MSD 2019 Final Project

A replication and extension of **Predicting Positive and Negative Links in Online Social Networks** by Jure Leskovec, Daniel Huttenlocher, and John Kleinberg, WWW 2010: ACM WWW International conference on World Wide Web, 2010

Sameer Jain (sj2736), Bailey Pierson (bp2471), Tanmay Chopra (tc2897)

Contents

INTRODUCTION

This project reproduces sections of Leskovec et. al's 2010 paper Predicting Positive and Negative Links in Online Social Networks, then extends their analysis to two new data sets in order to further investigate claims of underlying social-psychological laws of signed edge formation. We first provide brief summary of the paper, then provide technical details behind the collection of data and generation of node-level model features. We combine these features following the analytic framework presented by Leskovec et. al, and provide an analysis of the divergences between these replicated results and those presented in the paper. Our primary replicated results include a comparision of predicted accuracy across domains and varying embeddedness thresholds, and a tabulation of cross-domain model generalizability. There are significant differences between replicated and published results, but no evidence which directly refutes the claims of the paper. Our extension introduces a newly collected data set– a Wikipedia adminship voting network through May 2019– and also incorporates another signed trust network derived from Bitcoin user behavior. Ultimately, these extensions suggest clear limitations to Leskovec et. al's attempts to assert that balance and status theories can reveal immutable psychological laws which underlie the creation of social networks and allow for generalization across signed trust networks.

MOTIVATIONS

Leskovec et. al (2010) deals with two fundamental tasks: **edge sign prediction** and **generalization**. Edge sign prediction is formulated as follows. Suppose that we are given a social network with signs on all of its edges, but the sign from node u to node v has been hidden. How can we reliably infer the sign of this edge given information provided by the rest of the network? The authors approach this question using a logistic regression framework which allows for the combination of a range of structural features at the node- and triad-level in order to predict a binary edge sign. The investigation of this problem across a range of data sets leads to the consideration of the generalizability of models. Generalization is measured by the sign prediction performance of a model trained on one domain and tested on another. Remarkably, the authors' find that performance only degrades slightly in cross-domain testing, which encourages them to incorporate social-psychological theories of status and balance in their discussion. Ultimately, the inclusion of these theories motivates a claim that there are underlying behavioral rules which produce cross-domain consistencies in the structure of signed trust networks.

REPRODUCTION

Data Collection

The paper under replication considers from three large online social networks – Epinions, Slashdot, and Wikipedia. As part of our extension of the study, we also supplement the provided Wikipedia data with new data extending up to May 2019 and data from a network of Bitcoin traders. All of these data sets include edges which are explicitly signed as positive or negative.

Epinions, Slashdot & Bitcoin

The edge lists for network data from Epinions, Slashdot, and Bitcoin are publicly accessible on the Stanford Network Analysis Project (SNAP). Using data pulled directly from SNAP, our Epinions data has 841,371 edges and 119,845 nodes, which are both slightly higher than the 841,000 edges and 119,217 nodes recorded for this data set in Leskovec et. al (2010). Epinions was a product review site (defunct as of 2012) which allowed users to note whether they trusted or distrusted the reviews of another user on the site. This data spans from the conception of the site in 1999 until August 2003.

The Slashdot data we have pulled from SNAP has 549,202 edges and 82,144 nodes, which corresponds directly which the data reported in the paper. Slashdot is a tech news website which allows users to tag each other as “friend” or “foe” based on an individual’s opinions on the comments another user contributes to the site. This data set spans from 2002 to February 2009

The Bitcoin dataset was also found on SNAP, and is signed network where an edge constitutes a measure of trust between users. This network was collected from the activity of people who trade using Bitcoin on a platform called Bitcoin OTC. As bitcoin users are anonymous, there is a need to record users’ reputation to prevent transactions with fraudulent users. Members of Bitcoin OTC rate other members in a scale of -10 (total distrust) to +10 (total trust) in steps of 1. In order to standardize the prediction framework across domains, this scale of trust is re coded to a binary 0 or 1 value where negative edges are coded and 0, and positive edges as 1. This dataset contains 5,881 nodes and 35,592 edges.

The collection of these data sets into a clean edge list format was straightforward. After decompressing files from SNAP, unnecessary string characters were stripped from the start of files, and non-UTF characters where then removed. The code used to clean these files is included in the `nodeFeatures.py` file.

