Global and Local State Context Prediction

Jan Petzold, Faruk Bagci, Wolfgang Trumler, and Theo Ungerer

University of Augsburg
Institute of Computer Science
Eichleitnerstr. 30, 86159 Augsburg, Germany
{Petzold, Bagci, Trumler, Ungerer}@Informatik.Uni-Augsburg.DE

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 focuses on context prediction based on previous behavior patterns.
The proposed prediction algorithms originate in branch prediction techniques of current high-performance microprocessors which are transformed to handle context prediction. We focus on the two-level two-state predictors with local and global first-level histories. Evaluation is performed by simulating the predictors with behavior patterns of people walking through a building as workload. The evaluations show that the proposed context predictors perform well but exhibit differences in training and retraining speed and in their ability to learn complex patterns.

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 If the speculation fails, the failing must be recognized, the speculatively initiated actions withdrawn, and the predictor updated to improve future prediction accuracy.

To predict or anticipate a future situation learning techniques as e.g. Markov Chains [1], Bayesian Networks [4], Neural Networks [3] are obvious candidates. The challenge is to transfer these algorithms to work with context information.

The Adaptive House project [7] 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 Inferences are applied by Katsiri [6] to predict people's movement. Markov Chains are used by Kaowthumrong et al. [5] for active device selection.

The next section shortly describes the proposed context prediction algorithms and section 4 evaluates the predictors. The paper ends with the conclusions. More details on the algorithms proposed in this paper is presented in [8].

2 Context Prediction Algorithms

For our application domain we chose next location prediction instead of general context prediction. We use a floor plan with four rooms as example (see figure 1). 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 allow only to base the predictions on local movements from one room to its neighbor rooms.

1-State Context Predictor. The predictor stores a single prediction state for each neighbor room. For that reason we chose the denotation "1-state context predictor". When leaving a room R the target room T is stored. When the person reenters room R, room T will be predicted as next room.

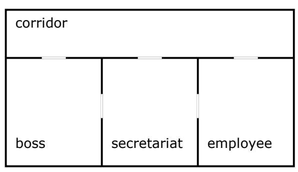

Fig. 1. Floor plan of corridor, boss' office, secretariat, and employee's office

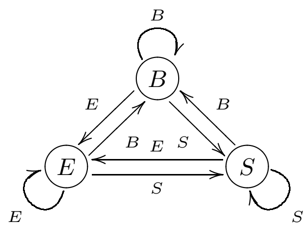

Fig. 2. Prediction graph of 1-state predictor for the corridor

The 1-state predictor works analogically to the one-bit branch predictor, if only two neighbor rooms exist. For three neighbor rooms the one-bit branch predictor has to be extended. For example the corridor has the secretariat S, the boss' office B, and employee's office E as neighbor rooms (see figure 1). Figure 2 shows the prediction graph of the 1-state predictor for the corridor. The states indicate which room will be predicted. If a person goes from the corridor into the boss' office B the predictor will be set to the state B. When the person reenters the corridor, the boss' office B will be predicted as the person's next location. If this prediction proves as correct, the predictor stays in state B. Otherwise, if the person enters another room, e.g. the secretariat S, the predictor changes to the state S and at the next time it will predict the secretariat.

The construction principle can be continued analogically in case of more neighbor rooms. Each room is represented by a node and all nodes are completely connected to each other.

The storage costs of the 1-state predictor are very low. Only the ID of the room that is predicted has to be stored for each room. An advantage of the

1-state predictor is its fast training. A person has to visit a room only once in the past and yet the next room can be predicted. A possible disadvantage is its potential too fast retraining as the following example shows. A person goes always from the corridor into the same room. If she interrupts her habit only once by entering another room, the predictor is already retrained and predicts next time the wrong room. The 1-state predictor is very sensible against one-time deviations from the habit. Here the 2-state predictor constitutes a remedy.

2-State Context Predictor. The 2-state context predictor is a modification of the two-bit branch predictor with saturation counter. The first entry denotes the next room as in the 1-state predictor. 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 "2-state context predictor" stems from the provision of two states for each predicted room.

