Global State Context Prediction Techniques Applied to a Smart Office Building

J. Petzold[1], F. Bagci[1], W. Trumler[1], T. Ungerer[1], and L. Vintan[2]

[1]Institute of Computer Science
University of Augsburg, Germany
{Petzold, Bagci, Trumler, Ungerer}@Informatik.Uni-Augsburg.DE

[2]Institute of Computer Engineering
University of Sibiu, Romania
lucian.vintan@ulbsibiu.ro

Abstract

Ubiquitous systems use context information to adapt appliance behavior to human needs. Even more convenience is reached if the appliance foresees the user's desires and acts proactively. This paper introduces context prediction techniques based on previous behavior patterns, in order to anticipate a person's next movement.

We focus on two-level predictors with global first-level histories and two-state predictors respectively frequency analysis in the second level and compare these predictors with the Prediction by Partial Matching (PPM) method. We evaluate the predictors by some movement sequences of real persons within an office building reaching up to 69% accuracy in next location prediction without pre-training and, respectively, up to 96% with pre-training.

Keywords: context awareness, context prediction, location awareness, location prediction, proactive

1. Introduction

Humans are creatures of habit. Humans typically act in a certain habitual pattern, however, they sometimes interrupt their behavior pattern and they sometimes completely change the pattern. Our aim is to relieve people of actions that are done habitually without determining a person's action. The system should learn habits automatically and reverse assumptions if a habit changes. The predictor information should therefore be based on previous behavior patterns and applied to speculate on the future behavior of a person. If the speculation fails, the failing must be recognized, the speculatively initiated actions withdrawn, and the predictor updated to improve future prediction accuracy.

For our application domain we chose next location prediction instead of general context prediction. The algorithms may also be applicable for other more general context domains, however, there exist already numerous scenarios within our application domain. Some sample scenarios may be the following:

- Smart doorplates that are able to direct visitors to the current location of an office owner based on a location-tracking system and predict if the office owner is soon coming back [13].
- Similarly, next location prediction within a smart building can be used to prepare the room which is presumably entered next by an habitant.
- Outdoor movement patterns can be used to predict the next region a person will enter.
- Elevator prediction could anticipate at which floor an elevator will be needed next.
- Routing prediction for cellular phone systems may predict the next radio cell a cellular phone owner will enter based on his previous movement behavior.

To predict or anticipate a future situation learning techniques as e.g. Neural Networks [5], Markov Models [2] or Hidden Markov Models [12], and Bayesian Networks [6] are obvious candidates. The challenge is to transfer these algorithms to work with context information.

The Adaptive House project [9] of the University of Colorado developed a smart house that observes the lifestyle and desires of the inhabitants and learned to anticipate and accommodate their needs. Occupants are tracked by motion detectors and a neural network approach is used to predict the next room the person will enter and the activities he will be engaged. Hidden Markov Models and Bayesian Networks are applied by Katsiri [8] to predict people's movement. Patterson et al. [10] presented a method of learning a Bayesian model of a traveler moving through an urban environment based on the current mode of transportation. The learned model was used to predict the outdoor location of the person into the future.

Markov Chains are used by Kaowthumrong et al. [7] for active device selection. Ashbrook and Starner [1] used location context for the creation of a predictive model of user's future movements based on Markov models. They propose to deploy the model in a variety of applications in both single-user and multi-user scenarios. Their prediction of future location is currently time independent, only the next location is predicted. Bhattacharya and Das [3] investigate the mobility problem in a cellular environment. They deploy a Markov model to predict future cells of a user.

The problem of predictive Markov models is its slow retraining after a habit change. We show that predictive Markov models are special cases within our own predictor collection and we propose other predictors with better retraining speed. Moreover we propose predictors with higher prediction accuracy.

Our approach originates in branch prediction and data compression algorithms that are transformed and developed further to fit the scenario of context prediction. We proposed several one- and two-level predictors in [11] and evaluated them by synthetic benchmarks. This paper focuses on two-level predictors with global first-level histories and two-state predictors respectively frequency analysis in the second level which are compared with the PPM predictor. We evaluate the predictors by some movement sequences of real persons.

