

Homogeneity Analysis for Object-Action Relation Reasoning in Kitchen Scenarios *

Hanchen Xiong Sandor Szedmak Justus Piater
Institute of Computer Science, University of Innsbruck
Technikerstr.21a A-6020, Innsbruck, Austria
{hanchen.xiong,sandor.szedmak,justus.piater}@uibk.ac.at

ABSTRACT

Modeling and learning object-action relations has been an active topic of robotic study since it can enable an agent to discover manipulation knowledge from empirical data, based on which, for instance, the effects of different actions on an unseen object can be inferred in a data-driven way. This paper introduces a novel object-action relational model, in which objects are represented in a multi-layer, action-oriented space, and actions are represented in an object-oriented space. Model learning is based on homogeneity analysis, with extra dependency learning and decomposition of unique object scores into different action layers. The model is evaluated on a dataset of objects and actions in a kitchen scenario, and the experimental results illustrate that the proposed model yields semantically reasonable interpretation of object-action relations. The learned object-action relation model is also tested in various practical tasks (e.g. action effect prediction, object selection), and it displays high accuracy and robustness to noise and missing data.

Categories and Subject Descriptors

I.2.9 [Robotics]: Manipulation; I.2.6 [Learning]: Knowledge acquisition—*Object-action relation learning*

General Terms

Algorithms, Experimentation

Keywords

Homogeneity analysis, Object-action relation learning

1. INTRODUCTION

*The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 270273, Xperience.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
MLIS'13, August 04 2013, Beijing, China
Copyright 2013 ACM 978-1-4503-2019-1/13/08 ...\$15.00.

Manipulations of objects are core and indispensable functions in robotic systems to fulfill various practical tasks. However, because of the diversity of real-world objects in shape, material and other properties, manipulation design at the instance level is very effort-consuming and thus prohibitive. Learning principles or correlation patterns of different actions based on trial experiences is an appealing direction of robotics research. In other words, an agent can acquire knowledge of object-action relations in a data-driven manner by making use of a limited number of experiments. In addition, the study of object-action relations has also attracted attention within the cognition and psychology communities [5, 10], since it is expected to be related to how human beings accumulate knowledge by physically interacting with different objects. Humans begin to interact with their environment in their infancy, and in many interactions, two elements are involved: objects and actions. Actions are executed on objects with the humans' motor capabilities, and the effects of these actions are observed with their perception abilities. Based on such repeated interactions, human beings can quickly acquire object-action knowledge, and easily fulfill different actions on various objects by transferring such knowledge to novel objects. Although the exact mechanism of how the human brain organizes and learns object-action relations is still unknown, it has been pointed out that computational modeling of object-action relations can be a plausible perspective for the study of both robotics and human cognition.

Nevertheless, modeling and learning object-action relations has been a difficult task. The difficulties mainly stem from two sources. First, the structure of descriptions of both objects and actions can be very complex. The descriptions are derived from several sources, and the corresponding feature spaces are high-dimensional (i.e., objects and actions are characterized by large numbers of parameters). The second difficulty is due to the small number of experiments which can confirm the effects of different actions on objects. Even worse, in some cases, the experiments might provide contradicting outcomes. In consequence, the empirical data are rather sparse and noisy.

In this paper we put forward a novel model of object-action relations, in which objects are represented in a multi-layer action-oriented space, and actions are represented in an object-oriented space. The object-action relations are encoded in these two spaces, on which various reasoning tasks can be performed. The training data for the model are constructed from two sources, objects and the collection of effects (positive and negative) of different actions

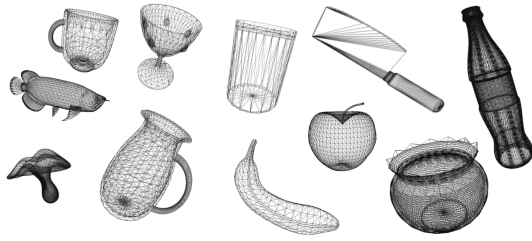


Figure 1: A sample set of kitchen objects

executed on objects. Two pieces of information are summarized in a structure called object-action profiles. Objects are represented by categorical indicators of basic properties and binary labels of various low- and high-level geometric features. Actions are represented by binary labels of object-dependent effects. The model is learned with *homogeneity analysis*. The strength of homogeneity analysis is that it can map multi-variate categorical/binary data to a homogeneous Euclidean space. Via these projections, objects and actions can be effectively represented with object scores and category quantifications. Basically, the object scores are computed as the average of quantifications of categories which they belong to, and category quantifications are computed as the geometric centroid of objects they belong to. These two projections are iteratively updated until convergence. Based on homogeneity analysis, action-category quantifications are represented in an object-oriented manner. The resulting object scores, however, do not fit our modeling scenario. The object scores are computed by treating all variables of object-action profiles equivalently, and therefore the scores are unique for all actions. By contrast, our model is designed to represent objects differently with respect to different actions. Therefore, we provide dedicated means to determine the dependencies between category quantifications of object and action variables, and decompose the object scores/representations into different action layers.

