User Preference Learning for Multimedia Personalization in Pervasive Computing Environment

Zhiwen Yu[1], Daqing Zhang[2], Xingshe Zhou[1], and Changde Li[1]

[1] School of Computer Science, Northwestern Polytechnical University, P.R. China
{zhiwenyu,zhouxs}@nwpu.edu.cn, changde_lee@126.com
[2] Context Aware Systems Department, Institute for Infocomm Research, Singapore
daqing@i2r.a-star.edu.sg

Abstract. Pervasive computing environment and users' demand for multimedia personalization precipitate a need for personalization tools to help people access desired multimedia content at anytime, anywhere, through any devices. User preference learning plays an important role in multimedia personalization. In this paper, we propose a learning approach to acquire and update user preference for multimedia personalization in pervasive computing environment. The approach is based on Master-Slave architecture, of which master device is a device with strong capabilities, such as PC, TV with STB (set-on-box) or PDR (Personal Digital Recorder), etc, and slave devices are pervasive terminals with limited resources. The preference learning and update is done in the master device by utilizing overall user feedback information collected from different devices as opposed to other traditional learning methods that just use partial feedback information in one device. The slave devices are responsible for observing user behavior and uploading feedback information to the master device. The master device is designed to support multiple learning methods: explicit input/modification and implicit learning. The implicit user preference learning algorithm, which applies relevance feedback and Naïve Bayes classifier approach, is described in detail.

1 Introduction

The rapid advances of communication technologies precipitate more and more embedded computing devices, such as PDAs, cell phones, etc., to be used in a wireless environment. This has led to a shift from traditional computer-centered to human-centered information access mode, which is known as *Pervasive Computing*. Multimedia information is widely used in pervasive computing environment in many application fields, such as multimedia digital libraries, home entertainment, live camera remote surveillance, etc. A major trend and requirement in today's multimedia information service is personalization. The capability to model and learn user interests is at the heart of a personalized information filtering system [1]. It is also true for multimedia personalization. Since the interest of a user is changing as time goes by, the quality of personalization mainly depends on whether the user profile really reflects the user preference. So we can conclude that user preference learning plays an important role in multimedia personalization.

R. Khosla et al. (Eds.): KES 2005, LNAI 3682, pp. 236–242, 2005.

Acquiring and updating user preference for multimedia personalization in pervasive computing is challenging due to the following factors:

- The poor human-machine interactivity of pervasive devices causes user explicitly inputting his preference to be nearly impossible.
- The limited computing power and storage make complex learning algorithm impractical.
- The mobility of user and devices causes insufficient time to acquire and update user preference. In pervasive computing environment, multiple devices are attached to a user and used at anytime anywhere. Each of the devices can gather user feedback information, but may be very fractional.

Fortunately, many of today's users own a device with strong capabilities, such as PC (Personal Computer), TV with STB (set-on-box) or PDR (Personal Digital Recorder), etc. In this paper, we address the above challenges by proposing a centralized learning approach with the "strong capability device" been used in conjunction with the mobile user devices for user preference acquisition and update. Our approach collects fragments of user feedback information in different pervasive devices so as to build an abundant feedback repository, and then use a strong capability device to implicitly learn user preference from the feedback repository.

Our early work, TV3P [2] employs an implicit and explicit profiling scheme for personalized TV experiences by integrating explicit input/modification, explicit feedback, and implicit feedback. However, it mainly serves for TV or desktop environment, but not for pervasive computing environment.

The approach proposed here has several advantages. First, it learns user preference by utilizing overall feedback information as opposed to other traditional methods that just use partial feedback information in one device. Second, our approach can relieve pervasive devices with limited resources from computation-consuming and storage-consuming learning tasks. Third, it can make full use of existed user profile residing in other devices. For instance, a user has watched television program through TV set for a long period, so the user profile in the TV set may be comprehensive and correct. When the user wants to watch program through his PDA, he can export his/her preference information from the TV set, and no further learning is required.

2 Overall Learning Approach

The preference leaning is done in a strong capability device, while the pervasive devices are relieved from computation-consuming and storage-consuming learning tasks. The only thing the pervasive devices need to do is to observe user behavior and upload feedback information to the strong capability device. We call the device with strong capability as *master device*, and corresponding pervasive devices as *slave device*.

