Inferring What to Imitate in Manipulation Actions by Using a Recommender System

Nichola Abdo Luciano Spinello Wolfram Burgard Cyrill Stachniss

***Abstract*— Learning from demonstrations is an intuitive way for instructing robots by non-experts. One challenge in learning from demonstrations is to infer *what* to imitate, especially when the robot only observes the teacher and does not have further knowledge about the demonstrated actions. In this paper, we present a novel approach to the problem of inferring what to imitate to successfully reproduce a manipulation action based on a small number of demonstrations. Our method employs techniques from recommender systems to include expert knowledge. It models the demonstrated actions probabilistically and formulates the problem of inferring what to imitate via model selection. We select an appropriate model for the action each time the robot has to reproduce it given a new starting condition. We evaluate our approach using data acquired with a PR2 robot and demonstrate that our method achieves high success rates in different scenarios.**

I. Introduction

Learning from demonstrations is a promising approach in robotics as it exploits activities shown by a teacher to speed up the learning process of the robot. There are two major challenges in this paradigm, namely the questions of *what* to imitate and *how* to imitate [4]. The first one aims at identifying the features, constraints, or symbols that are relevant for reproducing an action, whereas the latter addresses the issue of generating feasible trajectories of the robot's manipulators when imitating a motion. The focus of this paper is on *what* to imitate. Given a set of demonstrations of an action, we want to infer the relevant aspects of these demonstrations so that a robot can replicate the action when the actual starting configuration differs from the ones seen during the demonstrations. In general, it is hard to infer which aspects are important to successfully replicate an action. This is particularly challenging if the number of demonstrations is small.

In this work, we propose an approach to deal with the identification of the relevant features that describe an action. The most direct solution to this problem is to detect these features by making use of large sets of training data. From a robotics standpoint, it is highly impractical to generate them for every action. Our take to this problem is different: we only use a few training examples to build the model for the action by leveraging expert knowledge in a non-greedy manner. We encode this expert knowledge by borrowing ideas from the recommender systems theory. Recommender systems typically identify patterns in user preferences either through leveraging similarities between users [8], [13] or by analyzing a user's purchase history [3], [17]. A widespread application of such techniques is the product recommendation system of large online retail stores. The idea of our approach is to let multiple experts provide recommendations about which set of features is relevant for an action. These recommendations are functions of the state perceived by the robot and of the training demonstrations. The experts are users that have in-depth knowledge of robot manipulation. Their rules and their recommendations are collected offline, before training, and without knowledge of which specific action will be demonstrated. Based on the many recommended features, our system builds multiple probabilistic models of the action. Thus we formulate a model selection problem to evaluate which is the best explanation of the perceived state each time a new condition is presented to the robot.

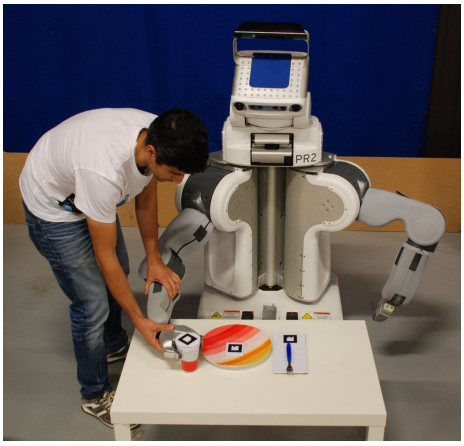


Fig. 1. The teacher demonstrating a manipulation action to the PR2 robot. He shows the action of how to place a cup in a certain pose relative to a plate and a fork. The robot has to infer relevant features of the action given a small number of demonstrations.

We implemented and tested our approach in simulation and by using data recorded by kinesthetic teaching on a real PR2 robot. As we illustrate in the experimental evaluation, we are able to successfully replicate several kinds of tabletop actions based on a small number of demonstrations. Overall, we obtain high success rates for different tabletop manipulation action scenarios.

