PhD Dissertation

International Doctorate School in
Information and Communication Technologies

DISI - University of Trento

Recognizing and Discovering Activities of Daily Living in Smart Environments

Umut AVCI

Advisor:

Dr. Andrea Passerini

University of Trento

December 2013

Abstract

Identifying human activities is a key task for the development of advanced and effective ubiquitous applications in fields like Ambient Assisted Living. Depending on the availability of labeled data, recognition methods can be categorized as either supervised or unsupervised. Designing a comprehensive activity recognition system that works on a real-world setting is extremely challenging because of the difficulty for computers to process the complex nature of the human behaviors.

In the first part of this thesis we present a novel supervised approach to improve the activity recognition performance based on sequential pattern mining. The method searches for patterns characterizing time segments during which the same activity is performed. A probabilistic model is learned to represent the distribution of pattern matches along sequences, trying to maximize the coverage of an activity segment by a pattern match. The model is integrated in a segmental labeling algorithm and applied to novel sequences. Experimental evaluations show that the pattern-based segmental labeling algorithm allows improving results over sequential and segmental labeling algorithms in most of the cases. An analysis of the discovered patterns highlights non-trivial interactions spanning over a significant time horizon. In addition, we show that pattern usage allows incorporating long-range dependencies between distant time instants without incurring in substantial increase in computational complexity of inference.

In the second part of the thesis we propose an unsupervised activity discovery framework that aims at identifying activities within data streams in the absence of data annotation. The process starts with dividing the full sensor stream into segments by identifying differences in sensor activations characterizing potential activity changes. Then, extracted segments are clustered in order to find groups of similar segments each representing a candidate activity. Lastly, parameters of a sequential labeling algorithm are estimated using segment clusters found in the previous step and the learned model is used to smooth the initial segmentation. We present experimental evaluation for two real world datasets. The results obtained show that our segmentation approaches perform almost as good as the true segmentation and that activities are discovered with a high accuracy in most of the cases. We demonstrate the effectiveness of our model by comparing it with a technique using substantial domain knowledge. Our ongoing work is presented at the end of the section, in which we combine pattern-based method introduced in the first part of the thesis with the activity discovery framework. The results of the preliminary experiments indicate that the combined method is better in discovering similar activities than the base framework.

Keywords

Activity recognition, Activity Discovery, Pattern Mining, Segmental Labeling, Graphical Models

Acknowledgements

First and foremost I would like to sincerely thank my supervisor, Dr. Andrea Passerini, for giving me the chance to enter the world of research. I truly appreciate his guidance, and countless advice during my academic journey. Things I have achieved so far would not have been possible without his invaluable support.

I would also like to thank members of my lab (LION - machine Learning and Intelligent OptimizatioN), Tin, Stefano, and Paolo, for their helpfulness and for providing a pleasant working environment.

Thanks also to my dear friends Carmen, Giuliano, and Galena for their endless cheerfulness and for stimulating discussions and valuable feedback on my research and on other aspects of life.

I am very lucky to have Begum, Gozde, Ece, Basak, Elmas, Fatih, and Umut, who never made me feel alone. Thank you all for being my source of joy and and for making my life happy in Trento.

Two other friends I must mention are Ilker and Gorkem. They increased my motivation by constantly asking "are you done yet?". Thank you for your encouragement, support, and most of all your humor. You both kept things light and me smiling.

I have no words to express my gratitude to my family for their endless love, support, and encouragement. I would like to acknowledge the sacrifices made by my parents for my better education and upbringing.

Umut

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Computing and sensor technologies are moving towards a stage of evolution where computerized systems are going from visible to invisible, or rather fading into the background of our lives with recent advances in information science and engineering. By being more pervasive, computers embedded in our environments enable us to concentrate on our tasks rather than on the technology, allowing natural data regarding human activities to be collected (Weiser, 1991). Development of methods that provide a deeper understanding and a better interpretation of such data leads to crucial applications in fields like healthcare monitoring, safety, or surveillance.

1.1 Motivation

Activity recognition aims at recognizing actions of an agent(s) in order to make inferences on high-level activities and goals by analyzing human behaviors as a series of observations (Liao et al., 2003). Techniques to learn activities from observations can simply be divided into two main strands: supervised and unsupervised. The former requires labeled data upon which a recognition model is trained. The model learns a probabilistic association between activities and observations. The latter works in the absence of labeled data and tries to reveal implicit relationships and regularities of the data. For this purpose, observations are modeled either by density estimation or clustering methods (Chen and Khalil, 2011). After learning the associations in a supervised or unsupervised manner, activities can then be classified as normal, abnormal, or dangerous by comparing new unknown observation sequence with the pre-labeled ones.

Consider a scenario in which a caregiver is responsible for a patient's well-being in a home environment. One of the tasks of the person in charge is to monitor the patient's activities to identify an emerging medical condition before it becomes critical, for instance by encouraging him/her to act more safely. A person under medical treatment should follow strict rules on the dosage and timings of medicines as their misuse has severe consequences. It is effortless for the caregiver to identify an ongoing activity and to take decision whether the patient is safe or not, by (1) collecting evidence concerning the patient's actions, e.g. observing his/her interactions with the medicine cabinet, (2) using past experience to interpret the evidence, e.g. detecting an unusual behavior that conflicts with the regular ones learned after a long period of monitoring like spending too much time near the cabinet, and (3) applying knowledge to synthesize the gathered information, e.g. taking necessary precautions based on expertise in nursing.

Similar to the human based decision process, an automatic activity recognition system comprises several stages, i.e. collecting evidence, using past experience, and applying knowledge correspond sensing, learning, and inference respectively. However, each task in the computerized one is highly challenging and necessitates special consideration as system requirements differ from one application to another (body-worn accelerometers are used in fall detection instead of RFID readers, availability of labeled data determines the recognition algorithm to be employed etc.).

Activity data is collected in the *sensing* phase by observing the environment remotely via camera, audio or infrared systems (Robertson and Reid, 2006; Kolovou and Maglogiannis, 2010), by using wearable sensors like accelerometers or gyroscopes (Lustrek and Kaluza, 2009), or by employing environmental sensors like RFID readers that are tagged onto objects to be interacted with (Krahnstoever et al., 2005). The area of the application (surveillance, assisted cognition etc.), type of the activities to be recognized (individual or group), and the environmental conditions (indoor or outdoor) are basic criteria that should be taken into consideration in sensor selection.

Although some information explaining the activities may be deduced by analyzing the collected data, it is not singly adequate to recognize activities. For this purpose, activities should be modeled and interpreted via *learning* and *inference* steps. One can group methods to be applied under three headings as supervised (e.g., Bayesian networks (Yang et al., 2010), Support Vector Machines (SVMs) (Cao et al., 2009), Hidden Markov Models (HMMs) (Sanchez et al., 2007), Gaussian Mixture Models (GMMs) (Lin et al., 2008)), semi-supervised (e.g., self-learning (Guan et al., 2007)), and unsupervised (e.g., fingerprint mining (Gu et al., 2010)) approaches. There is not a single system that provides a perfect recognition rate because of the shortcomings of the methods, the complexity of the activities, and the technical issues. Supervised techniques require a large number of labeled samples which is hard and expensive to obtain, efficiency of semi-supervised methods is highly related to proportion of unlabeled dataset and model assumptions, and unsupervised techniques have low accuracy.

The complex nature of human behaviors gives rise to other problems. In many real world applications, agents try to achieve multiple goals within a single sequence of actions. In such a scenario, users perform independent activities simultaneously (concurrent activities) and switch between activities in case of an interruption (interleaved activities) (Hu and Yang, 2008). Besides, a majority of the approaches focus on recognizing short activities while long-term dependencies are poorly explored (Duong et al., 2009). There are also situations where similar sequence of actions produce different activities (ambiguity interpretation) (Kim et al., 2010).

Designing a comprehensive activity recognition system that works on a real-world setting is extremely challenging due to above-mentioned problems. In order to address these issues, in this thesis, we present novel supervised and unsupervised approaches for recognizing complex human activities in several real-world smart environments composed of distinct sensor networks.

1.2 The Problem

The process of developing an activity recognition system starts with equipping the environment with sensing capabilities, enabling to extract discrete or continuous signals depending on the type of the sensor used, i.e. simple sensors such as RFID readers (Yang et al., 2012) and more complex ones such as accelerometers (Fujimoto et al., 2013) or cameras (Vishwakarma and Agrawal, 2012) produce the former and the latter respectively. Although numerous choices are available on how to build a sensing platform, it is hard to create one that can gather information over long periods of time and that is not obtrusive to users. Camera-based systems suffer from both problems as storing videos is computationally expensive and individuals do not like to be recorded due to privacy issues. Besides, changes in the environmental conditions, e.g. lighting, clutter, often worsen the recognition performance (Tapia et al., 2004). Wearable sensors are also perceived as intrusive because of the necessity for residents to equip with many electronic peripherals and batteries. In addition, such systems are not able to distinguish activities that involve composite signals representing complex physical motions (Chen et al., 2012). Selection of a sensing platform is followed by a feature extraction process, i.e. transforming sensor readings into a series of observation vectors. Each sensor infrastructure requires a distinct conversion process. Luckily, extraction methods are determined by soft borders with numerous works over years (Chen et al., 2011a).

In this thesis, we focus on recognizing "Activities of Daily Living" (ADLs) (Katz et al., 1970) by using environmental sensors to bypass the drawbacks stated above. Environmental sensors are mainly composed of binary state-change sensors, e.g. RFID readers, reed switches, pressure mats, motion sensors etc., that are deployed in various objects or locations within residents' home. Interactions with the objects or locations of the residents provide an unobtrusive way of monitoring which can be prolonged over long periods of time due to the simplicity of sensors. Such sensing infrastructure is especially useful in detecting activities performed on a daily basis, e.g. cooking, toileting, showering. Therefore, we considered ADLs as activities to be recognized by the proposed system.

Learning and *inference* steps, following the *sensing*, constitute the building blocks of the activity recognition system. Developing an efficient machine learning technique allows the system to recognize activities with high accuracy. From a machine learning viewpoint, activity recognition can be formalized as a sequence labeling task: given a sequence of sensor readings covering a timespan of interest (e.g. a day), predict the sequence of activities being performed. The timespan is typically divided into small time intervals, to be labeled with the activity or the activities taking place. A number of machine learning algorithms have been applied to this task, ranging from simple Naive Bayes (Bao and Intille, 2004) to sequential approaches like Hidden Markov Models (Philipose et al., 2004), Conditional Random Fields (Vail et al., 2007) and their variants (van Kasteren et al., 2010a).

1.2.1 Modeling Long-Range Interactions

Local techniques like Naive Bayes or standard Support Vector Machines label each time instant independently, possibly extending its input representation over neighboring instants. On the other hand, sequential approaches collectively assign labels to all instants within the period of interest. This allows exploiting the relationship between activities performed at different time and usually results in performance improvements, other things being equal (van Kasteren et al., 2010b). In modeling temporal interactions, however, these models are limited to rather small spans. Sequential approaches rely on a Markovian assumption to limit the number of parameters to be learned and keep inference tractable.

There are a number of attempts in the literature in order to account for longer-range dependencies. Hierarchical approaches aim at representing activity relations in different levels of a hierarchy. Dependencies between short-range activities in the lower level of the hierarchy are fed into higher levels for creating longer-range ones (Fine et al., 1998; Duong et al., 2009; Natarajan and Nevatia, 2007). However, creating hierarchies requires deep knowledge regarding the underlying structure of the problem. Adding shortcut links between arbitrary time instants along the sequence, e.g. in skip-chain CRF, is another alternative (Hu and Yang, 2008) but the complexity of the model and the cost of inference increase drastically depending on the number of shortcuts introduced.

An activity usually spans a certain amount of time, its average duration depending on the specific activity being performed (e.g. taking a shower or watching TV). An activity segment is defined as a sequence of consecutive time instants in which the same activity is performed. Segmental labeling can be accomplished by semi-Markov models (Yu, 2010), which explicitly account for duration information. However, incorporating long-range dependencies between observations within each segment is again a bottleneck.

Hence, the first problem we deal with is to design an effective and accurate supervised recognition algorithm that incorporates long-range dependencies between distant time instants without incurring in substantial increase in computational complexity of inference. Aside from this specific problem, we aim at developing a method that can distinguish similar activities from each other and that is robust to intra- and inter-subject variability in performing activities.

1.2.2 Activity Discovery

Most of the work in the field has focused on supervised approaches in order to train activity models. However, these require the availability of labeled sequences, an expensive and time consuming process. Furthermore, training data is specific to the setting involving the activities to be recognized and the persons involved, as the daily living habits change from an individual to another. Activity discovery aims at identifying activities within data streams in the absence of data annotation. Therefore, it can be used in any possible daily-life scenario. In health monitoring applications, for instance, one of the tasks is continuously checking the behavior of a patient in order to determine whether his/her routines are maintained, regardless of the type of activities being performed. Inconsistencies in daily routines, i.e. changes in the structure of performed activities, can suggest problems in patient's health. For example, a person under physical therapy should perform a systematic exercise program of multiple movements, each serves a distinct purpose. Deterioration in the exercise schema, e.g. skipped steps, in the absence of professional assistance may result in delay in treatment.

As many unsupervised learning tasks, activity discovery is a challenging problem: many activities tend to share a similar set of signals (e.g. kitchen sensors for food-related activities), short periods lacking any signal at all can occur during an activity, to be distinguished from truly "idle" periods where no activity is being performed. Finally the discovery needs to be robust enough to account for variations in the way activities can be performed as in the supervised case. Two relative studies have recently proposed solutions to bypass above-mentioned limitations by using object-use fingerprints (Gu et al., 2010) and evidential ontology networks (Hong and Nugent, 2013). However, the methods use activity definitions acquired from the Web or the experts, which deviates such works from being fully unsupervised. Therefore, the second problem we tackle is to develop a fully unsupervised activity discovery technique to address these issues.

1.3 Proposed Approach

In this thesis, novel supervised and unsupervised approaches to activity recognition are proposed in order to overcome the limitations cited in the previous section. Data acquired from the sensors are first processed in different ways to create feature representations. We evaluate their efficiencies in prediction tasks and present a roadmap for choosing an appropriate representation for distinct environmental settings.

In the first part of the thesis, we develop a supervised approach to improve the recognition accuracy. For this purpose, we present a segmental pattern mining approach in which patterns characterize interactions within activity segments, enabling long-range dependencies to be modeled. Our solution consists of mining segmental patterns covering segments corresponding to a certain activity. Allowing gaps between matches of individual pattern elements enables distant observations to be related. We then show how to integrate sequential pattern mining into probabilistic segmental labeling algorithms, providing improved capacity to model longer-term dependencies. We introduce a probabilistic duration model representing the distribution of pattern matches along sequences, and integrate it into a Hidden Semi-Markov Model (HSMM).

We will show that our novel supervised approach

- can be used in various kinds of environments regardless the sensor platform, e.g. state-change sensors or motion sensors
- accounts for the long-range dependencies
- is robust to variations in performing activities
- can discriminate similar activities (ambiguity interpretation)
- is promising for detecting interleaved activities
- is applicable to other sequential labeling problems

In the second part of the thesis, we present an activity discovery framework that identifies activities in sensor streams without requiring data annotation. The process starts with dividing the full sensor stream into segments by identifying differences in sensor activations characterizing potential activity changes. Then, extracted segments are clustered in order to find groups of similar segments each representing a candidate activity. Lastly, parameters of a sequential labeling algorithm are estimated using segment clusters found in the previous step and the learned model is used to smooth the initial segmentation. We then introduce our ongoing work which is built on the top of the base activity discovery framework. For this purpose, we show how to take advantage of patterns in discovering activities.

We will show that our novel unsupervised approach

- requires neither any assumptions on dataset, e.g. type and number of activities nor domain knowledge.
- succeeds in discovering activities in many situations
- can be used in any daily-life scenario

1.4 Structure of the Thesis

This thesis is composed of six chapters. Chapter 2 presents a detailed review of the machine learning techniques used in the activity recognition problems. The studies that are relevant to our research questions, i.e. dealing with long-range dependencies and pattern mining in activity recognition, are discussed in Sections 2.1.2 and 2.1.3 respectively. We then summarize popular techniques to discover activities. Chapter 3 gives basic information regarding the experimental conditions. These include properties of the datasets to be used, data representation format, feature representations, and evaluation metrics. A novel method for recognizing activities in a supervised manner is given in Chapter 4. We initially provide a detailed formalization of the machine learning techniques to be used throughout the thesis. Following, feature representations are evaluated to determine baselines for pattern mining and for comparisons of alternative techniques. After defining how to mine segmental patterns in Section 4.3, we propose our novel solution, Pattern-based Hidden Semi-Markov Model, in Section 4.4, that is followed by a detailed experimental evaluation. Two fully unsupervised approaches to activity discovery are presented in Chapter 5. We provide a three-step activity discovery framework and its performance evaluation in Section 5.1. We describe our ongoing work as an improvement to the base technique, i.e. integration of patterns into the activity discovery framework, and analyze the preliminary results in Section 5.2. Finally, in Chapter 6, we draw our conclusions and propose future works.

1.5 Publications

The work presented in this thesis has been partially published in the following papers.

- Umut Avci and Andrea Passerini, "Improving Activity Recognition by Segmental Pattern Mining", IEEE Transactions on Knowledge and Data Engineering, PrePrints, issn = "1041-4347", 2013.

- Umut Avci and Andrea Passerini, "A Fully Unsupervised Approach to Activity Discovery", In ACM Multimedia workshop on Human Behavior Understanding (HBU 2013), Barcelona, Spain, 2013.

- Umut Avci and Andrea Passerini, "Improving Activity Recognition by Segmental Pattern Mining", 8th IEEE International Workshop on Pervasive Learning, Life, and Leisure (PerEL 2012), Lugano, Switzerland, March, 2012.

Chapter 2

State of the Art

The main challenge of the activity recognition problem lies within developing models that reflect the real nature of the behaviors. In this section, we will present widely used models and their extensions (Shen, 2004; Atallah and Yang, 2009).

2.1 Activity Recognition

2.1.1 Probabilistic models for Activity Recognition

Decision Trees

A decision tree is one of the easiest learning algorithms that takes inputs as properties and produces discrete outputs. It shows decisions with decision nodes and consequences with terminal leaves. Every decision node applies a test function to yield outcome labeling (Russell and Norvig, 2002). Decision trees can be represented as a logical formula by defining each path from the root to a leaf as a conjunction of conditions and by combining paths with the same class disjunctively. This property brings decision trees speed and high representation power.

A realtime activity recognition system for mixture of activities, e.g. lying, sitting, walking, running, and cycling, is introduced by (Parkka et al., 2010). Four features, i.e. spectral density, spectral entropy, signal average, and signal variance, are selected and used for constructing a decision tree of four nodes such that the first node discriminates movements from static activities (via spectral density), the second node discriminates direction of activities, e.g. vertical, horizontal, (via signal average), the third node differentiates cycling from walking and running (via spectral entropy), and the last node differentiates walking from running (via signal variance). Results show that the selected classifier needs a few comparisons, has low computational cost, and provides acceptable classification accuracy despite its simplicity.

Maurer et al. also present a realtime activity recognition system based on the body-worn sensors (Maurer et al., 2006).

Although decision trees are one of the most efficient learning methods, they are not robust enough to small variations in the data such that variations in the way activities are performed might result in a completely different tree (Logan et al., 2007).

Artificial Neural Networks

Artificial neural networks (ANNs) try to imitate information processing procedure of a biological neural system whose components are composed of neurons and links. In the artificial system, each neuron is responsible for an arithmetic operation the output of which will be served as input to the successor neurons through links (Russell and Norvig, 2002).

A basic system can be represented by a perceptron which consists of a number of input neurons linked to an output node. In this basic setup, output is computed as a function of a weighted sum of the inputs: $f(\sum_i w_i * x_i)$ where w_i, x_i are weights and inputs over examples respectively, f is an activation function like logistic or sigmoid. For complex settings, on the other hand, network structure should be modified by adding hidden layers with an arbitrary number of neurons between input and output layers.

Yang et al. propose an approach to build neural classifiers (a pre-classifier, a static classifier, and a dynamic classifier) based on signals received from a triaxial accelerometer (Yang et al., 2008). Pre-classifier aims at discriminating static activities from dynamic ones by using body acceleration feature. Once the distinction is made, classifiers for static/dynamic activities (standing, sitting, walking, etc.) are constructed using a feature set originated from the acceleration data. Zhu et al. and Chen et al. address similar issues but assignment of initial weights remains as a problem (Zhu and Sheng, 2009; Chen et al., 2010)

Scalability is an important issue in activity recognition because in non-scalable systems any change in system configuration, e.g. sensor change, requires the network to be modeled and trained again. Helal et al. address this issue by developing an adaptive multi-layer neural network (Helal et al., 2010). In the continuous sequence of activities, agents make a number of transitions between activities. ANNs learn these activities automatically from new inputs and adopt its interior computations. This property is known as online adaptation (Rivera-Illingworth et al., 2005). ANNs are also capable of capturing concurrent tasks (Helal et al., 2010).

ANNs are criticized for being a "black-box", i.e. relations between inputs and outputs are hidden within the network structure, which makes the interpretation of the calculated results difficult. Besides, different network topologies need to be tried out empirically to achieve the best result.

Support Vector Machines

SVMs can be used for linear or non-linear classification problems. In both cases, the aim is to locate a hyperplane separating classes from each other with a maximum margin that is the distance between two data points in each class where their distance from the hyperplane is minimum. The closest points to the hyperplane are called support vectors (SVs). If the classes are not linearly-separable some classification error is allowed by adding slack variables. In a non-linear classification problem, data is transformed from the original input space into a higher dimensional space where approximate linear separation of data is possible. This transformation is achieved by so-called kernel functions (Ben-Hur and Weston, 2010).

Qian et al. define activities in a surveillance system via SVM decision trees (Qian et al., 2010). In this approach, differences between activities are learned by identifying boundaries between activity classes in a hierarchical way constructed by the decision tree where each node is represented by an SVM binary classifier. By integrating all SVMs in the nodes, a multi-class SVM is generated (SVM-BTA). The authors state that SVMs are suitable for activity recognition problems because of their robustness against limited sample size and high generalization power. Another SVM-based activity recognition technique is offered by (Cao et al., 2009) where human activities are extracted from a video system. Acquired video is represented by a set of filtered images which will be fed into a classification module.

Tian et al. and Zhanchun et al. address an anomaly detection problem in terms of one-class SVM and PCA-SVM respectively (Tian et al., 2010; Zhanchun et al., 2006). One-class classification assumes that there is only one class label which is accepted as true. Discriminative boundary learned for usual behaviors defines whether a new instance is normal or abnormal.

Support vectors used for creating the final hyperplane play a crucial role in robustness of the classifier. Those vectors placed near the decision hyperplane make SVMs sensitive to noises or outliers as it is highly possible for new samples to be wrongly classified when they are located close enough to the hyperplane (He et al., 2012a). Support vectors also determine the computational complexity of the method as they increase linearly with the size of the training data. Apart from the effects of SVs, SVMs are not able to model temporal interactions as they are not sequential learners, i.e. they predict each time instant independent of the others.

Bayesian Networks

Bayesian networks (BNs) are graphical models structured as a directed acyclic graph and are especially designed for visually symbolizing relations between variables (Russell and Norvig, 2002). Each node in the graph represents a random variable along with the probability of the corresponding variable. The directed arcs between nodes indicate their dependencies, such that one variable affects the other one directly and this effect can be defined by a conditional probability. The whole graphical model with nodes and arcs forms the topology of the network the parameters of which are conditional probabilities. Here, the network is static, i.e. nodes and links remain the same over time.

Two related subjects are Naive Bayesian Classifiers and BNs with hidden nodes. In the former, activity recognition can be deduced to a classification problem by considering activities as classes. Bayes classifier then predicts activity labels after the generation of training examples (Laerhoven et al., 2003). The latter adds hidden nodes to the BNs as unobserved variables in order to represent dependencies between children (Liao and Ji, 2009).

Activity recognition based on an interaction between a user and an environment is investigated in terms of BNs in (Descheneaux et al., 2007). For different parts of the environment, different network structures are proposed because the authors emphasize the fact that there is a high correlation between the place and the activity performed, e.g. kitchen-eating. In this study, nodes correspond either sensors to be activated or activities themselves while arcs represent the interaction between the user's actions and the objects. A similar problem is addresses by (Wren and Tapia, 2006) using hierarchical approach.

Exact inference in BNs is a NP-hard problem which constitutes the main drawback of the technique. Therefore, graphical structure of a complex network needs to be simplified to overcome the computational problem. However, simplification on highly complex systems is daunting and prone to errors (Nazerfard and Cook, 2012). As far as the internal working mechanism is concerned, the technique does not track the changes within the network forming the output, i.e. it is stateless (Carter et al., 2006). In addition, continuous features are hard to handle in BNs (Hu and Hao, 2012).

Dynamic Bayesian Networks

Bayesian Networks are unable to model temporal processes as directed arcs of the network do not give any information about the time. In order to overcome this limitation, Dynamic Bayesian Networks (DBNs) was proposed as an upgraded version of BNs.

