Lecture Notes in Social Networks

Reda Alhajj · H. Ulrich Hoppe
Tobias Hecking · Piotr Bródka
Przemyslaw Kazienko *Editors*

Network Intelligence Meets User Centered Social Media Networks

Springer

Lecture Notes in Social Networks

More information about this series at http://www.springer.com/series/8768

Reda Alhajj • H. Ulrich Hoppe • Tobias Hecking
Piotr Bródka • Przemyslaw Kazienko
Editors

Network Intelligence Meets User Centered Social Media Networks

Springer

Editors
Reda Alhajj
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada

Tobias Hecking
Computer Science and Applied
Cognitive Science
University of Duisburg-Essen
Duisburg, Nordrhein-Westfalen, Germany

Przemyslaw Kazienko
Faculty of Computer Science
and Management
The Engine Centre
Wroclaw University of Science
and Technology
Wroclaw, Poland

H. Ulrich Hoppe
Department of Computer Science and
Applied Cognitive Science
University of Duisburg-Essen
Duisburg, Nordrhein-Westfalen, Germany

Piotr Bródka
Department of Computational Intelligence
The Engine Centre, Faculty of
Computer Science and Management
Wroclaw University of Science
and Technology
Wroclaw, Poland

ISSN 2190-5428 ISSN 2190-5436 (electronic)
Lecture Notes in Social Networks
ISBN 978-3-319-90311-8 ISBN 978-3-319-90312-5 (eBook)
https://doi.org/10.1007/978-3-319-90312-5

Library of Congress Control Number: 2018944599

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Contents

Part I Centrality and Influence

Part II Knowledge and Information Diffusion

Part III Algorithms and Applications I

Part IV Algorithms and Applications II

Part V Content Analysis

Part I
Centrality and Influence

Targeting Influential Nodes for Recovery in Bootstrap Percolation on Hyperbolic Networks

Christine Marshall, Colm O'Riordan, and James Cruickshank

Abstract The influence of our peers is a powerful reinforcement for our social behaviour, evidenced in voter behaviour and trend adoption. Bootstrap percolation is a simple method for modelling this process. In this work we look at bootstrap percolation on hyperbolic random geometric graphs, which have been used to model the Internet graph, and introduce a form of bootstrap percolation with recovery, showing that random targeting of nodes for recovery will delay adoption, but this effect is enhanced when nodes of high degree are selectively targeted.

Keywords Bootstrap percolation · Bootstrap percolation with recovery · Hyperbolic random geometric graphs

1 Introduction

In this work, we examine agent-based modelling of Bootstrap Percolation on Hyperbolic Random Geometric graphs, and introduce a modified version of Bootstrap Percolation with Recovery on the same set of graphs. Bootstrap percolation has been used to model social reinforcement, with the rationale that individuals are more likely to adopt a new technology, for example, if a significant number of their contacts already use that technology. A key question in bootstrap percolation is whether the activity will percolate or not. Our focus is on the effects of spatial structure on the spread or percolation of an activity, specifically observing bootstrap percolation on hyperbolic random geometric graphs. Our motivation for using hyperbolic networks is they share many features with real-world complex

C. Marshall (✉) · C. O'Riordan
Discipline of Information Technology, National University of Ireland, Galway, Ireland
e-mail: c.marshall1@nuigalway.ie; colm.oriordan@nuigalway.ie

J. Cruickshank
School of Mathematics, National University of Ireland, Galway, Ireland
e-mail: james.cruickshank@nuigalway.ie

R. Alhajj et al. (eds.), *Network Intelligence Meets User Centered Social Media Networks*, Lecture Notes in Social Networks,
https://doi.org/10.1007/978-3-319-90312-5_1

networks; typically displaying power-law degree distribution, high clustering, short path lengths, high closeness and significant betweenness centralisation.

We have previously created a set of hyperbolic random geometric graphs from unconnected, with increasing edge density, to fully connected. In this work, we attach a population of agents to the graphs and explore the effect of percolation processes on these agents on this set of graphs. Our first question is whether we can identify a distinct percolation threshold as we increase the number of edges, above which the activity percolates and below which the activity fails to percolate.

Typically work on bootstrap percolation has looked at the effect of the initial active seed set on the final outcome of the process, looking at the choice of active seed, or at the fraction of active seeds required to guarantee percolation of the activity. This has been used in, for example, viral marketing to see which nodes might best be targeted to optimise the spread of information. Our overall interest is in targeting nodes which might prevent the percolation of the activity, given a random seed set of fixed size. This could model a situation where a network is randomly targeted with active seeds, and we wish to know which nodes might have the best chance of obstructing the process. Our focus is on minimising the spread of an activity given a small-scale attack at random points within the network.

We develop a modified version of standard bootstrap percolation which allows for the recovery of a defined percentage of active nodes after each activation step; by recovery we mean the transition from active state back to inactive state. Our next questions are whether this will impact the spread of the activity, and, furthermore, if we selectively target active nodes of high degree, will this have a greater impact on the spread of activity than the same percentage chosen at random, given that hub nodes are commonly regarded as influential nodes in a network.

