http://blog.csdn.net/app\_12062011/article/details/50673060

Abstract

With the development of the Internet, the online courses are becoming increasingly popular. There are several types of online course platform, which include online video, online quiz, distance course.

For online video, which is usually recorded by teachers, is content-rich, easy to understand, but the interaction is poor. Therefore it is suitable for simple introductory courses or lectures. However, it is not helpful for students to understanding some difficult knowledge, and apply theory into practice process.

For online quiz, students can learn knowledge by practice. However, most of the traditional online exam only provide some single, simple questions and strategy, so it is not enough for students in different learning levels. Moreover, in most of cases, the online quiz only use the standard answer for detecting whether the solution is correct, all of these disadvantages limit students' learning efficiency greatly.

For distance courses, teachers and students can interact online, and the teacher can be able to find the lack of students knowledge and answer the question of students immediately. It is also efficient to provide online testing, evaluation, feedback. But the human cost is very large in this form, it is not easy to copy extensively.

The Intelligent Teaching System(ITS) become more and more import, it provides thousands of education opportunities to people, but it still has far to go. One key to improve the ITS is to improve the Artificial Intelligent( AI) algorithm. Current algorithms are really simple, usually just records the number of correct or wrong answer, use basic regression methods of marching learning .In this thesis, we will discuss how to develop appropriate learning strategies for every student though history information of students with reinforcement learning method.

There are 7 parts of this thesis: part one is the introduction, part 2 explain the reinforcement learning algorithm and batch reinforcement learning, in part3, we apply the batch RL algorithm into a classical simple example (grid word), in part4, we apply the batch RL algorithm into real project, part 5 is we analysis the result, and give the conclusion, part 6 we discuss the disadvantage and wok we will do in the future, part 7 is the reference.

Introduction

With the development of the Internet, the online courses are becoming increasingly popular. There are several types of online course platform, which include online video, online quiz, distance course.

For online video, which is usually recorded by teachers, is content-rich, easy to understand, but the interaction is poor. Therefore it is suitable for simple introductory courses or lectures. However, it is not helpful for students to understanding some difficult knowledge, and apply theory into practice process.

For online quiz, students can learn knowledge by practice. However, most of the traditional online exam only provide some single, simple questions and strategy, so it is not enough for students in different learning levels. Moreover, in most of cases, the online quiz only use the standard answer for detecting whether the solution is correct, all of these disadvantages limit students' learning efficiency greatly.

For distance courses, teachers and students can interact online, and the teacher can be able to find the lack of students knowledge and answer the question of students immediately. It is also efficient to provide online testing, evaluation, feedback. But the human cost is very large in this form, it is not easy to copy extensively.

The Intelligent Teaching System(ITS) become more and more import, it provides thousands of education opportunities to people, but it still has far to go. One key to improve the ITS is to improve the Artificial Intelligent( AI) algorithm. Current algorithms are really simple, usually just records the number of correct or wrong answer, use basic regression methods of marching learning .In this thesis, we will discuss how to develop appropriate learning strategies for every student though history information of students with reinforcement learning method.

There are 7 parts of this thesis: part one is the introduction, part 2 explain the reinforcement learning algorithm and batch reinforcement learning, in part3, we apply the batch RL algorithm into a classical simple example (grid word), in part4, we apply the batch RL algorithm into real project, part 5 is we analysis the result, and give the conclusion, part 6 we discuss the disadvantage and wok we will do in the future, part 7 is the reference.

2,Algorithm 2500

According to traditional machine learning classification, there is no mention of reinforcement learning, and in the connectivism learning, the learning algorithm is divided into three types, unsupervised learning, supervised learning and reinforcement learning. Reinforcement learning is a kind of important machine learning method. It has many applications in the area of intelligent control robot and analysis and prediction. The key of reinforcement learning is that the intelligent system can learn from the environment directly and then choose a action, in order to get the maximum cumulative value of reward signal from the long term. The reinforcement learning is different from the supervision learning, mainly in the agent signal. In reinforcement learning, the agent signal provided by the environment is a kind of evaluation on actions’ quality (usually is a scalar signal), but not to tell reinforcement learning system (RLS) how to produce a correct action. Because the information provided by the external environment is limited, RLS has to learn from own experience. In this way, RLS obtains knowledge in an action-evaluation environment, and improves the action plan, to adapt to the environment. The main research of reinforcement learning is to make the agent to calculate the best policy from the environment.

