Learning Human Priors for Task-Constrained Grasping

Martin Hjelm[1], Carl Henrik Ek[1](✉), Renaud Detry[2], and Danica Kragic[1]

[1] Autonomous Systems and the Computer Vision and Active Perception Lab, CSC, KTH Royal Institute of Technology, Stockholm, Sweden
{martinhjelm,chek,danik}@csc.kth.se
[2] University of Liège, Liège, Belgium
renaud.detry@ulg.ac.be

Abstract. An autonomous agent using manmade objects must understand how task conditions the grasp placement. In this paper we formulate task based robotic grasping as a feature learning problem. Using a human demonstrator to provide examples of grasps associated with a specific task, we learn a representation, such that similarity in task is reflected by similarity in feature. The learned representation discards parts of the sensory input that is redundant for the task, allowing the agent to ground and reason about the relevant features for the task. Synthesized grasps for an observed task on previously unseen objects can then be filtered and ordered by matching to learned instances without the need of an analytically formulated metric. We show on a real robot how our approach is able to utilize the learned representation to synthesize and perform valid task specific grasps on novel objects.

1 Introduction

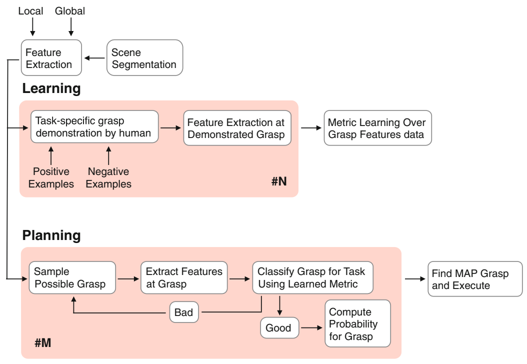
This paper focuses on the problem of how an autonomous agent can discover how different tasks, conditions grasping. The challenge in task based grasping is two-fold, first the agent needs to understand the task requirements, secondly it needs to translate these to the object. Formulating the requirements explicitly is not likely to scale beyond simple tasks and objects. Therefore a promising approach is to try to infer the requirements from data using a learning by demonstration approach.

As an example, given a set of demonstrations of pouring, the agent should realize and generalize that the actual constraints lies in, not covering the opening, and in the additional constraints that makes a placed grasp on a specific object successful. This is the fundamental challenge facing data-driven task based grasping.

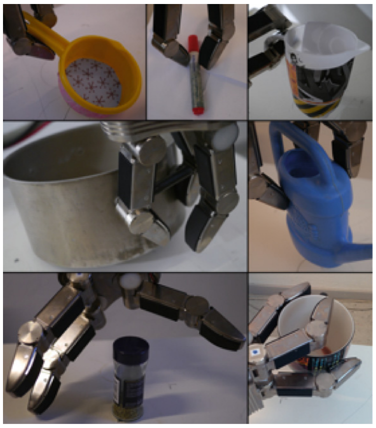
An important body of work on task based grasping, using a generative model, is Song et al. [1,2]. The authors build a Bayesian Network around gripper, object,

This work was supported by the Swedish Foundation for Strategic Research, the Belgian National Fund for Scientific Research (FNRS), the Swedish Research Council, and the EU project EU ERC FLEXBOT.

L. Nalpantidis et al. (Eds.): ICVS 2015, LNCS 9163, pp. 207–217, 2015.
DOI: 10.1007/978-3-319-20904-3_20


Local
Global
Feature Extraction
Scene Segmentation
Learning
Task-specific grasp demonstration by human
Feature Extraction at Demonstrated Grasp
Metric Learning Over Grasp Features data
Positive Examples
Negative Examples
#N
Planning
Sample Possible Grasp
Extract Features at Grasp
Classify Grasp for Task Using Learned Metric
Find MAP Grasp and Execute
Bad
Good
Compute Probability for Grasp
#M

(a) Flow diagram over learning and grasp planning.



(b) Grasps executed using learned task constraints. Handle, cork, pouring, and lift up.

Fig. 1. (a) Flow diagram over learning and grasp planning. (b) Grasps executed using learned task constraints. Handle, cork, pouring, and lift up.

and task parameters; making it possible to condition on task and synthesize task based grasps. Even though the approach shows impressive results on synthetic data, building statistical models of all observations, will place significant constraints on the dimensionality of the observations. It is thus questionable how the approach will perform when the number of tasks and objects increase.