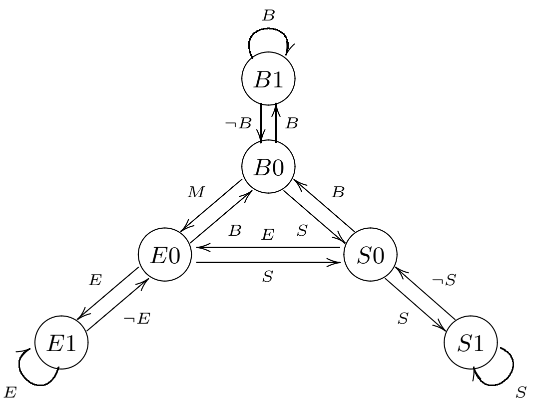

Fig. 3. Prediction graph of 2-state predictor for the corridor

The prediction graph of a room with three neighbor rooms (see figure 1) is shown in figure 3. The denotations of the states consist of the ID of the room and a counter. 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.

If a room has more than three neighbor rooms, the principle can be continued similarly. 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.

The storage costs of the 2-state predictor are still low. Besides the ID of the room that is predicted an additional one-bit counter has to be stored for each room. The computation costs for adapting the states are insignificantly larger than for the 1-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.

Global Two-level Context Predictors. 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 2-state predictor entry, which can be replaced by an arbitrary k-state predictor or a frequency analysis 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), S (secretariat), B (boss' office), and E (employee's office). Furthermore we assume an order of 4. Then there are $4 \cdot 3^3 = 108$ patterns and therefore 108 entries in the pattern history table (PHT). Figure 4 shows an extract of the PHT assuming the room sequence $C\ S\ B\ C\ E\ C\ B\ S\ E\ S\ E\ C\ B\ S\ \underline{E\ C\ B\ S}$. The prediction is that the office of the employee E will be entered next.

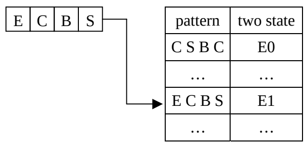

Fig. 4. Two-level predictor with 2-state predictor

The 2-state predictor in the second level can be replaced by a k-state predictor with different k to adjust the retraining speed or by the frequencies of all previous accesses to all neighbor rooms from the current room (frequency analysis).

An advantage of the global two-level predictors is that now complex movement patterns can be predicted. Since each pattern is treated separately, interferences [9] between two patterns cannot appear as is the case in branch prediction.

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 R 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.

Local Two-level Context Predictors. The local two-level context predictors use only neighbor room history and disregard the movement sequences through all rooms. The shift register does not contain the global history, i.e. which rooms were entered before, but the local history, meaning the rooms, which the person visited after the local room.

For each room exists a two-level predictor, i.e. the consecutively entered neighbor rooms form the pattern for each room. The local two-level predictors operate like the global two-level predictors. Thus again different prediction methods (2-state, k-state or frequency analysis) can be applied in the second level.

We look at the floor plan example of figure 1 assuming three neighbor rooms for the corridor. We assume a person moves through the rooms in the sequence $C\ B\ C\ S\ B\ C\ B\ E\ C\ S\ C$. The person is now in the corridor and the shift register used for the selection of the entry in the pattern history table of the corridor contains $B\ S\ B\ S$.

The storage costs are composed as follows. Let r be the order and let n_s be the number of the neighbor rooms of the room s, then a table with $(n_s)^r$ patterns has to be stored for each room s. The same room can occur successively in a pattern, in contrary to the global two-level predictors described above. An advantage over the global two-level predictors is that with a smaller order longer global sequences can be accounted for. Patterns of the kind "after repeated visits of a known room, another room is visited" can be recognized over a long sequence of movements. The disadvantage of this predictor is that the training phase is extremely long. Moreover, global movement patterns cannot always be distinguished. For instance the movement pattern $C\ B\ C\ S\ B\ C\ B\ E\ C\ S\ C$ and $C\ B\ C\ S\ E\ B\ E\ C\ B\ E\ C\ S\ C$ generate the same $B\ S\ B\ S$ pattern of order 4, because only the rooms entered after corridor C are taken into account to form the local pattern. Thus, from the global view, interferences may occur in pattern construction.

