Latent Factor Model for Traffic Signal Control

Y.-F. Zhao
Institute of Smart City Systems
Qingdao Academy of Intelligent Industries
Qingdao, China
yfyf.zhao@163.com

H. Gao, Y.-S. Lv and Y.-J. Duan
Qingdao Center for Intelligent Systems and Technology
Institute of Automation, Chinese Academy of Sciences
Qingdao, China
gaohang@casc.ac.cn, yisheng.lv@ia.ac.cn,
Catherine.dyj@163.com

***Abstract*—The increased ownership of motor vehicles has brought many urban problems, such as traffic congestion, environmental pollution. Traffic signal control is recognized as one of effective ways to alleviate these problems. However, it is still hard to automatically choose appropriate traffic signal timing plans for different traffic conditions due to the dynamics and uncertainty of transportation systems. In this paper, we propose a latent factor model based traffic signal timing plan recommendation method to address this problem. In the proposed method, we model the abstract traffic states as the "users" in recommendation systems, and timing plans as the "items". And there are many explicit or implicit factors in the interactions between "users" and "items". The latent factor model is successfully used to deal with uncertain factors which cannot be modeled accurately in math. The novel method adopted the model-free adaptive idea to solve the problem of modeling from the perspective of data mining and machine learning framework. And, the proposed method is tested by using simulation data generated by a microscopic traffic simulator called Paramics. The results are compared to the baseline Webster method. The results indicate that the proposed latent factor model based recommendation method outperforms the Webster method on reducing the delay.**

***Keywords*—Collaborative Filtering, Intelligent Transportation Systems, Recommendation System, Signal Control**

I. Introduction

Recommendation Systems (RS) have evolved and diversified rapidly since their inception around the early 1990s [1, 2] as the key concept and method in the complex systems to solve the problem called "information overload" [3]. They have become an established, promising research and application field drawing on and bringing together results and concepts from many disciplines, including AI, computer science, sociology, economics, psychology, data mining, and machine learning. They have many successful applications in e-commerce [4], movie and video websites [5, 6], personalized music radio, social network, etc., which are all complex systems with strong randomness and fuzzy. The essential function of RS is to find the best matching items for the users in an "information overload" environment [2]. In RS, users often refer to the various people, and items may be potential interesting merchandise, movies, videos, friends, and so on. The huge amount information may be useful or useless, definite or random, fuzzy and difficult to model, etc.

This work is supported in part by NSFC (Natural Science Foundation of China) projects (71232006,61104160, 61203166, 4142055).

However, RS has yet to achieve widespread use for controlling traffic management systems, because there are many common characteristics between them, such as randomness, fuzzy, and nonlinear [7]. In the traffic management system, Traffic Signal Control (TSC) is an effective way to improve traffic states, and it can be understood as finding the best matching timing plan for the current traffic state. Many control methods have prevailed in both theoretical studies and practical applications. The theoretical research work about TSC could date back to the mid-20th century [8], which proposed a classical control strategy called the Webster method. As the computing technologies are developed, the microscopic traffic models become more and more popular, such as Multi-Agent Systems (MAS) [9-13]. Based on MAS, an Artificial Transportation Systems (ATS) can grow up in a bottom up way [9, 14], with drivers, vehicles, roads, traffic lights being modeled as autonomous, collaborative and reactive agents. And there are many academic and commercial traffic simulators such as TSIS, TRANSIMS, PARAMICS and TransWorld [9].

Although the traditional control strategies have been confirmed to have good effects, we have to admit that it still needs to be improved. In the traditional methods, relatively accurate models tend to be large amount of calculation, complex process, and difficult to achieve. We often choose to sacrifice some accuracy to seek a balance point which can ensure certain speediness and real time. It causes that many theoretical methods based on the quantitative and idealized hypothesis cannot be applied in reality, or cannot achieve the desired results. The main reason for the problems mentioned above is that, there are too many uncertain factors which are difficult or even impossible to model exactly in a closed form, such as vehicles and pedestrians, roads, weather environment, legal policies [9, 14]. We need to rethink traffic management systems and reinvestigate the use of RS for TSC.

