Implicit Rating – A Case Study*

Song Wang[1], Xiu Li[2], and Wenhuang Liu[3]

[1] Department of Automation, Tsinghua University, Beijing, China 10084
Wangsong99@mails.tsinghua.edu.cn
[2] lixiu@tsinghua.edu.cn
[3] Liuwh@sz.tsinghua.edu.cn

Abstract. In this paper, the stable personal browsing patterns shown in Internet surfing are utilized to determine the users' preference on specific content. To be more specific, they are used to calculate the so called implicit ratings. We performed an experiment on all possible combinations of the implicit indicators to pick out the most significant indicators— elements of user browsing patterns. A thorough analysis and comparison are carried out before four indicators are selected as the input of an Artificial Neural Network which is adopted to calculate the implicit ratings. The mechanism of the implicit rating calculation is integrated into an educational resource sharing system as a featured module and works well.

1 Introduction

Seeking for appropriate educational materials in a large Educational Course Sharing System is a vapid job. A technique called Collaborative filtering [1], [2], [3] is then proposed to serve as the underlying recommending mechanism to alleviate such vapidity. Collaborative filtering takes a matrix as its input. Columns of the matrix are users' ratings on a specific course while each row corresponds to a single user. Thus each user is represented by a rating vector with some elements left blank. The vectors can be easily clustered into groups using existing clustering algorithms. Users in the same group generally share common interests. Hence, blank indicators in the rating vectors can be estimated by cross referencing between vectors in the same group.

Note that the input matrix is very sparse -- we should not expect users rate large portion of the courses because explicit rating actions are time consuming and will interrupt normal study processes[9] [10]. Implicit rating is then introduced as compensation.

Several papers have discussed the relative influence of a set of statistical parameters of user behaviors on implicit rating calculation. We examined the problem from another aspect. The contribution of our work is as the following: 1、Our methodology is different, rather than using statistical methods, we did some experiment and reached our conclusion by analyzing the results: 2、We integrated the whole implicit rating scheme into our educational resource sharing system and have got excellent performance.

This paper is organized as following: Section 2 briefly describes current research on implicit rating; In section 3, we give a detailed description about our research; In

* This work was supported by NSFC 70202008.

L. Wang, K. Chen, and Y.S. Ong (Eds.): ICNC 2005, LNCS 3611, pp. 574 – 583, 2005.

section 4, we designed a process to calculate the implicit rating; In section 5, we explained the deployment of the implicit rating in the educational resources sharing system and finally in section 6 we reached our conclusion and pointed out the future work need to do.

2 Related Research

We are directly inspired by the work of Mark Claypool et al. [4] [5]. In their work a browser was developed to record user actions as well as explicit ratings to find the correlation between them [5] [7] [8] [11] [14]. The authors then used Kruskal-Wallis test [6] to examine the degree of independence of the medians among each explicit rating groups for each implicit interest indicator. It is claimed that the higher the independence is, the less valuable an indicator may be for the explicit rating estimation purpose.

But a deeper look into the logic underlying the above reasoning will lead us to a question: from a practical point of view, the probability of the selected indicators not to work in certain circumstances is quite considerable. In another word, the work of Claypool et al. is inspiring and informative, but not very practical. To my knowledge, Kruskal-Wallis test is usually used to check if random numbers in different groups are of identical distribution, while in our case, let us take the time spent on a page for example, if Kruskal-Wallis test rejects the null hypothesis, which means that the distribution of the five ratings values are different, time spent on a page is then recognized as an effective implicit interest indicator. By following such a procedure, a set of indicators are filtered out.

If we choose to use the remaining indicators to predict the implicit rating, we are choosing a way too hard to follow. First we have to keep a trace of everyone's behavior, extracting some statistical features and do a sophisticate comparison to tell which group it should be classified into. We also should notice that such a classification is no longer a personalized one and is not fit for an online prediction.

There is another discrepancy. While calculating the precision of the prediction of the chosen set of indicators, difference between prediction results and explicit rating such '1' and '2' are treated as acceptable. It is quite easy to understand because difference between 1 and 2 or 4 and 5 are not large enough to distinguish user's preference between 'like' and 'dislike'. Here, 1~5 are treated as pure digits, rather than a set of labels, i.e. they are floating-point in the region [0, 5] rather than the set $\{1, \ldots, 5\}$.

