Path Weighted Regression:

A Statistical Model to Describe Dependency in Large Networks

Ernesto Calvo[*], Joan C. Timoneda[†], Tiago Ventura[‡]

Abstract

Dependency in large networks is among the most intractable challenges in social network analysis, as current approaches are computationally infeasible in networks with thousands of nodes. In this article, we introduce path-weighted regression (PWR), a novel and computationally efficient method to assess spatial dependency and non-stationarity in large networks. PWR estimates separate models for each node in the network and weighs nearby nodes more heavily than distant nodes. This approach allows us to estimate local effects in a large network efficiently. We illustrate PWR using a large Twitter network with 280,000 nodes showing the reaction by Republicans and Democrats to Trump's Covid-19 diagnosis

1 Introduction

In less than a decade, we entered in an era of massive data sets reporting inter-connected observations. Social media data provides one of the most visible examples, but it represents just the tip of the data iceberg. Wireless phone networks (Chung et al., 2007), live transit data (Pinelli et al., 2016; Verma and Bhatia, 2013), e-cities and e-government data (Grujic et al., 2014), citation networks (Radicchi et al., 2012; Ding et al., 2009), legislative records (Poole, 2005), scientific mentoring dyads (Keller and Blakeslee, 2014), are all part of vast, clean, and accessible sources of information currently at our disposal. As it was noted by Henry E. Brady recently: "Social scientists must come to grips with the current dramatic transformations in the communication of information, which parallel the striking changes in transportation in the nineteenth century" (Brady, 2019, pp. 299).

[*]iLCSS, GVPT, University of Maryland.
[†]iLCSS, Purdue University.
[‡]iLCSS, GVPT, University of Maryland.

Dependency in large networks is among the most difficult challenges currently faced by scholars trained in social network analysis, a field that developed most of its techniques to describe small and sparsely featured datasets (Freeman, 2004; Ward et al., 2011). As is the case for their smaller counterparts, observations in large networks are not independent and identically distributed draws from a population. Paraphrasing Tobler's first law of geography, everything is related to everything else, but connected things are more related to each other than unconnected ones. As we move from small and shallow relational data sets to large and fully featured ones, storing and representation, processing time, parallelization, data dependency, model specification, and model estimation, need to be considered together. In this article, we introduce a fast and simple Path Weighted Regression (PWR) strategy to model data dependency in large networks. The proposed statistical function allows researchers to model network dependency, giving more weight to observations that are closely connected and less weight to distant ones.[1]

Path Weighted Regression has a number of advantages for analyzing very large networks. First, different from most existing social network's statistical models, estimation can be divided into discrete processing batches, making parallelization easy. Second, computational demands increase linearly with network size, rather than exponentially. Therefore, as network size increases estimation does not become exponentially slower. Third, the weighting scheme used by PWR can be easily extended to any General Linear Model. Fourth, the Path Weighted Regression produces local estimates that can be used to smooth univariate distributions in large social networks, either for visualization purposes or as an intermediate product for two-stage error correction models.

Path Weighted Regression also has some important limitations. First, just as its cousin, Geographically Weighted Regression, local inferences lack statistical properties for hypothesis testing. Second, because distances in networks tend to have lower dispersion than in geography,

[1]The PWR is a close relative to other local linear estimation alternatives used to model dependency in geographic models Fotheringham et al. (2002); Lloyd (2011); Anselin and Bera (1998) and in survival models (Cai and Sun, 2003).

discrimination is also lower. Third, the PWR cannot easily accommodate complex network structures, such as triangles. While strong and weak ties can be calibrated, triadic closure is not evaluated.

Despite these limitations, PWR does provide a simple and fast alternative for local estimation in very large networks. Therefore, we think it provides a valuable tool for practitioners and researchers. In this short research note, we describe the PWR model and its R implementation.