Wikipedia

Old Wikipedia

Wikipedia is a very popular collectively authored online encyclopedia. A network data set is developed from the user application process for applying for adminship. In order to become an admin on the site, a user is either nominated by other admins, or self-nominates, then other Wikipedia users are able to cast a support, oppose, or neutral vote. While not specified in the paper, after investigation we determined that neutral edges are excluded from analysis, so a signed link in this data set indicates a positive or negative vote by a user on another user’s promotion to adminship status. The Wikipedia data associated with this paper on SNAP includes all adminship votes up to January 2008. While the paper states this data contains 103,747 edges and 7,118 nodes, the data set we pulled directly from SNAP has 110,086 edges and 7194 nodes.

The collection of this data into a clean edge list was somewhat more complex, as it’s initial structure was a *rough* approximation of an adjacency list. The code used to restructure this file is included in the `nodeFeatures.py` file.

New Wikipedia

Unfortunately, Leskovec et. al (2010) make no mention of how they extracted adminship network data from the 2008 data dump of the entirety of Wikipedia, and despite that fact they are explicitly mentioned in the paper as publicly available, no replication scripts were able to be found on SNAP or on the author's personal websites/GitHub accounts. Therefore, in order to collect new Wikipedia adminship data, a web scraper was written in order to pull out all votes on adminship requests from the conception of the adminship application process until May 2019. Exactly as specified in the paper, a edge between two users in this network is directed, and represents the support/oppose vote of one user on the promotion of another user to adminship. The script used to identify all pages associated with applications for adminship is `getPageId.py`; the script which implements a web scraper and writes voting behavior to an edge list is `getEdgelist.py`. This new Wikipedia adminship data set has a total of 11,581 nodes and 130,915 edges.

Feature Generation

After consulting with Jake, it was determined that we would reproduce the seven primary node-level features used by Leskovec et. al in their edge prediction task. These features include:

1. $d^+_{in}(u)$, the number of incoming positive edges to node u which has the directed edge $u \rightarrow v$
2. $d^-_{in}(v)$, the number of incoming negative edges to node v which has the directed edge $u \rightarrow v$
3. $d^+_{out}(u)$, the number of outgoing positive edges to node u which has the directed edge $u \rightarrow v$
4. $d^-_{out}(v)$, the number of outgoing negative edges to node v which has the directed edge $u \rightarrow v$
5. $d^+_{out}(u) + d^-_{out}(u)$, or the total out degree of u
6. $d^+_{in}(v) + d^-_{in}(v)$, or the total in degree of v
7. $C(u, v)$, the *embeddedness* of edge (u, v) or the total number of common neighbors of u and v in an un-directed sense

All of these features are computed for each data set using the scripts found in the `nodeFeatures.py` file.

Analysis and Visualization

We ultimately reproduced figures similar to the **Figure 1** and **Table 6** found in Leskovec et. al's (2010). It is important to note that these figures are *similar*, as we have reproduced 7 of the 23 data-derived features used in the paper, and none of the heuristically-derived features. However, we use the exact same "machine-learning" framework, essentially a logistic regression classifier, to combine evidence from our node-level features into an edge sign prediction. Our classifier learns a model of the form $P(+|x) = \frac{1}{1+e^{-(b_0+\sum_i^n b_i x_i)}}$ where x is a vector of features and $b_0 ... b_n$ are coefficients we estimate based on the training data. All of our results are reports of the average accuracy and estimated logistics regression coefficients over 10-fold cross validation.

Figure 1

Figure 1 shows the classification accuracy for the Epinions, Slashdot, and provided Wikipedia data sets across embeddedness thresholds of 0, 10, and 25. There are a few notable differences between our figure and the Figure 1 in Leskovec et. al (2010). First, as previously noted, we only predict using the node-level feature set. In comparison to the classification accuracy of predictions using only degree features by Leskovec et. al, our predictions are consistently more accurate. For instance, the lowest error rate obtained using degree-features in the paper is 11.45%, whereas our lowest rate is 6%. While this accuracy discrepancy between Leskovec et. al and our works holds across all data sets, it should be noted our observed that patterns in the relative accuracy across data sets and embedded thresholds are equivalent to those seen in

	Epinions	Slashdot	Wikipedia
Epinions	0.9342	0.9289	0.7722
Slashdot	0.9249	0.9351	0.7717
Wikipedia	0.9272	0.9260	0.8021

Table 1: Reported predictive accuracies when training on the "row" dataset and evaluating the prediction on the "column" dataset in the original paper.

the paper. These trends show that, for all data sets except Wikipedia, the explanatory power of our feature set increases as embeddedness does. This is anticipated as one of our node-level features, embeddedness, is obviously dependent on embeddedness and becomes more effective as a greater amount of neighboring node information becomes available. It is unclear why our classification accuracy is consistently higher than that reported by Leskovec et. al; considering this constancy, there might be a systematic difference between how our scripts determine a correct prediction. In addition, as noted in our sections on Data Collection, even the data pulled directly from SNAP is slightly different that the data sets described by Leskovec et. al. It is possible this discrepancy also contributes to the observed differences in accuracy.