TSC is well suited to a RS method because of the consistency of their essential task and the successful way of RS dealing uncertain factors that difficult or unable to model. In future we will not lack of traffic data. Because detecting technologies are more and more mature and so many models and algorithms can generate so many timing plans. So we can learn from RS. RS think the ratings can reflect the users' preference for the items, and use Latent Factor Models (LFM)

[15] to solve the problems about uncertain factors. LFM is an alternative approach that tries to explain the ratings by characterizing both items and users on, say, 20 to 100 factors inferred from the ratings patterns. Some of the factors are explicable, for example, for movies, the discovered factors might measure obvious dimensions such comedy versus drama, amount of action, or orientation to children; some are less well-defined dimensions such as depth of character development or quirkiness; some are even completely uninterpretable dimensions. For users, each factor measures how much the user likes movies that score high on the corresponding movie factor. There are also some indicators that can reflect the "preference" of the traffic states for the timing plans, such as delay, stop time, flow. We can also try to explain them by characterizing both traffic states and timing plans on several factors inferred from the "ratings" patterns. The factors maybe intersection environment nearby, weather condition, the structure of the roads, etc., and more factors can't be accurately explained, but it doesn't matter for most of the factors we don't need to know. From the perspective of machine learning and data mining, thousands of factors may be not known but good results can also achieve, which has been proved in RS when using LFM [16]. In this paper, we try to using LFM to solve the problem of modeling for uncertain factors in TSC, as a complement and optimization for traditional strategies, and this is the most important contribution of our work.

A brief introduction of related algorithms is given in section 2. In section 3, the modeling and methodology is presented, and experiments are demonstrated in section 4. Section 5 gives the conclusions and future work.

II. Literature Review

In this section we will introduce the realization of LFM combining matrix factorization and RS in movies, and classic Webster method in TSC will also be given.

A. Matrix Factorization Methods

In RS the most successful realizations of LFM are based on the matrix factorization, which characterizes both users and items with factor vectors inferred from user-item rating matrix. For example, each item is associated with a vector $q_t \in \mathbb{R}^f$, and each user is represented by a vector $p_s \in \mathbb{R}^f$. Here f is the number of latent factors. Then matrix factorization models map both items and users to a joint latent factor space of dimensionality *f*, a recommendation occurs in high correspondence between user and item factors, and the user-item interactions are modeled as inner products in that space. For example, every movie has some elements like history, disaster, love, or other uninterpretable factors. From the mathematical perspective, the idea of matrix factorization is to break a $m \times n$ rating matrix *R* down into a $n \times f$ user factor matrix *P* and an $m \times f$ item factor matrix *Q*, as shown in formulation (1),

$$R[u][i] = P[u][k] * T\left(Q[i][k]\right) \quad (1)$$

Here, *R*[*u*][*i*] represents the rating of user *u* for item *i*, *P*[*u*][*k*] measures the degree of interest the user has in item factor *k*, *Q*[*i*][*k*] represents the share the element *k* owns in the item *i*, and *T*(*Q*) represent the transpose of matrix *Q*.

Here we will give an example about matrix factorization. TABLE I, II, and III represent the rating matrix *R*, user factor matrix *P*, and item factor matrix *Q*, respectively. And Figure 1 shows the relationships among them. From TABLE I, we can see that the rating Andy gave Titanic is unknown. But we know the scores Andy gave other two films and Hang gave all three films. What's more, each film embodies three elements to varying degrees, and the degree of preference two persons have for three film elements is known. Then we could make a prediction about the degree Andy likes the movie Titanic according to the existing information aforementioned. Here we could easily predict that Andy will give a high rating to Titanic as it contains two film elements which are Andy's favorites.

TABLE I. Rating Matrix *R*

Rating matrix R	2012	Pearl Harbor	Titanic
Hang	3	4	3
Andy	3	4	?

TABLE II. User Factor Matrix *P*

User factor matrix P	love	history	Disaster
Hang	0.3	0.8	1
Andy	1	0.5	0.7

TABLE III. Item Factor Matrix *Q*

Item factor matrix Q	love	history	Disaster
2012	0	0	5
Pearl harbor	4	4	4
Titanic	5	3	4

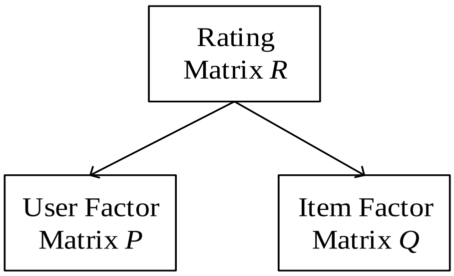

Figure 1. Matrix Factorization

After getting the parameters of (1), we can use (2) to predict the rating of user *u* for movie *i*,

$$\hat{r}_{u,i} = p_u q_i^T \quad (2)$$

where q_i represents the share each element *i* owns, and p_u denotes the preference of user *u* for elements.

