A Bayesian Approach for Task Recognition and Future Human Activity Prediction

Vito Magnanimo[+], Matteo Saveriano[†], Silvia Rossi* and Dongheui Lee[†]

***Abstract*— Task recognition and future human activity prediction are of importance for a safe and profitable human-robot cooperation. In real scenarios, the robot has to extract this information merging the knowledge of the task with contextual information from the sensors, minimizing possible misunderstandings. In this paper, we focus on tasks that can be represented as a sequence of manipulated objects and performed actions. The task is modelled with a Dynamic Bayesian Network (DBN), which takes as input manipulated objects and performed actions. Objects and actions are separately classified starting from RGB-D raw data. The DBN is responsible for estimating the current task, predicting the most probable future pairs of action-object and correcting possible misclassification. The effectiveness of the proposed approach is validated on a case of study, consisting of three typical tasks of a kitchen scenario.**

I. Introduction

In these years, understanding human's intentions became a central theme in many robotics applications. These applications mainly belong to the field of human-robot co-working. Indeed, to help a human to accomplish a task, it is fundamental to be able to recognize what he is doing and to forecast which will be his next operations.

Many common daily-life activities have well defined patterns, i.e. they consist of a sequence of manipulated objects and performed actions. By recognizing these patterns, the robot can anticipate human needs and help him in a proactive way. An example of sequential activity is the task of preparing a particular recipe. Let us consider a kitchen scenario with a chef (human) and his assistant (robot). The assistant is trained for understanding the task performed by the chef. During the training it learns step-by-step how the recipe is composed. It learns the ingredients to use and the actions to mix them together. Then, it can actively help the chef, taking for him the ingredients when they are needed, or mixing together two ingredients.

Having this in mind, we propose a probabilistic approach for recognizing human tasks and predicting what he will do in the future. In our system, the contextual information to interpret is a sequence of manipulated objects and performed actions, with associated probabilities. The knowledge, represented in a probabilistic fashion, is an abstraction of the task(s), where the term task indicates a sequence of manipulated objects and performed actions.

The rest of paper is organized as follows: in Section II we make a brief overview of some related works. In Section III our architecture is described, while in Section IV the selected case study and our results are illustrated. Finally, Section V discusses conclusions and future developments.

II. Related Works

One of the main issues to solve in recognizing human activities is the problem of binding different information sources. This theme is addressed in [1] where a multimodal architecture is used for fusing and interpreting the input from different sources, like voice and gestures.

Other authors proposed instead to merge information from object and/or gesture recognition. In [2] an approach is proposed for learning the semantics of object-action relations by observation. Another approach based on Petri nets is proposed for learning human task in [3].

A possible way for implementing task recognition is to use probabilistic graphical models. Hidden Markov Models (HMM), Bayesian Networks (BN) and Dynamic Bayesian Networks (DBN) [4] are widely used for speech recognition [5] [6] and biosequence analysis [7], but they are used also for task modeling and recognition.

In [8], for example, a Bayesian approach is proposed to simultaneously estimate the object type, the performed action and the type of interaction between them. This approach uses a Bayesian Network in order to improve the recognition of objects and actions. Another Bayesian approach for recognizing human activities is proposed in [9], but it relies only on objects. In [10] a Bayesian Network models the interaction motions between paired objects in a human-object way and uses the motion models to improve the object recognition reliability. In this approach actions are not taken into account. In [11] nursing activities are recognized from nurses interactions with tools and materials using a DBN. The recognized activities are used for preventing the cause of medical accidents and incidents. A Bayesian conditional probability is used in [12] for context-aware activities of daily living (ADL) recognition. In ADL recognition, the object is used as a sort of context information to correct the estimated human action.

