Algorithms for context prediction in Ubiquitous Systems

The state prediction approach

Stephan Sigg

Institute of Distributed and Ubiquitous Systems
Technische Universität Braunschweig

December 9, 2008

Overview and Structure

- Introduction to context aware computing
- Basics of probability theory
- Algorithms
  - Simple prediction approaches: ONISI and IPAM
  - Markov prediction approaches
  - The State predictor
  - Alignment prediction
  - Prediction with self organising maps
  - Stochastic prediction approaches: ARMA and Kalman filter
  - Alternative prediction approahces
    - Dempster shafer
    - Evolutionary algorithms
    - Neural networks
    - Simulated annealing

Overview and Structure

- Introduction to context aware computing
- Basics of probability theory
- **Algorithms**
  - Simple prediction approaches: ONISI and IPAM
  - Markov prediction approaches
  - **The State predictor**
  - Alignment prediction
  - Prediction with self organising maps
  - Stochastic prediction approaches: ARMA and Kalman filter
  - Alternative prediction approahces
    - Dempster shafer
    - Evolutionary algorithms
    - Neural networks
    - Simulated annealing

Outline

The state predictor method

1. Introduction and basic techniques
2. State predictors
3. Evaluation
4. Estimation of Reliability
5. Hybrid predictors
6. Properties of the state predictor method

Introduction

Historical remarks

- State predictor: Developed 2003-2005
- University of Augsburg: Jan Petzold[1]
- Origin: Branch prediction in microprocessors

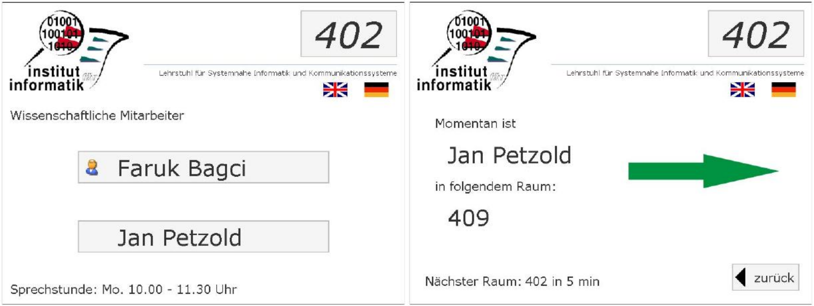
[1] Jan Petzold, *Zustandsprädiktoren zur Kontextvorhersage in ubiquitären Systemen*, PhD-thesis, 2005.

Introduction
Scenario

- Smart Doorplates
  - Display current work situation
    - Telephone interview
    - Meeting
    - Working at desk
  - Display relevant invormation when person is absent
    - Current location
    - Receive/forward messages
    - Prediction of return time
    - Prediction of next location

Introduction
Scenario

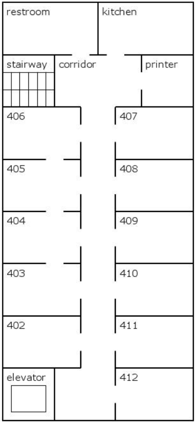
- Smart Doorplates



Introduction

Scenario

- Smart Doorplates
  - Generation of measurements

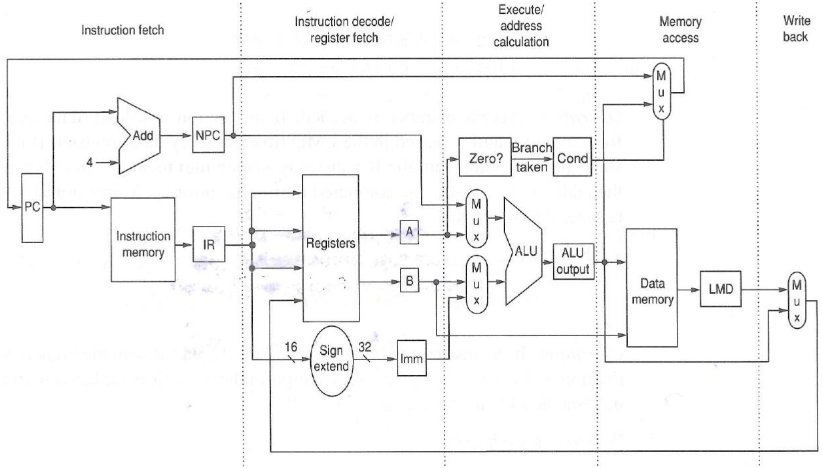


Introduction

Historical remarks

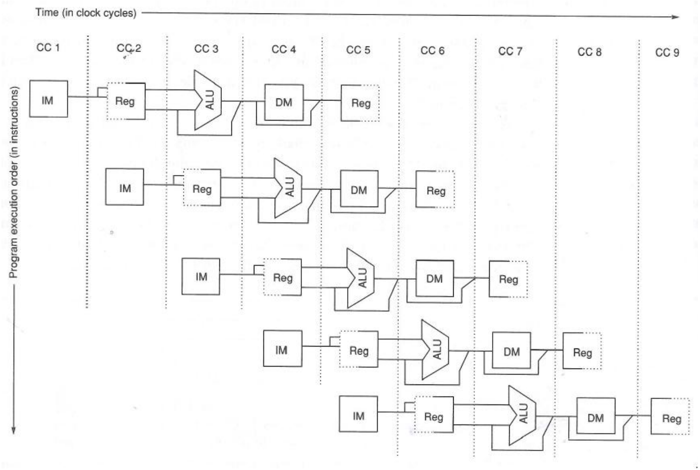
- Branch prediction
  - Loops in program code constitute branches
  - In a pipelined architecture, program execution cycles are interleaved
  - Branch target is only known after ALU.
  - At this time, other (possibly wrong) instructions are already loaded into the pipeline
  - Pipeline flush expensive

- Example: The DLX-Architecture



Introduction
Historical remarks

- Example: The DLX-Architecture



Introduction
Historical remarks

- Example: The DLX-Architecture

Branch instruction	IF	ID	EX	MEM	WB					
Branch successor		IF	*stall*	*stall*	IF	ID	EX	MEM	WB	
Branch successor + 1						IF	ID	EX	MEM	WB
Branch successor + 2							IF	ID	EX	MEM
Branch successor + 3								IF	ID	EX
Branch successor + 4									IF	ID
Branch successor + 5										IF