A number of papers instead applies a discriminative approach and learns distributions over local features of the object, Saxena et al. [3], Boularis et al. [4], etc. This avoids the complexity of the generative model but still requires a substantial amount of, often manually, labeled data, which is expensive to obtain.

Rather than using features a successful approach is to represent objects as a set of local templates. Grasping can then be learned on a template basis allowing for transfer of grasps between different objects. The template systems of Detry et al. [5], Herzog et al. [6], Kroemer et al. [7], and Ying et al. [8] have been shown to give good results. The main drawback of these template based systems is their analytical approaches to similarity and the complexity in constructing the template structure. Our method avoids both these drawbacks by learning similarity from the data by learning a representation and utilizing it for matching when synthesizing grasps.

Selecting the features that are relevant for a task to model task constraints is linked with affordances. Stark et al. [9] detects affordance cues, functional parts of the object that the human demonstrator uses for grasping. The agent then uses the cues to plan grasps on objects. Aleotti et al. [10] presents a similar idea which represents objects as a topology. Each demonstrated grasp is associated with one part of the topology. The agent then formulates task constraints around the associated parts.

Instead of trying to decompose or detect functional parts of objects our method revolves around relating previous grasping experience and grounding task constraints in sensory data, similarly to our work in [11]. The features captured from the human demonstrator can therefore be considered as object specific affordance cues that our method learns to generalize.

2 Methodology

Our aim is to be able generate a set of grasps on a previously unseen object that are compatible with a given task, and order them according to relevance.

In the learning process we observe a set of objects and grasps, placed by a human demonstrator, that are suitable to achieve the task. For each grasp on an object we extract a set of features, which capture global and local aspects, which we combine into a vector x. The features, x, captured from a demonstration are labeled with a task variable, t.

Global features describe the object as a whole, for example, how well it approximates a shape primitive, etc. Local features, are features at the grasp of the human demonstrator, for example, local curvature or color. We do not collect any pose information of the demonstrator, since robot-human pose transforms are not the aim of this paper. The demonstration procedure is illustrated in Fig. 3 and the features we capture are described in the feature section.

Given a novel object, and a desired task we wish to infer from a set of synthesized grasps, $G = \{g_i\}_{i=1}^{M}$, which of them are compatible with the task. For each synthesized grasp we extract a feature vector in the same manner as in the demonstration process, $X_G = \{x_i\}_{i=1}^{M}$; this is explained further in the Grasp Synthesis section. To select valid grasps from the synthesized set we apply k-Nearest Neighbor (k-NN) classification using the learned instances.

After extracting the set of valid grasps we want to produce a ranking, which reflects how similar a synthesized grasp is to learned instances. We use a Kernel Density Estimate (KDE) over the set of observed instances to produce a likelihood for each of the grasps. The KDE is formed using a Gaussian Kernel and the assumption that a suggested grasp, x^*, is conditionally independent of all other grasps except for the k nearest neighbors, that is,

$$p(x^*|\mathbf{X}_t) = \frac{1}{k} \sum_{x \in \mathbf{X}_k} \mathcal{N}(x^*; x, \mathbf{\Sigma}), \qquad (1)$$

where $\mathbf{X}_k$ is the set of k closest neighbors to x^* found in $\mathbf{X}_t$, the observed instances of the specific task, and $\mathbf{\Sigma}$ is a diagonal matrix. To execute the grasp on the object we choose the most likely synthesized grasp.

2.1 Metric Learning for Feature And Transfer-Selection

Both k-NN and the Gaussian Kernel uses a Euclidian metric to measure feature proximity. It has several shortcomings, as the dimension of the features increases it becomes susceptible to noise, and the inclusion of irrelevant features. Further on, different features have different natural distance measures, and it is unclear how one can weight these.

We instead consider the problem of measuring similarity as a representation learning problem, where we want to learn a task dependent embedding from the data such that similarity is translated into proximity in the embedding. In addition, we want the embedding to reflect which features are relevant to the task.

One set of metric learning algorithms, that approaches the problems specified above, are those that try to learn a Mahalanobis metric; that is, a positive semi-definite matrix (PSD), $\mathbf{M}$, that reflects information given about instance distances in the training data.

