A Tutorial Paper on Causal Inference in Networks

Tyler Maule

April 19th, 2021

Introduction

In causal inference and across the discipline of experimental design, a common simplifying assumption suggests that experimental units are independent from one another (Morgan & Winship, 2007). As the world grows more interconnected and researchers probe more complex questions, however, this assumption of complete independence and noninterference becomes less plausible. Especially in social science and business domains, known and unknown relationships connect experimental units in a manner that may amplify or suppress the magnitude of treatment effects. Researchers can more effectively undertake causal inference by representing these connected units in a network structure. Whether in experimental or observational settings, drawing on a network's structure allows for the understanding and quantification of interference. Reworked traditional assumptions paired with the use of common techniques including propensity score estimation, Monte Carlo simulations, and subclassification provide modern methodologies for approaching novel problems (Kolaczyk & Csárdi, 2020; Forastiere et. al., 2020).

This tutorial paper begins with a brief introduction to network science and a review of relevant concepts in causal inference. With this background established, a quick primer covers the complications that network structures present. The following section summarizes a widely accepted framework for addressing these complications when

estimating average treatment effects. Commentary on a recent technique for analyzing observational studies on networks then exemplifies active research in the domain. Returning to the experimental case, an analysis of a study on anti-bullying interventions teaches readers how to implement new techniques in R. References to additional resources and areas of active research round out the paper.

Network Science

At a foundational level, networks represent interrelated units, whether those units are on the scale of cells, individual people, or nations. The nature of their relationships stems from both the units and their underlying context. For example, a network could connect museums that have displayed the same painting at some point. It could connect wildebeests that had fought with one another. Even intergalactic empires that have engaged in trade could be structured as a network that draws on science fiction novels.

These networks, or "graphs" in the mathematical lexicon, consist of links (or "edges") between nodes (or "vertices") that signify a direct relationship between pairs of nodes (Barabási & Samii, 2016). In the context of an NGO, edges between individual employees (nodes) could communicate a manager-worker relationship in the NGO's organizational structure, that the two employees exchanged emails, that the employees eat lunch together, or even that the two employees often book the same conference room. Consider that some of those relationships are symmetric in nature, or "undirected"—if Rita often eats lunch with Bhavin, one can also state that Bhavin often eats lunch with Rita.

Directed networks consist of nonsymmetric relationships—if Rita is Bhavin's manager, that does not imply that Bhavin is Rita's manager.

A network **G** can then be represented as a system of *N* nodes and *E* edges, represented in ordered or unordered pairs (such that given distinct nodes u and v, $\{u,v\}=\{v,u\}$ for undirected networks and $\{u,v\}\neq\{v,u\}$ for directed networks).

Causal inference on networks typically deals with undirected "simple networks," where edges are not assigned weight values, and "proper edges" which contain no loops and a maximum of one edge per pair of nodes. In this context, nodes are "neighbors" if they are connected by an edge. If all nodes in a network are indirectly connected to one another through a series of edges, we refer to the network as "complete." Each node has a "degree," capturing the number of edges that connect it and other nodes. The "distance" between two nodes, then, is the minimum number of edges between them. One common representation of networks is the adjacency matrix, an *N* by *N* matrix wherein each entry denotes cells' pairwise relationship. If nodes *u* and *v* are linked by an edge, the adjacency matrix **A** will contain $A_{uv} = 1$; otherwise, A_{uv} will be 0 (Kolaczyk & Csárdi, 2020).

Although causal inference also works with more complex networks and specific types of networks (including DAGs), those systems and their properties are not dealt with in this tutorial paper.

Causal Inference

While the reader is expected to be familiar with core concepts in causal inference, this section outlines the notation used in this tutorial paper and provides a brief review. Like most modern statistical approaches to causal inference, methods for working with networks build from the potential outcomes framework. First explored by Neymann in 1923, the framework uses observed responses to treatment and counterfactuals for the discernment of treatments' effects (Imbens & Rubin, 2015). Two effects of interest are the average treatment effect (ATE), which captures the average effect across all experimental units, and the average effect of treatment on the treated (ATT) which specifically captures the average effect on units in the given treatment group.

To ensure these effects are indeed caused by treatments, researchers may randomize like units into different treatment groups prior to an experiment. Checking the balance of covariates across treatment groups can effectively validate, or provide evidence against, the success of randomization. Different treatment assignment mechanisms and experimental designs are often used to improve covariate balance, if not to ease the experiment's execution. Beyond requiring randomization, traditional causal inference makes the Stable Unit Treatment Value Assumption (SUTVA), formalized by Rubin in 1986 (Imbens & Rubin, 2015). The assumption dictates that units do not interfere with each other's outcomes and that there are no hidden variations on treatments.

Throughout this paper, let $Y_i(W_i)$ denote the potential outcome that would occur if unit i is assigned treatment W_i. By extension, $Y(W_j)$ is the vector of potential outcomes if all units received W_j, and W represents a treatment assignment vector (such that the first element W_1 is the treatment assigned to the first unit, W_2 is assigned to the second, and so on). Under the SUTVA assumption, $Y_i(W_i) = Y_i(W)$; the potential outcome for a given unit with a given treatment assignment does not depend on other units' treatment assignments. Then if W_1 is an assignment to the control group and W_2 an assignment to the treatment group, the treatment effect on unit Y_i will be $Y_i(W_2) - Y_i(W_1)$. In the aggregate, ATE is defined as $\hat{\tau}_{ATE} = \bar{Y}(W_2) - \bar{Y}(W_1)$. Finally, note that the covariates of unit i, X_i, are stored as columns in the covariate matrix **X**.

