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. [14] However, an increasing number of authors are noting a lack of agreement in how to approach the metrology of this problem. This includes rampant disconnects between the theoretical and methodological network analysis sub-communities[1], treatment of error as purely aleatory, rather than epistemic [15], or simply ignoring measurement error in network reconstruction entirely[7].

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*[22], 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.[19]

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.

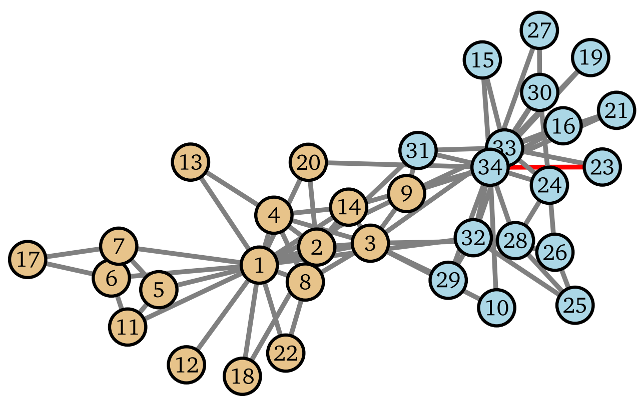


Figure 1.1: Zachary's Karate Club, with ambiguously extant edge 78 highlighted.

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.

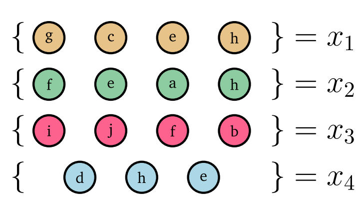


Figure 1.2

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. [20] 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”.

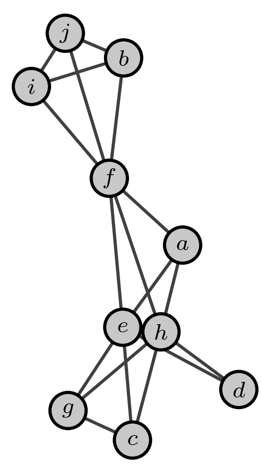


Figure 1.3

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 colleagues, perhaps the we prefer our edges represent "mutual willingness to collaborate". Edge "measurement", then, could involve records of events that show willingness to seek or participate in collaboration event, such as:

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

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

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

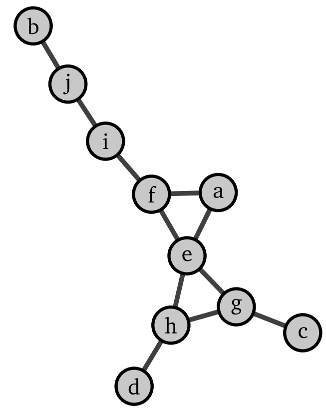


Figure 1.4: graph of mutual collaboration relationships i.e. the "ground truth" social network

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

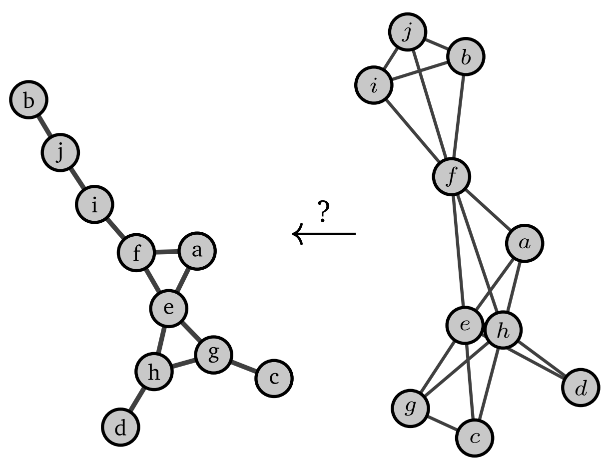


Figure 1.5: Recovering underlying dependency networks from node-cooccurrences.

Even more practically, networks created directly from bipartite-style data are notorious for quickly becoming far too dense for useful analysis, earning them the (not-so-)loving moniker "hairballs". Network "backboning," as it has come to be called 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.[11, 17] Meanwhile, underlying networks of dependencies that *cause* node activation patterns can provide this: they are almost always more sparse than their hairballs. Accessing the dependency *backbone* in a principled way is difficult, but doing so in a rapid, scalable manner is critical for practitioners to be able to make use of it to trim their hairballs.

1.3 Scope of this work

The purpose of this thesis is to provide a solid foundation for basic edge metrology when our data consists of binary node activations, by framing network analysis as a problem of *inference*, as suggested by Peel et al. [1]. We give special focus to binary activations that occur due to spreading processes, such as random walks or cascades on an underlying carrier graph. Recovering the carrier, or, "dependency" network from node activations is of great interest to the network backboning and causal modeling communities, but often involves either unspoken sources of epistemic and aleatory error, or high computation costs (or both). To begin addressing these issues, we present a guide to current practices, pitfalls, and how common statistical tools apply to the network recovery problem: a *Practitioner's Guide to Network Recovery*. We cover what "measurement" means in our context, and specifically the ways we encode observations, operations, and uncertainties numerically. Clarifying what different versions of what "relation" means (whether proximity or incidence) is critical, since network structure is intended to encode such relations as mathematical objects, despite common ambiguities and confusion around what practitioners intend on communicating through them. Then we use this structure to present a cohesive framework for selecting a useful network recovery technique, based on the available data and where in the data processing pipeline is acceptable to admit either extra modeling assumptions or information loss.

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

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

Part I

A Practitioner’s Guide to Network Recovery

Chapter 2: Metrology as matrices

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

[1]Not all observations are scalar, but they can become so. If individual 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 [16] and Kepner et al. [10].

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 smoothing".[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 Bayesian prior on the binomial probability (a.k.a. a *Beta-Binomial* distribution) $p \sim \text{Beta}(\alpha, \beta)$, such that the posterior distribution after our success/failure counts is $\text{Beta}(s+\alpha, k-s+\beta)$, which has the mean

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

which 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" of an average run of trials ($\alpha = 1$, nearly all evidence has been collected), or about halfway through an average-length run ($\alpha = 0.5$, 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) pair-wise relationships, the independence assumption required to estimate node occurrences as Beta-Binomial is clearly violated[7].

However, it's natural to make use of joint ($P(A \cup B)$, how often does A happen with B, out of all data?) and conditional ($P(A|B)$ how often A given B; or $P(B|A)$, how often B given A) probabilities between nodes, while trying to estimate dependencies. Once again, we can estimate the base probabilities for each node from marginal sums, 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 $G(i,j) : \mathbb{Z}^{n\times n}$ has the co-occurrence count for node "i" with "j".

The co-occurrence probability for each pair can 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, directly, and we also don't have a reliable way to estimate the opportunities each edge had to activate

[7]In fact, a recent method from [12] 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.

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

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

2.4 Proximity vs. Incidence

As alluded to in the previous section, co-occurrence seems to have a deep connection to a Gram matrix, which is a

2.4.1 Kernels & distances

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

$$G(\mathbf{x}_i, \mathbf{x}_j, R) = G_{ij} = \bigoplus_{k=1}^{n} x_{ik} \otimes x_{kj} \tag{2.2}$$

such that real-valued entries and a traditional "plus-times" inner product recovers the Gram matrix

$$G_{ij} = X^T X \sum_{k=1}^{n} x_{ik} x_{kj}$$

How "close" or "far away" things are.... Avrachenkov et al.

Important: these measurements often assume distance is defined in terms of the measurements/objects/data, but for *inverse problems*, structure learning, etc., they are more often applied in terms of the features/operators.

Example with doc-term matrices

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

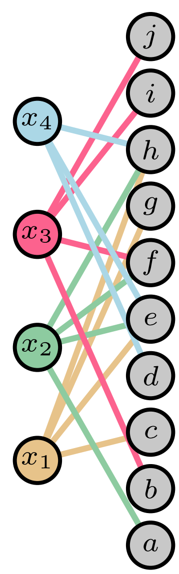
Similarly, if our goal is to determine a network of "whether two authors ever coauthored", we could perform a bipartite projection using the boolean inner product in the observation space i.e. the "authors" projection. It is this second projection, for determining a structure between features embedded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

2.4.2 Incidence structures & dependency

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

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

(a)



(b) Bipartite representation of node "activation" data

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

2.4.3 Implications for networks

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

- Discuss Complex Systems and their representation.

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

Chapter 3: Incidence through vector representation

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

Traditionally[3, 10], we might say a graph on nodes (or, "vertices") $v \in V = \{1, \cdots, n\}$ and "edges" E is a tuple:

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

The *adjacency matrix* A of G, degree matrix D, and graph/discrete Laplacian L

are then defined as:[1]

$$
\begin{aligned}
A(u,v) &= \mathbf{1}_E((u,v)) & A &: V\times V \to \mathbb{B}\\
D(u,v) &= \operatorname{diag}(\textstyle\sum_V A(u,v)) & D &: V\times V \to \mathbb{N}\\
L(u,v) &= D(u,v) - A(u,v) & L &: V\times V \to \mathbb{Z}
\end{aligned}
$$

However, if edges and their recovery is so important to us, defining them explicitly as pairs of nodes can be problematic when we wish to estimate their existence (or not) when pairs of nodes co-occur. 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 $\mathscr{I}$:

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

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

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

- 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

[2]never two flags with the same pair, i.e. $\mathscr{I}$ is a set, not a multiset.

[3]in the set of flags containing nodes u or v, only one e may be incident to both of them.

explicitly use incidence relation S to tell us whether each possible edge "exists" or not, simplifying our graph definition further[4]:

$$
\begin{aligned}
G = (V, E, B) \quad s.t. \quad & B : E \times V \to S \\
& v \in V = \{1, \cdots, n\} \\
& e \in E = \left\{1, \cdots, \binom{n}{2}\right\}
\end{aligned}
\tag{3.3}
$$

3.1.1 Embedding in vector space

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].

However, 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. This makes every incidence be embedded 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 vectors are:

---

[4] if we allow multi-edges, then

[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!

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

The unoriented incidence matrix is easily defined as having rows that are sums over the incidence vectors for each edge: $\mathbf{b}_e^+ = \sum_{v\in V_e} \delta_e(v)$

To build the oriented incidence matrix, one could define undirected graphs as equivalent 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 definition based on sums would hold. The resulting set of incidences would have twice the number of edges than our combinatoric limit for simple graphs, however. Plus, it would necessitate averaging of weights over different edge ID's to be prior to use of inner products, and many other implementation difficulties.

