Modeling Dynamical Influence in Human Interaction Patterns

Wei Pan, Manuel Cebrian, Wen Dong, Taemie Kim
and Alex Pentland
The Media Laboratory
Massachusetts Institute of Technology

Abstract

We present a new way for modeling 1) *social influence* and 2) the well-observed property of social influence – the influence strength between individuals changes over time (e.g., friendships break and reform). We show that our unsupervised generative switching Bayesian approach can simultaneously captures the system dynamics as the outcome of both (i) the influence between individuals (each modeled as an HMM), and (ii) the changes of influence itself using only individual observations. We describe here a variational Expectation-Maximization (EM) algorithm for inference. In our experiments, we illustrate applications of predicting turn taking by analyzing a real group discussion behavior dataset and understanding flu influence patterns between US states. Results demonstrate that our approach is a strong alternative for modeling complex interacting social systems.

1 Introduction

Our model tackles the problem of analyzing and understanding *who influences whom* in a social system, such as a group discussion process, which has been an interesting question for social scientists for the last six decades [1]. Influence is also interesting in the context of leadership where the influence between one another has been recognized as a significant factor of group performance [2]. However, it remains a difficult question to define and model the concept of *influence* in a formal mathematical way.

In this paper, we handle this problem by modeling each agent in a social system as a Hidden Markov Chain with a finite set of states, and all chains interact with each other according to a family of influence configurations, each of which describes a different interaction pattern among nodes. Influence between two agents is modeled as how the current state of one agent can effect the future states of the other agents.

In the prevailing studies on social computing, quantitative efforts have focused on the *static* picture of the influence [3] [4], namely who is influencing whom in a social system when longitudinal data on human interactions is aggregated in a snapshot. However, there is extensive evidence leading us to think that influence is indeed a dynamical process[5][6]. This can also be seen from many real-world experiences:

Friendship is not static, and the person who currently possesses the most influence over you may be different after some time; In a tedious neogotiation with many parties involved, your most active opponent may change due to topic shift and strategy shift over time... Therefore, we believe that, in a social system such as a group discussion session, the influence between subjects fluctuates as well, and a better model should take the changes of influence itself into consideration.

Our approach is in essence a switching version of influence model [7], a special type of Bayesian network. We are interested in the challenge of inferring influence and learning parameters in a social system based solely on individual observations over time, i.e., without actually knowing the individual interaction patterns. We define influence between two nodes as the conditional probability between the internal states of these two nodes in consecutive time frames. We approach the problem of dynamical influence by introducing a family of different influence configurations, each of which captures a different interaction pattern among nodes. A latent trace r_t is included also to represent the index of the current active influence configuration at time t. r_t gradually switches between different influence configurations, and itself is treated as a stochastic process as well. Therefore, our model not only captures the dynamics of individual behaviors, but also the underlying latent variables tracing changes in influence. It should be noted that in our approach system dynamics and changing influence are learned simultaneously in a unified framework, and the learning algorithm is unsupervised.

2 Related Work

The Bayesian network is a tool often used in understanding patterns in social interactions [8] [7]. Earlier projects have used coupled HMM [3], and more recent projects have used the influence model [9], dynamic system trees [9] and interacting Markov chains [10]. However, none of the above approaches is capable of modeling the influence dynamics. Therefore, our approach is unique in a sense that it shows Bayesian graph is still an elegant tool for modeling changing influence.

There are quite a few related projects if we treat the underlying influence configuration as a network adjacency matrice, and consider influence dynamics as the dynamics in network structure. Researchers have been studying the class of time-varying network models: from EGRM [11] to TESLA [12], to name two. We model the dynamics of every node and edge in a network instead of feature functions as in EGRM. Our work is also significantly different from TESLA in three aspects: a) We learn from individual observations only; b) We consider changing influence, which are continuous real values, as the topological dynamics, and c) Our generative model captures the interaction of nodes and the dynamics of the interaction strength simultaneously. our work is also very close to a recent project on learning network structure [13]. However, this work is not capable of inferring dynamical network structure.

Other relevant general multi-dimensional time series approaches such as LDS [14] and the prototype model [15] are not able to recognize the network structure and weights on edges between nodes in social systems.

The classical method for Bayesian network inference is the junction tree algo-

rithm [16], of which the complexity increases exponentially to the number of chains. Here we demonstrate a variational approach with polynomial time complexity. The accuracy of this type of approach has been satisfactory in other applications [17] [18] [7].

3 Our Model

3.1 Model Description

Our approach, the Dynamical Influence Process, is a switching extension to the existing influence model [9]. It is composed of C interacting chains. In this model, similar to an HMM, each chain $c \in \{1, \ldots, C\}$ takes one of a finite number of latent states at any discrete time t: $h_t^{(c)} \in \{1, \ldots, S\}$. Corresponding to each latent state $h_t^{(c)}$, we observe $O_t^{(c)}$ which follows a conditional probability distribution Prob$(O_t^{(c)}|h_t^{(c)})$, usually known as the emission probability in HMM literature. In practice, it can either be multinomial for discrete observations or Gaussian mixture for continuous observations.

We proceed to describe the cross-chain interactions of this system. $\mathbf{M}_{i,j}$ denotes the ith row and jth column of matrix $\mathbf{M}$ in the following discussion. In this model, we have J different influence configurations described by $C \times C$ matrices $\mathbf{R}^1, ..., \mathbf{R}^J$ and only one configuration specified by another latent variable $r_t \in \{1, \ldots, J\}$ is active at time t. Each configuration matrice fully describes how chains interacts with each other, and our algorithm learns from data each configuration matrice as well as the transition between those configurations as well. Users need to specify hyper-parameter J describing the number of configurations, and we will cover the detail of selecting the values of hyper-parameters in the experiment section. The model changes its interaction configurations $\{r_t\}_{t=1,2,...}$ slowly with respect to the sampling period according to the following Markov prices:

$$r_{t+1}|r_t \sim \text{multi}(V_{r_t,1}, \cdots, V_{r_t,J}), \tag{1}$$

where $\mathbf{V}$ is constrained by another hyper-parameter $p^V, p^V > 0$. A large p^V will ensure that our model switch slowly to other influence configurations and tend to remain in the current configuration:

$$(V_{r_t,1}, \ldots, V_{r_t,J}) \sim \text{Dirichlet}(10^0, 10^0, \ldots, \underset{\uparrow}{10^{p^V}}, \ldots, 10^0).$$
$$\underset{1}{\uparrow}, \quad \underset{2}{\uparrow}, \ \ldots, \ \underset{r_t}{\uparrow}, \ \ldots, \ \underset{J}{\uparrow}$$

