Knowledge Based Activity Recognition with Dynamic Bayesian Network

Zhi Zeng and Qiang Ji

Rensselaer Polytechnic Institute,
Troy, NY, 12180, USA
{zengz,jiq}@rpi.edu

Abstract. In this paper, we propose solutions on learning dynamic Bayesian network (DBN) with domain knowledge for human activity recognition. Different types of domain knowledge, in terms of first order probabilistic logics (FOPLs), are exploited to guide the DBN learning process. The FOPLs are transformed into two types of model priors: structure prior and parameter constraints. We present a structure learning algorithm, constrained structural EM (CSEM), on learning the model structures combining the training data with these priors. Our method successfully alleviates the common problem of lack of sufficient training data in activity recognition. The experimental results demonstrate simple logic knowledge can compensate effectively for the shortage of the training data and therefore reduce our dependencies on training data.

1 Introduction

During recent years, probabilistic graphical models have received increasing attention in computer vision research, such as image segmentation, object tracking and facial expression analysis. DBNs, which are designed to model temporal events, are widely adopted for recognizing human activity. Most of the existing DBN models for activity recognition are learned purely from training data, so when the amount of training data is insufficient, the performance of these models will decrease significantly. One solution to alleviate this problem is resorting to various kinds of domain knowledge.

First order logic is an expressive language in representing the logic relations in a domain and it is widely applied in many computer vision applications. Its combination with Markov networks, the Markov logic networks (MLN), can deal with rigorous logic reasoning while maintaining the capability of handling uncertainty. However, the construction of MLN requires relatively complete knowledge of the domain. If the knowledge is limited, it may lead to a highly biased model.

In our work, we first introduce a generic DBN model integrating multiple features for activity recognition, and then present a framework to learn the DBN model combining training data with domain knowledge. The domain knowledge is represented by a set of ffigure first-order probabilistic logics, which can be further transformed to the structure prior and qualitative parameter constraints on the activity model. These prior combined with the training data are used to

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 532–546, 2010.

learn the DBN structure and parameters in a CSEM framework. With simple and generic qualitative knowledge, we obtain more representative DBN structures and accurate parameters that produce better activity recognition results.

2 Related Work

Various types of DBNs have been proposed for recognizing different activities in the literature. Standard HMM [1][2] is employed for simple activity recognition, but it is not suitable for modeling complex activities that have large state and observation spaces. Different variants of HMM try to solve this problem through factorizing the state or observation space. Parallel HMMs (PaHMMs) [3], coupled HMM (CHMM) [4] and dynamic multiply-linked HMM(DML-HMM)[5] are proposed to recognize group activities by factorizing the state spaces into several temporal processes. PaHMMs ignore the interactions between different temporal processes except a zero-order synchronization, CHMM model the interactions among multiple objects through completely coupling the temporal processes, while DML-HMM tries to discover the necessary coupling links between these processes. In comparison, layered HMM [6], switching hidden semi-Markov models [7] and Hierarchical HMM [8] try to model activity at multiple levels, with the upper layers encoding the transitions among the high-level states (such as the constituent actions) and the bottom layer encoding the transitions among the low-level states (such as action primitives). Xiang et al. [9] introduce the multiple observation HMMs that factorize the observation space into several conditional independent factors to recognize activity with large dimensional feature vectors.

As the HMM variants are still restricted by their specific model structure, more general DBNs are also employed for activity modeling. Wu et al. [10] present a DBN that combines RFID and video data to infer the activity and object labels. Their model is essentially a layered HMM with multiple observations. Besides, Laxton et al. [11] define a hierarchical DBN leveraging temporal, contextual and ordering constraints to recognize complex activities.

The model structures of the above approaches, except DML-HMM, are all manually specified. For DML-HMM, only the coupling links are learned from training data. Moreover, these approaches assume that all the activities share the same model structure and sufficient training data are available to learn the models. In contrast, we are able to learn DBN structure for each activity, even when data are insufficient, with logic knowledge exploited from activity domain.

The first-order logics (FOLs) received increasing attention in computer vision due to its expressive power on interpreting knowledge in different domain. Recent researchers [12][13] begin to investigate Markov logic networks (MLN) [14], a combination of FOLs with Markov network, in activity recognition. While MLN successfully integrates logic reasoning with data-driven inference in activity recognition, there are still several points to be considered: first, the MLN can not represent naturally causal relationships between domain elements, which are common in human activity; second, we can view the structure of the MLN as completely specified by the prior knowledge, since the potentials corresponds to the logic groundlings. In case the logic knowledge is inaccurate, the constructed MLN can not work well in activity recognition. In comparison, we choose to represent the domain knowledge with first-order probabilistic logics, and combine these knowledge with training data to learn both the DBN structures and parameters, which can incorporate approximate and partial knowledge.