B. Parameter Learning

There is a question in the example aforementioned: there is a missing entry. How can the rating matrix be broken up into two complete matrixes? In fact, the rating matrix is often sparse, as users usually do not give a rating to each movie. Early researches relied on imputation to fill in missing ratings, and make the rating matrix dense. But the imputation could be very expensive as the data increased, and inaccurate imputation might distort the data considerably. So machine learning is taken into account. [17, 18, 19] model the known ratings directly, and avoid over fitting through a regularized model. To obtain the parameters in (1), the system should try to minimize the cost function (3) according to the set of known ratings:

$$\min \sum_{(u,i)\in\alpha} (r_{u,i} - p_u q_i^T)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2) \tag{3}$$

where α is the training set in machine learning, and the rating r_{ui} in α is known. λ is a constant which controls the degree of regularization, and it is usually determined by cross-validation. Many optimization approaches can applied to minimize (3), such as stochastic gradient descent and Alternating Least Squares (ALS).

C. Webster Signal Control

So far the fixed-time signal control mainly includes Webster method in Britain, ARRB method in Australia, HCM method in America and so on. Here we try an experiment on Webster method, which is based on the vehicles' delay time when travelling through the intersection. And the output parameters in Webster method contain the cycle time, delay time, effective green time of every phase etc.

The control target of Webster method is to minimize the total delay of the vehicles. Then the best signal cycle can be given by

$$C_0 = \frac{1.5L+5}{1-Y} \tag{4}$$

where C_0 is the best signal cycle time with the unit seconds. L is the total loss time in a cycle with the unit seconds, Y is the traffic flow rate. In fact, we often take $0.75C_0$ to $1.5C_0$ as the best cycle time.

The total loss time can be described as follows,

$$L = nl + AR \tag{5}$$

where l is the loss time of every phase, n is the phase number. Here AR stands for the all-red time in a cycle.

The traffic flow rate Y is the sum of the flow rate of all lanes with the maximum traffic flow in every phase, and Y can be described by

$$Y = \sum_{i=1}^{n} y_i \tag{6}$$

Critical lane refers to the lane which has the largest traffic flow in each signal phase. Traffic flow rate of critical lane is equal to the ratio of the traffic flow to the saturation flow.

III. Modeling And Methodology

In this part, we will first model for "users", "items" and "ratings" in the transportation systems. Then the elaboration for the application of matrix factorization will be given.

A. Modeling for "Ratings"

The "ratings" in the transportation systems should reflect the degree of traffic states' "preference" for timing plans, i.e. whether the timing plans are suitable to the traffic states. And we should know the improvement of traffic states according to the "ratings". So that we could judge whether the timing plans are better than the original one. In order to identify the congestion degree scientifically, and compare the pros and cons of different control strategies, we may select the delay time as the performance indicator. Because it is significant in traffic signal retiming and coordination of existing signals [20].

Delay displays statistics for links according to average delay per vehicle. The units for delay are seconds and it is defined as the actual time taken by vehicles to traverse the link minus their free flow time.

What we should note here is that, in RS, the higher the rating is, the better, while in the transportation systems, the lower the value of delay is, the better. The value of delay time may range very large, so the pre-process is necessary.

B. Modeling for "Users" and "Items"

Our goal is to find the most suitable timing plan for the current traffic state. So unquestionably the "items" would be timing plans, and the "users" would be the abstract traffic states. In this case, each timing plan is associated with a vector $q_t \in \mathbb{R}^f$, and each traffic state is represented by a vector $p_s \in \mathbb{R}^f$.

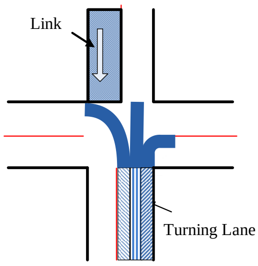

Figure 2. An intersection

Abstract traffic states should be based on the intersections. The model of a single intersection is shown in Figure 2. There are several lanes on each link. We model the vehicles in a lane as a queue with the fixed capacity. No more vehicles will enter the queue when the capacity is exhausted. The right most and left most are set to be turning lanes. When turning, we assume the vehicles follow given paths.