Instead we would like to allow for systematic differences between the incidence vectors, without ambiguity. But, now that we have removed the information on "which nodes an edge connects" from our definition (since every edge is a scalar ID), how do we construct V_e without a costly search over all incidences?

Because of our unique identification of edges up to the combinatoric limit, we can still actually provide a unique ordering of the nodes it each edge connects. Using an identity from Angeletti et al. [5], 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:

$$
\begin{aligned}
u_n(e) &= n-2-\left\lfloor \frac{\sqrt{-8e+4n(n-1)-7}-1}{2} \right\rfloor \\
v_n(e) &= e+u_n(e)+1-\frac{1}{2}\Big(n(n-1)+(n-u_n(e))^2-n+u_n(e)\Big) \\
e_n(u,v) &= \frac{1}{2}\Big(n(n-1)-((n-u)^2-n+u)\Big)+v-u-1
\end{aligned} \tag{3.4}
$$

Using this, we can unambiguously define a *partition* between incidences on e, without needing other metadata or flags to distinguish u from v, since those are defined through Equation 3.4 unambiguously w.r.t. e.

$$\mathbf{b}_e = \delta_e(u) - \delta_e(v)$$

This could be called an "edge-centric" view of simple graphs, since the nodes involved in a graph are now actually derived w.r.t. the edges and our mapping B, uniquely.

3.1.2 Inner products on B

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

Rescaling to achieve normaalization.

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

…

3.1.3 Metrological Considerations: Interaction Vectors

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

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

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

3.2 Graphs and Node Occurences

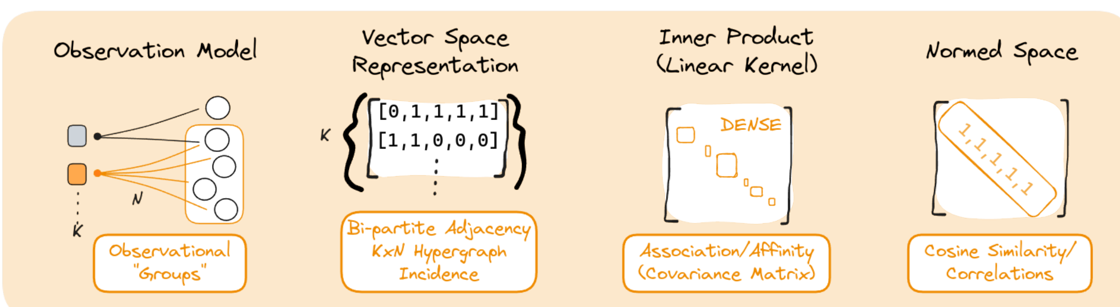


Figure 3.1: Hyperedge Relation Observational Model

3.2.1 Hyperedges as Vectors of Nodes

3.2.2 Inner product on Hyperedges

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

Leads to correlation/covariance, etc.

3.2.3 Combining Occurence & Dependence

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

Chapter 4: Roads to Network Recovery

4.1 Choosing a structure recovery method

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

4.2 Organizing Recovery Methods

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

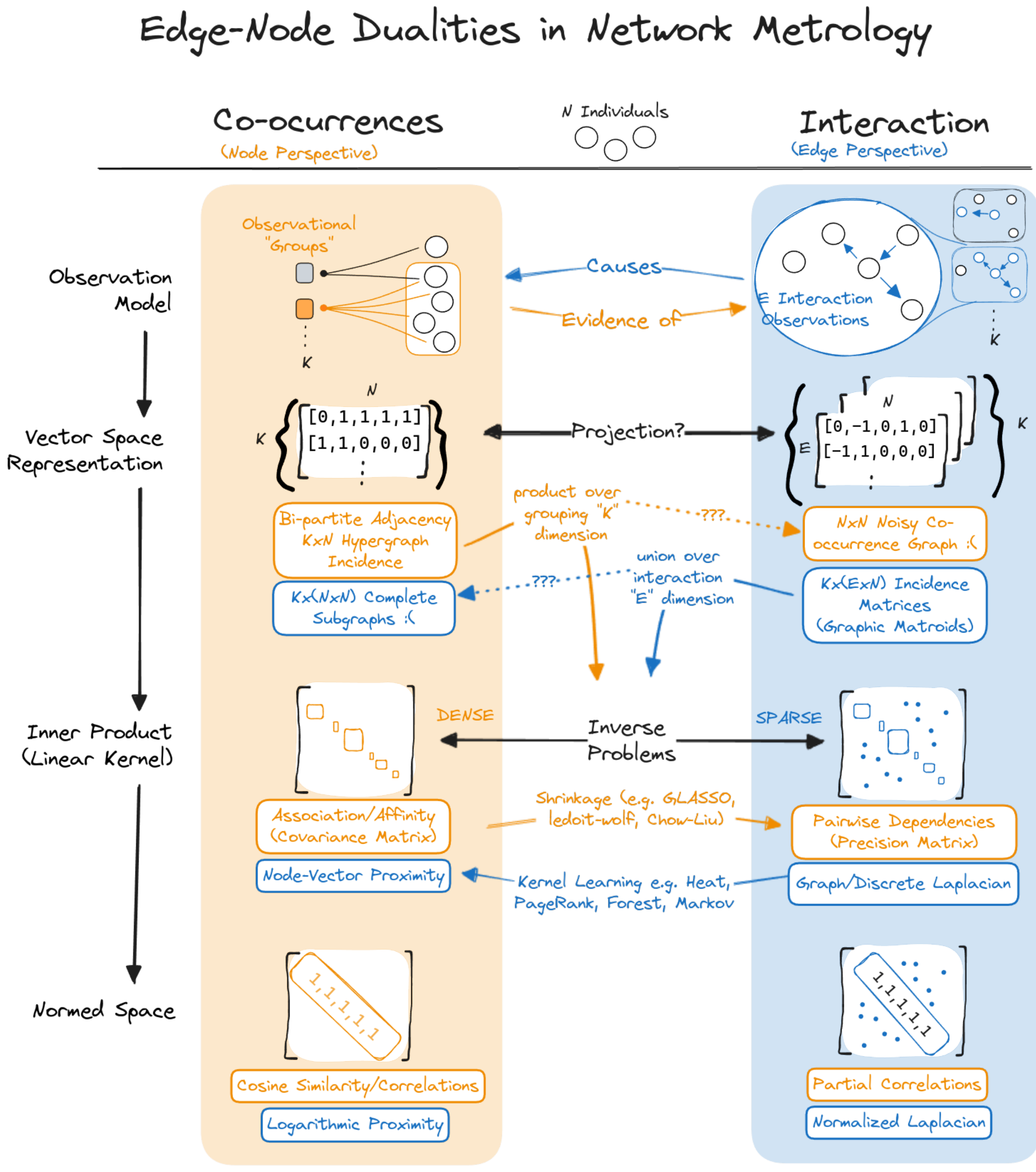


Figure 4.1: Relating Graphs and Hypergraph/bipartite structures as adjoint operators

4.2.1 Observing Nodes vs Edges

4.2.2 Embeddings, Inner Products, & Preprocessing

4.3 Tracing Information Loss Paths

4.3.1 Table of Existing Approaches

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

4.3.2 A Path Forward

Sorting algorithms… *none address all three!*

i.e. MOTIVATES FOREST PURSUIT

Part II

Nonparametric Network Recovery With Random Spanning Forests

Chapter 5: Desire Paths and Spanning Forests

Addressing gaps discussed in the previous section to reach a generative model for network recovery requires careful attention to the generation mechanism for node activations. While there are many ways we might imagine bipartite data to be 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. [8]. Then we use the implicit causal dependency tree structure of each observation, together with the Matrix Forest Theorem [13, 18] 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 [21], 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. [9] 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. [1]. They include a warning not to naively use correlational measures with a threshold, since even simple 3-node systems will easily yield false positives edges. Still, it would be helpful for practitioners to have a more explicit mental model of *why* co-occurence-based models yield systematic bias,

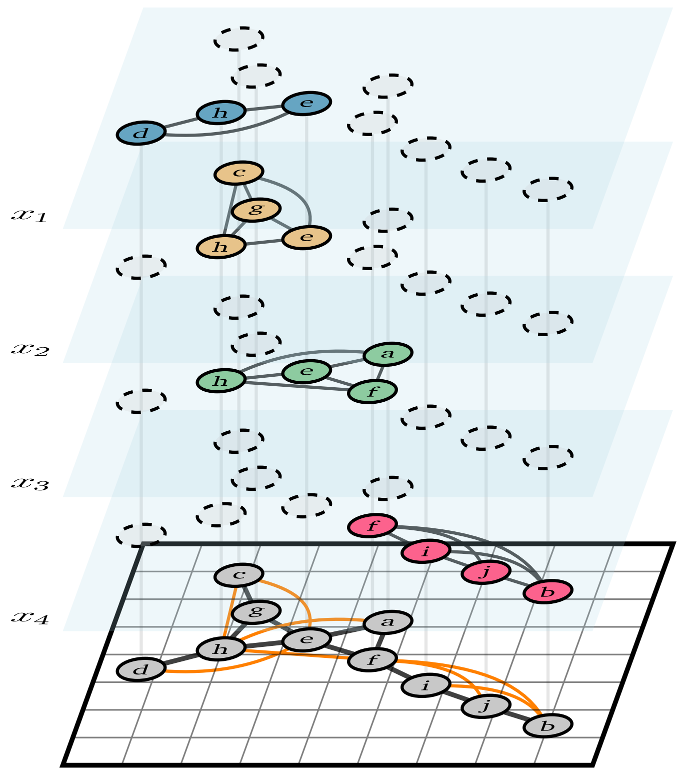
5.1.2 Inner Products as Sums of Cliques

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.

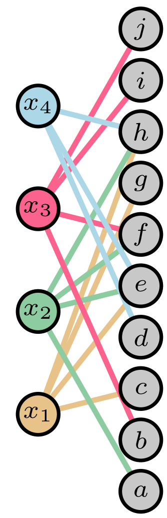
However, another (equivalent) way to view matrix multiplication is as a sum of outer products

$$G(i,j) = X^T X = \sum_{i=1}^{k} X(i,k)X(j,k) = \sum_{i=1}^{k} \mathbf{x}_k \mathbf{x}_k^T$$

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—i.e., they can be seen as adjacency matrices of the clique on nodes activated in the kth 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 5.1.