In the formalization of DBNs (Sanghai et al., 2005), the state at time t is represented by a set of random variables $Z_t = (Z_{1,t}, ..., Z_{d,t})$. In such state-space models, it is assumed that the observation (sensor activations) at time t was created by a process whose state Z_t (activities) is hidden from the observer. Prediction task aims at finding the most likely sequence of hidden states given the observations (Ghahramani, 2002). A state at a specific time t depends on the previous states but as a typical approach, a first-order Markov assumption is considered, i.e. each state is only dependent on the previous one. Representation of the transition distribution $P(Z_{t+1}|Z_t)$ is formed by considering a two-time-slice Bayesian Network fragment (2TBN), which includes two types of variable sets. The first variables are from Z_{t+1} whose parents are from Z_t and/or Z_{t+1} and the second variables are from Z_t without their parents. It is also assumed that the process is stationary, i.e. transition models for all time slices are the same: $B_1, B_2, ..., B_t, B_{\rightarrow}$. A DBN is then defined as a pair of Bayesian networks $(B_0, B_{\rightarrow})$ where B_0 represents the initial distribution $P(Z_0)$ and $B_{\rightarrow}$ represents a two-time-slice Bayesian network defining the transition distribution $P(Z_{t+1}|Z_t)$. Joint distribution is finally obtained as follows (under the assumption that the observed variables Y_t are dependent only on the current state variables X_t).

$P(X_0, ..., X_T, Y_0, ..., Y_T) = P(X_0)P(Y_0|X_0)\prod_{t=1}^{T} P(X_t|X_{t-1})P(Y_t|X_t)$

Recognition of activities from user-object interactions is tackled by (Inomata et al., 2009). Interactions between an agent and an object are detected via RFID tag system and this information is fed to DBNs for the recognition problem. Choice of DBNs is based on the misclassification doubt, e.g. a single object may be involved in performing several activities. Data of the interacted object is acquired by employing a sliding window approach, which will be used for training the network. By this way, conditional probability of the interaction-action pair is computed. Order of the actions is then explored via Hidden Markov Models. Authors state that misclassification generally occurs at transition between actions.

Muncaster et al. build a leveled DBN for recognizing complex events that include multiple sub-activities (Muncaster and Ma, 2007a). In this problem, sequence of sub-activities determines the relation between complex events as low-level actions/high-level activities. Levels in the hierarchy represent states except for the last level, which is the duration of simplest event. Lower levels of the DBN correspond to atomic actions. Then, lower levels are aggregated thorough the top to form complex actions. At the top, duration is modeled for each event. With the usage of DBN, dependencies between events are added to the structure in a systematic manner in order to keep parameters tractable. Another hierarchical DBN is proposed in (Du et al., 2006) for differentiating local features from global ones.

In (van Kasteren and Krose, 2007), three separate structures are used for inferring elderly activities. The proposed method includes a Naive Bayesian Classifier, a Dynamic Bayesian Network, and a history based dynamic Naive Bayesian Classifier. The results indicate that the dynamic model is superior to the static model as it takes the temporal aspects into account by providing information about the likelihood of a certain activity to follow another one. Inclusion of the sensor history helps capturing the correlations in the sensor patterns.

Static BNs are compared with DBNs for inferring the context of activities in (Frank et al., 2010). The authors emphasize the importance of transition modeling and encourage the use of dynamic inference models in case of continuous monitoring and high-frequency querying. It is pointed out that the authors trade-off better recognition accuracy for more processing time by using DBNs.

Similar to BNs, inference is again a problem especially when continuous data and loopy graphs are concerned. Effective optimization methods for learning and solving other graphical models are not applicable to DBNs (Oliver and Horvitz, 2005). Another limitation is related to the representation power of the network, stemming from the Markovian assumption. Activities with quantitative temporal constraints cannot be monitored completely (Colbry et al., 2002), i.e. only three temporal relations: precedes, follows, and equals can be captured (Zhang et al., 2013). Providing a solution for the second problem is of great importance. Complex activities are generally composed of several low-level tasks occurring in parallel or sequentially. Apart from identifying each action, decoding the dependencies of primitive tasks over different time instances enable one to fully understand and recognize the complex activities.

Hidden Markov Models

Hidden Markov Models (HMMs) are the simplest kind of DBNs with one discrete hidden node and one discrete or continuous observed node per slice. Most of the studies in activity recognition literature are based on HMMs because of their ability to represent spatio-temporal information.

HMMs are characterized by a number of components. These include the number of the states in the model (N), the number of observation symbols (M), the state transition probability distribution $A = a_{ij}$ where $a_{ij} = P[S_j = q_{t+1}|S_i = q_t], 1 \leq i, j \leq N$ (S and q represent state variable and state instantiation), the observation symbol probability distribution, for state j, $B = b_j(k)$ where $b_j(k) = P[O_t = v_k|S_j = q_t], 1 \leq j \leq N, 1 \leq k \leq M$ (O and v represent observation variable and observation instantiation), and the initial state distribution $\Pi = \Pi_i$ where $\Pi_i = P[S_i = q_1], 1 \leq i \leq N$. A complete model is defined by $\lambda = (A, B, \Pi)$ (Rabiner, 1989).

If we look at HMMs in terms of activity recognition, an activity is represented as a sequence of hidden states. A user is assumed to be in one of the states at each time and each state emits an observation (features). In the following time instance, the user makes a transition to another state in accordance with the transition probabilities between states. Once transition and emission probabilities have been learned from labeled data, activities are recognized by solving decoding problem, i.e. finding the most likely state sequence in the model that produced the observations (Aggarwal and Ryoo, 2011).

Although HMMs are one of the most popular methods, they suffer from some limitations. That's why, there are a number of extensions to the HMMs. Hidden Semi-Markov Model (HSMM) is one of them. Self-transitions in HMMs make state durations have geometric distribution implicitly which is not appropriate for many applications. Also in this classical model each state emits just one observation at a time. On the other hand, in HSMM, explicit state duration probability distribution is used instead of self-transition probabilities. By this way, states have variable durations and a number of observations are produced in each state according to the duration determined by the distribution (Hongeng and Nevatia, 2003).

Marhasev et al. take this issue one step forward (Marhasev et al., 2006). They state that the duration modeling is solely not enough to define the true nature of the activities. For this reason, non-stationary Hidden Semi-Markov Models (NHSMMs) are proposed in order to explicitly model dependencies of transition probabilities on state durations. With this update, transition probabilities change according to the time spent in a certain state by any user.

Another extension to the HSMMs is proposed by (Duong et al., 2005) for activity recognition and abnormality detection. The purpose of the study is to represent hierarchical structure of the activities and corresponding durations. Switching Hidden Semi-Markov Model (S-HSMM) achieves this by introducing a two-layered hierarchy where the bottom layer represents low-level actions and their durations with HSMMs, the top layer corresponds to a sequence of high-level activities each of which is composed of sequence of actions. In addition to the novel structure, state durations are modeled using Coxian distribution which is more realistic than Gaussian and provides better computational time (Skounakis et al., 2003).

Factorial Hidden Markov Models (FHMM) enable multiple dynamic processes to interact in order to produce a single output (Kulic et al., 2007). In this configuration, multiple independent dynamic chains contribute to the observed output and each chain has its own transition and output model. At each time instance, outputs of the dynamic chains are summed up and observed output is produced by feeding summed outputs to an expectation function. In a similar way (Liu and Chua, 2010) consider multi-agent activities where there is a single hidden process producing multiple observation sequences. Observation Decomposed HMMs (ODHMMs) allow modeling varying number of agents.

The problem of recognizing interleaved activities is addressed in (Landwehr, 2008) where hidden processes are interleaved such that only the process that produces an observation may pass to the following state. Coupled Hidden Markov Models (CHMMs) are able to model dynamic relations between several events by considering a set of HMMs where states at time t are conditioned by the states at time $t-1$ for all instances of HMMs (Ou et al., 2009). Here, each chain has a specific observation sequence. This approach is especially suitable for systems where activities have interactions.

Conditional Random Fields

Conditional Random Fields (CRFs) are undirected graphical models representing conditional probability of a sequence of hidden variables, e.g. activity labels, given a sequence of observations. That is the point CRFs differ from others by considering only labels in conditional probabilities, instead of joint probabilities of labels and observations. This makes CRFs a discriminative classifier rather than a generative one (Sutton and McCallum, 2007).

HMMs model the joint probability distribution computed by considering all possible observation sequences. Since it is computationally heavy, HMMs assume that the observations are conditionally independent given the state labels for keeping inference tractable. Contrarily, CRF does not require independence assumption between observations. This allows them to employ complex combinations of observations for defining features. That's why, CRFs enable more complex inputs to be handled easier than HMMs.

An extension to CRFs, semi-CRFs, is presented in (Sarawagi and Cohen, 2004) where labels are assigned to segments of the observation sequence instead of to any single observation. In another study, CRFs are extended to the Hidden CRFs in order to gain ability to represent underlying structure of classes by adding hidden states (Quattoni et al., 2007). Dynamic CRFs are introduced in (Liao et al., 2007) where distributed state representation is proposed as in Dynamic Bayesian Networks.

Nazerfard et al. propose an approach to compare performances of CRFs and HMMs (Nazerfard et al., 2010). Data is collected from motion and temperature sensors in a smart home environment. Selected features include a sensor identifier, time of the day, day of the week, previous activity and activity length. CRFs model is then trained with annotated data while feeding the system with features. Integration of features with states is of importance because it is not feasible to do so in HMMs due to factorization complexity. The results indicate that HMMs provide better performance for situations where the independence assumption holds.

van Kasteren et al. and Vail et al. make the same comparison with the previous work (van Kasteren et al., 2008b; Vail et al., 2007). The results in the former one show that HMMs outperform in real world dataset while CRFs outperform in artificial dataset. The latter one, on the contrary, indicates that discriminatively trained CRFs model performs better than HMMs even when the independence assumption holds.

Hu and Yang apply CRFs to an active research topic, inferring high-level goals from activity sequences that are interleaving and concurrent (Hu and Yang, 2008). In such a configuration activities are performed in a distributed way in order to achieve multiple goals. For the solution of the problem, a two-level approach is offered. Skip-chain CRFs are used for modeling interleaving goals. Then, concurrent goals are modeled by adjusting inferred probabilities via correlation graph.

By using CRFs, it is possible to model conditional probabilities without specifying the probability distribution of the observations, which is generally the most daunting phase. This property makes CRFs a convenient way for classifying complex and overlapped observations. However, training is computationally expensive when there are many features involved in the process.

Markov Logic Networks

A Markov Logic Network (MLN) is a statistical relational learning method that integrates first-order logic with probabilistic graphical models in order to represent complex probabilistic relations (Richardson and Domingos, 2006). It is a finite set of first-order logic formulas F_i each of which is attached to a real valued weight w_i. Each instantiation of F_i is given the same weight. Tran and Davis define how to build a Markov Network (MN) as follows (Tran and Davis, 2008):

- Each node of the MN corresponds to a ground atom x_k.
- If a subset of ground atoms $x_i \subset x$ are related to each other by a formula F_i, then a clique C_i over these variables is added to the network. C_i is associated with a weight w_i and a feature f_i defined as below.

$$f_i(x_i) = \begin{cases} 1 & \text{if } F_i(x_i) \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

After constructing the MLN, joint distribution of the set of binary ground atoms is calculated as follows, enabling to compute marginal distribution of any event given some observations using statistical inference.

$P(X = x) = \frac{1}{Z} exp(\sum_i w_i f_i(x_i))$ where Z is the normalization factor.

There are a few research papers on the applications of MLNs over activity recognition. Helaoui et al. (Helaoui et al., 2010) offer using MLNs to capture temporal relations and background knowledge for achieving better recognition performance. Background knowledge includes three main categories: information about a user (ID, location, and mental state), an environment (ID, type, state, weather, and temperature), and time (timestamps, part of the day, day of the week etc.). Simplified assumptions of the method are that (1) a few sensors are used to tag some of the objects, (2) activities occur one at a time, (3) only two temporal qualitative relationships (next, after) are considered. For two activities, actions, or events; a at timestamp t and b at timestamp d:

$$t + 1 = d \Longleftrightarrow next(b, a); t + 1 > d \Longleftrightarrow after(b, a)$$

Preliminary results show that temporal context has positive significant effect on the recognition accuracy even in a very basic configuration as in the experiment.

Wu and Aghajan focus on relationships between users and objects they interact (Wu and Aghajan, 2009). Authors refer to a MLN due to its ability to handle relational concepts as well as uncertainty in the knowledge base, observations, and decisions. The proposed technique is a three-step process. The first is analyzing user activities via video images. The second is modeling prior knowledge of object functions in the MLN. The last one is making inferences about location and identity of objects from observations acquired from user activities. With this scheme, objects are recognized regardless of their size, and place. Watson et al. introduce a comparison between MLNs and HMMs for recognizing discrete activity patterns (Watson et al., 2010). The results of the MLNs are reported as competitive.

Important advantages of Markov Logic Networks can be summarized as: (1) incorporating user activities as context information for object recognition. First-order logic part of the MLN intuitively represents knowledge, (2) dealing with uncertainties via probabilistic graphical models, and (3) modeling complex relations readily by using logic formulas while it is very hard to do directly in graphical models.

As common in many recognition approaches, MLNs suffer from the complexity of inference, which depends directly on graph structure. Logic formulas usually provide more sophisticated graphical structures than basic chain models. Besides, the size of the graph is correlated with the number of descriptive attributes and objects, which introduces a scalability problem for real-world datasets. Since exact inference is intractable, approximation approaches are employed on generative models (Khosravi and Bina, 2010). Accuracy of MLNs depends on the fine balance between logical and probabilistic parts. Having less balance between sides produces performance worsening (Watson et al., 2010).

2.1.2 Dealing with Long-range Dependencies

The problem of dealing with long-range dependencies is well-known in the machine learning community. A number of attempts at addressing it rely on hierarchical models like hierarchical HMMs (Fine et al., 1998) and their many recent variants (Duong et al., 2009; Natarajan and Nevatia, 2007). These try to model higher level dependencies between short-range activities which should account for the long-term dependencies. Du et al. assume that human activities can be decomposed into multiple interactive stochastic processes to represent distinct characteristics of activities, where each characteristic corresponds to a level in hierarchical DBNs (Du et al., 2008). Activity modeling is then achieved by modeling the interactive processes. However, hierarchical approaches require a good amount of knowledge about the underlying structure of the problem in building the hierarchies. The SAMMPLE architecture (Yan et al., 2012) learns high-level activities as combinations of low-level locomotive micro-activities (e.g. sitting). These latter are learned with appropriate classifiers trained on supervised instances of micro-activities. Micro-activities are also used in (Huynh et al., 2008) as building blocks, combined through Topic Models to predict daily routines like commuting or office work. Depending on the granularity and duration of the activities to be predicted, it is not always easy to develop effective hierarchical models. Indeed, our preliminary experiments showed that a two-level hierarchical HMM, modeling activity segments at the lower level and sequences of segments at the upper one, did not improve over plain HMM in any of our experimental scenarios. Another approach for modeling long-range dependencies consists of explicitly adding links between distant time instants which are deemed to be in direct relationship, like in Skip-chain CRFs (Sutton and McCallum, 2007). Hu et al. propose using this approach in order to recognize concurrent and interleaved activities (Hu and Yang, 2008). Interleaving goals are modeled by leveraging the skip chains while concurrent ones are identified by adjusting inferred probabilities via correlation graphs. However, the method requires a large amount of training data because of the many possible ways in which an ongoing activity can be interrupted and resumed. Furthermore, shortcut links considerably increase the complexity of the inference task and should thus be carefully selected. Skip-chain CRFs have been successfully applied to named-entity recognition, where shortcut links are added between pairs of identical capitalized words.

2.1.3 Pattern Mining in Activity Recognition

The idea of mining patterns from the sensor data has been extensively studied in the activity recognition community for various applications. A method to detect anomalies in human behavior is proposed in (Cardinaux et al., 2008). Patterns extracted for each type of activity are used to create probabilistic behavior models. Abnormal behaviors are then identified by investigating deviations from the models. Hasan et al. apply frequent set mining to create a low-dimensional feature representation from a large number of binary sensors (Hasan et al., 2010). Tao et al. develop a technique based on Emerging patterns to recognize sequential, interleaved and concurrent activities for single (Gu et al., 2009a) and multiple (Gu et al., 2009b) users. They attacked the activity recognition as a classification problem since Emerging Patterns define the significant changes between two classes of data. Contrary to previous work, Rashidi et al. exploit patterns in order to discover activities in an unsupervised manner (Rashidi et al., 2011). Introduced mining method is able to extract frequent patterns that may be discontinuous and might have variability in the ordering. Found patterns were then clustered by k-means algorithm using an amended edit distance as similarity function to measure pattern differences.

Patterns can be exploited differently to account for assumptions on the ways that activities are performed and on the sensor structure. Palmes et al. suggest that the lists of objects associated with each activity are robust to changes in performing activities and are unique across the activities (Palmes et al., 2010). Hence, the most relevant objects for each activity are mined from the web and highly discriminative ones are used to recognize activities. Similar to the previous method, object usage information is mined to discover frequently occurring object interactions in (Heierman and Cook, 2003). However, instead of using all frequent patterns, only those worthy of improving home automation are included in recognition process. By this means, noisy patterns (e.g. random ones) are filtered, enabling the prediction system to work efficiently. Former methods assume that sensors, accordingly patterns mined from them, remain the same during run-time. Activity recognition system proposed in (Roggen et al., 2013), on the other hand, adapts itself to changes in sensor structure by self-monitoring. The system extracts activity patterns in the beginning of recognition process and monitors its behavior by comparing existing patterns with those acquired from streaming signals. Patterns are adjusted based on the new system configuration if a change is detected. Understanding behavioral diversities in time is another point that one needs to take into consideration apart from the changes in sensor structure. Rashidi and Cook introduce a tilded-time approach in which behavioral patterns are extracted at a finer level for recent times and at a coarser level for older times (Rashidi and Cook, 2010). Such approach is useful particularly in situations where recent changes in behaviors need to be analyzed more carefully than older ones, e.g. a patient's condition after a surgery.

In most of the pattern-based activity recognition approaches, patterns are selected from those with higher occurrences of all observation sequences. However, such choice does not ensure activities to be represented accurately as extracted patterns may be frequent across several activities. A number of studies have been proposed to address this issue. Sim et al. introduce correlated patterns as those having higher occurrences only in activities that they are associated with (Sim et al., 2011). By this way, patterns guarantee that activities are uniquely characterized. The same rationale is followed by (Chikhaoui et al., 2011). Activities are first represented in different granularities by a hierarchical structure. Frequent pattern mining is then applied across the levels of the hierarchy to generate activity specific patterns. Finally, recognition is achieved by a mapping function between the frequent patterns and the activity models.

Understanding human activities requires not only recognizing individual actions but also identifying the relations between them occurring in different times. Therefore, analyzing temporal relations in activity recognition problems is of great significance. T-patterns are used in (Salah et al., 2010), where the authors propose two improvements over the base approach, namely testing independence between two temporal points and Gaussian Mixture Modeling of correlation times, to detect temporal patterns at a low computational cost and to make the model more robust to spurious patterns. Jakkula et al. investigate temporal relationships between frequently occurring events to detect anomalies in smart environments (Jakkula et al., 2009). Associations between frequent patterns are identified based on Allen's criteria (Allen and Ferguson, 1994), e.g. before, contains, starts. Probabilities calculated for temporal relationships define whether a given event is anomalous or not. However, none of these studies addresses the long-term dependency problem.

2.2 Activity Discovery

Methods proposed for discovering activities can be summarized in terms of sensor structure used in data gathering process. Vilchis et al. present a two-step process for activity identification and knowledge discovery from video (Vilchis et al., 2010). The system first extracts behavioral displacement patterns representing the origin and destination of moving objects by analyzing the object's entry and exit points in the scene. In the second step, more complex patterns, e.g. temporal information on interactions of objects, are extracted by aggregating soft-computing relations. Both simple and complex patterns are then modeled via fuzzy relations, allowing one to label activities in a human-like language. Patterns are exploited also in (Pusiol et al., 2010) in order to create generic activity models, i.e. discovered activities, from low-level visual clues. Discovery process comprises (1) identifying significant trajectories characterizing fundamental motions of an individual to perform basic tasks, (2) capturing meaningful regional transitions by using important trajectories, and (3) generating activity models as patterns of captured transition topology. The relation between action primitives and complicated scenarios regarding semantic interpretation of the monitored scene inspires (Muncaster and Ma, 2007b) to use hierarchical DBNs for recognizing activities automatically. Lower levels representing the atomic activities are determined by the deterministic annealing clustering method. Discovered actions are propagated incrementally towards the higher levels to form complex ones. The last level allocated for the duration modeling allows the system to clarify the varying durations of automatically recognized activities. In (Wiliem et al., 2009), the authors propose an adaptive system to classify human actions when prior information concerning activities is not available. For this purpose, each incoming video feed in continuous streaming is represented by Bags-Of-Words method using Term Frequency Inverse Document Frequency (TF-IDF) features. A datastream clustering algorithm is then employed to update the system's knowledge with the new incoming representation where similarity between feeds is computed by a modified normalized cosines distance.

Wearable sensors, as another type of sensing platform, produce continuous signal as time-series data. Common to many activity discovery approaches is finding frequently occurring patterns, which are called motifs in the context of time-series due to the close analogy to their discrete counterparts. Lin et al. are the first to introduce the concept of motifs for one-dimensional time-series (Lin et al., 2002). In most of the cases, on the other hand, recognizing activities by using wearable sensors requires employing multiple sensors, which creates higher dimensional time-series. Mining motifs, together with their estimated lengths, from such data is introduced in (Minnen et al., 2007) for automatic classification of activities. However, mining is performed over synchronous time intervals spread on each dimension of the data. Vahdatpour et al. introduce a realistic approach by considering the fact that motifs characterizing an activity have different length and timing properties in each dimension of the signal (Vahdatpour et al., 2009). The proposed technique extracts asynchronous multi-dimensional motifs, the elements of which have temporal, length, and frequency variations. The mining process includes extracting single dimensional motifs in all levels of the time-series data, and building multi-dimensional motifs by combining discovered single-dimensional ones via graph clustering. After achieving successful results in activity discovery, the authors customized their method for discovering abnormal activity occurrences in (Vahdatpour and Sarrafzadeh, 2010). Understanding structures in human behavior allows one to make inferences about how an activity is performed. A novel scheme for unsupervised detection of structure in activity data is introduced in (Huỳnh and Schiele, 2006). The idea behind the approach is to concentrate on significant dimensions in data characterizing distinct activities, which is achieved by using PCA. Since representing activities by a single linear eigenspace is too general to capture low-dimensional structure of the data, multiple eigenspaces are extracted from the general one, each corresponds to an individual activity.

Environmental sensors are used in the majority of studies in activity discovery. Employing sequential patterns in order to represent activities was proposed in (Rashidi et al., 2011) and (Chikhaoui et al., 2012). Both approaches are based on the idea that similar patterns can be used for representing the activities. Following this idea, patterns are first mined from data and then clustered by k-means and LDA respectively. Rasanen suggests using patterns in a hierarchical framework (Räsänen, 2012). Statistically significant recurring structures extracted for short-term activities in the lower levels of the hierarchy are fed to the higher levels, in which their presence are analyzed. Segments that are matched with the patterns are clustered into context categories. However, these techniques are bound to the quality and coverage of the extracted patterns. Hamid et al. suggest clustering segments instead of patterns extracted from them (Hamid et al., 2009). For this purpose, segments are represented by histograms each of which then corresponds to a node in an edge-weighted graph. Maximal cliques in the graph are identified as activity candidates. However, the method cannot be used for real activity discovery as segments are assumed to be known in advance. Hong et al. developed an activity discovery approach based on conceptual definitions of activities in terms of Evidential Ontology Networks (EON) (Hong and Nugent, 2011). A candidate segment that fits an EON best is recognized as the corresponding activity. Since the method works on previously segmented data, the authors later introduced three segmentation approaches (Hong and Nugent, 2013). Extracted segments are fed into the EON for determining activities. Another ontology-based activity discovery platform is presented by (Chen et al., 2011b). Similar to the previous approach, activity models are created through ontology engineering, in which prior knowledge is acquired from domain experts and relevant documents. Created ontologies, i.e. activity models, are used together with a sequence of sensor activations as an input to a discovery module, where ontological subsumption reasoning is carried out to infer the activity being performed. However, such methods are based on a deep knowledge of the activities being searched, needed to compile the set of rules which are used for the activity discovery phase. Automated knowledge acquisition as proposed in (Wyatt et al., 2005), i.e. web mining, on the other hand, may facilitate the usage of such techniques.

Chapter 3

Smart Environments

3.1 Datasets

Activity recognition is a popular subject in the field of machine learning. Researchers working on the topic evaluate their methods on data acquired from simulation or from sensors embedded in smart environments. However, the vast majority of the real-world datasets are not made publicly available because of privacy and copyright issues. Their availability, on the other hand, is of great significance as they provide the community standardized testbeds to be used for comparison purposes.

Among benchmark datasets that are freely available, only a few were created specifically with the aim of detecting activities of daily living. Characteristic features identifying human activities are collected from a camera-based system in (Pirsiavash and Ramanan, 2012), from wearable devices in (Huynh et al., 2008), and from environmental sensors in (Rashidi et al., 2011) and (Chikhaoui et al., 2010). Although we narrow our focus to environmental sensors (see Section 1.2), the last two options do not provide suitable testbeds as data is gathered from people performing scripted activities, i.e. pre-determined activities are repeatedly performed by several users. Such approach accounts for inter-subject variability, yet it is not sufficient for explaining real-world situations.

