A Survey of Prediction Approach in Pervasive Computing

Darine Ameyed, Moeiz Miraoui, Chakib Tadj

Abstract— Context awareness is one of the fundamental principles underpinning pervasive computing. Context prediction, a new trend in pervasive computing, is an open-ended research topic with a lot of challenges and opportunities for innovation. This work presents and analyses the development in this area and compares different context prediction techniques and approaches.

Index Terms — prediction, proactive adaptation, context-aware system, pervasive computing, ubiquitous system, adaptative service.

—————————— ◆ ——————————

1 Introduction

IN a pervasive system, we must provide service adaptation, depending on the system's environment and the user's profile in order to offer an adequate service. In the literature, adaptation is also called plasticity [1]. It determines the adaptation degree of a given application to new requirements and new situations. While using ubiquitous systems, users might perform tasks that are not part of their daily schedule. They might use various systems (smart phone, tablet, personal digital assistant (PDA), personal computer (PC), mobile or otherwise) depending on the nature of their activities, their preferences and other considerations. Users might make use of several modalities (keyboard, mouse, screen ...) to input data, depending on their profiles and the nature of the physical environment.

Furthermore, a ubiquitous system must provide the user with services well adapted to the overall context. Indeed services will be triggered dynamically and without an explicit user intervention in a proactive way. Making use of the context in applications, is a current area of research known as "context-awareness" [2]. A sensitive-context application must perceive the context of the users and their environment and adapt its behavior accordingly.

Most of the work on service adaptation in context-awareness, is focused on the current context. However, a new trend in this research field, takes into account the evolution of the context through time, in order to predict a future context [3]; [4]; [5]. Therefore, the future context allows the pervasive system to choose the most effective strategies to achieve its goals, and makes it possible to provide an active and fast adaptation to future situations. This raises our interest in context prediction and its integration in proactive service adaptation. In this paper, we present a survey of prediction methods, used in proactive adaptation for pervasive systems. We give an overview of the available research in the field; we analyze, and discuss these approaches against a set of criteria.

The remainder of the paper is structured as follows:

•First, we give a short background and introduce some basic concepts.

•Second, we present the various methods illustrated in some projects and cited in the literature, before analyzing them against our evaluation criteria matrix.

•Finally we present a summary with the different challenges yet to overcome in the field.

2 Proactive Computing

2.1 Proactive Context-Aware Applications

So far, most of the context-aware systems are merely reactive, adopting decisions based only on the current context. Research in anticipatory and proactive systems, including prediction of future situations is still in its beginnings. In these systems, a user is linked to a profile with a set of defining characteristics. Leveraging that profile, allows us to improve the systems' efficiency by anticipating the user's needs and adapting the services accordingly. Several approaches have been proposed to achieve this anticipatory aspect [3]. These approaches can be classified under two categories: passive or active:

a. Passive context-awareness

The passive context-awareness system is a reactive system that requires the user to be reactive. The system presents the new or updated context, as a recommendation, to an interested user. The adaptation is completed upon the user's reaction. System decisions are up to the user, who agrees or not on implementing the recommended adaptation [6]; [7]; [8].

b. Active context-awareness

An active context-awareness system is a proactive system. It automatically adapts to discovered contexts and changes the application behavior accordingly. Adaptation is fully automated and rely on the system itself to learn or acquire information allowing the improvement of its performance with experience [9]; [5].

2.2 Proactive Adaptation

In a system that supports context-awareness, adaptation is generally presented as the ability of the system to alter its behavior on its own, in order to improve its performance or pur-

• *Darine Ameyed is currently pursuing Phd degree program in engineering in ÉTS (École de technologie supérieure), MMS Laboratory, Canada. E-mail: darine.ameyed.1@ens.etsmtl.ca .*
• *Dr.* Moeiz Miraoui is currently Professor in Computer Science Department in Umm Al-Qura University, Saudi Arabia. E-mail: mfmiraoui@uqu.edu.sa.
• Dr. Chakib Tadj is currently Professor in Electrical Engineering Department, École de technologie supérieure, MMS Laboratory director, Canada. E-mail: ctadj@ele.etsmtl.ca.

sue ongoing tasks, in different environments.

Various types of adaptation can be differentiated: (i) content-based adaptation, (ii) behavior-based adaptation, (iii) presentation-based adaptation (interface), (iv) software components adaptation.

Adaptation has been mentioned in several occurrences of research work, and garnered interest among researchers for some time: [10];[11];[12];[13] [14];[15];[16]. Adaptation is still of interest in the pervasive computing field.

Service adaptation occurs when a change in the system's environment triggers a special event. Control of adaptation can be, manual from the user, automatic or generic automatic.

a. Manual: the user decides the system's behavior based on given recommendations.

b. Automatic: the adaptation routine is implemented within the system. This routine, is mainly based on a static set of conditional structures. This means, when a given input hits one of the expected states implemented within the conditional structures, the system must change its behavior appropriately. This also means that an unexpected input will go unnoticed by the system. And no change in behavior is triggered.

c. Generic automatic: the adaptation routine is generic. Which means the system can respond to unexpected events and refine and improve its behaviour over time.

2.3 Proactive System Challenge

Adaptive services, supporting dynamic environments, raise different challenges. The environment and the set of situations, a user might evolve through are described in the context. This context specifies the interaction between user and system. A question in this respect, still open, is how to find means of perceiving this context. The currently proposed solutions rely on the use of various sensors to capture environmental and user-related data. This generates heterogeneous data from a variety of sources. And makes services adaptation more and more complex to identify and model.

Another challenge is the extraction of knowledge such as the ability to recognize, identify, design knowledge relevant to the adaptation process. This challenge can be presented as a creation of intelligence: analysis of context information, deduction of meaning and construction of knowledge.

System adaptability entails fault tolerance, recovery from failure and also the ability to discover new resources and devise new strategies to respond to unexpected input.

The availability of user data through all the processes underlying context awareness raises some concerns about the user privacy and brings another challenge in this respect.

Finding operating solutions in real time are also another point which biases the interest of researchers in the field. Finally, another is to offer architectural models open and flexible.

These challenges are among the main research topics in ubiquitous computing. And this is where prediction comes into focus since it addresses some of them. In this paper we will survey relevant work in this respect.

3 Context Prediction

"A system is context-aware, if it uses context to provide pertinent information and / or services to the user, where relevance depends on the user's task" [17]. Also, we might define context-awareness of a system as its ability to make use of the context to improve its overall functioning. Context-awareness is one of the fundamentals of ubiquitous computing. Reasoning about the context is the process of acquiring new information from the context. Therefore, prediction of the context means the ability to predict the future context information in order to provide proactive service. Later in this section, we shall present the most frequently used methods to predict the context.