Introduction
Speedup due to pipelining

- Calculate the speedup achieved due to pipelining. Assume: Clock Cycle of 10 ns

	Taktzyklen (No pipelining)	Taktzyklen (Pipelining)	Occurence probability
Clock cycle	10ns	10ns	
ALU operations	4	4	40%
Branches	4	4	20%
Memory operations	5	5	40%
Pipelining overhead	0ns	1ns	

Introduction

Speedup due to pipelining

$$
\begin{aligned}
\text{Average instruction execution time} &= \text{Clock cycle} \cdot \text{Average CPI} && (1)\\
&= 10ns \cdot ((40\% + 20\%) \cdot 4 + 40\% \cdot 5)\\
&= 10ns \cdot 4.4\\
&= 44ns
\end{aligned}
$$

Pipelined implementation: Clock must be largest time for any stage in the pipeline (10ns) plus overhead: $10 + 1 = 11ns$. This is the average instruction execution time. Speedup from pipelining:

$$
\begin{aligned}
\text{Speedup from pipelining} &= \frac{\text{Average instr. time unpipelined}}{\text{Average instr. time pipelined}} && (2)\\
&= \frac{44ns}{11ns}\\
&= 4 \text{ times}
\end{aligned}
$$

Introduction

Historical remarks

- Branch prediction important to reduce program execution time
- Pipeline flush/stall expensive
- Important therefore: Accuracte prediction

Introduction

Historical remarks

- Simple but effective branch prediction schemes:
  - Always branch
  - Sustain last decision

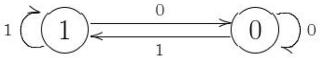
Basic techniques

Branch prediction – dynamic branch prediction

- Dynamic branch prediction schemes
  - Branch decision based on recent behaviour
  - Behaviour stored in branching-tables
  - Tables updated in case of erroneous predictions
- Various implementations
  - Two-bit predictor
  - Two stage adaptive predictor
  - Hybrid predictor

Basic techniques

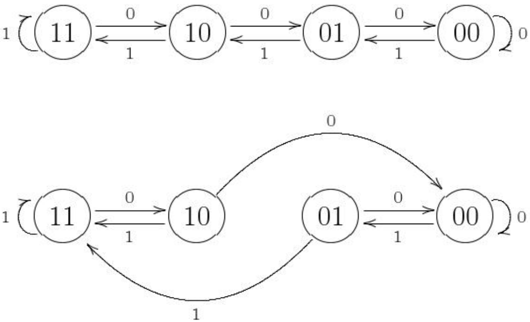
Branch prediction – One- and Two bit predictors



- One-bit predictor
  - Most simple dynamic branch prediction technique
  - Repeat last branching command
  - Every branch inside an iterated loop is correctly predicted
  - In nested loops: First and last branch incorrectly predicted
    - Two bit predictors more accurate in this case

Basic techniques

Branch prediction – One- and Two bit predictors



- Two-bit predictor

Basic techniques

Branch prediction – One- and Two bit predictors

- Two-bit predictor
  - Two wrong predictions in turn required to change prediction behaviour
  - In nested loops, only one wrong prediction: After the last iteration
- Two implementations of Two-bit predictors:
  - Saturation counter
  - Hysteresis counter
- Extension to *n*-bit predictors possible but nearly no improvement in predictoin accuracy

Basic techniques

Branch prediction – Correlation predictors

- Two bit predictors only consider the branch itself
- Intercorrelation between distinct branches are not considered
- However, intercorrelations do matter [2]

[2] Shien-Tai Pan, Kimming so and Joseph T. Rahmeh, *Improving the acuracy of dynamic branch prediction using branch correlation*, In: Proceedings of the fifth international conference on Architectural support for programming languages and operating systems (ASPLOS V), pp 76-84, 1992.

Basic techniques

Branch prediction – Two-stage adaptive predictors

- Two dimensional prediction tables[3]
  - First table selects prediction bits of second table
  - First table: Branch history (Shift register)
  - Second table: Pattern history

[3] Tse-Yu Yeh and Yale N. Patt, *Alternative implementation of two-level adaptive branch prediction*, In: Proceedings of the 19th annual symposium on computer architecture (ISCA-19), pp 124-134, 1992.

Basic techniques

Branch prediction – Two-stage adaptive predictors

- Classes of two-stage adaptive predictors[4]
  - First letter: [G,P,S] – Mechanism in first table
  - Last letter: [g,p,s] – Mechanism in second table
- BHR = Branch Histroy Register of k Bits length
- PHT = Pattern History Table

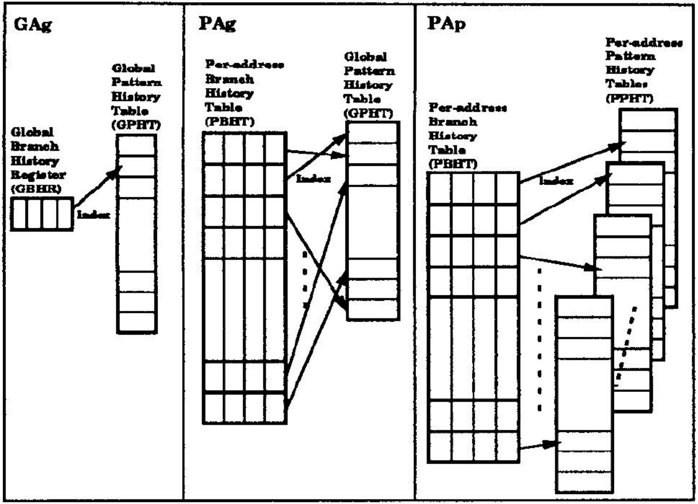
	global PHT	per-set PHTs	per-address PHTs
global BHR	GAg	GAs	GAp
per-address BHT	PAg	PAs	PAp

[4] Tse-Yu Yeh and Yale N. Patt, *A Comparison of Dynamic Branch predictors that use two levels of branch hostory*, In: Proceedings of the 20th annual symposium on computer architecture (ISCA-20), pp 257-266, 1993.

