NETWORK SCIENCE

10.1126/sciadv.1600028

Representing higher-order dependencies in networks

Jian Xu,[1,2] Thanuka L. Wickramarathne,[2,3,4] Nitesh V. Chawla[1,2,4]*

To ensure the correctness of network analysis methods, the network (as the input) has to be a sufficiently accurate representation of the underlying data. However, when representing sequential data from complex systems, such as global shipping traffic or Web clickstream traffic as networks, conventional network representations that implicitly assume the Markov property (first-order dependency) can quickly become limiting. This assumption holds that, when movements are simulated on the network, the next movement depends only on the current node, discounting the fact that the movement may depend on several previous steps. However, we show that data derived from many complex systems can show up to fifth-order dependencies. In these cases, the oversimplifying assumption of the first-order network representation can lead to inaccurate network analysis results. To address this problem, we propose the higher-order network (HON) representation that can discover and embed variable orders of dependencies in a network representation. Through a comprehensive empirical evaluation and analysis, we establish several desirable characteristics of HON, including accuracy, scalability, and direct compatibility with the existing suite of network analysis methods. We illustrate how HON can be applied to a broad variety of tasks, such as random walking, clustering, and ranking, and we demonstrate that, by using it as input, HON yields more accurate results without any modification to these tasks.

INTRODUCTION

Today's systems are inherently complex, whether it is the billions of people on Facebook powering a global social network, the transportation networks powering the commute and the economy, or the interacting neurons powering the coherent activity in the brain. Complex systems such as these are made up of a number of interacting components that influence each other, and network-based representation has quickly emerged as the norm by which we represent the rich interactions among the components of such a complex system. These components are represented as nodes in the network, and the edges or links between these nodes represent the (ranges and strengths of) interactions. This conceptualization raises a fundamental question: Given the data, how should one construct the network representation such that it appropriately captures the interactions among the components of a complex system?

A common practice to construct the network from data (in a complex system) is to directly take the sum of pairwise connections in the sequential data as the edge weights in the network—for example, the sum of traffic between locations in an interval (*1*–*4*), the sum of user traffic between two Web pages, and so on (*5*–*7*). However, this direct conversion implicitly assumes the Markov property (first-order dependency) (*8*) and loses important information about dependencies in the raw data. For example, consider the shipping traffic network among ports, where the nodes are ports and the edges are a function of the pairwise shipping traffic between two ports. When interactions are simulated on the network, such as how the introduction of invasive species to ports is driven by the movements of ships via ballast water exchange, the next interaction (port-port species introduction) only depends on the current node (which port the ship is coming from), although, in fact, the interaction may be heavily influenced by the sequence of previous nodes (which ports the ship has visited before). Another example is user clickstreams on the Web, where nodes are Web pages and interactions are users navigating from one Web page to another. A user's next page visit not only depends on the last page but also is influenced by the sequence of previous clicks. Thus, there are higher-order dependencies in networks and not just the first-order (Markovian) dependency, as captured in the common network representation. Here, we focus on deriving the network based on the specific set of interactions, namely, the interactions induced by movements among components of a complex system, wherein the sequence of movement patterns becomes pivotal in defining the interactions.

Let us again consider the process of constructing a network from the global shipping complex system by incorporating the movements from the ship trajectories (Fig. 1A) (*9*, *10*). Conventionally (*1*–*7*, *9*), a network is built by taking the number of trips between port pairs as edge weights (Fig. 1B). When ship movements are simulated on this first-order network, according to the network structure where the edge *Singapore* → *Los Angeles* and the edge *Singapore* → *Seattle* have similar edge weights, a ship currently at Singapore has similar probabilities of going to Los Angeles or Seattle, no matter how it arrived at Singapore. In reality, the global shipping data indicate that a ship's previous stops before arriving at Singapore influence the ship's next movement: the ship is more likely to continue on to Los Angeles if it came from Shanghai and more likely to go to Seattle if it came from Tokyo. A first-order network representation fails to capture important information like this because, in every step, the flow of traffic on the network is simply aggregated and mixed. As a consequence, trajectories simulated on the first-order network do not follow true ship movement patterns. By contrast, by breaking down the node *Singapore* into *Singapore*|*Tokyo* and *Singapore*|*Shanghai* (Fig. 1C), the higher-order network (HON) can better guide the movements simulated on the network. Because ships can translocate species along intermediate

[1]Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA. [2]Interdisciplinary Center for Network Science and Applications, University of Notre Dame, Notre Dame, IN 46556, USA.[3]Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA.[4]Environmental Change Initiative, University of Notre Dame, Notre Dame, IN 46556, USA.
*Corresponding author. Email: nchawla@nd.edu

A Original data

Vessel	Depart	Sailing_date	Arrive	Arrival_date
V-001	Shanghai	2013-01-01	Singapore	2013-01-15
V-001	Singapore	2013-01-16	Los Angeles	2013-02-05
V-002	Singapore	2013-02-01	Los Angeles	2013-03-08
… …	… …	… …	… …	… …

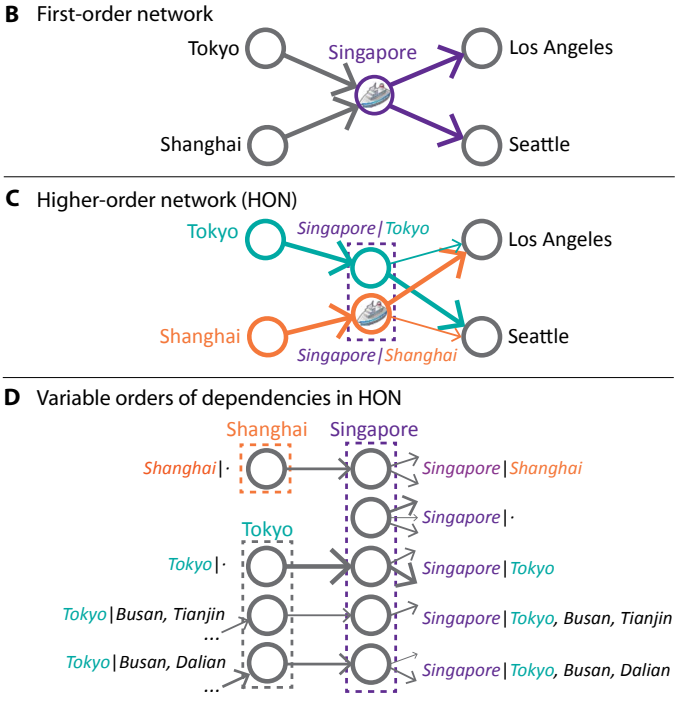

Fig. 1. Necessity of representing dependencies in networks. (**A**) A global shipping data set, containing ship movements as sequential data. (**B**) A first-order network built by taking the number of trips between port pairs as edge weights. A ship currently at Singapore has similar probabilities of going to Los Angeles and Seattle, no matter where the ship came to Singapore from. (**C**) By breaking down the node *Singapore*, the ship's next step from Singapore can depend on where the ship came to Singapore from and thus more accurately simulate movement patterns in the original data. (**D**) Variable orders of dependencies represented in HON. First-order to fourth-order dependencies are shown here and can easily extend to higher orders. Coming from different paths to Singapore, a ship will choose the next step differently.

stops via partial ballast water exchanges (*11*), the ability to distinguish between these cases is critical for producing accurate species introduction probabilities for each port.