The two-level context predictors can be extended using a method motivated by Prediction by Partial Matching (PPM) [2] 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 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.

3 Evaluation

No benchmarks exist so far for movement patterns in ubiquitous systems. Therefore we use synthetic patterns for movements of a person within an office building. We describe simulation results evaluating all context predictors using various movement sequences as workload. Each sequence consists of a movement cycle, which is repeated until the person performed 1,000 room changes. All movement sequences are based on the four room scenario of figure 1. An order of 4 is used for the local as well as the global two-level predictors. The measurements calculate the cumulative error of every predictor, i.e. the mispredictions are added up. A predictor learned a pattern if the gradient of the cumulative error graph is zero. Learning a pattern means that the misprediction rate converges to 0%. The misprediction rate can be calculated from the charts as the cumulative error after 1,000 predictions divided by 1,000. The predictor labels shown in the diagrams are defined in table 1.

Table 1. Abbreviations for the context predictors

L-1L-1S	local one-level 1-state predictor
L-1L-2S	local one-level 2-state predictor
L-2L-2S(4)	local two-level predictor with 2-state predictor and order 4
L-2L-F(4)	local two-level predictor with frequencies and order 4
G-2L-2S(4)	global two-level predictor with 2-state predictor and order 4
G-2L-F(4)	global two-level predictor with frequencies and order 4

These measurements examine the predictor's behavior regarding changing habits (many additional measurements are published in [8]). The change can be that a person always carries out a single special action after a certain sequence of actions. Following cycle is simulated:

Z1 - Z1 - Z1 - Z1 - Z1 - Z1 - Z2 - Z2 - Z2 - Z2 - Z2 (short: 6×Z1 - 5×Z2)

Auxiliary cycles Z1: $C \overset{4\times}{\rightleftharpoons} S \rightarrow C \overset{1\times}{\rightleftharpoons} B$ and Z2: $C \overset{4\times}{\rightleftharpoons} S \rightarrow C \overset{1\times}{\rightleftharpoons} E$

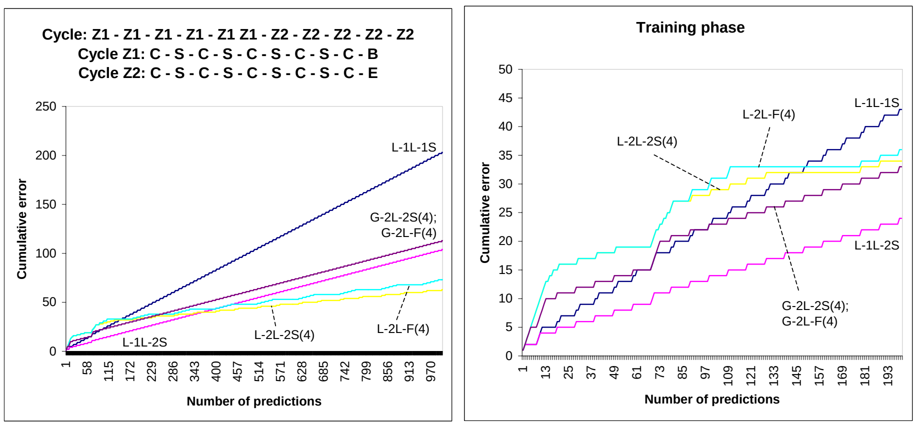

Fig. 5. Measurements

Figure 5 shows that the local two-level two-state predictor performs best followed by the local two-level with frequency analysis. The local one-level two-state and the global two-level predictors perform similar, but are distinguished by the faster training speed of the local one-level two-state predictor.

The local two-level two-state predictor performs better than the predictor with frequency analysis. The local two-level two-state predictor makes four errors per cycle, while the predictor with frequency analysis makes five errors. In the training phase both behave equivalent, but a noticeable difference can be spotted at the first retraining phase.

