Confidence Estimation of the State Predictor Method

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

Institute of Computer Science
University of Augsburg
Eichleitnerstr. 30, 86159 Augsburg, Germany
{petzold, bagci, trumler, ungerer}@informatik.uni-augsburg.de

Abstract. Pervasive resp. 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. By means of context prediction systems get ready for future human activities and can act proactively.

Predictions, however, are never 100% correct. In case of unreliable prediction results it is sometimes better to make no prediction instead of a wrong prediction. In this paper we propose three confidence estimation methods and apply them to our State Predictor Method. The confidence of a prediction is computed dynamically and predictions may only be done if the confidence exceeds a given barrier. Our evaluations are based on the Augsburg Indoor Location Tracking Benchmarks and show that the prediction accuracy with confidence estimation may rise by the factor 1.95 over the prediction method without confidence estimation. With confidence estimation a prediction accuracy is reached up to 90%.

1 Introduction

Pervasive resp. ubiquitous systems aim at more convenience for daily activities by relieving humans from monotonic routines. Here context prediction plays a decisive role. Because of the sometimes unreliable results of predictions it is better to make no prediction instead of a wrong prediction. Humans may be frustrated by too many wrong predictions and won't believe in further predictions even when the prediction accuracy improves over time. Therefore confidence estimation of context prediction methods is necessary. This paper proposes confidence estimation of the State Predictor Method [7,8,9,10], which is used for next location prediction. Three confidence estimation methods were developed for the State Predictor Method.

The proposed confidence estimation techniques can also be transferred to other prediction methods like Markov, Neural Network, or Bayesian Predictors. In the next section the State Predictor Method is introduced followed by section 3 that defines three methods of confidence estimation. Section 4 gives the evaluation results, section 5 outlines the related work, and the paper ends by a conclusion.

P. Markopoulos et al. (Eds.): EUSAI 2004, LNCS 3295, pp. 375–386, 2004.

2 The State Predictor Method

The State Predictor Method for next context prediction is motivated by branch prediction techniques of current high performance microprocessors. We developed various types of predictors, 1-state vs. 2-state, 1-level vs. 2-level, and global vs. local [7]. We describe two examples.

First, the 1-level 2-state predictor, or shorter the 2-state predictor, is based on the 2-bit branch predictor. Analogically to the branch predictor the 2-state predictor holds two states for every possible next context (a weak and a strong state).

Figure 1 shows the state diagram of a 2-state predictor for a person who is currently in context X with the three possible next contexts A, B, and C. If the person is the first time in context X and changes e.g. to context C the predictor sets $C0$ as initial state. Next time the person is in context X context C will be predicted as next context. If the prediction proves as correct the predictor switches into the strong state $C1$ predicting the context C as next context again. As long as the prediction is correct the predictor stays in the strong state $C1$. If the person changes from context X in another context unequal C (e.g. A), the predictor switches in the weak state $C0$ predicting still context C. If the predictor is in a weak state (e.g. $C0$) and misses again, then the predictor is set to the weak state of the new context which will be predicted next time. The 2-state predictor can be dynamically extended by new following contexts during run-time.

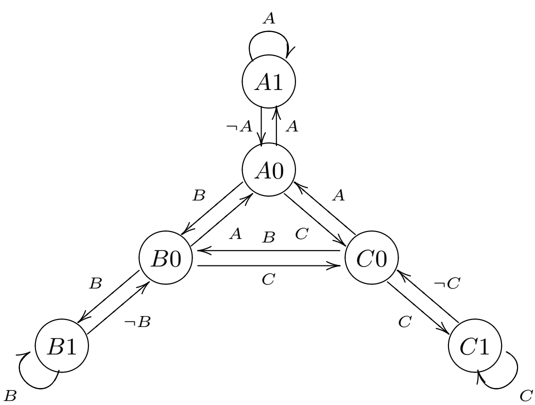

Fig. 1. 2-state predictor with three contexts

The second example is the global 2-level 2-state predictor which consists of a shift register and a pattern history table (see figure 2). A shift register of length n stores the pattern of the last n occurred contexts. By this pattern the shift register selects a 2-state predictor in the pattern history table which holds for every pattern a 2-state predictor. These 2-state predictors are used for the prediction. The length of the shift register is called the order of the predictor.

