Inferring Dynamic Dependency with Applications to Link Analysis

Michael R. Siracusa
Massachusetts Institute of Technology
77 Massachusetts Ave.
Cambridge, MA 02139

John W. Fisher III
Massachusetts Institute of Technology
77 Massachusetts Ave.
Cambridge, MA 02139

***Abstract*— Statistical approaches to modeling dynamics and clustering data are well studied research areas. This paper considers a special class of such problems in which one is presented with multiple data streams and wishes to infer their interaction as it evolves over time. This problem is viewed as one of inference on a class of models in which interaction is described by changing dependency structures, *i.e.* the presence or absence of edges in a graphical model, but for which the full set of parameters are not available. The application domain of dynamic link analysis as applied to tracked object behavior is explored. An approximate inference method is presented along with empirical results demonstrating its performance.**

I. Introduction

Consider a scene in which multiple objects such as people or vehicles are moving around in an environment. Given noisy measurements of their position over time, a question one may ask is, which, if any, of these objects are interacting at each point in time. The answer to this type of question is useful for social interaction analysis, anomalous event detection and automatic scene summarization. For example, Heider and Simmel [1] showed that when describing a cartoon of simple shapes moving on a screen, one will tend to described their behavior with terms like “chasing”, “following” or being independent of one another. These semantic labels describe the dependency between objects and allow humans to provide a compact description of what they are observing.

We view this problem as a specific example of a general class of problems which we refer to as dynamic dependency tests. A dynamic dependency test answers the following question: given multiple data streams, how does their interaction evolve over time? Here, interaction is defined in terms of changing graphical structures, *i.e.*, the presence or absence of edges in a graphical model. For example, in the object interaction case, “X is following Y” implies a casual dependency between Y and X.

We cast a dynamic dependency test as a problem of inference on a special class of probabilistic models in which a latent state variable indexes a discrete set of possible dependency structures on measurements. We refer to this class of models as dynamic dependence models and introduce a specific implementation via a hidden factorization Markov model (HFactMM). This model allows us to take advantage of both structural and parametric changes associated with changes in the state of interaction of a set of objects. We further show how inference and learning on an HFactMM can be used in an approximate inference procedure on a more expressive switching linear dynamic systems (SLDS) model.

The approach presented in this paper fits into the general category of data clustering and dynamic modeling. Two classic examples in this category are fitting mixture models and training hidden Markov models (HMMs) using the EM algorithm [2]. Typically these models assume fixed dependency structure for the observed data. The study of models whose graphical structure is contingent upon the values/context of the nodes in the graph can be traced back to Heckerman and Geiger’s similarity networks and multinets [3]. This class of models has been further explored and formalized by Boutilier et al.’s Context-Specific Independence [4] and more recently Milch et. al’s Contingent Bayesian Networks [5]. An HFactMM fits into this class of models and is closely related to Bilmes’s Dynamic Bayesian Multinets [6]. The focus of [6] was to show how learning state-indexed structure using labeled training data can yield better models for classification tasks. In contrast, here the dependency structures are defined by the problem and no labeled data is required.

HFactHMMs are also closely related to SLDS models used by the tracking community [7]. SLDS models are combinations of discrete Markov models and linear state-space dynamical systems. The hidden discrete state chooses between a predefined number of state-space models to describe the data at each point in time. SLDSs are primarily used to help improve tracking and track interpretation by allowing changes to the state-space model parameters. Exact inference and learning for such models is difficult. However, in this paper we are only interested in changes to the dependency structure of the observations and consider the latent dynamic state in the state-space model a nuisance variable. An HFactMM explicitly models varying dependence structure over time and allows for efficient learning and inference. This paper explores the tradeoff between using this simple model and incorporating a latent dynamic state using an SLDS model when performing a dynamic dependency test. We begin by describing HFactMMs and later show how inference on an HFactMM can be used as a subroutine in an approximate inference algorithm for SLDS models.

