The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Learning Task Constraints for Robot Grasping using Graphical Models

D. Song, K. Huebner, V. Kyrki and D. Kragic

Abstract— **This paper studies the learning of task constraints that allow grasp generation in a goal-directed manner. We show how an object representation and a grasp generated on it can be integrated with the task requirements. The scientific problems tackled are (i) identification and modeling of such task constraints, and (ii) integration between a semantically expressed goal of a task and quantitative constraint functions defined in the continuous object-action domains. We first define constraint functions given a set of object and action attributes, and then model the relationships between object, action, constraint features and the task using Bayesian networks. The probabilistic framework deals with uncertainty, combines a-priori knowledge with observed data, and allows inference on target attributes given only partial observations. We present a system designed to structure data generation and constraint learning processes that is applicable to new tasks, embodiments and sensory data. The application of the task constraint model is demonstrated in a goal-directed imitation experiment.**

I. INTRODUCTION

A major challenge in robotics is the integration of symbolic task goals and low-level continuous representations. In the research area of object grasping and manipulation, the problem becomes a formidable challenge. Objects have many physical attributes that may constrain planning of a grasp, as also robots have limited sensorimotor capabilities due to their various embodiments.

Considering the problem at hand, multiple approaches take their inspiration from imitation studies in developmental psychology: infants are able to infer the intention of others, and understand and reproduce the underlying task constraints through their own actions [1]. This goal-directed imitative ability is obtained along multiple stages in a developmental roadmap, both through the infant's own motor *exploration* (trial and error) and through the *observation* of others interacting with the world (imitation learning) [2]. Roboticists follow a similar developmental approach in order to design architectures for artificial agents [2], [3], [4], [5]. Most of these works, however, focus on the exploratory stage, where robots obtain object affordances through their empirical interaction with the world. The affordances being modeled are measured as the salient changes in the agent's sensory channels, which are interpreted as effects of specific actions applied on objects [4]. As an example, an effect of poking a ball is making it roll. Though it is an important step for a robot to discover this motor ability, another necessary step to achieve goal-directed behavior is to link this immediate motor act and its effects (as to poke the ball and let it roll), to the conceptual goal of an assigned task (as to provide the ball to a child). While trial-and-error-based exploration can be seen as inefficient to solve such goal learning problems, human supervision is helpful.

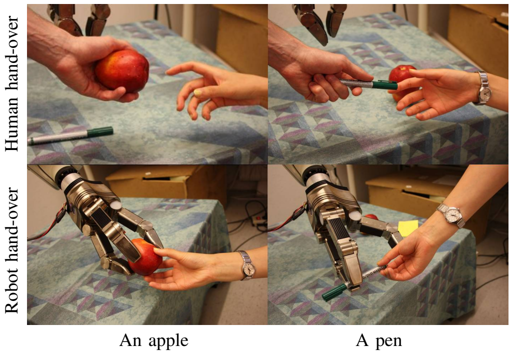


Fig. 1. The idea of goal-directed imitation and task constraint learning in a 'hand-over' task: though the embodiments, and thus hand configuration spaces, are very different, both hands follow similar task-based constraints.

This motivates an idea different from the classical developmental studies in such a way that it incorporates task-specific inputs from a human teacher. Thus, a system would be able to learn natural, goal-oriented types of grasps in a more efficient way. We clarify this idea in the hand-over task shown in Fig. 1. Such a task requires enough free area for another person to grasp the object. The robot should learn that an important constraint for this task is free area. There are numerous similar examples, e.g. pouring water from a cup requires the opening of a cup uncovered, and using a knife needs the robot to grasp the handle part. We believe these links can efficiently be learned by the input from a human expert. In this work, we develop such a method for learning of task goals and task relevant representations. The learning is performed in a high-dimensional feature space that takes into account different object representations and robot embodiments together with an input from a teacher.

II. RELATED WORK

Deriving quantified constraints from conceptual task goals presents a challenge similar to integrating high-level reasoning with low-level path planning and control systems in robotics. The main challenges originate from the representational differences in the two research fields. [6] addresses this problem through statistical relational models for a high-level symbolic reasoner, which is integrated into a robot controller. [7] proposes a coherent control, trajectory optimization, and

D. Song, K. Huebner and D. Kragic are with KTH – Royal Institute of Technology, Stockholm, Sweden, as members of the Computer Vision & Active Perception Lab., Centre for Autonomous Systems, www: http://www.csc.kth.se/cvap, e-mail addresses: {dsong,khubner,danik}@kth.se.

V. Kyrki is with Lappeenranta University of Technology, Finland, Department of Information Technology, www: http://www.it.lut.fi, e-mail address: Ville.Kyrki@lut.fi.

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

action planning architecture by applying the inference-based methods across all levels of representations.

Recently, imitation learning [8] and the concept of internal models [9] have received considerable attention in the field of robotics. The work in [2], [3] implements an internal model through Bayesian networks. In [10], it is shown that the internal models which represent the brain circuitry subserving sensorimotor control also participate in action recognition. They are used to predict the goal of observed behavior, and activate the correct actions to maintain or achieve the 'goal' state. A later work [11] extends the use of an internal model to the domain of visual-manual tasks.