Figure 1: Accuracy of predicting a sign of edge (u, v) given signs of all edges in a network

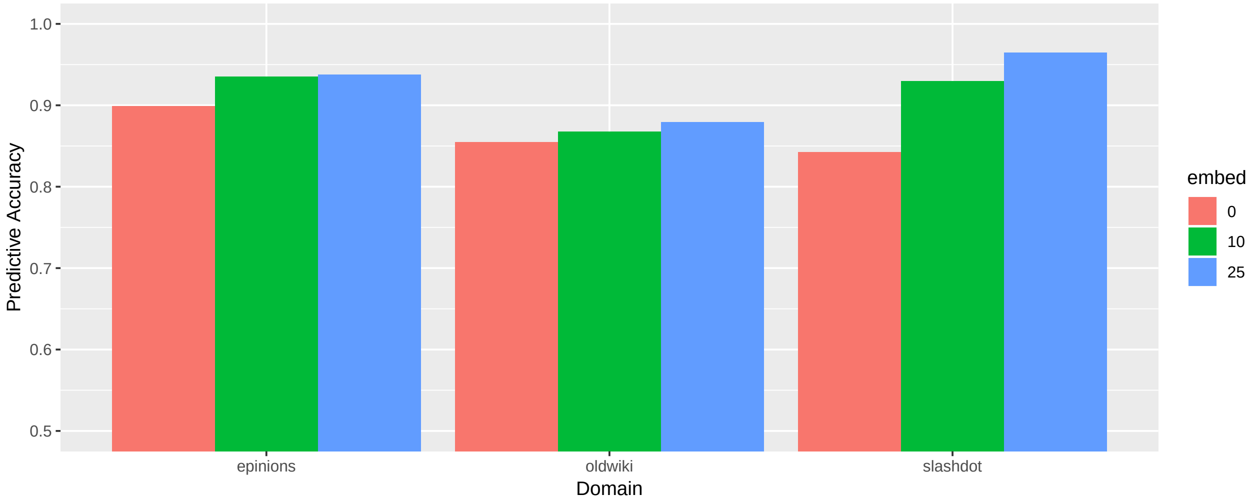


Table 6

In Leskovec et. al (2010), Table 6 shows the predictive accuracy when training on the “row” data set and evaluating on the “column” data set. For comparison purposes we reproduce this table here as **Table 1**. Again, it is important to note that these accuracies are computed from a model that uses both node and triad-level features. We only use node-level feature to reproduce a similar table, **Table 2**. There are a few important points to note here. First, the overall levels of accuracy across entire tables are remarkably high; off-diagonal entries are nearly as high as the diagonals, asserting that there is very good generalization across the three data sets. This is one of the most interesting findings of Leskovec et. al, their model must capture some underlying properties of signed social networks. Perhaps even more interestingly, we see similar (albeit consistently lower), levels of accuracy as well as patterns suggesting generalizability in our results in **Table 2**, which uses a smaller feature set. Some of the underlying properties of signed networks are clearly recorded at the node-level. However, we see larger disparities between the original results and our results with models tested on Slashdot. This suggests that triad-level features carry explanatory power for the Slashdot data set than the Epinions and Wikipedia data sets.