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



(b) Bipartite representation of node "activation" data

Figure 5.1: Inner-product projections imply observations of complete graphs, summed.

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 a room, look at a number of others (who all look exclusively at each other, as well), and 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 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. What we need is a way to retain the ease of use of inner-product network creation, with a more domain-appropriate graph at the observation level. Of course there are many 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 dependencies are strictly planar [planar maximally filtered graps][3]. Alternatively, the interactions might be scale free, small-world, 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. In other words, we limit our possible interaction graph to a set of all possible graphs in our class $\mathscr{C}$, and model the interactions allowed to be inferred from the activated nodes as $c \in \mathscr{C}$. With an associated probability measure $\mu_{\mathscr{C}}(c)$ defined on subgraphs of the complete graph on our node set V[4], we are able to say our interactions are

[2]e.g. for dependencies based on perception, such as human decision making tendencies, or causes based on color names.

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

[4]Assuming we can even find one, something we must return to shortly

sampled from a distribution over graphs in the class.

For simplicity, since unreported/hidden-but-activated nodes is outide the scope of our work, we narrow the distribution to be only on the induced subgraph $C(S_k) \in G[S_k])$, where S_k is the set of activated nodes in $\mathbf{x}_k$. More specifically, we say that an observation of $S \subset V = \{s_1, \cdots, s_t\}, \quad t = |S|$ activated nodes implies a distribution over edge vectors we could hav observed, determined by the chosen class,

$$\mathbf{x}_k^E \in$$

Once an analyst has provided epistemic justification for a *class of graphs* to model We propose that the computationally-efficient inner-product networks can still be used, but could be made far more effective by counting edge observation counts with something more appropriate than cliques.

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

5.2.1 Random Walk Activations

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

When network structure is known, the dynamics of random-walks are used to capture the network structure via sampling [LITTLEBALLOFFUR, etc], estimate node importance's[PAGERANK], or predict phase-changes in node states (e.g. infected vs. uninfected)[SIR I think] In our case, Since we have been encoding the activations as binary activation vectors, the "jump" information is lost—activations are "emitted" for observation only upon the random walker's initial visit. [CITE INVITE] In many cases, however, the existence of relationships is not known already, and analysts might *assume* their data was generated by random-walk-like processes, and want to use that knowledge to estimate the underlying structure of the relationships between nodes.

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

5.2.2 Dependencies as Trees

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

[GRAPHIC 1 - my data]

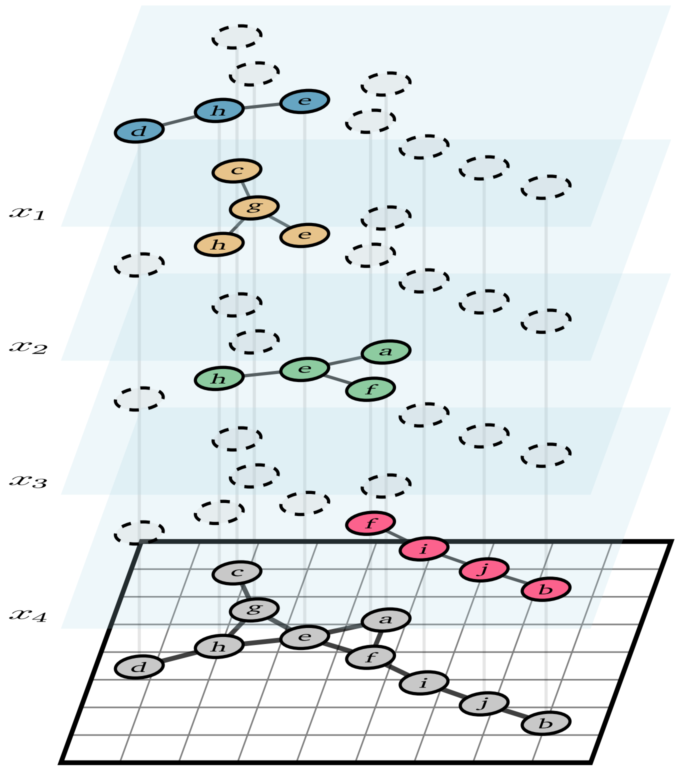


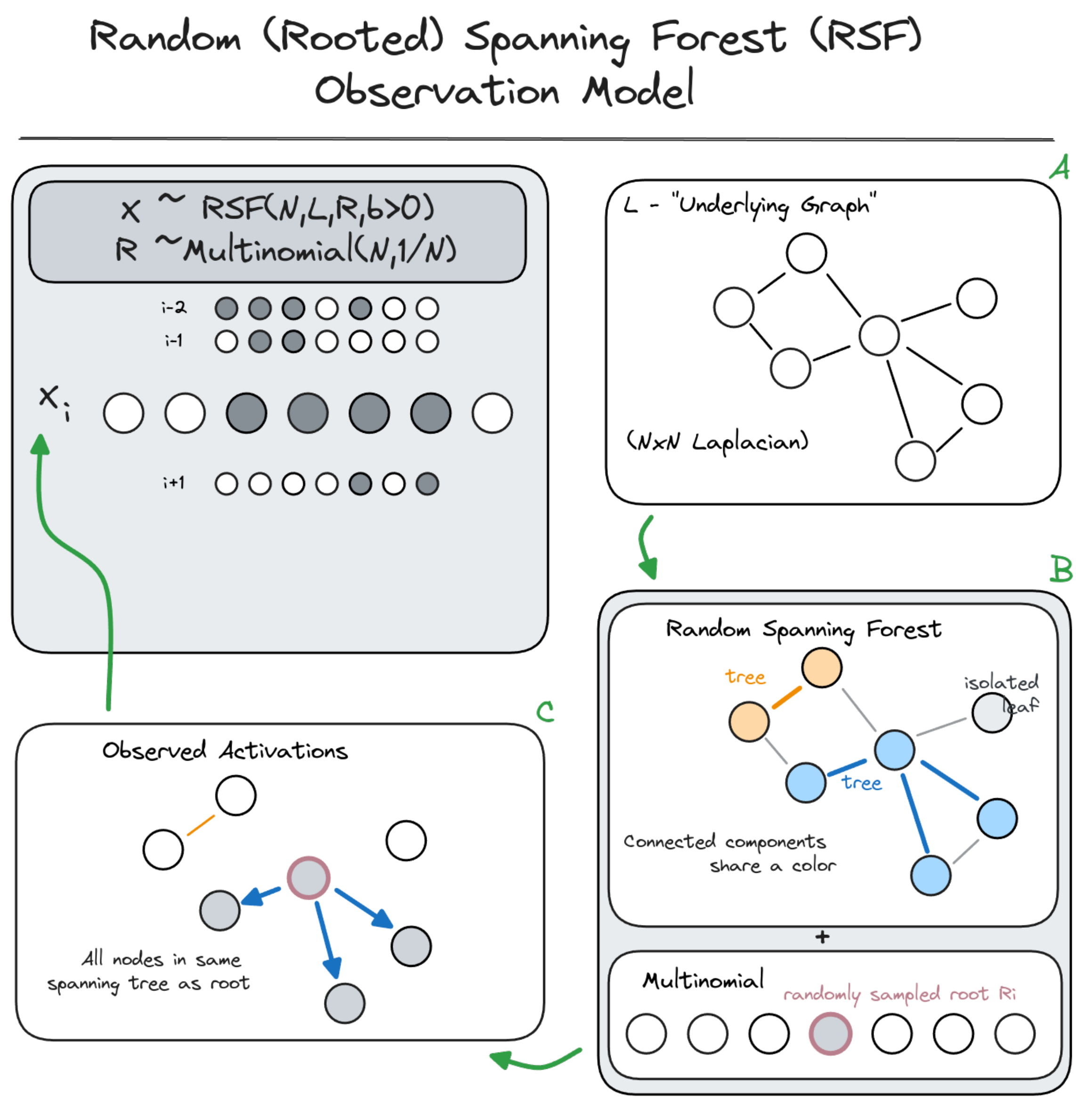
Figure 5.2: Edge Measurements with true (tree) dependencies known

[GRAPHIC 2 - infection vector from meta node]

5.2.3 Matrix Tree and Forest Theorems

- one from kirchoff
- one from Chebotarv

5.3 Generative Model Specification



- hierarchical model - marginalize over the root node.