3.1 Context Prediction Methods

3.1.1 Sequence Prediction Approach

This method stems from the prediction task, as defined in theoretical computer science. The general idea is, a problem could be described as sequence of events or states received in the time interval from 1 to t. The goal reduces then, to predicting the next set of events or the next state (Figure.1).

Davison and Hirsh were amongst the first researchers to investigate the usefulness of sequence-based prediction. They used it to predict commands in a UNIX shell[18]; [19]. This work inspired research in activity prediction, including research in pervasive computing and smart environment [20]; [21];[22];[3]. Starting from the LZ87 algorithm [23] Bhattacharya and Das developed different context prediction algorithms. They modified the original LZ87 algorithm to make it predict the next users' location in a cellular network, yielding the LeZi-Update approach. Later Das et al used this updated algorithm for the smart home environment [21]. LeZi-Updated inspired Cook et al.to develop the Active LeZi-Updated algorithm to predict the user's actions in a smart home. In their Project MavHome, Cook et al built a smart home management system acting on users' behalf and support them in different tasks. The system evaluates the next users' action and triggers a set of actions to fulfill the underlying tasks (e.g opening store, starting the lawn sprinklers) [22].

Mayrhofer used Active LeZi to develop a general context prediction approach, in order to proactively assist users. The prediction process comes in 5 steps: (i) collect context information from a heterogeneous sensor network. (ii) extract features. (iii) classify. (iv) label, and (v) predict. As shown in Figure.2. The process takes as input a times series (observation sets, each recorded at a specific time) and provides as output the user's current context and the expected future context.

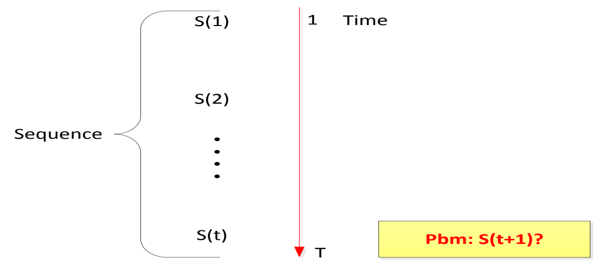


Fig.1. Sequence prediction approach

The context acquisition step concerns itself with gathering contextual information from physical and logical sensors. The captured data is then transformed into more meaningful features during the extraction step for better interpretation. The classification step then, recognizes recurring patterns, called clusters in the feature space, built in the previous step. Classification uses a feature vector which can be assigned to multiple clusters with different degrees of belonging. This degree represents the probability that a feature vector belongs to a cluster. Labeling allocates the descriptive names for individual clusters or combinations of several clusters. Based on the observed history, the future context cluster can be predicted. The prediction step uses the vector of clusters generated by the classification step. The final goal is to generate a cluster of vectors for the future, which matches the current cluster provided by the classification step. This classification is used for supplying the cluster vectors predicted in the labeling step to provide context label, intended for use in dynamic applications.

The sequence prediction proved to be an effective approach for context prediction. As shown previously, several researchers have contributed theoretical improvements to adjust it to the context prediction problem. It should also be noted, that sequence prediction has several shortcomings. Among those mention:

▪The reliability of prediction can be verified but the reliability of the observed data cannot be taken into consideration.

▪Abstracting the whole context into a simple sequence might result in the loss of information.

▪This approach does not deal with time: it cannot detect time-dependent patterns.

▪This approach can only be applied if the context is described as a flow of events.

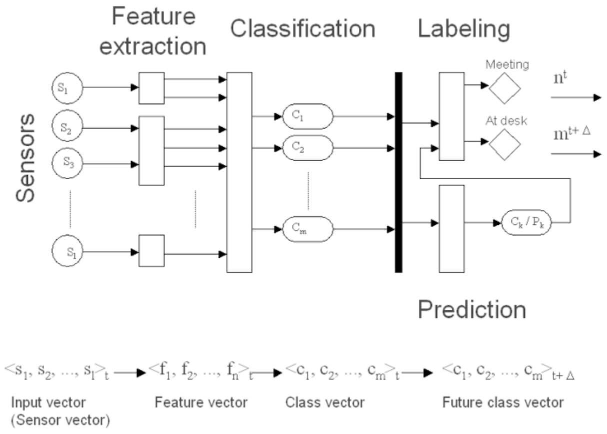


Fig. 2. Architecture for proactivity via predicted user contexts[3]

3.1.2 Markov Chains

A Markov chain is a set of states. The process starts in one of these states and moves incrementally from one state to another. Each move is called a step. If the chain is currently in state Xi, then it moves to state Xj at the next step with a probability denoted by pij, and this probability does not depend on the previous states.

{Xn, n > 0} Xn is the state being processed at time n

pij(n) is called transition probability from state i to state j at time n.

$$P[X_{n+1} = j \mid X_n = i_n, ..., X_0 = i_0] = P[X_{n+1} = j \mid X_n = i_n] = p_{ij}(n)$$

Prediction based on Markov chains is fairly widespread. Some projects used this technique for resolving context prediction problems. For example, Chen [24] used Markov chain to predict a person's movement. Zukerman [25] used them to predict the next user's request in a web context.

In the work of Kaowthumrong et al [26], the authors tackled the problem of active devices determination. The proposed solution was to provide the user with the next device to be used, when there are several devices in the vicinity of user. Based on a Markov property, the next device depended on the current device only. The context prediction process was built on a Markov chain representing the order of devices access. Krumm used discrete time Markov chain for route prediction during driving [27].

Despite some limitations of Markov chains, (limited number of states over a discrete timeline), they are a viable solution, easy to implement. But more complicated use cases need extension of the Markov chain. One of the most famous of these, is the Hidden Markov Chain (HHM) introduced by Baum and Petrie in 1966. This model can be used with unobserved (hidden) states.

Figure.3, outlines the basic layout an HMM. A circle maps to a random variable that might contain any sequence of values. x(t) represents the hidden state as a function of the time t. y(t) represents the observation as a function of the time t. The arrows in the diagram (often called a trellis diagram) denote conditional dependencies.

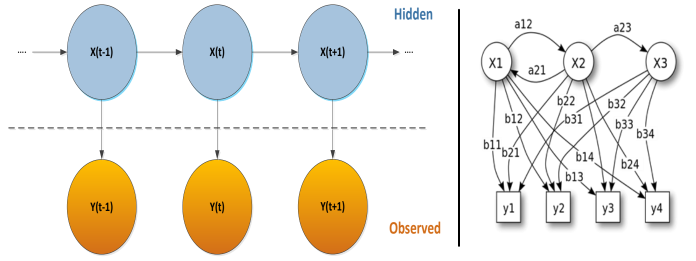


Fig. 3. General architecture of an HMM.

- X : states
- y : possible observations
- a : state transition probabilities
- b : output probabilities

From the diagram, the conditional probability distribution of the hidden variable x(t) at time t, given the values of the hidden variable x at all times, depends only on the value of the hidden variable x(t − 1): the values at time t − 2 and before have no influence. This is called the Markov property. Similarly, the value of the observed variable y(t) only depends on the value of the hidden variable x(t) (both at time t).

