Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2014, Article ID 514295, 8 pages
http://dx.doi.org/10.1155/2014/514295

Research Article

Research and Application of Personalized Modeling Based on Individual Interest in Mining

Baocheng Huang and Guang Yu

School of Management, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Baocheng Huang; huanghljhrbin@gmail.com

Received 25 May 2014; Accepted 12 July 2014; Published 5 August 2014

Academic Editor: Josep M. Rossell

Weibo services, provided by the service providers, is simple and changeless. The research based on the content of microblog reflects the user's personalized features. The method has important significance to improve user satisfaction and expand the scale of users. First, the interest classification problem called multiclass classification algorithm is proposed based on improving support vector machine of binary tree. Second, an improved model of mixed interest based on implicit feedback is proposed. This method is based on the shortcomings of the establishment of the interest model and the drift strategy in update phase among existing users. The improved model is applied to the user modeling of personalization, improving the authenticity and accuracy of the personalized modeling.

1. Introduction

The main purpose of the interest classification method is to classify the interest information and this interest classification information will be used to create the personalized interest model of weibo users. Because weibo users' interests in certain topics will continue for a long time, such as athletes' interests in sports which will be long term and even last for a lifetime, so the theme of interest in these categories in the general case will be a long term, and weibo users will often update weibo in association with the interest in this subject. Weibo users may, however, only in a specific time period focus on a particular interest in the category; it will be shown that the user within a period of time frequently updated weibo in relation to a topic, but after this period of time, the user will seldom update weibo in relation to this topic. When the World Cup is held, the user will focus on information related to the World Cup and update the weibo about the World Cup; after the end of the World Cup, the user pay little attention to the interest category, so they seldom update weibo in association with this category.

Short-term interest model only considers the user's immediate interest, ignoring the long-standing interest. These interests of users are formed for a long time. Although some interest categories of weibo hardly renew, these interest categories should not be excluded by the algorithm. Long-term interest model focuses too much on the time factor, neglecting to take the initiative to find new users' interests. Aiming at the shortcomings and deficiencies of short-term and long-term interest model, this paper proposes a hybrid model to optimize the interest model, through the comparison showing that this model reflects the real user's interest even more.

2. Interest Storage

In order to have a more accurate representation for users' interests, the text takes feature vector to represent the microblog information. Text feature vector is composed of feature words and their corresponding weights, which is represented in terms of their ability to document the importance of that feature words in a document for theirself. In other words, the more important a feature word is in the document, the higher its weights are. Currently, the researches in which many researchers study the term weights are relatively mature, and TF-IDF is now widely used as the calculation method [1]. The formula is as follows:

$$w_{ik} = tf_{ik} \log\left(\frac{N}{n_k} + 0.01\right). \tag{1}$$

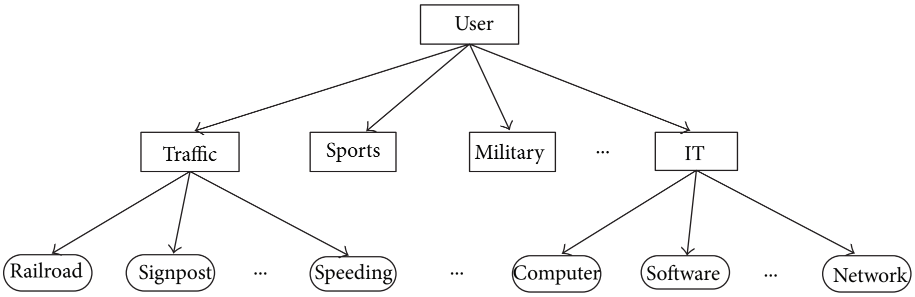

Figure 1: The logic structure of user interest model.

Among them, w_{ik} represents the weight of feature word in the document$_i$, f_{ik} represents the frequency of feature word$_k$ appearance in the document$_i$, N represents the number of documents, and n_k represents frequency of feature word$_k$ in the document [2]. The TF-IDF algorithm considers the relationship of feature word$_k$ in the entire document collection but does not consider the distribution of feature words on each interest category, so the accuracy of the weight of this will have some impact.

