Modeling Manipulation Interactions by Hidden Markov Models

Koichi Ogawara, Jun Takamatsu, Hiroshi Kimura[†], Katsushi Ikeuchi

Institute of Industrial Science, The Univ. of Tokyo
Tokyo, 153-8505, Japan
{ogawara, j-taka, ki}@cvl.iis.u-tokyo.ac.jp

[†]*Univ. of Electro-Communications*
Tokyo, 182-8585, Japan
hiroshi@kimura.is.uec.ac.jp

Abstract

This paper describes a new approach on how to teach everyday manipulation tasks to a robot under the “Learning from Observation” framework. In our previous work, to acquire low-level action primitives of a task automatically, we proposed a technique to estimate essential interactions to complete a task by integrating multiple observations of similar demonstrations. But after many demonstrations are performed, there are possibly interactions which are the same in nature. These identical interactions should be grouped so that each action primitive becomes unique. For this purpose, a Hidden Markov Model based clustering algorithm is presented which automatically determines the number of the independent interactions. We also show that obtained interactions can be used as discriminators of human behavior. Finally, a simulation result and an experimental result in which a real humanoid robot learns and recognizes essential actions by observing demonstrations are presented.

1 Introduction

At the present time, applicable fields of robots are extended to the environment where human beings co-exist and those robots are expected to closely support, substitute for and help us. Everyday tasks are their target tasks and the actual tasks are dependent on the purpose of the user; thus we can’t design the detailed behavior of the robot in advance as in the case of industrial robots, but the end user must have a way of expanding its ability.

That means ordinary people, not professionals, teach the robot a behavior and, because of that, a user-friendly instruction method is required. The easiest and desirable way of teaching is to just demonstrate the behavior so that the robot automatically learns it from observation and builds an abstract representation of the task, i.e. a Task Model, as we usually do. This framework is called “Learning from Observation” and has been actively studied[1, 2, 3, 4]. In these researches, a typical approach is to divide

a demonstrated task into a series of primitive representations, each of which is one of the finite set of possible phenomena appeared in that task domain. In most cases, those primitives themselves are designed manually in advance.

However, in typical manipulation tasks, several problems arise, for example, (1) it is hard to determine when and which objects are being interacted with, especially in cluttered environment, and (2) it is hard to determine which observed interactions are essential to the task without prior task-dependent knowledge which should be obtained from observation. Further, (3) it is hard to design all the primitive actions for everyday tasks in advance.

To solve these ambiguity simultaneously, we proposed a method to integrate multiple demonstrations which differ in representation but are the same in nature and to extract shared interactions among all the demonstrations as essential interactions[5]. These essential interactions are considered to be primitive actions which can be used as building blocks to represent a task model. However, after many demonstrations are performed, the same behavior may be represented in different primitives.

These interactions must be treated as a same one so that they constitute an independent primitive which represents a unique symbol corresponding to that action. For this reason, the collected essential interactions must be clustered into independent primitive groups. Each essential interaction is made up of 3 properties: the grasped object, the target object and the set of relative trajectories[5]. First of all, the entire set of interactions can be easily divided by vision into sub-groups each of which holds the same object pairs: the grasped object and the target object[5]. So, the remaining problem is how to divide each sub-group into independent clusters each of which shares the same type of relative trajectories.

To integrate trajectories without prior task-dependent knowledge, a probabilistic model-based approach based on Hidden Markov Model is pro-

posed in section 2. Hidden Markov Model is gaining a great deal of attention in the field of motion understanding and is used, for example, to learn and generate walking patterns of a humanoid robot[6] or to recognize assembly tasks[4]. However the number of Hidden Markov Models is typically determined manually and, for that reason, it is difficult to apply Hidden Markov Model to analyze everyday tasks whose variety is not known in advance. Our approach automatically determines the appropriate number of clusters and separates the set of trajectories into independent primitives. These primitives can also be used as discriminators of human behavior to build a task model. A method to generate and utilize discriminators based on Hidden Markov Models is presented in Section 3. Finally, in Section 4, these techniques are examined in simulation and are applied to a humanoid robot which acquires primitive actions and recognizes human behavior.

2 Clustering primitive actions

From the result of estimated essential interactions[5], a set of relative trajectories S_i are obtained; each of which is a discrete and variable-length sequence of the grasped object's position (3 dimensions) and orientation (4 dimensions, quaternion representation) in the target object's coordinate frame. Even if the 2 trajectories belong to the same primitive action, there is spatiotemporal difference to some extent between them.

In this paper, we utilize Hidden Markov Models (HMMs) to represent the trajectories. This is because HMM is a robust stochastic framework to model any time-series and also it can be used as a powerful discriminator of any time-series.

