Statistics Based Q-learning Algorithm for Multi-Agent System and Application in RoboCup

Ya Xie
Hunan Institute of Engineering, Xiangtan, China
Email:644051287@qq.com

Zhonghua Huang
Hunan Institute of Engineering, Xiangtan, China
Email: csu_xieya@126.com

Abstract—**This paper proposes statistic learning based Q-learning algorithm for Multi-Agent System, the agent can learn other agents' action policies through observing and counting the joint action, a concise but useful hypothesis is adopted to denote the optimal policies of other agents, the full joint probability of policies distribution guarantees the optimal action choice to the learning agent. The algorithm can also improve the learning speed because the conventional Q-learning space is cut from exponential one to linear one. The convergence of the algorithm has been proved; the successful application of this algorithm in the RoboCup shows its good learning performance.**

Index Terms—**Q-learning, Statistics, Multi-agent, RoboCup**

I . INTRODUCTION

Reinforcement Learning of Multi-Agent leads the traditional reinforcement learning technology into MAS (Multi-Agent System). Reinforcement learning is unsupervised learning, it is a kind of adaptive learning method with environmental feedback as input, which obtains the optimal strategy through interaction with the environment and continuous improvement strategy finally [1][2]. Because of its online learning and adaptive learning characteristics, reinforcement learning is an effective tool to solve the optimization problem for MAS, which is widely used in various fields and has become one of the research hotspots in the field of machine learning. Reinforcement learning algorithm in mature are: TD, Q learning, Dyna-Q and Q-PSP learning etc[3][4], this paper chooses Q learning as learning foundation due to model independent, simplicity and efficient characteristics of Q learning.

The MAS is very complex because of mutual influence between agents, Markov Model can't be used in MAS, the reinforcement learning algorithms based on Markov model can't be introduced into MAS directly, and therefore the reinforcement learning algorithm should be improved.

Firstly, the environment mode of reinforcement learning should be improved, the current state can be changed through personal actions for learning agent in MAS, but because of condition changes by other agents, the closure of system is lost, the successor state for learning agent is not inferable, successor state is not only decided by the current state that is defined S and action of learning agent that is defined a, that is to say, MAS is nondeterministic Markov System. For reinforcement learning of multi-agent, the return function and successor status function can't use $r(s, a)$ and $S'=\delta(s, a)$ to express.

Secondly, other agent's strategy should be considered for strategies selection of learning agent in MAS, the changes from current state to next state aren't all decided by actions of learning agent, other agents also choose actions which change system state, the uncertainty of successor function is caused by unknown of other agent's strategy[5][6]. In most cases, the other agent's behavior is not random, but can be considered as action strategy of probability distribution, which is random behavior that is subject to a certain probability distribution in a certain state. In Robocup as an example, a player can't predict the opponent's intentions accurately, but the actions of opponent can be judged through knowledge and observation: when our ball, the close opponent will intercept, attack will not be chosen, usually a defensive player will probably choose intercept larger and probably choose attack smaller, which illustrates that the choice of other agent's strategy obeys a certain probability distribution, the probability distribution and actions can be partially determine according to the prior knowledge and current state. So the other agent's behavior strategy and effect of environment can be learned through observation and statistics of other agent's behavior in learning process, at the same time, the return function $r(s, \vec{a})$ and successor status function $S'=\delta(s, \vec{a})$ are also determined. Therefore, the statistics method is introduced into learning algorithm, other agent's behavioral strategies are learned implicitly through the statistics of state and the motion vector, which can solve the above two problems effectively.

doi:10.4304/jsw.9.3.634-640

II. Algorithm of Statistics based Multi-Agent Q Learning

The learning goal is learning strategy algorithm that can be defined as π: $S \rightarrow A$, finite state sets as $S = \{S_i\}$, agent action sets as $A = \{a_i\}$, so this strategy describes the probability distribution as $\{P_1, P_2, \ldots P_i\}$ of selection action according to the current state as $s \in S$, from the state S_t began, the expected discount return as V^{π} according to the strategy π is as follows.

$$V^{\pi}(s_t) = E(\sum_{i=0}^{\infty} \gamma^i r_{t+i}) \tag{1}$$

Where $0 \le \gamma < 1$ is the discount factor, which reflects the choice between the current return and future return in the total return, r_t refers to bounded return every time, maximum strategy of formula (1) is obtained by optimum strategy π^*, that total return plus the statistical expectation computing in undetermined Markov environment.

