Context-Aware Mobile Computing: Learning Context-Dependent Personal Preferences from a Wearable Sensor Array

Andreas Krause, Asim Smailagic, *Senior Member*, *IEEE*, and Daniel P. Siewiorek, *Fellow*, *IEEE*

Abstract—Context-aware computing describes the situation where a wearable/mobile computer is aware of its user's state and surroundings and modifies its behavior based on this information. We designed, implemented, and evaluated a wearable system which can learn context-dependent personal preferences by identifying individual user states and observing how the user interacts with the system in these states. This learning occurs online and does not require external supervision. The system relies on techniques from machine learning and statistical analysis. A case study integrates the approach in a context-aware mobile phone. The results indicate that the method is able to create a meaningful user context model while only requiring data from comfortable wearable sensor devices.

Index Terms—Location-dependent and sensitive, wearable computers, mobile computing, machine learning, wearable AI, statistical models.

✦

1 INTRODUCTION

OUR goal is to make the mobile computer capable of sensing the users and their current state, exploiting context information to significantly reduce demands on human attention. To minimize user distraction, a pervasive computing system must be context-aware [1]. By estimating user context qualities such as location, activity, physiology, schedule, and ambient context information, a human's current state can be determined to a level which allows configuring the settings of a mobile device proactively. Wrong configuration can often lead to frustrating consequences, such as unwanted interruptions or additional work to undo the system's decisions. In addition, mobile computing applications face challenges such as varying resource availability. Any sensor data should be captured by a body-worn sensor array; wearable hardware must provide enough processing capability for data analysis and context inference.

This research focuses on how machine learning techniques can make a mobile computer capable of identifying typical user states and learning to automatically modify its settings based on experience. The general approach is developed and applied in a context-aware mobile phone case study. This study shows that predefined context-sensitive responses are not feasible and that machine learning methods allow individual independent learning of context-aware preferences. This case study is part of the general question of interruptibility assessment in a mobile scenario: Can wearable sensors be used to detect when it is appropriate to request the user's attention?

1.1 Prior Work

In [21], the authors propose a method for unsupervised and dynamic identification of physiological and activity context based on data collected from the BodyMedia SenseWear armband; in [22], they describe an initial approach toward the implementation of SenSay, a context-aware mobile phone. These preliminary investigations are the foundation of the present work.

1.2 Related Work

Microphone and video data have been used in a Wizard of Oz Study [3] to determine when an office worker can be interrupted. Self-reports are used to identify and evaluate a statistical model for interruptibility assessment in this stationary scenario. In [23], office-based sensors such as a silence detector, phone activity, door status, and an analysis of desktop computer usage, including keystrokes and mouse movement, are implemented to evaluate the proposed model.

In [4], accelerometers are used to distinguish different movement activities such as walking, running, and climbing stairs. The presented method relies on offline analysis of labeled data to construct their classifier.

A method based on Kohonen Self-Organizing Maps (KSOMs) and k-Nearest Neighbor classification is proposed in [5] to identify typical motion profiles. This approach relies on active training by pushing buttons while performing certain activities. These sparse labels are used to construct a supervised context transition profile based on a first order Markov process. This approach also suffers from the KSOM's inflexible memorizing process [14]. To make the KSOM training procedure converge, the neighborhood

- *A. Krause is with the Computer Science Department, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213.*
  E-mail: krausea@cs.cmu.edu.
- *A. Smailagic is with the Institute for Complex Engineered Systems, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213.*
  E-mail: asim@cs.cmu.edu.
- *D.P. Siewiorek is with the Human-Computer Interaction Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213. E-mail: dps@cs.cmu.edu.*

radius of the learning neurons must decrease over time. This results in an inability to keep the learning process running over a long time period [9]. In [6], an online algorithm is given for the combination of KSOMs and k-Means clustering. This approach also requires active training by specifying user labels.

Clarkson [2] uses high-level sensors such as video and audio to identify detailed sequential scenes such as entering a building, walking up stairs, and entering an office. This technique requires dedicated offline statistical analysis due to high computational complexity.

In Section 2, we describe our approach. Section 3 explains the machine learning techniques used in our model. In Section 4, the system architecture and prototype implementation for the context-aware cell phone case study are described. Section 5 describes experiments and evaluates results, Section 6 proposes future work, and Section 7 presents conclusions.

2 APPROACH

We believe that the user state is too manifold to be understood and estimated in its entirety by sensor data analysis. In our approach, we consider the user's state to be partially observable: A variety of different context qualities, such as Location, Activity, Physiology, Time, and Ambience, can be estimated to a certain extent by using wearable sensors. On the other hand, we expect that the way a user interacts with mobile devices depends on their current state. In constructing our statistical model, we therefore assume that the user state is a hidden variable acting as a cause both to the values acquired from wearable sensors and to certain interactions with the computer system. It is not our primary concern to identify which sensors are most useful in determining particular states. Instead, our machine learning approach uses statistical analyses to dynamically identify the relation between observed contexts and the user's behavior.

Our notion of a context quality is defined as a physical attribute partially measuring the user's state using sensors at any given point or short interval of time. A cluster within the values of a context quality is called context. Several context qualities can be used in composition to expand the perspective into the user's state. Our approach identifies recurring and statistically relevant locally constant contexts. For example, the context quality "Movement Activity" uses accelerometers to identify typical motion patterns observed during physical activities such as walking, running, or driving. Using the power spectrum of the acquired signal and the dimension reduction techniques described in Section 3.1, motion patterns can be described as locally constant contexts due to the inherent periodicity in human movements, as shown in Section 5.2.1.

2.1 SenSay—A Case Study

Throughout the paper, we motivate, develop, and evaluate our general approach by investigating a case study of SenSay, a context-aware cell phone that modifies its behavior based on its user's state and surroundings. It adapts to dynamically changing environmental, physiological, and activity states and also provides the remote caller information on the current context of the phone user. To acquire context information, SenSay uses several wearable

sensors. The user's preferences we want to predict are modifications of the ringer volume or vibrate alarm and the acceptance or rejection of incoming calls. We assume both of these can be considered preferences dependent on the user's state.

In our first approach, we built and evaluated a prototype context-aware mobile phone by using a predefined user-context model which consists of four states of social importance to cell phone users. These states are: Uninterruptible, High-Energy, Idle, and Normal. The Uninterruptible state was defined by high voice activity, conversation level ambient noise, and infrequent movements, prototypic for meeting situations. The High-Energy state is defined by either high physical activity, i.e., running down a street, or high ambient noise, e.g., loud outdoor environment. The user is considered idle under little motion and voice activity and low ambient noise. These states are accompanied by a default state. We identified a set of wearable sensors and performed the empirical threshold analysis described in Section 5.1 to construct a classifier. For each of these four states, we defined how the phone should react in terms of selecting the attention modality (e.g., vibrate state, ringer tone, and volume) and acceptance of incoming calls.

This approach turned out to be difficult to generalize to larger groups of study subjects since the states and desired phone behavior varied among individuals. As described in Section 5.1, a user survey showed that the preferences of how the phone should react in the aforementioned states varied significantly and there was no agreement on which states needed to be distinguished. The generalization of the threshold values also proved to be difficult. Since a cell phone is a mass-produced device, the outlined problems render the naive preconfigured approach inappropriate.

To overcome these problems, we propose the use of machine learning techniques to allow an automatic personalization both of the state classification and the context-dependent device configuration. Our motivation is derived from the observation that context does not require a descriptive label to be used for adaptability and contextually sensitive response. This makes an approach toward completely unsupervised learning feasible. By unsupervised learning, we mean the identification of the user's context without requiring external supervision, i.e., by manually annotating current user states.

2.2 A Two-Step Approach

Comparison of the quantities to be related, namely, sensor data and observed user interactions, reveals fundamental differences between them. Sensor data can be acquired in abundant quantity without requiring the user's attention. The values are from a continuous range and carry time-series dynamics. On the other hand, user interactions can only be observed sparingly, i.e., when the user deems it necessary to invest their attention. For the SenSay case study, the variables of interest are discrete and assumed to be temporally independent. This bias suggests a two-step approach, as displayed in Fig. 1. To make as much use of the unconsciously accessible sensor data as possible, typical contexts are identified independently for each context quality as well as from user interactions. Unsupervised machine learning techniques are used to realize this context identification. In general, it is difficult to detect the number
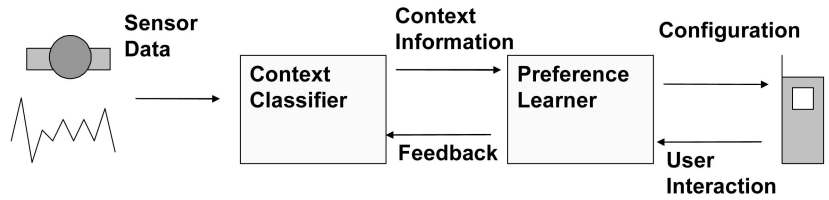
Fig. 1. Our two-step approach gathers data using wearable sensors. A Context Classifier identifies typical contexts and classifies acquired signals. The Preference Learner relates the user's device interactions to their current context and can provide feedback to the Context Classifier.

of classes in the data. This is the challenge of context abstraction. To keep the context identification flexible, the proposed method allows dynamic refinement during extended system use. Once typical contexts have been identified, they can be used to classify acquired sensor data.