HMMs model doubly stochastic processes, that are first-order Markov processes whose internal states are not directly observable and, thus, the term "hidden" is used. The observable output signal depends on probability distributions, fixed for each internal state. Since the model can disregard noise through a stochastic framework, it allows us to deal with the highly stochastic underlying structure of the process[7].

In this section, first, we consider the case where K , the number of clusters, is known and describe a method of clustering a set of trajectories based on HMMs. Then, we discuss how to determine the appropriate K .

2.1 Clustering into K groups

First, a HMM M_j is assigned to each trajectory S_i and its parameters are fitted to that trajectory. Then, for each trained HMM M_j , the log-likelihood

$L(S_i|M_j)$ of each of the N trajectories is calculated.

To divide N trajectories into K groups, we use "symmetrized distance" $L_{ij} = \frac{1}{2}(L(S_i|M_j) + L(S_j|M_i))$ as a measure of similarity between HMMs M_i and M_j [8]. Then hierarchical clustering with farthest-neighbor algorithm is applied based on this similarity measure[9]. As a hierarchical clustering, we take an agglomerative approach, and a pair of clusters whose similarity is the highest is searched for and merged until the specified number of clusters remain. Farthest-neighbor algorithm is a method to adopt the worst measure among the possible combinations of 2 trajectories when calculating similarity between 2 clusters, and it tends to generate compact clusters.

2.2 Determining the number of clusters

To determine the number of clusters, an evaluation function to measure the goodness of match of being a specified number of clusters is needed.

The posterior probability $P(K|S)$ of being K given the set of trajectories S increases monotonously as the number of clusters increases. So, in general, curvature of posterior probability or Bayesian Information Criterion (BIC) is employed as a goodness of match. The significant change in curvature of the successive posterior probabilities indicates the appropriate number of clusters. With regard to BIC, a penalty term is introduced in conjunction with EM algorithm to suppress the goodness of match when the number of clusters get bigger[10]. In this paper, we adopt a criterion using curvature.

We already know each posterior probability $P(S_j|M_i)$ from Baum-Welch estimation algorithm and the approximation of the posterior probability $P(K|S)$ can be estimated as described below.

We assume the number of clusters K_i are equally likely a priori, and calculate the posterior probability $P(K|S)$ by Bayesian rules as shown below (Eq.(1)).

$$\begin{aligned} P(K_i|S) &= \frac{P(S|K_i)p(K_i)}{\sum_{i=1}^N P(S|K_i)p(K_i)} \\ &= \frac{P(S|K_i)}{\sum_{i=1}^N P(S|K_i)} \end{aligned} \quad (1)$$

$$\begin{aligned} P(S|K_i) &= N! \cdot \prod_{j=1}^N P(S_j|K_i) \\ &= b \cdot \prod_{j=1}^N P(S_j|K_i) \end{aligned} \quad (2)$$

$$\begin{aligned} P(S_j|K_i) &= \sum_{t=1}^{K_i} P(S_j|M_t)P(M_t|K_i) \\ &\approx P(S_j|M_{T_j})P(M_{T_j}|K_i) \end{aligned} \quad (3)$$

$$P(M_t|K_i) = \frac{\text{Num. of traj.}(M_t)}{N} \quad (4)$$

Figure 1: Primitive actions.

where a and b are constant proportionality coefficients. In Eq.(3), only the highest probability is taken into account because, in usual case, the other probabilities are significantly smaller than the above one. M_t is one of K_i reconstructed HMMs which are clustered by using the method described in the previous section. T_j is the model number which outputs the highest posterior probability on S_j ($T_j = \operatorname{argmax}_t \{P(S_j|M_t)\}$). $P(M_t|K_i)$ is determined as the ratio of the number of training samples clustered in HMM M_t .

Finally, the log posterior probability $L(K_i|\mathbf{S})$ is induced as shown in Eq.(5).

$$L(K_i|\mathbf{S}) \approx c \cdot \sum_{j=1}^N \left(L(S_j|M_{T_j}) + \log \frac{\text{Num. of traj.}(M_{T_j})}{N} \right) \quad (5)$$

where $c = \log(ab)$.

To determine K_i , the approximation log posterior probability for each possible K_i is calculated, and $K_{i'}$ corresponding to the strongest curvature is selected to be a natural number of the total clusters. Until $K_{i'}$, log posterior probability tends to increase rapidly, and after that, increasing speed gets dramatically slower. In this paper, the curvature which has the highest acceleration value among the successive 5 log posterior probabilities is chosen.