A very recent work closely related to ours is the affordance model by Montesano *et al*. [4]. The authors adopt a self-supervised, developmental approach where the robot first explores its sensory motor capabilities, and then interacts with objects to learn their affordances. A Bayesian network is used to capture the statistical dependencies between actions, object features and the observed effects of the actions. They demonstrate an application of the affordance model for a robot to perform goal-directed imitation.

Concluding, we observe that most of the named references are either emphasizing higher-level planning systems [6], [7], or different domains with less complexity [2], [10], [11]. Even [4], though placed in the same domain of grasp affordance learning, is applied on fairly simple manipulation actions with discretely valued properties. Especially the latter presents a major drawback regarding the applicability in real world environments, which have to consider continuous and uncertain domains.

In our work, we directly approach the task-oriented grasping problem considering characteristics of a real robot system. We facilitate the generation of sensor and actor data using a selected grasp planning system [12] in a grasp simulation environment [13]. A simulator allows to capture embodiment-specific motor capabilities (by using different hand models), and also to include the wrench-space based grasp quality measures. It also enables supervised learning where knowledge of human experts can be efficiently used. A concept of providing expertise about task semantics through human tutoring has been implemented in [14]. Our grasps will be acknowledged by human experts to be suitable for given manipulation task(s), to let the system learn the underlying structure of the feature space in a probabilistic framework. To realize this, we take a widely used probabilistic graphical model, Bayesian Network [15]. This model will be used to encode the statistical dependencies between object attributes, grasp actions and a set of task constraints; therefore to link the symbolic tasks to quantified constraints. The statistical method exploits the co-occurrence of the stimuli in different sensory channels, much alike to similar mechanisms in the human brain [16].

The main contributions of our work are (i) introducing a semi-automated method for acquiring manually annotated, task-related grasps; (ii) learning probabilistic relationships between a multitude of task-, object- and action-related features with a Bayesian network; (iii) thus acquiring a hand-specific concept of affordance, which maps symbolic representations of task requirements to the continuous constraints; (iv) additionally, using a probabilistic framework, we can easily extend the object and action spaces, and allow flexible learning of novel tasks and adaptation in uncertain environments; (v) finally, our model can be applied to a goal-directed imitation framework, which allows a robot to learn from humans despite differences in their embodiments.

III. DEFINITION OF FEATURE SUBSETS

To introduce our approach, we first identify four subsets of features which play major roles in the consideration of a task-oriented grasp: task, object features, action features, and constraint features. These will define a frame for the creation of a Bayesion network learning approach presented in Section IV. Within these four subsets, we can flexibly instantiate a network with a specific constellation of network nodes, as will be demonstrated in Section V.

A. Task

In our notation, a *task* $T \in \mathcal{T} = \{T_1, ..., T_{n_T}\}$ refers to a 'basic task' that involves grasping or manipulation of a single object. According to [17], such a basic task can be called a *manipulation segment* which starts and ends with both hands free and the object at the stationary state. These manipulation segments are the building blocks for complex manipulation tasks. Though there may be an infinite number of complex tasks, we assume the basic building blocks form a finite set of object manipulation tasks. We therefore choose our task representation at the level of manipulation segments as each of them has an independent goal directly constraining how to grasp an object.

B. Object Features

An object feature set $\mathbf{O} = \{O_1, ..., O_{n_O}\}$ specifies the attributes (e.g. size) and/or categorical (e.g. type) information of an object. The features in $\mathbf{O}$ are not necessarily independent. The same attribute, such as shape, can be represented by different variables dependent on the capabilities of the perceptual system and the current object knowledge. For instance, eccentricity and convexity can be estimated from any kind of point cloud or mesh, while 3D shape representations like Zernike descriptors [18] can be used when a complete and dense 3D model of an object is available, i.e. when the object is known. Though apparently redundant, a system-dependent object representation offers flexibility in generalization across possibly different vision systems which can provide various levels of object knowledge.

C. Action Features

An action feature set $\mathbf{A} = \{A_1, ..., A_{n_A}\}$ describes the object-centered, *static* and *kinematic* grasp features, which may be the direct outputs of a grasp planner. $\mathbf{A}$ may include properties like grasp position, hand approach vector, or the grasp configuration.

D. Constraint Features

Finally, constraint feature set $\mathbf{C} = \{C_1, ..., C_{n_C}\}$ specifies a set of *constraint functions* which is defined by human experts; we term these to be a range of variables representing functions of both object and action features. As an example in a grasp scenario (like in Fig. 1), one may define the enclosure of the object volume as a constraint feature, which obviously depends on both object features (size and shape) and action features (grasp position and configuration). Thus, constraint features form the basic elements of general, task-dependent constraints in the sense that they can be used to quantitatively interpret the 'goal' or the 'requirements' of a given task.