The next section describes the proposed context prediction algorithms and section 3 evaluates the predictors. The paper ends with the conclusions.

2. Context Prediction Algorithms

2.1. Application Example for Explanation and Evaluation

We explain and evaluate the proposed state context predictors by the scenario of next location prediction in an office building as used in our Smart Doorplate project [13]. Assuming that an office owner is absent of his own office and a visitor arrives in front of his office, the visitor has to decide if he shall wait. Here a prediction of the next location of the office owner is made. If the prediction is that the office owner comes back soon, the visitor can wait for him. Our sample office scenario is a floor of our institute like that presented in figure 1.

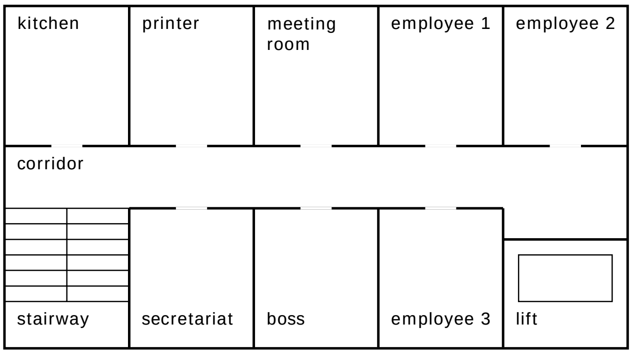

Figure 1. Floor plan

For an explanation of the predictors, we use a simplified floor plan with only four rooms (see figure 2) .

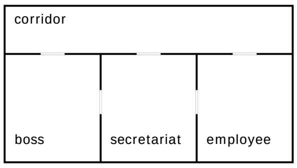

Figure 2. Floor plan of corridor, boss' office, secretariat, and employee's office

2.2. One-level Two-state Context Predictors

Similar to the branch prediction techniques used in computer architecture [14], our one-level predictors use only a single level—i.e. a single prediction table—for prediction, whereas a two-level predictor selects an entry within the prediction table by indexing from a first level of prediction which can be globally or locally defined. We start with the one-level context predictors that are applied in the second level of the more complex predictors below.

The two-state context predictor is derived from the two-bit branch predictor with saturation counter. The first entry denotes the next room to be predicted, the second entry is used for changing between the strong and weak states. The room stored in the first entry is thus always predicted independently of the second entry, which influences training and retraining speed. The denotation "two-state context predictor" stems from the provision of two states for each predicted room.

The prediction graph of a room—here the corridor—with three neighbor rooms (see figure 2) is shown in figure 3. The denotations of the states consist of the ID of the room and a counter. As a suggestive example, if a person enters for the first time the boss's office B from the corridor, the state B0 is set. If the person reenters the corridor, the office of the boss B is predicted as next location. If the prediction proves as correct, the predictor switches into the strong state B1. Thus, next time the office of the boss B will be predicted again. If the person interrupts her habit once by entering secretariat S or employee's office E, the state is set back from B1 to B0. Thus the boss' office is still predicted. If the person goes now from the corridor into the secretariat the predictor switches into the state S0 independently of the room entered from the corridor before, and predicts thus the secretariat as next.

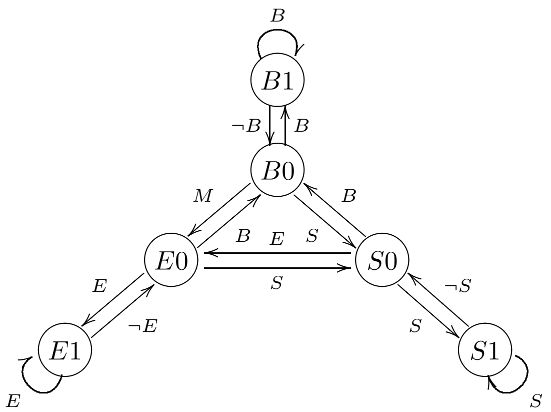

Figure 3. Prediction graph of two-state predictor for the corridor with three neighbor rooms (B - boss, E - employee, S - secretariat)

If a room has more than three neighbor rooms, the principle can be continued. The weak state node of each additional neighbor room will be connected to each of the existing weak state nodes (in figure 3: B0, E0, S0) and to the own strong state node. It should be noted that only the nodes with a 0 in the second entry, i.e. the weak states, form a completely connected graph as displayed in figure 3.