Currently, feature word weight algorithms have got some relatively mature calculation methods, but these methods still have many shortcomings and deficiencies. Many domestic and foreign researchers have been conducting research related aspects, and some researchers have found some reasonable weight algorithm. The weight of feature words is to be calculated based on the location where feature words are in the document and words' frequency:

$$p\left(f_{ti}\right) = \frac{\left(\operatorname{freq}\left(f_{ti}\right) + N_{\text{title}} + 0.5 \times N_{\text{begin}} + 0.5 \times N_{\text{end}}\right)}{\sum \operatorname{freq}\left(f_{ti}\right)}. \tag{2}$$

Algorithm 1. The calculation of weibo feature word weight.

Step 1. Statistics weibo's number N of all contents in interest category in this time period.

Step 2. First, find the feature words' set, $t = \{t_1, t_2, \ldots, t_m\}$; then this t is used as feature words' candidate set of user's interest categories vectors.

Step 3. Calculate document frequency n_i of feature word t_i.

Step 4. Use the method of TF-IDF-MI to calculate each feature word weights in a candidate set of feature words:

$$w_i = \mathrm{TF}_i \times \mathrm{IDF}_i = \sum_{j=1}^{n_j} tf_{ij} \times \log\left(\frac{N}{n_i}\right) \times \mathrm{MI}\left(t_i, c\right). \tag{3}$$

Among them, tf $(i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, N)$ represents the weights of feature word t_i, $\mathrm{MI}(t_i, c)$ indicates the mutual information value of the feature word t_i and interest categories. When the feature word's weight of each interest category is finished, you can get the user's feature vector in each interest category.

User interest model not only records the interest content, but also needs to record other information, such as interest update or time's creation and interest weights, in order to provide personalized service. For the user interest model, how to store the user interest model is very important. The interest tree is used to store the user interest model.

In this paper, the user interest model (including the long-term interest model, the short-term interest model, and the mixed interest model of optimization) uses the vector space model to represent. Vector space model is a user interest model with n-dimensional feature vectors $\{(c_1, w_1), (c_2, w_2), (c_3, w_3)\}$ to represent. Each dimension of the feature vector represents a type of user interest and the extent of its interest in the type of interest [3]. Representation of the vector space model can not only reflect the degree of interest in the various interest categories in the user model, but also relatively be easy to provide personalized service for users by vector calculation. Therefore, the user interest model of this paper logically forms the tree structure of the root node. As shown in Figure 1.

The root node of the tree structure indicates the user node; the middle layer represents the user interest categories nodes; the node at the bottom represents feature word node on each interest category. Counting the number of all weibo in interest category in this time period, the total number is recorded as N. In order to better represent user interest's changes, this paper adopts two user interest tree models which, respectively, represent short-term interest model and long-term interest model. Finally, this paper puts the user's short-term interest model and long-term interest model together to analyze the user's individuation.

3. Interest Classification

3.1. Support Vector Machines. From the discussion of the linear discriminant function, it can be drawn in this paper that the text is set linearly separable so that the text can always find the correct solution about division of the sample. In general, this paper can get infinitely many solutions, but it does not determine which is the optimal solution. The threshold for support vector machine is to separate the two types of vector texts set for the maximum interval.

As shown in Figure 2, a collection of two types of linearly separable texts is separated by a linear hyper plane. Clearly, in Figure 2(b), the separated interval of two types of text