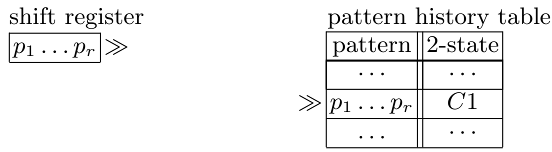

Fig. 2. Global 2-level 2-state predictor

The 2-level predictors can be extended using the Prediction by Partial Matching (PPM) [10] or the Simple Prediction by Partial Matching (SPPM). PPM means, a maximum order m is applied in the first level instead of the fixed order. Then, starting with this maximum order m, a pattern is searched according to the last m contexts. If no pattern of the length m is found, the pattern of the length $m-1$ is looked for, i.e. the last $m-1$ contexts. This process can be accomplished until the order 1 is reached. SPPM means, if the predictor with the highest order can't deliver a result, the predictor with order 1 is requested to do the prediction.

3 Confidence Estimation Methods

3.1 Strong State

The *Strong State* confidence estimation method is an extension of the 2-state predictor and be applied to the 2-state predictor itself and the 2-level predictor with 2-state predictor in the second level.

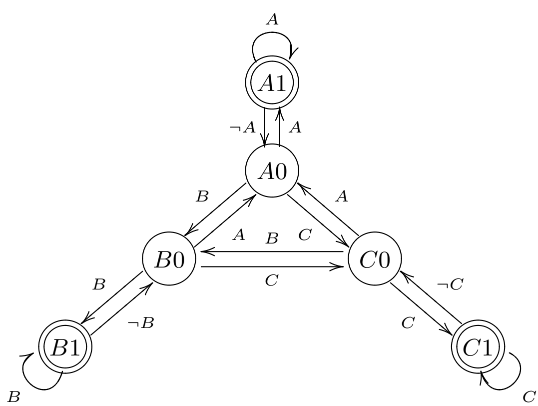

Fig. 3. *Strong State* - Prediction only in the strong states

The idea of the *Strong State* method is simply that the predictor supplies the prediction result only if the prediction is in a *strong state*. The difference between *weak state* (no prediction provided because of low confidence) and *strong state* (the context transition has been done at least two consecutive times) acts as barrier for a good prediction. Figure 3 shows the 2-state predictor for the three contexts A, B, and C. The prediction result of the appropriate context will be supplied only in the strong states which are double framed in the figure.

As a generalization this method can be extended for the n-state predictor that uses n ($n > 2$) states instead of the two states of the 2-state predictor. Therefore a barrier k with $1 \leq k \leq n$ must be chosen to separate the weak and the strong states. For the 2-state predictor ($n = 2$) it is $k = 1$. Let $1 \leq s \leq n$ be the value of the current state of the 2-state predictor, then we can define:

$$s \geq k : \text{supply prediction result} \tag{1}$$
$$s < k : \text{detain prediction result} \tag{2}$$

No additional memory costs arise from the *Strong State* method.

3.2 Threshold

The *Threshold* confidence estimation method is independent of the used prediction algorithm. This method compares the accuracy of the previous predictions with a given threshold. If the prediction accuracy is greater than the threshold the prediction is assumed confident and the prediction result will be supplied. For the 2-level predictors we consider all 2-state predictors for itself. That means for all patterns we investigate separately the prediction accuracy of the appropriate 2-state predictor. The prediction accuracy of a 2-state predictor is calculated from the numbers of correct and incorrect predictions of this predictor, more precisely, the prediction accuracy is the fraction of the correct predictions and the number of all predictions. Let c be the number of correct predictions, i the number of incorrect predictions, and α the threshold, then the method can be described as follows:

$$\frac{c}{c+i} \geq \alpha : \text{supply prediction result} \tag{3}$$
$$\frac{c}{c+i} < \alpha : \text{detain prediction result} \tag{4}$$

The threshold is a value between 0 (only mispredictions) and 1 (100% correct predictions in the past) and should be chosen well above 0.5 (50% correct predictions).

For the global 2-level predictors the pattern history table must be extended by two values, the number of correct predictions c and the number of incorrect predictions i (see figure 4).

