Detecting Path Anomalies in Sequential Data on Networks

Tim LaRock
larock.t@husky.neu.edu
In collaboration with





Vahan Nanumyan | Ingo Scholtes | Giona Casiraghi | Tina Eliassi-Rad | Frank Schweitzer

netsi
Northeastern University
Network Science Institute

This Talk

Motivation: Understanding mechanisms behind sequential data on networks

Today:

Motivate the study of **path anomalies**

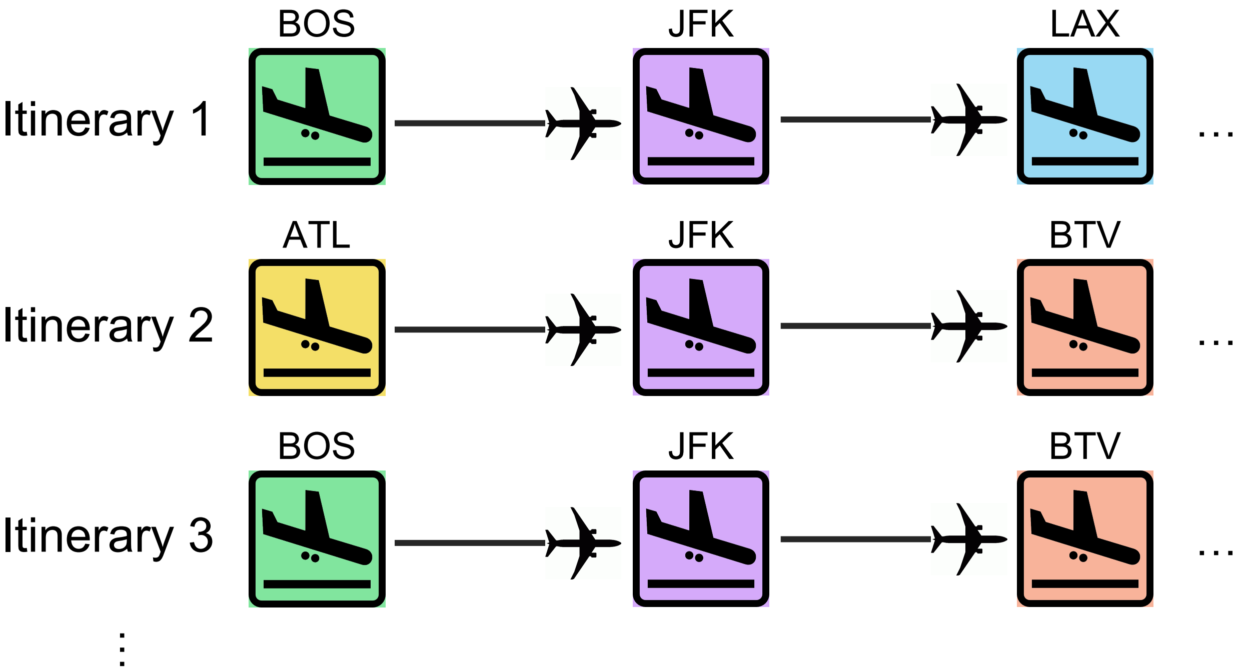
Introduce **de Bruijn graph** representation of sequential data

Define a tractable **null model** to measure deviation of path data from expectation

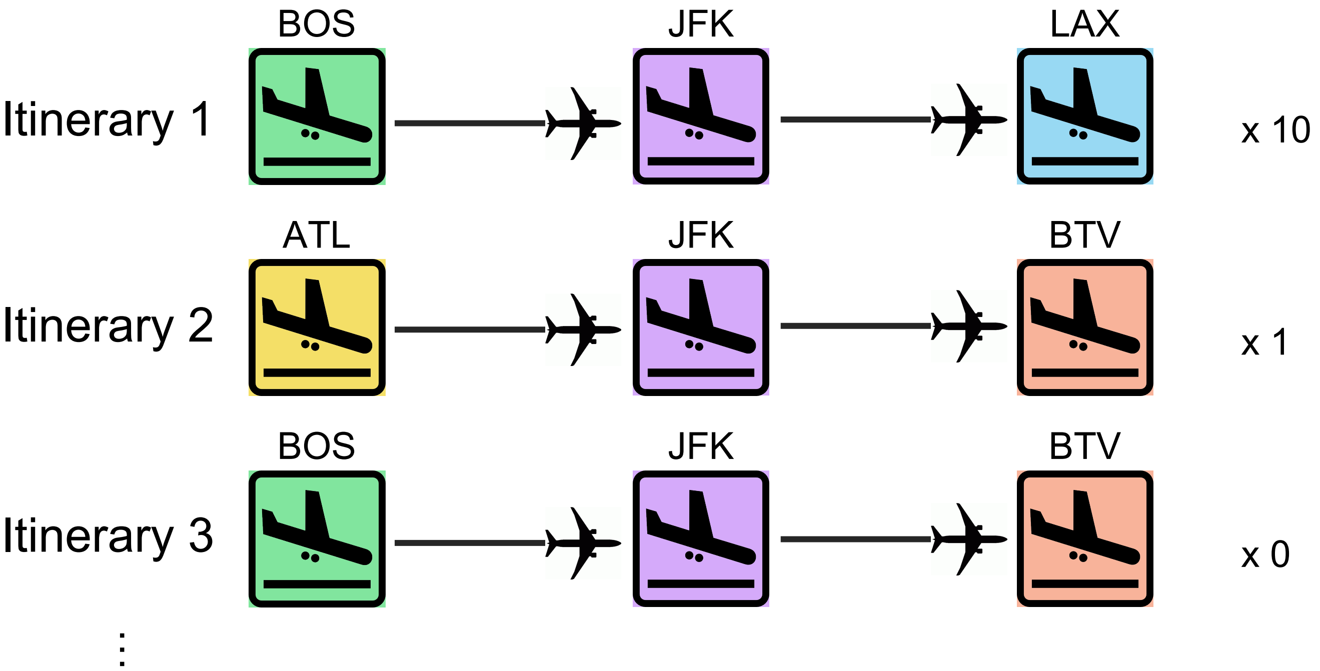
Application of methodology to a real system

Intuitive Example: Passenger Flight Data

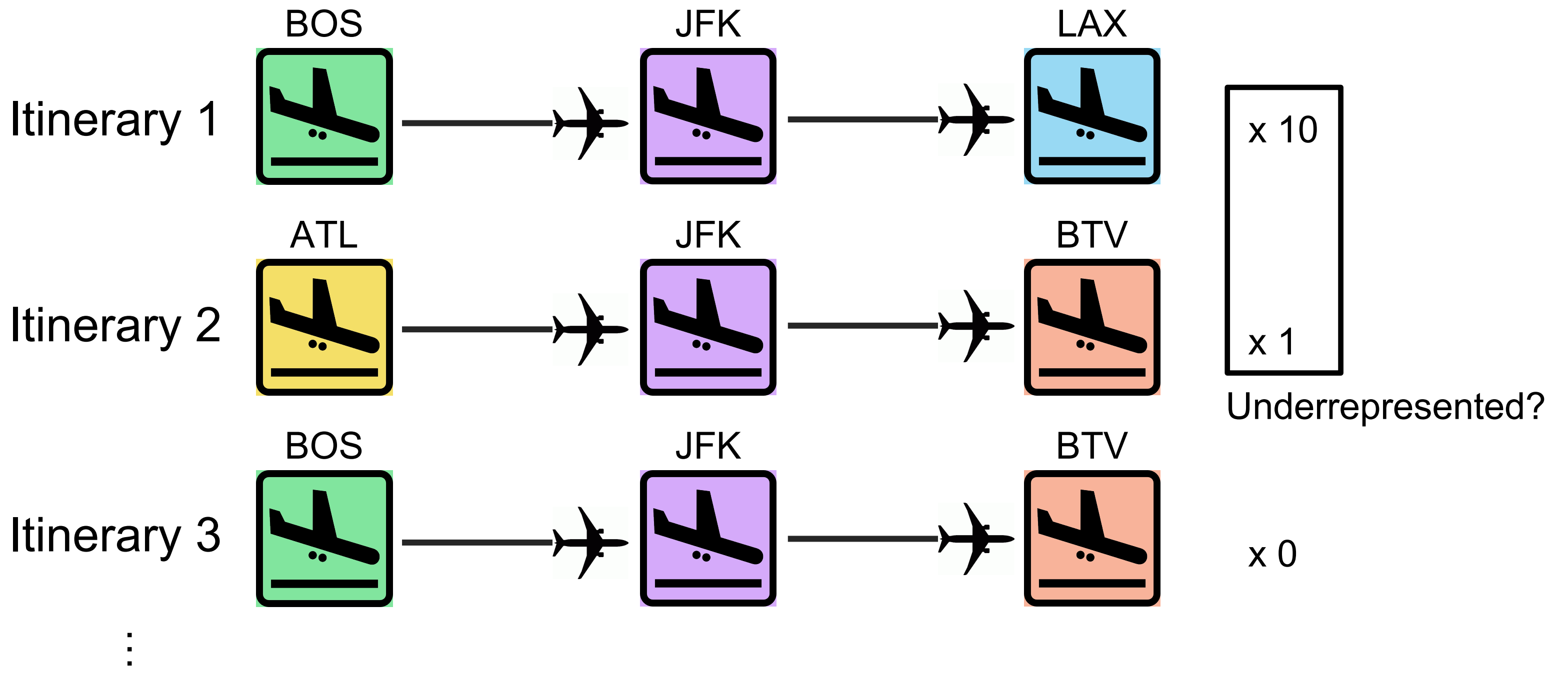
Intuitive Example: Passenger Flight Data



