Chapter 1

Activity Modeling and Recognition
Using Probabilistic Graphical Models

Xiaoyang Wang, Zhi Zeng, Qiang Ji

Rensselaer Polytechnic Institute,
110 Eighth Street, Troy, NY, USA 12180
wangx16@rpi.edu, zengz@rpi.edu, jiq@rpi.edu

In this chapter, we introduce the approaches for visual activity modeling and recognition using the time-sliced probabilistic graphical models, i.e. hidden Markov models (HMM) and dynamic Bayesian networks (DBN). Firstly, we briefly introduce the features used for activity modeling and recognition. Then, three kinds of DBNs, namely, the Generative DBN (GDBN), the Discriminative DBN (DDBN) and the first-order Probabilistic Logics knowledge DBN (PLDBN) are introduced for activity modeling and recognition. The GDBN is a standard generative time-sliced graphical model often used for activity recognition. We also incorporate prior knowledge into the GDBN classification procedure to improve the system performance. The DDBN employs a discriminative parameter learning approach for DBNs so that the criterions during training and testing are consistent. Empirical studies can show the proposed discriminative learning approach outperforms the maximum likelihood or EM algorithm for different activity recognition tasks. The PLDBN provides solutions on learning 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. The experimental results on PLDBN demonstrate simple logic knowledge can compensate effectively for the shortage of the training data and therefore reduce our dependencies on training data.

1.1. Introduction

Modeling and recognizing visual activities from videos is becoming one of the most promising applications in the computer vision field. It is fundamental to machine visual surveillance which attracts a growing interest from both academia and industry over the past years. Various graphical, syntactic, and description-based approaches\textsuperscript{1} have been introduced for modeling and understanding visual activities. Among those approaches, the time-sliced graphical models, i.e. Hidden Markov Models (HMMs) and Dynamic Bayesian Networks (DBNs), have become the most popular tools. This is not surprising, because graphical models match the nature
of activity recognition problem well. These graphical models offer clear Bayesian semantics, and provide expressive representation and propagation of uncertainty over a sequence of video frames.

Existing approaches for visual activity recognition can be mainly classified into two categories: the generative approach and discriminative approach. The generative approaches try to model the joint probability of the activity label and the evidence, and perform activity classification based on the posterior probability with a Bayesian decision scheme. Discriminative approaches, on the other hand, directly model the posterior probability distribution of the class label given the evidence, which avoid the intermediate step to estimate the joint distribution. Common HMMs and DBNs for visual activity recognition are generative models. The usual way of activity recognition with these generative HMMs or DBNs is first to learn one DBN/HMM model for each activity independently through maximum likelihood estimation (MLE), or expectation maximization (EM) if given incomplete training data. And then, the models perform the classification decision through identifying the activity with the highest likelihood given the observations extracted from the raw video, or equivalently, the activity class with the highest posterior probability assuming there is uniform prior over all activities. In this chapter, our generative DBN model for activity recognition is first introduced. This generative DBN model is later combined with prior knowledge using a Bayesian network. The vehicle detection result is an useful prior knowledge that can be used for activity recognition. Based on the output of the vehicle detector, we can compute the new posterior probability given both the evidence and the vehicle detection result. Classification based on this posterior probability performs better than the classification using only the likelihood outputted by the generative DBN model.

The generative approach for HMM or DBN based activity recognition produces a discrepancy between the learning objective and the classification criterion. The models learned by maximum likelihood or EM algorithm for each activity, though capturing the data dependency with the corresponding activity well, may not maximize the classification accuracy between different activities. Our solution to reduce this discrepancy between the learning and classification objective for activity classification is to learn the generative models discriminatively. The learning procedure would maximize the conditional likelihood and ensure a consistent criterion during learning and classification. In this chapter, we will compare our models trained on generative learning and discriminative learning respectively. Experiments on KTH data set\textsuperscript{2} showed that, discriminative learning works better than generative learning even for a generative model.

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 alleviating 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. In our work, we present a framework to learn the DBN model combining training data with domain knowledge represented with the first order probabilistic logic.

In this chapter, we will first introduce the activity modeling with our DBN models in section 1.2. The features for activity modeling and recognition will be briefly introduced in this section too. Then, in sections 1.3, 1.4 and 1.5, we will introduce our generative DBN model combined with prior knowledge, our discriminative learning method for the DBN model, and our algorithm of utilizing first order probabilistic logics to guide the DBN learning process respectively. The experiments of these models and methods are discussed in section 1.6. The chapter ends in section 1.7 with a summary.

1.2. Activity Modeling using DBNs

1.2.1. Data Input for Activity Modeling

The raw data of activities and events are presented in the format of video tracks. A single track and even various examples of the same event can have an arbitrary number of detections. So, in order to enforce a level of consistency for all events, the tracks are partitioned into track intervals during training and testing. A track interval consists of a predefined number of frames and are represented by the corresponding features on each frame. The features used to characterize each event include the position, speed, shape and appearance features.

1.2.2. Feature Extraction and Feature Selection

The features we used for activity recognition includes the position, speed, shape and the appearance features. For feature extraction, we first detect the moving object and extract its silhouette, and then measure the position, speed and shape of the object based on the silhouette. The position feature $O_Y$ is then measured as the distance to a reference point. The speed feature $O_V$ is evaluated as the change of the object center, 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.

The appearance features $O_A$ are selected from a feature pool consisting of the HoG (histogram of oriented gradient) and HoF (histogram of optical flow) features. Here, an Adaboost algorithm is executed on all features in the feature pool, where a decision stump is used as a weak classifier. Since the decision stump chooses the single most discriminative feature that minimizes the overall training error, a count of the selected features results in a ranking of the most discriminative features.
upon completion of Adaboost. The top 20 most discriminative appearance features $O_A$ are selected on a per event basis and are used as the continuous inputs to the observation nodes of their corresponding model.

In one of our DBN model which incorporates first-order probabilistic logics knowledge, we also used spatio-temporal feature $O_{ST}$ which is the histogram of cuboid types.

