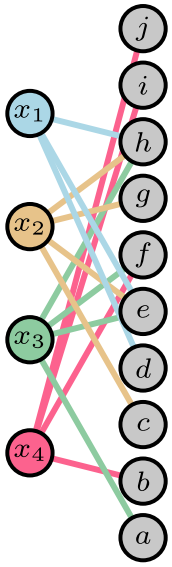
Figure 3.1: Hyperedge Relation Observational Model

[10]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

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

(a)



(b) Bipartite representation of node “activation” data

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

we can actually interpret ochiai/cosine as an approximate bayes update.

Say we observe binary variables A and B, along with others.

Now, we want a probability of conditional dependence between A and B (does information flow directly between a and b when a and b happen together?). So we want the probability ofand edge E_{ab} being used, given that such an edge had the opportunity to be used. E.g. did two people have a causal interraction to make each other sick, given a time when we know they were both exposed/became sick.

$$P(E \mid O) = P(O \mid E)P(E)/P(O)$$

The denominator is hard, because while we can estimate the frequency of each node as the occurrences/opportunities (i.e. events/exposure), so, n_i/N, we can't use that for an exposure for "number of times an edge between A,B could have been used". If we multiply $N(a)N(b)$, then we have the number of *ways* a or b could be related over all chances, but this won't be a fraction of the number of samples, and could possible be much bigger. So dividing the number of times both *did* happen together by that number won't get us a probability. So instead, we fib a bit.

The number of chances (out of all samples) that a pair had to happen together is somewhere between the chances each had separately. We make a pseudo-variable that uses this fact, but averaging the rates. but we are dealing with probabilities, which are based around "areas" and their ratios. So we want one count, such that watching it with a copy of itself has the same exposure as watching A and B separately. This is exactly what geometric means are for:

Then, a point estimate for the probability of an edge occurring is its actual co-occurrence $n_{a,b}/N$. It's the ratio of these that give us the Ochiai as a probability: divided by the estimate for the co-occurrence opportunities

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

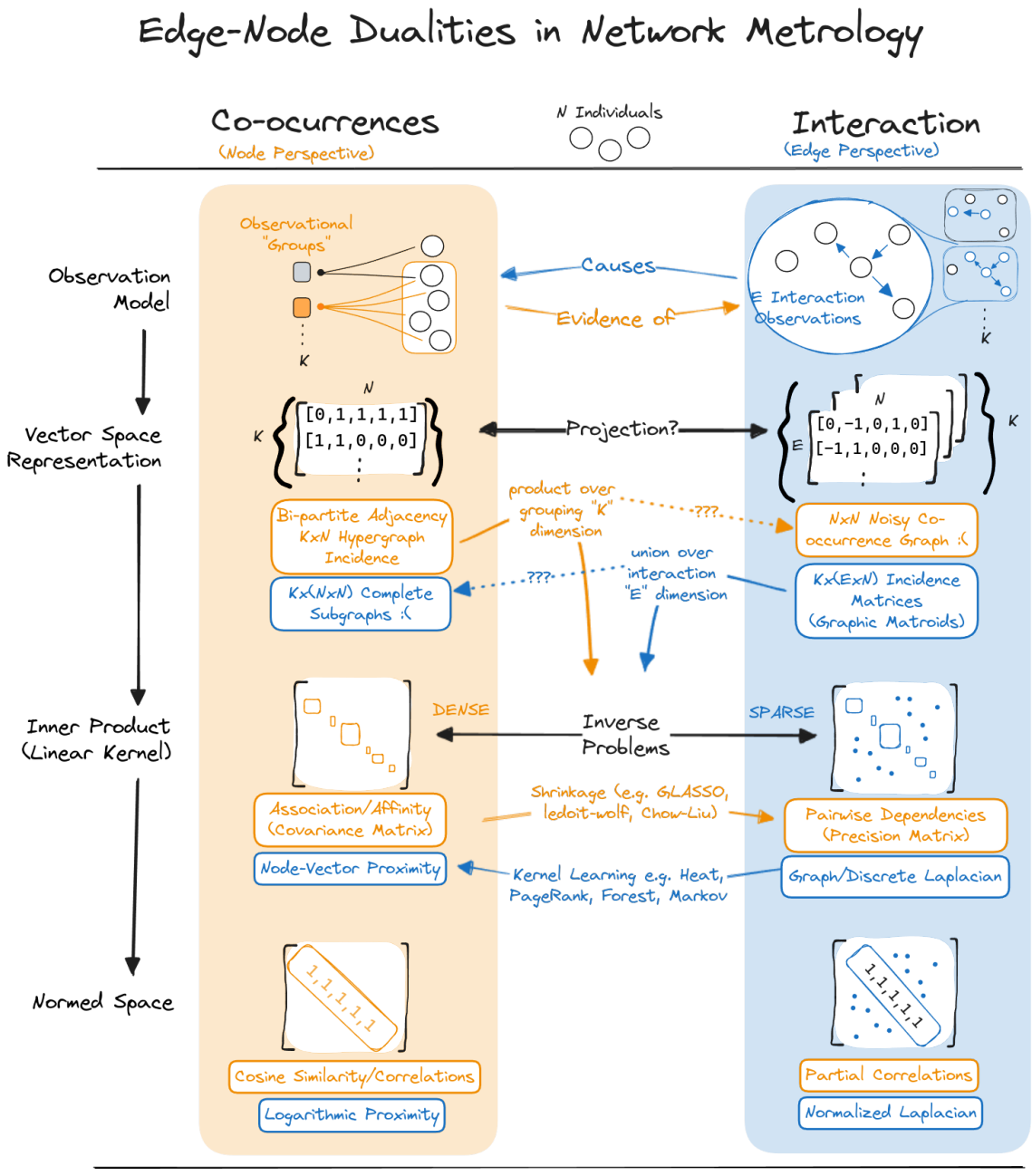


Figure 4.1: Relating Graphs and Hypergraph/bipartite structures as adjoint operators

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: Graph Reduce & Desire Paths

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.

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. [17]. Then we use the implicit causal dependency tree structure of each observation, together with the Matrix Forest Theorem [25, 38] 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 [42], where Whitehead and Dufault call this the *gambit of the group*.

> So, consiously or unconsciously, many ethnologists studying social organization makr what might be called the "gambit of the group": the 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,[1] where the author of Fisher et al. [18] advises that "group-based methods"

[1]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. [3]. 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-occurence-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

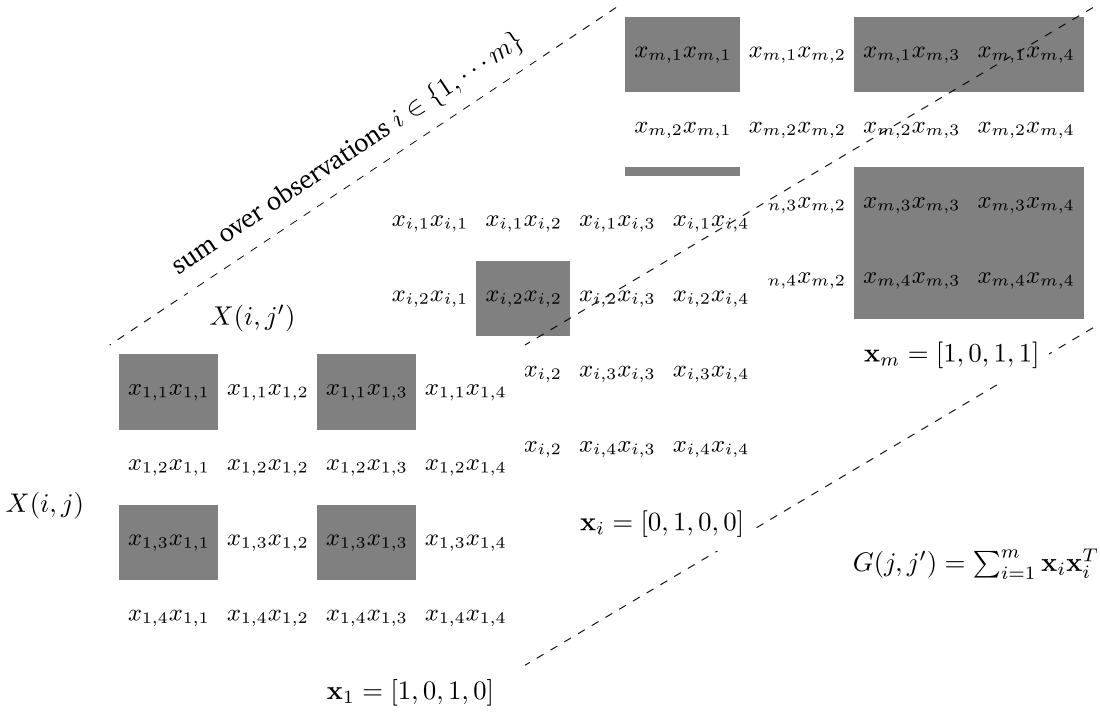
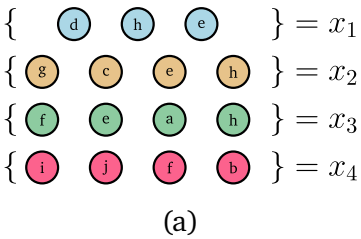


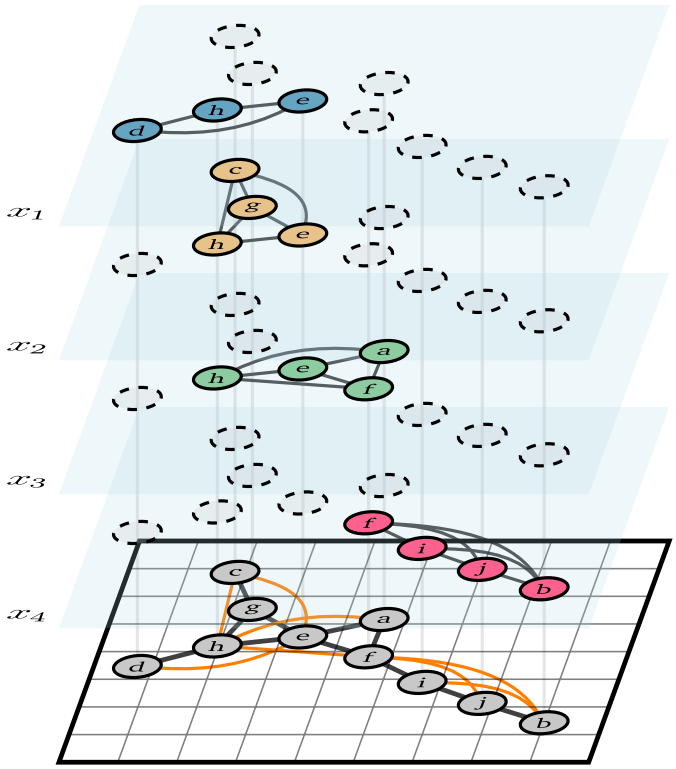
Figure 5.1: Gram matrix as sum of observation outer products

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



(a)



(b) Edge Measurements with Group Gambit (BoW) assumption

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 topolgy 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][2], or graphs whose modeled

[2]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 graps][3]. 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). 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).[4] So, each $\mathbf{x}_i$ will induce a subgraph $g_i = G^*[V_i]$, where $V_i = \{v \in \mathcal{V} \mid X(i, \mathcal{V}) = 1\}$. Then, our domain knowledge takes the form of a constraint on edges within that subgraph, which will induce a family of subgraphs on g_i. This family $\mathscr{C}_i$ belongs to the relevant class of graphs $\mathscr{C}$, limited to nodes V_i, i.e.

$$\begin{aligned} \mathscr{C}_i &= \{(V, E, B_i) \in \mathscr{C} \mid B_i(e, v) = B^*(e, v)\mathbf{1}_{V_i}(v)\mathbf{1}_{E_i}(e)\} \\ E_i &\in \{\mathscr{E} \in \mathscr{P}(E) \mid g_i[\mathscr{E}] \in \mathscr{C}\} V_i = \{v \in \mathcal{V} \mid X(i, \mathcal{V}) = 1\} \end{aligned} \tag{5.1}$$

[3] e.g. when interactions are limited to planar dependencies, like inferring ancient geographic borders.

[4] 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 [11] use a greedy embedding that minimizes distortion to estimate the need for added *Steiner* nodes.

[5]

In other words, the edges that "caused" to the node activations in a given observation must *together* belong to a graph that, in turn, belongs to our desired class, which must occur on the nodes that were activated.

Assuming we can define an associated measure $\mu_i(c)$ onfor each $c \in \mathscr{C}_i$, we are able to define a probability distribution over subgraphs in the class.[6] Using notation similar to distributions over spanning trees in Zmigrod et al. [8]:

$$\begin{aligned} p_i(c) &= \frac{\mu_i(c)}{\mathbb{Z}_i} \\ \mathbb{Z}_i &= \sum_{c \in \mathscr{C}_i} \mu_i(c) \end{aligned} \tag{5.2}$$

This can be represented using the vectorization in Equation 3.9, due to the one-to-one correspondence established, so that, with a slight abuse of notation treating $\mathscr{C}_i$ as the parameter of distribution p_i:

$$\mathbf{r}_i(e) \,|\, \mathbf{x_i} \sim p_i(\mathscr{C}, E, V) \tag{5.3}$$

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. With $p_i(c)$, we also have a mechanism to sample them when a partition function is available (or able to be

[5] $\mathscr{P}(A)$ is the *powerset* of A, i.e. the set of all subsets of A.

[6] This is certainly not a trivial assumption, and might either be ill-defined or require techniques like the Gumbel trick[9] to approximate, unless the partition function $\mathbb{Z}$ has a closed form, or μ is already a probability measure on some σ-algebra over $\mathscr{C}$. Closed-form $\mathbb{Z}$ values on $\mathscr{C}$ are known for a handful of graph classes, such as the space of spanning trees, $\mathscr{C} = \mathscr{T}$. However, another way this might be accomplished is through random geometric graphs[CITE], or geometric spanners like Urqhart[CITE] and Relative Neighborhood [CITE] graphs on a "sprinkling" of points that preserves their observed pairwise distances.
This is further discussed in Section 10.3.

approximated).

With these mechanics in place, we see the choice of a clique (implied by the inner product) is a trivial application of Equation 5.3, since that is equivalent to selecting the class of cliques on V_i nodes. This has only one element ($\|\mathscr{C}_{\text{clique}}\| = 1$), there is only 1 possible selection, with probability $p_i(K^{V_i}) = 1$.

5.2.2 Graph Unions as Desire Paths

With a distribution over subgraphs each observation, we are potentially able to samb-ple from them for bootstrap or Monte Carlo estimation purposes, or simply find a maximum likelihood estimate for each distribution. Assuming this is true, we may now sample or approximate a matrix $R(i,e) : I \times E \to \mathbb{B}$.

Methods for doing this efficiently in certain cases are the focus of Section 6.3 and Section 7.3. However, once $R(i,e)$ is estimated, a reasonable mechanism for estimating the support of the set of edges is to use $\frac{\text{count}}{\text{exposure}}$, but with a few needed modifications.

First, while the nodes counts in $\sum_i B(i,\cdot)$ are by assumption *not* independent, or even pairwise independent, the *edge traversal* counts $\sum_i R(i,\cdot)$ could more reasonably be considered so. A model certainly could be constructed where edge existence depends on other edges existing (or not). But nothing is inherently self-inconsistent with a model that assumes the traversability of individual edges will be independent of one another.

let P(e) be the probability that an edge is traversed (in any observation), and P(u,v) the probability that nodes u, v co-occur. To approximate the overall traversability of an edge, we can calculate an empirical estimate for the conditional probabil-

ity $P(e|(u,v)) = \frac{P(e) \cap P(u,v)}{P(u,v)}$ that an edge is traversed, given that the two incident nodes are activated. This estimate can use the same beta-bernoulli distribution from Equation 2.2.

