**An Agent-Based Model of Information Diffusion**

Neza Vodopivec

Applied Math and Scientific Computation Program

nvodopiv@math.umd.edu

Advisor: Dr. Jeffrey Herrmann  
Mechanical Engineering Department   
jwh2@umd.edu

**Abstract**: Understanding how information spreads throughout a population can help public health officials improve how they communicate with the public in emergency situations. In this project, I implement an agent-based information diffusion model inspired by the Bass model. I compare my discrete-time simulation to a traditional differential-equation conceptualization of the Bass model. Finally, I test my model by seeing how well it predicts the real-life spread of information through a Twitter network.

INTRODUCTION

**Motivation.** In the weeks following the events of 9/11, seven letters containing dangerous strains of *Bacillus anthracis* were mailed to senators and news agencies. Although the FBI never determined a sender or motive, the attacks informed the country to the possibility of bioterrorism and spurred public health agencies to plan out responses to similar, larger-scale scenarios. Anthrax is not contagious, but its dynamics require a fast dissemination of targeted public health information because newly infected individuals have a far better prognosis when they are treated quickly. To increase effectiveness of a targeted public health message, its broadcasters must understand how information spreads through a population.

**Traditional models of information diffusion.** The goal of an information diffusion model is to describe how a piece of information spreads through a given population over time. We are interested in the successive increases in the fraction of people who are aware of the information. Traditionally, information diffusion has been modeled with differential equations that describe the dynamics of a global system -- in this case, an entire population. A disadvantage of such models is that they describe only aggregate diffusion patterns, not taking into account that individuals behave in complex ways and that they function within social networks.

**A different approach: agent-based models.** Recently, bottom-up modeling in the form of agent-based simulation has gained attention. Agent-based models capture how patterns of behavior at the macro level emerge as the result of the interactions of individuals, or agents, at the micro level. Agent-based models are discrete-time simulations of the interactions in an ensemble of autonomous agents. At each iteration, each agent evaluates its situation and makes decisions according to a ruleset.

In my project, I will create an agent-based information diffusion model. I will compare my discrete-time simulation to an analytical differential equation model. Finally, I will test how well my model predicts the real-life spread of information through a Twitter network.

APPROACH

The implementation of my model is divided into two parts. First, I will create an agent-based information diffusion simulation. Then I will perform a statistical analysis based on the results of multiple executions of my simulation.

**The Bass model.** The Bass model (Bass, 1969) was originally developed by a marketer to model brand awareness, but it can also be applied more generally to the diffusion of information. The model is based on the assumption that people get their information from two sources, advertising and word of mouth.

**Formulation.** The Bass model describes the fractional change in a population’s awareness of a piece of information by:

where *F*(*t*) is the aware fraction of the population as a function of time, *p* is the advertising coefficient, and *q* is the word-of-mouth coefficient.

We can express F(t) directly as:

The Bass model can be interpreted a hazard-rate model, where P(t) = p + qF(t) is the conditional probability that a person will become aware of information at time t given that they are not yet aware.

**An agent-based Bass model.** We can formulate an agent-based model inspired by the classical Bass model. First, we discretize the problem, giving agents an opportunity to become aware of the information (given that they are not yet aware) at each time step. Then, instead of taking a deterministic time aggregate at each time step, we update each agent’s state probabilistically. Finally, we consider agents within the context of a social network: instead of allowing each agent to be influenced by the entire population, it is influenced only by its direct neighbors.

**Information diffusion through a Twitter network.** In my project, I implement an agent-based Bass model that simulates the diffusion of information through a Twitter network. (Twitter is a service which allows its users to post short messages and list which other users they read, or “follow”.) In this case, each agent corresponds to a Twitter user. A word-of-mouth transfer of information represents the exchange of information in the form of a Twitter post. The effect of advertising is any external transfer of information, that is, information obtained from a source other than Twitter. We define a Twitter user to be aware when he or she posts a message that conveys the relevant piece of information to followers.

**Network formation.** The agent-based Bass model assumes agents are arranged in some fixed, known network. Formally, the network is a directed graph with agents as its nodes. An agent’s neighbors are those who connect to it. The network structure for my simulations will be derived from real-world Twitter data. A directed edge from agent i to agent j denotes that agent j “follows” agent i on Twitter.

**The spread of information through the network.** The agent-based Bass model is a discrete-time model in which each agent has one of two states at each time step t: (1) unaware or (2) aware. At the beginning of the simulation, all agents are unaware. At each time step, an unaware agent has an opportunity to become aware. Its state changes with P, the probability that it becomes aware due to advertising or due to word of mouth. The probability that an agent becomes aware due to word of mouth increases as a function of the fraction of its neighbors who became aware in previous time steps. Once an agent becomes aware, it remains aware for the rest of the simulation.