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

filling the gap we saw in the literature

6.1 Sparse Dictionary Learning

6.1.1 Problem Specification

6.1.2 Matching Pursuit

6.1.3 Space of Spanning Forests

6.2 Forest Pursuit: Approximate Recovery in Near-linear Time

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

6.2.1 Uncertainty Estimation

6.2.2 Approximate Complexity

6.3 Simulation Study

6.3.1 Method

6.3.2 Results - Scoring

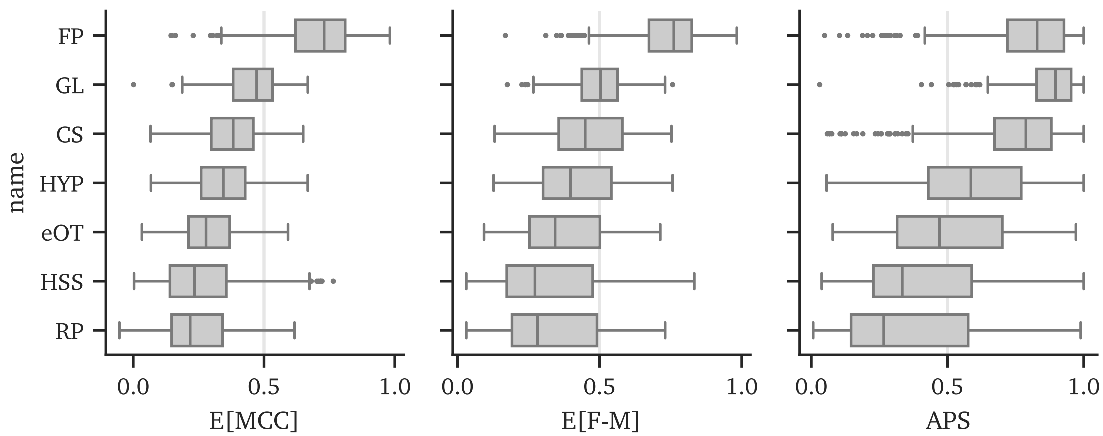


Figure 6.1: Comparison of MENDR recovery scores

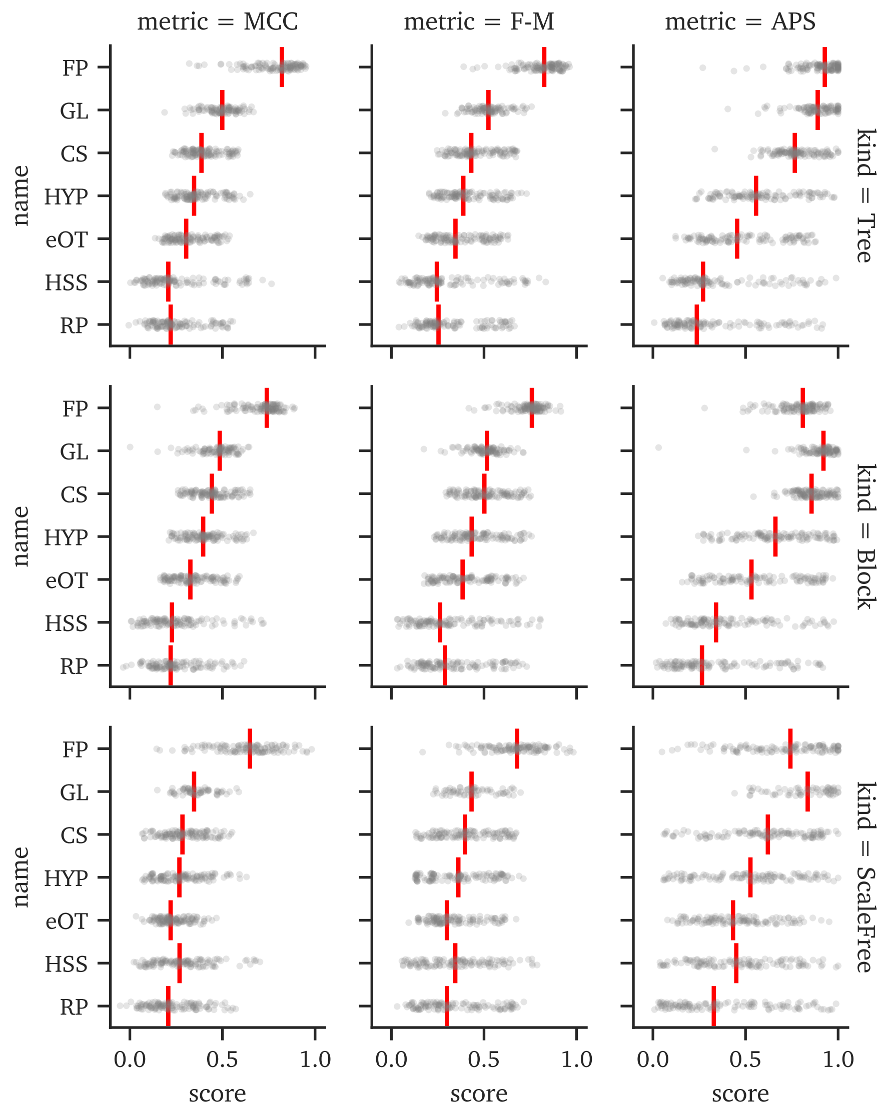


Figure 6.2: Comparison of MENDR Recovery Scores by Graph Type

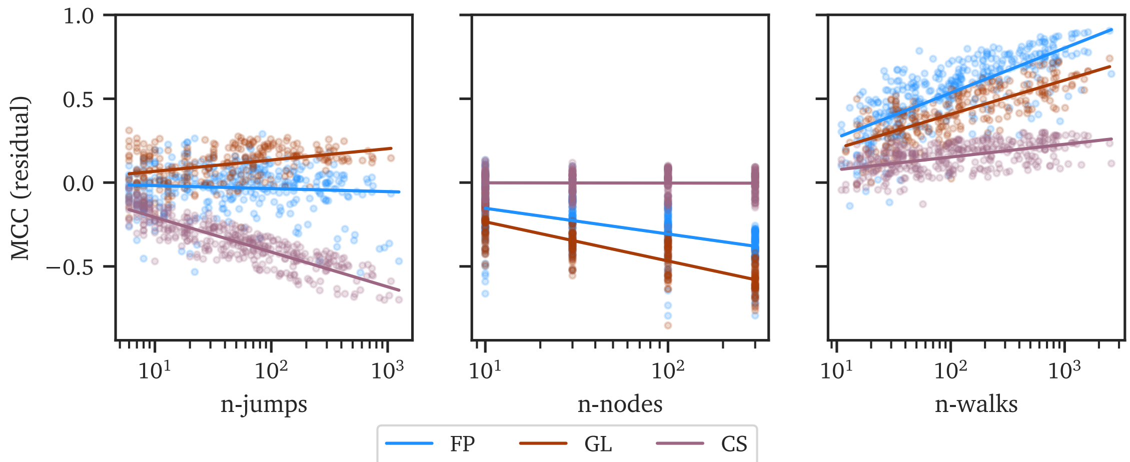


Figure 6.3: Partial Residuals (regression on E[MCC])

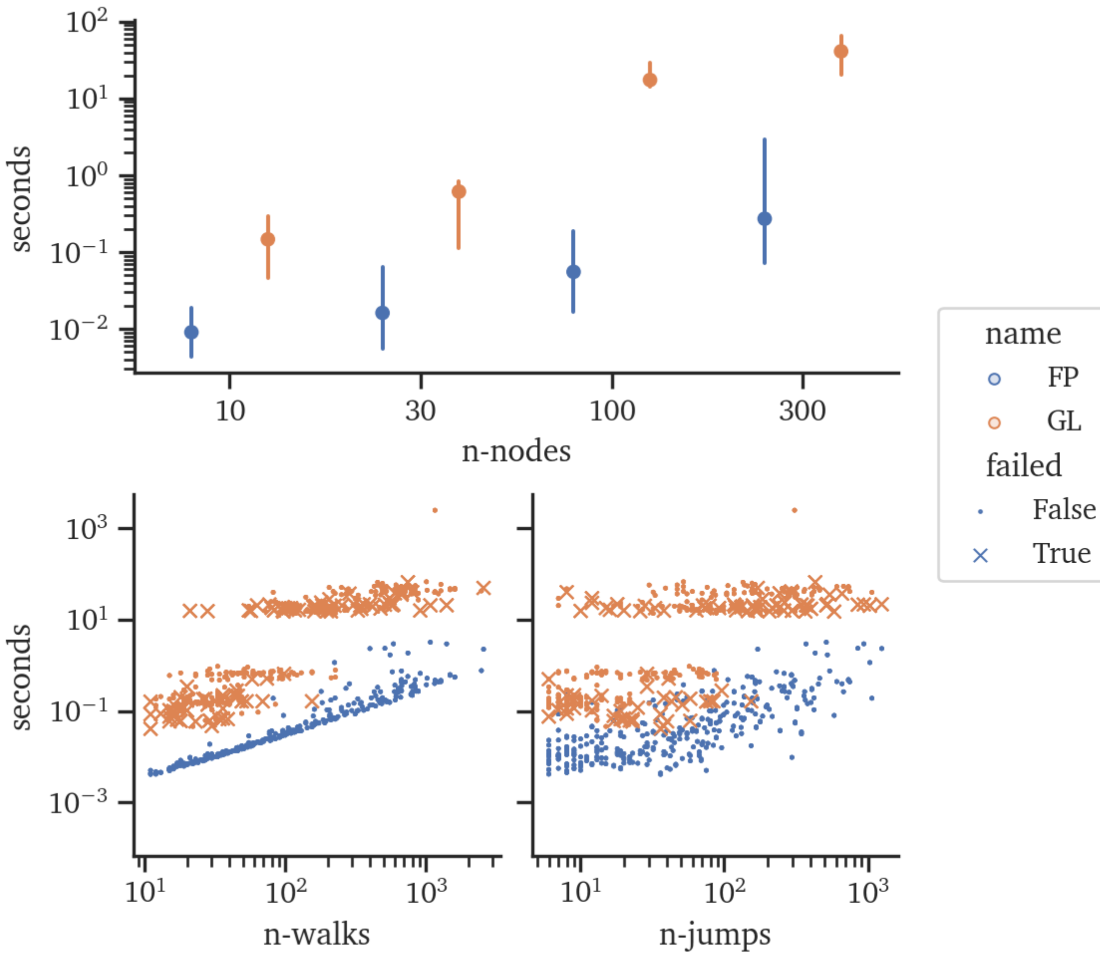


Figure 6.4: Runtime Scaling (Forest-Pursuit vs GLASSO)