2 An introduction to Path Weighted Regression models

When working with very large networks, many of the existing tools to model dependency are computationally unfeasible. In a recent article, Schmid and Desmarais (2017) note that ERGM models are computationally prohibitive once they reach 1,000 nodes, a modest network size by today's standards. As networks increase in size, they argue, Maximum Pseudolikelihood Estimation (MPLE) may provide the only computational alternative in real applications. Even in MPLE, however, the estimation of network structures, such as triangles, stars or geodesic forms, becomes computationally unfeasible for medium sized networks of thousands of nodes. PWR is a useful alternative for political scientists that seek to model spatial dependency and non-stationarity in very large networks efficiently. Estimation takes into consideration the path-distance to nearby nodes as well as local network heterogeneity. The PWR resembles two other widely used modelling approaches. First is Geographically Weighted Regression (GWR) (Fotheringham et al., 2002; Lloyd, 2011; Darmofal, 2008), a technique designed to model relationships that vary in space by weighting more heavily observations that are geographically closer. GWR constructs separate equations for each feature of the dataset and estimates neighborhood effects. As the PWR, the GWR is especially useful for large datasets. We draw from the logic of the GWR model to build the PWR approach in two important ways. We can approximate local estimates better by weighing more heavily the characteristics of the neighbors, and we can achieve this by running separate and independent regressions at the node level with a weight matrix.

The model takes as input the distance matrix of a network and gives more weight to those nodes that are connected by shorter paths. In doing so, the PWR strategy maps the local effects of unobserved factors across closely connected nodes.[2] Second, the PWR also bears similarity to LOESS models (Jacoby, 2000; Keele, 2008), a family of non-parametric approaches that use local weighted regression to reveal local non-linear trends in the data that parametric models may miss. Local estimates from LOESS weigh immediate neighbors more heavily through a vector of weights that also varies by node, as in the PWR. While the LOESS is useful to detect descriptive trends in the data that may be difficult to find through parametric models, it lacks properties for hypothesis testing and causal inference. This also applies to the PWR, which is especially helpful in understanding local differences in a network and provides a great new tool to assess dependency, primarily in large networks. It is not useful, however, for hypothesis testing.

2.1 The path weighted local regression

Consider a simple linear model on data drawn from a fully connected network, where the dependent (node) variable y_i is explained by a set of observed covariates x_N and unobserved parameters β_N:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + ... + \beta_n x_{ni} + \varepsilon_i \tag{1}$$

In OLS, we minimize the sum of square residuals and solve for the $\hat{\beta}$ parameters, so that:

$$MSE(\beta) = \frac{1}{N} \sum_{i=1}^{N} (y_i - x_i \beta)^2 \tag{2}$$

$$\hat{\beta} = (X'X)^{-1} X'y \tag{3}$$

[2] We direct the reader to some applications of GWR in the literature here (Darmofal, 2008; Cho and Gimpel, 2009; Calvo and Escolar, 2003).

For any given node, we consider that observations that are more closely connected (lower path distances) weigh more heavily than observations that are further away. Therefore, at each node in the network we estimate separate weighted linear regression models:

$$wMSE(\beta, w_1, ..., w_N) = \frac{1}{N} \sum_{i=1}^{N} w_i \left(y_i - x_i \beta\right)^2 \tag{4}$$

And solve for the local parameters:

$$\hat{\beta} = \left(X'WX\right)^{-1} X'Wy \tag{5}$$

PWR estimates separate models for each node (or for a sample of nodes) in the network. Observations that are further removed from each node weigh less than observations that are more closely connected. Therefore, all PWR estimates are local and the output of the model returns parameters that vary for each covariate at each node i connected to every other node j.

Model results provide a full distribution of $\hat{\beta}$ parameters for all nodes in the network, which can then be post-processed and visualized. Heterogeneous effects of the independent variables by node location, consequently, allow researchers to understand differences in content propagation at different regions of the network.

