

Context Prediction

FTW Tutorial day on "Personalization and context-dependent service enablers"
2. July 2008 14:00

Rene Mayrhofer, University of Vienna

Vision: A „Personal Digital Assistant“ that can live up to its name.

Motivation

Problem:

- Most information appliances are difficult to use for non-technology-savy users
- Devices only react to user input

Aim:

- Make information appliances „smarter“ in a sense that they are easier to use
- Devices should be proactive
- Devices should adapt to the user

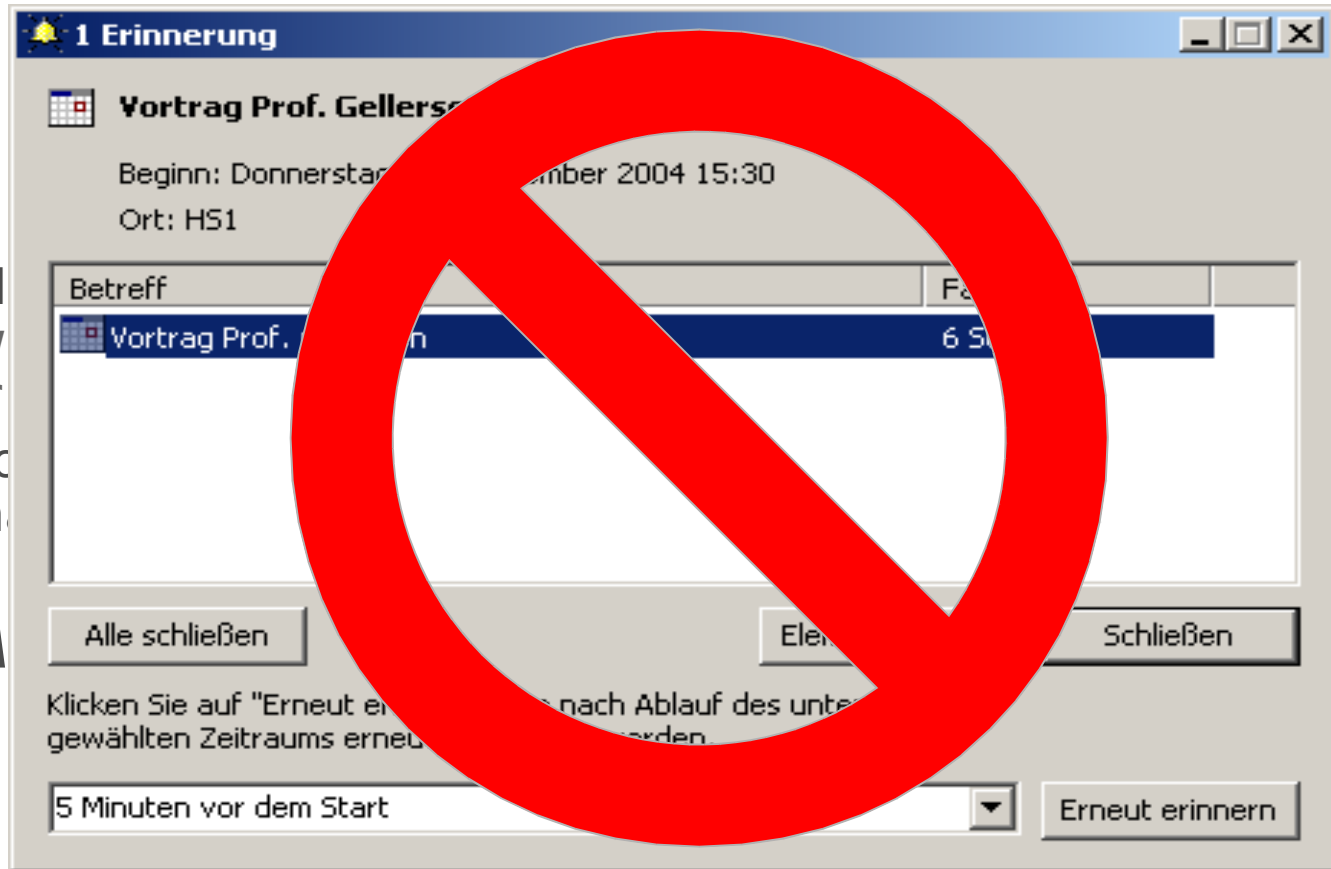
Problem statement:

How can an information appliance infer its (or its users) current context and predict future context?