We present our model and associated learning/reasoning procedures in the context of object-action relations within a kitchen scenario (Figure 1). A database of typical kitchen objects and actions is constructed as well to evaluate our model. The experimental results demonstrate that the model yields semantically good interpretation of object-action relations by displaying reasonable dependencies and correlations between object and actions variables. In addition, the experiment with sparseness and noise added into training data highlights the robustness of our model to noisy and missing data.

1.1 Related Work

For object manipulation knowledge modelling, the concept of *affordance* [5] has been widely used [9, 8, 10, 3, 4] to link objects and actions in terms of object-action-effect triples. An affordance defines how an object “affords” a manipulation by an agent based on its motor abilities, and how this manipulability can be perceived by the agent [5]. For instance, the grasping affordance of a stone is much higher for a human being than a dog since human hands have better motor control of fingers than dogs’ paws. More concretely, object affordances represent how can an agent interact with real-world environment by encoding the relations among actions, objects and effects. Although there have been numer-

ous studies on how affordances can be modelled such that they can be effectively learned and utilized to assist practical robotic manipulation, the object/action affordance problem, at its base, is about how an agent can understand objects based on interactions with them by using its motor and perceptual capabilities. However, most previous studies are limited to one isolated object affordance (e.g. grasping). In some cases, multiple objects are involved and interact with each other within one manipulation. For example, a single action such as cutting involves two objects, the cutting tool (e.g. knives) and the object being cut (e.g. an apple). In [7], the affordance definition was extended to object relations. However, since only geometric relation (distance, angle of orientations) between multiple objects are used in [7], it still cannot model concepts such as cutting affordances for objects. Our work, by contrast, seeks to model general object-action relations. Our relational model connects objects and all possible actions that can be performed on them.

Our model is mainly inspired by [3], of which the basic assumption is that objects that share similar parts (e.g. rim, handles) should also hold similar grasping affordances. We extend [3] in two ways: first, we consider general object-action relations instead of only grasping affordances; second, the dependency of actions on different parts can be learned, in which way, for different actions, different co-occurring parts among objects will be considered for their action-effect reasoning.


Other related work involves modeling of *sensorimotor coordination* [9], where a Bayesian network is employed to model multiple affordances associated with objects based on visual properties (e.g. color, size, concavity) and basic motor actions (grasping, touching, tapping). The dependences between actions, perception and effects are encoded in the directed edges within the Bayesian network. One shortcoming of this model is the dependency learning (i.e. the Bayesian network structure). Since in a Bayesian framework it is impractical to estimate the likelihoods of all possible dependency structures, Markov chain Monte Carlo (MCMC) sampling was used to approximate them. However, one practical problem with MCMC is that it can be quite inefficient (usually multiple chains are necessary); secondly, the approximation can be misleading when the training data is small in size, noisy and incomplete. By contrast, the dependency learning of our model is based on the category quantifications from homogeneity analysis, which is robust to noisy and missing data.


2. MODELING

In this section, two basic elements are explained for object-action relation modeling. First, we introduce a new data structure constructed from empirical object and action data (section 2.1). Secondly, section 2.2 presents an overview of the model structure (Figure 3), in which objects are represented in a multi-layered action-oriented space, and actions are likewise represented in an object-oriented space.

2.1 Data Structure

Since our objective is to learn the relations between objects and actions, the training data is constructed from two sources. One is the *object dataset*, in which basic properties (e.g. size, functionality, material) are labelled, and various low- and high-level geometric properties can be extracted by visual perception. The other source is the *action dataset*

	Size: Bigger than gripper range -
Low-Level Geometry Information: 3D features: e.g. edges, curvatures 2D features: e.g. contours, sketches	
High-Level Geometry Information: 3D part: e.g. rim + handle -	
Functionality: Container	
Material: Ceramic	
Action Log: Grasp by closing fingers + Roll + Cut - Chop - Grasp by expanding fingers +	

	Size: Smaller than gripper range +
Low-Level Geometry Information: 3D features: e.g. edges, curvatures 2D features: e.g. contours, sketches	
High-Level Geometry Information: 3D part: e.g. rim - handle -	
Functionality: Food	
Material: Plant	
Action Log: Grasp by closing fingers + Roll + Cut + Chop + Grasp by expanding fingers -	

(a)

O	Mesh	<Gripper	L_Geo		H_Geo		Func	Mate	Action log				
			3D	2D	rim	handle			Grasp_C	Roll	Cut	Chop	Grasp_E
1	file1	1			1	-1	1	1	1	-1	*	1	-1
2	file2	-1			-1	*	2	2	-1	*	1	1	*
3	file3	-1			1	1	2	5	*	1	*	1	1
4	file4	1			1	1	5	3	1	1	-1	*	1
5	file5	1			-1	-1	1	4	*	1		1	-1
6	file6	-1			1	1	4	6	1	-1	*	-1	1

