PAPER • OPEN ACCESS

Measuring the significance of higher-order dependency in networks

To cite this article: Jiaxu Li and Xin Lu 2024 *New J. Phys.* **26** 033032

IOP Publishing New J. Phys. 26 (2024) 033032 https://doi.org/10.1088/1367-2630/ad2db6

New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG
IOP Institute of Physics

Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

OPEN ACCESS

RECEIVED
29 November 2023

REVISED
22 February 2024

ACCEPTED FOR PUBLICATION
27 February 2024

PUBLISHED
19 March 2024

PAPER

Measuring the significance of higher-order dependency in networks

Jiaxu Li and Xin Lu*

College of Systems Engineering, National University of Defense Technology, Changsha, Hunan 410073, People's Republic of China
* Author to whom any correspondence should be addressed.

E-mail: xin.lu.lab@outlook.com

Keywords: higher-order dependencies, higher-order networks, higher-order Markov models, hypothesis testing, significance

Abstract

Higher-order networks (HONs), which go beyond the limitations of pairwise relation modeling by graphs, capture higher-order dependencies involving three or more components for various systems. As the number of potential higher-order dependencies increases exponentially with both network size and the order of dependency, it is of particular importance for HON models to balance their representation power against model complexity. In this study, we propose a method, significant k-order dependencies mining (SkDM), based on hypothesis testing and the Markov chain Monte Carlo (MCMC), to identify significant higher-order dependencies in real systems. Through synthetic clickstreams with elaborately designed higher-order dependencies, SkDM shows a powerful ability to correctly identify all significant dependencies at preset significance levels of $\alpha = 0.01, 0.05, 0.10$, performing as the only method, in comparison to the state of the arts, that can robustly maintain the Type I error rate, and without generating any Type II error across all the experimental settings. We further apply the SkDM method to various empirical networks, including journal citations, air traffic, and email communications. Empirical results show that among those tested networks, only 6.03%, 1.47%, and 1.28% of all potential dependencies are of statistical significance ($\alpha = 0.01$). The proposed SkDM method, therefore, provides an efficient tool for higher-order network analysis tasks at reduced computational complexity.

1. Introduction

Complex networks are essential tools for describing different systems comprising a large number of interacting components, such as biological systems, transportation systems, power systems, and so on [1–3]. Conventionally, components and direct interactions between them are represented as nodes and links, respectively, giving rise to the first-order network (FON) representation. However, traditional network models assuming the Markov property (first-order dependency) capture only pairwise interactions but lose higher-order dependencies involving three or more components in systems [4]. Yet, higher-order dependencies in real-world systems are ubiquitous, sequential and better explain how the components, directly and indirectly, influence each other [5–8]. For example, in a shipping traffic system, ports that a ship has arrived at in the past may heavily influence the ship's next destination [5]. In the citation analysis of publications, the outputs of citation flow from a journal depend on where the citations come from [9, 10]. In Web user clickstreams, the user's next page visit is affected by previous clicks [5]. As a result, traditional network modeling techniques only provide a limited representation of complex systems. In contrast, higher-order network models (HONs) that better capture many-body interactions can improve the analysis of various network analysis tasks [11–17]. Recent work has developed four different lines of modeling approaches to embed higher-order dependencies into HON models, including hypergraph models [11, 18–22], simplicial complex models [16, 23–30], motif-based higher-order models [31–38] and higher-order Markov models [5, 6, 9, 39–43].

The first three models capture many-body interactions using higher-order structures such as hyperedges, simplices, and motifs and uncover the higher-order dynamics in complex systems, such as percolation,

diffusion, contagion and synchronization processes. In hypergraphs, nodes represent components and hyperedges represent many-body interactions. The size of a hyperedge is the number of components in the corresponding many-body interaction. Many studies have shown that hypergraphs have an essential influence on dynamical processes [18–22]. For example, hypergraphs could enhance the synchronization in the coupled oscillator systems using a Kuramoto model and combinatorial Laplace operators [18–20]. Alvarez-Rodriguez *et al* [21] investigated the evolutionary dynamics of public goods games in social systems using hypergraphs, demonstrating how hubs and higher-order interactions influence the evolution of cooperation in systems. Further, Kumar *et al* [22] introduced many-body interactions into sender-receiver games, revealing that moral behavior could evolve on higher-order social networks, even when facing the temptation to lie.

In simplicial complexes, simplices describe many-body interactions between components in a complex system [12]. Compared to hypergraphs, simplicial complexes have a restrictive constraint that given a simplex, all its sub-simplices must exist in the system [23, 24]. Many studies have investigated the influence of simplicial complexes on network dynamical processes [16, 25–30]. Chen *et al* [16] proposed an extended bond percolation model on simplicial complexes, revealing that synergistic protection could enhance network robustness. In order to describe the effects of many-body interactions in contagion processes, the mean field approach (MFA) [25], the microscopic Markov chain approach (MMCA) [27, 28] and epidemic link equations (ELE) [28] have been applied to obtain analytical results of simplicial contagion models (SCMs). Results have shown that many-body interactions could lead to a discontinuous phase transition and time-varying higher-order simplices impede contagion processes [25–28]. Parastesh *et al* [29] investigated the synchronization process of a simplicial complex of Hindmarsh–Rose neurons, demonstrating that weaker second-order interactions can significantly reduce synchronization costs. Gao *et al* [30] proposed a reaction-diffusion model embedded in simplicial complexes to study the formation of Turing patterns, revealing a strong correlation between the structure of Turing patterns and the average degree of higher-order connections in simplicial complexes.

Dense subgraphs, called motifs, could also encode higher-order connectivity patterns. A generalized higher-order network analysis framework based on motifs has been developed, such as motif-based spectral clustering algorithms [31–33], higher-order clustering coefficients [34, 35], algorithms for fast counting temporal motifs [36], higher-order link prediction frameworks [37, 38], higher-order motif closures [38] and so on. Results have shown that motif-based higher-order modeling framework provided a foundation for developing network analysis methods.

The fourth type of method for HON modeling is higher-order Markov models, which extract k-order dependencies from sequential data based on higher-order Markov chains and demonstrate that such patterns with memory effects would affect various network analysis tasks [6]. To fully keep into account higher-order dependencies of real systems, many modeling frameworks have been proposed, such as second-order Markov network models [9, 39], the kth order aggregate network model [40], the sparse memory network model [41], the multi-order graphical model [42], the BuildHON model [5], the BuildHON+ model [44], the generative multi-order model [43] and so on. For example, based on the Kullback–Leibler divergence and entropy rates, the BuildHON model, which can overcome the limitations of fixed-order networks, incorporated variable orders of dependencies in systems [5]. Compared with the first-order network, it yields more accurate results on random walking, clustering and ranking. Furthermore, BuildHON+ was proposed and it outperformed other state-of-the-art algorithms in node classification [45] and anomaly detection [44] of sequential data.

Most of the research on HONs has only shown their high representative power and accuracy compared to FONs. Although these models successfully capture higher-order dependencies in complex systems, due to the increase of both network size and the order of dependency, current studies have been constrained to exponentially growing higher-order dependencies. As the representative power of a HON model increases, the complexity of network analysis tasks increases. Thus, it is critical for researchers to design an algorithm to select and embed the most significant higher-order dependencies into HON models to reduce a model's complexity. As far as we know, there is no generic mechanism to identify significant higher-order dependencies in real systems. In this work, we propose a framework, significant k-order dependencies mining (SkDM), based on the hypothesis testing and Markov chain Monte Carlo (MCMC) method, to focus on significant higher-order dependencies: recurring, significant, dependent patterns. We implement SkDM on two elaborately designed synthetic clickstream datasets with variable orders of dependencies—the CK1 dataset and the CK2 dataset—to validate the effectiveness of the proposed method. Furthermore, we use SkDM to extract significant higher-order dependencies from three real-world datasets covering journal citations, air traffic, and email communications—the APS dataset, the DB1B dataset, and the Enron Email dataset—to investigate the impacts of higher-order dependencies on network flow patterns.

2. Methods

2.1. Higher-order Markov network model

A system (V, S) is defined with $S = \{p_1, p_2, \cdots, p_N\}$ the set of sequences and $V = \{v_0, v_1, \ldots, v_{m-1}\}$ the component set, where $p_i = (v_\alpha \to \cdots \to v_\beta)$ for $v_\alpha, v_\beta \in V$ is an ordered record of $l_i + 1$ vertices. Based on a first-order Markov model, the system could be represented as a graph $G^{(1)} = \left(V^{(1)}, E^{(1)}\right)$ with first-order nodes $V^{(1)}$ and directed edges $E^{(1)} \subseteq V^{(1)} \times V^{(1)}$, where $(v_i, v_{i+1}) \in E^{(1)}$ for $i \in [0, l_i - 1]$ is defined in p_i. A k-1th order dependency with k nodes extracted from sequential data S could be defined as $I = [v_\alpha, \ldots, v_\beta]$ for $v_\alpha, v_\beta \in V$, aggregated in the set $\Gamma = \{I_0, I_1, \ldots, I_n\}$. Based on higher-order Markov chains, real systems (V, S, Γ) could naturally be represented as conditional probability models [42, 46].

First-order Markov network model. First-order networks rest on the assumption of Markovian property (first-order Markov process) which is memoryless-a random walker's next destination only depends on the currently visited component and is independent of its history [47, 48]. For example, when a walker is a web user, the first-order Markov model for user navigational behaviors assumes that the next page they visit depends solely on the current page and not on historical clickstreams. Given a real system (V, S), we view each component as a state and direct interactions between components as potential transitions in a standard Markovian network model. The model can be written as follows:

$$P\left(v_{i+1} | v_0 \to v_1 \to \cdots \to v_i\right) = P\left(v_{i+1} | v_i\right), \tag{1}$$

where $P\left(v_{i+1} | v_i\right)$ is the transition probability of a walker moving from first-order node v_i to first-order node v_{i+1}. The first-order transition probability matrix $P^{(1)}$ is given by

$$P^{(1)} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1|V^{(1)}|} \\ p_{21} & p_{22} & \cdots & p_{2|V^{(1)}|} \\ \vdots & \vdots & \ddots & \vdots \\ p_{|V^{(1)}|1} & p_{|V^{(1)}|2} & \cdots & p_{|V^{(1)}||V^{(1)}|} \end{bmatrix}, \tag{2}$$

$$p_{ij} = p\left(v_i \to v_j\right) = \frac{W\left(v_i \to v_j\right)}{\sum\limits_k W\left(v_i \to v_k\right)}, \tag{3}$$

where $\left|V^{(1)}\right|$ represents the number of elements in $V^{(1)}$ and the transition probability from first-order node v_i to first-order node v_j is proportional to the edge weight $W(v_i \to v_j)$.