The global 2-level 2-state predictor starts with an empty pattern history table. After the first occurrence of a pattern the predictor can't deliver a prediction result. If the context c follows this pattern the 2-state predictor will be initialized with the weak state of the context c. Furthermore 0 will be set as initial values for the entries of the number of correct predictions and the number of incorrect predictions. After the second occurrence of this pattern a first prediction is done but with unknown reliability. The system cannot estimate the confidence, since shift register

$\boxed{p_1 \ldots p_r} \gg$

pattern history table

	pattern	2-state	c	i
	...	...	...	...
$\gg$	$p_1 \ldots p_r$	$C1$	x	y
	...	...	...	...

Fig. 4. Global 2-level 2-state predictor with threshold

there are no prediction results yet. Also the values of the correct and incorrect predictions cannot be calculated by the formula for estimating the confidence, because the denominator is equal 0. If the prediction is proved as correct the value of the number of correct predictions will be increased to 1. Otherwise the number of incorrect predictions will be incremented. Furthermore the 2-state predictor will be adapted accordingly. After the next occurrence of the pattern the values can be calculated by the formula and the result of the formula can be compared with the threshold.

A disadvantage of the *Threshold* method is the "getting stuck" over the threshold. That means if the previous predictions were all correct the prediction accuracy is much greater than the threshold. If now a behavior change to another context occurs and incorrect predictions follow, it takes a long time until the prediction accuracy is less than the threshold. Thus the unconfident predictor will be considered as confident. A remedy is the method with confidence counter.

3.3 Confidence Counter

The *Confidence Counter* method is independent of the used prediction algorithm, too. This method estimates the prediction accuracy with a saturation counter. The counter consists of $n + 1$ states (see figure 5).

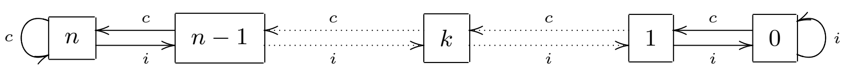

Fig. 5. *Confidence Counter* - state graph

The initial state can be chosen optionally. Let s be the current state of the confidence counter. If a prediction result is proved as correct (c) the counter will increase, that means the state graph changes from state s into the state $s + 1$. If $s = n$ the counter keeps the state s. Otherwise if the prediction is incorrect (i) the counter switches into the state $s - 1$. If $s = 0$ the counter keeps the state s. Furthermore there is a state k with $0 \leq k \leq n + 1$, which acts as a barrier value: If s is greater or equal k the predictor is assumed as "confident", otherwise the predictor is unconfident and the prediction result will not be supplied. The method can be described as follows.

$$s \geq k : \text{supply prediction result} \tag{5}$$

$$s < k : \text{detain prediction result} \tag{6}$$

The prediction accuracy will be considered separately for every 2-state predictor. The pattern history table must be extended by one column which stores the current state of the confidence counter *cc* (see figure 6). Memory costs are between the strong state and the threshold methods.

shift register

$\boxed{p_1 \dots p_r} \gg$

pattern history table

	pattern	2-state	cc
	$\dots$	$\dots$	$\dots$
$\gg$	$p_1 \dots p_r$	$C1$	k
	$\dots$	$\dots$	$\dots$

Fig. 6. Global 2-level 2-state predictor with confidence counter

In the following the initial phase for the example of the global 2-level 2-state predictor is explained. Initially the pattern history table is empty. After the first occurrence of a pattern the algorithm cannot deliver a prediction result. If the context c follows, the 2-state predictor of this pattern will be initialized. Furthermore the confidence counter of this pattern sets a value between 0 and n as initial value, for example the value k. After the second occurrence of the pattern the confidence of the 2-state predictor can be estimated by the initial state of the confidence counter. If k was chosen as the initial state the prediction result will be supplied, since k classifies the predictor as confident. Otherwise, if the current state of the confidence counter is less than k, the prediction result will be detained.

4 Evaluation

For our application, the *Smart Doorplates* [12], we choose next location prediction instead of general context prediction. The *Smart Doorplates* direct visitors to the current location of an office owner based on a location tracking system and predict the next location of the office owner if absent.

