

# User Behavior Mining for On-Line GUI Adaptation

Wei Pan<sup>1,2</sup>, Yiqiang Chen<sup>1</sup>, and Junfa Liu<sup>1,2</sup>

<sup>1</sup> Institute of Computing Technology, Chinese Academy of Sciences  
{panwei,yqchen,liujunfa}@ict.ac.cn

<sup>2</sup> Graduate University of Chinese Academy of Sciences

**Abstract.** On-Line Graphics User Interface (GUI) Adaptation technology, which can predict and highlight user's next operation in menu based graphics interface, is the key problem in next generation pervasive human computer interaction, especially for remote control device like Wiimote assisting TV interaction. In this paper, a hierarchical Markov model is proposed for mining and predicting user's behavior from Wiimote control sequence. The modal can be on-line updated and highlight the next possible operation and then improve the system's usability. We setup our experiments on asking several volunteers to manipulate one real education web site and its embedded media player. The results shows our modal can make their interaction with GUI more convenient when using Wii for remote control.

## 1 Introduction

Pervasive Computing is a popular concept and attracting more and more computer and communication experts. [Xu Guang-You:2007] defines it as an attempt to break the pattern paradigm of the traditional relationship between users and computational services by extending the computational interface into the user's environment, namely the 3D physical space. Graphics User Interface(GUI) is a basic interaction interface in this environment. Lots device used for simplifying and improving the interaction style has been invented recently. Take Wii Remote([nitendo], or wiimote) as an instance, which was originally designed as a TV game controller, has a great potential of becoming a popular HCI device, especially for GUI([Lee, Kim:2008]). The advantages of wiimote assisting GUI should be attributed to its appealing characteristics —wireless connection, simple operation, and human engineering shape. Moreover, the acceleration sensors equipped inside provide more useful information about motions of the user, thus it is possible to develop more intelligent functionalities based on it compared to traditional mouses, like recognizing the user's gestures [Kela:2006] and so on.

Although wiimote greatly improves the user experience, there are several limitations when it is used as a remote mouse with a simple keyboard (just six keys). Let think about the process using the controller. We first choose the input focus and press "OK" to trigger the operation. For example, we can use the

device to run a media player, pause, or change the video, etc. This kind of interaction would be very useful and attractive when demonstrating in public or in class. However, the survey from volunteers indicates that some operations are not as convenient as expected. For instance, tedious operations of “tab” must be pressed until we reach the right menu we want.

In this paper, we propose a solution to improve the usability of the interaction modal. We address the problem of decreasing redundant operations by predicting the user’s next possible operation in advance. While everyone has some general action habits, we aim to mine them out from history data to assist wiimote operation. We employ a hierarchy Markov to model wiimote control sequence. The predicted result using the model serve as a recommendation of next operation to the user to augment the usability and reduce unnecessary operations. We also setup our experiments on an educational web site.

The rest of this paper is organized as follows. Section 2 gives out some related work. The system architecture, as well as our behavior mining model is introduced in Section 3. In section 4, we describe the experimental results and give the evaluation. We conclude the work and propose the future work in the last section.

## 2 Related Work

Our wiimote controlling system is a kind of intelligent user interface, which has been discussed for years and some demo systems are already in use. [Hook:2000] discussed systemically four problems to overcome on the road to intelligent user interface:

- usability principle for intelligent interface
- reliable and cost-efficient intelligent user interface development method
- a better understanding of how and when intelligence can substantially improve the interaction
- authoring tools to maintenance the intelligent part of the system

There are also several works proposed to handle these challenges. [Lieberman:1995] explored a suggestion tool in web browsing, where some links that may be potentially interesting to users were provided to realize a quicker search. [Insung:2007] introduced an intelligent agent in a health-care system used for self-governing and self-customized by making wise decisions. These are early tries on intelligence interface, but not suitable for new interaction model. [Kela:2006] introducing a new convenient input style, which may be very useful in the pervasive computing environment. In this paper, We sufficiently study the character of wiimote assisting GUI and propose a solution to improve its intelligence.