4 Conclusion

This paper analyzed the suitability of branch prediction techniques known from the area of processor architecture for the context prediction. We transferred branch prediction techniques to context prediction using a person's movement patterns in a building. The evaluation of the implemented predictors used synthetic movement sequences, because of the lack of real movement pattern. The usage of various synthetic pattern lead to a good differentiation between the predictors, which are summarized as follows:

2-State Context Predictor. The training phase of the 2-state predictor works fast. A prediction can already be performed after one past transition. Also the retraining is fast, we need only two misprediction to retrain. The 2-state predictor is suitable as a "warming-up predictor" for complex predictors.

The local one-level predictors, i.e. the 1-, 2, and k-state predictors, are unable to learn complex non-constant patterns. The state number k determines the retraining speed.

Local Two-level Context Predictors. These predictors cannot learn the pattern with an order of 4, because of the 5 transitions from the corridor to the same room. The predictor can learn a pattern only if the local pattern exhibits a transition length that is less or equal than the order.

A disadvantage of the 2-state predictors in the second level is the potential oscillation between two weak states. The predictor with frequency analysis prevents this behavior. But on the other hand this kind of predictors needs long retraining processes. Considering only two possible neighbor rooms, the predictor with frequency analysis is the limit for k-state predictors $k \to \infty$.

The training phase for the local two-level predictors takes the longest time compared to all other regarded predictors, as each room must be left at least as often as the order to make a prediction. Retraining may concern two cases: First, if a new pattern is used, a room must be left as often as the order specifies. Second, retraining needed within a complex pattern depends on the second level. Here, the 2-state predictor performs well, in contrast to the predictor with frequency analysis.

Global Two-level Context Predictors. In contrast to the local two-level predictors, the global two-level predictors don't learn the pattern in all measurement cases, as the following room must not change for a global pattern to be learned. The patterns in the other figures can be learned with an order number of 5 or more.

The differences between the 2-state predictor and the frequency analysis in the second level is equivalent to those of the local two-level predictors.

In the training phase the global two-level predictors perform always better than the local once, because global patterns are learned faster. However, overall performance of the global two-level predictors is worse than the performance of the local one-level predictors. An analogical behavior is reflected at the retraining phase in the case of new patterns. In all other cases the relearning of a room, following a certain pattern, depends on the second level, equivalent to the local two-level predictors.

The advantages of several predictor types could be combined in a hybrid context predictor. As an example, a 2-state predictor could be used during the training phase of a two-level predictor and its prediction is substituted by the two-level predictor afterwards. 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. 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 and suppressed otherwise.

Our future work concerns construction of new predictors and evaluation of these and of the described predictors with real movement sequences. A person tracking system, currently build up at the University of Augsburg, will generate such real movement patterns. Time is another important point in learning human habits. Therefore the predictors shall be enhanced to be time-dependent.

References

1. Ehrhard Behrends. *Introduction to Marcov Chains*. Vieweg, 1999.
2. I-Cheng K. Chen, John T. Coffey, and Trevor N. Mudge. Analysis of Branch Prediction via Data Compression. In *ASPLOS VII*, pages 128–137, Cambridge, Massachusetts, USA, October 1996.
3. Kevin Gurney. *An Introduction to Neural Networks*. Routledge, 2002.
4. Finn V. Jensen. *An Introduction to Bayesian Networks*. UCL Press, 1996.
5. Khomkrit Kaowthumrong, John Lebsack, and Richard 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.
6. Eleftheria Katsiri. Principles of Context Inference. In *Adjunct Proceedings UbiComp'02*, Göteborg, Sweden, 2002.
7. Michael 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.
8. Jan Petzold, Faruk Bagci, Wolfgang Trumler, and Theo 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/.
9. J. Šilc, B. Robič, and T. Ungerer. *Processor Architecture - From Dataflow to Superscalar and Beyond*. Springer-Verlag, Berlin, Heidelberg, New York, 1999.