The evaluation was performed with the *Augsburg Indoor Location Tracking Benchmarks* [6]. These benchmarks consist of movement sequences of four test persons (two of these are selected in the following) in a university office building reported separately during the summer term and the fall term 2003. In the evaluation we don't consider the corridor, because a person can leave a room only to the corridor. Furthermore we investigate only the global 2-level 2-state predictors with order 1 to 5, the PPM predictor, and the SPPM predictor. All considered predictors were evaluated for every test person with the corresponding summer benchmarks followed by the fall benchmarks. The abbreviations in the figures and tables stand for: G - global, 2L - 2-level, 2S - 2-state. The order and the maximum order respectively is given in parentheses.

The training phase of the State Predictor Method depends on the order of the predictor. As some patterns might never occur for predictors with high order, the training phase of the 2-state predictors in the second level never starts.

Therefore we consider the training for every pattern separately. The training phase of the 2-state predictor for a pattern includes only the first occurrence of the pattern. After the second occurrence of the pattern the training phase of the 2-state predictor ends.

Next we introduce some definitions which will be needed for explanations:

- **demand** — number of the predictions which are requested by the user. In our evaluation this number corresponds with the number of rooms a test person has entered during the measurements.
- **supply** — number of prediction results which are delivered from the system. This corresponds with the number of confident predictions which can be calculated from the requested predictions minus the predictions of the training phase of every 2-state predictor and the unconfident predictions.
- **quality** — fraction of the number of the correct predictions and the supply.

$$\text{quality} = \frac{\#\ \text{correct predictions}}{\text{supply}}$$

- **quantity** — ratio of supply and demand

$$\text{quantity} = \frac{\text{supply}}{\text{demand}}$$

- **gain** — the factor which gives the improvement of the quality with confidence estimation opposite the quality without confidence estimation.

$$\text{gain} = \frac{\text{quality with confidence estimation}}{\text{quality without confidence estimation}}$$

4.1 Strong State

Figures 7 and 8 and tables 1 and 2 show the results of the measurements with the *Strong State* method. The figures show the quality and the quantity of the prediction method with and without confidence estimation. The graphs with confidence estimation are denoted by SS in brackets. Quality, quantity, and gain were measured for the two test persons.

Table 1. Gain - Person A

G-2L-2S	(1)	(2)	(3)	(4)	(5)
Gain	1,46	1,48	1,47	1,50	1,39

G-2L-2S	-PPM(5)	-SPPM(5)	
Gain	1,64	1,53	

Table 2. Gain - Person D

G-2L-2S	(1)	(2)	(3)	(4)	(5)
Gain	1,74	1,53	1,39	1,42	1,28

G-2L-2S	-PPM(5)	-SPPM(5)	
Gain	1,40	1,62	

The gain is always greater than 1, that means the quality improves with confidence estimation for all predictors. The greatest gain (1.74) is reached by the global 2-level 2-state predictor with order 1 of test person D (see figure 8 and

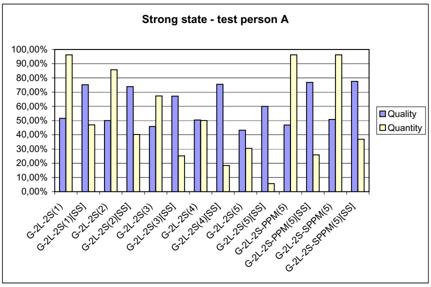

Fig. 7. Quality and Quantity - Person A

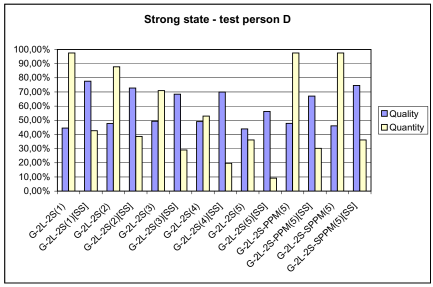

Fig. 8. Quality and Quantity - Person D

table 2). The reason is that the predictor very often changes between various weak states.

If we consider the quantity we can see that the value falls for all measurements. In the case of the global 2-level 2-state predictor with order 5 the quantity falls even below 10%. That means the algorithm delivers the prediction result only in one out of ten requests.

4.2 Threshold

Figures 9 to 14 show the measurement results of the *Threshold* method. For the two test persons the quality, the quantity, and the gain were measured. The threshold varies between 0.2 and 0.8.