The motion vector $\vec{a}$ is introduced in order to describe the behaviors of agent in the environment state; the Q learning algorithm is improved as follows.

$$\begin{aligned} Q(s,\vec{a}) &= E[r(s,\vec{a}) + \gamma V^*(\delta(s,\vec{a}))] \\ &= E(r(s,\vec{a})) + \gamma E[V^*(\delta(s,\vec{a}))] \\ &= E(r(s,\vec{a})) + \gamma \sum_{S'} P(s'|s,\vec{a}) V^*(s') \end{aligned} \tag{2}$$

The replace of formula (2) is as follows:

$$Q(s,\vec{a}) = E(r(s,\vec{a})) + r \sum_{s'} P(s'|s,\vec{a}) \max_{a'} Q(s',\vec{a}') \tag{3}$$

Where $P(s'|s,\vec{a})$ is the probability of successor state s' after joint motion vector of agent as $\vec{a} = (a_1, a2 \ldots a_i)$ under state s, where $\vec{a}'$ is joint action vector of agent in new state s'.

$\hat{Q}_t$ is approximation of Q value after t iterations, and then the Q value can be obtained by following iterative process as formula (4).

$$\hat{Q}_{t+1}(s,\vec{a}) \leftarrow (1 - a_t)\hat{Q}_t(s,\vec{a}) + a_t[r_t + y \max_{\vec{a}'} \hat{Q}_t(s',\vec{a}')] \tag{4}$$

Where α_t is a dynamic learning rate, π^*_1 is the best strategy of learning agent, $\hat{\pi}_t^i$ is the strategy approximation of agent i in t moment, which is t vector of overall strategy $\hat{\pi}_i$ for agent i, then formula (5) is as follows.

$$\begin{aligned} \hat{Q}_{t+1}(s,\vec{a}) \leftarrow (1 - a_t)\hat{Q}_t(s,\vec{a}) + \\ a_t[r_t + \gamma \max_{\vec{a}'} \sum \hat{\pi}_t^i \hat{Q}_t(s',\vec{a}')] \end{aligned} \tag{5}$$

Formula (5) is statistics based Q-learning algorithm for Multi-Agent system, the actions and state can be learned independently in MAS, the convergence and validity of the algorithm are analyzed in the following.

III. Analysis of Algorithm Convergence

A. The Definition and Proof of Lemma

Lemma 1-1: learning rate $a_t = \frac{1}{1 + \beta C_t(s,\vec{a})}$, where $C_t(s, \vec{a})$ indicates the occurrence number of state action $(s, \vec{a})$ in the t learning process, β is a constant.

The selection of learning rate α_t is based on the following idea: for state action $(s, \vec{a})$ with more appearance, last Q value is considered more because of numerous iterative approximation of Q value; The subsequent learning effectiveness is considered for state action $(s, \vec{a})$ with less appearance, $\alpha_t = \frac{1}{1 + \beta C_t(s,\vec{a})}$ is chosen, where $C_t(s, \vec{a})$ is the occurrence number of state action $(s, \vec{a})$ in the t learning process, the influence of statistics $C_t(s, \vec{a})$ is increased by introducing parameter $\beta \ge 1$, the proper value of parameter β will accelerate the convergence speed in the learning process (the following discussion will show that β can't affect convergence of learning).the modification amount of Q value for each iteration is weakened with the increase of $C_t(s, \vec{a})$ and $\alpha_t \rightarrow 0$, which makes the learning process gradually become stable.

In fact, the selection of learning rate α_t won't influence the convergence of the learning method; the hypothesis is that $\max_{\vec{a}'} \sum \pi_1^* \prod_{i=2}^{n} \hat{\pi}_t^i \hat{Q}_t(s',\vec{a}')$ part in formula (5) is equivalent to $\max_{\vec{a}'} \hat{Q}_t(s',\vec{a}')$ part in formula (4). Apparently, sequence $(\alpha_t \ldots)$ is an incomplete geometric progression, $\sum_{t=1}^{\infty} \alpha_t = \infty$, $0 < \alpha_t < 1$, but sequence $(\alpha_t^2 \ldots)$ is convergent as $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$, the selection of learning rate a_t is valid, which is indicated by $\hat{Q}_t(s,\vec{a})$ approach $Q(s,\vec{a})$ with $n \rightarrow \infty$ in Q learning for an undetermined Markov environmental when learning rate satisfies this condition.