In knowledge-based learning field, Tong et al. [15] have investigated qualitative constraints for BN parameter learning; however, the qualitative knowledge are expressed heuristically and not exploited for structure learning. In our work, with structure prior and parameter constraints obtained from domain knowledge, we propose a constrained structural EM algorithm to learn DBN structure combining incomplete training data these knowledge. Compared with the structural EM algorithm [16], the constrained structural EM algorithm is different at two aspects: firstly, it can estimate more reliable parameters for the candidate structures under the guide of the constraints; secondly, with the structure prior generated from the domain knowledge, we are able to employ the posterior probability rather than marginal likelihood (BIC) score for model evaluation.

3 Modeling Activity with DBN

3.1 Image Features

The feature set we used for activity recognition consists of the position, speed, shape and spatio-temporal features. For feature extraction, we first perform motion detection to detect the moving object and to extract its silhouette. Position O_Y is then measured as the distance to a reference point[1], speed O_V is evaluated as the change of the object center in pixels and the shape feature O_S includes four elements: aspect ratio of the bounding box of the moving object, filling ratio (the area of the object silhouette with respect to the area of the bounding box) and two first-order moments of the silhouette [9]. The spatio-temporal feature O_{ST} we use is the histogram of optical flow in the spatio-temporal cube.

3.2 DBN Model for Activity Recognition

As we usually observe the activity through object position, shape, speed and spatio-temporal features from the image sequence, the underlying states of these measurements provide a good representation of the activity state space. We can decompose the state X_t into a set of physical states corresponding to position state Y_t, shape state S_t, global speed state V_t and spatio-temporal state ST_t. Accordingly, the measurement O_t consists of four observations: OY_t, OS_t, OV_t and OST_t. Figure 1 shows an example of our DBN model for activity modeling. Besides nodes, there are two types of links in our model: intra-slice links and inter-slice links. While intra-slice links capture the relationships between states, and between states and their corresponding measurements. The inter-slice links

[1] We use the starting position as the reference point for Weizmann dataset and the car position as the reference point for the Parking lot dataset.

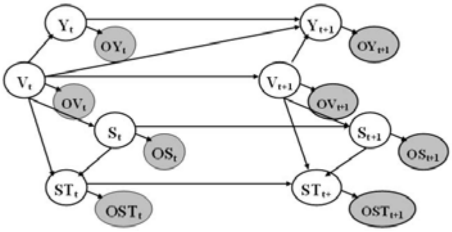

Fig. 1. Example DBN model for activity recognition

capture the dynamic relationships between states at different times. Except for the links between states and their observations, other links are learnt. Please note the links in figure 1 are just for illustration and do not always represent the true dependencies between the underlying states of different features. In next section, we will discuss how to find these dependencies through DBN structure learning. With the above modeling strategy, we can construct one DBN model for each activity and perform activity recognition through finding the model with the highest likelihood, which can be evaluated by the forward propagation of dynamic junction tree algorithm[17].

4 Knowledge Representation in Human Activity

For many computer vision applications, there often exists some approximate domain knowledge that governs the physics, kinematics, and dynamics of domain objects. Such knowledge, if exploited, can help regularize the otherwise ill-posed problems. In activity recognition, we can identify such knowledge in the form of logic rules, which can be feature-related or activity-specific. The simple feature-related low-level rules govern the formulation of most activities. Such rules are activity independent and the same rules can be applied to different types of activities. The activity-specific constraints, on the other hand, are related to the object types, interactions and dynamics for specific activities. In this paper, since the focus is single-level activity recognition, we mainly exploit the feature-related rulesin the form of first-order probabilistic logics, and then try to incorporate these knowledge in our activity model.

4.1 FOPL in Human Activity

First-order probabilistic logics is one type of knowledge representation language preserving the expressive power of first-order logic while introducing the probabilistic treatment of uncertainty. While several families of FOPLs have been proposed in the literatures [18], we keep the formal syntax and semantics defined by Halpern et al. [19].