	Epinions	Slashdot	Wikipedia
Epinions	0.8991	0.8125	0.7956
Slashdot	0.8916	0.8424	0.7870
Wikipedia	0.9249	0.8446	0.8549

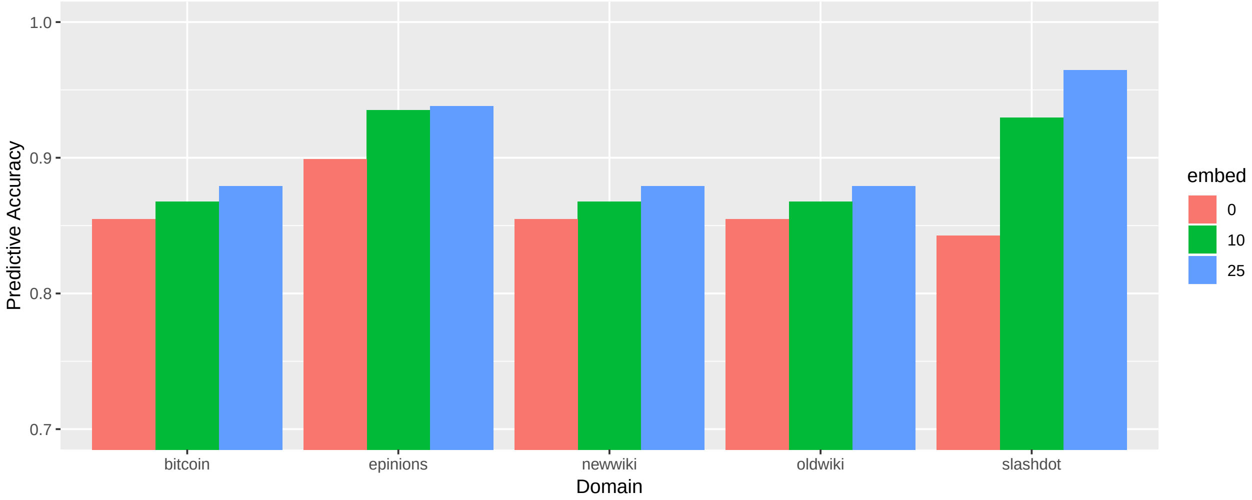
Table 2: Predictive accuracy when training on the "row" dataset and evaluating the prediction on the "column" dataset with our replicated results.

EXTENSION

New Wikipedia Data and Bitcoin Network Data

In order to extend the results of Leskovec et. al (2010), we include two additional data sets in our analysis: an updated Wikipedia adminship network and bitcoin certification network which are detailed in the Data Collection section. A reproduction of the classification accuracy bar charts with this data is included in **Figure 2**. The same patterns observed in our initial three data sets hold here. Accuracy is generally quite high, and accuracy increases as embeddedness does. We also observe that the accuracy for Slashdot is consistently lower than for other data sets; this is similar to our results in cross-training table **Table 2**.

Figure 2: Accuracy of predicting a sign of edge (u, v) given signs of all edges in a network



We also reproduce the cross-training table with these two new data sets as **Table 3**. There are a few important things to note here. First, models trained on the provided Wikipedia data and our new Wikipedia data show dissimilarity in predictive accuracy across the other data sets. There is no Wikipedia data set that consistently outperforms the other in regards to generalizability, but is clear changes in the network structure of Wikipedia adminship across time are significant. This is more readily apparent in the much higher predictive accuracy of models on the new Wikipedia data, which might suggest that recent developments in adminship behavior have produced patterns more similar to those found in the Epinions and Slashdot data.

Beyond the performance of the new Wikipedia data, it is interesting to observe that the Bitcoin network is a clear outlier in terms of generalizability– we suggest that this divergence is likely due to the fundamental differences in network structure. These differences are most apparent when we consider average values of embeddedness for each dataset. Interestingly, our data sets which have the lowest predictive accuracy, Bitcoin and Slashdot, have similar low levels of embededdness (4.6749337 and 3.9863368 respectively) in comparison to the provided and new Wikipedia data sets (0.7676364, 0.5546376) and Epinions (0.7786161). While variation in embeddedness is not the sole factor which underlies network structure, and similarity in embeddedness does not guarantee generalizability (as observed in the poor generalization of Slashdot on

	Epinions	Slashdot	Wikipedia	New Wikipedia	Bitcoin
Epinions	0.8991	0.8125	0.7956	0.8270	0.7143
Slashdot	0.8916	0.8424	0.7870	0.9105	0.5988
Wikipedia	0.9249	0.8446	0.8549	0.9046	0.5592
New Wikipedia	0.8921	0.8447	0.9378	0.9378	0.3738
Bitcoin	0.7262	0.7604	0.4826	0.4826	0.9378

Table 3: Predictive accuracy when training on the "row" dataset and evaluating the prediction on the "column" dataset with added Bitcoin dataset.

Bitcoin and vice versa), these large differences reflect inherently different network structures and suggest a clear limitation to Leskovec et. al (2010) claims assertions of overarching laws of signed edge generation.