The Problem of Causal Inference on Networks

When designing and analyzing an experiment on units in a network structure, the core SUTVA assumption of unit noninterference is violated (Kolaczyk & Csárdi, 2020). In some contexts, it is appropriate to treat entire networks of nodes as units, whether they be classrooms, families, or fire departments. Cluster randomized trials and sequential two-stage randomization can also provide unbiased estimates if interference only occurs within clusters.

However, due to limited time or resources, researchers may choose to focus on a single network. If the units and causal relationships of interest are specific to lower-level or smaller units, experiments that treat networks as units may not address researchers' goals.

An experiment using fire stations as units, for instance, may provide weaker evidence regarding how firefighters' age impacts the efficacy of a treatment relative to an experiment treating firefighters themselves as units. Researchers may be particularly interested in how the network structure affects the impact of treatments imposed, or they may seek to investigate which treatments have the greatest effect on untreated units. Clearly, there exist a range of reasons why an experiment might need to be conducted on a network directly.

One then acknowledges that the treatment assigned to one unit may have an impact on the potential and observed outcomes of other units. The effects of this interference are commonly known as "spillover effects," or "peer influence effects" in the social sciences (Forastiere et. al., 2020). Whether because of spreading treatment, units' outcomes affecting one another, intermediate variables, or other factors, $Y_i(W_i) \neq Y_i(W)$ rather than $Y_i(W_i) = Y_i(W)$. Not only does interference occur, but treatment receipt depends on the assignment mechanism—both elements of SUTVA are violated.

Without replacing SUTVA with new assumptions, it becomes difficult to conduct causal inference. Given the presence of interference, one must work with $Y_i(W)$ rather than $Y_i(W_i)$—the network's structure influences how units receive a treatment assignment W, so "exposure" to treatment plays into potential outcomes. Therefore, the common expression of ATE as $\hat{\tau}_{ATE} = \bar{Y}(W_2) - \bar{Y}(W_1)$ and its estimated variance $\widehat{Var}[\hat{\tau}_{ATE}]$ produce biased estimates. Indeed, instead of comparing potential outcomes given the t treatments a unit could be assigned, researchers need to account for t^N possible exposures to treatment (Aronow and Sammii, 2017). Inference and generalization become cumbersome, if not

impossible. An alternate methodology is clearly needed to approach the design and analysis of experiments on networks.

New Methodologies for Causal Inference on Networks

Work by Aronow and Sammii (2017) addressed the exponential number of possible exposures by introducing the concept of "exposure mapping" functions. Define a function $f = f(W,\mathrm{X}) = C$ that maps a given treatment assignment vector and covariate matrix to an exposure vector C whose values draw on a limited set of exposure conditions $c_1, c_2, \ldots, c_K$. Covariate matrix $\mathbf{X}$ typically includes the adjacency matrix $\mathbf{X} = \mathbf{X}(A)$ among its columns. Now each unit can be considered subject to one of K exposure conditions, rather than t^N exposure conditions. Naturally, the choice of exposure mapping is highly context-dependent, relying on subject matter expertise regarding the units, network structure, treatments, and nature of interference. Still, with good judgment this simplifying restriction can be implemented to produce unbiased estimates of treatment effects.

Aronow and Sammii introduce a four-level exposure mapping function to illustrate the method. With treatment assignment W and letting X_i correspond to the vector of the adjacency matrix for unit i, define:

$$f(W, A_i) = \begin{cases} c_{11} & W_i I_{[W^T A_i > 0]} = 1 \\ c_{10} & W_i I_{[W^T A_i = 0]} = 1 \\ c_{01} & (1 - W_i) I_{[W^T A_i > 0]} = 1 \\ c_{00} & (1 - W_i) I_{[W^T A_i = 0]} = 1 \end{cases}$$

where exposure condition c_{11} denotes direct *and* indirect exposure to treatment, c_{10} denotes direct exposure only, c_{01} denotes indirect exposure only, and c_{00} denotes no exposure to treatment.

In this case, exposure conditions depend only on whether *at least* one neighboring node is assigned to treatment. So if distinct but identically shaped adjacency matrices $A_0 \neq A_1$ are identical in column i, $A_{0i} = A_{1i}$, $Y_i(f(W_i,\text{X\_i}(A_{0i}))) = Y_i(f(W_i,\text{X\_i}(A_{1i})))$ is implied.

Generally, one common assumption that can be folded into the exposure mapping states that a node's exposure condition is only impacted by the treatment assignment of its immediate neighbors (its neighborhood). Forastiere (2018) also invokes Rubin's assumption of consistency: identical treatment assignments with different underlying treatment assignment mechanisms are still considered identical. Together with the above Neighborhood Interference Assumption and the guarantee of fixed, known connections within the network, she defines this as the "Stable Unit Treatment on Neighborhood Value Assumption" (SUTNVA). A unit's exposure condition may depend on its "unit covariates" X^{ind} unrelated to the network, as well as "neighborhood covariates" X^{neigh} pertaining to neighborhood structure, position of the node amongst neighbors, and even aggregations of neighborhood unit covariates.

Of course, much more complex exposure mappings may be utilized with a relaxation of SUTNVA. Beyond the neighborhood, exposure conditions could depend on the network as a whole. For the purposes of this paper and the following methods, however, assume that SUTNVA holds. In addition, the methods below primarily deal with two treatments (or

treatment and control), although they are easily extensible to approach experiments with more treatments.

