ALARM: A LOGISTIC AUTO-REGRESSIVE MODEL FOR BINARY PROCESSES ON NETWORKS

Ameya Agaskar[1,2] *and Yue M. Lu*[1]

[1]Harvard University, Cambridge, MA 02138, USA
[2]MIT Lincoln Laboratory, Lexington, MA 02420, USA
Email: {aagaskar, yuelu}@seas.harvard.edu

ABSTRACT

We introduce the ALARM model, a logistic autoregressive model for discrete-time binary processes on networks, and describe a technique for learning the graph structure underlying the model from observations. Using only a small number of parameters, the proposed ALARM can describe a wide range of dynamic behavior on graphs, such as the contact process, voter process, and even some epidemic processes. Under ALARM, at each time step, the probability of a node having value 1 is determined by the values taken by its neighbors in the past; specifically, its probability is given by the logistic function evaluated at a linear combination of its neighbors' past values (within a fixed time window) plus a bias term. We examine the behavior of this model for 1D and 2D lattice graphs, and observe a phase transition in the steady state for 2D lattices. We then study the problem of learning a graph from ALARM observations. We show how a regularizer promoting group sparsity can be used to efficiently learn the parameters of the model from a realization, and demonstrate the resulting ability to reconstruct the underlying network from the data.

Index Terms— Dynamic processes, logistic regression, vector autoregressive (VAR) models, Networks

1. INTRODUCTION

Binary dynamic processes on graphs can be used to model systems in fields as varied as power systems engineering, political science, and ecology [1, 2]. In these systems, each node is in one of two states (which we will model as 0 and 1), and the current state of a node in the network is influenced by the previous values of its neighboring nodes (and perhaps its own previous state). Several interesting questions arise in such models: we may wish to know whether they settle into some equilibrium, whether such an equilibrium is unique, whether the nodes are likely to coalesce to a single state, or even whether a small number of state flips can cascade across the network and transform the state of most of the nodes. Furthermore, we may want to understand how well the network itself can be learned by merely observing the sequence of values produced by the model.

Over the years, several models for such dynamic processes have been developed in various fields (*e.g*, [1, 3, 4]). In this paper, we introduce a logistic autoregressive model (ALARM), a simple yet very flexible model for stochastic processes on graphs. The proposed ALARM model is a natural vector autoregressive process taking on binary values: at time t, the probability that a node has the value 1 is the logistic function $\text{logit}^{-1}(\cdot) = \frac{\exp(\cdot)}{1+\exp(\cdot)}$ applied to a linear combination of its neighbors' (and its own) values at times $t-K, \dots, t-1$.

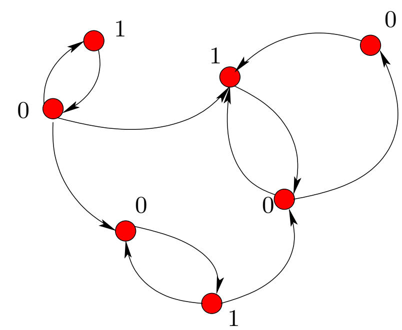

Fig. 1. A snapshot of the ALARM model at time t. The model is defined on a directed graph which captures the local interactions of the nodes. Each node has a value of 0 or 1 at time t that depends on its neighbors' values at times $t-K, \dots, t-1$.

Like many existing models, it can model the influence of neighbors on a node's value. However, it is more general than existing models: it can capture negative influences (a node favors a value opposite a neighbor's), it can model uncertainty even when a node's neighbors are unanimous, it allows control over a node's bias toward one value or another, and it can model node values that depend strongly on their history.

This could be used to model the spread of a rumor on a social network with varying levels of skepticism or distrust, the spread of an epidemic in a human interaction network, or of a virus in a computer network. Its behavior can encapsulate that of existing models, but because it is more general, we can learn what kind of model best captures the behavior in a given system.

In this paper, we describe the ALARM model in detail, along with some examples illustrating its intriguing behavior on 1D and 2D lattices. We also describe a technique for estimating the parameters of the model given observations from the model, effectively learning the graph structure underlying the model by using a group-sparsity-promoting regularizer under the assumption that the graph has bounded degree.