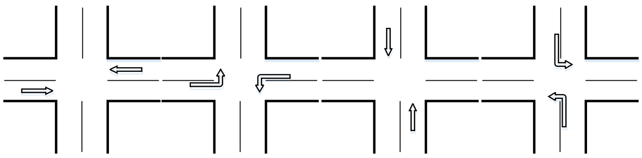

Figure 3. Phase sequence of the traffic lights

There are usually several phases in the traffic lights of an intersection. Figure 3 shows a common phase sequence which represents a general order of different phases. 4 phases are included, and they constitute a "cycle" that is the time for an intersection to traverse all phases at a pre-set order.

C. Adding Biases

However, a lot of intersections are influenced by different intrinsic properties and environment in reality, which may make traffic congested or unobstructed when the same traffic flow appears. There are some cases that can explain this phenomenon.

Case 1, the structure of the intersection. Some structures of the intersection are prone to traffic jams. For example, road A shown in Figure 4 is smaller than other three roads.

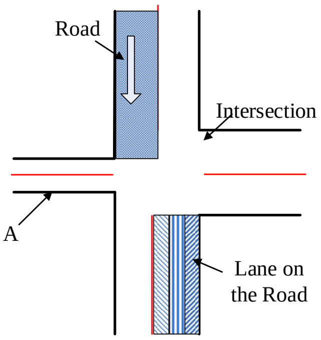

Figure 4. Road A is narrow in Case 1

Case 2, the special environment nearby. For example there is a school near road B, shown in Figure 5. Usually there should be a traffic light for the students to cross the road. In this case the traffic is not smooth.

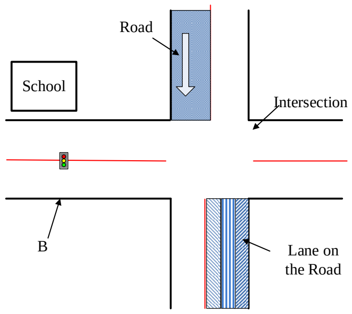

Figure 5. Road B with a traffic light

Case 3, if the four roads of the intersection are wide enough, congestion will be not likely to happen.

In these cases the inherent properties of traffic states are different, some may lead to large delay, while some do not. It is similar to partial users like giving low scores and moderate users enjoy giving high marks in RS. In the same way, some timing plans with unreasonable design is easy to cause traffic jams, while some may be generated by accuracy models and algorithms, which is perfect for most situations. These are the biases or intercepts of themselves, independent of any interactions of the traffic states and the timing plans, which should be considered and can express by the following formula,

$$b_{st} = u + b_s + b_t \tag{7}$$

The bias involved in "rating" r_{st} is denoted by b_{st} and accounts for the traffic states of intersection and timing plan effects. The average rating is denoted by u, while the parameters b_s and b_t indicate the observed deviations of traffic state s and timing plan t, respectively, from u.

D. Application of Matrix Factorization

According to what have been mentioned in section II, the matrix factorization model here should map both traffic states and timing plans to a joint latent factor space with the dimensionality f. The dimensionality f relates to the factors like weather conditions and pedestrian, or even other factors which cannot be modeled. But in fact they affect the traffic states and should not be ignored. So the interactions should be modeled as inner products in the space as shown in formula (8), which is called Model 1.

$$\hat{r}_{st} = q_t^T p_s \tag{8}$$

Take the bias into account, then the prediction model is called Model 2 shown in formula (9),

$$\hat{r}_{st} = u + b_s + b_t + q_t^T p_s \tag{9}$$

To learn the parameters, the systems should minimizes the regularized squared error in the set α where the delays of pairs (traffic state, timing plan) are known. And the formulation for (8) is

$$\min \sum_{(s,t)\in\alpha} (r_{st} - p_s q_t^T)^2 + \lambda(\|p_s\|^2 + \|q_t\|^2) \tag{10}$$

The formulation for (9) is

$$\min \sum_{(s,t)\in\alpha} (r_{st} - u - b_s - b_t - p_s q_t^T)^2 + \lambda(b_s^2 + b_t^2 + \|p_s\|^2 + \|q_t\|^2) \tag{11}$$

The process is as follows:

Step 1: normalize the original dataset and divide it into M sets randomly according to the uniform distribution. Then choose one as test set, and other M-1 sets are training sets.