Motivating Example

- Reminder to buy milk
- When to deliver: depends ...
 - not time/location specific
- How to deliver: depends ...
 - appropriate modality

Problem area



What is *Context*?

- e.g. Wikipedia conditions within an utterance or
- In the area of something h

es and longer
hich

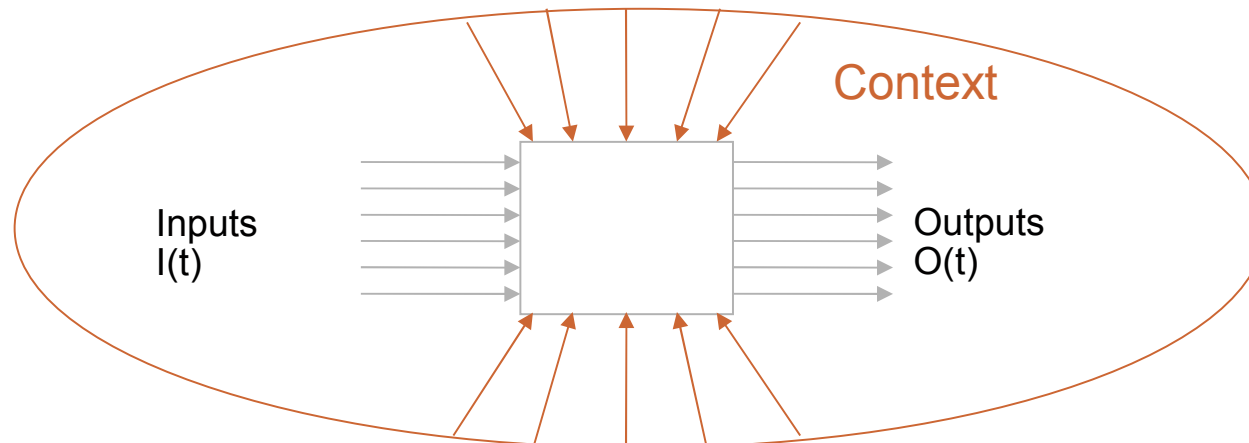
What is *Context A*

Context Awareness

Many definitions in this sense, e.g.:

- *"Three important aspects of context are: where you are, who you are with, and what resources are nearby [...]. Context encompasses more than just the user's location, because other things of interest are also mobile and changing."* [SAW 1994]
- *"any information that can be used to characterize the situation of an entity, where an entity can be a person, place or a physical or computational object"* [Dey 1999]
- A working model for context [Schmidt 2002]

Context is everything **except** the explicit in- and outputs [LS 2000].



[SAW 1994] B.N. Schilit, N. Adams, R. Want: „Context-aware computing applications“

[Schmidt 2002] A. Schmidt: „Ubiquitous Computing – Computing in Context“, PhD Thesis, Lancaster Univ.

[Dey 1999] A. Dey, G.D. Abowd, D.Salber: „A Context-based infrastructure for smart environments“

[LS 2000] H. Lieberman, T. Selker: „Out of context: Computer systems that adapt to, and learn from, context“

Context Awareness (2)

Context has a vast multitude of different aspects, e.g.

- time
- location
- physical (temperature, humidity, etc.)
- social (with colleagues / family etc.)

⇒ It seems sensible to use multiple small sensors instead of a single, but more powerful one (cf. Gellersen et.al.)