Given that the interaction configuration r_t is in effect, the latent state of chain c in this system at time $t+1$ is determined by another random chain $q_t^{(c)} \in \{1, \ldots, C\}$ according to a multinomial distribution described by the influence configuration $\mathbf{R}^{r_t}$.

$$q_t^{(c)}|r_t \sim \text{multi}(\mathbf{R}_{c,1}^{r_t}, ..., \mathbf{R}_{c,C}^{r_t}). \tag{2}$$

The state of chain c at $t+1$ thus is determined by:

$$h_t^{(c)}|h_{t-1}^{(1)},\ldots,h_{t-1}^{(C)},q_t^{(c)}=c'\sim \begin{cases}\text{multi}(\mathbf{E}^{(c)}_{h_t^{(c)},1},\ldots,\mathbf{E}^{(c)}_{h_t^{(c)},S}) & c'=c\\ \text{multi}(\mathbf{F}^{(c')}_{h_t^{(c')},1},\ldots,\mathbf{F}^{(c')}_{h_t^{(c')},S}) & c'\neq c\end{cases}. \tag{3}$$

We present some intuitions to the model description here: Eq. 2 and Eq. 3 define the concept of influence in our model. The intuition is that at each time t each chain c will sample $q_t^{(c)}$ from Eq. 2 to decide which chain will influence it at $t+1$. Notice that since the model is dynamical and r_t is changing, the distribution of $q_t^{(c)}$ is different at different t as we are sampling from different configuration matrices at different t. This is how changing influence is captured by switching between influence configurations in our model. If $q_t^{(c)}$ happens to be the same chain c, we will use transition matrice $\mathbf{E}^{(c)}$ and chain c's current state to determine its state at $t+1$; if $q_t^{(c)}=c'\neq c$, we will use transition matrice $\mathbf{F}^{(c')}$ and chain c''s current state to determine chain c's state at $t+1$. $\mathbf{E}^{(c)}$ and $\mathbf{F}^{(c)}$ are both $S\times S$ matrices, and they are similar to the transition matrix in HMM literature.

Given the model description, the likelihood function is

$$\mathcal{L}(O,h,q,r|\mathbf{E},\mathbf{F},\mathbf{R},\mathbf{V}) \tag{4}$$

$$\begin{aligned}=\ & \prod_{t=2}^{T}\Big\{\text{Prob}(r_t|r_{t-1})\\ & \times\prod_{c=1}^{C}\Big[\text{Prob}(O_t^{(c)}|h_t^{(c)})\\ & \times\ \text{Prob}(h_t^{(c)}|h_{t-1}^{(1,\ldots,C)},q_{t-1}^{(c)})\text{Prob}(q_t^{(c)}|r_t)\Big]\Big\}\\ & \times\prod_{c=1}^{C}\text{Prob}(O_1^{(c)}|h_1^{(c)})\text{Prob}(h_1^{(c)})\text{Prob}(r_1).\end{aligned} \tag{5}$$

Our model is illustrated in Fig. 1.

In contrast to the static influence process [4], the influence between different parts in the system is no longer static, but evolves slowly according to a Markov process. As can be seen in our experiment, this is indeed the case of a real social system.

3.2 Model Learning

Here we show key steps for our variational E-M algorithm. Definition is denoted by $\equiv$, and $\sim$ denotes the same distribution but the right side should be normalized accordingly. We refer to our supplementary materials for full details.

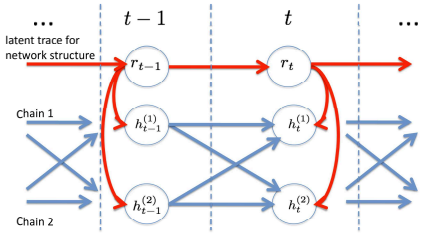


Figure 1: A graphical representation of our model when $C = 2$.

3.3 E-Step

We adopt a procedure similar to the forward-backward procedure in HMM literature. We compute the following forward parameters for $t = 1, ..., T$.:

$$\alpha_{t,c}^{r_t} \equiv \text{Prob}(h_t^{(c)}|r_t, O_{1:t}), \tag{6}$$

$$\kappa_t \equiv \text{Prob}(r_t|O_{1:t}), \tag{7}$$

where $O_{1:t}$ denotes $\{O_{t'}^{(c)}\}_{t'=1,...,t}^{c=1,...,C}$. However, exact inference is not tractable. We adopt the variational approach in [17][19], and E-M is still guaranteed to converge under variational approximation[19]. We proceed to decouple the chains by:

$$\text{Prob}(h_t^{(1)}, ..., h_t^{(C)}|O_{1:t}, r_t) \approx \prod_c Q(h_t^{(c)}|O_{1:t}, r_t), \tag{8}$$

and naturally:

$$\alpha_{t,c}^{r_t} \approx Q(h_t^{(c)}|O_{1:t}, r_t) \tag{9}$$

Using the same idea, we can compute the following backward parameters for all t:

$$\beta_{t,c}^{r_t} \equiv \text{Prob}(h_t^{(c)}|r_t, O_{t:T}), \tag{10}$$

$$\nu_t \equiv \text{Prob}(r_t|O_{t:T}). \tag{11}$$

3.4 M-step

With κ_t and ν_t, we can estimate:

$$\xi_{i,j}^t \equiv \text{Prob}(r_t = i, r_{t+1} = j|O_{1:T}) = \frac{\text{Prob}(r_t = i|O_{1:t})\text{Prob}(r_{t+1} = j|O_{t+1:T})\text{Prob}(r_{t+1}|r_t)}{\sum_{i,j}\text{Prob}(r_t = i|O_{1:t})\text{Prob}(r_{t+1} = j|O_{t+1:T})\text{Prob}(r_{t+1}|r_t)}, \tag{12}$$

$$\lambda_i^t = \text{Prob}(r_t = i|O_{1:T}) = \frac{\sum_j \xi_{i,j}^t}{\sum_i \sum_j \xi_{i,j}^t}, \tag{13}$$

and update V by:

$$\mathbf{V}_{i,j} \leftarrow \frac{\sum_t \xi_{i,j}^t + k}{\sum_t \sum_j \xi_{i,j}^t + p^V}, \tag{14}$$

where $k = p^V$ if $i = j$, 0 otherwise.