Besides the ID of the room that is predicted the storage costs of the two-state predictor are only an additional one-bit counter for each room. The computation costs for adapting the states are insignificantly larger than for the one-state predictor. The predictor is also rapidly trained. The retraining is slowed down such that an one-time change of the habit does not cause an effect. In the case of two successive deviations from the habit the system notes the change. If more than two deviations of a habit should not yet lead to a retraining, the number of states must be increased leading to a k-state context predictor with k>2. If retraining should be instantly, the one-state predictor can be used. Alternatively, the frequency of room entering can be stored. All one level predictors allow only to base the predictions on local movements from one room to its neighbor rooms.

2.3. Global Two-level Context Predictors with Two-state Predictors in the Second Level

The global two-level context predictors regard a sequence of the last rooms that a person entered to predict the next room. The visited rooms are stored in a kind of shift register that constitutes the first level of the predictor. If a new room is entered all entries of the register are shifted to the left and the new room is filled in from the right. The length of the shift register is called the order, which denotes the number of last visited rooms that influence the prediction. The second level consists of a pattern history table that stores all possible patterns of room sequences in different entries. Each entry holds additionally a two-state predictor entry. The pattern in the shift register is used to select an entry in the pattern history table.

We consider again the example with four rooms: C (corridor), B (office of the boss), S (secretariat), and E (office of the employee). Furthermore we assume an order of 3. Then there are $4 \cdot 3^2 = 36$ patterns and therefore 36 entries in the pattern history table. Figure 4 shows this case assuming the room sequence: C B S B S E C B S E C B S B S E C

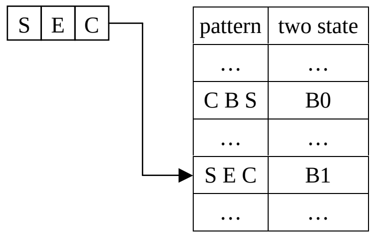

Figure 4. Two-level predictor with two-state predictor

After the first occurrence of the pattern S E C the initial state B0 is set for this pattern. The predictor changes to state B1 after the next occurrence of S E C. Now the prediction is that the boss' office B will be entered next.

The two-state predictor in the second level can be replaced by a k-state predictor with different k, in order to adjust the retraining speed.

2.4. Global Two-level Context Predictor with Frequency Analysis in the Second Level

Alternatively the second level stores for each pattern the frequencies of all previous accesses to all neighbor rooms from the current room, which is listed last in the pattern. Now the room with the largest frequency is predicted. This predictor corresponds to the Markov Predictor known from data compression [4], respectively the predictive Markov models applied by [1] and [3]. We consider the same example as above. Figure 5 shows the shift register and the pattern history table for the two-level predictor with frequencies.

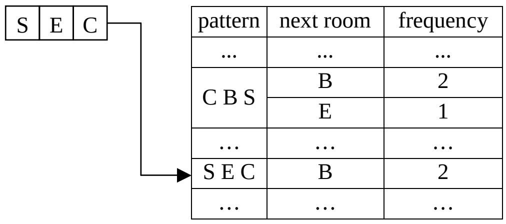

Figure 5. Two-level predictor with frequencies

An advantage of the global two-level predictors is that now complex movement patterns can be predicted. Since each pattern is treated separately, interferences between two patterns cannot appear as is the case in branch prediction [14].

A disadvantages of the frequency analysis method in the second level is that after many turns of a pattern, retraining needs a long time. For example if a room is entered 1,000 times after the same movement pattern, 1,000 times of entering another room after this pattern is needed before the prediction changes, whereas a two-state predictor in the second level is retrained after two mispredictions, able to correctly predict the changed habit.