We therefore conducted our experiments on a collection of freely available[1 2] benchmark datasets, in which activities can be performed at any order and at any time in an observation sequence, e.g. a month. van Kasteren's dataset includes information regarding three different houses comprising several wireless sensor networks (van Kasteren et al., 2010a; van Kasteren et al., 2010b). Each node of the network is attached to ad-hoc sensors, e.g., reed switches, passive infrared (PIR). Annotation of the activities was achieved by recording the start and end time of the corresponding activity either via handwritten diary or bluetooth headset. CASAS

[1] https://sites.google.com/site/tim0306/codeFramework.zip

[2] http://ailab.wsu.edu/casas/datasets/twor.2009.zip

dataset differs from the previous one as two residents are simultaneously monitored in the apartment, with a sensor network mainly composed of motion and utility usage sensors (Cook and Schmitter-Edgecombe, 2009). Annotators labeled the data using a 3D visualization tool and residents' diaries.

Table 3.1 presents a summary of the characteristics of the datasets. Note that in CASAS dataset activities performed by residents, hence activated sensors, differ from each other.

Table 3.1: Details of the datasets

		Duration	**Sensors**	**Activities**	**Data points**	**Annotation**
van Kasteren	House A	25 days	14	10	35486	Bluetooth
	House B	14 days	23	13	19968	Diary
	House C	19 days	21	16	26236	Bluetooth
CASAS	Resident 1	46 days	52	7	64795	Diary
	Resident 2	46 days	63	9	64785	Diary

Activities to be recognized were derived from the Katz ADL index which is a measure qualifying the ability of individuals to sustain their lives independently (Katz et al., 1970) for the first dataset, and from the clinical questionnaires (Reisberg et al., 2001) for the second dataset. The activities and sensors defined in these datasets are listed in Tables 3.2 and 3.3 respectively. 'Idle' indicates that none of the annotated activities is being performed. Some of the activities, namely 'Going to bed'/'Sleeping', 'Leaving house' and 'Idle' itself, take significantly more time than the others on average. Floor plans of the smart homes, together with the placement of the sensors, are depicted in Figures 3.1 and 3.2 for the van Kasteren and the CASAS datasets respectively. Sensor structure shown in the latter figure differs from the one presented in the original paper (Cook and Schmitter-Edgecombe, 2009) as we concentrate on the sensors that are fired while performing activities.

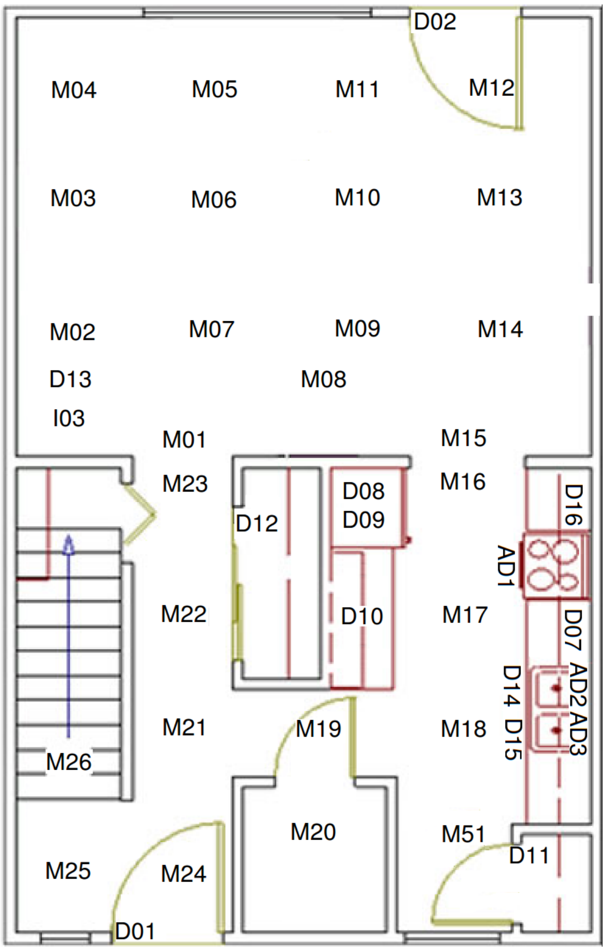
3.2 Data Representation

A dataset $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})^{(1)}, \ldots, (\mathbf{x}, \mathbf{y})^{(d)}\}$ is a collection of input-ouput sequences for a number of days d. An input example $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ consists of a consecutive sequence of observations, each covering a certain time instant t. An observation $\mathbf{x}_t$ is represented by the set of sensors which are active at that time instant (i.e. within its time interval). Different choices can be made in deciding when a sensor is considered active, as will be discussed in the next section. When feeding input sequences to labeling algorithms (see Section 4.1), observations will be represented as binary vectors rather than sets. Given N sensors, an observation $\mathbf{x}_t$

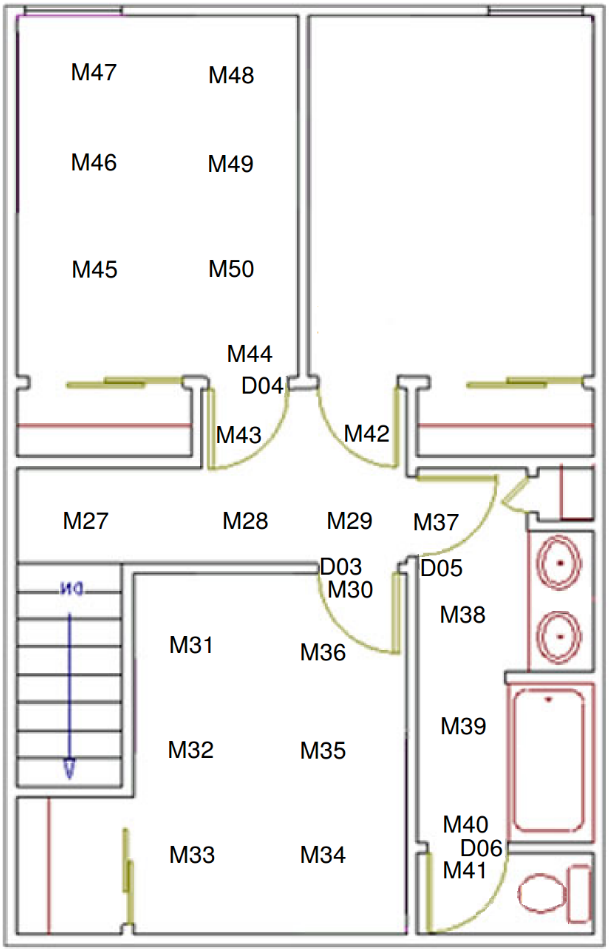
(a) House A (b) House B

(c) House C, First floor (d) House C, Second floor

Figure 3.1: Floor plans of houses A, B, and C for van Kasteren Dataset. Sensors are depicted as red boxes (Taken from (van Kasteren et al., 2010b))



(a) First floor



(b) Second floor

Figure 3.2: Floor plan for CASAS Dataset. (Taken from (Cook and Schmitter-Edgecombe, 2009))

Table 3.2: Activities performed in van Kasteren and CASAS datasets

van Kasteren			CASAS	
House A	**House B**	**House C**	**Resident 1**	**Resident 2**
Idle	Idle	Idle	Idle	Idle
Leaving house	Leaving house	Leaving house	Bed to toilet	Bed to toilet
Using toilet	Using toilet	Eating	Breakfast	Breakfast
Taking shower	Taking shower	Using toilet (down)	Grooming	Grooming
Brushing teeth	Brushing teeth	Taking shower	Sleeping	Preparing dinner
Going to bed	Going to bed	Brushing teeth	Working at computer	Preparing lunch
Preparing breakfast	Getting dressed	Using toilet (up)	Working at dining room	Sleeping
Preparing dinner	Preparing brunch	Shaving		Watching TV
Getting snack	Preparing dinner	Going to bed		Working at computer
Getting drink	Getting drink	Getting dressed		
	Washing dishes	Taking medication		
	Eating dinner	Preparing breakfast		
	Eating brunch	Preparing lunch		
		Preparing dinner		
		Getting snack		
		Getting drink		

will thus be encoded as a binary feature vector $\mathbf{x}_t = (x_t^1, \ldots, x_t^N)$, each feature being 1 if the corresponding sensor is active and 0 otherwise.

The labeling task consists of predicting a sequence of activity labels $\mathbf{y} = \{y_1, \ldots, y_T\}$, one for each time instant. Each label $y_t \in [1, L]$ is one of L possible activities, with one indicating no activity. We assume here that activities are not simultaneous, i.e. only a single activity is performed at each time instant. The segmental pattern mining algorithm can however be generalized to deal with multiple simultaneous activities, as will be discussed in Section 4.4.3. We define an activity segment as a sequence of consecutive time instants labeled with the same activity. A segment $s_u = (b_u, e_u, y_u)$ is represented by its starting and ending time instants $b_u, e_u \in [1, T]$, with $e_u \geq b_u$, and the segment label y_u. A label sequence $\mathbf{y}$ can be split into a sequence $\mathbf{s} = \{s_1, \ldots, s_U\}$ of activity segments such that $b_1 = 1$, $e_U = T$, $b_u = e_{u-1} + 1$ and $y_u \neq y_{u-1}$ for all u. We define as $\mathbf{x}_{b_u:e_u}$ the segment of $\mathbf{x}$ ranging from b_u to e_u included. A collection of $\mathbf{s}$ over all the days forms $\mathcal{S} = \{\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(d)}\}$ with d being the number of days. We define $\mathcal{S}_y$ as the set of segments for a particular activity y, i.e. $\mathcal{S}_y \subset \mathcal{S} : \forall s_u = (b_u, e_u, y_u) \in \mathcal{S}_y, y_u = y$. The corresponding set of input segments is $\mathcal{D}(\mathcal{S}_y) = \{\mathbf{x}_{b_u:e_u} : (b_u, e_u, y_u) \in \mathcal{S}_y\}$. A summary of notations is given in Table 3.4.

Table 3.3: Sensor infrastructure for van Kasteren and CASAS datasets

van Kasteren			CASAS
House A	**House B**	**House C**	
Cups cupboard	Balcony door	Bathroom door	Mxx - motion sensors
Dishwasher	Bathroom door	Bathroom sink	Ixx - television
Freezer	Bedroom door	Bathtub	Dxx - door\cabinet sensors
Fridge	Bedroom dresser	Bedroom door	AD1 - burner
Front door	Bedroom PIR	Bedroom dresser	AD2 - hot water
Groceries cupboard	Bed pressure mat (left)	Bed pressure mat (left)	AD3 - cold water
Hall bathroom door	Bed pressure mat (right)	Bed pressure mat (right)	
Hall bedroom door	Cutlery drawer	Cups cupboard	
Hall toilet door	Fridge	Cutlery drawer	
Microwave	Front door	Freezer	
Pans cupboard	Groceries cupboard	Fridge	
Plates cupboard	Kitchen PIR	Front door	
Toilet flush	Kitchen sink	Herbs cabinet	
Washing Machine	Microwave	Keys	
	Office chair pressure mat	Living room couch	
	Plates cupboard	Microwave	
	Seat pressure mat	Pan cupboard	
	Stove lid	Toilet door (down)	
	Toaster	Toilet flush (down)	
	Toilet door	Toilet flush (up)	
	Toilet flush	Towel drawer	
	Window		

3.3 Data Features

Data acquired from the sensors were processed in different ways to create feature representations. The *Raw* representation is the unprocessed one, where a sensor is active in all time instants in which it fires. The *Changepoint* (C) one considers a sensor active (value 1) only in the time instants in which it alters its state. The *Last-fired* (L) representation keeps considering the last sensor which changed state active in all following time instants, until another sensor changes its state. Note that in case multiple sensors change their state in the same time instant, all of them are considered active for that time instant. The last one that changed its state is carried over to the following time instants. Figure 3.3 depicts the working mechanism of these representations as compared to the original raw one. We also considered a *Dual Changepoint* (DC) representation, distinguishing between activation and de-activation events.

Table 3.4: Notation Summary of Data Representation

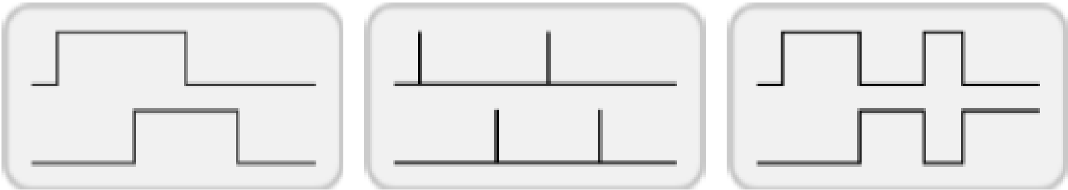
Notation	**Description**
L	number of activities (states)
N	number of sensors (observations)
T	length of the input sequence
t	current time instant, $1 \leq t \leq T$
$\mathbf{y} = \{y_1, \ldots, y_T\}$	activity labels, $y_t \in [1, L]$
$\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$	sequence of observations
$\mathbf{x}_t = (x_t^1, \ldots, x_t^N)$	binary feature vector at time t
$\mathcal{D} = \{(\mathbf{x}, \mathbf{y})^{(1)}, \ldots, (\mathbf{x}, \mathbf{y})^{(d)}\}$	dataset of input-output sequences
b_u, e_u, y_u	starting and ending time instants of an activity segment, and segment label. $b_u, e_u \in [1, T]$
$\mathbf{s} = \{s_1, \ldots, s_U\}$	sequence of activity segments, $b_1 = 1$, $e_U = T$, $s_u = (b_u, e_u, y_u)$, $1 \leq u \leq U$
$\mathbf{x}_{b_u:e_u}$	segment of observations ranging from b_u to e_u
$\mathcal{S} = \{\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(d)}\}$	set of $\mathbf{s}$ over all the days
$\mathcal{S}_y$	set of segments for activity y, $\mathcal{S}_y \subset \mathcal{S} : s_u = (b_u, e_u, y_u) \in \mathcal{S}_y, y_u = y$
$\mathcal{D}(\mathcal{S}_y) = \{\mathbf{x}_{b_u:e_u} : (b_u, e_u, y_u) \in \mathcal{S}_y\}$	set of input segments for $\mathcal{S}_y$

3.4 Evaluation Metrics

We assess the performance of our system by using *Precision*, *Recall*, *F1-measure*, and *Accuracy*. Precision gives information about which percentage of predicted labels are correctly classified, while Recall gives information about which percentage of true labels are correctly classified. F1-measure shows the performance of the target class in terms of tradeoff between Precision and Recall and is computed as their harmonic average. Among them, we give a special attention to F1-measure while evaluating the overall performance due to its ability to cope with unbalanced classes. This is particularly important in our case because some activities occur more frequently than others in both the van Kasteren and the CASAS datasets. *Accuracy*, on the other hand, is affected more from the classes with higher occurrence.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.2}$$



(a) Raw (b) Changepoint (c) Last-fired

Figure 3.3: Feature representations (Taken from (van Kasteren et al., 2010b))

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.3}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3.4}$$

Here TP and FP are the fraction of true and false positives respectively, while TN and FN are the fraction of true and false negatives respectively. We also consider *Class Accuracy* to represent the average percentage of correctly classified timeslices per class as follows:

$$\text{Class: } \frac{1}{C} \sum_{c=1}^{C} \frac{\sum_{n=1}^{N_c} [\text{inferred}_c(n) = \text{true}_c(n)]}{N_c} \tag{3.5}$$

where $[a = b]$ is a binary indicator returning 1 when true and 0 otherwise. C is the number of classes and N_c is the total number of time slices for class c.

Chapter 4

Activity Recognition

The focus of this chapter is to introduce a novel technique that can provide better activity recognition performance compared to well-known methods. To this aim, we present a model that relies on sequential pattern mining. We start with detailing activity recognition approaches to be used in our experiments. Among possible approaches discussed in Section 2.1.1, we especially focused on Hidden Semi-Markov Models. As a segmental labeling approach, HSMM was shown to be successful in activity recognition problems with its ability to model duration information given the fact that an activity spans a certain amount of time the average duration of which depends on the specific activity being performed. In addition, HSMM allows efficient incorporation of patterns owing to the semi-Markovianity. Note that although we concentrated on HSMM, our approach could be easily integrated into other segmental labeling approaches. Different feature representations are then evaluated in terms of the selected predictive models to determine a baseline for comparison purposes and to decide representations on which patterns will me mined. In the following phase, we show how to mine patterns for a given representation. Finally, we propose using extracted patterns to develop an improved predictive model (see Section 4.4).

4.1 Recognition Algorithms

An effective approach for performing activity recognition on temporal data should be able to model the relationships between time instants and between their respective labels. Hidden Markov Models (Rabiner, 1989) and Hidden Semi-Markov Models (Yu, 2010) are directed graphical models which have been successfully used to perform sequential and segmental labeling respectively. To see how well these models perform against nontemporal models, we put Naive Bayes Model (Peng, 2009) as a baseline into the evaluation. These algorithms have been recently compared (van Kasteren et al., 2010a; van Kasteren et al., 2010b) on a benchmark consisting of wireless sensor network data. All three algorithms model the joint probability distribution of input and output $p(\mathbf{x}, \mathbf{y})$ and return the output maximizing this probability, $\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{x}, \mathbf{y})$.

A Naive Bayes (NB) approach assumes that all data points, i.e. time instants, are independently and identically distributed. Therefore, it does not model temporal relations between sensor events. The joint probability of observations and labels can be factorized as:

$$p(\mathbf{x}, \mathbf{y}) = \prod_{t=1}^{T} p(\mathbf{x}_t | y_t) p(y_t) \tag{4.1}$$

where $p(y_t)$ is a prior probability over activities. In modeling the conditional probability of observations given a label, a common simple approach (also followed in (van Kasteren et al., 2010a; van Kasteren et al., 2010b)) consists of making a Naive assumption of independence between observation features given the label. The resulting probability is:

$$p(\mathbf{x}_t | y_t) = \prod_{i=1}^{N} p(x_t^i | y_t) \tag{4.2}$$

where for the binary case, probabilities for features $p(x_t^i | y_t)$ are represented as Bernoulli distributions.

A Hidden Markov Model (HMM) is a sequential approach where: 1) the label at each time instant depends on the label at the previous time instant only; 2) the observation at each time instant depends on the label at that time instant only; 3) probabilities do not depend on the specific time instants but only on the values of labels/observations at those instants. The resulting joint probability is given by:

$$p(\mathbf{x}, \mathbf{y}) = \prod_{t=1}^{T} p(\mathbf{x}_t | y_t) p(y_t | y_{t-1}) \tag{4.3}$$

where $p(y_1 | y_0)$ stands for the probability of having y_1 as the initial label. We follow the same assumption for computing the observation probability $p(\mathbf{x}_t | y_t)$ as presented in Eq. (4.2).

HMMs imply an exponential distribution for state durations. The probability of seeing label l for d consecutive instants is d times the probability of a self transition $p(y_t = l | y_{t-1} = l)$. This assumption is often not appropriate when durations tend to have specific patterns, as happens in activity recognition tasks. Explicit duration distributions can be represented by Hidden Semi-Markov Models (HSMM), which consider probability of segmental labeling $(\mathbf{x}, \mathbf{s})$. Recall that $\mathbf{s}$ is a sequence of U consecutive segments $s_u = (b_u, e_u, y_u)$. The corresponding joint probability is represented as:

$$p(\mathbf{x}, \mathbf{s}) = \prod_{u=1}^{U} p(y_u | y_{u-1}) p(d_u | y_u) p(\mathbf{x}_{b_u : e_u} | y_u) \tag{4.4}$$

where $d_u = e_u - b_u + 1$ is the duration of segment s_u, probability of which can be modeled by the desired distribution. Following (van Kasteren et al., 2010a), we employed a histogram distribution with 5 bins. Concerning the probability of a certain observation segment given its label, it is commonly computed (again, see e.g. (van Kasteren et al., 2010a)) as the product of the probabilities of its time instants:

$$p(\mathbf{x}_{b_u:e_u}|y_u) = \prod_{t=b_u}^{e_u} p(\mathbf{x}_t|y_u) \tag{4.5}$$

where $p(\mathbf{x}_t|y_u)$ is further decomposed as in Eq. (4.2). However, additional knowledge on the dynamics of a certain activity could help devising a more complex probabilistic model for the sequence of observations measured while performing it. We will use mined activity patterns to account for longer range dependencies.

Note that the approach that we introduce for directed graphical models can be straightforwardly applied to their undirected counterparts, Conditional Random Fields (Sutton and McCallum, 2007) with their Semi-Markov extension (Sarawagi and Cohen, 2004). We did not include them in our comparison, as they require much longer training time and were shown to provide comparable and often worse results with respect to their directed counterparts (HMM and HSMM) on this benchmark (van Kasteren et al., 2010a).

4.2 Evaluation of Feature Representations

Classification performances of predictive models differ greatly for varying feature representations. Evaluation of these differences reveals associations between dataset properties (e.g. sensor structure) and features, which is crucial in defining the representation on which patterns will be mined. Besides, determining a robust representation is of importance to form a baseline for comparison purposes. Figures 4.1, 4.2, and 4.3 report F1 measures for NB, HMM, and HSMM respectively using different feature representations for all experimental datasets. HMM and HSMM outperform NB in many cases. Both HMM and HSMM produce similar results, while HSMM performs slightly better than HMM. Therefore, we provided the following in-depth analysis based on the results of HSMM.

The Raw representation (R) is substantially worse than all others. In the van Kasteren dataset for instance, a sensor attached on a door that is left open after completion of an activity keeps firing continuously while other activities are being performed. In such situations, the R representation is incapable of capturing ongoing activities as it tends to have traces of already completed ones. Similarly, in the CASAS dataset, motion sensors tend to keep firing for a long time after the position was left, possibly due to problems in the sensor measurements. This

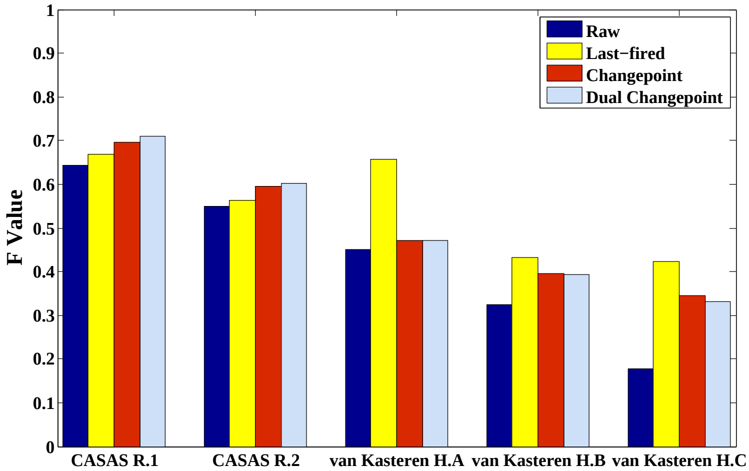


Figure 4.1: Comparison of feature representations for Naive Bayes

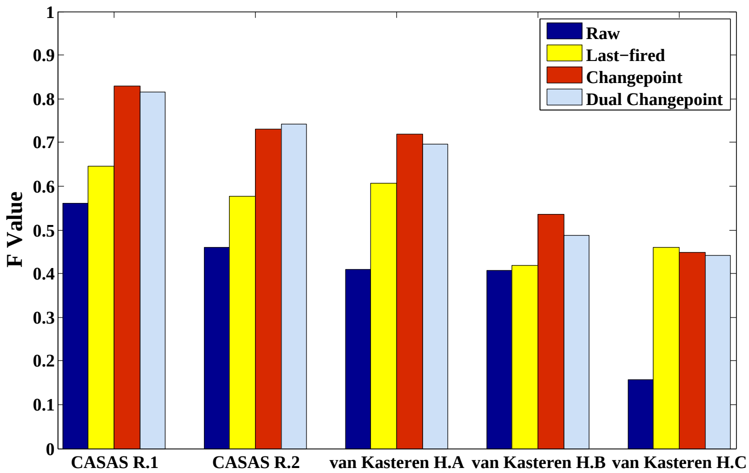


Figure 4.2: Comparison of feature representations for Hidden Markov Model

leads to multiple overlapping sensor activations, eventually deteriorating trajectory information. We thus did not consider this representation in the rest of the experimental evaluation. Detailed results of C and L representations that are used to create the figures, on the other hand, are presented in Tables 4.2 and 4.3 for the van Kasteren and the CASAS datasets respectively.