Based on these assumptions we can redefine the ATE as $\hat{\tau}_{ATE} = (1/N)\sum_{i=1}^{N}[\,Y_i(1) - Y_i(0)]$ where 1 is a vector of *N* ones referring to full treatment and 0 a vector of *N* zeroes denoting full control. This expression of ATE allows for the presence of interference, combining the effects of direct treatment on units and the spillover effects from neighbors. To distinguish between the direct and spillover treatment effects, evaluate $\tau(c_{10}, c_{00}) = (1/N)\sum_{i=1}^{N}[\,Y_i(c_{10}) - Y_i(c_{00})]$ to find the direct effect and $\tau(c_{01}, c_{00}) = (1/N)\sum_{i=1}^{N}[\,Y_i(c_{01}) - Y_i(c_{00})]$ to find the spillover effect. In a sense, the classical definition of average treatment effect can be generalized to produce a set of contrasts that determine marginal ATE for a given treatment or set of treatments.

Evaluating $Y_i(c_{lm})$ for a given *l, m, i* requires knowledge or estimation of treatment assignment probabilities.

$$P_i(c_k) = \sum_{W} P_W \, I_{[f(W,X_i)=c_k]}$$

In the case of controlled experiments, the assignment mechanism is known and can be leveraged to identify the probability of each node's assignment to treatment (Imbens & Rubin, 2015). But even with exposure mapping, the multitude of possible exposure conditions complicates finding the probability that a given node is subject to a given condition. In the simple four-level mapping example from Aronow and Sammii, the notion that nodes can receive indirect treatment if at least one of their neighbors receives direct treatment necessitates the evaluation of nodes' pairwise joint probabilities $P_{ij}(c_k, c_l)$ for

nodes *i*, *j* and conditions *k*, *l*. Increasingly complex formulations, such as exposure conditions being related to the total number of directly treated nodes in their neighborhood, require increasingly complex joint probability calculations.

Kolaczyk and Csárdi (2020) outline a common approach to treatment estimation given an exposure mapping: based on the known assignment mechanism, define P as a diagonal matrix with its diagonal carrying the probabilities of selection for the U potential treatment assignments vectors, $P_{W_1}, P_{W_2}, \dots, P_{W_U}$. Then with function $I_{f(W_v,X_v)=c_k}$ which indicates whether node *v* will receive exposure condition c_k given its covariate vector X_v and the treatment assignment W_v, note that:

$$\mathrm{I}_k \mathrm{PI}_k^T = \begin{bmatrix} P_1(c_k) & P_{12}(c_k) & \dots & P_{1N}(c_k) \\ P_{21}(c_k) & P_2(c_k) & \dots & P_{2N}(c_k) \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1}(c_k) & P_{N2}(c_k) & \dots & P_N(c_k) \end{bmatrix}$$

$$\mathrm{I}_k \mathrm{PI}_l^T = \begin{bmatrix} 0 & P_{12}(c_k, c_l) & \dots & P_{1N}(c_k, c_l) \\ P_{21}(c_k, c_l) & 0 & \dots & P_{2N}(c_k, c_l) \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1}(c_k, c_l) & P_{N2}(c_k, c_l) & \dots & 0 \end{bmatrix}$$

The first matrix provides the individual and joint probabilities of nodes' exposures to condition c_k, while the second matrix contains joint probabilities that each pair of nodes is exposed to a given pair of conditions c_k and c_l. Unfortunately, this closed-form solution is often excessively computationally expensive or impossible to solve directly. Instead, simulations can be leveraged to generate unbiased estimates of the above matrices.

With a Monte Carlo simulation, Kolaczyk and Csárdi (2020) explain, researchers can generate *n* treatment assignments using p_W and construct *N* x *n* matrices $\hat{I}_k$ for each

possible exposure condition $c_1, \ldots, c_K$. Then $(\hat{I}_k \hat{I}_k^T)/n$ serves as the estimator for $(I_k P I_k^T)$ and $(\hat{I}_k \hat{I}_l^T)/n$ estimates $(I_k P I_l^T)$.

Using these estimated probabilities, the familiar method of inverse probability weighting using Horvitz-Thompson estimators can be used to estimate each mean potential outcome. As long as each node has a positive exposure probability to each exposure condition, the estimators will be unbiased for the true mean potential outcome. It is simple to find $\tau(c_{10}, c_{00})$ and conduct desired contrasts using these estimated mean potential outcomes.

$$\hat{\bar{Y}}(c_k) = \frac{1}{N} \sum_{i=1}^{N} I_{[f(W,X_i)=c_k]} \frac{Y_i(c_k)}{p_i(c_k)}$$

$$Var[\hat{\bar{Y}}(c_k)]$$

$$= \frac{1}{N} [\sum_{i=1}^{N} p_i(c_k)(1 - p_i(c_k)(\frac{Y_i(c_k)}{p_i(c_k)})^2) + \sum_{i=1}^{N} \sum_{j \neq i} (p_{ij}(c_k) - p_i(c_k) p_j(c_k)) \frac{Y_i(c_k)}{p_i(c_k)} \frac{Y_j(c_k)}{p_j(c_k)}]$$

$$Var(\hat{\tau}(c_k, c_l)) = Var[\hat{\bar{Y}}(c_k)] + Var[\hat{\bar{Y}}(c_l)] - 2Cov[\hat{\bar{Y}}(c_k), \hat{\bar{Y}}(c_l)]$$

$$Cov[\hat{\bar{Y}}(c_k), \hat{\bar{Y}}(c_l)]$$

$$= \frac{1}{N^2} \left([\sum_{i=1}^{N} \sum_{j \neq i} \frac{\hat{\bar{Y}}_i(c_k)}{p_i(c_k)} \frac{\hat{\bar{Y}}_j(c_l)}{p_j(c_l)} (p_{ij}(c_k, c_l) - p_i(c_k) p_j(c_l))] - [\sum_{i=1}^{N} Y_i(c_k) Y_j(c_l)] \right)$$

Mean potential outcome variance estimates are unbiased as well, but only if all joint exposure probabilities are positive. Since the covariance of estimated mean potential outcomes remains unknown, the variance estimators for contrasts will inevitably include

bias. In ongoing research, statisticians have recently developed bias correction methods and ceilings for bias which can limit the extent of this problem (Kolaczyk & Csárdi, 2020).