Functionality	Container	Food	Cooker	Cutting tool	Eating tool
	1	2	3	4	5

Material	Plastic	Glass	Ceramic	Plant	Animal	Metal
	1	2	3	4	5	6

(b)

Figure 2: (a) Two examples of object-action profiles. (b) Collection of object-action profiles, * denoting missing data. Incompleteness (or sparseness) will always be a problem in training data.

that collects the effect of different actions applied on objects. In our study, the information from these two sources is merged into *object-action profiles*. Figure 2(a) presents two examples of object-action profiles. In the upper part, object shape is displayed. Some basic properties are labelled and geometric features are extracted. Because we are only concerned with the kitchen scenario, functionalities are limited to {container, food, cooker, cutting tool, eating tool}, and materials are limited to {plastic, glass, wood, plant, animal, metal}. For size, a binary indicator is used to check if it is smaller than the gripper’s maximum range. In addition, low-level and high-level geometric features of objects can be detected or labelled (although we currently only use high-level geometric features such as rim, handle¹, because they are more informative of our actions than low-level features). In the lower part, the resulting effects of different actions on the object are recorded with binary values (+1 means successful and -1 otherwise). We consider some more-or-less common kitchen actions {grasping by closing fingers, rolling, cutting, grasping by expanding fingers, chopping}. It is worth noting that the strategies of feature labelling and action selection used in this paper are just one among many ways of describing the proposed model (section 2.2) and learning/reasoning procedure (section 3); they can be replaced by equivalent or more elaborate mechanisms. It should also be noted that in practice a very limited number of action experiments or simulations can be conducted on only a few objects, so incompleteness (or sparseness) of experimental data is a fact we have to deal with (Figure 2(b)).

2.2 Model Structure

In this paper, the object-action relations are modeled as shown in Figure 3. Actions and objects are represented in different spaces, that is, *action space* and *object space* re-

¹Such labels can be obtained by straightforward shape analysis systems.

spectively. The object space is composed of different layers that correspond to different actions. In each layer of the object space, the objects are linked pairwise (Figure 3), and the connection between a pair of objects is weighted proportionally to their similarity with respect to the corresponding action. The similarities between objects can be measured based on co-occurring properties or geometric features that can influence the outcome of the action. For instance, if object *A* (mug) and *B* (goblet) are both containers (therefore exhibit hollow structure), their similarities will be high in the “Grasp by expanding fingers” layer. However, their similarity would be low in the “roll” layer since *A* has a handle but *B* does not, and having a handle or not is a decisive factor for rolling.

In action space there is only one layer. Different actions are connected with each other, likewise with the connections weighted proportionally to their similarities. The similarities between actions can be interpreted as the similarities between their corresponding layers in object space.

3. MODEL LEARNING AND REASONING

With training data organized in the form of Figure 2(b), we straightforwardly apply homogeneity analysis [2, 6] to project all columns of Figure 2(b) to category quantifications and rows to object scores (section 3.1). However, the object scores computed by homogeneity analysis are the same for all actions, which does not fit our multi-layer object space (section 2.2). The underlying principle of our multi-layer object representations is that the dependencies between every action and object properties and geometric features are different; therefore, objects should be represented differently with respect to different actions. Meanwhile, the dependency and correlation relations between different basic properties, geometric features and actions are usually complicated. Two examples of such dependencies can be seen in Figure 4. It can be easily imagined that if a container is smaller than the gripper range in size, then it probably can

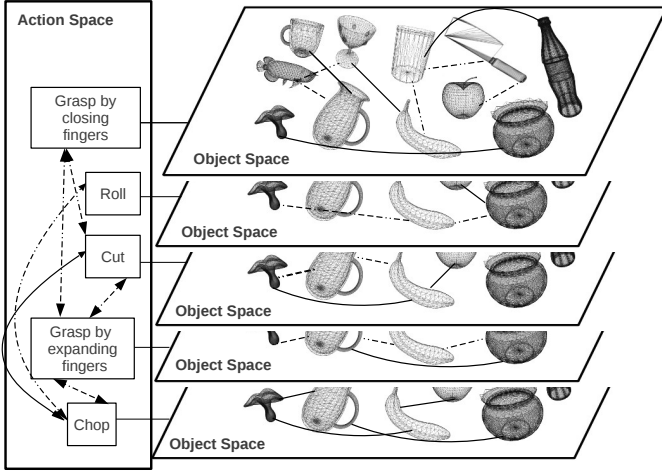


Figure 3: Object-action relational model. The object space is composed of action-specific layers, in which objects are interconnected (solid lines denote strong and dashed lines weak connections). There is only one layer in action space, and actions are connected in a similar way.

be graspable by expanding fingers, so there should be dependencies on “Container” and “<Gripper” for action “Grasp by expanding fingers”(Figure 4(a)). At the same time, containers smaller than the gripper are often made of ceramic or glass (e.g. bowls, mug, wineglass) in contrast to larger objects (e.g. plastic buckets or metal trash cans), so “Grasp by expanding fingers” might also be correlated with “Ceramic” and “Glass” (dashed lines). Similarly, usually an object is graspable if it is smaller than the gripper size or if it has a handle or rim, so it is reasonable to add dependencies between them (Figure 4(b)). Food items and plants are usually smaller than the gripper in a kitchen scenario, and they are unlikely to have handles. So extra dependencies on “Food” and “Plant” may be added as well. Instead of tediously reasoning about the dependencies for all actions, in section 3.2 a dependency checking mechanism is provided to remove unlikely or weak dependencies. The computed dependencies are also utilized to remap objects to different action layers with dependency weights.

