Ali Ridho Barakbah, Kohei Arai, "Pursuit Reinforcement Competitive Learning: an approach for on-line clustering",
The 2nd International Seminar on Information and Communication Technology Seminar (ICTS) 2006, Surabaya, Indonesia.

Information and Communication Technology Seminar (ICTS) 2006
Informatics Department, Faculty of Information Technology-ITS, Surabaya

PURSUIT REINFORCEMENT COMPETITIVE LEARNING: AN APPROACH FOR ON-LINE CLUSTERING

Ali Ridho Barakbah *, Kohei Arai
* Department of Information Technology, Electronic Engineering Polytechnic Institute of Surabaya-ITS
Kampus ITS Sukolilo, Surabaya 60111, Indonesia
E-mail : ridho@eepis-its.edu

ABSTRACT

Clustering is commonly used for static data. After gathering data, clustering can be done to organize data into groups whose members are similar in some way. This kind of clustering may not be used for stream data which comes unpredictably. On-line clustering is a kind of clustering for dynamic data. In this paper we proposed an approach for on-line clustering, called Pursuit Reinforcement Competitive Learning (PRCL). This algorithm is inspired by reinforcement learning based on pursuit method for stabilizing the update states. Experimental result in this paper performs the effectiveness of this proposed approach to reach stable convergence using Iris dataset. This paper includes the comparison results between the proposed method and the other algorithms for on-line clustering.

Keywords : on-line clustering, reinforcement learning.

1 INTRODUCTION

Clustering is an exploratory data analysis tool that deals with the task of grouping objects that are similar to each other [1,2,3]. For many years, many clustering algorithms have been proposed and widely used. It is commonly used in many fields, such as data mining, pattern recognition, image classification, biological sciences, marketing, city-planning, document retrieval, etc.

Many cases of clustering commonly used the static data. It means that the clustering can be made after the entire data have been collected, then grouped into clusters whose members are similar in some way. In the data mining, there is a kind of data which comes every time so that we can not stop it in a while in order to make clustering.

On-line clustering is a kind of clustering that is used for dynamic data. It is not consider a number of data, but only focus on a new data and previous centroids. However, determining position of each centroids because of a new data attracted some approaches. Vector Quantization (VQ) was a very simple approach to do on-line clustering. It is derived from concept of competitive learning network. Likas (1999) proposed Reinforcement Guided Competitive Learning (RGCL) as an approach for on-line clustering based on reinforcement learning. It utilized the concept of reward in the reinforcement learning from winning unit in the Learning Vector Quantization. The Sustained RGCL (SRGCL) was modification of RGCL in considering a sustained exploration in reinforcement learning.

In this paper, we proposed a new approach for on-line clustering based on reinforcement learning, called Pursuit Reinforcement Competitive Learning. PRCL is derived from pursuit method in reinforcement learning that maintain both action-value and action preferences, with the preferences continually pursuing the action that is greedy according to the current action-value estimates. This paper performs the experimental result of PRCL and compared to VQ, RGCL and SRGCL.

2 BASIC THEORIES

This chapter describes about the basic theories of reinforcement learning and the algorithms of on-line clustering which are Vector Quantization, Reinforcement Guided Competitive Learning (RGCL) and Sustained RGCL (SRGCL). The algorithms will be used as comparing methods in chapter 5.

2.1 Reinforcement learning

Reinforcement Learning is learning what to do---how to map situations to actions---so as to maximize a numerical reward signal [4]. The learner is not told which actions to take, as in most forms of machine learning, but instead must discover which actions yield the most reward by trying them. In the most interesting and challenging cases, actions may affect not only the immediate reward, but also the next situation and, through that, all subsequent rewards. These two characteristics (trial-and-error search and delayed reward) are the two most important distinguishing features of Reinforcement Learning.

Reinforcement Learning is defined not by characterizing learning algorithms, but by characterizing a learning problem. Any algorithm that is well suited to solving that problem we consider to be a Reinforcement Learning algorithm. Clearly such an agent must be able to sense the state of the environment to some extent and must be able to take actions that affect that state. The agent must also have a goal or goals relating to the state of the environment.