II. Hidden Factorization Markov Model

Let $\mathbf{O}_t = \{\mathbf{o}_t^1, \mathbf{o}_t^2, \ldots, \mathbf{o}_t^N\}$ be an observation of N random variables at time t with $\mathbf{o}_t^i \in \mathbb{R}^{d_i}$. Let $\mathbf{O}_{1:T}$ represent $\mathbf{O}_t$ from time 1 to T. Given $\mathbf{O}_{1:T}$, the goal is to label the sequence according to the dependency among the N random variables at each time t. To this end, we propose a pth order hidden factorization Markov Model, HFactMM(p), in which we assume that the observation $\mathbf{O}_t$ depend on the past p observations and a hidden state S_t. The states $S_{1:T}$ are Markov yielding

$$p(\mathbf{O}_{1:T}, S_{1:T}; \Theta) = p(S_{1:T}; \Theta) \prod_{t=1}^{T} p(\mathbf{O}_t | \mathbf{O}_{(t-p):(t-1)}, S_t; \Theta), \tag{1}$$

where Θ are the parameters. This model is an HMM when $p = 0$ and a Markov switching auto-regressive model (MSAR) for $p > 0$. However, it has the special property that the value $k \in [1...K]$ of the hidden state variable S_t indicates one of K possible factorizations F^k and parametrizations Θ^k such that

$$\begin{aligned} p(\mathbf{O}_t | (\mathbf{O}_{(t-p):(t-1)}, S_t = k; \Theta) &= \prod_{i=1}^{C_k} p(F_{i,t}^k | F_{i,(t-p):(t-1)}^k; \Theta^k) \\ &\triangleq p_{\Theta^k,p}(F_t^k), \end{aligned} \tag{2}$$

where F^k specifies a partitioning of a full set of N random variables into C_k subsets such that

$$\bigcup_{i=1}^{C_k} F_i^k = \{\mathbf{o}^1, \ldots, \mathbf{o}^N\} \quad \text{and} \quad F_i^k \bigcap F_j^k = \emptyset \tag{3}$$

$\forall i, j \in [1...C_k]$ when $i \neq j$. For example, given three objects, $p = 1$ and the factorization for state k, $F^k = \{\{\mathbf{o}^1, \mathbf{o}^2\}, \{\mathbf{o}^3\}\}$, yields $p_{\Theta^k,1}(F_t^k) = p(\mathbf{o}_t^1, \mathbf{o}_t^2 | \mathbf{o}_{t-1}^1, \mathbf{o}_{t-1}^2; \Theta^k) p(\mathbf{o}_t^3 | \mathbf{o}_{t-1}^3; \Theta^k)$ as the state conditional distribution of the observation at time t. Figure 1 shows an example order 0 HFactMM with two possible interactions ($K = 2$) in which $F^1 = \{\{\mathbf{o}^1, \mathbf{o}^2\}, \{\mathbf{o}^3\}\}$ and $F^2 = \{\{\mathbf{o}^2, \mathbf{o}^3\}, \{\mathbf{o}^1\}\}$. Note that the value of the state S_t determines the probabilistic structure of the observations at time t.

We consider situations in which the model parameters are not known *a priori*. This necessitates both a learning and inference step. The Baum-Welch/EM algorithm can be used with a slight modification for learning the parameters of an HFactMM, subsequently Viterbi decoding can be used for exact inference [2]. We construct and utilize an HFactMM model in the following way:

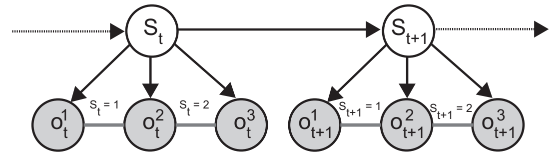


Fig. 1. Example HFactMM(0). Conditionally labeled edges are only present when the condition is true, following the notation in This graph notation follows that of CBNs [5].

1) Define the K possible dependency structures and parameterization of the HFactMM for your task.
2) Learning: Estimate $\hat{\Theta} = \arg\max_{\Theta} p(\mathbf{O}_{1:T}; \Theta)$ (via EM)
3) Inference: Find $\hat{s}_{1:T} = \arg\max_{s_{1:T}} p\left(s_{1:T} | \mathbf{O}_{1:T}; \hat{\Theta}\right)$

This approach does not use any training data and performs both learning and inference on the data being analyzed. We make the usual assumption that all K states are visited at least once (and typically multiple times) during the observed sequence.