The alphabet we used to represent the knowledge in human activity includes:

- Predicates: Is;

- Constants: POS(position), SH(shape), SP(speed), ST(spatio-temporal response), near (NR), far (FA), simple (SI), complex(CO), high(HI), low(LO);
- Function: Next;
- Connective symbols: $\vee, \wedge, \forall, \neg, |$;
- Variable: t, AS (denotes one of the three constants: POS, SH and SP), s;
- Probability operator: Pr;
- Basic numeric operator: +, *, =, >;

With the defined alphabet, we can describe the domain elements with two sorts of terms: the object term and numeric term. While the object term describes the non-numeric basic elements (i.e. "t", "shape", "position", "Next(t)") of the domain, the numeric term describes certain probabilities which are rational numbers in the interval [0 1] (i.e. Pr(Is(position, near, t))). Given these elements, we can interpret the logics of the activity domain with a set of well-formed formula, which, in our case, only consists of the relations between different probabilities. The logics we exploit in human activity include:

- Smoothness Logic
  This type of logic interprets the general knowledge about the smooth transitions between the states of the activity, and it is applicable to all states of the activities. *Logic rule*: the object is more likely to keep the previous state than transit to other states.

  $$Pr[Is(AS, s, Next(t)) \mid Is(AS, s, t)] \geq Pr[Is(AS, s, Next(t)) \mid \neg Is(AS, s, t)]$$

  Exemplar Instantiation: the speed of an object at a successive time is more likely to be low if its current speed is low than if its current speed is high.

  $$Pr[Is(SP, LO, Next(t)) \mid Is(SP, LO, t)] \geq Pr[Is(SP, LO, Next(t)) \mid Is(SP, HI, t)]$$

  This logic formula, in simplicity, can be transformed to a probabilistic constraint on the conditional probabilities of our activity model.

  $$P(V_{t+1} = L | V_t = L) \geq P(V_{t+1} = L | V_t = H)$$

  Here L denotes the low speed state and H denotes the high speed state.
- Position-motion Logic
  The position-motion logic encodes the logic relationship between the position and moving speed of the subject.
  Logic rule: The object is more likely to keep the same position state with low speed than with high speed, and meanwhile it is more likely to change position state with high speed than with low speed.

  $$\begin{aligned} &Pr[Is(POS, s, Next(t)) \mid Is(POS, s, t)] \wedge Is(SP, low, t)] \\ &\geq Pr[Is(POS, s, Next(t)) \mid Is(POS, s, t) \wedge Is(SP, high, t)]; \\ &Pr[\neg Is(POS, s, Next(t)) \mid Is(POS, s, t)] \wedge Is(SP, high, t)] \\ &\geq Pr[\neg Is(POS, s, Next(t)) \mid Is(POS, s, t) \wedge Is(SP, low, t)] \end{aligned}$$

Exemplar instantiation: With a high speed and near position in current frame, an object is more probable to be in far position in next frame than with a low speed and near position in current frame.

$$Pr[Is(POS, FR, Next(t)) \mid Is(POS, NR, t)] \wedge Is(SP, HI, t)]$$
$$\geq Pr[Is(POS, FR, Next(t)) \mid Is(POS, NR, t) \wedge Is(SP, LO, t)]$$

Similarly, we can transform this logic formula to a probabilistic constraint on conditional probabilities of the activity model.

$$P(Y_{t+1} = F|Y_t = N, V_t = H) \geq P(Y_{t+1} = F|Y_t = N, V_t = L)$$

Here N denote near position state; F: far position state.

- Shape-motion logic
There are also logic relationships between the shape and speed of the subject
Logic rule: Shape change is more likely to occur when speed is low.

$$Pr[\neg Is(SH, s, Next(t)) \mid Is(SH, s, t)] \wedge Is(SP, low, t)]$$
$$\geq Pr[\neg Is(SH, s, Next(t)) \mid Is(SH, s, t) \wedge Is(SP, high, t)]$$

Exemplar instantiation: It is more probable for an object to change from simple shape to complex shape with a low speed than with a high speed.

$$Pr[\neg Is(SH, CO, Next(t)) \mid Is(SH, SI, t)] \wedge Is(SP, LO, t)]$$
$$\geq Pr[\neg Is(SH, CO, Next(t)) \mid Is(SH, SI, t) \wedge Is(SP, HI, t)]$$

This formula can similarly be transformed to a probabilistic constraints on the activity model, which is:

$$P(S_{t+1} = 1|S_t = 0, V_{t+1} = L) \geq P(S_{t+1} = 1|S_t = 0, V_{t+1} = H)$$

Here $S_t = 1$ denotes complex shape and $S_t = 0$ denotes simple shape.