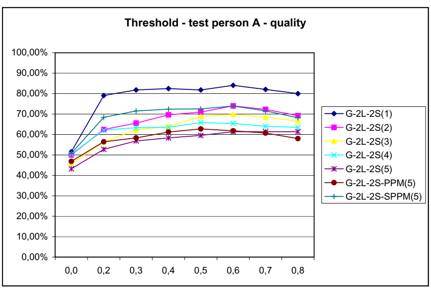

Fig. 9. Quality - Person A

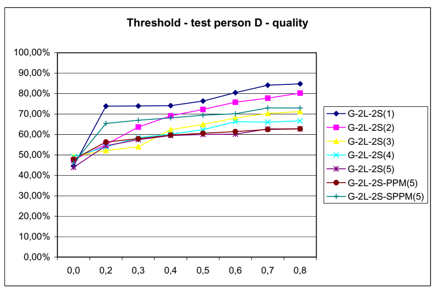

Fig. 10. Quality - Person D

Normally the quality can be increased by increasing the threshold. The figures of test person A show that the quality decreases by a threshold greater than or equal 0.7. The reason is that the 2-state predictors reach a prediction accuracy between 60 and 70 percent. The best quality with threshold was reached by the predictor with order 1 for all test persons.

Also here the quantity decreases by increasing the threshold. The best quantity with a high threshold was reached by the global 2-level 2-state predictor with Prediction by Partial Matching. The gain shows the same behavior as the

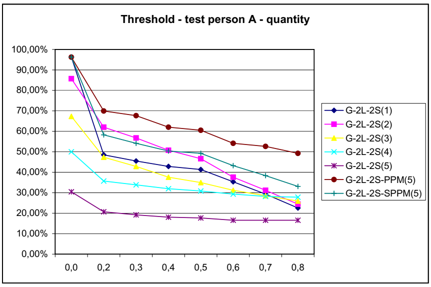

Fig. 11. Quantity - Person A

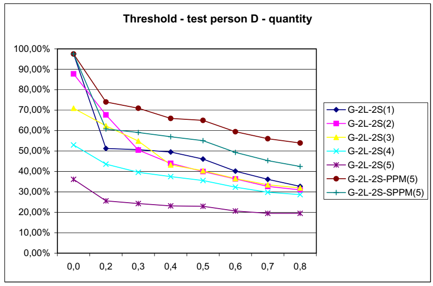

Fig. 12. Quantity - Person D

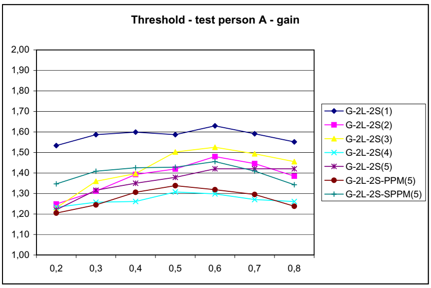

Fig. 13. Gain - Person A

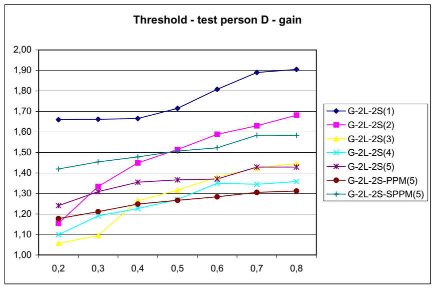

Fig. 14. Gain - Person D

quality. If the threshold increases the gain increases too. The measurements of test person A show that the gain decreases if the threshold is greater than or equal 0.7. The best gain of 1.9 was reached by the predictor G-2L-2S(1) with a threshold value of 0.9 for test person D (see figure 14).

4.3 Confidence Counter

Figures 15 to 20 show the result of the measurements of the *Confidence Counter* method. For evaluation a confidence counter with $n = 3$ was chosen, which can be described by two bits as follows:

$$\begin{tikzcd} \boxed{11} \arrow[loop left, "c"] \arrow[r, shift right, "i"'] & \boxed{10} \arrow[l, shift right, "c"'] \arrow[r, shift right, "i"'] & \boxed{01} \arrow[l, shift right, "c"'] \arrow[r, shift right, "i"'] & \boxed{00} \arrow[l, shift right, "c"'] \arrow[loop right, "i"] \end{tikzcd}$$

We set the state 10 as the initial state. We performed the measurements with $k \in \{00, 01, 10, 11\}$, whereas $k = 00$ means the predictor works analog to the correspond predictor without confidence estimation. Again the quality, the quantity, and the gain were measured.