[Etzioni, Weld:1994] is a good example of modeling based on pre-programmed rules. Similar works can be found in [Lester:2000] and [Virvor, Kabassi:2002]. Rules are very useful, however its defect is also obvious—rules are hard to be update online. Especially, powerful rules are hard to be built for wiimote operation sequence. Considering habits may change, on-line updating is indispensable.

In this paper, we setup a hierarchical Markov model to simulate the user habits, and introduce several updating algorithm for model adaptation.

### 3 Proposed System Architecture

#### 3.1 System Architecture

Fig. 1 illustrates the architecture of our recommendation system, which is consisted of four main modules. The user interacts with the system through a wiimote, and then the system responds by making a recommendation about next operation, which is predicted by the behavior mining model. If the user finds the recommendation useful, he/she need to press “OK” to the next step, or choose the one needed. The system will gather the wrong recommendations automatically for further study. There is also a module called *Updating Strategy*, which can update our model based on the latest collected data. Fig. 2 is the operation interface of an educational web site for primer school study, and our experiments are carried out on it. The white device on the right is a picture of wiimote.

#### 3.2 Model User Behavior

*Model* module is the most important part of the system in Fig. 1. Since the user’s behaviors on the web site is a sequence of click actions, taking the web

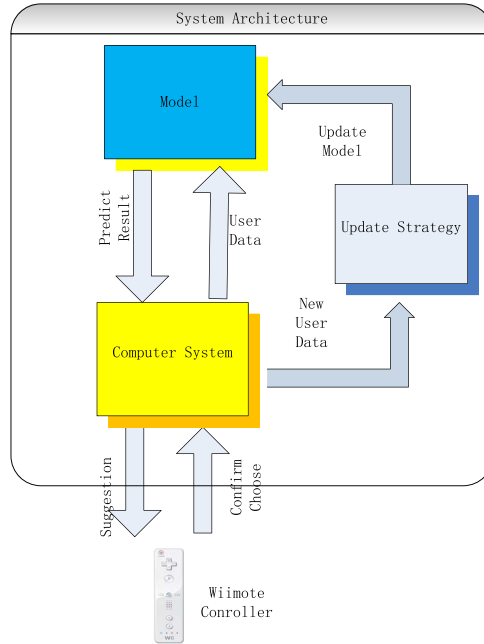


Fig. 1. Architecture of Recommendation System



Fig. 2. Web Site where and the device We do the Experiment

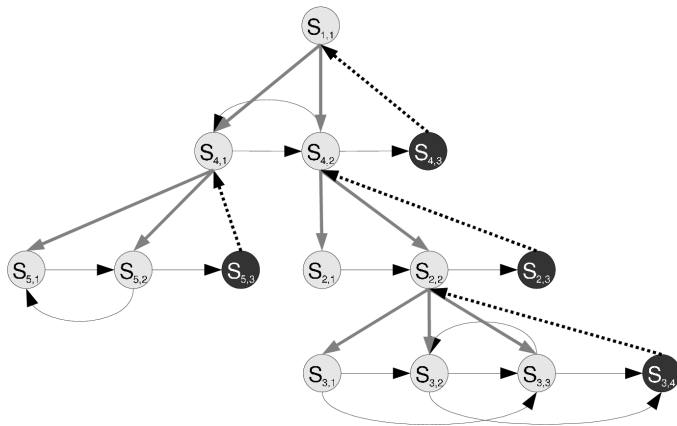


Fig. 3. Hierarchical Markov Chain

site hierarchy into account, we model it using a hierarchical Markov chain. [Meyn and Tweedie:2005] discussed Markov Model theory in detail, and [Fine:98] proposed hierarchical hidden markov model similar to the model we will discuss in this paper. Fig. 3 explain the structure of a hierarchical Markov model. Gray lines shows vertical transitions. The horizontal transitions are shown as black lines. The light gray circles are the internal states and the dark gray circles are the terminal states that returns control to the activating state. Let  $T$  represent the transition matrix. Suppose we have  $N$  statuses (operations) in the system, the size of  $T$  will be  $N \times N$ . Each element in  $T$  gives the probability of transition from one status to another. For example, the  $i$ th row gives the transition

probability from status  $i$  to all the other statuses.  $T[i][j] = k, (0 \leq k \leq 1)$  means the status  $i$  can be followed by status  $j$  with the probability  $k$ .

