Time Varying Dynamic Bayesian Network for Non-Stationary Events Modeling and Online Inference

Zhaowen Wang*, Ercan E. Kuruoğlu, Senior Member, IEEE, Xiaokang Yang, Senior Member, IEEE, Yi Xu, and Thomas S. Huang, Life Fellow, IEEE

Abstract—This paper presents a novel Time Varying Dynamic Bayesian Network (TVDBN) model for the analysis of non-stationary sequences which are of interest in many fields. The changing network structure and parameter in TVDBN are treated as random processes whose values at each time epoch determine a stationary DBN model; this DBN model is then used to specify the distribution of data sequence at the time epoch. Under such a hierarchical formulation, the changing state of network can be incorporated into the Bayesian framework straightforwardly. The network state is assumed to transit smoothly in the joint space of numerical parameter and graphical topology so that we can achieve robust online network learning even without abundant observations. Particle filtering is employed to dynamically update current network state as well as infer hidden data values. We implement our time varying model for data sequences of multinomial and Gaussian distributions, while the general model framework can be used for any other distribution. Simulations on synthetic data and evaluations on video sequences both demonstrate that the proposed TVDBN is effective in modeling non-stationary sequences. Comprehensive comparisons have been made against existing non-stationary models, and our proposed model is shown to be the top performer.

Index Terms—Bayesian networks, time varying, particle filters, event recognition.

I. INTRODUCTION

MODELING the evolution of temporal sequences is of great interest in many areas such as signal processing, automation, finance, computational biology, etc. Among the numerous tools designed for the analysis of temporal sequences, Dynamic Bayesian Networks (DBNs) [1] have been the most successful ones. A DBN is the extension of a Bayesian Network (BN) to temporal domain, in which conditional dependencies are modeled between random variables both within and across time slots. The conditional distributions are assumed to be homogeneous in DBN; that is, the structure and parameter of DBN are fixed throughout the time. Under this assumption, a DBN is effectively constructed by unrolling a BN in time axis, and the model learning procedure can be greatly simplified. However, this bold assumption limits the power of DBN in modeling many non-stationary sequences, where the intrinsic relationships among variables change from time to time. Such non-stationary sequences may arise in all aspects of our life. Some examples include: the steering pattern of a vehicle under different road conditions; the appearance of an object across multiple cameras; the gene interactions in different stages of a life cycle; and the stock prices in different economic periods. A fixed statistical model is obviously inadequate to model these data sequences at all time instances.

Incorporating the temporal variation of network structure and parameter into DBN is a natural way to handle non-stationary sequences. However, learning and inferring such a time varying network is a non-trivial job. The naive approach to learn a DBN independently for each time instance is not feasible since in most applications too few observations can be obtained from just one time instance. One way to ameliorate this data scarcity problem is to pre-segment temporal sequences into stationary epochs, in each of which data are generated from the same probability distribution. But the segmentation itself is hard due to the lack of knowledge for models in each epoch and the large solution space growing exponentially with the length of sequence. On the other hand, our observation data are often corrupted with noise, and there will inevitably be some perturbations associated with the statistics derived from them. The noise perturbation can be indistinguishable from the true variation of underlying data distribution, particularly when the distribution is changing gradually. Furthermore, real time processing is a crucial constraint in some applications such as stock price prediction and video surveillance. The demand for adapting DBN model to new data online poses even greater challenges.

A. Previous Work

Faced with all the difficulties mentioned above, researchers have been trying to extend DBN to non-stationary scenarios by imposing various conditions on the form of network and the way it can change. Earlier works have been mainly focused on non-stationary models with fixed structure. Among them the most studied one is the time varying autoregression (TVAR) model [2], which describes non-stationary linear dynamic systems with continuously changing linear coefficients and
noise variances. Normalized least square algorithm can be used to estimate the regression parameters recursively, and the estimation error is shown bounded when the parameters change smoothly [3]. Due to the well-established theory, TVAR model has found wide application in the researches of equity market [4], gene expression [5], and electroencephalogram (EEG) traces [6]. Extensions on TVAR have also been made for other time varying processes such as non-Gaussian autoregression [7] and Poisson counting process [8].

Another class of non-stationary model that has been extensively studied is the switching linear dynamic system (SLDS), in which a latent Markov chain is employed to describe the piecewise change of linear system. Time varying observation matrices are formulated in [9]; and time varying dynamic transition matrices are formulated in [10], [11]. An SLDS with arbitrary state duration distribution is proposed in [12]. An SLDS with an unknown number of stationary modes is proposed in [13] by using hierarchical Dirichlet process prior. There exist a bunch of learning algorithms for SLDS; besides the commonly used EM method, variational approximation [9], Grassmann manifold [14], and auxiliary particle filter [15] have also been applied for efficient and/or online learning.

The models of TVAR and SLDS only consider non-stationary parameter change in dynamic systems. Recently, graphical models that change dynamically in both parameter and structure have received more attention. With the assumption that data sequence is piecewise stationary in time, non-stationary models are constructed as a cascade of stationary models each of which is learnt from a pre-segmented stationary sub-intervals. A number of methods have been proposed to find these sub-intervals. In [16], Gaussian graphical model is used to represent the dependency among variables, and then the exact posterior distribution of switching times between stationary intervals is evaluated with the method of [17]. Curve manifold is employed to express time varying sequences in [18], and segmentation is carried out according to the geometric structure of manifold. Markov chain Monte Carlo (MCMC) sampling methods are used in [19], [20], which search MAP stationary sub-intervals through iterative local movements on network configuration. When the dimension of model parameter is not fixed, reversible jump MCMC can be utilized [21]. Change detection technique is used to monitor the fitness between incoming data and current network [22], and local adaptation is applied to network when large discrepancy is detected. In [23], network changes are represented by a hidden controller, whose optimal value is searched with a random hill climbing algorithm.

Since the piecewise stationary models are still not general enough for any applications, several more sophisticated non-stationary models have been developed with continuously changing network parameters taken into account. In [24], the structure of a binary network is treated as a hidden state, and its dynamic transition follows an exponential random graph process. Constraints of temporal smoothness and structural sparsity are imposed on time varying linear regression networks [25]–[27], in which linear coefficients are jointly optimized by minimizing lasso objective functions. In [28], the authors propose to model the dynamics of mixed member-

B. Proposed Method

In this paper, we propose a novel Time Varying Dynamic Bayesian Network (TVDBN) model for online inference of the underlying distribution of non-stationary sequences. We extend the basic DBN model so that both the structure and parameter of network become random variables that can change through time. These random variables are treated as additional hidden nodes in our graph model, and a smooth transition prior is imposed on their temporal variation to ease the problem of data scarcity. This novel representation of changing network allows a unified modeling of both data and network itself under the same dynamic Bayesian framework. In contrary to most off-line learning methods reported in literatures, we employ particle filter to dynamically infer the hidden states of network as well as missing data, if there are any. This key feature enables the application of our model in situations with real time constraint. The framework of our time varying model is general enough for data sequence of any distribution type; in this work, multinomial and Gaussian distributions are studied in particular. The effectiveness of the proposed TVDBN model is validated on both simulated non-stationary data and video sequences, with preliminary applications to tracking and event recognition.

