Network Regression

Anonymous Author(s)
Affiliation
Address
email

Abstract

Data often come in the form of networks such as friendships in a social network or links in the Web. Consider the problem of having a single set of n vertices connected through k different networks with adjacency matrices $\mathbf{A}_1, \ldots, \mathbf{A}_k$. There is an underlying matrix of interest $\mathbf{M} = \sum_{i=1}^{k} \beta_i \mathbf{A}_i$ which is an unknown linear combination of these known matrices. We observe a $\mathbf{y} \in \mathbb{R}^{1 \times n}$ where the elements correspond to values on the vertices at some point in time, and we also observe $\mathbf{y}_1 \approx \mathbf{z} = \mathbf{y}\mathbf{M}$. This paper demonstrates how to estimate the unknown values of the β_i through well understood regression techniques.

1 Introduction

Consider the problem from epidemiology: an outbreak of some disease has occurred at a school and it spreads from one person to the next through a friendship network and through grades. We observe who is sick at two time points, and we want to determine how important each of these networks (friendship, grades) is in the spread of disease. Or consider the problem concerning Web traffic: Web pages are connected through hyperlinks of several different types, and viewers follow one these different types of links with different probabilities. We want to recover the probabilities corresponding to the types of links that are followed based on the total number of page views of each website. Many questions like these which can be modeled by propagation or walking on a network can be answered with network regression.

Define $\mathcal{V}$ as a collection of n vertices, and a network $\mathbf{A}_i$ as a matrix in $\mathbb{R}^{n \times n}$ where the (u, v) entry corresponds to the weight $w_{i(u,v)} \in \mathbb{R}$ of the directed edge from u to v where $u, v \in \mathcal{V}$. Let $\mathbf{A}_1, \ldots, \mathbf{A}_k$ be a collection of known networks on the same set of n vertices. We observe $\mathbf{y} \in \mathbb{R}^{1 \times n}$ where the elements correspond to values on the n vertices in $\mathcal{V}$ and then at some later time point we observe $\mathbf{y}_1 \in \mathbb{R}^{1 \times n}$ where $\mathbf{y}_1 \approx \mathbf{z} = \mathbf{y}(\sum_{i=1}^{k} \beta_i \mathbf{A}_i) = \mathbf{y}\mathbf{M}$. From $\mathbf{A}_1, \ldots, \mathbf{A}_k$, $\mathbf{y}$, and $\mathbf{y}_1$ we want to estimate $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_k]' \in \mathbb{R}^{k \times 1}$ (where the prime indicates the transpose of a matrix) with $\hat{\boldsymbol{\beta}}$.

Let $\mathbf{X} = [(\mathbf{y}\mathbf{A}_1)', \cdots, (\mathbf{y}\mathbf{A}_k)']$ be an $n \times k$ matrix and observe that $\mathbf{y}\mathbf{M} = \mathbf{y} \sum_{i=1}^{k} \beta_i \mathbf{A}_i = \sum_{i=1}^{k} \beta_i \mathbf{y}\mathbf{A}_i = (\mathbf{X}\boldsymbol{\beta})'$. If $\mathbf{y}_1 = \mathbf{z}$ and the columns of $\mathbf{X}$ are linearly independent, then it is possible to calculate $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}$ exactly. One way to do this via ordinary least squares regression:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|(\mathbf{y}\mathbf{M} - \mathbf{y}_1)'\|_2^2 = \arg\min_{\boldsymbol{\beta}} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}_1'\|_2^2. \tag{1}$$

The interesting case is where $\mathbf{y}_1$ is only an approximation of $\mathbf{z}$. In this case, equation (1) still serves as a least squares approximation of $\hat{\mathbf{y}}_1' = \mathbf{X}\hat{\boldsymbol{\beta}}$ to $\mathbf{y}_1'$. This methodology is easily extensible beyond the least squares loss functions to different types of linear regression (such as generalized linear models) or other supervised learning techniques. By constructing the model to predict $\boldsymbol{\beta}$ by regressing $\mathbf{y}_1$ onto $\mathbf{X}$, the regression technique we use determines statistical characteristics of the estimates. In many types of regression these characteristics are already well understood.

Section 2 overviews the literature that influenced this research emphasizing developments in generalized linear models, PageRank, and epidemiological models. Section 3 concerns special cases in network regression such as when $\mathbf{y}$ is approximately the stationary distribution on $\mathbf{M}$, interpreting an intercept term in $\mathbf{X}$, including polynomial network variables, and interpreting non-network variables in $\mathbf{X}$. Section 4 examines the specific case of linear network regression with theoretical basics, a simulation, and an example from the New York City subway lines. Section 5 addresses logistic network regression covering theory, a simulation, and an example from the spread of the H1N1 flu through counties in the USA in May 2009. Section 6 discusses other applications where network regression could be used and identifies some important open questions.

2 Literature Review

2.1 Generalized Linear Models

Generalized linear models were first proposed in 1972 [22], and there have been many books devoted to this topic since then [21, 14]. Generalized linear models are a flexible generalization of ordinary least squares regression which allow for a link function g between the linear model and the response variable such that $\mathbb{E}(Y) = g^{-1}(\eta)$. Each outcome Y is believed to be generated from a distribution in the exponential family as $f_X(x|\theta) = h(x)g(\theta)\exp(\eta(\theta)T(x))$ where $T(x), h(x), \eta(\theta)$, and $g(\theta)$ are known functions and θ is a parameter and $\eta = \mathbf{X}\boldsymbol{\beta}$ is a linear predictor.