The HMM was used in some projects for context prediction. We mention Gellert and Vintan, [28], who used the HMM to predict the next room that user want to visit. Simmons et al, used the HMM for itinerary prediction [29].

One more extension of Markov chain is the Markov Decision Process (MDP), with the addition of actions (allowing choice) and rewards (giving motivation)

A MDP is a formal model that consists of the following (Figure.4)

- S is a finite set of states
- A is a finite set of actions (alternatively, A_s is the finite set of actions available from state S)
- $P_a(s,s') = \Pr(s_{t+1}=s' \mid s_t = s, a_t = a)$ is the probability that action a in state S at time t will lead to state s' at time t+1,
- $R_a(s,s')$ is the immediate reward (or expected immediate reward) received after transition to state s' from state S,
- $\gamma \in [0,1]$ Is the discount factor, which represents the difference in importance between future rewards and present rewards.

The MPD was used in location prediction (e.g: driving). For example Zeibart proposed PROCAB (Probabilistically Reasoning from Observed Context-Aware Behavior) for road navigation to predict route and destination based on the user behavior [30].

The Markov chain provides a simple system overview and may be applied if the context is decomposed into a finite set of not superposed states.

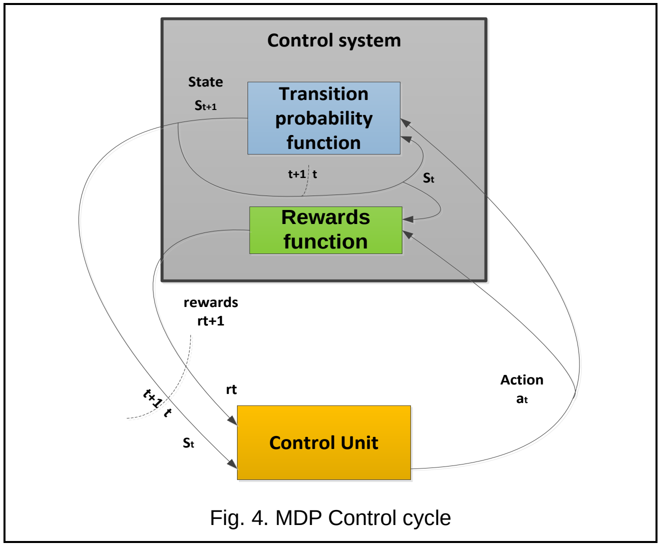


Fig. 4. MDP Control cycle

3.1.3 Bayesian Network

A Bayesian network (BN) is a probabilistic graphical model (a type of statistical models) that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG). The structure of a DAG is defined by two sets: the set of nodes (vertices) and the set of directed edges. The nodes represent random variables and are drawn as circles labeled by the variable names. The edges represent direct dependence among the variables and are drawn by arrows between nodes. Every node is associated with a fact and every directed edge represents the influence on one fact by another.

A Bayesian network specifies a joint distribution in a structured form. It represents dependence/independence via a directed graph composed with (i) Nodes (random variables) and (ii) Edges (direct dependence).

There two components in a Bayesian network, the graph structure (conditional independence assumptions) and the numerical probabilities (for each variable given its parents), which In general is:

$$p(X1, X2,....XN) = \Pi\ p(Xi \mid \text{parents}(Xi\ )$$

A Dynamic Bayesian Network (DBN) is a special Bayesian Network which relates variables to each other over adjacent time steps.

A DBN represents all variables at two (or more) points in time. Edges are drawn from the variables at the earlier time to those at the later time. For example, in the DBN on the above figure (Figure.6), we see that A at time t influences C, at time t+1, B influences for A and C.

Several researchers work on how to make use of DBN in context prediction, like Albrecht's [31]. The system, he developed, used DBN to predict further user actions in an adventure game. With a similar method, Horvits built a system to predict the goal of software users and run services proactively [32]. Another example is how Petzold used the dynamic Bayesian network to predict a person's movement inside some defined space [33]. Petzold represented the context as a DBA, where the current location depends on several locations previously visited and also the time spent at each location. This model has provided high prediction accuracy (around 70% to 90 %).

The BN approach can be seen as a generalization over the Markov models. A BN cannot model a feedback loop because by definition it cannot have loops, or cycles, in its simple forms. But this limit can be overcome with DBN. Indeed the DBN offers more flexibility but requires more training data.

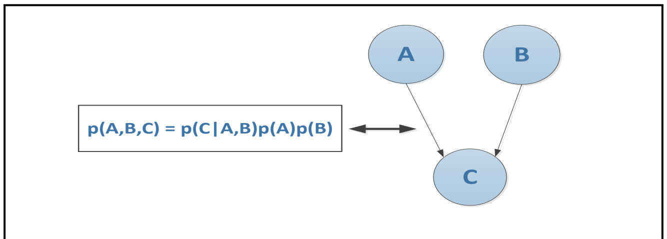


Fig. 5. Example of Simple Bayesian Network

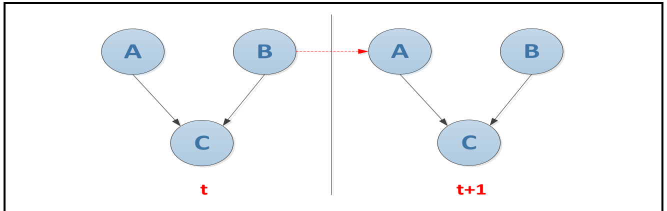


Fig. 6. Example of Dynamic Bayesian network

3.1.4 Neural Networks

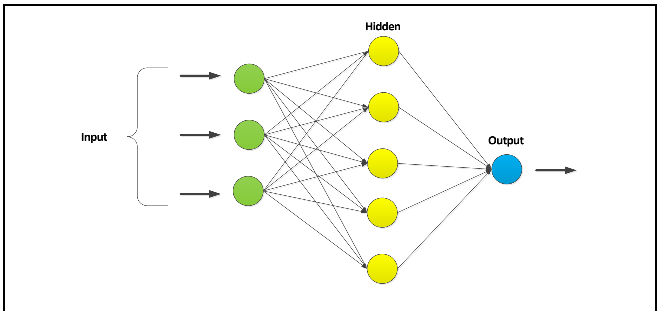


Fig. 7. Simplified structure of an artificial neural network

Neural networks are inspired from their biological counterparts. They are very popular in machine learning. And they implement formal mathematical models emulating the structure of biological neurons. Each node represents an artificial neuron and each arrow represents a connection from the output of one neuron to the input of another (Figure. 7)
This approach has become one of the very popular ones, to solve several problems pertaining to artificial intelligence such as: pattern recognition, association models, approximation theory, etc.
Neural networks have been used, first, to predict context through the work of Mozer, [34]. This work described a smart home with various services, such as space occupation models, estimates of hot water consumption rates. Vintan et al used this same approach to predict the location of an individual, the places he's most likely to visit, in a given environment [35].
The work of Al-Masri [36] also resulted in a model of neural networks to provide mobile services. Recently, Lin [37] used neural networks for a smart intracellular handoff of work.
There are several types of neural networks. Approaches based on them are flexible and efficient. However these approaches act like a black box, have no tolerance for data missing, are slow in their training phase (require long periods of time).