1.2.3. DBN Models

We build three different DBN models for the activity recognition task. The first DBN model, which is called the Generative DBN (GDBN), includes two hidden nodes $GM$ and $SA$ respectively in the first layer. The $GM$ node represents the global motion state, and the $SA$ node represents the appearance and shape state. These two states give a factorized representation of the subject state space. The second layer consists of two measurement nodes $OGM$ and $OSA$. The $OGM$ node denotes the combination of position feature $O_Y$ and speed feature $O_V$ which are all global motion measurements. The $OSA$ node denotes the appearance features $O_A$ selected from the raw HoG and HoF features using the Adaboost algorithm, and the shape feature $O_S$. Besides the nodes, there are two types of links in the model: intra-slice links and inter-slice links. The intra-slice links couple different states of the subject to encode their dependencies. And the inter-slice links represent the temporal evolution and capture the dynamic relationships between states at different time. Figure 1.1 shows the structure of GDBN model.

![Fig. 1.1. The Generative DBN model (GDBN) for activity modeling and recognition.](image)

The second DBN model is called the Discriminative DBN (DDBN), as shown in figure 1.2. It separates the global motion measurements into shape and motion for inputs into their own observation nodes, resulting in three hidden nodes per time slice ($S$, $A$ and $V$) and corresponding measurement states ($OS$, $OA$ and $OV$); but more importantly, it is trained in a discriminative fashion rather than generatively. Typically the model of a particular event is only trained with data from the event of interest using likelihoods; but is tested relative to all events using posterior probabilities. This discrepancy results in non-ideal models for each event given the testing method. To overcome this discrepancy, the DDBN model is trained in a discriminative manner, where each event’s model is trained with the data from all
events of interest. Promising results are obtained here using the KTH database.

Our third DBN model incorporates the first-order probabilistic logics knowledge (PLDBN). In this PLDBN model, we can decompose the object state space into a set of physical states corresponding to position state $Y$, shape state $S$, global speed state $V$ and spatio-temporal state $ST$. Accordingly, the measurement $O$ consists of four observations: $O_Y$, $O_S$, $O_V$ and $O_{ST}$. Figure 1.3 shows an example of our DBN model for activity modeling.

With the above different modeling strategy and different DBN model structures, we can construct one DBN model for each activity and perform activity recognition through finding the model with the highest likelihood, or finding the model with the highest posterior probability if given prior on the activity classes. Both of them can be evaluated by the forward propagation of dynamic junction tree.

1.3. The Generative DBN Model (GDBN) for Activity Recognition

1.3.1. **GDBN Model Description**

As shown in figure 1.1, the GDBN model use the position ($O_Y$), speed ($O_V$), shape ($O_S$) and appearance ($O_A$) features to recognize different activities. The $O_Y$ and
$O_V$ features are the measurement of the node $GM$ in the GDBN model, and are represented as the observation node $OGM$ in figure 1.1. Also, the $O_S$ and $O_A$ features are the measurement of node $SA$ in the GDBN model, and are represented as the observation node $OSA$ in the GDBN structure.

The GDBN models the temporal progression of the kinematic and image features for each activity. The collection of trained graphical models consists of all known activities that have enough training data, as well as, one “unknown activity” class; which together form a database of models. An unknown temporal track interval is tested against all models in this database and is assigned the activity id that corresponds to the model with the best fit to the track’s feature vector.

The best fit can be the activity with the highest likelihood given corresponding activity model, or the highest posterior probability using additional prior information. Here, we added the vehicle prior information acquired from pre-assumed vehicle detector output. The system performance can be improved using this posterior probability calculated from both the vehicle prior and the GDBN outputted likelihood.

1.3.2. GDBN Parameter Learning for Activity Recognition

Suppose we have $K$ different kinds of activities to recognize in total. To recognize these $K$ kinds of activities, we need to build $K$ GDBN models. Let the parameters of the $K$ GDBN models to be $\Theta_1, \Theta_2, \ldots, \Theta_K$ respectively, and the training sequences (extracted features) for activity $c$ is $E_c$ which contains $N_c$ sequences, i.e. $E_c = (E_1, E_2, \ldots, E_{N_c})$. As DBNs usually captures the joint distribution of sequence of variables, it is typically learned by maximizing the log likelihood of all training sequences. Let the learned parameter for the GDBN model of activity $c$ to be $\hat{\Theta}_c$, then we can write the general learning principle as:

$$\hat{\Theta}_c = \arg \max_{\Theta} P(E|\Theta) = \arg \max_{\Theta} \sum_{E \in E_c} \log P(E|\Theta) \quad (1.1)$$

In the GDBN model, there are two hidden nodes $GM$ and $SA$ that have two and three states respectively. Between the hidden nodes in different time slices, there are three kinds of parameters to learn: the initial state distribution for the $GM$ node, the intra-slice state conditional probability of $SA$ given $GM$, and the inter-slice state transition probability both from $GM_t$ to $GM_{t+1}$, and from $SA_t$ to $SA_{t+1}$. Our observations $OGM$ and $OSA$ are all continuous observations. Given their correspondent hidden states, we suppose the observations of $OGM$ and $OSA$ follow Gaussian distribution. Thus, for each state of the $GM$ and $SA$ nodes, we have a mean vector $\mu$ and the covariance matrix $\Sigma$ to estimate.

For the parameter learning, the data for parameter estimation is incomplete because we do not have direct observations of the hidden state status for each time slice $t$. As a result, the Expectation Maximization (EM) method for DBN models is adopted to estimate the parameters from the incomplete data. First, the
initialization step is taken to give a starting value of the above parameters for the EM procedure. The initial state distribution and the state transition matrices are all initialized randomly. Since the EM algorithm is very sensitive to the initialization of the observation node parameters, the initialization process uses the K-means clustering algorithm to identify the starting mean vector \( \mu \) and the covariance matrix \( \Sigma \) for each state. After the parameter initialization step, the GDBN parameters are learned by using a maximum of 20 iterations through a DBN EM where a junction tree inferencing engine is used for the expectation step.