Still, how do we ensure our estimate is approximating traversability, so that the probability that an edge probability converges toward 1 as long as it *has to be possible* for e to be traversed? Recall from the introduction that, from a measurement perspective, the underlying networks rarely "exist" in the sense that we never truly find the underlying network, but only samples from (or caused by) it. Imagine that the "real" network is a set of paved sidewalks: our procedure is similar to watching people walk along paths between locations, and wanting to estimate which of the paths would be tread "enough" to be paved. This is where we build on an intuition based on the popular idea of *desire paths* which is a colloquial name for paths that form when individuals walk over grass or dirt enough to carve a trail. In network analysis and recovery, we usually are only allowed to see the desire paths that might have formed from our data, never the actual "pavement".

If presented with two equivalent paths, an individual is likely to choose the one that has been tread more often before i.e. the "more beaten" path. So, we don't want a probability that an edge has been traversed, but a probability over fractions of the time we expect an edge to have been traversed *more often than not*: how likely it is to "be beaten". This is accomplished by forcing $\alpha, \beta < 1$. For the case where $\alpha = \beta = 0.5$, we call this special case an *arcsine* distribution.

In a coin tossing game where each "heads" gives a point to player A and "tails" to player B, then the point differential is modeled as a random walk. The arcsine

distribution Beta($\frac{1}{2}, \frac{1}{2}$) is exactly the probability distribution for the fraction of time we expect one player to be "ahead" of the other. [15]

> "*Contrary to popular opinion, it is quite likely that in a long coin-tossing game one of the players remains practically the whole time on the winning side, the other on the losing side.*"
>
> William Feller[50, Chapter III]

This is desirable behavior for a distribution over edge traversability! We expect the vast majority of edges to have a 0 or 1 as the most likely values, with the expected fraction of observations that an edge being traversed was "ahead" of it being *not* traversed matching our empirical count. We generalize this with $\alpha = 1 - \beta$, with $\alpha + \beta = 1$, such that the new beta posterior from Equation 2.2 with s successes over k trials is:

$$\begin{aligned} \pi &\sim \text{Beta}(\alpha + c, 1 - \alpha - c) \\ c &= \frac{s - ak}{k + 1} \end{aligned} \tag{5.4}$$

Important to note is that k is measured over the *co-occurrences* (u, v), and successes are the traversals e derived from our samples in R. This lets us formulate a likelihood model for each edge's traversibility in the global network G (i.e. whether $B(e, v) > 0$ for any v), which is i.i.d. Bernoulli.

$$\pi_e \sim \text{Beta}(\alpha, 1 - \alpha), e \in E$$

$$E \sim \text{Bernoulli}(\pi_e), e \in E$$

This does not yet specify a likelihood for $\mathscr{C}_i$, because we have not included a mechanism for the down-selection to each $\mathbf{x}_i$. This will be addressed more completely for the special case of $\mathscr{C} = \mathscr{F}$, the set of spanning forests on a graph, in Section 7.2. In general, however, failing to specify the prior distribution on $\mathscr{C}_i$ does not necessarily make Equation 5.3 unusable, but necessitates an "empirical bayes" approach. With the marginals and co-occurrence counts for nonzero values in X, we can make a point estimate for each edge *given* a node subset, without needing to consider a prior distribution for each $\mathbf{x_i}$.

The nonparametric approach, in cases that merit the use of spanning trees, will be central to accurate, scalable estimation of B through our proposed method, Forest Pursuit.

Chapter 6: Approximate Recovery in Near-linear Time

In this chapter, we build on the notion of "Desire Path" estimation of a dependency network from node activations — sampling from a class of subgraphs constrained to active nodes, then merging them. We present *Forest Pursuit* (FP), a method that is scalable, trivially parallelizes, and offline-capable, while also outperforming commonly used algorithms across a battery of standardized tests and metrics. The key application for using FP is in domains where node activation can be reasonably modeled as arising due to *random walks*—or similar spreading process—on an underlying dependency graph.

First, we build an intuition for the use of trees as an unbiased estimator for desire path estimation when spreading processes are at work on the latent network. Then, the groundwork for FP is laid by combining sparse approximation through *matching pursuit* with a loss function modeled after the Chow Liu representation for joint probability distributions. The approximate complexity for FP is linear in the spreading rate of the modeled random walks, and linear in dataset size, while running in $O(1)$ time with respect to the network size. This departs dramatically from other methods in the space, all of which assume to scale in the number of nodes. We then test FP against an array of alternative methods (including GLASSO) with

MENDR, a newly-developed standard reference dataset and testbench for network recovery. FP outperforms other tested algorithms in nearly every case, and empirically confirms its complexity scaling for sub-thousand network sizes.

6.1 Random Walks as Spanning Forests

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 often modeled as a random-walk on the dependency graph: visiting a node leads to its activation.

6.1.1 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.[21] Further, the ordering of emissions has been removed in our binary vector representation, leaving only co-occurrence groups in each $\mathbf{x}_i$.[1] 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.

6.1.2 Activations in a Forest

As a general setting, the number of node activations (e.g. for datasets like co-authorship) is much smaller than the set of nodes ($\|\mathbf{x}_i \in \mathbb{S}^n\|_0 \ll n$)[2]

Avrachenkov et al. [17] go to some length describing discrete- and continuous-time random walk models that can give rise to binary activation vectors like our $X(i,j) : I \times J \to \mathbb{B}$. The *regularized laplacian* (or *forest*)kernel of a graph[13] plays a central role in their analysis, as it will in our discussion going forward.

$$Q_\beta = (I + \beta L)^{-1} \tag{6.1}$$

In that work, it is discussed as the optimal solution to the semi-supervised "node

[1] For a brief treatment of the case that INVITE emission order is preserved, see Chapter 9

[2] $\|\cdot\|_0$ is the ℓ_0 "pseudonorm", counting non-zero elements (the support) of its argument.

labeling" problem, having a regularization parameter β, though its uses go far beyond this.[19, 25, 38] Q generalizes the so-called "heat kernel" $\exp(-t\tilde{L})$, in the sense that it solves a lagrangian relaxation of a loss function based on the heat equation. This can be related to the PageRank ($\exp(-tL^{\text{rw}})$) kernel as well, which is explicitly based on random walk transition probabilities.

In fact, Q can be viewed as a transition matrix for a random walk having a geometrically distributed number of steps, giving us a small expected support for $\mathbf{x}_i$, as needed.[3] However we interpret Q, a remarkable fact emerges due to a theorem by Chebotarev: each entry $q = Q(u, v)$ is equal to the probability[4] that nodes u, v are connected in a randomly sampled *spanning rooted forest*

In other words, co-occurring node activations due to a random walk or heat kernel are deeply tied to the chance that those nodes find themselves *on the same tree in a forest*.

6.1.3 Spreading Dependencies as Trees

With the overt link from spreading processes to counts of trees made, there's room for a more intuitive bridge.

For single-cause, single source spreading process activations—on a graph—the activation dependency graph for each observation/spread/random walk *must be a tree*. With a single cause (the "root"), which is the starting position of a random walker, a node can only be reached (activated) by another currently activated node.

[3] Q can also be interpreted as a continuous-time random walk location probability, after exponentially distributed time, if spending exponentially-distributed time in each node.

[4] edge weights scaled by □

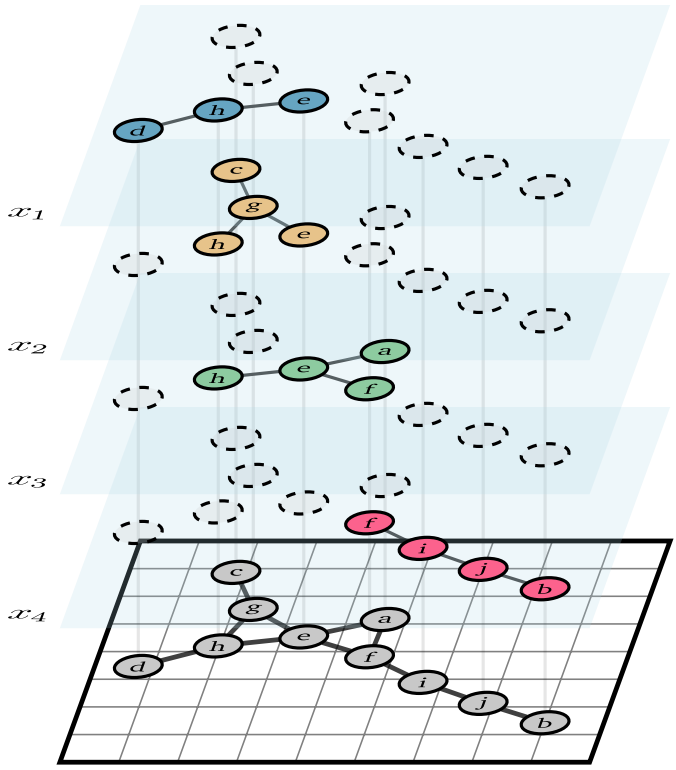


Figure 6.1: Edge Measurements with true (tree) dependencies known

If the random walk jumps from one visited node to another, previously visited one, that transition did not result in an activation , so the *dependency* count for that edge should not increase. This description of a random walk, where subsequent visits do not "store" the incoming transition, is roughly equivalent to one more commonly described as a *Loop-Erased* random walk. It is precisely used to uniformly sample the space of spanning trees on a graph.[43]

Much like a reluctant co-author "worn down" by multiple requests, we can even include random walks that "receive" activation potential from more than one source. Say a node is activated when some fraction of its neighbors have all originated a random walk transition to it, or a node activates on its second visit, or similar. We simply count (as dependency evidence) the ultimate transition that precipitated

activation. This could be justified from an empirical perspective as well: say we observe an author turn down requests for one paper from two individuals, but accept a third. We could actually infer a *lowered* dependency on the first two, *despite* the eventual coauthorship. Only the interaction that was observed as successful necessarily counts toward success-dependency, barring any contradicting information.

It's important to add here that *mutual convincing* by multiple collaborators simultaneously (or over time) is expressly left out. In other words, only pairwise interactions are permitted. This is not an additional assumption, but a key limitation of our use of graphs in the first place! As Torres et al. go to great lengths elaborating in [7], it is critical to correctly model dependencies when selecting a structural representation of our problem to avoid data loss. The possibility for multi-way interactions would necessitate the use of either a simplicial complex or a hypergraph as the carrier structure, *not a graph*.

Figure 6.1 demonstrates the use of trees as the distribution for subgraphs, instead of outer-products/cliques.

6.2 Sparse Approximation

As indicated previously, we desire a representation of each observation that takes the "node space" vectors ($\mathbf{x}_i$) to "edge space" ones ($\mathbf{r}_i$). We have separated each observation with the intention of finding a point-estimate for the "best" edges, such that the edge vector induces a subgraph belonging to a desired class. If we assume that each edge vector is in $\mathbb{B}^\omega$, so that the interactions are unweighted, undirected,

simple graphs, then for any family of subgraphs we will be selecting from at most $\omega \leq \binom{n}{2}$ edges.

Representing a vector as sparse combination of a known set of vectors (also known as "atoms") is called *sparse approximation*.

6.2.1 Problem Specification

Sparse approximation of a vector $\mathbf{x}$ as a representation $\mathbf{r}$ using a dictionary of atoms (columns of D) is specified more concretely as [34]:

$$\hat{\mathbf{r}} = \underset{\mathbf{r}}{\operatorname{argmin}} \|\mathbf{x} - D\mathbf{r}\|_2^2 \quad \text{s.t.} \|\mathbf{r}\|_0 \leq N \tag{6.2}$$

where N serves as a sparsity constraint (at most N non-zero entries). This is known to be NP-hard, though a number of efficient methods to approximate a solution are well-studies and widely used. Solving the lagrangian form of Equation 6.2, with an ℓ_1-norm in place of ℓ_0, is known as _Basis Pursuit[44], while greedily solving for the non-zeros of $\mathbf{r}$ one-at-a-time is called *matching pursuit*[45]. In that work, each iteration selects the atom with the largest inner product $\langle \mathbf{d}_{i'}, \mathbf{x} \rangle$.

We take an approach similar to this, but with the insight that the inner product will not result in desired sparsity (namely, a tree). Our dictionary in this case will be the set of edges given by B (see Section 5.2.1), while our sparsity is given by the relationship of the numbers of nodes and edges in a tree:

$$\hat{\mathbf{r}} = \underset{\mathbf{r}}{\operatorname{argmin}} \|\mathbf{x} - B^T\mathbf{r}\|_2^2 \quad \text{s.t.} \|\mathbf{r}\|_0 \leq \|\mathbf{x}\|_0 - 1 \tag{6.3}$$

There are some oddities to take into account here. As a linear operator (see Section 2.2), B^T takes a vector of edges to node-space, counting the number of edges each node was incident to. This means that, even with a ground-truth set of interactions, B^T would take them to a new matrix $X_{\text{deg}}(i,j) : I \times J \to \mathbb{N}$, which has entries of the number of interactions each individual in observation i was involved in. While very useful for downstream analysis (see Section 9.2.3), the MSE loss in Equation 6.4 will never be zero, since X_{deg} entries are not boolean. Large-degree "hub" nodes in the true graph would give a large residual, and the adjoint would subsequently fail to remove the effect of B^T on the edge vectors.

It might be possible to utilize a specific semiring, such as $(\min, +)$, to enforce inner products (see Section 2.4) that take us back to a binary vector.[5] Instead, we will take an empirical bayes approach to the estimation of sparse vectors.[36] As a probabilistic graphical model, we assume each observation is emitted from a (tree-structured) MRF [explained?] on the activated nodes. This is underdetermined (any spanning tree could equally emit the observed activations), so we use an empirical prior as a form of shrinkage: the co-occurrences of nodes across all observed activation patterns. This let's us optimize a likelihood from Equation 5.3, for the distribution of spanning trees on the subgraph of G^* inducted by $\mathbf{x}$.

$$\hat{\mathbf{r}} = \underset{\mathbf{r}}{\operatorname{argmax}} \mathscr{L}(\mathbf{r} \mid \mathbf{x}) \quad \text{s.t.} \quad \mathbf{r} \sim \mathscr{T}(G^*[\mathbf{x}]) \tag{6.4}$$

[5] This is more than a simple hack, and belies a great depth of possible connection to the problem at hand. It is known that "lines" (arising from equations of the inner product) in tropical projective space *are trees*.[39] In addition, the tropical equivalent to Kirchoff's polynomial (which counts over all possible spanning trees), is the direct computation of the minimum spanning tree.[5] For treatment of sparse approximation using tropical matrix factorization, see Omanović et al. [6]

6.2.2 Max. Spanning (Steiner) Trees

The point estimate $\hat{\mathbf{r}}$ is therefore the mode of a distribution over trees, which is precisely the maximum spanning tree.[8] If we allow the use of all observations X to find an empirical prior for $\mathbf{r}$, then we can calculate a value for the mutual information for the activated nodes, and use this to directly calculate the Chow-Liu estimate. One algorithm for finding a maximum spanning tree is Prim's[CITE?], which effectively performs the matching pursuit technique of greedily adding an edge (i.e. non-zero entry in our vector) one-by-one. In this way, we effectively *do* perform matching pursuit, but minimizing the KL-divergence between observed node activations and a tree-structured MRF limited to those nodes, alone (rather than the mean-square-error).