3.1.5 Branch Prediction Method

This approach, predicts the flow of instructions in a microprocessor, after a branching command is received. The purpose of the branch predictor is to improve the flow in the instruction pipeline. Branch predictors play a critical role in achieving high effective performance in many modern pipelined microprocessor architectures. To predict whether a branch is taken or not-taken, the branch prediction unit implements a hardware algorithm. There is a variety of branch prediction algorithms:

Static prediction: the idea is to make a distinction between the different types of branches with greater probability of being taken and those which will never or almost never be taken. The simplest static prediction algorithm makes a distinction between conditional branching and unconditional branches: an unconditional branch is always taken; conditional branches are never taken.

Saturating counter: it consists in keeping track of the number of times that branch has been taken, and then comparing this number with the total number of branch executions. Thus an average can be estimated to see if the branch is usually taken or not taken. This storage is based on counters, integrated into the branch prediction unit. There is a counter per branch stored in the branch target buffer. This counter is changed each time a branch is taken or not-taken: if it is taken, the counter is incremented (unless it is already at its maximum value); if it is not, its value is decremented.

Two-level adaptive predictor: if there are three ''if'' statements in a code, the third ''if'' statement might be taken depending upon whether the previous two were taken/not-taken. In such scenarios, a two-level adaptive predictor works more efficiently than saturation counters. Conditional jumps that are taken every second time or have some other regularly recurring patterns, are not predicted well by the saturating counter. A two-level adaptive predictor remembers the history of the last n occurrences of the branch and uses one saturating counter for each of the possible 2n history patterns. The general rule for a two-level adaptive predictor with an n-bit history is that it can predict any repetitive sequence with any period if all n-bit sub-sequences are different.

Global branch prediction: Global branch predictors make use of the fact that the behavior of many branches is strongly correlated with the history of other recently taken branches. Those ways they can keep a single shift register updated with the recent history of every branch executed, and use this value as an index in a table of bimodal counters.

Neural branch prediction*:* The conditional branch prediction is a Machine learning problem, where machine learns to predict the outcome of conditional branches. Artificial neural network is very efficient to learn, to recognize and classify patterns. Similarly the branch predictor also uses the classification of branches and tries to keep correlated branches together
Some context prediction systems used Branch Prediction Algorithms. Starting from this approach Petzold [38] provided a context prediction algorithm for a ubiquitous information system, in order to discover the daily movement pattern of an individual in a finite environment such as a house or an office. This system used several kinds of predictors; a counter predictor, a two-level-adaptive predictor and a global predictor.

This approach is simple and fast but it can detect only a very simple range of behaviour patterns.

3.1.6 Trajectory Prolongation Approach

Some context prediction approaches, handle the context data vector, from a multidimensional space perspective. Then the underlying algorithm, computes an approximation or an interpolation with the points given through the data vector, with some related function in order to predict the next set of events. This approach has been used in the work of Karbassi and Barth [39], which attempted to estimate the arrival time of a vehicle going through a traffic jam. The system built a linear regression between the traffic jam factor and the arrival time. Then, it computed the parameters from the vehicle's history. Also the work of Mayrhofer [3]; Sigg [40] suggested the use of an autoregressive model to predict the context. In Sigg's system the prediction model was based on an alignment method for predicting the most probable continuance of a time series from the observed sequence of suffix. This alignment prediction approach seeks to align two series using their differences so that the number of positions corresponding to the two series is maximized. Finally Sigg [41] offered a continuous learning module in order to adapt to changing environments or user habits, through continuous monitoring of a time series recorded in in the context's history and updates of the registered patterns.

The trajectory prolongation approach is an efficient approach for location prediction. But it has significant shortcomings: it can't handle non numerical data, and has no tolerance for missing data.

3.1.7 Expert System

An expert system is a tool, able to reproduce the cognitive processes of an expert in a particular field. Expert systems are designed to solve complex problems by reasoning about knowledge, represented primarily as if–then rules. An expert system is divided into two sub-systems: (1) the inference engine and (2) the knowledge base. The knowledge base represents facts and rules. The inference engine applies the rules to the known facts to deduce new facts (Figure 8).

This approach was based on expert systems and rule engines. It has been used in some implementation of context prediction. The end goal is, to build the prediction rules. This method allows a clear view of the system. An example is the work of Vainio, which presented a proactive context adaptation system for smart home environments. The system controls the state of an adaptable context (e.g: lighting, temperature…) according to the user's routine. The system adapts the context under the user's supervision. The system is based on fuzzy logic techniques, allowing it to control the environment in uncertain situations.

Another example is the work of [42]. They developed an expert system that infers rules in the execution's environment to find user preferences and provide related proactive services, based on their respective context history. The main challenge in this work is to deduce the user preferences for the current state from past states, and then figure out the corresponding service.

Context prediction based on expert systems is a promising approach allowing quick and efficient integration of previous knowledge. It also allows an easy integration of the adaptation action. This approach can ensure learning and self-correction.

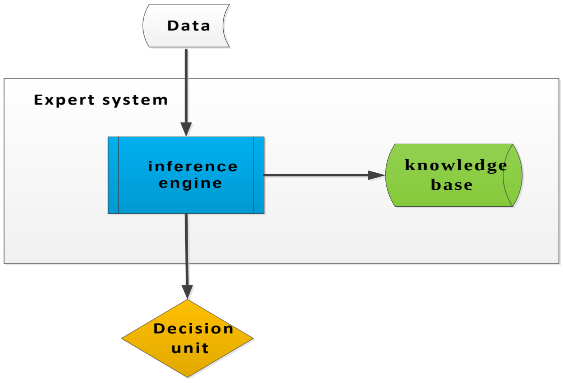


Fig. 8. Expert system structure.

3.1.8 Context Space Theory

Contextual spaces theory is an approach developed, in order to best define context-awareness and to deal with sensor problems that create uncertainty and incur a lack of reliability. This theory uses spatial metaphors to represent the context as a

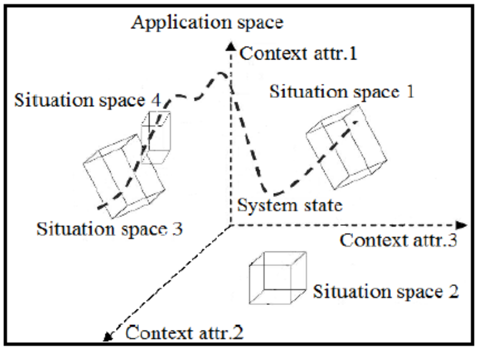


Fig. 9. Context spaces theory [43]

It is designed to make context-awareness clearer. In the space theory, any kind of data is used to reason about the context, is called a context attribute. Attributes are either directly measured by sensors or calculated from other context attributes. They may have a numeric value or a non-numeric value. A context state is represented by the entire relevant context at a specified point in time. All possible context states, define the application space. This allows seeing the application space as a multidimensional space, where the number of dimensions is equivalent to the number of context attributes in the context state.

