SIAM Workshop on Network Science (NS16)

Co-Chairs: John Gilbert and Blair D. Sullivan

July 15-16, 2016
The Westin Boston Waterfront, Boston, Massachusetts, USA

Table of Contents

NETWORK APPROACHES FOR BUILDING NEW INSIGHTS FROM GENE ANNOTATIONS

Michelle Girvan
University of Maryland, College Park

SIAM Workshop on Network Science 2016
July 15-16 · Boston

The Gene Ontology (GO) provides a controlled vocabulary of terms for describing gene functions and specifies how these functional terms are related to one other. Biologists then submit annotations connecting individual genes to appropriate functional terms. The resulting gene annotation databases are commonly used to evaluate the functional properties of experimentally derived gene sets. Here we discuss novel methods to analyze the network structure of gene annotations in order to (1) correct for biases in traditional functional enrichment statistics by appropriately accounting for heterogeneities of connections across genes and functions (2) establish an alternate natural grouping of biological functions that is very different from the conceptual hierarchical structure that relates functional terms in the Gene Ontology. To correct for biases in standard overlap statistics, we develop Annotation Enrichment Analysis (AEA), which accounts for heterogeneity of connections across genes and functions in bipartite annotation networks. We show that AEA is able to identify biologically meaningful functional enrichments that are obscured by numerous false-positive enrichment scores in traditional methods, and we therefore suggest it be used to more accurately assess the biological properties of gene sets. In order to identify relationships between biological functions, we use multi-scale network community finding methods to identify groups of functions that are closely related through shared connections to genes. Grouping terms by our alternate scheme provides a new framework with which to describe and predict the functions of experimentally identified groups of genes.

THROUGH THE LENS OF THE LAPLACIAN PARADIGM: BIG DATA AND SCALABLE ALGORITHMS – A PRAGMATIC MATCH MADE ON EARTH

Shang-Hua Teng
University of Southern California

SIAM Workshop on Network Science 2016
July 15-16 · Boston

In the age of Big Data, efficient algorithms are in higher demand now more than ever before. While Big Data takes us into the asymptotic world envisioned by our pioneers, the explosive growth of problem size has also significantly challenged the classical notion of efficient algorithms: Algorithms that used to be considered efficient, according to polynomial-time characterization, may no longer be adequate for solving today's problems. It is not just desirable, but essential, that efficient algorithms should be scalable. In other words, their complexity should be nearly linear or sub-linear with respect to the problem size. Thus, scalability, not just polynomial-time computability, should be elevated as the central complexity notion for characterizing efficient computation. In this talk, I will discuss the emerging Laplacian Paradigm, which has led to breakthroughs in scalable algorithms for several fundamental problems in network analysis, machine learning, and scientific computing. I will focus on three recent applications: (1) PageRank Approximation (and identification of network nodes with significant PageRanks). (2) Random-Walk Sparsification. (3) Scalable Newton's Method for Gaussian Sampling.

RUFFLED FEATHERS: WHEN CAN GENDER BE INFERRED ON SOCIAL NETWORKS?

Kristen M. Altenburger, Johan Ugander

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We perform a broad study of structural predictors of gender in social networks, contributing a taxonomy of frameworks to categorize and contrast subtly different approaches based on various sociological perspectives. We show these distinct approaches achieve dramatically different performance when predicting gender that we attribute to an empirical overdispersion of individual homophilic tendencies relative to a gender-shuffled graph. These extreme gender affinities introduce *gender canaries* in the graph that reveal other's gender, and we study their role in diffusion-based gender inference. These findings provide a new perspective on social network trait inference in general and gender in particular, complicating the already difficult task of protecting anonymity in social networks, and introducing new considerations for the study of social network covariates.

Gender Inference on Social Networks

Given a graph that is partially labeled with male and female labels, we propose and explore a taxonomy of three basic frameworks for gender inference based on social structure: (1) *label-independent inference* covers methods that employ structural features unrelated to the gender labels on the graph, (2) *label-dependent inference* includes methods that utilize gender-labeled structure, and (3) *relational inference*, which describes methods that harness relations between an inference target and other specific nodes in the dataset. We study label-independent/dependent inference as it is typically applied, employing a combination of local structural features (graph invariants) of the 1-hop neighborhood of an ego node as well as global features such as centrality. While previous efforts have separately evaluated label-independent/dependent [2] and relational inference [6], our work unifies these literatures with a comprehensive evaluation of the relative performance within a joint framework specifically aimed at understanding when gender can be inferred in social networks.

For label-independent inference, many social theories suggest gender is correlated with label-independent graph measures such as centrality [3] or measures of structural hole position [1]. We surprisingly find label-independent based metrics are not practically useful for gender inference, at least within the friendship network of the college population we study. For label-dependent inference, we observe that gender-labeled features of the 1-hop neighborhood are mildly predictive of an ego node's gender. We interpret this improvement in performance from label-dependent features to suggest that while the network positions of males and females are practically indistinguishable, the larger patterns of gendered connections is comparatively predictive of gender.

Within relational inference, a critical assumption underlying diffusion-based approaches is homophily, a commonly known phenomenon whereby "birds of a feather flock together". A documented but underappreciated challenge in predicting gender on social networks is the minimal presence of general gender homophily [5]. Even though there is only very slight gender homophily within the college network we study, the relational inference achieves strikingly higher performance than the label-dependent inference. We attribute the success of relational inference in part to a strong overdispersion of gender friendship affinities, which is plainly apparent when the distribution of individual gender preferences in the observed network is compared to the distribution of gender preferences in a null model network where the gender labels of nodes are randomly shuffled, see Figure 1(a). These extreme gender affinities introduce *gender canaries* in the graph, which reveal other's gender and serve as useful features for relational gender inference.

Analysis on a College Social Network

We focus our examination here on the performance of each framework on one college from the Facebook100 dataset [4], Amherst, which contains $n = 2,235$ users and $m = 90,954$ friendships. Based on self-reported gender labels, there are 45.4% female (F), 45.5% male (M), and 9.1% missing labels in the college graph. For evaluating gender inference performance, we restrict this dataset to

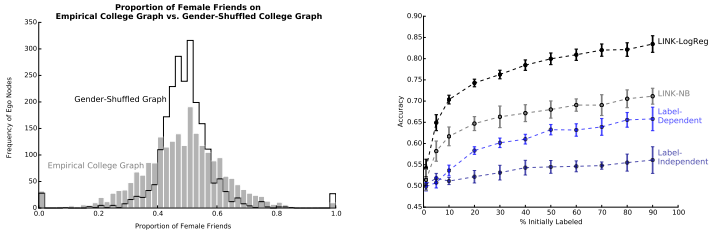


Figure 1: a) Left - Overdispersion of single gender friends on graph. b) Right - Accuracy results on graph.

nodes with at least 3 friends and with reported F/M labels ($n^* = 1999$ users; 1000 F, 999 M) but use all nodes when creating structural features.

We vary the percentage of initially labeled nodes from 1% to 95% by selecting a random sample of nodes to be initially labeled. For cross-validation, multiple sets of the label-dependent features are created as the features themselves depend on which set of nodes are initially labeled. Only one set of the label-independent and relational inference features are created as they do not depend on which set of nodes are initially labeled. We employ a standard arsenal of label-independent/dependent features [2]. Incidentally, we observe that the performance of both frameworks is comparable whether based on global structural features from the full network or based only on local structural features from ego networks. Here we report performances based on local features.

We train our models on the x% labeled individuals (training dataset), and measure performance based on the accuracy of each framework when classifying the remaining unlabeled nodes (testing dataset). We report our accuracies with averages and standard deviations across 10 iterations for each x% of initially labeled nodes in the cases of label-independent and relational inference, and 4 iterations for label-dependent inference (which necessitated recomputing features for each iteration, requiring considerably more computation).

Figure 1(b) shows the accuracy performance of label-independent/dependent methods based on a logistic regression model applied to each feature set where we define accuracy to be the proportion of correct gender predictions on the testing dataset. Beginning with the label-independent features, we see poor accuracy results, which is unexpected given earlier sociological work suggesting structural gender differences in friendship networks. For label-dependent features, we observe modest performance. When we examined relational inference via the LINK approach [6] fitting both a logistic regression (LINK-LogReg) and naive bayes model (LINK-NB), we see a drastic improvement over the previous two methods, across the full range of percentages of labeled nodes. This performance is particularly impressive in light of the fact that relational inference models can only learn relationships between traits (gender) and specific individuals, rather than between traits and generic structural features.

This analysis highlights that homophily is a *sufficient but not necessary condition for gender inference, and that overdispersion is a weaker but sufficient condition.* Finally, in order to isolate and generalize the consequences of overdispersed gender affinities in friendship networks, we introduce an overdispersed stochastic block model that can independently capture affinities with either block structure, overdispersion or both.

Conclusions

We pose three challenges to the network science community: First, our analysis provides no theoretical justification for limiting the predictive performance of label-dependent gender inference relative to relational gender inference, and it is possible that the "right" label-dependent features enable predictions on par with relational inference. Secondly, it is also possible, albeit improbable, that label-independent gender inference can perform better as well. Third, the formation process of this overdispersion property remains open for further investigation.

References

[1] R. S. Burt. The gender of social capital. *Rationality and Society*, 10(1):5–46, 1998.

[2] B. Gallagher and T. Eliassi-Rad. Leveraging label-independent features for classification in sparsely labeled networks: An empirical study. In *Proceedings of ASONAM*, pages 1–19, 2010.

[3] H. Ibarra. Paving an alternative route: Gender differences in managerial networks. *Soc Psych Quarterly*, pages 91–102, 1997.

[4] A. L. Traud, E. D. Kelsic, P. J. Mucha, and M. A. Porter. Comparing community structure to characteristics in online collegiate social networks. *SIAM Review*, 53(3):526–543, 2011.

[5] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the facebook social graph. *arXiv:1111.4503*, 2011.

[6] E. Zheleva and L. Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of WWW*, pages 531–540. ACM, 2009.

COMMUNITY DETECTION IN TEMPORAL MULTILAYER NETWORKS

Marya Bazzi, Mason A. Porter, Sam D. Howison

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We investigate a community-detection method known as "multilayer modularity maximization" for time-dependent networks represented as multilayer networks. We explore some of its theoretical properties and discuss how one can solve it in practice. We propose a benchmark for community detection in time-dependent networks and use it to compare the performance of community-detection methods and algorithms.

Introduction

Given a network representation of a system, it can be useful to apply a coarse-graining technique in order to investigate features that lie between features at the "microscale" and the "macroscale" An example of such structure is "community structure". Loosely speaking, a *community* in a network is a set of nodes that are "more densely" connected to each other than they are to nodes in the rest of the network [8].

Most methods for detecting communities are designed for static networks. However, in many applications, entities and/or interactions between entities evolve in time. In such applications, one can use the formalism of *temporal networks*, where nodes and/or their edge weights vary in time [5]. Two main approaches have been adopted to detect communities in time-dependent networks. The first entails constructing a static network by aggregating snapshots of the evolving network at different points in time into a single network. One can then use standard network techniques. The second approach entails using static community-detection techniques on each element of a time-ordered sequence of networks at different times or on each element of a time-ordered sequence of network aggregations over different time intervals and then tracking the communities across the sequence.

A third approach consists of embedding a time-ordered sequence of networks in a larger network. Each element of the sequence is a network *layer*, and nodes at different time points are joined by *interlayer* edges. This approach was introduced in [7] and the resulting network is a type of *multilayer network* [6]. The main difference between this approach and the previous approach is that the presence of nonzero interlayer edges introduces a dependence between communities identified in one layer and connectivity patterns in other layers. We show an example of a multilayer network with uniform, "diagonal" (i.e., they exist only between copies of the same node), and "ordinal" (i.e., they exist only between consecutive layers) interlayer coupling edges in Fig 1.

The authors of [7] proposed a generalization of *modularity maximization*, a popular clustering method for static networks, to multilayer networks. Modularity is a quality function that compares edge weights in an observed network to expected edge weights in a "null network" (generated from a "null model"), and modularity maximization is a discrete optimization problem that consists of maximizing this quality function over the space of partitions. To date, almost no theory has explained how a multilayer partition obtained with zero interlayer coupling (which reduces to single-layer modularity maximization on each layer independently) differs from a multilayer partition obtained with nonzero interlayer coupling. We prove several theoretical properties of an optimal solution for the multilayer maximization problem to better understand how such partitions differ and how one can exploit this difference in practice [2]. We also describe two computational issues that arise when using the popular Louvain heuristic [3] to solve the multilayer maximization problem, and we suggest ways to mitigate them. Finally, we propose a benchmark for community detection in time-dependent networks and use it to compare the performance of community-detection methods and algorithms [1].

Multilayer modularity maximization

We rewrite the multilayer modularity maximization problem in [7] as follows:

$$\max_{C\in\mathcal{C}} \left[\underbrace{\sum_{s=1}^{|\mathcal{T}|}\sum_{i,j=1}^{N} B_{ijs}\delta(c_{i_s},c_{j_s})}_{\text{intralayer modularity}} + 2\omega \underbrace{\sum_{s=1}^{|\mathcal{T}|-1}\sum_{i=1}^{N}\delta(c_{i_s},c_{i_{s+1}})}_{\text{persistence}} \right],$$

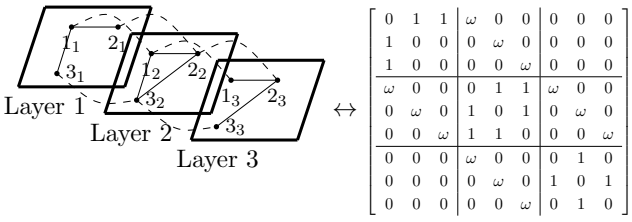


Figure 1: Example of (left) a multilayer network with unweighted intralayer connections (solid lines) and uniformly weighted interlayer connections (dashed curves) and (right) its corresponding adjacency matrix.

where $\mathcal{C}$ is the set of all multilayer partitions, N is the number of nodes in each layer, $|\mathcal{T}|$ is the number of layers, B_{ijs} is the modularity contribution between node i and node j in layer s, i_s is the i^{th} node in the s^{th} layer, $\omega \geq 0$ is the interlayer edge weight, and $\delta(c_{i_s}, c_{j_r})$ is the Kronecker delta function.

We define a measure that we call *persistence* and show that an optimal partition in multilayer modularity maximization reflects a trade-off between time-independent community structure within layers (i.e., "intralayer modularity") and persistence of community structure across layers. We prove several properties that describe the effect of interlayer coupling on an optimal solution and illustrate how one can exploit these in practice. Our multilayer analysis only depends on the form of the maximization problem and still holds if one uses a quality function other than the modularity quality function, provided it has the same form. Furthermore, we illustrate two issues that can arise when one uses the popular locally-greedy "Louvain" computational heuristic [3] to solve the multilayer maximization problem. We propose ways to try to mitigate these issues and show numerical experiments on real data as illustrations.

Temporal benchmark for community detection

While most would agree that a community should correspond to a set of nodes that is "surprisingly well-connected", there is no agreed-upon definition of community that one can compare against. Different applications warrant different interpretations of "surprisingly well-connected" and different methods were often developed with different definitions in mind [8]. Furthermore, most community-detection methods cannot be solved in polynomial time and popular scalable heuristics currently have few or no theoretical guarantees on how closely an identified partition resembles an optimal partition [4]. Benchmark networks with known structural properties are thus an important tool for analysing and comparing the performance of different community-detection methods and algorithms.

We propose a benchmark for community detection in temporal multilayer networks. In contrast to single-layer community-detection benchmarks, which one can use to generate a sequence of networks with uncorrelated planted community structure, we incorporate a simple probabilistic model for the persistence of community assignments between successive layers to generate a sequence of single-layer networks with correlated planted community structure. We take advantage of the analytic tractability of our model to highlight some of its theoretical properties and we comment on the effect of some of its parameters on the resulting benchmark multilayer partitions. Finally, we perform several numerical experiments using different methods and computational heuristics on the proposed benchmark.

References

[1] M. Bazzi*, L. G. S. Jeub*, A. Arenas, S. D. Howison, and M. A. Porter. Multilayer benchmark networks for community detection. In preparation.

[2] M. Bazzi, M. A. Porter, S. Williams, M. McDonald, D. J. Fenn, and S. D. Howison. Community detection in temporal multilayer networks, with an application to correlation networks. *Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal*, 14(1):1–41, 2016.

[3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 10:P10008, 2008.

[4] U. Brandes, D. Delling, M. Gaertler, R. Göke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, 2008.

[5] P. Holme. Modern temporal network theory: A colloquium. *The European Physical Journal B*, 88(234), 2015.

[6] M. Kivelä, A. Arenas, M. Barthélemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

[7] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela. Community structure in time-dependent, multiscale and multiplex networks. *Science*, 328:876–878, 2010.

[8] M. A. Porter, J.-P. Onnela, and P. J. Mucha. Communities in networks. *Notices of the American Mathematical Society*, 56:1082–1097, 1164–1166, 2009.

OPTIMAL CONTROL OF INTERDEPENDENT EPIDEMICS IN COMPLEX NETWORKS

Juntao Chen, Rui Zhang, Quanyan Zhu

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

Optimal control of interdependent epidemics spreading over complex networks is a critical issue. We first establish a framework to capture the coupling between two epidemics, and then analyze the system's equilibrium states by categorizing them into three classes, and deriving their stability conditions. The designed control strategy globally optimizes the trade-off between the control cost and the severity of epidemics in the network. A gradient descent algorithm based on a fixed point iterative scheme is proposed to find the optimal control strategy. In addition, the control will lead to switching between equilibria of the interdependent epidemics network. Case studies are used to corroborate the theoretical results finally.

Introduction

Control of epidemics in complex networks is a prevailing problem ranging from social science to engineering [1, 2]. A network containing two interdependent epidemics with a control $\mathbf{u} := (u_1, u_2) \in \mathbb{R}_+^2$ can be described by a mean-field model similar to the one in [3]:

$$\begin{aligned}\frac{dI_{1,k}(t)}{dt} &= -\gamma_1 I_{1,k}(t) + \zeta_1 k[1 - I_{1,k}(t) \\ &\qquad - I_{2,k}(t)]\Theta_1(t) - u_1 I_{1,k}(t), \\ \frac{dI_{2,k}(t)}{dt} &= -\gamma_2 I_{2,k}(t) + \zeta_2 k[1 - I_{1,k}(t) \\ &\qquad - I_{2,k}(t)]\Theta_2(t) - u_2 I_{2,k}(t),\end{aligned} \tag{1}$$

where $I_{1,k}(t)$ and $I_{2,k}(t)$ represent the densities of nodes at time t with degree k infected by virus strain 1 and strain 2, respectively; (γ_1, γ_2) and (ζ_1, ζ_2) are recovery and spreading rates of two strains; $\Theta_1(t) = \frac{\sum_{k'} k' P(k') I_{1,k'}(t)}{\langle k \rangle}$, $\Theta_2(t) = \frac{\sum_{k'} k' P(k') I_{2,k'}(t)}{\langle k \rangle}$, where $P(k)$ is the probability distribution of a node with degree k, and $\langle k \rangle = \sum_k k P(k)$.

The network cost over a time period $[0, T]$ is captured by two terms: the control cost $c_1(\mathbf{u})$ and the severity of epidemics $c_2(\bar{I}_1(t) + \bar{I}_2(t))$, where c_1 and c_2 are both monotonically increasing functions. In addition, $\bar{I}_1(t) := \sum_k P(k) I_{1,k}(t)$ and $\bar{I}_2(t) := \sum_k P(k) I_{2,k}(t)$, and they can be interpreted as the severity of epidemics in the network. The optimal control problem at network equilibrium can be formulated as

$$\begin{aligned}(\text{OP1}): \quad &\min_{\mathbf{u}} \; c_1(\mathbf{u}) + c_2\big(\bar{I}_1^*(u_1) + \bar{I}_2^*(u_2)\big) \\ &\text{s.t.} \quad \text{system dynamics (1)},\end{aligned}$$

where $\bar{I}_1^*(u_1)$ and $\bar{I}_2^*(u_2)$ denote the densities of the strains at the steady state under the control $\mathbf{u}$. Note that (OP1) can also be interpreted as the average cost minimization problem. At the steady state, $dI_{1,k}/dt = 0$ and $dI_{2,k}/dt = 0$, and we obtain $I_{1,k} = \frac{\psi_1 k \Theta_1}{1 + \psi_1 k \Theta_1 + \psi_2 k \Theta_2}$ and $I_{2,k} = \frac{\psi_1 k \Theta_1}{1 + \psi_1 k \Theta_1 + \psi_2 k \Theta_2}$, where $\psi_i = \zeta_i / (\gamma_i + u_i)$, $i = 1, 2$. Then, the control problem (OP1) can be reformulated as

$$\begin{aligned}(\text{OP2}): \quad &\min_{\mathbf{u}} \quad c_1(\mathbf{u}) + c_2\big(\bar{I}_1^*(u_1) + \bar{I}_2^*(u_2)\big) \\ &\text{s.t.} \quad I_{i,k}^*(u_i) = \frac{\psi_i k \Theta_i^*}{1 + \psi_i k \Theta_i^* + \psi_{-i} k \Theta_{-i}^*}, \; i = 1, 2,\end{aligned}$$

where $-i = \{1, 2\} \setminus \{i\}$, $\Theta_i^* = \frac{\sum_{k'} k' P(k') I_{i,k'}^*(u_i)}{\langle k \rangle}$, and $\bar{I}_i^*(u_i) = \sum_k P(k) I_{i,k}^*(u_i)$ is the total number of nodes infected by strain i.

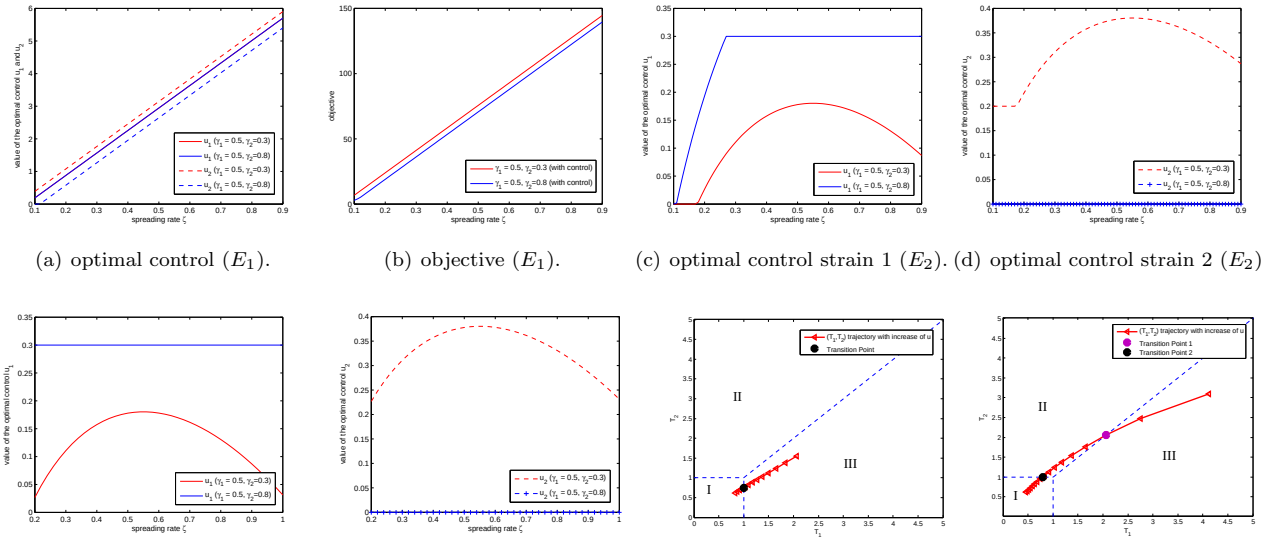
Our objective is to design a control strategy via solving (OP2) which jointly optimizes the control cost and the epidemics spreading level in the network.

Main Results

To solve (OP2), we first need to analyze the system's steady states. The equilibrium pair (Θ_1^*, Θ_2^*) needs to satisfy the following self-consistency equations for $i = 1, 2$:

$$\Theta_i = \frac{\psi_i}{\langle k \rangle} \sum_{k'} \frac{k'^2 P(k') \Theta_i}{1 + \psi_i k' \Theta_i + \psi_{-i} k' \Theta_{-i}}. \tag{2}$$

For equation (2), $(\Theta_1, \Theta_2) = (0, 0)$ is an obvious solution that results in $\bar{I}_1^* = \bar{I}_2^* = 0$ and leads to an epidemics-free equilibrium. By a closer checking of (2), we conclude that there exist no positive solutions, i.e., $\Theta_1 > 0$ and $\Theta_2 > 0$. Hence, besides the epidemics-free one, the system has another two exclusive equilibria, which lead to either entire population infected by strain 1 or by strain 2. The conditions that lead to different network equilibria are essential. Let $T_1 := \frac{\psi_1 \langle k^2 \rangle}{\langle k \rangle}$, $T_2 := \frac{\psi_2 \langle k^2 \rangle}{\langle k \rangle}$, and then, three possible equilibrium states can be summarized as follows:



(a) optimal control (E_1). (b) objective (E_1). (c) optimal control strain 1 (E_2). (d) optimal control strain 2 (E_2).

(e) optimal control strain 1 (E_3). (f) optimal control strain 2 (E_3). (g) single switching case. (h) double switching case.

Figure 1: Results of the optimal control for each equilibrium case, and the demonstration of switching of equilibrium.

(i): Epidemics-free equilibrium $\boldsymbol{E_1}$; **(ii)**: Exclusive equilibrium of strain 1, $\boldsymbol{E_2}$, if and only if $T_1 > 1$; **(iii)**: Exclusive equilibrium of strain 2, $\boldsymbol{E_3}$, if and only if $T_2 > 1$.

Stability Analysis: Through an eigenvalue analysis of the nonlinear dynamic system (1), we obtain the following results. **(i)**: E_1 is asymptotically stable if and only if $T_1 \leq 1$ and $T_2 \leq 1$. **(ii)**: E_2 is asymptotically stable if and only if $T_1 > 1$ and $T_1 > T_2$. **(iii)**: E_3 is asymptotically stable if and only if $T_2 > 1$ and $T_2 > T_1$.

Optimal Control: For each case, we can further obtain its corresponding control bounds. Then, the optimization problem (OP2) can be simplified by dividing it into three stable equilibrium cases. For example, under E_2, i.e., when $\bar{I}^*_{2,k} = 0$, (OP2) becomes $\min_{\mathbf{u}} c_1(\mathbf{u}) + c_2\big(\bar{I}^*_1(u_1)\big)$ with constraints $I^*_{1,k}(u_1) = \frac{\psi_1 k \Theta^*_1}{1+\psi_1 k \Theta^*_1}$, $u_1 < \frac{\zeta_1 \langle k^2 \rangle}{\langle k \rangle} - \gamma_1$ and $u_2 > \frac{\zeta_2(\gamma_1+u_1)}{\zeta_1} - \gamma_2$. By addressing the coupling terms $\bar{I}^*_{1,k}(u_1)$ and Θ^*_1, we obtain a fixed point equation as $\Theta^*_1 = \frac{1}{\langle k \rangle} \sum_{k'} \frac{k'^2 P(k') \psi_1 \Theta^*_1}{1+\psi_1 k' \Theta^*_1}$. We can show that there exists a unique solution Θ^*_1 to the fixed point equation, and also the mapping $u_1 \to \bar{I}^*_1(u_1)$ is continuous. The solution Θ^*_1 with respect to ψ_1 can be obtained via a fixed point iterative scheme of which the stability and convergence are guaranteed due to the contraction mapping. With obtained Θ^*_1, the objective function is only related to $\mathbf{u}$ and can be solved by the gradient descent method. Optimal control is achieved until both Θ_1 and $\mathbf{u}$ converge.

Another finding is that when the equilibrium state of the network without control is not epidemics-free, then it can switch to different states with the increase of control effort. Depending on the parameters of the epidemics, the control can lead to either single or double switching between equilibrium points (see Figs. 1(g) and 1(h)).

Numerical Experiments: Case studies based on a scale-free network are to validate the theoretical results. Specifically, the cost functions are chosen as $c_1 = 15u_1 + 10u_2$ and $c_2 = 50(\bar{I}^*_1(u) + \bar{I}^*_2(u))$. Strain 1 and strain 2 have the same spreading rate $\zeta_1 = \zeta_2$. For comparison, we have two cases: (1) $\gamma_1 = 0.5, \gamma_2 = 0.3$ and (2) $\gamma_1 = 0.5, \gamma_2 = 0.8$. For the epidemics-free case, the results are shown in Figs. 1(a) and 1(b). In addition, the results corresponding to the exclusive equilibrium of strain 1 and strain 2 are shown in Figs. 1(c) and 1(d) and Figs. 1(e) and 1(d), respectively. To demonstrate the switching of equilibria through control, we choose two cases: (1) $\zeta_1 = 0.2, \gamma_1 = 0.4, \zeta_2 = 0.15$ and $\gamma_2 = 0.4$; (2) $\zeta_1 = 0.1, \gamma_1 = 0.1, \zeta_2 = 0.15$, and $\gamma_2 = 0.2$. The obtained results are shown in Figs. 1(g) and 1(h).

References

[1] E. Hansen and T. Day. Optimal control of epidemics with limited resources. *Journal of mathematical biology*, 62(3):423–451, 2011.

[2] M. E. Newman. Spread of epidemic disease on networks. *Physical review E*, 66(1):016128, 2002.

[3] R. Pastor-Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Physical review letters*, 86(14):3200, 2001.

ANONYMIZING NETWORKS WITH B-EDGE COVERS AND B-MATCHINGS

Krzysztof Choromanski, Arif Khan, Alex Pothen, Tony Jebara (Google Research, Purdue, Columbia)

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We discuss the problem of publishing network data such that privacy requirements of the vertices can be satisfied by modifying the information in the network. We show that this problem can be formalized and solved with a b-Edge Cover or a b-Matching in a bipartite compatibility graph. Our algorithms can be implemented efficiently on multithreaded shared memory architectures, and are also space-efficient.

Introduction

We are given a network $G = (V, F, E)$, where V is a set of people, F is a set of features, and $E \subseteq V \times F$ is a set of edges, that we wish to publish so that others could experiment with the data (think about the infamous Netflix competition!). Each vertex v represents an individual for whom we have up to f feature values (binary for simplicity here). The unique identifying information (such as name, social security number, etc.) of v will be unpublished; v expresses a preference $b(v)$ for privacy, i.e., the remaining data of v should be confused with the data of at least $b(v) - 1$ other individuals in the result of any query. The problem is to suppress or add the fewest feature values while satisfying the privacy constraints.

Background

One of the most powerful techniques for privacy is differential privacy [3], which is a property of the algorithm, not the dataset itself. Hence it requires the prior knowledge of the algorithm that will be used on a dataset. However in many applications the users would like to release the raw data with some obfuscation for general exploration. There exist many other methods as k-anonymity, k-concealment, l-diversity and t-closeness [7] that are more suitable in this scenario.

A release of data has the k-anonymity property if the information for each person contained in the release cannot be distinguished from the information of at least $k-1$ other individuals. [6] showed that k-anonymity was NP-hard and presented approximation methods.

Our model is as follows. Given a network $G = (V, F, E, b)$ (recall the earlier notation), we wish to publish an anonymized network $G' = (X, F, E', b)$ such that the each vertex $v \in V$ is replaced by a key $x \in X$, and the edges in E' are obtained from E by deleting or adding edges as needed to satisfy the privacy constraints b. Given these conditions, we wish to minimize the difference between the sets E' and E.

We will say that vertex $v \in V$ is compatible with a key $x \in X$ if v can be confused with x after modifying (masking or adding) some edges incident on x. The utility of the suppression model is the number of edges changed from E to obtain E'. We can define a bipartite compatibility graph $B = (V, X, E'')$ in which an edge $(v, x) \in E''$ joins $v \in V$ with $x \in X$ with weight equal to the number of features in which they differ (dissimilarity). This represents the number of features that must be changed or suppressed so that v could be confused with key x.

There are a few papers that have employed graph matchings to solve anonymity problems. [2] employs a matching approach in which they compute a sequence of b perfect matchings in a complete bipartite graph, where the edge weights in the algorithm change after each perfect matching is computed. Their model considers both generalization (for numerical values) and suppression (for binary and categorical data), but the algorithms have high time complexities, and do not scale to massive data sets.

One drawback shared by all these approaches is that they assume a *uniform* desired privacy level k across all the users. In real life applications this is never the case, since there are conservative and liberal users who desire different levels of privacy.

Algorithm

From the suppression model, in order to provide privacy guarantees, each user v has to be in a group of $b(v)$ users, and we call this the *Grouping step*. This step is at the heart of each iteration of a variational algorithm for the problem (not described here), both in terms of solution quality and performance; it typically takes $> 90\%$ of the run time.

Our work shows that this step can be accomplished by computing a b-Edge Cover of minimum weight in the compatibility graph by choosing $b(v)$ equal to the privacy value desired by an individual v. The degree constraints on the vertices in the graph are lower bounds in the b-Edge Cover problem, and are satisfied exactly by the edge covers computed by the approximation algorithms.

Given a graph $G = (V, E)$ and a function $b(.)$ on the set of vertices, a b-Edge Cover is a subset of edges C such that at least $b(v)$ edges in C are incident on each vertex $v \in V$. When there are weights on the edges, we can compute a b-Edge Cover such that the sum of the weights of the edges is minimized. This minimum weight b-Edge Cover problem can be solved in $O(nmB)$ time, where n is the number of vertices, m is the number of edges, and B is the sum of $b(v)$ over all vertices v. However, here we employ an approximation algorithm that computes a b-Edge Cover whose weight is at most 3/2 that of the minimum, because this algorithm has $O(m\beta)$ time complexity, where β is the maximum value of $b(v)$, so that we can solve much larger problems. The approximation in the b-Edge Cover translates into a guarantee on the anonymity obtained by the algorithm.

We can also consider an approach to the Grouping step that uses a b-Matching with similarity weights in the compatibility graph. Since we seek to maximize the similarity between instances that will be grouped together, the problem is one of finding a b-Matching of maximum weight. This approach was proposed earlier by [1], using exact algorithms for b-Matching. We employ a 1/2-approximation algorithm for computing an approximate b-Matching to make the algorithm fast, but now the privacy constraints might not be satisfied for all instances; but often the number of violations are few, and we are able to add edges to the matching to satisfy all constraints.

Our method achieves strong anonymity and diversity properties, and is also robust when a few edges of the ground truth matching between vertices and keys are revealed to an adversary. To this end, we introduce a new stability property that our method satisfies.

Implementation and Performance

The 1/2-approximation algorithm for b-Matching was implemented in [4], and is called b-Suitor since it is an adaptation of the Suitor algorithm for edge weighted matching due to Manne and Halappanavar [5]. The b-Suitor algorithm can be implemented on parallel computers due to its high concurrency, and is also space-efficient since we can work with a subset of the edges bounded in size by a function linear in the number of vertices.

The approximation algorithm for computing a b-Edge Cover computes locally sub-dominant edges (an edge (u, v) of minimum weight relative to other edges incident on u and v) to add to the cover at each step. Our current results show that it takes more time (about 11 times on a set of problems) and more space (we need to store the entire graph here) than the b-Suitor algorithm.

We report results from seven problems from a Machine Learning Repository at UC Irvine, comparing our algorithm with earlier approaches for the anonymity problem. We have worked with a larger network derived from a Medicare-Medicaid data set, which has $1,000,000$ individuals in it, with 512 feature values (binary) available for each person. We randomly generated b values from 5 to 100 for each person. The information for this network would require 4 TB of memory to store. We used the space-efficient version of the b-Suitor algorithm on a Xeon processor with 20 cores and 256 GB memory to solve this problem. We are able to anonymize this data and publish more than 81% of it in under 11 hours. As far as we know, this is the largest problem on which similar anonymity techniques have been applied.

References

[1] K. M. Choromanski, T. Jebara, and K. Tang. Adaptive anonymity via b-matching. In *Advances in Neural Information Processing Systems*, pages 3192–3200, 2013.

[2] K. Doka, M. Xue, D. Tsoumakos, and P. Karras. k-anonymization by freeform generalization. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, ASIA CCS '15, Singapore, April 14-17, 2015*, pages 519–530, 2015.

[3] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[4] A. Khan, A. Pothen, M. Patwary, N. Satish, N. Sunderam, F. Manne, M. Halappanavar, and P. Dubey. Efficient approximation algorithms for weighted b-matching. *SIAM Journal on Scientific Computing*, 25 pages, 2016. To appear.

[5] F. Manne and M. Halappanavar. New effective multithreaded matching algorithms, in *IEEE 28th International Parallel and Distributed Processing Symposium*, May 2014, pp. 519–528.

[6] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *Proceedings of the Twenty-third ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 14-16, 2004, Paris, France*, pages 223–228, 2004.

[7] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

STRUCTURAL SPARSENESS AND COMPLEX NETWORKS

Erik Demaine, Felix Reidl, Peter Rossmanith, Fernando Sánchez Villaamil, Somnath Sikdar, Blair D. Sullivan.

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We identify a notion of structural sparseness that a) is exhibited by important random graph models used in network science, b) allows the design of efficient algorithms for a huge class of problems, and c) is empirically present in a large fraction of real-world networks. This demonstrates, for the first time on this scale, that the theory of structurally sparse graphs is applicable to networks stemming from real-world applications.

Introduction

During the last three decades, the theory of *structurally sparse graphs* has been revolutionised by Robertson and Seymour's graph minors programme. Many of the 'by-products' of their work, like the notions of *treewidth* and the decomposition theorem for graphs excluding a fixed minor, have had a tremendous impact on the research efficient graph algorithms.

Concurrently, the field of *complex networks* has seen a steady growth in the last decade, fuelled by an ever-growing collection of relational data that our life in the information age generates. While several structural commonalities of complex networks have been observed—*e.g.* low density, heavily skewed degree-distributions, or the small world property [11]—so far no property has been discovered that is algorithmically exploitable on a broad scale.

We work towards bringing the field of structural sparse graphs and the field of complex networks closer together. We identify two notions of structural sparseness based on the density of shallow minors as keys for this endeavour: classes of *bounded expansion* and *nowhere dense* classes as introduced by Nešetřil and Ossona de Mendez in their seminal work on a robust theory of sparseness [10]. We demonstrate that these sparse classes admit efficient algorithms for a huge number of problems, some of which have applications in domain-specific areas of network science. We further prove that several fundamental network models exhibit these properties and demonstrate empirically that this also holds true for a selection of real-world networks from various domains.

Important concepts

Given a graph G and H, we say H is an *r-shallow topological minor* of G if we can turn H into a subgraph of G by replacing the edges of H by paths of length at most $(2r+1)$.[1] For integers r, we denote by $G \,\widetilde{\nabla}\, r$ the collection of all graphs that are r-shallow topological minors of G. The *topological grad* now measures the density of these minors as a function of the depth r:

$$\widetilde{\nabla}_r(G) = \max_{H \in G \,\widetilde{\nabla}\, r} \frac{\|H\|}{|H|},$$

where $\|H\|$ and $|H|$ denote the number of edges and vertices of H, respectively. An infinite collection of graphs $\mathcal{G}$ (a *graph class*) has *bounded expansion* if there exists a function f such that

$$\widetilde{\nabla}_r(\mathcal{G}) = \sup_{G \in \mathcal{G}} \widetilde{\nabla}_r(G) \leqslant f(r).$$

That is: the maximum density of r-shallow minors is a function of their depth r, and in particular independent of the size of the graph they occur in.

The concept of *nowhere dense* graphs is defined similarly, but instead of measuring the density of r-shallow topological minors we measure their clique number. Every bounded expansion class is in particular nowhere dense and we will call classes with either property simply *structurally sparse*. This is motivated by a dichotomy result proved by Nešetřil and Ossona de Mendez [10] (based on an important result by Dvořák [4]) that states that graph classes can be categorised rigorously into *somewhere dense* and *nowhere dense* classes.

Results

We prove that both the Chung–Lu [2] and the configuration model [9] exhibit a phase-transition regarding their structural density which crucially depends on the tail of the input degree distribution. Formally:

[1] This somewhat arbitrary seeming term is a sensible choice in the broader context of the theory since it provides comparability with the related notion of *r-shallow minors*.

Theorem 1. *Let (D_n) be a sparse degree distribution sequence with tail $h(d)$. Both the configuration model $G^{CF}(D_n)$ and the Chung–Lu model $G^{CL}(D_n)$, with high probability,*

- *have bounded expansion for $h(d) = \Omega(d^{3+\varepsilon})$,*
- *are nowhere dense (with unbounded expansion) for $h(d) = \Theta(d^{3+o(1)})$,*
- *and are somewhere dense for $h(d) = O(d^{3-\varepsilon})$.*

Based on this, we show that perturbations of bounded degree graphs, which can be seen as a baseline model for percolation-type random graphs, exhibit structural sparseness. Surprisingly, this does not hold for Kleinberg's model [8]—the geographic dependence of random edges produces dense shallow structures with high probability.

To relate this result on network models to real-world instances, Felix Reidl [12] showed that a large fraction of our real-world network corpus a) follows degree distributions whose tail is best described as supercubic (vanishing faster than d^{-3}) and b) has a structural density that is lower than predicted by a random graph sampled with the same degree distribution. He established the former using statistical tests developed to distinguish pure power-law distributions from non-heavy-tailed distributions [3, 1] and, in line with this previous work, found that most of our networks present degree distributions which are better described as function with quickly vanishing tails (*e.g.* log-normal or power laws with exponential cutoff). To demonstrate the second part, he engineered a known, theoretical algorithm to make it applicable in practice. The algorithm measures a certain proxy-value (the indegree of so-called dtf-augmentations) that equivalently captures structural sparseness and turns out to be more efficiently computable than proxy-values we considered previously.

The algorithmic *usefulness* of this result is established theoretically by known algorithmic meta-theorems [5, 7, 6]. For more concrete applications, we design a linear-time algorithm to count graph motifs with a superior running time as well as a procedure to determine the sizes of local neighbourhoods which enables fast centrality estimation.

In conclusion, we can state that the theory of structurally sparse graphs is applicable to complex networks and, as a corollary, so is the rich algorithmic toolkit it provides. This connection offers researchers from both the field of algorithmic graph theory and network science new approaches, insights, and productive questions.

References

[1] J. Alstott, E. Bullmore, and D. Plenz. powerlaw: A python package for analysis of heavy-tailed distributions. *PLoS ONE*, 9(1):e85777, 01 2014.

[2] F. Chung and L. Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 99(25):15879–15882, 2002.

[3] A. Clauset, C. Shalizi, and M. Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

[4] Z. Dvořák. *Asymptotical Structure of Combinatorial Objects*. PhD thesis, Charles University, Faculty of Mathematics and Physics, 2007.

[5] Z. Dvořák, D. Král, and R. Thomas. Deciding first-order properties for sparse graphs. In *Proc. of 51st FOCS*, pages 133–142. IEEE Computer Society, 2010.

[6] J. Gajarský, P. Hliněný, J. Obdržálek, S. Ordyniak, F. Reidl, P. Rossmanith, F. Sánchez Villaamil, and S. Sikdar. Kernelization using structural parameters on sparse graph classes. *To appear in Journal of Computer and System Sciences*, 2016.

[7] M. Grohe, S. Kreutzer, and S. Siebertz. Deciding first-order properties of nowhere dense graphs. In *Proc. of 46th STOC*, pages 89–98, 2014.

[8] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proc. of 32rd STOC*, pages 163–170, 2000.

[9] M. Molloy and B. A. Reed. A critical point for random graphs with a given degree sequence. *Random Structures & Algorithms*, 6(2/3):161–180, 1995.

[10] J. Nešetřil and P. Ossona de Mendez. *Sparsity: Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and Combinatorics*. Springer, 2012.

[11] M. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[12] F. Reidl. *Structural sparseness and complex networks*. Dr., Aachen, Techn. Hochsch., Aachen, 2016. Aachen, Techn. Hochsch., Diss., 2015.

NEW STOPPING CRITERIA FOR SPECTRAL PARTITIONING (EXTENDED ABSTRACT)

James P. Fairbanks, Anita Zakrzewska, and David A. Bader

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Abstract

Spectral partitioning (clustering) algorithms use eigenvectors to understand structure in networks. We study spectral partitioning using sweep cuts of approximate eigenvectors of the normalized graph Laplacian. We introduce a novel, theoretically sound, parameter free stopping criterion for iterative eigensolvers designed for graph partitioning and experimentally validate it on real world networks.

Introduction and Related Work

Large graphs are found in many domains and finding good partitions for these graphs is a challenging data mining task. Spectral partitioning using approximate eigenvectors fits into a larger framework of studying how knowledge of the network analysis task can shape our choice of numerical procedure. Iterative methods need good stopping criteria to ensure a high quality solution is found in as little time as possible. In Theorem 1, we analyze eigenvalue accuracy in the context of spectral partitioning to derive a condition on approximate eigenvectors that provides the same theoretical guarantees as sweep cuts of the exact eigenvectors. This new stopping criterion based on convergence is the first that does not require the implementation to specify an error or residual tolerance. Unlike prior methods, the user need not choose an acceptable amount of error, which simplifies practical application of this method. Experiments show that our stopping criterion reduces the number of iterations compared to traditional stopping criteria on real world networks with only a small increase in the final conductance.

In spectral partitioning, vectors approximating some eigenvectors of a graph matrix are computed then used to cut the graph. The runtime/accuracy trade-off in the eigensolver step is rarely considered [3]. Iterative methods for solving the eigenvector problem $A\mathbf{x} = \lambda\mathbf{x}$ such as the implicitly restarted Arnoldi method (IRAM) can generate solutions to arbitrary approximation factors by increasing the number of iterations [5]. Power method approximations to the eigenvectors of a kernel matrix approximate the k-means objective function well [1].

Definitions and Notation

Notation for linear algebra terms are presented Table 1.

Name	Symbol	Definition
Adjacency Matrix	A	$a_{ij} = i \sim j \ ? \ 1 : 0$
Degree Matrix	D	$d_{ii} = (A\mathbf{1})_i$
Normalized Adjacency	$\hat{A}$	$D^{-1/2}AD^{-1/2}$
Laplacian Matrix	L	$D - A$
Normalized Laplacian	$\hat{L}$	$I - \hat{A}$
Eigendecomposition	Q, Λ	$\hat{L} = Q^T \Lambda Q$
Eigenpairs	$\mathbf{q}, \lambda_i$	$\hat{L}\mathbf{q} = \lambda_i \mathbf{q}$
Rayleigh Quotient	μ	$\frac{\mathbf{x}^T \hat{L}\mathbf{x}}{\mathbf{x}^T\mathbf{x}}$
Residual	r	$\left\|\hat{L}\mathbf{x} - \mu\mathbf{x}\right\|$

Table 1: Linear algebra notation.

Since general eigenproblems cannot be solved exactly, the residual, which is related to error [4], evaluates the quality of a solution. Iterations are taken until r is less than a prescribed, user-chosen tolerance. A cut is represented by a subset of the vertices S and $\bar{S} = V \setminus S$ the complement. Define $\mathrm{vol}(S) = \sum_{i,j \in S} a_{i,j}$ as the total weight of the edges within S. We measure the quality of a cut of the graph using conductance $\phi(S)$ defined as [2]:

$$\phi(S) = \frac{\sum_{i \in S j \notin S} a_{i,j}}{\min(\mathrm{vol}(S), \mathrm{vol}(\bar{S}))}$$

Sweep cuts of a vector $\mathbf{x}$ have the form $S_x^t = \{i \mid x_i > x_t\}$. The conductance of a vector is the conductance of the minimal sweep cut of that vector, i.e. $\phi(\mathbf{x}) = \min_t \phi(S_x^t)$. This work studies algorithms using sweep cuts of an approximate eigenvector to partition the graph.

Eigenvalue accuracy and Cheeger's inequality

The minimum conductance cut problem can be relaxed to $\min_{\mathbf{x} \perp \mathbf{1}} \frac{\mathbf{x}^T L\mathbf{x}}{\mathbf{x}^T D\mathbf{x}}$, which is solved by solutions to the generalized eigenvalue problem $L\mathbf{x} = \lambda D\mathbf{x}$. A pair $\lambda, \mathbf{y}$ solves the generalized eigenequation $L\mathbf{y} = \lambda D\mathbf{y}$, if and only if the pair $\lambda, \mathbf{x} = D^{-\frac{1}{2}}\mathbf{y}$ solves the eigenequation $\hat{L}\mathbf{x} = \lambda\mathbf{x}$. Computational tools for symmetric eigenvalue problems can solve $\hat{L}\mathbf{x} = \lambda\mathbf{x}$. Cheeger's inequality [2] guarantees that exact eigenvectors provide a sweep cut with conductance less than $\sqrt{2\lambda_2}$.

Our main contribution follows from observing that any partition with conductance less than $\sqrt{2\lambda_2}$ satisfies the same guarantee provided by an exact eigenvector. When running the solver, the true value of λ_2 is unavailable for use in a stopping criterion. Theorem 1 provides a stopping

criterion that satisfies the same guarantee as the exact solution.

Theorem 1. *Let $\hat{L}, \mathbf{x}, \mathbf{y}, \mu, r$ be as above. If $\mu - \lambda_2 < |\mu - \lambda|$ for all other eigenvalues λ, then $\phi(\mathbf{y}) < \sqrt{2(\mu - r)} = \psi(\mathbf{x})$ guarantees $\phi(\mathbf{y}) < \sqrt{2\lambda_2}$.*

Proof. For any $\epsilon > 0$, $\mu - \epsilon < \lambda_2$ implies $\sqrt{2(\mu - \epsilon)} < \sqrt{2\lambda_2}$. Using the eigendecomposition of $L = Q\Lambda Q^T$, let $\mathbf{z} = Q^T\mathbf{x}$. Since $\mathbf{x} \perp \mathbf{q}_1$, $z_1 = 0$. From the hypothesis that $|\mu - \lambda_2|$ is minimal, one sees

$$r^2 = \|(\Lambda - \mu I)\mathbf{z}\|^2 > (\lambda_2 - \mu)^2 \sum_i {z_i}^2 = (\lambda_2 - \mu)^2.$$

So $r > |\lambda_2 - \mu|$, and $\mu - r < \lambda_2$. Thus under these conditions $\phi(\mathbf{y}) \leq \sqrt{2(\mu - r)} < \sqrt{2\lambda_2}$. □

Experiments

We compare the criterion in Theorem 1 to the residual based stopping criterion with a tolerance of $r < 10^{-6}$, or a maximum of 800 iterations. The IRAM restart parameter of 15 and maximum number of iterations are chosen to balance time and memory constraints. We exploit the fact that eigenvalues of $\hat{L}$ are one minus the eigenvalues of A with the same eigenvectors to iterate with $M = \hat{A} - \mathbf{q}_1\mathbf{q}_1^T$. Experiments are conducted on matrices from the Newman [7] and the SNAP [6] collections[1]. These problems range from the small, well conditioned N/lesmis to the large, ill conditioned S/web-Google.

For each iterate ϕ, μ, and r are computed to evaluate $\phi(\mathbf{y}) < \psi(\mathbf{x})$ and $r < 10^{-6}$ Let I_F, I_C denote the number of eigensolver iterations according to the residual and conductance tolerances respectively[2]. We compare for each graph the conductance of the sweep cut when stopping at I_C, denoted ϕ_C, to the conductance at I_F, denoted ϕ_F. The iteration ration I_F/I_C quantifies the improvement in iteration count due to our method. The distribution of iteration ratios is shown in Figure 1. For most graphs this ratio is at least 2 and for at least 2 graphs this ratio is greater than 12.

For each graph, stopping at I_C results in a conductance less than five times the final conductance. On average across all graphs the conductance resulting from our approach is only 1.24 times greater. Our method reduces the iteration count by a factor of 4.15 and typically yields partitions outperforming the guarantee by at least a factor of $\lambda/5$.

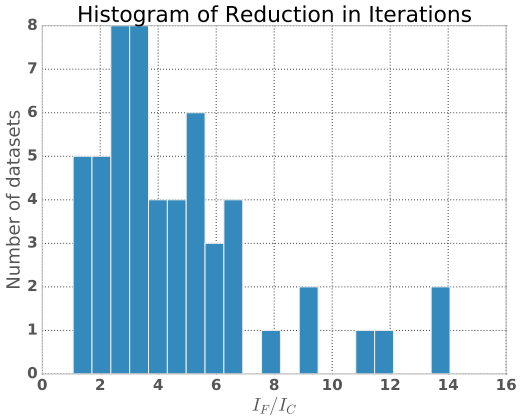


Figure 1: Distribution of relative iteration counts.

Conclusions

We show that analysis of both numerical accuracy and the network analysis application leads to an improved parameter free stopping criterion. On real world networks, this leads to a large reduction in the number of iterations used to solve this data mining problem with a small increase in conductance. For some graphs our method results in a smaller conductance. Since these problems take hours to solve, this reduction in running time is meaningful. Furthermore, the theory deepens our understanding of the relationship between accuracy of numerical solutions and quality of network analysis.

Acknowledgments

The authors thank Geoffrey D. Sanders and the NDSEG Fellowship for their support.

References

[1] C. Boutsidis, A. Gittens, and A. Kambandur. Spectral clustering via the power method - provably. In *Proc. of The 32nd International Conference on Machine Learning*, pages 40–48, 2015.

[2] F. R. Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[3] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51(3):497–515, 2004.

[4] T. Kato. On the upper and lower bounds of eigenvalues. *Journal of the Physical Society of Japan*, 4(4-6):334–339, 1949.

[5] R. Lehoucq and D. Sorensen. Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM Journal on Matrix Analysis and Applications*, 17(4):789–821, 1996.

[6] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[7] M. Newman. Newman Datasets. http://www-personal.umich.edu/~mejn/netdata/, April 2013.

[1] Adjacency matrices are made symmetric by taking $A + A^T$ and restricted to the largest connected component of each graph.

[2] For 34 of the graphs, the first time $\phi(\mathbf{x}) < \psi(\mathbf{x})$, the hypothesis of Theorem 1 is not satisfied, but by taking one more step this number drops to 17, decreasing the average conductance we find. We use the iteration after $\phi(\mathbf{x}) < \psi(\mathbf{x})$ as I_C.

QUANTIFYING UNCERTAINTY IN NETWORK REGRESSIONS

Bailey K. Fosdick, Tyler H. McCormick, Frank W. Marrs

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

Social networks are widely used to understand complex social phenomena. In this work we consider inference on regression coefficients in network regression models, where the strength of a connection between each pair of individuals is modeled as a linear function of observable node and edge covariates and structured, network dependent, error. We leverage a joint exchangeability assumption, nearly ubiquitous in the statistics literature on networks but not previously considered in the estimating equations formulation for network regressions, to derive parsimonious estimators of the covariance between relations. These estimates are shown to dramatically improve inference for the regression coefficients.

Introduction to network regression

Many social science questions are focused on the relationship between covariates of the actors, or nodes, and the relations. Contrary to the case for traditional network tools, here we assume the network relations are continuous measures of the strength of the relationship between individuals, e.g. amount of time school children play together, trade flow between countries. Researchers often build regression models to quantify the amount of variability that can be explained by the node and edge covariates. A network regression model can be expressed

$$y_{ij} = \mathbf{x}_{ij}^T\beta + \xi_{ij}, \tag{1}$$

where y_{ij} denotes the directed, continuous relationship from individual i to individual j. The $p \times 1$ vector $\mathbf{x}_{ij}$ contains covariates of the pair (i, j), ξ_{ij} represents unknown random error, and β is a vector of unknown coefficients. A core statistical challenge for inference is accounting for the complex dependencies among the error terms ξ_{ij}.

Existing approaches to capturing such dependencies can be characterized into two broad classes. The first approach, common in the statistics literature, attempts to model network dependence explicitly using, for example, latent variables models. The second approach, common in the economics literature, stems from estimating equations/moment conditions. These methods define a system of equations, known as estimating equations, that relate the parameters to data. The estimators, known as m-estimators, are the zeros of these estimating equations. An estimating equation $g(\cdot)$ is defined such that for all (i, j), $E\left(g(y_{ij}, \beta)\right) = 0$. The estimator $\hat{\beta}$ is then that which satisfies

$$\sum_{i,j} g(y_{ij}, \hat{\beta}) = 0. \tag{2}$$

Consider the model defined in (1) for continuous relations. A common $g(\cdot)$ is

$$g(y_{ij}, \beta) = \mathbf{x}_{ij}(y_{ij} - \mathbf{x}_{ij}^T\beta). \tag{3}$$

This corresponds to the score function of the multivariate normal likelihood with homoskedastic, independent errors and gives rise to the ordinary least squares estimate of β.

Under regularity conditions (e.g. [5]) and independence between directed pairs, the estimator satisfying (2) is consistent ($\hat{\beta} \to_p \beta$) and asymptotically normal

$$\sqrt{n}(\hat{\beta}_n - \beta) \to_d N\left(0, A^{-1}B(A^T)^{-1}\right), \tag{4}$$

where $A = E\left[\frac{\partial}{\partial\beta^T}G(Y, \beta)\right]$ and $B = E[G(Y, \beta)G(Y, \beta)^T]$ such that $G(Y, \beta)$ is the $p \times (n^2 - n)$ matrix of estimating equations. This asymptotic covariance estimator is commonly referred to as a "sandwich" estimator [6]. For $g(\cdot)$ as in (3), the A and B matrices are estimated by $\hat{A} = X^TX$ and $\hat{B} = X^T\hat{\Omega}X$ where $\Omega = \text{Cov}[\text{Y}_\text{v}]$ is the covariance matrix of relations. In contrast to the statistical approaches, which exert substantial effort in modeling the dependence among the errors explicitly using parametric latent structures, this second approach is model agnostic relying on empirical estimates of the dependence from the residuals $e_{ij} = y_{ij} - \mathbf{x}_{ij}^T\hat{\beta}$ to estimate Ω.

Dyadic clustering standard error estimators

Consider a directed pair (i, j) and define Θ_{ij} as the set consisting of all other directed pairs that contain an overlapping member with the pair (i, j). Generalizing the standard estimating equation framework, [3] and [2] propose and describe a standard error estimator which assumes

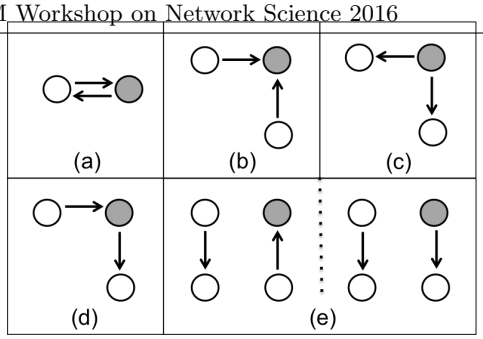


Figure 1: Five configurations of edge directed pairs involving the shaded node.

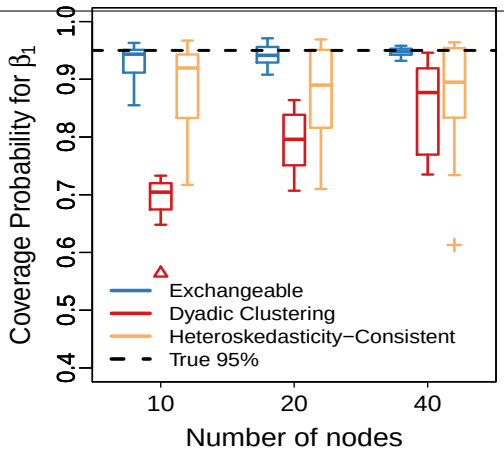


Figure 2: Probability regression coefficient is in 95% confidence interval.

relations for directed pairs (i,j) and (k,l) are independent, i.e. $\text{Cov}[\text{y}_{\text{ij}}, \text{y}_{\text{kl}}] = 0$, if (i,j) and (k,l) only if they *do not* share a member (i.e. $(k,l) \notin \Theta_{ij}$). For continuous relations, the asymptotic variance estimator has the form of that in (4) where Ω is estimated by

$$\hat{\Omega}_{DC} = \{\mathbf{e}\mathbf{e}^T \circ \mathbf{1}_{[\{i,j\}\cap\{k,l\}\neq\emptyset]}\}, \tag{5}$$

where $\mathbf{e} = \{e_{ij} : i \neq j\}$ is the vector of residuals, $\mathbf{1}_{[\{i,j\}\cap\{k,l\}\neq\emptyset]}$ is a matrix of indicators, and $\circ$ denotes the Hadamard product. The indicator matrix systematically introduces zeros to enforce the non-overlapping pair independence assumption. Note that the DC estimator estimates $\mathcal{O}(n^3)$ covariance parameters from only $\mathcal{O}(n^2)$ observed residuals. Furthermore, each non-zero covariance element independently with a *single residual product*.

