Social Network Analysis

Statistical Analysis and ERGM

Dr. Chun-Hsiang Chan

Department of Geography,
National Taiwan Normal University

Photo credit: midjourney

Outline

- Random Graph
  - Random Graph Models
  - Erdős–Rényi
  - Small World Models
  - Scale-Free Models
- Statistical Network Modeling
  - Why Exponential Random Graph Model
  - Logistic Regression
  - Exponential Random Graph Model (ERGM)
  - Predictors for ERGMs
- Paper Reading

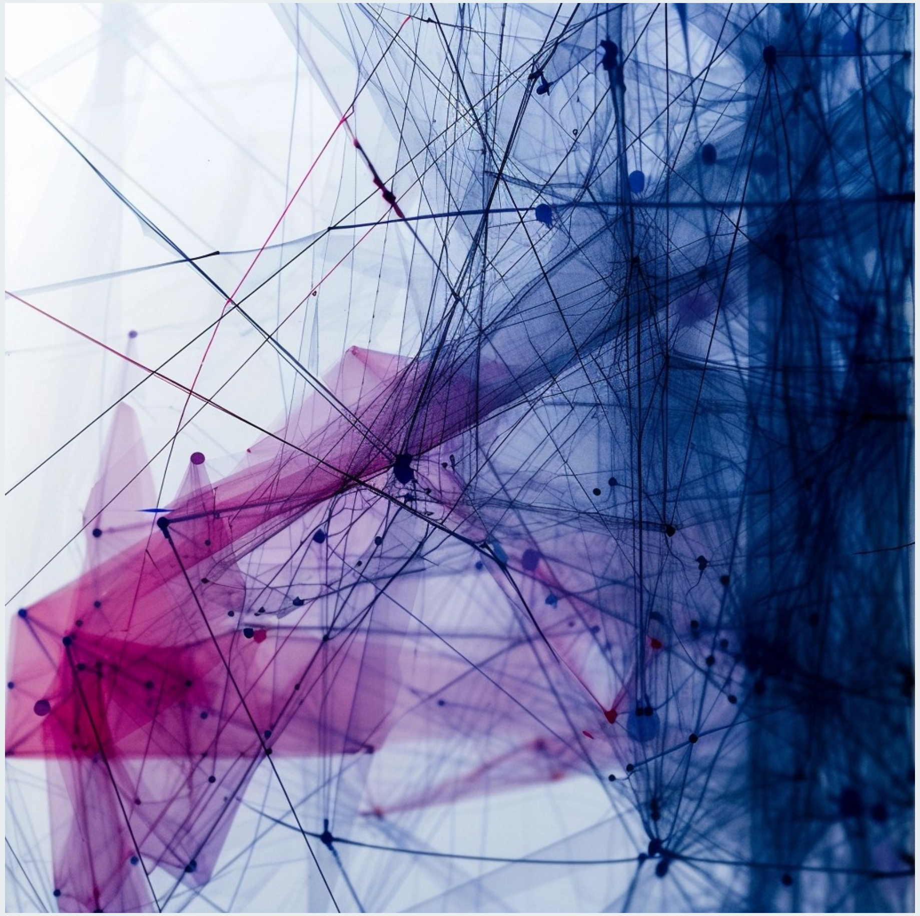


Photo credit: midjourney

Random Graph

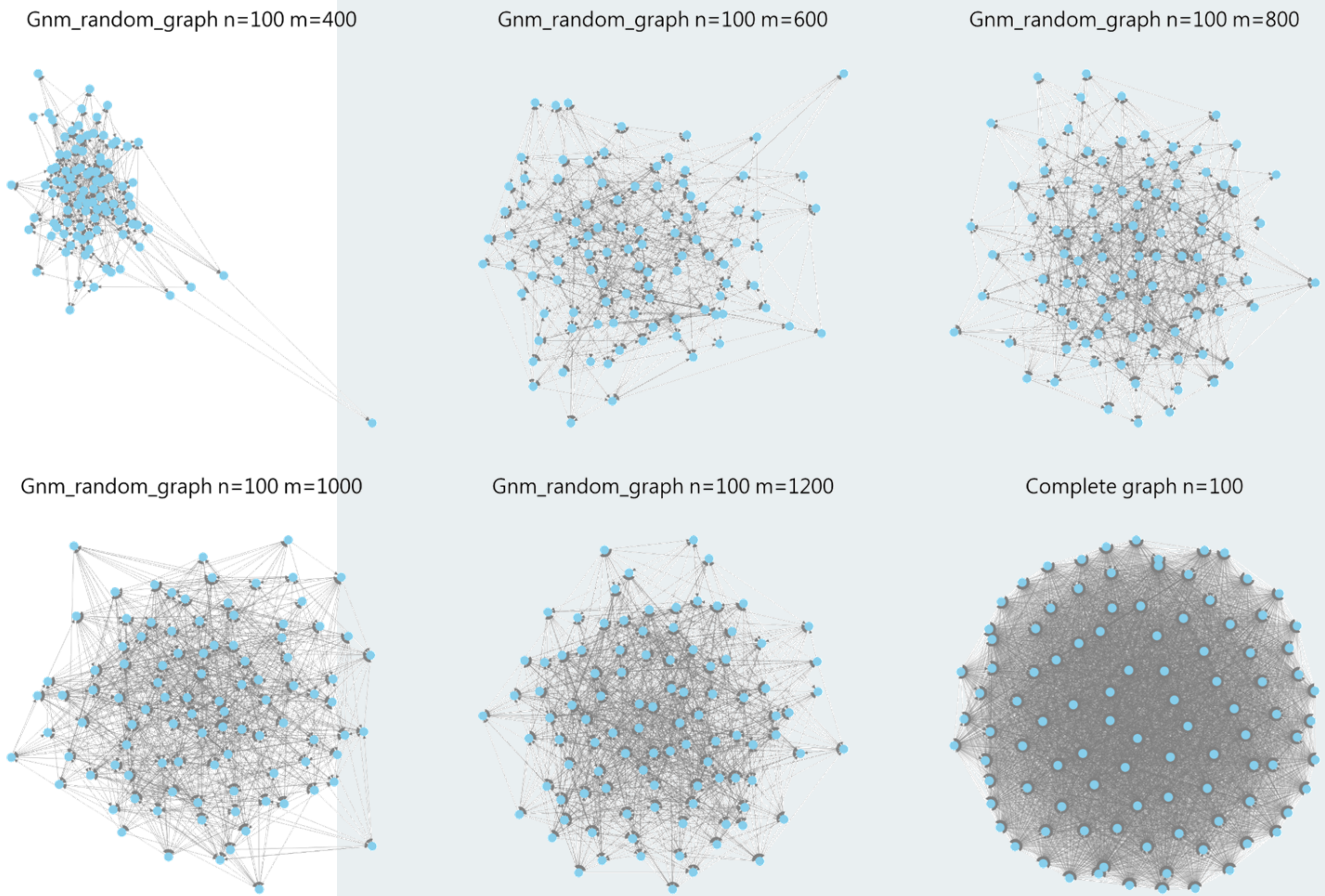
In any social networks, we may compute nodal/ edge/ subgraph/ whole graph level indicators to describe the structure or composition in a network.

While you describe a high-interaction graph, how much confidence do you have? Or, comparing to normal/ common situations, is it a very special property or just a common phenomenon.

Therefore, we need to have random graph to strengthen our arguments and make it as baseline for comparison.

