Extraction and Classification of User Behavior

Matheus L. dos Santos[1], Rodrigo F. de Mello[1], and Laurence T. Yang[2]

[1] University of São Paulo, Institute of Mathematics and Computer Science
São Carlos, SP, Brazil
{matheusl, mello}@icmc.usp.br
[2] St. Francis Xavier University, Antigonish, NS, Canada
lyang@stfx.ca

Abstract. The multimedia document generator, iClass system, has been used by professors from the Institutes of Chemistry, Mathematics and Computer Science from the University of São Paulo aiming at helping the multimedia content production and availability. Data from user interactions, available in iClass system, have motivated this work which aims at studying the user behavior under different circumstances. The behavior extracted makes possible the analyze of different patterns of the same user, among groups, and distinct users. Those pattern differences should help to understand user evolution in iClass system under diverse situations. The data are grouped by a neural network and afterwards Markov Chains are built to represent their behaviors in different time moments. The detected user or group behavior variations are related to classify profiles and comprehend them in different situations.

1 Introduction

The iClass system, developed by the Intermídia Laboratory at University of São Paulo, captures information from conventional environments (such as classrooms), allowing the production of multimedia documents which, afterwards, are delivered through the Web [1]. This system has been adopted to help professors during classes.

During the classes, students make experiments, works and tests using iClass system. The iClass works as a *whiteboard*, generating databases containing user interaction information. Such information is stored in XML[3] documents, which stores the interaction instant and the number of points defined by users. Words and pictures are composed in groups of points, named strokes.

The availability of such information motivates the study of user behaviors under different circumstances (experiments, tests, classes, etc). The extracted behavior makes possible the analysis of different user patterns, among groups and distinct users. Such pattern differences help to understand user evolution in the iClass system under diverse situations.

This behavior study is conducted applying statistics techniques (like Markov Chains [2][3]), classification by neural networks (SOM [4][5][6], ART [7]) and

[3] Extensible Markup Language – http://www.w3.org/XML.

detection of changes in pattern behavior by suing energy variation (entropy [8][9][10][11]). The statistics techniques help to detect differences in the interaction intervals. Users at low interaction intervals should demonstrate more familiarity with the tool or with the subject covered in classes. Neural networks assist the behavior classification of user interactions, resulting in groups (or *clusters*), which represent interaction states such as: low, medium and high interaction. The variation between such states help to understand user behaviors.

This paper proposes a model to identify and classify user behavior patterns to allow comparisons in different moments and understand their interactions in computational systems.

This paper is organized as follows: section 2 presents related work; The methodology used to classify user behavior is presented in section 3; section 4 compares two users interacting in the iClass system; Conclusions and considerations are presented in section 5; The plans for future works are described in section 6.

2 Related Work

Works considering methods of user behavior classification are found in literature. The search for information under the human behavior is a concern of diverse research areas such as [12][13][14][15][16][17].

Brosso [12] considers an user authentication system in computer networks, which uses behavior analysis and face recognition to obtain the confidence degree to identify a system user. By analyzing the user behavior, the author uses the concept of Context-aware Computing which is based in the study of applications that adapt to locality and changes that occur to people and objects during time [18]. Besides context-aware computing, this work also adopts the five semantic dimensions defined by Abowd [19][20], which are used to specify and model context information. Those dimensions characterize the relevance of information (*Who*, *Where*, *What*, *When*, *Why*). In this way, to analyze user behavior, matrix is defined which contains information based in context-aware computing. Such information allows to understand user actions, locality, the period of interaction, if some behavior is an habit (*why*) and the confidence restrictions effected by user.

Godoy and Amandi [13] consider a technique to generate user profiles by observing their interaction characteristics on the Web. This technique is inserted in the algorithm *Web Document Conceptual Clustering* [21], which allows to acquire profiles without having any previous knowledge about user interests. The profile of user interests is organized in an hierarchical tree, where at the highest level are the general interests and at the lowest, the most particular ones. Those interests can be any information accessed by the user such as: sports, works, news and games. The relevance degree of the user interests is measured by means of term frequency, according to the document access rate.

Lee *et al.* [14] consider a new load balancing policy for Web servers named PRORD (*Proactive Request Distribution*). PRORD preloads the most accessed

web pages, based on the probability of future accesses, for this, the system analyzes information about web server caches. Using such preloading scheme, the web server anticipates pages with high access probability, decreasing the response time and improving the efficiency.

Macedo *et al.* [15] proposes a system, named WebMemex, which recommends information to users by analyzing the navigation description of known users. The WebMemex in conjunction with a proxy web server, which provides user history requests to WebMemex. In this way, WebMemex captures information such as IP addresses, user IDs, user active time in system and the URL accessed. Such information is stored in a relational database for future analysis.

