

A Deterministic Model for History Sensitive Cascade in Diffusion Networks

Yu Zhang

Department of Computer Science
Trinity University
San Antonio, Texas, 78212
yzhang@cs.trinity.edu

Abstract— This paper studies information diffusion in networks. Traditional models are all history insensitive, i.e. only giving activated nodes a one-time chance to activate each of its neighboring nodes with some probability. But history dependent interactions between people are often observed in real world. This paper propose a new model called the *History Sensitive Cascade Model (HSCM)* that allows activated nodes to receive more than a one-time chance to activate their neighbors. HSCM is a deterministic model to decide the probability of activity for any arbitrary node at any arbitrary time step. In particular, we provide 1) a polynomial algorithm for calculating this probability in tree structure graphs, and 2) a Markov model for calculating the probability in general graphs. This paper makes a theoretical contribution on studying the information diffusion problem.

Keywords—diffusion network, information cascade

I. INTRODUCTION

Diffusion is a process by which information, viruses, ideas and new behavior spread over social networks [15]. The current work in this is an outgrowth of Granovetter's initial treatment of the phenomenon of collective behavior in 1978 [12]. He introduces a threshold model and uses it to examine the occurrence of riots and their perceived domino-effect growth pattern. The result was an early threshold model for collective behavior. In present day, diffusion models of social networks have been studied in a variety of fields ranging from epidemiology [16], to marketing [17], to technology transfers [4, 5], to computer virus transmission [2], and to power systems [22].

As described by Kempe et. al. [14], there are two basic diffusion models: 1) the *linear threshold model* [12], in which a node becomes active if a predetermined fraction, called a threshold, of the node's neighbors are active, and 2) the *independent cascade model* [8], whenever a node becomes active, it gets a one-time chance to activate each of its neighboring nodes with some probability. There is a rich literature in both models; especially the independent cascade model has gained much attention in present day. Goldenberg et al. [11] simulate Word-of-Mouth information diffusion through strong ties among members of the same network and weak ties

among individuals belonging to different network. They found the influence of weak ties on the information diffusion is almost as strong as the influence of strong ties. Cowan and Jonard [7] study diffusion of knowledge in different network structures. They find that the performance of the system exhibits clear “small world” properties, in that the steady-state level of average knowledge is maximal when the structure is a small world (that is, when most connections are local, but roughly 10 percent of them are long distance). In viral marketing, Leskovec et. al [15] simulate information cascade in a real person-to-person recommendation network. They discover that the distribution of cascade sizes is approximately heavy-tailed; cascades tend to be shallow, but occasional large bursts of propagation can occur.

The model that we propose is called the *History Sensitive Cascade Model (HSCM)*. It can be understood as a modified Independent Cascade Model, but is different from the generalized framework (proposed by Kempe) designed to unify the Independent Cascade Model and the Linear Threshold Model underneath a single mathematical threshold. For one major difference, our model allows that activated nodes receive more than a one-time chance to activate their neighbors. A node can switch from being uninformed to being informed, but not the reverse. The rationale of this model is the classical threshold mechanism of collective action: a consumer does not feel social pressure if just a few people around her behave in a particular way but once these people reach a certain number then she suddenly decide to change her mind and she behaves differently [12]. The history dependent interactions between people are often observed in real world. For example, in viral marketing and advertising, a customer may not decide to buy a product at their first time receiving the recommendation or watching the advertisement, but they may gradually accept the product and decide to buy it after several rounds of such interactions. We provide an in depth comparison between HSCM and several of the common diffusion models in Section 2.

This paper formally defines HSCM and provides two algorithms for calculating the probability of activity for any arbitrary node at any arbitrary time. The first is a polynomial algorithm for calculating the probability in tree structure graphs (Section 3). The polynomial feature of this algorithm is nice, comparing to the NP-Complete general complexity for this problem. There are real-world cases where tree structures would be useful tools: for example, the network of a

corporation could be modeled with a tree. Corporations have top-level CEOs, high-level managers, middle-managers, low-level office-managers, office workers, and so on. In cases like these, there is a lot of information traveling downward, but not much information traveling upward. The second algorithm is a Markov model for calculating the probability in general graphs (Section 4). In Section 5, we perform an empirical study on HSCM under different network settings. These simulations have showed its power to observe and explain the emergent phenomena in the macro level when changing parameters in the micro level.