Basic techniques

Branch prediction – Two-stage adaptive predictors

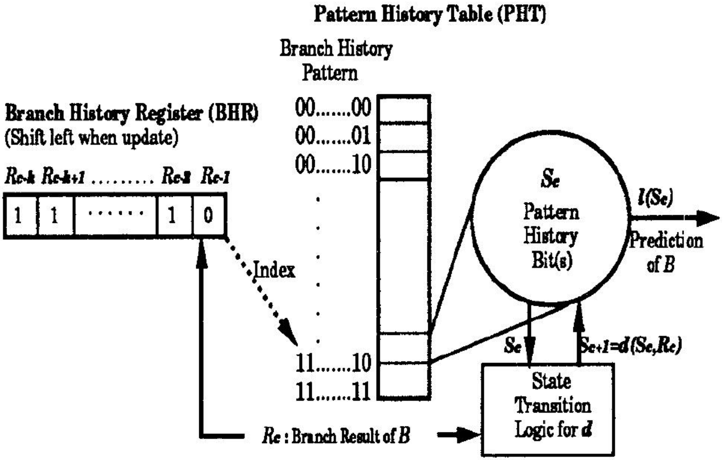
- Example: GAg(k) predictor



Basic techniques

Branch prediction – Two-stage adaptive predictors

- Example: GAg(4) predictor



Basic techniques

Branch prediction – Two-stage adaptive predictors

- Problems of two-stage adaptive predictors
  - The same bit pattern in BHR can be referenced to different parts of a program
  - This leads to possible PHT-interferences
    - Unrelated branching commands impact the prediction of each other
    - However, solutions to this problem exist

Basic techniques

Prediction by partial matching

- PPM-Algorithm of order n is composed of $n+1$ Markov Predictors of oder 0 to n
  - Algorithm tries to find a pattern that matches last n states with Markov predictor of order n
  - If this is not successful, Markov predictor with decreased order is instantiated.

Basic techniques

Prediction by partial matching

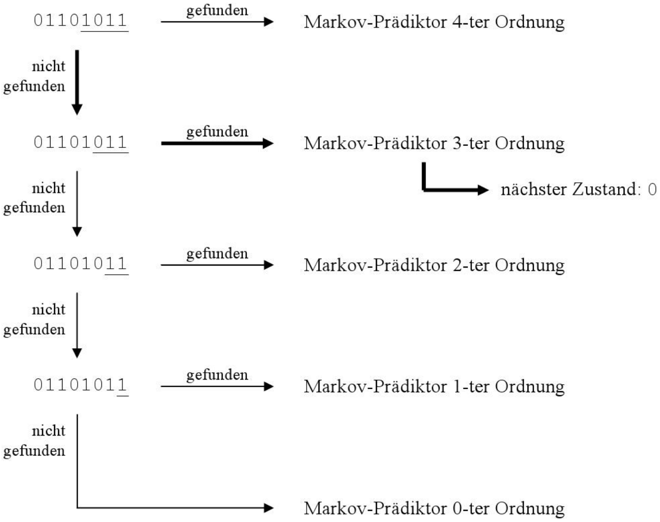
Prediction by partial matching

Input: $z_1, \ldots, z_{t-1}$ (Sequence of States)

Output: z_t (next state)

REPEAT
    Choose Markov predictor of order n
    If (pattern $z_{t-n}, \ldots, z_{t-1}$ is found)
        Calculate prediction
        BREAK
    Else
        $n = n - 1$
UNTIL $n = 0$
If $(n = 0)$
    Calculate prediction with predictor of order 0

Basic techniques
Prediction by partial matching



Outline

The state predictor method

1. Introduction and basic techniques
2. State predictors
3. Evaluation
4. Estimation of Reliability
5. Hybrid predictors
6. Properties of the state predictor method

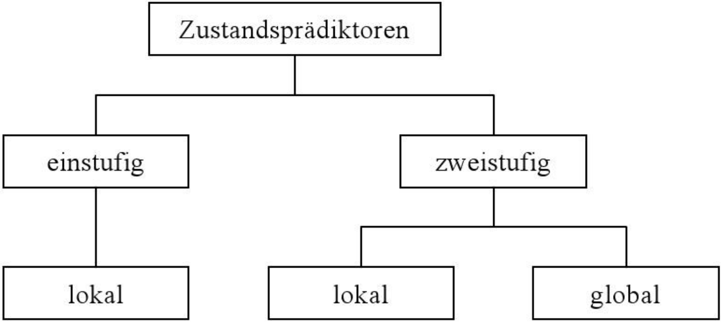
State predictors

Introduction

- The state predictor method
  - Utilise branch prediction techniques implemented on microprocessors
  - Distinction between local and global prediction
    - Example sequence: ACBCACBCABCAB
    - Global sub-sequence: BCAB
    - Local sequence with respect to A: CCBB
  - Local sequence: Context observes only neighbouring contexts that succeed its own occurrence

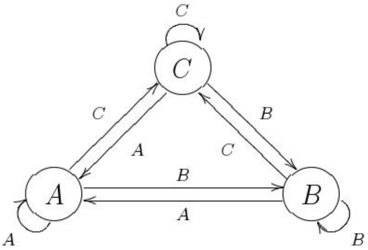
State predictors
Introduction

- Various state classes of state prediction methods



State predictors

One-stage state predictors

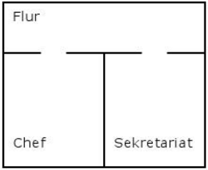


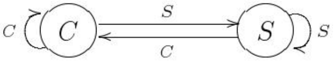
- One-state predictor
  - For every context a prediction graph of succeeding contexts is maintained
  - Prediction is always adapted to the observed succeeding context.
  - When prediction is incorrect, predicted context is adapted to the observed context

