Homogeneity Analysis for Object-Action Relation Learning

Xperience Summer School 2013

Hanchen Xiong

Institute of Computer Science
University of Innsbruck, Austria

October 4, 2013

Motivation

Object-Action Modeling and Learning: Enable an agent to discover manipulation knowledge from empirical data, based on which, different tasks can be done in a data-driven way.

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

Table: Applications of the object-action relation model.

Challenges:
(1) complex and (most of them are) useless representation of objects and actions;
(2) Few, incomplete and noisy empirical data.

Limited Scenario

Within a limited scenario, data can be probably enough.

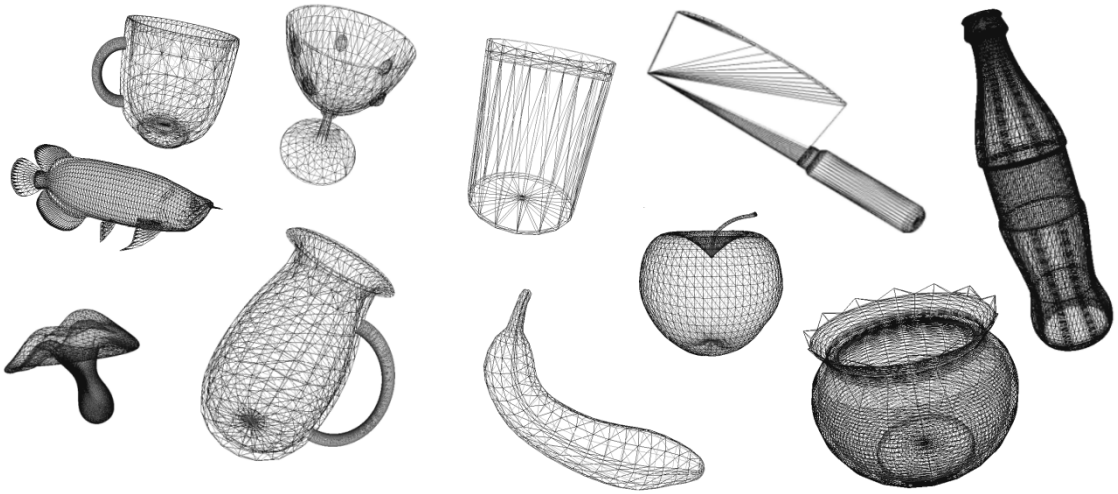


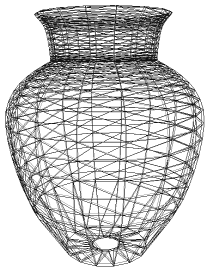
Figure: A sample set of objects in kitchen scenario

Limited Representations

- Effect representations: $E \in \{-1, 1\}$
- Object representations: $O_i = [v_1, v_2, \cdot, v_K]^\top$, where $v_k \in [1, N_k]$ (collection of discrete-valued attributes).

Data Structure

Object-Action Profiles:



Size:
Bigger
than gripper
range -

Low-Level Geometry Information:
3D features: *e.g. edges, curvatures*
2D features: *e.g. contours, sketches*

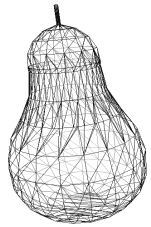
High-Level Geometry Information:
3D part: *e.g. rim* +
handle -

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	-

Data Structure

How training data looks like:

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

Figure: A collection of object-action profiles, red * denotes missing data

Modeling

The proposed model:

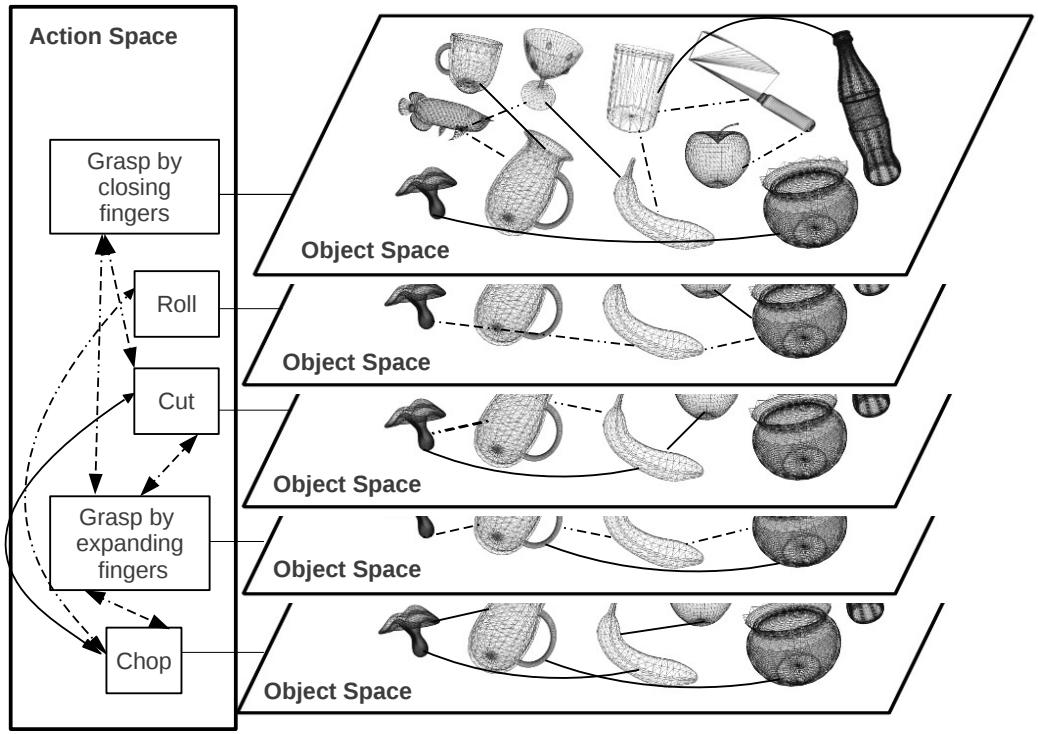
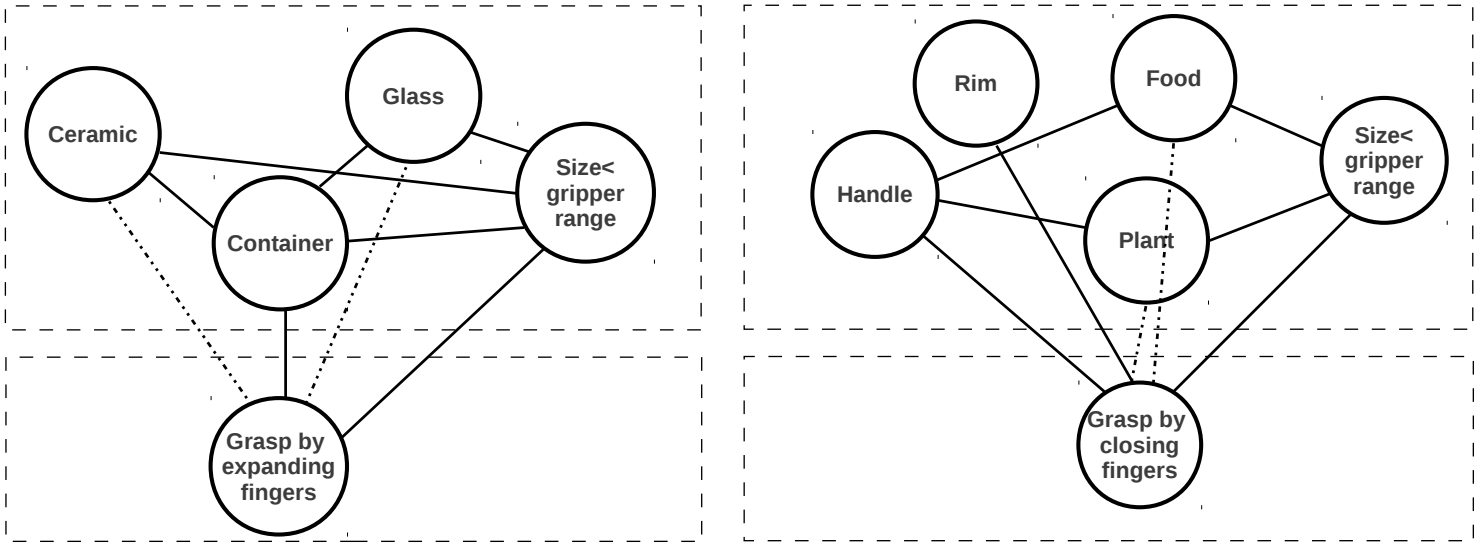