II. BACKGROUND AND MOTIVATION

We will use an example to introduce the related work on diffusion models as well as to show the difference between HSCM and these existing models. Fig. 1 is our graph.

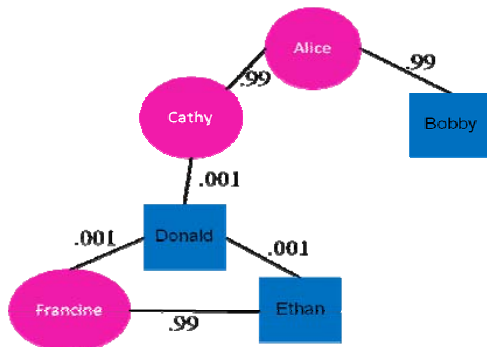


Fig. 1 An example Diffusion Network

Suppose Alice, in high school, is dating Bob. Also, Alice is best friends with Cathy who is barely acquainted with a college boy named Donald. Suppose Donald has two college friends named Ethan and Francine and that these two friends are also dating. The connecting lines represent avenues of contact: Alice routinely communicates with Cathy (her best friend) and Bobby (her boyfriend); but she doesn't know Donald, Francine, or Ethan. Now suppose that Cathy recently bought an iPod and likes it very much. The question we are interested in is what is the probability that other people in the network will be influenced by Cathy and buy an iPod at any given time in the future.

In literature, there are two common diffusion models that have been used to solve the above question that we are interested in. The first is the *Linear Threshold Model*, which says that a node becomes active if a predetermined fraction of the node's neighbors is active. Fig.2 illustrates this rule. Here the predefined threshold is 60%; red nodes mean the nodes that have already being activated via the link pointing to them, and we are determined if the centre bold node will be activated in the given situation. Since the centre node has five neighbors and three of them (i.e. 60%) have been active, so the centre node will be active too.

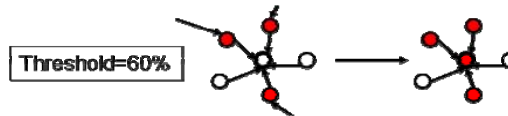


Fig. 2 An Example of the Linear Threshold Model

The second model is the *Independent Cascade Model*. Under this model, an active node gets a one-time chance to activate each of its neighboring nodes with some probability. Fig. 3 illustrates this rule. Here the probability is 50%, as a consequence, the already active central bold node makes two of its neighbors active too.

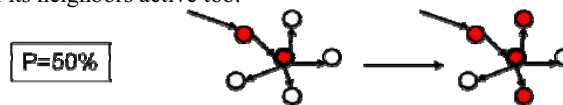


Fig. 3 An Example of the Independent Cascade Model

There are two common features of both the Linear Threshold Model and the Independent Cascade Model: 1) *progressiveness*: once a node becomes active, it will never deactivate, and 2) *historical insensitivity*: each non-active node is only triggered once, leaving the node either active or not depending on the predefined threshold. Then the node is passed and will never be influenced again.

The HSCM model that we propose shares the progressiveness feature of the existing models but it is *history sensitive*. We can find many examples in really world that people influence each other by more than one time and the previous influence enhances the current influence. Let's refer to Fig.1 again for an example scenario. The probability attached to each link is called the "spreading probability", representing the chance that the influence of the iPod ideology will spread across that line during a fixed length of time. In this example, let's suppose this length of time equals one week. It is reasonable to think that Alice has a higher chance of being influenced by Cathy not only because the spreading probability between them is higher but also this probability will be increased if they spend more time together, under the assumption that Alice's has no bias to iPod as others, i.e. nobody in this system has a strong attitude in favoring or disfavoring iPod originally.

It should be noted that we are not suggesting real world situations are always like this. Certainly one could argue that Alice after many weeks of being pestered by Cathy would decide out of annoyance never to buy an iPod. Or she may simply become "immune" to Cathy's suggestions, becoming less and less likely to buy an iPod with each passing exposure to Cathy's iPod ideology. These concerns are valid. But that doesn't mean historical data should be entirely disregarded as is the case in all diffusion models we have encountered.