CONCLUSIONS

We ran into a number of challenges attempting to reproduce this paper. First, although clearly mentioned by Leskovec et. al, no reproducibility code was found online. This required a us to put a substantial amount of work into generating a features for each dataset and implementing a framework for edge prediction. In addition, the incredibly vague explanation of how data from Wikipedia was collected ultimately required us to write our own web scraper to collect new data for extension. Although all of our other data sets were publicly available on SNAP and were explicitly cited in Leskovec et. al (2010), these data sets all differed slightly from their description in the paper. These challenges introduce a number of possible explanations for the slight variation between our replicated results and those found in the paper. It is possible that the prediction framework were developed differed from that used in the original paper; data uploaded on SNAP could differ significantly from that actually used by Leskovec et. al; similarly our newly collected data could have followed a different collection protocol resulting in a network that is not a direct extension of that used in the original paper. While these concerns are valid, our somewhat differing replicated results do not directly contradict those found by Leskovec et. al and suggest that many of their general conclusions based on these data sets are correct.

Our replicated results corroborate two general conclusions. First, we confirmed that a combination of seven relatively simple node-level features is able predict the sign of an edge with high accuracy. This accuracy increases alongside graph embeddedness. Second, we show that this simple model captures substantial underlying properties of signed social networks, allowing for generalizability of models across training and prediction data sets. However, findings from our extensions suggest a few caveats to this second conclusion. First, temporality is significant; the model derived from Wikipedia data including edges up to May 2019 differs substantially in generalizability from the provided Wikipedia data which only includes data up to 2008. Second, while the observed generalization across data sets motivates Leskovec et. al's discussion of social-psychological laws which could produce structures inherent to all signed trust networks, our extension suggests clear limitations to generalizability and psycho-structural interpretations. We show that high levels of generalizability do not hold for large signed social networks like the Bitcoin data. Further research which attempts to assert overarching laws of signed networked behavior must utilize a wider variety of data sources, and rigorously test for cohort effects in the behavior of users.

SYSTEM SETTINGS

The following is a list of all packages used to generate these results.

```
## R version 3.5.1 (2018-07-02)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS High Sierra 10.13.6
##
```

```
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] bindrcpp_0.2.2  caret_6.0-82    lattice_0.20-38 glmnet_2.0-16
##  [5] foreach_1.4.4   Matrix_1.2-15   forcats_0.3.0   stringr_1.3.1
##  [9] dplyr_0.7.8     purrr_0.3.2     readr_1.3.1     tidyr_0.8.2
## [13] tibble_2.0.1    ggplot2_3.1.0   tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.0         lubridate_1.7.4    class_7.3-15
##  [4] assertthat_0.2.0   digest_0.6.18      ipred_0.9-8
##  [7] R6_2.3.0           cellranger_1.1.0   plyr_1.8.4
## [10] backports_1.1.3    stats4_3.5.1       e1071_1.7-0.1
## [13] evaluate_0.12      httr_1.4.0         pillar_1.3.1
## [16] rlang_0.3.1        lazyeval_0.2.1     readxl_1.2.0
## [19] data.table_1.12.0  rstudioapi_0.9.0   rpart_4.1-13
## [22] rmarkdown_1.11     labeling_0.3       splines_3.5.1
## [25] gower_0.2.0        munsell_0.5.0      broom_0.5.1
## [28] compiler_3.5.1     modelr_0.1.2       xfun_0.4
## [31] pkgconfig_2.0.2    htmltools_0.3.6    nnet_7.3-12
## [34] tidyselect_0.2.5   prodlim_2018.04.18 codetools_0.2-16
## [37] crayon_1.3.4       withr_2.1.2        ModelMetrics_1.2.2
## [40] MASS_7.3-51.1      recipes_0.1.5      grid_3.5.1
## [43] nlme_3.1-137       jsonlite_1.6       gtable_0.2.0
## [46] magrittr_1.5       scales_1.0.0       cli_1.0.1
## [49] stringi_1.2.4      reshape2_1.4.3     timeDate_3043.102
## [52] xml2_1.2.0         generics_0.0.2     lava_1.6.5
## [55] iterators_1.0.10   tools_3.5.1        glue_1.3.0
## [58] hms_0.4.2          survival_2.43-3    yaml_2.2.0
## [61] colorspace_1.4-0   rvest_0.3.2        knitr_1.21
## [64] bindr_0.1.1        haven_2.0.0
```