(2) Lemma 1-2: The optimal action learning of other agent is the learning of maximum likelihood strategies.

$\max_{\vec{a}'} \sum \pi_1^* \prod_{i=2}^{n} \hat{\pi}_t^i \hat{Q}_t(s',\vec{a}')$ part in formula (5) describes combined action probability of agents in the strategy π_1^* and estimation strategy $\hat{\pi}_i$ of other agents, which decides the probability distribution of selection for action $\vec{a}'$ in the new state s', behavior strategy can be attributed to the action choice ultimately, $\hat{\pi}_t^i$ reflects the action choice of agent i in the current state, then the choice strategy of agent i has the greatest probability of action.

Supposing that the number of action that can be chosen by agent i is m, then the probability vector of strategy $\hat{\pi}_t^i$ is $(1/m, 1/m, \ldots)$ considered by learning agent i initially, which the agent i chooses all actions with equal probability,

the probability vectors need to be updated through the observation learning of strategy of agent *i* by learning agent continuously along with learning process development, the update rule is as follows.

Rule 1-1. $\hat{\pi}_t^i = (x_1/m, x_2/m, x_3/m, \ldots)$ represents the action probability vector of agent *i* at the moment *t*, if agent *i* chooses action *j* at the moment *t+1* through observation, then the formula (6) is obtained.

$$\hat{\pi}_{t+1}^i = [x_1/(m+1), x_2/(m+1), \ldots, x_{j+1}/(m+1), \ldots] \quad (6)$$

The proof is as follows.

$\sum_{k=1}^{m} x_k = m$ *at the moment t+1,* $x_1+x_2+\ldots+x_j+1+\ldots = m+1$ *at the moment t*+1, and $x_j < m \Rightarrow x_j+1 < m+11$, then the component of probability vector update is less than 1.

Different action with equal probability at the early learning stage in Rule 1-1 reflects that the learning agent is unknown to strategy model of other agent, learning agent learns other agent's strategy $\hat{\pi}$ through statistics of state joint action $(s, \vec{a})$ and rule 1-1, the probability component of high frequency action is increased ceaselessly that is guaranteed by update rule, but also to ensure that the low frequency action has chosen probably (the probability components is not 0, which is like ε greedy selection strategy, but the ε is incremented), so the strategy learning of learning agent to other agents has a certain redundancy error and flexibility.

Lemma 1-3: *a* indicates that agent *i* doesn't choose the best action for *a* times in learning for *k* times, then $\exists n \in N$, and $a < n$, also *a* is bounded.

According to the above discussion, that the other agent's strategies can be learned, the other agent's strategies will converge to the optimal strategy, then $\hat{\pi}_i \to \pi_i^*$, $\hat{\pi}_i$ represents estimation strategy, π_i^* and represents the best strategy. Considering $\hat{\pi}_t^i$ is the *t* vector of $\hat{\pi}_i$ only, then $\lim_{k\to\infty} |\hat{\pi}_{tk}^i - \pi_{tk}^{i*}| = 0$ which $\hat{\pi}_{tk}^i$ and π_{tk}^{i*} represent estimation strategies and optimal strategy of the *k* second iterative process of agent *i*, *k* represents the number of learning iterations, $\hat{\pi}_{tk}^i = [x_1/(m+k), x_2/(m+k), \ldots (k+1-a)/(m+k), \ldots]$, where *a* expresses that agent i doesn't choose the best action for *a* times in *k* times learning, which sets the j movement, then $x_j = k+1-a$, $\hat{\pi}_{tk}^i = [y_1/(m+k), y_2/(m+k), \ldots (k+1)/(m+k), \ldots]$, then we can get the following formula (7).