III. THE HSCM MODEL – AN OVERVIEW

Let $G = (V, E)$ where V is a set of vertices and E is a set of edges. Each vertex v in V corresponds to a Boolean value $A(v)$, denoting whether the vertex is "active" or not. Each edge $e_{v,u}$ (leaving v and entering u) is weighted with a value

$0 < W(e_{v,u}) \leq 1$, representing the probability that if v is active in time step $t=k$, then u will be active in time step $t=k+1$. $W(e_{v,u})$ is called the spreading probability. Let $targets(v)$ represent all u such that $e_{v,u}$ is in E . The activity value of vertices is updated as the HSCM Algorithm in Fig. 4.

```

Function HSCM[ $G=(V, E), W(e_{v,u})$ ]
Inputs:  $G=(V, E)$  where  $V$  is a set of vertices and  $E$  is a
set of edges, with some initially active
vertices.
 $W(e_{v,u})$ , the spreading probability, that if  $v$  is
active in time step  $t$ , then  $u$  will be active in
time step  $t+1$ .

For time step = 1 to  $k$ 
For each vertex  $v$  in  $V$ 
If  $A(v) = true$ 
For each vertex  $u$  in  $targets(v)$ 
random = a random number between 0 and 1;
If random <  $W(e_{v,u})$ 
Set  $A(u) = true$ ;

```

Fig. 4 The HSCM Algorithm

The question we wish to answer is called the Activation Probability Problem, which is defined below.

Definition 1 (Activation Probability Problem). Given some time step k and some vertex v , what is the probability that v will be active at $t=k$?

We assume (for the sake of eschewing trivialities) that at time step $t=0$, there exists at least one node a such that $A(a)=true$. Or, as we may alternatively posit, $P(a_0)=1$, meaning simply that the probability of a being active on the 0th time step is 1. If this were not the case, then there would be no activity for any nodes on any time step. The proof for this can easily be extrapolated from the above algorithm, whose only alterations to the state of the system occur within an If-statement, the condition of which is that $A(v)$ be *true*. Thus, a system starting out with $A(v)$ being *false* for all v will never change states.

It can be easily proved that the Activation Probability Problem is NP-Complete, by deducing it to the Set Covering problem. Therefore, in this paper, we provide two solutions to it: 1) a polynomial solution for tree structure graphs in Section IV, and 2) a Markov Model solution for general graphs in Section V.

IV. A POLYNOMIAL SOLUTION FOR TREE GRAPHS

We now go about examining properties of the HSCM model in order to derive a method for calculating the probability of activity for any arbitrary node at any arbitrary time in tree structure graphs. Throughout the discussion, suppose for simplicity's sake that we have only one active vertex a at time $t=0$. Ultimately, we will drop this assumption.

Theorem 1 (Definition of Inactivity): If $P(x_k) = 0$, then $P(x_{k-i}) = 0$, for all $0 <= i <= k$.

Proof: That is equivalent to asserting that, if a vertex is not active during $t = k$, it was never active at a previous time step. The proof follows from the fact that our previous description of the model does not provide for any conditions in which an active vertex might deactivate. So an inactive vertex was never active previously. ■

Calculating the probability the neighbor of an active vertex will be active in the next time step is a simple matter.

Lemma 1: Assuming that 1) x only has one edge coming to it from an active vertex (which we call a), 2) $P(x_0) = 0$, 3) $P(a_0) = 1$, and 4) $t = 1$,

$$P(x_1) = W(e_{a,x}). \quad (3)$$

Proof: The proof follows from the definition of the model. In time step $t = 0$, vertex a is active and, thus, will activate any inactive neighbor n with a probability equal to the weight of the edge from a to n . Since this lemma assumes that $t = 1$ and that x was not active at $t = 0$, we know that x has had no chance to become active before $t = 1$. Thus, $P(x_0) = 0$ and $P(x_1) = W(e_{a,x})$, as can be seen from the statement of the Simulation Algorithm. Matters, however, will not be so simple in the following case. ■