IV. BAYESIAN NETWORK MODELING

Given a complementary set of variables $\{T, \mathbf{O}, \mathbf{A}, \mathbf{C}\} = \mathbf{X}$, our focus is to model the dependencies between their elements using a Bayesian network (BN) [15] (see an example network in Fig. 2). A BN encodes the relations between the set of random variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$. Each node in the network represents one variable, and the directed arcs represent conditional independence assumptions. The topology of the directed arcs is referred to as the structure of the network, and it is constrained to be directed and acyclic, meaning there are no cyclic connections between the nodes. Given the network structure S, and a set of local conditional probability distributions (CPDs) of each variable X_i, the joint distribution of all the variables can be decomposed as

$$P(\mathbf{x}) = P(\mathbf{x}|\boldsymbol{\theta}_s, S) = \prod_{i=1}^{n} P(x_i|\mathbf{pa}_i, \boldsymbol{\theta}_i, S)\ , \qquad (1)$$

where $\mathbf{pa}_i$ denotes the parents of node X_i, and the parameter vector $\boldsymbol{\theta}_s = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_n)$ specifies the CPDs. Training a BN includes discovering, from a dataset $\mathbf{D} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T$, (i) how one variable X_i depends on others $X_{j \neq i}$, i.e. the CPDs encoded by $\boldsymbol{\theta}_i$, which is referred to as the *parameter learning*, and/or (ii) if one variable X_i is conditionally dependent or independent of others $X_{j \neq i}$, which is referred to as the *structure learning*. In this paper, we will focus on the parameter learning, and leave the structure learning as a topic of future work.

Given that $\mathbf{X}$ includes both discrete (T) and continuous variables ($\mathbf{O}, \mathbf{A}, \mathbf{C}$), the model is a mixed Bayesian network. We model the discrete variable T as a multinomial distribution. For the continuous variables, we use Gaussian Mixture Models (GMM) to represent the relatively complex distributions. The number of components of the GMM for a variable X_i is determined from its training instances based on a Bayesian information criterion. In the BN, the node with the GMM distribution has a discrete latent parent to store the mixture coefficients. The learning of the BN parameter therefore uses an expectation-maximization algorithm.

The structure, as exemplified in Fig. 2, is determined based on our knowledge on the causal relationships between the four variable subsets. A *coarse structure* of the network reflects the relationships between the four subsets of variables. As task T is determinant variable that constrains the selection of objects and how to grasp the objects, it is the root node. T therefore parents all the features in $\mathbf{O}$ and $\mathbf{C}$. $\mathbf{O}$ features also parent $\mathbf{C}$ features because object attributes can, independently of task, influence the grasping method. $\mathbf{A}$ features only have $\mathbf{C}$ features as their parents since we assume the constraint variables encode the task requirements that determine the distribution of grasping actions. In addition to the coarse structure, there are also dependencies between some variables within each subset, which we refer to as the *fine structure*. For example in Fig. 2, the $\mathbf{A}$ variables are fully connected as they all influence each other to some extent.

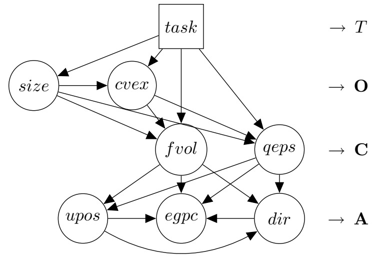


Fig. 2. Experimentally instantiated Bayesian network. The coarse structure of the BN specifies the subset dependencies between $T, \mathbf{O}, \mathbf{A}, \mathbf{C}$. The fine policy specifies the dependencies between variables within each feature subset. The latent variables for GMM nodes are not shown here.

V. EXPERIMENTAL SETUP

In this section, we will instantiate an exemplary set $\mathbf{X}$ for task-oriented grasp learning. The network illustrated in Fig. 2 will be used in our experiment to evaluate our approach to learn the task constraint for robot grasping. We first present the process of data generation using a simulation environment; then, we will explain in detail how to extract the selected features for the instantiated network; and finally we will describe how the BNs are trained and tested. The process described here can be visualized in the video material accompanying this paper.

A. Generating Data $\mathcal{D} = \{\mathbf{o}, \mathbf{a}, \mathbf{c}, \mathbf{t}\}$

We use GraspIt! 2.0 [13] as a simulation environment to provide the basis for data generation and visualization of our experiments. We decided on two hand models similar to those in Fig. 1 (a 20-DoF human hand model and a 7-DoF Schunk Dexterous hand model) for generating grasp hypotheses over 25 object models with 6 types taken from the Princeton Shape Benchmark (PSB) [19]: bottle, glass, mug, knife, hammer and screwdriver. To generate a set of grasp hypotheses on each object-hand pair, we use a planner for *Box Approximation, Decomposition and Grasping* (BADGr) [12]. BADGr can not only be used to extract action features $\mathbf{A}$, but also offers several shape representations to build an object feature set $\mathbf{O}$. Since $\mathbf{O}, \mathbf{A}$ are available, we extended the system to generate specified constraint features $\mathbf{C}$. We store, in an offline dataset $\{\mathbf{o}, \mathbf{a}, \mathbf{c}\}$ as instantiated feature vectors for only those grasps which result in valid force-closure grasps (GraspIt! provides stability criteria to identify those). To provide task labels for this set of stable grasps, each object-grasp combination (a grasp instance $\{\mathbf{o}_i, \mathbf{a}_i, \mathbf{c}_i\}$)

TABLE I

FEATURE SETS USED IN OUR EXPERIMENTS.

