Content-based Recommendation for Traffic Signal Control

Y. F. Zhao, F. Y. Wang, H. Gao, F. H. Zhu, Y. S. Lv
The State Key Laboratory of Management and Control for Complex Systems, CASIA
Beijing 100190, China
{yifei.zhao, feiyue.wang, hang.gao, fenghua.zhu, yisheng.lv}@ia.ac.cn

P. J. Ye
Southwest China Institute of Electronic Technology,
Chengdu 610036, China
Peijun_ye@163.com

***Abstract*—Traffic signal control is an effective way of solving urban traffic problems by providing appropriate signal control plans for various intersections. Essentially, the aim of Traffic Signal Control is to find the best matching timing plans to current traffic conditions. Inspired by recommendation technology, we regard traffic conditions as users, timing plans as items, and traffic indicators like delay time are regarded as the ratings that users give to items. By means of Content-based Recommendation technology and k-Nearest Neighbor method in Recommendation Systems, we first find the similar traffic conditions according to the characteristics of traffic conditions. Then the matching degree between current traffic conditions and various timing plans can be predicted by analyzing the history data of selected similar traffic conditions. What's more, Artificial Transportation Systems method was applied to recommend and sort the timing plans for various traffic conditions in this paper. With normalized Discounted Cumulative Gain, which is a measure of ranking quality, was chosen as the performance indicator, we conducted the experiments in Paramics. The results showed that the strategies based on our method outperform the classic Webster method.**

Keywords—Artificial Transportation Systems and simulation; Contented-based Recommendation; data mining and data analysis; Traffic Signal Control

I. INTRODUCTION

As known to all, traffic congestions have attracted more and more attention from all over the world. Because it is not only about social and economic, but it is all about people's livelihoods. To alleviate traffic congestion, Traffic Signal Control (TSC) appeared [1-3]. Through deploying and developing in the past few decades, TSC has greatly improved the traffic safety and efficiency.

In the theory, TSC could be traced back to the mid-20th century when Webster method was proposed [4]. Nowadays Webster is often chosen to be the benchmark in the TSC, because of its ease of implementation and effectiveness. What's more, three kinds of models of transportation systems were born, macroscopic, mesoscopic and microscopic traffic models. For example, Lighthill Whitham Richards (LWR) [5, 6] method is a famous macro-model, while Lattice Boltzman Method (LBM) [7] method is a typical meso-model. In addition to these, many microscopic simulation software with their own micro-model came into the world, which are good at describing the complex microscopic behaviors such as vehicle overtaking, lane changing and so on. Among them, Artificial Transportation Systems (ATS) [9], which is based on the Multi-Agent Systems (MAS) [8-11], become more and more popular. The actual transportation systems even could be replaced by an ideal ATS.

In the reality, there are some traffic management systems that have been implemented, such as SCOOT, SCATS, TRANSYT, and so on [8].

Although the traditional control strategies mentioned above have been confirmed to have good effects, they still need to be improved. Usually accurate models tend to be complex process and large amount of calculation in the traditional methods. We often seek a balance point which can ensure real time and certain speediness by sacrificing some accuracy. Despite this, many theoretical methods cannot achieve the desired results, or be applied in reality. Because they are usually based on the quantitative and idealized hypothesis. On the other hand, there are too many uncertain factors in the transportation systems, such as pedestrians and vehicles, roads, weather environment, legal policies [9, 10], so that it is impossible or difficult to model exactly.

In recent years, [12, 13] have proposed to improve TSC systems by using data mining and ATS. In [12], considering a great deal of uncertain factors in the transportation systems, which cannot be counted completely and quantified, authors applied the latent factor model [14, 15] in the transportation systems. By training huge history data, the latent factors could take those uncertain factors into consideration without knowing what they are. In [13], lots of event agents were generated by ATS with the help of ACP (Artificial System, Computational Experiments, Parallel Execution). Then it predicted the matching degree between new intersection and timing plan through computational experiments. Two paper all demonstrated that their methods are superior to the classic Webster method.

Above-mentioned two methods adopt the idea of Collaborative Filtering [16, 18-20], results-oriented method, and model-free adaptive control [21]. But they must rely on a great deal of satisfactory history data, which is usually not abundant in reality. So we attempt to apply the Content-based Recommendation method [17, 18] to find several most similar road conditions according to the characteristics of road conditions. Then recommend the best timing plans to current road conditions by using k-Nearest Neighbor (kNN) algorithm [19, 20].