Exchangeable Standard Error Estimator

A foundational property in the statistical literature on networks is exchangeability. Intuitively, exchangeability means the ordering of the network row and column labeling is uninformative. More formally, a probability distribution $\mathbb{P}(\cdot)$ on the network $Y = \{y_{ij}\}$ is jointly exchangeable if $\mathbb{P}(Y) = \mathbb{P}(\Pi(Y))$, where $\Pi(Y) = \{y_{\pi(i)\pi(j)}\}$ is the network Y with its rows and columns reordered according to permutation operator π [1].

Figure 1 shows the five distinguishable configurations of edge pairs under exchangeability involving a single node. (Not shown is the variance, meaning the covariance structure has six types of relations.) Exchangeability implies that the covariance between all edge pairs that have the same configuration in Figure 1 are equal. Thus, rather than estimating each non-zero covariance term in Ω separately, we propose averaging the residual products over the six equivalence classes given by the configurations. For example, we estimate

- $\widehat{\text{Var}}[\xi_{ij}] = \frac{1}{n(n-1)} \sum_i \sum_j e_{ij}^2$
- $\widehat{\text{Cov}}[\xi_{ij}, \xi_{ji}] = \frac{1}{n(n-1)} \sum_i \sum_j e_{ij} e_{ji}$

Estimators for the other four terms follow analogously.

Simulation results

Figure 2 shows the confidence interval coverage probability for a regression coefficient when data are generated from the following (exchangeable) network model [4]:

$$y_{ij} = \beta_0 + \beta_1 x_{ij} + \xi_{ij}, \qquad \xi_{ij} = a_i + b_j + z_i^T z_j + e_{ij}$$

where $z_i^T = (z_i^{(1)}, z_i^{(2)})$, $\{a_i, b_j, z_i^{(1)}, z_i^{(2)}, e_{ij}\}$ are independent standard normal random variables, and x_{ij} is a binary covariate. We see that our proposed estimator based on the exchangeability assumption vastly outperforms both the dyadic clustering estimators and heteroskedasticity-consistent estimators.

References

[1] D. J. Aldous. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598, 1981.

[2] P. M. Aronow, C. Samii, and V. A. Assenova. Cluster–robust variance estimation for dyadic data. *Political Analysis*, 23(4):564–577, 2015.

[3] M. Fafchamps and F. Gubert. The formation of risk sharing networks. *Journal of Development Economics*, 83(2):326–350, 2007.

[4] P. D. Hoff. Bilinear mixed-effects models for dyadic data. *Journal of the American Statistical Association*, 100(469):286–295, 2005.

[5] A. W. Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge University Press, 2000.

[6] H. White. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica: Journal of the Econometric Society*, pages 817–838, 1980.

COMPETING SPREADING PROCESSES ON MULTIPLEX NETWORKS: AWARENESS AND EPIDEMICS

C. Granell, S. Gómez, A. Arenas

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

When a contagious disease spreads in a population, information about the presence of this disease begins to spread in the population, and it is known that having access to this information (being *aware* of the disease) often leads to individuals changing habits and taking measures to prevent infection. In this work we study the interplay between the spreading of awareness and the spreading of a contagious disease. To represent these two competing processes we make use of a multiplex network, an appropriate representation for dynamics that take place in different topologies but that interact with each other. We wish to discover to what extent the awareness dynamics are able to change the outcome of the disease spreading process, assuming that an aware individual is able to reduce its chances to get infected. We also assess the effect of mass media awareness campaigns on the final outcome of the epidemics.

The generalized UAU-SIS model

In a recent work, the authors investigated the interplay between awareness and epidemic spreading in multiplex networks [2]. This scenario is represented by means of a two-layer multiplex network, where a layer represents the network of physical contacts and the other accounts for the network of information exchange. On the first, we assimilate a Susceptible-Infected-Susceptible (SIS) process, accounting for the contagious disease spreading dynamics. In this process, when an infected individual meets a susceptible one, the latter becomes infected according to the infectivity probability β. Also, infected individuals recover spontaneously at a rate μ. On the second layer, we incorporate what we call an Unaware-Aware-Unaware (UAU) process, the equivalent version of an SIS for the case of awareness spreading dynamics, where the awareness spreading rate and the recovery rate are λ and δ, respectively. In this first model we are making two assumptions: infection of the epidemics implies immediate awareness and awareness implies total immunization of the epidemics.

The generalized model proposed in this work (see Fig. 1) relaxes these two strong assumptions. The two parameters, self-awareness and degree of immunization, are regulated by probabilities κ and γ respectively. Now, this model also takes into account the effect of massive awareness information flowing through the network, the *mass media effect*. In our model, an external node representing the mass media (TV, radio, newspapers, etc.) is connected to all nodes in the information layer, regularly converting new aware individuals at a rate m.

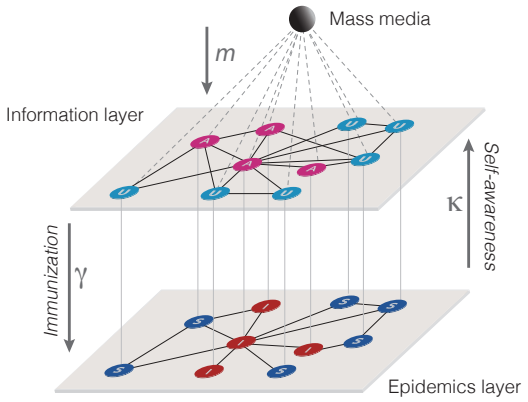


Figure 1: Sketch of the UAU-SIS model. Nodes in the awareness layer (top) may be Aware or Unaware, while nodes in the epidemic spreading layer (bottom) may be Susceptible or Infected. The *self-awareness* and *immunization* parameters regulate the interaction between the two dynamic processes. Mass media is connected to all nodes in top layer.

Results

It is possible to discover and solve the dynamical equations governing the previous system. We use the Microscopic Markov Chain Approach (MMCA) equations [1], which express the probability of a node being in each state at the current time as a function of its state in the previous time step (equations are omitted in this text because of spatial constraints, but we encourage the reader to check them in

[3]). Solving iteratively the previous system of equations we can track the time evolution of the awareness and the epidemics for any initial condition. Moreover, interestingly, we can solve the stationary state of the full system, and determine the onset of the epidemics as a function of the rest of the parameters of the model (see [3]).

Interestingly, we show that the onset of the epidemics does not always depend on the awareness spreading probability λ. Instead, there is a sort of *metacritical* point defined by the awareness dynamics and the topology of the information network, from which the onset increases and the epidemics incidence decreases (see Fig. 2).

Additionally, the results of the analysis reveal that, while the self-awareness parameter does not change the epidemic threshold nor does it significantly change the final fraction of infected nodes, the other two parameters are crucial. The immunization parameter γ is a key factor on delaying the onset of the epidemics as well as lowering the final fraction of infected nodes. Also the mass media parameter has a crucial effect: when the mass media is active ($m > 0$), the metacritical point vanishes (see Fig. 3).

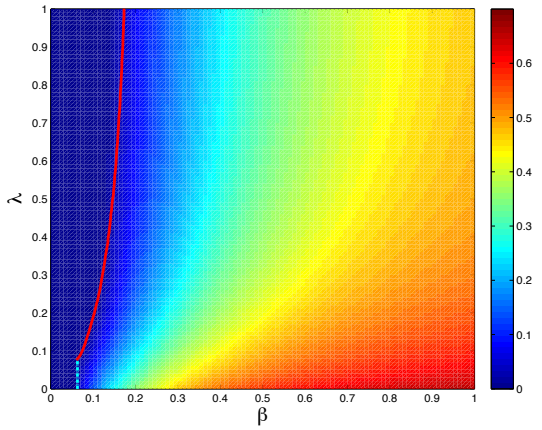


Figure 2: Full phase diagram λ-β for the multiplex setup proposed in this model, solved using MMCA equations. The multiplex network we use is the following: the bottom layer corresponding to the physical contacts network is a power-law degree distribution network generated with the configurational model with an exponent of 2.5 and a size of 1000 nodes. The top layer representing the information contacts is the same network with 400 additional (non overlapping with previous) links. Cooler colors represent a low fraction of infected individuals while warmer colors represent the opposite. The red line denotes the line of critical points where β_c depends on the awareness spreading probability λ, while the discontinuous cyan line denotes the values of β_c independent of λ. The junction between the two lines denotes the *meta-critical* point.

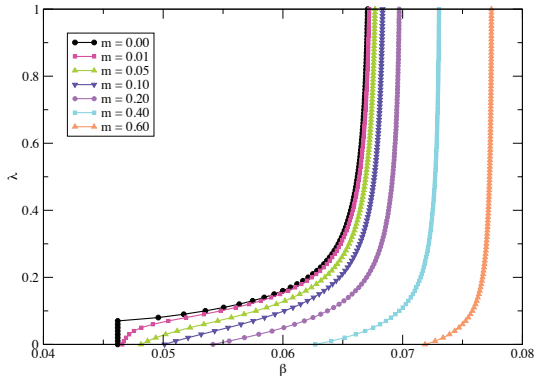


Figure 3: Representation of the line of critical points β_c as a function of λ, for different values of the mass media. We observe that for $m > 0$ the metacritical point vanishes. The multiplex structure is the same as described in Fig. 2, with $\delta = 0.6$, $\mu = 0.4$ and parameters $\kappa = 1.0$ and $\gamma = 0.0$ which imply maximum coupling between layers.

Discussion

We have presented an extended analysis of a generalization of a model of competing spreading processes on multiplex networks. The results show that the coexistence of different topologies spreading antagonistic effects raises interesting physical phenomena, as for example the emergence of a metacritical point, where the diffusion of awareness is able to control the onset of the epidemics. Given the specific nature of the awareness spreading proposed here, which is equivalent to a SIS process, the results are also valid to describe two competing infectious strains coexisting in a multiplex structure, the only difference being if the strains reinforce or weaken each other. Results reveal that while the self-awareness has almost no effect on the dynamics, the other two factors, namely the degree of immunization of aware individuals and the mass media, do change the critical aspects of the epidemics spreading.

References

[1] S. Gómez, A. Arenas, J. Borge-Holthoefer, S. Meloni, and Y. Moreno. Discrete-time Markov chain approach to contact-based disease spreading in complex networks. *Europhys. Lett.*, 89(3):38009, Feb. 2010.

[2] C. Granell, S. Gómez, and A. Arenas. Dynamical interplay between awareness and epidemic spreading in multiplex networks. *Phys. Rev. Lett.*, 111:128701, Sep 2013.

[3] C. Granell, S. Gómez, and A. Arenas. Competing spreading processes on multiplex networks: Awareness and epidemics. *Phys. Rev. E*, 90:012808, Jul 2014.

CENTRALITY ANALYSIS FOR WATTS-STROGATZ STYLE SMALL WORLD NETWORKS

Desmond J. Higham, Martin Paton, Kerem Akartunali, University of Strathclyde, UK

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We derive new, exact expressions for network centrality vectors associated with classical Watts-Strogatz style "ring plus shortcut" networks. We also derive easy-to-interpret approximations that are accurate in the large network limit. The analysis helps us to understand the role of the Katz parameter, to compare linear system and eigenvalue based centrality measures, and to predict the behavior of centrality measures on more complicated networks.

Centrality

Algorithms that quantify the importance of nodes in a network have proved extremely useful in a range of applications [2]. For example, Katz centrality [3] assigns the value $x_i \geq 0$ to node i, with a larger value indicating greater centrality, according to

$$(I - \alpha A)x = \mathbf{1}. \tag{1}$$

Here, $A \in \mathbb{R}^{N\times N}$ denotes the adjacency matrix of the network, which we assume to be unweighted and connected, $\mathbf{1} \in \mathbb{R}^{N\times N}$ denotes the vector of ones, and $\alpha > 0$ is a free parameter. Typically, α is assigned a value below the reciprocal of the spectral radius, $\rho(A)$, and several authors have suggested particular choices [1].

A Ring plus a Shortcut

Our work treats deterministic versions of the types of network introduced in the seminal small-world paper of Watts and Strogatz [5]. Because of space considerations, we describe here the simplest case of a nearest neighbor undirected periodic ring plus a single directed shortcut. Without loss of generality we assume that the extra shortcut edge points from node 1 to node L. Hence our adjacency matrix A in (1) has the form $A = C + E$, where the circulant matrix C has ones above and below the diagonal and in its upper right and lower left corners, and E is zero except for $E(1, L) = 1$. Liu, Strang and Ott [4] describe this as a *modification* of C, to emphasize that we have an $O(1)$ change in a matrix entry, rather than the type of small change studied in classical matrix perturbation theory. These authors studied the eigenvector associated with the dominant eigenvalue of A, and related matrices, and constructed accurate approximations to this vector.

Our work is strongly motivated by [4] but differs from it in four respects.

- Rather than deriving small residual approximations and then using stability arguments to bound the forward error, we construct exact solutions that can be expanded asymptotically. This more direct route leads to shorter proofs and sharper bounds.
- We consider Katz centrality (as well as the eigenvalue problem).
- We analyze more general lattice structures and modifications.
- We interpret the results from a network science perspective and use them to get new insights about behavior on more complicated networks.

Computational Illustration

In the upper picture of Figure 1, the asterisks show Katz centrality values; that is, components of x from (1). We chose a small network size in order to make the key effects visible. More precisely, we used an $N = 20$ node ring with a shortcut from node 1 to node $L = 8$, and with $\alpha = 0.3$. Because node 1 owns the extra, long-range edge, it attains the highest centrality score, at around 3.5. The most distant node, periodically, that is, node 10, is deemed the least central. Insight from [4], or from eyeballing the solution, suggests that components of x_i, when suitably shifted, might be varying geometrically as the index i moves periodically around the ring. Inserting an ansatz of this form leads us to the circles in the upper picture of Figure 1. The agreement is close—below 2×10^{-5} in Euclidean norm.

The lower picture in Figure 1 shows, on a log scale, the discrepancy between those asterisks (true solution) and circles (geometric decay ansatz). We see a very small contribution that, in contrast to the overall solution, *grows* geometrically as we move periodically away from node 1.

An Example Theorem

We now state a theorem that fully explains Figure 1. We can assume without loss of generality that the receiving node L is not beyond the half way, or "six o'clock", position on the ring. Letting $\lfloor \cdot \rfloor$ denote the integer part, for some fixed proportion $0 < \theta \leq 1$ we set $L = \lfloor \theta(N/2+1) \rfloor$ when N is even and $L = \lfloor \theta(N+1)/2 \rfloor$ when N is odd.

We assume that a fixed Katz parameter α is used, with $0 < \alpha < 1/2$. (The spectral radius for the underlying ring is 2.)

For convenience, we let $p(i)$ for any $1 \leq i \leq N$ denote the periodic distance from node i to node 1, that is, $p(i) = \min(i-1, N-i+1)$. We may then state the following result concerning the asymptotic limit where $N \to \infty$ (and hence $L \to \infty$) with α, θ fixed.

Theorem 1 *For the "undirected ring plus directed shortcut" network relating to Figure 1, the Katz system (1) has a unique solution of the form*

$$x_i = C + h_1 t_1^{p(i)} + h_2 t_2^{p(i)}. \tag{2}$$

Here, C, t_1, t_2, h_1, h_2 are constants, i.e., independent of i. In particular, $C = 1/(1-2\alpha)$ and t_1, t_2 are the roots of the quadratic $\alpha t^2 - t + \alpha$, so that

$$t_1 = \frac{1-\sqrt{1-4\alpha^2}}{2\alpha}, \quad t_2 = \frac{1+\sqrt{1-4\alpha^2}}{2\alpha}.$$

Hence, $t_2 = 1/t_1$ and $0 < t_1 < 1 < t_2$. Moreover, the final term in (2) is exponentially small asymptotically, in the sense that for all $1 \leq i \leq N$,

$$x_i = C + h_1 t_1^{p(i)} + O(t_1^{N/2}), \tag{3}$$

with $h_1 = O(1)$.

Extensions and Implications

In the example considered here it is intuitively obvious that node 1 will be assigned the highest centrality value, and that centrality will decay as we move periodically away. However, the type of analytical result in Theorem 1 allows us to see precisely how the measure depends on the Katz parameter, α. Moreover, we will show that the same analytical techniques apply to other, related networks, including cases where the conclusions are not straightforward: rewiring instead of adding shortcuts, undirected rather than directed modifications, paths rather than rings, k-neighbor connectivity, multiple long-range edges, and various types of surgically constructed combinations of these. In particular, we can devise and rigorously analyze networks where several nodes compete for the best centrality values and their overall ranking changes as α varies. We will also show that analogous results may be obtained when centrality is quantified in an alternative eigenvector-based sense [2]. In this way, by varying α between 0 and $1/\rho(A)$ we may compare degree, Katz and eigenvector centralities, shedding light on how these three widely used measures differ.

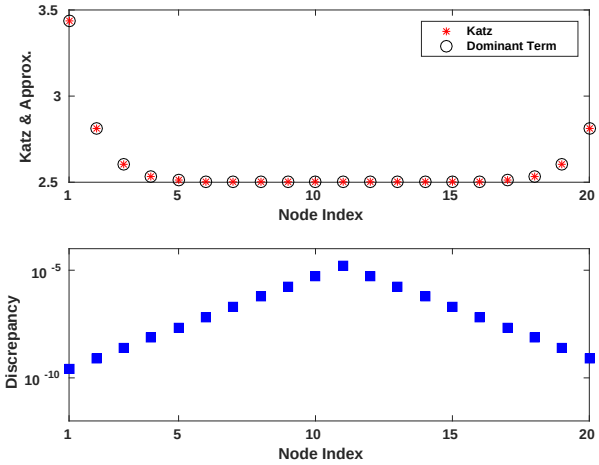


Figure 1: Upper picture: asterisks show components of Katz vector from (1) and circles show the approximation $C + h_1 t_1^{p(i)}$ from (2). Lower picture: the discrepancy $|x_i - C - h_1 t_1^{p(i)}|$. From Theorem 1, this quantity has the form $|h_2 t_2^{p(i)}|$, and hence grows geometrically away from the shortcut node. However, it is of $O(t_1^{N/2})$ for a fixed $0 < t_1 < 1$, and hence rapidly becomes negligible as the network size N increases.

References

[1] M. Aprahamian, D. J. Higham, and N. J. Higham. Matching exponential-based and resolvent-based centrality measures. *Journal of Complex Networks*, to appear, 2016.

[2] E. Estrada. *The Structure of Complex Networks*. Oxford University Press, Oxford, 2011.

[3] L. Katz. A new index derived from sociometric data analysis. *Psychometrika*, 18:39–43, 1953.

[4] X. Liu, G. Strang, and S. Ott. Localized eigenvectors from widely spaced matrix modifications. *SIAM J. Discrete Math.*, 16:479–498, 2003.

[5] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

THE GROUND TRUTH ABOUT METADATA AND COMMUNITY DETECTION IN NETWORKS

Daniel B. Larremore, Leto Peel, & Aaron Clauset

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

Community detection is a fundamental task of network science that seeks to describe the large-scale structure of a network by dividing the network's vertices into communities, blocks, or groups, based only on the pattern of edges. It is common to evaluate the performance of community detection algorithms by their ability to find so-called *ground truth* communities. This works well in synthetic networks with planted partitions because such networks' links are formed explicitly based on the planted partition. However, there are no planted communities in real world networks, so in their place, it is common to use discrete-valued node attributes, or *metadata*, to define a partition to act as ground truth. In this work, we first argue that metadata are not the same as ground truth, and that to treat them as such raises theoretical and practical concerns. Nevertheless, understanding the relationship between metadata and community structure is important, so we subsequently introduce two new techniques to rigorously explore this relationship.

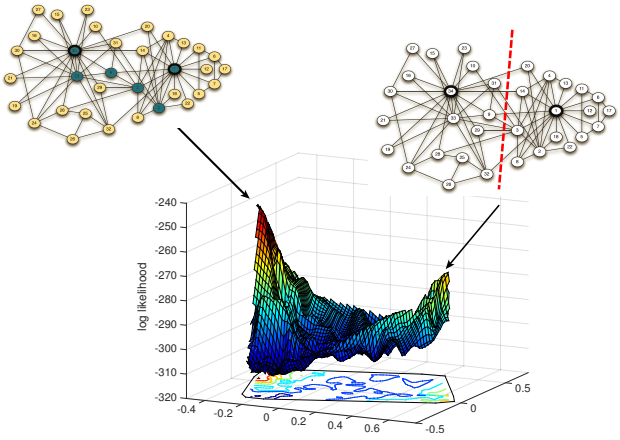


Figure 1: Zachary's Karate Club [7] is an example of a real-world network whose metadata are often treated as ground truth. But even in this simple network there are multiple alternative partitions [2, 5, 3]. Above, we show the stochastic blockmodel likelihood surface for a two-dimensional embedding of partitions. As indicated, the lower peak corresponds to the two-faction metadata partition, while the higher peak corresponds to a high/low-degree partition.

Metadata labels are not ground truth communities

The merit of using metadata labels as ground truth communities is that if there *is* correspondence between communities and metadata, it tells us that there is a relationship between the network structure and the metadata, while at the same time implying that the community detection algorithm is identifying useful communities. However, networks can have many plausibly "good" partitions [4], so when communities and metadata do not match it is not necessarily because the community detection algorithm does not perform well. In fact, when analyzing real-world networks, there are four possible reasons for mismatch between metadata and communities: (i) metadata do not relate to network structure, (ii) communities and metadata capture different aspects of network structure, as shown in Fig 1, (iii) the network contains no real structure, or (iv) the algorithm performed poorly. Despite these possibilities, typically the assumption is that (iv) is the only possible cause. This is indicative of a potentially substantial selection bias in the published literature: we, as a community, have developed methods that hit only a certain kind of target.

Methods to explore metadata-community relationships

While metadata labels should not serve as ground truth to calibrate community detection algorithms, the relationship between network structure and metadata can be meaningful. In many cases, metadata may enable us to learn about the processes responsible for link formation or dissolution. To better diagnose the actual relationship between metadata and structure, we briefly describe two methods below. First, we introduce a statistical test that measures the ability of the metadata to describe the network structure under a given model. It compares the metadata to a distribution of random partitions based on the likelihood of generating the network, and can be used with a vari-

ety of likelihood-based generative network models. This test addresses case (i) above. We then introduce a new stochastic blockmodel which incrementally relaxes its use of metadata, exploring the transition through partitions between metadata and best-fit communities. This method addresses case (ii) above. We validate both methods on synthetic and real-world examples.

The blockmodel entropy significance test

We introduce the blockmodel entropy significance test to determine whether or not the metadata and network structure are related. In this test we calculate the entropy of the network given the metadata, by fitting a stochastic blockmodel using the partition given by the metadata labels. We compare this entropy against a null distribution of entropies calculated by randomized metadata labels. Using this test we can determine the probability that the observed metadata labels were assigned at random relative to the network. The test is not limited to a specific model and in theory any likelihood-based model could be used to test for different types of relationships between metadata and network structure. Here we demonstrate it using different variants of the blockmodel, including degree-corrected [5], bipartite [6] and mixed-membership SBMs [1]. We identify relationships between metadata and network structure in real data and demonstrate cases where there exists a significant relationship between multiple sets of metadata and the network structure.

The neoSBM

We can use the test described above to determine *if* a significant relationship between metadata and network structure exists. However, if these metadata do not match the communities we detect, then we should diagnose *why* they differ. To do so, we introduce a new stochastic blockmodel called the neoSBM to determine if the communities and metadata capture the same or different aspects of the network structure.

The neoSBM extends the SBM by finding community structure, yet allowing metadata to exert an influence over the inferred communities. Specifically, the neoSBM chooses whether or not each node is assigned to its metadata community or if it is free to choose its own community at a cost. By varying this cost, the neoSBM effectively explores the space of partitions to find a path between the metadata and community partitions. The type of path tells us about how the two partitions are related. A smooth path between the two indicates that the metadata is close to the global optimum and suggests that they represent the same aspect of the network structure. On the other hand, the presence of a sharp phase transition in the path suggests that the metadata is at (or close to) a different local optimum, which we can interpret as a different aspect of the network structure. Figure 1 shows the partition space and likelihood under the SBM for the karate club network [7]. Here we see that the metadata corresponds to a local optima representing the assortative group structure of the network, while the global optimum captures a core-periphery structure. Both are relevant and interesting aspects of the network. As with the blockmodel entropy significance test, the neoSBM can be adapted to the broad class of stochastic blockmodels.

References

[1] B. Ball, B. Karrer, and M. Newman. Efficient and principled method for detecting communities in networks. *Physical Review E*, 84(3):036103, 2011.

[2] X.-Q. Cheng and H.-W. Shen. Uncovering the community structure associated with the diffusion dynamics on networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2010(04):P04024, 2010.

[3] T. S. Evans. Clique graphs and overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment*, 2010(12):P12037, 2010.

[4] B. H. Good, Y.-A. de Montjoye, and A. Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106, 2010.

[5] B. Karrer and M. E. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

[6] D. B. Larremore, A. Clauset, and A. Z. Jacobs. Efficiently inferring community structure in bipartite networks. *Physical Review E*, 90(1):012805, 2014.

[7] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473, 1977.

A NETWORK FLOW DUALITY FOUNDATION FOR HIERARCHICAL CLUSTER ANALYSIS

David W. Matula and Eli V. Olinick

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

Many popular data clustering and classification techniques from the social sciences lack a rigorous foundation in graph theory and mathematical optimization even though they are often based on graph and network models of interaction and affinity (or proximity). We show that a clustering method based on the fundamental graph-theoretic concept of density (i.e., *sparse cuts* separating dense clusters) and implemented via a duality to network flows can produce more comprehensive and meaningful results in appropriate problem domains.

Extended Abstract

The *maximum concurrent flow problem* (MCFP) is a peer-to-peer network flow problem defined on an edge-capacitated graph where the objective is to maximize the minimum *throughput*, i.e., the ratio of the flow delivered between a peer-to-peer pair in comparison to the corresponding demand for that pair [4, 5, 6, 7, 9, 10, 20]. The MCFP has applications in congested networks such as determining fair routing [1, 2] and modeling peer-to-peer traffic flow. Due to its relationship to the *sparsest cut problem* [13, 19, 20], it also has applications to VLSI circuit design [8], biological taxonomy [11, 16, 18], and hierarchical cluster analysis [15]. When the optimal throughput saturates only the set of *critical edges* and all others have slack, the *hierarchical* MCFP (HMCFP) then further maximizes the throughput in the slack edges determining a second throughput level and a second set of critical edges; and iterating further, a series of throughput levels is determined until all edges are critical, yielding a stratification portrayed in classification theory as a *dendrogram* [21]. The duality between sparsest cuts and MCF provides a natural characterization and foundation for the often stated, somewhat vague expression that objects in the same cluster have more affinity (connectivity) to each other and objects in different clusters are less similar (sparsely connected).

We discuss the MCFP and sparse cuts with applications to hierarchical cluster analysis and outline three key failings of a widely used class of data clustering algorithms that the HMCFP is designed to overcome. We propose a new community structure algorithm based on the HMCFP and its duality relation to the sequence of sparsest cuts, and discuss several theoretical properties which make it more accurate and often more robust than many non-deterministic algorithms in the clustering literature. Our algorithm is inspired by the idea of graph partitioning by sparsest cuts, which is appealing theoretically but difficult in practice as the sparsest cut problem is *NP*-hard [12, 19]. The MCF, which can be found efficiently, provides a good approximation since the value of the optimal throughput is bounded above by the capacity of a sparest cut (weak duality). Even in cases where the MCFP has multiple solutions, there is a set of critical edges that are saturated by every optimal flow. The HMCFP extends the MCFP by fixing the throughput for demand pairs that would otherwise be cut off by removing the critical edges and re-solving iteratively over the whole graph until all edges are critical at some level (i.e., throughput is fixed for all demand pairs). The successive MCFP solutions determine a series of sparse cuts or multipartite partitions. When the MCF solution identifies a partition into two to four parts, a sparsest cut can be identified [17]. Furthermore, it can be shown that the two to four parts are precisely the components obtained by removing the edges of all sparsest cuts. The sparsest cut can also be approximated using spectral techniques [3]; and so there is a connection between our approach and hierarchical spectral clustering. Partitioning via critical edges of the MCF generalizes the popular technique of removing edges with high betweenness centrality (e.g., [11]).

We summarize here a number of observations obtained either directly from analysis of the structure of the HMCFP or from examination of solutions of the HMCFP's in our lab.

Observation 1: Continuity: Small changes in capacities and/or demands in the HMCFP result in only small changes in the throughput levels of the demand pairs, even when the topology of the hierarchy (dendrogram) passes

through a transition point.

Observation 2: Hierarchical Independence: Sufficiently small changes in demands/capacities on the demand pairs/edges at certain throughput levels affect only the throughput levels on edges of the same or higher throughput (i.e., on those edges that are critical or have slack at that level). That is, these changes do not change the solutions of the MCFP's solved previously in the sequence at lower throughput levels.

Observation 3: Absorbing Backbone: There are absorbing edges at every throughput level. The absorbing edges induce a connected graph of the super nodes corresponding to the remaining components having slack at a given level.

Observation 4: Node Centrality: The HMCFP output produces a canonical node-centrality measure that is comprehensive and stable in the sense that small perturbations in capacity and/or demand result in only small changes in the centrality of any node. To see this consider that the HMCFP determines an absorbing edge backbone and concurrent flow levels for edges of every cut and multipartite cut. Even though the flows on particular paths are not necessarily unique there are many properties of the HMCFP solution that are unique; in particular, the HMCFP determines uniquely the amount of concurrent flow between each pair. Thus, the total terminal flow at a node is uniquely determined, with the balance of flow saturating the capacities of edges incident to the node being flow passing in and out through the node. The *flowthrough centrality* is the portion comprising the in-and-out flow, rather than the terminating flow, and those nodes of highest flowthrough centrality are important for many applications. The flowthrough centrality measure is more comprehensive and robust than simplistic measures based on degree and/or distance [14].

References

[1] M. Allalouf and Y. Shavitt. Maximum flow routing with weighted max-min fairness. In J. Solé-Pareta, M. Smirnov, P. V. Mieghem, J. Domingo-Pascual, E. Monteiro, P. Reichl, B. Stiller, and R. J. Gibbens, editors, *QofIS*, volume 3266 of *Lecture Notes in Computer Science*, pages 278–287. Springer, 2004.

[2] M. Allalouf and Y. Shavitt. Centralized and distributed algorithms for routing and weighted max-min fair bandwidth allocation. *IEEE/ACM Trans. Netw.*, 16(5):1015–1024, 2008.

[3] S. Arora, E. Hazan, and S. Kale. $O(\sqrt{\log n})$ approximation to sparsest cut in $\tilde{O}(n^2)$ time. *SIAM Journal on Computing*, 39(5):1748–1771, 2010.

[4] P.-O. Bauguion, W. Ben-Ameur, and E. Gourdin. Efficient algorithms for the maximum concurrent flow problem. *Networks*, 65(1):56–67, 2015.

[5] D. Bienstock and O. Raskina. Asymptotic analysis of the flow deviation method for the maximum concurrent flow problem. *Mathematical Programming, Series B*, 91:479–492, 2002.

[6] J. Biswas and D. W. Matula. Two flow routing algorithms for the maximum concurrent flow problem. In *ACM '86: Proceedings of 1986 ACM Fall joint computer conference*, pages 629–636, Los Alamitos, CA, USA, 1986. IEEE Computer Society Press.

[7] P. Chalermsook, J. Chuzhoy, A. Ene, and S. Li. Approximation algorithms and hardness of integral concurrent flow. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, pages 689–708, New York, NY, USA, 2012. ACM.

[8] S.-J. Chen and C.-K. Cheng. Tutorial on VLSI partitioning. *VLSI Design*, 11:175–218, 2000.

[9] S. Chiou. A combinatorial approximation algorithm for concurrent flow problem and its application. *Computers & Operations Research*, 32:1007–1035, 2005.

[10] L. K. Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. *SIAM J. Discret. Math.*, 13(4):505–520, 2000.

[11] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99:7821–7826, 2002.

[12] E. Gourdin. A mixed integer model for the sparsest cut problem. *Electronic Notes in Discrete Mathematics*, 36(0):111 – 118, 2010. ISCO 2010 - International Symposium on Combinatorial Optimization.

[13] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46:787–832, 1999.

[14] C. F. Mann. *Extensions of Maximum Concurrent Flow to Identify Hierarchical Community Structure and Hubs in Networks*. PhD thesis, Southern Methodist University, Dallas, TX 75275, 2008.

[15] C. F. Mann, D. W. Matula, and E. V. Olinick. The use of sparsest cuts to reveal the hierarchical community structure of social networks. *Social Networks*, 30(3):223–234, 2008.

[16] D. W. Matula. Cluster validity by concurrent chaining. In J. Felsenstein, editor, *Numerical Taxonomy*, pages 157–167. Springer-Verlag, Berlin, 1983.

[17] D. W. Matula. Concurrent flow and concurrent connectivity in graphs. In Y. A. et al., editor, *Graph Theory and its Applications to Algorithms and Computer Science*, pages 543–559. Wiley, 1985.

[18] D. W. Matula. Divisive vs. agglomerative average linkage hierarchical clustering. In W. Gaul and M. Schader, editors, *Classification as a tool of research*, pages 289–301. Elsevier Science Publishers B. V. (North Holland), Amsterdam, 1986.

[19] D. W. Matula and F. Shahrokhi. Sparsest cuts and bottlenecks in graphs. *Discrete Applied Mathematics*, 27:113–123, 1990.

[20] F. Shahrokhi and D. W. Matula. The maximum concurrent flow problem. *Journal of the ACM*, 32(2):318–334, April 1990.

[21] P. Sneath and R. Sokal. *Numerical Taxonomy. The Principles and Practice of Numerical Classification*. Freeman, 1973.

EARLY DETECTION OF FAILURE MECHANISMS IN RESILIENT BIO-STRUCTURES: A COMPLEX NETWORK STUDY

Reena R. Patel,Guillermo Riveros, Jan Hoover, Ed Perkins, David Thompson

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Abstract

Nature has optimized complex biostructures over billions of years to have remarkable resiliency, strength, and lightweight composition. This work will present a novel integrated approach that employs computational mechanics and complex network strategy to gain fundamental insights into the failure mechanisms of high performance, lightweight, structured composites by examining structural and material properties of the rostrum of the paddlefish (Figure 1). Results of the computational mechanics simulations and complex network analysis on the rostrum of the paddlefish will be presented with emphasis on early detection of failure mechanisms.

Introduction

Complex networks have been used to examine interactions within various systems such as traffic flow, energy flow through food webs in an ecosystem, fluid flow through pipelines, communication networks, electrical networks, community structure of company ownership, patrol routing problem, air-transportation, failure of quasi-brittle materials, and force transmission in dense granular media [1, 2]; however, this technique has never been used to study the complex hierarchical geometry of living specimens. Swimming enhancement induced by the rostrum has been studied by Riveros et. al. [3, 4, 5]. Recent computational efforts have revealed that the rostrum has far greater energy dissipation and impact resistance characteristics when compared to man-made engineered systems [6]. The rostrums lattice structure is a major contributor to its superior performance. However, the lattice is considered an indeterminate, non-linear structure with varying material types, and properties, with non-uniform stiffness and irregular shaped. Knowledge is lacking on how the structure is geometrically assembled to possess high-performance strength/toughness characteristics. A combinatorial approach that takes advantage of computational mechanics and complex network theories will assist in understanding the role redundant hierarchical lattice members have in achieving structural resiliency.

Figure 1: (a) Paddlefish (b) Paddlefish rostrum cartilage skeleton

Methodology

Biological systems use hierarchical geometrical arrangements and are comprised of heterogeneous constituents causing uncertainty as what dictates material response. Towards this end, the surface topology obtained from micro CT imagery and a weighting function based on strain and flow measurement, which are thought to be the most reliable data that can be measured, will be used to identify the patterns as force chains are short circuited. The pattern recognition will be used to correlate for the Early Detection of Failure Mechanisms in Resilient Bio-Structures. This study will use a novel mathematical technique to formulate the rostrum as a network flow problem [7, 8]. To achieve this, a flow network graph G= (V, E) must be developed from the computational mechanics model of the rostrum such that,

- V Represents the nodes obtained from the finite element model of the rostrum
- E Represents the edges, connecting the nodes in G, indicating the connectivity between the nodes
- Each edge (u,v) in E, has a cost C associated with it which is representative of the cost associated with sending one unit of flow (stress/strain or deformation

that is used to quantify the cost/constraint of the flow) through the edge

- Each edge (u,v) in E, has a capacity U associated with it which is representative of the maximum amount of flow that could be transmitted through the edge

- We identify two nodes in the network to represent the source s and target t, such that the flow can be transmitted from the source node s to the target node t. The selection of these nodes will be dependent on the force boundary conditions to which the rostrum will be subjected.

The problem of transmitting the maximum amount of flow through the network at the minimum cost can be approached using a state of the art mathematical algorithm. For example, the flow network shown in figure 2 is constructed from a small part of the rostrum subjected to an impulse loading. The flow network is constructed from the output obtained from a computational mechanics simulation on the rostrum. Normalized Von-Mises stresses are used for computing the cost in this demonstration model. The possible cuts are [(0-1), (0-5)], [(9-8), (6-8), (8-12)], [(5-6)] etc. The minimum source-target cut is (5-6) which has a capacity of 26 (highlighted in red in figure 2). Based on the flow patterns (governed by stresses, strains, or deformations) established at the onset of load applications, complex network approach can aid in detecting the failure site much earlier than any computational or analytical methodologies. The capacity and constraints of the network will be extracted from the dynamically evolving numerical simulation results.

Results and Discussion

Results obtained from the temporally and spatially evolving network graphs will be discussed. A one on one comparison will be shown to demonstrate how the mathematical flow network approach can aid in early detection of failure mechanisms in bio-structures.

Future Work

Future work will involve novel strain measurements on living biological specimens to validate the computational mechanics and mathematical models. Also, a state of the art parallel algorithm will be developed to tackle the dynamically evolving temporal/spatial data encountered in these analyses.

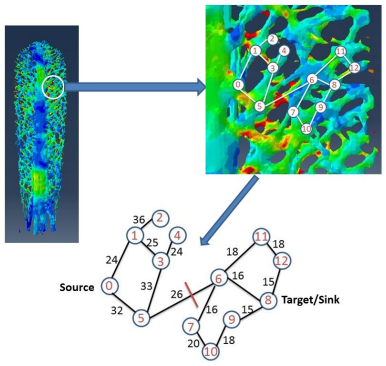


Figure 2: A demonstration model constructed from a small part of the rostrum to illustrate the modus operandi of complex network algorithm

Acknowledgements

The authors acknowledge the financial support provided by the U.S. Army Engineer Research and Development Center (ERDC) Military Engineering 6.1 Basic Research program. The authors are also grateful for the contributions made by Mississippi State University.

References

[1] V. Lefort, G. Pijaudier-Cabot, and D. Gregoire. Analysis by ripleys function of the correlations involved during failure in quasi-brittle materials: experimental and numerical investigations at the mesoscale. *Engineering Fracture Mechanics*, 2015.

[2] A. Tordesillas, D. M. Walker, G. Froyland, J. Zhang, and R. P. Behringer. Transition dynamics and magic-number-like behavior of frictional granular clusters. *Physical Review E*, 86(011306), 2012.

[3] G. Riveros, R. Patel, and J. Hoover. Swimming enhancement induced by the rostrum of the paddlefish (polyodon spathula) in laminar flows: A multiphysics, fluid-structure interaction analysis. *submitted to Applied Mathematical Modelling.*

[4] R. Patel and G. Riveros. Towards development of innovative bio-inspired materials by analyzing the hydrodynamic properties of polyodon spathula (paddlefish) rostrum. *ERDC/ITL TR-13-4.*

[5] J. B. Allen and G. A. Riveros. Hydrodynamic characterization of the polyodon spathula rostrum using cfd. *Journal of Applied Mathematics*, 2013.

[6] G. Riveros, R. Patel, and J. Hoover. Swimming and energy dissipation enhancement induced by the rostrum of the paddlefish (polyodon spathula): A multiphysics, fluid-structure interaction analysis. *Materials Research Society Fall Meeting, 30 Nov 5 Dec, 2014, Boston, MA.*

[7] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. Network flows: Theory, algorithms, and applications. *Proceedings of the National Academy of Sciences*, 1st Edition, 1993.

[8] M. E. J. Newman. Networks: An introduction. *Oxford University Press, Oxford*, 2010.

AN INTRODUCTION TO THE THEORY OF P-MODULUS ON NETWORKS

Pietro Poggi-Corradini

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Introduction

I will present on overview of ongoing research being done by the NODE[1] research group (https://node.math.ksu.edu/) at Kansas State University. The NODE group is an interdisciplinary team of researchers from the department of Mathematics and the department of Electrical and Computer Engineering, working principally in Network Science. My colleague, Nathan Albin (Math) will address the numerical aspects of the theory of p-modulus. Moreover, my Ph.D. student in Math, Nethali Fernando will report on p-modulus metrics, while my Ph.D. student, Heman Shakeri, co-advised with Professor Caterina Scoglio in ECE, will adress various applications and centrality measures derived from p-modulus.

The theory of conformal modulus was originally developed in complex analysis, see [1, p. 81]. The more general theory of p-modulus grew out of the study of quasiconformal maps, which generalize the notion of conformal maps to higher dimensional real Euclidean spaces and, in fact, to abstract metric measure spaces.

Intuitively, p-modulus provides a method for quantifying the richness of a family of curves, in the sense that a family with many short curves will have a larger modulus than a family with fewer and longer curves. The parameter p tends to favor the "many curves" aspect when p is close to 1 and the "short curves" aspect as p becomes large. This phenomenon was explored more precisely in [2] in the context of networks.

The concept of discrete modulus on networks is not new, see for instance [7, 10, 9]. However, our goal is to develop the theory of p-modulus as a graph-theoretic quantity [6, 2], with an eye to finding applications, for instance to the study of epidemics [11, 8].

p-Modulus

For simplicity, I will only discuss the case $1 \leq p < \infty$, however it also makes sense to send p to ∞ in various ways. Assume $G = (V, E, \sigma)$ is a simple finite undirected graph with an assignment of edge-weights $\sigma : E \to (0, \infty)$.

In the most general case we can measure the modulus of a collection Γ of functions $\gamma : E \to [0, \infty)$, which can also be viewed as a family of vectors $\Gamma \subset \mathbb{R}_+^{|E|}$. In practice, we often assume these functions to be indicator functions, with values in $\{0, 1\}$, hence representing families of subsets of E. For instance, we have studied the modulus of the family Γ_{spt} of all *spanning trees* of the graph G, as well as the family Γ_{loop} of all *simple cycles*, or the family Γ_{cut} of all *cuts*. When studying families of walks, it is more convenient to think of the corresponding integer-valued multiplicity functions $\gamma(e) = \mathcal{N}(e)$ that record the number of times a walk traverses a given edge e. Examples of families of walks are the *connecting walks* $\Gamma(a, b)$, consisting of all walks that start at node a and end at node b; the *via walks* $\Gamma(a, b; c)$, walks from a to b that must visit c along the way; the *long walks*, $\Gamma_{\text{long}}(L)$, walks that take at least L hops.

One of the major strengths of modulus is that it can quantify the richness of all of these different families of objects. Such flexibility is useful in practice when studying specific applications, see for instance [11].

Given a (possibly infinite) family $\Gamma \subset \mathbb{R}_+^{|E|}$ its *p-modulus* is defined as:

$$\text{Mod}_{p,\sigma}(\Gamma) := \inf_{\rho \in \text{Adm}(\Gamma)} \mathcal{E}_{p,\sigma}(\rho).$$

In other words, we minimize the *energy*

$$\mathcal{E}_{p,\sigma}(\rho) := \sum_{e \in E} \sigma(e)\rho(e)^p$$

over all *admissible densities* $\rho : E \to [0, +\infty]$. A density is admissible if it penalizes every element in Γ at least 1, meaning that

$$\ell_\rho(\gamma) := \sum_{e \in E} \gamma(e)\rho(e) \geq 1.$$

When $1 < p < \infty$ there is a unique *extremal density* which we denote by ρ^* [2]. Moreover, in all the examples mentioned above, even if families of walks are usually infinite, one can always find a finite subfamily Γ^* with the same admissible densities and hence the same modulus [6].

[1]NODE is supported by NSF grant n. 1515810

As a consequence, modulus becomes an *ordinary convex program*, and the standard tools of Lagrangian duality can be deployed [2, 5]. Moreover, we have also studied the notion of *blocking duality* pioneered by Fulkerson. For instance, we establish a formula relating the modulus of a family with the modulus of its *blocker* [4].

In the special case of connecting families $\Gamma(a, b)$ we recover some classical quantities. For instance, 2-modulus coincides with effective conductance, when viewing the graph as an electrical network with edge-conductances equal to σ. Also, 1-modulus is equal to Min Cut. And letting p tend to infinity, the p-th root of p-modulus tends to the reciprocal of shortest-path [2]. In general, p-modulus continuously interpolates between these classical measures. In fact, it is well-known that shortest-path and effective resistance are *metrics* on the set of nodes V, i.e., they satisfy the triangle inequality. We have shown that the reciprocal of Min Cut is also a metric and more generally $\mathrm{Mod}_p(\Gamma(a, b))^{-1/p}$ is a metric for all p's [3].

Algorithms

We have developed algorithms to compute modulus efficiently on large networks. This will be the topic of Nathan Albin's presentation. Currently, we have been able to handle networks with hundreds of thousands of edges, but we hope to improve our algorithms further. In [5] we showed that in theory one only needs at most $|E|$ elements of the family Γ to be able to compute p-modulus, and in some cases (depending on the specific family) even less. We call these subfamilies $\tilde{\Gamma}$ such that $\mathrm{Mod}_p(\tilde{\Gamma}) = \mathrm{Mod}_p(\tilde{\Gamma})$, *minimal* subfamilies. Our current algorithm builds a sequence of growing subfamilies whose modulus converges to the actual modulus [6], and in experiments it seems that these approximating subfamilies tend to want to become minimal.

Applications

Our current application have focused on using modulus to study epidemic spreading on a network. In [11], we develop some centrality measures using modulus and tested them by comparing them to other classical centralities. The comparison was done numerically by running many simulated epidemics on the graph and computing the efficiency of a mitigation strategy consisting of vaccinating a percentage of the nodes with highest centrality. The results were very favorable to centralities defined in terms of modulus. One other advantage of modulus defined centralities is the great flexibility of the concept of modulus, which for instance can be computed perfectly well on direct graphs as well.

In order to try to explain why modulus based centrality work so well with respect to epidemics, in [8], we introduced and studied the notion of Epidemic Hitting Time. This is the expected time it takes for an epidemic infection originating at node a to infect node b. We show that epidemic hitting time is also a metric on V and that it is always bounded below by effective resistance, namely the reciprocal of 2-modulus. So this bound gives at least a partial explanation for the effectiveness of modulus in the context of epidemic spreading.

Conclusion

The notion of p-modulus is a fundamental tool in complex analysis and geometric function theory more generally. Our main goal, is to study and apply the corresponding notion of p-modulus on networks. We have already obtained several results in this direction and there are still many open areas of exploration.

References

[1] L. V. Ahlfors. *Conformal Invariants: Topics in Geometric Function Theory*. McGraw-Hill, 1973.

[2] N. Albin, M. Brunner, R. Perez, P. Poggi-Corradini, and N. Wiens. Modulus on graphs as a generalization of standard graph theoretic quantities. *Conformal Geometry and Dynamics*, 19:298–317, 2015.

[3] N. Albin, N. Fernando, and P. Poggi-Corradini. Modulus metrics. Preprint.

[4] N. Albin and P. Poggi-Corradini. Blocking duality for p-modulus on networks. Preprint.

[5] N. Albin and P. Poggi-Corradini. Minimal subfamilies for p-modulus on graphs. Preprint.

[6] N. Albin, P. Poggi-Corradini, F. Darabi Sahneh, and M. Goering. Modulus of families of walks on graphs. arXiv:1401.7640.

[7] R. Duffin. The extremal length of a network. *Journal of Mathematical Analysis and Applications*, 5(2):200 – 215, 1962.

[8] M. Goering, N. Albin, F. Sahneh, C. Scoglio, and P. Poggi-Corradini. Numerical investigation of metrics for epidemic processes on graphs. 2016. Pre-accepted for the session "Epidemic Control in Networks" of the 49th Asilomar Conference on Signals, Systems and Computers, Nov. 8-11, 2015.

[9] P. Haïssinsky. Empilements de cercles et modules combinatoires. *Annales de l'Institut Fourier*, 59(no. 6):2175–2222, 2009. Version revisée et corrigée.

[10] O. Schramm. Square tilings with prescribed combinatorics. *Israel Journal of Mathematics*, 84(1-2):97–118, 1993.

[11] H. Shakeri, P. Poggi-Corradini, C. Scoglio, and N. Albin. Generalized network measures based on modulus of families of walks. *Journal of Computational and Applied Mathematics*, 2016.

A FAST PARAMETERIZED ALGORITHM FOR CO-PATH SET

Blair D. Sullivan and Andrew J. van der Poel

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

The k-Co-Path Set problem asks, given a graph G and a positive integer k, whether one can delete k edges from G so that the remainder is a collection of disjoint paths. We give a linear-time fpt algorithm with complexity[1] $O^*(1.838^k)$ for deciding k-Co-Path Set, significantly improving the previously best known $O^*(2.17^k)$ of Feng et al. [5]. We also present a $O^*(4^{tw(G)})$ algorithm for Co-Path Set using the Cut&Count technique. In general graphs, we combine this with a branching algorithm which refines the previously-known $6k$-kernel into bounded-treewidth reduced instances.

Introduction

Co-Path Set [1] is an NP-complete problem asking for the minimum number of edges whose deletion from a graph results in a collection of disjoint paths (such a set of edges is a *co-path set*).

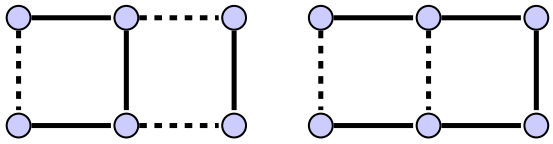


Figure 1: Two co-path sets (dashed edges) of a graph, including a solution to Co-Path Set (right).

Co-Path Set is naturally motivated by a special case of finding *radiation hybrid mappings* in genetics. These mappings are constructed by researchers working to determine the ordering of *genetic markers* on chromosomes using data collected from DNA fragments (formed by breaking chromosomes with gamma radiation) [2, 7, 8]. In general, given a set of known markers $M = \{m_1, \ldots, m_n\}$, each DNA fragment will contain a subset of M. A radiation hybrid mapping is a linear ordering of M which is consistent with the constraints implied by the fragments (e.g. if m_1 and m_3 co-occur alone in some fragment, then the ordering $m_1, m_2, m_3, \ldots, m_n$ is inconsistent since m_2 would have necessarily appeared in every fragment containing both m_1 and m_3). If no consistent ordering exists (likely due to errors in the data), one attempts to find a mapping which is consistent with as many fragments as possible.

Restricting to the setting in which fragments always contain a pair of markers is equivalent to Co-Path Set — each fragment gives an edge in a graph where the vertices correspond to markers, and we wish to remove the minimum number of edges so the resulting graph is a collection of disjoint paths (naturally giving linear orderings), as shown in Figure 2.

Parameterization Tools

Parameterized complexity [4] is a fine-graining of the complexity hierarchy, where the running time of parameterized algorithms can depend on both the original input and an additional parameter k. *Fixed-parameter tractable* variants of NP-hard problems can be solved in time that is some function in terms of k and polynomial in the size of the problem. More formally, FPT problems can be solved in time $f(k) \cdot |n|^{O(1)}$. A problem is linear-fpt if the complexity with respect to the size of n is linear.

We use kernelization, a polynomial time pre-processing technique, in our algorithm. This is a commonly used method in parameterized complexity, where the easy portions of the given instance are handled leaving the difficult components (the kernel) which need computationally complex (and expensive) techniques to solve. The size of the kernel is bounded in terms of the parameter.

We study k-Co-Path Set, the version of Co-Path Set using the natural parameter of edges deleted:

> k-Co-Path Set
> **Input:** A graph $G = (V, E)$ and $k \in \mathbb{Z}^+$
> **Parameter:** k
> **Problem:** Does there exist $F \subseteq E$ with $|F| = k$ such that $G[E \setminus F]$ is a set of disjoint paths?

In the context of radiation hybrid mappings k is the number of errors in the data, and since this parameter should be small, our algorithms will be fast.

[1] we use the notation $O^*(f(k))$ for $O(f(k)n^{O(1)})$ when denoting the complexity of fixed parameter tractable algorithms

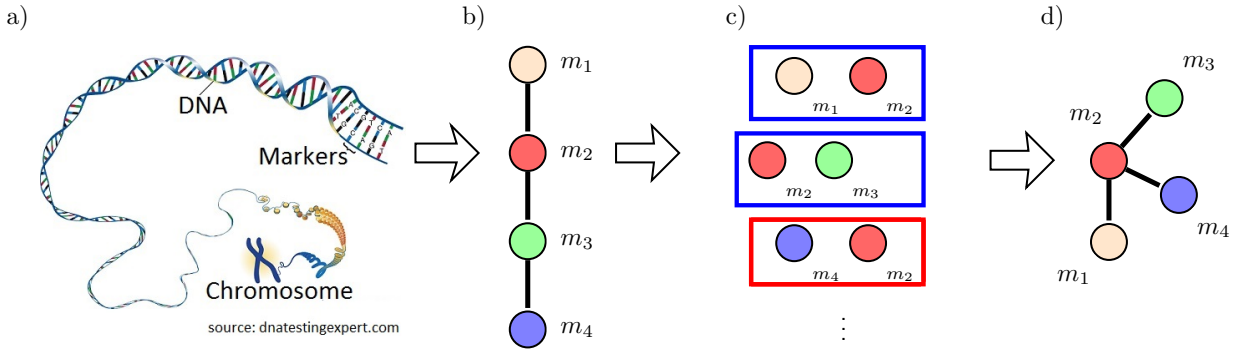


Figure 2: In genetic material (a), known genetic markers occur in some linear order (b). Fragments of DNA created using gamma radiation (c) are analyzed and labelled with the markers they contain (potentially with observational errors, as in the third fragment). When all fragments have exactly two markers, we can represent this as a graph (d), where a minimum co-path sets corresponds to a maximally informative linear ordering.

Our first algorithm studies Co-Path Set parameterized by treewidth using the Cut&Count framework of [3] to k-Co-Path Set, a technique which might be of independent interest to the community. We use this adaptation as a subroutine in our main algorithm for solving k-Co-Path Set on general graphs. The Cut&Count framework enables $O(c^{tw}n^{O(1)})$ one-sided Monte Carlo algorithms which solve connectivity-type problems with constant probability of a false negative. It uses dynamic programming over a nice tree decomposition [6], in addition to utilizing modulo-2 counting and parity tricks to provide fast parameterized algorithms. It is crucial to obtain tree decompositions of bounded size in order to use Cut&Count due to its run time dependence on treewidth.

Fastest Algorithm

Our major result is a linear-fpt algorithm which decides k-Co-Path Set in time $O^*(1.838^k)$, and is the fastest known algorithm for this problem. The algorithm uses a kernelization process [5] to find a kernel of size at most $6k$. We take this kernel and bound its degree using a branching algorithm which considers all possible subgraphs of the kernel with maximum degree of 6. This enables us to form tree decompositions which are compatible with the Cut&Count algorithm, which decides whether or not the given instance has a co-path set of size k.

Open Problems

One natural question is whether similar techniques extend to the generalization of Co-Path Set to k-uniform hypergraphs (as treated in Zhang et al. [9]). It is also open whether the dual parameterization asking for a co-path set of size k resulting in ℓ disjoint paths is solvable in sub-exponential fpt time.

References

[1] Y. Cheng, Z. Cai, R. Goebel, G. Lin, and B. Zhu. The radiation hybrid map construction problem: recognition, hardness, and approximation algorithms. Unpublished Manuscript, 2008.

[2] D. Cox, M. Burmeister, E. Price, S. Kim, and R. Myers. Radiation hybrid mapping: a somatic cell genetic method for constructing high-resolution maps of mammalian chromosomes. *Science*, 250(4978):245, 1990.

[3] M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. van Rooij, and J. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *FOCS*, pages 150–159. IEEE, 2011.

[4] R. G. Downey and M. R. Fellows. *Parameterized complexity*. Springer Science & Business Media, 2012.

[5] Q. Feng, Q. Zhou, and J. Wang. Kernelization and randomized parameterized algorithms for co-path set problem. *J. of Combin. Optimization*, pages 1–12, 2015.

[6] T. Kloks. *Treewidth, Computations and Approximations*, volume 842 of *LNCS*. Springer, 1994.

[7] C. Richard III, D. Withers, T. Meeker, S. Maurer, G. Evans, R. Myers, and D. Cox. A radiation hybrid map of the proximal long arm of human chromosome 11 containing the multiple endocrine neoplasia type 1 (men-1) and bcl-1 disease loci. *Am. J. Hum. Genet.*, 49(6):1189, 1991.

[8] D. Slonim, L. Kruglyak, L. Stein, and E. Lander. Building human genome maps with radiation hybrids. *Journal of Computational Biology*, 4(4):487–504, 1997.

[9] C. Zhang, H. Jiang, and B. Zhu. Radiation hybrid map construction problem parameterized. In *Combin. Opt. and App.*, volume 7402 of *LNCS*, pages 127–137. Springer, 2012.

DESIGNING EXIT FREQUENCY DISTANCE MEASURES FOR BIOLOGICAL NETWORKS

Andrew Beveridge, Mengfei Cao, Amanda Redlich and Lenore Cowen

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We prove a direct connection between the Diffusion State Distance (DSD) introduced by Cao et al [4, 3] and a certain class of mixing random walks from singleton nodes. This allows us to construct a closely related distance measure within the framework of *Exit Frequency Distances (EFDs)* We show empirically that the EFD distance matrix, where the underlying target distribution is the stationary distribution, is very close to the DSD distance matrix for the protein-protein interaction network for *S. cerevisiae,* and show that we can substitute the EFD matrix for the DSD matrix in classical function prediction algorithms with hardly any performance degradation.

This is significant because the EFD framework gives a natural mathematical way to generalize the distance measure to incorporate biological notions of *important nodes* or *important subnetworks* into the distance metric. We propose two natural alternative ways to do this. We apply this to improve our ability to extract the important genes related to human disease phenotypes from the known available network of protein-protein associations in humans.

Introduction

One of the best-studied classical problems on biological networks involves using proteins of known function, together with the structure of the network of known protein-protein interactions (PPI), to make predictions of functions of unlabeled protein nodes. This is an important problem because, even in the best-studied model organisms, such as *S. cerevisiae* (Baker's yeast), these networks contain many proteins whose function is still completely uncharacterized. In the classical PPI network, one places an edge between two proteins only if there is experimental evidence that the two proteins physically bind in the cell. Recently, high-throughput techniques have generated additional types of information about the interaction or association of genes or proteins in a cell that can be encoded in a graph, or network context. This can include information such as genetic interaction data, which represents epistasis events from double knockouts, co-expression data for genes that are active under similar conditions, or even, for the human PPI network, disease phenotypes.

In their earlier work, Cao et al. [4, 3] introduced a novel diffusion-based graph metric that they called the *Diffusion State Distance,* or DSD, and showed that when DSD was substituted in a straightforward way for ordinary shortest-path distance in several popular classical methods for predicting functional labels in yeast, prediction improved across the board. Let $He^k(v_i, v_j)$ denote the expected number of times that a random walk starting at node v_i visits node v_j in timestep 0 to k. Fixing v_i, we can collect these values into a vector

$$He^k(v_i) = (He^k(v_i, v_1), He^k(v_i, v_2), \ldots He^k(v_i, v_n)).$$

Then we define

$$DSD(u, v) = \lim_{k \to \infty} ||He^k(u) - He^k(v)||_1$$

where Cao et al [4] showed that DSD is a metric and converges as $k \to \infty$. While Cao et al. specifically considered the L_1 norm, a generalization to the L_q norm is equally natural. The connection between DSD, the discrete Green's function and heat kernels on graphs, has been a topic of recent study [2]. We also consider the cDSD measure from [3], which generalizes DSD to graphs with weighted confidence values on the edges in a natural way.