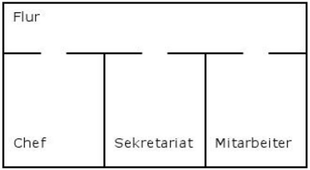
State predictors

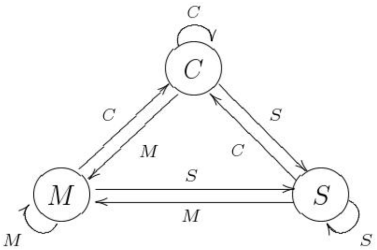
One-stage state predictors

- Example: Location prediction









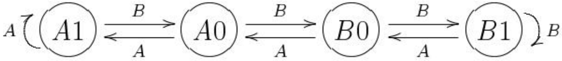
State predictors

One-stage state predictors

- Benefits
  - Small memory requirements
  - Fast 'adaptation' (lerning)
- Drawbacks
  - Learned behaviour is rapidly lost/forgotten

State predictors

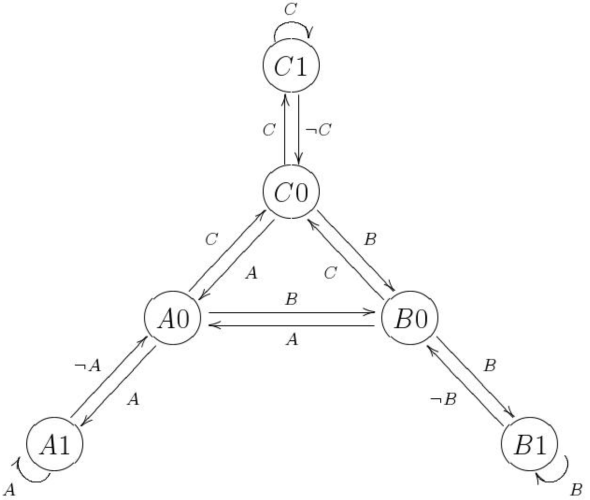
One-stage state predictors



- Two-state predictor
  - For every context a prediction graph of succeeding contexts is maintained
  - Two possible states for every possible succeding context
    - Weak state
    - Secure state

State predictors

One-stage state predictors



State predictors

One-stage state predictors

- Benefits
  - Small memory requirements
  - Fast 'adaptation' (lerning)
- Drawbacks
  - Learned behaviour is easily lost/forgotten

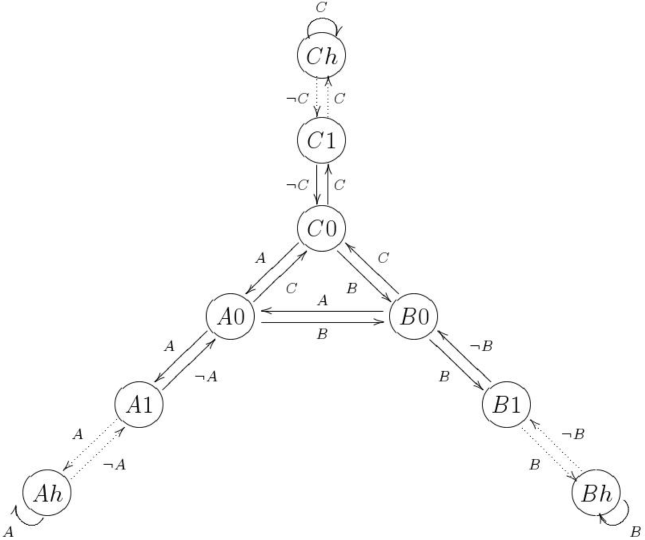
State predictors

One-stage state predictors

- k-state predictor
  - For every context a prediction graph of succeeding contexts is maintained
  - k possible states are associated with every possible succeeding context
    - $k - 1$ weak states
    - One secure state

State predictors

One-stage state predictors



State predictors

One-stage state predictors

- k-state predictor – Aspects
  - Benefits
    - Small memory requirements
  - Drawbacks
    - Learning of behaviour only restricted to local context view
  - Further aspects
    - Dimension k can also be learned

State predictors

Two-stage state predictors

- Global two-state predictors
  - Prediction on observed global sequence
  - Prediction stated by Two-state predictor
  - Also: Prediction by arbitrary k-State predictor possible

State predictors

Two-stage state predictors

- Example
  - Observed context pattern: ABCACBABACBACB

Schieberegister

$A\ C\ B$

Musterverlaufstabelle

Muster	Two-State-Prädiktor
$A\ B\ A$	$C0$
$A\ B\ C$	$A0$
$A\ C\ A$	–
$A\ C\ B$	$A1$
$B\ A\ B$	$A0$
$B\ A\ C$	$B1$
$B\ C\ A$	$C0$
$B\ C\ B$	–
$C\ A\ B$	–
$C\ A\ C$	$B0$
$C\ B\ A$	$C0$
$C\ B\ C$	–

State predictors

Two-stage state predictors

- Local two-state predictors
  - Prediction of local context sequences
  - Otherwise identical to global two-state predictors

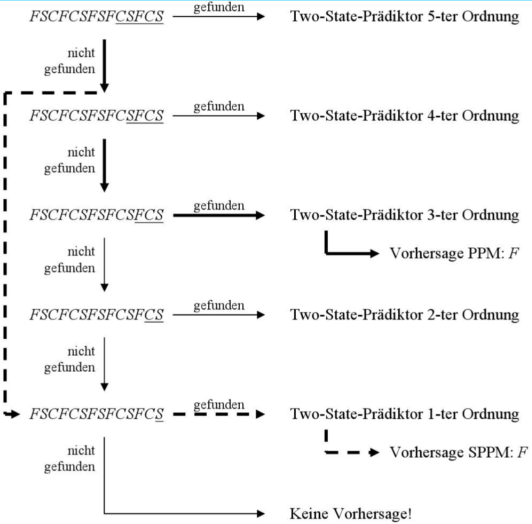
State predictor methods

Prediction by partial matching

- Extension of two-stage state predictor method by PPM
- Instead of fixed order of the prediction method: Variable order dependent on the maximum matching pattern length found in the obseved context sequence.

