Recognizing and Predicting Context by Learning from User Behavior

15. September 2003, Jakarta

Rene Mayrhofer
Institut für Praktische Informatik
Johannes Kepler Universität Linz, Austria
rene@soft.uni-linz.ac.at

Vision

A „Personal Digital Assistant" that can live up to its name.

Mission

Problem:

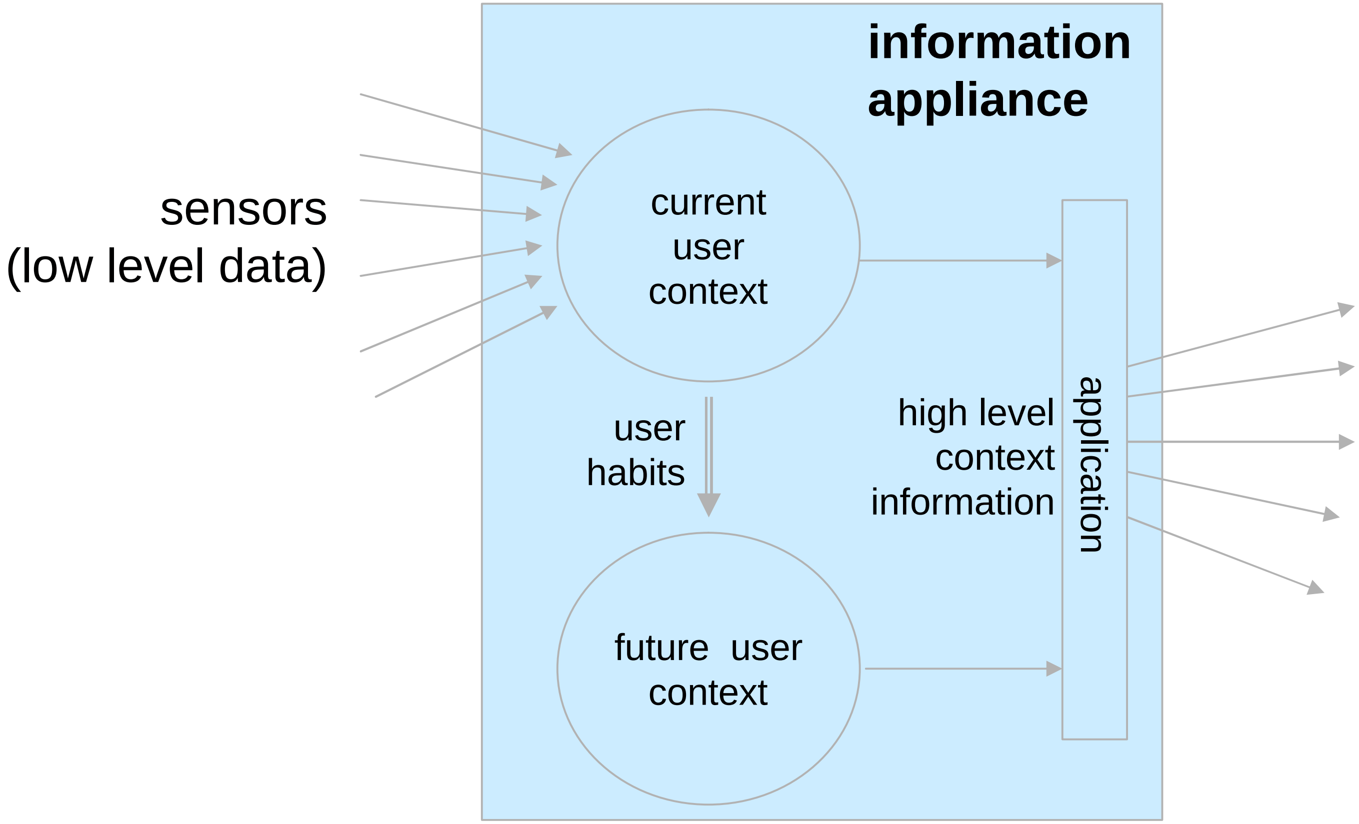
- Most information appliances are difficult to use for non-technology-savy users
- Devices only react to user input
- Users have to adapt to the devices

