MIT Media Lab Technical Report TR#560
To appear in the *Proceedings of the International Joint Conference on Neural Networks - Special Session on Autonomous Mental Development*. July 2003.

Learning Communities: Connectivity and Dynamics of Interacting Agents

Tanzeem Choudhury[1], Brian Clarkson[2], Sumit Basu[3], and Alex Pentland[1]

[1] MIT Media Lab
20 Ames Street
Cambridge, MA 02139
USA
{tanzeem,sandy}@media.mit.edu

[2]Sony Computer Science Laboratories
Takanawa Muse Bldg.
Shinagawa-ku, Tokyo, 141-0022
Japan
clarkson@csl.sony.co.jp

[3]Microsoft Research
One Microsoft Way
Redmond, WA 98052
USA
sumitb@microsoft.com

Intelligent agents need to learn how the communication structure evolves within interacting groups and how to influence the groups overall behavior. We are developing methods to automatically and unobtrusively learn the social network structure that arises within a human group based on wearable sensors. Computational models of group interaction dynamics are derived from data gathered using wearable sensors. The questions we are exploring are: Can we tell who influences whom? Can we quantify this amount of influence? How can we modify group interactions to promote better information diffusion? The goal is real-time learning and modification of social network relationships by applying statistical machine learning techniques to data obtained from unobtrusive wearable sensors.

1 Motivation

In almost any social and work situation our decision-making is influenced by the actions of others around us. Who are the people we talk to? For how long? How often? How actively do we participate in the conversations? Answers to these questions have been used to understand the success and effectiveness of a work group or an organization as a whole. Can we identify the differences between people's interactions? Can we identify the individuals who talk to a large fraction of the group or community members? Such individuals, often referred to the *connectors*, have an important role in information diffusion [1]. Thus, learning the connection structure and nature of communication among people are important in trying to understand the following phenomena: (i) diffusion of information (ii) consensus building (iii) coalition formation etc.

The goal of our research is twofold – (i) build systems and sensors that can play the role of a mythical "familiar" that sits perched on a user's shoulder, seeing what he sees, with the opportunity to learn what he learns (ii) build an algorithmic pipeline that can take these sensors data and model the dynamics and interconnections between different players in the community. Although some may claim the link structure or social connections are often known, in many cases it has been shown that informal networks coexist with the formal structure of an institution and these spontaneous networks enhance the productivity of the formal organization [2]. We believe the best way to learn the informal networks is through observations. We then need to have a mechanism to understand the dynamics of an individual and how they interact with each other from observations.

We are interested in models the can capture the dynamics of an individual and how they interact with others in their social network. While a variety of models are potentially appropriate, such as, the HMM or the coupled HMM, these require a very large number of parameters to describe the interactions, so learning the parameters of these models is difficult and interpretation of the model parameters is impossible.

We propose as an alternative an elaboration of Influence model developed by Chalee Asavathiratham in [3]. The Influence Model parameterizes the hidden state transition probabilities by taking a convex combination of the pairwise transitions with a constant "influence" parameter. A key property of the Influence Model is a framework for understanding the global behavior by doing eigenstructure analysis of the more tractable "influence matrix". This is important in trying to understand how the behavior of each individual agent affects the global dynamics.

We develop a learning algorithm for this model and show its abilities to model chain dependencies in comparison to other standard models with synthetic data. We also show early results of applying this model to human interaction data.

In our research, we hope to lay the groundwork for being able to automatically study how different groups within social or business institutions connect, understand how

information propagates between these groups and analyze the effects of new policy or new technology on the group structure.

2 Measuring Interactions – The Sociometer

In this section we describe how we use wearable sensors to measure interactions. The first step towards reliably measuring communication is to have sensors that can capture interaction features. For example, in order to measure face-to-face interactions we need to know who is talking to whom, the frequency and duration of conversations.

We describe an experiment we completed at the MIT Media lab. A group of people at the lab agreed to wear the sociometer – the wearable sensor package that measures social interactions. It is an adaptation of the hoarder board, a wearable data acquisition board, designed by the electronic publishing and the wearable computing group at the Media lab [4, 5]. The sociometer is especially designed for the comfort of the wearer, aesthetics, and placement of sensors that are optimal in getting reliable measurements of interactions [6]. The users have the device on them for six hours a day (11AM –5PM) while they are on MIT campus. We performed the experiment in two stages – (i) the single group stage where 8 subjects from the same research group wore the sociometer for 10 days (60 hours of data per subject). (ii) the multi-group stage where 23 subjects from 4 different research group wore the sociometer for 11 days (over two full work weeks and 66 hours of data per subject).

