Journal of Robotics, Networking and Artificial Life
Vol. 7(2); September (2020), *pp.* 81–85
DOI: https://doi.org/10.2991/jrnal.k.200528.002; ISSN 2405-9021; eISSN 2352-6386
https://www.atlantis-press.com/journals/jrnal

Research Article

Behavior Selection System for Human–Robot Cooperation using Tensor SOM

Moeko Tominaga[1,*], Yasunori Takemura[2], Kazuo Ishii[1]

[1]*Graduate School of Life and Science Systems Engineering, Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu-ku, Kitakyushu, Fukuoka 808-0196, Japan*
[2]*Department of Engineering, Nishinippon Institute of Technology Technology, 1-11 Aratsu, Kanda-town, Miyako-gun, Fukuoka 800-0397, Japan*

ARTICLE INFO

Article History
Received 29 November 2019
Accepted 01 April 2020

Keywords
Strategy
self-organizing map
team behavior
Tensor SOM
multi-agent system
human–robot cooperation

ABSTRACT

With the progress of technology, the realization of a symbiotic society with human beings and robots sharing the same environment has become an important subject. An example of this kind of systems is soccer game. Soccer is a multi-agent game that requires strategies by taking into account each member's position and actions. In this paper, we discuss the results of the development of a learning system that uses self-organizing map to select behaviors depending on the situation. A set of possible actions in soccer game is decided in advance and the algorithm is able to select the best option, given some specific conditions.

1. INTRODUCTION

Recently, the implementation of robots in society has become a possible solution to many problems, such as ensuring the safety of a sustainable society, responding to a rapid population aging and population decrease. Moreover, robots will represent the foundation of the future industry.

To properly implement a robotized society, it is necessary to conduct and to present research outcomes in a manner that is easy to understand, avoiding differences between social expectations and the direction of research and development.

Therefore, it is essential to discuss how to achieve coexistence with robots and what a symbiotic society should look like. In such a society, humans and robots interact with each other and they are capable of mutual understanding. Not only they need to be aware of their own actions, but also of all the other agents' actions, where agent means each of the active subject involved.

The aim of this work is to develop a suitable algorithm to create intelligent robots able to share the environments with humans. Since soccer involves strategies, cooperation, unpredictable movements and common targets, it represents a good test bed for developing such algorithm.

Tensor Self-Organizing Map (Tensor SOM) [1] is used for this scope.

*Corresponding author. Email: tominaga.moeko382@mail.kyutech.jp

2. COOPERATIVE BEHAVIOR

Cooperative behavior becomes a crucial aspect when different autonomous agents interact while performing a common task. Often a single agent is not much effective in accomplishing a task, and in the last years many researchers have been studying Multi-Agent Systems (MAS) to solve difficult problems.

The agents interact to each other and with the environment by taking real time decisions based on the data acquired from the sensors [2,3].

As a test bed of MAS, RoboCup [4], a project aimed to win the Soccer World Cup against Humans, encourages the cooperation of multi-agents using learning methods, such as reinforcement learning and neural networks. According to Sandholm and Crites [5], reinforcement learning can be used successfully for the iterated prisoner's dilemma, if sufficient measurements data and actions are available. In addition, Arai et al. [6] compared the Q-learning and Profit Sharing methods about the Pursuit Problem in a MAS, when the environment is modelled as a grid, and showed that cooperative behaviors emerge clearly among Profit Sharing. However, these studies have not yet considered applications for robots that operate in a real environment.

3. ROBOT LEARNING

Among the learning algorithms, unsupervised learning is a promising method for MAS systems.

The advantage in unsupervised learning is that the robot is not required to have any previous information concerning the environment and prerequisite knowledge of the robot itself. However, since many tests and experiences are needed for learning, it is important for the designer to define in advance and clearly what are the important parameters and variables to consider during the game, so as to define the state space and the action space, and to reduce the cost and the time.

The design of the state space depends on what the robot is able to do in the action space. The two spaces are mutually interconnected.

