Features Set Refinement in Programming by Demonstration

Hoai Luu-Duc and Jun Miura
Department of Computer Science and Engineering
Toyohashi University of Technology
Email: hoai@aisl.cs.tut.ac.jp, jun.miura@tut.jp

***Abstract*—In transferring knowledge from human to robot using Programming by Demonstration, choosing features which can represent the instructor demonstrations is an essential part of robot learning. With a relevant set of features, the robot can not only have a better performance but also decrease the learning cost. In this work, the feature selection method is proposed to help the robot determine which subset of the features is relevant to represent a task in Programming by Demonstration framework. We implement an experiment system for human-robot interaction in simple task as proofing our concept as well as showing the preliminary results**

Keywords: Human-Robot Interaction, Programming by Demonstration, Feature Selection

I. INTRODUCTION

Transferring knowledge is defined as allowing the instructor to share or disseminate the knowledge or experiment to the learner. In robotic field, human-robot knowledge transfer which allows non-robotic experts to have opportunities to interact with robots is one of the areas that attracts attention of mamy researchers. There are two parts in human-robot knowledge tranfer. The first part is the human-to-robot part in which a robot learns knowledge from a human through human instruction and the other part is the robot-to-human part where a robot teaches the task to a human [1]. To transfer knowledge from an instructor to a learner, Learning from Demonstration (LfD) is one of the approaches that enables the instructor can able to share the knowledge to the leaner by demosntration a senquence of examples [2].

Using LfD to transfer knowledge from a human to a robot, the task in our work is defined as follows. The world is characterized by its state and a robotic task is specified its desired state (goal state). From a sequence of demonstrations, the robot has to observe the current states and selects the actions based on its observation to transfer the current state to the next state, and eventually reaches the final state and achieves the task. Based on this definition, if each state is well defined by a set of selected features, the robot can properly understand the task and easily select the appropriate actions. However, the number of possible features is usually large due to a complex nature of the real world, and therefore selecting an appropriate set of features is difficult from a limited of demonstration. As a result, the demonstration might include irrelevant ones which not only do not represent the state of the task but also lead to poor performance in transferring knowledge process. Therefore, in this paper, the feature selection approach is proposed to help the robot evaluate and select a relevant subset of features which can represent the task in learning process from the human to the robot.

Feature Selection in robotics has been applied to several problems. Deuk et al. [3] used a feature selection to solve a mobile robot navigation problem. Loscalzo et al. [4] developed a feature selection method for a genetic policy search. Kim et at. [5] applied a feature selection to a rescue robot to classify the smoke and the fire. Bullard et al. [6] enabled the robot to interact with human requiring the features information. However, these works assume that demonstrations or examples are provided completely while demonstrations in LfD are limited and depend on the instructor. If the instructor can provide a good demonstration subset, the robot can have a good performance with a small set of demonstrations. In contrast, with a bad demonstration subset, the robot not only needs more demonstration to achieve the task but also has a low performance in executing the task. Feature selection is thus one of the promising approaches to cope with a limited amount of demonstrations by generating a relevant feature subset.

In this work, we propose a feature selection method to generate a promising subset of features incrementally after each demonstration. The advantage of our approach is that we do not need to wait for a completed set of demonstrations. The proposed method generates a promising subset after each demonstration then the robot will use that promising subset and the current demonstrations set to create the model of the task and execute it under the instructor observation. The main contribution of the paper is that instead of considering if the demonstration which are given by instructor is enough or not to apply the feature selection, the proposed feature selection approach can generate an appropriate set of features subset for the demonstrations so far. Moreover, we want the process of human-to-robot transferring knowledge is more likely human-to-human transferring knowledge.

The rest of the paper is organized as follows. Section 2 presents the learning mechanism in which the feature selection method is proposed to help the robot refine the relevant subset of features. Section 3 shows the experimental results and Section 4 presents concluding remarks and future work.

II. LEARNING TASK MODEL

A. Problem Statement

The problem is to determine a promising subset of features which is used for describing the task. To define the state, the robot is given a set of demonstrations with the binary label, L, and the list of all candidate features, F, among which the robot extracts promising ones based on the demonstration. The goal of the robot is to define which subset of the features, F', represents the set of demonstrations and then is used to classify the demonstration to create the task model. In this research, we choose a simple task which has only one action to teach the robot. In other words, a task is represented by the target state to achieve. We assume that the robot already has a planning and execution system that can transfer the one state to the another state.