Such higher-order dependencies exist ubiquitously and are indispensable for modeling vehicle and human movements (*12*), email correspondence, article and Web browsing (*13*–*15*), conversations (*16*), stock market (*17*), and so on. Although higher-order dependencies have been studied in the field of time series (*17*, *18*), information theory (*19*), frequent pattern mining (*20*), next-location prediction (*21*), variable-order Markov (VOM) (*22*–*24*), hidden Markov model (*25*), and Markov order estimation (*26*–*28*), they have focused on the stochastic process, rather than on how to represent higher-order dependencies in networks to adequately capture the intricate interactions in complex systems. In the field of network science, the frontier of addressing the higher-order dependencies still remains at the stage of assuming a fixed second order of dependency when constructing the network (*12*, *29*–*32*) or using multiplex networks (*33*), and there is neither a thorough discussion beyond second-order dependencies nor a systematic way of representing dependencies of variable orders in networks. Although there have also been efforts to incorporate HON structures for clustering (*34*, *35*), ranking (*36*), and so on, these approaches need to modify existing algorithms and are application-specific. As a result, these methods are not generalizable to broader applications, although we expect a network representation that is agnostic to the end-analysis methods (more discussions in Materials and Methods).

Here, we present a novel and generalizable process for extracting higher-order dependencies in the sequential data and constructing the HON that can represent dependencies of variable orders derived from the raw data. We demonstrate that HON is (i) more reflective of the underlying real-world phenomena (for example, when using HON instead of a first-order network to represent the global shipping data, the accuracy is doubled when simulating a ship's next movement on the network and is higher by one magnitude when simulating three steps); (ii) efficient in scaling to higher orders, because auxiliary higher-order nodes and edges are added to a first-order network only where necessary; and (iii) consistent with the conventional network representation, allowing for a variety of existing network analysis methods and algorithms to run on HON without modification. These algorithms and methods produce considerably different and more accurate results on HON than on a first-order network, thus demonstrating the broad applications and potential influences of this novel network representation.

We analyze a variety of real-world data including global shipping transportation, clickstream Web browsing trajectories, and Weibo retweet information diffusions. We show that some of them have dependencies up to the fifth order, which the conventional first-order network representations or the fixed second-order network representations simply cannot capture, rendering the downstream network analysis tools, such as clustering and ranking, with limited and possibly erroneous information about the actual interactions in data. We also validate HON's ability to reveal higher-order dependencies on a synthetic data set, where we introduced dependencies of variable orders through a process completely independent of the construction of HON. We show that HON accurately identifies all the higher-order dependencies introduced.

RESULTS

We start with an examination of the conventional network representation, showing its limitations and formally introducing the HON. Then, with multiple real-world and synthetic data sets, we compare our proposed network representation with the conventional ones in terms of accuracy, scalability, and observations drawn from network analysis tools.

The HON representation

Conventionally, a network (also referred to as a graph) $G = (V, E)$ is represented with vertices or nodes V as entities (for example, places, Web pages, etc.) and edges or links E as connections between pairs of nodes (for example, traffic between cities, user traffic between Web pages, etc.). Edge weight $W(i \rightarrow j)$ is a number associated with an edge $i \rightarrow j$ representing the intensity of the connection, which is usually assigned as the (possibly weighted) sum of pairwise connections $i \rightarrow j$ (for example, the daily traffic from i to j) in data (*1*–*7*, *9*).

A wide range of network analysis methods, such as PageRank for ranking (*37*), MapEquation (*38*) and Walktrap (*39*) for clustering, and link prediction methods (*40*, *41*) use random walking to simulate movements on networks (for example, ships traveling between ports, users clicking through Web pages, etc.). If the location of a random walker at time t is denoted as a random variable X_t, where X can take values from the node set V, then, conventionally (*40*, *42*), the transition probability from node i_t to the next step i_{t+1} is proportional to the edge weight $W(i_t \rightarrow i_{t+1})$

$$P(X_{t+1} = i_{t+1} | X_t = i_t) = \frac{W(i_t \rightarrow i_{t+1})}{\sum_j W(i_t \rightarrow j)} \quad (1)$$

This Markovian nature of random walking dictates that every movement simulated on the network is only dependent on the current node. In the conventional first-order network representation, every node maps to a unique entity or system component, so that every movement of a random walker is only dependent on a single entity (Singapore in Fig. 1B). Data with higher-order dependencies that involve more than two entities, such as "ships coming from Shanghai to Singapore are more likely to go to Los Angeles" in the global shipping data, cannot be modeled via the conventional first-order network representation. Thus, the simulation of movement performed on such networks will also fail to capture these higher-order patterns.

To represent higher-order dependencies in a network, we need to rethink the building blocks of a network: nodes and edges. Instead of using a node to represent a single entity (such as a port in the global shipping network), we break down the node into different higher-order nodes that carry different dependency relationships, where each node can now represent a series of entities. For example, in Fig. 1C, Singapore is broken down into two nodes: Singapore *given* Tokyo as the previous step (represented as *Singapore*|*Tokyo*) and Singapore *given* Shanghai as the previous step (represented as *Singapore*|*Shanghai*). Consequently, the edges *Singapore*|*Shanghai* → *Los Angeles* and *Singapore*|*Shanghai* → *Seattle* can now involve three different ports as entities and carry different weights, thus representing second-order dependencies. Because the out-edges here are in the form of $i|h \rightarrow j$ instead of $i \rightarrow j$, a random walker's transition probability from node $i|h$ to node j is

$$P(X_{t+1} = j | X_t = (i|h)) = \frac{W(i|h \rightarrow j)}{\sum_k W(i|h \rightarrow k)} \quad (2)$$

so that although a random walker's movement depends only on the current node, it now depends on multiple entities in the new network representation (as in Fig. 1C), thus being able to simulate higher-order movement patterns in the data. This new representation is consistent with conventional networks and compatible with existing network analysis methods, because the data structure of HON is the same as the conventional network (the only change is the labeling of nodes). This makes it easy to use HON instead of the conventional first-order network as the input for network analysis methods, with no need to change the existing algorithms. The algorithm to construct HON is in Materials and Methods.

Rosvall *et al.* (*12*) consider a higher-order dependency, albeit with a fixed second-order assumption. They propose a network representation composed of "physical nodes" and "memory nodes." As we will show with experiments, variable orders of dependencies can coexist in the same data set and can be up to the fifth order in our data. So 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. In HON, every node can represent an arbitrary order of dependency, so variable orders of dependencies can coexist in the same network representation, as shown in Fig. 1D. For example, the fourth-order dependency relationship following the path of *Tianjin* → *Busan* → *Tokyo* → *Singapore* can now be represented as a fourth-order node *Singapore*|*Tokyo*, *Busan*, *Tianjin*; the second-order path *Shanghai* → *Singapore* is now a node *Singapore*|*Shanghai*; first-order relationships are now in a node *Singapore*|·. Yet these nodes of variable orders all represent the same physical location Singapore. Compared with fixed-order networks, we will show that our representation is compact in size by using variable orders and embedding higher-order dependencies only where necessary.