To design the state space in an efficient way, Takahashi and Asada [7] proposed a method in which the state space is initially divided into two main states, and then divided recursively into many layers with an increasing number of states. Nonetheless, they found that some bias can still affect the behavior of the robot and cause some wrong action and habit.

3.1. Human–Robot Cooperation

As mentioned above, a humans–robots cooperative society is a topic of interest for many researchers and several works have been carried out.

In such a symbiotic system, it is mandatory for the robots to understand and interpret humans' behavior and act accordingly. However, the target in research is often about the robots' behavior only, or a behavior based on an interaction defined precisely between a specific number of humans and robots. Also, in a MAS the communication between humans and robots is usually carried out by means of some kind of interface, for example command voice or gestures. But this might not be enough, especially when the system is very complex.

Accordingly, the robot should be able to understand some situations and adapt its behavior in a predictive manner, just as humans do.

The main goal of this research is to analyze the cooperative behaviors among soccer players and apply them to robots (see Figure 1).

A neural network based on Tensor SOM is designed to achieve this target.

The input vector elements for a neural network is chosen so as to achieve the lowest possible gap between humans and robots' actions. To obtain this, it is important at first to understand humans' behavior and develop robots able to understand such behaviors and imitate them.

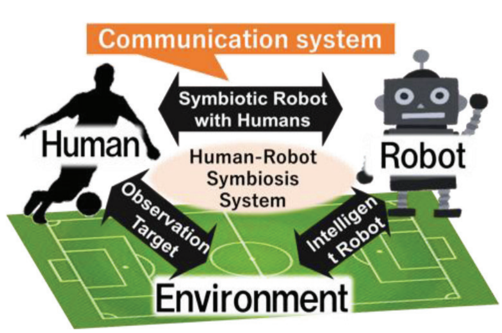


Figure 1 | A human–robot symbiotic system.

This paper focuses on the study of robots' behavior in a futsal game, because this represents a good test bed, with a dynamic environment, several constrains and it requires a real-time planning. These are the characteristics of a common symbiotic system in which the robots may operate in the future.

4. SOM AND TENSOR SOM

Self-organizing map [8] is an unsupervised learning technique and is known as competitive learning, similar to information processing via neural circuits. Competitive learning is an important concept in hierarchical neural networks. Each input neuron in the input layer (input space) is connected to all output layer neurons. The strength of the relation between two neurons in consecutive layers is decided by a specific weight assigned to each connection.

The input vectors "compete" with each other so as to find the best output vector. With best output, it is meant that the output vector whose elements are as similar as possible to the input vector elements.

To establish this similarity, the Euclidean distance is used. By doing this, clusters into the output map are obtained. The cooperative hierarchy deals with the output vectors. While performing the algorithm, the elements value of each vector is adjusted. If a high change of an element occurs in a vector, the vectors around it change accordingly, but in such a way that the strength of this change decreases with the distance in the map from the first vector. In particular, the amount of change of each element is decided by the Gaussian function [9].

Since each vector modifies itself depending on the surrounding vectors, this behavior is said to be cooperative. Input vectors can have high dimension but the output results are shown in a two-dimensional map.

With this structure, multi-dimensional input vectors can have reduced dimensions in the output layer, and the features are clustered, so as to have similar features in the same area of the map. Since the output results are shown in a two-dimensional form, SOM is suitable for the visualization of complex scenarios.

Furthermore, the cooperative hierarchy and the smoothing process performed by the Gaussian function allow to synthesize elements and contents that were not clearly specified in the input vectors. SOM has a high interpolation performance and it can generalize concepts starting from a limited amount of input data. Because of these characteristics, it is well suited for unsupervised learning and highly valid outputs are expected.

A limit of SOM is that input vectors with variables of different nature are difficult to normalize. Also, the relations between these inputs to produce outputs is relatively limited. An alternative approach is to use Tensor SOM. With Tensor SOM it is possible to create a higher order input variables and a wider combinations of inputs and, accordingly, output (Figure 2).