As expected the quality increases in all cases if k increases. Accordingly the quantity decreases with a higher k. The gain is analog to the quality. The best quality of 90% (see figure 15) and the best gain of 1.95 (see figure 20) was reached by the predictor G-2L-2S(1) with $k = 11$. For all test persons the best quantity

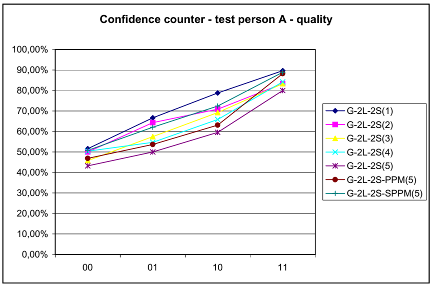

Fig. 15. Quality - Person A

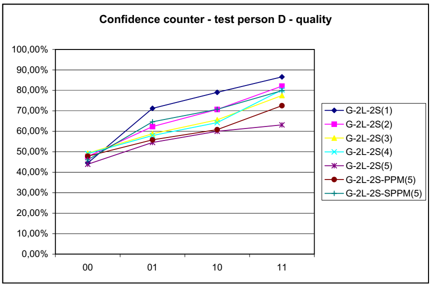

Fig. 16. Quality - Person D

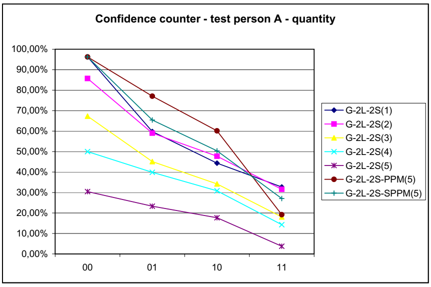

Fig. 17. Quantity - Person A

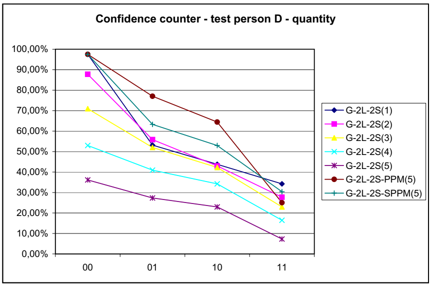

Fig. 18. Quantity - Person D

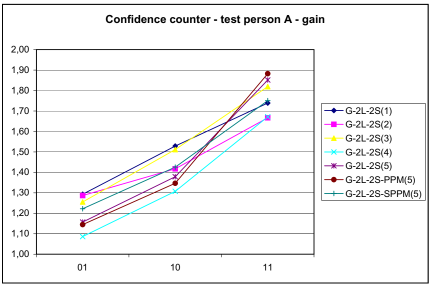

Fig. 19. Gain - Person A

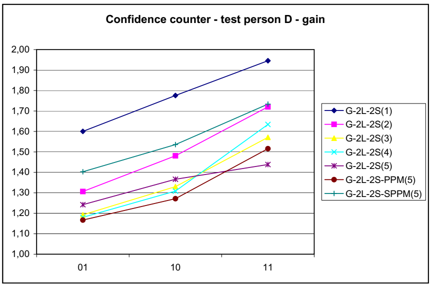

Fig. 20. Gain - Person D

were reached by the predictors with prediction by partial matching followed by the predictors with simple prediction by partial matching. The predictors with order 5 reached the lowest quantity in all cases.

5 Related Work

A number of context prediction methods are used in ubiquitous computing [1,4, 5,14,15]. None of them regards the confidence of the predictions.

Because the State Predictor Method was motivated by some branch prediction methods of current high performance microprocessors, we are influenced by the confidence estimation methods in this research area. Smith [11] proposed a confidence estimator based on the saturation counter method. This approach corresponds to the *Confidence Counter* method. Grunwald et al. [2] investigated additionally to the given techniques a method which counts the correct predictions of every branch and compare this number with a threshold. If this number is greater than the threshold the branch has a high confidence, otherwise a low confidence. This method corresponds to the *Threshold* confidence estimation method.