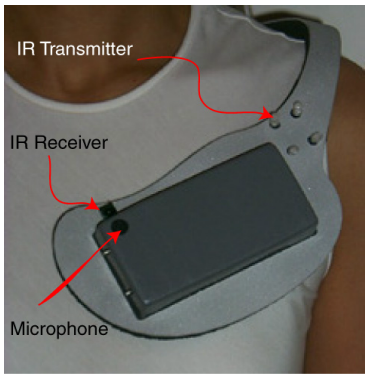

Figure 1 The shoulder mounted sociometer

The sociometer has an IR transceiver, a microphone, two accelerometers, on-board storage, and power supply. The wearable stores the data locally on a 256MB compact flash card and is powered by four AAA batteries. A set of four AAA batteries is enough to power the device for 24 hours. Everything is packaged into a shoulder mount so that it can be worn all day without any discomfort.

The sociometer stores the following information for each individual

(i) Information about people nearby (sampling rate 17Hz – sensor IR)
(ii) Speech information (8KHz - microphone)
(iii) Motion information (50Hz - accelerometer)

Other sensors (e.g. light sensors, GPS etc.) can also be added in the future using the extension board.

Using the sociometer, we can obtain the following information about people's communication – (i) who are the people an individual talks to (ii) the frequency of their communication (iii) the duration of their communication (iv) and also the flavor of their conversation. For more details about this please refer to [7, 8].

3 The Influence Model

The step after data collection is building a computational model that can use the data to infer the dynamics of the individuals and their interconnections. The learnability and interpretability of a model greatly depends on its parameterization. The "Influence Model," describes the connections between many Markov chains with a simple parameterization in terms of the "influence" each chain has on the others[3]. Asavathiratham showed how complex phenomena involving interactions between large numbers of chains could be simulated through this simplified model, such as the up/down time for power stations across the US power grid. The Influence model is a tractable framework for understanding the recurrent classes of the global system and its steady state behavior by doing eigenstructure analysis of the "influence matrix". This representation makes the analysis of global behavior possible, which otherwise would become intractable with increasing number of individuals or agents.

In Asavathiratham's description all states were observed. He did not develop a mechanism for learning the parameters of the model – he assumed that they were known apriori. Learning the model parameters from observation in an important requirement in our case. We extend his model by adding the notion of hidden states and observations. We describe algorithms for learning the parameters of the Influence model in section 4.

The graphical model for the influence model is identical to that of the generalized *N*-chain coupled HMM, but there is one very important simplification. Instead of keeping the entire $P(S_t^i \mid S_{t-1}^1, \ldots, S_{t-1}^N)$, we only keep $P(S_t^i \mid S_{t-1}^j)$ and approximate the former with:

$$P(S_t^i \mid S_{t-1}^1, \ldots, S_{t-1}^N) = \sum_j \alpha_{ij} P(S_t^i \mid S_{t-1}^j)$$

In other words, we form our probability for the next state by taking a convex combination of the pairwise

conditional probabilities for our next state given our previous state and the neighbors' previous state. As a result, we only have N QxQ tables and N α parameters per chain, resulting in a total of $NQ^2 + N^2$ transition parameters - far fewer parameters than any of the above models. The real question, of course, is whether we have retained enough modeling power to determine the interactions between the participants. Asavathiratham refers to the α's as "influences," because they are constant factors that tell us how much the state transitions of a given chain depend on a given neighbor. It is important to realize the ramifications of these factors being constant: intuitively, it means that *how much* we are influenced by a neighbor is constant, but *how* we are influenced by it depends on its state.

This simplification seems reasonable for the domain of human interactions and potentially for many other domains. Furthermore, it gives us a small set of interpretable parameters, the α values, which summarize the interactions between the chains. By estimating these parameters, we can gain an understanding of how much the chains influence each other.

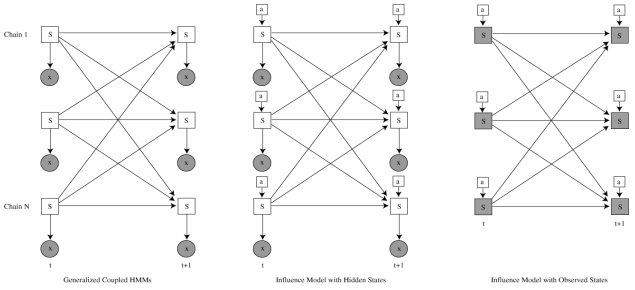

Figure 2 Graphs for (a) a generalized coupled HMM, (b) an Influence Model with hidden states, (c) an Influence Model with observed states.