It is worth noting the differences between our model and those closely related to it. A hidden variable is used to represent the change of network in [23], but it only serves as an auxiliary variable to facilitate implementation. In our model, the structure and parameter nodes are indispensable components of the whole graph model, and they represent the statistical attributes of current network. The online adaptation methods in [15], [22] can only model piecewise constant variation of network; while our method deals with both continuous change in parameter and discrete switch in structure. Smooth change of network is ensured in [25] with a kernel window applied on data sequences; we achieve similar goal via a smooth transition model, which is more favorable from a Bayesian perspective. In the Gaussian graphical model [16], network parameter is marginalized out and network structure is the only thing to investigate. Under our setting, the states of both structure and parameter are inferred to give a full description of current network.

The remainder of this article is organized as follows. First, we introduce the overall framework of TVDBN model in Section II. The transition distribution of network is detailed in Section III, in the context of two popular data distributions: multinomial and Gaussian. Section IV shows the procedure of inferring hidden variables in TVDBN using particle filter algorithm. A set of experimental results are reported in Section V. Finally, in Section VI, we conclude the paper and discuss the potential extension to large scale networks.
II. TIME VARYING DYNAMIC BAYESIAN NETWORK

A. Dynamic Bayesian Network

A Bayesian Network (BN) [29] is a graph model describing the statistical relationships among a group of $n$ random variables $\mathbf{X} = \{X_i\}_{i=1}^{n}$. A BN is determined by its graph structure $G$ and distribution parameter $\Theta$. $G$ is a directed acyclic graph with $n$ nodes corresponding to random variables $X_i$. An edge in $G$ directed from node $i$ to $j$ encodes $X_j$’s conditional dependence on $X_i$, and $X_i$ is called a parent of $X_j$. A variable $X_i$ is independent of its non-descendants given all its parents $\mathbf{Pa}(X_i)$ in $G$. Therefore, the joint probability distribution over $\mathbf{X}$ can be decomposed by the chain rule:

$$p(\mathbf{X}) = \prod_{i=1}^{n} p(X_i|\mathbf{Pa}(X_i)) \quad (1)$$

The parameter set $\Theta = \{\Theta_i\}_{i=1}^{n}$ specifies the parameters of each conditional distribution $p(X_i|\mathbf{Pa}(X_i))$ in Eq. (1). The meaning of $\Theta_i$ is interpreted according to the specific form of the distribution. When the distribution is multinomial, $\Theta_i$ is simply a conditional probability table; when the distribution is Gaussian, $\Theta_i$ may contain the values of mean and variance.

A Dynamic Bayesian Network (DBN) [1], [30] is the extension of a BN to model temporal processes. In DBN, a set of random processes are represented by the $\mathbf{X} = \{X_i\}_{i=1}^{n}$, and $X_i[t]$ is the random variable of process $X_i$ at discrete time $t$. The network structure $G$ now defines the dependency among variables over a period of time as well as those within the same time epoch. Usually, the processes $\mathbf{X}$ are assumed to be Markovian and causal. So a node in graph $G$ is only allowed to be linked from the other nodes in the same or previous epoch, i.e., $\mathbf{Pa}(X_i[t]) \subset \{X_i[t-1],X_i[t]\}$. Also, all the conditional distributions are assumed to be stationary. Thus we have:

$$p(\mathbf{X}[t+1]|\mathbf{X}[0 : t]) = p(\mathbf{X}[t+1]|\mathbf{X}[t]), \ t = 0, 1, 2... \quad (2)$$

The model parameter $\Theta$ is defined similarly as in BN. $\Theta$ and $G$ are kept constant over time under the stationarity assumption.

B. Time Varying Network Representation

The stationarity assumption has been favored for its simplicity in most applications of DBN so far. However, this assumption is not always valid in real life. For example, a driver will change his driving style according to different traffic conditions, and a constant velocity dynamic model is apparently incapable of describing the movement of vehicle. Such time varying characteristic is crucial to a better understanding of temporal events. To this end, we introduce a Time Varying Dynamic Bayesian Network (TVDBN) model whose network structure and parameter can vary as the underlying distribution of temporal sequence changes.

In our time varying formulation, the network structure and parameter are modeled as random processes whose values at time $t$ are denoted as $G[t]$ and $\Theta[t]$, respectively. These random variables, regarded as structure nodes and parameter nodes in our model, are used to construct a graph together with data nodes $\mathbf{X}[t]$. Since there is no way to observe the structure nodes and the parameter nodes directly, these nodes are hidden in nature. We can only infer them from the observations available on data nodes.

At each time epoch $t$, we have $n$ parameter nodes: $\Theta[t] = \{\Theta_i[t]\}_{i=1}^{n}$. Each $\Theta_i[t]$ is linked to node $X_i[t]$ and determines its conditional distribution jointly with other data nodes linked to it. Therefore, in TVDBN, the parents of a data node $X_i[t]$ are defined as:

$$\mathbf{Pa}(X_i[t]) \equiv \{\mathbf{Pa}(X_i[t]),\Theta_i[t]\} \quad (3)$$

where $\mathbf{Pa}(X_i)$, called $X_i$’s non-parameter parents, is the set of data nodes that are linked to $X$. The linkage between data nodes, or equivalently their probabilistic dependency, is determined by the network structure $G[t]$. Here $G[t]$ is represented as a set of directed edges:

$$G[t] = \{e_{ji}, e'_{ji}|i, j = 1...n\} \quad (4)$$

where $e_{ji}$ stands for an edge pointing from $X_j[t]$ to $X_i[t]$, and $e'_{ji}$ stands for an edge pointing from $X_j[t-1]$ to $X_i[t]$. We only consider the nodes at time $t - 1$ and $t$ as $X_i[t]$’s potential parents, as required by the Markovian property. To ensure $G[t]$ is an acyclic graph, it is also required that there is no sequence $i_1, ..., i_k$ such that $e_{i_1,i_2+1}$, $e_{i_2,i_3+1}$, ..., $e_{i_k,i_{k+1}}$ exist in $G[t]$ for $1 \leq j < k$ and $e_{i_k,i_1} \in G[t]$. Given $G[t]$, we can retrieve $X_i[t]$’s non-parameter parents as:

$$\mathbf{Pa}(X_i[t]) = \{X_j[t]|e_{ji} \in G[t]\} \cup \{X_j[t-1]|e'_{ji} \in G[t]\} \quad (5)$$

It should be noted that the dependency of $X_i[t]$ on $\Theta_i[t]$ is fixed and will not be eliminated over time.

To model the changing processes of $\Theta[t]$ and $G[t]$ in TVDBN, we adopt a first order Markov model $p(\Theta[t + 1], G[t + 1]|\Theta[t], G[t])$ as their joint transition distribution. Assume that given structure $G[t]$, each parameter $\Theta_i[t]$ transits independently, and we have

$$p(\Theta[t + 1], G[t + 1]|\Theta[t], G[t]) = p(G[t + 1]|G[t])p(\Theta[t + 1]|\Theta[t], G[t + 1]) = p(G[t + 1]|G[t])\prod_{i=1}^{n} p(\Theta_i[t + 1]|\Theta_i[t], G[t + 1]) \quad (6)$$

The dependency of $\Theta[t + 1]$ on $G[t + 1]$ and $G[t]$ is due to the fact that the dimension of distribution parameter is affected by network structure.

