Learning Probability Distributions over Partially-Ordered Human Everyday Activities

Moritz Tenorth
Institute for Artificial Intelligence & TZI*
University of Bremen, Germany
tenorth@cs.uni-bremen.de1

Fernando De la Torre
Carnegie Mellon University
Pittsburgh, PA, USA
ftorre@cs.cmu.edu

Michael Beetz
Institute for Artificial Intelligence & TZI*
University of Bremen
beetz@cs.uni-bremen.de

***Abstract*— We propose a method to learn the partially-ordered structure inherent in human everyday activities from observations by exploiting variability in the data. Using statistical relational learning, the system extracts a full-joint probability distribution over the actions that form a task, their (partial) ordering, and their parameters. Relevant action properties and relations among actions are learned as those that are consistent among the observations. The models can be used for classifying action sequences, but also for determining which actions are relevant for a task, which objects are usually manipulated, or which action parameters are typical for a person. We evaluate the approach on synthetic data sampled from partial-order trees as well as two real-world data sets of humans activities: the TUM kitchen data set and the CMU MMAC data set. The results show that our approach outperforms sequence-based models like Conditional Random Fields for activities that allow a large degree of variation.**

I. Introduction

When observing people cooking a meal, one will notice a large variability in *how* they perform the different actions: Some people first prepare all the tools and ingredients, others start to cook right away and get the things they need just in time. In addition, people get distracted, perform irrelevant actions in between, or forget something they need to make up for later on. As a result, the observations differ in terms of *which* actions have been performed, in which *order* they have been done, and what their *parameters* have been. This high degree of variability stems from the relative freedom in how many tasks can be performed: Though humans tend to describe them as sequences, many tasks are in fact much less constrained, and only impose a few partial ordering constraints among their sub-actions instead of a total ordering among all of them. These ordering constraints may be due to causal dependencies, e.g. if one action depends on the outcome of another, but may also result from person-specific habits or preferences. Both kinds of constraints can be useful: When planning robot actions, a model of dependencies among actions can serve for computing a suitable ordering and to exploit freedom for optimization. When observing human actions, such models describe different styles to perform an activity and can be used to spot differences and anomalies, for example caused by medical conditions.

In this paper, we propose a method for learning such action models from observation. Given a diverse training set of observed actions, we can *exploit* the variability in the data to learn about the structure and properties of the task. The more diverse the training set is, the more alternative ways of how to perform a task can be learned. Those actions, properties and relations that consistently appear in many examples will have a higher likelihood to be relevant for the task than those that are only incidentally observed.

The models represent a joint probability distribution over the types of actions, their parameters (like the hand that is used or the object that is manipulated), and their pairwise ordering. Combined, these pairwise ordering constraints result in a partial order imposed on all actions in a task. In order to learn such models from noisy, uncertain observations, one needs to be capable of representing both relational knowledge and uncertainty, which is why we employ statistical relational learning techniques. Our implementation uses Bayesian Logic Networks (BLNs, [1]), which are a relational extension of Bayesian Networks. The learned full-joint probability distribution can be used for various inference tasks:

- *Classification* of activities by checking which constraints are satisfied
- Verification if an action has been performed correctly with respect to a reference model
- *Identification* of relevant actions in the activity as those that consistently appear in the training data
- *Inference* of missing information like the type of an action or object given the overall activity model
- *Manual analysis* of the learned models can give important insight into how a person performs a task

The remainder of the paper is organized as follows: We start with a review of related work on modeling and recognizing partially ordered activities, and formally describe the representation of actions in the system and the applied statistical relational learning techniques. We then evaluate the approach on a synthetic and two real-world data sets and finish with a discussion of its scalability and generalization.

II. Related Work

The common approach for describing and recognizing human activities is to model the observed action sequences using techniques such as Hidden Markov Models (HMMs) [2], Conditional Random Fields (CRFs) [3] or Suffix Trees [4]. These models describe the sequences in terms of local action transitions, which is particularly suited to largely sequential activities. Once the order of actions is not that well-determined any more, this approach shows its limitations:

*The Centre for Computing Technologies (TZI).