Some of the more common distributions that use this methodology are the normal, exponential, gamma, inverse Gaussian, Poisson, binomial, and multinomial. This paper will discuss the normal distribution case (the same as ordinary least squares) and the binomial case (the same as logistic regression). Although these are the only two methods covered in the current paper, generalized linear models make for natural interpretations in applications of network regression.

2.2 PageRank and "Hubs and Authorities"

Not only does network regression immediately inherit the flexibility of regression, but it also allows for many of the developments and specializations on PageRank. Of particular interest are the interpretations and developments in personalized PageRank (see [13, 12, 17]). Personalized PageRank is mathematically simple to model, is computationally tractable to use on massive data sets, and has a natural interpretation both in PageRank and in network regression. The relationship network regression has to PageRank is discussed in section 3.2, and the relationship to personalized PageRank is discussed in section 3.4.

Some data can be represented as a bipartite network between vertices $\mathcal{V}$ (the vertices of interest) and $\mathcal{U}$ (additional vertices that make up the other part). If we take two steps on the bipartite network (starting from a vertex in $\mathcal{V}$, stepping to a vertex in $\mathcal{U}$, and returning to a vertex in $\mathcal{V}$ as discussed in section 3.3), then this model is similar to Hubs and Authorities [19]. In Network regression, the vertices $\mathcal{V}$ (the network of interest) are the authorities and the vertices in $\mathcal{U}$ (the other part of the bipartite graph) act as hubs. A good starting point to relate that literature to network regression is in [9] and [11].

2.3 Modeling Diffusion Through Networks and Nodal Influence

Compartmental models in epidemiology and information diffusion on networks also had a large influence in network regression (for surveys on the relevant network diffusion literature see [27, 16, 6, 8]). The SIS model (where vertices are susceptible to infection, become infected, and then return to a susceptible state) is the most similar model to logistic network regression (see section 5).

There is often an interest in how to prevent outbreaks (in epidemiology) or cause outbreaks (in information diffusion) on a single network. Since 1950 [3] there has been research on which vertices are most influential in a network (for different measures, see [25, 23, 10, 4]). It would be interesting to see how these methods perform on multiple networks (as in the network regression model) and develop these methods to take into account exogenous information in the way that network regression takes the vector $\mathbf{y}$.

3 Special Cases and Interpreting the Explanatory Variables

In linear regression, there are several terms which are natural include in the regression model. Some of the most common ones are the intercept term, polynomial terms, and factor terms. Analogies of these (and a few other special cases) relating to network regression are addressed in this section.

3.1 When $\mathbf{y}$ is a Stationary Distribution

When $\mathbf{y} \approx \mathbf{z}$ where $\mathbf{z} = \lambda \mathbf{z}\mathbf{M}$ (for some $\lambda > 0$) is the principal eigenvector (the eigenvector corresponding to the largest eigenvalue), we can estimate $\boldsymbol{\beta}$ (up to proportionality) with the single observation on the vertices $\mathbf{y}$ (i.e. without observing $\mathbf{y}_1$). This could be the case when $\mathbf{y}$ is the observed proportion of page views for each website in a collection of websites. To estimate $\boldsymbol{\beta}$, use the same methodology as in section 1: construct $\mathbf{X} = [(\mathbf{y}\mathbf{A}_1)', \cdots, (\mathbf{y}\mathbf{A}_k)']$ and regress $\mathbf{y}$ onto $\mathbf{X}$ to calculate $\hat{\boldsymbol{\beta}}$. Although this technique could estimate the wrong eigenvector of $\mathbf{M}$ if we believe that there is an intercept term in $\mathbf{M}$ (as described in section 3.2) then there will be a spectral gap so the principal eigenvector should be recovered.

3.2 The Network Intercept Term and PageRank

In the case where $\mathbf{M}$ is a Markov chain, then it is important to include additional constraints. If each of the $\mathbf{A}_i$ that makes up $\mathbf{M}$ is row stochastic and has only non-negative values for $w_{i(u,v)}$, then by constraining $\mathbf{1}'\hat{\boldsymbol{\beta}} = 1$ and $\hat{\boldsymbol{\beta}} \geq 0$, $\hat{\mathbf{M}}$ will be a Markov chain. An important special case where $\mathbf{M}$ is a Markov chain is in the PageRank model.

The PageRank model was the foundation of Google's search technology [7]. In this model there is a network $\mathbf{A}$ where $\mathcal{V}$ is a collection of websites (where every website has at least one link out) and $w_{(u,v)} > 0$ if there is a link from u to v (otherwise $w_{(u,v)} = 0$) weighted so $\sum_v w_{(u,v)} = 1$ and $w_{(u,v)} = w_{(u,w)}$ if u has a link to both v and w. In the PageRank model, $\mathbf{M} = \beta\frac{1}{n}\mathbf{1}\mathbf{1}' + (1-\beta)\mathbf{A}$. Since $\mathbf{A}$ and $\frac{1}{n}\mathbf{1}\mathbf{1}'$ are both row stochastic, $\mathbf{M}$ will be the transition matrix of a Markov chain.