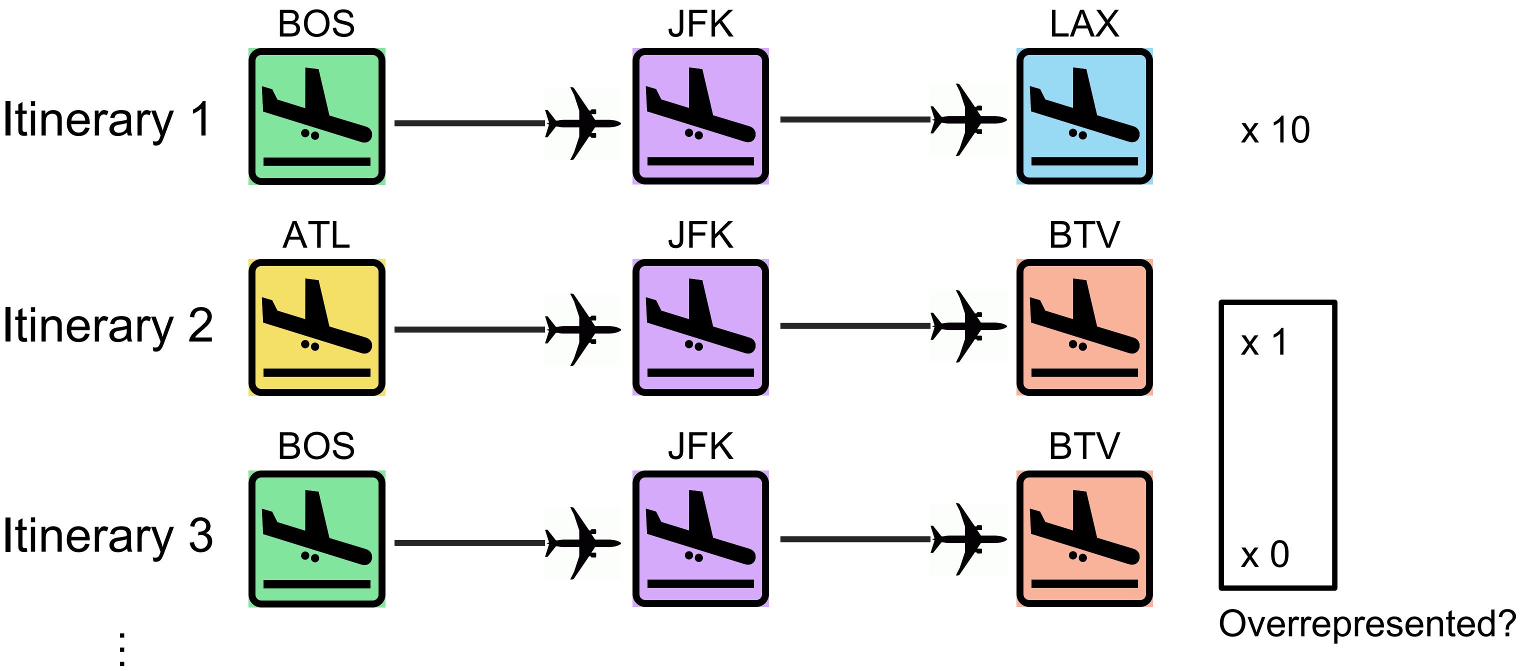
Intuitive Example: Passenger Flight Data



Intuitive Example: Passenger Flight Data

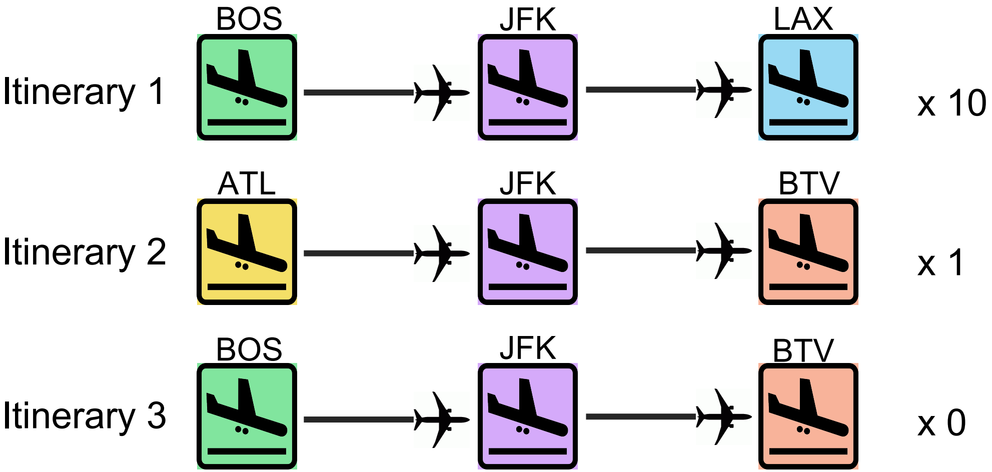


Intuitive Example: Passenger Flight Data



Research Question

Given this pathway dataset, can we determine whether…



is **significantly overrepresented?**

is **significantly underrepresented?**

Problem: Path anomaly detection

For a given graph G and integer k, identify paths of length k through G whose observed frequencies deviate significantly from random expectation in a $(k\text{-}1)$-order model of paths through G.

Problem: Path anomaly detection

For a given graph G and integer k, identify paths of length k through G whose observed frequencies deviate significantly from random expectation in a $(k\text{-}1)$-order model of paths through G.

When $k=2$, this corresponds to comparing a random walk with a single step of memory to a memoryless (Markovian) random walk on G.

Toy Example

Three Goals:

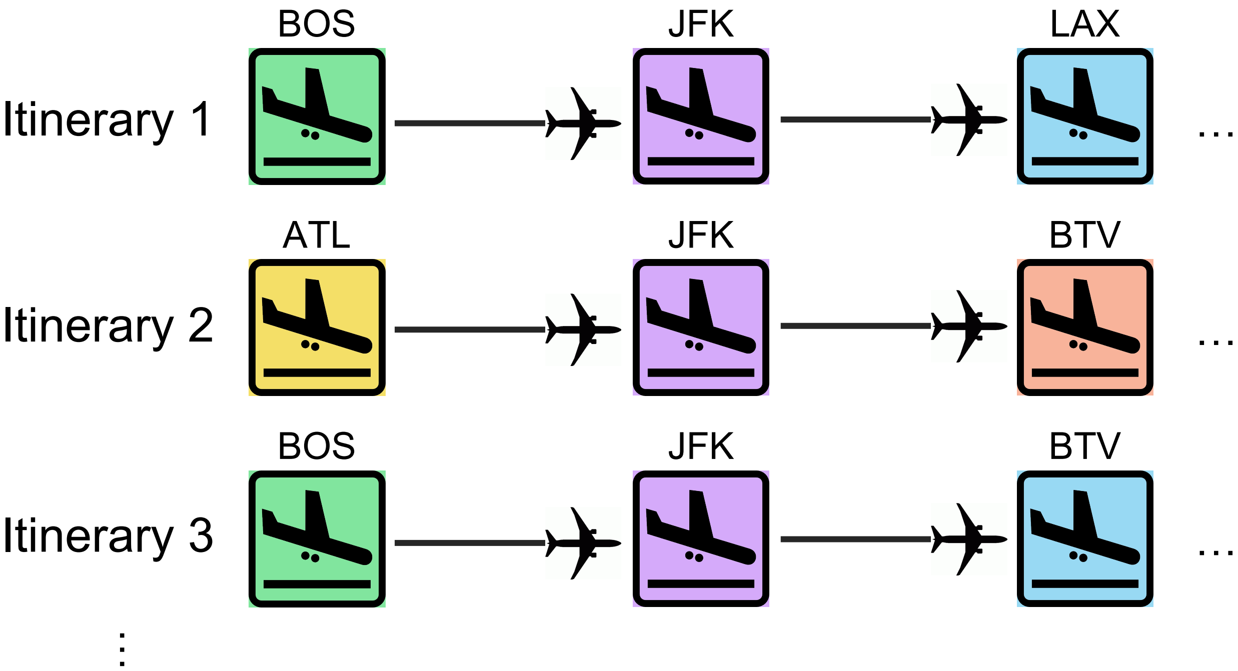
1. Introduce de Bruijn graphs as representations of sequential data