II. Related Work

Learning from demonstrations is a framework for teaching actions or tasks to robots [4]. There exist a number of approaches that address the issue of determining the relevant constraints for reproducing a demonstrated motion, i.e. *how*

All authors are with the University of Freiburg, 79110 Freiburg, Germany. Cyrill Stachniss is also with the University of Bonn, Inst. of Geodesy and Geoinformation, 53115 Bonn, Germany. This work has partly been supported by the German Research Foundation under research unit FOR 1513 (HYBRIS) and grant number EXC 1086.

to imitate [6], [10]. Calinon *et al.* [5] presented an approach that models the trajectories of the robot arm as a mixture of Gaussians. When reproducing a demonstrated action, the robot generates trajectories optimized with respect to a cost function that takes into account the spatial and the temporal correlations between the features along the trajectories. Eppner *et al.* [7] and Mühlig *et al.* [16] consider the variance in the demonstrations to determine less relevant parts of the tasks. The approach by Asfour *et al.* [2] models demonstrated arm movements using hidden Markov models and detects key points across demonstrations that the robot needs to reproduce. Similarly, Kulic *et al.* [14] use hidden Markov models to encode and reproduce demonstrated actions.

Other approaches focus on learning the relevant features or frames of reference for generalizing demonstrated actions, i.e. *what* to imitate. The method of Abdo *et al.* [1] analyzes the variations in the state during the demonstrations to identify the preconditions and effects of the individual actions. Veeraraghavan and Veloso [19] also learn symbolic representations of actions for planning by instantiating preprogrammed behaviors and learning the corresponding preconditions and effects. Jäkel *et al.* [9] use demonstrations to generate a so-called strategy graph that segments tasks into sub-goals. An evolutionary algorithm is used to eliminate irrelevant spacial and temporal constraints using a motion planner in simulation.

Song *et al.* [18] propose an approach that models the relations between object- and action-related features using a Bayesian network for learning strategies of grasping objects. Konidaris and Barto presented an approach for choosing between different state abstractions in a hierarchical reinforcement learning context where an agent learns different options (macro actions) out of primitive ones [11]. They applied their approach when segmenting demonstrated trajectories into different options represented in a skill tree [12]. Each option is assigned to a different abstraction that defines a small subset of relevant variables using a trade-off between model likelihood and model complexity. Our approach also considers several state/feature abstractions when learning a new action. However, we do not tackle the problem of decomposing tasks into sequences of actions.

Our approach relies on expert knowledge to make recommendations about subsets of features to use for learning new actions. This can be seen as a form of content-based recommendation systems, which make recommendations based on previously indicated user preferences [3], [17]. This typically requires learning user profiles describing which product categories a person is interested in. Similarly, our system recommends sets of features that could be used to explain a demonstrated action also from different initial conditions. The recommendations are computed by using the observed feature values in the demonstrations. Additionally, they leverage the expert knowledge about different manipulation actions. Related to this topic is the work of Matikainen *et al.*, who have applied concepts of recommender systems in the context of action recognition in videos [15]. Their approach is based on collaborative filtering and recommends which classifiers to use for addressing a certain vision task.

III. Learning an Action from Kinesthetic Demonstrations

The idea of this work is to learn an action through kinesthetic demonstrations where the teacher moves the manipulator of the robot from different starting states to the intended goal state of the action. We consider point-to-point tabletop actions that are defined by a start-to-goal motion of the robot's end-effector while it interacts with objects in its workspace.

The typical purpose of such actions is to reach a goal configuration. This goal configuration is often not a single state but all states that satisfy an unknown set of constraints. Often, some of the constraints can be described by the geometrical relationships (relative distances, orientations, etc.) of the objects to each other and to the robot. Actions such as sorting or tidying up also depend on properties of the objects like their colors, types, sizes, etc.

Given only a small number of demonstrations, an action can be "explained" in a number of different ways. For example, features describing the poses of some objects can be seen as relevant or irrelevant to the action depending on the observed variations in the starting conditions. Different explanations often conflict or restrict the ability of the robot to generalize that action to new situations. Note that we assume that the teacher demonstrates actions without errors.