Pepyne *et al.* [16] propose a method to classify user profiles using queue theory and logistic regression. This work explores system application in computer network security. The objective is to identify profiles of specialized user groups, such as bank tellers, which execute periodic computer tasks, from where it is possible to detect frauds using anomaly analysis of user behaviors.

Schuler and Perez [17] apply data mining techniques to discover user profiles in telecommunication systems. Two techniques of data mining are used: decision trees and neural networks. The rules generated by the decision tree represent the general profile of default users. Having the defined tree, historical data can be compared to any user to verify his/her relation to the determined class. Authors have concluded that decision trees represent user pattern/profile behavior, but present a great number of sub-groups becoming the comprehension impracticable.

As previously studied, the majority of works uses classification methods that, in some way, depends on the system or application goal. Such methods take into consideration the semantics of the analyzed data, and therefore they cannot easily be applied to other systems. Considering such problem, the work presented in this paper is motivated to develop a extraction and classification model to detect user behavior in any system or application.

3 The Model

This paper proposes a model to identify and classify user behavior patterns to allow comparisons in different moments and understand their interactions in computational systems. The model proposed in this paper is based on the study and evaluation of user interaction datasets; classification using artificial neural networks (ANNs); representation of user profiles using Markov chains; measure the energy variation to represent user behaviors; analyze user behavior patterns, comparing to other users and groups.

The first step of this work intended to create probability distribution functions (PDFs) to represent the user interactions. Consider the table 1 as an example of user interaction which contains the data available in iClass. The data represent user interactions to the system. Each interaction has a timestamp and the quantity of points (geometric forms or writing) made by an user.

Table 1. Example of user interaction in the iClass system

Time	Points	Object
12:03	1500	geometric shapes
12:05	200	geometric shapes
12:07	400	writing
12:08	200	writing
12:10	400	writing
12:11	200	writing
12:14	900	geometric shapes
12:17	300	writing
12:19	400	geometric shapes
12:20	50	writing
12:22	400	writing
12:24	400	geometric shapes
12:25	200	writing
12:26	200	writing
12:28	1200	writing
12:31	900	geometric shapes
12:35	1600	geometric shapes
12:38	900	writing
12:39	200	writing
12:40	400	geometric shapes

Table 2. Representation of user interaction in time intervals

Time Intervals	Points
12:00 – 12:03	1500
12:03 – 12:05	200
12:05 – 12:07	400
12:07 – 12:08	200
12:08 – 12:10	400
12:10 – 12:11	200
12:11 – 12:14	900
12:14 – 12:17	300
12:17 – 12:19	400
12:19 – 12:20	50
12:20 – 12:22	400
12:22 – 12:24	400
12:24 – 12:25	200
12:25 – 12:26	200
12:26 – 12:28	1200
12:28 – 12:31	900
12:31 – 12:35	1600
12:35 – 12:38	900
12:38 – 12:39	200
12:39 – 12:40	400

Table 3. Example of the probability distribution on the user interaction

Interval	Points	Interval	Points
00 – 01	500	20 – 21	200
01 – 02	500	21 – 22	200
02 – 03	500	22 – 23	200
03 – 04	100	23 – 24	200
04 – 05	100	24 – 25	200
05 – 06	200	25 – 26	200
06 – 07	200	26 – 27	600
07 – 08	200	27 – 28	600
08 – 09	200	28 – 29	300
09 – 10	200	29 – 30	300
10 – 11	200	30 – 31	300
11 – 12	300	31 – 32	400
12 – 13	300	32 – 33	400
13 – 14	300	33 – 34	400
14 – 15	100	34 – 35	400
15 – 16	100	35 – 36	300
16 – 17	100	36 – 37	300
17 – 18	200	37 – 38	300
18 – 19	200	38 – 39	200
19 – 20	50	39 – 40	100

In order to understand how the PDF should be useful, a better representation of the table 1 is presented in the table 2. For this last table, the number of points in time intervals can be observed. For instance: in the interval of $12:00$ to $12:03$ the user has produced 2000 points, this is, 2000 points in 3 minutes.

The data from the table 2 can be distributed in constant time intervals to simplify the PDF generation. The table 3 presents interaction data distributed in discrete time intervals of 1 minute, where the number of points p for interval is given by $p = \frac{np}{i}$ being np the quantity of points in the interval i. To better illustrate the construction of the table 3, consider the time interval between $12:03$ and $12:05$ (table 2), which has 2010 points in 2 minutes, in this case the time intervals must be divided in two periods of 1 minute: one from $12:03$ to $12:04$ and another from $12:04$ to $12:05$, with the number of points in each new same-sized interval to 1005 ($\frac{2010}{2} = 1005$) points per minute.

The PDF from the table 3 is presented in the figure 1. In this case, there is a large user interaction variation to the system.