$$\lim_{k\to\infty} |\hat{\pi}_{tk}^i - \pi_{tk}^{i*}| = \lim_{k\to\infty} [\frac{(x_1 - y_1)^2}{(m+k)^2} + \ldots + \frac{(k+1-a-k-1)^2}{(m+k)^2} + \ldots]^{1/2}$$

$$= \lim_{k\to\infty} [\frac{\sum_{i=1}^{j-1}(x_i - y_i)^2 + \sum_{i=j+1}^{m}(x_i - y_i)^2 + a^2}{(m+k)^2}]^{1/2}$$

$$= \lim_{k\to\infty} [\frac{\sum_{i=1}^{m}(x_i - y_i)^2 + a^2}{(m+k)^2}]^{1/2} \text{ where } x_j = y_j = 0$$

$$= \lim_{k\to\infty} [\frac{\sum(\frac{x_i - y_i}{k})^2 + \frac{a^2}{k^2}}{(1+\frac{m}{k})^2}]^{1/2} = 0 \quad (7)$$

Apparently, the denominator of formula (7) tends to 1 with $k \to \infty$, if molecular approaches 0, then formula (7) is near to 0, but the two parts of molecular are greater than 0, so each part of molecular should approach 0, so $\exists\ n \in N$, then $a < n$, we can get that *a* is bounded

B. Certification of Algorithm Convergence

The maximum *Q* value is obtained by search options of all combined action $\vec{a}'$ in the new state s' for the $\max_{\vec{a}'} \hat{Q}_t(s', \vec{a}')$ part in formula (4), which ensures the convergence of the Q learning. $\max_{\vec{a}'} \sum \pi_1^* \prod_{i=2}^{n} \hat{\pi}_t^i \hat{Q}_t(s', \vec{a}')$ part in formula (5) only searches agent's action space $|A|$; the greatest probability action in strategy $\hat{\pi}_t^i$ is chosen for other agent, namely action of the maximum likelihood estimate. That the learning search under certain conditions is equivalent to full search of the combined action space $|A|^n$ will be demonstrated below.

$\max_{\vec{a}'} \sum \pi_1^* \prod_{i=2}^{n} \hat{\pi}_t^i \hat{Q}_t(s', \vec{a}')$ part in formula (5) expresses selection on action space $|A| = m$, setting $p_1 = 1/m$, which learning agent adopts the blind search strategy for all other's action, then the action of maximum probability in strategy probability vector is chosen for agent i, selection probability $p_i = \frac{k+1-a_i}{m+k}$ according to lemma 1-2, where a_i expresses agent *i* didn't choose this action for a_i time in *k* time learning iteration, the search of joint action space(the optimal action) with the probability is as follows.

$$P = \sum_{l=1}^{m} \frac{\prod_{i=2}^{n}(k+1-a_i)}{(m+k)^{n-1}} p_1 \quad (8)$$

Where *P* expressed full probability of joint action selection in the learning algorithm, *k* is the number of learning iterations, then the learning search is equivalent to space full search of $|A|^n$ is evidenced by as long as that the proof of *P* with the increase of *K* approaches 1(ensuring optimal action selection from utility actually), which the convergence of the learning method is evidenced, then formula (8) is transformed as follows:

$$P = \frac{\prod_{i=2}^{n}(k+1 \quad a_i)}{(m+k)^{n\ 1}} \frac{m}{m} = \frac{\prod_{i=2}^{n}(k+1 \quad a_i)}{(m+k)^{n\ 1}} \qquad (9)$$

Set $a = max\ a_i$, the hypothesis is that the frequency of optimal action that didn't be chosen by agent in k times learning iteration is all maximum.

$$P > \frac{(k+1-a)^{n-1}}{(m+k)^{n-1}} = [\frac{1-\frac{a-1}{k}}{1+\frac{m}{k}}]^{n-1} \qquad (10)$$

The formula(10) is passed to the limit, denominator approaches 1 when $k\to\infty$ in $\lim_{k\to\infty}[\frac{1 \quad \frac{a \quad 1}{k}}{1+\frac{m}{k}}]^{n-1} = 1$ part , then a is bounded according to lemma 1-3, so $\lim_{k\to\infty}[\frac{1 \quad \frac{a \quad 1}{k}}{1+\frac{m}{k}}]^{n-1} = 1$ where $(a-1)/k \to 0$ and numerator tends to 1 , formula(11) as follows according to limit squeeze rule.