However, the mode of the tree distribution is not strictly the one that uses mutual information as edge weights. There is reason to believe that edge weights based on pairwise joint probabilities might also be appropriate. Namely, the Hunter-Worsley bound for unions of (dependent) variables says that the sum of marginal probabilities over-counts the true union of activations (including by dependence relations). This alone would be known as Boole's inequality, but the amount it overcounts is *at most* the weight of the maximum spanning tree over pairwise joint probabilities.[49] Adding the tree of joint co-occurrence probabilities is the most conservative way to arrive at the observed marginals from the probability of at least one node occurring (which could then be the "root").

Finally, we realize that the problem statement ("find the maximum weight tree

on the subgraph") is not the same as an MST, per-se, but rather the so-called "Steiner Tree" problem. In other words, we would like our tree of interactions to be of minimum weight on a node-induced subgraph *of the true graph*. The distribution of trees that our interactions are sampled from should be over the available edges in the recovered graph, *which we do not yet have*. Thankfully, a well-known algorithm for approximating the (graph) Steiner tree problem instead finds the minimum spanning tree over the *metric closure* of the graph.[47]

This metric closure is a complete graph having weights given by the shortest-path distances between nodes. While we don't know those exact values either, we do have the fact that the distance metric implied by the forest kernel (in Equation 6.1) is something of a relaxation of shortest paths. In the limit $\beta \to 0$, Q is proportional to shortest path distances, while $\beta \to \infty$ instead gives commute/resistance distances.[17] And that kernel is counting the probability of co-occurrence on trees in any random spanning forest!

All this is to say that node co-occurrence measures are more similar to node-node distances in the underlying graph, *not estimators of edge existence*. But we can use this as an empirical prior to approximate Steiner trees that *are on the true graph*.

6.3 Forest Pursuit

Instantiating the above, we propose *Forest Pursuit*, an relatively simple algorithm for correction of clique-bias under a spreading process assumption

6.3.1 Algorithm Summary

Once again, we assume m observations of activations over n nodes, represented as the design matrix $X : I \times J \to \mathbb{B}$. Like GLASSO, we assume that a Gram matrix (or re-scaling of it) is precomputed, for the non-streaming case.

Based on the discussion in Section 4.2.1 we will use the cosine similarity as a degree-corrected co-occurrence measure, with node-node distances estimated as $d_K = -\log \text{Ochiai}(j, j')$.[6]

For each observation, the provided distances serve to approximate the metric closure of the underlying subgraph induced by $\mathbf{x}$. This is passed to an algorithm for finding the minimum spanning tree. Given a metric closure, the MST in turn would be an approximation of the desired Steiner tree within $2 - \frac{2}{\|\mathbf{x}\|_0}$ of optimal.[47] For $\|\mathbf{x}\|_0 \ll n$, this error bound will often be close to 1, and approaching 2 as the expected number of activations-per-observation grows.

After the point estimates for $\mathbf{r}$ have been calculated as trees, we can use the desire-path beta-binomial model (Equation 5.4) to calculate the overall empirical Bayes estimate for $\hat{G}$. As a prior for α, instead of a Jeffrey's or Laplace prior, we bias the network toward maximal sparsity, while still retaining connectivity. In other words, we assume that n nodes only need about $n-1$ edges to be fully connected, which implies a prior expected sparsity of

$$\alpha^* = \frac{n-1}{\frac{1}{2}n(n-1)} = \frac{2}{n} \tag{6.5}$$

[6] Note that any kernel could be used, given other justification, though anectdotal evidence has the negative-log-Ochiai distance performing marginally better than MI distance or Yule's Q.

which we can use as a sparsity-promoting initial value for Beta($\alpha^*, 1-\alpha^*$).

Algorithm 6.1 Forest Pursuit

Require: $X \in \mathbb{B}^{m \times n}, d_K \in \mathbb{R}^{n \times n}_{\geq 0}, 0 < \alpha < 1$
Ensure: $R \in \mathbb{B}^{m \times \binom{n}{2}}$

procedure ForestPursuitEdgeProb(X, d_K, α)
  $R \leftarrow$ ForestPursuit(X, d_K)
  $\hat{\alpha}_m \leftarrow$ DesirePathBeta(X, R, α)
  return $\hat{\alpha}_m$
end procedure
procedure ForestPursuit(X, d_K)
  $R(\cdot, \cdot) \leftarrow 0$
  for $i \leftarrow 1, m$ **do** ▷ *each observation*
    $\mathbf{x}_i \leftarrow X(i, \cdot)$
    $R(i, \cdot) \leftarrow$ PursueTree($\mathbf{x}_i, d_K$)
  end for
  return R
end procedure
procedure PursueTree($\mathbf{x}, d$) ▷ *Approximate Steiner Tree*
  $V \leftarrow \{v \in \mathcal{V} \mid \mathbf{x}(\mathcal{V}) = 1\}$ ▷ *activated nodes*
  $T \leftarrow$ MST($d[V, V]$) ▷ *e.g. Prim's Algorithm*
  $\mathbf{u}, \mathbf{v} \leftarrow \{(j, j') \in J \times J \mid T(j, j') \neq 0\}$
  $\mathbf{r} \leftarrow e_n(\mathbf{u}, \mathbf{v})$ ▷ *unroll tree adjacency*
  return r
end procedure
procedure DesirePathBeta(X, R, α)
  $\mathbf{s} \leftarrow \sum_{i=1}^{m} R(i, e)$
  $\sigma \leftarrow \sum_{i=1}^{m} X(i, j)$
  $\mathbf{k} \leftarrow e_n(\sigma\sigma^T)$ ▷ *co-occurrence counts*
  $\hat{\alpha}_m \leftarrow \alpha + \frac{\mathbf{s} - \alpha\mathbf{k}}{\mathbf{k}+1}$
  return $\hat{\alpha}_m$
end procedure

Algorithm 6.1 outlines the algorithm in pseudocode for reproducibility.

6.3.2 Approximate Complexity

The Forest Pursuit calculation presented in Algorithm 6.1 assumes an initial value for the distance matrix, which is similar to the covariance estimate that is pre-computed

(as an initial guess) for GLASSO. Therefore we do not include the matrix multiplication for the gram matrix in our analysis, at least in the non-streaming case. Because every observation is dealt with completely independently, the FP estimation of R is linear in observation count. It is also trivially parallelizeable,[7] and the bayesian update for $\hat{\alpha}_m$ can be performed in a streaming manner, as well.

Each observation requires a call to "PursueTree", which involves an MST call for the pre-computed subset of distances onnodes activated for that observation. Note the use of MST algorithm could apply Prim's or Kr uskal's algorithm, or similar. It is recommended to utilize Prims in this case, however, since Prim's on a sufficiently dense graph can be made to run in $O(n)$ time for n activated nodes by using a d-tree for its heap queue.[CITE] Since we are always using the metric closure, Prim's will always run on a complete graph.

Importantly, this means that FP does not scale with the the size of the network, but only the worst-case activation count of a given observation, $O(s_{\max})$, where $s_{\max} = \max_i (\|X(i,\cdot)\|_0)$ We say this is *approximately* constant in node size:

- the total number of nodes is typically a *given* for a single problem setting
- in many domains, the basic *spreading* rate of diffusion model (e.g. R_0, or heat conductivity), does not scale with the total size of an observation

That last point means that constant scaling with network size is generally down to the domain in question. For instance, a heat equation simulated over a small area, having a given conductivity, will not have a different conductivity over a larger area;

[7] Although, no common python implementation of MST algorithms are as-yet vectorized or parallelized for simultaneous application over many observations. We see development of non-blocking MSTs in common analysis frameworks as important future work.

conductivity is a material property. Similarly, a virus might have a particular basic reproduction rate, or a set of authors might have a static distribution over how many collaborators they wish to work with. The former is down to viral load generation, and the latter a sociological limit: more a bigger department usually doesnt imply more co-authors.

Similar to Equation 6.5, we might reasonably assume that the expected degree of nodes is roughly constant with network size i.e. a property of domain. So, the the density of activation vectors (as a fraction of all possible edges) is going go scale with the inverse of n. This makes a process linear in activation count to be constant in network size. Then, if $\bar{s}$ is the expected non-zero count of each row of X, the final approximate complexity of FP is $O(m\bar{s})$.[8]

6.4 Simulation Study

To test the performance of FP against other backboning and recovery methods, we have developed a public repository `affinis` containing reference implementations for FP, along with many co-occurrence and backboning techniques. The library contains source code and examples for many of the presented methods, and more. [UPDATE w/ DOI?]

In addition, to support the community and provide for a standard set of benchmarks for network recovery from activations, the `MENDR` reference dataset and testbench was developed. To make reproducible comparison of recovery algorithms easier,

[8] In our reference implementation, which uses Kruskal's algorithm, the theoretical complexity is likewise $O(m\bar{s}^2 \log \bar{s})$, though in our experience the values of $\bar{s}$ are small enough to not impact the runtime significantly.

`MENDR` includes hundreds of randomly generated networks in several classes, along with random walks sampled *on those networks*. It can also be extended through community contribution, using data versioning to allow consistent comparison between different reports and publications over time.

6.4.1 Experimental Method

For each algorithm shown in Table 6.2, every combination of the parameters in Table 6.1 was tested. 30 random graphs for each of nodes were tested, which was repeated again for each of Tree, Block, and scale-free types.

Table 6.1: Experiment Settings (MENDR Dataset)

parameters	values
random graph **kind**	Tree, Block, BA($m \in \{1, 2\}$)
network n-**nodes**	10,30,100,300
random **walks**	1 sample $m \sim \text{NegBinomial}(2, n^{-1}) + 10$
random walk **jumps**	1 sample $n_s im \text{Geometric}(n^{-1}) + 5$
random walk **root**	1 sample $n_0 \sim \text{Multinomial}(\mathbf{n}^{-1}, 1)$
random **seed**	1, 2, … , 30

Table 6.2: Summary of algorithms compared

Algorithm	abbrev.	α?	class	source
Forest Pursuit	FP	Yes	hybrid	-

Algorithm	abbrev.	α?	class	source
GLASSO	GL	No	MRF	[28, 33]
Ochiai Coef.	CS	Yes	Counts	[46]
Hyperbolic Projection	HYP	No	Counts	[41]
Doubly-Stochastic	eOT	Yes	Transport	[23, 32]
High-Salience Skeleton	HSS	Yes	Transport	[27]
Resource Allocation	RP	Yes	Transport	[37]

For each algorithm that could be supplied a prior via additive smoothing, seen in Table 6.2 as "α?: Yes", a minimum connected (tree) level of sparsity wasy supplied $\alpha = \frac{2}{n}$. The others, esp. GLASSO, do not have a $\frac{\text{count}}{\text{exposure}}$ form, and could not be easily interpreted in a way that allowed for additive smoothing. However, since the regularizaiton parameter for GLASSO is often critical for finding good solutions, a 5-fold cross validation was performed for each experiment to select a "best" value, with the final result run using that value. While this does have a constant-time penalty for each experiment, the reconstruction accuracy is significantly improved with this technique, and would reflect common practice in using GLASSO for this reason.

The three classes of random graphs represent common use cases in sparse graph recovery. In addition, the block and tree graphs are types we expect GLASSO to correctly recover in this binary setting.[28] The block graphs of size n were formed by taking the line-graph of randomly generated trees of size $n+1$.

Trees were randomly generated using Prüfer sequences as impelmented in NetworkX [CITE]. To simulate possible social networks and other complex systems that show evidence of preferential attachment, scale-free graphs were sampled through the Barabási–Albert (BA) model, which was randomly seeded with a re-attachment paramenter m of either 1 or 2.

Every graph has a static ID provided by MENDR, along with generation and retrieval code for public review. New graphs kinds and sizes are simple to add for future benchmarking capability.

6.4.2 Metrics

To compare each algorithm consistently, several performance measures have been included in the MENDR testbench. They are all functions of the True Positive/Negative (TP/TN) and False Positive/Negative (FP/FN) values.

Precision (P) Fraction of positive predictions that are true, also called "positive predictive value" (PPV)

$$P = \frac{TP}{TP + FP}$$

Recall (R) Fraction of true values that were returned as positive. Also called the TP-rate (TPR), and has an inherent trade-off with precision.

$$R = \frac{TP}{TP + FN}$$

Matthews Correlation Coefficient (MCC) Balances all of TP,TN,FP,FN. Preferred

for class-imbalanced problems (like sparse recovery) [1]

$$\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Fowlkes-Mallows (F-M) Geometric mean of Precision and Recall, as opposed to the F-Measure that returns the harmonic mean. Also known to be the limit of the MCC as TN approaches infinity[2], which is useful as TN grows with n^2 but TP only with n.

$$\sqrt{P \cdot R}$$

Because this work is focused on unsupervised performance, specifically for the use of these algorithms by analysts investigating dependencies, we opt to calculate TP,TN,FP,FN at every unique edge probability/strength value returned by each algorithm. Then, because we do not know a priori which threshold level will be selected by an analyst in the unsupervised setting, we take a conservative approach and report the expected values E[MCC] and E[F-M] over all unique threshold values. To consistently compare the expected values, we transform the thresholds for every experiment to the range $[\epsilon, 1-\epsilon]$, to avoid division-by-0 at the extremes.

Another common approach is to report the Average Precision Score (APS). This is not the average precision over the thresholds however, but instead the expected precision over the possible recall values achievable by the algorithm. It is approximating

the integral under the parametric P-R curve, instead of the thresholds themselves.

$$\text{APS} = \sum_{e=1}^{\omega} P(e)(R(e) - R(e-1))$$

where $P(e)$ and $R(e)$ are the precision and recall at the threshold set by the edge e, in rank-order. This is more commonly done for supervised settings, however, and will report a high value as long as *any* threshold is able to return a both a high precision and a high recall, simultaneously.

6.4.3 Results - Scoring

The results over every experiment are shown in Figure 6.2. Only FP is able to report MCC and F-M values with medians over about 0.5, regularly reaching over 0.8. GLASSO is clearly the second-best at recovery in these experiments, though for scale-free networks the improvement over simply thresholding the Ochiai coefficient is negligible. For APS, both GLASSO and Ochiai are equally able to return high scores, indicating at least one threshold for each that performed well. A simple mechanism for FP to perform equally well at APS is discussed in Section 6.4.5.

Breaking down the results by graph kind in Figure 6.3, we see the remarkable ability of FP to dramatically outperform every other algorithm in MCC and F-M, showing remarkable accuracy *together with stability* over the set of threshold values. This is indicative of FP's ability to more directly estimate the support of each edge, with lower values occurring only when co-occurrences aren't being consistently explained with the same set of incidences.

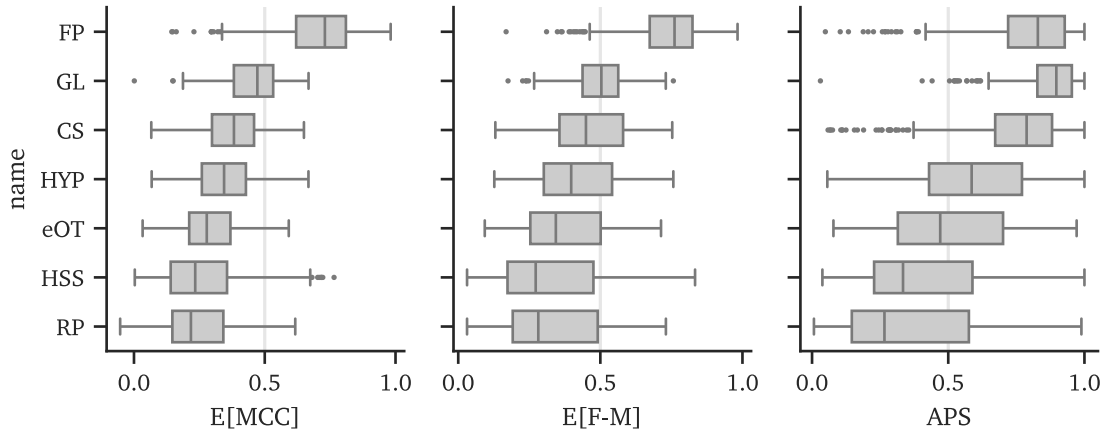


Figure 6.2: Comparison of MENDR recovery scores

Another important capability of any recovery algorithm is to improve its estimate when provided with more data. Of course, this also will depend on other factors, such as the dimensionality of the problem (network size), and specifically for us, whether *longer* random walks makes network inference better or worse.