In the second stage, the data densities of the PDF were classified by an ANN. Such classification was made submitting input vectors (table 3) in the form $\frac{np}{i}$ where np is the number of points and i is the time interval, for example: $[\frac{28}{1m}, \frac{10}{0.5m}, ..., \frac{50}{3m}]$. In this stage the ANN has created groups (*clusters*) in accordance with the inputs. When a pattern does not fit in an existing group, a new one is created to receive it. The figure 2 represents the interaction sequence of the ANN, illustrating the creation of groups.

In the third stage, having the groups, Markov Chains are built to represent user profiles. The figure 3 presents an example of a Markov Chain, where the circles represent states (groups) and the arcs indicate transitions in accordance with their respective probabilities.

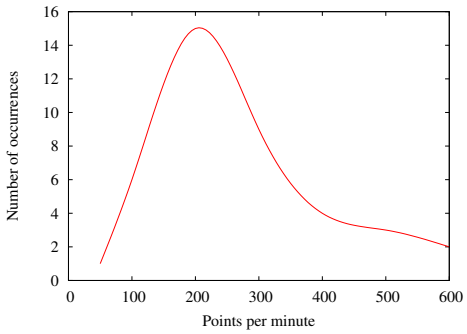


Fig. 1. Graphical representation of the probability distribution of user interaction

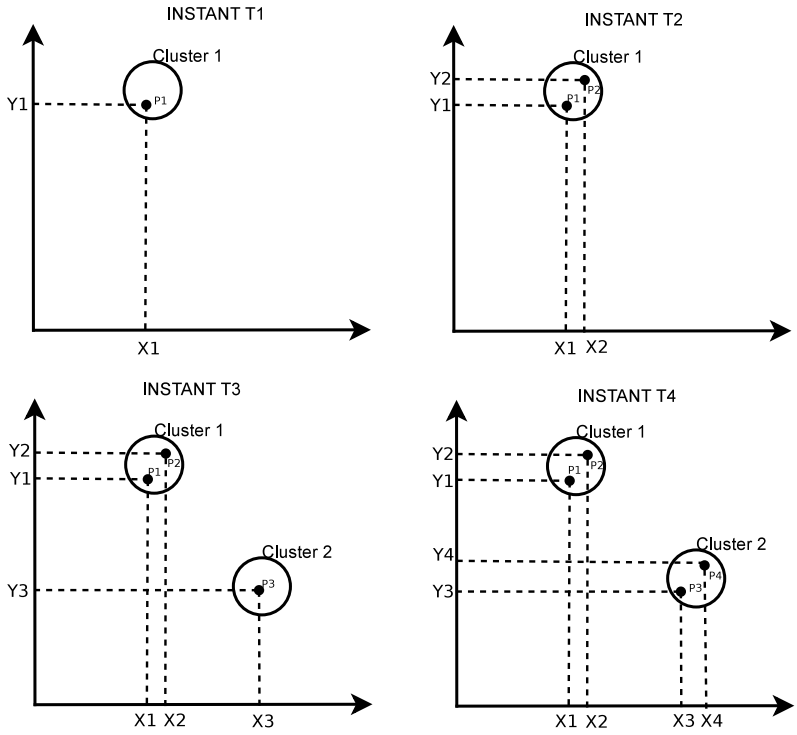


Fig. 2. ANN classifying patterns

As previously observed, the ANN creates a new group when a pattern does not correspond to a group. The creation of a new group indicates the occurrence of an unexpected system pattern. Groups, hardly ever visited, also depict un-

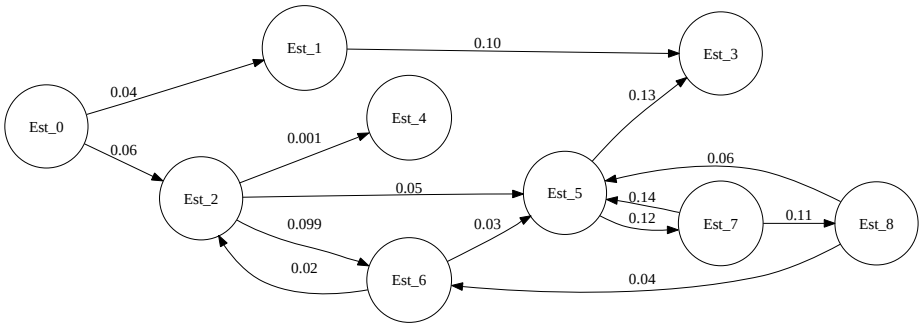


Fig. 3. Example of a Markov Chain representing a user profile in a time instant

expected behavior. To detect such new features [22], the system energy metric (entropy) is used.