Lemma 2: Dropping the forth assumption above, assume that 1) x has only one edge coming to it from an active vertex (which we call a), 2) $P(x_0) = 0$, 3) $P(a_0) = 1$.

$$P(x_{k+1}) = [1 - P(x_k)]W(e_{a,x}) + P(x_k). \quad (4)$$

Proof: It helps to note that, because of the first assumption above (combined with the Definitions of Activity and Inactivity), x has been the neighbor of one and only one active vertex ever since $t = 0$, i.e. $P(a_k) = 1$ for all k . This is useful because it means that we need only pay attention to the following two ways that x might become active in time step $t = k + 1$.

- 1) x had been activated by a already during some time step $t <= k$, or
- 2) x is activated by a during the step $t = k + 1$.

We can represent the first of these cases like so,

$$already_{k+1} = P(x_k). \quad (5)$$

And the second like so,

$$during_{k+1} = [1 - P(x_k)]W(e_{a,x}). \quad (6)$$

This equation makes use of the fact that $P(x_k)$ is the probability that x was active before $t=k+1$, and thus, $1 - P(x_k)$ is the probability that x was not active before $t = k + 1$. Since the probability of $A(x)$ being true already at $t = k+2$ and of becoming true during $t = k + 2$ are disjoint, the probability of x being active at $t = k + 2$ is given by a simple sum:

$$P(x_{k+1}) = during_{k+1} + already_{k+1} \quad (7)$$

which is the same as

$$P(x_{k+1}) = [1 - P(x_k)]W(e_{a,x}) + P(x_k) \quad \blacksquare$$

Lemma 3: Dropping, now, the third assumption above, we need not assume that $P(a_k) = 1$ for all k . Instead, we assume that 1) x has only one edge coming to it from any potentially active vertex a , and 2) $P(x_0) = 0$. In this case, the activity probability is:

$$P(x_{k+1}) = [P(a_k) - P(x_k)]W(e_{a,x}) + P(x_k) \quad (8)$$

Proof: This is almost identical to the one above except that the probability of x being activated during $t=k+1$ is this:

$$\text{during}_{k+1} = [P(a_k) - P(x_k)]W(e_{a,x}). \quad (9)$$

The justification for the term $P(a_k) - P(x_k)$ is again similar to the previous proof. It makes use of the fact that if x has ever had an active influencer, the probability of x being active is completely dependent on the probability of a being active, because a is and always has been x 's only means of becoming active. Finally,

$$P(x_{k+1}) = [P(a_k) - P(x_k)]W(e_{a,x}) + P(x_k) \quad \blacksquare$$

Using the above lemmas, we can calculate the activity probability of any arbitrary vertex on any arbitrary time step, provided that the graph is a tree and that no vertex has more than one incoming edge.

Theorem 3: Let $G = (V,E)$ be a graph without cycles, and let there be no two edges in E , $e_{w,x}$ and $e_{y,z}$, such that $x=z$. We will use the function $\text{influencer}(u)$ to denote the vertex v such that $\text{targets}(v)$ includes u .

$$P(v_{k+1}) = \begin{cases} 1, & \text{if } P(v_0) = 1 \\ [P(\text{influencer}(v)_k) - P(v_k)] \times W(e_{\text{influencer}(v),v}) + P(v_k), & \text{otherwise} \end{cases} \quad (10)$$

Proof: The first part of the formula follows from the definition of activity. The second part follows from observing that the vertices will always satisfy the assumptions of either Lemma 2 or 3:

- 1) Vertex v will always have only one potentially active influencer.
- 2) $P(v_0)$ will, in all cases be 0, or else the first part of the formula would have been applicable. \blacksquare

Example 1. Consider this initial network in left-top of Fig. 5. Also suppose that $\text{threshold}=0.65$ and that initially only node 1 is active. After 4 time steps, the network achieves activation. The probabilities are iteratively updated along with time steps by Equation (10).

