Applied Mathematical Sciences, Vol. 6, 2012, no. 132, 6577 - 6581

Anticipation in Simple Robot Navigation and Finding Regularities

Anna Gorbenko

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia
gorbenko.ann@gmail.com

Vladimir Popov

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia
Vladimir.Popov@usu.ru

Abstract

We consider robot self-awareness from the point of view of temporal relation based data mining. We consider the problem of finding regularities among effects of robot's actions and changes of the environment.

Keywords: robot self-awareness, anticipation, robot, fluents

The ability to anticipate the actions of others or some important events in the environment is something we take more or less for granted. We often do not appreciate the complexity of this ability. The robot needs some system of finding regularities (see e.g. [1] – [4]) to construct their own anticipation system. Note that the representation of knowledge of the surrounding world plays an important role in mobile robot navigation tasks (see e.g. [5] – [7]). Finding optimal solutions for such tasks usually requires to solve some hard problem (see e.g. [8] – [16]). Robot self-awareness and anticipation of some events gives the robot significant additional capabilities to solve such tasks (see e.g. [17] – [22]). In this paper, we consider the problem of finding regularities among effects of robot's actions and changes of the environment

We use autonomous mobile robot Kuzma-II as main testbed (see e.g. [20]) for our experiments. The basic robot control system developed in Java. The system is designed to work with devices. Intelligent functions assigned to the advanced robot control system. This system developed using the C#

programming language on the .NET 2.0 framework. Kuzma-II uses a visual navigation system. Using a wireless connection our robots have access to resources of a cluster. We use heterogeneous cluster.

Using of fluents allows us to establish a correspondence between sequences of images, actions of the robot, and changes of the environment. It is obvious that some part of obtained correspondence describes random dependences. Therefore, it needs to be further processed. Of course, this processing can be made by human. However, this is too expensive way. A natural approach to the detection of dependences between sequences of images, actions of the robot and changes of the environment is to find regularities in the correspondence.

Note that we can use distance functions for sequences of images. Therefore, to find regularities we use satellite models. There are two satellite models (see e.g. [23]). One called prefix model and the other consensus model.

A prefix model of a satellite is a string $w \in \Sigma^*$ that approximately matches a train of wagons. A wagon of w is a substring u in string x such that $\delta(w, u) \leq e$. A train of a satellite model w is collection of wagons $u_1, u_2, \ldots, u_p$ ordered by their starting positions in x and satisfying the following properties.

1. $p \geq min\_repeat$, where min_repeat is a fixed parameter that indicates the minimum number of elements a repetitive region must contain.

2. $left_{u_{i+1}} - left_{u_i} \in JUMP$, where $left_u$ is the position of the left-end of wagon u in x and

$$JUMP = \{y \mid y \in \cup_{x \in [1, max\_jump]} x \times [min\_range, max\_range]\},$$

with the three parameters min_range, max_range and max_jump fixed.

A prefix model w is said to be valid if there is at least one train of w in the string x. Similarly, a train, when viewed simply as a sequence of substrings of x, is valid if it is the train for some model w.

Consensus model is a prefix model which further satisfies the following property.

3. $left_{u_{i+1}} - right_{u_i} \in GAP$, where $right_u$ is the position of the right-end of wagon u, and

$$GAP = \{y \mid y \in \cup_{x \in [0, max\_jump-1]} x \times [min\_range, max\_range]\}.$$

The distance function in the consensus model is the consensus function, which finds a consensus string of wagons, i.e. string w such that the distance between the string w and each string in $\{u_1, u_2, \ldots, u_p\}$ is at most e. Another way to define a consensus string is to use the consensus error. The consensus error of a string w with respect to a given set $\{u_1, u_2, \ldots, u_p\}$ is the sum of the distances between w and all the strings in $\{u_1, u_2, \ldots, u_p\}$. Parameter max_jump allows us to deal with very badly conserved elements inside a satellite (by actually not counting them). Consensus error allows us to deal with relatively badly conserved wagons inside a satellite (and counting them) while we require that the satellite be relatively well conserved globally.

Let a consensus error model is a string $w \in \Sigma^*$ that approximately matches with consensus error a train of wagons, i.e. $\sum_{i=1}^{p} \delta(w, u_i) \leq e$.

Let us consider the following problem:

THE SATELLITE PROBLEM FOR CONSENSUS ERROR (SPCE):

INSTANCE: *Parameters min_repeat, min_range, max_range, max_jump, and e, a distance function δ, a string x.*

TASK: *Find a consensus error model w that is valid for x.*

Although SPCE is **NP**-hard [2], this problem describes the model which is the most general and interesting from the practical point of view for localization and extraction of regularities. In view of practical significance of this model, researchers have extensively studied algorithms for this model. In particular, under various constraints for this model proposed various combinatorial algorithms and ideas in [24, 25]. Using such algorithms some labeled data can be obtained. After this learning from unlabeled data can be used. In our framework used the following two genetic algorithms: algorithm for direct prediction of the location of regularities in a data sequence (DPL); algorithm for selection of combinatorial solvers. Co-training used to improve quality of both algorithms. Also co-training used to improve quality of the algorithm for direct prediction of values of parameters *min_repeat*, *min_range*, *max_range*, *max_jump*, and e (DPV).

Note that we can use parallel run of combinatorial solvers. However, the algorithm for selection of combinatorial solvers allows us significantly reduce the number of computational nodes.