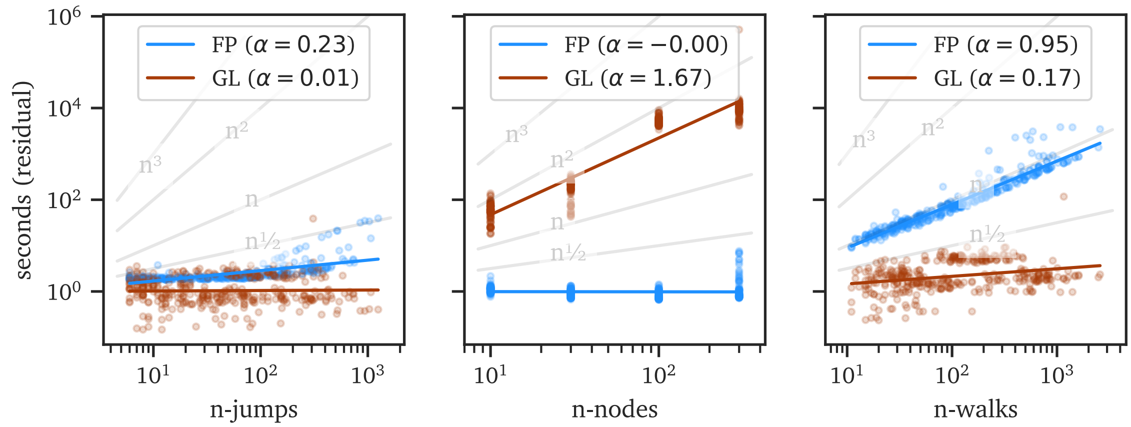
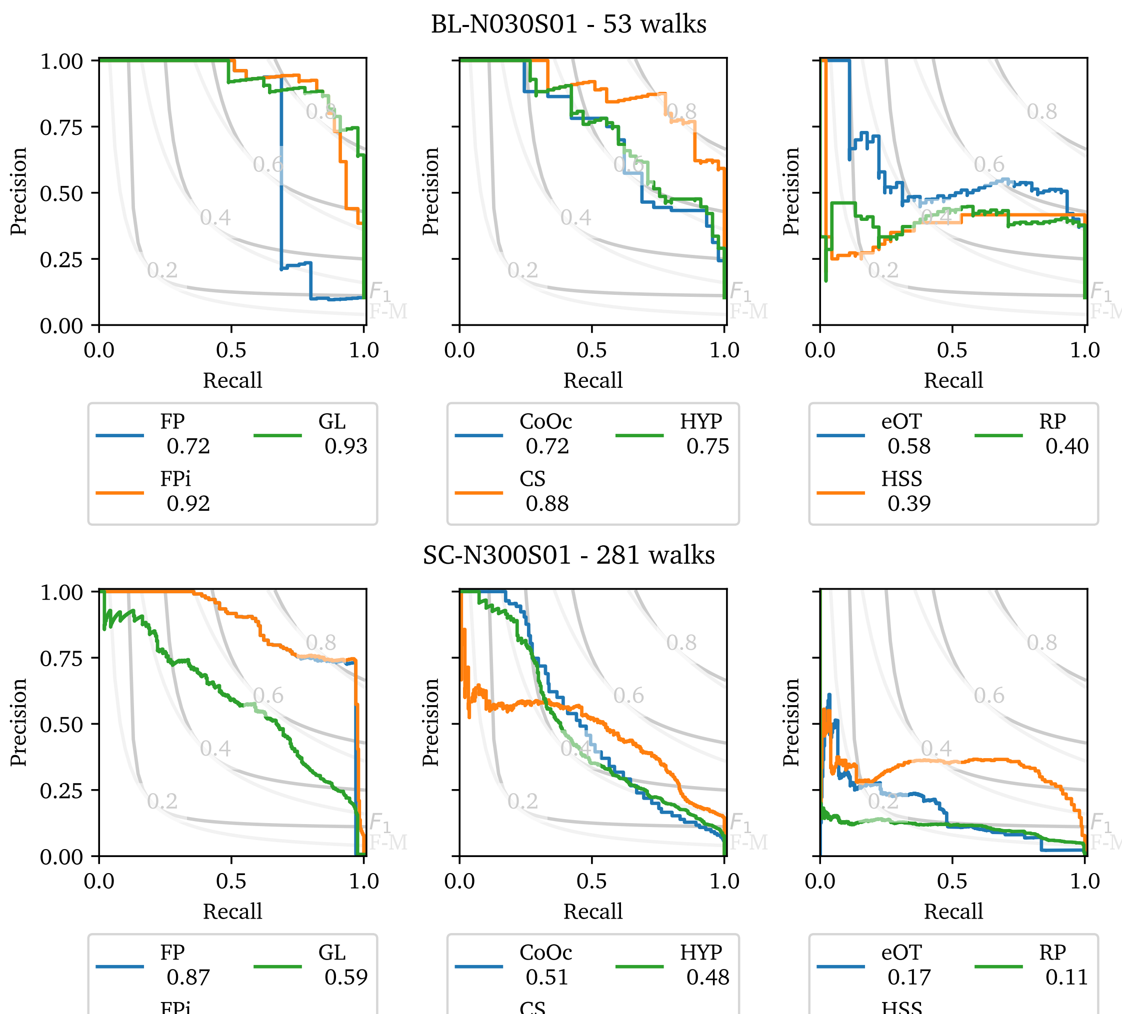


Figure 6.5: Partial Residuals (regression on computation time)

6.3.3 Results - Performance

6.4 Discussion

6.4.1 Interaction Probability



Chapter 7: LFA: Latent Forest Allocation

7.1 Radom Spanning Trees

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

7.2 Bayesian Estimation by Gibbs Sampling

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

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

7.3 Simulation Study

7.3.1 Score Improvement

7.3.2 Odds of Individual Edge Improvement

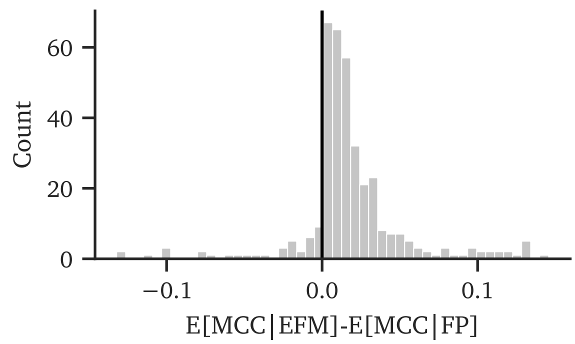


Figure 7.1: Change in Expected MCC (EFM vs FP)

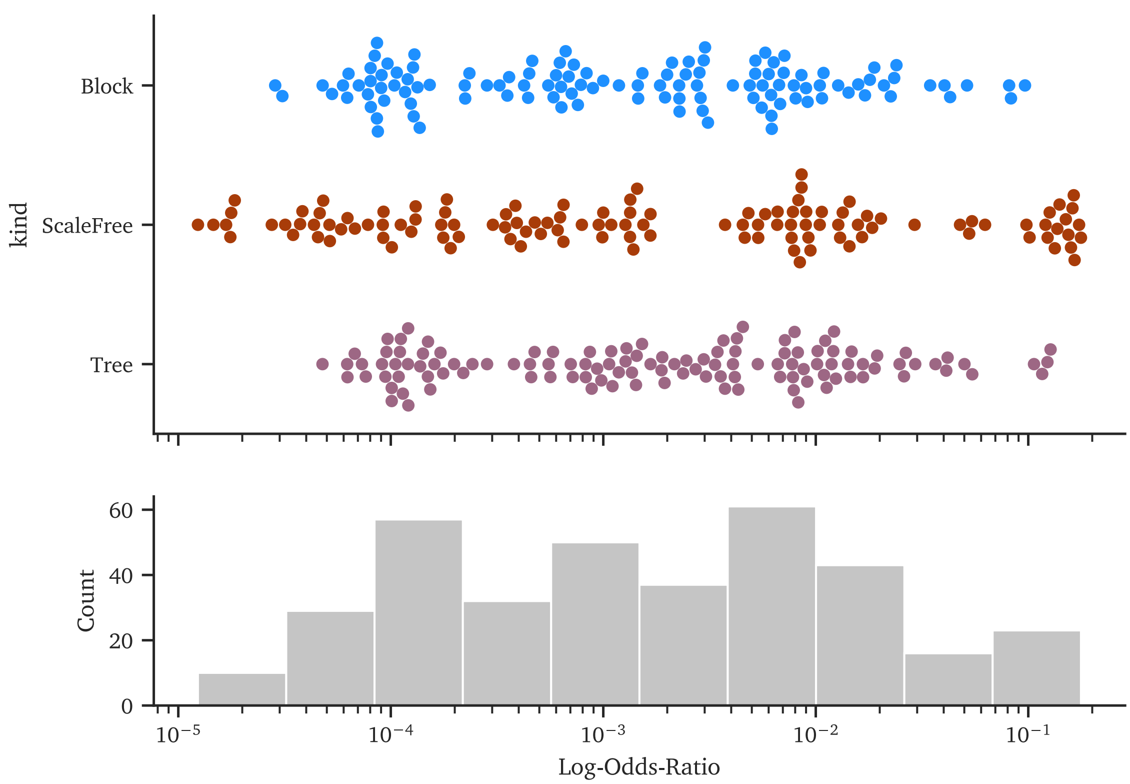


Figure 7.2: Logistic Regression Coef. (EFM - FP) vs. (Ground Truth)

Part III

Applications & Extentions

Chapter 8: Qualitative Application of Relationship Recovery

8.1 Network Science Collaboration Network

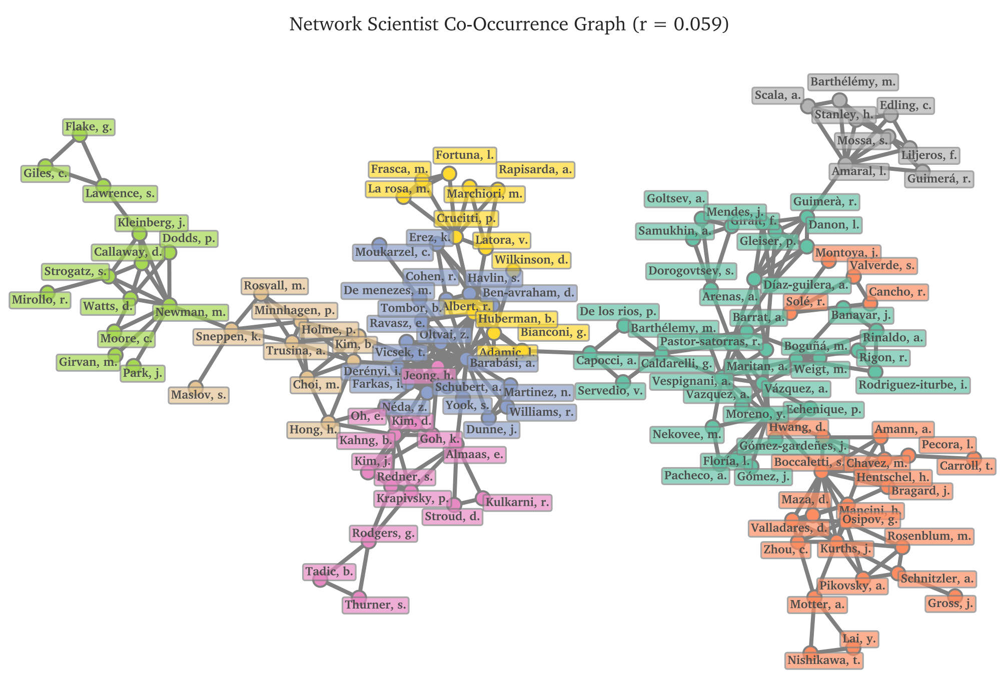


Figure 8.1: 134 Network scientists from [NEWMAN;BOCCALETTI;SNEPPEN], connected by co-authorship