Even if only a few actions can be shifted around in the task context, this will create much variation and a large number of possible local transitions that confuse sequence-based methods. Also, the Markov assumption that the subsequent action only depends on the current one does not hold for many such activities, but the history of which actions have already been done needs to be taken into account.

There are few other systems in the area of action recognition that also address the problem of learning models of partially ordered tasks: Shi [5] uses (manually specified) Dynamic Bayesian Networks to represent the partial order. Gupta [6] describes a method for learning story lines of actions in baseball games using an AND/OR graph. Ekvall [7] learns deterministic ordering constraints from multiple observations in a blocks-world setting. All these approaches focus on only the ordering among atomic action entities, while our system learns a distribution over the order as well as the action parameters.

In the fields of planning (e.g. [8]) and plan recognition, there is much work about partially ordered plans. Kautz and Allen's seminal paper [9] formalizes plan recognition as a logical inference problem. Goldman et al. [10] extend this work to a probabilistic model that can handle partially ordered and interleaved plans. Both approaches, as well as more recent ones, rely on manually created model of the complete task and have mainly been applied to synthetic problems. Research in preference learning also deals with learning and representing orderings, though 'partial order' in that context usually refers to a total order among the top-k elements in a set, as opposed to a partial ordering of the complete set.

III. Describing the Structure of Tasks

The proposed system is to learn a partially-ordered model of a task T from a set of observed action sequences $\mathcal{S}$ that are instantiations of the abstract task. Action sequences are described as

$$\mathcal{S} = \{S_s^T | S_s^T = \langle a_0, a_1, \dots \rangle\} \tag{1}$$

These actions can have different lengths due to missed actions as well as noise actions in between the relevant ones, and are also expected to show strong variation regarding the order of actions. The abstract model learned from these actions describes a set of tasks $\mathcal{T}$, each of which is described by a set of actions $\mathcal{A}_t$, a possibly empty set of action properties $\mathcal{P}_t$ and an ordering relation $\mathcal{O}_t$ among the actions.

$$\mathcal{T} = \{T_t | T_t = \langle \mathcal{A}_t, \mathcal{P}_t, \mathcal{O}_t \rangle\} \tag{2}$$

Observed actions in an action sequence are marked with a subscript index a_i, the prototypical actions in a task model have a superscript index a^i. Action sequences are related to tasks via the $activityT$ predicate.

$$activityT(S^T) = T \tag{3}$$

Each task model comprises a set of n actions, which have one of m different types $A^0, \dots A^m$ that correspond to classes in the robot's action ontology.

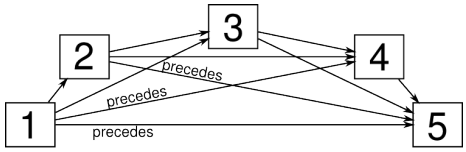

Fig. 1. Describing the partial order in the sequence $1-2-3-4-5$ by pairwise precedence relations.

$$\mathcal{A}_t = \{a^0, a^1, \dots, a^n\} \tag{4}$$

$$\forall i \in [0, n] : actionT(a^i) \in \{A^0, A^1, \dots, A^m\} \tag{5}$$

Actions may have different properties like the object manipulated or the hand used. $\mathcal{P}_t$ assigns a probability value to each property $\pi_j \in \Pi$ of each action a^i:

$$\mathcal{P}_t : \mathcal{A}_t \times \Pi \rightarrow [0, 1] \tag{6}$$

$$\Pi = \{\pi_0, \pi_1, \dots, \pi_p\} \tag{7}$$

$$P_{ij} = P(\pi_j(a^i) = True) \tag{8}$$

The ordering relation $\mathcal{O}_t$ for a task T describes the probability that an action a^i is executed before an action a^j in the respective task context.

$$\mathcal{O}_t : \mathcal{A}_t \times \mathcal{A}_t \rightarrow [0, 1] \tag{9}$$