An example of TVDBN with changing parameter and structure is illustrated graphically in Fig. 1. All the black edges represent the probabilistic dependency among nodes, which play the same role as the edges in a BN or DBN. While the green and blue edges in Fig. 1 have essentially different meanings. A green edge links a structure node $G[t]$ to the slice of network enclosed in a corresponding green rectangle, which contains all the edges connected towards $\mathbf{X}[t]$ from other data nodes of time $t - 1$ and $t$. The presence and absence of any of these edges, or equivalently all the $\mathbf{Pa}(X_i[t])’s$, are

1 The Markovian restriction is imposed here for convenience in discussion. In more general situation, any node at time earlier than $t$ can be parent of $X_i[t]$. 
determined by the state of node \( G_i[t] \). A blue edge links a structure node \( G[t] \) to a parameter node \( \Theta_i[t] \) of the same time epoch, and indicates that the dimension of parameter \( \Theta_i[t] \) is determined by the status of \( G[t] \). The dependency encoded in a green or blue edge is deterministic, as opposed to the probabilistic dependency encoded in a black edge. Therefore, our TVDBN model is different from traditional DBN model, and a new method should be designed for its online inference. Before addressing that in Section IV, we will first discuss the transition models for the structure and parameter in time varying network.

III. NETWORK TRANSITION DISTRIBUTION

The transition distribution in Eq. (6) plays a crucial role in our time varying model. The first term on the right side of Eq. (6), \( p(G[t + 1] | G[t]) \), characterizes the dynamic transition of network structure. Dynamic graph processes have been studied extensively in random graph theory. Some rules have been formulated to model the typical evolution of graphs over time, such as preferential attachment [31], copying [32], and preferential deletion [33]. However, most dynamic graph processes are devoted to modeling social and web-like networks, which usually feature different behaviors than Bayesian networks. Therefore, the specialized rules for random graphs are not applicable to our TVDBN model. Transition kernels for Bayesian network are proposed in [34] that make local modification to network structure in an iterative manner. Such transition kernel is sufficient for off-line learning of a stationary network, but is not flexible enough to model dynamically changing network.

Here we propose to model the sequence of network structure \( \{G[t]\} \) as a Markov chain, whose state space \( \mathcal{G} = \{G_i\} \) consists of all valid directed acyclic graphs that connect data nodes \( \mathbf{X}[t] \) across adjacent time epochs or within the same time epoch. When the number of data nodes \( n \) is fixed and moderately large, using Markov chain model is a reasonable approach because the finite set \( \mathcal{G} \) can be enumerated. Given the current network structure \( G_i \), the probability of transiting to structure \( G_j \) in next epoch is set to be:

\[
p(G[t + 1] = G_j | G[t] = G_i) \propto \exp(-\lambda_1 |G_j| - \lambda_2 |G_j - G_i|)
\]  

(7)

where \( |G_j| \) denotes the number of edges in graph \( G_j \), and \( |G_j - G_i| \) denotes the number of edges that have been changed (added or deleted) from \( G_i \) to \( G_j \). \( \lambda_1 \) and \( \lambda_2 \) are parameters controlling the relative importance of the two terms. The transition model of \( G[t] \) is designed in such a way to favor sparse network structure and smooth network transition. Sparseness requires the time varying model to adapt to data distribution with a network structure as simple as possible. In this way, the problem of model over-fitting can be avoided. This idea is inspired by Bayesian Information Criterion (BIC) score [35], which penalizes complex models according to the number of model parameters. The smoothness restriction is introduced to deal with data scarcity. Typically, the data observed in one time epoch alone are far from enough for us to learn the unknown network structure. A good prior knowledge of how the network changes will enable us to estimate its structure using both current and previous data observation. Therefore, we impose the restriction that the network can only change smoothly over time; i.e., the total number of edges being changed from one time epoch to the next is expected to be small. This smoothness assumption is valid for many data sequences in real life. For example, in video applications we have a much higher signal sampling rate (frame rate) than event occurring rate (human behavior), so the underlying data distribution can be thought as changing slowly and smoothly.

Fig. 2 illustrates the structure transition model of a TVDBN with two data nodes. To save space, we have only plotted part of all possible network topologies. The arrows indicate the most likely transitions between different structures, which are realized by adding or removing a single edge in the network. The second term on the right side of Eq. (6) is the product of transition probabilities of all network parameters. Under the same smoothness assumption as above, we require that the parameter transition distribution \( p(\Theta_j[t + 1] | \Theta_j[t], G[t + 1]) \) should have a large mass in the neighborhood around \( \Theta_j[t] \). In this way, we can learn the parameters of network robustly even in the absence of abundant observation data. On the other hand, the transition of \( \Theta_j[t] \) also depends on how it parameterizes the distribution of data node \( X_i[t] \). We should customize the form of \( p(\Theta_j[t + 1] | \Theta_j[t], G[t + 1]) \) for different data distributions. In the following, we shall discuss this point in the context of two particular cases where the conditional distribution of \( X_i[t] \) is multinomial/Gaussian. Data distributions of other types can also be incorporated in the framework of our TVDBN model as long as suitable parameter transition models are devised.

A. Time Varying Multinomial DBN

When the variables \( X_i[t] \)’s are discrete, the simplest way to represent the conditional dependency among them is using multinomial distribution. The parameter \( \Theta_j[t] \) of multinomial distribution is a conditional probability table containing a collection of probability vectors \( \{\Theta_{ij}[t]\} \), where \( \Theta_{ij}[t] \) corresponds to the \( j \)’th configuration of \( X_i[t] \)’s non-parameter

Fig. 1. Graphical representation of a TVDBN over time \( t = 1, 2, 3, 4 \). At each time epoch, there are two data nodes (above the dash line), two parameter nodes, and one structure node (below the dash line). The black edges indicate the probabilistic dependency among nodes. The green edges indicate the network topologies inside the green rectangles are determined by the structure nodes. The blue edges indicate the dimensions of parameter nodes are determined by the structure nodes.
parents – $\text{Pa}^j(X_i[t])$. Vector $\Theta_{ij}[t]$ contains the probabilities for all possible values of $X_i[t]$ given the parent:

$$p(X_i^k[t]|\text{Pa}^j(X_i[t]), \Theta_{ij}[t]) = \theta_{ijk} [t] \geq 0$$  \hspace{1cm} (8)

where $X_i^k[t]$ is the $k$th possible value of $X_i[t]$, and $\theta_{ijk}[t]$ is the $ijk$th element in $\Theta_{ij}[t]$.