3.1 Initial Learning with Homogeneity Analysis

Homogeneity analysis [2, 6] is a popular statistical tool for categorical multivariate analysis. Here we briefly review the procedure of homogeneity analysis with its application to object-action profile data. There are M object-action profiles in the dataset, each profile represented by a J -dimensional vector $O_i = [v_1, v_2, \dots, v_J]^T$ ($i = 1, \dots, M$), with each variable v_j denoting an attribute in the profile. Variable v_j takes on n_j categorical values (e.g., the action effect has binary values ± 1). By gathering the values of v_j over all M profiles in an $M \times n_j$ binary indicator matrix G_j , the whole set of indicator matrices can be gathered in a block matrix:

$$G = [G_1 | G_2 | \dots | G_J] \quad (1)$$

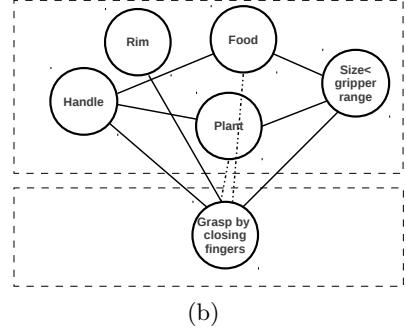
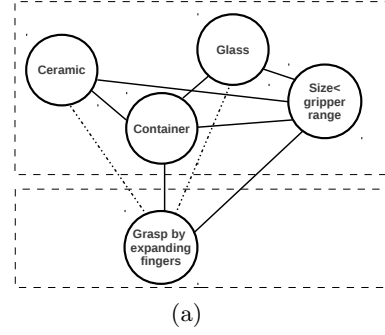


Figure 4: Two examples of dependencies between actions and objects’ basic properties and geometry features: (a) grasp by expanding fingers; (b) grasp by closing fingers.

The key feature of homogeneity analysis is that it simultaneously produces two projections to the same Euclidean space \mathbb{R}^p , one from J -dimensional profiles O_i , the other from the M -dimensional categorical attribute indicator vectors (columns of G). These projections are referred to as *object score* and *category quantification*, respectively [2, 6]. Suppose the collection of object scores is represented by an $M \times p$ matrix X , and category quantifications for variable v_j are represented by a $n_j \times p$ matrix Y_j . Then, the cost function of a projection can be formulated as:

$$f(X, Y_1, \dots, Y_J) = \frac{1}{J} \sum_{j=1}^J \text{tr}(X - G_j Y_j)^\top (X - G_j Y_j) \quad (2)$$

As emphasized above, in realistic cases the training dataset is usually sparse and incomplete, i.e., values of some v_j are missing. So for each G_j , we construct an $M \times M$ diagonal matrix S_j with diagonal values equal the sum of the rows of G_j , i.e., $S_j(i, i) = 0$ if the v_j value of O_i is missing. Then the corresponding cost function is

$$f(X, Y_1, \dots, Y_J) = \frac{1}{J} \sum_{j=1}^J \text{tr}(X - G_j Y_j)^\top S_j (X - G_j Y_j) \quad (3)$$

Usually two extra constraints are added to avoid trivial solution ($X = 0, Y_j = 0$):

$$\frac{1}{M} \mathbf{1}_{M \times 1}^\top S_* X = \mathbf{0} \quad (4)$$

$$\frac{1}{M} X^\top S_* X = I \quad (5)$$

Here, $S_* = \sum_{j=1}^J S_j$. The first constraint (4) essentially nor-

malizes the projected object scores to be centered around the origin. The second restriction (5) standardizes all p dimensions of object score by rescaling the square length of each dimension to M . In addition, another effect of (5) is that the p columns of X are imposed to be orthogonal to each other.

To minimize the cost function (3) under these constraints (4, 5), usually the alternating least squares (ALS) algorithm [2, 6] is used. The basic idea of ALS is to iteratively optimize with respect to X or to $[Y_1, \dots, Y_M]$ with the other held fixed. Assuming $X^{(0)}$ is provided arbitrarily at iteration $t = 0$, each iteration of ALS can be summarized as:

1. update Y_j :

$$Y_j^{(t)} = (G_j^\top S_j G_j)^{-1} G_j^\top X^{(t)}; \quad (6)$$

2. update X :

$$X^{(t+1)} = S_*^{-1} \sum_{j=1}^J G_j Y_j^t; \quad (7)$$

3. normalize X :

$$X^{(t+1)} = \text{Gram-Schmidt}(X^{(t+1)}). \quad (8)$$

It can be seen (6) that category quantification of Y_j is computed as the centroid of the object scores that belong to it. Step 2 (7) updates object scores X by taking the average of the quantifications of the categories it belongs to. In step 3 (8) a *Gram-Schmidt* procedure is used to find the normalized and orthogonal basis of updated object scores from the previous step.