Although the hypergraph (*43*) looks similar to HON in that its edges can connect to multiple nodes at the same time, it cannot directly represent dependencies. The reason is that dependencies are ordered relationships, but in a hypergraph the nodes connected by hyperedges are unordered. For example, in the shipping example, an edge in a hypergraph may have the form of set{*Tokyo*, *Busan*, *Tianjin*} → set{*Singapore*}, where Tokyo, Busan, and Tianjin are interchangeable and cannot represent the path of the ship before arriving at Singapore. On the contrary, the edges in HON have the form of {*Tokyo*|*Busan*, *Tianjin*} → {*Singapore*|*Tokyo*, *Busan*, *Tianjin*}, where the entities in nodes are not interchangeable. Thus, HON can represent dependencies of arbitrary order.

Higher-order dependencies in data revealed by HON

First, we show that HON can correctly extract higher-order dependencies from synthetic data. The synthetic data set has 10,000,000 generated movements, based on the predefined 10 second-order dependencies, 10 third-order dependencies, and 10 fourth-order dependencies (see note S1 for details). On this synthetic data with known variable orders of dependencies, HON (i) correctly captures all 30 of the higher-order dependencies out of the 400 first-order dependencies, with variable orders (from second-order to fourth-order) of dependencies mixed in the same data set correctly identified; (ii) does not extract false dependencies beyond the fourth order even if a maximum order of five is allowed; and (iii) determines that all other dependencies are first-order, which reflects the fact that there is no other higher-order dependency in the data.

We then explore higher-order dependencies in real data: the global shipping data containing ship trajectories among ports, the clickstream data containing user browsing trajectories among Web pages, and the retweet data containing information diffusion paths among users (see note S1 for details). The global shipping data reveal variable orders of dependencies up to the fifth order, indicating that a ship's movement can depend on up to five previous ports that it has visited. The clickstream data also show variable orders of dependencies up to the third order, indicating that the page a user will visit can depend on up to three pages that the user has visited before, matching the observation in another study on Web user browsing behaviors (*14*). The fact that dependencies of variable orders up to the fifth order exist in real data further justifies our approach of representing variable and higher-order dependencies instead of imposing a fixed first or second order. On the contrary, the retweet data (recording information

diffusion) show no higher-order dependency at all. The reason is that in diffusion processes, such as the diffusion of information and the propagation of epidemics, according to the classic spreading models (*44*), once a person A is infected, A will start to broadcast the information (or spread the disease) to all of its neighbors $\mathcal{N}(A)$, irrespective of who infected A. Because of this Markovian nature of diffusion processes, all diffusion data only show first-order dependencies, and HON is identical to the first-order network. This also agrees with a previous finding that assuming second-order dependency has "marginal consequences for disease spreading" (*12*).

Improved accuracy on random walking

Because random walking is a commonly used method to simulate movements on networks and is the foundation of many network analysis tools, such as PageRank for ranking, MapEquation and Walktrap for clustering, various link prediction algorithms, and so on, it is crucial that a naïve random walker (only aware of the current node and its outedges) can simulate the movements in the network accurately. If different network representations are built for the same sequential data set (consisting of trajectories), how will the network structure affect the movements of random walkers? Do the random walkers produce trajectories more similar to the real ones when running on HON?

We take the global shipping data to explain the experimental procedures (the clickstream data have similar results). As illustrated in Fig. 2A, for every trajectory of a ship, the last three locations are held for testing, and the others are used to construct the network. A first-order network (Fig. 2B), a fixed second-order network (*12*), and a HON (Fig. 2C) are constructed from the same data set, respectively. Given one of the networks, for every ship, a random walker simulates the ship's movements on the network: it starts from the last location in the corresponding training trajectory and walks three steps. Then, the generated trajectories are compared with the ground truth in the testing set: a higher fraction of correct predictions means that the random walkers can simulate the ship's movements better on the corresponding network. Random walking simulations in each network are repeated 1000 times, and the mean accuracies are reported. By comparing the accuracies of random walking, our intention here is not to solve a next-location prediction problem (*21*) or similar classification problems, but from a network perspective, we focus on improving the representative power of the network, as reflected by the accuracies of random walking simulations.

The comparison of results among the conventional first-order network, the fixed second-order network, and HONs with maximum orders of two to five is shown in Fig. 2D. It is shown that random walkers running on the conventional first-order network have significantly lower accuracies compared with other networks. The reason is that the first-order network representation only accounts for pairwise connections and cannot capture higher-order dependencies in ships' movement patterns. For example, a large proportion of ships are going back and forth between ports (for example, port *a* and port *b* in Fig. 2A), which is naturally a higher-order dependency pattern because each ship's next step is significantly affected by its previous steps. Such return patterns are captured by HON (Fig. 2C) but not guaranteed in a first-order network (Fig. 2B, where ships going from port a to port b may not return to port a). As shown in Table 1, the probability of a ship returning to the same port after two steps in a first-order network (10.7%) is substantially lower than that in HON (above 40%). From another perspective, in a first-order network, a random walker is given

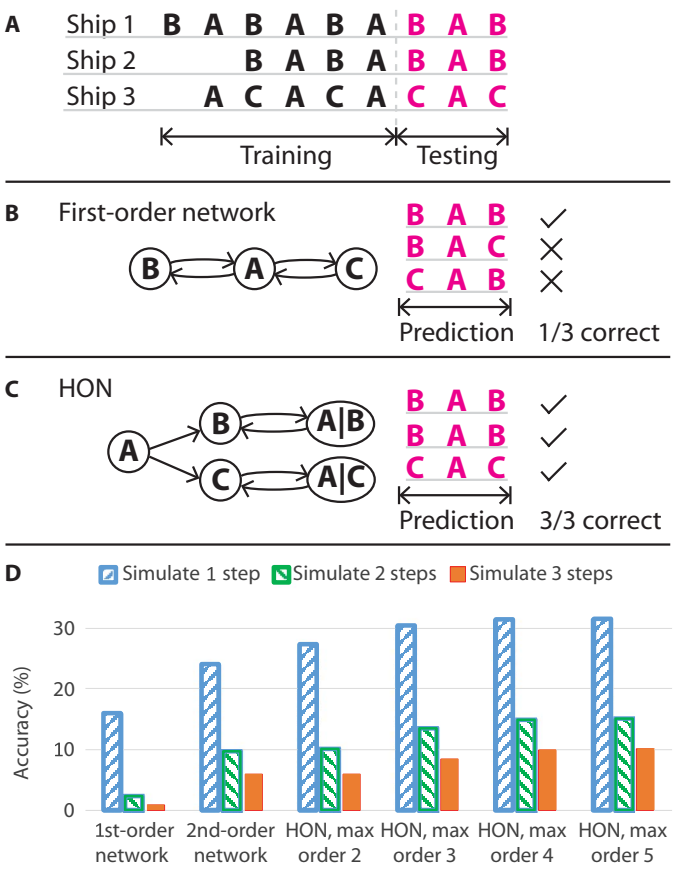

Fig. 2. Comparison of random walking accuracies. (**A**) For the global shipping data composed of ships' trajectories, hold the last three steps of each trajectory for testing and use the rest to build the network. (**B** and **C**) Given a generated shipping network, every ship is simulated by a random walker, which walks three steps from the last location in the corresponding training trajectory. The generated trajectories are compared with the ground truth, and the fraction of correct predictions is the random walking accuracy. (**D**) By using HON instead of the first-order network, the accuracy is doubled when simulating the next step and improved by one magnitude when simulating the next three steps. Note that error bars are too small to be seen (SDs on HONs are 0.11 ± 0.02%).

more choices every step and is more "uncertain" in making movements. Such "uncertainty" can be measured by the entropy rate (*12*, *19*), defined as

$$H(X_{t+1}|X_t) = \sum_{i,j} \pi(i) p(i \to j) \log p(i \to j) \quad (3)$$

where $\pi(i)$ is the stationary distribution at node i and $p(i \to j)$ is the transition probability from node i to node j, defined in Eq. 1. The entropy rate measures the number of bits needed to describe every step of random walking—the more bits needed, the higher the uncertainty. In Table 1, the first-order network has the highest entropy rate, indicating that every step of random walking is more uncertain because of the lack of knowledge of what the previous steps are, which leads to the low accuracy in the simulation of movements.