3 Recognition with primitive actions

In the previous section, the best number of clusters (primitives) is estimated and all the trajectories are clustered into one of them. So, we have a list of primitive actions composed of (1) the grasped object, (2) the target object and (3) the relative trajectory represented in the form of HMM as shown in the left part of Figure 1.

Each primitive action can be used to recognize and symbolize human behavior. Suppose the system observes the human behavior as shown in the top right of Figure 1. There are roughly 3 possible phenomena. (1) There is interaction between the grasped object and the right side object. (2) There is interaction between the grasped object and the left side object. (3) There is no meaningful interaction. To determine which phenomenon is happening, all the possible primitive actions are evaluated for each interaction candidate. In the case of (1), the system has 2 possible primitive actions in the list, the top and the second one, so these 2 HMMs are applied to the corresponding relative trajectory S' obtained by vision. After all the possible primitive actions are estimated, the system selects M_j of the highest log probability $\max_j L(S'|M_j)$ and symbolizes the scene by M_j .

However, it is possible that none of the primitives in the list doesn't describe the observed behavior, i.e. the observed behavior is not learned yet. The above criterion always outputs a primitive, so we have to check the appropriateness of the selected primitive.

Figure 2 shows the distribution of the log posterior

probability of the training samples when applied to the corresponding HMM. There are 6 primitive actions, i.e. HMMs, and all the samples corresponding to a same primitive output relatively similar value, so we roughly estimate the range of each distribution by Gaussian distribution. Because the total number of samples for each distribution is small, the boundary of the distribution is set within $\pm 6\sigma$. Each distribution is plotted as a Gaussian curve fitted to the samples. If the log posterior probability of the selected HMM is below the corresponding distribution, the selected HMM is rejected and the system concludes that no primitive action is appropriate.

Figure 2: Log probability distribution of the training samples.

4 Experiment

We expanded the Hidden Markov Model Toolkit (HTK)[11] and implemented our algorithm on top of that.

First, we apply our proposed method to the trajectories generated by a computer to verify our clustering algorithm. Then we apply our method to the trajectories obtained from human demonstrations[5]. These trajectories are grouped into independent clusters and become building blocks to recognize and describe human behavior. Finally, we show recognition and imitation result of demonstrations by using a humanoid robot.

4.1 Simulation result

We generate sample trajectories in 2D space and apply our algorithm to the data. We prepare 4 independent straight line trajectories and generate 5 variants for each by adding Gaussian noise ($\sigma = 0.2$) on them. In total, we get 20 trajectories which should be best clustered into 4 groups. Figure 3 shows 4 trajectories each of which are chosen from a different group.

Traj. 1 line (0,0) - (1,0), Traj. 2 line (0,0) - (1,1), Traj. 3 line (0,1) - (1,0), Traj. 4 line (0,1) - (1,1). Each trajectory is made of 100 sample points. Each sample point is added by Gaussian noise.

Figure 3: Generated trajectories in simulation.

A 5 state left-to-right HMM is assigned to each trajectory $S_i(t), 1 \leq t \leq T_i$ (as shown in the bottom right of Figure 1). Each HMM is trained using only one sample. The minimum Gaussian variance of each state is limited to be greater than 0.2 to avoid over-fitting.

Table 1: Result of K clustering.

Number of clusters			
2	3	4	5
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14			
15 16 17 18 19	5 6 7 8 9 10 11 12 13 14	5 6 7 8 9	5 6 8 9
	15 16 17 18 19	10 11 12 13 14	7
		15 16 17 18	10 11 12 13 14
			15 16 17 18 19

Table 1 shows the result of clustering for $K_i, 2 \leq i \leq 5$. For each number of clusters K_i , the log posterior probability $L(K_i|\mathbf{S})$ is calculated. Table 2 shows the total amount of the log posterior probability $L(K_4|\mathbf{S})$.

Table 3 shows the calculated approximation log posterior probabilities for the number of the clus-

Table 2: Result of the case of 4-clusters.

S_i	Log probability	
0	$L(S_i M_0)$	10.00
1		15.83
2		-1.85
3		16.99
4		-2.38
5	$L(S_i M_1)$	13.59
6		-7.12
7		-1.07
8		-0.84
9		10.13
10	$L(S_i M_2)$	3.54
11		-5.27
12		-10.70
13		12.70
14		1.62
15	$L(S_i M_3)$	20.12
16		6.73
17		10.25
18		-4.00
19		3.93
Total	$L(K_4 S)$	92.21

Table 3: Log posterior probability.

Num. of clusters	Log posterior probability
1	-1302
2	-884
3	-350
4	92
5	104
6	114

ters ranging from 1 to 6. The values increases monotonously as the number increases, but there's a significant curvature around the number 4. Figure 4 represents normalized minus log probabilities and shows this fact clearly.