Section 2 provides related information about bootstrap percolation and hyperbolic graphs. Section 3 introduces our conceptual framework for Bootstrap Percolation with recovery. Section 4 describes our experimental set-up, and our results are presented in Sect. 5. Our conclusions are discussed in Sect. 6, together with suggestions for future work.

2 Background

In this section we describe bootstrap percolation and hyperbolic random geometric graphs.

2.1 Bootstrap Percolation

Bootstrap percolation is a dynamic process in which an activity can spread to a node in a network if the number of active neighbours of that node is greater than a predefined activation threshold. This process can model forms of social reinforcement where the spread of an activity largely depends on a tipping point in the number of activated contacts. These activities can include the spread of opinions or the adoption of a new technology; the underlying assumption is that people are more likely to adopt a new activity if several of their contacts are already involved in the same activity. The idea of Bootstrap Percolation was introduced by Chalupa, Leath and Reich in their 1979 work on Bethe lattices studying the mechanisms of ferromagnetism, or how materials become magnetised [13].

The bootstrap percolation model is essentially a two-state model, with agents either active or inactive. In some of the literature, these states are termed infected or uninfected, but basically means that an activity is present or not present. Standard bootstrap percolation is a discrete-time process, where the activation mechanism occurs synchronously for all nodes in the graph in rounds, at each time step. Starting with an inactive population of agents, a set of nodes is chosen as the active seed set, this choice may be at random or deterministic. An integer value activation threshold is chosen and the activation mechanism occurs at each time step, whereby a node becomes activated if the number of its active neighbours is at least that value. The rounds of activation are then repeated until equilibrium, where no further state change is possible, or at some predetermined parameter, such as the number of rounds. In the spatial form of the process, agents are attached to nodes in a network and activation depends on the number of active nodes directly connected to a node.

Bootstrap percolation has been studied on a variety of random graphs and complex networks [2–4, 6, 19, 34]. A key question is whether the activity will completely percolate. A lot of research has concentrated on the relationship between the initial active seed set and the final outcome of the process; this has in essence involved looking at the choice of the initial active seed set to optimise the spread of the activity [20], or the size of the initial seed set to ensure percolation [11, 17]. Kempe et al. investigated algorithms for optimising this set selection on a variety of networks, noting that this was NP-hard [21]. The bootstrap percolation process has been used to model information diffusion [23–25] and viral marketing [15], behavioural diffusion [12], such as opinion formation and voter trends, the adoption of brands, technology and innovation in social networks [16, 18, 31], and also cascading failures in power systems.

In the standard form of bootstrap percolation, activated nodes must remain active, no reverse state is allowed. In 2014, Coker and Gunderson investigated the idea of bootstrap percolation with recovery on lattice grids based on an update rule that infected nodes with few infected neighbours will become uninfected [14]. They used a probabilistic approach to determine thresholds for the probability of percolation based on the size of the initial active seed set. By examining various configurations of 2-tiles (pairs of sites that share an edge or a corner in the lattice grid), they determine the critical probabilities for percolation.

In epidemiology, related models are the SI and SIS models of disease propagation [32] which have similar state changes to bootstrap percolation and bootstrap percolation with recovery, respectively. In the SI model, the population is compartmentalised into either Susceptible (i.e. uninfected) or Infected sinks. Each susceptible individual has a probability of becoming infected, based on the transmission rate, which is a function of the typical number of contacts per individual, and of the infectivity of the disease. Once infected, individuals move to the infected compartment and remain infected. The SIS model is an extension of this model where infected individuals will recover (and become susceptible again) based on the rate of recovery.

The key point of interest is the epidemic threshold, and the effects of varying rates of infection and recovery on the spread of the disease. In the SI model, the infection will ultimately spread to the entire population. In the SIS model, for high recovery rates the illness will die out in the population and for low recovery rates the illness will become endemic [5]. These models are compartmental models, where the population in each sink is homogeneous; each individual within a compartment has the same likelihood of changing state.

Bootstrap percolation is distinctly different from the epidemic models, as the population of agents is heterogeneous, each with local knowledge of node properties in their neighbourhood. Once the initial active set has been chosen, the spread of the activity is determined by the number of neighbourhood links to active nodes, which is a function of the underlying network topology.

To address issues of network structure in the epidemic models, Pastor-Satorras and Vespignani in 2001 [27] introduced new compartments based on node degree, with nodes of the same degree having the same likelihood of changing state. In large-scale free models, the hub nodes ensure that the infection is likely to spread [5]. However, this approach does not take into account community structure or local clustering. Agent-based modelling of bootstrap percolation on a network is a simple dynamic process ideally suited to accounting for individual node features.

2.2 Hyperbolic Random Geometric Graphs

In Random Geometric Graphs, pairs of nodes are connected if they lie within a specified distance parameter of each other. They were developed in the 1970s to model situations where the distance between nodes is of interest, as in modelling the spread of a forest fire. These graphs have generally been created in the Euclidean plane, particularly the unit square and unit disc. In computer science, these models are frequently used to model wireless ad-hoc and sensor networks [7], where proximity between nodes defines network connections. Other uses include modelling neural networks [9], mapping protein-protein interactions [28], and in percolation theory [11], modelling processes such as diffusion in a network, fluid percolation in porous materials, fracture patterns in earthquakes and conductivity [30].