State predictor methods

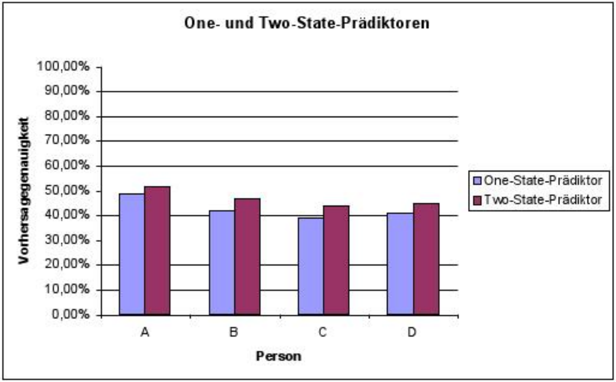
Prediction by partial matching – PPM and simple PPM



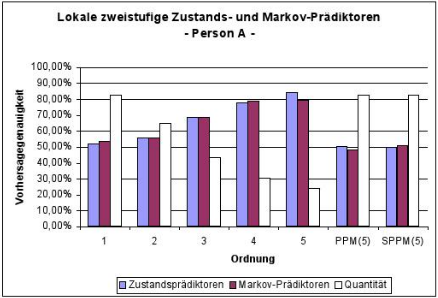
Outline

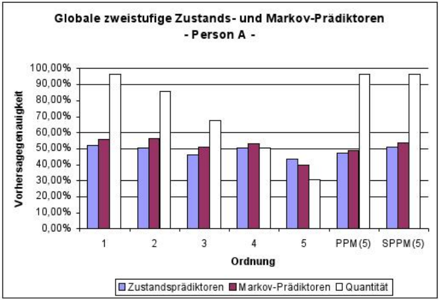
The state predictor method

Evaluation
Augsburg benchmarks

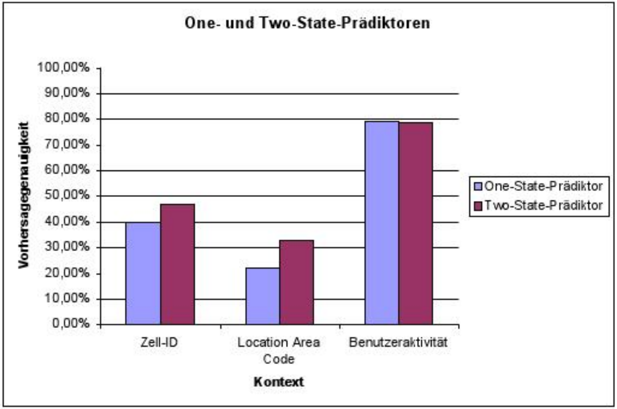


Evaluation
Augsburg benchmarks

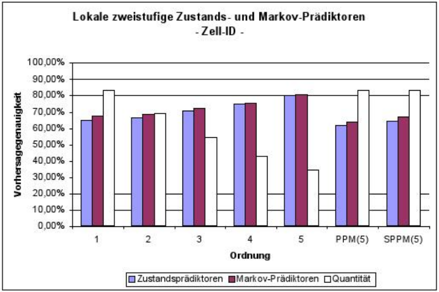


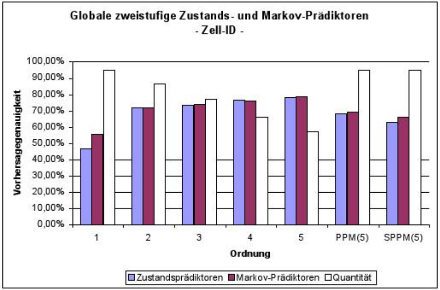


Evaluation
Nokia context data



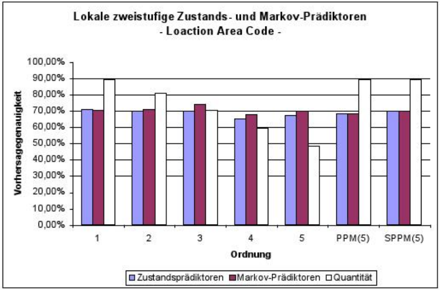
Evaluation
Nokia context data

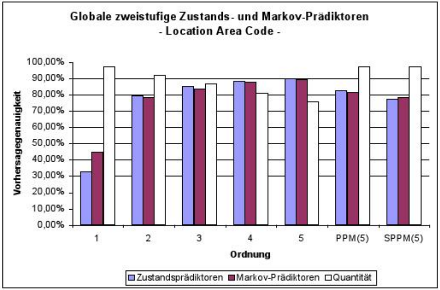




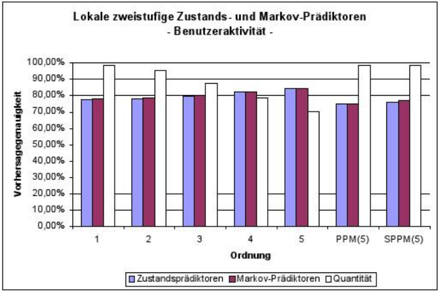
Evaluation

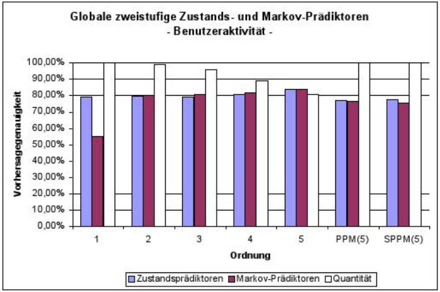
Nokia context data





Evaluation
Nokia context data





Outline

The state predictor method

Estimation of Reliability

Methods for reliability estimation

- Prediction is naturally error prone
- Sometimes, no prediction might be better than an erroneous prediction

Estimation of Reliability

Methods for reliability estimation

- Static reliability
  - Assumption:
    - For some patterns or contexts a prediction is not taken
    - Patterns are not classified as typical
    - Patterns frequently change – Low prediction accuracy
  - Contexts divided into reliable and non-reliable contexts