The theory of context space, initially submitted by Padovitz and Zaslavsky [44]. The authors attempt to provide a general model to help thinking and describing the context and develop context-aware applications. This work will be later, the basis for several researches of Zaslavzky and Boytsov [45]; [46]; [43]. Boytsov and Zaslavsky present the CALCHAS system, which offers context prediction, used extension to the context space theory to provide proactive adaptation. The authors see proactive adaptation as a reinforcement to learning tasks that aim at improving both the prediction as well as the adaptation decision [43].

This approach addresses the context prediction problem in a general sense.

3.1.9 Data-Mining Algorithm

Data mining or the Knowledge Discovery in Databases (KDD), an interdisciplinary subfield of computer science, is the computational process of discovering patterns in large data sets, involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining is to extract information from a data set and transform it into an understandable structure for further use. Data mining involves six common classes of tasks:

- Anomaly detection: the identification of unusual data records.
- Association rule learning: searches for relationships be-

tween variables.

•Clustering: is the task of discovering structures in the data that are similar, without using known structures in the data.

•Classification: is the task of generalizing known structures to apply to new data.

•Regression: attempts to find a function which models the data with the least error.

•Summarization: provides a more compact representation of the data set, including visualization and report generation.

Several methods are used for knowledge discovery in a large amount of data. Clustering is one of the most used methods. It uses unsupervised learning algorithms by similarity measures to group data elements with similar patterns (in the sense of these similarities) in clusters. In a context discovery work Y. Benazzouz, [47] presented a method based on clustering to measure common situations, according to a context clustering approach in the MIDAS project. He adopted a bottom-up discovery (In this approach the situation to discover, is defined beforehand) of common situations and identification of unusual situations to provide a proactive adaptation service according to user behavior.

This approach requires a database archiving user history. Data mining requires a first phase of collection to get enough data which will be processed later. In fact, that generates a significant amount of data to store and manage.

3.1.10 Similarity and Semantic Similarity

This approach uses existing relations between context values in order to deduce the emerging context. The relationships in a context are of two types: relationships between context values and relation between multiple context values. The relationship between context values, identify two entities belonging to different ontologies based on the similarity between them. This similarity can be between two atomic properties (if they have the same name and the same value of property), or between two classes or two non-atomic properties (if they have the same name or any property defined within one, is similar to a property in the other). Based on this similarity, four relationships can be identified between two context values, namely: intersection, complement, equivalence and independence. The relationships between multiple context values are described using an Entity-Relationship (ER) model. For each connection between two values, the values are converted into units of the model and the types of relationships relating nodes in the model.

As for the generation of search criteria, this approach defines generic rules in order to deduce the user's requests from the ER model. Then, it extracts the search criteria (key words) describing the user's requests, to find services to recommend. Xiao [7] developed a method to dynamically determine a context model in order to recommend services. This model supported various contextual types and values, and then recommends services using this information. The authors made use of relations of similarity between the context values to find potential services the user might need.

In her work on services prediction, Salma Najar, offered a mechanism of discovery and prediction guided both by the context and user intent [48]. She used semantic similarity techniques. The system is based on the implementation of a matching algorithm, which computes the matching degree between the intention and the current context of the user and the set of semantic services described accordingly. OWL- SIC (OWL-S Intentional & Contextual) is an extension of OWL-S (Web Ontology Language-Semantic, is ontology, within the OWL-based framework of the Semantic).

The similarity approach requires historical data, to select and recommend services that are not always available. In fact, it requires a first phase of collection to get enough data which will be processed thereafter. Intentional approach provided by Najar [48] is a user-centered approach, but can generate conflict: technically, a problem of interoperability between services. Indeed, two compatible intentions do not necessarily make two technically compatible services.

A number of prediction algorithms have been developed for many projects and in different domains in recent years. In this section we have reviewed and detailed approaches and research projects related to context prediction. In the next section, we summarize in a comparative matrix, the different approaches we studied.

4 Comparative Analysis and Evaluation

Based on the formal presentation of the context prediction task [43], we can formalize a prediction task as a follow:

$$Pr = G (S1, S2, \ldots,St)$$

- St: the context data at a certain moment in time.
- Pr: a prediction result
- G: operator of context prediction operation

A context predictor in the most general sense can be viewed as follows: (Figure 10).

To evaluate a prediction approach, we established criteria related to data (e.g: type) and criteria related to reasoning (e.g: runtime, accuracy …), based on our analysis of the approaches developed in the previous section. These criteria seem to be the most relevant to evaluate and compare these methods.

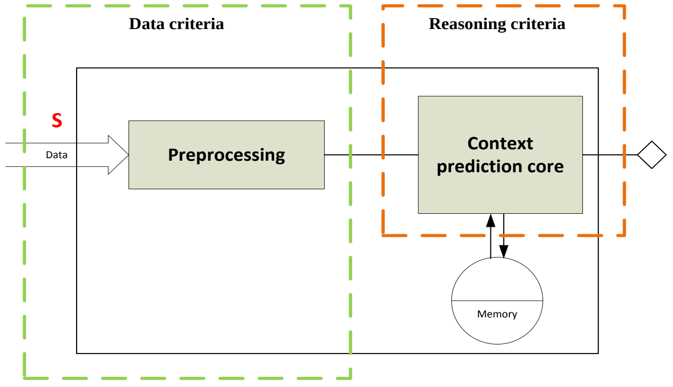


Fig. 10. General structure of context prediction (inspired from,[43])

4.1 Reasoning Evaluation Criteria

We start with the reasoning criteria: we established some criteria addressing efficiency, accuracy, time reasoning and scalability:

Accuracy: to assess the overall reliability and precision of the approach. Some projects were tested against real data to evaluate the accuracy of their approach, and are summarized in (Figure.11). Using the Lezi algorithm [22], testing with real data. They obtained prediction rates nearing 47%. Using MM, [26] the prediction accuracy was around 70-80%. Using HMM [29] prediction accuracy reached 98%. Using DBN, Petzold [33] had a prediction rating floating between 70% and 90%. In S.Najar's work, the system was based on the implementation of a matching algorithm. The prediction algorithm had a good quality result that neared 60% [48].