Hyperbolic random geometric graphs were developed by Krioukov et al. in 2010 [22]. In this model, pairs of nodes are connected if the hyperbolic distance between them is less than a specified distance parameter. To highlight the contrast between hyperbolic and Euclidean random geometric graphs, the Poincaré disc model transforms the negative curvature of the hyperbolic plane to a 2 dimensional disc. Points within the disc are not uniformly distributed; the hyperbolic model has more

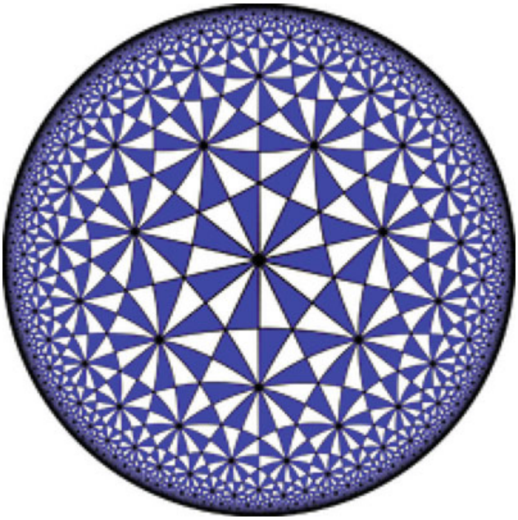

Fig. 1 Claudio Rocchini *Order-3 heptakis heptagonal tiling* 2007, distributed under a CC BY 2.5 licence [29]

"room" than the Euclidean disc. As the radius of a Euclidean disc increases, the circumference increases linearly; in a hyperbolic disc, the circumference increases exponentially. For example, a hyperbolic tree can represent more data as child nodes have as much space as parent nodes. This tends to give a fish-eye view to nodes within the disc, with greater emphasis placed on central nodes, while outer nodes are exponentially distant, this effect is captured in an image by Claudio Rocchini [29], see Fig. 1.

Hyperbolic geometric graphs have been used to explore structural properties and processes in complex networks, such as clustering [10] and navigability. Krioukov et al. have proposed that these models have potential for modelling the Internet graph, which they suggest has an underlying hyperbolic geometry [22]. Recent work by Papadopoulos et al. has developed a tool to map real-world complex networks, such as the Autonomous Systems Internet, to a hyperbolic space using the hyperbolic geometric model [26]. Recently, researchers have developed faster algorithms for creating large hyperbolic networks. Von Looz et al. have developed a speedy generator to create representative subsets of hyperbolic graphs, allowing for networks with billions of edges, while retaining key features of hyperbolic graphs [33], while Bringmann et al. have developed a generator to create a generalised hyperbolic model in linear time, by avoiding the use of hyperbolic cosines [8].

3 Conceptual Framework for Bootstrap Percolation with Recovery

The concept of bootstrap percolation with recovery allows for an activated node to become deactivated. This might model a change of mind situation, in trend adoption or behavioural diffusion, where individuals are initially interested in an activity popular with their contacts, but immediately change their mind and no longer endorse the activity, at least in the short term. In standard bootstrap percolation, a lot of work focuses on optimising the size and selection of the initial active seed set to ensure percolation. Our own focus is on a fixed-size randomly chosen seed set, motivated by the notion of a small-scale random seeding in a network, with a view to targeting recovery to inhibit the spread of the activity. Our proposed method of recovery involves targeting active nodes with specific node properties, such as high degree centrality; our aim is to identify those properties that might have the most impact on delaying the percolation process. We have chosen node degree, in the first instance, as node degree is a standard measure of influence in networks, [1, 5], and the hyperbolic geometric graphs display highly skewed degree centralisation, reflecting the high level of variance in node degree within each graph. We examine the effect that this has on the percolation of the activity, in comparison with the standard bootstrap percolation process, in which recovery is not permitted. As a control, we also simulate bootstrap percolation with random selection of active nodes for recovery.

3.1 Proposed Method of Bootstrap Percolation with Recovery

Our proposed method involves setting up the experiments as for the standard bootstrap process but introducing a recovery mechanism to follow the activation mechanism in each time step.

1. Set Parameters
   - Set size of active seed set A_o
   - Set integer value activation threshold r
   - Set recovery rate percentage $RR\%$
2. Define Mechanisms
   - Activation mechanism
     - For all inactive nodes, activate if the number of active neighbours is at least r.
   - Recovery mechanism
     - For all active nodes, select $RR\%$ to deactivate, by random or deterministic selection
3. Define Process
   - For each graph, attach an agent to each node in the graph
   - Set all agents inactive
   - Select active seed set
   - Apply activation mechanism followed by recovery mechanism at each time step until equilibrium

4 Experimental Set-up

Our simulations involve creating a set of hyperbolic random geometric graphs and observing the effect of graph topology on the spread of activity by embedding the bootstrap percolation process on these graphs.

