An Adaptive Probabilistic Graphical Model for Representing Skills in PbD Settings

Haris Dindo
Dipartimento di Ingegneria Informatica
University of Palermo - Italy
dindo@dinfo.unipa.it

Guido Schillaci
Dipartimento di Ingegneria Informatica
University of Palermo - Italy
schillaci@dinfo.unipa.it

***Abstract*—Understanding and efficiently representing skills is one of the most important problems in a general Programming by Demonstration (PbD) paradigm. We present Growing Hierarchical Dynamic Bayesian Networks (GHDBN), an adaptive variant of the general DBN model able to learn and to represent complex skills. The structure of the model, in terms of number of states and possible transitions between them, is not needed to be known *a priori*. Learning in the model is performed incrementally and in an unsupervised manner.**

***Index Terms*—Machine Learning, Imitation Learning, Incremental Learning, Dynamic Bayesian Network, Growing Hierarchical Dynamic Bayesian Network.**

I. Introduction

Understanding others' actions and intentions is an essential prerequisite for building robots able to interact with humans in collaborative and/or competitive settings. Programming by Demonstration (PbD) is a powerful mechanism for enhancing and accelerating learning, since such an approach minimizes or eliminates the explicit and tedious programming of a task by a human user and it makes the interaction between robots and humans more natural.

This work focuses on the problem of *representing skills* in a PbD setting. Current approaches to skill representation can be divided into (see [2] for further information): *trajectories encoding* (processes are represented as a non-linear mapping between sensory and motor data) and *symbolic encoding* (tasks are described symbolically using classical machine learning techniques). Here, we propose a probabilistic graphical model for task representation, which can represent complex behaviors at different levels of abstraction. Our approach, essentially, merges the most important features of two graphical models, Growing Hidden Markov Model [5], and Hierarchical Dynamic Bayesian Network (see [4] for a reviews on DBNs), resulting in a unique model: *Growing Hierarchical Dynamic Bayesian Network (GHDBN)*. The model is *hierarchical* (i.e. able to characterize structured behaviors), and *growing* (i.e. skills, represented by the states of the discrete variables of the model, are learned or updated - *at each level of abstraction* - every time a new observation sequence is available). The model, once learned, can also be used in representing and understanding intentions, both *proximal* (the most probable next high-level state, given the current observed behaviour) and *distal* (the most probable activation of a sequence of high-level states, given the current one).

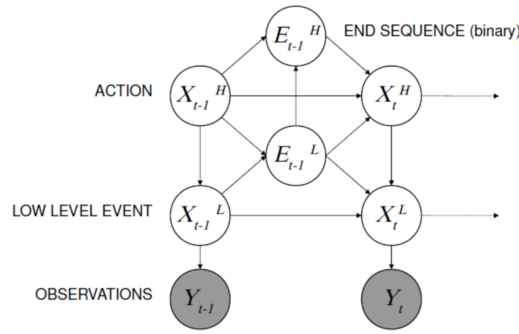

Fig. 1. The Growing Hierarchical Dynamic Bayesian Network

II. The GHDBN model

A probabilistic graphical model is identified by its structure and its parameters. The GHDBN (Figure 1) is a two-level HDBN. Each level is described by two stochastic variables, namely: X^H, high level representation of the skill/action; X^L, low-level behavior; two binary variables, E^H and E^L, representing the end-of-states-sequence markers for each level; Y, which models a multivariate Gaussian distribution from which are drawn the observations. The peculiarity of this model is that the number of states of X^H and X^L and their state transition structures are not constant: complex actions and elementary behaviors are learned incrementally without any prior information.

Both structure and parameters of the model are learned in an unsupervised online manner, processing observation by observation. The problem is challenging since we need to learn the structure of both X^H and X^L (that is, discover/modify/remove complex or basic behaviors and possible transitions between them), and the Conditional Probability Distributions of the variables (which quantify the probability of passing from each state, that is behavior, to another).

Starting with the structure of the low-level variable X^L, we claim that it should reflect the spatial structure of the feature space discretization. In other words, transitions between basic behaviors are only allowed if the corresponding nodes of a topological map of the feature space are connected. The observations' space is clustered incrementally by using the Instantaneous Topological Map algorithm [3], [5], which provides a discrete representation of a continuous feature

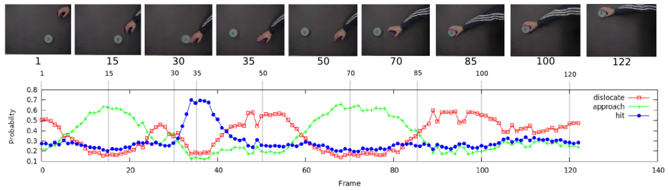

Fig. 2. An example of observing a sequence of complex actions. We have adopted a 3-dimensional feature vector represented by the following signals: *hand velocity variation*, *hand-object distance variation*, and *hand-object angle difference*. The lower graph depicts the probability distribution of X^H for each timestep. In the proximity of frames ♯30 and ♯50, the most probable action is *dislocate* . Actually, in these ranges, objects are not being moved, so the signal *distance-variation* between hand-object is constant, which is the main characteristic of the action *dislocate*.

space[1] in the form of a graph where feature space regions are represented as nodes and adjacent regions are connected by edges.

