Network Diffusion of Innovations in R: Introducing **netdiffuseR**

George G. Vega Yon Stephanie R. Dyal Timothy B. Hayes Thomas W. Valente

University of Southern California

1 Motivation

1.1 Network Diffusion of Innovations

- Tries to explain how new ideas and practices (innovations) spread within and between communities.
- While a lot of factors have been shown to influence diffusion (Spatial, Economic, Cultural, Biological, etc.), Social Networks is a prominent one.
- More complex than *contagion*, a single tie is no longer enough for an innovation to spread across a social system.
- We think of this in terms of adoption thresholds and social exposure.

1.2 Network thresholds

Network thresholds (Valente, 1995), τ, are defined as the required proportion or number of neighbors that leads you to adopt a particular behavior (innovation), $a = 1$. In (very) general terms

$$a_i = \begin{cases} 1 \text{ if } \tau_i \leq E_i \\ 0 \text{ Otherwise} \end{cases} \qquad E_i \equiv \frac{\sum_{j\neq i} \mathbf{x}_{ij} a_j}{\sum_{j\neq i} \mathbf{x}_{ij}}$$

Where E_i is i's exposure to the innovation and $\mathbf{X} = \{\mathbf{x}_{ij}\}$ is the adjacency matrix (the network). This can be generalized and extended to include covariates and other weighting schemes (that's what **netdiffuseR** is all about).

2 netdiffuseR: Network Diffusion of Innovations in R

- Is designed for visualizing, analyzing and simulating network diffusion data,
- Core functions are written in C++, which makes it fast,
- Graphs are stored as sparse matrices, so it can handle big data (> 4 billion elements),
- Already on CRAN (2 iterations) with 700 downloads since its first version, Feb 2016,
- A lot of features to easily read and manage network (dynamic) data,

```
rdiffnet(
  n=1e3, t=5, # Number of vertices and time points
  seed.nodes     = "random", # Set of initial adopters
  seed.p.adopt   = .15, # Proportion of initial adopters
  seed.graph     = "small-world", # Baseline graph
  rgraph.args    = list(p=4), # Arguments for the rgraph call
  rewire.args    = list(algorithm="swap", p=5), # Rewiring args after time 1
  threshold.dist = function(x) runif(1, .4, .8), # Distribution of thresholds
  exposure.args  = list(normalized=TRUE) # Args for computing exposures
)
```

Figure 1: Simulating diffusion networks is easy (and fast): Here we are simulating a diffusion network with 1,000 vertices and 5 time points. The baseline graph is small-world network. Further arguments are specified.

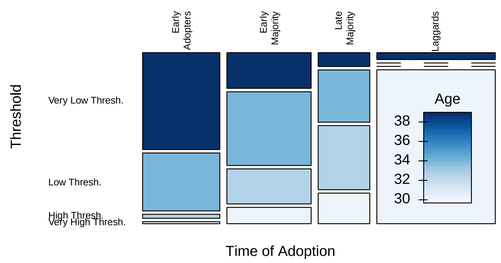

Figure 2: Adopters classification in the Korean Family data: From the mosaic we can see that in general low threshold levels and early adoption seem to be positively correlated with age.

3 Visualizing Diffusion Data

Most of the diffusion statistics in **netdiffuseR** can be plotted. The following figures show some of these.

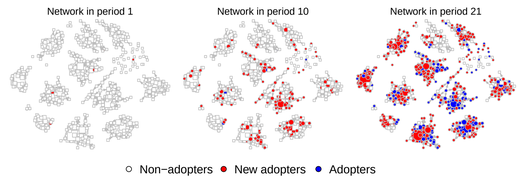

Figure 3: Diffusion of Hybrid Corn Seed (Brazilian Farmers) I: The set of adopters and non-adopters is shown for three time periods. At each time period non-adopters are colored white, new adopters in red, and adopters from previous periods in blue.

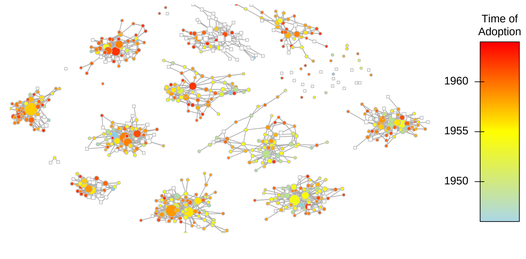

Figure 4: Diffusion of Hybrid Corn Seed (Brazilian Farmers) II: Same information as in the previous figure but using a different method. Early adopters are blue and laggards red. Non-adopters are shown in white.