***k*-order Markov network model.** The properties of the first-order Markov model indicates that it would be too simplistic for capturing higher-order dependencies that impact network analysis tasks, such as ranking, clustering, and link prediction [5, 41]. Since higher-order Markov chains are processes that could keep the impact of the past few states on the walker's next step, it can be used to capture higher-order dependencies to improve the accuracy of the representation of real systems. For example, a k-order Markov chain model (M_k) holds that a walker's next movement depends on the last k components visited.

In order to embed higher-order dependencies in systems, it is critical to reconstruct the network. A first-order node can be split into different higher-order nodes based on variable-order dependencies [5]. For instance, based on a k-order dependency $v_{i-k+1} \to \cdots \to v_i \to v_{i+1}$, the first-order node v_i can be broken down into a k-order node $v_i|v_{i-1}, \ldots, v_{i-k+1}$ (v_i given the previous k-1 components visited), which contains a series of entities $[v_{i-k+1}, \ldots, v_{i-1}, v_i]$. Noticeably, the k-order node $v_i|v_{i-1}, \ldots, v_{i-k+1}$ belongs to the first-order node v_i. Furthermore, the directed edge from the k-order node $v_i|v_{i-1}, \ldots, v_{i-k+1}$ to the first-order node v_{i+1} represents the k-order dependency $v_{i-k+1} \to \cdots \to v_i \to v_{i+1}$. The HON model with k-order dependencies can be written as follows:

$$P^{(k)} := P\left(v_{i+1} | v_{i-k+1} \to \cdots \to v_i\right) \tag{4}$$

$$= P\left(v_{i+1} | \left(v_i | v_{i-1}, \ldots, v_{i-k+1}\right)\right), \tag{5}$$

$$P\left(v_{i+1} \mid \left(v_i \mid v_{i-1}, \ldots, v_{i-k+1}\right)\right) = \frac{W\left(v_i \mid v_{i-1}, \ldots, v_{i-k+1} \to v_{i+1}\right)}{\sum_j W\left(v_i \mid v_{i-1}, \ldots, v_{i-k+1} \to v_j\right)}, \tag{6}$$

where $P\left(v_{i+1} | v_{i-k+1} \to \cdots \to v_i\right)$ is the probability that a random walker moves from v_i to v_{i+1} based on its (k-1)-step memory. $P\left(v_{i+1} | \left(v_i | v_{i-1}, \ldots, v_{i-k+1}\right)\right)$ stored in the k-order matrix $P^{(k)}$ is the transition probability from node $v_i|v_{i-1}, \ldots, v_{i-k+1}$ to node v_{i+1}, which is proportional to the edge weight $W(v_i|v_{i-1}, \ldots, v_{i-k+1} \to v_{i+1})$.

2.2. Significant *k*-order dependencies mining (S*k*DM)

Higher-order dependencies go beyond pairwise interactions and could better explain how the components directly and indirectly influence each other. Naturally, HON models embedding higher-order dependencies improve their descriptive powers and the accuracy of network analysis tasks [12]. However, the exponentially increased number of potential higher-order dependencies introduces problems such as high computational complexity and 'state space explosion'. It is critical to develop a method to identify significant higher-order dependencies from the system to reduce the complexity of HON models.

We develop a two-tailed statistical test framework, significant *k*-order dependencies mining (S*k*DM), based on hypothesis testing and the MCMC method to select significant higher-order dependencies from systems. The central topic is to focus on significant higher-order dependencies: Connected and sequential patterns occur in real-world complex systems at numbers significantly higher or lower than those in randomized networks generated according to the first-order transition probability matrix. Here, we plan to generate many randomized networks with the same first-order transition probability matrix as the real network: Each node in the randomized network has the same first-order transition probability as the corresponding node in the real network. In order to obtain randomized networks, we first generate 1000 simulation datasets with 10^6 trajectories using the MCMC method [49–51]. Then, we extract *k*-order dependencies from simulation datasets and compute the corresponding *k*-order transition probabilities based on higher-order Markov chains. Furthermore, we propose the hypothesis testing framework for judging whether a *k*-order dependency ϕ is significant. Specifically, the null hypothesis (H_0) and the alternate hypothesis (H_1) can be described as:

H_0: A *k*-order dependency ϕ is not significant such that

$$p_{\text{real}}(\phi) = \overline{p_{\text{random}}(\phi)}, \tag{7}$$

$$\hat{\theta} = \overline{p_{\text{random}}(\phi)}. \tag{8}$$

H_1: A *k*-order dependency ϕ significantly exists in the real system.

$$p_{\text{real}}(\phi) \neq \overline{p_{\text{random}}(\phi)}, \tag{9}$$

where $p_{\text{real}}(\phi)$ is the transition probability of *k*-order dependencies in real systems and $\overline{p_{\text{random}}(\phi)}$ is the sample mean $\hat{\theta}$ of transition probabilities of *k*-order dependencies in simulation datasets.

According to the central limit theorem, the distribution for a sample mean is approximately normally distributed for large simulation datasets [52, 53]. Therefore, we establish a standard normal random variable *z*-score (Z) as a test statistic

$$Z = \frac{\overline{p_{\text{random}}(\phi)} - p_{\text{real}}(\phi)}{SD\left(p_{\text{random}}(\phi)\right)}, \tag{10}$$

$$\sigma_{\hat{\theta}} = SD\left(p_{\text{random}}(\phi)\right), \tag{11}$$

where $SD\left(p_{\text{random}}(\phi)\right)$ is the standard deviation $\sigma_{\hat{\theta}}$ of transition probabilities of *k*-order dependencies in simulation datasets. Then we could derive a confidence interval with a confidence level $1-\alpha$ based on the population mean θ of transition probabilities of *k*-order dependencies in simulation datasets, as shown in equation (11) [52, 54, 55]

$$P\left(-z_{\alpha/2} \leqslant Z \leqslant z_{\alpha/2}\right) = P\left(-z_{\alpha/2} \leqslant \frac{\hat{\theta}-\theta}{\sigma_{\hat{\theta}}} \leqslant z_{\alpha/2}\right) \tag{12}$$

$$= P\left(-z_{\alpha/2} \times \sigma_{\hat{\theta}} + \hat{\theta} \leqslant \theta \leqslant z_{\alpha/2} \times \sigma_{\theta} + \hat{\theta}\right) \tag{13}$$

$$= 1-\alpha. \tag{14}$$

In addition, we introduce the *p*-value to represent the extent that the test statistic disagrees with H_0. The *p*-value is the probability of finding that the value of a test statistic (such as Z) is at least as contradictory to H_0, assuming H_0 is correct [56]. We could reject the null hypothesis when the *p*-value is less than the significance level α.

$$p-\text{value} = P\left(Z \geqslant z_{\alpha/2}\right) + P\left(Z \leqslant -z_{\alpha/2}\right). \tag{15}$$

In this study, we take $z_{0.005} = 2.58$, and the 99% confidence interval ($\alpha = 0.01$) is

$$\left[-2.58 \times \sigma_{\hat{\theta}} + \hat{\theta},\ 2.58 \times \sigma_{\hat{\theta}} + \hat{\theta}\right]. \tag{16}$$

Finally, we decide whether to accept or reject the null hypothesis. Since S*k*DM is of the two-tailed statistical test, both $[-\infty, -2.58 \times \sigma_{\hat{\theta}} + \hat{\theta}]$ and $[2.58 \times \sigma_{\hat{\theta}} + \hat{\theta}, \infty]$ are rejection regions. When the transition probability $p_{\text{real}}(\phi)$ of k-order dependencies in real systems falls in the rejection regions or the absolute value of Z is bigger than 2.58 ($|Z| > z_{0.005}$), we reject the null hypothesis and then obtain the significant k-order dependency ϕ.

2.3. Baselines

Here we compare the performance of S*k*DM with a classical sequential pattern mining method and two higher-order modeling methods: the Prefix-projected Sequential Pattern Mining algorithm (PrefixSpan), the multi-order graphical modeling framework (MON) and BuildHON+.

PrefixSpan quickly mines the complete set of sequential patterns in the database and greatly reduces the computation complexity [57]. It first scans the database to find the frequent 1-sequences and generates the projected database for each frequent 1-sequence. In this way, Prefixspan recursively generates the projected database for each frequent k-sequence to find frequent $(k+1)$-sequences. It has a *minSup* threshold, which represents the minimum support required to be considered a frequent sequential pattern.

MON is a nested structure of multi-order graphical models which could infer higher-order dependencies at multiple lengths and capture topological and temporal characteristics of real-world datasets [42, 43]. For example, two multi-order models $\bar{M}_K$ and $\bar{M}_{K+1}$ incorporate dependencies up to maximum orders K and $K+1$ respectively. $\bar{M}_K$ is considered as the null model, and $\bar{M}_{K+1}$ is the alternative model. The p-value of the null model $\bar{M}_K$ is as follows,

$$p = 1 - \frac{\gamma\left(\frac{d(K+1)-d(K)}{2}, -\log\frac{L(\bar{M}_K|S)}{L(\bar{M}_{K+1}|S)}\right)}{\Gamma\left(\frac{d(K+1)-d(K)}{2}\right)}, \tag{17}$$

where S is the sequential dataset and $d(K)$ are the degrees of freedom of $\bar{M}_K$. The likelihood ratio $\frac{L(\bar{M}_K|S)}{L(\bar{M}_{K+1}|S)}$ represents the relative fitness between the alternative model and the null model, given the observed dataset. Γ is the Euler Gamma function and γ is the lower incomplete gamma function. The algorithm iteratively checks whether the p-value is below a significance threshold ϵ or not. If the p-value is below ϵ, we reject the alternative model $\bar{M}_K$. This process continues until the maximum order K_{opt} is reached, where the p-value exceeds ϵ. Here, we set ϵ as 0.001 ($\epsilon = 0.001$).