2.2 Simple competitive learning

In competitive networks, output units compete for the right to respond. The goal of competitive learning is how can be considerable as a method of clustering by dividing the data into a number of clusters such that the inputs in the same cluster are in some sense similar.

A basic competitive learning network has one layer of input nodes and one layer of output nodes. Binary valued outputs are often (but not always) used. There are as many output nodes as there are classes.

Often (but not always) there are lateral inhibitory connections between the output nodes. However, in simulations, the function of the lateral connections can be replaced with a different algorithm.

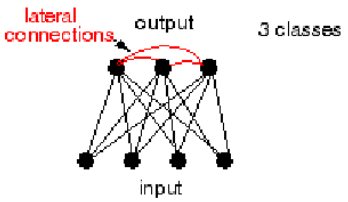

Figure 1. Lateral connection in output nodes

The output units are also often called grandmother cells. The term grandmother cell comes from discussions as to whether your brain might contain cells that fire only when you encounter your maternal grandmother, or whether such higher level concepts are more distributed.

2.3 Vector Quantization

Vector quantization is one example of competitive learning [5]. The goal here is to have the network "discover" structure in the data by finding how the data is clustered. The results can be used for data encoding and compression. One such method for doing this is called vector quantization.

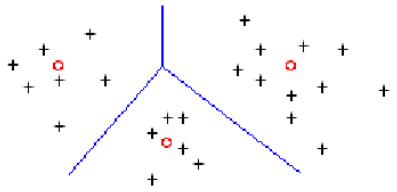

Figure 2. Illustration of prototype vector (red) between input (+)

In vector quantization, we assume there is a codebook which is defined by a set of M prototype vectors. M is chosen by the user and the initial prototype vectors are chosen arbitrarily. An input belongs to cluster i if i is the index of the closest prototype (closest in the sense of the normal euclidean distance). This has the effect of dividing up the input space into a Voronoi tessellation. Implementing Vector Quantization with a network is as shown in Figure 4. The prototypes are presented the weights vector connecting to each output node.

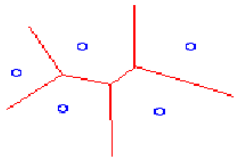

Figure 3. Voronoi tessellation

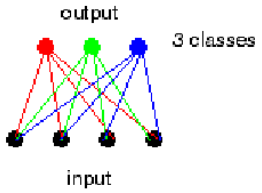

Figure 4. Implementing Vector Quantization with a network

Algorithm of Vector Quantization can be described as follows:

1. Choose the number of clusters M
2. Initialize the prototypes $w_1^*...w_m^*$ (one simple method for doing this is to randomly choose M vectors from the input data)
3. Repeat until stopping criterion is satisfied:
   - Randomly pick an input x
   - Determine the "winning" node k by finding the prototype vector that satisfies

$$| w_m^* - x | \leq | w_i^* - x | \text{ (for all } i \text{)} \quad (1)$$

   - Update only the winning prototype weights according to

$$w_k^* = w_k^* + m\,(x - w_k^*) \quad (2)$$

2.4 Reinforcement Guided Competitive Learning

Reinforcement Guided Competitive Learning (RGCL), proposed by A. Likas, is a new method to solve on-line clustering problem using Reinforcement Learning [6]. It considers that each cluster i ($i = 1,\ldots,L$) corresponds to a Bernoulli unit, whose weight vector $w_i = (w_{il},\ldots, w_{ip})^T$ corresponds to the prototype vector for cluster i. At each step, each Bernoulli unit i is fed with a randomly selected pattern x and performs the following operations.

First the distance $s_i = d(x, w_i)$ is computed and then the probability p_i is obtained as follows:

$$p_i = h(s_i) = 2\,(1 - f\,(s_i)) \quad (3)$$

where $f(x) = 1 / (1+ exp(-x))$. Function h provides values in (0, 1) (since $s_i \geq 0$) and is monotonically decreasing. Therefore, the smaller the distance si between x and w_i, the higher the probability p_i that the output y_i of the unit will be 1. Thus, when a pattern is presented to the clustering units, they provide output one with probability inversely proportional to the distance of the pattern from the cluster prototype. Consequently the closer (according to some proximity measure) a unit is to input pattern, the higher the probability of the unit to be active (ie. $y_i = 1$). The probabilities p_i provide a measure of the proximity between patterns and cluster centers. Therefore if a unit i is active, it is very probable that this unit is close to the input pattern.