As Figure 6.4 shows, FP is positively correlated with *all three*. Most importantly, the trend for FP is strongest as the number of observations increases, which is not a phenomenon seen in the other methods. In fact, it appears that count-based methods' scores are negatively correlated with added random walk length and added observations. Only HYP and CS scores are shown in Figure 6.4, but all other tested methods (other than FP and GLASSO) show the same trend.

However, because the graph sampling protocol includes n in the distributions for the observation count and random-walk length, we additionally performed a linear regression on the (log) parameters. The *partial resitual* plots are shown in Figure 6.5, which shows the trends of each variable after controlling for the others.

This analysis indicates that *all* methods should likely increase in their performance

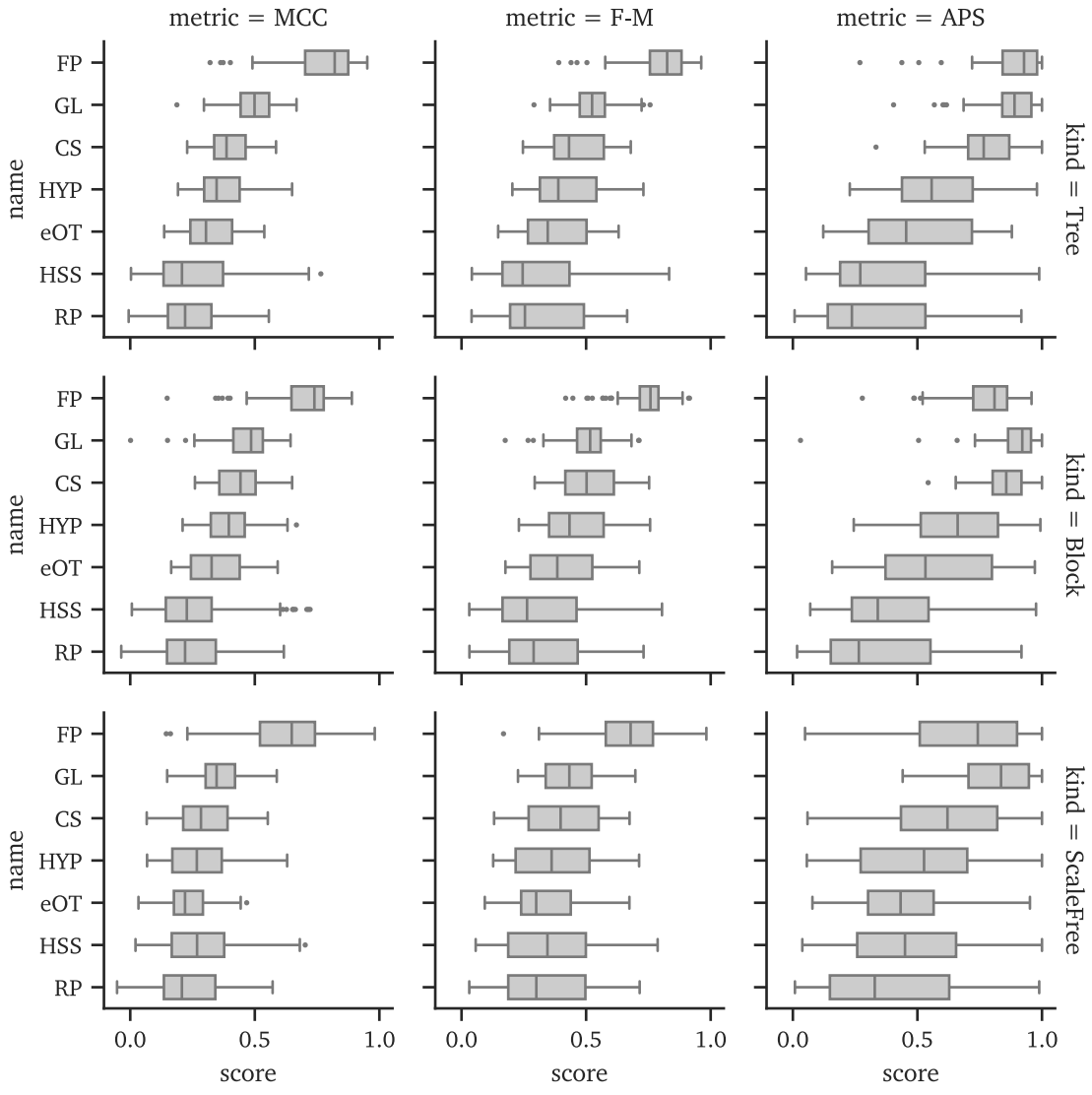
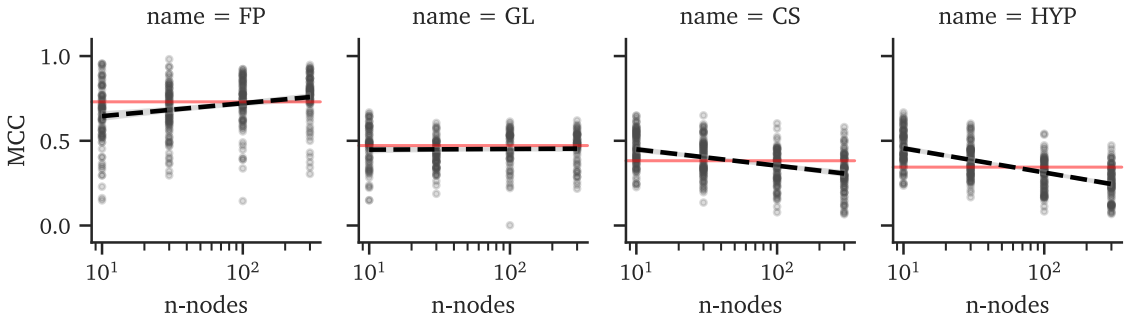
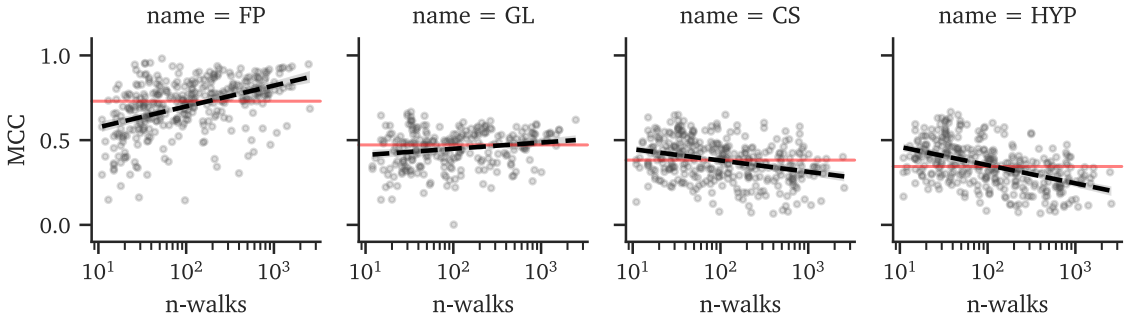


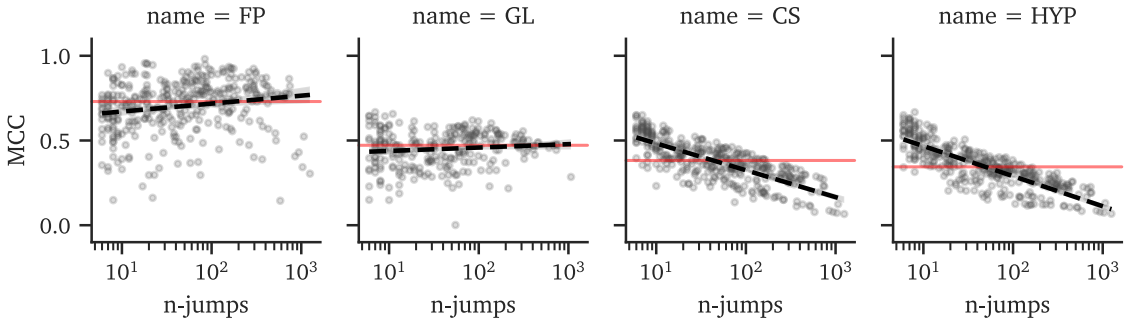
Figure 6.3: Comparison of MENDR Recovery Scores by Graph Type



(a) Trend: MCC vs network size



(b) Trend: MCC vs observation count



(c) Trend: MCC vs random-walk length

Figure 6.4: Score trends vs problem scaling

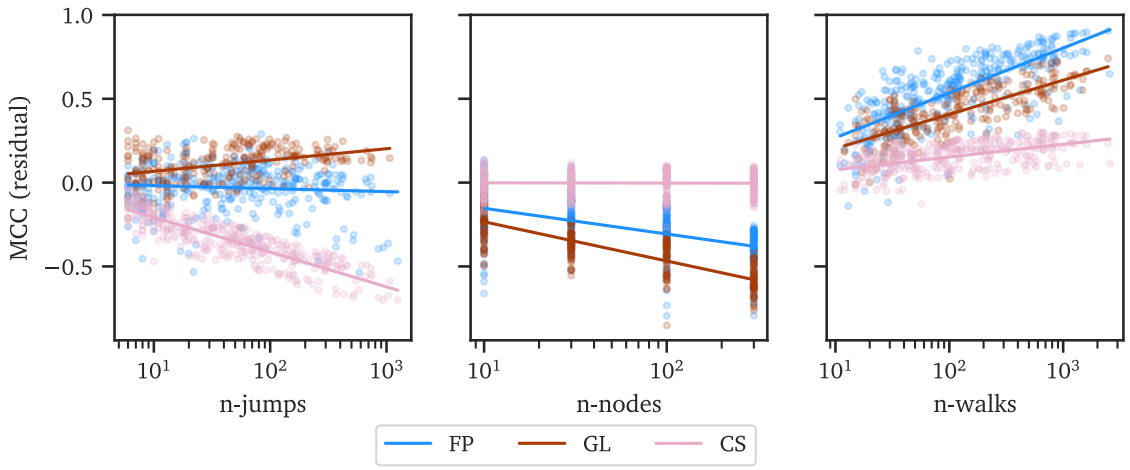


Figure 6.5: Partial Residuals (regression on E[MCC])

when extra observations are added, though FP does this more efficiently than either CS or GLASSO. Interestingly, CS is largely unaffected by network size, compared to FP and GL, though GL performs the worst in this regard. However, it is in the random-walk length that we see the benefit of dependency-based algorithms. The Ochiai coefficient suffers dramatically as more nodes are activated by the spreading process, since this means the implied clique size grows by the square of the number of activations. FP remains unaffected by walk-length, while (impressively) GLASSO appears to have a marginal boost in performance when walk lengths are high.

6.4.4 Results - Runtime Performance

For both Forest Pursuit and GLASSO, runtime efficiency is critical if these algorithms are going to be adopted by analysts for backboning and recovery. Figure 6.6 shows the (log-)seconds against the same parameters from before. For similar sized networks, FP is consistently taking 10-100x less time to reach a result than GLASSO does. Additionally, many of the experiments led to ill-conditioned matrices that failed

to converge for GLASSO under any of the regularization parameters tests (the “x” markers in Figure 6.6). As expected, the number of observations plot shows a clear limit in terms of controlling the lower-bound of FPs runtime, since in this serial version every observation runs one more call to MST. On the other hand, GLASSO appears to have signiicant banding for walk length and observation counts, likely indicating dominance of network size for its runtime.

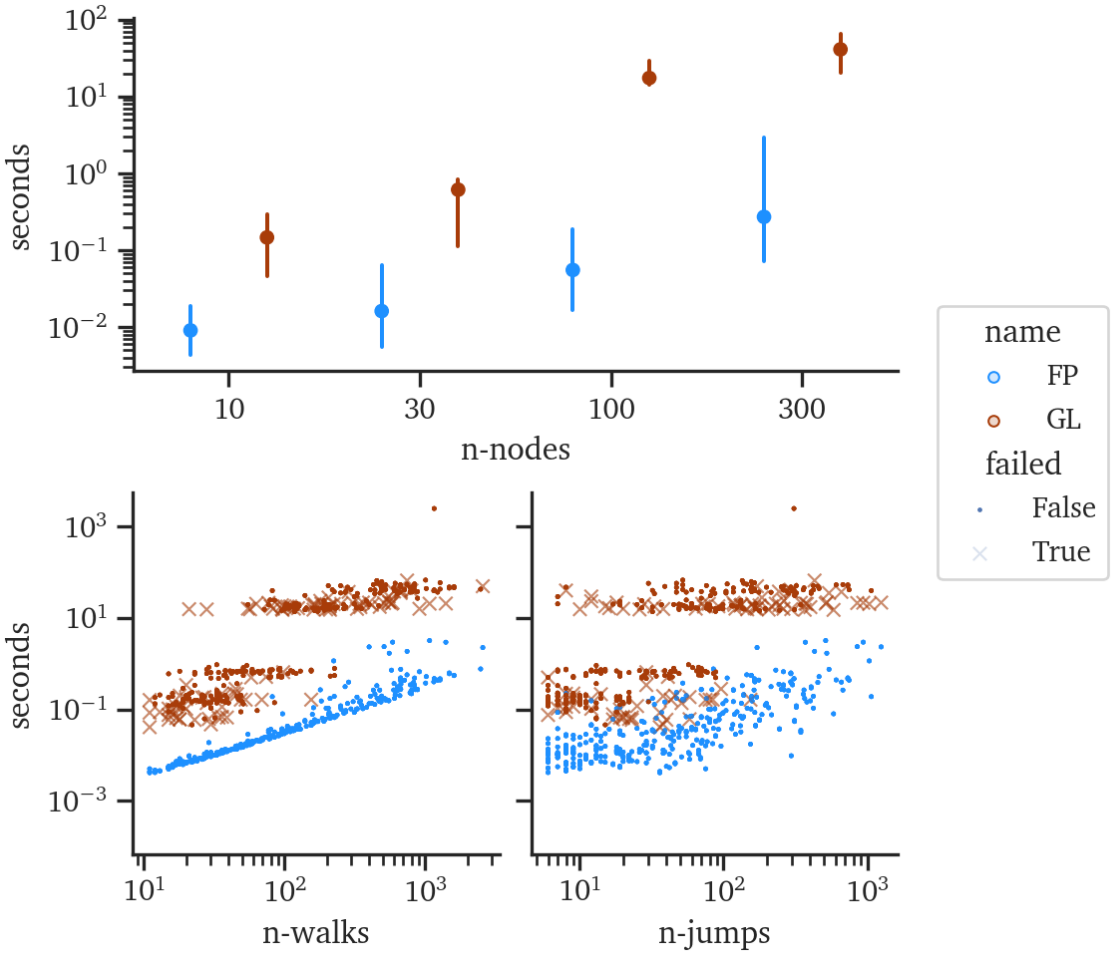


Figure 6.6: Runtime Scaling (Forest-Pursuit vs GLASSO)

To control for each of the variables, and to empirically validate the theoretical analysis in Section 6.3.2}, a regression of the same three (log-)parametwers was

performed against (log-)seconds. The slopes in Figure 6.7, which are plotted on a log-log scale, correspond roughly to polynomial powers in linear scale. In regression terms, we are fitting the log of

$$y_{\text{sec}} = a x_{\text{param}}^{\gamma}$$

so that the slope in a log-log plot is γ.