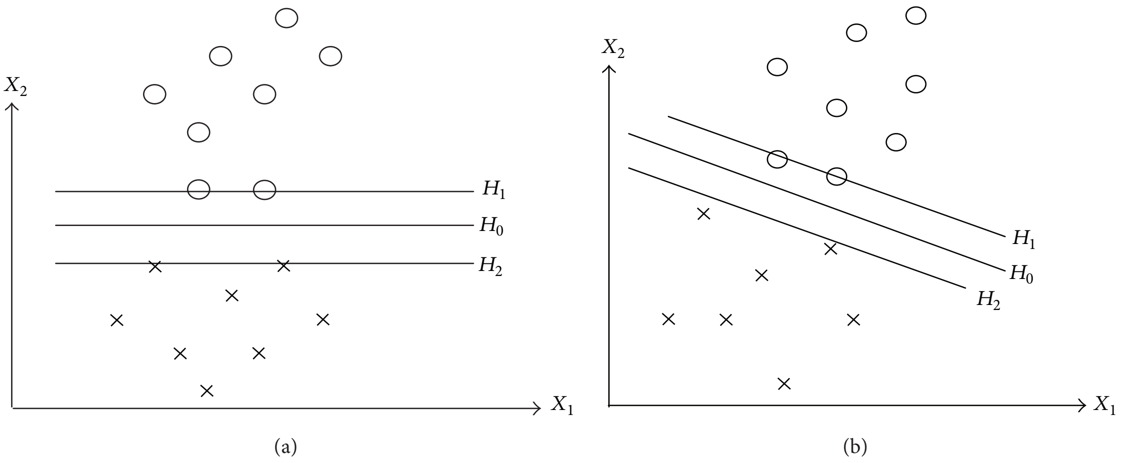

Figure 2: The two sets of linear category.

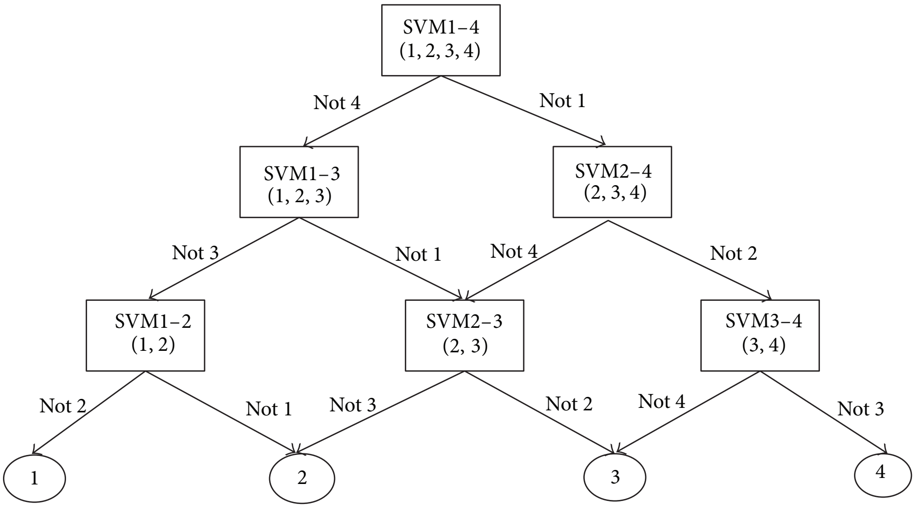

Figure 3: Directed acyclic graph SVM method.

sample should be larger than that in Figure 2(a), and if the H_0 in Figure 2(b) is a solution plane of maximum interval, this solution plane is optimal. And in Figure 2(b), those text vectors, which are at the nearest distance of the optimal interface that is located on the boundary, are called support vectors.

3.2. A Classification Algorithm of Improved Support Vector Machine. In this paper, the user interest category problems come down to text multiclass classification problems. One-to-many method first needs to train k second-class classifiers of support vector machine (SVM) [4–6]. One-to-one approach in all categories of training data supports second-class classifier of support vector machine. Directed acyclic graph approach in the training phase and one-to-one support vector machine approach are the same, in the testing phase, using a directed acyclic graph classification method to reduce test's time consuming. The directed acyclic graph has internal nodes and leaf nodes. Each internal node is the second-class classifier and the leaf node is the final classification category. For the test samples, it is based on the output result of the classifier to start from the root to determine whether their left subtree or right subtree will go until it reaches the leaf node. Specific algorithm process is shown in Figure 3.

Multiclass classification algorithm of support vector machine based on binary tree consists of training process and classification process, and the following description is based on training process and classification process of multiclass classification algorithm of support vector machine based on binary tree.

Algorithm 2. The algorithm is based on training process of multiclass classification algorithm of support vector machine based on binary tree.

Step 1. Calculate the category total of the training data set N.

Step 2. Construct a binary tree node.

Step 3. If $N > 2$, then go to Step 4 and if $N \le 2$, then go to Step 7.

Step 4. The first $N/2$ category data of the training data set is divided into the subset CateA, and the remainder data of the training data set is divided into subset CateB.

Step 5. Each training set of the training data subset CateA is marked as −1, and each training set of the training data subset CateB is marked as +1; then apply training data subsets of two types of CateA and CateB to construct a second-class classifier of support vector machines.

Step 6. Training data subset of CateA and training data subset of CateB repeat Steps 1–3, coming into the next training process.

Step 7. If $N = 2$, then go to Step 8; if $N = 1$, then go to Step 9.

