Designing Asymptotics and Transients of Linear Stochastic Automaton Networks

Article · August 2010
DOI: 10.2514/6.2010-8150

4 authors, including:

Sandip Roy
Washington State University
95 PUBLICATIONS 1,138 CITATIONS
SEE PROFILE

Yan Wan
University of North Texas
81 PUBLICATIONS 723 CITATIONS
SEE PROFILE

Ali Saberi
Washington State University
272 PUBLICATIONS 5,115 CITATIONS
SEE PROFILE

All content following this page was uploaded by Ali Saberi on 27 December 2016.

The user has requested enhancement of the downloaded file. All in-text references underlined in blue are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.
Designing Asymptotics and Transients of Linear Stochastic Automaton Networks

Mengran Xue*    Sandip Roy*    Yan Wan†    Ali Saberi*

July 14, 2010

Abstract

Stochastic automaton networks are widely used in both modeling and computation. Motivated by several of these applications (e.g., weather modeling), we study the problem of designing stochastic automaton networks to shape their transient and/or asymptotic responses. Specifically, for a broad but specially tractable class of stochastic automata networks known as influence models, we consider designing parameters of the interactions among the network components (as specified by a known graph), so that statistics of the network’s state at particular time-snapshots meet requirements. In this paper, we address such snapshot design for the asymptotic case in some generality, while focusing on an illustrative example (originating from the weather-modeling application) in designing transients.

1 Introduction

Stochastic automaton networks (SANs)—i.e., dynamical networks whose components’ discrete-valued statuses evolve through stochastic interactions—have found wide application in both modeling and computation (e.g., [1–7]). In particular, SANs—whether defined on a lattice or a more-general graph—have been used to abstractly represent diverse systems including voting and decision-making processes, spatial population dynamics, evolutionary processes, and communication/queueing dynamics. At the same time, SANs have also proved valuable in solving computational problems in networks, e.g. for achieving distributed consensus, partitioning graphs, generating random data, and solving other discrete-search and decision-making problems (e.g., [8–11]). With both types of applications in mind, here we introduce and pursue the problem of designing interactions in SANs so that their transients and/or asymptotics have desirable statistical properties. Specifically, for a broad class of linear SANs, we demonstrate design of network interactions so that a snapshot of the automaton has desirable statistics, either at particular transient times or asymptotically. In this introduction, let us 1) broadly

*School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA. Correspondence should be sent to mxue@eecs.wsu.edu. The authors are supported by NASA Grant NNA06CN26A and NSF Grants ECS-0725589 and ECS-0901137.
†Department of Electrical Engineering, University of North Texas, Denton, TX.
motivate the statistics-design problem in both modeling and computational applications; 2) overview the class of linear network models, namely influence models, which will be studied here; and 3) list some specific outcomes of the design work.

SANs are compelling for a range of modeling tasks because they allow generation of rather intricate discrete-valued dynamics (e.g., cyclic or persistent temporal responses, coalesced and fractalized spatial patterns, etc) from simple and sparse local interactions or update rules, in analogy with numerous natural and engineered networks [1-7]. Specifically, local interaction rules for SANs can be constructed so that a large-scale instance of the model (defined on a lattice or a more general graph) exhibits properties that are also observed in the actual dynamical network being modeled. Hence the SAN with these interaction rules can be advanced as an abstract representation of the actual dynamical network. However, while these various models qualitatively capture features of the networks they represent, they often are not easily amenable to parameterization for quantitative analysis, nor do they always permit easy statistical analysis of asymptotics or transients. While a few efforts have been made to parameterize some models of this sort from observation data or expected statistics (e.g. using the expectation-maximization algorithm [12]), very little is known in general about parameterizing such models: the existing procedures lack generality, and do not maintain an imposed graph structure or even sparsity in the network’s graph. Further, in practice, parameterizing a stochastic network model from data may be complicated by limited data availability: data or probabilistic information may be available at only a few time instances or locations in the network. With these needs in mind, we here pursue the systematic parameterization or design of a broad class of stochastic network models so that their snapshots meet statistical requirements. Once such parameterization has been achieved, the model can in turn be used for quantitative analysis of other spatiotemporal characteristics, and for simulation, in the modeling application of interest.

The SAN design problem that we pursue also is of value in the computational applications of these models [8,9,11]. Many computational applications of SANs are founded on the recognition that these discrete-valued models naturally permit distributed search and decision-making among a large but discrete set of possibilities—such as arise in graph-computation and integer-programming problems. In these computational applications, the SAN’s parameters are usually chosen to reflect the cost metric of the discrete-search or decision-making problem of interest. In these computational domains, designing the statistics of the SAN can permit faster solution of the discrete-search or decision-making problem, and facilitate analysis of the computational algorithm.