When Content-based Recommendation is implemented to predict the matching degree between traffic conditions and timing plans, the advantages are shown as follows: 1) it provides an idea which is similar to model-free adaptive control [21]. Not only it could block out the uncertainty details when modeling, but also ensure that the role of the uncertain factors is not ignored; 2) TSC could make it perform offline training, and then run online by combining with the current traffic conditions. Both accuracy and real-time performance can be expected; 3) it needn't depend on the massive data. At the same time, it takes the characteristics of road condition into account, and supports extensions.

All in all, Contented-based recommendation can be seen as a complement and optimization of Recommendation Systems in TSC. This is the most significant contribution of this paper.

II. METHODOLOGY

In this section, Content-based Recommendations will be implemented to create an ordered list filled with timing plans for current traffic conditions. Here normalized Discounted Cumulative Gain (nDCG) will be performance indicator of the system.

A. Intersection Modeling

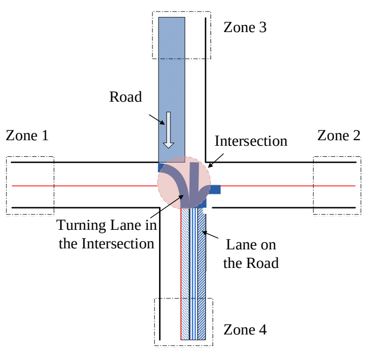

Fig. 1. A typical intersection

To demonstrate the effectiveness of proposed method, the recommendations are conducted on the basis of various traffic condition parameters, such as delay. For ease of understanding, we introduce a single intersection shown in Fig. 1. There are $m=4$ roads in this intersection. And the intersection can be represented as a vector $I=(r_1,r_2,\cdots,r_m)$, in which the element r_i is a 5 dimension vector $r_i=(l_1,l_2,l_3,l_4,l_5)$, where l_1 represents U-turn lane, l_2 represents left turn lane, l_3 represents going straight lane, l_4 represents going straight and right turn lane, l_5 represents right turn lane. They are all represented in sequence, and the numerical value of l_j represents the quantity of each kind of lanes. For example, $r_i=(0,1,2,0,1)$ means there are 1 left turn lane, 2 going straight lane and 1 right turn lane. We assume that the vehicles will follow given paths when turning.

B. Application of Content-based Recommendations

The initial task of Recommendation Systems is to recommend the most suitable items to meet the users' demands, and in this paper it is to find the most suitable timing plans to meet the demands of different traffic conditions. In Content-based Recommendations, we should find some common characteristics among all traffic conditions first. According to these common characteristics, we can get the similarity between arbitrary two traffic conditions. Then the best timing plan can be obtained from those adopted by several most similar traffic conditions.

Based on the model of road network above, we choose different traffic conditions as users in Content-based Recommendations, through configuring different OD (origin-destination) matrices [25]. Then some timing plans C are selected as the items to be recommended. Meanwhile, the delay under every pair of condition and timing plan will be recorded as ratings. So a user-item matrix filled with ratings will be generated. What we should note is that some preprocessed steps should be conducted on the delay data, so that they will be a lot more like ratings.

The process for Content-based Recommendations to recommend the best timing plan can be described as follows :

Step 1: Carry out some preprocesses for the original data. Then a standard rating matrix will come into being.

Step 2: Here the number of vehicles that travel through every two zones is chosen to be the common characteristic among all traffic conditions. Then traffic condition a and b can be represented by vector $(a_1,a_2,\cdots,a_i,\cdots)$ and $(b_1,b_2,\cdots,b_i,\cdots)$.

Step 3: Use Euclidean Metric to get the similarity between arbitrary two traffic demands. The definition of similarity between traffic condition a and b is as follows,

$$sim(a,b)=\frac{1}{1+\sqrt{\sum(a_i-b_i)^2}} \tag{1}$$

Step 4: When certain traffic condition occurs (it is often checked at the end of every cycle), kNN algorithm [19, 20] will choose K most similar traffic conditions, according to the similarity generated in **Step 3**, to predict the delay ratings that the traffic condition gives to other unused timing plans. By taking the average values of K nearest neighbors into account, a possible formula for predicting the rating, traffic condition a gives to timing plan c, is defined as follows:

$$pred(a,c)=\frac{\sum_{b\in K} sim(a,b)*r_{b,c}}{\sum_{b\in K} sim(a,b)} \tag{2}$$

Here $r_{b,c}$ is the rating that traffic condition b gives timing plan c, it may be the reciprocal of delay.