At a higher abstraction level, the use of this discretization allows us to model observations about the user as an assignment of discrete values to a collection of random variables: *Context variables* describe evidence about the user's context, and preference, or *system variables*, describes evidence about the user's preferences. Each time a user interaction is observed, it is interpreted as a labeled example which can be used to construct a statistical model for context-dependent preferences. This second step can provide feedback to the context identifier: If similar user preferences are detected for separate contexts, this could be an indication for merging; if the user's behavior is inconsistent within single contexts, this would indicate the necessity to split the context.

This collection of evidence and corresponding model update motivates a Bayesian Statistics approach. Estimating the joint distribution of the described model allows predictions, even with incomplete data. For example, if location data is not available, the remaining sensors might still provide enough confidence to detect an ongoing meeting. This suggests providing certain redundancy in choosing sensors and observed context qualities. Fig. 2 provides the learning strategies involved in our approach.

3 METHOD

In this section, we describe the context identification and the probabilistic reasoning strategy for learning user preferences based on the identified contexts.

To introduce our context identification method, we first present an offline algorithm which uses a training sample time-series as input and constructs a classifier from it. We then describe how we turn our offline algorithm into an online algorithm, which can use real-time sampling data.

Our offline approach can be summarized as follows: After certain data preprocessing, a Kohonen Self-Organizing Map is trained with the sensor values. The KSOM is a vector quantization technique representing the training data with a finite number of codebook vectors, optimally in least-squares sense. Contrary to classical vector quantization, during the training process, the KSOM not only adjusts the best matching unit per sample, but also neighboring units, where the neighborhood is defined by the map topology. This enables the self-organization process. More details can be found in [11]. A k-Means clustering is performed on the map's codebook vectors, where k is chosen as guided by the Davies Bouldin index; a description and analysis of this clustering quality measure can be found in [17]. After the clustering, a first order Markov model is trained with respect to the cluster transition probabilities. A graph reduction strategy is applied to the resulting transition graph. This graph reduction eliminates transient states. Codebook vectors for eliminated states are reassigned to nontransient states by further iterations of the
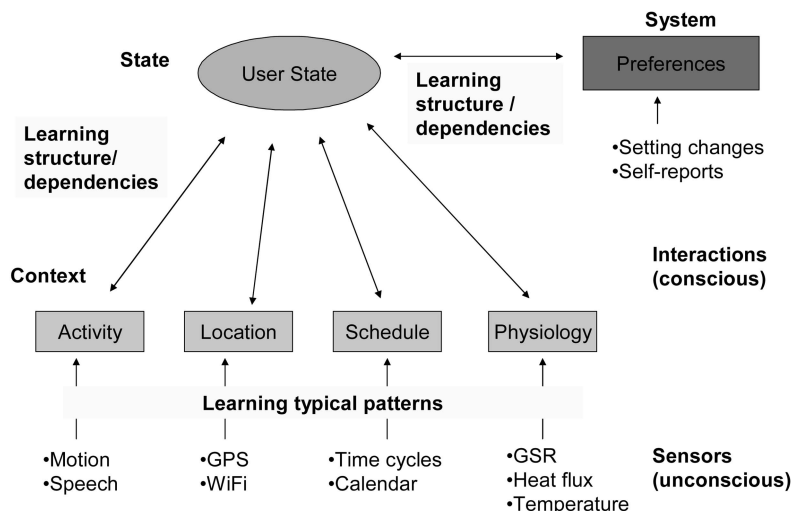

Fig. 2. In our Learning approach, the unobservable user's state causes the user's preferences and manifests in various observable context qualities which independently identify several contexts using a set of sensors. Machine learning is used both to learn typical patterns and automatically discretize the raw sensor data, as well as to relate the user's preferences with the observed contexts.
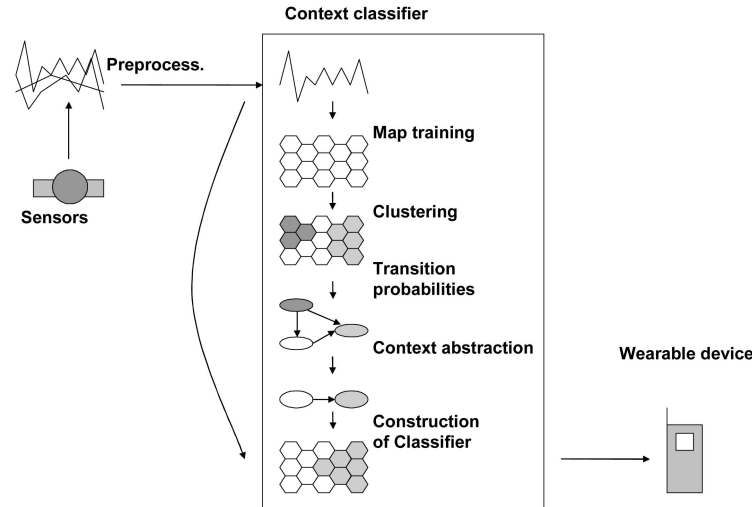
Fig. 3. The offline method uses data from wearable sensors to construct a classifier and to classify current sensor readings. The resulting classification is sent to a context-aware wearable device.

k-Means algorithm. The resulting clustering is used as a classifier for test samples; the first order Markov model on the final clustering can serve for prediction. Fig. 3 gives a graphical illustration of this system.

We proceed with more details on the construction of the classifier utilizing data from the BodyMedia SenseWear armband and the microphones described in Section 4. In the remainder of this section, we describe how our offline algorithm was changed into an online algorithm and provide details on our preference learning strategy. The method described in Sections 3.1 to 3.5 will subsequently be referred to as the Context Identification method.

3.1 Preprocessing

For each individual context quality, a priori knowledge as well as empirical evidence suggests the selection of sensors, sampling rates, and feature extraction. For the physiological sensors, polling at 1 sample per second and using running averages (RA) and between-sample sums of absolute differences (SAD) proved to be useful. Six data channels of the BodyMedia device described in Section 4 have been recorded, forming a six-dimensional feature space. These channels comprised the RA and SAD values from the Galvanic Skin Response, heat flux, and skin thermometer. We also use running averages for the voice and ambient microphone's amplitudes, each resulting in a one-dimensional feature space.

For motion classification, we calculate the Fast Fourier Transform (FFT) power spectra of the acquired accelerometer values. It turned out that a 64 point FFT (i.e., spectra of 8 second time-frames at 8 Hz sampling rate) was a good resolution to discern various different fine-grain movement patterns. Since the spectra are symmetric, only the first 32 points were used. For the high data rate samples, the spectra of the two accelerometers were recorded separately, making it a 64-dimensional feature space. To distinguish between a user's speech and wind noise or whistling, we use 256 point power spectra of the voice microphone sampled at 8 kHz. These 256 point windows are sampled at 1 second intervals. This window size is suggested and

experimentally validated in [16]. To give each dimension equal importance in terms of the self-organization process, we normalized the data to unit coordinate-wise variance as proposed in [9].

Since learning in high-dimensional feature spaces is adversely affected by the so-called curse of dimensionality [10], techniques for dimension reduction were evaluated. Principal Component Analysis (PCA), which is a projection of the data vectors to the most important eigenspaces (in terms of eigenvalue size) of the covariance matrix, showed good results on the accelerometer data. In our experiments, the top five eigenspaces explained most of the sensor variance, so the input data was projected onto the first five principal components. For the audio samples, we calculate a predefined linear transformation averaging over 16 adjacent frequency components to obtain an eight channel subband decomposition, which is normalized to unit-length in order to ignore intensity which is taken into account separately, as suggested in [16]. Table 1 summarizes the various preprocessing steps done for the different sensors.

3.2 Clustering

We use a 20×23 cell hexagonal topology for the map, a size suggested by Kohonen [11] for our data size. Training is performed by the batch training algorithm described in [9]. Although the training of the self-organizing map is already a clustering process, codebook vectors must be further grouped together to reach representative cluster sizes. Therefore, a k-Means clustering is performed on the codebook vectors. Our approach does not require user labels for determining the initial cluster centroid starting values; the clustering is completely driven by the unlabeled data. We initially perform a k-Means clustering with quite large k (growing as the square root of the number of codebook values is a common heuristics). Clustering is performed for the k values lying in a certain interval proposed by this heuristics. The Davies-Bouldin index is used as a further guidance.