BuildHON+ is a scalable and parameter-free algorithm based on the Kullback–Leibler divergence, designed to extract variable and higher-order dependencies from big data [44]. A k-order dependency ϕ could extend to ϕ_{ext} of order $k_{\text{ext}} = k+1$ when the distribution of ϕ_{ext} (D_{ext}) is significantly different from the distribution of ϕ (D). BuildHON+ uses the Kullback–Leibler divergence measuring the difference between ϕ and ϕ_{ext}, as shown in equation (18),

$$\mathcal{D}_{\text{KL}}(D_{\text{ext}} \| D) > \delta = \frac{k_{\text{ext}}}{\log_2(1 + Support(\phi_{\text{ext}}))} \tag{18}$$

where δ is a dynamic threshold and $Support(\phi_{\text{ext}})$ is the number of observations ϕ_{ext}. When $\mathcal{D}_{\text{KL}}(D_{\text{ext}} \| D)$ is bigger than δ, the k-order dependency ϕ could extend to ϕ_{ext} of order k_{ext}.

3. Model validation and comparison

To validate the effectiveness of the proposed method, we apply S*k*DM to two elaborately designed synthetic clickstream datasets containing variable orders of dependencies, the CK1 dataset and the CK2 dataset. The CK1 dataset incorporates only first-order dependencies, while the CK2 dataset includes both first and second-order dependencies.

3.1. Synthetic data

First, we assumed 1000 users navigating through 100 web pages, arranged in a 10 × 10 grid and indexed from 0 to 99. Each page had 2 out-links to neighboring pages, including a down-link and a right-link with wrapping, i.e. a user could move from a rightmost page to the leftmost page in the same row or from a bottom page to the top page in the same column. Each user started from a random page numbered from 0 to 99 and navigated through 100 pages within the given time. As a result, each clickstream dataset contained 1000 records with 100 000 clicks in a period.

We aimed to generate two synthetic clickstream datasets with variable orders of user navigating dependencies, the CK1 and CK2 datasets. The CK1 dataset contained only first-order dependencies: Each user had a 50% chance of moving rightward or downward in the next step. Keeping the previous (first-order)

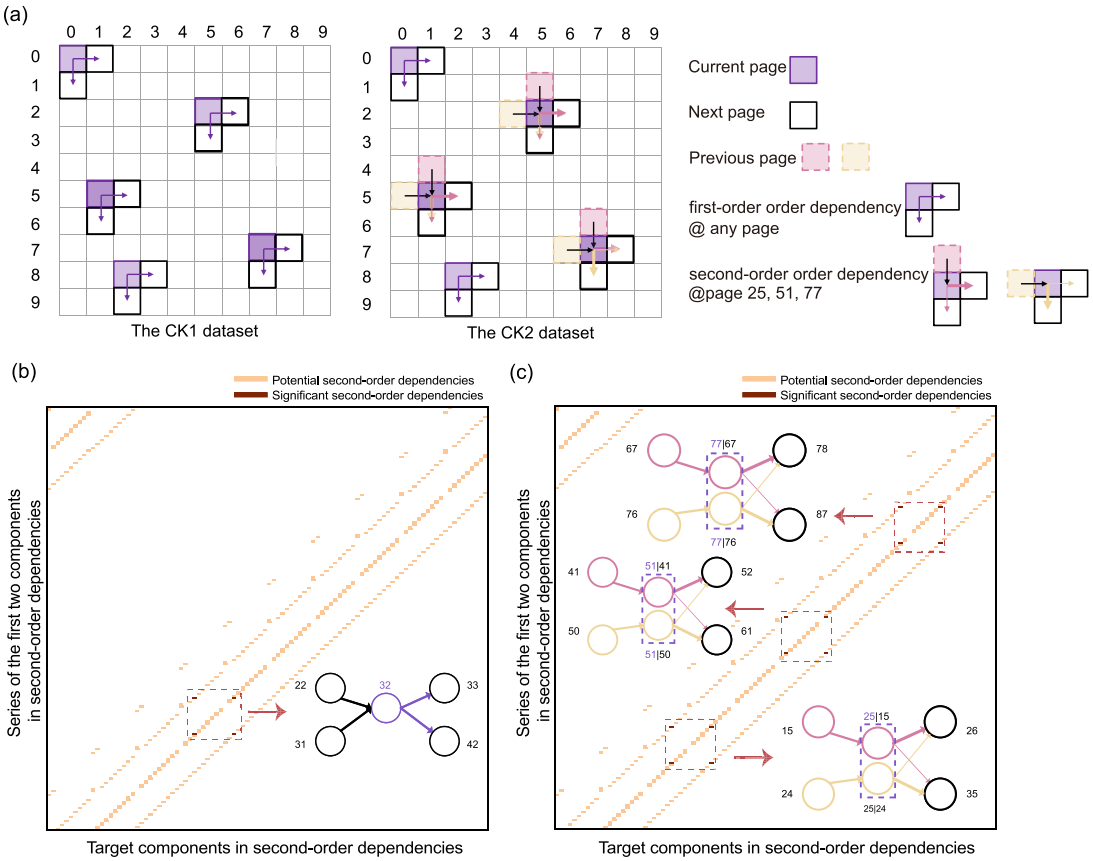

Figure 1. Significant second-order dependencies ($\alpha = 0.01$) extracted from two synthetic clickstream datasets: the CK1 and CK2 datasets. (a) Two synthetic web clickstream datasets, the CK1 and CK2 datasets, with variable orders of user-navigating preferences on 100 web pages as a 10 × 10 grid. The CK1 dataset contains only first-order dependencies: Each user had a 50% chance of navigating to the page on the right and 50% chance of navigating down in the next step. Twelve second-order dependencies on pages 25, 51 and 77 are embedded in the CK2 dataset, such as 15→25→26, 24→25→35, 41→51→52. For example, the second-order dependency 15→25→26 represents that users coming from page 15 to page 25 are more likely to click on page 26 rather than page 35. (b)–(c) The horizontal axis represents the last target component v_k in a second-order dependency $v_i \to v_j \to v_k$. The vertical axis represents the second-order node $v_j|v_i$ which is the series of the first two components in a second-order dependency $v_i \to v_j \to v_k$. For example, the second-order node 51|41 represents page 51 given page 41 as the previous step. S*k*DM extracts 4 significant second-order dependencies (highlighted in saddle brown) out of 400 potential second-order dependencies (highlighted in orange) from the CK1 dataset (see (b)). All 12 significant second-order dependencies are correctly extracted by S*k*DM from the CK2 dataset and represented in the higher-order clickstream network (see (c)).

dependencies, the CK2 dataset introduced 12 second-order dependencies to keep its first-order transition probability matrix $P^{(1)}$ the same as the first dataset. The imposed second-order dependencies were that (1) all users coming from page 24 to 25 (50 to 51, 76 to 77) would have a 10% chance of moving right ($p_d = 0.1$) and a 90% chance of moving down ($1 - p_d = 0.9$) in the next step; (2) all users coming from page 15 to 25 (41 to 51, 67 to 77) would have a 90% chance of moving right ($1 - p_d = 0.9$) and a 10% chance of moving down ($p_d = 0.1$) in the next step; (3) all users coming from page 50 to 51 would have a 10% chance of moving right ($p_d = 0.1$) and a 90% chance of moving down ($1 - p_d = 0.9$) in the next step; (4) all users coming from page 41 to 51 would have a 90% chance of moving right ($1 - p_d = 0.9$) and a 10% chance of moving down ($p_d = 0.1$) in the next step; (5) all users coming from page 76 to 77 would have a 10% chance of moving right ($p_d = 0.1$) and a 90% chance of moving down ($1 - p_d = 0.9$) in the next step; and (6) all users coming from page 67 to 77 would have a 90% chance of moving right ($1 - p_d = 0.9$) and a 10% chance of moving down ($p_d = 0.1$) in the next step. Figure 1(a) shows these defined dependencies in a 10 × 10 grid.

3.2. Model validation

Since our goal is to find significant second-order dependencies in the web clickstream datasets, the null hypothesis (H_0) and alternate hypothesis (H_1) in the selection process of S*k*DM can be described as follows:

H_0: the selected second-order pattern ϕ is not a significant dependency in the dataset;
H_1: the selected second-order pattern ϕ is a significant dependency in the dataset.

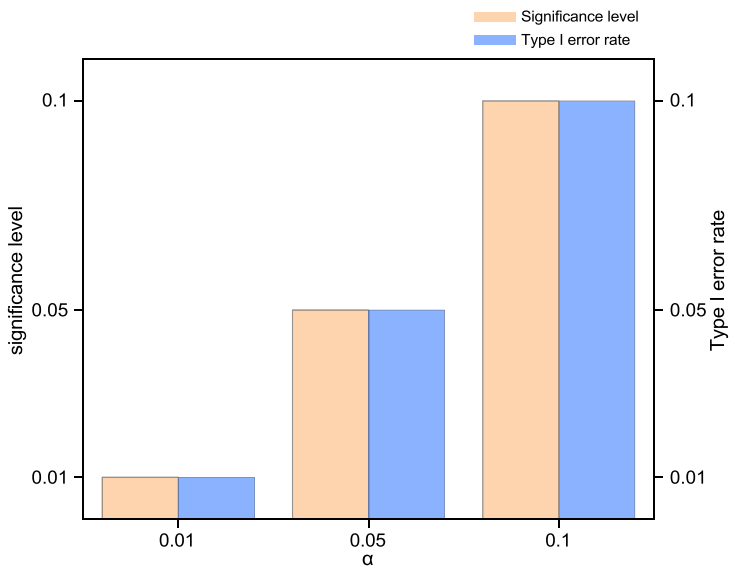

Figure 2. Type I error rates at different significance levels ($\alpha = 0.01, 0.05, 0.10$) by S*k*DM. The horizontal axis represents different values of parameter α. The left y-axis shows the significance level and the right y-axis shows the probability of making a type I error.

Since S*k*DM can yield only one of two outcomes for each second-order pattern, either rejecting or not rejecting H_0, the test could be subject to the type I error. Furthermore, the type I error rate (or significance level) is represented by the probability of rejecting the null hypothesis given that it is true, denoted by α [52].