In general, the analysis of SANs—and especially their transient dynamics—is quite computationally intensive, often scaling exponentially in the number of vertices. The influence model is a special SAN that can abstractly represent a wide class of stochastic dynamics on networks [7,9,11,12], and yet is specially tractable in the following sense. Statistics (probabilities/joint probabilities) of small groups of components’ statuses can be found over time using low-order linear recursions, with the size of the recursion growing gracefully with the order of the statistic to be found [7,13,14]. This special tractability, which originates from a moment-linearity property of the model, permits the analysis of relevant individual-component statistics and correlations that we desire. Given this tractability, and the wide freedom in capturing stochastic dynamics afforded by the model, we adopt the influence model here as a framework for addressing the statistics-
design problem. Specifically, we pursue design of the interaction parameters of the influence model—which capture both the strengths of the directed influences (or, equivalently, interaction likelihoods) and the status-transitions effected by such influences—to set component statistics (i.e., components’ status probabilities) asymptotically or during the transient.

The following specific results will be presented in this article:

- The problem of designing/parametrizing an influence model to meet statistical specifications (which we call the snapshot design problem) will be formulated in generality, after a brief review of the influence model and its tractabilities (Section 2).
- The design problem will be addressed in the case where asymptotic requirements are given (Section 3).
- Design will be pursued in the case where transient requirements are in force (Section 4), with a particular focus on a specific model for weather propagation.

In this paper, we will illustrate the designs for broad sub-classes of the general problem, with the expectation that the problem will be addressed in full generality in future work.

2 Review and Problem Formulation

The influence model is a Markovian SAN, which can flexibly represent a variety of stochastic evolutions on networks and yet is amenable to low-order linear analysis of statistics (specifically, component status probabilities and joints) [7, 13, 14]. We here review the model’s dynamics (Section 2.1), and briefly summarize its tractabilities (Section 2.2). We then motivate and pose the problem of designing the influence model so that snapshots (during transients or in steady-state) meet first-order statistical requirements (Section 2.3).

2.1 The Influence Model and its Dynamics

The influence model [7, 13] comprises a network of n components or nodes or sites, which we label 1, . . . , n. WLOG, each site has a binary status that evolves in discrete time. WLOG, we assume that each site in the network has only two status. We find it convenient to represent the status of each site i at a time k using a two-element 0−1 indicator vector, which we call the local status vector or simply status vector for site i and denote as $s_i[k]$. The sites’ statuses evolve in time due to probabilistic influences from neighboring sites (possibly including the site itself), as specified by a directed network graph $\Gamma^1$. Specifically, a site j’s next- (or time $k + 1$) status is influenced by a site i’s current (time-$k$) status (where i may equal j), if there is an edge from i to j in $\Gamma$. This influence is codified with two parameters: a scalar interaction strength or weight $d_{ji}$ (where $0 \leq d_{ji} \leq 1$, and $\sum_i d_{ji} = 1$) that indicates the frequency with which i influences j, and a $2 \times 2$ row-stochastic local transition matrix $A_{ij}$ that specifies the probabilistic

---

1We permit self-loops, i.e., edges directed from vertices back to themselves, in the network graph $\Gamma$. 

rule by which site j’s next status is determined by site i’s current status. Precisely, the sites’ next-statuses are determined from their current statuses as follows:

- For each site j, a (neighboring) site i is chosen as its influencing or determining site independently with probability $d_{ji}$, where $0 < d_{ji} \leq 1$ and $\sum_i d_{ji} = 1$ (and $d_{ji}$ is defined to be nil if there is no edge from i to j in $\Gamma$).

- Each site’s next-status is generated independently, according to a probability vector (or probability mass function) specified by its determining site’s current status. Specifically, the next-status of site j is chosen according to the probability mass function (pmf) specified in the vector $s_i'[k]A_{ij}$. That is, if $s_i[k] = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, the entries in the first row of $A_{ij}$ specify the probabilities that $s_j[k+1]$ equals $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$, respectively; similarly, the second row specifies next-status probabilities when the determining site has status vector $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

We have thus specified the time-evolution of the influence model. We will be concerned with analyzing and designing this evolution. For convenience in analysis, we find it useful to assemble certain influence-model parameters into matrices/vectors. In particular, we define a network matrix $D = [d_{ij}]$, and also a full status vector $s[k] = \begin{bmatrix} s_1[k] \\ \vdots \\ s_n[k] \end{bmatrix}$. We note that $D$ is a row-stochastic matrix.