The probability vectors are assumed to propagate independently, so that the transition distribution of $\Theta_i[t]$ can be decomposed as:

$$p(\Theta_i[t+1]|\Theta_i[t], G[t+1]) = \prod_{j=1}^{n_i[t+1]} p(\Theta_{ij}[t+1]|\Theta_{ij}[t], G[t+1])$$

where $n_i[t+1]$ is the number of possible configurations of $\text{Pa}(X_i[t+1])$ given the current network structure $G[t+1]$. The temporal evolution of each probability vector is further modeled by a hierarchical Dirichlet distribution [36]:

$$p(\Theta_{ij}[t+1]|\Theta_{ij}[t], G[t+1]) \sim \text{Dir}(\Theta_{ij}[t+1]; \alpha \cdot \Theta_{ij}) \propto \prod_k \theta_{ijk}[t+1]^{\alpha \cdot \theta_{ijk}[t+1]}$$

where $\text{Dir}(\cdot)$ denotes the Dirichlet distribution, $\Theta_{ij}$ is the distribution center to be derived from $\Theta_{ij}[t]$, and $\alpha$ is a smoothing coefficient. The Dirichlet distribution is well suited for modeling the density of probability vectors, since its pdf is supported on the probability simplex $\{\Theta_{ij}[t+1]|\theta_{ijk}[t+1] \geq 0, \sum_k \theta_{ijk}[t+1] = 1\}$. Moreover, the hierarchical structure can restrict the variation of $\Theta_{ij}[t+1]$ within the space around $\Theta_{ij}$, a mechanism keeping the network parameter change smoothly over time. The smoothness can be easily adjusted via the coefficient $\alpha$.

When the new network structure $G[t+1]$ does not change from its previous value $G[t]$ — or more precisely, $X_i[t+1]$ shares the same set of parent nodes with $X_i[t]$ (with a shift in $t$) — the $\Theta_{ij}$ in Eq. (10) is simply chosen to be the previous probability vector that corresponds to the same parent configuration: $\Theta_{ij} = \Theta_{ij}[t]$. However, if $G[t+1]$ changes from $G[t]$ with some edges being added and/or deleted, further elaboration is required in the design of $\Theta_{ij}$. Consider the node $X_i[t]$ and its non-parameter parents $\text{Pa}(X_i[t])$. Suppose the network structure $G[t+1]$ changes from $G[t]$ such that $\text{Pa}(X_i[t+1])$ becomes different from $\text{Pa}(X_i[t])$ with an addition of nodes $Y$ and a deletion of nodes $Z$. More formally, let us denote

$$\text{Pa}(X_i[t]) = \text{Pa}(X_i[t, t+1]) \cup Z[t]$$

$$\text{Pa}(X_i[t+1]) = \text{Pa}(X_i[t, t+1], t) \cup Y[t+1]$$

where $\text{Pa}(X_i[t, t+1])$ and $\text{Pa}(X_i[t, t+1], t)$ are the non-parameter parents shared by $X_i[t]$ and $X_i[t+1]$ (with a shift in time) at $t$ and $t+1$, respectively. Under the assumption of smooth model changing, the conditional distribution of $X_i[t+1]$ can be approximated as follows:

$$p(X_i[t+1]|\text{Pa}(X_i[t+1])) \approx p(X_i[t, t+1]|\text{Pa}(X_i[t, t+1], t))$$

$$\approx \sum p(X_i[t]|\text{Pa}(X_i[t, t+1], t)) p(Z[t]|\text{Pa}(X_i[t, t+1], t))$$

where $\text{Pa}(X_i[t, t+1], t)$ is equal to the sub-vector of $\text{Pa}(X_i[t, t+1], t)$ truncated. Thus, a good choice of $\Theta_{ij}$ can be:

$$\Theta_{ij} = \sum_k p(Z[k]|\theta_{ij}, \theta_{ij}[t])$$

where $\text{Pa}(X_i[t, t+1], t)$ is equal to the sub-vector of $\text{Pa}(X_i[t, t+1], t)$ truncated.

### B. Time Varying Gaussian DBN

When data nodes $X[t]$ represent continuous variables, linear Gaussian system is often used to describe their joint dynamic transition:

$$X[t] = F[t]X[t-1] + \nu[t]$$

where $F[t]$ is a $n \times n$ matrix with elements $\{ f_{ij}[t] \}$, and $\nu[t]$ is a Gaussian noise sampled from $\mathcal{N}(\nu[t]; 0, \Sigma[t])$. Since $X[t]$ are joint Gaussian, they can be equivalently represented by a
Gaussian belief network [37], and the conditional distribution of a single node \( X_i[t] \) can be expressed as:

\[
p(X_i[t]|\text{Pa}(X_i[t]), \Theta_i[t]) = \mathcal{N}(X_i[t]; m_i[t] + \sum_{X_j[t] \in \text{Pa}(X_i[t])} b_{ij}[t](X_j[t] - m_j[t]), v_i[t])
\]

where \( m_i[t] = \sum_j f_{ij}[t] X_j[t-1] \) is the unconditional mean of \( X_i[t] \); \( v_i[t] \) is the conditional variance of \( X_i[t] \) given the value of parent nodes \( \text{Pa}(X_i[t]) \). \( \text{Pa}(X_i[t]) \) are specified by network structure \( G[t] \) for all \( t \)'s; and \( b_{ij}[t] \) is a linear coefficient measuring the strength of dependency of \( X_i[t] \) on \( X_j[t] \in \text{Pa}(X_i[t]) \). The parameter set \( \Theta_i[t] \) for node \( X_i[t] \) in this time varying Gaussian DBN is defined as:

\[
\Theta_i[t] = \{ \{ f_{ij}[t] \}, v_i[t], \{ b_{ij}[t] \} \}
\]

(17)

Obviously, we have \( v_i[t] > 0 \) for all \( i \), and \( b_{ij}[t] = 0 \) for \( X_j[t] \notin \text{Pa}(X_i[t]) \). As will be seen below in the definition of parameter transition distribution, it is convenient to adopt the form of parameters as in Eq. (17). The transformation from \( \{ \{ v_i[t] \}, \{ b_{ij}[t] \} \} \) to \( \Sigma[t] \) can be accomplished using the procedure described in [38].

We assume that all the parameters in \( \Theta_i[t] \) propagate independently:

\[
p(\Theta_i[t+1]|\Theta_i[t], G[t+1]) = p(v_i[t+1]|v_i[t]) \prod_j p(f_{ij}[t+1]|f_{ij}[t], G[t+1]) \\
\times p(b_{ij}[t+1]|b_{ij}[t], G[t+1])
\]

(18)

Also, Eq. (16) can be interpreted as a linear regression of \( X_i[t] \) with coefficients \( f_{ij}[t] \), \( b_{ij}[t] \), and variance \( v_i[t] \). The time varying auto-regression model [2] offers a convenient way to model the transition of the time varying parameters. According to [2], the variation of linear coefficients can be modeled by random walk processes:

\[
p(f_{ij}[t+1]|f_{ij}[t], G[t+1]) = \mathcal{N}(f_{ij}[t+1]; f_{ij}[t], \sigma_f^2) \quad (19)
\]

\[
p(b_{ij}[t+1]|b_{ij}[t], G[t+1]) = \mathcal{N}(b_{ij}[t+1]; b_{ij}[t], \sigma_b^2), \quad \text{if} \ e_{ij} \in G[t+1]
\]

(20)

where \( \sigma_f \) and \( \sigma_b \) are standard deviations. The coefficients are set to zero when the corresponding edges are missing in \( G[t+1] \). To model the change of non-negative variance \( v_i[t] \), we employ a multiplicative random walk process as in [2]:

\[
p(v_i[t+1]|v_i[t]) = \text{Beta}(\frac{v_i[t]}{v_i[t+1]} d; a, b)
\]

(21)

where \( a \) and \( b \) are the parameters of Beta distribution, and \( d \) is a discount factor.