The L representation prevents these degenerate behaviors by focusing on the last sensor changing its state, forgetting about the state of previously recorded sensors. On the other hand, this can propagate a sensor activation too long if no other sensor is observed. This problem is especially relevant when an activity is followed by "idle", where no sensor fires. This case is mostly observed in the CASAS dataset, where a number of activities (e.g. Sleeping, Watching TV) do not produce movement sensor firings. The C representation tends to provide the best results on average. The DC representation has very similar results to the C one while

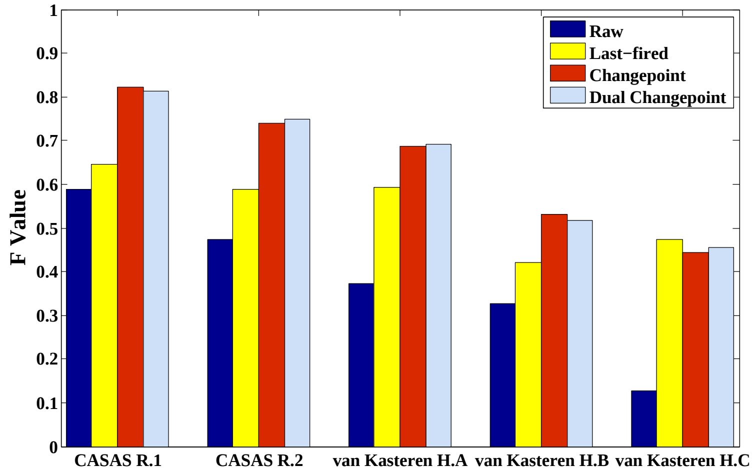


Figure 4.3: Comparison of feature representations for Hidden Semi-Markov Model

being slightly more computationally expensive due to additional features. We thus used the C representation as a baseline in all following experiments.

While C and DC representations provide reasonable information for all types of sensors, L one is meaningful for object interaction sensors (e.g. toilet light switch activation remaining active indicates a 'using toilet' activity), mostly found in the van Kasteren dataset, but is misleading for the motion sensors characterizing the CASAS dataset (due to deterioration in the trajectory information). Indeed L representation is always harmful in the CASAS dataset, as will be seen in the experimental results. We mine patterns from C and L representations since they contain different information depending on the dataset (see Section 4.4.2).

4.3 Segmental Pattern Mining

Our aim is mining patterns characterizing timespans during which a certain activity is performed. Algorithm 1 shows the pseudocode of our segmental pattern miner. Training sequences are first split into activity segments, each labeled with the corresponding activity. These segments are fed to a sequential pattern miner (procedure SequentialMiner). We employed PBOOST (Nowozin et al., 2007) which supports discriminative mining, i.e. mining of patterns distinguishing sequences of a certain class from the others. The algorithm takes as input sets of positive and negative examples, each example being a sequence of sets of integers (the sensor identifiers in our case). A pattern is itself a sequence of sets of integers. The algorithm mines for patterns matching positive and not negative examples. Let $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m)$ be a pattern of length m. Let $\mathbf{p}_i$ be a pattern element, corresponding to a non-empty set of active sensors. The pattern $\mathbf{p}$ matches sequence $\mathbf{x}$ if there is a match $(t_1, t_2, \dots, t_m)$ such that: for all $i > j$, $t_i > t_j$; for all i, $\mathbf{x}_{t_i}$ contains active sensors $\mathbf{p}_i$, i.e.

Algorithm 1 Procedure for segmental pattern mining

Input:
$\mathcal{D}$: input-output sequences
ϕ: pattern selection threshold
Output:
$\mathcal{P}$: patterns for activities along with their gap sizes

procedure SEGMENTALMINER($\mathcal{D}$, ϕ)
  Initialize $\mathcal{P}$ to the empty set
  Split training sequences into activity segments $\mathcal{S}$
  for all activities y **do**
    $\mathcal{S}_y \leftarrow$ segments for y
    $\mathcal{S}_{\bar{y}} \leftarrow$ segments for $y' \neq y$
    $\mathcal{P}_y \leftarrow \textsc{SequentialMiner}(\mathcal{D}(\mathcal{S}_y), \mathcal{D}(\mathcal{S}_{\bar{y}}))$
    for all $\mathbf{p} \in \mathcal{P}_y$ **do**
      if $\textsc{Score}(\mathbf{p}, \mathcal{D}(\mathcal{S}_y), \mathcal{D}(\mathcal{S}_{\bar{y}})) \geq \phi$ **then**
        $g \leftarrow \textsc{MedianGap}(\mathbf{p}, \mathcal{D}(\mathcal{S}_y))$
        $\mathcal{P} \leftarrow \mathcal{P} \cup \{(\mathbf{p}, g)\}$
  return $\mathcal{P}$
end procedure

$\forall \mathbf{p}_i^k \in \mathbf{p}_i$, $\mathbf{x}_{t_i}^{\mathbf{p}_i^k} = 1$, where $\mathbf{x}_{t_i}^{\mathbf{p}_i^k}$ is the value of sensor $\mathbf{p}_i^k$ in the observation vector for time instant t_i. A *gap* is defined as a sequence of time instants separating two consecutive pattern element matches from each other. We define gap length g as the overall sum of time instants occuring between consecutive pairs of element matches along the pattern. More formally, pattern $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m)$ matches a sequence $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ with gap length g if there exist time instants $1 \leq t_1 \leq t_2 \leq \cdots \leq t_m \leq n$ such that $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m$ are respectively contained in $\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \ldots, \mathbf{x}_{t_m}$ and $t_m - t_1 + 1 - m = g$. By defining a canonical ordering for sequences, the pattern space is searched in a tree-based fashion starting from the empty pattern (Pei et al., 2004). Pruning of the search space is conducted by combining the standard notion of *support* (i.e. number of matching sequences) with that of *gain*: PBOOST considers each pattern as a feature and learns a linear classifier (LPBoost (Demiriz et al., 2002)) on top of them, discriminating between positive and negative examples. The gain provided by a feature can be compared with an upper bound on the maximal gain achievable by further extending the corresponding pattern. If the current gain exceeds or equals the upper bound there is no need to proceed in this search direction.

For each of the possible activities, PBOOST is run providing segments of the target activity as positive examples and all other segments as negative ones. Each of the returned patterns $\mathbf{p}$ is evaluated according to its discriminative power, computed as its F1 score on the training segments (procedure SCORE). The F1 measure trades off precision, i.e. the fraction of segments covered by the pattern which do belong to the target activity, and recall, i.e. the

Algorithm 2 Procedure computing median gap length

Input:
$\mathbf{p}$: a pattern for a specific activity
$\mathcal{D}(\mathcal{S}_y)$: set of training segments for that activity

Output:
$\textsc{Median}(\mathcal{G})$: gap length of the pattern

procedure $\textsc{MedianGap}(\mathbf{p}, \mathcal{D}(\mathcal{S}_y))$
  Initialize $\mathcal{G}$ to the empty set
  $m \leftarrow \textsc{length}(\mathbf{p})$
  for all $\mathbf{x} \in \mathcal{D}(\mathcal{S}_y)$ **do**
    $g_{max} \leftarrow -1$
    for all $t_1 \in [1, T]$ **do**
      $(M_1, M_2) \leftarrow \textsc{longestMatch}(\mathbf{x}, t_1, \mathbf{p}, T - m)$
      $g \leftarrow M_2 - M_1 + 1 - m$
      if $g > g_{max}$ **then** $g_{max} \leftarrow g$
    if $g_{max} > -1$ **then**
      $\mathcal{G} \leftarrow \mathcal{G} \cup \{g_{max}\}$
  return $\textsc{Median}(\mathcal{G})$
end procedure

fraction of segments of the target activity covered by the pattern. All patterns with a score lower than a certain threshold ϕ are discarded.

Sequential patterns extracted by $\textsc{pboost}$ allow for arbitrary gaps between the pattern elements. Our aim is that these patterns cover the largest possible portion of an activity segment. However, in the test phase when used to label a novel time sequence, the activity segmentation will be unknown and the patterns will be applied to the whole sequence (i.e. a day). We thus need to estimate the expected number of gaps in pattern matches for activity segments. This is crucial for a correct use of patterns: during test, allowing for arbitrary gaps within a pattern could produce matches involving very distant time instants (e.g. in the morning and afternoon), which likely belong to different activity segments. The procedure $\textsc{MedianGap}$ described in Algorithm 2 estimates how many gaps should be expected on average for a match covering the largest possible portion of an activity segment. Given a pattern characterizing a certain activity, and a set of training segments for that activity, the procedure finds the longest possible match of the pattern on each segment and computes the corresponding gap length. Let $(\mathbf{p}, g)$ be a pattern $\mathbf{p} = (\mathbf{p}_1, \ldots, \mathbf{p}_m)$ of length m with gap length g. A g-gap-bounded match $(t_1, \ldots, t_m)$ for the pattern is a sequence of time instants such that: for all $i > j$, $t_i > t_j$; for all i, $\mathbf{p}_i$ is contained in active sensors of $\mathbf{x}_{t_i}$; the sequence has at most g gaps, i.e. $t_m - t_1 + 1 \leq m + g$. The procedure $\textsc{longestMatch}(\mathbf{x}, t_1, \mathbf{p}, g)$ finds the longest g-gap-bounded match of pattern $\mathbf{p}$ in sequence $\mathbf{x}$ starting at instant t_1. The procedure is used here for $g = T - m$, i.e. asking to cover the largest possible portion of the entire input sequence. As will be seen in Section 4.4, the same sub-routine is also

Algorithm 3 Longest pattern match computation

Input:
- $\mathbf{x}$: a sequence to be matched
- t_1: starting instant of the sequence
- $\mathbf{p}$: a pattern for a specific activity
- g: gap length of the pattern $\mathbf{p}$

Output:
- (M_1, M_2): initial and final positions of the match

```
procedure longestMatch(x, t_1, p, g)
    m ← length(p)
    if not Match(x_{t_1}, p_1) then
        return (0, −1)
    if m > 2 then
        c ← t_1 + 1
        for all u ∈ [2, m − 1] do
            while c − t_1 < u + g do
                if Match(x_c, p_u) then
                    break
                c ← c + 1
            if c − t_1 ≥ u + g then
                return (0, −1)
    else
        c ← t_1
    v ← t_1 + m + g − 1
    while v > c do
        if Match(x_v, p_m) then
            return (t_1, v)
        v ← v − 1
    return (0, −1)
end procedure
```

employed during inference to return the largest possible pattern match within the estimated gap length. The procedure is detailed in Algorithm 3. The function $\textsc{Match}(x_c, p_u)$ checks whether a single position matches with a pattern element, i.e. $\mathbf{x}_c$ contains active sensors $\mathbf{p}_u$. The procedure verifies that the first $u \in [1, m-1]$ patterns are matched within their allowed lengths $u + g$, with the first pattern matching position t_1. Then it searches for the longest possible match by matching the last pattern $\mathbf{p}_m$ to a position as close as possible to the overall maximal allowed length $m+g$. The initial (M_1) and the final (M_2) positions of the match are returned as the border points. If the pattern does not match with the sequence, border points are assigned to $(0, -1)$. The median of the gap length computed over all activity segments is returned by the $\textsc{MedianGap}$ procedure as an estimate of the gap length which should be expected for a pattern match covering the longest possible portion of an activity segment. The final outcome of the segmental pattern mining algorithm is a set of patterns characterizing all activities, together with their estimated gap lengths.

Figure 4.4 (a) and (b) illustrate the pattern mining process and the usage of gaps during pattern match respectively. Three input sequences, corresponding for instance to three days, are split into segments for activities a_1, a_2 and a_3. A time instant is either made up of sensor identifiers of interacted objects or empty in the absence of sensor interactions. Having provided the segments of activity a_1 as positive examples and the others as negative examples, the algorithm finds discriminative patterns (represented by solid and dashed squares) in the form of object interactions for a_1. Let us assume that MEDIANGAP procedure returns three as the gap length of the pattern (cd, bc, fr). 3-gap-bounded matches are searched over novel segments. The pattern (cd, bc, fr) matches the three segments with gap lengths 3, 5 and 2 respectively. Only segments one and three are thus 3-gap-bounded matches for the pattern (boldface) while match for segment two exceeds the limits.

Note that our algorithm is not bound to the specific mining technique, and can be fed with patterns obtained by any sequential pattern mining approach (see e.g. (Hirate and Yamana, 2006; Zhu and Wu, 2007; Li et al., 2012)).

4.4 Pattern-based Hidden Semi-Markov Model

Patterns extracted during the mining phase are integrated into a probabilistic segmental labeling algorithm, providing improved capacity to model longer-term dependencies by allowing gaps between matches of individual pattern elements. We introduce a probabilistic duration model representing the distribution of pattern matches along sequences, and integrate it into a Hidden Semi-Markov Model (HSMM).

We begin by showing how to identify pattern matches within a sequence (Algorithm 4). The algorithm takes as inputs a pattern with its estimated gap length and the sequence to be scanned for matches, e.g. a full day, and outputs a list of segments representing pattern matches. For all possible starting instants t_1, it uses the LONGESTMATCH$(\mathbf{x}, t_1, \mathbf{p}, g)$ procedure to compute the longest possible match with at most g gaps. The rationale is that the pattern match should try to cover the longest possible time span, within the estimated limits characterizing the cover of an activity segment by that pattern. The lower and upper borders of this match are added to the list of matching segments, unless the segment overlaps with the previously inserted one (recovered by TOP), in which case the two are merged into a single match spanning both segments.

Pattern matches are used to compute the probability that a certain pattern covers a sequence segment given the segment label. Let $C^{(\mathbf{p},g)}_{\mathbf{x}_{b_u:e_u}}$ be a boolean random variable modeling approximate coverage of pattern $(\mathbf{p}, g)$ over segment $\mathbf{x}_{b_u:e_u}$, i.e.:

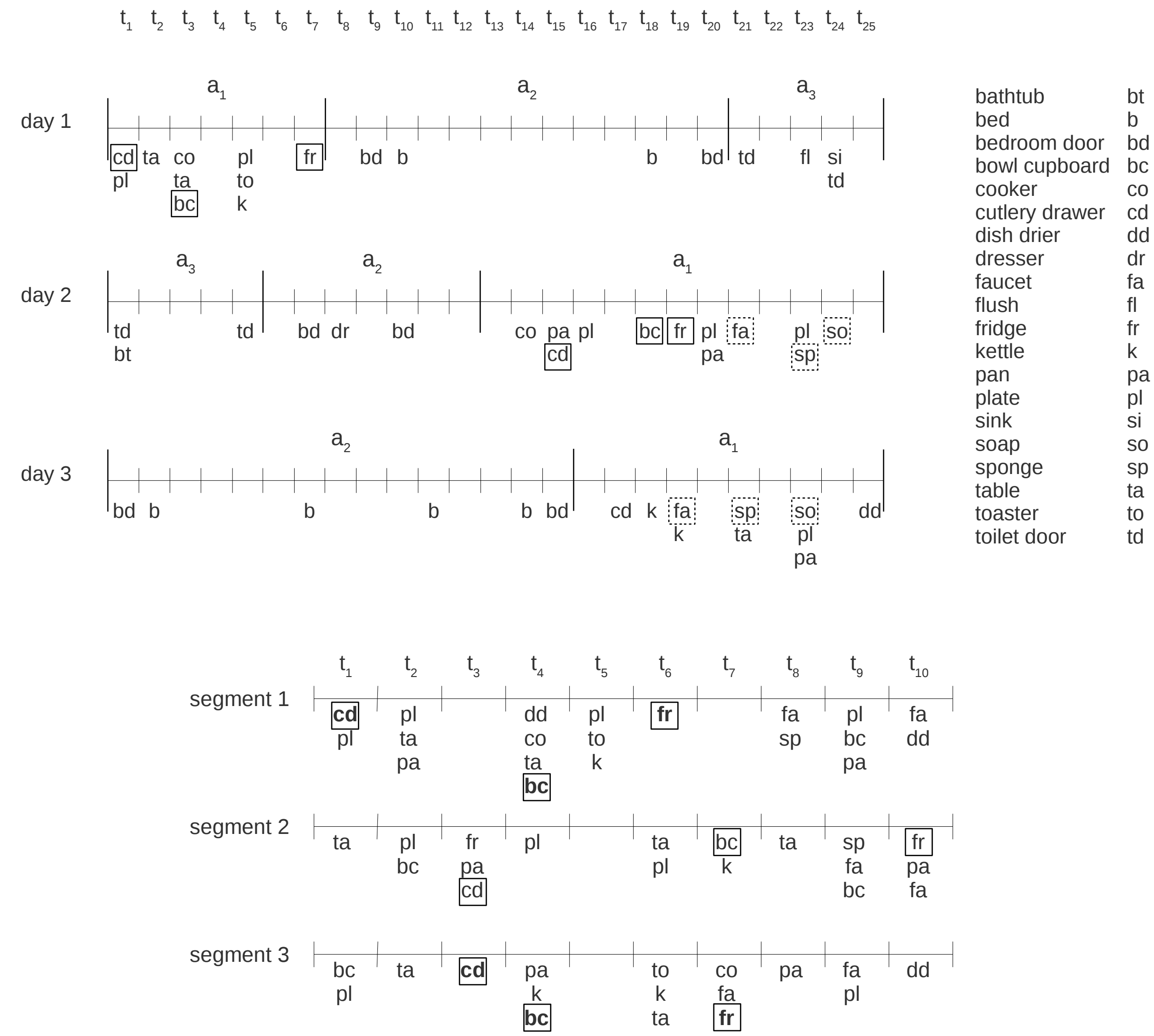


Figure 4.4: Example of discriminative pattern mining output (a-top): the miner finds two sequential patterns discriminating segments for activity a_1 from segments for other activities. 3-gap-bounded match (b-bottom): segments one and three match pattern (cd, bc, fr) with three and two gaps respectively, while match for segment two contains five gaps and is thus not a 3-gap-bounded match.

Algorithm 4 Procedure for recovering pattern matches

Input:
$\mathbf{p}$: a pattern for a specific activity
g: gap length of the pattern $\mathbf{p}$
$\mathbf{x}$: sequence to be scanned for matches

Output:
$\mathcal{M}$: list of matching segments for the pattern $\mathbf{p}$

```
procedure PATTERNMATCHES(p,g,x)
    Initialize M to the empty list
    for all t_1 ∈ [1, T] do
        (M_1, M_2) ← LONGESTMATCH(x,t_1,p,g)
        if M_2 > −1 then
            if M is empty then
                add (M_1, M_2) to M
            else
                (M'_1, M'_2) ← TOP(M)
                if M_1 ≤ M'_2 then
                    replace M'_2 with MAX(M'_2, M_2)
                else
                    add (M_1, M_2) to M
    return M
end procedure
```

$$C^{(\mathbf{p},g)}_{\mathbf{x}_{b_u:e_u}} = \begin{cases} 1 & \text{if } \textsc{cover}(\mathbf{x}_{b_u:e_u}, \mathbf{p}, g, \tau) \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

Here $\textsc{cover}(\mathbf{x}_{b_u:e_u}, \mathbf{p}, g, \tau)$ is true if segment $\mathbf{x}_{b_u:e_u}$ is approximately *covered* by pattern $\mathbf{p}$. This happens if there is a g-gap-bounded match of the pattern spanning almost all the segment (with a threshold τ defining the desired approximation). Largest coverage of activity segments was indeed the driving principle when mining patterns on known activity segments. The (approximate) coverage is formally defined as:

$$\textsc{cover}(\mathbf{x}_{b_u:e_u}, \mathbf{p}, g, \tau) \iff \frac{d_I}{d_u} \geq \tau \tag{4.7}$$

where

$$d_I = \sum_{(b_v,e_v)\in\mathcal{M}^{(\mathbf{p},g)}} |(b_v, e_v) \cap (b_u, e_u)| \tag{4.8}$$

is the fraction of segment (b_u, e_u) covered by any match of pattern $\mathbf{p}$ and τ is the desired coverage approximation, which was set to 0.9 in our experiments (similar values generated similar results, while choosing substantially lower coverages produced performance worsening). Note that the summation runs over disjoint matches as overlapping ones have already been merged by the PATTERNMATCHES procedure. The $\text{COVER}(\mathbf{x}_{b_u:e_u}, \mathbf{p}, g, \tau)$ procedure returns true also if segment $\mathbf{x}_{b_u:e_u}$ is contained in a g-gap-bounded match as a proper subsequence, i.e. the match exceeds the borders of the segment. This is consistent with the fact that pattern matches are maximal within the g-gap-length limit. A probabilistic model of match duration, combined with the standard segment duration probability of HSMM, contributes to determine during inference the optimal segmentation according to the learned probabilities, as will be discussed in the following.

The pattern-based HSMM model is obtained by modifying the conditional probability of seeing an observation segment given a segment label (Eq. 4.5) in order to include evidence concerning patterns:

$$p(\mathbf{x}_{b_u:e_u}|y_u) = \prod_{t=b_u}^{e_u} p(\mathbf{x}_t|y_u) p(C_{\mathbf{x}_{b_u:e_u}}^{(\mathbf{p},g)^{(1)}}, \ldots, C_{\mathbf{x}_{b_u:e_u}}^{(\mathbf{p},g)^{(m)}}|y_u) \tag{4.9}$$

where $m = |\mathcal{P}|$ is the number of patterns and the joint probability ranges over all patterns $(\mathbf{p}, g)^{(i)} \in \mathcal{P}$. As for the term modeling sensor activations only, we make a Naive Bayes assumption of independency between patterns given the segment label. This substantially simplifies the probabilistic model, allowing for efficient inference as will be shown in Section 4.4.1. Note however that this assumption can be easily violated when patterns share common elements. We decided to trade off expressivity for tractability, but a more complex probabilistic model could be conceived. The simplified probability becomes:

$$p(\mathbf{x}_{b_u:e_u}|y_u) = \prod_{t=b_u}^{e_u} p(\mathbf{x}_t|y_u) \prod_{(\mathbf{p},g)\in\mathcal{P}} p(C_{\mathbf{x}_{b_u:e_u}}^{(\mathbf{p},g)}|y_u) \tag{4.10}$$

The conditional probability of observing a pattern coverage is computed as the product of the probability of a match given the activity y_u under consideration, times the probability that the (approximately) covered segment has a certain duration d_u. Given a boolean random variable $M_{\mathbf{x}_{b_u:e_u}}^{(\mathbf{p},g)}$ indicating a pattern match and a random variable $D_{\mathbf{x}_{b_u:e_u}}^{(\mathbf{p},g)}$ modeling the duration of the covered segment, the probability can be written as:

$$\begin{aligned} p(C_{\mathbf{x}_{b_u:e_u}}^{(\mathbf{p},g)} = 1|y_u) &= P(M_{\mathbf{x}_{b_u:e_u}}^{(\mathbf{p},g)} = 1, D_{\mathbf{x}_{b_u:e_u}}^{(\mathbf{p},g)} = d_u|y_u) \\ &= P(M_{\mathbf{x}_{b_u:e_u}}^{(\mathbf{p},g)} = 1|y_u) P(D_{\mathbf{x}_{b_u:e_u}}^{(\mathbf{p},g)} = d_u|M_{\mathbf{x}_{b_u:e_u}}^{(\mathbf{p},g)} = 1, y_u) \end{aligned} \tag{4.11}$$

The former term is estimated during training as the fraction of y_u activity segments matching the pattern. The latter can be modeled with any appropriate duration distribution. For consistency in our experiments we choose the same distribution used for modeling segment duration, i.e. a multinomial distribution over n_b duration bins. Given the longest match duration d_{max} found in the training set, a uniform bin width bw is computed as:

$$bw = \max(1, \frac{d_{max}}{n_b}) \tag{4.12}$$

Probabilities for each bin are then computed as normalized counts of the training activity segments whose duration fall into the bin. The duration probability for a novel segment then corresponds to the probability of the bin that d_u falls into. We run experiments for varying number of bins (from 3 to 15), achieving similar performance and consistent comparative behavior. All reported results are for $n_b = 5$.

Given a test sequence, decoding consists of finding the most probable sequence of activity segments, i.e.:

$$\mathbf{s}^* = \arg\max_{\mathbf{s}} p(\mathbf{x}, \mathbf{s}) \tag{4.13}$$

which boils down to identify sequences of activity labels and segment durations. The problem can be addressed by the well-known Viterbi algorithm (Yu, 2010) appropriately modified to account for the novel pattern-based model.