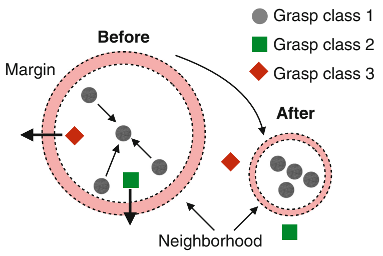


Fig. 2. Figuratively LMNN optimization decides on a set of neighbors for each instance and then tries to push those neighbors towards the instance, while pushing non-class members out of the perimeter of the neighborhood.

Since $\mathbf{M}$ is PSD the Mahalanobis metric is equivalent to applying a transform, $\mathbf{L}$ to the data points, where $\mathbf{M} = \mathbf{L}^T\mathbf{L}$, and then computing the Euclidian distance; that is,

$$D(x, y) = (x - y)^T \mathbf{M}(x - y) = \| \mathbf{L}(x - y) \|_2^2 \, . \tag{2}$$

The method we employ in this paper is the Large Margin Nearest Neighbor Classification (LMNN) algorithm [12]. [12] formulates the learning of $\mathbf{M}$ as a semidefinite program (SDP) which tries to compress distances between subsets of instances of the same class, called target neighborhoods, while pushing out non-class members, called impostors, outside a specified margin. See Fig. 2 for an illustration.

$$\begin{aligned}
&\textbf{minimize}\\
&(1-\mu)\Sigma_{i,i\rightsquigarrow j}(x_i - x_j)^T\mathbf{M}(x_i - x_j) + \mu\Sigma_{i,i\rightsquigarrow j,l}(1-y_{il})\xi_{ijl}\\
&\textbf{subject to}\\
&(x_i - x_l)^T\mathbf{M}(x_i - x_l) - (x_i - x_j)^T\mathbf{M}(x_i - x_j) \geq 1 - \xi_{ijl}\\
&\xi_{ijl} \geq 0\\
&\mathbf{M} \succeq 0
\end{aligned} \tag{3}$$

The first term in the objective function is the Mahalanobis distance which penalizes large distances between target neighbors. The second term is the slack variables that mimics the convex hinge loss. The indices i, j specify target neighbors, and l impostors. The slack variables, ξ_{ijl}, are over all triplets of target neighborhoods and measure the amount of violation of the margin in the transformed space.

The target neighbors subsets are determined by either a priori specified information about similarity or in the absence as the instances with minimum

Euclidian distance in the original space. The cardinality of the target neighborhoods is set manually.

M is learned by exploiting the fact that most target neighborhoods are homogenous allowing the optimization to work on subsets of the constraints involving the neighborhoods. **M** can thus be found using sub-gradients of the loss function with standard hill-climbing algorithms, together with a step where **M** is projected onto the cone of all positive semidefinite matrices [12].

Learning **M**, for D-dimensional data, requires learning D^2 entries of the matrix, the parameters of the model. With few data points that are high-dimensional overfitting can occur. Since data points are expensive to obtain we want to reduce the number of parameters. By letting **M** be diagonal, we only need to learn D parameters. Instead of penalizing any non-zero off-diagonal elements in **M** in the objective function we set all off-diagonal elements to zero, in the update step. The optimization process will thus find the diagonal matrix that minimizes the cost function.

LMNN translates neatly into our feature representation for grasps. The diagonalization of **M** allows us to treat the diagonal elements as feature weights, where low values indicates non-discriminative features and therefore irrelevant features for a task. The matrix found by LMNN thus implicitly facilitates feature selection. By analyzing the magnitudes for a specific task we can compare with our expectations of which features are important.

2.2 Features

We are interested in exploiting the fact that objects with similar function follow similar form. There is also evidence that humans make inference about the world by considering both global and local observations, [13], [14], and that both aspects enhance inference [15].

Given the above observations we are interested formulating features that capture both global and local aspects of the object. Since we a priori cannot know which of them will be important for a task it is reasonable to select many simple and broad features over those that are more discriminatory. In addition, having multiple features will help when sensors fails, or give incorrect information.

We capture information from a grasp by extracting features from a bounding box around the grasp and merging them with global descriptors of the object into a 104 dimensional vector. The process is illustrated in Fig. 3. We go over the features briefly below and provide motivation when needed.