With the EM method, we can get the parameters of \( K \) models for the \( K \) activities to be recognized. Here, we denote the learned \( K \) models to be \( \hat{\Theta}_1, \hat{\Theta}_2, \ldots, \hat{\Theta}_K \).

### 1.3.3. Activity Recognition with the Likelihood of GDBN

The input to the usual activity recognition model is a sequence of video with the track information which includes the starting and ending time of the activity, and the bounding box of the detected objects for this activity. During testing, we extract the same features from the above video and track information. For GDBN inference, suppose the evidence of the testing sequence to be \( E_T \).

We can obtain the likelihood by using the forward propagation of the dynamic junction tree. For the \( c \)-th model where \( c \in [1, K] \), denote the likelihood of the evidence \( E_T \) as: \( P(E_T | \hat{\Theta}_c) \). The evidence is best fitted to the model with the maximum likelihood. Thus, the classification result \( \hat{c} \) can be written as:

\[
\hat{c} = \arg \max_{c \in [1, K]} P(E_T | \hat{\Theta}_c) \tag{1.2}
\]

This classification criterion picks the activity class with the highest likelihood, or equivalently, the activity class with the highest posterior probability assuming there is uniform prior over all activities.

### 1.3.4. Activity Recognition of GDBN combined with Prior Knowledge

In the activity recognition problem, there is always some kind of objects that may appear in the activity and would help us identify the activity. For example, for classifying the human activities of Walking and Approaching a Vehicle, the appearance of a vehicle in the activity bounding box may be an important extra information. But, this information is not thoroughly reflected in our extracted image and kinematic features. On the other hand, for the video sequence itself, it is usually feasible to detect the vehicles from the rest contents in the image. If we take this vehicle detection information as a prior knowledge, and incorporate this prior knowledge into the posterior probability that we will use to classify the input evidence, we can utilize this additional vehicle information systematically.

To describe the causal relationships between the activity class, the physical appearance of a vehicle, the detection of the physical appeared vehicle, and the
evidence of the activity sample that we need to classify, we can build up a Bayesian network as shown in figure 1.4. We call this Bayesian network the Vehicle Prior Bayesian Network (VPBN).

![Figure 1.4. The Bayesian network describing the causal relationships when the vehicle detection prior knowledge is incorporated.](image)

In VPBN, the activity node $A$ is the root node of this Bayesian network. It has $K$ discrete values where $K$ stands for the $K$ different kinds of activities to recognize. The node $V$ is the “vehicle” node. Thus, it has two values “1” and “0”, where “1” stands for the physical presence of a vehicle in the activity, and “0” stands for no vehicle presence in the activity. For different activities, the probability of the presence of a vehicle is different. The video sequence of the activity *Approaching a Vehicle* would have a much higher probability for the presence a vehicle than the video sequence of the *Walking* activity. And, the link from node $A$ to node $V$ would reflect the probability of the presence (or non-presence) of a vehicle given the value $c \in [1, K]$ of node $A$.

The node $VD$ stands for the “vehicle detection”. In theory, we can have two different vehicle detection results as “Vehicle Detected” and “No Vehicle Detected” for a single video sequence. The link from $V$ to $VD$ would reflect the probability of the vehicle detection when there exist or do not exist a vehicle. We set two values “1” and “0” for the node $VD$. “1” stands for the “Vehicle Detected”; and “0” stands for “No Vehicle Detected”. The link from node $V$ to node $VD$ is quantified by $P(VD|V)$, which characterizes the accuracy of the vehicle detector.

The node $E$ is the evidence node which is continuous. The link from node $A$ to node $E$ reflects the probability of evidence $E$ given activity class $A$. This is the same with the likelihood we estimated in section 1.3.3.

The VPBN can incorporate the vehicle detection prior knowledge with the existing likelihood provided by GDBN, and they two would provide us an improved posterior probability for classification. Here, we will briefly introduce the learning of VPBN parameters at first, and then discuss the inference with VPBN to get the posterior probability of the activity given the evidence and vehicle detection result, i.e. $P(A|E, VD)$, or the posterior probability $P(A|E)$ when the vehicle detector is not included in the system.
1.3.4.1. VPBN Parameter Learning

In VPBN, we have four sets of parameters to estimate. They are the prior probability of root node $A$ and the three conditional probabilities represented by the three links in the graphical model shown in figure 1.4. The prior probability of the root node $A$ is assumed to be uniform. And the conditional probability of the node $E$ given $A$ is actually the likelihood outputted by GDBN, as analyzed before. The conditional probability $P(V|A)$ can be learned by estimating the vehicle appearance rate given all the training sequences for every activity. This estimation is straightforward because the data for both two nodes are complete.

The conditional probability $P(VD|V)$ reflects the performance of the applied vehicle detector. $P(VD|V)$ has two independent parameters in its conditional probability table, i.e., $P(VD = 1|V = 0)$ and $P(VD = 1|V = 1)$. In fact, $P(VD = 1|V = 0)$ reflects the false alarm rate (FAR) of the vehicle detector, and $P(VD = 1|V = 1)$ is the positive detection rate (PD). To estimate these two parameters, we can first evaluate the performance of the applied vehicle detector in the data set with the FAR and PD. And then take the evaluated FAR and PD as the learned VPBN parameters. Here, we denote $\alpha = P(VD = 1|V = 1)$ and $\beta = P(VD = 1|V = 0)$ for convenience.

