Dynamical Systems Trees

Tony Jebara
Department of Computer Science
Columbia University
New York, NY 10027
jebara@cs.columbia.edu

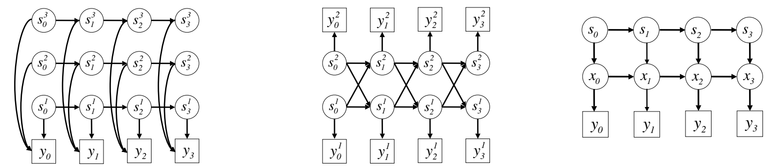
Andrew Howard
Department of Computer Science
Columbia University
New York, NY 10027
ahoward@cs.columbia.edu

Abstract

We propose dynamical systems trees (DSTs) as a flexible model for describing multiple processes that interact via a hierarchy of aggregating processes. DSTs extend nonlinear dynamical systems to an interactive group scenario. Various individual processes interact as communities and sub-communities in a tree structure that is unrolled in time. To accommodate nonlinear temporal activity, each individual leaf process is modeled as a dynamical system containing discrete and/or continuous hidden states with discrete and/or Gaussian emissions. Subsequent, higher level parent processes act like hidden Markov models that mediate the interaction between leaf processes or between other parent processes in the hierarchy. Aggregator chains are parents of the child processes the combine and mediate, yielding a compact overall parameterization. We provide tractable inference and learning algorithms for arbitrary DSTs topologies via structured mean-field. Experiments are shown for real trajectory data of tracked American football plays where a DST tracks players as dynamical systems mediated by their team processes mediated in turn by a top-level game process.

1 INTRODUCTION

Dynamical Bayesian networks are popular instantiations of graphical models and have shown promise in many applied settings such as computational biology, speech, and vision. Recently, graphical models and approximate inference methods have extended traditional dynamical systems, improving upon classical linear Kalman filters and hidden Markov models (HMMs) and exploring couplings and interactions between multiple hidden Markov chains. Such extensions include factorial HMMs [2] which indirectly link multiple Markov chains through a common output emission stream (Figure 1(a)). Meanwhile, coupled HMMs [8] directly link hidden states of multiple interacting processes that have Markovian temporal dynamics which generate different output emission streams (Figure 1(b)). Other extensions involve



(a) Factorial HMM (b) Coupled HMM (c) Switched LDS

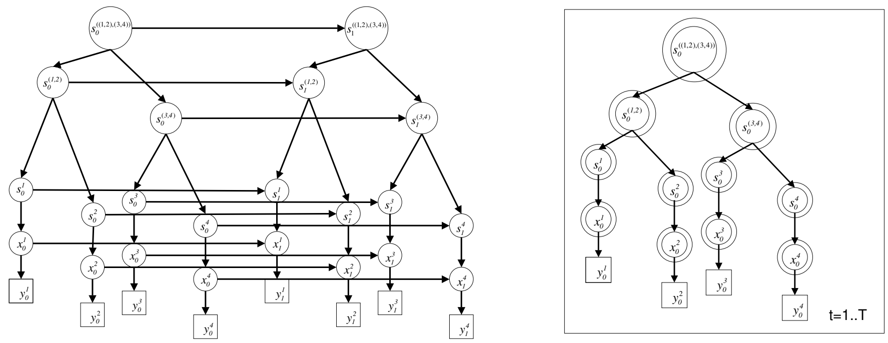
Figure 1: Switched and Interacting Dynamical Systems.

linking discrete and continuous Markov chains through so-called switched dynamical systems (SLDSs) that combine Kalman filters and HMMs [9, 1, 7, 5] to obtain nonlinear continuous dynamics (Figure 1(c)). All the above systems basically link (directly or indirectly) hidden Markov chains together so they can influence each other in time. But, unlike simpler models, these variants involve hard inference and may require sampling [7] or structured mean field approximations [10].