[**Shape Primitives**] - Scoring function for how well the object resembles a spherical, cylindrical, and cuboid.
[**Elongatedness**] - The ratio of the minor axes to the major axis.
[**Relative Free volume**] - The volume of the part of the object enclosed by the grasp cuboid divided by the volume of the object.
[**Relative Grasping Position**] - The relative position of the grasp on the axes of the object, that is, the ratio of the position of the grasp on the main axes divided by the axes

[**Openings**] - The opening feature detects openings on objects, parametrized as a 3D circle, and produces: - A binary value for the existence of an opening. - The angle the grasp cuboid center makes with respect to the openings normal - The signed relative orthogonal distance to the openings plane.

[**Bag of Words Histogram over Fast Point Feature**] - To encode local curvature we create a Bag of Words model over the Fast Point Feature Histogram (FPFH) [16] using a small set of 30 keywords. For each of the points in a recorded grasp we extract count-normalized histogram (CNH) over the keywords.

[**Intensity and Gradient Filters**] - We apply gradient filters of first, second, and third order over the 2D window containing the object. We also extract the intensity values. For each of the points in a recorded grasp we compute CNHs for each of the filter values and the intensity values.

[**Color Quantization**] - We are interested in comparing color against color at the grasp point, as there are reoccurring color patterns in human design. We segment the image using the segmentation algorithm of [17] and encode each patch with its closest color from a chart of 15 different colors, in CIELab space. Each point in the point cloud of the object thus gets a quantized color assigned to it. For a grasp we compute the CNH of the quantized colors of the points captured by the grasp.

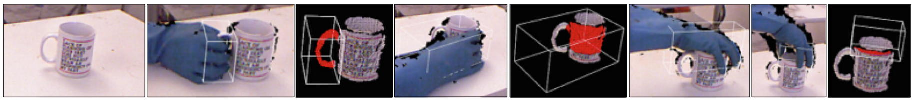
[**Color Uniformity**] - We are also interested in the uniformity of the colors since objects, usually have areas that have a single colors to indicate some form of function, for a grasp we compute the entropy, mean and variance of the CNH color quantization.

2.3 Grasp Synthesis

Generating constructive grasps on a partial point cloud of an object is difficult. Models that rely on collision detection are not applicable due to the partial representation of the object. Our method for generating grasps mimics the method found in [18] but instead uses the information found in the point cloud and point cloud normals. In short, we slice the object with a plane constructed by points in the point cloud, and select an approach vector that is orthogonal to the plane. The width of gripper is computed by taking the minimum point spread of the two orthogonal directions to the plane. The height of the gripper is set to a random value. The bounding box generated by the gripper is used to extract the feature representation for the synthesized grasp.

3 Experiments

In evaluating our approach, we want to verify a number of things: - That our method can order its nearest neighbors better than k-NN - That the ordering is reasonable - That the approach can separate different groupings of grasps reasonably well - That the features that LMMN picks out resonates with what could be expected for the specific task - That the grasps we generate are in fact useful, placed at the proper positions and executable



The object is placed in front of the robot, who examines it and records a set of global and local features from image and point cloud data.

A human demonstrates a set of grasps on the object to the robot. For each grasp the robot records a bounding box around the grasp and extracts the local features contained in the box. Each grasp is annotated for belonging to classes of tasks that the grasps is valid for. In the figure the first and second grasp are positive grasps for pouring, while the third is a negative example.

Fig. 3. Procedure for learning from demonstration.

3.1 Data Collection

We perform the collection of object-grasp data using a Kinect camera with objects placed on a flat surface in front of the robot. The robot observes and segments out the object and records features over the object. A human demonstrator performs both positive and negative grasp examples for four different tasks: *lift up, pouring, handle, and corks.* The robot observes the grasp placement and the local features as captured by a box around the hand placement. The procedure is illustrated in Fig. 3.

3.2 Learning

We preprocess the non-histogram features by scaling to unit variance. To evaluate the learning we run the LMNN and k-NN a 100 times, using random splits of the collected data with a ratio of 70 % training and 30 % test data. We also make sure that each split contains at least 25 % positive and negative examples. The reason for this lies in the formulation of LMNN that allows instances some slack in fulfilling the optimization constraints. The gradient descent optimization is thus prone to allowing the smaller class to violate the constraints while optimizing for the bigger class. If the positive example's relevant features have been erased by the feature learning process then comparison to previously learned instances in the grasp synthesis phase becomes reliant on comparison to irrelevant features.