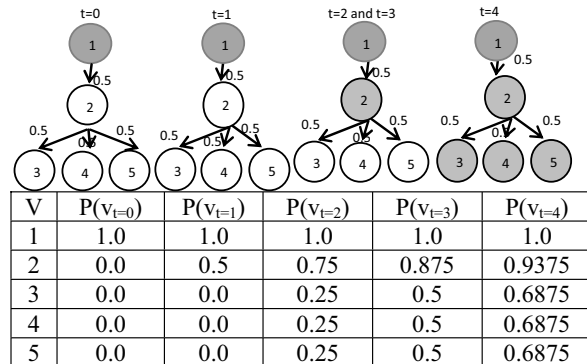


Fig. 5 An Example of HSCM for a Tree Graph

Two features can be directly observed from Fig. 5 regards the HSCM model for a tree graph. First, at any given time step k , the activation probability of an influencer node is always higher than that of its targeting nodes. Second, for any given node v , its activation probability at time step $k+1$ is always higher than that at time step k . Next we prove these two features.

Theorem 4: $\forall t=k, P(\text{influencer}(v)_k) > P(v_k)$, if $P(v_0) \neq 1$ and $0 < W(e_{\text{influencer}(v),v}) \leq 1$.

Proof by Induction: Fig. 6 illustrates a general status of a tree graph.

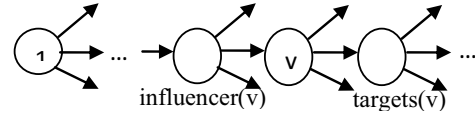


Fig. 6 A General Status of a Tree Graph

Basis Step: At time step $t=0$, there are only two types of nodes in G : the initially active root node or nodes in top levels whose probability is 1, and the non-active low level nodes whose probability is 0. So the basis step is proved.

Inductive Step. Assume that at time step $t=k$, we have

$$\dots > P(\text{influencer}(v)_k) > P(v_k) > P(\text{targets}(v)_k) > \dots \quad (11)$$

We want to prove that at time step $t=k$, we have

$$\dots > P(\text{influencer}(v)_{k+1}) > P(v_{k+1}) > P(\text{targets}(v)_{k+1}) > \dots \quad (12)$$

Proof. From Equation (10):

$$P(v_{k+1}) = [P(\text{influencer}(v)_k) - P(v_k)] \times W(e_{\text{influencer}(v),v}) + P(v_k) \quad (13)$$

$$P(\text{targets}(v)_{k+1}) = [P(v_k) - P(\text{targets}(v)_k)] \times W(e_{v,\text{targets}(v)}) + P(\text{targets}(v)_k) \quad (14)$$

$$\begin{aligned} \text{Therefore, } P(v_{k+1}) - P(\text{targets}(v)_{k+1}) &= [P(\text{influencer}(v)_k) - P(v_k)] \times W(e_{\text{influencer}(v),v}) + P(v_k) - \\ & [P(v_k) - P(\text{targets}(v)_k)] \times W(e_{v,\text{targets}(v)}) + P(\text{targets}(v)_k) \\ &= [P(\text{influencer}(v)_k) - P(v_k)] \times W(e_{\text{influencer}(v),v}) + \\ & [P(v_k) - P(\text{targets}(v)_k)] \times (1 - W(e_{v,\text{targets}(v)})) \end{aligned} \quad (15.a)$$

$$> 0, \text{ because both } (15.a) > 0 \text{ and } (15.b) > 0. \quad (15.b)$$

So $P(v_{k+1}) > P(\text{targets}(v)_{k+1})$. By similar method, we can prove $P(\text{influencer}(v)_{k+1}) > P(v_{k+1})$. \blacksquare

Theorem 5: $\forall v \in V, P(v_{k+1}) > P(v_k)$.

Proof: From *Theorem 4*:

$$\begin{aligned} & [P(\text{influencer}(v)_k) - P(v_k)] \times W(e_{\text{influencer}(v),v}) > 0 \quad (16) \\ \Rightarrow & [P(\text{influencer}(v)_k) - P(v_k)] \times W(e_{\text{influencer}(v),v}) + P(v_k) > P(v_k) \\ \Rightarrow & P(v_{k+1}) > P(v_k) \end{aligned} \quad \blacksquare$$

V. A MARKOV SOLUTION FOR GENERAL GRAPHS

Expanding our ability to perform this calculation on graphs with cycles presents several problems. The tip of the iceberg with regard to cycles can be illustrated by the situation in Fig. 7.