TABLE 1
Context Qualities Investigated in Our Self-Report Study

Context Quality	Sensor	Freq	Preprocessing (dimensions)
Activity	Accelerometer (2 axis)	8 Hz	64 pt FFT, 5 comp. PCA (5)
Physiology	GSR, heat flux, thermometers	1 Hz	Run. Avg., SAD (6)
Voice	BlueSpoon	4 Hz	Run. Avg. of Amplitude (1)
VcSpect	BlueSpoon	1 Hz	256pt FFT, norm. 8 channel sub-band decomposition. (8)
Ambience	ECM-T6 lapel microphone	1 Hz	Run. Avg. of Amplitude (1)
Hour	Intern. clock	1/min	-
Location	GPS, 802.11b	20/min	See Section 3.5

3.3 Context Abstraction

By this procedure, about 14 clusters are found for a typical low data rate sample for 20 hours of training data from the SenseWear armband. Since the chosen data preprocessing is expected to smooth out abrupt context changes, clusters representing these context changes are expected to be found (i.e., switching from running to walking and back). Although these transition states are interesting in terms of context change prediction, they adversely affect learning adaptive system behavior since the system must rely on robust context state estimates as the basis for learning usage patterns. To identify transient states, transition probabilities are taken into account. Based on the clustering described above, a first order Markov model is trained using the cluster transition probabilities. Transient states can be characterized by having a low "loop" probability, namely, remaining in the same state for a long time is improbable. A graph reduction algorithm is performed on the Markov model, eliminating states with a "loop" probability (the diagonal values in the transition matrix) below a certain threshold. The corresponding transitive corrections, including the update of the transition probabilities, are implemented as follows: Each state v with a loop probability below a threshold α is removed from the transition graph; 0.3 turned out to be appropriate although results do not show dramatic changes for variations of this value between 0.2 and 0.4. The main step in the algorithm is the following transitivity update,

$$P(r, c) \leftarrow P(r, c) + P(r, v) * P(v, c)/(1 - P(v, v)),$$

which redistributes the probabilities of out-adjacent edges of v to corresponding transitive out-adjacent edges of v's predecessors. P denotes the state transition probability matrix, i.e., $P(i, j) = \mathrm{Prob}(X_t = j | X_{t-1} = i)$, where X_t is the cluster index assigned to the sensor value vector at time t and $\mathrm{Prob}(X|Y)$ is the conditional probability of X for the prior Y.

After the elimination of the transient clusters, a few more iterations of the $k\text{-Means}$ algorithm are necessary to reassign now unclustered codebook vectors to new clusters. Successively, a new transition probability Markov model is trained on the reduced set of clusters; this model can then be used to predict possible context changes.

3.4 Online Algorithm

The algorithm described above is offline, i.e., the entire sensor values time-series must be available at training time.

Our online algorithm is buffer-based and it proceeds as follows: Sensor values are recorded until a certain time interval (i.e., some hours or days, experiments with recorded data used four hour intervals) has passed. This data is used to construct a classifier and context transition probability model as described above. To refine this model, a training sample is constructed consisting of the compressed prior training sample and the newly buffered data. The prior training sample is compressed by selecting a certain percentage of its data vectors at equally spaced intervals. Using these training samples, another map is trained and the new codebook vectors are again clustered by a $k\text{-Means}$ algorithm with the existing cluster centers as initial values plus a certain number of randomly initialized cluster centers. Successively, the steps described above—build a Markov model, do graph reduction, build another Markov model—are applied again. By adding these randomly initialized cluster centers, potentially, new context can be identified. Fig. 4 illustrates the construction of this online classifier.

Since the empirical covariance matrix changes as new sensor data becomes available, PCA can only be performed for fixed sets of data. We consider the principal component directions as properties of the sensors and the sensor placement. Therefore, we expect that data from a limited time interval can determine an initial estimate of the principal component direction. The linear transformation determined by these directions can be used to project new data in real-time to the previously determined principal components. This system of linear equations is solvable if the number of principal components remains the same. Recalibration can be triggered as soon as the amount of variance explained by the principal components drops beneath a certain threshold.

By buffering a certain fraction of the past sensor data, we can still keep track of important past clusters while maintaining the KSOM's ability to learn new contexts. Since the fraction is fixed, the size of the learning data converges. This memory decay strategy is geometric in nature, which is the discrete version of exponential decay. Another advantage of the buffer-based update is that the computationally expensive training and clustering procedure can be implemented in parallel to the data acquisition, classification, and prediction algorithms.

3.5 Online Algorithm for Location Awareness

The context identification algorithm described in Sections 3.1 to 3.4 is used to recognize patterns within the data gathered
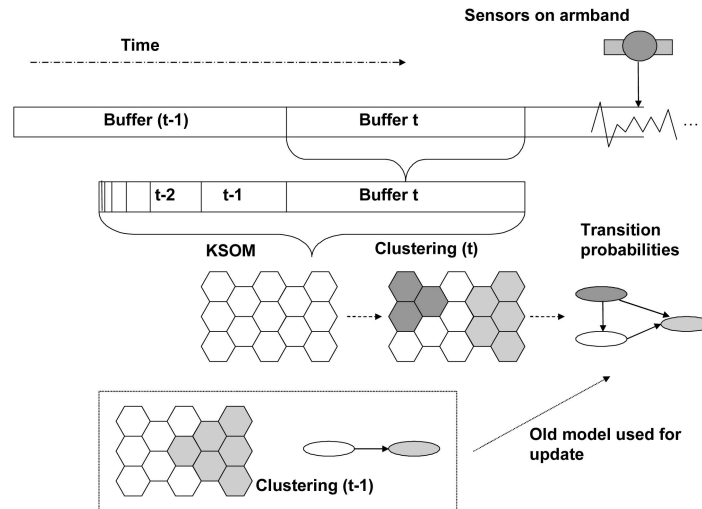
Fig. 4. Online algorithm: The top row represents the buffering of the input data. In the second row, the geometric memory decay is sketched. The third row shows the training of the KSOM, the clustering, the Markov model, and the integration of the prior clustering.

from the sensors integrated in the sensor armband, as well as from the microphones. These sensors provide values in a fixed range which can be assumed constant over time. This algorithm cannot be used for location sensing due to scaling problems, e.g., separate rooms in buildings on different continents should still be separable by the algorithm. Two different, complementary technologies are used for location awareness in the SenSay case study. For outdoor location sensing, the Global Positioning System (GPS) is used. For indoor location sensing, the 802.11b wireless local area network is utilized. Each point in the range of a wireless network can be described as a *signal strength footprint (SSF)*, a high-dimensional vector, where each component is associated with a specific Media Access Control (MAC) address of an access point and the component value is defined by the respective signal level. Due to the limited range of access points, SSFs are sparse vectors. MAC addresses can be associated with their physical location, which is typically stationary. This model is an idealization. Experience shows that signal strength's of adjacent access points fluctuate heavily depending on furniture, network interference, and other environmental influences.

Several research groups have investigated pattern matching and signal strength triangulation algorithm for location sensing in wireless networks. In [18], it is shown that an accuracy of up to five feet can be achieved in the majority of measurements within a large campus-wide wireless network. Although these methods are very precise, they require a large amount of training. To cover the Carnegie Mellon campus wireless network, 1,500 training points were needed for the CMU-TMI algorithm, described in [18]. For the SenSay case study, this was not considered to be a practical solution since the system should be able to use a wireless network as a reference even without prior training. Therefore, in analogy to the method described above, a clustering algorithm is used. Both GPS coordinates and SSFs are classified by their respective cluster indices; if both the wireless network and satellites are accessible, only the GPS coordinates are used for clustering. If the satellite view is lost and no wireless network is available, the last visible coordinates are considered an entrance cluster. Entrances for which the GPS coordinates do not differ by

more than 50 meters are assigned the same index. This allows establishing confidence in a previous location sensing result even if both technologies are unavailable.

The clustering algorithm currently employed solves the scaling problem by accepting the restriction of a specified cluster radius. Upon collection of either GPS data or SSFs, each sample which is not within range of a previously detected cluster is considered a hypothesis for a new cluster center. For both vectors, the Euclidean norm is used as the distance function. For the SSFs, only the five strongest access points which were in range during at least 80 percent of the last n samples are taken into account. Any hypothesis for which 80 percent of n successive samples stay within the predefined radius is accepted as a new cluster and assigned an index, which is then used for classifying new acquired location data.

3.6 Bayesian Networks for Probabilistic Reasoning

To realize the Bayesian statistics approach toward learning context dependent user preferences, an efficient way must be found to construct and incrementally refine the context and system-variable model described in Section 2. This implies the estimation of a joint distribution of many variables. Separating this estimation from our context identification method allows treating all variables as discrete, which has important computational advantages. If no assumptions can be made about the dependence of the random variables, the number of parameters needed to describe the joint distribution can grow exponentially with the number of variables. If all variables are independent, the number of parameters is linear in the number of variables since the joint distribution function can be factored. Independence is usually too strong an assumption, but there is the more subtle notion of conditional independence. Bayesian Networks [20] utilize this notion, which often allows factoring the joint distribution into a number of local terms with a subexponential number of parameters. Given certain data, it is usually a difficult problem even to find a good network structure, i.e., a factorization with as few as possible parameters.