A set of hyperbolic geometric graphs, each with 1000 nodes, was created with varying edge density from disconnected to fully connected, with 20 graphs created at each value of the distance parameter. Bootstrap percolation was embedded on this set of graphs and then our modified versions of bootstrap percolation with recovery, both random and targeted, were simulated on the same set of graphs to compare outcomes. The recovery experiments were repeated with varying recovery rate percentages ($RR\%$) of 10–90%, in steps of 10%.

4.1 Graph Creation

In this research, hyperbolic random geometric graphs are created using the model outlined by Krioukov et al. [22]. This approach situates the graph in a disc of radius R within the Poincaré disc model of hyperbolic space. Nodes in the graph have polar coordinates (r, θ) and are positioned by generating these coordinates as random variates; edges are created between pairs of nodes where the hyperbolic distance between them is less than the radius R of the disc of interest, varying R from 0.1 to 12 in increments of 0.1, to create graphs with an increasing number of edges, from disconnected to fully connected, with 20 graphs created at each distance parameter R. The relationship with R and edge density is not linear, see Fig. 2; therefore, R is used in all our results for ease of comparison.

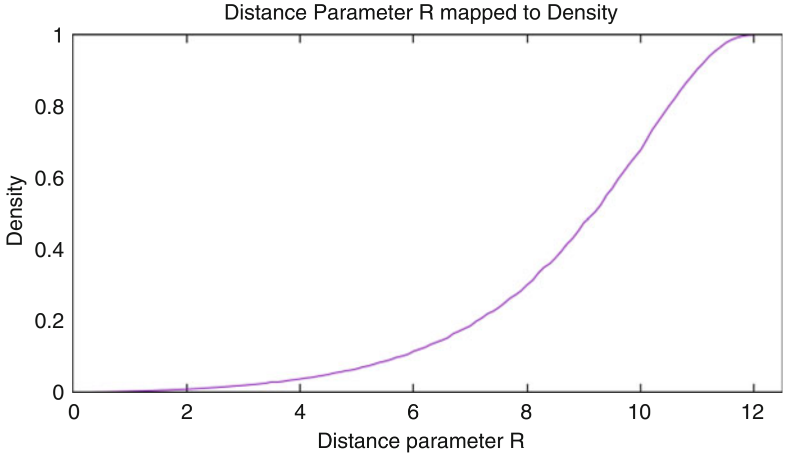

Fig. 2 Density parameter R mapped to edge density

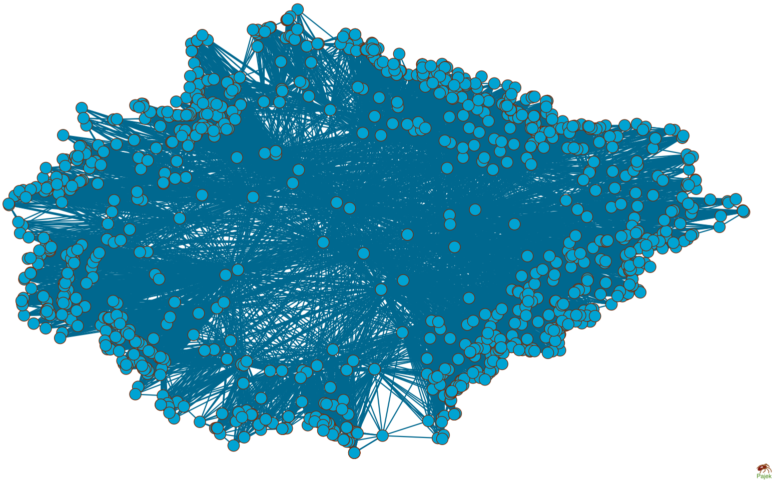

Fig. 3 Hyperbolic random geometric graph with 1000 nodes and an edge density of 0.037

Figure 3 shows a hyperbolic random geometric graph with 1000 nodes, distance parameter $R = 4$, and an edge density of 0.037. This shows a tree-like structure, with the major hub nodes in the centre and the majority of nodes, that is, those of lower degree, are situated towards the boundary.

4.2 Simulation of Bootstrap Percolation on the Set of Hyperbolic Graphs

A population of agents, engaged in the process of bootstrap percolation, is attached to each graph to observe any spread in activity. An agent is assigned to each node, with all agents initially inactive and, from these, 20 nodes are randomly selected to form the initial active seed set, A_o. The activation mechanism involves an activation threshold r whereby an inactive node becomes active if it has at least r active neighbours. At each time step, this activation mechanism is applied synchronously to all nodes in the graph and is repeated until no further state change is possible and a stable equilibrium is reached; the number of active nodes is recorded at this point. Our experiments are then repeated varying the activation threshold r from 2 to 10.