Figure: Object-action relation modeling: the object space is composed with many layers, in which objects are connected each other (solid lines denote strong connections and dashed lines for weak connections; there is only one layer in action space, and actions are connected similarly.

Difficulties of Model Learning



(a) (b)

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

Learning with Homogeneity Analysis

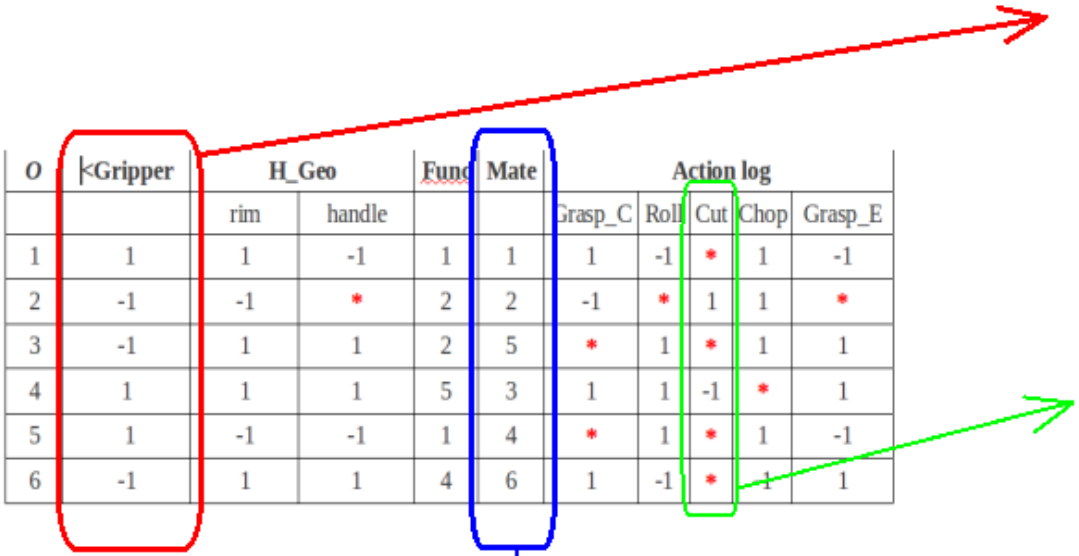
Homogeneity analysis is a popular statistics tool for categorical multivariate analysis.

Assume there are M object-action profiles in the dataset, each profile is represented by a J-dimensional vector $O_i = [v_1, v_2, \cdots, v_J]^\top (i \in [1, M])$ with each variable v_j denotes an attribute in the profile. For variable v_j, it takes on n_j categorical values (e.g. the action effect has binary values: $+1$ and -1).

Our data is categorical and multivariate, so straightforward.

How Homogeneity Analysis Works

Data Reformulation By gathering the values of v_j over all M object-action profiles in an $M \times n_j$ binary indicator matrix $G_j, j \in \{1, 2, \cdots J\}$



O	<Gripper	H_Geo		Func	Mate	Action log				
		rim	handle			Grasp_C	Roll	Cut	Chop	Grasp_E
1	1	1	-1	1	1	1	-1	*	1	-1
2	-1	-1	*	2	2	-1	*	1	1	*
3	-1	1	1	2	5	*	1	*	1	1
4	1	1	1	5	3	1	1	-1	*	1
5	1	-1	-1	1	4	*	1	*	1	-1
6	-1	1	1	4	6	1	-1	*	1	1

<Gripper	-1	1
O_1	0	1
O_2	1	0
O_3	1	0
O_4	0	1
O_5	0	1
O_6	1	0

Cut	-1	1
O_1	0	0
O_2	0	1
O_3	0	0
O_4	1	0
O_5	0	0
O_6	0	0

Mate	1	2	3	4	5	6
O_1	1	0	0	0	0	0
O_2	0	1	0	0	0	0
O_3	0	0	0	0	1	0
O_4	0	0	1	0	0	0
O_5	0	0	0	1	0	0
O_6	0	0	0	0	0	1

How Homogeneity Analysis Works

The key strength of homogeneity analysis is that **it can 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

How Homogeneity Analysis Works, cont.