We describe an action at any discrete point in time t by a collection of features. We compute features based on the perceived state. We define two different kinds of features: features that describe object properties and features that describe pairwise relations between objects. "Object color" is an example for the first kind, "distance" for the second. We generate features by feature functions:

$$f(o_1) \to \mathbb{R}, \tag{1}$$

$$g(o_1, o_2) \to \mathbb{R}, \tag{2}$$

where o is an object from $\mathcal{O}$, the set of all objects in the scene and the robot end-effectors. We define a feature vector for the action as:

$$\mathbf{f} = [f, \ldots, f_{N_1}, g_1, \ldots, g_{N_2}], \tag{3}$$

where $M = N_1 + N_2$ are the number of available feature functions. Note that all f are computed $\forall o \in \mathcal{O}$, and all g with $\forall \{o_1, o_2\} \in \binom{|\mathcal{O}|}{2}$ (the two-combinations of $|\mathcal{O}|$). By design, $\mathbf{f}$ is a high-dimensional feature vector. Assuming a time-discrete system, we represent an action by a sequence of feature vectors over time:

$$\mathcal{F} := (\mathbf{f}_{t_s}, \ldots, \mathbf{f}_{t_e}), \tag{4}$$

where t_s and t_e are respectively the starting and ending time steps of a demonstration.

Note that we do not address the perception problem in this paper. Rather, we assume that the robot can identify relevant objects in the scene along with their poses. In our current implementation, we solve this by using fiducial markers attached to the objects and an out-of-the-box detector.

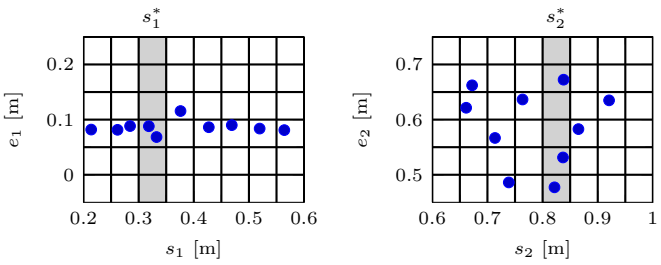


Fig. 2. An example of two feature distributions for the action of placing a grasped object A on top of another object B, based on ten demonstrations. The left side shows feature 1, which describes the pose of A relative to B along the x direction. The right side depicts feature 2 describing the pose of the gripper relative to the robot's torso frame along the x direction. Given a new starting point s_i^*, the distribution of the goal states for the first feature is more concentrated compared to the second one. This results in a higher likelihood for the first feature dimension, indicating a higher relevance to the action compared to feature 2.

This work is also not concerned with modeling or generalizing trajectories, i.e., *how* to imitate. Rather, we only consider the feature values at the start and end of an action as relevant. Thus, we rewrite Eq. (4) as:

$$\mathcal{F} := (\mathbf{f}_{t_s}, \mathbf{f}_{t_e}). \tag{5}$$

A. Modeling an Action

Learning an action is the process of building a model for the action that is based on the demonstrations $\mathcal{D}$. To describe an action, we are interested in learning a probability density function that models the relations between features at the start and the end of an action.

For an action a, we assume that each element of the vector $\mathbf{f}$ is independent from the others. Let s_i be the i-th dimension of $\mathbf{f}_{t_s}$ and e_i the i-th dimension of $\mathbf{f}_{t_e}$, we construct a bivariate probability density function ϕ_i that describes the start and goal distribution for each feature in Eq. (5):

$$\phi_i(s_i, e_i) := \eta_i I_i(\lfloor s_i \rfloor, \lfloor e_i \rfloor), \tag{6}$$

where the operator $\lfloor \cdot \rfloor$ is a quantization operator that returns the bin in the histogram I_i that corresponds to dimension i. The term η is a normalizer. We learn the entries of I_i by accumulating data from the training set $\mathcal{D}$.

When the robot reproduces an action from a new initial state, only $\mathbf{f}_{t_s}^*$ is available but not $\mathbf{f}_{t_e}^*$. We do not have direct access to the goal features because, in general, the action has never been executed starting from $\mathbf{f}_{t_s}^*$. Thus, we compute $\phi_i \mid \mathbf{f}_{t_s}^*$, the conditional bivariate distribution. We do this by taking the column vector of I relative to $\lfloor \mathbf{f}_{t_s}^* \rfloor$ for each i-th feature and then calculating the normalization. The shape of this distribution is important as it models the distribution of possible goals in $\mathcal{D}$ with respect to a given starting pose in the i-th feature dimension. If the distribution is concentrated in a few possible values, we can interpret it as an indication of goodness. According to the observed demonstrations, the goal state corresponding to that feature is known with high certainty. Fig. 2 depicts an example.