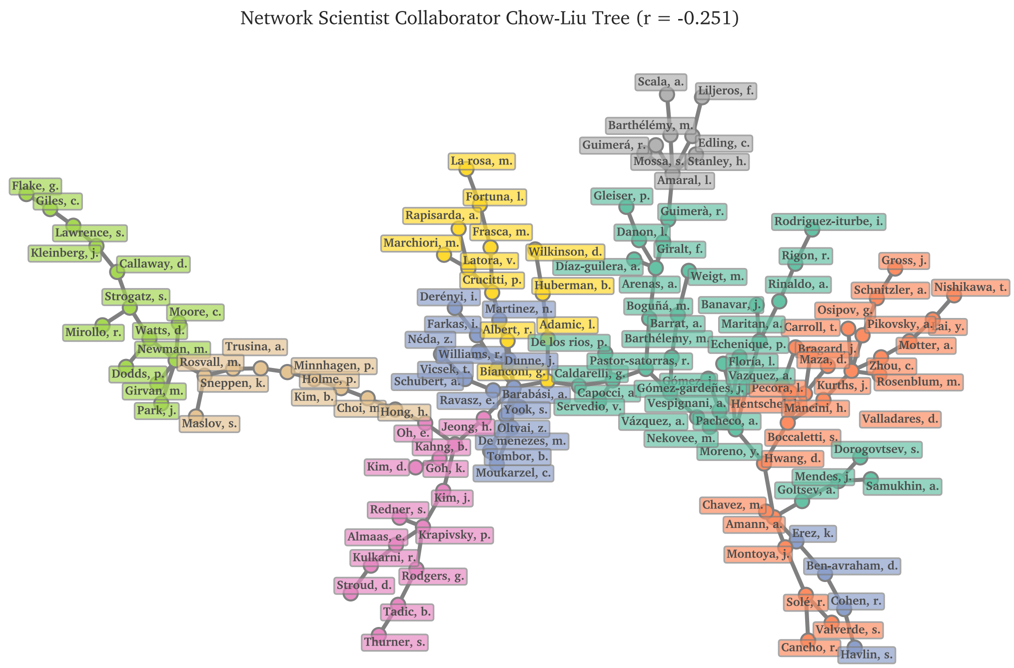


Figure 8.2: Max. likelihood tree dependency structure to explain co-authorships

8.2 Les Miserables Character Network

8.2.1 Backboning

8.2.2 Character Importance Estimation

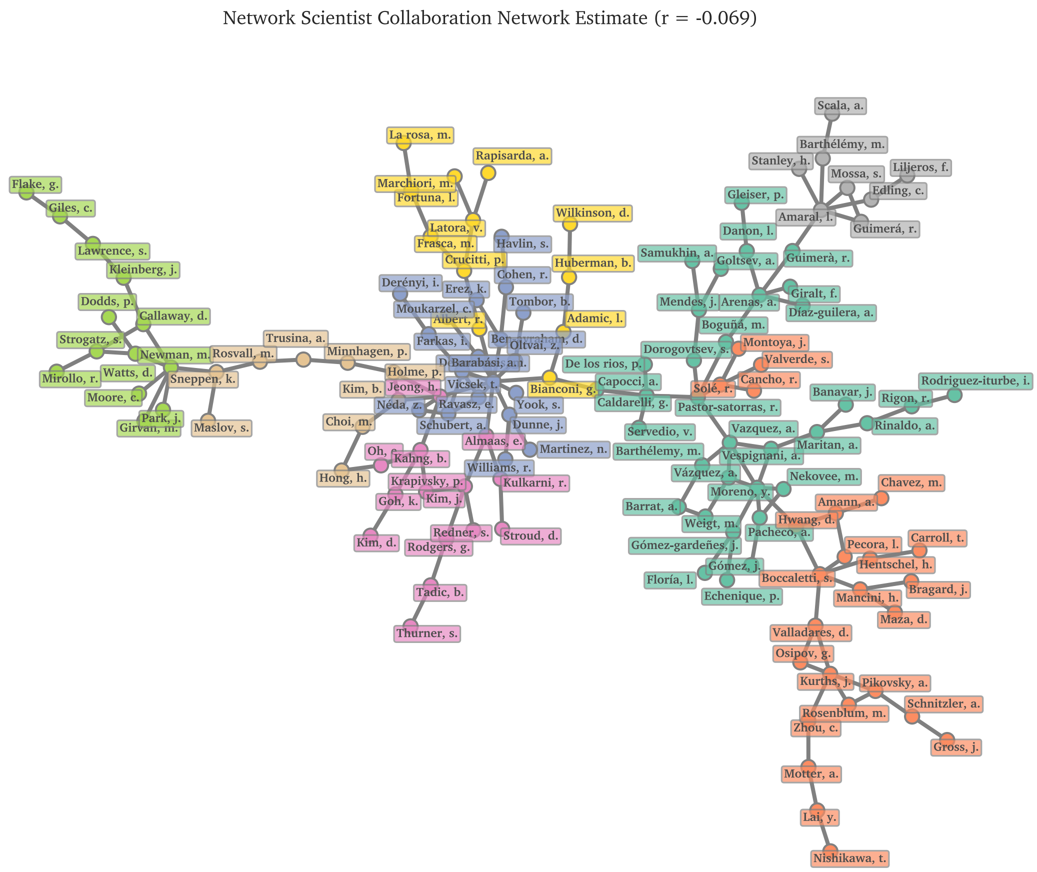


Figure 8.3: Forest Pursuit estimate of NetSci collaborator dependency relationships