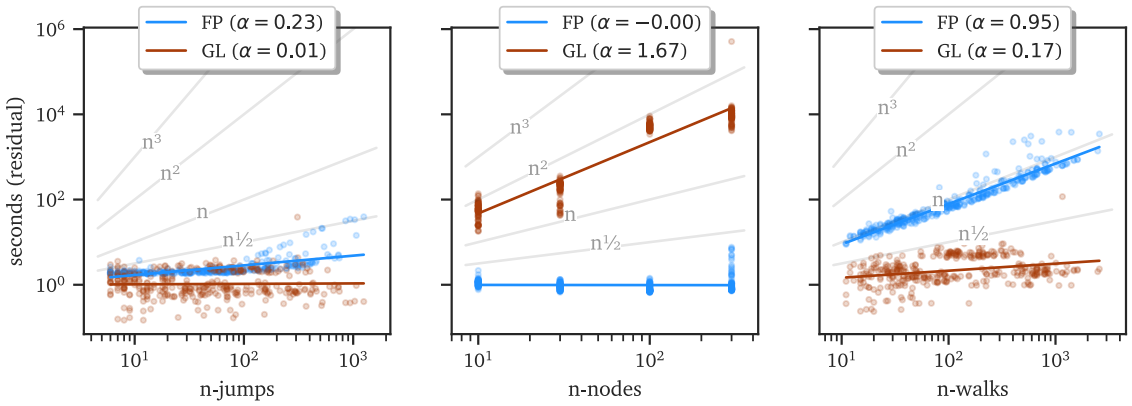


Figure 6.7: Partial Residuals (regression on computation time)

In a very close match to our previous analysis, the scaling of FP is almost entirely explained by the observation count and random-walk length, alone: the coefficient on network size shows constant-time scaling. Similarly, the scaling with observation count is very nearly linear time, as predicted. The residuals show non-linear behavior for the random-walk length parameter, which would make sense, due to the theoretical $||E|| \log ||V||$ scaling of kruskal's algorithm. At this scale, $n \log n$ and $n^2 \log n$ complexity might appear smaller than linear time, due to the log factor. GLASSO hardly scales with random walk length, and only marginally with observation count. In typical GLASSO, the observation count has already been collapsed to calculate

the empirical covariance matrix, so its effects here might be due instead to the cross-validation and the need to calculate empirical covariance for observation subsets. The big difference, however, is GLASSO scaling in significantly superlinear time—almost $O(n^2)$. This is usually the limiting factor for analyst use of such an algorithm in network analysis more generally.

6.4.5 Interaction Probability

Without using stability selection[31], GLASSO is not directly estimating the "support" of the edge matrix, but the strength of each edge. to do similar with FP, we could directly estimate the frequency of edge occurrence using $R(i, e)$ marginal averages, rather than conditioning on co-occurrence. Simply multiplying each FP edge probability by the co-occurrence probability of each node-node pair gives this as well, which we call FPi: the direct "interaction probability" for each pair of nodes.

By doing this simple re-weighting, FPi actually beats GLASSO's median APS for the dataset, but at the cost of MCC and F-M scores (which both drop to between FP and GLASSO), as Figure 6.8 demonstrates. Similarly, the individual breakdown by graph kind in Table 6.3 shows a similar pattern, with FPi coming close to GLASSO for scale-free networks, but exceeding it for trees and matching for block graphs. Still, the difference is small enough, and at such a significant penaly to MCC and F-M scores over a variety of thresholds, that it is hard to recommend the FPi re-weighting unless rate-based edge analysis is desired, e.g. if Poisson or Exponential occurrence models are desired.

To illustrate what is going on, we have selected two specific experiments as a

Table 6.3: Comparing scores for FP, FPi and GLASSO

	FP	FPi	GL
Block			
APS	0.78	0.9	0.9
F-M	0.74	0.57	0.51
MCC	0.7	0.55	0.46
ScaleFree			
APS	0.69	0.76	0.81
F-M	0.67	0.46	0.44
MCC	0.63	0.43	0.36
Tree			
APS	0.9	0.92	0.88
F-M	0.81	0.66	0.53
MCC	0.78	0.65	0.49

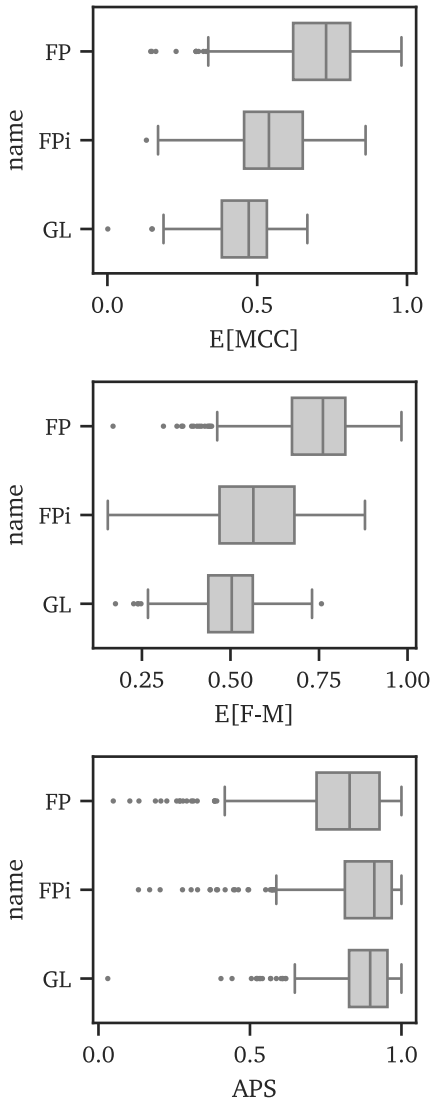


Figure 6.8: FPi shows best APS, lower MCC,F-M

case study, in Figure 6.9 In the first, `BL-N030S01`, a 30-node block graph with 53 random walk samples, has FP performing worse than GLASSO and Ochiai, in terms of APS (which is reported in the legends). We see that FP shows high precision, which drops off significantly to increase recall at all. Only a few edges had high probability (which is usually desirable for sparse approximation), and some of the true edges were missed this way. However, FPi rescaling makes rarer edges fall off earlier in the thresholding, letting the recall rise by dropping rare edges, rather than simply the low-confidence ones.

In the second, `SC-N300S01` is a 300-node scale-free network with 281 walks. Both FP and FPi show significantly better recovery capability, since enough walks have visited a variety of nodes to give FP better edge coverage. In this graph, no algorithm comes within 0.25 of FP's impressive 0.88 APS for 300 nodes fewer than that many walks.

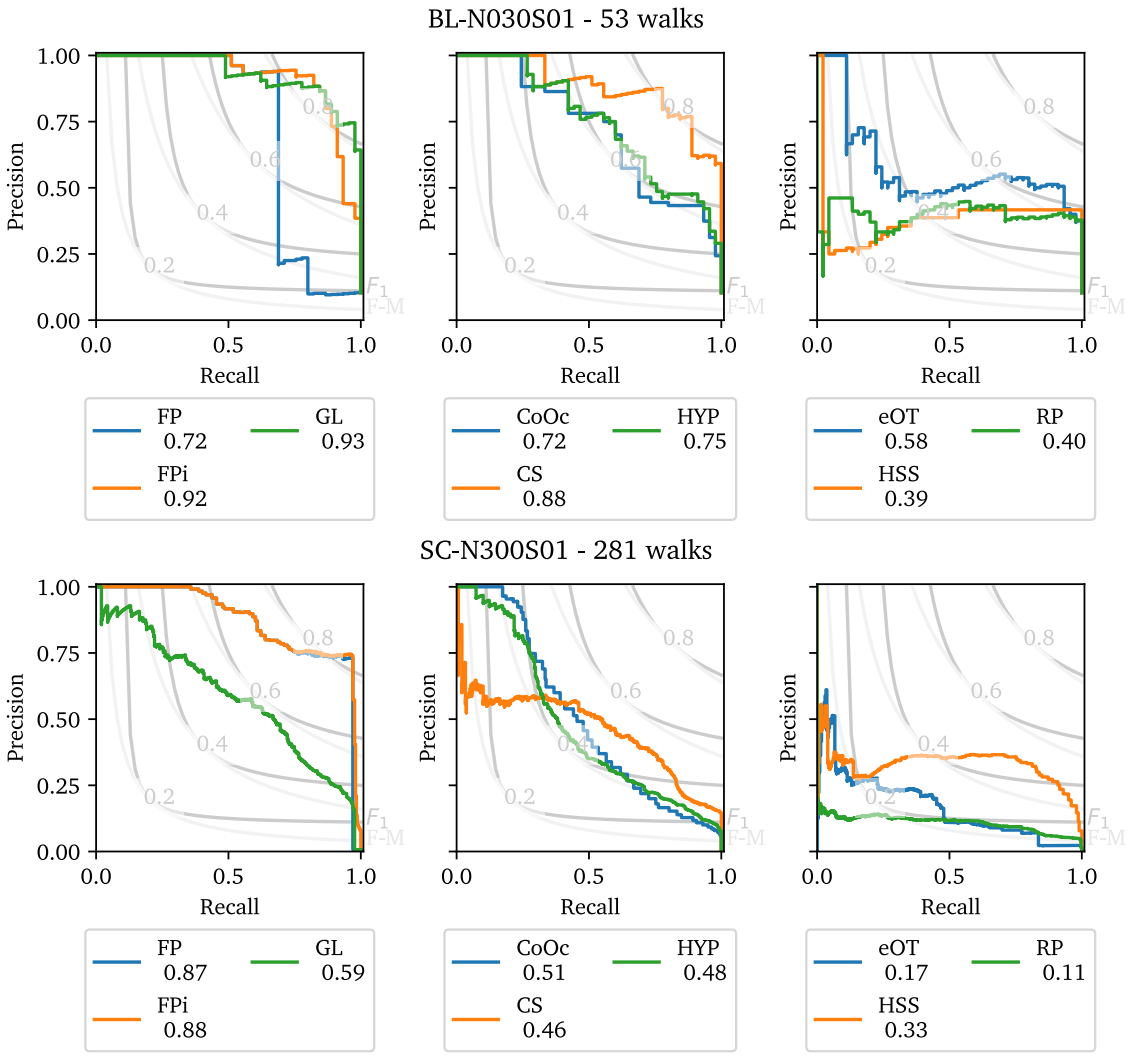


Figure 6.9: P-R curves for two experiments

Chapter 7: LFA: Latent Forest Allocation

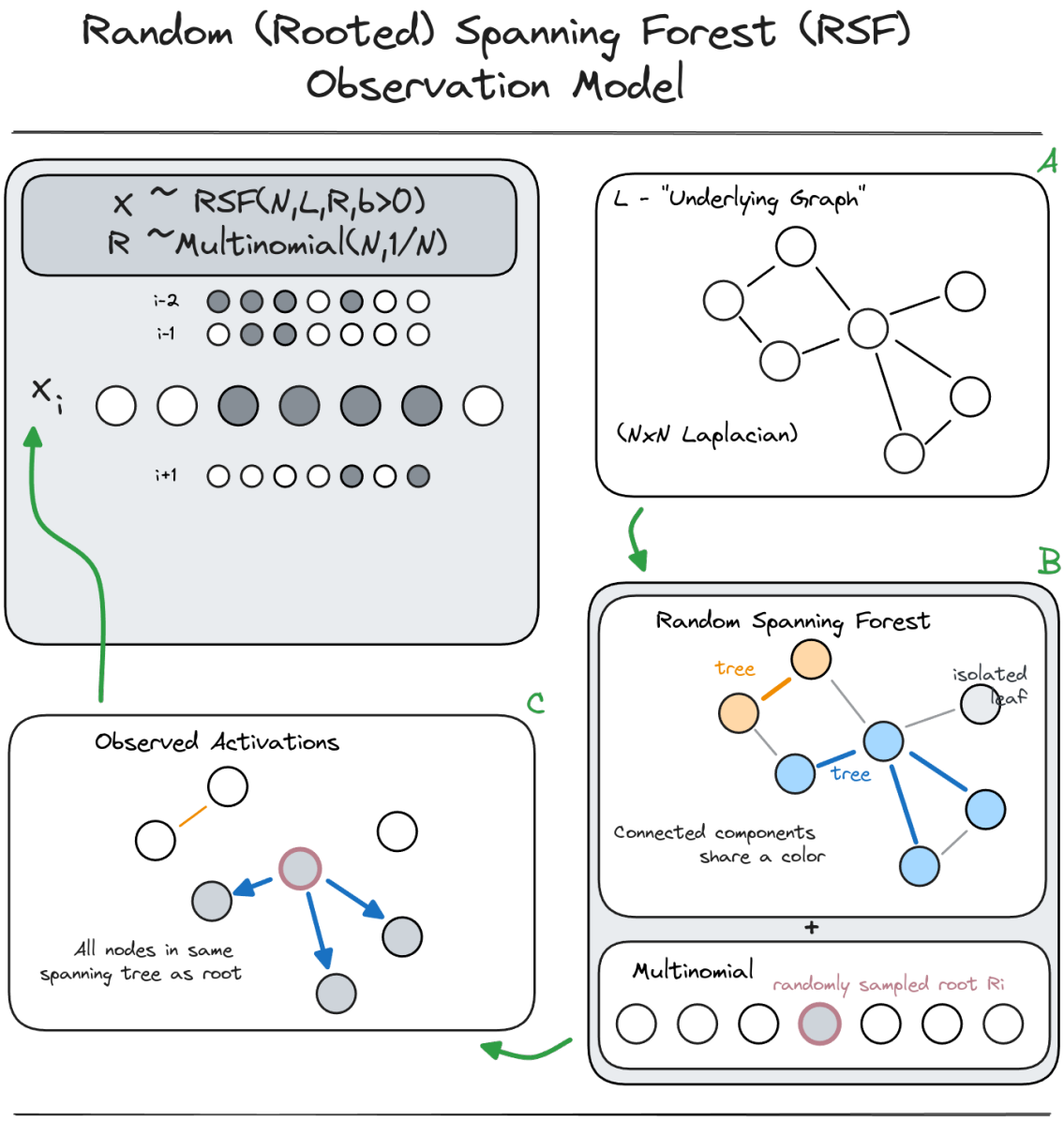
7.1 LFA as Factorization

7.1.1 Dictionary Learning

7.1.2 Interaction Degree, not Binary

- “plan” from avrachenkov

7.2 Generative Model Specification



- hierarchical model - marginalize over the root node.