1.3.4.2. VPBN Inference with Vehicle Detector

For the activity classification, we need the posterior probability of the activity given the evidence and the result of a vehicle detection, i.e., $P(A|E, VD)$. Using the VPBN in figure 1.4, this probability can be calculated as:

$$
P(A|E, VD) = \frac{P(A, E, VD)}{P(E, VD)} = \frac{\sum_A P(A)P(E|A)P(V|A)P(VD|V)}{\sum_A \sum_V P(A)P(E|A)P(V|A)P(VD|V)} \quad (1.3)
$$

In section 1.3.4.1, we have already assumed that the prior probability $P(A)$ is uniform. Considering the nodes $V$ and $VD$ both have only two values, and we have defined $\alpha = P(VD = 1|V = 1)$ and $\beta = P(VD = 1|V = 0)$. Thus, the equation 1.3 can be:

$$
P(A|E, VD = 1) = \frac{P(E|A)[\alpha P(V = 1|A) + \beta P(V = 0|A)]}{\sum_A P(E|A)[\alpha P(V = 1|A) + \beta P(V = 0|A)]}
$$

$$
P(A|E, VD = 0) = \frac{P(E|A)[(1 - \alpha)P(V = 1|A) + (1 - \beta)P(V = 0|A)]}{\sum_A P(E|A)[(1 - \alpha)P(V = 1|A) + (1 - \beta)P(V = 0|A)]} \quad (1.4)
$$

Equation 1.4 can be used to calculate the conditional probability $P(A|E, VD)$ for all $A$ values. And the calculated probability is in $[0,1]$ space.

With the vehicle detection result, the classification result $\hat{c}$ can be written as:

$$
\hat{c} = \arg \max_{c \in [1,K]} P(A = c|E, VD) \quad (1.5)
$$
where \( P(A = c|E, V D) \) is calculated in equation 1.4. This criterion picks the model with the maximum posterior probability \( P(A|E, V D) \).

### 1.3.4.3. VPBN Inference without Vehicle Detector

When the vehicle detector is not included in the system, we recognize the activity using the posterior probability \( P(A|E) \), which is the probability of the activity given the evidence. Calculation of this posterior probability can be unified into the VPBN model, as derived in the following:

\[
P(A|E) = \frac{\sum_V \sum_{V D} P(A, E, V, V D)}{\sum_A \sum_V \sum_{V D} P(A, E, V, V D)}
\]  
\[
(1.6)
\]

The probability \( P(A, E, V, V D) \) in equation 1.6 can be decomposed in chain rules w.r.t the VPBN model as shown in figure 1.4:

\[
\sum_V \sum_{V D} P(A, E, V, V D) = \sum_V P(A) P(E|A) P(V|A) P(V D|V)
\]

\[
= P(A) P(E|A) \sum_V \sum_{V D} \{ P(V|A) P(V D|V) \} = P(A) P(E|A) \{ \alpha P(V = 1|A) + (1 - \alpha) P(V = 0|A) \} + \beta P(V = 0|A) + (1 - \beta) P(V = 0|A) \}
\]

\[
= P(A) P(E|A)
\]  
\[
(1.7)
\]

Substitute equation 1.7 into equation 1.6, we have from VPBN model that:

\[
P(A|E) = \frac{P(A) P(E|A)}{\sum_A P(A) P(E|A)} \propto P(A) P(E|A)
\]

\[
(1.8)
\]

We can pick the activity model with the maximum posterior probability \( P(A|E) \) as the classification result.

If we still assume that the prior probability \( P(A) \) is uniform, the posterior probability \( P(A|E) \) given by VPBN model would be proportional to the likelihood of GDBN. In this case, the activity classification of VPBM without vehicle detector would be the same as activity recognition with the likelihood of GDBN.

### 1.4. The Discriminative DBN Model (DDBN) for Activity Recognition

As mentioned above, in classification, our decision is based on comparing the likelihoods or posterior probabilities of different models, so these models are not independently applied. However, the traditional way to learn activity models is to learn the parameters of each DBN model independently through maximum likelihood estimation or EM algorithm. Though ensuring a representative model for each activity, it in general can not guarantee the best performance in classification, because of the discrepancy between the training criterion and test criterion. In this section, we introduce a discriminative learning approach to learn the DBN models for all
activity models together, which can guarantee better classification performance on the training set. This Discriminative DBN Model (DDBN) is shown in figure 1.2.

1.4.1. Discriminative Learning Formulation

The goal of classification is to predict the class label \( c \) given the evidence \( E \). Under the Bayesian decision framework, the optimal prediction for data \( E \) is the class that maximizes \( P(c|E) \). In activity recognition, as we can evaluate the likelihood \( P(E|c) \) for each activity \( c \), we can compute the posterior probability based on Bayesian theorem.

\[
P(c|E) = \frac{P(E|c)P(c)}{\sum_{c'} P(E|c')P(c')}
\]

(1.9)

If we do not have prior on the activities, \( p(c|E) \) becomes the normalized likelihood.

As DBN usually captures the joint distribution of sequence of variables, it is typically learned by maximizing the log likelihood of all training sequences.

\[
\hat{\Theta} = \arg \max_{\Theta} P(E|\Theta) = \arg \max_{\Theta} \sum_{n} \log P(E^n|\Theta)
\]

Here \( E = (E^1, E^2, \ldots, E^N) \) denotes all \( N \) training sequences.

For the activity recognition problem, with generative learning, we learn the parameters \( \hat{\Theta}_c \) of each activity model \( c \) independently through maximizing its likelihood (or expected likelihood).

\[
\hat{\Theta}_c = \arg \max_{\Theta} P(E_c|\Theta) = \arg \max_{\Theta} \sum_{E \in E_c} \log P(E|\Theta)
\]

where \( E_c \) denotes the training sequences for activity \( c \). In this way, we can ensure to obtain a representative model for each activity, but it cannot guarantee the best performance in classification, since the objective function of the maximum likelihood learning is not consistent with our prediction criterion \( P(c|E) \). Hence, a better objective function for learning DBNs for activity recognition would be the conditional log likelihood \( CLL(C|E) \).

\[
CLL(C|E) = \sum_{n=1}^{N} \log P(c^n|E^n) = \sum_{c} CLL(c|E_c) = \sum_{c} \sum_{E \in E_c} \log P(c|E)
\]

where \( C = (c^1, c^2, \ldots, c^N) \) denote the activity labels of all \( N \) training sequences, \( c \) are the labels of the sequence in \( E_c \) (actually, \( c \) is a vector of \( c \)'s).