Exit Frequency Distance

The exit frequency distance comes from the theory of exact stoping rules for random walks on graphs studied by Beveridge and Lovász [1]. Given an initial vertex v_i and a target distribution τ, a **stopping rule** $\Gamma(v_i, \tau)$ halts a random walk started at vertex v_i so that the final state is governed by τ. A stopping rule is **optimal** if it minimizes the expected number of total steps, among all (v_i, τ)-stopping rules. The **exit frequencies** $x_k(\Gamma)$ of the stopping rule are the total number of expected exits of vertex v_k during a walk governed by the rule, which can be shown to be the same for every optimal stopping rule

from v_i to τ. Thus a stopping rule is optimal if it achieves $\min_{\Gamma:v_i\to\tau}\sum_{k\in V}x_k(\Gamma)$. We denote the optimal exit frequency for vertex v_k by $x_k(v_i,\tau)$. Collecting the optimal exit frequencies from each vertex to the distribution τ into a matrix, we define

$$X_\tau(i,j)=x_j(v_i,\tau).$$

Setting $\tau=\pi$, the stationary distribution, we define

$$EFD_q(u,v)=||(1_u-1_v)X_\pi||_q.$$

Yeast Function Prediction Experiment

The Network

We use the physical protein-protein interaction dataset for *S. cerevisiae* S288c from BioGRID, version 3.4.128 (download date: Sept 17th, 2015). With the same preprocessing as in [4], we obtain a simple graph with 5074 nodes and 74351 confidence-weighted edges. The graph is connected and has diameter 6. Figure 1 shows the distribution of the DSD and EFD distance values for this network.

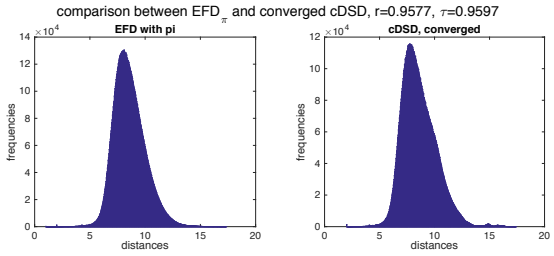


Figure 1: Comparison of Distribution of Distances

Node annotation

We use the functional annotation dataset consisting of the first three levels from MIPS FunCat version 2.1. MIPS FunCat is a shallow, levelled hierarchical classication scheme of nested protein functional catagories of increasing specificity. There are 17/74/154 unique functional catagory levels in the first/second/third level of the MIPS FunCat hierarchy that are used to annotate the nodes on our network. There are 4468/4452/4078 nodes that are annotated with at least one label in the first/second/third level of MIPS FunCat annotations in our network. We perform 10 runs of 2-fold cross validation on this network, and report the mean accuracy and F1 scores for each method.

Table 1: Protein function prediction of MIPS annotations

Method	1st Level		2nd Level		3rd Level	
	Accur.	F1	Accur.	F1	Accur.	F1
MV	0.5065	0.4211	0.3977	0.3064	0.3677	0.2919
wMV	0.5445	0.4451	0.4475	0.3411	0.4208	0.3308
EFD_π	0.6664	0.4943	0.5420	0.3901	0.4912	0.3634
cDSD	0.6689	0.4917	0.5528	0.3999	0.5015	0.3716

Results

We compare the following four methods for assigning labels to nodes. Table 1 gives a subset of our function prediction results across all 3 levels of the MIPS hierarchy. The first two methods are the baseline methods *Majority Vote,* and *Weighted Majority Vote* which have all direct neighbors vote for their labels (with equal weight, or in proportion to the confidence edge weights, respectively), and assigns each unlabeled node the label which receives the greatest number of votes. The other methods, rank all the nodes in a network in order of that node's cDSD or EFD distance, respectively, and have the closest $r=10$ nodes[1] vote for their labels. As can be seen, EFD_π does nearly as well as cDSD in predicting the labels.

Generalization to other EFD Distances

In the full paper, we fully develop the connection between the EFD and the DSD, explaining why it is not surprising that they behave so similarly. We then generalize the EFD by replacing the stationary distribution with other interesting target distributions. In particular, we present EFD distances that can represent a subnetwork S of nodes as more important, with applications to the human diseasome.

References

[1] A. Beveridge and L. Lovasz. Exit frequency matrices for finite Markov chains. *Combinatorics, Probability and Computing*, 19(4):541–560, 2010.

[2] E. Boehnlein, P. Chin, A. Sinha, and L. Lu. Computing diffusion state distance using Green's function and heat kernel on graphs. In *Algorithms and Models for the Web Graph*, pages 79–95. Springer, 2014.

[3] M. Cao, C. M. Pietras, X. Feng, K. J. Doroschak, T. S. ffner, J. Park, H. Zhang, L. J. Cowen, and B. Hescott. New directions for diffusion-based prediction of protein function: incorporating pathways with confidence. *Bioinformatics*, 30:i219–i227, 2014.

[4] M. Cao, H. Zhang, J. Park, N. M. Daniels, M. E. Crovella, L. J. Cowen, and B. Hescott. Going the distance for protein function prediction. *PLOS One*, 8:e76339, 2013.

[1] We vary r in the full paper.

A MULTI-NETWORK ANALYSIS OF SCIENTISTS ON SOCIAL MEDIA AND THEIR SCIENTIFIC CO-AUTHORSHIP GRAPHS

Patrick Mackey, Jennifer B. Webster

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

A promising area for network science research is the topic of multilayer networks, in which two or more distinct graphs are interconnected in some manner [5]. In our research, we have collected data for two different, but interrelated networks: one based on the interactions of well-known scientists on Twitter, the other based on their co-authorship of scientific publications on arXiv [1]. Each graph represents a distinct social environment that these researchers belong to. We have gathered a number of metrics on these networks which show relevant correlations and differences between them. By examining these networks in conjunction, we believe we can gain insights that may not have occurred when looking at either one in isolation. We will discuss these insights in this talk with the goal of promoting research on multilayer networks.

Data

To create our multilayer network, we first looked for well-known researchers who were also active Twitter users. A number of websites list such people [7] [8] which we used in our initial search. Our second requirement was the ability to get their co-authorship network. Currently, we have been limited to using the open access site arXiv for this information. We also initially restricted our research to scientists working in three domains: astrophysics, mathematics and computer science. Given these requirements, we began with an initial set of 23 researchers to build our network from.

Co-authorship graphs A_1 and A_2 were created using a breadth-first-search from our initials seeds 1 and 2 hops away, respectively. Any researcher who had co-authored a paper with one of our seeds was given an edge between them and all other co-authors of the paper, regardless of whether or not the other authors had Twitter accounts. A Twitter graph T_M was created by adding edges from our seeds to any other user they had mentioned in a tweet. By using mentions instead of follower information we were able to make use of temporal, weighted edges.

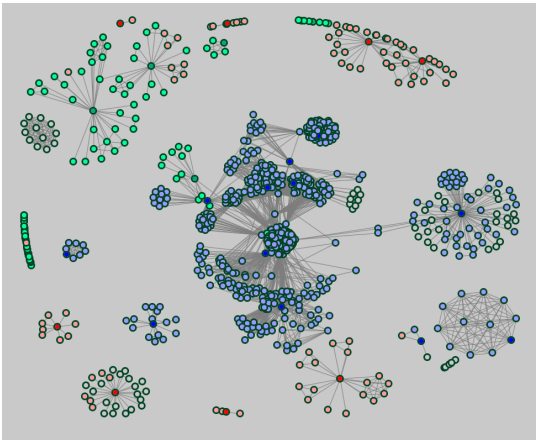


Figure 1: Co-authorship network of the initial 23 researchers, A_1. Color represents research field.

After creating these initial graphs, we examined the users being mentioned on Twitter by our initial 23 scientists. Interestingly, of these 9,385 users, we discovered 853 of them appeared to have publications on arXiv as well. This was determined using an exact name match between their real name as listed on Twitter and the name used for publication. While this method likely resulted in both false positives and false negatives, a manual examination of a number of these users seemed to show that most were in fact matched correctly. Using this information, we expanded our initial list of seeds to 876 people, creating significantly larger data sets to explore. A_1' had 30,807 nodes and 1,316,156 links. T_M' had 236,051 nodes and 1,815,743 links. By using the publication categories in arXiv, we were also able to make educated guesses towards the research discipline of each the new Twitter users based on what the majority of their publications were categorized as.

Analysis

While containing many of the same people, there were many significant differences between the Twitter and co-

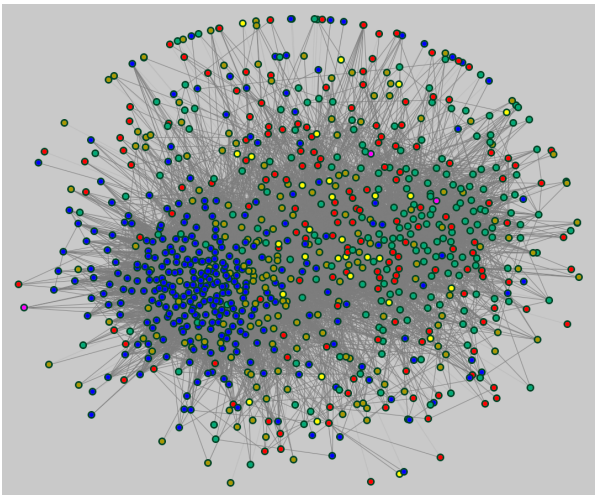


Figure 2: Network of Twitter mentions between the expanded set of 876 researchers.

authorship networks of these scientists. This included the directed nature of Twitter mentions, as well as the natural cliques that formed from publication co-authorship. An additional difference was the time scales of the interactions. Twitter mentions tended to be fast paced, while publications could be much further apart in time, representing a longer period of involvement among individuals for each edge.

Several analytics were performed on the networks, including a comparison of centrality [2], geodesic distances [6], and Jaccard vertex similarity [3]. Our calculations showed some correlations between the two networks for these metrics, although in some cases not as strong as initially expected.

One area with a strong correlation was between the research topics of a scientist's Twitter neighborhood and that of his own research. For example, a user had a 78% chance of being a mathematician if the majority of his neighbors in T'_M were also mathematicians (compared to an 18% chance over all). Similar percentages held for computer scientists (75%) and astronomers (79%) as well.

Another interesting correlation was that between co-authorship on arXiv and mentions on Twitter. Only a small percentage of our researchers directly co-authored a publication together on arXiv. There was a 0.1% chance that any two of our 876 authors were co-authors. However, if one of the authors had mentioned the other on Twitter, that probability increased to 2.0%. The likelihood of mentioning someone if you had co-authored a paper with them was also highly correlated. There was only a 3.8% chance of any two of our authors having a mention between them, but this rose to 51.9% if they had co-authored a paper together.

A fascinating reverse correlation related to the out-degree of scientists on Twitter compared to their number of publications on arXiv. While slight, it seems to indicate that activity on Twitter does not necessarily lend itself to increased publications for researchers. On the other hand there was a positive correlation between in-degree in T'_M and number of publications on arXiv. This may relate to the generally accepted notion that in-degree represents the importance of a person in a social network [4].

These metrics represent just a small sample of what could be said about this multilayer network. In our talk we will discuss these and other findings. We hope the information presented will be useful to others interested in this kind of multilayered social network analysis, and perhaps become a springboard for future research.

Acknowledgements

This research was performed at Pacific Northwest National Laboratory (PNNL) operated by Battelle for the U.S. Department of Energy under Contract DE-AC05-76RL01830 and funded by the U.S. Department of Defense and by the Laboratory Directed Research and Development Program at PNNL.

References

[1] Cornell University Library. arXiv. http://arxiv.org/, 2016.

[2] L. C. Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1978.

[3] P. Jaccard. *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Impr. Corbaz, 1901.

[4] C. Kadushin. *Understanding social networks: Theories, concepts, and findings*. OUP USA, 2012.

[5] K.-M. Lee, B. Min, and K.-I. Goh. Towards real-world complexity: an introduction to multiplex networks. *The European Physical Journal B*, 88(2):1–20, 2015.

[6] M. E. Newman. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Physical review E*, 64(1):016132, 2001.

[7] Teach Thought. 100 scientists on Twitter by category. http://www.teachthought.com/uncategorized/100-scientists-on-twitter-by-category/, 2012.

[8] J. You. The top 50 science stars of Twitter. http://www.sciencemag.org/news/2014/09/top-50-science-stars-twitter/, 2014.

ENHANCED DETECTABILITY OF COMMUNITY STRUCTURE IN MULTILAYER NETWORKS THROUGH LAYER AGGREGATION

Dane Taylor, Saray Shai, Natalie Stanley and Peter J. Mucha

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

Many systems are naturally represented by a multilayer network in which edges exist in multiple layers that encode different, but often correlated, types of interactions. Using random matrix theory, we analyze the effect of layer aggregation on the detectability of community structure in networks in which the layers are drawn from a common stochastic block model. Our analysis provides insight into the common – but not well understood – practice of thresholding data matrices to obtain sparse network representations.

Layer Aggregation in Multilayer Networks

Multilayer networks [1] are ubiquitous in biological, social and technological systems, wherein different layers can encode different classes of categorical social ties, types of critical infrastructure, or a temporal network at different instances in time. In some applications network layers are correlated with one another and encode redundant information, and in such situations it is beneficial to seek a more concise representation in which certain layers are aggregated [2]. Identifying sets of repetitive layers (referred to as "strata" [3]) amounts to a clustering problem, and it is closely related to the topic of clustering networks in an ensemble of networks [4]. Much remains to be studied regarding *when* layer aggregation is appropriate and *how* it should be implemented.

In this research [5], we study the effect of layer aggregation on community structure in multilayer networks [6]. Community detection is a central pursuit for understanding the structure and function networks, and it is important to study fundamental limitations on detectability. That is, if the community structure is too weak, it cannot be found upon inspection of the network [7, 8, 9]. In Fig. 1, we depict in panels (A)–(D) the adjacency matrices of networks that have an increasing prevalence of community structure; the communities are undetectable in panel (A), whereas they are detectable (and have an increasing accuracy of inference) in the subsequent panels.

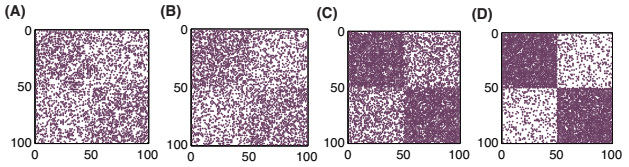


Figure 1: *Phase transition in detectability for a two-community stochastic block model (SBM).* Network adjacency matrices are shown for a single-layer network with $N = 100$ nodes, which are allocated into two equal-sized communities. Community structure is implemented by creating edges with probability p_{in} (p_{out}) for nodes in the same (different) communities. Tuning $\Delta = p_{in} - p_{out} > 0$ controls the prevalence of community structure. Panels (A)–(D) reflect networks with increasing values of Δ at a fixed mean edge probability, $\rho = (p_{in} + p_{out})/2 = 0.5$.

Multilayer Stochastic Block Modeling (MLSBM)

We analyze detectability limitations for a network model in which each layer $l \in \{1, \ldots, L\}$ is generated from a common stochastic block model (SBM). SBMs are a paradigmatic model for complex structure in networks and are particularly useful for studying limitations on detectability [7, 8, 9]. Despite growing interest in multilayer SBMs [3, 10, 11] – which we note focus on *multiplex* networks in which nodes are identical in every layer and edges are restricted to connecting nodes in the same layer [1] – the effect of layer aggregation on detectability has yet to be explored outside the infinite-layer, $L \to \infty$, limit [12].

We study detectability limitations for multiplex networks with finitely many layers. We develop analysis for networks layers that follow an SBM with equal-sized communities in which the probability of an edge (i, j) is given by p_{in} if nodes i and j are in the same community and p_{out} if they are in different communities. It is also convenient to study the mean edge probability $\rho = (p_{in} + p_{out})/2$ and the probability difference $\Delta = p_{in} - p_{out} > 0$. The adjacency matrices of several example SBMs are shown in Fig. 1.

Random Matrix Theory for Aggregated Layers

Following previous research [8, 9], we study detectability limits by developing random matrix theory for the modularity matrix $\mathbf{B}$. One benefit of this approach is that the community labels of nodes can be inferred using spectral bi-partitioning based on the dominant eigenvector $\mathbf{v}$ (i.e., $\mathbf{Bv} = \lambda_1 \mathbf{v}$, where λ_1 is the largest eigenvalue). In particular, provided that the community structure is sufficiently strong, the eigenvector entries $\{v_i\}$ are correlated with the community assignments: $v_i > 0$ for nodes $\{i\}$ in one community and $v_i < 0$ for nodes $\{i\}$ in the other community. By solving how $\mathbf{v}$ depends on SBM parameters ρ, and Δ, and on the number of layers L (which amounts to studying a spectral gap between the largest and second-largest eigenvalues of $\mathbf{B}$), we analyze how the detectability limit Δ^* is affected by the aggregation of layers. We focus on two methods of layer aggregation – (i) summing the layers' adjacency matrices that encode the network layers and (ii) thresholding this summation at some value $\tilde{L}$ – and find that the method of aggregation significantly influences detectability.

Scaling Behavior for Many Layers

Our main contribution is our analysis of the scaling behavior for how Δ^* behaves as the number of layers increases, $L \to \infty$. When the aggregate network corresponds to summing the layers' adjacency matrices, aggregation always improves detectability. In particular, the detectability limit Δ^* vanishes with increasing L and decays as $\mathcal{O}(L^{-1/2})$. Because the summation of L adjacency matrices can often yield a weighted and dense network – which increases the computational complexity of community detection [13] – we also study binary adjacency matrices obtained by thresholding this summation at some value $\tilde{L}$. We find that the detectability limit Δ^* is very sensitive to the choice of threshold $\tilde{L}$; however, we also find that there exist thresholds (e.g., the mean edge probability ρ for the case of two homogeneous communities) that are optimal in that the detectability limit also decays as $\mathcal{O}(L^{-1/2})$. We illustrate this scaling behavior in Fig. 2 by plotting a scale-invariant (that is, it becomes invariant for large N and L) variable $\Delta^*\sqrt{NL}$ for various values of ρ. Results are shown for an SBM with $N = 1000$ nodes.

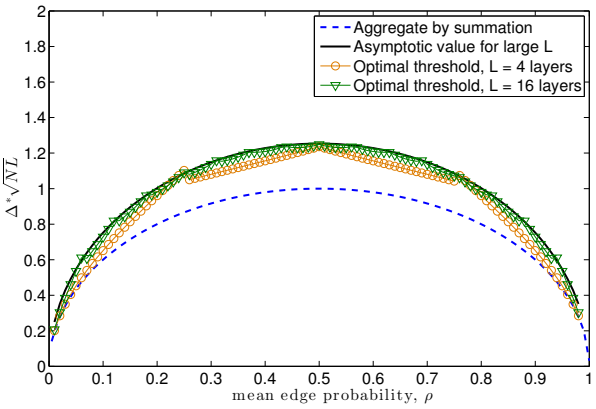


Figure 2: *Layer aggregation enhances the detectability of community structure.* We show the detectability limit Δ^* versus mean edge probability ρ for several methods of aggregating layers, including summing the layers' adjacency matrices (blue dashed curve) and thresholding this summation at an optimal threshold $\tilde{L}$ (symbols). We show results for two choices of number of layers, $L \in \{4, 16\}$. The solid black curve indicates our semi-analytical solution for Δ^* in the limit $L \to \infty$. Note that we have scaled the vertical axes by $\sqrt{NL}$, which reflects that Δ^* vanishes as $\mathcal{O}(1/\sqrt{NL})$ when either $L \to \infty$.

References

[1] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, M. A. Porter, *Journal of Complex Networks* 2(3), 203–271 (2014).

[2] M. De Domenico, V. Nicosia, A. Arenas and V. Latora, *Nature Communications* 6, 6864 (2015).

[3] N. Stanley, S. Shai, D. Taylor, P. J. Mucha, Preprint available online at http://arxiv.org/abs/1507.01826 (2015).

[4] J.-P. Onnela, D. J. Fenn, S. Reid, M. A. Porter, P. J. Mucha, M. D. Fricker, N. S. Jones, *Physical Review E* **86**, 036104 (2012).

[5] D. Taylor, S. Shai, N. Stanley and P. J. Mucha. Preprint available online at http://arxiv.org/abs/1511.05271 (2015).

[6] P. J. Mucha, T. Richardsons, K. Macon, M. A. Porter, J.-P. Onnela, *Science* 328(5980), 876–878 (2010).

[7] A. Decelle, F. Krzakala, C. Moore and L. Zdeborová, *Physical Review Letters* 107(6), 065701 (2011).

[8] R. R. Nadakuditi and M. E. J. Newman, *Physical Review Letters* 108(18), 188701 (2012).

[9] T. P. Peixoto, *Physical Review Letters* 111(9), 098701 (2013).

[10] T. P. Peixoto, Preprint available online at http://arxiv.org/abs/1504.02381 (2015).

[11] A. Ghasemian, P. Zhang, A. Clauset, C. Moore, and L. Peel, Preprint available online at http://arXiv.org/abs/1506.06179 (2015).

[12] Q. Han, K. Xu, and E. Airoldi, *Proceedings of the 32nd International Conference on Machine Learning*, 1511–1520 (2015).

[13] C. Aicher, A. Z. Jacobs and A. Clauset, *Journal of Complex Networks* 3(2), 221–248 (2015).

OPTIMIZING ADIABATIC QUANTUM PROGRAM COMPILATION USING A GRAPH-THEORETIC FRAMEWORK

Timothy D. Goodrich, Travis S. Humble, Blair D. Sullivan

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We introduce a graph-theoretic *virtual hardware* framework for embedding optimization problems into adiabatic quantum computers. We utilize biclique virtual hardware to emulate the existing TRIAD "native clique" embedding, while providing an additional mechanism for reducing hardware footprint based on program connectivity. Finally, we exploit bipartite structure in program graphs to obtain a novel algorithm that embeds larger programs while using less hardware resources than previous approaches.

Introduction

Modern advances in production-level quantum hardware have sparked significant interest in using quantum annealing to solve difficult combinatorial optimization problems; recent reports indicate that D-Wave System's 1152-qubit 2X hardware can solve problems 10^8 times faster than classical heuristics such as simulated annealing [3]. Unfortunately, compiling an input optimization problem into the hardware is itself computationally difficult, bottlenecked by the NP-hard GraphMinorEmbedding problem. While specific applications in computational chemistry [4], machine learning [7] and computer vision [6] have been run by hand-embedding the problems into the hardware, developing more generalized *embedding algorithms* is essential for accessibility and widespread adoption.

Embedding algorithms for cliques and sparse program graphs are well-studied. Early embedding algorithms from Choi [2] and Klymko et al. [5] embedded cliques into the D-Wave hardware, but these embeddings require a quadratic number of qubits to achieve the connectivity required for a clique. Cai et al. [1] took a fundamentally different approach, providing stochastic heuristics for incrementally constructing embeddings. While the core heuristic succeeded in reducing hardware footprint for sparse graphs, it also introduced the possibility of false negatives at a rate proportional to the program's edge density.

Unfortunately, between the two extremes of small cliques and large sparse graphs is a significant class of dense graphs not covered by prior methods. Additionally, previous hardware footprint reduction routines cannot be applied to these dense graphs. To address both issues, we introduce a *virtual hardware* layer that provides a simplified hardware interface, a framework for additional footprint reduction post-processing, and facilitates new embedding methods for specific graph classes such as nearly-bipartite program graphs.

Virtual Hardware

Generally, a virtual hardware graph will represent an allocation of qubits from the *physical* hardware into *virtual* qubits; we define this allocation as a minor embedding.

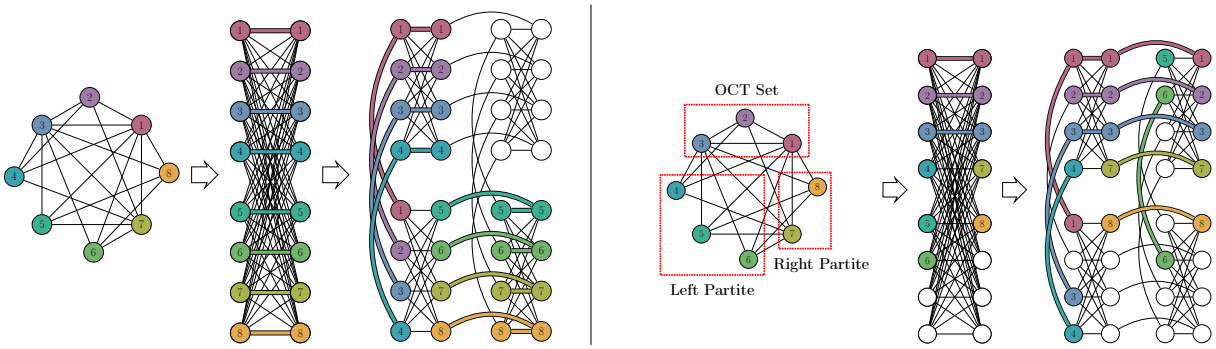


Figure 1: (Left) Embedding a dense, 8-vertex program graph using TRIAD-Embed and Native-Expand schemes. (Right) Embedding the same program using OCT-Embed and 2Ex-Expand schemes yields a 17% smaller embedding.

Definition 1 *Given two graphs G and H, a mapping ϕ is a* minor embedding *of H in G if:*

1. *$\phi(u) \cap \phi(v) = \emptyset$ for $u, v \in V(H)$ and $u \neq v$,*
2. *$\phi(v)$ induces a connected subgraph for $v \in V(H)$, and*
3. *The graph G_ϕ, formed by contracting every set $\phi(v)$ for $v \in V(H)$, contains a subgraph isomorphic to H.*

We then embed programs into the physical hardware by (1) embedding the program into the virtual hardware graph, and (2) expanding this embedding by replacing each logical qubit with an appropriate subset of its allocated physical qubits. We compute these two steps with *embedding* and *expansion* functions, respectively. This distinction is particularly useful for addressing the problems of embeddability and footprint reduction independently, enabling modular and specialized subroutines.

We apply this framework to identify a complete bipartite (biclique) virtual hardware in Chimera(L, M, N) – a generalization of D-Wave hardware defined as an $M \times N$ grid of biclique $K_{L,L}$ cells. This grid is composed of vertical and horizontal edges, and allocating vertices along these edges gives a $K_{LM,LN}$ biclique virtual hardware. Figure 1 shows Chimera(4, 2, 2) with a $K_{8,8}$ virtual hardware.

New Embedding Algorithms

We evaluate against Choi's TRIAD embedding [2], the best algorithm for dense graphs in fault-free hardware, which we can also emulate with schemes TRIAD-Embed and Native-Expand. Additionally, we reduce TRIAD's hardware footprint by reordering the virtual hardware embedding with a pair-exchange local search 2Ex-Expand scheme.

The largest improvement, however, comes from our scheme for efficiently embedding "bipartite-like" graphs into the biclique virtual hardware. We start by computing an odd cycle transversal decomposition of the program:

Definition 2 *An* odd cycle transversal (OCT) *for a graph G is a set of vertices whose deletion makes G bipartite.*

Computing an OCT set decomposes the program into a directly-embeddable bipartite graph and a "tangled" OCT set that is easily handled with TRIAD-Embed. This OCT-Embed scheme also benefits from allocating less qubits per bipartite vertex, enabling a more effective 2Ex-Expand footprint reduction. In total, coupling OCT-Embed with 2Ex-Expand leads to a new algorithm capable of embedding programs with up to $L(M + N)$ vertices (double the limit of complete graphs [2]), while experimentally constructing smaller embeddings than existing algorithms (Figure 2).

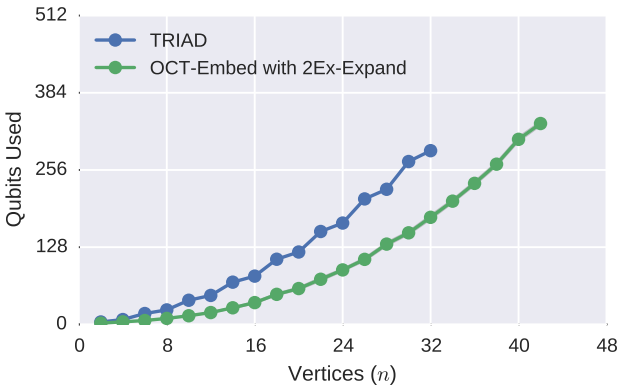


Figure 2: Embedding Erdős–Rényi(n, $p = 0.25$) graphs in Chimera(4, 8, 8); averaging over 10 trials per n. OCT-Embed with 2Ex-Expand embeds 30% more graphs and has up to a 50% smaller footprint than TRIAD.

Future Work

Virtual hardware is amenable to two major extensions. First, physical implementations of the Chimera graph typically contain *hard faults* (i.e. defective qubits), invalidating standard embeddings. Biclique virtual hardware proves resilient to this change and models it with missing edges, facilitating the development of appropriate embedding and expansion functions. Second, identifying optimal virtual hardware for a given program remains open beyond the "bipartite-like" case. The Chimera graph contains several other interesting virtual hardwares such as grids, dense cores, etc., and utilizing these structures could lead to improvements for other program classes.

References

[1] J. Cai, W. G. Macready, and A. Roy. A practical heuristic for finding graph minors. *arXiv:1406.2741*, 2014.

[2] V. Choi. Minor-embedding in adiabatic quantum computation: II. minor-universal graph design. *Quantum Information Processing*, 10(3):343–353, 2011.

[3] V. S. Denchev et al. What is the Computational Value of Finite Range Tunneling? *arXiv:1512.02206*, 2015.

[4] I. Kassal et al. Simulating chemistry using quantum computers. *Annual Review of Physical Chemistry*, 62(1):185–207, 2011.

[5] C. Klymko, B. D. Sullivan, and T. S. Humble. Adiabatic quantum programming: minor embedding with hard faults. *Quantum information processing*, 13(3):709–729, 2014.

[6] H. Neven, G. Rose, and W. G. Macready. Image recognition with an adiabatic quantum computer I. Mapping to quadratic unconstrained binary optimization. *arXiv:0804.4457*, 2008.

[7] K. Pudenz and D. Lidar. Quantum adiabatic machine learning. *Quantum Information Processing*, 12(5):2027–2070, 2012.

REDUNDANCY, DEGENERACY, AND ROBUSTNESS IN PROTEIN-INTERACTION NETWORKS

Alice C.U. Schwarze, Mason A. Porter, Jonny Wray

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

To assess the impact of a drug on a biological system, *network pharmacology* uses impact measures that describe the effect of perturbations on networks. The plausibility of different impact measures depends on their ability to capture the robustness of biological systems. The literature has highlighted links between redundancy, degeneracy, and robustness. We investigate how to quantify the biological notions of redundancy and degeneracy with analysis of cellular networks by adapting an information-theoretic approach that was previously applied in theoretical neuroanatomy.

Introduction

Network pharmacologists model complex biological systems as networks and use tools from network analysis to gain insights into biological problems. This approach can facilitate multi-target drug discovery [12]. For example, one can associate biological systems with protein-interaction networks (PINs) and chemical compounds affecting these systems with network perturbations. By analysing structural changes induced by a perturbation on a PIN, it is possible to elucidate the systemic impact of a chemical compound on a biological system.

It is unclear how biological function relates to structural properties of the corresponding PINs. A better understanding of this relationship is needed to define suitable measures for the above perturbations. Theoretical and computational work should guide and assist experimental research in this endeavour. Purely experimental investigation of the relationship between structure and function of PINs is prohibitively expensive and difficult, because uncertainty in data on protein interactions, individual differences between cells, and a wide range of environmental factors for which one needs to control make it difficult to identify correlations between structural properties of PINs and biological function in experiments.

In collaboration with e-Therapeutics, we aim to develop computational methods for identifying structural properties that may be linked to the integrity of biological functions in a cell. To that end, we identify links between biological and structural robustness of PINs and combine insights from evolutionary biology on the robustness of complex biological systems with findings on the percolation properties of PINs and random-graph models thereof.

Impact Measures and their Plausibility

Consider the *impact* I_S of a perturbation as the relative change in a structural property S of the network,

$$I_S = \frac{|S(\mathcal{N}^*) - S(\mathcal{N})|}{S(\mathcal{N})}\,, \tag{1}$$

where $\mathcal{N}$ denotes the network before perturbation and $\mathcal{N}^*$ denotes the network after perturbation. It is not clear which structural properties S lead to meaningful impact measures I_S. One can require an impact measure to align with the notion that biological networks are generally robust to random failures but vulnerable to targeted attacks [10, 11]. This implies that one expects to observe a large impact when the set of nodes to be removed is chosen according to some ranking of the importance of the nodes and a small impact when the set of nodes is chosen uniformly at random.

Robustness of Random-Graph Models for Protein-Interaction Networks

When investigating a network's robustness, researchers often consider to the robustness of the network's mean shortest path length to removal of the highest-degree nodes [1, 6]. However, it is unclear if this robustness property is the best choice for characterising the robustness of biological functions. Using impact measures (see Eq. (1)), we analysed the robustness of five structural properties (fragmentation, mean shortest path length, global and mean local clustering, and communicability [3]) to targeting nodes by degree, node betweenness, eigenvector, and subgraph centrality [4].

We tested several random-graph models [7, 8, 10] for their ability to capture the robustness properties of PINs. We found that the configuration model [7] — a commonly

used null model — describes impacts of perturbations in PINs better than two variants of the vertex-duplication model [8, 10], which are popular for modelling PINs. However, none of the models captured the consistently low impact of attacks when targeting nodes by subgraph centrality as opposed to when targeting by other centrality measures. This suggests that cyclic graphlets — a structural property highlighted by subgraph centrality — can affect a PIN's robustness in a way that the above graph models do not take into account.

Robustness of Biological Networks

Researchers have suggested that cycles in a network's structure play an important role for key characteristics of biological systems in the context of other network applications [2, 5]. The biological literature has highlighted links between *redundancy*, *degeneracy*, and robustness [11]. Here, we refer to redundancy as the existence of structurally similar parts of a network that perform the same function. Degeneracy indicates the existence of parts that are structurally different but can perform the same function [9]. In early work on structural robustness, network scientists demonstrated that redundancy can lead to robustness in networks with power-law degree-distributions [1, 6].

As indicated in Refs. [5, 11], mechanisms that lead to robustness in biological networks likely differ from the one proposed in the early works [1, 6] and rather rely on degeneracy of a network's structure than its redundancy.

Quantifying Degeneracy

We use information-theoretic measures for quantifying biological redundancy and degeneracy that were proposed by Tononi et al. [9] for linking the structure of cortical networks to biological characteristics. We assume linear noisy dynamics of the form

$$\mathbf{x}\,dt + d\mathbf{x} = \mathbf{A}\mathbf{x}\,dt + d\mathbf{w}\,, \tag{2}$$

with state vector $\mathbf{x}$, PIN adjacency matrix $\mathbf{A}$ (with weighted elements), and Wiener process $\mathbf{w}$. If Eq. (2) has a stationary state, one can derive expressions for entropy and mutual information from the state's stationary covariance matrix [2, 9]. The *functional overlap* indicates the existence of parts of a network — either structurally similar or structurally dissimilar — that can perform the same function. Defining a subset of the network as the output or observable set $\mathbf{o}$, we can calculate the functional overlap

$$\mathrm{FO}(\mathbf{x},\mathbf{o}) = \sum_{j=1}^{n-1} [\mathrm{MI}(x_j,\mathbf{o})] - \mathrm{MI}(\mathbf{x},\mathbf{o})\,, \tag{3}$$

where x_j is the j-th element of $\mathbf{x}$ and MI is the mutual information. The system's degeneracy is

$$D(\mathbf{x},\mathbf{o}) = \sum_{k=1}^{n-1} \left[\frac{k}{n}\mathrm{FO}(\mathbf{x},\mathbf{o}) - \langle \mathrm{FO}(\mathbf{x}_k,\mathbf{o})\rangle_k\right]\,, \tag{4}$$

where $\mathbf{x}_k$ is a subset of $\mathbf{x}$ of size k and $\langle\cdot\rangle_k$ is the mean over all $\mathbf{x}_k$.

In analogy to an approach presented in Ref. [2], we investigate the relationship between small-scale structures in $\mathbf{A}$ and D to link D to graphlet frequencies in the network.

References

[1] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, 2000.

[2] L. Barnett, C. L. Buckley, and S. Bullock. Neural complexity and structural connectivity. *Physical Review E*, 79(5):051914, 2009.

[3] E. Estrada and N. Hatano. Communicability in complex networks. *Physical Review E*, 77(3):036111, 2008.

[4] E. Estrada and J. A. Rodriguez-Velazquez. Subgraph centrality in complex networks. *Physical Review E*, 71(5):056103, 2005.

[5] P. M. Gleiss, P. F. Stadler, A. Wagner, and D. A. Fell. Relevant cycles in chemical reaction networks. *Advances in Complex Systems*, 4(02n03):207–226, 2001.

[6] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.

[7] M. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[8] R. Pastor-Satorras, E. Smith, and R. V. Solé. Evolving protein interaction networks through gene duplication. *Journal of Theoretical Biology*, 222(2):199–210, 2003.

[9] G. Tononi, O. Sporns, and G. M. Edelman. Measures of degeneracy and redundancy in biological networks. *Proceedings of the National Academy of Sciences USA*, 96(6):3257–3262, 1999.

[10] A. Vázquez, A. Flammini, A. Maritan, and A. Vespignani. Modeling of protein interaction networks. *Complexus*, 1(1):38–44, 2003.

[11] J. M. Whitacre. Biological robustness: paradigms, mechanisms, and systems principles. *Front Genet*, 3:67, 2012.

[12] M. Young, S. Zimmer, and A. Whitmore. Drug molecules and biology: Network and systems aspects. In R. Morphy and C. J. Harris, editors, *Designing Multi-Target Drugs*, pages 32–49. Royal Society of Chemistry, Cambridge, UK, 2012.

MEASURING THE SOCIAL FLOW OF INFORMATION AND ITS ROLE IN PREDICTION

James P. Bagrow[1] and Lewis Mitchell[2]

[1]Mathematics and Statistics, University of Vermont, USA

[2]School of Mathematical Sciences, University of Adelaide, Australia

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Massive datasets of human activity are now available, revolutionizing research on human dynamics and computational social science. We study the complete text streams of thousands of Twitter users and their online followers. Twitter is an online microblogging platform where users interact with one another and post short messages called tweets. Information flows across the social network by means of these tweets. But how much information flows, how much influence do users have on one another, and can we accurately measure these effects?

Treating each Twitter user's text stream as a symbolic (word) time series, the entropy rate measures how much information about a future word choice is available in the past history (Fig 1A). However, the basic Shannon entropy underestimates the information in, e.g., natural language because it only considers the frequencies of words and not correlations in their occurrence. We apply theorems from data compression to compute an estimated *correlated entropy* that accounts for both temporal ordering and long-range correlations in the data [1]. The correlated entropy rate estimates the inherent uncertainty about someone's future word choice. We then relate this uncertainty to the *predictability* Π [2, 3], a bound on how well a perfect prediction algorithm can guess, in this case, the next word a user will post.

Crucially, this technique can also measure the information transfer between pairs of users (denoted egos and alters) [4]. To do this we instead compute the cross-entropy to estimate how much information about the ego's future word choice is present in the alter's past words[1] (Fig. 1B).

We find that most online users have a predictability of over 50% (Fig. 1C, red distribution). The cross-entropy is not sharply peaked however, meaning there is a range of information flow between egos and alters (Fig. 1C, blue distribution). Some alters contain nearly as much information about the ego as the ego itself, but other ego-alter pairs show little information flux. For comparison, random pairs of users (Fig. 1C, grey distribution) tend to have lower predictability, $\Pi < 0.3$, for many pairs.

Figure 1 is limited to a single pair of users, but in principle more information may be available about the ego from the *set* of alters. In Fig. 2 we study the cross-entropy (Fig. 2A) and predictability (Fig. 2B) as we increase the number of alters examined. Incorporating the text of multiple alters greatly increases the potential information and predictability had from the ego. Random controls, where either ego-alter pairs were shuffled or messages were shuffled between users, do not show this effect.

Figure 2 indicates that the social neighborhood of a social media user contains distinct and potentially actionable information about the user, even when that user is completely excluded.

Taken together, these results provide new quantitative bounds on information transfer in social networks, useful for better understanding the spread of ideas and influence in human populations.

[1]We exclude *retweets*, where one user directly quotes the exact text of another user, and consider only original, primary source text.

References

[1] I Kontoyiannis, P.H. Algoet, Y. M. Suhov, and A. J. Wyner. Nonparametric entropy estimation for stationary processes and random fields, with applications to english text. *IEEE Transactions on Information Theory*, 44(3):1319–1327, May 1998. 00104.

[2] M. Feder and N. Merhav. Relations between entropy and error probability. *IEEE Transactions on Information Theory*, 40(1):259–266, January 1994. 00123.

[3] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, November 2012. 35580.

[4] G. Ver Steeg and A. Galstyan. Information transfer in social media. In *Proceedings of the 21st international conference on World Wide Web*, pages 509–518. ACM, 2012.

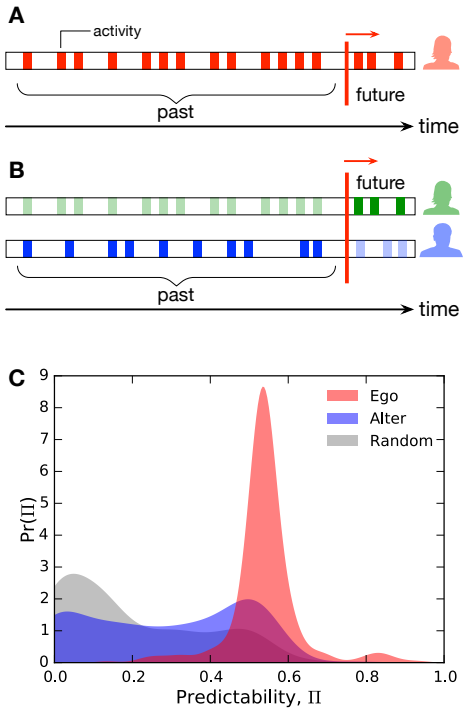


Figure 1: (**A**) We measure the correlated entropy rate and predictability to understand how much information about a user's future activity is present in her historical activity. (**B**) To measure information flux and influence we extend this measure to the cross-entropy, capturing how much information about an ego's (green) future is present in an alter's (blue) past. (**C**) These correlated entropies can be related to the predictability Π, an upper bound on the maximum predictive accuracy of a perfect prediction algorithm ($\Pi = 1$ indicates perfect accuracy and no mistakes, $\Pi = 0$ indicates a complete absence of predictive potential). The distribution of Π for the ego (red) is sharply peaked around 0.55; a perfect algorithm has the potential to predict a user's activity with over 50% accuracy. Considerable information on the activity of many (but not all) egos is available in the alter's past (blue), while a random alter generally provides little information (grey; $\Pi < 0.3$ for most pairs).

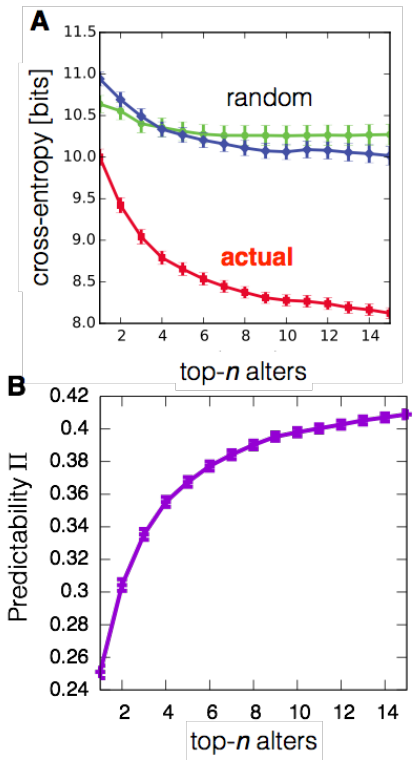


Figure 2: Estimating the information about the ego not within a single alter but from multiple alters simultaneously. (**A**) We rank alters by contact frequency on Twitter (number of mentions). The cross-entropy between the alter(s) and the ego drops quickly as more alter text data is used. Random controls, computed by randomizing ego-alter pairs or by randomly shuffling tweet messages between users, have far larger cross-entropies (less information) and do not display the steep decrease in entropy (increase in information) that real tweets and real alters show. (**B**) Computing the predictability from the cross-entropy shows that, while self-predictability can exceed 50%, the alters of an ego by themselves provide nearly as much information, with over 40% predictability.

INCORPORATION OF GAUSSIAN ATTRIBUTE DATA IN STOCHASTIC BLOCK MODEL INFERENCE

Natalie Stanley, Roland Kwitt, Marc Niethammer, Peter J. Mucha

SIAM Workshop on Network Science 2016
July 15-16 · Boston

The stochastic block model (SBM) [3] is a probabilistic model for community structure in which nodes in a community are connected to nodes within and between communities in a characteristic way. Typically, fitting an SBM to a network only considers the connectivity patterns between nodes encoded in the adjacency matrix. However, recently, there have been advancements in the incorporation of node attributes, or metadata, in SBM inference [7, 1, 4]. These existing approaches are applicable for dealing with binary attributes [7], modularity-based community detection [1], and for incorporating a single piece of categorical or continuous metadata [4]. In this work, we seek to extend the classic inference procedure required in fitting the SBM to take into account multiple measured continuous attributes. This can be particularly useful in cases where either the connectivity or attribute information is noisy and uninformative and hence complicates the identification of meaningful communities.

We consider the case where each node in the network has an associated vector of continuous attributes that is dependent on its community membership. Using this information, we have developed a model to incorporate attribute information ($\mathbf{X}$) and network connectivity ($\mathbf{A}$) to fit an SBM. To do this, we assume that node i in community c has an associated p-dimensional vector of attribute data, $\mathbf{x}_i$, that is drawn from a multivariate Gaussian distribution, parameterized by $\mathbf{\Psi}_c = \{\boldsymbol{\mu}_c, \mathbf{\Sigma}_c\}$, where $\boldsymbol{\mu}_c$ and $\mathbf{\Sigma}_c$ specify mean and covariance parameters, respectively.

Figure 1 shows the associated graphical model, with a notable foundational model assumption that $\mathbf{X}$ and $\mathbf{A}$ are assumed to be conditionally independent, given the node-to-community assignment labels, $\mathbf{Z}$. Further, $\mathbf{X}$ is specified by the appropriate multivariate Gaussian parameters, generically given by $\mathbf{\Psi}$, and hence for a network with K communities, K multivariate Gaussians are fit to the data in $\mathbf{X}$. The adjacency matrix $\mathbf{A}$ can be generated according to $\boldsymbol{\theta}$, the $K \times K$ matrix of stochastic block model parameters. To learn the most appropriate model parameters maximizing the model likelihood, we apply expectation maximization (EM) [2], and we can compute the probability that node i belongs to community c,

$$p(z_{ic} = 1 \mid \mathbf{x}_i, \mathbf{a}_i) = \frac{p(\mathbf{x_i} \mid z_{ic} = 1)p(\mathbf{a}_i \mid z_{ic} = 1)\pi_c}{\sum_{c=1}^{K} p(\mathbf{x_i} \mid z_{ic} = 1)p(\mathbf{a}_i \mid z_{ic} = 1)\pi_c}. \quad (1)$$

Note that here π_c gives the probability of belonging to community c, while $\mathbf{a}_i$ gives the connectivity pattern (i.e. row in the adjacency matrix) for node i.

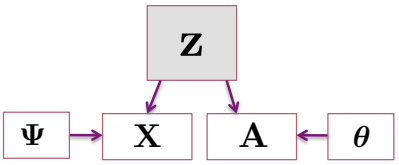


Figure 1: **Graphical model for our approach**. Ultimately, we seek $\mathbf{Z}$ that gives node-to-community assignments. We assume that given $\mathbf{Z}$, connectivity patterns encoded through the adjacency matrix $\mathbf{A}$ and attribute information, $\mathbf{X}$ can be inferred, and these two sources are conditionally independent given the node-to-community assignments. The adjacency matrix, $\mathbf{A}$, is parameterized by a matrix of SBM edge probability parameters, $\boldsymbol{\theta}$, while attributes for nodes within a community are parameterized by $\mathbf{\Psi} = \{\boldsymbol{\mu}, \mathbf{\Sigma}\}$, giving the mean and covariance matrix for a multivariate gaussian, respectively.

We can show that this approach performs well on synthetic networks in regimes where the attribute and adjacency matrices are both noisy. These circumstances correspond to conditions where community structure is not detectable from the graph alone, and clusters are undiscernable according to the attribute information. In experiments shown in figure 2, networks were generated from a stochastic block model with within-community edge probability, $p_{in} = .25$ and between community probability, $p_{out} = .1$. An example network is shown in figure 2a.

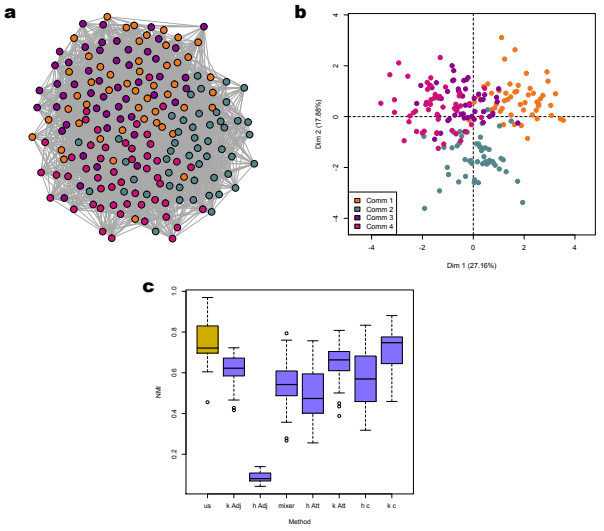


Figure 2: **Synthetic Experiments.** 25 network + attribute datasets were created. Networks have $N = 200$ nodes, $K = 4$ communities $p_{in} = .25$, $p_{out} = .1$. Each network, attribute pair corresponds to one synthetic experiment. **a.** Visualization of the network generated for one synthetic experiment. **b.** PCA plot of the attribute vectors associated with the nodes from the network in the example synthetic experiment shown in a. **c.** Distribution of NMIs over the 25 different synthetic experiments. The performance of our method "us"(gold) in comparison to several baselines (purple).

Similarly, for each community, we generated its attributes according to a random mean vector, $\boldsymbol{\mu}_c$ and a covariance matrix, $\boldsymbol{\Sigma}$ with $\boldsymbol{\Sigma} = \text{diag}(.25)$. The 2-dimensional PCA plot for nodes according to these attributes is shown in figure 2b, where there is an obvious mixing of classes. Figure 2c. shows the distribution of normalized mutual information (NMI) obtained from running this experiment on 25 network+attribute pairs through our approach (gold, labeled "us") in comparison to several baselines (purple).

As an application for this model, we can apply it for image segmentation tasks, where the objective is to label pixels according to the object in the image to which they belong. Graph-based segmentation has shown to be useful [6], but generative models for segmentation are lacking. Applying our model to segmentation tasks allows us to incorporate spatial regularization between pixels through the adjacency matrix and features of the image, such as color and texture, through the attribute matrix. Figure 3 gives an example of an image of an elephant from the dataset in [5] segmented according to our approach.

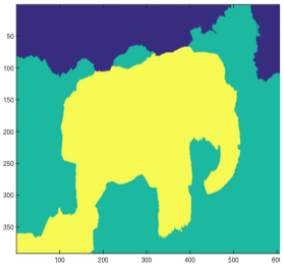


Figure 3: **Example segmentation result under our model.** We apply our model to segmentation tasks, where the adjacency matrix encodes spatial proximity between image pixels and the attribute information reflects color and texture information.

The methods and applications discussed in this work make important contributions to the networks and image analysis communities. First, specifying a probabilistic model for networks with node attributes is important for dealing with the increasing amount of annotated network data across fields. Finally, the well established methods within the image analysis community for techniques such as parameter inference and regularization could advance the network science field in novel ways.

References

[1] D. Combe, C. Largeron, M. Géry, and E. Egyed-Zsigmond. I-louvain: An attributed graph clustering method. In *Advances in Intelligent Data Analysis XIV*, pages 181–192. Springer, 2015.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[3] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[4] M. Newman and A. Clauset. Structure and inference in annotated networks. *arXiv preprint arXiv:1507.04001*, 2015.

[5] J. Santner, T. Pock, and H. Bischof. Interactive multi-label segmentation. In *Proceedings 10th Asian Conference on Computer Vision (ACCV), Queenstown, New Zealand*, November 2010.

[6] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[7] J. Yang, J. McAuley, and J. Leskovec. Community detection in networks with node attributes. In *Data mining (ICDM), 2013 ieee 13th international conference on*, pages 1151–1156. IEEE, 2013.

SPECTRAL CLUSTERING FOR BILLION-NODE GRAPHS

Da Zheng[1], Disa Mhembere[1], Youngser Park[2], Joshua Vogelstein[3], Carey E. Priebe[2], and Randal Burns[1]

[1]Department of Computer Science, Johns Hopkins University
[2]Department of Applied Mathematics and Statistics, Johns Hopkins University
[3]Institute for Computational Medicine, Department of Biomedical Engineering, Johns Hopkins University

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We implement semi-external memory (SEM) spectral clustering for massive graphs by utilizing solid-state drives (SSDs). We extend the Anasazi eigensolvers [2] in the Trilinos Project [10] to use SSDs for computing eigenvalues of a massive graph. We further perform SEM K-Means on the eigenvectors to cluster vertices. Our SEM spectral clustering implementation takes less than 4 hours to cluster a graph with 3.5 billion vertices and 129 billion edges in a single machine. To our knowledge, this is the first implementation that can scale to billion-node graphs and the code is released in https://github.com/zheng-da/FlashX.

Background

Spectral clustering is a very useful technique for clustering vertices in a graph. There are variants of spectral clustering algorithm in the literature [19, 15] and each of them computes eigenvectors of a graph and performs K-Means on the eigenvectors. Spectral clustering has relatively low computation complexity when compared to other graph clustering algorithms [11, 8, 14], but it is still computationally challenging to apply this technique to massive graphs such as Facebook's social network and today's web graph, that have billions of vertices.

One of the challenges is to compute eigenvalues and eigenvectors of a large matrix. The computation complexity of computing all eigenvalues of a square matrix is $O(n^3)$ [16], where n is the number of rows and columns of the matrix. Numerous algorithms [12, 4, 18, 3] have been developed to compute a small number of eigenpairs of a large matrix. We can apply these algorithms to a large graph to compute its eigenvalues. However, when computing eigenpairs of a graph at the billion scale, neither the sparse matrix that represents a graph nor the vector subspace fits in the RAM of a single machine. Large-scale eigenvalue problems are generally solved in a supercomputer [2, 9], where the aggregate memory is sufficient to store the sparse matrix and the vector subspace. Sparse matrix multiplication on graphs in distributed memory leads to significant network communication and is usually bottlenecked by the network. As such, this operation requires a fast network to achieve performance. However, a supercomputer with fast network communication is not accessible to many people.

Even though K-Means is less computationally intensive than computing eigenvalues, it can easily become the bottleneck to scaling spectral clustering. K-Means is an iterative algorithm, where when given a matrix, $\mathbb{R}^{nd}$, to be clustered into k clusters, the computation complexity of each iteration is $O(ndk)$. In many cases, we need to try different k values and run K-Means many times to find the *right* number of clusters. As such, performing K-Means on the eigenvectors of a billion-node graph can become as expensive as computing eigenvalues.

Semi-external memory Eigensolver

We develop an SSD-based eigensolver framework called FlashEigen, which extends Anasazi eigensolvers to SSDs, to compute eigenvalues of a graph with hundreds of millions or even billions of vertices in a single machine. When computing the eigenvalues of a graph, the most computationally expensive operation in an eigensolver is sparse matrix multiplication. FlashEigen performs sparse matrix multiplication in a semi-external memory fashion, i.e., it keeps the sparse matrix on SSDs and the dense matrix in memory. Besides sparse matrix multiplication, Anasazi eigensolvers require a set of dense matrix operations on the vectors in the subspace. FlashEigen provides both in-memory and external-memory implementations for the dense matrix operations. As such, when computing eigenvalues, FlashEigen keeps a specified number of vectors in the subspace in memory and the remaining vectors on

Graph datasets	# Vertices	# Edges	Directed
Friendster [21]	65M	1.7B	No
RMAT-40 [6]	100M	3.7B	No
RMAT-160 [6]	100M	14B	No
Page [20]	3.4B	129B	Yes

Table 1: Graph data sets.

SSDs. The number of vectors in memory is determined by the memory size available to the eigensolver. The implementation details of FlashEigen is described in [22].

Semi-external memory K-Means (SEM-kmeans)

When run in-memory the best memory bound achievable is $O(nd+kd)$. When n is in the billions it quickly becomes unfeasible to compute without distributed computing. We provide an SEM implementation, SEM-kmeans, to overcome this scalability challenge. We extend the original definition of the semi-external memory model [1, 17], defining it in this setting to be an in-memory state of $O(n)$ during computation. SEM-kmeans has a memory bound of $O(n + Tkd)$, where T is the number of threads and $Tkd << nd$.

SEM-kmeans achieves high performance by mostly merging the 2-phases of Llyod's algorithm [13], via unshared per-thread data structures. SEM-kmeans is optimized by minimizing remote memory reads in NUMA multiprocessor architectures and maximally utilizing vectorized CPU instructions. Furthermore, we modify the triangle inequality computation pruning technique [7] and reduce it's memory requirements from $O(n+Tkd+nd)$ to $O(2n+Tkd)$ by eliminating the lower bound matrix that we empirically determined to only prune less than 5% of computation in real world graphs [5]. We further add support for matrix-like computations within the FlashGraph [23] engine and develop a per-matrix-partition row cache to reduce the effect of I/O latency by an order of magnitude compared to FlashGraph's page cache.

Performance evaluation

We conduct all experiments on a non-uniform memory architecture machine with four Intel Xeon E7-4860 processors, clocked at 2.6 GHz, and 1TB memory of DDR3-1600. Each processor has 12 cores. The machine installs 24 OCZ Intrepid 3000 SSDs whose aggregate capacity is around 10TB. The machine runs Ubuntu Linux 14.04. We use 48 threads for the experiments.

We use the real-world graphs in Table 1 for evaluation.

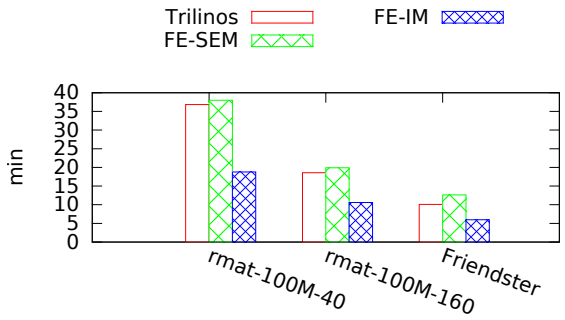


Figure 1: The runtime of the Trilinos KrylovSchur, FlashEigen-SEM and FlashEigen-IM KrylovSchur eigensolvers on smaller graphs in Table 1 when computing eight eigenvalues. The Page graph is too large for Trilinos KrylovSchur and FlashEigen-IM KrylovSchur.

Stage	Runtime	Memory
Eigen-decomposition	3.7 hours	120 GB
K-Means	3 min	44 GB

Table 2: Preliminary results of runtime and memory consumption of spectral clustering on the Page graph.

The smallest graph is the Friendster graph with 65 million vertices and 1.7 billion edges and the largest graph is the Page graph with 3.4 billion vertices and 129 billion edges, almost two order of magnitude larger than the Friendster graph. We use RMAT [6] to generate two synthetic graphs to fill the gap between the smallest graph and the largest graph.

FlashEigen in semi-external memory (FlashEigen-SEM) is able to at least achieve 40%-60% performance of its in-memory implementation (FlashEigen-IM) and has performance comparable to the Anasazi eigensolvers (Figure 1). In this experiment, FlashEigen-SEM only keeps the dense matrices involved in sparse matrix multiplication in memory and all the vectors in the subspace and the sparse matrix on SSDs, which results in minimal memory consumption.

We further demonstrate the efficiency and scalability of FlashEigen and SEM-KMeans on the Page graph with 3.4 billion vertices and 129 billion edges. It takes about four hours to compute eight eigenvalues of the billion-node graph using 120 GB memory.

References

[1] J. Abello, A. L. Buchsbaum, and J. R. Westbrook. A functional approach to external graph algorithms. In *Algorithmica*. Springer-Verlag, 1998.

[2] P. Arbenz, U. L. Hetmaniuk, R. B. Lehoucq, and R. S. Tuminaro. A comparison of eigensolvers for large-scale 3D modal analysis using AMG-preconditioned iterative methods. *International Journal for Numerical Methods in Engineering*, 2005.

[3] P. Arbenz, U. L. Hetmaniuk, R. B. Lehoucq, and R. S. Tuminaro. A comparison of eigensolvers for large-scale 3D modal analysis using AMG-preconditioned iterative methods. *International Journal for Numerical Methods in Engineering*, 2005.

[4] D. Calvetti, L. Reichel, and D. C. Sorensen. An implicitly restarted lanczos method for large symmetric. . . *ETNA*, 2:1–21, 1994.

[5] D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-mat: A recursive model for graph mining. In *SDM*, volume 4, pages 442–446. SIAM, 2004.

[6] D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-MAT: A recursive model for graph mining. In *In SDM*, 2004.

[7] C. Elkan. Using the triangle inequality to accelerate k-means. In *ICML*, volume 3, pages 147–153, 2003.

[8] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[9] V. Hernandez, J. E. Roman, and V. Vidal. SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Trans. Math. Softw.*, 2005.