The prediction of future human operations is considered in [13]. A simple assembly task, where a human worker assembles a θ-shape assembly using bars, is considered. Using a Temporal Bayesian Network, the next used bar and the time instant in which the bar is needed are predicted. This

[+] Kuka Laboratories GmbH, Technologieentwicklung, Augsburg, Germany, vito.magnanimo@kuka.com

[†] Fakultät für Elektrotechnik und Informationstechnik, Technische Universität München, Munich, Germany matteo.saveriano@tum.de, dhlee@tum.de

* Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione, Universitá degli Studi di Napoli Federico II, Naples, Italy, silvia.rossi@unina.it

information is then used by the robot to select a proactive behaviour and to minimize the waiting time for both the human and the robot. In [13] the task and the action are unique and they are known from the beginning. Hence, task and action recognition are not considered.

Our goal is to estimate the current task and to predict the future activities, which are necessary information for a fruitful human-robot cooperation. The task is imagined as a sequence of objects and actions. We say that the task is a sequential activity with n time slices, where each time slice consists of a pair action-object.

Tasks are modeled using a DBN. Indeed, probabilistic approaches are proved to be more robust and accurate than deterministic ones. Moreover, in a sequential activity, the results in the previous time slice affect the results in the current one. As known, a *dynamic* network is suitable to model the inter-slice relationships. The DBN structure gives us also the possibility to implement a simple and effective algorithm to perform one-step prediction on the next activity (see Sec. III-C.1). Finally, despite many probabilistic approaches, any conditional relationship between action-object (or object-action) is assumed. We share, in principle, the same idea of [8]: the manipulated object affects the performed action and vice versa. This leads to a more complicated system, because both the object and action recognition algorithms are required, but makes the architecture more general and significantly increases the performance. Indeed, a technique is proposed to correct a misclassification of both the object and action recognizers (see Sec. III-C.2).

III. Task Modeling, Recognition and Prediction

The proposed architecture for modeling, estimating, and predicting the human task is depicted in Fig. 1. Raw data from an RGB-D camera (sensor module) are processed by the *Objects Tracking and Recognition (OTR)* module and by the *Human Actions Recognition (HAR)* module. These modules are used to extract from the raw data the n-best (most probable) objects and actions.

Finally, estimations from previous modules are used in the *task recognition* and *prediction* module, the core of our architecture, to identify the current task and predict the next manipulated object as well as the performed action.

A. Objects Tracking and Recognition

This module aims to track the position and the orientation of the objects present in the scene. Furthermore, we want to classify the objects. The module is divided in two submodules: *tracking* and *recognition*.

The tracking submodule manages the tracking process. It is based on the PCL tracking library[1] [14] which implements a particle filter [15]. It works by first segmenting the scene to cut off the table (where the objects are supposed to be) and then by clustering the points belonging to each object (Euclidean clustering).

[1]http://www.willowgarage.com/blog/2012/01/17/tracking-3d-objects-point-cloud-library

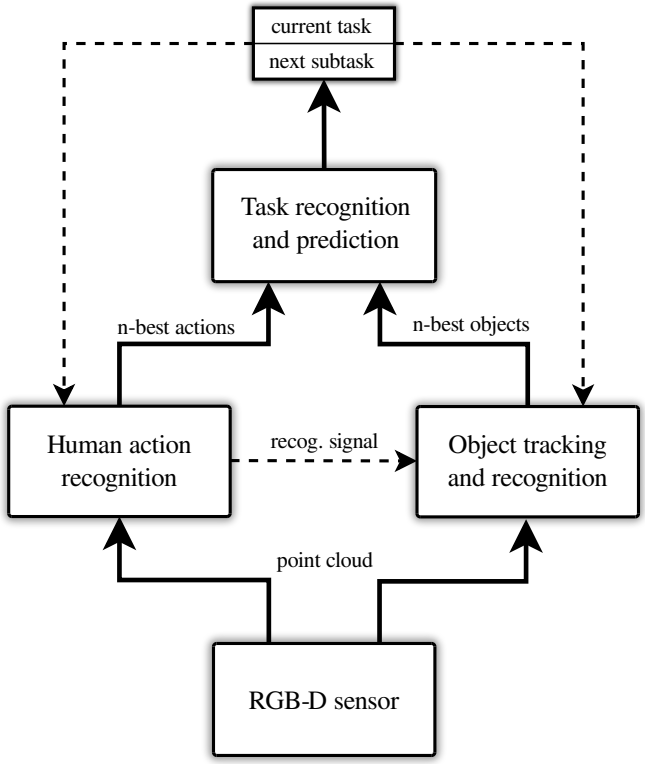

Fig. 1: System architecture

Then, it tracks the manipulated object, by matching the points of the model with the new points coming at every step. The process of extracting (and updating) the models is called *segmentation*. With manipulated object we simply mean the object closest to the user hand.