The influence model is a promising tool for abstractly representing various stochastic network dynamics, because it can capture heterogeneous stochastic influences in a network with general graph structure [13]. Specifically, although the model is very specially structured in certain senses (e.g., in each site’s selection of a determining site at each time step), the model also permits wide latitude in capturing stochastic influences in networks—including in allowing arbitrary graphical structures for influence, and in permitting generic and heterogeneous local-influence rules (e.g., copying or anti-copying influences as well as more general stochastic influences). Because of the model’s ability to capture varying influence structures, it encompasses a diversity of stochastic network dynamics, including both ergodic and non-ergodic dynamics; settling, periodic, and apparently chaotic responses; and long-range spatial correlations and persistences in the asymptotics. This wide representation capability makes the influence model suitable for numerous applications, including in modeling voting and decision-making processes, in abstractly modeling failures in infrastructure networks, in capturing weather evolution, and in graph partitioning and distributed-agreement computations [7, 9, 11–14]. Because the influence model has a wide capability to represent diverse network dynamics, we believe that it provides a natural forum for studying design of SAN dynamics.

2.2 Tractabilities of the Influence Model

The special update structure of the influence model facilitates both transient and steady-state statistical analysis. The key tractability afforded by the model is as follows [7, 13]:

4
status probabilities for individual sites (pmfs for each site’s status at particular times) and joint-status probabilities for small groups of sites can be found by solving low-order discrete-time linear systems models. In particular, computation of these probabilities and joint probabilities does not require tracking status configuration probabilities for the entire network (which would have a storage/computation cost that is exponential in the number of sites), and instead the computation order grows gracefully in the number of sites whose joint status probabilities are required. In turn, these probabilistic analyses of status and joint-status probabilities yield qualitative insights into the transient and asymptotic characteristics of the network. The key tractability of the influence model can be extended in various ways, including toward characterizing statuses across multiple time steps and giving graph-theoretic measures of settling and steady-state performance.

The statistical analysis of the influence model is developed in detail in [7, 13]. Here, let us summarize the analysis of the individual sites’ status probabilities, since this analysis is needed for the design problem that we pursue. Specifically, let us consider the (unconditioned) pmf vector for site i’s status at time k, which we denote \( p_i[k] \). With a simple conditional-probability analysis, it can be shown that the full status probability vector \( p[k] \) containing the individual sites’ pmfs (\( p[k] = \begin{bmatrix} p_1[k] \\ \vdots \\ p_n[k] \end{bmatrix} \)) evolves according to the following linear difference equation:

\[
p'[k + 1] = p'[k]H,
\]

where

\[
H = \begin{bmatrix}
d_{11}A_{11} & d_{21}A_{12} & \cdots & d_{n1}A_{1n} \\
d_{12}A_{21} & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & d_{n1}A_{n1} \\
d_{nn}A_{nn} & \cdots & \cdots & d_{nn}A_{nn}
\end{bmatrix}.
\]

This expression shows that individual sites’ status probabilities can be found using a low-order linear recursion (linear system), of dimension \( 2n \). Thus, local status probabilities can be found with quite-reasonable computational effort.

Similar recursions can be developed for joint status probabilities of groups of sites, with the dimension of the recursion on the order of \( n^\alpha \) where \( \alpha \) is the number of sites whose joint is being found; we refer the reader to [13] for details. These higher-order recursions permit efficient analysis of correlations among site statuses, and hence give insight into the pattern of statuses in the network. We stress that the efficient analysis of these partial statistics is a special property of the influence model: other SAN models such as dynamic Bayesian network models generally do not permit such analysis. This tractability is beneficial to our development, in that it permits design of the model to meet specific statistical criteria and in that the designed model can be characterized. As a further extension, joint probabilities for status configurations across multiple time-steps can also be obtained using low-order linear recursions [14].

Also of significance to our development, influence model dynamics have been characterized and classified in terms of the model’s underlying graph structure (i.e., the graph \( \Gamma \), as well as other graphical representations of the \( A_{ij} \) and \( H \)), in part by relating fea-
tures of these graphs to the spectra of status-probability recursion matrices (e.g., $H$). We kindly ask the reader to see [13, 14] for details.

2.3 The Influence-Model Snapshot Design Problem

Our group has been studying use of the influence model in several application domains, including in weather- and infrastructure-network-modeling [7, 15]; and in distributed-agreement, partitioning, and data generation. In several of these applications, we are encountering problems of the following form: the interaction topology of the influence-model representation (as specified by the network graph $\Gamma$) is clear and the model appears to capture qualitative features of the dynamics of interest, but parameterization of the model (i.e., proper selection of the interaction weights and transitions) is needed to permit quantitative prediction using the model. While the information available for selection of the weights and transitions varies among the applications, the following paradigm seems especially common: some rudimentary information on local statistics is available at a few snapshots in time. Influence model parameters that match these probabilistic snapshots must be obtained, whereupon the model can be used for simulation and to compute first- and higher-order statistics at any time step. A problem of this sort arises, for instance, in spatial weather modeling over a one- to multi-day time horizon, where aggregations of global forecast models can be used to obtain rough probabilities of weather conditions/impacts in a region at a few time instances. If a lattice-structure influence model that meets these probability maps can be developed, it can subsequently be used to simulate weather (or its impact), interpolate the probability map, and find correlations in weather among multiple locations.