- Spatio-temporal Logic The spatio-temporal logic encodes the relationship between the spatio-temporal state and the shape change.
Logic rule: It is more probable to have high spatio-temporal response if the object undergoes shape change, than the object stays in the same shape.

$$Pr[\neg Is(ST, high, Next(t)) \mid Is(SH, s, t)] \wedge \neg Is(SP, s, Next(t))]$$
$$\geq Pr[\neg Is(ST, high, Next(t)) \mid Is(SH, s, t) \wedge Is(SP, s, Next(t))]$$

Exemplar instatiation: An object is more likely to have a high spatio-temporal response if it has a simple shape in current frame and a complex shape at next frame, than if its shape at current frame and next frame are both simple.

$$Pr[\neg Is(ST, HI, Next(t)) \mid Is(SH, SI, t)] \wedge \neg Is(SP, CO, Next(t))]$$
$$\geq Pr[\neg Is(ST, HI, Next(t)) \mid Is(SH, SI, t) \wedge Is(SP, SI, Next(t))]$$

The probabilistic constraints transformed from this logic formula is:

$$P(ST_{t+1} = 1|S_t = 0, S_{t+1} = 1) \geq P(ST_{t+1} = 1|S_t = 0, S_{t+1} = 0)$$

Here $ST_t = 1$ is the high spatio-temporal response and $ST_t = 0$ is low spatio-temporal response.

4.2 Incorporate FOPL in Activity Model

The discussion above exploits different types of domain knowledge in the form of FOPL, which can be transformed to a set of qualitative constraints on the conditional probabilities of the activity model. Now we begin to investigate how to incorporate these knowledge in our DBN model. Two types of model prior can be generated from these logic knowledge.

Parameter Constraints. First, the domain knowledge, in terms of qualitative constraints on the model conditional probabilities, can be used to regularize the parameter learning for the activity model. However, they are not necessarily the constraints on the parameters of the activity model. For example, the smoothness logic for the position state finally involves the conditional probability $P(Y_{t+1}|Y_t)$. With the example model structure in figure 1, Y_t is not the only parent of Y_{t+1}, which means $P(Y_{t+1}|Y_t)$ is not a model parameter. Thus, we still need to translate the constraints on state variables into the constraints on the model parameters. For example, if we are expected to impose the following constraint related to the conditional probability $P(A|B)$,

$$P(A = k_1|B = j_1) \geq P(A = k_2|B = j_2) \tag{1}$$

There are three possible cases according to model structure,

- B is the only parent of A: we can directly impose this constraint as $P(A|B)$ is the model parameter;
- B is not the parent of A: we do not impose this constraint as this constraint will become highly nonlinear if we represent the conditional probability $P(A|B)$ using the model parameters. In this case, the logic knowledge will be described by the structure prior, which penalize the absence of the link from B to A by the structure prior.
- B is, but not the only parent of A: Let C be the other parents of A, as $P(A|B) = \sum_C P(A|B,C)P(C|B)$, constraint in equation 1 becomes:

  $$\sum_l P(A = k_1|B = j_1, C = l) \cdot P(C = l|B = j_1) \geq \sum_l P(A = k_2|B = j_2, C = l) \cdot P(C = l|B = j_2)$$

  where l is the configuration of C. Approximating $P(C = l|B = j)$ by the expected sufficient statistics $n_{C=l,B=j}/n_{B=j}$, the above equation becomes a linear constraints on model parameter $P(A|B,C)$.

With the above strategy, the qualitative constraints can be translated to a set of linear constraints on the model parameters θ, denoted as $g_c(\theta) = a_c^T\theta - b_c \leq 0$, where a_c and b_c are the coefficients for constraint c.

Structure Prior. The existing approaches combining logic knowledge with Bayesian network often assume the existence of edge from the conditioning variable to the dependent variable, which can be viewed as hard structural constraints. In our work, we alleviate this hard constraints to a soft structure prior, which can then allow imperfect specification of the domain knowledge to certain degree. The structure prior, together with the training data, are used to learn the model structure in a Bayesian manner as we will discuss in next section.