[10] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley. An overview of the Trilinos project. *ACM Trans. Math. Softw.*, 2005.

[11] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 49(2):291–307, 1970.

[12] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 1950.

[13] S. P. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[14] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[15] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*. 2002.

[16] V. Y. Pan and Z. Q. Chen. The complexity of the matrix eigenproblem. In *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing*, 1999.

[17] R. Pearce, M. Gokhale, and N. M. Amato. Multithreaded asynchronous graph traversal for in-memory and semi-external memory. In *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, 2010.

[18] G. W. Stewart. A KrylovSchur algorithm for large eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 2002.

[19] D. L. Sussman, M. Tang, D. E. Fishkind, and C. E. Priebe. A consistent adjacency spectral embedding for stochastic blockmodel graphs. *Journal of the American Statistical Association*, 107(499):1119–1128, 2012.

[20] Web graph. http://webdatacommons.org/hyperlinkgraph/, Accessed 4/18/2014.

[21] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, 2012.

[22] D. Zheng, R. Burns, J. Vogelstein, C. E. Priebe, and A. S. Szalay. An SSD-based eigensolver for spectral analysis on billion-node graphs. *CoRR*, abs/1602.01421, 2016.

[23] D. Zheng, D. Mhembere, R. Burns, J. Vogelstein, C. E. Priebe, and A. S. Szalay. FlashGraph: Processing billion-node graphs on an array of commodity SSDs. In *13th USENIX Conference on File and Storage Technologies (FAST 15)*, 2015.

PERSISTENT HOMOLOGY OF DYNAMIC ONLINE REFERRAL TRAFFIC NETWORKS

Tyler Foxworthy

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

The topology and scale-free characteristics of the internet have been extensively characterized on the basis of the distribution of directed links between web domains [1]. Due to the inherent difficulties in the analysis and interpretation of complex networks on multiple scales, dynamic online referral traffic networks, reflecting the functional connectivity of the web through inter-domain user movement over time, have not been widely explored. In this report we demonstrate a method for computing persistent homology of referral networks based on inter-domain traffic patterns over time. Quantifying the persistence of n-dimensional simplical complexes arising from these networks as they evolve is of significant value for marketers aiming to maximize user exposure to advertising content across a minimal number of sites.

Background

Like information transfer between neurons and regions of the brain, the flow of user traffic between web domains often exhibits significant temporal variation due to marketing efforts, seasonality, and countless other exogenous influences [2]. These temporal effects introduce significant topological complexity at multiple scales by introducing transient paths and clusters [3].

To measure the stability of inter-domain connections over time while accounting for periodic and asymmetric traffic profiles, we computed the entropy rate of the connections modeled as a binary renewal process, a method having been previously applied to quantify information transfer between neurons [4].

In recent years, persistent homology, a method of computational topology, has proven to be a highly efficient mechanism for computing topological features of weighted directed networks at multiple scales that are not accessible by conventional combinitorial or spectral techniques [5, 2, 3, 6]. We used persistent homology to identify topological features of referral networks, with those features which persist across maximal range of entropy rates being the most significant.

Methods

A year of monthly estimated inter-domain traffic volume data was obtained for a network of 27753 web domains comprising the known referral ecosystem of five competitive e-retailers using SimilarWeb, a commercial data service providing competitive intelligence data aimed at marketers. Referrals reflect the flow of users from one web domain to another via hyperlinks and paid advertisements.

From this data, we formed a directed graph weighted by the entropy rate of a binary sequence $\mathbf{X}^{i,j} \in \{0,1\}^n$ denoting the transfer of traffic from domain i to j within each measurement interval t. Assuming the sequences follow a renewal process, a generalized Poisson process based on the distribution of arrival intervals between transfers, the entropy rate of each binary string was estimated as

$$H(\mathbf{X}) = E(\mathbf{X})H(\mathbf{Y}) = -E(\mathbf{X})\sum_j q_j \log q_j \qquad (1)$$

where $\mathbf{Y}$ denotes a i.i.d geometric sequence of inter arrival times of $\mathbf{X}$ such that $\{Y_i = t_{i+1} - t_i\}$ for $X_t = 1$ and q_j the empirical distribution of unique arrival times [4].

We parameterized a simplical complex over binary networks $\mathcal{B}(V, \epsilon)$ by a filtration parameter ϵ such that $i, j \in V$ are adjacent if $\omega(i,j) < \epsilon$, where η is the maximum transfer entropy in the network and

$$\omega(i,j) = \begin{cases} 1 - \eta^{-1}H(\mathbf{X}^{i,j}) & : (\sum_t X_t^{i,j}) > 1 \\ 1 & : (\sum_t X_t^{i,j}) = 1 \end{cases}$$

The Vietoris-Rips construction [7, 2] was used to compute the persistent homology of sampled subnetworks from this parameterized complex on the filtration interval $\epsilon \in [0,1]$.

Results

Figure 1 reflects the global filtration parameter distribution for the referral network based on the inter-domain renewal entropy, grouped by the total number of intervals $\mathbf{X}_t > 0$. The progressively overlapped distributions are indicative of the ability of the renewal entropy estimator

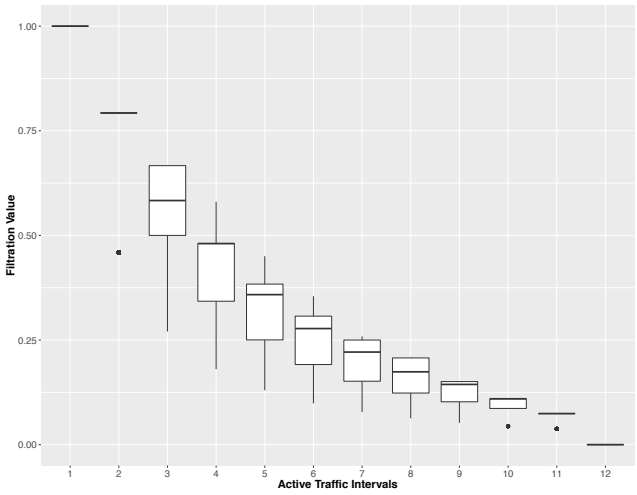


Figure 1: Edge filtration parameter density by total connections measured across referral network

to meaningfully distinguish between equal rank sequences with varying stability (e.g. 1001001001 vs 1011000010).

Figures 3 and 2 show the persistent homology (represented by a barcode diagram [6]) and corresponding sequence of filtrations for a randomly sampled subnetwork. Although not shown for the sake of brevity, the majority of sampled networks displayed similar topological features and infrequent higher dimensional persistent structures.

Conclusions

We have briefly outlined the procedure for computing persistent homology of referral networks based on the renewal entropy of inter-domain traffic sequences. Identifying persistent n-dimensional simplical complexes of referral networks will allow marketers to identify sets of domains with stable referral traffic behavior and can be used to optimize marketing strategy. Extensions of this work will focus on extensions to larger networks and datasets with greater temporal granularity.

References

[1] Albert-László Barabási, Réka Albert, and Hawoong Jeong. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications*, 281(1):69–77, 2000.

[2] Hyekyoung Lee, Hyejin Kang, Moo K Chung, Bung-Nyun Kim, and Dong Soo Lee. Persistent brain network homology from the perspective of dendrogram. *Medical Imaging, IEEE Transactions on*, 31(12):2267–2277, 2012.

[3] Danijela Horak, Slobodan Maletić, and Milan Rajković. Persistent homology of complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(03):P03034, 2009.

[4] Yun Gao, Ioannis Kontoyiannis, and Elie Bienenstock. Estimating the entropy of binary time series: Methodology, some theory and a simulation study. *Entropy*, 10(2):71–99, 2008.

[5] Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[6] Robert Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

[7] Andrew Tausz. The phom package: Users manual. 2013.

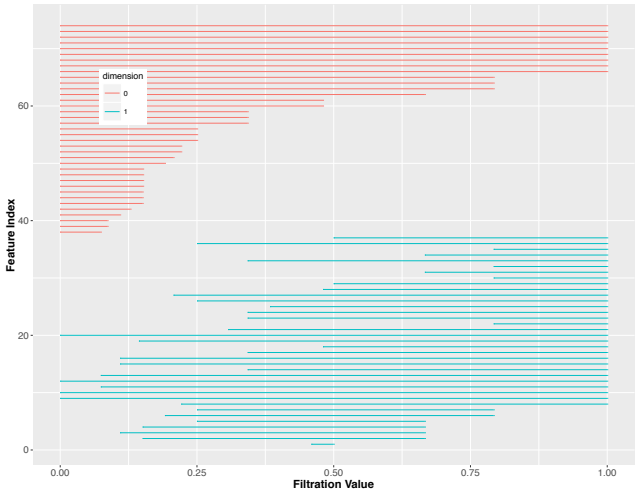


Figure 2: Barcode homology representation for example subnetwork, filtration $\epsilon \in [0, 1]$

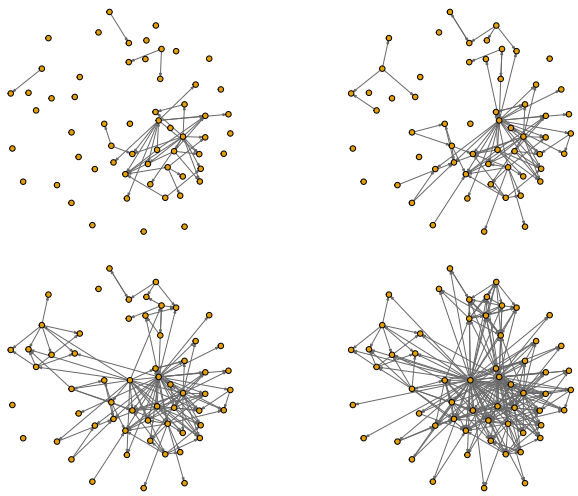


Figure 3: Example subnetwork at filtration levels 0.01, 0.25, 0.5,1

EXACT AND APPROXIMATED NULL MODELS FOR WEIGHTED DIGRAPHS

P. Singh, P. Karampourniotis, E. A. Horvat, B. Szymanski, G. Korniss, B. Uzzi

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

Null models are established tools used in network analysis for hypothesis testing [2]. Despite the prevalence of weighted directed networks, we lack established methods to construct weighted, directed degree-preserving random networks that are suited for null models. Here, we propose such a method that is inspired by the configuration model. It can be approximated by a closed formula thus, making our approach scalable to massive networks. We apply this method to obtain the null model distribution of weights in the crowdfinancing network *Kiva* [1] and use it to analyze the observed lending patterns in the network.

Generating random weighted networks

Degree-preserving network randomization is a powerful technique for assessing the statistical significance of observed structural properties of networks [3, 7]. One of the most widely used models for generating random networks with a given degree sequence is the configuration model [4]. The idea is that all edges are broken into two stubs and then reconnected uniformly at random. In directed networks, this mechanism preserves both the in- and out-degrees of nodes. The configuration model is the fastest and most convenient network sampling method when multi-edges and self loops are admitted. A straightforward extension of this method to networks with discrete weights considers an edge with weight x, as separate edges between the same two nodes (similar to [8]). The analytical formulation for exactly calculating the moments of the weights as they would be expected at random is challenging. Further, even though the permutation of the stubs under the Configuration Model scales linearly with number of (multi)edges, multiple realizations of those permutations have to be executed for the weights to asymptotically converge to the exact value. However, by preserving the degree of a node on an average as opposed to preserving the exact degree sequence, the model turns from a microcanonical to canonical ensemble [6].

The proposed model allows us to estimate weights for all node pairs of a given weighted network, as expected by this randomization process in the canonical ensemble. A comparison between the actual network and its randomized counterpart can be used for example to identify node pairs which exhibit significantly different weights than expected by chance [3, 5].

Approximating weights with a closed formula

Let k_i^{out} denote the out-degree of node i. Similarly, k_j^{in} is the in-degree of node j. Assuming that the probability of observing a link is independent of all other links, the probability of appearance of an edge from node i to j is independent of the connectivity of the rest of the edges, and it is given by

$$p_{ij} = \frac{k_i^{out} k_j^{in}}{N_E^2}, \tag{1}$$

where N_E corresponds to the total number of edges in the network. Using this probability, the expected number of links from i to j is:

$$E_{ij} = N_E p_{ij} = \frac{k_i^{out} k_j^{in}}{N_E} \tag{2}$$

with standard deviation

$$\sigma_{ij} = \sqrt{N_E p_{ij}(1 - p_{ij})} \tag{3}$$

The knowledge of the expected weight and its standard deviation allows us to compare the observed weight O_{ij} with the expected weight E_{ij} and identify non-random features in the network.

For a sufficiently large count, the binomial distribution is well approximated by the normal distribution. In that case, one can use E_{ij} and σ_{ij} to obtain the z-score defined as

$$z_{ij} = \frac{O_{ij} - E_{ij}}{\sigma_{ij}} \tag{4}$$

and its corresponding p-value for a given observation. However, for different distributions, knowledge of the probability distribution is required in order to compute the p-value. The probability of observing t links from node i to node j is obtained by the binomial distribution with

success probability p_{ij}.

$$P(X = t) = \binom{N_E}{t} p_{ij}^t (1 - p_{ij})^{N_E - t} \quad (5)$$

The sum of probability of observing more than t links is

$$P(X > t) = \sum_{X=t+1}^{\infty} \binom{N_E}{X} p_{ij}^X (1 - p_{ij})^{N_E - X} \quad (6)$$

and

$$P(X \leq t) = \sum_{X=0}^{t} \binom{N_E}{X} p_{ij}^X (1 - p_{ij})^{N_E - X} \quad (7)$$

The left- and right-tailed p-values for the observed number of links O_{ij} are

$$p_l = P(x \leq (E_{ij} - \delta)) \quad (8)$$

$$p_r = P(x > (E_{ij} + \delta)) \quad (9)$$

where $\delta = |O_{ij} - E_{ij}|$. The values of p_l and p_r are obtained numerically using Equations 6 and 7.

Experimental results on a global crowdfinancing network

To assess the quality of the approximation, we compare the adapted configuration model with the analytical approximation on the example of the *Kiva* crowdfinancing network between 220 countries in 2012 [1]. Every edge in this country-to-country network is weighted by the number of lending contributions from a certain country to another one. The out-degree distribution of the network is highly skewed with United States accounting for more than half of the loans. We compute the expected weight and its standard deviation for all possible country pairs by equations 2 and 3. Then we compare these values with the values found in the degree-preserving randomized networks. Since the expected values are independent of the ensemble, the expected weights obtained from the two methods are in very good agreement (not shown here). However, the two methods produce different standard deviations, especially when the source country is United States (as shown in Figure 1). The analytical approximation of σ is systematically higher than in the sampled networks because of the variance associated with the node degree. The discrepancy is greater for United States because of its extremely high out-degree. Finally, we apply this null model to detect non-random lending patterns in the *Kiva* network. We compare the null model distribution with the observed weights to identify the country-pairs with significantly higher (or lower) than expected weights.

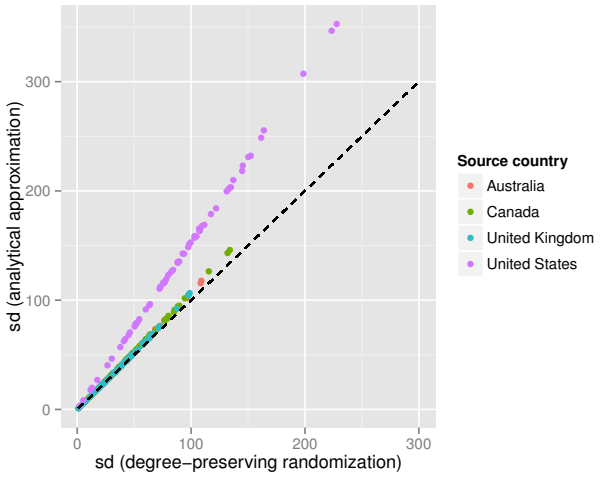


Figure 1: Comparison between the standard deviation of the number of links between two nodes as obtained by the degree-preserving randomization and by our analytical approximation, which preserves the degrees of the nodes on an average. The dashed line is the line of equality.

Conclusion

The proposed method can be used to construct a null model for directed networks with discrete weights. It allows us to obtain expected weights along with their probability distributions. The estimates for expected weights agree with those obtained by degree-preserving randomization (extended configuration model) hence making this approach viable alternative specially for large networks.

References

[1] https://www.kiva.org/.

[2] Y. Chen, P. Diaconis, S. P. Holmes, and J. S. Liu. Sequential monte carlo methods for statistical analysis of tables. *Journal of the American Statistical Association*, 100(469):109–120, 2005.

[3] N. Gotelli and G. Graves. *Null models in ecology*. Smithsonian Institution Press, 1996.

[4] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA, 2010.

[5] M. E. J. Newman and J. Park. Why social networks are different from other types of networks. *Phys. Rev. E*, 68:036122, Sep 2003.

[6] J. Park and M. E. J. Newman. Statistical mechanics of networks. *Phys. Rev. E*, 70:066117, Dec 2004.

[7] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Phys. Rev. E*, 74:016110, Jul 2006.

[8] O. Sagarra, C. J. Pérez Vicente, and A. Díaz-Guilera. Statistical mechanics of multiedge networks. *Phys. Rev. E*, 88:062806, Dec 2013.

COLLECTIVE FREQUENCY VARIATION IN NETWORK SYNCHRONIZATION AND REVERSE PAGERANK

Per Sebastian Skardal, Dane Taylor, Jie Sun, and Alex Arenas

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Introduction

Self-organization of network-coupled dynamical units is a universal phenomenon that is vital to the functionality of systems ranging from the power grid to cardiac excitation [1]. In many cases, the robust and efficient functionality of such systems depends not only on the ability to synchronize, but also the nature of the dynamics of the synchronized state. For instance, a power grid needs not only to be synchronized to avoid power failures, but functions most efficiently near a reference frequency of $\sim$ 50-60 Hz [3]. In the majority of studies of network synchronization, it is assumed that this eventual frequency, i.e., the collective frequency, of the synchronized state is equal to the average of the frequencies of the individual units, i.e., natural frequencies. In this work [2] we calculate the collective frequency of a generic directed network explicitly and show that in fact it can vary significantly from the mean of the natural frequencies.

Collective Frequency Variation

The collective frequency of a network of size N whose oscillators have natural frequencies $\{\omega_i\}_{i=1}^N$ is denoted Ω, and thus the collective frequency variation is given by $\Omega - \langle\omega\rangle$, where $\langle\cdot\rangle$ represents the mean over the network. In particular, we show that the collective frequency variation is given explicitly by

$$\Omega - \langle\omega\rangle = \frac{\langle \boldsymbol{u}^1, \boldsymbol{\omega} - \langle\omega\rangle \mathbf{1}\rangle}{\langle \boldsymbol{u}^1, \mathbf{1}\rangle}, \qquad (1)$$

where $\boldsymbol{u}^1$ is the first left singular vector (corresponding to the singular value $\sigma_1 = 0$) of the network Laplacian matrix, and $\langle\cdot,\cdot\rangle$ represents the inner-product. Thus, Eq. (1) gives the collective frequency variation as a weighted average of the natural frequency vector, where the weights are determined by the entries of the first left singular vector. Applying this formulation to general networks, we find that in generic directed networks the collective frequency variation is almost always nonzero, and in fact can be quite large. The only networks for which the collective frequency variation is zero in general is when the in- and out- degrees are balanced at each node, i.e., $k_i^{\text{in}} = k_i^{\text{out}}$ for all i. Whenever this balance is broken, a non-zero collective frequency variation should be expected. We demonstrate this effect for small, simple networks in Fig. 1, where we show the distribution of collective frequency variations for a generic directed network and a degree-balanced network.

Connection with Google's PageRank

Given the formulation of the collective frequency variation, the weights induced by the first left singular vector of the network Laplacian play a central role. Upon further inspection, we find that this weighting in fact induces a centrality measure on the network that is precisely a reverse analogue of PageRank centrality [4]. In particular, while PageRank favors nodes with a large in-flow, the left singular vector centrality favors nodes with large out-flow. Thus, the collective frequency variation highlights a surprising connection between network synchronization and Markovian random walks on networks, which define the PageRank centrality.

Collective Frequency Variation in Power Grids

Furthermore, we study collective frequency variation in an important real-world context: power grids. While power grids tend to be structurally undirected (and thus degree-balanced), the presence of heterogeneous damping coefficients in the equations of motion for the power generators and consumers result in an effective directed network structure [5]. Specifically, we consider a commonly-used power grid model on coarse-grain versions of the UK and Scandinavian power grid networks, and present the result in Fig. 2. In particular, we find that the analytical prediction given by Eq. (1) reproduces almost perfectly the observed collective frequency, and the moreover the collective frequency variation can be significant. Thus, the collective frequency variation can play a role in the dynamics of vital real-world networks.

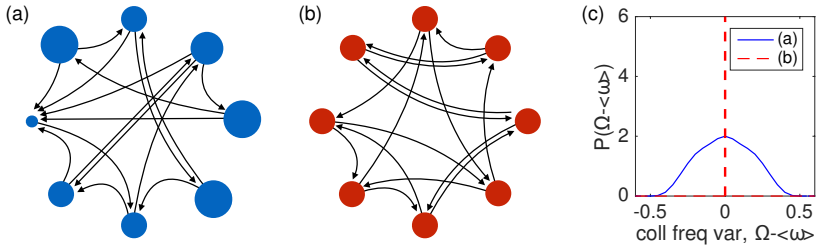


Figure 1: *Collective frequency variation.* (a),(b) Two networks of size $N = 8$ with 16 links: in (b), the in- and out-degrees match at each node ($k_i^{\text{in}} = k_i^{\text{out}} = 2$), but in (a) this balance is broken ($k_i^{\text{in}} \neq k_i^{\text{out}}$ at several nodes). Each node's area is proportional to the ratio $k_i^{\text{out}}/k_i^{\text{in}}$. (c) The density $P(\Omega - \langle\omega\rangle)$ of collective frequency variations $\Omega - \langle\omega\rangle$ observed in networks (a) and (b) (solid blue and dashed red, respectively) for different permutations of a normally distributed frequency vector $\boldsymbol{\omega}$ with zero mean and unit variance.

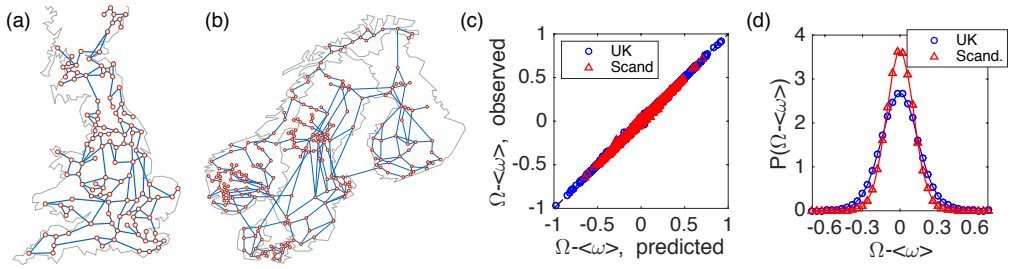


Figure 2: *Collective frequency variation in power grid networks.* (a),(b) Course-grain representations of the UK and Scandinavian power grids. (c) Collective frequency variation $\Omega - \langle\omega\rangle$ as observed in direct simulations of the power grid model in Ref. [5] vs the theoretical prediction on the UK and Scandinavian power grid networks. (d) Distribution of collective frequency variations on each network.

Outlook

Collective frequency variation in self-organizing networks can have a significant on systems that must synchronize in a particular range of reference frequencies. While we have demonstrated its effect in the case of real power-grid networks, we emphasize that this is a general result that applies to network science more broadly. The collective frequency variation itself is a result of both the heterogeneity of the directed-ness and heterogeneity of the network structure as well as the heterogeneity in the local dynamics of each individual dynamical unit in the network. Moreover, the formulation of the collective frequency variation uncovers a surprising link between synchronization dynamics and Markovian random-walk dynamics that define a network's PageRank. Given the reverse relationship between PageRank centrality and the first left singular vector centrality that defines collective frequency variation, we hypothesize that well-studied problems such as PageRank optimization can be readily applied to networks synchronization, for instance in order to mitigate the collective frequency variation in various networks.

References

[1] S. H. Strogatz. *Sync: the Emerging Science of Spontaneous Order* (Hypernion, 2003).

[2] P. S. Skardal, D. Taylor, J. Sun, and A. Arenas. Collective frequency variation in network synchronization and reverse PageRank. *Phys. Rev. E*, In press. arxiv:1510.02018.

[3] M. Rohden, A. Sorge, M. Timme, and D. Witthaut. Self-organized synchronization in decentralized power grids. *Phys. Rev. Lett.* 109: 064101 (2012).

[4] D. F. Gleich. PageRank beyond the web. *SIAM Rev.* 57: 321–363 (2015).

[5] T. Nishikawa and A. E. Motter. Comparative analysis of existing models for power-grid synchronization. *New J. Phys.* 17: 015012 (2015).

NUMERICAL METHODS FOR p-MODULUS ON NETWORKS

Nathan Albin

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

The concept of p-modulus on a network provides a general tool for quantifying the "richness" of a set of objects defined on the network. Depending on the particular choice of object, p-modulus provides various types of information about the overall network structure. Examples include the modulus of all walks connecting two particular nodes (generalizing several graph-theoretic concepts of distance), the modulus of simple cycles (providing information about clusters within the network), and the modulus of spanning trees (related to link criticality). One of the challenges in applying modulus to applications is the need for efficient algorithms for computing it. We present a simple, general algorithm for modulus, along with several specializations for modulus computations of particular importance. We also discuss open problems and interesting directions of research.

Introduction

The concept of conformal modulus was first developed as a tool in complex analysis [1,4] and metric spaces [7]. The concept also generalizes naturally to network structures and was explored in special cases (though not using the "modulus" vocabulary) as early as 60 years ago [5,6]. More recently, the discrete version of modulus has been put to use as an analytical tool with interesting applications [9, 10]. But the careful analysis of p-modulus as an object of interest in its own right has only begun in the past few years, and is one of the goals of the NODE[1] research group at Kansas State University.

In its most basic form, p-modulus is defined on a family of walks, Γ. Examples of such Γ include all walks connecting two distinct nodes, all simple cycles, all walks of a prescribed length, etc. In order to define p-modulus, we first define an admissible set, $\mathrm{Adm}(\Gamma)$, of densities (non-negative functions on the edges). Each density gives rise to a ρ-length, $\ell_\rho(\gamma)$, defined for a walk γ as the sum of $\rho(e)$ over all edges e traversed by γ. Then, $\mathrm{Adm}(\Gamma)$ is defined to be the set of all densities such that $\ell_\rho(\gamma) \geq 1$ for every $\gamma \in \Gamma$. The p-modulus is defined as

$$\mathrm{Mod}_p(\Gamma) := \min_{\rho \in \mathrm{Adm}(\Gamma)} \sum_{e \in E} \rho(e)^p,$$

for some choice of $p \in [1, \infty)$. (There is also an ∞-modulus, not defined here.)

There are a number of immediate extensions of this idea, e.g., to weighted and/or directed graphs and to multigraphs. Moreover, there is no need to restrict Γ to a family of walks; one can allow, e.g., Γ to be any family of multisets, with the ρ-length of such a set determined by the densities and multiplicities of the edges included. For example, one can define the modulus of the set of spanning trees: Γ is the set of all spanning trees, and $\ell_\rho(\gamma)$ is the total weight of the tree γ, given edge weights $\rho(e)$.

The idea of modulus has several important connections to graph-theoretic concepts (max-flow/min-cut, effective resistance, and shortest path) [2,3] and has been shown to have applications in understanding epidemic dynamics in contact networks. [8,11]

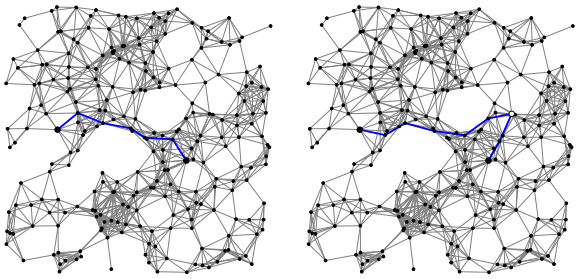
Algorithms for p-modulus

Modulus can be formulated as a convex optimization problem, so there are many existing algorithms that can, in principle, compute its value. The primary difficulty lies in the size of the constraint set, which can be large or even infinite. In most cases, it is simply not feasible to enumerate all constraints, thus rendering standard interior point methods unusable.

However, it is often possible to obtain rapid convergence using exterior point methods and a good choice of violated constraint search. For example, consider the network shown in Figure 1(a) with Γ the family of simple paths connecting two marked nodes s and t. The number of simple paths connecting the two nodes—although difficult to count—is very large. However, the algorithm can find $\mathrm{Mod}_2(\Gamma) \approx 2.67$ to three digits of accuracy in 1106 iterations and with a final active constraint set of only 637 important paths.

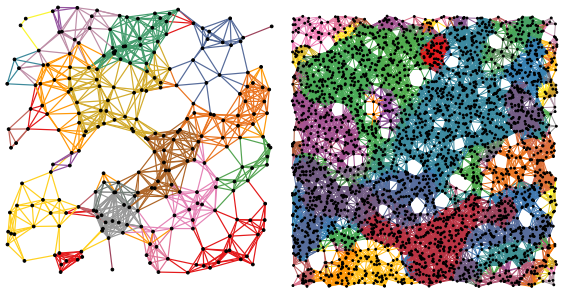
If the family Γ is changed to the family of all walks beginning at s, terminating at t, and visiting a particular

[1] https://node.math.ksu.edu/



(a) Γ = walks from s to t (b) Γ = walks from s to t via u

Figure 1: The larger solid dots represent the nodes s and t, the starting and terminating points of the walks considered. The larger open dot represents the intermediate node, u, in the case of via walks. The blue walk in each case shows the *most important walk*, identified by the constraint with largest dual variable in the convex optimization.



(a) (b)

Figure 2: The networks used in the spanning tree modulus examples. The optimal density ρ takes relatively few distinct values. Edges with identical ρ values share the same color in each figure. The resulting clusters of like edges have interesting interpretations in the context of random spanning trees.

node u along the way, a minor modification to the modulus code allows the numerical approximation of $\mathrm{Mod}_2(\Gamma)$ (see Figure 1(b)). Again, the number of constraints is difficult to count, but very large. However, the code is able to compute the value $\mathrm{Mod}_2(\Gamma) \approx 1.14$ to three digits in 1064 iterations, identifying only 567 active constraints.

As a final example of the efficiency and flexibility of this class of algorithm, we consider the family Γ of spanning trees, as described earlier. In this case, we may use Kirchoff's matrix tree theorem to count the number of constraints. The network from the previous two examples contains approximately 4.85×10^{301} spanning trees, far too many constraints to enumerate. But, the modulus code computes the value of $\mathrm{Mod}_2(\Gamma) \approx 0.0187043$ to six digits of accuracy in 815 iterations, identifying 595 active constraints (see Figure 2(a)).

Each of the three examples discussed thus far were computed on a laptop computer with a 1.9GHz Intel Core i3 processor using a Python implementation of the algorithm. All three computations were completed in 60 seconds or less. The algorithm also scales well to larger networks. For example, the spanning tree modulus on the network in Figure 2(b), with 3000 nodes and over $34K$ edges, was computed in approximately 16 minutes on a desktop workstation with a 1.6 GHz Intel Xeon processor running a more optimized Cython version of the code. Further increases in efficiency as well as parallel implementations are part of the ongoing research.

References

[1] L. V. Ahlfors. *Collected papers. Vol. 1.* Contemporary Mathematicians. Birkhäuser, Boston, Mass., 1982. 1929–1955, Edited with the assistance of Rae Michael Shortt.

[2] N. Albin, M. Brunner, R. Perez, P. Poggi-Corradini, and N. Wiens. Modulus on graphs as a generalization of standard graph theoretic quantities. *Conformal Geometry and Dynamics*, 19:298–317, 2015.

[3] N. Albin, P. Poggi-Corradini, F. Darabi Sahneh, and M. Goering. Modulus of families of walks on graphs. preprint: http://arxiv.org/abs/1401.7640, 2015.

[4] A. Beurling. *The collected works of Arne Beurling. Vol. 1.* Contemporary Mathematicians. Birkhäuser Boston, Inc., Boston, MA, 1989. Complex analysis, Edited by L. Carleson, P. Malliavin, J. Neuberger and J. Wermer.

[5] R. Duffin. The extremal length of a network. *Journal of Mathematical Analysis and Applications*, 5(2):200 – 215, 1962.

[6] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8(3):399–404, 1956.

[7] B. Fuglede. Extremal length and functional completion. *Acta Mathematica*, 98(1):171–219, 1957.

[8] M. Goering, N. Albin, F. Sahneh, C. Scoglio, and P. Poggi-Corradini. Numerical investigation of metrics for epidemic processes on graphs. 2016. Pre-accepted for the session "Epidemic Control in Networks" of the 49th Asilomar Conference on Signals, Systems and Computers, Nov. 8-11, 2015.

[9] P. Haïssinsky. Empilements de cercles et modules combinatoires. *Annales de l'Institut Fourier*, 59(no. 6):2175–2222, 2009. Version revisée et corrigée.

[10] O. Schramm. Square tilings with prescribed combinatorics. *Israel Journal of Mathematics*, 84(1-2):97–118, 1993.

[11] H. Shakeri, P. Poggi-Corradini, C. Scoglio, and N. Albin. Generalized network measures based on modulus of families of walks. *Journal of Computational and Applied Mathematics*, 2016.

IDENTIFYING STABLE NETWORKS

Vladimir Ufimtsev, Univ. of Nebraska at Omaha; Sanjukta Bhowmick, Univ. of Nebraska at Omaha; Soumya Sarkar, IIT Kharagpur, India; Animesh Mukherjee , IIT Kharagpur, India;

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Abstract

All networks created from real world datasets contain some noise, manifested in the form of missing or additional edges. A network is said to be stable if the analysis results, such as the top-k centrality metrics do not alter significantly under small amounts of noise. Here we discuss the three main factors that the affect the stability of the network and posit that examining these factors on a case by case basis will help us determine whether a network is stable.

Introduction

All real word datasets inherently contain some noise, and the models based on the data also inherit these inaccuracies. Networks, which are used to model complex systems of interacting entities, are no exception. In networks, the noise is often manifested as missing or additional edges. Network analysis has become a very important tool for analyzing properties of complex systems. For example, high centrality vertices, indicate lethal genes in a gene correlation network. However, if networks inherently contain some amount of noise, then it is important to know how much we can rely on the results.

One approach to answering this question is to perturb the network using noise models that determine how many and which edges are selected for addition or deletion. The analysis results (e.g. the top-k vertices with high degree) from the original network are compared with the results from the perturbed network. If the analysis results do not change significantly, then for that particular analysis objective, and that noise model, the network produces accurate results. Networks that produce accurate results are deemed to be stable.

There have been several studies on network stability using different noise models [1, 3, 2, 4]. However, the findings from the different experiments often contradict each other. For example, the main results in [2] is that the accuracy of the centrality measures decreases with increasing error while in [3] it is shown that degree, closeness, and eigenvector are stable while betweenness is not.

These differing claims arise because there is yet no standard noise model. Indeed, given the variations in real life data and analysis objectives creating a standard model would be limiting the scope of the problem. We therefore posit that instead of trying to develop a uniform model, a more effective method would be to identify *factors that determine whether a network is stable given a noise model.* Here we present our preliminary results on identifying these factors [1].

Factors Affecting Network Stability

Consider the following noise model proposed in [1]. Given a network $G = (V, E)$, and a parameter $\epsilon, 0 \leq \epsilon \leq |V|$ an edge not part of the original network has a probability of $\frac{\epsilon}{|V|}$ of being added. The edges to be added are selected in random. We compute the Jaccard Index (JI) of top-k centrality vertices between the original and the perturbed networks. Because the graphs are sparse, the percentage of edges being deleted is much lower and does not make a significant impact on the network structure.

Given a specific noise model, the network stability is based on the following three factors:

- **The Centrality Metric** Based on the noise model we aim to determine how the centrality metric and the relative ranking of the top-k vertices will change.
  Adding edges can increase (or keep constant) the *degree* and *closeness centrality* (CC) of a vertex. The change in *betweenness centrality* (BC) is more complicated. If two low degree nodes get connected then they create a shorter path across themselves, and therefore reduce the BC value of a high degree node. On the other hand, if a high degree node gets connected to another node, then the BC of the high degree vertex can increase.
  When the edges are added randomly, then the rank of high degree vertices are unlikely to change. However, it is difficult to determine for this model whether adding edges will alter the ranking of the high CC or BC vertices.
- **The Value of** k The stability also depends on how

[1] A longer version of this paper is in submission to a conference.

many of the top centrality vertices are considered. We have observed that there are some consecutively ranked vertices whose values are relatively close to each other, and there are other sets of consecutively ranked vertices that have a high difference in the centrality value.

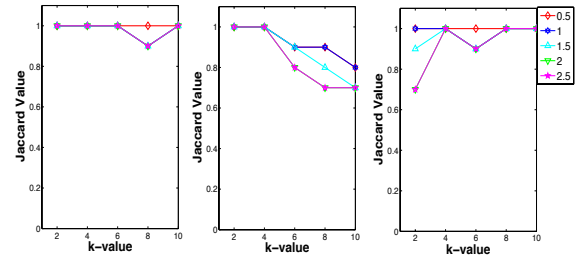
We posit that it is more difficult to change the relative ranking between two vertices if they have a huge difference in their values. We can therefore use the relative difference between consecutively ranked vertices to group similarly valued vertices into clusters. Within a cluster the ranks change under small perturbations, therefore if the value of k falls within the cluster the Jaccard Index is likely to change. If the value of k falls at the end of the cluster, then the ranking becomes more stable as it is harder for other vertices to move to the next k values due to the large relative difference.

- **Local Topology of the Network** The stability of the high ranked vertices also depend on their local connections. In particular, for CC and BC, if the high ranked vertices are tightly connected then they show more stable ranking. This is because having a high centrality vertex is more likely to increase the centrality of the vertex.

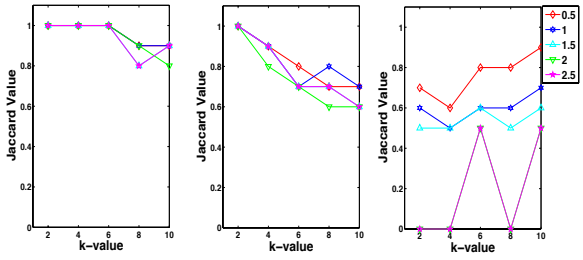
The first factor serves as a filter to determine which metrics are least likely to get affected by the noise model. For example, here it is degree centrality. The last two factors determine which specific nodes are more likely to change rank. These two factors can be tested on the original network itself without even applying noise levels.

Results

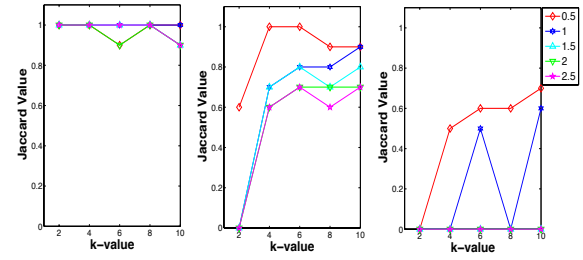
Figure shows the effect of noise with ϵ of values $.5, 1, 1.5, 2, 2.5$ on three networks, C. Elegans(V=453, E=2025), Karate (34,78) and GrQc(5242,14496). For each network, we created a set of 10 perturbed networks per *epsilon* value. The JI given is the average over these 10 networks. C. Elegant is most stable, followed by Karate and GrQc is the least stable. Table 1 shows the density of the subgraphs of the top-10 high ranked vertices for different centrality metrics. Note that the networks with more stable results have higher density subgraphs.



(a) Stability of Degree Centrality



(b) Stability of Betweenness Centrality



(c) Stability of Closeness Centrality

Figure 1: Stability of Centrality Metrics. X-axis: Values of k Y-axis: Jaccard Index for different noise levels. Left: C. Elegans; Middle: Karate; Right: GrQc

Table 1: Density of Subgraphs Induced by Top 10 High Ranked Vertices

Network	Degree	Betweenness	Closeness
C. Elegans	.8	.66	.82
Karate	.48	.44	.51
GrQc	1	.11	.26

.

References

[1] A. Adiga and A. K. Vullikanti. How robust is the core of a network? In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases - Volume 8188*, ECML PKDD 2013, pages 541–556, New York, NY, USA, 2013. Springer-Verlag New York, Inc.

[2] S. P. Borgatti, K. M. Carley, and D. Krackhardt. On the robustness of centrality measures under conditions of imperfect data. *Social networks*, 28(2):124–136, 2006.

[3] S. Segarra and A. Ribeiro. Stability and continuity of centrality measures in weighted graphs. *Signal Processing, IEEE Transactions on*, 64(3):543–555, 2016.

[4] D. J. Wang, X. Shi, D. A. McFarland, and J. Leskovec. Measurement error in network data: A re-classification. *Social Networks*, 34(4):396–409, 2012.

IDENTIFYING THE COUPLING STRUCTURE IN COMPLEX SYSTEMS THROUGH THE OPTIMAL CAUSATION ENTROPY PRINCIPLE (OCSE), WITH APPLICATIONS

Erik Bollt, Jie Sun

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Underlying the usual story of forces and energy exchange in the natural world, there is the dual story of encoding of the precise measurements one may make of the corresponding states, and the exchange of this information between the elements of a complex system that may interact. This modern perspective has been developing for quite some time in physics [1, 4], and has recently started to gain increasing recognition with theoretical break-throughs that lie at the intersection of complex systems, dynamical systems, ergodic theory, statistical mechanics, stochastic processes, and information theory. The reason for the resurgence of this perspective are several fold. A great deal of theoretical progress has been made in all of these areas. They are now coming together with new relevance thanks to the advent of modern computers which makes it possible process tremendous volumes of data. These underpinnings of information flowing between coupled processes in the natural world not only have theoretical attractiveness, but are on the verge of leading to a new and extremely applicable computational venue for inference, control, and exploring scientific, social and engineering problems from practically any area for which big data are collected. However, this has not been just a computer boon of data mining, it has encouraged a parallel resurgence and greater perspective in the underlying theoretical developments. A natural question in measurable dynamical systems is to ask which parts of a partitioned dynamical system influence other parts. Detecting dependencies between variables is a general statistical question, and in a dynamical systems context, this relates to questions of causality. There are many ways one may interpret and computationally address dependency. The concept of transfer entropy (TE) was recently developed by Schreiber [2] (see [3] for an equivalent formalism) to be a statistical measure of information flow, with respect to time, between states of a partitioned phase space in a dynamical system. Unlike other methods that simply consider common histories, transfer entropy explicitly computes the directional and asymmetric information flow from one part of the partitioned phase space to another in a dynamical system. By design, transfer entropy is suitable specifically for the detection of complex information flow between pairs of stochastic processes. However, we have recently shown that the application of transfer entropy to the problem of causal inference in a complex coupled process with many components can systematically result in many false positives [4]. This intrinsic limitation was previously not fully appreciated by the general community that continues to apply TE to this purpose. We have developed a generalized analysis called Causation Entropy (CSE) together with a construction that we call optimal Causation Entropy (oCSE) (a minimax discovery principle [5]-[8]), that we prove generally avoids the false positives issue, while correctly inferring causal links in the system in a data-efficient manner. This was shown rigorously and explicitly in [5], summarized as the optimal Causation Entropy (oCSE) principle which states that the direct causal influences of a given component in a system is the unique minimal set of components that maximizes the (unconditional) CSE to that component. This allows the development of efficient algorithms to infer cause-and-effect relationships without running into the common issue of systematically biased false positives. Validation in terms of analytical and numerical results for Gaussian processes on large random networks highlights that inference by our algorithm outperforms previous leading methods, including conditional Granger causality and transfer entropy. Interestingly, our numerical results suggest that the number of samples required for accurate inference depends strongly on network characteristics such as the density of links and information diffusion rate and not necessarily on the number of nodes. With this tool it begets the basis for several applications directions in both fundamental theoretical descriptions as well as applications in many engineering and scientific fields. Here we will discuss and present, 1) Brain functional inference based on information flow in a human brain as sensed by fMRI, for mapping of information flow, functional group-

ings, beyond the usual anatomical groupings. 2) Insects warming motions as sensed by PIV like methods for inference of communications channels between animals as they intermittently dance in different pairings and groupings. 3) Identification of Boolean networks, and as related to gene regulatory networks.

References

[1] M. Mezarda, A. Montanari, Information, Physics, and Computation (Oxford University Press, 2009).

[2] T. Schreiber, Measuring Information Transfer. Phys. Rev. Lett. 85, 461 (2000).

[3] M. Palus, V. Komarek, Z. Hrncr, and K. Sterbova, Synchronization as adjustment of information rates: Detection from bivariate time series. Phys. Rev. E 63, 046211 (2001).

[4] E. Bollt, Synchronization as a Process of Sharing and Transferring Information, Internat. J. Bifur. Chaos Appl. Sci. Engrg. 22, 1250261 (2012).

[5] J. Sun, D. Taylor, and E. M. Bollt, Causal network inference by optimal causation entropy. SIAM Journal of Applied Dynamical Systems, Vol. 14, No. 1, pp. 73106, 2015.

[6] J. Sun and E. M. Bollt, Causation Entropy Identifies Indirect Influences, Dominance of Neighbors and Anticipatory Couplings. Phys. D 267, 4957 (2014).

[7] J.Sun,C.Cafaro, and E.M.Bollt, Identifying coupling structure in complex systems through the optimal causation entropy principle. Entropy 16 34163433 (2014).

[8] Carlo Cafaro, Warren M. Lord, Jie Sun, and Erik M. Bollt, Causation Entropy from Symbolic Representations of Dynamical Systems, CHAOS 25, 043106 (2015).

THE DIAMETERS OF TRANSPORTATION POLYTOPES SATISFY THE HIRSCH CONJECTURE

*Steffen Borgwardt, Jesús A. De Loera, Elisabeth Finhold**

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We show that the Hirsch conjecture holds for two-way $M\times N$ transportation polytopes. Thus, their diameter is bounded above by $M+N-1$.

Introduction

Network flow problems are fundamental problems in optimization. The special case of min-cost flow problems on bipartite networks are called a *transportation problems*. They model the minimum-cost of transporting goods from M suppliers to N demand locations, where each of these $M+N$ locations sends, respectively receives, a specified quantity of a product (we call these quantities $u \in \mathbb{R}^M$ and $v \in \mathbb{R}^N$ the *margins*). The *transportation polytope* is the convex polytope of all feasible solutions defined by the constraints

$$TP(u,v) = \left\{ y \in \mathbb{R}^{M\times N}_{\geq 0} : \sum_{j=1}^{N} y_{ij} = u_i \ \forall i,\ \sum_{i=1}^{M} y_{ij} = v_j \ \forall j \right\},$$

where y_{ij} is the flow from supplier i to demand location j.

The *combinatorial diameter* of a polyhedron is the maximum number of edges (or 1-faces) needed to connect any two of its vertices. Alternatively, it can be defined as the diameter of the skeleton (or 1-skeleton) of the polyhedron. Motivated by the study of the worst-case performance of the simplex algorithm to solve linear optimization problems, researchers have considered the geometric problem of deciding what is the largest possible (combinatorial) diameter of convex polytopes with given number of facets and dimension. The famous *Hirsch conjecture* claimed an upper bound of $f-d$ on the diameter any d-dimensional polyhedron with f facets [3]. It is known to be true for several special classes of polyhedra, but the Hirsch bound does not hold in general [5, 6].

Even though network flow problems and transportation problems are among the simplest possible linear optimization problems, their exact diameter has until now remained open. For $M\times N$ transportation polytopes, the Hirsch conjecture claims an upper bound of $M+N-1-\mu$, where μ is the number of so-called *critical pairs* of a supply and a demand node. These are the variables that are strictly positive in every feasible solution to our transportation problem. The Hirsch bound was shown to hold for $2\times N$ and $3\times N$ transportation polytopes [1, 4]. The best published general upper bound on their diameter is $8(M+N-2)$ in [2]. We finally prove the Hirsch conjecture is true for all $M\times N$ transportation polytopes.

Theorem 1 *The diameter of an $M\times N$ transportation polytope is bounded above by $M+N-1-\mu$, where μ is the number of critical pairs of the transportation polytope. Therefore, the Hirsch conjecture is true for all $M\times N$ transportation polytopes.*

Note that for all $M\geq 3$, $N\geq 4$ there are $M\times N$ transportation polytopes that attain that bound. To prove Theorem 1, we give an algorithm that connects any two vertices of a transportation polytope by a walk on the skeleton that has length at most $M+N-1-\mu$. We remark that this is not necessarily a walk of minimum length between the two vertices. However, the walk stays in the minimal face containing both vertices. Therefore, we get the following corollary.

Corollary 1 *All faces of two-way transportation polytopes satisfy the Hirsch conjecture.*

The vertices and 1-faces of transportation polytopes

To study the diameter of transportation polytopes, we need a characterization of the vertices that are connected by a 1-face. We think of the supply and demand points as nodes in the complete bipartite graph $K_{M,N}$. For a solution y to the transportation problem, we consider the subgraph of $K_{M,N}$, that contains the edges of non-zero flow. Then y is a *vertex* of a non-degenerate transportation polytope if and only if this subgraph is a spanning tree [7]. Therefore, we refer to the vertices of a transportation polytope simply as 'trees'. (See also Figure 1.)

Further, two trees (vertices) C and C' are *connected by a 1-face* if they differ by exactly one edge [7]. In particular, a *step along the skeleton* is described by an edge that is inserted into the current tree C. The margins of the

transportation polytope then determine an edge that is deleted from C, giving the neighboring tree C'.

The algorithm

We present an algorithm that takes two trees O and F, corresponding to vertices of a transportation polytope $TP(u, v)$, as input. It constructs a sequence of trees that corresponds to a walk from O to F on the skeleton of $TP(u, v)$. The walk has length at most $M+N-1-\mu$, where μ is the number of critical pairs of $TP(u, v)$. This proves Theorem 1. The Hirsch conjecture is true for all $M \times N$ transportation polytopes.

Observe that the μ edges corresponding to the critical pairs of $TP(u, v)$ exist in every tree, and thus any two trees differ in at most $M+N-1-\mu$ edges. This suggest the following principle as the basic idea for our algorithm:

Construct a sequence of trees from a tree O to a tree F by successively inserting edges contained in F such that no previously inserted edge is ever deleted.

To keep track of the edges that may not be deleted, we *shade* them. We shade every edge we insert, but we may also shade an edge that already exists in the current tree of the sequence. When refering to both situations at the same time, we say we *"(insert and) shade an edge"*.

The most important aspect of our algorithm is the order in which edges are (inserted and) shaded. The order of insertion is determined by a *+/−labeling of the edges in the final tree F* (recall that these are precisely the edges that have to be inserted). These labels do not change during the algorithm and are preprocessed:

- Choose an arbitrary demand node δ^* and consider all paths in F starting at δ^*. Label the edges on these paths alternatingly + and −, beginning with a +.

Observe that each supply node is incident to exactly one +edge. Among the edges incident to a supply node, this +edge will be the last edge to be (inserted and) shaded; the −edges may be (inserted and) shaded in arbitrary order. Figure 1 gives an example for such a labeling.

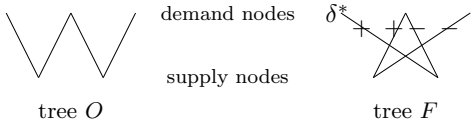


Figure 1: Tree O unshaded; tree F with edge labels.

The algorithm starts with the initial tree O with all edges unshaded. Then the edges that are contained in the final tree F are successively (inserted and) shaded. We proceed like this until we reach F with all edges shaded.

In each iteration of the algorithm, we consider the current tree C of the walk. A subset of the edges of C might already be shaded and these edges must not be deleted in this step. We choose a supply node σ that satisfies a special property and (insert and) shade an edge incident to σ. The decision of which particular edge to (insert and) shade is based on the above labeling: The unique +edge is chosen only if all −edges incident to σ are already shaded.

Figure 2 depicts the iterations of the algorithm for the input from Figure 1; dashed edges are (inserted and) shaded in the respective iteration, bold edges are already shaded and thus may not be deleted. Note that for each supply node, the +edge is the last edges to be shaded. After iteration 4 we arrive at F with all edges shaded.

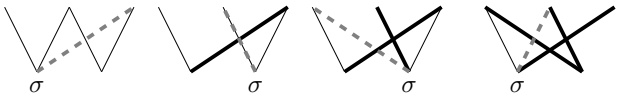


Figure 2: Iterations for O and F as in Figure 1.

The key aspect of the correctness proof of this algorithm is to show that we avoid deletion of a shaded edge. In fact, this is ensured by the careful selection of the supply node σ with the special property and the order of shading edges, based on the edge labels we fix in the beginning.

References

[1] S. Borgwardt, J. A. De Loera, E. Finhold, and J. Miller. The Hierarchy of Circuit Diameters and Transportation Polytopes. *Discrete Applied Mathematics*, http://dx.doi.org/10.1016/j.dam.2015.10.017, 2015.

[2] G. Brightwell, J. Heuvel, and L. Stougie. A Linear Bound on the Diameter of the Transportation Polytope. *Combinatorica*, 26:133–139, 2006.

[3] G. Dantzig. *Linear Programming and Extensions*. Princeton Univ. Press, 1963.

[4] J. A. De Loera and E. D. Kim. *Combinatorics and Geometry of Transportation Polytopes: An Update,* in Discrete Geometry and Algebraic Combinatorics, volume 625 of *Contemporary Mathematics*, pages 37–76. American Math. Society, 2014.

[5] V. Klee and D. W. Walkup. The d-step conjecture for polyhedra of dimension $d < 6$. *Acta Mathematica*, 117:53–78, 1967.

[6] F. Santos. A counterexample to the Hirsch conjecture. *Annals of Mathematics*, 176:383–412, 2012.

[7] V. A. Yemelichev, M. M. Kovalëv, and M. K. Kravtsov. *Polytopes, graphs and optimisation*. Cambridge University Press, Cambridge, 1984.

PARALLEL COMPUTATION OF BETWEENNESS CENTRALITY FOR LARGE PLANAR GRAPHS

Guillaume Chapuis, Hristo Djidjev
Los Alamos National Laboratory

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Introduction

The *betweenness centrality* [8] of a vertex v in a network is defined as the number of pairwise shortest paths that go through v. This measure of the importance of the vertices has been extensively used in the analysis of a variety of networks ranging from social networks to protein/protein interaction networks. As larger and larger networks become available, finding efficient methods to compute the betweenness centrality becomes increasingly crucial for network analyses.

Considering a directed graph $G(V, E)$ with a set of vertices V and a set of edges between them E with positive weights wt, we denote $n = |V|$ and $m = |E|$. The *betweenness centrality* of a vertex v is defined as $BC(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}$, where $\sigma(s,t)$ is the total number of shortest paths between s and t and $\sigma(s,t|v)$ is the number of shortest paths between s and t that go through v. The dependency δ of s and t on v is defined as $\delta(s,t|v) = \frac{\sigma(s,t|v)}{\sigma(s,t)}$ and can be seen as the contribution of the pair (s,t) to the betweenness centrality of v.

Betweenness centrality can be naively computed for all vertices in $O(n^3)$ time and $O(n^2)$ space using a modified Floyd-Warshall algorithm [6]. In [2], Brandes proposes a modified Dijkstra algorithm [3] to compute betweenness centrality in $O(n^2 \log(n))$ time and $O(n + m)$ space. The key to the effectiveness of that algorithm is the introduction of the notion of dependency of a vertex s on a vertex v defined as $\delta(s|v) = \sum_{t \in V} \delta(s,t|v)$ and a recursive formula that allows computing $\delta(s|v)$ in $O(m)$ time. If $P_s(v) = \{w \in v \mid (w,v) \in E, d(s,w) + wt(w,v) = d(s,v)\}$, then Brandes formula is $\delta(s|v) = \sum_{P_s(w) \ni v} \frac{\sigma(s,v)}{\sigma(s,w)}(1+\delta(s|w))$. Other previous work on the subject includes computing different measures of centrality, approximation algorithms, and parallel implementation (see [5] for relevant references).

In this paper we describe an algorithm for computing betweenness centrality of planar graphs that is based on graph partitioning. The algorithm is applicable to other classes of graphs that have good partitions such as some graphs with good community structure, but we focus the discussion on planar graphs so that we can evaluate the running time complexity. The advantage of our algorithm is that it allows the parallel use of many processors, which reduces the running time and allows data to be stored distributedly, thereby making it applicable to very large graphs. In [5], a sequential algorithm based in similar ideas is proposed, which uses divide and conquer for better performance in a version of the problem where betweenness centrality is computed with respect only to paths between a substantially small set of vertices $S \subset V$. Our algorithm is also related to [4], where an all-pairs shortest paths algorithm for planar graphs and its implementation on a GPU cluster is described.

Algorithm description

Our algorithm takes as an input a planar network with weights wt on its edges represented as a graph G and outputs the dependencies $\delta(s|v)$ of its vertices, which can be then summed appropriately to get the betweenness centralities of all vertices. The idea is to compute and store shortest path information associated with a small number of vertices found using graph partitioning and then use it for computing the dependencies of the vertices in a distributed manner. The algorithm consists of four phases.

In the first phase, G is divided into k parts, where k is a parameter that can be used to fine-tune the performance of the algorithm, such that no part has more than $\lceil n/k \rceil$ vertices and there are no more than $O(\sqrt{n/k})$ vertices in each part that are adjacent to vertices in other parts, called *boundary* vertices. Such a partition can be constructed in $O(n)$ time [1, 7]. We denote the subgraphs induced by these parts by $G_1, \ldots, G_k$ and call them *components* of the partition. We denote by B_i the set of the boundary vertices of G_i.

The second phase of the algorithm computes preliminary

values for d and σ for each G_i independently and in parallel. At this phase, we only need to compute distances and path counts between vertices of B_i. For this purpose, we run, for all i, Dijkstra's algorithm in G_i from each vertex of B_i. In order to avoid counting some shortest paths multiple times in subsequent steps, we stop recording shortest path once another boundary vertex is reached. We denote the computed values as d_{G_i} and σ_{G_i}.

In the third phase, we construct a graph BG called *boundary graph* that has as vertices $\cup_i B_i$ and whose edges are of the following two types: (i) edges between all pairs of vertices v and w from the same set B_i, with weight $wt_{BG}(v,w) = d_{G_i}(v,w)$ and initial value $\sigma_{BG}(v,w) = \sigma_{G_i}(v,w)$, and (ii) edges in G between vertices v and w from components $B_i \neq B_j$ with $wt_{BG}(v,w) = wt(v,w)$ and $\sigma_{BG}(v,w) = 1$. By running Dijkstra's algorithm from all vertices of BG, we compute values for d_{BG} and σ_{BG} that can be shown to be the correct distances in G.

The fourth phase of the algorithm uses the d and σ values found for BG to compute correct values for d, σ and δ for the entire graph. First, we construct $\forall i$ a graph $\bar{G}_i$ that consists of G_i and edges between those pairs of vertices from B_i for which there is a shortest path π_{ext} between them that does not cross another vertex from B_i and contains at least one vertex from another component. For each such edge, the corresponding value for d is the one computed in phase 3 and the value for σ that we compute is the number of such paths π_{ext}. Dijkstra is then run from each vertex s of G_i to recompute the values d_{G_i} and σ_{G_i}, but unlike in the first step, these values are now correct for G due to the edges from BG. We then extend the computation of d and σ for vertices of different components. For a given pair $v \in I$, $w \in J$, this is done using the following formulas: $d(v,w) = \min\{d(v,b_i) + d(b_i,b_j) + d(b_j,w) \mid (b_i,b_j) \in B_i \times B_j\}$; $\sigma(v,w) = \sum_{X_{vw}}(\sigma(v,b_i) * \sigma(b_i,b_j) * \sigma(b_j,w))$, where $X_{vw} = \{(b_i,b_j) \in B_i \times B_j \mid d(v,w) = d(v,b_i) + d(b_i,b_j) + d(b_j,w)\}$. The values used for σ at this step only take into account paths that do not cross either boundary. We now have correct values for d and σ for all pairs of vertices in G. Finally, for each source vertex $s \in G$, we compute $\delta(s|t)$ $\forall t \in G$ in three steps. In the first step, we compute potentially incorrect but required values for $\delta(s|v)$ for $v \in V_{BG}$ by running the δ-computing part of Brandes' algorithm (referred to as Br_δ) and formulas modified as in [5] on G_i, for each i. In a second step, we compute accurate $\delta(s|v)$ for $v \in V_{BG}$ by running Br_δ on the boundary graph using the potentially incorrect values from step 1 as initial values. In the third step, we compute $\delta(s|t)$ for $t \in V$ by running Br_δ in G_i $\forall i$ using values from step 2 as initial values for boundary vertices. The space and time complexity of our algorithm can easily be evaluated given the bounds $|V_i| = O(n/k)$ and $|B_i| = O(\sqrt{n/k})$ valid for planar graphs of bounded degree. We assume that we have p processors and that $p = k$ so that each processor can be assigned to exactly one component of the input graph after partitioning. Under these assumptions, our approach computes betweenness centrality in $O\left(\frac{n^2 \log\lceil n/p \rceil}{p}\right)$ time and $O(n + (n/p)^2)$ space for $p = k < n$.

