Combining Browsing Behaviors and Page Contents for Finding User Interests

Fang Li, Yihong Li, Yanchen Wu, Kai Zhou, Feng Li, Xingguang Wang, and Benjamin Liu*

Abstract This paper proposes a system for finding a user's interests based on his browsing behaviors and the contents of his visited pages. An advanced client browser plug-in is implemented to track the user browsing behaviors and collect the information about the web pages that he has viewed. We develop a user-interest model in which user interests can be inferred by clustering and summarization the viewed page contents. The corresponding degree of his interest can be calculated based on his browsing behaviors and histories. The calculation for the interested degree is based on *Gaussian process regression* model which captures the relationship between a user's browsing behaviors and his interest to a web page. Experiments show that the system can find the user interests automatically and dynamically.

1 Introduction

The Internet has become an important part of our daily life. Everyone has their own purposes or interests on the Internet access. How to find their interests has become one of the recent research goals for personalized search, collaborative filtering and recommendation systems.

Research on user interests can be broadly divided into two sides: client side and server side. Client-side research focuses on learning user model based on browsing history or behaviors, such as the time spent on the page, the count of clicking or scrolling, as well as other user activities [2]. The research on server side focuses more on extracting common interests or community patterns using web or search logs. However, there is no reason to believe that every user may need whatever the

F. Li, Y. Li, Y. Wu, K. Zhou, F. Li, X. Wang, and B. Liu
Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
e-mail: fli@sjtu.edu.cn

* Intel Asia-Pacific
Shanghai, China

AQ: Please check the link for asterisk is correct.

B. Mahr, S. Huanye (eds.) *Autonomous Systems – Self-Organization, Management, and Control*,

other users would need. One hundred user could have one hundreds needs. Therefore, many recent researches [2, 7, 8] have moved to the client side to analyze user web navigation and interactive behaviors to explore the use of personalized applications by tailoring the information presented to individual users. Our research also focuses on the client side, combining browsing behaviors and content analysis to generate the user interests automatically. We all know that if a user does not like sports, he will probably not view sports pages; if a user spends more time on specific pages, he would show some interests in the content of the pages. Thus, the content of viewed web pages may reflect the interests of a user, that the browsing behaviors are valuable indicator to infer a user's interests.

The rest of this paper is organized as follows. Section 2 introduces a plug-in tool to collect user browsing information. Section 3 describes our model and method. Section 4 presents the experiments of our system. Finally we make some conclusions.

2 Information Collection

In order to collect information during a user's surfing on the Internet, we implemented a browser plug-in. It can be divided into two parts: the page data collector (PDC) and the user behavior collector (UBC). The PDC is used to collect the information about visited pages and their contents. The UBC is to track the browsing activities and collect user browsing behaviors.

2.1 Page Data Collection

The information of a web page can be divided into two categories: page information and content information, given as follows:

Page Information

- The uniform resource locator (URLs) of the visited page: It is useful for extracting contents and evaluating its relevance to the next visited page.
- The page capacity: It includes how large the literal content of a page, how many pictures or images, how many out-links of the page. These factors have strong relation with the time that a user spent on the page.

Content Information

- The title of a page: It describes the main idea of a page.
- The content of a page: It describes the essence of a page which can reflect a user's interests.

2.2 Web Page Denoising

In order to obtain the content of a page, the module of Web Page Denoising (WPD) is implemented to eliminate irrelevant (noisy) information of the page, such as advertisements, copyrights, navigations and page scripts etc. WPD consists of two parts: page segmentation and noise filtering. The page segmentation part parses the structure of a web page and segments the page into several visual blocks. The noise filtering part calculates the importance of the above blocks based on their features and then filters out noising blocks.

The Step of WPD algorithm is briefly given as follows:

Step1: parse the structure of a given HTML page into a document object model (DOM) tree.
Step2: use the *vision-based page segmentation algorithm* (VIPS) [3] to segment a page into a certain number of visual blocks.
Step3: represent each of the visual blocks as a feature vector based on our block importance model. The spatial features (position and size) and other features (the number of images and out-links) are extracted and used to construct a feature vector.
Step4: train the block importance model based on Support Vector Machine with RBF kernel (the Radius Basis kernel Function) (RBF-SVM) [1].
Step5: extract the content of the most important visual blocks as the content of a page. Noisy information of unimportant visual blocks is filtered out thereafter.

