THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

A Framework for Mobile Activity Recognition

Jiahui Wen

B.E. (Computer science & Technology), M.Tech(Computer science & Technology)

A thesis submitted for the degree of Doctor of Philosophy at

The University of Queensland in 2017

School of Information Technology & Electrical Engineering

Abstract

Activity recognition is being applied in an increasing number of applications. They include health monitoring of the elderly, discovery of frequent behavioural patterns, monitoring of daily life activities (e.g. eating, tooth brushing, sleeping), and analysis of exercise activities (e.g. swimming, running). Current approaches for activity recognition usually use the process of data preprocessing, feature extraction, activity model learning and activity recognition. Most of the previous research pipeline these steps and create static models for processing activity data and recognizing activities. The static models have predefined data sources that are tightly coupled with the models and never change once the models are created. However, the static models are unable to deal with sensor failures and sensor replacements that are quite common in real scenarios. Moreover, additional information provided by newly available data sources from dynamically discovered new sensors may potentially refine the activity model if this information can discriminatively characterize a specific activity class. However, the static models cannot leverage this additional information for self-refinement due to the static assumption of data sources.

The primary goal of our research is to design and develop frameworks for activity recognition with dynamically available data sources, and propose and develop algorithms for activity model adaptation with the additional information provided by those data sources. In this thesis, we first provide a critical literature review in the areas of contexts modelling, context management, sensor modelling and sensors in mobile devices, activity recognition, activity model retraining and adaptation, and sensor dynamics in activity recognition. We then present the research on our activity recognition framework that makes the following key contributions.

First, we propose a hybrid method that integrates Latent Dirichlet Allocation with conventional classifiers for learning a generic activity model with minimum annotated data. The hybrid method is able to alleviate the problem of data sparsity and requires a little amount of labelled activity data. Furthermore, it can deal with different variants of activity patterns since it is created with activity data of multiple users. The generic activity modelling serves as the starting point of our activity model adaptation with dynamically available sensor data. However, it can also serve as an independent component for other applications such as activity personalization.

Second, based on the generic model, we propose a framework for low-level activity (e.g. running, walking) recognition with dynamically available sensors. The components of the

framework include a basic classifier, instance selection and smoothing. Firstly, we use AdaBoost as our basic classifier as it is flexible with feature dimensionality and it can automatically select the discriminative features during the learning process. Secondly, we propose to select the most informative instances for activity model adaptation in an unsupervised manner. The instances contain features of the new sensor data, and the information of new sensors are incorporated seamlessly through the adaptation process. Finally, we design smoothing methods by integrating the graphical models such as Hidden Markov Model and Conditional Random Field with the basic classifier AdaBoost.

Finally, we propose a framework for high-level activity (e.g, making coffee) recognition with dynamically available contexts. We propose sensor and activity models to address sensor heterogeneity and populating contextual information. Knowledge-driven and data-driven methods are proposed for incorporating the new contexts. The knowledge-driven method specifies the parameters of the new contexts with external knowledge in an unsupervised manner, and the data-driven method learns the parameters of the new contexts with the users' data using the proposed learning-to-rank technique and temporal regularization. Extensive experiments and comprehensive comparisons demonstrate the effectiveness of the proposed frameworks.

Declaration by author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my research higher degree candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the policy and procedures of The University of Queensland, the thesis be made available for research and study in accordance with the Copyright Act 1968 unless a period of embargo has been approved by the Dean of the Graduate School.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis.

Publications during candidature

[1] Jiahui Wen, Jadwiga Indulska, Mingyang Zhong, **Adaptive Activity Learning with Dynamically Available Context**, In Proc. of the IEEE International Conference on Pervasive Computing and Communications (PerCom), Sydney, Australia, March 2016.

[2] Mingyang Zhong, Jiahui Wen, Peizhao Hu, Jadwiga Indulska, **Advancing Android Activity Recognition Service with Markov Smoother**, In Proc. of the IEEE PerCom workshop on Context and Activity Modelling and Recognition (CoMoRea), St. Louis, USA, March, 2015.

[3] Jiahui Wen, Mingyang Zhong, Jadwiga Indulska, **Creating General Model for Activity Recognition with Minimum Labelled Data**, In Proc. of the ACM International Symposium on Wearable Computers (ISWC), Osaka, Japan, September, 2015.

[4] Jiahui Wen, Seng Loke, Jadwiga Indulska, Mingyang Zhong, **Sensor-based Activity Recognition with Dynamically Added Context**, In Proc. of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous), Coimbra, Portugal, July 2015.

[5] Jiahui Wen, Jadwiga Indulska, Zhiying Wang, **Discovering Latent Structures for Activity Recognition in Smart Environments**, In Proc. of the IEEE International Conference on Ubiquitous Intelligence and Computing (UIC), Bali, Indonesia, December 2014.

Publications included in this thesis

Jiahui Wen, Mingyang Zhong, Jadwiga Indulska, **Creating General Model for Activity Recognition with Minimum Labelled Data**, In Proc. of the ACM International Symposium on Wearable Computers (ISWC), Osaka, Japan, September, 2015. - Incorporated in Chapter 3.

Contributor	Statement of contribution
Author Jiahui Wen (Candidate)	Problem identification and concept design (90%) Methodology design (80%) Experimental evaluation (100%) Paper writing and editing (80%)
Author Mingyang Zhong	Problem identification and concept design (10%) Methodology design (10%)
Author Jadwiga Indulska	Paper writing and editing (20%) Methodology design (10%)

Jiahui Wen, Seng Loke, Jadwiga Indulska, Mingyang Zhong, **Sensor-based Activity Recognition with Dynamically Added Context**, In Proc. of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous), Coimbra, Portugal, July 2015. - Incorporated in Chapter 4.

Contributor	Statement of contribution
Author Jiahui Wen (Candidate)	Problem identification and concept design (80%) Methodology design (80%) Experimental evaluation (100%) Paper writing and editing (60%)
Author Seng Loke	Problem identification and concept design (10%) Methodology design (20%) Paper writing and editing (15%)
Author Jadwiga Indulska	Problem identification and concept design (5%) Paper writing and editing (15%)
Author Mingyang Zhong	Problem identification and concept design (5%) Paper writing and editing (10%)

Jiahui Wen, Jadwiga Indulska, Mingyang Zhong, **Adaptive Activity Learning with Dynamically Available Context**, In Proc. of the IEEE International Conference on Pervasive Computing and Communications (PerCom), Sydney, Australia, March 2016. - Incorporated in sections of Chapter 5.

Contributor	Statement of contribution
Author Jiahui Wen (Candidate)	Problem identification and concept design (70%) Methodology design (80%) Experimental evaluation (100%) Paper writing and editing (50%)
Author Jadwiga Indulska	Problem identification and concept design (20%) Methodology design (20%) Paper writing and editing (35%)
Author Mingyang Zhong	Problem identification and concept design (10%) Paper writing and editing (15%)

Contributions by others to the thesis

Prof Jadwiga Indulska, Dr Seng Loke and Dr Peizhao Hu provided invaluable guidance and feedback throughout the work presented in this thesis, and assisted in concept development, system design, thesis revision.

Statement of parts of the thesis submitted to qualify for the award of another degree

None.

Acknowledgements

I would like to express the deepest and most sincere gratitude to my advisory team, Professor Jadwiga Induska, Dr Seng Loke and Dr Peizhao Hu for all the invaluable guidance and advice they have provided to me over the past few years. Without their encouragement and endless support, I would not have been able to finish this thesis.

In particular I thank my principal supervisor, Professor Indulska who helped me in applying for the PhD program at The University of Queensland. She also helped me in getting a living scholarship from the National ICT Australia (NICTA). She introduced me to the field of context awareness, activity recognition and provided the best support throughout my PhD study.

I would like to thank Dr Seng Loke. He is an Associate Professor at La Trobe University. He offered me very sound advice on a range of aspects related to my research and my research publications.

I would like to thank Dr Peizhao Hu. He is one of the past PhD students of Prof Indulska. He helped me in developing my logical thinking. His encouragement throughout my PhD is invaluable.

Special thanks also go to my colleague Dr Mingyang Zhong for discussing ideas.

Finally I would like to thank my family who supported me through the whole journey of my PhD.

Keywords

Activity recognition, activity learning, activity adaptation, regularization, hybrid model.

Australian and New Zealand Standard Research Classifications (ANZSRC)

ANZSRC code: 080504, Ubiquitous Computing, 60%

ANZSRC code: 080109, Pattern Recognition and Data Mining, 30%

ANZSRC code: 080502, Mobile Technologies, 10%

Fields of Research (FoR) Classification

FoR code: 0805, Distributed Computing, 70%

FoR code: 1005, Communications Technologies, 30%

Table of Contents

List of Figures

List of Tables

List of Abbreviations

The abbreviated terms are provided for reference throughout the thesis.

LDA - Latent Dirichet Allocation

EM - Expectation-Maximization

HMM - Hidden Markov Model

CRF - Conditional Random Field

SVM - Support Vector Machine

KNN - k-nearest neighbors

LDA - Linear Discriminant Analysis

ANN - Artificial Neural Network

DBN - Dynamic Bayesian Network

NLP - Natural Language Processing

DPGMM - Dirichlet Process Gaussian Mixture Models

CML - Context Modelling Language

CHAPTER 1

Introduction

1.1 Motivation

Activity recognition has experienced its wide application in the past decades. For example, recognizing human lifestyle can help to evaluate energy expenditure [3]; monitoring human activity in smart homes enables just-in-time activity guidance provisioning for elderly people and those suffering from cognitive deficiencies [20]; detecting walk and counting steps can help to monitor elderly health [15]. On the one hand, constantly emerging sensing devices and sensing techniques provide an unprecedented opportunity for recognizing different activities. The examples include resistive pressure sensing matrix [137] for recognizing gym exercises, sensors [74] on smartphone for classifying daily activities, thermal sensors [49] for monitoring household activities. On the other hand, some activity recognition applications are driven by social needs. For example, the world wide growing elderly population [1] requires the development of activity monitoring programs for the elderly safety. Fitness tracking functions [2] become popular in social network applications, and they enable the users to monitor their daily activity level.

Currently, most of the research approaches in this area create static models for processing activity data and recognizing activities. The static models have pre-defined data sources that never change once the models are created. However, sensor failure and sensor replacement/addition are quite common in real scenarios, and this has been demonstrated

[1]https://data.oecd.org/pop/elderly-population.htm
[2]https://www.techinasia.com/wechat-sports-fitness-tracker

by Luis et al. [92] in their work. More importantly, additional information provided by dynamically available data sources may potentially refine activity models, since this information is usually correlated with a specific activity class. For example, Riboni et al. [124] find that additional location information can help to filter out impossible candidate activity classes and resolve the ambiguity. Zhan et al. [162] demonstrate in their work that vision features are able to improve the recognition accuracy for static activities, probably because static activities have similar acceleration features but have different vision signs. However, the static models cannot make use of those additional information as their data sources are pre-defined. The underlying reason is that in conventional machine learning methods for activity recognition, the sensor data is usually processed into feature vectors, and the parameters of a particular machine learning method map the feature vectors into the activity classes through a series of mathematical operations. The parameters are learned with the annotated data through minimizing the loss function (varies across different machine learning methods). With new data sources coming, the parameters corresponding to their sensors data are unknown, and the trained classifier cannot leverage the new sensor data for classification.

There exists some research that considers a dynamic sensor selection for context recognition. However, those works assume prior knowledge about the dynamically selected sensors, because they have a separate classifier for each of the sensors, or the parameters for this sensor data are pre-trained.

The primary goal of the research presented in this thesis is to design and develop activity recognition frameworks that are able to incorporate information provided by dynamically available data sources for both low-level primitive and high-level complex activity recognition, and design and develop algorithms for activity model adaptation with the new information. The low-level physical activities (e.g. running, cycling) are primitive, and they are usually recognized with on-body sensors (e.g. accelerometers, gyroscopes). On the contrary, high-level daily activities (e.g. making sandwich) are more complex, and they are usually recognized with environment-instrumented sensors (e.g. object sensors) and on-body sensors. Sensor readings of low-level activities are not semantically interpretable, so in general machine learning methods are employed to map the features extracted from continuous sensor readings to target activities. In contrast, high-level activities are characterised with contexts that are human readable, and those contexts are processed from sensor data. For example "making sandwich" can be described with location context "kitchen" and object contexts "knife" and "bread".

1.2 Challenges

Incorporating information provided by dynamically available sensors for activity recognition and activity model adaptation is a non-trivial task, and there are several challenges that need to be addressed in developing such frameworks. The challenges include:

- Challenge 1: How to learn a generic activity model that caters for people performing activities differently.

People perform activities differently due to their differences in physical conditions, age, etc. While a generic activity model can be achieved by learning the model with labelled data, annotating a large amount of activity data is expensive and time-consuming [3].

- Challenge 2: How to perform the activity model adaptation to incorporate information provided by new sensors.

New sensors result in new features for the activity model, the parameters for those features can be either determined with knowledge-driven methods, or learned with data-driven methods. The knowledge-driven methods leverage the existing knowledge base or common sense to specify those parameters, while data-driven methods select the new instances to retrain the model and learn the parameters. The challenges are:

- Challenge 3: How to get the knowledge from existing knowledge base, and how to use the knowledge for high-level activity model creation and adaptation.
- Challenge 4: How to select the most informative instances for retraining the activity model without supervision in data-driven methods.

[3] Annotating activity data costs 4-6 times more than collecting activity data [2].

For the activity model to be adapted automatically with newly available sensor data, we select the instances along with their classified class label for retraining. Existing methods for classifier retraining and adaptation usually select the instances classified with high confidence, but those high-confidence instances usually reside far from the decision boundaries and do not benefit the model refinement and accuracy improvement.

- Challenge 5: How to make sure that the newly incorporated information does not negatively impact the recognition performance of the activity model.

Incorporating more information can cause the problem of feature redundancy or model overfitting. In other words, how to select the most discriminative features to guarantee the improvement of recognition performance of the activity model after adaptation.

- Challenge 6: How to exploit the temporal information in human behaviour for both low- and high-level activity learning, activity model adaptation and activity recognition.

The advantage of temporal information in human activity recognition has been extensively demonstrated by previous work. Embedding the temporal characteristic into activity modelling and activity model adaptation is a non-trivial task.

- Challenge 7: How to deal with sensor heterogeneity for high-level activity model adaptation.

There are different types of sensors that can be used for high-level activity recognition, and the readings of sensors of different types may need to be processed differently for activity model adaptation when the sensors are dynamically available.

1.3 Thesis statement and contributions

In this thesis, we address aforementioned challenges and the shortcomings of the conventional activity recognition methods that assume pre-defined data sources and create static activity recognition models. This thesis proposes to design and develop activity recognition frameworks that are able to perform activity model adaptation when new data sources become available. The adaptation process integrates the new information into the frameworks, and hence refines the activity models in terms of recognition accuracy, scalability, robustness. The key features of the frameworks include: a collaborative semi-supervised method for creating a generic activity model with minimum labelled data from multiple users; temporal smoothing and regularization methods that leverage the temporal information in the human activity for filtering out the outliers; an instance selection method that is able to select the most informative instances for activity model adaptation without human intervention; a data-driven learning-to-rank activity model adaptation method that personalizes the activity model to a particular user to achieve high accuracy; a knowledge-driven activity model adaptation method that leverages third-party knowledge and incorporates the new sensor data without supervision. The contributions of this thesis are presented in more details in the following subsections.

1.3.1 Generic activity modelling

We design and develop a generic low-level activity modelling method that learns activity model with minimum labelled data of different users. We demonstrate that an activity model learned with data of one user cannot be scaled to others due to the differences in their physical conditions, gender and age, etc. For the activity model to be generic, we learn the model with data from multiple users so that it is able to cope with variants of activity patterns. Latent Dirichlet Allocation (LDA) is leveraged to model the activity data as it is effective in collaborative learning and is able to leverage the partially labelled data of various users to overcome the problem of data sparsity [164]. Since LDA cannot be applied to the activity data directly, we hybrid it with conventional machine learning methods. Therefore, in the hybrid approach, we first use the initial labelled data to train the conventional classifiers, and then estimate the class assignment of the activity data (labelled and unlabelled) with the hybrid method. In the third step, we use the class assignment of the activity data from the second step to retrain the classifiers. This process is repeated until it converges. The generic activity modelling serves as the starting point of our activity model adaptation with dynamically available sensor data. However, it can also serve as an independent component for other applications such as activity personalisation.

We evaluate the proposed method with a large number of datasets, and show that it outperforms the supervised method and conventional semi-supervised method. We also examine the factors (e.g. labelling percentage) that have impact on the recognition performance.

The hybrid method for creating the generic activity model addresses challenge 1. In the second step of the hybrid method, the topic assignment of a particular instance considers the topic assignment of the neighbouring instances, and hence it addresses challenge 6.

1.3.2 Physical activity recognition with dynamically available sensors

We develop a framework for low-level physical activity (e.g. walking, running) recognition with dynamically available sensors and semi-supervised learning method. The components of the framework include basic classifier, instance selection and smoothing. Specifically, we investigate conventional machine learning methods and find that AdaBoost suitably serves our purpose, as it is flexible with feature dimensionality and it can automatically select the discriminative features during the learning process. Second, we propose to select the most informative instances for retraining without human annotation. Since the sensor data from physical activities cannot be semantically interpreted, we incorporate the dynamically available sensor data by retraining the activity model with the selected instances that contain the features of the new sensor data. Third, we design smoothing methods by integrating the graphical models such as Hidden Markov Model and Conditional Random Field with the basic classifier AdaBoost. The smoothing methods leverage the temporal information embedded in the human activities that the current activity is more likely to be continued in the next time slot. Finally, we investigate the conditions under which the opportunistically discovered sensors are beneficial to the recognition performance, we propose two hypotheses and validate them with controlled experiments.

The proposed framework that is able to incorporate new sensors dynamically for low-level activity recognition addresses challenge 2; the instance selection method addresses challenge 4; the basic classifier, AdaBoost, is able to select the most discriminative features, and it addresses challenge 5; the combination with HMM and CRF leverages the temporal information and addresses challenge 6.

1.3.3 High-level activity recognition and adaptation with dynamically available contexts

We develop a framework for high-level activity recognition with dynamically discovered contexts. Specifically, we design sensor models that facilitate the pre-processing of sensor readings of the dynamically available sensors into high-level contexts, and the activity models that facilitate incorporation of those high-level contexts into the activity models. There are different types of sensors in real environments, and even the sensor readings of the same types of sensors can be interpreted differently if they are used for different purposes. The sensor models provide a description of how to process the sensor readings into proper contexts for recognizing activities, while the activity models interrelate the contexts with activities.

Second, we propose a knowledge-driven method for the incorporation of the dynamically available contexts without supervision. The parameters of the contexts with respect to different activity classes are estimated using descriptive texts of the activities from external sources (e.g. website) using natural language processing methods. High-level activities are usually described (or characterised) by different kinds of contexts, and those descriptions can be obtained from external databases. Therefore, when discovering a sensor, the parameters can be specified from the descriptive texts without human intervention.

Third, we propose a data-driven method for the incorporation of the dynamically discovered contexts with the learning-to-rank machine learning method and temporal regularization. The data-driven method is a personalized method as it performs machine learning with the activity data of a specific user. In the knowledge-driven method, the parameters of new contexts are obtained from third-party databases, while in the data-driven method the parameters are learned from the new activity data containing new contexts. The temporal regularization is embedded into the activity learning process and it encourages the neighbouring instances to have the same activity class. We also select the most informative instances for the activity model adaptation and propose a thresholding method to guarantee class balance.

The proposed framework that is able to integrate the new sensors for high-level activity recognition and activity models adaptation addresses challenge 2; the proposed knowledge-driven method addresses challenge 3; the personalized learning-to-rank machine learning method and temporal regularization in the proposed data-driven method addresses challenge 5 and 6 respectively; the proposed sensor models address challenge 7.

1.4 Thesis structure

The remainder of this thesis is structured as follows:

- Chapter 2 surveys the related work including context modelling, context management, sensor modelling, overview of sensors in mobile devices, activity recognition methods, activity model retraining and adaptation, sensor dynamic for activity recognition. We discuss shortcomings of the existing approaches and analyse why they are inapplicable in dynamic environments.
- Chapter 3 presents a generic activity modelling method that learns the generic activity model with minimum labelled data.
- Chapter 4 describes and evaluates the framework of physical activity recognition and its adaptation with dynamically available sensor data.
- Chapter 5 presents and evaluates the framework of high-level activity recognition and its adaptation with dynamically available contexts.
- Chapter 6 concludes the thesis with a summary of this thesis contributions and discusses potential future research.

Chapter 2

Critical Literature Survey

In order to create a framework for mobile activity recognition, several areas of previous related work need to be critically reviewed. They include context modelling, context management, sensor modelling and sensors in mobile devices, activity recognition, activity model adaptation and sensor dynamics in activity recognition. The reasons we review this related research are multifold. First, activity recognition in dynamic environments needs to incorporate sensors dynamically and automatically. Without prior knowledge about the sensors we do not know how to process the sensor readings and use the information they provide for recognising activities. The existing context and sensor modelling methods facilitate the process of sensor discovery, preprocessing of sensor readings, and activity modelling and adaptation, and it is necessary to evaluate how they need to be modified/refined to support activity recognition in dynamic environments. Second, by reviewing context sources in mobile devices, we show the state-of-the-art sensors that are widely used and their applications in daily lives. We also analyse the general architecture of context sensing in mobile devices and analyse whether it can be applied to activity recognition in dynamic environments. Third, as sensor-based activity recognition has been well studied by previous work, our aim is not to extend previous artificial intelligence methods to achieve higher accuracy. Instead, our goal is to leverage the statistic and probabilistic characteristics used in artificial intelligence to address the challenges met by activity recognition in dynamic environments (i.e. opportunistic discovery of sensors, activity model adaptation). Finally, we review the research on activity model adaptation and sensor dynamics in activity recognition, that are related to our work.

In what follows, we review several context modelling and management methods, followed by sensor modelling methods and the state-of-the-art sensors in mobile devices. After that, we discuss the activity recognition techniques in the previous sensor-based activity recognition area. Finally, we review the most profound works in activity model adaptation and sensor dynamics in activity recognition. In this survey we discuss the advantages and shortcomings of the existing approaches and analyse why they are inapplicable in our case.

2.1 Context modelling and management

More than a decade of research on software engineering of context-aware applications has led to the approach that creates a context model for each context-aware application. The context management system is used for gathering, preprocessing, and reasoning upon context information on behalf of the application based on this application's context model. This approach makes the design, development, and management of context-aware applications easier and allows to reuse the gathered context for multiple context-aware applications. The approach is an extension of the distributed computing middleware - the middleware is extended by the context management system that allows application designers to design the context model (context information types required by the application, sensor readings preprocessing rules, and a logic for reasoning upon context information to recognize situations). In this section, we review representative context modelling and management methods.

2.1.1 Context modelling

Fact based context modelling

Henricksen et al. [47, 48] proposed a fact-based Context Modelling Language (CML). Central to the approach is the fact-based classification of contextual information that is of interest for a particular context-aware application. In addition, the approach also allows designers to define the rules that reason upon the gathered contextual information and recognize situations (high level abstract contexts). Those situations are usually important for the context-aware applications as they need to perform adaptation when the situations change. The

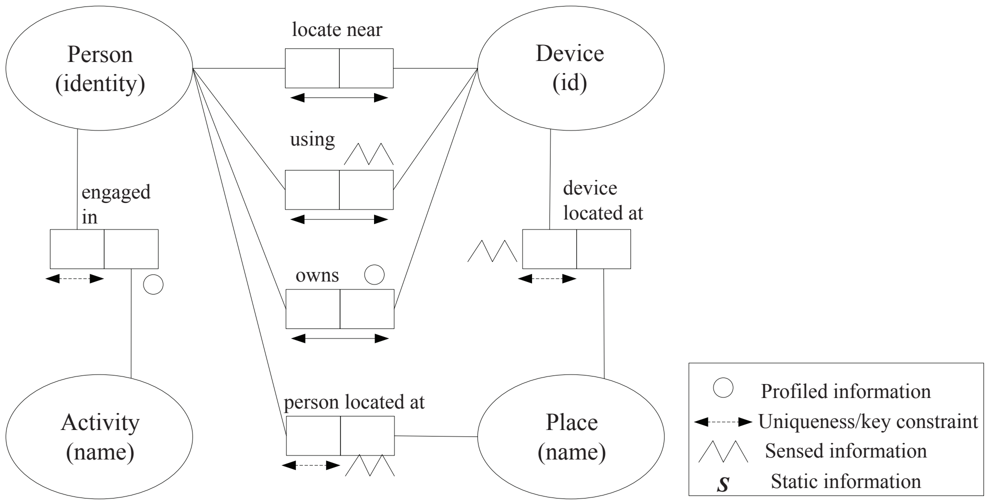


Figure 2.1: Graphical context modelling.

designers use a variant of first-order-logic to define situations that are of interest to the application. Therefore in this approach, the context information used by the application is predefined and the situations that need to be reasoned upon are also static.

The CML language is an extension of Object-Role Modelling (ORM) and models context fact types as object types (ellipses in Figure 2.1) and relationships between the object types. For example, Figure 2.1 shows facts that a person can be characterised by the other contexts such as activity or device (e.g. the person is engaged in certain activity or using a device). Context fact types are classified into four types, they are *static*, *profiled*, *sensed* and *derived* fact types. Static and profiled context information is provided by the users (static does not change in its life-time and profiled is rarely updated), while sensed data is sensor-provided (physical or logical sensors) and can be updated with increasingly available data. Derived information is derived by from available context using some derivation rules.

Ontology based context modelling

Ontological context modelling leverages the description logic (DL) to explicitly define concepts and their relationships. Therefore in ontology based context modelling, context reasoning is based on the description logic [13]. In the ontological approach, context information (e.g. activity) is organized into a hierarchical structure of classes, with each class being described by a number of properties. The datatype properties specify the characteristics of the classes, and the object properties define the relationships between classes. The proper-

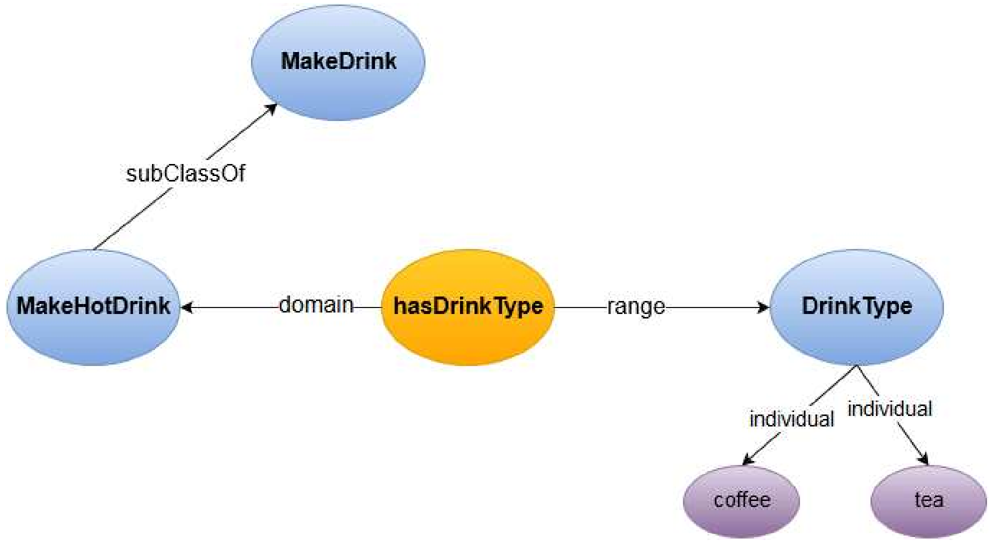


Figure 2.2: An example of ontology-based context modelling.

ties are characterized by their domain and range, with the domain referring to the classes that can be described by the properties and the range referring to restrictions of either all the classes whose instance can be assigned to the property (object property) or the datatype (datatype property).

As an example shown in Figure 2.2, the ontology class *MakeHotDrink* is a subclass of *MakeDrink*, and *hasDrinkType* is one the properties of *MakeHotDrink*. Therefore, the domain of *hasDrinkType* is *MakeHotDrink* and the range is the *DrinkType* class. The individuals of *DrinkType*, e.g. coffee and tea, are the possible values of the *hasDrinkType* property.

The hierarchy of classes in ontological modelling results in an architecture of superclass and subclass, the subclass inherits all the properties of the superclass. As such, contextual information can be arranged at multiple levels of granularity. Together with class properties and individuals (instances), the ontologies are able to capture and encode domain knowledge. The knowledge base is divided into *TBox* and *ABox*, where *TBox* contains knowledge describing class hierarchies (i.e. relations between classes) while *ABox* contains ground sentences stating where in the hierarchy individuals belong (i.e., relations between individuals and classes).

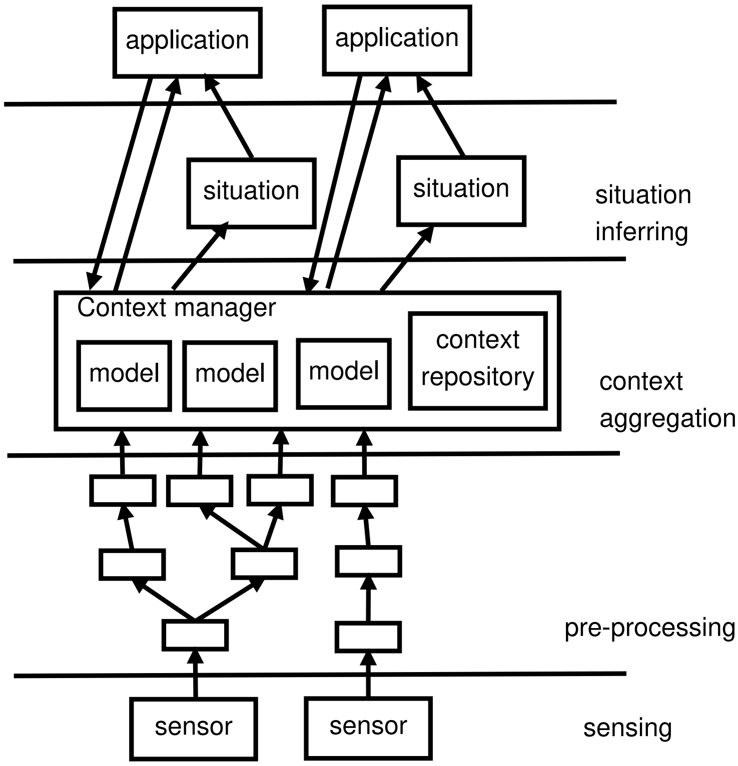


Figure 2.3: General architecture of context management system.

2.1.2 Context management

The general architecture of the context management system is illustrated in Figure 2.3. In the low layer, sensor readings are processed into context information that is required by the application, i.e. defined by the context model. In the middle layer, the components include the context models that describe context information types required by the application and a repository of context information that is gathered based on these models. In the higher layer, there is a situation model that describes how the gathered context is reasoned upon to derive high level abstract context (situations), hence the higher layer describes the logic that is used for reasoning upon context. The context management system provides interfaces so that the applications are able to query the context information or request notifications upon situation changes that require the applications perform adaptation.

The context model and logic used for context reasoning in this general architecture vary from one approach to another. For example, Henricken et al. define situations in the form of predicate logic: $S(v_1, \cdots, v_n) : \varphi$, where S is the name of the situation, $\{v_1, \cdots, v_n\}$ are variables and φ is logical expression in which the variables are confined to set $\{v_1, \cdots, v_n\}$. The logical expression consist of basic expressions connected through logical connectives, with each basic expression denoting one facet of the situation. At runtime, sensor readings are processed into context information and the aggregation of multiple contexts are reasoned upon against the pre-defined rules to infer high level situation.

While in ontology based methods, individuals are instantiated based on the sensor observations and a situation can be established by interlinking the multisource sensor observations, this is equivalent to relating the individuals by properties. The individuals and the relations between them, which represent the current fact, are then put into *ABox* and ontologies reasoning is performed either to check knowledge consistency or to derive new knowledge (find the most specific class for the individual). Ontology based context modelling is widely applied due to fact that it is supported by various tools. For example, the ontologies can be created and edited by graphical tools such as protege[2], several reasoners such as Pellet[3], FACT++[4], Jena[5] and OWLAPI[6] can be used to reason over the created ontologies. They even provide APIs for program languages such as Java so that the reasoning can be performed in real time progressively.

Hu et al. [52] extended the general architecture even further since they introduce autonomy into the context management system. Central to their system is the dynamic discovery and self-configuration of the context sources. In order to deal with the heterogeneity of sensors, they use Process Chains to pre-process the data from context sources into required context information, which is supported by SensorML that specifies the information required for processing sensor readings. To facilitate the dynamic discovery of context sources, the context management system introduces several managers, they include *Context Source Manager* which manages communication with/for context sources, provides sensor discovery, and registration and configuration services; *Application Context Subscription Manager* which stores context subscription defined by application designers; *Reconfiguration Manager* which performs cross-layer context mapping.

[2]http://protege.stanford.edu/
[3]http://clarkparsia.com/pellet/
[4]http://owl.man.ac.uk/factplusplus/
[5]https://jena.apache.org/
[6]http://owlapi.sourceforge.net/

2.1.3 Discussion

Both of the context modelling and management approaches have their own disadvantages. For example, CML represents all context facts uniformly in a “flat” model, which results in its lack of expressiveness for hierarchical architectures, and emphasizes unevenly on particular dimensions of context facts [13].

Hu et al. propose an autonomic context management system able to discover new sensors and provide a mapping from the required context to available sources of context information. However, they focus on reliability and self-configuration of the system and do not address the problem of adapting the reasoning techniques due to dynamically discovered context sources, especially the ones of which we do not have prior knowledge.

In ontology based context managements, even though a probabilistic ontological framework [45] has been proposed to incorporate temporal information and data uncertainty, it is still vulnerable in realistic scenarios due to its guarantee of decidable reasoning procedures. Moreover, the overhead of realization process should never be overlooked [13].

In addition, both of the mentioned approaches use a particular logic (e.g. FOL and DL) to infer high level situations. Even though expressive and easy to understand, they are vulnerable to data uncertainty and sensor noises. Moreover, the inference rules are pre-defined with the domain knowledge, hence the static inference engine makes them inapplicable in dynamic environments as they are not able to adapt situation reasoning to new types of context information gathered from newly available sensors.

2.2 Context sources

In this section, we review the representative sensor modelling methods in the previous work, followed by the prevalent sensors used for context sensing in off-the-shelf mobile devices and the general sensing architecture.

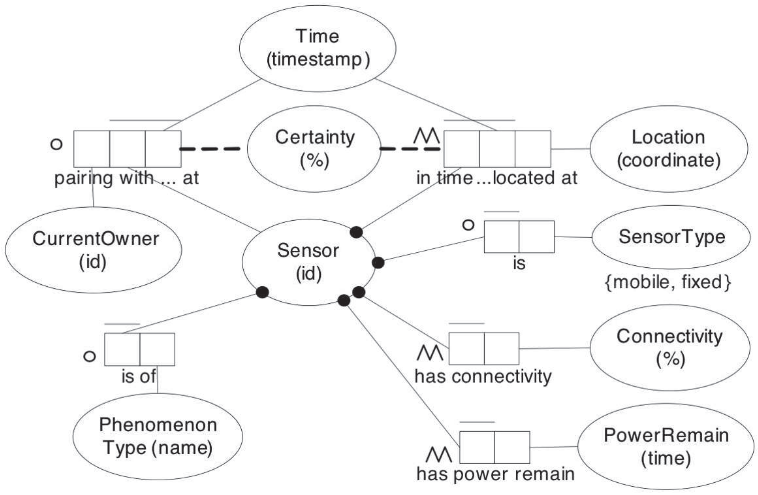


Figure 2.4: Sensor context model.

2.2.1 Sensor modelling

Sensor modelling for autonomic context management system

Hu et al. [52] propose an autonomic context management system that supports (i) dynamic discovery and self-configuration of sensors, and (ii) mapping between context information required by an application and sensor observations (the mapping may include necessary preprocessing of sensor readings to achieve the level of abstraction required by the context information). To facilitate the mapping between context information and sensor observations, they proposed a sensor context model (Figure 2.4) that is able to capture both static features and dynamically changing information for a sensor. In Figure 2.4, the sensor context model example describes only a subset of the dynamic context information required to model sensors, including ownership, location, communication connectivity and remaining power. The modelling of the sensors facilitates the dynamic incorporation of the information they provide when they are discovered and registered to the context management system.

To support opportunistic discovery of the sensor models, they advocate the use of the IEEE 1451 standard that allows the sensors to introduce themselves to external systems that they can communicate with. Central to the standard is the TEDS (Transducer Electronic Data Sheet) data that describes the sensor specifications and the TEDS template that specifies the meaning of the TEDS data. The TEDS template also instructs how to decode and extract the TEDS data from the binary encoding. As the TEDS template only describes limited sensor information, they propose a hybrid method to discover and incorporate new sensors.