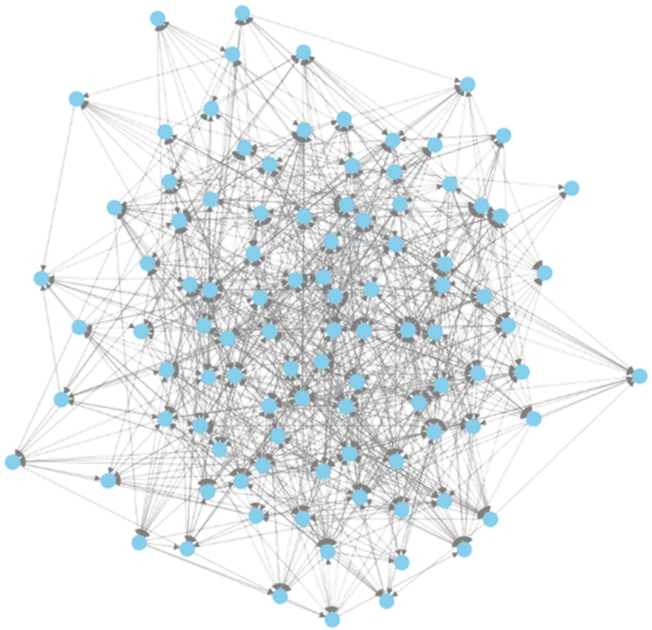
Random Graph

Here is an example:

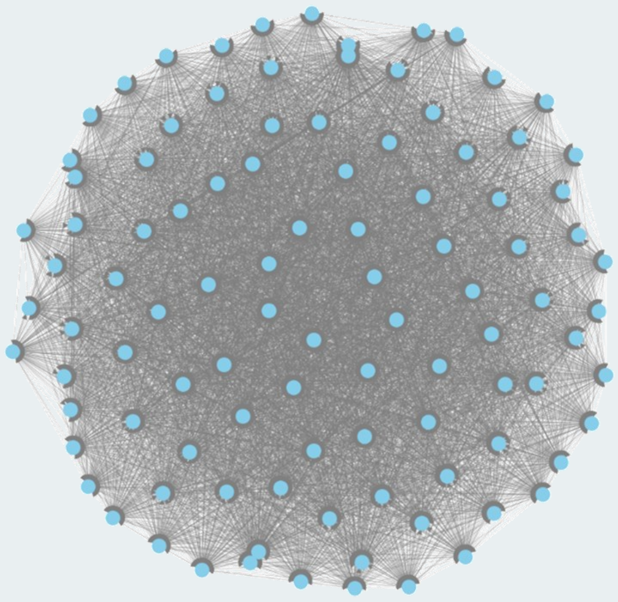


Random Graph

How to know your graph properties is special or just a normal case?



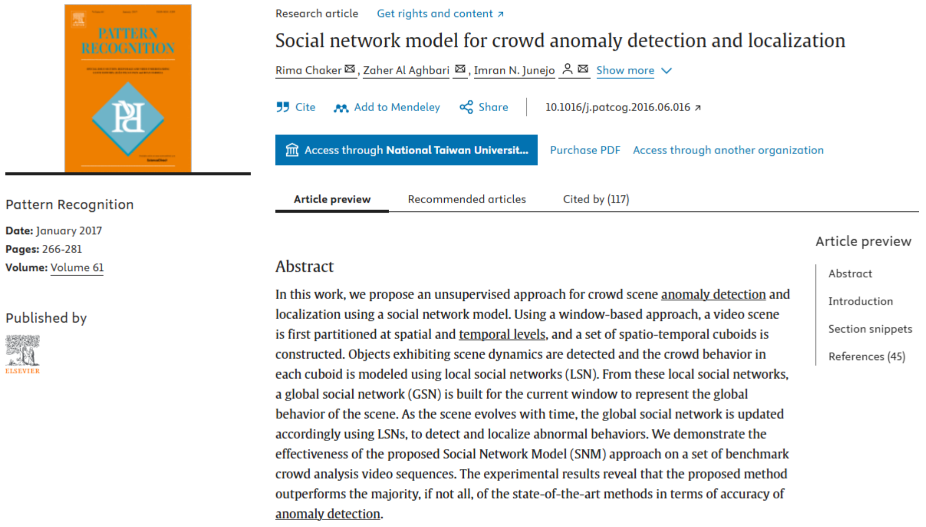
Gnm_random_graph n=100 m=800



Complete graph n=100

Random Graph

Imagine a transportation network, how to know today has a special event or just a normal weekday or weekend?



Research article Get rights and content

Social network model for crowd anomaly detection and localization

Rima Chaker, Zaher Al Aghbari, Imran N. Junejo Show more

Cite Add to Mendeley Share 10.1016/j.patcog.2016.06.016

Access through National Taiwan Universit... Purchase PDF Access through another organization

Article preview Recommended articles Cited by (117)

Pattern Recognition

Date: January 2017

Pages: 266-281

Volume: Volume 61

Published by

Abstract

In this work, we propose an unsupervised approach for crowd scene anomaly detection and localization using a social network model. Using a window-based approach, a video scene is first partitioned at spatial and temporal levels, and a set of spatio-temporal cuboids is constructed. Objects exhibiting scene dynamics are detected and the crowd behavior in each cuboid is modeled using local social networks (LSN). From these local social networks, a global social network (GSN) is built for the current window to represent the global behavior of the scene. As the scene evolves with time, the global social network is updated accordingly using LSNs, to detect and localize abnormal behaviors. We demonstrate the effectiveness of the proposed Social Network Model (SNM) approach on a set of benchmark crowd analysis video sequences. The experimental results reveal that the proposed method outperforms the majority, if not all, of the state-of-the-art methods in terms of accuracy of anomaly detection.

Article preview

Abstract

Introduction

Section snippets

References (45)

Pattern and Anomaly Detection in Urban Temporal Networks

Mingyi He[1]* Shivam Pathak[1]* Urwa Muaz[1]*
Jingtian Zhou[1] Saloni Saini[1] Sergey Malinchik[2] Stanislav Sobolevsky[1]
[1]Center for Urban Science and Progress
New York University
New York, NY, 11201
[2]Lockheed Martin Advanced Technology Lab
Lockheed Martin Corporation
Cherry Hill, NJ, 08002

Abstract

Broad spectrum of urban activities including mobility can be modeled as temporal networks evolving over time. Abrupt changes in urban dynamics caused by events such as disruption of civic operations, mass crowd gatherings, holidays and natural disasters are potentially reflected in these temporal mobility networks. Identification and early detecting of such abnormal developments is of critical importance for transportation planning and security. Anomaly detection from high dimensional network data is a challenging task as edge level measurements often have low values and high variance resulting in high noise-to-signal ratio. In this study, we propose a generic three-phase pipeline approach to tackle curse of dimensionality and noisiness of the original data. Our pipeline consists of i) initial network aggregation leveraging community detection ii) unsupervised dimensionality reduction iii) clustering of the resulting representations for outlier detection. We perform extensive experiments to evaluate the proposed approach on mobility data collected from two major cities, New York City and Taipei. Our results empirically prove that proposed methodology outperforms traditional approaches for anomaly detection. We further argue that the proposed anomaly detection framework is potentially generalizable to various other types of temporal networks e.g. social interactions, information propagation and epidemic spread.

Keywords: Temporal Network, Anomaly Detection, Urban Mobility

1 INTRODUCTION

Anomaly detection can be simply defined as the identification of observations that deviate significantly from the normal patterns[1]. In urban systems, besides mitigating operational disruptions it has applications in the domain of public safety for early detection of threats and suspicious activities. Examples of these scenarios includes 2015 New Year's Eve celebration in Shanghai, where overcrowding resulted in a stampede causing 36 casualties. Similarly, in 2016 Central Park, New York City witnessed a dangerous stampede of 'Pokemon Go' players who gathered in hundreds to pursue a game character. Early detection of these events can enable the authorities to take preventive measures to effectively mitigate the adverse consequences. In various research works [2–5] social media data has been utilized for the detection of civic unrest. Bohannon discusses that social media data can be leveraged to forecast civic riots[3]. Abdelhaq et al [4] proposes a novel framework for localized event detection from twitter data. In [4, 5], scan statistics

*Mingyi He,Shivam Pathak and Urwa Muaz contributed equally to the paper,the list is alphabetically ordered.

Chaker, R., Al Aghbari, Z., & Junejo, I. N. (2017). Social network model for crowd anomaly detection and localization. *Pattern Recognition*, *61*, 266-281.
He, M., Pathak, S., Muaz, U., Zhou, J., Saini, S., Malinchik, S., & Sobolevsky, S. (2019). Pattern and anomaly detection in urban temporal networks. *arXiv preprint arXiv:1912.01960.*

Random Graph

Imagine a cyber network, how to know there is an anomaly connections or requests today?

Chapter 15
Applications of Social Network Analysis to Managing the Investigation of Suspicious Activities in Social Media Platforms

Romil Rawat, Vinod Mahor, Sachin Chirgaiya, and Abhishek Singh Rathore

15.1 Introduction

The social network analysis is the technique of visualizing nodes, edges to analyze the attributes, and features of connected nodes. The sociogram is used to identify and represent the ties acting between the nodes or the users. This detail is used by police and security intelligence agencies for tracking the chained nodes involved in criminal or terrorist activities.

SNA applications have been broadly aligned toward credibility, business customer reach, fake node identification, data mining and filtering, hidden link prediction and information source modeling, user behavior, nature, location-based analysis with attributes understanding, communities identification and function analysis, social events sharing analysis and identification with meta tagging [1], recommendation or suggestion based on previous activities, and identification of weak ties and strong ties.