2. Show how path anomalies emerge in a simple setting

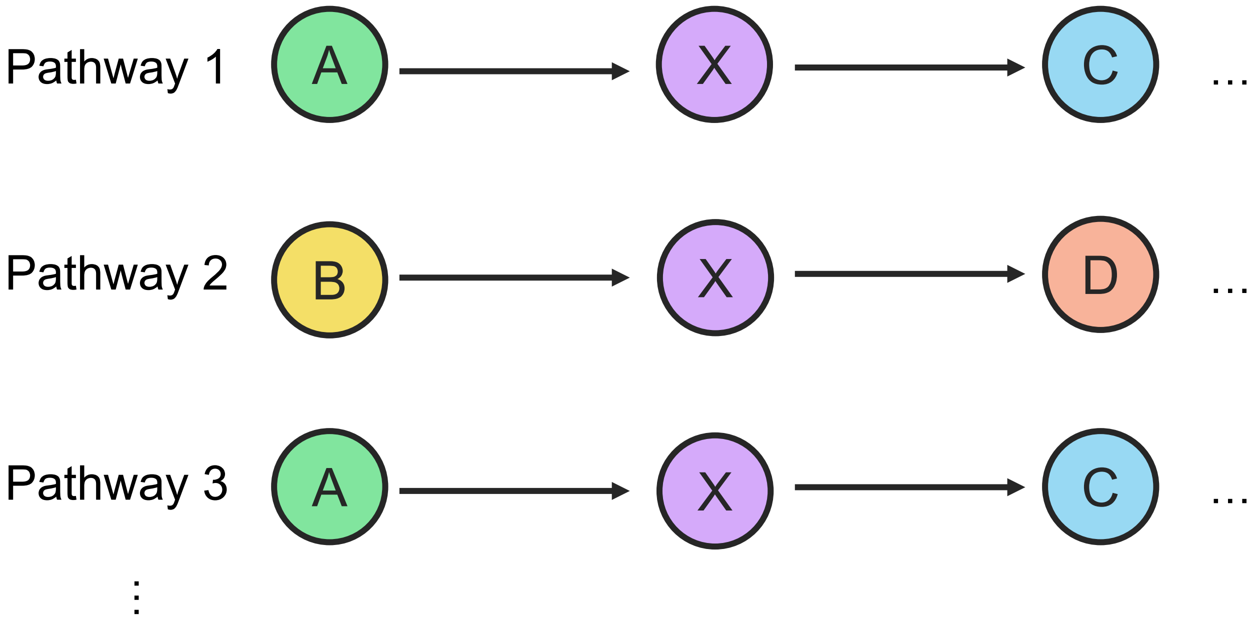
3. Show how path anomalies can be detected through a random walk simulation approach

(Spoiler: Simulation approach is infeasible for real world datasets!)

Toy Example

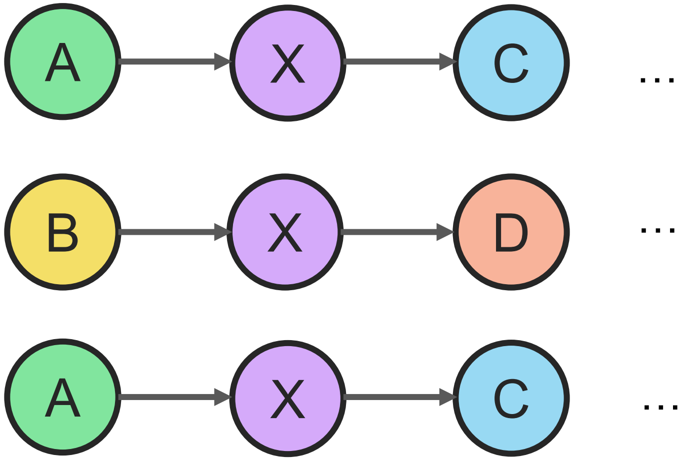

Itinerary 1
BOS
JFK
LAX
…
Itinerary 2
ATL
JFK
BTV
…
Itinerary 3
BOS
JFK
BTV
…
⋮

Toy Example



Toy Example: Data

$$S = \{ \quad \begin{matrix} p_1 \\ p_2 \\ p_3 \\ \vdots \\ p_N \end{matrix}$$



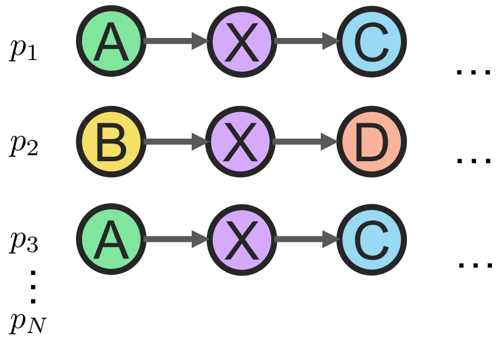
Toy Example: Data

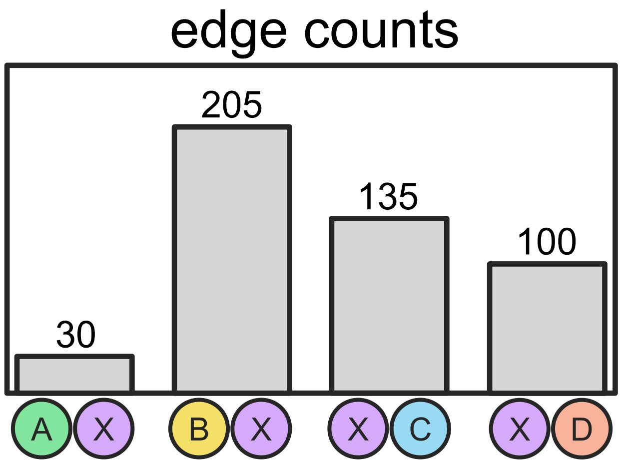
$$S = \left\{ \begin{array}{ll} p_1 & A \rightarrow X \rightarrow C \quad \cdots \\ p_2 & B \rightarrow X \rightarrow D \quad \cdots \\ p_3 & A \rightarrow X \rightarrow C \quad \cdots \\ \vdots & \\ p_N & \end{array} \right.$$