Maximizing the conditional likelihood is not trivial since the \( CLL \) objective is non-convex in general. However, we can optimize it locally through gradient search. In this chapter, a Quasi-Newton method called BFGS with the line search under Armijo rule\(^\text{11} \) is employed to perform the optimization.

A key step for the optimization is to evaluate the gradient of the \( CLL \). In general, for sample \((E, c)\), the gradient of \( CLL \) with respect to model parameter \( \Theta \)
is

\[ \frac{\partial \log P(c|E)}{\partial \Theta} = \frac{\partial \log [P(E|c)P(c)]}{\partial \Theta} - \frac{\partial \log P(E)}{\partial \Theta} \]

\begin{align*}
= \frac{\partial \log P(E|c)}{\partial \Theta} - \frac{\partial \log P(E)}{\partial \Theta}
\end{align*}

(1.10)

Please note that the second term of equation 1.10 can be evaluated as the expectation of the first term,

\[ \frac{\partial \log P(E)}{\partial \Theta} = E_{P(c|E)}[\frac{\partial \log P(E|c)}{\partial \Theta}] = \sum_c P(c|E) \frac{\partial \log P(E|c)}{\partial \Theta} \]

(1.11)

Now we consider this gradient with respect to the parameter \( \Theta_{c'} \) of a specific activity model \( c' \). With equation 1.11 and the fact

\[ \frac{\partial \log P(E|c)}{\partial \Theta_{c'}} = 0 \text{ if } c' \neq c \]

we can get

\[ \frac{\partial \log P(c|E)}{\partial \Theta_{c'}} = \begin{cases} (1 - P(c|E)) \frac{\partial \log P(E|c)}{\partial \Theta_{c'}} \text{ if } c' = c \\ -P(c'|E) \frac{\partial \log P(E|c')}{\partial \Theta_{c'}} \text{ if } c' \neq c \end{cases} \]

As \( P(c|E) \) can be evaluated with equation 1.9, we mainly focus on computing \( \frac{\partial \log P(E|c)}{\partial \Theta_{c}} \) when evaluating the derivative of the \( \text{CLL} \). Please note that \( \frac{\partial \log P(E|c)}{\partial \Theta_{c}} \) is just the derivative of the log likelihood of DBN model \( c \) with respect to its own parameters \( \Theta_{c} \).

### 1.4.2. Incomplete Data

When the training data are incomplete, or the model has hidden nodes, \( P(E|c) \) is not decomposable, so evaluating the derivative of \( \text{CLL} \) becomes difficult. One natural choice of learning the DBN parameters is the EM algorithm, with the objective function substituted by the \( \text{CLL} \). However, in this case, in each maximization step, there is no analytical solution for estimating the parameter and we still need to go through the optimization procedure for the “completed” case. To avoid this double-looped optimization procedure, an efficient way is needed to directly compute the gradient of \( \text{CLL} \) with incomplete data. We resort this to the existing exact inference algorithms in the hybrid model. In the following parts, we show the derivative \( \frac{\partial \log P(E)}{\partial \theta} \) for three parent-child configurations except for the solved DP-DC case.\(^{12}\)

- **DP-CC:**
  In the case of discrete parents with continuous child, the derivatives \( \frac{\partial \log P(E)}{\partial \mu_{ij}} \) and \( \frac{\partial \log P(E)}{\partial A_{ij}} \) can be computed as follows:
For simplicity, we denote $E_{p(x_{t,i}, \pi_{t,i}, \pi_{t,i}) \mid E} \{ (x_{t,i} - \mu_{t,i}) (x_{t,i} - \mu_{t,i})^T \}$ as $E_{p(x_{t,i}, \pi_{t,i}, \pi_{t,i}) \mid E} \{ (x_{t,i} - \mu_{t,i}) (x_{t,i} - \mu_{t,i})^T \}$.

• CP-CC:

With continuous parents and continuous child, we have:

$$\frac{\partial \log P(E)}{\partial \mu_{t,i}} = -A_i^2 \sum_t [\mu_{t,i} - E(x_{t,i} \mid E] + W_i E(\pi_{t,i} \mid E)]$$

$$\frac{\partial \log P(E)}{\partial W_i} = -A_i^2 \sum_t [\mu_{t,i} E(\pi_{t,i}^T \mid E] - E(x_{t,i}, \pi_{t,i}^T \mid E] + W_i E(\pi_{t,i} \pi_{t,i}^T \mid E)]$$

$$\frac{\partial \log P(E)}{\partial A_i} = \sum_t \left[ A_i^{-1} - A_i E_{p(x_{t,i}, \pi_{t,i}, \pi_{t,i}) \mid E} \{ (x_{t,i} - \mu_{t,i} - W_i \pi_{t,i}) (x_{t,i} - \mu_{t,i} - W_i \pi_{t,i})^T \} \right]$$

The terms $E_{p(x_{t,i}, \pi_{t,i}, \pi_{t,i}) \mid E} \{ (x_{t,i} - \mu_{t,i} - W_i \pi_{t,i}) (x_{t,i} - \mu_{t,i} - W_i \pi_{t,i})^T \}$, $E(x_{t,i}, \pi_{t,i}^T \mid E])$ and $E(\pi_{t,i} \pi_{t,i}^T \mid E]$ can be computed as follows:

$$E(x_{t,i}, \pi_{t,i}^T \mid E] = \text{cov}(x_{t,i}, \pi_{t,i}) + E(x_{t,i}) E(\pi_{t,i})^T$$

$$E(\pi_{t,i} \pi_{t,i}^T \mid E] = \text{cov}(\pi_{t,i}) + E(\pi_{t,i}) E(\pi_{t,i})^T$$

$$E_{p(x_{t,i}, \pi_{t,i}, \pi_{t,i}) \mid E} \{ (x_{t,i} - \mu_{t,i} - W_i \pi_{t,i}) (x_{t,i} - \mu_{t,i} - W_i \pi_{t,i})^T \}$$