Usually $\beta = 0.15$ [24, 7], but in network regression β can be explicitly optimized if given $\mathbf{y}$ an observed proportion of visits to each website in $\mathcal{V}$. Following the technique in section 3.1, write $\mathbf{X} = [(\mathbf{y}(\frac{1}{n}\mathbf{1}\mathbf{1}'))', (\mathbf{y}\mathbf{A})'] = [\frac{1}{n}\mathbf{1}, (\mathbf{y}\mathbf{A})']$ and (for a squared error loss function) solve the convex optimization problem of minimizing $\|\mathbf{X}[\beta, (1-\beta)]' - \mathbf{y}'\|_2^2$ subject to $\beta \in [0, 1]$ (see [5]). So the complete network $\frac{1}{n}\mathbf{1}\mathbf{1}'$ acts as a scaled intercept term in this constrained linear regression problem.

The network intercept term $\frac{1}{n}\mathbf{1}\mathbf{1}'$ is practical to include in network models for several reasons. It ensures connectedness in $\hat{\mathbf{M}}$. It forces a spectral gap making it safe to assume that the random walk is mixing fast, which is particularly important when we have only one observation $\mathbf{y}$ which we believe is close to the principal eigenvector of $\mathbf{M}$. Since the matrix $\frac{1}{n}\mathbf{1}\mathbf{1}'$ is never actually constructed, this methodology takes advantage of the sparsity in $\mathbf{A}$ when solving for $\hat{\beta}$. As an added bonus, the intercept term is easy to incorporate into the vast statistical literature on regression.

3.3 Polynomial Terms, Bipartite Networks, and Cliques

As in linear regression models, it is easy to include an analogy to polynomial terms. For example, to include a squared term in a network regression, set $\mathbf{X} = [\frac{1}{n}\mathbf{1}, (\mathbf{y}\mathbf{A})', ((\mathbf{y}\mathbf{A})\mathbf{A})']$. Note that no matrix by matrix multiplication is necessary to construct $\mathbf{X}$. Using this same technique, we can also include interaction terms (although it is important to remember that matrix multiplication is not commutative, so the interpretation for including the term $((\mathbf{y}\mathbf{A}_1)\mathbf{A}_2)'$ is to first take a step on $\mathbf{A}_1$ and then a step on $\mathbf{A}_2$ whereas including the term $((\mathbf{y}\mathbf{A}_2)\mathbf{A}_1)'$ has the order of the steps reversed).

Extending this idea further, it is also possible to include many other types network structures. For example, if we believe that there is some other set of m vertices $\mathcal{U}$ such that $\mathcal{U} \cap \mathcal{V} = \emptyset$ and we have a bipartite network $\mathbf{B} \in \mathbb{R}^{n \times m}$ between $\mathcal{V}$ and $\mathcal{U}$ or a network $\mathbf{C}$ on $\mathcal{U}$ then we include complicated terms such as $((\mathbf{y}\mathbf{B})\mathbf{B}')'$ or $(((\mathbf{y}\mathbf{B})\mathbf{C})\mathbf{B}')'$. It is important to make sure not to include so many networks or combinations of networks that the model over fits the data.

Using two steps on bipartite networks, it is equivalent to construct the network intercept term by taking $\mathbf{B} = (1/\sqrt{n})\mathbf{1}$ to be the complete bipartite network between $\mathcal{V}$ and a single vertex $\mathcal{U} = \{u\}$ and taking two steps on this. Similarly, clique networks ($\frac{1}{|C|}\mathbf{1}_C\mathbf{1}_C{}'$, $C \subset \mathcal{V}$ where $w_{(u,v)} = 1/|C|$ for all $u, v \in C$ and 0 everywhere else) can be modeled by taking two steps on an $n \times 1$ bipartite network and can be included in the model by including the column $\frac{1}{|C|}(\mathbf{y}\mathbf{1}_C)\mathbf{1}_C$ (all entries in this column are either 0 or $\mathbf{y}\mathbf{1}_C$) in $\mathbf{X}$. In this sense, a clique in network regression is similar to a (scaled) factor variable in linear regression. Also like the intercept term, it is not necessary to ever construct the clique's adjacency matrix.

3.4 Non-Network Variables

When there is additional information on the vertices, we can include this information in $\mathbf{X}$ in the same way as in regression models. If a set of l variables $\tilde{\mathbf{X}} \in \mathbb{R}^{n\times l}$ has some influence on $\mathbf{z}'$ then it is possible to include a non-network term in the model as $\mathbf{X} = [\mathbf{y}\mathbf{A}_1, \ldots, \mathbf{y}\mathbf{A}_k, \tilde{\mathbf{X}}]$. This leads to the more general model $\mathbf{z}' = (\mathbf{y}\mathbf{M})' + \tilde{\mathbf{X}}\boldsymbol{\gamma}$ where $\boldsymbol{\gamma}$ is an $l \times 1$ vector of parameters, $\mathbf{y}_1 \approx \mathbf{z}$, and we regress $\mathbf{y}_1'$ onto $[\mathbf{X}, \tilde{\mathbf{X}}]$ (an $n \times (k + l)$ matrix). The reason for using a non-network variable in this way is if we believe there is some external source of new observations at $\mathbf{y}_1$.

This leads to a distinction between the network intercept term $\mathbf{y}\frac{1}{n}\mathbf{1}\mathbf{1}'$ and the external intercept term $\frac{1}{n}\mathbf{1}$ (these will be the same when $\mathbf{y}\mathbf{1} = 1$). In the PageRank model, another interpretation of the intercept term (as an external intercept) is that with probability $(1 - \beta)$, someone already walking on the network is picked to follow a link, and with probability β a new users starts walking on the network (starting at a page in $\mathcal{V}$ uniformly at random). In general, we can think of these non-network variables as network walkers being born at the vertices at rates given in the column vector of $\tilde{\mathbf{X}}\boldsymbol{\gamma}$.