Step 8. A category in the second-class training data is marked as 1 and the other category is marked as + 1; train a second-class classifier of support vector machine (SVM).

Step 9. The category label of the training data is set as a leaf node of binary tree structure; this category label is the classification label when test data goes into the leaf node.

Step 10. When the training is completed, the second-class classifier of support vector machine is regarded as an intermediate node of binary tree structure, Step 9 being a leaf node of binary tree structure.

First, we define that class distance $d_{i,j}$ is how much the distance of sample category i and sample category j is between the mean vectors minus their respective class average radius:

$$\begin{aligned} d_{i,j} &= \left\| m_i - m_j \right\|^2 - r_i - r_j \\ r_i &= \frac{1}{n_i} \sum_{k=1}^{n_i} \left\| x_k^j - m_i \right\|^2 \\ m_i &= \frac{1}{n_i} \sum_{k=1}^{n_i} x_k^j. \end{aligned} \tag{4}$$

Among them, m_i represents class mean vector of the sample category i; $\|m_i - m_j\|^2$ shows distance between class mean vectors of the sample Category i and Category j; r_i and r_j, respectively, show the class average radius of the sample category i and the sample category j; n_i shows the number of samples of the sample category i; $d_{i,j}$ represents the class mean distance between the sample category i and the sample category j.

Automatic text classification refers to the use of a computer program to determine the text category process according to the content of the text under a given system of classification. Automatic text classification is intended to estimate relationship of dependency between the input and output of the system based on the known training data set, so that the unknown output can make accurate predictions as possible [7–10].

4. Interest Model

User's interest in real life often tends to slowly change as time goes on. The user will gradually forget a once interest category and at the same time slowly find interest in a new category. In this paper, the change process of the user's interest is referred to as "interest drift." Interest drift phenomenon makes the user's interest model change accordingly with the passage of time. Therefore, the drift strategy of user's interest model should be considered in the study of user's interest model. In the study of the user models, there are two methods of interest drift that researchers frequently used: the first is the use of a sliding time window model to represent user interest model. This way lays too much emphasis on real-time user interests, ignoring the persistent performance. The second is the use of forgetting function to attenuate samples. This way lays too much emphasis on forgotten strategy, ignoring to discover new interests for users [11–15].

This paper directs the defects of interest drift strategy of the existing user interest model in establishing and updating stage, putting forward improvement strategies of interest model. First, we create a model of user interest vector and propose the user model attenuation algorithm and then analyze defects of interest drift strategy of the current user. Finally, the improved user interest model is proposed in view of these drawbacks.

4.1. User Interest Model Attenuation Algorithm. Human memory follows the laws of forgotten nature, and it means that human memory gradually weakens with the passage of time. This paper assumes that attenuation of user interest also follows the same laws of forgotten nature like memory, and it means that user interest gradually weakens with the passage of time, and attenuation is fast before it is slow [16]. Users frequently updated recently interest category representing the short-term interests of users, and as for interest categories that have not been updated for a long time, we can let its "senescence" achieve the objective of the filter.

Therefore, this paper introduces the concept of forgetting factor to forget the user interest model. When updating the user interest model, users not only add the newest interest categories to the user interest model, but also adjust the weight of existing interest categories in the interest models. It means to fix feature word weights of the interest category by forgetting factor and to gradually eliminate those "old" features words that are no longer in use.

Forgetting factor $F(x)$ is as follows:

$$F(x) = e^{\log_2(\mathrm{cur}-\mathrm{est})/\mathrm{hl}}. \tag{5}$$

Among them, cur represents the current time, est represents the time when feature word first appears in user interest model, and hl represents half-life of interest, which means that the feature word weight of interest model decays by half after hl days. In this paper, interest model attenuation algorithm is used only in long-term interest model, and the attenuation speed of interest is directly controlled [17].

As shown in Figure 4, if interest's half-life is set to 10 days, then after interest attenuation of ten days, the feature word

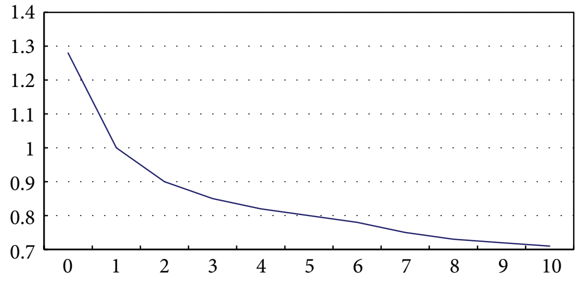

Figure 4: The attenuation curve of interest degrees.

weight of user interest model will be half forgotten, and it means that forgetting factor is 0.5.