Estimation of Reliability

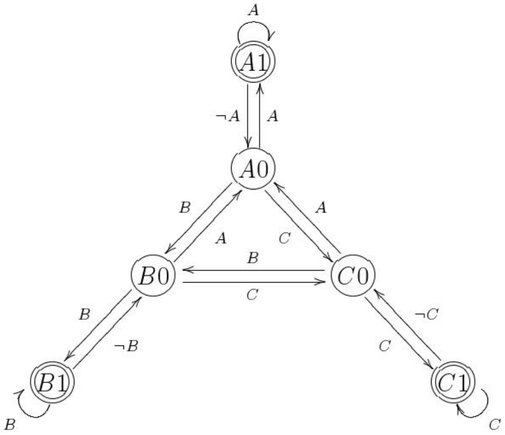
Methods for reliability estimation

- Consideration of secure states
  - Only applicable to two-state predictors
  - Two state predictor has for every context two states
    - Weak state
    - Secure state
- When context is observed often, it becomes secure
- Predictions are provided exclusively in secure states

Estimation of Reliability

Methods for reliability estimation

- Consideration of secure states



Estimation of Reliability

Methods for reliability estimation

- Threshold method
  - Compare recent prediction accuracy with predefined threshold.
    - Secure state: Accuracy above threshold
    - Insecure state: Accuracy below threshold
  - Correct predictions: c
  - Incorrect prediction: i

$$\frac{c}{c+i} \geq \alpha : \text{Secure state} \tag{3}$$

Schieberegister

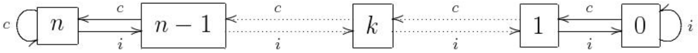
$p_1 \dots p_r$

Musterverlaufstabelle

Muster	Two-State-P.	c	i
...	...	...	...
$p_1 \dots p_r$	$C1$	x	y
...	...	...	...

Estimation of Reliability

Methods for reliability estimation



- Reliability counter
  - Accuracy of predictions 'counted' by reliability counter
  - Initial position of counter arbitrary
  - When prediction is correct, counter is increased
  - When prediction is incorrect, counter is decreased
  - When counter exeeds threshold, prediction is provided

Schieberegister

$p_1 \dots p_r$

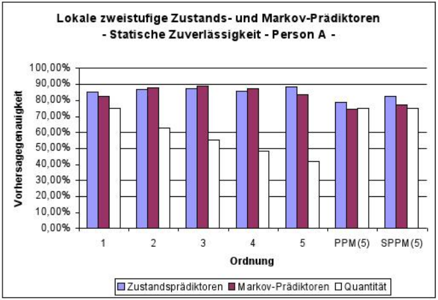
Musterverlaufstabelle

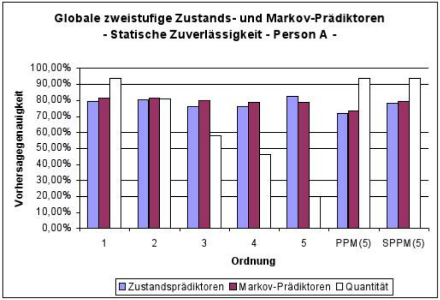
Muster	Two-State-P.	cc
...	...	...
$p_1 \dots p_r$	$C1$	k
...	...	...

Estimation of Reliability

Evaluation

- Static reliability – Augsburg Benchmarks

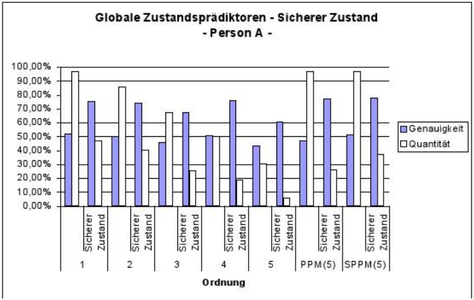




Estimation of Reliability

Evaluation

- Secure states – Augsburg Benchmarks

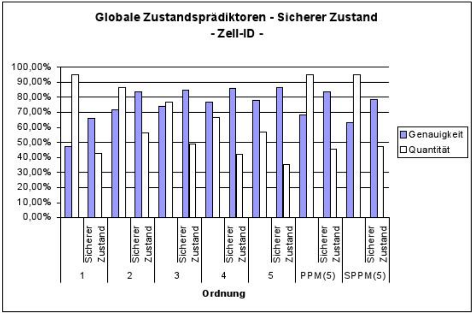


Person A	
Ordnung	Steigerung
1	48,80%
2	47,66%
3	39,41%
4	50,65%
5	29,57%
PPM(5)	56,35%
SPPM(5)	54,39%

Estimation of Reliability

Evaluation

- Secure state – Nokia context data

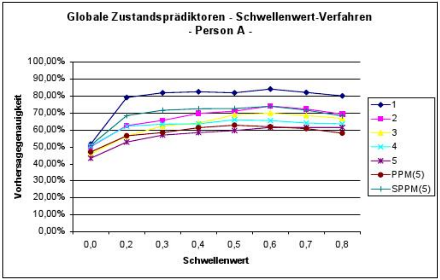


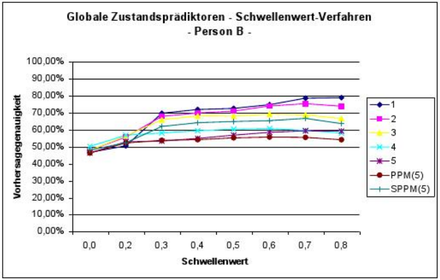
Zell-ID	
Ordnung	Steigerung
1	35,79%
2	41,29%
3	41,82%
4	39,23%
5	38,04%
PPM(5)	47,96%
SPPM(5)	40,85%