2.3 User Behaviors Collection

The browsing behaviors are valuable indicator to infer his interests. After installed the browser plug-in on the client side, our UBC can track the user's activities with the web browser and instantly log the following kinds of user browsing behaviors:

- The time that a user spends on browsing a page
- The amount of scrolling
- The URLs sequence of a user clicks during a visit

Some browsing behaviors may not directly show whether the user likes or dislikes a page. For example, a user may have to slide the page several times to go through it. A long page possibly costs a user much time. Combination of browsing behaviors and page capacity can produce user interests.

3 Finding User Interests

3.1 Problem Definition

Let $\mathbf{P} = \{P_1, P_2, ..., P_n\}$ denotes a set of pages that a user has browsed. Each page can be represented as an *information vector*.

Definition 1 (Information Vector). An *information vector* of a page is defined as a 7-tuple $info_i = \left[size_i, link_i, image_i, scroll_i, \ time_i, dscroll_i, dtime_i\right]$, in which $size_i$ is the literal capacity of the page, $link_i$ is the number of out-links, $image_i$ is the number of images, $scroll_i$ is the amount of user scrolling and $time_i$ denotes the time spent on the page.

Definition 2 (User-Interest Vector). A *user's interest vector* is defined as an m-tuple $\mathbf{I_G} = \left[(G_1, \mathrm{I}_{G_1}), \ldots, (G_m, \mathrm{I}_{G_m})\right]$, where m denotes the number of the user's interests, G_i denotes the i^{th} interest and I_{G_i} denotes the corresponding interested degree of G_i. Each of his interests is represented as a set of keywords.

The first problem we need to solve is to develop a reasonable user-interest model which can infer a user's interested degree to a viewed web page based on his browsing behaviors. The second problem is to find what an interest is, and how to calculate its interested degree. As follows, Section 3.2 introduce our method to solve the problem 1, Section 3.3 tackles the problem 2.

3.2 User-Interest Model

AQ: Please confirm if renumbering of equations is correct.

According to the definition of an *information vector*, 5 of the 7 features can be obtained directly from the plug-in, while the last two features $dscroll_i$and $dtime_i$ need to explain as follows:

We observe that a web page is accessed by a hyperlink of the previous page. For example: a user may go to http://www.sony.com and follow an out link on the page: http://www.sonystyle.com/digitalmaging/cyber-shot.htm in order to find his interested information. The previous page is used to help him to find the latter page. In order **to accumulate the time spent and the scrolling of the previous page, two features**, $dscroll_i$ and $dtime_i$, are defined to evaluate the interest gain of a latter page. The two features are calculated using Eqs. 1 and 2.

$$dscroll_i = Sim_{url}(url_i, url_{i-1}) \cdot I_{P_{i-1}} \cdot [scroll_i - scroll_{i-1}] \tag{1}$$

$$dtime_i = Sim_{url}(url_i, url_{i-1}) \cdot I_{P_{i-1}} \cdot [time_i - time_{i-1}] \tag{2}$$

where Sim_{url} is the URL similarity between two consecutive pages which is calculated in Eq. 9.

$$Sim_{url}(url_i, url_j) = \frac{2 \cdot \mathrm{length}(\mathrm{common}(url_i, url_j))}{\mathrm{length}(url_i) + \mathrm{length}(url_j)} \tag{3}$$

where common(url_i, url_j) denotes common prefix substrings of two URLs and length(url_i) denotes the length of url_i.

