Higher-order networks: Exploring air traffic dynamics

Bradley Dice
bdice@bradleydice.com
University of Michigan
Department of Physics
Ann Arbor, Michigan

Daniel McCusker
dmccuske@umich.edu
University of Michigan
Applied Physics
Ann Arbor, Michigan

Shannon Moran
moranse@umich.edu
University of Michigan
Department of Chemical Engineering
Ann Arbor, Michigan

1 PROBLEM DEFINITION

Traditional network representations of data implicitly assume first-order dynamic processes. For instance, the next step of a random walker on a network depends only on its current position, and not its history. This has implications for network measures like PageRank and community detection which implement random walkers. In fact, the history of a trajectory on a network has significant bearing on predicting the trajectory's next step; for instance, a passenger flying from one city to the next is most likely to return to the first city on a round trip, rather than randomly flying to another city serviced by the second airport. Implementing higher-order network representations that account for such histories is key to achieving an accurate graph-level understanding of network data.

We propose applying the `BuildHON+` algorithm of Xu *et al.* in order to build a **higher-order network representation of airline itinerary data** [2, 6]. We will study the resulting **community structure using the MapEquation framework**, and compare the results to a first-order and a fixed second-order Markov network to illustrate the implications of higher-order representations.

2 CHALLENGES

First, why do we need higher-order networks, or HONs? HONs break up the physical nodes of the network into *state nodes* which incorporate trajectory history. For instance, the node *Atlanta* may be broken up into *Atlanta | Chicago*, *Atlanta | New York*, etc. for a fixed second-order representation.

However, a challenge in such fixed-order representation is the combinatorial explosion as the order k increases. Moreover, such networks suffer from overfitting when not all state nodes are statistically significant. For these reasons, we've chosen to adopt the HON representation method of Xu *et al.*, a variable-order method which determines only the statistically significant state nodes.

3 RELATED PRIOR WORK

We have identified three core parts of this project, building upon prior literature: (1) use of the airline data set as a test case for higher-order network (HON) methods, (2) representing data sets as HONs, and (3) identifying communities in HONs.

3.1 Prior airline analysis

Past work on the DB1B airline itinerary data set has used a fixed second-order network on data from 2011 Q1 to Q3 [4]. This data set was a benchmark for their HON model which shows much less entropy in the MapEquation model for a second-order network compared to a first-order network.

This study found that 79% of the paths in the data were of length 3 or greater, though the network used for analysis was strictly second order. We hope to expand on this by comparing different, longer subsets of the entire data set of 100 quarters (instead of just 3 quarters) to extract longer-ranged changes in communities that may be attributable to market shifts, mergers and acquisitions, and global events affecting the airline industry such as terrorism. This approach is subject to change, pending computational difficulty of working with such data sets.

3.2 Methods for forming higher-order networks from trajectory data

There are two primary ways to form higher-order networks from trajectory-like data. The simplest approach is to take a fixed order k and divide each of the graph's N physical nodes into "state nodes" representing trajectories of length k ending with that physical node. These state nodes encode the history of a trajectory, e.g. $\vec{hij}$ might represent a state node for the physical node j which had previously visited h and i before arriving at j [4]. Fixed order representations must ignore trajectories with total length less than k, since the trajectory cannot be represented in a second order state node [4]. Additionally, the number of state nodes must increase as N^k to record all possible trajectories, limiting this approach to small k (typically 2) [4]. A more complex representation is that of Xu *et al.* [7], later improved to be parameter-free [6], where the order is determined dynamically from data: state nodes represent trajectories of variable length, and higher-order state nodes are created if and only if their creation is statistically significant for the network's predictive power. This method allows for a sparse representation that can capture higher-order dynamics than fixed-order models, which are computationally constrained by the explosive growth in state nodes as k increases. As the authors clarify, "if the dependency is assumed as fixed second order, it could be redundant when first-order dependencies are sufficient and could be insufficient when higher-order dependencies exist" [7].

3.3 Clustering or other centrality algorithms for higher-order networks

An insight of Xu *et al.* is that typical network analysis methods, such as clustering, centrality, and ranking, can be applied to study higher-order networks, once a network of higher-order state nodes has been produced [7]. For instance, they apply MapEquation, a clustering algorithm based on the entropy of random walks, to a shipping data set. By merging the state nodes into physical nodes at the end, they naturally reveal a rich overlapping community structure, with important implications for studying the transmission of invasive species. This structure is hidden in the first-order network analysis.

4 CHOICE OF DATA SET

We were interested in choosing a data set representing time-based trajectories. Additionally, we wanted to choose one data set and apply multiple methods.

We considered data sets referenced in [5] and the network database maintained by Prof. Clauset at UC Boulder [1]. We chose data from the Airline Origin and Destination Survey (DB1B) collected by the Office of Airline Information of the Bureau of Transportation Statistics, because of its richness, long time span, and use in other papers in the field [2]. The data include origin, destination, and other itinerary details of passengers transported. This database can be used to describe air traffic patterns, air carrier market shares, and passenger flows.

We have already downloaded the complete data set, including *Coupon*, *Market*, and *Ticket* data aggregated on a quarterly basis from 1993 Q1 to 2018 Q3. The data size is 25 GB in zipped files. This includes 837 million observations of itinerary coupons, 507 million observations of origin/destination markets, and 286 million observations of airline tickets. We have some experience using PySpark[1], and plan to use this framework for our data analysis because of its compatibility with our university's Hadoop cluster, scalability, and ease of use for rapid prototyping in Python.