2.5. Prediction by Partial Matching

The two-level context predictors can be extended using a method motivated by Prediction by Partial Matching (PPM) [4] from the area of data compression. Here a maximum order m is applied in the first stage instead of the fixed order. Then, starting with this maximum order m, a pattern is searched according to the last m rooms. If the pattern matches the predicted room will be the most frequently entered room after that certain movement pattern. If no pattern of the length m is found, the pattern of the length $m-1$ is looked for, i.e. the last $m-1$ rooms. This process can be accomplished until the order 1 is reached. If even a predictor of order 1 doesn't generate any prediction there will be no prediction. We propose to stop the PPM with order 1 because location prediction doesn't make sense when the currently room isn't known (in other words, a Markov predictor of order 0 hasn't sense in this context).

Analog to the two-level predictors we must distinguish two PPM predictors. First, the saturated PPM predictor which uses the two-level predictor with two-state predictor in the second level, and second, the non-saturated PPM predictor with frequency analysis in the second level.

3. Evaluation

For evaluation we have recorded the movements of several people over currently several weeks in the office building that is displayed in figure 1. We present here two movement sequences, one of the boss and one of an employee. In the sequences the corridor is omitted because it is the room that must be entered from every other room. Currently our movement sequences are limited to 160 movements for the boss and 365 for the employee, without considering the corridor. Over time we will enlarge the length of the sequences. Furthermore we omit to predict the target room when leaving the own office, because the target could be every other room, depending on the persons intention. Also in our Smart Doorplate scenario, prediction is only made if the office owner is absent from his office.

The evaluation is separated in two parts. First we have analyzed the two-level predictors with two-state predictor and frequency analysis as well as the corresponding PPM predictors. Second we performed an analysis with trained predictors, that means the predictors were trained with the movement sequences and after this the predictors run again. This scenario shows how the predictors behave on rather complex patterns and what may be reached by pre-training. The predictor labels shown in the charts are defined in table 1.

Table 1. Abbreviations for the context predictors

G-2L-2S(k)	global two-level predictor with two-state predictor and order k
G-2L-2S-PPM(k)	saturated PPM by G-2L-2S beginning with order k
G-2L-F(k)	global two-level predictor with frequencies and order k
G-2L-F-PPM(k)	PPM by G-2L-F beginning with order k

First we consider the untrained predictors. Both charts, boss (figure 6) and employee (figure 7), show that the predictors with a lower order have a better accuracy as the predictors with high order because the predictors with a higher order need more time for the learning process. Also the PPM predictors with two-state respectively frequency analysis have about the same prediction accuracy like the two-level predictors with an order of 1. Also here the reason is the slow training speed of predictors with high order.

When we compare both charts (boss and employee) we see that the prediction accuracy of employee's predictors are

