Reverse Engineering an Agent-Based Hidden Markov Model for Complex Social Systems

Hung-Ching Chen, Mark Goldberg, Malik Magdon-Ismail, and William A. Wallace

Rensselaer Polytechnic Institute, Troy, NY 12180, USA
{chenh3, goldberg, magdon}@cs.rpi.edu, wallaw@rpi.edu

Abstract. The power of social values that helps to shape or formulate our behavior patterns is not only inevitable, but also how we have surreptitiously responded to the hidden curriculum that derives from such social values in our decision making can be just as significant. Through a machine learning approach, we are able to discover the agent dynamics that drives the evolution of the social groups in a community. By doing so, we set up the problem by introducing an agent-based hidden Markov model, in which the acts of an agent are determined by *micro-laws* with unknown parameters. To solve the problem, we develop a multistage learning process for determining the *micro-laws* of a community based on observed set of communications between actors without the semantic contents. We present the results of extensive experiments on synthetic data as well as some results on real communities, *e.g.*, Enron email and movie newsgroups.

1 Introduction

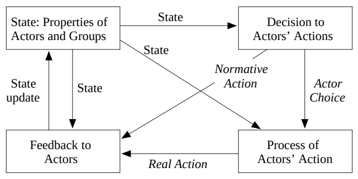
Each day, individuals from all parts of the social ramifications respond and react toward the values they perceive from the world. In the past decades, high tech has been integrated aggressively into our daily life. The rapid exchanges of communication between individuals have gone from surfing online for information to providing information, building individual Space / Blog as well as getting connected through various instant messaging communities. It is apparent that online communities have become one of the influential medium to the journey of social evolution. Yet, regardless of the impact of the online communities; the role of social value continue to play an imperative factor on the dynamics of the online communities as it has been for the offline communities rapid growth, sudden emergence or hastily dissipated due to changes of demands, needs, and values of the existing society. Therefore, it is essential to acquire ranges of more comprehensive and objective social factors that might have propelled the evolution of the society.

A social group is a collection of agents, or actors who share some common context [1]. The dynamics of the social groups are governed by the actor dynamics actors join groups and leave groups. An actors actions are governed by collective values that are direct or indirect results of the social context: personal attributes, the actions of other actors, and the social structure in the community. In summary, any reasonable model for an agent based evolving community must necessarily involve complex interactions between actors attributes and the social group structure itself. Therefore, the explosion of online communities provides an ideal pasture on which to groom and test social science theories, in particular the most natural question is: *what are the micro-laws [2]* *which govern a particular society*? Furthermore, an efficient approach to answering this question on a given community also yields a powerful tool for a sociologist.

Due to the growing popularity and interests of social network analysis (SNA), researchers have started to use different methods to help them collect and study the structure of the social network as well as analyze the ranges / factors of social dynamics, [3–5]. In this paper, We uses an agent-based hidden Markov model of a social system to identify the appropriate micro-laws of a community (appropriate parameters in the model) based on the set of observable communication edges between actors without semantic contexts. Our approach uses a multistage learning process to reduce the learning complexity and also the noise in the communication data. We identify the appropriate micro-laws by solving a mixed optimization problem because our model combines discrete and continuous parameters, and to avoid the resulting combinatorial explosion, we appropriately approximate and optimize the objective within an expectation-maximization setting. To test the quality of our approximations and the feasibility of the approach, we present the results of extensive experiments on synthetic data as well as some results on real data about Enron email and movie newsgroups.

Paper Organization. Next, we briefly give an overview of the agent-based hidden Markov model in Sec. 2. Then, we present our approach to learning the group structure and evolution from the observed communications and also learning the appropriate parameters of the model in Sec. 3. We, then, give experimental results on synthetic data and real data in Sec. 4 and conclude in Sec. 5.