As shown in figure 1(b), our method captures 4 significant second-order patterns on page 32 at the significance level $\alpha = 0.01$ from the CK1 dataset. However, there are only first-order dependencies in the CK1 dataset: a user currently on page 32 has the same probability of visiting page 33 and page 42 in the CK1 dataset (figure 1(b)). Thus, S*k*DM rejects H_0 (when, in fact, it is true) and makes a type I error:

$$P(TypeIerror) = \alpha = \frac{N(FS)}{N(FS) + N(TNS)}, \tag{19}$$

where $N(FS)$ is the number of false significant dependencies extracted by S*k*DM and $N(TNS)$ is the number of true non-significant patterns extracted by S*k*DM. Since there are 4 false significant second-order dependencies and 396 true non-significant second-order patterns by S*k*DM, $P(TypeIerror)$ is 0.010. The result that the probability of making a type I error is equal to the preset significance level α validates our method.

Furthermore, figure 2 presents the probabilities of making a type I error at different significance levels of $\alpha = 0.01, 0.05, 0.10$ in the CK1 dataset. The results show that S*k*DM could select higher-order dependencies at different significance levels from systems to reduce computational complexity. Furthermore, we establish two measures, the test statistic z-score (denoted as Z) and p-value, to represent the observed significance level of each higher-order dependency (see Materials and Methods). The test statistic Z describes the distance that test results differ from the null hypothesis, whether above or below the mean, measured in units of the standard deviation [52, 56]. The p-value indicates the extent to which the test statistic Z disagrees with H_0 [52]. Using S*k*DM, we could calculate the p-values and z-scores of all higher-order dependencies to indicate their observed significance levels (see figure 6 and tables 4–5 in appendices).

In order to evaluate the ability of S*k*DM to capture significant dependencies, we embed 12 second-order dependencies on pages 25, 51, and 77 in the CK2 dataset; that is, a user's next click depends on not only the current page but also the previous page. For example, the probabilities of a user currently at page 25 moving to page 26 and page 35 are $p_d = 0.10$ and $1 - p_d = 0.90$, respectively, if the previous click is page 24; the probabilities of a user currently at page 25 moving to page 26 and page 35 are $1 - p_d = 0.90$ and $p_d = 0.10$, respectively, if the previous click is page 15. Figure 1(b) shows that our method correctly captures all 12 significant second-order dependencies in the CK2 dataset. Using S*k*DM, we could also calculate the p-values and z-scores of all higher-order dependencies to indicate their observed significance levels (see figure 7 and tables 6–7). The above findings imply that S*k*DM powerfully extracts significant higher-order dependencies to represent true flow patterns in systems.

Table 1. Second-order dependencies captured by four algorithms in the CK2 dataset with different values of p_d. $N(d)$ represents the number of patterns captured by each method. α and β represents the type I error rate and type II error rate of the four methods, respectively.

	$p_d = 0.10$					$p_d = 0.20$					$p_d = 0.30$				
Algorithm	$N(d)$	α	β	*minSup*	Runtime (s)	N_d	α	β	*minSup*	Runtime (s)	N_d	α	β	*minSup*	Runtime (s)
PrefixSpan	12	0.00	0.00	0.37	184.14	12	0.01	0.33	0.38	184.18	12	0.03	0.83	0.35	184.16
MON	400	1.00	0.00	—	928.79	400	1.00	0.00	—	928.78	400	1.00	0.00	—	928.72
BuildHON+	12	0.00	0.00	—	1.20	12	0.00	0.00	—	1.22	0	0.00	1.00	—	1.20
SkDM	**16**	**0.01**	**0.00**	—	**9.80**	**16**	**0.01**	**0.00**	—	**9.88**	**16**	**0.01**	**0.00**	—	**9.92**

Table 2. Second-order dependencies captured by the PrefixSpan algorithm in the CK2 dataset with different values of p_d and *minSup*.

PrefixSpan	$p_d = 0.10$	$p_d = 0.20$	$p_d = 0.30$
$N(d) = N(TS) + N(FS)$	12	12	12
minSup	0.37	0.38	0.35
$N(FS)$	0	4	10
$N(TNS)$	388	388	378
$N(FNS)$	0	4	10
$N(TS)$	12	8	2
$\alpha = \frac{N(FS)}{N(FS)+N(TNS)}$	0.00	0.01	0.03
$\beta = \frac{N(FNS)}{N(FNS)+N(TS)}$	0.00	0.33	0.83

3.3. Model comparison

We compare the performance of S*k*DM with a classical sequential pattern mining method and two higher-order modeling methods: the Prefix-projected Sequential Pattern Mining algorithm (PrefixSpan) [57], the multi-order graphical modeling framework (MON) [42] and the BuildHON+ algorithm [44] (see Baselines). Table 1 represents the performances and average runtimes in seconds of these four algorithms in capturing higher-order dependencies in the CK2 dataset with different values of p_d ($p_d = 0.10, 0.20, 0.30, 1 - p_d = 0.90, 0.80, 0.70$).

Results show that S*k*DM captures 16 significant second-order patterns, of which 12 are true significant dependencies when p_d is 0.10, 0.20 and 0.30, respectively. Since there are 4 false significant dependencies ($N(FS) = 4$) and 384 true non-significant second-order patterns ($N(TNS) = 384$) by S*k*DM with different values of S*k*DM, the type I error rate of S*k*DM is 0.010. Furthermore, we could calculate the type II error rate, which is the probability of failing to reject the null hypothesis when it is actually false, denoted by β

$$P(TypeII\,error) = \beta = \frac{N(FNS)}{N(FNS)+N(TS)}, \tag{20}$$

where $N(FNS)$ is the number of false non-significant patterns extracted by S*k*DM and $N(TS)$ is the number of true significant dependencies extracted by S*k*DM. Since there are 0 false non-significant patterns and 12 true significant second-order dependencies extracted by S*k*DM, the type II error rate of S*k*DM is 0.00.

As the output of PrefixSpan depends on the model parameter *minSup*, we represent the performance of PrefixSpan in capturing higher-order dependencies in the CK2 dataset with different values of p_d and *minSup* in table 2. The *minSup* threshold represents the minimum frequency a pattern must meet to be considered frequent. In order to compare PrefixSpan with other methods, we identify the top 12 frequent patterns as captured patterns. Subsequently, we could obtain the values of *minSup* with different values of p_d. In table 2, PrefixSpan could extract all 12 second-order dependencies from the CK2 dataset when p_d is 0.10, and the type I error rate and type II error rate are both 0.00. It shows that PrefixSpan could capture all dependencies from the dataset when p_d is 0.10. However, as p_d growing up, the ratio of false significant dependencies to the total number of frequent patterns, $\frac{N(FS)}{N(d)}$ also increases. When p_d is 0.30, the type II error rate is as high as 0.83. It shows a large proportion of non-significant patterns among the frequent patterns when p_d is 0.30. These results indicate that PrefixSpan effectively identifies frequent sequential patterns no less than the minimum support threshold. However, higher-order dependencies are sequential patterns with significantly different numbers than random patterns. Furthermore, S*k*DM is approximately twenty times faster than PrefixSpan (see table 1). Consequently, the PrefixSpan algorithm is not optimal for identifying higher-order dependencies.

We observe that the MON method treats all 400 second-order patterns in the CK2 dataset as dependencies when p_d is 0.10, 0.20 and 0.30 (see table 1). MON could extract 388 false significant

second-order dependencies and 0 false non-significant patterns, resulting in a type I error rate of 1.00 and a type II error rate of 0.00 (see table 8 in appendices). Furthermore, S*k*DM is approximately 100 times faster than MON (see table 1).

Table 1 shows that the BuildHON+ method successfully extracts all 12 second-order dependencies when p_d is 0.10 and 0.20. However, BuildHON+ fails to identify any second-order dependencies when p_d is 0.30, resulting in a type I error rate of 0.00 and a type II error rate of 1.00 (see table 9 in Appendices). As we have seen, compared to MON and PrefixSpan, S*k*DM has a relatively low rate of identifying non-significant patterns as higher-order dependencies. Despite being a time-efficient method and demonstrating satisfactory performance on the CK2 dataset at $p_d = 0.10$ and $p_d = 0.20$, BuildHON+ fails to distinguish significant differences between first-order and second-order dependencies at $p_d = 0.30$. In conclusion, S*k*DM is an effective and robust solution to capture significant higher-order dependencies in networked systems.

4. Empirical demonstration and analysis

To study significant higher-order dependencies in real networks using S*k*DM, we collected three public real-world datasets: publications of the American Physical Society (the APS Dataset), flight itineraries between U.S. cities (the DB1B Dataset), and email communications between 146 executives in the Enron Corporation (the Enron Email Dataset). The detailed descriptions of the datasets are as follows.

4.1. Data description

APS Dataset. The American Physical Society (APS) provides rich data based on its publications about physics for research about networks science. The dataset covers 116 years, with 468 291 articles in nine journals and 906 398 citations from 1893 to 2009, and is available at https://journals.aps.org/datasets (table 11) [58]. By modeling citations between journals, we construct the APS citation network and study interdisciplinary knowledge flows. In the APS citation network, a node represents a journal and an edge represents a citation from one journal to another.

DB1B Dataset. The DB1B dataset contains 19 415 369 itineraries between 464 U.S. airports from the first three quarters of 2011 and is available at https://transtats.bts.gov/PREZIP/ [59]. By modeling passengers' itineraries between 413 U.S. cities aggregated from 464 airports, we construct a U.S. air traffic network and study passengers' flight patterns among cities. In the U.S. air traffic network, a node represents a city and an edge represents a passenger's itinerary from one city to another.

Enron Email Dataset. The Enron email dataset (www.cs.cmu.edu/~enron/) contains 116 525 messages generated by 146 senior executives from Enron Corporation disclosed during the investigation of the Enron scandal [60]. By modeling messages, we construct an Enron email network and study the communication patterns between senior executives. In the Enron email network, a node represents a senior executive and an edge represents messages between them.