A similar analogy can be drawn between a personalized PageRank network and a column in $\tilde{\mathbf{X}}$. Since we can personalize PageRank to any set of proportions on the vertices that we want, the interpretation of including a non-network term in $\mathbf{X}$ is that there are new observations in $\mathbf{y}_1$ coming from some source outside the network according to the columns of $\tilde{\mathbf{X}}$.

In some cases it might be interesting to distinguish between the network intercept term and an external intercept term (for example when the network intercept term is positive and the external intercept term is negative). With just $\mathbf{y}$ and $\mathbf{y}_1$ it is not possible to predict both the external intercept term parameter and the network intercept term parameter simultaneously (since the corresponding vectors in $\mathbf{X}$ will be exactly correlated). In more complicated models (such as a time series on networks with $\mathbf{y}, \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t$ with fixed $\boldsymbol{\beta}$, then we may be able to include both the external and network intercept terms. In the more general simpler case, we must decide which of these is the appropriate interpretation and include that in the model.

4 Linear Regression

4.1 Theory

Since network regression simply applies regression in a new way, it lends itself to the rich statistical literature already developed on regression. It is possible to use penalized regression models to insure uniqueness of $\hat{\mathbf{y}}$ or to allow for sparsity in the number of nonzero $\hat{\beta}_i$. Standard regression techniques apply when evaluating the significance of the variable $\mathbf{y}\mathbf{A}_i$ in $\mathbf{X}$. This can be used with standard model selection techniques to determine which networks to include in the model. Understanding network regression models is as straightforward as any other regression model.

In section 1 $\mathbf{y}_1$ an approximation of $\mathbf{z}$. In the case of ordinary least squares linear network regression, the appropriate model for this approximation is that $\mathbf{y}_1 = \mathbf{z} + \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon}$ is normal noise independent of $\mathbf{M}$ and with $\mathbb{E}(\boldsymbol{\epsilon}) = \mathbf{0}$.

$$\mathbb{E}(\mathbf{y}_1) = \mathbb{E}(\mathbf{y}\mathbf{M}) = \mathbb{E}(\mathbf{z}\mathbf{M}) + \mathbb{E}(\boldsymbol{\epsilon}\mathbf{M}) = \mathbf{z}\mathbf{M} \tag{2}$$

$$\text{Var}(\mathbf{y}_1) = \text{Var}(\mathbf{y}\mathbf{M}) = \mathbf{M}' \text{Var}(\mathbf{y})\mathbf{M} = \mathbf{M}' \text{Var}(\boldsymbol{\epsilon})\mathbf{M} \tag{3}$$

So for $\text{Var}(\boldsymbol{\epsilon}) = \sigma_{\boldsymbol{\epsilon}}^2$, if there are $\mathcal{O}(m)$ edges in $\mathbf{M}$ then the variance of $\mathbf{y}_1$ is bounded by $\mathcal{O}(n^2 m)$.

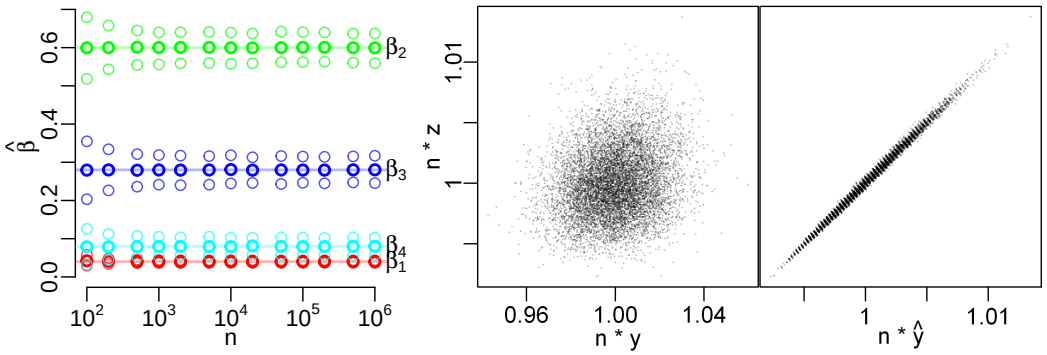

Figure 1: (Left) We plot estimates of $\hat{\boldsymbol{\beta}}$ versus n in where each $\mathbf{A}_i$ is a random directed graph with a fixed number of edges (multi-edges and loops allowed). Simulations were done 1,000 times for each value of n. From these simulations, the mean and the sample 5% and 95% are plotted against n (log scale). By exploiting sparse matrix structure, it is easy to do this on networks with 10^6 vertices. (Center) We plot $n\mathbf{z}$ versus $n\mathbf{y}$ for one simulation where $n = 10^4$. The correlation between $\mathbf{y}$ and $\mathbf{z}$ in this sample is 0.1895. (Right) We plot $n\mathbf{z}$ (on the same scale) versus $n\hat{\mathbf{y}}$ where $\hat{\mathbf{y}}$ is the fitted values from regressing $\mathbf{y}$ on the networks. The correlation of $\hat{\mathbf{y}}$ and $\mathbf{z}$ is 0.9966, so even with $\boldsymbol{\epsilon}$ quite large relative to $\mathbf{z}$, linear network regression is able to retrieve $\mathbf{z}$ almost exactly with $\hat{\mathbf{y}}$.