Therefore, each iteration of the RGCL clustering algorithm consists of the following steps:

1. Randomly select a sample x from the data set.
2. For $i = 1,\ldots, L$ compute the probability p_i and decide the output y_i of cluster unit i.
3. Specify the winning unit i^* with $p_{i^*} = max_i p_i$.
4. Compute the reinforcements r_i ($i = 1,\ldots, L$) as follows:

$$r_i = \begin{cases} 1 & \text{if } i = i^* \text{ and } y_i = 1 \\ -1 & \text{if } i = i^* \text{ and } y_i = 0 \\ 0 & \text{if } i \neq i^* \end{cases} \quad (4)$$

5. Update the weight vectors w_i ($i = 1,\ldots, L$) as follows:

$$\Delta w_{ij} = a\, r_i\, (y_i - p_i)\, (x_j - w_{ij}) \quad (5)$$

where a is learning rate.

2.5 Sustained RGCL

The modification of RGCL that employs the above weight update scheme will be called the SRGCL algorithm (Sustained RGCL) [6]. Consequently the update equation of the RGCL algorithm now takes the form:

$$\Delta w_{ij} = a\, r_i\, (y_i - p_i)\, (x_j - w_{ij}) - \eta w_{ij} \quad (6)$$

The parameter $\eta > 0$ must be much smaller than the parameter a so the term $-\eta w_{ij}$ does not affect the local search properties of the algorithms, ie the movement towards local minimum states.

3 PROPOSED APPROACH

In this topic of clustering, we carried out research dealing with applicability of Reinforcement Learning for clustering problem. We utilized the pursuit algorithm in Reinforcement Learning to select the winning weight factors. Therefore we call as Pursuit Reinforcement Competitive Learning (PRCL).

The algorithm of PRCL is described as follows:

1. First of all, we determine the winning unit i^* from:

$$d_{i^*} = \arg\min_i d(x, w_i) \quad (7)$$

2. Update the reward track of x for all weights, as follows:

$$\begin{aligned} & r(x, w_{ij}) = r(x, w_{ij}) + \beta\,(1 - r(x_j, w_{ij})) \quad \text{if } i = i^* \\ \text{and } & r(x, w_{ij}) = r(x, w_{ij}) + \beta\,(0 - r(x_j, w_{ij})) \quad \text{if } i \neq i^* \end{aligned} \quad (8)$$

3. Select the winning i^* from maximizing the reward as:

$$w_i^* = \arg\max_i r(x, w_i) \quad (9)$$

4. Update the weight vectors as follows:

$$\begin{aligned} & \Delta w_{ij} = a\,(x_j - w_{ij}) \quad \text{if } i = i^* \\ \text{and } & \Delta w_{ij} = 0 \quad \text{if } i \neq i^* \end{aligned} \quad (10)$$

where a is learning rate and β is reward rate.

4 EXPERIMENTAL RESULT

We make experiments for applicability of Reinforcement Learning for clustering with minimizing objective function $J = \sum min_r d(x_i, w_r)$ for Iris data set. We involve four methods: Vector Quantization (VQ), RGCL, Sustained RGCL (SRGCL) and PRCL. We set the parameter values as follows:

- VQ → $a = 0.1$
- PRCL → $a = 0.1$, $\beta = 0.1$
- RGCL 1 → $a = 0.5$ (iteration ≤ 500), $a = 0.1$ (iteration > 500)
- RGCL 2 → $a = 0.1$
- SRGCL 1 → $a = 0.5$ (iteration ≤ 500), $a = 0.1$ (iteration > 500), $\eta = 0.0001$
- SRGCL 2 → $a = 0.1$, $\eta = 0.0001$

We compute for each experiment with 1500 numbers of iteration. We record the average results from 10 experiments for each algorithm. Figure 5 performs the comparison result of each algorithms.