We continue to compute the joint distribution Prob$(h_t^{q_{t+1}^{(c)}}, h_{t+1}^{(c)}, q_{t+1}^{(c)}, r_{t+1}|O_{1:T})$, and update parameters $\mathbf{R}$, $\mathbf{E}$ and $\mathbf{F}$ by marginalizing this joint distribution. Please refer to the supplementary material for detail.

4 Toy Example

In this toy example, we demonstrate how our algorithm can be applied to find structural changes in network dynamics. From a dynamical influence process composed of two interacting HMM chains, we sample two binary sequences of 600 time steps. The data are shown in Table 1 (left). Each chain has two hidden states with a random transition biased to remaining in the current state. To simulate a switch in influence dynamics, we sample with configuration $\mathbf{R}^1$ (shown in Table 1) in the first 200 frames, and later on we sample with configuration $\mathbf{R}^2$. We purposely make the two configuration matrices different from each other. $\mathbf{E}^{(c)}$ and $\mathbf{F}^{(c)}$ are randomly generated for this process. All parameters are initialized randomly.

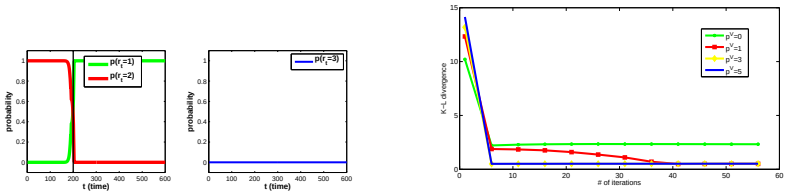
Table 1: Left: Part of the two input toy sequences for a two-chain dynamical influence process. Right: The original two influence matrices of the toy model and the same matrices learned by our algorithm with $J = 3$ and $p^V = 10^1$.

SEQ. NO.	DATA(PARTIALLY)
1	221111121212212...
2	112111212121122...

	$\mathbf{R}^1$	$\mathbf{R}^2$
True	$\begin{pmatrix} 0.90 & 0.10 \\ 0.10 & 0.90 \end{pmatrix}$	$\begin{pmatrix} 0.05 & 0.95 \\ 0.95 & 0.05 \end{pmatrix}$
Learned	$\begin{pmatrix} 0.93 & 0.07 \\ 0.10 & 0.89 \end{pmatrix}$	$\begin{pmatrix} 0.08 & 0.92 \\ 0.94 & 0.06 \end{pmatrix}$

Choosing Hyper-Parameters: We now discuss the selection of hyper-parameters J and p^V. For the number of active configuration matrices J, we here illustrate its characteristics by running the same example with $J = 3$. We show the poster distribution of r_t (calculated in Eq. 32) in Fig. 2(a). Our algorithm discovers the sudden change of influence weights accurately at $t = 200$. In addition, we discover that since the toy process only has two true configuration matrices, the posterior probability of the 3rd configuration being active is almost zero for any t. The system properties are fully captured by the other two configuration matrices during the training. The learned configuration matrices (shown in Table 1) are correctly recovered. Based on Fig. 2(a) and experiments with other values for J (which we can not show here due to the space limitation), we conclude that readers should gradually increase J until the newly added configuration matrices are no longer useful in capturing additional dynamical information from the data, by ensuring there is no constant zero posterior probability as in the right plot in Fig. 2(a).

We demonstrate convergence of the K-L Divergence between the true distributions



(a) (b)

Figure 2: (a): The posterior of r_t given observations is shown above when $J = 3$ at convergence. The middle black vertical line on the left indicates the true structural change (i.e. $\mathbf{R}^1$ is replaced by $\mathbf{R}^2$ during sampling). (b): The K-L divergence between learned parameters and the true distributions with respect to number of iterations.

of the transition probability and the learned distributions in Fig. 2(b) with different values of p^V. As can be seen in Fig. 2(b), the algorithm converges quickly within 50 iterations. We discover that when p^V is small, we may encounter over-fitting where the learned model rapidly switches between different configurations to best suit the data. Therefore, in Fig. 2(b), the divergence for $p^V = 0$ remains higher than other p^V values at convergence. In conclusion, we advise users to increase p^V gradually until the posterior of r_t does not fluctuate.

5 Experiments on Human Interaction Data

5.1 Dataset Description and Preprocessing

The dataset used in this experiment comes from a group discussion experiment in [20]. Researchers in [20] recruited 40 groups with four subjects in each group for this experiment. During the experiment, each subject was required to wear the sociometric badge on their necks for audio recording, and each group was required to perform two different group discussion tasks: a brainstorming task and a problem solving task. Each task usually lasted for 3 to 10 minutes. We kindly refer readers to the original paper for details on data collection and experiment preparations.

The groups were asked to perform these tasks in two different settings: (a) being *co-located* in the same room around a table and (b) being *distributed* in two rooms with only audio communication being available between the pairs. (The badge is deployed in both cases for audio collecting.) Later in the paper we refer to these two settings as CO and DS respectively. Since discussions are held in four-person groups, each sample for a discussion session is composed of four sequences collected by the four badges on participants' chests. The audio sequence picked up by each badge is split into one-second blocks. Variances of speech energy are calculated for each block. We then applied a hard threshold to convert them into binary sequences. In all experiments, we only use binary sequences as data input.

Table 2: The description for four different categories of all the samples.