B. Problem Domain

In this work, we give the robot a pick-and-place task. The instructor demonstrates a task by putting an object to an appropriate place under specific rules. For example, the color of the object must be the same with that of the place. The robot can observe three types of attributes (color, shape and size) of objects and places. The label of demonstration is a binary label; zero means a false state and one means a true state. There are 30 features (shown in Table I) that can be observed by the robot. Redundant features are added to increase the complexity and the noise of the learning task. The robot's goal is to determine which subset of features is relevant to the demonstration subset.

TABLE I: The list of features in Simulation

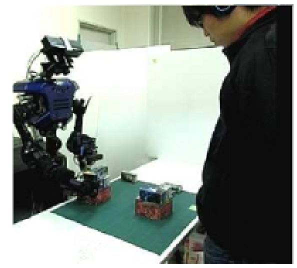
Object	Features	Values
Blocks	background color	red, green
	bounding color	red, green
	outside shape	rectangle, ellipse
	inside shape	rectangle, ellipse
	size	1,2
	redundant features [f1-f10]	0,1
Place	background color	red, green
	bounding color	red, green
	outside shape	rectangle, ellipse
	inside shape	rectangle, ellipse
	size	10
	redundant features [f1-f10]	0,1

C. Learning from Demonstration Process

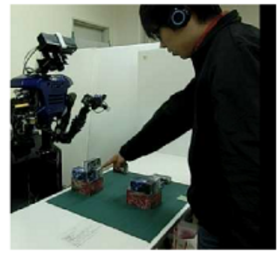
The Learning from Demonstration is conducted by the following steps:

1) Demonstration by instructor: the instructor demonstrates the task by picking an object then putting it into an appropriate location. The definition of the task is given to the instructor before teaching. For example, the blue object must be put into the red place as shown in Fig. 1a.
2) Observation and feature selection by the robot: the robot observes the demonstration, then extracts all information from demonstration (i.e, the features and their values). Then, the feature selection algorithm is applied to calculate which subset of features is relevant to the task and the robot chooses the highest relevant subset of feature as a model.

(a) Instructor demonstration



(b) Robot execution



(c) Instructor judgment

Fig. 1: Programing by Demonstation Framework

3) Task execution by the robot: using the model in step 2, the environment of the task is refreshed then the robot executes the task under the supervision of the instructor as shown in Fig. 1b.
4) Judgment: after observing the robot's execution, the instructor judges if the action of the robot is correct as shown in Fig. 1c. If the robot fail to execute the task correctly, the new demonstration will be presented, go to step 1. If the robot success in the task, the learning process will finish.

D. Learning Framework

Figure 2 shows the learning process. After demonstrating the first demonstration, the Sequential Forward Selection (SFS) algorithm [7] will be executed to generate a promising subset of features, then the model is created based on the subset and the current demonstration set. In our work, the ID3 Decision Tree algorithm [8] is used as the classifier. After the creating model step, the training accuracy is calculated to determine if the current promising features subset can represent the current demonstration set. If the training accuracy is equal to 100 percent, the Sequential Backward Selection (SBS) algoritm [7] is applied to remove irrelevant features and the final task model is created and the robot wait for the next demonstration.

In the second demonstration, after observing the instructor demonstration, the robot will calculate the testing accuracy of the final model in the last demonstration with the new demonstration. If the testing data high enough (in this case, it equal to 100 percent, the SBS is applied to remove irrelevant features . If not, the SFS will add more features into the previous promising features subset and create a new model of the task. The learning process will finish when both the testing accuracy and the training accuracy are equal to 100 percent.