In contrast with GWR in spatial models, path distances in networks are shorter, with a relatively small set of discrete values that connect all nodes (e.g. small world network). Therefore, bandwidths across different networks will be similar to each other and the bandwidths narrower than in GWR. For our implementation of the PWR, we consider the minimum number of paths connecting each pair of nodes, creating a distance weights matrix. In dense areas of a network, therefore, there will be a larger sample of minimum paths connecting all nodes. By contrast, in sparsely connected regions of a network the opposite holds.

More important, shorter paths in densely packed communities will display lower variation across nodes as information will travel faster. Meanwhile, longer distances across communities

will result in higher parameter discrimination. That is, more dense networks will reduce local effects, as the distance across all pairs of nodes will be smaller. Meanwhile, local effects will be more distinct in sparse networks, where path distances across nodes are on average larger.

Similar to GWR, building the weights matrix requires that researchers decide the relative contribution of nodes through a decay function. Several options are available, with the Gaussian weighting function being the most common choice in the the existing literature (Fotheringham et al., 2002; Darmofal, 2015). The Gaussian discount function takes the form:

$$w_{ij} = exp\left(\frac{-I}{2\frac{P_{ij}}{b}^2}\right)$$

In the previous equation, P_{ij} describes the minimum number of paths connecting node i to node j, and b is the bandwidth for the path-decay of the weighting function. A larger bandwidth results in estimates that are more distinctively local. By contrast, a smaller bandwidth produces estimates that are roughly similar across nodes. As in other local polynomial models, there is a trade-off between bias and variance in the choice of the PWR bandwith. To approximate an optimal bandwidth, we employ leave-one-out cross validation selection, a widely used data-driven strategy (Fotheringham et al., 2002):

$$CV = \sum_{i=1}^{N}(y_i - \hat{y}_{i \neq i})^2$$

The cross validation procedure uses a leave-one-out process in which each local estimation of the observation i, where the local parameter is centered, receives a weight equal to zero. Then, the score takes the square difference of the y_i and the prediction of the model for the observation where the weight was set to zero. We provide an analytical solution to find the point where the optimal value of the CV score is minimized after performing a grid search from the default values of our package ranging from a exponential function from 0 to 3 using intervals of .5. We suggest taking a sample of different points when dealing with big network data. The same procedure

is suggested in applications of the Geographically Weighted Regression (Fotheringham et al., 2002).

One of the advantages of the path weighted regression model is that it can be easily parallelized. We run separate regressions for each node, weighing more heavily those vertices that are closer to each other. The model can independently fit as many linear regressions as there are nodes in the network. Since calculations are computed using the same bandwidth on a steady weights' matrix, regressions can be run in parallel rather than sequentially, which translates into faster processing times.[3] Additionally, the weight matrix in a PWR model does not increase exponentially and, as a result, computational demands remain manageable even as network size increases. Thus, PWR is a fast and efficient modelling strategy for large networks.

3 Application: Analysis of Twitter Networks After Trump Contracted the Coronavirus

To provide an example of the usefulness of PWR, we take as dependent variable the reaction time on twitter when President Donald Trump Jr. posted news of his COVID-19 diagnostic. We then model trolling behavior in social media, describing the change in the time-to-retweet when the posted tweet includes the emoji "face-with-tears-of-joy". In this section we first describe the social activity in Twitter when Trump announced his COVID-19 diagnostics. We then provide PWR model estimates with node specific parameters describing trolling behavior.

At 12:54am on October 2, 2020, Trump published a tweet announcing that he had tested positive for COVID-19. This tweet would become one of his most viral messages on the platform, gathering 1.8 million likes and 400,000 retweets. Figure 1 provides descriptive information on network activity that leverages pre- and post-announcement activity related to Trump on Twitter.

[3]The extent of these savings will depend on the number of cores available and the clock speed of each core, virtual or real. Gains are particularly important for large networks, which are more computationally demanding.