CATEGORY	TASK DESCRIPTION
CO+PS	Four people perform a problem solving task in the same room.
CO+BS	Four people perform a brainstorming session in the same room.
DS+PS	Four people perform the same problem solving task in two rooms with Skype.
DS+BS	Four people perform the same brainstorming session in two rooms with Skype.

5.2 Predicting Turn Taking in Discussion

One important aspect of modeling interaction dynamics is the ability to predict turn taking–who will speak next in the interaction process. We here explain an application of our dynamical influence process to predict turn taking, and we show that it is possible to achieve good accuracy in prediction given only the binary audio volume variance observations, with no information from the audio content. We consider in this application that influence is the effect of someone speaking on other participants' turn taking behavior.

We separate all samples into four categories according to their original context and content, and each category is explained in Table 2. All audio data are converted into binary sequences with the procedure described in the beginning of this section. We end up with 44 samples, and each of them is composed of four sequences from all four subjects in the group. Ten occurrences of turn taking behavior from each sample are selected for prediction purposes.

Three prediction methods are implemented and evaluated: a) the dynamical influence model; b) the TESLA-based prediction (using the implementation from the authors in [21]) and c) the nearest-neighbor (NN) approach.

In the dynamical influence model, we train all parameters using the data before t to predict the turn taking behavior at t. Each chain in our model, which represents one person in the group, has two hidden states. All parameters are initialized randomly and learned by the E-M inference algorithm in this example. Since our algorithm is a generative process, we sample time t from our model, and mark the chain that changes the most toward the high-variance observational state as the turn taker.

For TESLA, we use the official implementation[21] to obtain the dynamical weights between pairs of nodes, and we pick the node which has the strongest correlation weight to other nodes at $t-1$ as the turn taker at t. We have tried different ways of using the output from TESLA for prediction and manually tuned hyper parameters. *Therefore, the results in this paper are the best among many different trials we did with TESLA.*

To predict the turn taking at time t using the nearest neighbor method, we look over all previous instances of turn taking behaviors that have the same speaker as the one in $t-1$, and predict by using the most frequent outcomes.

The accuracy for each algorithm is listed in Table 3. We also show the prediction accuracy for the half of all samples that have more complex interactions, i.e., higher

Table 3: Accuracy for different turn taking prediction methods on both the full dataset and the half of the dataset with more complex interactions. The random guess accuracy is 33%. Human accuracy is typically around 50% for similar tasks[22].

	ACCURACY ALL SAMPLES				**ACCURACY COMPLEX INTERACTION SAMPLES**			
METHODS	**DS+BS**	**DS+PS**	**CO+BS**	**CO+PS**	**DS+BS**	**DS+PS**	**CO+BS**	**CO+PS**
TESLA	0.41	0.42	0.32	0.25	0.44	0.37	0.37	0.17
NN	**0.58**	0.60	0.48	0.50	**0.47**	0.47	0.38	0.26
Ours(J=1)	0.45	**0.67**	**0.75**	**0.63**	0.45	0.56	**0.77**	0.62
Ours(J=2)	0.46	0.58	0.65	0.34	**0.47**	0.58	0.67	0.46
Ours(J=3)	0.50	0.60	0.55	0.48	**0.47**	**0.73**	0.65	**0.65**

entropy. For our dynamical influence based approach, we list error rates for $J = 1, 2$ and 3. Except DS+BS, We notice that our algorithm outperforms others in all categories with different J. This performance is quite good considering that we are using only volume and that a human can only predict at around 50% accuracy for similar tasks[22].

More importantly, our model seems to perform much better than the competing methods for more complex interactions. For simple interactions, it seems that $J = 1$ or even NN perform the best due to the fact that there is little shift in influence structure during the discussion. However, when handling complex interaction processes, the introduction of a switching influence dynamics dramatically improves the performance as shown in Table 3. Our results strongly suggest that the dynamical influence assumption in our model is reasonable and necessary in modeling complex group dynamics, and in one case it can improve prediction accuracy to above 60% for PS tasks. However, in simple cases, the model achieves the highest performance only when $J = 1$, i.e. the influence is static, and a higher J will only lead to overfitting.

5.2.1 Discussion on Other Time-varying Network Algorithms

TESLA fails in most prediction tasks, and we believe that the formulation of TESLA doesn't suit our applications, because it only captures the correlations between observations from dyad nodes at each time step and models the process as a time-varying network. We speculate that similar discriminative approaches adopted in [11] may not be suitable for our applications either. *It can be concluded that while time-varying models provide valuable information on network structural changes, the dynamics of influence and interaction in social systems are different from dynamics of network structures.* On the other hand, we consider a dynamical network as a full graph with changing influence instead of a binary re-wiring graph, and influence is modeled as time-varying Markov property between hidden states from two frames. Our generative model becomes a new effective and competitive perspective for such applications.

6 Modeling Flu Epidemics as Influence Dynamics

In this experiment, we demonstrate that flu spreading can be modeled as the outcome of interaction between people living in different regions in US.

We apply our algorithm to the weekly US flu activity data from Google Flu Trend [23]. All 50 states in US are divided into ten regions by their geo-location. We model each region as one chain in our dynamical influence model. As the data is continuous, six hidden states are used for each chain, and we choose Gaussian distribution as the emission distribution. We set the emission probability distribution to be the same for each chain and keep it fixed during the learning process so that the same hidden states for each chain share the same semantic meaning. The first 290 weeks (from 2003 to early 2009) are used for training and the remaining for testing. We set $J = 3, p^V = 10^{-1}$ for optimized performance.

We show the posterior for r_t in Fig. 3 after our model converges with the training data. While there are many small peaks suggesting changes in influence, the probability changes dramatically around Christmas, which suggests that the influence patterns among these ten regions are very different during the holiday season. Our algorithm actually reveals Christmas traveling by looking at only flu data in a fully unsupervised manner.

