New Recommendation System Using Reinforcement Learning

Pornthep Rojanavasu
University of Phayao

Phaitoon Srinil
Burapha University

Ouen Pinngern
Ramkhamhaeng University

New Recommendation System Using Reinforcement Learning

Pornthep Rojanavasu*, Phaitoon Srinil**, Ouen Pinngern***

Research Center for Communications and Information Technology
King Mongkut’s Institute of Technology Ladkrabang
Department of Computer Engineering
King Mongkut’s Institute of Technology Ladkrabang
Email: s8060022@kmitl.ac.th*, s8060020@kmitl.ac.th**, kpouen@kmitl.ac.th***

Abstract

Recommendation system are widely used in e-commerce that is a part of e-business. It helps users locate information or products that they would like to make offers. In this paper, we purpose a new web recommendation system based on reinforcement learning, which is different from another system using Q-learning method. By using ε-greedy policy combined with SARSA prediction method, another powerful method of reinforcement learning is obtained. The system gives customer more chance to explore new pages or new products which are not popular which may match with their interests. The system composes of two models. First, a global model, the model for all customers to discover behavior of system. We can know another users direction or trend by global model. Second, a local model, which uses to keep records of user browsing history and makes offer from each customers. We report experimental studies that show the click rate of recommendation list.

1. Introduction

The growth of information on World Wide Web make users more difficult to search for relevant information as the amount of product in e-business increases rapidly. Customers suffer from searching for interested products. To avoid this problem, many websites use recommendation system to help customer finding the satisfy products. Recommendation systems are categorized into two major classes: content-based filtering and collaborative filtering [1]. In content-based filetering, the system tries to match the content of product with user profile, both content of product and user profile represented by keywords. Robin van Meteren and Maarten van Someren proposed PRES [3] that use content-based filtering techniques to suggest document that relevance to user profile. The user profile was created by user feedback. In collaborative filtering, the system try to match user pattern with another users that had the same taste then predict the most user’s interest to items. GroupLens [2] is a collaborative filtering of netnews, by rating articles after read, to suggest customers the articles that related with their interests.

In this paper we propose new web recommendation system based on reinforcement learning. The nature of customers when they want to buy something. They will take time for choosing produces that best match with their styles. During the choosing period, the produces that they choose will closely and closely match with their interests. Like customers when browse the produces on website, we use reinforcement learning for capture the past behavior of customers browsing and suggest customers the products expected that customers want to make offered.

In section 2 we present a brief overview of reinforcement learning. In section 3 we explain system architecture and show you how to implement reinforcement learning with recommendation system. Section 4 shows you some experiment result. Section 5 concludes the paper and the direction of future research.

2. Reinforcement Learning

Reinforcement learning is one of powerful machine learning algorithm [4]. Learning from reinforcement is a trial-and-error learning scheme. Agent can learn to perform an appropiate action by recieving evaluation feedback. The objective is trying to maximize the expected sum of future values for each state. One major component of reinforcement algorithm is On-Policy TD control, called SARSA method. It consists of state-action pair and we can learn from the changing of value state Q(s,a) between state-action pair to another state-action pair. The value state defined as:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (1)$$

where s_t is the state of agent at time t, a_t is the action of agent at time t, r_{t+1} is reward of state s that action a, α is learning rate ($0 \le \alpha < 1$) and γ is discount rate ($0 \le \gamma < 1$).

The Sarsa control algorithm is given in figure 1.

```
Initialize Q(s, a)
Repeat
    Initial s
    Choose a from s using policy
    Repeat
        Take action a, observe r′, s′
        Choose a′ from s′ using policy
        Update Q(s, a)
        s′←s; a′←a;
    until s is terminal
```

Figure 1. Sarsa learning algorithm

First, we initialize value of Q(s,a) and choose the initial state s. Second, we select action *a* at state *s* using policy. The policy can be greedy or ε-greedy policy. Next, repeat take action *a*, observe the reward *r* and the next state *s′*, choose *a′* from *s′* using policy for compute the value of future next state and then update the Q(s,a) of current state and change *s*←*s′*, *a*←*a′*.

