

The Spacey Random Walk: A Stochastic Process for Higher-Order Data*

Austin R. Benson[†]
David F. Gleich[‡]
Lek-Heng Lim[§]

Abstract. Random walks are a fundamental model in applied mathematics and are a common example of a Markov chain. The limiting stationary distribution of the Markov chain represents the fraction of the time spent in each state during the stochastic process. A standard way to compute this distribution for a random walk on a finite set of states is to compute the Perron vector of the associated transition matrix. There are algebraic analogues of this Perron vector in terms of transition probability tensors of higher-order Markov chains. These vectors are nonnegative, have dimension equal to the dimension of the state space, and sum to one, and they are derived by making an algebraic substitution in the equation for the joint-stationary distribution of a higher-order Markov chain. Here, we present the *spacey random walk*, a non-Markovian stochastic process whose stationary distribution is given by the tensor eigenvector. The process itself is a vertex-reinforced random walk, and its discrete dynamics are related to a continuous dynamical system. We analyze the convergence properties of these dynamics and discuss numerical methods for computing the stationary distribution. Finally, we provide several applications of the spacey random walk model in population genetics, ranking, and clustering data, and we use the process to analyze New York taxi trajectory data. This example shows definite non-Markovian structure.

Key words. random walk, tensor, eigenvector, vertex-reinforced random walk

AMS subject classifications. 15A69, 15B51, 05C65, 05C81, 05C82, 37C25, 65H04, 90C35

DOI. 10.1137/16M1074023

1. Random Walks, Higher-Order Markov Chains, and Stationary Distributions. Random walks and Markov chains are two of the most well-known and studied stochastic processes and are common tools in applied mathematics. A random walk

*Received by the editors May 5, 2016; accepted for publication (in revised form) September 28, 2016; published electronically May 5, 2017.

<http://www.siam.org/journals/sirev/59-2/M107402.html>

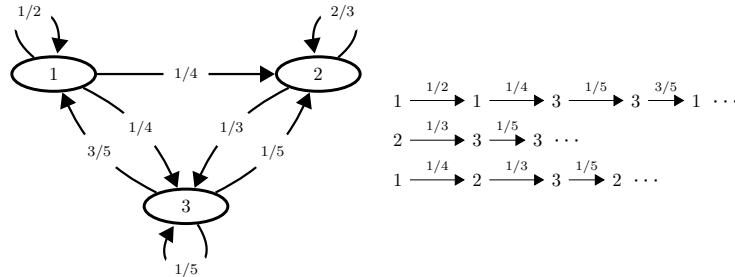
Funding: The work of the first author was supported by a Stanford Graduate Fellowship. The work of the second author was supported by the NSF through grants CCF-1149756, IIS-1422918, IIS-1546488, and the DARPA SIMPLEX program and was also partially supported by an Alfred P. Sloan Research Fellowship. The work of the third author was supported by AFOSR grant FA9550-13-1-0133, DARPA grant D15AP00109, and NSF grants IIS-1546413, DMS-1209136, and DMS-1057064.

[†]Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305 (arbenson@stanford.edu).

[‡]Department of Computer Science, Purdue University, West Lafayette, IN 47907 (dgleich@purdue.edu).

[§]Computational and Applied Mathematics Initiative, Department of Statistics, University of Chicago, Chicago, IL 60637 (lekheng@uchicago.edu).

on a finite set of states is a process that moves from state to state in a manner that *depends only on the last state* and an associated set of transition probabilities from that state to the other states. Here is one such example, with a few sample trajectories of transitions:



This process is also known as a Markov chain, and in the setting we consider here the two models, Markov chains and random walks, are equivalent. (For the experts reading this article, we are considering time-homogeneous walks and Markov chains on finite state spaces.) Applications of random walks include the following:

- Google’s PageRank model (Langville and Meyer, 2006). States are webpages and transition probabilities correspond to hyperlinks between pages.
- Card shuffling. Markov chains can be used to show that shuffling a card deck seven times is sufficient; see, for example, Bayer and Diaconis (1992) and Jonsson and Trefethen (1998). In this case, the states correspond to permutations of a deck of cards, and transitions are the result of a riffle shuffle.
- Markov chain Monte Carlo and Metropolis Hastings (Asmussen and Glynn, 2007). Here, the goal is to sample from complicated probability distributions that could model extrema of a function when used for optimization or from the uniform distribution if the goal is a random instance of a complex mathematical object such as matchings or Ising models. For optimization, states reflect the optimization variables, and transitions are designed to stochastically “optimize” the function value; for random instance generation, states are the mathematical objects, and transitions are usually simple local rearrangements.

One of the key properties of a Markov chain is its stationary distribution, which models where we expect the walk to be “on average” as the process runs to infinity. (Again, for the expert reader, we are concerned with the Cesàro limiting distributions.) To compute and understand these stationary distributions, we turn to matrices. For an N state Markov chain, there is an $N \times N$ column stochastic matrix \mathbf{P} , where \mathbf{P}_{ij} is the probability of transitioning to state i from state j . The matrix for the previous chain is

$$\mathbf{P} = \begin{bmatrix} 1/2 & 0 & 3/5 \\ 1/4 & 2/3 & 1/5 \\ 1/4 & 1/3 & 1/5 \end{bmatrix}.$$

A stationary distribution on the states is a vector $\mathbf{x} \in \mathbb{R}^N$ satisfying

$$(1) \quad \mathbf{x}_i = \sum_j \mathbf{P}_{ij} \mathbf{x}_j, \quad \sum_i \mathbf{x}_i = 1, \quad \mathbf{x}_i \geq 0, \quad 1 \leq i \leq N.$$

Table 1 Transition probabilities for a second-order Markov chain model on the state space $\{1, 2, 3\}$.

Second last state	1			2			3		
Last state	1	2	3	1	2	3	1	2	3
Prob. next state is 1	0	0	0	1/4	0	0	1/4	0	3/4
Prob. next state is 2	3/5	2/3	0	1/2	0	1/2	0	1/2	0
Prob. next state is 3	2/5	1/3	1	1/4	1	1/2	3/4	1/2	1/4

The existence and uniqueness of \mathbf{x} , as well as efficient numerical algorithms for computing \mathbf{x} , are all well understood (Kemeny and Snell, 1960; Stewart, 1994).

A natural extension of Markov chains is to have the transitions depend on the past few states, rather than just the last one. These processes are called *higher-order Markov chains* and are much better at modeling data in a variety of applications including airport travel flows (Rosvall et al., 2014), e-mail communication (Rosvall et al., 2014), web browsing behavior (Chierichetti et al., 2012), and network clustering (Krzakala et al., 2013). For example, Table 1 shows the transition probability table for a second-order Markov chain model on the state space $\{1, 2, 3\}$. Thus, for instance, if the last two states were 2 (last state) and 1 (second last state), then the next state will be 2 with probability $2/3$ and 3 with probability $1/3$.

Stationary distributions of higher-order Markov chains show which states appear “on average” as the process runs to infinity. To compute them, we turn to hypermatrices. We can encode these probabilities into a transition hypermatrix where \underline{P}_{ijk} is the probability of transitioning to state i , given that the last state was j and the second last state was k . So, for this example, we have $\underline{P}_{321} = 1/3$. In this case, $\sum_i \underline{P}_{ijk} = 1$ for all j and k , and we call these stochastic hypermatrices. (For an m -order Markov chain, we will have an $m+1$ -order stochastic hypermatrix.) The stationary distribution of a second-order Markov chain, it turns out, is simply computed by converting the second-order Markov chain into a first-order Markov chain on a larger state space given by *pairs* of states. To see how this conversion takes place, note that if the previous two states were 2 and 1 as in the example above, then we view these as an ordered pair $(2, 1)$. The next pair of states will be either $(2, 2)$ with probability $2/3$ or $(3, 2)$ with probability $1/3$. Thus, the sequence of states generated by a second-order Markov chain can be extracted from the sequence of states of a first-order Markov chain defined on *pairs* of states. Figure 1 shows a graphical illustration of the first-order Markov chain that arises from our small example.

The stationary distribution of the first-order chain is an $N \times N$ matrix \mathbf{X} where \mathbf{X}_{ij} is the stationary probability associated with the pair of states (i, j) . This matrix satisfies the stationary equations

$$(2) \quad \mathbf{X}_{ij} = \sum_k \underline{P}_{ijk} \mathbf{X}_{jk}, \quad \sum_{i,j} \mathbf{X}_{ij} = 1, \quad \mathbf{X}_{ij} \geq 0, \quad 1 \leq i, j \leq N.$$

(These equations can be further transformed into a matrix and a vector $\mathbf{x} \in \mathbb{R}^{N^2}$ if desired, but for our exposition, this is unnecessary.) The conditions when \mathbf{X} exists can be deduced from applying the Perron–Frobenius theorem to this more complicated equation. Once the matrix \mathbf{X} is found, the stationary distribution over states of the second-order chain is given by the row and column sums of \mathbf{X} .

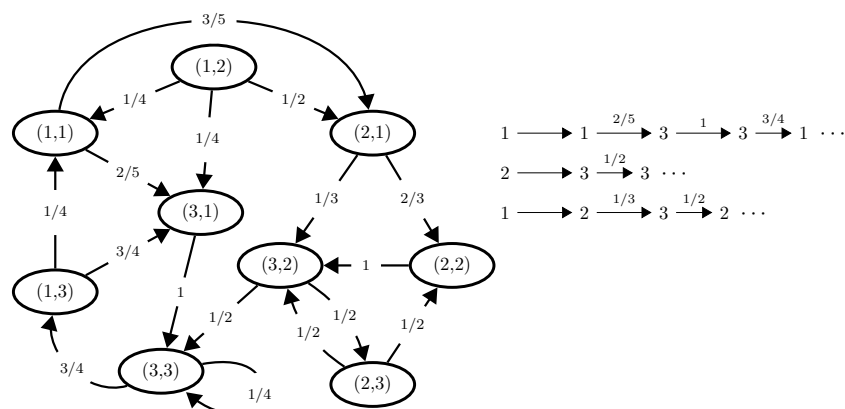


Fig. 1 The first-order Markov chain that results from converting the second-order Markov chain in Table 1 into a first-order Markov chain on pairs of states. Here, state (i, j) means that the last state was i and the next-to-last state was j . Because we are modeling a second-order Markov chain, the only transitions from state (i, j) are to state (k, i) for some k .