Conclusion

We propose a new algorithm to compute betweenness centrality in large planar graphs, suitable for balanced distributed computations over a cluster of processors. While we haven't yet implemented the algorithm, in a previous work [4] our implementation computes all-pairs shortest paths on a graph with a million vertices and average degree 6 on a cluster of 256 GPU nodes in under 6 minutes. Given that both algorithms have similar structure, we expect the proposed new algorithm to have a similar performance. Our algorithm can also be generalized to compute other centrality measures.

References

[1] L. Aleksandrov, H. Djidjev, H. Guo, A. Maheshwari, D. Nussbaum, and J.-R. Sack. Approximate shortest path queries on weighted polyhedral surfaces. *Discrete & Computational geometry*, (44):762–801, 2010.

[2] U. Brandes. A faster algorithm for betweenness centrality*. *Journal of mathematical sociology*, 25(2):163–177, 2001.

[3] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[4] H. Djidjev, G. Chapuis, R. Andonov, S. Thulasidasan, and D. Lavenier. All-pairs shortest path algorithms for planar graph for gpu-accelerated clusters. *J. Parallel Distrib. Comput.*, 85:91–103, 2015.

[5] D. Erdős, V. Ishakian, A. Bestavros, and E. Terzi. A divide-and-conquer algorithm for betweenness centrality. In *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 2015.

[6] R. W. Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

[7] G. N. Frederickson. Planar graph decomposition and all pairs shortest paths. *Journal of ACM*, 38(1):162–204, 1991.

[8] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

PARTITIONING RANDOM GEOMETRIC GRAPHS INTO BIPARTITE BACKBONES

Zizhen Chen and David W. Matula

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We investigate the problem of verification and computation all determination of a partition of a random geometric graph (RGG) into k disjoint subgraphs satisfying the following conditions.

All but one of the subgraphs are connected $(1-\epsilon)$ dominant bipartite (planar) subgraphs of similar size and structure termed "backbones", the other of comparably small size composed of the "noise" in the random distribution.

The verification that such backbone partitions exist employing relatively few dense backbones with little loss to random distribution noise is of interest to the rapidly growing field of wireless sensor networks (WSN's)[6, 5, 1, 4].

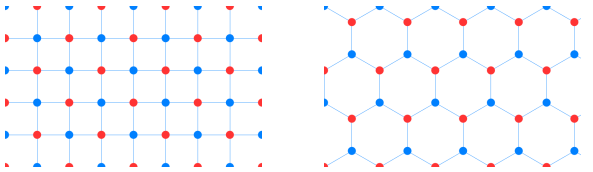
WSN's employ spatially distributed autonomous sensors to monitor physical conditions like sound, temperature, humidity and so on[2, 3, 9]. We use a random geometric graph concept in computer science to model WSNs by placing a random set of points either in a planar region or over the surface of the globe. Our goal is to determine disjoint subsets of the sensors that each can serve as a backbone for monitoring the whole region.

Our algorithm contains two-phase sequential coloring procedures (smallest-last coloring based on smallest-last ordering[7] and relay coloring based on an adaptive relay ordering) which are efficiently used to determine well-connected backbones to achieve the goal.

Extended Abstract

Given numerous randomly placed wireless sensors, how can we organize them into multiple communicating network grids (backbones) each covering the region[8]?

The bipartite planar Cartesian lattice grid with regular degree four and bipartite planar hexagonal (honeycomb) lattice grid with triangular lattice independent sets of regular degree three, see Figure 1 provides idealized placement that can be offset and replicated k times to form k backbones using all vertices. Here the face sizes are four and six.



(a) Cartesian Lattice Grid (b) Hexagonal Lattice Grid

Figure 1: Two lattice grids with face size 4 and 6

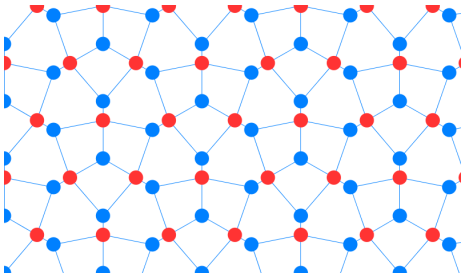


Figure 2: Bi-regular 3,4 Lattice Grid

Our question becomes if points are distributed randomly, can we select at least some minimally distributed grids with similar domination and patterns of face size primarily between four and six (like Figure 2 shows the bi-regular degree 3 and 4 lattice)?

Let a random geometric graph (RGG) denote a graph $G(N,r)$ with vertex set formed by choosing n points in a uniform random manner on the unit square, and introducing an edge between every vertex pair whose Euclidian distance is less than r. Our problem is to partition vertices into k disjoint sets $\{V_1, V_2, ..., V_k\}$ whose induced subgraphs $\langle V_1\rangle, \langle V_2\rangle, ..., \langle V_{k-1}\rangle$ are connected bipartite subgraphs with each part an independent set that dominates all or nearly all N vertices of $G(N,r)$. Let $V_1, V_2, ..., V_{k-1}$ be a partition of a majority of the vertices of $G(N,r)$ into disjoint sets where each set V_i induces a connected bipartite subgraph of $G(N,r)$. Specifically, we shall term $V_1, V_2, ..., V_{k-1}$ a bipartite component partition $BCP(\delta,\epsilon)$ of the random geometric graph $G(N,r)$ if the union of the vertex sets V_i comprise $(1-\delta)N$ of the vertices and if the induced subgraphs $\langle V_i\rangle$ on average each dominate

$(1-\epsilon)N$ of the vertices. Our goal is to determine such partitions $BCP(\delta, \epsilon)$ for δ and ϵ suitably small, practically for example, with $\delta \approx 1/k$ and $\epsilon < 0.01$.

Our primary result is a linear time algorithm that for sufficiently large N and $k \approx 15$ constructively verifies the existance of a k-part partition with $(k-1)$ subgraphs each forming connected $(1-\epsilon)$ dominant bipartite (planar) subgraphs of similar structure and size $\approx N/k$. More generally, our algorithm provides a tool to analyze and display these bipartite "backbones" both for uniform distributions on the square and on the surface of the sphere. The latter is applicable to WSN's spanning the globe.

Sample Results

Table 1 shows data of RGG's employing the Square Topology and Table 2 shows data of RGG's employing the Sphere Topology. We tested the RGG benchmarks of vertex sizes 8000, 16000, 32000, 64000 and 128000 and all of the graphs are of around average degree 60. "Surplus" denotes the portion of the vertices not partitioning in the backones and is given by $|V_k|$. "Two-core" denotes the connected bipartite subgraph without "tails" (which is the vertices of degree 1), then each vertex in the "two-core" subgraph will have degree larger or equal to 2. The "two-core" subgraph will generate a more well-connected network backbone and further details will be discussed in the following paper of this abstract.

Figure 3 shows some screenshots of benchmarks on both square and sphere topology with $G(16000, 0, 045)$ vertices which indicates the ability of our developed tool. We also believe it is a great method for research via graphical implementation to identify new patterns or features.

Table 1: Data for RGG of Square Topology

N	8000	16000	32000	64000	128000
r	0.049	0.035	0.025	0.017	0.012
Avg. degree	60.01	59.03	60.33	58.99	60.25
Backbones	14	14	14	14	14
Surplus $\lvert V_k \rvert$	871	1679	3368	6255	12422
Avg. backbone size	509.21	1022.93	2045.14	4124.64	8255.57
Avg. backbone avg. degree	2.54	2.55	2.57	2.58	2.59
Avg. backbone face size	9.66	9.64	9.43	9.27	9.19
Avg. two-core face size	7.07	7.04	7.13	7.07	7.08
Avg. backbone dominates	99.63%	97.41%	98.07%	98.68%	98.98%

Table 2: Data for RGG of Sphere Topology

N	8000	16000	32000	64000	128000
r	0.175	0.123	0.087	0.062	0.044
Avg. degree	60.05	60.11	58.98	59.93	60.21
Backbones	15	15	14	14	14
Surplus $\lvert V_k \rvert$	833	1421	2904	5975	11581
Avg. backbone size	477.80	971.93	2078.29	4144.64	8315.64
Avg. backbone avg. degree	2.53	2.53	2.61	2.61	2.62
Avg. backbone face size	10.17	10.38	9.03	8.96	8.82
Avg. two-core face size	7.05	7.13	7.04	7.03	7.00
Avg. backbone dominates	94.39%	96.68%	99.66%	99.45%	99.53%

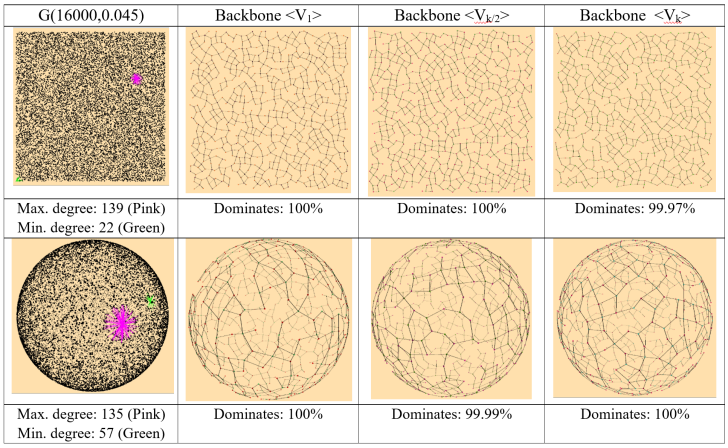


Figure 3: Screenshots of benchmarks on square model and sphere model

References

[1] Z. Chen and D. W. Matula. Partitioning rggs into disjoint (1 -) dominant bipartite subgraphs. *CSC*, pages 48–50, 2014.

[2] J. He, S. Ji, P. Fan, Y. Pan, and Y. Li. Constructing a load-balanced virtual backbone in wireless sensor networks. *Computing, Networking and Communications (ICNC)*, pages 959–963, 2012.

[3] S. Kumar, T. H. Lai, and J. Balogh. On k-coverage in a mostly sleeping sensor network. *Proceedings of the 10th annual international conference on Mobile computing and networking*, September 2004.

[4] Z. Liu, B. Wang, and Q. Tang. Approximation two independent sets based connected dominating set construction algorithm for wireless sensor networks. *Inform. Technol. J*, 9(5):864–876, 2010.

[5] D. Mahjoub and D. W. Matula. Employing $(1-\epsilon)$ dominating set partitions as backbones in wireless sensor networks. *ALENEX*, pages 98–111, 2010.

[6] D. Mahjoub and D. W. Matula. Constructing efficient rotating backbones in wireless sensor networks using graph coloring. *Computer Communications*, 35(9):1086–1097, 2012.

[7] D. W. Matula and L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *ACM*, pages 417–427, 1983.

[8] W. Y. Poe and J. B. Schmitt. Node deployment in large wireless sensor networks: coverage, energy consumption, and worst-case delay. *Asian Internet Engineering Conference*, pages 77–84, 2009.

[9] P.-J. Wan and C.-W. Yi. Coverage by randomly deployed wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 14(6):2658–2669, 2006.

ANALYZING LOCAL DENSITY IN KRONECKER MODELS

Alex J. Chin, Timothy D. Goodrich, Michael P. O'Brien, Felix Reidl, Blair D. Sullivan, Andrew van der Poel

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

Stochastic Kronecker Graphs give rise to a family of random graph models widely used in creating synthetic network data. We analyze these models through the lens of structural graph theory and prove that minor differences in the generating algorithm can significantly alter the properties of the resulting networks. Our results give further insight into the extent to which these synthetic networks capture the behavior of real-world data.

Introduction

The rapidly increasing availability of large relational data sets has consequently brought network science to the forefront of a diverse set of fields like business, social sciences, natural sciences, and engineering. Due to privacy restrictions or the desire to have testing data at larger scales, generating synthetic data from random graph models to evaluate new algorithms or techniques has become a common practice. A significant amount of research has focused on creating random graph models that produce networks whose properties mimic those of real data sets.

One popular family of random graph models are based on Stochastic Kronecker Graphs (SKG). This family includes the R-MAT model [1], and has been claimed to also include the Chung-Lu model [6]. The SKG family relies on a recursive partitioning of the adjacency matrix to determine the placement of edges. Its output can accurately replicate the degree distribution, eigenvalue distribution, diameter, and density of real data [4]. While these statistics are classically important from a network science perspective, they fail to capture structural properties that can be exploited algorithmically. Such properties—like the treewidth, degeneracy, or expansion of a network—have so far received little attention even though they are critically important in the design of efficient algorithms (particularly for solving problems that are NP-hard in general).

We expand upon previous work on the structural density of simpler network models [2, 3] and analyze the structural density of the SKG family to further map out which models can be used as predictors for algorithmic performance.

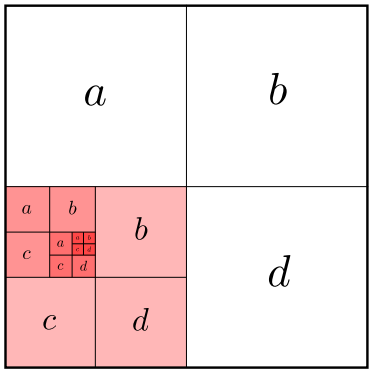


Figure 1: Recursively partitioning to reach a single entry.

Model definition

All graph generators in the SKG family base the likelihood of an edge occurring on the position of that edge in the adjacency matrix and four parameters a, b, c, d. More specifically, the adjacency matrix can be recursively partitioned into four (equal) quadrants[1] until the resulting partitions contain single elements. To reach a particular entry in this manner, one must choose a specific sequence of quadrants (i.e. upper/lower right/left) into which to recurse, as shown in Figure 1. In this way, each edge has a unique "address" and the edge probability is dependent on the number of times each quadrant, and hence which of the four factors, was chosen. The address is also encoded in the binary representation of its two endpoints.

In its original formulation [4], SKG generates each edge independently by flipping a coin with probability determined by the address. This requires n^2 operations, which can be prohibitive for generating large networks. The R-MAT generator provides a more efficient alternative[2]. In one iteration of R-MAT, a single edge is "thrown" into the matrix by recursing into a quadrant at random until it "lands" on a single entry (requiring $\log n$ steps). This procedure is repeated m times, where m is the expected

[1] The model allows partitioning into more subsections, but we restrict our attention to the simplest and most common method.

[2] R-MAT additionally imposes the condition $a + b + c + d = 1$.

number of edges. Since these models are often treated as equivalent in the literature we set out to study their asymptotic properties.

We consider three variants of R-MAT: the edge deletion, edge rethrow, and the binomial sampling model. With the edge deletion model, an element of the adjacency matrix is set to one if at least one thrown edge lands at its address; the number of edges in G may then be less than the number thrown. With edge rethrow, any throw that lands on an existing edge is rethrown until it lands on an unoccupied address. The binomial algorithm differs by running m rounds in which a coin is independently flipped for each address in the adjacency matrix. Our first result is the following equivalence between the three models:

Theorem. *All three R-MAT models are asymptotically equivalent for $m = O(n)$.*

This justifies treating them as exchangeable: all reasonable network statistics over these models will converge with increasing network size. What about SKG? As it turns out, the SKG model can be seen as a first-order approximation of the binomial R-MAT model. If p_{uv} is the probability assigned by R-MAT to the edge uv, then the binomial R-MAT model with m edges will contain this edge with probability $1 - (1 - p_{uv})^m$. We can apply the first-order approximation

$$1 - (1 - p_{uv})^m \approx mp_{uv},$$

which matches the probability that the edge will appear in SKG under the same parameters.

A first observation is that this approximation only works if the largest entry in the generator matrix is less than $1/2$. Otherwise SKG will generate a sublinear number of edges and we cannot properly convert between the models. The R-MAT models, on the other hand, provably generate graphs of unbounded degeneracy, contrary to a conjecture by Seshadhri *et al.* [7] that SKG families struggle to generate graphs of unbounded degeneracy.

Even when the largest entry is less than $1/2$, the quality of the above approximation crucially depends on how p_{uv} scales with n. For $p_{uv} \sim 1/n$ the relative approximation error converges to roughly $2/3$ whereas for $p_{uv} \sim 1/n^2$ it quickly converges to 1. Both cases will appear within the same graph for most generator matrices and while most edge-probabilities are essentially the same, a non-vanishing fraction will be different. Our analysis provides us with the means to identify statistics that will disagree because of his inherent difference. For example, we identify a function f such that the density between the $f(k)$ vertices of highest degree differ between the two models (cf. Figure 2).

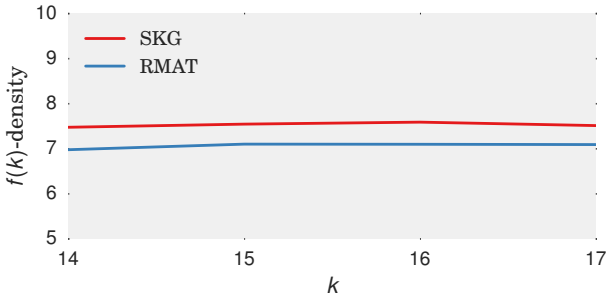


Figure 2: The density of the $f(k)$ vertices of highest degree is consistently different between SKG and R-MATH for sizes from 2^{15} to 2^{17}. Parameters used were $a = .45, b = c = .22, d = .11$ and $m = 4n$.

Significance and Application

Our results both affirm and refute previous claims about the SKG family. Moreno et al. [5] showed that the three R-MAT algorithms do not sample from the same statistical distribution as SKG; we prove that these differences extend to the structure of the graphs too. Taken together, this provides strong evidence that SKG and R-MAT algorithms cannot be used interchangeably as was initially presumed [4].

References

[1] D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-mat: A recursive model for graph mining. In *SDM*, volume 4, pages 442–446. SIAM, 2004.

[2] E. D. Demaine, F. Reidl, P. Rossmanith, F. Sánchez Villaamil, S. Sikdar, and B. D. Sullivan. Structural sparsity of complex networks: Bounded expansion in random models and real-world graphs. *CoRR*, 2014.

[3] M. Farrell, T. D. Goodrich, N. Lemons, F. Reidl, F. S. Villaamil, and B. D. Sullivan. Hyperbolicity, degeneracy, and expansion of random intersection graphs. In *WAW*, volume 9479 of *LNCS*, pages 29–41. Springer, 2015.

[4] J. Leskovec and C. Faloutsos. Scalable modeling of real graphs using Kronecker multiplication. In *Proceedings of ICML*, pages 497–504. ACM, 2007.

[5] S. Moreno, J. P. III, J. Neville, and S. Kirshner. A scalable method for accurate sampling from Kronecker models. In *Proceedings of ICDM*, 2014.

[6] A. Pinar, C. Seshadhri, and T. G. Kolda. The similarity between stochastic Kronecker and Chung–Lu graph models. *CoRR*, abs/1110.4925, 2011.

[7] C. Seshadhri, A. Pinar, and T. G. Kolda. An in-depth analysis of stochastic Kronecker graphs. *J. ACM*, 60(2):13:1–13:32, May 2013.

CONTAGION IN BANKING NETWORKS: THE ROLE OF UNCERTAINTY

Stojan Davidovic, Mirta Galesic, Konstantinos Katsikopoulos, Amit Kothiyal, Nimalan Arinaminpathy

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We study the role of information and confidence in the contagion of financial shocks in interbank markets. In this project we add realism to a model of interbank markets by introducing uncertainty into what banks know about other banks. In contrast to previous work, which assumes complete information, we show that the asymmetric distribution of information has a striking impact on the confidence dynamics and market outcomes.

Introduction

It is only recently that psychological effects, such as confidence of financial agents, have been introduced in computational models studying the stability of financial networks. However, so far it has been assumed that all agents have complete information about the system. We add realism to a model of interbank markets by introducing uncertainty into what banks know about other banks. In our model, information spreads through the lending network and the quality of information depends on the proximity of the information source. Instead of having complete information, banks receive information that is delayed, noisy, or local. We show that introducing uncertainty leads to a substantial increase in the probability of whole-system collapse after an idiosyncratic bank failure. In contrast, when a shock is distributed among multiple smaller banks, uncertainty mitigates the impact of the shock. To sum up, we demonstrated that the asymmetric distribution of information has a significant impact on the confidence dynamics and market outcomes.

Confidence and information in banking networks

While many models of financial networks treat contagion as being directly transmitted between institutions, it is also widely appreciated that psychological effects, such as market panics, also play a critical role in financial crises [2]. Previous work by Arinaminpathy, Kapadia, and May [1] (AKM) combined such "confidence effects" with network models in a simple way, presenting a framework where system distress affected how individual institutions responded to their counterparties, and vice versa. For simplicity, this work assumed that institutions have complete information (CI) about the rest of the system. In reality, however, uncertainty can play a powerful role in confidence effects. In particular, reporting is not done in real time, the reports are not always fully reliable (e.g., as in the case of Lehman Brothers), all relevant indicators are not included in the reports, and informal channels of communication facilitate further information asymmetries. Therefore, we introduced uncertainty in the AKM model to test how confidence dynamics are affected by a more realistic information flow and what are the corresponding consequences for the stability of the system.

Nodes, edges, and network

Nodes or banks in the network can be small and large, and are represented as simplified balance sheets, paremetrized by empirical data [1]. Banks are connected by borrowing and lending relationships established at the interbank market. Interbank lending of a bank corresponds to outgoing loans to other banks in the system, thus giving rise to a lending network. The network is a directed random graph with $N = 120$ banks in which the in-degree and out-degree of banks are determined by a Poisson distribution with parameter $z = 5$ for small banks and $q \cdot z = 50$ for large banks. Each edge in the network is a loan with direction from lender to borrower. A bank can withdraw its loans condition on its confidence, its health, and the health of its borrower. The confidence of a bank corresponds to its assessment of the remaining assets and interbank loans in the system. Given that information flow affects this assessment it also affects confidence dynamics and thus market outcomes. The model considers three mechanism of contagion: liquidity hoarding, counterparty-credit default, and asset price contagion.

Modeling uncertainty

We consider three uncertainty scenarios: local information (LI), delayed information (DI), and noisy information (NI). In the LI scenario, information is available only up to a certain "interbank" distance. That is, a bank determines

its confidence based on the information about itself and all banks placed within the fixed value of distance d_{max}. In the DI scenario, we model information delay as a function of distance–the further the information source the longer the delay. In the NI scenario, noise in information increases with distance.

Results

We applied various kinds of shocks to the system, but here we present a result obtained by randomly picking a large bank and forcing it to fail (Figure 1).

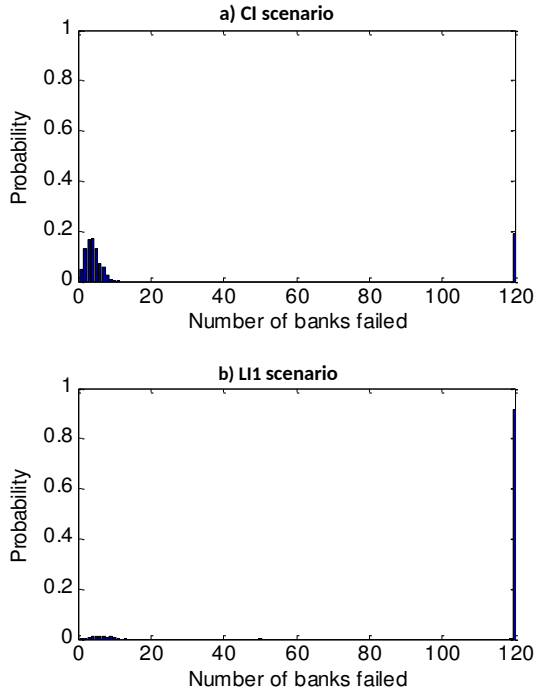


Figure 1: Probability distributions of number of failed banks after a large-bank shock in (a) the CI scenario and (b) LI1 scenario. In the CI scenario, banks form their confidence based on complete information about the system. In the LI1 scenario, banks form their confidence based on their local neighborhood within distance $d_{max} = 1$. CI = complete information; LI = local information.

The result is expressed as the probability of systemic failure–that all banks in the system fail as a result of the applied shock. We obtained the most striking result for the scenario LI1 for which a bank determines its confidence based on information only from its closest neighbors. The results show that introducing uncertainty leads to a sticking increase in the probability of systemic failure when compared to the CI scenario.

In contrast to the CI scenario, in which confidence is assessed over the extent of the whole system, in the LI1 scenario we have introduced the notion of "locally perceived" confidence that can vary with the neighborhood of different banks. The local impact of an initiating shock is therefore more intense than in a CI scenario but limited to the neighborhood, leaving the confidence of the remaining system initially intact. Yet, this local impact is subsequently transmitted through the system (analogous to the dynamics of crack propagation in a solid medium), resulting overall in a higher risk of system collapse than in the CI scenario.

We conducted a series of further analysis and demonstrated that the variability and the slope of the timecourse of confidence can account for the obtained results.

Discussion

This study demonstrates that the flow of information in a banking system is highly relevant for the dynamics of market behavior and resulting outcomes. While it is clear that both the CI and LI1 scenarios are oversimplifications of reality, our exercise shows how departing from the CI assumption can have a striking impact on the results of the model. Our main insights are that after uncertainty is introduced, the system becomes far more vulnerable to large-bank failures, as well as that the impact of the large failures becomes less predictable. The overall results clearly indicate that it is high time to recalculate the price of having large banks in the system and adjust regulation practices accordingly.

References

[1] N. Arinaminpathy, S. Kapadia, and R. M. May. Size and complexity in model financial systems. *Proceedings of the National Academy of Sciences of the United States of America*, 109(45):18338–43, 2012.

[2] C. Ó. Gráda and E. N.White. The Panics of 1854 and 1857: A View from the Emigrant Industrial Savings Bank. *Journal of Economic History*, 63(1):213–240, 2003.

RANDOM DOT PRODUCT MODELS FOR MULTIGRAPHS

Daryl R. DeFord

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

This talk presents a new generative model for multigraphs (graphs that permit multiple edges between nodes) based on the Random Dot Product Model for simple networks. This model is used to analyze a scientific collaboration network.

Background

The random dot product network (RDPN) is a particular type of latent space model, where each node in the graph process is associated to a vector in $\mathbb{R}^d$ and the probability of an edge occurring between node i and node j is given by $\langle X_i, X_j \rangle$ [6]. These models are a generalization of the traditional stochastic block model, where the block parameters/associations are controlled by the vector dot products. Because each node is associated to a vector, geometric techniques are used to study networks with RDPN models [10, 11]. The RDPN process also motivates a particular adjacency spectral embedding that has proved to be useful for proving consistency results about stochastic block model derived graphs, including a hypothesis testing procedure over the distribution of original latent positions [8, 9].

Multiplex and multilayer network models are important tools for describing all types of complex systems [2]. One of the main tasks of this research area is constructing relevant generalizations of techniques for simple networks that respect the novel properties of multiplex structures. Recently, both [5] and [7] have presented generative models that construct a multigraph directly, instead of as an aggregate of simple networks, as has been previously considered in the literature. Our model has the same property and we will show that the Poisson model presented in [5] is a one–dimensional, restricted version of our model.

Generative Model

We propose the following construction: select the vectors $\{W_i\} \subset \mathbb{R}^d$ according to a fixed distribution W. Then, place k edges between nodes i and j, where k is drawn from a Poisson distribution with parameter $\langle W_i, W_j \rangle$. We note that our formulation does not arise as a fixed, finite sum of independent, simple RDPNs since the entrywise distributions are distinct.

Restrictions of this model provide natural generalizations of other commonly studied simple network generative processes. For example, restricting the distribution to a finite set of vectors defines a multigraph block model, with community structure, while further restricting W to a single vector gives a single Poisson parameter for the entire multigraph as a generalized Erdös–Renyi model.

Two further, lower–dimensional simplifications in the choice of W reflect the interpretations of this model in the social network context described in [10]. First, we can restrict W to the unit sphere in $\mathbb{R}^d$ so that only the angle between the vectors is relevant to the dot product. Alternatively, we can realize the one–dimensional Poisson model considered in [5], where each node is associated to a positive real number, as a special case of our model by fixing an arbitrary $W_0 \neq 0$ and choosing a distribution over $\mathbb{R}_+ W_0$.

Results

One of the interesting features of the simple RDPN model is that for a broad class of distributions, W, the expected networks exhibit desirable properties, such as local clustering and small diameter, that are associated with small–world behavior in social networks. When computing the expected properties of a mutigraph model we distinguish between properties that rely only on the binary topological connectivity, like average path length or diameter, and properties that depend on the existence of multiple edges such as the degree or flow volumes. In order to extend results that rely solely on connectivity we can use $1 - e^{-\langle W_i, W_j \rangle}$ as an edge existence probability, as in the standard dot product model. This allows us to generalize the diameter bounds for RDPN given in [10] using a continuity argument. Generalizing the other expected metrics presented in that paper is currently ongoing research.

The matrix factorization algorithm given in [6] generalizes naturally to the multigraph setting, allowing us to

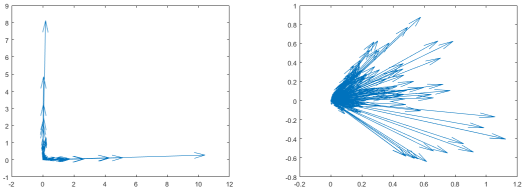
estimate the vectors $\{W_i\}$, up to an orthogonal transformation, given a particular network of interest. This allows us to use geometric techniques to study data–generated multigraphs. The interpretations, described in [6, 11] of the vector directions, representing similarity in link formation patterns between the nodes, and the magnitude of the vectors, representing propensity to communicate, are still present in this model. In particular, the angular k–means algorithm presented in [6] can be applied to vectors learned from our model.

Collaboration Networks

Scientific collaboration networks are often studied as a proxy for the professional interaction networks of researchers (see [3] as an example). In the most common formulation of these networks, the nodes are scientists and two scientists are connected by an edge if they have written a paper together. However, these interactions also have a natural multigraph structure, where the number of edges between two scientists is computed as a (weighted) sum of the papers coauthored by them [4].

For these networks, we can interpret the two attributes of the vectors, direction and magnitude, in the context of our model. Two researchers are more likely to have a higher number of coauthored papers together if they share similar interests or connectivity patters, i.e., their vectors point in similar directions, or if one or both of them is particularly prolific, represented by large magnitude. Here we consider the large connected component of a collaboration network from the field of computational geometry [1], with 7,343 authors and 11,898 publications, where the edges are weighted by the number of co–publications.

Using the iterative algorithm presented in [6] we construct a low–dimensional representation of the multigraph adjacency matrix. Comparing the embedding of the multigraph (Figure 1(a)) to the embedding of the underlying unweighted simple graph (Figure 1(b)), which is much more uniformly distributed, shows that the clustering into "nearly orthogonal" components, centered on particularly prolific scientists/subfields, is much stronger in the multigraph setting than for the simple network. Only the two dimensional embedding is shown here, but this behavior extends to higher dimensions, allowing us to readily determine the community structure of the collaboration network from the multigraph analysis. This is consistent with other collaboration network examples that we have computed.



(a) Multigraph (b) Simple Graph

Figure 1: Two–dimensional vector embeddings of the computational geometry collaboration network.

Future Work

In addition to exploring further applications to collaboration networks, we plan to extend the statistical work of [8] and [9] to our model, proving a consistent spectral embedding and constructing a hypothesis test comparison method. Also, we intend to generalize the expected network statistics for RDPN as computed in [10] to our multigraph model. Finally, we will investigate extensions of this model to arbitrarily weighted networks, by replacing the Poisson distribution in the generative process.

References

[1] V. Batagelj and A. Mrvar: *Pajek datasets*, (2006), URL: http://vlado.fmf.uni-lj.si/pub/networks/data/.

[2] M. Kivela, A. Arenas, M. Barthelemy, J. Gleeson, Y. Moreno, and M. Porter: *Multilayer Networks*, Journal of Complex Networks, 2 (3), (2014), 203–271.

[3] M. Newman: *The structure of scientific collaboration networks*, PNAS, 98 (2), (2001), 404–409.

[4] M. Newman: *Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality*, Physical Review E, 64, 1–7.

[5] J. Ranola, S. Ahn, M. Sehl, D. Smith, and K. Lange: *A Poisson Model for random multigraphs*, Bioinformatics, 26, (2010), 2004–2011.

[6] E. Scheinerman and K. Tucker: *Modeling graphs using dot product representations*, Computational Statistics, 25, (2010), 1–16.

[7] T. Shafie: *A Multigraph Approach to Social Network Analysis*, Journal of Social Structure, 16, (2015), 1–21.

[8] D. Sussman, M. Tang, D. Fishkind, and C. Priebe: *A consistent adjacency spectral embedding for stochastic blockmodel graphs*, Journal of the Americal Statistical Association, 107, (2012), 1119–1128.

[9] M. Tang, A. Athreya, D. Sussman, V. Lyzinski, and C. Priebe: *A nonparametric two–sample hypothesis testing problem for random graphs*, Arxiv: 1409.2344v2, (2014), 1–24.

[10] S. Young and E. Scheinerman: *Random Dot Product Models for Social Networks*, Algorithms and Models for the Web-Graph, Lecture Notes in Computer Science, 4863, (2007), 138–149.

[11] S. Young and E. Scheinerman: *Directed Random Dot Product Graphs*, Internet Math, 5, (2008), 91–112.

AN INVESTIGATION OF NODE-BASED METRICS ON NETWORKS ARISING FROM P-MODULUS (POSTER)

Nethali Fernando

SIAM Workshop on Network Science 2016
July 15-16 · Boston

What is a metric?

A *metric* is an ordered pair (M,d) where M is a set and d is a function $d : M \times M \to \mathbb{R}$ such that for any $x, y, z \in M$, the following hold:

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \Leftrightarrow x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$

Next we give some examples of metrics on graphs. Let $G = (V, E, \sigma)$ be a finite, simple, undirected, weighted graph, with edge-weights given by $\sigma : E \to (0, \infty)$.

Shortest-path distance

The shortest-path distance $d_s(a, b)$ between two vertices in an unweighted graph is the minimal number of edges in a path connecting a and b. Dijkstra's algorithm can be used to compute the shortest-path distance and the shortest path between any two nodes. There maybe more than one shortest path between two nodes and if there are no connecting paths then the distance is set to be infinity.

Effective resistance

The effective resistance $\operatorname{eff}\mathcal{R}(a, b)$ between two nodes a and b of a weighted graph is the electrical effective resistance of a resistor network with edge-conductances given by the edge weights. Namely, it equals the voltage potential drop needed to pass a unit current flow from a to b. It is well-known that $\operatorname{eff}\mathcal{R}(a, b)$ is a metric on V.

p-Modulus

The *p-energy* of a density $\rho : E \to [0, \infty]$ is

$$\mathcal{E}_p(\rho) := \begin{cases} \sum_{e\in E} \sigma(e)|\rho(e)|^p & \text{if } 1 \leq p < \infty \\ \max_{e\in E} |\rho(e)| & \text{if } p = \infty \end{cases}$$

The definition for $\mathcal{E}_\infty$ is consistent in the sense that

$$\forall \rho : E \to \mathbb{R} \qquad \lim_{p\to\infty} \mathcal{E}_p(\rho)^{1/p} = \mathcal{E}_\infty(\rho).$$

Consider a given family of walks Γ. We say that a density $\rho : E \to [0, \infty)$ is *admissible* for Γ (and write $\rho \in \operatorname{Adm}(\Gamma)$) if

$$\ell_\rho(\gamma) := \sum_{e\in E} \mathcal{N}(\gamma, e)\rho(e) \geq 1 \qquad \forall \gamma \in \Gamma,$$

where $\mathcal{N}(\gamma, e)$ is the number of times the walk γ crosses the edge e.

For $1 \leq p \leq \infty$, the *p-modulus* of Γ is defined as

$$\operatorname{Mod}_p(\Gamma) := \inf_{\rho\in\operatorname{Adm}(\Gamma)} \mathcal{E}_p(\rho).$$

(The modulus of an empty family $\operatorname{Mod}_p(\varnothing)$ is defined to be zero, since the choice $\rho \equiv 0$ is trivially admissible.)

A particularly useful class of walk family is the *connecting family*, denoted $\Gamma(a, b)$, of walks originating at a vertex $a \in V$ and terminating at a distinct vertex $b \in V \setminus \{a\}$.

Connection to classical quantities

Here we summarize some results from [1]. In the special case of connecting families $\Gamma(a, b)$ we recover some classical quantities. For instance, 2-modulus coincides with effective conductance, when viewing the graph as an electrical network with edge-conductances equal to σ, i.e.,

$$\operatorname{Mod}_2(\Gamma(a, b))^{-1} = \operatorname{eff}\mathcal{R}(a, b).$$

Also, 1-modulus is equal to the classical notion of Min Cut where the weight of a cut is measured by adding the edge-weights.

Moreover, letting p tend to infinity, the p-th root of p-modulus tends to the reciprocal of shortest-path, i.e.,

$$\operatorname{Mod}_\infty(\Gamma(a, b))^{-1} = d_s(a, b).$$

In general, p-modulus continuously interpolates between these classical measures. Since, as we have seen, $\operatorname{Mod}_p(\Gamma(a, b))^{-1}$ is a metric for $p = 2$ and $p = \infty$, it is natural to ask what happens for other values of p.

Main theorem

We can show that the reciprocal of Min Cut, $\operatorname{Mod}_1(\Gamma(a, b))^{-1}$ is also a metric and more generally $\operatorname{Mod}_p(\Gamma(a, b))^{-1/p}$ is a metric for all p's [2].

We actually have two different proofs for this result.

When p tends to infinity we recover shortest-path, but when $p = 2$ we get that the square-root of effective resistance is a metric. However, whenever d is a metric, then d^ϵ is always a metric for $0 < \epsilon < 1$ (this is sometimes known as "snowflaking"). Therefore, it is natural to ask what is the optimal function $\psi\left(\mathrm{Mod}_p(\Gamma(a,b))^{-1}\right)$ that yields a metric for each value of p.

Numerical experiments

We intend to explore this question numerically and will be presenting our findings in the poster.

References

[1] N. Albin, M. Brunner, R. Perez, P. Poggi-Corradini, and N. Wiens. Modulus on graphs as a generalization of standard graph theoretic quantities. *Conformal Geometry and Dynamics*, 19:298–317, 2015.

[2] N. Albin, P. Poggi-Corradini, and N. Fernando. Modulus metrics. Preprint, 2016.

ESTIMATING SUSTAINABLE PERTURBATIONS IN COMPLEX NETWORK SYNCHRONIZATION

Jeremie Fish, Jie Sun

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

For network synchronization a question of key importance is that of stability. Until recently, the focus of research was that of global stability [1,2,5-9], that is determining network structures which allowed the basin of synchronization to occupy the entire state space. For large complex networks, however, the dynamics often exhibit multistability, where multiple stable synchronous states coexist with one another (see Figure 1 below for illustration). This leads to the concept of sustainable perturbations which are best understood with the question, how large of a perturbation can the synchronous network sustain and still return to a stable synchronous state? We classify sustainable perturbations through the introduction of what we call master synchronization basins (MSB).

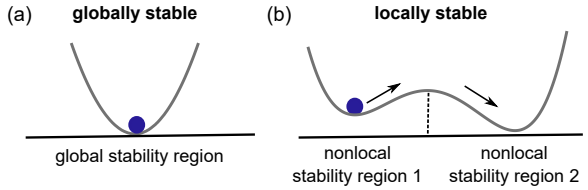


Figure 1: Comparison of (a) a globally stable state and (b) a multistable state which has locally stable regions but no globally stable state.

Synchronization of Oscillators Coupled Over a Network

The often adopted model for network synchronization can be given in the following form [6]:

$$\dot{x} = f(x_i) + \sigma \sum_{j=1}^{n} A_{ij} h(x_i, x_j), i = 1, 2, ...n. \quad (1)$$

Here $x_i(t)$ is the state of node i at time t, f represents the dynamics of each isolated oscillator, $h(x, y)$ is the coupling function, $\sigma > 0$ is the coupling strength, and A_{ij} is the adjacency matrix of the network. The network is said to synchronize if $x_1(t) = x_2(t) = x_3(t) = ... = x_n(t)$ as $t \to \infty$

Basin of Synchronization

We characterize the synchronization attractor as:

$$\mathcal{M} = \{\vec{x} = (x_1, ..., x_n) | x_1 = x_2 = ... = x_n \in \mathcal{A}\} \quad (2)$$

where $\mathcal{A}$ is an attractor of the isolated system. We thus define the basin of synchronization

$$\Omega(\mathcal{M}) = \{\vec{x}(0) | \vec{x}(t) \to \mathcal{M}, t \to \infty\} \quad (3)$$

Single and Multinode Basins

In order to investigate the effects on the network of perturbing a single node we define a single node basin to be

$$\Omega^{(i)}(s) = \{x_i(0) | x_j(0) = s \forall j \neq i, \vec{x}(0) \in \Omega(\mathcal{M})\} \quad (4)$$

where $s \in \mathcal{A}$ is an existing synchronous state. This is the set of all perturbations that can be sustained by node i before the network is knocked out of synchronization. Similarly the multinode basin is approximated as the direct product of the single node basins

$$\Omega^{(I)}(s) \approx \{(x_{i_1}, x_{i_2}, ..., x_{i_m}) | x_{i_k} \in \Omega^{(i_k)}(s)\} \quad (5)$$

where $I = \{i_1, i_2, ..., i_m\}$ is the set of multiple nodes which are being perturbed from the synchronous state

Basin Stability

The single and multinode basins in general are difficult to estimate for networks larger than a few nodes because of the high dimensionality and poor scaling which results. This complicates the estimation of the basin of synchronization for large networks since the basin is frequently non-convex. Menck et. al. [3,4] were able to work around finding the full basin of synchronization by introducing the a scalar quantity that is referred to as the basin stability. Using a subset of the state space Q basin stability is defined as:

$$\mathcal{B}(Q) = |\Omega(\mathcal{M}) \cap Q| / |Q| \quad (6)$$

with $|\cdot|$ representing the measure of a set. Basin stability is thus the fraction of initial conditions drawn from Q which are in the basin of synchronization.

Low Dimensional Approximation

In our approach we begin by considering a low-dimensional prototype system as demonstrated in Figure 2. The prototype system of two coupled oscillators is given by:

$$\begin{aligned} \dot{x} &= f(x) + \alpha h(x, y) \\ \dot{y} &= f(y) + \beta h(y, x) \end{aligned} \tag{7}$$

Integration of the low dimensional prototype equation allows us to estimate the single node or multinode basins, which we refer to as the master synchronization basins (MSB). Note that by using this system we bypass the usual need for integrating over an entire network, thus drastically reducing the effort necessary to estimate the single node basin. To match the prototype system to an arbitrary network we allow $\alpha_i = \sigma d_i$ where d_i is the degree of the perturbed node i and $\beta_j = \sigma A_{ji}$. In the typical situation of weak coupling we can see that $\beta_j \ll 1$, in those situations we typically replace β_j with 0. To validate

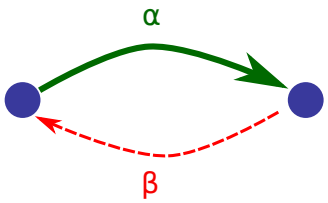


Figure 2: Low Dimensional Prototype system. The α and β terms are the coupling between the nodes, note that β is often much smaller than α and thus can typically be ignored and thus often is set to 0

the MSB approach we begin with a network of coupled one dimensional cubic equations. In this example the isolated dynamics of oscillators obey the cubic equation $f(x) = x(x-1)^2$ which has a stable fixed point at $x = 0$. The basin of attraction for th isolated oscillators is $(-1, 1)$. We chose the coupling function $h(x, y) = f(y) - f(x)$. The results of network simulations were then compared with the analytical solution for the basin of attraction we obtained with the MSB (see Figure 3). Finally we moved on to the more complicated case of chaotic oscillators. For this we chose the isolated dynamics to be Rössler oscillators and coupled the oscillators through the first component. We have obtained results for the Rössler system which also match quite well over a range of coupling strengths.

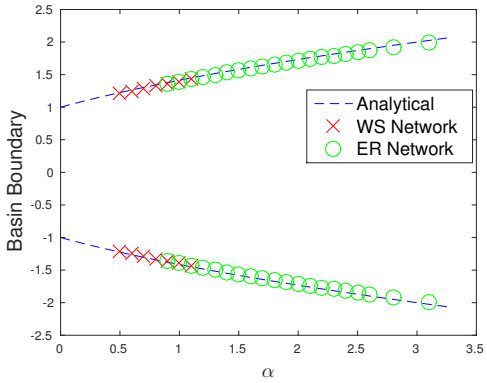


Figure 3: The solid blue line represents the analytical approximation of the basin boundary with $\beta = 0$ for the cubic oscillator system. Any initial condition starting outside the boundary will not lead to a synchronous state. The red "x" markers represent a Watts-Strogatz network and the green "o" markers represent and Erdös-Rényi network. Each marker pair represents a different degree and thus a different effective α value. As can be seen the analytical boundary and the boundary calculated from network simulations match quite well.

References

[1] A. Arenas, A. D??az-Guilera, J. Kurths, Y. Moreno, and C. Zhou. Synchronization in complex networks. *Physics Reports*, 469(3):93–153, 2008.

[2] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes. Critical phenomena in complex networks. *Reviews of Modern Physics*, 80(4):1275–1335, 2008.

[3] P. J. Menck, J. Heitzig, N. Marwan, and J. Kurths. How dead ends undermine power grid stability, 2014.

[4] P. J. Menck and J. Kurths. How basin stability complements the linear-stability paradigm. *Nature Physics*, 9:89–92, 2013.

[5] A. E. Motter, C. Zhou, and J. Kurths. Network synchronization, diffusion, and the paradox of heterogeneity. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 71(1):1–9, 2005.

[6] L. M. Pecorra and T. L. Carroll. Master Stability Functions for Synchronized Coupled Systems, 1998.

[7] B. Ravoori, A. B. Cohen, J. Sun, A. E. Motter, T. E. Murphy, and R. Roy. Robustness of Optimal Synchronization in Real Networks, 2011.

[8] J. G. Restrepo, E. Ott, and B. R. Hunt. Onset of synchronization in large networks of coupled oscillators. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 71(3):1–12, 2005.

[9] P. S. Skardal, D. Taylor, and J. Sun. Optimal synchronization of complex networks. *Physical Review Letters*, 113(14):1–5, 2014.

TEMPORAL REACHABILITY IN DYNAMIC NETWORKS

Aric Hagberg, Nathan Lemons, and Sidhant Misra

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We construct random temporal graph models using Markov chains that conserve the random graph structure at each step and have tunable dynamic properties. We analyze these models to determine the time it takes when starting from a random vertex to reach a large fraction of the other vertices by traversing temporal edges. The models we study are chosen for their simplicity and ability to be generalized for more complex models of threats in cybersecurity authentication systems.

Introduction

Dynamic network processes appear in many contexts such as spreading of infectious disease [10], synchronization of electric power generators [5], learning in the brain [2], and computer communication systems [1]. The most commonly studied case is when the network itself is not changing, or only changing slowly in time so that a static network topology is a good approximation. However, in many applications of interest the network topology is itself dynamic. When the dynamics *of* the network structure occur on roughly the same time scales as the dynamic process *on* the network, the former cannot be satisfactorily ignored. Motivated by the study of centralized computer authentication systems [6], we construct and analyze temporal network models. For those models we compute how the time to reach a large fraction of the vertices by traversing temporal edges changes with the structure and rate of change of the networks.

Models

We used publicly available data collected in the Los Alamos National Laboratory centralized authentication system [8] as a motivation in building dynamic graph models. In particular, by binning the data, we found that while approximately 25% of the edges change from day to day, many graph properties such as vertex degrees; edge counts; and small motif counts were quite stable [6]. Thus we are interested in temporal random graph models, i.e. distributions on the space $\{G_t\}_{t\geq 0}$, whose graph properties are independent of time. We used Markov chains, specifically a generalization of the edge-Markovian random graphs introduced by Clementi et al. [4].

Definition 1 (Edge Markovian)**.** A random temporal graph $\{G_t\}$ on n vertices is called *edge Markovian* if there exist maps $P, Q : E(K_n) \to [0,1]$ such that for all $t \geq 0$, each edge e of G_{t+1} is determined independently with probability

$$\mathbb{P}[e \in G_{t+1}] = \begin{cases} P(e) \text{ if } e \notin G_t\,, \\ 1 - Q(e) \text{ if } e \in G_t\,. \end{cases} \tag{1}$$

Such models create a natural Markov chain, which we call the *induced Markov chain*, defined on the space of graphs on n vertices where the transition probability of moving from G_a to G_b is given by $\mathbb{P}[G_{t+1} = G_b | G_t = G_a]$. Let P be the matrix of transition probabilities between graphs on n vertices. A distribution π on all such graphs is called *stationary* if $\pi = \pi P$. Note that if G_0 is picked according to the stationary distribution, then each subsequent G_t, $t > 0$, will also be distributed according to the stationary distribution.

For our application the most important non-trivial statistic to capture in the dyanmic model is the degree distribution so we consider a temporal version of the expected degree (Chung-Lu) model [3].

Definition 2 (Chung-Lu random temporal graph)**.** A Chung-Lu random temporal graph $G(n, W, \alpha)$ on n vertices is defined by a distribution W on the positive reals and a function $\alpha : \mathbb{R} \to (0,1)$. Let $W_1, W_2, \ldots, W_n$ be i.i.d. sampled from the distribution W. Let $p_{ij} = W_i W_j/(n\mathbb{E}[W])$ and $\alpha_i = \alpha(W_i)$. Then the temporal Chung-Lu model is defined as an edge-Markovian model with

- $P(v_i, v_j) = \sqrt{\alpha_i \alpha_j} p_{ij}$,
- $Q(v_i, v_j) = \sqrt{\alpha_i \alpha_j}(1 - p_{ij})$, and
- G_0 is distributed as $CL(n, \{W_i\})$, the static Chung-Lu model on n vertices with each edge (v_i, v_j) present independently with probability p_{ij}.

We assume W is bounded to ensure the p_{ij}'s are probabilities for large enough n. The resulting graphs have expected degrees distributed according to $\{W_i\}$.

Given the model, we now consider the problem of determining the expected time necessary to reach a linear fraction δn vertices from a random starting vertex v in a Chung-Lu temporal graph.

Definition 3 (Reachability)**.** The vertices $v_1, v_2, \ldots, v_k$ form a *temporal path* in $G = \{G_t\}$ if there exist times $t_1, t_2, \ldots, t_{k-1}$ such that for each i, $t_i \leq t_{i+1}$ and $(v_i, v_{i+1}) \in G_{t_i}$. If such a path exists, we say v_k is *reachable* from v_1 within time t_{k-1}.

Note that more than one edge can be traversed at each time step so this definition allows reaching the entire connected component at time t containing a vertex visited at time t.

Results

As a measure for the vulnerability of a network, we estimate how long it would take to traverse a large fraction of the network starting from a given vertex. We give asymptotic results for the reachability time for the Chung-Lu random temporal graph model. The reachability measure in Definition 3 allows traversal to the entire connected component of the graph at a given time so the analysis hinges on the sizes of the connected components in the graph. In the static Chung-Lu model there are two regimes for the connected component sizes which depend on the model parameters. We find these regimes in the temporal setting as well. In the "subcritical case" for each time-step the components are all small; the reachability in the graph is controlled by connecting pathways between the components over time. In the "supercritical case" the graphs at each time step have a component of size $O(n)$ and the reachability time is fast, $o(\log n)$. The proofs of these results can be found in [7].

Theorem 1 (Subcritical case)**.** *Let $G = G(n, W, \alpha)$ be a Chung-Lu random temporal graph with $\mathbb{E}[W^2] < \mathbb{E}[W]$. Let v be a vertex of G. Then there exists a constant ρ such that for each $\epsilon > 0$, asymptotically almost surely*

$$T_{1-\epsilon} \geq \rho \log n + o(\log n), \tag{2}$$

where $T_{1-\epsilon}$ is the time required to reach $(1-\epsilon)n$ vertices from v and

$$\rho = \left(1 + \frac{\mathbb{E}\left[\alpha(W)W^2\right]}{\mathbb{E}[W]} + \frac{\mathbb{E}\left[\sqrt{\alpha(W)}W^2\right]^2}{\mathbb{E}[W]^2(1 - \mathbb{E}[W^2]/\mathbb{E}[W])}\right)^{-1}.$$

We conjecture that equality holds in Equation (2).

Theorem 2 (Supercritical case)**.** *Let $G = G(n, W, \alpha)$ be a Chung-Lu random temporal graph with $\mathbb{E}[W^2] > \mathbb{E}[W]$. Suppose that the support of W and $\alpha(W)$ do not contain 0. Let v be a vertex of G. Then asymptotically almost surely*

$$T_{1-\epsilon} = o(\log n),$$

where $T_{1-\epsilon}$ is the time required to reach $(1-\epsilon)n$ vertices from v.

References

[1] N. Adams and N. Heard, editors. *Data Analysis for Network Cyber-Security*. World Scientific, 2014. ISBN 978-1-78326-374-5.

[2] D. S. Bassett, N. F. Wymbs, M. A. Porter, P. J. Mucha, J. M. Carlson, and S. T. Grafton. Dynamic reconfiguration of human brain networks during learning. *Proceedings of the National Academy of Sciences*, 108(18):7641–7646, 2011. doi: 10.1073/pnas.1018985108.

[3] F. Chung and L. Lu. Connected components in random graphs with given expected degree sequences. *Ann. Comb.*, 6(2):125–145, 2002.

[4] A. E. Clementi, C. Macci, A. Monti, F. Pasquale, and R. Silvestri. Flooding time of edge-Markovian evolving graphs. *SIAM journal on discrete mathematics*, 24(4):1694–1712, 2010.

[5] F. Dörfler, M. Chertkov, and F. Bullo. Synchronization in complex oscillator networks and smart grids. *Proceedings of the National Academy of Sciences*, 110(6):2005–2010, 2013. doi: 10.1073/pnas.1212134110.

[6] A. Hagberg, N. Lemons, A. Kent, and J. Neil. Connected components and credential hopping in authentication graphs. In *SITIS 2014*, pages 416–423. IEEE, Nov 2014. doi: 10.1109/SITIS.2014.95.

[7] A. Hagberg, N. Lemons, and S. Misra. *Dynamic Networks and Cyber-Security*, volume 1 of *Security Science and Technology*, chapter Temporal reachability in dynamic networks. World Scientific, May 2016. ISBN 978-1-78634-074-0.

[8] A. D. Kent. Anonymized user-computer authentication associations in time. 2014. doi: 10.11578/1160076.

[9] B. Neuman and T. Ts'o. Kerberos: an authentication service for computer networks. *Communications Magazine, IEEE*, 32 (9):33–38, Sept 1994. ISSN 0163-6804. doi: 10.1109/35.312841.

[10] M. E. J. Newman. Spread of epidemic disease on networks. *Phys. Rev. E*, 66:016128, Jul 2002. doi: 10.1103/PhysRevE.66.016128.

EXPONENTIAL DECAY OF CONNECTIVITY AND UNIQUENESS IN PERCOLATION ON FINITE AND INFINITE GRAPHS

Kathleen E. Hamilton and Leonid P. Pryadko

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary. We give an upper bound for the uniqueness transition on an arbitrary locally finite graph $\mathcal{G}$ in terms of the limit of the spectral radii $\rho\,[H(\mathcal{G}_t)]$ of the non-backtracking (Hashimoto) matrices for an increasing sequence of subgraphs $\mathcal{G}_t \subset \mathcal{G}_{t+1}$ which converge to $\mathcal{G}$. With the added assumption of *strong local connectivity* for the oriented line graph (OLG) of $\mathcal{G}$, connectivity on any finite subgraph $\mathcal{G}' \subset \mathcal{G}$ decays exponentially for $p < (\rho\,[H(\mathcal{G}')])^{-1}$.

Introduction. Percolation is widely used in network theory applications, yet formation of an infinite cluster is not sufficient to ensure high likelihood that an arbitrary pair of selected sites are connected, since the percolation cluster may not be unique. In this work we give upper bounds on the connectivity in site percolation on finite and infinite graphs in terms of the corresponding non-backtracking (Hashimoto) matrices, and related bounds for the uniqueness transition.

Definitions. For a graph $\mathcal{G}$ with the vertex set $\mathcal{V} \equiv \mathcal{V}(\mathcal{G})$ and edge set $\mathcal{E}$, we also consider the set of arcs (directed edges) $\mathcal{A}(\mathcal{G})$. The Hashimoto[6] matrix $H \equiv H(\mathcal{G})$ is the adjacency matrix of the oriented line graph of $\mathcal{G}$. For any pair of arcs $\{a,b\} \subset \mathcal{A}$, $H_{a,b} = 1$ iff $\{a,b\}$ form a non-backtracking walk of length two, i.e., the head of a coincides with the tail of b, but b is not the reverse of a.

In site percolation on a connected undirected graph $\mathcal{G}$, each vertex is chosen to be open with the fixed probability p, independent from other vertices. We focus on a subgraph $\mathcal{G}' \subseteq \mathcal{G}$ induced by all open vertices on $\mathcal{G}$. For each vertex v on $\mathcal{G}'$, let $\mathcal{C}(v) \subseteq \mathcal{G}'$ be the connected component of $\mathcal{G}'$ which contains the vertex v, otherwise $\mathcal{C}(v) = \emptyset$. Denote[9] by

$$\theta_v \equiv \theta_v(\mathcal{G},p) = \mathbb{P}(|\mathcal{C}(v)| = \infty), \tag{1}$$

the probability that $\mathcal{C}(v)$ is infinite. If $\mathcal{C}(v)$ is infinite, for some v, we say that percolation occurs. The percolation transition occurs at the critical probability $p_c = \sup_p\{p : \theta_v = 0\}$. Similarly, introduce the local susceptibility,

$$\chi_v \equiv \chi_v(\mathcal{G},p) = \mathbb{E}(|\mathcal{C}(v)|), \tag{2}$$

the expected cluster size connected to v, and the associated critical value $p_T = \inf\{p : \chi_v = \infty\}$. Generally, $p_c \le p_T$; on quasitransitive graphs the two thresholds coincide[8]. A third critical value, p_u, corresponds to a transition associated with the number of infinite clusters. For $p > p_u$ there can be only one infinite cluster and in general $p_u \ge p_c$. This inequality is strict on non-amenable graphs[2]. The uniqueness phase can be characterized by the connectivity,

$$\tau_{u,v} \equiv \tau_{u,v}(\mathcal{G},p) = \mathbb{P}\big(u \in \mathcal{C}(v)\big), \tag{3}$$

the probability that vertices u and v are in the same cluster. Indeed, if the percolating cluster is unique, for $p > p_u$, the connectivity is bounded from below, $\tau_{u,v} \ge \theta_u\theta_v$.

For any non-negative matrix H (finite or infinite) we define p-norm growth,

$$\mathrm{gr}_p\, H \equiv \sup_v\Big\{\lambda > 0 : \liminf_{m\to\infty} \frac{\|e_v^T H^m\|_p}{\lambda^m} = 0\Big\}, \tag{4}$$

and a similarly defined $\overline{\mathrm{gr}}_p\, H$ using limit superior. Here e_v is a vector with the only non-zero element at v equal to one. We note that for any finite graph, $\mathrm{gr}_p\, H = \overline{\mathrm{gr}}_p\, H = \rho(H)$. Moreover, if H is the Hashimoto matrix associated with a tree $\mathcal{T}$, $\|H^m e_v\|_1$ is the number of sites reachable in m non-backtracking steps from the arc $v \in \mathcal{A}(\mathcal{T})$. Then, $\mathrm{gr}_1\, H$ is exactly the growth of the tree[7], and $\overline{\mathrm{gr}}_1\, H$ is the uniformly limited growth[1]. Furthermore, on a tree, $\mathrm{gr}_2\, H = (\mathrm{gr}_1\, H)^{1/2}$ is the point spectral radius[7]. More generally, for any graph $\mathcal{G}$, $\overline{\mathrm{gr}}_2\, H$ gives an upper bound for the spectral radius $\rho_{l^2}(H)$ of H treated as an operator on $l^2(\mathcal{A})$; it satisfies the following inequalities

$$(\mathrm{gr}_1\, H)^{1/2} \le \rho_{l^2}(H) \le \overline{\mathrm{gr}}_2\, H \le \overline{\mathrm{gr}}_1\, H, \tag{5}$$

where the rightmost inequality is strict if $\mathcal{G}$ is non-amenable.