$$\text{Total number of paths } N = |S| = 235$$

Toy Example: Data to (first-order) graph

$$S = \{$$

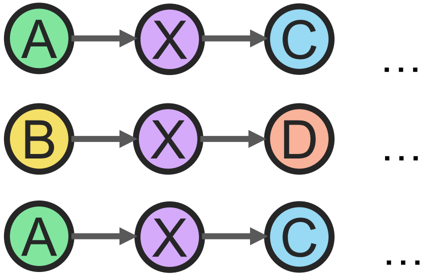


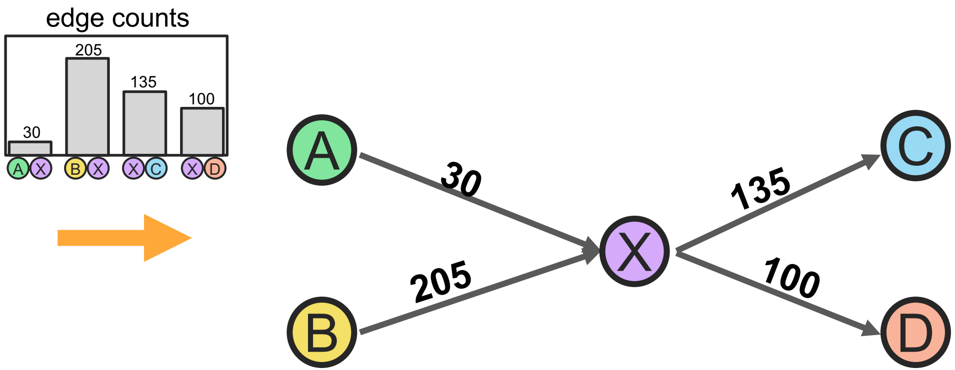


Total number of paths $N = |S| = 235$

Toy Example: Data to (first-order) graph

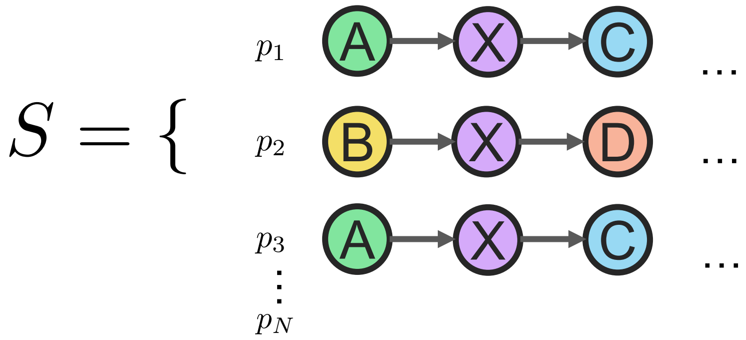
$S = \{$ p_1, p_2, p_3, $\vdots$, p_N



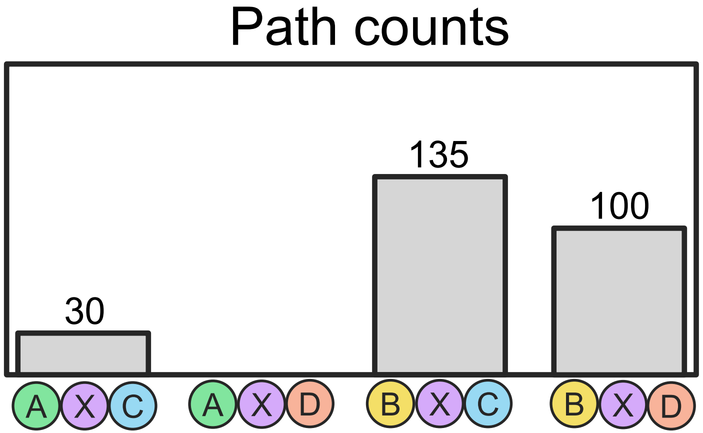


Total number of paths $N = |S| = 235$

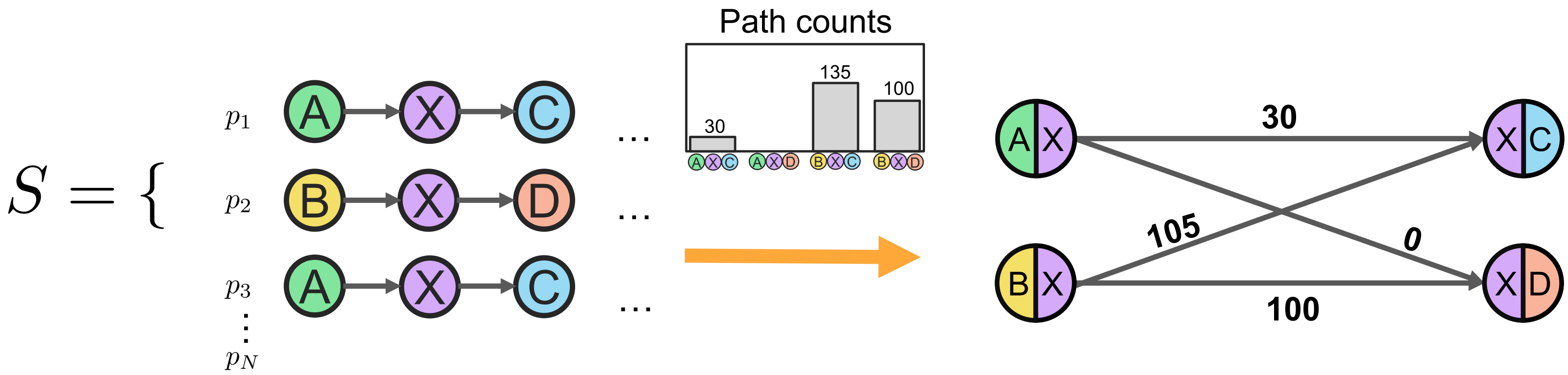
Toy Example: Data to 2nd order de Bruijn graph



Total number of paths $N = |S| = 235$



Toy Example: Data to 2nd order de Bruijn graph



Total number of paths $N = |S| = 235$

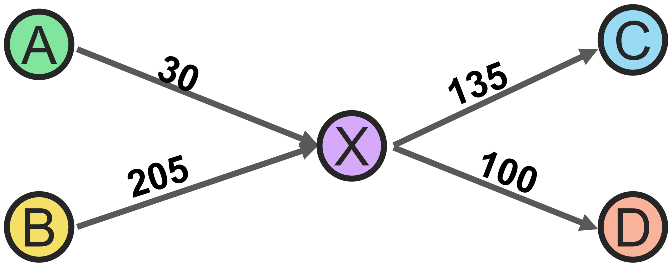
Toy Example: Path Anomalies via Simulations

For a given graph G and integer k, identify paths of length k through G whose observed frequencies deviate significantly from random expectation in a $(k-1)$-order model of paths through G.

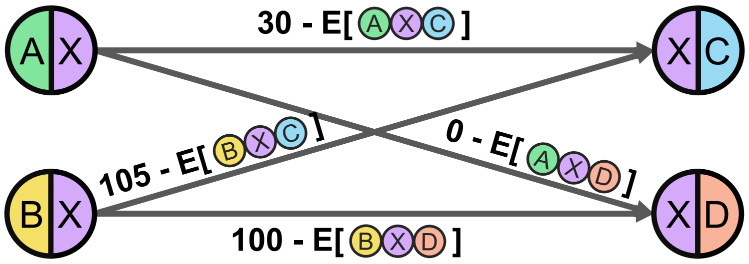
Toy Example: Path Anomalies via Simulations

For a given graph G and integer k, identify paths of length k through G whose observed frequencies deviate significantly from random expectation in a $(k\text{-}1)$-order model of paths through G.

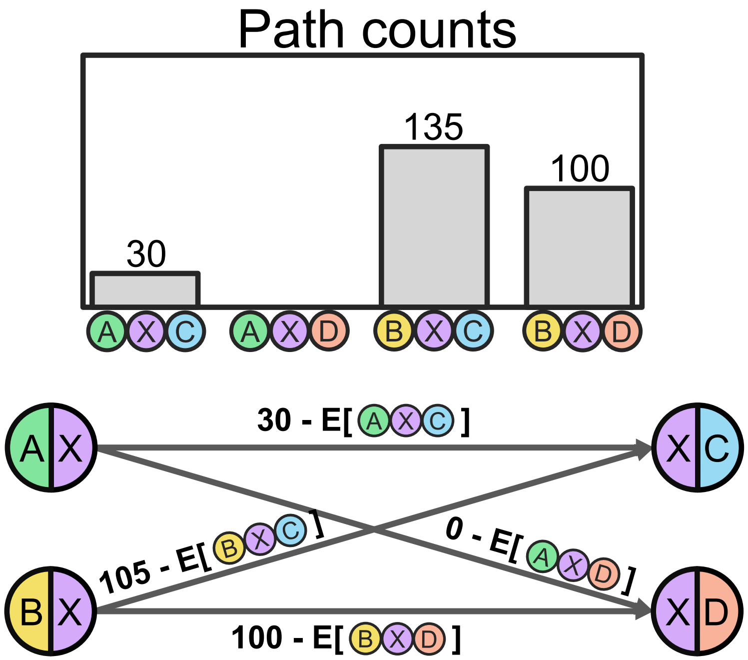
Simulate many random walk datasets

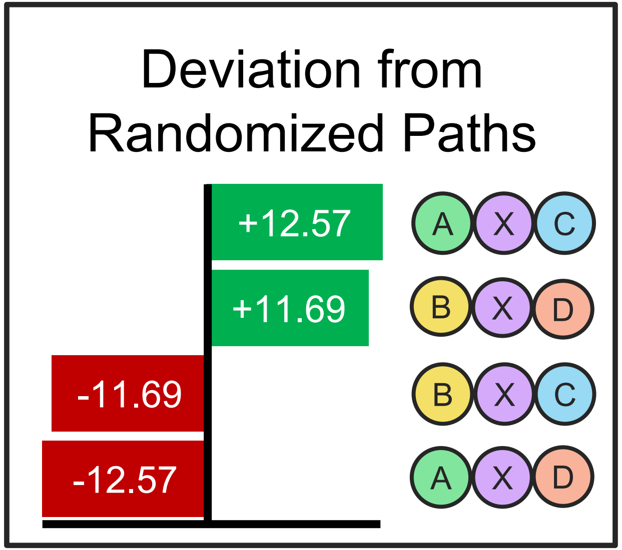


Compute expected frequency of each pathway and subtract from observed value



Toy Example: Path Anomalies via Simulations





Challenges

Path Anomaly Detection: Challenges

Detecting path anomalies via simulations → computationally intensive

Result is expected value, no concrete notion of significance

Alternative: detect path anomalies analytically by developing a tractable null model

Null Model: Challenges

Traditional null models (e.g. configuration model) cannot be applied directly

Null Model: Challenges

Traditional null models (e.g. configuration model) cannot be applied directly

Edges between higher-order nodes can not be randomized by stub-matching across whole network (need suffix-prefix matching)

Null Model: Challenges

Traditional null models (e.g. configuration model) cannot be applied directly

Edges between higher-order nodes can not be randomized by stub-matching across whole network (need suffix-prefix matching)

Need to randomize *edge weight distribution* in de Bruijn graph models, since connectivity structure is fixed by 1st-order topology

HYPA: Efficient Detection of Path Anomalies

Generalized Hypergeometric Ensemble

Generalization of the configuration model to weighted, directed networks.

Generalized Hypergeometric Ensemble

Generalization of the configuration model to weighted, directed networks.

Fixes the *expected* weight of every node, rather than the *exact* degree sequence.

Generalized Hypergeometric Ensemble

Generalization of the configuration model to weighted, directed networks.

Fixes the *expected* weight of every node, rather than the *exact* degree sequence.

Intuition: Urn problem where each pair of nodes *i, j* that can possibly connect is assigned a color, and K_{ij} balls are added, where K_{ij} is the total possible multi-edges between the nodes (e.g. $K_{ij} = k_i^{\text{out}} k_j^{\text{in}}$, as in the config model).

Generalized Hypergeometric Ensemble

Generalization of the configuration model to weighted, directed networks.

Fixes the *expected* weight of every node, rather than the *exact* degree sequence.

Intuition: Urn problem where each pair of nodes *i, j* that can possibly connect is assigned a color, and K_{ij} balls are added, where K_{ij} is the total possible multi-edges between the nodes (e.g. $K_{ij} = k_i^{\text{out}} k_j^{\text{in}}$, as in the config model).

To sample from the ensemble, draw $m = \Sigma_{ij} W_{ij}$ (total observed edges) multi-edges randomly from the urn.