The figure 4 illustrates as the energy behaves at every instant of a Markov Chain. At the instant t_0, as the probabilities to change from a state are equal, there is almost no behavior variation, therefore the energy E_0 is low. At the instant t_1, a system variation is caused by the creation of an unexpected state Est_2, which increases the variation in the probabilities of state change and, consequently, the energy goes up to $E_1 = 0.92637354$ (the energy variation is equal to $\Delta E = 0.23322636$).

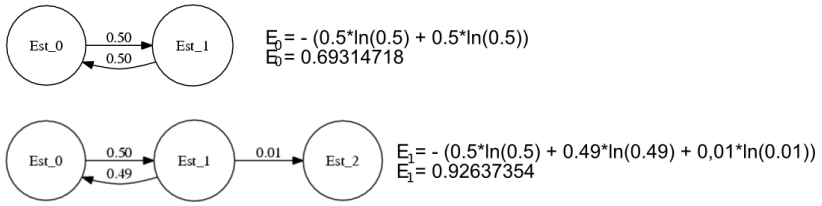


Fig. 4. Example of energy measurements in the Markov Chain

In the fourth stage user behaviors are analyzed. The evaluation of behavior differences are carried out comparing the group labeling obtained by the ANN, joining to the Markov chain of the user at the instant t_0 and t_1. The labeling is represented by vectors which summarize the characteristics of the classified patterns in a certain group of the ANN. Those vectors are in the form $r = [r_0, r_1, ..., r_{n-1}]$, where n is the number of input data and r is the relevance of the input. To allow the comparisons among groups, it is necessary to normalize vectors $r \in [0, 1]$ dividing each element for the summing up of the elements $(\frac{r_{n-1}}{r_0+r_1+...+r_{n-1}})$.

The behavior of an user is compared in each instant, this is, the instant t_0 is compared (using the labeling vectors) to t_1, t_2, and so forth, to detect behavior changes in user interactions. Different users can also have their behavior compared. That comparison is usually made considering the behavior at the same time instant t_k.

4 Experiments and Results

Experiments were conducted to evaluate the proposed model. In iClass, the information about the user interaction is stored, in a XML file named "session.xml". In this file exists several *tags* which contain information about user interaction such as: user name, page resolution, pen color, *timestamp* of *stroke*, number of points per *stroke* and others.

A parser using the Java language was developed to extract information about user interactions from the file "session.xml". From these information (number of points and *timestamps* of each *stroke*), probability distribution functions were generated to analyze data.

Initially, user interactions generated during sessions of the Sudoku[4] game were captured (figure 5). The game took place in the iClass system. From such data, some probability distributions were created to characterize and understand user behavior (figure 6).

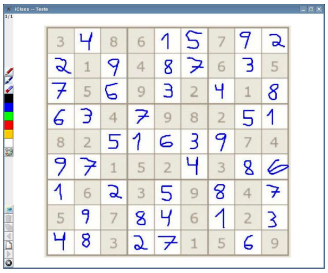
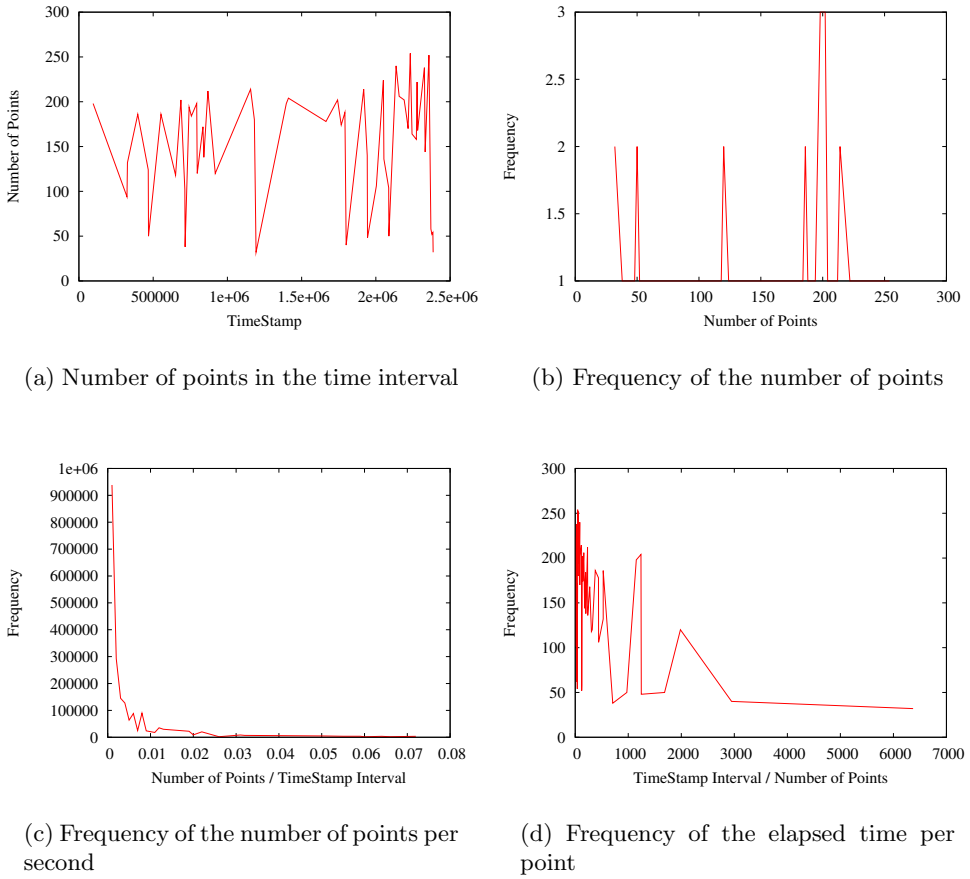