Computing the stationary distribution in this manner requires $\Theta(N^2)$ storage, regardless of any possible efficiency in storing and manipulating \underline{P} . For modern problems on large datasets, this is infeasible.¹

Recent work by Li and Ng (2014) provides a space-friendly alternative approximation through z eigenvectors of the transition hypermatrix, as we now explain. Li and Ng (2014) consider a “rank-one approximation” to \mathbf{X} , i.e., $\mathbf{X}_{ij} = \mathbf{x}_i \mathbf{x}_j$ for some vector $\mathbf{x} \in \mathbb{R}^N$. In this case, (2) reduces to

$$(3) \quad \mathbf{x}_i = \sum_{jk} \mathbf{P}_{ijk} \mathbf{x}_j \mathbf{x}_k, \quad \sum_i \mathbf{x}_i = 1, \quad \mathbf{x}_i \geq 0, \quad 1 \leq i \leq N.$$

Without the stochastic constraints on the vector entries, \mathbf{x} is called a z eigenvector (Qi, 2005) or an l^2 eigenvector (Lim, 2005) of \underline{P} . Li and Ng (2014) and Gleich, Lim, and Yu (2015) analyze when a solution vector \mathbf{x} for (3) exists and provide algorithms for computing the vector. These algorithms are guaranteed to converge to a unique solution vector \mathbf{x} if \underline{P} satisfies certain properties. Because the entries of \mathbf{x} sum to one and are nonnegative, they can be interpreted as a probability vector. However, this transformation is algebraic. We do not have a canonical process like the random walk or second-order Markov chain connected to the vector \mathbf{x} .

In this article, we provide an underlying stochastic process, the *spacey random walk*, where the limiting proportion of the time spent at each node—if this quantity exists—is the z eigenvector \mathbf{x} computed above. The process is an instance of a *vertex-reinforced random walk* (Pemantle, 2007). The process acts like a random walk but edge traversal probabilities depend on previously visited nodes. We will make this idea formal in the following sections.

¹We wish to mention that our goal is not the stationary distribution of the higher-order chain in a space efficient matter. For instance, a memory-friendly alternative is directly simulating the chain and computing an empirical estimate of the stationary distribution. Instead, our goal is to understand a recently proposed *algebraic* approximation to this stationary distribution in terms of tensor eigenvectors.

Vertex-reinforced random walks were introduced by Coppersmith and Diaconis (1987) and Pemantle (1988), and the spacey random walk is a specific type of a more general class of vertex-reinforced random walks analyzed by Benaïm (1997). A crucial insight from Benaïm’s analysis is the connection between the discrete stochastic process and a continuous dynamical system. Essentially, the limiting probability of time spent at each node in the graph relates to the long-term behavior of the dynamical system. For our vertex-reinforced random walk, we can significantly refine this analysis. For example, in the special case of a two-state discrete system, we show that the corresponding dynamical system for our process always converges to a stable equilibrium (section 4.1). When our process has several states, we give sufficient conditions under which a standard method for integrating dynamical systems will converge to a stationary distribution (section 5.3).

We provide several applications of the spacey random walk process in section 3 and use the model to analyze a dataset of taxi trajectories in section 6. Our work adds to the long and continuing history of the application of tensor² or hypermatrix methods to data (Harshman, 1970; Comon, 1994; Sidiropoulos, Giannakis, and Bro, 2000; Allman and Rhodes, 2003; Smilde, Bro, and Geladi, 2004; Sun, Tao, and Faloutsos, 2006; Anandkumar et al., 2014). For example, eigenvectors of structured hypermatrix data are crucial to new algorithms for parameter recovery in a variety of machine learning applications (Anandkumar et al., 2014). In this paper, our hypermatrices come from a particular structure—higher-order Markov chains—and we believe that our new understanding of the eigenvectors of these hypermatrices will lead to improved data analysis.

The software used to produce our figures and numerical results is publicly available at <https://github.com/arbenson/spacey-random-walks>.

2. The Spacey Random Walk. We now describe the stochastic process, which we call the *spacey random walk*. This process consists of a sequence of states $X(0), X(1), X(2), \dots$, and it uses the set of transition probabilities from a second-order Markov chain. In a second-order Markov chain, we use $X(n)$ and $X(n-1)$ to *look up* the appropriate column of transition data based on the last two states of history. In the spacey random walker process on this data, once the process visits $X(n)$, it *spaces out* and forgets its second last state (that is, the state $X(n-1)$). It then invents a new history state $Y(n)$ by randomly drawing a past state $X(1), \dots, X(n)$. Then it transitions to $X(n+1)$ as a second-order Markov chain as if its last two states were $X(n)$ and $Y(n)$. We illustrate this process in Figure 2.

We formalize this idea as follows. Let \underline{P}_{ijk} be the transition probabilities of the second-order Markov chain with N states such that

$$\Pr\{X(n+1) = i \mid X(n) = j, X(n-1) = k\} = \underline{P}_{ijk}.$$

The probability law of the spacey random surfer is given by

$$(4) \quad \Pr\{Y(n) = k \mid \mathcal{F}_n\} = \frac{1}{n+N} (1 + \sum_{s=1}^n \text{Ind}\{X(s) = k\}),$$

$$(5) \quad \Pr\{X(n+1) = i \mid X(n) = j, Y(n) = k\} = \underline{P}_{ijk},$$

where \mathcal{F}_n is the σ -field generated by the random variables $X(i)$, $i = 1, \dots, n$, and $X(0)$ is a provided starting state. This system describes a random process with reinforcement (Pemantle, 2007) and, more specifically, a new type of generalized

²These are coordinate representations of hypermatrices and are often incorrectly referred to as “tensors.” In particular, the transition hypermatrix of a higher-order Markov chain is coordinate dependent and is not a tensor. We refer to Lim (2013) for formal definitions of these terms.

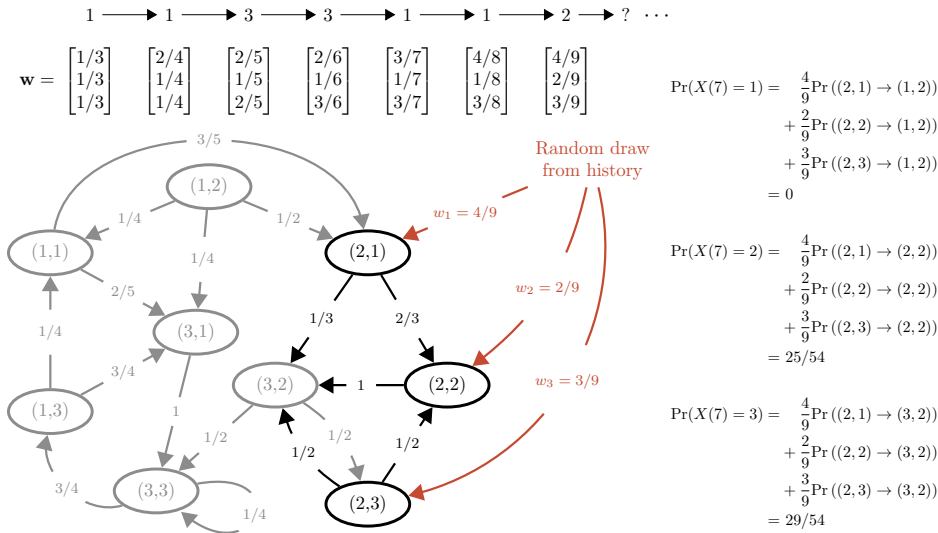


Fig. 2 The spacey random walk process uses a set of transition probabilities from a second-order Markov chain (which we represent here via the first-order reduced chain; see Figure 1) and maintains an occupation vector \mathbf{w} as well. The vector \mathbf{w} keeps track of the fraction of time spent in each state, initialized with one visit at each state. In the next transition, the process chooses the past state following this vector (red lines) and then makes a transition following the second-order Markov chain (black lines). In this case, the last state is 2 and the second last state will be generated from \mathbf{w} , giving the transition probabilities to the next state $X(7)$ on the right.

vertex-reinforced random walk (see section 2.1). For notational convenience, we let the $N \times N^2$ column-stochastic matrix \mathbf{R} denote the flattening of $\underline{\mathbf{P}}$ along the first index:

$$\mathbf{R} := [\underline{\mathbf{P}}_{\bullet\bullet 1} \mid \underline{\mathbf{P}}_{\bullet\bullet 2} \mid \dots \mid \underline{\mathbf{P}}_{\bullet\bullet N}].$$

(We will use the term *flattening*, but we note that this operation is also referred to as an *unfolding* and sometimes denoted by $\mathbf{R} = \underline{\mathbf{P}}_{(1)}$ (Golub and Van Loan, 2012, Chapter 12.1).) We refer to $\underline{\mathbf{P}}_{\bullet\bullet i}$ as the i th *panel* of \mathbf{R} and note that each panel is itself a transition matrix on the original state space. Transitions of the spacey random walker then correspond to (i) selecting the panel from the random variable $Y(n)$ in (4) and (ii) following the transition probabilities in the column of the panel corresponding to the last state.

2.1. Relationship to Vertex-Reinforced Random Walks. The spacey random walk is an instance of a (generalized) vertex-reinforced random walk, a process introduced by Benaïm (1997). A vertex-reinforced random walk is a process that always picks the next state based on a set of transition probabilities, but those transition probabilities evolve as the process continues. A simple example would be a random walk on a graph where the edge traversals are proportional to the amount of time spent at each vertex so far (Diaconis, 1988). At the start of the process, each vertex is assigned a score of 1, and after each transition to vertex i from vertex j , the score of vertex i is incremented by 1. At any point in the process, transitions from vertex j are made proportional to the score of vertex j 's neighbors in the graph. As discussed by Diaconis (1988), this process could model how someone navigates a new city, where familiar locations are more likely to be traversed in the future.

Formally, the vertex-reinforced random walk is a stochastic process with states $X(i)$, $i = 0, 1, \dots$, governed by

$$(6) \quad X(0) = x(0),$$

$$(7) \quad \mathbf{s}_i(n) = 1 + \sum_{s=1}^n \text{Ind}\{X(s) = i\}, \quad \mathbf{w}(n) = \frac{\mathbf{s}(n)}{N + n},$$

$$(8) \quad \Pr\{X(n + 1) = i \mid \mathcal{F}_n\} = [\mathbf{M}(\mathbf{w}(n))]_{i, X(n)},$$

where again \mathcal{F}_n is the σ -field generated by the random variables $X(i)$, $i = 1, \dots, n$, $x(0)$ is the initial state, and $\mathbf{M}(\mathbf{w}(n))$ is an $N \times N$ column-stochastic matrix given by a map

$$(9) \quad \mathbf{M}: \Delta_{N-1} \rightarrow \{\mathbf{P} \in \mathbb{R}^{N \times N} \mid \mathbf{P}_{ij} \geq 0, \mathbf{e}^T \mathbf{P} = \mathbf{e}^T\}, \quad \mathbf{x} \mapsto \mathbf{M}(\mathbf{x}),$$

where Δ_{N-1} is the simplex of length N probability vectors that are nonnegative and sum to one. In other words, transitions at any point of the walk may depend on the relative amount of time spent in each state up to that point. The vector \mathbf{w} is called the *occupation vector* because it represents the empirical distribution of time spent at each state. However, note that the initial state $X(0)$ does not contribute to the occupation vector.