The Case of Observational Studies

While the assignment mechanism is known in the context of controlled experiments, researchers may not have a clear understanding of said mechanisms in quasi experimental and observational study settings. Instead, they must use subject-matter expertise to compose a treatment assignment probability vector (Liu et. al., 2016; Van der Laan, 2014) or estimate this vector by relying on covariate data. Forastiere et. al. (2020) provide a novel approach that uses propensity scores and sub-classification to estimate treatment effects. Rather than computing effects as a series of contrasts, the propensity score method estimates main (direct) effects separately from spillover effects.

Take $\mu(c_k)$ to be the ADRF, an average dose-response function that captures the marginal mean of potential outcome $Y_i(c_k)$. Then define:

$$\mu(C) = \sum_{x \in \mathcal{X}} E\left[Y_i | C_i = c, \mathrm{X}_i = x, i \in V_g(A)\right] P(\mathrm{X}_i = x | i \in V_g(A))$$

where $\mathcal{X}$ is the set of possible covariate vectors and $V_g(A)$ denotes the neighborhood g within the adjacency matrix **A**.

Then use a familiar formulation of the propensity score, which denotes the probability that a unit receives a given exposure condition conditional on the unit's unit-level and neighborhood covariates:

$$\psi(w; g; x) = P(C_i = c|\mathrm{X}_i = x)$$

Now estimate a unit-level propensity score $\phi(1; X_i^W)$ using logistic regression based on the unit-level covariates X_i^{ind} alone. Predicted unit-level propensity scores are used to form subclasses $B_1, B_2, \ldots, B_J$. Subclasses should not only group units by unit-level propensity score, but also ensure balanced covariates within each subclass. Subclasses are further divided into B_j^g, denoting overlapping subsets of node neighborhoods.

Treating each class separately, produce neighborhood propensity scores, or "generalized propensity scores" (GPS), by using a logistic regression model trained on neighborhood covariates to predict a node's spillover effects. Since subclasses are divided by unit-level propensity scores and spillover effects are a function of the treatment assignment, predicted neighborhood propensity scores can be used to effectively predict potential outcomes for each node. To deal with the complexity of working with networks, it is recommended that researchers use a semi-parametric model to predict potential outcomes.

Then each subclass' ADRF for a given exposure condition can be calculated as the average of estimated potential outcomes for that condition. Ultimately, the overall ADRF for exposure condition c_k can be derived by taking the weighted average of subclass-level ADRFs for c_k.

$$\widehat{\mu_J}(c) = \frac{\sum_{B_j^g} \widehat{Y_\iota}(c)}{|B_j^g|}$$

$$\hat{\mu}(c) = \sum_{j=1}^{J} \widehat{\mu_j}\,(c)\,\frac{|B_j^g|}{N_g}$$

Borrowing exposure condition notation from the experimental case, we could then calculate the effect of direct treatment alone as $\hat{\mu}(c_{10})$, that of indirect treatment alone as $\hat{\mu}(c_{01})$, and that of both direct and indirect exposure to treatment as $\hat{\mu}(c_{11})$.

While Forastiere et. al. (2020) do not introduce closed-form equations for the calculation of estimate variance, various resampling techniques including bootstrapping can be leveraged to measure uncertainty. In addition, the authors present a series of simulation studies to demonstrate how their subclass and GPS method effectively reduces bias and RMSE. As long as SUTNVA holds and researchers account for confounding covariates, they claim, the method's estimators will be unbiased.

The problem of how to approach causal inference based on observational studies on networks remains an active subject of research. Forastiere et. al. (2020) suggest further study on resampling methods, Bayesian semiparametric methods, and implementation of the Horvitz-Thompson estimators in observational contexts. More generally, further research may explore the estimation of treatment effects when a network's structure is uncertain or even dynamic.

Methodology in Practice

While many of the above techniques have become commonplace in causal inference on networks, there is not yet a canonical set of methods or comprehensive R package due to

the field's novel nature. Instead, a combination of existing R packages provides researchers the analytic toolkit required. *igraph* allows users to easily create, edit, visualize, and summarize network structures. The *plm* package provides *vcovHC*, an object to aid other functions in robust covariance matrix estimation. To facilitate cluster-robust variance estimation with small-sample adjustments, *clubSandwich* works with a variety of standard statistical models. With these packages in use, package *lmtest* then allows for coefficient estimation and confidence interval construction. *cobalt*, although not demonstrated below, aids the researcher in gauging covariate balance. Finally, the *tidyverse* package may not be required for causal inference, but its use significantly eases the data preprocessing needed to analyze network data.

These R packages are demonstrated below to illustrate some core applications of modern methodologies. The demonstration centers data provided by Elizabeth L. Paluck, Hana R. Shepherd, and Peter Aronow in their 2020 paper "Changing Climates of Conflict: A Social Network Experiment in 56 Schools, New Jersey, 2012-2013." In the underlying study, education researchers investigated how anti-bullying interventions can affect student behavior, both on an individual level and when accounting for students' relationships within schools. Researchers randomly assigned 56 participating New Jersey public elementary schools to either control (no anti-bullying interventions), and treatment schools (interventions applied). Within each treatment school, they randomly selected 15% of the student body and invited selected students to participate in conflict resolution trainings.