Predicting enterprise cyber incidents using social network analysis on dark web hacker forums

Soumajyoti Sarkar
Arizona State University
Tempe, Arizona, U.S.A.

Mohammad Almukaynizi
Arizona State University
Tempe, Arizona, U.S.A.

Jana Shakarian
Cyber Reconnaissannce, Inc.
Tempe, Arizona, U.S.A.

Paulo Shakarian
Arizona State University
Tempe, Arizona, U.S.A.

ABSTRACT

With the rise in security breaches over the past few years, there has been an increasing need to mine insights from social media platforms to raise alerts of possible attacks in an attempt to defend conflict during competition. We use information from dark web forums by leveraging the reply network structure of user interactions with the goal of predicting enterprise cyberattacks. We use a suite of social network features on top of supervised learning models and validate them using a binary classification problem that attempts to predict whether there would be an attack on any given day for an organization. We conclude from our experiments, which gathered information from 53 forums on the dark web over a span of 12 months and attempted to predict real-world cyberattacks across 2 security incidents, that analyzing the path structure between groups of users is better than merely studying centralities like Pagerank or relying on user-posting statistics in forums.

INTRODUCTION

With recent data breaches at organizations such as Yahoo, Uber, and Equifax[1] emphasizing the increasing financial and social impacts of cyberattacks, there has been an enormous requirement for technologies that could alert such organizations to possible data breaches. These breaches are a direct or indirect result of cyber, electronic, and information operations to infiltrate systems and infrastructure as well as gain unauthorized access to information, thus setting an example of conflict during competition. On the vulnerability

Detecting cyber threats through social network analysis: short survey

Kirichenko Lyudmyla

Doctor, Professor, Department of Applied Mathematics, Kharkiv National University of Radioelectronics, Ukraine

Radivilova Tamara

Ph.D., Associated Professor, Department of Infocommunication Engineering, Kharkiv National University of Radioelectronics, Ukraine

Carlsson Anders

Lecturer, Department of Computer Science and Engineering, Blekinge Institute of Technology, Sweden

Abstract

This article considers a short survey of basic methods of social networks analysis, which are used for detecting cyber threats. The main types of social network threats are presented. Basic methods of graph theory and data mining, that deals with social networks analysis are described. Typical security tasks of social network analysis, such as community detection in network, detection of leaders in communities, detection experts in networks, clustering text information and others are considered.

Keywords: social network analysis, data mining, threats, social network security.

JEL Classification: C38, C45, C55, C61, C63.

Introduction

The use of the Internet social networks makes it possible to communicate with old friends, make new acquaintances, express their thoughts on a very wide audience, join groups of interest. By coverage of audiences some groups in social networks and popular bloggers can compete with many media.

According to efficiency information transmission social networks are often superior to most of media, they are able to disseminate information around the world in seconds, thereby expediting the progress of operation, but this does not mean that television and radio have lost their popularity.

In modern conditions there is a symbiosis of major television giants with such networks as WikiLeaks, Facebook, Twitter, YouTube, reinforcing the ultimate effect of informational influence.

The rapid development of social networks and ability to collect information from them led to a noticeable increase of interest to social network analysis and the occurrence of its new methods become increasingly popular and they are used in various fields [5, 6]: expert search system, gathering a team of specialists, social recommendations, search engines people and documents, marketing, communications, advertising, and many others. Nowadays social network analysis (SNA) is used to study a variety of economic and organizational phenomena and processes [25, 42, 61].

The Global Risk Report 2017 [87] just published by the World Economic Forum ahead of its annual meeting in Geneve, continues to receive attention from the business world and reaches a level of credibility in a crowded arena of forecasting. Cyber attacks on businesses will increase with a denial of service, data breach, cloud provider compromise and extortion being major concerns for IT departments. The sensitive geopolitical context, the rise of cyberattacks and major data breaches and hacks, as well as the global insurgency of violent extremism and radicalization have led many countries to the adoption of security measures and counterterrorism laws that have increased scrutiny and restrictions on the participation of societal actors [87]. In this case many countries in the world create the National Cyber Security Centres and Cyber Security Strategy Documents [68, 69, 89, 97].

Rawat, R., Mahor, V., Chirgaiya, S., & Rathore, A. S. (2021). Applications of social network analysis to managing the investigation of suspicious activities in social media platforms. In *Advances in Cybersecurity Management* (pp. 315-335). Cham: Springer International Publishing.
Sarkar, S., Almukaynizi, M., Shakarian, J., & Shakarian, P. (2019). Predicting enterprise cyber incidents using social network analysis on dark web hacker forums. *The Cyber Defense Review*, 87-102.
Kirichenko, L., Radivilova, T., & Carlsson, A. (2018). Detecting cyber threats through social network analysis: short survey. *arXiv preprint arXiv:1805.06680*.
Shrivastava, N., Majumder, A., & Rastogi, R. (2008, April). Mining (social) network graphs to detect random link attacks. In *2008 IEEE 24th international conference on data engineering* (pp. 486-495). IEEE.

Random Graph

Building a Null Model

Erdős-Rényi Random Graph

ER is the first and most common use random graph in social network analysis because you only need to give G(n, p) and get your random graph. n, p represents the number of nodes and linkage probability, respectively.

ER suppose all relations are independent and the probability of linkage generation are equal.

Random Graph

Building a Null Model

Erdős-Rényi Random Graph

The Erdős–Rényi model assumes that all possible relations are generated independently and with the same probability of edge formation.

$$P\left(Y_{ij} = 1\right) = p \mid G(n, p)$$

All possible ties are independent

- Whether node i connects to node j does not affect whether any other pair connects.

All ties have the same probability p

- Every dyad has an equal chance of forming a link.

Random Graph

Erdős–Rényi

In the Erdős–Rényi $G(n,p)$ random graph model, each edge is generated independently with an identical probability p.

Therefore, for any given node, its degree can be regarded as the sum of $n-1$ independent Bernoulli trials, and the degree distribution follows:

$$k_i \sim Binomial(n-1,p)$$

Random Graph

Erdős–Rényi

- When the number of nodes n is large, the edge probability p is small, and the average degree

$$\lambda = p(n-1)$$

- the Binomial distribution can be approximated by a Poisson distribution:

$$P(k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Random Graph

Erdős–Rényi

This also represents one of the **major differences** between Erdős–Rényi random graphs and real-world social networks.

- In ER networks, the degree distribution is typically concentrated around the mean degree, resulting in relatively small degree variance.
- In contrast, real social networks often exhibit strong heterogeneity, heavy-tailed degree distributions, and the presence of hub nodes.
- Consequently, the ER model is generally insufficient for capturing important structural properties of real social systems, such as preferential attachment, community structure, and clustering phenomena.

Random Graph

Erdős–Rényi

Edge Probability p	Network Structure	Information Diffusion	Characteristics
Small p	Isolated nodes and small components	Difficult to spread globally	Sparse network
Intermediate p	Components gradually merge	Diffusion expands rapidly	Giant component may emerge
Large p	Most nodes become connected	Efficient global diffusion	Short path length, but weak clustering

Random Graph

Erdős–Rényi

[Practice] The Dynamic of Erdős–Rényi random graph:

Given a undirected graph with $n = 100$ and $p = \{0.1, 0.2, \ldots, 1\}$.

Please draw the trend of degree, clustering coefficient, and degree distribution (degree vs log(frequency)).

Random Graph

Small World Network

Many real-world networks exhibit both high clustering and short average path length, which appear contradictory.

ER random graphs produce short paths but low clustering, whereas regular lattices preserve high clustering but generate long path lengths.

The Watts–Strogatz model demonstrates that a small amount of random rewiring can simultaneously maintain local clustering and create global shortcuts, producing the small-world phenomenon.

Random Graph

Small World Network :: Watts–Strogatz Model

The model begins with a regular ring lattice, where each node is connected to a fixed number of neighboring nodes. Therefore, the initial network exhibits strong local regularity and high clustering.

A proportion of edges are then rewired to distant nodes with rewiring probability β. When β is small, only a few long-range shortcuts are introduced into the network.

These shortcuts preserve most local clustering while dramatically reducing the average path length, thereby producing a small-world network.

Random Graph

Small World Network :: Watts–Strogatz Model

Rewiring Probability β	Network Structure	Clustering	Path Length
$\beta = 0$	Regular lattice	High	Long
$0 < \beta < 1$	Small-world network	Still high	Rapidly decrease
$\beta = 1$	Random graph	Low	short

Random Graph

Small World Network :: Watts–Strogatz Model

[Practice] The Dynamic of Watts–Strogatz random graph:

Given a undirected graph $G(n, k, \beta)$ with $n = 100$, $k = 4$ and $\beta = \{0.0001, 0.001, 0.01, 0.05, 0.1, 0.5, 1\}$.

