Advanced Science and Technology Letters
Vol.82 (SoftTech 2015), pp.32-36
http://dx.doi.org/10.14257/astl.2015.82.07

Hierarchical Reinforcement Learning with Classification Algorithms

Shanhong Zhu[1,2], Weipeng Dong[1], Wei Liu[2]

[1]School of Computer and Information Engineering ,Xinxiang university, Henan, China
[2]InternationalSchool of Software, Wuhan University, Wuhan, China
zshazhu@163.com

Abstract. In autonomous system, Agent assigns to their task through interaction with the environment, using hierarchical reinforcement learning technology can help the Agent in the large, complex environment to improve learning efficiency. Through the experimental results the effectiveness of the proposed algorithm is demonstrated. The goal of this paper is to provide a basic overview for both specialists and non-specialists to how to decide a good reinforcement learning algorithm for classification.

Keywords: Reinforcement learning; KNN Classification algorithm; Classifying Option

1 Introduction

Reinforcement learning is put forward by Minsky for the first time in the 1950s, the problem to be solved is: an agent can sense the surrounding environment, how to choose by learning to choose reach the goal of the optimal action. When the agent is to make each activity in the environment, it will give appropriate reward or punishment, to show whether the results of the state are correct or not. For example, when agent is training for a chess game, game victory gives positive returns, but the game failed gives negative returns, the other is zero returns [1]. Agent's task is to learn from the indirect learn and delayed return, so that the subsequent action can produce the largest cumulative returns. Reinforcement learning is the mapping study of intelligent body state from the environment to the action, getting the biggest reward value of the accumulation of action from the environment. This method is different from supervised learning techniques which can inform what behavior through positive cases and counterexamples, it is to find the optimal behavior strategy through trial error.

2 KNN classification analysis

K nearest neighbors (KNN) classifier is a classical supervised method in the field of machine learning, based on statistical data. In 1967, Cover and Hart had proved the KNN classification deviation approximates to optimal Bayesian classifier [2]. The

KNN classifier is widely used in such areas as text classification, pattern recognition, image processing and spatio-temporal classification.

The decision-making of the KNN algorithm is to find test sample *k* nearest or most similar training samples in the feature space, and then the test sample is assigned to a majority vote of its *k* nearest neighbors. Essentially, it is still based on statistical method. In the real world, most of the practical data does not obey the typical model (such as Gaussian mixture, linearly separable, etc). Non parametric algorithms like KNN happen to deal with the situation. If domain classes of test sample sets has more cross or overlap, KNN is the more appropriate method.

The advantages of KNN algorithm are as follows: (1)There is no explicit training phase or it is very minimal. Once the data is loaded into memory, it began to classify. This means the training phase is pretty fast. (2)The data set has been labeled, that is to o say the selected neighbors of test sample have been classified correctly. (3)Independent of any data distribution, it only needs to be adjusted or assigned an integer parameter *k*.

The disadvantages of KNN algorithm are as follows: (1) It has a costly testing phase. The cost is in terms of both time and memory. More memory is needed to store all training data [3]-[4]. (2) It is highly dependent of number of samples. Along with the number of training samples increases, the classifier performance will be better and better. (3) Classification accuracy is affected because of the property is equivalent to the weight. (4) It is difficult to select an appropriate *k* value. Small or large value of *k* has different characteristics.

[illegible]

a

2.1 Q-learning

$t \quad t \quad Q \quad)$

Reinforcement learning (RL) is a kind of method to obtain the optimal strategy through the Agent to trial and error. Standard is the discrete time MDP model with a limited set of state set S and operating set A, at any moment t, the Agent, from the current state of *st*□*S*, chooses next executable action A (st), to obtain the immediate reward value*rt* and take the value function of a state-action as the target function. The purpose of reinforcement learning is to search the optimal strategy of *π*:S→A*, which maximizes the value function.