4.3 Bootstrap Percolation with Recovery

The bootstrap percolation process is modified to allow for reverse state change from active to inactive, as outlined in Sect. 3.1 by introducing a recovery phase in each time step, immediately following activation. Our first set of experiments involves random selection of active nodes for recovery. The experiments are then repeated targeting active nodes of high degree for recovery. In both sets of simulations, the experiments are repeated with varying Recovery Rate percentages ($RR\%$) from 10 to 90% in steps of 10% for each activation threshold in the range 2–10, as before.

As for the standard process, a population of agents, engaged in the process of bootstrap percolation, is attached to each graph to observe any spread in activity. An agent is assigned to each node, with all agents initially inactive and, from these, 20 nodes are randomly selected to form the initial active seed set, A_o. The activation mechanism involves an activation threshold r whereby an inactive node becomes active if it has at least r active neighbours. The recovery mechanism involves selecting a percentage of the active nodes for deactivation.

At each time step, the activation mechanism is applied synchronously to all nodes in the graph followed by the recovery mechanism applied to all active nodes. This cycle of activation and recovery at each time step is repeated until equilibrium; the number of active nodes is recorded at this point. Our experiments are then repeated at this Recovery Rate varying the activation threshold r from 2 to 10. We then repeat the experiments incrementing the recovery rate by 10% for each set, from 10 to 90%.

Our test simulations showed that equilibrium was reached rapidly for all graphs, and we therefore set an arbitrary cut-off point at 100 time steps, to ensure we captured equilibrium. In the case of graphs which had complete percolation in the standard bootstrap process, equilibrium cycled between full activation and deactivation of the recovery rate fraction.

Random Recovery

After each activation time step, the recovery rate percentage of active nodes is randomly selected to become inactive.

Targeted Recovery Based on Node Degree Ranking

After each activation time step, active nodes were ranked according to node degree, from highest to lowest, the recovery rate percentage of the top-ranked active nodes is then selected to become inactive.

5 Results

In this section we describe our results from simulating the processes of bootstrap percolation and bootstrap percolation with random recovery and then targeted recovery, all on the same set of hyperbolic random geometric graphs, using averaged data from the 20 graphs created at each parameter R.

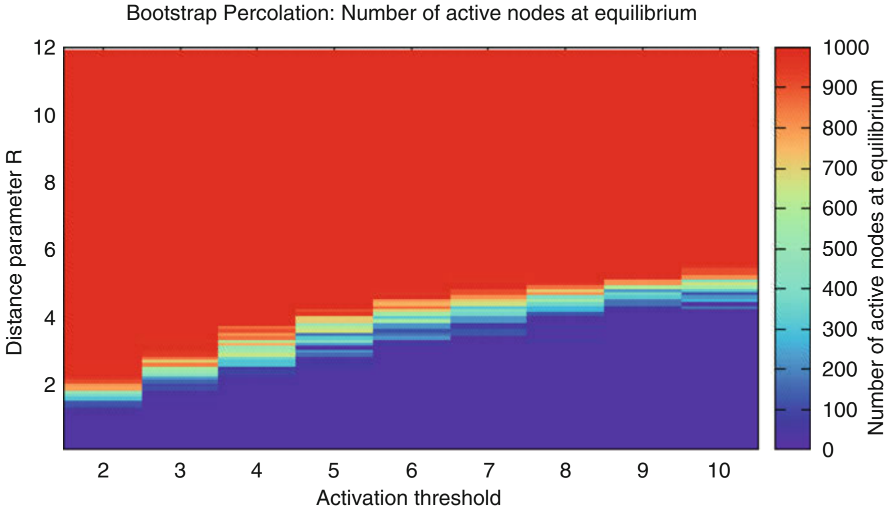

Fig. 4 Heat Map for $A_o = 20$, with AT from 2 to 10, and $R = 0.1$ to 12, showing the number of final active nodes at equilibrium

5.1 Bootstrap Percolation

The heatmap in Fig. 4 shows a distinct threshold above which the activity completely percolated and below which the activity failed to percolate. This percolation threshold lay between values of R from 1.5 to 5.8, representing edge densities of 0.005–0.1, respectively, demonstrating increasing edge density with each increase in activation value. Each increase in activation threshold has an inhibiting effect on the spread of activity, as it requires more of a node's contacts to be active before the activity will spread. Increasing the edge density allows an inactive node greater potential access to activated contacts, and this is manifested by the increase in the percolation threshold.

In all of our simulations, the heat maps displayed a clear percolation threshold. For comparison purposes, we developed a simple algorithm to find a representative point within the threshold, to allow for representation of the percolation threshold as a curve.

5.2 Bootstrap Percolation with Recovery

In the following charts, the standard Bootstrap Percolation process is depicted as a case of 0% recovery. Each fence plot has a differing recovery rate percentage ($RR\%$) and represents the percolation threshold for that particular recovery rate.