Set	Symbol	Description	Type
	T *task*	$\mathcal{T} = \{$*hand-over*, *pouring*, *tool-use*$\}$	$\mathbb{D}^3$
$\mathbf{O}$	O_1 *size*	Object (bounding box) size.	$\mathbb{R}^3$
	O_2 *cvex*	Object convexity.	$\mathbb{R}^1$
$\mathbf{A}$	A_1 *egpc*	Eigengrasp pre-configuration.	$\mathbb{R}^2$
	A_2 *upos*	Unified position.	$\mathbb{R}^3$
	A_3 *dir*	Quaternion orientation.	$\mathbb{R}^4$
$\mathbf{C}$	C_1 *fvol*	Free volume.	$\mathbb{R}^1$
	C_2 *qeps*	Grasp Stability.	$\mathbb{R}^1$

is visualized as a 3D scene in GraspIt!. The human tutor then selects, among a set of possible tasks $\mathcal{T} = \{T_1, \ldots, T_{n_T}\}$ the affordable task(s) for this object-grasp configuration. If this grasp instance is labeled to be valid for at least one task $t_i = T_k \in \mathcal{T}$, a data instance $\{\mathbf{o}_i, \mathbf{a}_i, \mathbf{c}_i, t_i\}$ will be included in the final complete dataset $\mathcal{D} = \{\mathbf{o}, \mathbf{a}, \mathbf{c}, \mathbf{t}\}$ for training and testing. Note that, for this reason, our training is based on positive examples, not considering negative (i.e. non-force-closure, or non-labeled) examples.

B. Network Instantiation

The dataset $\mathcal{D} = \{\mathbf{o}, \mathbf{a}, \mathbf{c}, \mathbf{t}\}$ includes the instances for a large set of feature variables. In this section, we select an experimental small set of features $\mathbf{X} = \{\mathbf{O}, \mathbf{A}, \mathbf{C}, T\}$ to evaluate the capabilities of our learning framework (see the eight variables in Fig. 2 and Tab. I). The following text presents the technical details of the feature extraction.

Task (T): For the current study, we use a single discrete node to represent the task variable with three states $\mathcal{T} = \{$*hand-over*, *pouring*, *tool-use*$\}$. This decision connects to our choice of the six object types as they are a set of hand-over-able, pour-able, or tool-use-able objects.

Object Size (O_1): A first step of BADGr is to envelop the object's point cloud by a minimum volume bounding box, the so-called 'root' box B_0. The size of the object is taken to be the three dimensions of this box, thus corresponding to width, depth and height of the object. We note that all objects have been adjusted in such a way that their 'top' points in positive z-axis direction.

Object Convexity (O_2): A subsequent step of BADGr is to decompose the root box B_0 and re-approximate until a fitness measure is reached (details in [12]). After this process, a number n of bounding boxes $B_{0<i<n}$ has emerged which envelop parts of the object. The object's convexity is approximated to be the ratio of volumes before and after the decomposition, as $cvex = volume(B_0)/\sum_{i=1}^{n} volume(B_i)$.

Eigengrasp Pre-Configuration (A_1): BADGr only plans the pose of each hand model (grasp position and hand orientation), and does not plan grasp pre-configuration, or pre-shape. Since the hand configuration spaces are high-dimensional, we use the idea of [20] to project those into 2D Eigengrasp spaces. The two dimensions roughly depict spread and extension of each hand. The mapping for the human hand comes with GraspIt! which is based on [20]; for the Schunk hand, the first dimension is mapped to the spread joint, and the second dimension to the extension of the three finger joints. For each grasp pose that BADGr generates, we sample 5 random Eigengrasps.

Unified Grasp Position (A_2): The grasp position with respect to the object will be an important feature of our experiment. For example, a mug should not be grasped from above in the case of a pouring task. We represent this parameter using a *unified grasp position*, which is calculated by projecting the 3D grasp position to a 2D unified spherical space. This sphere is defined by the center point of the specific object (taken from the root box) and a fixed radius (which only has to ensure that all objects are inside this radius). We then intersect this sphere with the grasp approach vector emitted from the grasp position.

Grasp Orientation (A_3): Though the unified grasp position is using the grasp direction, it is not encoding it. Thus, we consider this value in a separate variable. The grasp orientation is embedded in each grasp generated by BADGr, in terms of a quaternion representation.

Free Volume (C_1): The free-volume constraint defines the percentage of the object volume that corresponds to the non-covered part in a grasp configuration. Briefly, we span a tetrahedron Δ using the palm position and the three contact points that maximize the volume of Δ. Considering the box decomposition $B_{0<i<n}$, we compute to what extent Δ intersects each B_i, and sum up the volumes to V_{occ} (occupied volume). The free volume is then acquired as $fvol = 1 - V_{occ}/\sum_{i=1}^{n} volume(B_i)$.

Grasp Stability (C_2): To incorporate force-related task constraints, we use one of the commonly used measures of grasp stability that GraspIt! provides, eps_{L1}. It describes stability of each grasp in terms of force-closure (see [21]).

C. Training and Testing the Bayesian Network

To train and use the BNs (Fig 2) for human and Schunk hands, we use the BNT [22], the Bayes Net Toolbox for Matlab. In the following experiments, the training data comes from the on-line labeling by only one human expert. For the human-hand BN, the training set includes 600×3 instances, with 600 instances per task, and around 100 instances per object type; and for the Schunk-hand BN, the training set includes 1200×3 instances, with 1200 instances per task, and around 200 instances per object type. The testing set comes from the 6 objects that are not included in the training set. Each of the 6 objects belongs to one of the 6 object types. This is to evaluate how well the trained network can generalize to the unknown objects.