Denoting $X \in \mathbb{R}^p$ as the object score vector, and $Y_j \in \mathbb{R}^{n_j \times p}$ as the category quantifcation matrix of v_j, then the objective function is:

$$f(X, Y_1, \cdots, Y_J) = \frac{1}{J} \sum_{j=1}^{J} \mathbf{tr}(X - G_j Y_j)^\top (X - G_j Y_j) \tag{1}$$

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

$$f(X, Y_1, \cdots, Y_J) = \frac{1}{J} \sum_{j=1}^{J} \mathbf{tr}(X - G_j Y_j)^\top S_j (X - G_j Y_j) \tag{2}$$

$$\frac{1}{M} \mathbf{1}_{M\times 1}^\top S_* X = \mathbf{0} \tag{3}$$

$$\frac{1}{M} X^\top S_* X = I \tag{4}$$

How Homogeneity Analysis Works, cont.

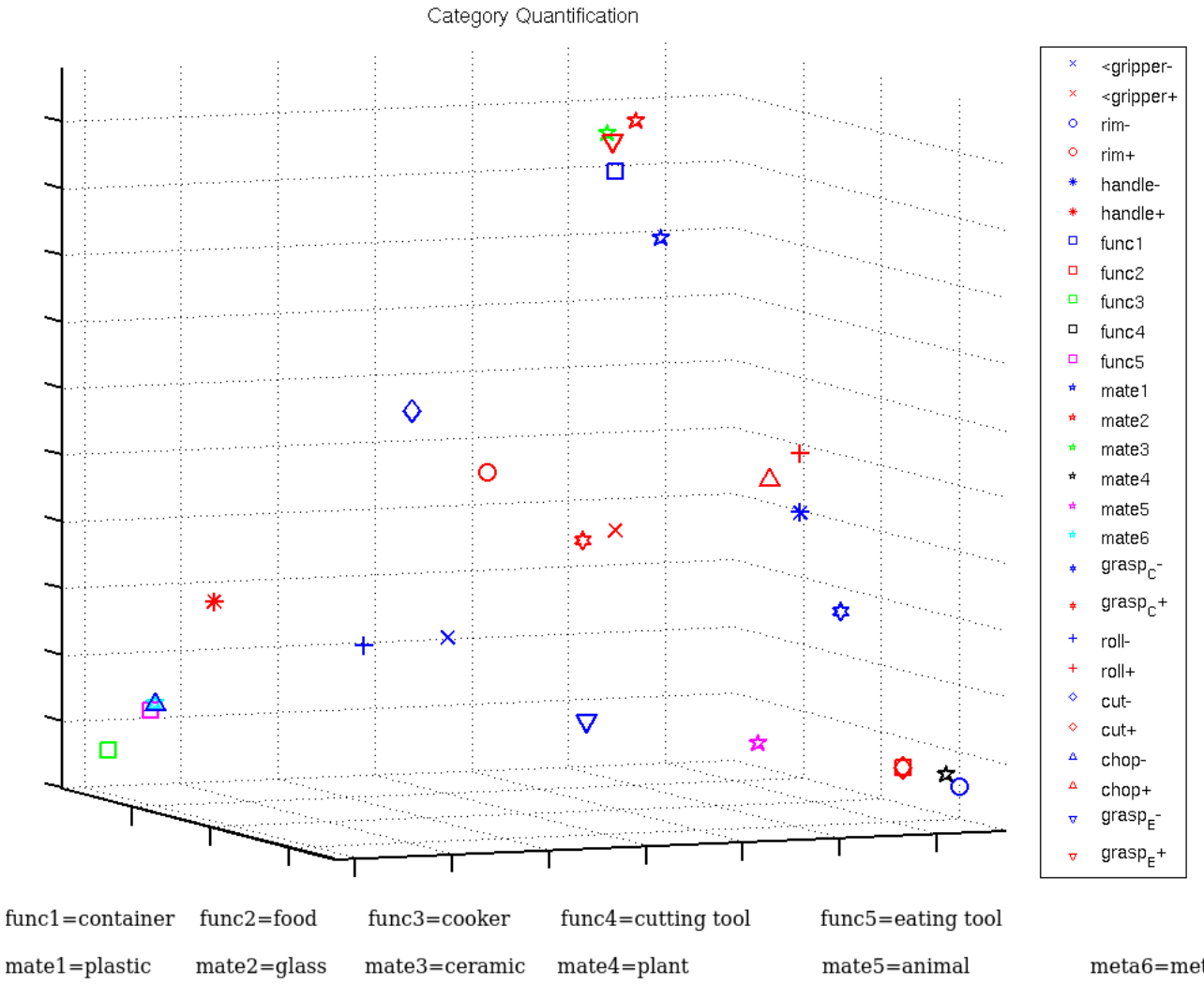
$$f(X, Y_1, \cdots, Y_J) = \frac{1}{J} \sum_{j=1}^{J} \mathbf{tr}(X - G_j Y_j)^\top S_j (X - G_j Y_j) \tag{5}$$

alternating least squares (ALS) algorithm is used. The basic idea of ALS is to iteratively optimize with respect to X or to $[Y_1, \cdots, Y_M]$ 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:
$$Y_j^{(t)} = (G_j^\top S_j G_j)^{-1} G_j^\top X^{(t)}; \tag{6}$$
2. update X:
$$X^{(t+1)} = S_*^{-1} \sum_{j=1}^{J} G_j Y_j^t; \tag{7}$$
3. normalize X:
$$X^{(t+1)} = \textit{Gram-Schmidt}(X^{(t+1)}). \tag{8}$$