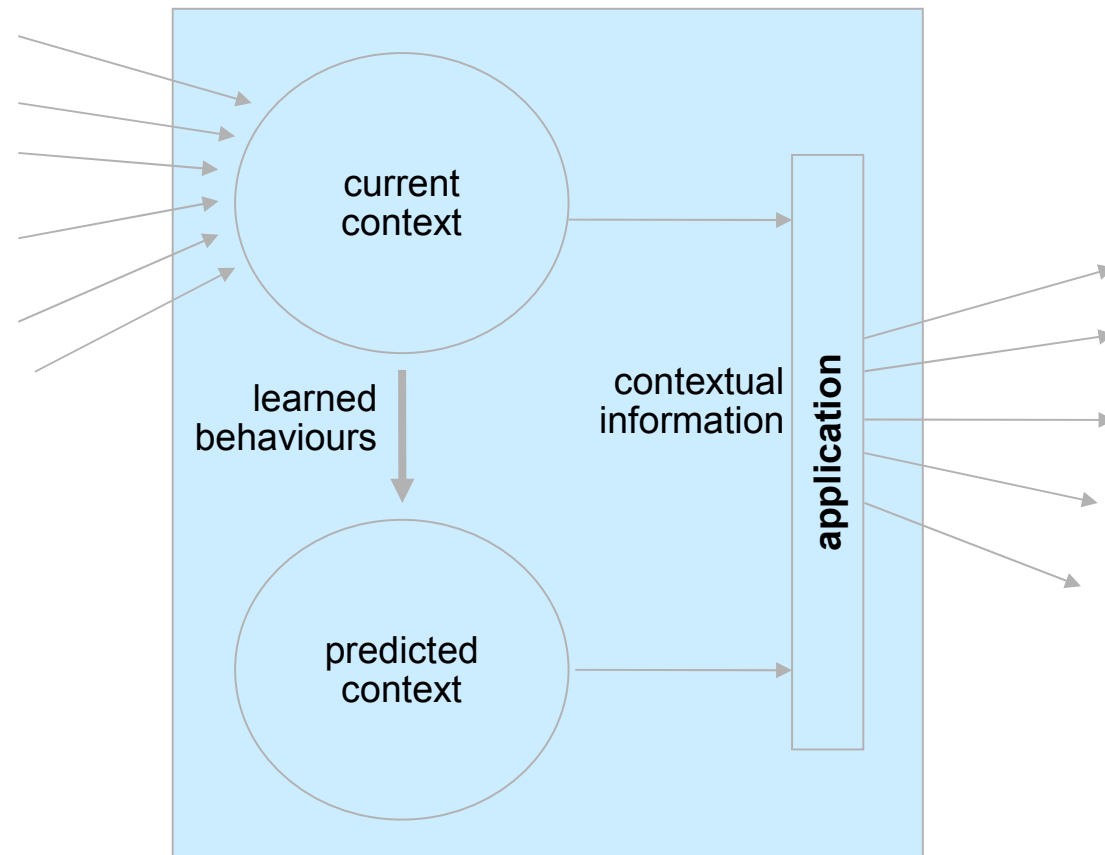
What is and what isn't context?

- You, as end user or designer, decide:
Application dependent and situation-specific
- ... based on real-world, noisy, and incomplete data
- Domain-specific knowledge helps, but modeling the world is hard!
- ⇒ How much context can we infer using data-driven machine learning approaches?

Basic approach to context prediction

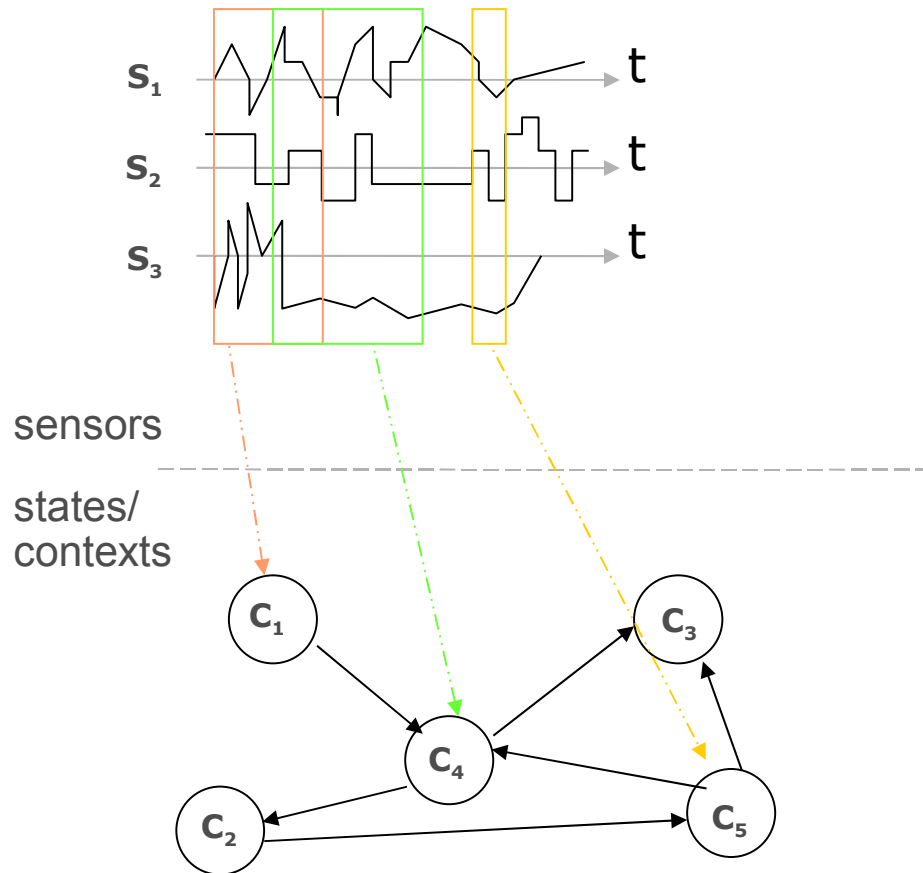
- Recognize the current context based on multiple sensors (e.g. [Schmidt 2002])
- Predict future context by learning user behaviour
- ... as far as possible, **without domain specific knowledge**