$$= \text{cov}(x_{t,i}) + (E(x_{t,i}) - \mu_{t,i}) (E(x_{t,i}) - \mu_{t,i})^T - \text{cov}(x_{t,i}, \pi_{t,i}) - (E(x_{t,i}) - \mu_{t,i}) E(\pi_{t,i})^T$$

$$-W_i E(\pi_{t,i} \pi_{t,i}^T \mid E]) + W_i E(\pi_{t,i} \pi_{t,i}^T \mid E]$$

In the above equations, $E(x_{t,i})$, $\text{cov}(x_{t,i})$, $E(\pi_{t,i})$ and $\text{cov}(\pi_{t,i})$ are the mean and variance of the posterior distribution of $x_{t,i}$ and $\pi_{t,i}$ respectively, $\text{cov}(x_{t,i}, \pi_{t,i})$ is the covariance of $x_{t,i}$ and $\pi_{t,i}$ with respect to posterior distribution $p(x_{t,i}, \pi_{t,i} \mid E)$, which can all be obtained through the inference of the hybrid Bayesian network.

For simplicity, we denote $E_{p(x_{t,i}, \pi_{t,i}, \pi_{t,i}) \mid E} \{ (x_{t,i} - E(x_{t,i})) (x_{t,i} - E(x_{t,i}))^T \}$ as $\text{cov}(x_{t,i})$. 

\[ \frac{\partial \log P(E)}{\partial \mu_{t,j}} = -\sum_t P(\pi_{t,i} = j|E) A^2_{t,j} \mu_{t,j} + A^2_{t,j} \sum_t E_{p(x_{t,i}, \pi_{t,i} = j|E) \mid E} \{ (x_{t,i} - \mu_{t,i}) (x_{t,i} - \mu_{t,i})^T \} \]

\[ \frac{\partial \log P(E)}{\partial A_{t,j}} = -\sum_t P(\pi_{t,i} = j|E) A^{-1}_{t,j} \]

\[ -A_{t,j} \sum_t E_{p(x_{t,i}, \pi_{t,i} = j|E) \mid E} \{ (x_{t,i} - \mu_{t,i}) (x_{t,i} - \mu_{t,i})^T \} \]
After obtaining these terms, we can then compute the derivatives \( \partial \log P(E)/\partial \mu_i \), \( \partial \log P(E)/\partial W_i \) and \( \partial \log P(E)/\partial \Sigma_i \).

Up to this point, we can compute the \( \partial \log P(E)/\partial \theta \) (\( \theta \in \{\mu, \Sigma, W\} \)). Further, based on the discussion in section 1.4.1, we can finally obtain the gradient of the CLL with respect to \( \mu_{ij} \) and \( \Sigma_{ij} \) of model \( c \).

1.5. Activity Recognition using DBN combined with the first-order probabilistic logics knowledge (PLDBN)

For the activity recognition applications, there often exists some approximate yet generic 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 the PLDBN model, we identify such knowledge in the form of first-order probabilistic logics (FOPLs), and then try to incorporate these knowledge in our activity model.\(^{16}\)

1.5.1. FOPLs for activity recognition

FOPLs is one type of knowledge representation language preserving the expressive power of first-order logic while introducing the probabilistic treatment of uncertainty. Several FOPLs have been proposed,\(^ {14}\) and we keep the formal syntax and semantics defined by Halpern et al.\(^ {15}\)

We use the following alphabet to represent the knowledge for activity recognition:

- **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);
- **Connective symbols**: \( \lor, \land, \forall, \neg, \mid \);
- **Variable**: t, AS (denotes one of the three constants: POS, SH and SP), s;
- **Probability operator**: Pr;
- **Basic numeric operator**: +, *, =, >;
- **Function**: Next;

We describe the domain elements with two sorts of terms: the object term and numeric term. 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. \( \Pr(\text{Is(position, near, t)}) \)). 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. Then, we can further transform the logic formula of the such knowledge into four probabilistic constraints on the conditional probabilities of our activity model in PLDBN.
The first logic is the smoothness logic. It stands for the fact that: the speed of an object at a successive time is more likely to be low if its current speed is low than its current speed is high. In FOPL, it is:

\[ \Pr[\text{Is}(SP,LO,\text{Next}(t)) \mid \text{Is}(SP,LO,t)] \geq \Pr[\text{Is}(SP,LO,\text{Next}(t)) \mid \text{Is}(SP,HI,t)] \]

It can be written in the following probabilistic constraint format:

\[ P(V_{t+1} = L \mid V_t = L) \geq P(V_{t+1} = L \mid V_t = H) \quad (1.13) \]

Here \( L \) denotes the low speed state and \( H \) denotes the high speed state.

The second logic is the position-motion logic. It stands for the fact that: 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. The FOPL format is:

\[ \Pr[\text{Is}(\text{POS,FR,Next}(t)) \mid \text{Is}(\text{POS,NR},t) \land \text{Is}(\text{SP,HI},t)] \geq \Pr[\text{Is}(\text{POS,FR,Next}(t)) \mid \text{Is}(\text{POS,NR},t) \land \text{Is}(\text{SP,LO},t)] \]

Similarly, it can be written in the following probabilistic constraint format.

\[ P(Y_{t+1} = F \mid Y_t = N, V_{t+1} = H) \geq P(Y_{t+1} = F \mid Y_t = N, V_t = L) \quad (1.14) \]

Here \( N \) denote near position state; and \( F \) is far position state.

The third logic is the shape-motion logic. It stands for the fact that: It is more probable for an object to change from simple shape to complex shape with a low speed than with a high speed. We have

\[ \Pr[\neg \text{Is}(\text{SH,CO,Next}(t)) \mid \text{Is}(\text{SH,SI},t) \land \neg \text{Is}(\text{SP,CO,Next}(t))] \geq \Pr[\neg \text{Is}(\text{SH,CO,Next}(t)) \mid \text{Is}(\text{SH,SI},t) \land \text{Is}(\text{SP,HI},t)] \]

The probabilistic constraint is:

\[ P(S_{t+1} = 1 \mid S_t = 0, V_{t+1} = L) \geq P(S_{t+1} = 1 \mid S_t = 0, V_{t+1} = H) \quad (1.15) \]

Here \( S_t = 1 \) denotes complex shape and \( S_t = 0 \) denotes simple shape.