The Lincoln Laboratory portion of this work was sponsored by the Department of the Air Force under Air Force Contract #FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

2. THE ALARM MODEL

2.1. Definition

A logistic autoregressive model (ALARM) is defined on a directed graph $G = (V, E)$, where $V = \{v_1, \ldots, v_N\}$ is the set of N vertices and E is the set of directed edges, each of which is an ordered pair of vertices. We write $v_i \to v_j$ if $(v_i, v_j) \in E$, and $v_i \sim v_j$ if either $v_i \to v_j$ or $v_j \to v_i$. The indegree $\mathrm{indeg}(v_i) = |\{v_j : v_j \to v_i\}|$ of a vertex is the number of incoming edges, the outdegree $\mathrm{outdeg}(v_i) = |\{v_j : v_i \to v_j\}|$ is the number of outgoing edges, and the degree $\deg(v_i) = |\{v_j : v_i \sim v_j\}|$ is the total number of vertices connected to v_i one way or another. We define G^* as the undirected version of G, containing an edge $\{v_i, v_j\}$ if $v_i \sim v_j$ in G. We will assume that the indegree is bounded by a constant, so $\mathrm{indeg}(v_i) < D$ for every i, and the $D \ll N$.

Under the ALARM model, we obtain a sequence of random vectors $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(T)} \in \{0,1\}^N$. Each element $y_i^{(t)}$ of $\boldsymbol{y}^{(t)}$ is independent of the others, conditioned on the previous K vectors $\boldsymbol{y}^{(t-1)}, \ldots, \boldsymbol{y}^{(t-K)}$, and takes the value 1 with probability

$$\Pr\left(y_i^{(t)} = 1 \,\middle|\, \boldsymbol{y}^{(t-1)}, \ldots, \boldsymbol{y}^{(t-K)}\right) = \mathrm{logit}^{-1}\left(\sum_{k=1}^{K}\sum_{j=1}^{N} h_{ij}^{(k)} y_j^{(t-k)} + b_i\right) \quad (1)$$

where $\mathrm{logit}^{-1}(x) = \frac{\exp(x)}{1+\exp(x)}$ is the logistic function [the inverse of the function $\mathrm{logit}(x) = \log\left(\frac{x}{1-x}\right)$].

The parameters of the ALARM model are the K matrices $H^{(1)}, \ldots, H^{(K)}$ and the vector $\boldsymbol{b}$. Their elements are effectively logistic regression coefficients linking previous values of the dynamic process to the current values, giving the model its name: it is a vector autoregressive model with a logistic link function. This is the standard link function for generalized lienar regression when the response variables are Bernoulli-distributed.

Our assumption is that the $H^{(\cdot)}$ matrices respect the graph structure, *i.e.*, $h_{ij}^{(k)} \neq 0$ only if $v_j \to v_i$ or $i = j$. Of course, the model is well-defined even on a complete graph, which would allow for every coefficient to be nonzero. But on a true network, the model obeys the structure in a way that can be exploited, as we will show in Section 3. If we treat each time series $(y_i^{(1)}, \ldots, y_i^{(T)})$ as a random variable, the ALARM model is a graphical model described by the graph G^*, meaning that if $j \not\sim i$, then the time series at v_i is independent of the one at v_j conditioned on the time series at all of v_i's neighbors.

2.2. Properties

The ALARM process is a Kth order Markov chain with 2^N states. In general, such a Markov chain requires $2^{NK}(2^N - 1)$ real parameters to define. The ALARM model in general requires at most $N^2K + N$ real parameters (and only $NDK + N$ parameters under the bounded indegree condition). Despite its compact parametric representation, the ALARM model can capture a wide range of interactions.

Consider even just the special case of $K = 1$, so that the state at time t is dependent on the past only through the state at time $t-1$. If $h_{ij}^{(1)} > 0$, then $y_j^{(t-1)} = 1$ makes it more likely that $y_i^{(t)}$ will be 1. If $h_{ij}^{(1)} < 0$, then the opposite is true, and $y_i^{(t)}$ seeks the opposite state of $y_j^{(t-1)}$. If the diagonal element $h_{ii} > 0$, then y_i has "inertia" and may try to stay in the same state; if $h_{ii} < 0$, then y_i may oscillate between 1 and 0 (the specifics depend on the other coefficients and neighboring values).