In our system, both $\mathcal{P}_t$ and $\mathcal{O}_t$ are described using probabilistic relations that are learned from the training set of sequences S_{train}^T and described as predicates combined with a probability that this relation holds. The relative ordering of two actions is expressed using the $precedes$ predicate (Figure 1):

$$\forall a_i, a_j \in S_s : (i < j) \Leftrightarrow precedes(a_i, a_j, S_s) \tag{10}$$

Observations of actions are also described in terms of the predicates used in the action models like $actionT$, $precedes$, and optional predicates for action parameters like the $objectActedOn$. In the example, $N1$, $N3$, and $N4$ are action classes, while $O3$ is an object class.

$$\begin{aligned} activityT(Act_0) &= Act-2 \\ &\wedge actionT(N_1) = N1 \wedge objectActedOn(N_1, O_1) \\ &\wedge objectT(O_1) = O3 \\ &\wedge actionT(N_2) = N3 \wedge actionT(N_3) = N4 \dots \\ &\wedge precedes(N_1, N_2, Act_0) = True \\ &\wedge precedes(N_1, N_3, Act_0) = True \wedge \dots \end{aligned}$$

IV. Bayesian Logic Networks

In this paper, we apply Bayesian Logic Networks (BLNs) [1] to representing the aforementioned action structures. BLNs are a form of probabilistic logics and combine the expressiveness of first-order logics, required to describe the complex interactions between actions, parameters of these actions, with the representation of uncertainty. A BLN serves as a template for the construction of a ground mixed network to which standard Bayesian network (BN) inference

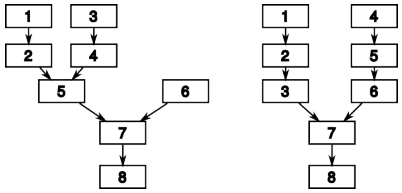

Fig. 2. Precedence graphs for the fictional activities *Act-1* (left) and *Act-2* (right) which were used for sampling the synthetic action data .

techniques can be applied. For our experiments, we use Backward Sampling [11], an approximate BN inference algorithm. Due to space limitations, we will only briefly describe BLNs and refer to [1] for details.

Formally, a BLN is described as a tuple $\mathcal{B} = (\mathcal{D}, \mathcal{F}, \mathcal{L})$ consisting of the declarations of types and function $\mathcal{D}$, a set of fragments of conditional probability distributions $\mathcal{F}$, and a set of hard logical constraints $\mathcal{L}$ as formulas in first-order logic. The fragments $\mathcal{F}$ describe dependencies of abstract random variables, in our case for instance between $precedes(a_i, a_j, S_s)$ and $actionT(a_i)$. Compared to Bayesian Networks, BLNs abstract away from concrete entities and represent generic relations between classes of entities, similar to the way predicate logic abstracts away from the concrete entities in propositional logics. Examples of the BLN fragments are shown in Figure 3, where the oval nodes denote random variables and the rectangular nodes contain preconditions for the respective fragments to be applicable. For a given set of entities (in our case observations of actions), the BLN gets instantiated to a ground mixed network, expanding the abstract relations with the concrete domains of e.g. actions and objects. Learning BLNs requires determining the conditional probability tables in the fragments in $\mathcal{F}$, which reduces to simply counting the relative frequencies of the relations in the training set. While the declarations $\mathcal{D}$, the fragments $\mathcal{F}$ and logical constraints $\mathcal{L}$ are defined manually, they only describe the form of the observed actions and that a partial order among them may exist. The actual action types, their properties, relations to objects, as well as their ordering relations are learned from data.

V. Experiments

We evaluate the system first on synthetic data, and then on two real-world data sets of human activities. Due to space limitations, we cannot show every aspect of evaluation for each of them, but concentrate on the most interesting aspects, respectively.

A. Synthetic Data

First, we tested the approach with synthetic data sequences that have been sampled from the two precedence graphs in Figure 2. Note that both graphs consist of the same basic actions, i.e. no single action can be used as a hint which activity is performed, but only the order contains information. This is certainly more difficult than most real-world applications, but for instance required when distinguishing between different styles of performing the same activity.