4.2. User Interest Drift Strategy. In recent years, researches of the user interest drifting technology mostly adopt the strategy of time window attenuation algorithm. This strategy is controlled by time, and any interest attenuation is equal. Grabtree and Soltysiak adopt the time window method [17] according to user interest drift problems, and user model is set up only according to user's recent activity behaviors.

In order to better tap user interest from the user data, many researchers try to give a lower weight to the stale data, give the higher weight to the new data, and use all the user data to model, avoiding the disadvantages of time window method. Koychev and Schwab believe user interest attenuation is similar to natural forgotten regularity [18]. They propose user forgotten interest model. As is shown in Figure 5, forgotten model that is often based on old user model introduces the latest user data, giving higher weight to the latest user data, giving a lower weight to the old data in a user mode, and recombining to generate new user model. Maloof and Michalski's study adopts forgetting function strategy algorithm [19], giving an age to each sample data by forgetting function attenuation user model. Age will grow over time. When the age exceeds a certain threshold, the sample data is completely forgotten. Only sample data that is not forgotten is used to establish the user interest model.

Although the forgetting process model can handle the changing process of user interest in a longer period of time, it is difficult to respond to sudden changes of the short-term user interest. In order to solve this problem, this paper puts forward the hybrid model method which is combining the short-term interest model with long-term interest model. As is shown in Figure 6, the user interest model is composed of short-term interest model and long-term interest model. According to the characteristics of the two models, the different drift strategies are, respectively, adopted for short-term interest model and long-term interest model [15, 20–25].

Short-term interest model drift strategy: short-term interest corresponds to user's current interest, which is active and frequently changing. The short-term interest updating method requires that system can quickly respond, so it adopts the tactics of sliding time window. The window data is the user's current interest, and short-term interest model is also updated along with the user's updating weibo.

The formula is as follows:

$$U_{t_k}^{\text{cur}} = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{S_j}\sum_{i=1}^{S_j} w\left(t_k, p_i\right). \tag{6}$$

Among them, S_j represents the number of updating weibo for the users in the jth day and N represents the size of the sliding time window.

Long-term interest model drift strategy states the following: firstly, long-term interest reflects the interests of the user for a long time, it is relatively fixed. However, the user's interest in certain interest category will be forgotten and the degree of the interest category will gradually decrease over time. When a user's long-term interest drifts, this paper uses attenuation algorithm of user's interest model 4.3.2 to update the user's long-term interest model; secondly, the updated interest model is combined with the new interest model; finally, the combined interest model is the latest long-term interest model of the user. Long-term strategic interest drift is calculated as follows:

$$\begin{aligned} U_{t_k}^{t\text{-per}} &= U_{t_k}^{\text{per}} e^{-\log_2(d-d_{\text{new}})/\text{hl}^{\text{per}}} \\ U_{t_k}^{\text{per}} &= U_{t_k}^{t\text{-per}} + U_{t_k}^{t\text{-cur}}. \end{aligned} \tag{7}$$

Among them, $U_{t_k}^{t\text{-per}}$ represents user interest which is obtained by the original long-term interest through processing the user's forgetting function; $U_{t_k}^{t\text{-per}}$ indicates user interest which is obtained by short-term interest after processing of forgetting function; the user long-term interest is the sum of the two updated interests.

Among them, d represents the current update time of long-term interest model, d_{new} represents last updated time of the feature word t_k, hl^{cur} represents the half-life of short-term interest, and hl^{per} represents the half-life of long-term interest [16, 26–28].

5. The Result of the Experiment and Analysis

5.1. The Result of the Experiment. In experiment, this paper selects crawl data from sina weibo to establish personalized model for the user. This experiment selects Brooklyn to establish personalized model for users. Collecting the Brooklyn's updated 526 weibo in the recent period of time, 423 useful microblogs are extracted and analysised before mining. Then the text of the micro-blogs are handled, respectively. First, extracting the data of the first 15 days is used for initializing short-term interest model, and extracting the data of the first 30 days is used for initializing long-term interest model; then extracting data is used to update the short-term interest model every 15 days. The long interest model is updated in every 30 days and experiments are performed on each time point, respectively. Finally, the impact of the long-term interest model and the short-term interest model on user interest model at various time points calculates the proportion of short-term interest and long-term interest at various points of time and then gets optimized mixed interest model.