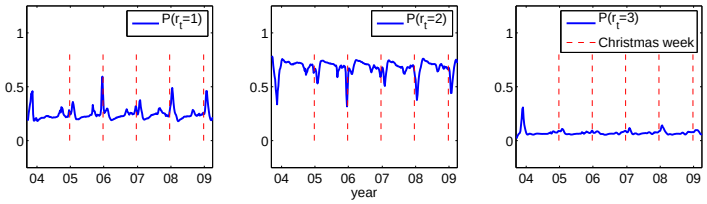


Figure 3: The inferred posterior for r_t given all observations after convergence is shown here. While there are many small peaks indicating changes in influence, the largest peaks occur at Christmas holiday seasons, which implies that influence between states are very different in Christmas holidays comparing with other dates. This matches the common sense that travelling patterns are different around holiday seasons. In our experiment, we find that three configuration matrices are good enough to capture the flu dynamics.

We proceed to study the long-term prediction abilities of different models. The prediction error results are shown in Table 4. We compare with different approaches in predicting flu data: the Linear Dynamical System(LDS) (based on a modern implementation [24]) , our approach with $J = 3$, and ours with $J = 1$ (the influence model). As a benchmark, we also show the results of predicting flu using the average value of the historical flu data of the same date. For our model, when predicting at t, we use a configuration matrix $\mathbf{R} = a_t^1\mathbf{R}_1 +, ..., +a_t^J\mathbf{R}_J$, where a_t^j is Prob$(r_{t'} = j|$training data$)$, and t' denotes the same date of t in the previous year. Prediction is done by sampling our generative model with trained parameters and estimating expectations as the prediction.

The flu data in 2009 is very different from previous years. However, as we anticipated, people still maintain similar traveling routines, and our model captures the underlying influence dynamics rather than the raw prototype of flu seasonal patterns. Therefore, our model prediction is better than using the historical data directly as illustrated in Fig. 4.

Our approach performs better when $J = 3$, which verifies our assumption that the dynamics of the influence patterns are very important to prediction accuracy. As for LDS, the performance is unsatisfactory. We believe that the complicated nature of the system is unlikely to be captured by a linear approach such as LDS.

There are several high performance flu prediction models that offer comparable performance. However, our work reveals one novel perspective that the dynamics of flu spreading can be modeled partially by the influence between states and the changes of the influence itself. In particular, the influence dynamics learned by our model correlate well with the holiday seasons, which strongly suggests that the influence dynamics learned in our model are indeed capturing real interaction dynamics among US states.

We present another application of detecting influence structural change with the same group discussion dataset in the supplementary material.

Table 4: This table shows accumulated square error for different algorithms in predicting weekly flu epidemic from April, 2009 to August, 2009.

METHOD	ERROR (RESCALED)
Historical Data	3.44
LDS	5.41
Our Approach (J=1)	1.38
Our Approach (J=3)	**1.18**

7 Conclusions

We have developed an unsupervised generative model that captures the system dynamics as the outcome of both the influence between individuals and the dynamics of influence. Our model directly tackles the important sociological question of analyzing who influence whom in social systems. In our model, C HMM chains interact with each other according to a family of influence matrices describing different interaction patterns, and switch between them over time. Our model only uses individual observations as input to infer influence and changes in influence. A fast variational inference scheme is also developed to handle large datasets. We have demonstrated the performance of our model in applications of detecting structural change, predicting turn taking and understanding epidemic dynamics. Our model provides a new and unique perspective for influence dynamics using a Bayesian network approach, and we believe that this model will be an essential tool for analyzing many social and societal systems ranging from group discussion to flu dynamics.

In particular, our model provides a formal definition and a rich understanding for patterns of influence in social groups, and in future works we can leverage such infor-

mation to provide benefit to the groups themselves. As group dynamics have a strong relationship with the group's performance [25], there have been efforts to measure the communication patterns and provide feedback to the group in real-time [20, 26]. Our approach can enable such feedback specific to influence structures which may lead to more effective leadership, stronger ties among group members and higher performance of teams.

Our immediate next step is also to apply our approach to larger and longer individual human behavioral datasets which we are currently collecting. This would fully leverage our fast inference scheme, because the current interest in computational social science is in the scale of thousands of individuals [27]. This would also allow us to leverage the power of machine learning to reach a better understanding of changes in human interaction patterns, and make significant quantitative contributions to social science.

References

[1] E. Katz and P.F. Lazarsfeld. *Personal influence*. Free Pr., 1955.

[2] G.F.. Farris and F. Lim. Effect of performance on leadership, cohesiveness, influence, satisfaction and subsequent performance. *Journal of Applied Psychology*, 53(6):490–497, 1969.

[3] M. Brand, N. Oliver, and A. Pentland. Coupled hidden Markov models for complex action recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 994–999, 1997.

[4] W. Dong and A. Pentland. Modeling influence between experts. *Lecture Notes in Computer Science*, 4451:170, 2007.

[5] S.A. Ansari, V.S. Springthorpe, and S.A. Sattar. Survival and vehicular spread of human rotaviruses: possible relation to seasonality of outbreaks. *Reviews of infectious diseases*, 13(3):448–461, 1991.

[6] J.P. Onnela, J. Saramaki, J. Hyvonen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A.L. Barabási. Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences*, 104(18):7332, 2007.

[7] W. Dong, B. Lepri, A. Cappelletti, A.S. Pentland, F. Pianesi, and M. Zancanaro. Using the influence model to recognize functional roles in meetings. In *Proceedings of the 9th international conference on Multimodal interfaces*, pages 271–278. ACM, 2007.

[8] T. Choudhury and S. Basu. Modeling conversational dynamics as a mixed memory markov process. In *Proc. of Intl. Conference on Neural Information and Processing Systems (NIPS)*. Citeseer, 2004.

[9] S. Basu, T. Choudhury, B. Clarkson, A. Pentland, et al. Learning human interactions with the influence model. *MIT Media Laboratory Technical Note*, 2001.

[10] D. Zhang, D. Gatica-Perez, S. Bengio, and D. Roy. Learning Influence Among Interacting Markov Chains. 2005.

[11] F. Guo, S. Hanneke, W. Fu, and E.P. Xing. Recovering temporally rewiring networks: A model-based approach. In *Proceedings of the 24th international conference on Machine learning*, page 328. ACM, 2007.