IV. SEQUENTIAL MONTE CARLO INFERENCE OF TVDBN

After constructing the time varying dynamic Bayesian network model and its transition distribution, we are ready to do on-line inference of the unknown variables in network. The unknown variables of our interest at each time \( t \) include unobserved data\(^2\), network structure, and network parameter. We combine them as a hidden state \( s_t \):

\[
s_t = [X[t], G[t], \Theta[t]]
\]

(22)

The dynamic transition distribution of state \( s_t \) can be decomposed into the product of the transition distributions defined in Section III:

\[
p(s_t|s_{t-1}) = p(G[t]|G[t-1])p(\Theta[t]|\Theta[t-1], G[t]) \\
\times p(X[t]|X[t-1], \Theta[t], G[t])
\]

(23)

At each time epoch, we acquire an observation \( o_t \), which measures part or all of the data nodes. The relationship between observation and state is governed by an observation distribution \( p(o_t|s_t) \).

Given the initial state \( s_0 \), we are going to recursively estimate the current state posterior \( p(s_t|o_1:t) \) using the observations up to now. The dynamic evolution of \( s_t \) is very complicated, because \( s_t \) may contain both discrete and continuous variables, and even its dimensionality is not fixed (due to the varying dimension of \( \Theta[t] \)). Therefore, it is generally impossible to get a closed-form solution for the posterior distribution of \( s_t \). We can only resort to some numerical method such as particle filtering to find an approximated solution.

Particle filtering (or sequential Monte Carlo) [39], [40] is employed here to estimate the state posterior because of its capability to handle arbitrary system and observation models. In a particle filter, the posterior distribution is approximated by a finite set of \( N_s \) state samples (particles) \( \{ s_t^i \} \) and the associated weights \( \{ w_t^i \} \):

\[
p(s_t|o_1:t) \approx \sum_{i=1}^{N_s} w_t^i \delta(s_t^i - s_t)
\]

(24)

When \( N_s \) approaches infinity, the approximation can be arbitrarily close to the true distribution.

At each time epoch \( t \), we start the filtering with the sample set \( \{ s_t^i \}_{i=1,..,N_s} \) of previous time. New samples are first drawn from a proposal distribution:

\[
s_t^i \sim q(s_t^i|s_{t-1}^i, o_t)
\]

(25)

The proposal distribution \( q(\cdot) \) has its support over the whole state space of \( s_t \). In this way, different network structures and network parameters of different dimensions can be explored within the same framework as hidden data are inferred. This “trans-dimensional” proposal density is similar in spirit to the transition kernel of RJMCMC [34]; the difference is that we have used the idea to learn distribution in a dynamic system.

After obtaining the new samples \( \{ s_t^i \} \), we measure the likelihood of each of them with observation model, and then update their weights according to

\[
w_t^i \propto w_{t-1}^i \frac{p(o_t|s_t^i)p(s_t^i|s_{t-1}^i)}{q(s_t^i|s_{t-1}^i, o_t)}
\]

(26)

Finally, a resample step will be taken if the variance of particle weights is too large [40]. The resultant sample set

\(^2\)here we denote unobserved data using \( X[t] \) abusively, even if some data nodes are observable.
There are a total of 21 situations, we should integrate the knowledge of the dimension of the state space is high, most of the samples accumulation of likelihoods. We will use this distribution as reduces the weight updating procedure in Eq. (26) into simple proposal density in order to increase sampling efficiency. An

$\pi$ to the performance of particle filter. State transition distribution and network.

Choosing a good proposal density $q(s_t|s_{t-1}, \omega_t)$ is critical to the performance of particle filter. State transition distribution $p(s_t|s_{t-1})$ is a popular choice for proposal density, and it reduces the weight updating procedure in Eq. (26) into simple accumulation of likelihoods. We will use this distribution as proposal density in most of our experiments. However, when the dimension of the state space is high, most of the samples drawn from the transition distribution will have very small weights — a problem known as “degeneracy”. In such a situation, we should integrate the knowledge of $\omega_t$ into the proposal density in order to increase sampling efficiency. An example of how to do this for multinomial data distribution will be shown in Section V-A.

V. EXPERIMENTAL RESULTS

In this section, several experiments are conducted to validate the effectiveness of the proposed TVDBN model on multinomial and Gaussian data distributions. Our test data come from both simulation and video sequences. In all the experiments we have achieved online model adaptation, which is a key advantage over most existing algorithms. To ensure fair comparison, we apply all the off-line models in our experiments ([16], [23], [25]) in a “pseudo online” manner; i.e., they are re-estimated at each epoch from scratch using the observations up to current time.

A. Simulation

We first consider a two-node TVDBN model as shown in Fig. 2. Both nodes are discrete and observable, and their conditional distributions are multinomial. $X_1[t]$ has 2 states; $X_2[t]$ has 3 states. Parameters $\Theta_1[t]$ and $\Theta_2[t]$ are defined as the conditional probability tables for the two nodes, which are represented by matrices with 2 and 3 columns respectively. The network structure $G[t]$ is indexed by a positive integer. There are a total of 21 structural variations, some of which are plotted in Fig. 2.

Our test data are generated by a synthetic time varying DBN over 150 time epochs. The network shows three different structures during the whole period, and the corresponding parameters either change linearly or keep piecewise constant. A full description of this time varying network is given in Table I. We draw 300 sample sequences of $X[t]$ from the network and use them as observations.

We use 500 particles to estimate the posterior distribution of network structure and parameters. The initial state of network is assumed to be known. Since all the data nodes are observable, the hidden state in this experiment is just $s_t = [G[t], \Theta[t]]$. The dimension of multinomial parameter $\Theta_i[t]$ is usually high, so we try to improve the sampling efficiency of particle filter with a sub-optimal proposal density:

$$q(s_t|s_{t-1}, \omega_t) = p(G[t]|G[t-1])q(\Theta[t]|\omega_t, \Theta[t-1], G[t])$$

$$= p(G[t]|G[t-1]) \prod_{i=1}^{n} \prod_{j=1}^{n} q(\Theta_{ij}[t]|\omega_t, \Theta_{ij}[t-1], G[t]) (27)$$

where

$$q(\Theta_{ij}[t]|\omega_t, \Theta_{ij}[t-1], G[t]) \sim Dir(\Theta_{ij}[t]; \alpha \cdot \Theta_{ij} + \sum_{j} s_{ij}(\omega_t)) (28)$$

The only difference between this proposal density and the state transition distribution is the introduction of sufficient statistics $ss_{ij}(\omega_t)$, which counts the instances of $X_i[t]$, $Pa^{'v}(X_i[t])$ observed from $\omega_t$. With this additional term, the proposal density becomes closer to its optimal form $p(s_t|s_{t-1}, \omega_t)$ [39], and at the same time remains simple to manipulate. The MAP expectation of particle filtering result at each time step is calculated as the estimated network state $G[t]$ and $\Theta[t]$.

