A NEW NAVIGATION METHOD BASED ON REINFORCEMENT LEARNING AND ROUGH SETS

HONG-YAN WU, SHU-HUA LIU, JIE LIU

School of Computer Science, Northeast Normal University, Changchun, 130024, China
E-MAIL: wuhy836@nenu.edu.cn liush129@nenu.edu.cn

Abstract:

The ability of autonomous navigation and adaptability to environment are key issues for the application of mobile robots in complex and unknown environments. A new method based on reinforcement learning and rough sets is proposed to accomplish robot navigation tasks in unknown environment. With reinforcement learning, robot can achieve autonomous navigation in an unknown environment. Because of the navigation knowledge of robot has the characteristic of incompleteness and inaccuracy, rough sets is an effective mathematical tools to deal with incompleteness. Rough sets can deal with robot initial navigation knowledge and simplify the complexity of navigation, therefore it can speedup the learning process of autonomous mobile robot and improve the obstacle avoidance ability of navigation system. This is the reason why we lead rough sets into reinforcement learning process. Finally, the effectiveness of the presented method is verified in simulation environment. The simulation results show that our method not only provides an effective way for the self-learning of mobile robot but also has good obstacle avoidance ability.

Keywords:

Reinforcement learning; Q learning; Rough sets; Autonomous navigation

1. Introduction

Because there is little prior knowledge for mobile robot in unknown environment, the mobile robot should have the ability of dealing with emergence and adapting to the environment to complete tasks safely and reliably. It's required that navigation system should have flexibility and adaptability, while reactive control paradigm is an important means to improve real-time and flexibility in unknown environment. In recent years, among most reactive control methods, reinforcement learning has been broadly applied into robot navigation field in unknown environment because of its self-learning and on-line learning abilities [1][2][3].

The basic principle of reinforcement learning is that agent interacts with environment at each of discrete time steps to obtain knowledge or experience so as to improve action process.

Although reinforcement learning is an effective tool to implement mobile robot reactive navigation, but it has a few limits when applied into navigation area [4]:

(1) The computational complexity increases exponentially when the state-action pairs increases. Therefore, it hardly meets the real-time requirement.

(2) The trial-and-error when reinforcement learning interacts with environment may lead to risk for navigation system.

Reducing the risk of trial-and-error of tabular reinforcement learning and improving the obstacle avoidance capability of navigation system become key issues. At the beginning of learning, the robot has no experience, just constantly accumulates experience and obtains required rules through practical collision avoidance learning. It's a long time between no experience and experienced. Some researchers present the general and intuitive approach for incorporating prior knowledge into the reinforcement learning system, speedup the robot learning. Kevin presented a method which build an off-policy reinforcement learning controller [5], train robot off-line, incorporate the gained experience into the real learning process, simulation results show that it can improve learning performance effectively, reduce learning time; Meiping Song combined conventional rules control with reinforcement learning [6], instruct learning controller using known regulations, simulation results prove learning convergence more quickly, the robot move smoothly along the obstacle fringe, hardly collision with the obstacle.

According to domain experience, we create an initial obstacle avoidance decision table for robots and incorporate the decision table into navigation system as robot's initial prior knowledge. The environment information, which the robot gains from sensor with a lot of noises and errors, is imperfect and inaccurate due to sensor limitation. The rough sets introduced by Zdzislaw Pawlak is a new

978-1-4244-2096-4/08/$25.00 ©2008 IEEE

mathematical tool to deal with vagueness and uncertainty [7]. The core properties of rough sets are attribute reduction and rule extraction. In this paper, rough sets is used to deal with initial obstacle avoidance decision table and obtain the simplest obstacle avoidance complete rule table which directs the robot how to avoid collision. Combining Rough sets with reinforcement learning, the avoidance obstacle ability of robots can be improved greatly.

The paper is organized as follows: In section 2, The basic concepts of reinforcement learning and Q learning; In section 3, Rough sets knowledge representation system; In section 4, Reinforcement learning and rough sets based navigation algorithm is given; In section 5, Simulation results; In section 6, Conclusions and future work.