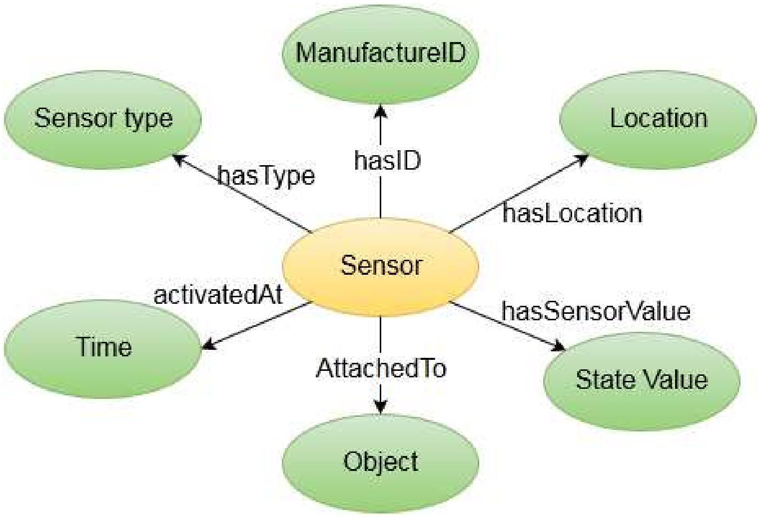


Figure 2.5: An example of sensor model in smart environment.

First, the TEDS data is extracted and decoded with the TEDS template when a sensor is discovered through physical communication interfaces. After that, the TEDS specification is mapped to the SensorML description that describes how to incorporate the sensor into the context management system; then the information provided by the sensor can be populated into the system, as shown in Figure 2.4. The mapping process is done through an ontology knowledge base that shows relationships between sensor types that can, after preprocessing, produce the same context facts.

Sensor modelling in smart environments

Chen et al. [20] propose to model sensor for unsupervised activity recognition in smart environments. As shown in Figure 2.5, the information describes a sensor is explicitly modelled, so that each sensor is linked to certain physical or semantic entity such as object, location. For instance, a contact sensor is attached to the oven in the kitchen. Through explicitly encoding these information into the sensor model, it is possible to get the object and the location when the sensor is activated, and this implies that the a user is carrying out an activity in the location with the object.

Table 2.1: Sensors and their applications.

Sensor	Related work	Application
Accelerometer, Gyroscope	[8, 101, 91, 9, 46, 113, 103, 60, 59, 88, 58, 57, 171, 33, 112]	place characterisation, motion detection and analysis, transportation mode, gesture recognition, activity recognition, context management system
Microphone	[8, 26, 90, 109, 100, 93, 118, 89, 157, 61, 43, 40, 71, 103, 148, 58, 57, 91]	place characterisation, sound detection and classification, conversation group clustering, emotion and stress sensing, speaking counting, sleeping monitoring, tooth brushing monitoring, activity recognition, context management system
GPS	[7, 73, 84, 103, 59, 148, 171, 33, 91, 112]	Localisation, significant place identification, activity recognition, speed detection
Bluetooth	[36, 112, 107, 164]	social context detection
WiFi	[68, 68, 171, 84, 103, 148]	Localisation, activity recognition
Light sensor	[8, 169, 60, 59]	place characterisation, illumination sensing
Camera	[8, 26, 33, 117]	place characterisation
Barometer	[127, 155]	transportation mode, door event

2.2.2 Sensors in mobile devices

The prevalence of smart phones has offered an unprecedented opportunity for mobile sensing as multiple sensors in smart phones can provide various context information. For example, Miluzzo et al. [101] present the CenceMe application based on off-the-shelf mobile phones. The application leverages the on-board accelerometers, microphone, GPS and Bluetooth to infer users' activity and social interactions. In this subsection, we review the kinds of context sources in mobile devices that can be used for context sensing, and the general context sensing architecture of state-of-the-art context sensing applications. Our discussion focuses on accelerometer, gyroscope, microphone, GPS, Bluetooth and WiFi access points. The related works are categorized according to the sensors they used, and the corresponding applications are also described, as listed in Table 2.1

2.2.3 Discussion

In this section, we review the sensor modelling methods and the applications of the sensors in mobile devices. From this literature we can conclude that most of the previous works fo-

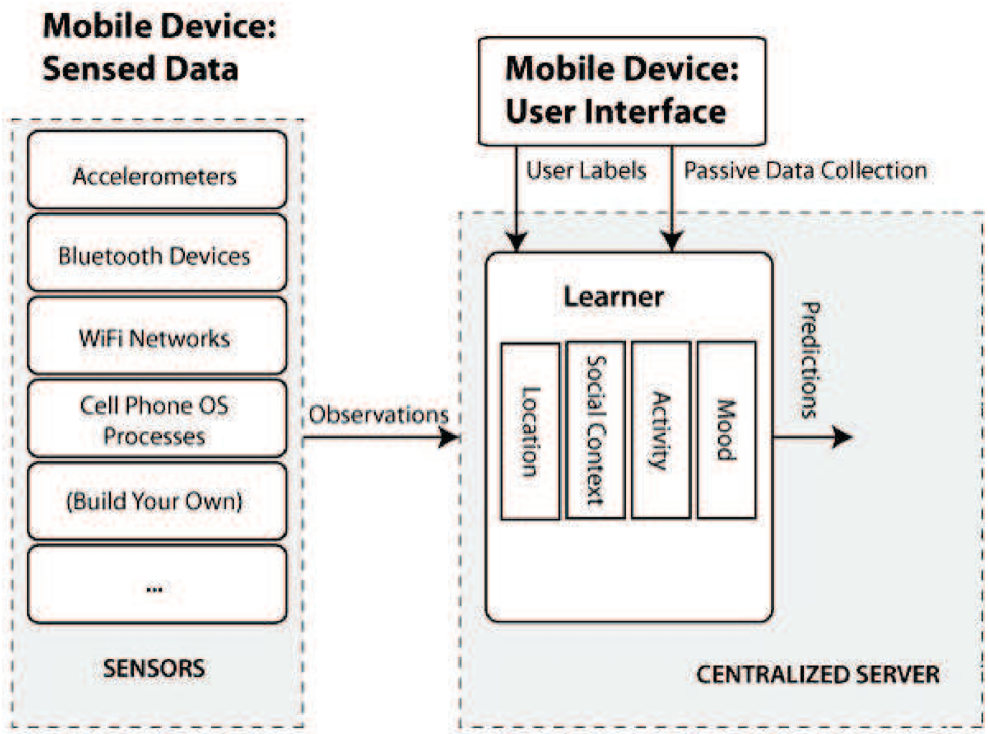


Figure 2.6: General architecture of mobile context sensing system.

cus on on-board sensors of mobile devices such as GPS, microphone and accelerometer [76]. The architecture of the systems always includes sensing, learning and prediction, as shown in Figure 2.6, in which the context sources are tightly coupled with context information in the sensing system. Some researchers assume a collaborative manner [11, 93, 79, 100, 109] of obtaining context information from other users. However, the data-flow is predefined and the systems know the semantic meaning of the data sources. As we can envisage, embedded context sources will become prevalent in ubiquitous environments and the existing techniques (i.e. sensing and learning) will become impractical when confronted with dynamically available sensors.

2.3 Sensor-based activity recognition

Human activity is one of the most important high level contexts in the area of context-aware computing, due to the fact that many context-aware applications take actions based on human activities. The following review of sensor-based activity recognition is divided into data-driven methods, knowledge-driven methods and a hybrid approach.

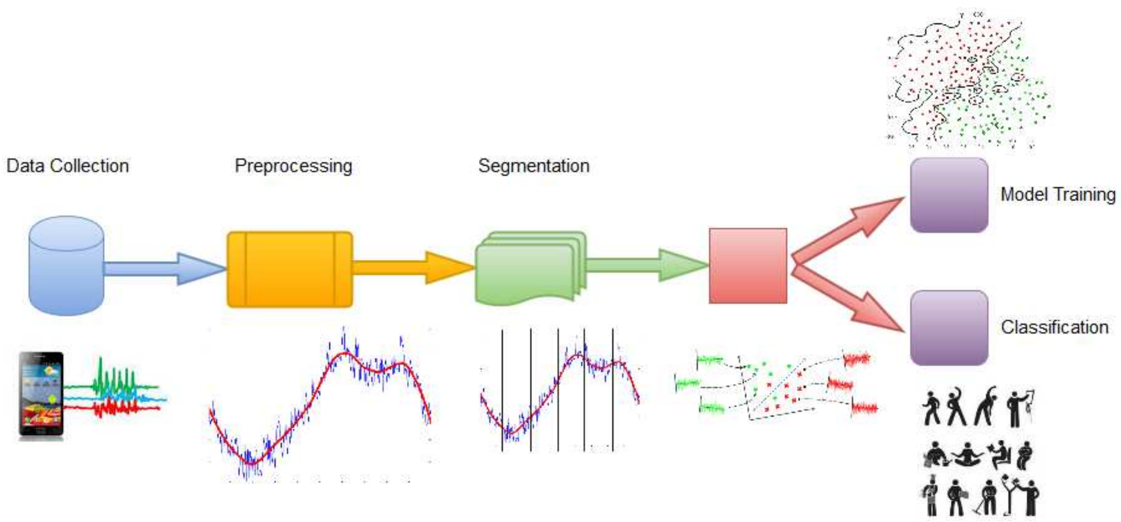


Figure 2.7: Pipeline of conventional human activity recognition.

2.3.1 Data-driven methods

With data-driven approaches to activity recognition, the activities are modelled completely from the available data, and the traditional processes can be characterised by (1) Data acquisition, (2) Preprocessing, (3) Segmentation, (4) Feature extraction, and (5) Model training and classification. Therefore, the degree of fitting that the created models have with the data is influenced by various factors, such as placement of sensors, training data, feature selection and classifiers.

Feature selection

Raw data generated from sensors are usually pre-processed into feature vectors before it is fed into recognition models. Since large number of features would result in redundancy and jeopardise the accuracy and energy-efficiency of the model, it is wise to select a subset of the features that have the most discriminative power. Another rationale behind the feature selection is that a small set of the representative features can be computationally inexpensive, and hence can greatly reduce the training and testing overhead. This is very important for mobile devices that are constrained by the limited battery power. The previous works proposed various methods for feature selection in the wearable activity recognition area. For instance, Chen et al. [22] propose an online LDA (Linear Discriminant Analysis) that is able to add or delete an instance dynamically. LDA is a dimensionality reduction method that is able to preserve the differences between classes when projecting the original feature space to a lower dimensional one. Huynh et al. [55] propose *cluster precision* which selects the features that can most precisely cluster the training instances. Könönen et al. [70] leverage the Sequential Floating Forward Selection (SFFS) to select (discard) the feature that makes the biggest (smallest) contribution to the classification accuracy.

Supervised learning

Supervised learning methods require the training data to be labelled. A lot of previous research employ the conventional machine learning methods for human activity learning and classifying. Training a recognition model is equivalent to learning the correlations between the feature vectors and activity classes, and the correlations can be interpreted as different parameters in different classifiers, such as the branch nodes in Decision Tree, support vectors in Support Vector Machine (SVM), transition and emission matrix in Hidden Markov Model (HMM), etc. Based on the way that the parameters are learned, the models can be generally classified into generative models and discriminative models.

- *Generative models*: generative models assume the observable sensor data is generated from a distribution with some hidden parameter, and explicitly modelling the joint distribution over the sensor observations and the activity classes. The parameters of the models are learned by maximising the likelihood of the training data. To simplify the modelling and inference process, assumptions are usually made to simplify the relations among the variables. However, large amounts of labelled data is required to train the model, and unseen dependency under the available training data can negatively affect the performance of the model. Commonly used generative models for human activity recognition include Naive Bayesian [10, 115, 102, 138, 143], Hidden Markov Model [120, 80, 161, 81, 153], and Dynamic Bayesian Network [114]
- *Discriminative Models*: contrary to generative models, discriminative models directly model the conditional probability distribution of latent activity classes over the sensor observations. One of the advantage of discriminative models is that all sorts of rich overlapping features can be incorporated without violating any independence assumptions [141]. Therefore, discriminative classifiers perform well even with a small amount of training data. Examples of discriminative models for activity recognition in-

clude Decision tree [86, 10, 30, 31, 98], Support Vector Machine (SVM) [70, 22, 4, 53, 72], Conditional Random Field (CRF) [144, 69, 104, 162], Decision table [120, 10], k-nearest neighbors (KNN) [120, 14, 70, 4], Artificial Neural Network (ANN) [66, 4, 67].

- *Hybrid models*: Hybrids of generative and discriminative models usually perform better than any single one of them, since they take advantage of the discriminative model to maximize the margin between different classes while utilizing the generative model to smooth the outliers. For example, in [82] the testing data is first fed into Adaboost to obtain for each data sample a posterior probability for each activity class, which in turn are regarded as feature vectors and fed into multiple HMMs. The data sample is then classified as the class corresponding to the HMM that has the maximum likelihood. Fahad et al. [32] propose the hybrid recognition by transforming the instances into the representations of distance minimisation and probability estimation with respect to different activity classes. Then they input those representations as features for the SVM training. Other hybrid methods include hybrid of Decision tree and HMM [96, 94].

- *Others*: Many approaches other than the traditional machine learning methods have been proposed for activity recognition recently. For example, Gu et al. [38] propose an emerging patterns based approach to sequential, interleaved and concurrent activity recognition. The underlying idea is to mine contrast patterns for each class that occur frequently in one class and rarely in other classes, such patterns are expected to have high discriminative power. Ghasemzadeh et al. [34] use a motion transcripts matching approach for collaborative action recognition. The idea behind this method is the similarity matching between action templates generated in the training phase and the candidate samples in the testing phase. Other works also incorporate complex models such as Skip-chain CRF [44], coupled HMM or Factorial CRF [146] to recognize interleaved or concurrent activities. However, these are out of scope for our discussion.

As the supervised machine learning method AdaBoost is used in our approach presented in the later chapters, we briefly introduce it here. The core of AdaBoost is to train an ensemble of weak classifiers and combine them to form a more robust and accurate classifier. Each weak classifier makes decisions based on a single feature and needs only be slightly better than random guessing. The final classifier is a linear combination of the weak classifiers, with each classifier being weighted by the error it makes during the training process; more weight is given to the classifier that makes fewer errors.

As depicted in Algorithm 1, the AdaBoost learning algorithm takes as input the instances, the initial instance weights and maximum iterations. The training of AdaBoost follows an iterative process. In each iteration, each weak learner is fitted to the training dataset, and the one with the minimum weighted error is chosen (step 2). After that, the instance weights are updated, so that more weights are given to the misclassified instances (step 4). During the next iteration, the weak classifiers will focus more on those problematic instances. The output of the training process is an ensemble of weak learners (step 6). Notice that in step 2, it trains a weak learner for each dimension of the feature space, but only selects the one with minimum weighted error. In our approach, we will adopt decision stump (i.e. one-level decision tree) as the weak learner, and then training weak learner $h_t^k(x)$ for dimension k is equivalent to finding the threshold θ_k in that dimension to minimise the weighted error such that $h_t^k(x_i) = h_t^k(x_i^k) = 1$ if $x_i^k > \theta_k$ and $h_t^k(x_i) = -1$ otherwise, where x_i^k is the value of k^{th} dimension of instance x_i.

Algorithm 1 AdaBoost.

Input:

Instances $(x_1, y_1), \cdots, (x_n, y_n)$ where $x_i \in \Re^k$ a is k-dimension feature vector, $y_i \in \{+1, -1\}$;
Initial weight of n instances $D_0(i) = 1/n$ for $i = 1, \cdots, n$;
Weak learners $h(x) \in \{+1, -1\}$;
Max iterations T;

Output:

Ensemble of weak learners;

1: **for** $t = 1$ to T **do**
2: Find weak learner $h_t(x)$ that minimizes the weighted error: $h_t(x) = argmin_{h_t^k(x)} \sum_{i=1}^{n} D_t(i) I[h_t^k(x_i) \neq y_i]$
$\epsilon_t = \sum_{i=1}^{n} D_t(i) I[h_t(x_i) \neq y_i]$;
3: Compute the weight for the weak learner $h_t(x)$: $\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$;
4: Update the weight of instances: $D_{t+1}(i) = \frac{D_t(i) exp(-\alpha y_i h_t(x_i))}{\sum_i D_t(i) exp(-\alpha y_i h_t(x_i))}$ for $i = 1, \cdots, n$;
5: **end for**
6: **return** $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$;

AdaBoost is a discriminative classifier, and it performs classification by giving the definitive decision. This approach has a potential problem that even if the classifier is uncertain with the class of the instance, it chooses the class against which the instance has the maximum evidence as the prediction. We argue that the posterior probability of an instance is much more helpful, since it reflects the confidence in that prediction. This is important to the later stages such as the stage of *learning to adapt*. To this end, we calculate the posterior probability for instances using the method from [82].

$$P(y_i|x_i) = \begin{cases} \frac{e^{\psi m(x)}}{e^{\psi m(x)}+1} & \text{if } y_i = +1 \\ \frac{e^{-\psi m(x)}}{e^{-\psi m(x)}+1} & \text{if } y_i = -1 \end{cases} \tag{2.3.1}$$

where ψ is a constant and $m(x) = \frac{\sum_{t=1}^{T} \alpha_t h_t(x)}{\sum_{t=1}^{T} \alpha_t}$. $P(y_i|x_i)$ is thus regarded as the posterior distribution of instance x_i. Notice that the binary AdaBoost can be easily extended to multi-class classifiers by training a set of weak learners for each activity class i to separate itself from others:

$$H_i(x) = \sum_{t=1}^{T} \alpha_t^i h_t^i(x) \tag{2.3.2}$$

Accordingly, the prediction is made by $argmax_i(H^i(x))$ for an instance x.

Semi-supervised learning

The aforementioned supervised learning methods suffer from several drawbacks. Firstly, the labelling of training data for activity modelling is time-consuming and sometimes error-prone, considering huge amount of data from sensors in realistic scenarios. Further, the supervised methods lack scalability and adaptability, because the models learned from labelled data are static, hence they cannot adapt to the evolving behaviours of the users, nor can they be scaled to other users who perform the activities differently.

To overcome the shortcomings of supervised learning, many researchers turn to the direction of semi-supervised techniques, in which the models are created with small amount of labelled data, and the unlabelled instances classified with high confidence are added into the training data to refine the models. For example, Guan [41] explore several semi-supervised learning methods for activity recognition with labelled and unlabelled data. In Co-training, two classifiers are created on two sufficient and redundant sets of attributes. In the iterative learning process, each classifier adds the instances that it classifies with high confidence to the other's training set, and the classifiers are retrained with the augmented training data.

In En-Co-training, different types of classifiers are trained on the same training data, and during the iterative learning process, the unlabelled instance which all the classifiers agree with the predicted label are used to augment the training data and retrain the classifiers. In [131], the authors propose a graph-based label propagation semi-supervised learning. The graph defines the similarity between the labelled and unlabelled data, and the labels of the labelled data are transferred to the unlabelled data based on the similarities, the basic idea is that similar instances are more likely to have the same activity labels. In [132], the authors present a positive-and-unlabelled semi-supervised approach. The labelled data is regarded as positive class, while the unlabelled data is regarded as negative data to train the initial SVM. During the learning phase, the SVM is used to classify the unlabelled data and select the instances that are properly positive and add them into positive class. This process is repeated until converges. Mahdaviani et al. [97] propose a virtual evidence boosting semi-supervised CRF training method for human activity recognition. In addition to the conditional pseudo-likelihood of labelled, the objective function also takes into account the unlabelled conditional entropy.

One disadvantage of semi-supervised learning is that the confident instances are not informative enough [133, 21], especially for discriminative classifiers which perform classifications based on the boundaries between different classes, as the confident samples are usually far away from the boundaries, and contribute little to the boundaries adjustment. Another problem with semi-supervised approach is that even though many feature vectors have comparable likelihood in a step after classification, it only considers the top-rated class as the label and ignores the others.

Active learning

Active learning is another way of reducing the labelling effort for human activity recognition. However, unlike traditional semi-supervised learning methods that choose the most confident instances along with the predicted activity classes for retraining, active learning usually chooses the most informative instances along with the ground truth for retraining. The basic idea is that the most informative instances usually reside near the classification boundary, and the classifier is uncertain about the predicted activity classes. Therefore, choosing those instances for retraining can quickly converge the classifier [133].

There are many existing works using active learning methods to alleviate the training data collection and the labelling effort in the activity recognition pipeline. For example, in [50], the authors first cluster the unlabelled data into clusters with the dynamic k-means clustering algorithm, then they propose a method of finding the most informative instances by combining the entropy measurement and the similarities between the instances and different clusters. The authors in [121] propose an online activity recognition system that incrementally and actively queries the user for the labels of the activity instances. They use the Growing Neural Gas (GNG) algorithm to select the most informative instances. GNG creates a graph to approximate the distribution of the data, with each node in the graph corresponding a region in the dataset. The basic idea is to query the user for the label of each node, so as to reduce the labelling effort. In [24], the authors explore multiple informativeness measurements in active learning for human activity recognition. They include least confident, minimum margin, maximum entropy. Other works that also simply leverage active learning for activity recognition include [87, 133].

Transfer learning

In semi-supervised and active learning, the activity classes of the labelled data and the unlabelled are in the same domain, while in transfer learning, the labelled data in one domain is used to recognize the activities in another domain, so that the data acquisition in the target domain is not required.

The most paramount works are as follows. Zheng et al. [165] propose to use the knowledge from websites to measure the similarities between the activities in the source domain and those in the target domain. Then the data in the source domain is labelled with the activities in the target domain along with the similarities, and this data is used to train the SVM activity model. The loss function of each instance is further weighted with the similarity so that the dissimilar instance contributes less to the model training. In [142], the authors recognise the activities in the target home setting with activity data in the source home setting. The knowledge transferring is implemented through the *meta-feature space* of the sensors (e.g. microwave sensor in the source domain and stove sensor in the target domain have the common meta-feature "kitchen heating"). By transforming the sensor readings into the *meta-feature space*, they are able to recognize activities in the target environment with only the labelled data from the source environment. The authors in [51] propose to perform knowledge transferring both in the feature space and in the label space. In the feature space, the instances in target domain are labelled with the labels of the source domain based on the similarity of the sensor readings distribution between them. In the label space, the intermediate labels of the instances in the source domain are mapped to the labels in the target domain by considering the semantic difference between them and the temporal consistency.

One of the shortcomings of transfer learning is the fact that it usually sacrifices the recognition performance for minimising labelling effort.

Zero-shot learning

In contrast to transfer learning that recognise activities in the target domain with activity data of the source domain, zero-shot learning recognises new activity classes that have never been observed before. In zero-shot learning, a hidden attribute layer is abstracted from the sensor data, and domain knowledge is leveraged to recognized the new activity classes with data from the hidden attribute layer instead of the raw sensor data.

For example, Cheng et al. [24] first map the raw sensor data into semantic attributes such as *upper arm back*, *upper arm down*, and then define the activity-attribute matrix that shows the presence of each semantic attribute in different activity classes (including the unseen activities). In the testing phase, they first use a trained SVM to classify the sensor readings into semantic attributes, and then recognize the unseen activities by referring to the activity-attribute. In [23], they extend the previous work by incorporating the sequence nature of the activity data with Conditional Random Fields (CRF). Specifically, potential functions are defined between the variables (i.e. activity-attribute, attribute-feature vector, attribute-attribute), and the weights of the potential functions are learned by minimising the loss function on the training data. In [106], the authors propose to recognize new activities with minimum labelled data. As the training data for new activities is limited, they propose to combine feature-based learning with attribute-based learning to overcome the problem of low recall in recognising new activities. Feature-based learning is the traditional method using the features extracted from sensor data for activity recognition, while attribute-based learning is to detect the semantic attributes from the sensor readings and use them for recognizing activities.

Zero-shot learning suffers from the drawback that it is a non-trivial work to define the activity-attribute matrix, especially when the target activity classes are diverse. In addition, classifying the sensor data into different semantic attributes requires additional annotated data for classifiers training.

Unsupervised learning

Without any labelled data, unsupervised learning method is to uncover latent activity patterns from the data. There are many previous studies trying do discover frequent activity patterns from the unlabelled data. For example, Rashidi et al. [119] propose a technique to discover frequent patterns and model them as multiple HMMs. They used the variant edit distance method to measure the similarity among patterns, which is able to deal with discontinuities. The frequency of patterns are measured based on the description length principle. Cook et al. [27] also incorporate this method to discover and label frequent activities from the partially-labelled dataset in order to improve recognition accuracy.

As recognising physical activities such as sitting and standing is primitive, it is more reasonable to cluster daily routines such as working and commuting which can reflect the personal lifestyle. In [54], the author performs Latent Dirichlet Allocation (LDA) on accelerometer readings to cluster daily routines such as commuting, office and lunch. While Sun et al. [136] extend this work and perform clustering with Hierarchical Dirichlet Process (HDP), in which the parameters such as topic number do not have to be specified. They generate words with Dirichlet Process Gaussian Mixture Models (DPGMM) and cluster topic proportions resulted from HDP, which shows superior performance compared to the previous works. The idea underlying LDA and HDP is to explore the whole dataset and cluster the frequently co-occurring (discrete) or closest (continuous) data points into the same topic. It is also based on the observation that the same human routine tends to have the same data distribution (e.g. office routine mostly comprises sitting and has less variation in accelerometer readings), which result in similar topic proportions. In [128], Seiter et al. introduce a topic modelling approach to discover daily routines from sensor data. Unlike traditional topic modelling methods, they incorporate semantic similarity between words during the

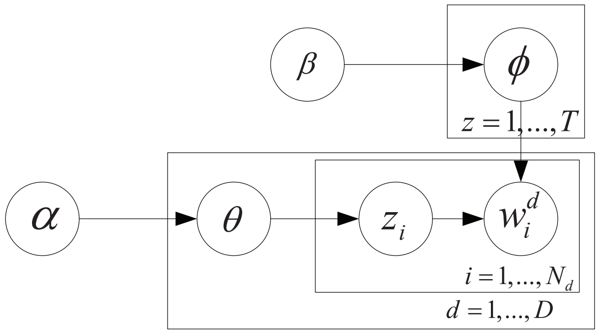


Figure 2.8: Graphical model of LDA

clustering process.

However, small amount of labelled data is required in the unsupervised methods for ground truth provisioning, otherwise the frequent patterns or daily routines discovered by aforementioned unsupervised approaches do not have semantic meaning and cannot self-interpret. For example, dataset in [119, 27] is mostly comprised of sensor data from binary motion sensors, as a result, the method actually creates a map between activity patterns and location. While in [54, 136], the data distributions resulted from accelerometer are meaningless if we do not have predefined routines.

We will be using LDA in our approach as a hybrid with traditional classifiers to create a generic activity model described in the next chapter, we therefore briefly describe LDA here. LDA is a hierarchical Bayesian model, being primarily used for text mining. In LDA, the document is modelled as a multinomial distribution over the latent topics, while the latent topic is modelled as a multinomial distribution over the words. LDA explores the documents and clusters the frequently co-occurring words into the same latent topic.

The graphical representation of LDA is depicted in Figure 2.8, where T is the pre-specified number of topics, and N_d is the number of words in document d. To generate a word, the topic distribution of the corresponding document is sampled from a prior Dirichlet distribution parametrised by α, $\theta_d \sim Dir(\alpha)$. And then the topic assignment z_i of the word is drawn from a multinomial distribution $z_i \sim Multi(\theta_d)$, and the word is generated by sampling $w_i \sim multi(\phi_{z_i})$. Notice that ϕ_{z_i} specifies the word distribution of topic z_i, which is drawn from a prior Dirichlet distribution parametrized by β. Therefore, the likelihood of the words in the corpus is:

$$\mathcal{L}(\alpha,\beta) = \prod_{d=1}^{D} \int\int p(\phi|\beta)p(\theta|\alpha) \prod_{i=1}^{N_d} \sum_{z=1}^{T} p(w_i^d|z,\phi)p(z|\theta)d\theta d\phi \tag{2.3.3}$$

Building the topic model is equivalent to finding the topic-word assignments that can maximise the likelihood. The assignment can be estimated via the collapsed Gibbs sampling, which iteratively samples the topic for each word while fixing the topic assignment of all the others, and then uses the topic assignments to estimate parameters (such as document-topic distribution and topic-word distribution):

$$P(z_i^d = k|\boldsymbol{z}_{-i}, \boldsymbol{w}) \propto (\alpha + n_k^{d,-i})p(w_i^d|\boldsymbol{w}_{k,-i}) \tag{2.3.4}$$

where $\boldsymbol{z}_{-i}$ is the topic assignments in the previous iteration excluding the current word w_i^d, and $n_k^{d,-i}$ is the number of all the other words that are assigned to topic k in document d. $\boldsymbol{w}_{k,-i}$ is the set of words across all the documents which are currently assigned to topic k, excluding w_i^d. The likelihood term $p(w_i^d|\boldsymbol{w}_{k,-i})$ can be computed by finding the proper conjugate prior and marginalizing out the parameters ϕ. Since the topic-word distribution is assumed to have a Dirichlet prior (parametrized by β) and the word is drawn from a multinomial distribution, the predictive likelihood of w_i^d given dataset $\boldsymbol{w}_{k,-i}$ can be obtained by fraction counting [164]:

$$p(w_i^d|\boldsymbol{w}_{k,-i}) = \frac{\beta + n_{kw}^{-i}}{\sum_v(\beta + n_{kv}^{-i})} \tag{2.3.5}$$

where n_{kw}^{-i} denotes the number of the other words in topic k that have the same symbol as w_i^d, while $\sum_v n_{kv}^{-i}$ is the total number of words in topic k excluding w_i^d. Eq.(2.3.5) is the likelihood of word w_i^d being generated by topic k given the current word distribution in topic k. Therefore, the clustering is done by assigning the word to the topic in which it has the maximum likelihood.

Discussion

Data-driven methods are able to achieve high recognition accuracy, as they can learn complex patterns in the data with advanced machine learning approaches. However, they all have their own drawbacks that make them inapplicable for our goal. In supervised methods, large amount of annotated data is required. In semi-supervised and active learning methods, recognition accuracy is sacrificed for alleviating labelling effort. In other methods, such as transfer learning and zero-short learning methods, human knowledge is required to interpret the semantic meaning of sensor data. More importantly, all the aforementioned methods assume pre-defined data sources for activity recognition and are not applicable in our scenario.

2.3.2 Knowledge-driven methods

The knowledge-driven approach leverages domain knowledge for activity modelling and inferring. The underlying observation is that human daily activities contain rich in common-sense knowledge that interlinks activities and surroundings. The knowledge-driven approach is more compelling than the data-driven for several reasons. First, activities in realistic scenarios (i.e, cooking, preparing a drink) may comprise amounts of same physical actions, and the order that the subjects perform the activities may not be consistent all the time. These characteristics of the activities pose a challenge to recognise them solely based on physical signals from sensors such as accelerometers and gyroscopes. However, those activities can be differentiated by taking into account a diversity of the surrounding context. Since most activities usually take place at different time, location and with different object interactions, thus this additional context can be used to better characterise the activities. For example, the activity "brush teeth" usually happens in the bathroom in the morning and at night, and the objects involved are usually toothpaste and toothbrush. Moreover, the knowledge used for activity recognition can be explicitly specified or mined from external information sources, thus avoiding the processes of manual labelling, feature extraction and learning in the data-driven approaches to activity recognition. Previous works also demonstrate that, with carefully defined domain knowledge, it is possible to achieve the equivalent recognition performance to the HMM [125]. In what follows, we discuss two knowledge-driven approaches, one is to mine the knowledge from external information sources (e.g. web) and the other is to explicitly specify the knowledge (e.g. ontology). Their advantages and disadvantages are also discussed.

Mining-based approach

The basic idea behind this approach is that the activity models can be created by mining knowledge from the existing external sources such as websites, which provide the instructions on performing the activities and the objects that are required. Hence, through information retrieval methods, activities are modelled by establishing the relationship between the activities and the required objects in a probabilistic manner. Then given the objects used at a specific time point, which are usually captured through sensors, the probabilities of the activity classes that current activity belongs to are calculated and the one that has the maximum probability is chosen to label the current activity.

Perkowitz et al. [116] propose a method to create an activity model by mining the web. By tagging each word in the sentences with its part of speech, they are able to extract the objects used in the activities. Then they automatically calculate the probabilities of the objects usage in the activities using Google conditional probabilities APIs. As the objects involved in an activity cannot be exhaustively mined from the web, Tapia et al. [139] propose a way to deal with unseen objects. They create a hierarchical ontology of synonymous words for functionally similar objects. By performing shrinkage over the ontology of objects, they calculate the probabilities for the unseen objects in a probabilistic way. Instead of relying on object probabilities for activity recognition, Gu et al. [37] mine the activity descriptive texts from the websites, they then use natural language processing method to extract the objects used in the activities from the texts and information retrieval methods to calculate the weights of each object with respect to different activities. Finally, they construct contrast patterns for each activity based on the object terms and their relevance weights they mined from the web, so as to maximize the discriminative power of fingerprints for each activity class.

Ontology-based approach

Rather than mining objects usage information from the external resources, an ontology-based approach is to explicitly specify the activity models with a description-based method. In [20], the authors model sensors and activities as classes in ontology separately, with each class described by a number of properties. For example, each sensor class has the state property, indicating the state of the object to which the sensor is attached. The activation of the sensor can be interpreted as the object-interaction. Each activity class has the properties such as *hasLocation* and *useArtifact*, denoting the location in which the activity is performed and objects involved in the activity. The aggregation of sensor activations at a specific time point can be used to establish a situation, which is then reasoned against already established activity models. In this light, the sensor ontologies model the situation at a specific time point that interrelates the context information and sensor observations. Activity ontologies interlink the activities and contextual information through object properties, and activity recognition is equivalent to reasoning on a dynamically constructed situation against activity ontologies. The activity ontologies are organized in a hierarchical structure, where subclass inherits all the properties of superclass. As more sensor observations are aggregated at runtime, the recognized activity can be narrowed down from the class hierarchy, as thus, the ontology-based approach is able to recognise both coarse-grained and fine-grained activities.

Riboni et al. [124] even propose to combine ontological reasoning with statistical reasoning. The environment is modelled using ontology, then using domain knowledge, they perform ontological reasoning to infer the possible activities in each location. At run-time, they use statistical reasoning to obtain for each data sample a posterior for each activity class, the possible activities are then filtered out by the previous inferred knowledge from the ontology.

The obvious problem with the ontology-based approach is that the temporal reasoning is not supported. Moreover, it is vulnerable to information uncertainty, due to the fact that all the object properties must be satisfied in order to infer specific activities. Helaoui et al. [45] propose probabilistic ontological approach to recognize multilevel human activities. In particular, they leverage the log-linear description logics (DLs) to integrate DLs with probabilistic log-linear models [108]. They add weighted axioms into the ontology as long as they are consistent with the axioms already in the ontology. There may be several ontologies with different set of axioms that are consistent with each other, only the one that have the maximum a-posterior (MAP) is chosen for reasoning activity. In order to take temporal information into account, they strictly define the order of actions in each activity. This method is robust with data uncertainty in that the axioms are assigned with different weights, the larger the weight the stronger the rule holds.

Discussion

Both of the mining-based and the ontology-based approaches have their own disadvantages. For example, activity models established from the object probabilities mined from the external information sources are general models that do not allow to achieve high recognition performance as people perform activities quite differently. For ontology-based method, even though data uncertainty and temporal reasoning are solved by the probabilistic ontological framework, defining the ontology and specifying the weights of the axioms are non-trivial tasks. Although it is possible to learn the weights from the data, advantage of non-manual-labelling of knowledge-driven approach is affected. Moreover, the explicitly specified order of actions in the ontology disregards the fact that activities can be performed differently by various users, even the same person may perform the same activity in various manners at different time. The goal of our research is not to propose solutions to solve aforementioned problems. On the contrary, we will leverage the domain knowledge as a starting point, and incrementally refine the activity recognition model with opportunistic sensors and increasingly available data.

2.3.3 Hybrid methods

There are some works combining the knowledge-driven methods with data-driven methods for activity recognition. One of the key advantage of the hybrid method is that it is able to achieve high accuracy with limited labelled or even no labelled data. For example, Wang et al. [147] consider actions taken in the activities in addition to the object usage. They use domain knowledge to create a Dynamic Bayesian Network (DBN), and the DBN is able to infer the possible actions given the activity and object. Then, the possible actions are used to label the accelerometer data and those labelled data are used to train boosting classifiers. Notice that the training of the classifiers also takes into account the probability of the action as virtual evidence. Therefore, the inference of the DBN and the training of the boosting classifier works in a joint manner. In [154], the authors also present a Dynamic Bayesian Network that incorporates RFID and vision data to jointly infer the most possible activity sequence and object labels. To do that, they use domain knowledge to specify the parameters of DBN, such as the object-activity probabilities, activity transition probabilities and object transition probabilities. Then, they use the EM method to estimate the vision-object probabilities. Specifically, the E-step estimates the distribution of the object labels given the observations and current parameters, the M-step updates the parameters with the results from the E-step.

2.3.4 Other concerns

There are also many previous works addressing other aspects in human activity recognition. They include robustness, segmentation, personalisation and additional features for activity recognition.

Robustness

Maekawa et al. [95] address the issue of scalability in activity recognition. They employ the information of the end user to find other users whose sensor data may be similar to those of the end user, and model the activities of the end user in an unsupervised way with the data from other similar users. While in [77], the authors leverage the similarity (physical similarity, lifestyle similarity and sensor-data similarity) between different subjects to populate and enlarge training data. Kapitanova et al. [62] address the issue of fault-tolerance. They train multiple models with each model excluding a certain set of sensors. Through observing the performance achieved by different models, it is possible to locate the failed sensors.

Segmentation

Continuous activity recognition usually incurs the problem of data segmentation. Data collected in a small window may not be sufficient to recognize activity, while a large window may result in situation that data from different activity classes collide into the same window. Gu et al. [39] propose Max-Gain which is based on the observation that different feature items weight differently for different activities. This post-segmentation method requires correct recognition of adjacent activities, so that the weight of feature items against the activities can be determined. Okeyo et al. [111] propose window shrinking and expanding methods based on whether sensor events in the current window are enough to describe the most specific activity or additional sensor events are needed to be recruited. This approach has the danger of errors accumulating. Krishnan et al. [72] propose a probabilistic approach to dynamically determine the window size, which, however, requires large amount of training data.