5 PROPOSED APPROACH

5.1 Converting data sets into HONs

We plan to use the improved `BuildHON+` algorithm of Xu *et al.* [6] to build a higher-order network of this data. We will subset the data to explore the network effects of real-world events such as major airline mergers (e.g. American Airlines and TWA, 2001; Delta and Northwest Airlines, 2008-2010; American Airlines and US Airways, 2013-2015). First, events will be merged on their unique identifiers to form trajectories. These trajectories are the input for the `BuildHON+` algorithm. Previous work on DB1B data by Rosvall *et al.* made some helpful assumptions about paths' endpoints (e.g. choice of memory nodes) that we plan to replicate in our analysis for simplicity and reproducibility [4].

5.2 Implementing HON representation using Python

There is an open-source Python implementation[2] of the HON algorithm of Xu *et al.* but our preliminary reviews of that code suggest that we may need to re-implement a portion of it for improved performance and scalability via PySpark's `DataFrames` and resilient distributed data sets (RDDs) [6, 7]. The algorithm is not too complex, so this should be within a reasonable scope for this project. We plan to vary the maximum HON order k from 1 to 5. The algorithm's time complexity is not strongly dependent on k after a certain point, as seen in Table 1 of the HON paper where the run time of $k = 3$ and $k = 5$ differ by only 8% [7]. This is because there are typically far fewer correlations of high order than low order, so testing correlations of order k for extension to order $k + 1$ gets cheaper as the number of correlations decreases.

[1] https://spark.apache.org/docs/latest/api/python/index.html
[2] https://github.com/xyjprc/hon

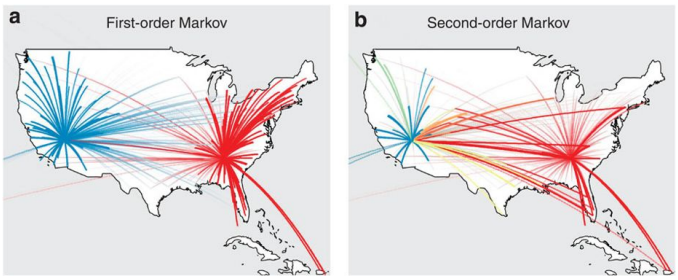

Figure 1: An example of community detection using the MapEquation framework on airline traffic [4]. While Las Vegas and Atlanta are their own communities in first-order Markov dynamics (a), using second-order Markov dynamics (b) we see communities (color) of "round-trip" tickets (line weight) to Las Vegas represented (and Las Vegas is included in 8 communities). Further discussion in text.

5.3 Methods used for cluster analysis of flight data HON.

We plan to implement MapEquation, the random walker-based community detection algorithm, to detect overlapping communities in the airport nodes [3]. This will allow us to directly compare our results to the fixed second-order airline representation in Rosvall's 2014 paper, as well as to the overlapping communities in shipping data in Xu's original HON paper [4, 7]. Xu's second paper [6], which introduces the `BuildHON+` algorithm, also uses anomaly detection. We might also investigate the airline network dynamics using this technique, and use the results, in addition to those from MapEquation, to build a comprehensive, qualitative description of the dynamics on the airline network.

6 EVALUATION OF METHODS

We will evaluate our results by benchmarking them to those found in the literature. At minimum our goal will be to replicate previous findings on flight data [4] and find communities that are not detectable without higher-order network topology.

As an example of what we will be comparing our work against, we present a case study from the literature in Figure 1[4]. For a subset of air traffic data between US cities, a first-order Markov model assigns Las Vegas and Atlanta to their own singular communities (Figure 1a). However, a second-order Markov model captures the "two-step return rate" that allows communities to be built based on short round-trip trajectories. Of interesting note in Figure 1b, Las Vegas is part of 8 communities while Atlanta is part of only 2. This is likely explained by Atlanta being a transfer hub while Las Vegas is a tourist destination, something only captured by a second order network. While some communities appear to be regionally close, others don't make sense (e.g. *Las Vegas | NYC* being in the same community as Atlanta). We suspect this would be improved with a variable k approach, and aim to test this hypothesis in our project.

REFERENCES

[1] Aaron Clauset, Ellen Tucker, and Matthias Sainz. 2016. The Colorado Index of Complex Networks. https://icon.colorado.edu/

[2] Bureau of Transportation Statistics. [n. d.]. Airline Origin and Destination Survey (DB1B), 1993-2018. https://www.transtats.bts.gov/tables.asp?DB_ID=125

[3] M. Rosvall, D. Axelsson, and C. T. Bergstrom. 2009. The map equation. *The European Physical Journal Special Topics* 178, 1 (nov 2009), 13–23. https://doi.org/10.1140/epjst/e2010-01179-1

[4] Martin Rosvall, Alcides V. Esquivel, Andrea Lancichinetti, Jevin D. West, and Renaud Lambiotte. 2014. Memory in network flows and its effects on spreading dynamics and community detection. *Nature Communications* 5, 1 (aug 2014). https://doi.org/10.1038/ncomms5630

[5] Ingo Scholtes. 2017. When is a Network a Network?: Multi-Order Graphical Model Selection in Pathways and Temporal Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17)*. ACM, New York, NY, USA, 1037–1046. https://doi.org/10.1145/3097983.3098145

[6] Jian Xu, Mandana Saebi, Bruno Ribeiro, Lance M. Kaplan, and Nitesh V. Chawla. 2017. Detecting Anomalies in Sequential Data with Higher-order Networks. arXiv:arXiv:1712.09658

[7] Jian Xu, Thanuka L. Wickramarathne, and Nitesh V. Chawla. 2016. Representing higher-order dependencies in networks. *Science Advances* 2, 5 (2016). https://doi.org/10.1126/sciadv.1600028