To evaluate the uncertainty in this distribution, we compute the entropy H_i of the conditional distribution $\phi_i \mid \mathbf{f}_{t_s}^*$, where the entropy of a discrete random variable X is:

$$H(X) = -\sum P(X) \ln P(X). \tag{7}$$

Based on the entropy value H_i, we define the model likelihood of an action a as:

$$p(\mathbf{f}_{t_s}^* \mid \mathbf{\Phi}) := \prod_i e^{-H_i}, \tag{8}$$

where $\mathbf{\Phi}$ is the set of all ϕ. Note that if the new starting state for some feature dimension corresponds to a bin where no data points are available, we use the distribution over all starting states observed in the demonstrations for computing the entropy of the conditional distribution.

The problem of this procedure lies in the dimensionality of $\mathbf{f}$. Many features, each measuring different aspects of the scene, play a role in Eq. (8). These features may have contrasting effects and render the action unlikely. This is especially true when data is scarce. The standard solution is to use large amounts of data to add samples for Eq. (6). This, however, is impractical in our application.

B. Feature Selection by Using a Recommender System

From a robotics standpoint, it is impractical to generate large sets of training examples for each action. Instead of collecting a large amount of data, we seek to reduce the dimensionality of the problem.

The key element of our approach is to perform feature selection by means of a recommender system. This system proposes a portfolio of various low dimensional feature spaces that are able to explain *an* action. We compute these spaces from a set of feature functions and they are provided by domain experts. The experts are users who have in-depth knowledge of robot manipulation. We collect their recommendations offline, before training, and without knowledge of the demonstrated action.

For brevity, we define a template $\mathcal{T}$ that is a a set of feature functions. A template is not necessarily tailored for one action and may be used for describing more than one action. In practice, an expert gives an informed opinion about what is *usually* relevant given $\mathcal{D}$. We consider the case of multiple experts suggesting multiple templates.

The next step is to define a way of making use of the expert knowledge. This can be interpreted as a form of content-based recommender systems. Such systems predict the preference of a user by analyzing his profile or purchase history. In our context, we aim to predict which template to use from an expert by analyzing the teacher demonstrations and the new starting state from which the robot has to reproduce the action. For example, if the teacher demonstrates to the robot how to place object A on top of object B, one expects small variations in the relative pose between the two objects. Therefore, an expert could recommend a template involving these features for reproducing the action. Note that the same template (features) can be recommended for the action of placing object C inside or next to object D.

To apply this theory to our problem, we need each expert e to define a *relevance* function $b(\cdot)$, which performs

a selection of templates based on the perceived situation and the demonstrations $\mathcal{D}$. The Boolean function $b(\mathcal{T}_i, \mathcal{D})$ is *true* iff conditions defined on features that the expert defines hold. An example for that is: $b(\mathcal{T}_i, \mathcal{D}) = 1$ iff there is a change bigger than σ in any feature dimension belonging to $\mathcal{T}_i$ and the gripper moved an object.

Based on $b(\cdot)$, we build a binary rating matrix $\mathbf{E}$. For clarification, we illustrate an example with three experts and only three relevance functions:

$$\mathbf{E} = \begin{array}{c|ccc} & e_1 & e_2 & e_3 \\ \hline \mathcal{T}_1 & b_1(\mathcal{T}_1, \mathcal{D}) & b_2(\mathcal{T}_1, \mathcal{D}) & b_3(\mathcal{T}_1, \mathcal{D}) \\ \mathcal{T}_2 & b_1(\mathcal{T}_2, \mathcal{D}) & b_2(\mathcal{T}_2, \mathcal{D}) & b_3(\mathcal{T}_2, \mathcal{D}) \\ \mathcal{T}_3 & b_1(\mathcal{T}_3, \mathcal{D}) & b_2(\mathcal{T}_3, \mathcal{D}) & b_3(\mathcal{T}_3, \mathcal{D}) \end{array} \tag{9}$$

where e.g.: $\mathcal{T}_1$ is the distance of an object to the robot and $\mathcal{T}_3$ is color. We can now select the features for the action demonstrated by $\mathcal{D}$. For this, we take all templates related to the K rows that have at least a 1. We call this set Θ. Each $\mathcal{T} \in \Theta$ generates a low dimensional feature space of size $L \ll M$. Our aim is not to find a minimum set of features but to find a valid set able to describe an action.