4.1. Algorithm

The Tensor SOM algorithm proceeds according to the following steps.

When an input vector is introduced in the algorithm, its Euclidean distance from all the weights vectors is evaluated. The unit (neuron)

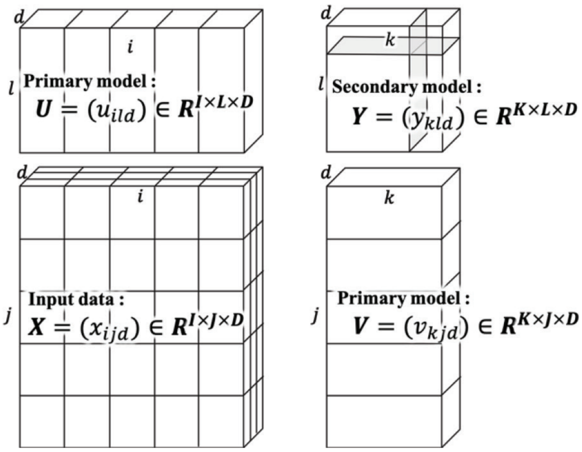


Figure 2 | Structure of input/output tensor in Tensor SOM.

with the most similar weights is selected as the Best Matching Unit (BMU).

Figure 2 shows the structure used to update the weights and training of the neural network. This method is detailed in Iwasaki and Furukawa [1].

The input tensor ***X*** consists of all the input data. These inputs are divided into some categories, three in Figure 2, and the indices are: *d*, *i* and *j*.

The secondary model represents the weights output tensor. At first this is initialized with some random values.

In order to update the output tensor values, two primary models are used. In the first of the primary model (above the input tensor in Figure 2) the combinations of all the *d* and *i* conditions are evaluated. This forms a stack of *l* layers. Same happens for the other primary model (at the right of the input tensor in Figure 2), with the combinations of all *d* and *j* conditions, producing a stack of *k* layers.

The steps required for the algorithm are detailed below.

4.1.1. Choose best matching unit

The selection of the BMU in each layer, k_i^*, l_i^* is carried out for each input vector u_{ild}, v_{kjd} with the following Equations (1) and (2), in which the Euclidean distance is calculated.

$$k_i^* = \arg\min_k \sum_{l=1}^{L}\sum_{d=1}^{D}(u_{ild} - y_{kld})^2 \tag{1}$$

$$l_j^* = \arg\min_l \sum_{k=1}^{K}\sum_{d=1}^{D}(v_{kjd} - y_{kld})^2 \tag{2}$$

where k_i^* is the BMU in the k_i layer, l_j^* is the BMU in the l_j layer, and y_{kld} represents the outputs in the secondary model.

4.1.2. Weight adjustments

Based on the distance from the BMU in the layer and on the radius σ, the weight adjustments are defined as in Equations (3) and (4).

$$\alpha_{k_i} = \exp\left[-\frac{1}{2\sigma^2}\left\|\zeta_{k_i^*}^{(1)} - \zeta_k^{(1)}\right\|^2\right] \tag{3}$$

$$\beta_{l_j} = \exp\left[-\frac{1}{2\sigma^2}\left\|\zeta_{l_j^*}^{(2)} - \zeta_l^{(2)}\right\|^2\right] \tag{4}$$

where $\zeta_{k_i^*}^{(1)} - \zeta_k^{(1)}$, $\zeta_{l_j^*}^{(2)} - \zeta_l^{(2)}$ is the distance between the BMU and the surrounding units.

With the weights adjustments found above, the secondary and primary models values are adjusted as in Equations (5)–(9).