2. Reinforcement learning

Reinforcement learning is an unsupervised and on-line learning method [8]. Agent is given feedback through interacting with the environment. The basic idea is that actions correlated with high reward have higher repeated probability, while those correlated with low reward have lower repeated probability [9]. Therefore, reinforcement learning is an incremental and real-time learning method.

The basic elements of reinforcement learning include: possible state set S, possible action set A, policy π, instantaneous reward R and utility function V(S). Every possible state s belongs to a finite state sets S, while the decision which policy π acts belongs to a finite control action sets.

The control policy is denoted $\pi:(S \to A)$. At each time step, the agent implements a mapping from states to probabilities of selecting each possible action. The mapping function is: $a=\pi(x)$, $x \in S, a \in A$.

The instantaneous reward is an environment feedback after agent acts a possible action. Usually it is a scalar reward and reflects the direct effect of the action.

Utility function V(s) is a measure of distance between state s to the goal state, defined as：under a control policy, the utility function is formally defined as the expected value of the sum of all future rewards, discounted by their delay, given that the system starts in x.

$$V(x)=E[\sum_{t=0}^{\infty}\gamma^t r_{t+1} \mid x_k = x] \qquad (1)$$

Where, E is mathematical expectation; γ is a discount factor, $\gamma \in [0,1]$; r_{t+1} is the reinforcement signal in t+1 moment. The larger of utility function, the nearer to goal. In reinforcement learning, the evaluate value of utility function gradually approximation maximum evaluate value under the optimal policy, while the control policy approximation the optimal policy.

Reinforcement learning has many implementation methods. Among them, Q learning is often adopted because of it's maturity and simplicity.

2.1. Q learning

Q learning is a form of model-free reinforcement learning [10]. It provides agent with the capability of learning to act optimally in Markov domains by experiencing action consequences. Agent selects optimal policy through iterating Q function. Q learning usually stores Q value relative with every state-action in a lookup table. Watkins has showed that Q learning convergence correctly under certain conditions.

In Q learning, there are sequences of episodes, learning steps repeat in every episode, in t^{th} time step:

step1 observe current state s_t;

step2 select and perform an action a_t;

step3 observe the subsequent state s_{t+1};

step4 receive the immediate reward $r_t = r(s_t, a_t)$;

step5 update $Q_{t-1}(s_t, a_t)$ according to:

$$Q(s_t,a_t)=\begin{cases}(1-\alpha_t)Q_{t-1}(s_t,a_t)+\alpha_t[r_t+\gamma \max_{a\in A} Q_{t-1}(s_{t+1},a\ )] & s=s_t, a=a_t \\ Q(s_t,a_t) & otherwise\end{cases} \qquad (2)$$

step6 $t \leftarrow t+1$ turn next moment.

step7 Repeat above steps until stop criterion is satisfied.

Where α is the learning rate and γ (0 < γ < 1) is the discount factor that reduces the influence of future expected rewards.

3. Rough sets knowledge representation system

Rough sets was introduced by Zdzislaw Pawlak in the early 1982s [7] , which is a new mathematical tool to deal with vagueness and uncertainty. The prominent advantage of rough set theory is that it does not require any additional empirical information of data sets. By analyzing and reasoning the database, this theory can discover the patterns and rules hidden in database. Rough sets has been widely used in Machine learning, knowledge acquisition, decision analysis and pattern recognition so on.

3.1. Basic concepts

Definition 1: The decision table is the 4-tuples in the form of S= (U,A,V,f). where, U is a finite set of objects, denotes as objects domain; A is a finite of set of attributes, non-empty subset C and D while $C\cup D=A, C\cap D=\varnothing$ are condition and result attributes respectively. $V=\bigcup_{a\in A} V_a$ where for every $a\in A$,there is a function f：$U\times A\rightarrow V$ is an information function, in which V_a is the value set of a (V_a = a(x) ,where $x\in U$).