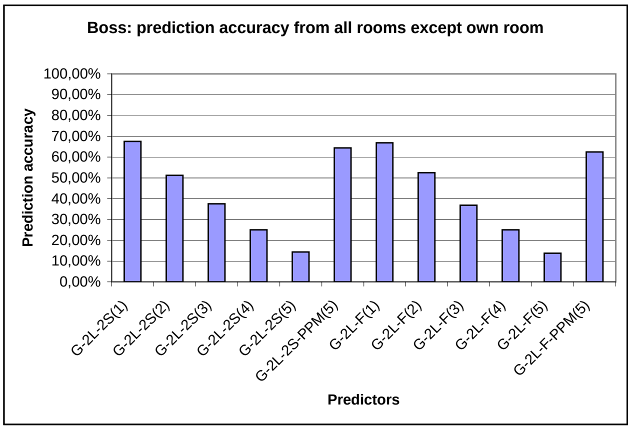

Figure 6. Prediction accuracy of boss' movement

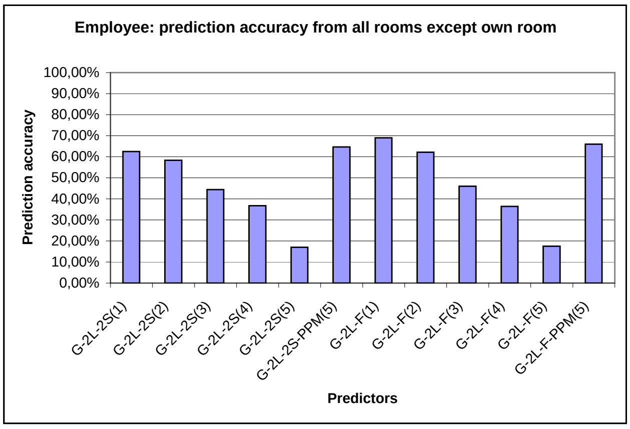

Figure 7. Prediction accuracy of employee's movement

mostly better than that of boss' predictors. An explanation for the better prediction's accuracy of the employee's movements is that the employee visited less people—mostly people within his own specific project—than the boss, who supervises several projects.

The charts in figure 8 and 9 show the prediction accuracy with trained predictors. Now with trained predictors the two-level predictors with a higher order are better than the predictors with low order. The reason for that fact is again the learning time of the predictors and the ability of predictor with high order to learn complex pattern. Especially in our evaluation both two-level predictors—two-state respectively frequency analysis—with an order of 3, 4, and 5 have very powerful capabilities.

As overall observation we can say that some predictors obtained very high accuracies taking into account that a

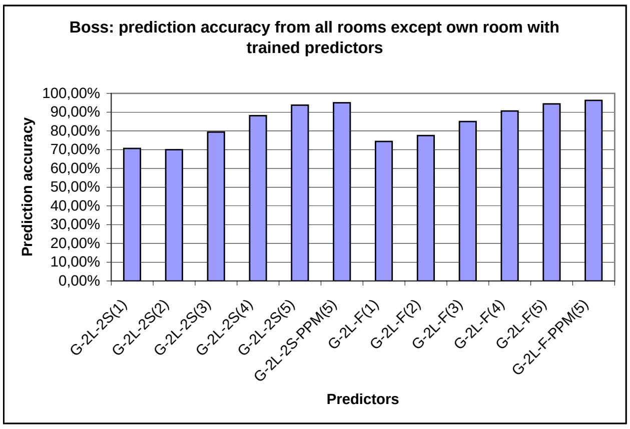

Figure 8. Prediction accuracy of boss' movement using trained predictors

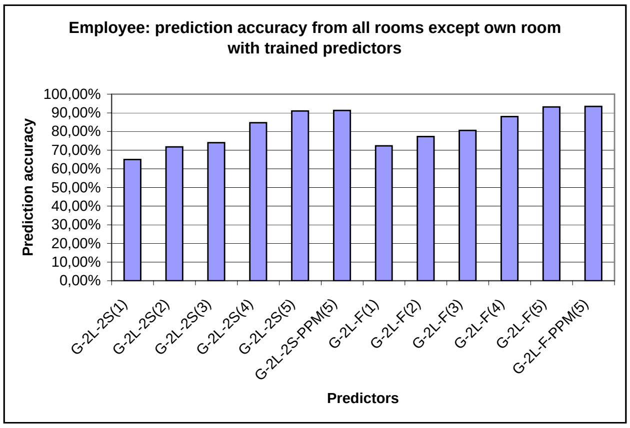

Figure 9. Prediction accuracy of employee's movement using trained predictors

non-saturated PPM predictor is claimed to be an ultimate limit of context predictability [4]. Even our implemented saturated PPM points out this fact.

Table 2. Gain of trained vs. untrained predictors

G-2L-2S	(1)	(2)	(3)	(4)	(5)	PPM(5)
Boss	1.05	1.37	2.12	3.53	6.52	1.48
Employee	1.04	1.23	1.67	2.31	5.35	1.41
G-2L-F	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**	**PPM(5)**
Boss	1.11	1.48	2.31	3.63	6.86	1.54
Employee	1.05	1.24	1.75	2.41	5.31	1.41

Table 2 shows the gain of trained vs. untrained predictors, yielded by dividing the prediction accuracy of trained to the untrained emphasizing the importance of training in particular for the two-levels predictors of higher order.