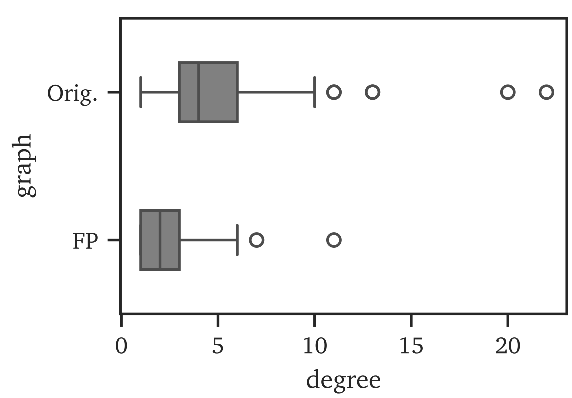


Figure 8.4

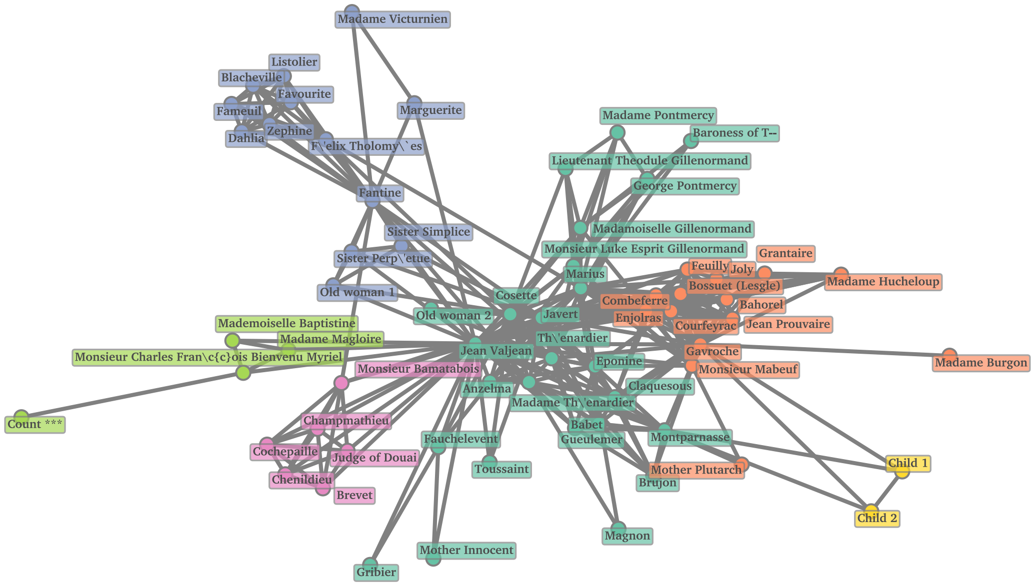


Figure 8.5

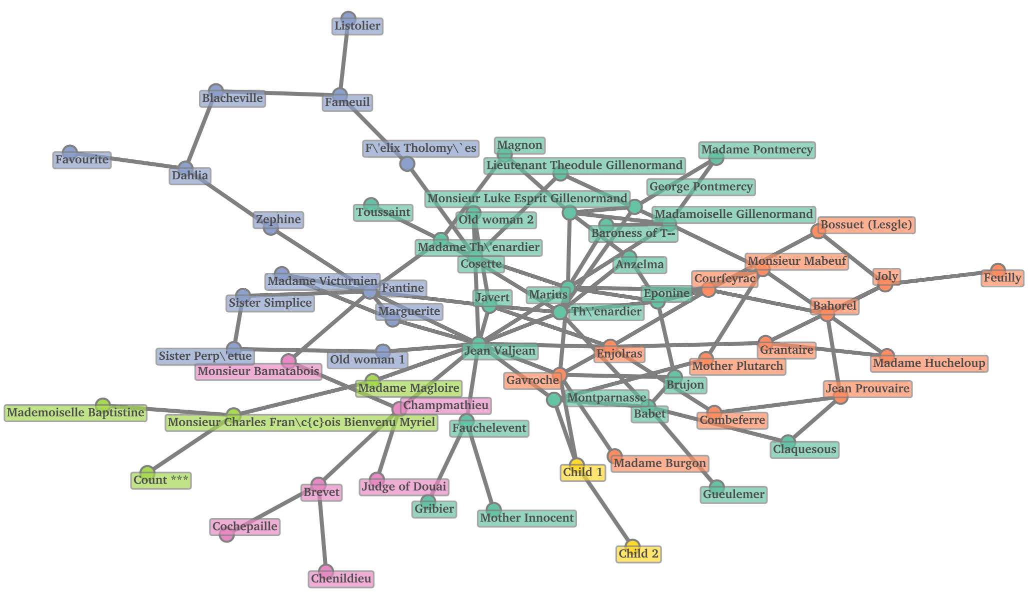


Figure 8.6

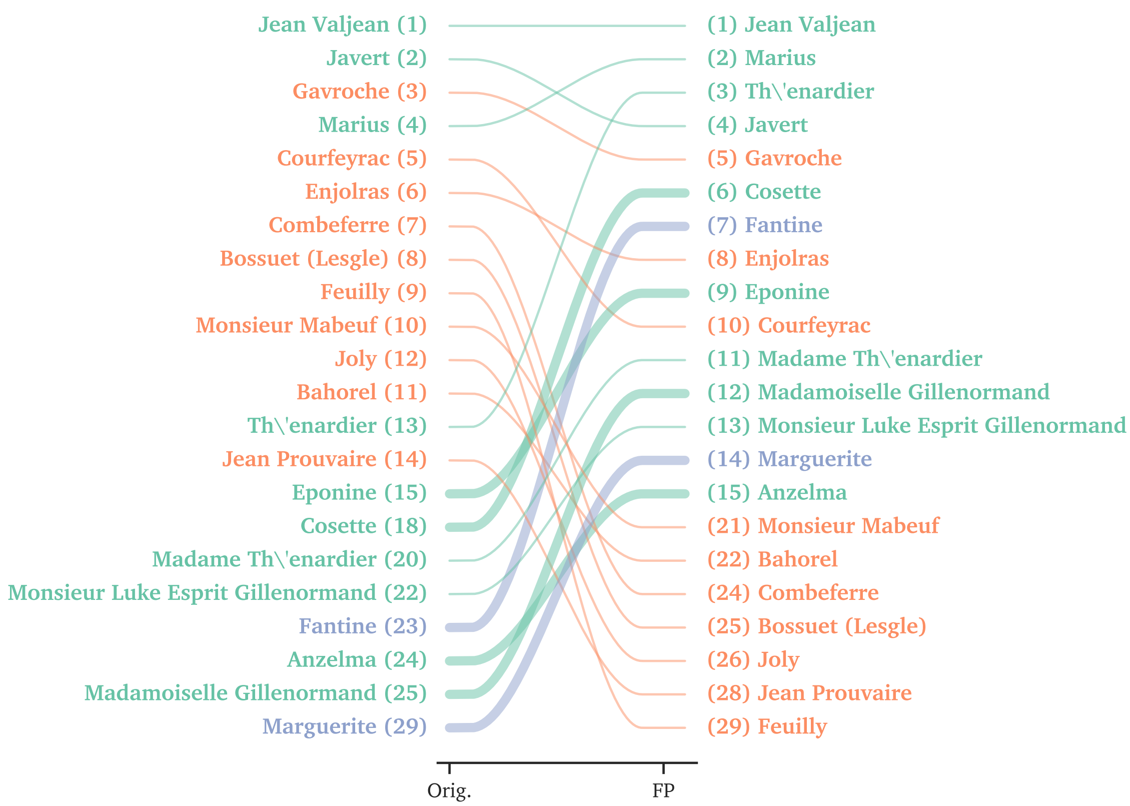

Jean Valjean (1)
Javert (2)
Gavroche (3)
Marius (4)
Courfeyrac (5)
Enjolras (6)
Combeferre (7)
Bossuet (Lesgle) (8)
Feuilly (9)
Monsieur Mabeuf (10)
Joly (12)
Bahorel (11)
Th\'enardier (13)
Jean Prouvaire (14)
Eponine (15)
Cosette (18)
Madame Th\'enardier (20)
Monsieur Luke Esprit Gillenormand (22)
Fantine (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 [4, 6]

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.