Definition 2: (Consistency): A decision table is consistent if for every set of objects whose attribute values are the same, the corresponding decision attributes are identical; otherwise inconsistency table.

Definition 3: (Attribute reduction and core): For decision table S=(U, A, V, f) and arbitrary subset B⊆A, define an indiscernibility relation in U as follows:

ind(B)={(x,y)∈U×U:f(x,a)=f(y,a), for all a∈A}

In other words, if given any (x,y) ∈ind(B), then both x and y have the same attributes. According to indiscernibility relation ind(B), the objects which have the same attribute values are classified into one equivalence class. U was partitioned into r equivalence class C_1, C_2, C_3 …Cr, such that U/ind(B)={C_1,C_2,C_3,…C_r}. If U/ind(B)=U/ind(A), then B is a reduction of A. Such that if a set of attributes and its superset define the same indiscernibility relation, then any attribute that belongs to the superset and not belongs to the set is redundant. The intersection of all the reduction sets is called the core.

Based on the rough set theory, we construct an initial obstacle avoidance decision table for robot. The environment states which robot may stay consist of the condition attributes, while forbidden set of actions in this states denote the decision attribute. Because of the robot in the same obstacle states, the subsequent forbidden actions may more than one, based definition 2 in this chapter, the obstacle avoidance decision table we construct is an inconsistency decision table.

The Skowron's discernibility matrix usually reduces attributes for consistent decision table [11], however, the avoidance obstacle decision table is inconsistent. Therefore, we employed improved discern matrix algorithm proposed by Zhi Tao [12].

4. Reinforcement learning and rough sets

4.1. States of robot

The autonomous mobile robot perceives its environment state through sensor input signal. We assume that the sensor can detect all obstacles in its detection range. The front of robot are partitioned three zones: forward(F), left(L), right(R). In each zone, there are three sonar sensors. Any environment state s can be denoted as Cartesian product:

$$s_1\times s_2\times s_3\times s_4 \tag{3}$$

Where, set $s_i(1\le i\le 3)$ is the distanced of robot to obstacles of F, L and R respectively, s_4 is the distance between robot and goal.

Because states space is continuous highly dimension, the state-action tuples are exponentially increasing when the number of robot is increasing. Therefore, we use quantitative method of Box proposed by Michie to discrete the subset of states S [13]. The subset $s_i\,(1\le i\le 4)$ are partitioned into finite measure of different degree belongs to distance measure. Let da, md, sa and rf denotes danger, mid distance safe within the sensor detection range and out of sensor detection range respectively. The distance between robot and goal is partitioned into four discrete measures : uv，vf，vm，vn，according to invisible out of sensor detection range, or further, mid distance and near within the sensor detection range. Let $d_{1,}$ $d_{2,}$ d_3 be the distance value detected by the sensor in each zone, d_4 is the distance value to goal, define mid=min(d_1,d_1,d_3) as the minimal distance to obstacles in this zone. Define d_s as the minimal safe distance of robot and d_r as the maximum detecting radius of sensor. The environment information of robot between F, L, R and goal is denoted as equation (4) and (5):

$$\underset{(1\le i\le 3)}{s_i} = \begin{cases} da & \text{mid} < d_s \\ md & d_s < \text{mid} < d_r/2 \\ sa & d_r/2 < \text{mid} < d_r \\ rf & \text{mid} > d_r \end{cases} \tag{4}$$

$$s_4 = \begin{cases} uv & d_4 > d_r \\ vf & \tfrac{2}{3}d_r < d_4 < d_r \\ vm & \tfrac{1}{3}d_r < d_4 < \tfrac{2}{3}d_r \\ vn & d_4 < \tfrac{1}{3}d_r \end{cases} \tag{5}$$

4.2. Actions of robot

The actions include moving and rotating. The moving is forward or back while the rotate actions are partitioned into five discrete actions according to different angle as follows:

$A=\{a_i,\ i=1, \cdots,5\}=\{0°, \pm 20°, \pm 45°\}$. When robots turn left, the angle is positive, or else negative.