3.2 Dependency Learning

According to the description in the previous section, the objects and action effects can be projected into two spaces (object scores X and category quantifications Y_j of action variables v_j) by applying homogeneity analysis on the set of object-action profiles. Although this observation is close to how we model object-action relations (section 2.2), there still exist some obstacles that prevent us from directly putting them to practical use. First, by using homogeneity analysis, basic properties, geometric features and action effects are simultaneously projected to their corresponding category quantifications without modelling their interrelations explicitly. As we illustrated in Figure 4, the dependency between them is an important factor in our object-action relational model, so we must disentangle how each action depends on different basic properties and geometric features. Secondly, in our model the objects are represented at different layers corresponding to different actions, while the representations of objects with homogeneity analysis are unique object scores. Hence, it is also required to strategically decompose the object scores into different action layers.

To resolve these two problems, some extra steps can be developed to exploit more information from the object scores and category quantifications. First, the J variables $[v_1, v_2, \dots, v_J]$ of each object O_i are divided into two groups, the object (variable) group V_o which covers basic properties and geometry features, and the action (variable) group V_a which contains action effects on the object O_i . We initially assume that each variable in action group $v_\beta^a \in V_a$ depends on all variables of the object group V_o . Then, for variable

v_β^a , we find its corresponding positive and negative category quantifications $Y_{\beta,+}^a$ and $Y_{\beta,-}^a$, and compute the distances between them and all categories' quantifications in the object group as

$$d(Y_{\beta,+/-}^a, Y_{\omega,k}^o) = \|Y_{\beta,+/-}^a - Y_{\omega,k}^o\|_2 \quad (9)$$

where $Y_{\omega,k}^o$ denotes the k -th category quantification of variable v_ω^o in the object group. We compute the maximum ratio between them as

$$\lambda_{\omega,k}^\beta = \max \left\{ \frac{d(Y_{\beta,+}^a, Y_{\omega,k}^o)}{d(Y_{\beta,-}^a, Y_{\omega,k}^o)}, \frac{d(Y_{\beta,-}^a, Y_{\omega,k}^o)}{d(Y_{\beta,+}^a, Y_{\omega,k}^o)} \right\} \quad (10)$$

and eliminate the dependencies between action variable v_β^a and category quantifications in V_o if

$$\frac{\lambda_{\omega,k}^\beta}{\sum_{\omega,k} \lambda_{\omega,k}^\beta} < \sigma \quad (11)$$

where $\sigma \in [0, 1]$ is a predefined threshold. The elimination criterion (11) is defined based on the concept that the object variables on which the action variable depends should have good discriminative abilities between its positive and negative categories.

Once the dependencies have been found, the second problem can be solved as well. Instead of computing object scores as the average of the all quantifications of the categories they belong to (7), the representations of objects in each action layer β are computed as the weighted average of quantifications of the (positive and negative) action categories and the category quantifications in V_o which the action is dependent on:

$$X_\beta = \hat{S}_{*,\omega,k}^{-1} \sum_{\omega,k \in \text{dependent}(\beta)} \pi_{\omega,k} \hat{G}_{\omega,k} \hat{Y}_{\omega,k} \quad (12)$$

where the $\hat{Y}_{\omega,k}$ are the category quantifications (out of n_ω) of variable v_ω^o on which action variable v_β^a depends. $\hat{G}_{\omega,k}$, \hat{S}_* are the corresponding indicator matrix and diagonal matrix. $\pi_{\omega,k}$ denotes the normalized dependency weights which reflect how β depends on quantifications in $\hat{Y}_{\omega,k}$:

$$\pi_{\omega,k} = \frac{\lambda_{\omega,k}^\beta}{\sum_{\omega,k \in \text{dependent}(\beta)} \lambda_{\omega,k}^\beta} \quad (13)$$

Correspondingly, the centroid of object representations which belongs to positive and negative category in β action layer is:

$$\beta_c^{+/-} = (G_{\beta,+/-}^\top S_{\beta,+/-} G_{\beta,+/-})^{-1} G_{\beta,+/-}^\top X_\beta \quad (14)$$

where $G_{\beta,+/-}$ is the positive/negative-category column in G_β and $S_{\beta,E}$ is corresponding diagonal counting matrix.

The dependencies between action variables can be also similarly learned to find the correlation or anti-correlation between object effects. Since our model is dedicated to relations between objects and actions, action-action relations will be exploited in our future work.

3.3 Reasoning

Given the object-action relational model learned with the procedure above, typical reasoning tasks are presented in Table 1. First, we discuss effect (E) prediction given object (O) and action (β). Assume O is an unseen object. Its representation in action layer β can be computed (12), and then

input	output	applications
object & action	effect	effect outcome prediction
action & effect	object	object selection
object & effect	action	action planing/recognition

Table 1: Typical applications of the object-action relation model.