The news that President Trump contracted COVID-19 yielded well-known activity of shared attention that is described by Lin et al. (2014) in their study of "rising tides and rising stars". As expected, the data shows a large inflow of new messages by a broader periphery of users ("rising tides") that is accompanied by more hierarchical sharing of publications posted by a few nodes ("rising stars").

As part of the election coverage effort of the iLCSS, we captured streaming data from Twitter's API V.1 (forward capture), collecting all tweets with the character string *"Trump"* posted during the 12-hour period around the 12:54am announcement –that is, between 6:54pm on October 1 and 6:54am on October 2. We then filter out unique tweets that were not shared and implemented a community detection algorithm to retain the primary connected network engaged on the COVID-19 topic. The process yielded a total of 3,994,860 tweets by a total of 280,041 high activity unique users.

We first provide a visual representation of the #TrumpCovid network in 1. The vertical axis describes binned shared times, time-to-retweet(LN), for batches of tweets shared during the 12 hour window of the study. The horizontal axis describes the time of each set of binned retweets. As it is possible to oberve, there is significant twitter related to Trump before and after the announcement. However, activity increases and time-to-retweet declines (lower reaction time) after the announcement of the COVID-19 diagnostics. Figure 1 shows that at the time of the diagnostics, time-to-retweet decline 78%, $exp(7)/exp(8.5) = 0.223$ among democratic users and 40%, $exp(8.3)/exp(9.2) = 0.406$, among Republican users. Attention to the event was larger among Democratic users, who were more engaged and displayed faster reaction times. To facilitate visual interpretation of the results, Figure 1 overlays on the upper-left side the [x,y] network coordinates for the Democratic and Republican users,[4] with blue and red solid circles to describe Democratic and Republican users identified by the walk.trap algorithm in *igraph*

[4]Estimation of the layout was implemented using the Fruchterman-Reingold layout function of igraph on the primary connected cluster of the network.

1.0.[5]