VI. EXPERIMENTAL RESULTS

In this section, we will describe the application of the trained BN for three different experiments. While two of them will mainly provide a view on the evaluation of the technique, the third one will show a setup for robot imitation based on task-constraints. For each experiment, we formulate the corresponding semantic questions to the system.

TABLE II

EXPERIMENT IV.A: DISTRIBUTION OF UNIFIED POSITION CONDITIONED ON TASKS AND OBJECT FEATURES $P(upos|task, size, conv)$.

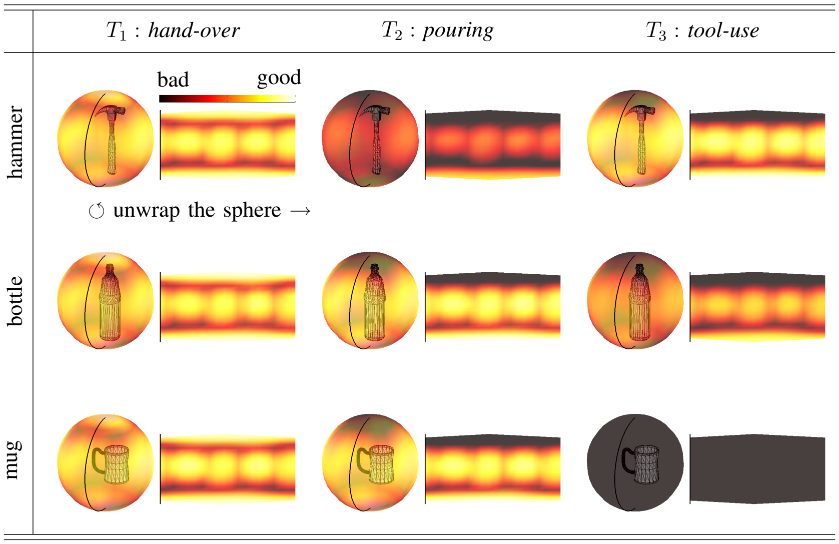


TABLE III

EXPERIMENT IV.B: CONFUSION MATRIX.

$T \mid \mathbf{O}$

0.51	0.15	0.34
0.22	0.78	0.00
0.15	0.10	0.75

$T \mid \mathbf{O}, \mathbf{A}$

0.56	0.35	0.09
0.13	0.87	0.00
0.12	0.00	0.88

$T \mid \mathbf{O}, \mathbf{A}, \mathbf{C}$

0.70	0.21	0.09
0.11	0.89	0.00
0.11	0.00	0.89

A. *"From where to grasp an object, given a task?"*

Formulating this question as $P(upos|task,\ size,\ conv)$, our goal is to observe how our three tasks influence the *position* of a grasp, *upos*. As representatives for the experimental results, we select a hammer, a bottle, and a mug out of the 25 object models as the test set, and train the Bayesian network using the Schunk hand data stored from the remaining 22 models. We then compute P for all 3 test objects, and all 3 tasks. The results for this experiment are shown in Tab. II.

Analyzing the results, we have the following observations: (i) the BN is clearly affected by the BADGr planner, providing a lot of "from where to grasp" hypotheses from the four sides, top and bottom of an object. (ii) Given a *hand-over* task, the results do not substiantially differ, and all major directions are valid. (iii) Given a *pouring* task, the network clearly rejects to grasp from the top in cases of bottle and mug. We see that also the hammer has some (but much less) likelihood to be poured from these directions. This is because objects are only represented by size and convexity, and the hammer has similar size as a bottle, but higher convexity. (iv) For *tool-use*, the network emphasizes the hammer, from sides and bottom, to be tool-use-able. It correctly rejects grasps from the top. In a same way, and for the same reasons as in the "pourable hammer" case, the bottle is tool-usable. The mug is identified as being non-tool-use-able at all, since it is very much different in size and convexity from a usual tool (which should be long and convex up to some extent).

B. *"What tasks is this (object / object and action / object, action and constraint) good for?"*

Dependent on the characteristic of *this*, the question can be formulated as $P(T|\mathbf{O})$, $P(T|\mathbf{O},\mathbf{A})$, or $P(T|\mathbf{O},\mathbf{A},\mathbf{C})$. Since the task is represented by a single discrete node, we can identify each problem as a classification, given different amounts of observations. Our goal is to analyze how good these classifications work for unknown objects. We train the networks for the human and the Schunk hand, leaving out 1 object per object type. Thus, our training set includes data from 19 objects with the test data covering all 6 object types. The resulting confusion matrices are shown in Tab. III.

Analyzing the results, we can make the following observations: (i) object features contain important information for task, in particular *pouring* (78% classification rate) and *tool-use* (75%); (ii) introduction of action features improves correct classification of these tasks (87% and 88%), but does not affect *hand-over* (56%); (iii) when introducing constraint features, *pouring* and *tool-use* do not improve significantly (89% and 89%), but *hand-over* (70%).