The recognition submodule performs a classification over every object, returning the probability the object belongs to the i-th class. The features computed from the Voxelized Shape and Color Histograms (VOSCH) descriptor [16] are used to train a Support Vector Machine (SVM) classifier [17]. VOSCH is rotation invariant[2], and it takes into account both color and shape information to increase the discriminative power of the descriptor.

We created a database of 10 objects, using 20 different views for each object: *cutting board*, *apple*, *pear*, *knife*, *cup*, *boiler*, *sugar box*, *tea box*, *coffee box* and *milk container*. We recorded point cloud and RGB data using a Microsoft Kinect sensor.

The recognition process is performed every time a recognition signal is raised by the *HAR* module (see Fig. 1), i.e. every time the user performs a particular action. We adopted this solution because the segmentation and recognition are computationally expensive, while the tracking algorithm can be tuned to work close to the Kinect frame rate.

The overall module is summarized in Algorithm 1, while an example of how the module works is shown in Fig. 2.

[2]Scale invariance is not of importance since our work space is quite limited.

Algorithm 1 Object Tracking and Recognition Algorithm

Data: PointCloud, RGB
if *first_iteration* **then**
  Table = segment_table(PointCloud)
end
if *recognition_signal* ***or*** *first_iteration* **then**
  Objects = segment_objects(PointCloud, Table)
  Descriptors = compute_VOSCH(PointCloud, RGB)
  ObjectClasses = classify_SVM(Descriptors)
end
ObjectPose = track_manipulated_object(Objects)

B. Human Actions Recognition

Recognizing human actions in daily life scenarios is a challenging problem. Gestures can be performed by different people in slightly different manners and observed from different view points. Moreover, since human actions are observed online, the gesture segmentation (i.e. the starting and ending points) is unknown.

In order to (partially) alleviate these problems, we adopted the invariant representation of motion proposed in [18]. This representation is invariant to roto-translations (useful when the user's pose changes) and linear scaling factors (useful when gestures are performed by different users). The representation consists of two scalar features (one for representing the position of a body, one for the orientation). Using these features, we trained an HMM [5], used to recognize new input motions. The sliding window approach in [18] is used to segment continuous gestures.

The recorded dataset consists of five repetitions of eight gestures: *rest*, *take*, *release*, *cut*, *pour*, *point*, *stop* and *start*. We collect data at *30Hz* using a Microsoft Kinect sensor, considering only the positions of *right hand*, *left hand*, *right elbow*, *left elbow* and *torso*.

All gestures are executed with one arm, except for the *rest* (i.e. idle gesture). *Take*, *release*, *pour* and *cut* are performed with the right arm, *point*, *stop* and *start* with the left.

C. Task Recognition and Filtering

The basic idea is to represent the knowledge of a task in a hierarchical way [19][20]. In particular, from the tracked object and the recognized action we infer the "local" knowledge of the task, that we will call *subtask*. With the subtask estimation we can infer the "global" knowledge that is the *task*. We are assuming that a task is composed by a set of subtasks, and every subtask is a pair *(action, object)*.

1) Proposed DBN: Despite other approaches, we chose to separate the object recognition from the action recognition. Objects and actions are separately recognized using classification algorithms, but they are coupled via the subtask node. In this way we can correct wrong or poor (low associated probability) estimations from both the object and action recognizers.