Estimation of Reliability

Evaluation

- Threshold method – Augsburg Benchmarks

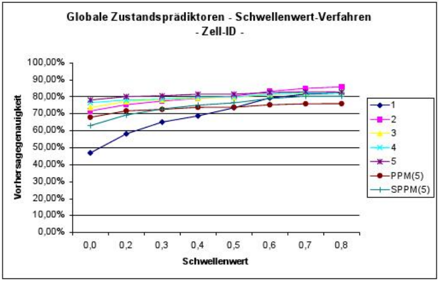


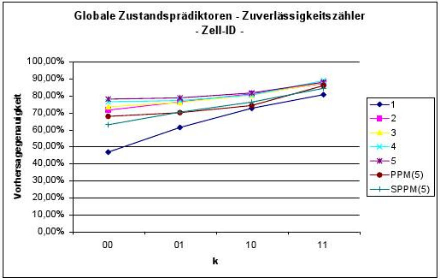


Estimation of Reliability

Evaluation

- Threshold method – Nokia context data

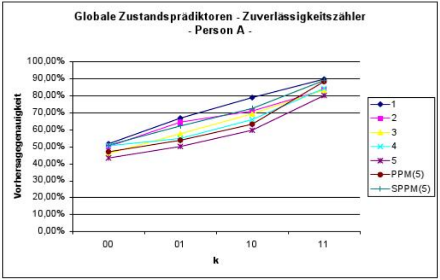


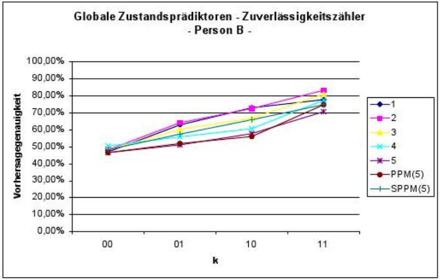


Estimation of Reliability

Evaluation

- Reliability counter – Augsburg Benchmarks

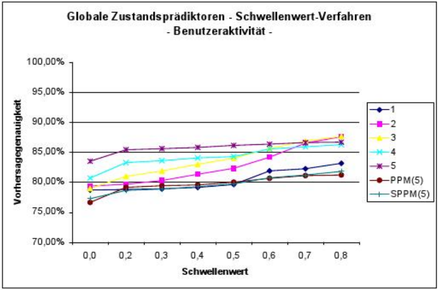


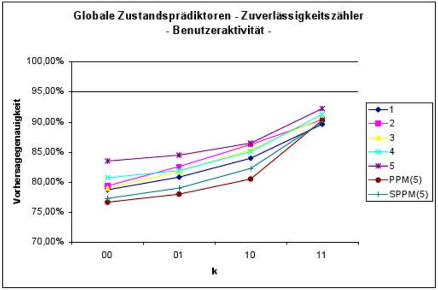


Estimation of Reliability

Evaluation

- Reliability counter – Nokia context data





Estimation of Reliability

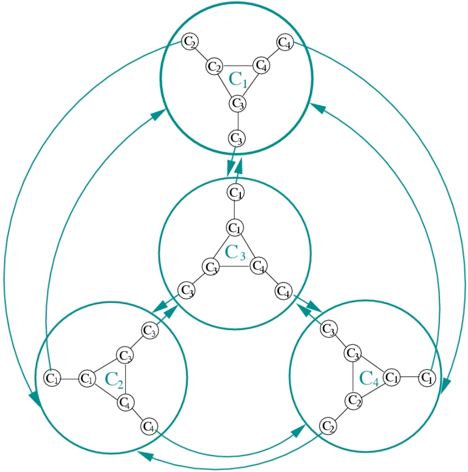
Conclusion

- Is the state prediction method a Markov prediction class algorithm?

Estimation of Reliability

Conclusion

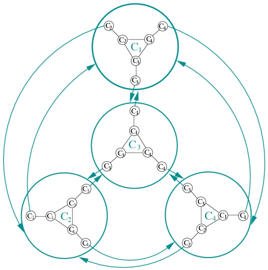
- Is the state prediction method a Markov prediction class algorithm?



Estimation of Reliability

Conclusion

- Is the state prediction method a Markov prediction class algorithm?
  - The state prediction approach defines the mechanism to adapt transition probabilities
  - Only transition probabilities 1 and 0 possible
  - Markov prediction more powerful



Outline

The state predictor method

1. Introduction and basic techniques
2. State predictors
3. Evaluation
4. Estimation of Reliability
5. Hybrid predictors
6. Properties of the state predictor method

Hybrid predictors

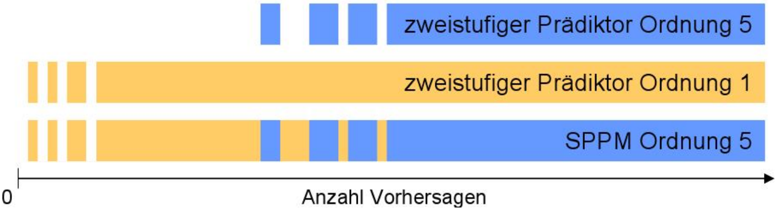
Introduction

- Can the combination of multiple prediction approaches improve the prediction accuracy?
  - Warm-up predictor
  - Majority predictor
  - Reliability predictor

Hybrid predictors

Warm-up-predictor

- Prediction approaches with low-order often provide quick szenario adaptation
- Complex patterns not possible with low-order models



Hybrid predictors

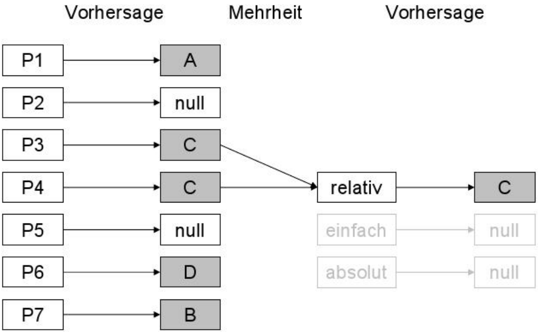
Majority prediction

- Compute prediction by various prediction approaches
- Majority of prediction results determines actual prediction
  - Relative majority
    - Prediction that was stated most often by all approaches
  - Bare majority
    - More than half of the stated predictions are identical
  - Absolute majority
    - More than half of the possible predictions identical