Step 5: After **Step 4**, we can obtain an ordered recommended list $(c_1, c_2, \cdots, c_i, c_{i+1}, \cdots)$. Among them, c_i is the *i*-th most appropriate timing plan, which is better than c_{i+1}.

Step 6: Verify the obtained timing plans in Paramics. According to the results of Paramics, we can get another ordered list $(c_1', c_2', \cdots, c_i', c_{i+1}', \cdots)$.

Step 7: Compare list $(c_1, c_2, \cdots, c_i, c_{i+1}, \cdots)$ with list $(c_1', c_2', \cdots, c_i', c_{i+1}', \cdots)$ to evaluate the effectiveness of prediction.

C. *Performance Indicator*

The essential of "getting a recommended list through predicting the matching degree between current road conditions and unused timing plans" is to search several most appropriate timing plans which constitute an ordered list. We hope to get a high-quality ordered recommended list. So the higher the matching degree is, the more front in the list the timing plan should be. At the same time, all timing plans in the list should be as possible as matching with the current road conditions. Then we adopted nDCG (normalized Discounted Cumulative Gain) as the performance indicator, which is defined as follows:

$$nDCG_p = \frac{DCG_P}{IDCG_P} \qquad (3)$$

where DCG_P represent the DCG accumulated at a particular rank position p. $IDCG_P$ (Ideal DCG) is the maximum possible DCG till position p. $nDCG_P$ is the normalized DCG.

DCG is a measure of ranking quality, it is often used to measure effectiveness of web search engine algorithms or related applications [22]. Using a graded relevance scale of documents in a search engine result set, DCG measures the usefulness, or gain, of a document based on its position in the result list. The gain is accumulated from the top of the result list to the bottom with the gain of each result discounted at lower ranks [23]. An alternative formulation of DCG [24] places stronger emphasis on retrieving relevant documents:

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)} \qquad (4)$$

where rel_i is the graded relevance of the result at position i.

Search result lists vary in length depending on the query. Comparing the performance of searching from one query to the next cannot be consistently achieved using DCG alone, so the cumulative gain at each position for a chosen value of p should be normalized across queries. Then IDCG is introduced. The nDCG values for all queries can be averaged to obtain a measure of the average performance of a search engine's ranking algorithm. Note that in a perfect ranking algorithm, the DCG_p will be the same as the $IDCG_P$ producing an nDCG of 1.0.

III. EXPERIMENTS

In the experiments, we simulate on a road network with single intersection shown in Fig. 1. Then we set some different r_i to fill the intersections mentioned above, so our simulations can cover the common situations. After getting the data for simulation experiments, we can utilize Contented-based recommendation technology to mine the internal relationship of the data, i.e. computing the similarity between arbitrary traffic conditions and obtain the sequential recommended list for the current traffic conditions.

A. *Data Setup*

The simulation road network of single intersection with 4 roads is shown in Fig. 1, i.e. $I = (r_1, r_2, r_3, r_4)$. And each road has 3 lanes, 1 left turn lane, 1 going straight lane and 1 right turn lane, i.e. $r_1 = r_2 = r_3 = r_4 = (0,1,1,0,1)$. In Paramics, the time step is set to be 0.5s, while the duration of simulation is 1 hour. The length of each lane is 1000m.

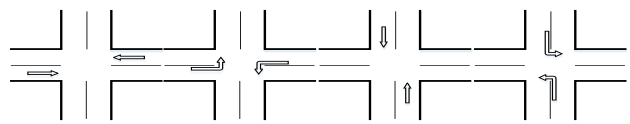

Fig. 2. Phase sequence of the traffic lights

Fig. 2 shows a typical phase configuration in the crossroad. In every timing plan, we set amber time of each phase to be 3s. Note that, right turning is always allowed in the experiments.

In the experiments, we use 40 groups of various ODs to represent 40 different traffic demands which can create 40 different road conditions. And 40 fixed timing plans, which can vary from 88s to 122s, will be contained. The range of the timing plans is an empirical value which could cover a usually used range. The vehicle number can vary from 120 to 23400.

B. *Simple Case*

First, we will show a simple case in which the proposed method will be applied. Meanwhile this case will help readers to understand the method.