The task can be seen as a sequence of objects and actions, so it persists for a certain number of time steps. Estimating a task with a static Bayesian network is difficult, so we used a dynamic network. The designed DBN is shown in Fig. 3. Let us have a look at the network in the time slice t: there are two observable nodes, the tracked object O and the observed action A. These two observable nodes are used to recognize the current subtask S, that is a pair (a_i, o_j) where a_i is the *i*-th action and o_j is the *j*-th object. The obtained estimation is used to recognize the task node T that contains the information of the performed task. Every node in the network is a discrete node and every transition Matrix is a *CPT* (*Conditional Probability Table*). A Dynamic Bayesian Networks can "store" the knowledge of the previous instant and can use it for estimating the values of certain hidden variables at a time step $t + 1$. For doing so, it is important to define an inter slice topology. The inter slice topology is the connection between the network nodes at time step t and the network nodes at time step $t + 1$.

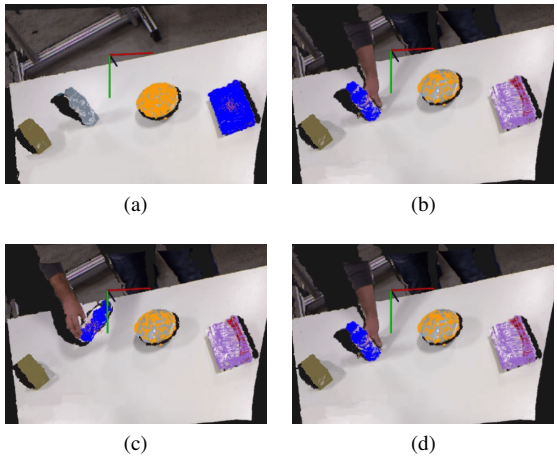

Fig. 2: Object tracking and recognition algorithm. (a) The objects are classified (different colors) and the object closest to the sensor is tracked (blue) when the system does not see human hands. (b) By approaching the boiler, it will became the new tracked object. (c) Tracking in action during a pour. (d) After the pour action, the boiler returns to the initial state.

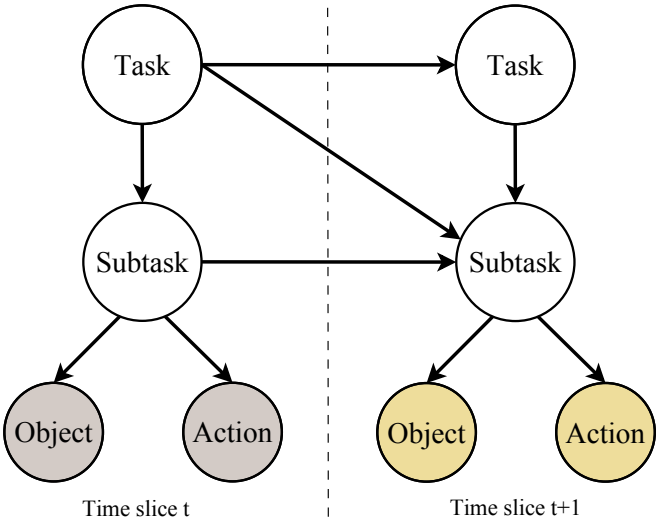

Fig. 3: Designed DBN: connections between two time slices

We used the topology in Fig. 3. To estimate the task at time $t+1$ we used the estimation of the task at time t. Instead, to estimate the subtask at time $t+1$, we used the subtask at time t alongside with the task at time t. The last choice gives us the possibility to "weight" the subtask prediction over the task knowledge.

From the designed network it is possible to estimate the current task and evaluate what the human is doing. We can also predict the next subtask and estimate what the human will do in the next step (one-step prediction).

To estimate the current task, or, in other words, to evaluate the distribution $p(T_t|T_{t-1}, S_t)$, we used the Boyen-Koller algorithm [21]. This is an approximate inference algorithm and it represents a good compromise between speed and accuracy.