Since a Bayesian Network can be described as a directed acyclic graph (DAG), an optimal structure can be found by

scoring all possible DAGs. Unfortunately, their number is superexponential in the number of variables, so this approach is not feasible. There is a fast algorithm for structure learning called K2 [19], which—given a certain ordering on the variables—greedily adds parents to individual nodes until no further change increases the likelihood of the data given the model. The strongest limitation of this method is that the ordering of the variables needs to be known. This is usually not the case. In our two-step approach, we assume that the user interactions are more directly caused by the user's state. Therefore, we consider system variables possible parents to context variables but not vice versa. Thus, we assume that, in any ordering, all system variables must appear before the context variables. This still leaves uncertainty in the ordering within the two classes of random variables, but assures that any cause-effect relation discovered by the K2 algorithm is consistent with the a priori knowledge we invest in constructing the model. Since we are not dealing with hidden variables in this approach, all parameter estimation and inference can be computed efficiently in closed form. Using our research prototype and the Bayesian Networks for Java (BNJ) toolbox, the K2 algorithm for 100 examples with eight context variables and four system variables executes in less than one second. One reason for this high performance is the low fan-in. Since the system variables we want to predict are fewer than the estimated context variables, the expected fan-in for each variable is smaller than if the observed context was considered as the cause and the user's behavior as the effect.

4 SYSTEM ARCHITECTURE AND PROTOTYPE IMPLEMENTATION

To provide sensor input, the SenSay prototype relies on various wearable sensors. The SenseWear armband from BodyMedia, as shown in Fig. 5, is used to measure activity and physiological context. Further description of wearable sensors can be found in [7]. The small armband combines five distinct sensors: two accelerometers, galvanic skin response, skin temperature, and near-body ambient temperature. The armband is worn on the back of the upper arm. The sensors are described regarding this position, with the wearer standing up, arms at his side.

A two-axis micro-electro-mechanical sensor device measures motion. These accelerometers are oriented in the transverse (through the chest parallel to the ground) and longitudinal (head-to-toe) planes. By taking into account gravity, they measure orientation and motion.

Heat flux is a measure of the amount of heat being dissipated by the body. The sensor uses very low thermally resistant materials and extremely sensitive thermocouple arrays. It is placed in a thermally conductive path between the skin and the side of the armband exposed to the environment.

Galvanic skin response (GSR) uses two hypoallergenic stainless steel electrodes to measure electrical conductivity between two points on the wearer's arm. Skin conductivity is affected by sweat from physical activity and by emotional stimuli. GSR can be used as an indicator of evaporative heat loss by identifying the onset, peak, and recovery of maximal sweat rates.

Fig. 5. Wearable Platform for current SenSay prototype: (A) SenseWear armband, (B) BlueSpoon headset, (C) backpack encompassing Life-Book P-1120 with 802.11b transceiver, (D) Tungsten W smart phone, and (E) external antenna for GPS receiver. The SenseWear armband is worn in contact with the skin and normally hidden from view.

Skin temperature is measured using a highly accurate thermistor-based sensor located on the backside of the armband. It reflects the body's core temperature activity.

The near-body ambient thermometer uses a highly accurate thermistor-based sensor to measure the air temperature immediately around the wearer's armband. This directly reflects the change of environmental conditions around the armband.

The SenseWear is worn on the back of the upper right arm for numerous reasons. Comfort is key, as, during normal use, it is worn day and night, only being taken off to wash. The armband's position and shape make it unobtrusive and quickly "unnoticeable" [8]. Additionally, the asymmetric heat flux sensor gives the most accurate readings when worn in this position.

The armband can record sensor data at specified sampling rates. Each of the integrated sensors is sampled 32 times a second, regardless of the sample interval setting. Two measures of the sensors—the mean value and the sum of the absolute differences (SADs)—are calculated once per sampling period. Sensor readings recorded in the armband are direct values from the internal analogue-digital (AD) converter, in the range [0, 4095].

The armbands also contain a transceiver for live data transmission using the Scientific and Medical Instruments (ISM) license free band. The recharging and docking cradle for the PC contains a corresponding transceiver. A continuous wireless broadcasting method instructs the armband to send out specified sensor readings as fast as possible. The armband stops broadcasting when it is taken off-body and no longer detects contact with skin. Data rates up to 16 samples a second can be achieved with this method.

A Bluetooth headset is used to detect if the user is speaking. The BlueSpoon headset integrates an omnidirectional microphone, a speaker, and Bluetooth transceiver while only weighing 10 g. It acts as a server providing the

Bluetooth Headset Profile. Audio communication is full duplex; the headset is also used to acoustically notify the user about error messages or low battery life. The microphone provides different exchangeable springs to comfortably fit most types of human ears. To detect ambient noise, a Sony ECM-T6 lapel microphone is used as attached to a backpack shoulder strap at the level of the user's chest. The omnidirectional microphone has a frequency response between 50 Hz and 12 kHz.

The algorithms for data processing described in Section 3 run on a Fujitsu P-1120 Life-Book, an ultralight laptop weighing only 2.5 lbs. It is powered by a Transmeta Crusoe processor running at 800 MHz and is equipped with 256 Megabytes of main memory. A BlueTake Poke 2 Bluetooth transceiver is connected to the USB port. The second USB port connects to the armband's wireless receiver. The laptop integrates an 802.11b receiver with an Intersil Prism chipset which is used for indoor location awareness using signal strength triangulation. It is connected to an Electrovaya 160 Wh external battery pack. A TripNav 202 PCMCIA GPS card is used for outdoor location sensing. The laptop and all attached devices are packaged in a comfortable backpack, which has been custom designed for wearability and convenient access to the laptop.

SenSay's user interface is implemented on a palmOne Tungsten W PDA and GSM cell phone hybrid device which provides both voice communication and wireless internet access using GPRS technology. This device uses a Bluetooth transceiver connected to the PDA's Secure Digital Input/Output (SDIO) interface.

Although the sensors presently used cannot be found in any cell phone, they are realistic for the cell phone scenario. Instead of the armband, accelerometers would have to be integrated in the cell phone, as well as an ambient microphone. The headset is a common accessory and PDAs with built in Bluetooth and WLAN are already available. Battery life of all the components exceeds 10 hours, except the Bluetooth headset, which needs to be recharged after 5 hours. It can be placed next to the user and still collect voice data while placed in the cradle.

The software prototype is implemented in Java. An important focus was on modeling extensibility of the framework in order to allow the easy integration of additional sensors and modification of algorithms. Another goal was to prove that the proposed method can run on a computationally weak platform; a highly abstract Java implementation is generally less efficient than highly optimized native code.

The complex sensor fusion setup integrating the various data sources is modeled as a directed acyclic graph; sensors and user interfaces are sources and the learning algorithms act as sinks. The complete setup can be configured using XML. To allow sampling at different frequencies from different sensors, push semantics is used. Events are propagated through the sensor fusion graph with special gating nodes handling timing issues. All events can be streamed into a database which allows playback at different speeds to experiment with various settings for the algorithms. Events can also be distributed over the network to allow convenient maintenance and potentially incorporate stationary sensor infrastructures.

To provide feedback in case of a component failure, the CMU Festival-Lite speech synthesis software (Flite) is used to communicate error messages to the user even while they are wearing the backpack. Fig. 6 displays the system architecture.

5 EXPERIMENTS, RESULTS, AND EVALUATION

In this section, we describe three studies performed to evaluate our approach. In Section 5.1, we first present that the predefined context-aware responses used in the original SenSay application do not generalize to larger groups of subjects, which both a user survey and threshold analysis indicated. We do not claim that pretrained, factory-default context-aware behavior is impossible to realize in general, but, considering our experience, we conjecture that a machine-learning-based approach is the more promising alternative. In Section 5.2, we present results about the Context Identification method. We show that, for our study subjects, the clustering results correspond to manual annotation and that it is possible to use our method for unsupervised construction of a motion classifier. In Section 5.3, experiments are described which evaluate the performance of the Bayesian Networks approach for learning individual user's preferences on top of the Context Identification method. The studies and experiments as well as their results are summarized in Table 2.

To relate the different sets of labels used for the self-report experiments (Sections 5.2 and 5.3), it should be remarked that these self-reports are merely considered to be partial measurements of the user state, manifesting themselves in system variables—analogously to the sensor measurements, which manifest themselves as context variables. They are not short verbal descriptions of the user's state in its entirety. For the definition of context and system variables, please refer to Section 2. In each of the experiments, a different system variable is measured to evaluate how the proposed method performs in predicting its values. Table 3 presents an overview on which system variable is investigated in which experiment.

5.1 Study 1—Motivation of the Machine Learning Approach

To argue in favor of our conjecture that machine learning techniques for learning preferences are superior to a threshold-based approach, we performed two experiments, described in Sections 5.1.1 and 5.1.2. This part of our evaluation should point out difficulties in generalizing the naive threshold-based approach, while, on the other hand, trying to identify a set of factory default states and associated phone behavior. These defaults could be used as a starting point for the machine learning algorithms to accelerate the desired individualization process and overcome instability issues until the learning algorithms stabilize.