1) Continuous Non-stationary Change: We first examine the performance of TVDBN in estimating parameter $\Theta_1[t]$, which changes smoothly over time. It is shown in Fig. 3 (a) that the estimated $\Theta_1[t]$ follows closely to ground truth values. To get a principled measurement on how well the estimated network parameter matches the true underlying conditional distribution of node $X_i[t]$, Kullback-Leibler (KL) divergence between them is evaluated at each time instance and plotted in Fig. 4 (a). The resultant KL divergence of our method is compared with those of several other algorithms including the fully connected dynamic Bayesian network (DBN) [1], the kernel re-weighted TVDBN (K-TVDBN) [25], the adaptive

<table>
<thead>
<tr>
<th>$t$</th>
<th>$G[t]$</th>
<th>$\Theta_1[t]$</th>
<th>$\Theta_2[t]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,50]</td>
<td>6</td>
<td>$\begin{pmatrix} 0.804 - 0.004t &amp; 0.196 + 0.004t \end{pmatrix}$</td>
<td>$(0.3, 0.4, 0.3)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>[51,100]</td>
<td>4</td>
<td>$\begin{pmatrix} 0.6 &amp; 0.4 \end{pmatrix}$</td>
<td>$(0.3, 0.2)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>[101,150]</td>
<td>15</td>
<td>$\begin{pmatrix} 1.2 - 0.006t &amp; -0.2 + 0.006t \ -0.3 + 0.008t &amp; 1.3 - 0.008t \end{pmatrix}$</td>
<td>$(0.6, 0.1, 0.3)$</td>
</tr>
</tbody>
</table>

TABLE I

NETWORK STRUCTURE AND PARAMETERS OF THE SYNTHETIC TIME VARYING DBN.
Bayesian network (aBN) [22], the switching linear dynamic system (SLDS) [11], and the hidden controller hill climb algorithm (HCHC) [23]. A separate data set of 200 sequences is used to train the last two models before online inference. It is seen from Fig. 4 (a) that our method achieves the lowest divergence from true distribution most of the time expect for the time period [51, 100]. The HCHC and aBN methods perform slightly better during that period because their piecewise stationarity assumption is consistent with the unchanging network state at that time. Although the TVDBN model is more susceptible to observation noise in such situation, its adaptability is heavily rewarded during non-stationary period when continuous network change takes place.

2) Piecewise Stationary Change: The parameter $\Theta_2[t]$ changes in a piecewise constant way in the synthetic network, and its estimation can shed some light on TVDBN’s performance in piecewise stationary network. Some elements of the estimated $\hat{\Theta}_2[t]$ are plotted in Fig. 3 (b). It is observed that TVDBN can correctly track $\Theta_21[t]$ and quickly converge to the true value even when abrupt changes occur at time 51 and 101. The quick response to abrupt parameter change is attributed to an adaptive selection of smoothing coefficient $\alpha$ in the parameter transition model Eq. (10). We have chosen a large (1000) for parameter transition conditioned on unchanging structure ($G[t+1] = G[t]$); and a relatively small $\alpha$ (20) for parameter transition conditioned on changing structure ($G[t+1] \neq G[t]$). In this way, we can obtain stable parameter estimation within stationary period and remain sensitive to abrupt change points simultaneously. The KL divergences for the distributions of node $X_2[t]$ estimated with different methods are also compared in Fig. 4 (b). The performance of TVDBN is close to piecewise stationary models such as aBN and HCHC, and is much better than other models under comparison. For piecewise stationary models, large estimation error will be incurred if a wrong stationary mode is selected. However, TVDBN is free of this trouble because it can change both abruptly and continuously.
TABLE II
AVERAGE KL DIVERGENCES FOR EACH NODE DISTRIBUTION AND THE OVERALL DISTRIBUTION. DIFFERENT MODELS ARE COMPARED: PF-TVDBN (OUR METHOD); DBN [1]; K-TVDBN [25]; aBN [22]; SLDS [11]; HCHC [23].

| Method      | $p(X_1[t] | t)$ | $p(X_2[t] | t)$ | $p(X[t] | t)$ |
|-------------|---------------|---------------|---------------|
| PF-TVDBN    | 0.000984      | 0.004649      | 0.004482      |
| DBN         | 0.032258      | 0.042540      | 0.074798      |
| K-TVDBN     | 0.001765      | 0.007176      | 0.007176      |
| aBN         | 0.013058      | 0.020831      | 0.033888      |
| SLDS        | 0.008353      | 0.007917      | 0.016270      |
| HCHC        | 0.008294      | 0.004860      | 0.013154      |

3) Overall Performance: The estimated network structure $\hat{G}[t]$ is plotted in Fig. 5 together with ground truth. It can be seen the estimated value is correct most of the time, except for a few errors around model switching points. The effective exploration of network structure space can guide the estimation of network parameters, which is another reason for TVDBN’s better performance over its competitors.

Fig. 6 shows the comparison of KL divergences for the distribution over all nodes. The average KL divergences over time are summarized in Table II. Our method outperforms all the others and has very small KL divergence on average.

4) Random Initialization: To further investigate the online adaptability of TVDBN, we try to employ it to track time varying network with unknown initial status. Data sequences generated by the above synthetic network in time period [101, 150] are used as observations. The particle filter is initialized with random network structures and parameters. For each of the 21 possible initial network structures, the algorithm is run for 10 trials with randomly sampled parameters. The estimated network status at $t = 150$ is compared with ground truth in Fig. 7. As shown in (a), the true structure $G[150] = 15$ can be recovered with high probability from most of the random initial structures. In some trials initialized with structure 6, 7, 12, 14, the estimated network converges to a wrong mode with $G[150] = 11$, which only differs from true structure by one edge. For those trials converging to correct structure, the data distribution represented by the estimated parameters is further compared with ground truth, and the KL divergences between them are plotted in (b). The divergence is low for most trials, which shows the estimated parameters also converge to correct values. In Fig. 8, the time needed to reach convergence is plotted for each initial network structure. Convergence is defined to occur when the posterior probability of true structure is higher than 0.95 and the KL divergence from true distribution is less than 0.01. On average we can reach convergence in less than 20 time steps from any initial state, with the network still evolving in the meantime.

5) Tuning Parameters: In all the experiments above, we have set $\lambda_1 = 1.0$ and $\lambda_2 = 2.5$ for the structure transition model in Eq. (7). To get a better understanding of their
roles played in TVDBN model, we try different combinations of $\lambda_1$ and $\lambda_2$, and repeat the previous random initialization experiments with a fully connected initial network structure. For each pair of $\lambda_1$ and $\lambda_2$, the percentage and time of convergence to true structure are plotted in Fig. 9. It can be seen that as the penalty on structure complexity ($\lambda_1$) decreases and the penalty on structure change ($\lambda_2$) increases, our model will have a better chance to converge, but the time taken to reach convergence will be longer. Therefore, parameters $\lambda_1$ and $\lambda_2$ can be tuned to control the tradeoff between change sensitivity and estimation stability.

**B. Active Camera Tracking**

Video object tracking is a popular application of dynamic Bayesian network where target state is described by hidden nodes $X[t]$. In the simplest case, let $X[t] = (X_1[t], X_2[t])^T$ be the 2D image coordinates of target position at time $t$. The target dynamics can be specified by a second order autoregression model of Gaussian distribution, and the likelihood of state can be measured from image features such as HSV color histogram [41].