Fig. 5. User playing the Sudoku game in the iClass system

The figure 6(a) presents a direct data distribution, this is, the data extracted from the file "session.xml" without modifications. It is difficult to identify some characteristics by using such figure, for instance, the elapsed time in each interaction (user knowledge about the process) and the time interval between two interactions (thinking time). In the same way, the data presented in the figures 6(b), 6(c) and 6(d) do not directly represent user behavior, therefore the data in these charts are grouped according frequencies, and in this way, information

[4] http://en.wikipedia.org/wiki/Sodoku

as the thinking time and the user knowledge are mixed, making difficult their identification.



(a) Number of points in the time interval

(b) Frequency of the number of points

(c) Frequency of the number of points per second

(d) Frequency of the elapsed time per point

Fig. 6. Example of data distributions on the user interactions in the Sudoku game

The found solution was to represent the distribution demonstrated in the figure 6(d) in a time discrete manner, in this way, the data on the elapsed time in each user interaction is represented at the same time interval, making possible the direct comparison between the chart and the user interaction. The figure 7 represents this new data distribution, where can be visualized the time intervals in which the user interacts to the system (Sudoku game in such case). Each unevenness presented in the charts depicts an user action in the game.

However, any distribution will present a deficit in the data representation, not separating the interaction time from the time between interactions. This occurs

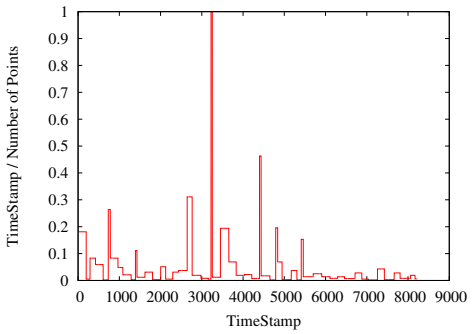


Fig. 7. Distribution representing the elapsed time per point throughout the user interaction

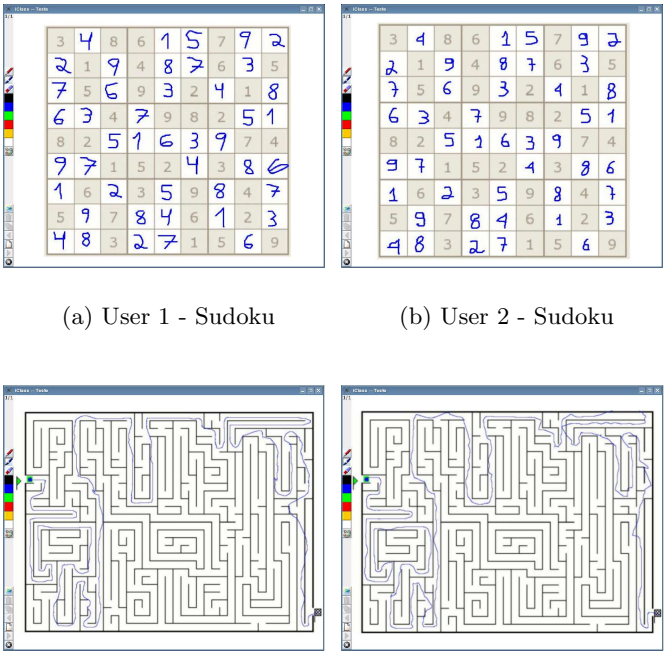
because the way the information is stored by iClass. The extracted information from iClass is in the form: *timestamp* for the number of points drew by users. In this way, to obtain the interaction time of one *stroke*, the *timestamp* of the next has to be deducted from the current one. The stroke interaction time represents the moment of user interaction, the idle time between interactions is common as the user make an action and later spends a time until the next one initiates.

After having defined the data distribution to represent user behavior, the experiments were conducted to evaluate the interaction of two different users. Those users had interacted, by means of the iClass system, to the Sudoku game and had solved a maze problem. The figure 8 represents the end of the interaction carried out by User 1 and User 2.