Results. We prove the following bounds:

Theorem 1. *Consider site percolation on a locally finite graph G characterized by the Hashimoto matrix H. Then $p_T \ge 1/\,\overline{\mathrm{gr}}_1\, H$, $p_c \ge 1/\,\mathrm{gr}_1\, H$.*

The first inequality is obtained by evaluating a union bound for χ_v over all non-backtracking walks starting with v [5, 4]; the second by using the bound on the percolation transition on a graph in terms of the transition on the universal cover[3]. The following connectivity bound follows directly from the alternative definition of $\rho_{l^2}(H) = \lim_{m\to\infty} \|H^m\|_2^{1/m}$:

Theorem 2. *Consider site percolation on an infinite graph $\mathcal{G}$ with maximum degree $d_{\max}$, characterized by the Hashimoto matrix H with $\rho \equiv \rho_{l^2}(H)$. Then, if $p < 1/\rho$, connectivity between any pair of sites decays exponentially with the distance, i.e., there exists a base $\rho' < 1$ and a constant $C \geq d_{\max}(1-p\rho)^{-1}$ such that*

$$\forall\{u,v\} \subset \mathcal{V}(\mathcal{G}), \quad \tau_{u,v} \leq C(\rho')^{d(u,v)}. \tag{6}$$

We say that an OLG of a connected graph $\mathcal{G}$ is strongly ℓ-connected, if for any arc $a \in \mathcal{A}(\mathcal{G})$, there is a non-backtracking walk of length at most ℓ from a to its reverse, $\bar{a}$. When such a graph is finite, the ratios of the components of the Perron-Frobenius vector of H corresponding to any pair of mutually reverted arcs are uniformly bounded (up to a constant). This gives

Theorem 3. *Consider site percolation on a finite graph $\mathcal{G}$ whose OLG is locally strongly ℓ-connected. Let H be the Hashimoto matrix of $\mathcal{G}$. Then, if $\lambda \equiv p\rho(H) < 1$, the connectivity between any pair of vertices satisfies*

$$\tau_{i,j} \leq \max(\deg i, \deg j)\frac{1+[\rho(H)]^\ell}{1-\lambda}\,\lambda^{d(i,j)}. \tag{7}$$

Moreover, for any locally-finite graph $\mathcal{G}$ whose OLG is locally strongly ℓ-connected, we have:

Theorem 4. *Consider an increasing sequence of subgraphs $\mathcal{G}_t \subset \mathcal{G}_{t+1} \subset \mathcal{G}$ convergent to a locally-finite graph $\mathcal{G}$. The following limit exists*

$$\rho_0 \equiv \lim_{t\to\infty} \rho(H_t) \leq \rho_{l^2}(H). \tag{8}$$

The upper bound is saturated, $\rho_0 = \rho_{l^2}(H)$, if the OLG of $\mathcal{G}$ is locally strongly ℓ-connected.

The same parameter ρ_0 also defines a lower bound on the uniqueness transition:

Theorem 5. *For a locally finite graph $\mathcal{G}$, the uniqueness transition satisfies $p_u \geq 1/\rho_0$.*

This follows from a bound on the expected number of self-avoiding cycles passing through a given arc, and the related analysis of cluster stability[4].

Example 1. *A degree-d infinite tree $\mathcal{T}_d$ can be obtained as a limit of an increasing sequence of its subgraphs, t-generation trees $\mathcal{G}_t = \mathcal{T}_d^{(t)}$. We have $\rho(H_t) = 0$ for any t, thus $\rho_0 = 0$, consistent with the known fact that there is no uniqueness phase for percolation on $\mathcal{T}_d$.*

Conclusions. We give lower bounds for all three transitions usually associated with site percolation on infinite graphs. We also identify a region of p where connectivity decays exponentially with the distance. For certain graphs with many short cycles, we give an improved upper bound on connectivity's exponential falloff with the distance, with explicitly specified parameters.

Acknowledgments. We are grateful to N. Delfosse for enlightening discussions. This work was supported in part by the U.S. Army Research Office under Grant No. W911NF-14-1-0272 and by the NSF under Grant No. PHY-1416578. LPP also acknowledges hospitality by the Institute for Quantum Information and Matter, an NSF Physics Frontiers Center with support of the Gordon and Betty Moore Foundation.

References

[1] O. Angel, J. Friedman, and S. Hoory. The non-backtracking spectrum of the universal cover of a graph. *Transactions of the American Mathematical Society*, 367:4287–4318, 2015.

[2] O. Häggström and J. Jonasson. Uniqueness and non-uniqueness in percolation theory. *Probability Surveys*, 3:289–344, 2006.

[3] K. E. Hamilton and L. P. Pryadko. Tight lower bound for percolation threshold on an infinite graph. *Phys. Rev. Lett.*, 113:208701, Nov 2014.

[4] K. E. Hamilton and L. P. Pryadko. Algebraic bounds for site-dependent percolation on directed and undirected graphs. *arXiv preprint arXiv:1505.03963*, 2015.

[5] K. E. Hamilton and L. P. Pryadko. Spectral bounds for percolation on directed and undirected graphs. *arXiv preprint arXiv:1503.00410*, 2015.

[6] K. Hashimoto. Zeta functions of finite graphs and representations of p-adic groups. In K. Hashimoto and Y. Namikawa, editors, *Automorphic Forms and Geometry of Arithmetic Varieties*, volume 15 of *Advanced Studies in Pure Mathematics*, pages 211–280. Kinokuniya, Tokyo, 1989.

[7] R. Lyons. Random walks and percolation on trees. *Ann. Probab.*, 18(3):931–958, 07 1990.

[8] M. V. Menshikov. Coincidence of critical points in percolation problems. *Soviet Mathematics, Doklady*, 33:856–859, 1986.

[9] R. van der Hofstad. Percolation and random graphs. In I. Molchanov and W. Kendall, editors, *New Perspectives on Stochastic Geometry*, chapter 6, pages 173–247. Oxford University Press, 2010. ISBN 978-0-19-923257-4.

SELF-CONTROL OF NETWORKS VIA ADAPTIVE LINK WEIGHT ADJUSTMENT

Mahboobeh Hejazibakhsh (1), Hiroki Sayama (1,2,3)
(1) Department of Systems Science and Industrial Engineering, Binghamton University, State University of New York, United States (e-mail: mhejazi1@binghamton.edu)
(2) Center for Collective Dynamics of Complex Systems, Binghamton University, State University of New York, United States (e-mail: sayama@binghamton.edu)
(3) Center for Complex Network Research, Northeastern University, United States

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Abstract

We discuss a model of a self-controlling adaptive network capable of effectively responding to exogenously imposed stresses. This model helps a network to control itself in a distributed way through adaptive link weight adjustment using a second-order control approach. Effectiveness and robustness of the proposed model are studied via numerical simulations. Results show that the proposed model behaves significantly better than a model with the traditional first-order control approach as it converges to the desired pre-defined final state much faster with less oscillatory behaviors.

Introduction

Complex networked systems, characterized by the presence of adaptive mechanisms, abound in nature and technology. In modeling all these real-world networks, it is often realistic to consider that the strength of the interactions between nodes is non-homogeneous and dynamically changing in response to different environmental conditions.

From a control viewpoint, it is required that a network has to reach a pre-defined desirable final state in finite time. Eq. (1) states the canonical linear control problem [1]:

$$\frac{dx}{dt} = Ax + Bu \tag{1}$$

where x represents the state vector of the system, A is the system matrix, defining how nodes interact with each other and the strength of the interactions among them, B the input matrix, and u the control input. By manipulating u, the controller intends to navigate the system state x to the desired final state x_d.

In this paper, however, we focus on a different scenario: self-control of networks. Specifically, we discuss how to design a networked system that can stay close to its own desired state by adjusting link weights (A) adaptively [2, 3] when external stress Bu is applied from the outside. Eq. (2) shows a traditional first-order adaptive control approach based on the difference between the current state of the system x and the desired state x_d [4]:

$$\frac{dA}{dt} = c(x_d - x)x^T \tag{2}$$

Here, in this paper, we propose Eq. (3) as a new second-order control model for adaptive link weight adjustment:

$$\frac{dA}{dt} = c[c'(x_d - x) - \frac{dx}{dt}]x^T \tag{3}$$

This model, similar to the traditional first-order model, describes the main driving force by measuring the difference between the system's current state and its desired final state $(x_d - x)$. However, by also measuring another level of difference between $(x_d - x)$ and $\frac{dx}{dt}$, the suggested second-order model attempts to navigate the *direction of movement* of the system's state toward a desired *direction*, not directly controlling its *position* in the state space. Our expectation is that this second-order approach may realize a smoother, faster convergence toward the desired state.

Results and Discussion

To evaluate the robustness of the proposed second-order adaptive control model, a series of numerical simulations were performed. Initial state of the system x, system matrix A, and final desired state x_d were selected randomly. The coefficients of c and c' were assumed as 1. External stress Bu was assumed to be time-invariant and randomly selected. Figures 1 and 2 show typical behaviors of the traditional first-order and suggested second-order models for networks with different number of nodes. While the first-order control model converges to the desired state in most cases, the second-order control model reaches the

desired state in less time and in a smoother way with significantly less oscillations.

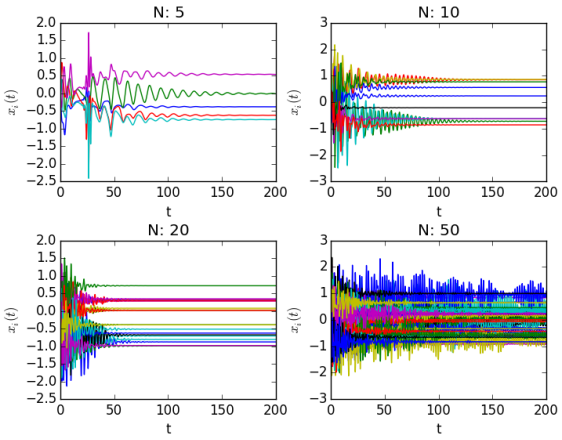


Figure 1: Typical behaviors of the first order model.

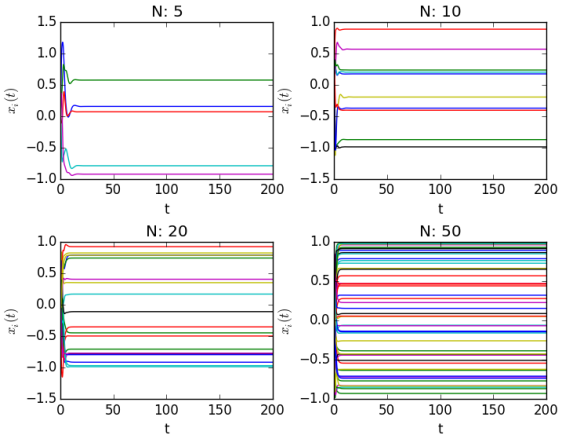


Figure 2: Typical behaviors of the second order model.

To quantitatively compare the stability of the two models, the distance of the system state from the desired state was calculated as distance function $D(t)$ (Eq. (4)) for each simulation run.

$$D(t) = \|x(t) - x_d(t)\| \tag{4}$$

The stability of the behavior was quantified by fitting the following model equation to the distance function $D(t)$:

$$D(t) \sim ae^{bt} \tag{5}$$

The parameter b indicates the estimated stability of the behavior. The stability was measured for various simulation runs while the number of nodes (N) and the connection density (p) were systematically varied. The results are shown in Figs. 3 and 4.

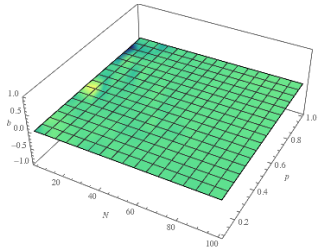


Figure 3: Stability (b) of the first-order control model.

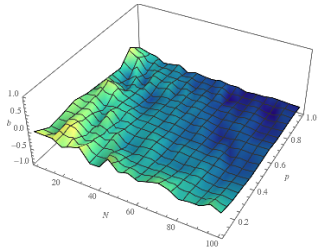


Figure 4: Stability (b) of the second-order control model.

These results support the previous observation that the second-order control model is consistently stable ($b < 0$), while the stability of the first-order model (Figure 3) is achieved only in a limited parameter regime.

Conclusion

We have proposed a model for self-controlling networked systems. The main idea is to take into consideration the second-order change in the state of network, in addition to the first-order difference between the current and final desired states. Numerical simulation results support robustness of the suggested model. Our next step will include mathematical stability analysis of the proposed model and its applications to real-world networks with non-trivial topologies.

References

[1] Kailath, T. (1980). *Linear Systems* (Vol. 156). Englewood Cliffs, NJ: Prentice-Hall.

[2] Gross, T., & Sayama, H. (2009). *Adaptive Networks: Theory, Models and Applications*. Springer Berlin Heidelberg.

[3] Sayama, H., & Sinatra, R. (2015). Social diffusion and global drift on networks. *Physical Review E*, 91(3), 032809.

[4] Narendra, K. S., & Annaswamy, A. M. (2005). Stable Adaptive Systems. Dover.

LOCAL HOMOLOGY DIMENSION AS A NETWORK SCIENCE MEASURE

Cliff Joslyn, Brenda Praggastis, Emilie Purvine, Arun Sathanur (Pacific Northwest National Laboratory)
Michael Robinson (American University)
Stephen Ranshous (North Carolina State University)

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

While topological methods are gaining prominence in other areas of data analytics [1], there is only sporadic attention to the topological roles played by specific graph structures, including vertices, edges, and cliques [3, 4]. We describe a class of measures on the clique (flag) complex of a network based on the local topological structure surrounding the multidimensional faces, and the role they play within the the overall graph structure. This representation of a graph as a topological complex admits an Alexandroff topology, and the dimensions of the local homology groups of neighborhoods of faces are computable as measures on nodes, edges, or any other higher-order cliques. Their propertries are considered and compared to other network measures both on the corresponding contracted neighborhoods in the graph, and in consideration of the planarity of the underlying graph. Examples are provided for both standard test and random graphs. We conclude with a couple of analytical results.

Local Homology on Flag Complexes of Graphs

For an undirected graph $\mathcal{G} = \langle V, E\rangle$ on a finite set of vertices V with edge set $E \subseteq V^2$, its flag complex $\mathcal{X}$ is a collection of nonempty subsets $F \subseteq V$ where $F \in \mathcal{X}$ if and only if F is a clique in $\mathcal{G}$. Each $F \in \mathcal{X}$ is a k-dimensional *simplex* or *face* in $\mathcal{X}$, where $k = |F| - 1$. $\mathcal{X}$ has the *Alexandroff topology*, whose open sets are arbitrary unions of "stars" $\star(F) = \{G \in \mathcal{X} : F \subseteq G\}$ for $F \in \mathcal{X}$. We also have the closure $\mathrm{cl}(F) = \{G \in \mathcal{X} : G \subseteq F\}$. In general for any set of faces, $Y \subseteq \mathcal{X}$, the star, $\star(Y)$, is the subset of $\mathcal{X}$ containing Y and the star of each of its elements. We similarly define the closure of any set of faces. For each set of faces $Y \subseteq \mathcal{X}$, define the 0-neighborhood of Y as $N_0(Y) = \star(Y)$ and for each $k > 0$, the k-neighborhood as $N_k(Y) = \star(\mathrm{cl}(N_{k-1}(Y))$. For every open subset, $Y \subseteq \mathcal{X}$, we introduce the measure $LH_j(Y) := \dim(H_j(\mathcal{X}, \mathcal{X}\setminus Y))$. For $j \geq 0$ $LH_j(Y)$ is the j'th local Betti number of the space (a cell complex) produced by taking the quotient of $\mathcal{X}$ by $\mathcal{X} \setminus Y$, collapsing everything outside of Y to a single point.

Fig. 1 shows an example, where on the left a graph on 7 nodes has the flag complex $\mathcal{X}$ shown with a tetrahedron, two triangles, and an edge as maximal faces. $N_0(\mathrm{cl}(\{A, K\}))$ is shown in green (the red points and edges are excluded), and the quotient space, $\mathcal{X}/(\mathcal{X} \setminus N_0(\mathrm{cl}(\{A, K\})))$ is on the right. Here we have $LH_1(N_0(\mathrm{cl}(\{A, K\}))) = 1$: focusing on $N_0(\mathrm{cl}(\{A, K\}))$ yields a single loop (shown), in that the points E and C are separated from V, T_1, and T_2. This is *not* the case, for example, with $LH_1(N_0(\{A, V\})) = 0$, since it sits on the boundary and does not divide the space.

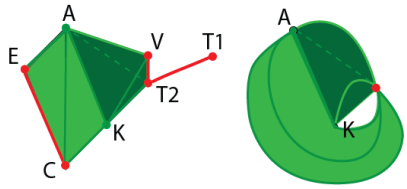


Figure 1: Example (left) flag complex of a graph and (right) the cell complex produced by focusing on the 0-neighborhood of the $\mathrm{cl}(\{A, K\})$.

Fig. 2 shows an example over a graph of USA borders: two states are connected when they have an adjoining border. As a flag complex, there is one maximal tetrahedron and one maximal edge, otherwise maximal triangles. The top and bottom show LH_1 and LH_2 of the 0-neighborhoods of all faces, as identified. We observe that LH_1 serves to identify cut faces and cuttable regions; while LH_2 serves to identify the border, including the four corners, which is measured as part of the border due to its high internal connectivity.

Observational Comparison With Network Measures

While comparing LH with vertex and edge measures used in network science is straightforward, to do so for higher dimensional faces we build a new face contracted graph,

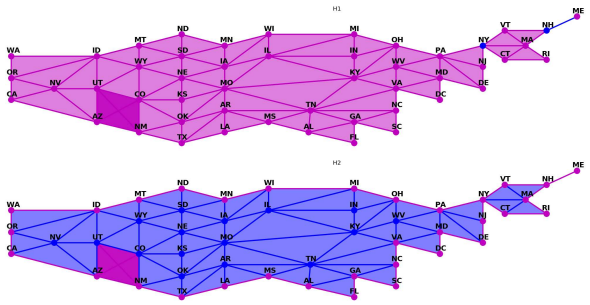


Figure 2: LH_k measures over the 0-neighborhoods of an example flag complex. (Top) $LH_1 = 0$ (magenta), $LH_1 = 1$ (blue); (Bottom) $LH_2 = 0$ (magenta), $LH_2 = 1$ (blue).

$\mathcal{G}^F$, in which vertices of $F \in \mathcal{X}$ are replaced by a single vertex, v_F, which will be adjacent to the union of the sets of neighbors of vertices in F that remain in $\mathcal{G}^F$. For any graph measure g (e.g. centrality), we let $g(F) := g(v_F)$ as computed in $\mathcal{G}^F$, noting that this will not alter the computation of any network measures on vertices.

We used NetworkX [2] and wrote a Python library [5] to calculate $LH_1(N_0(\mathrm{cl}(F))$ on all faces, F, of (1) the Zachary Karate Club social network (34 vertices, 78 edges); (2) a synthetic Erdős-Rényi (ER) graph (40 vertices, 146 edges); and (3) a synthetic Barabási-Albert (BA) preferential attachment graph (40 vertices, 144 edges). Fig. 3 shows select scatter plots comparing LH_1 with centrality measures and clustering coefficient (CC) for the Karate club. We observed strong positive correlation between LH_1 and a number of centrality measures, and strong negative correlation with the local CC.

Analytical Results

We also have some analytical results which bolster our observations above. First, we have proved [4, Thm. 12] that when $\mathcal{X}$ is an abstract simplicial complex (as all flag complexes are), and connected, then for a face with $N_0(F) \subseteq \mathcal{X}$, $LH_1(N_0(F)) + 1$ is an upper bound on the number of connected components of $\mathcal{X} \setminus N_0(F)$; and when $H_1(\mathcal{X})$ is trivial (the usual, global homology), that upper bound is attained. Thus we have $LH_1(N_0(\{v\})) = C - 1$, where C is the number of connected components in the subgraph of $\mathcal{G}$ induced by the neighbors of v (not including v itself). This is visible in Fig. 2 for the NY and NH vertices and the NH-ME edge: with $LH_1(N_0(F)) = 1$, their removal splits the local vicinity into $1 + 1 = 2$ connected components; and these reach the upper bound since the (global) $H_1 = 0$.

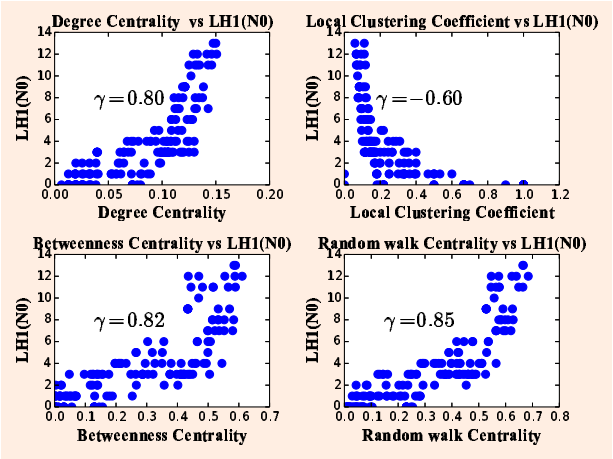


Figure 3: Scatter plots comparing network measures with LH_1 for the Karate network.

We have also found a functional relationship between CC and $LH_1(N_0(\{v\}))$. If $\mathcal{G}$ is planar, then the number of triangles incident to v is bounded by a linear function of the number of neighbors of v. This allows us to prove

$$\begin{aligned} d_v - 1 - \tfrac{d_v(d_v-1)CC(v)}{2} &\leq LH_1(N_0(\{v\})) \\ &\leq d_v - 1 - \tfrac{d_v(d_v-1)CC(v)}{6}, \end{aligned}$$

where d_v is the degree of v and $CC(v)$ is its clustering coefficient. The lower bound is true for any simple graph (i.e., not necessarily planar), but a similar upper bound cannot be shown for all simple graphs. This tells us that $LH_1(N_0(\{v\}))$ is bounded above by the pointwise maximum of a set of negatively sloped linear functions in $CC(v)$ sweeping out an "L" shaped curve. This is reflected in our experiments depicted in Figure 3 in which the data points are following the predicted upper bound. Though our graphs are not planar, the neighborhoods generally are which is sufficient for this result to hold.

References

[1] R. Ghrist. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45:1:61–75, 2007.

[2] A. Hagberg, D. Schultz, and P. Swart. Exploring network structure, dynamics, and function using NetworkX. In *SciPy2008*, pages 11–15, Pasadena, CA USA, Aug. 2008.

[3] J. Jonsson. *Simplicial Complexes of Graphs*. Springer-Verlag, Berlin, 2008.

[4] M. Robinson. Analyzing wireless communication network vulnerability with homological invariants. *GlobalSIP 2014*, pages 900–904, Atlanta, GA, 2014. IEEE.

[5] M. Robinson, B. Praggastis, and C. Capraro. Python Sheaf Library `https://github.com/kb1dds/pysheaf`, 2016

THE CO-EVOLUTION OF INNOVATION NETWORKS: THE COLLABORATION BETWEEN EAST AND WEST GERMANY FROM 1972 TO 2014

Bogang Jun, Seung-Kyu Yi, Tobias Buchmann, Mattias Mueller

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

This study aims to describe the co-evolution of German innovation networks, in which East and West German networks started to coevolve after German reunification in 1990, by analyzing German publication data that covers from 1972 to 2014. We firstly figures out that the quantitative structural change of East-West network happened in the early 2000. Second, network figures using energy model and list of top ranked regions in terms of degree centrality explain the qualitative aspect of the structural change. Third, the degree distribution and power law show that regions with a few collaborators rather than those with a large number of collaborators have dominated the properties of the German innovation networks. Last, the change in cliquishness and path length compared to four different types of benchmark models provide the network property that East and West German regions has been likely to connect new regions that locate in their community or in their surrounding, instead to jump into the new regions. Our finding supports the German government's effort for building network between East and West German regions.

Data

This paper defines a co-authorship network that reflects the regional collaboration links through research projects. In this network, the nodes are the German regions in NUTS 3 level for authors in Germany, and the name of countries for authors in outside of Germany. Two nodes are linked if scientist located in these regions write a paper together. Our dataset was collected from Web Of Science, mainly from SCI web version DB, regardless the types of article, including journal article, proceeding paper, review, letter, news item, and book review from 1972 to 2014.

The databases contain address of author; the number of co-author; the field of study; and the institution that author belongs to of all relevant journals in the all research fields categorized by Web of Science. The number of published paper we consider is 2,897,322 as a raw data, and 1,371,639 after removing single authored papers. From the papers with more than two authors, we can get the nodes and edges. The minimum number of nodes is 95 in 1972 and the maximum value is 545 in 2014. Given NUTS of Germany, which is Nomenclature of Territorial Units for Statistics, consists of 429 districts in its level 3, the average value 379 is relevant value of nodes. Regarding edges of dataset, the number of edges is 255 in 1972 as a minimum value, and it reaches 22,456 in 2014 after continuous growing over time.

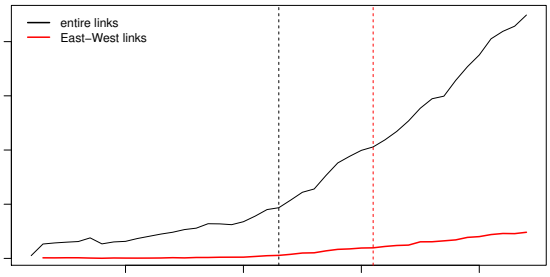


Figure 1: The number of links over time with structural break points

Results

By using Chow test in ARIMA model, as shown in Figure 1 we figured out that the quantitative structural changes of networks happen in the early 1990s for the entire network and the early 2000s for East-West network.

Second, to examine how the structural change has been proceeded, we pick 5 time points, which are 1974, 1984, 1994, 2004, and 2014, and draw networks. By analyzing the network figures, we can argue that the East-West regions were separated before the reunification and started to co-evolving over time, reaching very blended moment among German regions and other countries around 2004, and finishing to restructure in 2014 in the way that East-West ties became tighter repelling the bridge provided by other countries.

Figure 2 shows the change in the top 10 regions in terms

1974	1979	1984	1989	1994	1999	2004	2009	2014
Munich	Munich	Munich	East Berlin	Berlin	Berlin	Berlin	Berilin	Berilin
Bonn	Heidelberg	East Berlin	Munich	Munich	Munich	Munich	Munich	Munich
West Berlin	Bonn	Heidelberg	Heidelberg	Heidelberg	Heidelberg	Heidelberg	Bonn	Heidelberg
Hamburg	Hamburg	Bonn	Hamburg	Bonn	Bonn	Bonn	Heidelberg	Bonn
Heidelberg	East Berlin	Hamburg	Bonn	Hamburg	Hamburg	Hamburg	Hamburg	Hamburg
Breisgau-Hochschwarz wald	Hannover	Hannover	Breisgau-Hochschwarz wald	Breisgau-Hochschwarz wald	Alb-Donau-Kreis	Alb-Donau-Kreis	Hannover	Dresden
East Berlin	Frankfurt	Göttingen	Frankfurt	Mainz	Breisgau-Hochschwarz wald	Hannover	Dresden	Breisgau-Hochschwarzwald
Ulm	West Berlin	Frankfurt	Duisburg, Kreisfreie Stadt	Hannover	Hannover	Dresden	Alb-Donau-Kreis	Hannover
Göttingen	Mainz	Duisburg, Kreisfreie Stadt	Hannover	Frankfurt	Mainz	Borken	Leipzig	Alb-Donau-Kreis
Hannover	Breisgau-Hochschwarz wald	Breisgau-Hochschwarz wald	Göttingen	Borken	Borken	Frankfurt	Frankfurt	Leipzig

Figure 2: Top 10 nodes over time

$P(z) = az^{-\tau}$	1974	1984	1994	2004	2014
a	792.889	1189	2472	5173	8554
Std.Error	4.6989***	4.576***	5.375***	5.623***	13.95***
τ	-1.7255	-1.625	-1.598	-1.517	-1.425
Std.Error	0.01786***	0.01021***	0.005561***	0.002494***	0.003284***

Figure 3: Power Law

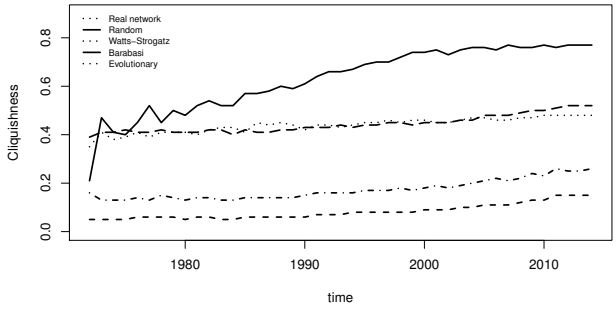


Figure 4: Cliquishness

of degree centrality. Before the reunification, Munich had been ranked top and East Berlin was the only Eastern regions within the top 10 regions. However, East Berlin ranked the top just before the reunification, which means that the innovation system of East Germany was more centralized than that of West. After the reunification, taking more than 10 years, Dresden and Leipzig joined the top 10 cities in 2004 and 2014, respectively. We can insist that Berlin region got more dominant role in the reunified Germany and it took more than 10 years for the innovation system of East regions to recover from the unprepared reunification and to be on the way of catch up that of West regions.

According to [4], the distribution that τ is equal to 2 is a dividing line, which locates between two fundamental different networks' behavior. If τ is less than 2, the few individuals with a large number of collaborator play a dominant role in deciding the average properties of network, while the networks' properties are dominated by individuals with a few collaborators when τ is greater than 2. The slope, τ of innovation network in Germany has not been greater than 2 and decreasing over time. Therefore, we can state that the properties of German innovation networks have not been dominated by a few cities with a large number of collaborators, and the trend also heads this direction.

Last, this study analyzes the change in cliquishness and path length compared to four different types of benchmark models, which are Erdos and Renyi model [2], the Watts and Strogatz model [5], the Barabási and Albert model [1] and a so called evolutionary model [3]. The change in cliquishness and path length over time provide the part of answer why German government has needed to make an effort to build a network or why still we cannot say the reunified Germany has already achieved the real integration. As the clustering coefficient increases over time compared to the benchmark networks, we can conclude that the new links and nodes haven been attached in very close cliques, which means that East and West German regions has been likely to connect new regions that locate their surrounding, instead to jump into the new regions. Considering these properties of the German innovation networks, we can conclude that innovation policy boosting networking between two German regions is effective and necessary to achieve the real unification.

References

[1] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 5439(2):509–512, 1999.

[2] P. Erdös and A. Renyi. On random graphs I. *Publicationes Mathematicae*, 6:290–297, 1959.

[3] M. Mueller, T. Buchmann, and M. Kudic. *Simulating Knowledge Dynamics in Innovation Networks*, chapter 4. Micro Strategies and Macro Patterns in the Evolution of Innovation Networks: An Agent-Based Simulation Approach, pages 73–95. Springer, 2014.

[4] M. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–9, 2001.

[5] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

AN EVOLVING NETWORK IN AN EVOLVING ENVIRONMENT: A CASE STUDY OF THE BRAZILIAN AIRLINE AND SOCIOECONOMIC NETWORKS

Austen Kelly, Saray Shai, Emanuele Strano, Peter J. Mucha

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We have reached a point where more than half of the world's population lives in cities, and thus understanding cities' growth and behavior has become crucial to their future efficiency and sustainability. While there are many studies that point to universal mathematical laws relating socioeconomic activities and city growth [1], here we explore the application of those laws to airline networks, which are also becoming ever important as our world globalizes. Our goal is to understand the interplay between the two evolving processes of a time-varying airline network and population growth.

Introduction

The airline network is vast, servicing many thousands of flights around the world each day, thus playing a critical role in our society, from mobility patterns to epidemic spreading. However, it is also a highly dynamic system which is affected by frequent changes. Each time a route is cancelled, an airport closes, or a new one is opened the structure of the network must once more adapt. This dynamic nature of the flight network makes it complicated to understand and to predict how it will behave locally or even as a system over time. In this work, we use network analysis as a tool to examine these systems in a case study of the Brazilian airline and population networks. Doing so will allow us to see both how the overlying structures of the networks and each individual city behave over time.

Datasets

Our analysis is based upon three data sets: population, gross domestic product (GDP), and airline flights. Each data set provides information on approximately 350 Brazilian cities over time. We use population data from 1970, 1980, 1991, 2000, and 2010, while GDP is given per city yearly from 2006 to 2010. The airline data spans 1995 to 2006, providing information in each year of which routes existed between airports, along with the number of passengers, flights, and cargo on each route [3]. Further, we use data of geographical coordinates of cities to embed the network in physical space, allowing for calculations of flight distances. From this data, we created yearly undirected networks in which cities (airports) are nodes, and those nodes are connected by an edge if a flight between two cities exists in that year. Edges can then be weighted, such as by total number of passengers along that edge. Finally, since the population grows with time, we linearly interpolated the logarithms of the data to estimate the populations for each year from 1995 to 2006.

Analysis

The first step towards understanding the relationship between population and airport growth lies in understanding each system individually. Much work has been done examining the properties of population dynamics, along with its relationship to economic growth, pointing towards a "superlinear scaling" phenomenon where various metrics including crime, total housing and GDP, increase consistently at a nonlinear rate with population [1]. Our data of Brazilian cities agrees with this finding, with a corresponding rate of 1.11 ± 0.02 in 2010. This agreement is an indicator of the robustness of our population and GDP data sets, and also is a measure of the relative importance of each city. This only provides snapshot views of the system, though. Figure 1(a) shows the cumulative distribution function (CDF) of population in each given year on a log-log scale, serving as a measure of global distribution and growth trends. It tells us that the population growth is heavy-tail distributed and governed by a power-law. Though the tail-end slope is constant, the distribution moves to the right over time, indicating that the population is growing steadily over time. However, something entirely different is happening on a city-level scale. Inspired by Batty's rank clocks of cities [2], Figure 1(b) shows the rank of each city over time, sorted on the $x - axis$ by rank in 1995. The great variance from the $x = y$ axis shows that although the population is growing

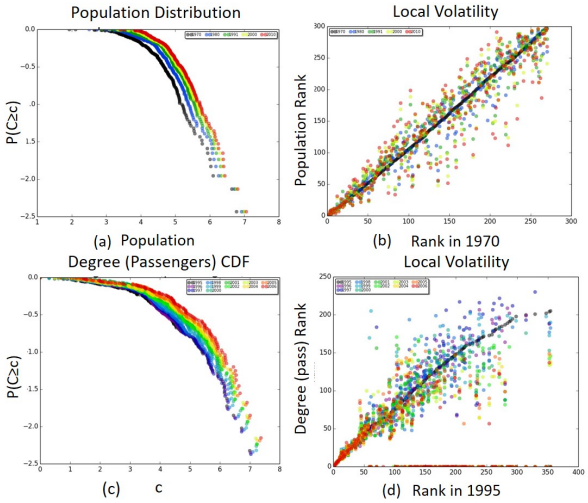


Figure 1: **Macro-stability vs. Micro volatility** (a) Cumulative distribution function of populations of cities in Brazil over time. (b) Ranking (largest = 1) of population over time, sorted on the x-axis by ranking in 1970. (c) CDF of degree weighted by number of passengers flying through the airport. (d) Ranking over time of degree weighted by passengers. Rank = 0 indicates the airport did not service any flights that year.

steadily on the global scale, the population ranking of each city is changing significantly, particularly in smaller (lower ranked) cities. This conflicting behavior between micro and macro dynamics is an indication that population behavior is more complex and dynamic than it would initially seem from the CDF alone.

Remarkably, the airline network behaves in a very similar manner to the population in both global and local analyses of temporal dynamics. Figure 1(d) displays that each airport changes in rank over time, so the airline network fluctuates on the local scale just like the population. It is noticeable though that the largest degree airports are significantly more stable in ranking than smaller ones, indicating that this network favors the hubs. Correspondingly, in figure 1(c) we note that the CDF of the degree (weighted by passengers) of airports is also heavy-tail distributed and growing with time. Unlike the population, though, the slope is decreasing, so the airports are becoming more heterogeneous. In fact, even over the course of these twelve years it is evident that there are fewer airports and routes over time, in contrast to the simultaneous increase in number of passengers and weight of cargo [3]. As such, the latter part of our analysis aims to classify the cities by their behavior in a way that can predict an airport's success based upon factors such as the city's corresponding population and geographic location relative to other cities in the network.

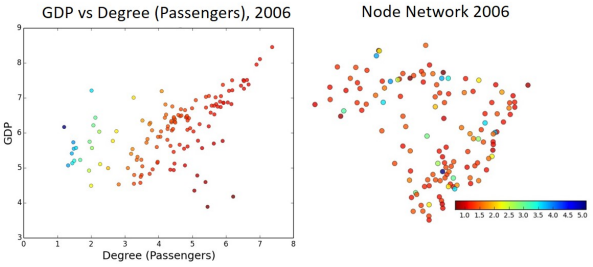


Figure 2: **GDP vs Population** (a) GDP vs degree weighted by passengers in 2006, colored by ratio of $\frac{log(GDP)}{log(pop.)}$. (b) The geographical layout of the corresponding airline node network, colored by the same ratio.

Discussion

A preliminary analysis of the relationship between these two dynamic systems indicates that despite all of the similarities between their micro and macro scale dynamics, they do not correlate with one another as consistently as population and GDP do. In figure 2 (a) we see that the high end of population and degree are described by a strong linear correlation, yet the lower portion of airports display no correlation. The observation that large and small cities appear to be governed by completely different underlying mechanics motivates a deeper analysis of what characterizes the division between these groups and of what other factors (such as geographical location, as in figure 2 (b)) might serve as a better predictor of behavior for those small airports.

References

[1] L. M. A. Battencourt. Growth, innovation, scaling, and the pace of life in cities. *Proceedings of the National Academy of Sciences*, 104(17):7301–7306, 2007.

[2] M. Batty. Rank clocks. *Nature*, 444(2):592–596, 2006.

[3] L. E. C. da Rocha. Structural evolution of the brazilian airport network. *Journal of Statistical Mechanics*, (3):15–30, 2009.

METHODS FOR GRAPH-BASED SIGNAL PROCESSING

A. V. Knyazev, D. Tian, H. Mansour, A. Gadde, A. Vetro (MERL), A. Malyshev (U. Bergen)

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We review some of our recent work in graph-based signal processing. We propose conjugate gradient-like and Nesterov iterative acceleration of repeated application of graph-based edge-preserving denoising filters: bilateral, guided, and total variation. Edge-enhancing graph-based denoising, using negative graph weights, is demonstrated. Guided and consistent signal reconstructions are combined into a reconstruction set, e.g., for image magnification.

Introduction to graph-based signal processing

Graph-based signal processing deals with signals defined on graphs, as their domain. It can be viewed as a natural extension of traditional signal processing of discrete signals from linear to general graphs. Graph-based processing can be applied to traditional signals as well as, e.g., in Figure 1 for temporal signal denoising.

Acceleration of repeated application of graph-based edge-preserving denoising filters

Graph-based spectral denoising is low-pass filtering of noisy signals based on eigendecompositions of a graph Laplacian matrix, as a generalization of a discrete cosine transform (DCT) for the traditional signals. While basis functions of DCT are explicitly known, computing the full eigendecomposition of the graph Laplacian is numerically expensive. Polynomial filtering avoids costly computations by projecting on Krylov subspaces.

Initial publications [2, 7, 12] start with repeatedly applying a fixed smoothing filter, whose coefficients are determined by a guiding signal defining the corresponding graph. The authors of [2] propose to accelerate filtering using Chebyshev polynomials. In [12], we additionally propose constructing the polynomials by the conjugate gradient method. In [7], we formulate a special variant of the conjugate method, which accelerates denoising of signals on graphs, and demonstrate that similar acceleration can be achieved with the Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) method from [4]. LOBPCG has been earlier used, e.g., for image segmentation in [9] and for multi-billion size matrices from material sciences by two Gordon Bell Prize finalists at ACM/IEEE Conferences on Supercomputing in 2005 [14] and 2006 [13] implemented on Japan's Earth Simulator. LOBPCG is publicly available in open source parallel software BLOPEX [10].

Our subsequent work [8] introduces a nonlinear iterative application of smoothing filters, where the filter at each iteration is determined by the currently processed signal. The resulting transform yields a nonlinear smoothing filter in contrast to the linear smoothing filter given by the repeated application the fixed filter at each iteration. The paper [8] presents a special variant of a nonlinear preconditioned conjugate gradient method and numerically demonstrates its high efficiency for accelerated denoising of one-dimensional signals.

In addition to the bilateral and guided image filters, we consider total variation denoising, formulate it in a filter form, and test all three filters for image denoising in [6]. We also numerically compare the conjugate gradient acceleration of nonlinear iterative smoothing filters and Nesterov's acceleration—the latter commonly used in a very different context of convex minimization.

Edge-enhancing using negative graph weights

In [6], the graph-based denoising is performed by projecting the noisy image to the Krylov subspace of the graph Laplacian, constructed using nonnegative weights determined by distances between image data corresponding to image pixels, serving as graph vertices. We extend in [5] the construction of the original graph Laplacian to a case, where some graph weights can be *negative*, in contrast to defining a *signed Laplacian*. Removing the positivity constraint provides a more accurate inference of a graph model behind the data, and thus can improve quality of filters for graph-based signal processing, e.g., denoising, compared to the standard construction, without affecting the costs. The use of the graph Laplacian in [5], where some weights can be negative, enhances the edges in the demoised signal as shown in Figure 1; also see, e.g., [3, 11].

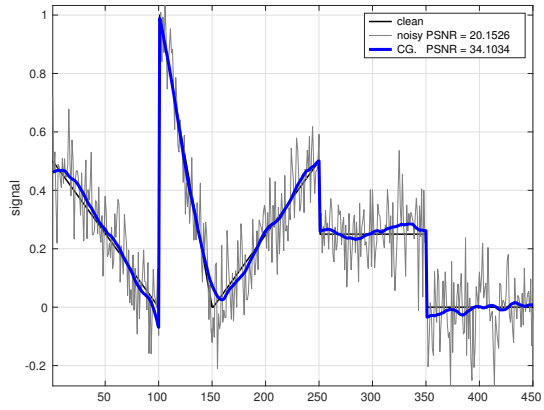


Figure 1: Negative graph weights for edge enhancing.

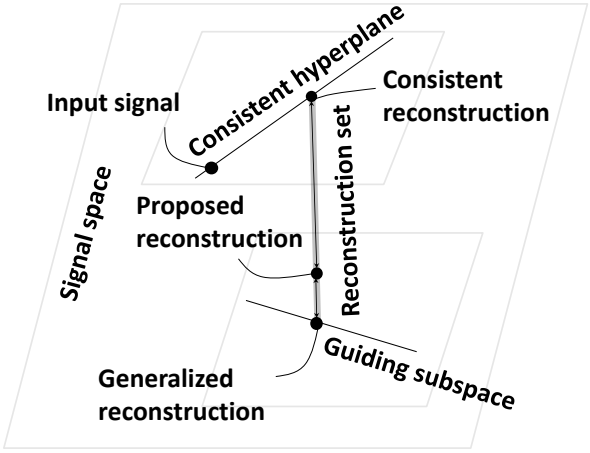


Figure 2: The reconstruction set.

Consistent and guided signal reconstructions

In [1], we investigate reconstructing a signal from its projection on a subspace, utilizing a guiding subspace that represents desired properties of reconstructed signals. Optimal reconstructed signals form a compact convex hull of consistent and generalized reconstructions, called the *reconstruction set*, as shown in Figure 2. We develop iterative conjugate gradient methods to approximate optimal reconstructions with low memory and computational costs. The effectiveness of the proposed approach is demonstrated for image magnification, where the reconstructed image quality is shown to exceed that of both consistent and generalized reconstruction schemes for noisy sampling.

References

[1] A. Gadde, A. Knyazev, D. Tian, and H. Mansour. Guided signal reconstruction with application to image magnification. *IEEE GlobalSIP*, pages 938–942, 2015. doi:10.1109/GlobalSIP.2015.7418335.

[2] A. Gadde, S. K. Narang, and A. Ortega. Bilateral filter: Graph spectral interpretation and extensions. *ICIP*, 2013. doi:10.1109/ICIP.2013.6738252.

[3] G. Gilboa, N. Sochen, and Y. Zeevi. Forward-and-backward diffusion processes for adaptive image enhancement and denoising. *IEEE Trans. Image Processing*, 11(7):689–703, 2002. doi:10.1109/TIP.2002.800883.

[4] A. Knyazev. Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method. *SIAM J. Scientific Computing*, 23(2):517–541, 2001. doi:10.1561/0600000020.

[5] A. Knyazev. Edge-enhancing filters with negative weights. *IEEE GlobalSIP*, pages 260–264, 2015. doi:GlobalSIP.2015.7418197.

[6] A. Knyazev and A. Malyshev. Accelerated graph-based nonlinear denoising filters. *CoRR*, abs/1512.00389, 2015. http://arxiv.org/abs/1512.00389.

[7] A. Knyazev and A. Malyshev. Accelerated graph-based spectral polynomial filters. In *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*, page 16, 2015. doi:10.1109/MLSP.2015.7324315.

[8] A. Knyazev and A. Malyshev. Conjugate gradient acceleration of non-linear smoothing filters. *IEEE GlobalSIP*, pages 245–249, 2015. doi:10.1109/GlobalSIP.2015.7418194.

[9] A. V. Knyazev. Modern preconditioned eigensolvers for spectral image segmentation and graph bisection. In Boley, Dhillon, Ghosh, and Kogan, editors, *Proceedings of the workshop Clustering Large Data Sets; Third IEEE International Conference on Data Mining (ICDM 2003)*, pages 59–62, Melbourne, Florida, 2003. IEEE Computer Society. http://math.ucdenver.edu/~aknyazev/research/conf/ICDM03.pdf.

[10] A. V. Knyazev, M. E. Argentati, I. Lashuk, and E. E. Ovtchinnikov. Block locally optimal preconditioned eigenvalue xolvers (blopex) in hypre and petsc. *SIAM J. Scientific Computing*, 29(5):2224–2239, 2007. doi:10.1137/060661624.

[11] L. Tang and Z. Fang. Edge and contrast preserving in total variation image denoising. *EURASIP Journal on Advances in Signal Processing*, 2016(1):1–21, 2016. doi:10.1186/s13634-016-0315-5.

[12] D. Tian, A. Knyazev, H. Mansour, , and A. Vetro. Chebyshev and conjugate gradient filters for graph image denoising. In *Proceedings of IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–6, Chengdu, 2014. doi:10.1109/ICMEW.2014.6890711.

[13] S. Yamada, T. Imamura, T. Kano, and M. Machida. High-performance computing for exact numerical approaches to quantum many-body problems on the earth simulator. In *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, SC '06, New York, NY, USA, 2006. ACM. doi:10.1145/1188455.1188504.

[14] S. Yamada, T. Imamura, and M. Machida. 16.447 tflops and 159-billion-dimensional exact-diagonalization for trapped fermion-hubbard model on the earth simulator. In *Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference*, pages 44–44, Nov 2005. doi:10.1109/SC.2005.1.

A SPECTRAL GEOMETRY BASED CONJECTURE FOR FAMILIES OF LARGE SPARSE STIELTJES MATRICES

P. Robert Kotiuga

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

A canonical identification of Diagonally Dominant Stieltjes (DDS) matrices with resistive electrical networks, yields graphs, graph Laplacians, and canonically associated zeta-functions and interrelationships. Finite Element discretizations of Boundary Value Problems for Laplace's equation yield DDS matrices. Under h-refinement and as a function of the dimension of the underlying domain, we obtain families of zeta-functions encoding spectral asymptotics, and outline a connection between certain classes of graphs with excluded minors, and their zeta-functions, in a manner relatively insensitive to overall graph size.

Preliminaries: From Networks to DD Stieltjes

A *Stieltjes* matrix $\mathbf{S}$ is a symmetric positive definite (SPD) with non-positive off-diagonal entries. As such,

$$\mathbf{S} = k\mathbf{I} - \mathbf{B} \tag{1}$$

where $\mathbf{B}$ is symmetric with non-negative entries, and k is positive. A geometric series argument shows that the inverse of $\mathbf{S}$ is SPD with non-negative entries. A symmetric matrix is diagonally dominant (DD) if any diagonal entry is greater or equal to the sum of the absolute values of the off-diagonal entries in the corresponding row. We will focus on DD Stieltjes (DDS) matrices.

Consider an electrical network with underlying graph $\mathcal{G}$, adjacency matrix $\mathbf{A_{adj,0}}$, edge-node incidence matrix $\mathbf{A_0}$ and diagonal branch conductance matrix $\mathbf{G}_B$, so the *nodal analysis* matrix becomes:

$$\mathbf{G_0} = \mathbf{A_0^T G_B A_0} \tag{2}$$

The graph Laplacian, $\triangle_{\mathcal{G}}$, of $\mathcal{G}$ is just $\mathbf{G_0}$ when $\mathbf{G}_B = \mathbf{I}$,

$$\triangle_{\mathcal{G}} \stackrel{\triangle}{=} \mathbf{A_0^T A_0} \tag{3}$$

Since any column of $\mathbf{A}_0$ has a ± 1 pair as nonzero entries, $\mathbf{A_{adj,0}}$ and $\triangle_{\mathcal{G}}$ are related by[1]:

$$\triangle_{\mathcal{G}} = \mathbf{D_0} - \mathbf{A_{adj,0}}, \tag{4}$$

where $\mathbf{D_0}$ is the diagonal matrix of node degrees. For regular graphs each node has degree d, $\mathbf{D_0}$ is a multiple of $\mathbf{I}$, and the spectrum of $\triangle_{\mathcal{G}}$ is that of $\mathbf{A_{adj,0}}$, shifted by d.

$\mathbf{G_0}$ is singular; the dimension of its null space equals the number of connected components of $\mathcal{G}$, and null vectors correspond to floating potentiasl on connected components of the network. In particular, it is *not Stieltjes*. Engineers sidestep this ambiguity by *grounding* one node in each connected component of the network. That is, they delete a row in the incidence matrix corresponding to a node in each connected component. The matrix is DDS:

$$\mathbf{G} = \mathbf{A^T G_B A} \tag{5}$$

A special case of $\mathbf{G}$, the *grounded graph Laplacian*,

$$\triangle_{g\mathcal{G}} \stackrel{\triangle}{=} \mathbf{A^T A} = \mathbf{D} - \mathbf{A_{adj}} \tag{6}$$

is also DDS. Here, the columns of $\mathbf{A}$ no longer have two nonzero entries if a branch ends up on a grounded node, $\mathbf{D}$ is the diagonal matrix of degrees of nongrounded nodes, and $\mathbf{A_{adj}}$ is the adjacency matrix with rows and columns corresponding to grounded nodes deleted. Since $\mathbf{G}$ and $\triangle_{g\mathcal{G}}$ are connected by a homotopy in the space of DDS matrices, their spectra are connected by an induced homotopy. Furthermore, for regular graphs, the spectrum of $\triangle_{g\mathcal{G}}$ is again that of $\mathbf{A_{adj}}$, shifted by the nodal degree d.

From DDS to Networks and on to Zeta Functions

An $(n-1)$ by $(n-1)$ DDS matrix, $\mathbf{S}$, can be identified with an n by n conductance matrix, $\mathbf{G_0}$ of a network as follows. Let $\mathbf{S}$ be the upper left $(n-1)$ by $(n-1)$ block of $\mathbf{G_0}$ and then complete the remaining row and column of $\mathbf{G_0}$ to obtain an n by n conductance matrix, any row or column of which sums to zero; off-diagonal elements in the last row and column of $\mathbf{G_0}$ represent the conductance from the nodes to ground. Identifying the ground node, and constructing a network in this manner, $\mathbf{G}$ is just $\mathbf{S}$.

Defining zeta functions via characteristic polynomials makes their elementary properties are transparent. First,

the *zeta function of a resistive network* by

$$\zeta_{\mathbf{G}}(t) \stackrel{\triangle}{=} \det((\mathbf{I} - t\mathbf{G})^{-1}). \tag{7}$$

It specializes to our *grounded graph zeta function*,

$$\zeta_{g\mathcal{G}}(t) \stackrel{\triangle}{=} \det((\mathbf{I} - t\triangle_{\mathbf{g}\mathcal{G}})^{-1}) \tag{8}$$

This is distinct from the generating function for counting nonself-intersecting loops of given lengths.

$$\zeta_{\mathbf{A_{adj}}}(t) \stackrel{\triangle}{=} \det((\mathbf{I} - t\mathbf{A_{adj}})^{-1}) \tag{9}$$

The latter two zeta functions are distinct from *graph zeta functions* or *Ihara zeta functions*[8], but their spectra are related in the case of regular graphs.

This canonical identification between DDS matrices and resistive electrical networks yields graphs, graph Laplacians, canonically associated zeta-functions, and connections between all these objects. This identification also relieves us from making parallel definitions for DDS matrices. We will now produce classes and families of DDS matrices which are useful for characterizing classes of networks spectrally.

Finte Elements, h-adaption and families of networks

We introduce the *finite element method* (FEM), and associated mesh refinement techniques, in order connect, the spectral asymptotics of elliptic boundary value problems and spectral graph theory. The starting point is the observation that 1st order FE discretizations of the Dirichlet integral subject to suitable boundary conditions not only yields a DSS matrix, but a network model where one has explicit fromulae for the branch conductances.

On a compact orientable manifold with boundary, the Dirichlet BVP involves a self-adjoint Fredholm operator and a spectral zeta function given by a Dirichlet series:

$$\zeta_{BVP}(s) = \sum_{i=1} \lambda_i^{-s} \tag{10}$$

There is also an analogous spectral expression for the zeta function of a network which is a result of Newton's identities being applied to the characteristic polynomial. The work of Dodziuk[2] shows that, under h-refinement, the network zeta functions converge as meromorphic functions to the continuum limit. Furthermore, in the continuum, one has formulae due to Weyl, for the asymptotic distribution of the eigenvalues. Hence one can make precise statements about the spectral asymptotics of the BVP as a function of the underlying domain. In this way, the zeta function is a meromorphic function which encoded the entire spectrum and this is the basis of spectral geometry. This is in contrast to problems like graph partitioning where only the lowest few eigenvalues are exploited.

Excluded minors[5] enter the graph partitioning literature only in the context of planar and low genus graphs[7],[3],[4]. However, combinatorial Hodge theory and its widespread use in finite element theory show that there is no problem defining and setting up a convergence theory for zeta functions of higher dimensional Dirichlet problems[2],[6]. From the identification of FE discretizations of Dirichlet boundary value problems (BVPs) with DDS matrices and network models, one concludes that, to the extent that the 1-skeleta of the finite element mesh are characterized as graphs with excluded minors, the Weyl asymptotics associated with the convergence under h-refinement is a statement about families of graphs with excuded minors. These 1-skeleta of d-dimensional simplicial FE meshes have no (d+2)-cliques, but further constraints involving excluded minors depend on d.

The simpest nontrivial conjecture asserts that once one has an analog of Kuratowski 's theorem for a d-dimensional FE mesh then, under h-refinement, there is a spectral characterization of 1-skeleton of the finite element mesh via the spectral zeta function. In low dimensions results like Kuratowski's characterization of planar graphs or others characterizations of linkless graphs are tied to Weyl asymptotics via the zeta function. However, in higer dimensions one has to exploit information beyond the 1-skeleton. This ensures that the conjecture is nontrivial.

References

[1] R. A. Brualdi. *The Mutally Beneficial Relationship between Graphs and Matrices*. Number 115 in CBMS. AMS, 2011.

[2] J. Dodziuk. Finite-difference approach to the hodge theory of harmonic forms. *American Journal of Mathematics*, 1976.

[3] G. N. P. J. A. Kelner, J. R. Lee and S.-H. Teng. Metric uniformaization and spectral bounds for graphs. *GAFA*, 2011.

[4] L. T. James R. Lee, Shayan O. Gharan. Multiway spectral partitioning and higher-order cheeger inequalities. *Journal of the ACM*, 2014.

[5] L. Lovasz. Graph minor theory. *Bulletin of the AMS*, 2006.

[6] W. Muller. Analytic torson and r-torsion of riemannian manifolds. *Advances in Mathematics*, 1978.

[7] S. R. P. Biswal, J. Lee. Eigenvalue bounds, spectral partitioning, and metrical deformations via flows. *J. ACM*, 2010.

[8] A. Terras. *Zeta Functions on Graphs*. Camb UP, 2011.

NODE AND LINK BASED EVOLUTIONARY GAMES ON COEVOLVING NETWORKS

Hsuan-Wei Lee, Nishant Malik, Peter J. Mucha

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

Investigations have taken place on different evolutionary games played on random graphs and social networks. Individuals could play a prisoner dilemma game on a graph with various strategies and the option to switch partners, hence, this game could be put into a framework of coevolving networks. We follow the node-based dynamics, which means each node can use only one strategy (cooperate or defect) towards all his or her neighbors. In it, we improve the existing pair approximation (PA) by using approximate master equations (AMEs), and we explore the parameter spaces and get accurate approximations of the dynamics and the degree distribution of the stationary states. Then we extend this node-based model to a link-based model, which means each node will not have to cooperate or defect with all of his or her neighbors. We also give different levels of analytical approximation to study the evolution. This paper is the first evolutionary games on networks based on link properties, and it gives us a more realistic scenario of the real world dynamics corresponding to decision making. We study the evolution, predict the stationary states of the networks, and compare the total utilities and other qualitative differences between these two models.

Node based Dynamics

On a coevolving network with a game, the vertices represent players and the edges denote the pairwise partnership between individuals. We want to study how players play a Prisoner's Dilemma (PD) game with the ability to change strategies on each individual links they have and switch partners. We start with an Erdős-Rényi network with N nodes and and M edges. Each node has an equal probability to be a cooperator (C, denoted by two-dimensional unit vector $s = [1, 0]^T$) or defector (D, $s = [0, 1]^T$) on one end of each link and engages in pairwise interactions with his immediate neighbors defined by the partner network. That is, individual i plays a PD game with all his social partners and obtains an income as

$$P_i = \sum_{j \in \mathcal{N}_i} s_i^T P s_j,$$

where $\mathcal{N}_i$ represents the neighborhood set of i and the 2 by 2 payoff matrix P would be

$$\begin{array}{c} \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cc} C & D \end{array} \\ \begin{pmatrix} 1 & 0 \\ 1+u & u \end{pmatrix} \end{array}$$

where a single parameter cost-to-benefit ratio $u \in (0, 1)$ is used to rescale the payoff matrix.

Follow [1], in each time step, we first randomly pick an edge that connects a pair of players with different strategies on all of their links, i.e., a CD link denoted by E_{ij} to update. Later on we will extend the model to allow all individuals to switch their defective partners, that is, both CC and DD links can be rewired. With a given probability w, node i and node j connected by the edge E_{ij} update their strategies; otherwise, E_{ij} is rewired (with probability $1 - w$). When one node updates its strategy, the node has a probability ϕ given by the Fermi function to change its state [3]. When link E_{ij} is rewired, the player with end state C unilaterally gets rid of the partnership with its neighbor with end state D on the edge E_{ij}. Suppose node i has the end with state C, then it will randomly pick a player k from the remainder population as its new partner. We will investigate the trade-off between strategy dynamics (controlled by the parameter u) and partner network adaptation (controlled by the parameter w) throughout the coevolution of network topologies and the dynamics on it.