We set the prior probabilities of the candidate structures through measuring their consistency with the logics, which is defined as:

$$P(S) = ak^{\delta_{S,C}} \tag{2}$$

where a is a normalization constant, k is a constant factor between 0 and 1 controlling the prior strength and $\delta_{S,C}$ is the total number of logic links that are absent from structure S[2]. The intuition for defining this structure prior is to penalize the model structures that are inconsistent with our domain knowledge.

5 Knowledge Based DBN Learning

In this section, we focus on incorporating the domain knowledge in the process of learning the activity model. As the dependencies among the state variables are not apparent, and different activities may have different state dependencies, discovering the DBN structure is a key step for constructing the activity model. In general, the objective of structure learning is searching for a network that fits the best with the prior knowledge and the training data. A complete structure learning scheme requires two components: a criterion to measure how well a candidate structure fits with the prior knowledge and the data, and a model searching strategy used to find the structure with the highest score by the criterion.

5.1 Criterion for Model Selection

A widely used criterion for learning the DBN structure is the BIC score. According to [20], the BIC score $BIC(S)$ can be considered as an approximation of the log marginal likelihood $\log P(D|S)$ of the structure S using Laplacian approximation.

When the prior of the candidate structures is readily available for our activity model, we can learn the model structure in a Bayesian manner, which uses the log posterior probability (LPP) as the criterion for model selection.[3]:

$$\begin{aligned} Q(S) &= \log P(S|D) = \log P(D|S) + \log P(S) - \log P(D) \\ &\approx L(\hat{\theta}_S) + \log(ak^{\delta_{S,C}}) - \frac{d}{2}\log N - \log P(D) \end{aligned} \tag{3}$$

here θ_S is the parameter for structure S , $L(\hat{\theta}_S)$ is the log likelihood of $\hat{\theta}_S$, d is the number of parameters in S, N is the number of samples from all sequences.

[2] A logic link is defined as follows: if the logic constraint finally involves conditional probability $P(A|B)$, link $B \to A$ is a logic link.

[3] Since $\log P(D)$ is a constant, we can ignore it for model comparison.

5.2 Model Search

With incomplete training data, a widely adopted approach for DBN model search is the structural EM (SEM) algorithm [16]. One bottleneck of the SEM algorithm is that it requires a large amount of training sequences. Since the data is often limited, but there exists very generic logic knowledge in terms of qualitative constraints about the human activities, we propose the constrained structural EM (CSEM) algorithm to learn the model structure combining the training data with these constraints.

Before introducing the CSEM algorithm, we define the related notations as follows[4]: θ denotes the parameter of a given DBN structure, $L(\theta) = \log P(D|\theta)$ and $EL(\theta) = E_z[\log P(D, z|\theta)]$ is the log-likelihood and expected log-likelihood of θ respectively, i is the node index, k is the state of node i, n_{ijk} is the expected count of the cases in all the transition slices that node i has the state k with parent configuration j.

Given these definitions, the procedure of the CSEM algorithm is summarized in algorithm 1.

Algorithm 1. Constrained structural EM algorithm

```
For $n = 0, 1, \ldots$ until convergence
E-step
  1. Estimate the parameter $\theta_n$ of the current model structure $S_n$ with the qualitative
     constraints;
  2. Find all the local candidate structures of $S_n$ through adding, removing or re-
     versing one link from $S_n$ (we only change the links between the state nodes and
     do not reverse the temporal links);
  3. "Complete" the data based on $S_n$ and $\theta_n$ and compute the expected counts for
     all candidate structures
  4. For each candidate structure $S$, estimate the parameter $\theta_S$ through maximizing
     the expected log likelihood $EL(\theta_S)$ subject to the constraints;
  5. For each candidate structure $S$, compute the expected LPP score $EQ(S)$
        $EQ(S) = EL(\theta_S) + \log(ak^{\delta_{S,C}}) - \frac{d}{2}\log N - \log P(D)$
M-step
  1. Set $S_{n+1}$ to be the structure with the highest expected score;
```

In E-step 1, we employ the constrained EM (CEM) algorithm to estimate the parameter θ_n for model structure S_n. The E-step of the CEM algorithm is the same as the traditional EM algorithm, which first "complete" the data based on the current parameter and then compute the expected counts $\{n_{ijk}\}$. The M-step of the CEM algorithm finds the new parameters that maximizes $EL(\theta)$ subject to the set of parameter constraints $g_c(\theta) \leq 0$ that we discussed in section 4.2. We formulate this step as a constrained optimization problem:

[4] Strictly the defined terms should depend on a given structure S. we ignore S in the notation just for simplicity.