sensors



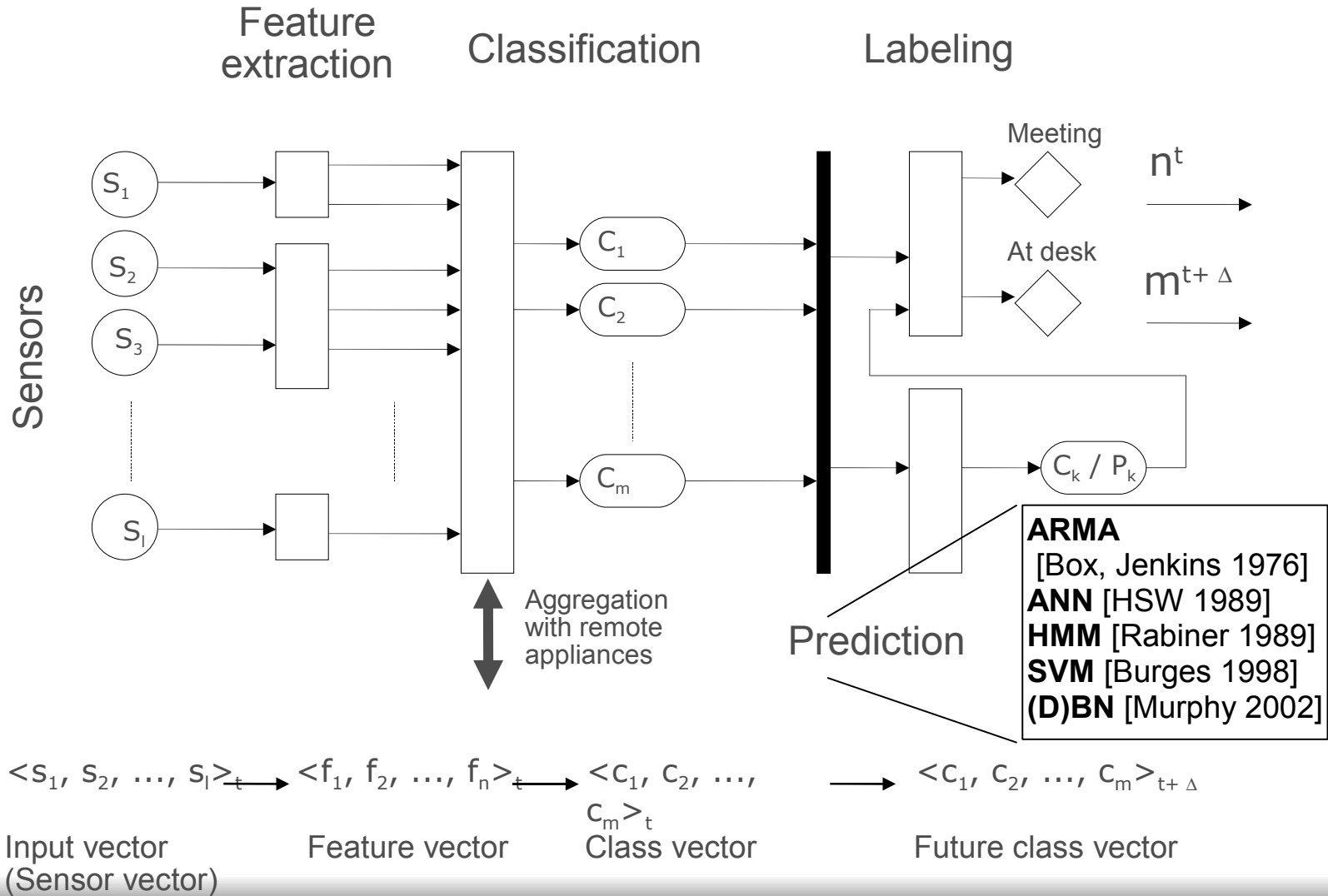
[Schmidt 2002] A. Schmidt: "Ubiquitous Computing – Computing in Context", PhD Thesis, Lancaster Univ.