Note that learning stops here, i.e., models are fit and evaluated *each time* a request for reproduction of an action arrives. The reasoning behind this choice is that a template consists of feature functions. Feature functions depend on the number of objects and on the perceived state of the environment. With our technique, we want to be able to generalize to changing conditions between training and testing.

IV. Predicting and Reproducing an Action

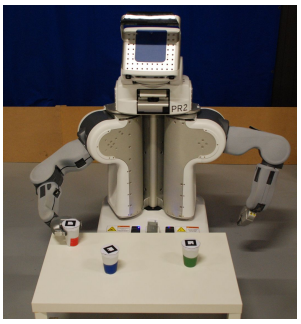
After completing the training phase, Θ is available and we can use our system to allow the robot to reproduce the action. Given the new start position $\mathbf{f}^*_{t_s}$, the system has to determine how to successfully complete the action. The idea is to select the best feature subset $\mathcal{T}_i \in \Theta$ that explains the perceived state to then reproduce the action.

As a first step, we instantiate all feature functions that occur in $\mathcal{T}_i$ by using the training data $\mathcal{D}$ with respect to the current number of objects and other aspects of the perceived state. We compute the likelihood of the action by evaluating Eq. (8) but considering only the feature dimensions of $\mathcal{T}_i$. At this point, the system has to select which model from the ones proposed by the templates is the one that best represents reality. We address this as a model selection problem for selecting a template. For each $\mathcal{T}_i \in \Theta$, we compute a score β_i that combines model fitting and maximizing the number of features used:

$$\beta_i = -2\ln(p(\mathbf{f}^*_{t_s} \mid \hat{\mathbf{\Phi}}_i)) - \alpha L_i \ln(|\mathcal{D}|), \tag{10}$$

where $\hat{\mathbf{\Phi}}_i$ are the distributions related to the features of $\mathcal{T}_i$. The first term of Eq. (10) is the likelihood and the second term, weighted by α, encourages the usage of templates consisting of a large number of features. We select the best template $\mathcal{T}^*$ as:

$$\mathcal{T}^* = \underset{i}{\operatorname{argmin}}(\beta_i). \tag{11}$$



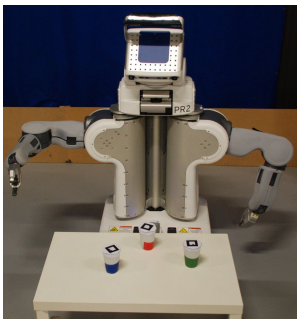
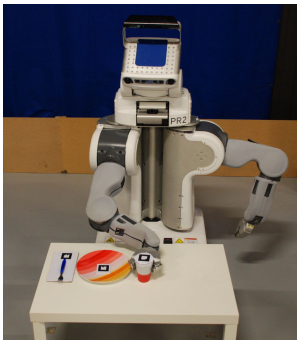


Fig. 3. We demonstrated to the robot how to reach for a specific object (red cup). The positions of the other objects on the table do not change in the demonstrations.



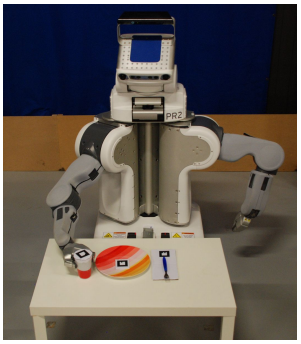


Fig. 4. We demonstrated to the robot how to place a grasped cup next to a plate on the table. The cup is always placed on the side opposite to that of the fork.

We carry out this procedure for each new action reproduction request as it depends on the start state $\mathbf{f}^*_{t_s}$. In this way, we do not commit to a model before seeing the state to start the reproduction from. Instead, we keep multiple possible explanations of the action that we select depending on the starting condition. After selecting $\mathcal{T}^*$, the final task is to reproduce the action.