$$\begin{aligned} \max_{\theta} \quad & EL(\theta) = \sum_i \sum_j \sum_k n_{ijk} \log \theta_{ijk} \\ s.t. \quad & \sum_k \theta_{ijk} = 1 \ \forall i, j \quad , \quad g_c(\theta) \leq 0 \ \forall \text{ constraint } c \end{aligned} \tag{4}$$

In E-step 4, we need to estimate the parameter θ through maximizing the expected log-likelihood $EL(\theta_S)$ for each candidate structure S. Since the expected counts $\{n_{ijk}\}$ are available from E-step 3, we can also estimate θ_S through solving the optimization problem in equation 4.

With the CSEM algorithm, the logic knowledge can influence the expected score $EQ(S)$ of the candidate structures in two ways: first they control the prior probabilities of the structures; secondly, they can regularize the parameter estimation for each structure and then alter the expected log-likelihood score. When the training data is limited, adding the structure prior or regularizing the parameter estimation can help improve structure learning by avoiding some local maxima caused by the noisy data in structure search process.

The CSEM algorithm is guaranteed to achieve a local optimum since it improves the model score ($Q(S)$) at each step. The proof of convergence is similar to the SEM algorithm with small difference on handling the structure prior.

5.3 Learning Activity-Dependent DBNs

People usually assume all the activities share the same model structure; the real activities, however, do not have the same dependency among the basic states. For example, the dependency between the shape and speed varies from activity to activity. People usually keep similar shape in *walking*, so the dependency between the speed and shape is weak. In comparison, this dependency is strong for *bending* as people usually undergoes large shape variation during the bending process. Thus, we learn both the model structure and parameter which capture the dependency type and strength for each activity respectively.

6 Experiments

6.1 Weizmann Dataset

The weizmann dataset contains 10 different behaviors performed by 9 people. There are total of 93 video sequences. In the experiments, we learn the DBN models with different number of training sequences (1, 3, 5, 8). The knowledge base used include 8 smoothness logic groundings, 4 position-motion logic groundings, 4 shape-motion logic groundings and 4 spatio-temporal logic groundings.

Evaluation on Activity-Dependent Structure. Figure 2 compares the activity recognition performance of activity-independent model and activity-dependent model on Weizmann dataset. We can find that the activity-dependent model outperforms activity-independent model almost in all cases (with 1 training sequences the performance is quite close). We also include the results for

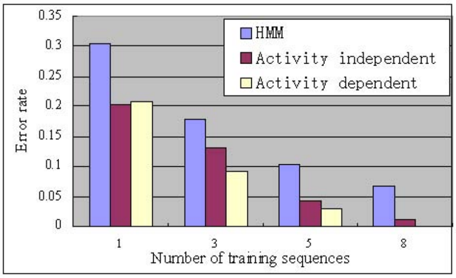

Fig. 2. Comparison of activity-independent DBN, activity-dependent DBN and HMM on Weizmann dataset

HMM with exactly the same set of features. It is easy to get from figure 2 that our DBN model outperforms the baseline HMM model significantly through explicitly modeling the dependencies among different features.

Evaluation on CSEM for Activity-Dependent Structure Learning. In table 1, we report the recognition results of the knowledge-based CSEM algorithm and the data-based SEM algorithm. With 5 and 8 training sequences, the advantage of CSEM algorithm over SEM algorithm is not significant since we have already obtained nearly perfect recognition result with SEM algorithm. However, when data becomes scarce, the CSEM algorithm gradually shows its superiority over SEM algorithm. In case of 1 training sequence, the activity-dependent model learned with CSEM algorithm outperforms the model learnt with SEM algorithm by 6.5% with same set of image features.

Comparison with Other Approaches. Since the results reported by the state-of-art approaches on Weizmann dataset are evaluated using leave-one-out cross validation, it is hard to compares our algorithm with them in the case of insufficient data. Thus, we compare our result with these approaches using 8 training sequences for each activity. Table 2 shows the comparison of our work with previous approaches. Our activity-dependent DBN models achieve the state-of-art performance on Weizmann dataset.