Based on the *information vector*, the *Gaussian process regression* model (GPR) [6] is used to train our user-interest model M. This interest model captures how the *information vector* of a page is related to the interested degree of the page using the Eq. 4.

$$\mathrm{I}_{P_{new}} = \sum_{i=1}^{N} \alpha_i \cdot k(Info_i, Info_{new}) \tag{4}$$

Where N is the number of pages in a given training set, $\boldsymbol{\alpha} = (\mathbf{K} + \delta^2 \mathbf{E}_N)^{-1}$ and $\mathbf{K} = \left(k(Info_i, Info_j)\right)_{N \times N}$. The radius basis function (RBF) is considered as the kernel function used in Eq. 4, and calculated as follows:

$$k(Info_i, Info_j) = \exp\left\lfloor -\gamma (Info_i - Info_j)^2 \right\rfloor \tag{5}$$

Where the hyper-parameter γ denotes the length scale, of which the value is 1.0. Given a new page P_{new} and its *information vector Info*$_{new}$, the RBF-GPR model can predict a user's interested degree of the page. For simplicity, the degree of a user's interest is defined as the sum of the corresponding interested degrees of all those pages related to this interest.

$$\mathrm{I}_{G_i} = \sum_{P_j \in G_i} \mathrm{I}_{P_j} \tag{6}$$

Where G_i denotes a user's interest, $\mathrm{P}_j \in G_i$ and P_j is a single web page.

3.3 Finding User Interests

Page contents are important for finding user interests. Our clustering algorithm first utilizes Kaufman approach (KA) [5] for initialization, which initializes clustering by successively selecting representative page instances until *m* instances have been found. Then it uses the selected *m* seeds as the initial centroids and finally performs the spherical K-Means algorithm [4] to divide all the pages into*m* clusters. Based on the clustering results, some keywords are extracted to represent user interests and the *summarization method* is implemented to provide some detailed information for each interest.

4 Experiments

To evaluate the performance of our system, we conducted two experiments. One experiment was conducted to evaluate the effectiveness of the user-interest model based on RBF-GPR. The other one was conducted to validate whether users were satisfied with the results found by the system. There were 13 voluntary students jointed in our experiments. Each participant was given three tasks according to the following requirements:

1. Use Internet Explorer (6.0 or 7.0) browser embedded with our plug-in to surf the web.
2. Assign an interest score of [0, 100] to a visited page.
3. Predefine his interests by using some words and phrases.

From the first task, we collected a dataset of about 2-week information (from September 29, 2007 to October 10, 2007) gathered from 13 voluntaries, the dataset consists of the page data and the browsing behaviors. The total number of the visited web page was about 3,350 which covers different topics including politics, culture, economy, science, entertainment and so on. From the second task, we obtained the interested degree for each visited pages rated by each participant. From the third task, we have collected the predefined interests of 13 voluntaries manually. These interests are considered as the reference results. The average number of predefined interests was 10.5, minimum number was 7 and maximum was 13.

4.1 Evaluation of User-Interest Model

The dataset was divided into two groups: 65% of our dataset were used for training the model, and the rest was used for testing. There are two experiments:

Evaluate the RBF-GPR model for measuring prediction performance

For comparison purposes, we use the *mean square error* (MSE) to validate the effectiveness of our proposed user-interest model.

$$MSE(U) = \frac{1}{|D_U|} \cdot \sum_{x \in D_U} \left(\hat{f}(info(x)) - f(info(x))\right)^2 \tag{7}$$

Where D_U is a set of the pages visited by the user U, x is a page instance in D_U and *info*(x) denotes its information vector, $f(\mathit{info}(x))$ denotes the user-predefined interested degree of x and $\hat{f}(\mathit{info}(x))$ denotes the model-inferred interested degree of x. Table 1 shows the results.

Table 1 The evaluation results of the RBF-GPR model for measuring prediction performance

User	Pages	Mean Square Error
1	672	0.046
2	213	0.046
3	122	0.045
4	122	9.945
5	144	0.081
6	167	0.018
Ave.	240	0.0485

Table 2 The results of evaluation of the RBF-GPR model based on the distribution of a user's interests

User	System-Found Interests Number	MAE	SRCC
1	9	0.0101	1
2	9	0.0086	1
3	10	0.0091	0.952
4	9	0.0189	0.917
5	9	0.0075	0.833
6	9	0.0110	0.917
Ave.	9.17	0.0109	0.936

Evaluate the RBF-GPR model based on its influence on the distribution of a user's interests

We use the *mean absolute error* (MAE) to measure the influence of our model on the distribution of the user's interests.

$$MAE(U) = \frac{1}{|\mathbf{G}_U|} \cdot \sum_{G \in \mathbf{G}_U} \left(\hat{d}(G) - d(G)\right)^2 \tag{8}$$