The prediction of *n* successive subtasks (*n*-step prediction) in DBNs is still an open problem. On the contrary *1*-step prediction, i.e. to estimate the distribution $p(S_{t+1}|T_t, S_t,)$, is quite straightforward to implement. First, the distributions $p(S_t|S_{t-1}, T_{t-1}, a_t, o_t)$ and $p(T_t|T_{t-1}, S_t)$ must be evaluated. Then, the transition matrix A of the subtask nodes, namely $p(S_{t+1}|S_t)$, is extracted from the DBN. Finally, using the Bayes rule it is possible to evaluate $p(S_{t+1}|T_t, S_t)$ and to predict the next subtask S_{t+1}. This process can be summarized by the following formula:

$$\hat{x}_{t+1} = Ax_t \tag{1}$$

where $\hat{x}_{t+1}$ is the predicted next state, A is the transition matrix and x_t is the current filtered state.

2) Object and Action Correction: It is possible that one between the HAR and OTR module gives us wrong or poor (low associated probability) information. This information can be corrected or improved using the previous knowledge of the network. To this end we made only the assumption that the performed task is consistent, i.e. the human is performing the task correctly and he does not change the task during its execution.

Let us consider an example on the object case, since the action case is symmetric. At the time step t we have a prediction of the state (subtask) $\hat{x}_t$ performed at time $t-1$. With the prediction we have a probability related to every pair (a_i, o_j). Such probability is used as a correction coefficient for the object in the recognition process. If an object occurs more than once, we compute the correction coefficient summing up the probabilities.

After having calculated these correction coefficients, we multiply them with the probabilities given by the object tracking and recognition module, normalizing the result. The result of these multiplications is used to select the new class of the object. Obviously, we choose the class with the maximum value.

This procedure can be seen as a different application of the Bayes rule. The prediction $\hat{x}_t$ can be seen as a likelihood, the observed object is the measurement (or prior) and the result is the belief.

IV. Experimental Results

Our testing set up is depicted in Fig. 4a. We used two Kinect sensors, one for tracking the object, that is focused on the table; one for recognizing the human actions, that is focused on the human. The entire architecture is implemented in *C++* under ROS[3]. The main problem during the experiments was the segmentation of a subtask. We assumed that a subtask cannot occur for two consecutive time steps. This assumption makes the segmentation process easier.

A. Case Study

We tested our architecture on three different tasks: *prepare food*, *prepare tea* and *prepare coffee*. Objects involved in these tasks are: cutting board (CB), apple (A), pear (P), knife (K), cup (C), boiler (B), sugar box (S), tea box (T), coffee box (CF) and milk container. The object dataset is shown in Fig. 4b. Actions involved in these tasks are: take, release, pour and cut. We used the point gesture to generate the recognition signal. Every subtask is represented using a string action-object. We assume that the task is correctly recognized if the filtering gives a task probability greater than 60%.

The preparing food task is a distinctive task since this task shares a minimum number of states (two) with the other two tasks. The other two tasks are designed to be ambiguous. In other words, the execution sequence of both tasks is the same until a certain time slice.

We created a synthetic training set for the DBN, providing three training sequences for each task. Recall that our DBN receives and manages discrete data, in particular, a set of labels. Hence, it is useless in this context to create the set of labels using the real data. Each training sequence consists of the most probable object, the most probable action, related subtask and task for each time slice. To indicate the final state of a task, a *nothing* subtask is introduced. The meaning of this subtask is no object and no action.

During the design we made the assumption that a subtask cannot occur in two adjacent time slices. For example, if we have a subtask (a_x, o_y) at time t, this cannot occur again in $t+1$. This helps us to easily segment the continuous stream of data coming from the object and action recognition modules. In practice, we update the observables node of the DBN only if the current pair is not same as the previous one.

In all the experiments we started the filtering (task recognition) from the time slice $t = 2$. This is due to the network implementation that cannot allow this operation from the first time step. Of course, the prediction also starts at $t = 2$ and affects the next time slice. Moreover, we will not report task and subtask with probabilities $< 5\%$.