We study this generalized coevolving network model with a combination of simulations and approximate analytic models. The frameworks of Mean Field Theory (MF), Pair Approximation (PA) and Approximate Master Equations (AME) have all been used effectively as analytical tools in similar settings. Among these approximations, AME can be used to achieve greater accuracy [2] [4]. The PA estimation was obtained by [1], and here we provide

the AME approximation. Let $C_{kl}(t)$ and $D_{kl}(t)$ be the fraction of cooperative and defective sites of total degree k which have number of defective neighbors l. We have the following ODE governing the time evolution of the C_{kl} compartment:

$$\begin{aligned}\frac{dC_{kl}}{dt} = w\Big\{ & \phi^D_{k,l}(k-l)D_{k,l} - \phi^C_{k,l} l C_{k,l} \\ & + \phi^C_{k,l+1}\gamma^S(l+1)C_{k,l+1} - \phi^C_{k,l}\gamma^S l C_{k,l} \\ & + \phi^C_{k,l-1}\beta^S(k-l+1)C_{k,l-1} - \phi^C_{k,l}\beta^S(k-l)C_{k,l}\Big\} \\ & + (1-w)\Big\{\frac{N_C}{N}\big[(l+1)C_{k,l+1} - lC_{k,l}\big] \\ & + \frac{N_{CD}}{N}\big[C_{k-1,l} - C_{k,l}\big]\Big\}\end{aligned}$$

Similarly the ODE governing the time evolution of the D_{kl} compartment is:

$$\begin{aligned}\frac{dD_{kl}}{dt} = w\Big\{ & -\phi^D_{k,l}(k-l)D_{k,l} + \phi^C_{k,l} l C_{k,l} \\ & + \phi^D_{k,l+1}\gamma^I(l+1)D_{k,l+1} - \phi^D_{k,l}\gamma^I l D_{k,l} \\ & + \phi^D_{k,l-1}\gamma^I(k-l+1)D_{k,l-1} - \phi^D_{k,l}\gamma^I(k-l)D_{k,l}\Big\} \\ & + (1-w)\Big\{\big[(k-l+1)D_{k+1,l} - (k-l)D_{k,l}\big] \\ & + \frac{N_{CD}}{N}\big[D_{k-1,l} - D_{k,l}\big]\Big\}\end{aligned}$$

- For a $D_{k,l}$ ($C_{k,l}$ similarly):
  $P_D = l\cdot u + (k-l)\cdot(1+u)$, $P_C = 1\cdot\frac{2N_{CC}}{N_C} + 0\cdot\frac{N_{CD}}{N_C}$, and

$$\phi^D_{k,l}(D \leftarrow C) = \frac{1}{1+exp[\beta(P_D - P_C)]}.$$

Link based Dynamics

We use a very similar setting with the node-based dynamics. The utility matrix is still controlled by the parameter u, and the individual still has probability $1-w$ to switch a partner. Different from the node-based dynamics, note that a node here doesn't have its own state, instead, it has different states on its side of its edges. Figure 2 shows different levels of analytical approximation of the link based dynamics. Let N_{k,m_C} denote the quantity of node with degree k and m_C of C-stubs and $N_{k,m_{CC},m_{CD},m_{DC},m_{DD}}$ denote the quantity of node with degree k, m_{CC} of CC links and so on. There are corresponding N_{k,m_C} and $N_{k,m_{CC},m_{CD},m_{DC},m_{DD}}$ differential equations to MF and

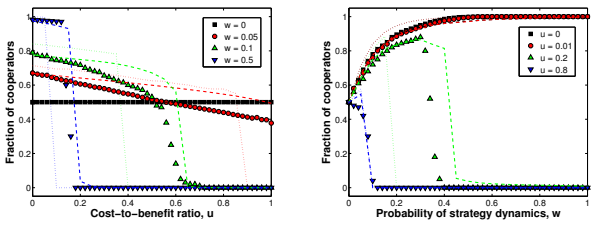


Figure 1: Fraction of cooperators versus cost-to-benefit ratio u with different w values (left) and fraction of cooperators versus w with different u values (right) in stationary states. Dots are the averages of 100 of simulation results, dotted lines are the pair approximation (PA), and the dashed lines are the results of approximate master equations (AME).

PA. In NS16, we will present these analytical estimations, and more importantly, give an overall comparison of the behaviors of node and link based dynamics.

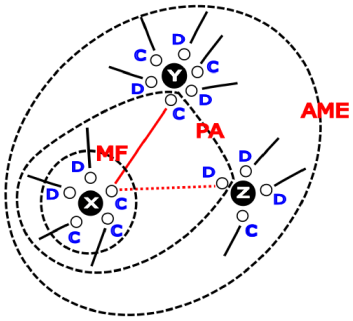


Figure 2: Different levels of analytical approximations, mean field (MF), pair approximation (PA), and approximate master equation (AME), respectively, in link-based dynamical evolutionary games in a network setting.

References

[1] F. Fu, T. Wu, and L. Wang. Partner switching stabilizes cooperation in coevolutionary prisoner?s dilemma. *Physical Review E*, 79(3):036101, 2009.

[2] J. P. Gleeson. Binary-state dynamics on complex networks: Pair approximation and beyond. *Physical Review X*, 3(2):021004, 2013.

[3] C. Hauert and G. Szabó. Game theory and physics. *American Journal of Physics*, 73(5):405–414, 2005.

[4] V. Marceau, P.-A. Noël, L. Hébert-Dufresne, A. Allard, and L. J. Dubé. Adaptive networks: Coevolution of disease and topology. *Physical Review E*, 82(3):036116, 2010.

SOFTWARE LIBRARY FOR SCALABLE MULTI-PHYSICS MULTI-SCALE NETWORK SIMULATION: APPLICATION TO WATER DISTRIBUTION SYSTEMS

Daniel A. Maldonado, Hong Zhang, Shrirang Abhyankar

Argonne National Laboratory

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

In this work, we present an application of DMNetwork [1], a software library for scalable network simulation included in the scientific computing library PETSc[2], on transient analysis of water distribution system. We use DMNetwork for the development of WASH, a scalable simulator for water distribution systems that includes high-fidelity physical model couplings.

Introduction

Developing scalable numerical software for large-scale network applications is challenging due to its underlying unstructured geometry and the lack of abstractions to encapsulate general networks. The common thread in all network simulations is that computations are done on nodes and edges of a graph with the components (physics) describing the nature of the problem to be solved. With this simplistic design emanating from nodes, edges, and components, we have developed a software library, DMNetwork[1], that provides the capability of rapidly developing network simulations and access to a variety of high-performance linear, nonlinear, and time-stepping solvers through PETSc. In this work, we use DMNetwork for the development of WASH, a simulator for hydraulic distribution systems. As opposed to previous DMNetwork applications on power grid and radio networks, WASH is unique because the edges represent water pipes modeled by partial differential equations, which is a typical characteristic of commodity-supply networks.

DMNetwork

DMNetwork is a class recently developed in PETSc to provide abstractions for representing general unstructured networks, such as communication networks, power grid, and graphs. It is general and flexible in such a way that, the nodes can be used to present various physical models, the edges build the connections between the models. New nodes and edges can be easily inserted, and the existing ones can be removed or updated with minimum local changes. On multiple processors, DMNetwork can partition the network using the available graph partitioning packages (ParMetis and Chaco) and move the user data describing the physics to the appropriate processor. DMNetwork being an in-built object in PETSc can take advantage of its myriad of linear, nonlinear, and time-stepping solvers available. Moreover, the choice of creating a overarching solver comprising solvers (for e.g. Schur-complement) for each individual system or using a monolithic solver (for e.g. LU factorization) can be done at run-time.

WASH

The WASH software package is built on top of DMNetwork. Its goal is the simulation of a highly complex water network where models of different physical nature interact. We strive to build a platform where scientist of different areas of expertise can collaborate and include their models without the need of understanding the mechanics of the couplings. The equations governing WASH represent transient analysis to determine maximum pressures and flows along a network for a given disturbance after some disturbance has occurred. This disturbance can be the closure of a valve, the change of water demand, the failure of a pump, or other events of interest. A sudden surge in pressure can lead to the burst of a pipe or the mis-functioning of a pump.

The physical model of the water network from [3, Chap. 3] can be described with the following set of partial differential equations for each pipe:

$$\frac{\partial Q}{\partial t} + gA\frac{\partial H}{\partial x} + RQ\,|Q| = 0 \tag{1}$$

$$a^2\frac{\partial Q}{\partial x} + gA\frac{\partial H}{\partial t} = 0 \tag{2}$$

which are the momentum and continuity equations for the flow $Q(x,t)$ and pressure $H(x,t)$, with f being the friction coefficient, D and A the diameter and area of the pipe, and $R = f/(2DA)$.

In order to have a stable computation for the finite-difference scheme, we have to take into account the Courant-Friedrichs-Lewy (CFL) condition in the discretization scheme. In this case, for the unidimensional flow in pipes:

$$C_N = a\frac{\Delta t}{\Delta x} \leqslant 1 \tag{3}$$

where a is related to the speed of sound in the fluid.

Since the time step must be the same for all pipes, the discretization in space needs to be the same too. This introduces difficulty when the network contains pipes of acute different length since the shorter pipe will determine the discretization of the whole network. Traditionally, this has been addressed though interpolation or simplification [3] of the overall model. The whole set of pipes in the network is coupled by the continuity of flows at each node.

In the following figures, we show preliminary results obtained on the WASH code on contiguously linked pipes. The figures show the pressure of the wave (head, in meters), the period and how it dissipates over time. The overall length of the network increases as more pipes are added. The pressure wave period will change depending on the overall length of the transmission network. We can see this, by comparing the period of Figure 1 and Figure 2.

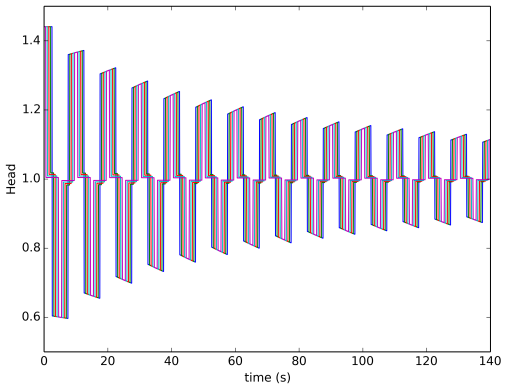


Figure 1: Network with 5 contiguous pipes.

Moreover, as the wave travels more distance between both ends, the friction losses are more acute. This phenomenon can be observed in the referenced plots. A modified version of the 5 pipes network where all the pipes have different diameter and friction, and one of the pipes has a very high friction coefficient is shown in Figure 3.

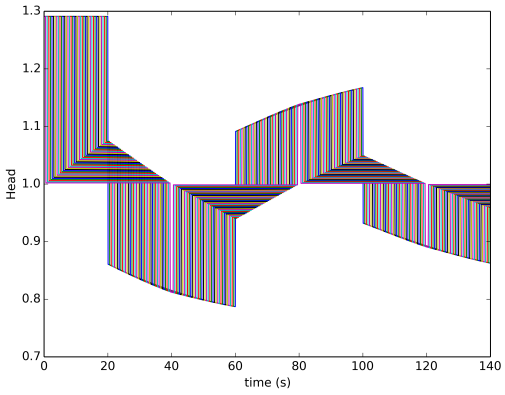


Figure 2: Network with 40 contiguous pipes.

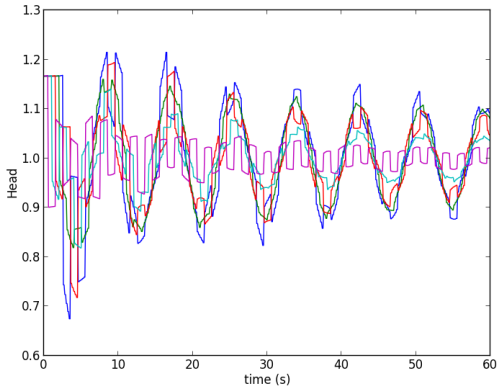


Figure 3: Pressure wave in a complex network.

References

[1] S. Abhyankar, J. Brown, M. Knepley, F. Meier, and B. Smith. Abstractions for expressing network problems in petsc. In *SIAM Workshop on Network Science*, 2014.

[2] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, and H. Zhang. PETSc Web page. http://www.mcs.anl.gov/petsc, 2015.

[3] M. Chaudhry. *Applied Hydraulic Transients*. SpringerLink : Bücher. Springer New York, 2013.

GEODESIC DISTANCES IN RANDOM DELANNOY LATTICES

Hans J. Haucke and Ira S. Moskowitz

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

The random Delannoy graph is a square lattice with randomly inserted diagonal edges. Expected geodesic distances are calculated exactly for some finite cases, and upper and lower bounds are shown for lattices of arbitrary size. This is an instance of the longest monotone subsequence problem in which repeated values are allowed.

Introduction

Graphs can abstract features of networks which are physically instantiated in Euclidean space. Frequently the graph properties, such as the well-known small world property, have no particular relation to the physical and geometric characteristics. In that situation, the graph and geometry give very different views of the same structure. However, in this paper we focus on the complementary circumstance, where the graph structure and geometric structure are very similar. In such cases, physics-based models are well motivated models for data flow.

A tractable, but non-trivial example, the random Delannoy lattice [7], is discussed. For this case two metrics, the geodesic distance (minimal number of links between two nodes) and the geometric distance (L^2 norm), may be quantitatively close, but not asymptotically proportional. Note that the Delannoy lattice is a probabilistic construction, so when we refer to the geodesic distance, the mean value is intended.

We show that geodesics for the random Delannoy lattice are closely related to the problem of the longest monotone subsequence in a sequence of integers that my have repeated values. A calculational approach is outlined, similar to that used for the extensively studied non-repeating problem [1]. We present solutions for special cases.

In earlier work [4], we discussed how geodesic distances in a network may approximate Euclidean distances. We demonstrated [4] that for certain graphs, a close correspondence exists between geodesic and Euclidean distances, such that geodesic distance can serve as an accurate proxy for the Euclidean distance. Graphs in the beta-skeleton family [5] including the Gabriel graph [3] were studied numerically.

Here we study a particularly interesting network consisting of the two-dimensional infinite square lattice $\mathbb{Z}^2$ with added diagonal edges. This lattice has 4 rectilinear edges at each node. To these, next-nearest-neighbor diagonals are added in a probabilistic manner, with $+45°$ and $-45°$ diagonals equally likely, but independently chosen. Owing to the simple geometry, this lattice, which we call the *random Delannoy lattice* $\mathcal{L}_\lambda$, is more amenable to analysis than networks having random node locations. A Bernoulli parameter λ, with $0 \leq \lambda \leq 1$, controls the density of diagonal linkages.

In this study, the quantity of interest is the geodesic distance [6] $\mathfrak{d}$ between any two nodes. For quantities averaged over an ensemble of lattices with a fixed value of λ, one can take the two nodes, with no loss of generality, to be the origin and the node at p, q (in the northeast sector, $p \geq q > 0$). For a particular instantiation, the geodesic distance between these two nodes is $\mathfrak{d}(p, q)$; ensemble averaging over the positions of the diagonals gives $E(\mathfrak{d}(p, q); \lambda)$. Note that if all the diagonals were present, the number of possible lattice paths would be counted by the Delannoy number [7]. Provided $\lambda < 1$, typically only a subset of those paths are accessible in any particular instantiation of the random lattice. Trivially, $\mathfrak{d}(p, q) = max(p, q)$ for $\mathcal{L}_1$ while $\mathfrak{d}(p, q) = p + q$ for $\mathcal{L}_0$, and it is apparent that $max(p, q) \leq \mathfrak{d}(p, q) \leq p + q$ for $\mathcal{L}_\lambda$.

Geodesic distances on the lattice clearly differ from Euclidean distances in the extremal cases corresponding to no diagonals or all diagonals being present. Euclidean behavior would require $\mathfrak{d}(p, q) \to \sqrt{p^2 + q^2}$ as p, $q \to \infty$. Based on numerical evidence, near $\lambda \sim 0.3$, for $\sqrt{p^2 + q^2} \sim 10$, geodesic distances in $\mathcal{L}_\lambda$ are nearly isotropic, and approximately proportional to the Euclidean distance. However, simple analysis will demonstrate that actual isotropy does not occur in this model.

Geodesic Distance on the Delannoy Lattice

To compute the geodesic distance between the origin and (p, q), note that only $+45°$ diagonals play a role. Travers-

ing an opposite diagonal will never lead to a shorter path than moving along the perpendiculars. Further inspection shows that it is also not advantageous to traverse positive diagonals outside the bounding rectangle formed by the origin and (p, q). The shortest path always moves up and/or to the right at each step and, subject to that constraint, includes as many positive diagonals as possible. Therefore for any instantiation $\mathfrak{d}(p, q)$ is the same for the finite rectangular patch of lattice bounded by the origin and (p, q), as for the infinite lattice. Henceforth we will focus on $\mathfrak{d}(p, q)$ and its ensemble average; the $+45^\circ$ diagonals will be referred to as the "elements" of the lattice.

Computing $\mathfrak{d}(p, q)$ is therefore reduced to considering all potential shortest paths and determining the maximum number of elements. Those paths move from the origin up, right, or diagonally northeast. Let k denote the maximum number of elements in any shortest path; then $\mathfrak{d}(p, q) = p + q - k$. Consider m elements on the finite lattice patch, the northeast corner of the diagonals being located at m distinct nodes $\{(p_i, q_i)\}$, with $0 < p_i \leq p$ and $0 < q_i \leq q$. The expected value for m is λpq. Values for both p_i, q_i may repeat, and for large p (or q), typically there will be multiple elements in a row (or column). We want to know the expected value for the geodesic distance $E(\mathfrak{d}) = p+q-E(k)$. It suffices to search all permutations σ having lengths k, with $1 \leq k \leq min(p, q)$, checking that both $p_{\sigma(1)} < p_{\sigma(2)} < ... < p_{\sigma(k)}$ and $q_{\sigma(1)} < q_{\sigma(2)} < ... < q_{\sigma(k)}$. If the inequalities are satisfied, there is a path including at least k elements. These elements form a poset chain.

We concentrate the rest of our efforts on determining the expected chain length $E(k)$, which in turn gives us the expected geodesic distance as $p + q - E(k)$.

The binomial distribution gives the probability of inserting exactly m elements into the lattice. We define $N_{p,q}(m, k)$ as the number of ways to form a path of length k in a $p \times q$ lattice having m elements. Then the expected chain length is given by weighting k by {probability of m elements} $\times$ {probability of finding a length k chain given m elements}. One way of counting all possible ways of inserting m elements is to ask how long a chain they form, thus $\sum_{k=1}^{min(p,q,m)} N_{p,q}(m, k) = \binom{pq}{m}$. There must be one row, column, and element for each ordered element in the path, so $N_{p,q}(m, k) > 0$ only when $k \leq \min(p, q, m)$. Each element in a path lies on a unique row and column and there are $N_{p,q}(k, k) = \binom{p}{k}\binom{q}{k}$ different ways to choose k distinct rows and columns (as in the Delannoy number). Though we have the same lattice as for Delannoy's problem, here we wish to iterate through the randomly placed elements in the path, and ignore the multiplicity arising from various choices of the rectilinear paths. Unlike the problem of finding the longest monotone subsequence of a random permutation, here there are typically many repeated entries in each row and column.

A well-known problem is that of determining the longest monotone subsequence [1][2] in a random sequence of m *distinct* integers, or a random permutation of m elements. Each permutation has a length k, defined as the number of elements in the longest monotone subsequence, and it is known that for large m, the (mean of k) $\to 2\sqrt{m}$, see [1]. In the Delannoy lattice, given p, q, m, the probability distribution for k is $\mathcal{P}_{p,q}(m, k) := N_{p,q}(m, k)/\binom{pq}{m}$. Assuming that p, q are very large compared to the number of elements, the node locations can be considered real valued rather than integer valued. Denote the number of monotone subsequences of length k for permutations of m elements as $M(m, k)$. One expects $\lim_{p,q\to\infty} \mathcal{P}_{p,q}(m, k) = M(m, k)/m!$. Both expressions give the chain length probability; the right-hand expression does this over all permutations, while the left side does this for random, possibly repeating, orderings, but repetition disappears in the limit $\lambda = \frac{m}{pq} \to 0$. (The infinite lattice limit could be called the "dilute" limit, as the concentration of diagonals $\lambda \to 0$.) Thus the longest monotone subsequence should provide a bounding case for the random Delannoy lattice.

In the full version of the paper, an efficient algorithm for $N_{p,q}(m, k)$ is outlined. Additionally, we develop exact and Monte Carlo calculations for the mean geodesic length, and derive analytic bounds.

References

[1] R. Baer and P. Brock. Natural sorting over permutation spaces. *Mathematics of Computation*, 22, 1968.

[2] J. Baik, P. Deift, and K. Johansson. On the distribution of the length of the longest increasing subsequence of random permutations. *J. AMS*, 12, 2003.

[3] K. Gabriel and R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18, 1969.

[4] H. Haucke and I. Moskowitz. asymptotically euclidean graphs. *PREPRINT*.

[5] D. G. Kirkpatrick and J. D. Radke. A framework for computational morphology. *Journal of Integer Sequences*, 1985.

[6] M. Newman. *Networks*. 2010.

[7] R. Sulanke. Objects counted by the central delannoy numbers. *Journal of Integer Sequences*, 6, 2003.

COUNTING MOTIFS IN STRUCTURALLY SPARSE GRAPHS

Michael P. O'Brien, Felix Reidl, Blair D. Sullivan

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

Exploiting the structure of a sparse network can yield efficient algorithms for NP-hard problems like motif counting. We introduce CONCUSS, the first implementation of an algorithmic pipeline for bounded expansion graph classes. Based on empirical evaluation and testing of CONCUSS, we discuss new theoretical advances targeted to improve the experimentally-identified weaknesses.

Introduction

Scalable graph algorithms are a key component of deriving observations from relations in large data sets. For example, counting the number of occurrences of a particular subgraph, often called *motif counting*, has proven useful in comparing brains between species [1] and detecting cyberattacks [7]. Unfortunately, polynomial time algorithms for counting arbitrary motifs are unlikely to exist [8]. This has led to heuristic and sampling approaches that vary significantly across domains and whose reasons for success are not well understood.

The mathematics and theoretical computer science communities have a rich history of algorithmic research that circumvents these scalability issues by exploiting the *structural sparsity* of graphs [3, 5, 10, 12]. These algorithms are designed to efficiently operate on graphs with certain underlying structural features. An approach of this sort is attractive because real-world data sets are not arbitrarily structured. For example, it has been observed that graphs in multiple, unrelated domains are sparse, exhibit clustering [14], and have heavy-tailed degree distributions [2].

However, employing sparse graph algorithms in large-scale data analytics is a non-trivial task. The primary objective of the algorithms community has historically been to design efficient algorithms for a given problem, measured by the worst-case asymptotic computational complexity. As a result, algorithms exploiting structural sparsity often have massive constants hidden in big-O notation and/or non-trivial implementation details left unaddressed. Simply put, the existing literature is ripe with *efficient* algorithms but not necessarily *practical* ones.

Structural Sparsity

Information about the structure of a graph can allow some NP-hard problems to be solved efficiently. For example, if the graph is a tree we can use dynamic programming to count the number of motifs in polynomial time [9]. In this way, there is an efficient algorithm for motif counting on an entire *class* of graphs (trees). Though data sets like social networks are unlikely to be trees, the *sparse graph hierarchy* identifies a number of other graph classes, each of which has associated algorithms that operate efficiently on graphs in that class. These classes are organized in a nested fashion, e.g. trees are a subset of graphs with bounded treewidth. As a result, moving up the hierarchy implies a tradeoff: including more diverse graphs gives less structure to exploit algorithmically.

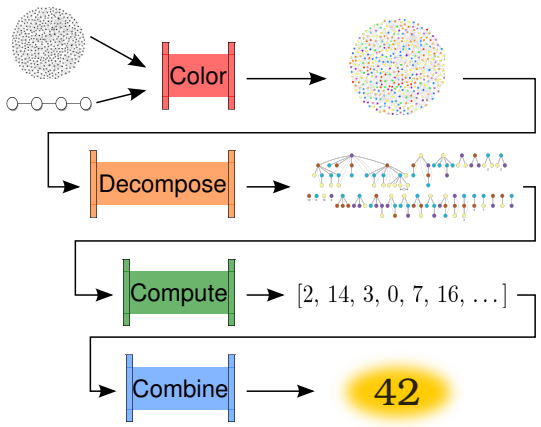


Figure 1: The workflow for counting motifs efficiently in bounded expansion classes.

Recent work has identified that classes of *bounded expansion* may occupy a "sweet spot" in the sparse graph hierarchy: high enough to capture the structure of many real-world data sets but low enough to provide useful algorithmic tools [6]. Bounded expansion graphs can be characterized as globally sparse, but having pockets of localized density. This corresponds to the previously known existence of communities in data from multiple domains.

The main algorithmic tools for bounded expansion classes are *p-centered colorings* [4], which guarantee each set of fewer than p colors can be decomposed into subgraphs belonging to a lower class on the hierarchy known as *bounded treedepth*. Appropriate efficient algorithms for bounded treedepth classes can be run on the subgraphs, and then their subsolutions can be combined to create a solution for the entire graph (Figure 1). Though this approach requires enumerating all subsets of $p-1$ colors, graphs in classes of bounded expansion admit p-centered colorings with a bounded number of colors, which decouples the number of color subsets from the graph size.

CONCUSS

The algorithmic framework in Figure 1 had previously only been described theoretically [10, 6] and it is unclear to what extent the asymptotic analysis glosses over practical performance problems. For this reason, we pursue an algorithm engineering methodology [13]: experimentally evaluate an algorithm, determine its weaknesses, address them with new theory, and repeat. To accomplish the first goal, we created CONCUSS [11], an open-source software tool written in Python and the first implementation of any bounded expansion algorithmic pipeline. Many of the non-trivial details of CONCUSS were implemented multiple ways. We designed experiments to test how the different choices of these details affect the performance on graphs of different sizes and to determine how the different stages contributed to the total run time.

Alternative Colorings

These experiments identified the Color stage as the portion of CONCUSS in most need of improvement. The current state-of-the-art algorithms for finding p-centered colorings iteratively add constraints to the vertices until a greedy coloring becomes p-centered. The number of added constraints increases with each iteration, which in turn cause the algorithm to be time- and memory-consuming.

The number of colors assigned in the Color stage also has ramifications on downstream computation because the Decompose, Compute, and Combine stages are each executed once per subset of $p-1$ colors. While the aforementioned approach does give bounds on the number of colors used, it may use far more than the minimum number of colors required. The existence of this gap was verified empirically; adding simple heuristics in CONCUSS often significantly reduced the number of colors.

To mitigate these problems, we introduce *p-linear colorings* as an alternative to p-centered colorings. We created algorithms for finding p-linear colorings, bounded the number of colors required in a bounded expansion class, and proved algorithmically useful properties of the subsequent decomposed subgraphs. They can be found using similar methods to p-centered colorings, but with fewer iterations and using fewer colors in total. Like p-centered colorings, p-linear colorings also allow decomposition into bounded treedepth subgraphs. As a tradeoff, these decompositions may have larger treedepth than those from the p-centered colorings. In future work, we look to implement p-linear colorings in CONCUSS and evaluate whether the tradeoff results in a lower running time.

References

[1] J. Berg and M. Lässig. Local graph alignment and motif search in biological networks. *PNAS*, 101(41):14689–14694, 2004.

[2] A. Clauset, C. R. Shalizi, and M. E. Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

[3] M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. van Rooij, and J. O. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *FOCS 2011*, pages 150–159. IEEE, 2011.

[4] P. O. de Mendez et al. *Sparsity: graphs, structures, and algorithms*, volume 28. Springer Science & Business Media, 2012.

[5] E. D. Demaine and M. Hajiaghayi. The bidimensionality theory and its algorithmic applications. *The Computer Journal*, 51(3):292–302, 2008.

[6] E. D. Demaine, F. Reidl, P. Rossmanith, F. S. Villaamil, S. Sikdar, and B. D. Sullivan. Structural sparsity of complex networks: Random graph models and linear algorithms. *CoRR*, abs/1406.2587, 2014.

[7] W. Eberle, L. Holder, and D. Cook. Identifying threats using graph-based anomaly detection. In *Machine Learning in Cyber Trust*, pages 73–108. Springer, 2009.

[8] M. Jerrum and K. Meeks. The parameterised complexity of counting connected subgraphs. *CoRR*, abs/1308.1575, 2013.

[9] D. W. Matula. Subtree isomorphism in $O(n^{5/2})$. *Algorithmic aspects of combinatorics*, 2:91, 2011.

[10] J. Nešetřil and P. O. de Mendez. Grad and classes with bounded expansion ii. algorithmic aspects. *European Journal of Combinatorics*, 29(3):777–791, 2008.

[11] M. P. O'Brien et al. CONCUSS: Version 1.0, Sept. 2015. 10.5281/zenodo.30281.

[12] G. Philip, V. Raman, and S. Sikdar. Solving dominating set in larger classes of graphs: Fpt algorithms and polynomial kernels. In *Algorithms-ESA 2009*, pages 694–705. Springer, 2009.

[13] P. Sanders. *Algorithm engineering*. Springer, 2011.

[14] D. J. Watts and S. H. Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):440–442, 1998.

LATENT FEATURE DISCOVERY IN NOISY NETWORK DATA

Chong Zhou, Randy Paffenroth

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Computer Network Data Challenges

There are many important aspects of network analysis and cyber security, and two key tasks are intrusion and anomaly detection. For example, many anti-virus tools are based on "signatures" of known viruses, worms, and Trojan horses. Here, a "signature" usually means a certain sequence of bits in executable code, TCP/UDP packets, or events (e.g., network ports accessed in a certain order). If any of the signatures are detected during a scan, the anomalies or attacks will be flagged. Clearly, the availability of appropriate features and signatures plays a pivotal role in the accuracy and effectiveness of such detection systems [1].

One of the main limitations of such "supervised" detection approaches is that they can not detect new forms of anomalous or malicious behaviors that do not match existing signatures. In addition, many classes of aberrant behavior can be embedded in large swaths of legitimate activity. Accordingly, it is our goal to build systems that extend the classes of features that are available to such systems. To this end, we have deployed two methods, robust principal component analysis applied to second order data and robust deep auto-encoders, to uncover latent features in network data. Both of these methods mitigate the fact that outliers in sensor networks can skew the approximated low-dimension representation arbitrarily far away from the true low-dimension representation [6]. The robust principal component analysis (RPCA) we propose assumes that multiple-sensor activities throughout a network are linearly correlated, and it represents the whole network's activity using a linear combination of fewer features. Both outliers and the low-rank latent structure of the data can be captured *simultaneously* by such second order analysis. On the other hand, the deep auto-encoder we propose tries to recover the whole network activity through a *non-linear* combination of features. The two approaches are tightly coupled since the auto-encoder we propose leverages the *same* type of sparse technique for detecting sparse outliers as do the RPCA techniques. One promising part of such deep auto-encoders is that, by extending the number of layers, deep auto-encoders can capture highly non-linearity aspects of the network data [3].

Latent Feature with Robust PCA

Robust principal component analysis (RPCA) refines principal component analysis (PCA) by making PCA robust to outliers. RPCA allows for the careful teasing apart of sparse outliers so that the remaining low-rank approximation is faithful to the true low-rank subspace describing the raw data [2][8][9][10]. We argue that our input data M, for which we classically use second order covariance information, can be decomposed into three parts

$$M = L_0 + S_0 + \varepsilon,$$

where L_0 is a low-rank matrix which can be linearly recovered by factors Y through $L_0 = YY^T$, S_0 is a sparse matrix which can not be captured by the low-rank features and ε is point-wise error. This matrix decomposition can created by way of the following optimization problem [2]:

$$\underset{L_0,S_0}{argmin} \;\; ||L_0||_* + \lambda||S_0||_1$$

$$s.t. \; |M - L_0 - S_0| \preceq \varepsilon$$

where the $||\cdot||_*$ is nuclear norm, and $||\cdot||_1$ is one norm. Through this model, the low-rank matrix L_0 can be interpreted as the background network features which widely influence the network sensors, and the sparse matrix S_0 are anomalies which can not be captured by low-dimensional features. Anomalies in S_0 do not necessarily imply malicious intent, but such sparsely correlated phenomena often bear closer examination in real-world network problems.

Latent Feature Discovery with Deep Models

In RPCA, the low-rank structure of L_0 implies it is a linear combination of latent features. However, non-linear combinations could capture more complicated sensor activities throughout the network. We assume that the majority of the network activities could be represented by non-linear combinations of a few latent features, and the rest of the activities are outliers. Our target is to learn these

latent features by non-linearly projecting and combining the observations of sensor activities. The outliers will also emerge as the unrepresentable parts from the non-linear combination of the latent features. We deploy deep models to discover non-linear latent features. In particular, deep auto-encoders are often used for learning a representation or effective encoding of the original data, in the form of parameters in the hidden layers [3]. One promise of deep learning is to replace handcrafted features with efficient algorithms for unsupervised or semi-supervised feature learning and hierarchical feature extraction[5].

An auto-encoder is a feed forward multi-layer neural network which the output target is the input itself. This process perhaps seems trivial, but the meaningful part is the dimension-reduced hidden layers which are trained to be as lossless as possible representations of the input. A typical auto-encoder with one hidden layer consists of an encoder $E(\cdot)$ and a decoder $D(\cdot)$. The low-dimensional representation $h = E(X)$ is computed after the cost function $||X - D(E(X))||_2$ minimized. The desired non-linearity comes from non-linear encoders $E(\cdot)$ and decoders $D(\cdot)$. However, some classic types of auto-encoders could suffer heavy losses in the presence of outliers. In particular, outliers may mislead the direction or skew the curvature of the manifold. Eliminating the influences of outlying observations promises to improve the robustness of deep models and makes them more applicable to the real-world network data.

Accordingly, in our work, we propose an improved model that is a novel combination of deep auto-encoders and RPCA. Similar to the previous section, we decompose our input data into $X = L + S$ where the L is a matrix that can be represented by a non-linear manifold and the S contains the outliers which will corrupt and skew the non-linear manifold. Our loss function for a given layer is a summation of the sparseness of S and the reconstruction error of L, namely

$$\underset{W,b,S}{argmin} \; ||L - D_{W,b}(E_{W,b}(L))||_2 + \lambda||S||_1$$

$$s.t.\; X - L - S = 0$$

where $E(\cdot)$ denotes an encoder, $D(\cdot)$ denotes a decoder, S captures the outlying observations, and L is a low-dimension manifold. We used L as input data to a standard deep auto-encoder model to learn a low-dimensional representation on a non-linear manifold. After training the whole model, L should retain a good representation of X inside the hidden layer.

Network Applications

The effectiveness of such robust methods have already been demonstrated in the literature [2], and in this poster, we propose to underscore their applicability to network problems. In particular, we will show how PCAP captures of computer networks can be processed using these ideas to provide more efficient features for the detection of anomalies. In particular, such ideas have been used in other domains, such as image processing, where the raw features (e.g., pixel values) are used to generate high-level features (e.g., parts of faces), and facial recognition is performed on these high-level features. Similarly, in our work, raw packet captures are processed to create high-level features which are better for anomaly detection.

References

[1] Marcus A. Maloof "Machine Learning and Data Mining for Computer Security: Methods and Applications" *Advanced Information and Knowledge Processing ISSN 1610-3947 ISBN-10: 1-84628-029-X*

[2] Paffenroth, Randy, et al. "Space-time signal processing for distributed pattern detection in sensor networks." *Selected Topics in Signal Processing, IEEE Journal of 7.1 (2013): 38-49.*

[3] Deng, Li, and Dong Yu. "Deep learning: methods and applications." *Foundations and Trends in Signal Processing 7.34 (2014): 197-387.*

[4] Bengio, Yoshua, Aaron Courville, and Pierre Vincent. "Representation learning: A review and new perspectives." *Pattern Analysis and Machine Intelligence, IEEE Transactions on 35.8 (2013): 1798-1828.* APA

[5] Bengio, Yoshua. "Learning deep architectures for AI." *Foundations and trends in Machine Learning 2.1 (2009): 1-127.*

[6] Ringberg, Haakon, et al. "Sensitivity of PCA for traffic anomaly detection." *ACM SIGMETRICS Performance Evaluation Review. Vol. 35. No. 1. ACM, 2007.*

[7] Vincent, Pascal, et al. "Extracting and composing robust features with denoising autoencoders." *Proceedings of the 25th international conference on Machine learning. ACM, 2008.*

[8] Cands, Emmanuel J., et al. "Robust principal component analysis?." *Journal of the ACM (JACM) 58.3 (2011): 11.*

[9] Zhou, Zihan, et al. "Stable principal component pursuit." *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on. IEEE, 2010.*

[10] Ringberg, Haakon, et al. "Sensitivity of PCA for traffic anomaly detection." *ACM SIGMETRICS Performance Evaluation Review. Vol. 35. No. 1. ACM, 2007.*

VERTEX NOMINATION VIA LOCAL NEIGHBORHOOD SEEDED GRAPH MATCHING

Heather G. Patsolic, Vince Lyzinski, Carey E. Priebe

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

The graph matching problem (GMP), finding a map between the vertices of one graph and the vertices of another graph which minimizes the number of edge disagreements between the two graphs, has been of much continuing interest in the past several decades. We are interested in a sub-problem of graph matching in which, given a vertex of interest (VOI) in one network, we seek to identify corresponding vertices in a second network. We propose the use of seeded graph matching on local neighborhoods near the VOI in order to generate a soft nomination list of vertices in the second network that are likely to correspond to the VOI in the first network.

Background - Setting the Stage

The GMP has a wide range of applications in fields such as pattern recognition, machine learning, and object recognition (see [2], [4], [6], and [1] for a few of these applications). In the GMP, the goal is to provide a bijective map across multiple networks that minimizes edge discrepencies using no additional information. As it is plausible that a portion of the bijective map is known, in [3] the authors propose a method of graph matching that utilizes these known correpsondences, called seeds, in order to improve accuracy.

In the work described above, the goal is to match two graphs entirely; however, what should be done if we are interested in only identifying the bijective map for a select few vertices? For example, consider the scenario in which we have a friend whose profile is known in one network and we wish to find which profile belongs to that same friend in a second network. Or perhaps there is a group of neurons in one connectome whose corresponding neurons we would like to locate in a second connectome. In both of these cases, we would expect the structure around the vertices of interest to be similar in both of the corresponding networks. Given a vertex of interest in one graph, we propose a vertex nomination scheme based on local neighborhood seeded graph matching in order to identify the corresponding vertex in a second graph.

Vertex Nomination Via Local Neighborhood Matching

Given a vertex of interest in a network, we seek the corresponding vertex in a second network. We present a principled methodology appropriate for situations in which the networks are too large for brute-force graph matching. Our methodology identifies vertices adjacent to the vertex of interest in the first network that have verifiable corresponding vertices (seeds) in the second network. Leveraging these known correspondences, we match the induced subgraphs in each network generated by the neighborhoods of these verified seeds using a modified version of the seeded graph matching algorithm presented in [3]. We then rank the vertices of the second network in terms of the most likely matches to the original vertex of interest. Letting v^* be the VOI, this ordered list of vertices is referred to as the nomination list for v^*. We demonstrate the applicability of our methodology through simulations and real data examples.

Simulation and Real Data Results

Let $G_i = (V_i, E_i)$ for $i = 1, 2$ be two graphs generated from the same marginal distribution, such that for vertices j and l in V_i, an edge present between j and l in the first graph is correlated with an edge present between nodes j and l in the second graph, and edge presence is otherwise independent across the two graphs. In this section, we demonstrate how our methodology is affected by changes in the number of seeds used and differences in the sizes of the graphs to be matched. After exploring these effects, we demonstrate how the number of seeds affects our algorithm using two network examples.

Simulations: Exploring the Effects of seeds, and differences in graph size

In order to explore how the number of seeds changes the location of the VOI in the nomination list, we vary the number of seeds, s, from 1 to 10, run our algorithm to match 100 pairs of graphs, each a 300 vertex stochastic block model, and record the average location of the VOI in the nomination list, along with a confidence interval

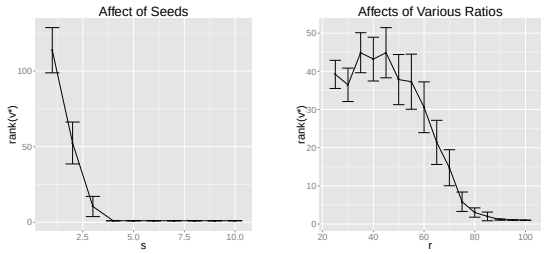


Figure 1: Plot of the average location of the VOI in the nomination list against: the number of seeds used in the matching (left) and the ratio of the size of the smaller graph to the larger (right).

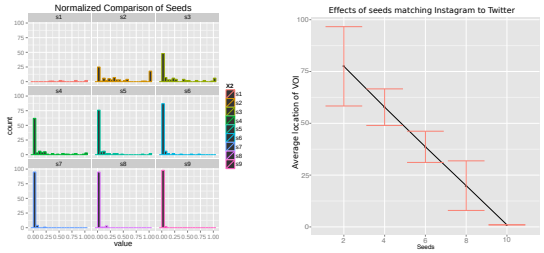


Figure 2: Example of how using seeds lowers location of VOI in nomination list: pair of high school networks (left) and pair of social networks (right).

(mean ± 2*se). As can seen in Figure 1 (left), as the number of seeds increases, the location of the VOI in the nomination list decreases.

To explore how the accuracy of our methodology is influenced by matching graphs which differ in size, we next consider pairs of graphs on different sized vertex sets such that the number of vertices in the smaller graph is r times the number of vertices in the larger graph (300), for $r = 0.25, 0.30, \ldots, 1$. For each r, we plot the average location of the VOI in the nomination list along with a confidence interval, as before, and plot these values in Figure 1 (right). In this figure we can see that when the graphs to match have a large discrepancy between the sizes of their vertex sets there is less accuracy in the algorithm.

Exploring real pairs of networks

We now explore the effect that the number of seeds has on our methodology in two examples. The first involves a pair of high-school friendship networks [5], and the second is a comparison of subnetworks of Twitter and Instagram.

For the first high school network, we choose one of the vertices to be the VOI and apply our methodology using seedsets of size $s \in \{1, \ldots, 9\}$, where all seeds are adjacent to the VOI in the first graph. We create a histograph for each s, shown in Figure 2 (left), displaying the normalized location of the VOI in the nomination list with respect to the size of the second neighborhood. Thus, values of 0, 0.5, and 1 imply that the VOI was first, half-way down, and last in the nomination list, respectively. As can be seen in the plot, as the number of seeds increases, the algorithm improves in performance.

For the Twitter and Instagram networks, we were given 11 correspondences. Letting one vertex be the VOI we obtain the average and confidence interval (as before) for the location of the VOI in the nomination list when using an even size subset of the remaining 10 vertices, see Figure 2 (right).

Concluding Remarks

In all, we provide a methodology which uses seeded graph matching applied to local networks in order to generate a nomination list pertaining to a vertex of interest. We demonstrate the performance of our methodology via simulations and real-data examples.

Acknowledgements

The authors would like to thank XDATA program of the Defense Advanced Research Projects Agency and the Duncan Research Fund.

References

[1] A. C. Berg, T. L. Berg, and M. J. Shape matching and object recognition using low distortion correspondences. *2005 IEEE Conference on Computer Vision and Pattern Recognition*, pages 26–33, 2005.

[2] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(3):265–298, 2004.

[3] D. E. Fishkind, S. Adali, and C. E. Priebe. Seeded graph matching. *arXiv:1209.0367*, 2012.

[4] P. Foggia, G. Perncannella, and M. Vento. Graph matching and learning in pattern recognition in the last 10 years. *Internation Journal of Pattern Recognition and Artificial Intelligence*, 28(1), 2014.

[5] R. Mastrandrea, J. Fournet, and A. Barrat. Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PLoS ONE*, 2015.

[6] J. T. Vogelstein, J. M. Conroy, V. Lyzinski, L. J. Podrazik, S. G. Kratzer, E. T. Harley, D. E. Fishkind, R. J. Vogelstein, and C. E. Priebe. Fast approximate quadratic programming for large (brain) graph matching. *PLoS ONE*, 2015.

EXPLORING THE UTILITY OF NETWORK COMMUNITY STRUCTURE IN THE CONTEXT OF INFLUENCE MAXIMIZATION

Arun Sathanur and Mahantesh Halappanavar (Pacific Northwest National Labs, Richland, WA)

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Introduction

The study of diffusion processes on complex networks has recently been receiving a lot of attention and is of concern to a diverse set of practitioners. A particular important problem that has been extensively investigated is the *Influence Maximization* problem the objective of which is to identify a set of k vertices on the network that when activated result in maximal activation of vertices on the entire network under a given diffusion model. The problem initially posed in the context of viral marketing by Domingos and Richardson [2] was formalized in its present form and solved using a greedy hill-climbing optimizer for the linear-threshold and the independent-cascade diffusion models by Kempe et al. in [3]. Following these two landmark works, several researchers have scrutinized different aspects of the problem that include approaches with algorithm complexity lower than the greedy optimizer and variations in the diffusion models for different scenarios.

Influence Maximization with Community Detection

In this work we examine the impact of network community structure on the influence maximization problem, both from a qualitative perspective as well as from a computational view-point. The literature on utilizing community structure to accelerate the mining of influential nodes in a complex network is comparatively sparse. Notable works on this topic include the works by Wang et al. [5] and Chen et al. [1]. While the former first selects a community that results in maximum marginal influence gain and then attempts influence maximization on that community, the latter selects candidate seeds based on the community features and heuristics followed by a pruning step that finalizes the overall seed set. Our strategies for exploring the benefits of utilizing community structure for the influence maximization problem revolves around four motivating factors namely

- The recognition that the dynamical processes on complex networks are strongly influenced by the modular nature of the underlying graph whereby phenomena such as information flow or random walks tend to have strong components within the communities.
- The need to target each of the communities from the point of view of representing communities of different sizes in certain applications such as in viral marketing
- Accelerate the mining of influential nodes by exploiting community structure without incurring a significant deterioration in the number of activations.
- Leveraging HPC platforms to parallelize the independent Monte Carlo runs thereby resulting in additional speedups

We seek to identify L seed sets $S_1, S_2, \ldots S_L$ such that the seed set S_p correspond to the community C_p. Accordingly, our approach also involves strategies to distribute the overall budget k for the full seed-set size into L smaller budgets corresponding to the L communities. The strategy that we consider in this work involves assigning the seed set sizes proportional to the size of the communities and run the influence maximization algorithms on each of the communities and eventually aggregate the seeds thus obtained. Given the gold-standard nature of the results based on the greedy hill-climbing optimizer, we have chosen the greedy optimizer as a representative algorithm due to its simplicity, and as a test case to prove the utility and scalability of our approach using community detection. More advanced optimization methods can replace the hill-climbing optimizer in our framework for further scalability.

Datasets and experiments

Our dataset comprises three graphs - two based on real-world social networks and one based on a synthetic generative model. Our first graph (LFR-1k) is based on synthetic networks that follow generative LFR model with community structure. Our second graph (PBlogs) represents a real-world network and is based on the political blogosphere during the 2004 U.S. election. The third graph (WikiVotes) is also based on a real-world network in the form of the Wikipedia vote network. The number of vertices range from 1000 to 4160 while the number of edges

Input	**LT-W**	**LT-C**	**IC-W**	**IC-C**
LFR-1k	1.309*e*6	1.289*e*6	7.756*e*5	8.346*e*5
P-Blogs	8.075*e*5	7.854*e*5	4.688*e*5	5.110*e*5
Wiki-Votes	2.315*e*6	2.298*e*6	1.442*e*6	2.321*e*6

Table 1: Cumulative number of activations across all samples (3200) for each algorithm is computed using the four algorithms detailed.

range from 11,433 to 100,132. We leveraged the parallel community detection tool *Grappolo* that is based on a parallel implementation of the Louvain algorithm [4]. For the two real-world graphs we filtered out isolated vertices and edges including degree-1 nodes that do not contribute to influence spread and that degrade the quality of the communities detected. Figure 1 presents a visualization of the three graphs with their communities.

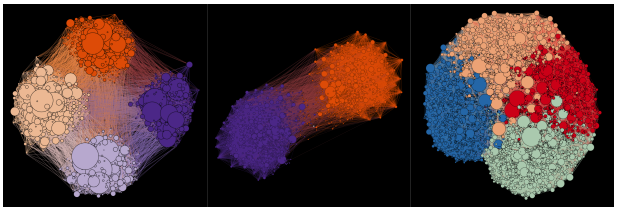


Figure 1: From L to R : LFR-1k, PBlogs and the WikiVotes graphss. Colors indicate communities and the relative sizes of the nodes, their out-degree values.

The abbreviations for different algorithms / models that we considered are as follows. LT-W denotes the linear threshold (LT) model on the entire graph while LT-C denotes the LT model with communities. Similarly IC-W denotes the independent cascade (IC) on the entire graph whereas IC-C denotes the IC model with communities. We assessed the quality of influence spread obtained by our community-based influence maximization implementation by inputting the overall seed-set from the communities and then running the diffusion model on the entire graph. In a second experiment we computed the cumulative number of activated vertices by running the influence maximization algorithms on the entire graph. The results are tabulated in table 1 for all the algorithms considered. These results show very little degradation in the net activations between the LT and LT-C models while the net activations improved for the IC-C model when compared to the IC-W model.

Next, we present the speedups obtained by our workflow involving influence maximization via community detection. The results are tabulated below in table 2. The speedups range from **3**× to **28**× for the three graphs in our dataset

Input	LT-W/LT-C	**SpUp**	IC-W /IC-C	**SpUp**
LFR-1k	6203/1066	**5.82**	65265/2266	**28.78**
P-Blogs	7867/2582	**3.05**	40979/1921	**3.44**
Wiki-Votes	79600/6453	**12.33**	250387/18668	**13.41**

Table 2: Runtimes (in seconds) to select 100 seeds using 3200 samples and 40 threads.

and for the different diffusion models. Specifically we note that the graphs considered in this work are moderate in size with a small number of communities. Complexity analyses of the algorithms for the diffusion models reveal that higher speedups are possible for larger graphs that contain a correspondingly larger number of communities. In addition to the speedups obtained from the utilization of the community structure, scalability results from the parallelization of the random samples in the influence maximization workflows show up to 6.3X speedup (for the PBlogs graph) on 20 cores relative to the baseline run on 2 cores.

Conclusions and Future Work

In this work we presented the idea of accelerating the mining of influential nodes in a complex network by leveraging community detection as a pre processing step. The simple approach that we adopted is shown to provide significant speedups in the computation of the influential seeds and the resulting influence spread does not deteriorate significantly when compared to the influence maximization on the entire graphs. In fact as reported, the number of activations improves for some of the cases. We will be working on providing a more rigorous theoretical justification for our approach. In terms of scaling the approach to work for human-scale graphs, we will be exploring more advanced optimization strategies in conjunction with hierarchical versions of our approach that leverage multi-level community detection.

References

[1] Y.-C. Chen et al. CIM: community-based influence maximization in social networks. *ACM Transactions on Intelligent Systems and Technology*, 5(2):25, 2014.

[2] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD*, pages 57–66. ACM, 2001.

[3] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *ACM KDD*, pages 137–146, New York, NY, USA, 2003. ACM.

[4] H. Lu, M. Halappanavar, and A. Kalyanaraman. Parallel heuristics for scalable community detection. *Parallel Computing*, 47:19 – 37, 2015.

[5] Y. Wang, G. Cong, G. Song, and K. Xie. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *ACM KDD*, pages 1039–1048. ACM, 2010.

WARPING THE URBAN SPACE: THE EFFECT OF FAST SUBWAY ON STREET NETWORKS

Saray Shai, Dane R. Taylor, Peter J. Mucha

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

Street networks – acting as the urban substrate for social and economical developments – control many aspects of our society, from disease spread to urban sprawl and population growth. As cities and their transportation systems become increasingly complex and multimodal, it is important to understand the implications of such advances to the topology and geometry of street networks. Using tools from topological data analysis, we study how the introduction of fast subway systems affect the embedding space of the multilayer transportation network.

Introduction

Recent developments in network science together with advances in technology and data collection have opened the path for studies on the structure and function of multilayer transportation networks. Just like in other multilayer networks, the interactions and dependencies between different transportation modes have been shown to have a profound effect on their behavior, from centralities and congestions [5] to resilience and navigability [3]. Here we propose to apply tools from topological data analysis to analyze multilayer transportation networks. In particular, network embedding is an exciting research direction with a wide range of applications: revealing the underlying geometry of a network provides us with new lens to examine their structural (e.g. heterogeneous degree distributions and strong clustering [4]) and functional (e.g. spreading processes [6]) properties. While street networks alone can be well approximated by a two dimensional Euclidian space, it is no longer the case when adding shortcuts such as fast subway lines. In this case, we seek to determine the appropriate embedding space, whether it is a higher dimensional Euclidian space or a different geometry altogether (e.g. hyperbolic).

Additional details

Let $G_{\text{street}} = (V_{\text{street}}, E_{\text{street}}, w_{\text{street}})$ be a weighted graph of a connected street network in its "primal" representation, with nodes being street junctions and edges representing the street segments connecting them weighted by the street length, $w_{\text{street}}(e \in E_{\text{street}}) = l(e)$. Let $G_{\text{subway}} = (V_{\text{subway}}, E_{\text{subway}}, w^{\beta}_{\text{subway}})$ be a weighted graph of a connected subway network with nodes representing subway stations and links connecting successive stations on the same line. To account for the rapidity of subway networks compared with roads, we associate subway links with a parameter $0 < \beta \leq 1$ such that the weight of en edge in G_{subway} is the length of the line segment multiplied by β, $w^{\beta}_{\text{subway}}(e \in E_{\text{subway}}) = \beta\, l(e)$, i.e. subway links are considered to be $\frac{1}{\beta}$ times *faster* than street links [5].

Finally, the multilayer network is defined as the union of these two networks, $G_{\text{multi}} = (V_{\text{street}} \cup V_{\text{subway}}, E_{\text{street}} \cup E_{\text{subway}} \cup E_{\text{inter}}, w^{\beta}_{\text{multi}})$ together with the addition of inter-network edges E_{inter}, connecting each subway station with its closest street junction (obviously, subway stations are accessible from more than one point on the street, but this simplification will not change the bulk structure of shortest paths [5]). The weight of these additional edges is simply the Euclidian distance between the corresponding street junction and subway station:

$$w^{\beta}_{\text{multi}}(e) = \begin{cases} w_{\text{street}}(e) = l(e), & \text{if } i,j \in V_{\text{street}} \\ w^{\beta}_{\text{subway}}(e) = \beta\, l(e), & \text{if } i,j \in V_{\text{subway}} \\ d(i,j) & \text{otherwise} \end{cases}$$

where $d(i,j)$ is the Euclidian distance between i and j.

Results

As a first step in seeking an appropriate embedding space of a transportation network as described above, we examine the embedding of nodes in a two dimensional Euclidian space, see Fig. 1. The mapping of street nodes to points in a two dimensional space, obtained using Isomap [2], are based on weighted shortest paths in the corresponding graph. In Fig. 1(a) we consider shortest paths in the street network, G_{street}, without the subway shortcuts: the obtained embedding bears a strong resemblance with the original map shown in Fig. 1(d). This is somewhat expected since indeed weighted shortest path in the street

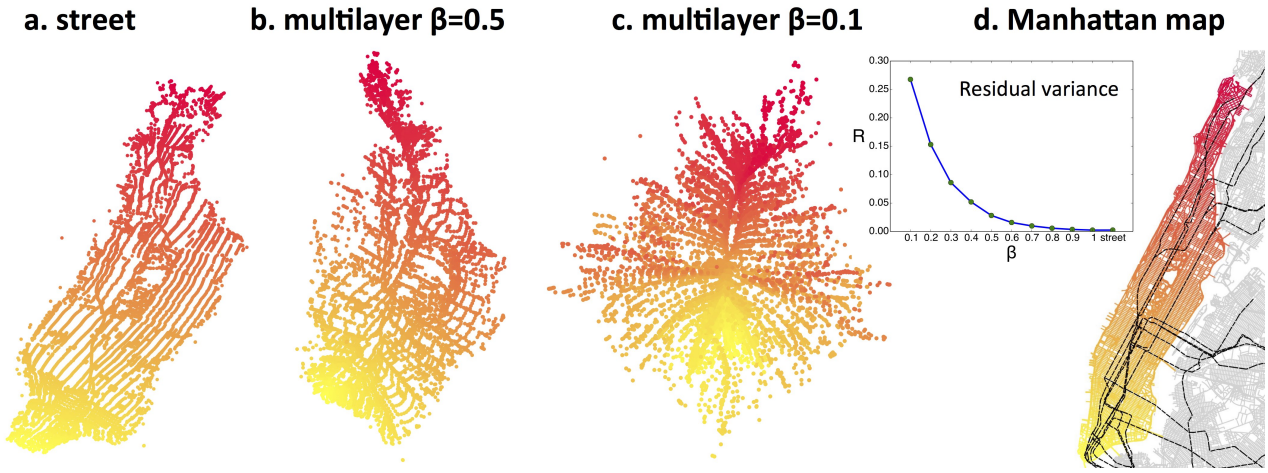


Figure 1: **2D embedding of the street network** a-c) Two dimensional embedding of the street nodes based on weighted shortest path distances in the street (a) and multiplex (b)-(c) networks. Edge colors match the map (d) where edges (street segments) are colored according to their distance from the top (most north) point. Subway links are shown in black. Inset: Residual variance of the embedding, R, as a function of β.

network can be well approximated by the Euclidean distance between two street junctions [1]. However, when introducing fast subway links, the obtained embedding is very different, see Fig. 1(b)-(c). Qualitatively, introducing a fast subway is "pinching" the urban space, leading to a more even distribution of accessibility, but also altering the spatial distribution of betweenness centrality and consequently congestions [5]. More importantly, the resulting embedded space is no longer well approximated by a two dimensional Euclidean space. This can be quantified by calculating the residual variance $R = 1 - \rho^2$, where ρ the Pearson correlation coefficient between the mapped distances to the shortest path network distances [2]. We observe that R is non-linearly increasing with a decreasing β, see inset in Fig. 1. At $\beta = 0.5$, which is close to the NYC empirical value [5] (obtained by dividing the average speed of subway trips, 17.4 mph, by the average speed of taxi trips over weekdays, 9.7 mph), we obtain $R = 0.028$, an order of magnitude larger than $R = 0.002$ obtained for the street network alone.

Discussion

We have studied the effect of introducing fast subway links to an otherwise two dimensional planar street network. Those additional links constitute shortcuts between distant parts of the city, thus significantly changing the mapping of street nodes to a low-dimensional space. We have shown that the two dimensional Euclidean space is no longer well approximating shortest paths on the multilayer network. In other words, the network is being warped into a higher dimension, and we seek a mapping that better describes the network and provide us with information about the functionality of the urban space. Finding such a mapping is an important problem with an extensive variety of applications in urban science and civil and environmental engineering.

References

[1] M. Barthélemy. Spatial networks. *Phys. Rep*, 499:1 – 101, 2011.

[2] T. F. Cox and M. Cox. *Multidimensional Scaling*. Chapman and Hall/CRC, 2nd edition, 1994.

[3] M. De Domenico, A. Solé-Ribalta, S. Gómez, and A. Arenas. Navigability of interconnected networks under random failures. *Proc. Natl. Acad. Sci. USA*, 111(23):8351–8356, 2014.

[4] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguñá. Hyperbolic geometry of complex networks. *Phys. Rev. E*, 82(3):036106, 2010.

[5] E. Strano, S. Shai, S. Dobson, and M. Barthelemy. Multiplex networks in metropolitan areas: generic features and local effects. *J. R. Soc. Interface*, 12(111), 2015.

[6] D. Taylor, F. Klimm, H. A. Harrington, M. Kramar, K. Mischaikow, M. A. Porter, and P. J. Mucha. Topological data analysis of contagion maps for examining spreading processes on networks. *Nat Commun*, 6, 2015.

A GENERALIZED CLUSTERING COEFFICIENT BASED ON P-MODULUS OF LOOPS

Heman Shakeri

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Introduction

Complex networks exhibit properties such as small-world phenomenon [13], scale-free degree distribution [3], and local clustering of nodes [13]. However, this clustering tendency is difficult to quantify. A proposed measure of clustering for node v [13] is to compute the fraction of edges between neighbors of v that actually are in the network, over all possible ones. The clustering coefficient measures the tendency of nodes to cluster and create close-knit groups and represents the probability of the presence of a link in the network for random graphs [7].

The authors in [4] pointed out the importance of closed paths (loops) in the cluster and discussed computation of the clustering coefficient using the fraction of the number of loops with length 3 (triangles) that include node v, over all possible triangles that might include v. Because this measure fails to describe the clustering of grid-like parts of networks, the authors improved the measure by counting loops with length 4 (quadrilaterals) and proposed a new measure that considers different types of quadrilaterals. Similarly [6] adresses bipartite networks, that do not contain triangles and thus for which the standard clustering coefficient is not useful. They also emphasize the importance of longer loops in the network. Finally, [5] proposed another similar clustering coefficient for bipartite networks.

The authors in [12], showed that clustering coefficient measures are highly correlated with degree, and they proposed a measure that preserves the degree sequence for the maximum possible links among neighbors of node v, thus avoiding correlation biases. Although none of these methods considered weighted and directed networks, [10] introduced a version of clustering coefficient that considers weighted network, and [8] introduced a way to measure a general clustering coefficient for weighted and directed networks.

These shortcomings of previous methods and numerous versions of the proposed clustering coefficients expose the need for a generalized measure that works for a wide range of applications. We apply the concept of modulus of families of loops as a tool to study structural properties of network clustering. Research in [2] showed that modulus is a convex optimization problem that can be solved effectively. Pietro Poggi-Corradini and Nathan Albin, from the NODE[1] research group (`https://node.math.ksu.edu/`), will present the theoretical aspects of these notions and the available efficient numerical algorithms to compute them.

In this work we explore the versatility of modulus of families of loops and show that it provides a deeper approach to the study of network clustering properties. We also propose a new clustering coefficient that can explain nontrivial situations that conventional methods cannot handle. Moreover, we show that preprocessing the loops in the network can improve spectral clustering to partition disjoint and overlapped communities.