A. Learning

Let $\Theta = \{\pi, Z, \Theta^1, \ldots, \Theta^K\}$ be the parameter set for the model where $\pi_k = p(S_1 = k)$ are the prior state probabilities, Z is a $K \times K$ matrix with $Z_{ij} = p(S_{t+1} = i | S_t = j)$, and Θ^k are the parameters for factorization F^k (*i.e.*, parameters for $p_{\Theta^k,p}(F^k)$). As with typical HMMs and MSAR models, the EM algorithm can be applied to models with this structure in order to find the parameters, $\hat{\Theta}$, that maximize the likelihood function [2]. While the E-step is unchanged, the HFactMM requires a minor change to the M-step of EM. Since the state conditional model $p_{\Theta^k,p}(F^k)$ breaks up into the C_k factors of F^k, the structure of the M-step updates simplify accordingly yielding a more structured learning procedure with savings in storage and computation.

B. Inference

Having learned the parameters, $\hat{\Theta}$, the data sequence is labeled by finding the most likely state sequence

$$\{\hat{s}_{1:T}\} = \arg\max_{s_{1:T}} p(s_{1:T} | \mathbf{O}_{1:T}; \hat{\Theta}). \tag{4}$$

This can be done efficiently with the Viterbi algorithm [2]. Viterbi decoding implicitly performs an M-ary hypothesis test comparing all $M = K^T$ possible state sequences. This alternative view exposes how state sequences distinguish themselves via both structural and general statistical model differences between the learned state conditional distributions. Consider a binary hypothesis test between two different state sequences $S_{1:T}^{H_1}$ and $S_{1:T}^{H_2}$ given the learned parameters $\hat{\Theta}$. The test has the form

$$\hat{L}_{H_1,H_2} \triangleq \log \frac{p\left(\mathbf{O}_{1:T} | S_{1:T}^{H_1}; \hat{\Theta}\right)}{p\left(\mathbf{O}_{1:T} | S_{1:T}^{H_2}; \hat{\Theta}\right)} \underset{H_2}{\overset{H_1}{\gtrless}} \log\left(\frac{p\left(S_{1:T}^{H_2}; \hat{\Theta}\right)}{p\left(S_{1:T}^{H_1}; \hat{\Theta}\right)}\right). \tag{5}$$

Taking the expected value of the log likelihood ratio when H_1 is true yields

$$E_{H_1}\left[\hat{L}_{H_1,H_2}\right] = \sum_{\substack{\forall t \text{ s.t.} \\ S_t^{H_1} \neq S_t^{H_2}}} D\left(p_{\hat{\Theta}^{S_t^{H_1}},p}\left(F^{S_t^{H_1}}\right) || p_{\hat{\Theta}^{S_t^{H_1}},p}\left(F^{\emptyset}\right)\right) + \sum_{\substack{\forall t \text{ s.t.} \\ S_t^{H_1} \neq S_t^{H_2}}} D\left(p_{\hat{\Theta}^{S_t^{H_1}},p}\left(F^{\emptyset}\right) || p_{\hat{\Theta}^{S_t^{H_2}},p}\left(F^{S_t^{H_2}}\right)\right), \tag{6}$$

where $F^{\emptyset}$ is the common dependency structure (set of edges) shared by all F^k (*e.g.*, if $\{\mathbf{o}^1, \mathbf{o}^2\}$ is common to all K structures then it will also appear in $F^{\emptyset}$). Note that any parameter set Θ^k can be applied to the $F^{\emptyset}$ factorization by marginalizing over $p_{\Theta^k}(F^k)$ appropriately. An equivalent break down occurs under H_2 (with a negation). Notice that the expected log likelihood ratio decomposes into two terms. The first term concerns purely structural differences. It compares the true structure with the common structure, under the true parameters. The second term contains both structural and parameter differences. It compares the common structure with the true parameters to the incorrect hypothesis. Typically HMMs and MSAR models assume a fixed dependency structure and thus only the second term is non-zero and depends only on parametric differences. The other extreme occurs when performing windowed dependency tests which only take advantage of the first term [8].