Let us give a general, mathematical description of the influence model design problem described above. We assume that the stochastic dynamics of interest can be represented using an influence model with known graph $\Gamma$. In general, interaction parameters associated with a subset of the edges in the graph $\Gamma$ are unknown, and must be designed/inferred. Specifically, for a particular subgraph $\Gamma_1$ of $\Gamma$, the weight $d_{ji}$ and/or the transition matrix $A_{ij}$ associated with each edge $\{i, j\}$ of $\Gamma_1$ must be designed, subject to the constraints that $d_{ji} \geq 0$, $\sum_i d_{ji} = 1$, and that the $A_{ij}$ are stochastic matrices. We aim to design these parameters so that the first-order statistics at particular time snapshots meet desired values. Precisely, length-2n desired first-order probability vectors $\bar{p}_{[k_1]}, \ldots, \bar{p}_{[k_s]}$ are specified for s time instances $k_1, \ldots, k_s$. Our goal is to design the influence model parameters so that the first-order probability vector $p_{[k]}$ meets these desired statistics at the proper times, i.e. so that $p_{[k_1]} = \bar{p}_{[k_1]}, \ldots, p_{[k_s]} = \bar{p}_{[k_s]}$. We refer to this problem as the snapshot design problem for the influence model. We are also interested in the case where asymptotics of the snapshot probabilities need to be designed. For this case, we consider only two time instances (i.e., times $k_1$, $k_2$ with $s = 2$). That is, we pursue design of the influence model so that $p_{[k_1]} = \bar{p}_{[k_1]}$ and $\lim_{k_2 \to \infty} p_{[k_2]} = \bar{p}$, where $\bar{p}$ is again a desired first-order probability vector. We refer to this problem as the asymptotic snapshot design problem. We note that the snapshot-design problems can only possibly have a solution if the entries in the desired probability vectors corresponding to each influence model site are valid pmfs. We assume this to be true throughout.

In this paper, we solve the snapshot probability design problem for some broad classes
of network topologies (e.g., lattices, more general directed graphs), and for some applicable sets of designable parameters (e.g., all the transition matrices in the network). For these cases, we pursue two analyses, namely 1) evaluation of whether or not a set of desired probabilities can be achieved, and 2) development of procedures to find parameter values that meet the desired probabilities.

3 Asymptotic Snapshot Design

Let us first study the asymptotic snapshot design problem. In this section, we address the case that the interaction weights $d_{ji}$ on edges in $\Gamma$ are already given, while the $A_{ij}$ are free to be designed. This case is relevant in many application areas since interaction strengths (as well as the graph structure) are often known or can be obtained/estimated prior to the design. This example also serves as a stepping stone toward the general case, where some $d_{ji}$ and $A_{ij}$ are designable, while others are known.

In particular, let us assume (WLOG) that the first time instance involved here is $k_1 = 0$. We note that the desired $\bar{p}[0]$ is often simply a specified initial full status vector $s[0]$ (referred as the initial condition as well), but statistical models for the desired snapshots are also allowed. In the example problem, we note that the $n$-site influence model has a given network matrix $D$. Since an asymptotic snapshot design is being considered, the desired asymptotic individual status probability vector for each site $i$ is also specified; we use the notation $\bar{p}_i = \begin{bmatrix} \bar{p}_{i,1} \\ \bar{p}_{i,2} \end{bmatrix}$ for it. Our goal here is to design the local state transition matrices $A_{ij}$ so that the given desired asymptotic individual status probability vectors are achieved, i.e., $\lim_{k_2 \to \infty} p_i[k_2] = \bar{p}_i$ for all $i$.

The procedure that we use for solving the asymptotic snapshot design problem draws heavily on the graph-theoretic- and spectral- analysis of the influence model. In the interest of space, we exclude the details of how this analysis is applied in solving the design problem in this paper. We kindly ask the reader to see [7, 13], which give a detail analysis of the influence model. Briefly, this analysis shows that the first-order probability vector $p[k]$ approaches a steady-state that has no dependence on the initial condition, under broad conditions on the network matrix $D$ and the $A_{ij}$. The design problem thus can be re-formulated as that of selecting $A_{ij}$ so that 1) these broad conditions for existence of a steady-state are satisfied and 2) the desired snapshot probabilities are a fixed point for the first-order probability recursion.

Given any acceptable asymptotic snapshot probabilities $\bar{p}$ that are not nil or unity, it turns out that feasible solutions exist to the asymptotic snapshot design problem, as we show in the following Theorem 1. The associated proof gives a procedure/algorithm for designing $A_{ij}$ that achieve the desired asymptotic snapshot probabilities, and in fact gives a family of viable designs.