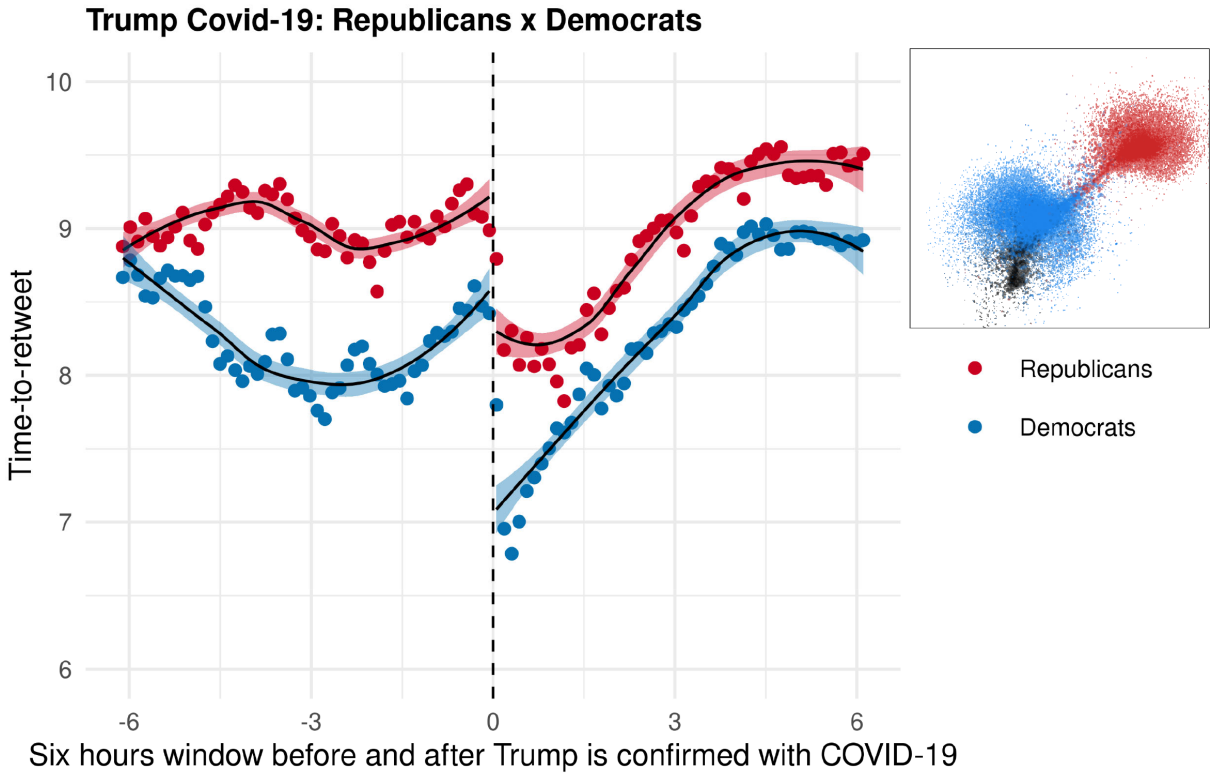

Figure 1: Time-to-retweet (LN) in Primary Connected Network of #TrumpCovid. Layout of the descriptive network overlayed in the upper right. Total number of users (nodes) is 280,041, with layout and community detection from 3,994,860 retweets. Blue dots describe Democratic users. Red dots describe Republican users. Layout of nodes estimated using the Fruchterman-Reingold algorithm in *IGraph*. Community detection using the Walk.trap algorithm in *IGraph*, (Csardi and Nepusz, 2006).

Modeling Trolling: Time-to-retweet and "face-with-tears-of-joy" in Twitter

To describe trolling in our network, we estimate a PWR specification with reaction time as our dependent variable and the emoji "face-with-tears-of-joy" as a covariate. Results describe heterogeneous effects on time-to-retweet within communities as well as the change in time-to-retweet when the the selected emoji was inserted in a tweet.

The emoji "face-with-tears-of-joy" was named word of the year (WOTY) in 2016 by the

[5] The walk.trap algorithm (Pons and Latapy, 2006) retrieved an index community value for each node (Csardi and Nepusz, 2006), sorting network nodes into three communities that we identified by their highest in-degree users as Republicans (146,224 users – red), Democrats (102,671 users –blue) and independents or unaffiliated accounts (31,146 users –maroon).

Oxford Dictionary (Skiba, 2016) and has been frequently analyzed as an efficient emotional enhancer (Gullberg, 2016). In its two flavors (straight and slanted face), it is frequently used as a teaser, with positive and negative interpretations when it celebrates novel occurrences or, alternatively, laughs at the other user's misfortune. This second alternative is considered a form of malicious joy (*Schadenfreude*). It is frequently used in political networks as a trolling emoji and can be readily extracted as useful information. We exemplify a model specification that uses "face-with-tears-of-joy", with parameters that summarize within- and across-community heterogeneity in its use and acceptance.

Estimation of PWR

We first retrieve an optimal bandwidth for the weight matrix of path distances. To this end, we select a 10% random samples of nodes, using the mean of these distributions to estimate the full cross-validation model. Figure 2 plots the sampling distribution of the bandwidth, which return an optimal bandwith of 1.2 to be used on the complete dataset. It is worth noting that smaller bandwidths will increase local effects while larger bandwidths will provide local estimates that are closer to the overall network mean. Finally, as suggested in the previous section, we use parallelization for faster model estimation. Generally, the PWR can generate local estimates for large networks between 10 and 15 times faster than other available approaches.[6]