The value of b_i is a kind of bias. If $b_i = 0$, then $y_i^{(t)} = 1$ with probability 1/2 if all of the neighbors $y_j^{(t-1)}$ were zero. $b_i > 0$ biases $y_i^{(t)}$ toward 1, and $b_i < 0$ biases it toward 0. Thus we can model behavior where neighbors influence each other either positively or negatively, nodes are biased one way or another, and nodes are either stuck in their current value or prone to flip-flopping.

2.3. Examples

To illustrate some of the intriguing behavior that this model can produce, we consider the following special cases. Let $K = 1$, and suppose G is a 1D or 2D lattice graph (undirected) with N nodes. Let A be the adjacency matrix of the graph. For some $\beta > 0$ we define $H = \beta(A + I)$ and $\boldsymbol{b} = -\frac{1}{2}H\boldsymbol{1}$. This value of $\boldsymbol{b}$ ensures the identity $\Pr\left(y_i^{(t)} = 1 \,\middle|\, \boldsymbol{y}^{(t-1)}\right) = \Pr\left(y_i^{(t)} = 0 \,\middle|\, \boldsymbol{1} - \boldsymbol{y}^{(t-1)}\right)$, so that flipping every state in $\boldsymbol{y}^{(t-1)}$ does the same to $\boldsymbol{y}^{(t)}$.

As in [1], we can use this to model influence in a social network. A node whose neighbors are evenly divided will have an equal chance of choosing either state. As the proportion of neighbors in a particular state deviates from that equilibrium, the logistic link function provides for an approximately linear response in the beginning; if the neighbors are nearly unanimous, the logistic function saturates and the node is very likely to join them.

If the initial state $\boldsymbol{y}^{(0)}$ is i.i.d. Bernoulli(1/2), then at time t, every state is as probable as its inverse. If we run the model for some time, does this mean that the final state will have an equal number of 0's and 1's? The question is a practical one: if we are modeling influence on a social network as in [1], then this tells us whether we settle into a consensus decision or a divided state. We might expect that for small β, the interactions are not strong enough to create a consensus, but as β increases, we end up with the vast majority of states either 0 or 1 (with each consensus equally probable).

We simulated the model to answer this question. The results for these two graphs are illustrated in Figure 2. We ran the model for 3000 time steps, and measured the size of the majority group. In each case, the graph size is 1024. Majority sizes near 512 indicate that no consensus is reached, whereas majority sizes nearer to 1024 indicate a consensus. A sharp phase transition is evident in the 2D lattice. As the interaction strength β increases past 1.3, we quickly move from a disordered phase to an ordered one with a strong consensus. Meanwhile, in the 1D case, even allowing β to go as high as 15 does not reveal any such phase transition. The final state is disordered even though the interaction strength is extremely strong.

This result hints at a connection to the Ising model of statistical physics [5]. A realization of the Ising model is a vector $\boldsymbol{z} \in \{-1, +1\}^N$ with probability given by $\Pr(\boldsymbol{z}) \propto \exp\left(\beta \boldsymbol{z}^T A \boldsymbol{z}\right)$, where A is the adjacency matrix of the interaction graph of the system, and β is the inverse temperature. It is a well-known result in physics that the Ising model undergoes a similar phase transition to the one we observe in the ALARM model when the graph is a lattice of dimension 2 or greater, and that there is no phase transition on a 1D lattice [5]. The ALARM model is similar to Markov chain Monte Carlo techniques used to simulate the Ising model; but deeper study of the connection is warranted.