Apply it to the wiimote operation sequence. The top node indicates the entrance to the web site, the second level gathers three nodes representing three main modules of the web site. The third level contains more detail functional nodes provided by the web site. Lines among nodes represent possible operation ways. The nodes in the same level has the lines connected between each other. Each node has some child nodes in the next level with lines connected.

### 3.3 On-Line Adaptation

It is apparent that new user behavior data should be utilized to improve the accuracy of our algorithm. A model with online updating will change itself with the real-world environment to be adaptive and intelligent. In our experiment we aim to update one or more transition matrixes. In this paper, we propose two alternative ways based on transition matrix: no history data, and constant number history data.

The first one, we just use the latest data to update the transition matrixes. Suppose we get a new instance user behavior data, say, from node  $i$  to node  $j$ . If we successfully predict node  $j$ , we can remain  $T$  unchanged. But if next node we predict is node  $k$ , then we should update  $T$  by the formula below.

$$\begin{aligned} T[i][k] &= T[i][k] + \delta \\ T[i][j] &= T[i][j] - \delta \end{aligned} \quad (1)$$

The value of  $\delta$  is the critical task, which can be determined by some experiments.

The second alternative method, we can preserve a constant number of history data in the system, say  $N$ , as a database. When we receive new  $K$  instances, just replace them with the oldest  $K$  instances. Then we can rebuild the model by the new data with enduring computing time.

## 4 Experiments

### 4.1 Implement System

We setup our experiment on an education web site, which is a typical GUI based system, see Fig. 2. We will assign each node in the system with a unique indicator. Nodes in one level can be classified and share the same transition matrix. We make the prediction by two steps. Firstly, retrieve the user position in the system, say  $l$ . Then the system will predict the next most possible operation based on the transition matrix  $T_l$ . Updating of the model can be done based the new gathered wiimote control sequence.

### 4.2 Data Collection

First, we analyze the the web site and pick out all the possible operations, resulting in 72 nodes. Then we classify them into 5 classes, which means we will create 5

transition matrix in total. In our experiment, we employ 100 volunteers to use the education web site, each one is asked to visit the web site 20 times. And we get about  $100 \times 20 = 2000$  groups of behavior series. We randomly choose 80% of the these data as a training data set, and the rest as the testing data set.

### 4.3 Experimental Results

First of all, we train the transition matrix on the training data set. Then test the transition matrix on training data set and testing data set respectively. The result of the Hierarchical Markov Model is shown in Fig.4. The blue line represents the prediction accuracy on the training data set, while the red line represents the prediction accuracy on the testing data set. It is very easy to find out that prediction accuracy on the original training data set is better than that on the testing data set. We give the mean and variance statistics in Table. 1. According to the table, the mean prediction of the testing data is nearly 70%, while the we have about 75% on the original data. That both of their variances are very small implies that the prediction result is stable.

Following, we exam the first updating algorithm —updating the model with constant history data set. In our experiment, we choose 50 user’s behavior series stored in the system. Once we receive new instance of user’s behavior, we replace the oldest data with new gathered data. Model is updated based on the new history data set. Fig.5 gives the testing result. We can find that the prediction accuracy is a little better than the model without any updating. We

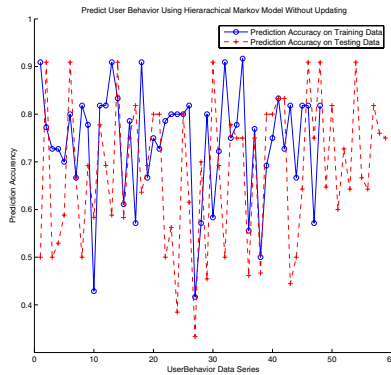
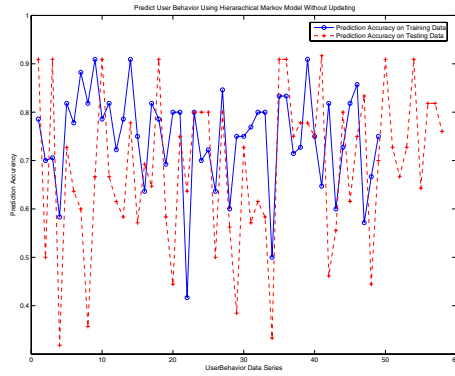