2 Overview of Agent-Based Hidden Markov Model


State: Properties of Actors and Groups
State
Decision to Actors' Actions
State
Normative Action
State update
State
Actor Choice
Feedback to Actors
Real Action
Process of Actors' Action

We give a overview of the probabilistic evolving social group model ViSAGE, details of which can be found in [6], and the communication model. The foundation of ViSAGE is a Markov process, and the figure on the right hand side shows the process for each time step. There are actors, groups, the *state* of the society which is defined by properties of the actors and groups. There are three kinds of actions – *join a group*, *leave a group*, and *do nothing*. Based on the current *state* of the society, each actor decides which action she most likely wants to execute, which is known as the *Normative Action*. Nonetheless, under the influence of the present communities, actions of actors are affected. After being influenced, each actor eventually performs the *Real Action*. Depending on the feedbacks from actors' *Normative Action* and *Real Action*, properties of actors and groups are updated accordingly. We also develop a communication model to produce communication networks based on the *state* of the society; in which we only consider about the communication edges between actors without the semantics of the communications. Combining ViSAGE with the communication model, the whole process represents a hidden Markov model (HMM).

2.1 State of the Society

Many parameters govern how actors decide whether to join or leave a group, or do nothing, and also which group the actor desires to join or leave; *i.e.*, parameters such as the group memberships, how long each actor has been in a group, the ranks of the actors and the amount of resources available to an actor (in excess of what is currently being used to maintain the actor's current group memberships). Thus, the state of the society at time t can be defined as

$$S_t = \{type_i, \boldsymbol{r_i}^t, {R_E^i}^t, \{{G_l^i}^t\}_l\}_{i=1}^N, \tag{1}$$

where includes N actors, $i = 1 \cdots N$, a set of groups $\{{G_l^i}^t\}_l$ actor i joined at time t, and the properties of actors and groups describing as follow:

Type Each actor has a private set of attributes, which we refer to as its *type*. In the model, an actor's type simply controls the actor's group size preferences and her "ambition" (how quickly her rank increases in a group). There are 3 kinds of type in the model; *Leader* who prefers small groups and is the most ambitious, *Follower* preferring large groups and the least ambitious, and *Socialite* in the middle.

Rank Each actor has a rank in each group to present the actor's position in the group. As actors spend more time in a group, their position in the group changes. There is a tendency for more senior members to have a higher position than junior members. $\boldsymbol{r_i}^t$ is the set of ranks of actor i in all groups at time t.

Qualification Each actor has a qualification to represent an actor's prestige. It is determined as the average rank of the actor among all the groups, and the rank is weighted to give a stronger weight to ranks from larger groups. The qualification of an actor is used for an actor to determine which group she more likely join or leave. Similarly, each group has its $qualification$ defined as the average qualification of actors currently participating in the group. The higher a group's qualification, the more attractive it will appear to other actors looking for a group to join.

Resources We use R_E^i as the available resources for actor i , and R_E^i depends on how many resources an actor needs to maintain a membership in a group. In addition, the actors' ranks and the number of groups the actor is in influences how many resources an actor needs to maintain a membership in a group. And R_E^i also influences what kind of action an actor can complete at next time step.

2.2 State Transitions

At each time step, every actor needs to decide on leaving one group, joining one group, or remaining in the same groups. The decision depends on an actor's available resources (R_E^i). The actor will tend to join another group when she has more R_E^i; otherwise, the actor will tend to leave a group in order to lessen the cost needed. We call this action as $Normative$ action. Ideally, the actor would always choose to perform the $Normative$ action, since this creates a state of stability. However, we assume that the actors sometimes make non-rational decisions, regardless of the amount of available resources they have. An actor chooses an action she is going to perform based on a stochastic process. After an actor has chosen which action she would like to perform, she needs to decide which group to join or leave. The actor takes into account the size and qualification of the group during decision making. The group can accept or reject the actor's application based a stochastic process, which is related to the group's qualification and the actor's qualification.

The final step of the process at each time step is to update the properties of actors and groups. To update properties of actors and groups is based on all actors' $Normative$ actions and real actions and the society reward/penalty parameters θ_{reward}. The reward/penalty parameters θ_{reward} determine how to update an actor's resources, and it is summarized heuristically by $Reward\left(action, R_E^i, \theta_{action}, \theta_{reward}\right)$, where θ_{action} indicates some parameters related to actors' actions.