Personalisation

Existing methods usually incorporate sensor data from multiple users to build robust activity models, termed as a general model. Since people perform activities differently due to their physical characteristic, age and gender, the general model may best fit a certain specific user. Actually, as demonstrated in [150], a personalised activity model greatly outperforms a general one. Several previous works propose methods to personalise activity models. For instance, [28] trains general and user-specific classifiers and uses a meta-classifier to determine which classifier is more likely to predict the class correctly. Samples are added into the training dataset if their classification confidences surpass a threshold. In [163], the authors first classify the samples with a Decision tree, and then perform clustering method to re-organise the samples. The parameters (thresholds in branch nodes) are re-estimated with the re-organised samples. While in [123], the personalisation is performed in two phases. In the training phase, the weight of each classifier is measured by its consistency with a set of classifiers in terms of recognition performance. In the testing phase, the probability that a classifier is chosen for prediction is proportional to its weight.

Additional features

Most of the existing activity recognition systems usually employ physical signals or object usage as features, there are also many works considering additional context information to improve the recognition performance. For example, Lara et al. [78] incorporate vital signs into the feature vector. They use an arbitrary function to fit the sensor data, then the parameters of the best fitted function are regarded as features. Due to the delay of vital signs, they also consider the *trend* and *magnitude* of the change of vital signs. While Maekawa et al. [96] try to recognize ADLs (activity of daily living) by employing many kinds of sensors including cameras, microphones and accelerometers. The underlying observation is that images of activity relevant objects and the sound emitted when a user performs an ADL can help to better recognize activities. In [65], the authors present an activity recognition system without the accelerometer. Instead, they use the energy harvesting power signal when the energy is harvested to power the device. The rational of this method is that different activities produce distinguishable energy, and hence have the unique fingerprints in the harvested energy signals. Wang et al. [149] use Wi-Fi signals for activity tracking and recognition. The basic idea activities in different places of the house leave their signatures in the Wi-Fi signals. In [135], Suarez et al. demonstrate that by splitting the accelerometer data into low and high frequency component with a low pass filter, they are able to improve the recognition accuracy significantly.

2.3.5 Activity recognition in specific domains

Most of the previous works recognise physical activities (e.g. standing, sitting) or activities in daily lives (e.g. kitchen activities, activities in smart environment). With the prevalence of various sensors and mobile devices, recently many researchers study the recognition of activity in a specific domain. They are not only recognising the target activities, but also monitoring if the activities are performed in a normal way. The most paramount works are as follows:

- *Tooth Brushing*: In [71], Korpela et al. present a method to evaluate the tooth brushing performing with audio gathered from a smartphone. They first classify different tooth brushing activities with HMMs, and then they use the output (e.g. duration of each tooth brushing activity) of the HMMs to build regression models for the tooth brushing performance scores estimation.
- *Locomotion mode*: Hemminki et al. [46] present a technique for accurate and fine-grained locomotion mode detection with accelerometer data. They first estimate the gravity component of accelerometer data for calculating the gravity eliminated vertical and horizontal acceleration. Then extract multiple features from the acceleration data for building hierarchical classifiers. In [127], the authors propose to use barome-

ter for the detection of transportation mode (e.g. walking, vehicle). The basic idea is that different transportation modes make the pressure change differently.

- *Eating*: In [168], the authors present a system for nutrition monitoring with a smart table cloth. The table cloth is equipped with a weight sensitive tablet and a fine grained pressure textile matrix, so that they are able to spot different actions based on the pressure change when the users are eating on the table cloth. Thomaz et al. [140] implement an approach for detecting eating moments with a 3-axis accelerometer on a smartphone. Other works use different sensing modalities such as neck-attached [159] or ear worn microphones [5] to detect eating related sound for dietary monitoring.

- *Smoking*: Nguyen et al. [105] present an activity recognition for recognising smoking activity. They try to address the problem of activity recognition in open world where an unlabelled instance can belong to any of the possible activities instead of one of the predefined activities. They propose Multi-class Positive and Unlabelled Learning to reduce the false positive in recognizing smoking in open world. Therefore, they merge the unlabelled instances into the negative set so that the negative set can form a representative set of negative instances, and learning with the positive and negative set can result a correct decision boundary. Kawamoto et al. [63] monitor changes in respiratory rate during sleep with wrist-worn accelerometer, and use the data for the detection of multiple physical conditions such as smoking cessation.

- *Sleeping*: Hao et al. [43] present iSleep - an individual's sleep monitoring system using off-the-shelf smartphone. They develop a lightweight Decision-tree-based algorithm to classify the microphone data of the smartphone into multiple sleep related events such as body movement, couch and snore, and use the classification results for evaluating sleep quality. In [40], the authors move a step ahead and detect sleep stages with sensors on smartphone. The basic idea is that different sleep stages are accompanied by different body movements and acoustic signals features. They use linear Conditional Random Field to integrate these feature and make further inference.

- *Swimming*: Bächlin et al. [9] present a swimming monitoring system called *SwimMaster* that monitors swimming performance and technique with acceleration sensors at the wrist and at the back.

- *Activities in hospital*: Bardram et al. [12] propose to detect the progress of the work inside an operating room with embedded sensors and body-worn sensors. In [56], Inoue et al. recognize nursing activities such as *blood pressure measurement* with mobile devices recording the acceleration data. By considering the prior probability of the activities happening in a specific time slot, they are able to outperform conventional classifiers with large margin.

- *Activity recognition meets social network*: While previous works recognize human activities with sensor data, there are some approaches trying to recognise activities with social network data. Zhu et al. [170] recognise activities based on the tweets from social media. They crowd-label the tweets and process them into a feature vector with natural language processing methods, so that they can be fed into conventional classifiers for training and testing. In [29], Du et al. propose to predict the attendance of social activity published on the website. They consider multiple features such as content, spatial and temporal context of the social event as features and use matrix factorisation for attendance prediction. Zheng et al. [167] mine location features and activity-activity correlations from the web and perform matrix factorisation for activity and location recommendation.

2.3.6 Discussion

Most of the aforementioned activity recognition techniques are inapplicable in our scenario, since most of them use static models with predefined data sources, and are not able to adapt with dynamically available sensors that may be potentially beneficial to the recognition accuracy. Even though the knowledge-based method can be used to deal with the unseen contexts provided by the sensors, the parameters specified by the method is general knowledge and cannot achieve satisfactory performance due to the fact that they usually sacrifice the recognition performance for the alleviation of labelling efforts. On the other hand, some researchers personalise the activity model to a specific user for achieving high accuracy. However, they do not consider the dynamically available sensors that are common in realistic scenarios. The lack of related work and the importance of addressing sensor dynamics have motivated us to propose methods for incorporating the sensors dynamically for activity recognition.

2.4 Activity model retraining and adaptation

In this thesis, we first create a generic activity model with limited labelled data, and then perform activity model adaptation with the additional information provided by dynamically available sensors. Therefore, our work is related to traditional semi-supervised activity recognition methods that select the instances classified with high confidence to retrain and adapt the activity model. The most profound semi-supervised activity recognition methods have been reviewed in the previous sections. The shortcoming of these methods is that the high-confidence instances are not informative [133, 21] and do not help to converge the activity model. Moreover, those methods do not consider information provided by dynamically available sensors.

Personalisation of an activity model also leverages the labelled and unlabelled data to adapt the model. The related works are also reviewed in the previous sections. The main problem with those methods is that labelled data of a specific user is required for the activity model to be adapted to his/her activity, and annotating activity data is time-consuming, expensive and error-prone. In other words, the activity models in those works are not able to adapt automatically.

2.5 Sensor dynamics in activity recognition

The state of the art activity recognition models usually rely on a static model, where only pre-defined data sources are considered while opportunistically available contexts that may potentially refine the systems are ignored. Previous works show that more contextual information can further improve the activity recognition accuracy. For example, in [162], the authors demonstrate that additional features such as vision features can help to improve the recognition accuracy for human activities, especially for static activities (e.g. sitting). Maekawa et al. [96] show in their work that, contextual information, such as the objects that the subjects interact with and the sound during the interaction, captured by camera and microphone can help to improve activity recognition performance. Riboni et al. [124] find that *location* context can help to solve the ambiguities of the recognition results that are based solely on wearable sensors. They build an ontology and reason for each location which activities could possibly happen. At runtime, *location* context is incorporated to filter out impossible activity candidates provided by statistical reasoning results based on wearable sensors. In [8], the authors perform localization by considering surrounding context such as *sound*, *color*, *light* and *Wi-Fi AP*. The basic idea is that those contexts in a place can be indications of the type of place, and then from knowing the place we can infer what the users are possibly engaged in (e.g., eating in a restaurant). Extensive works prove that extra information such as vital signs [78], readings from thermal sensor [49] and barometer [127] are also important for activity recognition accuracy.

Even though the aforementioned extra data sources are important for the activity recognition accuracy, they are explicitly embedded in the activity recognition system when creating the activity models. Therefore, these approaches are not able to incorporate the information provided by dynamically available sensors. In addition, all the aforementioned extra data sources are specific to the post-deployment environment. Therefore, considering all the contextual information at the beginning of activity modelling is infeasible, due to the problem of data sparsity and the changes in the environment during post-deployment. Another motivation for our work is that sensors deployed for activity sensing are often broken and updated [92], so it is extremely important that the activity monitoring system can automatically evolve with the changing environment. Our work is inspired by [52], where the authors propose an autonomic context management system which is able to populate dynamically discovered contextual information sources for autonomic context provisioning. However, there are several challenges that need to be addressed in order to achieve an activity recognition framework that is able to incorporate dynamically available sensors. They include creating a generic activity model that serves as the starting point for activity model adaptation (challenge 1 in Section 1.2); adapting activity model with general knowledge or informative instances (challenges 2,3,4); identifying the most discriminative information provided by the dynamically available sensors (challenge 5); exploiting temporal information of human activities in developing the activity recognition framework (challenge 6); and processing sensor readings into proper contexts for recognising activity given the sensor heterogeneity (challenge 7).

There exist some works that leverage external knowledge to create activity models in an unsupervised manner. Even though those works are not focusing on incorporating the dynamically available sensors for activity recognition, the knowledge-driven methods can be used to specify the parameters of the contexts provided by new sensors. For example, Gu et al. [37] interpret the relations between the activities and context (e.g. object usage) by mining the external web pages with natural language processing and statistical methods. Tapia et al. [139] compute the conditional probability of an unseen object given the activities by linearly combining the conditional probabilities of existing similar objects. The similarities are measured through WordNet. In [147], the authors perform activity recognition based on the object usage and human actions. With no labels for the action data, they use common sense knowledge to build an activity model by jointly training Dynamic Bayesian Network and Virtual AdaBoost. They leverage common sense and Dynamic Bayesian Network (DBN) to derive the most likely sequence for the accelerometer data. The sequence together with the accelerometer data is then fed to VirtualBoost to learn the action model, which in turn is combined with DBN to recognize activity. In some other works such as [142], the authors leverage domain knowledge to transfer activity models from one domain to the others, so that the contextual information in other domains (e.g. different smart houses) can be used for activity recognition. These approaches use domain knowledge or external knowledge to interpret the relations between the contextual information and activities, so they are actually creating general models for activity recognition. As people perform activities quite differently [152, 168], those general models cannot obtain accurate recognition results in realistic environments. Moreover, they are not applicable in the situation that we have no prior knowledge about dynamically discovered data sources (e.g. continuous sensor readings of accelerometers).

Other researchers perform activity recognition with dynamic sensor selection or information fusion. For example, in [60], the authors generate multiple processing plans for the context to be monitored. The system dynamically updates the processing plans when sensors are newly registered or de-registered. The logical processing plans represent a set of processing modules (i.e. feature extraction, classification modules) to derive the context while the physical processing plans associate the logical processing plans with different sensors and computing sources. Specifically, their system tries to achieve a desired classification accuracy while prolonging the system lifetime by minimising the number of activated sensors. In another work, Zappi et al. [161] introduce a scheme to dynamically select the sensor set for activity recognition in order to achieve the trade-off between accuracy and power. In [35], the authors propose an energy-efficient activity recognition based on prediction. They use the current and historical context to predict possible future activities, and only a subset of the sensors are activated to distinguish those activities that are likely to happen. While Yan et al. [158] dynamically adjust the sampling rate and classification features in real time to balance the trade off between accuracy and energy consumption. The idea behind is that, to obtain the same accuracy, distinct activities require different sampling rates and features. Since those work mainly focus on the aspect of energy-efficiency, they train each activity with all the available sensors, so that when the sensors are registered at runtime, the system already has the knowledge of how to post-process the sensor data, hence this limits the scalability of the system. For example, in [35] the authors have to calculate the recognition losses of all the possible combinations of sensors at the training time, so that they are able to select the optimal set of sensors that save the most energy and meet the recognition accuracy requirement for any predicted activities.

CHAPTER 3

Creating Generic Activity Model with Minimum Labelled Data

3.1 Motivation

The primary goal of this thesis is to develop activity recognition frameworks that are able to incorporate sensor data from dynamically available sensors for low- and high-level activity recognition. For low-level activities, we propose to create a generic model for recognising low-level physical activities (e.g. walking, running) with currently available sensors, so that the created activity model can be further adapted with newly coming sensors. The underlying reason is twofold. First, the sensor readings (e.g. accelerometer, gyroscope) triggered by low-level human activities are not semantically interpretable, so the machine learning methods are more suitable to create the generic activity model by mapping the sensor data into target activity classes. Second, the generic activity model should be learned with activity data of various users so that it can deal with variants of activity patterns. However, annotating huge amount of activity data to create a generic model is expensive, time-consuming and error-prone. Therefore, we aim to create a generic activity model with minimum labelled data while maintaining a satisfactory accuracy. Notice that, there is a tradeoff between the amount of labelled data and the activity recognition accuracy, so the concept of minimum annotation in this chapter is that for obtaining the same recognition accuracy we require less labelled data (or with the same amount of labelled data, we can achieve higher recognition accuracy) compared with the baseline methods.

There exist some works that employ unsupervised methods [166, 95] to create an activity model for a specific user with labelled data of other users that present similar physical characteristics. However, the activity data of others needs to be annotated. Moreover, there is no significant difference between selecting users based on their physical similarity and random selection, since the physical characteristics and the activity patterns may not have correlation [123].

The general activity model created in this chapter addresses challenge 1 listed in Section 1.2. In addition, the creation of the generic model leverages the temporal information of the activity data and hence addresses challenge 6 in Section 1.2. The key contributions in this chapter are summarised as follows:

- We demonstrate with public datasets that people perform activities differently. The demonstration is illustrated with pairwise training and testing that trains the activity model with one user's activity data and tests it on another user.
- We combine LDA with conventional classifiers to create the generic activity model. LDA is known to be effective in collaborative learning, as it is able to leverage the partially labelled data from multiple users to overcome the problem of data sparsity. Since LDA cannot be applied directly to the activity data, we combine it with traditional classifiers to perform collaborative learning.
- We exploit the temporal information in the human activities during the topic assignment process of LDA. The topic assignment of one instance takes into account the topic assignments of the temporally neighbouring instances.

3.2 Latent Dirichlet Allocation

We describe the motivation of using LDA for creating a generic activity model and analyse why it cannot be applied directly to activity data. LDA models the likelihood of the words in the documents by introducing a latent layer of topics. In LDA, each document is assumed to be a multinomial distribution over the topics, and each topic is modelled as a multinomial distribution over the vocabulary. By maximising the likelihood, the words are assigned to proper topics. LDA is detailed in Section 2.3.1.

LDA is a Bayesian framework, and it has demonstrated its effectiveness in unsupervised and collaborative learning, such as activity discovery [25], mobile context discovery [164], frequent human routines discovery [136, 107] and collaborative learning [166]. The primary motivations of using LDA in our work is to collaboratively leverage limited labelled data from multiple users to overcome the problem of data sparsity and learn a generic activity model. As we can see from Eq.(2.3.4), document-topic distribution may be different across the documents, while the estimation of topic-word distribution makes use of the words across the documents and the resulted parameters are globally shared. This characteristic makes it reasonable to leverage LDA to create a generic activity model, as each user may have different proportions of activity data, while the estimation of the parameters for each activity class utilizes the corresponding data from all the users, and the resulted parameters are globally shared among the users. Therefore, the latent topics are mapped to human activities, while the words are viewed as feature vectors extracted from sensor data. The activity data of each user forms an individual document and the data of all the users comprises the corpus.

However, LDA cannot be directly applied to activity data. Since the instances are multi-dimensional feature vectors extracted from continuous sensor data, it is rare that two instances are exactly the same even if they belong to the same activity class. Therefore, it is infeasible to assume each activity class is multinomially distributed over the feature vectors. A potential solution is to assume the Gaussian distribution over the instances that belong to the same activity class [107]. However, as instances from on-body sensors usually consist of high-dimensional features, the estimation of a large number of parameters would cause the problem of overfitting [136]. For example, given 561-dimensional instances, we need to estimate a 561-dimensional mean vector and a 561×561-dimensional covariance matrix for each Gaussian component. In the next section, we introduce our method using conventional classifiers to estimate the predictive likelihood in Eq.(2.3.5).

3.3 Conventional classifiers for posterior probability estimation

Since LDA cannot be applied to the activity data directly, we propose to combine it with conventional classifiers, AdaBoost, Decision Tree, RandomForest. AdaBoost is detailed in Section 2.3.1, so we briefly introduce some of its notions along with those of Decision Tree and RandomForest. We also describe how to estimate their posterior probabilities that are important to the hybrid approach.

3.3.1 AdaBoost

AdaBoost is a performance boosting classifier. The essence of AdaBoost is to train an ensemble of weak classifiers and combine them to form a more robust and accurate classifier. Each weak classifier makes decisions based on a single feature and needs only be slightly better than random guessing. The final classifier is a linear combination of the weak classifiers, with each classifier weighted by the error it makes during the training process; more weight is given to the classifier that makes less false predictions. AdaBoost is also able to select the most discriminative features to perform classification, thus avoids the problem of feature redundancy.

Suppose the learning process trains an ensemble of weak classifiers for each activity class k:

$$H^k(x_i) = \sum_{t=1}^{T} \alpha_t^k h_t^k(x_i) \tag{3.3.1}$$

where the T is the total number of weak classifiers for class k, while $h_t^k(x_i) \in \{+1, -1\}$ is the tth weak classifier and α_t^k is the corresponding weight. $H^k(x)$ can be viewed as computing a score for feature vector x_i against class k, and the classification is performed by returning the activity class that has the maximum score:

$$y_i = argmax_k(H^k(x_i)) \tag{3.3.2}$$

The posterior probability of AdaBoost can be approximated using softmax function [82]:

$$P(y_i = k|x_i) = \frac{e^{H^k(x_i)}}{\sum_k e^{H^k(x_i)}} \tag{3.3.3}$$

$P(y_i = k|x_i)$ is thus regarded as the posterior probability that instance x_i belongs to activity class k.

3.3.2 Decision tree

A Decision tree is a flowchart-like structure where each leaf represents a target activity class, and each inner node represents a classification on a specific feature and the branches following that node represent the outcome of the classification; all the instances that satisfy the condition on a branch go through that branch to the next node for the further classification.

Training the Decision tree is to find the feature to split the training data so that the difference in the information entropy before and after the split is maximum. The information entropy of a training data is defined as follows:

$$H = -\sum_k \frac{|\{x_i|y_i = k\}|}{|T|} log_2 \frac{|\{x_i|y_i = k\}|}{|T|} \tag{3.3.4}$$

where T is the set of training data, and $\{x_i|y_i = k\}$ is the set of instances that belong to activity class k. Let $vals(j)$ be the set of all possible values of the jth feature, then the information gain of splitting the data on jth feature can be defined as follows:

$$IG(T, j) = H(T) - \sum_{v \in vals(j)} \frac{|\{x_i \in T|x_{i,j} = v\}|}{|T|} H(\{x_i \in T|x_{i,j} = v\}) \tag{3.3.5}$$

where $\{x_i \in T|x_{i,j} = v\}$ is one of the sub-datasets resulting from the splitting, the jth feature of all the instances in the sub-dataset take on the value v.

The training process iterates the splitting until all the instances at the leaf nodes belong to the same class. In prediction, an instance is classified by walking down from the root to the leaf node and is assigned the label of the training instances at that leaf node. In practise, a pruning technique is usually employed when the tree reaches the maximum depth to avoid overfitting. Therefore, an instance is assigned the majority of the class label at the leaf node it resides, and the posterior probability of an instance with respect to a class k can be approximated with the fraction of the class labels at the leaf node [75]:

$$P(y_i = k|x_i, leafnode = j) = \frac{|\{x_i|y_i = k, leafnode = j\}|}{\sum_k |\{x_i|y_i = k, leafnode = j\}|} \tag{3.3.6}$$

Decision tree is one of the most widely used classifiers [129], and one advantage of Decision tree is that it is able to generate decision rules that are understandable and interpretable.

3.3.3 Random forest

Random forest is an ensemble learning method that trains multiple Decision trees at the learning phase and outputs the class that is the majority predictions of those individual Decision trees. Random forest aims to overcome the problem of overfitting in Decision tree. An individual Decision tree may be sensitive to the noisy training data, the average of the multiple uncorrelated Decision trees is not. Therefore, the training of Random forest usually employs the bagging algorithm to de-correlate the ensemble of Decision trees. Algorithm 2 presents the bagging process of Random forest training.

Algorithm 2 Random forest.

```
Input:
    Set of training data: T = {(x_1, y_1), ··· , (x_n, y_n)}
    Max iterations: B;
    Number of sampled instance in each iteration: m;
Output:
    Ensemble of Decision trees
for b = 1 to B do
    Sample m instances from T; call these T_b.
    Train a Decision tree f_b on T_b.
end for
return f = mode(f_1(x), f_2(x), ··· , f_B(x));
```

Another bagging method is feature bagging that randomly selects a subset of the features for training a Decision tree in each iterations. The rational of feature bagging is that in the original bagging algorithm, if one or a few features have strong discriminative power, they will be selected for constructing the Decision tree multiple times, causing the ensemble of trees to be correlated.

The posterior probability of an instance given by Random forest is the average over the posterior probabilities given by the ensemble of Decision trees.

$$P(y_i = k|x_i) = \frac{1}{B}\sum_{b} P_b(y_i = k|x_i) \tag{3.3.7}$$

where $P_b(y_i = k|x_i)$ is the posterior probability given the bth Decision tree as described in previous subsection.

3.3.4 Virtual evidence

We address scalability of aforementioned classifiers by introducing the concept of "virtual evidence". Take training data $T = \{(x_1, y_1, p_1), ((x_2, y_2, p_2)), \cdots, (x_n, y_n, p_n)\}$ for example, each instance (x_i, y_i, p_i) in the dataset is associated with a confidence value p_i, denoting the probability of the instance x_i belonging to the corresponding activity class y_i. In our case, the confidence value is the classified confidence (posterior probability) for each instance. The advantage of embedding the soft assignment in the training process has been demonstrated in the previous works [54, 122, 147, 165]. Training the classifiers is equivalent to minimising the loss function on the training dataset, and the loss function is usually the summed error made by each instance. By multiplying the error of each instance by the corresponding confidence value [165, 122], the less confident instance contributes less to the objective function.

It is easy to incorporate the virtual evidence into the AdaBoost training, as it explicitly calculates the training error in the learning process. For Decision tree and Random forest, we incorporate the confidence value into the calculation of the information gain:

$$H = -\sum_{k} \frac{\sum\{p_i|x_i \in T, y_i = k\}}{\sum\{p_i|x_i \in T\}} log_2 \frac{\sum\{p_i|x_i \in T, y_i = k\}}{\sum\{p_i|x_i \in T\}} \tag{3.3.8}$$

$$IG(T, j) = H(T) - \sum_{v \in vals(j)} \frac{\sum\{p_i|x_i \in T, x_{i,j} = v\}}{\sum\{p_i|x_i \in T\}} H(\{x_i \in T|x_{i,j} = v\}) \tag{3.3.9}$$

3.4 Creating generic model

In the previous section, we described how to approximate the posterior probability of a classified instance using conventional classifiers. In this section, we present the hybrid of LDA and traditional classifiers for generic activity learning. In particular, we demonstrate the feasibility of approximating the predictive likelihood in Eq.(2.3.5) with the posterior probability of the conventional classifiers.

In LDA, due to the underlying assumption of multinomial distribution and the conjugate prior distribution, the likelihood of a word w_i^d in a topic k $p(w_i^d|\boldsymbol{w}_{k,-i})$ is proportional to the fraction of the word with the same symbol in the topic, and it can be estimated from the words currently assigned to topic k. Similarly, the probability that a multi-dimensional instance belongs to an activity class can also be estimated through the instances that are currently assigned to that class:

$$p(x_i|\boldsymbol{x}_k) = p(x_i|y_i = k) = \frac{p(y_i = k|x_i)p(x_i)}{p(y_i = k)} \propto p(y_i = k|x_i) \tag{3.4.1}$$

where $\boldsymbol{x}_k$ are the instances that are currently assigned to activity class k, $p(x_i)$ is a constant for different activity classes and $p(y_i = k)$ is assumed to be equal for different k by class balancing when training the traditional classifiers. In this way, the predictive likelihood in Eq.(2.3.5) can be approximated with the posterior probability of the classifiers as introduced in Section.3.3. In particular, as in Eq.(2.3.5) the instance is assigned to a topic in which it has the maximum fraction, while an activity instance should be assigned to the latent activity against which it has the maximum posterior probability. From another different point of view, training the conventional classifiers is to minimise the training error and minimise the posterior probabilities of the ground truth labels, and this is equivalent to maximising the likelihood of the instances in Eq.2.3.3.

We also exploit the temporal information when sampling the latent activity for the instances, as temporarily adjacent instances tend to have the same activity label. Therefore, we need to consider the topic assignments of neighbouring instances when sampling the topic for current instance, formulated as follows:

$$P(x_i|\boldsymbol{x}_k) \propto \frac{P(y_i = k|x_i)\prod_{j \in N(i) \setminus i} P(y_j = k|x_j)}{Z} \tag{3.4.2}$$

where $N(i)$ indicates the neighbouring instances of x_i and Z is the normalization function.

By combing Eq.(2.3.4) and Eq.(3.4.2), the topic sampling for an instance can be formulated as:

$$P(x_i^d = k | \boldsymbol{z}_{-i}, \boldsymbol{x}) \propto (\alpha + n_k^{d,-i}) P(x_i | \boldsymbol{x}_k) \tag{3.4.3}$$

Algorithm 3 Hybrid generic activity learning.

Input:
Labelled dataset from the different users L
Unlabelled dataset U
Convergence criteria σ
Output:
generic activity model

1: Create initial model: train traditional classifiers (i.e. AdaBoost, Decision tree, Random forest) with labelled data
2: Classify the unlabelled data with the classifiers and obtain for each instance a posterior probability for each activity class, $P(y_i = k | x_i), k = 1 \cdots K$. ($K$ is the number of activity classes)
3: **while** not converged **do**
4:     //e step:
5:     Sample the topic(latent activity) for each instance with Eq.(3.4.3)
6:     //m step:
7:     Retrain the classifiers with labelled data and currently soft topic assignments of the instances given by previous step.
8:     Classify the unlabelled instances using retrained classifiers, and obtain the posterior probabilities for each instance
9: **end while**
10: **return** the trained classifiers;

The algorithm of creating a generic activity model from labelled and unlabelled data is presented in Algorithm 3, which follows an iterative Expectation-Maximization process. At E step, we sample the topic for each instance (line 5), and obtain the predictive likelihood that it belongs to each topic. At M step, these predictive likelihoods are viewed as "virtual evidences" and used to train the classifiers (line 7). The reason of using "virtual evidence" is that it is robust to noise and misassignments of the topics [54]. Initially, when we are uncertain about the latent topic of an instance, the contribution it makes to the training error is further weighted by the "virtual evidence". As the iterative process proceeds, the model is able to confidently estimate the labels corresponding to the instances, then the "virtual evidence" approximates the real assignment and the EM process (line 7) results in a more accurate classifier [110].

3.5 Evaluation

3.5.1 Datasets

In order to evaluate the proposed methods, we experiment with datasets that contain activity data from multiple users. The Smartphone activity dataset [130], UCI HAR dataset [6] and Heterogeneity Dataset for Human Activity Recognition (HHAR) [134] meet the requirement.

Smartphone Activity Dataset (SAD) [130]: Activity data is collected from accelerometer, gyroscope and magnetometer on an Android device worn in different body position (arm, belt, waist and pocket), when the 10 subjects perform 7 activities. We compute time domain features such as mean, standard deviation, median, zero crossing rate, variance, root mean square for each axis of the sensors with a 2-second sliding window and 50% overlapped. Since magnetometer is demonstrated to be less discriminative in their work [130], we only focus on features from accelerometer and gyroscope.

UCI HAR dataset [6]: The dataset is collected with accelerometer and gyroscope from a Samsung Galaxy SII smartphone worn by 30 volunteers within an age bracket of 19-48 years. The smartphone is fixed at the waist when the subjects perform 6 activities. They compute 561 features based on the sliding window of 2.56 second and 50% overlapped. The sensor data was collected at the 50Hz and manually labelled.

Heterogeneity Dataset for Human Activity Recognition (HHAR) [134]: The activity data is collected from on-board accelerometer and gyroscopes on 8 smartphones and 2 smartwatches worn by 9 subjects performing six activities. We choose a smartphone data for each type of smartphone and segment the data with a 2-second sliding window and 50% overlapping.

The summaries of the datasets are presented in Table.5.2. Notice that in SAD, data is collected from 5 body positions, resulting in 5 separate datasets (i.e. SAD-ARM, SAD-BELT, SAD-POCKET, SAD-WRIST, SAD-) with each of them having the same activity classes and instances. In HHAR, the data is split into different sensors and different devices (i.e. acc-nexus4, acc-s3, acc-s3mini, acc-samsungold, gyro-nexus4, gyro-s3, gyro-s3mini), for exam-

Table 3.1: Dataset description.

Datasets	Users	Activities (Instances)
SAD	10	walking (8950), standing (8950), jogging (8950), sitting (8950), biking (8950), upstairs (8950), downstairs (8900)
UCI	30	walking (1722), upstairs (1544), downstairs (1406), sitting (1777), standing (1906), lying (1944)
HHAR	9	Biking (17650), Sitting (19169), Standing (17751), Walking (20385), Stair up (16905), Stair down (15199)

ple, acc-nexus4 stands for the dataset collected from the accelerometer on device nexus4.

3.5.2 Pre-analysis

To validate the motivation of creating a generic activity model, we examine the differences in performing activities among the users by training the activity model with data from one person and testing it on another. We record the f1-score ($f1 - score = \frac{2*precision*recall}{precision+recall}$) and present the cumulative distribution function and histogram of the f1-score for each dataset in Figure 3.1 and Figure 3.2 respectively. Figure 3.2 shows that in most cases, the activity model achieves a low f1-score (0.2-0.8), if it is trained on one person and used to test the data from another. It can be seen from Figure 3.1 that 97% of the tests obtain the f1-score under 80%. This experiment demonstrates that, people perform the activities quite differently, and the model trained on an individual person cannot be scaled to other people who have different activity patterns. To deal with the problem, we need to create a generic model.

3.5.3 Comparison

In this section, we compare the proposed method that creates the generic activity model with the following methods:

- *Semi-supervised method*: traditional classifiers that are trained with the initial labelled data, and used to classify the unlabelled instances, then the instances classified with high confidences are selected to retrain the classifiers. For example, the hybrid of LDA and AdaBoost is compared with semi-supervised AdaBoost.

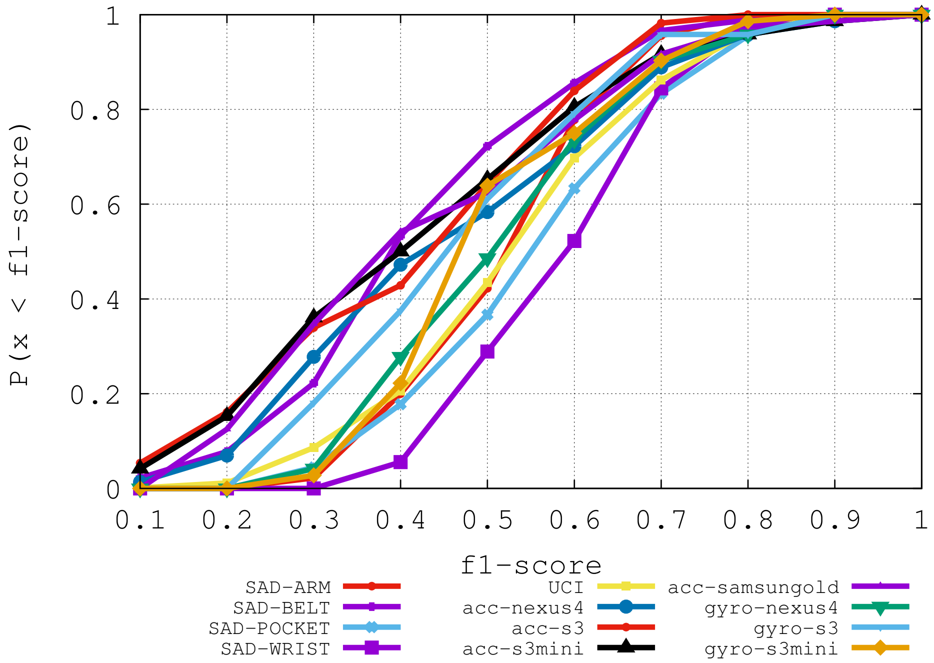


Figure 3.1: Cumulative distribution function (CDF) of the f1-score across the datasets when performing pairwise training and testing.

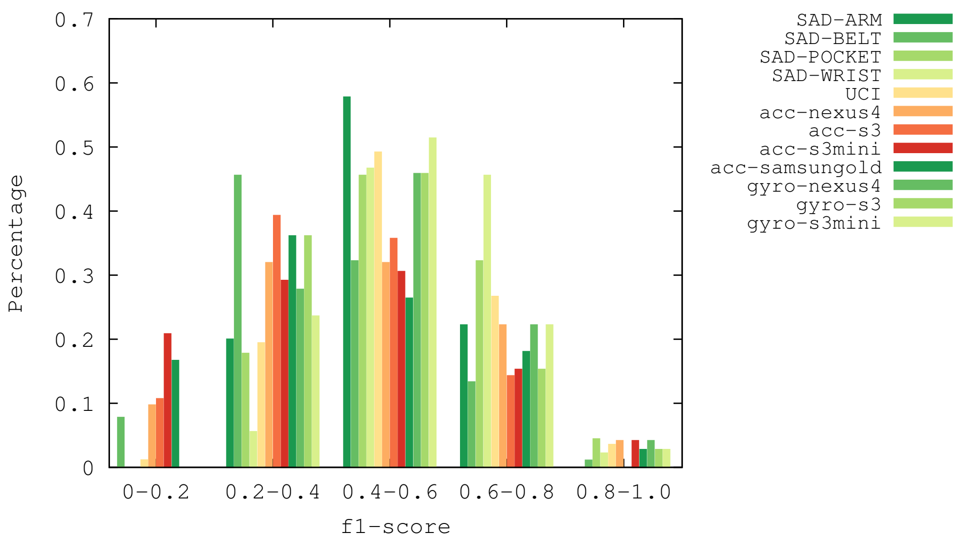


Figure 3.2: Histogram of the f1-score across the datasets when performing pairwise train and test.

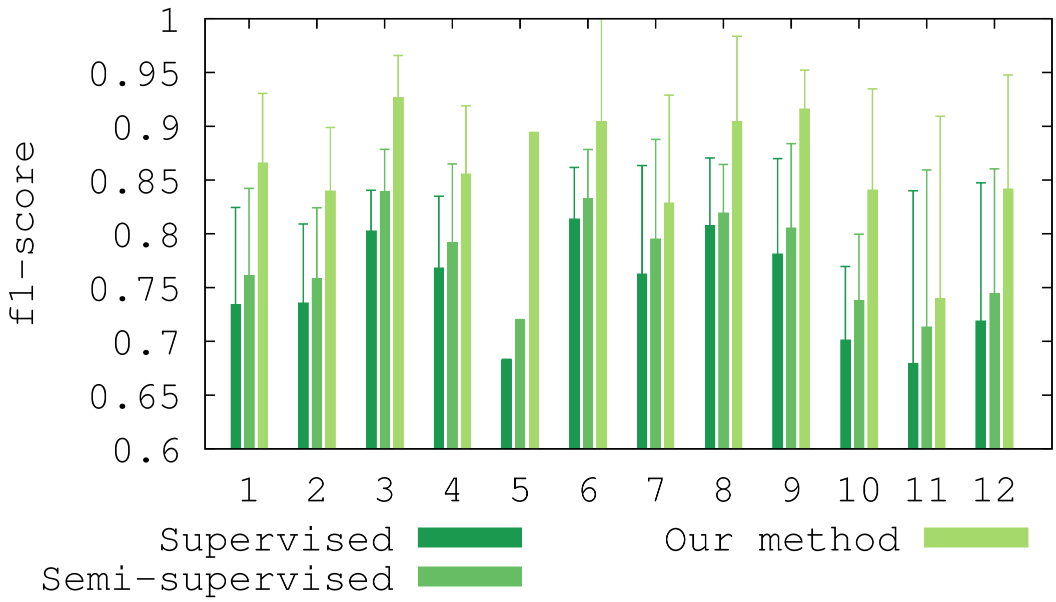


Figure 3.3: Comparing the hybrid of LDA and AdaBoost with baselines.