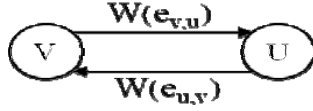


Fig. 7 A Simple Graph with a Loop

In Fig. 7, we assume that $P(v_k) > 0$ and v has an edge to u , which is inactive on time step $t = k$. On next two steps, according to Equation (10), we have:

$$t=k+1, P(u_{k+1})=P(v_k) \times W(e_{v,u}) < P(v_k), \text{ and} \quad (17)$$

$$t=k+2, P(v_{k+2})=[P(u_{k+1})-P(v_k)] \times W(e_{u,v}) + P(v_k) < P(v_k). \quad (18)$$

But (17) is against *Theorem 4* and (18) is against *Theorem 5*. Therefore Equation (10) can't be applied to general graphs. This example is just a cycle of length one. Much more complicated cases arise when cycles are longer.

First, we consider the graph to be a finite state system, where "state" is understood as some combination of activated vertices in V . The state of the graph can be represented as a Boolean-valued "activity" vector a of length n , where n is the size of V . Let each vertex be arbitrarily labeled with a unique value between 0 and $n-1$, and let this label value be denoted $L(v)$. A value of 1 in the index i (from the right) of the vector a indicates that the vertex v , such that $L(v) = i$, is active. More compactly, $a[L(v)] = A(v)$. Thus, the vector

$$a=[1 \ 0 \ 1 \ 0 \ 0 \ 1]$$

indicates that for all v such that $L(v) = 0, 3, \text{ or } 5, A(v) = true$. Now, let us use another binary-valued vector called s to denote every possible binary number that the vector a can represent. We call s the "state" vector and it will be of length 2^n . s will have only a single 1-value; the rest will be 0s. The index of s ' 1-value, when converted to base two will be the binary number represented by a corresponding activity vector a . Thus, the state space vectors

$$s=[1 \ 0 \ 0 \ 0] \quad s=[0 \ 1 \ 0 \ 0] \quad s=[0 \ 0 \ 1 \ 0] \quad s=[0 \ 0 \ 0 \ 1]$$

would map to the following activity vectors respectively:

$$a=[0 \ 0] \quad a=[1 \ 0] \quad a=[0 \ 1] \quad a=[1 \ 1]$$

We are introducing the s vector in order to serve as the initial state vector for a Markov chain that we will build from G , allowing us to calculate the exact probability of some vertex being active at some time step. Furthermore, each binary value represented by the list a can be considered an index into the matrix we will construct. We give the algorithm in Fig. 8 for constructing the state transition matrix for a general graph $G = (V, E)$.

Upon obtaining the stochastic state transition matrix A , the calculation of each possible state reachable from an initial binary state vector s – containing only a single 1-value, can be obtained as follows:

$$p = s \times A^k \quad (19)$$

where k is the time step for which you want a value. The resulting vector p has the property that $p[i]$ equals the probability that $s[i] = 1$ at time step $t = k$.

To find $P(v_k)$ for an arbitrary vertex v , we simply need to sum the elements of p whose state involves v being active:

$$P(v_k) = s \times A^k \times AllRelevantStates(L(v)) \quad (20)$$

where $AllRelevantStates(L(v))$ merely returns a vector x such that $x[j]$ is 1 if the binary representation of j contains a 1 in the $L(v)$'s location. Otherwise $x[j]$ is a 0. So in the end, $P(v_k)$ can be written:

$$P(v_k) = p \times AllRelevantStates(L(v)). \quad (21)$$

This sums up the values of p that refer to the probability of G being in a state where v is active. The reason we can sum these probabilities is that each state whose transition probability we calculate during the construction of a is disjoint, as can be seen from the portion of the algorithm labeled *Make Probabilities Disjoint*, which calculates, for every initial state S , the probability of activating any subset of all vertices a that could be activated by vertices in S , and (most importantly) the probability of *not* activating any other vertices in a . Thus, state transition probabilities do not "overlap," so to speak. Fig. 8 is the integrated algorithm for updating the state transaction matrix.