Putting it all together: HYPA scores

$$\mathrm{HYPA}^{(k)}(\vec{v}, \vec{w}) := \Pr(X_{\vec{v}\vec{w}} \leq f(\vec{v}, \vec{w}))$$

If "close" to **0**, then the pathway is **underrepresented**.

If "close" to **1**, then pathway is **overrepresented**.

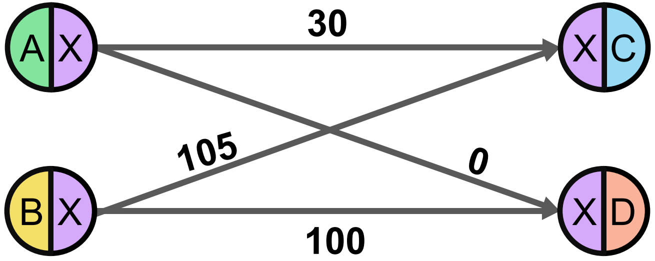
Putting it all together: HYPA scores

$$\mathrm{HYPA}^{(k)}(\vec{v},\vec{w}) := \Pr\left(X_{\vec{v}\vec{w}} \le f(\vec{v},\vec{w})\right)$$

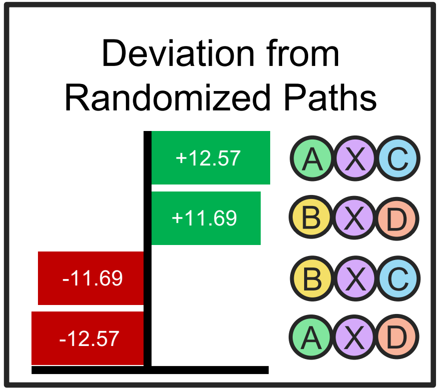
If "close" to **0**, then the pathway is **underrepresented**.

If "close" to **1**, then pathway is **overrepresented**.

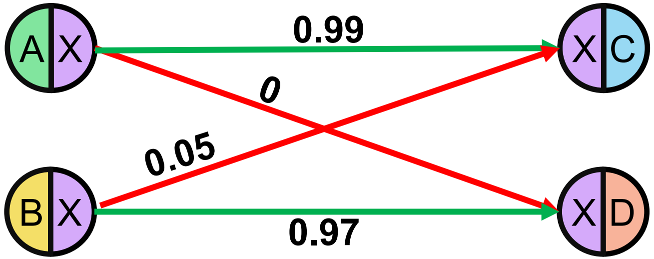
Observed Data



Simulation Result



HYPA Scores



Application to Flight Data

Airlines

Airlines

Hypotheses:

1. Return flights should be over-represented, since people most often travel round trip.

Airlines: Return trips are over-represented

α	**Return**	**Non-return**
0.05	0.915	0.340
0.01	0.851	0.130
0.001	0.760	0.023
0.0001	0.688	0.004
0.00001	0.628	0.001

Fraction of over-represented return/non-return flights for various discrimination thresholds.

Airlines: Return trips are over-represented

α	Return	Non-return
0.05	0.915	0.340
0.01	0.851	0.130
0.001	0.760	0.023
0.0001	0.688	0.004
0.00001	0.628	0.001

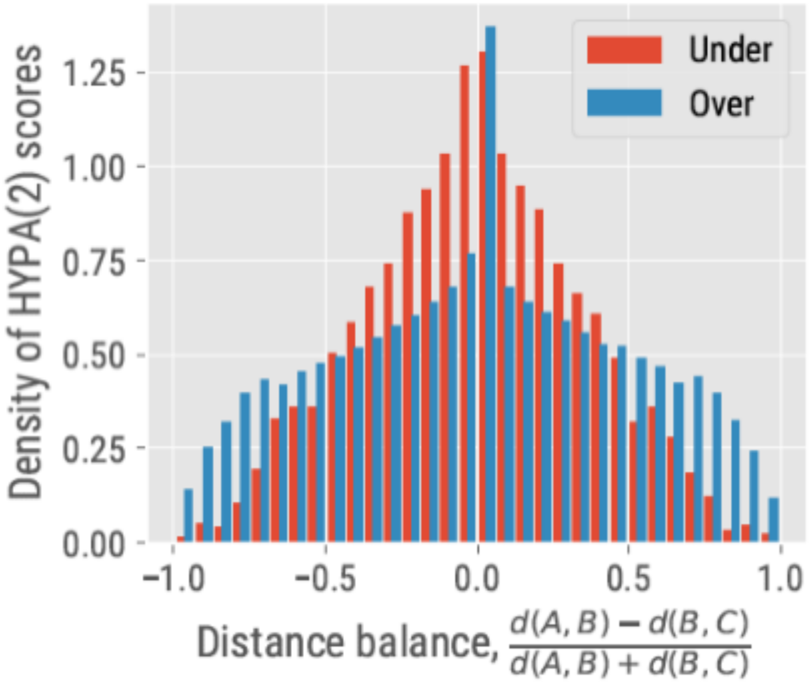
Fraction of over-represented return/non-return flights for various discrimination thresholds.

Airlines

Hypotheses:

1. Return flights should be over-represented, since people most often travel round trip.
2. Over-represented non-return flights are due to regional/national hubs, since people need to fly from small airports → regional hub → large airport.

Airlines: Trip Balance

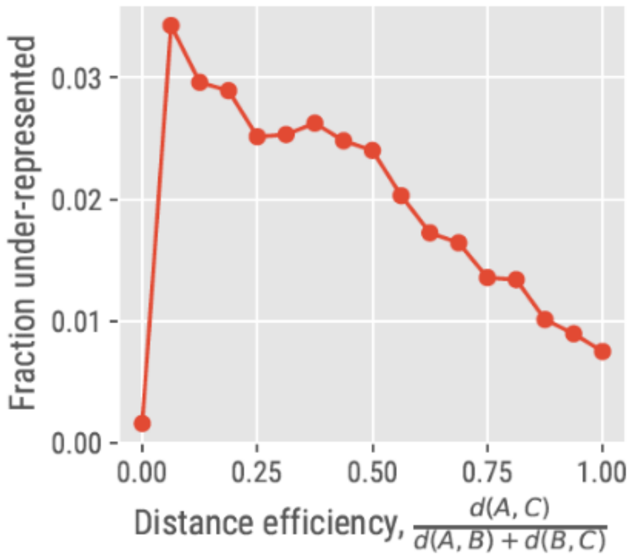


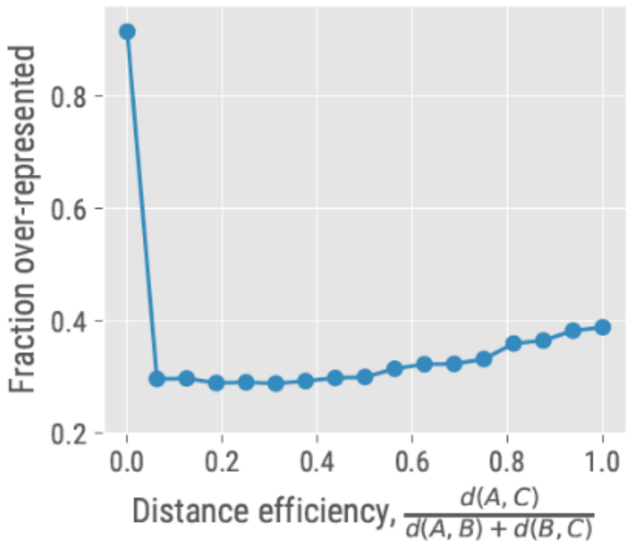
Airlines

Hypotheses:

1. Return flights should be over-represented, since people most often travel round trip.
2. Over-represented non-return flights are due to regional/national hubs, since people need to fly from small airports → regional hub → large airport.
3. “Efficient” paths are more likely to be over-represented.

Airlines: Efficiency





Northeastern University
Network Science Institute

Thanks!

Tim LaRock
larock.t@husky.neu.edu
tlarock.github.io

References

Scholtes, Ingo. "When is a network a network?: Multi-order graphical model selection in pathways and temporal networks." *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017.

Casiraghi, Giona, et al. "Generalized hypergeometric ensembles: Statistical hypothesis testing in complex networks." *arXiv preprint arXiv:1607.02441* (2016).

Transport for London Open Data: https://tfl.gov.uk/info-for/open-data-users/

Definition: kth-order de Bruijn Graph

For a given graph $G = (V, E)$ and positive integer k we define a k-th order De Bruijn graph of paths in G as a graph $G^k = (V^k, E^k)$, where (i) each node $\vec{v} := v_0 v_1 \vec{\ldots} v_{k-1} \in V^k$ is a path of length $k-1$ in G, and (ii) $(\vec{v}, \vec{w}) \in E^k$ iff $v_{i+1} = w_i$ for $i = 0, \ldots, k-2$.