C. *"Can you imitate this demonstrated grasping task?"*

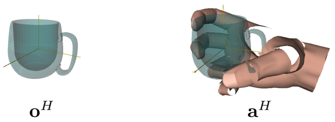
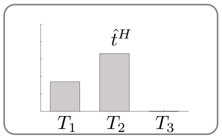
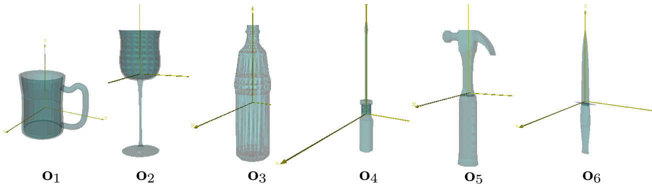
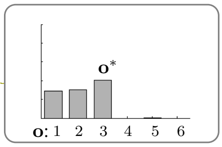
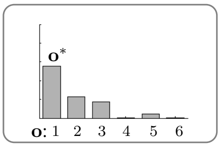
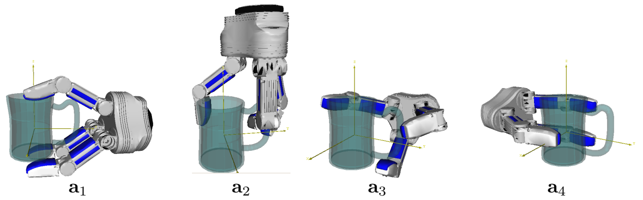
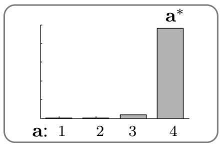
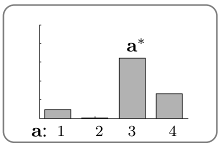
In this section, we demonstrate the use of the task constraint Bayesian network in a goal-directed imitation experiment. The experiment is implemented using the human hand model as the demonstrator, and the Schunk hand as the imitator. The goal is to imitate the demonstrator performing a *pouring* task using a mug $\mathbf{o}^H$. We therefore train the networks for both hands, letting out the seven test objects $\mathbf{o}^H$ and $\mathbf{o}_1$ to $\mathbf{o}_6$ presented in Tab. IV. Superscript ^{H} is used to indicate features from human demonstration. We first describe the general formulation in a two-step imitation framework.

In the first step, the robot observes a human performing a grasp on an object, and estimates the intention (task) t^H of the human action. $P^H(T|\mathbf{O},\mathbf{A},\mathbf{C})$ encodes the probability of the tasks for the demonstrated object-grasp combination, where P^H means that the BN is specific to the demonstrator's embodiment. We denote the maximum-likelihood estimate of the task as $\hat{t}^H$.

In the second step, the robot finds the most compatible grasp on the object(s) it perceived, in order to achieve the same task $\hat{t}^H$. This step can be formulated as a Bayesian decision problem, where a reward function r defines the degree

TABLE IV

Experiment IV.C: Goal-directed imitation on *'pouring'* task.

Scenes	Objective Functions	
Step 1 Human demonstration: recognize task $\hat{t}^H$  $T_1 =$ *hand-over* $T_2 =$ *pouring* $T_3 =$ *tool-use*	$P^H(\, t \mid \mathbf{o}^H, \mathbf{a}^H, \mathbf{c}^H)$ 	
Step 2.1 Select object $\mathbf{o}^*$: matching $\hat{t}^H$, or also similar to $\mathbf{o}^H$  $\mathbf{o}_1$ $\mathbf{o}_2$ $\mathbf{o}_3$ $\mathbf{o}_4$ $\mathbf{o}_5$ $\mathbf{o}_6$	$P^R(\, T_2 \mid \mathbf{o})$ 	$P^R(\, T_2 \mid \mathbf{o}) \cdot 0.2$ $+ S(\mathbf{o},\, \mathbf{o}^H \vert T_2) \cdot 0.8$ 
Step 2.2 Select action $\mathbf{a}^*$: matching $\hat{t}^H$, or also similar to $\mathbf{a}^H$  $\mathbf{a}_1$ $\mathbf{a}_2$ $\mathbf{a}_3$ $\mathbf{a}_4$	$P^R(\, T_2 \mid \mathbf{o}_1, \mathbf{a})$ 	$P^R(\, T_2 \mid \mathbf{o}_1, \mathbf{a}) \cdot 0.2$ $+ S(\mathbf{a},\, \mathbf{a}^H \vert T_2) \cdot 0.8$ 

of similarity in the set of features $\mathbf{x} = \{\mathbf{o}, \mathbf{a}, \mathbf{c}, t\}$ between the demonstrator and the robot. As the knowledge over this feature set is not certain, the expectation $E()$ is taken over the reward function. For instance, $P^R(T|\mathbf{O}, \mathbf{A}, \mathbf{C})$ encodes the probability of the suitable task given an object-grasp combination, where P^R means that the BN is specific to the robot's embodiment. The general optimization function for decision making is

$$\langle \mathbf{a}^*, \mathbf{o}^* \rangle = \operatorname*{argmax}_{\mathbf{a}\in\mathcal{A},\mathbf{o}\in\mathcal{O}} E\Big(r(\mathbf{a}^H, \mathbf{o}^H, \mathbf{c}^H, \hat{t}^H, \mathbf{a}, \mathbf{o}, \mathbf{c}, t)\Big) \;, \qquad (2)$$

where the maximization is over a set of stable grasp hypotheses $\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_{n_a}\}$ generated by the robot's grasp planner, and/or available objects $\mathcal{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_{n_o}\}$ presented to the robot. We present two imitation scenarios to illustrate the fomulation:

1) Matching of Tasks: The objective is to plan a grasp to match the same task while the robot is given six objects (see Tab. IV). In step 1, the robot estimates the most likely task of the demonstrated grasp to be *pouring* $\hat{t}^H = T_2$. In the second imitation step, the robot first follows step 2.1 to select the object $\mathbf{o}^* \in \mathcal{O}$ that best affords T_2, and then step 2.2 to select the grasp action $\mathbf{a}^* \in \mathcal{A}$ that best affords T_2. The results of the second step are illustrated in the two left bar plots in Tab. IV. The reward is a simple indicator function of the demonstrated task, $r(T_2)$, which equals 1 if $t = T_2$, 0 otherwise. The optimization is then

$$\langle \mathbf{a}^*, \mathbf{o}^* \rangle = \operatorname*{argmax}_{\mathbf{a}\in\mathcal{A},\mathbf{o}\in\mathcal{O}} rP^R(t = T_2|\mathbf{a}, \mathbf{o}, \mathbf{c}) \;, \qquad (3)$$

where $P^R(t = T_2|\mathbf{a}, \mathbf{o}, \mathbf{c})$ is the likelihood of a task being T_2 given the features of an object and the grasp generated on it, i.e. the 'task-affordances' of the object and the grasp.

As shown in step 2.1 in Tab. IV, the network clearly rejects the three tools, since their features $\{size, cvex\}$ can well discriminate between *pouring* and *tool-use* tasks. Among the three 'container' objects, the network produces similar likelihood levels, with a slight favor towards the bottle $\mathbf{o}_3$. This may be explained by the simpler shape of the bottle which enables more stable grasps to be generated that afford *pouring*. We assume the robot selects the mug $\mathbf{o}_1$ and show the grasp selection on $\mathbf{o}_1$ in step 2.2. The network selects the grasp hypothesis $\mathbf{a}_4$ as it best affords the *pouring* task. As expected, the network rejects the top grasp $\mathbf{a}_2$ because it blocks the opening of the mug; and rejects $\mathbf{a}_1$ as it has one finger poking into the mug thus does not afford *pouring*.

2) Matching of Tasks and Features of Object and Action: In this scenario, the objective is not only to choose the object and action that afford the task, but also to select those that are similar to the object used by the human and the grasp the human applied, i.e. matching their features. This requires adding to the objective function a similarity measure between $\mathbf{o}$ and $\mathbf{o}^H$, and between $\mathbf{a}$ and $\mathbf{a}^H$. The results of the second step of imitation are illustrated in the two right bar plots in Tab. IV. Note that the feature vectors $\mathbf{o}$ and $\mathbf{a}$ are both concatenations of multiple variables, such as *egpc* and *upos* for $\mathbf{a}$. If we let o_i and a_i represent the individual variables, the similarity between, for example $\mathbf{a}$ and $\mathbf{a}^H$, is defined as

$$S(\mathbf{a}, \mathbf{a}^H|t) = \sum_{i=1}^{n_A} \frac{P^R(a_i^H|t)}{(a_i - a_i^H)^T \Sigma_{a_i}^{-1} (a_i - a_i^H)} \;, \qquad (4)$$

where n_A is number of the action variables, and Σ_{a_i} is the covariance matrix of a_i. This is a linear combination of the similarity in each a_i (calculated as the inverse of the

Mahalanobis distance between a_i and a_i^H), which is weighed by the likelihood of a_i^H in the robot's BN for a given task t. This is because that given a task, the i^{th} reference action feature a_i^H may have lower likelihood in P^R than others, i.e. it is less confidently represented in the robot's knowledge base. For this feature, the similarity is weighed down. The optimization is then formulated as

$$\begin{aligned}\langle \mathbf{a}^*, \mathbf{o}^* \rangle = \underset{\mathbf{a}\in\mathcal{A},\mathbf{o}\in\mathcal{O}}{\operatorname{argmax}} \quad & r\,\{P^R(t=T_2|\mathbf{a},\mathbf{o},\mathbf{c})\cdot(1-\lambda) \\ & + S(\mathbf{o},\mathbf{a},\mathbf{o}^H,\mathbf{a}^H|t=T_2)\cdot\lambda)\},\end{aligned} \quad (5)$$

where λ is used to specify the relative importance of matching the features as compared to matching the task.

Tab. IV shows the results on object and action selections when $\lambda = 0.8$, i.e. the main goal (80%) is to find $\mathbf{o}^*$ and $\mathbf{a}^*$ that are similar to human demonstration, with only 20% requirements of matching the *pouring* task. In step 2.1, we see that the network selects the mug $\mathbf{o}_1$ because it is most similar to the mug used in the demonstration. In step 2.2 the selected grasp is $\mathbf{a}_3$. This is because the action feature $upos$ of $\mathbf{a}_3$ is most similar to the demonstration: the grasping positions of two hands are both on the negative side of the $x-$axis in the object's local coordinate frame.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we proposed a probabilistic framework for learning of task constraints in grasp selection. Our constraint learning model links the semantic requirements of manipulation tasks to the continuous feature space of the objects and grasp actions. Our approach is semi-automated and embodiment-specific. A simulation-based grasp planner generates a set of hand-specific, stable grasp hypotheses on a range of objects. A teacher provides the knowledge of task requirements by labeling each hypothesis with the suitable manipulation task(s). The underlying relations between the conceptual task goals and the continuous object-action features are encoded by the probabilistic dependencies in a Bayesian network. Using this network as a knowledge base, the simulation experiments showed that the robot is able to infer the intended task of a human demonstration, choose the object that affords this task, and select the best grasp action to fulfill the task requirements. Though we implement and test the current framework based on the BADGr grasp planner [12], this task constraint framework can be integrated with any grasp planning system.