Please draw the trend of degree, clustering coefficient, path length distribution, and degree distribution (degree vs log(frequency)).

Random Graph

Small World Network :: Watts–Strogatz Model

Small-world networks explain the coexistence of high clustering and short average path length, but they do not fully capture the degree inequality commonly observed in real-world networks.

Many social and information networks exhibit highly unequal degree distributions, where a small number of nodes possess a disproportionately large number of connections, such as influential scholars, major cities, hub airports, popular websites, or high-traffic transportation nodes.

Random Graph

Scale-Free Network :: Barabási–Albert Model

The **Barabási–Albert (BA)** model explains the emergence of hubs through two key mechanisms: network growth and preferential attachment.

The BA model generates networks whose degree distribution approximately follows a power-law:

$$P(k) \sim k - \gamma, \gamma \sim 3$$

Random Graph

Scale-Free Network :: Barabási–Albert Model

The **Barabási–Albert (BA)** model

- Most nodes have few connections
- A small number of nodes become highly connected hubs
- This heterogeneous structure is the characteristic of a scale-free network.

Random Graph

Scale-Free Network :: Barabási–Albert Model

The key mechanisms are:

– **Growth**
  The network continuously adds new nodes.

– **Preferential attachment**
  New nodes are more likely to connect to already well-connected nodes:

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j}$$

Random Graph

Scale-Free Network :: Barabási–Albert Model

The BA model consists of two core mechanisms: first, network growth, in which new nodes are continuously added to the network; second, preferential attachment, where new nodes are more likely to connect to existing nodes with many connections ➔ "**rich-get-richer**" phenomenon

In social science, this mechanism is closely related to cumulative advantage, prestige formation, centralized resource allocation, and the emergence of inequality.

Random Graph

Scale-Free Network :: Barabási–Albert Model

[Practice] The Dynamic of Barabási–Albert random graph:

Given a undirected graph $G(n, m)$ with $n = 100$ and $m = \{1,2,4,8,16\}$.

Please draw the trend of degree, clustering coefficient, path length distribution, and degree distribution (degree vs log(frequency)).

Random Graph

Summary

BA networks are generally robust to random node failures because randomly removed nodes are most likely to be low-degree nodes with limited influence on overall connectivity.

However, the network becomes highly vulnerable when hub nodes are intentionally targeted. Removing a small number of hubs can rapidly fragment the network and disrupt global connectivity.

Consequently, hubs simultaneously provide efficiency and connectivity while also introducing structural centralization and systemic vulnerability.

Random Graph

Summary

In social networks, airline systems, transportation infrastructures, cybersecurity, and epidemic spreading, understanding the role of hubs is therefore essential for designing effective governance, protection, and intervention strategies.

Models	Assumption/ Mechanism	Major Network Property	Limitations
ER	Random connection	Short path Establish null models	Low clustering, homogeneous degree
WS	Rewiring shortcuts	Small-world High clustering & short path length	Cannot explain hubs
BA	Preferential attachment + growth	Scale-free hubs Hubs, degree heterogeneity	Weak community realism

Random Graph

Applications

Given a graph $G(n,m)$ and its average clustering coefficient and density are acc and den, respectively. Please prove these two properties of this graph is significantly different from random graphs.

networkx

gnm_random_graph(n, m, seed=None, directed=False, *, create_using=None)

gnp_random_graph(n, p, seed=None, directed=False, *, create_using=None)

erdos_renyi_graph(n, p, seed=None, directed=False, *, create_using=None)

binomial_graph(n, p, seed=None, directed=False, *, create_using=None

Statistical Network Modeling

Why Exponential Random Graph Model (ERGM)

ER, WS, BA, or other models could generate random graph; however, we want to know whether a social mechanism affects/ raises/ declines the probability of linkage?

For example, same class, same department, same gender, same race, similar geographical distance, common friends, or same level.

Statistical Network Modeling

Why Exponential Random Graph Model (ERGM)

Exponential Random Graph Models (ERGMs) are among the most important statistical models in social network analysis.

Unlike traditional random graph models that assume independent edge formation, ERGMs aim to statistically explain why social ties are formed.

Statistical Network Modeling

Why Exponential Random Graph Model (ERGM)

The core value of ERGMs is that they allow researchers to:

- estimate multiple network formation mechanisms simultaneously,
- integrate social theory with network statistics,
- and move from descriptive network analysis toward statistical inference.
- E.g., a high clustering graph comes from highly social actors, homophily, triadic closure, and hierarchy.

Statistical Network Modeling

Why Exponential Random Graph Model (ERGM)

ERGMs can be used to describe networks based on their structural properties.

They also allow researchers to examine whether individual attributes (node characteristics) are associated with specific network structures.

Furthermore, ERGMs can evaluate whether behaviors remain associated with social ties after controlling for structural network effects.

For example, after accounting for gender similarity and reciprocity, researchers may investigate whether smoking behavior is more likely to occur among friends.

Statistical Network Modeling

[Review] Exponential Family

An exponential family distribution is a class of probability distributions that can be expressed in a unified exponential form using model parameters and sufficient statistics.

$$P(Y = y) = \frac{1}{\kappa(\theta)} exp\left(\theta^T g(y)\right)$$

where $g(y)$ represents statistics extracted from the observed data, and θ represents the corresponding model parameters.

Statistical Network Modeling

[Review] Exponential Family

Many commonly used statistical models belong to the exponential family, including:

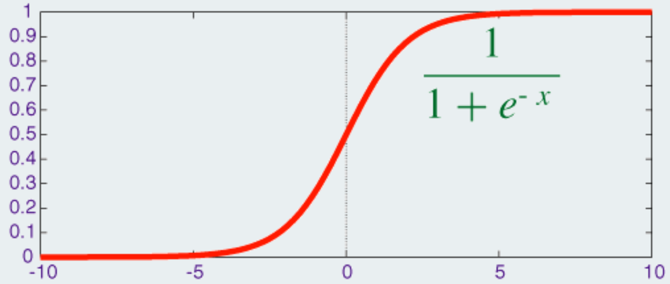
Normal	Exponential	Bernoulli	Geometric
Gamma	Chi-squared	Binomial	Poisson
Beta	Dirichlet	Negative binomial	Zeta

Statistical Network Modeling

[Review] Exponential Family

Random Variables	Probability Density Function
Normal	$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$
Exponential	$f(x, \lambda) = \lambda e^{-\lambda x}$
Gamma	$f(x, \alpha, \theta) = \frac{x^{\alpha-1} e^{-x/\theta}}{\theta^\alpha \Gamma(\alpha)}$
Poisson	$f(k, \lambda) = \Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$

Statistical Network Modeling



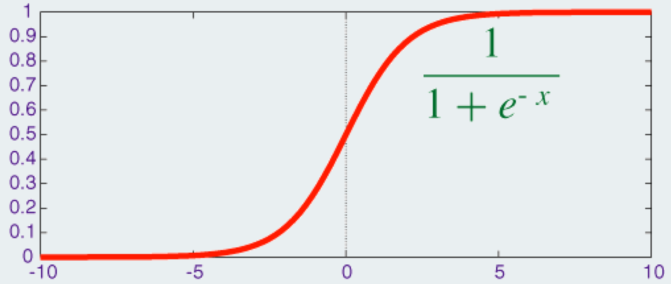
[Review] Logistic Regression

Logistic regression is a supervised machine learning and statistical algorithm used primarily for binary classification tasks to predict the probability of an event occurring.

$$z = f(x) = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$$

$$\sigma(z) = \frac{1}{1+e^{-z}}, \begin{cases} p = \dfrac{e^{f(x)}}{1+e^{f(x)}} \\ 1-p = \dfrac{1}{1+e^{f(x)}} \end{cases}, where\ e^{f(x)} = \frac{p}{1-p} = odds$$

Statistical Network Modeling



[Review] Logistic Regression

Due to the exponential term, we take natural logarithm for both sides, and then we get:

$$e^{f(x)} = \frac{p}{1-p} = odds$$

$$\Rightarrow \ln(odds) = \ln\left(\frac{p}{1-p}\right) = f(x) = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$$

Statistical Network Modeling

[Review] Logistic Regression

Logistic regression models the probability of an event occurring.

In social network analysis, the event of interest becomes whether a tie exists between two nodes.

ERGMs extend the logic of logistic regression by incorporating network structures and social mechanisms into the probability of tie formation.