Now let us recall the prediction process in which different behavioral patterns are recognized and labeled. 1~5 here are not only digits but labels representing a specific pattern.

Apparently, treating implicit rating as a label of some user's behavioral pattern is inappropriate. Implicit rating should fall into a continuous region and we designed a implicit rating generating system supplying such an requirement.

3 Indicator Selection

Analysis of Claypool et al.'s work gives us some clues about parameter selection. We will employ a more convincing method to avoid the inherent empirical nature of statistical analysis – statistically different parameters may indicate different user preference, but that is not guaranteed in theory.

3.1 Data Preprocessing

Data with explicit rating are effective data. Only effective data (about 80% of the dataset) which are meaningful for our experiment are kept and the other are filtered out. Then we get a dataset containing 1823 items. The ranges of indicators of the

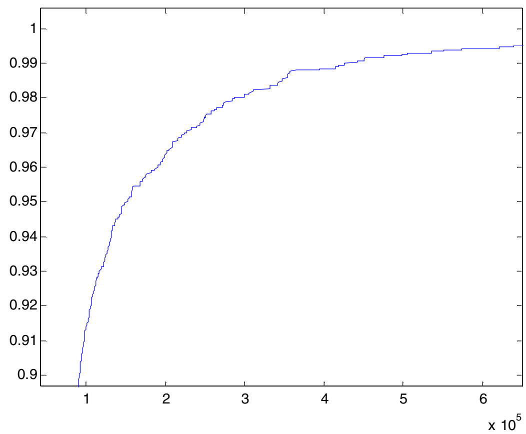

Fig. 1. ECDF of indicator 'time_spent_on_a_page', x-axis stands for the value of 'time_spent_on_a_page', y-axis stands for the ECDF varying with the x-axis value.

And then we plot the histogram:

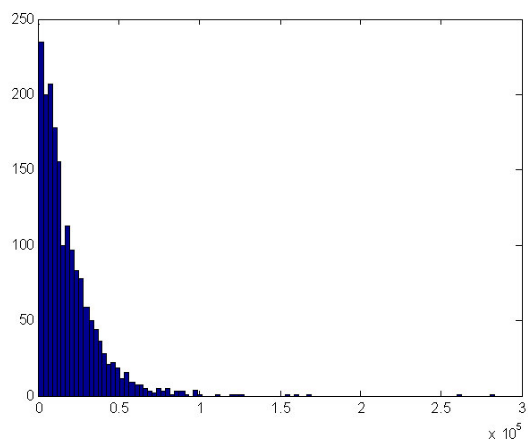

Fig. 2. Histogram of indicator 'time_spent_on_a_page', x- axis stands for the value of 'time_spent_on_a_page', y-axis stands for the histogram value of each bin according to x-axis value.

items are not determined—some may be too large to have a reasonable explanation. Take time_spent_on_a_page for example, its value may reach several hours in the case that a web user went to do something else and forget to close the browser. Such abnormal values are called outliers. We placed a control-boundary on each set of indicators to handle the outliers.

Here is what we do to find the control-boundaries. Again, let us take indicator 'time_spent_on_a_page' for example.

First, we plot the curve of ECDF (Empirical cumulative distribution function) of indicator 'time_spent_on_a_page'.

From the above two figures we can see that less than 2% of value of indicator 'time_spent_on_a_page' are greater than 3*10^5 milliseconds. So we can safely choose 3*10^5 milliseconds as the control-boundary of indicator 'time_spent_ on_a_page'. Following the same way, we find control-boundaries of the other twelve indicators and listed them as following:

Table 1. Control-boundaries of indicators

Indicator	Control-boundary
1. time_spent_on_a_page	300000 ms
2. time_spent_horizontal_scrolling	45 ms
3. time_spent_vertical_scrolling	50000 ms
4. number_of_scroll_events	10 times
5. time_spent_moving_the_mouse	40000 ms
6. number_of_the_mouse_clicks	20 times
7. '↑'times	10 times
8. '↓'times	30 times
9. '↑'time	1000 ms
10. '↓'time	5000 ms
11. 'page up' time	2000 ms
12. 'page up' times	4 times
13. 'page down' time	2500 ms
14. 'page down' times	5 times

We can also see that, from the histogram, values of indicator 'time_spent_ on_a_ page' scatters in a very wide range which is a desired property for good indicators. As a comparison, let us take a look at the histogram of indicator 'horizontal_scrolling_time':

Value of indicator 'horizontal_scrolling_time' squeezes in such a narrow region that we can not expect it give us any successful prediction of user's preference.