In this article we propose a novel variant of dynamical systems for characterizing interacting processes that form groups and sub-groups. For instance, consider a football game where players are each modeled as a switched dynamical system. Players could interact with other members of their team through a hidden parent team state which has its own Markovian dynamics. The other team has its own Markovian team state which couples its players. Finally, an overall game state is a parent of and couples the two team states mediating their interaction. We call this model a dynamical systems tree (DST) since it permits the interacting processes to couple to each other by being mediated through an arbitrary tree hierarchy of aggregating hidden processes. The DST's arbitrary hierarchical tree structure has high level aggregating hidden states coupling groups of hidden Markov states that are parents of sub-group of leaf dynamical systems (i.e. SLDSs or HMMs). Since higher level Markov chains are parents of lower level chains, a recursive structured mean field algorithm for inference is easy to derive for arbitrary tree structures and group/sub-group arrangements for the interacting processes. Thus, we are free to consider various ways that the individual leaf dynamical systems interact in a group scenario. This article describes and motivates the generative model for DSTs. Parameter estimation for DSTs is then derived via Expectation-Maximization and a structured mean field inference algorithm which can be applied recursively on any DST topology. We then provide and discuss promising experimental results with DSTs on football data from real player trajectories.

2 DYNAMICAL SYSTEMS TREES

Instead of modeling interaction by direct coupling (as in coupled HMMs) or through shared outputs (as in factorial HMMs), we propose that processes interact through parent hidden Markov chains that act as mediators or aggregators of the sub-processes. These parent chains have their own Markovian dynamics and we can also consider tree-like hierarchies of parent chains (hidden parent chains coupling multiple hidden lower-level parent chains). We call this graphical model a dynamical system tree (DST) (see Figure 2(a)). For example, multiple agents can interact and be aggregated by writing messages on some form of common bulletin board which



(a) Unrolled 2 Time Steps (b) Using Replicators

Figure 2: A Dynamical Systems Tree Graphical Model.

evolves with hidden Markovian dynamics. Alternatively, a mediator or aggregator state could represent a coach directing a team of players or a script guiding multiple actors. One advantage of such a topology is it has few parameters and avoids over-fitting. Conversely, a full HMM (as well as coupled and factorial HMMs to a certain extent) over all interacting agents must model the cross-product of their individual states which is inefficient. Furthermore, the DST's mediated interaction approach lends itself nicely to hierarchical extensions. DSTs can have aggregators that themselves aggregate lower-level mediating chains to allow multiple scales of influence or layers of interaction. For instance, when modeling a university one may describe interacting people by mediating chains representing their research groups which are in turn aggregated and mediated by various departments, then schools and then a single mediating hidden state representing the evolution of the university as a whole. Alternatively, one may model the dynamics of a human, via a hierarchy over the individual limbs, fusing into upper and lower torso, etc. Admittedly, mediating variables could be reinterpreted as children (not parents) of the individual dynamical systems, but we prefer the DST's *mediating-parent* style of establishing interaction between processes. It avoids moralizing large cliques during inference, is nicely compatible with structured mean field derivations, and permits estimation of model parameters for an arbitrary tree hierarchy of interaction.

To construct a DST's probability distribution, we start from the bottom up by first considering a collection of simple independent dynamical systems we call leaf-processes. These individual systems are either continuous linear dynamical systems, or discrete HMMs or a hybrid as in a switched linear dynamical system (SLDS). Without loss of generality, we will assume all leaf-processes are SLDSs (as in [9] and in Figure 1(c)) since these basically subsume both HMMs and Kalman filters. Furthermore, we assume that transitions between continuous hidden states are given by conditioned Gaussians and that emissions are continuous vectors from a Gaussian distribution given the continuous hidden state. On their own, the individual SLDSs do not capture the interactive nature of group dynamical behavior. To couple individual leaf-processes and model complex interaction, we have a hierarchy of aggregating Markovian processes that couple leaf-processes (or lower level aggregating-processes) as their children. Each aggregating process a is denoted by its discrete Markovian hidden variables s^a and defined as follows:

Definition 1 *An aggregating-process is a Markov chain of hidden discrete states with at most one parent process and one or more children processes. Children processes may be either other aggregating-processes themselves or leaf-processes. An aggregating-process' states are denoted by $s^a = \{s^a_0, \ldots, s^a_T\}$. Given its (possibly null) parent process $\pi(a)$ which has discrete hidden states $s^{\pi(a)} = \{s_0^{\pi(a)}, \ldots, s_T^{\pi(a)}\}$ the aggregating-process has the following conditional distribution:*

$$p\left(s^a | s^{\pi(a)}\right) = p\left(s^a_0 | s^{\pi(a)}_0\right) \prod_{t=1}^{T} p\left(s^a_t | s^a_{t-1}, s^{\pi(a)}_t\right)$$