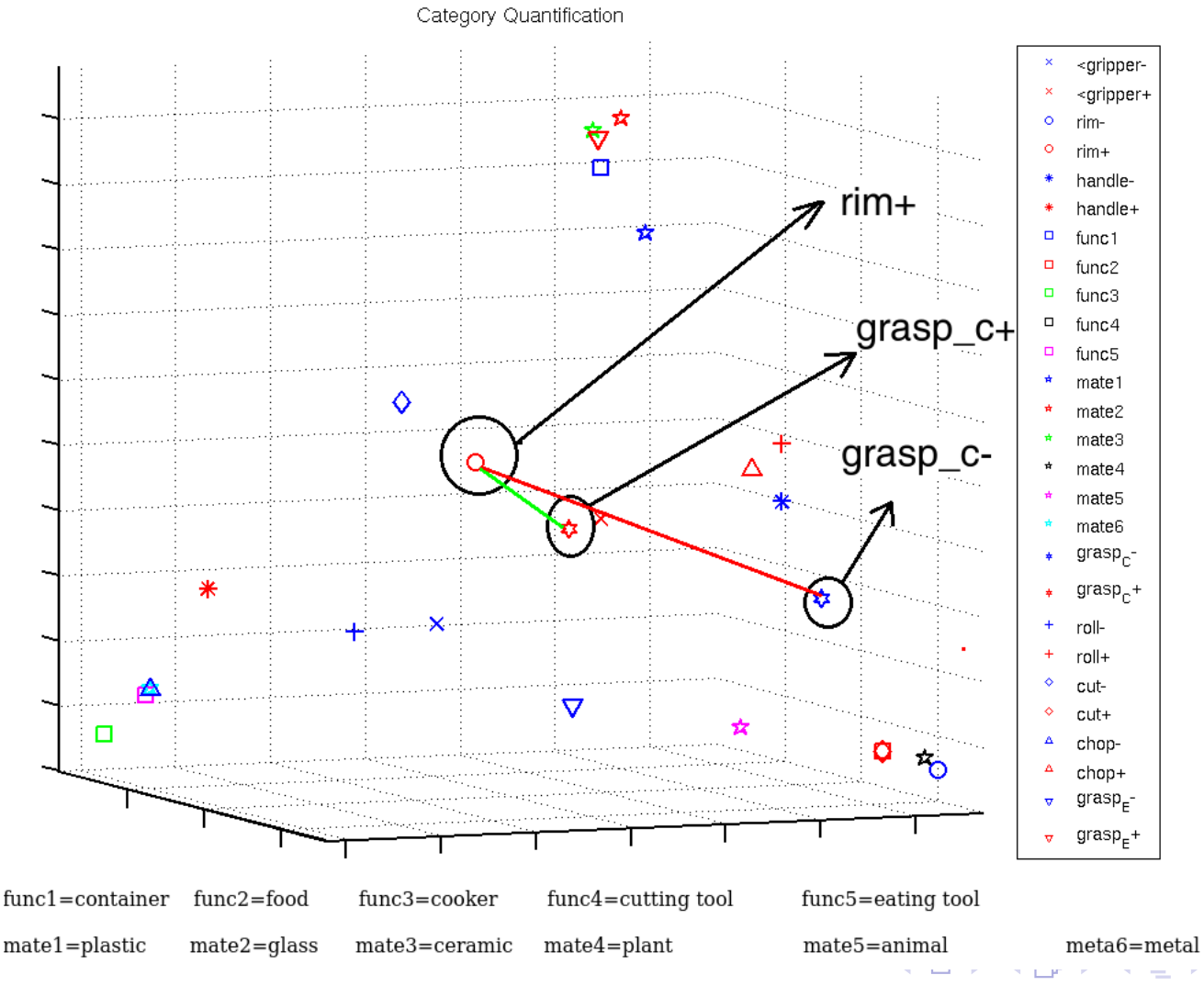
How Homogeneity Analysis Works, cont.

How Object Scores X_i and Category Quantifications Y_j look like



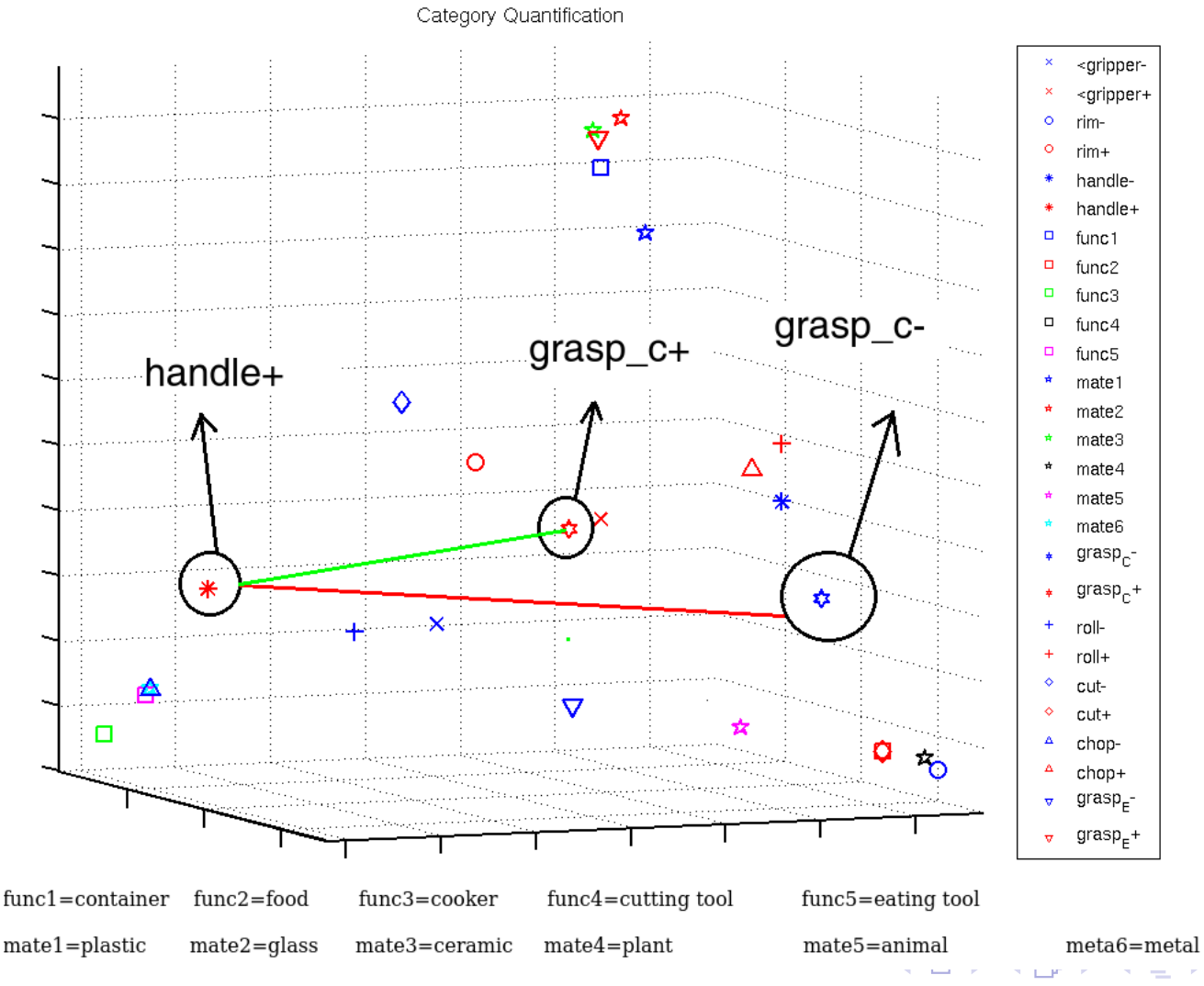
How Homogeneity Analysis Works, cont.