2.3 Communications

We have developed a social networks model to produce the communication links between actors without considering the semantics of the messages. The basic idea is that the more joined groups two actors have in common, the higher probability these two actors should communicate with each other; however, if two actors have no any joined group in common, they still have a chance to communicate with each other. A more general model also consider actors' friends; if two actors are not in a same group but they have a common friend, then there is another probability for this kind of communication. We can also consider how many levels of the friendship, *e.g.*, friend's friends in common or friend's friends' friends in common, etc.

3 Learning Process

The common learning algorithms for solving the problems in a hidden Markov model are like *forward-backard algorithm* [7], *Viterbi algorithm* [8], and *Baum-Welch algorithm* [9]. The complexities of these three algorithms are the same, $O(TM^2)$, where T is the total time steps, and M is the number of states. In our model, if there are N actors and K groups in a society, then, in each time step, there are ($2^{(NK)}/K!$) possible actors' combinations for group structure, the term $\{\{G_l^{it}\}_l\}_{i=1}^N$ in (1). If we have data for T time steps, the complexity of using the above algorithms is $\Omega(T \times \frac{2^{NK}}{K!}) \approx \Omega(T \times 2^{K(N-logK)})$, which is exponential computation time and is very time consuming. Therefore, we develop a multistage learning process. In the first stage, we find the group structures at each time step based on the communication networks, and then we discover the group evolution using the group structures. In the last stage, we learn from the group evolution to identify the appreciate parameters in ViSAGE.

3.1 Learning From Communications

The challenge with real data is that the groups structure and their evolution are not known, especially in online communities. Instead, one observes the communication dynamics. However, the communication dynamics are indicative of the group dynamics, since a pair of actors who are in many groups together are likely to communicate often. One could place one more probabilistic layer on the model linking the group structure to the communications, however, the state space for this hidden Markov model would be prohibitive. We thus opt for a simpler approach. The first step in learning is to use the communication dynamics to construct the set of groups.

In our communication model, for instance, let P_g be the probability that two actors in each same joined group would like to communicate, and P_b be the probability that two actors having no any same joined group would like to communicate. Let i, j refer to actors, and x_{ij} be a boolean value presenting the communication between actor i and j. Then the problem can be define as maximizing

$$Prob = \prod_{i,j} P_e(i,j)^{x_{ij}}(1-P_e)^{(1-x_{ij})}, \tag{2}$$

where $P_e(i,j) = \begin{cases} P_b & \text{, if } i,j \notin \text{ same group.} \\ 1-(1-P_g)^{k_{ij}} & \text{, if } i,j \text{ have } k_{ij} \text{ groups in common, and } k_{ij} > 0. \end{cases}$

We need to find P_b, P_g, and x_{ij} for all i, j, but any reasonable formulation of this problem is NP-hard, and so we need some efficient heuristic for finding the clusters in a graph that correspond to the social groups. In particular, the clusters should be allowed to overlap, as is natural for social groups. This excludes most of the traditional clustering algorithms, which partition the graph. We use the algorithms developed by Baumes et al. [4], which efficiently find overlapping communities in a communication graph. We consider time periods $\tau_1, \tau_2, \ldots, \tau_{T+1}$ and the corresponding communication graphs $G_{\tau_1}, \ldots, G_{\tau_2}$. The time periods need not be disjoint, and infact choosing them to overlap is preferable since there is considerable noise in the communications – aggregation, together with ovelap smoothens the time series of communication graphs. Given a single graph G_{τ_t}, the algorithms in [4] output a set of overlapping clusters, $\mathcal{D}_t$ (a set of groups at time step t). After knowing the group structure, we get x_{ij} for all i, j, and then we can solve the P_b and P_g maximizing (2). In this way, we can verify how good the overlapping algorithm works with the communication model.

3.2 Learning From Group Structure

From the previous stage, we have a set of group structures $\mathcal{D}_t$, $t = 1 \cdots T$. However, in order to use the learning method in next stage, one needs to construct the paths of each actor. This means we need the correspondence between groups of time step t and $t+1$, in order to determine actors' actions. Formally, we need a matching between the groups in $\mathcal{D}_t$ and $\mathcal{D}_{t+1}$ for $t = 1, \ldots, T-1$: for each group in $\mathcal{D}_t$, we must identify the corresponding group in $\mathcal{D}_{t+1}$ to which it evolved. If there are more groups in $\mathcal{D}_{t+1}$, then some new groups arose. If there are fewer groups in D_{t+1}, then some of the existing groups disappeared. In order to find this matching, we use a standard greedy algorithm.