Table. 1. Comparing different network representations of the same global shipping data.

Network representation	No. of edges	No. of nodes	Network density	Probability of returning after two steps	Probability of returning after three steps	Entropy rate (bits)	Clustering time (min)	Ranking time (s)
Conventional first-order	31,028	2,675	4.3×10^{-3}	10.7%	1.5%	3.44	4	1.3
Fixed second-order	116,611	19,182	3.2×10^{-4}	42.8%	8.0%	1.45	73	7.7
HON, maximum order of two	64,914	17,235	2.2×10^{-4}	41.7%	7.3%	1.46	45	4.8
HON, maximum order of three	78,415	26,577	1.1×10^{-4}	45.9%	16.4%	0.90	63	6.2
HON, maximum order of four	83,480	30,631	8.9×10^{-5}	48.9%	18.5%	0.68	67	7.0
HON, maximum order of five	85,025	31,854	8.4×10^{-5}	49.3%	19.2%	0.63	68	7.6

By assuming an order of two for the whole network, the accuracies on the fixed second-order network increase considerably as in Fig. 2D, because the network structure can help the random walker remember its last two steps. Meanwhile, the accuracies on HON with a maximum order of two are comparable and slightly better than the fixed second-order network, because HON can capture second-order dependencies while avoiding the overfitting caused by splitting all first-order nodes into second-order nodes. Increasing the maximum order of HON can further improve the accuracy and lower the entropy rate; particularly, ship movements in bigger loops need more steps of memory and can only be captured with higher orders, as reflected in Table 1, where the probability of returning in three steps increases from 7.3 to 16.4% when increasing the maximum order from two to three in HON. By increasing the maximum order to five, HON can capture all dependencies below or equal to the fifth order, and the accuracy of simulating one step on HON doubles that of the conventional first-order network.

Furthermore, when simulating multiple steps, the advantage of using HON is even bigger. The reason is that in a first-order network, a random walker "forgets" where it came from after each step and has a higher chance of disobeying higher-order movement patterns. This error is amplified quickly in a few steps—the accuracy of simulating three steps on the first-order network is almost zero. On the contrary, in HON, the higher-order nodes and edges can help the random walker remember where it came from and provide the corresponding probability distributions for the next step. As a consequence, the simulation of three steps on HON is one magnitude more accurate than on first-order network. This indicates that, when multiple steps are simulated (which is usually seen in methods such as PageRank and MapEquation that need multiple iterations), using HON (instead of the conventional first-order network) can help random walkers simulate movements more accurately; thus, the results of all random walking–based network analysis methods will be more reliable.

Effects on clustering

One important family of network analysis methods is clustering, which identifies groups of nodes that are tightly connected. A variety of clustering algorithms, such as MapEquation (*38*) and Walktrap (*39*), are based on random walking, following the intuition that random walkers are more likely to move within the same cluster rather than between different clusters. Because using HON instead of a first-order network alters the movement patterns of random walkers on the network, a compelling question becomes: How does HON affect the clustering results?

Consider an important real-world application of clustering: identifying regions wherein aquatic species invasions are likely to happen. Because the global shipping network is the dominant global vector for the unintentional translocation of non-native aquatic species (*45*) [species get translocated either during ballast water uptake/discharge or by accumulating on the surfaces of ships (*11*)], identifying clusters of ports that are tightly coupled by frequent shipping can reveal ports that are likely to introduce non-native species to each other. The limitation of the existing approach (*10*) is that the clustering is based on a first-order network that only accounts for direct species flows, whereas in reality the species introduced to a port by a ship may also come from multiple previous ports at which the ship has stopped because of partial ballast water exchanges and hull fouling. These indirect species introduction pathways driven by ship movements are already captured by HON and can influence the clustering result. As represented by the HON example in Fig. 1C, following the most likely shipping route, species are more likely to be introduced to Los Angeles from Shanghai (via Singapore) rather than from Tokyo, so the clustering (driven by random walking) on HON prefers grouping Los Angeles with Shanghai rather than with Tokyo. In comparison, indirect species introduction pathways are ignored when performing clustering on a first-order network (Fig. 1B), thus underestimating the risk of invasions via indirect shipping connections.

By clustering on HON, the overlap of different clusters is naturally revealed, highlighting ports that may be invaded by species from multiple regions. Because there can be multiple nodes representing the same physical location in HON (for example, both *Singapore|Tokyo* and *Singapore|Shanghai* represent Singapore) and the ship movements through these nodes can be different, these higher-order nodes can belong to different clusters, so that Singapore as an international port belongs to multiple clusters, as one would expect.

The clustering results (using MapEquation) on a first-order network and HON are compared in Fig. 3. For example, let us consider

Malta, a European island country in the Mediterranean Sea. Malta has two ports: Valletta is a small port that mainly serves cruise ships in the Mediterranean, and Malta Freeport, on the contrary, is one of the busiest ports in Europe (many international shipping routes have a stop there). The clustering on the first-order network cannot tell the difference between the two ports and assigns both to the same Southern Europe cluster. On the contrary, the clustering on HON effectively separates Valletta and Malta Freeport by showing that Malta Freeport belongs to three additional clusters than Valletta, implying long-range shipping connections and species exchanges with ports all over the world. In summary, on HON, 45% of ports belong to more than one cluster, among which the Panama Canal belongs to six clusters, and 44 ports (1.7% of all) belong to as many as five clusters, including international ports such as New York, Shanghai, Hong Kong, Gibraltar, Hamburg, and so on, indicating challenges to the management of aquatic invasions, as well as opportunities for devising targeted management policies. These insights are gained by adopting HON as the network representation for the global shipping data, whereas the MapEquation algorithm is unmodified.

Effects on ranking

Another important family of network analysis methods is ranking. PageRank (*37*) is commonly used in assessing the importance of Web pages by using random walkers (with random resets) to simulate users clicking through different pages, and pages with higher PageRank scores have higher chances of being visited. It has been shown that Web users are not Markovian (*14*), and PageRank on the conventional network representation fails to simulate real user traffic (*46*). Because HON can help random walkers achieve higher accuracies in reproducing movement patterns, how can HON affect the PageRank scores, and why?