In this paper, we aim at reproducing the action with the closest resemblance to a demonstration. Specifically, the robot reproduces the trajectory found in $\mathcal{D}$ by:

$$\mathcal{F}^* = \operatorname{argmin} \|\mathbf{f}_{t_s} - \mathbf{f}^*_{t_s}\|_{\mathcal{T}^*}, \quad \forall \mathbf{f}_{t_s} \in \mathcal{D}, \tag{12}$$

where $\|\cdot\|_{\mathcal{T}^*}$ is the distance that considers only the dimensions selected by $\mathcal{T}^*$ and the i-th dimension is weighted by e^{-H_i}. In this way, we can introduce a confidence measure on the selection of the trajectories based on the same criterion we use for computing the model likelihood. For executing the selected trajectory, the robot uses its trajectory execution system to reproduce the most similar demonstration transformed in the relevant frames defined by the template features.

V. Experimental Evaluation

This section summarizes the evaluation of our approach conducted using kinesthetic demonstrations of tabletop actions recorded with a Willow Garage PR2 robot. As we do not address perception in the scope of this paper, we used fiducial markers attached to the objects and measured their

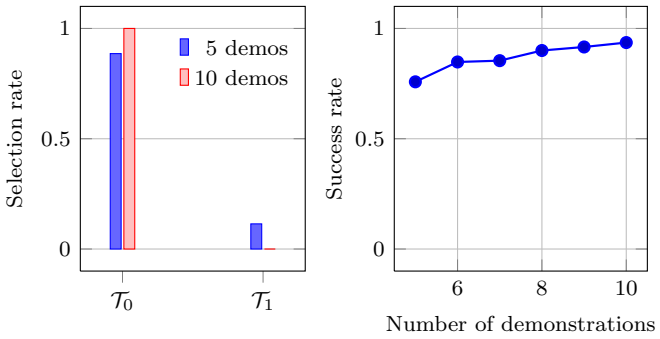


Fig. 5. Results for the action of placing a grasped object onto another object on the table. The left figure shows the selection of templates given five and ten demonstrations for learning the action. Out of the large number of available templates, the recommender system suggested only two. Our system selected $\mathcal{T}_0$ most of the time, as this template considers features describing the pose of the gripper and the grasped object relative to the target object on the table. The right figure shows that the success rate increases up to 93.6% with ten demonstrations.

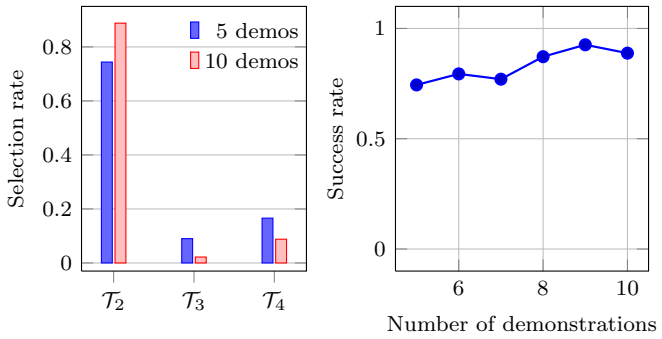


Fig. 6. Results for the action of reaching for a specific object. The left figure shows that by increasing the number of demonstrations, the system prefers $\mathcal{T}_2$ to the other templates. This template contains features describing the pose of the gripper relative to the target object. The right figure shows the success rate as a function of the number of training demonstrations. The success rate increases with the number of training demonstrations, reaching up to 92.6%.

pose using a camera mounted on the robot's head, see Fig. 1. We present our evaluation on three scenarios.

A. Placing an Object on Another

In this experiment, we provided the robot with ten demonstrations of how to place a grasped cup on a coaster. We ran 500 simulation runs where the robot has to reproduce the action from different starting poses of its gripper and the coaster. The success rates are shown in Fig. 5-right for increasing numbers of initial demonstrations. The figure shows that the robot is able to solve more cases with the increase in the number of demonstrations used in training. Fig. 5-left shows the selection of templates given five and ten demonstrations for learning the action. The robot explained the action using two templates, $\mathcal{T}_0$ and $\mathcal{T}_1$, achieving a success rate of 75.8% when given five demonstrations. With ten demonstrations, the robot solved 93.6% of the cases using $\mathcal{T}_0$. This template includes features that describe the poses of the gripper and the cup relative to the coaster. On the other hand, $\mathcal{T}_1$ contains features describing the pose of the coaster relative to the robot torso frame and less successfully explains the action.