Statistical Network Modeling

[Review] Logistic Regression

Logistic regression models the probability of an event occurring.

In social network analysis, the event of interest becomes whether a tie exists between two nodes.

ERGMs extend the logic of logistic regression by incorporating network structures and social mechanisms into the probability of tie formation.

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

From logistic regression, we formulate ERGM as follows

$$P_\theta(X = x) \propto \exp\left(\theta^t s(x)\right)$$

$$P_\theta(X = x) = \frac{\exp\left(\theta^t s(x)\right)}{c(\theta)}, c(\theta) = \sum_{\substack{all\ possible \\ graphs\ y}} \exp\left(\theta^t s(y)\right)$$

where X is a random network on n nodes ($\in \{0,1\}$); θ is a vector of parameters; $s(x)$ is a known vector of graph statistics on x.

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Given a graph x and a pair (i, j) of nodes,
$x_{ij} = 0\ or\ 1$, depending on whether an edge exists
x_{ij}^c, denotes the status of all pairs in x other than (i, j)
x_{ij}^+, denotes the same graph as x but with $x_{ij} = 1$
x_{ij}^-, denotes the same graph as x but with $x_{ij} = 0$

Conditional on $X_{ij}^c = x_{ij}^c$, X has only two status based on whether $x_{ij} = 0\ or\ 1$. Here, we calculate the ratio of the two respective probabilities.

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Compute odds and log-odds

$$\frac{P(X_{ij}=1|X_{ij}^c=x_{ij}^c)}{P(X_{ij}=0|X_{ij}^c=x_{ij}^c)}=\frac{\exp\left(\theta^t s(x_{ij}^+)\right)}{\exp\left(\theta^t s(x_{ij}^-)\right)}$$

$$\frac{P(X_{ij}=1|X_{ij}^c=x_{ij}^c)}{P(X_{ij}=0|X_{ij}^c=x_{ij}^c)}=\exp\left(\theta^t s\left(x_{ij}^+\right)-\theta^t s\left(x_{ij}^-\right)\right)=\exp\left(\theta^t\left[s\left(x_{ij}^+\right)-s\left(x_{ij}^-\right)\right]\right)$$

$$\log\left(\frac{P(X_{ij}=1|X_{ij}^c=x_{ij}^c)}{P(X_{ij}=0|X_{ij}^c=x_{ij}^c)}\right)=\theta^t\left[s\left(x_{ij}^+\right)-s\left(x_{ij}^-\right)\right]$$

$\Delta\left(s(x)\right)_{ij}$ denotes the vector of change statistics, $\Delta\left(s(x)\right)_{ij}=s\left(x_{ij}^+\right)-s\left(x_{ij}^-\right)$, so $\Delta\left(s(x)\right)_{ij}$ is the conditional log-odds of edge (i,j).

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Basic statistics model

$$P_{\theta,\mathcal{Y}}(Y = y) = \frac{\exp\left(\theta^T g(y)\right)}{\kappa(\theta, \mathcal{Y})}, y \in \mathcal{Y}$$

$$P(Y = y) = \exp\left\{\sum_{k=1}^{K} \theta_k g_k(y)\right\} / \kappa(\theta)$$

$$logit\left[P\left(y_{ij} = 1\right) \middle| Y^{ij}\right] = \theta_1 \delta_1\left(y^{ij}\right) + \theta_2 \delta_2\left(y^{ij}\right) + \cdots + \theta_n \delta_n\left(y^{ij}\right)$$

δ is the network structural changes, the change in covariate value $g(y)$ when the ij tie changes from 1 to 0. θ is the impact of the covariate on the log-odds of a tie.

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

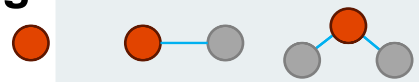
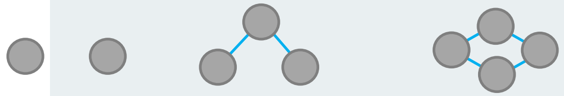
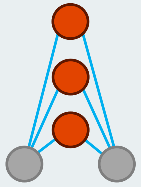
Finally, we get this formula:

$$\log\left(\frac{P(X_{ij}=1|X_{ij}^c=x_{ij}^c)}{P(X_{ij}=0|X_{ij}^c=x_{ij}^c)}\right)=\theta^t\left[s\left(x_{ij}^+\right)-s\left(x_{ij}^-\right)\right]$$

$$\Rightarrow p=\frac{1}{1+\exp(-\theta)}=\frac{\exp(\theta)}{1+\exp(\theta)}$$

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Dyad Independent Terms	**Node attributes**	**Group Heterogeneity** - Average activity - Mixing by group Individual Heterogeneity
	Link attributes	**Heterogeneity** - Duration - Types (e.g., sex, drug)
Dyad Dependent Terms	**Configurations**	**Degree distributions** (or stars)  **Cycle distribution** (2,3,4, etc.)  **Shared partner distributions** 

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

$$P(Y=y)=\exp\left\{\sum_{k=1}^{K}\boldsymbol{\theta_k}\boldsymbol{g_k(y)}\right\}/\kappa(\theta)$$

Covariates: what terms in the model?

Coefficient: **homogeneity constraints** for coefficient on terms?

For example:

Covariate: edges

Coefficient: each edge has the same probability vs
each edge has different probabilities or
some edges have different probabilities

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Building a Null Model

```
# import library
library(ergm)
library(statnet)

# set working directory
setwd('C:\\Users\\ian92\\Desktop\\ERGm')

# load data
load('TCNetworks.Rdata')
```

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Building a Null Model

```
# summary
summary(TCdiss)
```

```
Network attributes:
  vertices = 25
  directed = FALSE
  hyper = FALSE
  loops = FALSE
  multiple = FALSE
  bipartite = FALSE
  title = IN_Diffusion
 total edges = 103
   missing edges = 0
   non-missing edges = 103
 density = 0.3433333

Vertex attributes:

 agency_cat:
   numeric valued attribute
   attribute summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    2.00    2.00    3.24    5.00    6.00

 agency_lvl:
   numeric valued attribute
   attribute summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    1.00    2.00    2.04    3.00    3.00

 lead_agency:
   numeric valued attribute
   attribute summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    0.00    0.00    0.04    0.00    1.00

 tob_yrs:
   numeric valued attribute
   attribute summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    3.00    4.50    6.76    9.00   21.00
  vertex.names:
   character valued attribute
   25 valid vertex names

No edge attributes
```

```
# build null model
DSmod0 <- ergm(TCdiss ~ edges, control=control.ergm(seed=419))
```

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Building a Null Model

```
# model summary
summary(DSmod0)
```

```
Call:
ergm(formula = TCdiss ~ edges, control = control.ergm(seed = 419))

Maximum Likelihood Results:

      Estimate Std. Error MCMC % z value Pr(>|z|)
edges  -0.6485     0.1216      0  -5.333   <1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 415.9  on 300  degrees of freedom
 Residual Deviance: 385.9  on 299  degrees of freedom

AIC: 387.9  BIC: 391.6  (Smaller is better. MC Std. Err. = 0)
```

```
# transform the coefficient
1/(1+exp(-coef(DSmod0)))
```

```
    edges
0.3433333
```

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Adding Node Attributes

```
# node attributes
DSmodl1 <- ergm(TCdiss ~ edges +
                         nodefactor('lead_agency')+
                         nodecov('tob_yrs'),
                         control=control.ergm(seed=419))
# model summary *plz see next slide
summary(DSmodl1)
```

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Adding Node Attributes

```
# model summary
summary(DSmodl1)
```

```
Call:
ergm(formula = TCdiss ~ edges + nodefactor("lead_agency") + nodecov("tob_yrs"),
    control = control.ergm(seed = 419))

Maximum Likelihood Results:

                          Estimate Std. Error MCMC % z value Pr(>|z|)
edges                     -1.67835    0.32931      0  -5.097  < 1e-04 ***
nodefactor.lead_agency.1  17.93665  933.55514      0   0.019  0.98467
nodecov.tob_yrs            0.05994    0.02283      0   2.625  0.00867 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 415.9  on 300  degrees of freedom
 Residual Deviance: 323.5  on 297  degrees of freedom

AIC: 329.5  BIC: 340.6  (Smaller is better. MC Std. Err. = 0)
```

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Adding Dyadic Predictors

ERGM allows us to statistically test whether similarity between actors increases the probability of tie formation.