Prepare Tea I: In this experiment the task of prepare tea is executed. The task is a sequence of 9 subtasks, namely (Take, Cup)-(Release, Cup)-(Take, Boiler)-(Pour, Boiler)-(Release, Boiler)-(Take, Sugar Box)-(Release, Sugar)-(Take, Tea Box)-(Release, Tea Box). The subtask sequence is exactly one of the sequences in the training set. The task is ambiguous,

[3]http://www.ros.org

(a) (b)

Fig. 4: (a) Testing set up. (b) object dataset.

hence it can be a prepare coffee until the Tea is taken. The ambiguity affects the task recognition (see Fig. 5), in fact prepare tea and prepare coffee have the same probability until the time slice $t = 8$ (when the tea is taken).

Fig. 6 contains the prediction results. The indecision in the evaluation of the performed task can be seen also in this process. In fact, at the time slices $t = 3$ and $t = 8$, the network predicts as possible subtasks both take coffee and take tea, which cannot occur together. Among the predictions at time slice $t = 6$ there is also the nothing subtask, that is the final state of a task. This strongly depends on the training data. In fact, in our training set, 2 of the three sequences have the boiler as final object.

The tasks of prepare tea and prepare coffee are indistinguishable until the tea (or the coffee) is taken. As expected, the network detects this situation, assigning the task to prepare tea and prepare coffee with exactly the same probability. Also the prediction is correct, all the predicted subtask, except for the take coffee in time slice $t = 8$, can be in both sequences.

Prepare Tea II: In this experiment we tested a different prepare tea sequence (Take, Cup)-(Release, Cup)-(Take, Tea Box)-(Release, Tea Box)-(Take, Boiler)-(Pour, Boiler)-(Release, Boiler)-(Take, Sugar Box)-(Release, Sugar Box). This sequence does not belong to the dataset. The sequence is a prepare tea where we have switched the order of the boiler and the sugar. The sequence is not ambiguous from time slice $t = 3$, when the tea is taken, to the end. Filtering results are shown in Fig. 7. The task is correctly recognized (99.5%) at time slice $t = 3$ and maintained also when a different object is manipulated. Same results are obtained for the prediction (Fig. 8), that is not affected by the switch.

Prepare food: In the last experiment we tested the object correction process[4]. The testing sequence is a prepare food: (Take, Cutting Board)-(Release, Cutting Board)-(Take, Cup)-(Release, Cup)-(Take, Apple)-(Release, Apple)-(Take, Knife)-(Cut, Knife)-(Release, Knife). The action recognition module is able to recognize the take, but the object recognition usually fails in recognizing knife because the knife is too small to be correctly detected from point clouds of the low cost RGB-D sensor.

For the testing process we used artificial data. In particular, we generated an *unknown* object at time $t = 7$. Unknown object here means that we assigned the same probability for every class, i.e. the object is not classified correctly. Having ten classes, each class has a probability of 10%.

From the prediction at $t = 7$ we know that the most probable subtask at $t = 8$ is (take, knife), with the probability of 99%. So, the correction coefficient for the knife is 0.99. The other objects have very low coefficients that allow to increase the probability of knife. Indeed, after multiplying and normalizing, the most probable belonging class for the tracked object is knife. The results of this procedure are shown in Fig. 9.

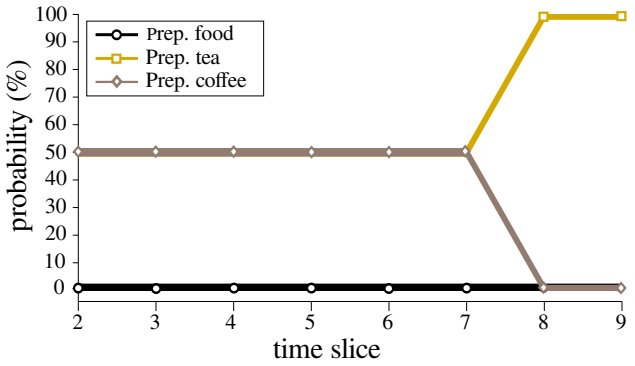

Fig. 5: Prepare tea I: task recognition results.