Aim:

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

Approach

- Personal information appliances should learn from user's habits
- Exploiting multiple sensors for context awareness
- Predicting future user context by learning from the past



Content

- Introduction
  - Context awareness
  - Proactivity
- Architecture
- Step 0: Sensor data acquisition
- Step 1: Feature extraction
- Step 2: Classification
- Step 3: Labeling
- Step 4: Prediction
- First results

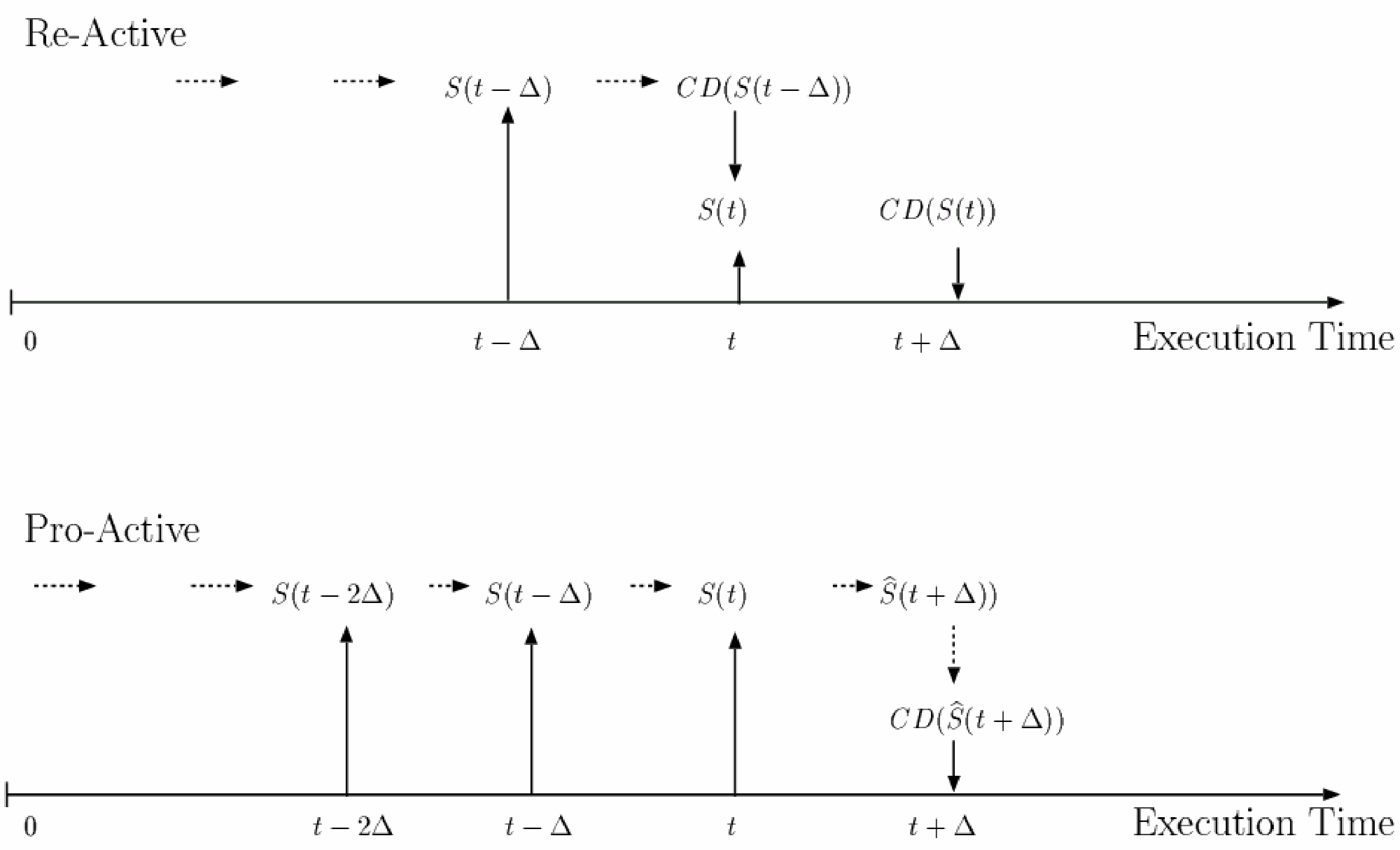
Context awareness

- Many definitions for context, e.g. by Dey as

  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

- Context has many aspects
- Using multiple simple sensors seems more reasonable to capture different aspects of context