5.1.1 User Survey

A survey was performed among college cell phone users to identify a set of states important with respect to cell phone usage. College students have been chosen as our test group since they exhibit ultimate mobility in their use of electronic devices. In [1], they were termed nomads for this reason. Ten students, averaging 21.3 years, were interviewed, each
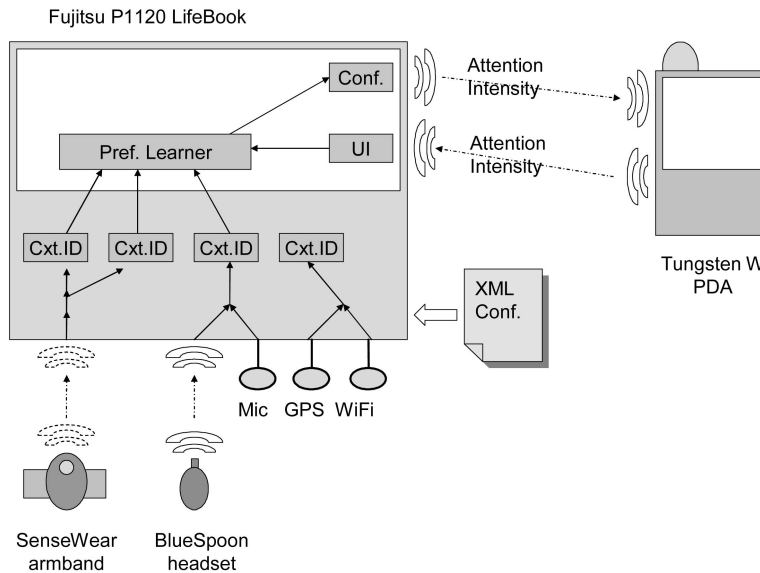
Fig. 6. System architecture is comprised of three major components which are communicating wirelessly: a laptop, PDA, and wearable sensor array. Acquired sensor data are propagated through a preprocessing and sensor fusion graph which is configured using XML.

TABLE 2
Summary of the Conducted Experiments and Results

Sensors	Experiment	Results
Study 1	*Motivation for machine learning approach (Section 5.1)*	
None	Survey college phone users (5.1.1)	Consensus set of states very abstract; varying user preferences for context sensitive response
Accelerometers, voice microphone	Threshold analysis (5.1.2)	Predefined threshold values do not generalize well to larger groups of subjects
Study 2	*Evaluation of context identification method (Section 5.2)*	
Accelerometers, GSR, Heat flux, Thermometer	Self-report study (5.2.1)	Manual annotation corresponds to clustering results
Accelerometers	Movement identification (5.2.2)	Real-time movement pattern identification and classification possible
Study 3	*Evaluation of preference learning method (Section 5.3)*	
Accelerometer, voice microphone	SenSay training (5.3.1)	Training and recalling of context dependent personal preferences possible
All	Self-report study (5.3.2)	Activity and location contexts consistent with self-reports, insight into Bayesian network variable dependencies

interview lasting between 30 and 60 minutes and comprised three main components.

For the survey's first component, we asked the participants to fill in a table describing their daily activity and corresponding phone usage from the previous day. Subjects were given a sample template as a guideline. In the second component, the experimenter reviewed the table together with the participants and probed for additional detail. In the third component, the participants were asked to group their activities using any subjective criteria they deemed appropriate. The rationale of this categorization was then discussed together with the experimenter. Several users categorized their day according to what actions they were taking: working, socializing, eating, etc. Others categorized their day by how they dealt with their phone in each particular activity (phone off, speaking on the phone, charging the phone). Still others

categorized their day according to how interruptible they were during a particular activity.

The number of states varied between four and nine states, averaging 5.8 states. Comparing the 10 collected categorizations, a set of six consensus states was identified, encompassing "Relaxation," "Important work/meeting," "Unimportant work," "Transit/Commuting," "Social," and "Daily routine." For more details on the definitions of these states, see the report in [15]. This experiment indicated that there is relatively high variability in the assessment which abstract states should be detected and on how the phone should react to state changes, even in the narrowly defined group of college cell phone users.

5.1.2 Threshold Analyses

The default states suggested by the survey are very abstract and might be defined differently from person to person.

TABLE 3
Labels Used for System Variables in Different Experiments

Experiment	Context Variables	System Variables	Labels for System Variables
Survey (5.1.1.)	N/A	State annotation (free answers)	Relaxation, Important work / Meeting, Unimportant work, Transit / Commuting, Social, Daily Routine
Threshold analysis (5.1.2.)	Activity thresholds	Movement annotation	Walk, Sit, Run
	Voice thresholds	Acoustic annotation	Silence, Conversation, Music
Self-report (5.2.1.)	Activity, Physiology	State annotation (free answers)	Busy, Commute, Driving, Eating, Fitness, at Home, Social, Meeting, Not Busy, Office, Relaxing, Sleeping.
Movement study (5.2.2.)	Activity	Movement annotation	Walking, Running, Sitting, Kneebends, Waving arms, Climbing stairs
SenSay training (5.3.1.)	Activity, VcSpect, Voice	AttnMode (phone ringer/vibrate setting)	Silent, Vibrate, Low, Medium, High, Maximum
Self-report (5.3.2.)	Activity, VcSpect, Voice, Physiology, Location, Ambience, Hour	Interruptibility	Idle, Available, Busy, Uninterruptible
		Calls	Any, Most, I'd be annoyed, Close friends, Close family, Emergency
		State	Working, Meeting, Relaxing, Walking, Exercising, Eating, Socializing

Proceeding as discussed in Section 2, in the following, we do not make an attempt toward predicting such kinds of abstract states, but rather describe an experiment to distinguish simple, well-defined contexts. Eleven more subjects, including undergraduate and graduate students as well as research faculty, were asked to perform various activities while data was recorded from the microphones and motion sensors. The activities comprised of: sitting, walking, running, silence, conversation, and listening to music.

Two threshold analyses to differentiate between low, medium, and high physical activity were performed in the first test. The two-axis accelerometer integrated in the BodyMedia armband was used to capture movement intensity, while 11 subjects were asked to perform the following test. After walking for 40 seconds, they were asked to sit down for 10 seconds. They were then asked to run for 30 seconds and walk again. After walking for 20 more seconds, they sat down again for an additional 25 seconds, concluding the experiment. Thresholds for differentiating between low, medium, and high activity were found and annotated on Fig. 7.

In the second test, the running average of the BlueSpoon wireless microphone's amplitude was analyzed to identify a threshold for a binary classifier constructed to detect whether the user is speaking. After 50 seconds of silence, the study subjects were engaged in a conversation with varying intensity. Fig. 7 shows the results of this experiment. A threshold value was found and annotated in the diagram.

Although appropriate threshold values could be found valid for most subjects, the constructed classifier can only distinguish very simple activity contexts and might be heavily influenced by environmental properties such as high ambient noise. Depending on different anatomy of the subjects and wearing position of the sensors, it turned out that, even for these simple contexts and limited number of subjects, threshold values were hard to find—i.e., medium-high thresholds conflicted with low-medium thresholds of other participants. The identified thresholds can still be used as a starting point for the machine learning algorithms proposed to accelerate the initial learning phase. This phase is assumed to be critical for the acceptance of the proposed application in mass-produced devices such as cell phones.

For other context qualities, especially for location or schedule, it is clear that no threshold-based approach can work for all users.

5.2 Study 2: Evaluation of the Context Identifier Method

To investigate our unsupervised machine learning approach toward context identification, we performed two experiments.

In the first experiment, two subjects wore the BodyMedia armband over the duration of several days, personally annotating their state—i.e., working in the office, being in a meeting, commuting, relaxing in the sun—by use of the armband button and additional note-taking. Subject A is one of the authors, a graduate student in his mid-twenties leading a moderately active lifestyle. Subject B is a graduate student in her mid-twenties, leading an active lifestyle. The subjects were required to mark time stamps close to their personally felt state change and verbally classify the corresponding state. In summary, over 240 hours of studies were done by recording one sample per minute (low data rate). For both subjects, the labeling and granularity of manual annotation corresponded to the clustering results and granularity of the automatic classification. The granularity is consistent with the results of Section 5.1.1.

The second experiment investigates the performance of our classifier to determine motion patterns. The movement identification data set was sampled at eight samples per second. All data samples were recorded at specified sampling rates on the armband's onboard storage. As an improvement upon related work on motion classification, our results indicate that the models can be viably constructed in an unsupervised manner. Additional results, such as influence of the geometric buffer size on the clustering performance, have been published in [21].