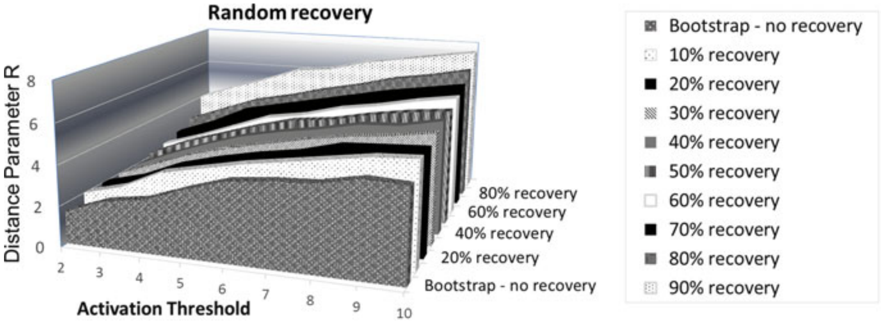

Fig. 5 Bootstrap percolation with RANDOM recovery

It can be clearly seen that either form of recovery significantly raises the threshold for complete percolation of the activity and that, in general, each increase in $RR\%$ increases this stepwise.

Bootstrap Percolation with Random Recovery

Figure 5 shows bootstrap percolation with recovery randomly targeting a percentage of active nodes following activation in each time step. It can be seen that, overall, the edge density of the threshold increases as the recovery rate increases from 10 to 90%, with the form of each curve similar to its neighbour. However, after the initial increase at 10%, increasing the recovery rate had little further effect before 50% recovery.

We can also gain information by looking at the edges of each curve. Percolation threshold values arising from an activation threshold of 2 are depicted on the left wall of the fence plot, whereas percolation threshold values arising from increasing the activation threshold to 10 are depicted on the right edge; the increase in activation threshold means that more active contacts are required for an inactive node to become activated. For an activation threshold of 2, these values for R at the percolation threshold increased from 2 to 4.7, representing edge densities ranging from 0.01 to 0.055. For an activation threshold of 10, the values for R at the percolation threshold increased from 5.2 to 8, representing edge densities from 0.075 to 0.3.

Bootstrap Percolation with Targeted Recovery

Figure 6 shows bootstrap percolation with recovery selectively targeting the top-ranked active nodes of the highest degree after each activation period. This clearly shows that the corresponding edge density of thresholds has increased significantly for all parameters, when compared with randomly selected recovery. Unlike random recovery, where the increase in recovery percentage had no effect between 10

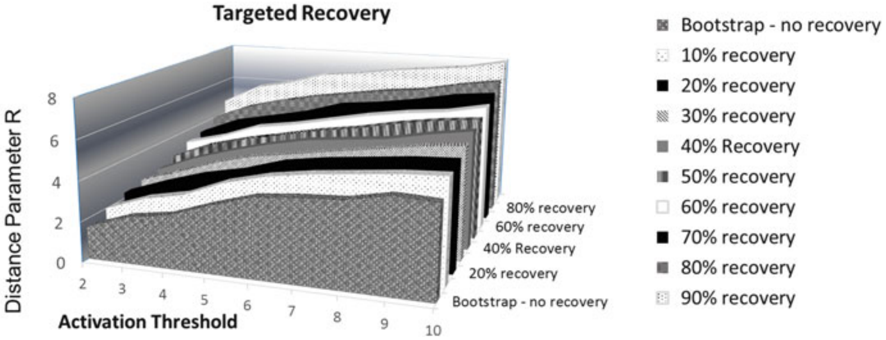

Fig. 6 Bootstrap percolation with TARGETED recovery

and 50%, the percolation threshold curves for targeted recovery showed a regular stepwise increase as the recovery percentage increased, for all percentages.

The most notable observation, with each percentage increase in recovery rate, was that for an activation threshold of 2, the percolation threshold occurred at values of R between 2 and 4, representing edge densities in the range 0.01–0.037, and that for an activation threshold of 10, the values for R at the percolation threshold increased from 5.2 to 7.6, representing edge densities of 0.075–0.25, with the form of each curve again similar to its neighbour. This demonstrates that selectively targeting the nodes of the highest degree for recovery has had a significantly greater impact on the spread of the activity compared with random selection, and particularly when compared with the standard bootstrap process.

6 Conclusion

We have studied bootstrap percolation on hyperbolic geometric graphs and noticed that there was a clear threshold above which the activity percolated to all nodes and below which the activity failed to percolate. We developed a modified form of bootstrap percolation which allowed for recovery of active nodes to inactive state. In our experiments with recovery on the same set of graphs, we noticed that with random recovery, after the initial increase at 10%, there was little further impact until we increased the recovery rate to 50%, at which the percolation threshold was significantly increased to graphs with higher edge density, with all curves of similar form to their neighbours.

Our experiments with targeted recovery, based on the top-ranked nodes of the highest degree, showed that this delayed the threshold even further, and had a stepwise increase in the threshold for all recovery rate increases. This suggests that nodes of high degree are influential in the bootstrap percolation process. Following these results, we surmise that it may be possible to fine-tune our targeting, by choosing differing node properties, or by targeting nodes highly ranked for a combination of properties.

Our next intention is to target active nodes for recovery based on properties which are significantly skewed in the hyperbolic graphs, such as clustering coefficient, betweenness centralisation and closeness centralisation. Our intuition is that finding measures which are highly skewed in the hyperbolic graphs will allow us to significantly delay percolation.