Large positive parameter estimate: homophily effects

```
# observe the agency homophily
mixingmatrix(TCdiss, 'agency_lvl')
```

```
   1  2  3
1 13 24 14
2 24 16 23
3 14 23 13
```

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Adding Dyadic Predictors

```
# add dyadic homophily
DSmod2a <- ergm(TCdiss ~ edges +
                         nodecov('tob_yrs')+
                         nodematch('agency_lvl'),
                         control=control.ergm(seed=419))
# model summary
summary(DSmod2a)
```

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Adding Dyadic Predictors

```
# add dyadic homophily
DSmod2a <- ergm(TCdiss ~ edges +
                         nodecov('tob_yrs')+
                         nodematch('agency_lvl', diff=TRUE),
                         control=control.ergm(seed=419))
# model summary
summary(DSmod2b)
```

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Adding Dyadic Predictors

DSmod2a

```
ergm(formula = TCdiss ~ edges + nodecov("tob_yrs") + nodematch("agency_lvl"), 
    control = control.ergm(seed = 419))

Maximum Likelihood Results:

                     Estimate Std. Error MCMC % z value Pr(>|z|)    
edges                -2.48077    0.34131      0  -7.268   <1e-04 ***
nodecov.tob_yrs       0.11331    0.02006      0   5.648   <1e-04 ***
nodematch.agency_lvl  0.68751    0.27698      0   2.482   0.0131 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 415.9  on 300  degrees of freedom
 Residual Deviance: 342.2  on 297  degrees of freedom
 
AIC: 348.2  BIC: 359.4  (Smaller is better. MC Std. Err. = 0)
```

DSmod2b

```
ergm(formula = TCdiss ~ edges + nodecov("tob_yrs") + nodematch("agency_lvl", 
    diff = TRUE), control = control.ergm(seed = 419))

Maximum Likelihood Results:

                       Estimate Std. Error MCMC % z value Pr(>|z|)    
edges                  -2.77924    0.36851      0  -7.542  < 1e-04 ***
nodecov.tob_yrs         0.13314    0.02174      0   6.123  < 1e-04 ***
nodematch.agency_lvl.1  1.61450    0.49835      0   3.240  0.00120 ** 
nodematch.agency_lvl.2 -0.21484    0.39735      0  -0.541  0.58873    
nodematch.agency_lvl.3  1.30157    0.44220      0   2.943  0.00325 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 415.9  on 300  degrees of freedom
 Residual Deviance: 329.9  on 295  degrees of freedom
 
AIC: 339.9  BIC: 358.4  (Smaller is better. MC Std. Err. = 0)
```

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Adding Dyadic Predictors

```
# add dyadic homophily
DSmod2c <- ergm(TCdiss ~ edges +
                         nodecov('tob_yrs')+
                         nodemix('agency_lvl', base=1),
                         control=control.ergm(seed=419))
# model summary
summary(DSmod2c)
```

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Adding Dyadic Predictors

DSmod2c

```
ergm(formula = TCdiss ~ edges + nodecov("tob_yrs") + nodemix("agency_lvl",
    base = 1), control = control.ergm(seed = 419))

Maximum Likelihood Results:

                   Estimate Std. Error MCMC % z value Pr(>|z|)
edges              -1.17567    0.53718      0  -2.189  0.02862 *
nodecov.tob_yrs     0.13405    0.02225      0   6.026  < 1e-04 ***
mix.agency_lvl.1.2 -1.58049    0.55005      0  -2.873  0.00406 **
mix.agency_lvl.2.2 -1.83543    0.60280      0  -3.045  0.00233 **
mix.agency_lvl.1.3 -1.53626    0.56587      0  -2.715  0.00663 **
mix.agency_lvl.2.3 -1.70730    0.54454      0  -3.135  0.00172 **
mix.agency_lvl.3.3 -0.31105    0.61186      0  -0.508  0.61119
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 415.9  on 300  degrees of freedom
 Residual Deviance: 329.7  on 293  degrees of freedom

AIC: 343.7  BIC: 369.6  (Smaller is better. MC Std. Err. = 0)
```

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Adding Relational Predictors

```
# add Relational Predictors
DSmod3 <- ergm(TCdiss ~ edges +
                        nodecov('tob_yrs')+
                        nodemix('agency_lvl', diff=TRUE) +
                        edgecov(TCdist, attr='distance') +
                        edgecov(TCcnt, attr='contact'),
                        control=control.ergm(seed=419))
```

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Adding Relational Predictors

```
# model summary
summary(DSmod3)
```

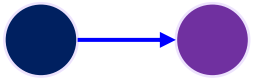
```
ergm(formula = TCdiss ~ edges + nodecov("tob_yrs") + nodematch("agency_lvl",
    diff = TRUE) + edgecov(TCdist, attr = "distance") + edgecov(TCcnt,
    attr = "contact"), control = control.ergm(seed = 419))

Maximum Likelihood Results:

                        Estimate Std. Error MCMC % z value Pr(>|z|)
edges                 -4.8506191  0.6296655      0  -7.703  < 1e-04 ***
nodecov.tob_yrs        0.1287501  0.0282702      0   4.554  < 1e-04 ***
nodematch.agency_lvl.1  1.7951025  0.6256978      0   2.869  0.00412 **
nodematch.agency_lvl.2 -0.6460148  0.5081639      0  -1.271  0.20363
nodematch.agency_lvl.3  1.7218505  0.5468704      0   3.149  0.00164 **
edgecov.distance      -0.0001840  0.0002533      0  -0.726  0.46776
edgecov.contact        1.1242535  0.1469568      0   7.650  < 1e-04 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 415.9  on 300  degrees of freedom
 Residual Deviance: 237.0  on 293  degrees of freedom

AIC: 251  BIC: 276.9  (Smaller is better. MC Std. Err. = 0)
```

Statistical Network Modeling

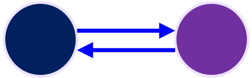
Exponential Random Graph Model (ERGM)

Adding Local Structural Predictors (directed)

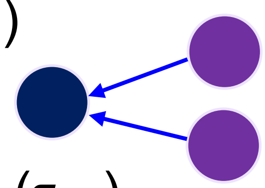
Density (τ_{15})



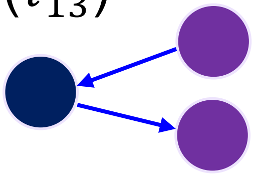
Reciprocity (τ_{11})



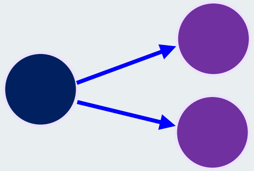
Two-in-stars (τ_{14})



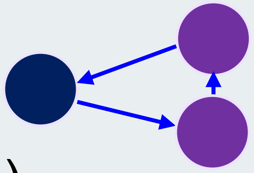
Two-mixed-stars (τ_{13})



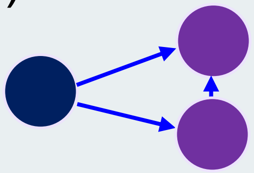
Two-out-stars (τ_{12})



Cyclic triads (τ_{10})



Transitive triads (τ_9)

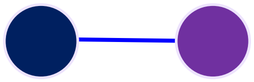


Statistical Network Modeling

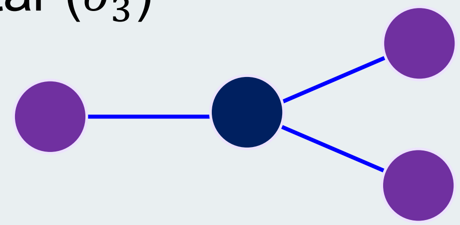
Exponential Random Graph Model (ERGM)

Adding Local Structural Predictors (undirected)

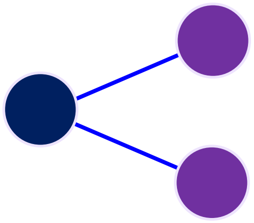
Density or edge (θ)



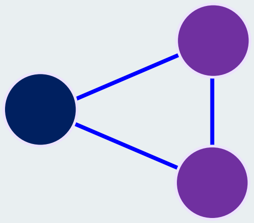
Three-star (σ_3)



Two stars (σ_2)



Triangle (τ)



Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Adding Local Structural Predictors (directed)

```
# add Relational Predictors
DSmod4 <- ergm(TCdiss ~ edges +
                        nodecov('tob_yrs')+
                        nodemix('agency_lvl', diff=TRUE) +
                        edgecov(TCdist, attr='distance') +
                        edgecov(TCcnt, attr='contact') +
                        gwesp(0.7, fixed=TRUE),
                        control=control.ergm(seed=419))
```