Hypergeometric Ensemble

$$m = \Sigma_{ij} W_{ij}$$

$$K_{ij} = k_i^{\text{out}} k_j^{\text{in}}$$

$$Pr(X_{vw} = f(v,w)) \propto \binom{K_{vw}}{f(v,w)} \binom{\sum_{ij} K_{ij} - K_{vw}}{m - f(v,w)}$$

$$m = \Sigma_{ij} W_{ij}$$

$$K_{ij} = k_i^{\text{out}} k_j^{\text{in}}$$

Hypergeometric Ensemble

$$Pr(\underbrace{X_{vw} = f(v,w)}) \propto \binom{K_{vw}}{f(v,w)} \binom{\sum_{ij} K_{ij} - K_{vw}}{m - f(v,w)}$$

Probability of observing frequency *f(v,w)* given the entire weighted network structure.

$$m = \Sigma_{ij} W_{ij}$$

Hypergeometric Ensemble

$$K_{ij} = k_i^{\text{out}} k_j^{\text{in}}$$

$$Pr(X_{vw} = f(v,w)) \propto \underbrace{\binom{K_{vw}}{f(v,w)}} \binom{\sum_{ij} K_{ij} - K_{vw}}{m - f(v,w)}$$

Number of ways to pick *f(v,w)* multiedges from K_{vw} possible.

Hypergeometric Ensemble

$$m = \Sigma_{ij} W_{ij}$$

$$K_{ij} = k_i^{\text{out}} k_j^{\text{in}}$$

$$Pr(X_{vw} = f(v,w)) \propto \binom{K_{vw}}{f(v,w)} \binom{\sum_{ij} K_{ij} - K_{vw}}{\underbrace{m - f(v,w)}_{\text{Number of ways to pick everything else.}}}$$

Putting it all together: HYPA scores

$$\mathrm{HYPA}^{(k)}(\vec{v}, \vec{w}) := \Pr\left(X_{\vec{v}\vec{w}} \leq f(\vec{v}, \vec{w})\right)$$

Pseudocode

Algorithm 1 ComputeHYPA(S, k): *Compute kth order HYPA scores for sequence dataset S.*

Input: S (sequences), k (desired order)
Output: $\mathrm{HYPA}^{(k)}$ score for all k-th order paths

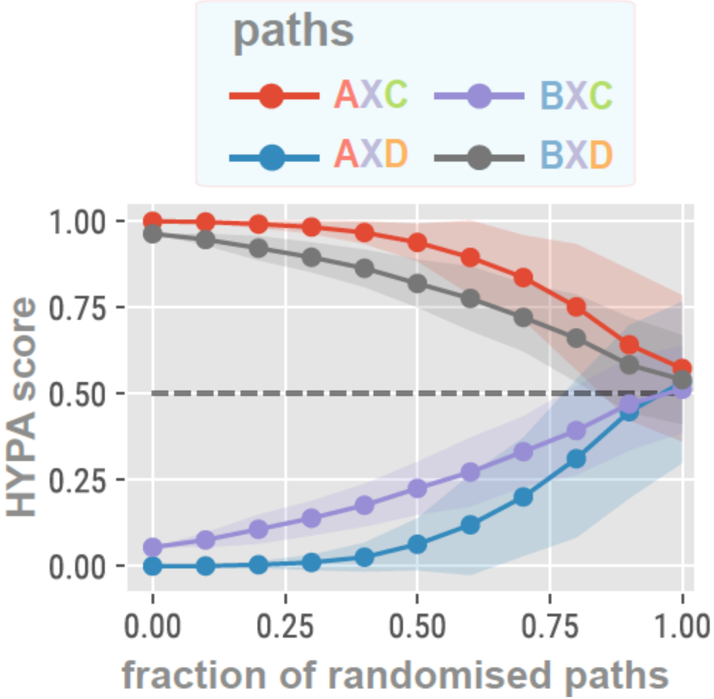
1: $G^k \leftarrow$ DeBruijnGraph(S, k) # Construct kth order graph

2: $\Xi \leftarrow$ fitXi(G^k, tolerance) # Optimization (Algorithm 2 in Appendix A.1)

3: **for** $(\vec{v}, \vec{w}) \in G^k$ **do**
4: $\quad \mathrm{HYPA}^{(k)}(\vec{v}, \vec{w}) \leftarrow \Pr(x_{vw} \leq (\vec{v}, \vec{w}) \mid m, \Xi)$
$\quad$ # Compute CDF
5: **return** $\mathrm{HYPA}^{(k)}$

Validation

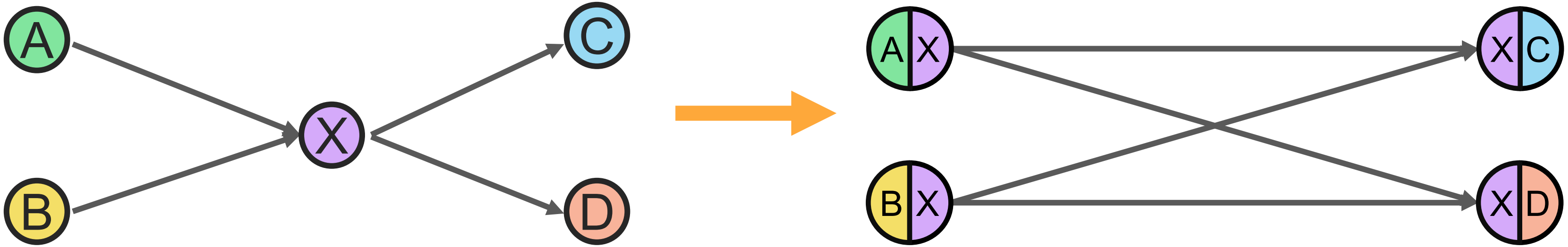
Noise via Path Randomization



Synthetic Anomalies: Setup

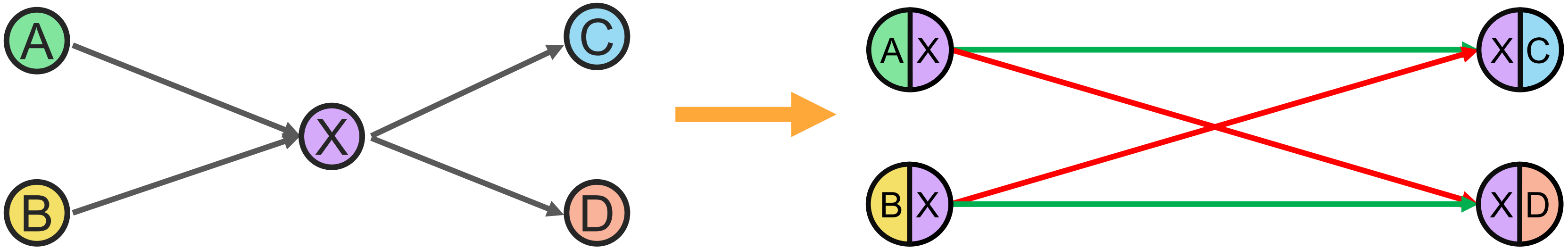
Synthetic Anomalies: Setup

Start with an arbitrary first order topology, then construct the *kth*-order de Bruijn graph