Finding Matchings. Let $\mathcal{X} = \{X_1, \ldots, X_n\}$ and $\mathcal{Y} = \{Y_1, \ldots, Y_n\}$ be two collections of sets, and we allow some of the sets in $\mathcal{X}$ or $\mathcal{Y}$ to be empty. We use the symmetric set difference $d(x,y) = 1 - |x \cap y|/|x \cup y|$ as a measure of error between two sets. Then, we consider the complete bipartite graph on $(\mathcal{X}, \mathcal{Y})$ and would like to construct a matching of minimum total weight, where the weight on the edge (X_i, Y_j) is $d(X_i, Y_j)$. This problem can be solved in cubic time using max-flow techniques [10]. However, for our purposes, this is too slow, so we use a simple greedy heuristic. First find the best match, i.e. the pair (i^*, j^*) which minimizes $d(X_i, Y_j)$ over all pairs (i, j). This pair is removed from the sets and the process continues. An efficient implementation of this greedy approach can be done in $O(n^2 \log n)$, after $d(X_i, Y_j)$ has been computed for each pair (i, j).

3.3 Learning From Group Evolution

We first introduce some notation. The set of actors is $\mathcal{A}$; we use i, j, k to refer to actors. The data $\mathcal{D} = \{\mathcal{D}_t\}_{t=1}^{T+1}$ is the set of social groups at each time step, where each $\mathcal{D}_t$ is a *set* of groups, $\mathcal{D}_t = \{G_l^t\}_l$, $G_l^t \subseteq \mathcal{A}$; we use l, m, n to refer to groups. Collect all the parameters which specify the model as $\boldsymbol{\Theta}_M$, which includes all the parameters specific to an actor (***e.g.***, $type$) and all the global parameters in the model (***e.g.***, $\theta_{action}, \theta_{reward}, \theta_{group}$). We would like to maximize the likelihood, $\mathcal{L}(\boldsymbol{\Theta}_M) = Prob(\mathcal{D}|\boldsymbol{\Theta}_M)$. We define the path of actor i, $\mathbf{p}_i^T = (p_i(1), \ldots, p_i(T))$, as the set of actions it took over the time steps $t = 1, \ldots, T$. The actions at time t, $p_i(t)$, constitute deciding to join, leave or stay in groups, as well as which groups were left or joined. Given $\mathcal{D}$, we can construct $\mathbf{p}_i^T$ for every actor i, and conversely, given $\{\mathbf{p}_i^T\}_{i=1}^{|\mathcal{A}|}$, we can reconstruct $\mathcal{D}$. Therefore, we can alternatively maximize

$$\mathcal{L}(\boldsymbol{\Theta}_M) = Prob(\mathbf{p}_1^T, \ldots, \mathbf{p}_{|\mathcal{A}|}^T \| \boldsymbol{\Theta}_M). \tag{3}$$

It is this form of the likelihood that we manipulate. Typical ways to break up this optimization is to iteratively first improve the continuous parameters and then the combinatorial (discrete) parameters. The continuous parameters can be optimized using a gradient based approach, which involves taking derivatives of $\mathcal{L}(\boldsymbol{\Theta}_M)$. This is generally straightforward, though tedious, and we do not dwell on the technical details. The main problem we face is an algorithmic one, namely that typically, the number of actors, $|\mathcal{A}|$ is very large (thousands or tens of thousands), as is the number of time steps, T, (hundreds). From the viewpoint of actor i, we break down $\boldsymbol{\Theta}_M$ into three types of parameters: θ_i, the parameters specific to actor i, in particular its type and initial capital; $\theta_{\bar{i}}$, the parameters specific to other actors; and, θ_G, the parameters of the society, global to all the actors. The optimization style is iterative in the following sense. Fixing parameters specific to actors, one can optimize with respect to θ_G. Since this is a fixed number of parameters, this process is algorithmically feasible. We now consider optimization with respect to θ_i, fixing $\theta_{\bar{i}}, \theta_G$. This is the task which is algorithmically non-trivial, since there are $\Omega(|\mathcal{A}|)$ such parameters.