$$1> \lim_{k\to\infty} P> \lim_{k\to\infty}[\frac{1 \quad \frac{a \quad 1}{k}}{1+\frac{m}{k}}]^{n-1}=1 \Rightarrow P=1 \qquad (11)$$

P approaches 1 with increase of K according to formula (11), the learning search is equivalent to the full search of space as $|A|^n$, then the learning algorithm is convergent, the learning rate in formula (5) is chosen effectively according to lemma 1-1, Q-learning is convergent by formula (5).

IV. Analysis of Algorithm Effectiveness

A. Analysis of Algorithm Error

Action selection of other agent is based on the maximum probability of strategy $\hat{\pi}_t^i$ in algorithm, the hypothesis is that the best action is action of maximum probability for agent i at the moment t, if the probability of strategy $\hat{\pi}_t^i$ is error, the selection of joint action will abandon, which will affect the validity of the algorithm, for the error analysis of probability vector $\hat{\pi}_t^i$ in the learning process, set m actions that agent i can choose, m components corresponding to probability vector $\hat{\pi}_t^i$, then the probability of action selection strategy in the k time learning iteration is as follows(the action is observed $l \ge k/2$ times at least)

$$\sum_{j=l}^{k} C_k^j (\frac{1}{m})^j (\frac{m \quad 1}{m})^{k\ \ j} \text{, where } l \ge k/2 \qquad (12)$$

Then the learning space is as follows:

$$|H| = [\sum_{j=l}^{k} C_k^j (\frac{1}{m})^j (\frac{m \quad 1}{m})^{k\ \ j}]^{-1} \text{, where } l \ge k/2 \qquad (13)$$

According to PAC guidelines, when the learning times (sample size) K meets $K \ge (\ln|H| - \ln\delta)/\varepsilon$, this hypothesis (here for the best selection action) is learned by learning agent with (1- δ) probabilities at least, and the generalization error of this hypothesis is less than ε. The learning time k is realized through instantiation of corresponding parameter, set $m = 10$, $\delta = 0.01$ and $\varepsilon = 0.05$ in action space, then $k \ge 112$, the hypothesis is learned with 0.99 probability and the error is less than 0.05 when the learning time is less than 115, therefore when the learning times is larger (for example, greater than 115), the learning of strategy $\hat{\pi}_i$ will achieve very good results (probability vector error is very small) , abandon probability of joint action choice is also very small.

B. Discussion of the Feasibility of Algorithm

The requirements of $\max_{\vec{a}'} \hat{Q}_t(s', \vec{a}')$ part in formula (4) are the selection of all combined actions in new state s', a Q- learning algorithm for multi-agent is given in Michael.W text, each action can be chosen ensured by total probability distribution of joint action in this algorithm, for MAS with n agents, action selection space of each agent is $|A|$, learning search space for each state is index space $|A|^n$, so when the n and $|A|$ increase, learning efficiency will drop sharply, $\max_{\vec{a}'} \sum \pi_1^* \prod_{i=2}^{n} \hat{\pi}_t^i \hat{Q}_t(s', \vec{a}')$ part in formula (5) only search action space $|A|$ of learning agent, the action with greatest probability in strategy $\hat{\pi}_t^i$ is chosen for other agent, also is the action with maximum likelihood estimate, so the total learning search space is a linear space $|A|$, which effectively reduces the complexity of the algorithm. In fact, because the algorithm for selection strategy of joint action is very simple, so the learning algorithm has a better performance according to the Occam 's razor theorem.