**Theorem 1** Consider an influence model with a given graph $\Gamma$ and network matrix $D$. For any acceptable desired asymptotic status probability vectors $\bar{p}_i$ for each site $i$ such that $0 < \bar{p}_{i,1} < 1$ and $0 < \bar{p}_{i,2} < 1$, the asymptotic snapshot design problem can be solved. That is, there always exists row-stochastic state transition matrices $A_{ij}$ such that the asymptotic probabilities are achieved.
Proof 1 Here, we will draw on the spectral and graph-theoretic analysis of the influence model to find a feasible design for the $A_{ij}$’s. Specifically, given an acceptable desired asymptotic snapshot $\bar{p} = \begin{bmatrix} \bar{p}_1 \\ \vdots \\ \bar{p}_n \end{bmatrix}$ (as specified in the problem statement), we will show that a set of feasible $A_{ij}$’s (i.e., ones satisfying the condition that $A_{ij}$’s are row-stochastic matrices) can always be found.

The spectral and graph-theoretic analysis of the influence model given in [13] yields the following: the vector $\bar{p}$ is the unique steady-state pmf vector for an influence model, if 1) $\bar{p}^tH = \bar{p}'$, and 2) a construct known as an autonomously-recurrent class of the matrix $H$ has a strictly dominant eigenvalue at 1 (see [13] for details). Building on the analysis given in Chapter 4 of [13], it can be shown that the second condition is automatically satisfied as long as the $A_{ij}$ have strictly non-zero and non-unity entries (in addition to being stochastic). Thus, we will aim to find $A_{ij}$ of this form, such that $\bar{p}^tH = \bar{p}'$.

To continue, in vector notation, we have

$$\begin{bmatrix} \bar{p}_1' \\ \vdots \\ \bar{p}_n' \end{bmatrix} H = \begin{bmatrix} \bar{p}_1' \\ \vdots \\ \bar{p}_n' \end{bmatrix}. \quad (3)$$

Then, for each site $j$, we have

$$\sum_{i=1}^{n} \bar{p}_i'(d_{ji}A_{ij}) = \bar{p}_j'. \quad (4)$$

To simplify the development, let us also specify that $A_{1j} = A_{2j} = \cdots = A_{nj} = A_j = \begin{bmatrix} a_j & 1 - a_j \\ b_j & 1 - b_j \end{bmatrix}$. Then, Eq. 4 becomes $(\sum_{i=1}^{n} d_{ji}\bar{p}_i')A_j = \bar{p}_j' = [\bar{p}_{j,1} \bar{p}_{j,2}]$. Given the requirement on $\bar{p}_j$, these two equations are linearly dependent, and we need consider only one of them:

$$\sum_{i=1}^{n} d_{ji}\bar{p}_{i,1}a_j + (1 - \sum_{i=1}^{n} d_{ji}\bar{p}_{i,1})b_j = \bar{p}_{j,1}. \quad (5)$$

From the above equation, we note that the solution space to the pair of unknowns, $(a_j, b_j)$, is a line on the $a_j - b_j$ plane.

Next, for feasibility, we need to show that this line has at least one intersection with the range $(0, 1) \times (0, 1)$. Many feasible solutions exist; one is $a_j = b_j = \bar{p}_{j,1}$. Thus, we can find at least one feasible solution for the $A_{ij}$.

Remark: In fact, the proof of Theorem 1 yields a family of feasible solutions to this example problem of asymptotic snapshot design since many candidates of $A_{ij}$ can satisfy Eq. 5. In many cases, we need to choose a proper solution from the feasible solution family so as to meet other criteria. For instance, it is easy to check that making $a_j$ large and $b_j$ small will yield models with larger spatial correlation in statuses as well as longer temporal persistence.

4 Transient Snapshot Design

As mentioned in the introduction, for some applications (e.g., weather modeling), the transient behaviors/snapshots of a stochastic network dynamics are often of great interest [15]. For a general influence model, a site’s transient status probabilities will have
a dependence on the initial condition. Therefore, designing/parameterizing an influence model to achieve some given desired transient behaviors is more complex than the asymptotic case. We do not attempt to address the more complex (transient) snapshot design problem fully in this paper. However, for some applications, the transient influence-model design problem is much more constrained, i.e., some of the parameters are fixed and not free to be designed, or the network graph $\Gamma$ has a special structure (e.g., lattice, acyclic graph). With this in mind, in this section, we will study an example snapshot design problem that is motivated by real applications, but also give some indication of how the problem can be addressed more generally. Again, we expect to give a much more comprehensive treatment in future work. Since we are illustrating the snapshot design with a practical example, we find it useful to formulate the example from a particular application, namely weather modeling. Let us begin with the formulation, and then present some design results.