A recursive procedure computes for each time instant t a value $\delta_t(y_j, d)$, representing the probability that activity y_j is performed in the time segment $(t - d + 1, t)$, given the most probable segmentation and activity assignment for all past time instants. This is obtained by recursively maximizing over duration d' and activity label y_i of the previous segment:

$$\delta_t(y_j, d) = \max_{1 \le y_i \le L} \left(\max_{1 \le d' \le D} \left(\delta_{t-d}\left(y_i, d'\right)\right) p(y_j|y_i) \right) \times p(d|y_j) p(\mathbf{x}_{t-d+1:t}|y_j) \tag{4.14}$$

where segment observation probabilities $p(\mathbf{x}_{t-d+1:t}|y_j)$ are computed according to Equation 4.10. The maximum possible duration D is estimated as the maximum duration of all activity segments in the training set. The base step of the recursion is computed for segments starting at the beginning of the sequence (i.e. $t' \le d$) as:

$$\delta_{t'}(y_j, d) = p(d|y_j) p(\mathbf{x}_{t'-d+1:t'}|y_j) \pi_0(j) \tag{4.15}$$

where $\pi_0(j)$ is the initial probability for activity y_j, computed as the fraction of training days starting with that activity. At the end of the recursion, the probability of the best activity assignment for the whole sequence is computed as:

$$p^* = \max_{1 \leq y_j \leq L} \left(\max_{1 \leq d \leq D} \delta_T(y_j, d) \right) \tag{4.16}$$

In order to recover the activity assignment corresponding to p^*, an auxiliary variable $\psi_t(y_j, d)$ is used to keep information on the configuration originating $\delta_t(y_j, d)$, i.e. the most probable previous activity label and segment duration (y_i^*, d'^*) in case time segment $(t - d + 1, t)$ is labeled with activity y_j:

$$\begin{aligned} \psi_t(y_j, d) &= (y_i^*, d'^*) \\ &= \arg\max_{1 \leq y_i \leq L} \left(\arg\max_{1 \leq d' \leq D} \left(\delta_{t-d} \left(y_i, d' \right) \right) p(y_j | y_i) \right) \end{aligned} \tag{4.17}$$

Note that terms outside the maximization were discarded here as they are irrelevant for deciding what the maximal configuration is. The best sequence of activity segments is recovered backtracking through these variables, i.e.:

$$\begin{aligned} (y_T^*, d_T^*) &= \arg\max_{1 \leq y_T \leq L} \left(\arg\max_{1 \leq d_T \leq D} \delta_T(y_T, d_T) \right) \\ (y_{T-d}^*, d_{T-d}^*) &= \psi_{t-d}(y_T^*, d_T^*) \\ &\dots \end{aligned} \tag{4.18}$$

For a detailed description of inference for plain HSMM models, see (Yu, 2010). Our version differs in the probability of observing a certain segment given its predicted label, i.e. Eq. 4.10. We can efficiently compute this probability by keeping for each pattern $\mathbf{p}$ some auxiliary structures throughout the inference process, for t ranging from 1 to T: $\mathcal{M}$ is the list of matches for pattern $\mathbf{p}$ (we dropped the superscript to avoid clumsy notation), pre-computed using the PATTERNMATCHES procedure; seg is the index of the currently active segment, initialized at the first segment (or at zero if there are no matches for pattern $\mathbf{p}$); $match$ is a flag indicating whether the current time instant t is after the beginning of the active segment seg (i.e. $\mathcal{M}^1_{seg} < t \leq \mathcal{M}^2_{seg}$), or not (i.e. $t \leq \mathcal{M}^1_{seg}$); $\mathbf{cov}$ is a zero-initialized vector of length D which is kept updated so that at time instant t, cov_d contains the coverage of the segment $[t-d, t]$ by pattern matches (for $d = [0, D-1]$). The procedure COVERAGE$(\mathcal{M}, \mathbf{cov}, seg, match, t)$ updates these structures at each time instant t of the inference process. Algorithm 5 describes the update. First, the **cov** vector is shift of one position to the right, filling the first position

Algorithm 5 Procedure for updating coverage

Input:
$\mathcal{M}$: list of matching segments for the pattern $\mathbf{p}$
$\mathbf{cov}$: vector representing coverage of a segment by pattern matches
seg: index of the current active segment
$match$: flag indicating the position of t wrt. seg
t: current time instant

Output:
$\mathbf{cov}$: updated $\mathbf{cov}$
seg: updated seg
$match$: updated $match$

procedure COVERAGE($\mathcal{M}, \mathbf{cov}, seg, match, t$)
  if $seg = 0$ **then return** $null$ ▷ end of $\mathcal{M}$ reached
  Shift $\mathbf{cov}$ one pos to right adding zero
  if not $match$ **then**
    if $\mathcal{M}^1_{seg} = t$ **then** ▷ t first pos in seg
      $match \leftarrow True$
  else if $\mathcal{M}^2_{seg} < t$ **then** ▷ t after last pos in seg
    $match \leftarrow False$
    $seg \leftarrow 0$
    for all $nseg \in [seg+1, |\mathcal{M}|]$ **do**
      if $\mathcal{M}^1_{nseg} \geq t$ **then**
        $seg \leftarrow nseg$
        if $\mathcal{M}^1_{nseg} = t$ **then**
          $match \leftarrow True$
        break
  if $match$ **then** ▷ update coverage
    increment all elements of $\mathbf{cov}$ by 1
  return $(\mathbf{cov}, seg, match)$
end procedure

with a zero, in order to account for the increased time instant. If the current segment is not matched, the algorithm checks whether t corresponds to its first position ($\mathcal{M}^1_{seg}$), otherwise it checks whether the current match is lost (i.e. $\mathcal{M}^2_{seg} < t$). In this latter case, it searches for the next active segment, updating $match$ in case t corresponds to its starting position. Finally, if $match$ is true (i.e. t is in a pattern segment match), the **cov** vector is updated by one. Note that COVER($\mathbf{x}_{t-d:t}, \mathbf{p}, g, \tau$) can be easily computed as $(cov_d/d) \geq \tau$ for all d in $[0, D-1]$.

4.4.1 Computational Complexity

Complexity of the LONGESTMATCH is $O(\ell m N)$, where m is the pattern length, $\ell = m + g$ is the overall maximal length of a possible match and N is the number of sensors. We assume here that the matching procedure MATCH is linear in the number of sensors for a pattern element, as sensor activations are stored in lookup tables thanks to the limited size of N. Complexity of PATTERNMATCHES is thus $O(T\ell mN)$, and recovering all maximal matches for all patterns costs $O(PT\ell mN)$ with P the total number of mined patterns. This is a loose upper bound on the actual complexity, as for instance the number of pattern match evaluations is typically much smaller than ℓ. We experimentally verified that the cost of this procedure is negligible with respect to the overall cost of inference. Complexity of MEDIANGAP is $O(|\mathcal{S}|T^2mN)$ where $|\mathcal{S}|$ is the number of activity segments, and the overall complexity of median gap computation for all patterns is $O(P|\mathcal{S}|T^2mN)$. The procedure actually runs in time comparable to the PATTERNMATCHES one, given the average length difference between activity segments used in the former and full days used in the latter (difference ignored in the asymptotic analysis).

The inference step of plain HSMM has complexity $O(TL^2D)$ where L is the number of activities (states) and D their maximal duration. The COVERAGE procedure has complexity $O(D)$. The pattern-related portion thus contributes with $O(TPD)$ to the overall inference complexity, which becomes $O(T(L^2 + P)D)$. Given that the number of patterns P is usually smaller the squared number of states, the complexity of inference is basically unaffected.

The computational bottleneck of the overall approach is the pattern mining step, i.e. the SEQUENTIALMINER procedure. Discriminative pattern mining is especially expensive as the anti-monotonic property typically used in standard mining does not hold. The PBOOST algorithm, for instance, needs to train a linear classifier in order to compute the discriminant power of the patterns during the mining procedure. The cost of the mining procedure widely varies on the different scenarios, strongly depending on the degree of sparsity of the sensor activations, but can be one or two orders of magnitude slower than the inference process in the worst cases. Sequential pattern mining is a popular research area, and a number of alternative approaches have been suggested in the literature (see (Mabroukeh and Ezeife, 2010) for a recent review). We did not investigate the performance of the different algorithms in our activity recognition scenario, e.g. discriminative vs non-discriminative approaches, as our contribution is focused on proposing a probabilistically sound framework to incorporate them. Further research on these efficiency issues is anyhow necessary in order to allow segmental pattern mining over large-scale datasets.

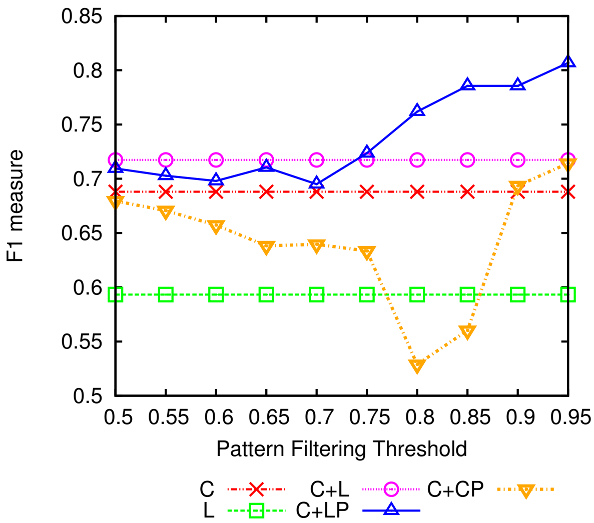
4.4.2 Experiments

In this section, we present a series of experiments to evaluate the performance of pattern-based HSMM (PHSMM) as opposed to baseline feature representations with plain HSMM, and to appraise the usefulness of patterns. A 'Leave-one-day-out' (LOO) approach was used to split the datasets, i.e. the van Kasteren and the CASAS, into training and test sets. Each day was in turn considered as a test set, while all other days made up the training set. We evaluated the performance of the proposed system for each class by using standard measures, i.e. precision, recall, F1, and accuracy as detailed in Section 3.4.

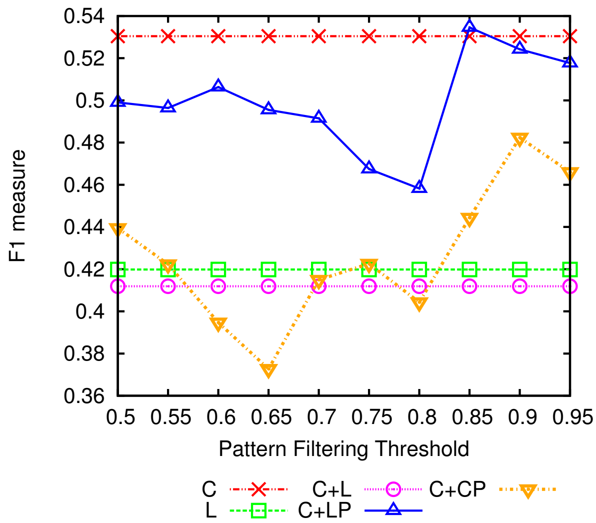
Figures 4.5 and 4.6 report results of pattern-based HSMM (PHSMM) for increasing values of the threshold (ϕ) over pattern discriminative power, as compared to a set of baseline alternatives with plain HSMM. C (Changepoint) and L (Last-fired) curves correspond to the plain feature representations introduced in Section 3.3. Basic concatenation of C and L produces C+L as another plain representation. Figures also contain two curves for pattern-based approach as C+LP and C+CP, where patterns mined on the L representation (LP) are integrated into C one in the former and patterns mined on the C representation (CP) are integrated into C one in the latter. Comparisons show that basic combination of C and L representations (C+L) does not allow improving over the best of the two in general. Results obtained by using pattern-based models differ depending on informativeness of the representation on which patterns are mined. As explained when introducing sensor encodings, the L representation is informative when applied to object interaction sensors, i.e. the van Kasteren dataset. Indeed C+LP models are usually better than any model lacking patterns, i.e. C, L or C+L, and are almost always better than C+CP models. The improvement is more evident when the L representation is clearly informative and complementary to the C one (e.g. the van Kasteren, House A), and fades away when it tends to be harmful, as in the van Kasteren, House B. When dealing with motion sensors, i.e. the CASAS dataset, the L representation is always misleading, as previously explained. Hence, C+LP models mostly fail to provide improvement over baseline models. However, patterns mined on the C representation (CP) succeed in improving performance when integrated into plain C features, as shown in Figure 4.6.

We also evaluated the relevance of the segmental probabilistic model in exploiting the discriminative power of patterns. To this aim we labeled each time instant within a pattern match with the activity represented by the pattern (allowing for multiple labels for the same instant). The performance in this case is worse than that of all other methods for all values of the pattern selection threshold ϕ (results not shown).

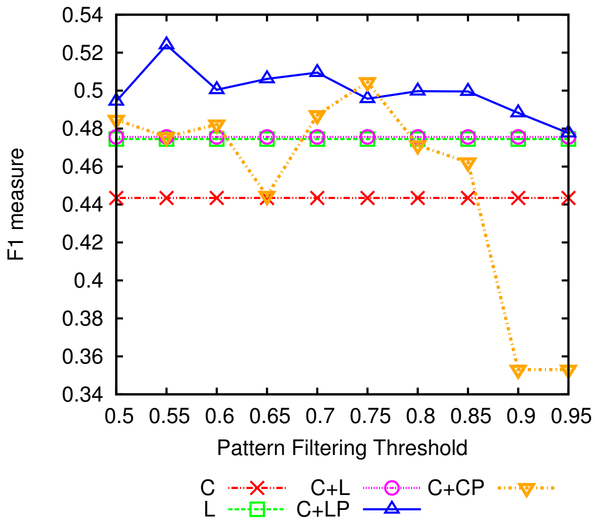
Concerning the number of patterns to be used, there is a trade-off between having enough patterns to model all activities, and focusing on the most discriminative ones. While improvements (apart from the single case in which patterns are useless, i.e. the van Kasteren, House B) over all alternative representations are observed for most values of the threshold on average, the best trade-off differs in the different scenarios. In the van Kasteren House A, for instance, the best results are obtained when only the most discriminative patterns are used. Less discriminative ones suffer from the tendency of the L representation to extend the signal of an activity to the following "idle" period, as previously discussed. On the other hand, in the van Kasteren, House C (see Fig. 4.5(c)) a too high threshold leads to a decrease in performance. This is due to a lack of patterns discriminating similar activities. 'Brushing teeth', 'Shaving',



(a) House A

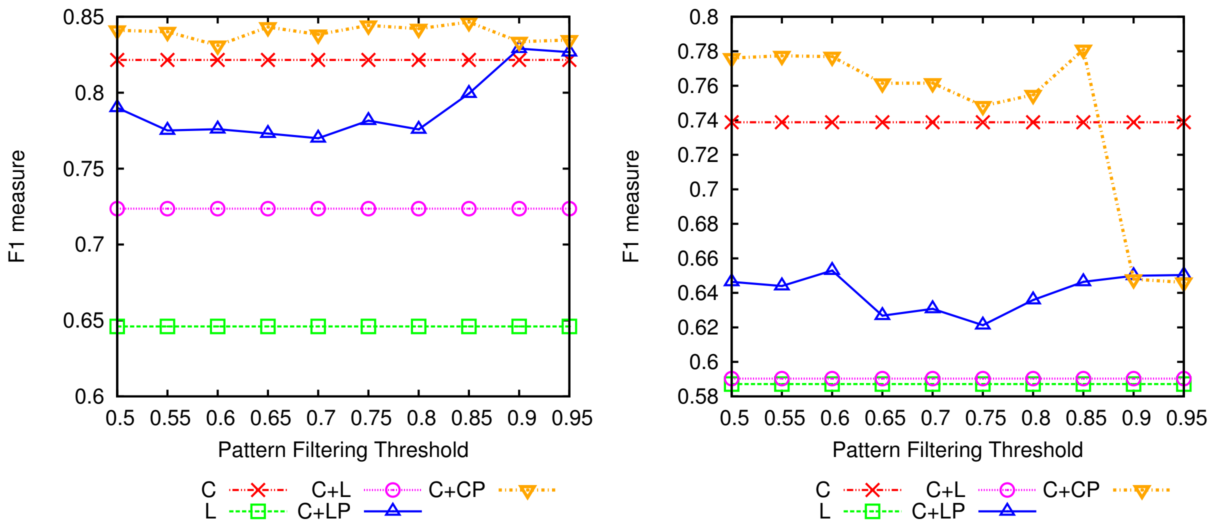


(b) House B



(c) House C

Figure 4.5: PHSMM results for varying pattern thresholds: van Kasteren Dataset



(a) Resident 1 (b) Resident 2

Figure 4.6: PHSMM results for varying pattern thresholds: CASAS Dataset

and 'Taking medication' are held in front of the faucet by interacting with very similar objects. Applying a high threshold yields patterns only for the 'Shaving' among the three and leads to confusion in the prediction. The problem of wrongly missing "idle" periods for lower thresholds is less severe here, as in this case there are patterns specifically characterizing the "idle" period (usually repetitions of the couch pressure sensor). A similar behavior can be observed in the CASAS, Resident 2 (see Fig. 4.6(b)), where focusing on a small set of patterns discards these "idle" characterizing patterns. The large drop in performance observed for thresholds larger than 0.85 is mostly due to an overprediction of the 'Sleeping' activity. This is characterized by patterns with no sensor activations at the borders, movement related sensors in between and quite large gaps. While these patterns correctly discriminate 'Sleeping' activity segments from the others during training, when applied to full days they tend to span time periods consisting of pairs of "Idle" segments with some other motion-related activity in between. The presence of patterns for "idle" as well as other activities helps to disambiguate these cases for lower pattern selection thresholds.

In order to provide a more in depth analysis of the results, we report a set of experiments with a fixed value for the filtering threshold. Determination of the threshold value was automated by performing internal cross validation within the training folds of the LOO approach. The training set of each fold was again split into training and testing sets by applying the same LOO approach. For each internal validation step, a range of possible threshold values was evaluated and the one maximizing the sequential learner F1 measure was recorded as a candidate ϕ. The final ϕ was chosen as the best performing when averaging across internal cross validations, and kept fixed for the outer cross validation procedure. Thresholds for the different datasets are shown in Table 4.1. We present detailed results for the thresholds highlighted in boldface in the table, as LP and CP are known to perform better for the van Kasteren and the CASAS datasets respectively. The other values of the thresholds are used for comparison purposes.

Table 4.1: Thresholds obtained from the internal CV procedure for PHSMM

	van Kasteren			CASAS	
Thresholds	**H. A**	**H. B**	**H. C**	**R. 1**	**R. 2**
C+CP	0.95	0.9	0.5	**0.75**	**0.5**
C+LP	**0.95**	**0.85**	**0.55**	0.85	0.5

Tables 4.2 and 4.3 depict performance measures of baseline models with plain HSMM and pattern-based HSMM (for the selected thresholds) for the van Kasteren and the CASAS datasets respectively, averaged over days and activities. The results provide a more complete picture of the differences between alternatives and conform to those presented in Figures 4.5 and 4.6. The values highlighted in boldface correspond to the best performing models and show that the inner cross validation process is a proper method to determine a correct pattern selection threshold. In the van Kasteren case, C+LP outperforms not only the plain representations but also C+CP. The results of the CASAS dataset are also consistent with our previous observations. Since L representation deteriorates the trajectory information, C+LP fails to provide improvement over the baseline. C+CP, on the other hand, achieves the best performance of all models.

Tables 4.4, 4.5 and 4.6 show the breakdown of the results by activity for the van Kasteren datasets at the (boldface) thresholds determined by the inner cross validation. Results indicate that the contribution of the L representation is rather unstable across activities, preventing an overall improvement for the combined C+L representation. Conversely, the C+LP is much more robust, managing to combine the advantages of the representation and the patterns.

Table 4.2: Results of the experiments averaged across activities for the van Kasteren Dataset

	House A				House B				House C			
Feature	**Prec.**	**Rec.**	**F-1**	**Acc.**	**Prec.**	**Rec.**	**F-1**	**Acc.**	**Prec.**	**Rec.**	**F-1**	**Acc.**
C	67±16	72±15	69±15	90±9	48±16	61±15	53±15	81±16	41±10	49±13	44±10	78±15
L	54±17	68±14	59±15	89±8	40±14	46±20	42±18	48±26	43±10	55±16	47±12	84±14
C+L	66±17	79±12	72±15	94±4	37±14	47±19	41±16	49±26	42±10	56±17	48±12	85±13
C+CP	71±15	73±12	71±13	87±9	44±13	56±14	48±14	72±21	43±17	56±15	49±16	81±16
C+LP	80±14	82±12	**81±12**	96±4	48±12	62±16	**54±13**	83±16	48±16	58±14	**52±15**	86±14

Table 4.3: Results of the experiments averaged across activities for the CASAS Dataset

	Resident 1				**Resident 2**			
Feature	**Precision**	**Recall**	**F-1**	**Accuracy**	**Precision**	**Recall**	**F-1**	**Accuracy**
C	84±13	81±11	82±11	91±8	74±15	74±15	74±15	90±7
L	61±26	73±16	65±24	65±31	55±23	68±13	59±22	66±28
C+L	71±26	78±15	72±24	76±31	55±23	68±13	59±22	70±28
C+CP	84±15	86±13	**84±13**	95±7	75±18	80±15	**78±16**	92±8
C+LP	77±15	84±14	80±13	91±15	62±24	73±15	65±23	76±26

Consider, for instance, the activities 'Taking shower', 'Idle' and 'Leaving house'. These are closely related as they are typically performed in a row by the resident. A simple pattern commonly found in House A consists of a long repetition of the 'front door' sensor. This clearly indicates a 'Leaving house' activity is taking place. It also helps in disambiguating temporally close activities like the just mentioned 'Idle' and 'Taking shower' ones. Conversely, including L representation introduces noisy features: a sensor activated while taking shower will continue to be considered active when the resident is actually idle, leading to a drop in precision for the 'Taking shower' prediction. Concerning House B, overall improvements are limited as patterns fail to improve recognition of the 'Idle' "activity", which is by far the most common one. The main improvement is observed for the 'Going to bed' (aka 'sleeping') activity. This is due to a common pattern made of a sequence of bed pressure mat sensor activations, which mainly disambiguates it with respect to 'Leaving house'. Note that simple C+L representation leads to substantial performance worsening with respect to C representation alone in this setting. Spurious activations of kitchen sensors are wrongly taken as indication of kitchen activities when the resident is actually outside, while the pattern-based model is robust to these noisy observations. Concerning House C, a commonly found pattern consists of a sequence of sensor activations for the fridge, the herbs cupboard and the fridge again, characterizing the 'Preparing breakfast' activity. Let us consider a possible scenario for this situation. A resident starts preparing his breakfast by taking ingredients from the fridge. After a while, flavoring spices taken from the herbs cupboard are added into the blend. As soon as the meal is ready, the resident puts the remaining ingredients back to the fridge. Introducing such patterns allows relating observations which are not sufficient to discriminate among similar activities if taken alone. For instance, a similar scenario involving usage of fridge and other kitchen appliances can be observed for 'Getting snack'. The patterns found for this last activity include activations for the cutlery drawer, the bowl cupboard, and the fridge. Both activities are actually better recognized using the C+LP model.

Activity by activity analysis on the CASAS dataset using the cross validated (boldface)

Table 4.4: Breakdown of the results by activity for van Kasteren Dataset: House A

	C			**C+L**			**C+LP**		
Activity	**Precision**	**Recall**	**F-1**	**Precision**	**Recall**	**F-1**	**Precision**	**Recall**	**F-1**
Idle	86.1	57.5	68.9	96.6	58.4	72.8	94.9	76.0	84.4
Leaving house	92.0	99.5	95.6	98.4	99.7	99.0	97.9	99.9	98.9
Using toilet	74.1	82.5	78.1	72.9	78.9	75.8	76.9	78.4	77.6
Taking shower	97.8	34.7	51.2	35.9	92.4	51.7	94.7	78.9	86.1
Brushing teeth	13.3	34.4	19.1	11.0	31.3	16.3	25.0	43.8	31.8
Going to bed	90.2	89.2	89.7	98.1	99.5	98.8	95.7	99.4	97.5
Preparing breakfast	61.6	70.1	65.6	67.1	56.3	61.3	64.1	75.9	69.5
Preparing dinner	60.8	57.8	59.3	25.7	85.4	39.5	64.2	70.0	67.0
Getting snack	38.3	54.8	45.1	18.2	61.9	28.1	54.8	54.8	54.8
Getting drink	73.2	61.2	66.7	50.9	59.2	54.7	76.9	61.2	68.2

Table 4.5: Breakdown of the results by activity for van Kasteren Dataset: House B

	C			**C+L**			**C+LP**		
Activity	**Precision**	**Recall**	**F-1**	**Precision**	**Recall**	**F-1**	**Precision**	**Recall**	**F-1**
Idle	71.3	57.6	63.7	56.0	56.8	56.4	60.6	61.7	61.1
Leaving house	85.1	91.1	88.0	95.7	32.5	48.6	88.1	91.3	89.7
Using toilet	41.7	71.4	52.6	40.0	72.7	51.6	44.4	67.5	53.6
Taking shower	92.8	92.8	92.8	13.8	91.9	23.9	83.3	90.1	86.6
Brushing teeth	13.3	22.2	16.7	11.9	13.9	12.8	8.4	22.2	12.2
Going to bed	95.4	69.9	80.6	93.4	80.4	86.4	99.1	75.0	85.4
Dressing	20.0	65.2	30.6	13.3	69.6	22.4	19.9	60.9	30.0
Preparing brunch	43.7	61.9	51.2	35.3	35.7	35.5	47.9	53.6	50.6
Preparing dinner	40.0	53.5	45.8	25.2	38.0	30.3	42.7	53.5	47.5
Getting drink	17.7	42.9	25.0	11.4	28.6	16.3	15.4	42.9	22.6
Washing dishes	15.6	47.6	23.5	5.6	9.5	7.0	0	0	0
Eating dinner	0.7	14.3	1.4	0.4	42.9	0.8	0.7	14.3	1.4
Eating brunch	11.5	21.2	14.9	0.9	26.7	1.8	13.3	21.2	16.3

Table 4.6: Breakdown of the results by activity for van Kasteren Dataset: House C

	C			**C+L**			**C+LP**		
Activity	**Precision**	**Recall**	**F-1**	**Precision**	**Recall**	**F-1**	**Precision**	**Recall**	**F-1**
Idle	58.5	41.3	48.4	65.1	69.8	67.4	67.9	75.6	71.5
Leaving house	77.6	94.5	85.2	98.3	86.0	91.7	98.1	86.2	91.8
Eating	38.5	43.6	40.9	40.1	32.2	35.7	35.0	51.9	41.8
Using toilet down	38.5	57.0	45.9	26.1	68.4	37.8	31.4	63.9	42.1
Taking shower	61.6	32.1	42.2	34.5	58.4	43.4	52.3	72.1	60.6
Brushing teeth	16.8	27.7	20.9	21.8	31.7	25.8	16.3	15.8	16.1
Using toilet up	19.0	36.3	24.9	20.1	45.0	27.8	37.6	43.8	40.5
Shaving	0	0	0	13.1	11.6	12.3	27.8	21.7	24.4
Going to bed	99.7	82.6	90.3	99.7	98.2	98.9	99.7	98.3	99.0
Dressing	52.5	65.2	58.2	62.1	68.8	65.3	64.5	71.4	67.8
Taking medication	10.8	26.7	15.4	11.8	26.7	16.3	11.2	66.7	19.2
Preparing breakfast	4.2	2.8	3.4	4.4	5.6	5.0	6.8	25.4	10.8
Preparing lunch	12.9	25.0	17.1	5.1	11.7	7.1	9.9	11.7	10.7
Preparing dinner	64.2	47.6	54.7	23.1	75.2	35.3	28.4	56.6	37.8
Getting snack	18.2	41.7	25.3	15.7	45.8	23.4	31.8	29.2	30.4
Getting drink	0	0	0	0	0	0	0	0	0

thresholds is presented in Table 4.7 and Table 4.8. C+CP model manages to provide improvements over almost all activities. Relatively lower cross-validated thresholds with respect to the van Kasteren dataset enable more patterns to be retained. All activities (but 'Bed to toilet' for Resident 1) possess corresponding patterns for both residents. Found patterns mostly characterize activities in terms of trajectories, sometimes combined with the object usage like burner or faucet. Taking the first resident into consideration (see Table 4.7), one can anticipate that each activity is performed in a specific location, e.g. 'Grooming' in the bathroom, 'Sleeping' in one corner of the bedroom, 'Working at computer' in another corner of the bedroom, yielding distinct patterns. 'Grooming' patterns, for instance, show a trajectory going from the bedroom to the faucet (and possibly the mirror) in the bathroom, where some time is spent (indicated by multiple activations of the motion sensor facing the faucet). The pattern for 'Sleeping' includes activations of two adjacent motion sensors (representing the locations on the bed) combined with the 'no sensor activation' in one corner of the bedroom. 'Working at computer' contains repetitive activations of a single motion sensor (representing the location of the computer) in another corner.