- *Supervised method*: traditional classifiers that are only trained with the initial labelled data. For example, the hybrid of LDA and AdaBoost is compared with supervised AdaBoost.

Since the amount of unlabelled data selected for retraining in a semi-supervised method is a free parameter, we vary the parameter and choose the best result of the semi-supervised method for comparison, which means we always compare our method with the best fine-tuned semi-supervised methods. We perform leave-one(user)-out validation and present the average f1-score along with the standard deviation across all the subjects for each dataset. To study the effect of the amount of initial labelled data, we vary the percentage of labelled data from 1% to 9%. Notice that those percentages of labelled data are randomly sampled to avoid bias.

The comparison results are presented in Figure 3.3, Figure 3.4 and Figure 3.5, where the x-axis stands for the dataset (1. SAD-ARM, 2. SAD-BELT, 3. SAD-POCKET, 4. SAD-WRIST, 5. UCI, 6. acc-nexus4, 7. acc-s3, 8. acc-s3mini, 9. acc-samsungold, 10. gyro-nexus, 11. gyro-s3, 12. gyro-s3mini). We first discuss the results for the labelled percentage set to 1%. The figures show that the proposed method is able to create a robust generic model, even if the labelled data is limited. Specifically, the hybrid of LDA and Adaboost is able to achieve an average f1-score 11.4% (max: 21.1%, min: 6.0%) and 8.7% (max: 17.4%, min: 2.8%) higher than the corresponding *Supervised* and *Semi-supervised* baselines, respectively. The average

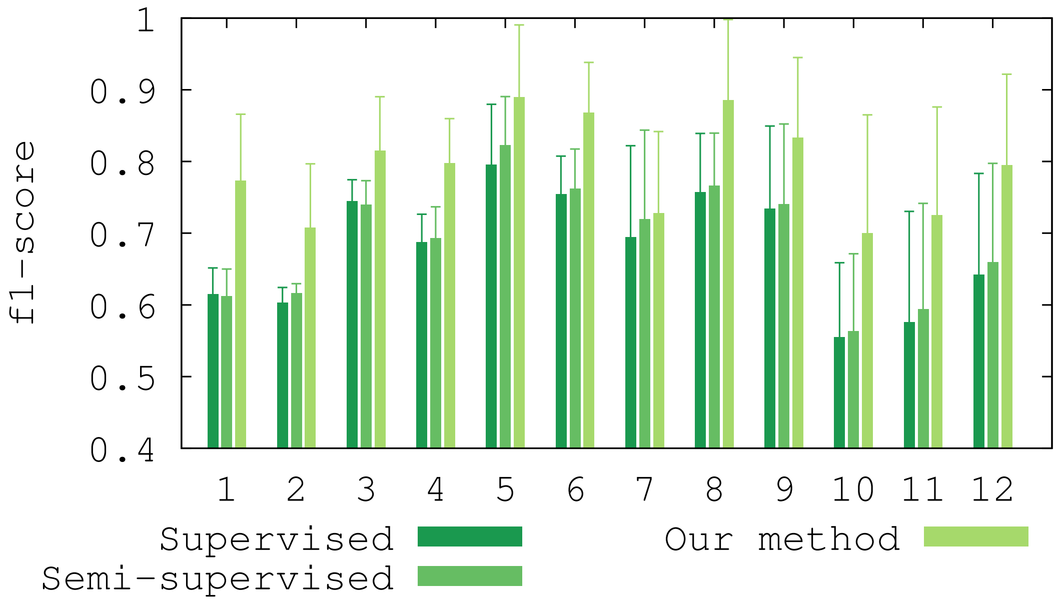


Figure 3.4: Comparing the hybrid of LDA and Decision tree with baselines.

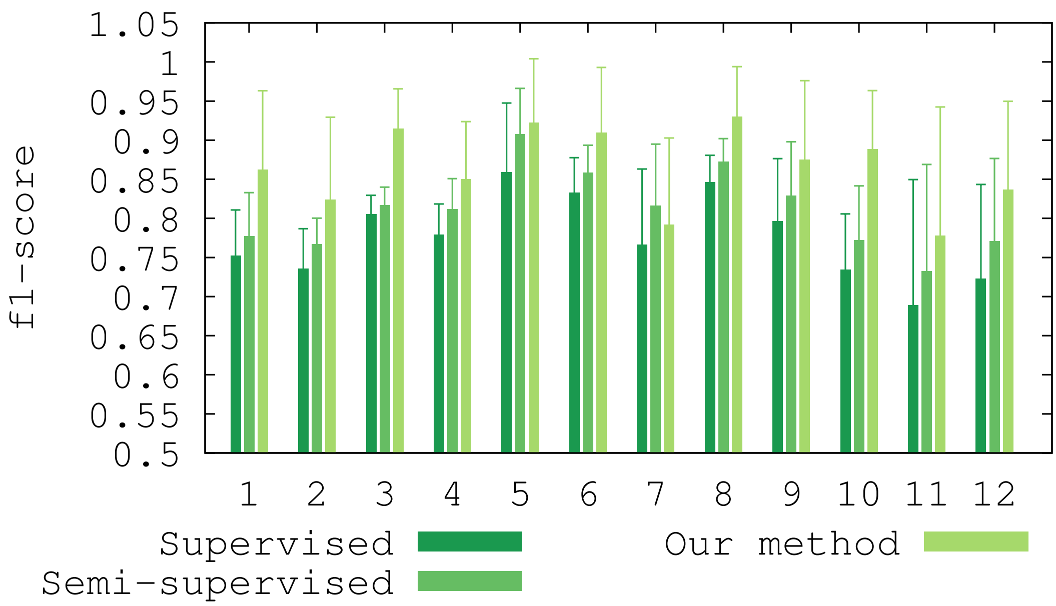


Figure 3.5: Comparing the hybrid of LDA and Random forest with baselines.

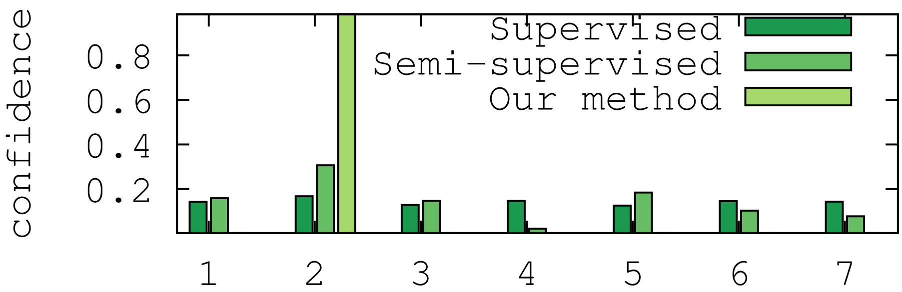
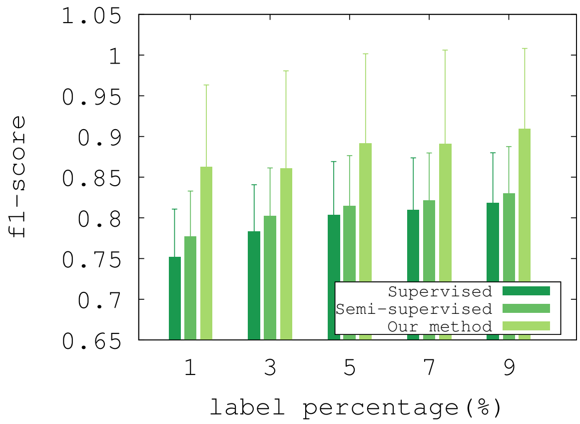


Figure 3.6: The posterior distribution of an instance when the iterative process converges.

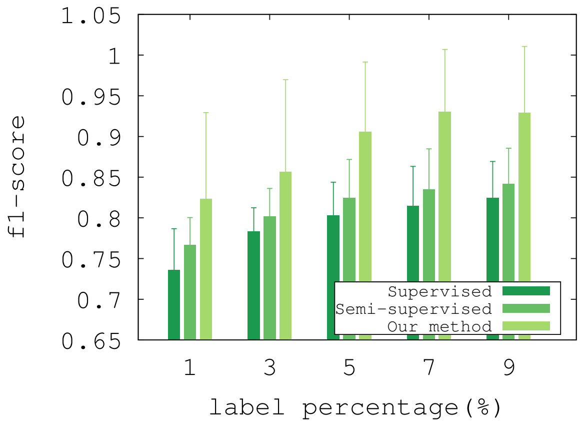
f1-score improvement of LDA+Decision tree over the two baselines are 11.3% (max: 15.8%, min: 3.4%) and 10.2% (max: 16.1%, min: 0.8%), and the average f1-score improvement of LDA+Random forest over the two baselines are 8.9% (max: 15.4%, min: 2.5%) and 5.4% (max: 11.6%, min: -2.5%).

The underlying reason for the advantage of our method can be found in Figure 3.6 (the x-axis represents different activity classes), which shows the posterior distribution of a typical unlabelled *walking* instance before and after the iterative process both in our method and *Semi-supervised*. When the instance is originally classified by *Supervised*, the posterior distribution is rather "flat", which means it is quite uncertain about the true label of the instance. This is because the classifier is trained with limited labelled data from various users, and hence contains much uncertainty when it makes predictions. However, after the iterative EM steps of sampling and retraining, the confidence (0.98) of the instance being activity *walking* approximates to 1 and retraining AdaBoost with this virtual evidence is equivalent to retraining with the true label. As for *Semi-supervised*, the maximum posterior probability (0.306) is still not significant when compared with others, and then retraining with these low confident instances results in less accurate AdaBoost.

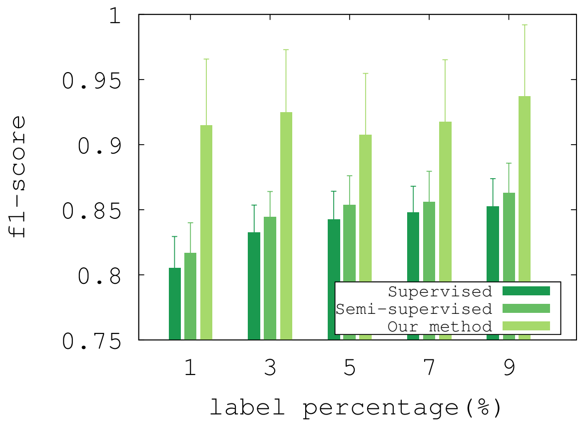
From the figures we can also find that the standard deviation of the f1-score across different datasets can be as high as 11%. Inspecting the classification report in detail, we find that certain activities of some users are totally misclassified. The underlying reason is that some people perform certain activities significantly differently from the others. As we perform leave-one-out validation, those activity patterns of the testing subjects are never presented in the training dataset and are frequently mis-recognised during the testing phase, and this problem is exacerbated when we assume that neighbouring instances are related. The reason can also be interpreted as the training data does not experience enough variants of the



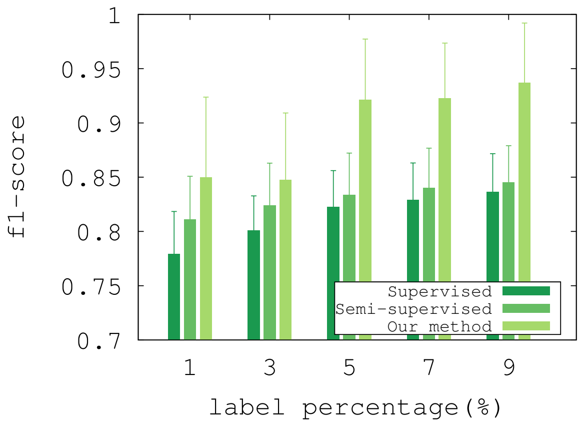
(a) SAD-ARM



(b) SAD-BELT



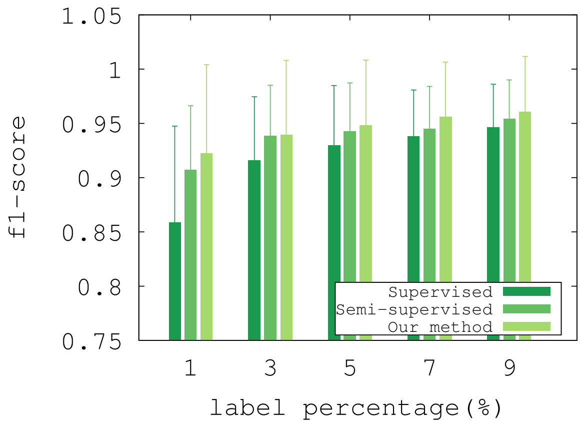
(c) SAD-POCKET



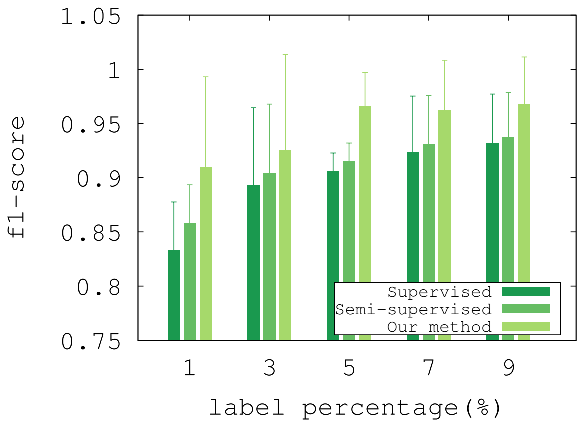
(d) SAD-WRIST

activity patterns to create a generic model. The extreme example can be found when we use LDA+Random forest to classify the dataset acc-s3 (Figure 3.5). Our method experiences performance decrement of 2.5%.

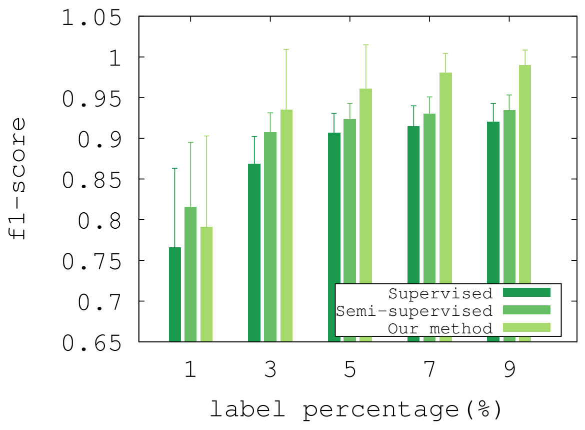
To validate our assumption, we increase the percentage of labelled data from 1% to 9%. We only present the result of LDA+Random forest, as the other hybrid methods have the similar trend. The results are presented in Figure 3.7a-Figure 3.7l. One can observe from the experiment on some datasets (e.g. Figure 3.7g) that the standard deviation indeed decreases when we increase the amount of labelled data, demonstrating that including more labelled data enable the activity model to be able to deal with more variants of the activity patterns. As for some other datasets, the f1-score difference among the subjects is still large even though we increase the labelled data to 9%. The reason is that the activity data from the other subjects (except the one used for validation) is not diverse enough to create a generic activity model, and adding the activity data of the user used for validation is able to boost the recognition performance [152].



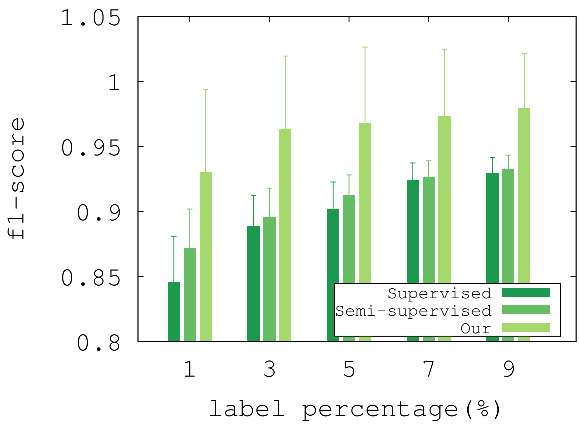
(e) UCI



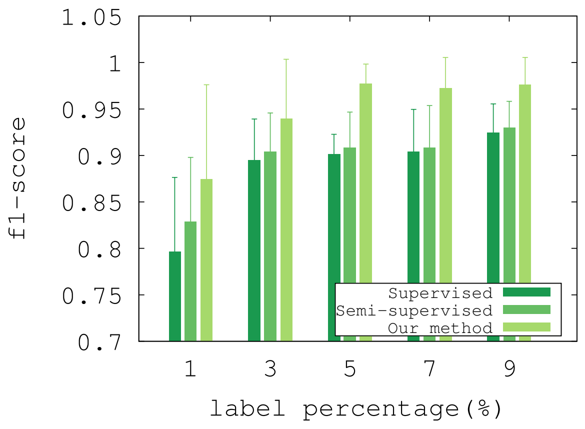
(f) acc-nexus4



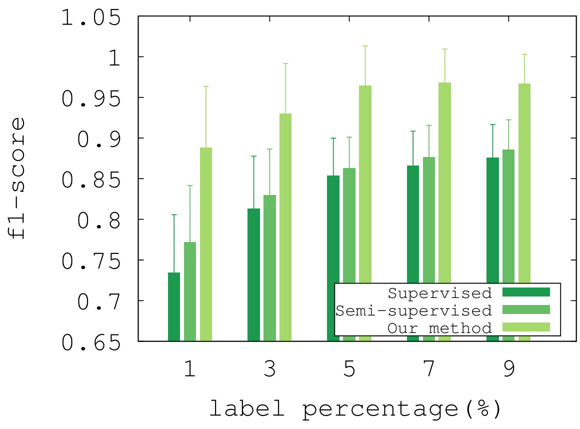
(g) acc-s3



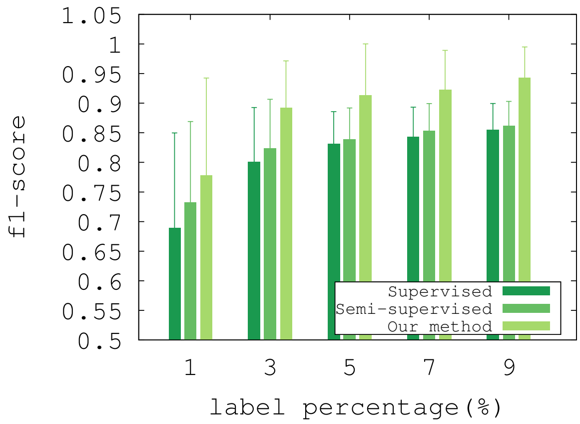
(h) acc-s3mini



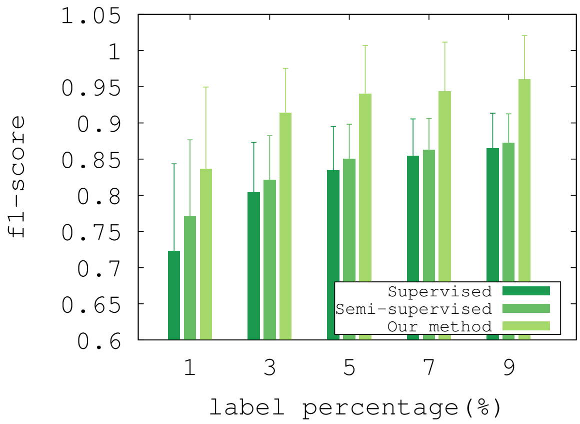
(i) acc-samsungold



(j) gyro-nexus4



(k) gyro-s3



(l) gyro-s3mini

Figure 3.7: f1-score with 1%-9% labelled data across datasets.

3.6 Summary

In this chapter, we collaboratively create a generic activity model with partially labelled data by combining LDA and conventional classifiers. Since the sensor data is not semantically interpretable, we employ a machine learning method to create the generic activity model that maps the sensor readings to target activity classes. To alleviate the data annotation effort, we leverage LDA to collaboratively create the generic model with partially labelled data of various users. We combine traditional classifiers with LDA as it cannot be applied to activity data directly. The proposed generic activity modelling in this chapter addresses challenge 1 in Section 1.2, and the topic assignment in Section 3.4 addresses challenge 6 in Section 1.2.

The generic activity model is created with currently available sensor data, and it serves as the starting point for further activity model adaptation when new sensors become available. In later chapters, we consider activity adaptation and refinement by incorporating dynamically available data sources.

CHAPTER 4

Physical Activity Recognition with Dynamically Available Sensors

4.1 Motivation

The previous chapter describes how to create a generic model for activity recognition with limited labelled data. In this chapter, we move a step ahead and deal with the problem of automatically incorporating the dynamically available sensors for physical activity recognition and activity model adaptation. Previous works have demonstrated that extra contextual information can improve the activity recognition accuracy. As described in Section 2.5, this contextual information include location of the user [124], vision feature from the on-body camera [162], objects (e.g. cup) in the environment [96], etc. The underlying reason is that this contextual information is correlated with a particular activity class, and it can better differentiate the activity class from others. Other contextual information include sound, environment color, light and Wi-Fi [8].

Most of the existing works only consider currently available information for creating activity models, and ignore information provided by the dynamically available sensors. The benefit of the additional contextual information motivates us to develop an activity recognition framework that is able to incorporate the information provided by the dynamically available sensors. Another motivation lies in the fact sensors for activity recognition are constantly broken and need to be replaced [92], so a robust activity recognition framework should be able to deal with sensor dynamics in the changing environment. However, several challenges need to be addressed in developing such an activity recognition framework: 1) how to deal with the change in the feature space and the problem of feature redundancy caused by the dynamically available sensors (challenge 5 in Section 1.2), 2) how to select the most informative instances for activity model adaptation (challenges 2, 4 in Section 1.2), 3) how to exploit the temporal information for smoothing activity predictions (challenge 6 in Section 1.2).

There are some works [139, 147] that use knowledge-driven method to specify the parameters of the sensor data with respect to the target activity classes in an unsupervised manner. The parameters are usually specified with common sense or information from a third party knowledge base (e.g. website). Their works are not focusing on the integration of dynamically available sensors for activity recognition, but the knowledge-driven methods can be used to specify the parameters of the dynamically available sensor data and incorporate the sensor automatically. However, the knowledge-driven methods require the sensor readings to be human readable (e.g. door sensor monitors door open event), so it cannot be applied to those sensor readings that are not semantically interpretable. Moreover, knowledge-driven methods usually sacrifice the recognition accuracy for avoiding annotating data, as the knowledge from these methods is not personalised to the activity data of a specific user. There exists some research [60, 161, 35, 158] on dynamically selecting sensors at the runtime to achieve the trade-off between energy-efficiency and activity recognition accuracy. Even though they propose methods to deal with sensor dynamics automatically, the parameters of all possible combinations of sensors with respect to the target activity classes are pre-trained at the training time. Therefore, they are not able to dynamically select the untrained sensors.

In this chapter, we propose methods that address the challenges in physical activity recognition with dynamically available sensors. The key contributions are summarised as follows:

- We propose an activity recognition framework that can automatically incorporate discriminative information provided by dynamically available sensors, so as to improve activity recognition performance.
- We propose a method that selects the profitable and informative instances (containing information from dynamically available sensors) to retrain and adapt activity mod-

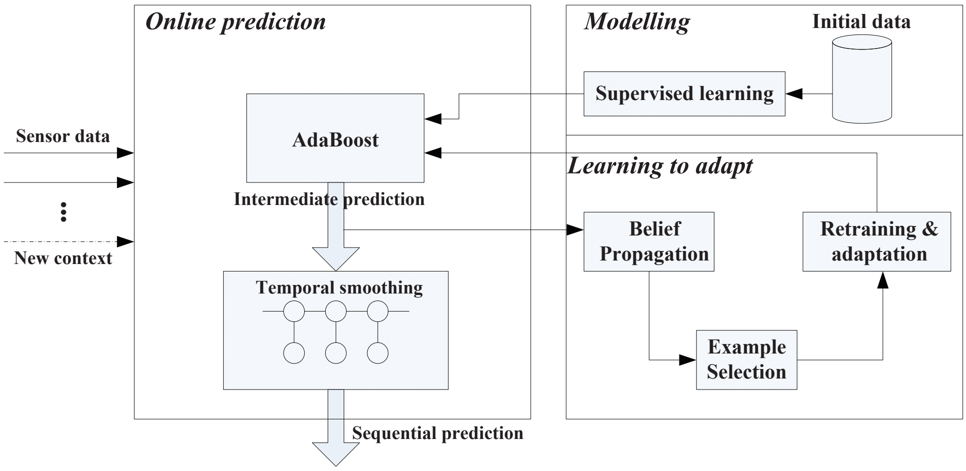


Figure 4.1: Top level framework.

els without human intervention. We also propose a novel way of combining basic classifier (i.e., AdaBoost) with graphical models (i.e. Hidden Markov model and Conditional Random Field) in order to exploit the temporal information to improve the recognition accuracy.

- We demonstrate our methods with three publicly available datasets and analyse their effectiveness through comprehensive experimental and comparison studies. We also investigate the conditions under which the opportunistically available information is beneficial to recognition performance.

4.2 Framework

The workflow of our framework can be divided into three phases: *modelling*, *learning to adapt* and *online prediction*. In the *modelling* phase, an initial activity model is created with currently available sensor data. As new data sources become dynamically available, we perform adaptation of the activity model by considering the dynamic data sources in the *learning to adapt* phase. In the *prediction* phase, the initial model is combined with the graphical models to exploit the temporal information to further improve the recognition performance. It should be noted that *prediction* is not the final stage. Instead, we keep looping between *learning to adapt* and *prediction* as long as new sensor data is available.

- *Modelling*. We choose AdaBoost as our basic classifier, as it is lightweight enough for on-body devices and has been demonstrated to be robust for classification tasks [64]. The rationale for choosing AdaBoost also lies in the fact that it is flexible in the dimension of the feature space, and is able to automatically select the most discriminative features in the training process. The characteristics of AdaBoost makes it extremely suitable for our framework since we need to dynamically incorporate context into our framework, which would change the feature space. Also, we only consider the discriminative context which is beneficial to the recognition performance.

- *Learning to adapt*. When new data sources (e.g. accelerometer, gyroscope) are dynamically available, the information they provide may be beneficial to improving the recognition accuracy. The goal of this stage is to perform adaptation for the activity models, so as to incorporate the information provided by the new data source (if it is discriminative enough). To achieve this, we perform belief propagation on the predictions given by AdaBoost and select instances for retraining. The selected instances, which contain newly discovered context, are fed into AdaBoost to retrain and adapt the classifier. Belief propagation is to exploit the temporal information to rectify the posterior distribution for the instances, based on which we propose a method to select the informative instances without human intervention.

- *Prediction*. AdaBoost makes prediction individually and assumes no dependency between the posterior probability of neighbouring instances. We combine AdaBoost with graphical models to provide sequence predictions, as those models make temporal assumptions between adjacent predictions and are able to smooth out the outliers. We found that the posterior probability distribution of each instance and learned weak learners of AdaBoost make it extremely feasible to combine AdaBoost with graphical models.

4.3 Methodology

4.3.1 Basic modelling

AdaBoost is selected as our basic classifier as it is flexible in the dimensionality of the feature space, and it is able to automatically select the most discriminative features in the training process. As AdaBoost incrementally builds weak classifiers on the training dataset, it is more flexible in the dimensional changes of the feature space. When discriminative context is detected during the *learning to adapt* phase, all AdaBoost has to do is training a weak learner on the context and add it to the ensemble along with its weight, without the necessity to change the feature space and retrain the whole model. Also, in each iteration, AdaBoost only chooses the weak learner with minimum training error. In this light, it presents an effective and tractable way to automatically select the features with maximum discriminative power [82]. Therefore, it is not necessary to evaluate the discrimination of the new context manually.

AdaBoost learns an ensemble of weak classifiers for each activity class k:

$$H^k(x_i) = \sum_{t=1}^{T} \alpha_t^k h_t^k(x_i) \tag{4.3.1}$$

and the posterior probability of class k given instance x_i is:

$$P(y_i = k|x_i) = \frac{e^{H^k(x_i)}}{\sum_k e^{H^k(x_i)}} \tag{4.3.2}$$

AdaBoost is detailed in Section 2.3.1.

4.3.2 Belief propagation

As new sensors are dynamically discovered, we need to select instances that contain the new sensor data to adapt AdaBoost. The aim in this stage is to leverage belief propagation to smooth the outliers and rectify the results produced by AdaBoost, so as to select the most profitable and informative instances to learn the new context and adapt the activity model.

Due to the temporal characteristic of human behaviours, the current activity is more likely to be continued in the next time slot. Therefore, there are strong correlations among the sequential predictions of the instances. Apparently, AdaBoost makes no use of the temporal information, since it assumes no dependencies among the instances, and performs classifications based solely on the local features. As a result, sensor noises or temporary interruption

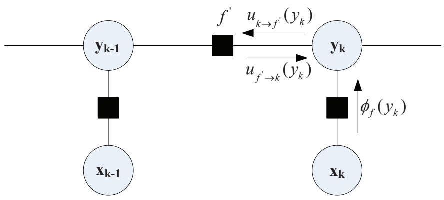


Figure 4.2: Belief propagation between hidden variables

of the activities would certainly result in misclassifications.

Belief propagation is mainly performed for inference in graphical models, and in the form of message passing between the nodes. The messages passed among the nodes are actually exerting influence from one variable to the others. In this light, the belief propagation is to send messages to the connected node and tell it what it should believe [160], and the hidden state of a node depends not only on local observations, but also the product of all incoming messages from locally connected nodes. Upon convergence, the marginal distribution of the variable nodes can be approximated with:

$$p(y_k|\boldsymbol{X}) = \frac{\phi_f(y_k)\prod_{f' \in N(k) \backslash f} \mu_{f' \to k}(y_k)}{\sum_{y'_k} \phi_f(y'_k)\prod_{f' \in N(k) \backslash f} \mu_{f' \to k}(y'_k)} \tag{4.3.3}$$

where $\phi_f(y_k)$ is the local evidence, and $\mu_{f' \to k}(y_k)$ is the message from neighbouring factor nodes of node k, as shown in Figure 4.2.

In our scenario, the belief propagation is performed among the observation nodes and hidden nodes. The observation node at time t is the feature vector collected from the sensor data while the hidden node is the latent activity. Since the latent activity is unknown, the latent variable y_k is represented in the form of a multinomial distribution over all the activities. The multinomial distribution is iteratively updated by incorporating the messages from not only local observations, but also adjacent nodes.

In our framework, we only consider pairwise connections (Figure 4.3) between the hidden nodes when performing belief propagation. Therefore, the messages that a node receives are the posterior probabilities of its neighbouring nodes based on their own local observations, as shown in (4.3.4)

$$p(y_k|\boldsymbol{X}) = \frac{p(y_k|x_k)\prod_{i \in N(k) \backslash i: y_i = y_k} p(y_i|x_i)}{\sum_{y'_k} p(y'_k|x_k)\prod_{i \in N(k) \backslash i: y_i = y'_k} p(y_i|x_i)} \tag{4.3.4}$$

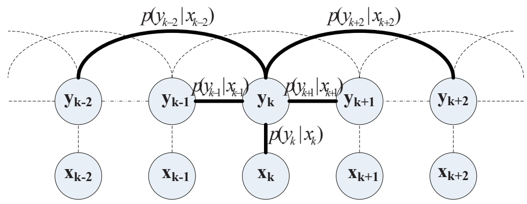


Figure 4.3: Belief propagation in our scenario. The solid lines show the messages received by node k from neighbouring four nodes.

Therefore, belief propagation is performed with an inference step and followed by several iterative update steps. In the inference step, for each observation, AdaBoost generates a posterior probability distribution over the hidden activities using Eq.(2.3.1). In the propagation step, those initial estimations of posterior probabilities are propagated to neighbouring nodes. Those recipient nodes k then combine the received probability distribution over y_i together with its local evidence given by AdaBoost and convert them into a distribution over y_k, using Eq.(4.3.4). The iterative process can be repeated until convergence. In our experiments, we found that running belief propagation for only one iteration is sufficient to converge the posterior distribution.

The belief propagation is slightly modified in our implementation. As the instances classified with high confidence usually tend to be the correct classification, we do not update the posterior distribution for those high-confidence instances during the iterative process of belief propagation, so that their beliefs can be propagated to the uncertain instances.

To demonstrate the effectiveness of belief propagation in smoothing the outliers, we perform physical activity recognition on smart phone sensor data from [130]. The mobile phone is fixed on the belt of the subject performing the activities, inertial data from accelerometer and gyroscope are collected, which is known to be effective for physical activity recognition. The setup parameters such as sliding window length and features are given in the experiment section. We firstly present the results produced by AdaBoost and then those given by running belief propagation, as shown in Figure 4.4 and Figure 4.5 respectively.

The x-axis represents time sequence, while the y-axis represents the posterior probability of the instances. The classifier chooses the activity class that has the maximum confidence in the posterior distribution as the prediction (we highlight those instances with circle). We only plot the classifications of **Standing**, since it is the activity where the most mis-recognitions happen. From Figure 4.4 we can see that, most of the time activity **Standing**

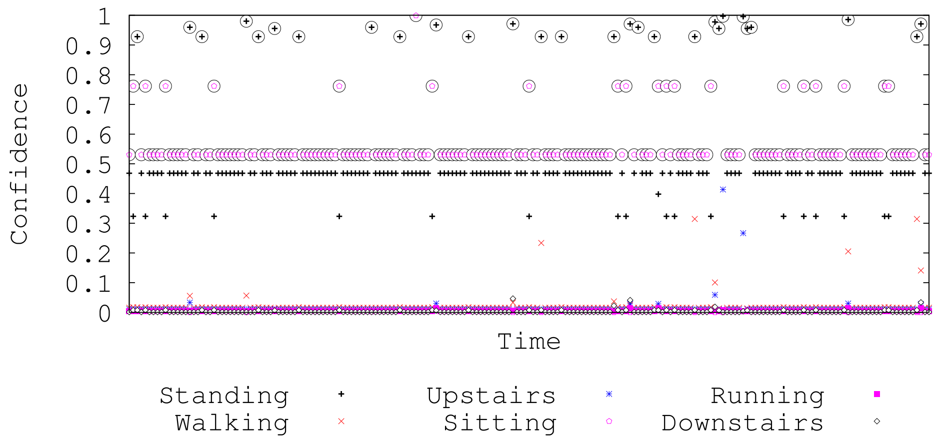


Figure 4.4: Activity recognition results given by AdaBoost

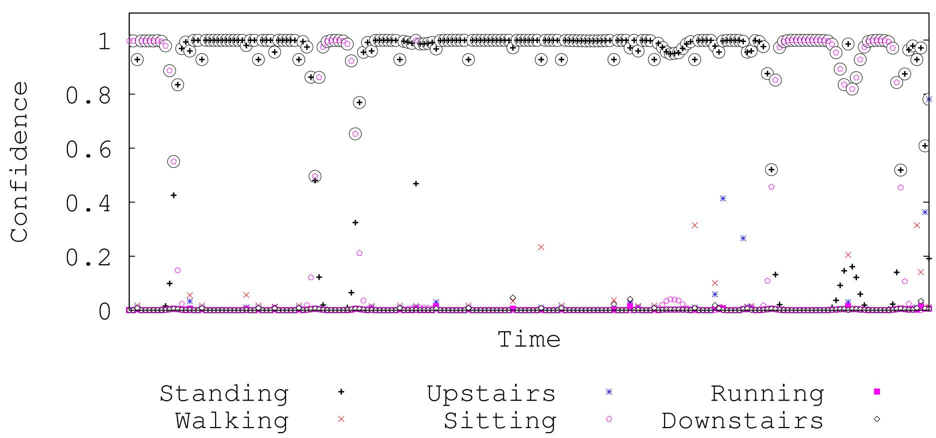


Figure 4.5: After running belief propagation on the results given by AdaBoost

is classified as **Sitting**, due to the similar patterns of those two activities when the data is collected from the belt. However, after belief propagation, most of the misclassifications are rectified, as presented in Figure 4.5. The underlying reason is that, when the instance is mispredicted, the maximum confidence in the posterior distribution is quite low (50%), and the prediction is uncertain. However, when the activity is correctly predicted, the corresponding confidence would reach a rather high level (usually more than 90%). As a result, when running the belief propagation, the nodes with high confidence are able to propagate their belief to neighbouring nodes, so as to clarify the uncertainty. While the nodes with "flat" posterior distribution have little impact on adjacent nodes, because they propagate nearly the same information for each hidden activity.

4.3.3 Instances selection

In this subsection, we introduce the method to select the instances for classifier retraining and adaptation. The instances contain dynamically discovered context, and AdaBoost is able to automatically incorporate the new context if it is discriminative enough. In this way, AdaBoost can be self-adapted or -refined. We perform instances selection after the belief propagation for the sake of selecting the informative and profitable instances to quickly converge the classifier without human intervention.

Measurements

First of all, we introduce the measurements that can evaluate the profitability of an instance, so that based on those quantitative criteria, the instances can be selected to adapt the model. The first metric we consider is the “drift" in the posterior distribution before and after the belief propagation. Belief propagation is able to smooth out the outliers by exploiting the temporal information. Those instances that experience huge “drift" in their posterior distributions are much more valuable, since they are not modelled by the initial activity model and have a greater chance of residing near the classification boundaries. The Jensen-Shannon divergence can be used to measure the “drift", as it has been proved to be efficient to measure the distance between two distributions in previous work [136]. Supposing p_i and q_i are the posterior distributions of instance i before and after belief propagation, respectively, and then the JS-divergence is:

$$JS(p_i, q_i) = \frac{1}{2}D_{KL}(p_i||m) + \frac{1}{2}D_{KL}(q_i||m) \tag{4.3.5}$$

where $m = \frac{1}{2}(p_i + q_i)$ and $D_{KL}(p_i||m) = \sum_j p_{ij} log\frac{p_{ij}}{m_j}$ is the Kullback-Leibler divergence between two distributions. Therefore, we derive the first measurement as:

$$score_{i1} = \frac{JS(p_i, q_i) - JS_{min}(p, q)}{JS_{max}(p, q) - JS_{min}(p, q)} \tag{4.3.6}$$

We normalize the JS-divergence, so that the measurement based on the posterior distribution “drift" is always ranged in [0,1], in this way it is able to cater for characteristics of different activity data sets.