III. Incorporating a State-Space Model

A HFactMM with order $p > 0$ explicitly models the dynamics of the observation sequence. If we assume a pth order Gaussian auto-regressive model the state conditional distribution will have the form

$$p(\mathbf{O}_t|\mathbf{O}_{(t-p):(t-1)}, S_t) = N(\mathbf{O_t} - \mathbf{A}^{S_t}\left[\mathbf{O}_{(t-p):(t-1)}\right]; 0, \mathbf{\Sigma}). \tag{7}$$

i.e., the observation at time t will be a linear combination of the previous p with the addition of Gaussian noise. $\mathbf{A}^{S_t}$ determines the linear combination when the state is S_t. This is a simple model in which the dynamic state of system is being directly observed. However, it is common to assume there is a latent dynamic state that is only partially observed through a noisy measurement process. Such systems are often assumed to be linear with Gaussian noise following a state-space model of the form

$$\begin{aligned} \mathbf{X}_t &= \mathbf{A}\mathbf{X}_{t-1} + \mathbf{V}_t \\ \mathbf{O}_t = \mathbf{Y}_t &= \mathbf{C}\mathbf{X}_t + \mathbf{W}_t, \end{aligned} \tag{8}$$

where $\mathbf{X}_t$ is a hidden dynamic state variable at time t and $\mathbf{V}_t$ and $\mathbf{W}_t$ are independent Gaussian noise sources. An SLDS assumes there is an additional hidden discrete state S_t that determines $\mathbf{A}$, $\mathbf{C}$, and the covariance of the noise sources. The high level structure of such models is shown in Figure 2(a). For the problem of object interaction analysis S_t indexes how the distribution of the dynamic state $\mathbf{X}_t$ factorizes via the structure of $\mathbf{A}$ and the noise source $\mathbf{V}_t$. Note that the dynamics on the observations are induced via the dynamics on the latent dynamic state $\mathbf{X}_t$.

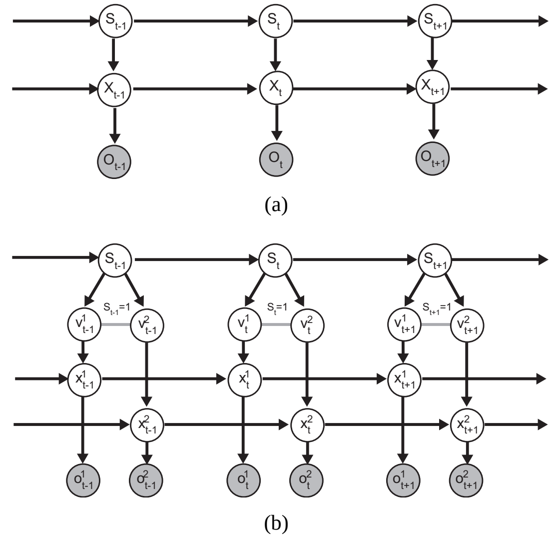


Fig. 2. High level SLDS structure (a) and detailed structure of model used in experiments (b)

Learning and exact inference on an SLDS model is difficult. For example, if S_t can take K discrete values which indexes K different Gaussian models for $\mathbf{X}_t$, the distribution $p(\mathbf{X}_t|\mathbf{O}_{1:t})$ is a mixture of K^t Gaussians, each one associated with a possible state sequence $S_{1:t}$. This exponential growth can be dealt with in many ways from collapsing the K^t mixture into a smaller mixture at each t [9], [10], to using a variational approach which fits a simpler model to maximizing a lower bound on the likelihood of the data [7].

As an alternative we take an iterative coordinate ascent approach. If one were given the state sequence $S_{1:T}$ EM can be used to learn the parameters of the model and inference on the dynamic state $\mathbf{X}_{1:T}$ can be performed using Rauch-Tung-Strieber smoothing with the appropriate model parameters (which are specified at each time t by S_t). Similarly if one were given the dynamic state $\mathbf{X}_{1:T}$ learning and inference for the $S_{1:T}$ is simple. That is, conditioned on $\mathbf{X}_{1:T}$, $\mathbf{O}_t$ can be ignored and one is left with an HFactMM of order 1 which treats $\mathbf{X}_{1:T}$ as its observation.

Thus we begin by first initializing to a random state sequence $\hat{S}_{1:T}$, then perform learning and inference on $\mathbf{X}_{1:T}$ to produce an estimate $\hat{\mathbf{X}}_{1:T}$. We next condition on this estimate to update $\hat{S}_{1:T}$ and repeat. It is important to remember that for the application of interest S_t indexes a particular structure via factorization F^{S_t}. For this state-space model this factorization describes how the dynamic state $\mathbf{X}_{1:T}$ evolves. We will give a specific example in the next section.