It would also be interesting to specifically target the most influential nodes in the network, with a priori rankings, so that whenever such a node became active it would always revert to inactivity. This would effectively immunise that node from activity and would represent an influential player constantly resisting an onslaught of activity.

Another potential area of study would concentrate on graphs within the percolation threshold zone of transition, investigating graph or individual node properties that facilitate the process above a certain edge density, or those which impede percolation below a certain edge density. It might be useful to observe the impact of targeted recovery within this percolation threshold in order to determine the local structural properties of individual nodes, or neighbourhoods, which might have the greatest impact on global outcomes in the graph.

References

1. Albert, R., Jeong, H., Barabási, A.L.: Error and attack tolerance of complex networks. Nature **406**(6794), 378–382 (2000)
2. Amini, H., Fountoulakis, N.: Bootstrap percolation in power-law random graphs. J. Stat. Phys. **155**(1), 72–92 (2014)
3. Balister, P., Bollobàs, B., Johnson, J.R., Walters, M.: Random majority percolation. Random Struct. Algoritm. **36**(3), 315–340 (2010)
4. Balogh, J., Pittel, B.G.: Bootstrap percolation on the random regular graph. Random Struct. Algoritm. **30**(12), 257–286 (2007)
5. Barabási, A.L.: Network Science. Cambridge University Press, Cambridge (2016)
6. Baxter, G.J., Dorogovtsev, S.N., Goltsev, A.V., Mendes, J.F.: Bootstrap percolation on complex networks. Phys. Rev. E **82**(1), 011103 (2010)
7. Bénézit, F., Dimakis, A.G., Thiran, P., Vetterli, M.: Order-optimal consensus through randomized path averaging. IEEE Trans. Inf. Theory **56**(10), 5150–5167 (2010)
8. Bringmann, K., Keusch, R., Lengler, J.: Geometric inhomogeneous random graphs. Preprint (2015). arXiv:1511.00576
9. Bullmore, E., Bassett, D.: Brain graphs: graphical models of the human brain connectome. Annu. Rev. Clin. Psychol. **7**, 113–140 (2011)
10. Candellero, E., Fountoulakis, N.: Clustering and the hyperbolic geometry of complex networks. In: Bonato, A., Graham, F., Pralat, P. (eds.) Algorithms and Models for the Web Graph. WAW 2014. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8882, pp. 1–12. Springer, Cham (2014)
11. Candellero, E., Fountoulakis, N.: Bootstrap percolation and the geometry of complex networks. Stoch. Process. Appl. **126**, 234–264 (2015)
12. Centola, D.: The spread of behavior in an online social network experiment. Science **329**(5996), 1194–1197 (2010)
13. Chalupa, J., Leath, P.L., Reich, G.R.: Bootstrap percolation on a bethe lattice. J. Phys. C Solid State Phys. **12**(1), L31 (1979)

14. Coker, T., Gunderson, K.: A sharp threshold for a modified bootstrap percolation with recovery. J. Stat. Phys. **157**(3), 531–570 (2014)
15. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '01, pp. 57–66. ACM, New York (2001)
16. Gleeson, J.P.: Cascades on correlated and modular random networks. Phys. Rev. E **77**(4), 046117 (2008)
17. Gomez Rodriguez, M., Leskovec, J., Krause, A.: Inferring networks of diffusion and influence. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1019–1028. ACM, New York (2010)
18. Jackson, M.O., López-Pintado, D.: Diffusion and contagion in networks with heterogeneous agents and homophily. Netw. Sci. **1**(01), 49–67 (2013)
19. Janson, S., Łuczak, T., Turova, T., Vallier, T.: Bootstrap percolation on the random graph $g_{n,p}$. Ann. Appl. Probab. **22**(5), 1989–2047 (2012)
20. Kempe, D., Kleinberg, J.M., Tardos, É.: Influential nodes in a diffusion model for social networks. In: ICALP, vol. 5, pp. 1127–1138. Springer, Berlin (2005)
21. Kempe, D., Kleinberg, J.M., Tardos, É.: Maximizing the spread of influence through a social network. Theory Comput. **11**(4), 105–147 (2015)
22. Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A., Boguñá, M.: Hyperbolic geometry of complex networks. Phys. Rev. E **82**, 036106 (2010)
23. Leskovec, J., Backstrom, L., Kleinberg, J.: Meme-tracking and the dynamics of the news cycle. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 497–506. ACM, New York (2009)
24. Liben-Nowell, D., Kleinberg, J.: Tracing information flow on a global scale using internet chain-letter data. Proc. Natl. Acad. Sci. **105**(12), 4633–4638 (2008)
25. Myers, S.A., Zhu, C., Leskovec, J.: Information diffusion and external influence in networks. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 33–41. ACM, New York (2012)
26. Papadopoulos, F., Psomas, C., Krioukov, D.: Network mapping by replaying hyperbolic growth. IEEE/ACM Trans. Networking **23**(1), 198–211 (2015)
27. Pastor-Satorras, R., Vespignani, A.: Epidemic spreading in scale-free networks. Phys. Rev. Lett. **86**(14), 3200 (2001)
28. Pržulj, N.: Biological network comparison using graphlet degree distribution. Bioinformatics **23**(2), e177–e183 (2007)
29. Rocchini, C.: Order-3 heptakis heptagonal tiling. https://commons.wikimedia.org/wiki/File:Order-3_heptakis_heptagonal_tiling.png (2007). Accessed 15 May 2017
30. Sahini, M., Sahimi, M.: Applications of Percolation Theory. CRC Press, Boca Raton (1994)
31. Shrestha, M., Moore, C.: Message-passing approach for threshold models of behavior in networks. Phys. Rev. E **89**(2), 022805 (2014)
32. Tassier, T.: Simple epidemics and SIS models. In: The Economics of Epidemiology, pp. 9–16. Springer, Berlin (2013)
33. von Looz, M., Staudt, C.L., Meyerhenke, H., Prutkin, R.: Fast generation of dynamic complex networks with underlying hyperbolic geometry. Preprint (2015). arXiv:1501.03545
34. Watts, D.J.: A simple model of global cascades on random networks. Proc. Natl. Acad. Sci. **99**(9), 5766–5771 (2002)