Concept



- Sensors yield time series
- Particular patterns in the input streams can be interpreted as the states of a (observable but not controllable) state machine
- These states are understood as the device (or user) contexts
- Then it becomes possible to predict future contexts by extrapolating the state trajectory into the future

Architecture



Step 1: *Sensor data acquisition*

- Sensors yield time series
- Sampled either at regular intervals or based on events
- Examples for currently available “sensors” that can help to determine the current context on a typical mobile off-the-shelf device:
 - time
 - application being used
 - brightness
 - microphone
 - Bluetooth
 - WLAN
 - docked/undocked
- Additional sensors can be connected easily:
 - GPS
 - GSM
 - compass
 - accelerometers
 - tilt sensors
 - temperature sensors
 - pressure sensors, etc.
- **Sharing of sensor values between nearby devices**



Step 2: *Feature extraction*

- Transforms raw sensor values into more meaningful features
- Applying domain-specific knowledge
- Multiple features can be generated from a single sensor data stream (and vice versa)

⇒ **High dimensional feature space**

- Different types of features:
 - Numerical (continuous): e.g. brightness, heart rate, microphone
 - Numerical (discrete): e.g. number of access points in range
 - Ordinal: e.g. day of week
 - Nominal: e.g. current WLAN SSID, list of WLAN/BT devices in range
- Notice: only two operations are necessary for each feature dimension
 - **similarity measure** (distance metric)

- **adaptation operator**

$$\alpha(f, g, a) : \alpha(f, g, a) := \begin{cases} f & \text{if } a \leq 0.5 \\ g & \text{if } a > 0.5 \end{cases}$$

[MRF 2003b] R. Mayrhofer, H. Radi, A. Ferscha: „Feature Extraction in Wireless Personal and Local Area Networks“, Proceedings of 6th IFIP MWCN 2003, World Scientific, October 2003

[MRF 2003b] R. Mayrhofer, H. Radi, A. Ferscha: „Feature Extraction in Wireless Personal and Local Area Networks“, Proceedings of 6th IFIP MWCN 2003, World Scientific, October 2003