In our model, the actors at each time step take independent actions. At time t, the state of the society $\mathcal{I}_t$ can be summarized by the group structure, the actor ranks in each group and the actor surplus resources. Given $\mathcal{I}_t$, each actor acts independently at time t. Thus, we can write

$$\mathcal{L}(\boldsymbol{\Theta}_M) = Prob(\mathbf{p}_1^{T-1}, \ldots, \mathbf{p}_{|\mathcal{A}|}^{T-1} | \boldsymbol{\Theta}_M) \times \prod_{i=1}^{|\mathcal{A}|} Prob(p_i(T) | \boldsymbol{\Theta}_M, \mathcal{I}_T). \tag{4}$$

Continuing in this way, by induction,

$$\mathcal{L}(\boldsymbol{\Theta}_M) = \prod_i \prod_t Prob(p_i(t) | \boldsymbol{\Theta}_M, \mathcal{I}_t). \tag{5}$$

The actions of $\bar{i} \neq i$ depends on θ_i only through $\mathcal{I}_t$, which is a second order dependence, therefore we ignore the second term in optimizing the parameters specific to actor i, and take logarithm,

$$\theta_i^* \leftarrow \text{argmax} \sum_t \log Prob(p_i(t) | \boldsymbol{\Theta}_M, \mathcal{I}_t). \tag{6}$$

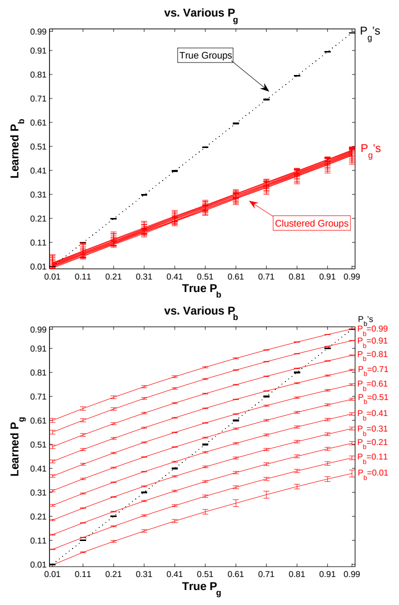
Thus the maximization over a single actor's parameters only involves that actors path and is a factor of $|\mathcal{A}|$ more efficient to compute than if we looked at all the actor paths. Therefore, the entire learning process can be summarized by maximizing over each parameter successively, where to maximize over the parameters specific to an actor, we use only that actor's path.

4 Experiments

In our model, there are a lot of parameters which can be learned, however, here we show the results about learning communication probabilities and actors' type from the synthetic data and also from real data, such as Enron email and movie newsgroups.

4.1 Results on Synthetic data

To evaluate performance, we use an instance of the model to generate synthetic data for training and testing. Since we know the values of parameters in the model, we can compare the true type with the learned type to compute the accuracy. We simulate 50, 100, 150, 200 and 250 time steps of training data (averaged over 20 data sets). Each data set was created by randomly assigning about 1/3 of the actors to each class. All others parameters except types and distributions of group size preference were held fixed.

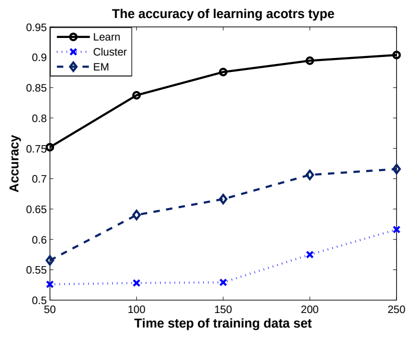


Learning Communication Probabilities. The figures at right show the results of learning communication probabilities. When the group structures are known, the algorithm can learn the P_b and P_g very well (dot lines). Meanwhile, as we apply the overlapping clustering algorithms in [4] to get the group structures, the upper figure show the learned P_b's are not influenced by different P_g's. However, different P_b's have an impact on the learned P_g's (shown at the lower figure) because some outsiders have been included in the group. The bottom line of learning communication probabilities is finding the monotonic relationship.