[12] A. Ahmed and E.P. Xing. Recovering time-varying networks of dependencies in social and biological studies. *Proceedings of the National Academy of Sciences*, 106(29):11878, 2009.

[13] M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1019–1028. ACM, 2010.

[14] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Nonparametric Bayesian learning of switching linear dynamical systems. In *Neural Information Processing Systems 21*. MIT Press, 2009.

[15] W. Pan and L. Torresani. Unsupervised hierarchical modeling of locomotion styles. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM New York, NY, USA, 2009.

[16] M.I. Jordan. *Learning in graphical models*. Kluwer Academic Publishers, 1998.

[17] R.J. Weiss and D.P.W. Ellis. A variational EM algorithm for learning eigenvoice parameters in mixed signals. In *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing-Volume 00*, pages 113–116. IEEE Computer Society, 2009.

[18] Z. Ghahramani and M.I. Jordan. Factorial hidden Markov models. *Machine learning*, 29(2):245–273, 1997.

[19] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[20] T. Kim, A. Chang, L. Holland, and A.S. Pentland. Meeting mediator: enhancing group collaborationusing sociometric feedback. In *Proceedings of the ACM 2008 conference on Computer supported cooperative work*, pages 457–466. ACM, 2008.

[21] A. Ahmed and E.P. Xing. Recovering time-varying networks of dependencies in social and biological studies. *Proceedings of the National Academy of Sciences*, 106(29):11878, 2009.

[22] D. Schlangen. From reaction to prediction: Experiments with computational models of turn-taking. In *Ninth International Conference on Spoken Language Processing*. Citeseer, 2006.

[23] J. Ginsberg, M.H. Mohebbi, R.S. Patel, L. Brammer, M.S. Smolinski, and L. Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–1014, 2008.

[24] S.M. Siddiqi, B. Boots, and G.J. Gordon. A constraint generation approach to learning stable linear dynamical systems. *Advances in Neural Information Processing Systems*, 2007.

[25] M. Shaw. *Group Dynamics: The Psychology of Small Group Behavior*. McGraw-Hill, 1976.

[26] J.M. DiMicco, KJ Hollenbach, A. Pandolfo, and W. Bender. The Impact of Increased Awareness while Face-to-Face. *Human-Computer Interaction*, 22(1), 2007.

[27] M.C. González, C.A. Hidalgo, and A.L. Barabási. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

[28] C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines, 2001.

8 Appendix A: Model Learning

We here show detail steps for our variational E-M algorithm. Definition is denoted by $\equiv$, and $\sim$ denotes the same distribution but the right side should be normalized accordingly.

8.1 E-Step

We adopt a procedure similar to the forward-backward procedure in HMM literature. We compute the following forward parameters for $t = 1, ..., T$.:

$$\alpha_{t,c}^{r_t} \equiv \text{Prob}(h_t^{(c)}|r_t, O_{1:t}), \tag{15}$$

$$\kappa_t \equiv \text{Prob}(r_t|O_{1:t}), \tag{16}$$

where $O_{1:t}$ denotes $\{O_{t'}^{(c)}\}_{t'=1,...,t}^{c=1,...,C}$. However, exact inference is not intractable. We apply the variational approach in [17] [19]. The variational E-M process is still guaranteed to converge because of the lower bound property of the variational method [19]. We decouple the chains by:

$$\text{Prob}(h_t^{(1)}, ..., h_t^{(C)}|O_{1:t}, r_t) \approx \prod_c Q(h_t^{(c)}|O_{1:t}, r_t), \tag{17}$$

and naturally:

$$\alpha_{t,c}^{r_t} \approx Q(h_t^{(c)}|O_{1:t}, r_t) \tag{18}$$

We define:

$$\alpha_{t,c} \equiv \text{Prob}(h_t^{(c)}|O_{1:t}) = \sum_{r_t} \kappa_t \alpha_{t,c}^{r_t}, \tag{19}$$

and

$$\widehat{\alpha^{r_t}}_{t-1,c} \equiv \text{Prob}(h_{t-1}^{(c)}|r_t, O_{1:t-1}) = \sum_{r_{t-1}} \widehat{\kappa^{r_t}}_{t-1} \alpha_{t-1,c}^{r_{t-1}}, \tag{20}$$

where:

$$\widehat{\kappa^{r_t}}_{t-1} \quad = \quad \text{Prob}(r_{t-1}|O_{1:t-1}, r_t). \tag{21}$$

We define $Q(h^{(c)})$ to be in the form of $\frac{\text{Prob}(h_t^{(c)}|O_{1:t-1},r_t)s_t^{(c)}}{\sum_{h_t^{(c)}} \text{Prob}(h_t^{(c)}|O_{1:t-1},r_t)s_t^{(c)}}$, which captures both the evidence from previous states($\text{Prob}(h_t^{(c)}|O_{1:t-1}, r_t)$) and the evidence($s_t^{(c)}$) from observations. We then have:

$$\begin{aligned} &Q(h_t^{(1,...,C)}|O_{1:t}, r_t) \\ &\equiv \prod_c \frac{\underbrace{\left(\sum_{h_{t-1}^{(c)}} \mathbf{R}_{c,c}^{r_t} \mathbf{E}_{h_{t-1}^{(c)}, h_t^{(c)}}^{(c)} \widehat{\alpha^{r_t}}_{t-1,c} + \sum_{c',c' \neq c} \sum_{h_{t-1}^{(c')}} \mathbf{R}_{c,c'}^{r_t} \mathbf{F}_{h_{t-1}^{(c')}, h_t^{(c)}}^{(c')} \widehat{\alpha^{r_t}}_{t-1,c'}\right)}_{\Psi^{(c)}} \times s_t^{(c)}}{\sum_{h_t^{(c)}} \left(\sum_{h_{t-1}^{(c)}} \mathbf{R}_{c,c}^{r_t} \mathbf{E}_{h_{t-1}^{(c)}, h_t^{(c)}}^{(c)} \widehat{\alpha^{r_t}}_{t-1,c} + \sum_{c',c' \neq c} \sum_{h_{t-1}^{(c')}} \mathbf{R}_{c,c'}^{r_t} \mathbf{F}_{h_{t-1}^{(c')}, h_t^{(c)}}^{(c')} \widehat{\alpha^{r_t}}_{t-1,c'}\right) \times s_t^{(c)}} \end{aligned} \tag{22}$$