3. PARAMETER ESTIMATION

In this section we present an algorithm for learning the parameters $H^{(1)}, \ldots, H^{(K)}$ and $\boldsymbol{b}$ of the ALARM model from a sequence of observations froom the model. The log-likelihood of the ALARM

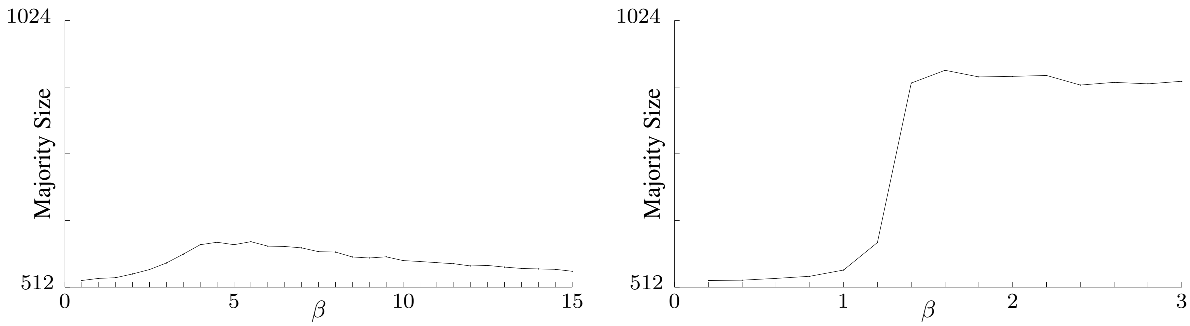

Fig. 2. The size of majority after 3000 steps of the ALARM model is illustrated, for 1D (top) and 2D (bottom) lattice graphs. The initial states are i.i.d. Bernoulli(1/2). The model parameters are $H = \beta(A+I)$, $\boldsymbol{b} = -\frac{1}{2}H\mathbf{1}$, The 2D graph, unlike the 1D graph, has a phase transition. This is reminiscent of the behavior of the Ising model in physics [5].

model [conditioned on the initial states $\boldsymbol{y}^{(1-K)}, \ldots, \boldsymbol{y}^{(0)}$] is given by

$$\begin{aligned}
\ell_{\{\boldsymbol{y}^{(t)}\}}&\left(H^{(1)}, \ldots, H^{(K)}, \boldsymbol{b}\right) \\
&= \sum_{t=1}^{T}\sum_{i=1}^{N}\left[y_i^{(t)}\left(\sum_{k=1}^{K}\sum_{j=1}^{N} h_{ij}^{(k)} y_j^{(t-k)} + b_i\right)\right. \\
&\quad \left. - \log\left(1 + \exp\left(\sum_{k=1}^{K}\sum_{j=1}^{N} h_{ij}^{(k)} y_j^{(t-k)} + b_i\right)\right)\right] \qquad (2) \\
&= \sum_{i=1}^{N} \ell^i_{\{\boldsymbol{y}^{(t)}\}}\left(h_{i\cdot}^{(1)}, \ldots, h_{i\cdot}^{(K)}, b_i\right), \qquad (3)
\end{aligned}$$

where the $\ell^i_{\{\boldsymbol{y}^{(t)}\}}$ are likelihoods for the parameters associated with the response of y_i to the neighboring values:

$$\begin{aligned}
\ell^i_{\{\boldsymbol{y}^{(t)}\}}&\left(h_{i1}^{(1)}, \ldots, h_{iN}^{(1)}, \ldots, h_{i1}^{(K)}, \ldots, h_{iN}^{(K)}, b_i\right) \\
&\stackrel{\text{def}}{=} \sum_{t=1}^{T}\left[y_i^{(t)}\left(\sum_{k=1}^{K}\sum_{j=1}^{N} h_{ij}^{(k)} y_j^{(t-k)} + b_i\right)\right. \\
&\quad \left. - \log\left(1 + \exp\left(\sum_{k=1}^{K}\sum_{j=1}^{N} h_{ij}^{(k)} y_j^{(t-k)} + b_i\right)\right)\right]. \qquad (4)
\end{aligned}$$

The separability of the likelihood means we can learn the coefficients associated with the ith node independently of the others (but note that each independent log-likelihood uses all of the data.) This will simplify the analysis and allow for embarassingly parallel algorithms to learn all the parameters. This learning really amounts to N logistic regression problems.