A key result from Benaïm (1997) is the relationship between the discrete vertex-reinforced random walk and the dynamical system

$$(10) \quad \frac{d\mathbf{x}}{dt} = \pi(\mathbf{M}(\mathbf{x})) - \mathbf{x},$$

where π is the map that sends a transition matrix to its stationary distribution. Essentially, the possible limiting distributions of the occupation vector \mathbf{w} are described by the long-term dynamics of the system in (10) (see section 5.1 for details). Put another way, *convergence of the dynamical system to a fixed point* is equivalent to *convergence of the occupation vector* for the stochastic process, and, thus, the convergence of the dynamical system implies the existence of a stationary distribution. In order for the dynamical system to make sense, $\mathbf{M}(\mathbf{x})$ must have a unique stationary distribution. We formalize this assumption for the spacey random walker in section 2.4.

PROPOSITION 1. *The spacey random walk is a vertex-reinforced random walk defined by the map*

$$\mathbf{x} \mapsto \sum_{k=1}^N \mathbf{P}_{\bullet\bullet k} \mathbf{x}_k = \mathbf{R} \cdot (\mathbf{x} \otimes \mathbf{I}).$$

Proof. Given $Y(n) = k$, the walker transitions from $X(n) = j$ to $X(n + 1) = i$ with probability \mathbf{P}_{ijk} . Conditioning on $Y(n)$,

$$\begin{aligned} & \Pr\{X(n + 1) = i \mid X(0), \dots, X(n), \mathbf{w}(n)\} \\ &= \sum_{k=1}^N \Pr\{X(n + 1) = i \mid X(n), Y(n) = k\} \Pr\{Y(n) = k \mid \mathbf{w}(n)\} \\ &= \sum_{k=1}^N \mathbf{P}_{i, X(n), k} \mathbf{w}_k(n) = [\mathbf{R} \cdot (\mathbf{w}(n) \otimes \mathbf{I})]_{i, X(n)}. \end{aligned}$$

Hence, given $\mathbf{w}(n)$, transitions are based on the matrix $\sum_{k=1}^N \mathbf{P}_{\bullet\bullet k} \mathbf{w}_k(n)$. Since $\mathbf{w}(n) \in \Delta_{N-1}$ and each $\mathbf{P}_{\bullet\bullet k}$ is a transition matrix, this convex combination is also a transition matrix. \square

The set of stationary distributions for the spacey random walk are the fixed points of the dynamical system in (10) using the correspondence to the spacey random walk map. These are vectors $\mathbf{x} \in \Delta_{N-1}$ for which

$$(11) \quad 0 = \frac{d\mathbf{x}}{dt} = \pi(\mathbf{R} \cdot (\mathbf{x} \otimes \mathbf{I})) - \mathbf{x} \iff \mathbf{R} \cdot (\mathbf{x} \otimes \mathbf{x}) = \mathbf{x} \iff \mathbf{x}_i = \sum_{ij} \mathbf{P}_{ijk} \mathbf{x}_j \mathbf{x}_k.$$

In other words, fixed points of the dynamical system are the z eigenvectors of \mathbf{P} that are nonnegative and sum to one, i.e., satisfy (3).

2.2. Intuition for Stationary Distributions. We now develop some intuition for why (3) must be satisfied for any stationary distribution of a spacey random walk without concerning ourselves with the formal limits and precise arguments, which are provided in section 5.1. Let $\mathbf{w}(n)$ be the occupation vector at step n . Consider the behavior of the spacey random walk process at some time $n \gg 1$ and some time $n + L$ where $n \gg L \gg 1$. The idea is to approximate what would happen if we ran the process for an extremely long time and then look at what changed at some large distance in the future. Since $L \gg 1$ but $L \ll n$, the vector $\mathbf{w}(L + n) \approx \mathbf{w}(n)$ and, thus, the spacey random walker $\{X(n)\}$ approximates a Markov chain with transition matrix

$$\mathbf{M}(\mathbf{w}(n)) = \mathbf{R} \cdot (\mathbf{w}(n) \otimes \mathbf{I}) = \sum_k \mathbf{P}_{ijk} \mathbf{w}_k(n).$$

Suppose that $\mathbf{M}(\mathbf{w}(n))$ has a unique stationary distribution $\bar{\mathbf{x}}(n)$ satisfying $\mathbf{M}(\mathbf{w}(n))\bar{\mathbf{x}}(n) = \bar{\mathbf{x}}(n)$. Then, if the process $\{X(n)\}$ has a limiting distribution, we must have $\bar{\mathbf{x}}(n) = \bar{\mathbf{x}}(n + L)$; otherwise, the distribution $\bar{\mathbf{x}}(n)$ will cause $\mathbf{w}(n + L)$ to change. Thus, the limiting distribution $\bar{\mathbf{x}}$ heuristically satisfies

$$(12) \quad \bar{\mathbf{x}} = \mathbf{M}(\bar{\mathbf{x}})\bar{\mathbf{x}} = \mathbf{R} \cdot (\bar{\mathbf{x}} \otimes \mathbf{I})\bar{\mathbf{x}} = \mathbf{R} \cdot (\bar{\mathbf{x}} \otimes \bar{\mathbf{x}}).$$

Based on this heuristic argument, then, we expect stationary distributions of spacey random walks to satisfy the system of polynomial equations

$$(13) \quad \mathbf{x}_i = \sum_{1 \leq j, k \leq N} \mathbf{P}_{ijk} \mathbf{x}_j \mathbf{x}_k.$$

In other words, \mathbf{x} is a z eigenvector of \mathbf{P} . Furthermore, \mathbf{x} is the Perron vector of $\mathbf{M}(\mathbf{w}(n))$, so it satisfies (3).

Pemantle further develops these heuristics by considering the change to $\mathbf{w}(n)$ induced by $\mathbf{x}(n)$ in a continuous time limit (Pemantle, 1992). To do this, note that, for the case $n \gg L \gg 1$,

$$\mathbf{w}(n + L) \approx \frac{n\mathbf{w}(n) + L\mathbf{x}(n)}{n + L} = \mathbf{w}(n) + \frac{L}{n + L}(\mathbf{x}(n) - \mathbf{w}(n)).$$

Thus, in a continuous time limit $L \rightarrow 0$ we have

$$\frac{d\mathbf{w}(n)}{dL} \approx \lim_{L \rightarrow 0} \frac{\mathbf{w}(n + L) - \mathbf{w}(n)}{L} = \frac{1}{n}(\mathbf{x}(n) - \mathbf{w}(n)).$$

Again, we arrive at the condition that, if this process converges, it must converge to a point where $\mathbf{x}(n) = \mathbf{w}(n)$.

2.3. Generalizations. All of our notions generalize to higher-order Markov chains beyond second-order. Consider an order- m hypermatrix $\underline{\mathbf{P}}$ representing an $(m - 1)$ th-order Markov chain. The spacey random walk corresponds to the following process:

1. The walker is at node i , spaces out, and forgets the last $m - 2$ states. It chooses the last $m - 2$ states j_1, \dots, j_{m-2} at random based on its history (each state is drawn i.i.d. from the occupation vector \mathbf{w}).
 2. The walker then transitions to node i with probability $\underline{\mathbf{P}}_{ij_1 \dots j_{m-2}}$, $1 \leq i \leq N$.
- Analogously to Proposition 1, the corresponding vertex-reinforced random walk map is

$$\mathbf{x} \mapsto \mathbf{R} \cdot \underbrace{(\mathbf{x} \otimes \dots \otimes \mathbf{x})}_{m-2 \text{ terms}} \otimes \mathbf{I},$$

where \mathbf{R} is the $N \times N^{m-1}$ column-stochastic flattening of $\underline{\mathbf{P}}$ along the first index.

One natural generalization to the spacey random walker is the *spacey random surfer*, following the model in the seminal paper by Page et al. (1999). In this case, the spacey random surfer follows the spacey random walker model with probability α and teleports to a random state with probability $(1 - \alpha)$; formally,

$$\Pr\{X(n + 1) = i \mid X(n) = j, Y(n) = k\} = \alpha \underline{\mathbf{P}}_{ijk} + (1 - \alpha) \mathbf{v}_i,$$

where \mathbf{v} is the (stochastic) teleportation vector. We will use this model to refine our analysis. Note that the spacey random surfer model is an instance of the spacey random walker model with transition probabilities $\alpha \underline{\mathbf{P}}_{ijk} + (1 - \alpha) \mathbf{v}_i$. This case was studied more extensively in Gleich, Lim, and Yu (2015).

2.4. Property B for Spacey Random Walks. Recall that in order to use the theory from Benaïm (1997) and the relationship between the stochastic process and the dynamical system, we must have that $\mathbf{M}(\mathbf{x})$ has a unique stationary distribution. We formalize this requirement in the following definition.

DEFINITION 2 (Property B). *We say that a spacey random walk transition hypermatrix $\underline{\mathbf{P}}$ satisfies Property B if the corresponding vertex-reinforced random walk matrix $\mathbf{M}(\mathbf{w})$ has a unique Perron vector when \mathbf{w} is on the interior of the probability simplex Δ_{N-1} .*

This property is trivially satisfied if $\underline{\mathbf{P}}$ is strictly positive. We can use standard properties of Markov chains to generalize the case when Property B holds.

THEOREM 3. *A spacey random walk transition hypermatrix $\underline{\mathbf{P}}$ satisfies Property B if and only if $\mathbf{M}(\mathbf{w})$ has a single recurrent class for some strictly positive vector $\mathbf{w} > 0$.*

Proof. Property B requires a unique stationary distribution for each \mathbf{w} on the interior of the probability simplex. This is equivalent to requiring a single recurrent class in the matrix $\mathbf{M}(\mathbf{w})$. Now, the property of having a single recurrent class is determined by the *graph of the nonzero elements* of $\mathbf{M}(\mathbf{w})$ alone and does not depend on their value. For any vector \mathbf{w} on the interior of the simplex, and also any positive vector \mathbf{w} , the resulting graph structure of $\mathbf{M}(\mathbf{w})$ is then the same. This graph structure is given by the *union* of graphs formed by each panel of \mathbf{R} that results when $\underline{\mathbf{P}}$ is flattened along the first index. Consequently, it suffices to test any strictly positive $\mathbf{w} > 0$ and test the resulting graph for a single recurrent class. \square

3. Applications. We now propose several applications of the spacey random walk and show how some existing models are instances of spacey random walks. We remind

the reader that in all of these applications, the entries of the hypermatrix \underline{P} come from a higher-order Markov chain, but the spacey random walk describes the dynamics of the process.

3.1. Population Genetics. In evolutionary biology, population genetics is the study of the dynamics of the distribution of alleles (types). Suppose we have a population with n types. We can describe the passing of type in a mating process by a hypermatrix of probabilities:

$$\Pr\{\text{child is type } i \mid \text{parents are types } j \text{ and } k\} = \underline{P}_{ijk}.$$

The process exhibits a natural symmetry $\underline{P}_{ijk} = \underline{P}_{ikj}$ because the order of the parent types does not matter. The spacey random walk traces the lineages of a type in the population with a random mating process:

1. A parent of type j randomly chooses a mate of type k from the population, whose types are distributed from the occupation vector \mathbf{w} .
2. The parents create a child of type i with probability \underline{P}_{ijk} .
3. The process repeats with the child becoming the parent.