Fig. 1 shows the schematic architecture of the learning strategy. There are three primary elements: multimedia service provider, master device, and slave device. There are six interfaces between different elements in the whole learning process. They are depicted as follows:

1) When the user wants to access personalized multimedia information through his/her pervasive mobile devices, the slave device first connects with his/her master device asking for the latest user profile.
2) Then the master device sends user preference to the slave device.
3) The slave device sends a request to multimedia service provider, encapsulating user preferences, slave device capabilities, and network characteristics into the request information.
4) Now, the service provider selects and delivers multimedia information to the slave device in an appropriate modality, e.g. video, audio, image, text, or synthetic model, so as to meet the display capability of the device.
5) In the viewing process, the slave device observes user behavior for a specific multimedia content, and sends the feedback information to the master device. The feedback captured by the slave device can be sent at once when it is obtained or sent after a set of feedbacks gathered.
6) When the user profile is updated, the master device sends the latest user profile to the slave device. If the user will, the process goes to (3), otherwise keeps the updated user profile in the slave device.

The master device is designed to support multiple methods for preference learning: explicit input/modification and implicit learning. Explicit input/modification provides Graphic User Interface for the user to input interests when registration or modify preference after login. Implicit learning analyzes user viewing history or feedback information generally applying probability statistics, and then updates user preference. The feedback information is collected through any devices at anytime anywhere. The implicit learning algorithm is described in Section 3 in detail.

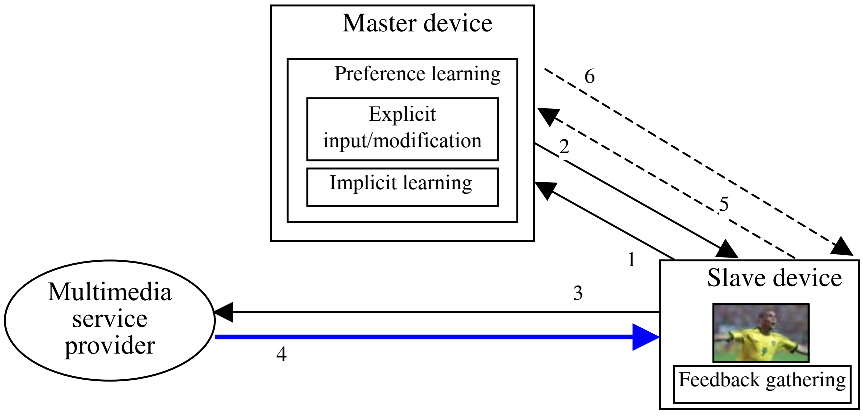

Fig. 1. System schematic architecture

In a pervasive computing environment, the working spot of a device may be at home, in office, on a train, or on road, etc. A diverse set of network technologies are used by different slave devices to communicate with master device. For instance, the master device may be a PC at home. The slave devices may include an interactive TV, PC in office, laptop, PDA, and cellular phone. To communicate with the master device, the interactive TV at home can use IEEE 802.11/Bluetooth, PC in office can use Ethernet, laptop can use ISDN, PDA and cellular phone can use GPRS.

3 Implicit User Preference Learning Algorithm

The general model of the implicit learning algorithm is based on relevance feedback [3], which is an effective and efficient information retrieval technique that can be used to adjust user preference approaching to user's real interests. The algorithm is defined as follows:

$$W_i' = \alpha \times W_i + \beta \times F_{pos} - \gamma \times F_{neg} \tag{1}$$

where W_i' is the updated weight of feature f_i; W_i is the initial weight of feature f_i. α, β and γ are the feedback parameters to be set. The well-known and effective feedback parameters is instantiated by Rocchio [3] as α=1, β=2, and γ=0.5. F_{pos} and F_{neg} represent the weights of positive feedback and negative feedback to feature f_i respectively.

The positive estimation (F_{pos}) and negative estimation (F_{neg}) were performed by using Naïve Bayes classifier [4] approach. All user actions for multimedia objects are divided into two classes, *Positive* or *Negative*, that is, positive feedback records class *PC*, and negative feedback records class *NC*. The records are about user's viewing attributes and corresponding viewed multimedia objects. The general form is:

```
< class | action time | action type | watching duration | total
duration | title | genre | actor | director | language | keyword >
```

The "class" determines which class (*PC* or *NC*) the action belongs to; "action time" specifies the time that the action takes place; "action type" specifies the type of action performed by the user, e.g., "Record", "View", etc.; "watching duration" specifies the duration of the action; "total duration" specifies the total duration of the multimedia content, such as a TV program; for static modality, such as image, or text, the "watching duration" and "total duration" can be set as the same default value (e.g. 5 seconds); "title", "genre", "actor", "director", "language", and "keyword" are descriptive information about the multimedia content, which can be extracted from the multimedia metadata.

