

On the Adjustment of the Weights of the Levenshtein Distance for the Description of Sequences of Rhythmic Motor Primitives

Anna Gorbenko

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia

Vladimir Popov

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia

Copyright © 2015 Anna Gorbenko and Vladimir Popov. This article is distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Recently, some modifications of the approximate period problem was considered as models for the investigation of sequences of rhythmic motor primitives. Some weighted Levenshtein distance is needed for such models. To obtain efficient models we need to find a proper adjustment of the weights of the Levenshtein distance. In this paper, we consider an approach to proper adjustment of the weights.

Subject Classification: 68T40

Keywords: symbolic trajectory description, rhythmic motor primitives, robot navigation, approximate period problem, Levenshtein distance

Problems of the task-level robot learning from demonstration has received substantial attention recently (see e.g. [1]). Among other, we can mention that learning from demonstration methods are very important for humanoid robots (see e.g. [2, 3]). Also, it should be noted that learning from demonstration is extensively studied for mobile robots (see e.g. [4, 5]). Motor skills can be represented by motor primitives. Note that motor tasks can be divided into two major groups: discrete motor tasks and rhythmic motor tasks [6]. It should be noted that rhythmic motor tasks are very important. There are many different rhythmic motor tasks that could be learned [6]. Recently, the approximate period problem was considered for the investigation of sequences of motor primitives [7, 8]. It should be noted that some weighted Levenshtein distance is needed for the approximate period models. To obtain efficient models we need to find a proper adjustment of the weights of the Levenshtein distance. In this paper, we consider an approach to proper adjustment of the weights.

In general, we can assume that each trajectory is performed in the tele-operated mode. Some sequences of motor commands can be considered as symbols. During teleoperation, the robot is operated by the teacher. It is clear that the robot can records a sequence of commands that obtained from the teacher. However, the recorded and really performed command sequences can differ. For instance, if the recorded sequence of commands is equal to

Forward 2 Right 90

then really performed command sequence can be equal to

Forward 2 Right 90.

At the same time, if the recorded sequence of commands is equal to

Right 90 Forward 2

then really performed command sequence can be equal to

Right 45 Forward 2.

In the last case, the robot does not have enough time to execute the command *Right 90*. We need some confirmation of the execution of commands to establish a correspondence between the recorded and really performed command sequences. In particular, we need to distinguish fully and partially executed commands. For this purpose, we consider the computational predicate logic [9]. In general, we assume that there is a difference between recorded and really performed commands. For any recorded command K , we denote by $[K]$ corresponding really performed command. So, if

$$K_1 K_2 \dots K_n$$

is a sequence of recorded commands, then

$$[K_1][K_2] \dots [K_n]$$

is the sequence of really performed commands. In general, we assume that $K \neq [K]$. It should be noted that the value of $[K]$ depends on a number of different factors,

- material of the floor covering;
- texture of the floor covering;
- moisture of the floor covering;
- viscosity of the floor covering;
- stickiness of the floor covering;
- relief;
- presence of movable obstacles;
- presence of permeable immovable obstacles;
- presence of impermeable immovable obstacles;
- CPU utilization;
- memory utilization;
- parallel execution of specific programs.

In this paper, we consider only CPU and memory utilization.

It is clear that values of $[K]$ can be found during autonomous practice. However, it is difficult task, which requires a very large number of experiments. To minimize the number of experiments, we can use a special module that builds and applies different logic rules. We consider the classical logic (CL) and the computational predicate logic (CPL). We use

$$\forall, \exists, \neg, \wedge, \vee$$

for CL and

$$\forall_c, \exists_c, \neg_c, \wedge_c, \vee_c$$

for CPL . For instance,

$$CPL \models \forall_c x \forall_c y ([xy] = [x][y]),$$

$$CL \models \forall x (x \vee \neg x),$$

$$\begin{aligned}
CPL &\not\models \forall_c[x]([x] \vee_c \neg_c[x]), \\
CL &\models \forall x \forall y \forall z (y = z \rightarrow xy = xz), \\
CPL &\not\models \forall x \forall y \forall z (y = z \rightarrow xy = xz).
\end{aligned}$$

In particular, we can consider $x = \textit{Right } 90$, $y = \textit{Backward } 1$ $\textit{Forward } 1$, and $z = \emptyset$ in the last case. Also, it should be noted that, for instance, truth of

$$CPL \models \forall_c x \forall y \forall z (y = z \rightarrow [x]y = [x]z)$$

depends on factors that the we should take into account.