**Probability that an agent becomes aware.** At each iteration, an unaware agent i becomes aware with probability

Pi(t) = p ∆t + q ∆t [ni(t) /mi] – (p q ∆t 2 [ni(t) /mi]),

where mi is the number of neighbors of agent i, ni(t) is the number of neighbors of agent i that became aware before time t; and p and q are parameters which indicate the effectiveness of advertising and word of mouth per unit of time, respectively. The first term is the probability that an agent becomes aware due to advertising, the second term that it becomes aware due to word of mouth, and the third term that it becomes aware due to both.

**Summary of Algorithm.**

Arbitrarily identify the N agents with the set 1, …, N. Let A denote the E×2 matrix listing all (directed) edges of the graph as ordered pairs of nodes.

INPUT: matrix A, parameters p and q.

1. Keep track of the state of the agents in a length-N bit vector initialized to all zeros.
2. At each time step, for each agent:
3. Check the bit vector to determine if the agent is already aware. If so, skip it.
4. Make the agent newly aware with probability p.
5. Look up the agent’s neighbors in A. Determine what fraction of them are aware. Make the agent newly aware with probability q times that fraction.
6. Once all agents have been processed, record the newly aware ones as aware in the bit vector.
7. Stop once all agents have become aware, or after a maximum number of iterations.

OUTPUT: complete history of the bit vector.

**Statistical analysis of results.** Since the simulation is stochastic, I plan to run it numerous times and analyse the resulting data. I wish to examine the empirical distribution of the aware fraction Φ(t) of the network at each time t. To do so, I will compute the first two moments of the distributions. Then I will plot, as a function of time, the mean ***Φ***(t) surrounded by 90 percent confidence intervals.

IMPLEMENTATION

**Hardware and software.** All code will be implemented in MATLAB (The MathWorks Inc., 2010). The simulation will run on an AMD Opteron computer with 32 cores and 256 GB of RAM. Outside software, created by Auzolle and Herrmann (2012), will be used for validation and testing. This software consists of an implementation of the agent-based Bass model written in NetLogo (Tisue and Wilensky, 2004), a programming language used to develop agent-based simulations.

**Increasing code efficiency.** The initial version of my code will run with a single thread of execution. In a later version it will be parallelized so that multiple simulations run simultaneously. Each of the runs will be logged and the complete set of data will then be analysed. Also, I believe, the network’s adjacency matrix only enters the problem as a means for looking up a node’s neighbors. I will investigate the possibility of ranging through neighbors by use of a more efficient data structure, perhaps something closer to the original input, which is essentially given as a sparse matrix.

DATABASES

The database I use will serve two purposes: it will be used to implement my model and, secondly, to test it. The networks I use to create my model will be obtained from Twitter follower data with nodes representing Twitter users and directed edges representing the flow of information. Specifically, I will use a database containing two Twitter networks, each given in the form of an E×2 matrix listing the E directed edges of a graph as ordered pairs of nodes. The graphs contain approximately 5,000 and 2,000 nodes, respectively. Additionally, I will use the database for testing my algorithm to see how well it predicts the actual spread of information through a Twitter network. I will use the above matrices along with an N-long vector (where N is the size of the network) giving the time (if ever) when a node (Twitter user) changed states from unaware to aware.

VALIDATION

To verify that I have implemented the (conceptual) agent-based Bass model correctly, I will validate my code in the following ways:

1. I will compare my results to those obtained in a simulation performed with NetLogo, software used in agent-based modeling.
2. I will perform traditional verification techniques used in agent-based modeling.
3. I will verify that my results are well approximated by the analytical differential-equation-based Bass model.

**Comparing my implementation to a NetLogo implementation.** In their paper, “Agent-Based Models of Information Diffusion”, Auzolle and Herrmann (2012) describe their implementation of an agent-based diffusion simulation written in the programming language NetLogo. The goal of my project is to replicate their implementation with one modification: I will implement the model in MATLAB. Since I will keep everything else -- including the algorithm and the databases – exactly the same, comparing against this NetLogo implementation is a reasonable way to validate my model.

**Commonly-Used Techniques to Validate Agent-Based Models.** I will perform the following three validation methods: (1) testing *Corner Cases*, (2) testing *Sampled Cases*, and (3) performing *Relative Value Testing*. Corner Cases test a model to make sure that it behaves as expected when extreme values are given as inputs. I will run my model with the parameters p=0 and q=1 to make sure that all agents remain unaware at the end of the simulation. I will then verify that when I use the parameters p=1 and q=0, all agents become aware during the first iteration. My next validation technique, testing Sampled Cases,verifies that a model produces a reasonable range of results. I will input various values of p > 0 and verify that for each such p, there exists some (bounded) number of iterations after which all agents in the network become aware. Finally, my last validation method, Relative Value Testing,verifies that the relationship between inputs and outputs is reasonable. To perform this test, I will verify that as I increase p and q, the time until all agents become aware decreases. Next, I will record, as two separate outputs, the fraction of agents that becomes aware due to advertising and the fraction that becomes aware due to word of mouth. I will use my results to verify that if I increase q while keeping p constant, the fraction of the population that becomes aware due to word of mouth increases, while the fraction that becomes aware due to advertising remains unaffected.