7.3 Bayesian Estimation by Gibbs Sampling

7.3.1 Uniform Random Spanning Trees

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

7.3.2 Approximate Forest Samples

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

  I.e. the unwritten paper, modifying technique by Duan and Dunson [4] for RSF instead of RSTs

7.4 Expected Forest Maximization

7.4.1 Alternating Directions

- estimate laplacian to get Q_i as shortest path distance

7.4.2 degree normalization

sym laplacian before forest kernel

7.4.3 algorithm overview

7.5 Simulation Study

7.5.1 Score Improvement

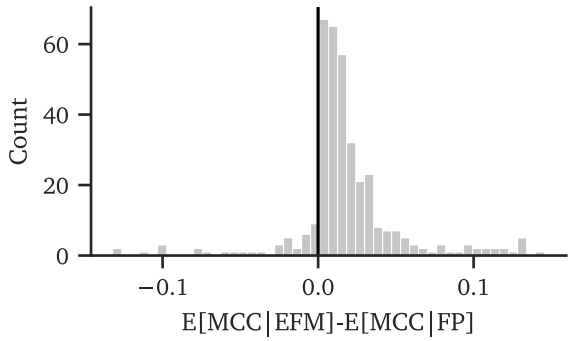


Figure 7.1: Change in Expected MCC (EFM vs FP)

7.5.2 Odds of Individual Edge Improvement

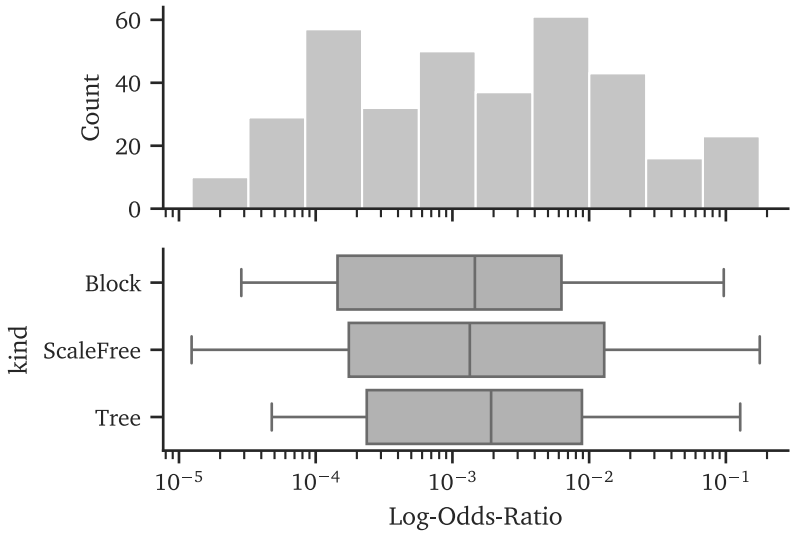


Figure 7.2: Logistic Regression Coef. (EFM - FP) vs. (Ground Truth)

7.5.3 Runtime Cost

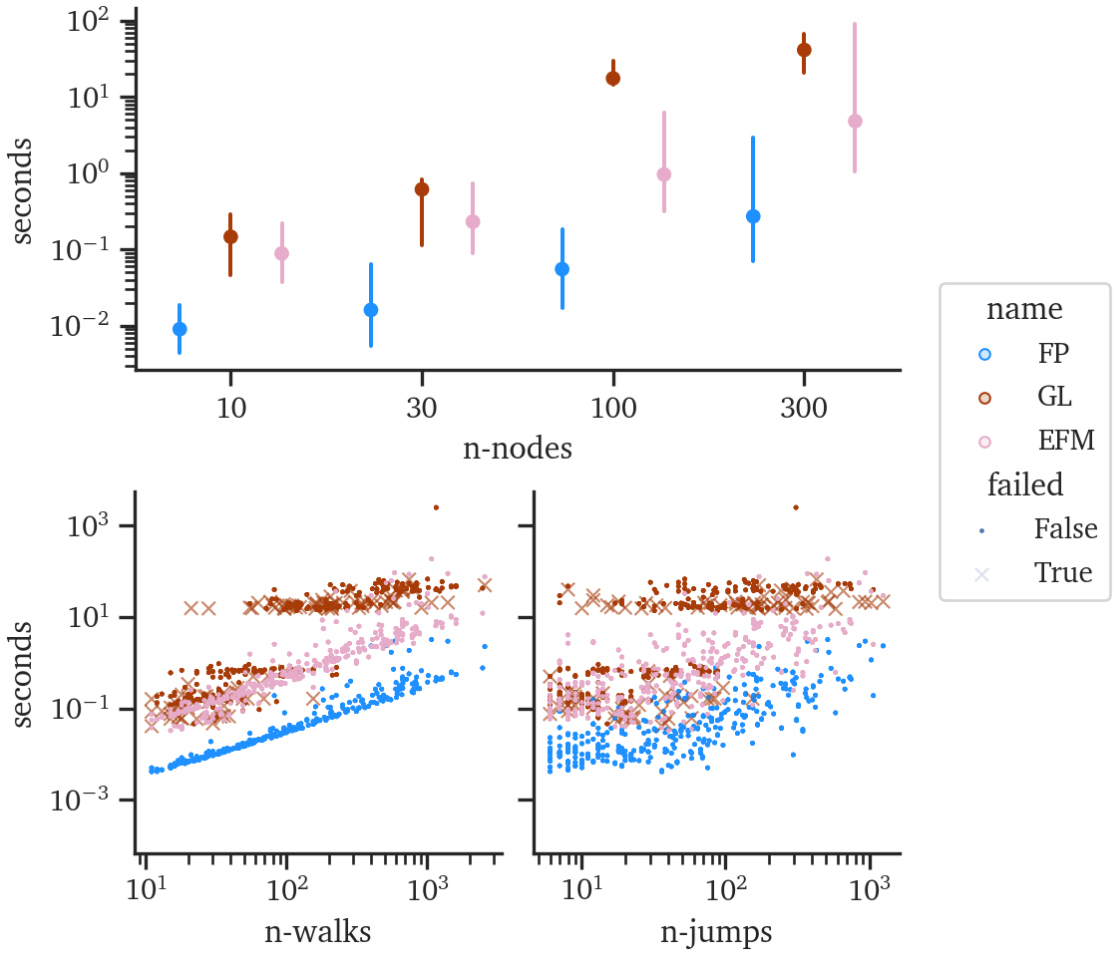


Figure 7.3: Runtime Scaling (Forest-Pursuit vs GLASSO)

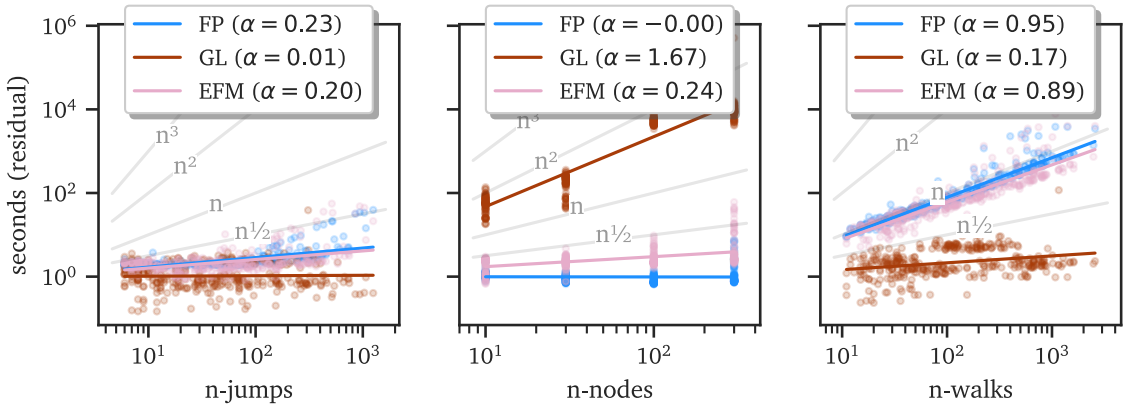


Figure 7.4: Partial Residuals (regression on computation time)

Part III

Applications & Extentions

Chapter 8: Qualitative Application of Relationship Recovery

8.1 Network Science Collaboration Network

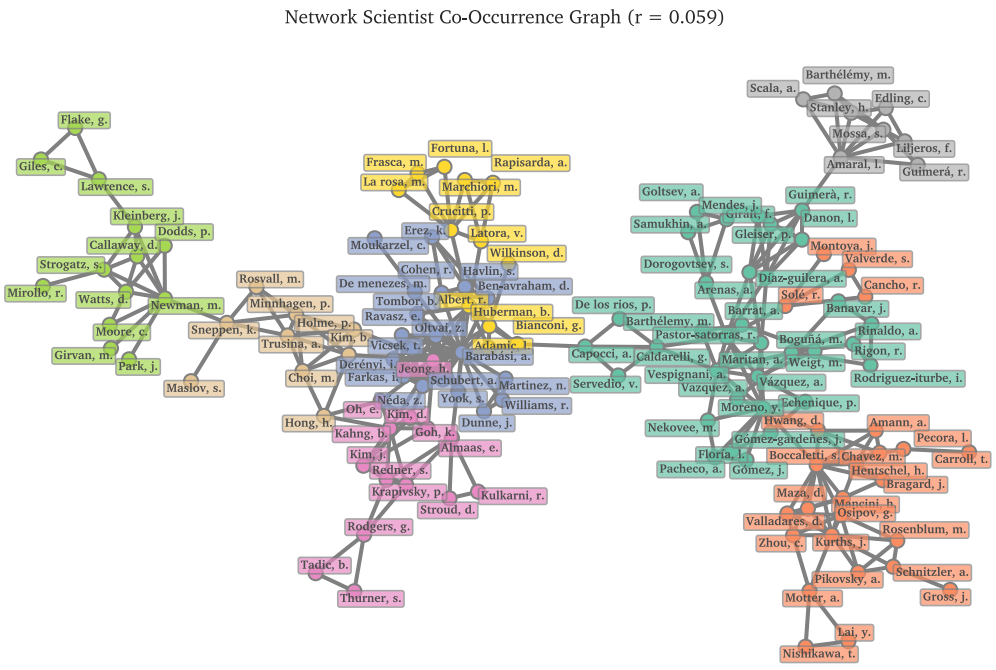


Figure 8.1: 134 Network scientists from [NEWMAN;BOCCALETTI;SNEPPEN], connected by co-authorship

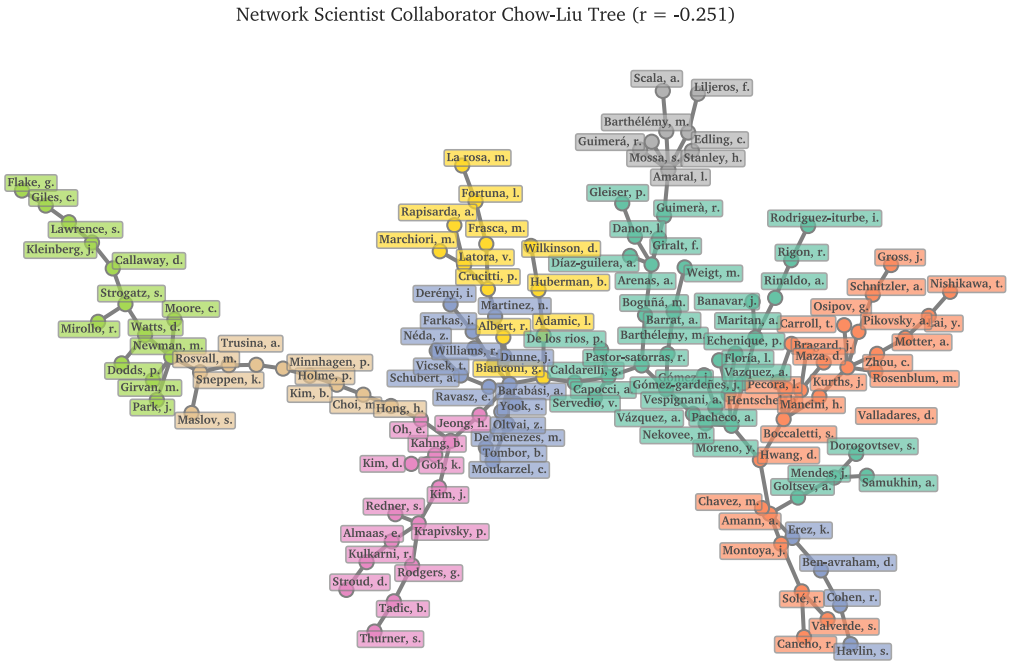


Figure 8.2: Max. likelihood tree dependency structure to explain co-authorships

8.2 Les Miserables Character Network

8.2.1 Backboning

8.2.2 Character Importance Estimation

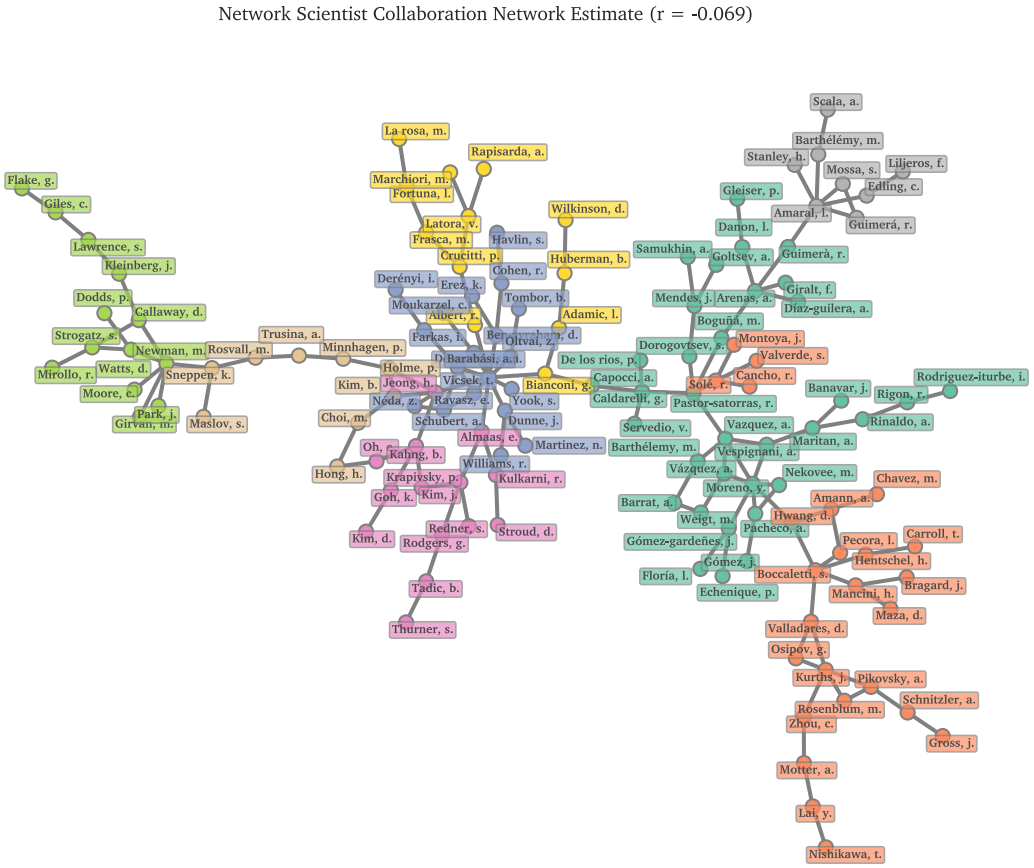


Figure 8.3: Forest Pursuit estimate of NetSci collaborator dependency relationships