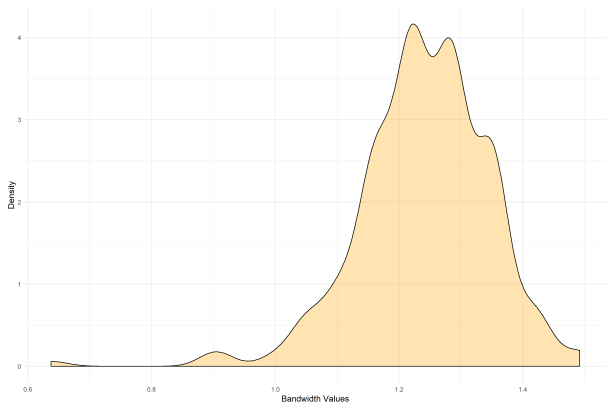

Figure 2: Sampling distribution of the bandwidth selection procedure (10% sample of nodes)

[6]With 12 real cores and 128GB of RAM, the total processing time for the 280,041 node network was 2 hours and 35 minutes, as compared with over 24 hours with a fast single-core processor.

Figure 3 displays the distribution of local β coefficients for the PWR model. We color the nodes by quintile to facilitate visualization. The size of the nodes describes the time to retweet. Plot (a) shows local intercepts and Plot (b) depicts the slopes of the model for each node.

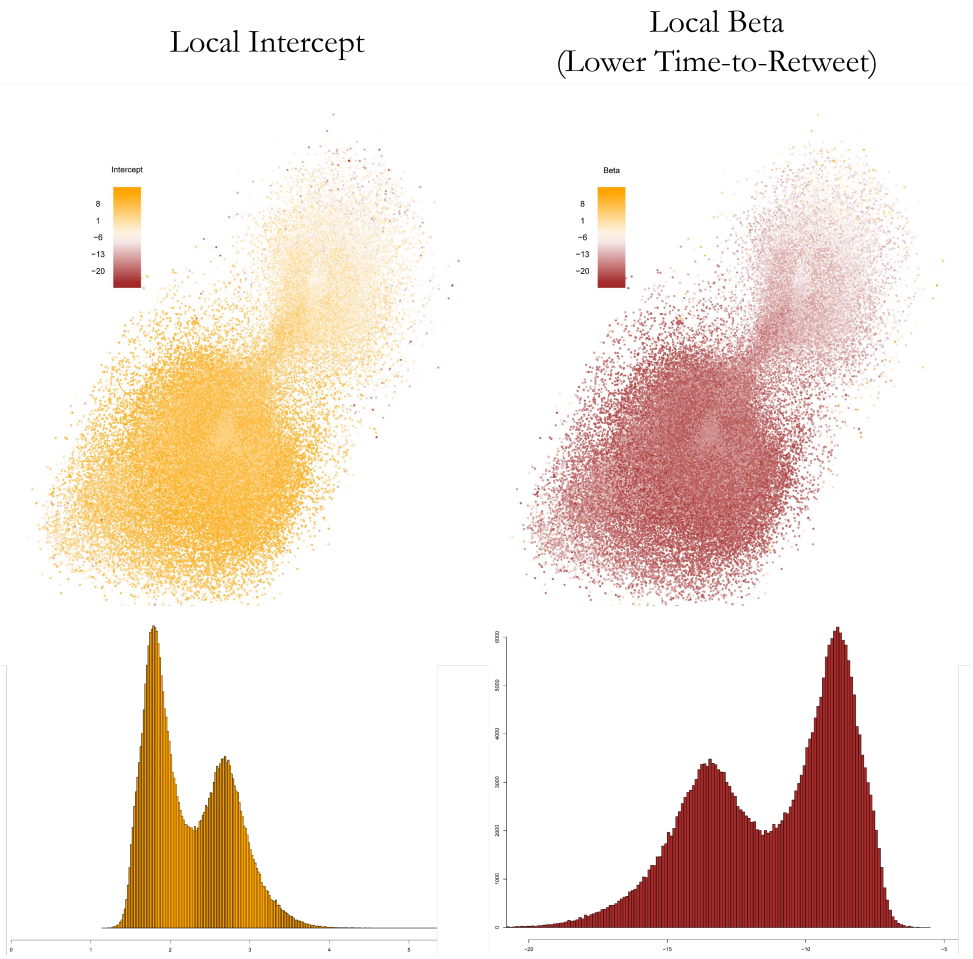

Figure 3: PWR Results in the network format. The model runs locally the mean reaction time (inverse of the time-to-retweet) by node as well as the effect of the the emoji "face-with-tears-of-joy". Results show significant heterogeneity in the decline in reaction time among Democratic users, indicating that the presence of the emoji produced higher within community discrimination. That is, reaction time to a retweet with "face-with-tears-of-joy" increase the differences among Democratic users.