4.2 Simulation

This simulation is a simple demonstration of the theory in sections 1 and 3. We let $\mathbf{A}_1 = \mathbf{1}\mathbf{1}'$ be the complete network, and we construct $\mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4$ by making directed random graphs which are allowed to have multiple edges and self loops. $\mathbf{A}_2$ has $2n$ edges, $\mathbf{A}_3$ has $5n$ edges, and $\mathbf{A}_4$ has $15n$ edges. We fix $\boldsymbol{\beta} = [1, 15, 7, 2]'$ then let $\mathbf{z}$ (standardized so that $\mathbf{z}\mathbf{1} = 1$) be the principal eigenvector of $\mathbf{M} = \beta_1\mathbf{A}_1 + \beta_2\mathbf{A}_2 + \beta_2\mathbf{A}_3 + \beta_4\mathbf{A}_4$. We sample $\boldsymbol{\epsilon}$ as a vector of length n of independent normals with variance $2n^{-3}$ and set $\mathbf{y} = \mathbf{z} + \boldsymbol{\epsilon}$ and then standardize it to sum to 1 so that $\mathbf{y}\mathbf{A}_1 = \mathbf{1}$. We construct $\mathbf{X} = [\mathbf{1}, (\mathbf{y}\mathbf{A}_2)', (\mathbf{y}\mathbf{A}_3)', (\mathbf{y}\mathbf{A}_4)']$ and regress $\mathbf{y}'$ onto $\mathbf{X}$ to calculate $\hat{\boldsymbol{\beta}}$. This process is repeated 1,000 times for values of n from 100 to 1,000,000. Recall that with this model (in section 3.1 where we have $\mathbf{y}_1 = \mathbf{y}$ which we believe is approximately the principal eigenvector of $\mathbf{M}$) we can only estimate $\boldsymbol{\beta}$ up to a multiplicative constant, so we compare a scaled $\hat{\boldsymbol{\beta}}$ (so the elements sum to 1) to a scaled $\boldsymbol{\beta}$ (so the elements sum to 1). The true (scaled) $\boldsymbol{\beta}$ along with the mean and the empirical 0.05 and 0.95 quantile (scaled) $\hat{\boldsymbol{\beta}}$ are plotted as a against n in figure 1. As expected, the estimates of these parameters are (approximately) constant as a function of n (except for small n).

4.3 Real Data Example: New York City Subways

The New York City (NYC) Metropolitan Transportation Authority (MTA) subway system connects subway stations all across the city. On their website http://www.mta.info/developers/ the MTA freely share lots of data, including the number of entries and exits at each turnstile in each subway stations in NYC. From this data, we have the total number of entries and exits recorded at each subway station between midnight on April 30 and midnight on May 7, 2011.

Two subway stations (St. Lawrence Av and Elder Av, both on the '6' line) were closed and undergoing rehabilitation at the time (and so were excluded from this dataset). The data for two stations (the Aqueduct Racetrack stop on the 'A' line and a station which only had identifying tags connecting to the Newark Airport Express bus) were excluded from the original dataset because those stations did not correctly identify any turnstile tags in the turnstile data. All of the other subway stations are included. 9 turnstiles malfunction during that week (every effort has been made to correct for this, and at most 2 hours of reads from each of those turnstiles were lost). The number of exits is about 19% less than the number of entries, and 3 subway stations have 0 recorded exits.

Assume that passengers enter a subway station, ride on one subway line which stops at that station, and then exit. Although a more complicated model may be much more appropriate, this one suffices for the demonstration. In this dataset there are $n = 387$ subway stations which are the vertices. There are $k = 22$ subway lines in NYC, and each of these lines represents an adjacency matrix

Line:	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'9'	'A'	'B'	'C'	'D'	'E'	'F'	'G'	'J'	'L'	'M'	'N'	'Q'	'R'	'Z'
step:	8.2	0	4.8	5.4	3.6	7.9	10	0	2.8	2.4	4.0	3.1	5.1	3.2	0	4.2	8.9	0	4.8	6.4	4.3	0
cvx:	8.1	0.7	4.6	5.6	3.1	8.0	10	0.0	2.8	2.4	4.0	3.1	5.1	3.2	0.0	3.5	9.0	0.0	4.8	6.4	4.3	0.8

Table 1: $\hat{\boldsymbol{\beta}}$ for the different subway lines in New York City (scaled so the max is 10) using stepwise model selection and solving a constrained convex problem when we regress $\mathbf{y}_1$ the proportion of exits from each station onto $\mathbf{X} = [(\mathbf{y}\mathbf{A}_1)', \cdots, (\mathbf{y}\mathbf{A}_{22})']$ where $\mathbf{y}$ is the proportion of entries into each subway station and the $\mathbf{A}_i$ are the different lines. Larger coefficients mean that the model predicts that those lines had more passengers.

where two vertices are connected if they are on the same line (but there are no self loops). That is, there are 22 clique networks which overlap in different ways, and they are scaled so that the $\sum_v w_{i(u,v)} = 1$ if u is on subway line i and 0 otherwise (passengers only exit at one stop). These adjacency matrices $\mathbf{A}_1, \ldots, \mathbf{A}_{22}$ each have a clique structure and it is easy to take advantage of that to avoid explicitly constructing them. Let $\mathbf{y}$ be the proportion of entries into each subway station during the week of observations, and $\mathbf{y}_1$ be the proportion of exits from each subway station. Now regress $\mathbf{y}_1$ on $\mathbf{X} = [(\mathbf{y}\mathbf{A}_1)', \ldots, (\mathbf{y}\mathbf{A}_{22})']$ to calculate $\hat{\boldsymbol{\beta}}$.