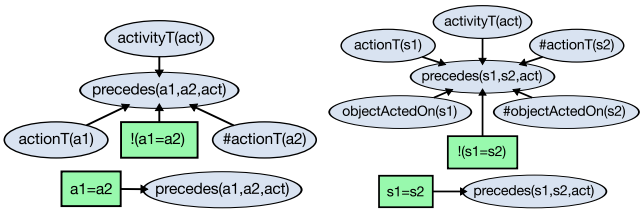

Fig. 3. The model structure for the synthetic data (left) and the TUM kitchen data (right) with dependencies as conditional probability distribution fragments.

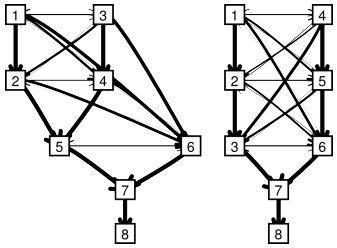

Fig. 4. Learned dependencies in the synthetic data set, the thickness of the lines depicts the probability that one action is performed before another. The partial-order structure could successfully be recovered from the observed data.

The sampling is performed using the following procedure: Let $\mathcal{N}$ represent the set of nodes whose ordering constraints are met and who can thus be selected in the next step, and let $prereq(n)$ be the set of nodes that are prerequisites for node n. The sampling starts with the set of nodes

$$\mathcal{N}_0 = \{n_n : \forall n_k \neq n_n \Rightarrow n_n \notin prereq(n_k)\}, \qquad (11)$$

thus the set of all actions that are not prerequisites for any other action. At each sampling step i, a random element n_i is chosen, and the sampling continues with

$$\mathcal{N}_{i+1} = (\mathcal{N}_i \cup prereq(n_i)) \setminus n_i \qquad (12)$$

All actions occur exactly once in this data set, thus for both graphs is $m = n = 8$, and there are no action properties, i.e. $\mathcal{P} = \emptyset$. The data can be modeled with the very simple BLN in Figure 3 (left).

1) Learning the partial order: The learning algorithm should be able to recover the partial order from the data. Figure 4 visualizes the conditional probabilities inside the $precedes$-node of the BLN. In this visualization, redundant relations have been pruned, i.e. when $P(precedes(A, B)) = 1$, $P(precedes(A, C)) = 1$ and $P(precedes(B, C)) = 1$, we did not draw the edge $A - C$ to improve clarity. As can be seen in the picture, the algorithm successfully recovered the partial-order structure the data was sampled from.

Interconnections between for instance the nodes $N1$, $N2$, $N3$, and $N4$, which are not present in the original graph, reflect the properties of the sampling algorithm. It is equally likely to switch to a different branch of the activity (i.e. between $N1 - N2$ and $N3 - N4$) as it is to continue on the same branch. If observations of humans show such structures, these interconnections can reflect an alternating behavior as opposed to a stringent execution of a sequence of actions.

2) Classification in the presence of noise: Observations of activities often comprise irrelevant actions that are performed in between the essential actions, like wiping up

ID	activityT	actionT	most likely types
12	Act-1	N8	N8(0.5760), N7(0.4135), X5(0.0042)
25	Act-1	N7	N7(0.4837), N5(0.2022), N6(0.0846)
33	Act-1	N8	N8(0.7667), N7(0.2211), X5(0.0117)
43	Act-1	N1	N1(0.5303), N3(0.4243), N2(0.0447)
24	Act-2	N6	N6(0.2867), N3(0.2498), N7(0.1395)
37	Act-2	N4	N4(0.5940), N1(0.3800), N5(0.0220)
48	Act-2	N4	N4(0.3950), N2(0.2860), N5(0.1860)

TABLE I

INFERRING THE TYPE OF UNKNOWN ACTIONS IN AN ACTIVITY.

spilled liquids or drinking a glass of water while cooking a meal. Similar *action noise* can result from errors in the segmentation of observations into single actions.