4.3. Reinforcement signal

Reinforcement signal is equivalent to instantaneous reward relative with environment state and subjective goal. How to design reward will directly affect quality and speed of learning.

Tucker Balch had classified the reward of reinforcement learning by a series of descriptor [14]. In this paper, we use discrete reward that is simple and effective.

As defined in section 4.1, the safe distance of robot is d_s, the minimum distance between robot and obstacles is mid. In t-1 moment, the distance between robot and goal is d_0, while in moment t, the distance between robot and goal is d_1, the instantaneous reward are designed as:

$$r=\begin{cases} -2 & mid \le d_s \\ -0.2 & mid > d_s\ ,\ (d_1-d_0)>0 \\ 0.2 & mid > d_s\ ,\ (d_1-d_0)\le 0 \\ 2 & d_0 < d_s \end{cases} \qquad (6)$$

4.4. The initial obstacle avoidance decision table

The decision table is defined in section 3.1, the condition attributes are environment states including obstacles which robot may stay in, whereas the decision attributes are forbidden set of actions in this state. The condition attribute is defined as:

C={Disleft,Disright,Disahead,Disgoal}

Every element is quantified into between 0 and 3. 0 denotes out of sensor range,1 denotes the obstacle near the robot, 2 denotes middle distance and 3 denotes far from robot. Decision attribute D correspond to five optional action, 0：robot forward with angle 0°；1: robot turn left with angle 20°；2: robot turn left with angle 45°；3: robot turn right with angel 20°；4: robot turn right with angle 45°.

Based on domain experience, we construct a robot initial obstacle avoidance decision table S=(U,A,V,f), incorporate into reinforcement learning process as robot initial prior knowledge. There are forty objects in initial decision table, part of table as in Table 1.

The table is inconsistent according to definition 2 in section 3.1. To reduce attributes of initial decision table, we use the method improved by Skowron's discernibility matrix that was proposed by Zhi Tao [12]. Those redundant condition attributes {Disgoal, Disahead} are removed, then the same records are incorporated as in Table 2.The objects of table decrease from 40 to 12.

In Table 2, it's clear that not all decision rules are necessary for the decision algorithm, some rules can be deleted and the decision process not be affected. After using the attribute value reduction method in paper [15], the simplest rule complete table is shown in Table 3.

Table 1 Initial decision table for obstacle avoidance

U	Disgoal	Disleft	Disright	Disahead	D
1	0	1	1	1	0
2	0	1	2	1	0
3	0	2	1	1	0
4	0	2	2	1	0
5	0	1	1	1	1
6	0	1	1	2	1
7	0	1	2	1	1
8	0	1	2	2	1
9	0	1	1	1	2
10	0	1	1	2	2
11	0	1	2	1	2
12	0	1	2	2	2
13	0	1	1	1	3
…	…	…	…	…	…
40	3	2	1	2	4

Table 2 Decision table without abundant attributes

U	Disleft	Disright	D
1	1	1	0
2	1	2	0
3	2	1	0
4	2	2	0
5	1	1	1
6	1	2	1
7	1	1	2
8	1	2	2
9	1	1	3
10	2	1	3
11	1	1	4
12	2	1	4

Table 3 Table of minimal decision rules

U	Disleft	Disright	D
1	1	1	0
2	1	2	0
3	2	_	0
4	1	_	1
5	1	1	2
6	1	2	2
7	_	1	3
8	1	1	4
9	2	1	4

where, "_" denote the condition attribute which can be omitted.

4.5. Navigation algorithm based on Reinforcement learning and rough sets

A hybrid method involving rough sets and reinforcement learning can be conducted as follows:

Step1 setting the robot initial and goal position, initialize the learning parameters.

Step2

(1) Detect in current position. Discrete distance s_1, s_2, s_3, s_4, get current state s.

(2) According to action selection policy, select an action from action sets.

(3) If current state s belongs to condition attributes of obstacle avoidance decision table, then the corresponding action is selected by probability equal to the forbidden action sets, goto (2). Otherwise goto (4).