Afterwards, each user interaction behavior is represented by the data distribution extracted from the file "session.xml". The data distributions of interaction of each user is demonstrated in the figure 9, where the behavior of the User 1 interacting to the Sudoku game and the maze is respectively represented by the figure 9(a) and figure 9(c) and, the behavior of the User 2 is represented by the figure 9(b) and 9(d), respectively.

The next steps, following the model in the section 3, is: to carry out the data classification, building the Markov chains and later measuring the average energy variation between the two chains (entropy). Using the neural network SONDE [23], the data distributions represented in the figure 9 were classified. As the main purpose of SONDE is to detect new features (novelties) by analyzing energy variations in datasets, the Markov chains and the energy variation chart were automatically generated. The figure 10 represents only the Markov chains at the last user interaction in the Sodoku and in the Maze (remember each time instant is represented by a Markov chain).

After generating the Markov chains, the user behavior variation is represented by the energy variation between Markov Chains. This behavior variation is represented in the figure 11.



(a) User 1 - Sudoku (b) User 2 - Sudoku

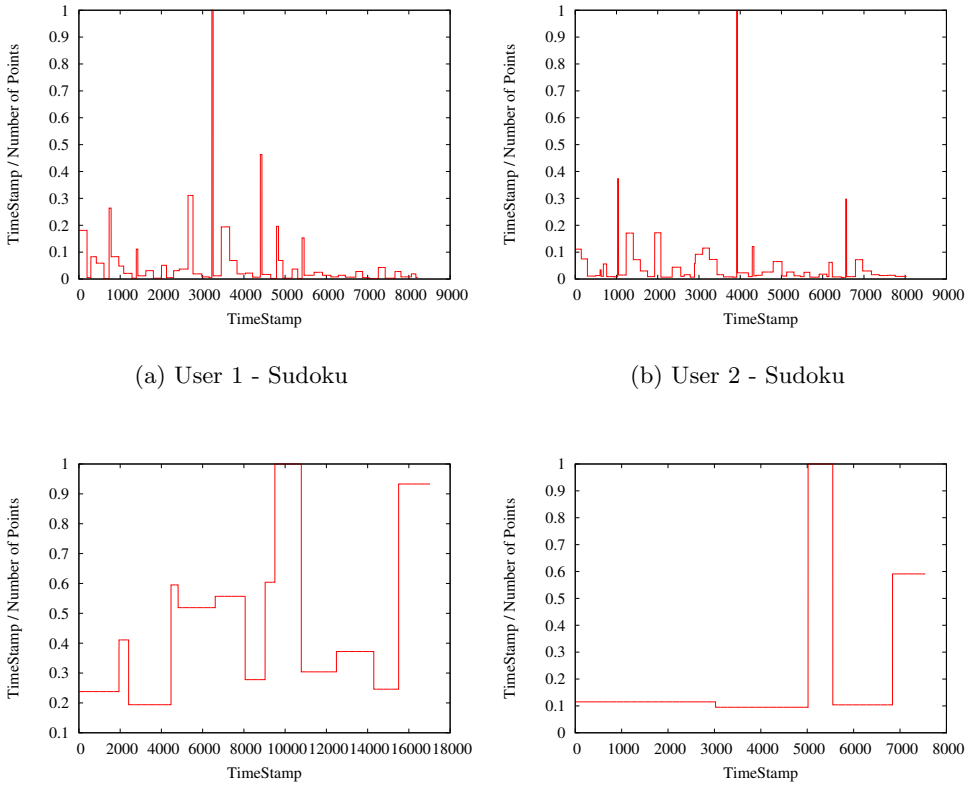
(c) User 1 - Maze (d) User 2 - Maze

Fig. 8. Two users interacting to the iClass System

By analyzing the figure 11, User 1 keeps some characteristics in the two interactions (Sudoku and Maze), what also occurs to User 2. In the figure 11(a), some stable points in the user behavior (declivity) are observed, this characteristic also is observed in the figure 11(c). A detailed observation of the User 2 allows to notice a common pattern among the interactions. In the figure 11(b), the energy level is increasing, presenting steps, and the same occurs in the figure 11(d). By this, we confirm that User 2 contains a bigger dynamism in his/her actions, not having pauses throughout interactions. User 1 also presents a level of increasing energy, although, differently from User 2, the User 1 probably present pauses during its interactions. Maybe such pauses are related to the thinking time throughout interactions, in contrast of User 2 that thinks about the problem before starting deciding.