Let us consider the problem of learning the parameters associated with a single vertex: $h_{i\cdot}^{(1)}, \ldots, h_{i\cdot}^{(K)}$ and b_i. The unknown graph structure described in Section 2 guarantees that for each k, the only non-zero variables out of $h_{i1}^{(k)}, \ldots, h_{iN}^{(k)}$ are the D variables $h_{ij}^{(k)}$ for $j \to i$. This is a *group sparsity* [6] constraint on the parameter vector $\boldsymbol{\theta} = \left(\boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_N^T\right)^T \stackrel{\text{def}}{=} \left(h_{i1}^{(1)}, \ldots, h_{i1}^{(K)}, \ldots, h_{iN}^{(1)}, \ldots, h_{iN}^{(K)}\right)^T$. Unlike a sparsity constraint, which would limit the number of nonzero entries of $\boldsymbol{\theta}$, the group sparsity constraint limits the number of subvectors $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$ that are not identically $\mathbf{0}$. Each subvector is associated with a neighboring vertex, and so at most D can be nonzero.

Directly incorporating this constraint into the maximum likelihood procedure would result in a hard combinatorial problem. But we can use the standard approach of relaxing the constraint using the $\ell_{2,1}$ mixed norm defined by $\|\boldsymbol{\theta}\|_{2,1} = \sum_{i=1}^N \|\boldsymbol{\theta}_i\|_2$ as a convex regularizer. The ℓ_2 part of the norm does not privilege any direction in the subspace associated with each vertex; but the ℓ_1 part of the norm promotes a group-sparse solution where only a small number of vertices are associated with non-zero values. We obtain the estimator

$$\left(\widehat{\boldsymbol{\theta}}, \widehat{b}_i\right) = \arg\min_{\boldsymbol{\theta}, b_i} \ell^i_{\{\boldsymbol{y}^{(t)}\}}(\boldsymbol{\theta}, b_i) + \lambda\|\boldsymbol{\theta}\|_{2,1}, \qquad (5)$$

or, more explicitly,

$$\begin{aligned}
&\left(\widehat{h}_{i1}^{(1)}, \ldots, \widehat{h}_{iN}^{(1)}, \ldots, \widehat{h}_{i1}^{(K)}, \ldots, \widehat{h}_{iN}^{(K)}, \widehat{b}_i\right) \\
&\quad = \arg\min_{h_{i\cdot}^{(\cdot)}, b_i} \ell^i_{\{\boldsymbol{y}^{(t)}\}}(h_{i\cdot}^{(\cdot)}, b_i) + \lambda \sum_{j=1}^{N} \sqrt{\sum_{k=1}^{K} h_{ij}^{(k)2}}, \qquad (6)
\end{aligned}$$

where λ is a nonnegative regularization parameter. The regularization function does not include b_i because we have no reason to expect that $\boldsymbol{b}$ is sparse. The function to be minimized in (6) is convex, so it should be efficiently solvable. In fact, it is closely related to lasso and group-lasso logistic regression problems, for which several efficient algorithms exist [7,8], and which can be shown to be consistent estimators [9].

To illustrate the utility of such techniques, we consider the problem of reconstructing the graph G from a realization of the ALARM model. The analogous problem for linear multivariate autoregressive models with Gaussian noise has been considered in [10]. Suppose we have a model with an unknown graph and $K = 1$. If we use ℓ_1-regularized logistic regression to reconstruct each row of $H \stackrel{\text{def}}{=} H^{(1)}$, then we will obtain a matrix with many zero entries, due to the sparsity-recovery properties of the ℓ_1 regularization. As λ increases, more and more entries of $\widehat{H}$ will be set to zero. If $v_j \to v_i$ but $\widehat{h}_{ij} = 0$, then we will characterize that as a mis-detection; if $v_j \not\to v_i$ but $\widehat{h}_{ij} \neq 0$, then we will characterize it as a false alarm. Varying λ, we obtain a ROC curve. In Figure 3 we illustrate the results of this experiment for various graph structures and connection strengths. (We define $h_{\min} = \min_{i,j:h_{ij}\neq 0} |h_{ij}|$.)