Jacobson et al. [3] use a table of shift registers as confidence estimator, the so called n-bit *correct/incorrect registers* additionally to the base branch predictor. For every correct prediction a 0 and for a incorrect prediction a 1 will be entered into the register. A so-called reduction function classifies the confidence as high or low. An example of the function is the number of 1's compared to a threshold. If the number of 1's is greater than the threshold, the confidence is classified as low, otherwise as high. Furthermore Jacobson et al. propose a two level method which works with two of the introduced tables.

Tyson et al. [13] investigated the distribution of the incorrect predictions. They observed that a low number of branches include the main part of incorrect predictions. They proposed a confidence estimator which assigns a high confidence to determined number of branches, and a low confidence to the other.

The methods of Jacobson et al. and Tyson et al. may influence further confidence estimation methods in context prediction.

6 Conclusion

This paper introduced confidence estimation into the State Predictor Method. The motivation was that in many cases it is better to make no prediction instead of a wrong prediction to avoid frustrations by too many mispredictions. The confidence estimation methods may automatically suppress predictions in a first training state of the predictor where mispredictions may occur often. We proposed three methods for confidence estimation.

For all three methods the prediction accuracy increases with confidence estimation. The best gain of 1.95 was reached for the *Confidence Counter* method. In this case the prediction accuracy without confidence estimation was 44.5%, and the prediction accuracy with confidence estimation reached 86.6%. But the quantity decreases with the estimation, that means the system delivers prediction results less often. The best result was reached again with the *Confidence Counter* method. Here a prediction accuracy of about 90% is reached. But in this case the quantity is about 33%.

We plan to combine confidence estimation with other predition methods, in particular Markov, Neural Network, and Bayesian Predictors.

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. D. Grunwald, A. Klauser, S. Manne, and A. Pleszkun. Confidence Estimation for Speculation Control. In *Proceedings of the 29th Annual International Symposium on Computer Architecture*, pages 122–131, Barcelona, Spain, June-July 1998.
3. E. Jacobsen, E. Rotenberg, and J. E. Smith. Assigning Confidence to Conditional Branch Predictions. In *Proceedings of the 29th Annual International Symposium on Microarchitecture*, pages 142–152, Paris, France, December 1996.
4. 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, USA, 1998.
5. D. J. Patterson, L. Liao, D. Fox, and H. Kautz. Inferring High-Level Behavior from Low-Level Sensors. In *5th International Conference on Ubiquitous Computing*, pages 73–89, Seattle, WA, USA, 2003.
6. J. Petzold. Augsburg Indoor Location Tracking Benchmarks. Technical Report 2004-9, Institute of Computer Science, University of Augsburg, Germany, February 2004. http://www.informatik.uni-augsburg.de/skripts/techreports/.
7. 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/.
8. J. Petzold, F. Bagci, W. Trumler, and T. Ungerer. Global and Local Context Prediction. In *Artificial Intelligence in Mobile Systems 2003 (AIMS 2003)*, Seattle, WA, USA, October 2003.
9. J. Petzold, F. Bagci, W. Trumler, and T. Ungerer. The State Predictor Method for Context Prediction. In *Adjunct Proceedings Fifth International Conference on Ubiquitous Computing*, Seattle, WA, USA, October 2003.
10. J. Petzold, F. Bagci, W. Trumler, T. Ungerer, and L. Vintan. Global State Context Prediction Techniques Applied to a Smart Office Building. In *The Communication Networks and Distributed Systems Modeling and Simulation Conference*, San Diego, CA, USA, January 2004.
11. J. E. Smith. A Study of Branch Prediction Strategies. In *Proceedings of the 8th Annual Symposium on Computer Architecture*, pages 135–147, Minneapolis, MI, USA, May 1981.
12. 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.
13. G. Tyson, K. Lick, and M. Farrens. Limited Dual Path Execution. Technical Report CSE-TR 346-97, University of Michigan, 1997.
14. K. van Laerhoven, K. A. Aidoo, and S. Lowette. Real-time Analysis of Data from Many Sensors with Neural Networks. In *Proceedings of the International Symposium on Wearable Computers (ISWC 2001)*, Zurich, Switzerland, October 2001.
15. K. van Laerhoven and O. Cakmakci. What Shall We Teach Our Pants? In *Proceedings of the 4th International Symposium on Wearable Computers (ISWC 2000)*, pages 77–83, Atlanta, GA, USA, 2000.