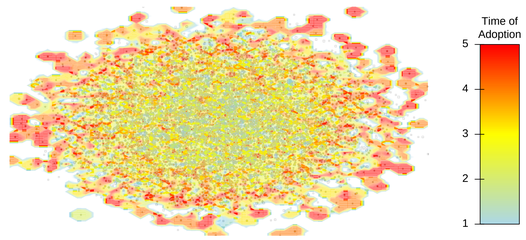

Figure 5: Time of adoption on a scale-free diffusion network: Simulated with 50K vertices, diffusion started at the central nodes. The figure has two layers, the first one is the graph itself with colored vertices as in the previous figure, and the second layer is a bi-dimmensional kernel smooth of the times of adoption using the graph structure.

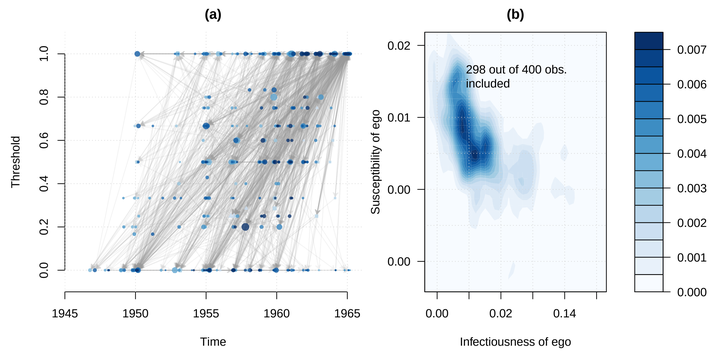

Figure 6: Thresholds and infectiousness/susceptibility: (a) Shows threshold levels vs times of adoption for the Brazilian Farmers data, and (b) Shows the joint distribution (smoothed) of infectiousness and susceptibility of a random bernoulli diffusion graph.

4 Statistical inference

- **netdiffuseR** is being used as a platform for implementing a new non-parametric test for network + behavioral data.
- The rewiring–which preserves degree-sequence–is done to evaluate a particular statistic that combines both network structure and a behavioral variable.
- The following illustrates its implementation on the three datasets included in **netdiffuseR**. The null hypothesis is that times of adoption are independent from the network structure:

$$H_0 : \mathcal{G} \not\perp \text{Time of Adoption}$$
$$H_a : \mathcal{G} \perp \text{Time of Adoption}$$

For each graph we generated 1,000 rewired versions of the original graph. Each rewired version took roughly 400,000 steps. All done in parallel using the **boot** package (so no effort for the researcher)

	Korean Family	Brazilian Farmers	Medical Innovation
p-val	0.1440	0.0000	0.8440
Obs. Avg. threshold t_0	0.6199	0.6191	0.6067
Sim Avg. threshold $\bar{t}$	0.6107	0.5813	0.6026

Table 1: Test results for each dataset: The first row shows p-values according to the null presented above; and the last two rows present the average threshold for the observed data and the null. Only the Brazilian Farmers data suggest that times of adoption are somehow dependent on network structure.

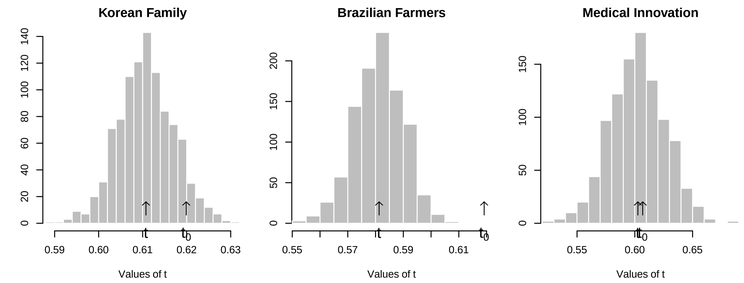

Figure 7: Null distribution vs observed average threshold: Same as before, there's evidence of adoption depending on network structure for the Brazilian Farmers data.

More info at the project website:

https://github.com/USCCANA/netdiffuseR

Acknowledgements

netdiffuseR *was created with the support of grant R01 CA157577 from the National Cancer Institute/National Institutes of Health.*

netdiffuseR *has benefited from input provided by participants of the Center for Applied Network Analysis (CANA), and the Computational Social Science Lab (CSSL) at the University of Southern California.*

netdiffuseR*'s original code was developed by* **Thomas Valente***, improved by* **Stephanie Dyal** *and* **Timothy Hayes***, and extended by* **George Vega Yon**.