In supervised learning or unsupervised learning, we always give a sample x, and then give or not to y label. And then apply this sample for fitting, classification, clustering or dimensionality reduction and other operations. However, for a lot of sequence decision or control problems, it is difficult to have such a regular sample. For example, the control problem of a robot, the robot even doesn’t know which action it should move at the beginning, and does not know how to find the right direction automatically in the process of moving. In addition to a chess AI design, every step is actually a decision making process. Although there are some AI heuristic methods in the simple chess, but in the complex situation, it is import to let the machine to consider a few back steps before deciding where to go in order to get good result. Therefore, it is necessary to use a better decision-making method.

There is a solution to this kind of decision problem. We design a reward function, if the learning agent (such as the robot program, chess AI program) obtains the results after a decisive step, then we return some reward to agent (for example, if the result is good, the reward of return function positive, if the result is poor, the reward of return function is negative). As for the robot, if it took a forward or close step to the goal, then the return function is positive, if it took a back or far step to the goal, the return function is negative. If we can evaluate for each step and get the corresponding reward function, it would be easy to deal with this problem. we just need to find a path whose sum of return value is maximum, which is the best path. Reinforcement learning has been successfully applied in many fields, such as automatic helicopter, robot control, mobile network routing, market decision, industrial control, high performance web page index, etc.

Next, we will introduce the Markov decision process (Markov, decision processes MDP) first.

First , we give the definition of Markov property. A state signal that succeeds in retaining all relevant information is said to be Markov, or to have the Markov property.[book]. For example, in the decision processes of chess, we define the state as the location of every chessman, the current state include all of the history information, the next station depends on the current state, but not relates to the state before current state.

A reinforcement learning task that satisfies the Markov property is called a

Markov decision process, or MDP. If the state and action spaces are finite,

then it is called a finite Markov decision process (finite MDP).

A Markov decision process is composed of a group of 5-tuple.

\* S representation of states. (For example, in an automated helicopter system, the current position consists of a set of coordinates)

\* A represents a set of actions (actions). (For example, using a joystick to control helicopter’s direction, let it forward, backward, etc.)

\* represents the state transition probability. A state in S will change to another state of S according to the action of A. Therefore the transition probability is the probability distribution from current state to next state with a kind of action of A. (the current state may jump to a lot of state with the same action).

is the discount factor.

\* , R is a return function (reward function), the return function often writing S function (only related to S), so the words, R rewriting to like this: .

MDP dynamic process is as follows: the initial state of a agent, and then from the A to select an action to perform, after the implementation, agent random transfer to the next state,. And then perform an action, it is transferred to the next, and then the implementation of... , we can use the following diagram to show the whole process

We define the following after the transfer path, the sum of the payoff function is as follows

If S is only related to R, then the type can be written

Our goal is to select a group of the best action, which makes all the returns weighted and the expectation maximum.

From the above formula, we can find the return value decrease gradually after time t due to the discount factor y, which means the behind states will return the smaller reward and the impact is smaller. In order to maximize the expected value, it is better to put the big value of as high as possible, put small value of as far as possible.

There is a state s, the process will use a certain strategy to select the next action a, and then switch to another state s'. We call this strategy Policy, each policy is actually a mapping function: , from a state to action. When state is given, the action is known , which means if the process know the policy, it can get every action from every state in each step.

In order to judge the different quality of a policy, it is necessary to define a value function, the value function is also called the discounted cumulative reward.

In the current state of S, after selecting a good policy, the value function will return the weighted sum of expectations. It is actually very easy to understand, given a future plan of action, this action plan will be through one action by one action, and then each state will return a value while it arrives. The future state is closer to the current state, the weight is higher. This is similar with playing chess, during the game, we might consider several steps, and select the best plan, the next step is most import to current game, so the weight is highest. The step after next step is less import, the weight is lower than first step, but it is still higher than third step.