4.2 Clustering of pouring motion

In this section, the proposed method was applied to the data which were obtained from multiple observations of human demonstrations in the previous work[5]. Figure 5 shows the same essential interactions, pouring the content of the cylindrical container with grip into the box, obtained from different demonstrations. There are 15 corresponding trajectories and 3 of them are appeared in Figure 5. Among them, the trajectories from No.1 to No.5 belong to one primitive action and the remaining 10 trajectories belong to another primitive action.

Figure 4: Normalized log posterior probability.

Figure 5: Demonstrated trajectories.

As described in the previous section, 5 state left-to-right HMM is assigned to each trajectory and the calculated approximation log posterior probabilities corresponding to the number of the clusters ranging from 1 to 6 is shown in Table 4. The value increases monotonously as in the previous case, and a significant curvature appears around the number 2 in this case. Figure 6 represents normalized minus log probabilities.

Table 4: Log posterior probability.

Num. of clusters	Log posterior probability
1	-56269
2	-36095
3	-35047
4	-34290
5	-34121
6	-33473

4.3 Recognition

The top half of Figure 7 shows some scenes from a newly demonstrated task. This behavior is recognized using the technique described in the previous section and the fourth primitive action in Figure 1 is detected.

Figure 6: Normalized log posterior probability.

Figure 7: Recognition and reproduction of human behavior using primitive actions.

The bottom half of Figure 7 shows the reproduced task by the robot. The robot-arm is controlled to maintain the same relative trajectory during each essential interaction. The reference relative trajectory is the mean trajectories calculated in [5]. Any tasks can be interpreted as a sequence of primitive actions. Actions which are not learned beforehand are just ignored. The system can understand whether an action in a scene is known or not, and can add a new primitive action at anytime.

5 Conclusion

In this paper, we discuss that essential interactions detected in the previous work can be used as building blocks, primitive action, to represent a task model and also be used as discriminators to symbolize human demonstrations.

To be a building block, each primitive action is unique among others and the primitives which are the same in nature must be grouped. Each primitive is composed of information about the interrelated objects and their relative trajectory, and to divide the collected interactions into independent primitives, a HMM based clustering algorithm is proposed which

automatically determines the number of clusters. We also show that the obtained primitives made of HMM can be used to recognize human behavior.

Finally, an experimental result is presented in which a humanoid robot learns primitive actions and recognizes human demonstrations.

Acknowledgments

This work is supported in part by the Japan Science and Technology Corporation (JST) under the Ikeuchi CREST project, and in part by the Grant-in-Aid for Scientific Research on Priority Areas (C) 13224025 of the Ministry of Education, Culture, Sports, Science and Technology.

References

- [1] K. Ikeuchi and T. Suehiro. Toward an assembly plan from observation part i: Task recognition with polyhedral objects. *IEEE Trans. Robotics and Automation*, 10(3):368–384, 1994.
- [2] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching. *IEEE Trans. Robotics and Automation*, 10(6):799–822, 1994.
- [3] K. Ogawara, S. Iba, T. Tanuki, H. Kimura, and K. Ikeuchi. Recognition of human task by attention point analysis. In *Int. Conference on Intelligent Robot and Systems*, volume 3, pages 2121–2126, 2000.
- [4] G. E. Hovland, P. Sikka, and B. J. McCarragher. Skill acquisition from human demonstration using a hidden markov model. In *Int. conf. on Intelligent Robots and Systems*, pages 2706–2711, 1996.
- [5] K. Ogawara, J. Takamatsu, H. Kimura, and K. Ikeuchi. Generation of a task model by integrating multiple observations of human demonstrations acquiring hand-action models by attention point analysis. In *Int. Conference on Robotics and Automation*, pages 1545–1550, 2002.
- [6] T. Inamura, Y. Nakamura, H. Ezaki, and I. Toshima. Imitation and primitive symbol acquisition of humanoids by the integrated mimesis loop. In *Int. conf. on Robotics and Automation*, pages 4208–4213, 2001.
- [7] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSP MAGAZINE*, pages 4–16, 1986.
- [8] Juang B. H. and L. R. Rabiner. A probabilistic distance measure for hidden markov modeling. *AT&T Technical Journal*, 64(2), 1985.
- [9] P. Smyth. Clustering sequences with hidden markov models. *Advances in Neural Information Processing*, (9), 1997.
- [10] Chris Fraley and Adrian E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.
- [11] S. J. Young. *HTK: Hidden Markov Model Toolkit V2.2*. Entropic Research Lab Inc., Washington DC, 1999.