In our general evaluation of different approaches, we estimate the general accuracy as ranging from low/worst performance to high/best performance, depending on the capacity of the approach to be effective in a ubiquitous environment, taking into account some other criteria: e.g the loss of data, running time, etc.

Tolerance of missing values: dealing with missing data is a typical problem of preprocessing. Several machine learning methods implemented their own strategies for dealing with missing data. Even if some approaches are more sensitive to data loss. For example in similarity approach the similarity measure can be easily distorted by error in data value.

Evolvability: defined as the capacity of an approach for adaptive evolution and providing a scalable system.

Speed of runtime: indicates the processing time. These criteria evolved the overall time of the procedure; training time and real-time reasoning. Some methods were faster in reasoning time but needed a long training time e.g NN method.

Observability: if the method is a white box, its parameters show in a rather perspicacious way, how the system works. Looking at the system parameters, we can know exactly what the system has learned. And we can have some preliminary estimation; thus it becomes possible to configure the system accordingly. Otherwise if the method is a black box, it remains capable of prediction, but an expert cannot see the underlying reasons for the current decision, even if the expert knows all system parameters.

Prior knowledge inference: evaluates whether the approach can incorporate prior knowledge inference. If the method cannot do this, it could lead to low efficiency at the start. To overcome the problem we can pre-train the predictor and use generated training data before the system starts.

4.2 Data Evaluation Criteria

Data loss in preprocessing: the loss of information is dependent on both the prediction process and the data capturing the system. Preprocessing contextual information can causes a data loss that could influence the prediction results.

Type: numeric/ no numeric: if some type of data will not be accepted by the prediction approach a preprocessing will be required.

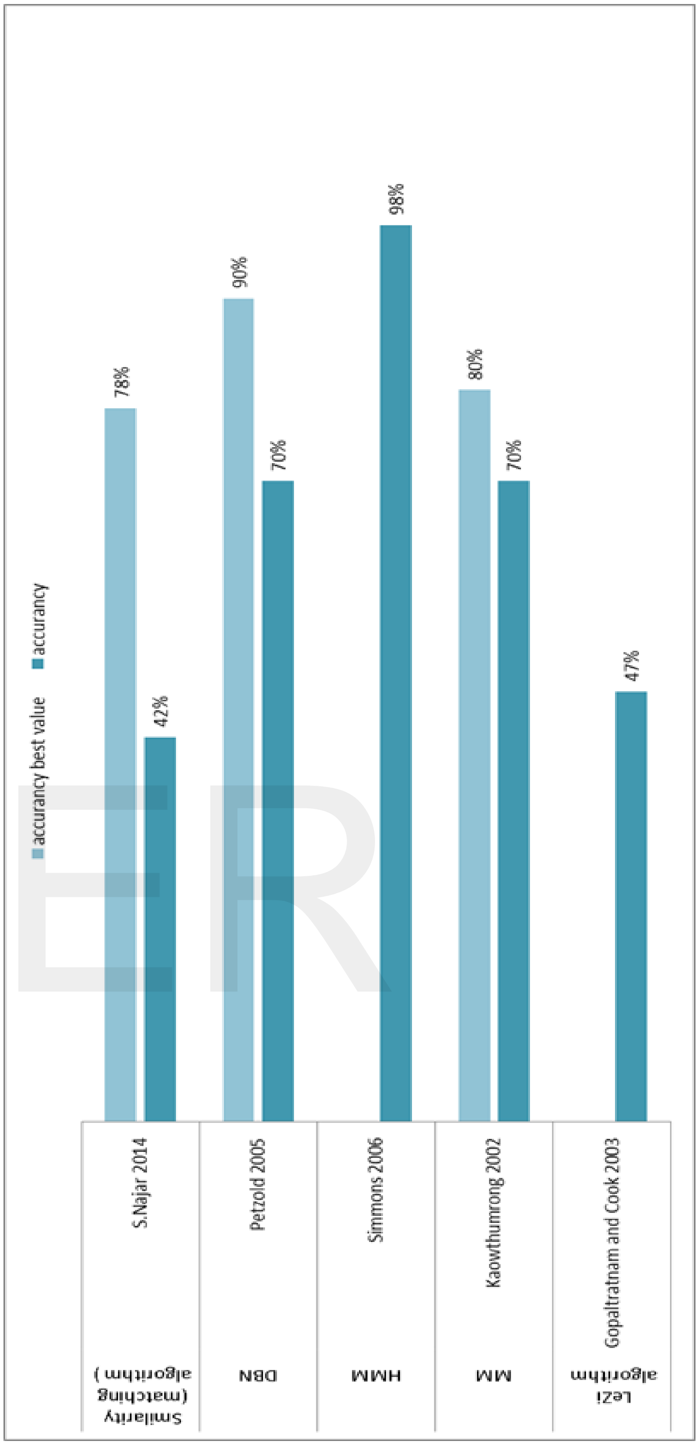
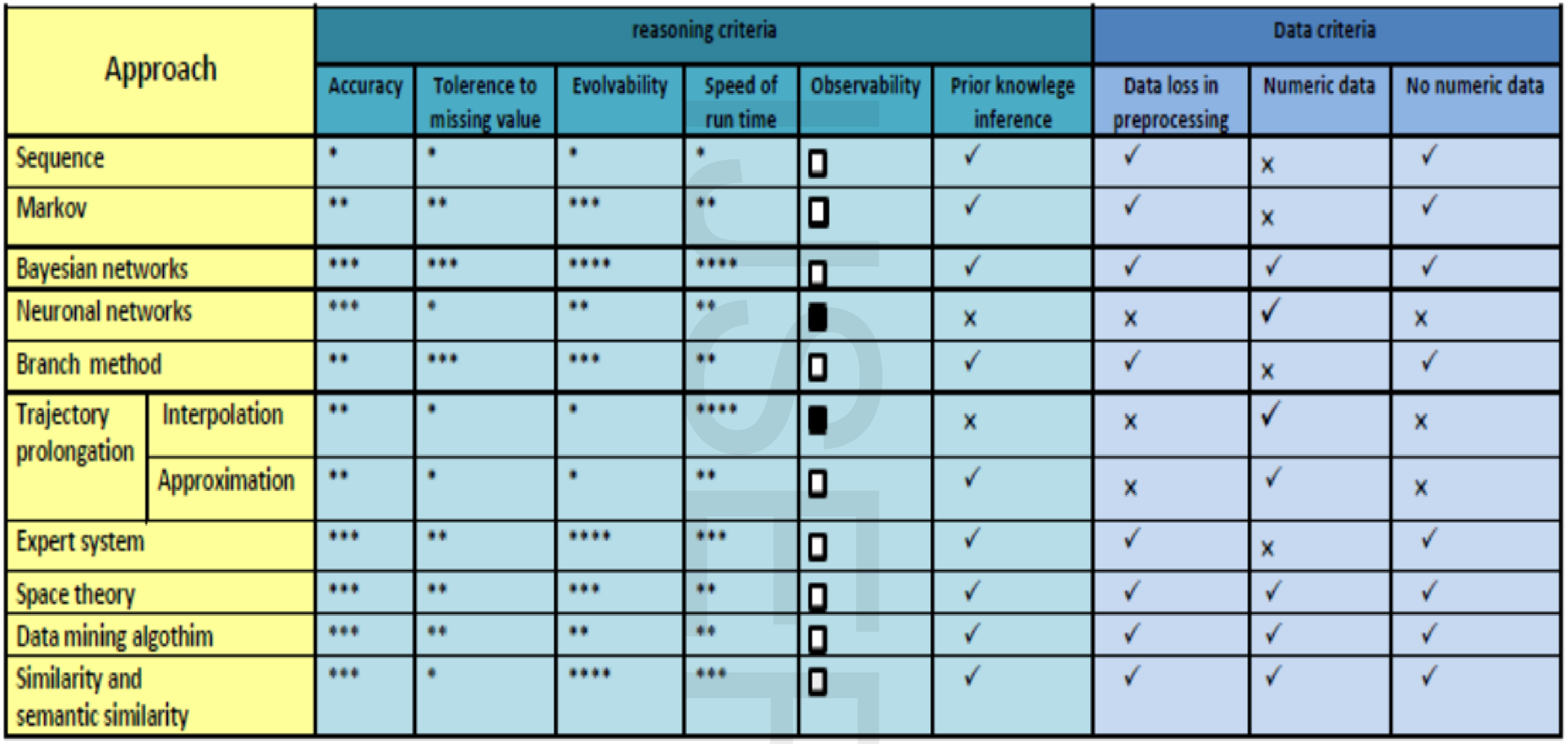