The hierarchy of aggregating-processes is terminated by leaf-processes which contain both discrete and continuous hidden Markov dynamics as well as the actual emission or observation variables which we specify as follows:

Definition 2 *A leaf-process is a switched linear dynamical system at the lowest level in the dynamical systems tree hierarchy. A leaf-process has at most one parent process and no children processes. The i'th leaf-process has discrete Markovian hidden states $s^i = \{s^i_0, \ldots, s^i_T\}$ as parents of continuous Markovian hidden states $x^i = \{x^i_0, \ldots, x^i_T\}$ as parents of independent emissions $y^i = \{y^i_0, \ldots, y^i_T\}$. Given its parent process $\pi(i)$ with discrete hidden states $s^{\pi(i)} = \{s_0^{\pi(i)}, \ldots, s_T^{\pi(i)}\}$ the leaf-process has the following conditional distribution:*

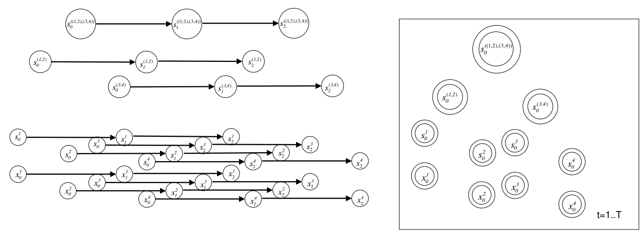
$$p(s^i, x^i, y^i | s^{\pi(i)}) = p(s^i_0 | s^{\pi(i)}_0) p(x^i_0 | s^i_0) p(y^i_0 | x^i_0) \prod_{t=1}^{T} p(s^i_t | s^i_{t-1}, s^{\pi(i)}_t) p(x^i_t | x^i_{t-1}, s^i_t) p(y^i_t | x^i_t)$$

Given $\mathcal{A}$ aggregating processes and $\mathcal{L}$ leaf-processes, the joint distribution $\mathcal{P}(\mathcal{S}, \mathcal{X}, \mathcal{Y})$ over all variables in the DST (namely $\{\mathcal{S}, \mathcal{X}, \mathcal{Y}\}$ which correspond to discrete hidden, continuous hidden and emission variables, respectively) is given by:

$$\mathcal{P}(\mathcal{S}, \mathcal{X}, \mathcal{Y}) = \prod_{a=1}^{\mathcal{A}} p(s^a | s^{\pi(a)}) \prod_{i=1}^{\mathcal{L}} p(s^i, x^i, y^i | s^{\pi(i)})$$

An example of a DST graphical model is shown in Figure 2(a) (unrolled time steps $t = 0 \ldots 1$). This DST has 4 leaf-processes and 3 aggregating-processes. The bottom aggregating processes $s^{(1,2)}$ and $s^{(3,4)}$ are parents of the pair of leaf processes in their superscripts. The bottom aggregating processes are themselves aggregated through one final parent process called $s^{((1,2),(3,4))}$. To avoid drawing DSTs unrolled in time, we plot them more compactly by only showing a single time instance of the DST at time t and drawing a *replicator box* that indicates the network is repeated $t = 1 \ldots T$ times. Traditionally, replicator boxes show independent (iid) nodes (possibly linked to parent parameter nodes which we omit for clarity). We indicate Markovian dynamics between nodes in box $t-1$ to nodes in box t by drawing extra replicator circles around all the nodes who inherit a link from their instantiation at the previous time step $t-1$. Nodes without the extra replicator circle (such as emission nodes) only have parents in the current replicator box t. Figure 2(b) depicts the DST in this compact replicator notation.

We now specify parameters for the aforementioned DST conditional distributions. Our discrete distributions are multinomials while our continuous distributions are



```
init variational parameters
forward-backward inference on all chains
compute  B(Q^0,Θ)
while  (B(Q^{t+1},Θ) − B(Q^t,Θ) ≥ ∈)
{   recurse through tree hierarchy
      for each chain in recursion
            iterate updates of variationals
            do forward-backward inference
      update  B(Q^{t+1},Θ)   for all chains     }
```

(a) Unrolled 3 Time Steps (b) Using Replicators (c) Update Pseudo-Code

Figure 3: A Variational $\mathcal{Q}$ Distribution for DSTs and an Update Algorithm.