Process-Driven Betweenness Centrality Measures

Mareike Bockholt and Katharina A. Zweig

Abstract In network analysis, it is often desired to determine the most central node of a network, for example, for identifying the most influential individual in a social network. Borgatti states that almost all centrality measures assume that there exists a process moving through the network from node to node (Borgatti, Soc Netw 27(1):55–71, 2005). A node is then considered as central if it is important with respect to the underlying process. One often used measure is the betweenness centrality which is supposed to measure to which extent a node is "between" all other nodes by counting on how many shortest paths a node lies. However, most centrality indices make implicit assumptions about the underlying process. However, data containing a network and trajectories that a process takes on this network are available: this can be used for computing the centrality. Hence, in this work, we use existing data sets, human paths through the Wikipedia network, human solutions of a game in the game's state space, and passengers' travels between US American airports, in order to (1) test the assumptions of the betweenness centrality for these processes, and (2) derive several variants of a "process-driven betweenness centrality" using information about the network process. The comparison of the resulting node rankings yields that there are nodes which are stable with respect to their ranking while others increase or decrease in importance dramatically.

Keywords Network analysis · Centrality measures · Network processes · Path data analysis

M. Bockholt (✉) · K. A. Zweig
Department of Computer Science, University of Kaiserslautern, Kaiserslautern, Germany
e-mail: mareike.bockholt@cs.uni-kl.de; zweig@cs.uni-kl.de

R. Alhajj et al. (eds.), *Network Intelligence Meets User Centered Social Media Networks*, Lecture Notes in Social Networks,
https://doi.org/10.1007/978-3-319-90312-5_2

1 Introduction

An often performed task in network analysis is the identification of the most important nodes in a network. The goal might be to find the most influential individual in a social network, the most vulnerable location in a transportation network, or the leader in a terrorist network [6, 17]. The identification of such nodes is usually done with a centrality measure which computes a value for each node of the network based on the network structure [2, 10, 20]. The concept of centrality in networks was first introduced by Bavelas in the late 1940s who considered human communication networks [2]. Inspired by this idea, a large number of different methods for measuring the centrality of a node were proposed in the following decades, where the best known centrality measures are degree centrality [9], closeness centrality [9], betweenness centrality [1, 9], and Eigenvector centrality [3] (for an overview, cf. [5] or [14]).

An important contribution was made by Borgatti who states that almost all centrality indices are based on the assumption that there is some kind of traffic (or communication or process) flowing through the network by moving from node to node [4]. This might be the propagation of information in a social network, packages being routed through the WWW, or the spreading of a disease in human interaction networks. A node is then considered as central, that is it is assigned a high value of the measure, if it is somehow important with respect to this underlying process. However, different centrality measures make different assumptions about the process flowing through the network. Some measures are based on shortest paths in the network, assuming that the underlying process moves on shortest paths. Others assume that, if whatever flows through the network is at one node, it will spread simultaneously to all the node's neighbors, while others assume that it can only be at one node at a time.

Borgatti provides a typology of the most popular centrality measures by considering the network process [4] and argues that centrality measure values for a network can only be interpreted in a meaningful way, if the assumptions of the measure respect the properties of the process. He identifies two different dimensions by which the process can vary: First, on which type of trajectory does the process move through the network, and second, how does the process spread from node to node? For the first dimension, he differentiates between shortest paths, paths (not necessarily shortest, but nodes and edges can only occur at most once in it), trails (edges might occur several times in it, while nodes cannot be repeated), and walks (in which nodes and edges might occur several times). The second identified dimension is the "mechanism of node-to-node transmission": A process might *transfer* a good from node to node (e.g. a package), or it passes something to the next node by *duplicating* it: whatever flows through the network is passed to the next node and simultaneously stays at the current node (e.g. information flowing through a network or a viral infection: it will still be at the current node after it was passed to another node). Duplication can take place in a serial (one neighbour at once) or in a