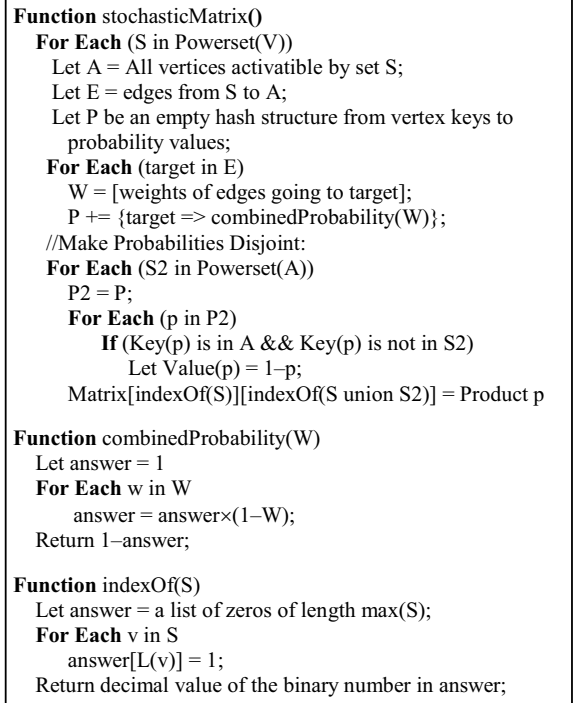
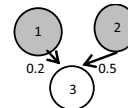


Fig. 8 Stochastic Matrix Construction Algorithm

Example 2. We consider a simple graph like this. Initially both nodes 1 and 2 are active. We show how to calculate the activate probability of node 3 at any given time step t .



Our transition matrix A would look like this in Fig. 9. $A[i,j]$ is defined as the probability that the network will move from state i to state j . When $i=j$, $A[i,i]$ is the probability that the network will remain in its current state – meaning that no new nodes become active.

ij	1	2	1,2	3	3,1	3,2	3,1,2
1	1.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.8	0.0	0.0	0.0	0.2	0.0
$1,2$	0.0	0.0	0.5	0.0	0.0	0.0	0.5
3	0.0	0.0	0.0	0.4	0.0	0.0	0.6
$3,1$	0.0	0.0	0.0	0.0	0.4	0.1	0.4
$3,2$	0.0	0.0	0.0	0.0	0.0	0.5	0.5
$3,1,2$	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Fig. 9 The Transition Matrix

As an example, we show how $A[i,j]=0.6$ when $i=(1,2)$ and $j=(3,1,2)$. Now we are assuming that only nodes 1 and 2 are active. We want to know the probability that 3 becomes active. There are two edges to 3. Either edge can activate 3, but not both. So we use the principle of inclusion/exclusion:

$$0.5 + 0.2 - (0.5 \times 0.2) = 0.6.$$

Upon obtaining the stochastic state transition matrix A , next we calculate $P(3_t)$, the probability that node 3 is activated at any time step t . We need an Initial State Vector. Let $I=[0,0,0,1,0,0,0,0]$. Assume threshold=0.65. We need to multiply $I \times (A^t)$ and add up all the values in the result that correspond to a column pertaining to the vertex 3. We add up the last three columns because (if we look at the column headers of the transition matrix above) we can see that these three values correspond to states in which the vertex 3 is active. Next two timesteps:

$P(3_{t=1}): I \times (A^1) = [0.0 \ 0.0 \ 0.0 \ 0.4 \ 0.0 \ 0.0 \ 0.0 \ 0.6]$; Sum of last three columns = 0.6.

$P(3_{t=2}): I \times (A^2) = [0.0 \ 0.0 \ 0.0 \ 0.16 \ 0.0 \ 0.0 \ 0.0 \ 0.84]$; Sum of last three rows = 0.84 (now it is active).

VI. CONCLUSION

We have formalized the HSCM model. One limitation of our model (in the case of general graphs) is its highly intractability for most graphs. In the future, we will simplify the model by further analyzing its features. For example, the computation can be decreased by utilizing the fact that the low triangle of the transition matrix in Fig 9 is 0.