Given the sparsity of C representation (with respect to the L one), patterns sometimes contain 'no sensor activation', especially towards the end of the activity, which can occasionally lead to confusion with the 'Idle' activity. This explains the few observed performance drops, occurring for 'Working at computer' and 'Working at dining room'. On the other hand, a long sequence of 'no sensor activation' characterizes the 'Idle' activity and helps disambiguating it from the 'Bed to toilet' one, characterized by a much shorter duration, which is in turn better predicted without having patterns in itself. The second resident highlights how patterns help disambiguating among similar activities, like the ones performed in the kitchen: 'Preparing breakfast', 'Preparing dinner', and 'Preparing lunch'. 'Preparing breakfast' can be easily distinguished from others thanks to its rich unique patterns modeling trajectories, e.g. from the kitchen towards the upper floor to the room of the resident or from the kitchen to the cellar to the upper floor. It is obvious from the patterns that the resident prefers having his breakfast in his room, which is a distinct property.

4.4.3 Conclusions

We presented a segmental pattern mining approach to improve the activity recognition performance and to incorporate long-range dependencies between distant time instants. For this purpose, we mine sequential patterns characterizing interactions within activity segments and introduce Pattern-based Hidden Semi-Markov Model (PHSMM) in which a probabilistic model to represent the distribution of pattern matches along sequences is learned for maximizing the coverage of an activity segment by a pattern match. This type of representation can complement sensor-based ones by adding information concerning long-range interactions, which

Table 4.7: Breakdown of the results by activity for CASAS Dataset: Resident 1

	C			**C+L**			**C+CP**		
Activity	**Precision**	**Recall**	**F-1**	**Precision**	**Recall**	**F-1**	**Precision**	**Recall**	**F-1**
Idle	87.9	99.9	93.5	96.4	62.7	76.0	95.8	98.1	97.0
Bed to toilet	43.5	17.0	24.4	37.5	67.5	48.2	45.1	34.8	39.2
Preparing breakfast	96.1	75.1	84.3	95.0	75.5	84.1	95.7	87.3	91.3
Grooming	73.4	75.3	74.3	80.0	66.2	72.4	76.8	77.2	77.0
Sleeping	99.6	77.9	87.4	68.0	99.4	80.8	97.0	96.0	96.5
Working at computer	99.3	83.8	90.9	51.1	85.2	63.9	96.5	81.5	88.4
Working at dining room	70.5	86.0	77.5	41.4	87.2	56.1	41.1	65.8	50.6

Table 4.8: Breakdown of the results by activity for CASAS Dataset: Resident 2

	C			**C+L**			**C+CP**		
Activity	**Precision**	**Recall**	**F-1**	**Precision**	**Recall**	**F-1**	**Precision**	**Recall**	**F-1**
Idle	88.4	99.7	93.7	92.9	55.8	69.7	94.1	97.8	95.9
Bed to toilet	47.3	83.6	60.4	50.0	82.1	62.2	66.3	88.1	75.6
Preparing breakfast	44.7	52.8	48.4	42.2	46.8	44.3	56.0	60.2	58.0
Grooming	95.5	62.2	75.3	43.6	87.8	58.2	86.3	94.2	90.1
Preparing dinner	47.3	32.5	38.5	44.4	32.3	37.4	37.8	24.7	29.9
Preparing lunch	26.8	10.1	14.7	25.1	10.2	14.5	19.8	22.3	21.0
Sleeping	99.4	86.1	92.3	64.0	98.6	77.6	96.3	91.6	93.9
Watching TV	84.5	71.7	77.6	62.4	85.1	72.0	85.5	89.8	87.6
Working at computer	99.4	84.3	91.2	39.1	88.4	54.2	99.1	84.7	91.3

would likely be lost otherwise. Our experimental evaluation shows that PHSMM provides better recognition performance than baseline representations in most of the cases and that discovered patterns highlights non-trivial interactions spanning over a significant time horizon. However, the amount of improvement varies greatly from one feature representation to another. The choice of the appropriate representation can be made according to its robustness in correlating sensor activation patterns and activities being performed. Sensors signaling object interactions, like in the van Kasteren dataset, are suitable to the L representation, since the latest sensor activation tends to indicate the activity that is to be performed, e.g. bedroom door for sleeping, front door for leaving. The C representation, on the other hand, is useful in identifying activities represented by trajectories, like in the CASAS dataset, where each motion sensor activation constitutes a part of a trajectory. We indeed achieved the best performance by using C+LP and C+CP for the houses with contact switch sensors and with motion sensors respectively.

In our experiments, we observed that no single pattern selection threshold provided the best performance in all scenarios. A common problem causing a drop in performance for certain threshold choices stems from mishandling of idle class, e.g. lack of idle patterns for higher thresholds (resident 2, CASAS) and co-occurrence of distinct activity patterns suppressing the effect of those representing the idle one (house A, van Kasteren). A possible solution consists of treating the idle class as a separate case, for which e.g. patterns with a lower threshold could be retained.

The method is applicable for scenarios rich in features enabling distinctive activity patterns to be discovered. In the case of very sparse feature representations, matching patterns with wide range of possible gap lengths might be expensive. Cautious consideration could be required to constrain the set of valid patterns.

Our approach is currently limited to non-concurrent activities. However, many real world scenarios involve overlapping and concurrent activities, especially when modeling multiple interacting agents. The underlying idea can be generalized to such scenarios by combining mined patterns with more expressive sequential models like factorial HSMMs. The case of multiple agents is especially challenging as it requires to disambiguate those sensors which do not provide information on the agent involved (e.g. infrared sensors). Finally, our approach assumes a batch setting, in which a whole (temporal) sequence has to be jointly labeled after being fully observed. Online activity recognition, where the aim is predicting the currently performed activity, requires to adapt the segmental pattern mining algorithm to search for incremental patterns modeling increasingly long portions of an activity segment and deal with the increased complexity of the mining, matching, modeling and inference steps.

Chapter 5

Activity Discovery

5.1 Unsupervised Activity Discovery

Activity recognition has long been studied by the machine learning community. Most of the work in the field has focused on supervised approaches in order to train activity models. We have also presented a supervised technique in the previous chapter. However, these techniques require the availability of labeled sequences, an expensive and time consuming process. Furthermore, training data is specific to the setting involving the activities to be recognized and the persons involved, as the daily living habits change from an individual to another. Activity discovery aims at identifying activities within data streams in the absence of data annotation. Therefore, it can be used in any possible daily-life scenario. In health monitoring applications, for instance, one of the tasks is continuously checking the behavior of a patient in order to determine whether his/her routines are maintained, regardless of the type of activities being performed. Inconsistencies in daily routines, i.e. changes in the structure of performed activities, can suggest problems in patient's health. As many unsupervised learning tasks, activity discovery is a challenging problem: many activities tend to share a similar set of signals (e.g. kitchen sensors for food-related activities), short periods lacking any signal at all can occur during an activity, to be distinguished from truly "idle" periods where no activity is being performed. Finally the discovery needs to be robust enough to account for variations in the way activities can be performed.

In this section, we present an activity discovery approach that addresses the above-mentioned challenges. The rationale behind the approach is that distinct activities should correspond to separate sets of sensors, e.g. activations of pairs or triplets of sensors, possibly repeated over time, jointly indicating a certain activity. With this reasoning, we assume that transition from one set to another indicates the possible time of an activity change. Following this idea, we propose two segmentation algorithms looking for the change points in a sensor stream. Extracted segments are then clustered in order to find groups of similar segments each representing a candidate activity. Finally we use segments labeled with cluster identifiers to train the parameters of a sequential labeling algorithm which is then used for smoothing the initial segmentation.

5.1.1 Activity Discovery Framework

Our activity discovery process consists of three steps:

1. **Sequence segmentation**: in this step, the full sensor stream is partitioned into segments which represent candidate activities, i.e. each segment should approximately span the whole time horizon in which an activity is continuously conducted. The segmentation procedure scans the stream searching for *changepoints* suggesting a change in the activity being performed.

2. **Segment clustering**: once segments have been identified, a clustering algorithm is employed in order to group together similar segments, each group representing a distinct candidate activity. Designing an appropriate segment representation is crucial here in order to boost performance.

3. **Sequential labeling**: the final step employs the segment clusters produced by the previous step to train the parameters of a sequential labeling algorithm. The learned model is then used to run inference on the full sensor stream obtaining the final segmentation output. The rationale of this component is that the learned probabilistic model should allow smoothing segment borders with respect to the segmentation and clustering output, possibly improving recognition accuracy.

In the following we detail each step of the process.

Sequence Segmentation

The aim of the segmentation phase is to partition the sensor stream into fragments so that each fragment characterizes the occurrence of an activity. As we have already seen in the previous chapters, activity datasets used for the evaluation differ from each other by the number and the type of the sensors. The fact that the quality of the segmentation is highly dependent on the dataset properties calls for a specific handling in algorithm design. For this purpose, we propose two novel approaches, distance-based and context-based segmentation, for each experimental setting.

Distance-based segmentation is based on the idea that an activity is related to the sensor events occurring within a specific range. More specifically, the consecutive activation of two sensors whose distance to each other is less than a threshold (ϕ) is likely to indicate the

Algorithm 6 Distance-based Segmentation

Input:
- D: sequence of observations
- ϕ: separation threshold
- C: sensor coordinates

Output:
- B: candidate activity borders

procedure DISTANCESEGMENTATION(D,ϕ,C)
  Initialize border candidates (B) to the empty set
  Calculate Manhattan distance matrix (M) of sensor pairings with (C)
  Find active time instants $T = (t_1, t_2, \ldots, t_m)$ in D
  for all $i \in [1, m-1]$ **do**
    Initialize pairwise distances (P) to the empty set
    for all sensor pairing $(j, k) \in (D(t_i), D(t_{i+1}))$ **do**
      $P \leftarrow P \cup M(j, k)$
    if $count(P > \phi)/|P| > 0.5$ **then**
      $B \leftarrow B \cup t_i$
      if $t_{i+1} - 1 \notin T$ **then**
        $B \leftarrow B \cup t_{i+1} - 1$
  $B \leftarrow B \cup length(D)$
  return B
end procedure

persistence of the same activity. For example, preparing dinner is typically characterized by activation of kitchen sensors. Any sensor event occurring in the bedroom, however, is probably unrelated to the dinner activity. Selection of the threshold can be done in a number of ways depending on the dataset. One can assume that every activity is bounded with a certain room in the apartment. In this case, the threshold is computed as the distance between the two closest rooms. We used this type of distance in the van Kasteren dataset (see Section 3.1) as activities are known to be performed in separate rooms.

Algorithm 6 shows the pseudocode of our distance-based segmentation technique. The algorithm takes as inputs a sequence of observations (D) as sensor activations and the spatial coordinates of all sensors (C), plus a threshold ϕ controlling when to introduce a breakpoint in the sequence. It first computes a matrix M of pairwise distances between sensors using the Manhattan metric, as it provides a natural measure of walking path length. The algorithm then identifies all time instants having at least one sensor activation and iteratively processes each of them. In order to decide whether to introduce a breakpoint at active time instant t_i, the algorithm compares its active sensors with those of the next active time instant t_{i+1}, using the previously computed distance matrix. If more than half of the comparisons have a distance greater than the threshold ϕ, i.e. sensors from the two time instants tend to be far apart, a breakpoint is added at time instant t_i. Note that t_{i+1} is not necessarily the time instant immediately following t_i, as they can be separated by a sequence of time instants lacking any sensor activation, likely indicating an idle "activity". In this case the algorithm introduces an additional breakpoint at time instant $t_{i+1} - 1$, isolating the segment with no activations. Note that conversely, null segments separating two active time instants with spatially close active sensors are merged in the segment containing t_i and t_{i+1}. This can be reasonable as activities often include short periods with no activations, but can miss longer null segments potentially representing idle cases. At the end of this section we introduce a post-processing procedure addressing this problem. Our algorithm resembles the one in (Hong and Nugent, 2013), and indeed the two produce very similar segmentations, but the cited one requires much more information concerning the location where specific activities are performed and ad-hoc rules extracted via profound investigation of the sensor stream.

The distance-based segmentation approach is suitable for datasets in which the location information is closely related to the activity being performed. In many cases, on the other hand, information regarding the locations of the sensors is not available, which makes the proposed method inapplicable. This is especially relevant when sensors are attached on mobile objects the location of which are not fixed. Assume that sensors are deployed on a vacuum cleaner and on cleaning products. Places of these object may change during the cleaning activity. If they are far apart in any part of the cleaning process, the activity cannot be detected. Therefore a general approach that does not depend on any kind of knowledge is required. We thus propose the context-based segmentation in which change points in sensor activation patterns are extracted. The rationale behind the approach is that two activities should be related to two distinct patterns of sensor activations, e.g. pairs or triplets of sensors jointly activated in the same time instant. This is implemented by extracting features for each time instant. In order to evaluate the effect of different representations, we introduce three distinct features. Each time instant is represented by the set of its active sensors (1), by the set of its n-grams (2), and by the set of its up to n-grams (3), where an n-gram is a set of n sensors jointly active in the time instant. The similarity between two time instants with sets A and B is then computed using the Jaccard index:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{5.1}$$

We extend this similarity to include the *context* of a time instant by defining a *frame* as a sequence of time instants of a certain length (τ) and considering similarity between frames. The similarity between time instants t_i and t_{i+1} is then computed considering the frames $[t_{i-\tau+1} : t_i]$ and $[t_{i+1} : t_{i+1+\tau}]$, representing each frame by the union of the sets of (up to) n-grams of its time instants, and computing the Jaccard index between the two frames. Algorithm 7 outlines the context-based segmentation approach, where frames exceeding the

Algorithm 7 Context-based Segmentation

Input:
D: sequence of observations
τ: frame size
n: gram size
Output:
B: candidate activity borders

procedure $\textsc{ContextSegmentation}(D,\tau,n)$
  Initialize border candidates (B) to the empty set
  $L \leftarrow length(D)$
  for all $i \in [1, L-1]$ **do**
    if Jaccard$(n, D[t_{i-\tau+1} : t_i], D[t_{i+1} : t_{i+\tau}]) = 0$ **then**
      $B \leftarrow B \cup i$
  $B \leftarrow B \cup L$
  return B
end procedure

borders of the sequence are appropriately trimmed.

The border generation process may result with a number of segments larger than the true one. In order to fix the most obvious cases, we applied a pruning procedure based on a number of simple reasonings. (1) There are a few occasions in which the distance-based algorithm extracts segments of size one when two consecutive time instants have exactly the same active sensors, and these come from two different locations. For instance, toilet activity is interleaved with sleeping, and is characterized by sensor activations from both the bedroom (bedroom door) and the toilet (e.g. toilet flush). If this occurs in a row for a small number of time instants, we merge them together in a single segment. (2) Let a segment without any sensor activation (zero segment) be preceded and followed by two segments whose active sensors either occur in the same location or are similar. These three consecutive segments should be merged into one to represent a single activity or be kept separate as two distinct segments of the same activity separated by an idle segment, depending on the length of the zero segment in the middle. We choose the former option if the resulting segment is smaller than a threshold representing the typical duration of activities, and the latter otherwise. The threshold is computed as the average length of the segments obtained by the segmentation algorithms, after excluding very long segments which likely represent peculiar activities like sleeping or idle. As an example of the merge operation, consider a dinner activity as taking food from the fridge followed by heating food in the microwave and then eating. The time we waited for heating is a zero segment, yet belongs to the same dinner activity. A three minute toilet activity performed two hours after another occurrence of toilet activity, on the other hand, should not be merged with the previous one since it is clear that they are distinct activity occurrences separated by another activity (e.g. sleeping).

Segment Clustering

The purpose of the clustering step is to determine the intrinsic grouping of the segments extracted in the previous phase so that each group represents an activity. We represent segments in terms of histograms of time instant-based features collected over each segment. Time instant-based features are extracted in the same manner as we did in the context-based segmentation, i.e. active sensors (1), n-grams of sensors (2), and up to n-grams of sensors (3) are extracted for each time instant in the input sequence. Here, we introduce another representation in addition to the three existing ones. 2D-n-grams is a sliding window of size n, running in two dimension. In the first dimension, n-grams of sensors for each time instant in the window are extracted. In the second dimension, one of the previously extracted n-grams is selected from each time instant within the window and joint occurrence of the selected n-grams, e.g. pairs (when n=2) of n-grams or triplets (when n=3) of n-grams, creates a feature. Feature creation within the window continues until all possible joint occurrences of n-grams are found. We aimed at including temporal relations of sensors with this kind of representation. Found features are then collected over the segments and used for creating histograms as the counts of each feature. An illustrative example presented in Figure 5.1 includes histograms for a segment spanning two time instants (where $n = 2$).

Segment representations are then fed to a clustering algorithm. This has to deal with high-dimensional data, as coming from the up to n-gram feature representation, and automatically identify the number of clusters, which is not known in advance. We rely on the HDDC method (Bergé et al., 2012) which satisfies both requirements. It is based on a modified Gaussian Mixture Model with a dimensionality reduction technique which determines the specific subspace in which each class is located by using eigenvectors of the covariance matrix. Models representing the subspaces are used to choose the number of clusters. To this end, clustering results are computed for different number of clusters and different models and the one maximizing Bayesian Information Criterion is selected. Further details can be found in the original paper (Bouveyron et al., 2007). We chose HDDC method because of its simplicity but any other clustering technique can be used as long as our requirements are satisfied.

Sequential Labeling

In principle, our algorithm could end up with the groupings returned by the clustering algorithm, each group representing a candidate activity. However, both segmentation and clustering steps are prone to errors and only provide approximations of actual segments and true groups. We use these approximations to train a sequential labeling algorithm, which assigns a label to each time instant in the sequence. The learned model is then used to run inference on the full sensor stream, providing the final sequential labeling. Each cluster in our setting corresponds

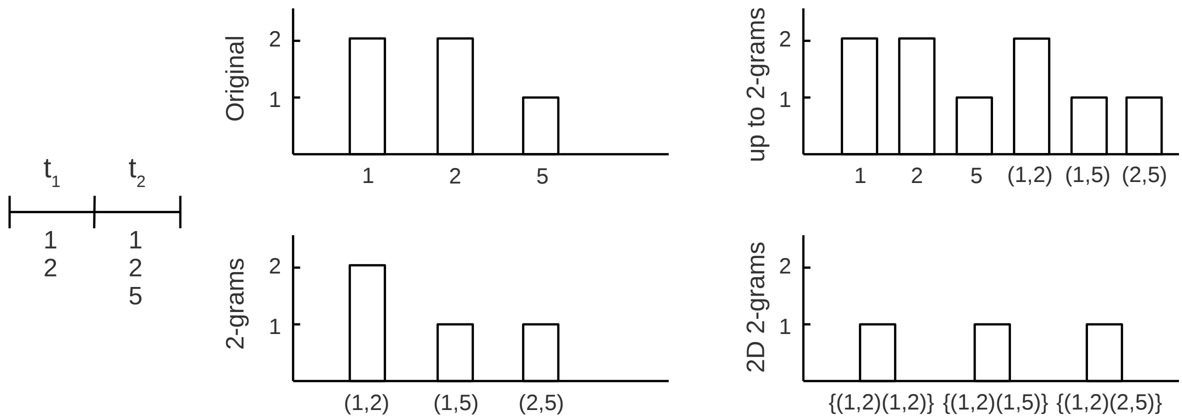


Figure 5.1: Histogram representations of a segment for active sensors (top-left), 2-grams (bottom-left), up to 2-grams (top-right), 2D 2-grams (bottom-right).

to a different label in the sequential model.

We employ Hidden Semi-Markov Model (HSMM) as sequential labeling approach, which is appropriate to label sequences where consecutive time instants tend to share the same label. HSMM models duration distributions explicitly for different states, making it especially useful for segmenting sequences into fragments, each characterized by the same label. We follow the same formalization and the duration distribution as described in Section 4.1.

Parameters of all probabilities can be readily estimated from counts over the cluster segments given that each label is associated with a cluster from the previous step. The transition probability between labels y_u and y_v, for instance, can be computed as the fraction of times in which a segment from cluster y_v follows one from cluster y_u in the original sequence, with respect to the overall number of segments in cluster y_u.

Once parameters are learned from the clustering results, inference is run on the whole sequence providing the labeled segmentation with maximal probability, which is computed by the Viterbi algorithm as previously discussed.

5.1.2 Experiments

The performance of the proposed framework was evaluated again on the van Kasteren and the CASAS datasets. However, the van Kasteren dataset used in this part has some differences from the one that was presented in Section 3.1. We made this change to compare our method properly with a recently proposed unsupervised activity recognition approach, the evaluation of which was performed on the selected dataset. Although the floor plans of the houses and the sensor structure remained the same, the dataset was collected for 8 activities (see Table 5.1) over 28 days only in the house A (van Kasteren et al., 2008a).

In our experiments, we used Changepoint (C) feature representation as it was shown to provide overall the best HSMM results in comparison to plain representations, i.e. Raw and Last-fired. In addition, it has been observed that C is robust against noise and is capable of tolerating the dataset specific sensor failures (see Section 4.2). We followed the notation in representing activity data as presented in Section 3.2. The performance of the system was evaluated by using the class accuracy metric proposed in (van Kasteren et al., 2008a) (see Section 3.4).

Tables 5.1, 5.2, and 5.3 show confusion matrices computed from the class accuracies for the van Kasteren and the CASAS dataset respectively. As discussed in the first part of Section 5.1.1, we applied the distance-based segmentation to the former and the context-based one to the latter. In all experiments the maximum gram size was set to two as higher values provided similar results while increasing computational complexity. As a time instant-based feature extraction method, *up to n-gram* was performed both in context-based segmentation and in segment clustering. In our experiments, original (active sensors) representation was incapable of discriminating segments of similar activities as it fails to model the relationships between sensors, e.g. those activating together. Using n-gram representation appeared to be problematic especially in the segmentation step. When different combinations of similar sensor events occur in two consecutive frames, similarity relation between common sensor events may be lost. Let two consecutive frames of size two be $s^1_{t_1,t_2}$ and $s^2_{t_3,t_4}$, with active sensors in time instants $s^1_{t_1} = \{1,2\}$, $s^1_{t_2} = \{2,3\}$, $s^2_{t_3} = \{1,3\}$, and $s^2_{t_4} = \{2\}$. 2-gram representation will thus be $\{(1,2),(2,3)\}$ and $\{(1,3),(2)\}$ for the former and the latter respectively and their Jaccard similarity will be zero even if they share the same sensor events. 2D-n-gram representation suffers from the lack of generalizability since it yields very segment specific histograms by considering orders of the sensor activations through the time instants. However, the way how each activity is performed changes from one instance to another. The selected one, *up to n-gram*, on the other hand was observed to be robust to such scenarios (results of the alternative segment representations were omitted).