B. Reaching for a Specific Object

In this experiment, we consider an action where the robot has to reach for a specific object. We provided only ten demonstrations of reaching for a red cup placed on the table in front of the robot. In all demonstrations, we varied the starting position of the cup while leaving two other cups (green and blue) in fixed positions on the table, see Fig. 3. We ran experiments by starting the action from 500 different random starting poses of the gripper, from different placements of all the cups, and by changing the table height in simulation. In each run, we recorded the number of times the robot correctly executed the action by reaching for the red cup.

Fig. 6-right shows the results. When using the full training dataset, the overall success rate is 88.8%. We quantified the influence of the number of training demonstrations on the success rate by increasing the number of training demonstrations from five to ten. As expected, the success rate increases with the size of the training set. Additionally, we analyzed which templates are selected over the varying number of training demonstrations, see Fig. 6-left. The system frequently selected three templates, $\mathcal{T}_2$ - $\mathcal{T}_4$, when reproducing the action from varying starting poses. $\mathcal{T}_2$ contains features describing the pose of the gripper relative to the red cup. On the other hand, $\mathcal{T}_3$ and $\mathcal{T}_4$ contain features involving the poses of the other two cups as well as the pose of the gripper relative to the robot. Combined, they explained 74% of the 500 trials given five demonstrations. With ten demonstrations, $\mathcal{T}_2$ successfully explained 88.8% of all the initial configurations.

We compared our method with Abdo *et al.* [1]. That method is unable to reproduce the action starting from positions that are substantially different from the ones demonstrated by the teacher. This is due to learning false-positive constraints related to the poses of the blue and green cups relative to the robot, as that method does not include feature selection strategies.

C. Setting a Table

In this experiment, we considered a table arrangement action. We provided the robot with twelve demonstrations of how to place a grasped cup next to a plate and a fork on the table. The teacher always placed the cup on one side of the plate and the fork on the other, providing six training demonstrations for each setting (see Fig. 4).

We ran simulation experiments by starting the action from 500 different poses of the objects and with varying table heights. In all starting configurations, we placed the fork to the left of the plate. We recorded the success rate as the number of times the robot placed the cup on the right side of the plate (i.e., opposite to the fork). The results are shown in Fig. 7-right. As can be seen, we achieved a success rate of 96% when using all twelve training examples.

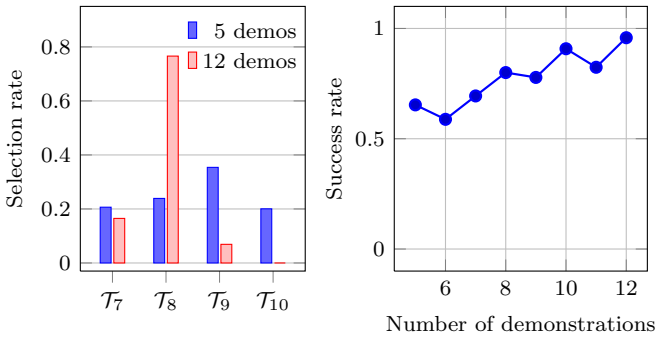


Fig. 7. Results for the action of placing a grasped cup next to a plate and fork. The left figure shows the selection of templates given five and twelve demonstrations of the action. The system prefers templates that describe the pose of the cup relative to the other objects on the table. The right figure shows the success rate as a function of the number of training demonstrations, reaching up to 96% given twelve demonstrations.

Additionally, we analyzed the influence of the number of training data on the success rate. As expected, the success rate increased with the quantity of the learning examples. We also analyzed the template selection over the varying number of training demonstrations, see Fig. 7-left. With five training demonstrations, four templates were selected approximately the same number of times during the 500 trials. Even with such a small number of demonstrations, our method succeeds in 65% of the times by finding for each initial starting pose a suitable explanation. The most selected templates included features describing the pose of the grasped cup relative to the fork and plate. With twelve training demonstrations, there is an increasing preference for explaining the action by template $\mathcal{T}_8$ which considers features of the pose of the gripper and the grasped cup relative to the fork. Using the recommended templates, our system achieved 96% success rate given twelve demonstrations.