A stationary distribution \mathbf{x} for the type distribution satisfies the polynomial equations $\mathbf{x}_i = \sum_k \underline{P}_{ijk} \mathbf{x}_j \mathbf{x}_k$. In population genetics, this is known as the Hardy–Weinberg equilibrium (Hartl and Clark, 1997). Furthermore, the dynamics of the spacey random walk describe how the distribution of types evolves over time. Our results offer new insights into how these population distributions evolve as well as how one might learn \underline{P} from data.

3.2. Transportation. We consider the process of taxis driving passengers around to a set of locations. Here, second-order information can be significant. For example, it is common for passengers to make length-2 cycles: to and from the airport, to and from dinner, etc. Suppose we have a population of passengers whose base (home) location is from a set $\{1, \dots, N\}$ of locations. Then we can model a taxi’s process as follows:

1. A passenger with base location k is drawn at random.
2. The taxi picks up the passenger at location j .
3. The taxi drives the passenger to location i with probability \underline{P}_{ijk} .

We can model the distribution of base locations empirically, i.e., the probability that a passenger has base location k is simply relative to the number of times that the taxi visits location k . In this model, the stochastic process of the taxi locations follows a spacey random walk. We explore this application further in section 6.3.

3.3. Ranking and Clustering. In data mining and information retrieval, a fundamental problem is ranking a set of items by relevance, popularity, importance, etc. (Manning et al., 2008). Recent work by Gleich, Lim, and Yu (2015) extends the classical PageRank method (Page et al., 1999) to “multilinear PageRank,” where the stationary distribution is the solution to the z eigenvalue problem $\mathbf{x} = \alpha \underline{R}(\mathbf{x} \otimes \mathbf{x}) + (1 - \alpha) \mathbf{v}$ for some teleportation vector \mathbf{v} . This is the spacey random surfer process discussed in section 2.3. As discussed in section 5, the multilinear PageRank vector corresponds to the fraction of time spent at each node in the spacey random surfer model. In related work, Mei, Guo, and Radev (2010) use the stationary distribution of Pemantle’s vertex-reinforced random walk to improve rankings.

Another fundamental problem in data mining is network clustering, i.e., partitioning the nodes of a graph into clusters of similar nodes. The definition of “similar” varies widely in the literature (Schaeffer, 2007; Fortunato, 2010), but almost all of

the definitions involve first-order (edge-based) Markovian properties. Drawing on connections between random walks and clustering, Benson, Gleich, and Leskovec (2015) used the multilinear PageRank vector \mathbf{x} to partition the graph based on *motifs*, i.e., patterns on several nodes (Alon, 2007). In this application, the hypermatrix $\underline{\mathbf{P}}$ encodes transitions on motifs.

3.4. Pólya Urn Processes. Our spacey random walk model is a generalization of a Pólya urn process, which we illustrate with a simple example. Consider an urn with red and green balls. At each step, we (i) draw a random ball from the urn, (ii) put the randomly drawn ball back in, and (iii) put another ball of the same color into the urn. Here, we consider the sequence of states $X(n)$ to be the color of the ball put in the urn in step (iii). The transitions are summarized as follows:

Last ball selected	Randomly drawn ball	
	Red	Green
Red	Red	Green
Green	Red	Green

In this urn model, the spacey random walk transition probabilities are independent of the last state, i.e.,

$$(14) \quad \Pr\{X(n+1) = i \mid X(n) = j, Y(n) = k\} = \Pr\{X(n+1) = i \mid Y(n) = k\}.$$

Nonetheless, this is still a spacey random walk where the transition hypermatrix is given by the following flattening:

$$(15) \quad \mathbf{R} = \left[\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{array} \right].$$

By (14), each panel (representing the randomly drawn ball) has a single row of ones.

We can generalize to more exotic urn models. Consider an urn with red and green balls. At each step, we (i) draw a sequence of balls b_1, \dots, b_m , (ii) put the balls back in, and (iii) put a new ball of color $C(b_1, \dots, b_m) \in \{\text{red, green}\}$ into the urn. This process can be represented by a 2×2^m flattening \mathbf{R} , where the panels are indexed by (b_1, \dots, b_m) :

$$\mathbf{R}_{b_1, \dots, b_m} = \left[\begin{array}{cc} \text{Ind}\{C(b_1, \dots, b_m) = \text{red}\} & \text{Ind}\{C(b_1, \dots, b_m) = \text{red}\} \\ \text{Ind}\{C(b_1, \dots, b_m) = \text{green}\} & \text{Ind}\{C(b_1, \dots, b_m) = \text{green}\} \end{array} \right].$$

Furthermore, the spacey random walk also describes urn models with probabilistic choices for the color. For example, suppose we draw a sequence of balls b_1, \dots, b_m and add a new red ball into the urn with probability $p(b_1, \dots, b_m)$ and a green ball with probability $1 - p(b_1, \dots, b_m)$. Then the panels of \mathbf{R} would be

$$\mathbf{R}_{b_1, \dots, b_m} = \left[\begin{array}{cc} p(b_1, \dots, b_m) & p(b_1, \dots, b_m) \\ 1 - p(b_1, \dots, b_m) & 1 - p(b_1, \dots, b_m) \end{array} \right].$$

Although these two-color urn processes can be quite complicated, our analysis in section 4 shows that, apart from easily identifiable corner cases, the dynamics of the system always converge to a stable equilibrium point. In other words, the fraction of red balls in the urn converges.

4. Dynamics with Two States. We now completely characterize the spacey random walk for the simple case where there are only two states. This characterization covers the Pólya urn process described in section 3.4.

4.1. The General Two-State Model. In two-state models, the probability distribution over states is determined by the probability of being in any one of the two states. This greatly simplifies the dynamics. In fact, we show in Theorem 6 that in all cases, the dynamical system describing the trajectory of the spacey random walker converges to a stable stationary point.

Consider the case of an order- m hypermatrix \underline{P} with each dimension equal to two. Let \mathbf{R} be the flattening of \underline{P} along the first index. We define the map \mathbf{z} that sends a single probability to the probability simplex on two points:

$$\mathbf{z}: [0, 1] \rightarrow \Delta_1, \quad x \mapsto \begin{bmatrix} x \\ 1 - x \end{bmatrix}.$$

REMARK 4. *Since the identity is the only 2×2 stochastic matrix without a unique Perron vector, Property B (Definition 2) can be reduced to*

$$x \in (0, 1) \rightarrow \mathbf{R} \cdot \underbrace{(\mathbf{z}(x) \otimes \cdots \otimes \mathbf{z}(x) \otimes \mathbf{I})}_{m - 2 \text{ terms}} \neq \mathbf{I}$$

in the case of a two-state spacey random walk.

We have a closed form for the map π that sends a 2×2 stochastic matrix, with a unique Perron vector, to the first coordinate of its Perron vector. (As noted in the remark, a 2×2 stochastic matrix has a unique Perron vector if it is not the identity, so π applies to all 2×2 stochastic matrices that are not the identity matrix.) Specifically,

$$\pi \left(\begin{bmatrix} p & 1 - q \\ 1 - p & q \end{bmatrix} \right) = \mathbf{z} \left(\frac{1 - q}{2 - p - q} \right).$$

The corresponding dynamical system on the first coordinate is then

$$(16) \quad \frac{dx}{dt} = \left[\pi \left(\mathbf{R} \cdot \underbrace{(\mathbf{z}(x) \otimes \cdots \otimes \mathbf{z}(x) \otimes \mathbf{I})}_{m - 2 \text{ terms}} \right) \right]_1 - x.$$

When the dependence on \mathbf{R} is clear from the context, we will write this dynamical system as $dx/dt = f(x)$ to reduce the notation in the statements of our theorems.

Figure 3 shows the dynamics for the following flattening of a fourth-order hypermatrix:

$$\mathbf{R} = \left[\begin{array}{cc|cc|cc|cc} 0.925 & 0.925 & 0.925 & 0.075 & 0.925 & 0.075 & 0.075 & 0.075 \\ 0.075 & 0.075 & 0.075 & 0.925 & 0.075 & 0.925 & 0.925 & 0.925 \end{array} \right].$$

The system has three equilibrium points, i.e., there are three values of x such that $f(x) = 0$ on the interval $[0, 1]$. Two of the points are stable in the sense that small enough perturbations to the equilibrium value will always result in the system returning to the same point.

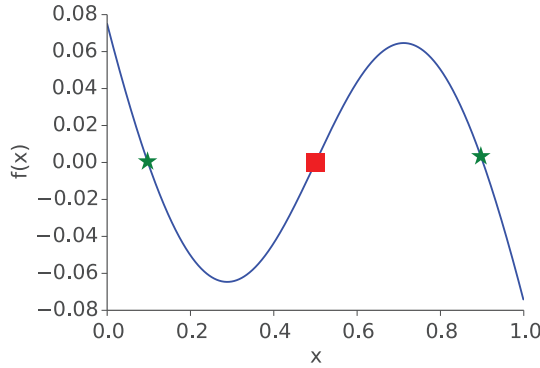


Fig. 3 The dynamics of the spacey random walk for a $2 \times 2 \times 2 \times 2$ hypermatrix shows that the solution has three equilibrium points (marked where $f(x) = 0$) and two stable points (the green stars).

4.2. Existence of Stable Equilibria. The behavior in Figure 3 shows that $f(0)$ is positive and $f(1)$ is negative. This implies the existence of at least one equilibrium point on the interior of the region (in the figure, there are three). We now show that these boundary conditions always hold.

LEMMA 5. Consider an order- m hypermatrix \underline{P} with $N = 2$ that satisfies Property B. The forcing function $f(x)$ for the dynamical system has the following properties:

- (i) $f(0) \geq 0$;
- (ii) $f(1) \leq 0$.

Proof. Write the flattening of \underline{P} as \mathbf{R} with panels $\mathbf{R}_1, \dots, \mathbf{R}_M$ for $M = 2^{m-2}$. For part (i), when $x = 0$, $\mathbf{z}(x) = [0 \ 1]^T$ and

$$\mathbf{R} \cdot \underbrace{(\mathbf{z}(x) \otimes \dots \otimes \mathbf{z}(x))}_{m-2 \text{ terms}} \otimes \mathbf{I} = \mathbf{R}_M = \begin{bmatrix} p & 1-q \\ 1-p & q \end{bmatrix}$$

for some p and q . If p and q are both not equal to 1, then $f(0) \geq 0$. If $p = q = 1$, then we define $f(x)$ via its limit as $x \rightarrow 0$. We have

$$f(\epsilon) = \left[\pi \left((1-\epsilon) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \epsilon \begin{bmatrix} a & b \\ 1-a & 1-b \end{bmatrix} \right) \right]_1 - \epsilon = \frac{b}{1-a+b} - \epsilon.$$

By Property B, $(a, b) \neq (1, 0)$ for any ϵ , so $\lim_{\epsilon \rightarrow 0} f(\epsilon) \geq 0$.