4. Conclusions and Further Work

This paper analyzed context prediction. In ubiquitous computing environments often relatively simple prediction algorithms are required e.g. due to the PDA's memory, computing, and communication restrictions. This was the main reason for developing predictors based on two-level adaptive branch predictor schemes [4, 14].

The evaluations show that two-level predictors with a low order, especially the predictor with order 1, are very fast in training and retraining, whereas the two-level predictors with a higher order are better suited for complex patterns. The advantages of both types could be combined in a hybrid context predictor. As an example, a two-level two-state predictor with order 1 could be used during the training phase of a two-level predictor with an order of 5 or higher. Such a hybrid predictor might be suitable for real world applications, as nobody wants to wait a long time for a system to adopt itself.

We envision two distinct training processes. First a static training process (pre-training), meaning that the predictors are trained before working in a real environment, for example trained them into a PC and after this putting them into PDAs. Second a dynamic training process, meaning that the predictors are learning during run-time.

To avoid misguidance of persons or systems with wrong predictions, the confidence of the predictions should be taken into account. Meaning that a prediction should only be made, if the prediction reaches a high confidence level. Time is another important point in learning human habits. Therefore the predictors should be enhanced to treat time-dependencies, e.g. time of the day and day of the week.

In the long run we expect predictors to be combined with user's profiles and knowledge about certain human behaviors e.g. from a person's scheduler in order to avoid mispredictions or even over-predictions.

References

[1] D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.

[2] E. Behrends. *Introduction to Marcov Chains*. Vieweg, 1999.

[3] A. Bhattacharya and S. K. Das. LeZi-Update: An Information-Theoretic Framework for Personal Mobility Tracking in PCS Networks. *Wireless Networks*, 8:121–135, 2002.

[4] I.-C. K. Chen, J. T. Coffey, and T. N. Mudge. Analysis of Branch Prediction via Data Compression. In *ASPLOS VII*, pages 128–137, Cambridge, Massachusetts, USA, October 1996.

[5] K. Gurney. *An Introduction to Neural Networks*. Routledge, 2002.

[6] F. V. Jensen. *An Introduction to Bayesian Networks*. UCL Press, 1996.

[7] K. Kaowthumrong, J. Lebsack, and R. Han. Automated Selection of the Active Device in Interactive Multi-Device Smart Spaces. In *Workshop at UbiComp'02: Supporting Spontaneous Interaction in Ubiquitous Computing Settings*, Göteborg, Sweden, 2002.

[8] E. Katsiri. Principles of Context Inference. In *Adjunct Proceedings UbiComp'02*, pages 33–34, Göteborg, Sweden, 2002.

[9] M. C. Mozer. The Neural Network House: An Environment that Adapts to its Inhabitants. In *AAAI Spring Symposium on Intelligent Environments*, pages 110–114, Menlo Park, CA, 1998.

[10] D. J. Patterson, L. Liao, D. Fox, and H. Kautz. Inferrring High-Level Behavior from Low-Level Sensors. In *5th International Conference on Ubiquitous Computing*, pages 73–89, Seattle, WA, USA, 2003.

[11] J. Petzold, F. Bagci, W. Trumler, and T. Ungerer. Context Prediction Based on Branch Prediction Methods. Technical Report 2003-14, Institute of Computer Science, University of Augsburg, Germany, July 2003. http://www.informatik.uni-augsburg.de/skripts/techreports/.

[12] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *IEEE*, 77(2), February 1989.

[13] W. Trumler, F. Bagci, J. Petzold, and T. Ungerer. Smart Doorplate. In *First International Conference on Appliance Design (1AD)*, Bristol, GB, May 2003. Reprinted in Pers Ubiquit Comput (2003) 7: 221-226.

[14] J. Šilc, B. Robič, and T. Ungerer. *Processor Architecture - From Dataflow to Superscalar and Beyond*. Springer-Verlag, Berlin, Heidelberg, New York, 1999.