5.2.1 Self-Report Study

In this experiment, Subject A collected data over four days for an overall recording time of 74 hours and subject B gathered data during daytime for 22 days with a total recording time of 168 hours. Both subjects wore the SenseWear armband on the back of their right upper arm. The collected data were comprised of the running average of
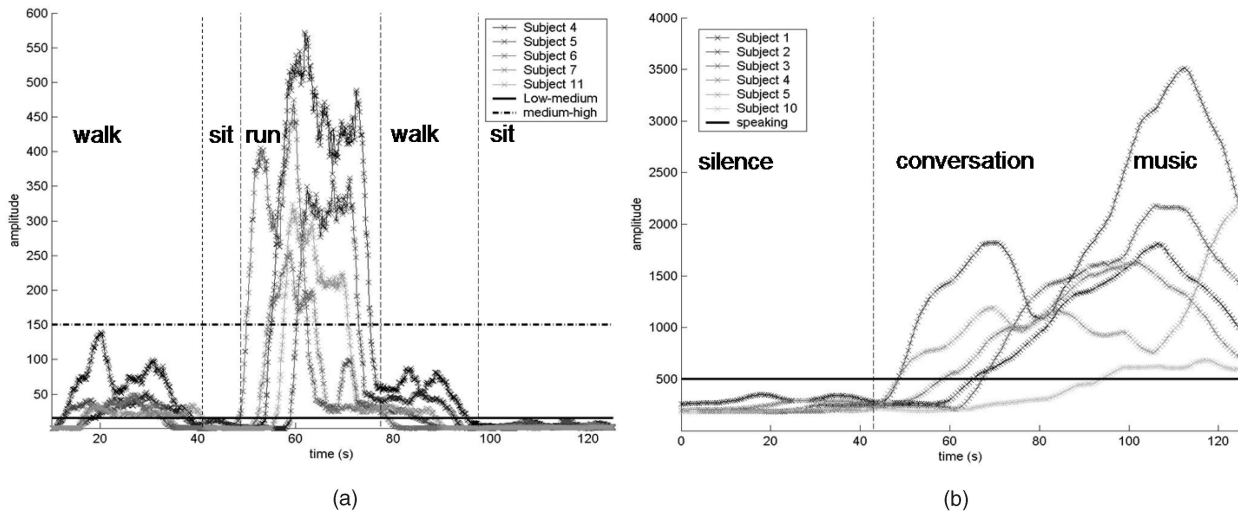
Fig. 7. Identification of threshold values for (a) low-medium and medium-high motion and for (b) silence and voice activity. A representative subset of the collected data is shown.

the GSR, heat flux, both thermometers, and both accelerometers as well as the SAD values of both accelerometers. The sampling rate was set to one sample per minute. Data was analyzed in blocks of varying lengths, including a variety of different contexts. An initial clustering with, on average, 15 clusters was done per block. After the elimination of transient states, an average of five clusters was identified for subject A, who continuously wore the device for four days, and an average of four clusters was identified for subject B, who had varying recording time ranges. All recordings were done during weekdays.

The data was processed by our classifier and the resulting clustering was manually compared with the data annotation. It was examined whether equally annotated states correspond to single or separate clusters. Fig. 8 displays the performance of our method on a 20 hour data sample. Our method could compete with the subjective context classification. The study shows that—consistent with the results described in Section 5.1.1—both subjects only identified a small number of personally typical states; the number of states annotated is only slightly higher than the number of clusters found by our classifier. The frequent use of the armband's timestamp button by subject B to subjectively annotate context changes indicates the existence of a number of minor contexts, e.g., short interruptions of the current activity which were not deemed important enough to annotate and accounting for too little data to justify the creation of new clusters due to the one sample per minute sampling rate. The number of state transitions determined by our classifier closely matches the number of timestamps taken by both subjects, so the nonidentified clusters might have been eliminated by a too strict context abstraction threshold as described in Section 3.2.

Table 4 shows the results of the first experiment. Column 1 is the sample number, column 2 denotes recording duration and involved days, column 3 lists abbreviations for major contexts identified by wearer, column 4 shows the number of clusters found, column 5 gives the absolute number of time stamps, and column 6 shows the number of stable (only one transition per five minutes) transitions. Column 7 gives

the offline map training quantization error, i.e., the average squared distances between the training vectors and their best matching codebook vectors. The data from the experiment was input to the offline and online algorithm which showed comparable classification performance for the number of clusters and the transition locations.

5.2.2 Movement Identification

A movement study at eight samples per second was used to test the quality of our method to discern fine grain movement patterns. Subject A performed six distinct kinds of movements over a period of about three minutes (1,632 samples in total), in the following order: walking, running, walking, sitting, knee-bends, walking, waving his arms, walking, and climbing up and down stairs (one stair a step). Six clusters were identified, clearly corresponding to the trained movement patterns. Projection of adjacent values of the sensor time-series onto the map exhibited close locality while the subject performed movements corresponding to a certain activity.

The map constructed by the offline training algorithm was evaluated by testing its performance as a classifier on test data, which consisted of recorded movement patterns of four minutes, 2,204 samples in total. Movements were recorded in the following order: walking slowly, knee-bends, running, climbing up and down stairs (two stairs a step), walking, and sitting. All movements were correctly classified. The algorithm was very successful even in discerning the transition from running to climbing stairs; the classification of walking slowly was wrong only in the first few seconds, when it was misclassified as corresponding to the "sitting" cluster. Since the FFT is calculated from an eight second time frame, the transition lag varied between 0 and 4 seconds, 1 second being the average classification lag.

5.3 Study 3: Evaluation of the Preference Learner

To assess the performance of our Bayesian Networks approach toward learning context-dependent personal preferences, we conducted two experiments, described in next two sections.
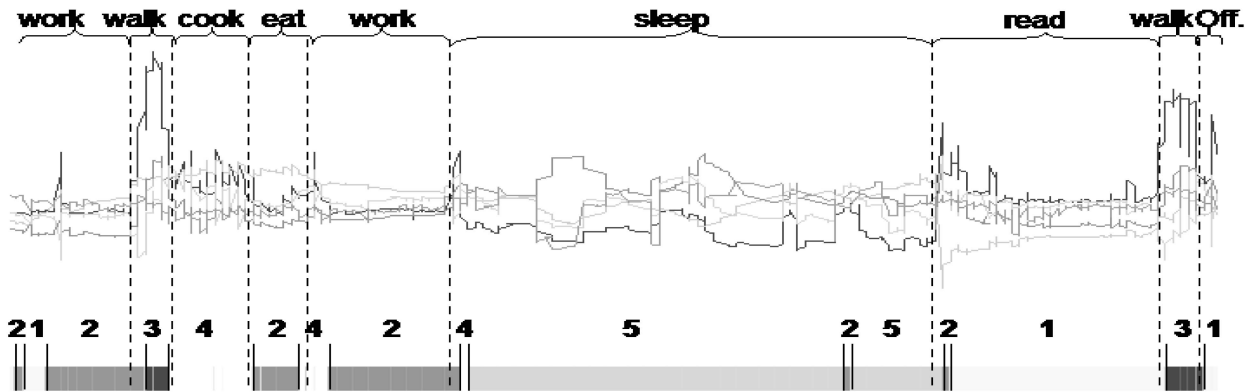
Fig. 8. Analysis of the clustering—annotation correspondence. The graph shows the five first principal components of a 20 hour data sample with manual annotations (top row) and clustering results (bottom row and numbers). Each shade of gray relates to a cluster and the black lines denote cluster transitions.

TABLE 4
Results of the Context Identification Experiment

Sample	Size	Annotated Contexts	Clusters	Timestamps	Transitions	Quant. Error
A-1	20h / 2d	C,E,H,O,S	6	9	11	0.5
A-2	25h / 2d	C,E,F,H,O,R,S	6	9	14	0.3
A-3	29h / 2d	C,E,H,O,S	5	8	17	0.4
B-1	57h / 6d	B,C,H,M,O,S	4	26	35	0.6
B-2	17h / 3d	M,O	2	20	25	0.5
B-3	26h / 4d	C,M,N,O,S	4	18	24	0.6
B-4	22h / 3d	C,D,E,L,O,M	4	25	27	0.8
B-5	46h / 5d	C,E,L,M,O	3	37	35	0.9

Abbreviations used for manual annotations: B - Busy, C - Commute, D - Driving, E - Eating, F - Fitness, H - at Home, L - Social, M - Meeting, N - Not Busy, O - Office, R - Relaxing, and S - Sleeping.

In the first experiment, the BodyMedia armband sensor was accompanied by the BlueSpoon headset in order to train SenSay to react appropriately in a variety of scenarios. For five participants, all graduate and undergraduate college students, training and testing data were collected as described in Section 5.3.1, and the time required to achieve a clear distinction among several context dependent user preferences was measured.

In a second experiment, three subjects wore the complete hardware platform described in Section 4, each for several hours. Participant A is one of the authors, the same as for the Context Identification study, and wore the system for three days and an overall data collection time of 25 hours. Participants C and D, both female graduate students in their mid-twenties, wore the platform for 8 and 5 hours, respectively. The experiment was an experience sampling technique [13], sometimes referred to as a beeper study. The PDA asked the subjects to answer three multiple choice questions at irregular time intervals and recorded the time it took the subjects to answer the questionnaire. This experiment shows that, even during short system usage, for all subjects, activity and location contexts could be identified consistent with their self-reports. Furthermore, it provides insight into the structure of the Bayesian networks obtained.