From Figure 5 we can evaluate the performance with both function objective J and speed calculated by number of iteration. Figure 5 shows that PRCL, RGCL 1 and RGCL 2 can reach the minimum J. RGCL 1 is faster to find the minimum J, but it has not converged yet in next iterations. Compared with RGCL 1, PRCL performs better to find the stable condition when reach the minimum J. From this experimental result, the proposed approach in this paper is a fastest algorithm to find a stable convergence in the minimum J compared with VQ and RGCL.

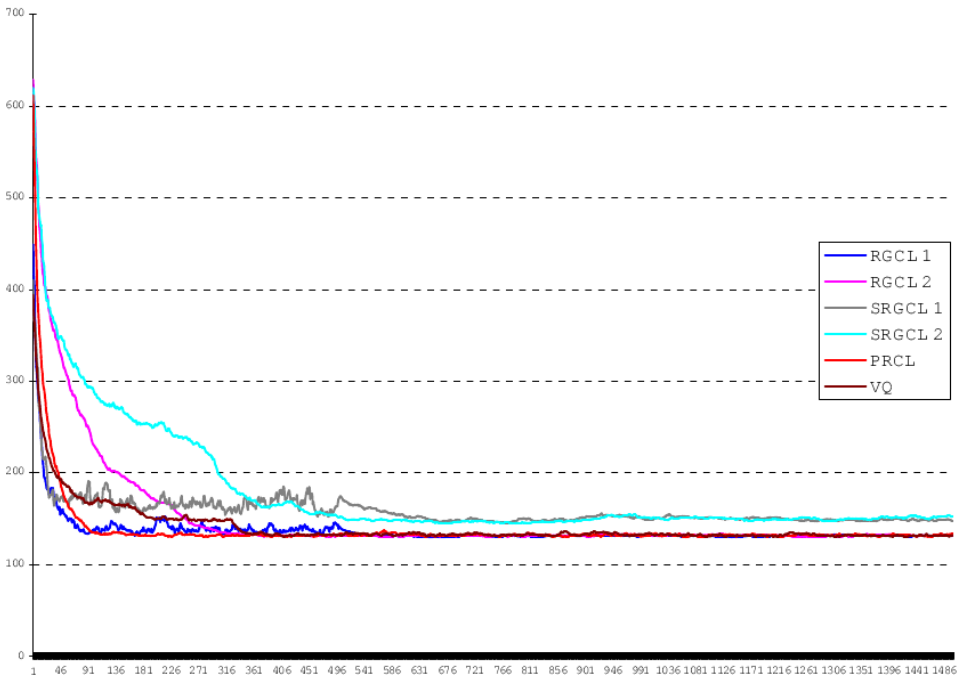

Figure 5. The comparison of experimental result

REFERENCE

Paper : author, title, journal(italic), edition number, year and the number of pages. Example :

[1] G. Karypis, E.H. Han, V. Kumar, Chameleon: a hierarchical clustering algorithm using dynamic modeling, *IEEE Computer: Special Issue on Data Analysis and Mining*, 32(8):68W5, 1999.

[2] M. Halkidi, Y. Batistakis, M. Vazirgiannis, Clustering algorithms and validity measures, *Proceedings of the 13th International Conference on Scientific and Statistical Database Management, July 18–20, IEEE Computer Society*, George Mason University, Fairfax, Virginia, USA, 2001.

[3] W.H. Ming and C.J. Hou, Cluster analysis and visualization, *Workshop on Statistics and Machine Learning*, Institute of Statistical Science, Academia Sinica, 2004.

[4] R.S. Sutton, A.G. Barto, Reinforcement learning: an introduction, *The MIT Press*, 1998.

[5] Genevieve B. Orr, Simple competitive learning, *Neural networks course*, http://www.willamette.edu/~gorr/classes/cs449/Unsupervised/competitive.html.

[6] A. Likas, A reinforcement learning approach to on-line clustering, *Neural Computation*, 11:1915-32, 1999.