conditioned Gaussians. The parameters for the aggregating-processes (indexed by a) and the SLDS leaf-processes (indexed by i) are:

$$\begin{aligned} p(s_0^a = j | s_0^{\pi(a)} = k) &= \phi^a(j,k) & \quad p(s_t^a = j | s_{t-1}^a = k, s_t^{\pi(a)} = l) &= \Phi^a(j,k,l) \\ p(s_0^i = j | s_0^{\pi(i)} = k) &= \psi^i(j,k) & \quad p(s_t^i = j | s_{t-1}^i = k, s_t^{\pi(i)} = l) &= \Psi^i(j,k,l) \\ p(x_0^i | s_0^i = j) &= \mathcal{N}(x_0^i | \mu_j^i, q_j^i) & \quad p(x_t^i | x_{t-1}^i, s_t^i = j) &= \mathcal{N}(x_t^i | A_j^i x_{t-1}^i, Q_j^i) \\ p(y_0^i | x_0^i) &= \mathcal{N}(y_0^i | C x_0^i, R) & \quad p(y_t^i | x_t^i) &= \mathcal{N}(y_t^i | C x_t^i, R) \end{aligned}$$

Basic operations needed for DSTs include computing the likelihood of observations, inferring hidden states from an observation and estimating parameters from data. Essentially, EM learning and computing likelihood hinge on performing inference over the hidden states. It is immediately evident that DST inference involves an intractable network since even the sub-component SLDSs are intractable. Therefore we appeal to structured mean field for inference and perform approximate E-steps.

3 A STRUCTURED MEAN FIELD ALGORITHM

To avoid the intractabilities in the DST, we perform inference with a surrogate variational distribution that approximates our posterior $\mathcal{P}(\mathcal{S}, \mathcal{X}|\mathcal{Y})$ over the hidden variables given the observed data. We denote the simpler optimized surrogate distribution $\mathcal{Q}(\mathcal{S}, \mathcal{X})$ and display it in Figure 3(a) unrolled in time for 3 time steps or in Figure 3(b) using replicator notation. This distribution resembles $\mathcal{P}$ except that all Markov chains are unlinked from each other and thus only require forward-backward algorithms for inference [1]. Given a current setting of all our model parameters Θ and observation sequences, we can update a variational distribution on our DST by using the elegant formalisms outlined by [6, 1, 4]. More specifically, we have the following inequality on the incomplete log-likelihood:

$$\log \mathcal{P}(\mathcal{Y}|\Theta) \geq \sum_{\mathcal{S}} \int_{\mathcal{X}} \mathcal{Q}(\mathcal{S}, \mathcal{X}) \log \frac{\mathcal{P}(\mathcal{S}, \mathcal{X}, \mathcal{Y}|\Theta)}{\mathcal{Q}(\mathcal{S}, \mathcal{X})} d\mathcal{X}$$

Where the right hand side is denoted by $\mathcal{B}(\mathcal{Q}, \Theta)$ for short and is a variational bound that makes contact with the left hand side at $\Theta = \Theta^*$ when $\mathcal{Q}(\mathcal{S}, \mathcal{X}) = \mathcal{P}(\mathcal{S}, \mathcal{X}|\mathcal{Y}, \Theta^*)$. Since $\mathcal{Q}$ is a simpler and more factorized distribution than $\mathcal{P}(\mathcal{S}, \mathcal{X}|\mathcal{Y}, \Theta^*)$, the bound will be lowered and in general can no longer make tangential contact. We instead optimize the parameters of $\mathcal{Q}$ to get it as close as possible to the posterior in terms of Kullback-Leibler divergence $KL(\mathcal{Q}(\mathcal{S}, \mathcal{X}) \| \mathcal{P}(\mathcal{S}, \mathcal{X}|\mathcal{Y}))$.