IV. EXPERIMENTS

Here we consider the $N = 2$ case in which we have noisy observations of position for two objects. We assume there are two states of interaction specified by the hidden discrete state S_t. If $S_t = 0$ we assume they are moving independently and if $S_t = 1$ they are interacting. We begin with a constant velocity model where a latent dynamic state evolves via

$$\begin{bmatrix} \mathbf{x}_t^1 \\ \mathbf{x}_t^2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}^1 & 0 \\ 0 & \mathbf{A}^2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{t-1}^1 \\ \mathbf{x}_{t-1}^2 \end{bmatrix} + \begin{bmatrix} \mathbf{v}_t^1 \\ \mathbf{v}_t^2 \end{bmatrix}, \tag{9}$$

where the dynamic state is position and velocity in 2D, $\mathbf{x}_t^i = \begin{bmatrix} x_t^i & y_t^i & \dot{x}_t^i & \dot{y}_t^i \end{bmatrix}^\top$, and

$$\mathbf{A}^1 = \mathbf{A}^2 = \begin{bmatrix} \mathbf{I} & \Delta\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{10}$$

where $\mathbf{I}$ and $\mathbf{0}$ are 2x2 identity and zero matrices. Noisy observations of position are obtained via

$$\begin{bmatrix} \mathbf{o}_t^1 \\ \mathbf{o}_t^2 \end{bmatrix} = \begin{bmatrix} \mathbf{C}^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{C}^2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_t^1 \\ \mathbf{x}_t^2 \end{bmatrix} + \begin{bmatrix} \mathbf{w}_t^1 \\ \mathbf{w}_t^2 \end{bmatrix}, \tag{11}$$

where $\mathbf{C}^1 = \mathbf{C}^2 = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \end{bmatrix}$ and the observation noise is independent of the state S_t, *i.e.*,

$$p(\mathbf{w}_t^1, \mathbf{w}_t^2 | S_t) = p(\mathbf{w}_t^1, \mathbf{w}_t^2) = N\left(\mathbf{w}_t^1; 0, \sigma_w \mathbf{I}\right) N\left(\mathbf{w}_t^2; 0, \sigma_w \mathbf{I}\right) \tag{12}$$

Furthermore, in this simple model we assume that all information about the interactive state is captured via the driving noise. That is, when $S_t = 0$ the driving noise is independent and factorizes as

$$p(\mathbf{v}_t^1, \mathbf{v}_t^2 | S_t = 0) = N\left(\mathbf{v}_t^1; 0, \mathbf{\Sigma}_v\right) N\left(\mathbf{v}_t^2; 0, \mathbf{\Sigma}_v\right) \tag{13}$$

and when $S_t = 1$ the noise is fully dependent

$$p(\mathbf{v}_t^1, \mathbf{v}_t^2 | S_t = 1) = N\left(\begin{bmatrix} \mathbf{v}_t^1 \\ \mathbf{v}_{t;}^2 \end{bmatrix}; 0, D \begin{bmatrix} \mathbf{\Sigma}_v & \rho\mathbf{\Sigma}_v \\ \rho\mathbf{\Sigma}_v & \mathbf{\Sigma}_v \end{bmatrix}\right) \tag{14}$$

with correlation ρ and a factor D more variance. Here ρ describes the amount of dependency when they are interacting. It is the only parameter that affects the first term in Equation 6. The parameter D changes only the second term in Equation 6. The structure of this model is shown in Figure 2(b). This give us a simple model with parameters that are easy to interpret. It is important to note that the HFactMM and SLDS models used in this paper are not limited to this form and can capture other types of interaction (*e.g.*, via changing a $\mathbf{A}$ matrix).

A. Illustrative Example

For the purpose of testing the approach under known conditions, we draw multiple sample paths from the model described above with various settings of ρ, D and observation noise variance σ_w. We do this with a fixed dynamic on the state S_t with $\pi_0 = \pi_1 = .5$, $Z_{00} = Z_{11} = .95$ and $Z_{12} = Z_{21} = .05$. For each setting of ρ, D, and σ_w we draw $T = 400$ samples (of both $\mathbf{X}_t$ and $\mathbf{O}_t$) from the model. This is performed 50 times for each setting and the average performance of a dynamic dependency test using 3 different approaches is recorded.