Analyzing richness of loops in a network with p-modulus

Let $\mathcal{G} = (V, E)$ be a network with nodes V and links E. Using standard terminology, a *loop* γ on a network is represented by a finite string of nodes $v_1v_2v_3 \dots v_rv_1$, such that v_i and v_{i+1} are linked with an edge and the v_i's are all distinct. We call Γ the family of all loops. We define the *ρ-length* of loop γ as

$$\ell_\rho(\gamma) := \sum_{e\in\gamma} \rho(e) \tag{1}$$

where $\rho : E \to [0, \infty)$ is a density, interpreted as a penalty or cost the walker must pay for traversing link e. When $\rho_0(e) \equiv 1$, ℓ_{ρ_0} represents the hop-length of γ. We define the ρ-length of Γ as $\ell_\rho(\Gamma) = \inf_{\gamma\in\Gamma} \ell_\rho(\gamma)$. A density ρ is admissible for a family of loops Γ if $\ell_\rho(\Gamma) \geq 1$. Let $A(\Gamma)$ be the set of all admissible densities for Γ. Let $w : E \to (0, \infty)$ be a positive weight function. Then, for $1 < p < \infty$, $\mathrm{Mod}_{p,w}(\Gamma)$ is defined as

$$\mathrm{Mod}_{p,w}(\Gamma) = \inf_{\rho\in A(\Gamma)} \mathcal{E}_{p,w}(\rho) = \mathcal{E}_{p,w}(\rho^*), \tag{2}$$

where $\mathcal{E}_{p,w}(\rho) = \sum_{e\in E} w(e)\,|\rho(e)|^p$ is the energy of the density ρ and ρ^* is the unique minimizer [1, Lemma 2.1].

[1]NODE is supported by NSF grant n. 1515810

For example, if $\mathcal{G}$ is a tree, $\mathrm{Mod}_p(\Gamma) = 0$; if $\mathcal{G}$ is a complete graph, $\mathrm{Mod}_p(\Gamma) = \frac{1}{3^p}\binom{n}{2}$. We consider $p = 2$ for the rest of this discussion due to its physical interpretations and computational costs. In [11] we showed that 2-modulus has properties that allow quantification of the richness of various family of walks (e.g., loops).

Clustering coefficient with modulus of family of loops

A node v has a high clustering coefficient when many short loops pass through v and its close-by nodes. The standard method of counting triangles considers the smallest loops, while other methods considers the next shortest loops, quadrilaterals. A method must be devised to compare these loops and evaluate the combined influence to improve clustering coefficient measures. The previous section introduced a way to evaluate a family of loops using the modulus of the family. We propose a comprehensive measure of clustering. We define our clustering coefficient by

$$c_v^L = \frac{\mathrm{Mod}_2(\Gamma_{g_v^h})}{\mathrm{Mod}_2(\Gamma_{K_v})},$$

where g_v^h is the subgraph that contains all loops with length$\leq h$ rooted to v and K_v is the complete graph with the same number of nodes as g_v^h and For the global clustering coefficient we can either average the above coefficient over all the nodes or find $c^L = \frac{\mathrm{Mod}_2(\Gamma_{\mathcal{G}^h})}{\mathrm{Mod}_2(\Gamma_{K_n})}$.

A simple example of the proposed clustering coefficient is presented in Figure 1. As one can see the standard method on the right is unable to capture loops with length 4, but our proposed measure on the left indicates that $c_1^L = c_2^L = 0.75$ and $c_5 = 0.56$ while in the standard version $c_5 = 1$ and $c_1 = c_2 = 0.67$. Additional examples for bipartite, directed, and weighted networks will be presented.

Loop analysis of the network and improvement of the partitioning algorithms

Analyzing loops in a graph provides information about the cluster structure and emphasizes the importance of edges in these clusters. After we computed the modulus of loops in a network, the extremal density $\rho^*(e)$ gives extra information about the structure of partitions that contains many short loops and the importance of edges in these clusters. We can substantially improve the performance of some partitioning methods such as spectral partitioning by preprocessing the network into a weighted network with edge weights $\rho^*(e)$'s.

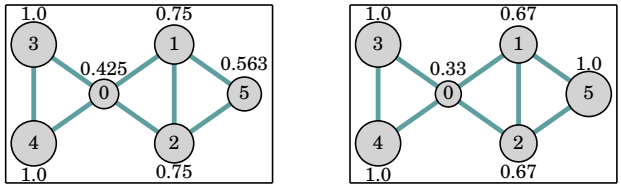


Figure 1: Clustering coefficient computed for each node in the network by the proposed Loop Modulus method (left) and the Standard method (right).

References

[1] N. Albin, M. Brunner, R. Perez, P. Poggi-Corradini, and N. Wiens. Modulus on graphs as a generalization of standard graph theoretic quantities. http://arxiv.org/abs/1504.02418.

[2] N. Albin, F. Sahneh, M. Goering, and P. Poggi-Corradini. Modulus of families of walks on graphs. *arXiv:1401.7640*, 2014. http://arxiv.org/abs/1401.7640.

[3] A. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[4] G. Caldarelli, R. Pastor-Satorras, and A. Vespignani. Structure of cycles and local ordering in complex networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):183–186, 2004.

[5] M. Latapy, C. Magnien, and N. Del Vecchio. Basic notions for the analysis of large two-mode networks. *Social networks*, 30(1):31–48, 2008.

[6] P. G. Lind, M. C. González, and H. J. Herrmann. Cycles and clustering in bipartite networks. *Physical review E*, 72(5):056127, 2005.

[7] M. E. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[8] T. Opsahl and P. Panzarasa. Clustering in weighted networks. *Social networks*, 31(2):155–163, 2009.

[9] K. Paton. An algorithm for finding a fundamental set of cycles of a graph. *Communications of the ACM*, 12(9):514–518, 1969.

[10] J. Saramäki, M. Kivelä, J.-P. Onnela, K. Kaski, and J. Kertesz. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E*, 75(2):027105, 2007.

[11] H. Shakeri, P. Poggi-Corradini, C. Scoglio, and N. Albin. Generalized network measures based on modulus of families of walks. *Journal of Computational and Applied Mathematics*, 2016.

[12] S. N. Soffer and A. Vazquez. Network clustering coefficient without degree-correlation biases. *Physical Review E*, 71(5):057101, 2005.

[13] D. J. Watts and S. H. Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):440–442, 1998.

EXPOSING MESOSCALE CONNECTIVITY PATTERNS IN THE STRUCTURAL BRAIN NETWORK.

Ann E. Sizemore, Chad Giusti, Matthew Cieslak, Scott Grafton, Danielle S. Bassett

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

Encoding the axon bundles between brain regions as a complex network has provided novel insights into brain function and disease. Standard network tools describe local or aggregate global phenomena, however many neural functions occur at the mesoscale. Here we employ a recently developed method from algebraic topology using regions involved in all-to-all connected subgraphs to illuminate closed circuit connection patterns that consistently exist across eight human subjects, and highlight the potential of this topological view of the brain network.

Additional Detail

Networks Studied

We first translate diffusion spectrum imaging (DSI) data from eight healthy individuals in triplicate into undirected, weighted networks [1]. For both the average network across subjects and individual networks, we compute topological statistics and discern mesoscale connectivity patterns. We compare these results with our model, a minimally wired graph created using coordinates of brain regions as nodes [3] and edges with weights inversely proportional to distance between nodes.

Topological Calculations

Given G, a graph with vertex set V, define a k-clique as a set of k nodes where all pairwise connections exist. Cliques exist in the brain network as sets of completely connected brain regions (Fig. 1a). As any subset of nodes in a clique must also form a clique, called a *face*, and so we enumerate only *maximal* cliques, or cliques that are not faces of any other (Fig. 1b). The maximal clique distribution can then be compared to the network created from minimally wiring region coordinates from individuals (Fig. 1c). As nodes may be involved in varying numbers and degrees of cliques, Fig. 1d displays the distribution of node participation in ranges of clique dimensions which correspond to regions shaded in Fig. 1c.

The number of cliques containing a node of interest carries information about the connectivity of the node. Indeed, we see correlation between node participation and node strength, communicability [2], and k-core and s-core decompositions [4] as expected.

In order to detect closed circuits in the weighted DSI network, we first detect structure in a binary graph, and repeat this process for every weight threshold. Given a binary graph created from one threshold level, *homology* allows us to roughly find structural cavities enclosed by cliques arranged in specific patterns called *cycles*. Applying this notion to weighted networks, *persistent homology* describes how cycles in one threshold map to cycles in the next (Fig. 2a). Such a mapping provides the edge density (ρ) at which a cycle first exists (Fig. 2a; ρ_{birth}) and when a cycle is completely triangulated by cliques and thus no longer encloses a cavity (Fig. 2a; ρ_{death}).

Essential Cycle Examples

Cycles essential to the network architecture are generally considered to be those which evade triangulation for the longest ranges of threshold levels. We show two examples of such cycles in Fig. 2b. The left, green cycle encloses a two-dimensional cavity which exists in every scan of all eight individuals. The right, purple cycle composed of seven nodes created from 3-cliques is found in at least one scan in six out of eight individuals. A permutation test using the minimally wired model suggests cycle existence is unlikely ($p< 0.03$ for green cycle, $p<0.001$ for purple cycle within each hemisphere).

References

[1] M. Cieslak and S. Grafton. Local termination pattern analysis: A tool for comparing white matter morphology. *Brain imaging and behavior*, 8(2):292–299, 2014.

[2] E. Estrada and N. Hatano. Communicability in complex networks. *Physical Review E*, 77(3):036111, 2008.

[3] F. Klimm, D. S. Bassett, J. M. Carlson, and P. J. Mucha. Resolving structural variability in network models and the brain. *PLOS Comput Biol*, 10(3):e1003491, 2014.

[4] M. P. van den Heuvel and O. Sporns. Rich-club organization of the human connectome. *The Journal of neuroscience*, 31(44):15775–15786, 2011.

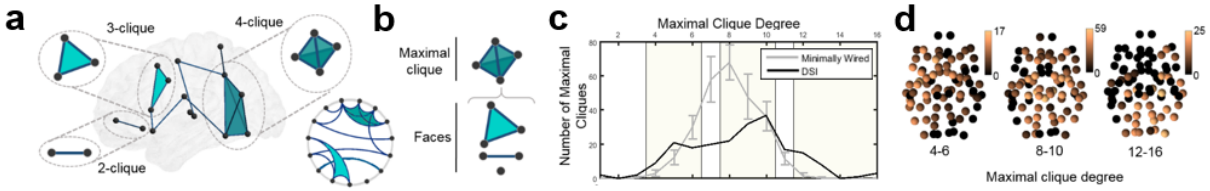


Figure 1: Clique mapping on the brain network indicates regions of complete connectivity. *(a)* Cliques of degree 2, 3, and 4 shown in the brain network, highlighted in the circle plot, and as familiar representations. *(b)* Example of a maximal 4-clique and a list of its faces. *(c)* Distribution of maximal cliques in the DSI and minimally wired networks. *(d)* Locations of the brain connect in differing ranges of clique degrees. Ranges shown correspond to shaded regions in panel *(c)*.

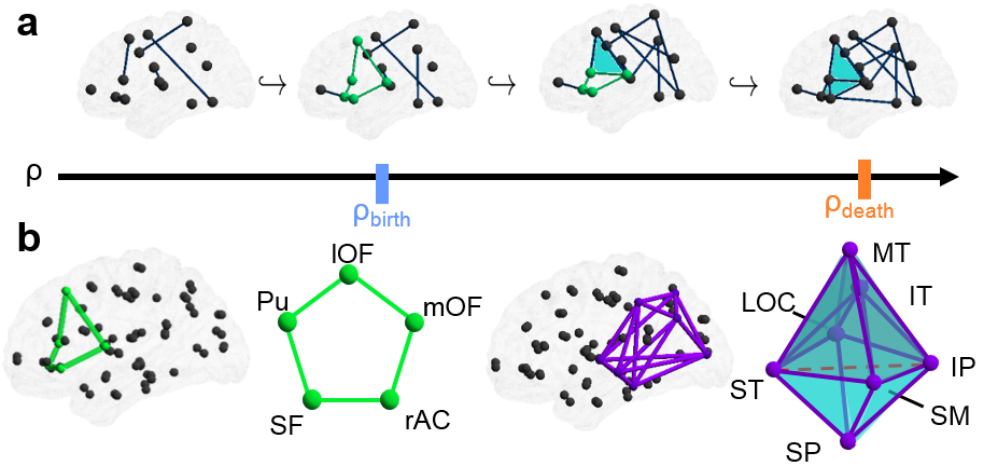


Figure 2: Topological methods detect cycles in the brain network. *(a)* Example network at four threshold levels with increasing edge density (ρ). Green cycle forms at $\rho = \rho_{birth}$, shrinks in length in the next threshold shown, and is completely triangulated at $\rho = \rho_{death}$. *(b)* Cycles found using persistent homology. The green cycle contains the lateral orbitofrontal (lOF), medial orbitofrontal (mOF), rostral anterior cingulate (rAC), suprior frontal (SF), and putamen (Pu) regions. The purple cycle includes the medial temporal (MT), inferior temporal (IT), inferior parietal (IP), supramarginal (SM), superior parietal (SP), superior temporal (ST), and lateral occipital regions (LOC). Each cycle shown in the brain and as a schematic for a cycle of 2-cliques (left, green) and one of 3-cliques (right, purple).

EVALUATING THE TOPOLOGICAL ROBUSTNESS OF POWER GRIDS TO LINE FAILURES

Saleh Soltan, Gil Zussman
Electrical Engineering, Columbia University, New York, NY
{*saleh,gil*}*@ee.columbia.edu*

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We use the *mutual edge flow change ratios* (the ratio between the change of flow on an edge, and the initial flow on the failed edge) to evaluate the topological robustness of power grids to line failures. In particular, we show that mutual edge flow change ratios are independent of the power supply/demand distribution and solely depend on the grid structure. Then, we define and analytically compute the *failure cost of an edge* and the *average edge failure cost in a graph*, and demonstrate that the results can be used to study the robustness of power grids to a single line failure.

Model

We adopt the linearized (or DC) power flow model, which is widely used as an approximation for the AC power flow model [1,4]. We represent the power grid by an undirected graph $G = (V, E)$ where V and E correspond to the buses and transmission lines, respectively. p_v is the active power *supply* ($p_v > 0$) or *demand* ($p_v < 0$) at node $v \in V$ (for a *neutral node* $p_v = 0$). We assume *pure reactive* lines, where each edge $\{u, v\}$ is characterized by its *reactance* $x_{uv} = x_{vu}$. A *power flow* is a solution (f, θ) of:

$$\sum_{v \in N(u)} f_{uv} = p_u, \ \forall\ u \in V \tag{1}$$

$$\theta_u - \theta_v - x_{uv} f_{uv} = 0, \ \forall\ \{u, v\} \in E \tag{2}$$

where $N(u)$ is the set of neighbors of node u, f_{uv} is the power flow from node u to node v, and θ_u is the phase angle of node u. Eq.(1)-(2) are equivalent to the matrix equation: $A\Theta = P$, where $\Theta \in \mathbb{R}^{|V|\times 1}$ is the vector of phase angles, $P \in \mathbb{R}^{|V|\times 1}$ is the power supply/demand vector, and $A = [a_{ij}] \in \mathbb{R}^{|V|\times|V|}$ is the *admittance matrix* of the graph G. The power flow equations can be solved by using the *Moore-Penrose Pseudo-inverse* of the admittance matrix, $A^+ = [a^+_{ij}]$ [2].

To study the effects of a *single edge (e') failure*, we define the ratio between the change of flow on an edge, e, and the initial flow on the failed edge, e', as *mutual edge flow change ratio*: $M_{e,e'} = |\Delta f_e / f_{e'}|$. The mutual edge flow change ratio corresponds to the Line Outage Distribution Factor (LODF) defined in [4, P. 307].

Failure Impact

The following theorem provides an analytical rank-1 update of the pseudo-inverse of the admittance matrix.

Theorem 1. *If $\{i, j\}$ is not a cut-edge, then,*

$$A'^{+} = (A + a_{ij} X X^t)^+ = A^+ - \frac{1}{a_{ij}^{-1} + X^t A^+ X} A^+ X X^t A^+$$

in which X is an $n \times 1$ vector with 1 in i^{th} entry, -1 in j^{th} entry, and 0 elsewhere.

Corollary 1. *The flow on an edge $\{r, s\}$ after a failure in the non-cut-edge $\{i, j\}$ is,*

$$f'_{rs} = f_{rs} - \frac{a_{rs}}{a_{ij}} \frac{(a^+_{ri} - a^+_{rj}) - (a^+_{si} - a^+_{sj})}{a_{ij}^{-1} - 2(a^+)_{ij} + (a^+)_{ii} + (a^+)_{jj}} f_{ij}.$$

To focus solely on topological robustness, in this abstract we assume that $x_{uv} = 1\ \forall \{u, v\} \in E$. In this case, the admittance matrix A is the *Laplacian matrix* of the graph and using Corollary 1 the mutual edge flow change ratios can computed as follows.

Lemma 1. *The mutual edge flow change ratio for an edge $e = \{r, s\} \in E$ after a failure in a non-cut-edge $e' = \{i, j\} \in E$ is,*

$$M_{e,e'} = \left| \frac{(a^+_{ri} - a^+_{rj}) - (a^+_{si} - a^+_{sj})}{-1 - 2(a^+)_{ij} + (a^+)_{ii} + (a^+)_{jj}} \right|.$$

The Lemma implies that the mutual edge flow change ratios are independent of the power supply/demand distribution and solely depend on the grid structure.

This abstract summarizes some of the results that appear in [2]. This work was supported in part by DTRA grant HDTRA1-13-1-0021, CIAN NSF ERC under grant EEC-0812072, and the People Programme (Marie Curie Actions) of the European Unions Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no. [PIIF-GA-2013-629740].11.

Network Robustness

Definition. *The* failure cost *of an edge e in G is denoted by FC_e and defined as follows:* $FC_e := \frac{1}{m-1}\sum_{\substack{e'\in E\\ e'\neq e}}(M_{e',e})^2$.

The failure cost of an edge e is a good measure of the average changes that occur in the flows of the other edges as a result of the failure in an edge e. Determining the costs can help constructing a reliable power grid in two ways: (i) by designing networks with a minimum maximum failure cost, and (ii) by setting the power supply and demand values such that edges with high failure costs carry small flows. The following Lemma analytically shows the relation between the failure cost of a non-cut-edge and the *resistance distance* between its end nodes. The resistance distance between two nodes $i, j \in V$ is $r(i,j) := a^+_{ii} + a^+_{jj} - 2a^+_{ij}$.

Lemma 2. *In a connected graph G, for any non-cut-edge $e = \{i,j\}$,*

$$FC_e = \frac{1}{m-1}\frac{r(i,j)}{1-r(i,j)}. \tag{3}$$

Eq. (3) is very insightful. Intuitively, it demonstrates that failures in edges with high resistance distance values have a strong effect on the other edges. Moreover, (3) allows to obtain a bound on the average edge failure cost, which is defined below as a metric for the robustness of a graph to a single edge failure.

Definition. *In a graph G with n nodes and m edges, the* average edge failure cost *is defined as,* $\overline{FC}_G := \frac{1}{m}\sum_{e\in E} FC_e$.

Using (3), the following Lemma provides a lower bound on the average edge failure cost in a graph.

Lemma 3. *In a 2-edge-connected graph G,*

$$\frac{1}{m}\Big(\frac{m-1}{n-1} - \frac{m-1}{m}\Big)^{-1} \leq \overline{FC}_G, \tag{4}$$

and equality holds, if for any two edges $e = \{i,j\}$ and $e' = \{p,q\}$, $r(i,j) = r(p,q)$.

Corollary 2. *In a symmetric graph G, $\overline{FC}_G = (\frac{m^2-m}{n-1} - (m-1))^{-1}$. Moreover, for any graph H with the same number of nodes and edges as G, $\overline{FC}_H \geq \overline{FC}_G$.*

Corollary 2 demonstrates that symmetric graphs have the lowest average edge failure cost among all the graphs with the same number of nodes and edges. Moreover, from Lemma 3 and Corollary 2 it can be concluded that as graphs become more symmetrical, their average edge failure cost ($\overline{FC}_G$) decreases. To demonstrate this numerically, Fig. 1 shows the average edge failure cost of the graph ($\overline{FC}_G$) and the maximum mutual edge flow change ratio ($\max_{e,e'\in E} M_{e,e'}$) versus the probability of rewiring (p) in Watts and Strogatz graphs [3] with 30 nodes and 60 edges. Initially ($p = 0$), G is a 4-regular graph (namely, every node is connected to exactly 4 other nodes). However, as p increases, G tends toward a random graph with no symmetry. Thus, an increase in p in the Watts and Strogatz graph can be considered as decrease in the symmetry of the graph. As expected, the figure shows that as p increases, both the average edge failure cost of the graph ($\overline{FC}_G$) and the maximum mutual edge flow change ratio ($\max_{e,e'\in E} M_{e,e'}$) increase.

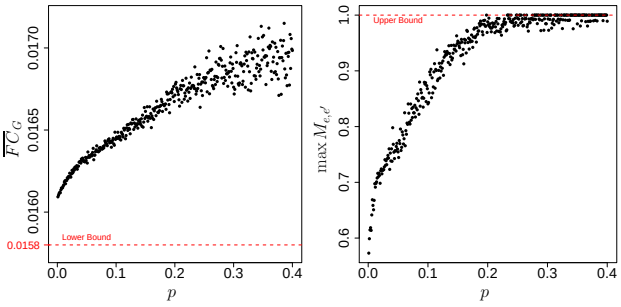


Figure 1: The average edge failure cost of the graph ($\overline{FC}_G$) and the maximum mutual edge flow change ratio ($\max_{e,e'\in E} M_{e,e'}$) versus the probability of rewiring (p) in a Watts and Strogatz graph with 30 nodes and 60 edges. Each point is the average over 100 generated graphs with the same parameters.

Overall, the results suggest that as graphs become more symmetrical, they become more robust against single edge failures.

References

[1] D. Bienstock and A. Verma. The $N-k$ problem in power grids: New models, formulations, and numerical experiments. *SIAM J. Optimiz.*, 20(5):2352–2380, 2010.

[2] S. Soltan, D. Mazauric, and G. Zussman. Analysis of failures in power grids. *IEEE Trans. Control Netw. Syst. (to appear).*

[3] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, 1998.

[4] A. J. Wood and B. F. Wollenberg. *Power generation, operation, and control.* John Wiley & Sons, 3rd edition, 2012.

INFORMATION-THEORETIC REVERSE ENGINEERING OF BIOLOGICAL NETWORKS

Jie Sun, Erik M. Bollt

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

Understanding the dynamics and functioning of biological systems is one of the most challenging tasks faced in modern science. Our goal is to develop a principled information-theoretical approach to infer the causal network structure underlying a biological system from data.

Basic Concepts from Information Theory

We start by reviewing some basic concepts from information theory [5, 2]. The Shannon *entropy* of a discrete random variable X is given by

$$H(X) = -\sum_x p(x)\log p(x), \tag{1}$$

where $p(x) = \text{Prob}(X = x)$. The *joint entropy* between two variables X and Y is defined as

$$H(X,Y) = -\sum_{x,y} p(x,y)\log p(x,y), \tag{2}$$

where $p(x,y) = \text{Prob}(X = x, Y = y)$ is the joint probability. The *conditional entropy* of X given Y is

$$H(X|Y) = -\sum_{x,y} p(x,y)\log p(x|y), \tag{3}$$

where $p(x|y) = \text{Prob}(X = x|Y = y)$ is the conditional probability. The *mutual information* between two variables X and Y is

$$I(X;Y) = H(X) - H(X|Y), \tag{4}$$

and can also be written as $I(X;Y) = H(X) + H(Y) - H(X,Y)$. Mutual information is nonnegative and symmetric: $I(X;Y) = I(Y;X)$. Finally, the *conditional mutual information* between X and Y given Z is

$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z). \tag{5}$$

Conditional mutual information is also generally nonnegative, and symmetric with respect to X and Y.

Causation Entropy

Consider a stationary multivariate stochastic process $\{X_t^i\}$, with $i = 1, 2, \ldots, n$. For the simplicity of discussion, here we assume that the process is Markov order one (see Ref. [8] for treatment of higher-order Markov processes). Denote the collective of all variables at time t as: $\vec{X}_t$, then the Markov condition implies that $p(x_t^i|\vec{x}_{t-1}, \vec{x}_{t-2}, \ldots) = p(x_t^i|\vec{x}_{t-1})$. It is possible that this conditional probability can be further reduced to a minimal subset of *causal* components, $\mathcal{N}_i \subset \{1, 2, \ldots, n\}$, such that

$$p(x_t^i|\vec{x}_{t-1}) = p(x_t^i|x_{t-1}^{\mathcal{N}_i}). \tag{6}$$

To identify the set of causal components $\mathcal{N}_i$ for a given node i in the network, we introduce a quantity called *causation entropy* as a type of (time-shifted) conditional mutual information [6, 8]. In particular, the causation entropy from the set of nodes J to node i conditioned on the set of nodes K is defined as

$$C_{J\to i|K} = I(X_{t-1}^J; X_t^i|X_{t-1}^K). \tag{7}$$

In Ref. [6, 8], we showed that the causation entropy as a generalization of transfer entropy can be effectively used as a measure for causal influence in a multivariate setting. The key is to achieve an appropriate conditioning using a suitable algorithmic approach, as we will discuss next.

Inferring Networks via Optimal Causation Entropy

To uncover the set of causal components $\mathcal{N}_i$ of node i in a network, we devised an iterative optimization scheme that contains two stages [7, 8]. Starting with $K = \varnothing$. In the "forward" stage, in each iteration the node that maximizes the causation entropy $C_{j\to i|K}$ is selected and added to K, until the value of the maximum causation entropy reaches zero. Then, in the "backward" stage, each node j in K is selected and removed from K if $C_{j\to i|K} = 0$.

In Ref. [8] we proved that the "optimal causation entropy" procedure (oCSE) correctly and exactly identifies the minimal and unique set of causal components under mild conditions imposed on the underlying probability distribution of the Markov process. Numerical tests suggest that the number of samples needed for accurate inference scales as the average degree rather than the total number of nodes in the network, making oCSE an attractive approach for the inference of large sparse networks.

Additional Details

In practice, to apply oCSE for network inference, causation entropy need to be *estimated* from time series data, $\{\vec{x}_t\}$. One popular approach when dealing with biological data is to discretize the data into a finite number of states (for example, treating the state of a gene as either "on" or "off" while ignoring the actual level of expression). Then, the effective state of each variable can be regarded as a discrete random variable and the entropy estimation becomes straightforward by estimating the discrete probabilities. The estimation of causation entropy (and other types of entropy) is more challenging when dealing with continuous random variables that cannot be treated as discrete variables [1]. In this case more sophisticated estimation techniques need to be adopted, many of which are non-parametric, with the k-nearest neighbor estimation being a typical choice for high-dimensional systems [3, 9, 10].

Another practical issue is to determine whether an estimated causation entropy $C_{j\to i|K}$ should be regarded as zero (as opposed to positive). A common approach is to "shuffle" the related time series to construct an empirical cdf for the null hypothesis that $C_{j\to i|K} = 0$ and compare with the estimated value. In particular, we found that (time) shuffling the time series of X_t^j while leaving those of X_t^i and X_t^K unchanged works out reasonably well [7, 8].

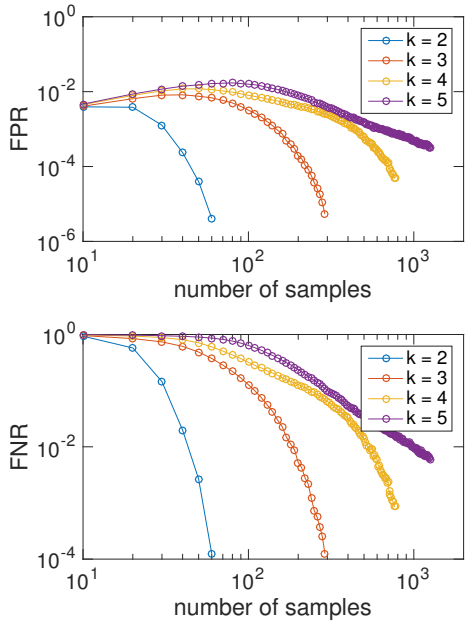


Figure 1: Inference of random networks of $n = 320$ genes via oCSE. Each gene is directly controlled by k genes. Top: false positive ratio (FPR) as a function of the number of sampled transitions for $k = 2, 3, 4, 5$. Here FPR is defined as the number of false positive links in the inferred network normalized by the total number of possible false positives. Bottom: same as top, for the false negative ratio (FNR).

Reverse Engineering of Biological Networks

Biological systems are commonly modeled as dynamics on networks. A notable example is gene-regulatory networks. In such a network the genes are turned "on" and "off" depending on the state of the other genes. Each gene can be regarded as a node and its causal components are the set of genes that directly influence its state. We generate synthetic data from a random network of $n = 320$ genes where the on/off state of each gene is determined by k randomly chosen genes. Applying the oCSE approach, we are able to infer the underlying network with relatively small number of samples and the algorithm remains computationally efficient even when the largest degree is not known a priori, in sharp contrast to classical gene-network inference methods which typically works for $k \leq 3$ [4].

References

[1] C. Cafaro, W. M. Lord, J. Sun, and E. M. Bollt. Causation entropy from symbolic representation of dynamical systems. *Chaos*, 25:043106, 2015.

[2] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Hoboken, NJ, 2 edition, 2006.

[3] A. Kraskov, H. Stogbauer, and P. Grassberger. Estimating mutual information. *Phys. Rev. E*, 69:066138, 2004.

[4] S. Liang, S. Fuhrman, and R. Somogyi. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing*, 3:18–29, 1998.

[5] C. E. Shannon. A mathematical theory of communications. *Bell System Tech. J.*, 27:379–423, 1948.

[6] J. Sun and E. M. Bollt. Causation entropy identifies indirect influences, dominance of neighbors and anticipatory couplings. *Physica D*, 267:49–57, 2014.

[7] J. Sun, C. Cafaro, and E. M. Bollt. Identifying coupling structure in complex systems through the optimal causation entropy principle. *Entropy*, 16:3416–3433, 2014.

[8] J. Sun, D. Taylor, and E. M. Bollt. Causal network inference by optimal causation entropy. *SIAM Journal on Applied Dynamical Systems*, 14:73–106, 2015.

[9] M. Vejmelka and M. Palus. Inferring the directionality of coupling with conditional mutual information. *Phys. Rev. E*, 77:026214, 2008.

[10] I. Vlachos and D. Kugiumtzis. Nonuniform state-space reconstruction and coupling detection. *Phys. Rev. E*, 82:016207, 2010.

GENERATING MAXIMALLY DISASSORTATIVE GRAPHS WITH GIVEN DEGREE DISTRIBUTION.

Pim van der Hoorn, Luidmilla Ostroumova Prokhorenkova, Egor Samosvat

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary
We consider the optimization problem of generating graphs that minimize degree correlations. We describe an algorithm that solves this problem and obtain a complete characterization of the joint degree structure of these maximally dissassortative graphs. More interestingly, we show that for maximally disassortative graphs with scale-free degree distribution, the asymptotic value of the rank correlation measure Spearman's rho increases as the exponent of the distribution increases.

Introduction
Since their introduction in [2] the correlation between the degrees of a randomly sampled edge, called degree correlations, have become part of the standard set of topological features of networks. Many research has been done on finding consistent measures and null-models for these correlations as well as analyzing their impact on other properties of, and processes on networks. However, there are still many open questions related to these correlations, such as the existence and structure of graphs with extreme (positive)negative correlations. In [1] a first attempt is made to analyze asymptotic properties of such extreme graphs. Here we continue this idea and investigate both the construction of graphs which are maximally disassortative as well as the behavior of the minimal value of the correlation measure on such graphs.

Degree distribution
Given a graph G_n of size n and degree sequence $\mathbf{D}_n = \{D_1, \dots, D_n\}$, we let L_n be the number of edges and denote the empirical degree and size-biased degree distribution by

$$f_n(k) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{D_i=k\}}, \tag{1}$$

$$f_n^*(k) = \frac{1}{2L_n}\sum_{i=1}^{n} D_i \mathbb{1}_{\{D_i=k\}}. \tag{2}$$

We will assume that there exist distributions f and f^*, on the positive integers, such that for some $\varepsilon > 0$,

$$\lim_{n\to\infty} \mathbb{P}\left(\max(d_1(F_n, F), \|F_n^* - F^*\|_1) \le n^{-\varepsilon}\right) = 1. \tag{3}$$

Here the capital letters denote the corresponding cumulative distributions, d_1 is the Wasserstein metric and $\|F_n^* - F^*\|_1 = \sum_{k=0}^{\infty} |f_n^*(k) - f^*(k)|$.

Spearman's rho
A consistent measure for degree correlations in graphs, introduced in [3], is Spearman's rho. If we define $\mathcal{F}_n^*(k) = F_n^*(k) + F_n^*(k-1)$, it follows from [4] that we have the following asymptotically equivalent expression for Spearman's rho

$$\rho(G_n) = \frac{3}{L_n}\sum_{i\to j} \mathcal{F}_n^*(D_i)\mathcal{F}_n^*(D_j) - 3 \tag{4}$$

This equation shows that $\rho(G_n)$ is completely determined by the wiring of the graph and the size-biased degree distribution.

Generating disassortative graphs
We propose a very straight forward Disassortative Graph Algorithm (`DGA`), for generating disassortative graphs with given degree sequence.

Given a degree sequence $\mathbf{D}_n$, we rank the nodes, in ascending order, by their degree, and let $\phi(i)$ denote the node whose rank is i, i.e $D_{\phi(n)} \ge D_{\phi(n-1)} \ge \cdots \ge D_{\phi(1)}$. Define z_n to be the unique integer such that $H_n(z_n) \ge 1/2$ and $H_n(z_n - 1) < 1/2$. We will create two list S and T, of stubs as follows;

- S: Starting with node $\phi(n)$ we add $D_{\phi(n)}$ stubs to S labeled $\phi(n)$. We do the same for node $\phi(n-1)$ and proceed until we reach a node with $D < z_n$.
- T: Let N_i denote the number of nodes with degree $D_{\phi(i)}$. We start by taking $D_{\phi(1)}$ copies of the set of stubs labeled $\phi(1), \dots, \phi(N_1)$ and add them to T. Then we proceed to node $\phi(N_1 + 1)$ and continue until we reach a node with $D > z_n$.

For an example of S and T, see Figure 1. To create edges we move down these lists simultaneously, pairing stubs in S to the one in T with the same index, until the degrees in both lists equals z_n. Observe that if $H_n(z_n) = 1/2$ we will have paired all available stubs. If this is not the case we are left with stubs belonging to all nodes with degree

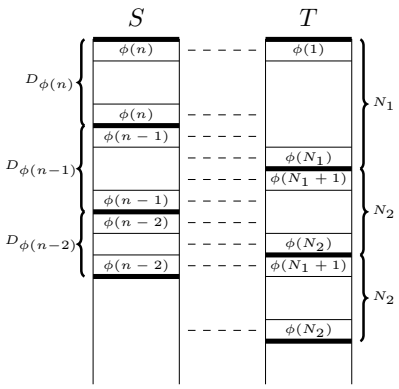


Figure 1: Example of the ordering of stubs in the lists S and T, and the pairing as done by DGA when $D_{\phi(1)} = 1$ and $D_{\phi(2)} = 2$.

z_n. We now pair these stubs in a greedy fashion that minimizes the number of self loops and multiple edges.

To see that DGA generates graphs that the minimizes $\rho(G_n)$ observe that by (4) this problem is equivalent to

$$\min_{\sigma \in S_{2L_n}} \sum_{i=1}^{2L_n} a_i a_{\sigma(i)},$$

where $0 \le a_1 \le \cdots \le a_{2L_n}$ correspond to the ordered degrees of the stubs and S_{2L_n} is the set of permutations of $\{1, \ldots, 2L_n\}$. Since any specific wiring corresponds to a permutation σ and the above minimum is attained for any permutation $\sigma(1) \ge \cdots \ge \sigma(2L_n)$, it follows that DGA solves the optimization problem of minimizing $\rho(G_n)$.

Joint degree distribution

Let $h_n(k, \ell)$ denote the empirical joint degree distribution

$$h_n(k, \ell) = \frac{1}{2L_n} \sum_{i \to j} \mathbb{1}_{\{D_i = k\}} \mathbb{1}_{\{D_j = \ell\}},$$

then we have the following

Theorem 1.1. *Let G_n be generated by the DGA. Then*

$$h_n(k, \ell) \xrightarrow{\mathbb{P}} \psi(k, \ell)\mathcal{E}(k, \ell) \text{ as } n \to \infty,$$

where

$$\psi(k, \ell) = \mathbb{1}_{\{1 - F^*(k) < F^*(\ell)\}} \mathbb{1}_{\{1 - F^*(k-1) > F^*(\ell - 1)\}} \text{ and}$$
$$\mathcal{E}(k, \ell) = \min(1 - F^*(k-1), F^*(\ell)) - \max(1 - F^*(k), F^*(\ell - 1)).$$

Lower bound on Spearman's rho

Since DGA minimizes $\rho(G_n)$ we can use convergence results from [3] to obtain that, for any $0 < \delta < \min(\varepsilon, 1)$,

$$\lim_{n \to \infty} \mathbb{P}\left(\rho(G_n) > \rho(X, Y) - n^{-\delta}\right) = 1,$$

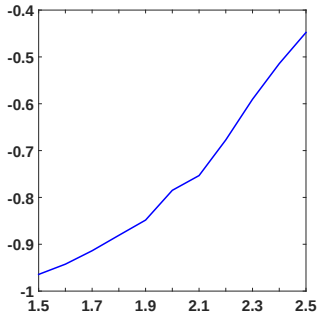


Figure 2: Plot of $\mathbb{E}[\rho(G_n)]$ (y-axis), for graphs of size 10^7 and degree distribution (5) for different γ (x-axis), generated by the DGA.

where X and Y have joint degree density $h(k, \ell) = \psi(k, \ell)\mathcal{E}(k, \ell)$ and $\rho(X, Y)$ denotes the value of Spearman's rho for X and Y. When $f^*(1) > 1/2$ we can obtain the more explicit lower bound $9f^*(1)^2 - 6f^*(1)^3 - 3$, which can be strictly larger than -1 for specific densities f^*. To illustrate this effect we consider degrees that have a scale-free distribution

$$\mathbb{P}(D > t) \sim t^{-\gamma} \quad \gamma > 1. \tag{5}$$

For different values of γ we constructed 10^3 degree sequences, by sampling them in an i.i.d. fashion from (5), generated graphs G_n using DGA and computed $\rho(G_n)$. The result are plotted in Figure 2, where we clearly observe that the minimal value of $\rho(G_n)$ increases as γ increases.

Conclusion

Although many aspects of degree correlations, such as measures and neutral mixing models, have received much attention in the literature, not much is known about networks with extreme (positive)negative correlation. We provide a first step by characterizing the joint degree structure of maximally disassortative graphs and showing that the measured value of these correlations depends on the (size-biased) degree distribution.

References

[1] J. Menche, A. Valleriani, and R. Lipowsky. Asymptotic properties of degree-correlated scale-free networks. *Physical Review E*, 81(4):046103, 2010.

[2] M. E. Newman. Assortative mixing in networks. *Physical review letters*, 89(20):208701, 2002.

[3] R. van der Hofstad and N. Litvak. Degree-degree dependencies in random graphs with heavy-tailed degrees. *Internet mathematics*, 10(3-4):287–334, 2014.

[4] P. van der Hoorn and N. Litvak. Degree-degree dependencies in directed networks with heavy-tailed degrees. *Internet Mathematics*, 11(2):155–179, 2015.

UNCOVERING POLITICAL IDEOLOGIES USING SOCIAL NETWORKS' TRACES

Corentin Vande Kerckhove, Mickael Temporão, Yannick Dufresne

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

This paper estimates the left-right ideological positions of 43,127 social network users by analyzing their published textual data and the network data derived from their interactions with other users. It appears that estimations from textual and network data are in line with the left-right ideology as measured with survey data.

Introduction

Reference to left-right ideology is prominent in everyday political discussions. Many studies have already demonstrated the potential of the large quantities of textual and network data increasingly available through social media to determine ideological positions [2, 4]. Despite the fundamental contribution to the field, many doubts prevail regarding the use of social networks data.

This paper makes use of a large-scale survey of Twitter users ($n = 43,127$) to validate ideal-point estimation of individual social media users' ideology. It innovates by comparing ideological estimations from social network analysis *and* unsupervised automated content analysis of textual data.

Material

This analysis distinguishes between two types of social network users: candidate users and all the other users. The candidate users are those running as candidates for one of the four principal national parties[1] during the 2015 Canadian federal campaign. Candidates are analysed separately as their political views can be easily associated to those of the political parties they are running for.

Our research is based on a large online survey of social media users ($n = 43,127$) collected by Vox Pop Labs[2]. Respondents had the possibility to share their Twitter account information for research purposes. Publicly-available network and textual data were then collected using Twitter's API. In order to circumvent the comparison problem of multilingual textual analysis, the analyses is restricted to English-speaking users.

Methods

We estimate political ideologies of individuals using two different methods. The first method uses individual users' *network information* ("followers"), and the second method uses *textual data* that they expressed online ("tweets"). Users' estimated ideology scores are then validated with users' position on a left-right ideological scale built from users' responses to 30 survey questions on political issues. Such additive scales are superior to single ideological self-placement questions as they have been shown to considerably reduce measurement error [1].

In the following, we denote a_{ik} the attitude of a respondent i related to a specific issue k in the survey data. In this work, we assume the existence of an underlying ideology $\boldsymbol{x}$ in a one-dimensional space, interpreted as the corresponding left-right position of the individual user. The related factor analysis model (1) assumes independent error terms ϵ_k. The 30 loading factors θ_k and the expectation value of users' ideologies x_i are estimated by maximum-likelihood methods using expectation-maximization.

$$a_{ik} = \theta_k x_i + \epsilon_k \tag{1}$$

The ideological estimates based on *network information* are computed from the existing links between users' and candidates' accounts. The extraction process relies on the assumption that social networks are homophilic. The model states that users tend to follow candidates with specific attitudes that lie close to their opinions. Equation 2 assigns to a dependent variable y_{ij} a value of 1 when a user i follows a candidate j, and a value of 0 otherwise.

$$Pr(y_{ij} = 1) = logit^{-1}(s_i + p_j - \sum_{k \in K} ||a_{ik} - a_{jk}||^2)) \tag{2}$$

The term K represents a subset of issues that have a major impact on users' decisions to follow candidates. The

[1] Canadian political parties can be roughly sorted from left to right: New Democrats, Greens, Liberals, Conservatives.

[2] This survey is a subsample of a larger online survey (more than 1,300,000 respondents) through the Vote Compass application. For more information: www.votecompass.com.

two variables p_j and s_i models respectively the effect of candidates popularity and users susceptibility to follow political accounts. Replacing the attitude variables by $\theta_k x_i$ and $\theta_k x_j$ (as suggested by equation 1) conducts to the popular roll-call model [3]. MCMC methods allows extracting a set of ideologies estimates ($\hat{x}_i^{net}$,$\hat{x}_j^{net}$) for the user and candidate respondents [2]. The procedure does not require any prior on K since we only consider the single parameter $\gamma = \sum_{k \in K} \theta_k^2$ in the roll-call model.

The ideological estimates based on *textual data* are computed on the frequency distribution of bigrams (i.e., sequences of two adjacent words) in individual users' tweets. The assumption made in the extraction process suggests that the use of particular bigrams reflects the publisher's attitudes. A specific bigram dictionary is constructed using common popular words expressed by candidates during the campaign period. To this end, the term a_{ib} can be interpreted as user i's global attitude towards an underlying context related to the bigram b. The observed frequency of bigrams in "tweets" are supposed to follow a Poisson process parametrised by the attitudes:

$$\lambda = \exp(s_i + p_b + a_{ib}) \tag{3}$$

Once again, the model introduces two parameters p_b and s_i to deal with the more frequent bigrams and more active users or candidates. Similarly, equation 1 suggests that a_{ib} can be decomposed in the product of θ_b and x_i. The resulting ideologies $\hat{x}_i^{text}$ are then estimated using the "Wordfish" scaling algorithm [4].

Preliminary results

The preliminary results are based on 78 candidates and a random sample of 1,890 users who meet the filtering criteria. In a first step, we assess the extraction methods on candidates as they can easily be linked to their respective party. Party discipline is strong in Canada. Therefore, we expect our estimates to identify clusters of candidates belonging to the same party. Our results show (Figure 1) that the two methods (text and network ideal points) are effective at scaling candidates in our sample. We observe a 90% of correlation between the output of the two ideal point estimates for candidates based on social networks data. The identified clusters are consistent with party positions found in the survey data. It is worth noting that in the textual data analysis, ideal points of bigrams are successful where unigrams fail.

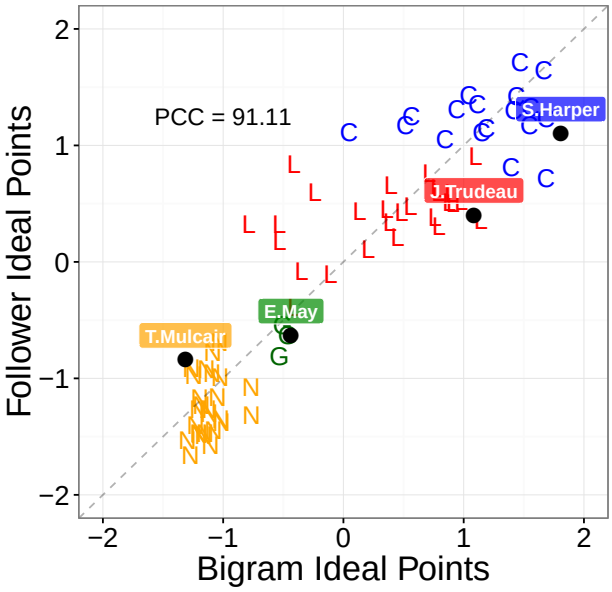


Figure 1: Comparison of ideal points of 78 candidates ideologies with textual data (bigrams) and network data ("followers")

In a second step, to validate the two methods at the user level, we compared the ideal points to the survey data ideal points. The comparison of the ideal points of users' networks to their ideal points of surveys' answers highlights a 60% correlation. Similar results are observed for textual data at the user level.

At the end, this work illustrates the potential of social network data to extract valid ideological positions of individual users by combining two complementary extraction methods.

References

[1] S. Ansolabehere, J. Rodden, and J. M. Snyder. The strength of issues: Using multiple measures to gauge preference stability, ideological constraint, and issue voting. *American Political Science Review*, 102(02):215–232, 2008.

[2] P. Barberá. Birds of the same feather tweet together: Bayesian ideal point estimation using twitter data. *Political Analysis*, 23(1):76–91, 2015.

[3] J. Clinton, S. Jackman, and D. Rivers. The statistical analysis of roll call data. *American Political Science Review*, 98(02):355–370, 2004.

[4] J. B. Slapin and S.-O. Proksch. A scaling model for estimating time-series party positions from texts. *American Journal of Political Science*, 52(3):705–722, 2008.

MEAN-FIELD MODELS FOR TIME-AGGREGATED TEMPORAL NETWORKS

Haley A. Yaple, Catherine Northrup, Elisabeth Rutter, Kerry Stapf

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We measure the effects of time step aggregation on dynamics for networks where links represent transient connections. We derive mean-field models and corrections that depend on the number of active links at any given time. This work was completed by a team of three undergraduate students as part of a summer research program.

Motivation

When networks are used in dynamics simulations to model interactions, often links are assumed to exist for all time. However, this may neglect important information, allowing flows through the network that are not possible when links are short-lived. For example, to model a fast-spreading disease, one should take into account the order in which an infected person visited their friends: if they visit an infectious friend at the beginning of the day they may spread the disease to their other friends later, while if they visit the infectious friend at the end of the day they cannot spread the infection further. Temporal scales of such interactions have been studied [1], and several models have been proposed [2]. However, the effect on dynamics, the focus of this work, is not yet fully understood.

Dynamics

We use SI (susceptible/infected) epidemic spread to study how time step aggregation may affect dynamics. This model was chosen due to its simplicity. We further simplify the dynamics by assuming guaranteed disease transmission. Thus, any susceptible node connected by an active edge to an infected node becomes infected at the next time step.

Modeling Temporal Connections

We make several simplifying assumptions. First, we assume each link occurs at a distinct interval of time of uniform duration. Second, we assume each link is only active once throughout a simulation (no repeated interactions). We build random networks according to the Erdős-Réyni and Barabási-Albert algorithms, then select edges at random over which infections may spread.

To investigate the regime between no aggregation and complete aggregation, we select w edges from the network per simulation time step, with order determined at random. In this way, w acts as a parameter in our model: when $w = 1$ causality is completely respected and for $w = m$ all m edges exist at simultaneously. Then $1 < w < m$ represents a simplification wherein w time steps are aggregated.

Mean-Field Equations

We derive mean-field differential equations describing the SI dynamics, based on the fraction of active edges. This fraction is w/m, or $2w/(nz)$ if there are n nodes of average degree z. It follows that the resulting equation is equivalent to the mean-field model where time is rescaled by the fraction of edges that exist. That is, the fraction of infected nodes I is described well by

$$\frac{dI}{dt} = \frac{2w}{nz} I(1 - I). \tag{1}$$

While the above matches simulation well for an Erdős-Réyni network, more care is necessary for other network types. Assuming that the heterogeneous degree distribution plays a major role in the dynamics for Barabási-Albert networks, we extend (1) using the same scale factor of $2w/(nz)$ with a degree-based mean field model [3]. This modification gives significant improvement.

To help validate our model's performance, we tested its convergence to simulation behavior as network size varies. The measured error between theory and simulation decreases rapidly as network sizes increases and other parameters are held fixed (see Figure 1).

Corrections in Aggregating Time Steps

In the initial formulation above, time steps are aggregated simply by allowing edges to exist simultaneously, without allowing for transmission of disease to next-neighbors of infected nodes. In general, to better represent aggregation of time steps, there should be w rounds of infection between neighboring nodes. Neglecting this scenario is a good first approximation, as it will only occur when activated edges share a common node. While uncommon for small w, this becomes likely when w/m is near one. Figure 2 shows a schematic of this situation.

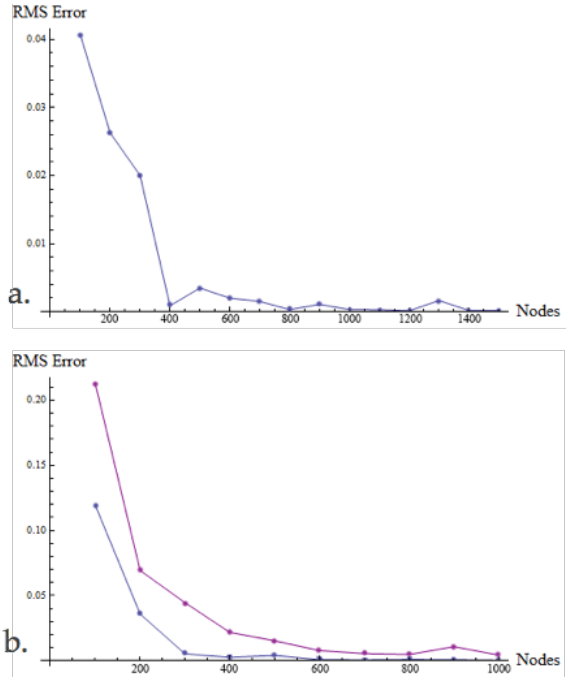


Figure 1: Convergence testing for (a) Erdős-Réyni and (b) Barabási-Albert networks. For the latter, agreement with both standard mean-field model (upper curve), and degree-based mean-field model (lower curve) are shown. Note that for each, the error between simulation and theory rapidly decreases as the number of nodes increases.

We derive corrections that account for this, by hand for $w = 1, 2, 3$. For large w, we generate the corrected mean-field equation using an algorithm coded in Mathematica. See Figure 3 for a comparison of the corrected model with simulations that allow for multiple rounds of infection.

Discussion

We have found that time step aggregation in a mean-field model is equivalent to rescaling time by the fraction of edges activated. The same behavior is observed when we move to the more specialized degree-based mean field model, providing evidence that this rescaling may hold in general. The correction terms to allow for next-neighbor transmission during an aggregated time step are small when few edges are aggregated.

While the results shown above are promising, they do not show that our model (even with corrections) is a good fit for w/m near one. We hypothesize that this is due to the small diameter of the networks used in our simulations. For example, including the theoretical likelihood of infecting your fifth-nearest neighbor is not important if the network has diameter four. This may be tested in the future by comparing results for networks generated using the low-diameter Barabási-Albert algorithm with configuration model networks of equivalent degree distributions. Testing our results in the case of non-guaranteed transmission may also illuminate underlying effects.

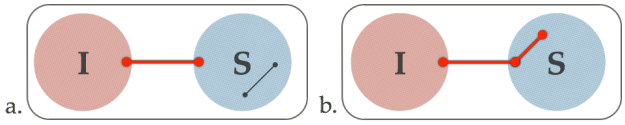


Figure 2: In this diagram of possible scenarios for disease spread when $w = 2$, the infected group of nodes is labeled I and the susceptible group S. The more likely scenario is shown in (a): only nearest-neighbors may become infected. In (b) the edges overlap at a susceptible node, meaning that the next-neighbor of the infected node will also become infected (when simulation allows for this).

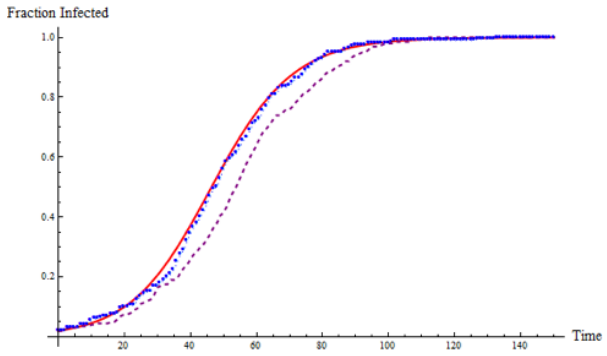


Figure 3: Comparison of mean-field theory, with corrections to account for multiple rounds of infection. Results shown are for $w = 4$, $n = 100$. Model theory is solid red line, simulation with multiple rounds of infection is blue dotted line, simulation with single rounds of infection is purple dashed line.

References

[1] Caceres, Rajmonda Sulo, Tanya Berger-Wolf, and Robert Grossman. "Temporal scale of processes in dynamic networks." *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on.* IEEE, 2011.

[2] Holme, Petter, and Jari Saramki. "Temporal networks." *Physics reports* 519.3 (2012): 97-125.

[3] Porter, Mason A., and James P. Gleeson. "Dynamical systems on networks: a tutorial." *arXiv preprint arXiv:1403.7663* (2014).

ROBUST COMBINATORIAL OPTIMIZATION ON MULTIPLE NETWORKS

Serena Yuan and Gordon Peng

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

We present a data-based approach to robust combinatorial optimization for multiple network samples with cost uncertainty, motivated by the hypothesis-testing based scheme proposed by [4] for matching appropriate uncertainty sets to robust optimization problems. We include hypothesis tests developed in [1], who developed a framework for comparing one or more samples of networks, and some analogues of the classical one- and two-sample t-statistics in the space of networks. From these tests and their confidence regions, we provide new uncertainty sets for effective robust formulations of optimization problems. As a result we can examine multiple network-based datasets through the robust combinatorial optimization solutions, with applications to diverse settings, such as network flow problems, and network-based representations of images in neuroimaging, diffusion tensor imaging, and functional magnetic resonance imaging.

Abstract

Robust Optimization is popular in optimizing under uncertainty, whose approach generally involves defining a set of possible realizations of uncertain parameters, and to optimize against worst-case realizations within this "uncertainty" set. As is well understood by the optimization community, RO is useful in dealing with erroneous or noise-corrupted data. In particular, for the user to have some understanding of the structure of the uncertainty set, it provides a set of tools and techniques that are useful in solving different kinds of uncertainties; the "model error" or "noisy data" as well as complex, stochastic kinds of uncertainty of a explicit model, in a way that is computationally feasible [3]. In Robust formulation of discrete optimization problems, it is natural associate a network with a matrix, where we must deal with uncertainty associated to the matrix A and the cost vector c [**?**].

Combinatorial optimization is a class of discrete optimization where the decision variables are binary, $x \in X \subseteq \{0,1\}^n$. A prominent example of the generalized combinatorial optimization problem is:

$$\begin{aligned} &\text{minimize } c'x \\ &\text{subject to } x \in X. \end{aligned}$$

Examples of combinatorial optimization problems include the shortest path, the minimum spanning tree, traveling salesman, the minimum assignment, the vehicle routing, and matroid intersection problems.

There are many questions in the RO literature involving how the choice of uncertainty set influences certain attributes of the optimization process.

A big issue in RO includes tractability of several data-driven uncertainty sets as outlined by [4].It involves the issue of how to structure the uncertainty set R so that the resulting problem is tractable and optimally trades off expected return with loss probability, in the terms of portfolio optimization. There are fundamental connections between distributional ambiguity, risk measures, and uncertainty sets in RO.

A prominent connection is given by probability guarantees. This issue asks, what does robust feasibility imply about probability of feasibility, or, what is the smallest ϵ we can find such that

$$x \in X(\mathcal{U}) \Rightarrow \mathbb{P}(f_i(x, u_i) > 0) \leq \epsilon$$

under assumptions on a distribution for u_i?

Choosing a good set is very important, as it would yield tractable optimization problems from robust models whose solutions perform well. Poorly chosen sets lead to robust models that may be overly conservative or computationally intractible.

Development of a data-driven theory of RO is interesting from a theoretical perspective and also has many applications in the real-world of data. As stated in [3], most of the models in the RO literature are not directly connected to data. In [4], a data-driven methods for designing uncertainty sets for robust optimization are described, drawing upon what the confidence region of hypothesis testing can tell us about the distribution from the data. We generally

follow the scheme [[4] Sec. 3] to match certain problems with optimal uncertainty sets based on three factors: a priori assumptions on the distribution, the data, and a hypothesis test. When the data are drawn i.i.d. from an unknown distribution Q, uncertainty sets that built from the scheme imply a probabilistic guarantee for Q at any desired level ϵ.

For example, several sets given in [4] are $\mathcal{U}_\epsilon^{\chi^2}$,$\mathcal{U}_\epsilon^{G}$,$\mathcal{U}_\epsilon^{M}$,$\mathcal{U}_\epsilon^{LCX}$,$\mathcal{U}_\epsilon^{CS}$, and $\mathcal{U}_\epsilon^{DY}$, or the uncertainty sets corresponding to hypothesis testing methods of χ^2-test, G-test, Marginal samples, Linear convex ordering, Shawe-Taylor and Cristianini (2003) [5], and Delage and Ye (2010) [2], respectively.

Moreover, we can combine this novel procedure with [1]'s hypothesis testing method for statistical inference on samples of networks. We can consider the confidence regions for these proposed hypothesis tests.

What [1] illustrates is the connection between the mathematical properties of the geometric space of networks, and the mean averaging of these networks.

Let $G = (V, E, W)$ denote a weighted undirected graph with weights $w_{ij} = w_{ji} \geq 0$, and associate to each G its Laplacian $L = D(W) - W$, where D denotes a diagonal matrix of weighted degrees; i.e., $D_{jj} = d_j(W) = \sum_{i \neq j} w_{ij}$. We also assume that G is simple, so that there is a one-to-one correspondence with graphs G and Laplacian matrices L. Therefore, the space of networks corresponds with the space of Laplacians. In this setting, we select a notion of averaging based on the Fréchet mean to derive a central limit theorem for sequences of network averages. Specifically, let $G_1, \cdots, G_n$ denote n (simple) graphs that have the same number of vertices, $|V|$. Let $L_1, \cdots, L_n$ be corresponding combinatorial Laplacians that are independent and identically distributed according to distribution Q.

From this method of hypothesis testing over sequences of network averages, we develop a new type of uncertainty set for robust optimization.

We run numerical experiments on these newly developed uncertainty sets for combinatorial optimization problems with network data and we compare the tractability, probability guarantees, and effectiveness of solutions associated with each of these sets. We prove that robust optimization problems over uncertainty sets we have developed are tractible.

References

[1] S. R. Cedric E. Ginestet, Prakash Balanchandran and E. D. Kolacyk. Hypothesis testing for network data in functional neuroimaging. *AMS*, 2014.

[2] E. Delage and Y. Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):596–612, 2010.

[3] D. B. B. Dimitris Bertsimas and C. Caramis. Theory and applications of robust optimization. volume 53, pages 464–501. Society for Industrial and Applied Mathematics, 2011.

[4] N. K. Dimitris Bertsimas, Vishal Gupta. Data-driven robust optimization. *Operations Research*, November 2014.

[5] J. Shawe-Taylor and N. Cristianini. Estimating the moments of a random vector with applications, 2003.