3. System Architecture

The reinforcement recommendation system architecture shown in Figure. 2.

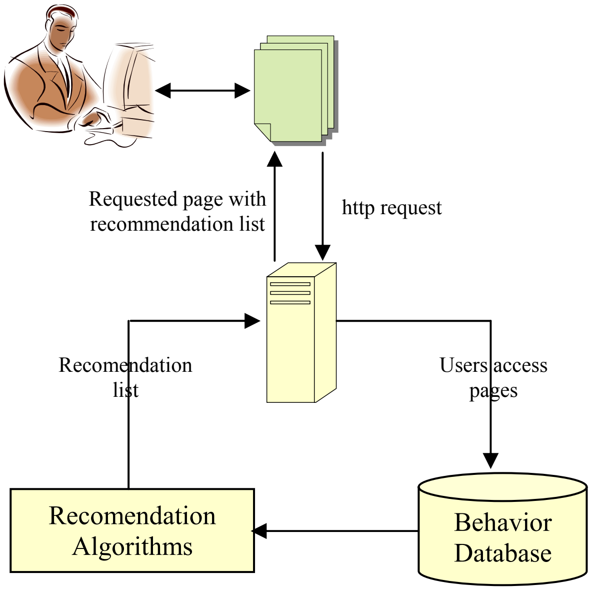

Figure 2. System architecture

1. Behavior Database. The system keeps behavior of customers into two groups. First, the global behavior collected behavior of all customers, we can know another customers direction. Second, the local behavior collected behavior of each customer, we can know which product bought by them.
2. Recomendation Algorithms. It's a part for learning by reinforcement learning and send recommendation list to users.

When the customers login to system and open a page. The pages can be viewed as states *s* of the system and links in a page can be viewed as action *a* of state. The system puts the state diagram (page-to-page) as shown in fig.3 and the changing value between states into Q-matrix.

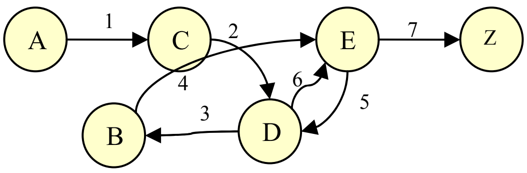

Figure 3. An example a state diagram of user behavior.

The system has two type of Q-matrix. First, global Q-matrix keeps the changeablility of whole system. The global Q-matrix can tell you the trend of customers or most popular products. Second, local Q-matrix keeps the changeablility of each customer. The local Q-matrix likes customer profile that record customer browsing behavior. To update of both Q-matrix system will get reward from click on the products and make offered, we call customer feedbacks. Customer feedbacks are important part of recommendation system. The Customer feedbacks may be explicit and implicit. Customers can send the explicit feedback by rating the products. The system will update Q-Matrix of that produce page. For example, the customer may give rate 3 out of 5. Although explicit rating is accurate, there are few customers who gave rating for produces after they used. The system needs another feedback from customers. Implicit feedback is percieved by keeping customer's behavior. It has two type of implicit feedback. First, when customer changes state that means click on product page. Second, when customer changes state to final state that means product added into shopping card and bought its.

In Table.1 shown you a local Q-matrix of customer who has the changing state like in Figure 3. The customer logins into website and click on product A. Then he/she has sequence of the changing state like A→C, C→D, D→B, B→E, E→D, D→E and bought product E. When customer click on each produce page, system will update both Q-matrix by plus 1 and when customer bought the product plus 3 to that product.

Table 1. Q-matrix of user

Action State	A	B	C	D	E
A			1		
B					1
C				1	
D		1			1+1+3
E				1	

To predict next state or next product that customers may prefer to offer, the system will rank products. The ranking system separated into 2 parts. First part is the ranking of whole system, global ranking, system uses data from global Q-matrix to choose the next state by using ε-greedy policy.