3.1 The Influence Matrix

There is clearly a computational advantage of using the influence model framework in terms of the relatively small number of parameters that must be learned. Another benefit of this representation is comes also during the analysis the global dynamics of the system. It has been shown in [3] that we can make statements about the recurrent states and the steady state probabilities of the global system by analyzing the structure of the influence matrix.

The influence matrix H is defined as the Kronecker product [9] of the network influence matrix A and the local state transition matrix P.

$$H = \mathrm{A} \otimes P$$

The recurrent state of a Markov chain is a state j if $P_j(T<\infty) = 1$ otherwise if $P_j(T=\infty) >0$, j is a transient state. Here T = time of the first visit to state j.

As the dimension of G is exponential in the number of nodes (agents) in the system it is practically impossible to do computation on G. Thus being able to relate G and H and thereby do some computation on H that provides results for G will be of great practical importance. The following connections have been shown between G and H – (i) the recurrent classes of the master Markov chain can be inferred from the structure of the graph H (ii) the influence matrix H has a dominant eigenvalue at 1 and its algebraic multiplicity if equal to the number of recurrent classes of G (iii) one can also track the evolution of the steady state probability of the influence model E(s[k]) instead of the state probability of the global Markov model E(f[k]).

The advantage of doing such analysis in the domain of human interaction is in understanding how connections between people effect the overall group behavior. The connections we have with the others increase the correlation of the statuses of connected sites. How can we manipulate our links to better propagate information or stop the flow of information among the group? If we want consensus among nodes what kind of network graph will help achieve that, i.e. can we identify and modify the recurrent states of the network?

Our framework allows us to take a data driven approach to modeling social dynamics. In the following section we describe how we learn individual's states from observation data of their communication and use those to learn the network influence matrix. By doing so we will have learned all the parameters necessary to define the in Influence model. We will then be ready to understand the global properties of the system and how we may go about changing these properties.

4 Learning for the Influence Model

The problem of estimating the Influence Model from data can be stated as follows. We are given sequences of observations, $\{x_t^i\}$, from each chain, i. The goal is to estimate the amount of influence, α_{ij}, that chain j has on chain i, along with the pairwise conditional probability distributions that describe this inter-chain influence, $P(S_t^i \mid S_{t-1}^j)$. In this section we develop methods for doing this and illustrate them with synthetic data.

4.1 Expectation-Maximization Method

In Figure 2 we show the graphical model for the most general form of the Influence Model with hidden states and

continuous observations. Fitting this model to data requires us to maximize the likelihood of Influence Model over its free parameters. The likelihood function can be readily written as:

$$P(S,X)=\left(\prod_i P(S_0^i)P(x_0^i \mid S_0^i)\right)\prod_i\prod_t P(x_t^i \mid S_t^i)\sum_j \alpha_{ij}P(S_t^i \mid S_{t-1}^j)$$

One possibility for estimating the parameters of this model is Expectation-Maximization. The E-step requires us to calculate $P(S \mid X)$ which in most cases amounts to applying the Junction Tree algorithm (exact inference) or other approximate inference algorithms. We will discuss the possibilities for doing inference on this model later. The M-step is specific to this model and requires maximizing the lower bound obtained in the E-step. Examining this expression we can see that the M-step for all the parameters except the α_{ij}'s is only trivially different from the HMM[10]. However, we can readily write down the update equations for the α_{ij}'s by noticing that they are mixture weights for *N* conditional probability tables analogous to a mixture of Gaussians. The α_{ij} update equations are obtained by following the derivation of the M-step for a Gaussian mixture (i.e. introduce a hidden state to represent the "active" mixture component and then take an expectation over its sufficient statistics):

$$\alpha_{ij}^{new}=\frac{\sum_t\sum_k\sum_l P(c_t^i=j,S_t^i=k,S_{t-1}^j=l \mid X)}{\sum_t\sum_k\sum_l P(S_t^i=k,S_{t-1}^j=l \mid X)}$$

The "$c_t^i=j$" event means that at time t chain i was influenced by chain j, and the "$S_t^i=k$" event means that chain i was in state k during time t.

Unfortunately, exact inference of the Influence model is computationally intractable because of the densely connected hidden variables [11]. Variational methods or approximate inference techniques may be alternate tractable methods for learning the full model. However, in the next section we take a different approach towards tackling the intractability problem.