In the current implementation, we do not address the learning of the network structure, but manually connect the nodes based on our knowledge. In the future, we intend to introduce more tasks, constraint functions, as well as different and potentially redundant object and action features. In such cases, building the network structure based purely on human knowledge will be cumbersome and nonreliable. Data-driven, automated structure learning is needed to identify the task-relevant variables, and discover the underlying dependencies between these variables. In addition, we would like to implement the approach on the real robot sensory-motor platforms. For example, we can introduce different object representations by applying different vision modules, allowing the network to encode uncertainty in the perception system.

ACKNOWLEDGMENTS

This work was supported by EU IST-FP7-IP GRASP, EU IST-FP6-IP-027657 PACO-PLUS, and Swedish Foundation for Strategic Research.

REFERENCES

[1] A. N. Meltzoff, *Elements of a Developmental Theory of Imitation*. Cambridge, MA, USA: Cambridge University Press, 2002, pp. 19–41.

[2] R. Rao, A. Shon, and A. Meltzoff, "A Bayesian Model of Imitation in Infants and Robots," in *Imitation and Social Learning in Robots, Humans, and Animals*, 2004, pp. 217–247.

[3] D. B. Grimes and R. P. N. Rao, "Learning Actions through Imitation and Exploration: Towards Humanoid Robots that Learn from Humans," in *Creating Brain-Like Intelligence*, ser. Lecture Notes in Computer Science, vol. 5436. Springer, 2009, pp. 103–138.

[4] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning Object Affordances: From Sensory–Motor Coordination to Imitation," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 15–26, 2008.

[5] C. Acosta-Calderon and H. Hu, "Robot Imitation: Body Schema and Body Percept," *Applied Bionics and Biomechanics*, vol. 2, no. 3-4, pp. 131–148, 2005.

[6] D. Jain, L. Mösenlechner, and M. Beetz, "Equipping Robot Control Programs with First-order Probabilistic Reasoning Capabilities," in *IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 3130–3135.

[7] M. Toussaint, N. Plath, T. Lang, and N. Jetchev, "Integrated Motor Control, Planning, Grasping and High-level Reasoning in a Blocks World using Probabilistic Inference," in *IEEE International Conference on Robotics and Automation*, 2010, to appear.

[8] C. L. Nehaniv and K. Dautenhahn, Eds., *Imitation and Social Learning in Robots, Humans, and Animals: Behavioural, Social and Communicative Dimensions*. Cambridge University Press, 2004.

[9] D. Wolpert and M. Kawato, "Multiple Paired Forward and Inverse Models for Motor Control," *Neural Networks*, vol. 11, no. 7-8, pp. 1317–1329, October 1998.

[10] Y. Demiris and M. Johnson, "Distributed, Predictive Perception of Actions: A Biologically Inspired Robotics Architecture for Imitation and Learning," *Connection Science*, vol. 15, no. 4, pp. 231–243, 2003.

[11] E. Oztop, D. Wolpert, and M. Kawato, "Mental State Inference using Visual Control Parameters," *Cognitive Brain Research*, vol. 22, no. 2, pp. 129–151, February 2005.

[12] K. Huebner, S. Ruthotto, and D. Kragic, "Minimum Volume Bounding Box Decomposition for Shape Approximation in Robot Grasping," in *IEEE Int. Conf. on Robotics and Automation*, 2008, pp. 1628–1633.

[13] A. Miller and P. Allen, "Graspit! A Versatile Simulator for Robotic Grasping," *Robotics and Automation*, vol. 11 (4), pp. 110–122, 2004.

[14] Z. Xue, A. Kasper, M. J. Zoellner, and R. Dillmann, "An Automatic Grasp Planning System for Service Robots," in *14th International Conference on Advanced Robotics*, 2009.

[15] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, September 1988.

[16] A. Rao, B. A. Olshausen, and M. Lewicki, Eds., *Probabilistic Models of the Brain: Perception and Neural Function*. MA: MIT Press, 2002.

[17] R. Zöllner, M. Pardowitz, S. Knoop, and R. Dillmann, "Towards Cognitive Robots: Building Hierarchical Task Representations of Manipulations from Human Demonstration," in *IEEE International Conference on Robotics and Automation*, 2005, pp. 1535–1540.

[18] M. Novotni and R. Klein, "Shape Retrieval using 3D Zernike Descriptors," *Computer-Aided Design*, vol. 36 (11), pp. 1047–1062, 2004.

[19] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton Shape Benchmark," in *International Conference on Shape Modeling and Applications*, 2004, pp. 167–178.

[20] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dexterous Grasping via Eigengrasps: A Low-dimensional Approach to a High-complexity Problem," in *RSS 2007 Manipulation Workshop*, 2007.

[21] C. Ferrari and J. Canny, "Planning Optimal Grasps," in *IEEE Int. Conference on Robotics and Automation*, vol. 3, 1992, pp. 2290–2295.

[22] K. Murphy, "BNT – Bayes Net Toolbox for Matlab," **[URL]** http://code.google.com/p/bnt/, 1997. Last visited July 15, 2010.