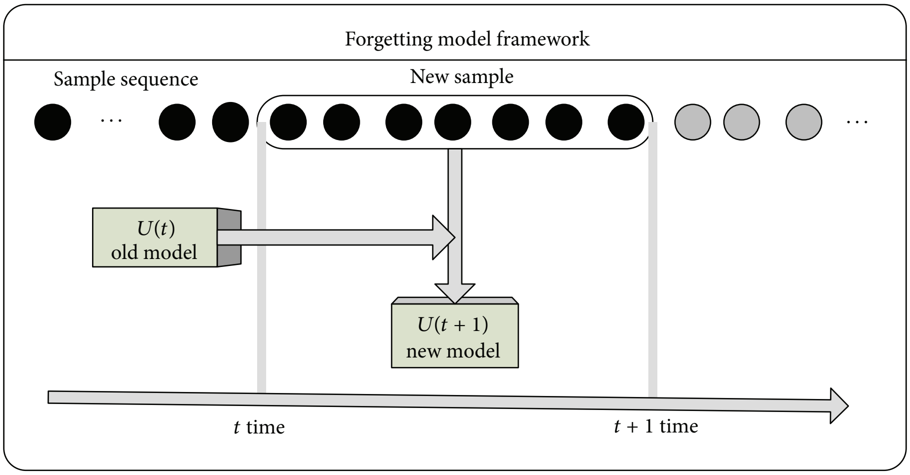

Figure 5: The framework of forgotten model.

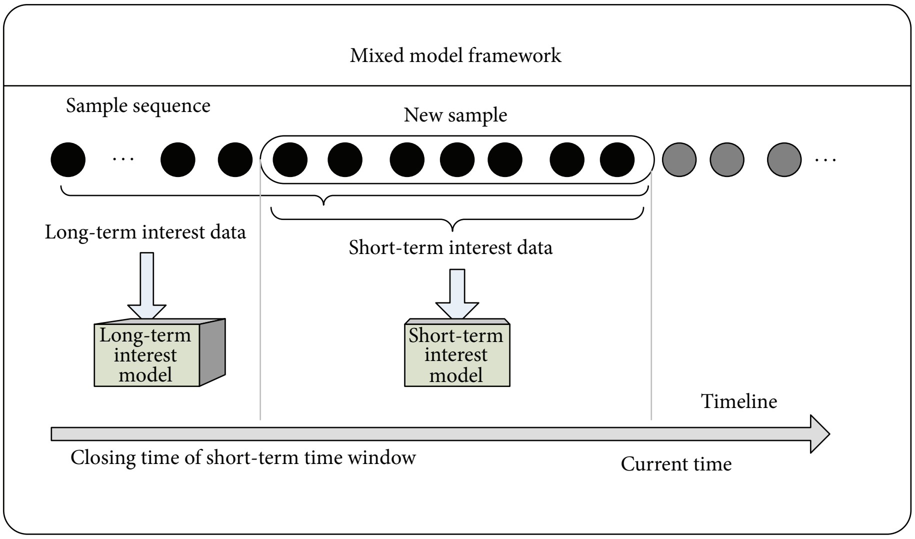

Figure 6: The framework of mixed interest model.

This paper takes a variety of combination tests to the half-life $\mathrm{hl}^{\mathrm{cur}}$ of short-term interest model and half-life $\mathrm{hl}^{\mathrm{per}}$ of long-term interest model. In this paper, it is concluded that short-term interest has a half-life of 10 days and long-term interest has a half-life of 25 days. They conform to the forgetting rule of short-term interest and long-term interest to some extent. Here, t is the changing process of the user interest model.

Short-term interest weight of each interest category in each period of time is shown in Table 1.