4.2. Extracting higher-order dependencies in citation flows

The above experiments on CK1 and CK2 show that S*k*DM can correctly identify higher-order dependencies with the preset significance level. Furthermore, we apply S*k*DM to the APS dataset to extract significant higher-order dependencies and investigate citation flows in the network. The APS dataset comprises 468 291 articles published in nine journals and 906 398 citations from 1893 to 2009. figure 3(a) shows significant second-order dependencies extracted from the APS citation network by S*k*DM are highlighted in saddle brown. Table 3 represents the performances and average runtimes in seconds of these four algorithms in capturing higher-order dependencies in the APS dataset. Results show that 44 significant second-order dependencies are filtered out from 729 potential second-order dependencies in the network, with exactly 6.03% (see table 3). S*k*DM is approximately eighty times faster than MON in capturing higher-order dependencies. Therefore, by eliminating non-significant higher-order dependencies from HON models, S*k*DM greatly reduces the computational complexity and improves the efficiency of network analysis tasks.

To show citation flows among journals, we then build eight ego HONs for eight APS journals, simultaneously containing first-order, second-order, and third-order nodes. Figures 3(b)–(i) represents each dependency as a path from a first-order node to a third-order node. For example, the second-order dependency PRA→PRL→PRA is described as path PRA→PRL|PRA→PRA|PRL,PRA from the first-order node PRA through the second-order node PRL|PRA to the third-order node PRA|PRL,PRA. Note that a k-order node $v_i|v_{i-1},\ldots,v_{i-k+1}$ represents the component v_i; for instance, the first-order node PRA and the second-order node PRA|PRL represent the same journal, *Phys Rev A*. As shown in figures 3(b)–(i), we find a 'commuting pattern', that is, citation flows mostly return to journals where they come from. For example, affected by *Phys Rev A*, the citation flow from *Phys Rev Lett* most likely returns to *Phys Rev A* (figure 3(b)). Influenced by *Phys Rev STAB*, there is no network flow from *Phys Rev Lett* to *Phys Rev B*. Yet, 36.81% of

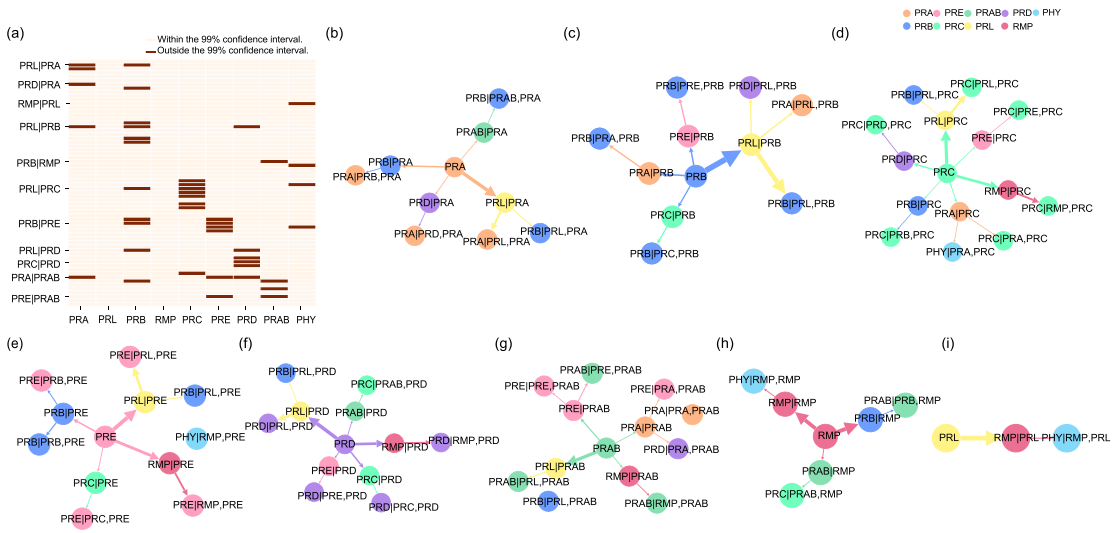

Figure 3. Significant second-order dependencies extracted from the APS citation network. (a) Significant second-order dependencies at the significance level of $\alpha = 0.01$ are highlighted in saddle brown, such as PRA→PRL→PRA. The horizontal axis represents the target journals in second-order dependencies. For example, PRA is the target journal of the second-order dependency PRA→PRL→PRA. The vertical axis represents second-order nodes which are the series of the first two components in second-order dependencies, such as PRL|PRA. (b)–(i) Eight ego HONs composed of significant second-order dependencies. Different colors represent different journals in the APS citation network. Each node is colored based on the journal it indicates. For example, since the second-order node PRL|PRA represents *Phys Rev Lett*, it is marked with yellow. The thickness of an edge is proportional to its weight, representing the number of citations between two nodes.

publications in *Phys Rev Lett* are cited by *Phys Rev B*, ignoring references of *Phys Rev Lett* (see figure 8). These findings indicate that HONs could reflect the true network flows in systems by embedding higher-order dependencies.

4.3. Extracting higher-order dependencies in air traffic

We continue by performing S*k*DM on the U.S. air traffic network to capture significant higher-order dependencies to investigate passengers' flight patterns between different cities using the DB1B dataset. The DB1B dataset contains 19 415 369 itineraries between 464 U.S. airports from the first three quarters of 2011. table 3 shows that 65 212 significant second-order dependencies extracted from the network accounted for 1.47% of all potential 4448 032 second-order dependencies. Moreover, S*k*DM is faster than MON and PrefixSpan in capturing higher-order dependencies. These results demonstrate that our method is extremely efficient at filtering insignificant higher-order dependencies in systems. Figure 4(a) shows a diagonal line and several horizontal zones composed of significant second-order dependencies marked in saddle brown. The diagonal line suggests that passengers often return to the city they departed from; this is the commuting pattern. Several horizontal zones show that passengers influenced by a previously visited city prefer to travel to several fixed cities. For example, using a Sankey diagram, we find that when passengers come from Atlanta (29), they mostly return to Atlanta or travel to other cities such as Chicago (284), Branson (50), Baltimore (69), Detroit (124), Newark (146), Houston (197), and Minneapolis (312; figure 4(b)). Those patterns demonstrate significant second-order dependencies in the air traffic network: a passenger's travel to the next city depends on the currently and previously visited cities.

To embed significant second-order dependencies into the network model, we reconstruct the higher-order U.S. air traffic network containing 405 first-order nodes, 9842 second-order nodes, and 10 676 directed edges with transition probabilities larger than 0.3 (figure 4(c)). Each second-order dependency $v_i \to v_j \to v_k$ is represented as a directed edge $v_j|v_i \to v_k$ from a second-order node $v_j|v_i$ to a first-order node (a target city) v_k. For example, the directed edge from Chicago|Atlanta to Atlanta, Chicago|Atlanta→Atlanta indicates the second-order dependency Atlanta→Chicago→Atlanta. Figure 4(c) shows several clusters centered around different cities (such as Atlanta, Chicago, Los Angeles, Dallas, and Denver). There are many directed edges from second-order nodes to the first-order node in a cluster; this demonstrates that the airports in those cities are busy transportation hubs that serve a large passenger flow in the U.S. The results are consistent with reality: As shown in the list of the busiest airports by passenger traffic distributed by Airports Council International, Hartsfield-Jackson Atlanta International Airport, O'Hare International Airport in Chicago, Los Angeles International Airport, Dallas/Fort Worth International Airport, and Denver

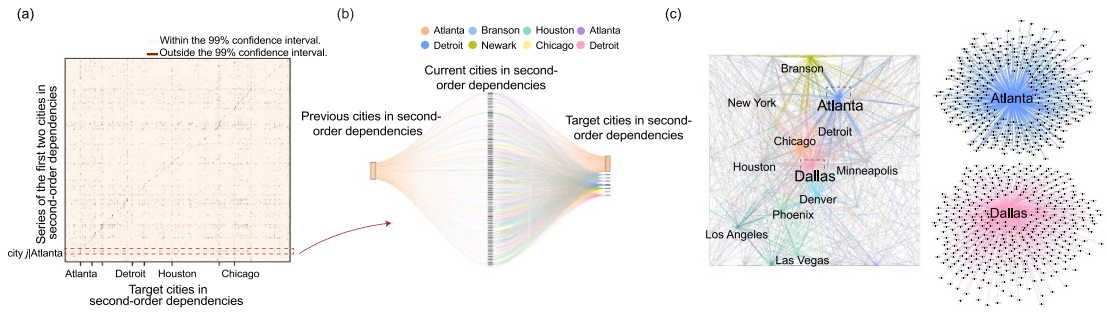

Figure 4. Significant second-order dependencies extracted from the U.S. air traffic network. (a) Significant second-order dependencies at the significance level of $\alpha = 0.01$ are highlighted in saddle brown. The horizontal axis represents the target cities in second-order dependencies, such as Baltimore and Detroit. The vertical axis represents the series of the first two cities in second-order dependencies, such as Branson|Atlanta. (b) A Sankey diagram describes the traffic volume of passengers departing from Atlanta. (c) Higher-order U.S. air traffic network. Based on significant second-order dependencies, we reconstruct a higher-order U.S. air traffic network composed of 405 first-order nodes (a target city v_k), 9842 second-order nodes (a city v_j given a city v_i, denoted as $v_j|v_i$), and 10 676 directed edges ($v_j|v_i \to v_k$) with transition probabilities greater than 0.3. Several clusters are centered around different cities (such as Atlanta, Chicago, Los Angeles, Dallas, and Denver), represented in distinct colors. In addition, two clusters composed of second-order nodes pointing to Atlanta and Dallas are represented in detail.

Table 3. Second-order dependencies captured by four algorithms in three real-world datasets. $N(d)$ represents the number of patterns captured by each method.

	APS dataset		DB1B dataset		Enron email dataset	
Algorithm	$N(d)$	Runtime (s)	$N(d)$	Runtime (s)	$N(d)$	Runtime (s)
PrefixSpan	8	124.18	100	444.04	138	328.04
MON	458	998.32	285 298	29 188.82	2948	6127.82
BuildHON+	36	10.44	65 000	45.11	1100	22.58
S*k*DM	**44**	**12.24**	**65 212**	**55.24**	**1299**	**26.04**

International Airport were the top 5 busiest airports in the U.S. in 2011 (The world's top 100 airports: listed, ranked and mapped).