Step 2: in the training set, use stochastic gradient descent method to get the minimum of formulation (10) and (11). Then the parameters in formulation (8) and (9) can be inferred.

Step 3: predict the delay ratings through the test set, and get the RMSE. Obviously, the smaller the RMSE is, the better.

Step 4: use the best model aforementioned to predict the delay ratings of the current traffic state for the timing plans that never be used before, and choose the best one. Then compare the best one with the delays of other timing plans that the traffic state has used before to decide the final best timing plan.

Step 5: After applying the timing plan in the actual operation, the system should record the newest delay of the pair of traffic state and timing plan, to further enrich database.

IV. Computational Experiments

A. Description of Dataset

The experiment data is generated by microscopic traffic simulation software Paramics, and the simulation network is shown in Figure 2. Among them, 4 roads all have two links, every link has three lanes, and the length of each link is 500 meters. The duration of simulation is 1 hour, while the time step is set to be 0.5s.

When performing simulation, we collect the data from the 10th minute to the end of the simulation, because the first ten minutes are used as road initialization time. In the simulation, we set 7 kinds of different traffic ODs representing 7 different traffic states from unimpeded to congested states. And 22 signal timing plans were generated. Then 154 groups of delay values and 154 groups of stop time are obtained. In reality, it is impossible that every intersection has used every kind of timing plans. So we randomly choose 70 groups from the 154 groups above as the known experiment data.

Additionally, we get 7 optimal signal control plans according to the Webster method. In the Webster method, the related parameters are shown in TABLE IV.

TABLE IV. Webster Parameters

AR (s)	L (s)	S (vehs/s)	minimum green time (s)	maximum green time (s)	optimum cycle (s)
3	5	2000	7	41	C_0

B. Training and Selecting

We divide the dataset into 7 subsets, then train the model through k-fold cross-validation and stochastic gradient descent method. We tried many different parameterizations for factorization. Here we set k = 4, the learning rate r = 0.01, and the regularization parameter λ = 0.05. We adjust the number of latent factors f to find the best model. The RMSE of models without bias and with bias are shown in Figure 6.

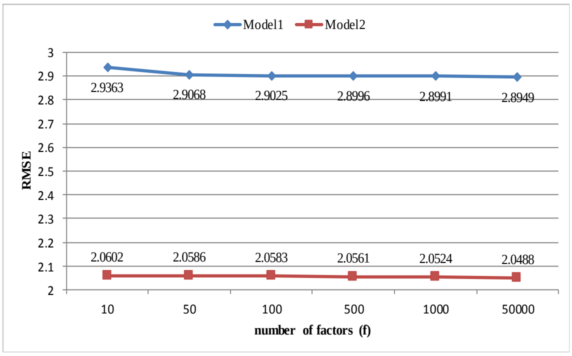

Figure 6. RMSE of models with different parameters

From Figure 6 we can obtain two conclusions. One is that the overall performance of Model 2 is superior to Model 1, i.e. the performance of the model with bias is better; the other is that the accuracy of each factor model is improved by increasing the number of involved parameters, which is equivalent to increasing the dimensionality of factor model. In conclusion we would like to choose Model 2. And its best performance is RMSE = 2.0488, while Model 1's is 2.8949, which is much higher than Model 2's.

C. Result Analysis

Based on the model selected above, we predicted the delay ratings for 7 traffic states under different timing plans which haven't been used before. Then according to the predictions, 7 best matching timing plans can be chosen which is described in section III. The cycles of timing plans and the corresponding delays are shown in TABLE V (the unit of delay is seconds).

TABLE V. Delay under LFM

	S1	S2	S3	S4	S5	S6	S7
C_i	120	72	40	48	88	148	120
Delay	176.1	65.3	35.4	53.7	111.2	280.4	1144

TABLE VI. Delay under Webster

	S1	S2	S3	S4	S5	S6	S7
Cwi	140	88	64	72	104	176	176
Delay	228.3	86.4	49.9	60	129.2	541.3	1297.2

The best cycles from Webster are $C_{w1} \sim C_{w7}$, and the corresponding delays are shown in TABLE VI (the unit of delay time is seconds).