4.2 The Constrained Gradient Descent Method

Due to the difficulties involved in doing the inference required for E-step, we decided to simplify the estimation problem by allowing the states S_t^i to be observed for each chain (see Figure 2). In practice, we found we could obtain reasonable state sequences by fitting an HMM to each chain's observations and performing a Viterbi decoding. Then the chain transition tables can be easily estimated (by frequency counts) directly from these state sequences. Since our goal is to estimate the inter-chain influences (via the α_{ij}'s) this "clamping" of the observation and chain transition parameters help combat the overfitting problems of the full model.

We now have an unusual DBN where the observed nodes are strongly interconnected and the hidden states are not. This presents serious problems for inference because marginalizing out the observed state nodes causes all the hidden states to become fully connected across all time and all chains. Unless we apply an approximation that can successfully decouple these nodes, a maximization procedure such as EM will not be tractable. However, there is a far similar way to estimate the α_{ij} values in our observed scenario. Let us first examine how the likelihood function simplifies for the observed Influence Model:

$$P(S \mid \{\alpha_{ij}\})=\left(\prod_i P(S_0^i)\right)\prod_i\prod_t\sum_j \alpha_{ij}P(S_t^i \mid S_{t-1}^j)$$

Converting this expression to log likelihood and removing terms that are not relevant to maximization over α_{ij} yields:

$$\alpha_{ij}^*=\arg\max_{\alpha_{ij}}\left[\sum_i\sum_t \log\sum_j \alpha_{ij}P(S_t^i \mid S_{t-1}^j)\right]$$

We can further simplify this expression by keeping terms relevant to chain i:

$$\alpha_{ij}^*=\arg\max_{\alpha_{ij}}\left[\sum_t \log\sum_j \alpha_{ij}P(S_t^i \mid S_{t-1}^j)\right]$$

This *per chain* likelihood is concave in α_{ij}, which can be easily shown as follows: Let $\alpha=\begin{bmatrix}\alpha_{i0}\\ \vdots \\ \alpha_{iN}\end{bmatrix}$

$B_t^i = \begin{bmatrix} P(S_t^i \mid S_{t-1}^0) \\ \vdots \\ P(S_t^i \mid S_{t-1}^N) \end{bmatrix}$ then the *per chain* likelihood becomes: $f_i(\alpha) = \sum_t \log \left\langle \alpha, B_t^i \right\rangle$.

This is concave since for any $0 < w \le 1$ and α_0, α_1:

$$\begin{aligned} f((1-w)\alpha_0 + w\alpha_1) &= \sum_t \log \left\langle (1-w)\alpha_0 + w\alpha_1, B_t^i \right\rangle \\ &= \sum_t \log \left[(1-w) \left\langle \alpha_0, B_t^i \right\rangle + w \left\langle \alpha_1, B_t^i \right\rangle \right] \\ &\ge \sum_t (1-w) \log \left\langle \alpha_0, B_t^i \right\rangle + w \log \left\langle \alpha_1, B_t^i \right\rangle \\ &= (1-w) f(\alpha_0) + w f(\alpha_1) \end{aligned}$$

(using Jensen's inequality)

Now take the derivative w.r.t. α_{ij}:

$$\frac{\partial}{\partial \alpha_{ij}}(.) = \sum_t \frac{P(S_t^i \mid S_{t-1}^j)}{\sum_k \alpha_{ik} P(S_t^i \mid S_{t-1}^k)} = \sum_t \frac{P(S_t^i \mid S_{t-1}^j)}{\left\langle \alpha_i, B_t^i \right\rangle}$$

Here we notice the gradient and the per-chain likelihood expression above are inexpensive to compute with appropriate rearranging of the conditional probability tables to form the B_t^i vectors. This along with the facts that the per-chain likelihood is concave and the space of feasible α_{ij}'s is convex means that this optimization problem is a case of constrained gradient ascent with full 1-D search (see [12]). Furthermore, in all examples in this paper, 20 iterations were sufficient to ensure convergence.

4.3 Performance of the Learning Algorithms

To evaluate the effectiveness of our learning algorithm we show results on synthetic data. The data is generated by an Influence Model with 3 chains in lock step: one leader which was evolving randomly (i.e., flat transition tables) and 2 followers who meticulously followed the leader (i.e., an influence of 1 by chain 2 and a self-influence of 0). We sampled this model to obtain a training sequence of 50 timesteps for each chain. These state sequences were then used to train another randomly initialized Influence Model. For this learned model, the $P(S_t^i \mid S_{t-1}^j)$ were estimated by counting and the α_{ij}'s by maximizing the likelihood with gradient ascent as described above. The resulting influence graph is shown along with a typical sample sequence in Figure 3. Note how the "following" behavior is learned exactly by this model – chains 1 and 3 follow chain 2 perfectly.