Clustering does not assign names to the detected groups. However, it is clear that if a cluster contains mostly segments corresponding to a certain activity, it can be considered as an approximation of that activity. In order to identify the most likely activity for each cluster, we try all possible distinct activity assignments to clusters and choose the one maximizing class accuracy. If the algorithm identifies more clusters than the true number of activities, the best assignment will assign the clusters in excess to a dummy wrong activity. We do not explicitly report dummy clusters in the Tables, but include their assignments when computing the percentage of correct predictions (i.e. predicted rows do not always sum to one). Note that this best assignment measure is a fair evaluation procedure, as we simply identify for each cluster which is the activity it is most likely representing, forcing each cluster to represent a

Table 5.1: Detailed activity discovery results of van Kastaren Dataset (values as percentages)

		Idle	Leaving	Toileting	Showering	Sleeping	Breakfast	Dinner	Drink
Idle	EON	72	4	13	3	5	0	3	0
	CLU	100	0	0	0	0	0	0	0
	SEG+CLU+HSMM	74(74)	12(1)	3(4)	0(2)	8(3)	2(2)	1(2)	0(0)
Leaving	EON	0	74	11	0	14	1	0	0
	CLU	0	80	20	0	0	0	0	0
	SEG+CLU+HSMM	4(41)	96(59)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
Toileting	EON	0	56	27	5	1	11	0	0
	CLU	9	0	76	6	9	0	0	0
	SEG+CLU+HSMM	4(13)	1(1)	78(32)	5(4)	4(4)	0(1)	0(0)	0(10)
Showering	EON	0	0	0	100	0	0	0	0
	CLU	0	0	0	100	0	0	0	0
	SEG+CLU+HSMM	9(15)	8(0)	1(0)	76(85)	0(0)	0(0)	0(0)	0(0)
Sleeping	EON	0	0	0	0	100	0	0	0
	CLU	62	0	3	7	28	0	0	0
	SEG+CLU+HSMM	35(27)	21(0)	0(1)	0(0)	44(44)	0(0)	0(0)	0(24)
Breakfast	EON	0	21	14	0	4	44	14	3
	CLU	0	0	0	0	0	100	0	0
	SEG+CLU+HSMM	18(31)	0(0)	5(0)	0(0)	9(1)	68(68)	0(0)	0(0)
Dinner	EON	0	0	59	0	13	0	28	0
	CLU	7	0	0	0	0	57	36	0
	SEG+CLU+HSMM	23(12)	2(0)	0(0)	0(0)	0(0)	36(35)	38(53)	0(0)
Drink	EON	37	0	0	0	0	0	0	63
	CLU	14	0	0	0	0	0	0	86
	SEG+CLU+HSMM	9(16)	1(0)	1(5)	0(0)	0(0)	80(64)	8(15)	0(0)

distinct activity.

The first set of experiments aims at comparing our approach with the evidential ontology network (EON) model proposed in (Hong and Nugent, 2013) and evaluated on the van Kasteren dataset. To the best of our knowledge, EON is the latest study that addresses the problem of automatically discovering activities from a sensor stream. The method includes a segmentation phase similar to our approach while activity identification is performed by taking into account conceptual knowledge of relations between object interactions and activities. This allows us to compare not only our segmentation algorithms but also our fully unsupervised technique against a knowledge-based unsupervised method. Table 5.1 shows confusion matrices for the different activities, where rows indicate true activities and columns predicted ones. `EON` rows report results for the EON model, while `CLU` shows results of our clustering step

Table 5.2: Detailed activity discovery results of CASAS Dataset: Resident 1 (values as percentages)

		Idle	Bed to Toilet	Breakfast	Grooming	Sleeping	Working at computer	Working at dining room
Idle	CLU	100	0	0	0	0	0	0
	SEG+CLU+HSMM	100(83)	0(7)	0(6)	0(0)	0(0)	0(0)	0(0)
Bed to Toilet	CLU	68	0	0	31	1	0	0
	SEG+CLU+HSMM	28(9)	0(38)	0(1)	70(33)	1(9)	1(3)	0(0)
Breakfast	CLU	7	0	93	0	0	0	0
	SEG+CLU+HSMM	6(29)	0(3)	92(35)	2(7)	0(0)	0(0)	0(0)
Grooming	CLU	12	0	0	88	0	0	0
	SEG+CLU+HSMM	9(6)	0(30)	0(5)	91(41)	0(8)	0(1)	0(0)
Sleeping	CLU	2	0	0	40	58	0	0
	SEG+CLU+HSMM	6(3)	0(3)	0(2)	0(1)	94(51)	0(4)	0(0)
Working at computer	CLU	3	13	0	0	0	82	2
	SEG+CLU+HSMM	12(10)	0(1)	0(6)	0(1)	0(2)	86(36)	2(3)
Working at dining room	CLU	0	0	0	0	0	38	62
	SEG+CLU+HSMM	7(34)	0(0)	0(0)	0(2)	0(0)	8(8)	45(23)

applied to the true segmentation, the same setting used in (Hong and Nugent, 2011). Our approach outperforms the competitor[1] in six out of eight activities and is on par on one. The only case where we get worse results is on Sleeping, which is characterized by a single sensor activation of bedroom door and a long period of no sensor activation. Sleeping is divided into many parts as it is interleaved by the Toilet activity. Segments separating two consecutive Toilet activities do not have any sensor activation, and are thus wrongly clustered in the Idle group[2].

`SEG+CLU+HSMM` rows report results of our complete approach, while results in brackets show the performance of the segmentation and clustering steps only (`SEG+CLU`). Clustering generates nine clusters for both true segmentation and predicted one. Incorporating sequential labeling (`SEG+CLU+HSMM` rows) provides overall better results, by improving recognition of Leaving and Toileting (while performance for Showering and Dinner are slightly degraded). For Leaving, the improvement is achieved by recovering from incorrect segmentations introducing spurious segments predicted as Idle within a Leaving activity. For Toilet, the clustering algorithm actually spreads segments containing Toilet activities in two different clusters, together with some segments from other activities, while the HSMM manages to identify most of them as

[1] Note that by substantially extending the knowledge concerning activities being searched it is possible to achieve much higher recognition accuracy (Hong and Nugent, 2011). However, our aim here is to perform activity detection without any specific knowledge on the activities being performed.

[2] (Hong and Nugent, 2013) Hong et al., Segmenting sensor data for activity monitoring in smart environments

Table 5.3: Detailed activity discovery results of CASAS Dataset: Resident 2 (values as percentages)

		Idle	Bed to Toilet	Breakfast	Grooming	Dinner	Lunch	Sleeping	Watching TV	Working at Computer
Idle	CLU	79	0	0	0	0	0	0	21	0
	SEG+CLU+HSMM	100(54)	0(19)	0(3)	0(2)	0(2)	0(5)	0(7)	0(3)	0(3)
Bed to Toilet	CLU	10	0	0	90	0	0	0	0	0
	SEG+CLU+HSMM	1(4)	0(31)	0(8)	99(33)	0(0)	0(8)	0(5)	0(0)	0(7)
Breakfast	CLU	0	0	87	0	6	7	0	0	0
	SEG+CLU+HSMM	9(13)	0(2)	21(53)	1(3)	65(13)	1(14)	0(0)	3(0)	0(2)
Grooming	CLU	0	0	2	98	0	0	0	0	0
	SEG+CLU+HSMM	21(9)	0(2)	2(10)	77(64)	0(0)	0(3)	0(0)	0(1)	0(11)
Dinner	CLU	15	0	39	0	33	13	0	0	0
	SEG+CLU+HSMM	28(14)	0(0)	1(16)	0(0)	63(39)	4(26)	0(0)	4(0)	0(0)
Lunch	CLU	23	0	50	0	16	11	0	0	0
	SEG+CLU+HSMM	23(7)	0(0)	4(37)	0(0)	48(17)	16(25)	0(0)	9(10)	0(0)
Sleeping	CLU	7	0	0	0	23	0	70	0	0
	SEG+CLU+HSMM	3(6)	4(32)	0(0)	0(1)	0(0)	0(0)	93(46)	0(0)	0(0)
Watching TV	CLU	14	29	1	0	0	24	0	32	0
	SEG+CLU+HSMM	16(32)	0(2)	1(5)	0(0)	7(2)	0(3)	0(0)	51(29)	0(0)
Working at computer	CLU	13	0	0	0	0	0	0	0	87
	SEG+CLU+HSMM	12(8)	0(3)	0(3)	0(4)	0(0)	0(3)	0(7)	0(0)	88(51)

belonging to the same class. A current limitation of the approach is that activities generated by similar sensor activations tend to be merged into the same cluster. Activities performed in the kitchen, for instance, are clustered together in predicted segmentation as they share basically the same sensor activations, explaining the overprediction of Breakfast. Drink, however, is assigned to a distinct cluster in true segmentation which decreased the confusion in prediction of kitchen-oriented activities.

Tables 5.2 and 5.3 report results of `CLU`, `SEG+CLU+HSMM` and (in brackets) `SEG+CLU` for resident 1 and resident 2 of the CASAS dataset in order. Clustering algorithm generates 10 and 11 clusters respectively. The complete model provides significant improvements over `SEG+CLU` in almost all cases. This is mostly due to recovering portions of segments which were assigned to spurious clusters (clustering detects more clusters than the true number of activities), thanks to the smoothing effect of HSMM and its capacity of correctly modeling duration of activities. This allows the complete model to even slightly improve over the clustering applied to the true segmentation, as shown by comparing rows `CLU` and `SEG+CLU+HSMM`. CASAS dataset also suffers from the problem of clustering similar activities together. Bed to Toilet and Grooming similarity holds for both residents and neither `CLU` nor `SEG+CLU+HSMM` is able to identify these activities since they are very similar in terms of sensor activations involved. In both activities, residents move from the bedroom to the toilet. After spending some "idle" time in the toilet, they go back to the bedroom. If residents spend more time in front of the mirror (indicating personal care, hence grooming), segment representation becomes richer and the activity is predicted as Grooming. In other cases, "idle" time spent in the toilet dominates the activations in the trajectory, causing the activity to be predicted as Idle. Kitchen-based activities performed by the resident 2 were clustered as Breakfast and as Dinner for the true segmentation and the predicted segmentation respectively. Therefore segments of these activities tend to be predicted in favor of the class that was assigned as the representative of kitchen-based activities.

5.1.3 Conclusion

The effectiveness and suitability of our approach was evaluated in two smart home datasets. Initial results show that proposed approach succeeds in discovering activities in many situations. Although our technique does not depend on any assumptions on dataset, e.g. type of activities, number of clusters, it outperformed a method using activity definitions as domain knowledge. We observed that our segmentation algorithm produces segments which are quite close to the true ones. The final sequential labeling model succeeds in further refining the results, by smoothing segment borders and recovering part of the segments assigned to spurious clusters.

The proposed framework, however, suffers from a number of limitations. Similar activities tend to be clustered together and are hard to distinguish. In order to prevent this, a better way to represent segments or additional features (e.g. time of the day, duration of the activity etc.) can be defined. Interleaved activities also decrease performance, as when repeatedly going to toilet during the night. Relationships between neighbouring segments could be included in the clustering phase in order to address this problem.

5.2 Pattern-based Unsupervised Activity Discovery

In this section, we present our ongoing work that is built on the top of the activity discovery framework proposed in the previous section. It has been seen in Chapter 4 that patterns provide important information for distinguishing activities and that their usage in predictive models significantly improves the recognition performance. With this motivation, behavioral patterns are integrated into the activity discovery framework.

We follow the same reasoning as in the previous section for discovering activities, i.e. the three-step segmentation-clustering-labeling approach. The main aim here is to promote improvements both in clustering quality and in prediction performance of the sequential labeling approach by taking advantage of the discriminative power of the patterns. Therefore, segmentation step is used without any change, i.e. an input stream is partitioned into fragments by using Distance-based or Context-based segmentation algorithm depending on the dataset properties.

The first difference with respect to the base discovery framework is in the clustering phase the inputs of which are segments represented as histograms of time instant-base features (see *Segment Clustering* in Section 5.1.1). The success of this step depends directly on the informativeness of the segment representation. The ability of the clustering method to discriminate activity groups can be improved by encoding more information about activity characteristics into the representation. For this purpose, we enrich segment representations with pattern-based features. Patterns are extracted by using PrefixSpan (Pei et al., 2004) instead of pboost (see Section 4.3) due to the absence of the labeling information (positive and negative examples cannot be determined). Initially, PrefixSpan takes as input candidate segments generated in the segmentation step, each represented as a sequence of set of active sensor identifiers (or 0 as an inactivity identifier) over time instants of the segment, and extracts frequent sequential patterns. Afterwards, for each segment, we compute how many times each pattern is matched with the segment. Finally, the segment representations, e.g. up to n-grams, are concatenated with the corresponding number of pattern matches to create pattern-based representations. Note that, number of pattern features is equal to the number of patterns mined. The resulting enriched representations are then fed to the HDDC clustering algorithm (see *Segment Clustering* in Section 5.1.1).

We introduce the second difference in the sequential labeling step. HSMM used in the base version is replaced with the Pattern-based HSMM introduced in Section 4.4 since it has been proven to perform better than conventional methods (see Section 4.4.2). Similar to the previous step, patterns extracted by PrefixSpan are employed in PHSMM approach. We omit explanation of the procedure because a comprehensive and detailed analysis has been provided in Sections 4.4 and 5.1.1 (*Sequential Labeling*).

5.2.1 Preliminary Experiments

The proposed approach was evaluated on the van Kasteren dataset with the setting presented in Section 5.1.2. We stick to the previous experimental conditions to ensure consistency in our experiments. Changepoint (C) feature was selected as the representation method and the activity data was created as defined in Section 3.2. The performance of the framework was measured by using the class accuracy metric.

For the Pattern-based activity discovery framework, we also made similar choices to those made in the base three-step technique. Candidate segments were generated by employing the Distance-based segmentation algorithm. As a time instant-based feature extraction method, *up to n-gram* was performed in segment clustering, where the gram size (n) was set to two. We extracted top-k frequent patterns from Last-fired feature by using PrefixSpan to enrich the segment representations for the clustering step. The same patterns were also used in the sequential labeling phase, i.e. PHSMM (parameters, e.g. duration distribution, coverage, were kept the same as in Section 4.4). In our experiments, we set number of patterns (k) to ten.

We first evaluated the effect of pattern usage on the clustering performance for the true segmentation. Table 5.4 shows confusion matrices for the different activities, where rows indicate true activities and columns predicted ones. `CLU` rows report clustering results presented in Table 5.1, while `PCLU` shows results of the clustering step after enriching segment representations with the pattern-based features. When PrefixSpan was run on the true activity segments, patterns for Sleeping and Toileting were produced. Hence, during enrichment process, segment representations for these activities benefit from the pattern-based features. Indeed, `PCLU` allows improving the clustering performance in these two cases.

Table 5.5 presents results of the complete models. `SEG+CLU+HSMM` rows are the same as those given in Table 5.1. In `SEG+CLU+PHSMM`, we used the same candidate segments and cluster labels with the base complete model, and run PHSMM instead of HSMM. Here, patterns were mined after the clustering step and used only in the sequential labeling phase. `SEG+PCLU+PHSMM` rows show results of our Pattern-based activity discovery framework. Using patterns only in the sequential labeling (`SEG+CLU+PHSMM`) outperforms the base complete model in five out of eight classes, is on par on one, and falls behind in two cases, i.e. Toileting and Idle. Performance improvement proves that PHSMM takes advantage of the discriminative power of the patterns, resulting increase in overall prediction. Pattern-based model

Table 5.4: Detailed clustering results of van Kastaren Dataset (values as percentages)

		Idle	Leaving	Toileting	Showering	Sleeping	Breakfast	Dinner	Drink
Idle	CLU	100	0	0	0	0	0	0	0
	PCLU	100	0	0	0	0	0	0	0
Leaving	CLU	0	80	20	0	0	0	0	0
	PCLU	1	77	1	0	21	0	0	0
Toileting	CLU	9	0	76	6	9	0	0	0
	PCLU	7	0	83	4	6	0	0	0
Showering	CLU	0	0	0	100	0	0	0	0
	PCLU	0	0	0	100	0	0	0	0
Sleeping	CLU	62	0	3	7	28	0	0	0
	PCLU	27	0	2	1	70	0	0	0
Breakfast	CLU	0	0	0	0	0	100	0	0
	PCLU	0	0	0	0	0	100	0	0
Dinner	CLU	7	0	0	0	0	57	36	0
	PCLU	7	0	0	0	0	57	36	0
Drink	CLU	14	0	0	0	0	0	0	86
	PCLU	14	0	0	0	0	0	0	86

(`SEG+PCLU+PHSMM`) provides even better results in a way that it is able to distinguish kitchen-based activities that have been misclassified in the former two cases. In addition, results of Toileting and Sleeping are consistent with those acquired in Table 5.4, i.e. enriching segment representation with pattern-based features allows improving clustering performance. The decrease in the prediction of Idle, however, is due to lack of patterns for Idle class and the usage of Last-fired representation in mining patterns. As discussed in Section 4.2, extracted patterns inherit the deficiency of the representation, i.e. activation of the last sensor is propagated over the idle time instants. Therefore, Idle is wrongly predicted as the preceding activity. This is currently the limitation of our approach. We are working on other ways to identify idle segments, to be combined with the pattern-based approach for the other activities.

Table 5.5: Detailed Pattern-based discovery results of van Kastaren Dataset (values as percentages)

		Idle	Leaving	Toileting	Showering	Sleeping	Breakfast	Dinner	Drink
Idle	SEG+CLU+HSMM	74	12	3	0	8	2	1	0
	SEG+CLU+PHSMM	0	6	15	14	5	5	42	0
	SEG+PCLU+PHSMM	0	6	28	14	5	0	25	22
Leaving	SEG+CLU+HSMM	4	96	0	0	0	0	0	0
	SEG+CLU+PHSMM	0	100	0	0	0	0	0	0
	SEG+PCLU+PHSMM	0	100	0	0	0	0	0	0
Toileting	SEG+CLU+HSMM	4	1	78	5	4	0	0	0
	SEG+CLU+PHSMM	0	1	75	4	4	0	2	0
	SEG+PCLU+PHSMM	0	1	88	4	3	0	1	1
Showering	SEG+CLU+HSMM	9	8	1	76	0	0	0	0
	SEG+CLU+PHSMM	0	0	2	97	0	1	0	0
	SEG+PCLU+PHSMM	0	0	2	97	0	0	0	1
Sleeping	SEG+CLU+HSMM	35	21	0	0	44	0	0	0
	SEG+CLU+PHSMM	0	0	3	0	97	0	0	0
	SEG+PCLU+PHSMM	0	0	3	0	97	0	0	0
Breakfast	SEG+CLU+HSMM	18	0	5	0	9	68	0	0
	SEG+CLU+PHSMM	0	0	11	0	1	74	14	0
	SEG+PCLU+PHSMM	0	0	11	0	1	67	3	19
Dinner	SEG+CLU+HSMM	23	2	0	0	0	36	38	0
	SEG+CLU+PHSMM	0	1	1	0	0	1	97	0
	SEG+PCLU+PHSMM	0	1	0	0	0	2	85	13
Drink	SEG+CLU+HSMM	9	1	1	0	0	80	8	0
	SEG+CLU+PHSMM	0	1	4	0	0	9	86	0
	SEG+PCLU+PHSMM	0	1	4	0	0	1	26	67

Chapter 6

Conclusion and Future Work

The problem of recognizing human activities from sensor data is a popular research subject in the field of machine learning. Depending on the availability of the labeled data, recognition methods are simply divided into two categories as supervised and unsupervised. Compared to the plethora of works in the supervised approaches, not much research has so far been devoted to unsupervised ones, and this area remains largely open to prospective research. This fact led us to pose two questions: (1) how to improve the predictive performance of existing supervised activity recognition systems, and (2) how to create a framework that is able to discover activities in a fully unsupervised manner.

Chapter 4 allows us to answer the first question. We proposed mining sequential patterns characterizing activities and integrating them into a temporal probabilistic model, i.e. HSMM. This novel approach was shown to be successful in improving the performance of the recognition algorithms. The results of the technique showed that patterns play a significant role in determining which activity is being performed by providing supplementary information about the occurrence of activities. The information provided for each activity is unique since patterns and their match durations differ from one activity to another, allowing distinct activities to be recognized. This is especially relevant when a dataset includes similar activities in terms of sensor activations involved. 'Getting snack'-'Preparing breakfast' (House C, van Kasteren) or 'Bed to Toilet'-'Grooming' (Resident 2, CASAS), for instance, are similar in terms of interacted objects and trajectories respectively. Pattern matches for 'Getting snack' and 'Bed to Toilet' are characterized by much shorter durations when compared with those for 'Preparing breakfast' and 'Grooming', which enables the approach to distinguish them. Our method is also able to account for long-range dependencies by connecting distant time instants with gapped pattern matches. Activities taking significantly more time than the others on average, i.e. 'Going to bed'/'Sleeping', 'Leaving house' and 'Idle', are best recognized by the Pattern-based HSMM consistently for all datasets. Although pattern usage results in an increase in recognition performance, the amount of the improvement depends on the feature representation on which patterns are mined and their discriminative power. For the former, one needs to take serious consideration in choosing a feature representation that matches with the dataset properties. Since patterns inherit characteristics of a feature representation, those mined from a specific representation that suits for a dataset may be harmful for another dataset. Experiments revealed that Last-fired representation and Changepoint one are suitable for datasets in which activities are represented by object-interactions and by trajectories respectively (see Section 4.4.2). The selection of Last-fired and Changepoint representations as two alternatives was due to the fact that they perform significantly better than the others (see Section 4.2). For the latter, it was observed that having enough patterns to model majority of activities, i.e. the less discriminative ones, generally produces the best performance as long as there are patterns for "Idle" class. The absence of Idle patterns causes performance worsening owing to the confusion in the prediction of Idle, e.g. misprediction of Idle as preceding activities because of the L representation in the van Kasteren Dataset, House A or overprediction of Sleeping in the CASAS Dataset, resident 2. Concentrating on the most discriminative patterns does not allow improving results (except the van Kasteren Dataset, House A) due to the lack of patterns discriminating similar activities and also due to the lack of Idle patterns as explained.

The second question of this thesis addresses the problem of discovering activities in the absence of data annotation. In Chapter 5, we presented two multistep approaches that allow us to identify candidate activity segments, groups of similar activities, and labels of the segments in order. Candidate segments were determined based either on the distance between sensor events or on the difference between two sets of sensor events. Extracted segments were automatically clustered by using Gaussian Mixture Models to create groups of similar activities. Final labeling was obtained by employing a segmental labeling technique trained on segments labeled with cluster identifiers. We compared the results of our method with those acquired from a knowledge-based one and from our clustering step applied to the true segmentation (see Section 5.1). The evaluation showed that the proposed activity discovery framework is successful in discovering activities in many cases. Experiments revealed that the performance of the segmentation and the clustering steps is affected by the selection of the time instant-based features. Using up to n-gram representation provides the proper balance between being too general (e.g. original) and too specific (e.g. 2D-n-gram) and accounts for the differences between distinct time instants. We observed that the segmental labeling step is of great importance with its ability to recover segment fragments from spurious clusters and to refine the final labeling through correct modeling of the activity durations. In order to overcome the current limitations of the approach, i.e. distinguishing similar and interleaved activities, we proposed using activity patterns in the discovery framework (see Section 5.2) following the experience gained in the supervised case. Experiments on the true segmentation showed that enriching segment representations with the pattern-based features provides better clustering results than those found with the plain representations. When it is combined with PHSMM instead of HSMM, the new pattern-based discovery framework gains the ability to discriminate similar and interleaved activities, and outperforms the base discovery framework.

There are a number of directions to move our research forward. As far as the supervised recognition techniques are concerned, the segmental mining strategy can also be used for suggesting promising topologies for graphical models trying to directly incorporate long-range dependencies. Our segmental pattern miner extracts patterns which should approximately span activity segments. Their matches are thus natural candidates to add shortcuts as in skip-chain CRF, possibly connecting distant segments representing the same or closely related activities. Another possibility is to apply our approach to other labeling problems. The proposed method is not limited to activity recognition tasks and is readily applicable to sequential labeling problems characterized by segments of consecutive positions sharing the same label (e.g. intron-exon identification in DNA sequences). For the unsupervised case, we have been working on the approach presented as an ongoing work in Section 5.2. Although pattern-based activity discovery framework produced very promising results, we were unable to detect Idle class. As a possible solution, a preprocessing step can be performed to identify idle and non-idle segments. Patter-based activity discovery method can then be applied to the non-idle segments. Other alternative solutions include explicitly introducing patterns for Idle, or post-processing the final labeling to identify Idle.

Bibliography

Aggarwal, J. K. and Ryoo, M. S. (2011). Human activity analysis: A review. *ACM Computing Surveys*. in press.

Allen, J. F. and Ferguson, G. (1994). Actions and events in interval temporal logic. Technical report.

Atallah, L. and Yang, G.-Z. (2009). The use of pervasive sensing for behaviour profiling - a survey. *Pervasive and Mobile Computing*, 5(5):447–464.

Bao, L. and Intille, S. (2004). Activity recognition in the home setting using simple and ubiquitous sensors. In *Proc. of PERVASIVE 2004*, pages 273–297.

Ben-Hur, A. and Weston, J. (2010). A user's guide to support vector machines. *Methods in molecular biology (Clifton, N.J.)*, 609:223–239.