Figure 5: Robot hand used for action labelling

the binary effect classification can be easily done by majority voting of the k -nearest neighbouring objects of training set (or using any other suitable classifier).

Second, the model can perform object (O) selection out of a set of candidates \mathbf{C} based on action (β) and effect ($E \in [-1, 1]$). Given the desired category E of action β , first object representations in candidate set $X_{\beta}^{(O \in \mathbf{C})}$ can be computed (12). Then the ratio of the distance between each $X_{\beta}^{(O)}$ and β_c^E to the distance between $X_{\beta}^{(O)}$ and β_c^{-E} (14) can be computed:

$$\phi_O = \frac{d(X_{\beta}^{(O)}, \beta_c^E)}{d(X_{\beta}^{(O)}, \beta_c^{-E})} \quad (15)$$

The optimal object O^\dagger is the one with smallest ϕ_O . Alternatively, with the ratios of all objects in \mathbf{C} computed, the object retrieval result can be ranked by their ratios in increasing order.

Finally, action selection or planning is also useful to find an optimal action among many that share similar semantic effects based on certain criteria. For example, both cutting and chopping are actions that break objects into smaller parts. However, they are executed with different tools (cleavers for chopping and knives for cutting) and with different strength. So if the task is to break an object O into parts with minimum strength from the higher-level planner, then one may want to perform a chopping action only if necessary. To this end, we compute the representation of O in cutting and chopping layers respectively and predict their corresponding effects, based on which the most energy-saving action will be selected.

4. EXPERIMENTS

4.1 Synthetic Database and Model Learning

To evaluate the proposed object-action relational model and learning method, we constructed a synthetic dataset of object-action profiles. We collected 140 kitchen objects (Figure 1) from the web [1] and annotated them as shown in Figure 2. The labeling and actions are set in the same way as described in section 2.1. Basic properties and high-level

geometry features² of objects were labelled by a student volunteer. The effects of different actions applied on objects are labeled as well based on common sense³. The robot gripper is presented to the labeller (Figure 5) for the consideration of different actions.

First, the model is learned with full and noisy-free data. By applying homogeneity analysis as described in section 3.1, we obtain 3-dimensional category quantifications of 10 variables in object-action profiles (Figure 6). With extra maximum ratio computation (10) (Figure 7), the dependency between each action and objects’ basic properties and geometric features are discovered (Table 2). Table 2 shows that “grasp by expanding fingers” and “grasp by closing fingers” exactly match our previous dependency analysis in Figure 4, i.e. the proposed model yields semantically reasonable object-action relations.

4.2 Reasoning Tasks

To quantitatively evaluate the proposed model, the following experiments test the model on two reasoning tasks, effect prediction and object selection⁴. In both experiments, the 140 object-action profiles are randomly divided into training set (100) and test set (40). In addition, as we already pointed out, in practice the empirical object-action data can be noisy and incomplete because of inaccuracy of perception systems and lack of real (or simulated) experiments. Therefore, to test the robustness of the model to noise and missing data, 10% noise are added and 20% entries are removed from the 100 training instances. The noise is generated by shifting the labels of variables with probability 0.1, and entries in Figure 2 are removed with probability 0.2.

Effect Prediction

According to the reasoning procedure described in section 3.3, 40 test objects are first projected to different representations at different action layers. Then the final effects of actions are decided by using a simple k -nearest-neighbour (KNN) classifier with the 100 representations of training objects. We use $k = 10$ for both full-data and missing-and-noisy-data conditions. We ran 50 trials in which different size-100 training (both full and missing-and-noisy) and size-40 test data sets are randomly generated. The average precision of correct effect classification of five actions are presented in Figure 8(a), from which it can be seen that the prediction results with both full training data and missing-and-noisy data are rather accurate, with the former slightly outperforming the latter (as is to be expected).

Object Selection

The object selection experiment is set up to test how accurate an object can be “recommended” to meet the effect of an action. The reasoning is based on the procedure in section 3.3, and the recommendation is ranked based on ratios (15). Similarly to the effect-prediction experiment, 50 trials with different training and test data are run, and the average results of 5 actions (positive and negative) are pre-

²We did not use low-level geometric features in our experiments.

³In future work, we plan to use simulated and ultimately physical robotic action.

⁴Since action selection applications usually require higher-level planners to handle constraints, robustness criteria etc., we did not consider them in our pilot experiments.