Step 3: *Classification*

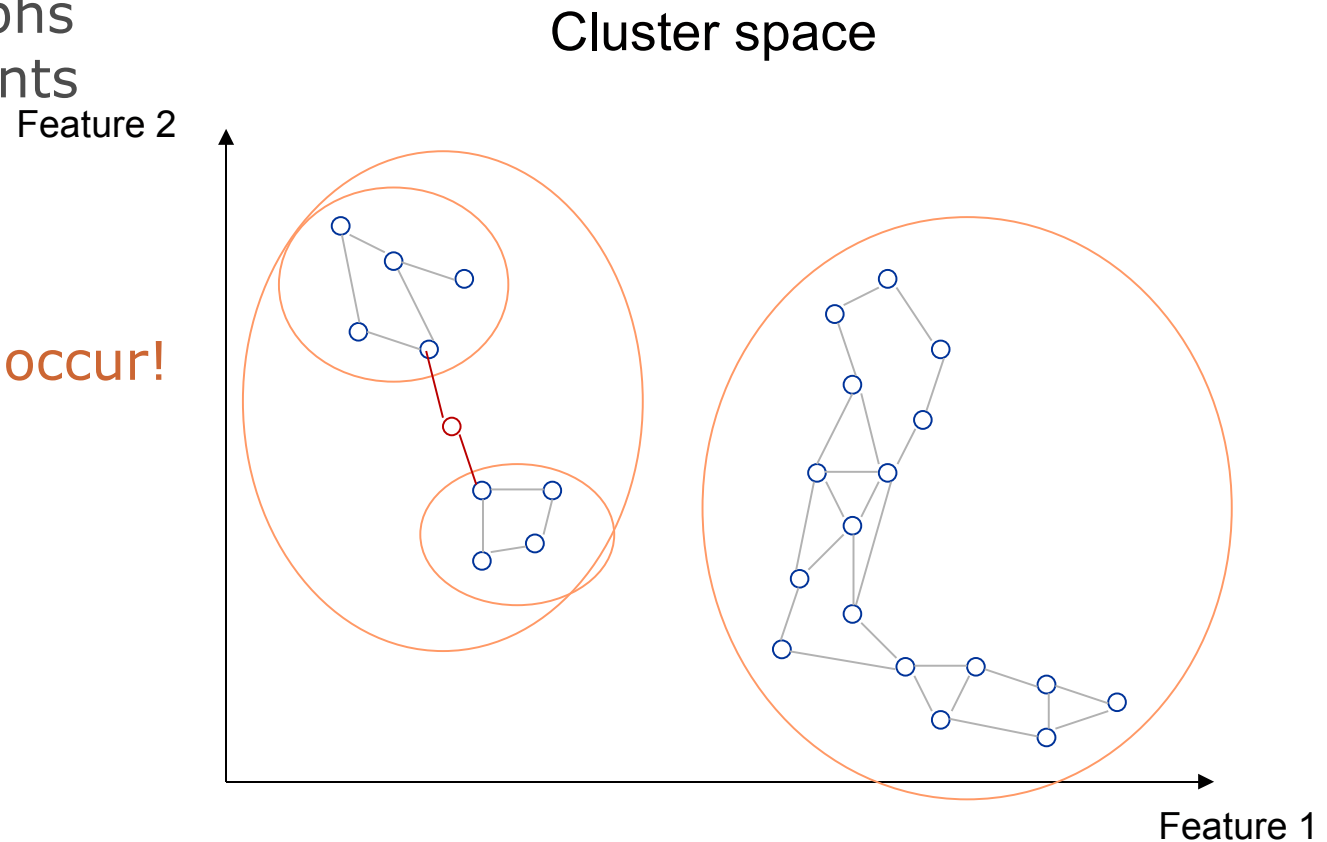
- Classifies features and recognizes common patterns (clusters) in the input data ⇒ possible without user interaction, **unobtrusive**
- Different types of classification algorithms
 - Type (**partitioning** / hierarchical)
 - „**soft**“ / „hard“ classification
 - supervised / **unsupervised**
- Requirements on algorithms for context recognition:
 - Online / incremental
 - Adaptive
 - Dynamic number of classes and dynamic structure
 - Finding clusters in sub spaces
 - “Soft” classification
 - Robustness against noise
 - Low resource consumption
 - Simplicity
 - Interpretability of classes / protection of user privacy