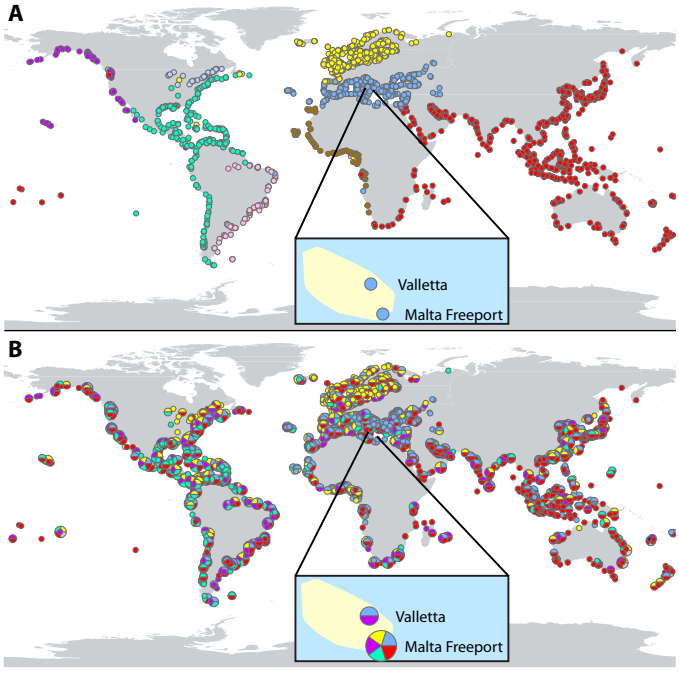

Fig. 3. Clustering of ports on different network representations of the global shipping data. Ports tightly coupled by frequent shipping in a cluster are likely to introduce non-native species to each other. MapEquation (*38*) is used for clustering, and different colors represent different clusters. (**A**) Clustering on the first-order network. Although Valletta and Malta Freeport are local and international ports, respectively, the clustering result does not distinguish the two. (**B**) Clustering on HON. The overlapping clusters indicate how international ports (such as Malta Freeport) may suffer from species invasions from multiple sources.

With the clickstream data, we can construct both a first-order network and a HON as the input for PageRank. In HON, the PageRank scores of multiple higher-order nodes representing the same Web page are summed up as the final score for the page. As shown in Fig. 4, by using HON instead of the first-order network, 26% of the Web pages show more than 10% of relative changes in ranking; more than 90% of the Web pages lose PageRank scores, whereas the other pages show remarkable gains in scores. To have an idea of the changes, we list the Web pages that gain or lose the most scores by using HON as the input to PageRank, as shown in table S1. Of the 15 Web pages that gain the most scores from HON, 6 are weather forecasts and 4 are obituaries, as one would expect considering that this data set is from Web sites of local newspapers and TVs. Of the 15 Web pages that lose the most scores, 3 are the lists of news personnel under the "about" page, which a normal reader will rarely visit, but are overvalued by ranking on the first-order network.

To further understand how the structural differences of HON and the first-order network lead to changes in PageRank scores, we choose Web pages that show significant changes in ranking and compare the corresponding subgraphs of the two network representations. A typical example is a pair of pages, *PHOTOS: January 17th snow - WDBJ7 / news* and *View/Upload your snow photos - WDBJ7 / news*—these two pages gain 131 and 231% PageRank scores, respectively, on HON. In the first-order network representation, as shown in Fig. 5A, where edge widths indicate the transition probabilities between Web pages, it appears that after viewing or uploading the snow photos, a user is very likely to go back to the WDBJ7 home page immediately. However, in reality, once a user views and uploads a photo, the user is likely to repeat this process to upload more photos while less likely to go back to the home page. This natural scenario is completely ignored in the first-order network but captured by HON, indicated by the strong loop between the two higher-order nodes (Fig. 5B). The example also shows how the higher probability of returning after two (or more) steps on HON can affect the ranking results. Again, all these insights are gained by using HON instead of the conventional first-order network, without any change to the PageRank algorithm. Besides the ranking of Web pages, HON may also influence many other applications of ranking, such as citation ranking and key phrase extraction.

Scalability of HON

We further show the scalability of HON, derived from its compact representation. In previous research (where a fixed second order is assumed for the network), from Table 1, it is shown that the network is considerably larger than the conventional first-order network, and assuming a fixed order beyond the second order becomes impractical because "higher-order Markov models are more complex" (*12*), due to combinatorial explosion. A network that is too large is computationally expensive to perform further analysis upon. On the contrary, although HON with maximum order of two has comparable accuracies in terms of random walking movement simulation, it has less nodes and about half the number of edges compared with a fixed second-order network,

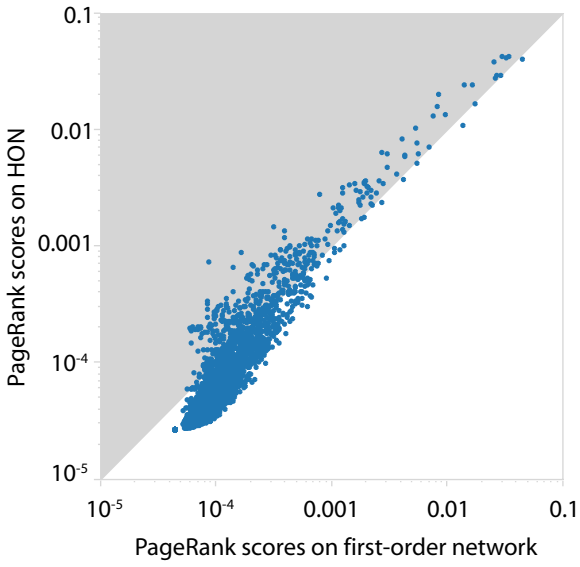

Fig. 4. Change of Web page rankings by using HON instead of first-order network. PageRank (*37*) is used for ranking. Twenty-six percent of the pages show more than 10% of relative changes in ranking. More than 90% of the Web pages lose PageRank scores, whereas the other pages show remarkable gain in scores. Note that log-log scale is used in the figure, so a deviation from the diagonal indicates a significant change of the PageRank score.

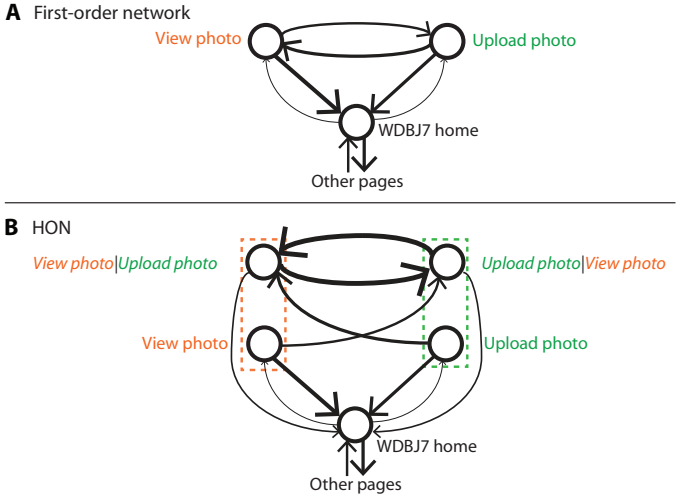

Fig. 5. Comparison of different network representations for the same clickstream data. Edge widths indicate the transition probabilities. (**A**) First-order network representation, indicating that a user is likely to go back to the home page after viewing or uploading snow photos. (**B**) HON representation, which not only preserves the information in the first-order network but also uses higher-order nodes and edges to represent an additional scenario: once a user views and uploads a photo, the user is likely to repeat this process to upload more photos and is less likely to go back to the home page. Consequently, these photo viewing and uploading pages will receive higher PageRank scores (*37*) because the implicit random walkers of PageRank are more likely to be trapped in the loop of the higher-order nodes.

because it uses the first order whenever possible and embeds second-order dependencies only when necessary. Even when increasing the maximum order to five, HON still has less edges than the fixed second-order network, whereas all the useful dependencies up to the fifth order are incorporated in the network, resulting in considerably higher accuracies on random walking simulations.