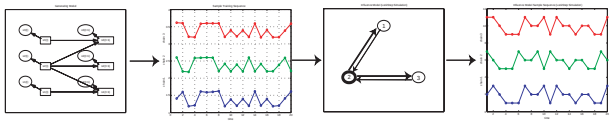

Figure 3 The evaluation pipeline for testing the Influence Model on the lockstep synthetic data: (a) the graph for the generating model at time t and t+1 (b) the training sequence (c) the learned influences (α's) – the thickness of the lines corresponds to the magnitude of the influence. Note that the strong influence of chain 2 on 1 and 3 was correctly learned. (d) Sample paths from the learned model. Note how chains 1 and 3 (the followers) follow chain 2 perfectly.

We also evaluate EM with Junction Tree on the Generalized Coupled HMM (i.e. full state transition tables instead of the mixtures of pairwise tables). Again we sample from the lock step model as before and train a randomly initialized fully connected model. In this case, the learned model performed reasonably well, but was unable to learn the "following" behavior perfectly due to the larger number of parameters it had to learn ($P(S_t^i \mid S_{t-1}^1, ..., S_{t-1}^N)$ vs. $P(S_t^i \mid S_{t-1}^j)$).

5 Results from Pilot Experiment

We used a pilot dataset of natural human interactions collected using the sociometer to learn the influences α among individuals. The details of the data collection process is described in section 2 and here we are using the data from the single-group stage. Figure 4 shows in the influence graph for the network where the strength of the link is proportional to the influence values. Qualitative comparison of our results with surveys filled out by participants show strong correlation in the influence values. However, we are in the process of doing rigorous analysis and obtaining quantitative results for our algorithms by comparing the accuracy of our result to hand-labeled ground truth data of the interactions. We plan to do analysis on the global behavior of the network, and also get results on how well our learned model can do prediction on future datasets.

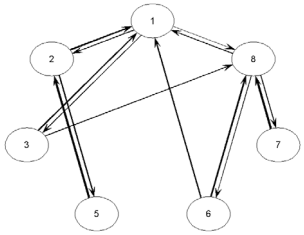

Figure 4 The influence graph for the single-group stage

6 Conclusion

In this paper, we present a method for analyzing the connectivity and dynamics of interacting groups using data gathered from wearable sensors. We develop computational models that use these sensors measurements to estimate how much influence the members of a network have on the dynamics of one another. Using these models, we can also analyze how the influence relationships affect the global network properties and how changes in an individual's connection with others can potentially change the behavior of the whole network.

References

1. Gladwell, M., *The Tipping Point: How little things make can make a big difference*. 2000, New York: Little Brown.
2. Huberman, B. and Hogg, T., *Communities of Practice: Performance and Evolution.* Computational and Mathematical Organization Theory, 1995. **1**: p. 73-95.
3. Asavathiratham, C., *The Influence Model: A Tractable Representation for the Dynamics of Networked Markov Chains*, in *Dept. of EECS*. 2000, MIT: Cambridge. p. 188.
4. Gerasimov, V., Selker, T., and Bender, W., *Sensing and Effecting Environment with Extremity Computing Devices.* Motorola Offspring, 2002. **1**(1).
5. DeVaul, R. and Weaver, J.,*MIT Wearable Computing Group*. 2002. http://www.media.mit.edu/wearables/.
6. Choudhury, T. and Clarkson, B., *Reference Design for A Social Interaction Sensing Platform*,, Media Lab Internal Design Document, 2002: Cambridge, MA.
7. Choudhury, T. and A, P. *The Sociometer: A Wearable Device for Understanding Human Networks*. In *Computer Supported Cooperative Work - Workshop on Ad hoc Communications and Collaboration in Ubiquitous Computing Environments* -. 2002. New Orleans, LA.
8. Basu, S., *Conversation Scene Analysis*, in *Dept. of Electrical Engineering and Computer science*. Doctoral. 2002, MIT. p. 1-109.
9. Schafer, R.D., *An Introduction to Nonassociative Algebras*. 1996, New York: Dover.
10. Blimes, J., A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models, 1997, University of California, Berkely, ICSI-TR-97-021
11. Lauritzen, S.L. and Spiegelhalter, D.J., *Local computations with probabilities on graphical structures and their application to expert systems.* Journal of the Royal Statistical Society, 1988. **B50**: p. 157-224.
12. Bertsekas, D.P., *Nonlinear Programming*. 1995, Belmont: Athena Scientific.