In this regression, some of these coefficients have negative values which is not very interpretable. One way to address this is to use basic convex optimization techniques (using `cvx` in Matlab) to force the estimates to be non-negative. Alternatively, note that the subway lines which have negative coefficients also have very high p-values in the linear model. Using the `step` function in `R` to do forward-backward stepwise model selection produces a model in which the coefficients are all non-negative. Table 1 includes results from both the stepwise model selection and the convex optimization problem, standardized so the largest weight is 10. The coefficient of determination $R^2 = 1 - (\text{SSE}/\text{SST})$ for the stepwise linear regression is 0.6216, and for the constrained optimization problem it is 0.6188. These methods very closely agree on the $\hat{\boldsymbol{\beta}}$ and suggest that a considerable amount of the variance is explained with this naive model of passengers taking a single subway line. The interpretation of the coefficients in table 1 is that (assuming this model) the '7' line and the 'L' line had the most passengers during the week of measurements, the '1', '6', and 'Q' also had a large number of passengers, and the 'M', '9', and 'G' had the fewest passengers.

This model does not assume that $\hat{\mathbf{M}}$ is row stochastic, thus clearly omitting essential information. One quick fix to this is simply to standardize the rows of $\hat{\mathbf{M}}$ to sum to one giving $\hat{\mathbf{W}}$ (this is possible in the convex optimization problem because the minimum row sum in $\hat{\mathbf{M}}$ is strictly greater than 0, this is not the case in the stepwise model selection). Setting $\hat{\mathbf{y}}$ to the principal eigenvector of $\hat{\mathbf{W}}$ gives a coefficient of determination equal to 0.6350.

5 Logistic Regression

5.1 Theory

For logistic regression, there are n vertices connected through k different networks in which vertex u is able to infect vertex v if they are connected through some network. At some initial time point we observe $\mathbf{y} \in \{0,1\}^n$ indicating if a vertex is infected. Define $\mathbf{z}' = (\mathbf{1}' + \exp(-((\mathbf{y}\sum_{i=1}^k \beta_i \mathbf{A}_i)' + \tilde{\mathbf{X}}\boldsymbol{\gamma})))^{-1}$ where the exp and the inverse are both taken element wise on the column vector. We also observe $\mathbf{y}_1 \in \{0,1\}^{1\times n}$ in which the jth element takes the value 1 with probability equal to the jth element in $\mathbf{z}$. To estimate the parameters $\boldsymbol{\beta}, \boldsymbol{\gamma}$ above, we construct $\mathbf{X} = [(\mathbf{y}\mathbf{A}_1)', \ldots, (\mathbf{y}\mathbf{A}_k)', \tilde{\mathbf{X}}]$, rewrite the $\mathbf{z}$ as

$$\mathbf{z}' = \left(\mathbf{1}' + \exp\left(-\mathbf{X}\begin{bmatrix}\boldsymbol{\beta}\\ \boldsymbol{\gamma}\end{bmatrix}\right)\right)^{-1} \tag{4}$$

and now this is simplified into logistic regression of $\mathbf{y}_1'$ onto $\mathbf{X}$.

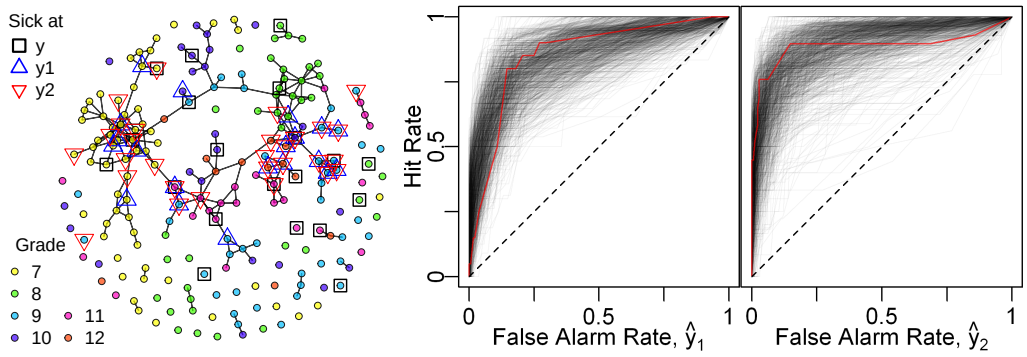

Figure 2: (Left) Single instance of an epidemic on the faux-mesa-high network. 20 initial vertices are selected uniformly at random to be sick (black squares), and the disease propagates to the first time step (blue triangles) and then to the second time step (red triangles). The first time step is used as a training set from which parameters are estimated, the estimates of those parameters are then used to estimate which vertices will get sick at the following time point. (Right) This process is repeated 1,000 times, and the ROC curves are drawn (the red curve is from the sample shown to the left) for $\hat{\mathbf{y}}_1$ predicting $\mathbf{y}_1$ (training) and for $\hat{\mathbf{y}}_2$ predicting $\mathbf{y}_2$ (test).