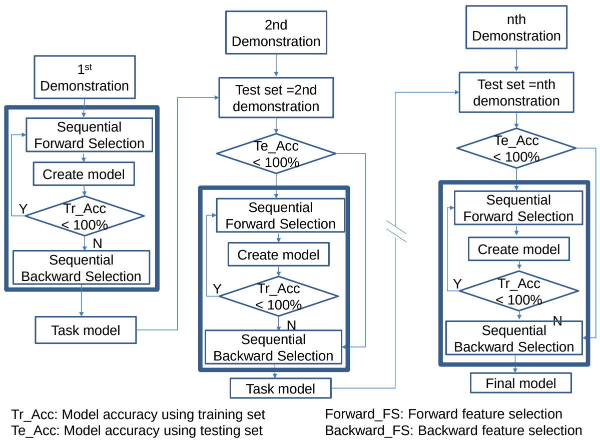


Fig. 2: Learning Framework using Programming by Demonstration

1) Sequential Forward Selection: To generate a promising features subset, there are the following three main methods in feature selection: filter method, wrapper method and embedded method. Bullard [6] had shown that the filter method have more efficiency in generating the promising features subset in terms of the computation cost and accuracy. So we use filter method to create the promising features subset. The criteria which is applied to choosing the relevant feature are Mutual Information (MI) and Conditional Mutual Information. The algorithms of SFS is shown in algorithm 1.

Algorithm 1 The Sequential Forward Selection algorithm

Input: Demonstration
Output: Promising Feature Subset S
extract $featureslist$ and data from demonstration
$listMI = NULL$
if S == $NULL$ **then**
    for $feature$ in $features$ $list$ **do**
        $listMI$.append(calculate MI($feature$,$label$))
    end
    S.append($feature$ which have highest score)
else
    features which are in S are removed from $featurelist$
    for $feature$ in $features$ $list$ **do**
        $listMI$.append(the conditional MI($feature$,S[last appended feature]))
    end
    S.append($feature$ which have highest score)
end
return S

2) Creating Task Model: After finishing Forward Feature Selection, the promising feature subset and the current demonstration set is used for training and creating the task model. After training process, the training accuracy is calculated to determine if the current features subset can present the latest demonstration set. If it is not, more features will be added into the current subset by SFS.

3) Sequential Backward Selection: If the task model which is built based on the promising features subset and the latest demonstration set pass the testing accuracy criteria, the SBS module is executed to remove irrelevant features in the subset. The main purpose of this module is that the demonstration is limited while the number of features in the scene is very large, and that lead to the irrelevant features might join into the promising features subset. To remove irrelevant features, Symmetrical Uncertainty [9] is used to estimate the redundancy value of each feature in the current subset.

The SBS algorithm is shown in algorithm 2.

Algorithm 2 The Sequential Backward Selection algorithm

Input: Demonstration, Promising Features Subset S
Output: Removing Feature Subset Sr
for $feature_i$ in S **do**
    calculate SUi,c for $feature_i$
end
sort S in descending SUi,c value
for $feature_j$ in S **do**
    for $feature_i$ $(i <> j)$ in S **do**
        calculate SUi,j for $feature_j$ and $feature_i$
        if $SUi,j > SUi,c$ **then**
            Sr.append($feature_i$)
            S.remove($feature_i$)
        end
    end
end
return Sr

After getting the list of irrelevant features, each feature in the list will be removed if it does not change the training accuracy. The training accuracy is used in this step because the demonstration is limited and the final model is required to have 100 percent in training accuracy. The final task model is the model that have highest training accuracy and lowest number of features in the promising feature subset.

III. EXPERIMENT RESULTS

In our experiment, the task that the robot needs to learn is the matching task, for example, the background color of the object must be the same with the background color of the place. In this case, the rule of matching depends on the instructor. The goal of the robot is to find the features subset which can represent the rule of the task and execute it correctly. To test the proposed method, a simple simulator is created to let the instructor demonstrate the task as shown in Fig. 3

In this simulator, the places that the block must be put are on the left side while the blocks are on the right side. To analyze easily the subset of features, we use only two type of color (red and green) and two type of shape (rectangle and ellipse) in this experiment. The instructor uses mouse or joystick to move a block to an appropriate location in demonstrating the task under the specific rule. After moving all the blocks to its location, the instructor uses