As a baseline, the first approach is given the true dynamic state $\mathbf{X}_{1:T}$ as its observations and uses an HFactMM(1) model to do learning and inference. The full set of parameters used for this baseline model are $\Theta = \{\pi, Z, \Theta^0, \Theta^1\}$ where the independent ($S_t = 0$) parameters are $\Theta^0 = \{\mathbf{A}^1, \mathbf{A}^2, \mathbf{\Sigma}_v^1, \mathbf{\Sigma}_v^2\}$, and the interacting/dependent ($S_t = 0$) parameters are $\Theta^1 = \{\mathbf{A}^{12}, \mathbf{\Sigma}_v^{12}\}$. The independent model parameters match Equation 9, while a more generic model is used for the dependent case with $\mathbf{X}_t = \mathbf{A}^{12}\mathbf{X}_{t-1} + \mathbf{V}_t$ and $\mathbf{V}_t$ drawn from a zero mean Gaussian with full covariance $\mathbf{\Sigma}_v^{12}$. Given $\mathbf{X}_{1:T}$ the baseline model learns the most likely set of parameters $\hat{\Theta}$ and then outputs the most likely state sequence.

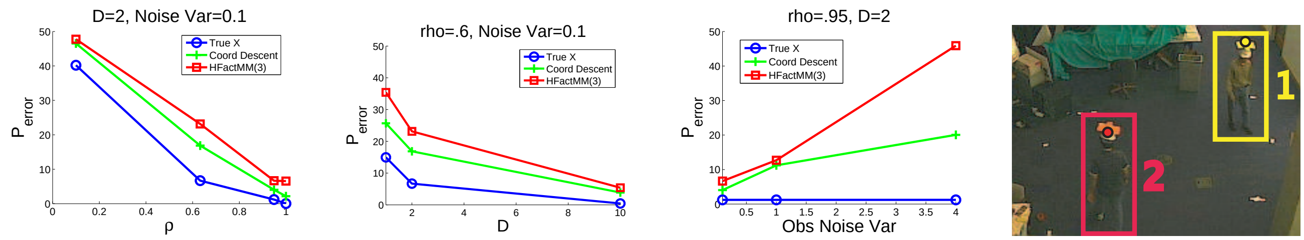
The second approach is given only the observation sequence $\mathbf{O}_{1:T}$ and performs coordinate ascent on the full SLDS model. It has the same parameter set as the baseline approach with the addition of the observation matrix $\mathbf{C}$ and observation noise variance σ_w. For simplicity we assume these observation process parameters are known. We set a maximum number of iterations of coordinate ascent to 10 but found that it usually converges in less than 5 iterations for our experiments.

The third approach is given only the observation sequence and fits a higher order HFactMM(3). While this model yields efficient inference and learning it does not model the latent dynamic state $\mathbf{X}_t$. It models the effects of any latent dynamics through a higher order auto-regressive model on the observations. Again, note that for the problems of interest in this paper, $\mathbf{X}_t$ is a nuisance variable. Our primary focus is on accurately estimating the interactive state S_t and not on producing accurate estimates of the dynamic state $\mathbf{X}_t$. A question we wish to explore is whether not this simple model is useful even when there is a true underlying latent dynamic state.

Figure 3(a) shows for fixed D and σ_w the average probability of error for these three techniques as a function ρ. As expected all three approaches improve dramatically as ρ increases. A similar trend occurs for a fixed ρ and σ_w and an increasing D in Figure 3(b). Figure 3(c) shows how observation noise degrades the performance of the approaches that do not have the benefit of directly observing the dynamic state. In all of the figures it is clear even though the coordinate ascent technique comes with more computational complexity and approximate inference it outperforms the simple HFactMM(3) by incorporating a latent dynamic state. However this performance gap disappears when the state conditional models are easily distinguishable (*i.e.*, high ρ or D).

B. Interacting People

Next we analyzed the interaction of two objects moving in a real environment. These two objects are individuals wearing hats designed to be easy to visually track. The



(a) (b) (c) (d)

Fig. 3. Avg. probability of error for synthetic data as a function of (a) ρ (b) D (c) σ_w. (d) Sample frame from interacting people data.