4.4. Extracting higher-order dependencies in email communications

We finally perform S*k*DM on the Enron email dataset to capture significant higher-order dependencies in human communications. The Enron email dataset consists of 116 525 messages sent between 146 senior executives of Enron from 1999 to 2003. Table 3 shows that 1299 significant second-order dependencies extracted from the network accounted for 1.28% of all 94 176 potential second-order dependencies. Moreover, S*k*DM is faster than MON and PrefixSpan in capturing higher-order dependencies. This demonstrates that our method is adept at selecting vital higher-order dependencies from large datasets. In addition, the percentage of significant dependencies in 2948 existing second-order patterns is as high as 44.06%, which shows a strong second-order Markov effect in email communications. Furthermore, in figure 5(a), we can observe a diagonal line marked by significant second-order dependencies. This suggests a significant commuting pattern in email communications between executives; that is, the next receiver to which an email is forwarded is influenced by the email's previous sender. When an executive received an email, the executive tended to reply to the sender and forward the received email to other executives. A Sankey diagram additionally describes the emails forwarded by John J. Lavorato (47), a chief operating officer of Enron Americas (figure 5(b)). We find that after receiving emails from Lavorato, most users replied to Lavorato and forwarded the received messages to others.

Based on significant second-order dependencies, we construct the higher-order Enron email network, including 131 first-order nodes (an executive k), 645 second-order nodes (an executive j given an executive i, denoted as $j|i$), and 1299 directed edges ($j|i \to k$; figure 5(c)). Using the Louvain method, second-order nodes $j|i$ and its target node k form several tightly knit clusters in figure 5(c). Furthermore, we select 13 clusters centered around different executives and present their structures in detail. As shown in figure 5(c), the conditions i of most second-order nodes $j|i$ are the same as the target nodes k they point to in a cluster. For example, when 16 second-order nodes point to the target node 47|, 14 of them take 47 as conditions, denoted as $j|47$ (approximately 88%). This also confirms the significant commuting pattern in email communications; that is, an individual j most likely sends an email to an individual i if they received an email from j.

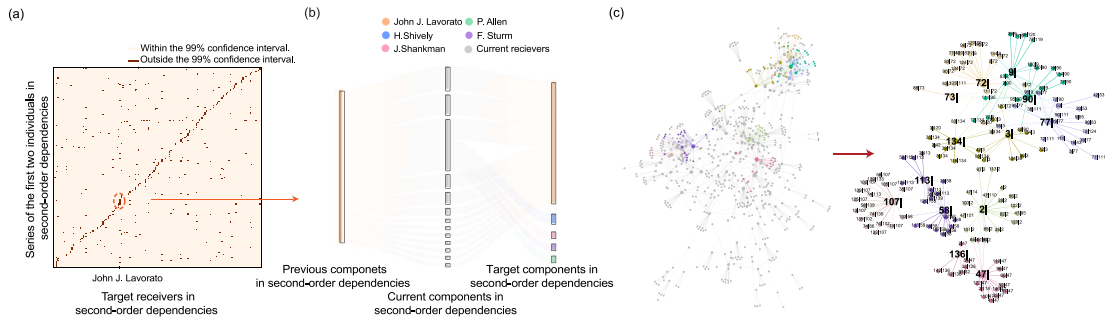

Figure 5. Significant second-order dependencies extracted from the Enron email network. (a) Significant second-order dependencies at the significance level of $\alpha = 0.01$ are highlighted in saddle brown. The horizontal axis represents the target receivers in second-order dependencies. For example, John J. Lavorato is the target receiver of the second-order dependency John J. Lavorato→Greg L. Whalley→John J. Lavorato. The vertical axis represents the series of the first two individuals in second-order dependencies, such as Greg L. Whalley|John J. Lavorato. (b) A Sankey diagram represents the emails forwarded by John J. Lavorato. (c) Higher-order Enron email network. The higher-order Enron email network comprises first-order nodes (an executive i), second-order nodes (an executive j given an executive i, denoted as $j|i$) and directed edges that take the number of emails between nodes as their edge weights. Each executive is indexed by a number, e.g. the index of John J. Lavorato is 47. Using the Louvain method, there are several clusters centered on different executives, represented in different colors. The details of clusters centered around B. Rapp (9), K. Watson (72), M. Lokay (90), S. Harris (77), R. Hayslett (134), M. Heard (113), S. Shackleton (107), M. Taylor (58), L. Kitchen (2), and John J. Lavorato (47) are also shown.

5. Conclusion and discussion

Given that the exponentially increased number of higher-order dependencies introduces problems such as high computational complexity and 'state space explosion', in this paper, we propose a statistical test framework-significant k-order dependencies mining (SkDM)-based on hypothesis testing and the MCMC to identify significant higher-order dependencies from systems. The simulation results indicate that our method can capture all embedded higher-order dependencies at different preset significance levels of $\alpha = 0.01, 0.05, 0.1$ and calculate the p-values of all higher-order dependencies. While existing state-of-the-arts extract higher-order dependencies with high Type I and Type II error rates, our method demonstrates a robust capability for accurately identifying all significant dependencies, maintaining a low Type I error rate and without generating any Type II error across diverse experimental settings. Empirical results on three real-world networks (the APS citation network, the U.S. air traffic network, and the Enron email network) demonstrate that our method can eliminate maximumly 98.7% insignificant higher-order dependencies in large datasets. By capturing significant dependencies, our method can greatly reduce the computational complexity of network analysis tasks. It is worth noting, that in human communications, the proportion of significant dependencies in existing higher-order patterns is considerable (44.06%). Moreover, using SkDM, HON models can precisely describe true network flow patterns in systems, such as commuting patterns, where the network flow often returns to the component it previously visited.

Further research may consider the following two aspects. On the one hand, SkDM can be generalized to select different orders of dependencies from complex systems such as third-order or fourth-order dependencies. Additional research should determine how to reconstruct HONs, which could simultaneously embed variable orders of dependencies. On the other hand, higher-order Markov models offer a temporal-topological perspective for understanding complex systems. It is essential to extend SkDM to investigate how significant higher-order dependencies influence the dynamical processes taking place on time-varying systems, such as epidemic spreading and diffusion.

Data availability statement

Source data are provided with this paper. The APS, DB1B, and Enron Email datasets are publicly available at https://journals.aps.org/datasets, https://transtats.bts.gov/PREZIP/ and http://www.cs.cmu.edu/~enron/. Custom codes for the realization of SkDM is available at: https://gitee.com/jiaxucode/HON-Sig.

Acknowledgments

We thank professor Tao Jia for helpful discussions. This work was supported by the National Natural Science Foundation of China (72025405, 72088101), the National Social Science Foundation of China (22ZDA102),

the Hunan Science and Technology Plan Project (2020TP1013, 2023JJ40685), and the Innovation Team Project of Colleges in Guangdong Province (2020KCXTD040).

J X L and X L have contributed equally to this work. X L conceived the project. J X L and X L designed the models and methods. J X L and X L performed the experiments and wrote the paper. The authors declare no competing interests.

Appendices

Appendices include:

Figure 6: Distribution of z-scores and p-values of dependencies in the CK1 dataset.

Figure 7: Distribution of z-scores and p-values of dependencies in the CK2 dataset.

Figure 8: The Sankey diagram of the first-order APS citation network.

Table 4: p-values of dependencies at preset significance levels of $\alpha = 0.01, 0.05, 0.10$ in the CK1 dataset.

Table 5: z-scores of dependencies at preset significance levels of $\alpha = 0.01, 0.05, 0.10$ in the CK1 dataset.

Table 6: p-values of dependencies at preset significance levels of $\alpha = 0.01, 0.05, 0.10$ in the CK2 dataset.

Table 7: z-scores of dependencies at preset significance levels of $\alpha = 0.01, 0.05, 0.10$ in the CK2 dataset.

Table 8: Second-order dependencies captured by the MON method in the CK2 dataset with different values of p_d.

Table 9: Second-order dependencies captured by the BuildHON+ method in the CK2 dataset with different values of p_d.

Table 10: Basic statistics of three real-network datasets.

Table 11: Journals in the APS dataset.

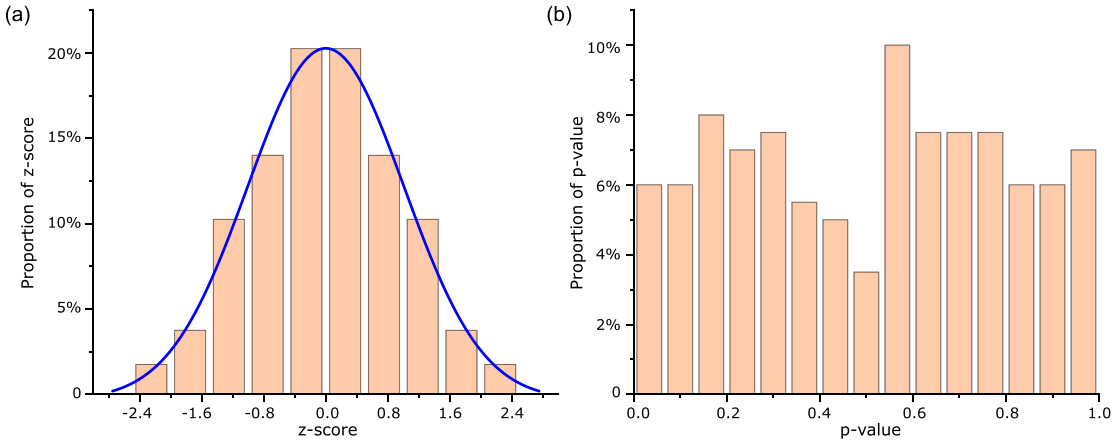

Figure 6. Distribution of *z*-scores and *p*-values of dependencies in the CK1 dataset. (a) The distribution of *z*-scores of all potential second-order dependencies is closely approximated by a normal distribution. (b) The distribution of *p*-values of all potential second-order dependencies.