We use 5 nearest neighbors to evaluate the classification rate. As can be seen in Fig. 4 LMNN is superior to k-NN performing better, roughly by 3–7 % for all grasps classes. The curves for the class agreement as a function of the nearest neighbor in Fig. 6, are also consistent for the LMNN while k-NN declines quite rapidly as the order increases. Further on, k-NN has much more variance. A possible explanation for the decline in the lift up task is that the features that explain a correct grasping position might be less distinct compared to handles or corks.

	Pouring	Lift Up	Handle	Cork
k-NN	[92.3%, 9.8]	[77.6%, 9.3]	[77.0%, 8.4]	[88.5%, 7.0]
LMNN	[95.2%, 5.5]	[80.5%, 10.2]	[84.5%, 7.6]	[95.5%, 3.9]

Fig. 4. k-NN and LMNN mean accuracy and standard deviation in classification of demonstrated grasp instances.

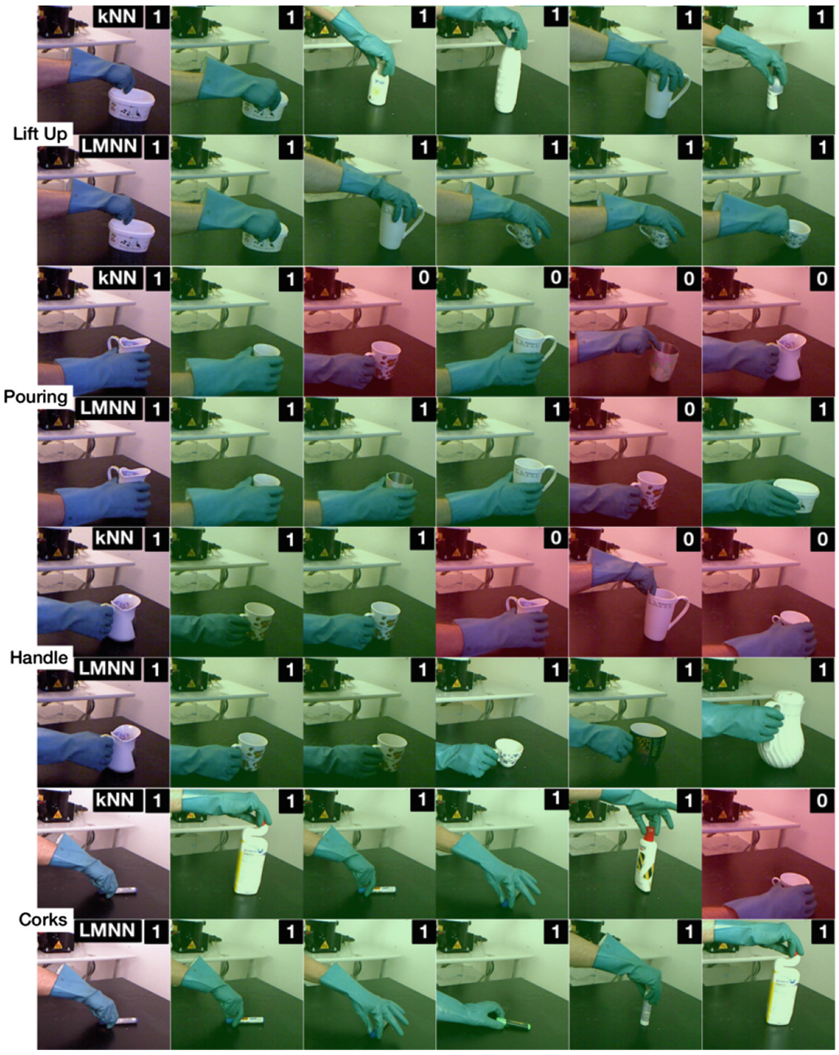
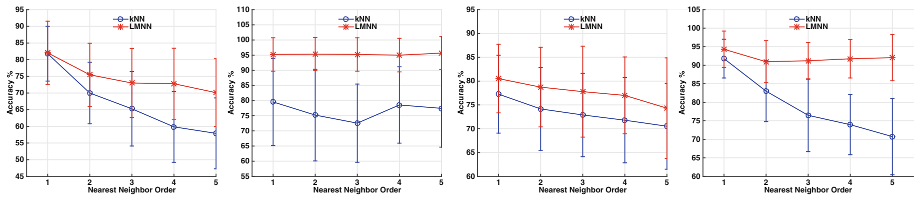