Here 9 cross-over experiments between 3 ODs and 3 timing plans will be conducted first, and the delay data of each experiment will be collected. 3 ODs in Paramics are set as TABLE I, II, and III. Among them, OD 1 reflects that the vehicles come from Zone 1 and 2 are obviously more than Zone 3 and 4, while the vehicles come from Zone 3 and 4 are more in OD 2. And in OD 3, the vehicle number of every zone is equal to each other's.

TABLE I. OD 1 IN EXPERIMENTS

	Zone 1	**Zone 2**	**Zone 3**	**Zone 4**	**Total**
Zone 1		800	400	200	1400
Zone 2	800		200	400	1400
Zone 3	100	200		500	800
Zone 4	200	100	500		800
Total	1100	1100	1100	1100	4400

TABLE II. OD 2 IN EXPERIMENTS

	Zone 1	Zone 2	Zone 3	Zone 4	Total
Zone 1		500	200	100	800
Zone 2	500		100	200	800
Zone 3	200	400		800	1400
Zone 4	400	200	800		1400
Total	1100	1100	1100	1100	4400

TABLE III. OD 3 IN EXPERIMENTS

	Zone 1	Zone 2	Zone 3	Zone 4	Total
Zone 1		366	366	366	1098
Zone 2	366		366	366	1098
Zone 3	366	366		366	1098
Zone 4	366	366	366		1098
Total	1098	1098	1098	1098	4392

Here the cycle time is set to be 100s in each timing plan, and 3 timing plans are shown in TABLE IV. Among them, Timing Plan 1 pays more attention to Phase 1 and 2, while Timing Plan 2 pays more to Phase 3 and 4. And Timing Plan 3 treats equally without discrimination.

TABLE IV. 3 TIMING PLANS IN EXPERIMENTS

	Phase 1	Phase 2	Phase 3	Phase 4
Timing Plan 1	30	30	14	14
Timing Plan 2	14	14	30	30
Timing Plan 3	22	22	22	22

The obtained delay data from 9 groups of experiments can be converted into a delay matrix, which can be described by TABLE V. The unit of delay time is seconds.

TABLE V. THE OBTAINED DELAY TIME IN EXPERIMENTS

	OD 1	OD 2	OD 3
Timing Plan 1	2149.397238	6582.446324	2806.645205
Timing Plan 2	9250.780908	2327.28644	3848.699365
Timing Plan 3	4354.481777	4301.957392	519.6162856

In addition to these 3 ODs, we add one more that we will predict the matching degree for under 3 timing plans. The added OD 4 is showed in TABLE VI.

TABLE VI. OD 4 IN EXPERIMENTS

	Zone 1	Zone 2	Zone 3	Zone 4	Total
Zone 1		600	300	300	1200
Zone 2	600		300	300	1200
Zone 3	300	300		600	1200
Zone 4	300	300	600		1200
Total	1200	1200	1200	1200	4800

Then we can compute the similarity between OD 4 and other three ODs by using Content-based Recommendation method, which is shown in TABLE VII.

TABLE VII. THE SIMILARITY BETWEEN OD 4 AND OTHER THREE ODS

	OD 1	OD 2	OD 3
OD 4	2.037083274	2.037083274	1.980758427

By using the obtained data mentioned above, we can predict the matching degree between OD 4 and 3 timing plans through the method mentioned in section II. Here the result of prediction is that Timing Plan 3 is best matching with OD 4, Timing Plan 2 is worst. On the other hand, the result obtained from Paramics showed that Timing Plan 3 is still best, while Timing Plan 2 is still worst, i.e. nDCG here is 1. This is the same with the predicted result.

C. Further Simulation

However, the problem will not be so easy like the simple case showed above. In reality, it is impossible for every traffic condition to traverse all kinds of timing plans, and obtain the delay information won't be so integrated. So the delay matrix (TABLE IV) in section III should be sparse. The more practical problem is to predict the matching degree between traffic conditions and its unused timing plans in a sparse matrix.

First, we got a sparse matrix filled with delay time by simulating with 40 ODs and 40 timing plans in Paramics, and here the sparse degree is 75%. Then through the methods mentioned in section II, we recommended 6 unused timing plans to each OD. Then the sequential recommendation list $L_i = (c_1, c_2, c_3, c_4, c_5, c_6)$ will be obtained, where L_i represents the list of *i*-th OD. Furthermore, we simulated with each OD under its 6 recommended timing plans in L_i in Paramics. Then the ideal sequential list $L_i' = (c_1', c_2', c_3', c_4', c_5', c_6')$ for each OD would be got according to the delay data. We assumed that the matching degree between OD_i and $c_j, j = 1,2,3,4,5,6$ in L_i' is 5, 4, 3, 2, 1, and 0 in turn. Then nDCG of L_i for each OD can be calculated, and the results are shown as follows.