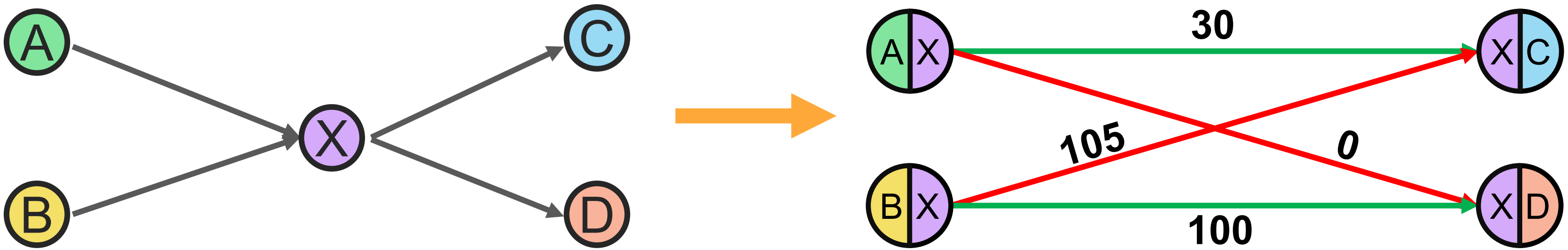
Synthetic Anomalies: Setup

Randomly choose some edges to label over-represented



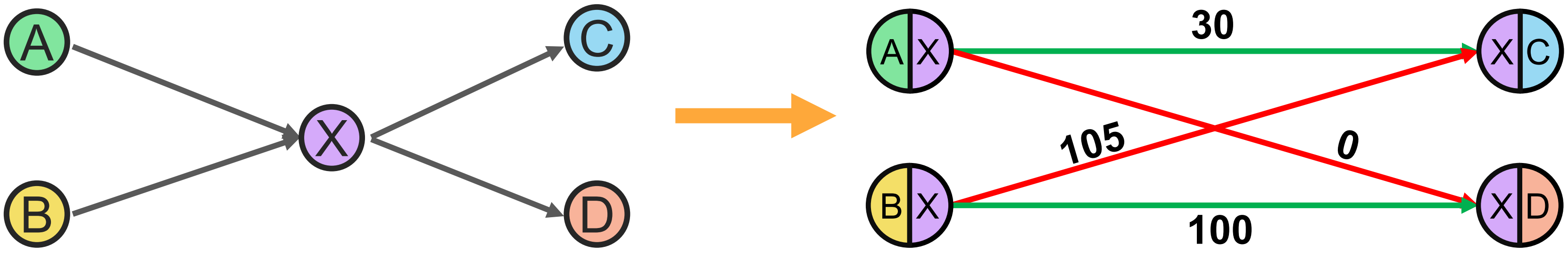
Synthetic Anomalies: Setup

Assign heterogeneous weights based on label



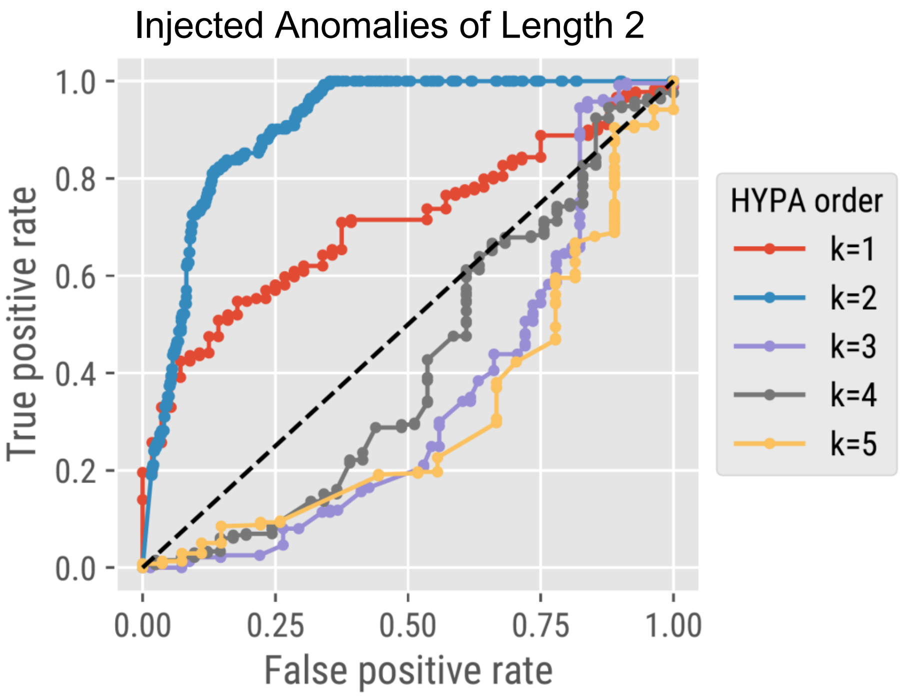
Synthetic Anomalies: Setup

Generate paths via random walks on this model, then evaluate ability of HYPA to detect injected anomalies (binary classifier).



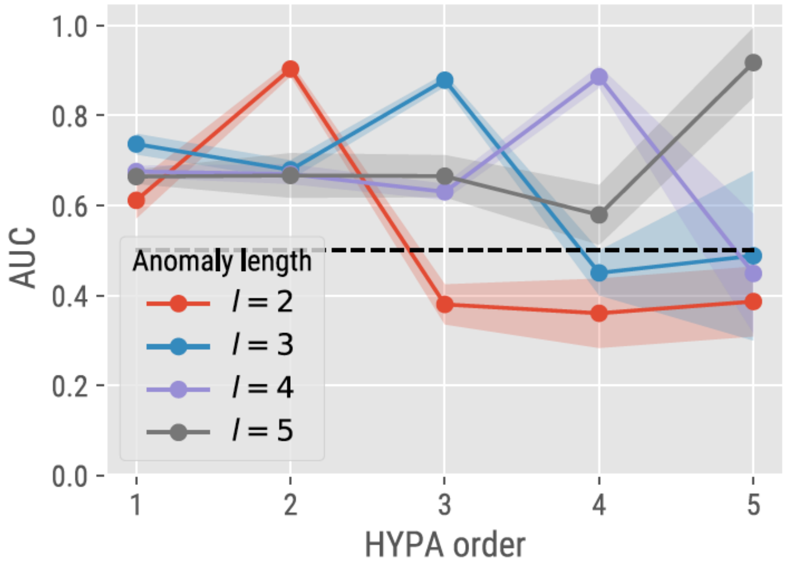
Synthetic Anomalies: ROC Example

Synthetic Anomalies: ROC Example



Northeastern University
Network Science Institute

Synthetic Anomalies: AUC Results



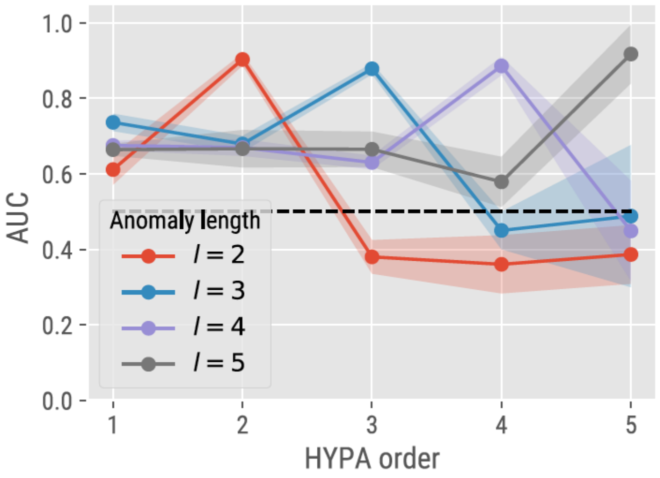
Naïve Baseline Comparison

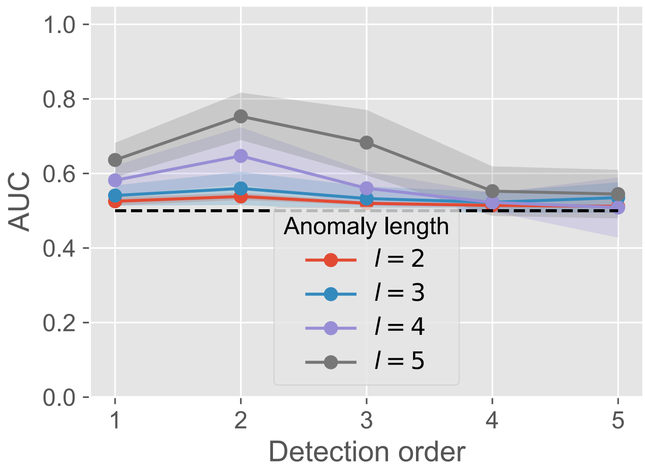
Frequency-Based Anomaly Detection (FBAD)

Compute mean, μ, and standard deviation, σ, of kth order edge weights

Given scaling factor α, label edges with frequencies larger than $\mu + \sigma\alpha$ over-represented, and smaller than $\mu - \sigma\alpha$ under-represented

Synthetic Anomalies

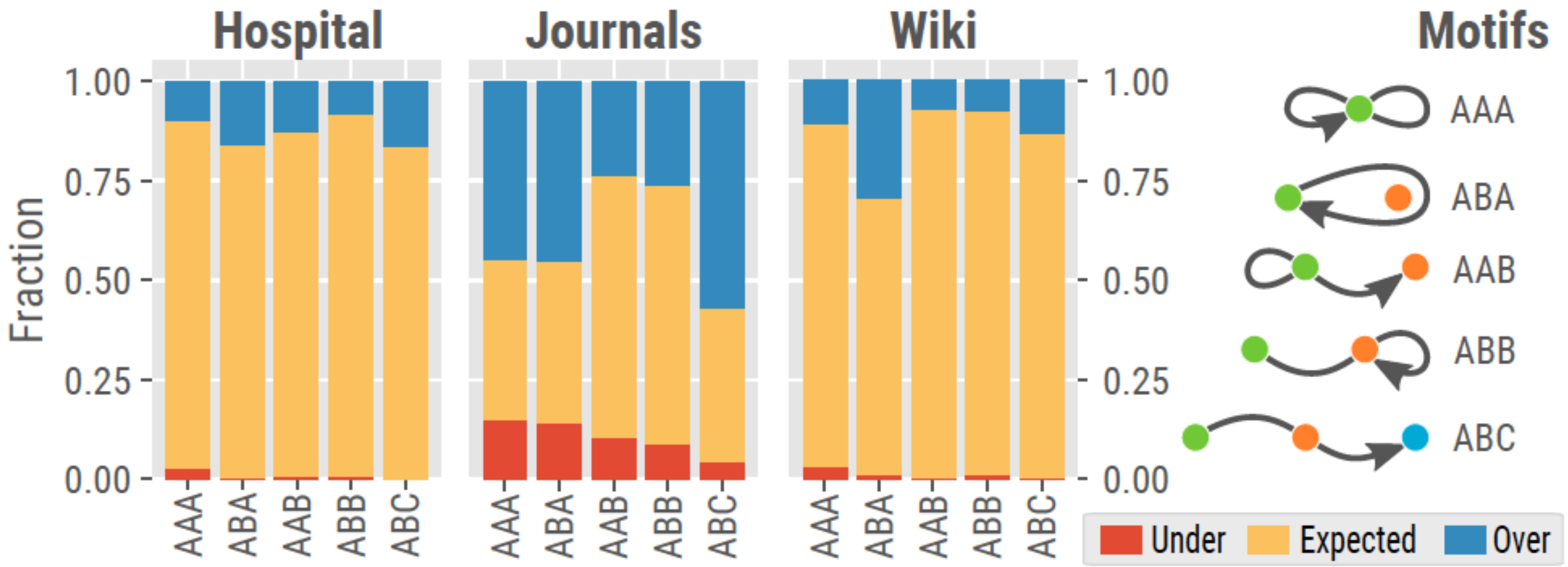




Real Data

	Topology		Sequences			
Data	Nodes	Edges	Total	Unique	$l^{\max}$	$\langle l \rangle$
Tube	268	646	4295731	67015	35	6.75
Flights	382	6933	185871	88539	10	2.48
Journals	283	1743	480496	309565	35	14.8
Hospital	75	1138	28422	2561	5	1.19
Wiki	100	1598	29682	7431	21	1.64