Fig. 5. 5 Nearest Neighbors to test instances for the four tasks for k-NN and LMNN. The number 0 and red overlay indicates an instance of another class than the test instance. The LMNN ordering is much more sensical than the k-NN. For example in the corks task, k-NN chooses the cork of a cleaning fluid bottle as the nearest neighbor while LMNN picks the cork of similar looking pen as the nearest neighbor. In the pouring task, where we have defined the handle grasps as not good for pouring for the robot's limited gripper, LMMN manages produce a much more sensical ordering than the k-NN which includes several handle grasps instances.



(a) Lift up (b) Pouring (c) Handle grasps (d) Corks

Fig. 6. k-NN versus LMNN accuracy if using only the k-th neighbor for classification. The low negative incline of the LMNN versus the k-NN indicates that learned transform improves consistency.

Quantitatively LMNN outperforms k-NN as could be expected. Less clear is if the embedding actually re-orders objects in an order that is sensical to a human observer. In Fig. 5 we have included a set of grasp instances and their 5 closest neighbors. As can be seen the ordering in the LMNN embedding makes more sense. For example, in the case of the pen cork k-NN has it that the nearest neighbor is the cork of a cleaning fluid bottle while LMNN puts a similar pen as the nearest neighbor. Similar orderings can be observed in the other examples and indicates that LMNN through the embedding finds a more sensical ordering.

3.3 Feature Selection

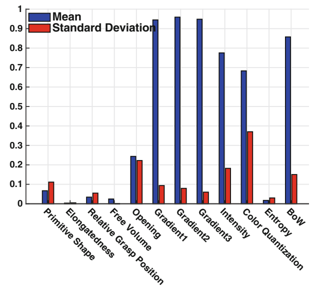
Feature selection in our context means that the LMNN diagonal have low to zero values for irrelevant features and high for relevant. One way of analyzing the selection is to look at the mean value and standard deviation across the 100 runs. A detailed analysis is provided in Fig. 7.

3.4 Grasping

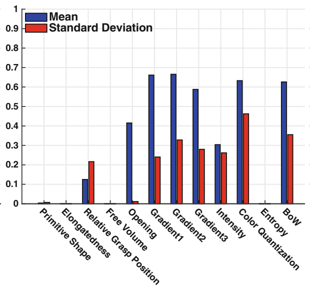
To evaluate how the model is able to transfer the demonstrated instances to grasp novel objects we select a set of unseen objects for each task and generate a 10000 positive samples using 1-NN for classification. We pick the most likely grasp as the one to perform on the object. The final grasps are illustrated in Fig. 1.

4 Conclusion

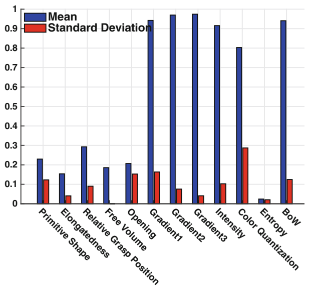
We have presented an approach for task based grasping based on metric learning. Our approach use demonstrations to learn a task-specific representation with the characteristic that distances reflect grasp similarity according to the demonstrated semantic. This allows us to use simple distance based heuristics to transfer grasps from objects in the training data-base to novel objects. We show experiments on a real robot where execution is performed on previously unseen objects according to the semantic learned during demonstration. In future work