How Object Scores X_i and Category Quantifications Y_j look like



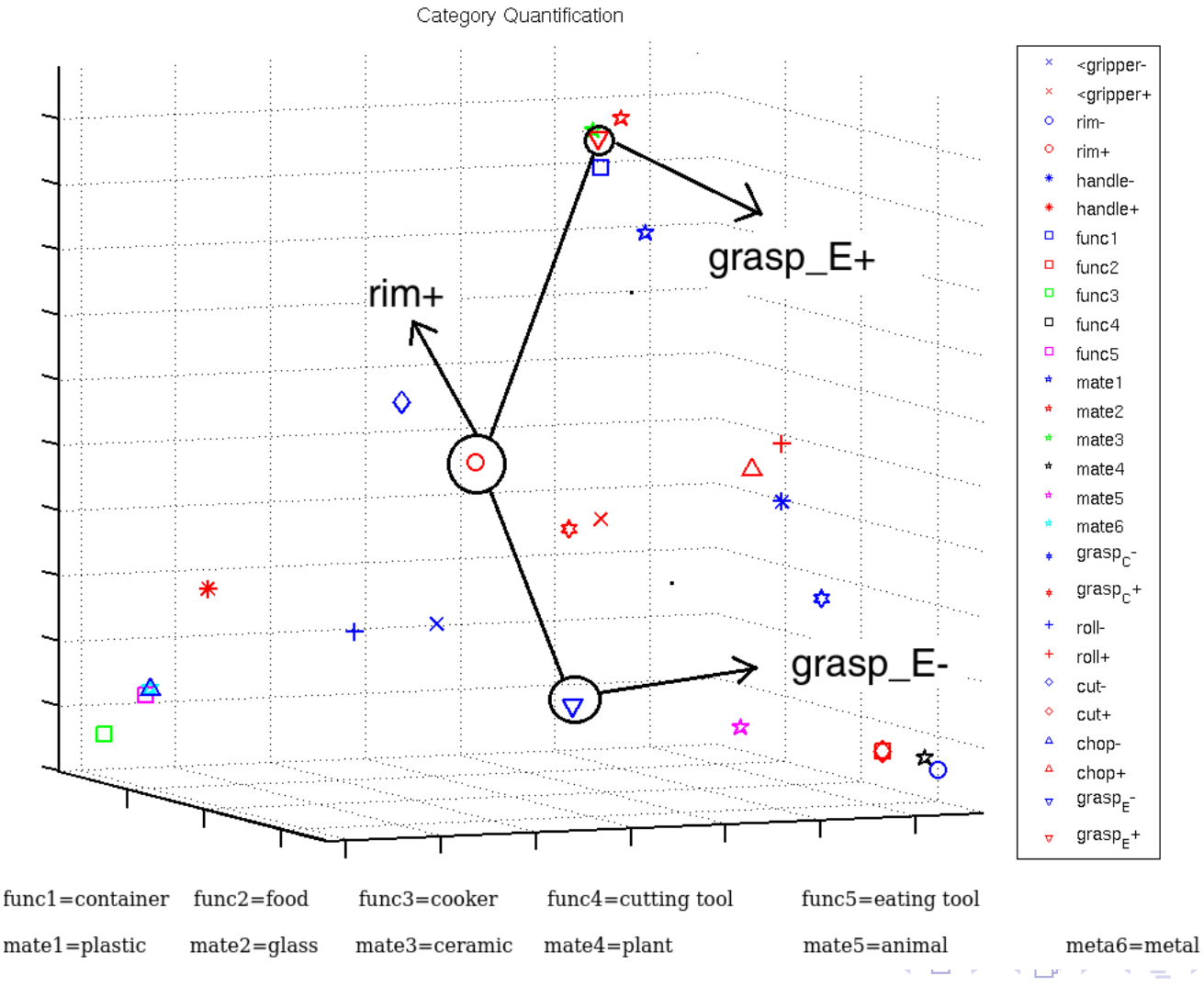
How Homogeneity Analysis Works, cont.