3.2 Experiment on Indicator Selection

We take matlab as the platform of our experiment. The experiments are carried out as following:

Step one, 'wash the dataset', the preprocessed data are free from unreasonable/odd outliers;

Step two, divide the clean dataset into 5 partitions evenly, four of them will be used as training set and the other is to be used as test set;

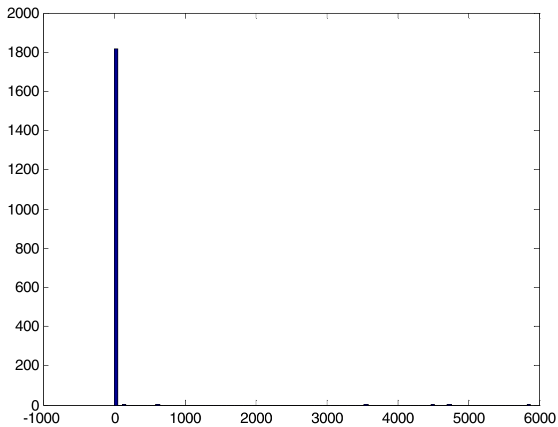

Fig. 3. Histogram of indicator 'horizontal_scrolling_time', x- axis stands for the value of 'horizontal_scrolling_time', y-axis stands for the histogram value of each bin according to x-axis value.

Step three, for every possible combination of the indicators, we train an artificial neural network for it and perform a corresponding test;

Step four, check the prediction performance of each individual indicator；

Step five, verified the obtained good indicators by finding the combination which produces the best prediction.

Now, we will focus on step three and four.

In step three, we are supposed to experiment on all possible combinations of the 14 indicators, that is 214=16384. For convenience, we will denote a specific combination by a binary number with 14 digits. To be clear, we have an example.

The corresponding binary number of 2301 is 00, 1000, 1111, 1101, the combination of indicators denoted by 2301 can then be decided, with indicators 3、7、8、9、10、11、12、14 selected and others unselected.

The following example is designed to show how prediction performance of each individual indicator is compared in step four.

We will define the complementary of two combinations regarding indicator I before we go on.

Suppose C1 and C2 are binary numbers denoting two different combinations (of the 14 indicators). C1 xor C2 (xor is and operator, we can simply treat it as 'not carry binary adder' here) will produce a third binary, if the corresponding decimal number of this third binary is 215-I, then C1 and C2 are called complementary regarding indicator I.

For example : C1=10,0001,1101,0010, C2=10,0101,1101,0010. C1 xor C2 = 00,0100,0000,0000, so C1 and C2 are complementary regarding indicator 4.

Intuitively, difference between predication performances of complementary combinations regarding indicator I are proportional to the importance of indicator I. So we gather all complementary combinations regarding each indicator and then can find their relative importance. Due to lack of space, we cannot give a detailed description of our experiment, the result is plotted afterwards:

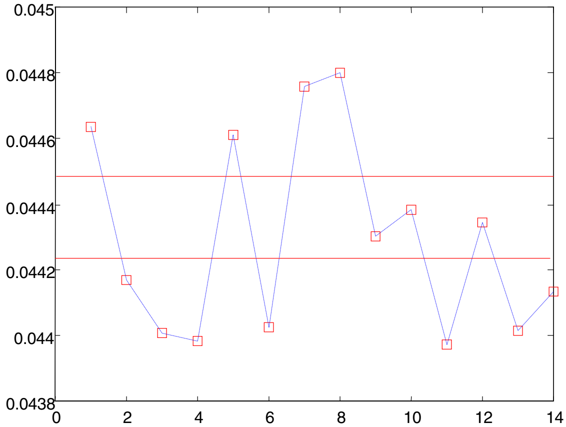

Fig. 4. Importance of indicators, x-axis stands for the indicator NO., y-axis stands for the average error of each indicator in predicting the implicit rating.