Another important advantage of HON over a fixed-order network is that network analysis algorithms can run faster on HON because of HON's compact representation. In addition, HON is sparser than the fixed-order representation, and many network toolkits are optimized for sparse networks. Table 1 shows the running time of two typical network analysis tasks: ranking [with PageRank (*37*)] and clustering [with MapEquation (*38*)]. Compared with the fixed second-order network, these tasks run almost two times faster on HON with a maximum order of two and about the same speed on HON with a maximum order of five (which embeds more higher-order dependencies and is more accurate).

It is worth noting that the number of additional nodes/edges needed for HON (on top of a first-order network) is determined by the number of higher-order dependencies in the data and that additional size is affected neither by the size of the raw data nor by the density of the corresponding first-order network. For example, even if the first-order network representation of a data set is a complete graph with 1 million nodes, if 100 second-order dependencies exist in the data, HON needs only 100 additional auxiliary second-order nodes on top of the first-order network, rather than making the whole network the second order. Thus, the advantage of HON is being able to effectively represent higher-order dependencies, while being compact by trimming redundant higher-order connections.

DISCUSSION

We have shown that for sequential data with higher-order dependencies, the conventional first-order network fails to represent such dependency patterns in the network structure, and the fixed second-order dependency can become limiting. If the network representation is not truly representative of the original data, then it will invariably lead to unreliable conclusions or insights from network analyses. We develop a new process for extracting higher-order dependencies in the raw data and for building a network (the HON) that can represent such higher-order dependencies. We demonstrate that our novel network representation is more accurate in representing the true movement patterns in data in comparison with the conventional first-order network or the fixed second-order network: for example, when using HON instead of a first-order network to represent the global shipping data, the accuracy is doubled when simulating a ship's next movement on the network and is higher by one magnitude when simulating three steps, because the higher-order nodes and edges in HON can provide more detailed guidance for simulated movements. Besides improved accuracy, HON is more compact than fixed-order networks by embedding higher-order dependencies only when necessary, and thus, network analysis algorithms run faster on HON.

Furthermore, we show that using HON instead of conventional network representations can influence the results of network analysis methods that are based on random walking. For example, on HON, the clustering of ports takes indirect shipborne species introduction pathways into account and naturally produces overlapping clusters

that indicate multiple sources of species invasion for international ports; the ranking of Web pages is corrected by incorporating the higher-order patterns of users' browsing behaviors such as uploading multiple photos. Our work has the potential to influence a wide range of applications, such as improving PageRank for the task of unsupervised key phrase selection in language processing (*47*), because the proposed network representation is consistent with the input expected by various network analysis methods. Because nodes could be split into multiple ones in HON, it may require postprocessing to aggregate the results for interpretation. In the current method, the choice of parameters may influence the structure of the resulting network, so we provide parameter discussions in note S3.

In future work, we look forward to (i) extending the applications of HON beyond the simulation of movements to more dynamic processes such as dynamic network anomaly detection (*48*), and (ii) improving the algorithm by reducing the parameters needed.

MATERIALS AND METHODS

The construction of the HON consists of two steps: rule extraction identifies higher-order dependencies that have sufficient support and can significantly alter a random walker's probability distribution of choosing the next step; then, network wiring adds these rules describing variable orders of dependencies into the conventional first-order network by adding higher-order nodes and rewiring edges. The data structure of the resulting network is consistent with the conventional network representation, so existing network analysis methods can be applied directly without being modified. We use global shipping traffic data as a working example to demonstrate the construction of HON, but it is generalizable to any sequential data.

Rule extraction

The challenge of rule extraction is to identify the appropriate orders of dependencies in data; when building the first-order network, this step is often ignored by simply counting pairwise connections in the data. We define a path as the movement from source node A to target node B, though with nodes that differ from those in a conventional network: a node here can represent a sequence of entities, no longer necessarily a single entity. Then, among those paths, given a source node A containing a sequence of entities $[a_1, a_2,\ldots, a_k]$, if including an additional entity a_0 at the beginning of A can significantly alter the normalized counts of movements (as probability distribution) to target nodes set $\{B\}$, it means $\{B\}$ has a higher-order dependency on $A_{ext} = [a_0, a_1, a_2,\ldots, a_k]$, and paths containing higher-order dependencies like $A_{ext} \rightarrow B$ are defined as rules. Then, a rule like $\mathrm{Freq}([a_0, a_1] \rightarrow a_2) = 50$ can map to an edge in the network in the form of $a_1|a_0 \rightarrow a_2$ with edge weight 50. What are the expectations for the rule extraction process?

First, rules should represent dependencies that are significant. As in fig. S1 (step 3), if the probability distribution of a ship's next step from Singapore is significantly affected by knowing that the ship came from Shanghai to Singapore, there is at least second-order dependency here. On the contrary, if the probability distribution of going to the next port is the same no matter how the ship reached Singapore, there is no evidence for second-order dependency (but third- or higher-order dependencies may still exist and can be checked similarly).

Second, rules should have sufficient support. Only when some pattern happens sufficiently many times can it be considered as a "rule" rather than some random event. Although this requirement of minimum support is not compulsory, not specifying a minimum support will result in a larger and more detailed network representation, and more infrequent routes are falsely considered as patterns, ultimately lowering the accuracy of the representation (see the discussions of parameters in note S3).

Third, rules should be able to represent variable orders of dependencies. In real-world data, such as the global shipping data, different paths can have different orders of dependencies; for example, in Fig. 1D, the next step from Singapore is dependent on Tianjin through the fourth-order path *Tianjin* → *Busan* → *Tokyo* → *Singapore*, as well as on Shanghai through the second-order path *Shanghai* → *Singapore*. When variable orders can coexist in the same data set, the rule extraction algorithm should not assign a fixed order to the data but should be able to yield rules representing variable orders of dependencies.

Following the aforementioned three objectives of rule extraction, it is natural to grow rules incrementally: start with a first-order path, try to increase the order by including one more previous step, and check if the probability distribution for the next step changes significantly (fig. S1, step 3). If the change is significant, the higher order is assumed; otherwise, keep the old assumption of order. This rule-growing process is iterated recursively until (i) the minimum support requirement is not met or (ii) the maximum order is exceeded. The detailed algorithm is given in note S2.

Network wiring

The remaining task is to convert the rules obtained from the last step into a graph representation. It is trivial for building conventional first-order networks because every rule is first-order and can directly map to an edge connecting two entities, but such direct conversion will not work when rules representing variable orders of dependencies coexist. The reason is that during rule extraction, only the last entity of every path is taken as the target node, so that every edge points to a first-order node, which means higher-order nodes will not have in-edges. Rewiring is needed to ensure that higher-order nodes will have incoming edges, while preserving the sum of edge weights in the network. The detailed steps are illustrated as follows:

1. Converting all first-order rules into edges. This step is exactly the same as constructing a first-order network, where every first-order rule (a path from one entity to another) corresponds to a weighted edge. As illustrated in fig. S2A, *Shanghai* → *Singapore* is added to the network.