Also in the future, we will study the influence maximization problems under different time constraints for our model. The influence maximization problems are typically phrased in the following terms [9]: given some value k and some diffusion network with a set of nodes N , the goal is to select an initially active k -node subset from N , such that the number of nodes in N that eventually becomes active is maximized. The maximization problem is NP-Hard [14] but the answer can be approximated by heuristics like greedy [19], hill climbing [14], and simulated annealing [13], and can be

bounded on the quality of the cost-effective outbreak detected in the network [15].

ACKNOWLEDGMENT

This work was supported in part by the U.S. National Science Foundation under Grants IIS 0755405 and CNS 0821585.

REFERENCES

- [1] Ahn Y, Han S, Kwak H, *Analysis of Topological Characteristics of Huge Online Social Networking Services*, International World Wide Web Conference, 2007.
- [2] Albert R, Jeong H and Barabasi A, *Error and Attack Tolerance of Complex Networks*, Nature, 406: 378-382, 2000.
- [3] Albert R, Barabási AL, *Statistical Mechanics of Complex Networks*, Reviews of Modern Physics, 74(1): 47-97, 2002.
- [4] Bass F, *A New Product Growth Model for Consumer Durables*, Management Science, 15: 215-227, 1969.
- [5] Brown J and Reinegen P, *Social Ties and Word-of-Mouth Referral Behavior*, Journal of Consumer Research, 14(3): 350-362, 1987.
- [6] Carley KM, Diesner J, Reminga J, Tsvetovat M, *Toward an Interoperable Dynamic Network Analysis Toolkit*, Decision Support Systems, Special Issue on Cyberinfrastructure for Homeland Security: Advances in Information Sharing, Data Mining, and Collaboration Systems, 43(4): 1324-1347, 2007.
- [7] Cowan R and Jonard N, *Network Structure and the Diffusion of Knowledge*, Journal of Economic Dynamics and Control, 28(8): 1557-1575, 2004.
- [8] Culotta A, *Maximizing Cascades in Social Networks*, University of Massachusetts, Amherst, MA, Technical Report, 2003.
- [9] Domingos P and Richardson M, *Mining the Network Value of Customers*, KDD, pp, 57-66, 2001.
- [10] Gladwell M, *The Tipping Point: How Little Things Can Make a Big Difference*, Little Brown, LONDON, UK, 2000.
- [11] Goldenberg J, Libai B, Muller E, *Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth*, Marketing Letters, 3(12): 211-223, 2001.
- [12] Granovetter M, *Threshold Models of Collective Behavior*, American Journal of Sociology 83(6):1420-1443, 1978.
- [13] Jackson MO and Yariv L, *Diffusion on Social Networks*, Public Economics, 16(1): 3-16, 2005.
- [14] Kempe D, Kleinberg J, and Tardos E, *Maximizing the Spread of Influence through a Social Network*, The 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137-146, 2003.
- [15] Leskovec J, Krause A and Guestrin C, *Cost-effective Outbreak Detection in Networks*, KDD, 2007.
- [16] Morris S, *Contagion*, The Review of Economic Studies, 67(1): 57-78, 2000.
- [17] Richardson M and Domingos P, *Mining Knowledge-Sharing Sites for Viral Marketing*, in Proceedings of KDD'08, pp, 61-70, 2002.
- [18] Rogers EM, *Diffusion of Innovations*, Free Press of Glencoe, Macmillan Company, 1962.
- [19] Rolfe M, *Social Networks and Threshold Models of Collective Behavior*, Preprint, University of Chicago, 2004.
- [20] Wasserman S, Faust K, *Social Network Analysis: Methods and Applications*, Cambridge University Press, Cambridge, 1994.
- [21] Watts DJ, *Small Worlds: The Dynamics of Networks Between Order and Randomness*, Princeton Press, 1999.
- [22] Watts DJ, *A Simple Model of Global Cascades in Random Networks*, in Proceedings of the National Academy of Sciences, 99(9): 5766-5771, 2002.
- [23] Watts, DJ and Peretti J, *Viral Marketing for the Real World*, Harvard Business Review. May 1, 2007.