Change above formula into Bellman equation:

The first item has nothing to do with , The second item is the maximum reward expected value of new state s’.

After defining the optimal value function V\*, then we define the optimal policy as follows:

The next optimal action of each state is determined by the optimal policy.

According to the above formula, we can know that

The optimal value function V\*, is derived from the optimal policy, therefore the reward is better than the other policies..

3. Value Function (V function)

According to the statement above, the reinforcement learning to learn is a mapping from the environmental state to action, called policy PI: S - > a. However, the reward in reinforcement learning will delay, if you lose the game in the first n, then only the state Sn and action An received an immediate reward R (Sn, An) =-1, all States in front of Sn received the reward as 0. So for any state before the s and action a, immediately reward function R (s, a) cannot show weather the strategy is good or bad. So it is necessary to define the value function to show the long-term effects of current policy PI.

There are three kinds of value functions:

γ∈[0,1] is discount factor, indicating how important the future reward is to the current reward. In particular, γ=0 means the process doesn’t consider the long-term reward, γ=1 means long-term reward and immediate reward is equally important.

Now we expand the formula, ri is the reward in step i，s' is the next state:

Given policy π and initial state s, then action a=π(s), the state will change to next state s’ with probability p(s'|s,a)，then rewrite the formula:

action value function (Q function)

the definition of Q function：

Given current state and current action, the state will change to next state s’ with probability p(s'|s,a)，then rewrite the formula:

In Qπ(s,a)，not only the policy π and initial state sis given，but also the action a is given. This is the main difference between Qπ(s,a) and Vπ(s).

And we can transform Vπ(s) to Qπ(s,a):

3,Fitted Q iteration in Grid world 2500

Grid world is a simple but useful example to explain the basic RL algorithm.

[3,3,3,3,3;

5,4,5,3,4;

4,2,5,3,4;

2,2,4,3,4;

2,2,4,2,1];

Figure 3.5a

Figure 3.5a uses a 5\*5 matrix grid world to illustrate

value functions for a simple finite MDP. The agent starts from the start points ( the upper left corner), and stops at in the target location( the lower right corner), which is also called absorbing state. The immediate reward of absorbing state is 100. In order to facilitate the description, remember the first I line, the state of the first j column, in each state, there are four kinds of up and down about four kinds of optional actions, denoted as Au, ad, Al, AR, SIJ,, and so on. (up, down, left, right first letter), and the probability that the state is selected by the action of a is 1.

The cells of the grid correspond to

the states of the environment. At each cell, four actions are possible: north,

south, east, and west, which deterministically cause the agent to move one

cell in the respective direction on the grid. Actions that would take the agent

off the grid leave its location unchanged, but also result in a reward of −1.

Other actions result in a reward of 0, except those that move the agent out

of the special states A and B. From state A, all four actions yield a reward of

+10 and take the agent to A 0 . From state B, all actions yield a reward of +5

and take the agent to B 0 .

Suppose the agent selects all four actions with equal probability in all

states. Figure 3.5b shows the value function, v π , for this policy, for the dis-

counted reward case with γ = 0.9. This value function was computed by solv-

ing the system of equations (3.12). Notice the negative values near the lower

edge; these are the result of the high probability of hitting the edge of the grid

there under the random policy. State A is the best state to be in under this policy, but its expected return is less than 10, its immediate reward, because from

A the agent is taken to A 0 , from which it is likely to run into the edge of the

grid. State B, on the other hand, is valued more than 5, its immediate reward,

because from B the agent is taken to B 0 , which has a positive value. From B 0 the

expected penalty (negative reward) for possibly running into an edge is more

than compensated for by the expected gain for possibly stumbling onto A or B.

Fitted Q iteration

As in the general reinforcement learning problem defined by Sutton and Barto

(1998), the task in the batch learning problem is to find a policy that maximizes

the sum of expected rewards in the familiar agent-environment loop. However, differing from the general case, in the batch learning problem the agent itself is not allowed to interact with the system during learning. Instead of observing a state s, trying an action a and adapting its policy according to the subsequent following state Batch Reinforcement Learning.