where $\Psi^{(c)}$ is actually $\text{Prob}(h_t^{(c)}|O_{1:t-1}, r_t)$. We also have:

$$\text{Prob}(h_t^{(1,\ldots,C)}, O_t|O_{1:t-1}, r_t) = \prod_c \underbrace{\left(\sum_{h_{t-1}^{(c)}} \mathbf{R}_{c,c}^{r_t} \mathbf{E}^{(c)}_{h_{t-1}^{(c)}, h_t^{(c)}} \widehat{\alpha^{r_t}}_{t-1,c} + \sum_{c', c' \neq c} \sum_{h_{t-1}^{(c')}} \mathbf{R}_{c,c'}^{r_t} \mathbf{F}^{(c')}_{h_{t-1}^{(c')}, h_t^{(c)}} \widehat{\alpha^{r_t}}_{t-1,c'} \right)}_{\Psi^{(c)}} \text{Prob}(O_t^{(c)}|h_t^{(c)}), \tag{23}$$

and

$$\text{Prob}(h_t^{(1,\ldots,C)}|O_{1:t-1}, r_t) = \frac{\text{Prob}(h_t^{(1,\ldots,C)}, O_t|O_{1:t-1}, r_t)}{\text{Prob}(O_t|O_{1:t-1}, r_t)}. \tag{24}$$

We continue to minimize the K-L divergence between $\text{Prob}(h_t^{(1,\ldots,C)}|O_{1:t}, r_t)$ and $Q(h_t^{(1,\ldots,C)}|O_{1:t}, r_t)$, that is:

$$\begin{aligned}
\arg\min_{s_t^{(c)}} \mathbb{D} &\equiv \mathbb{E}_Q \left(\log Q(h_t^{(1,\ldots,C)}|O_{1:t}, r_t) \right) - \mathbb{E}_Q \left(\text{Prob}(h_t^{(1,\ldots,C)}|O_{1:t}, r_t) \right) \\
&= \mathbb{E}_Q \left(\sum_c \log \Psi^{(c)} + \sum_c \log s_t^{(c)} - \sum_c \log(\sum_{h_t^{(c)}} \Psi^{(c)} s_t^{(c)}) \right) \\
&\quad - \mathbb{E}_Q \left(\sum_c \log \Psi^{(c)} + \sum_c \log \text{Prob}(O^{(t)}|h_t^{(c)}) \right) + \underbrace{\text{Prob}(O_t|O_{1:t-1}, r_t)}_{\text{unrelated to } s_t^{(c)}}
\end{aligned} \tag{25}$$

By taking the derivative we have:

$$\begin{aligned}
\frac{\partial \mathbb{D}}{\partial s_t^{(c)}} &= \sum_{h_t^{(c)}} \frac{\partial \widehat{\alpha^{r_t}}_{t,c}}{\partial s_t^{(c)}} \left(s_t^{(c)} - \text{Prob}(O_t^{(c)}|h_t^{(c)}) \right) = 0 \\
&\Rightarrow s_t^{(c)} = \text{Prob}(O_t^{(c)}|h_t^{(c)})
\end{aligned} \tag{26}$$

We then compute κ_t using Bayes' rule:

$$\kappa_t \sim \text{Prob}(O_t|r_t, O_{1:t-1}) \text{Prob}(r_t|O_{1:t-1}). \tag{27}$$

where $\text{Prob}(O_t|r_t, O_{1:t-1})$ can be evaluated using the previous approximation results. The prior part of Eq. 27 can be evaluated using $\mathbf{V}$ and κ_{t-1}.

Using the same idea, we can compute the following backward parameters for all t:

$$\beta_{t,c}^{r_t} \equiv \text{Prob}(h_t^{(c)}|r_t, O_{t:T}), \tag{28}$$

$$\nu_t \equiv \text{Prob}(r_t|O_{t:T}), \tag{29}$$

$$\beta_{t,c} \equiv \text{Prob}(h_t^{(c)}|O_{t:T}) = \sum_{r_t} \nu_t \beta_{t,c}^{r_t}. \tag{30}$$

8.2 M-step

With κ_t and ν_t, we can estimate:

$$\xi^t_{i,j} \equiv \text{Prob}(r_t = i, r_{t+1} = j|O_{1:T}) = \frac{\text{Prob}(r_t = i|O_{1:t})\text{Prob}(r_{t+1} = j|O_{t+1:T})\text{Prob}(r_{t+1}|r_t)}{\sum_{i,j}\text{Prob}(r_t = i|O_{1:t})\text{Prob}(r_{t+1} = j|O_{t+1:T})\text{Prob}(r_{t+1}|r_t)}, \quad (31)$$

and

$$\lambda^t_i \quad = \quad \text{Prob}(r_t = i|O_{1:T}) = \frac{\sum_j \xi^t_{i,j}}{\sum_i \sum_j \xi^t_{i,j}}. \quad (32)$$

We then update V by:

$$\mathbf{V}_{i,j} \leftarrow \frac{\sum_t \xi^t_{i,j} + k}{\sum_t \sum_j \xi^t_{i,j} + p^V}, \quad (33)$$

where $k = p^V$ if $i = j$, 0 otherwise.