5.2. The Result of Analysis. In the experiment, the parameters of the selected model are as follows: $a = 0.6$, $b = 0.4$, $\mathrm{hl}^{\mathrm{per}} = 25$, $\mathrm{hl}^{\mathrm{cur}} = 10$. This paper uses crawl sina weibo data to test the effectiveness of the proposed algorithm. Here user Brooklyn is selected to be the object of study. Then four interest models are used to establish to research the relevance of search results, respectively. These models are sliding time window interest model, forgetting strategy interest model, fixed proportion interest model ($a = 0.6$, $b = 0.4$) and optimization mixed interest model. The keyword search is performed 100 times at each time point. The first 15 results are detemined by the system whether these are the micro-blogs which the user is interested in. Last the proportion of interested weibo is calculated. Test results are shown in Figure 7.

It can be seen from the experimental results that the sliding time window model, namely, short-term interest model, only refers to user's updated weibo of the recent 15 days, so the model can accurately grasp the short-term interests of users. When users are more concerned about the short-term interest, sliding time window model performance is slightly better than forgetting policy model. Because sliding time window model takes into account long-term interests of the users, the overall performance of the model is the worst.

The results can be seen from Figure 5 and the performance of the optimization hybrid interest model is better than sliding time window model, forgetting policy model,

Table 1: The short-term interest model of Brooklyn.

Timestamp	Traffic	Sports	Military	Medicine	Politics	Education	Environment	Economic	Art	IT
January 10	0	80.6	54.1	3.2	0	0	0	70.8	53.4	0
January 25	0	50.1	10.6	0	69.3	0	0	36.5	31.9	90.6
February 10	0	60.5	2.8	0	59.9	45.6	24.3	19.4	6.3	15.6
February 23	4.9	75.3	0.3	4.9	66.2	8.9	39.8	4.2	1.2	55.9
March 10	0	63.8	0.1	1.6	47.5	18.6	6.7	0.5	80.6	57.3
March 26	0.3	59.7	0	0	8.9	39.5	26.8	0.1	15.2	70.7

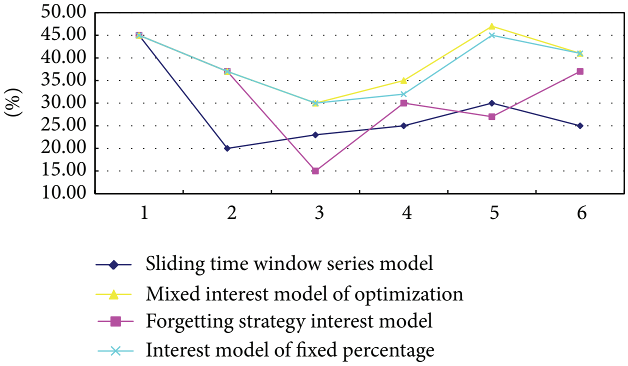

Figure 7: The comparison diagram of different interest drift strategy experimental results.

and fixed scale model. This model not only develops different strategies towards interest short-term and long-term interest of users, but also bases them on the change of user's interest and makes adjustments in time to the interest model. The adjusted model can reflect a more realistic user interest.

6. Conclusion

This paper puts forward an improved mixed interest model based on implicit feedback. The research and development of the current personalized modeling technology are reviewed. This paper introduces the common user modeling method and highlights the interest classification method based on support vector machine (SVM) and interest drift method combining sliding time window with forgotten strategy. Experiment result shows that the proposed method achieves personalized modeling of weibo users, which has good performance and scalability.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Project no. 71171068). The authors highly appreciate the above financial supports.

References

[1] L. U. Song, X. Li, and B. Shuo, "Wang improved real word document weight calculation method," *Chinese Information Technology*, vol. 14, no. 6, pp. 8–13, 2002.

[2] C.-F. Zhang, *Based on the behavior of users interested in the study*, Beijing University of Posts and Telecommunications, 2008.

[3] P. Ralph and J. Parsons, "A framework for automatic online personalization," in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS '06)*, pp. 137–146, Big Island, Hawaii, USA, January 2006.

[4] D. Mladenic and M. Grobelnik, "Feature selection for unbalanced class distribution and Naïve Bayes," in *Proceedings of the 6th International Conference on Machine Learning (ICML '99)*, pp. 258–267, 1999.

[5] Y. Yang and J. P. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the 14th International Conference on Machine Learning (ICML '97)*, pp. 412–420, Nashville, Tenn, USA, 1997.