To test the influence of irrelevant actions in between the important ones, we modified the sampling algorithm described earlier so that, in each step, a noise action may be chosen instead of one of the relevant actions with a certain probability. Formally, equation (12) changes to

$$\mathcal{N}_{i+1} = (\mathcal{N}_i \cup prereq(n_i) \cup \mathcal{X}) \setminus n_i \tag{13}$$

where $\mathcal{X}$ is a set of noise actions, i.e. actions that are irrelevant to the activity. In the experiments, we sampled from $\|\mathcal{X}\| = 10$ noise actions, denoted $x_0 \dots x_9$, with a probability of 10%, 20% and 50% respectively. The sequences in both the training and the testing set comprised these noise actions, so the system did not know a priori which actions are actually relevant.

Figure 5 (right) shows the classification performance (F1 value) of our system. The results were obtained by approximate inference on the BLN model using Backward Sampling with 5000 samples (note that this is not the size of the training or testing database, but the number of samples drawn by the Bayesian network inference algorithm). Even with the very noisy sequences, in which about half of the actions are irrelevant to the activity, the system is still able to learn a model that allows for good classification. If there is few noise (lines without markers), as few as five example sequences suffice for reasonable performance, while the more noisy data requires about 15 sequences to obtain similar results.

We compare the classification results to those obtained using Hidden Conditional Random Fields (HCRF, [12]), which have been shown to outperform Hidden Markov Models and Conditional Random Fields, the probably most commonly used methods in action recognition. HCRF model the sequence of actions, but cannot take longer-range dependencies like global ordering constraints into account. The results in Figure 5 suggest that the model gets confused by the large variation in the data and the significant amount of noise. While the results are still rather stable for low-noise data (lines without markers), they get much worse when the proportion of irrelevant actions increases.

3) Inferring the types of single actions: Since the models learn a joint probability distribution over all aspects of the action, they can be used for different inferences, for example to infer the most likely type of a single action in a sequence:

$$argmax(P(actionT(a_i)|S^T)) \tag{14}$$

We randomly sampled sequences from the nosiest version of both activities (50% noise actions), removed the type of an arbitrary action in the test sequence, and inferred this type given the rest of the sequence. The exemplary results in Table I show that it is possible to infer the type of an action given the type of the activity and the surrounding actions. The results also indicate that the model has learned which actions are easy to identify. Action N8, for example, is always the last non-noise action in every sequence and can thus easily be identified (seq. 12, 33). When there is confusion, it is mostly between actions on a similar level of the precedence graph (e.g. N4 and N1 in seq. 37) or between direct predecessors and successors (as in seq. 25, where N5 and N6 are direct predecessors of N7).

action	probability	action	probability
N1	0.83	X0	0.29
N2	0.83	X1	0.31
N3	0.83	X2	0.39
N4	0.83	X3	0.40
N5	0.83	X4	0.27
N6	0.83	X5	0.26
N7	0.83	X6	0.36
N8	0.83	X7	0.34
		X8	0.37
		X9	0.31

TABLE II

RELEVANCE OF AN ACTION AS ITS PROBABILITY GIVEN AN ACTIVITY.

4) Identifying (ir)relevant actions: A priori, the system does not know which actions are relevant and which are just noise. Using the proposed models, the probability of an action given the activity can be calculated.

$$P(actionT(a_i) = A^j|activityT(S^T)). \tag{15}$$

Table II shows that, even in the extreme case of 50% noise actions, the relevant actions are more consistent across the observed episodes and therefore have a higher probability. Since both activities consist of the same number of actions, the results are identical for both the $Act-1$ and $Act-2$ activity.