Fig11. Comparative accuracy diagram

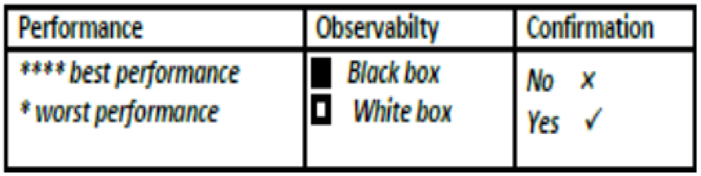
We shall summarize our analysis and valuation in the following table (Table 1)

TABLE 1
Comparative overview of prediction approaches



Approach		reasoning criteria						Data criteria		
		Accuracy	Tolerence to missing value	Evolvability	Speed of run time	Observability	Prior knowlege inference	Data loss in preprocessing	Numeric data	No numeric data
Sequence		*	*	*	*	□	✓	✓	×	✓
Markov		**	**	***	**	□	✓	✓	×	✓
Bayesian networks		***	***	****	****	□	✓	✓	✓	✓
Neuronal networks		***	*	**	**	■	×	×	✓	×
Branch method		**	***	***	**	□	✓	✓	×	✓
Trajectory prolongation	Interpolation	**	*	*	****	■	×	×	✓	×
	Approximation	**	*	*	**	□	✓	×	✓	×
Expert system		***	**	****	***	□	✓	✓	×	✓
Space theory		***	**	***	**	□	✓	✓	✓	✓
Data mining algothim		***	**	**	**	□	✓	✓	✓	✓
Similarity and semantic similarity		***	*	****	***	□	✓	✓	✓	✓

Legend



Performance	Observabilty	Confirmation
**** *best performance* * *worst performance*	■ *Black box* □ *White box*	*No* × *Yes* ✓

IJSER

5 Challenges of Context Prediction

Context prediction is a relatively new problem in computer science. Ground work on finding solutions and approaches for context prediction is still in progress. This work still faces many challenges including:

• Lack of general approaches, in much of what has been attempted so far, to tackle the context prediction.There have been only a few attempts to define and solve context prediction tasks with general approaches.

• Lack of generic automatic adaptation: most of the work is focused on the prediction, while neglecting the effective integration of results in services adaptation. With the notable exception of a few solutions implementing Markov Decision Processes, most of the remaining propositions, rely on expert systems to process the outcome of a context-prediction, thus making these system, less adaptive or autonomous.

• Interdependence between prediction and service adaptation: prediction systems process the context sequentially in a simple fashion. They predict, then pipe the result into the adaptation routine to act on it. This might be enough for simple problems, but will fall short for more complex problems. And as far as we could find, there is almost no attempt to explore this area.

• Optimization: using pervasive system in health care, as in as other high-risk endeavours. Prediction algorithm must be able to deliver prediction with High accuracy and minimal delays for computing.

We believe that once these challenges are resolved, we will be able to develop proactive and dynamic awareness-context systems that can handle more sophisticated use cases and situations, and improve their efficiency. Thus we might facilitate the overall use of such computer systems, and promote their integration within our daily activities.

6 Conclusion

In this paper we presented an overview of the literature on prediction approaches. We reviewed projects and research work related to context prediction. We established a set of comparative criteria we used for comparing and analyzing these approaches. We presented also several concepts that are required for context prediction. And we identified open issues for context prediction, such as many existing challenges.

References

[1] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, N. Souchon, L. Bouillon, et al., "Plasticity of user interfaces: A revisited reference framework," in In Task Models and Diagrams for User Interface Design, 2002.

[2] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in Handheld and ubiquitous computing, ed: Springer Berlin Heidelberg, 1999, pp. 304-307.

[3] R. Mayrhofer, "Context prediction based on context histories: Expected benefits, issues and current state-of-the-art," COGNITIVE SCIENCE RESEARCH PAPER-UNIVERSITY OF SUSSEX CSRP, vol. 577, p. 31, 2005.

[4] P. Nurmi, M. Martin, and J. A. Flanagan, "Enabling proactiveness through context prediction," in Proceedings of the Workshop on Context Awareness for Proactive Systems, Helsinki, 2005.

[5] A. Boytsov and A. Zaslavsky, "Context prediction in pervasive computing systems: Achievements and challenges," in Supporting Real Time Decision-Making, ed: Springer, 2011, pp. 35-63.

[6] S. Abbar, M. Bouzeghoub, and S. Lopez, "Context-aware recommender systems: A service-oriented approach," in VLDB PersDB workshop, 2009, pp. 1-6.

[7] H. Xiao, Y. Zou, J. Ng, and L. Nigul, "An approach for context-aware service discovery and recommendation," in Web Services (ICWS), 2010 IEEE International Conference on, 2010, pp. 163-170.

[8] K. Yu, B. Zhang, H. Zhu, H. Cao, and J. Tian, "Towards personalized context-aware recommendation by mining context logs through topic models," in Advances in Knowledge Discovery and Data Mining, ed: Springer, 2012, pp. 431-443.

[9] M. Meiners, S. Zaplata, and W. Lamersdorf, "Structured context prediction: A generic approach," in Distributed Applications and Interoperable Systems, 2010, pp. 84-97.