Adding (removing) a node of the topological map causes increasing (decreasing) of the number of X_L states, while adding (removing) an edge causes setting a not null (null) value in the corresponding element of the Conditional Probability Tables[2] of the variable X^L, where a null value corresponds to an impossible transition between the considered states.

The structural learning of X^H (the complex behavior) is slightly more complicated since this variable is not directly observed. The GHDBN could be imagined as many GHMM [5] (each representing a complex skill) unified in a single structure. Having a sequence of observation data, we can compute a likelihood score for each of the submodels to determine the most probable behavior (or GHMM, represented by an X^H state) which has generated the observations. We used the Likelihood Ratio Test. The inference in the model is performed by a Particle Filter.

Once the structure of the model has been learned, the second problem is how to learn the parameters of the GHDBN, that is the entries of the Conditional Probability Tables. The classical approach for parameters learning in probabilistic graphical models is the Expectation-Maximization (EM) algorithm. We have adopted a purely online EM algorithm[1] which updates the parameters by processing observation after observation, without considering past information.

Let X_i be a node of a generic DBN, Pa_i the set of the parents of X_i. An entry in the CPT of the variable X_i is

$$\theta_{ijk} = P(X_i = x_i^k | Pa_i = pa_i^j)$$

[1]As an example, in a monodimensional feature space representing the velocity, the basic behaviors (nodes) could be: *still* (null value), *walking* (medium value), *running* (high value).

[2]Each variable has associate a Conditional Probability Distribution. Such a CPT quantifies how a variable is conditioned by its parents. In a dynamic model, a parent of a variable could be another variable in the same time slice, or the same variable in the previous time slice (Transition Probability Table). In the first time step, this CPT represent the Prior Probability Table (e.g. the probability that an action starts with a particular behavior). Another table is defined to specify the End-of-Sequence Probability Distribution (e.g. which is the last behavior of a specific action?).

So, the updating rule of the EM(η) algorithm is:

$$\theta_{ijk}^T = (1-\eta)\theta_{ijk}^{T-1} + \eta\left[\frac{P(x_i^k, pa_i^j | y_t, \theta_{ijk}^{T-1})}{P(pa_i^j | y_t, \theta_{ijk}^{T-1})}\right]$$

where the learning rate η introduces a forgetting bias, but let the parameters oscillate around the real values, permitting to get out from local maxima when the environment changes and the model needs to be readapted.

TABLE I
THE GHDBN LEARNING ALGORITHM

Observe action.

a. Create a new X^H state (initializing the CPT as uniform) and learn only the submodel conditioned on it;
b. Compute the Likelihoods conditioned on each X^H state;
c. Compute, for each couple of new-state / old-state, the Lik. Ratio;
d. If all these LRs are greater than a threshold (dependent on the number of parameters and on the length of the sequence), probably the sequence is relative to a new complex process. Maintain the new X^H state. However, remove it and learn the whole model normally.

Learn. For each observation:

a. Iterate ITM. Update X^L dimension and the structure of its CPTs.
b. If a new X^H state has been added previously, apply the parameters update rules to all the CPTs conditioned on the new X^H state, that is, model only the new complex action.
c. Else, apply the parameters update rules to all the CPTs.

The model has been learned sequentially with observations relative to three actions, in order: `dislocate`, `approach` and `hit`. The value of η has been set high in order to achieve a fast learning of parameters. The performance of the GHDBN has been tested with observations taken from 24 videos containing various sequences of complex actions, for a total of 1655 frames (observations) analyzed. Each frame is classified as the observation of a particular action if the corresponding state in the current probability distribution of X^H is the most probable. Table II depicts the confusion matrix which shows promising results of the GHDBN model.

TABLE II
CONFUSION MATRIX PERCENTAGE.

		Actual		
		Dislocate	Approach	Hit
Predicted	Dislocate	67.58%	22.26%	12.07%
	Approach	5.80%	76.26%	6.90%
	Hit	26.62%	1.48%	81.03%

REFERENCES

[1] E. Bauer, D. Koller, and Y. Singer. Update rules for parameter estimation in bayesian networks, 1997. In Uncertainty in Artificial Intelligence (UAI), pages 313.
[2] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot programming by demonstration, 2008. In Handbook of Robotics. Chapter 59, Springer.
[3] J. Jockusch and H. Ritter. An instantaneous topological mapping model for correlated stimuli, 1999. In Proc. of the International Joint Conference on Neural Networks, Washington (US), vol. 1, pp. 529-534.
[4] K. P. Murphy. Dynamic bayesian networks: representation, inference and learning, 2002. University of California, Berkeley.
[5] Vasquez, Fraichard, and Laugier. Incremental learning of statistical motion patterns with growing hidden markov models, 2007. Transactions on intelligent transportation systems.