The fourth logic which is the spatio-temporal logic stands for the fact that: An object is more likely to have a high spatio-temporal response if it has a simple shape at current frame and a complex shape at next frame, than if its shape at current frame and next frame are both simple. In FOPL, it is:

\[ \Pr[\neg \text{Is}(\text{ST,HI,Next}(t)) \mid \text{Is}(\text{SH,SI},t) \land \neg \text{Is}(\text{SP,CO,Next}(t))] \geq \Pr[\neg \text{Is}(\text{ST,HI,Next}(t)) \mid \text{Is}(\text{SH,SI},t) \land \text{Is}(\text{SP,SI,Next}(t))] \]

The probabilistic constraint is:

\[ P(ST_{t+1} = 1 \mid S_t = 0, S_{t+1} = 1) \geq P(ST_{t+1} = 1 \mid S_t = 0, S_{t+1} = 0) \quad (1.16) \]

Here \( ST_t = 1 \) is the high spatio-temporal response and \( ST_t = 0 \) is low spatio-temporal response.
1.5.2. Incorporate the FOPLs with the activity model

Having the four conditional probabilistic constraints mentioned in section 1.5.1, we can generate two types of model priors. First, the domain knowledge, in terms of qualitative constraints on the model conditional probabilities, can be used as parameter constraints to regularize the parameter learning for the activity model. Secondly, we apply the probability constraints as a soft structure prior, which can 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.\(^6\)

We use the log posterior probability (LPP) as the criterion for model structure selection,\(^6\) with the soft structure prior generated from FOPLs. In the case of incomplete training data, a widely adopted approach for DBN model search is the structural EM (SEM) algorithm.\(^7\) 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 used the constrained structural EM (CSEM) algorithm to learn the model structure combining the training data with these constraints.\(^6\)

With the parameter constraints, the model selection criterion combined with soft structure prior, and the CSEM algorithm for incomplete data, we can then incorporate the domain knowledge in the processing of learning the activity model.

1.6. Experiments

We trained and tested our GDBN, DDBN and PLDBN models in experiments discussed in this section. The KTH data set\(^2\) is used in the training and testing for all our three models. We compared our performances with the state-of-art approaches on this dataset, as shown in table 1.3. We also used the dataset from distant view and the parking lot dataset for GDBN and PLDBN respectively.

1.6.1. Generative DBN (GDBN) Model Experiments

1.6.1.1. GDBN Experiments on the dataset from distant view

Firstly, a set of preliminary experiments are performed using the dataset from distant view to train and test the GDBN model without and with the vehicle detection prior knowledge.

This data set from distant view is a set of videos that contains a handful of activities related to vehicles. Thus, we choose this data set to test the concept of incorporating the object detection knowledge with the traditional time-sliced graphical models like HMMs and DBNs. While, videos from this data set are shot far from the activities. The image resolution is relatively low and the contrast between the background and the moving objects is limited. Moreover, we used the computed track both for training and testing. Due to the limited image quality, the
quality of the tracker and bounding boxes are also relatively low.

We chose eight activities from this data set to test the GDBN model. The eight activities are *Opening a Trunk* (OAT), *Loading a Vehicle* (LAV), *Entering a Facility* (EAF), *Unloading a Vehicle* (UAV), *Getting into a Vehicle* (GOV), *Getting out of a Vehicle* (GIV), *Exiting a Facility* (XAF), *Closing a Trunk* (CAT). Except for the EAF and XAF, all other six activities are involved with a vehicle. Recognition results on this data set is given in table 1.1.

Table 1.1. Comparison of GDBN results without or with the vehicle (Veh.) prior knowledge.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\alpha = 0.4$</td>
<td>$\beta = 0.6$</td>
<td>$\alpha = 0.6$</td>
<td>$\beta = 0.4$</td>
<td>$\alpha = 1.0$</td>
</tr>
<tr>
<td>OAT</td>
<td>0.0909</td>
<td>0</td>
<td>0.1818</td>
<td>0.1818</td>
<td>0.1818</td>
</tr>
<tr>
<td>LAV</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.1000</td>
<td>0.1000</td>
</tr>
<tr>
<td>EAF</td>
<td>0.5000</td>
<td>0.3929</td>
<td>0.5357</td>
<td>0.6429</td>
<td>0.6786</td>
</tr>
<tr>
<td>UAV</td>
<td>0.6087</td>
<td>0.5217</td>
<td>0.6087</td>
<td>0.6087</td>
<td>0.6087</td>
</tr>
<tr>
<td>GIV</td>
<td>0.2000</td>
<td>0.2000</td>
<td>0.2000</td>
<td>0.2000</td>
<td>0.2000</td>
</tr>
<tr>
<td>GOV</td>
<td>0.4118</td>
<td>0.4118</td>
<td>0.4118</td>
<td>0.4706</td>
<td>0.4706</td>
</tr>
<tr>
<td>XAF</td>
<td>0.3421</td>
<td>0.2632</td>
<td>0.4211</td>
<td>0.6053</td>
<td>0.6842</td>
</tr>
<tr>
<td>CAT</td>
<td>0.0667</td>
<td>0</td>
<td>0.0667</td>
<td>0.1333</td>
<td>0.1333</td>
</tr>
</tbody>
</table>

RRNA$^a$ | 0.2838   | 0.2299     | 0.3095     | 0.3678     | 0.3822     |
| OARR$^b$ | 0.3234   | 0.2635     | 0.3533     | 0.4311     | 0.4551     |

$^a$Recognition Rate Numerical Average over eight activities.

$^b$OverAll Recognition Rate.

Fig. 1.5. With or without prior knowledge for GDBN model.