TABLE I
RESULTS FOR INTERACTING PEOPLE DATA. CV=CONSTANT VELOCITY. B=BROWNIAN MOTION

	Probability of Error		
Sequence	Coord. Asc. CV	Coord. Asc. B	HFactMM(1)
1	16.8	1.9	2.0
2	26.6	3.3	3.5

two individuals move around an enclosed environment and are randomly instructed to “interact” for a limit period of time. When interacting they are either following or trying to mirror each other’s actions. Each sequence was over 2.5 minutes long and contained between 8 to 12 transitions between interacting and moving independently. In the first sequence the individuals move faster when interacting. This is not the case for the second sequence in which individuals maintain the same speed when interacting and moving independently. Both sequences were recorded using a camcorder at 12 fps. Simple background subtraction and color filtering with smoothing provided a simple blob tracker that was further labeled/tracked by hand to provide (x, y) positions for each individual for each frame. Ground truth of interaction state sequence was also established by hand labeling. Figure 3(d) shows a sample frame from one of the sequences.

We ran dynamic dependency tests on both of these sequences using the coordinate ascent approach assuming a constant velocity model (4 dimensional dynamic state initialized with an $\mathbf{A}$ matrix with constant velocity structure). The results are shown in the second column of Table I. After analyzing the learned parameters for each sequence we found that the dynamic state transition matrix $\mathbf{A}$ was close to diagonal indicating that a Brownian motion model may be more appropriate. Rerunning the dependency test assuming a simple Brownian motion model (the state is position only) we obtained the improved results in column three of Table I. This indicates that a constant velocity model may not be correct for this data in which individuals make many turns in a confined space. Lastly if we apply a simple first order HFactMM to the data we do just as well. This indicates that this data has strong dependency information when the individuals are interacting (*i.e.*, like the high ρ case above).

V. CONCLUSION

In this paper we looked at the problem of inferring latent dynamic dependency structure to determine the interaction among multiple moving objects. We have shown that by modeling dependency as a dynamic process one can exploit both structural and parameter differences to distinguish between hypothesized states of dependency/interaction. We discussed the use of an HFactMM for such tasks and showed how it can be used to perform approximate inference on an SLDS model which incorporates a state-space description of object dynamics. Experiments were performed on both controlled synthetic and recorded data of people/objects interacting. Empirical performance demonstrated the utility of approximate inference based on coordinate ascent for determining interaction states.

REFERENCES

[1] F. Heider and M. Simmel, “An experimental study of apparent behavior,” in *American Journal of Psychology*, 1944, vol. 57, pp. 243–256.
[2] L.R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” in *Proc. IEEE*, 1989, vol. 77 No. 2, pp. 257–286.
[3] D. Geiger and D. Heckerman, “Knowledge representation and inference in similarity networks and bayesian multinets,” in *Artificial Integlligence*, 1996, vol. 82, pp. 45–74.
[4] Craig Boutilier, Nir Friedman, Moises Goldszmidt, and Daphne Koller, “Context-specific independence in Bayesian networks,” in *UAI*, 1996, pp. 115–123.
[5] B. Milch, B. Marthi, D. Sontag, S. Russell, D.L. Ong, and A. Kolobov, “Approximate inference for infinite contingent bayesian networks,” in *10th Intl. Workshop on Artificial Intelligence and Stats.*, 2005.
[6] J. A. Bilmes, “Dynamic bayesian multinets,” in *Proc. of 16th conf. on Uncertainty in Artificial Intelligence*, 2000, pp. 38–45.
[7] Z. Ghahramani and G. E. Hinton, “Variational learning for switching state-space models,” in *Neural Computation*, 1998, vol. 12, pp. 963–996.
[8] A.T. Ihler, J.W. Fisher, and A.S. Willsky, “Nonparameteric hypothesis tests for statistical dependency,” in *Trans. on signal processing, special issue on machine learning*, 2004.
[9] Y. Bar-Shalom and X. Li, *Estimation and Tracking: Principles, Techniques and Software*, Artech House, 1993.
[10] C-J. Kim, “Dynamic linear models with markov-switching,” in *Journal of Econometrics*, 1994, vol. 60, pp. 1–22.

This material is based upon work supported by the AFOSR under Award No. FA9550-06-1-0324. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the Air Force.