To learn about students' opinions, actions, values, and interrelationships, surveys bookended the one-year intervention. Part of the surveys asked students to name the students they spent the most time with, the students they considered best friends, and the students with whom they had conflict. Results from those questions were used to create a network of students with self-reported interactions.

Paluck et. al. (2020) ultimately concluded that "treatment significantly reduced average levels of disciplinary reports of peer conflict in treatment compared with control schools," driving an estimated 30% decrease in the number of peer-reported conflicts within treatment schools. For the demonstration below, focus is restricted to a single treatment school and its student network. In addition, the research question is narrowed to: what is the effect of the anti-bullying intervention on the proportion of students that agree with the statement "[s]ometimes you have to be mean to others as a way to survive at this school" on the survey distributed at the end of the study.

Since researchers completely randomly assigned students within the school to treatment, the treatment assignment mechanism is known; this is an experiment rather than an observational study. When working with an observational study, readers should use methods presented by Forastiere et. al. (2020) of propensity scoring and subclassification in place of Monte Carlo simulations and Horvitz-Thompson estimation. Methods for working with observational studies are very much in development and less supported by R documentation, hence their omission from this demonstration.

For the sake of simplicity and a focus on methods rather than results, this analysis assumes covariate balance and includes only one covariate. In practice, it would be

essential to maximize covariate balance and to add impactful covariate data to models (where impact is determined by subject matter expertise, exploratory data analysis, DAGs, and the like).

Begin by installing and loading the R packages relevant for causal inference on networks.

```
#import packages needed for this application of causal inference in networks
library(igraph)

library(tidyverse)

library(lmtest)

library(clubSandwich)

library(plm)

library(cobalt)
```

As usual, load the dataset of interest and finish any preprocessing work. Paluck et. al. share generally cleaned and complete data, but most original datasets will require more care. In this case the data only requires filtering by the school in question (School ID 45) and exclusion of student records with essential information missing.

```
#load and import data from Paluck et. al. (accessible at https://www.icpsr.um
ich.edu/web/civicleads/studies/37070/publications)
load("ICPSR_37070_2/DS0001/df_37070_0001_Data.rda")
school_df <- da37070.0001

#select only students from school 45, a school assigned to treatment
school_1 = school_df %>% filter(SCHID == 45, substr(TREAT,1,1) == "(", substr
(CMOSW2,1,1) == "(",substr(CMOS,1,1) == "(")
```

With the dataset ready for analysis, compile a dataframe including only the unit-level IDs (in this case student IDs) and relevant covariates. This *school_1.v* dataframe will serve as a list of nodes for use by the *igraph* network constructor.

```
#create a list of vertices (nodes) from the list of students at school 20, al
ong with key characteristics of each student
school_1.v <- school_1 %>% select(ID, TREAT, GENC, HT, WT, RETURN, GR, starts
_with("ETH"), AAPP, HLANG, MOVE, MED, starts_with("LIVEW"), LOSTJOB, starts_w
ith("SIBS"), COLL, CELL, COMP, FSCH, starts_with("ACT"), starts_with("SM"), s
tarts_with("GAME"), starts_with("DN"), starts_with("PN"), starts_with("TOME")
, CIL, CFL, CSCA, CIHC, CBNP, CMOS, FLIB, FLSH, FLSD, ADSC, ADTS, ADNP, ADTH,
HOFC, HOSN, CRUT, CRUNT, starts_with("RTSM"), starts_with("NOM"), starts_with
("DE"), CMOSW2)

#minor data processing -- convert the treatment variable into a binary indica
tor; drop NA values and duplicate values
school_1.v$TREAT2 <- ifelse(school_1.v$TREAT == "(1) Treatment", 1, 0)
school_1.v <- school_1.v %>% filter(TREAT2 < 2)
school_1.v$ID <- as.numeric(school_1.v$ID)
school_1.v <- rbind(school_1.v %>% filter(ID != 859), school_1.v %>% filter(I
D == 859, GENC == "(1) Boy", RETURN == "(0) New to school")) %>% arrange(ID)

school_1.v[1:3,1:5]
```

```
##   ID                          TREAT     GENC               HT          WT
## 1  4 (0) Not treatment or control (0) Girl (4) 60-62 inches   (2) 86-95
## 2  5 (0) Not treatment or control (0) Girl (3) 57-59 inches (5) 116-125
## 3  8 (0) Not treatment or control (0) Girl (4) 60-62 inches (4) 106-115
```

Next, create a second dataframe *school_1.e* to store edge data: the collection of pairwise relationships between units. The IDs in each row can be matched with student IDs from the node dataset *school_1.v*. As stated earlier, students' self-reported regular contacts, best friends, and enemies comprise the relationships for this network. To avoid errors, ensure that there are no duplicate node-to-node relationships listed.

```
#create a dataframe with two columns, where each row will represent an edge:
the first capturing each student ID, and the second capturing IDs of students
who were identified as friends, best friends, and peers with whom the first s
tudent has conflict
school_1.e <- data.frame(ID = rep((as.numeric(school_1$ID)),17), NEIGHBOR = c
(school_1$ST1, school_1$ST2, school_1$ST3, school_1$ST4, school_1$ST5, school
_1$ST6, school_1$ST7, school_1$ST8, school_1$ST9, school_1$ST10, school_1$CN1
, school_1$CN2, school_1$CN3, school_1$CN4, school_1$CN5, school_1$BF1, schoo
l_1$BF2)) %>% arrange(ID, NEIGHBOR)

#ensure that there are no duplicate edges
school_1.e <- (school_1.e[complete.cases(school_1.e),] %>% filter(ID != 1, NE
IGHBOR != 1)) %>% arrange(ID, NEIGHBOR)
school_1.e <- school_1.e[school_1.e$NEIGHBOR %in% school_1.v$ID,]

head(school_1.e,3)

##   ID NEIGHBOR
## 3  4      127
## 4  4      127
## 7  5       93
```