As for the second measurement of profitability, we consider the number of consecutive neighbouring instances that have the same predicted results.

$$\begin{aligned} N_i &= min(N_i^{forward}, N_i^{backward}) \\ score_{i2} &= \frac{N_i - min(N)}{max(N) - min(N)} \end{aligned} \tag{4.3.7}$$

where $N_i^{forward}$ and $N_i^{backward}$ are the number of consecutive neighbouring observations that have the same predictions along the two directions of time series, from the current observation i. It is normalized due to the same reason as $score_{i1}$. This measurement shows the extent to which the neighbouring nodes have the consensus predictions, and the higher the number, the more likely that the prediction is correct. Obviously, $score_{i2}$ is proposed based on the temporal characteristic of human behaviour. One extreme condition is that the observation happens to be in the middle of an ongoing activity, and the $score_{i2}$ tends to be large and it is more confident about the prediction.

Finally, we consider the confidence of the instances after the belief propagation. The posterior distribution itself provides the information about the confidence of an instance. Adding the instances with the highest confidence is equivalent to locating the class center, which in turn also helps to adapt the model to some extent, even though those instances are less informative. Therefore, the third measurement is formulated as $score_{i3} = max(p(y_i|x_i))$ (Eq.(4.3.4)).

To decide which instance is more profitable, we need to take into account all the aforementioned metrics. Therefore, we determine the final score for the profitability of an instance based on the corresponding scores for each of the metrics. The combined score is defined as follows:

$$\begin{aligned} score_i &= \alpha_1 score_{i1} + \alpha_2 score_{i2} + \alpha_3 score_{i3} \\ s.t. \sum_{i=1}^{3} \alpha_i &= 1 \end{aligned} \tag{4.3.8}$$

where the weights α_i is manually given. In our method, we evenly distribute the importance to the three metrics by setting $\alpha_1 = \alpha_2 = \alpha_3$. However, by giving different weights, the model may present different characteristics. For example, by increasing α_3 we give more weight to the high-confidence instances, and then the model adapts conservatively and the convergence is quite slow. By contrast, when we put more weight to $score_{i1}$, the model only takes those instances whose posterior distribution changes dramatically before and after belief propagation, and then the adaptation is performed aggressively. There is a danger that noisy data may be added and the model is jeopardised.

Retraining

Upon selecting the instances for model adaptation, AdaBoost can automatically determine the discriminative power of the new context (if there is any) in the instance, and dynamically incorporate them for classification if they are discriminative enough. In this way, the model is adapted to new coming data.

One issue should be addressed when selecting the instances is that the amount of retraining data among different activity classes should be balanced during the adaptation process. During the experiments we found that for an activity class with a small training dataset, the iterative process of training weak learners is unexpectedly terminated earlier. As a result, the trained ensemble of classifiers for that class overfits the small amount of data. That is the reason that AdaBoost focuses more on training activity classes with unevenly large proportions of annotated data [64]. Therefore, in this paper, we accumulate for each activity class the same amount of data before retraining.

4.3.4 Sequential prediction

When the adapted AdaBoost is deployed for online prediction, we combine it with graphical models to further smooth outliers. Even though the basic idea behind this stage and belief propagation are both to exploit the temporal information among the activity data, belief propagation is deployed for offline data analysis, sufficient data should be accumulated and analysed for model adaptation (second stage in Figure 4.1), while graphical models cater for online lightweight predictions (third stage in Figure 4.1). Furthermore, belief propagation

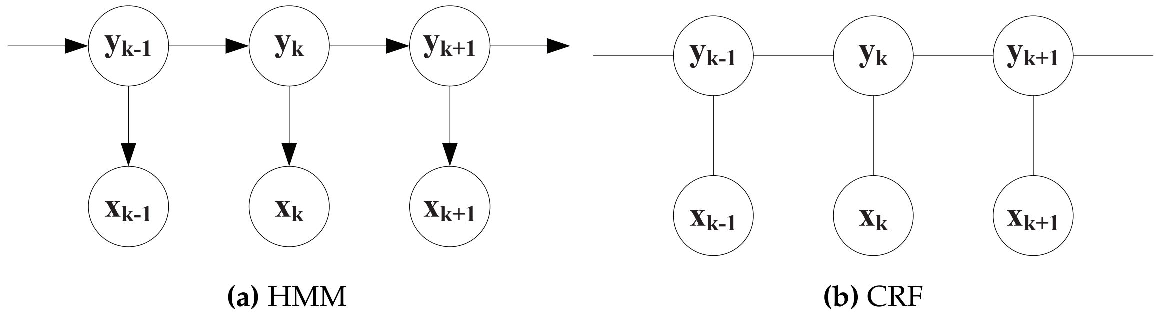


Figure 4.6: Graphical model of HMM and CRF

requires the posterior distribution to evaluate the profitability of the instances.

In this section, we introduce the methods of combining AdaBoost with Conditional Random Field, referred to as BoostCRF. It should be noted that, hybrid classifiers are not new topics, in [82, 96, 162] the authors used the posterior probabilities from discriminative classifiers as new input features to train HMM or CRF. However, the modelling of discriminative classifiers is dissociated from the modelling of structured classifiers. Therefore, the two classifiers are trained independently, using the output of one classifier as input for another. Moreover, they train HMM for each activity class separately, and during the inference phase for all the instances in a sequence, they produce the same label that has maximum likelihood. Therefore, they do not model the transitions among different classes.

In a Hidden Markov model, the variables include hidden states and observations. As shown in Figure 4.6a, it models the joint distribution of those variables and naively assumes that hidden state y_k at each time step k only depends on hidden state at previous time step, y_{k-1}, while observation x_k at time k only depends on the hidden state at the same time slice, as shown in Figure 4.6a. Therefore, HMM can be mathematically described by three parameters: the initial state y_1, transition distribution $p(y_k|y_{k-1})$, and emission probability $p(x_k|y_k)$, then the joint distribution of the variables can be formulated as follows:

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(y_1)p(x_1|y_1)\prod_{k=2}^{K} p(y_k|y_{k-1})p(x_k|y_k) \tag{4.3.9}$$

Now we show how to combine AdaBoost with HMM. At each time slice, we can obtain the posterior probabilities $p(y_k|x_k)$ for the observation using Eq.(2.3.1). Then the emission probability can be obtained according to Bayes' rule:

$$p(x_k|y_k) = \frac{p(y_k|x_k)p(x_k)}{p(y_k)} \propto p(y_k|x_k) \tag{4.3.10}$$

where prior knowledge $p(y_k)$ is identical for different activities because we balance the training data over all the activity classes. For a variable x_k that is observed at time k, $p(x_k)$ is a constant when calculating its evidence against different classes. Therefore, the emission probability is proportional to the posterior probability given by AdaBoost, and the joint distribution can be re-formulated as follows:

$$p(\boldsymbol{x}, \boldsymbol{y}) \propto p(y_1)p(y_1|x_1)\prod_{k=2}^{K} p(y_k|y_{k-1})p(y_k|x_k) \tag{4.3.11}$$

As for transition probability, we manually set the self-transition probabilities to be large to temporally smooth out the activities, and encourage them to continue unless observable evidence strongly suggests a different activity [147], denoted as follows:

$$p(y_k|y_{k-1}) = \begin{cases} 1-\epsilon & y_k = y_{k-1} \\ \epsilon & \text{otherwise} \end{cases} \tag{4.3.12}$$

We experimentally set ϵ to be 0.1, as it is demonstrated to be effective enough to achieve reasonable accuracy. Inferring the hidden states is equivalent to finding the sequences that maximize the joint probability depicted in Eq.(4.3.11), which can be performed by the Viterbi algorithm.

It should be noted that, it is infeasible to apply HMM directly on feature vectors from sensor data. Since feature vectors from activity data usually consist of a large number of dimensions. When we model the feature vectors as the Gaussian distribution, a large number of parameters in the covariance matrix would result in the problem of overfitting [136]. Moreover, changes in the feature space resulting from the incorporation of new contexts would require the whole model to be retrained. We use a sliding window with a constant number of observations, and perform the Viterbi algorithm on this sequence within the window. The window is shifted along the time axis as new instances come in. In this way, we can provide real-time prediction.

BoostCRF

Rather than modelling the joint distribution of the variables, Conditional Random Field (CRF) models the conditional distribution of the hidden variables over the observations. The relationships between the connected nodes are now described with potential functions that map them to positive numbers. One advantage of the CRF over HMM is that, it does not assume the dependencies among variables, and it is much more flexible to define the potential function.

Due to the flexible definition of the potential functions, CRF has various structures. We only consider linear-chain CRF (Figure 4.6b). Therefore, we need to define local potential functions between observation and hidden node at each time step, and pairwise potential functions between consecutive hidden nodes. The conditional distribution can be formulated as:

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{x}) &= \frac{1}{Z(x)} exp\left(\sum_{k=1}^{K} \lambda^T f(y_k, y_{k-1}, x_k)\right) \\ &= \frac{1}{Z(x)} exp\left(\sum_{k=1}^{K} \left(\lambda_s^T f_s(y_k, y_{k-1}) + \lambda_j^T f_j(y_k, x_k)\right)\right) \end{aligned} \tag{4.3.13}$$

where $f_j(y_k, x_k)$ and $f_s(y_k, y_{k-1})$ are the local and pairwise potential functions at time k. λ_s and λ_j are the corresponding weights. $Z(x)$ is the normalization factor, formulated as $\sum_{\boldsymbol{y}} exp\left(\sum_{k=1}^{K} \lambda_k f_k(y_k, y_{k-1}, x_k)\right)$.

Inspired by [83], we map the weak learners trained in AdaBoost to the local potential functions in CRF, while the weights of the potential functions are mapped to the weights of the weak learners. This is reasonable since more weights are given to the potential functions that can better explain the data, whereas weak learners with less error rate have a larger weight. Using Eq.(2.3.2), the weighted sum of local potential functions against activity class i is:

$$\lambda_j^T f_j(y_k, x_k) = \sum_{t=1}^{T} \alpha_t^i h_t^i(x_k) \tag{4.3.14}$$

However, mapping the weight of a pairwise potential function is non-trivial. To deal with this, we define a pairwise potential function that characterises the temporal transition between activities:

$$f_{ij}(y_k, y_{k-1}) = \begin{cases} 1 & y_k = i, y_{k-1} = j \\ 0 & \text{otherwise} \end{cases} \tag{4.3.15}$$

where potential function f_{ij} characterises the transition from activity j to activity i. Assume that there is a weak learner $h^i(y_k = i, y_{k-1} = j)$ in AdaBoost that can be mapped to the potential function f_{ij}. Obviously, the error rate of the weak learner can be estimated from the training dataset by frequency counting:

$$\epsilon_{ij} = 1 - \frac{\text{expected number of transitions from j to i}}{\text{expected number of transitions out of j}} \tag{4.3.16}$$

then according to Algorithm 1, the weight of the weak learner can be approximated as:

$$\alpha_{ij} = \frac{1}{2} ln(\frac{1 - \epsilon_{ij}}{\epsilon_{ij}}) \tag{4.3.17}$$

The weight of weak learner $h^i(y_k = i, y_{k-1} = j)$, α_{ij}, is mapped to the weight of the pairwise potential function f_{ij} in CRF. Once we have the parameters, the inference process can be carried by loopy belief propagation to find the most likely assignment of the latent activities. Notice that, we have T local potential functions, but only 1 pairwise potential function, thus the temporal evidence weighs less when compared with local evidence. Therefore, we multiply the pairwise potential functions with a constant (average number of weak learners of the activity classes), so that the inferred results do not overfit the local evidences.

4.4 Experiment

In this section, we validate our methods introduced in the previous sections. We firstly describe the datasets, and then specify the method to evaluate our approaches.

4.4.1 Datasets

- Smartphone dataset (SD) [130]: Activity data is collected from accelerometer, gyroscope and magnetometer on an Android device worn in different body positions (arm, belt, wrist and pocket), when the subject performs standing, walking, upstairs, sitting,

Table 4.1: Dataset description.

Datasets	Users	Sensors	Activities (Instances)
SD	1	accelerometer, gyroscope, magnetometer, linear acceleration sensor	walking (2521), sitting (2391), standing (2392), running (2311), upstairs (1744), downstairs (1487)
SAD	10	accelerometer, gyroscope, magnetometer, linear acceleration sensor	walking (1790), standing (1790), jogging (1790), sitting (1790), biking (1790), upstairs (1790), downstairs (1780)
UCI	30	accelerometer, gyroscope	walking (1722), upstairs (1544), downstairs (1406), sitting (1777), standing (1906), lying (1944)

running and downstairs. The sample rate is set to be 50Hz. We compute time domain features such as mean, standard deviation, median, zero crossing rate, variance, root mean square for each axis of the sensors with a 2 sec sliding window and 50% overlap.

- Sensors activity dataset (SAD) [130]: Sensor data is collected when the 10 volunteers perform standing, walking, upstairs, sitting, downstairs, jogging and biking. The activity data is collected from four different body positions (i.e. arm, belt, wrist and pocket), but we only use the data collected from arm, as experiments on the other dataset present the same trends. The data preprocessing methods (e.g. segmentation, feature extraction) is the same as the previous dataset.

- UCI HAR dataset [6]: The dataset is collected with accelerometer and gyroscope from a Samsung Galaxy SII smartphone worn by 30 volunteers. The smartphone was fixed on the waist when the subjects perform six activities (walking, walking_upstairs, walking_downstairs, sitting, standing, laying). The 561 features were computed based the sliding window of 2.56 sec and 50% overlap. In our experiment, we only consider time domain features, as it is computationally expensive to compute the frequency domain features on the mobile phone during online prediction. Therefore, we have 80 features from the gyroscope and 120 features from the accelerometer.

4.4.2 Set up

To validate the proposed framework, each of the datasets is divided into three portions, in accordance with the three stages in Figure 4.1. Specifically, for all the datasets, we train the activity model with the first part of the dataset that contains **only** gyroscope data at the first stage. At the second stage, the activity model is used to classify the second part of the dataset which contains **both** accelerometer and gyroscope data, and after offline data analysis we select the profitable instances to retrain the activity model, and features from the accelerometer are automatically incorporated into AdaBoost if they are discriminative. In the final stage, we classify the third part of the dataset with the adapted model and compare the results with ground truth.

The first dataset is personalised, we evenly partition the dataset into three parts and perform 6-fold cross validation. While the SAD and UCI datasets involve multiple volunteers, so we perform leave-one-user-out cross validation.

In what follows, we will validate the effectiveness of our framework in terms of several aspects, especially the ability to incorporate new context, the importance of belief propagation and instances selection, the benefit of combining AdaBoost with graphical models. Finally, we investigate the conditions under which our methods provide a marginal improvement or even jeopardise the initial model.

4.4.3 Incorporating new context

In this section, we validate our method by building an activity model with gyroscope data, and dynamically incorporating accelerometer data to refine the model. 300 weak learners are trained for each activity and the *score* threshold is set to be 0.7 to select instances for retraining, as it is low enough to select sufficient training data and high enough to exclude the noisy instances. We do not perform the iterative process to select the instances and retrain the model, as we found that additional iterations do not provide significant accuracy improvement according to our experiments. On the other hand, repeatedly retraining the model is expensive. For all the experiments, we compare the recognition performance in terms of **f1-score** (f1-score=$\frac{2*precision*recall}{precision+recall}$).

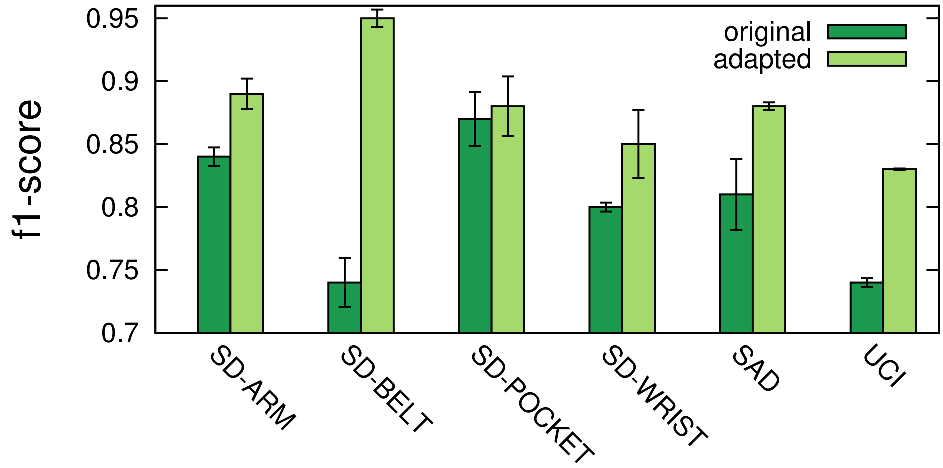


Figure 4.7: F1-score improvement by dynamically and automatically incorporating accelerometer data.

In Figure 4.7, we can see that, our method (*adapted*) can improve the recognition accuracy to some extent across the datasets, especially for the dataset that the user fixes the smartphone on the belt. Because it is difficult to distinguish standing and sitting with gyroscope when the device is put on the belt. However, as belief propagation is able to correct most of the uncertainties, and then the retraining instances would help to refine the initial model. Furthermore, the f1-score improvement in SD-POCKET setting is marginal. When debugging the learning process, we found that only one weak learner is trained to classify the activity **Sitting**, that means the weak learner overfits the retraining dataset and is unable to classify **Sitting** during the prediction stage if the activity presents variance. However, when we lower the *score* threshold and collect more instances for retraining, the f1-score achieves 0.94.

In order to confirm the usefulness of extra features, we count the proportion of weak learners that are trained on the new features during the retraining process. Since AdaBoost is able to automatically select the weak learner that has the minimum weight error rate in each iteration, the more that the weak learners are trained on the new features, the more discriminative the new features are. As is presented in Figure 4.8, for most of the dataset the proportions of weak learners trained on new features are more than 50%. From the figure we can see that dataset SD-BELT and SD-POCKET have the proportions of 62% and 38% respectively. The underlying reason is that, for the dataset SD-BELT the accelerometer features can better distinguish standing and sitting, and then during the retraining process, more weak learners are trained on the accelerometer data. While in SD-POCKET dataset, the retraining process terminates unexpectedly early for the activity **Sitting**, and fewer weak learners are

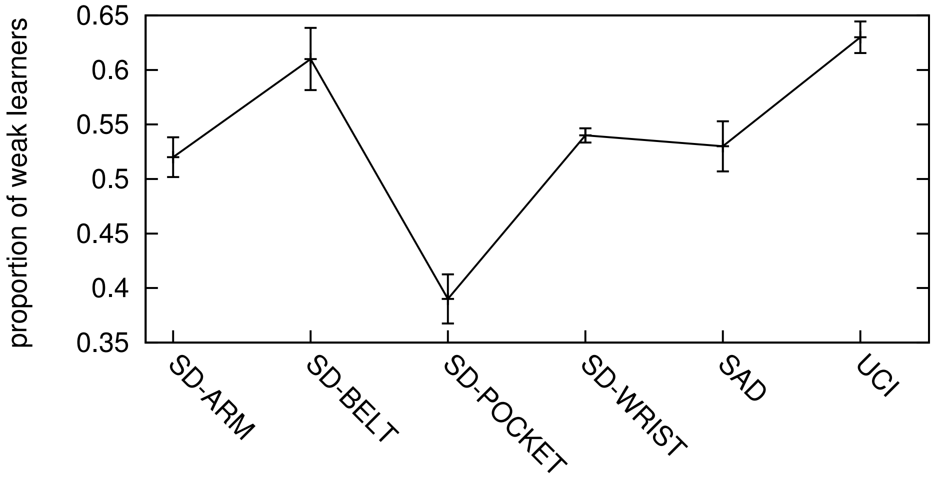


Figure 4.8: Proportion of weak learners trained on new features during the retraining process across the datasets.

trained on the retraining dataset and hence the new features cannot be sufficiently leveraged for performance improvement.

4.4.4 Role of belief propagation

In this subsection, we will examine the role that belief propagation plays in our framework. For comparison, we introduce the following baselines:

- *noBelief*: setting without belief propagation on the intermediate predictions of AdaBoost and select the instances classified with high confidence for retraining. Notice that the instances for retraining contain the dynamically available features (i.e. accelerometer data).
- *noExtra*: setting without belief propagation and not considering the dynamically available sensors. This is exactly the traditional semi-supervised learning that selects the most confident instances to adapt the activity model.

The configurations for these two methods are the same as ours except that the confidence threshold is set to be 0.7 to select instances for retraining. The result is presented in Figure 4.9, from which we can see that for most of the datasets, *noBelief* and *noExtra* provide marginal f1-score improvement. In some case, *noExtra* even experiences performance loss. The reasons are two-fold. On the one hand, high-confidence instances are usually less in-

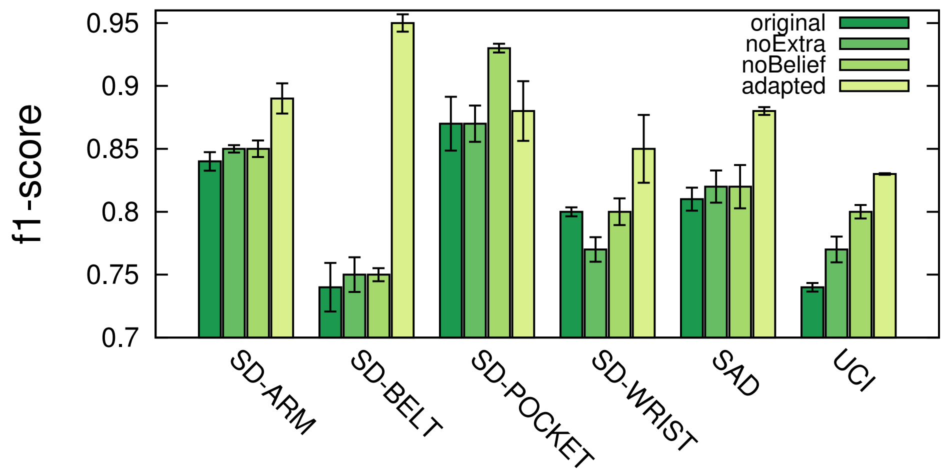


Figure 4.9: Comparison with *noBelief* and *noExtra* in terms of f1-score.

formative and make less contribution to the f1-score improvement. On the other hand, it is difficult to set a universal confidence threshold for all datasets. For example, in the dataset SAD, the activity **Sitting** is frequently classified with a confidence lower than 0.7 (the confidence threshold). Due to the enforcement of retraining data balance, insufficient data of sitting results in a small amount of retraining dataset and hence, less contribution in f1-score improvement. While in the dataset SD-WRIST, a confidence threshold of 0.7 introduces the noisy instances and has a negative impact on the recognition performance.

An exception is found in the dataset SD-POCKET, in which the *noBelief* achieves the f1-score as high as 0.93, as gyroscope performs better than accelerometer in pocket position, confirmed by [130]. Therefore, initial model with gyroscope is able to correctly recognize most of the activities with high confidence, and provides true labels for the retraining with the combination of accelerometer and gyroscope data, hence the resulting model can then significantly improve the recognition performance. As discussed in the previous subsection, our method is able to obtain 0.94 in f1-score when we lower the *score* threshold.

It should be noted that for most of the datasets (except UCI), traditional semi-supervised method (*noExtra*) does not provide performance improvement. However, it does not necessarily mean the contradiction between our experiments and previous work [133]. In our cases, the recognition performance is limited by the discriminative power of the features rather than the amount of training data, as we create the initial model with sufficient training data, especially for the later two datasets which include activity data from multiple users. The dataset SD-POCKET supports our conclusion. Both *noExtra* and *noBelief* take the exactly the same data for retraining, but only *noBelief* results in model refinement, due to

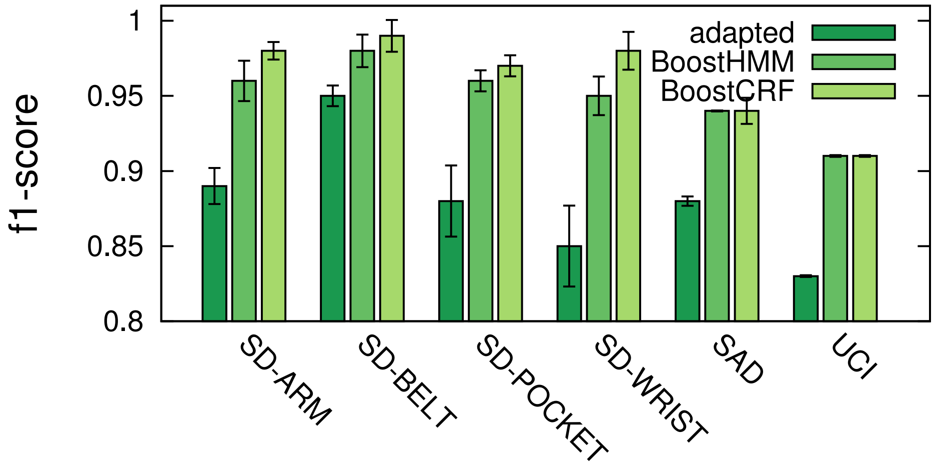


Figure 4.10: Combining adapted AdaBoost with HMM and CRF.

the fact that it incorporates acceleration features.

To conclude, by incorporating newly discovered features, our method outperforms traditional methods that simply consider the most confident instance, and belief propagation followed by the instances selection scheme achieves significant improvement in terms of the recognition performance.

4.4.5 Role of graphical models

In this subsection, we evaluate the recognition performance by combining AdaBoost with CRF, which is to smooth the accidental predictions given by AdaBoost.

The results are shown in Figure 4.10, from which we can see that by temporarily smoothing the outliers, the f1-score can be improved by 7.9% and 8.2% with BoostHMM and BoostCRF respectively. The figure also shows that BoostCRF performs slightly better than BoostHMM, which has been confirmed by previous work [144]. The reason is that, BoostHMM makes strong assumptions among the variables while BoostCRF have more flexible structures and relationships between connected nodes. Actually, when we look at the results provided by BoostHMM, instances of some continuous activities are still sporadically classified as other classes.

For the datasets SAD and UCI, BoostCRF seems to present no advantage over BoostHMM. This is because we only perform one iteration during the inference process for BoostCRF. It

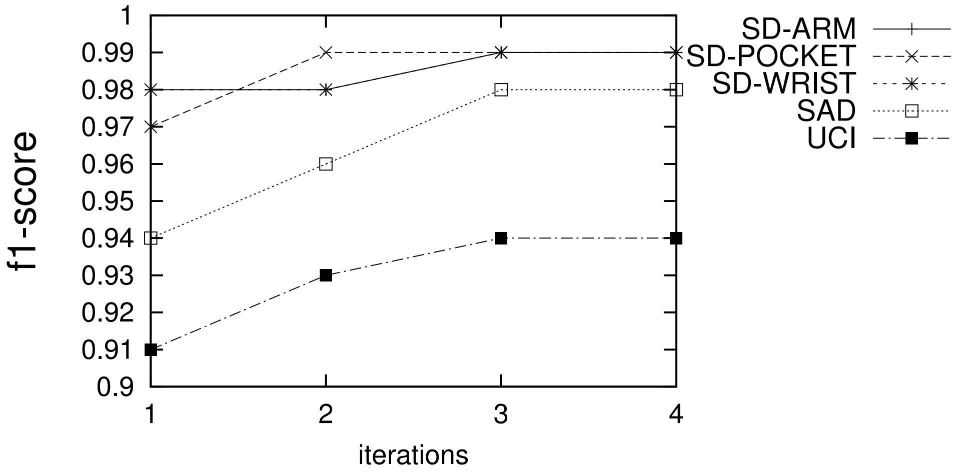


Figure 4.11: F1-score corresponding to the number of iterations during inference process for BoostCRF.

seems that one iteration is not enough to converge the model, because more iterations can still improve the f1-score, as shown in Figure 4.11. It should be noted that, the authors in [6] use Support Vector Machines (SVMs) to classify the activity with the same dataset, UCI, and obtain the average accuracy of 89.0%. By comparison, we are able to achieve the f1-score of 94.0% with BoostCRF. However, we only use the 80 gyroscope features while they build their model on the all of the 561 features.

4.4.6 Investigation of the usefulness of extra context

In this subsection, we investigate the conditions under which the extra context cannot help with the accuracy improvement. To this end, we make the following assumptions and perform experiments with the datasets to validate those hypotheses.

- When the extra context provides less discriminative information compared with existing features.
- When the initial model is not accurate enough to perform adaptation.

The basic idea is that extra context, which cannot better characterise the activity classes or are less discriminative than the features upon which the initial model is built, are automatically ignored during the retraining process. Secondly, if the initial model is not accurate enough, misclassified instances would be selected for retraining and jeopardise the model.

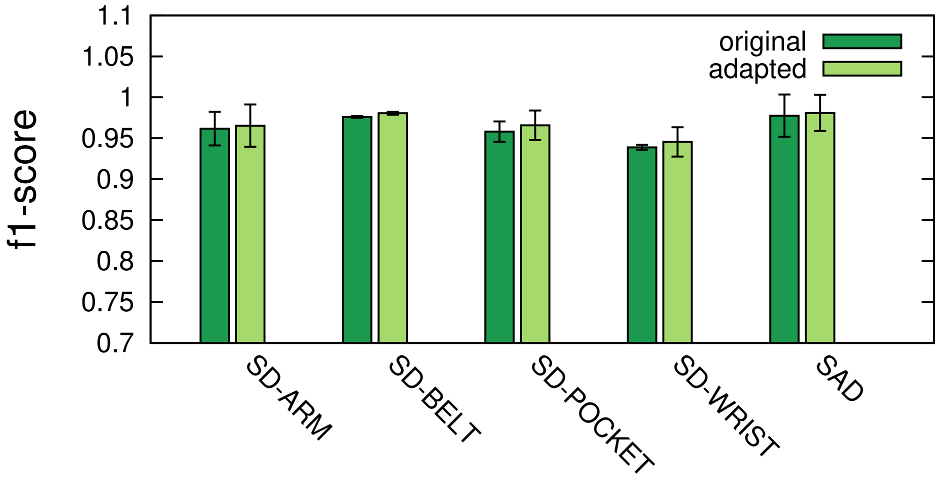


Figure 4.12: Performance(f1-score) improvement by incorporating magnetometer features, we do not experiment on dataset UCI as it does not provide magnetometer data.

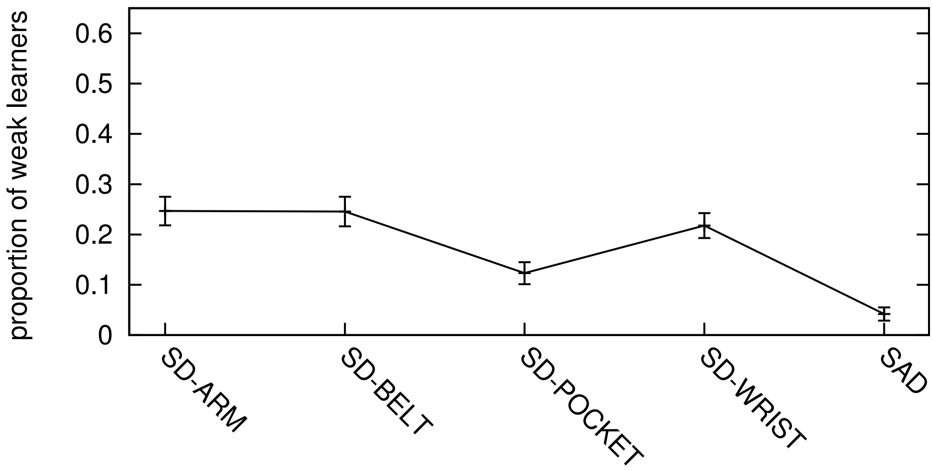


Figure 4.13: Percentage of weak learners that are trained on magnetometer features during the adaptation process.

To validate the first assumption, for dataset SD and SAD, we build the initial model with accelerometer and gyroscope data. During the learning and adaptation stage, the instances contain accelerometer, gyroscope and magnetometer data. Magnetometer feature is demonstrated to be less discriminative [130]. The results are illustrated in Figure 4.12, from which we can see the f1-score improvement is insignificant, less than 1% on average. Figure 4.13 provides a more insightful reason, which shows that only a small portion of weak learners are trained on dynamically available features, since they are less discriminative and not beneficial to the accuracy improvement.

In order to validate the second assumption, we limit the size of initial training dataset, so that the initial model would overfit the dataset and result in an inaccurate classifier. We use 5% of the training data to build the initial model, and present the results in Figure 4.14. From the

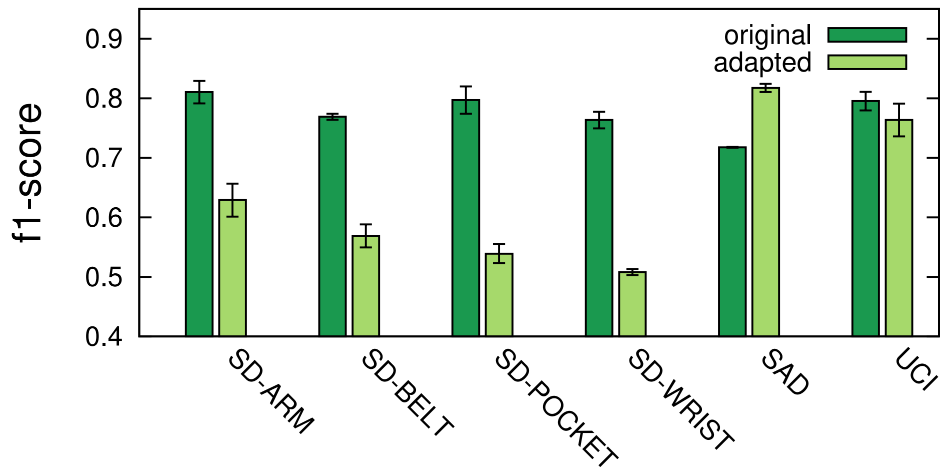


Figure 4.14: Performance(f1-score) decrement with an inaccurate initial model.

figure one can see that, the adapted model would be negatively affected if the initial model is not accurate enough. The underlying reason is that wrongly predicted instances are added to retrain the model. One potential solution to this problem is to be more conservative and increase the weight α_3 in Eq.(4.3.8). However, it is out of the scope for this thesis and is left for future work.

4.5 Summary

In this chapter, we develop a framework that automatically incorporates dynamically available sensors for low-level activity recognition. In the framework, the most informative instances are selected to adapt and refine the initially created activity model. AdaBoost can automatically select the most discriminative features during the adaptation process. We also leverage the temporal information of human behaviour to boost the performance, both in the off-line data analysis and online predictions. Experimental results show that the recognition performance can be significantly improved with dynamically available sensor data. The proposed method is able to select the valuable instances to adapt and refine the model without human intervention, and its combination with the graphical models is able to further improve the recognition accuracy.

The framework proposed in this chapter addresses challenge 2 described in Section 1.2; the instance selection method proposed in Section 4.3.3 addresses challenge 4 described in Section 1.2; the combination with HMM and CRF addresses challenge 6 in 1.2; the basic classifier in the framework, AdaBoost, is flexible with the dimensionality of the instances and selects the most discriminative features in the training and adaptation process, hence it addresses challenge 5 in Section 1.2

Sensors used for recognising primitive physical activities are limited to on-body sensors such as accelerometers and gyroscopes. As a result, we know how to process the sensor readings into proper features (e.g. time domain features, frequency domain features) when they are dynamically available. However, recognising high-level daily activities (e.g. kitchen activities) require different types of environment-instrumented sensors and on-body sensors. The motivations and methods for incorporating dynamically available sensors for high-level activity recognition are described in the next chapter.

CHAPTER 5

High-level Activity Recognition and Adaptation with Dynamically Available Contexts

5.1 Introduction

The previous chapter describes the methods that automatically incorporate dynamically available on-body sensors (e.g. accelerometer, gyroscope) for low-level activity recognition and activity model adaptations. Since the sensor readings from those sensors are not semantically interpretable, conventional machine learning methods are required to learn the mapping from the sensor readings to the activity classes. While for high-level activity recognition, a variety of knowledge can be leveraged to specify the interrelations between the activities and contexts. Examples include knowledge from the website [116, 156, 37], knowledge from the existing data [165, 142, 139] and knowledge manually specified by domain experts [147, 154]. This knowledge can be used for unsupervised activity modelling and recognition [19, 125], activity transfer learning [51], unseen activity class learning [23] and unseen object parameter learning [139].

However, there are still some challenges that need to be addressed for high-level activity recognition with dynamically available contexts. They include 1) how to process readings of dynamically available sensors into proper contexts for activity model adaptation (challenge 7), 2) what kinds of algorithms should be used to incorporate dynamically available sensors and perform activity model adaptation (challenges 2, 3 in Section 1.2), 3) how to exploit temporal information for high-level activity recognition and activity model adaptation (challenge 6 in Section 1.2). Sensors that can be used for human activity classification usually have different modalities, so data produced by the same type of sensors may need to be interpreted differently when used for different purposes. As for the activity model adaptation, even though domain knowledge can be used to integrate dynamically available sensors by specifying the parameters of contexts with respect to the target activities, this domain knowledge cannot obtain optimal results as people perform activities differently [168, 152].