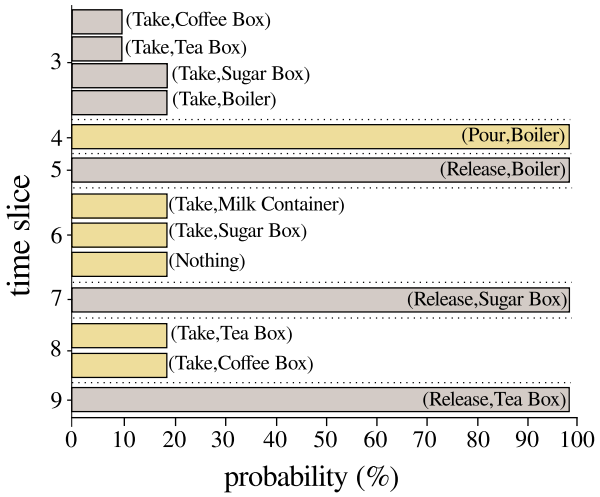

Fig. 6: Prepare tea I: prediction results.

V. Conclusions and Future Developments

We proposed an approach to recognize the task that the human is executing, and to predict which will be the next performed action and the manipulated object.

To this end, we designed a modular architecture to transform raw data from a RGB-D sensor into an estimation of an action and an object class. A Dynamic Bayesian Network, able to represent the task in a probabilistic fashion, is also designed and implemented. With the proposed architecture is possible to infer simple tasks, in a way that is robust to variations in the execution sequence. The proposed network is also capable to predict the next subtask (one-step prediction)

[4]For the sake of simplicity we show only the object correction. Remember that for the actions the process is the same.

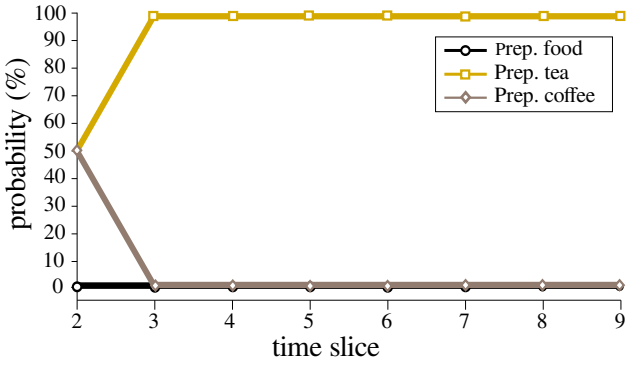

Fig. 7: Prepare tea II: task recognition results.

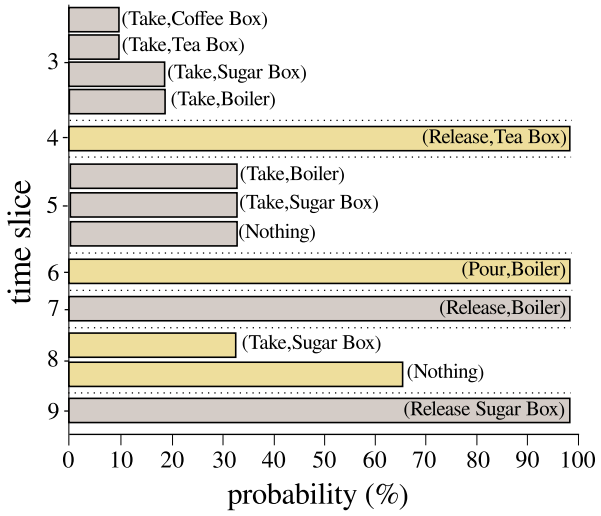

Fig. 8: Prepare tea II: prediction results.

and to correct, using its knowledge, wrong or poor object or action estimation. Experiments on synthetic and real data showed the effectiveness of our approach.

The proposed approach has been tested only considering observations from the same user. Tests on different users and public datasets, as well as considering real human-robot interaction scenarios, will be part of our future work.