Update the secondary model:

$$y_{kld} = \frac{1}{g_k g_l}\sum_{i=1}^{I}\sum_{j=1}^{J}\alpha_{k_i}\beta_{l_j}x_{ijd} \tag{5}$$

$$g_k = \sum_{i=1}^{I}\alpha_{k_i} \tag{6}$$

$$g_l = \sum_{j=1}^{J}\beta_{l_j} \tag{7}$$

Update the primary model:

$$u_{ild} = \frac{1}{g_l}\sum_{j=1}^{J}\beta_{l_j}x_{ijd} \tag{8}$$

$$v_{kjd} = \frac{1}{g_k}\sum_{i=1}^{I}\alpha_{k_i}x_{ijd} \tag{9}$$

5. EXPERIMENTS

The target is to verify whether the Tensor SOM can be used as a behavior selection algorithm in a futsal game. In this case the player we want to take some decision is that one who has the ball during the game action analyzed.

As a first step, the tensor inputs need to be specified. We decided to use three categories of inputs: agents, environments and behaviors (Figure 3).

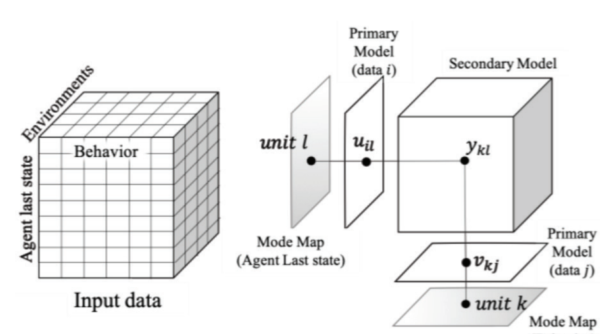


Figure 3 | Image of input data.

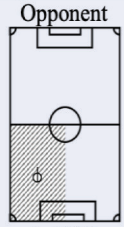
Environments	Behavior				Agent last state
	PASS	DRIBBLE	SHOOT	KICK OUT	Opponent  1 : RUN
Press & Helper	4	3	1	2	
Nobady	2	4	3	1	
Helper	2	3	4	1	
Press	2	3	1	4	

Figure 4 | Structure of input data.

Table 1 | Training parameters using Tensor SOM

Number of iterations	n	10
Map size		100 × 100
Max neighboring radius	σ_{max}	2.0
Minimum neighboring radius	σ_{min}	0.2
Number of learning	Epoch	200

The agent here is the target player, namely the player with the ball, and his current action, in movement (named *run*) or standing (named *stop*).

Environments consist of four situations:

1. Press & Helper: around the target player there are both teammates and opponents.
2. Nobody: the target player has no teammates nor opponents in his surroundings.
3. Helper: around the target player there are only teammates.
4. Press: around the target player there are only opponents.

Finally, the behaviors consist of four possible actions performed by the target player:

1. Pass: the ball is passed to a teammate.
2. Dribble: the player decides to continue running with the ball.
3. Shoot: the player tries to score a goal.
4. Kick out: the player decides to send the ball out of the field.

To each of this possible behavior is assigned a score from 1 to 4, based on the situation in the game (environments inputs), where 4 means best choice and 1 means worst choice.

In Figure 4, the inputs with the score for each behavior is shown, given a position of the player in the field and his current action, for example *run*. In this case, if nobody is in the surroundings of the target player, the best choice is to keep running with the ball, while kick out the ball would be the worst decision.

With the defined conditions we trained the algorithm. In Table 1 are shown the training parameters used.

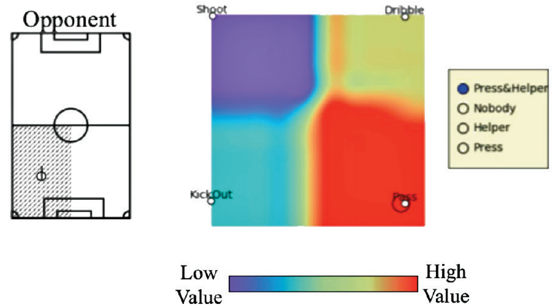


Figure 5 | Result of Tensor SOM expressed by the Component Plane Matrix after learning.

5.1. Experimental Results and Discussion

The results are analyzed by means of the Component Plane Matrix.

In Figure 5, it is shown the agent position and current action on the left, the four environments conditions on the right side and finally the matrix with the behaviors in the center.