6.2 KTH Dataset

The KTH dataset consists of 600 video clips with 6 human activities, each of which is performed by 25 subjects in four different scenarios: outdoors, outdoors

Table 1. Recognition error of activity-dependent structures learned with CSEM and SEM

# Training sequences	1	3	5	8
SEM	0.247	0.091	0.028	0.000
CSEM	0.182	0.067	0.019	0.000

Table 2. Comparison with previous work on Weizmann dataset

Our method (SEM)	100%
Our method (CSEM)	100%
Fathi et al. [21]	100%
Jhuang et al. [22]	98.8%
Thurau et al. [23]	94.4%
Niebles et al. [24]	72.8%

with scale variation, outdoors with different clothes and indoors. The knowledge base we used in evaluating our approach on this dataset is exactly the same as on the Weizmann dataset. In the experiments, we vary the number of training sequence for each activity from 50 to 500 to study the effectiveness of the knowledge-based learning on alleviating the dependency on the data.

Table 3 compares the knowledge-based CSEM algorithm with the standard SEM algorithm in learning the activity model with different number of training subjects. We can clearly see that, when the number of training subjects is large, CSEM is only marginally better than SEM algorithm. However, when the number of training subjects becomes smaller, the knowledge we exploited gradually play more important role in activity recognition. With the complement of the logic knowledge, the CSEM algorithm can perform significantly (7.1%) better than the SEM algorithm when the number of training subjects is small.

Table 3. Comparison of CSEM and SEM

# Training Subjects	4	8	12	16	20
EM	0.760	0.828	0.862	0.880	0.892
CSEM	0.831	0.863	0.904	0.921	0.925

We also compare our approach with the state-of-art approaches on this dataset. Similar to the posted results in the literature, I use the data from 16 subjects for training. Table 4 shows that we can achieve comparable result to the state-of-art approaches.

Table 4. Comparison with previous works on KTH dataset

Our method (SEM)	88.0%
Our method (CSEM)	92.1%
Yuan et al. [25]	93.3%
Laptev et al. [26]	91.8%

6.3 Parking Lot Dataset

We also apply our algorithm to the problem of recognizing human activities in the parking lot. The dataset consists of 108 sequences for 7 activities: *walking*

(WK), *running* (RN), *leaving car* (LC), *entering car* (EC), *bending down* (BD), *throwing* (TR) and *looking around* (LA). These activities are performed by several people with scale variation, view change and shadow interference. In the experiment, we randomly split the original dataset into training set and testing set. Different algorithms are compared using training set with 10, 20, 40, 80 sequences. Each size is tested 10 times and the average recognition error is used for evaluation. We use the constraints set as those for the Weizmann dataset.

In figure 3, we compare the knowledge-based CSEM with data-based SEM algorithms in learning both activity-dependent and activity-independent model structures.

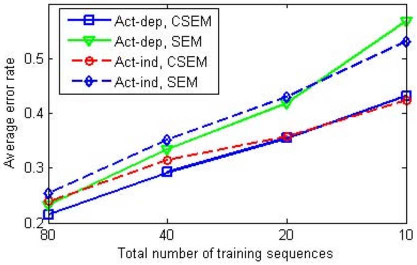

Fig. 3. Comparison of CSEM and SEM for learning activity-dependent and activity-independent models

First, we look at the performance of the activity-dependent models learnt with the CSEM algorithm and SEM algorithm. As the number of training sequences decreases, the CSEM algorithm gradually shows its advantage over SEM, which means our knowledge in terms of constraints play more and more important roles on regularizing the structure learning as data size decreases.

From figure 3, we can also find that, with 20 or 10 training sequences, the activity-dependent model obtains comparable results with activity-independent model learnt using CSEM with the same data size, while it performs worse if we learn the structure without constraints. Moreover, the activity-dependent model with CSEM learning (method 1) requires only half training data to obtain comparable result to activity-independent model with SEM learning (method 2) when the data is insufficient. Specifically, with only 10 training sequence, the recognition error of method 1 is 43.2%, while the recognition error of method 2 is 43.0% given 20 training sequence. With 20 training sequence, the recognition error of method 1 is 35.5%; in comparison, the recognition error of method 2 is 35.2% given 40 sequences. Thus, we can see that exploiting the generic logic knowledge in the activity can greatly alleviate the problem of insufficient data.