Bergé, L., Bouveyron, C., and Girard, S. (2012). Hdclassif: An r package for model-based clustering and discriminant analysis of high-dimensional data. *Journal of Statistical Software*, 46(6):1–29.

Bouveyron, C., Girard, S., and Schmid, C. (2007). High-dimensional data clustering. *Computational Statistics & Data Analysis*, 52(1):502–519.

Cao, D., Masoud, O. T., Boley, D., and Papanikolopoulos, N. (2009). Human motion recognition using support vector machines. *Computer Vision and Image Understanding*, 113:1064–1075.

Cardinaux, F., Brownsell, S., Hawley, M., and Bradley, D. (2008). Modelling of behavioural patterns for abnormality detection in the context of lifestyle reassurance. In *Proceedings of the 13th Iberoamerican congress on Pattern Recognition: Progress in Pattern Recognition, Image Analysis and Applications*, CIARP '08, pages 243–251.

Carter, N., Young, D., and Ferryman, J. (2006). A combined bayesian markovian approach for behaviour recognition. In *18th International Conference on Pattern Recognition, ICPR 2006.*, volume 1, pages 761–764.

Chen, C., Das, B., and Cook, D. J. (2010). A data mining framework for activity recognition in smart environments. In *Proceedings of the International Conference on Intelligent Environments*.

Chen, L., Hoey, J., Nugent, C. D., and Biswas, J. (2011a). *Activity Recognition in Pervasive Intelligent Environments*. Atlantis ambient and pervasive intelligence. Atlantis Press.

Chen, L., Hoey, J., Nugent, C. D., Cook, D. J., and Yu, Z. (2012). Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42(6):790–808.

Chen, L. and Khalil, I. (2011). *Activity Recognition in Pervasive Intelligent Environments*, volume 4 of *Atlantis Ambient and Pervasive Intelligence*, chapter 1, pages 1–31. Atlantis Press.

Chen, L., Okeyo, G., Wang, H., Sterritt, R., and Nugent, C. D. (2011b). A systematic approach to adaptive activity modeling and discovery in smart homes. In *4th International Conference on Biomedical Engineering and Informatics (BMEI)*, pages 2192–2196.

Chikhaoui, B., Wang, S., and Pigot, H. (2010). A new algorithm based on sequential pattern mining for person identification in ubiquitous environments. In *KDD Workshop on Knowledge Discovery from Sensor Data*, pages 19–28.

Chikhaoui, B., Wang, S., and Pigot, H. (2011). A frequent pattern mining approach for adls recognition in smart environments. In *Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on*, pages 248–255.

Chikhaoui, B., Wang, S., and Pigot, H. (2012). Adr-splda: Activity discovery and recognition by combining sequential patterns and latent dirichlet allocation. *Pervasive and Mobile Computing*, 8(6):845–862.

Colbry, D., Peintner, B., and Pollack, M. E. (2002). Execution monitoring with quantitative temporal bayesian networks. In *In 6 th International Conference on AI Planning and Scheduling*.

Cook, D. J. and Schmitter-Edgecombe, M. (2009). Assessing the quality of activities in a smart environment. *Methods of Information in Medicine*, 48:480–485.

Demiriz, A., Bennett, K. P., and Shawe-Taylor, J. (2002). Linear programming boosting via column generation. *Mach. Learn.*, 46:225–254.

Descheneaux, C., Latfi, F., and Lefebvre, B. (2007). Activity recognition in a smart environment. In *FICCDAT'07*.

Du, Y., Chen, F., Xu, W., and Li, Y. (2006). Recognizing interaction activities using dynamic bayesian network. In *Proceedings of the 18th International Conference on Pattern Recognition*, pages 618–621.

Du, Y., Chen, F., Xu, W., and Zhang, W. (2008). Activity recognition through multi-scale motion detail analysis. *Neurocomputing*, 71:3561–3574.

Duong, T., Phung, D., Bui, H., and Venkatesh, S. (2009). Efficient duration and hierarchical modeling for human activity recognition. *Artif. Intell.*, 173:830–856.

Duong, T. V., Bui, H. H., Phung, D. Q., and Venkatesh, S. (2005). Activity recognition and abnormality detection with the switching hidden semi-markov model. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR'05)*, pages 838–845. IEEE Computer Society.

Fine, S., Singer, Y., and Tishby, N. (1998). The hierarchical hidden markov model: Analysis and applications. *Mach. Learn.*, 32:41–62.

Frank, K., Roeckl, M., Nadales, M. J. V., Robertson, P., and Pfeifer, T. (2010). Comparison of exact static and dynamic bayesian context inference methods for activity recognition. In *Proceedings of the 8th Annual IEEE International Conference on Pervasive Computing and Communications(PerCom'10)*, pages 189–195. IEEE.

Fujimoto, T., Nakajima, H., Tsuchiya, N., Marukawa, H., Kuramoto, K., Kobashi, S., and Hata, Y. (2013). Wearable human activity recognition by electrocardiograph and accelerometer. In *IEEE 43rd International Symposium on Multiple-Valued Logic (ISMVL)*, pages 12–17.

Ghahramani, Z. (2002). Hidden markov models. chapter An Introduction to Hidden Markov Models and Bayesian Networks, pages 9–42. World Scientific Publishing Co., Inc.

Gu, T., Chen, S., Tao, X., and Lu, J. (2010). An unsupervised approach to activity recognition and segmentation based on object-use fingerprints. *Data and Knowledge Engineering*, 69:533–544.

Gu, T., Wu, Z., Tao, X., Pung, H. K., and Lu, J. (2009a). epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition. In *PerCom*, pages 1–9.

Gu, T., Wu, Z., Wang, L., Tao, X., and Lu, J. (2009b). Mining emerging patterns for recognizing activities of multiple users in pervasive computing. In *Mobile and Ubiquitous Systems: Networking Services, MobiQuitous, 2009. MobiQuitous '09. 6th Annual International*, pages 1–10.

Guan, D., Yuan, W., Lee, Y. K., Gavrilov, A., and Lee, S. (2007). Activity recognition based on semi-supervised learning. In *Proceedings of the 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications(RTCSA'07)*, pages 469–475. IEEE Computer Society.

Hamid, R., Maddi, S., Johnson, A., Bobick, A., Essa, I., and Isbell, C. (2009). A novel sequence representation for unsupervised analysis of human activities. *Artificial Intelligence*, 173(14):1221–1244.

Hasan, M. K., Lee, S., and Lee, Y.-K. (2010). Using sensor sequences for activity recognition by mining and multi-class adaboost. In *IC-AI*, pages 735–740.

He, W., Guo, Y., Gao, C., and Li, X. (2012a). Recognition of human activities with wearable sensors. *Journal on Advances in Signal Processing*, 2012(1):1–13.

He, W., Yow, K. C., and Guo, Y. (2012b). Recognition of human activities using a multiclass relevance vector machine. *Optical Engineering*, 51(1):1–12.

Heierman, E. O. and Cook, D. J. (2003). Improving home automation by discovering regularly occurring device usage patterns. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, pages 537–540, Washington, DC, USA. IEEE Computer Society.

Helal, S., Member, S., Kim, E., and Hossain, S. (2010). Scalable approaches to activity recognition research. In *Proceedings of the Workshop titled "How to do good activity recognition research? Experimental methodologies, evaluation metrics, and reproducibility issues," In conjunction with Pervasive 2010*.

Helaoui, R., Niepert, M., and Stuckenschmidt, H. (2010). A statistical-relational activity recognition framework for ambient assisted living systems. In *Proceedings of the International Symposium on Ambient Intelligence(ISAmI'10)*.

Hirate, Y. and Yamana, H. (2006). Generalized sequential pattern mining with item intervals. *Journal of Computers*, 1(3):51–60.

Hong, X. and Nugent, C. D. (2011). Implementing evidential activity recognition in sensorised homes. *Technology and Health Care*, 19(1):37–52.

Hong, X. and Nugent, C. D. (2013). Segmenting sensor data for activity monitoring in smart environments. *Personal Ubiquitous Computing*, 17(3):545–559.

Hongeng, S. and Nevatia, R. (2003). Large-scale event detection using semi-hidden markov models. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, pages 1455–1462. IEEE Computer Society.

Hu, D. H. and Yang, Q. (2008). Cigar: Concurrent and interleaving goal and activity recognition. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence(AAAI'08)*, pages 1363–1368. AAAI.

Hu, F. and Hao, Q. (2012). *Intelligent Sensor Networks: The Integration of Sensor Networks, Signal Processing and Machine Learning*. CRC Press.

Huynh, T., Fritz, M., and Schiele, B. (2008). Discovery of activity patterns using topic models. In *Proceedings of the 10th international conference on Ubiquitous computing*, UbiComp '08, pages 10–19, New York, NY, USA. ACM.

Huỳnh, T. and Schiele, B. (2006). Unsupervised discovery of structure in activity data using multiple eigenspaces. In *Proceedings of the Second international conference on Location- and Context-Awareness*, LoCA'06, pages 151–167. Springer-Verlag.

Inomata, T., Naya, F., Kuwahara, N., Hattori, F., and Kogure, K. (2009). Activity recognition from interactions with objects using dynamic bayesian network. In *Proceedings of the 3rd ACM International Workshop on Context-Awareness for Self-Managing Systems(Casemans'09)*, pages 39–42. ACM.

Jakkula, V. R., Crandall, A. S., and Cook, D. J. (2009). Enhancing anomaly detection using temporal pattern discovery. pages 175–194.

Katz, S., Downs, T., Cash, H., and Grotz, R. (1970). Progress in the development of the index of adl. *Gerontologist*, 10:20–30.

Khosravi, H. and Bina, B. (2010). A survey on statistical relational learning. In *Proceedings of the 23rd Canadian conference on Advances in Artificial Intelligence*, AI'10, pages 256–268. Springer-Verlag.

Kim, E., Helal, S., and Cook, D. (2010). Human activity recognition and pattern discovery. *IEEE Pervasive Computing*, 9:48–53.

Kolovou, X. and Maglogiannis, I. (2010). Video-surveillance and context aware system for activity recognition. In *Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments(PETRA'10)*, pages 64:1–64:2. ACM.

Krahnstoever, N., Rittscher, J., Tu, P., Chean, K., and Tomlinson, T. (2005). Activity recognition using visual tracking and rfid. In *Proceedings of the Seventh IEEE Workshops on Application of Computer Vision(WACV/MOTION'05)*, pages 494–500. IEEE Computer Society.

Kulic, D., Takano, W., and Nakamura, Y. (2007). Representability of human motions by factorial hidden markov models. In *Proceedings of the International Conference on Intelligent Robots and Systems(IROS'07)*, pages 2388–2393.

Laerhoven, K. V., Kern, N., Gellersen, H.-W., and Schiele, B. (2003). Towards a wearable inertial sensor network. In *In Proceedings of IEE Eurowearable*, pages 125–130.

Landwehr, N. (2008). Modeling interleaved hidden processes. In *Proceedings of the 25th international conference on Machine learning*, pages 520–527. ACM.

Li, C., Yang, Q., Wang, J., and Li, M. (2012). Efficient mining of gap-constrained subsequences and its various applications. *ACM Transactions on Knowledge Discovery from Data*, 6(1):2:1–2:39.

Liao, L., Fox, D., and Kautz, H. (2007). Extracting places and activities from gps traces using hierarchical conditional random fields. *International Journal of Robotics Research*, 26:119–134.

Liao, L., Patterson, D. J., Fox, D., and Kautz, H. (2003). Inferring high-level behavior from low-level sensors. In *Proceedings of the International Conference on Ubiquitous Computing (UBICOMP-03)*, pages 73–89. Springer.

Liao, W. and Ji, Q. (2009). Learning bayesian network parameters under incomplete data with domain knowledge. *Pattern Recognition*, 42(11):3046–3056.

Lin, J., Keogh, E., Lonardi, S., and Patel, P. (2002). Finding motifs in time series. In *In Second Workshop on Temporal Data Mining, (KDD'02)*, pages 53–68.

Lin, W., Sun, M.-T., Poovendran, R., and Zhang, Z. (2008). Human activity recognition for video surveillance. In *Proceedings of the IEEE International Symposium on Circuits and Systems(ISCAS'08)*, pages 2737–2740. IEEE.

Liu, X. and Chua, C.-S. (2010). Multi-agent activity recognition using observation decomposed hidden markov models. *Image and Vision Computing*, 24:166–175.

Logan, B., Healey, J., Philipose, M., Tapia, E. M., and Intille, S. (2007). A long-term evaluation of sensing modalities for activity recognition. In *Proceedings of the 9th international conference on Ubiquitous computing*, UbiComp '07, pages 483–500. Springer-Verlag.

Lustrek, M. and Kaluza, B. (2009). Fall detection and activity recognition with machine learning. *Informatica*, 33:205–212.

Mabroukeh, N. R. and Ezeife, C. I. (2010). A taxonomy of sequential pattern mining algorithms. *ACM Comput. Surv.*, 43(1):3:1–3:41.

Marhasev, E., Hadad, M., and Kaminka, G. A. (2006). Non-stationary hidden semi markov models in activity recognition. In *In Proceedings of the AAAI Workshop on Modeling Others from Observations*.

Maurer, U., Smailagic, A., Siewiorek, D. P., and Deisher, M. (2006). Activity recognition and monitoring using multiple sensors on different body positions. In *International Workshop on Wearable and Implantable Body Sensor Networks(BSN'06)*, pages 113–116.

Minnen, D., Starner, T., Essa, I., and Isbell, C. (2007). Improving activity discovery with automatic neighborhood estimation. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 2814–2819, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Muncaster, J. and Ma, Y. (2007a). Activity recognition using dynamic bayesian networks with automatic state selection. In *Proceedings of the IEEE Workshop on Motion and Video Computing(WMVC'07))*, pages 30–38.

Muncaster, J. and Ma, Y. (2007b). Hierarchical model-based activity recognition with automatic low-level state discovery. *Journal of Multimedia*, 2(5).

Natarajan, P. and Nevatia, R. (2007). Hierarchical multi-channel hidden semi markov models. In *Proc. of IJCAI'07*, pages 2562–2567.

Nazerfard, E. and Cook, D. J. (2012). Bayesian networks structure learning for activity prediction in smart homes. In *8th International Conference on Intelligent Environments (IE)*, pages 50–56.

Nazerfard, E., Das, B., Holder, L. B., and Cook, D. J. (2010). Conditional random fields for activity recognition in smart environments. In *Proceedings of the 1st ACM International Health Informatics Symposium(IHI'10)*, pages 282–286. ACM.

Nowozin, S., Bakir, G., and Tsuda, K. (2007). Discriminative subsequence mining for action classification. In *Proc. ICCV 2007*, pages 1–8.

Oliver, N. and Horvitz, E. (2005). A comparison of hmms and dynamic bayesian networks for recognizing office activities. In *Proceedings of the 10th international conference on User Modeling*, UM'05, pages 199–209.

Ou, Y., Cao, L., and Zhang, C. (2009). Adaptive anomaly detection of coupled activity sequences. *IEEE Intelligent Informatics Bulletin*, 10:12–16.

Palmes, P., Pung, H. K., Gu, T., Xue, W., and Chen, S. (2010). Object relevance weight pattern mining for activity recognition and segmentation. *Pervasive and Mobile Computing*, 6(1):43–57.

Parkka, J., Cluitmans, L., and Ermes, M. (2010). Personalization algorithm for real-time activity recognition using pda, wireless motion bands, and binary decision tree. *IEEE Transactions on Information Technology in Biomedicine*, 14(5):1211–1215.

Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., and Hsu, M.-C. (2004). Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Trans. on Knowl. and Data Eng.*, 16:1424–1440.

Peng, Q. (2009). Bayesian networks for data prediction. In *Proceedings of the 2009 International Forum on Computer Science-Technology and Applications - Volume 01*, IFCSTA '09, pages 101–102. IEEE Computer Society.

Philipose, M., Fishkin, K., Perkowitz, M., Patterson, D., Fox, D., Kautz, H., and Hähnel, D. (2004). Inferring activities from interactions with objects. *IEEE Pervasive Computing*, 3:50–57.

Pirsiavash, H. and Ramanan, D. (2012). Detecting activities of daily living in first-person camera views. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2847–2854. IEEE.

Pusiol, G., Bremond, F., and Thonnat, M. (2010). Trajectory based activity discovery. In *Proceedings of the 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, AVSS '10, pages 270–277. IEEE Computer Society.

Qian, H., Mao, Y., Xiang, W., and Wang, Z. (2010). Recognition of human activities using svm multi-class classifier. *Pattern Recognition Letters*, 31(2):100–111.

Quattoni, A., Wang, S., Morency, L.-P., Collins, M., and Darrell, T. (2007). Hidden conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:1848–1852.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applicationsi in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286.

Räsänen, O. (2012). Hierarchical unsupervised discovery of user context from multivariate sensory data. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 2105–2108.

Rashidi, P. and Cook, D. J. (2010). Mining sensor streams for discovering human activity patterns over time. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 431–440. IEEE Computer Society.

Rashidi, P., Cook, D. J., Holder, L. B., and Schmitter-Edgecombe, M. (2011). Discovering activities to recognize and track in a smart environment. *IEEE Trans. on Knowl. and Data Eng.*, 23(4):527–539.

Reisberg, B., Finkel, S., Overall, J., Schmidt-Gollas, N., Kanowski, S., Lehfeld, H., Hulla, F., Sclan, S. G., Wilms, H.-U., Heininger, K., Hindmarch, I., Stemmler, M., Poon, L., Kluger, A., Cooler, C., Bergener, M., Hugonot-Diener, L., Robert, P. H., and Erzigkeit, H. (2001). The alzheimer's disease activities of daily living international scale (adl-is). *International Psychogeriatrics*, 13:163–181.

Richardson, M. and Domingos, P. (2006). Markov logic networks. *Machine Learning*, 62(1-2):107–136.

Rivera-Illingworth, F., Callaghan, V., and Hagras, H. (2005). A neural network agent based approach to activity detection in ami environments. In *Proceedings of the IEE International Workshop on Intelligent Environments(IE'05)*, pages 92–99.

Robertson, N. and Reid, I. (2006). A general method for human activity recognition in video. *Computer Vision and Image Understanding*, 104:232–248.

Roggen, D., Förster, K., Calatroni, A., and Tröster, G. (2013). The adarc pattern analysis architecture for adaptive human activity recognition systems. *Journal of Ambient Intelligence and Humanized Computing*, 4:169–186.

Russell, S. J. and Norvig, P. (2002). *Artificial Intelligence: A Modern Approach*. Pearson Education.

Salah, A. A., Pauwels, E., Tavenard, R., and Gevers, T. (2010). T-patterns revisited: Mining for temporal patterns in sensor data. *Sensors*, 10(8):7496–7513.

Sanchez, D., Tentori, M., and Favela, J. (2007). Hidden markov models for activity recognition in ambient intelligence environments. In *Proceedings of the Eighth Mexican International Conference on Current Trends in Computer Science(ENC'07)*, pages 33–40. IEEE Computer Society.

Sanghai, S., Domingos, P., and Weld, D. (2005). Relational dynamic bayesian networks. *Journal of Artificial Intelligence Research*, 24:1–39.

Sarawagi, S. and Cohen, W. W. (2004). Semi-markov conditional random fields for information extraction. In *In Advances in Neural Information Processing Systems 17*, pages 1185–1192.

Shen, J. (2004). Machine learning for activity recognition. Technical report, Oregon State University. Dept. of Computer Science.

Sim, K., Phua, C., Yap, G.-E., Biswas, J., and Mokhtari, M. (2011). Activity recognition using correlated pattern mining for people with dementia. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC*, pages 7593–7597.

Skounakis, M., Craven, M., and Ray, S. (2003). Hierarchical hidden markov models for information extraction. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 427–433. Morgan Kaufmann.

Sutton, C. and McCallum, A. (2007). An introduction to conditional random fields for relational learning. In Getoor, L. and Taskar, B., editors, *Introduction to Statistical Relational Learning*. MIT Press.

Tapia, E. M., Intille, S. S., and Larson, K. (2004). Activity recognition in the home using simple and ubiquitous sensors. *Pervasive Computing*, pages 158–175.

Tian, J., Gu, H., Gao, C., and Lian, J. (2010). Local density one-class support vector machines for anomaly detection. *Nonlinear Dynamics*, pages 1–4.

Tran, S. D. and Davis, L. S. (2008). Event modeling and recognition using markov logic networks. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, pages 610–623. Springer-Verlag.

Vahdatpour, A., Amini, N., and Sarrafzadeh, M. (2009). Toward unsupervised activity discovery using multi-dimensional motif detection in time series. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pages 1261–1266. Morgan Kaufmann Publishers Inc.

Vahdatpour, A. and Sarrafzadeh, M. (2010). Unsupervised discovery of abnormal activity occurrences in multi-dimensional time series, with applications in wearable systems. In *Proceedings of the SIAM International Conference on Data Mining*, SDM'10, pages 641–652.

Vail, D., Veloso, M., and Lafferty, J. (2007). Conditional random fields for activity recognition. In *Proc. AAMAS 2007*, pages 1331–1338.

van Kasteren, T., Englebienne, G., and Kröse, B. (2010a). Activity recognition using semi-markov models on real world smart home datasets. *Ambient Intelligence and Smart Environments thematic issue on Smart Homes*, 2(3):311–325.

van Kasteren, T., Englebienne, G., and Kröse, B. (2010b). Human activity recognition from wireless sensor network data:benchmark and software. In *Activity Recognition in Pervasive Intelligent Environment*, pages 165–186. Atlantis Press Book.

van Kasteren, T. and Krose, B. (2007). Bayesian activity recognition in residence for elders. In *Proceedings of third International Intelligent Environments Conference(IE'07)*, pages 209–212.

van Kasteren, T., Noulas, A., Englebienne, G., and Kröse, B. (2008a). Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing*, UbiComp '08, pages 1–9. ACM.

van Kasteren, T., Noulas, A. K., and Krose, B. (2008b). Conditional random fields versus hidden markov models for activity recognition in temporal sensor data. In *Proceedings of the 14th Annual Conference of the Advanced School for Computing and Imaging*.

Vilchis, J. L. P., Bremond, F., and Thonnat, M. (2010). Activity discovery from video employing soft computing relations. In *2010 IEEE International Joint Conference on Neural Networks*, pages 1–8, Barcelone, Espagne. IEEE.

Vishwakarma, S. and Agrawal, A. (2012). A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*, pages 1–27.

Watson, J. B., Ward, J. A., and Gellerson, H. G. (2010). Analysis of markov logic networks for activity recognition. online:http://www.lancs.ac.uk/ug/watsonj2/paper.pdf.

Weiser, M. (1991). The Computer for the 21st Ceuntury. *Scientific American*, 3(265):94–104.

Wiliem, A., Madasu, V. K., Boles, W. W., and Yarlagadda, P. K. (2009). Adaptive unsupervised learning of human actions. In *The 3rd International Conference on Imaging for Crime Detection and Prevention*. IEEE.

Wren, C. R. and Tapia, E. M. (2006). Toward scalable activity recognition for sensor networks. *In Lecture Notes in Computer Science*, pages 168–185.

Wu, C. and Aghajan, H. (2009). Using context with statistical relational models – object recognition from observing user activity in home environment. In *Proceedings of the Workshop on Use of Context in Vision Processing(UCVP'09)*, pages 5:1–5:6. ACM.

Wyatt, D., Philipose, M., and Choudhury, T. (2005). Unsupervised activity recognition using automatically mined common sense. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 1*, AAAI'05, pages 21–27. AAAI Press.

Yan, Z., Chakraborty, D., Misra, A., Jeung, H. Y., and Aberer, K. (2012). Sammple: Detecting semantic indoor activities in practical settings using locomotive signatures. In *16th International Symposium on Wearable Computers (ISWC)*, pages 37–40.

Yang, J., Lee, J., and Choi, J. (2012). Activity recognition based on rfid object usage for smart mobile devices. *Journal of Computer Science and Technology*, 26(2):239–246.

Yang, J.-Y., Wang, J.-S., and Chen, Y.-P. (2008). Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. *Pattern Recognition Letters*, 29:2213–2220.

Yang, X., Dinh, A., and Chen, L. (2010). Implementation of a wearerable real-time system for physical activity recognition based on naive bayes classifier. In *Proceedings of the The 4th International Conference on Bioinformatics and Biomedical Engineering(iCBBE2010)*, pages 101–105.

Yu, S.-Z. (2010). Hidden semi-markov models. *Artificial Intelligence*, 174(2):215–243.

Zhanchun, L., Zhitang, L., and Bin, L. (2006). Anomaly detection system based on principal component analysis and support vector machine. *Wuhan University Journal of Natural Sciences*, 11(6):1769–1772.

Zhang, Y., Zhang, Y., Swears, E., Larios, N., Wang, Z., and Ji, Q. (2013). Modeling temporal interactions with interval temporal bayesian networks for complex activity recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2468–2483.

Zhu, C. and Sheng, W. (2009). Human daily activity recognition in robot-assisted living using multi-sensor fusion. In *Proceedings of the IEEE international conference on Robotics and Automation(ICRA'09)*, pages 3644–3649.

Zhu, X. and Wu, X. (2007). Mining complex patterns across sequences with gap requirements. In *Proceedings of the 20th international joint conference on Artificial intelligence*, IJCAI'07, pages 2934–2940.