For real time tracking, it is often a desirable feature that camera can actively follow the moving target. However, the inconsistent self-motion of camera will be superimposed on target’s motion in image plane, which makes it no longer suitable to model the target dynamics with a stationary autoregression process. Therefore, we propose a time varying Gaussian dynamic distribution for object tracking with active camera:

$$p(X[t]|X[t-2:t-1], v[t]) = \mathcal{N}(X[t]; 2X[t-1] - X[t-2], v[t|t])$$

(29)

where $v[t] = [v_1[t], 0; 0, v_2[t]]$ is the time varying covariance matrix. The temporal variation of $v_1[t]$ and $v_2[t]$ interprets the changing accuracy of autoregression prediction caused by irregular camera motion. In this time varying Gaussian DBN, $v_1[t]$ and $v_2[t]$ constitute the network parameter $\Theta[t]$. The transition of $v_i[t]$ follows the multiplicative random walk...
distribution in Eq. (21), with $a = 5$, $b = 5$, $d = 1/2$. Network structure $G[t]$ stays constant in this experiment.

The time varying Gaussian TVDBN model is tested on a video sequence recorded by a tilt-pan camera. A yellow toy car in the video is tracked using particle filter and its initial position is labeled manually. Although both of the camera and the car are moving rapidly, our tracking results match the target positions very well, as can be seen from the snapshots in Fig. 10. The distribution of particles is visualized for frame 106 in (b), which shows the variance of particles becomes large in vertical direction when the camera turns upwards, and the uncertain motion in that direction is well handled. The strong correlation between dynamic variance and camera motion can be seen more clearly in Fig. 11, which plots the estimated value of $\hat{v}_i[t]$ and the inter-frame image translations found by global motion estimation.

The accuracy of tracking is quantitatively measured using tracking rate [42], which is defined as the percentage of frames whose tracking result overlaps with ground truth by at least 30%. Fig. 12 plots the tracking rates achieved by TVDBN model as well as stationary DBN [1], kernel re-weighted TVDBN (K-TVDBN) [25], and segmentation-based Gaussian graphic model (seg-GGM) [16]. The comparisons are conducted under different settings of particle number, and for each setting 10 trials are run. As seen from the figure, our
method achieves much higher tracking rates than the other methods under all particle settings.

C. Multiple Targets Interaction Recognition

In this experiment, TVDBN model is used to recognize the class of interaction among multiple targets based on the transition distribution of their trajectories. The similar problem has been investigated in [43] using DBN model. Consider the case of two targets, whose coordinates in image at time $t$ are represented by $(X_1[t], Y_1[t])$ and $(X_2[t], Y_2[t])$, respectively. Suppose there are three classes of interaction between the targets: wandering, joining, and moving as a group. The conditional dependency among target states will be different for each interaction class, and can be represented by a distinct network structure. In a video, the interaction between targets may change from time to time, and their trajectories are modeled by TVDBN as non-stationary sequences. We can recognize the interaction class by inferring the current network structure of TVDBN model.

The network structure $G[t]$ and the transition distribution of node $X_i[t]$ are listed in Table III for all interactions. The transition distribution of $Y_i[t]$ is defined similarly as that of $X_i[t]$. When two targets are wandering, there is no interaction between them, and the transition distribution is modeled by first order autoregression. When two targets are joining, they approach each other through the line connecting their previous positions. The speed of joining is controlled by time varying linear coefficient $b_j[t]$. In the case where two targets move together as a group, we assume that $X_1[t]$ is the state of leader target which moves by its own will, and $X_2[t]$ is the state of follower target which mimics the movement of leader target. The extent to which the follower is affected is controlled by time varying linear coefficient $b_M[t]$. Therefore, we have two time varying parameters in this TVDBN model: $\Theta[t] = \{b_j[t], b_M[t]\}$. The transition of these parameters follows the random walk model in Eq. (20), with $\sigma_{b_j} = 0.0005$ and $\sigma_{b_M} = 0.05$.

The experiment of interaction recognition is carried out on a video from the CAVIAR database [44]. We use 400 particles to track the positions and interaction class of two people simultaneously. The target positions are tracked successfully throughout the video, as shown by the trajectories in Fig. 13.

To recognize the interaction between targets, the posterior

<table>
<thead>
<tr>
<th>Interaction</th>
<th>$G[t]$</th>
<th>$p(X_i[t] \mid Pa(X_i[t]), G[t])$</th>
</tr>
</thead>
<tbody>
<tr>
<td>wandering</td>
<td>$G_{WP} = 4$</td>
<td>$X_i[t] \sim N(X_i[t]; X_i[t-1], 0.1), \ i = 1, 2$</td>
</tr>
</tbody>
</table>
| joining              | $G_J = 9$ | $X_1[t] \sim N(X_1[t]; X_1[t-1] + b_j[t](X_2[t-1] - X_1[t-1]), 0.2)$  
|                      |        | $X_2[t] \sim N(X_2[t]; X_2[t-1] + b_j[t](X_1[t-1] - X_2[t-1]), 0.2)$ |
| moving as a group    | $G_M = 17$ | $X_1[t] \sim N(X_1[t]; X_1[t-1], 0.4)$  
|                      |        | $X_2[t] \sim N(X_2[t]; X_2[t-1] + b_M[t](X_1[t] - X_1[t-1]), 0.3)$ |

Fig. 13. Trajectories of two targets (marked by colored cross and circle) tracked using TVDBN model. The color bar illustrates the time of trajectory points.

Fig. 14. The inferred posterior distribution of network structure $G[t]$ over three possible values.

Fig. 15. GT (ground truth) of interaction type versus time, and the recognition results found by PF-TVDBN (our method), DBN [1], K-TVDBN [25], seg-GGM [16], NLMS [3], HCHC [23], and SLDS [11].
The interaction corresponding to the MAP mode of \( G[t] \) is evaluated as shown in Fig. 14. The interaction corresponding to the MAP mode of \( G[t] \) is recognized as the current interaction type. Our recognition results (PF-TVDBN) are plotted in Fig. 15, together with the ground truth. The recognition results generated by several other non-stationary models are also given for comparison, which include the stationary dynamic Bayesian network (DBN) [1], the kernel re-weighted TVDBN (K-TVDBN) [25], the segmentation-based Gaussian graph model (seg-GGM) [16], the normalized least square algorithm (NLMS) [3], the hidden controller hill climb algorithm (HCHC) [23], and the switching linear dynamic system (SLDS) [11]. The confusion matrices for each method are shown in Fig. 16, and the recognition precisions are compared in Table IV. Our method over other non-stationary models, and show its prospective applications in active camera tracking and multiple targets interaction recognition.

**Discussion on Scalability**

Our work is a preliminary exploration of modeling non-stationary dynamic networks. One of a few important problems that remain to be investigated is how to model large-scale time varying networks, such as genetic network and social network. As the number of nodes grows in a network, the number of possible network topologies and the dimension of network parameter will all grow exponentially. This will aggravate the data scarcity problem and makes online network adaptation more challenging. One way to restrict the exploding solution space is making use of prior knowledge on network structure. For example, in social network, the maximal degree of a single node is usually bounded and does not increase with the size of network. Also, a huge social network may be decomposed into many small-sized cliques which are loosely connected with each other. If we can cluster all the nodes into such cliques (which is also an open question), then modeling the network will be reduced to modeling each of these cliques individually.