5 Conclusion

This paper proposes a model to extract data from user interactions to detect and classify user behavior patterns. The model summarizes user interactions through Markov Chains, and the user behavior profile is represented by a set of Markov



(a) User 1 - Sudoku

(b) User 2 - Sudoku

(c) User 1 - Maze

(d) User 2 - Maze

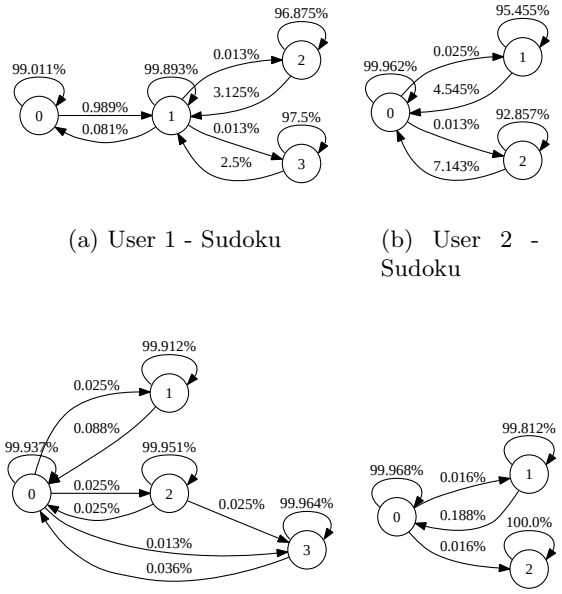
Fig. 9. Interaction data probability distribution of users in the iClass system

Chains. The energy variation (entropy) between Markov Chains represents the user behavior variation along interactions.

Experiments were conducted to evaluate the proposed model in the iClass system. The obtained results show that the model detects users behavior profiles interacting in the iClass system. Even interacting in distinct problems, users may present similar behavior characteristics between experiments. It is important to notice that such results are preliminary and are not conclusive.

With the manipulation and analysis of user interaction data, we may establish relations among different behaviors and profiles, being possible to relate an user or group actions in classrooms to their test performances or still to identify an user by analyzing interactions.

Using techniques of energy measurement (entropy) in a system, it is possible to detect user behavior changes, relate them to unexpected events such as some special fact occurred in classroom, personal or medical problems and others.



(a) User 1 - Sudoku

(b) User 2 - Sudoku

(c) User 1 - Maze

(d) User 2 - Maze

Fig. 10. Markov chains at the last user interaction in the iClass system

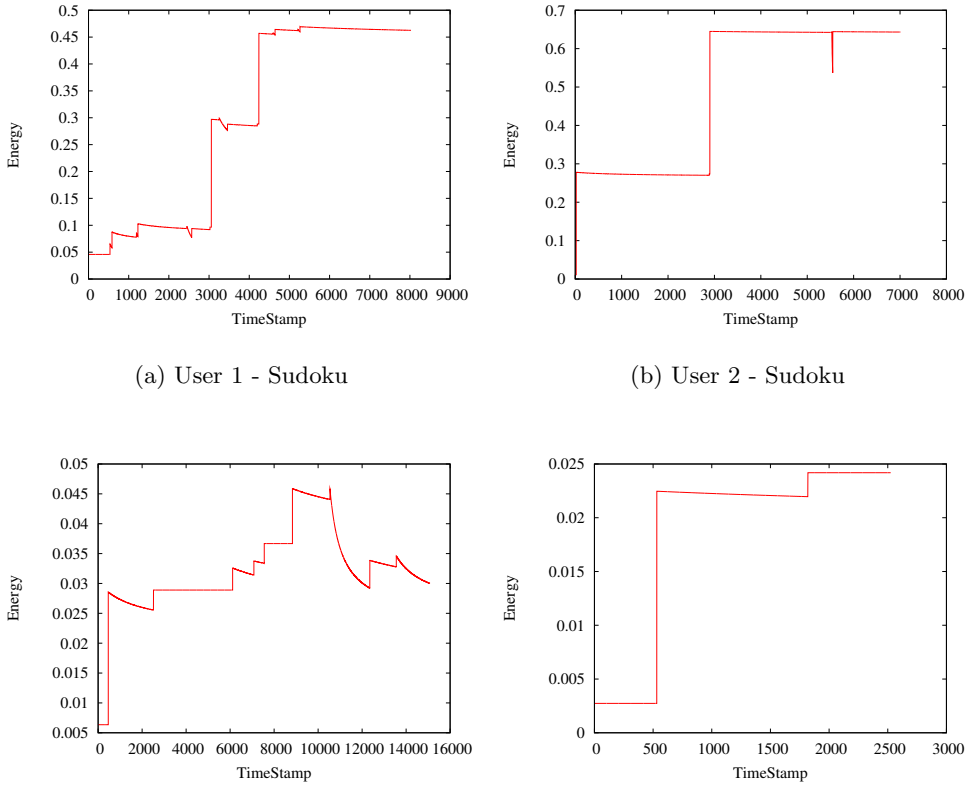
Besides establishing user behavior patterns, it is also possible to anticipate user actions. This technique can assist in the artificial intelligence of games, Web sites recognizing users from interactions.