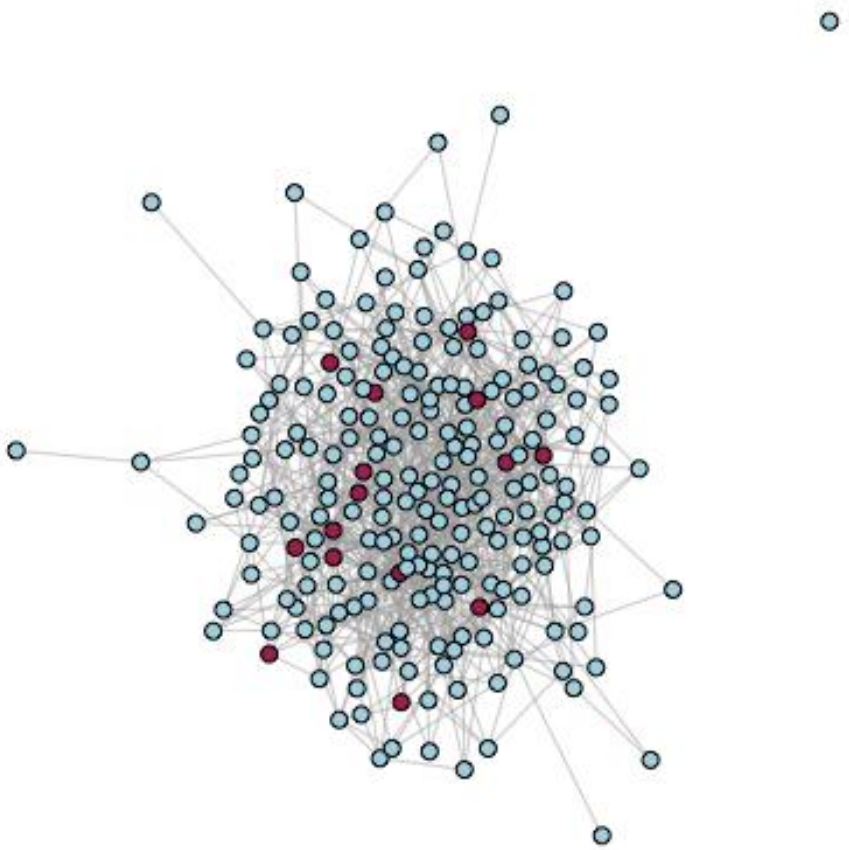
With both the nodes dataframe and the edges dataframe complete, the *igraph* function *graph_from_data_frame* can be invoked to create the network object itself. The *simplify* function then converts the network to a simple graph with no weighted edges, multiple edges, or loops.

igraph provides an extensive selection of visualization options for viewing the complete network. These include the ability to color-code specific nodes and choose the layout algorithm for network display. Below the *layout_nicely* option allows *igraph* to select which layout is most appropriate given the network structure. Pink nodes represent students who are assigned to the anti-bullying intervention, while blue nodes represent students assigned to control.

```r
#use the igraph package to create a graph object based on the node dataframe
and edge dataframe; simplify the graph to remove any remaining multi-edges an
d loops
school_1.g = graph_from_data_frame(school_1.e, directed = FALSE, vertices = s
chool_1.v)
school_1.g = igraph::simplify(school_1.g)

#adjust visualization settings to color directly treated nodes in red, and th
e control nodes in blue
V(school_1.g)$color <- ifelse(V(school_1.g)$TREAT2 == 1, "maroon", "light blu
e")
igraph_options(vertex.size = 4, vertex.label = NA, edge.arrow.size = 0.25, ed
ge.width = 0.5)

#plot by letting igraph choose a "nice" layout based on the graph structure
par(mar=c(0,0,0,0))
plot(school_1.g, layout = layout_nicely)
```



Causal inference itself begins by instantiating a vector to capture treatment assignments as well as an adjacency matrix that represents the network structure. Vector *I.ex.nbrs* draws on the treatment assignment vector and adjacency matrix to show which nodes receive indirect treatment.

The use of *I.ex.nbrs* and the simplicity of its calculation reflect use of the elementary exposure mapping introduced earlier. Again, units can be exposed to direct exposure (assignment to treatment), no exposure (assignment to control), only indirect exposure (if they are not assigned treatment but at least one of their neighbors is), and direct and indirect exposure (both the unit in question and at least one of its neighbors were assigned treatment). A more complex exposure mapping may be implemented here, likely a function with taking the treatment assignment vector and adjacency matrix as parameters.

```
#create a binary vector to hold the treatment assignment
trt_vector <- V(school_1.g)$TREAT2

#create an adjacency matrix based on our graph, and create a second vector to
capture which nodes receive indirect treatment; a more complex exposure mappi
ng could be implemented here
A = as_adjacency_matrix(school_1.g)
I.ex.nbrs <- as.numeric(trt_vector%*%A > 0)
```

Treatment vectors and indirect exposure vectors are then combined to create binary vectors, each of which represents whether each node receives a given exposure condition. These binary vectors can be appended to the dataframe *school_1.v* with node data.

```
#using the direct and indirect treatment vectors, identify which nodes receiv
e which exact exposure condition
school_1.v$both = trt_vector*I.ex.nbrs
school_1.v$indirect = (1-trt_vector)*I.ex.nbrs
school_1.v$direct = trt_vector*(1-I.ex.nbrs)
school_1.v$neither = (1-trt_vector)*(1-I.ex.nbrs)
```

Attention now turns to the use of Monte Carlo simulations to estimate the individual and joint exposure probabilities for each exposure condition. Increasing *n*, the number of simulation runs, decreases the bias of estimators.