Table 5 reports the recognition result of the activity-dependent models learnt with CSEM algorithm on 80 training sequences. Our algorithm can correctly classify 78.6% of the testing sequences. The result is reasonable since the misclassifications occur between similar activities (i.e. *walking* and *looking around*), or for the activities with poor observation (i.e. *leaving car* and *entering car*)

Table 5. Confusion table of the activity recognition test on activity-dependent models learnt with constraints

	WK	RN	LC	EC	BD	TR	LA
WK	.90	.06	.00	.00	.00	.00	.04
RN	.08	.88	.00	.00	.00	.04	.00
LC	.00	.00	.65	.25	.10	.00	.00
EC	.00	.00	.35	.60	.00	.05	.00
BD	.02	.00	.04	.00	.80	.08	.06
TR	.00	.10	.00	.04	.12	.72	.02
LA	.125	.025	.025	.00	.025	.05	.75
	Overall Accuracy: 78.6%						

7 Conclusion

In this paper, we focus on exploiting prior knowledge from human activity domain and investigating a constrained structure learning method to learn activity model combining these prior knowledge with training data. Our contributions include : first, we exploit various generic while effective domain knowledge in the form of first-order probabilistic knowledge; second, after transforming the FOPLs to the structure prior and qualitative parameter constraints, we propose a constrained DBN learning approach to combine domain knowledge with training data. The experimental results demonstrate the effectiveness of our knowledge-based learning scheme in reducing the dependence on training data and alleviating the over-fitting problem when data is insufficient. It also shows promise of the activity-dependent structures in improving activity recognition. Although our learning framework is only tested on single-subject activity recognition, we are planning to apply it to multi-subject and more complex activity recognition in the future.

Acknowledgement

This project is supported in part by Defense Advanced Research Projects Agency (DARPA)'s Video and Image Retrieval and Analysis Tool (VIRAT) program under Contract No. HR0011-08-C-0135 via Kitware.

References

1. Yamato, J., Ohaya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden markov model. In: CVPR (1992)
2. Zhang, D., Perez, D., McCowan, I.: Semi-supervised adapted hmms for unusual event detection. In: CVPR (2005)
3. Vogler, C., Metaxas, D.: A framework for recognizing the simultaneous sspects of american sign language. In: CVIU (2001)

4. Oliver, N.M., Rosario, B., Pentland, A.P.: A bayesian computer vision system for modeling human interations. PAMI (2000)
5. Xiang, T., Gong, S.: Beyond tracking: Modelling activity and understanding behavior. In: IJCV (2006)
6. Oliver, N., Horvitz, E., Garg, A.: Layered representation for human activity recognition. CVIU (2004)
7. Duong, T., Bui, H., Phung, D.: Activity recognition and abnormality detection with the switching hidden semi-markov model. In: CVPR (2005)
8. Lv, F., Nevatia, R.: Single view human action recognition using key pose matching and viterbi path searching. In: CVPR (2007)
9. Xiang, T., Gong, S.: Video behavior profiling for anomly detection. PAMI (2008)
10. Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., Regh, J.: A scalable approach to activity recognition based on object use. In: ICCV (2007)
11. Laxton, B., Lim, J., Kriegman, D.: Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video. In: CVPR (2007)
12. Tran, S., Davis, L.: Event modeling and recognition using markov logic networks. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 610–623. Springer, Heidelberg (2008)
13. Biswas, R., Thrun, S., Fujimura, K.: Recognizing activities with multiple cues. In: IEEE Works. on Human Motion (2007)
14. Richardson, M., Domingos, P.: Markov logic networks. In: Machine Learning (2006)
15. Tong, Y., Ji, Q.: Learning bayesian network with qualitative constraints. In: CVPR (2008)
16. Friedman, N.: The bayesian structural em algorithm. In: UAI (1998)
17. Murphy, K.: Dynamic bayesian networks: representation, inference and learning. Ph.D. dissertation, University of California (2002)
18. Milch, B., Russell, S.: First-order probabilistic languages: Into the unknown. In: ILP (2006)
19. Halpern, J.: An analysis of first-order logics of probability. Artificial Intelligence (1990)
20. Heckerman, D.: A tutorial on learning with bayesian networks. Learning in Graphical Models (1999)
21. Fathi, A., Mori, G.: Action recognition by learning mid-level motion features. In: CVPR (2008)
22. Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A biologicallyinspired system for action recognition. In: ICCV (2007)
23. Thurau, C., Hlavac, V.: Pose primitive based human action recognition in videos or still image. In: CVPR (2008)
24. Niebles, J., Fei-Fei, L.: A hierarchical model of shape and appearance for human action classification. In: CVPR (2008)
25. Yuan, J., Liu, Z., Wu, Y.: Discriminative subvolume search for efficient action detection. In: CVPR (2009)
26. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human acions from movies. In: CVPR (2008)