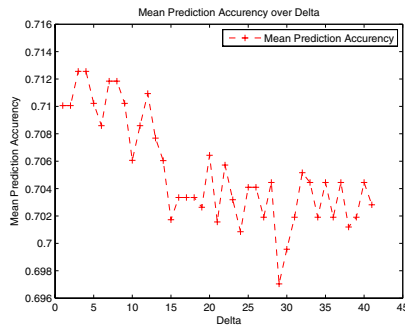
Fig. 4. Test Accuracy of the Hierarchical Markov Model Without out Updating

Table 1. Some Statistics of the Prediction Result over all the models

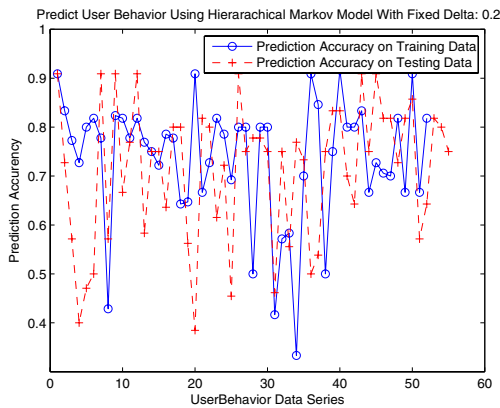
Model	TrainingMean	TrainingVariance	TestingMean	TestingVariance
ModelWithoutUpdating	0.7466	0.0137	0.6903	0.0225
ModelUpdatedWithConstData	0.7227	0.0197	0.7134	0.0181
ModelUpdtedWithDelta(0.2)	0.7373	0.0164	0.7129	0.0203



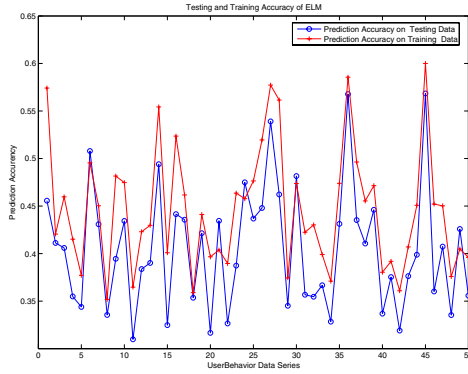
**Fig. 5.** Test Accuracy of the Hierarchical Markov Model With Const History Data



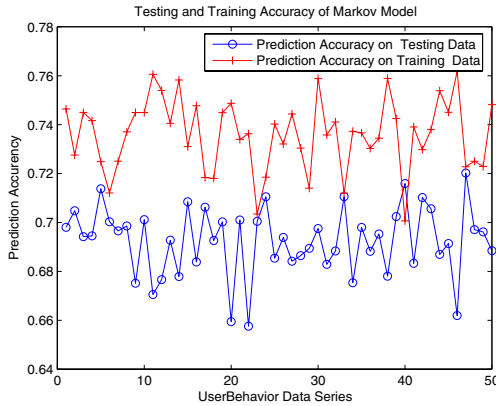
**Fig. 6.** Test Accuracy over  $\delta$



**Fig. 7.** Prediction Accuracy when  $\delta$  is 0.2



(a) Experiment Result Using ELM



(b) Experiment Result Using Hierarchical Markov Model

**Fig. 8.** Comparison Between ELM([G.-B. Huang:2006]) and Hierarchical Markov Model

also notice that prediction on some instance is rather low, even below 50%. It will be discussed in Section 5.

Another updating algorithm seems to be quite simple, for it need not recompute all the history data set and just update the model by the newest received data. In this method, one of the core problem is how to decide  $\delta$ . Fig.6 shows the result of testing  $\delta$  from 0.1 to 0.3 with a step of 0.055. According to the figure, there will be a higher prediction accuracy when  $\delta \in 0.1 \sim 0.2$ . Larger or smaller value will cause over-computing and decrease the prediction accuracy.