In this chapter, we propose methods to approach the aforementioned challenges, and the key contributions are summarised as follows:

- We propose a high-level activity recognition framework that is able to integrate dynamically available sensors upon their discovery, and to adapt the activity models to take advantage of these additional contexts produced by newly available sensors.

- We propose sensor and activity models to facilitate sensor readings pre-processing of dynamically discovered sensors, and the incorporation of the contexts that the sensors provide into the activity models.

- We develop a knowledge-driven method for incorporating dynamically available contexts without supervision. The parameters of the contexts with respect to different activity classes are estimated using descriptive texts of the activities from external sources (e.g. website) and natural language processing methods.

- We propose a data-driven method for incorporating dynamically available contexts with the learning-to-rank machine learning method and temporal regularization. The data-driven method is a personalised method as it performs machine learning with the activity data of a specific user.

- We validate the proposed data-driven method using one of the most complex human activity recognition datasets, OPPORTUNITY, and demonstrate the advantage of the proposed data-driven method over traditional personalised activity recognition methods.

- We validate the proposed knowledge-driven method using both the OPPORTUNITY dataset and a simulation dataset.

5.2 Context modelling

Various types of sensors can be used for activity recognition, they include body-worn sensors, object sensors and ambient sensors [126], etc. Incorporating the dynamically available sensors into the recognition framework may potentially benefit the recognition task. If a run-time discovery of such a sensor happens then raw sensor data provided by the sensor may need to be pre-processed into the context information required by the recognition framework. For example, for a binary sensor that is used for object interaction monitoring, the output value of the sensor directly indicates whether the user is interacting with the object or not. While for inertial sensors (e.g. accelerometer, gyroscope) that are used for the same purpose, the continuous sensor values may need to be pre-processed into a proper feature vector and a clustering method used to indicate the interaction. Due to the heterogeneity of the sensors, we propose to model the context of sensors so that after the sensor dynamic discovery we can find out from the model how the sensor readings can be processed and incorporated into the activity recognition framework.

The sensor context models capture the necessary information about the sensors. There are many approaches to context modelling. We adopt the *fact-based* approach from [47] to model the sensors. It is able to model the type of sensors and also metadata that is necessary to integrate the sensors into activity recognition. Notice that Hu et al. [52] proposed to model sensors for autonomic mapping between the abstract context required by applications and sensors providing raw sensor data (in order to be able to replace one type of sensor by a different sensor that can produce the same abstract context), while we model sensors so that the dynamically available sensors can be automatically integrated for activity recognition and activity model adaptation by processing the modelled context information.

Figure.5.1 illustrates the necessary context information that is required to model a sensor for our purpose. The model includes the sensor type (e.g. on-body sensor, ambient sensor), sensor reading type (e.g. continuous, discrete), a model for pre-processing sensor readings

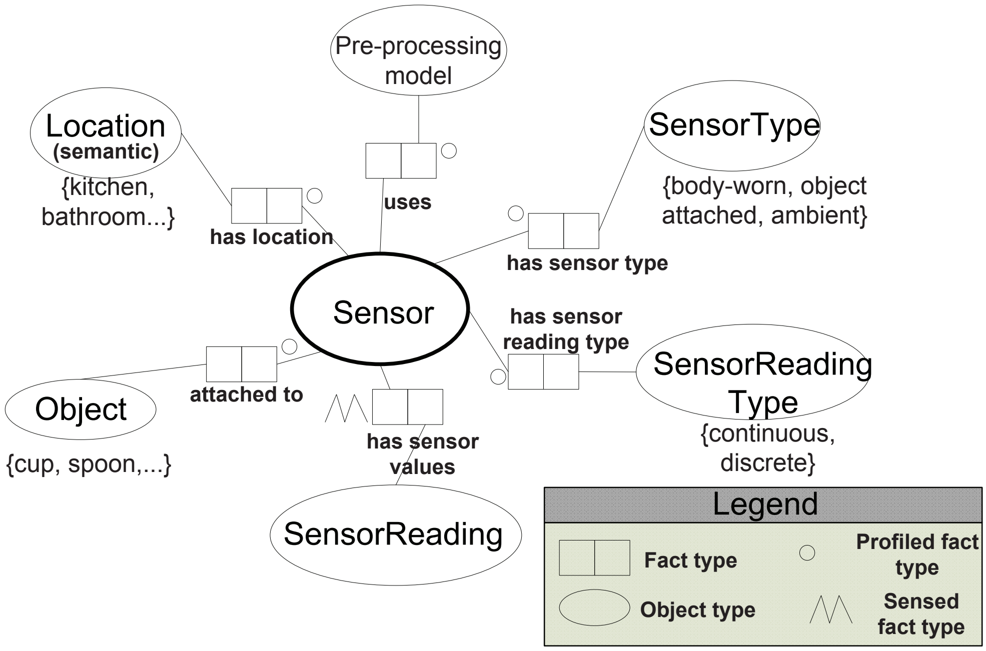


Figure 5.1: Sensor modelling.

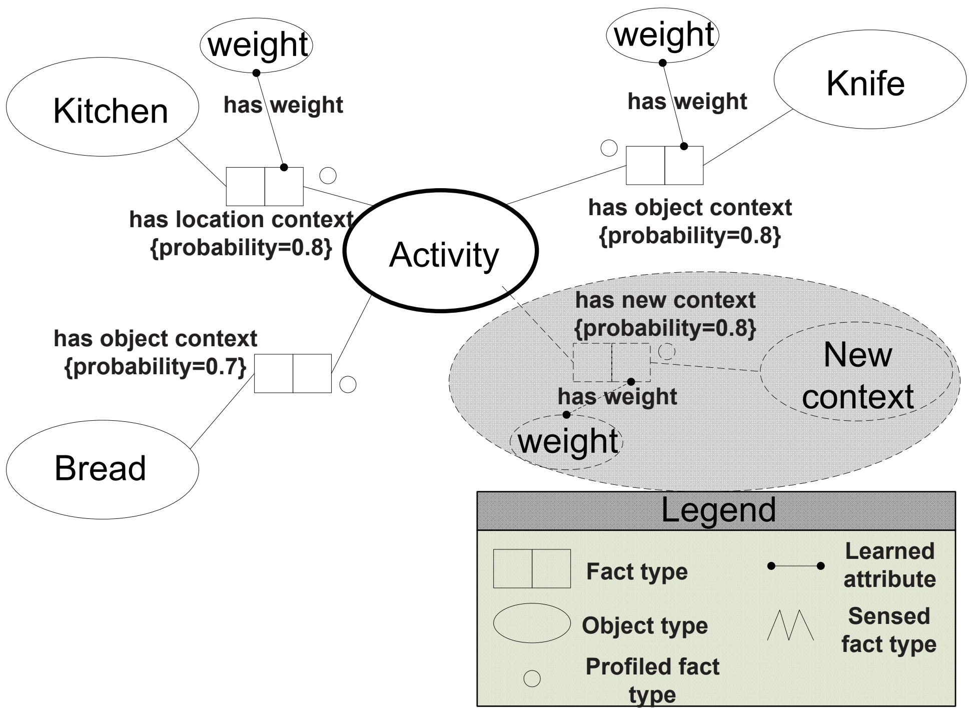


Figure 5.2: Example of an activity model.

into abstract context, location (e.g. kitchen), attached to (e.g. cup). The IEEE 1451 standard describes standard sensor interfaces through which sensors can be queried upon discovery and can present information about themselves. Based on this query a discovered sensor can be matched to its sensor context model. The context information in this model provides the guideline to pre-process the sensor readings into high-level context (e.g. interaction with objects) for activity recognition. For example, given a sensor that produces a binary output, an ambient sensor (e.g. motion sensor) may indicate the location context while an object sensor implies object usage. The pre-processing of sensor readings into the abstract context requires a pre-processing model that is different for different kinds of sensors. The pre-processing of sensor readings used in our approach is described in Section 5.6.

It is also necessary to model abstract context used for recognition of a particular human activity, so that the information provided by the dynamically available sensors can be used (after pre-processing sensor readings into the abstract context) for dynamic integration for this activity recognition. The context information model used for activity recognition can be viewed as the pattern of the activity - and activity recognition is performed to match this information with the context information derived from the sensor readings. Another motivation for activity modelling is that we can leverage domain knowledge for activity recognition and avoid the requirement of data labelling. The basic idea is that the contextual information used to describe high-level activities is human readable, and hence common sense can be used to correlate the contexts and activities as the starting point [1] (e.g. 'cooking'⇒'kitchen' AND ('walking' or 'standing')).

Figure.5.2. shows an example of activity modelling. Each fact type interrelating the activity and context is associated with a probability. The probability specifies the possibility of observing the context given the activity, and it can be manually specified with domain knowledge [142, 147] or mined from external sources [37, 139].

The result of such an adaptation is illustrated in the grey part with dash lines in Figure.5.2. When a new sensor is discovered, the sensor data it provides is pre-processed into high-level context information (i.e., the context abstraction used in the activity model). The high-level context information is then populated into the activity model and adaptation of activity recognition is performed. The details are explained in Section 5.5.3. For example, when a sensor is dynamically discovered and queried, we analyse the corresponding sensor context model, and we learn that it is an *accelerometer* attached to a *cup* and it produces *continuous* readings. Therefore, it is a sensor used for object usage monitoring, and the sensor readings need to be pre-processed (e.g. by clustering) to indicate whether the context *cup* is observed or not. The context provided by this sensor is populated into all the activities (e.g. *Make tea, Make coffee*) that relate to the context cup. The corresponding activity models are adapted by incorporating the new context and its parameters.

5.3 Problem definition

In this section, we introduce the concepts and definitions used in this chapter and then formally define activity recognition as a classification problem.

Let $L = \{(x_i, y_i)\}_{i=1,\cdots,|L|}$ denotes the set of labelled activity instances with x_i being the i^{th} feature vector and $y_i \in \{1, 2, \cdots, C\}$ being the corresponding activity class. C indicates the total number of activity classes. A feature vector x_i is the aggregation of contextual information using a sliding window, and it is formally defined as a $N-$dimension binary vector $x_i = \{x_i^1, \cdots, x_i^N\}$ with $x_i^j \in \{0, 1\}$ indicating whether j^{th} context is observed in the sliding window or not [144]. The problem is: how to recognize the set of testing instances $x \in \{0, 1\}^{1\times(N+d)}$, given the set of training data L, where d is the number of dynamically available contexts. In the later sections, we describe how to learn context weights for activities, followed by the activity recognition adaptation to the newly available context.

Let $P \in \mathbb{R}^{C\times N}$ be the probability matrix with P_{kj} defining the probability of j^{th} context given k^{th} activity. Matrix P can be mined from the knowledge database [116], learned from the labelled data [38] or even manually defined.

5.4 Knowledge-driven method

In this section, we describe how to leverage an external knowledge base to create activity models and perform the activity model adaptation with dynamically available sensors in an unsupervised manner. High-level activities are usually characterised by different kinds of contexts, (e.g. "making sandwich" can be described by location context "kitchen", and object contexts "knife" and "bread"). Moreover, there exists some descriptive texts that specify the instructions of how to perform high-level activities. Therefore, contexts characterising the activities can be extracted from the texts using natural language processing methods, and then we can calculate the parameters of the contexts with respect to different activities. Finally, with the parameters we are able to create activity models and perform activity model adaptation using the Bayesian framework. In this light, dynamically available sensors are incorporated into the activity recognition framework automatically. In what follows, we first describe the mining process of the probability matrix P from third party sources, followed by the activity modelling, prediction and activity model adaptation based on the probability matrix.

5.4.1 Knowledge base

In this section, we describe how to mine the knowledge (i.e. context-activity conditional probabilities) from the websites, www.wikihow.com and www.ehow.com [116, 156]. Both of these two websites describe how to perform daily activities and involved contexts. The basic idea of this knowledge-driven method is that the probability of observing a context in an activity is related to the probability of the textual representation of the context appearing in the textual description of the activity. We first crawl the websites and get the descriptive documents for each target activity class, and then identify the contexts involved in each activity using the natural language processing method. Finally, we calculate the context-activity conditional probability of each context with respect to different activities. The mining process can be described by the following steps:

- Search the two aforementioned websites for the target activities. As illustrated in Figure 5.3, the website lists multiple superlinks that redirect to the webpages that describe how to perform the activities step by step. We automatically crawl all the pages for each target activities. As we search for the target activities in the same website, the webpages that describe the activities have the same html schema. This makes it feasible to automatically crawl the textual descriptions.
- When we get the textual descriptions for the target activities, natural language process-

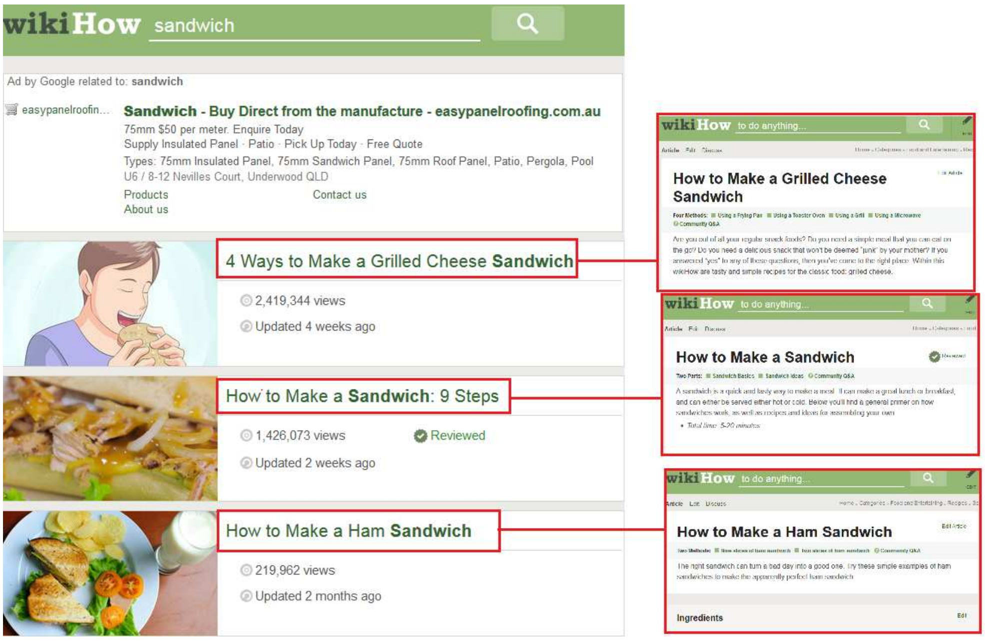


Figure 5.3: Search for activity description.

"Here is what you'll need to make a mocha coffee drink using brewed coffee" -> ['*Here*', '*is*', '*what*', '*you*', "'*ll*", '*need*', '*to*', '*make*', '*a*', '*mocha*', '*coffee*', '*drink*', '*using*', '*brewed*', '*coffee*']

Figure 5.4: Example of tokenization

ing methods are used to extract the interesting contexts from the text. The processing of the texts from the webpage goes through the following pipeline: tokenisation, part-of-speech (POS) tagging, lowercase, stemming, WordNet filtering. We first tokenise the sentences in the texts into a list of single words (shown in Figure 5.4) so that they can be further processed by later phases.

At the second step, we tag the tokenised words with part-of-speech tags as shown in Figure 5.5. Since the contexts involved in the activities are nouns, we only select those words tagged with "NOUN" for further analysis.

We then change the capital letter into lowercase and stem the morphological variants of a word that have the similar meanings to their stemmed or root forms (e.g. standing-stand, bottles-bottle). The rationale behind these two steps is that words that have different meaning or variants should have the unique representation in our case. Finally, since the contexts involved in the activities are objects or substances in the physical

[('Here', u'ADV'), ('is', u'VERB'), ('what', u'PRON'), ('you', u'PRON'), ("'ll", u'VERB'), ('need', u'VERB'), ('to', u'PRT'), ('make', u'VERB'), ('a', u'DET'),('mocha', u'NOUN'), ('coffee', u'NOUN'), ('drink', u'NOUN'), ('using', u'VERB'), ('brewed', u'VERB'), ('coffee', u'NOUN')]

Figure 5.5: Example of POS tagging

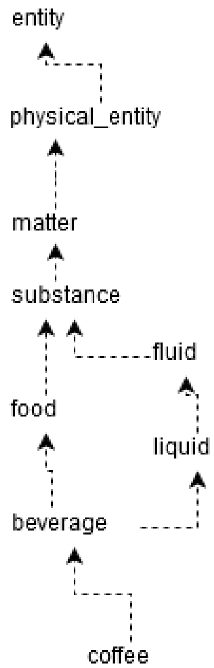


Figure 5.6: Example of hypernyms path

space, we used the knowledge base WordNet for filtering. In WordNet, each word has its hypernyms, and the relations between the word and its hypernyms follow the "is-a" relationship (e.g. coffee is-a [beverage, tree, seed, brown]). For each word, we walk through its hypernyms paths, and the word is categorised as an object or a substance if the word "object" or "substance" reside in any of its hypernyms paths. Figure 5.6 shows that "coffee" is classified as an object as there are multiple hypernyms paths walking through "substance".

- After the processing phases, we get thousands of contextual terms, some of them are not discriminative and not useful for the activity recognition task. In this step, we propose to find the top-k most important contexts for each activity class. Specifically, we calculate the term frequency-inverse document frequency (tf-idf) of each context term with respect to the activity classes as the measurement of the discriminative power, and choose the contexts for each activity class based on this measurement.

$$tf-idf_{c,y} = \frac{n_{c,y}}{\sum_c n_{c,y}} \cdot log\frac{|\{d\}|}{|\{d : c \in d\}|} \tag{5.4.1}$$

where $n_{c,y}$ is the number of occurrences of context c in activity class y. $|\{d\}|$ is the total number of collected texts describing different activity classes, and $|\{d : c \in d\}|$ is the number of texts where context c appears. The first term $\frac{n_{c,y}}{\sum_c n_{c,y}}$ denotes the frequency of the context in a specific activity class. If the context appears frequently in an activity

Table 5.1: Examples of context-activity conditional probability

1	make coffee		2	make tea		3	make pasta		4	make oatmeal	
	coffee	0.93		tea	0.89		pasta	0.85		bowl	0.69
	water	0.69		water	0.87		water	0.66		mix	0.62
	cup	0.68		cup	0.69		salt	0.61		oatmeal	0.55
	sugar	0.45		sugar	0.50		oil	0.58		sugar	0.49
	pot	0.36		leaf	0.43		sauce	0.58		oat	0.48

class y, then the whole term is larger, meaning that probability of observing the context is higher in this activity class. The second term log $log\frac{|\{d\}|}{|\{d:c\in d\}|}$ is the inverse document frequency for the context c. This is used to punish the context that is universal and appears in almost all documents, as it provides little discriminative power.

- Finally, we calculate the context-activity probability of those selected contexts with respect to different activity classes based on the processed descriptive texts. Specifically, we calculate the context-activity probability with the Naive Bayesian method. Let $P(c|y)$ be the context-activity probability (i.e. probability of context c occurring in documents that describe activity y), let $n_k(c)$ be the number of texts that describe activity class $y = k$ in which context c is observed; and let N_k be the total number of texts of that activity class. Then we can estimate the parameters of the context likelihood as,

$$P(c|y = k) = \frac{n_k(c)}{N_k} \tag{5.4.2}$$

the relative frequency of documents of activity class $y = k$ that contain context c. In practice, we use a small superparameter α for smoothing[1].

$$P(c|y = k) = \frac{n_k(c) + \alpha}{N_k + |\{c\}|\alpha} \tag{5.4.3}$$

where $|\{c\}|$ is the total number of contexts. Table 5.1 shows examples of some activity classes and the related contexts with high probabilities.

5.4.2 Activity modelling

We use the Bayesian framework to formulate the activity model, and enforce the Markov smoother on the neighbouring activity instances to encourage the same activity to be con-

[1]https://en.wikipedia.org/wiki/Additive_smoothing

tinued to avoid accidental misclassifications. Therefore, we model the joint distribution of the observed activity feature vector sequence $\mathbf{x}$ and the latent activity sequence $\mathbf{y}$.

$$P(\mathbf{x}, \mathbf{y}) = p(y_1)p(x_1|y_1)\prod_{i=2}^{I} p(y_i|y_{i-1})p(x_i|y_i) \tag{5.4.4}$$

By assuming the independences among the different contexts, we can have:

$$p(x_i|y_i) = \prod_{n=1}^{N} p(x_{i,n}|y_i) \tag{5.4.5}$$

where N is the total number of contexts that are currently available. In practice, we found that Bernoulli Naive Bayes performs better than others such as Gaussian and Multinomial Naive Byes. Therefore, the decision rule is:

$$p(x_{i,n}|y_i) = p(c_n|y_i)x_{i,n} + (1 - p(c_n|y_i))(1 - x_{i,n}) \tag{5.4.6}$$

where $x_{i,n}$ is a binary value, indicating the presence of the nth context in the ith instance as described in Section 5.3, and $p(c_n|y_i)$ is the conditional probability of nth context given activity class y_i, as described in Section 5.4.1. If context c_n is present, then $x_{i,n} = 1$ and the required probability is $p(c_n|y_i)$. Otherwise, the required probability is $1 - p(c_n|y_i)$. Therefore, Bernoulli Naive Bayes also considers the non-occurrences of the contexts.

In Section 5.4.1, we described how to leverage the external source to create the knowledge base that specifies those conditional probabilities, so that when we dynamically discover new contexts we can use those probabilities in the knowledge base for activity recognition. Suppose that there are d contexts dynamically available, then the emission probability needs to be updated to incorporate the new contexts with the probabilities from the knowledge base:

$$p(x_i|y_i) = \prod_{n=1}^{N} p(x_{i,n}|y_i) \prod_{n=N+1}^{N+d} p(x_{i,n}|y_i) \tag{5.4.7}$$

Activity prediction is equivalent to finding the latent activity sequence that is able to maximize the joint distribution, and this is can be solved with the Viterbi dynamic programming.

5.4.3 Activity prediction

Now that we have the emission probabilities (e.g. context-activity probabilities) from the knowledge base, we still need the transition probabilities among the activity classes so that we can infer the latent activity sequence on the sequence of the context observations. We manually set the transition probabilities with domain knowledge, similar as in previous work [154, 147]. The basic idea is that a human usually carries out activities for a certain amount of time, and current activity is more likely to be continued in the next time slice. Therefore, the self-transition probabilities are much higher than the probabilities of transiting one activity class to a different one. We experimentally set the self-transition probabilities to be 0.9 for each activity class, as we proved in [151] that this setting is able to achieve sufficient high accuracy:

$$p(y_i|y_{i-1}) = \begin{cases} 0.9 & y_i = y_{i-1} \\ \frac{1-0.9}{C} & \text{otherwise} \end{cases} \tag{5.4.8}$$

where C is the number of activity classes as specified in Section 5.3.

Given a sequence of context observation $x = x_1, x_2, \cdots, x_m$, the latent activity classes can be estimated by finding the corresponding latent activities of those observations, so as to maximize the joint distribution $p(x_1, x_2, \cdots, x_m, y_1, y_2, \cdots, y_m)$. To solve the problem, we define the forwarding variable,

$$\alpha_j(i) = \max_{y_1, y_2, \cdots, y_{i-1}} p(x_1, x_2, \cdots, x_t, y_1, y_2, \cdots, y_i = j) \tag{5.4.9}$$

$$s.t. \quad 1 \leqq i \leqq m \tag{5.4.10}$$

$$j \in \{1, 2, \cdots, C\} \tag{5.4.11}$$

to be the highest possibility of the ith observation being activity j, with respect to the previous $i-1$ latent activity classes. Maximising the joint distribution is equivalent to solving

$\underset{j}{max}\alpha_j(m)$. By inducing iterative relationship between the forwarding variables:

$$\alpha_j(1) = p(y_1 = j)p(x_1|y_1 = j) \tag{5.4.12}$$

$$\alpha_j(i+1) = (\max_k \alpha_k(i)p(y_{i+1} = j|y_i = k))p(x_{i+1}|y_{i+1} = j) \tag{5.4.13}$$

in each iteration, we choose the activity class that maximises the forwarding variable as the prediction:

$$y_{i+1} = \underset{j}{argmax}\ \alpha_j(i+1) \tag{5.4.14}$$

5.5 Data-driven method

The previous section describes how to mine knowledge from external sources for activity modelling and prediction. One of the advantages of this method is that training data is not required for the parameters learning, and the activity model can be modelled and adapted with dynamically available contexts without supervision. However, activity learning and adaptation with common sense general knowledge is usually not able to achieve high accuracy, due to the fact that people perform activities differently. Therefore, the activity model needs to be personalised to a specific user for high recognition performance achievement. The personalisation process takes the activity data of a specific user as input, and employs the data-driven machine learning method for learning the parameters of the activity models. The basic idea of machine learning is to find the parameters that minimise the empirical error given the training data, so it is also expected to minimise the testing error given the assumption that the user activities remain consistent during a short period of time.

To do this, we extend the activity model in Figure 5.2 and context-activity conditional probability is further associated with a weight, shown in Figure 5.2. The rationale of introducing the weight is twofold. The first one, as described previously, is that the weight can be used for personalising the activity model and obtaining optimal recognition accuracy (i.e. show how important is a particular context for an activity recognition of a particular person). Second, since margins between activity classes may change due to additional context information provided by newly discovered sensors, learning the weights from the context data will provide activity recognition adaptation to adjust the classification margins. For exam-

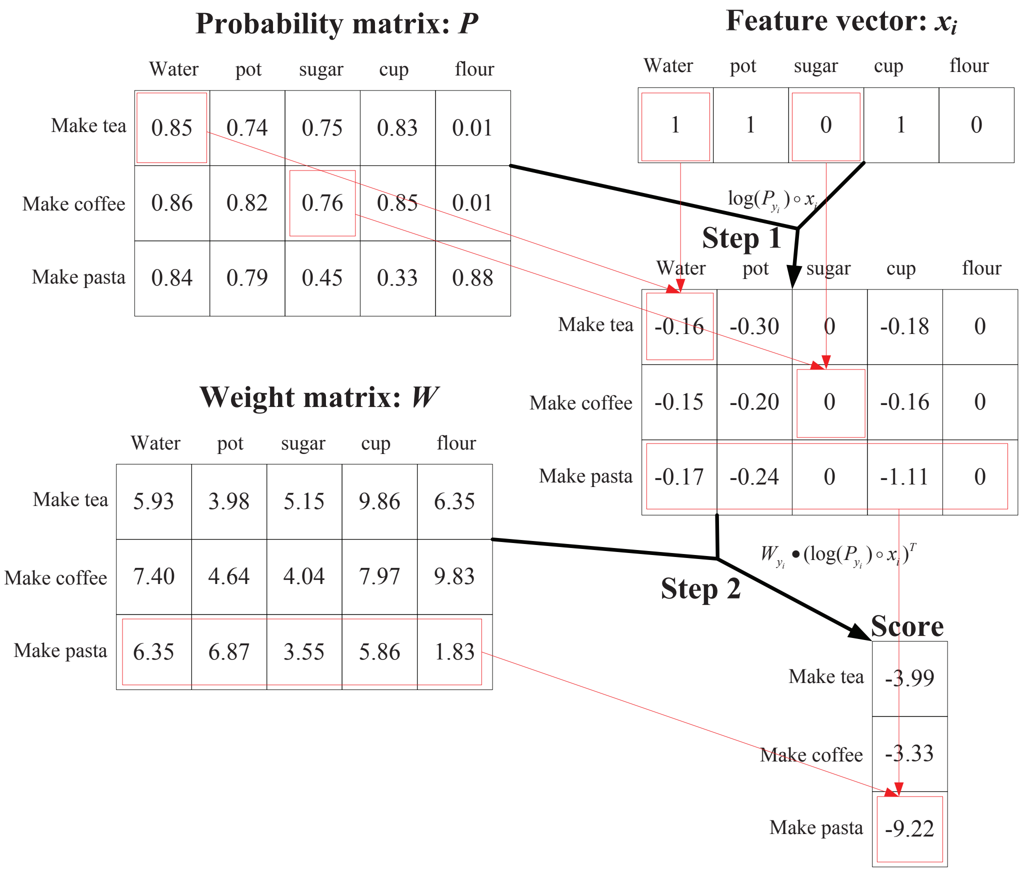


Figure 5.7: An example of the probability and weight matrices, and the recognition process of a feature vector.

ple, by mining knowledge from websites [37], we know the probabilities of *sugar* are almost the same in activities *Make tea* and *Make coffee*. However, a specific individual may use *sugar* heavily in *Make tea* and infrequently in *Make coffee*. Therefore, the weight of *sugar* in *Make tea* needs to be increased for this person activity recognition to indicate the important role it plays in the activity. In the later sections, based on the sensor and activity context models, we formulate a machine learning problem and propose methods to learn the weight matrix.

5.5.1 Activity recognition

In the modified activity model (Figure.5.2), each context probability is further associated with a weight. The activity recognition is, for a given a feature vector x_i, to calculate a score for each activity class and choose the class that has the maximum score as the prediction. To calculate the score for an activity, we first transform the feature vector x_i into the log-probability vector of the observed contexts with the Hadamard product $logP_{y_i} \circ x_i$ where $(logP_{y_i} \circ x_i)_j = logP_{y_i,j} * x_i^j$, and then linearly combine it with the weight vector

$W_{y_i} \cdot (logP_{y_i} \circ x_i)^T$, where $W \in \mathbb{R}^{C \times N}$ is the weight matrix (Figure.5.7) with W_{kj} being the weight associated with P_{kj}. These two processes are illustrated as two steps in Figure.5.7. After that, the activity that has the maximum score is chosen as the prediction:

$$prediction = argmax_{y_i} W_{y_i} \cdot (logP_{y_i} \circ x_i)^T \tag{5.5.1}$$

In this chapter, the weight matrix W needs to be learned from the dataset. There are numerous ways of obtaining matrix P described in the previous section, and we calculate P based on the available labelled data by frequency counting. Notice that the weight matrix cannot be learned using publicly available machine learning toolkits, as the input vector (i.e. the log probabilities of the observed contexts) extracted from a feature vector varies when calculating evidence against a different activity class. As illustrated in Figure.5.7, by coupling the feature vector with the probability matrix, we can obtain distinct input vectors (e.g. each row in matrix $logP \circ x_i$) for different activity classes. Also, matrix P specifies the generic relations between activities and contexts, and remains unchanged during the learning process, while matrix W maximizes the boundaries among activity classes. In this light, our method can be seen as a hybrid of the generic and discriminative model. Performing parameters learning based solely on the feature vectors x_i is a conventional machine learning method (we compare those methods with ours in the experiment section). In the next section, we describe how to use the learning-to-rank method to approach the context weight learning problem.

5.5.2 Activity learning

In this section, we draw the idea from learning-to-rank [17] to learn the weight matrix W. The learning-to-rank algorithm is commonly used in conventional recommender systems to recommend interesting items (e.g. products, friendships, point-of-interests) to the users. Specifically, it learns the *interest* of each user and the *functionality* of each item [85]. An item is recommended to a user if the item's functionality matches the user's interests. Normally, the *interest* and *functionality* are represented as latent features, and the degree of match can be measured by the inner-product of these two latent features, the higher the inner product, the more the item is preferred by the user. In the pairwise learning-to-ranking problem, the recommender system learns the latent features by ranking the preferred item higher than the less-preferred item.

Similarly in our case, the parameters (e.g. W) of each activity class can be seen as the *functionality* of the class, while the feature vector can be regarded as the *interest* of the activity instance. We compute a score against each activity class for each instance during the learning process, and the scores can be regarded as the rankings of activity classes given the particular instance. The learning process is to find the matrix W, so that for all the training instances, the correct activity classes are ranked higher than the others.

$$\begin{aligned} r(y_i, x_i) &> r(y, x_i) \\ s.t.\ (x_i, y_i) &\in L, y \in \{1, \cdots, C\}, y \neq y_i \\ r(y_i, x_i) &= W_{y_i} \cdot (logP_{y_i} \circ x_i)^T + b_{y_i} \\ r(y, x_i) &= W_y \cdot (logP_y \circ x_i)^T + b_y \end{aligned} \tag{5.5.2}$$

where b_y is the displacement variable we introduce for the case that an activity class is barely described by any contexts. Solving the above inequality is equivalent to maximizing the value of the Area Under the ROC Curve (AUC) which is commonly used in classification problems. Generally the larger the value of AUC is, the more the correct activity class ranks higher than the others. Given all the instances in the labelled training set L, the AUC value can be calculated as follows,

$$AUC = \frac{\sum_{(x_i,y_i)\in L} \sum_{y \neq y_i} I(r(y_i, x_i) - r(y, x_i))}{|L|(|C| - 1)} \tag{5.5.3}$$

where $I(.)$ is an indicator function that is equal to 1 if $r(y_i, x_i) > r(y, x_i)$ and 0 otherwise.

Learning the parameters W, b (b is the vector of $b_y, y \in \{1, \cdots, C\}$) is equivalent to maximising the AUC value. It is a common practice to introduce a differentiable function to approximate function $I(.)$ when performing the optimisation. Many approaches [18] use the sigmoid function in the form of $\sigma(x) = \frac{1}{1+e^{-x}}$, to approximate the function $I(.)$. As the result, the final object optimisation function can be derived, usually represented as a log form as follows,

$$max \sum_{(x_i,y_i)\in L} \sum_{y \neq y_i} log(\sigma(r(y_i, x_i) - r(y, x_i))) - regularization \tag{5.5.4}$$

where the *regularization* is introduced to address the problem of overfitting during the learning process, which is detailed later in this section.

Regularization

Employing the learning-to-rank algorithm facilitates the addition of different kinds of constraints. For example, adding the l_2-norm of the parameters W into the object function Eq.(5.5.4) can avoid the problem of overfitting. Moreover, the regularization term can be leveraged to perform collaborative learning (e.g. neighbouring instances are more likely to belong to the same activity class). Human activities present strong temporal relationships and the current activity is more likely to be carried out in the next time slice [147]. Therefore, when performing optimisation for an instance, we consider not only the local evidences (i.e. context observations), but also evidences from temporally adjacent instances. In this way, neighbouring instances are encouraged to have the same activity labels to smooth out the outliers. Adding these constraints into the object function, Eq.(5.5.4) can be reformulated as follows,

$$\begin{aligned} max \sum_{(x_i,y_i)\in L} \sum_{y\neq y_i} & log(\sigma(r(y_i, x_i) - r(y, x_i))) \\ & - \frac{\beta_1}{2} \sum_{(x_i,y_i)\in L} \sum_{j\in N(i)} (r(y_i, x_i) - r(y_i, x_j))^2 \\ & - \frac{\beta_2}{2} \sum_{i=1}^{C} (||W_i||^2 + ||b_i||^2) \end{aligned} \tag{5.5.5}$$

where $j \in N(i)$ indicates that instance (x_j, y_j) is temporally adjacent to instance (x_i, y_i), and β_1, β_2 control the tradeoff between the training error and regularization.

Parameter learning

In this chapter, we employ the widely used stochastic gradient descent (SGD) to learn the parameters W, b by maximising the object function Eq.(5.5.5). In the learning process, the parameters are initially randomised and then iteratively updated based on the initial labelled data. Specifically, we iteratively choose a random labelled instance, calculate the derivative of the object function and update the corresponding parameters by walking along the ascending gradient direction,

$$\begin{aligned} W &= W + \eta * \frac{\partial Obj(W,b)}{\partial W} \\ b &= b + \eta * \frac{\partial Obj(W,b)}{\partial b} \end{aligned} \tag{5.5.6}$$

where $Obj(.)$ denotes the object function parameterised by W and η is the learning rate. Given an instance (x_i, y_i), the detailed gradients of the corresponding parameters can be derived as follows,

$$\begin{aligned} \frac{\partial Obj(W,b)}{\partial W_{y_i}} &= (1 - \sigma(r(y_i, x_i) - r(y, x_i))) * (logP_{y_i} \circ x_i) \\ &- \beta_1 \sum_{j \in N(i)} (r(y_i, x_i) - r(y_i, x_j)) * (logP_{y_i} \circ (x_i - x_j)) \\ &- \beta_2 W_{y_i} \\ \frac{\partial Obj(W,b)}{\partial b_{y_i}} &= (1 - \sigma(r(y_i, x_i) - r(y, x_i))) - \beta_2 b_{y_i} \\ &(s.t.\ y \neq y_i) \end{aligned} \tag{5.5.7}$$

For the other W_y, b_y that $y \neq y_i$:

$$\begin{aligned} \frac{\partial Obj(W,b)}{\partial W_y} &= -(logP_y \circ x_i) * \sigma(r(y_i, x_i) - r(y, x_i)) - \beta_2 W_y \\ \frac{\partial Obj(W,b)}{\partial b_y} &= -\sigma(r(y_i, x_i) - r(y, x_i)) - \beta_2 b_y \end{aligned} \tag{5.5.8}$$

The iterative learning process terminates when either one of the following two conditions is satisfied: (1) the l_2-norm of the derivative of the object function is smaller than a threshold (e.g. 1e-4), and (2) the maximum number of iterative steps is reached. The pseudocode of the SGD learning process is presented in Algorithm 4. Figure.5.8 presents an example of the update of the weight matrix. Notice that the update of the parameters (W, b) only happens during the learning process, and the learning process happens in the initial activity learning and the activity model adaptation described later in the section. For illustrative purpose, we do not consider regularization items. Suppose the ground truth label of the feature vector x_i is *Make tea*. However, x_i is misclassified, as *Make coffee* achieves the highest score with current parameters, W. During the learning process, we compute the derivative of the object function with Eq.(5.5.7) and Eq.(5.5.8) (step 2), and update the weight matrix with Eq.(5.5.6) (step 3). Computing the scores for the activities with the updated parameters, we find that the score for the ground true activity is increased, while the scores for the others are decreased.