Update rules for $\mathcal{Q}$ are easily derived using the Hamiltonians (the energy function in the log domain) of the probability distributions [1]:

$$\mathcal{P}(\mathcal{S},\mathcal{X},\mathcal{Y}) = \tfrac{1}{Z}\exp(-H(\mathcal{S},\mathcal{X},\mathcal{Y})) \qquad \mathcal{Q}(\mathcal{S},\mathcal{X}) = \tfrac{1}{Z_{\mathcal{Q}}}\exp(-H_{\mathcal{Q}}(\mathcal{S},\mathcal{X}))$$

The $\mathcal{Q}$ distribution has the following variational parameters (which vary with each time replicator) for each aggregating-process and leaf-processes:

$$\begin{aligned}
\mathcal{Q}\left(s_0^a = j\right) &= \hat{\phi}^a(j) \qquad & \mathcal{Q}\left(s_t^a = j | s_{t-1}^a = k\right) &= \hat{\Phi}_t^a(j,k) \\
\mathcal{Q}(s_0^i = j) &= \hat{\psi}^i(j) \qquad & \mathcal{Q}(s_t^i = j | s_{t-1}^i = k) &= \hat{\Psi}_t^i(j,k) \\
\mathcal{Q}(x_0^i) &= \mathcal{N}(x_0^i | \hat{\mu}^i, \hat{q}^i) \qquad & \mathcal{Q}(x_t^i | x_{t-1}^i) &= \mathcal{N}(x_t^i | \hat{A}_t^i x_{t-1}^i, \hat{Q}_t^i)
\end{aligned}$$

As in [1] to find a $\mathcal{Q}$ that minimizes the KL-divergence, we set to zero the derivatives of the difference of Hamiltonians $D = H_{\mathcal{Q}} - H$. These derivatives are taken with respect to the sufficient statistics [1] of $\mathcal{Q}$ and give update rules:

$$\frac{\partial\langle D\rangle}{\partial\langle \mathbf{s}_t^a\rangle} = \frac{\partial\langle D\rangle}{\partial\langle \mathbf{s}_t^a \mathbf{s}_{t-1}^a\rangle} = \frac{\partial\langle D\rangle}{\partial\langle \mathbf{s}_t^i\rangle} = \frac{\partial\langle D\rangle}{\partial\langle \mathbf{s}_t^i \mathbf{s}_{t-1}^i\rangle} = \frac{\partial\langle D\rangle}{\partial\langle x_t^i\rangle} = \frac{\partial\langle D\rangle}{\partial\langle x_t^i x_{t-1}^i\rangle} = 0$$

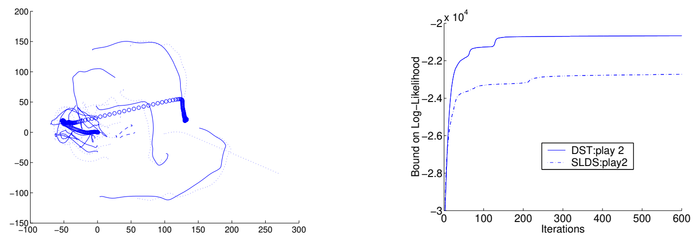
Solving the above updates the variational parameters for each aggregator-processes (indexed by a) and each leaf-processes (indexed by i) as follows:

$$\begin{aligned}
\hat{\Phi}_t^a(j,k) \quad &\propto \quad \exp\Big(\textstyle\sum_l \langle s_t^{\pi(a)}(l)\rangle \log \Phi^a(j,k,l) + \sum_{c\in\mathrm{Child}(a)} \sum_{h,i} \langle s_t^c(h) s_{t-1}^c(i)\rangle \log \Phi^c(h,i,j)\Big) \\
\hat{\phi}^a(j) \quad &\propto \quad \exp\Big(\textstyle\sum_l \langle s_t^{\pi(a)}(l)\rangle \log \phi^a(j,l) + \sum_{c\in\mathrm{Child}(a)} \sum_h \langle s_0^c(h)\rangle \log \Phi^c(h,j)\Big) \\
\hat{\Psi}_t^i(j,k) \quad &\propto \quad \exp\Big(\textstyle\sum_l \langle s_t^{\pi(i)}(l)\rangle \log \Psi^i(j,k,l) - \frac{1}{2}\log|Q_j^i| \; - \frac{1}{2}\big\langle (x_t^i - A_j^i x_{t-1}^i)'(Q_j^i)^{-1}(x_t^i - A_j^i x_{t-1}^i)\big\rangle\Big) \\
\hat{\psi}^i(j) \quad &\propto \quad \exp\Big(\textstyle\sum_l \langle s_t^{\pi(i)}(l)\rangle \log \psi^i(j,l) - \frac{1}{2}\log|q_j^i| \; - \frac{1}{2}\big\langle (x_0^i - \mu_j^i)'(q_j^i)^{-1}(x_0^i - \mu_j^i)\big\rangle\Big) \\
\hat{A}_t^i \quad &= \quad \hat{Q}_t^i \textstyle\sum_j \langle s_t^i(j)\rangle (Q_j^i)^{-1} A_j^i \\
(\hat{Q}_t^i)^{-1} \quad &= \quad \textstyle\sum_j \langle s_t^i(j)\rangle (Q_j^i)^{-1} + \sum_j \langle s_{t+1}^i(j)\rangle (A_j^i)'(Q_j^i)^{-1} A_j^i - (\hat{A}_{t+1}^i)'(\hat{Q}_{t+1}^i)^{-1}\hat{A}_{t+1}^i \\
\hat{\mu}^i \quad &= \quad \hat{q}^i \textstyle\sum_j \langle s_0^i(j)\rangle (q_j^i)^{-1} \mu_j^i \\
\hat{q}^i \quad &= \quad \textstyle\sum_j \langle s_0^i(j)\rangle (q_j^i)^{-1} + \sum_j \langle s_1^i(j)\rangle (A_j^i)'(Q_j^i)^{-1} A_j^i - (\hat{A}_1^i)'(\hat{Q}_1^i)^{-1}\hat{A}_1^i
\end{aligned}$$