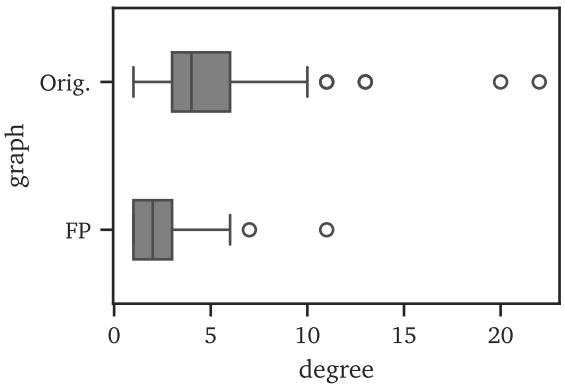


Figure 8.4

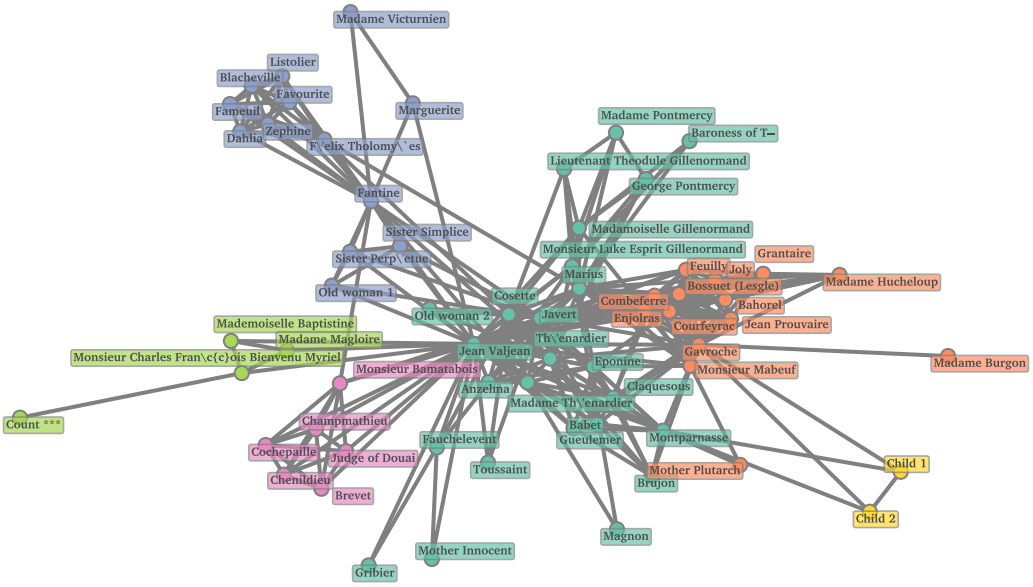


Figure 8.5

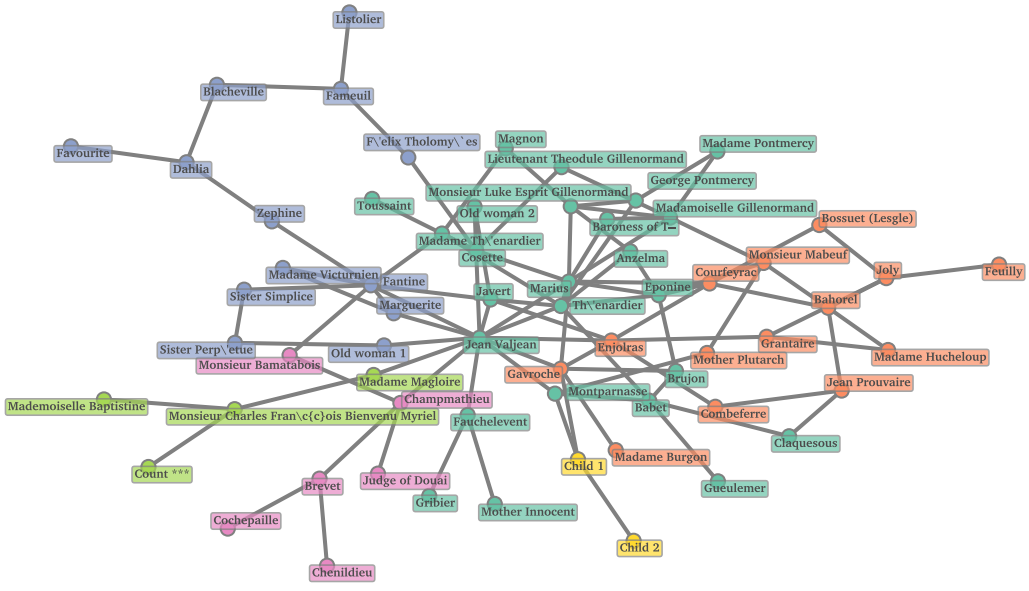


Figure 8.6

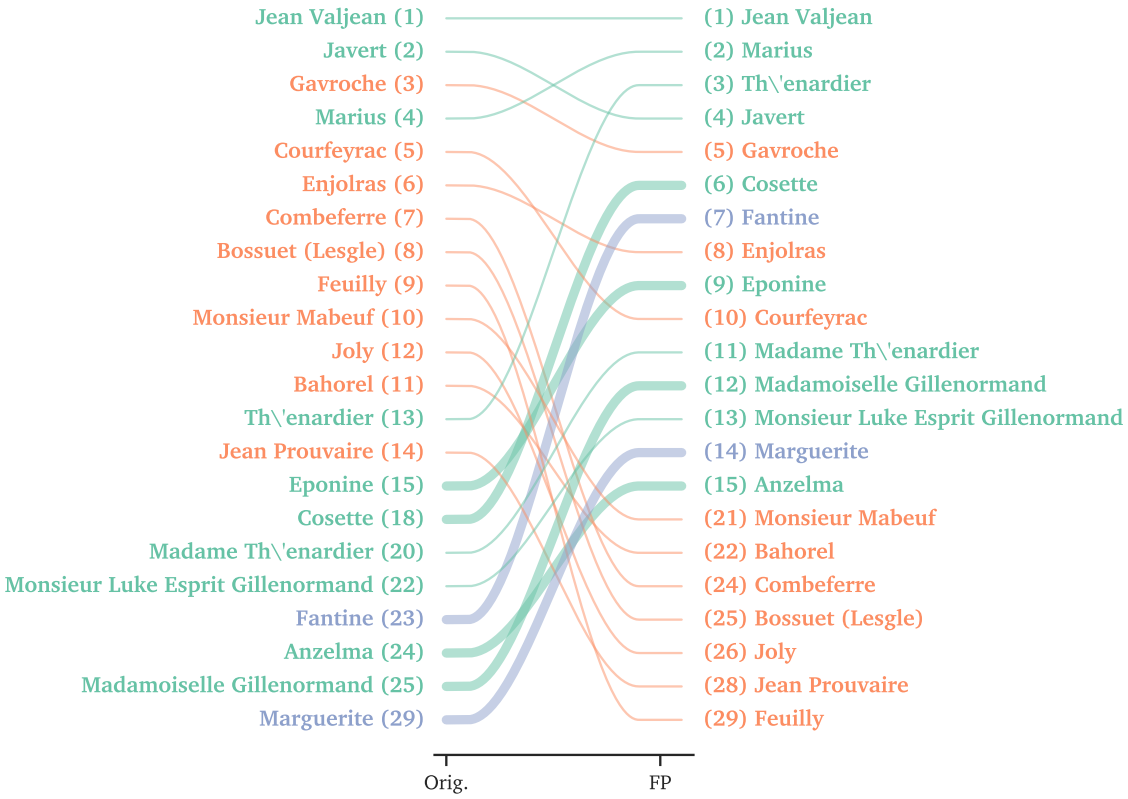

Jean Valjean (1)
Javert (2)
Gavroche (3)
Marius (4)
Courfeyrac (5)
Enjolras (6)
Combeferre (7)
Bossuet (Lesgle) (8)
Feuilly (9)
Monsieur Mabeuf (10)
Joly (12)
Bahorel (11)
Th\'enardier (13)
Jean Prouvaire (14)
Eponine (15)
Cosette (18)
Madame Th\'enardier (20)
Monsieur Luke Esprit Gillenormand (22)
Fantine (23)
Anzelma (24)
Madamoiselle Gillenormand (25)
Marguerite (29)
(1) Jean Valjean
(2) Marius
(3) Th\'enardier
(4) Javert
(5) Gavroche
(6) Cosette
(7) Fantine
(8) Enjolras
(9) Eponine
(10) Courfeyrac
(11) Madame Th\'enardier
(12) Madamoiselle Gillenormand
(13) Monsieur Luke Esprit Gillenormand
(14) Marguerite
(15) Anzelma
(21) Monsieur Mabeuf
(22) Bahorel
(24) Combeferre
(25) Bossuet (Lesgle)
(26) Joly
(28) Jean Prouvaire
(29) Feuilly
Orig.
FP

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 [10, 14]