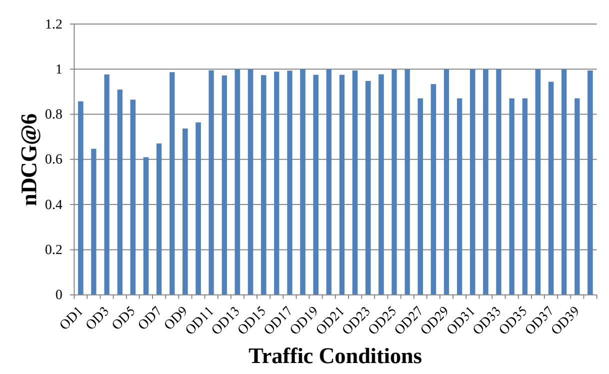

Fig. 3. nDCG@6 for 40 traffic conditions

Fig. 3 shows a so good performance that nDCG values of all ODs are more than 0.6. What's more, most of them are nearly or equal to 1. The results seem to be so perfect, and it demonstrates the effectiveness of Content-based recommendation for TSC.

D. Comparison Experiment

Further, we can easily find that every c_1 in L_i is the best unused timing plan for OD_i. Then we should compare this c_1 with used timing plans to obtain the final best timing plan c_{best}.

Additionally, corresponding to 40 different traffic demands, we will get 40 optimal signal control plans through Webster method [4]. When adopting Webster method, the related parameter settings are shown in TABLE VIII, in which *AR* represents the all-red time with the unit second, *l* is the average loss time per phase in seconds (excluding any all-red periods or sequent ambers), and *S* is the saturation flow with the unit vehs/h.

TABLE VIII. WEBSTER PARAMETERS

AR (s)	*l* (s)	*S* (vehs/s)	Minimum green time (s)	Maximum green time (s)	Optimum cycle (s)
0	5	2200	10	60	C_0

Then compare the results of 40 ODs with their own c_{best} with those of Webster method, we can get the following Fig. 4.

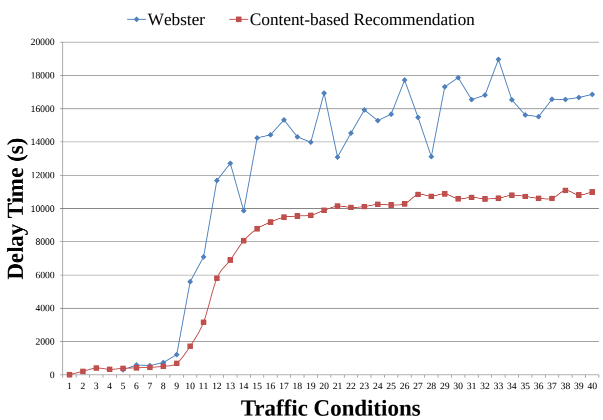

Fig. 4. Delay comparison

From Fig. 4, we can find that, compared with Webster method, the overall delay time under Content-based Recommendation algorithm is far less. Furthermore, the figure shows an upturn in the difference between two delays when traffic demand is situated between 9 and 20. Nonetheless, in the first 5-8 traffic demands, there is no obvious difference between two methods. It is easy to imagine that ordinary fixed-time control strategies will be difficult to show their advantages when traffic demand is too small. Similarly, when traffic demand is too large, delay time will fluctuate within a small range. In addition, we should point out that there is no experimental data under Webster method when traffic demand is 1, 2, 3 and 4. This is because the Webster method has a scope of application. Usually, the sum Y (traffic flow rate) of y_i in Webster should be kept between 0.4 and 0.9 [4]. But in these cases, Content-based Recommendation algorithm works well.

IV. CONCLUSIONS

This paper is inspired by the idea of recommendation systems, and 'search' the high matching degree between unused timing plans and each traffic condition through Contented-based Recommendation technology. The experiments are based on ATS, and the results show that the feedback sequential recommended lists are high related with the ideal sequential list, which proves this method works well. Then, we compare the method with classic Webster, and it is gratifying that Content-based Recommendation method outperform Webster obviously.

We may focus on the following aspects in the future: 1) take more factors which can affect traffic environment and represent the traffic states into account; 2) combine the Content-based Recommendation method with Collaborative Filtering method.