The intercept of the model described in the upper plot of 3 shows that the average time to retweet by users that are located at the center of each community is slower than the average times of the second periphery. More interesting, the slope estimates displayed in plot (b) show that the effect of in-degree on time to retweet is larger at the center of each community. That is, as in-degree increases, the decline in time to retweet is considerably larger for core nodes in the pro- and anti-Kavanaugh networks than in their peripheries. By contrast, nodes in the periphery

of each community are quick to propagate content even when their in-degree increases. The lack of effect of in-degree on content propagation on the second periphery of both communities lends considerable support for the presence of bots and computer managed accounts.

Interpretation of the results is straightforward, with more negative slope coefficients describing slower time to retweet as the in-degree increases. The fact that authorities (i.e. users with higher in-degree) would take a longer time to retweet information is expected for two reasons. First, high in-degree authorities will be more risk averse in sharing content that may reflect poorly on them. Second, bots, trolls, and other managed response systems tend to have smaller numbers of followers and are set up to quickly share content from priority accounts. The heterogeneous effects of in-degree on content propagation are large and substantively interesting. As expected, the model shows that nodes will take longer to share messages from other users as in-degree increases. Users at the center of each community have a stronger decay function while, for loosely connected nodes on the periphery, being an authority matters less for quickly propagating news.

One caveat is important here. As the size of the nodes on figure 3 indicate, authorities tend to be in the center of each community. Therefore, the average in-degree of the nodes is higher in the more populated areas of the network. With that feature in mind, the PWR model allows us to assume that the effects of popularity exhibit increasing returns on the rate of content activation. One additional retweet on users' on the periphery of the network who rarely receive attention have a negligible effect on one's speed of content's activation; nonetheless, the same increase in popularity has more substantial adverse effects in the speed of propagation on authorities located at the center of the network.

4 Concluding Remarks

As data availability increases, finding novel strategies to model big relational data has become a more pressing issue. In this article, we introduce readers to a fast and computationally simple

strategy to describe dependency in large networks. In the last decade, statistical advances to study small and relatively homogeneous social networks has been remarkable. However, powerful new techniques such as exponential random graph models become technically challenging or altogether unfeasible as network size increases. PWR provides an alternative tool for researchers that seek to take advantage of the information in contiguous nodes of a large network. It provides a computationally feasible alternative for large heterogeneous networks.

We exemplify the usefulness of the PWR in one application: an analysis of the reaction time after President Donald Trump discloses a positive COVID-19 diagnostics. Our results show that the emoji "face-with-tears-of-joy" lowers the average time-to-retweet and it reduces reaction time more dramatically among democrats. More important, the results of the model show increased within-democratic discrimination in reaction time. That is, the use of the emoji reduces reaction time only among some democratic users that are further in the periphery of the democratic network. This is a substantively interesting result, showing not just that democrats reduce their "time-to-retweet" but that the use of the emoji is not universally accepted in this particular case.

References

Anselin, L. and Bera, A. K. (1998). Spatial dependence in linear regression models with an introduction to spatial econometrics. *Statistics textbooks and monographs*, 155:237–290.

Brady, H. E. (2019). The challenge of big data and data science. *Annual Review of Political Science*, 22:297–323.

Cai, Z. and Sun, Y. (2003). Local linear estimation for time-dependent coefficients in cox's regression models. *Scandinavian Journal of Statistics*, 30(1):93–111.