Acknowledgements

This work has been partially supported by the European Community within the FP7 ICT-287513 SAPHARI project and Technical University Munich, Institute for Advanced Study, funded by the German Excellence Initiative.

References

[1] S. Rossi, E. Leone, M. Fiore, A. Finzi, and F. Cutugno, "An extensible architecture for robust multimodal human-robot communication," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013, pp. 2208–2213.

[2] E. E. Aksoy, A. Abramov, J. Dörr, K. Ning, B. Dellen, and F. Wörgötter, "Learning the semantics of object-action relations by observation," *Int. J. Rob. Res.*, vol. 30, no. 10, pp. 1229–1249, 2011.

[3] G. Chang and D. Kulic, "Robot task learning from demonstration using petri nets," in *IEEE International Symposium on Robot and Human Interactive Communication*, 2013, pp. 31–36.

[4] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

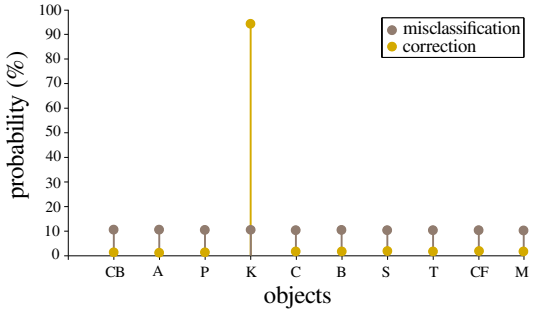

Fig. 9: Object misclassification: the same probability (10%) is initially assigned to all the objects. Using the subtask prediction, the manipulated object (knife) is correctly classified with a probability of 94%.

[5] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, 1989, pp. 257–286.

[6] G. Zweig, "Bayesian network structures and inference techniques for automatic speech recognition," *Computer Speech & Language*, vol. 17, pp. 173–193, 2003.

[7] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[8] A. Gupta and L. S. Davis, "Objects in action: An approach for combining action understanding and object perception," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[9] V. Osmani, S. Balasubramaniam, and D. Botvich, "A bayesian network and rule-base approach towards activity inference," in *IEEE Vehicular Technology Conference*, 2007, pp. 254–258.

[10] S. Ren and Y. Sun, "Human-object-object-interaction affordance," in *Robot Vision (WORV), 2013 IEEE Workshop on*, 2013, pp. 1–6.

[11] T. Inomata, F. Naya, N. Kuwahara, F. Hattori, and K. Kogure, "Activity recognition from interactions with objects using dynamic bayesian network," in *Proceedings of the 3rd ACM International Workshop on Context-Awareness for Self-Managing Systems*, 2009, pp. 39–42.

[12] J. Fu, C. Liu, Y.-P. Hsu, and L.-C. Fu, "Recognizing context-aware activities of daily living using rgbd sensor," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013, pp. 2222–2227.

[13] W. Kwon and I. H. Suh, "A temporal bayesian network with application to design of a proactive robotic assistant," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 3685–3690.

[14] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 1–4.

[15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[16] A. Kanezaki, Z.-C. Marton, D. Pangercic, T. Harada, Y. Kuniyoshi, and M. Beetz, "Voxelized shape and color histograms for rgb-d," in *Intelligent Robots and Systems (IROS), Workshop on Active Semantic Perception and Object Search in the Real World*, 2011.

[17] C. M. Bishop, *Pattern recognition and machine learning (information science and statistics)*. The MIT Press, 2007.

[18] M. Saveriano and D. Lee, "Invariant representation for user independent motion recognition," in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2013.

[19] Y. Zhang and Q. Ji, "Active and dynamic information fusion for multisensor systems with dynamic bayesian networks." *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 36, no. 2, pp. 467–472, 2006.

[20] S. Park and H. Kautz, "Hierarchical recognition of activities of daily living using multi-scale, multiperspective vision and rfid. the 4th," in *IET International Conference on Intelligent Environments*, 2008.

[21] K. P. Murphy, "Dynamic bayesian networks: Representation, inference and learning," Ph.D. dissertation, UC Berkeley, 2002.