In Fig.7, we choose  $\delta = 0.2$  to make a model according to the third solution. Table. 1 gives all results of these models. All of them have the mean accuracy about 70%, and the variances are also quite small(smaller than 0.03). In this viewpoint, we may come to a conclusion that the original hierarchical Markov



model is good enough, and other improvement tries make little contribution to the prediction result.

Compared to other models, such as neuron network, hierarchical Markov model is more suitable for the user behavior model. Fig.8 gives a typical comparison. It is obvious that our model (72%) is much better than neuron network (35%). This result should be ascribed to the similarity between the structure of wiimote operation sequence and hierarchical markov model, which is the weak point of neuron network.

## 5 Conclusion and Future Work

In our experiment, we ask the volunteers some questions, such as whether it improve the usability, whether we bring in new inconvenience in operation. Most of them think the recommendation provided by the system is helpful. However, some argue that, if they do not operate normally, the prediction is often wrong. It is because the system is trained for normal user behaviors, and once it encounters some abnormal operation series, the reaction would not meet the user's requirement. One of the solutions is provide two or more recommendation options. Another question is that which recommendation style is appreciate. Here we skip the input focus into the recommendation operation menu item. Most of the volunteers think this is helpful, and some of them suggest using pop-up dialogs to provide two or more recommendation. These advices are useful for improving our system in the future.

Devices like wiimote empower GUI. The model with on-line adaptation we setup provides a solution of intelligence interface, which is the main character of next generation pervasive computing. It greatly enhances the user operation experience. The future of pervasive computing HCI should absorb advances from such exciting tries.

## Acknowledgements

We would like to thank Juan Liu for data collection and all of volunteers. This work is support of the National HighTechnology Research and Development Program("863"Program) of China(Grant No.2007AA01Z305) and National Natural Science Funds(Grant No.60775027).

## References

- [Kallio:2006] Kallio, S., Kela, J., et al.: User independent gesture interaction for small handheld devices. *International Journal of Pattern Recognition and Artificial Intelligence* 20(4), 505–524 (2006)
- [Crampton:2007] Crampton, N., Fox, K., et al.: Dance, Dance Evolution.:Accelerometer Sensor Networks as Input to Video Games. *Haptic, Audio and Visual Environments and Games* (2007)

- [Kela:2006] Kela, J., Korpipaa, P., et al.: Accelerometer-based gesture control for a design environment. *Personal and Ubiquitous Computing* 10(5), 285–299 (2006)
- [Hook:2000] Hook, K.: Steps to take before intelligent user interfaces become real. *Interacting with Computers* 12(4), 409–426 (2000)
- [Insung:2007] Insung Jung, D.T., Wang, G.-N.: Intelligent Agent Based Graphic User Interface (GUI) for e-Physician (2007)
- [Lester:2000] Lester, W.L., Johnson, J.W., Rickel, J.C.: Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments (2000)
- [Lieberman:1995] Lieberman, H.: Letizia: An Agent That Assists Web Browsing (1995)
- [Virvor, Kabassi:2002] Virvou, M., Kabassi, K.: Rendering the interaction more human-like in an intelligent GUI. *Information Technology Interfaces* (2002)
- [Etzioni, Weld:1994] Etzioni, O., Weld, D.: A Softbot-Based Interface to the Internet. *Communications of the ACM* (1994)
- [nintendo] <http://www.nintendo.com>
- [Lee, Kim:2008] Lee, H.-J., Kim, H., et al.: WiiArts: Creating collaborative art experience with WiiRemote interaction. In: *Proceeding of the Second International Conference on Tangible and Embedded Interaction* (2008)
- [Xu Guang-You:2007] Guang-You, X., Li-Mi, T., et al.: Human Computer Interaction for Ubiquitous/Pervasive Computing Mode (2007)
- [Meyn and Tweedie:2005] Meyn, S.P., Tweedie, R.L.: *Markov Chains and Stochastic Stability*. Cambridge University Press, Cambridge (2005)
- [G.-B. Huang:2006] Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme Learning Machine: Theory and Applications. *Neurocomputing* 70, 489–501 (2006)
- [Fine:98] Fine, S., Singer, Y., Tishby, N.: The hierarchical hidden markov model: Analysis and applications, 41–62 (1998)