Algorithm 4 Pseudocode of the SGD learning process.

Input:
  Initial labelled data: $L = (x_i, y_i)_{i=1,\cdots,|L|}$
  Probability matrix: P
  Convergence criteria: σ
  The number of instances selected for learning in each iteration: n
Output: Parameters: W, b

```
Randomize W, b
while not converged or maximum steps not reached do
    Randomly sample n instances
    for each instance do
        update W, b with Eq.(5.5.7) and Eq.(5.5.8)
    end for
end while
return W, b
```

5.5.3 Activity recognition adaptation

When new sensors that provide new context are discovered, we need to perform adaptation of the activity model, so that the activity model can leverage the new context to potentially improve the recognition performance.

Adaptation data selection

Suppose that there are d contexts dynamically available, the feature vectors are now expanded to $N + d$ dimensions and the learning problem is now to learn the corresponding parameters $W_{y,[N+1:N+d]}$ $s.t.$ $y \in \{1, 2, \cdots, C\}$ from data. Learning the parameters requires the instances that contain the new context and their corresponding activity classes. Many previous works [131] utilise semi-supervised approaches to select high-confidence instances for model retraining. However, the high-confidence instances are less informative and have little contribution to the parameters learning. For example, in high-confidence instances, an activity class is ranked higher than the others, and $(1 - \sigma(r(y_i, x_i) - r(y, x_i)))$ is close to 0. Therefore, there is barely any gradient descend on these instances and training on them

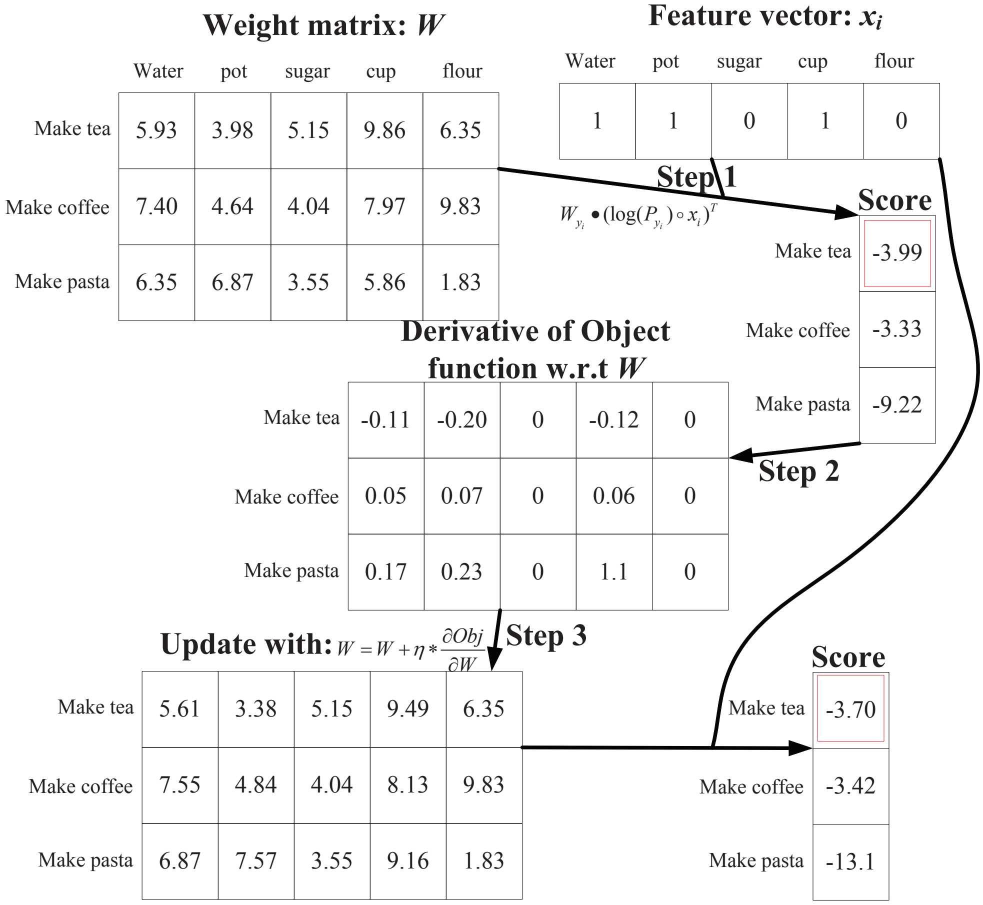
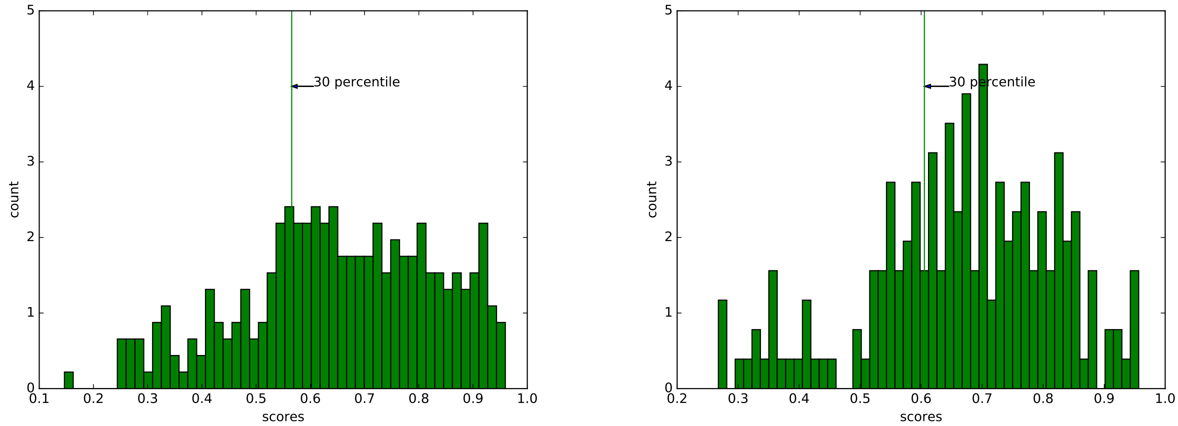


Figure 5.8: An example of the parameters learning process.



(a) Activity *coffee time* **(b)** Activity *sandwich time*

Figure 5.9: Distribution over the scores of different activities

does not minimize the training errors. Moreover, the recognition performance of the activity model retrained with the semi-supervised approach usually depends on the original model, and the accumulated error may seriously affect the recognition performance.

In this section, we employ the method proposed in Section 4.3.3 for selecting the most informative instances without supervision, so that the activity model adaptation can be performed automatically. Specifically, for each instance we calculate a score that takes into account multiple factors (e.g. posterior probability, the number of neighbouring instances that have the same predicted class label). Then we select the instances with high score and their predicted classes for activity model adaptation. However, different activity classes may have different distributions over the scores (e.g. Figure 5.9). Therefore, to maintain class balance, we set different thresholds for different classes, so that for each class all the instances having the score higher than the threshold of that classes are selected as adaptation data. The threshold can be set to certain percentile (e.g. 30, 50) of the distribution over the score for each class, so that class balance can be guaranteed.

Activity model adaptation

With the adaptation data, we perform learning to rank to obtain more accurate activity models by weighting each context. Let $Q = \{(x_j, y_j)\}_{j=1,\cdots,|Q|}$ be the adaptation dataset, where (x_j, y_j) is the j^{th} instance in the adaptation dataset that contains dynamically available d contexts: $x_j = \{x_j^1, \cdots, x_j^N, \cdots, x_j^{N+d}\}$ with $x_j^k \in \{0,1\}$. As a result, the object function in

Eq.(5.5.5) can be reformulated as follows,

$$\begin{aligned} max & \sum_{(x_i,y_i)\in L} \sum_{y\neq y_i} log(\sigma(\hat{r}(y_i,x_i) - \hat{r}(y,x_i))) \\ & + \sum_{(x_j,y_j)\in Q} \sum_{y\neq y_j} log(\sigma(r(y_j,x_j) - r(y,x_j))) \\ & - \frac{\beta_1}{2} \sum_{(x_i,y_i)\in L\cup Q} \sum_{j\in N(i)} (r(y_i,x_i) - r(y_i,x_j))^2 \\ & - -\frac{\beta_2}{2} \sum_{i=1}^{C} (||W_i||^2 + ||b_i||^2) \end{aligned} \tag{5.5.9}$$

where

$$\hat{r}(y_i,x_i) = W_{y_i,[0:N]} \cdot (logP_{y_i,[0:N]} \circ x_i)^T + b_{y_i} \ , \ (x_i,y_i) \in L \tag{5.5.10}$$

The object function can also be maximised with SGD in the same way as in the previous section, we do not detail the process here. During the learning process, the parameters (i.e. W, b) will be iteratively adjusted to discriminate one activity from another, and a certain activity class will allocate a large weight to the context that is important to the activity and a small weight to the less discriminative context. In this light the activity model adaptation process is able to automatically determine the useful context. Notice that even though we only query a small set of the activity instances for retraining, we can still leverage the unlabelled instance for temporal regularization (3^{rd} term in Eq.5.5.9).

Activity prediction: Predicting the activity class given the context observations is equivalent to solving Eq.(5.5.1). However, we include the temporal regularization into the equation and the classification of each instance also considers the classification results of neighbouring ones,

$$prediction = argmax_{y_i} \big(r(y_i,x_i) - \frac{\beta_1}{2} \sum_{j\in N(i)} (r(y_i,x_i) - r(y_i,x_j))^2 \big) \tag{5.5.11}$$

where β_1 controls the tradeoff between temporal regularization and local evidence.

5.6 Experiment

5.6.1 Public dataset

We validate the proposed methods using the OPPORTUNITY dataset [126]. The dataset contains activity data from 4 subjects when they perform Activities of Daily Living (ADLs) in a home setting. In total, 72 sensors, including 21 ambient sensors, and 14 object sensors, are deployed to monitor the activities with the sampling rate of 30/second. The activities of the user in the scenario are annotated on different levels, including locomotion (e.g. standing), gesture (e.g. opening), high-level activities (e.g. Coffee time). For object and ambient sensors, even though there are several sensors of the same type, they are used for monitoring different contexts. Therefore, there is 1-to-1 correspondence between contexts and sensor readings for those object and ambient sensors. For the other wearable sensors used to recognize low-level locomotions (e.g. standing), we treat them as a group and use their readings to recognise the low-level locomotive contexts (e.g. standing) and use recognition results as inputs for high-level activity recognition (e.g. Coffee time). The reason is that wearable sensors produce different readings when they are fixed in different body positions with different orientation. As a result, the sensor readings they produce do not have semantic meanings (in contrast to low-level activities), so it is impossible to apply domain knowledge on these readings.

Each of the four subjects (Subjects 1, 2, 3 and 4 are represented as S1, S2, S3 and S4) performs the ADLs for 5 runs. In each run, the subjects are instructed to perform the activities with a high-level script and are encouraged to perform the activities in an usual way with all the variations they are used to. In our demonstration of the effectiveness of the proposed method for recognising high-level complex activities we use data for 3 subjects, not 4. We do not include the data of the 4th subject as rotational noises have been artificially added and therefore the sensor data does not represent the activity data captured in realistic scenarios [16]. We use a sliding window of 5 seconds with 50% overlap to segment the streaming data, and the data description is presented in Table.5.2. The window length of 5 seconds is a tradeoff between delay and recognition performance, and examining the influence of the window length is out of scope for this thesis.

Table 5.2: Dataset description.

Datasets	Activities (Instances)
S1	Cleanup (283), Coffee_time (376), Early_morning (283), Relaxing (100), Sandwich_time (576), null (380)
S2	Cleanup (274), Coffee_time (273), Early_morning (216), Relaxing (120), Sandwich_time (749), null (290)
S3	Cleanup (205), Coffee_time (279), Early_morning (369), Relaxing (167), Sandwich_time (507), null (229)

Table 5.3: Simulation activity classes.

1	Make coffee	7	Brush teeth	13	Clean table
2	Make tea	8	Wash clothes	14	Play PC games
3	Make pasta	9	Make orange juice	15	Watch TV
4	Make oatmeal	10	Watch DVD	16	Put on make-up
5	Fry eggs	11	Take pills	17	Use toilet
6	Make phone call	12	Read books		

5.6.2 Simulation dataset

We also manually generate sensor data for the validation of the knowledge-driven method. We generate sensor data for commonly performed daily activities that are listed in Table 5.3. The generation of the sensor data is based on the context-activity probability matrix P. We assume that there is 1-to-1 correspondence between the sensors and context the same as in previous works [154, 147, 116, 95, 37]. Algorithm 5 describes the generation process.

The prior distribution of the activity classes is proportional to the number of descriptive texts that we can crawl from the websites for each activity class. In Algorithm 5, we first generate the class label $Data_{i,-1}$ for i^{th} instance, and then generate context presences for that instance. Specifically, for each context j, the presence of that context in activity class $Data_{i,-1}$ is drew from the Bernoulli distribution $Bernoulli(P_{Data_{i,-1},j})$. We use the Bernoulli distribution for generating the context presence as it has been demonstrated in previous work [99] that the real sensor event distribution is Bernoulli distribution parameterised with the firing probability. Due to the assumption of the 1-to-1 correspondence between sensors and contexts, drawing the context presences is equivalent to generating the sensor firings (i.e. sensor events).

Algorithm 5 Simulation data generation process.

Input:

prior: activity classes prior distribution

P: context-activity conditional probability matrix, P_{kj} stands for the probability of observing j^{th} context in k^{th} activity class

N: the total number of generated activity instances

Output:

Data: the generated sensor data, where $Data_i$ represents the i^{th} instance, $Data_{i,-1}$ is the corresponding class label and $Data_{i,j} \in 0,1$ indicates the presence of j^{th} context in i^{th} instance.

```
for i = 1 to N do
    Draw the activity class from Dirichlet Distribution parameterized with prior:
    Data_{i,-1} ~ Dirichlet(prior)
    for each context j do
        Draw the presence of the context from Bernoulli distribution parameterized with
        P_{Data_{i,-1},j}: Data_{i,j} ~ Bernoulli(P_{Data_{i,-1},j})
    end for
end for
return Data
```

5.6.3 Validation of knowledge-driven method

In this subsection, we describe the validation of the knowledge-driven method for activity recognition and adaptation. Specifically, we demonstrate the possibility of incorporating dynamically discovered contexts for activity recognition. We first introduce the validation method, followed by an illustrative example, and finally the experimental results.

Validation method

For the OPPORTUNITY dataset, we do not recognise gesture contexts, as they are highly correlated with object contexts [147], and they are difficult to recognise based solely on the wearable sensors [16]. Therefore, each instance consists of 27 features, including locomotion, object and ambient contexts. For the binary sensors, the produced values are used directly as features (i.e., pre-processing of sensor readings into context is not needed). For object and ambient sensors that produce continuous values and these values require pre-processing to achieve context information that can be used for activity recognition, we use K-means to cluster the standard deviation of the sensor values into 2 components, indicating whether the sensors are triggered or not. Therefore, even though those sensors produce continuous

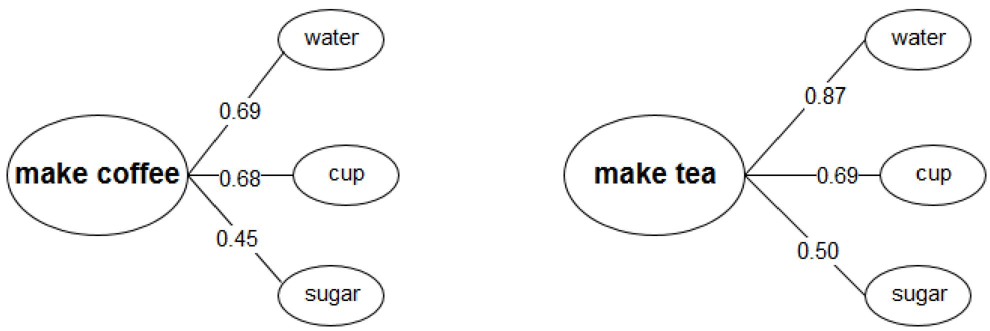


Figure 5.10: Original activity model of *make coffee* and *make tea*

readings, they are treated as binary sensors in a logical sense. The different preprocessing methods are the results of sensor heterogeneity and motivate the sensor modelling.

For the simulation dataset, we use the tf-idf for selecting the 10 most significant contexts for each activity class as described in Section 5.4.1, and that results in 149 contexts for all the activities (some activities share common contexts).

To validate the feasibility of incorporating new contexts for activity recognition, for the OPPORTUNITY dataset, we create two activity models. The first activity model contains X contexts (X is a parameter and it is varied in our experiment), while the second model also contains dynamically available contexts in addition to the original X contexts. The X contexts are randomly sampled and this process is repeated 50 times to avoid biases. The average recognition performance of these two activity models is compared in this experiments, and the second model is supposed to have better performance as it contains additional contexts. The same validation method is applied to the simulation data, except that we generate separate datasets for the two activity models. Notice that activity modelling and prediction are introduced in Section 5.4.2 and Section 5.4.3, respectively.

Example

The following example illustrates the process of incorporating dynamically available context for activity recognition and adaptation. Suppose we have the activity class *make coffee* and *make tea*, characterized by contexts *cup*, *water*, and *sugar* with different probabilities as shown in Figure 5.10.

For the feature vector (shown in Figure 5.11) where all the current available contexts *cup*,

	water	cup	sugar
x_i	1	1	1

Figure 5.11: An example of feature vector with context *water*, *cup* and *sugar*

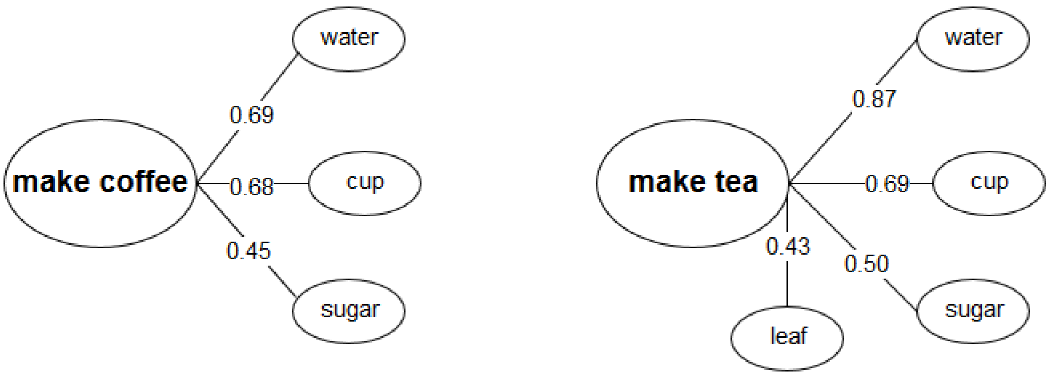


Figure 5.12: Activity models with additional context *leaf*

	water	cup	sugar	leaf
x_i	1	1	1	1

$p(\text{make coffee}|x_i) = 8.26\text{e-}3$
$p(\text{make tea}|x_i) = 1.29\text{e-}1$

Figure 5.13: An example of feature vector with additional context *leaf* present

	water	cup	sugar	leaf
x_i	1	1	1	0

$p(\text{make coffee}|x_i) = 2.03\text{e-}1$
$p(\text{make tea}|x_i) = 1.71\text{e-}1$

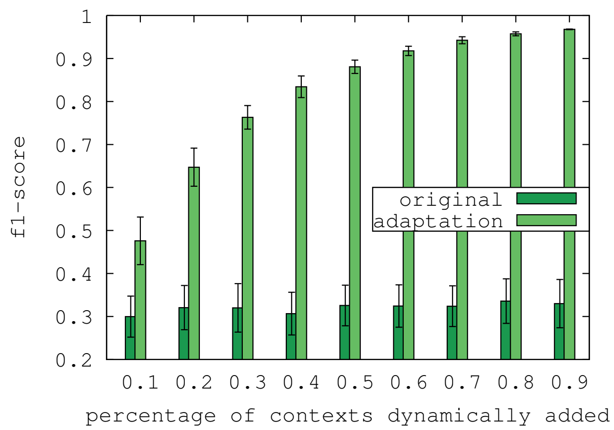
Figure 5.14: An example of feature vector with additional context *leaf* not present

water and *sugar* are present, it is always recognized as *make tea* since *make tea* has high posterior probability calculated with the parameters and the feature vector. As *make coffee* is also characterized by those three contexts with similar probabilities, misclassification occurs when the user is actually carrying out the activity *make coffee*.

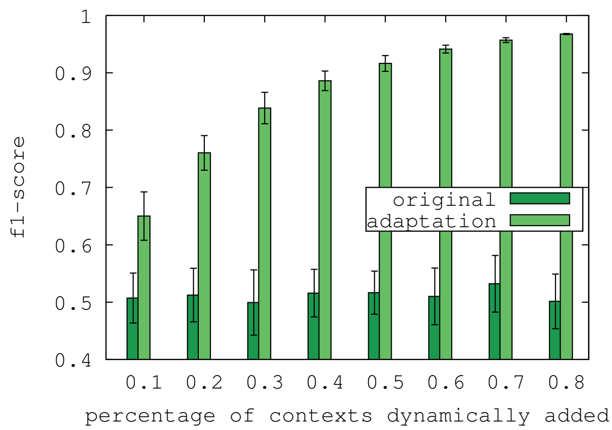
Suppose now we dynamically discover a sensor and it provides the context *leaf* that is used to characterize activity *make tea*, then the activity model can be adapted with the parameters mined from the website as shown in Figure 5.12. The context *leaf* provides additional information that is discriminative enough to differentiate *make coffee* from *make tea*. For example, feature vector where the context *leaf* is present is classified as *make tea* (shown in Figure 5.13), while feature vector where the context *leaf* is not present is classified as *make coffee* (shown in Figure 5.14).

Experiment results

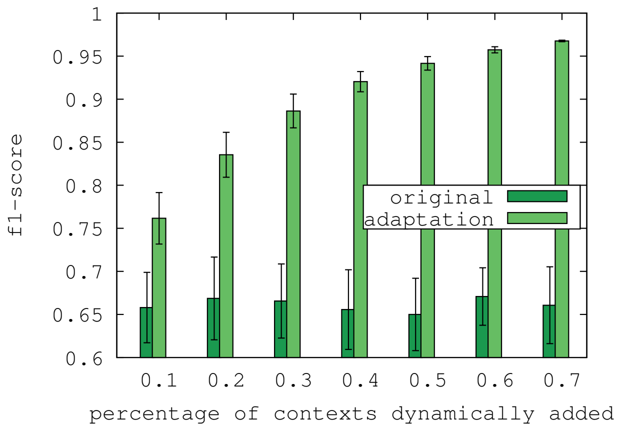
The experiment results are presented in Figure 5.15. Notice that in the simulation data, we create one dataset for the original activity model that only contains certain percentage of the contexts, and a second dataset for the adaptation activity model that contains new contexts in addition to the original ones. The percentage of the contexts in the first dataset is varied from 10% to 90% (shown in Figure 5.15a∼ Figure 5.15i). Also, the percentage of contexts that are dynamically available is also varied from 10% till all the remaining contexts are added. As for the OPPORTUNITY dataset, the number of contexts in the original activity model is set to 24, so the number of dynamically available contexts is 3.



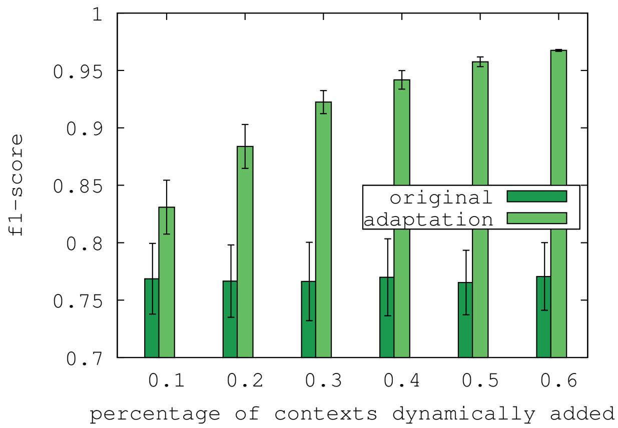
(a) First dataset contains 10% of all contexts



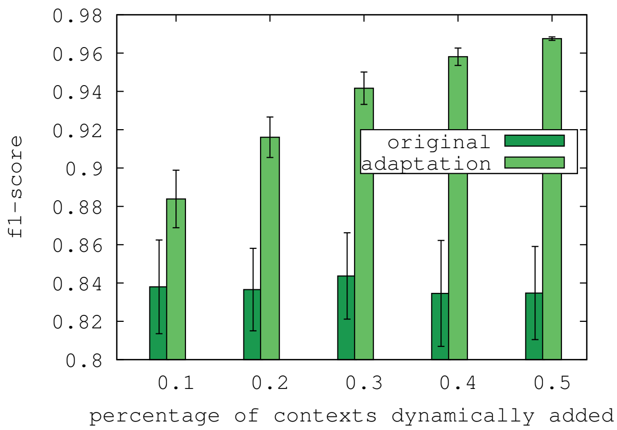
(b) First dataset contains 20% of all contexts



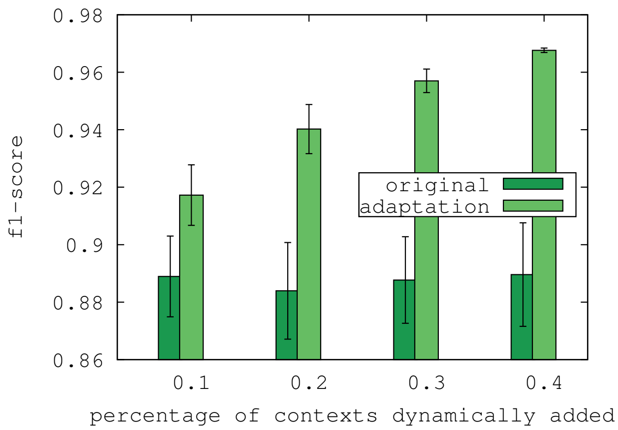
(c) First dataset contains 30% of all contexts



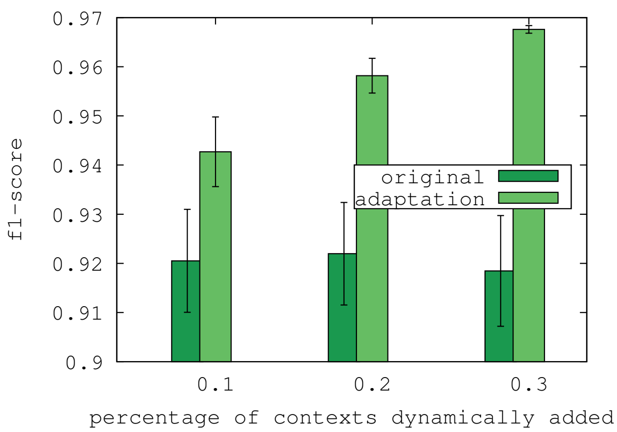
(d) First dataset contains 40% of all contexts



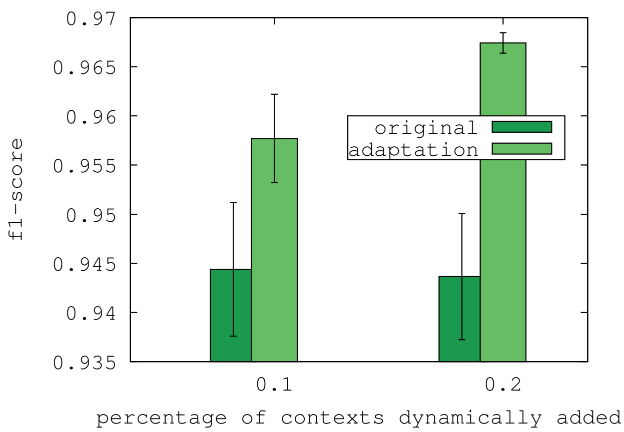
(e) First dataset contains 50% of all contexts



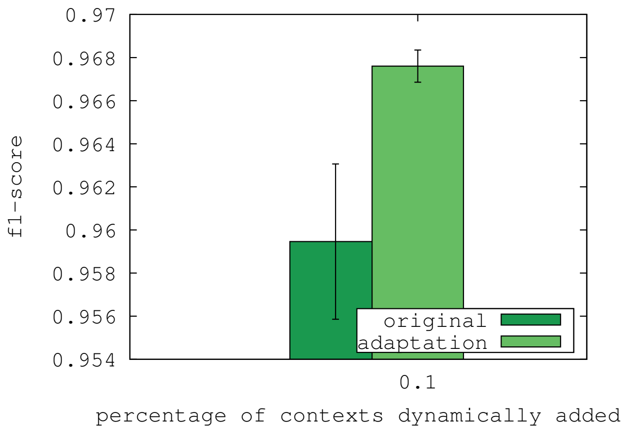
(f) First dataset contains 60% of all contexts



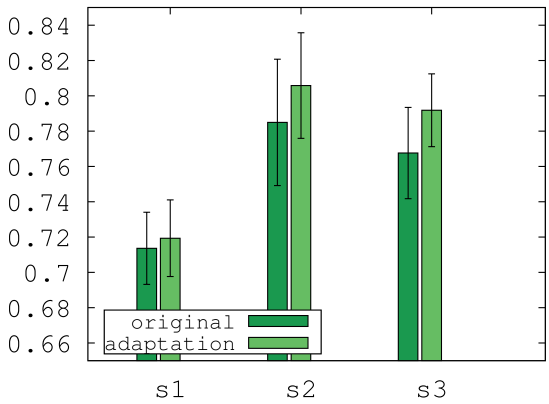
(g) First dataset contains 70% of all contexts



(h) First dataset contains 80% of all contexts



(i) First dataset contains 90% of all contexts

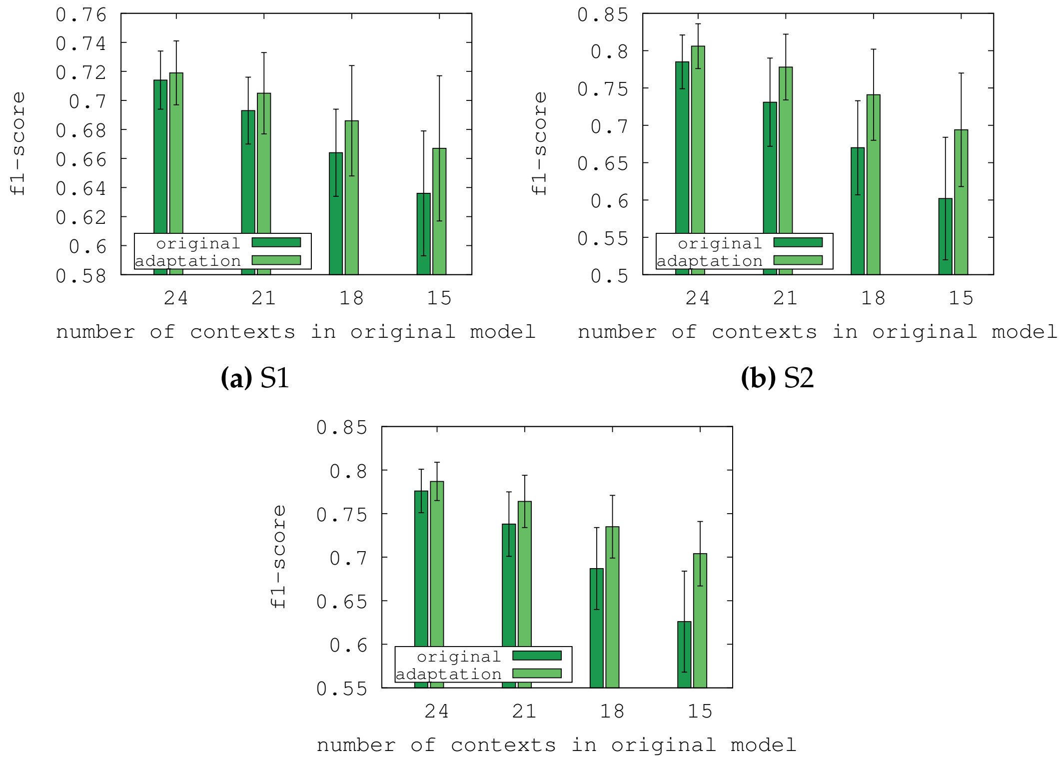


(j) OPPORTUNITY:$p = 24$

Figure 5.15: Experiment results on simulation data and OPPORTUNITY

From the figures, we can see that we are able to improve the recognition performance by incorporating new contexts dynamically using the knowledge from the websites without supervision. The amount of accuracy improvement is proportional to the number of dynamically available contexts, as more contexts provide more information for the activity models. Also, we are able to achieve high accuracy (97%) using all contexts as shown in the figures, and we believe the reasons are two-fold. First, the accuracy depends on the types of activity classes we are to classify. As most of the activities listed in the previous table are characterised by distinguished contexts, their distinct activity patterns make them easy to recognise. Second, in the experiment we used the context-activity probability matrix to generate the simulation data and then used the probabilities to create the Bernoulli Naive Bayesian activity model. Therefore, the dataset contains a certain amount of biases.

We can also observe that experiments on OPPORTUNITY shows a little recognition performance improvement (1%$\sim$ 3%). This is because OPPORTUNITY is a realistic activity dataset that contains a lot of activity patterns variants. Therefore, it is difficult to recognise the activities, and the extreme example can be seen on the first subject that only experiences 1% f1-score improvement. In addition, we only dynamically incorporate 3 contexts, some of which may not be discriminative enough to improve the recognition performance significantly. To validate this assumption, we vary the parameter X from 24 to 15, and present the result in Figure 5.16. From the figures we can observe that, with lower X we experience larger recognition performance gain, as decreasing X means that we incorporate more contexts dynamically. However, the recognition performance of the activity model suffers as we lower X. This is expected as less discriminative information is available for the original ac-



(a) S1 (b) S2 (c) S3

Figure 5.16: Vary X from 24 to 15 in experiments with OPPORTUNITY

tivity model. The marginal f1-score improvement on dataset OPPORTUNITY demonstrates that the general knowledge cannot be personalised to a specific user for achieving high accuracy, and this inspires us to use data-driven method so that the activity model can be adapted to a specific user with his/her own activity data.

5.6.4 Validation of data-driven method

Validation method

In this section, we use the OPPORTUNITY dataset for validating the data-driven method described in Section 5.5. The preprocessing of the dataset is the same as that described in Section 5.6.3. Notice that, we do not demonstrate the data-driven method with the simulation data. The reason is that we aim to demonstrate that the data-driven method is able to personalise and adapt the activity model to a specific user for achieving high accuracy. However, the simulation data is generated using general knowledge, and it does not represent activity data of specific real users.

We perform leave-one (run)-out cross validation (LORO-CV) on each dataset. Specifically, one run of the data is left out for validation, and 50% of the left runs are used as the initial training data to create the initial activity model. Classification is performed on the other 50% of the remaining runs and a certain percentage of those data is selected as the adaptation dataset. Notice that in Section 5.5.3, we introduce an adaptation data selection method that computes a score for each classified instance and selects the instances scored higher than a threshold for the adaptation. The threshold is set to certain percentile (e.g. 30 percentile) of the distribution over the scores of each class. Finally, the model is validated on the left out run. Notice that the rationale of choosing the LORO-CV rather than commonly used 10-fold-CV is threefold. First, the temporal information preserved in LORO-CV can be used for regularization both in the training and testing process. Second, the testing process in LORO-CV classifies the testing instances sequentially, this is more similar to the real-time activity recognition. Finally, in 10-fold-CV, the data in training set and testing are correlated to some extent as the data streaming is segmented with a 50% overlapped sliding window. Therefore, 10-fold-CV does not reflect the real performance of classifier [42].

To emulate the impact of incorporating new sensors (and the context that can be derived from their readings) we first use a subset of the original OPPORTUNITY dataset, and the remaining portion of the dataset is used to emulate sensor readings from newly discovered sensors. In other words, we perform leave-n (contexts)-out cross validation, where the instances in the initial training data contain information of $(27 - n)$ contexts, while the instances in the selected adaptation dataset contain information about all the 27 contexts. This cross validation process can be seen as: we recognise the activity with X (i.e. $27 - n$) contexts, and then the same activities are recognised with potentially better accuracy with $X + Y$ (i.e. 27) contexts, where Y (i.e. n) new contexts are provided by newly incorporated sensors. This kind of cross validation is commonly used in zero-shot learning [23]. The description of the cross validation is presented in Table.5.4.

All the recognition results are presented in the form of f1-score (f1-score=$\frac{2\text{*precision*recall}}{\text{precision+recall}}$).

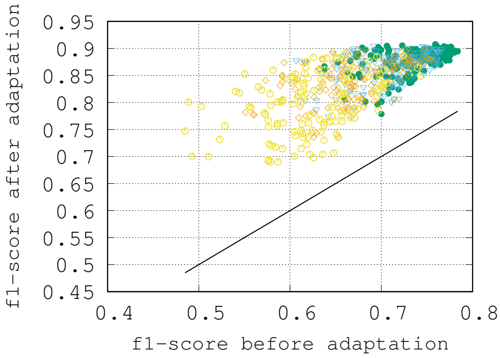
Table 5.4: Cross validation description.