B. TUM Kitchen Data Set

As a real-world data set, we use the TUM Kitchen Data Set [13] for evaluation which contains several observations of different subjects performing a table-setting task. In addition to motion-capture data, it also provides information about objects that are manipulated (from RFID readings) and doors and drawers being opened (via magnetic sensors). All subjects perform the same activity (setting the table for one person), using the same objects, but in different order: Some behave like an (inefficient) robot that transports the objects one-by-one, others are more human-like in carrying several objects at once. On the one hand, this makes this data set quite structured, but on the other hand, it creates a difficult classification challenge since all objects and actions are identical for both classes. In total, there are $m = 8$ types of actions like $Reaching$ or $OpeningACupboard$, and the observation sequences have a length of about 70 action segments. $\mathcal{P} = \{objectActedOn\}$, the object an action is performed on, is the only property. The BLN structure for this data set is shown in Figure 3 (right). In this paper, we do not deal with the problem of segmenting the continuous motion, but rather use the manually created labels provided

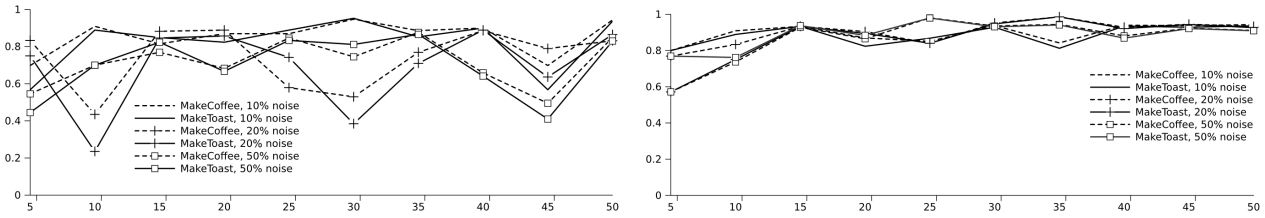

Fig. 5. Recognition rates (F1 value) on synthetic data with different noise levels (10%, 20% and 50% probability of choosing a noise action) and sizes of the training and testing set (5 to 50 samples, see x-axis). Left: HCRF. Right: BLN (our approach).

with the data set. Inferring these segments from the data is a challenge by itself, and some first work on this topic has been presented by the authors of the data set [13].

Visualizing the learned model is difficult since the object type influences the order. However, when plotting the conditional probability for each action a_1 over the object o_1 $\times$ the subsequent action a_2 with object o_2, a peaked, sparse distribution can be observed (Figure 6). Many values are zero because several object-action pairs never occur (like *opening a knife*). Some actions always occur before others (conditional probability of one), others have softer ordering constraints as can be seen by the lower peaks in the diagram. We noticed in our experiments that such sparse, peaked distributions are typical for problems that show a distinct partial order.

1) Classification performance: We tested the model by discriminating between two different styles of setting the table, in the following referred to as *robot-like* (transporting one object at a time) and *human-like* (a more natural behavior, including e.g. grasping all pieces of silverware at once). Due to a lack of data, the test sequences were manually created to be a typical example of each activity style by changing the order of the transported objects, adding noise actions, and shortening sequences where some object interactions were omitted. One sequence (*HumanRobot*) was constructed by concatenating the first half of a human-like and the second half of a robot-like sequence.

Table III presents the inference results obtained using Backward Sampling with 5000 samples and, as a comparison, the classification obtained from the HCRF (identical results for $m = 3, 5, 10, 20$ hidden states). Features for the classification were the action class and the object.

The HCRF fails to classify the sequences and labels all of them as $Human$, supposedly because it did not learn the subtle differences in the ordering. Our system correctly classified almost all the sequences, only the $HumanRobot$ sequence was classified as $Human$, whereas an indecisive result would have been expected. Apparently, the parts of the $Human$ sub-sequence are more salient than those in the $Robot$ part of the sequence.

As mentioned before, all actions and objects are identical for both classes and only the order differs. In other cases, the distinction between different activities would obviously become much easier.

C. CMU MMAC Data Set

The CMU MMAC Data Set [14] provides observations of 43 subjects cooking 5 different recipes. So far, only part of the data has been labeled, namely a subset of the '*making brownies*' and the '*cooking an omelette*' recipes, which we use for learning the models. On this data, we will present some queries that show that the models do not only represent the ordering, but a complete joint probability distribution over different aspects of the observed actions.