F_{pos} and F_{neg} equals the conditional probabilities of feature f_i in class *PC* and *NC*:

$$F_{pos} = P(f_i \mid PC) \tag{2}$$

$$F_{neg} = P(f_i \mid NC) \tag{3}$$

Suppose $n(PC, f_i)$ means the times of feature *fi* occurring in *PC*, while $n(NC, f_i)$ means the times of feature *fi* occurring in *NC*. $n(PC)$ and $n(NC)$ denote the sum of times of all features occurring in *PC* and *NC* respectively.

For a recommendation item, we consider three action types: Record, View, Delete.

(1) If the action type is "Record", obviously the feedback is *positive*, $n(PC, f_i)$ and $n(PC)$ are calculated as follows:

$$n(PC, f_i)' = n(PC, f_i) + 1 \tag{4}$$

$$n(PC)' = n(PC) + 1 \tag{5}$$

$n(PC, f_i)'$: the updated $n(PC, f_i)$;

$n(PC)'$: the update $n(PC)$;

(2) If the action type is "View" and the recommended item is a video/audio content, whether the feedback is positive or negative depends on the user's real watching duration. We assume that if the real watching duration is longer than a threshold θ (such as 30 seconds), the user really likes the content (the user gives implicit positive feedback); otherwise, the user dislikes it (the user gives implicit negative feedback). $n(PC, f_i)$, $n(PC)$, $n(NC, f_i)$, and $n(NC)$ are calculated as follows:

$$n(PC, f_i)' = n(PC, f_i) + \frac{T_r - \theta}{T_t} \qquad T_r \geq \theta \tag{6}$$

$$n(PC)' = n(PC) + \frac{T_r - \theta}{T_t} \qquad T_r \geq \theta \tag{7}$$

$$n(NC, f_i)' = n(NC, f_i) + \frac{\theta - T_r}{\theta} \qquad T_r < \theta \tag{8}$$

$$n(NC)' = n(NC) + \frac{\theta - T_r}{\theta} \qquad T_r < \theta \tag{9}$$

T_r : user's real watching duration;

T_t : total duration of a specific multimedia object;

θ : the threshold of the time duration;

$n(NC, f_i)'$: the updated $n(NC, f_i)$;

$n(NC)'$: the updated $n(NC)$;

If the recommended item is an image or text, the action "View" is regarded as the same as "Record". $n(PC, f_i)$ and $n(PC)$ are calculated according to Equation (4) and (5).

(3) If the action type is "Delete", obviously the feedback is *negative*, $n(NC, f_i)$ and $n(NC)$ are calculated as follows:

$$n(NC, f_i)' = n(NC, f_i) + 1 \tag{10}$$

$$n(NC)' = n(NC) + 1 \tag{11}$$

Suppose $|V_1|$ and $|V_2|$ denote the total number of features occurring in *PC* and *NC* respectively. According to Lidstone's Law of succession [5], $P(f_i \mid PC)$ and $P(f_i \mid NC)$ can be estimated as follows:

$$P(f_i \mid PC) = \frac{n(PC, f_i) + \lambda}{n(PC) + \lambda |V_1|} \tag{12}$$

$$P(f_i \mid NC) = \frac{n(NC, f_i) + \lambda}{n(NC) + \lambda |V_2|} \tag{13}$$

λ is a positive number, normally between 0 and 1. The case of $\lambda = 0.5$ is called Jeffrey Perks Law or Expected Likelihood Estimation (ELE), which is widely adopted in various applications. If λ is set as 1, the Lidstone's Law is the same as Laplace's Law.

If the updated W_i' is larger than its upper bound, let W_i' =upper_bound (e.g. 1.0).

If the updated W_i' is less than its lower bound, let W_i' =lower_bound (e.g. –1.0);

If the absolute value of updated W_i' is not less than a preset threshold ξ (that is, $|W_i'| \geq \xi$), we will keep it in the user's profile, otherwise discard it, because it is too trivial.