Note that these are conditional distributions and should be properly normalized. After iterating the variational parameter updates, we perform forward-backward inference on $\mathcal{Q}$ to get normalized probabilities and marginals. We use these to compute expectations over our hidden variables. Since $\mathcal{Q}$ is a set of disconnected chains, we need the following marginals: $p(x_t)$, $p(x_t, x_{t-1})$, $p(s_t)$ and $p(s_t, s_{t-1})$ for all hidden variables. After the above approximate E-step, an M-step update of the parameters Θ is trivial via expectations of the complete likelihood using the current $\mathcal{Q}$ distribution. The update rules for the model parameters for a Kalman filter (with continuous dynamics and continuous emission models) are shown in [1]. Similarly, updating the discrete Markov chain's transition matrix (or tensor) is immediate.

Computing the model's true log-likelihood, however, remains intractable. We instead evaluate the bound, $\mathcal{B}(\mathcal{Q}, \Theta)$. During learning, the bound increases monotonically as we iterate variational parameter updates in $\mathcal{Q}$ and model parameter

[1] Note sufficient statistics for M-state discrete variables are *not* expectations $\langle s_t\rangle$ and $\langle s_t, s_{t-1}\rangle$. In the exponential family, they are the $M-1$ dimensional truncation of the multinomial variable s_t which we call $\mathbf{s}_t$. Therefore, rewrite Hamiltonians with $\mathbf{s}_t$ and replace $s_t(M)$ with $1 - \sum_{m=1}^{M-1} \mathbf{s}_t(m)$ before computing expectations and derivatives.



(a) Player Trajectories (b) Training Log-Likelihood Bounds

Figure 4: Player Trajectory Data and Variational EM Training

updates in Θ to achieve a local maximum. We compute the bound via the expected Hamiltonian H under $\mathcal{Q}$ summed with the entropy of the $\mathcal{Q}$ distribution: $\mathcal{B}(\mathcal{Q}, \Theta) = E_{\mathcal{Q}(\mathcal{S},\mathcal{X})}\{H(\mathcal{S}, \mathcal{X}, \mathcal{Y}) - H_{\mathcal{Q}}(\mathcal{S}, \mathcal{X})\}$. These expectations are easy to compute and only involve expectations over (at most pairwise) cliques of $\mathcal{Q}$. In all the above computations, it is easy to recurse through the DST to compute the $\mathcal{B}(\mathcal{Q}, \Theta)$ terms for each leaf-process and aggregator-process. Furthermore, variational parameter updates are nicely decoupled and M-steps for Θ parameters are independent given the inferred expectations under the $\mathcal{Q}$ distribution. In Figure 3(c) we show pseudo-code for propagating through the hierarchy tree to update variational parameters. This is interleaved with re-estimation of the model parameters.