Classification: Algorithms

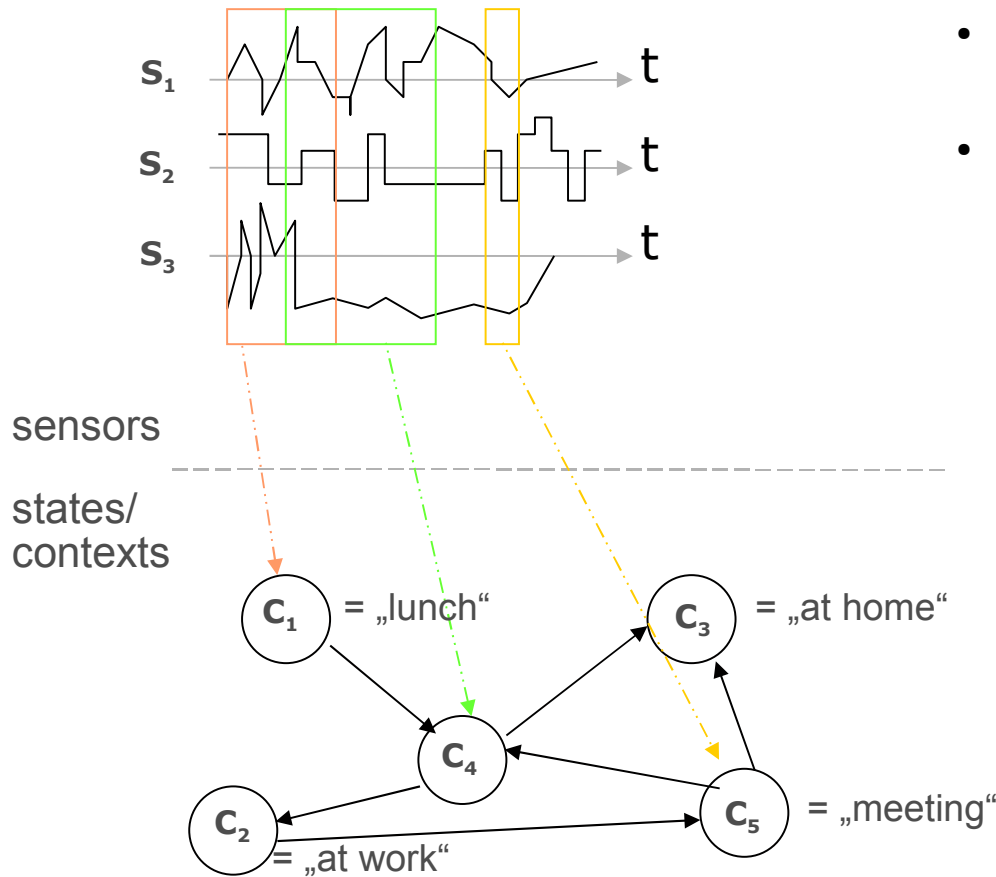
Algorithm	Network topology	Topology preserving	Competitive
SOM [SAT 1999]	fixed	yes	soft
RSOM	fixed	yes	soft
K-Means	fixed	no	hard
Leader	variable	no	hard
Growing K-Means [DWM 2002]	variable	no	hard
Neural Gas	variable	no	soft
Neural Gas + Competitive Hebbian Learning	variable	yes	soft
Growing Neural Gas [Fri 1995]	variable	yes	soft
Incremental DBSCAN [SWX]	variable	No	hard

(LL)GNG: Meta clusters

- Arbitrary shapes in cluster spaces formed by edges between clusters
⇒ *Meta-clusters*
- Cyclic, undirected graphs consisting of components
- Each component can be assigned an ID
- **But: meta-cluster splits and merges can occur!**



Step 4: *Labeling*



- 1: $\{0,1\}$ mapping of (meta) clusters to context labels
- basically two options:
 - for stable (meta) clusters from classification step: direct mapping of (meta-) cluster ID to labels
 - for significantly varying (meta) clusters (by learning/adaptation): additional, simple clustering step [Lae 2001]

[Lae 2001] K. van Laerhoven and S. Lowette: "Real-time analysis of data from many sensors with neural networks", Proceedings of ISWC, IEEE Press, October 2001

Step 5: *Prediction*

- Recognized contexts can be interpreted as „states“ of a state machine
- Monitoring the state trajectory allows to extrapolate and thus to predict it
- Important aspects of time series prediction:
 - periodical patterns (e.g. week ends, regular meetings)
 - sequential patterns (e.g. travel preparations, preparing a meal)
 - trends (changing behaviours)
- Requirements on algorithms for context prediction:
 - Unsupervised model estimation
 - Online
 - Incremental model growth
 - Confidence estimation
 - Automatic (implicit) feedback
 - Manual (explicit) feedback
 - Long term vs. short term prediction

Options for context prediction

- Based upon the trajectory of context classes
- Advantage over the independent prediction of different feature values: consideration of all aspects of context that are recognized by the available sensors
- Two options:
 - Predict each dimension of the class vector as **continuous time series**
⇒ does not consider relationships between context classes, but allows for overlapped contexts
 - Aggregate all classes to a single, **categorical time series** by using the most probably context class at each time step: *AAACCBCCAAADDDDDDEEEEEEBBAAAA.....*
⇒ implicitly considers relationships between context classes, because they are considered as mutually exclusive

⇒ Flat vs. hierarchical/overlapping **context model**

- Many known methods for time series prediction to select from for a specific application
- Selection is necessary, because the requirements/features of the specific methods are too diverse (there is no “best” method for time series prediction yet)

Prediction: Possible algorithms (1)