Q-learning is a kind of independent learning environment model proposed by Watkins, and its beauty is: value of Current state-action sums up all the needed information, to determine a discount of the total reward value of the optimal sequence which is executed:

Where, γ is discount factor. In Q-learning, getting a reward signal will update Q value at a time:

$$[illegible] \quad _k \; t \quad _k \; t \; 1 \quad t \quad t \quad k \quad t \quad t \; 1 \quad t$$

Where, k is the number of iterations, is the learning efficiency, to control learning speed.

2.2 Option

In order to build the hierarchical learning system, we make the introduction of semi-markov process (SMDP). Between state s and execution action a has a time interval τ.Option is a macro Q-learning proposed by Sutton in 1999.Learning task is abstracted into several options, and will make these options as a special kind of "action" added to the original action focus. We use a triple (I_i,π_i,β_i) to denote an Option.I_i □ S is state set for entry, $\pi_i$is internal strategy, and $\beta_i$is termination condition. Internal strategy includes the strategy $\pi_{i_a}:I_i\rightarrow A_i$defined on the original action and the strategy$\pi_{i_o}:I_i\rightarrow O_i$ defined on Option, so (I_i,π_{io},β_i) forms the layered Option.

According to the optimal strategyπ_{i_o}*, the implemented state-action value function is defined as:

According to the optimal strategyπio*, the implemented state-Option value function is defined as:

The Option is updated only at the end of an Option in the Option study, the corresponding update formula is:

3 The experimental simulation and analysis

In order to assess the performance of the proposed method, it is tested under the environment of the taxi [5]. As shown 5 * 5 grid environment in figure 1, it contains a taxi and a passenger. Specify the location of the four special B, G, R and Y, passengers can choose any position of the four to get on the bus, and then get off at any position. The mission of the taxi is to the initial position to passengers, takes him to the destination and puts him down. Each grid position on the taxi has six original actions: North, South, East, West, Pickup and Putdown, the probability of each movement is performed at 0.8. When the taxi arrives to the initial position of passengers the Pickup action is effective, and will gain + 10 bonus to perform. At the same time when the taxi has passengers and on the target position, the Putdown actions are effective, performing bonus of + 20,but performing other actions is - 1.

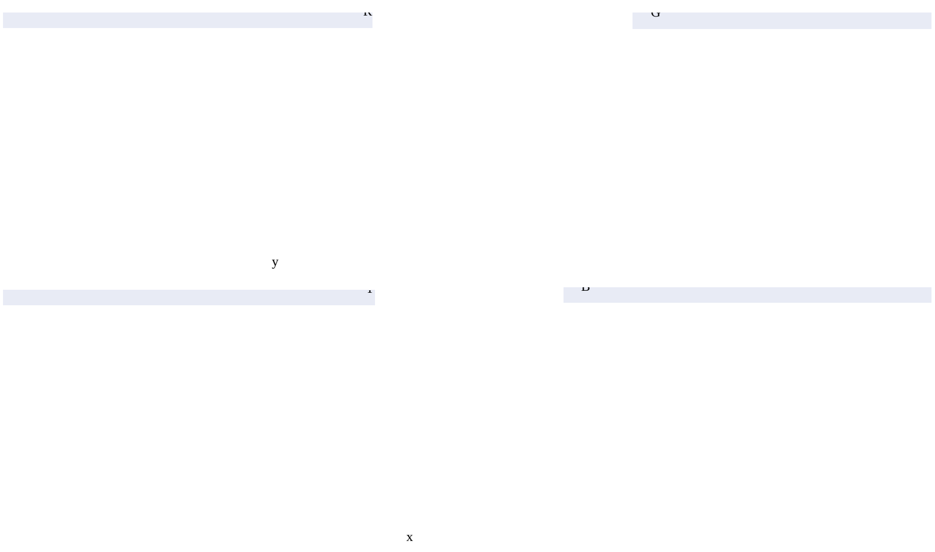

Fig. 1. The taxi problem

To reflect whether the proposed new algorithm performance is good, the experimental results are compared with the Q-learning and SCC. Three lines in figure 2 are respectively the taxi experimental results to solve the problems by Q-learning algorithm, SCC algorithm and the Ant-O algorithm. The ant algorithm to optimize system parameter is set tont=200, nk=10, α=0.98, ρ=0.98. Experiment has two stages. In the initial learning stage, the Agent is traversal in the environment and records the transformation process. Then based on the information transition diagram, obtains sub-goals. Next, the Agent uses the Option learned in the previous stage, to continue the interaction with the environment, learns the optimal strategy.