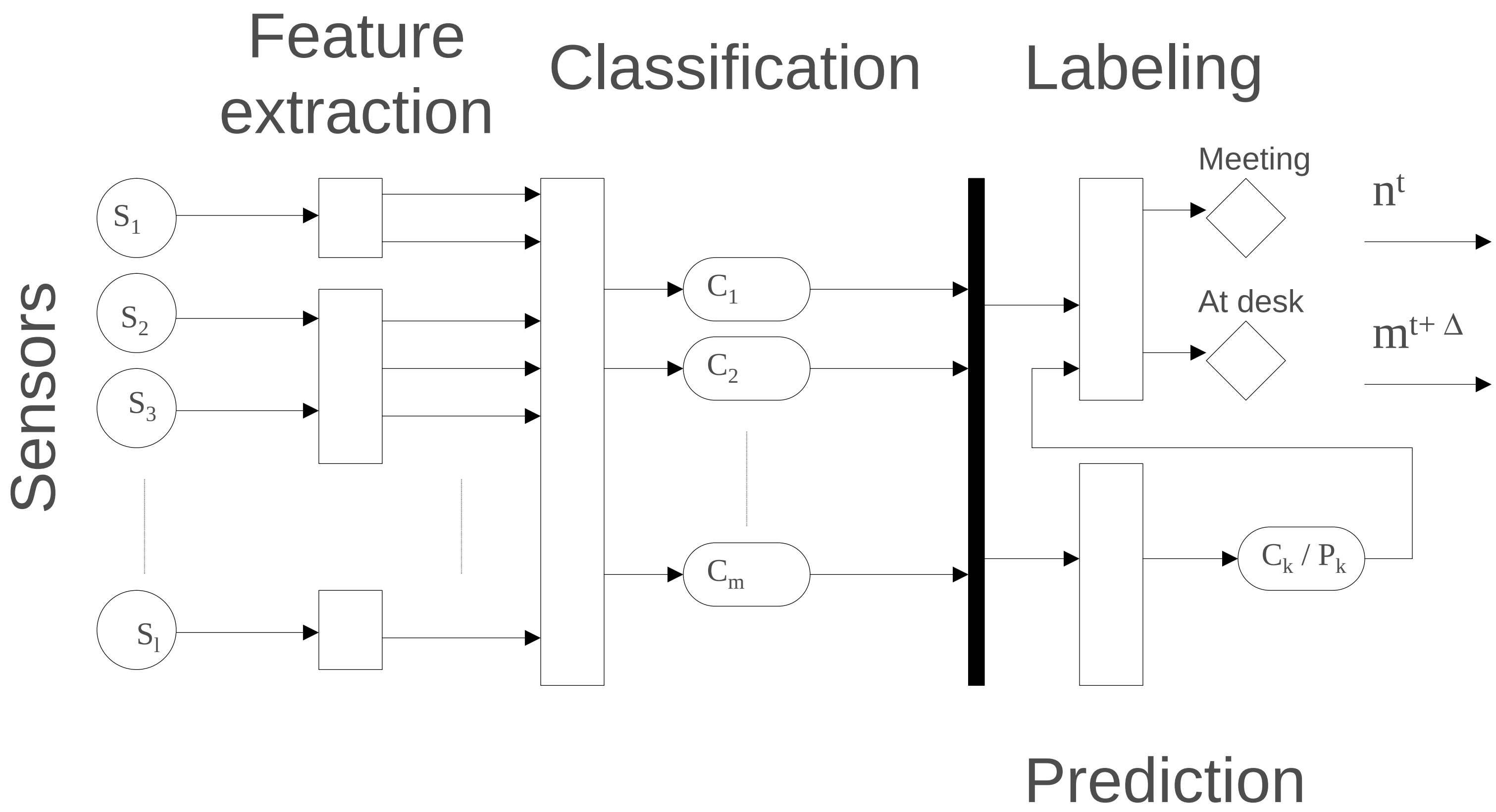
Proactivity



Content

- Introduction
- Architecture
- Step 0: Sensor data acquisition
- Step 1: Feature extraction
- Step 2: Classification
- Step 3: Labeling
- Step 4: Prediction
- First results

Architecture



$$\langle s_1, s_2, \ldots, s_l \rangle_t \longrightarrow \langle f_1, f_2, \ldots, f_n \rangle_t \longrightarrow \langle c_1, c_2, \ldots, c_m \rangle_t \longrightarrow \langle c_1, c_2, \ldots, c_m \rangle_{t+\Delta}$$

Input vector (Sensor vector) | Feature vector | Class vector | Future class vector

Content

- Introduction
- Architecture
- Step 0: Sensor data acquisition
- Step 1: Feature extraction
- Step 2: Classification
- Step 3: Labeling
- Step 4: Prediction
- First results

Sensors for (mobile) information appliances

Typical „sensors“ available for monitoring the user context:

- Time
- Application/Window manager
- Brightness
- Microphone
- Bluetooth
- Wireless LAN
- Docked / undocked

Other suitable sensors can be connected:

- GPS
- GSM
- Compass
- Accelerometer
- Tilt sensor
- Temperature sensor
- Pressure sensor

Sharing of sensor data between appliances

Content

- Introduction
- Architecture
- Step 0: Sensor data acquisition
- Step 1: Feature extraction
- Step 2: Classification
- Step 3: Labeling
- Step 4: Prediction