	BLN		HCRF	
activityT	SttHuman	SttRobot	SttHuman	SttRobot
Human1	**1.0000**	0.0000	**1**	0
Human2	**1.0000**	0.0000	**1**	0
Human1short	**1.0000**	0.0000	**1**	0
HumanRobot1	**1.0000**	0.0000	**1**	0
Robot	0.0009	**0.9991**	1	0
Robot1	0.0001	**0.9999**	1	0
Robot1short	0.2678	**0.7322**	1	0
Robot1noisy	0.2680	**0.7320**	1	0

TABLE III
CLASSIFICATION OF TABLE-SETTING SEQUENCES.

1) Identifying (ir)relevant actions and objects: A priori, the system does not know which actions or objects are relevant for a task. Using the learned models, the probability of an action or object given the activity can be calculated. Those actions that occur several times per activity obviously have a higher probability, and those that are only rarely performed are much less likely.

P(actionT(A1) | inActivity(A1, Act)=True
∧ activityT(Act)=MakingBrownies
=⟨ TakingSomething:0.25,
PuttingSomethingSomewhere:0.15, Pouring:0.13,
OpeningSomething:0.13, ClosingSomething:0.08,
Stirring:0.08, Walking:0.03, TurningOnDevice:0.04,
Reading:0.03, [...] ⟩

P(objectActedOn(A1) | inActivity(A1, Act)=True
∧ activityT(Act)=CookingOmelette
= ⟨ Egg-Chickens:0.19, Cupboard:0.15, FryingPan:0.12,
VegetableOil:0.08, TableSalt:0.06, Bowl-Mixing:0.05,
Fork-SilverwarePiece: 0.05, [...] ⟩

2) Person-specific preferences: Some subjects put the frying pan back onto the stove, others put it into the sink after finishing cooking. Such preferences are implicitly learned by the models.

P(doneBy(A1) | inActivity(A1, Act)=True
∧ activityT(Act)=CookingOmelette
∧ actionT(A1)=PuttingSomethingSomewhere
∧ objectActedOn(A1)=FryingPan
toLocation(A1)=Sink)
= ⟨ P3: 0.32, P5: 0.31, P4: 0.18, P0: 0.17 ⟩

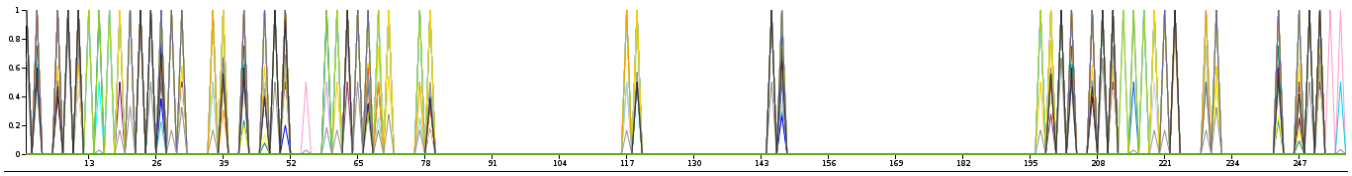

Fig. 6. Conditional probability distribution of the precedes-node in the TUM data set. Each curve corresponds to the first action in a pair (a_1), the values on the x-axis denote the set $o_1 \times a_2 \times o_2$, and the value of the curve is the conditional probability that a_1 performed on o_1 precedes a_2 performed on o_2. The very peaked distribution indicates distinct ordering constraints.

The subjects P3 and P5 put the frying pan into the sink several times while cooking in order to drain some spare oil. The remaining subjects put the pan back onto the stove and not into the sink.

VI. Discussion

As we demonstrated in this paper, human everyday activities like household chores, assembly tasks in a factory, or games show a significant partial ordering among their actions. However, this is not reflected in many data sets that have been often recorded in very controlled settings in which the sequence of actions is completely determined, resulting in an artificially imposed total ordering. Lower-level data, e.g. observations on the motion level, also shows a more linear structure since, in smooth motions, subsequent poses mainly depend on the previous ones and less on the global task context. This is why models that are based on the Markov assumption (HMM, CRF) perform well on this kind of data.