How Object Scores X_i and Category Quantifications Y_j look like



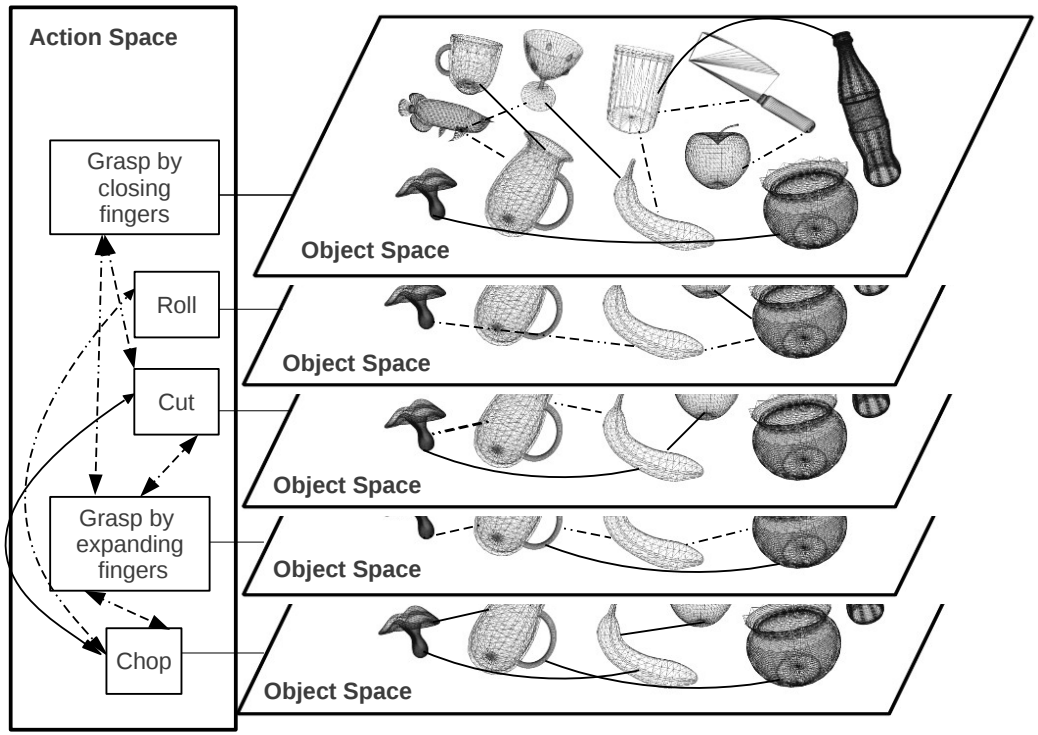
How Homogeneity Analysis Works, cont.

How Object Scores X_i and Category Quantifications Y_j look like



Dependency Learning

Homogeneity analysis provides us Object Scores X_i and Category Quantifications Y_j, we are very close but not exactly



Dependency Learning, cont.

First, the J variables $[v_1, v_2, \ldots, v_J]$ 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.

Dependency Learning, cont.

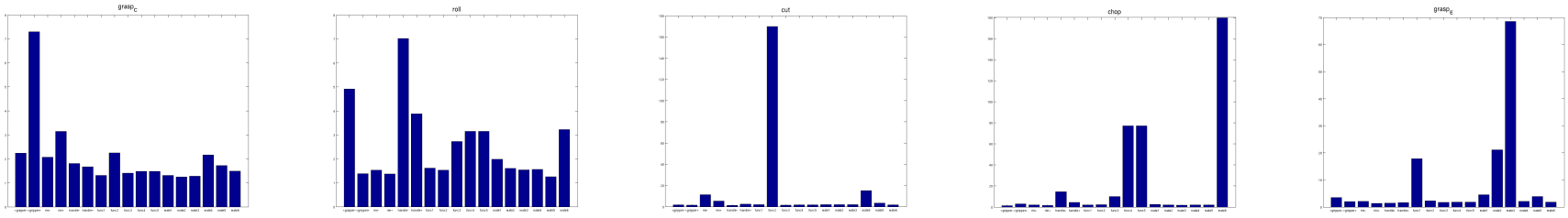
Then, for variable v^a_β, we find its corresponding positive and negative category quantifications $Y^a_{\beta,+}$ and $Y^a_{\beta,-}$, and compute the distances between them and all categories' quantifications in the object group as

$$d(Y^a_{\beta,+/-}, Y^o_{\omega,k}) = ||Y^a_{\beta,+/-} - Y^o_{\omega,k}||_2 \tag{9}$$

where $Y^o_{k,w}$ denotes the k-th category quantification of variable v^o_ω in the object group. We compute the maximum ratio between them as

$$\lambda^\beta_{\omega,k} = \max\left\{\frac{d(Y^a_{\beta,+}, Y^o_{\omega,k})}{d(Y^a_{\beta,-}, Y^o_{\omega,k})}, \frac{d(Y^a_{\beta,-}, Y^o_{\omega,k})}{d(Y^a_{\beta,+}, Y^o_{\omega,k})}\right\} \tag{10}$$

Dependency Learning, cont.



(a) (b) (c) (d) (e)

Figure: 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+, handle-, handle+, rim-,rim+, container, food, cooker, cutting tool, eating tool, plastic, glass, ceramic, plant, animal, metal).

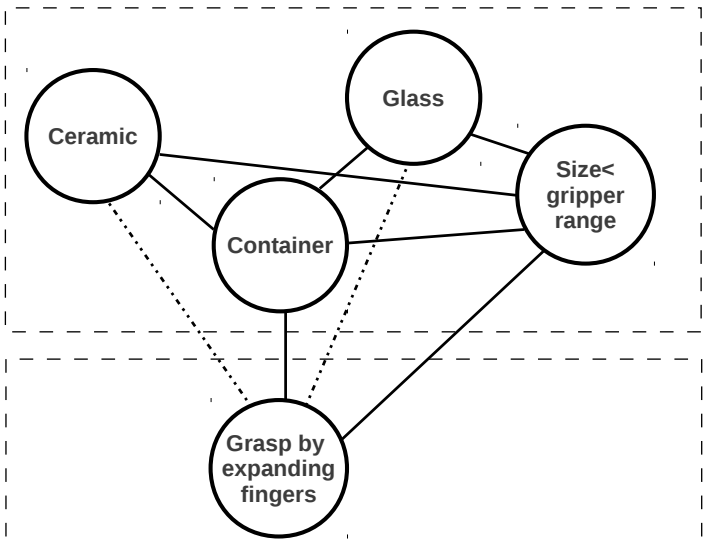
Eliminate the dependencies between action variable v_β^a and category quantifications in V_0 if

$$\frac{\lambda_{\omega,k}^{\beta}}{\sum_{\omega,k} \lambda_{\omega,k}^{\beta}} < \sigma \tag{11}$$