Where $\mathbf{G}_U$ is a set of a user's (U) interests found by our system, G is one of the user's interests, $d(G)$ denotes the real interested degree of G obtained from the user-predefined interested degrees of the pages and $\hat{d}(G)$ denotes the system-evaluated interested degree of G. The Spearman correlation coefficient (SRCC) is used to calculate the relevant strength of the system rating and human rating.

$$SRCC(rank', rank) = 1 - \frac{6 \times SRDS(rank', rank)}{u \cdot (u^2 - 1)} \tag{9}$$

where $rank'$ is the system-evaluated ranking of the found interests, $rank$ is the user-given ranking of the found interests, $SRDS(rank', rank)$ denotes the difference between $rank'$ and $rank$ and $u = |rank|$.

Table 2 indicates two kinds of rankings are highly similar, the influences caused by our model based on the differences between the system-generated distribution and the real distribution of a user's interests can be ignored.

4.2 Validation of User Interests

The experiment was conducted to validate whether the users are satisfied with the results. Each user was asked to check each interest generated by our system and assign a score (0 to 5) for the interest. The fair-score of the system-found interests of the 13 participants is on the average 3.041 and the average best-score is 4.154, which proves that our system can find user's interests well, and the users are generally satisfied with the results. However, the average worst-score of the found interests is 1.385, which indicates that our system does not always yield good performance. The reason is that the clustering algorithm can not achieve 100% precision and noise information of the page also decreases the clustering result.

Conclusion

In this paper, we propose a system to investigate the problem of finding user interests. Our system utilizes the implemented plug-in to collect and track browsing behaviors of a user. The system combines the page contents and browsing behavior analysis to find and generate the user's interests automatically. Experiments show that our system can infer the interested degrees of visited pages based on user's browsing behaviors. A summary can compensate the incorrectness of keywords and indicate more detailed information about each interest. In the future, we will improve the quality of clustering algorithm by using more language technologies. We plan to use hierarchical clustering algorithm, to identify the hierarchical structure of user interests.

Acknowledgments We are grateful to Mr. Zhenggang Wang for implementation of the plug-in. We also thank Mr. Zuoer Lee for analyzing many clustering algorithms. Finally we want to express our sincere thanks to Prof. Bo Yuan for English language correction. This research is supported by the Intel China Lt. Co. and the UDS-SJTU joint research lab for language technologies (http://lt-lab.sjtu.edu.cn)

References

1. A Library for Support Vector Machines (LIBSVM). http://www.csie.ntu.edu.tw/∼cjlin/libsvm/
2. Atterer R, Wnuk M, and Schmidt A (2006) Knowing the User's Every Move: User Activity Tracking for Website Usability Evaluation and Implicit Interaction. In Proceeding of the 15th International Conference on World Wide Web (Edinburgh Scotland, May 23–26, 2006). WWW'06, ACM Press, New York, pp 203–212
3. Cai D, Yu SP, Wen JR and Ma WY (2003) VIPS: a Vision-based Page Segmentation Algorithm. Microsoft Technical Report (MSR-TR-2003-79), November, 2003
4. S.M. Wild (2003) Seeding non-negative matrix factorizations with the spherical K-Means clustering. MS Thesis for the Department of Applied Mathematics, University of Colorado, April 2003
5. Lozano JA, Pena JM and Larranage, P (1999) An empirical comparison of four initialization methods for the k-means algorithm. Pattern Recognition Letters, 20: 1027–1040, 1999
6. Rasmussen CE and Williams CKI (2006) Gaussian Processes for Machine Learning, MIT Press, 2006
7. Weinreich H, Obendorf H, Herder E, and Mayer M (2006) Off the Beaten Tracks: Exploring Three Aspects of Web Navigation. In Proceeding of the 15th International Conference on World Wide Web (Edinburgh Scotland, May 23–26, 2006). WWW'06, ACM Press, New York, pp 133–142
8. White RW, and Drucker SM (2007). Investigating Behavioral Variability in Web Search. In Proceeding of the 16th International Conference on World Wide Web (Alberta Canada, May 8–12, 2007). WWW'07, ACM Press, New York, pp 21–30

AQ: Please provide publisher location details for Ref. 6.