The joint probability of action under the strategy π^*_1 of learning agent and estimation strategy $\hat{\pi}_1$ of other agent is described in the $\pi_1^* \prod_{i=2}^{n} \hat{\pi}_i$ part in formula (5), which decides the probability distribution of selection action $\vec{a}'$ in the new state s′, it should be noted here, because the motion vector $\vec{a}$ is composed of multiple-agent decision, the realization of search strategy is also dependent on other agent's behavior for learning agent, further, if other agent's strategy satisfies: $\lim_{t\to\infty} P(|\hat{\pi}_i - \pi_i^*| > \varepsilon) = 0$, namely that the other agent's strategy is convergent, the strategy model can be obtained by learning agent after observing repeatedly, convergence strategies is unknown for learning agent, in this case, the joint probability between agents can ensure search of the whole problem space, which also guarantees convergence of multi-agent Q-learning algorithm

according formula (5). The action selection through trial an error for the learning agent, at the same time, the statistics and learning of other agents' strategy action in the beginning learning process, with the development of the learning process, the learning agent is familiar with other agents gradually and can establish its effective strategy model with relevant knowledge, the strategies mutation of other agent (may be caused by the unexpected behavior) after learning many times is given only small probability of recognition, the large probability events is the main goal of learning, so learning in undetermined Markov environment according to formula(5) is suitable.

It should be noted here, because of problem solving in a larger space for MAS, the *Q* function expressed by *Q* value will cause the dimension disaster, in order to achieve generalization of enforcement learning in mass or a continuous state space, neural network is used as unction approximation of reinforcement learning.

V. The Application of Statistics Based Q-learning Algorithm in RoboCup

RoboCup provides a fully distributed control, real-time asynchronous multi-agent environment, RoboCup is carried out in the standard computer environment, using Client/Server, by the RoboCup association provides a standard Server, the team writes each client (Client) program, simulation of the actual football team[7][8].

The agent in Robocup should have some basic skills such as kicking the ball, interception, drag the ball, but the team as a whole should also have high tactical strategy, which will not only focus on the players themselves, but also to consider how to cooperate with other team members and confrontation[9]. High-level strategy of each agent can be considered as the optimal control strategy, the agent chooses the corresponding behavior according to the strategy. Because of the complexity of the multi-agent environment, manual preparation of high-level strategy is inefficient and sometimes even impossible, the high-level strategy is learned by agent through learning techniques[10].

A. The learning of Defensive Strategy

In the game when a player obtained the ball, it can choose three actions: dribbling, passing, shooting. The specific choice should be decided based on the current environment state, because the race is a continuous process, the environmental state space is huge, in order to describe the current state, the state discretization method is adopted, so partition on the playing field, according to the designer's experience, 9 zones is divided according to the important degree of defense, which is shown in Fig. 1.

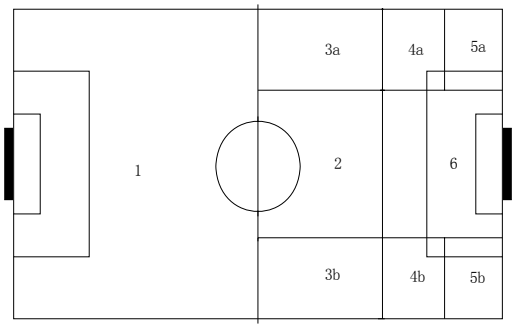

Figure 1. Site zoning in RoboCup

In this paper, for the defender (right of defense, 1 district except), the off-line learning of action strategies is underway through statistics based Multi-Agent Q-learning. Using the following characteristics to describe the current state of *s*: the coordinates of player; the coordinates of opponent in radius R which is described as *op[i], 2<R<18, 0≤i<7;* the coordinates of teammate in the current zone and adjacent area which is described as *p[i], 0≤i<7.*

Any time the player with the ball (defender) has three optional action: dribbling, passing, clearing the ball (the ball out of play), which of course should choose in legal circumstances (such as in the offside situation cannot pass), players choose any action with equal probability , then the evaluation criteria is given for every selected action, for dribbling, if the successful breakthrough of competitors and there is no opponent within a radius of 2 meters, the action choice is correct, then to return the positive; for passing, if the teammate obtained the ball successfully and there is no opponent within a radius of 1.5 meters, then to return the positive; for clearing the ball, if the ball is out of bounds in the middle of about 5 meters, then to return the positive, so that continuous learning, according to the state, action and reward, Q values can be calculated, neural network is adopted as approximator for Q value, for each action, the neural network with same structure (currently 3) to approximate the Q value network with 30 input nodes, 45 hidden nodes, one output node, for state as input, the output is the Q value, The learning algorithm is as follows.

The algorithm begins.