**Validating the agent-based Bass model with the analytical Bass model.** To validate the agent-based model analytically, we first make the simplifying assumption that all agents in the network are connected. As a result, local network structure is no longer important. Since each agent i has the whole network, including itself, as its neighbor set, ni(t)/mi is simply F(t), the aware fraction of the network. Therefore, we can rewrite

Pi(t) = p ∆t + q ∆t [ni(t)/mi] – (p q ∆t 2 [ni(t) /mi])

as P(t) = p ∆t + q ∆t F(t) – p q ∆t2 F(t) for all i.

Multiplying the probability that an agent becomes aware by the fraction of unaware agents, we obtain ∆F(t), the change in the aware fraction of the population:

∆F(t) = P(t) [1- F(t)] = [p ∆t + q ∆t F(t) – p q (∆t)2 F(t)] [1- F(t)]

Dividing through by ∆t and letting ∆t 🡪 0 recovers the analytical Bass model:

∆F/∆t = [p + q F(t)] [1- F(t)].

This result shows that in the special case of a completely connected network, the dynamics of the agent-based Bass model are well approximated by the analytical Bass model. Moreover, if they are viewed as physical quantities measured in, say, s-1, the coefficients p and q of the agent-based Bass model are identical to the p and q of the analytical model.

As pointed out by Dr. Ide, there is a subtle problem with this manipulation. In the equation on the final line, p and q are rates measured in units of probability per second and can meaningfully take on any positive values they please. In the equation on the initial line, p and q are also rates measured in units of probability per second, but they cannot meaningfully take on values greater than 1/∆t if Pi(t) is to be interpreted as a probability.

From a view that the initial equation is ground truth, there is no resolution to the problem: p∆t and q∆t are more fundamental quantities than p and q, they represent probabilities that some event will occur over a time step, and they **must** be chosen less than 1. From a view that the final equation is ground truth, the resolution is to note that we are not free to choose ∆t as we please if we want a good correspondence between the final equation and the initial: it **must** be small relative to the physical scale of the problem (1/p and 1/q).

To validate my discrete-time implementation, I will compute the total fraction Φ(t) of the network that has become aware as a function of time, recompute Φ(t) multiple times to obtain an average ***Φ***(t), and compare this average to:

the cumulative aware fraction of the network predicted by the Bass model.

TESTING

I will test my model by seeing how well it predicts the actual spread of information through a Twitter network. The two real-world cases I will use to assess my model track the spread of the information about the attack that killed Osama bin Laden and the news of Hurricane Irene, respectively, through Twitter networks. The second form of testing that I will perform is a comparison of the efficiency of my code against that of an existing NetLogo implementation.

Project Schedule and Milestones

October: Develop basic simulation code. Develop code for statistical analysis of results.

November: Validate simulation code by checking corner cases, sampled cases, and by relative testing. Validate code against analytical model.

December: Validate simulation against existing NetLogo implementation. Prepare mid-year presentation and report.

January: Investigate efficiency improvements to code. Incorporate sparse data structures.

February: Parallelize code. Test code efficiency against existing NetLogo implementation.

March: Test model against empirical Twitter data. Create visualization of model, time permitting.

April: Write final project report and prepare presentation.

DELIVERABLES

My primary deliverables will be two pieces of software: the code for my simulation and the code for my statistical analysis. I will also produce a graph showing at each time step the mean and both ends of a 90 percent confidence interval based on data collected from numerous runs of the simulation. Additionally, I will provide a detailed comparison of my code’s running time against that of the existing NetLogo implementation. Finally, I will present, side by side, the graphs of my simulation results compared with the real-world observed Twitter data.

REFERENCES

Auzolle, Ardechir and Herrmann, Jeffrey (2012). “Agent-Based Models of Information Diffusion”. Working paper, University of Maryland, College Park, Maryland.

Bass, Frank (1969). “A new product growth model for consumer durables”. *Management Science* 15 (5): p. 215–227.

Chandrasekaran, Deepa and Tellis, Gerard J. (2007). “A Critical Review of Marketing Research on Diffusion of New Products”. *Review of Marketing Research*, p. 39-80; Marshall School of Business Working Paper No. MKT 01-08.

Dodds, P.S. and Watts, D.J. (2004). “Universal behavior in a generalized model of contagion”. *Phys. Rev. Lett.* 92, 218701.

MATLAB version 7.10.0. (2010) Natick, Massachusetts: The MathWorks Inc.

Mahajan, Vijay, Muller, Eitan and Bass, Frank (1995). “Diffusion of new products: Empirical generalizations and managerial uses”. *Marketing Science* 14 (3): G79–G88.

Rand, William M. and Rust, Roland T. (2011). “Agent-Based Modeling in Marketing: Guidelines for Rigor (June 10, 2011)”. *International Journal of Research in Marketing*; Robert H. Smith School Research Paper No. RHS 06-132.

Tisue, S. and Wilensky, U. (2004). *NetLogo: A simple environment for modeling complexity.* Paper

presented at the Fifth Proceedings of the International Conference on Complex Systems, Boston.