5.2 Simulation

The dataset `faux.mesa.high` from the package `ergm` in `R` is a simulation of a social network of a high school ($n = 205$, see the documentation in [15] for more details). Each vertex represents a student and we observe the friendship network of the students along with the grade that they are in (see figure 2 (left)). This gives two networks: $\mathbf{A}_1$ is a friendship network, and $\mathbf{A}_2$ is a block diagonal matrix network where everyone in the same grade is in a clique. We also included an external intercept $\mathbf{x}_3$ (corresponding to a propensity to become sick) and an indicator term for if the student was sick at the previous time point $\mathbf{x}_4$. 20 students are selected (uniformly at random) to be sick initially (for $\mathbf{y}$). We arbitrarily fixed the true parameters at $\boldsymbol{\beta} = [2, 0.2, -4.5, 2.5]'$, and calculated $\mathbf{z}$ as in section 5.1. The vector $\mathbf{y}_1$ is then sampled to be sick with probability $\mathbf{z}$ (element wise). We repeated this process to construct $\mathbf{z}_1$ from $\mathbf{y}_1$ and then sampled $\mathbf{y}_2$ according to $\mathbf{z}_1$. The goal is to try to predict $\mathbf{y}_2$ from the parameters estimated in constructing $\hat{\mathbf{y}}_1$. In figure 2 (left) we see one such instance of this, and (right) we see 1,000 ROC curves in predicting $\mathbf{y}_1$ (training) and $\mathbf{y}_2$ (test). The average area under the curve of the ROC for $\hat{\mathbf{y}}_1$ is 0.851 and for $\hat{\mathbf{y}}_2$ it is 0.890. Although it is uncommon to fit test data better than training data, on average there are 18.6 sick vertices in $\mathbf{y}_1$ and 26.2 in $\mathbf{y}_2$ making it easier to identify students who are most at risk to get sick (it is likely an artifact of selecting the 20 students uniformly at random for $\mathbf{y}$).

5.3 Real Data Example: H1N1 Flu Spreading Through Counties in the USA, May 2009

In May 2009, the H1N1 flu pandemic was sweeping through the USA. Data regarding which counties have had a reported case of the flu is freely available at http://flutracker2.rhizalabs.com. In the `R` package `maps` there is a map of the counties and in the `R` package `spam` there is an adjacency matrix of the counties in the USA corresponding to whether the counties share a border. Let $\mathbf{y}$ be the counties which have at least one case of the flu on May 16, 2009, similarly for $\mathbf{y}_1$ on May 23 (training set) and $\mathbf{y}_2$ on May 30 (test set). Let $\mathbf{A}$ be the adjacency matrix on counties in the USA. Let $\mathbf{X} = [\mathbf{1}, (\mathbf{y}\mathbf{A})', ((\mathbf{y}\mathbf{A})\mathbf{A})', \mathbf{y}']$ which corresponds to an external intercept term, one step on the county adjacency matrix, two steps on the county adjacency matrix, and an indicator if the counties were previously sick.

Logistic network regression of $\mathbf{y}_1$ onto the matrix $\mathbf{X}$ provides parameter estimates of $[-4.096, 0.2056, 0.1073, 7.933]'$. Note that we should expect the last parameter to be ∞ since if there was a confirmed case at time $\mathbf{y}$ then there should also be at least one confirmed case at the later time in $\mathbf{y}_1$, but there were two counties that redacted this confirmation. These parameters and $\mathbf{X}_1 = [\mathbf{1}, (\mathbf{y}_1\mathbf{A})', ((\mathbf{y}_1\mathbf{A})\mathbf{A})', \mathbf{y}_1']$ are used to estimate which counties would be sick in $\mathbf{y}_2$. Despite how naive this model is (that the H1N1 flu only travels to nearby counties) it proved to

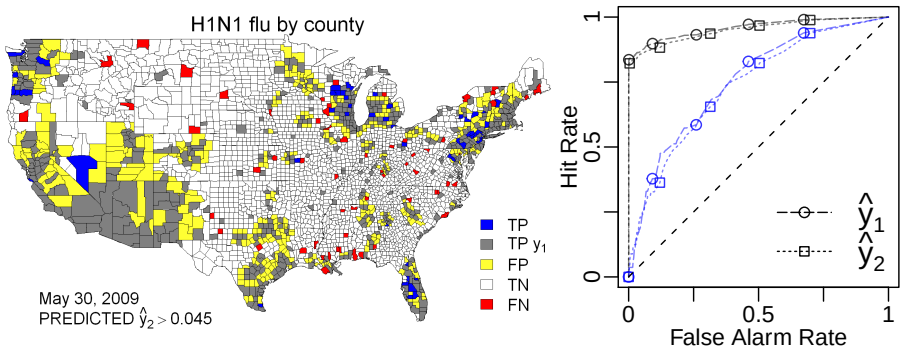

Figure 3: (Left) Predicting H1N1 flu in US counties with a threshold of $\hat{\mathbf{y}}_2 > 0.045$. Blue is correctly predicting a case of H1N1 in a previously uninfected county, gray means that we already knew there was a case there, yellow is incorrectly predicting H1N1 would spread there, white is correctly predicting H1N1 would not spread there, red is failing to predict that H1N1 would spread there. (Right) ROC curves for predicting H1N1 for $\mathbf{y}_1$ (training - long dash) and $\mathbf{y}_2$ (test - short dash). The gray curves indicate the ROC for predicting all counties, and the blue curves indicate the ROC for predicting just the counties which were previously uninfected. The points on the curves correspond (from left to right) to a classification threshold value of $0.5, 0.1, 0.05, 0.03, 0.02, 0.01$.