As we can see from figure 4, indicator 1, 5, 7, 8, corresponding to time_spent_on_a_page, time_spent_moving_the_mouse, '↑'times and '↓'times respectively, are the most important indicators, which complies with our experience. The 2 red bars partitioned the indicators into 3 parts, the first part, as we pointed out above, has the most satisfying predicting power, the second part has less power while the lowest part; part 3 can only be treated as noise in the predicting process.

4 Calculation of Implicit Ratings

We adopt Artificial Neural Network technique to calculate the implicit rating [8] [15]. From section 3 we know the input number is fixed as 4(the out put is the estimated

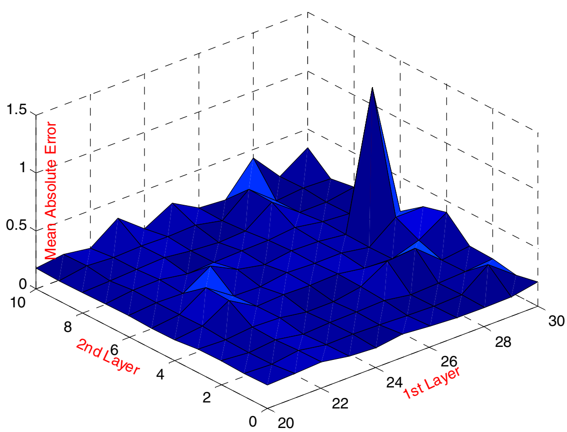

Fig. 5. Error of differently structured ANN

rating value), the work we need to do next is choosing right layer numbers and neuron numbers in each layer. Ref [6] has proved that an ANN can be used to fit any complicated curves in any degree, so we limit the ANN to have no more than 2 layers.

For different neuron numbers in each layer, we do separate experiments, different structure leads to quite different predicting performance.

When training the ANN model, we take mean absolute error as the performance function and use Levenberg-Marquardt back-propagation training algorithm.

To help understand the results better, we plot the result in figure 5.

Clearly, ANN with 30 neurons in the first layer and 7 neurons in the second layer has the least prediction error and that is the best structure we are seeking for.

5 Implementation

The project is supported by the Ministry of Education of P.R. China. The goal of the project is to establish a knowledge management platform for the 1500 nationally approved courses. The courses are open to all those who may need them. To increase the efficiency of the platform, a recommender system is introduced and has shown its power. Figure 6 show the framework of the recommender system.

Notice that implicit rating module stands at a crucial position. To facilitate the calculation of implicit rating, we will keep a set of weights for the ANN of each user, as well as a rating for each user/web page pair and a buffer to maintain the indicators from the latest visit. The whole process is described here:

Step1, the user clicks the ‘submit’ button to submit his/her indicator values or explicit rating;

Step2, push the indicator values into the buffer and set the flag if explicit rating is given;

Step3, train the ANN for the user/web page pair which has a newly come explicit rating if the CPU has spare time;

Step4, overwrite the old ANN with the learned one;

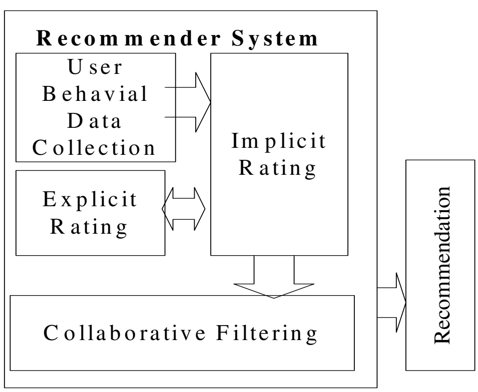

Fig. 6. Framework of the recommender system

Step5, if no explicit rating are given or system are busy to train the ANN, calculate the implicit rating using existed weights in the database and return it to the user as a feedback.

As we have seen, the implicit rating module has two working state: training state and evaluating state.

In the training state, the implicit rating module use indicator values and explicit rating to renew the weights of ANN. This enables the module to trace the browsing pattern and hence sustain/enhance the predicting performance.