REFERENCES

[1] Z. Liu, "A survey of intelligence methods in urban traffic signal control," IJCSNS International Journal of Computer Science and Network Security, vol. 7, no. 7, pp. 105-112, July 2007.

[2] D. Zhao, Y. Dai, and Z. Zhang, "Computational intelligence in urban traffic signal control: A survey," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. PP, no. 99, pp. 1-10, 2011.

[3] C. Chen, F.-H. Zhu, and Y.-F. Ai, "A survey of urban traffic signal control for agent recommendation system," in 2012 15th International IEEE Conference on Intelligent Transportation Systems, Anchorage, AK, 2012, pp. 327-333.

[4] F. V. Webster. (1958). Traffic signal settings. Great Britain Department of Scientific A. [Online]. 39. Available: www.bl.uk/services/document/lps.html

[5] M. J. Lighthill, and G. B. Whitham, "On kinematic waves. II. A theory of traffic flow on long crowded roads," Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, vol. 229, no. 1178, pp. 317-345, 1995.

[6] P. I. Richards, "Shock waves on the highway," Operations research vol. 4, no. 1, pp. 42-51, 1956.

[7] S. Chen, and G. D. Doolen, "Lattice Boltzmann method for fluid flows." Annual review of fluid mechanics, vol. 30, no. 1, pp. 329-364, 1998.

[8] C. Chen, F.-H. Zhu, and Y.-F. Ai, "A survey of urban traffic signal control for agent recommendation system," in 2012 15th International IEEE Conference on Intelligent Transportation Systems, Anchorage, AK, pp. 327-333, 2012.

[9] F.-Y. Wang, and S.-M. Tang, "Concepts and frameworks of artificial transportation systems," Complex Systems and Complexity Science, vol. 1, no. 2, pp. 52-59, 2004 (in Chinese).

[10] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 3, pp. 630-638, Sept. 2010.

[11] F.-Y. Wang, "Agent-based control for networked traffic management systems," IEEE Intelligent Systems, vol. 20, no. 5, pp. 92-96, Sept.-Oct. 2005.

[12] Zhao, Y. F., Gao, H., Lv, Y. S., & Duan, Y. J. (2014, October). Latent factor model for traffic signal control. In Service Operations and Logistics, and Informatics (SOLI), 2014 IEEE International Conference on (pp. 227-232). IEEE

[13] Zhao, Y. F., Kong, Q. J., Gao, H., Zhu, F. H., & Wang, F. Y. (2014, October). Parallel management for traffic signal control. In Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on(pp. 2888-2893). IEEE

[14] Y. Koren, et al, "Matrix factorization techniques for recommender systems," Computer, vol. 42, no. 8, pp. 30-37, 2009.

[15] D. Billsus and M. J. Pazzani, "Learning Collaborative Information Filters," ICML, 1998.

[16] D. Goldberg, et al., "Using collaborative filtering to weave an information tapestry," Communications of the ACM, vol. 35, no. 12, pp. 61-70, 1992.

[17] Pazzani M J, Billsus D. Content-based recommendation systems[M]//The adaptive web. Springer Berlin Heidelberg, 2007: 325-341.

[18] Pazzani M J. A framework for collaborative, content-based and demographic filtering[J]. Artificial Intelligence Review, 1999, 13(5-6): 393-408.

[19] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," ACM Trans. Inf. Syst., vol. 22, no. 1, pp. 5-53, Jan. 2004.

[20] R. M. Bell, and Y. Koren, "Scalable collaborative filtering with jointly derived neighborhood interpolation weights," in Seventh IEEE International Conference on Data Mining, Omaha, NE, 2007, pp. 28-31.

[21] Z.-S. Hou, and S.-T. Jin, Model Free Adaptive Control: Theory and Applications, CRC Press, 2013.

[22] http://en.wikipedia.org/wiki/Discounted_cumulative_gain

[23] Kalervo Jarvelin, Jaana Kekalainen: Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems 20(4), 422–446 (2002)

[24] Burges C, Shaked T, Renshaw E, et al. Learning to rank using gradient descent[C]//Proceedings of the 22nd international conference on Machine learning. ACM, 2005: 89-96.

[25] Gao H, Zhao Y F, Li S S, et al. Paramics-based PD-type iterative learning control for on-ramp metering[C]//Service Operations and Logistics, and Informatics (SOLI), 2014 IEEE International Conference on. IEEE, 2014: 344-349.