For part (ii), we have the same argument except, given that we are subtracting 1, the result is always nonpositive. \square

We now show that the dynamical system for the spacey random walker with two states always has a stable equilibrium point.

THEOREM 6. Consider a hypermatrix \underline{P} that satisfies Property B with $N = 2$. The dynamical system given by (10) always converges to a Lyapunov stable point starting from any $x \in (0, 1)$.

Proof. Note that $f(x)$ is a ratio of two polynomials and that it has no singularities on the interval $(0, 1)$ by Property B. Because $f(x)$ is a continuous function on this interval, we immediately have the existence of at least one equilibrium point by Lemma 5.

If either $f(0) > 0$ or $f(1) < 0$, then at least one of the equilibrium points must be stable. Consider the case when both $f(0) = 0$ and $f(1) = 0$. If $f(x)$ changes sign on the interior, then either 0, 1, or the sign change must be a stable point. If $f(x)$ does not change sign on the interior, then either 0 or 1 is a stable point depending on the sign of f on the interior. \square

We note that it is possible for $f(x) \equiv 0$ on the interval $[0, 1]$. For example, this occurs with the deterministic Pólya urn process described by the flattened hypermatrix in (15). However, Lyapunov stability only requires that, after perturbation from a stationary point, the dynamics converge to a nearby stationary point and not necessarily to the *same* stationary point. In this sense, any point on the interval is a Lyapunov stable point when $f(x) \equiv 0$.

4.3. The $2 \times 2 \times 2$ case. We now investigate the dynamics of the spacey random walker for an order-3 hypermatrix \underline{P} with flattening

$$\mathbf{R} = \left[\begin{array}{cc|cc} a & b & c & d \\ 1-a & 1-b & 1-c & 1-d \end{array} \right]$$

for arbitrary scalars $a, b, c, d \in [0, 1]$. In this case, the dynamical system is simple enough to explore analytically. Given any $x \in [0, 1]$,

$$\begin{aligned} \mathbf{R} \cdot (\mathbf{z}(x) \otimes \mathbf{I}) &= x \begin{bmatrix} a & b \\ 1-a & 1-b \end{bmatrix} + (1-x) \begin{bmatrix} c & d \\ 1-c & 1-d \end{bmatrix} \\ &= \begin{bmatrix} c-x(c-a) & d-x(d-b) \\ 1-c+x(c-a) & 1-d+x(d-b) \end{bmatrix}, \end{aligned}$$

and the corresponding dynamical system is

$$\begin{aligned} \frac{dx}{dt} &= [\pi(\mathbf{R} \cdot (\mathbf{z}(x) \otimes \mathbf{I}))]_1 - x = \frac{d-x(d-b)}{1-c+d+x(c-a+b-d)} - x \\ (17) \qquad \qquad \qquad &= \frac{d+(b+c-2d-1)x+(a-c+d-b)x^2}{1-c+d+x(c-a+b-d)}. \end{aligned}$$

Since the right-hand side is independent of t , this differential equation is separable:

$$\int_{x_0}^x \frac{1-c-d+(c-a+d-b)x}{d+(b+c-2d-1)x+(a-c+d-b)x^2} dx = \int_0^t dt,$$

where $x(0) = x_0$. Evaluating the integral,

$$\begin{aligned} (18) \qquad & \int \frac{\delta x + \epsilon}{\alpha x^2 + \beta x + \gamma} dx \\ &= \begin{cases} \frac{\delta}{2\alpha} \log |\alpha x^2 + \beta x + \gamma| + \frac{2\alpha\epsilon - \beta\delta}{\alpha\sqrt{4\alpha\gamma - \beta^2}} \tan^{-1} \left(\frac{2\alpha x + \beta}{\sqrt{4\alpha\gamma - \beta^2}} \right) + C & \text{for } 4\alpha\gamma - \beta^2 > 0, \\ \frac{\delta}{2\alpha} \log |\alpha x^2 + \beta x + \gamma| - \frac{(2\alpha\epsilon - \beta\delta) \tanh^{-1} \left(\frac{2\alpha x + \beta}{\sqrt{\beta^2 - 4\alpha\gamma}} \right)}{\alpha\sqrt{\beta^2 - 4\alpha\gamma}} + C & \text{for } 4\alpha\gamma - \beta^2 < 0, \\ \frac{\delta}{2\alpha} \log |\alpha x^2 + \beta x + \gamma| - \frac{2\alpha\epsilon - \beta\delta}{\alpha(2\alpha x + \beta)} + C & \text{for } 4\alpha\gamma - \beta^2 = 0, \end{cases} \end{aligned}$$

where

$$\alpha = a - c + d - b, \quad \beta = b + c - 2d - 1, \quad \gamma = d, \quad \delta = -c - a + d - b, \quad \epsilon = 1 - c - d,$$

and C is a constant determined by x_0 . Equation (17) also gives us an explicit formula for the equilibrium points.

PROPOSITION 7. *The dynamics are at equilibrium ($\frac{dx}{dt} = 0$) if and only if*

$$x = \begin{cases} \frac{1 + 2d - b - c \pm \sqrt{(b + c - 2d - 1)^2 - 4(a - c + d - b)d}}{2(a - c + d - b)} & \text{for } a + d \neq b + c, \\ \frac{d}{1 + 2d - b - c} & \text{for } a + d = b + c, \\ & b + c \neq 1 + 2d, \end{cases}$$

or for any $x \in [0, 1]$ if $a + d = b + c$ and $b + c = 1 + 2d$. In this latter case, the transitions have the form

$$(19) \quad \mathbf{R} = \left[\begin{array}{cc|cc} 1 & b & 1 - b & 0 \\ 0 & 1 - b & b & 1 \end{array} \right].$$

Proof. The first case follows from the quadratic formula. If $a + d = b + c$, then an equilibrium point occurs if and only if $d + (b + c - 2d - 1)x = 0$. If $b + c \neq 1 + 2d$, then we get the second case. Otherwise, $a = 1 + d$, which implies that $a = 1$ and $d = 0$. Subsequently, $b + c = 1$. \square

Of course, only the roots $x \in [0, 1]$ matter for the general case $a + d \neq b + c$. Also, by Property B, we know that the value of b in (19) is not equal to 0. Finally, we note that for the Pólya urn process described in section 3.4, $a = b = 1$ and $c = d = 0$, so the dynamics fall into the regime where any value of x is an equilibrium point, i.e., $dx/dt = 0$.

5. Dynamics with Many States: Limiting Distributions and Computation.

Now that we have covered the full dynamics of the two-state case, we analyze the case of several states. Theorem 9 in section 5.1 formalizes the intuition from section 2.2 that any stationary distribution \mathbf{x} of a spacey random walk for a higher-order Markov chain $\underline{\mathbf{P}}$ must be a z eigenvector of $\underline{\mathbf{P}}$ (i.e., satisfy (3)). Furthermore, for any higher-order Markov chain, there always exists a stochastic z eigenvector of the corresponding hypermatrix $\underline{\mathbf{P}}$ (Li and Ng, 2014, Theorem 2.2). However, this does not guarantee that any spacey random walk will converge.

Subsequently, we study the relationship between numerical methods for computing z eigenvectors and the convergence of spacey random walks and find two conclusions. First, the dynamics of the spacey random walk may converge even if the commonly-used power method for computing the z eigenvector does not (Observation 10). On the other hand, in regimes where the power method is guaranteed to converge, the dynamics of the spacey random surfer will also converge (Theorem 13). A complete characterization of the computation of solutions to the algebraic sets of equations for the z eigenvector in (3) and the convergence of the continuous dynamical system is an open question for future research.

5.1. Limiting Behavior of the Vertex-Reinforced Random Walk.

We now summarize Benaïm’s key result for relating the limiting distribution of the occupation vector \mathbf{w} for a vertex-reinforced random walk to the dynamical system. Recall from

(10) that for an arbitrary vertex-reinforced random walk, the dynamical system is defined by $d\mathbf{x}/dt = \pi(\mathbf{M}(\mathbf{x})) - \mathbf{x} = f(\mathbf{x})$. Since the function is smooth and Δ is invariant under f , it generates a flow $\Phi: \mathbb{R}_+ \times \Delta \rightarrow \Delta$, where $\Phi(t, \mathbf{u})$ is the value at time t of the initial value problem with $\mathbf{x}(0) = \mathbf{u}$ and dynamics f . A continuous function $X: \mathbb{R}_+ \rightarrow \Delta$ is said to be an *asymptotic pseudotrajectory* (Benaïm and Hirsch, 1996) of Φ if, for any $L \in \mathbb{R}_+$,

$$\lim_{t \rightarrow \infty} \|X(t + L) - \Phi(L, \mathbf{x}(t))\| = 0$$

locally uniformly. The following theorem, proved by Benaïm (1997), provides an asymptotic pseudotrajectory of the dynamical system in terms of the occupation vector \mathbf{w} of a vertex-reinforced random walk.

THEOREM 8 (Benaïm (1997)). *Let $\tau_0 = 0$ and $\tau_n = \sum_{i=1}^n 1/(i + 1)$. Define the continuous function W by $W(\tau_n) = \mathbf{w}(n)$ and W affine on $[\tau_n, \tau_{n+1}]$. In other words, W linearly interpolates the occupation vector on decaying time intervals. Then W is almost surely an asymptotic pseudotrajectory of Φ .*

Next, we place Benaïm’s results in the context of spacey random walks. Let \mathbf{R} be the flattening of an order- m hypermatrix \mathbf{P} of dimension n representing the transition probabilities of an m th-order Markov chain, and let \mathbf{w} be the (random) occupation vector of a spacey random walk following \mathbf{R} .

THEOREM 9. *Suppose that \mathbf{R} satisfies Property B. Then we can define a flow $\Phi_S: \mathbb{R}_+ \times \Delta_{N-1} \rightarrow \Delta_{N-1}$, where $\Phi_S(t, \mathbf{u})$ is the solution of the initial value problem*

$$\frac{d\mathbf{x}}{dt} = \pi(\mathbf{R} \cdot \underbrace{(\mathbf{x} \otimes \cdots \otimes \mathbf{x} \otimes \mathbf{I})}_{m-2 \text{ terms}}) - \mathbf{x}, \quad \mathbf{x}(0) = \mathbf{u}.$$

Then, for any $L \in \mathbb{R}_+$, $\lim_{t \rightarrow \infty} \|W(t + L) - \Phi_S(L, \mathbf{x}(t))\| = 0$ locally uniformly, where W is defined as in Theorem 8. In particular, if the dynamical system of the spacey random walk converges, then the occupation converges to a stochastic z eigenvector \mathbf{x} of \mathbf{P} .

This result does not say anything about the existence of a limiting distribution, the uniqueness of a limiting distribution, or computational methods for computing a stationary vector. We explore these issues in the following sections.