In the evaluation state, the module only utilizes the trained ANN to simulate the browsing pattern and approximates the actual rating of the user.

We use a slide bar to collect the explicit rating so that the value we get is continuous rather than discrete numbers. Continuous number has its advantages when training the ANN and making more precise prediction.

Fig. 7. User behavioral data and explicit rating submission

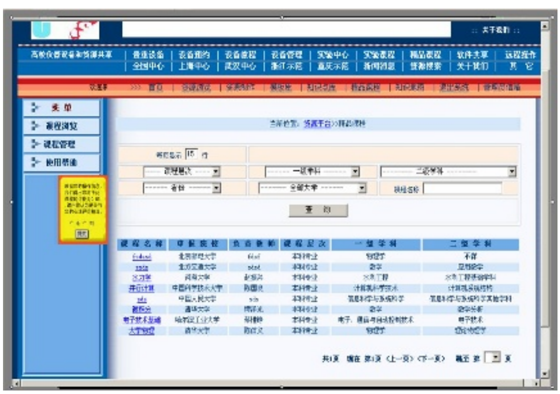

Fig. 8. Feedback of implicit rating

Meanwhile, we calculate the implicit rating and return it to the user. There are two reasons: first of all, to encourage the user to provide information useful to us later on and secondly, to check if our prediction reflects the actual user preference from the user's feedback.

6 Conclusion

The ANN we have introduced into the recommender system for the educational resource system removed the cost of explicit rating input. The sparsity problem facing the Collaborative Filtering has also be solved.

We are now working on the integration of the implicit rating and collaborative filtering. The implementation of collaborative filtering algorithm is also being carried out.

References

1. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl: Item-Based Collaborative Filtering Recommendation Algorithms. Proc. of the 10th International World Wide Web Conference (WWW10), Hong Kong, May 2001
2. John S. Breese, David Heckerman, Carl Kadie: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, July, 1998
3. Badrul M. Sarwar, George Karypis, Joseph A. Konstan, John T. Riedl: Application of Dimensionality Reduction in Recommender System -- A Case Study. ACM WebKDD 2000 Web Mining for E-Commerce Workshop 2000
4. Claypool, M., et al.: Implicit interest indicators, in Proceedings of ACM Intelligent User Interfaces Conference (IUI), Santa Fe, New Mexico. 2001, ACM
5. P.Le and M.Waseda: Curious browsers. Major Qualifying Project MQP-DCB-9906, May 2000
6. John A. Rice: Mathematical statistics and data analysis. Beijing: China Machine Press, 2003
7. Liberman, H: Letizia: An Agent that Assists Web Browsing. ZJCAZ-95, pp. 924-929
8. J.Goecks and J.W.Shavlik: Learning users' interests by unobtrusively observing their normal behavior. In proceedings of the ACM Intelligent User Interfaces Conference (IUI), Jan. 2000
9. White, R. W., Ruthven, I., and Jose, J. M.: Finding relevant documents using top ranking sentences: An evaluation of two alternative schemes. In Proceedings of the 24th Annual International Conference on Research and Development in Information Retrieval (SIGIR '02), Finland, 57-64.
10. Rafter, R., and Smyth, B.: Passive Profiling from Server Logs in an Online Recruitment Environment. In Proceedings of the IJCAI Workshop on Intelligent Techniques for Web Personalization (ITWP 2001), USA, 35-41
11. Maglio, P. P., Barrett, R., Campbell, C. S., and Selker, T.: SUITOR: An attentive information system. In Proceedings of the 5th International Conference on Intelligent User Interfaces (IUI '00), USA, 169-176.
12. Kim. J., Oard, D.W., and Romanik, K.: Using implicit feedback for user modeling in internet and intranet searching. University of Maryland CLIS Technical Report 00-01 .

13. Fischer, G., and Stevens, C.: Information access in complex, poorly structured information spaces. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI `91), USA, 63-70.
14. Billsus, D., and Pazzani, M. J.: A personal news agent that talks, learns and explains. In Proceedings of the 3' International Conference on Autonomous Agents (AGENTS 99), USA, 268-275
15. Liqun Han: ANN, theory, design and application. Chemical industry press