2. Converting higher-order rules. In this step, higher-order rules are converted to higher-order edges pointing out from higher-order nodes (the nodes are created if they do not already exist in the network). Figure S2B shows the conversion of rules *Singapore*|*Shanghai* → *Los Angeles* and *Singapore*|*Shanghai* → *Seattle*, where the second-order node *Singapore*|*Shanghai* is created and two edges pointing out from the node are added.

3. Rewiring in-edges for higher-order nodes. This step preserves the sum of edge weights while solving the problem that higher-order nodes have no incoming edges, by pointing existing edges to higher-order nodes. When adding the second-order node *Singapore*|*Shanghai*, a lower-order rule and the corresponding edge *Shanghai* → *Singapore* are guaranteed to exist, because during rule extraction when a rule is added, all preceding steps of the path are also added, as in AddToRules in algorithm S1. As shown in fig. S2C, the edge from *Shanghai* to *Singapore* is redirected to *Singapore*|*Shanghai*. Converting higher-order

rules (step 2) and rewiring (step 3) are repeated for all rules of first order, then second order, and likewise up to the maximum order, to guarantee that edges can connect to nodes with the highest possible orders. This step also implies that any two nodes that represent the same physical location will not have incoming edges from the same node.

4. Rewiring edges built from *Valid* rules. This step, after representing all rules as edges in HON, is necessary due to the fact that the rule extraction step takes only the last entity of paths as targets, such that edges built from *Valid* rules in algorithm S1 always point to first-order nodes. In fig. S2D, the node *Singapore*|*Shanghai* was pointing to a first-order node *Seattle*. However, if a node of higher order, *Seattle*|*Singapore*, already exists in the network, the edge *Singapore*|*Shanghai* → *Seattle* should point to *Seattle*|*Singapore*; otherwise, the information about previous steps is lost. To preserve as much information as possible, the edges built from *Valid* rules should point to nodes with the highest possible orders.

Following the above process, the algorithm for network wiring is given in algorithm S2 in note S2, along with more detailed explanations. Given a set of parameters, the result of HON is unique, so there is no optimization or greedy methods for the algorithm. Note that we have made the entire source code available as well (at https://github.com/xyjprc/hon).

Comparison with related methods

Although VOM models can be used on sequential data to learn a VOM tree (*22*–*24*) for predictions (*49*), our goal is to build a more accurate network representation that captures higher-order dependencies in the data. Although these two objectives are related, there are several key differences: (i) a VOM tree contains probabilities that are unnecessary (for example, nodes that are not leaves) for representing higher-order dependencies in a network; (ii) additional conditional probabilities are needed to connect nodes with different orders in HON, which are not guaranteed to exist in a pruned VOM tree; and (iii) VOM usually contains lots of unnecessary edges because of the "smoothing" process for the unobserved data, which is not desired for a network representation. Therefore, our work is not simply contained in a VOM implementation. Note S4 elaborates the differences and provides an empirical comparison between the HON and VOM.

Although a fixed kth-order Markov model can be directly converted to a first-order model (*12*), the state space S^k grows exponentially with the order. There has been plenty of research on Markov order estimation to determine the order k, such as using different information criteria (*26*, *27*), cross-validation (*14*), and surrogate data (*28*), but these approaches produce a single global order for the model rather than variable orders, and no discussion was given to network representation. Other Markov-related works, such as hidden Markov model (*25*), frequent pattern mining (*20*), and next-location prediction (*21*) focus more on the stochastic process, rather than the network representation problem. For example, the hidden states in hidden Markov model do not represent clear dependency relationships like the higher-order nodes do in HON, and we are not learning a hidden layer that have "emission probabilities" to observations. From the network perspective, although there have been efforts to incorporate HON structures for clustering (*34*, *35*), ranking (*36*), and so on, these methods are modifications of existing algorithms and are application-specific; instead, we embed higher-order dependencies into the network structure, so that the wide range of existing network analysis tools can be applied without modification.

SUPPLEMENTARY MATERIALS

Supplementary material for this article is available at http://advances.sciencemag.org/cgi/content/full/2/5/e1600028/DC1

note S1. Data sets.
note S2. Algorithms.
note S3. Parameter discussion.
note S4. Empirical comparison with the VOM model.
fig. S1. Rule extraction example for the global shipping data.
fig. S2. Network wiring example for the global shipping data.
fig. S3. Parameter sensitivity of HON in terms of accuracy and network size.
fig. S4. Comparison between the HON and the VOM model.
table S1. Changes of PageRank scores by using HON instead of a first-order network.
table S2. Comparison of the number of rules extracted from HON and VOM.
algorithm S1. The rule extraction algorithm.
algorithm S2. The network wiring algorithm.
References (*50*, *51*)

REFERENCES AND NOTES

1. G. Chowell, J. M. Hyman, S. Eubank, C. Castillo-Chavez, Scaling laws for the movement of people between locations in a large city. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* **68**, 066102 (2003).
2. A. De Montis, M. Barthélemy, A. Chessa, A. Vespignani, The structure of interurban traffic: A weighted network analysis. *Environ. Plann. B Plann Des.* **34**, 905–924 (2007).
3. G. Bagler, Analysis of the airport network of India as a complex weighted network. *Physica A* **387**, 2972–2980 (2008).
4. M. E. J. Newman, Analysis of weighted networks. *Phys. Rev. A* **70**, 056131 (2004).
5. A. Barrat, M. Barthélemy, R. Pastor-Satorras, A. Vespignani, The architecture of complex weighted networks. *Proc. Natl. Acad. Sci. U.S.A.* **101**, 3747–3752 (2004).
6. C. Song, Z. Qu, N. Blumm, A.-L. Barabási, Limits of predictability in human mobility. *Science* **327**, 1018–1021 (2010).
7. M. E. J. Newman, Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Phys. Rev. E* **64**, 016132 (2001).
8. A. Markov, *Theory of Algorithms*, J. J. Schorr-Kon and PST staff, Transl. (Imprint Moscow, Academy of Sciences of the USSR, Moscow, 1954).
9. P. Kaluza, A. Kölzsch, M. T. Gastner, B. Blasius, The complex network of global cargo ship movements. *J. R. Soc. Interface* **7**, 1093–1103 (2010).
10. J. Xu, T. L. Wickramarathne, N. V. Chawla, E. K. Grey, K. Steinhaeuser, R. P. Keller, J. M. Drake, D. M. Lodge, in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2014), pp. 1699–1708.
11. J. M. Drake, D. M. Lodge, Global hot spots of biological invasions: Evaluating options for ballast–water management. *Proc. Biol. Sci.* **271**, 575–580 (2004).
12. M. Rosvall, A. V. Esquivel, A. Lancichinetti, J. D. West, R. Lambiotte, Memory in network flows and its effects on spreading dynamics and community detection. *Nat. Commun.* **5**, 4630 (2014).
13. P. Singer, D. Helic, B. Taraghi, M. Strohmaier, Detecting memory and structure in human navigation patterns using Markov chain models of varying order. *PLOS One* **9**, e102070 (2014).
14. F. Chierichetti, R. Kumar, P. Raghavan, T. Sarlos, in *Proceedings of the 21st International Conference on World Wide Web* (ACM, New York, 2012), pp. 609–618.
15. M. Deshpande, G. Karypis, Selective Markov models for predicting web page accesses. *ACM T. Internet Techn.* **4**, 163–184 (2004).
16. T. Takaguchi, M. Nakamura, N. Sato, K. Yano, N. Masuda, Predictability of conversation partners. *Phys. Rev. X* **1**, 011008 (2011).
17. G. Janacek, Time series analysis forecasting and control. *J. Time Ser. Anal.* **31**, 303 (2010).
18. J. D. Hamilton, *Time Series Analysis* (Princeton Univ. Press, Princeton, NJ, 1994), vol. 2.
19. C. E. Shannon, A mathematical theory of communication. *ACM SIGMOBILE Mobile Comput. Commun. Rev.* **5**, 3–55 (2001).
20. J. Han, J. Pei, Y. Yin, in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, *SIGMOD '00* (ACM, New York, 2000), pp. 1–12.
21. A. Monreale, F. Pinelli, R. Trasarti, F. Giannotti, in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2009), pp. 637–646.
22. P. Bühlmann, A. J. Wyner, Variable length Markov chains. *Ann. Statist.* **27**, 480–513 (1999).
23. A. Shmilovici, I. Ben-Gal, Using a VOM model for reconstructing potential coding regions in EST sequences. *Computation Stat.* **22**, 49–69 (2007).
24. I. Ben-Gal, A. Shani, A. Gohr, J. Grau, S. Arviv, A. Shmilovici, S. Posch, I. Grosse, Identification of transcription factor binding sites with variable-order Bayesian networks. *Bioinformatics* **21**, 2657–2666 (2005).