(1) The first step. Initialization of strategy π^*_I={0.33, 0.33, 0.33} and $\hat{\pi}_i$={0.25, 0.25, 0.25, 0.25} , arguing that the agent selected action with equal probability (three actions), and other offensive agent can also choose shooting action (a total of four movements);

(2) The second step. Initialization of neural network;

(3) The third step. Modification of $\hat{\pi}_I$, $Vt_t(s, \vec{a})$ and α_t according to the observed joint action $\vec{a}$;

(4) The forth step. The calculation of Q value and training the network;

(5) The fifth step. If reaching the training accuracy, the learning algorithm is end.

Then, the algorithm terminates.

The whole learning process takes about 10000 times training. The learning process can be underway automatically through coaching process, the training process is as follows.

(1) The first step. 8 defenders are placed randomly on the site (zone 1 is except and 3 strikers is not defense);

(2) The second step. 7 offensive players are placed randomly;

(3) The third step. The defender with the ball (selection in number 2, 3, 4, 5 randomly) chosen an action randomly;

(4) The forth step. Other teammate chosen to attack or defense according to respective strategy;

(5) The fifth step. The coach process is responsible for the initialization and end of training, record the current state, the choice and return of players;

(6) The sixth step. Learning is done according to the training results in accordance with the above learning algorithm.

View from the table 1, the high-level strategy of improved team is strengthened and optimized, the rate of successful interception, passing and controlling the ball between teammates has been greatly improved, defensive ability has been greatly improved, and only 1 ball is lost in 5 games.

TABLE 1.

THE PERFORMANCE STATISTICS BETWEEN IMPROVED TEAM AND ORIGINAL TEAM

score	rate of successful passing ball	rate of successful interception ball	ball control rate
3 :0	55%	99%	68%
2:0	54%	99%	66%
3 :0	53%	99%	68%
3:1	54%	99%	70%
5:0	58%	99%	75%

B. Front-court Confrontation and Cooperation Strategy Learning

As shown in Fig. 2, the attack between offensive team with three agent and defensive team with four agent, the goal of offensive team is score a goal, keep control of the ball, and do not let the ball out of bounds, No. 9 is an offensive player that uses statistics based Q-learning algorithm to learn high-level strategy under this scenarios, we define the offensive player from the initial position to shoot success or outside of the ball as a scenario, and also, define the reward value of successful and unsuccessful behavior, if the action of player leads to good results, then the action is successful behavior, otherwise is not successful behavior, where score a goal, surpass are good results, but lost the ball, outside of the ball are not good results.

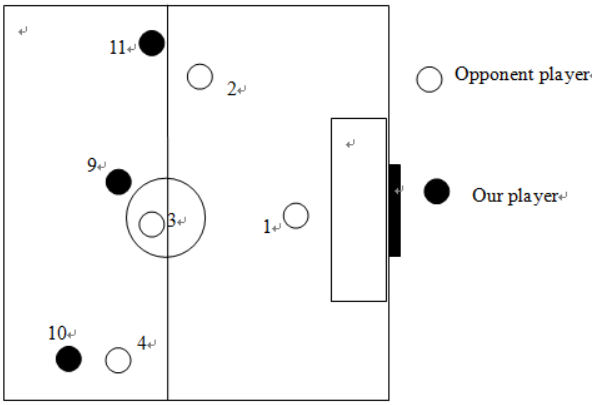

Figure 2. The training scenarios of agent

Fig. 3 is the performance curve of the learning agent, the number of unsuccessful action in 100 experiments is adopted as the performance index (ordinate), the abscissa is the number of iterative learning control, comparing to the traditional Q algorithm in the same period, view from Figure 3, the action of strategy of agent with the adoption of this learning algorithm after some learning steps is better than the strategy before unlearning; the learning algorithm can be convergent in the 4000 learning steps, while the Q learning algorithm can not be convergent after 6000 learning steps. (in fact, the Q learning algorithm can not be convergent after 10000 learning steps, this shows that the traditional Q learning cannot guarantee the convergence of learning in multi-agent environment), a funny thing is that if all agents use this learning algorithm, then how is their performance? This is another research direction.