Feature Extraction

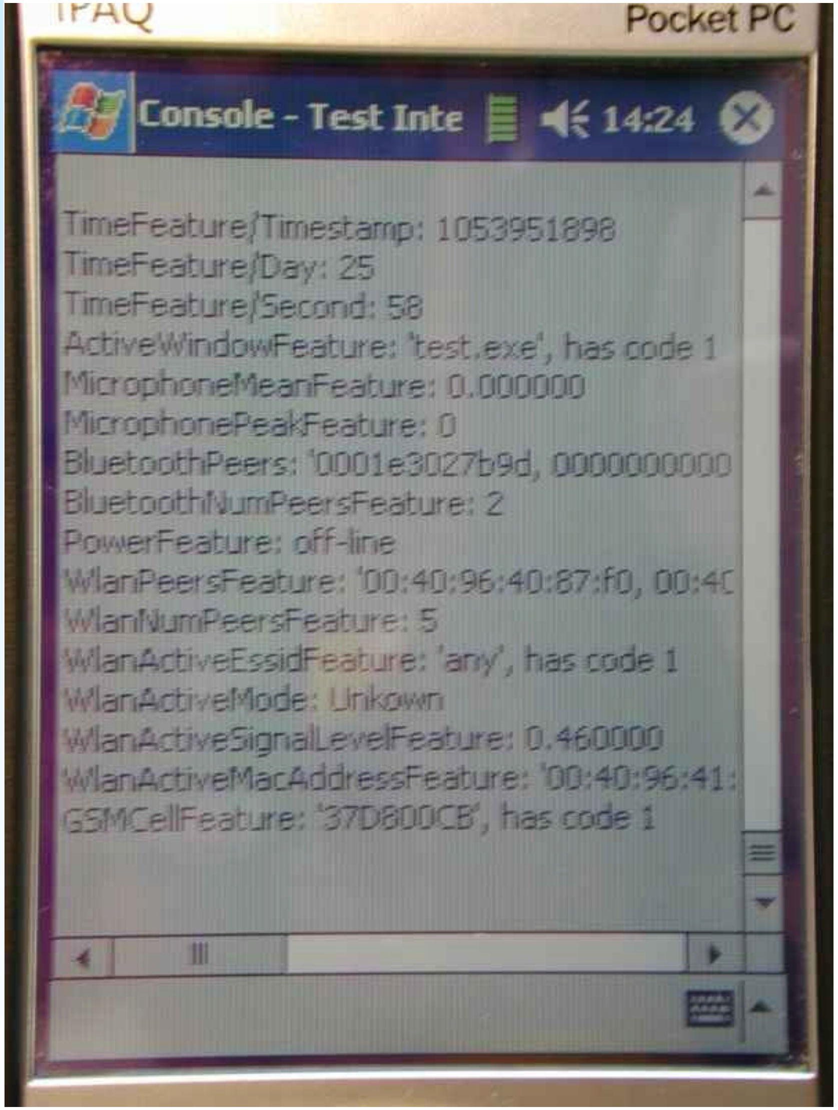
- Raw sensor data is transformed into more meaningful features
- Feature extraction exploits domain-specific knowledge
- Multiple features extracted from a single sensor

⇒ **High-dimensional input vectors**

- Different types of features:
  - Numerical (continuous) sensors: e.g. brightness sensor
  - Numerical (discrete) sensors: e.g. number of access points in range
  - Ordinal sensors: e.g. day of week
  - Nominal sensors: e.g. WLAN-SSID, list of Bluetooth devices in spatial proximity

- Only two operations necessary for each feature:
  - Distance metric
  - Adaptation operator

Feature extraction on a current PDA



- With current computational power easily possible
- Every feature implements distance metric and adaptation operator

⇒ On-line classifications possible with any combination of features (user-selectable and loaded dynamically during startup)

Content

- Introduction
- Architecture
- Step 0: Sensor data acquisition
- Step 1: Feature extraction
- Step 2: Classification
- Step 3: Labeling
- Step 4: Prediction
- First results

Classification: Introduction

- Classifies feature vectors and finds common patterns in sensor data
- Different types of classification algorithms
  - Type (partitioning / hierarchical)
  - Soft / hard classification
  - Supervised / unsupervised

- Requirements for classifying user context in information appliances:
  - On-line learning
  - Adaptivity
  - Variable number of classes and variable topology
  - Soft classification
  - Noise resistance
  - Limited resources
  - Simplicity
  - Interpretability of classes / protection of data privacy