Regarding scalability, models that represent a partial order are more complex compared to those describing only a sequence, theoretically scaling quadratically with in the length of the sequence, the number of actions and the parameters. In practice, however, the conditional probability table representing the precedence relation is often sparse: Many combinations of actions and objects do not make sense and thus have zero probability (Figure 6), so that the table can efficiently be represented using decision trees [15]. Even without such optimizations, our implementation smoothly handles inference in models of about 40 segments with about 10 action and object classes. Compared to the inference, learning BLNs is generally much less of a problem because parameter learning of Bayesian networks comes down to counting. Training on 20,000 sequences runs very fast without problems.

VII. Conclusions

In this paper, we presented a system for modeling human activities based on Bayesian Logic Networks. The models are learned from observations and represent a full-joint probability distribution over the actions, their properties and their (partial) ordering. Therefore, they can not only be used for classifying activities, but also for more advanced reasoning on action-related properties.

We evaluated the system on two real-world data sets of human activities as well as synthetic data in order to analyze in detail the properties of the learned models. This evaluation shows that the approach outperforms models often used in activity recognition like Conditional Random Fields for common tasks since they are much less confused by the variation inherent in human activities.

VIII. Acknowledgments

This work is supported in part by the EU FP7 Projects *RoboEarth* (grant number 248942) and *RoboHow* (grant number 288533).

References

[1] D. Jain, S. Waldherr, and M. Beetz, "Bayesian Logic Networks," IAS Group, Fakultät für Informatik, Technische Universität München, Tech. Rep., 2009.

[2] D. Patterson, D. Fox, H. Kautz, and M. Philipose, "Fine-grained activity recognition by aggregating abstract object usage," in *Ninth IEEE International Symposium on Wearable Computers, 2005. Proceedings*, 2005, pp. 44–51.

[3] D. L. Vail, M. M. Veloso, and J. D. Lafferty, "Conditional random fields for activity recognition," in *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2007, pp. 1–8.

[4] R. Hamid, S. Maddi, A. Bobick, and I. Essa, "Structure from statistics-unsupervised activity analysis using suffix trees," *IEEE*, vol. 206, p. 7, 2007.

[5] Y. Shi, Y. Huang, D. Minnen, A. Bobick, and I. Essa, "Propagation networks for recognition of partially ordered sequential action," *cvpr*, vol. 02, pp. 862–869, 2004.

[6] A. Gupta, P. Srinivasan, J. Shi, and L. Davis, "Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Citeseer, 2009, pp. 2012–2019.

[7] S. Ekvall and D. Kragic, "Learning task models from multiple human demonstrations," in *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*. IEEE, 2006, pp. 358–363.

[8] J. Penberthy and D. Weld, "UCPOP: A sound, complete, partial order planner for ADL," in *proceedings of the third international conference on knowledge representation and reasoning*. Citeseer, 1992, pp. 103–114.

[9] H. Kautz and J. Allen, "Generalized plan recognition," in *Proceedings of the Fifth National Conference on Artificial Intelligence*, 1986, pp. 32–38.

[10] R. Goldman, C. Geib, and C. Miller, "A new model of plan recognition," in *Proceedings of the 1999 conference on uncertainty in artificial intelligence*, vol. 13, 1999, p. 14.

[11] R. M. Fung and B. Del Favero, "Backward Simulation in Bayesian Networks," in *UAI '94: Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence*, 1994, pp. 227–234.

[12] A. Quattoni, M. Collins, and T. Darrell, "Conditional random fields for object recognition," in *NIPS*. MIT Press, 2004, pp. 1097–1104.

[13] M. Tenorth, J. Bandouch, and M. Beetz, "The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition," in *IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS), in conjunction with ICCV2009*, 2009.

[14] F. De la Torre, J. Hodgins, J. Montano, S. Valcarcel, and J. Macey, "Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) Database," CMU-RI-TR-08-22, Robotics Institute, Carnegie Mellon University, Tech. Rep., 2009.

[15] N. Friedman and M. Goldszmidt, "Learning Bayesian Networks with Local Structure," in *Proceedings CUAI*, 1996.