work very well. In predicting $\mathbf{y}_1$ (training) the area under the ROC curve is 0.9590 for all counties and 0.7530 for previously uninfected counties. In predicting $\mathbf{y}_2$ (test), the area under the curve is 0.9513 for all counties and 0.7364 for previously uninfected counties (see figure 3). The p-value according to a Mann-Whitney U test statistic comparing this to random guessing (from the package `verification` in `R`) for predicting the previously uninfected counties in $\mathbf{y}_2$ is $8.88 * 10^{-16}$, and the p-value for the other three ROC curves is $< 10^{-16}$.

6 Conclusions and Open Questions

The primary insight from this research is that it is possible to use well understood regression or supervised learning techniques on network data by taking a step on each network as an explanatory variable and fitting the response $\mathbf{y}_1'$ to $\mathbf{X}$. Although this paper only discusses how to use this model in linear and logistic regression, it should be clear how this extends widely to other regression models such as generalized linear models or other supervised learning techniques.

Network regression could be applied to various fields such as web traffic, epidemiology, social/political influence, citation network analysis, protein interactions, and product sales/reviews. In many application only the highest valued vertices in $\hat{\mathbf{y}}$ are of important. It may be fruitful to try to improve on the estimates of the extreme values of $\hat{\mathbf{y}}$ at the possible expense of lower values in that vector. In practice, it may be unreasonable to assume that the $\mathbf{A}_i$ are known exactly. One way to approach this problem in linear regression (when the explanatory variable has an error term) is to us principal component regression, so looking in the same direction for network regression may be fruitful. From a network structure prediction perspective, it may be possible to combine network regression with supervised learning techniques that can be used to predict or recommend links in networks (see [2, 18]). The voting theoretic axioms that PageRank satisfies (isomorphism, self edge, vote by committee, collapsing, and proxy) are examined in [1]. Continuing this game theoretic research by looking at propagation and walks on networks like in the network regression model from an axiomatic perspective could provide further insight about network regression models. Although there are many clear connections between network regression and PageRank, it remains to look more carefully at the relationship between Hubs and Authorities and network regression which involves bipartite networks. It would be interesting to relate $\hat{\mathbf{M}}$ to kernels on graphs (which measure similarities between vertices in a graph, see[20, 26] for some intuition on constructing graph kernels).

References

[1] A. Altman and M. Tennenholtz. Ranking systems: the pagerank axioms. In *Proceedings of the 6th ACM conference on electronic commerce*, pages 1–8. ACM, 2005.

[2] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 635–644. ACM, 2011.

[3] A. Bavelas. Communication patterns in task-oriented groups. *Journal of the Acoustical Society of America*, 1950.

[4] S. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, 2005.

[5] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge Univ Pr, 2004.

[6] F. Brauer. Compartmental models in epidemiology. *Mathematical epidemiology*, pages 19–79, 2008.

[7] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[8] V. Capasso. *Mathematical structures of epidemic systems*. Springer Verlag, 1993.

[9] C. Ding, H. Zha, X. He, P. Husbands, and H. Simon. Link analysis: hubs and authorities on the world wide web. *SIAM review*, 46(2):256–268, 2004.

[10] M. Everett and S. Borgatti. Ego network betweenness. *Social Networks*, 27(1):31–38, 2005.

[11] F. Fouss, M. Saerens, and J.-M. Renders. Links between Kleinberg's hubs and authorities, correspondence analysis, and markov chains. *Data Mining, IEEE International Conference*, pages 521–534, 2003.

[12] T. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE transactions on knowledge and data engineering*, pages 784–796, 2003.

[13] T. Haveliwala, S. Kamvar, and G. Jeh. An analytical comparison of approaches to personalizing pagerank. Technical report, Stanford University, 2003.

[14] J. Hoffmann. *Generalized linear models*. Allyn & Bacon, 2003.

[15] D. Hunter, M. Handcock, C. Butts, S. Goodreau, and M. Morris. ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software*, 24(3), 2008.

[16] M. Jackson. *Social and economic networks*. Princeton Univ Pr, 2008.

[17] G. Jeh and J. Widom. Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*, pages 271–279. ACM, 2003.

[18] M. Kim and J. Leskovec. The network completion problem: Inferring missing nodes and edges in networks. In *SIAM International Conference on Data Mining (SDM)*, 2011.

[19] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[20] R. Kondor and J.-P. Vert. *Diffusion Kernels*, chapter 8, pages 171–192. The MIT press, 2004.

[21] P. McCullagh and J. Nelder. *Generalized linear models*. Chapman & Hall/CRC, 1989.

[22] J. Nelder and R. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384, 1972.

[23] M. Newman. A measure of betweenness centrality based on random walks. *Social networks*, 27(1):39–54, 2005.

[24] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.

[25] G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966.

[26] A. Smola and R. Kondor. Kernels and regularization on graphs. In *Learning theory and Kernel machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003: proceedings*, page 144. Springer Verlag, 2003.

[27] S. Wasserman and J. Galaskiewicz. *Advances in social network analysis: Research in the social and behavioral sciences*. Sage Publications, Inc, 1994.