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Adding Local Structural Predictors (directed)

```
# model summary
summary(DSmod4)
```

```
ergm(formula = TCdiss ~ edges + nodecov("tob_yrs") + nodematch("agency_lvl",
    diff = TRUE) + edgecov(TCdist, attr = "distance") + edgecov(TCcnt,
    attr = "contact") + gwesp(0.7, fixed = TRUE), control = control.ergm(seed = 419))

Monte Carlo Maximum Likelihood Results:

                       Estimate Std. Error MCMC % z value Pr(>|z|)    
edges                -6.3282237  0.9897166      0  -6.394  < 1e-04 ***
nodecov.tob_yrs       0.0998301  0.0297706      0   3.353 0.000799 ***
nodematch.agency_lvl.1  1.5868680  0.6023058      0   2.635 0.008422 ** 
nodematch.agency_lvl.2 -0.2841774  0.4780155      0  -0.594 0.552182    
nodematch.agency_lvl.3  1.4923842  0.5305872      0   2.813 0.004913 ** 
edgecov.distance     -0.0001504  0.0002450      0  -0.614 0.539349    
edgecov.contact       1.0486859  0.1486384      0   7.055  < 1e-04 ***
gwesp.fixed.0.7       0.8542586  0.3799970      0   2.248 0.024572 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 415.9  on 300  degrees of freedom
 Residual Deviance: 230.8  on 292  degrees of freedom
 
AIC: 246.8  BIC: 276.4  (Smaller is better. MC Std. Err. = 0.1884)
```

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Name	Unit of counting	Description	Num. of statistics
edges	edges	number of edges	1
nodefactor	units of the attribute	number of times that nodes with a given level of a categorical nodal attribute appear within the edgeset	1 per level of attribute included
nodematch	edges	number of edges whose incident nodes match on value of nodal attribute	if diff=F, 1; if diff=T, 1 per level of attribute included
nodemix	edges	number of edges whose incident nodes fall into each combination of values for a nodal attribute	1 per combo of attribute included
nodecov	units of the attribute	the sum of a nodal attribute value for the incident nodes of an edge, summed across all edges	1
absdiff	units of the attribute	the absolute difference in a nodal attribute between incident nodes for an edge, summed across all edges	1
mutual	dyad	number of mutual edges (a->b and b->a)	1
degree	node	number of nodes of degree x	1 per each value of x included
idegree	node	number of nodes of in-degree x	1 per each value of x included
odegree	node	number of nodes of out-degree x	1 per each value of x included
b1degree	node	number of mode 1 nodes of degree x	1 per each value of x included
b2degree	node	number of mode 2 nodes of degree x	1 per each value of x included
mindegree	node	number of nodes of at least degree x	1 per each value of x included
triangles	triangles	number of triangles	1 (more if attributes used)
gwesp	complex	geometrically weighted edgewise shared partners	1

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Model Performance

```
# GOF
DSmod.fit <- gof(DSmod4, GOF = ~ distance + espartners +
                   degree +  triadcensus,
                   control = control.gof.ergm(MCMC.burnin = 1e5,
                   MCMC.interval = 1e5))
```

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Model Performance

GOF

DSmod.fit

Goodness-of-fit for triad census

	obs	min	mean	max	MC p-value
0	832	584	754.72	931	0.32
1	759	786	876.92	968	0.00
2	517	373	503.11	606	0.86
3	192	124	165.25	265	0.28

Goodness-of-fit for degree

	obs	min	mean	max	MC p-value
0	0	0	0.82	3	0.70
1	1	0	0.75	3	1.00
2	2	0	0.96	3	0.54
3	3	0	1.12	4	0.20

Goodness-of-fit for edgewise shared partner

	obs	min	mean	max	MC p-value
0	1	0	0.95	4	1.00
1	8	0	4.35	14	0.20
2	10	2	10.12	23	1.00
3	7	4	16.14	26	0.04
4	15	9	19.93	33	0.38
5	11	5	17.77	27	0.06
6	9	3	13.18	29	0.38

Goodness-of-fit for model statistics

	obs	min	mean	max	MC p-value
edges	103.0000	88.0000	103.4300	124.0000	1.0
nodecov.tob_yrs	1743.5000	1477.0000	1750.0000	2093.5000	1.0
nodematch.agency_lvl.1	13.0000	9.0000	12.9900	18.0000	1.0
nodematch.agency_lvl.2	16.0000	9.0000	15.7900	22.0000	1.0
nodematch.agency_lvl.3	13.0000	7.0000	13.2600	19.0000	1.0
edgecov.distance	42304.3720	30286.9110	42493.3537	54816.1530	1.0
edgecov.contact	281.0000	242.0000	281.4100	322.0000	1.0
gwesp.fixed.0.7	187.0609	156.2016	188.2785	235.4067	0.9

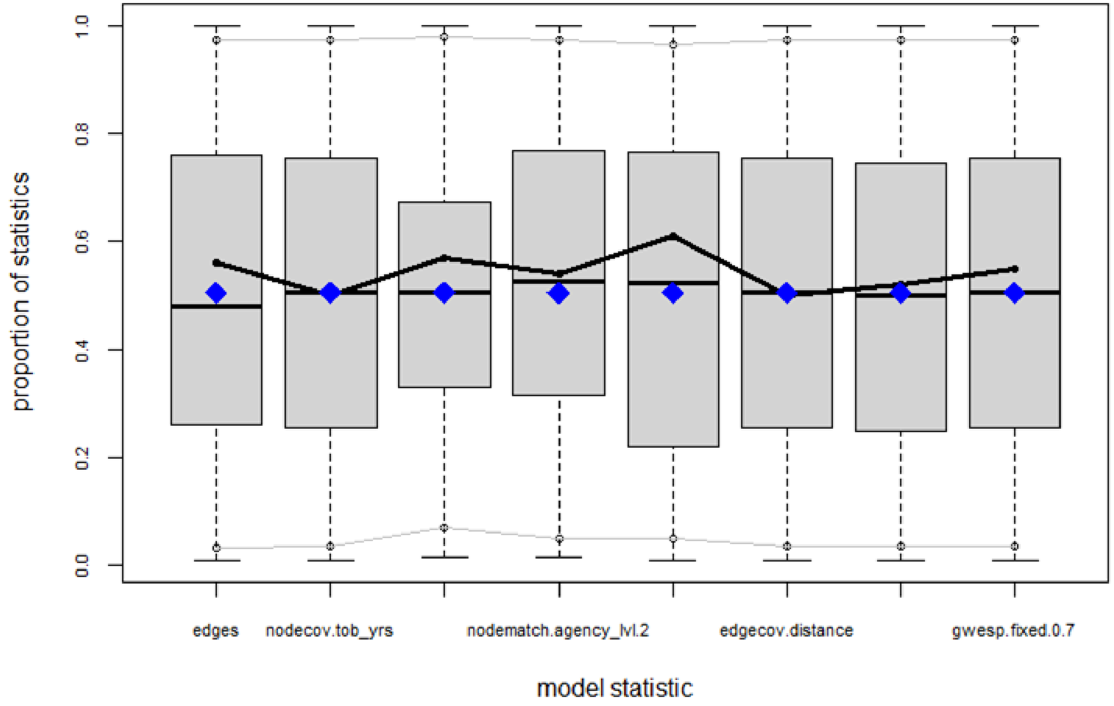
Goodness-of-fit for minimum geodesic distance