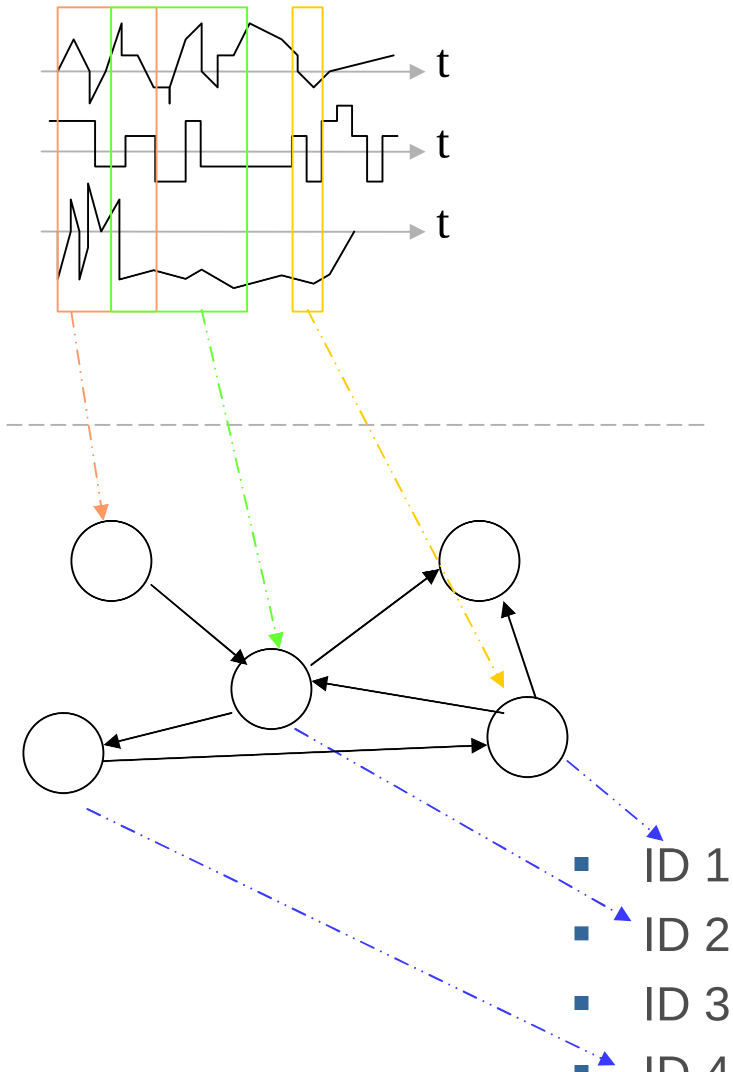
Classification: Algorithms

Algorithm	Network topology	Topology preserving	Competitive
SOM	fixed	yes	soft
RSOM	fixed	yes	soft
K-Means	fixed	no	hard
Leader	variable	no	hard
Growing K-Means	variable	no	hard
Neural Gas	variable	no	soft
Neural Gas + Competitive Hebbian Learning	variable	yes	soft
Growing Neural Gas	variable	yes	soft
Incremental DBSCAN	variable	No	hard

Content

- Introduction
- Architecture
- Step 0: Sensor data acquisition
- Step 1: Feature extraction
- Step 2: Classification
- Step 3: Labeling
- Step 4: Prediction
- First results

Assigning user-defined labels to user context



- 1:{0,1} assignment of (meta-) clusters to context names
- Two possibilities:
  - If the (meta-) clusters at the output of the classification step are stable, direct assignment to names
  - If the (meta-) clusters are unstable due to learning and adaptation, use a second, simple classification step [van Laerhoven, 2001]

- ID 1 = “at work“
- ID 2 = “meeting“
- ID 3 = “lunch”
- ID 4 =
- ID 5 = “at home“

INFORMATIK
UNIVERSITÄT LINZ

Content

- Introduction
- Architecture
- Step 0: Sensor data acquisition
- Step 1: Feature extraction
- Step 2: Classification
- Step 3: Labeling
- Step 4: Prediction
- First results

Prediction of User Context

- Recognized context classes can be regarded as “states“ of an abstract state machine
- Monitoring the state trajectory allows to predict future states

- Prediction algorithm requirements for predicting context “states”:
  - Unsupervised model estimation
  - On-line learning
  - Incremental model growing
  - Confidence estimation
  - Automatic feedback
  - Manual feedback
  - Long-term vs. short-term