More intuitively, Figure 7 shows the comparison between the two results:

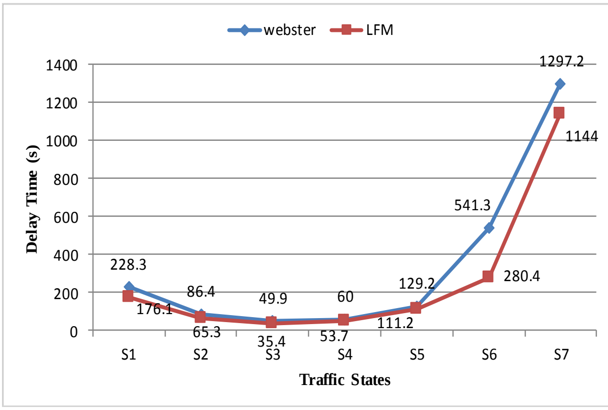

Figure 7. Delay comparison

From Figure 7, we can deduce the conclusion that under the same traffic demand, the overall delay of timing plan when using latent factor model is less than that of Webster method. But there are some delays of LFM that are very close to the corresponding delays of Webster, which have no definite advantage, such as S3 and S4. And although some results of LFM outperform that of Webster, they are still barely satisfactory, such as S6 and S7. Because there are no better timing plan in the database, LFM cannot mine a more ideal timing plan for the traffic states. However, the experiment results have proved that LFM idea in RS outperforms the Webster, and we can make it perform better through two possible aspects. One is about the dataset. As LFM is based on machine learning framework, the ample dataset is necessary. So we may establish a database which includes the delay of pairs of timing plans and traffic states as many as possible, and update or enrich with the method mentioned in this paper continuously. Another is the accuracy of the model. Although LFM can consider many factors that are clear or fuzzy, they are all static. Transportation system is a complex system of dynamic change with high speed, so the neighbor information in time domain should be taken into account in order to be closer to the reality. These are what we could do to improve the traffic states better. Anyhow, the methods have proved that it can be used into traffic signal control as a complement and optimization to traditional timing strategies.

V. CONCLUSION & FUTURE WORK

In this paper we applied LFM into TSC from the perspective of interactions between traffic states and timing plans. It can effectively model the uncertain factors which are difficult or even impossible to be modeled by traditional control strategies. What's more, we trained some models to optimize the traffic states, then compared with the classic method Webster. The experiments achieved a good result, which could prove the feasibility of our new idea.

In the future, we will focus on the following aspects: 1) the establishment of database which contains enough data of various traffic states and timing plans, 2) the usage of other models and algorithms in RS and the optimization of them.

REFERENCES

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

[2] P. Resnick and H. R. Varian , "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56-58, 1997.

[3] J. O'Donovan and B. Smyth, "Trust in recommender systems," *Proceedings of the 10th International Conference on Intelligent User Interfaces*, ACM, 2005.

[4] M. Levene, An introduction to search engines and web navigation, John Wiley & Sons, 2011.

[5] http://cdn-0.nflximg.com/us/pdf/Consumer_Press_Kit.pdf

[6] J. Davidson, et al., The YouTube video recommendation system. *Proceedings of the fourth ACM conference on Recommender systems*, ACM, 2010.

[7] 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.

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

[9] F. Y. Wang, "Parallel Control and Management for Intelligent Transportation Systems: Concepts, Architectures, and Applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, pp. 630-638, 2010-01-01 2010.

[10] B. Chen and H. H. Chen, "A review of the applications of agent technologies in traffic and transportation systems," *IEEE Trans. on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 485-497, Jun. 2010.

[11] Z. Shen, K. Wang and F. Zhu, "Agent-based traffic simulation and traffic signal timing optimization with GPU," *Proceedings of the 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 145-150, Washington D. C., 2011.

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

[13] Y. S. Lv, et al., "Real-Time Highway Traffic Accident Prediction Based on the k-Nearest Neighbor Method," *IEEE Int Conf. Measuring Technology and Mechatronics Automation*, vol. 3, pp. 547-550, 2009.

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

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

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

[17] S. Funk, "Netflix Update: Try This at Home," Dec. 2006, http://sifter.org/~simon/journal/20061211.html.

[18] Y. Koren, "Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model," *Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, ACM Press*, pp. 426-434, 2008.

[19] A. Paterek, "Improving Regularized Singular Value Decomposition for Collaborative Filtering," *Proc. KDD Cup and Workshop, ACM Press*, pp. 39-42, 2007.

[20] D. Srinivasan, et al., "Neural networks for real-time traffic signal control," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, no. 3, pp. 261-272, 2006.