3F4-08

Automatic Generation of Macro Actions

Yu Chen[*1] Takanori Fukao[*2] Norihiko Adachi[*3]

Graduate School of Informatics, Kyoto University

Macro action is proposed to improve the performance of reinforcement learning agent. It has been proved to be able to increase learning efficiency if properly defined. So the problem of automatic generation of useful macro actions becomes very meaningful. Algorithm that uses data mining method to generate macro action automatically is discussed in this paper.

1. Reinforcement Learning with Macro Actions

A reinforcement learning problem is an optimal control problem where the controller is given a scalar reinforcement signal (or cost function) indicating how well it is performing. The reinforcement signal is a function of the state of the system and the control signal. The goal is to maximize the expected total discounted reinforcement.

The traditional reinforcement learning algorithms do not fit well to reinforcement learning problems with large action space because it takes the learning agent too much time to find the goal and the optimal. Macro action is proposed to stand for higher level actions and to accelerate learning speed when applied to reinforcement learning problems with large action space or large state space.[3] Each macro action is specified by a closed-loop policy, which determines the primitive actions when the macro action is in force and by a completion function, which determines when the macro action ends. When the macro action completes, a new primitive or macro action can be selected.

A macro action is defined to be a triple:

- s, the state to which the action applies
- π, a policy that specifies how the action is executed
- β, a completion function, specifying the probability of completing the macro action on every time step

In different cases there should be different terminating conditions accordingly. The occurrence of special events like hitting the wall or finding the way out of a room would be good terminating condition.

2. Methods to Find Macro Actions Automatically

In the first paper using the phrase "macro action" by Sutton,[3] the way to find useful macro action is to be human-defined and pre-learned. In his paper, the agent is required to find the goal in a four-room-with-doorway environment. Obviously doorway macro action will be the right choice. It could be only useful in an environment wit h doorway property and it is defined according to the judgment of human operator (left in Fig. 1). Though the efficiency of macro actions is fully proved in this way, how to find proper macro actions without pre-knowledge of the environment still remains unsolved. Amy Mcgovern came up with an approach called 'acQurie-macros'[2] which can find interesting states by finding the peaks in the temporal history of rewards and examining the state visitation frequencies. This is a great advance than Sutton's Door-Way macro action because this is done online without preknowledge and not limited to certain type of environment.

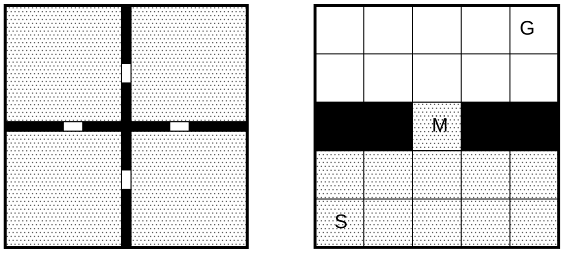

Fig.1 Door-Way and Bottleneck Macro Action

Another approach, Diverse Density by Amy McGovern [1]is based on the idea of "mining" an ensemble of behavioral trajectories accumulated by the agent as it interacts with its environment. The focus is on discovering subgoals of achievement by searching online for "bottlenecks" in observation space. Informally, a bottleneck is a region in the agent's observation space that the agent tends to visit frequently on successful paths to a goal but not on unsuccessful paths (for some suitable definition of success).

3. Making Clusters in Environment

As defined in Chap. 1, there are three basic elements necessary to define a macro action. The algorithms above are mainly focusing on finding termination condition β automatically, while little study has been carried out on the S which stands for the states from where certain macro action is available. In Amy McGovern's Diverse Density approach which features bottleneck, § is defined in a nature way, shown in right part in Fig.1 that the bottleneck is the doorway between the upper and lower half and thus for all states in the lower half, "go-to-bottleneck" must be a good macro action. An automatic generation algorithm should be able to generate both termination condition β and the possible states S automatically. So in order to solve the

Contact: Yu Chen, Graduate School of Informatics, Kyoto University, Kyoto, 075-753-4940, yuchen@sys.i.kyoto-u.ac.jp

problem of defining proper S automatically, we propose the clustering method which will divide the state space into clusters.