9.2 Verbal Fluency Animal Network

9.2.1 Edge Connective Effiency 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.

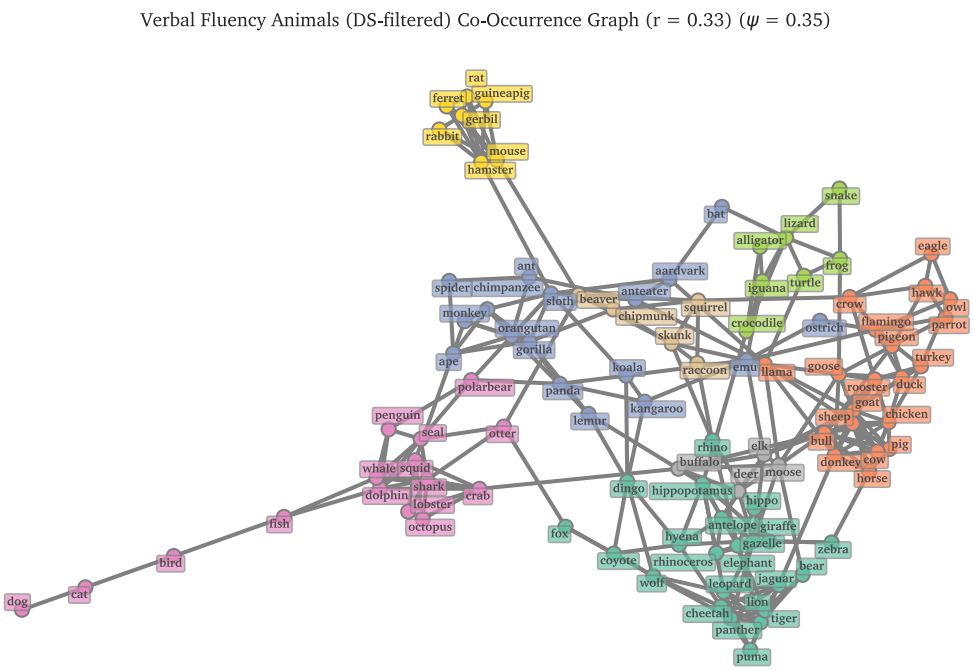


Figure 9.1

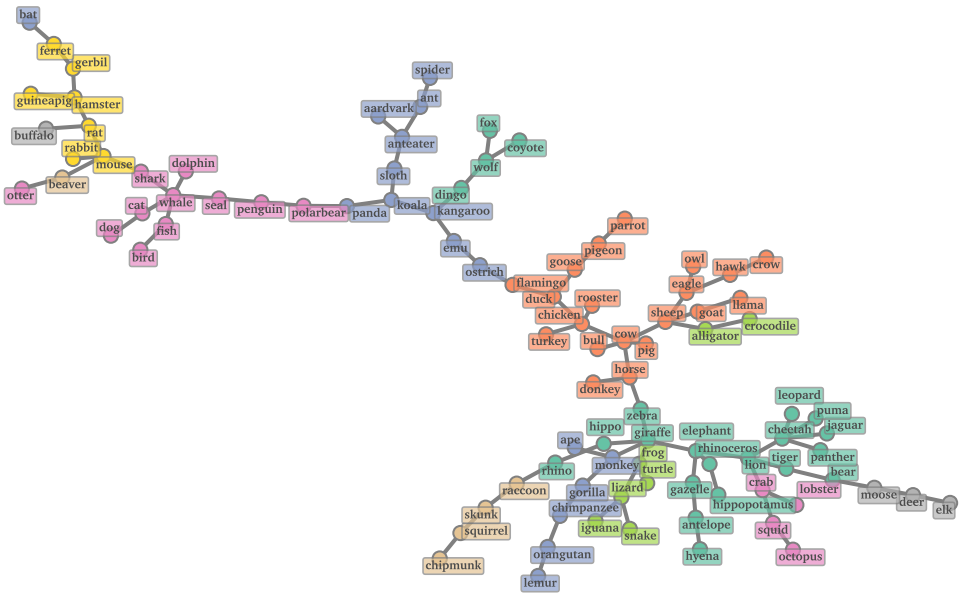


Figure 9.2

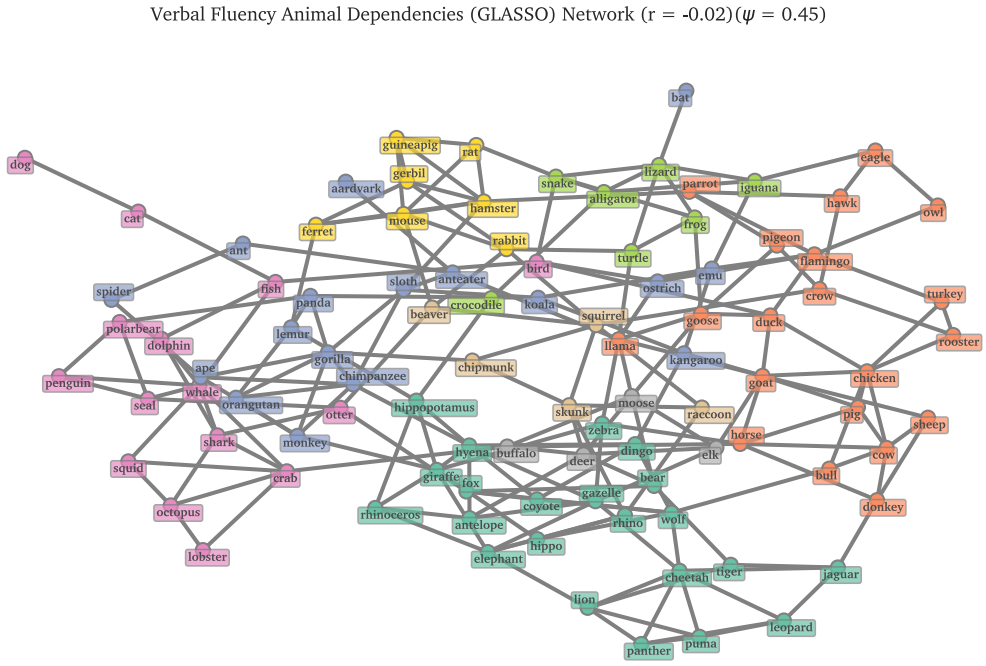


Figure 9.3

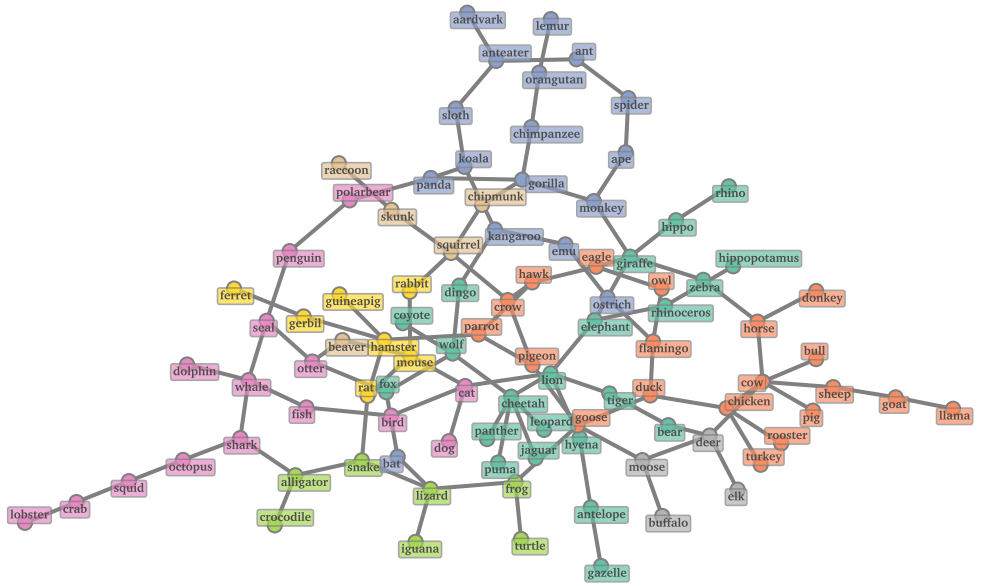
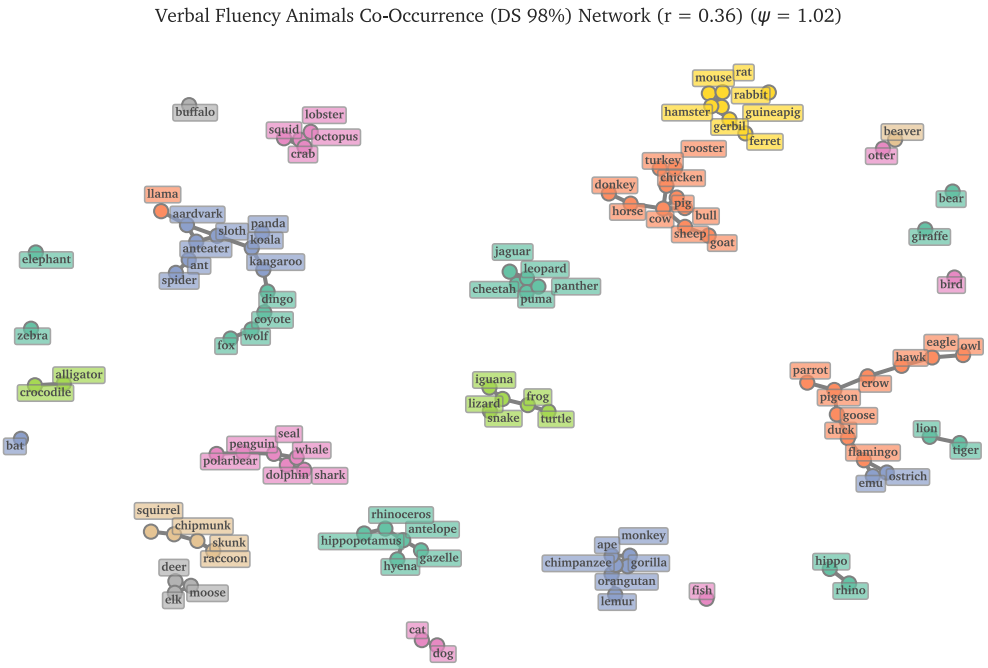
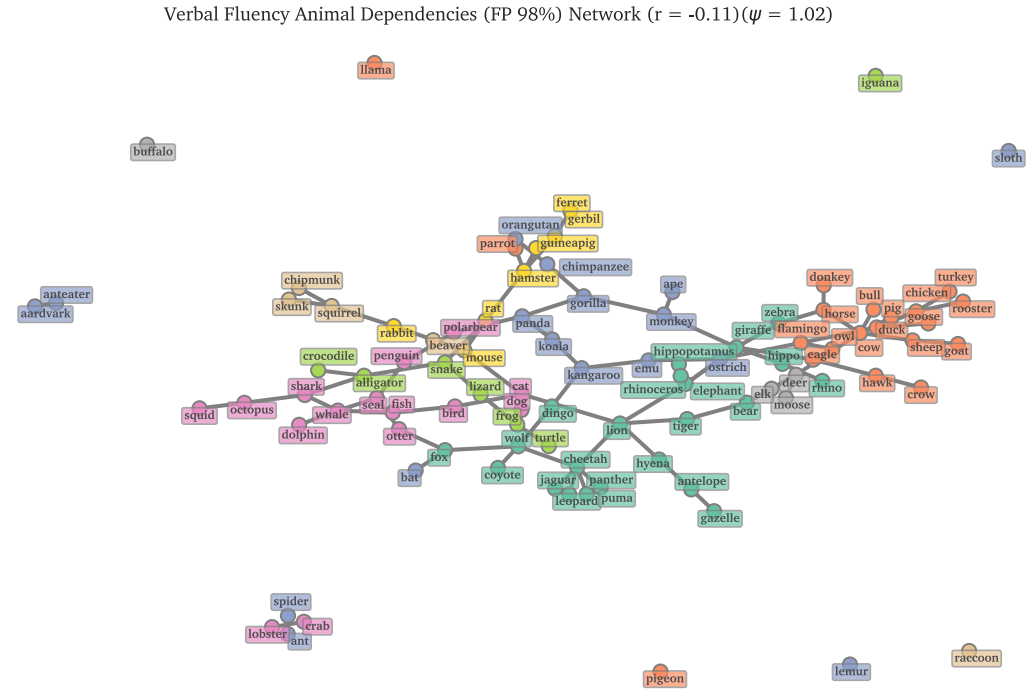


Figure 9.4: Comparison of backboning/dependency recovery methods tested vs. Forest Pursuit

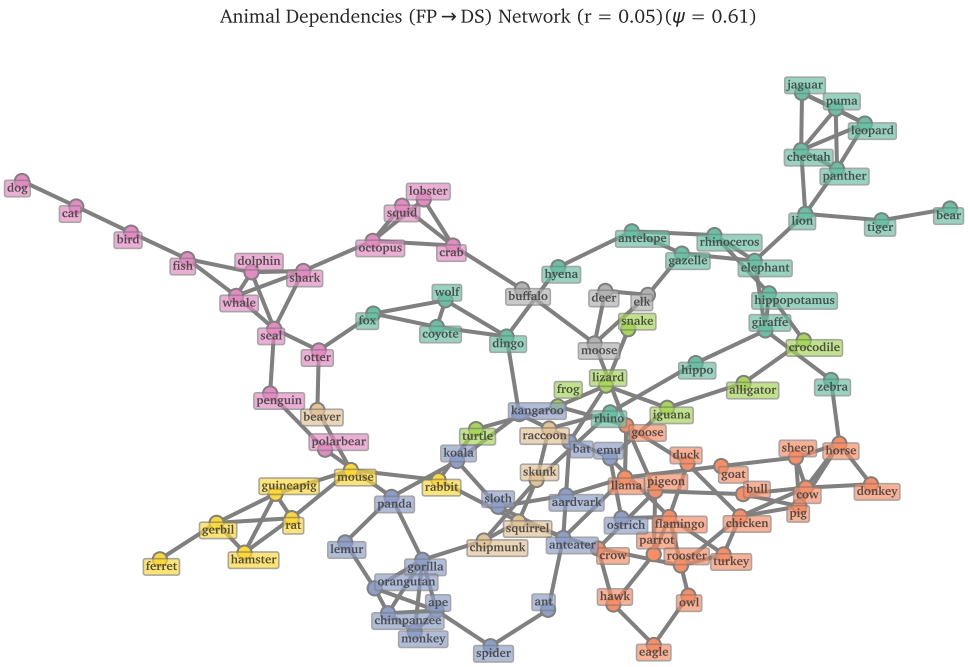


(a) co-occurrence methods will retain local communities at the cost of global structure

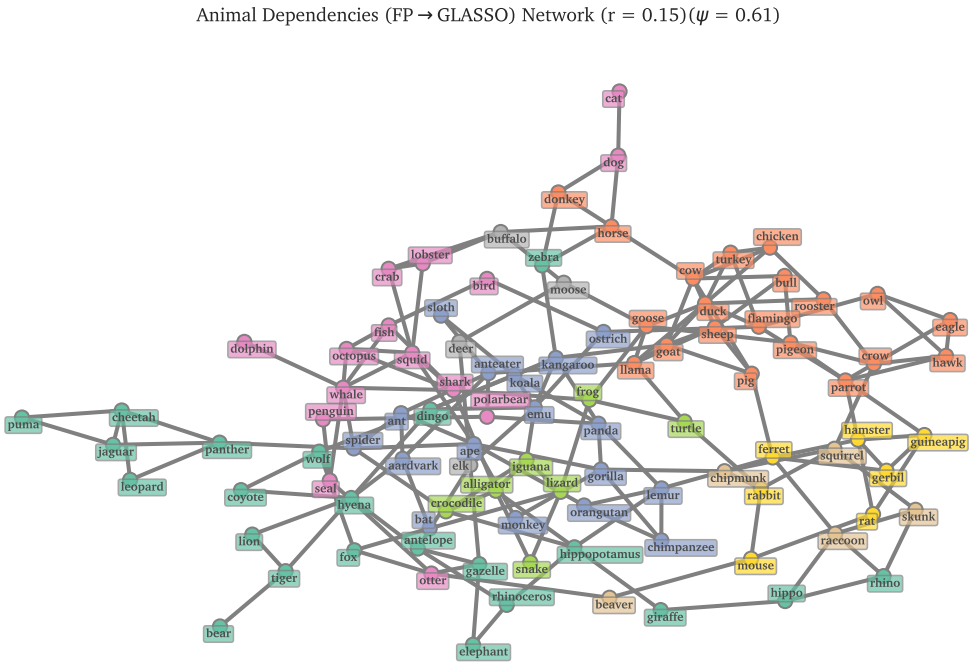


(b) dependency network drops rarer nodes from the preserved central structure at higher uncetainty cutoffs

Figure 9.5: 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 9.6: We might prefer to drop low-certainty/rare nodes from a preserved central structure.

Chapter 10: Conclusion & Future Work

10.1 Limitations of Desire Path Densities

10.2 Modifications and Extensions to Forest Pursuit

10.3 Generalizing Inner Products on Incidences

Bibliography

[1] D. Chicco and G. Jurman, "A statistical comparison between matthews correlation coefficient (MCC), prevalence threshold, and fowlkes–mallows index," *Journal of Biomedical Informatics*, vol. 144, p. 104 426, Aug. 1, 2023, ZSCC: 0000018, ISSN: 1532-0464. DOI: 10.1016/j.jbi.2023.104426. Accessed: Sep. 13, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1532046423001478.

[2] J. Crall, "The mcc approaches the geometric mean of precision and recall as true negatives approach infinity," Apr. 30, 2023. DOI: 10.48550/ARXIV.2305.00594. arXiv: 2305.00594 [cs.CV].

[3] 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.

[4] 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.

[5] S. Jukna and H. Seiwert, "Tropical kirchhoff's formula and postoptimality in matroid optimization," *Discrete Applied Mathematics*, vol. 289, pp. 12–21, Jan. 31, 2021, ZSCC: NoCitationData[s0], ISSN: 0166-218X. DOI: 10.1016/j.dam.2020.09.018. Accessed: Feb. 15, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166218X2030439X.

[6] A. Omanović, H. Kazan, P. Oblak, and T. Curk, "Sparse data embedding and prediction by tropical matrix factorization," *BMC Bioinformatics*, vol. 22, no. 1, Feb. 2021, ISSN: 1471-2105. DOI: 10.1186/s12859-021-04023-9.

[7] 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.

[8] 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. Accessed: May 8, 2023.

[9] 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.

[10] 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.

[11] 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. arXiv: 2005.03847.

[12] 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.

[13] 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. Accessed: May 8, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0195669818300155.

[14] 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.

[15] E. Ackelsberg, "What is the arcsine law?," 2018. [Online]. Available: https://math.osu.edu/sites/math.osu.edu/files/What_is_2018_Arcsine_Law.pdf.

[16] 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.

[17] 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.

[18] 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.

[19] J. Pang and G. Cheung, "Graph laplacian regularization for image denoising: Analysis in the continuous domain," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1770–1785, Apr. 2017, ZSCC: 0000231, ISSN: 1941-0042. DOI: 10.1109/TIP.2017.2651400.

[20] 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.

[21] K.-S. Jun, J. Zhu, T. T. Rogers, Z. Yang, et al., "Human memory search as initial-visit emitting random walk," *Advances in neural information processing systems*, vol. 28, no. 20, pp. 2389–2393, 2015, ISSN: 0375-9601. DOI: 10.1016/j.physleta.2019.04.060.

[22] 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. Accessed: May 7, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378873314000343.

[23] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13, Lake Tahoe, Nevada: Curran Associates Inc., 2013, pp. 2292–2300.

[24] 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.

[25] 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.

[26] 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.

[27] D. Grady, C. Thiemann, and D. Brockmann, "Robust classification of salient links in complex networks," *Nature Communications*, vol. 3, no. 1, May 2012, ISSN: 2041-1723. DOI: 10.1038/ncomms1847.

[28] P.-l. Loh and M. J. Wainwright, "Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses," in *Advances in Neural Information Processing Systems*, vol. 25, Curran Associates, Inc., 2012. DOI: 10.1214/13-aos1162. Accessed: Feb. 1, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2012/hash/6ba1085b788407963fe0e89c699a7396-Abstract.html.

[29] 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.

[30] 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.

[31] N. Meinshausen and P. Bühlmann, "Stability selection," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 72, no. 4, pp. 417–473, Aug. 2010, ISSN: 1467-9868. DOI: 10.1111/j.1467-9868.2010.00740.x.

[32] 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. Accessed: May 7, 2024. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.0904725106.

[33] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics (Oxford, England)*, vol. 9, no. 3, pp. 432–441, Jul. 2008, ZSCC: 0005870, ISSN: 1465-4644. DOI: 10.1093/biostatistics/kxm045. Accessed: Feb. 15, 2023. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3019769/.

[34] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit," *Cs Technion*, vol. 40, no. 8, pp. 1–15, 2008.

[35] G. E. Moorhouse, "Incidence geometry," *University of Wyoming*, pp. 3–5, 2007.

[36] D. P. Wipf and B. D. Rao, "An empirical bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3704–3716, Jul. 2007, ISSN: 1053-587X. DOI: 10.1109/tsp.2007.894265.

[37] T. Zhou, J. Ren, M.̌. Medo, and Y.-C. Zhang, "Bipartite network projection and personal recommendation," *Phys. Rev. E*, vol. 76, no. 4, p. 046 115, 4 Oct. 2007, ISSN: 1550-2376. DOI: 10.1103/PhysRevE.76.046115. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.76.046115.

[38] 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].

[39] "The tropical grassmannian," *advg*, vol. 4, no. 3, pp. 389–411, Jul. 2004, ISSN: 1615-715X. DOI: 10.1515/advg.2004.023.

[40] 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.

[41] 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.

[42] 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.

[43] D. B. Wilson, "Generating random spanning trees more quickly than the cover time," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*, ser. STOC '96, ACM Press, 1996, pp. 296–303. DOI: 10.1145/237814.237880.

[44] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, Apr. 1995, ISSN: 1095-7111. DOI: 10.1137/s0097539792240406.

[45] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993, ISSN: 1053-587X. DOI: 10.1109/78.258082.

[46] S. Janson and J. Vegelius, "Measures of ecological association," *Oecologia*, vol. 49, no. 3, pp. 371–376, Jul. 1981, ISSN: 1432-1939. DOI: 10.1007/bf00347601.

[47] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for steiner trees," *Acta Informatica*, vol. 15, no. 2, pp. 141–145, 1981, ISSN: 1432-0525. DOI: 10.1007/bf00288961.

[48] 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.

[49] D. Hunter, "An upper bound for the probability of a union," *Journal of Applied Probability*, vol. 13, no. 3, pp. 597–603, Sep. 1976, ISSN: 1475-6072. DOI: 10.2307/3212481.

[50] W. Feller, *An Introduction to Probability Theory and its Applications, Volume 1*. J. Wiley & Sons: New York, 1968.