If you maintain estimates of the action values, then at any time there is at least one action whose estimated value is greatest. We call this a greedy action. If you select a greedy action, we say that you are exploiting your current knowledge of the values of the actions. If instead you select one of the nongreedy actions, then we say you are exploring because this enables you to improve your estimate of the nongreedy action's value [4]. The ε is a small probability to select an action at random. The advantage of ε-greedy policy over greedy policy are the ε-greedy policy continue to explore and improve their chances of recognizing the product that customer may make offered. The greedy policy will choose only the state that has maximum values. If the ε=0.2, the method explores more than, and usually finds the optimal action earlier, but never selects it more than 81% of the time. The ε=0.1 method improves more slowly. For example if the system have 5 ranks and ε=0.1. In each position of the system will have chance to choose the highest Q-value equal to 90% and choose another equal to 10%. To do like this the system gives a chance for new products or product that have few clicks on but may be match with your interest.

Second part is the ranking of each customer, local ranking, the system considered from local Q-matrix. To choose the next state, the system uses inverse ε-greedy policy. The states that customer ever visit, system will decrease important and gives a chance to explore other products. After that system finds Q_{total} by using eq 2.

$$Q_{total} = Q_{local} + wQ_{global} \qquad (2)$$

where w is the weight of Q_{global} and $w \in (0,1]$. Final, system will rank produces by using Q_{total} and recoment to customers.

4. Experiments

The objective of experiment was to find relationship between ε and user click rate, the click rate is a measure of how many of the presented products in recomendation list that customer clicks on [1]. Tab. 2 shows the average customer clicks rate where w=0.8.

Table2. The average customer click rate.

degree of ε	average custumer click rate
0.10	62.75 %
0.15	68.50 %
0.20	75.50 %
0.25	71.30 %
0.30	60.75 %
0.40	58.00 %
0.50	52.50 %

We found that ε = 0.2 can preserve balancing exploration and exploitation power. If ε less than 0.2 the system will exploit to the trend of system and gave a small chance to explore new product. So in the early time of user login the clicks rate was high and after that the system showed that the clicks rate dropped for the same previous product. Although Q_{local} try to promote the new product, it is less effective than Q_{global}. If ε more than 0.2 the system will too much explore the products. The system may include the product that does not match the customer.

5. Conclusions and Future Work

In this paper we have presented a general framework for recommendation system based on reinforcement learning. The system can learn direct from customer's behavior. The learning process is using SARSA method and ε-greedy policy. The system consists of two parts, global model and local model. One important aspect of learning method is balance of exploration and exploitation. The ε-greedy policy can give a chance to explore new produces, but the same time its exploit to the trend of system. We showed this in a simple experiment about the effect of ε value and user click rate. The result of this experiment can explain that if you explore too much it has a chance to comment the uninterested product to users, if you exploit too much it has a chance to stick with other opinion and never see the other products.

In the real world, it requires more space to keep global state and local state of all users. We plan to continue reducing the space by using another data structure. We also study in the effect of w to find the optimal w for Q_{total}.

6. References

[1] G. Adomavicius, A.Tuzhilin. (2005) “Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions”, *IEEE Transaction on knowledge and data engineering*, VOL 17, NO.16, JUNE 2005.

[2] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl. “GroupLens: An open architecture for collaborative filtering of Netnews”, *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, Chapel Hill, NC: Pages 175-186

[3] R. V. Meteren, M. V. Someren (2000). “Using Content-Based Filtering for Recommendation”, *MLnet / ECML2000 Workshop*, May 2000, Barcelona, Spain.

[4] Sutton, R.S. and Barto, A.G. (1998) *Reinforcement Learning: An Introduction*, MIT Press, Cambridge.