For the experiment, we used either a random geometric graph ($N = 100$, $D = 12$) or a 2D lattice ($N = 256$, $D = 4$). We set $H = \beta A$, where A was the adjacency matrix of the graph, and $\beta = 0.1$ or $\beta = 0.2$. A realization of the ALARM model was generated with $T = 2000$. We used a `l1_logreg`, a publicly available code for performing ℓ_1-regularized logistic regression [7]. The regularization parameter λ was varied and the detection and false alarm probabilities were computed to create the ROC curves. The results are shown in Figure 3. At the small values of $h_{\min} = \beta$ we used,

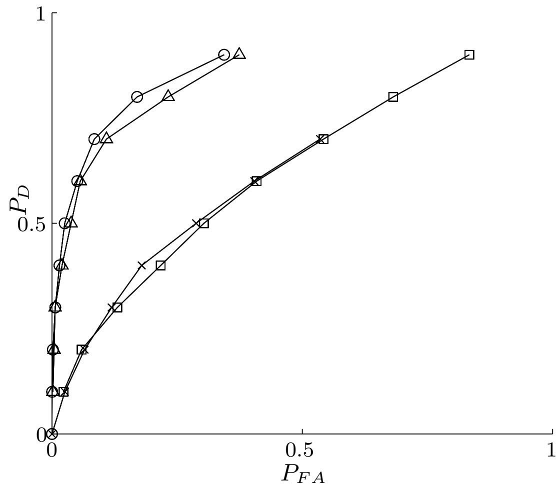

Fig. 3. ROC curves for the detection of edges based on ALARM realizations on various graph structures are illustrated. A random geometric graph with D = 12 is used with $h_{\min}$ = 0.2 (∘) and $h_{\min}$ = 0.1 (×), and a 2D lattice graph with D = 4 is used with $h_{\min}$ = 0.2 (△) and $h_{\min}$ = 0.1 (□). The performance appears to depend most strongly on $h_{\min}$.

perfect reconstruction is not possible, leading to meaningful ROC curves. Despite the different types of graph, the ROC curves are nearly identical so long as $h_{\min}$ is the same. Analytical characterization of the performance is left to future work.

4. CONCLUSIONS

We introduced the ALARM model, a logistic autoregressive model for binary processes on networks. This model is very flexible, able to capture several kinds of interactions between nodes on a network. We illustrated some of the interesting behavior that this model can produce, such as a phase transition when the underlying graph is a 2D lattice that is absent when it is only a 1D lattice. We also considered the problem of estimating the parameters of the system and thereby reconstructing the underlying graph. We showed how a group-sparsity-promoting regularizer can be used to aid in the recovery of the graph structure. Questions of consistency, as well as further analysis of the intriguing phase-transition behavior and connections to the Ising model, are left to future work.

5. REFERENCES

[1] C. Asavathiratham, “The influence model: A tractable representation for the dynamics of networked Markov chains,” Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[2] A. Barrat, *Dynamical processes on complex networks*. Cambridge: Cambridge University Press, 2008.

[3] R. A. Holley and T. M. Liggett, “Ergodic theorems for weakly interacting infinite systems and the voter model,” *The Annals of Probability*, vol. 3, no. 4, pp. 643–663, Aug. 1975.

[4] M. Granovetter, “Threshold models of collective behavior,” *American Journal of Sociology*, vol. 83, no. 6, pp. 1420–1443, May 1978.

[5] L. E. Reichl, *A Modern Course in Statistical Physics*. Wiley, Aug. 2009.

[6] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, p. 49–67, 2006.

[7] K. Koh, S.-J. Kim, and S. Boyd, “An interior-point method for large-scale l1-regularized logistic regression,” *Journal of Machine learning research*, vol. 8, no. 8, p. 1519–1555, 2007.

[8] L. Meier, S. Van De Geer, and P. Bühlmann, “The group lasso for logistic regression,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 1, p. 53–71, 2008.

[9] P. Ravikumar, M. J. Wainwright, and J. D. Lafferty, “High-dimensional Ising model selection using l1-regularized logistic regression,” *The Annals of Statistics*, vol. 38, no. 3, p. 1287–1319, 2010.

[10] A. Bolstad, B. Van Veen, and R. Nowak, “Causal network inference via group sparse regularization,” *IEEE Transactions on Signal Processing*, vol. 59, no. 6, pp. 2628 –2641, Jun. 2011.