If the treatment assignment mechanism is complete randomization, it is straightforward to randomly sample from all nodes without replacement until the number of units under treatment is reached. More involved treatment assignment mechanisms could be substituted at this point. It is also possible, although not well-studied, to replace exposure probabilities with propensity scores at this point.

With direct treatment assignment complete, an exposure mapping function can map assignments to the selected exposure conditions. For each exposure condition binary vectors are constructed during iterations simulation and appended to the appropriate exposure condition binary matrix. Once all simulation runs are complete, these exposure condition matrices can be used to estimate individual and joint exposure probabilities of each node to each exposure condition.

```
set.seed(42)

num_v <- vcount(school_1.g) #capture the number of vertices in the graph (stu
dents)
```

```
num_trt <- 25  # Number of treated students -- should reflect the true experi
ment
n <- 5e5 # Number of Monte Carlo trials -- more trials will provide lesser bi
as

#set up empty matrices to capture the simulation-generated exposure probabili
ties for each exposure condition
I11 <- matrix(,nrow=num_v,ncol=n)
I10 <- matrix(,nrow=num_v,ncol=n)
I01 <- matrix(,nrow=num_v,ncol=n)
I00 <- matrix(,nrow=num_v,ncol=n)

#Run the Monte Carlo simulation
for(i in 1:n){
   #randomly assign treatment to 25 students
   z <- rep(0,num_v)
   reps.ind <- sample((1:num_v),num_trt,replace=FALSE)
   z[reps.ind] <- 1

   #identify which nodes receive indirect treatment
   reps.nbrs <- as.numeric(z%*%A > 0)

   #store each randomly generated vector of exposure conditions in our matric
es
   I11[,i] <- z*reps.nbrs
   I10[,i] <- z*(1-reps.nbrs)
   I01[,i] <- (1-z)*reps.nbrs
   I00[,i] <- (1-z)*(1-reps.nbrs)
}

#calculate the estimated matrix of individual and joint exposure probabilitie
s to each of the exposure conditions
I11.11 <- I11%*%t(I11)/n
I10.10 <- I10%*%t(I10)/n
I01.01 <- I01%*%t(I01)/n
I00.00 <- I00%*%t(I00)/n
```

Calculate relevant summary statistics based on the observed data and grouped by each exposure condition. The variable of interest in this example is binary (did the student agree or disagree with the statement: “sometimes you have to be mean to others as a way to survive at this school”?). Hence the proportion of students who agreed with the statement under each exposure condition can serve as the relevant statistic.

```
#calculate the proportion of students who did and did not agree with the stat
ement under each exposure condition
school_1.v %>% filter(substr(CMOSW2,1,1) == "(") %>% group_by(both, indirect,
direct, CMOSW2) %>% summarise(count = n()) %>% mutate(prop = count/sum(count)
)

## `summarise()` has grouped output by 'both', 'indirect', 'direct'. You can
override using the `.groups` argument.

## # A tibble: 7 x 6
## # Groups:   both, indirect, direct [4]
##    both indirect direct CMOSW2          count  prop
##   <dbl>    <dbl>  <dbl> <fct>           <int> <dbl>
## 1     0        0      0 (0) Don't agree    53 0.5
## 2     0        0      0 (1) Agree          53 0.5
## 3     0        0      1 (0) Don't agree     8 1
## 4     0        1      0 (0) Don't agree    37 0.346
## 5     0        1      0 (1) Agree          70 0.654
## 6     1        0      0 (0) Don't agree     3 0.429
## 7     1        0      0 (1) Agree           4 0.571
```

Drawing on both observed and simulated results, an initial estimate of average treatment effects can be calculated. Observed exposure condition vectors are paired with exposure probability data to reweigh the summary statistics. Using nodes that received no treatment as a baseline, practitioners can easily calculate each exposure condition's ATE.

```
#store the proportion in each group who did agree with the statement
O.c11 <- 0.5714286
O.c10 <- 0
O.c01 <- 0.6542056
O.c00 <- 0.5

#store the vectors of students who received direct and indirect treatment
z <- school_1.v$TREAT2
reps.nbrs <- as.numeric(z%*%A > 0)

#recapture vectors of each exposure condition
c11 <- z*reps.nbrs
c10 <- z*(1-reps.nbrs)
```

```
c01 <- (1-z)*reps.nbrs
c00 <- (1-z)*(1-reps.nbrs)

#calculate the individual exposure probabilities for each condition and each
individual by taking the diagonal of estimated exposure probability matrices
(slight imputation to avoid the issue of probabilities equal to 0)
d11 <- diag(I11.11)
d11[d11 == 0] <- mean(d11)

d00 <- diag(I00.00)
d00[d00 == 0] <- mean(d00)

d10 <- diag(I10.10)
d10[d10 == 0] <- mean(d10)

d01 <- diag(I01.01)
d01[d01 == 0] <- mean(d01)

#estimate mean potential outcomes under each exposure condition
Obar.c11 <- O.c11*mean(c11/d11)
Obar.c10 <- O.c10*mean(c10/d10)
Obar.c01 <- O.c01*mean(c01/d01)
Obar.c00 <- O.c00*mean(c00/d00)

#calculate average treatment effects
print(c(Obar.c11,Obar.c10,Obar.c01)-Obar.c00)
```

```
## [1] -0.355172515 -0.609541508 -0.004679465
```

However, this method of adjusting statistics neither accounts for the binary nature of the response variable nor allows for the inclusion of important covariates. By instead using a vector with inverse-probability weights in a logistic model, practicioners can exercise more discretion in adding covariates and adjusting the model's specification.