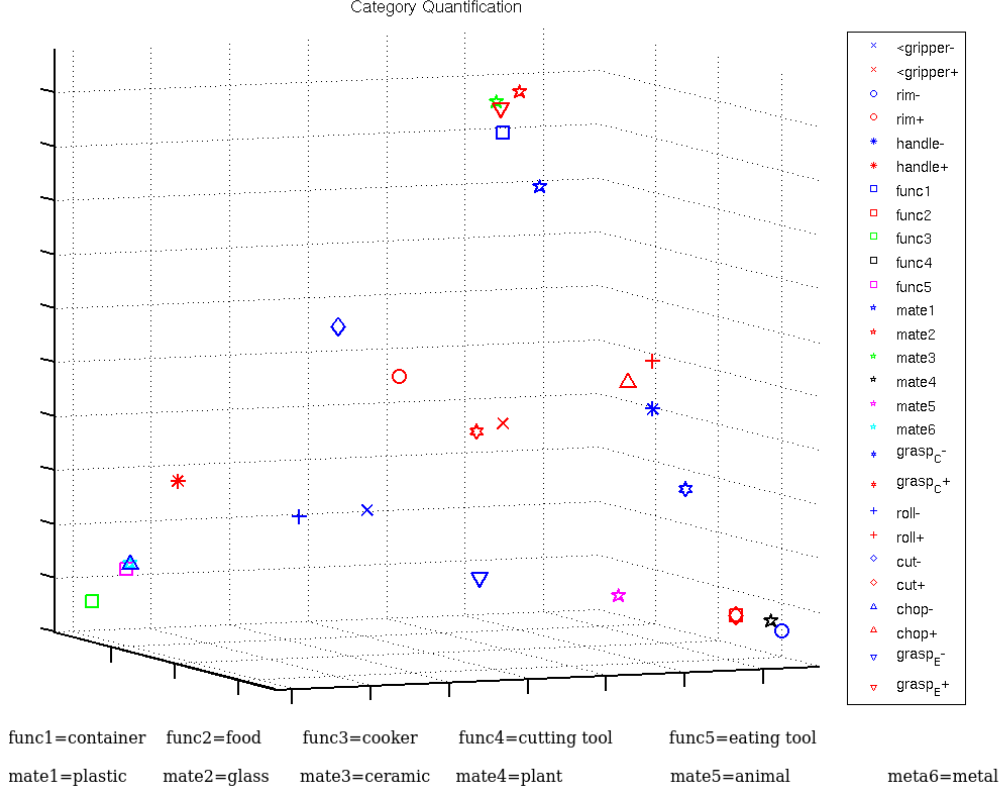


Figure 6: Category quantifications of variables in object-action profiles (best viewed in color).

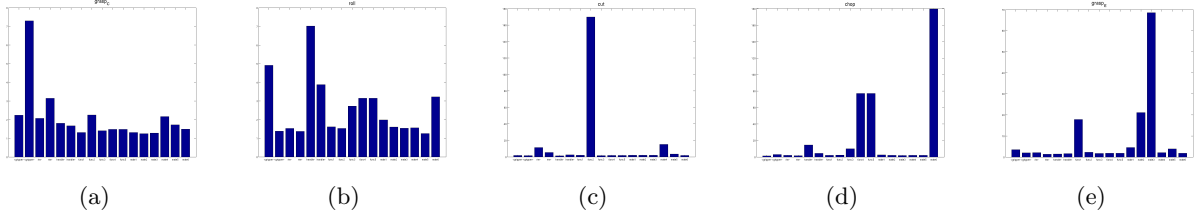


Figure 7: Check the dependency of five actions ((a) grasp by closing fingers (b) roll (c) cut (d) chop (e) grasp by expanding fingers) on category quantifications of object variables (from left to right bars denotes the maximum ratios (10) of <gripper-, <gripper+, rim-,rim+, handle-, handle+, container, food, cooker, cutting tool, eating tool, plastic, glass, ceramic, plant, animal, metal).

V_a	Depended category quantification of variable in V_o
$grasp_C$	<gripper-, <gripper+, handle-, rim-, rim+,function=food, material=plant
$roll$	<gripper-, handle-, handle+, function=cooker,function=cutting tool, function=eating tool, material=metal
cut	function=food, material=plant
$chop$	function=cutting tool, function=eating tool, material=metal
$grasp_E$	function=container, material=glass, material=ceramic

Table 2: Dependency of five actions on category quantifications of object variables after elimination (11).

sented in Figure 8(b)-8(f) with precision-recall curves. It can be seen that except for the poor results on grasping by closing fingers, object retrieval of all other actions and effects are acceptable. The reason for poor performance in the negative case of grasping by closing fingers, according to our preliminary analysis, is that there are too few in-

stances of $grasp_C$ - in the training data; most objects in the kitchen are graspable. The results with missing-and-noisy training data are slightly inferior to those with full training data. Two obvious performance gaps appear in the negative case of chopping, and in the positive case of grasping by expanding fingers. In conclusion, both effect prediction and