Exploring Motifs



Case Study: London Tube

Data:

- Origin → destination statistics between London Tube stations
    - (origin, destination, #observations)
- Shortest paths between stations
    - Assume people follow shortest paths

London Tube

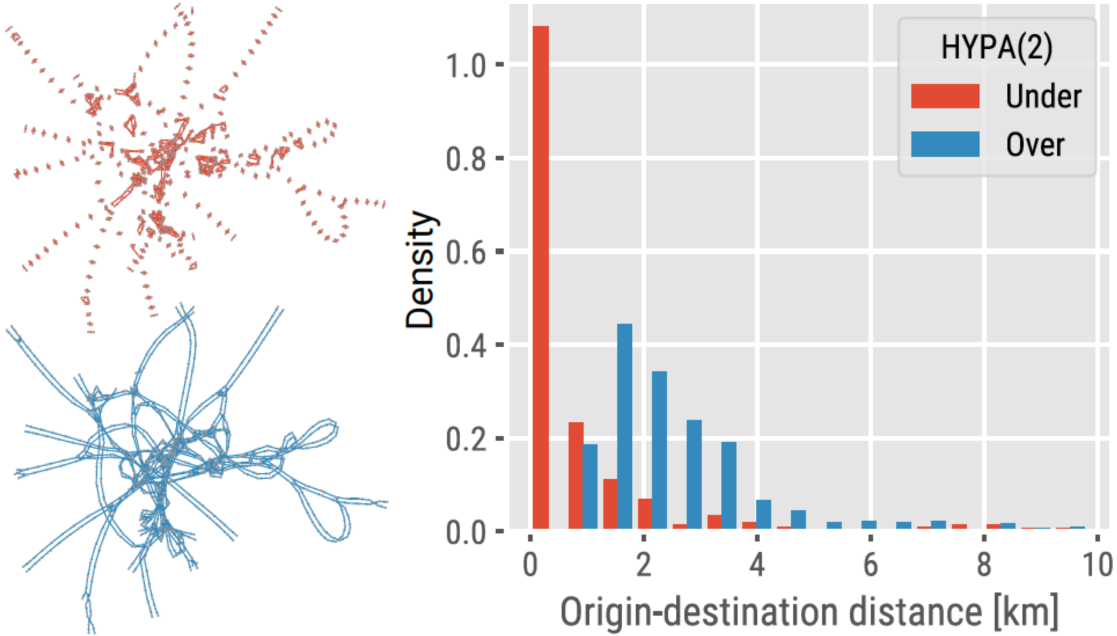
Hypothesis:

- People typically use public transportation to travel large geographic distances
- *Overrepresented* pathways should cover *larger* distances

Test:

- Measure distance between every station
- For 2nd order transitions A-B-C, compute distance between nodes A and C
- Analyze distributions of distance in over vs. under represented transitions
    - Expect to see distribution shifted towards higher values for over-represented transitions

London Tube


HYPA(2)
Under
Over
Density
Origin-destination distance [km]
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10

London Tube

$\mathrm{HYPA}^{(k)}$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
Under [km]	0.00	2.38	3.29	4.60	5.43
Over [km]	2.20	2.93	3.79	5.21	5.63
***p*-value**	$< 10^{-170}$	$< 10^{-7}$	$< 10^{-4}$	0.006	0.08

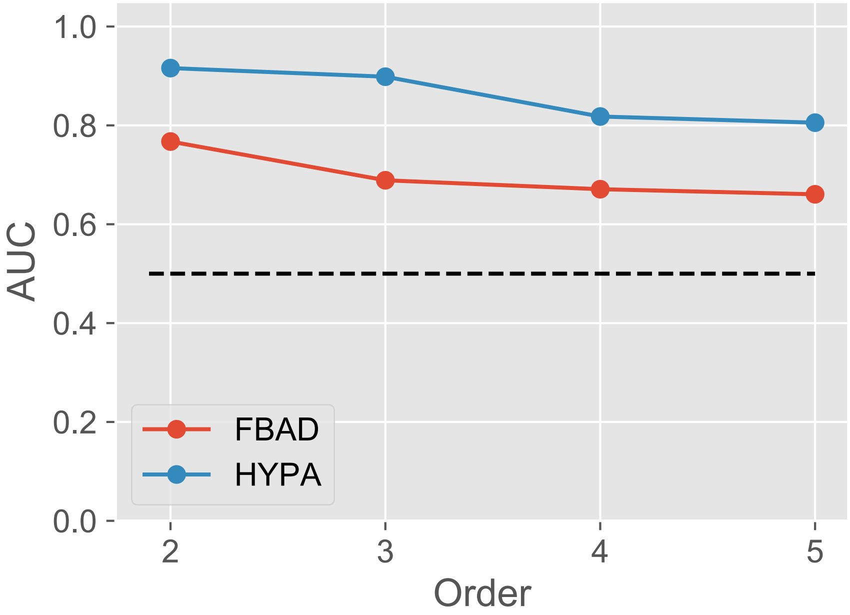
Median distance between source and destination nodes in under/over represented transitions.

Constructing Ground Truth

Construct ground truth based on the method discussed earlier:

- Randomize path data using k-1st order random walks
- Compute kth-order path statistics
- Repeat m times, noting the frequency of each path
- Estimate multinomial distribution and its CDF from these statistics
- If CDF(path) > threshold, label over-represented

Tube Data - Ground Truth



Hypergeometric Ensemble

$$m = \Sigma_{ij} W_{ij}$$

$$\Xi_{ij} = k_i^{out} k_j^{in}$$

$$\Pr\left(X_{\vec{v}\vec{w}} = f(\vec{v},\vec{w})\right) = \binom{\sum_{ij}\Xi_{ij}}{m}^{-1} \binom{\Xi_{vw}}{f(\vec{v},\vec{w})} \binom{\sum_{ij}\Xi_{ij} - \Xi_{vw}}{m - f(\vec{v},\vec{w})}$$

Northeastern University
Network Science Institute

netsi

$$m = \Sigma_{ij} W_{ij}$$

Hypergeometric Ensemble

$$\Xi_{ij} = k_i^{out} k_j^{in}$$

$$\Pr(X_{\vec{v}\vec{w}} = f(\vec{v}, \vec{w})) = \binom{\sum_{ij} \Xi_{ij}}{m}^{-1} \binom{\Xi_{vw}}{f(\vec{v}, \vec{w})} \binom{\sum_{ij} \Xi_{ij} - \Xi_{vw}}{m - f(\vec{v}, \vec{w})}$$

Probability of observing frequency $f(\vec{v}, \vec{w})$ given the entire weighted network structure.

Hypergeometric Ensemble

$$m = \Sigma_{ij} W_{ij}$$

$$\Xi_{ij} = k_i^{out} k_j^{in}$$

$$\Pr\left(X_{\vec{v}\vec{w}} = f(\vec{v}, \vec{w})\right) = \underbrace{\binom{\sum_{ij} \Xi_{ij}}{m}^{-1}}_{} \binom{\Xi_{vw}}{f(\vec{v}, \vec{w})} \binom{\sum_{ij} \Xi_{ij} - \Xi_{vw}}{m - f(\vec{v}, \vec{w})}$$

Normalization. Total number of ways to pick m multiedges from total possible.

Hypergeometric Ensemble

$$m = \Sigma_{ij} W_{ij}$$

$$\Xi_{ij} = k_i^{out} k_j^{in}$$

$$\Pr\left(X_{\vec{v}\vec{w}} = f(\vec{v},\vec{w})\right) = \binom{\sum_{ij}\Xi_{ij}}{m}^{-1} \binom{\Xi_{vw}}{f(\vec{v},\vec{w})} \binom{\sum_{ij}\Xi_{ij} - \Xi_{vw}}{m - f(\vec{v},\vec{w})}$$

Number of ways to pick $f(\vec{v},\vec{w})$ multiedges from $\Xi_{\vec{v}\vec{w}}$ possible.

$$m = \Sigma_{ij} W_{ij}$$

Hypergeometric Ensemble

$$\Xi_{ij} = k_i^{out} k_j^{in}$$

$$\Pr\left(X_{\vec{v}\,\vec{w}} = f(\vec{v}, \vec{w})\right) = \binom{\sum_{ij} \Xi_{ij}}{m}^{-1} \binom{\Xi_{vw}}{f(\vec{v}, \vec{w})} \underbrace{\binom{\sum_{ij} \Xi_{ij} - \Xi_{vw}}{m - f(\vec{v}, \vec{w})}}_{\text{Number of ways to pick everything else.}}$$