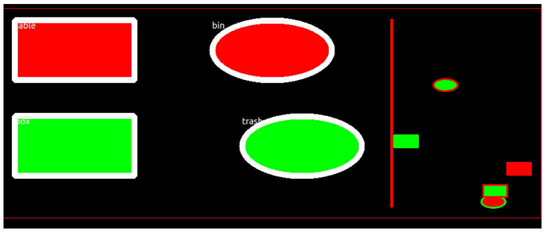


Fig. 3: The simulator for Task Learning

the finish button to inform the robot that the demonstration has finished. The blocks in each demonstration are randomly created and the instructor may not be able to provide enough instances in one demonstration. In each demonstration, five blocks are created to be put in the corresponding place. Moreover, when a block is put in an appropriate place, one positive data and three negative data are created. The positive data is the matching between this block and its place while negative data are the matching between this block and the others place. In this task, there are totally 16 instances which are necessary to represent the goal state. After each demonstration, the robot will show the promising features subset to the instructor and the learning process will finish when the promising feature subset is the same with the instructor rule or the demonstration set is enough.

Table II shows some results of learning a matching task. In this table, we implement two type of demonstrations. One is random demonstration generation in which the instructor just demonstrate random demonstrations that are generated by the simulator. The other is deliberate or careful demonstration demonstration where the instructor carefully choose demonstrations. In the random demonstration generation, the robot failed in choosing the promising features subset because the robot use the greedy search based on the Mutual Information and the conditional Mutual Information to choose the relevance features, so that with a limited of demonstrations, there are a variety of results that suitable with the current demonstration set. In this case, the instructor stop the learning process after 24 demonstrations because the demonstration set is completed. In the deliberate or careful demonstration learning, the instructor deliberately chooses the demonstrations which can represent the task easily without duplication. In this case, the robot can reach the true features set after 4 demonstrations and execute the task correctly.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, a feature selection method to help the robot achieve the task is proposed. By adding demonstration and using a feature selection during the learning process, the robot can choose the relevant features and execute the task correctly. However, the promising feature subset is depend on the demonstration set, that is if the robot has a good demonstration subset, the robot can get the promising feature subset easily, and otherwise, the robot might take a long time in learning refining the features subset.

Using only feature selection method is not enough to refine the features subset which represent the task, so as future work the human-robot interaction must be considered to improve the accuracy of the system. For example, the robot may ask the instructor about the features information or acquire the demonstration from the instructor during the learning process.

TABLE II: Experiment Results

Learning Type and Number of Demonstrations	True Features Set	The Promising Feature Set
Random generation demonstration (24 demonstrations)	Object background color Object outside shape Place background color Place outside shape	Object background color Object inside shape Place f2 Place inside shape
Deliberate demonstration (4 demonstrations)	Object background color Object outside shape Place background color Place outside shape	Object background color Object outside shape Place background color Place outside shape

REFERENCES

[1] K. Yamada, J. Miura. "Ambiguity-Driven Interaction in Robot-to-Human Teaching" *Proceedings of the Fourth International Conference on Human Agent Interaction.* ACM, 2016.

[2] B.D. Argall,et al., "A survey of robot learning from demonstration", *Robotics and Autonomous Systems (2009)*, doi:10.1016/j.robot.2008.10.024

[3] C. Diuk, L. Li, and B. R. Leffler, "The adaptive k-meteorologists problem and its application to structure learning and feature selection in reinforcement learning," *International Conference on Machine Learning.* ACM, 2009, pp. 249–256.

[4] S. Loscalzo, R. Wright, and L. Yu, "Predictive feature selection for genetic policy search," *Autonomous Agents and Multi-Agent Systems,* vol. 29, no. 5, pp. 754–786, 2015

[5] J.-H. Kim, S. Jo, and B. Y. Lattimer, "Feature selection for intelligent firefighting robot classification of fire, smoke, and thermal reflections using thermal infrared images," *Journal of Sensors,* vol. 2016, 2016.

[6] K. Bullard, S. Chernova1, and A.L. Thomaz. "Human-Driven Feature Selection for a Robot Learning Classification Tasks from Demonstration" *Robotics and Automation (ICRA), 2018 IEEE International Conference on. IEEE* ,2018.

[7] L. Yu, and H. Liu, "Efficient feature selection via analysis of relevance and redundancy." *Journal of machine learning research 5.Oct (2004).*

[8] J. R. Quinlan, "Induction of Decision Trees. " Mar. 1986

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery,"Numerical Recipes in C." *Cambridge University Press, Cambridge,* 1988.