Algorithms DPL and DPV allow us significantly reduce processing time of data sequence. It should be noted that the performance of these algorithms considerably depends from the number of co-training steps and the length of data sequence (see Table 1). Dependencies of the quality of prediction from the number of co-training steps are shown in Table 2.

	10^3	10^5	10^7
brute force	5.2 h	19.7 h	63.7 h
DPL (10^3 steps)	1.3 h	3.1 h	7.2 h
DPL+DPV (10^3 steps)	51 min	1.3 h	3.1 h
DPL (10^5 steps)	1.1 h	2.6 h	5.4 h
DPL+DPV (10^5 steps)	11 min	28 min	57 min

Table 1: The dependence from the number of co-training steps and the length of data sequence.

	10^2	10^3	10^5	10^7	10^8
DPL	44 %	68 %	93 %	94 %	94 %
DPV	36 %	57 %	82 %	87 %	87 %

Table 2: The quality of prediction for DPL and DPV.

References

[1] A. Gorbenko and V. Popov, The set of parameterized k-covers problem, *Theoretical Computer Science*, 423 (2012), 19-24.

[2] V. Popov, Multiple genome rearrangement by swaps and by element duplications, *Theoretical Computer Science*, 385 (2007), 115-126.

[3] V. Popov, The approximate period problem for DNA alphabet, *Theoretical Computer Science*, 304 (2003), 443-447.

[4] A. Gorbenko and V. Popov, On the Longest Common Subsequence Problem, *Applied Mathematical Sciences*, 6 (2012), 5781-5787.

[5] A. Gorbenko and V. Popov, A Real-World Experiments Setup for Investigations of the Problem of Visual Landmarks Selection for Mobile Robots, *Applied Mathematical Sciences*, 6 (2012), 4767-4771.

[6] A. Gorbenko and V. Popov, On Face Detection from Compressed Video Streams, *Applied Mathematical Sciences*, 6 (2012), 4763-4766.

[7] A. Gorbenko, A. Lutov, M. Mornev, and V. Popov, Algebras of Stepping Motor Programs, *Applied Mathematical Sciences*, 5 (2011), 1679-1692.

[8] A. Gorbenko and V. Popov, Task-resource Scheduling Problem, *International Journal of Automation and Computing*, 9 (2012), 429-441.

[9] A. Gorbenko and V. Popov, The Binary Paint Shop Problem, *Applied Mathematical Sciences*, 6 (2012), 4733-4735.

[10] A. Gorbenko and V. Popov, The Problem of Selection of a Minimal Set of Visual Landmarks, *Applied Mathematical Sciences*, 6 (2012), 4729-4732.

[11] A. Gorbenko and V. Popov, Computational Experiments for the Problem of Selection of a Minimal Set of Visual Landmarks, *Applied Mathematical Sciences*, 6 (2012), 5775-5780.

[12] A. Gorbenko and V. Popov, Programming for Modular Reconfigurable Robots, *Programming and Computer Software*, 38 (2012), 13-23.

[13] A. Gorbenko and V. Popov, On the Problem of Placement of Visual Landmarks, *Applied Mathematical Sciences*, 6 (2012), 689-696.

[14] A. Gorbenko, M. Mornev, V. Popov, and A. Sheka, The problem of sensor placement for triangulation-based localisation, *International Journal of Automation and Control*, 5 (2011), 245-253.

[15] A. Gorbenko, M. Mornev, and V. Popov, Planning a Typical Working Day for Indoor Service Robots, *IAENG International Journal of Computer Science*, 38 (2011), 176-182.

[16] A. Gorbenko, V. Popov, and A. Sheka, Localization on Discrete Grid Graphs, *Lecture Notes in Electrical Engineering*, 107 (2012), 971-978.

[17] A. Gorbenko and V. Popov, Robot Self-Awareness: Occam's Razor for Fluents, *International Journal of Mathematical Analysis*, 6 (2012), 1453-1455.

[18] A. Gorbenko and V. Popov, The Force Law Design of Artificial Physics Optimization for Robot Anticipation of Motion, *Advanced Studies in Theoretical Physics*, 6 (2012), 625-628.

[19] A. Gorbenko, V. Popov, and A. Sheka, Robot Self-Awareness: Exploration of Internal States, *Applied Mathematical Sciences*, 6 (2012), 675-688.

[20] A. Gorbenko, V. Popov, and A. Sheka, Robot Self-Awareness: Temporal Relation Based Data Mining, *Engineering Letters*, 19 (2011), 169-178.

[21] A. Gorbenko and V. Popov, Anticipation in Simple Robot Navigation and Learning of Effects of Robot's Actions and Changes of the Environment, *International Journal of Mathematical Analysis*, 6 (2012), 2747-2751.

[22] A. Gorbenko and V. Popov, The c-Fragment Longest Arc-Preserving Common Subsequence Problem, *IAENG International Journal of Computer Science*, 39 (2012), 231-238.

[23] M. Crochemore and M.-F. Sagot, Motifs in sequences: localization and extraction, *Handbook of Computational Chemistry*, (2004), 47-97.

[24] B. Clift, D. Haussler, R. McConnell, T. D. Schneider, and G. D. Stormo, Sequence landscapes, *Nucleic Acids Research*, 14 (1986), 141-158.

[25] M. Crochemore, An optimal algorithm for computing the repetitions in a word, *Information Processing Letters*, 12 (1981), 244-250.

Received: September, 2012