4 EXPERIMENTS

We evaluated DSTs and other dynamical models on real-world trajectory data from American football plays [3]. Players are tracked using computer vision to obtain spatial coordinates in the football field (with some normalization). Each human generates a continuous time series of two dimensional coordinates. Our training data consisted of 5 example plays of *dig* maneuvers and testing data was two new *dig* exemplars. Figure 4(a) shows the trajectories for multiple players during a play. A naive approach to modeling our data is to stack or concatenate each player's time series into a single multivariate series. We start with the simplest multivariate time series model, a *single* Kalman Filter (LDS) or *single* SLDS (which do not treat players as individual temporal processes). The table below shows low test log-likelihoods for *single* LDSs and SLDSs as we evaluate them on the two unseen dig plays, even as we increase dimensionality of hidden states x or s. To instead model players as separate temporal processes, we trained *multiple independent* SLDSs for each player in isolation. Each has 2 dimensional continuous x_t state and 2 states for the switches s_t. Yet such SLDSs completely ignore interactions between players. A DST, however, can couple many separate temporal interactions by fusing SLDS structures (as above) with two additional binary-state team-chains aggregating the two teams of players and a top level binary-state game-chain aggregating the two teams. In Figure 4(b) we show the monotonic convergence of the EM-style algorithm while training on 5 example plays for both the DST and the independent SLDSs. Unlike LDSs, both SLDSs and DSTs estimate (conservative) lower bounds on test likelihoods and require approximate inference (variationals converge in 5-10 iterations for each test play). Five random initializations are done and we show the mean and standard deviation of the log-likelihood bound for both the DST and

SLDSs. Test results in the table below show that independent SLDSs improve on single LDSs and SLDSs yet the DST has the best generalization on testing data[2].

Model	Log-Likelihoods Test Play 1	Log-Likelihoods Test Play 2
Single LDS $dim_X = 1$	-1.9E5	-1.5E5
Single LDS $dim_X = 2$	-1.9E5	-1.5E5
Single LDS $dim_X = 3$	-2.7E7	-3.2E7
Single SLDS $dim_S = 2$	-7.7E5 $\pm\epsilon$	-1.5E6 $\pm\epsilon$
Single SLDS $dim_S = 4$	-1.8E5 $\pm\epsilon$	-1.5E5 $\pm\epsilon$
Multi SLDSs $dim_S = 2$	-1.4E4 $\pm$ 3.4E2	-1.6E4 $\pm$ 1.0E3
DST $dim_S = 2$	-5.6E3 $\pm$ 1.8E2	-6.1E3 $\pm$ 1.4E2

The weakness of single LDSs and SLDSs suggests we treat each time series for each player separately (not as a single multivariate series). Using independent multi SLDS models for each player does improve modeling yet still fails to capture interactions between players. Thus, it is important to *fuse* (not just concatenate) multiple interacting time series at a *higher level* as in DSTs which ultimately yielded highest test likelihoods. We find DSTs are promising and flexible dynamical Bayesian networks for coupling multiple interacting processes in a tree structured hierarchy of influence. Results indicate that, for certain real temporal datasets, they are more appropriate than simpler alternatives and capture elaborate interdependencies without compromising computational tractability. Perhaps most interestingly, they provide an easily reconfigurable and intuitive architecture for modeling temporal interaction data.

References

[1] Z. Ghahramani and G.E. Hinton (1998). Variational Learning for Switching State-Space Models. *Neural Computation*, 12(4):963-996.

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

[3] S.S. Intille and A.F. Bobick (2001). Recognizing planned, multi-person action. *Computer Vision and Image Understanding*, 81(3):414-445.

[4] T.S. Jaakkola (2000). Tutorial on variational approximation methods. In *Advanced mean field methods: theory and practice.* MIT Press.

[5] U. Lerner, B. Moses, M. Scott, S. McIlraith, D. Koller (2002). Monitoring a Complex Physical System using a Hybrid Dynamic Bayes. *UAI.*

[6] R.M. Neal and G.E. Hinton (1998). A new view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models.*, Kluwer.

[7] B. North, A. Blake, M. Isaard, and J. Rittscher (2000). Learning and Classification of Complex Dynamics, *IEEE PAMI*, 22(9).

[8] N. Oliver, B. Rosario and A. Pentland (1998). Graphical Models for Recognizing Human Interaction. *NIPS 11.*

[9] V. Pavlovic, J.M. Rehg, and J. MacCormick (2001). Learning Switching Linear Models of Human Motion. *NIPS 13.*

[10] L. Saul and M.I. Jordan (1996). Exploiting tractable substructures in intractable networks. *NIPS 8.*

[2] In similar experiments on *wham* maneuvers, DSTs again performed best on testing. More results (omitted due to space) are at http://www.cs.columbia.edu/~jebara/dst.