5.2. Power Method. We now turn to numerical solutions for finding a stationary distribution of the spacey random walk process. In terms of the tensor eigenvector formulation, an attractive, practical method is a “power method” or “fixed point” iteration (Li and Ng, 2014; Chu and Wu, 2014; Gleich, Lim, and Yu, 2015):

$$\mathbf{x}(n + 1) = \mathbf{R} \cdot (\mathbf{x}(n) \otimes \mathbf{x}(n)), \quad n = 0, 1, \dots$$

Unfortunately, the power method does not always converge. We provide one example from Chu and Wu (2014), given by the flattening

$$(20) \quad \mathbf{R} = \left[\begin{array}{cc|cc} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{array} \right].$$

Here we prove directly that the power method diverges. (Divergence is also a consequence of the spectral properties of \mathbf{R} (Chu and Wu, 2014).) The steady-state solution is given by $\mathbf{x}_1 = (\sqrt{5} - 1)/2$ and $\mathbf{x}_2 = (3 - \sqrt{5})/2$. Let $a(n)$ be the first

component of $\mathbf{x}(n)$ so that $\mathbf{x}(n) = [a(n) \ 1 - a(n)]^T$. Then the iterates are given by $a(n+1) = 1 - a(n)^2$. Let $a(n) = \mathbf{x}_1 - \epsilon$. Then

$$a(n+1) = 1 - (x_1 - \epsilon)^2 \approx 1 - \mathbf{x}_1^2 + 2\epsilon\mathbf{x}_1 = \mathbf{x}_1 + 2\epsilon\mathbf{x}_1,$$

and the iterate moves further from the solution,

$$|a(n+1) - \mathbf{x}_1| = 2\epsilon\mathbf{x}_1 > \epsilon = |a(n) - \mathbf{x}_1|.$$

Next, note that $a(n+2) = (1 - (1 - a(n))^2)^2 \geq a(n)$ for $a(n) \geq \mathbf{x}_1$. Furthermore, this inequality is strict when $a(n) > \mathbf{x}_1$. Thus, starting at $a(n) = \mathbf{x}_1 + \epsilon$ will not result in convergence. We conclude that the power method does not converge in this case. However, since this is a two-state system, the dynamical system for the corresponding hypermatrix must converge to a stable point by Theorem 6. We summarize this result in the following observation.

OBSERVATION 10. *Convergence of the dynamical system to a stable equilibrium does not imply convergence of the power method to a tensor eigenvector.*

5.3. Numerical Convergence of the Dynamical System for the Spacey Random Surfer Case. We will use the case of the spacey random surfer process (section 2.3) in many of the following examples because it has enough structure to illustrate a definite stationary distribution. Recall that in this process, the walker transitions to a random node with probability $1 - \alpha$ following the teleportation vector \mathbf{v} .

REMARK 11. *For any $\alpha < 1$, the spacey random surfer process satisfies Property B for any stochastic teleportation vector \mathbf{v} .*

This remark follows because for any occupation vector, the matrix $\mathbf{M}(\mathbf{w}(n))$ for the spacey random surfer corresponds to the convex combination of a column-stochastic matrix and a rank-1 column-stochastic matrix. The rank-1 component emerges from the teleportation due to the surfer. These convex combinations are equivalent to the Markov chains that occur in Google's PageRank models. They have only a single recurrent class determined by the location of the teleportation. By way of intuition, just consider the places that could be visiting *starting* from anywhere the teleportation vector is nonzero. This class recurs because there is always a finite probability of restarting. Thus, the matrix $\mathbf{M}(\mathbf{w}(n))$ always has a unique stationary distribution (Berman and Plemmons, 1994, Theorem 3.23).

Gleich, Lim, and Yu (2015) design and analyze several computational methods for the tensor eigenvector problem in (12). We summarize their results on the power method for the spacey random surfer model.

THEOREM 12 (Gleich, Lim, and Yu (2015)). *Let $\underline{\mathbf{P}}$ be an order- m hypermatrix for a spacey random surfer model with $\alpha < 1/(m-1)$ and consider the power method initialized with the teleportation vector, i.e., $\mathbf{x}(0) = \mathbf{v}$.*

1. *There is a unique solution \mathbf{x} to (12).*
2. *The iterate residuals are bounded by $\|\mathbf{x}(n) - \mathbf{x}\|_1 \leq 2[\alpha(m-1)]^n$.*

These previous results reflect statements about the tensor eigenvector problem, but not the stationary distribution of the spacey random surfer. In the course of the new results below, we establish a sufficient condition for the spacey random surfer process to have a stationary distribution.

We analyze the forward Euler scheme for numerically solving the dynamical system of the spacey random surfer. Our analysis depends on the parameter α in this model,

and, in fact, we find the same α dependence as in Theorem 12. The following result gives conditions for when the forward Euler method will converge and also when the stationary distribution exists.

THEOREM 13. *Suppose there is a fixed point \mathbf{x} of the dynamical system. The time-stepping scheme $\mathbf{x}(n+1) = hf(\mathbf{x}(n)) + \mathbf{x}(n)$ to numerically solve the dynamical system in (10) converges to \mathbf{x} when $\alpha < 1/(m-1)$ and $h \leq (1-\alpha)/(1-(m-1)\alpha)$. In particular, since the forward Euler method converges as $h \rightarrow 0$, the spacey random surfer process always converges to a unique stationary distribution when $\alpha < 1/(m-1)$.*

Proof. Consider a transition probability hypermatrix $\underline{\mathbf{P}}$ representing the transitions of an m th-order Markov chain, and let \mathbf{R} be the one-mode unfolding of $\underline{\mathbf{P}}$. The fixed point \mathbf{x} satisfies

$$\mathbf{x} = \alpha \mathbf{R} \underbrace{(\mathbf{x} \otimes \cdots \otimes \mathbf{x})}_{m-1 \text{ terms}} + (1-\alpha)\mathbf{v}.$$

Let $\mathbf{y}(n)$ be the stationary distribution satisfying

$$\mathbf{y}(n) = \alpha \mathbf{R} \underbrace{(\mathbf{x}(n) \otimes \cdots \otimes \mathbf{x}(n) \otimes \mathbf{I})}_{m-2 \text{ terms}} \mathbf{y}(n) + (1-\alpha)\mathbf{v},$$

and let $\|\cdot\|$ denote the 1-norm. Subtracting \mathbf{x} from both sides,

$$\begin{aligned} \|\mathbf{y}(n) - \mathbf{x}\| &= \alpha \|\mathbf{R}\| \cdot \left\| \underbrace{\mathbf{x}(n) \otimes \cdots \otimes \mathbf{x}(n)}_{m-2 \text{ terms}} \otimes \mathbf{y}(n) - \underbrace{\mathbf{x} \otimes \cdots \otimes \mathbf{x}}_{m-1 \text{ terms}} \right\| \\ &\leq \alpha ((m-2)\|\mathbf{x}(n) - \mathbf{x}\| + \|\mathbf{y}(n) - \mathbf{x}\|). \end{aligned}$$

The inequality on the difference of Kronecker products follows from Gleich, Lim, and Yu (2015, Lemma 4.4). This gives us a bound on the distance between $\mathbf{y}(n)$ and the solution \mathbf{x} :

$$\|\mathbf{y}(n) - \mathbf{x}\| \leq \frac{(m-2)\alpha}{1-\alpha} \|\mathbf{x}(n) - \mathbf{x}\|.$$

Taking the next time step,

$$\begin{aligned} \mathbf{x}(n+1) - \mathbf{x} &= hf(\mathbf{x}(n)) + \mathbf{x}(n) - \mathbf{x} \\ &= h(\mathbf{y}(n) - \mathbf{x}) + (1-h)(\mathbf{x}(n) - \mathbf{x}). \end{aligned}$$

Finally, we bound the error on the distance between $\mathbf{x}(n)$ and the fixed point \mathbf{x} as a function of the previous time step:

$$(21) \quad \|\mathbf{x}(n+1) - \mathbf{x}\| \leq h \frac{(m-2)\alpha}{1-\alpha} \|\mathbf{x}(n) - \mathbf{x}\| + (1-h)\|\mathbf{x}(n) - \mathbf{x}\|$$

$$(22) \quad = \left(1 - \frac{h(1-(m-1)\alpha)}{1-\alpha} \right) \|\mathbf{x}(n) - \mathbf{x}\|.$$

Now fix $T = hn$ and let $\mathbf{u}(T)$ denote the continuous time solution to the initial value problem at time T . Then $\|\mathbf{u}(T) - \mathbf{x}\| \leq \|\mathbf{u}(T) - \mathbf{x}(n)\| + \|\mathbf{x}(n) - \mathbf{x}\|$. Since the dynamical system is Lipschitz continuous and Euler's method converges, the first term goes to 0 as $h \rightarrow 0$. Simultaneously taking the limit as $n \rightarrow \infty$, the second term is bounded by $2e^{-T(1-(m-1)\alpha)/(1-\alpha)}$ through (22). Thus, the dynamical system converges.

Uniqueness of the fixed point \mathbf{x} follows from Lemma 4.6 of Gleich, Lim, and Yu (2015). \square

We note that the forward Euler scheme is just one numerical method for analyzing the spacey random surfer process. However, this theorem gives the same convergence dependence on α as Theorem 12 gives for convergence of the power method.

6. Numerical Experiments on Trajectory Data. We now switch from developing theory to using the spacey random walk model to analyze data. In particular, we will use the spacey random walk to model trajectory data, i.e., sequences of states. We first show how to find the transition probabilities that maximize the likelihood of the spacey random walk model and then use the maximum likelihood framework to train the model on synthetically generated trajectories and a real-world dataset of New York City taxi trajectories.

6.1. Learning Transition Probabilities. Consider a spacey random walk model with a third-order transition probability hypermatrix and a trajectory $X(1), \dots, X(Q)$ on N states. With this data, we know the occupation vector $\mathbf{w}(n)$ at the n th step of the trajectory, but we do not know the true underlying transition probabilities. For any given transition probability hypermatrix $\underline{\mathbf{P}}$, the probability of a single transition is

$$\Pr\{X(n+1) = i \mid X(n) = j\} = \sum_{k=1}^N \Pr\{Y(n) = k\} \underline{\mathbf{P}}_{ijk} = \sum_{k=1}^N \mathbf{w}_k(n) \underline{\mathbf{P}}_{ijk}.$$

We can use maximum likelihood to estimate $\underline{\mathbf{P}}$. The maximum likelihood estimator is the minimizer of the following negative log-likelihood minimization problem:

$$(23) \quad \begin{aligned} & \underset{\underline{\mathbf{P}}}{\text{minimize}} && - \sum_{q=2}^Q \log \left(\sum_{k=1}^N \mathbf{w}_k(q-1) \underline{\mathbf{P}}_{X(q)X(q-1)k} \right) \\ & \text{subject to} && \sum_{i=1}^N \underline{\mathbf{P}}_{ijk} = 1, \quad 1 \leq j, k \leq N, \quad 0 \leq \underline{\mathbf{P}}_{ijk} \leq 1, \quad 1 \leq i, j, k \leq N. \end{aligned}$$

(Equation (23) represents negative log-likelihood minimization for a single trajectory; several trajectories introduces an addition summation.) Since $\mathbf{w}_k(q)$ is just a function of the data (the $X(n)$), the objective function is the sum of negative logs of an affine function of the optimization variables. Therefore, the objective is a smooth convex function. For our experiments in this article, we optimize this objective with a projected gradient descent algorithm. The projection step computes the minimal Euclidean distance projection of the columns $\underline{\mathbf{P}}_{\bullet jk}$ onto the simplex, which takes linear time (Duchi et al., 2008).