4.1 Example: A Weather Influence Model and Snapshot Design Problem

Let us begin with by introducing a special weather influence model and weather snapshot design problem for which we develop a series of results. This model has been carefully motivated and introduced as a tool for air traffic management design in [15]. Here, we focus on the mathematical formulation. In this specific influence model, for each site $i$ (which represents a geographic region in the airspace), we use an indicator vector $[1, 0]'$ to represent the status that site $i$ is subject to weather impact, and $[0, 1]'$ to represent that there is no weather impact at site $i$. We view the weather (impact) as stochastically propagating or spreading among the geographic regions of interest. Thus, as a first simple example, we model that each site can only be influenced by itself and its geographically “previous” or upstream sites (since weather can be viewed as moving in certain directions, as governed by wind directions). When a site is influenced, we model it as copying the weather of the determining site, i.e., all the state transition matrices are the 2 by 2 identity matrices $I_2$. For example, as shown in Fig. 1, the geographic map of some districts of interest can be abstracted into a lattice graph where each vertex represents one region; the directed arrow between pairs of vertices (regions) represents the possible motion of the weather between the corresponding regions. Initially, we may know that there is some weather at the site in the top left corner. The weather has two directions of motion: east and south. Therefore, in this weather influence model, each site can be directly influenced by the site above it, the site left to it, and itself as well. More generally, since the weather motion is often directional within a certain period of time (there is no reverse or cyclic movement), we assume that the underlying network graph $\Gamma$ is directional as well (i.e., there is no cycle in $\Gamma$). In formulating the influence model for weather, we note one complexity: weather from upstream of the modeled region can influence the region of interest, and these influences cannot be directly modeled in the framework considered. Since we are concerned with weather only in the modeled regions, we represent these external influences in aggregate using a single self-influencing site.

\footnote{In [15], we allow for more complex local transition matrices transition matrices, that for instance also capture formation and dissipation of convective cells; the simpler model is sufficient to illustrate the design strategy, so we limit ourselves to this case here.}
Figure 1: A directed lattice structure.

whose status remains at $[0, 1]'$. Let us formulate a snapshot design problem of interest for the weather influence model. To do so, we note that we can always divide the vertices into, say, $L + 1$ ordered levels and label them by the level order $(\text{Level}_0, \ldots, \text{Level}_L)$, in such a way that weather can only move from lower levels to higher levels. For example, Fig. 2 shows the ordered levels of the directed lattice graph as shown in Fig. 1.

Now we are ready to pose the snapshot design problem for the weather model. Broadly, our goal is to parameterize the model so that it meets probabilistic forecasts at particular times [15]. Specifically, our goal is to design the weights in the graph $\Gamma$, to achieve desired first-order probability vectors at two times $k_1 = 0$ and $k_2 = k$. In this first example, we will assume that only the vertices in $\text{Level}_0$ of the graph may have some weather at time 0 and this weather is known. This assumption is sensible because we want to track the uncertain propagation of the weather from its current position (corresponding to vertices at the lowest level in the graph), along the direction of the network graph. Meanwhile, we will assume that an arbitrary probabilistic forecast must be matched at time $k_2 = k$. We envision this forecast, containing desired status probabilities, would be obtained from global weather models [16].

In sum, we see that the design problem is to construct the interaction weights for a weather influence model for which all the local state transition matrices are the identity matrix $I_2$, and the underlying network graph $\Gamma$ is directional and can be divided into $L + 1$ levels. Also at two time instances $k_1 = 0$ and $k_2 = k$, the acceptable desired first-order probability vectors $\bar{p}[k_1]$ and $\bar{p}[k_2]$ are given as a known $s[0]$ (which indicates weather only in $\text{Level}_0$) and an arbitrary pmf $\bar{p}[k]$. Our goal now becomes to design a proper
network matrix $D$ so that the desired behaviors are achieved.

4.2 Results: Weather Snapshot Design Problem

Let us present several results on the weather snapshot design problem, including a general iterative solution for the transition probabilities, and a more explicit solution for the line-graph case that helps to characterize what probabilistic forecasts are achievable. First, the following theorem presents an iterative method to solve the problem.

**Theorem 2** Consider the weather snapshot design problem. Assume $s_i[0] = [1, 0]'$ for $i \in \text{Level}_0$ and $s_i[0] = [0, 1]'$ otherwise. Also, assume that an acceptable desired full status probability vector $\bar{p}[k]$ for time $k$ ($k \geq 1$) has been given. An iterative method can be developed to find the weights $d_{ij}$ associated with $\Gamma$ so that $p[k] = \bar{p}[k]$, i.e., to solve the design problem.