[10] M. Fayad and M. P. Cline, "Aspects of software adaptability," Communications of the ACM, vol. 39, pp. 58-59, 1996.

[11] M. Yarvis, P. Reiher, and G. J. Popek, "Conductor: A framework for distributed adaptation," in Hot Topics in Operating Systems, 1999. Proceedings of the Seventh Workshop on, 1999, pp. 44-49.

[12] D. Narayanan, J. Flinn, and M. Satyanarayanan, "Using history to improve mobile application adaptation," in Mobile Computing Systems and Applications, 2000 Third IEEE Workshop on., 2000, pp. 31-40.

[13] G. South, A. Lenaghan, and R. Malyan, "Using reflection for service adaptation in mobile clients," in XVII World Telecommunications Congress, Birmingham, UK, 2000.

[14] C. Efstratiou, K. Cheverst, N. Davies, and A. Friday, "An architecture for the effective support of adaptive context-aware applications," in Mobile Data Management, 2001, pp. 15-26.

[15] M. Aksit and Z. Choukair, "Dynamic, adaptive and reconfigurable systems overview and prospective vision," in Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on, 2003, pp. 84-89.

[16] J. Keeney and V. Cahill, "Chisel: A policy-driven, context-aware, dynamic adaptation framework," in Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on, 2003, pp. 3-14.

[17] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in Handheld and ubiquitous computing, 1999, pp. 304-307.

[18] B. D. Davison and H. Hirsh, "Toward an adaptive command line interface," in HCI (2), 1997, pp. 505-508.

[19] B. Davison and H. Hirsh, "Probabilistic online action prediction," in Proceedings of the AAAI Spring Symposium on Intelligent Environments, 1998, pp. 148-154.

[20] A. Bhattacharya and S. K. Das, "LeZi-update: an information-theoretic approach to track mobile users in PCS networks," in Pro-

ceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, 1999, pp. 1-12.

[21] S. K. Das, D. J. Cook, A. Battacharya, E. O. Heierman III, and T.-Y. Lin, "The role of prediction algorithms in the MavHome smart home architecture," Wireless Communications, IEEE, vol. 9, pp. 77-84, 2002.

[22] [22] K. Gopalratnam and D. J. Cook, "Active LeZi: An incremental parsing algorithm for sequential prediction," International Journal on Artificial Intelligence Tools, vol. 13, pp. 917-929, 2004.

[23] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," Information Theory, IEEE Transactions on, vol. 24, pp. 530-536, 1978.

[24] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," Journal of the American statistical association, vol. 93, pp. 1032-1044, 1998.

[25] I. Zukerman, D. W. Albrecht, and A. E. Nicholson, Predicting users' requests on the WWW: Springer, 1999.

[26] K. Kaowthumrong, J. Lebsack, and R. Han, "Automated selection of the active device in interactive multi-device smart spaces," in Workshop at UbiComp, 2002.

[27] J. Krumm, "A markov model for driver turn prediction," SAE Technical Paper2008.

[28] A. Gellert and L. Vintan, "Person movement prediction using hidden Markov models," Studies in Informatics and control, vol. 15, p. 17, 2006.

[29] R. Simmons, B. Browning, Y. Zhang, and V. Sadekar, "Learning to predict driver route and destination intent," in Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE, 2006, pp. 127-132.

[30] B. D. Ziebart, A. L. Maas, A. K. Dey, and J. A. Bagnell, "Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior," in Proceedings of the 10th international conference on Ubiquitous computing, 2008, pp. 322-331.

[31] D. W. Albrecht, I. Zukerman, and A. E. Nicholson, "Bayesian models for keyhole plan recognition in an adventure game," User modeling and user-adapted interaction, vol. 8, pp. 5-47, 1998.

[32] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse, "The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users," in Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, 1998, pp. 256-265.

[33] J. Petzold, A. Pietzowski, F. Bagci, W. Trumler, and T. Ungerer, "Prediction of indoor movements using bayesian networks," in Location-and Context-Awareness, ed: Springer, 2005, pp. 211-222.

[34] M. C. Mozer, "The neural network house: An environment hat adapts to its inhabitants," in Proc. AAAI Spring Symp. Intelligent Environments, 1998, pp. 110-114.

[35] L. Vintan, A. Gellert, J. Petzold, and T. Ungerer, "Person movement prediction using neural networks," in First Workshop on Modeling and Retrieval of Context, 2004.

[36] E. Al-Masri and Q. H. Mahmoud, "A context-aware mobile service discovery and selection mechanism using artificial neural networks," in Proceedings of the 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet, 2006, pp. 594-598.

[37] T. Lin, C. Wang, and P.-C. Lin, "A neural-network-based context-aware handoff algorithm for multimedia computing," ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), vol. 4, p. 17, 2008.

[38] J. Petzold, F. Bagci, W. Trumler, and T. Ungerer, "Context prediction based on branch prediction methods," ed: Citeseer, 2003.

[39] A. Karbassi and M. Barth, "Vehicle route prediction and time of arrival estimation techniques for improved transportation system management," in Intelligent Vehicles Symposium, 2003. Proceedings. IEEE, 2003, pp. 511-516.

[40] S. Sigg, S. Haseloff, and K. David, "A novel approach to context prediction in ubicomp environments," in Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on, 2006, pp. 1-5.

[41] S. Sigg, S. Haseloff, and K. David, "An alignment approach for context prediction tasks in ubicomp environments," IEEE Pervasive Computing, vol. 9, pp. 90-97, 2010.

[42] J. Hong, E.-H. Suh, J. Kim, and S. Kim, "Context-aware system for proactive personalized service based on context history," Expert Systems with Applications, vol. 36, pp. 7448-7457, 2009.

[43] A. Boytsov, Context reasoning, context prediction and proactive adaptation in pervasive computing systems: Luleå tekniska universitet, 2011.

[44] A. Padovitz, S. W. Loke, and A. Zaslavsky, "Towards a theory of context spaces," in Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on, 2004, pp. 38-42.

[45] A. Boytsov, A. Zaslavsky, and K. Synnes, "Extending context spaces theory by predicting run-time context," in Smart spaces and next generation wired/wireless networking, ed: Springer, 2009, pp. 8-21.

[46] A. Boytsov and A. Zaslavsky, "Extending context spaces theory by proactive adaptation," in Smart Spaces and Next Generation Wired/Wireless Networking, ed: Springer, 2010, pp. 1-12.

[47] Y. Benazzouz, "Découverte de contexte pour une adaptation automatique de services en intelligence ambiante," Ecole Nationale Supérieure des Mines de Saint-Etienne, 2011.

[48] S. Najar, M. K. Pinheiro, and C. Souveyet, "A new approach for service discovery and prediction on Pervasive Information System," procedia computer science, vol. 32, pp. 421-428, 2014.