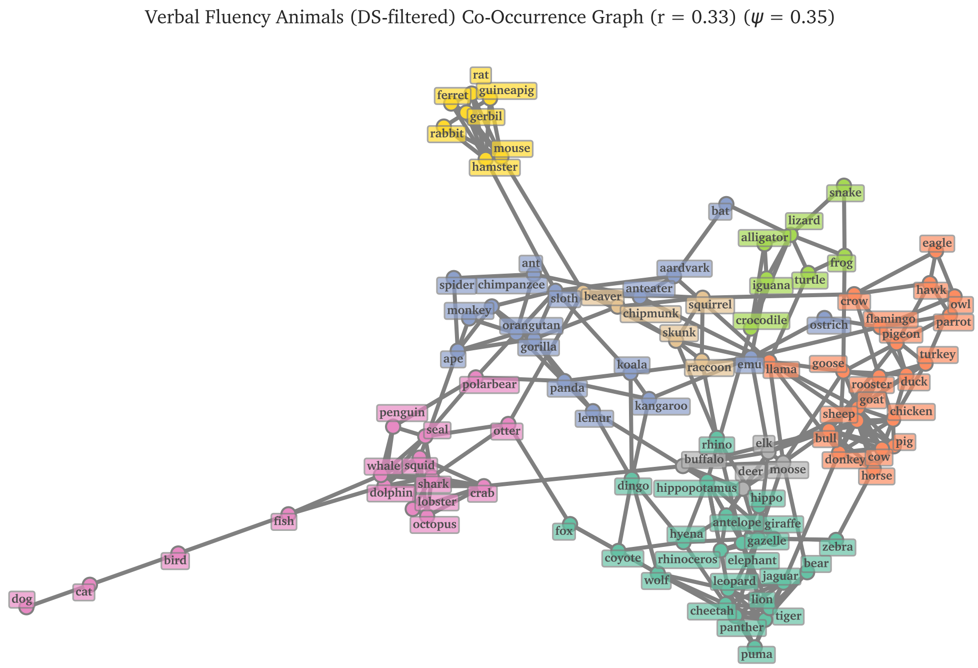


Figure 9.1

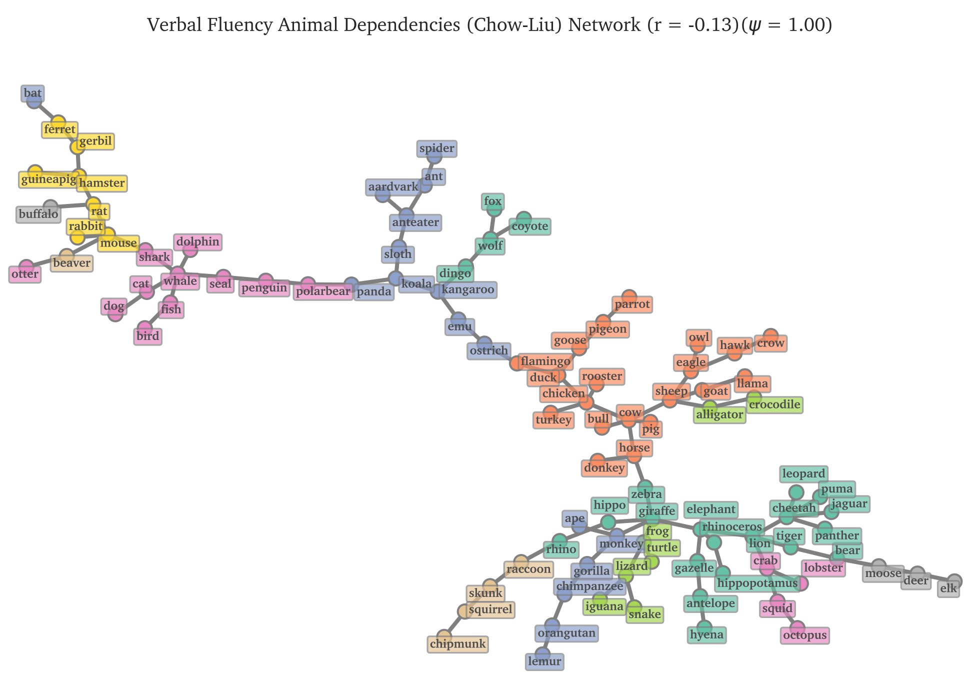


Figure 9.2

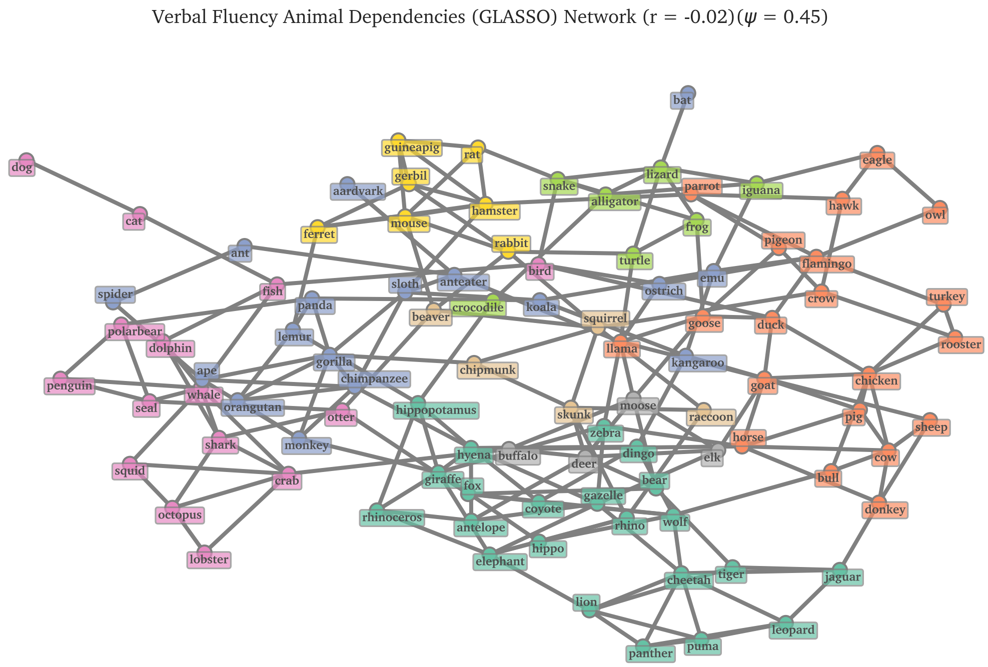


Figure 9.3

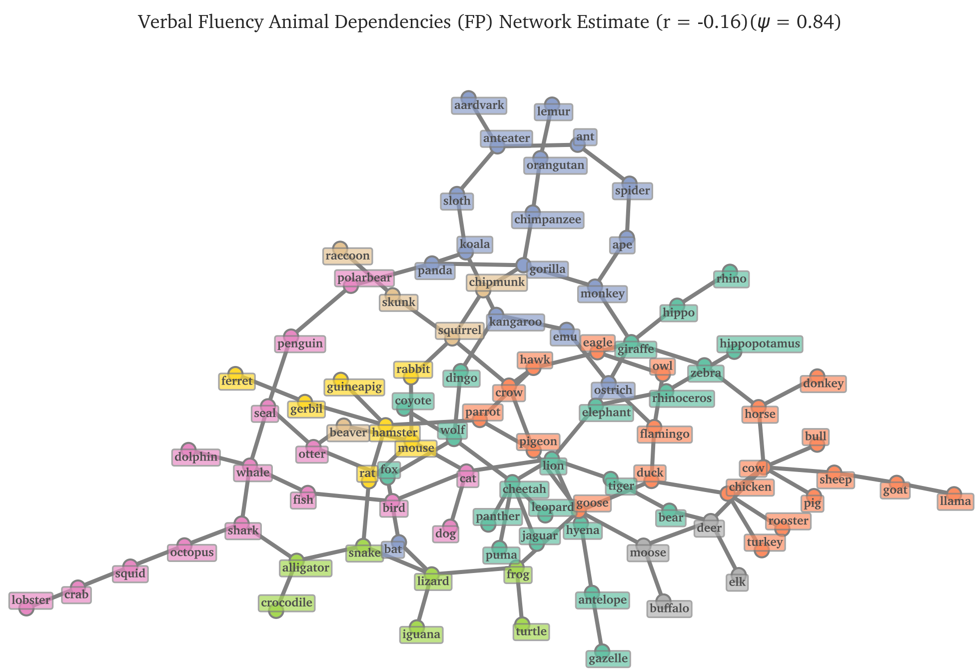
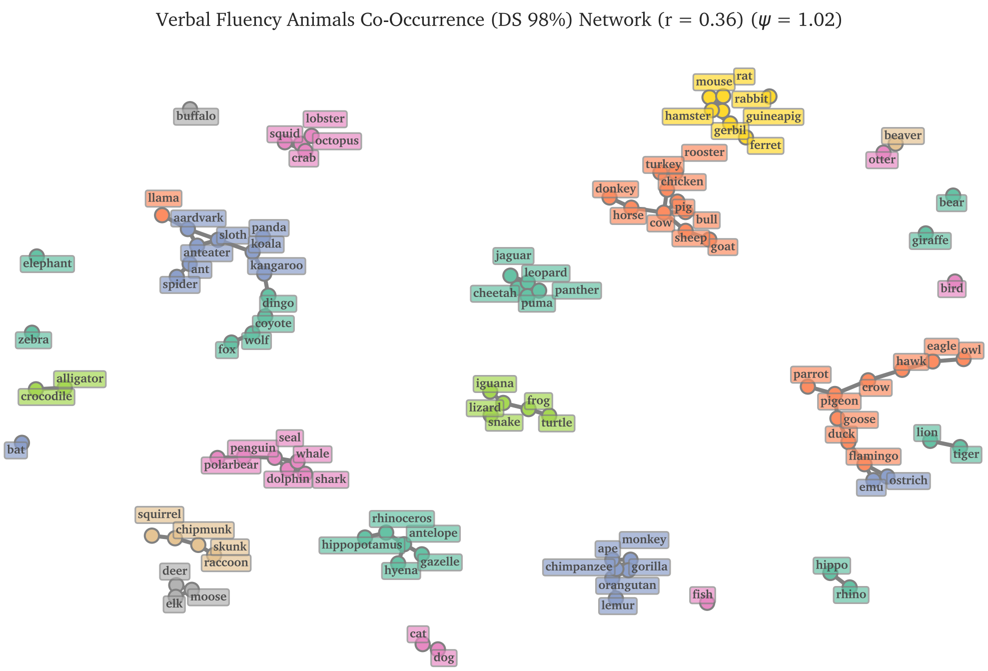
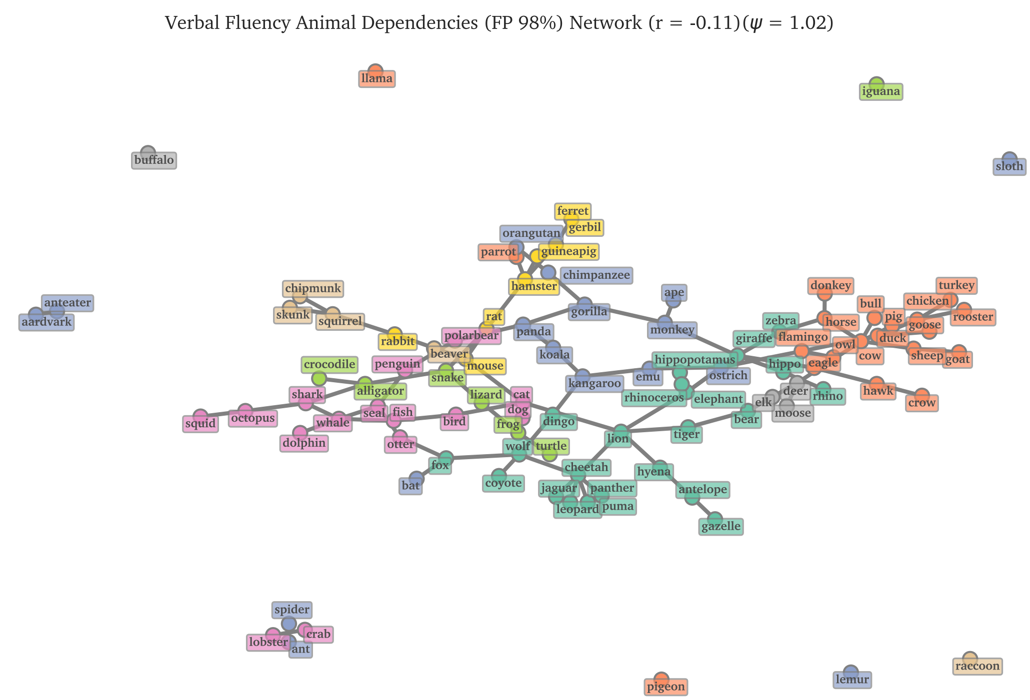


Figure 9.4: Comparison of backboning/dependency recovery methods tested vs. Forest Pursuit

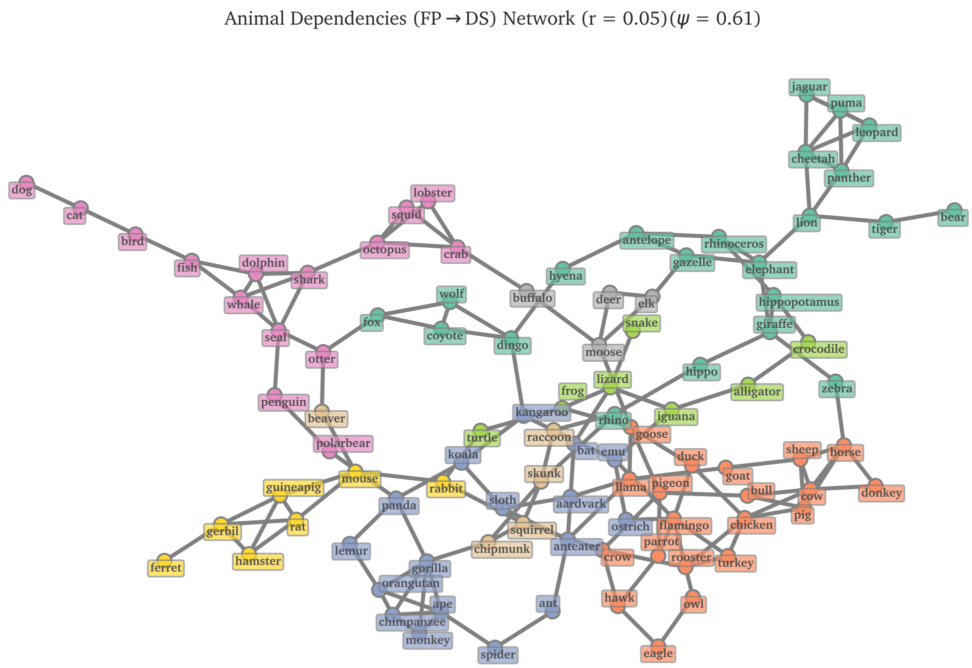


(a) co-occurrence methods will retain local communities at the cost of global structure

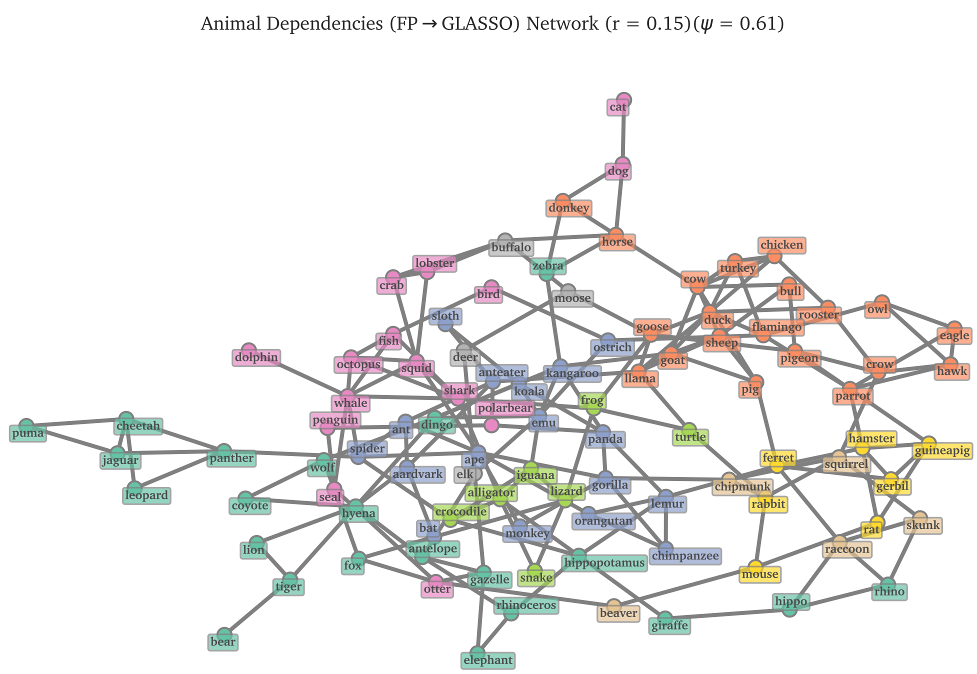


(b) dependency network drops rarer nodes from the preserved central structure at higher uncetainty cutoffs

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.

Bibliography

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

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

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

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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].

[19] 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.

[20] 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.

[21] 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.

[22] 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.