In the above formulation we assume that storage of $\underline{\mathbf{P}}$ is not an issue, which will be the case for our examples in the rest of this section. In practice, one could modify the optimization problem to learn a sparse or structured $\underline{\mathbf{P}}$.

6.2. Synthetic Trajectories. We first tested the maximum likelihood procedure on synthetically generated trajectories that follow the spacey random walk stochastic process. Our synthetic data was derived from two sources. First, we used two transition

probability hypermatrices of dimension $N = 4$ from Gleich, Lim, and Yu (2015):

$$\mathbf{R}_1 = \left[\begin{array}{cccc|cccc|cccc|cccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1/2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1/2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1/2 & 0 & 1 & 0 \end{array} \right],$$

$$\mathbf{R}_2 = \left[\begin{array}{cccc|cccc|cccc|cccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1/2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1/2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right].$$

Note that the transition graph induced by $\mathbf{R}_i \cdot (\mathbf{x} \otimes \mathbf{I})$ is strongly connected for any $\mathbf{x} > 0$ with self-loops on every node, $i = 1, 2$. Thus, by Theorem 3, both \mathbf{R}_1 and \mathbf{R}_2 satisfy Property B. Second, we generated random transition probability hypermatrices of the same dimension, where each column of the flattened hypermatrix is drawn uniformly at random from the simplex Δ_3 . We generated 20 of these hypermatrices. Since the entries in these hypermatrices are all positive, they all satisfy Property B.

For each transition probability hypermatrix, we generated 100 synthetic trajectories, each with 200 transitions. We used 80 of the trajectories as training data for our models and the remaining 20 trajectories as test data. We trained the following models:

- The spacey random walk model where the transition probability hypermatrix is estimated by solving the maximum likelihood optimization problem in (23) with projected gradient descent.
- A “zeroth-order” Markov chain where the probability of transitioning to a state is the empirical fraction of time spent at that state in the training data. We interpret this model as an “intelligent” random guessing baseline.
- A first-order Markov chain where the transitions are estimated empirically from the training data:

$$P_{ij} = \frac{\#(\text{transitions from } j \text{ to } i)}{\sum_l \#(\text{transitions from } j \text{ to } l)}.$$

It is well known that these transition probabilities are the maximum likelihood estimators for the transition probabilities of a first-order Markov chain.

- A second-order Markov chain where the transition probabilities are estimated empirically from the training data:

$$\underline{P}_{ijk} = \frac{\#(\text{transitions from } (j, k) \text{ to } (i, j))}{\sum_l \#(\text{transitions from } (j, k) \text{ to } (l, j))}.$$

As was the case for the first-order chain, these transition probabilities are the maximum likelihood estimators for the transition probabilities.

For every state visited in every trajectory of the test set, each model gives a probability p of transitioning to that state. We computed the root mean square error (RMSE), where the error contribution for each visited state is $1 - p$. The RMSEs are summarized in Table 2. We include the test errors for the “true” spacey random walk model, i.e., we use the known synthetic parameters for prediction. These errors represent the smallest errors we could expect from any model.

For the transition probability hypermatrices \mathbf{R}_1 and \mathbf{R}_2 , the spacey random walk clearly outperforms the Markov chains and has performance on par with the

Table 2 Root mean square error (RMSE) on three test datasets with zeroth-order (ZO), first-order (FO), and second-order (SO) Markov chain (MC) models and spacey random walk (SRW) models. The true SRW model was used to generate the training and test data. Thus, the RMSE for “true SRW” represents the best we can expect from any model. For the “Random” (Rand.) data, we list the mean error plus or minus one standard deviation over 20 datasets. For the constructed transition probability hypermatrices given by \mathbf{R}_1 and \mathbf{R}_2 , the learned SRW model outperforms the Markov chain models. For the randomly generated transitions, all the first- and second-order MC models and the learned SRW model predict as well as the true generative model.

	ZOMC	FOMC	SOMC	Learned SRW	True SRW
\mathbf{R}_1	0.690	0.465	0.457	0.434	0.434
\mathbf{R}_2	0.526	0.325	0.314	0.292	0.292
Rand.	0.743±0.007	0.718±0.010	0.718±0.010	0.717±0.011	0.717±0.011

true underlying model. At a high level, both \mathbf{R}_1 and \mathbf{R}_2 are adversarial to Markov chain modeling because they have multiple fixed points (vectors that satisfy (3)). For instance, the unit basis vectors \mathbf{e}_2 and \mathbf{e}_3 are each stationary points for both hypermatrices. In fact, through MATLAB’s symbolic toolbox, we found four stationary points for each of the hypermatrices. Although these fixed points may not be limiting distributions of the spacey random walk, their presence is evident in finite trajectories. For example, there are several trajectories with long sequences of state 2 or state 3 (corresponding to the \mathbf{e}_2 and \mathbf{e}_3 stationary points) in addition to more heterogeneous trajectories. The spacey random walk is able to understand these sequences because it models the occupation vector.

On the other hand, the random transition probability hypermatrices are much more amenable to modeling by Markov chains. Apart from the zeroth-order Markov chain, each model has roughly the same performance and predicts as well as the true model on the test data. We verified through MATLAB’s symbolic toolbox that each of the 20 random hypermatrices has exactly one stationary point. Therefore, we do not see the same effects of multiple stationary points on finite trajectories, as with \mathbf{R}_1 and \mathbf{R}_2 . Furthermore, if the spacey random walk process converges, it can only converge to a single first-order Markovian process.

6.3. Taxi Trajectories. Next, we modeled real-world transportation data with the spacey random walk process. We parsed taxi trajectories from a publicly available New York City taxi dataset.³ Our dataset consists of the sequence of locations for 1000 taxi drivers. (Recall from section 3.2 how the spacey random walk models this type of data.) We consider locations at the granularity of neighborhoods in Manhattan, of which there are 37 in our dataset. One of these states is an “outside Manhattan” state to capture locations outside of the borough. If a driver drops off a passenger in location j and picks up the next passenger in location j , we do not consider it a transition. However, if a driver picks up the next passenger in location $i \neq j$, we count the transition from location j to i . Figure 4 shows the lengths of the trajectories and the empirical geographic distribution of the taxis. Most trajectories contain at least 20,000 transitions, and the shortest trajectory contains 243 transitions. The geographic distribution shows that taxis frequent densely populated regions such as the Upper East Side and destination hubs like Midtown.

We again compared the spacey random walk to the Markov chain models. We

³The original dataset is available at http://chriswhong.com/open-data/foil_nyc_taxi/. The processed dataset used for the experiments in this paper is available with our codes.

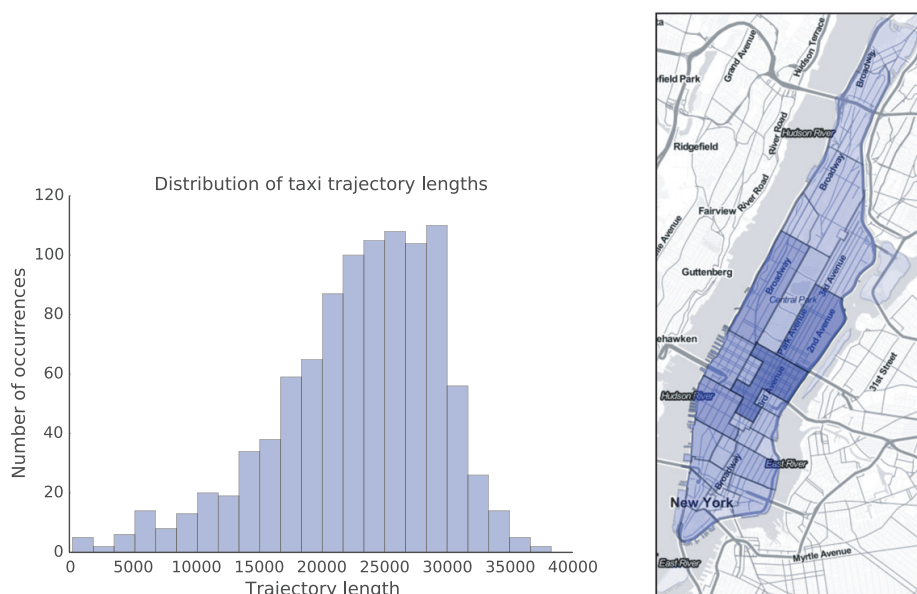


Fig. 4 Summary statistics of taxi dataset. Left: Distribution of trajectory lengths. Right: Empirical geographic frequency at the resolution of Manhattan neighborhoods (log scale). Darker colors are more frequently visited by taxis.

Table 3 RMSEs on the test set for the taxi trajectory data. The spacey random walk and second-order Markov chain have the same performance and are better predictors than a first-order Markov chain. The zeroth-order Markov chain corresponds to predicting based on the frequency spent at each state in the training data.

Zeroth-order Markov chain	First-order Markov chain	Second-order Markov chain	Learned spacey random walk
0.922	0.846	0.835	0.835

used 800 of the taxi trajectories for model training and 200 trajectories for model testing. Table 3 lists the RMSEs on the test set for each of the models. We observe that the spacey random walk and the second-order Markov chain have the same predictive power, and the first-order Markov chain has worse performance. If the spacey random walk converges, we know from the results in section 5 that the model is asymptotically (first-order) Markovian. Thus, for this dataset, we can roughly evaluate the spacey random walk as obtaining the same performance as a second-order model with asymptotically first-order information. We note that, at first glance, the performance gains over the first-order Markov chain appear only modest (0.011 improvement in RMSE). However, the results summarized in Table 2 show that it is difficult to gain substantial RMSE improvements on a four-state process that is actually generated from the spacey random walk process. With the taxi data, we have even more states (37) and the difference between the zeroth-order Markov chain baseline and the first-order Markov chain RMSEs is only 0.076. We conclude that the spacey random walk model is not a perfect model for this dataset, but it shows definite improvement over a first-order Markov chain model.

7. Discussion. This article analyzes a stochastic process related to eigenvector computations on the class of hypermatrices representing higher-order Markov chains. The process, which we call the spacey random walk, provides a natural way to think about tensor eigenvector problems which have traditionally been studied as just sets of polynomial equations. As is common with hypermatrix generalizations of matrix problems, the spacey random walk is more difficult to analyze than the standard random walk. In particular, the spacey random walk is not Markovian—instead, it is a specific type of generalized vertex-reinforced random walk. However, the intricacies of the spacey random walk make its analysis an exciting challenge for the applied mathematics community. In this paper alone, we have relied on tools from dynamical systems, numerical linear algebra, optimization, and stochastic processes.