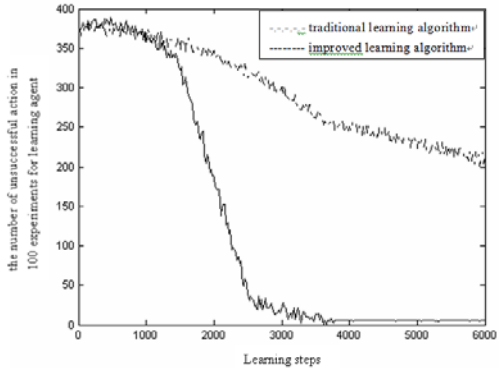

Figure 3. The performance curve of the learning agent

VI. CONCLUSIONS

A new learning algorithm based Q-learning is proposed for MAS in undetermined Markov environment. The problems of partly perception, uncertain information and strategy learning of other agent are solved effectively in MAS through organic fusion of statistical learning and reinforcement learning, the other agent's strategy is learned by statistics of joint action, and the choice of joint optimal action is guaranteed through full probability distribution of probability vector of strategy, which ensures the convergence of the algorithm in theory, the index space of learning space is lower to the linear space, which improves the learning efficiency effectively.

Environment parameter selection, the agent setting and training process of training is similar to the previous chapter, here no longer repeated discussion, Because the training scenes is simple, so the number of training is less, a total of about 6000 training.

In the design of learning model, the M-Dyna-Q algorithm proposed by Weiss is used to construct and improve the planning part, 3 layer feed forward neural network is used as the prediction part, the SOM network (the network node is determined by the state parameter) is adopted as the state schema mapping.

REFERENCES

[1] Sutton R S. Open theoretical questions in reinforcement learning. *Computational Learning Theory*. London, UK: Springer, 1999. 11-17.

[2] Yi Wan, ChengWen Wu. A New Intelligent Model for Structural Reliability Identification Based on Optimal Machine Learning. *Journal of Computers*. Vol 7, No 2, 2012, 362-370.

[3] Sun R., Sessions C. Extracting plans from reinforcement learners. *Proceedings of the 1998 International Symposium on Intelligent Data Engineering and Learning*. New York, USA: Springer-Verlag, 1998. 243-248.

[4] Bing Jia, Yongjian Yang, Jun Zhang. Study on Learner Modeling in Adaptive Learning System. *Journal of Computers*. Vol 7, No 10, 2012, 2578-2584.

[5] Stone P. Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer. *Cambridge, MA: MIT Press*, 2000.

[6] Tian-Wei Sheu, Tzu-Liang Chen, Jian-Wei Tzeng, Ching-Pin Tsai, Masatake Nagai. Study on the Conception of Learning Problems of Students by Combining the Misconception Domain and Structural Analysis Methods. *Journal of Computers*. Vol 8, No 5, 2013, 1247-1254.

[7] Michael W., Jeffrey S R. Best-Response Multiagent Learning in Non-Stationary Environments. *AAMAS'04*, 2**,** July 19 - 23, New York, USA: 2004.

[8] Yanan Zhang, Shifei Ding, Xinzheng Xu, Han Zhao, Wanqiu Xing. An Algorithm Research for Prediction of Extreme Learning Machines Based on Rough Sets. *Journal of Computers*. Vol 8, No 5 ,2013, 1327-1334.

[9] Weiss G. A multiagent variant of Dyna-Q. *In 4th International Conference on Multi-Agent Systems.Boston*, Massachusetts*: IEEE Computer Society,* 2000. 461-462.

[10] Takahashi Y., Edazawa K., and Asada M. Multi-Module learning system for behavior acquisition in multi-agent environment. *In IEEE/RSJ International Conference on Intelligent Robots and Systems.* 2002, 927-931.

Ya Xie was born in Changde, China, in 1982. She earned the M.S. degrees in Application of Computer from Central South University in 2006. At the same time, she is a teacher in department of computer and communication, Hunan Institute of Engineering (Xiangtan, China) from 2010. As the first author more than 10 papers were published. Her current research interests include intelligent control and machine learning.

Zhonghua Huang: Born in Hunan Province of China in 1979, doctor graduate is achieved from Central South University in 2006, now worked in Hunan Institute of Engineering for an associate, main research fields: function control, neural network etc.