5.3.1 SenSay Preference Training Test Cases

In this experiment, different context-dependent preferences as listed in Table 5 were taught to the SenSay implementation, only using the armband and Bluetooth microphone. The system was trained by the participants, who repeatedly performed different activities, such as walking, working at the computer, doing driving movements, and talking while they were wearing the system. Within about one hour of training, enough examples were given to reliably distinguish the activities "Working," "No motion," "Walking," and "Driving" as well as "Talking" and "Silence." The examples realistically varied in intensity, shape, and pace. At that time, the "Activity" context quality had identified four clusters; the "Voice" context quality describing the running average of the voice amplitude had identified three clusters according to different intensities of the user's voice and the "VcSpect" context quality mapped into two different labels. These cluster counts are averaged across the subjects. Afterward, the preferences were trained by successively giving examples for all the rows listed in Table 5 and changing the *AttnMode* system variable, namely, the phone's attention intensity setting which comprises the values "Silent," "Vibrate," "Low," "Medium," "High," and "Maximum," as depicted in Fig. 9.

After a further 30 minutes of training and 48 assigned labels (six per case), the model state was frozen to evaluate its performance as a classifier. This training consisted of clicking on the PDA's attention intensity selection buttons, which are displayed in Fig. 9, while performing the same activities which were used to train the context identification method. The mobile computer was instructed to initiate a preference change on the PDA every three seconds if the value assignment surpassed a probability of 70 percent.

Within another 30 minutes, the participants tested the system online by providing examples for all test cases,

TABLE 5
Example Preferences Trained in Study 3

Test case	Activity	Audio	Phone action
1	Working	Talking	Silent
2	Working	Silence	Medium
3	No motion	Talking	Silent
4	No motion	Not talking	Medium
5	Walking	Talking	Low
6	Walking	Not talking	Maximum
7	Driving	Talking	Vibrate
8	Driving	Not talking	Low

Fig. 9. User interface on the PDA displaying the automatically configured setting of the *AttnMode* system variable.

performing three trials separate from the training data. For each scenario, the time was taken until the desired preference change was initiated. If the right setting was not assigned within 20 seconds or the correct assignment did not remain stable for at least 60 seconds, the attempt was counted as a failure; otherwise, it was counted as successful. Table 6 shows the results, no failure occurred during the experiment.

Fig. 10a displays the structure of the learned Bayesian Network model. From this training study, the Attention Intensity setting apparently was described best by both the activity and the voice spectrum, which has been used to distinguish between talking and whistling. For visualization of the transition graphs, the GraphViz package [12] was used. This experiment indicates that—at least for the five participants—the system is able to reliably determine contexts and learn preferences online, without supplying any pretrained model.

5.3.2 Self-Report Study

To analyze how the addition of more context variables such as Location, Physiology, the hour of the day, and ambient noise influence the structure of the Bayesian Network, we performed a beeper study. The first question asked was: "What is your current interruptibility?" The available answers were: "Idle," "Available," "Busy," and "Uninterruptible." This question was asked first to get the conscious classification of the user's interruptibility as close to the actual point of interruption as possible. This consideration is based on observations mentioned in [3], which suggests that people experiencing an interruption undergo a certain task change which might forfeit precise measurements with increased delay. To normalize different individual interpretations of interruptibility and to more specifically target the cell phone case study, a second, closely related question was asked: "Which phone calls would you accept in your current state?" The allowed answers had a more fine-grained
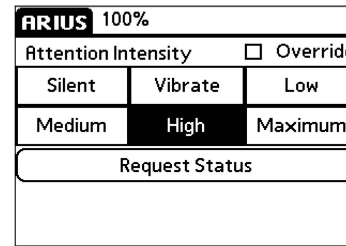
scale, encompassing "any calls," "most calls," "most calls, but I'd be annoyed," "only calls from close friends," "only calls from close family," and "only dire emergency calls." To also collect a conscious annotation of the user's state, we asked: "What is your current state?" The participants were required to choose from "working," "meeting," "relaxing," "walking," "exercising," "eating," and "socializing." The selection of this set of states was motivated by self-reports obtained from experiments de-scribed in Section 5.2. Questions were asked at random time intervals every 8 to 15 minutes, averaging to 30 labels per 5 hour recording. Although [3] suggests using voice recording for collecting the self-report to guarantee minimal intrusiveness and quicker response, having to get access to the PDA more closely resembles having to accept or decline a phone call on a cell phone.

Several hours of data were recorded to prove that important activity and location contexts can be learned and consistently related to self-reports even during short system usage. A "walking" state was identified for all subjects and always consistently related with their state self-report. For Subject B, 13 indoor and two outdoor locations were learned. The recording spanned both working and social hours; two components, each consisting of several indoor locations and separated by external locations, attributed to this fact. During working hours, interruptibility annotation appeared to be mostly dependent on the location, where one location seemed to be associated with a meeting room. For Subject C, seven locations were learned, one related to 78 percent of the recording, which appeared to be the office; self-report mostly indicated a working state and the call acceptance was mostly set to "close friends."

For subject A, nine indoor locations and four outdoor locations have been learned during 25 hours of data collection. Of the indoor locations, one is clearly attributed to the subject's office, assigned to over 80 percent of the recording time. Five other indoor locations were assigned each to at least 0.7 percent which is about 8 minutes of the

TABLE 6
Results of Recalling the Scenarios from Table 5

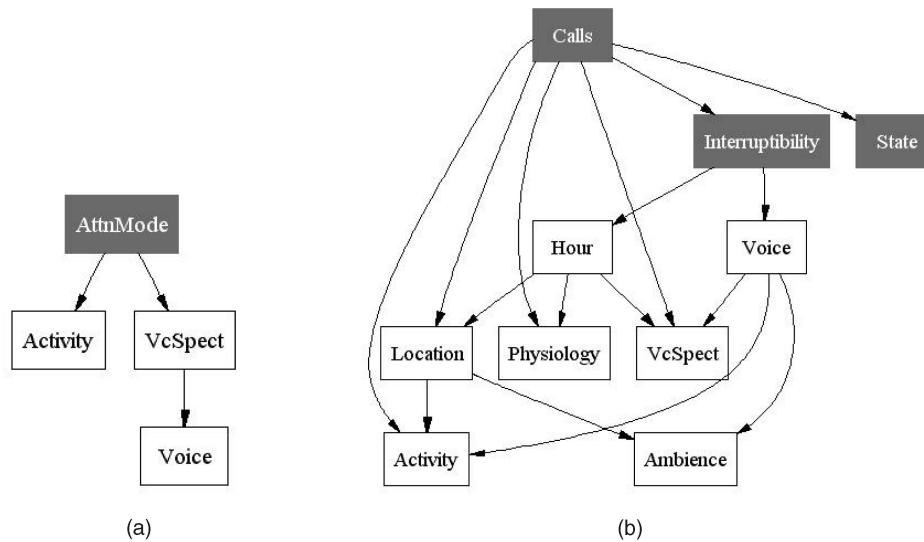| Test case	1	2	3	4	5	6	7	8
Participant 1, avg. time (s)	9.2	14.0	6.5	8.5	10.2	11.0	8.2	9.0
Participant 2, avg. time (s)	8.3	9.3	5.3	8.3	7.3	6.0	7.3	4.0
Participant 3, avg. time (s)	7.6	6.3	9.6	10.0	13.3	8.3	11.0	8.3
Participant 4, avg. time (s)	11.3	12.0	4.6	10.6	12.3	13.3	17.0	14.0
Participant 5, avg. time (s)	12.0	7.6	11.3	11.3	8.0	11.6	9.0	7.6

Fig. 10. (a) The Bayesian Network structure obtained during the SenSay training experiment and (b) a 25 hour self-report data set. The nodes refer to the context variables enumerated in Table 1. The edges were determined by the K2 structure learning algorithm and encode the dependencies between the different context and system variables as described in [20].

collected data. Apart from the office location, the subject spent about 2 hours in the institute's conference room and an additional hour in a separate meeting room which both have been assigned indoor locations. The remaining three locations are associated with not more than one minute of recording and are apparently caused by signal strength fluctuations in the wireless network. Of the four outdoor locations, one is associated with the campus food trucks, one with a place were the subject had lunch, and one with a bench in front of the institute on which the participant had coffee. A fourth location is only related to one minute of recording time and was probably learned while the participant was briefly talking to some person on his way back from lunch before entering the institute's building again. Fig. 10b shows the graph structure obtained from 176 self-reports of Subject A. Analysis of the graph structure obtained from the K2 structure learning algorithm suggests several observations since edges are only introduced if the model change significantly increases the likelihood of the data given the model [20]. Apparently, the "Calls" variable indicating the user's willingness to accept incoming calls is dependent upon many context qualities and closely related to the other two queried variables "Interruptibility" and "State." The close relation to the state annotation indicates that the user's state greatly influences the decision to accept or decline incoming phone calls. On the context variable side, this preference depends most closely on the user's movement activity (Activity variable) and the voice spectrum (VcSpect variable), analogous to the results obtained from the SenSay training experiments. Additionally, the location and physiology contexts play an important role in deciding which incoming calls to accept.