4 Performance Evaluation

We evaluate system performance in terms of learning speed and learning efficacy. Learning speed is crucial for mobile devices, because long-time computation will consume large battery power. Learning efficacy, on the other hand, reflects how much the learned preference approaches to the user's real interests, which directly influences the quality of personalization. The data sets used for experimentation and performance analysis were taken from the Internet Movie Database (IMDb, http://us.imdb.com). The description information for each movie in IMDb is abundant, which includes title, director, genre, actor, keywords, language, country, etc. For each movie, 15 features were extracted from its description information in IMDb to represent its metadata in our system.

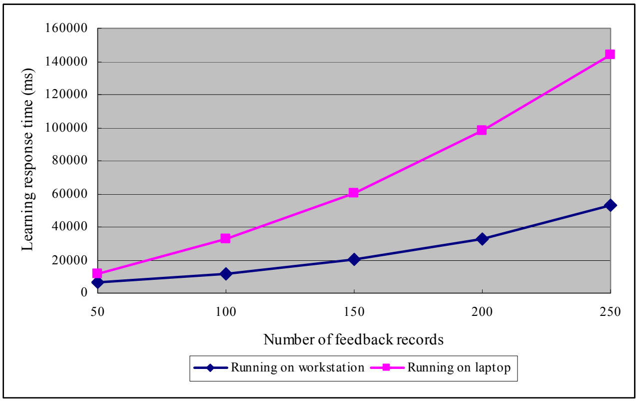

Fig. 2. Experimental result of learning speed

Experiment 1: We evaluate learning speed by measuring learning response time on the master device (workstation) and the slave device (laptop) respectively. We compared the time costs by varying the number of feedback records ranging from 50 to 250. The evaluation result is shown in Fig. 2, from which we can see that the user preference learning is computationally intensive and the learning time is largely dependent on the size of the feedback records. We also can observe that the learning response time on the laptop takes much longer than that on the workstation, and with the number of feedback records increasing from 50 to 250, the time difference is remarkable. When the size of feedback records reaches 250, the learning time on the laptop is very long (144s), which consumes much battery power and causes inconvenience to the user. But the learning time on the workstation is much less and acceptable. This result proved that the idea of running preference learning on a strong capability device other than pervasive mobile devices is appropriate.

Experiment 2: We use *precision* [6] to evaluate the efficacy of user profile learning. It denotes the ratio of the number of movies that user really interested and recorded to the total number of movies recorded. We took 50 multimedia contents as training set, and 200 contents as testing set. The testing set is divided into 5 sessions with each session having 40 contents. We first let the user browse all of the movies' title and plot beforehand, and classify them into two classes: *interested* and *not-interested*. Then, the system chooses and records movies for the user according to the user profile learned. After every session finished, we calculated the values of precision. The result is encouraging with the average precision value being 0.83. The experimental result proved that our system could keep track of user preference changing over time, and perform good filtering effectiveness, i.e. record most of the movies that the user likes.

5 Conclusion

In this paper, we propose an approach of user preference acquisition and update for multimedia personalization in pervasive computing environment. The major contributions of the paper are: (1) introducing a centralized learning approach to utilize overall feedback information and relieve pervasive devices from learning tasks; (2) providing the implicit user preference learning algorithm which applies relevance feedback and Naïve Bayes classifier approach.

Acknowledgments

This work was supported by the National Science Foundation of China, and the Doctorate Foundation of Northwestern Polytechnical University of China.

References

1. Dwi H. Widyantoro, Thomas R. Ioerger and John Yen: An Adaptive Algorithm for Learning Changes in User Interests. Proc. of the ACM Intl Conf. on Information and Knowledge Management, Kansas City, Missouri, USA (1999) 405-412
2. Zhiwen Yu and Xingshe Zhou: TV3P: An Adaptive Assistant for Personalized TV. IEEE Transactions on Consumer Electronics. Vol. 50, No. 1 (2004) 393-399
3. J.J.Rocchio: Relevance feedback in information retrieval. The Smart System--Experiments in Automatic Document Processing. Prentice Hall (1971)
4. Joachims T.: A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization. Proc. of the 14th Intl Conf. on Machine Learning (1997) 143-151
5. Lidstone, G.J: Note on the general case of the Bayes-Laplace formula for inductive or a posteriori probabilities. Transactions of the Faculty of Actuaries, 8:182-192.
6. C.J. van Rijsbergen: Information Retrieval. Butterworths, second edition (1979)