We can see as the target player chose its behavior according to the decision criteria summarized in Figure 4.

In fact, in this situation he is running in his own field half, there are both teammates and opponents in his surroundings and his choice is to pass the ball (high value in the Component Plane Matrix).

6. CONCLUSION

In this paper, we created a behavior selection system for soccer robot players in a futsal game.

At first we developed an algorithm using Tensor SOM. This algorithm was selected because of its ability to deal with complex systems where several inputs are present. Also, it is suited for unsupervised learning.

These inputs consist of three categories: behaviors, agents and environments. Then we created a set of training data to train the network.

The results provided in the Component Plane Matrix are in agreement with the expectations and shows as this algorithm is able to select the predicted behavior. Given a position and current action of the target player and some specific situation in the game, he was able to choose the best action. Tensor SOM can be a powerful tool in the development of MAS, even though further tests and experiments in real conditions are required to evaluate its potential.

As a future work, this algorithm will be implemented on the real soccer robots used for RoboCup Middle Size League and tested in real conditions.

CONFLICTS OF INTEREST

The authors declare they have no conflicts of interest.

REFERENCES

[1] T. Iwasaki, T. Furukawa, Tensor SOM and tensor GTM: nonlinear tensor analysis by topographic mappings, Neural Netw. 77 (2016), 107–125.

[2] P. Stone, M. Veloso, Multiagent systems: a survey from a machine learning perspective, Autonom. Rob. 8 (2000), 345–383.

[3] K. Ueda, Emergence and multi agent system, Bifuran (2007) (Article in Japanese).

[4] S. Behnke, R. Sheh, S. Sariel, D.D. Lee (Eds.), RoboCup 2016: robot World Cup XX, Springer-Verlag, Berlin, Heidelberg, 2017, p. 645.

[5] T.W. Sandholm, R.H. Crites, On multiagent Q-learning in a semi-competitive domain, In: G. Weiß, S. Sen (Eds.), Adaption and Learning in Multi-Agent Systems. IJCAI 1995. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), Springer, Berlin, Heidelberg, 1995, pp. 191-205.

[6] S. Arai, K. Miyazaki, S. Kobayashi, Methodology in multi-agent reinforcement learning: approaches by Q-learning and profit sharing, J. Japanese Soc. Artif. Intell. 13 (1998), 609–618 (Article in Japanese).

[7] Y. Takahashi, M. Asada, Behavior acquisition by multi-layered reinforcement learning, J. Robot. Soc. Japan 18 (2000), 1040–1046.

[8] A. Ultsch, H.P. Siemon, Kohonen's self organizing feature maps for exploratory data analysis, Proceedings of the International Neural Network Conference (INNC), Kluwer Academic Press, Dordrecht, Netherlands, 1990, pp. 305–308.

[9] L.A. Tu, Sustainability in urban planning and design, Improving feature map quality of SOM based on adjusting the neighborhood function, IntechOpen, 2019, pp. 1–17.

AUTHORS INTRODUCTION

Ms. Moeko Tominaga

She received her M.S. degree from Department of Human Intelligence Systems, at Kyushu Institute of Technology, Japan, in 2018. She is pursuing the PhD at Kyushu Institute of Technology, Department of Life Science and Systems Engineering, under the supervision of Prof. Kazuo Ishii. Her research interests include symbiosis of humans and robots.

Dr. Yasunori Takemura

He received PhD degree from Kyushu Institute of Technology, Japan in 2010 and now he is working as an Associate Professor at Nishinippon Institute of Technology, in Japan. His research area is about machine learning, data mining and robotics.

Prof. Kazuo Ishii

He received PhD degree from the Department of Naval Architecture and Ocean Engineering, University of Tokyo, Tokyo, Japan, in 1996. He is currently a Professor with the Department of Human Intelligences Systems, and the Director of the Center for Socio-Robotic Synthesis, Kyushu Institute of Technology, Kitakyushu, Japan. His research interests include underwater robots, agricultural robots, RoboCup soccer robots and intelligent systems.