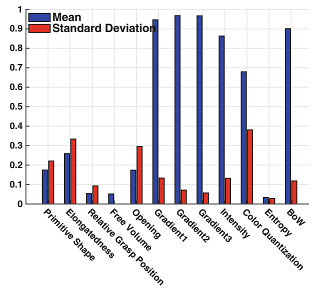
(a) Lift up



(b) Pouring



(c) Handle grasps



(d) Corks

Fig. 7. Mean values and standard deviation of the LMNN transformation matrix diagonal as per feature. A value close to zero means that the feature is considered irrelevant for separation by the LMNN. A value close to 1 and low standard deviation indicates that the feature is always present and of low relevance for separation by the LMNN. In (**a**), the lift up task, we expect that no feature is predominant since lift-up is more pose dependent than feature dependent. And as can be seen intensity, color quantization, and the BoW are the most relevant features. In (**b**), the pouring task, we would expect the opening to be the most relevant feature, and it has a value of around 0.4 with low standard deviation as could be expected. The BoW, gradient and intensity are all significant factors which can be explained by the similarity of objects that affords pouring. In (**c**), the handle grasp task, we would expect the shape to play a predominant role. However, as the diagram indicates the relative grasping position, color quantization and the opening features are also factors. In (**d**), the grasping of corks, the color quantization is an important factor as could be expected. Primitive shape and elongatedness are also factors which can be explained by the fact that most items with corks are elongated and have a more cylindrical shape.

we plan to extend the model to include information about how the human demonstrator grasped the object, such as approach vector, type of grasp, etc. Including this information will give the agent the ability to understand how the human demonstrator's grasp relates to its own embodiment.

References

1. Song, D., Huebner, K., Kyrki, V., Kragic, D.: Learning task constraints for robot grasping using graphical models. In: IROS (2010)
2. Song, D., Ek, C.H., Huebner, K., Kragic, D.: Embodiment-specific representation of robot grasping using graphical models and latent-space discretization. In: IROS, pp. 980–986 (2011)
3. Saxena, A., Driemeyer, J., Ng, A.Y.: Robotic grasping of novel objects using vision. Int. J. Robot. Res. **27**(2), 157–173 (2008)
4. Boularias, A., Kroemer, O., Peters, J.: Learning robot grasping from 3-D images with Markov Random Fields. In: IROS, pp. 1548–1553 (2011)
5. Detry, R., Ek, C.H., Madry, M., Kragic, D.: Learning a dictionary of prototypical grasp-predicting parts from grasping experience. In: ICRA (2011)
6. Herzog, A., Pastor, P., Kalakrishnan, M., Righetti, L., Asfour, T., Schaal, S.: Template-based learning of grasp selection. In: ICRA (2012)
7. Kroemer, O., Ugur, E., Oztop, E., Peters, J.: A kernel-based approach to direct action perception. In: ICRA, pp. 2605–2610 (2012)
8. Ying, L., Fu, J.L., Pollard, N.S.: Data-driven grasp synthesis using shape matching and task-based pruning. IEEE Trans. Visual Comput. Graphics **13**(4), 732–747 (2007)
9. Stark, M., Lies, P., Zillich, M., Wyatt, J.C., Schiele, B.: Functional object class detection based on learned affordance cues. In: Gasteratos, A., Vincze, M., Tsotsos, J.K. (eds.) ICVS 2008. LNCS, vol. 5008, pp. 435–444. Springer, Heidelberg (2008)
10. Aleotti, J., Caselli, S.: Part-based robot grasp planning from human demonstration. In: ICRA, pp. 4554–4560 (2011)
11. Hjelm, M., Detry, R., Ek, C.H., Kragic, D.: Cross-object grasp transfer. In: ICRA, Representations for Cross-task (2014)
12. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. **10**, 207–244 (2009)
13. Bertenthal, B.I.: Origins and early development of perception, action, and representation. Annu. Rev. Psychol. **47**(1), 431–459 (1996)
14. Berthier, N.E., Clifton, R.K., Gullapalli, V., McCall, D.D., Robin, D.J.: Visual information and object size in the control of reaching. J. Mot. Behav. **28**(3), 187–197 (1996)
15. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV, vol. 1, pp. 1–2. Prague (2004)
16. Rusu, R.B., Blodow, N., Beetz, M.: Fast Point Feature Histograms (FPFH) for 3D registration. In: ICRA, pp. 3212–3217 (2009)
17. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. Int. J. Comput. Vis. **59**(2), 167–181 (2004)
18. Bergström, N., Bohg, J., Kragic, D.: Integration of visual cues for robotic grasping. In: Fritz, M., Schiele, B., Piater, J.H. (eds.) ICVS 2009. LNCS, vol. 5815, pp. 245–254. Springer, Heidelberg (2009)