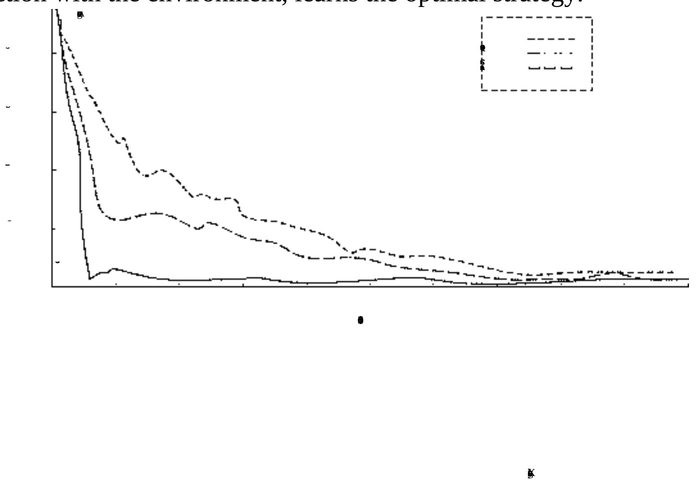

Fig. 2. Algorithm performance comparison

Experimental results are shown in figure 3, with the goal state steps taken to depict algorithm performance, [illegible] gradually convergence, and three methods are ultimately to find the optimal path.

However Ant-O algorithm proposed in this paper in a relatively short period of learning, makes search faster convergence to the optimum solution, shortens the time to learn. Compared with the Q-learning algorithm, the Ant-O introduces orderly directed and weighted graph, which is arranged on the shortest path pheromone concentration.

4 Conclusion

In essence KNN classification is a supervised learning method based on statistical data. Firstly, we present the general procedure of classification algorithm. There are three factors which affect classification performance, that is the number of neighbor, distance measure and decision rule. We have analyzed different Parameter of KNN, such as different distance function. Then we have summarized several important issues and recent developments of classification problems. These include building index structure to find nearest neighbor, reducing dimension of high-dimensional dataset, and being integrated into SVM or neural network. We do not discuss problems such as feature selection, dimension, class number, which are other important and difficult problem of classification. Specifically, the Option of KNN algorithm, through the combination of the ACO study method, using edge roughness find bottlenecks, makes hierarchical state space. Tasks are divided into multiple sub tasks, division of space, decrease the learning complexity, under the premise of not to affect the solution quality of the algorithm, significantly speed up the solution.

Acknowledgments. This project is supported by National Natural Science Foundation of China (Grant No. 51375001, 51309045).

References

1. Imsek, O., Barto, A. G.: Using relative novelty to identify useful temporal abstractions in reinforcement learning. Proc of the 21st International Conference on Machine Learning. New York: ACM Press, (2004)
2. Digney, B.: Learning hierarchical control structures for multiple tasks and changing environments. The 5th international conference on the Simulation of Adaptive Behavior, pp:321–330. (1998)
3. Mannor, S., Menache,I.: Dynamic abstraction in reinforcement learning via classifying. The 21st international conference on Machine learning, pp: 71–78. (2004),
4. Kazemitabar, S. J., Beigy, H.: Automatic discovery of subgoals in reinforcement learning using strongly connected components. ICONIP 1(5506). 829–834. (2008).
5. Taghizadeh, N.: Autonomous skill acquisition in reinforcement learning based on graph classifying. Islamic Republic of Iran: Sharif University of Technology, (2011).
6. Erik, D., Gilbert, P.: Scaling Ant Colony Optimization with Hierarchical Reinforcement Learning Partitioning. The 10th annual conference on Genetic and evolutionary computation.pp:25-32. (2008).