(4) Update the robot's position, get new learning state s' and receive instantaneous reward.

(5) Update the $Q_{t-1}(s_t,a_t)$ value based on the iteration equation:

$$Q(s_t,a_t)=(1-\alpha_t)Q_{t-1}(s_t,a_t)+\alpha_t[r_t+\gamma\max_{a\in A}Q_{t-1}(s_{t+1},a)] \quad (7)$$

(6) Check if the robot collide with the obstacle, if true, then keep Q(s,a) , goto (1), else goto (2).

(7) Check if the robot reach the goal, if true, halt; else goto (2).

5. Simulation results

In TeamBots, we implement the algorithm based on reinforcement learning and rough sets versus traditional reinforcement learning. In the following all experiments, the learning rate α is set to 0.5, γ is set to 0.9, and use Boltzmann exploration, the temperature T is set to 5 which decreases over time to decrease exploration. The TeamBots simulation environment is as Figure 1.

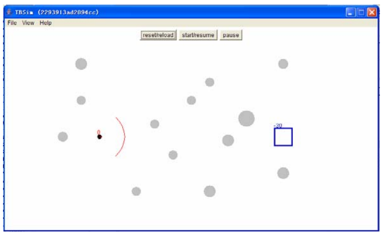

Figure 1 TeamBots simulation

The robot is in start position, square denotes goal and gray circles denote obstacles in environment.

A learning cycle of robot is defined as the process between start position and goal position. At the beginning of learning, we train robot in a simple environment using hybrid method based Q learning and rough sets versus tabular Q learning respectively. Figure 2 and Figure 3 show the relative performance of two methods in the same environment after 150 learning cycles.

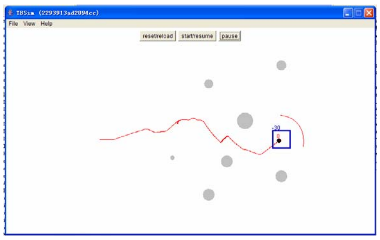

Figure 2 Tabular Q leaning in simple environment

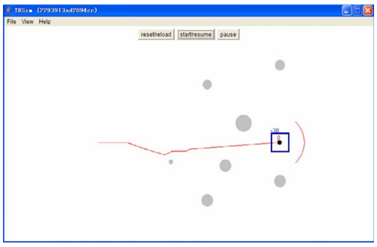

Figure 3 Hybrid learning methods in simple environment

The results demonstrate the algorithm we proposed is more effective than tabular Q learning in reducing learning time. Then we increase the complexity of environment, after 300 learning cycles, the effect of two methods is show in Figure 4 and Figure 5. The method we proposed shows better adaptability when environment changes and the learning speed is faster than tabular Q learning.

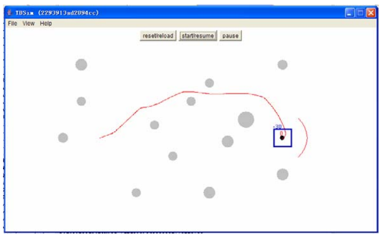

Figure 4 Tabular Q learning in complex environment

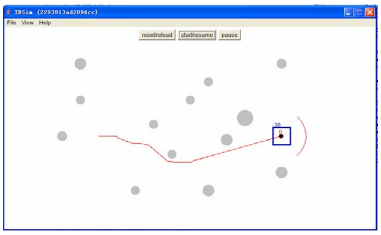

Figure 5 Hybrid learning method in complex environment

To compare the ability of avoiding obstacles of two methods in complex environment shown in Figure 4 and Figure 5, we calculate the accumulative number of collisions, while training robot under 300 learning cycles, as Figure 6:

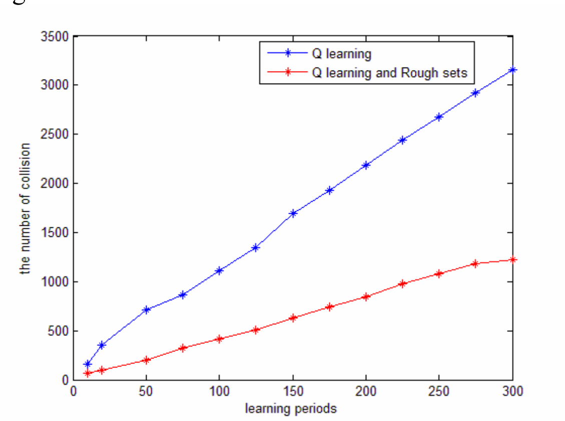

Figure 6 Obstacle avoidance of two methods in complex environments

From the graph above, with the same environment and learning cycles, we conclude that the hybrid method based on Q learning and rough sets has greatly reduced the probability of collision versus tabular Q learning. The hybrid method effectively improves the ability of obstacle avoidance of robot when learning.

6. Conclusions

This paper proposed a new method of robot navigation based on reinforcement learning and rough sets. An initial obstacle avoidance decision table is constructed, then we reduce redundant attributes by rough sets. The simplified decision table is used as robot's prior knowledge. The simulation results show the algorithm proposed in this paper not only simplifies the complexity of navigation system but also accelerates the process of learning. In addition, the results also demonstrate that the hybrid method has better avoidance obstacle capability than tabular Q learning, especially when environment becomes complex. Therefore, the hybrid method we proposed has better performance when the real-time is emergent required.

References

[1] Smart W.D, Kaelbling L.P. Effective Reinforcement Learning for mobile robots, Proceeding of the IEEE International Conference on robotics and Automation, Washington D.C. 2002, pp 3404-3410.

[2] Woo Y.K, Sanghoon L, Hong S. A Reinforcement Learning Approach Involving a Shortest path Finding Algorithm[C]. IEEE/RSJ Intl. Conference on Intelligent Robots and Systems LasVegas. Nevada. 2003, 10:436-441.

[3] Arthur Plínio de S. Braga, Aluizio F. R. Araújo: Influence zones: A strategy to enhance reinforcement learning. Neurocomputing , 2006,70(1-3): 21-34.

[4] Steven D.W, Lin L J. Reinforcement learning of Non-Markov Decision Processes. Artificial Intelligent, 1995, 73:271-306.

[5] Kevin R. Dixon, Richard J. Malak, Pradeep K. Khosla. Incorporating Prior Knowledge and Previously Learned Information into Reinforcement Learning. Technical report, Carnegie Mellon University, Institute for Complex Engineered Systems, 2000.

[6] Meiping Song, Guochang Gu, Rubo Zhang. Adaptive action fusion method for mobile robot, Journal of Harbin Engineering University, 2005,26(586-590).

[7] Pawlak Z. Rough Sets [J]. International Journal of Information and Computer Science, 1982, 11 (5) : 341-356.

[8] Kaelbling L.P, Littman M.L, Moore A.W. Reinforcement learning : A Survey[J]. Machine learning. 1988, 3(1):9-44.

[9] Sutton R.S, Barto A.G. Reinforcement learning:An Introductin[M].MIT Press, Cambridge, MA, 1998.

[10] Watkins C, Dayan P. Q-learning. Machine Learning,1992,8:279-292.

[11] Skowron A, Rauszer C. The discernibility matrices and functions information systems[M]//Intelligent Decision Support hand book of Application and Advances of the Rough Sets Theory. Dordrecht:Kluwer Academic Publisher,1991:331-362.

[12] Zhi Tao, Qingzheng Liu, Weimin Li. Algorithm for attribute reduction based on reinforcement learning on improved discernibility matrix. Computer Engineering and Application , 2007,43(32):83-85.

[13] Michie D, Chambers R.A. Box: An experiment in adaptive control[M]. Machine intelligent,1974, 137-152.

[14] Tucker B. Behavioral diversity in learning robot teams:[Ph.D.thesis]. Atlanta: Georgia Institute of Technology, 1998.

[15] Chengdong Wu, Ying Zhang, Mengxin Li. A rough set GA-based hybrid method for mobile robot, International Journal of automation and computing 2006,29-34.