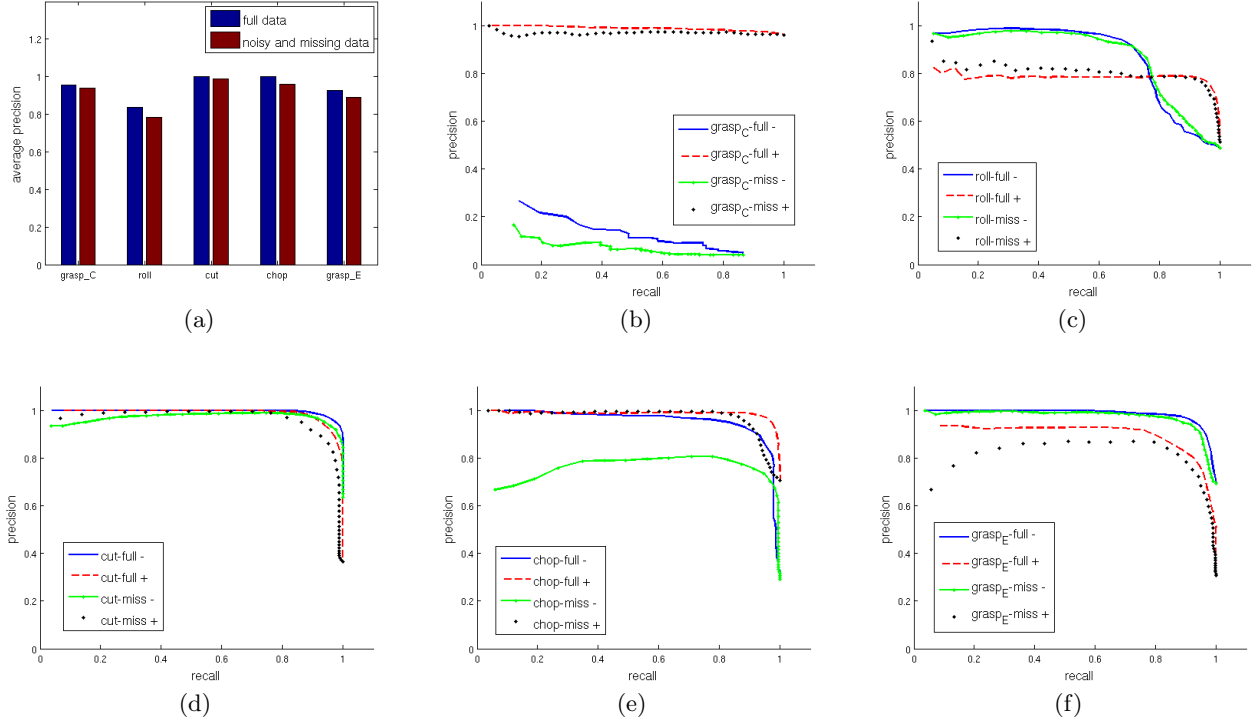


Figure 8: (a) The average precision of correct effect prediction of five actions; (b)-(f) the precision-recall curves of average object selection results in all positive and negative of five actions.

object selection experiments quantitatively demonstrate the promising capabilities of our object-action relational model by displaying its high accuracies and robustness to noisy and incomplete data.

5. CONCLUSION

We presented a novel computational model of object-action relations. Actions are represented in terms of their effects on objects, and objects are represented as well in an action-oriented manner. The model can be effectively learned with homogeneity analysis and extra discovery of dependencies between action and object variables. One strength of the proposed model is that it does not require complex, highly-combinatorial descriptions of objects and actions. The object representations with respect to different actions are computed with only a small number of the most decisive object variables. Actions are presented by their positive and negative action-effect category quantifications. Another merit of the model, according to experimental results, is that it is robust to noisy and missing data, which is an unavoidable problem in practice.

6. REFERENCES

- [1] www-roc.inria.fr/gamma/download/.
- [2] J. de Leeuw and P. Mair. Homogeneity Analysis in R: The Package homals. . Technical report, Department of Statistics, UCLA, 2007.
- [3] R. Detry, C. H. Ek, M. Madry, J. Piater, and D. Kragić. Generalizing Grasps Across Partly Similar Objects. In *International Conference on Robotics and Automation*, pages 3791–3797. IEEE, 2012.
- [4] R. Detry, D. Kraft, O. Kroemer, L. Bodenhagen, J. Peters, N. Krüger, and J. Piater. Learning Grasp Affordance Densities. *Paladyn Journal of Behavioral Robotics*, 2(1):1–17, 2011.
- [5] J. J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.
- [6] G. Michailidis and J. de Leeuw. The Gifi System of Descriptive Multivariate Analysis. *Statistical Science*, 13:307–336, 1998.
- [7] B. Moldovan, P. Moreno, M. van Otterlo, J. Santos-Victor, and L. De Raedt. Learning relational affordance models for robots in multi-object manipulation tasks. In *IEEE International Conference on Robotics and Automation, ICRA 2012*, pages 4373–4378, May 2012.
- [8] L. Montesano and M. Lopes. Learning grasping affordances from local visual descriptors. In *IEEE 8TH International Conference on Development and Learning*, China, 2009.
- [9] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor. Learning object affordances: From sensory-motor coordination to imitation. *IEEE Transactions on Robotics*, 24(1):15–26, Feb 2008.
- [10] E. Oztop, N. Bradley, and M. Arbib. Infant grasp learning: a computational model. *Experimental Brain Research*, 158(4):480–503, 2004.