For this data set with limited quality, our GDBN model can generally reach an overall recognition rate of more than 32% for eight activities. And the GDBN model combined with vehicle detection prior knowledge performed better than the tradition GDBN model even if the vehicle detector has only 60 percent of positive detection rate, and as high as 40 percent of false alarm rate. As the vehicle detector
performance improved, the GDBN model incorporated with vehicle detection prior knowledge would improve significantly, as shown in figure 1.5.

1.6.1.2. **GDBN Experiments on KTH Dataset**

The KTH dataset\(^2\) is a human activity dataset with 6 basic human activities: walking, jogging, running, boxing, hand waving and hand clapping. Each activity is performed by 25 subjects in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes and indoors. So, there are totally 600 video clips in the dataset. In this chapter, we apply our three DBN model discussed in section 1.2.3, as well as the discriminative learning algorithm, and FOPLs transformed into structural prior and parameter constraints on recognizing these 6 activities.

In KTH data set, we trained the GDBN model with 16 training subjects for each activity, as the way by Yuan et al.\(^{18}\) and Laptev et al.\(^{19}\) The recognition rate is given in table 1.3. The KTH data set consists only single human activities, thus the vehicle detection prior knowledge information is not incorporated in this experiment. We can find that this GDBN performance without prior knowledge is slightly lower than the state of the art performance on the KTH dataset with simple activities. While, the further developed discriminative learning DBN and the PDBN can be comparable. These results will be discussed in the following.

1.6.2. **Discriminative DBN (DDBN) Experiments on KTH Dataset**

1.6.2.1. Discriminative Learning vs. Generative Learning

![Fig. 1.6. Discriminative learning vs. generative learning on KTH dataset.](image)

We first focus on comparing the generative learning approach with the discriminative learning approach with different training size here. Since the basic states of the subject (motion, shape and appearance) are not labeled, the generative learning approach we used is the EM algorithm. For discriminative learning, as our approach can only guarantee a local optimum of the conditional log likelihood, one critical is-
sue is the initialization of the model parameters. In all our experiments, we use the result of the generative learning as the initialization for the discriminative learning.

We can compare the training error of the discriminative learning with generative learning based on the results in figure 1.6. It can be easily get that the discriminatively learned DBN performs consistently better than generatively learned DBN. When the number of training subjects is small (4 and 8), the discriminatively learned model achieves zero error rate on the training set. This is mainly due to the consistency of the training objective and classification criterion: the conditional likelihood we maximize is closely related to the classification rate on training set.

From figure 1.6, we can also compare the classification performance of the discriminatively learned model with generatively learned model. When the number of training subjects is large, the discriminatively learned models perform obviously better than the generatively learned models. More specifically, given 16 and 20 subjects for training, the error rates of discriminative learning are 4.5% and 2.5% lower than the generatively learning respectively. However, when the number of training subjects become smaller, discriminative learning suffers more from overfitting than generative learning. We can see that the classification error of the discriminatively learned model is 4.4% higher than generatively learned model given the sequences from 4 subjects for training.

1.6.2.2. **Comparison with Other Approaches on KTH dataset**

We also compare our approach with the state-of-art approaches in the KTH dataset. As the results from Yuan et al.\textsuperscript{18} and Laptev et al.,\textsuperscript{19} we compare our results with 16 training subjects. The comparison is given in table 1.3. While the classification performance of our DBN model learned with generative learning are about 4% worse than the state-of-art approaches, it can achieve comparable results to the state-of-art approaches if learned discriminatively.

Also, we can see that, our DDBN model performance and the PLDBN model performance are very close when both models are trained with the sequence of 16 subjects. The DDBN here is slightly better. While, PLDBN is very effective especially when we have limited training subjects.

1.6.3. **PLDBN model experiments**

1.6.3.1. **PLDBN model experiments with Parking Lot Dataset**

We first apply our PLDBN algorithm to the problem of recognizing human activities in the parking lot. The data set 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 data set 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.

In figure 1.7, we compare the knowledge-based CSEM with data-based SEM in learning both activity-dependent and activity-independent model structures.

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 1.7, 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.

1.6.3.2. PLDBN model experiments with KTH Dataset

Table 1.2 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.

We also compared our approach with the state-of-art approaches on this dataset.

![Fig. 1.7. Comparison of CSEM and SEM for learning activity-dependent and activity-independent models for PLDBN in parking lot dataset.](image-url)
As the way discussed before, we use the data from 16 subjects for training. Table 1.3 shows that we can achieve comparable result to the state-of-art approaches.

### Table 1.3. Compare our DBN models (GDBN, DDBN and PLDBN) with previous work on KTH dataset.

<table>
<thead>
<tr>
<th>Recognition rate</th>
<th>88.0%</th>
<th>92.5%</th>
<th>88.0%</th>
<th>92.1%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method - Generative DBN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Our method - Discriminative DBN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Our method - PLDBN with SEM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Our method - PLDBN with CSEM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yuan et al.\cite{18}</td>
<td>93.3%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laptev et al.\cite{19}</td>
<td>91.8%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 1.7. Summary

In this chapter, after briefly discussing the input data and the features for activity recognition, we introduced three kinds of Dynamic Bayesian Networks for the activity modeling and recognition problem. The GDBN is a standard general Dynamic Bayesian Network and it has two hidden nodes reflecting the states of the kinematic feature observations and image feature observations respectively in every time-slice. Other than the traditional classification criterion that choose the model with the highest likelihood, we also introduced our work on combining the likelihood outputted by DBN model with prior knowledge obtained from vehicle detection. Experiments showed that this combination during classification can improve the system performance.

The DDBN is learning the generative DBN models in a discriminative way. It can reduce the discrepancy between the training and testing objective for activity recognition in generative models. Compared to the generative learning approaches, the DDBN approach has a more consistent objective in the training stage with the classification criterion, which can guarantee a better classification performance on the training set. Based on our experiments on the real data from KTH activity dataset, we demonstrate the advantage of discriminative learning over generative learning when training data is sufficient.

For the PLDBN model, 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. 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.

References