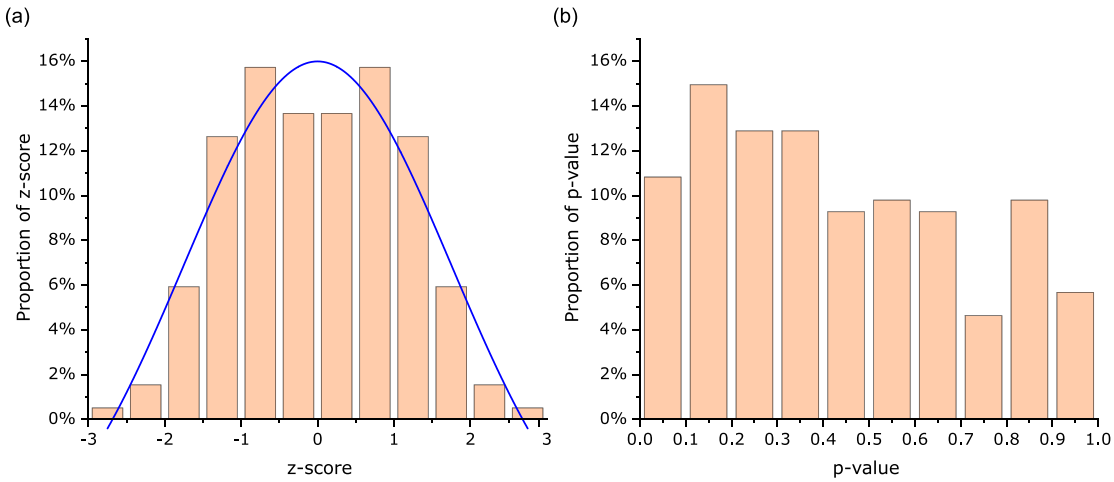

Figure 7. Distribution of *z*-scores and *p*-values of dependencies in the CK2 dataset. (a) The distribution of *z*-scores of potential second-order dependencies (without twelve elaborately designed second-order dependencies on pages 25, 51 and 77) is closely approximated by a normal distribution. (b) The distribution of *p*-values of potential second-order dependencies (without twelve elaborately designed second-order dependencies on pages 25, 51 and 77).

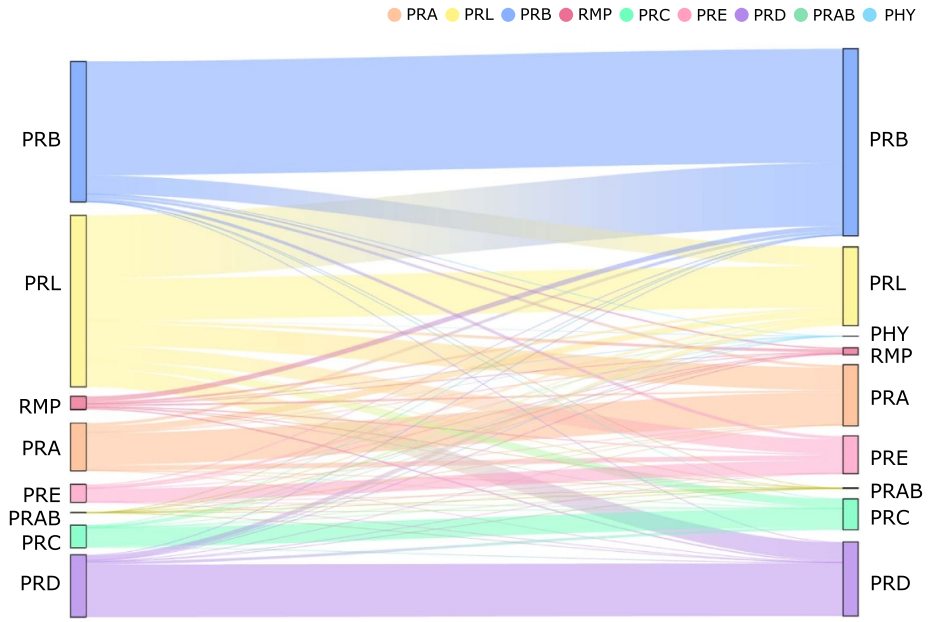

Figure 8. The Sankey diagram of the first-order APS citation network. Different colors represent different journals in the APS citation network. The thickness of each edge is proportional to the citation flow that passes through it.

Table 4. p-values of dependencies at preset significance levels of $\alpha = 0.01, 0.05, 0.1$ in the CK1 dataset.

Second-order dependency	p-value	Second-order dependency	p-value
31→32→33	0.0080	64→65→66	0.0501
31→32→42	0.0080	64→65→75	0.0501
22→32→33	0.0007	60→61→62	0.0566
22→32→42	0.0007	60→61→71	0.0566
33→34→35	0.0134	79→70→71	0.0685
33→34→44	0.0134	79→70→80	0.0685
24→34→35	0.0134	63→64→65	0.0772
24→34→44	0.0134	63→64→74	0.0772
19→29→20	0.0248	68→78→79	0.0773
19→29→39	0.0248	68→78→88	0.0773
52→53→54	0.0295	18→19→10	0.0866
52→53→63	0.0295	18→19→29	0.0866
45→55→56	0.0378	55→65→66	0.0869
45→55→65	0.0378	55→65→75	0.0869
51→52→53	0.0413	87→97→7	0.0874
51→52→62	0.0413	87→97→98	0.0874
61→71→72	0.0422	34→35→36	0.0953
61→71→81	0.0422	34→35→45	0.0953
50→51→52	0.0466	12→13→14	0.0979
50→51→61	0.0466	12→13→23	0.0979

Table 5. z-scores of dependencies at preset significance levels of $\alpha = 0.01, 0.05, 0.1$ in the CK1 dataset.

Second-order dependency	z-score	Second-order dependency	z-score
31→32→33	2.6305	64→65→66	1.9595
31→32→42	−2.6305	64→65→75	−1.9595
22→32→33	−3.3927	60→61→62	1.9064
22→32→42	3.3927	60→61→71	−1.9064
33→34→35	−2.4737	79→70→71	1.8215
33→34→44	2.4737	79→70→80	−1.8215
24→34→35	2.4736	63→64→65	1.7673
24→34→44	−2.4736	63→64→74	−1.7673
19→29→20	−2.2451	68→78→79	1.7667
19→29→39	2.2451	68→78→88	−1.7667
52→53→54	−2.177	18→19→10	1.7134
52→53→63	2.177	18→19→29	−1.7134
45→55→56	2.0775	55→65→66	1.7117
45→55→65	−2.0775	55→65→75	−1.7117
51→52→53	2.0407	87→97→7	1.7095
51→52→62	−2.0407	87→97→98	−1.7095
61→71→72	2.0316	34→35→36	1.668
61→71→81	−2.0316	34→35→45	−1.668
50→51→52	1.9895	12→13→14	1.6551
50→51→61	−1.9895	12→13→23	−1.6551

Table 6. p-values of dependencies at preset significance levels of $\alpha = 0.01, 0.05, 0.1$ in the CK2 dataset. The twelve elaborately designed second-order dependencies on pages 25, 51 and 77 are: 15→25→26, 15→25→35, 24→25→26, 24→25→35, 41→51→52, 41→51→61, 50→51→52, 50→51→61, 67→77→78, 67→77→87, 76→77→78 and 76→77→87.

Second-order dependency	p-value	Second-order dependency	p-value
15→25→26	0	26→36→46	0.0415
15→25→35	0	26→36→37	0.0415
24→25→26	0	29→20→21	0.0474
24→25→35	0	29→20→30	0.0474
41→51→52	0	11→12→13	0.0485
41→51→61	0	11→12→22	0.0485
50→51→52	0	52→53→54	0.0569
50→51→61	0	52→53→63	0.0569
67→77→78	0	74→84→85	0.0584
67→77→87	0	74→84→94	0.0584
76→77→78	0	16→26→27	0.0588
76→77→87	0	16→26→36	0.0588
71→81→82	0.0094	39→49→40	0.0650
71→81→91	0.0094	39→49→59	0.0650
80→81→82	0.0089	66→76→77	0.0654
80→81→91	0.0089	66→76→86	0.0654
69→79→89	0.0195	17→27→28	0.0708
69→79→70	0.0195	17→27→37	0.0708
78→79→70	0.0212	63→64→65	0.0867
78→79→89	0.0212	63→64→74	0.0867
35→36→37	0.0285	27→28→38	0.0886
35→36→46	0.0285	27→28→29	0.0886
43→53→54	0.0345	1→11→21	0.0889
43→53→63	0.0345	1→11→12	0.0889
48→49→40	0.0412	10→11→12	0.0946
48→49→59	0.0412	10→11→21	0.0946

Table 7. z-scores of dependencies at preset significance levels of $\alpha = 0.01, 0.05, 0.1$ in the CK2 dataset. The twelve elaborately designed second-order dependencies on pages 25, 51 and 77 are: 15→25→26, 15→25→35, 24→25→26, 24→25→35, 41→51→52, 41→51→61, 50→51→52, 50→51→61, 67→77→78, 67→77→87, 76→77→78 and 76→77→87.

Second-order dependency	z-score	Second-order dependency	z-score
15→25→26	95.5157	26→36→46	2.0380
15→25→35	−95.5157	26→36→37	−2.0380
24→25→26	97.1868	29→20→21	1.9831
24→25→35	−97.1868	29→20→30	−1.9831
41→51→52	99.0834	11→12→13	1.9728
41→51→61	−99.0834	11→12→22	−1.9728
50→51→52	96.1407	52→53→54	1.9043
50→51→61	−96.1407	52→53→63	−1.9043
67→77→78	95.5764	74→84→85	1.8925
67→77→87	−95.5764	74→84→94	−1.8925
76→77→78	95.4083	16→26→27	1.8893
76→77→87	−95.4083	16→26→36	−1.8893
71→81→82	2.5990	39→49→40	1.8454
71→81→91	−2.5990	39→49→59	−1.8454
80→81→82	2.6141	66→76→77	1.8427
80→81→91	−2.6141	66→76→86	−1.8427
69→79→89	2.3357	17→27→28	1.8064
69→79→70	−2.3357	17→27→37	−1.8064
78→79→70	2.3038	63→64→65	1.7131
78→79→89	−2.3038	63→64→74	−1.7131
35→36→37	2.1909	27→28→38	1.7031
35→36→46	−2.1909	27→28→29	−1.7031
43→53→54	2.1145	1→11→21	1.7014
43→53→63	−2.1145	1→11→12	−1.7014
48→49→40	2.0411	10→11→12	1.6717
48→49→59	−2.0411	10→11→21	−1.6717

Table 8. Second-order dependencies captured by the MON method in the CK2 dataset with different values of p_d.