Following the work of Benaïm (1997), we connected the limiting distribution of the discrete spacey random walk to the limiting behavior of a continuous dynamical system. Through this framework, we fully characterized the dynamics of the two-state spacey random walk and reached the positive result that it always converges to a stable equilibrium point. Analysis of the general case is certainly more complicated and provides an interesting challenge for future research.

One major open issue is understanding the *numerical* convergence properties of the spacey random walk, including algorithms for computing the limiting distribution (if it exists). For standard random walks, a simple power method is sufficient for this analysis, and generalizations of the power method have been studied for computing the stationary distribution in (3) (Chu and Wu, 2014; Gleich, Lim, and Yu, 2015; Li and Ng, 2014). However, we have shown that convergence of the dynamical system does *not* imply convergence of the power method for computing the stationary distribution of the spacey random walk. Nevertheless, the power method still often converges even when convergence is not guaranteed by the theory (Gleich, Lim, and Yu, 2015). We suspect that convergence of the power method is a sufficient condition for the convergence of the dynamical system.

A final outstanding issue is determining *when* the occupation vector of the spacey random walk will converge. We suspect that this problem is undecidable as similar results have been shown for general dynamical systems (Buescu, Graça, and Zhong, 2011). One approach for this problem is to show that the spacey random walker can simulate a Turing machine, at which point determining convergence is equivalent to the halting problem (see the work of Moore (1990) for a similar approach).

Acknowledgments. We thank Tao Wu for observing subtleties in our use of the term stable point in the analysis of the two-state case when $f(x) \equiv 0$ and our convergence analysis of the dynamical system in the limit of Euler’s method when the step size h approaches 0 in the limit.

REFERENCES

- E. S. ALLMAN AND J. A. RHODES (2003), *Phylogenetic invariants for the general Markov model of sequence mutation*, Math. Biosci., 186 (2), pp. 113–144. (Cited on p. 325)
- U. ALON (2007), *Network motifs: Theory and experimental approaches*, Nature Rev. Genetics, 8 (6), pp. 450–461. (Cited on p. 331)
- A. ANANDKUMAR, R. GE, D. HSU, S. M. KAKADE, AND M. TELGARSKY (2014), *Tensor decompositions for learning latent variable models*, J. Mach. Learn. Res., 15 (1), pp. 2773–2832. (Cited on p. 325)
- S. ASMUSSEN AND P. W. GLYNN (2007), *Stochastic Simulation: Algorithms and Analysis*, Springer, New York. (Cited on p. 322)
- D. BAYER AND P. DIACONIS (1992), *Trailing the dovetail shuffle to its lair*, Ann. Appl. Probab., 2 (2), pp. 294–313, <https://doi.org/10.1214/aoap/1177005705>. (Cited on p. 322)

- M. BENAÏM (1997), *Vertex-reinforced random walks and a conjecture of Pemantle*, *Ann. Probab.*, 25 (1), pp. 361–392. (Cited on pp. 325, 326, 327, 329, 336, 343)
- M. BENAÏM AND M. W. HIRSCH (1996), *Asymptotic pseudotrajectories and chain recurrent flows, with applications*, *J. Dynam. Differential Equations*, 8 (1), pp. 141–176. (Cited on p. 336)
- A. R. BENSON, D. F. GLEICH, AND J. LESKOVEC (2015), *Tensor spectral clustering for partitioning higher-order network structures*, in *Proceedings of the 2015 SIAM International Conference on Data Mining*, SIAM, Philadelphia, 2015, pp. 118–126, <https://doi.org/10.1137/1.9781611974010.14>. (Cited on p. 331)
- A. BERMAN AND R. J. PLEMMONS (1994), *Nonnegative Matrices in the Mathematical Sciences*, SIAM, Philadelphia. (Cited on p. 337)
- J. BUESCU, D. GRAÇA, AND N. ZHONG (2011), *Computability and dynamical systems*, in *Dynamics, Games and Science I*, Springer, New York, pp. 169–181. (Cited on p. 343)
- F. CHERICHETTI, R. KUMAR, P. RAGHAVAN, AND T. SARLOS (2012), *Are web users really Markovian?*, in *Proceedings of the 21st International Conference on World Wide Web*, ACM, New York, pp. 609–618. (Cited on p. 323)
- M. T. CHU AND S.-J. WU (2014), *On the Second Dominant Eigenvalue Affecting the Power Method for Transition Probability Tensors*, Technical report. (Cited on pp. 336, 343)
- P. COMON (1994), *Independent component analysis, a new concept?*, *Signal Process.*, 36 (3), pp. 287–314, [https://doi.org/10.1016/0165-1684\(94\)90029-9](https://doi.org/10.1016/0165-1684(94)90029-9). (Cited on p. 325)
- D. COPPERSMITH AND P. DIACONIS (1987), *Random Walks with Reinforcement*, unpublished manuscript, pp. 187–220. (Cited on p. 325)
- P. DIACONIS (1988), *Recent progress on de Finetti’s notions of exchangeability*, in *Bayesian Statistics 3*, J. M. Bernardo, M. H. DeGroot, D. V. Lindley, and A. F. M. Smith, eds., Oxford University Press, New York, pp. 111–125. (Cited on p. 326)
- J. DUCHI, S. SHALEV-SHWARTZ, Y. SINGER, AND T. CHANDRA (2008), *Efficient projections onto the ℓ_1 -ball for learning in high dimensions*, in *Proceedings of the 25th International Conference on Machine Learning*, pp. 272–279. (Cited on p. 339)
- S. FORTUNATO (2010), *Community detection in graphs*, *Phys. Rep.*, 486 (3), pp. 75–174. (Cited on p. 330)
- D. F. GLEICH, L.-H. LIM, AND Y. YU (2015), *Multilinear PageRank*, *SIAM J. Matrix Anal. Appl.*, 36 (4), pp. 1507–1541, <https://doi.org/10.1137/140985160>. (Cited on pp. 324, 329, 330, 336, 337, 338, 340, 343)
- G. H. GOLUB AND C. F. VAN LOAN (2012), *Matrix Computations*, JHU Press, Baltimore, MD. (Cited on p. 326)
- R. A. HARSHMAN (1970), *Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multimodal factor analysis*, *UCLA Working Papers in Phonetics*, 16, pp. 1–84. (Cited on p. 325)
- D. L. HARTL AND A. G. CLARK (1997), *Principles of Population Genetics*, Sinauer Associates, Sunderland, MA. (Cited on p. 330)
- G. F. JONSSON AND L. N. TREFETHEN (1998), *A numerical analyst looks at the “cutoff phenomenon” in card shuffling and other Markov chains*, in *Numerical Analysis 1997: Proceedings of the 17th Dundee Biennial Conference*, Addison Wesley Longman, Reading, MA, pp. 150–178. (Cited on p. 322)
- J. G. KEMENY AND J. L. SNELL (1960), *Finite Markov Chains*, van Nostrand, Princeton, NJ. (Cited on p. 323)
- F. KRZAKALA, C. MOORE, E. MOSSEL, J. NEEMAN, A. SLY, L. ZDEBOROVÁ, AND P. ZHANG (2013), *Spectral redemption in clustering sparse networks*, *Proc. Natl. Acad. Sci. USA*, 110 (52), pp. 20935–20940. (Cited on p. 323)
- A. N. LANGVILLE AND C. D. MEYER (2006), *Google’s PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, Princeton, NJ. (Cited on p. 322)
- W. LI AND M. K. NG (2014), *On the limiting probability distribution of a transition probability tensor*, *Linear and Multilinear Algebra*, 62 (3), pp. 362–385. (Cited on pp. 324, 335, 336, 343)
- L.-H. LIM (2005), *Singular values and eigenvalues of tensors: A variational approach*, in *CAM-SAP2005: Proceedings of the 1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pp. 129–132. (Cited on p. 324)
- L.-H. LIM (2013), *Tensors and Hypermatrices*, in *Handbook of Linear Algebra*, CRC Press, Boca Raton, FL, pp. 231–260. (Cited on p. 325)
- C. D. MANNING, P. RAGHAVAN, AND H. SCHÜTZE (2008), *Introduction to Information Retrieval*, Cambridge University Press, Cambridge, UK, 2008. (Cited on p. 330)
- Q. MEI, J. GUO, AND D. RADEV (2010), *DivRank: The interplay of prestige and diversity in information networks*, in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, pp. 1009–1018. (Cited on p. 330)

- C. MOORE (1990), *Unpredictability and undecidability in dynamical systems*, Phys. Rev. Lett., 64 (20), pp. 2354–2357. (Cited on p. 343)
- L. PAGE, S. BRIN, R. MOTWANI, AND T. WINOGRAD (1999), *The PageRank Citation Ranking: Bringing Order to the Web*, Tech. report 1999-66, Stanford InfoLab, Stanford, CA. (Cited on pp. 329, 330)
- R. PEMANTLE (1988), *Random Processes with Reinforcement*, Ph.D. thesis, MIT, Cambridge, MA. (Cited on p. 325)
- R. PEMANTLE (1992), *Vertex-reinforced random walk*, Probab. Theory Related Fields, 92 (1), pp. 117–136. (Cited on p. 328)
- R. PEMANTLE (2007), *A survey of random processes with reinforcement*, Probab. Surv., 4, pp. 1–79. (Cited on pp. 324, 325)
- L. QI (2005), *Eigenvalues of a real supersymmetric tensor*, J. Symbolic Comput., 40 (6), pp. 1302–1324. (Cited on p. 324)
- M. ROSVALL, A. V. ESQUIVEL, A. LANCICHINETTI, J. D. WEST, AND R. LAMBIOTTE (2014), *Memory in network flows and its effects on spreading dynamics and community detection*, Nature Commun., 5, 4630. (Cited on p. 323)
- S. E. SCHAEFFER (2007), *Graph clustering*, Comput. Sci. Rev., 1 (1), pp. 27–64. (Cited on p. 330)
- N. D. SIDIROPOULOS, G. B. GIANNAKIS, AND R. BRO (2000), *Blind PARAFAC receivers for DS-CDMA systems*, IEEE Trans. Signal Process., 48 (3), pp. 810–823. (Cited on p. 325)
- A. SMILDE, R. BRO, AND P. GELADI (2004), *Multi-Way Analysis*, Wiley, Chichester, UK. (Cited on p. 325)
- W. J. STEWART (1994), *Introduction to the Numerical Solutions of Markov Chains*, Princeton University Press, Princeton, NJ. (Cited on p. 323)
- J. SUN, D. TAO, AND C. FALOUTSOS (2006), *Beyond streams and graphs: Dynamic tensor analysis*, in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, pp. 374–383, <https://doi.org/10.1145/1150402.1150445>. (Cited on p. 325)