25. L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **77**, 257–286 (1989).
26. G. Schwarz, Estimating the dimension of a model. *Ann. Stat.* **6**, 461–464 (1978).
27. H. Akaike, A new look at the statistical model identification. *IEEE Trans. Automat. Contr.* **19**, 716–723 (1974).
28. M. J. van der Heyden, C. G. C. Diks, B. P. T. Hoekstra, J. DeGoede, Testing the order of discrete Markov chains using surrogate data. *Physica D* **117**, 299–313 (1998).
29. M. F. Heath, M. C. Vernon, C. R. Webb, Construction of networks with intrinsic temporal structure from UK cattle movement data. *BMC Vet. Res.* **4**, 11 (2008).
30. I. Scholtes, N. Wider, R. Pfitzner, A. Garas, C. J. Tessone, F. Schweitzer, Causality-driven slow-down and speed-up of diffusion in non-Markovian temporal networks. *Nat. Commun.* **5**, 5024 (2014).
31. I. Scholtes, N. Wider, A. Garas, Higher-order aggregate networks in the analysis of temporal networks: Path structures and centralities. arXiv preprint arXiv:1508.06467 (2015).
32. M. T. Schaub, J. Lehmann, S. N. Yaliraki, M. Barahona, Structure of complex networks: Quantifying edge-to-edge relations by failure-induced flow redistribution. *Network Sci.* **2**, 66–89 (2014).
33. M. De Domenico, A. Solé-Ribalta, S. Gómez, A. Arenas, Navigability of interconnected networks under random failures. *Proc. Natl. Acad. Sci. U.S.A.* **111**, 8351–8356 (2014).
34. A. R. Benson, D. F. Gleich, J. Leskovec, *Proceedings of the 2015 SIAM International Conference on Data Mining* (SIAM, Vancouver, British Columbia, Canada, 2015), pp. 118–126.
35. C. C. Klymko, D. Gleich, T. G. Kolda, Using triangles to improve community detection in directed networks. arXiv preprint arXiv:1404.5874 (2014).
36. D. F. Gleich, L.-H. Lim, Y. Yu, Multilinear PageRank. *SIAM J. Matrix Anal. & Appl.* **36**, 1507–1541 (2015).
37. L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66 (Stanford InfoLab, Stanford, CA, 1999).
38. M. Rosvall, C. T. Bergstrom, Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. U.S.A.* **105**, 1118–1123 (2008).
39. P. Pons, M. Latapy, Computing communities in large networks using random walks. *J. Graph Algorithms Appl.* **10**, 191–218 (2006).
40. F. Fouss, A. Pirotte, J. M. Renders, M. Saerens, Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Kwol. Data En.* **19**, 355–369 (2007).
41. L. Backstrom, J. Leskovec, in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining* (WSDM '11) (ACM, New York, 2011), pp. 635–644.
42. J.-C. Delvenne, S. N. Yaliraki, M. Barahona, Stability of graph communities across time scales. *Proc. Natl. Acad. Sci. U.S.A.* **107**, 12755–12760 (2010).
43. C. Berge, *Hypergraphs: Combinatorics of Finite Sets* (Elsevier, New York, 1984) vol. 45.
44. A. Vespignani, Modelling dynamical processes in complex socio-technical systems. *Nat. Phys.* **8**, 32–39 (2012).
45. J. L. Molnar, R. L. Gamboa, C. Revenga, M. D. Spalding, Assessing the global threat of invasive species to marine biodiversity. *Front. Ecol. Environ.* **6**, 485–492 (2008).
46. M. R. Meiss, F. Menczer, S. Fortunato, A. Flammini, A. Vespignani, *Proceedings of the 2008 International Conference on Web Search and Data Mining* (ACM, New York, 2008), pp. 65–76.
47. R. Mihalcea, P. Tarau, *Proceedings of Conference on Empirical Methods in Natural Language Processing 2004* (EMNLP, Barcelona, Spain, 2004), pp. 404–411.
48. L. Akoglu, H. Tong, D. Koutra, Graph based anomaly detection and description: A survey. *Data Min. Kowl. Disc.* **29**, 626–688 (2014).
49. R. Begleiter, R. El-Yaniv, G. Yona, On prediction using variable order Markov models. *J. Artif. Intell. Res.* **22**, 385–421 (2004).
50. J. Zhang, B. Liu, J. Tang, T. Chen, J. Li, *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence* (IJCAI '13) (AAAI, Palo Alto, CA, 2013), pp. 2761–2767.
51. S. Kullback, R. A. Leibler, On information and sufficiency. *Ann. Math. Statist.* **22**, 79–86 (1951).

Acknowledgments: We thank D. Lodge, Y. Dong, and R. Johnson for their valuable comments. **Funding:** This work is based on research supported by the University of Notre Dame Office of Research via Environmental Change Initiative (ECI) and NSF awards EF-1427157, IIS-1447795, and BCS-1229450; the research was also supported by the Army Research Laboratory, and was accomplished under Cooperative Agreement no. W911NF-09-2-0053 [the ARL Network Science CTA (Collaborative Technology Alliance)]. **Author contributions:** J.X., T.L.W., and N.V.C. collectively conceived the research. J.X., T.L.W., and N.V.C. designed the analyses. J.X. conducted the experiments. J.X., T.L.W., and N.V.C. conducted the analyses. J.X., T.L.W., and N.V.C. wrote the manuscript. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are available in https://github.com/xyjprc/hon or present in the paper and/or the Supplementary Materials. Additional data related to this paper may be requested from the authors.

Submitted 7 January 2016
Accepted 20 April 2016
Published 20 May 2016
10.1126/sciadv.1600028

Citation: J. Xu, T. L. Wickramarathne, N. V. Chawla, Representing higher-order dependencies in networks. *Sci. Adv.* **2**, e1600028 (2016).

ScienceAdvances
AAAS

Representing higher-order dependencies in networks
Jian Xu, Thanuka L. Wickramarathne and Nitesh V. Chawla (May 20, 2016)
Sci Adv 2016, 2:.
doi: 10.1126/sciadv.1600028