- Architecture allows prediction algorithms to be realized as plug-ins

⇒ Algorithms can be changed according to the specific application needs

Content

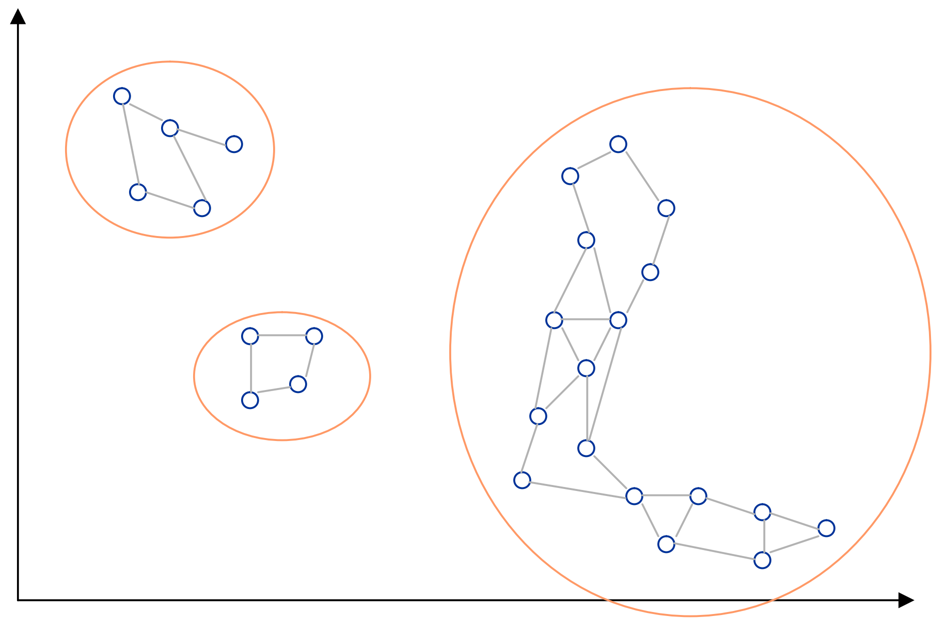
- Introduction
- Architecture
- Step 0: Sensor data acquisition
- Step 1: Feature extraction
- Step 2: Classification
- Step 3: Labeling
- Step 4: Prediction
- First results

First Results

- Test data gathered over 10 days for a selection of features:
  - Timestamp
  - Bluetooth: list of MAC addresses in spatial proximity
  - Bluetooth: number of peers in spatial proximity
  - Wireless LAN: list of MAC addresses in spatial proximity (clients associated to a nearby access point)
  - Currently active application
- Distance metrics and adaptation operators implemented in straight-forward ways (e.g. Hamming distance for lists of MAC addresses)
- Implementation of "Lifelong Growing Neural Gas (*LLGNG*)" as classification algorithm

First Results

- One-pass (online) run through the log file yields (multiple test runs):
  - ~ 40 clusters
  - ~ 6 meta-clusters
  - < 0.12 overall classification error (average distance of input data points to cluster centers)



Summary

- Architecture allows recognition and prediction of user context from raw sensor data in an un-supervised, non-obtrusive way in 4 steps:
  - Feature extraction
  - Classification
  - Labeling
  - Prediction
- Simple interfaces between the architecture layers
- Multiple, simple sensors instead of a single, complex one
- Heterogeneous feature vector handled via specific implementation of:
  - distance metric
  - adaptation operator

  for each feature

- First implementations of feature extraction and classification show promising results

Conclusions and Future Outlook

- Proactive applications could allow the development of more intuitive user interfaces
- A "*Personal Digital Assistant*" and information appliances in general should become proactive to achieve a wider acceptance
- Context awareness is one possibility to achieve proactivity in applications
- Learning user behavior allows to predict future user context and thus future user actions

- Future research on non-obtrusive user interfaces for labeling user context necessary
- Qualitative comparison of prediction methods will be performed using real-world data

It is hard to predict, especially the future

Neils Bohr

Winner of a Nobel Prize for physics

The best way to predict the future is to invent it.

Alan Kay

Inventor of Smalltalk

Thank you for your attention !