Learning Actors' Type. We evaluate the learning results from the following 3 different algorithms (details can be found in [6]):

– Learn: The learning algorithm described in Section 3.3 with true distributions for group size preference.
– Cluster: For each actor i, let $size_i$ be the average size of groups actor i joined. We cluster $\{size_i\}_{i=1}^{|\mathcal{A}|}$ into 3 groups using the standard 3-means algorithm. This is a simple heuristic based on the observation that leaders join small groups and followers large groups.
– EM: With unknown distributions for group size preference, we use expectation-maximization (EM) algorithm cooperating with Learn and Cluster to learn the actors' type as well as the distributions for group size preference.

The figure on the right hand side shows the accuracy (%) of Learn, Cluster and EM algorithms with different time steps of training data set, and for comparison, the accuracy of randomly assigning type is 0.33. The results tell that the accuracy for Learn algorithm is the best because it uses the true distribution of group size preference and only need to learn the actors' type. The Cluster algorithm has the worst result because it only considers the group size preference and omits the interactions with other actors. The EM algorithm learn actors' type also the distribution of group size preference. The figure tell that the EM algorithm does improve the results from the Cluster algorithm. The Cluster algorithm is only based on the average size of groups the actor joined which can be detected from *observable* group evolution data. On the other hand, the Learn and EM algorithms learn actors' *hidden* curriculum based on the interactions with other actors and the influences of the environment, which cannot be observed from the data. From the results, we also can tell when the length of the time period of training data set increases, we obtain better results from all algorithms. The reason is that more data points we can learn from, more accuracy of the results we can achieve.



4.2 Results on Real Data

Our results on real data are based on communications because it is difficult to collect data that includes the group evolution from the real world. Hence, we use the multistage learning process and algorithms in Section 3 to learn the parameters.

Movie Newsgroup. We collected the communication data from a movie newsgroup, which includes 1528 active actors. We apply both EM and Cluster algorithms on the data set, and the results are shown in the table below. Based on Cluster algorithms, the

	Learned Actors' Types		
	Leader	*Socialite*	*Follower*
Number of Actor	822	550	156
Percentage	53.8%	36.0%	10.2%

(a) Cluster algorithm

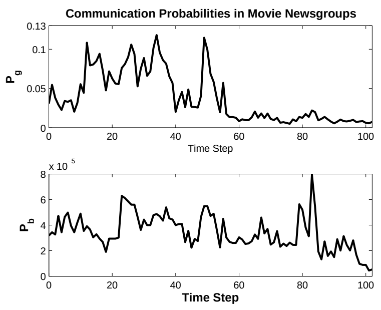
	Learned Actors' Types		
	Leader	*Socialite*	*Follower*
Number of Actor	532	368	628
Percentage	34.8%	24.1%	41.1%

(b) EM algorithm

majority of actors are leaders which *only* meant that they joined the small groups – yet, this does not represent these actors' preferences in group size. This result of Cluster algorithm in which group size that actors joined proves to match the finding that was done by hands in Butler's social analysis in the newsgroups data [11]. However, the result of EM algorithm shows that the number of *Follower* increases 30.9%, the number of *Leader* decreases 19%, and the number of *Socialite* decreases 11.9%.

According to the research data shown as above, there is a significant difference between both results: in Butler's finding (Cluster algorithms), it is easily for one to locate which size of the groups an actor joined manually but it is difficult for one to detect the actor's actual group size preferences, *i.e.*, social interactions between actors can play an influential role in the actors' decision making. By applying EM algorithm approach, one can not only consider the observable groups size that an actor joined but also the social interactions between the actors. As can be seen from above data, the approach in using EM algorithm yields similar result that shows the majority of actors would more likely to read news (*Follower*) than to post news (*Leader*) in a movie newsgroup community.

The figures at the right show the learned the communication probabilities at different time step, P_g and P_b. A comparison from both figures show that people in the same group communicate more frequently than people in the different group (P_b is much smaller than P_g). In addition, from the upper figure, we see two ranges of obvious activities - more active communications and much reduced communications.