We compute the following joint distribution.

$$\text{Prob}(h_t^{q^{(c)}_{t+1}}, h^{(c)}_{t+1}, q^{(c)}_{t+1}, r_{t+1}|O_{1:T}) = \quad (34)$$
$$\begin{cases} \frac{1}{Z} T^{(c)}_{h^{(c)}_t, h^{(c)}_{t+1}} \times \widehat{\alpha^{r_t}}_{t,c} \beta^{r_t}_{t+1,c} \lambda^t \text{Prob}(q^{(c)}_{t+1}|r_{t+1}), \\ \qquad\qquad \text{if } q^{(c)}_{t+1} = c, \\ \frac{1}{Z} F^{q^{(c)}_{t+1}}_{h^{q^{(c)}_{t+1}}_t, h^{(c)}_{t+1}} \times \widehat{\alpha^{r_t}}_{t,q^{(c)}_{t+1}} \beta^{r_t}_{t+1,c} \lambda^t \text{Prob}(q^{(c)}_{t+1}|r_{t+1}), \\ \qquad\qquad \text{if } q^{(c)}_{t+1} \neq c. \end{cases}$$

Z denotes the normalization factor that can be calculated easily by summing all possible values for each variable. This is fast to compute since the joint distribution is made of only four variables. By marginalizing Eq. 34, we can update parameters $\mathbf{R}$, $\mathbf{E}$ and $\mathbf{F}$:

$$\mathbf{R}^j_{c_1,c_2} \leftarrow \frac{\sum_t \text{Prob}(q^{(c_1)}_t = c_2, r_t = j|O_{1:T})}{\sum_t \sum_c \text{Prob}(q^{(c_1)}_t = c, r_t = j|O_{1:T})}, \quad (35)$$

$$\mathbf{E}^{(c)}_{s_i,s_j} \leftarrow \frac{\sum_t \text{Prob}(h^{(c)}_t = s_i, h^{(c)}_{t+1} = s_j, q^{(c)}_t = c|O_{1:T})}{\sum_t \sum_s \text{Prob}(h^{(c)}_t = s_i, h^{(c)}_{t+1} = s, q^{(c)}_t = c|O_{1:T})}, \quad (36)$$

and

$$\mathbf{F}^{(c)}_{s_i,s_j} \leftarrow \frac{\sum_t \sum_{c'} \text{Prob}(h^{(c)}_t = s_i, h^{(c')}_{t+1} = s_j, q^{(c')}_{t+1} = c|O_{1:T})}{\sum_t \sum_{c'} \sum_s \text{Prob}(h^{(c)}_t = s_i, h^{(c')}_{t+1} = s, q^{(c')}_{t+1} = c|O_{1:T})}. \quad (37)$$

9 Appendix B: Detecting Structural Changes in the Discussion Dynamics

We here provide an additional example for detecting structural changes using the same dataset described in Section 5 of our paper.

One important feature of this model is its ability to capture changes in influence dynamics given only observed time series for each node. In this section, we will demonstrate the performance of our model in detecting changes with the group discussion dataset.

In our discussion, *a sample* refers to the set of four sequences collected by the four badges in deployed in a group discussion session. We adopt the following evaluation procedure: One mixed binary audio sample for each four-person group is generated by concatenating the co-located discussion session sample and the the distributed discussion session sample of the same group. It is known that [20] the interaction pattern in a distributed discussion session is often different from a co-located discussion session. Therefore, we are able to create ground truth about changes of influence patterns by switching from a distributed discussion sample to a co-located discussion sample manually. It should be noted that we only use binary sequences by thresholding the volume variance. Thus, we have eliminated all information in the audio content. Two samples from each group are included in our final evaluation set: a) the original sample of the co-located discussion session (CO) and b) the mixed sample as described above (CO+DS). We end up with a total of 28 groups and 56 samples in the final set. Lengths of each sample vary from 100 seconds to 500 seconds.

We apply our model on both samples for each group. The emission probability in our model is used to tolerate possible error due to hard thresholding and possible noise. We choose $J = 2$, and p^V is optimized for best performance. The posterior of r^t for the two samples from each group is stored as the output of the algorithm.

We continue to develop simple heuristics for distinguishing DS+CO from CO by looking at the difference of the expected influence matrix ($\sum_j \lambda_j^t R_j$) at $t = 1$ and $t = 0.8T$ for each sample, and the one with larger difference is labeled as the CO+DS sequence. Given the pair of samples for each group, we test the labeling accuracy based on the output of our model. For comparison, we also implement two other techniques: a) classification based on one single feature, the turn taking rate, and b) S.V.M.-based classification (using implementation in [28]). It is well recognized that the turn taking rate is an important indicator for group dynamics. We compute the two turn taking rates for each pair of samples and compare them to determine sample labels. For S.V.M., we compute the turn taking rate and the speaking durations for each group member as the feature vector for each sample. Its performance is obtained via a four-fold cross validation. It should be emphasized that the S.V.M. classification task is different from the other two, and it is naturally more challenging: all samples are mixed together before fed to S.V.M. rather than being fed to other two algorithms in a pairwise manner.

We must point out that the ground truth in our evaluation may not be accurate: There is no guarantee in the dataset that a group of people behave and interact with each other differently when they are performing discussions using remote communication tools rather than being in the same room.

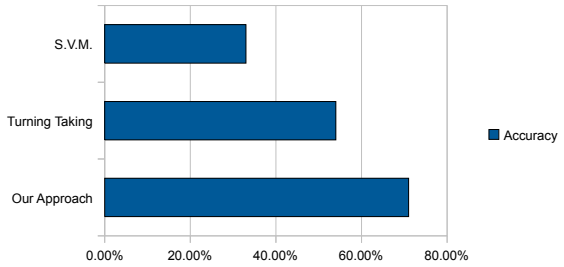


Figure 4: The accuracy rates for classifying CO+DS samples from CO samples are shown above. Our algorithm performs significantly better than the other two methods, which are based on simple statistical features.

We illustrate the accuracy rates in Fig. 4. As we expected, our algorithm reaches 71% accuracy and outperforms the other two methods. We argue that the influence dynamic is an intrinsic property of the group, which can not be fully revealed using simple statistical analysis on observable features. To investigate and visualize the dynamical characteristics of human interaction patterns, a more sophisticated model, such as our dynamical influence process, must be deployed to reveal the subtle differences in influence dynamics.

In addition, we claim that our model is capable of modeling, quantifying and tracking occurrences of such shifts in face-to-face dynamics accurately. Our model fits its parameters to best suit switches between different influence patterns, and the parameters will be helpful for sociologists to objectively investigate the micro relationship in a group discussion session. Information discovered by our algorithm will also be useful in applications such as understanding possible interventions in human interactions[20][26].