Here, a weighted logistic regression model that includes the students' response to the statement in the first survey (variable *CMOS*) is used to estimate the ATEs. Again, in practice more influential covariates such as *age*, *gender*, *grade* and so on should also be included.

```r
#create a weights vector for inverse-probability weighting
ipw <- 1/(school_1.v$both * (diag(I11.11)+0.1) + school_1.v$direct * (diag(I1
0.10)+0.1) + school_1.v$indirect * (diag(I01.01)+0.1) + school_1.v$neither *
(diag(I00.00)+0.1))

#fit a weighted logistic regression model with the response and exposure cond
itions
mod <- glm(CMOSW2~both+indirect+direct+CMOS, data = school_1.v, family=binomi
al(link = "logit"), weights=ipw)

#fit and evaluate the model with robust error handling
coeftest(mod, vcov = vcovHC, type = "HC2")

##
## z test of coefficients:
##
##                Estimate Std. Error  z value  Pr(>|z|)
## (Intercept)    -0.53484    0.25448  -2.1017   0.03558 *
## both            0.24579    0.99174   0.2478   0.80426
## indirect        0.67414    0.31131   2.1655   0.03035 *
## direct        -17.42072    0.59035 -29.5093 < 2.2e-16 ***
## CMOS(1) Agree   1.42600    0.36245   3.9343 8.342e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

confint(mod, vcov=vcovHC, type = "HC2")

## Waiting for profiling to be done...

##                      2.5 %     97.5 %
## (Intercept)     -0.8713045 -0.2068325
## both            -0.4637840  0.9666703
## indirect         0.2400114  1.1158386
## direct        -203.4770166  3.1497346
## CMOS(1) Agree    1.0015867  1.8656595
```

The above coefficients can be transformed from the log-odds scale to odds ratios for a better understanding of their impact on students' responses.

```r
#transform coefficients to interpret treatments' relation into odds ratios
exp(coef(mod))
```

```
##   (Intercept)          both      indirect        direct CMOS(1) Agree
##  5.857652e-01  1.278634e+00  1.962347e+00  2.718179e-08  4.162028e+00
```

By working through and building on this demonstration, practitioners can effectively learn to estimate the average treatment effect under different exposure conditions. In the simplest case presented above, condition exposure probabilities can be estimated using an exposure mapping, knowledge of the treatment assignment mechanism, and Monte Carlo simulations. Different treatment assignment mechanisms and exposure mappings can be implemented within the structure presented. The methodology can also be modified for analysis of observational studies, starting by incorporating propensity scores rather than exposure probabilities.

Conclusion

Whether working with experimental or observational studies, researchers must account for the complexities introduced by network structures to conduct causal inference. Interference between units and the impact of the treatment assignment mechanism on potential outcomes violate the traditional SUTVA assumption.

But by working with exposure mappings, researchers can constrain the multitude of possible treatment assignments into a limited number of exposure conditions. Through Monte Carlo simulations and use of the Horvitz-Thompson estimator, exposure condition probabilities can be used to adjust and improve ATE estimates. For observational studies, propensity scores and a subclassification scheme similarly produce unbiased results.

Especially in the cases of observational studies, uncertainty about network structures, and dynamic network structures, causal inference on networks is a new and active research area.

As a result, there are currently no universally accepted R packages dedicated to applying this theory. With a combination of pre-existing R packages, however, practitioners can implement, test, and expand on the methodologies presented in the literature. The above demonstration, based on "Changing climates of conflict: A social network experiment in 56 schools" by Paluck et. al., provides readers a starting point for conducting their own causal inference on networks.

Works Cited

Aronow, P. M., & Samii, C. (2017). Estimating average causal effects under general interference, with application to a social network experiment. The Annals of Applied Statistics, 11(4), 1912–1947. https://doi.org/10.1214/16-AOAS1005

Barabási, A.-L., & Pósfai, M. (2016). Network science . Cambridge University Press.

The igraph core team. (2019, April 22). Get started with R igraph. igraph R package. https://igraph.org/r/.

Kolaczyk, E. D., & Csárdi, G. (2020). Statistical analysis of network data with r. ProQuest Ebook Central https://ebookcentral.proquest.com

Imbens, G., & Rubin, D. (2015). Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction. Cambridge: Cambridge University Press. doi:10.1017/CBO9781139025751

Forastiere, L., Airoldi, E. M., & Mealli, F. (2020). Identification and estimation of treatment and interference effects in observational studies on networks. Journal of the American Statistical Association, 1-18. doi:10.1080/01621459.2020.1768100

Liu, L., Hudgens, M. G., & Becker-Dreps, S. (2016). On inverse probability-weighted estimators in the presence of interference. Biometrika, 103(4), 829–842. https://doi.org/10.1093/biomet/asw047

Morgan, S. L., & Winship, C. (2007). Counterfactuals and causal inference : Methods and principles for social research. ProQuest Ebook Central https://ebookcentral.proquest.com

Paluck, E. L., Shepherd, H., & Aronow, P. M. (2016). Changing climates of conflict: A social network experiment in 56 schools. Proceedings of the National Academy of Sciences, 113(3), 566-571. doi:10.1073/pnas.1514483113

van der Laan, M. J. (2014). Causal Inference for a Population of Causally Connected Units. Journal of Causal Inference, 2(1), 13–74. https://doi.org/10.1515/jci-2013-0002