[6] L. Gang, F. Z. Hua, W. Yong, and Z. Yong, "An application of neural networks in text classification," *Computer Engineering and Applications*, vol. 39, no. 36, pp. 73–75, 2003.

[7] T. Joachims, *Making Large Scale SVM Learning Practical*, MIT Press, Cambridge, Mass, USA, 2008.

[8] J. Platt, "Fast training of SVMs using sequential minimal optimization," in *Advances in Kernek Methods-Support Vector Learning*, B. Schllopf, C. J. C. Burges, and A. J. Smola, Eds., pp. 85–208, MIT Press, Cambridge, Mass, USA, 1998.

[9] D. D. Lewis, "Evaluating and optimizing autonomous text classification systems," in *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 246–254, Seattle, Wash, USA, July 1995.

[10] Z. Jingbo and T.-S. Yao, "FIFA text-based classification algorithm," *Microcomputer Applications*, vol. 23, no. 1, pp. 18–20, 2002.

[11] S. Yin, H. Luo, and S. Ding, "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 2402–2411, 2014.

[12] S. Yin, G. Wang, and H. Karimi, "Data-driven design of robust fault detection system for wind turbines," *Mechatronics*, vol. 24, no. 4, pp. 298–306, 2014.

[13] S. Yin, S. X. Ding, A. H. A. Sari, and H. Hao, "Data-driven monitoring for stochastic systems and its application on batch process," *International Journal of Systems Science*, vol. 44, no. 7, pp. 1366–1376, 2013.

[14] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and

process monitoring methods on the benchmark Tennessee Eastman process," *Journal of Process Control*, vol. 22, no. 9, pp. 1567–1581, 2012.

[15] S. Yin, X. Yang, and H. R. Karimi, "Data-driven adaptive observer for fault diagnosis," *Mathematical Problems in Engineering*, vol. 2012, Article ID 832836, 21 pages, 2012.

[16] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 2, pp. 435–447, 2008.

[17] I. B. Crabtree and S. J. Soltysiak, "Identifying and tracking changing interests," *International Journal on Digital Libraries*, vol. 2, no. 1, pp. 38–53, 1998.

[18] I. Koychev and I. Schwab, "Adaptation to drifting users interests," in *Proceedings of the ECML 2000 Workshop: Machine Learning in New Information Age*, pp. 39–45, Barcelona, Spain, 2010.

[19] M. A. Maloof and R. S. Michalski, "Selecting examples for partial memory learning," *Machine Learning*, vol. 41, no. 1, pp. 27–52, 2000.

[20] B. Xiao, Q. Hu, and Y. Zhang, "Adaptive sliding mode fault tolerant attitude tracking control for flexible spacecraft under actuator saturation," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 6, pp. 1605–1612, 2012.

[21] B. Xiao, Q.-L. Hu, and G. Ma, "Adaptive sliding mode backstepping control for attitude tracking of flexible spacecraft under input saturation and singularity," *Proceedings of the Institution of Mechanical Engineers. Part G. Journal of Aerospace Engineering*, vol. 224, no. 2, pp. 199–214, 2010.

[22] X. Zhao, X. Liu, S. Yin, and H. Li, "Improved results on stability of continuous-time switched positive linear systems," *Automatica*, vol. 50, no. 2, pp. 614–621, 2014.

[23] S. Yin, G. Wang, and X. Yang, "Robust PLS approach for KPI-related prediction and diagnosis against outliers and missing data," *International Journal of Systems Science*, vol. 45, no. 7, pp. 1375–1382, 2014.

[24] S. Yin, G. Wang, and H. R. Karimi, "Data-driven design of robust fault detection system for wind turbines," *Mechatronics*, vol. 24, pp. 298–306, 2014.

[25] S. Yin, X. Li, H. Gao, and O. Kaynak, "Data-based techniques focused on modern industry: an overview," *IEEE Transactions on Industrial Electronics*, 2014.

[26] L. H. Ungar and D. P. Foster, "Clustering methods for collaborative filter," in *Proceedings of the Workshop on Recommender Systems*, Technical Report WS-98-08, pp. 84–88, Menlo Park, Calif, USA, 1998.

[27] M. Balabanović and Y. Shoham, "Content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[28] R. Girardi and L. Balby Marinho, "A domain model of Web recommender systems based on usage mining and collaborative filtering," *Requirements Engineering*, vol. 12, no. 1, pp. 23–40, 2007.