Calvo, E. and Escolar, M. (2003). The local voter: A geographically weighted approach to ecological inference. *American Journal of Political Science*, 47(1):189.

Cho, W. K. T. and Gimpel, J. G. (2009). Rough Terrain: Spatial Variation in Campaign Contributing and Volunteerism. *American Journal of Political Science*, 54(1):74–89.

Chung, W.-Y., Yau, C.-L., Shin, K.-S., and Myllyla, R. (2007). A cell phone based health monitoring system with self analysis processor using wireless sensor network technology. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3705–3708. IEEE.

Csardi, G. and Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal Complex Systems*, 1695:1695.

Darmofal, D. (2008). The Political Geography of the New Deal Realignment. *American Politics Research*, 36(6):934–961.

Darmofal, D. (2015). *Spatial analysis for the social sciences*. Cambridge University Press, New York, NY.

Ding, Y., Yan, E., Frazho, A., and Caverlee, J. (2009). Pagerank for ranking authors in co-

citation networks. *Journal of the American Society for Information Science and Technology*, 60(11):2229–2243.

Fotheringham, A. S., Brundson, C., and Charlton, M. (2002). *Geographically weighted regression : the analysis of spatially varying relationships*. Wiley, Chichester, West Sussex, England ; SE - xii, 269 pages : illustrations, maps ; 26 cm.

Freeman, L. (2004). The development of social network analysis. *A Study in the Sociology of Science*, 1(687):159–167.

Grujic, I., Bogdanovic-Dinic, S., and Stoimenov, L. (2014). Collecting and analyzing data from e-government facebook pages. *ICT Innovations*, pages 86–96.

Gullberg, K. (2016). Laughing face with tears of joy: A study of the production and interpretation of emojis among swedish university students.

Jacoby, W. G. (2000). Loess:: a nonparametric, graphical tool for depicting relationships between variables. *Electoral Studies*, 19(4):577–613.

Keele, L. (2008). *Semiparametric regression for the social sciences*. Wiley, Chichester, England; Hoboken, NJ.

Keller, T. E. and Blakeslee, J. E. (2014). Social networks and mentoring. *Handbook of youth mentoring*, pages 129–142.

Lin, Y.-R., Keegan, B., Margolin, D., and Lazer, D. (2014). Rising tides or rising stars?: Dynamics of shared attention on twitter during media events. *PloS one*, 9(5):e94093.

Lloyd, C. D. (2011). Local models for spatial analysis.

Pinelli, F., Nair, R., Calabrese, F., Berlingerio, M., Di Lorenzo, G., and Sbodio, M. L. (2016). Data-driven transit network design from mobile phone trajectories. *IEEE Transactions on Intelligent Transportation Systems*, 17(6):1724–1733.

Pons, P. and Latapy, M. (2006). Computing communities in large networks using random walks. *J. Graph Algorithms Appl.*, 10(2):191–218.

Poole, K. T. (2005). *Spatial models of parliamentary voting*. Cambridge University Press.

Radicchi, F., Fortunato, S., and Vespignani, A. (2012). Citation networks. *Models of science dynamics*, pages 233–257.

Schmid, C. S. and Desmarais, B. A. (2017). Exponential random graph models with big networks: Maximum pseudolikelihood estimation and the parametric bootstrap. In *Big Data (Big Data), 2017 IEEE International Conference on*, pages 116–121. IEEE.

Skiba, D. J. (2016). Face with tears of joy is word of the year: are emoji a sign of things to come in health care? *Nursing education perspectives*, 37(1):56–57.

Verma, P. and Bhatia, J. (2013). Design and development of gps-gsm based tracking system with google map based monitoring. *International Journal of Computer Science, Engineering and Applications (IJCSEA)*, 3(3):33–40.

Ward, M. D., Stovel, K., and Sacks, A. (2011). Network analysis and political science. *Annual Review of Political Science*, 14:245–264.