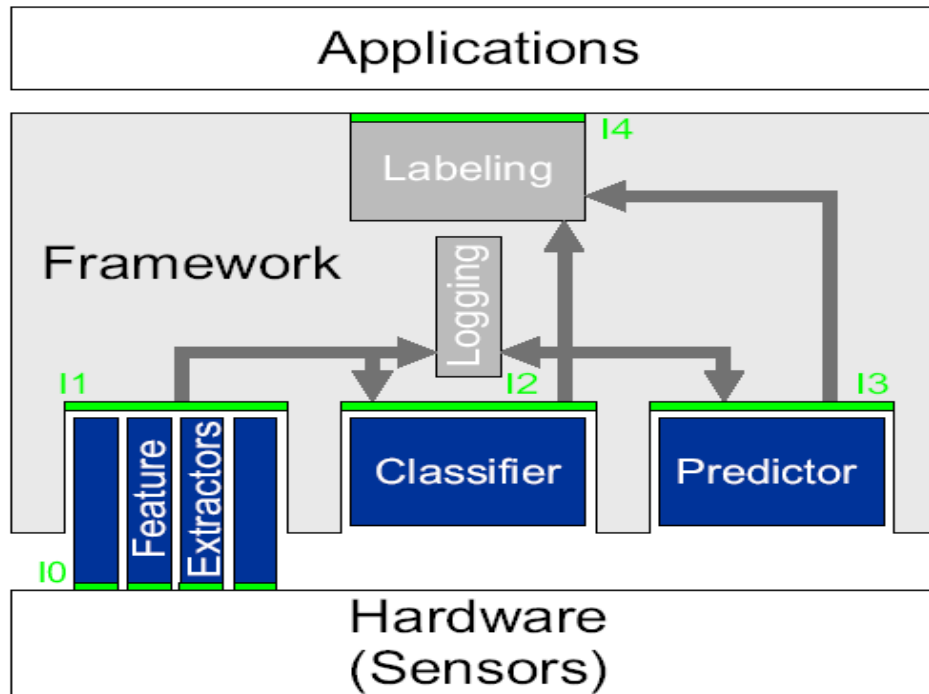
- ARMA: suitable for single-dimension, continuous time series forecast (option 1)
But: relationship between context classes not regarded
- ANN (e.g. back-propagation MLP): suitable for multi-dimensional, continuous time series forecast
But: needs known output (supervised learning) and long training time with many samples; no online mode, no incremental model growing, no confidence estimation
- HMM: allows discrete forecast
But: assumes statistical independence of events, which is definitely not the case for subsequent contexts

Prediction: Possible algorithms (2)

- MavHome project [Das 2002] uses prediction algorithms (currently) for predicting user locations
- Two algorithms for covering both aspects:
 - *Active Lempel-Ziv* for finding and predicting sequential patterns: online / incremental
 - *Episode Discovery (ED)* for identifying events at some regular intervals or in response to other events
- Back propagation ANN for mixing results of both algorithms

[Das 2002] S.K. Das, D.J. Cook, A. Bhattacharya, E. Heiermann, T. Lin: „The role of prediction algorithms in The MavHome smart home environment”, IEEE Wireless Communications 9(6)

Implementation as Software Framework



Cross-platform:

- Currently for Win32, Windows CE (>=3.0), Linux IA32 and ARM and (partially) Symbian OS
- Based on modules that are loadable at run-time:
 - Feature extractors (= sensor data acquisition + feature extraction)
 - Classifiers
 - Predictors
- Labeling is realized by providing network transparent interfaces (SOAP) ⇒ splitting HCI issues from context recognition/prediction issues
- Designed for resource limited devices

Further Challenges

Taxonomy

- What context is important? When?
- How do you represent context?
- Standards to allow exchange of context?

Real-World Knowledge

- Knowledge about how the world works for advanced reasoning

Interpretation

- Lots of interesting work here
- Most context-awareness deals with simple forms of context
- Sophisticated applications require more sophisticated context
- Machine learning, sensor fusion, modeling uncertainty, etc

Privacy

- Capturing/collecting lots of information about people, places and devices
- Users uncomfortable when they don't know what is being collected and how it's used
- Seemingly harmless information can lead to undesired (and potentially misleading) inferences when combined with other data

QoS

- Context is derived from sensor observation of the world: inherently limited and noisy
 - Coverage, Reliability
 - Resolution, Accuracy, Precision
 - Frequency, Timeliness

Thank you for your attention!

Slides: <http://www.mayrhofer.eu.org/presentations>
Later questions: rene.mayrhofer@univie.ac.at

OpenPGP key: 0xC3C24BDE
7FE4 0DB5 61EC C645 B2F1 C847 ABB4 8F0D C3C2 4BDE