Hybrid predictors

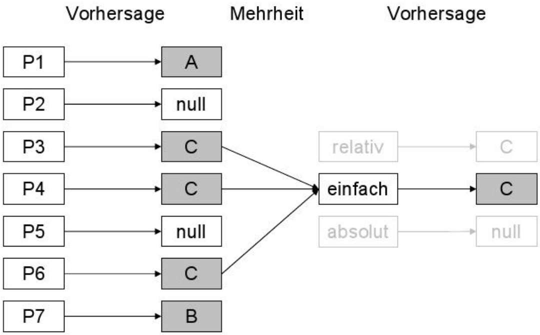
Majority prediction

- Relative majority



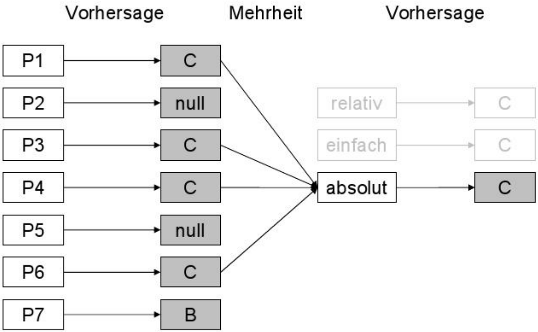
Hybrid predictors
Majority prediction

- Bare majority



Hybrid predictors
Majority prediction

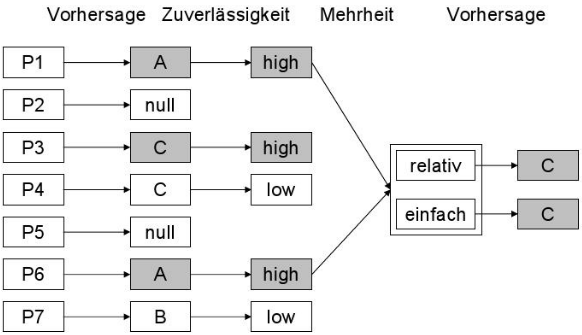
- Absolute majority



Hybrid predictors

Reliability predictor

- Choose the prediction with highest prediction accuracy



Hybrid predictors
Reliability predictor

- Choose the prediction with highest prediction accuracy
  - Several selection criteria possible
    - Primary selection criterium
    - Secondary selection criterium
    - Tertiary selection criterium

Hybrid predictors

Reliability predictor

- Selection criteria

primär	sekundär	tertiär
Sicherer Zustand	relativ	ohne Barriere
		mit Barriere
	einfach	ohne Barriere
		mit Barriere
Schwellenwert-Verfahren	relativ	ohne Barriere
		mit Barriere
	einfach	ohne Barriere
		mit Barriere
Zuverlässigkeitszähler	relativ	ohne Barriere
		mit Barriere
	einfach	ohne Barriere
		mit Barriere

Hybrid predictors
Conclusion

- It was shown, that the warm-up predictors achieve low accuracy
- With Majority predictors and reliability predictors the prediction accuracy can be improved

Outline

The state predictor method

1. Introduction and basic techniques
2. State predictors
3. Evaluation
4. Estimation of Reliability
5. Hybrid predictors
6. Properties of the state predictor method

Properties of the state predictor approach

Processing load

- Runtime for computing a prediction: ($O(1)$)
    - Current state directly prediction

Properties of the state predictor approach

Memory requirements

- Memory requirements
  - Dependent on the number of contexts observed – size of the transition matrix
  - Order 1: $O(|C|^2)$
  - Order k: $O(|C|^{k+1})$

Properties of the state predictor approach

Prediction horizon

- Prediction horizon can be extended by iterative prediction
  - Utilise predicted contexts as input
- Problem: Less accurate
  - Predicted contexts more error prone than measured values

Properties of the state predictor approach

Adaptability

- The state prediction approach is able to adapt to changing environments
  - Adaptation only to simple patterns

Properties of the state predictor approach

Multi-dimensional time series

- The state prediction algorithm is not suited for multi-dimensional time series
  - Designed for one-dimensional Input
  - Possible: Aggregation of multi-dimensional time series to one-dimensional time series.

Properties of the state predictor approach

Iterative prediction

- Iterative Prediction possible
  - Steep decrease in prediction accuracy expected since prediction horizon is only 1
  - Increase of prediction horizon possible by Aggregation of context sequence of fixed length in one state
    - Prediction horizon fixed
    - Increase in Memory consumption and processing time
    - When l contexts are aggregated: l^C states
    - Runtime:

      $$O(n \cdot l^{C^2}).$$
    - Memory consumption:

      $O(l^{C^2})$ (order one)

      $O(l^{C^{k+1}})$ (order k)

Properties of the state predictor approach

Prediction of context durations

- Prediction of context duration not possible
  - Only simple sequence of occurring contexts possible

Properties of the state predictor approach

Approximate matching of patterns

- Exact pattern matching
  - The state prediction algorithm utilises exact pattern matching

Properties of the state predictor approach

Context data types

- All context data types supported
  - Every distinct context type one state
  - Probably drastic increase in runtime and memory consumption for numeric context types
  - Possible: Assign intervals to states

Properties of the state predictor approach

Pre-processing

- Pre-processing required to construct context transition probabilities
- On-line approach feasible – learning
- Runtime: $O(k)$
  - Count frequency of specific context transitions in training time series of length k

Aspects of prediction algorithms

Summary

	IPAM	ONISI	Markov	CRF
Numeric Contexts	yes	no	yes	
Non-numeric Contexts	yes	yes	yes	
Complexity	$O(k)$	( )	$O(C^2)$	
Learning ability	(no)	yes	yes	
Approximate matching	no	no	no	
Multi-dim. TS	(no)	(no)	(no)	
Discrete data	yes	yes	yes	
Variable length patterns	no	yes	no	
Multi-type TS	yes	no	(no)	
Continuous data	no	no	no	
Pre-processing	$O(k)$	–	$O(k)$	
Context durations	no	no	no	
Continuous time	no	no	yes	

Aspects of prediction algorithms

Summary

	SPM	Align	SOM	PCA
Numeric Contexts	yes			
Non-numeric Contexts	yes			
Complexity	$O(1)$			
Learning ability	(yes)			
Approximate matching	no			
Multi-dim. TS	(no)			
Discrete data	yes			
Variable length patterns	yes			
Multi-type TS	no			
Continuous data	no			
Pre-processing	$O(k)$			
Context durations	no			
Continuous time	no			

Properties of the state predictor approach

Conclusion

- Simple, straightforward prediction approach
- Model can be applied to numerical and non-numerical data alike.
- Special case of a Markov predictor
- Less powerful than Markov prediction
- Not suited for complex prediction scenarios
- Prediction that reaches farther into future implicitly utilises already predicted data which might consequently decrease the prediction accuracy.