We assume that the module that builds and applies logic rules uses a simple genetic algorithm to select rules. Let $N(R)$ denotes the number of adjusted weights by using the rule R . Let $|R|$ denotes the number of variables in R We consider

$$\frac{N(R)}{|R|}$$

as the fitness function.

In our experiments, we use experimental setup that described in the paper [8]. In particular, we consider the problem (K,P)-AP that proposed in the paper [8]. In our experiments, we use the special module that builds and applies logic rules and consider the time of the adjustment of the parameters. We use for comparison results from [8] that obtained purely during autonomous practice. Selected experimental results for test sets T[1] and T[2] from [8] are given in the Table 1.

	T[1]	T[2]
autonomous practice	3.214002	5.701104
autonomous practice and logic rules	2.733012	3.640019

Table 1: The time of a proper adjustment of the parameters.

It is easy to see that the time of the adjustment of the parameters is significantly less for the case of autonomous practice and logic rules. Moreover, the advantage increases with increasing the size of environments.

We consider the avatar model [10] as a supervisor of the process of autonomous practice. It is assumed that the avatar model used for replacement of operator's sequences of commands by typical human correct actions. Selected experimental results for test sets T[1] and T[2] are given in the Table 2.

It is clear that the time of the adjustment of the parameters is significantly less for the case of autonomous practice and logic rules. However, we do not obtain significant advantage with increasing the size of environments.

	T[1]	T[2]
autonomous practice	1.401173	2.612207
autonomous practice and logic rules	1.142574	2.375403

Table 2: The time of a proper adjustment of the parameters for the avatar model.

ACKNOWLEDGEMENTS. This research was partially financially supported by RFBR, research project No. 13-01-00048 a, and under the Agreement 02.A03.21.0006 of 27.08.2013 between the Ministry of Education and Science of the Russian Federation and Ural Federal University.

References

- [1] B. Argall, S. Chernova, M. Veloso, and B. Browning, A survey of robot learning from demonstration, *Robotics and Autonomous Systems*, **57** (2009), 469-483. <http://dx.doi.org/10.1016/j.robot.2008.10.024>
- [2] M. Mataric, Getting Humanoids to Move and Imitate, *IEEE Intelligent Systems*, **15** (2000), 18-24. <http://dx.doi.org/10.1109/5254.867908>
- [3] M. Muhlig, M. Gienger, and J. Steil, Interactive imitation learning of object movement skills, *Autonomous Robots*, **32** (2012), 97-114. <http://dx.doi.org/10.1007/s10514-011-9261-0>
- [4] B. Argall, B. Browning, and M. Veloso, Mobile Robot Motion Control from Demonstration and Corrective Feedback, *Studies in Computational Intelligence*, **264** (2010), 431-450. http://dx.doi.org/10.1007/978-3-642-05181-4_18
- [5] S. Niekum, S. Osentoski, G. Konidaris, S. Chitta, B. Marthi, and A. Barto, Learning Grounded Finite-State Representations from Unstructured Demonstrations, *The International Journal of Robotics Research*, **34** (2015), 131-157. <http://dx.doi.org/10.1177/0278364914554471>
- [6] J. Kober, J. Peters, *Learning Motor Skills: From Algorithms to Robot Experiments*, Springer-Verlag, Berlin – Heidelberg, 2014. <http://dx.doi.org/10.1007/978-3-319-03194-1>
- [7] A. Gorbenko, On the approximate period problem, *IAENG International Journal of Applied Mathematics*, **44** (2014), 1-9.

- [8] A. Gorbenko and V. Popov, Description of Sequences of Rhythmic Motor Primitives, *Advanced Materials Research*, **1016** (2014), 612-616. <http://dx.doi.org/10.4028/www.scientific.net/amr.1016.612>
- [9] R. Constable, *Formalizing Decidability Theorems About Automata*, NATO ASI Series, **165** (1999), 179-213. http://dx.doi.org/10.1007/978-3-642-58622-4_6
- [10] A. Gorbenko and V. Popov, Mechanical Research with Intelligent Avatars for Robot Learning from Demonstration, *Advanced Materials Research*, **952** (2014), 287-290. <http://dx.doi.org/10.4028/www.scientific.net/amr.952.287>

Received: June 7, 2015; Published: September 9, 2015