6 FUTURE WORK

To prove the ability to achieve personalization in the described method, extensive studies are planned, involving more subjects than in our current self-report study. On the

modeling side, we are also experimenting with more complex Bayesian Network structures; the introduction of hidden variables might give a better understanding of the relations between the context and system variables. The use of Dynamic Bayesian Networks would allow capturing more complicated patterns of user interactions since they do not assume temporal independence among the observed evidence. The mobile computing challenges, such as limitation of computational power, impose restrictions on the available algorithms which need to be explored in more depth.

Furthermore, we intend to investigate the proposed method in applications other than the cell phone scenario; among them are an application for personal health and wellness monitoring and an application mediating between the importance of a request, the user's interruptibility, and the modality which is used to request the user's attention.

7 CONCLUSIONS

We have presented a method for learning context-dependent personal preferences from wearable sensors and evaluated our approach in a cell phone case study. We developed, implemented, and evaluated a system which can identify a set of context states without requiring supervision, based on data from a comfortable wearable sensor armband, microphones, and location sensors. Our improvements in wearability and usability for context-awareness are advances toward making context-aware wearable computers feasible for real life applications.

REFERENCES

[1] A. Smailagic and D.P. Siewiorek, "Application Design for Wearable and Context-Aware Computers," *IEEE Pervasive Computing,* vol. 1, no. 4, pp. 20-29, Dec. 2002.

[2] B. Clarkson, "Life Patterns: Structure from Wearable Sensors," PhD thesis, 02, MIT Media Lab, 2002.

[3] S.E. Hudson, J. Fogarty, C.G. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J.C. Lee, and J. Yang, "Predicting Human Interruptibility with Sensors: A Wizard of Oz Feasibility Study," *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI 2003),* 2003.

[4] R. DeVaul, M. Sung, J. Gips, and A. Pentland, "MIThril 2003: Applications and Architecture," *Proc. Seventh Int'l Symp. Wearable Computers,* 2003.

[5] K. Van Laerhoven and O. Cakmakci, "What Shall We Teach Our Pants," *Proc. Fourth Int'l Symp. Wearable Computers,* 2000.

[6] K. Van Laerhoven, "Combining the Kohonen Self-Organizing Map and K-Means for On-Line Classification of Sensor Data," *Artificial Neural Networks—ICANN 2001,* G. Dorffner, H. Bischof, and K. Hornik, eds., pp. 464-470, 2001.

[7] J. Farringdon, A.J. Moore, N. Tilbury, J. Church, and P.D. Biemond, "Wearable Sensor Badge & Sensor Jacket for Context-Awareness," *Proc. Third Int'l Symp. Wearable Computers,* pp. 77-83, 1999.

[8] F. Gemperle et al., "Design for Wearability," *Proc. Second Int'l Symp. Wearable Computers,* pp. 116-122, 1998.

[9] T. Kohonen, *Self-Organizing Map,* second ed., pp. 127-128. Berlin: Springer-Verlag, 1995.

[10] R. Bellman, *Adaptive Control Processes.* Princeton Univ. Press, 1961.

[11] T. Kohonen, "Things You Haven't Heard about the Self-Organizing Map," *Proc. Int'l Conf. Neural Networks (ICNN),* pp. 1147-1156, 1993.

[12] E.R. Gansner and S.C. North, "An Open Graph Visualization System and Its Applications to Software Engineering," *Software—Practice and Experience,* vol. 30, no. 11, pp. 1203-1233, 2000.

[13] L. Feldman-Barrett and D.J. Barrett, "Computerized Experience-Sampling: How Technology Facilitates the Study of Conscious Experience," *Social Science Computer Rev.,* vol. 19, pp. 175-185, 2001.

[14] R.M. French, "Catastrophic Forgetting in Connectionist Networks," *Trends in Cognitive Science,* vol. 3, no. 4, pp. 128-135, 1999.

[15] N. Moraveji and A. Smailagic, "Analysis of Interruptibility and User States Among College Students Using Mobile Phones," technical report, Inst. of Complex Engineered Systems, Carnegie Mellon Univ., 2003.

[16] M. Stäger, P. Lukowicz, N. Perera, T. von Büren, G. Tröster, and T. Starner, "SoundButton: Design of a Low Power Wearable Audio Classification System," *Proc. Seventh Int'l Symp. Wearable Computers,* Oct. 2003.

[17] D.L. Davies and D.W. Bouldin, "A Cluster Separation Measure," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 1, no. 2, pp. 224-227, 1979.

[18] A. Smailagic and D. Kogan, "Location Sensing in a Context-Aware Computing Environment," *IEEE Wireless Comm.,* Oct. 2002.

[19] G. Cooper and E. Herskovits, "A Bayesian Method for the Induction of Probabilistic Networks from Data," *Machine Learning,* vol. 9, pp. 309-347, 1992.

[20] D. Heckerman, "A Tutorial on Learning with Bayesian Networks," *Learning in Graphical Models,* M.I. Jordan, ed., pp. 301-354, Dordrecht: Kluwer, 1998.

[21] A. Krause, D.P. Siewiorek, and A. Smailagic, "Unsupervised, Dynamic Identification of Physiological and Activity Context," *Proc. Seventh Int'l Symp. Wearable Computers,* Oct. 2003.

[22] D.P. Siewiorek, A. Smailagic, J. Furakawa, A. Krause, N. Moraveji, K. Reiger, J. Shaffer, and F.L. Wong, "SenSay: A Context-Aware Mobile Phone," *Proc. Seventh Int'l Symp. Wearable Computers,* Oct. 2003.

[23] S.E Hudson, J. Fogarty, C.G. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J.C. Lee, and J. Yang, "Examining the Robustness of Sensor-Based Statistical Models of Human Interruptibility," *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI 2004),* submitted.

Andreas Krause graduated in computer science and mathematics from the Technische Universität München. He is now a doctoral student in the Computer Science Department at Carnegie Mellon University, where he is investigating a machine learning approach toward context-aware mobile computing. His research interests include machine learning theory, clustering methods, probabilistic reasoning, and efficient algorithms with applications to wearable computing and biomathematics.

Asim Smailagic is a research professor at the Institute for Complex Engineered Systems, the College of Engineering, and the Department of Electrical and Computer Engineering at Carnegie Mellon University (CMU). He is also the director of the Laboratory for Interactive and Wearable Computer Systems at CMU. This lab has designed and constructed more than two dozen generations of novel mobile/wearable computer systems over the last 12 years. He received the Fulbright postdoctoral award at Carnegie Mellon Univeristy in computer science in 1988, has been a program chairman of seven IEEE conferences, and was program chairman of the Cambridge Conference on High Performance Distributed Computer Systems '89. He has acted as guest editor and associate editor in many leading archival journals, is one of the cofounders of the IEEE Symposium on Wearable Computers, and is a new chair of the IEEE Technical Committee on Wearable Information Systems. He was a recipient of the 1992 Tempus European Community Award for scientific cooperation resulting in new curriculum development, the 2000 Allen Newell Award for Research Excellence from Carnegie Mellon's School of Computer Science, the 2003 Carnegie Science Award for Excellence in Information Technology, and the 2003 Steve Fenves Award for Systems Research from CMU's College of Engineering. He has written or edited several books in the areas of computer systems design and prototyping, mobile computers, and VLSI system design. His research interests include pervasive context aware computing, wearable computers, human centric computing, and rapid prototyping of computer systems. He has made major contributions to several projects that represent milestones in the evolution of advanced computer systems: from CMU's Cm* Multiprocessor System and Edinburgh Multi-Microprocessor Assembly (EMMA) to CMU's current projects on wearable computer systems, smart modules, Communicator, Aura pervasive computing, and the RADAR Personal Cognitive Assistant. He is a senior member of the IEEE.

Daniel P. Siewiorek received the BS degree in electrical engineering from the University of Michigan and the MS and PhD degrees, both in electrical engineering, from Stanford University. He is now a professor of electrical and computer engineering at Buhl University, a professor of computer science at Carnegie Mellon University, and director of the Human-Computer Interaction Institute. He helped to produce the Cm* multiprocessor system and contributed to the dependability design of 24 commercial computer systems. He has published more than 450 technical papers and eight textbooks. In 1981, he was elected as an IEEE fellow for contributions to the design of modular computer systems and, in 1988, he received the Eckert-Mauchly Award for his contributions to computer architecture and was elected as a member of the 1994 Inaugural Class of ACM fellows. He was elected a member the National Academy of Engineering in 2000. His memberships include the IEEE Computer Society, ACM, Tau Beta Pi, Eta Kappa Nu, and Sigma Xi.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.