**Proof 2** First, at any time $k > 0$, from the first-order recursion for the influence model, the probability of site $i \in \text{Level}_0$ still having weather (i.e., having status $s_i[k] = [1, 0]'$) equals $d_{ii}^k$. Thus, we must choose $d_{ii} = (\bar{p}_{i,1}[k])^{1/k}$ to achieve the desired probabilities at time $k$ for the sites in $\text{Level}_0$.

Now say that we have found the weights $d_{ji}$ for all $j \in \text{Level}_0, \ldots, \text{Level}_l(l - 1)$, and consider finding $d_{ji}$ for $j \in \text{Level}_l$. To find these weights, notice that the influence on site $j \in \text{Level}_l$ only comes from the lower levels ($\text{Level}_0, \ldots, \text{Level}_l(l - 1)$). At time
1 \leq k \leq l$, the probability of site \( j \) in Level \( l \) having weather, \( p_{j,1}[k] \), can be written as following (based on the first-order recursion):

\[
p_{j,1}[k] = \sum_{h=0}^{k-1} \left( \sum_{i \in \text{Level}\, k} d_{ji} p_{i,1}[k-1] + d_{jj} p_{j,1}[k-1] \right).
\]  

(6)

At time \( k > l \), \( p_{j,1}[k] \) simply becomes

\[
p_{j,1}[k] = \sum_{h=0}^{l-1} \left( \sum_{i \in \text{Level}\, k} d_{ji} p_{i,1}[k-1] + d_{jj} p_{j,1}[k-1] \right).
\]  

(7)

We note that since the weights \( d_{ji} \), for all \( j \in \text{Level}\, 0, \ldots, \text{Level}\, (l-1) \), have been found, the pmf vectors of the sites in these \( l \) levels can be obtained for any time, and that the initial condition is given as well. Thus, in Eq. 6/Eq. 7, the only unknowns are the weights \( d_{jj} \) and \( d_{ji} \) for site \( j \), and the status-probability \( p_{j,1}[k-1] \) at the previous time step at site \( j \). This previous status probability, however, can also be iteratively represented in terms of known probabilities, \( d_{ji} \), \( d_{jj} \), and even earlier status probabilities at site \( j \). Repeating this process, we eventually obtain an expression for \( p_{j,1}[k] \) in which only the \( d_{ji} \) and \( d_{jj} \) are unknowns (and the sum of these probabilities is 1). Arbitrarily selecting all but one free parameter and re-writing the expression in terms of the free parameter (say \( d_{jj} \)), we immediately obtain a polynomial equation that can be solved to find this remaining parameter. As long as all probabilities obtained in this way are in the interval \([0,1]\), the determining probabilities for site \( j \) have been found. Repeating this process for each site in \( \text{Level}\, \mathcal{L} \), the \( l \)th step of the iteration is complete, and the result has been proved by induction.

In Theorem 2, although the iterative method can give us a design solution, this solution is not unique. Conversely, there is no guarantee that the solution obtained is a valid one, in the sense that the weights \( d_{ji} \) are within the acceptable ranges \((0 \leq d_{ji} \leq 1)\). In order to gain a better understanding of these issues in parameterizing the weather influence model, it is useful to quantify the possible ranges of snapshot probabilities at time \( k \) over the range of model parameters. We expect to better characterize this range of possibilities in future work, by exploiting linearities in the update-rule structure. As a first step in this direction, let us pursue an explicit computation of the transient status-pmf probabilities in terms of the designable parameters in a simple example, specifically one with a line structure (Fig. 3). In Fig. 3, we assume that weather is initially in site 1, with site 0 included only so that the model is a valid influence model. As before, we assume that the weather initially starts at site 1 (i.e., \( \text{Level}\, 0 \) in the graph), and that a set of first-order probabilities \( \mathbf{p}[k] \) must be achieved by the model at time \( k \).

For such a line structure model, we present an explicit expression of the status pmf vector \( \mathbf{p}[k] \) at time snapshots in terms of the free entries of \( D \), in the following theorem.

**Theorem 3** Consider a weather influence model with a line network graph \( \Gamma \) as shown in Fig. 3. Assume that at time 0, only site 1 has weather. Also assume that all the self-influence weights \( d_{ii} \)'s are different. Then, the probability of site \( i \) \((i \neq 0)\) having weather at time \( k \) \((k \geq 1)\) is

\[
p_{i,1}[k] = \begin{cases} 
\left( \prod_{x=2}^{i} d_{x,x-1} \right) \frac{d_{i1}^{k}}{\sum_{z=1}^{i} \prod_{y=1, y \neq z}^{i} d_{zz} d_{yy}}, & \text{for } i = 1; \\
\prod_{x=1}^{i} d_{x,x-1} \frac{d_{ii}^{k}}{d_{ii}^{k} - d_{ii}}, & \text{otherwise}.
\end{cases}
\]
Proof 3: Since the transition matrices are identical for every interaction in the influence model, the first-order pmf computation can be simplified to the following:

\[ p[k] = (D^k \otimes I_2)s[0], \]

where \( D \) is the network matrix of the influence model and \( \otimes \) represents the standard Kronecker product of two matrices. Noting that the status of site 0 is \([0,1]^T\) (i.e., weather is not present), we immediately see that the probabilities of having weather at site \( i \), for \( i = 1, \ldots, n \), at time \( k \) can be written as

\[
\begin{bmatrix}
p_{1,1}[k] \\
p_{2,1}[k] \\
\vdots \\
p_{n,1}[k]
\end{bmatrix} = \hat{D}^k \begin{bmatrix}
s_{1,1}[0] \\
s_{2,1}[0] \\
\vdots \\
s_{n,1}[0]
\end{bmatrix},
\]

where \( \hat{D} = \begin{bmatrix} d_{11} & 0 & 0 & \cdots & 0 & 0 \\ d_{21} & d_{22} & 0 & \cdots & 0 & 0 \\ 0 & d_{32} & d_{33} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & d_{n,n-1} & d_{nn} \end{bmatrix} \) is a sub-stochastic matrix.

To obtain explicit expressions for \( p_{i,1}[k] \), we perform an eigenanalysis on \( \hat{D} \). Since \( \hat{D} \) is lower triangular, the \( n \) (distinct) eigenvalues of \( \hat{D} \) are \( d_{ii} \), for \( i = 1, \ldots, n \). The \( j \)th entry of the right eigenvector \( \mathbf{v}_i \) and the left eigenvector \( \mathbf{w}_i \) associated with eigenvalue \( d_{ii} \) can be shown to be the following:

\[
\mathbf{v}_{i,j} = \begin{cases} 
0, & \text{for } 1 \leq j < i; \\
1, & \text{for } j = i; \\
\prod_{z=1}^{i-j} \frac{d_{i+z,i+z-1}}{d_{ii}-d_{i+z,i+z}}, & \text{for } i < j \leq n.
\end{cases}
\]

\[
\mathbf{w}_{i,j} = \begin{cases} 
0, & \text{for } 1 \leq j < i; \\
1, & \text{for } j = i; \\
\prod_{z=1}^{i-j} \frac{d_{i+z,i+z-1}}{d_{ii}-d_{i+z,i+z-1}}, & \text{for } i < j \leq n.
\end{cases}
\]

Now consider \( d_{ij}^{(k)} \), i.e. the \( i,j \)th entry in \( \hat{D}^k \). Based on the eigen decomposition, we have

\[
d_{ij}^{(k)} = \begin{cases} 
\sum_{z=j}^{i} d_{z}^k \mathbf{v}_{z,i} \mathbf{w}_{z,j}, & \text{for } i \geq j; \\
0, & \text{otherwise}.
\end{cases}
\]
With some algebraic effort, we obtain the following expression for \( d_{ij}^{(k)} \).

\[
d_{ij}^{(k)} = \begin{cases} 
(\prod_{x=j+1}^{i} d_{x,x-1}) \sum_{z=j}^{i} \prod_{y=j,y \neq z}^{i} d_{zz}^k, & \text{for } i > j; \\
1, & \text{for } i = j; \\
0, & \text{for } i < j.
\end{cases}
\]

Finally, noting that only site 1 initially has weather \( (s_1[0] = [1, 0]' \) while \( s_i[0] = [0, 1]' \) for other \( i \)), we find that the probability of site \( i \) having weather at time \( k \) is as follows:

\[
p_{i,1}[k] = \begin{cases} 
(\prod_{x=2}^{i} d_{x,x-1}) \sum_{z=1}^{i} \prod_{y=1, y \neq z}^{i} d_{zz}^k, & \text{for } i = 1; \\
(\prod_{x=j+1}^{i} d_{x,x-1}) \sum_{z=j}^{i} \prod_{y=j,y \neq z}^{i} d_{zz}^k, & \text{for } i > j; \\
1, & \text{for } i = j; \\
0, & \text{for } i < j.
\end{cases}
\]

Thus, the theorem is proved.

A couple notes about the results are needed.

1. For convenience in presentation, we have limited ourselves to the case that the self-loop probabilities are distinct (however slightly). Since our aim here is to design the influence model, and a model having identical self-loop probabilities can be approximated arbitrarily well with one having distinct probabilities, this limitation is insignificant. In fact, the more general case can also be easily analyzed, but requires some further notation regarding the number of repetitions of these probabilities.

2. Say that we are interested in designing the status probabilities at stage \( L \) (site \( L + 1 \)) in the model at a time snapshot. Reordering the earlier stages in the line influence model does not modify the status probabilities.

3. The explicit expression for the status probabilities allows us to extremize the achievable probabilities at each time step, and hence (in tandem with the recursive approach) to characterize the desired status probabilities that can be achieved.

References