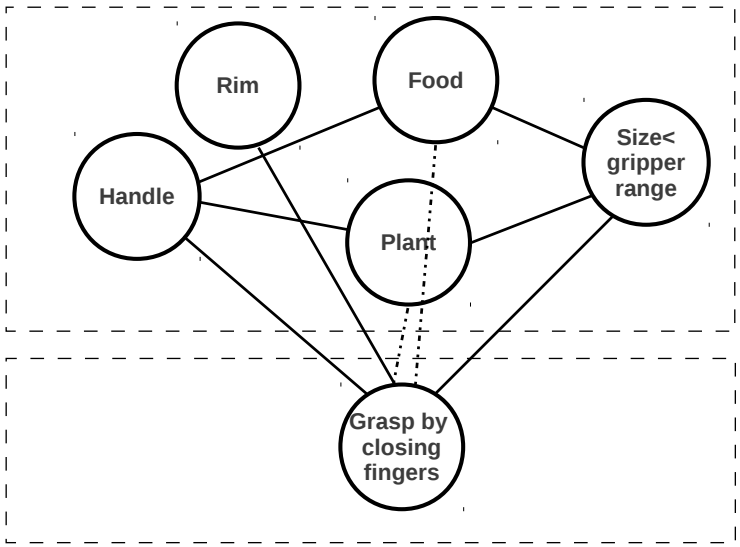
Dependency Learning, cont.

Action variable in 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

(a)



(b)



(c)

Object Scores Decomposition

$$X^{(t+1)} = S_*^{-1} \sum_{j=1}^{J} G_j Y_j^t; \tag{12}$$

(12) updates object scores X by taking the average of the quantifications of the categories it belongs to.
Object score/representation at β action layer

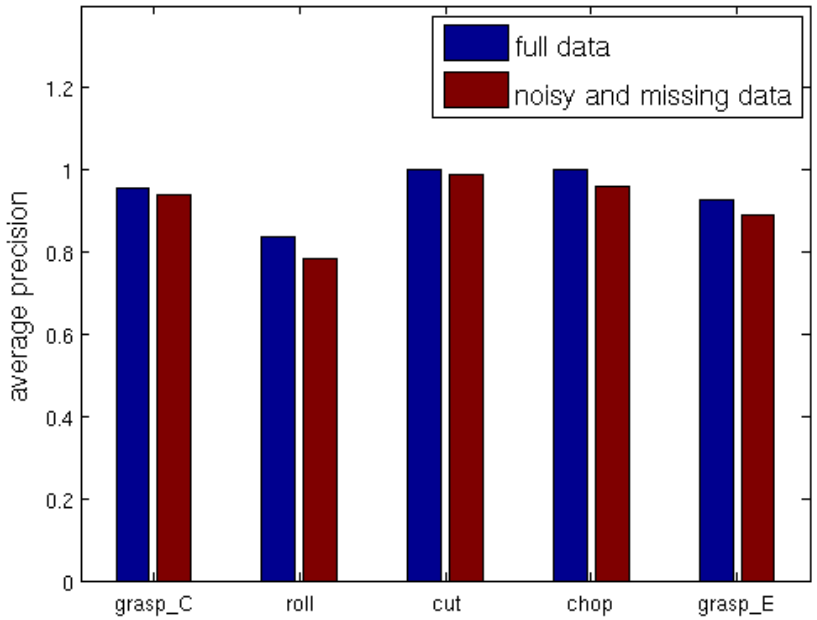
$$X_\beta = \hat{S}_{*,\omega,k}^{-1} \sum_{\omega,k \in \text{dependent}(\beta)} \pi_{\omega,k} \hat{G}_{\omega,k} \hat{Y}_{\omega,k} \tag{13}$$

$\pi_{\omega,k}$ denotes the normalized dependency weights which reflect how β depends on quantifications in $\hat{Y}_{\omega,k}$:

$$\pi_{\omega,k} = \frac{\lambda_{\omega,k}^{\beta}}{\sum_{\omega,k \in \text{dependent}(\beta)} \lambda_{\omega,k}^{\beta}} \tag{14}$$

Action Effect Prediction

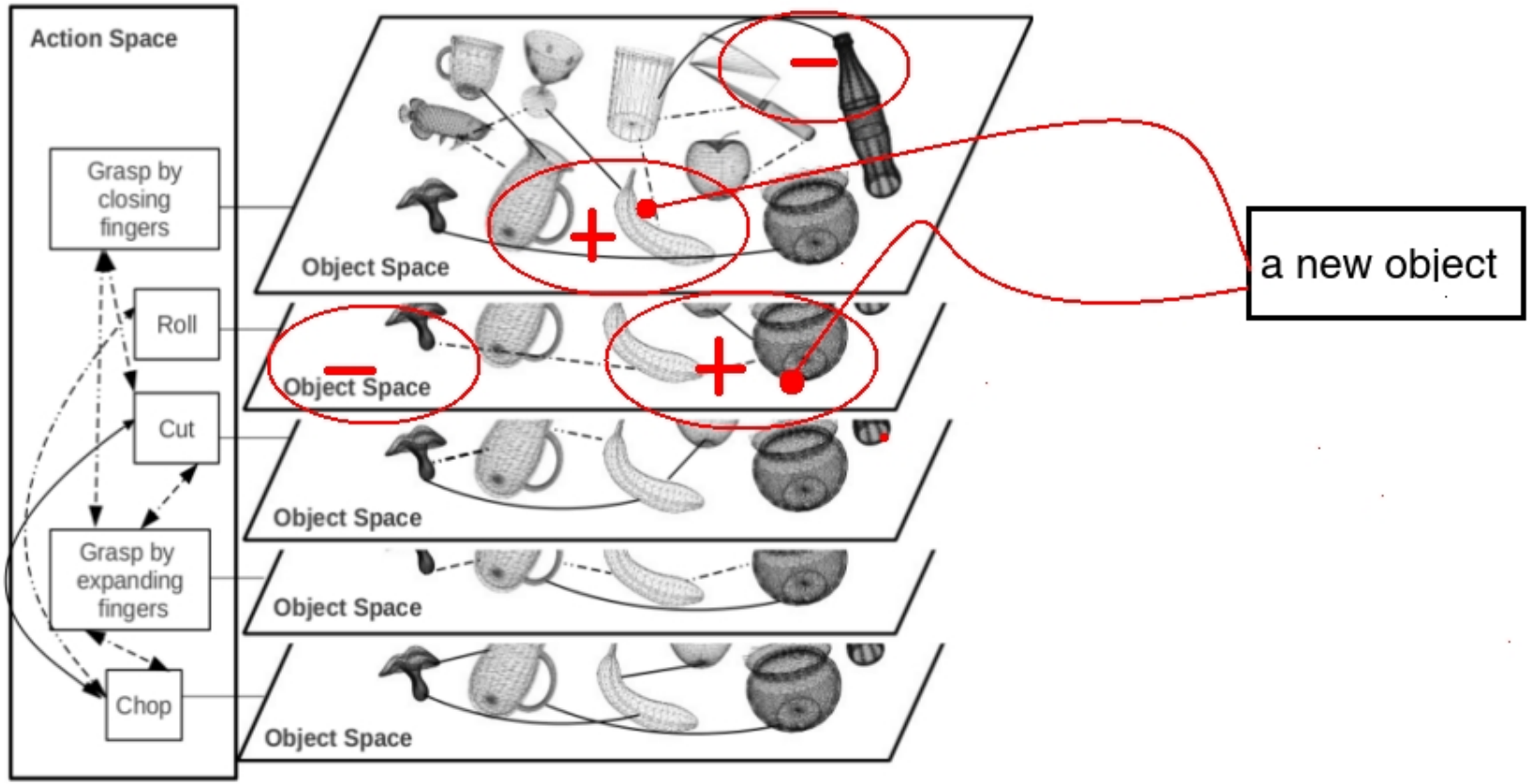
Assume O is an unseen object. Its representation in action layer β can be computed (13), and then the binary effect classification can be easily done by majority voting of the k-nearest neighbouring objects of training set.