We compared our method with Abdo *et al.* [1]. Also here, that method failed to reproduce the action if features such as the distance of the objects relative to the robot varied largely compared to the initial demonstrations. Instead, our approach successfully reproduced the action under different starting conditions. To illustrate this, we ran the experiment again after removing the fork from the table and by requesting the robot to reproduce the action. In this case, we consider a run a success when the robot places the cup on either side of the plate. Our method is able to adapt when the number of objects changes. In that case, it often selected a template that considers the pose of the cup relative to the plate. Under these settings, we achieved a success rate of 82% given twelve initial training demonstrations.

VI. Conclusions

In this paper, we presented an approach for learning manipulation actions from a small number of demonstrations by leveraging expert knowledge. Our method uses techniques inspired from recommender system theory to select which features are relevant for reproducing an action given the current state of the scene. By following these recommendations, our method builds multiple probabilistic models that are evaluated for each new initial condition. In this way, we are able to account for several models of the demonstrations and select the best one depending on the scene. We conducted extensive experiments in different tabletop scenarios. Our method achieves an high success rate and is able to select appropriate sets of features for reproducing each action.

References

[1] N. Abdo, H. Kretzschmar, and C. Stachniss. From low-level trajectory demonstrations to symbolic actions for planning. In *ICAPS Workshop on Combining Task and Motion Planning for Real-World App.*, 2012.

[2] T. Asfour, F. Gyarfas, P. Azad, and R. Dillmann. Imitation learning of dual-arm manipulation tasks in humanoid robots. In *Int. Conf. on Humanoid Robots*, 2006.

[3] C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *National Conf. on Artificial Intelligence*, 1998.

[4] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot programming by demonstration. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*. Springer, 2008.

[5] S. Calinon, F. Guenter, and A. Billard. On learning the statistical representation of a task and generalizing it to various contexts. In *Int. Conf. on Robotics & Automation*, 2006.

[6] P. Englert, A. Paraschos, J. Peters, and M. P. Deisenroth. Model-based imitation learning by probabilistic trajectory matching. In *Int. Conf. on Robotics & Automation*, 2013.

[7] C. Eppner, J. Sturm, M. Bennewitz, C. Stachniss, and W. Burgard. Imitation learning with generalized task descriptions. In *Int. Conf. on Robotics & Automation*, 2009.

[8] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Comm. of the ACM*, 1992.

[9] R. Jäkel, P. Meissner, S. R. Schmidt-Rohr, and R. Dillmann. Distributed generalization of learned planning models in robot programming by demonstration. In *Int. Conf. on Intelligent Robots and Systems*, 2011.

[10] J. Kober and J. Peters. Imitation and reinforcement learning - practical algorithms for motor primitive learning in robotics. (2):55–62, 2010.

[11] G. D. Konidaris and A. G. Barto. Efficient skill learning using abstraction selection. In *Int. Conf. on Artificial Intelligence*, 2009.

[12] G. D. Konidaris, S. R. Kuindersma, R. A. Grupen, and A. G. Barto. *Int. J. of Robotics Research*, 2012.

[13] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. 2008.

[14] D. Kulic, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura. Incremental learning of full body motion primitives and their sequencing through human motion observation. *Int. J. of Robotics Research*, 31(3), 2012.

[15] P. Matikainen, R. Sukthankar, and M. Hebert. Model recommendation for action recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.

[16] M. Mühlig, M. Gienger, S. Hellbach, J.J. Steil, and C. Goerik. Task-level imitation learning using variance-based movement optimization. In *Int. Conf. on Robotics & Automation*, 2009.

[17] M. J. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 1997.

[18] D. Song, K. Huebner, V. Kyrki, and D. Kragic. Learning task constraints for robot grasping using graphical models. In *Int. Conf. on Intelligent Robots and Systems*, 2010.

[19] H. Veeraraghavan and M. Veloso. Teaching sequential tasks with repetition through demonstration. In *Int. Conf. on Autonomous Agents and Multiagent Systems*, 2008.