On the other hand, sometimes we are more interested in

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.90</td>
<td>0.87</td>
<td>0.71</td>
<td>0.83</td>
<td>0.85</td>
<td>0.80</td>
<td>0.83</td>
</tr>
</tbody>
</table>

It can adaptively track the current state of network with the latest data observation. Second, our algorithm provides a general framework that can be applied to non-stationary sequences of any distribution. The cases of multinomial and Gaussian distributions have been studied in detail. Third, we have validated the effectiveness of the proposed model with extensive experiments, including both simulation and tests on video sequences. The results demonstrate the superiority of our method over other non-stationary models, and show its prospective applications in active camera tracking and multiple targets interaction recognition.

VI. CONCLUSION AND DISCUSSION

We have proposed a new time varying dynamic Bayesian network model that is capable of describing the evolution of non-stationary temporal sequences. The network structure and parameter are assumed to change smoothly over time, and their transition distributions are designed accordingly. Particle filtering is employed for online inference of the hidden nodes and changing network, so that data space and network configuration can be explored simultaneously. Our contribution is threefold. First, the proposed TVDBN model works online.
the global behavior of the nodes in a large-scale network than the characteristic of an individual node. In such cases, all the nodes can be regarded as homogeneous and a common statistic model can be employed to describe their distributions. For example, if the relationship between any two linked nodes is governed by a common potential function, the network will reduce to a pairwise Markov random field, and the number of its parameters will no longer increase with network size.

Computational complexity is another thing to consider if TVDBN model is used for large-scale networks. As the size of network increases, a prohibitively large number of particles will be required for robust inference, which hampers the real time application of our model. In order to reduce the number of required particles, we should design more efficient proposal distribution by leveraging domain knowledge and historical data. Techniques such as Rao-Blackwellization may be used to restrict sampling space. The burgeoning field of cloud computing may also provide a solution to the computation burden associated with large networks.

ACKNOWLEDGMENT

This work was supported in part by Hi-Tech Research and Development Program of China 863 (2006AA01Z124), NSFC (60902073, 60828001, 61025005), 973 Program (2010CB731401) and the 111 Project (B07022). Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. E. E. Kuruoğlu gratefully acknowledges the support of the Institute of Image Communication and Information Processing, SJTU under 111 project and partial support from CNR-short term mobility program.

REFERENCES

WANG et al.: TIME VARYING DYNAMIC BAYESIAN NETWORK FOR NON-STATIONARY EVENTS MODELING AND ONLINE INFERENCE


Zhaowen Wang received the B.E. and M.S. degrees in electrical engineering from the Shanghai Jiao Tong University, Shanghai, China, in 2006 and 2009, respectively. He is currently pursuing Ph.D. degree in the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. Since spring 2010, he has been with Department of Electrical and Computer Engineering, UIUC. His research interests include computer vision, statistical learning, and video events analysis.

Ercan E. Kuruoglu (M’98-SM’06) was born in 1969 in Ankara, Turkey. He received the B.Sc. and M.Sc. degrees in electrical and electronics engineering from Bilkent University, Turkey, in 1991 and 1993, and the M.Phil. and Ph.D. degrees in information engineering from the University of Cambridge, Cambridge, U.K., in 1995 and 1998, respectively.

Upon completion of his studies, he joined the Xerox Research Center Europe, Cambridge, as a permanent member of the Collaborative Multimedia Systems Group. He was an ERCIM Fellow in 2000 at INRIA-Sophia Antipolis, France. In January 2002, he joined ISTI-CNRI, Pisa, Italy. He was a visiting professor at Georgia Tech-Shanghai in Autumn 2007. He is currently a Senior Researcher and Associate Professor at ISTI-CNRI. His research interests are in the areas of statistical signal and image processing and information and coding theory with applications in astrophysics, bioinformatics, telecommunications, and intelligent user interfaces.

Dr. Kuruoglu was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING between 2002-2006. He is currently an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and is on the editorial board of Digital Signal Processing: A Review Journal. He acted as the technical chair for EUSIPCO 2006. He is a member of the IEEE Technical Committee on Signal Processing Theory and Methods.

Xiaokang Yang (M’00-SM’04) received the B.S. degree from Xiamen University, Xiamen, China, in 1994, the M.S. degree from the Chinese Academy of Sciences, Shanghai, China, in 1997, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, in 2000. He is currently a Professor and the deputy Director of the Institute of Image Communication and Information Processing, Department of Electronic Engineering, Shanghai Jiao Tong University. From August 2007 to July 2008, he visited the Institute for Computer Science, University of Freiburg, Germany, as an Alexander von Humboldt Research Fellow. From September 2000 to March 2002, he worked as a Research Fellow at the Centre for Signal Processing, Nanyang Technological University, Singapore. From April 2002 to October 2004, he was a Research Scientist at the Institute for Infocomm Research (I2R), Singapore. He has published over 130 refereed papers, and has filed 14 patents.

His current research interests include visual processing and communication, media analysis and retrieval, and pattern recognition.

Dr. Yang received the Microsoft Young Professorship Award 2006, the Best Young Investigator Paper Award at IS&T/SPIE International Conference on Video Communication and Image Processing (VCIP2003), and awards from the A-STAR and Tan Kah Kee Foundations. He currently holds a number of Design and Implementation of Signal Processing Systems (DESPS) Technical Committee of the IEEE Signal Processing Society and a member of Visual Signal Processing and Communications (VSPC) Technical Committee of the IEEE Circuits and Systems Society.

Yi Xu received her B.S. and M.S. degrees from Nanjing University of Science and Technology, Nanjing, China, in 1996 and 1999, respectively, and Ph.D. degree in Information and Communication Engineering from Shanghai Jiao Tong University, in 2005.

Dr. Xu is currently an assistant professor in the Institute of Image Communication and Information Processing, Department of Electronic Engineering, Shanghai Jiao Tong University, where she works in the area of video analysis and understanding. Her research interests and sequence analysis and video content understanding, especially recognition of moving objects and abnormal events.

Thomas S. Huang (F’01) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., and the M.S. and Sc.D. degrees in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge.

He was on the Faculty of the Department of Electrical Engineering at MIT from 1963 to 1973; and on the Faculty of the School of Electrical Engineering and Director of its Laboratory for Information and Signal Processing at Purdue University from 1973 to 1980. In 1980, he joined the University of Illinois at Urbana-Champaign, where he is now William L. Everitt Distinguished Professor of Electrical and Computer Engineering, and Research Professor at the Coordinated Science Laboratory, and at the Beckman Institute for Advanced Science he is Technology and Co-Chair of the Institute’s major research theme Human Computer Intelligent Interaction. His professional interests lie in the broad area of information technology, especially the transmission and processing of multidimensional signals. He has published 21 books, and over 600 papers in network theory, digital filtering, image processing, and computer vision.

Dr. Huang is a Member of the National Academy of Engineering; a Member of the Academy Sinica, Republic of China; a Foreign Member of the Chinese Academies of Engineering and Sciences; and a Fellow of the International Association of Pattern Recognition, IEEE, and the Optical Society of America. Among his many honors and awards: Honda Lifetime Achievement Award, IEEE Jack Kilby Signal Processing Medal, and the K.S Fu Prize of the Int. Asso. for Pattern Recognition.