(d)

Figure: The average precision of correct effect prediction of five actions.

Object Selection, cont.



Object Selection

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 (13). 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} can be computed:

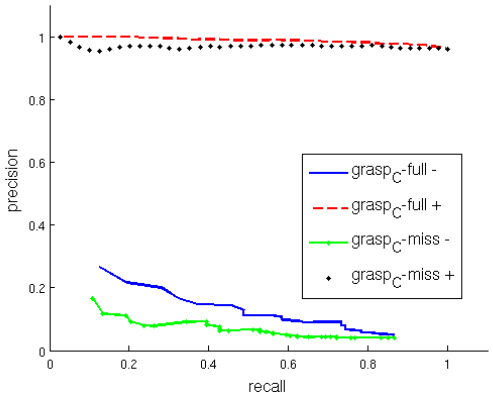
$$\phi_O = \frac{d(X_\beta^{(O)}, \beta_c^E)}{d(X_\beta^{(O)}, \beta_c^{-E})} \tag{15}$$

where $\beta_c^{+1/-1}$ are the centroids of object representations which belongs to positive and negative category in β action layer:

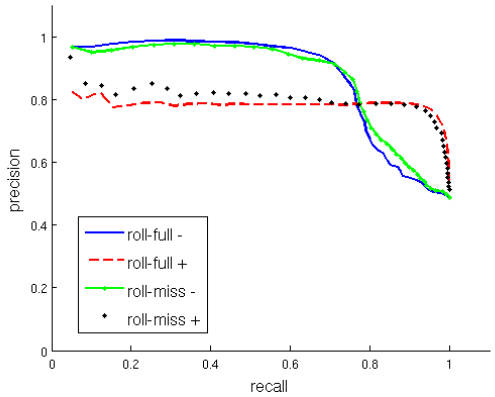
$$\beta_c^{+/-} = (G_{\beta,+/-}^\top S_{\beta,+/-} G_{\beta,+/-})^{-1} G_{\beta,+/-}^\top X_\beta \tag{16}$$

Object Selection, cont.

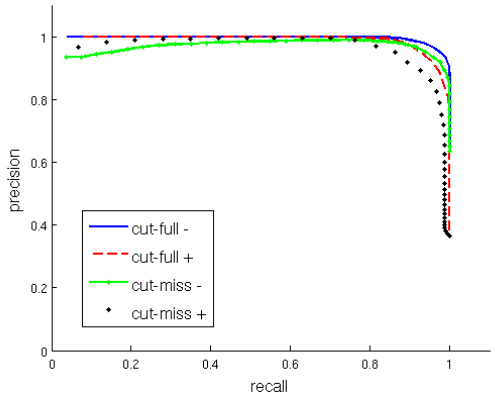
The optimal object $O^\dagger$ is the one with smallest ϕ_O. Alternatively, with the ratios of all objects in $\mathbf{C}$ computed, the object retrieval result can be ranked by their ratios in increasing order.



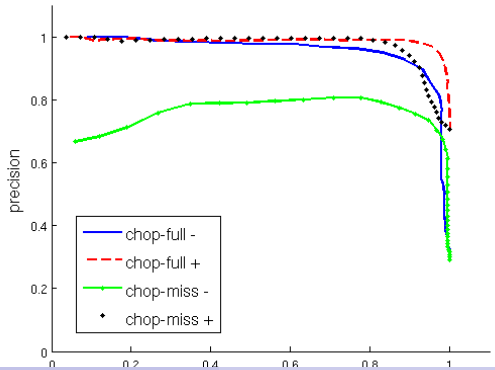
(a)

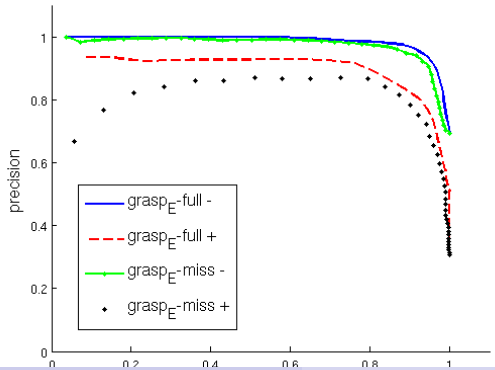


(b)



(c)





Conclusive Remarks

- object-action relations are exploited;
- multi-layer structure, in which actions are represented object-oriented manner, and objects are represented in a semi action-oriented manner;
- novel object-action relation is straightforward with multi-layer presentations;

Future Work

- action-action relations are also straightforward;
- grounding with real-features (low-level and high level);
- go further to parameters level.

END