Enron Email. By using the strategies in [12] to cleaning Enron email data set from November 13th, 1998 to June 21st, 2002 and obtain the communication network for 154 active actors, we are able to obtain learning results of EM and Cluster. Both of the EM and Cluster results are very similar (table shown below). The reason being

	Learned Actors' Types		
	Leader	*Socialite*	*Follower*
Number of Actor	28	50	76
Percentage	18.2%	32.5%	49.3%

(a) Cluster algorithm

	Learned Actors' Types		
	Leader	*Socialite*	*Follower*
Number of Actor	24	62	68
Percentage	15.6%	40.2%	44.2%

(b) EM algorithm

that in a company, an individual's preference is usually masked because the employees cannot change their "jobs" as freely as their responsibilities will change accordingly. Yet, in the movie newsgroups, actors can change groups anytime according to one's desire. Therefore, the communications within Enron email network are based upon the need of work, and employees (*Socialite* or *Follower*) cannot just join a group due to the attraction of the manager (*Leader*) of that group.

5 Conclusions

We have presented a parameterized agent-based hidden Markov model for learning actors' dynamics and the micro-laws governing the society's social group dynamics. The benefits of the multistage learning process are to extracting different information about the actor and the society dynamics, to reduce the learning noise, and to setup the checking point for evaluating the performance of algorithms, at each learning stage. Our main contributions are the application of efficient algorithms and heuristics toward learning the parameters in the specific application of modeling social groups and communications. Our results on synthetic data indicate that when the model is well specified, the learning is accurate. Since the model is sufficiently general and grounded in social science theory, any given instance of the model can be appropriate for a given society.

Therefore, under this stance, almost any general model of this form which is founded in social science theory will yield outputs that can serve as productive reference to one's decision making or stimulating triggers to new research studies.

Acknowledgments. This material is based upon work partially supported by the U.S. National Science Foundation (NSF) under Grant Nos. IIS-0621303, IIS-0522672, IIS-0324947, CNS-0323324, NSF IIS-0634875 and by the U.S. Office of Naval Research (ONR) Contract N00014-06-1-0466 and by the U.S. Department of Homeland Security (DHS) through the Center for Dynamic Data Analysis for Homeland Security administered through ONR grant number N00014-07-1-0150 to Rutgers University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the position or policy of the National Science Foundation, Office of Naval Research, Department of Homeland Security, or the U.S. Government, and no official endorsement should be inferred or implied.

References

1. Wasserman, S., Faust, K.: Social Network Analysis. Cambridge University Press (1994)
2. Goldberg, M., Horn, P., Magdon-Ismail, M., Riposo, J., Siebecker, D., Wallace, W., Yener, B.: Statistical modeling of social groups on communication networks. In: First conference of the North American Association for Computational Social and Organizational Science. (2003)
3. Backstrom, L., Huttenlocher, D., Kleinberg, J., Lan, X.: Group formation in large social networks: Membership, growth, and evolution. In: KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. (2006) 44–54
4. Baumes, J., Goldberg, M., Magdon-Ismail, M.: Efficient identification of overlapping communities. In: IEEE International Conference on Intelligence and Security Informatics (ISI). (2005) 27–36
5. Berger-Wolf, T.Y., Saia, J.: A framework for analysis of dynamic social networks. In: KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM Press (2006) 523–528
6. Chen, H.C., Goldberg, M., Magdon-Ismail, M., Wallace, W.A.: Inferring agent dynamics from social communication network. In: Joint 9th WebKDD and 1st SNA-KDD Workshop at KDD. (2007)
7. Baum, L.E., Sell, G.R.: Growth functions for transformations on manifolds. Pacific Journal of Mathematics **27**(2) (1968) 211–227
8. Viterbi, A.J.: Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. IEEE Transactions on Information Theory **IT-13**(2) (April 1967) 260–269
9. Baum, L.E., Egon, J.A.: An inequality with applications to statistical estimation for probabilistic functions of a markov process and to a model for ecology. Bulletin of the American Mathematical Soc. **73** (1967) 360–363
10. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. 2nd edn. Mcgraw-Hill, Cambridge, MA (2001)
11. Butler, B.: The dynamics of cyberspace: Examining and modeling online social structure. Technical report, Carnegie Melon University, Pittsburgh, PA (1999)
12. Zhou, Y., Goldberg, M., Magdon-Ismail, M., Wallace, A.: Strategies for cleaning organizational emails with an application to enron email dataset. In: 5th Conf. of North American Association for Computational Social and Organizational Science. (2007)