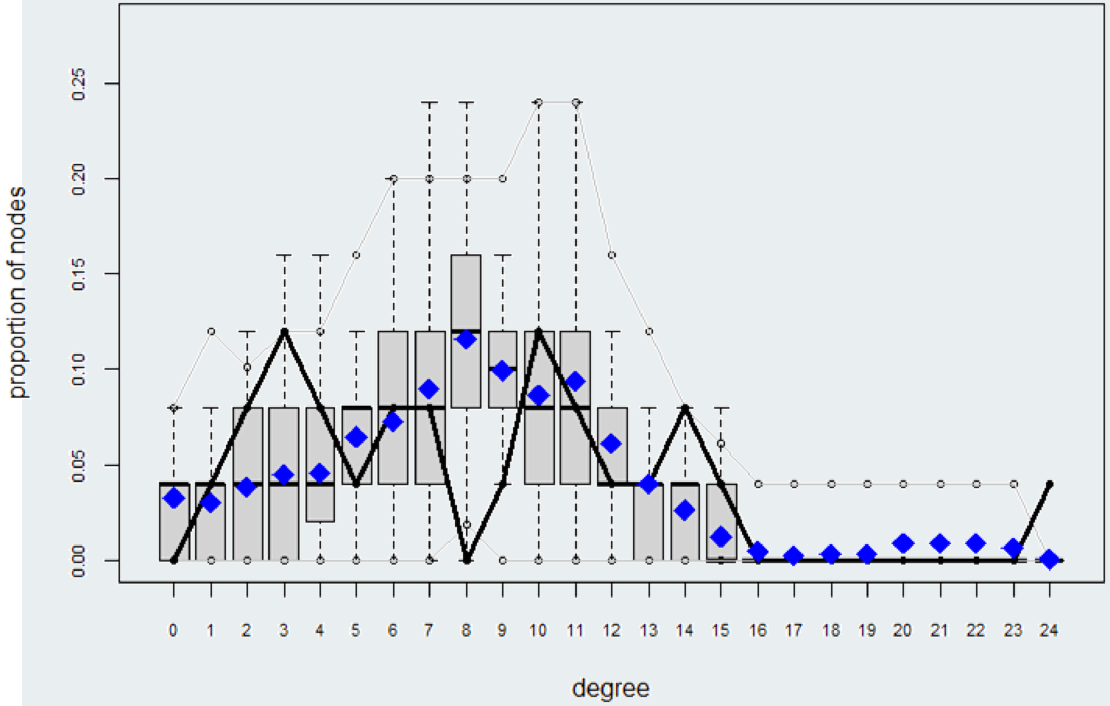
	obs	min	mean	max	MC p-value
1	103	88	103.43	124	1.00
2	197	132	167.06	205	0.06
3	0	0	9.90	35	0.28
4	0	0	0.11	3	1.00
Inf	0	0	19.50	69	0.70

Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Model Performance



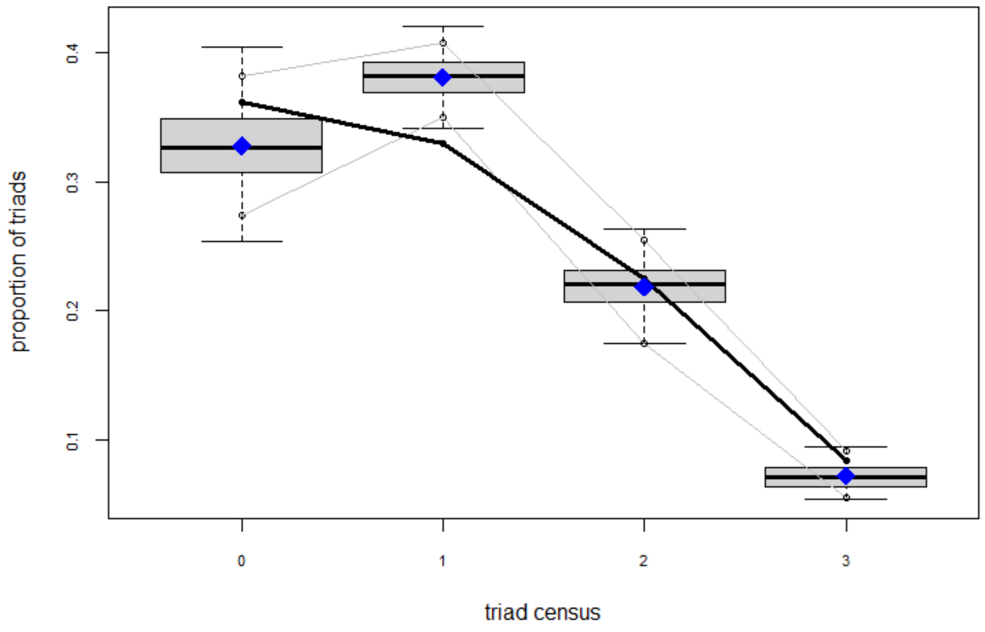


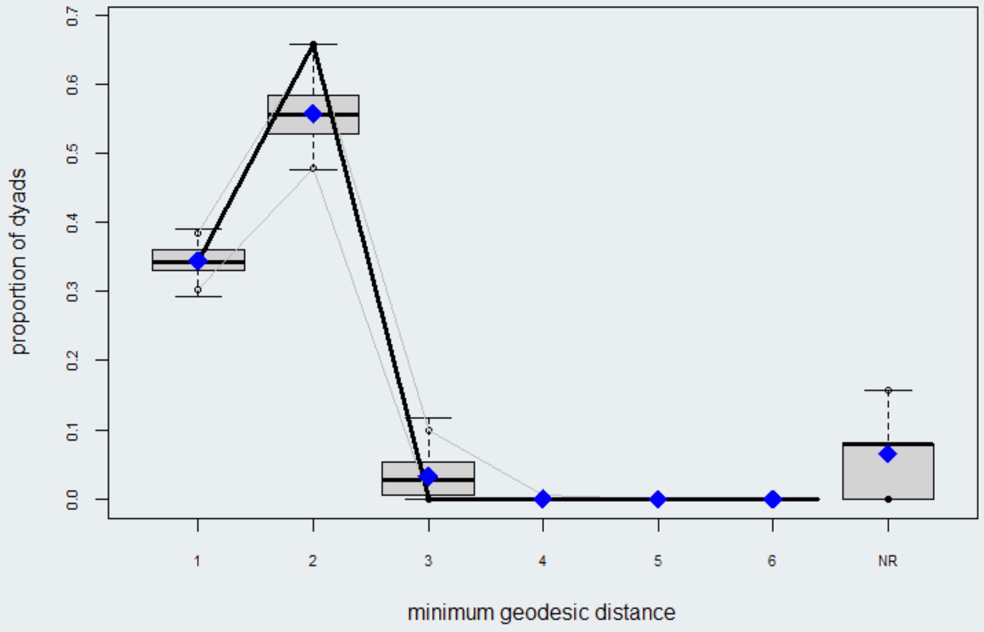
Statistical Network Modeling

Exponential Random Graph Model (ERGM)

Model Performance

Goodness-of-fit diagnostics





Paper Reading

Journal of Complex Networks (2020) **00**, 1–14
doi: 10.1093/comnet/cnaa041

The impact of human mobility networks on the global spread of COVID-19

Marian-Gabriel Hâncean[†]
Department of Sociology, University of Bucharest, Bucharest, Romania
[†]Corresponding author. Email: gabriel.hancean@unibuc.ro

Mitja Slavinec
Faculty of Natural Sciences and Mathematics, University of Maribor, Maribor, Slovenia

and

Matjaž Perc
Department of Medical Research, China Medical University Hospital, China Medical University, Taichung, Taiwan & Complexity Science Hub Vienna, Vienna, Austria

Edited by: Ernesto Estrada

[Received on 7 October 2020; editorial decision on 13 October 2020; accepted on 14 October 2020]

Human mobility networks are crucial for a better understanding and controlling the spread of epidemics. Here, we study the impact of human mobility networks on the COVID-19 onset in 203 different countries. We use exponential random graph models to perform an analysis of the country-to-country global spread of COVID-19. We find that most countries had similar levels of virus spreading, with only a few acting as the main global transmitters. Our evidence suggests that migration and tourism inflows increase the probability of COVID-19 case importations while controlling for contiguity, continent co-location and sharing a language. Moreover, we find that air flights were the dominant mode of transportation while male and returning travellers were the main carriers. In conclusion, a mix of mobility and geography factors predicts the COVID-19 global transmission from one country to another. These findings have implications for non-pharmaceutical public health interventions and the management of transborder human circulation.

Keywords: COVID-19 case importations; human mobility networks; incoming migration; inbound tourism; complex networks.

Hâncean, M. G., Slavinec, M., & Perc, M. (2020). The impact of human mobility networks on the global spread of COVID-19. Journal of Complex Networks, 8(6), cnaa041.

Questions:

1. What is the objective of this paper?
2. How did the authors formulate their graph?
3. In QAP, what is the primary outcome?
4. In ERGM, what is the primary outcome?
5. If you want to leverage ERGM into your area, please give at least one example and explain how does ERGM apply in your work?

References

- Erdős, P., & Rényi, A. (1959). On Random Graphs I. Publicationes Mathematicae Debrecen, 6, 290-297.
- Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of small-world networks. Nature, 393, 440-442.
- Barabási, A.-L., & Albert, R. (1999). Emergence of scaling in random networks. Science, 286, 509-512.
- Robins, G., Pattison, P., Kalish, Y., & Lusher, D. (2007). An introduction to exponential random graph models for social networks. Social Networks, 29, 173-191.
- Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. Journal of Anthropological Research, 33, 452-473.

Photo credit: midjourney

The End

Thank you for your attention!

Email: chchan@ntnu.edu.tw
Website: toodou.github.io

Photo credit: midjourney