MON	$p_d = 0.10$	$p_d = 0.20$	$p_d = 0.30$
$N(d) = N(TS) + N(FS)$	400	400	400
$N(FS)$	388	388	388
$N(TNS)$	0	0	0
$N(FNS)$	0	0	0
$N(TS)$	12	12	12
$\alpha = \frac{N(FS)}{N(FS)+N(TNS)}$	1.00	1.00	1.00
$\beta = \frac{N(FNS)}{N(FNS)+N(TS)}$	0.00	0.00	0.00

Table 9. Second-order dependencies captured by the BuildHON+ method in the CK2 dataset with different values of p_d.

BuildHON+	$p_d = 0.10$	$p_d = 0.20$	$p_d = 0.30$
$N(d) = N(TS) + N(FS)$	12	12	0
$N(FS)$	0	0	0
$N(TNS)$	388	388	388
$N(FNS)$	0	0	12
$N(TS)$	12	12	0
$\alpha = \frac{N(FS)}{N(FS)+N(TNS)}$	0.00	0.00	0.00
$\beta = \frac{N(FNS)}{N(FNS)+N(TS)}$	0.00	0.00	1.00

Table 10. Basic statistics of three real-network datasets. $|V|$ and $|E|$ is the number of components and directed pairwise links. The domain is the research field of a dataset.

Network	$\|V\|$	$\|E\|$	Domain
APS citation network	468 291	906 398	Citation
US air traffic network	413	19 415 369	Transportation
Enron email network	146	116 525	Communication

Table 11. Journals in the APS dataset.

Journal	Journal Code	Abbreviation	Research Fields	Number of articles
Physical Review Letters	*Phys. Rev. Lett.*	PRL	Important fundamental research in allfields of physics	149 766
Physical Review A	*Phys. Rev. A*	PRA	Atomic, molecular, and optical physics andquantum information	53 655
Physical Review B	*Phys. Rev. B*	PRB	Condensed matter and materials physics	137 999
Physical Review C	*Phys. Rev. C*	PRC	Nuclear physics	29 935
Physical Review D	*Phys. Rev. D*	PRD	Particles, fields, gravitation, andcosmology	56 616
Physical Review E	*Phys. Rev. E*	PRE	Statistical, nonlinear, biological, andsoft matter physics	35 944
Physical Review Accelerators and Beams	*Phys. Rev. Accel. Beams*	PRAB	Accelerators and Beams	1257
Reviews of Modern Physics	*Rev. Mod. Phys*	RMP	Reviews of physics	2926
Physics	*Physics*	PHY	Latest researches on all aspects of physics	193

ORCID iDs

Jiaxu Li https://orcid.org/0000-0002-3740-1866
Xin Lu https://orcid.org/0000-0002-3547-6493

References

[1] Newman M E 2003 *SIAM Rev.* **45** 167–256
[2] Latora V, Nicosia V and Russo G 2017 *Complex Networks: Principles, Methods and Applications* (Cambridge University Press)
[3] Caldarelli G 2007 *Scale-Free Networks: Complex Webs in Nature and Technology* (Oxford University Press)
[4] Peixoto T P and Rosvall M 2017 *Nat. Commun.* **8** 582
[5] Xu J, Wickramarathne T L and Chawla N V 2016 *Sci. Adv.* **2** e1600028
[6] Lambiotte R, Rosvall M and Scholtes I 2019 *Nat. Phys.* **15** 313–20
[7] Nguyen D T, Shen Y and Thai M T 2013 *IEEE Trans. Smart Grid* **4** 151–9
[8] Petrović L V, Wegner A and Scholtes I 2023 Higher-order patterns reveal causal timescales of complex systems (arXiv:2301.11623)
[9] Rosvall M, Esquivel A V, Lancichinetti A, West J D and Lambiotte R 2014 *Nat. Commun.* **5** 4630
[10] Holmgren A, Edler D and Rosvall M 2023 *Appl. Netw. Sci.* **8** 42
[11] Kovalenko K *et al* 2022 *Chaos Solitons Fractals* **162** 112397
[12] Battiston F *et al* 2021 *Nat. Phys.* **17** 1093–8
[13] Majhi S, Perc M and Ghosh D 2022 *J. R. Soc. Interface* **19** 20220043
[14] Williams O E, Mazzarisi P, Lillo F and Latora V 2022 *Phys. Rev.* E **105** 034301
[15] Battiston F and Petri G 2022 *Higher-Order Systems* (Springer)
[16] Chen Q, Zhao Y, Li C and Li X 2023 *New J. Phys.* **25** 113045
[17] Zhu Y, Li C and Li X 2023 *New J. Phys.* **25** 113043
[18] Carletti T, Fanelli D and Nicoletti S 2020 *J. Phys. Complex.* **1** 035006
[19] Tang Y, Shi D and Lü L 2022 *Commun. Phys.* **5** 96
[20] Zhang Y, Lucas M and Battiston F 2023 *Nat. Commun.* **14** 1605
[21] Alvarez-Rodriguez U, Battiston F, de Arruda G F, Moreno Y, Perc M and Latora V 2021 *Nat. Hum. Behav.* **5** 586–95
[22] Kumar A, Chowdhary S, Capraro V and Perc M 2021 *Phys. Rev.* E **104** 054308
[23] Battiston F, Cencetti G, Iacopini I, Latora V, Lucas M, Patania A, Young J G and Petri G 2020 *Phys. Rep.* **874** 1–92
[24] Bianconi G 2021 *Higher-Order Networks* (Cambridge University Press)
[25] Iacopini I, Petri G, Barrat A and Latora V 2019 *Nat. Commun.* **10** 2485
[26] Li Z, Deng Z, Han Z, Alfaro-Bittner K, Barzel B and Boccaletti S 2021 *Chaos Solitons Fractals* **152** 111307
[27] Chowdhary S, Kumar A, Cencetti G, Iacopini I and Battiston F 2021 *J. Phys. Complex.* **2** 035019
[28] Matamalas J T, Gómez S and Arenas A 2020 *Phys. Rev. Res.* **2** 012049
[29] Parastesh F, Mehrabbeik M, Rajagopal K, Jafari S and Perc M 2022 *Chaos* **32** 013125
[30] Gao S, Chang L, Perc M and Wang Z 2023 *Phys. Rev.* E **107** 014216
[31] Benson A R, Gleich D F and Leskovec J 2016 *Science* **353** 163–6
[32] Yin H, Benson A R, Leskovec J and Gleich D F 2017 *Proc. 23rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* pp 555–64
[33] Carranza A G, Rossi R A, Rao A B and Koh E 2020 *Proc. 26th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining* pp 25–35
[34] Yin H, Benson A R and Leskovec J 2018 *Phys. Rev.* E **97** 052306
[35] Yin H, Benson A R and Leskovec J 2019 *Proc. 12th ACM Int. Conf. on Web Search and Data Mining* pp 303–11
[36] Paranjape A, Benson A R and Leskovec J 2017 *Proc. 10th ACM Int. Conf. on web Search and Data Mining* pp 601–10

[37] Benson A R, Abebe R, Schaub M T, Jadbabaie A and Kleinberg J 2018 *Proc. Natl Acad. Sci.* **115** E11221–30
[38] Rossi R A, Rao A, Kim S, Koh E and Ahmed N 2020 *Companion Proc. Web Conf. 2020* pp 42–43
[39] Scholtes I, Wider N, Pfitzner R, Garas A, Tessone C J and Schweitzer F 2014 *Nat. Commun.* **5** 5024
[40] Scholtes I, Wider N and Garas A 2016 *Eur. Phys. J.* B **89** 1–15
[41] Edler D, Bohlin L and Rosvall M 2017 *Algorithms* **10** 112
[42] Scholtes I 2017 *Proc. 23rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* pp 1037–46
[43] Gote C, Perri V and Scholtes I 2023 *Proc. 2022 IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining* pp 109–16
[44] Saebi M, Xu J, Kaplan L M, Ribeiro B and Chawla N V 2020 *EPJ Data Sci.* **9** 15
[45] Saebi M, Ciampaglia G L, Kaplan L M and Chawla N V 2020 *Big Data* **8** 255–69
[46] Ching W K, Huang X, Ng M K and Siu T K 2013 *Higher-Order Markov Chains* (Springer US) pp 141–76
[47] Salnikov V, Schaub M T and Lambiotte R 2016 *Sci. Rep.* **6** 1–13
[48] Chierichetti F, Kumar R, Raghavan P and Sarlos T 2012 *Proc. 21st Int. Conf. on World Wide Web* pp 609–18
[49] Rubinstein R Y and Kroese D P 2016 *Simulation and the Monte Carlo Method* (Wiley)
[50] Hammersley J 2013 *Monte Carlo Methods* (Springer)
[51] Jones G L and Qin Q 2022 *Annu. Rev. Stat. Appl.* **9** 557–78
[52] Jensen W A and Alexander M 2016 *J. Qual. Technol.* **48** 297–9
[53] Devore J 2006 *J. Am. Stat. Assoc.* **101** 393–4
[54] Fisher E, Schweiger R and Rosset S 2020 *J. Comput. Graph. Stat.* **29** 140–8
[55] Morey R D, Hoekstra R, Rouder J N, Lee M D and Wagenmakers E J 2016 *Psychonomic Bull. Rev.* **23** 103–23
[56] Greenland S, Senn S J, Rothman K J, Carlin J B, Poole C, Goodman S N and Altman D G 2016 *Eur. j. Epidemiol.* **31** 337–50
[57] Han J, Pei J, Mortazavi-Asl B, Pinto H, Chen Q, Dayal U and Hsu M 2001 *Proc. 17th Int. Conf. on Data Engineering* pp 215–24
[58] American Physical Society 2011 APS data sets for research (available at: https://journals.aps.org/datasets accessed) (Accessed 16 June 2023)
[59] United States Department of Transportation 2012 DB1BCoupon (available at: https://transtats.bts.gov/PREZIP/) (Accessed 16 June 2023)
[60] The CALO Project 2015 Enron Email dataset (available at: http://www.cs.cmu.edu/~enron/) (Accessed 16 June 2023)