6 Future Work

The results presented in this paper were obtained by experiments conducted with two users. In order to give statistical relevance such experiments will be executed in a larger population. Besides the experiments, techniques of validation and comparison of user behavior patterns have been studied.

7 Acknowledgments

This paper is based upon work supported by CAPES, Brazil under grant no. 032506-6 and FAPESP, Brazil. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the CAPES or FAPESP.



(a) User 1 - Sudoku

(b) User 2 - Sudoku

(c) User 1 - Maze

(d) User 2 - Maze

Fig. 11. Variation in the user pattern behavior in the iClass system

References

1. Cattelan, R.G., Andrade, A.R., Rocha, C.F.P., Pimentel, M.d.G.C.: iclass: um sistema para captura e acesso de sessões em ambiente educacional. Revista Eletrônica de Iniciação Científica - REIC **3**(1) (2003) 10–28
2. Grinstead, C.M., Snell, J.L.: Introduction to Probability. American Mathematical Society; 2nd Rev edition (July 1, 1997), United States of America (1997)
3. Nogueira, F.: Cadeias de markov. http://www.engprod.ufjf.br/fernando/epd042/cadeiaMarkov.pdf (02 2006)
4. Kaski, S., Oja, E.: Kohonen Maps. Elsevier Science Inc., New York, NY, USA (1999)
5. Kohonen, T., Kaski, S., Lagus, K., Salojrvi, J., Honkela, J., Paatero, V., Saarela, A.: Self organization of a massive document collection (2000)
6. Makhfi, P.: Competitive learning. http://www.makhfi.com/tutorial/clearn.htm (2006)

7. Carpenter, G.A., Grossberg, S.: The ART of adaptive pattern recognition by a self-organizing neural network. Computer **21**(3) (1988) 77–88
8. Shannon, C.: A mathematical theory of communication. Bell System Technical Journal **27** (July and October 1948) 379–423 and 623–656
9. Santos, E.M.d.S., Albuquerque, M.P., Mello, A.d.R.G., Caner, E.S., Esquef, I.A.: Fundamentos da teoria da informação. Technical report, Centro Brasileiro de Pesquisas Físicas (Dezembro 2004)
10. Boltzmann, L.: Vorlesungen uber Gastheorie. Volume 1, 2. J. A. Barth Leipzig (1896) English Translation by S.G. Brush: Lecture on Gas Theory, Cambridge Univ. Press, Cambridge, 1964.
11. Freeman, J.A., Skapura, D.M.: Neural networks: algorithms, applications, and programming techniques. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA (1991)
12. Brosso, M.I.L.: Autenticação Contínua de Usuários em Redes de Computadores. Tese de doutorado, Politécnica da Universidade de São Paulo, São Paulo, SP, Brasil (2006)
13. Godoy, D., Amandi, A.: User profiling for web page filtering. IEEE Internet Computing **9**(4) (2005) 56–64
14. Lee, H.K., Vageesan, G., Yum, K.H., Kim, E.J.: A proactive request distribution (prord) using web log mining in a cluster-based web server. In: ICPP '06: Proceedings of the 2006 International Conference on Parallel Processing, Washington, DC, USA, IEEE Computer Society (2006) 559–568
15. Macedo, A.A., Truong, K.N., Camacho-Guerrero, J.A., da Graça Pimentel, M.: Automatically sharing web experiences through a hyperdocument recommender system. In: HYPERTEXT '03: Proceedings of the fourteenth ACM conference on Hypertext and hypermedia, New York, NY, USA, ACM Press (2003) 48–56
16. Pepyne, D.L., Hu, J., Gong, W.: User profiling for computer security. American Control Conference, 2004. Proceedings of the 2004 **2**(6) (2004) 982–987
17. Schuler, A.J.J., Perez, A.L.F.: Análise do perfil do usuário de serviços de telefonia utilizando técnicas de mineração de dados. RESI - Revista Eletrônica de Sistemas de Informação **7**(1) (2006)
18. Schilit, B., Theimer, M.: Disseminating active map information to mobile hosts. IEEE Network **8**(5) (1994) 22–32
19. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P.: Towards a better understanding of context and context-awareness. In: HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, London, UK, Springer-Verlag (1999) 304–307
20. Abowd, G.D., Mynatt, E.D.: Charting past, present, and future research in ubiquitous computing. ACM Trans. Comput.-Hum. Interact. **7**(1) (2000) 29–58
21. Godoy, D., Amandi, A.: Modeling user interests by conceptual clustering. Inf. Syst. **31**(4) (2006) 247–265
22. Markou, M., Singh, S.: Novelty detection: a review – part 2: neural network based approaches. Signal Process. **83**(12) (2003) 2499–2521
23. Albertini, M.K., Mello, R.F.: A self-organizing neural network for detecting novelties, ACM Symposium on Applied Computing (2007)