3.1 Data Clustering

The clustering problem is defined as the problem of finding homogeneous groups of data points in a given data set. Each of these groups is called a cluster and can be defined as a region in which the density of objects is locally higher that in other regions. The most commonly used partitional clustering strategy is based on the square-error criterion. The general objective is to obtain that partition which, for a fixed number of clusters, minimizes the square-error.

Suppose that the given set of n patterns in d dimensions has somehow been partitioned into K clusters $C_1, C_2, ..., C_K$ such that cluster C_K has n_K patterns and each pattern is in exactly one cluster, so that

$$\sum_{k=1}^{k} n_k = n \tag{1}$$

The mean vector or center of cluster C_k is defined as the centroid of the cluster or

$$m^{(k)} = (1/n_k) \sum_{i=1}^{nk} x_i^{(k)} \tag{2}$$

where $x_i^{(k)}$ is the ith pattern belonging to cluster C_k. The square-error for cluster C_k is the sum of the squared Euclidean distances between each pattern in C_k and its cluster center $m(k)$. This square-error is also called the within-cluster variation.

$$e_k^2 = \sum_{i=1}^{nk} (x_i^{(k)} - m^k)^T (x_i^{(k)} - m^{(k)}) \tag{3}$$

The square-error for the entire clustering containing K clusters is the sum of the within-cluster variations:

$$E_K^2 = \sum_{k=1}^{K} e_k^2 \tag{4}$$

The objective of a square-error clustering method is to find a partition containing K clusters that minimizes E_k^2 for fixed K The basic idea of an iterative clustering algorithm is to start with an initial partition and assign patterns to clusters so as to reduce square-error.

3.2 Algorithm

The algorithm using clustering method is like the following:

- Initialize
- Make K clusters over the environment with K center points. The cluster is made in a way that square-error of each data points to the center point is minimized and the total square-error of all clusters is also minimized. As a result the whole environment is divided into K small clusters.
- Find the terminating state in every cluster except the one containing goal state. This can be done using Diverse Density method, or here just find the state with highest Q value.
- Adjust cluster according to terminating state selected so that total $Q_{increase}$ of all states in the cluster to the terminating state is maximized.
- Update terminating state, cluster together with Q value.
- Execute local policy π_k in every cluster when the membership is stabilized through iterations.

4. Simulation

The above algorithm is applied in a robot navigation simulation problem. The goal is set on the right corner of the environment. The result of clustering is shown in Fig.2.

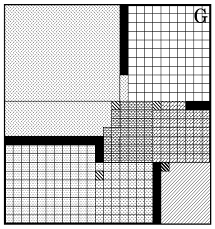

Fig.2 Clusters in the the Environment

The environment is divided into 7 clusters first according to the the geographical property and then adjusted according to the Q value. The proposed algorithm is proved to be able to make proper clusters in environments with less obvious properties. Thus when combined with terminating state searching algorithms like the Divers Density or simply Max Q Increase algorithm, the automatic generation of macro actions can be realized.

References

[1] Amy McGovern and Andrew G.Barto: Automatic Discovery of Subgoals in Reinforcement Learning using Diverse Density , *In Proceedings of the 20011 International Conference on Machine Learning,2001* (2001)

[2] Amy McGovern: acQuire-macros: An Algorithm of Automatically Learning Macro-actions, *NIPS 98 Workshop on Abstraction and Hierarchy in Reinforcement learning(1998)* (1998)

[3] Amy McGovern and Richard Sutton: Macro-Actions in Reinforcement Learning : An empirical Analysis. *University of Massachusetts ,Amherst Technical Report*, Number 98-70,(1998).

[4] L. Kaufman, P.J. Rousseeuw : Finding Groups in Data: an Introduction to Cluster Analysis, *1990 John Wiley &Sons* (1990)