Dataset	Composition	Description
initial training set	50% of the remaining runs	the 50% data is randomly sampled, each instance contains information about (X) contexts, X is varied from 24 to 15.
adaptation dataset	classified instances scored higher than the threshold (e.g. 30 percentile of the distribution over the scores of each class), each instance contains information of the 27 contexts	
validation dataset	one run	

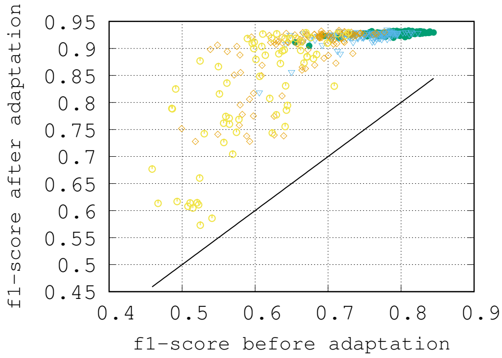
Impact of adaptation

We study the f1-score gain after incorporating dynamically available context in this experiment. The threshold for selecting the adaptation is set to 30 percentile of the scores of the classified instances, and the number of contexts in the initial dataset, X, is varied from 24 to 15. For a given X, we randomly sample X contexts and repeat this process for 200 times to avoid biases. As a result, each point in Figure.5.17 represents a round of training, adaptation and validation. The X-axis of each data point is the f1-score before the model adaptation, and Y-axis is the f1-score after the adaptation. Therefore, any data points on the right side of the line $f(x) = x$ indicate that there is f1-score improvement after the adaptation in these rounds of experiments, larger distance from the line means greater improvement.

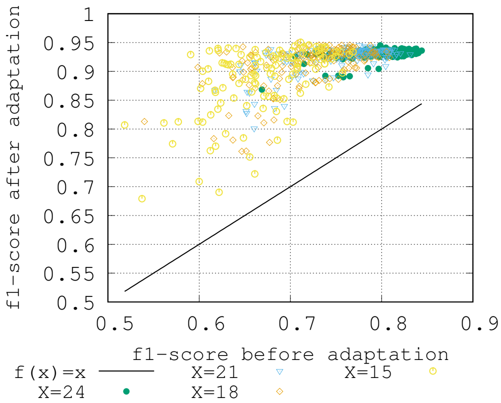
Figure.5.17 shows that incorporating dynamically available context to adapt activity model is able to increase the recognition performance. Generally, the f1-scores of setting $X = 24$ are more stable, while f1-scores of setting with smaller X become more scattered. The underlying reason is that the f1-score improvement depends on the discriminative power of the dynamically available contexts. Therefore, integrating more contextual sources dynamically will provide diversified discriminative information, and results in a more diversified f1-score gain. Basically, the more contexts are incorporated, the higher improvement of recognition performance is expected. To validate, Figure.5.18 presents the CDF (cumulative distribution function) of the f1-score gain across the datasets. It can be seen from the



(a) S1



(b) S2



(c) S3

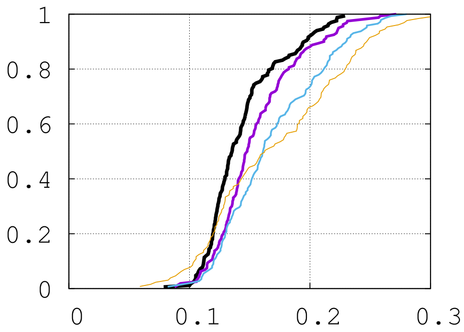
Figure 5.17: F1-score before and after adaptation across the datasets

figure that incorporating more contexts for adaptation can obtain more f1-gain in general than fewer contexts . Take S3 for example, to achieve f1-score gain more than 20%, the probability is 60% if we incorporate 12 contexts ($X = 15$). By contrast, the probability is 40% and 20% if we incorporate 9 ($X = 18$) and 6 contexts ($X = 21$) respectively.

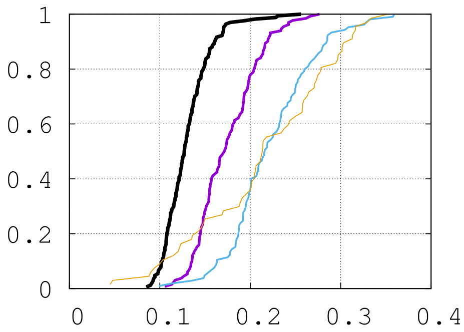
Impact of adaptation data

In this subsection, we examine the impact that the size of adaptation data has on the recognition performance. Figure.5.19 shows the impact of the amount of adaptation data on the f1-score across the datasets in different scenarios (i.e. number of leave-out contexts). The x-axis represents the number of contexts in the initial train set and the y-axis stands for the f1-score after adaptation. For each $X \in \{15, 18, 21, 24\}$, the threshold for selecting the adaptation data is varied from 30 percentile to 90 percentile of the scores of the classified instances. Higher threshold means less adaptation for retraining.

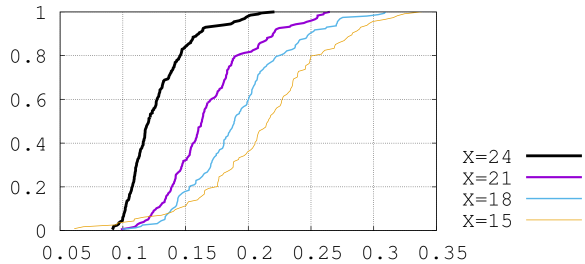
From the figures, we can draw the following conclusions. Firstly, activity models with fewer



(a) S1

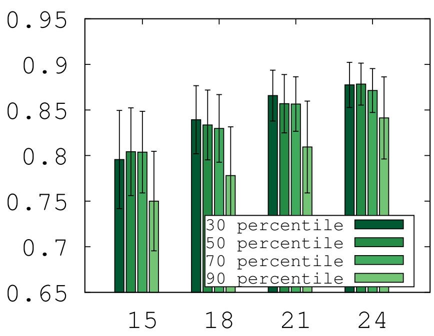


(b) S2

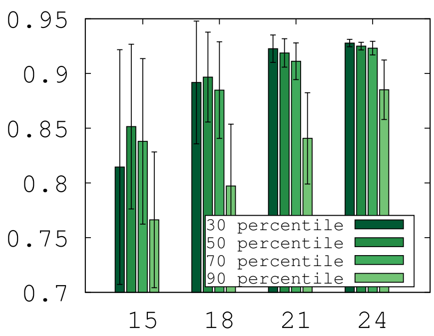


(c) S3

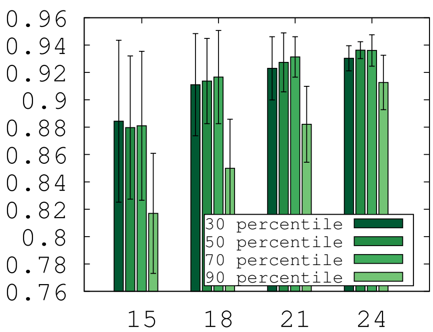
Figure 5.18: CDF of the f1-score across the datasets.



(a) S1



(b) S2



(c) S3

Figure 5.19: The impact of the amount of adaptation data on the recognition performance (measured with f1-score).

contexts initially are more sensitive to the amount of adaptation data, as shown in the standard deviation of the results. According to Vapnik's theory [145], the testing error of a classifier is upper bound by the testing error plus a term that is proportional to the complexity of the classifier and inversely proportional to the amount of training samples. Incorporating more contexts means that the activity models need to estimate more parameters, and hence increases the complexity. Therefore, the amount of adaptation data becomes critical to the testing error, and increasing the amount of adaptation set will lower the testing error dramatically.

Secondly, activity models with more contexts initially perform better than those with fewer contexts. This is because activity models trained with an initial train set that contains more contexts are able to yield higher recognition performance, and they can predict the instances with higher accuracy and select the correctly predicted instances for adaptation.

Finally, there is no significant difference in the recognition performance when we vary the threshold from 30 percentile to 70 percentile. However, the f1-scores drop sharply when we set the the threshold to 90 percentile. The reason is that we do not have sufficient adaptation data with high threshold. Therefore, training the parameters with insufficient data results in overfitting and suboptimal activity models.

Influence of regularization weight

In this subsection we study impact of the temporal regularization. The temporal regularization term β_1 in Eq.(5.5.9) controls the tradeoff between the local contextual information and the information from neighbouring instances. It is involved in both of the learning and prediction processes. Figure.5.20 illustrates the f1-score as a function of the temporal regularization weight which is varied from 0 to 0.4. As shown in Table 5.5, the threshold for selecting adaptation data is 30 percentile of the scores, and the number of initial contexts is set to 24. We do not present the results of other settings (i.e. $X \in \{15, 18, 21\}$, threshold = 50,70,90 percentile) here as they present the similar trend. From the figure we can see that by putting more weight on the pairwise evidence from neighbouring instances, we are able to smooth out the accidentally mis-classified instances and improve the overall f1-score. After a certain threshold (e.g. 0.2), the recognition performance becomes stable. This figure also

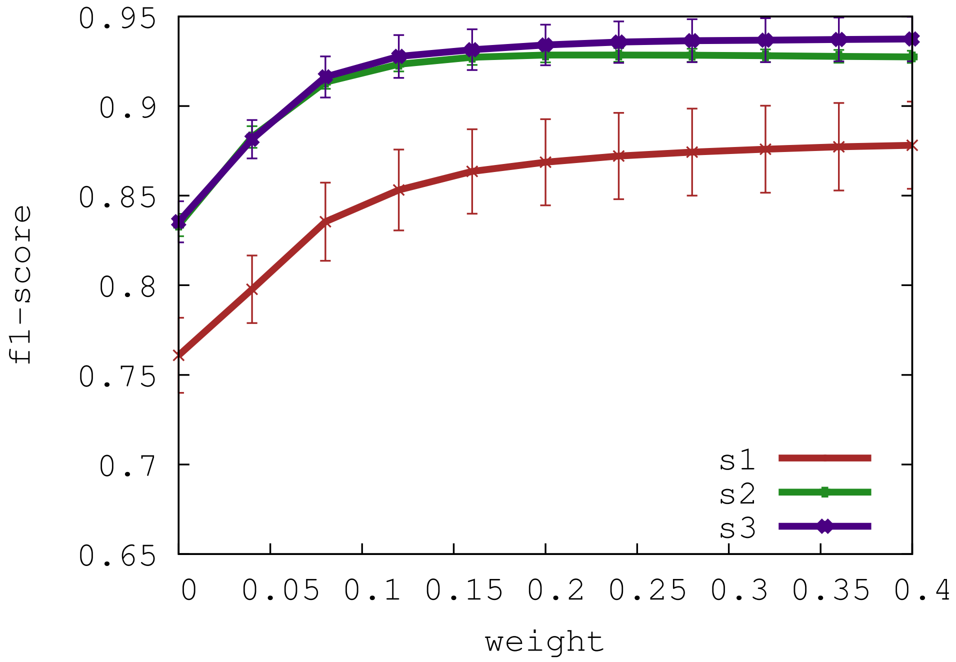


Figure 5.20: The f1-score as a function of the temporal regularization weight.

shows that it is easy to specify the regularization weight-setting: the weight of any values larger than 0.2 can obtain the optimal result.

Table 5.5: Parameters setting description.

Parameters	Value
Threshold for selecting the adaptation data	30 percentile of scores of the classified instances
Number of contexts in the initial train set	X=24

Comparison with conventional methods

In this subsection, we compare the proposed method with the conventional machine learning methods. All the other settings (e.g. training set, validation set, method of cross-validation) are the same except that the learning process is performed based on the input feature vector x_i described in Section 5.3. During the adaptation process, the instances in the initial training set $x_i = \{x_i^1, \cdots, x_i^N\}$ are extended to $x_i = \{x_i^1, \cdots, x_i^N, 0^{N+1}, \cdots, 0^{N+d}\}$ to guarantee a common classifier is trained.

The baselines introduced for the purpose of comparison include: SVM (support vector machine), RF (Random Forest) and LR (logistic regression). The parameters for those classifiers are obtained through the grid search cross validation, as shown in Table 5.6. The threshold for selecting the adaptation data is set to 30 percentile and the comparison results across the datasets are presented in Figure.5.21.

Table 5.6: Parameters for baselines.

Classifier	Parameters
SVM	kernel=RBF, $\gamma = 0.01$, $C = 10e4$
RF	$n\_estimators = 100$
LR	$C = 10e4$

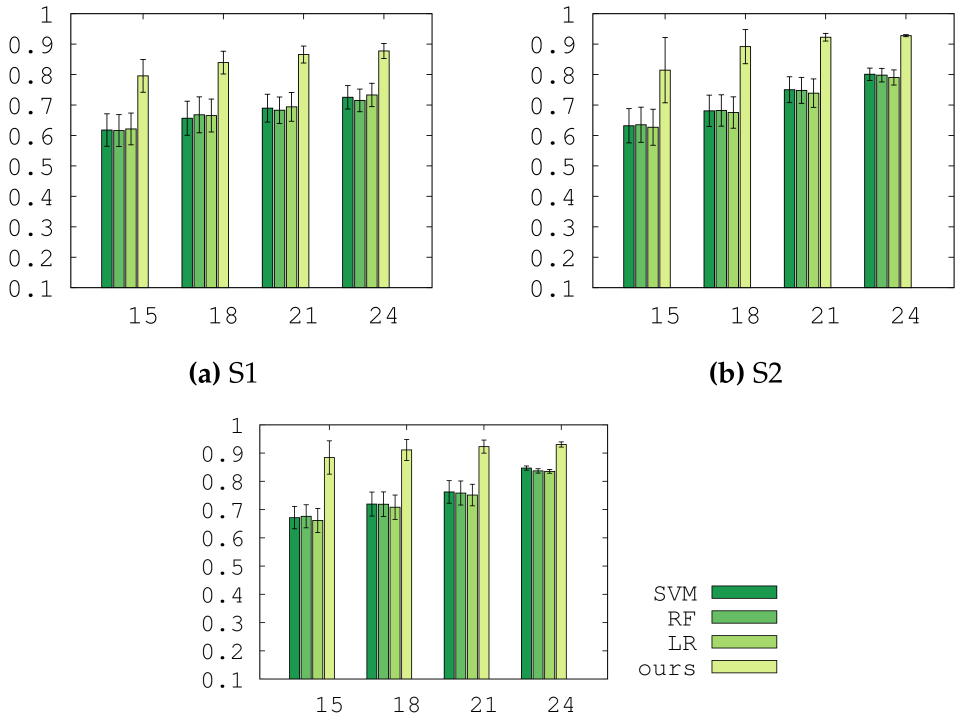
Figure.5.21 shows that the proposed method outperforms all the baselines with a significant margin. We also vary the number of contexts in the initial dataset from 24 to 15 for each dataset. On average, our method is 16.5% (max: 17.4%, min: 14.4%), 17.2% (max: 21.0%, min: 12.7%) and 16.1% (max: 20.8%, min: 8.34%) higher than the second best baseline on dataset S1, S2 and S3 respectively, in terms of f1-score. These experiments demonstrate the advantage of our method in performing adaptive activity learning, and we believe the underlying reasons are twofold. Firstly, embedding the temporal regularization into the learning and prediction processes enables the proposed method to effectively leverage the temporal characteristic of human activities for obtaining desired predictive outcomes. Secondly, the weighted model in our method learns a weight for each context probability. Therefore, it encodes domain knowledge into the activity model and is equivalent to feature transformation to some extent.

Comparison with hybrid classifiers

It seems unfair to compare our method with the conventional machine learning methods, as they make prediction for each instance independently and do not consider the temporal

Table 5.7: Comparison with hybrid classifiers.

Classifiers	S1				S2				S3			
	Number of contexts in the initial train set											
	X=15	X=18	X=21	X=24	X=15	X=18	X=21	X=24	X=15	X=18	X=21	X=24
SVM+HMM	0.62	0.662	0.697	0.735	0.635	0.685	0.759	0.814	0.675	0.728	0.775	0.858
RF+HMM	0.621	0.658	0.689	0.722	0.637	0.686	0.752	0.803	0.68	0.725	0.765	0.846
LR+HMM	0.631	0.674	0.71	0.752	0.631	0.683	0.751	0.805	0.672	0.723	0.769	0.85
SVM+CRF	0.626	0.668	0.703	0.742	0.64	0.691	0.764	0.82	0.685	0.737	0.783	0.866
RF+CRF	0.627	0.665	0.697	0.732	0.64	0.691	0.76	0.812	0.692	0.737	0.778	0.857
LR+CRF	0.633	0.675	0.711	0.753	0.634	0.688	0.757	0.812	0.679	0.731	0.778	0.86
Ours	0.796	0.839	0.866	0.877	0.815	0.892	0.923	0.928	0.884	0.911	0.923	0.93



(a) S1 **(b)** S2 **(c)** S3

Figure 5.21: The impact of the amount of adaptation data on the recognition performance (measured with f1-score).

information of neighbouring instances. Actually, the previous work [16, 151] proposes to combine typical classifiers with graphical models (e.g. HMM, CRF) to smooth out the outliers. As those graphical models make assumptions about the dependency among the latent activity classes of the instances, and the classification of one instance also considers the predictive outcomes of neighbouring ones, we perform a comparison between our method and those hybrid classifiers.

The threshold for selecting adaptation data is 30 percentile of the scores, and the number of initial contexts is varied from 24 to 15. Table.5.7 presents the comparison results. From the table we can see that our method still outperforms the best baseline in terms of f1-score. Specifically, when compared with the best HMM-hybrid conventional classifier, the average f1-score advantage is 15.29% (S1), 16.55% (S2) and 15.22% (S3) respectively. The average advantage is 15.16% (S1), 16.04% (S2) and 14.28% (S3) when compared with the best CRF-hybrid classifier. As has been confirmed in previous work [144], CRF performs slightly better than HMM. Notice that even though we only query limited amount of the labelled data in the adaptation phase, we are still able to leverage the neighbouring unlabelled instances for temporal regularization (Eq.(5.5.9)).

5.7 Summary

In this chapter, we address the problem of adaptive high-level activity recognition with dynamically available sensors. The existing research shows that additional contextual information can potentially improve the recognition accuracy, and sensor addition or replacement is very common in activity recognition systems. Therefore, it is extremely important that the activity recognition framework is able to evolve to use dynamically available contexts.

Due to the sensor heterogeneity, we propose to model context of sensors and activities, so when the sensors are available dynamically, the raw sensor data can be pre-processed properly for the recognition task. Based on these models, we propose the knowledge-driven and data-driven method for activity modelling and activity model adaptation with new contexts. In the knowledge-driven method, we mine external resources (e.g. websites) to specify the parameters in the activity models. While in the data-driven method, we use the predicted instances to learn the parameters of new contexts in the activity models. With the knowledge-driven method, we can perform activity model adaptation with new contexts in an unsupervised manner. However, the knowledge mined from the websites is general across users and cannot achieve high accuracy, due to the fact that people perform activity differently. On the contrary, the data-driven method can personalise the activity model to a specific user with his/her own data.

In the data-driven method, we propose the learning-to-rank approach for activity learning and adaptation. One advantage of the learning-to-rank approach is that we can add various regularization terms to exploit the characteristics of the data. In our work, we add temporal regularization into the learning and testing phases to capture the consistency of human behaviours.

Our experiments based on public and simulation datasets show that we are able to improve the recognition performance by adaptation of the activity model with dynamically available context. The improvements vary and depend on several factors such as the amount of adaptation data, the weight of the temporal regularization and the number of contexts in the initial train set. To validate the advantages of the proposed method in adaptive learning, we compared it with the conventional machine learning algorithms, and the experiments demonstrate that our methods for activity learning and activity model adaptation outperform the baselines with a large margin.

In this chapter, the proposed framework for high-level activity recognition and activity models adaptation addresses challenge 2; the proposed knowledge-driven method addresses challenge 3; the personalised learning-to-rank machine learning method and temporal regularization in the proposed data-driven method addresses challenge 5 and 6, respectively; the proposed sensor models address challenge 7.

CHAPTER 6

Conclusions and Future Work

We conclude the thesis with a summary of the key contributions and a discussion of future work.

6.1 Summary of contributions

In this thesis, we have addressed the challenges in developing frameworks for mobile activity recognition with dynamically available sensor data. Those challenges were described in Section 1.2 as 1) how to learn a generic activity model as the starting point that caters for people performing activity differently, 2) how to perform an activity model adaptation to incorporate the information provided by the new sensors, 3) how to leverage the existing knowledge base for activity modelling and activity model adaptation, 4) how to select the most informative instances for retraining the activity model without supervision, 5) how to make sure that the newly incorporated information does not negatively impact the recognition performance of the activity model, 6) how to exploit the temporal information in human behaviour to improve the recognition accuracy. This thesis addressed these challenges in the presented research on mobile activity recognition with dynamically available sensor data.

In summary, this thesis made the following key contributions.

- It researched and developed a generic activity modelling method with limited labelled activity data,

- It researched and developed a framework for low-level physical activity recognition with dynamically available sensors and semi-supervised learning; the components of the framework include basic classifier, instance selection and smoothing.
- It designed and developed a framework for high-level activity recognition with dynamically discovered contexts. The key components of the framework include sensor and activity models, knowledge-driven and data-driven methods for activity learning and activity model adaptation.

In Chapter 2, we surveyed the related work in the area of context modelling, context management, sensor modelling and sensors in mobile devices, activity recognition, activity model adaptation and sensor dynamics for activity recognition. We also discussed their shortcomings and identified the open exploration issues that motivated our research on developing activity recognition frameworks with dynamically available sensor data. In response to those issues, Chapter 3, 4 and 5 proposed solutions for addressing the challenges in incorporating dynamically available contexts for activity recognition.

Chapter 3 presented a generic activity modelling method with minimum labelled data. This method learned the activity model with data from various users, so that the model was generic to be scaled to different users. It also mitigated the data labelling effort as it required minimum labelled data from each user. We leveraged LDA to model the activity data since it was effective in collaborative learning and powerful in dealing with data sparsity. However, LDA cannot be applied to the activity data directly, and therefore we created a hybrid approach with conventional classifiers. In the hybrid model, the initial labelled data was used to train the classifiers, then the hybrid model sampled the class assignment for the activity data. After that, the output of the hybrid model was fed back to retrain the classifiers. This joint training process resulted in generic activity models that can cope with variants of activity patterns of different users. We also examined the factors (e.g. labelling percentage) that had impact on the recognition performance of the generic activity model.

Chapter 4 presented a framework for physical activity recognition with dynamically available sensors. It used AdaBoost as the basic classifier as it is flexible with feature dimensionality and can select the discriminative features automatically. We proposed a method that selected the most informative instances considering several factors. The instances were used to retrain AdaBoost and the information of the newly available sensors was incorporated into the framework through the retraining process. Further, the temporal characteristic of activity data was leveraged by a novel combination of graphical models with the basic classifier AdaBoost. Finally, we investigated the conditions under which the information provided by dynamically available sensors did not benefit the recognition performance of the activity recognition framework.

Chapter 5 presented a framework for high-level activity recognition with dynamically available contexts. Sensor and activity models were proposed in the framework. The sensor models provided the guideline to pre-process the sensor readings into high-level context (e.g. interaction with objects) for activity recognition that can deal with sensor heterogeneity. The activity models explained the relations between the activities and contexts with parameters. Based on the sensor and activity models, a knowledge-driven method was proposed to incorporate the contexts provided by the new sensors. Since the activities were semantically explainable by the contexts, the knowledge-driven method leveraged the external knowledge base to specify the parameters of the new contexts with respect to different activities in an unsupervised manner. However, the knowledge-driven method was unable to achieve high accuracy since the parameters it specified were general knowledge and people have their own ways to perform activities. To overcome the shortcomings of knowledge-driven method, a data-driven method was proposed. The data-driven method personalised the activity models to a specific user with his/her own activity data using the learning-to-rank method and temporal regularization. The parameters of the new contexts were learned during the retraining process with the most informative instances, and a thresholding method was proposed to guarantee the class balance. Extensive experiments had been carried out for validating the proposed framework and the factors that can impact the recognition performance were also investigated.

6.2 Future work

In this section, we discuss how this research can be further extended.

Activity model adaptation with more sensor modalities

In the current framework for activity recognition and activity model adaptation, the sensing devices are confined to inertial sensors such as accelerometers and gyroscopes for on-body sensors. As for environment-instrumented sensors, the sensing devices in our current framework include inertial sensors, object sensors and ambient sensors that produce binary readings. Even though those sensors are effective in recognising daily activities with high accuracy, we can envision that an increasingly large number of new sensing devices will be available for monitoring daily activities. For example, microphones [96, 8] and cameras [96, 162] that capture audio and vision features are able to differentiate different activities. Existing works also demonstrate the feasibility of using various sensor signals such as Wi-Fi signal [149], thermal [49] and barometer [127] readings for identifying activities.

However, incorporating those sensors for activity recognition and activity model adaptation is a non-trivial task. Due to the variety of sensor modalities, the sensor readings need to be processed differently. For example, for Wi-Fi signals, the signal strengths of the access points can be used directly for activity recognition [149]. While for audio signals, it is more reasonable to extract the frequent domain features [96]. Therefore, the contextual information of different types of sensors needs to be modelled, so that when the sensors are dynamically available, the sensor readings can be processed into proper representations for activity recognition and activity model adaptation. In the future, we will demonstrate the effectiveness of the proposed frameworks given the sensor diversity.

Activity model adaptation in the long run

In this thesis, we propose to select the most informative instances for activity model adaptation with dynamically available sensors. We select the classified instances based on the scores considering many factors without data annotation, so that the activity model can be adapted automatically in an unsupervised manner. However, it is possible that the misclassified instances can be selected for the activity model adaptation, and the accumulated misclassified instances can possibly jeopardise the activity models in the long run [37].

One possible solution to this potential hazard is to introduce data annotation while minimising human interruption. The existing works [24, 121] on the activity model retraining and personalisation employ active learning methods to ask the labels of the most informative instances. However, they do not evaluate the methods with activity data in the long term to demonstrate the robustness of their activity recognition systems. One of our future goals is to achieve the trade-off between activity recognition accuracy and human intervention, so that the activity models can, in the long term, maintain satisfactory accuracies when performing adaptation with dynamically available sensors, while minimising the human labelling interruption.

Bibliography

[1] M. Alam, N. Pathak, and N. Roy. Mobeacon: an ibeacon-assisted smartphone-based real time activity recognition framework. In *Proceedings of the 12th international conference on mobile and ubiquitous systems: computing, networking and services*, 2015.

[2] M. A. U. Alam, N. Roy, S. Holmes, A. Gangopadhyay, and E. Galik. Automated functional and behavioral health assessment of older adults with dementia.

[3] F. Albinali, S. Intille, W. Haskell, and M. Rosenberger. Using wearable activity type detection to improve physical activity energy expenditure estimation. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 311–320. ACM, 2010.

[4] K. Altun and B. Barshan. Human activity recognition using inertial/magnetic sensor units. In *Human behavior understanding*, pages 38–51. Springer, 2010.

[5] O. Amft, M. Stäger, P. Lukowicz, and G. Tröster. Analysis of chewing sounds for dietary monitoring. In *UbiComp 2005: Ubiquitous Computing*, pages 56–72. Springer, 2005.

[6] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *Ambient assisted living and home care*, pages 216–223. Springer, 2012.

[7] D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.

[8] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 261–272. ACM, 2009.

[9] M. Bächlin and G. Tröster. Swimming performance and technique evaluation with wearable acceleration sensors. *Pervasive and Mobile Computing*, 8(1):68–81, 2012.

[10] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *Pervasive computing*, pages 1–17. Springer, 2004.

[11] X. Bao and R. Roy Choudhury. Movi: mobile phone based video highlights via collaborative sensing. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 357–370. ACM, 2010.

[12] J. E. Bardram, A. Doryab, R. M. Jensen, P. M. Lange, K. L. Nielsen, and S. T. Petersen. Phase recognition during surgical procedures using embedded and body-worn sensors. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, pages 45–53. IEEE, 2011.

[13] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180, 2010.

[14] N. Bicocchi, M. Mamei, and F. Zambonelli. Detecting activities from body-worn accelerometers via instance-based algorithms. *Pervasive and Mobile Computing*, 6(4):482–495, 2010.

[15] A. Brajdic and R. Harle. Walk detection and step counting on unconstrained smartphones. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and ubiquitous computing*, pages 225–234. ACM, 2013.

[16] H. Cao, M. N. Nguyen, C. Phua, S. Krishnaswamy, and X.-L. Li. An integrated framework for human activity classification. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 331–340. ACM, 2012.

[17] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM, 2007.

[18] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu. Collaborative personalized tweet recommendation. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 661–670. ACM, 2012.

[19] L. Chen and C. Nugent. Ontology-based activity recognition in intelligent pervasive environments. *International Journal of Web Information Systems*, 5(4):410–430, 2009.

[20] L. Chen, C. D. Nugent, and H. Wang. A knowledge-driven approach to activity recognition in smart homes. *Knowledge and Data Engineering, IEEE Transactions on*, 24(6):961–974, 2012.

[21] M. Chen, J.-T. Sun, X. Ni, and Y. Chen. Improving context-aware query classification via adaptive self-training. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 115–124. ACM, 2011.

[22] Y.-P. Chen, J.-Y. Yang, S.-N. Liou, G.-Y. Lee, and J.-S. Wang. Online classifier construction algorithm for human activity detection using a tri-axial accelerometer. *Applied Mathematics and Computation*, 205(2):849–860, 2008.

[23] H.-T. Cheng, M. Griss, P. Davis, J. Li, and D. You. Towards zero-shot learning for human activity recognition using semantic attribute sequence model. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 355–358. ACM, 2013.

[24] H.-T. Cheng, F.-T. Sun, M. Griss, P. Davis, J. Li, and D. You. Nuactiv: Recognizing unseen new activities using semantic attribute-based learning. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 361–374. ACM, 2013.

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

[26] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao. Automatically characterizing places with opportunistic crowdsensing using smartphones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 481–490. ACM, 2012.

[27] D. J. Cook, N. C. Krishnan, and P. Rashidi. Activity discovery and activity recognition: A new partnership. *Cybernetics, IEEE Transactions on*, 43(3):820–828, 2013.

[28] B. Cvetkovic, B. Kaluza, M. Luštrek, and M. Gams. Semi-supervised learning for adaptation of human activity recognition classifier to the user. In *IJCAI Workshop*, pages 24–29. Citeseer, 2011.

[29] R. Du, Z. Yu, T. Mei, Z. Wang, Z. Wang, and B. Guo. Predicting activity attendance in event-based social networks: Content, context and social influence. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 425–434. ACM, 2014.

[30] M. Ermes, J. Pärkkä, and L. Cluitmans. Advancing from offline to online activity recognition with wearable sensors. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 4451–4454. IEEE, 2008.

[31] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *Information Technology in Biomedicine, IEEE Transactions on*, 12(1):20–26, 2008.

[32] L. G. Fahad and M. Rajarajan. Integration of discriminative and generative models for activity recognition in smart homes. *Applied Soft Computing*, 37:992–1001, 2015.

[33] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt. Micro-blog: sharing and querying content through mobile phones and social participation. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 174–186. ACM, 2008.

[34] H. Ghasemzadeh, V. Loseu, and R. Jafari. Collaborative signal processing for action recognition in body sensor networks: a distributed classification algorithm using motion transcripts. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 244–255. ACM, 2010.

[35] D. Gordon, J. Czerny, T. Miyaki, and M. Beigl. Energy-efficient activity recognition using prediction. In *Wearable Computers (ISWC), 2012 16th International Symposium on*, pages 29–36. IEEE, 2012.

[36] M. Götz, S. Nath, and J. Gehrke. Maskit: privately releasing user context streams for personalized mobile applications. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 289–300. ACM, 2012.

[37] T. Gu, S. Chen, X. Tao, and J. Lu. An unsupervised approach to activity recognition and segmentation based on object-use fingerprints. *Data & Knowledge Engineering*, 69(6):533–544, 2010.

[38] T. Gu, Z. Wu, X. Tao, H. K. Pung, and J. Lu. epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition. In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pages 1–9. IEEE, 2009.

[39] T. Gu, Z. Wu, L. Wang, X. Tao, and J. Lu. 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. IEEE, 2009.

[40] W. Gu, Z. Yang, L. Shangguan, W. Sun, K. Jin, and Y. Liu. Intelligent sleep stage mining service with smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 649–660. ACM, 2014.

[41] D. Guan, W. Yuan, Y.-K. Lee, A. Gavrilov, and S. Lee. Activity recognition based on semi-supervised learning. In *Embedded and Real-Time Computing Systems and Applications, 2007. RTCSA 2007. 13th IEEE International Conference on*, pages 469–475. IEEE, 2007.

[42] N. Y. Hammerla and T. Plötz. Let's (not) stick together: pairwise similarity biases cross-validation in activity recognition. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1041–1051. ACM, 2015.

[43] T. Hao, G. Xing, and G. Zhou. isleep: unobtrusive sleep quality monitoring using smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 4. ACM, 2013.

[44] D. Hao Hu, S. J. Pan, V. W. Zheng, N. N. Liu, and Q. Yang. Real world activity recognition with multiple goals. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 30–39. ACM, 2008.

[45] R. Helaoui, D. Riboni, and H. Stuckenschmidt. A probabilistic ontological framework for the recognition of multilevel human activities. In *Ubicomp*, pages 345–354. ACM, 2013.

[46] S. Hemminki, P. Nurmi, and S. Tarkoma. Accelerometer-based transportation mode detection on smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 13. ACM, 2013.

[47] K. Henricksen and J. Indulska. A software engineering framework for context-aware pervasive computing. In *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*, pages 77–86. IEEE, 2004.

[48] K. Henricksen and J. Indulska. Developing context-aware pervasive computing applications: Models and approach. *Pervasive and mobile computing*, 2(1):37–64, 2006.

[49] P. Hevesi, S. Wille, G. Pirkl, N. Wehn, and P. Lukowicz. Monitoring household activities and user location with a cheap, unobtrusive thermal sensor array. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 141–145. ACM, 2014.

[50] H. S. Hossain, N. Roy, and M. A. A. H. Khan. Active learning enabled activity recognition. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–9. IEEE, 2016.

[51] D. H. Hu and Q. Yang. Transfer learning for activity recognition via sensor mapping. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1962, 2011.

[52] P. Hu, J. Indulska, and R. Robinson. An autonomic context management system for pervasive computing. In *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*, pages 213–223. IEEE, 2008.

[53] T. Huáżşnh, U. Blanke, and B. Schiele. Scalable recognition of daily activities with wearable sensors. In *Location-and context-awareness*, pages 50–67. Springer, 2007.

[54] T. Huynh, M. Fritz, and B. Schiele. Discovery of activity patterns using topic models. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 10–19. ACM, 2008.

[55] T. Huynh and B. Schiele. Analyzing features for activity recognition. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, pages 159–163. ACM, 2005.

[56] S. Inoue, N. Ueda, Y. Nohara, and N. Nakashima. Mobile activity recognition for a whole day: recognizing real nursing activities with big dataset. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1269–1280. ACM, 2015.

[57] M. U. Iqbal, M. Handte, S. Wagner, W. Apolinarski, and P. J. Marrón. Enabling energy-efficient context recognition with configuration folding. In *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*, pages 198–205. IEEE, 2012.

[58] Y. Ju, C. Min, Y. Lee, J. Yu, and J. Song. An efficient dataflow execution method for mobile context monitoring applications. In *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*, pages 116–121. IEEE, 2012.

[59] S. Kang, J. Lee, H. Jang, H. Lee, Y. Lee, S. Park, T. Park, and J. Song. Seemon: scalable and energy-efficient context monitoring framework for sensor-rich mobile environments. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 267–280. ACM, 2008.

[60] S. Kang, Y. Lee, C. Min, Y. Ju, T. Park, J. Lee, Y. Rhee, and J. Song. Orchestrator: An active resource orchestration framework for mobile context monitoring in sensor-rich mobile environments. In *Pervasive Computing and Communications (PerCom), 2010 IEEE International Conference on*, pages 135–144. IEEE, 2010.

[61] P. G. Kannan, S. P. Venkatagiri, M. C. Chan, A. L. Ananda, and L.-S. Peh. Low cost crowd counting using audio tones. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 155–168. ACM, 2012.

[62] K. Kapitanova, E. Hoque, J. A. Stankovic, K. Whitehouse, and S. H. Son. Being smart about failures: assessing repairs in smart homes. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 51–60. ACM, 2012.

[63] K. Kawamoto, T. Tanaka, and H. Kuriyama. Your activity tracker knows when you quit smoking. In *Proceedings of the 2014 ACM International Symposium on Wearable Computers*, pages 107–110. ACM, 2014.

[64] M. Keally, G. Zhou, G. Xing, J. Wu, and A. Pyles. Pbn: towards practical activity recognition using smartphone-based body sensor networks. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pages 246–259. ACM, 2011.

[65] S. Khalifa, M. Hassan, and A. Seneviratne. Pervasive self-powered human activity recognition without the accelerometer. In *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on*, pages 79–86. IEEE, 2015.

[66] A. Khan, Y. Lee, and S. Lee. AccelerometerâĂŹs position free human activity recognition using a hierarchical recognition model. In *12th IEEE International Conference on e-Health Networking Applications and Services (Healthcom)*, pages 296–301, 2010.

[67] A. M. Khan, Y.-K. Lee, S. Y. Lee, and T.-S. Kim. A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. *Information Technology in Biomedicine, IEEE Transactions on*, 14(5):1166–1172, 2010.

[68] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava. Sensloc: sensing everyday places and paths using less energy. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 43–56. ACM, 2010.

[69] E. Kim, S. Helal, and D. Cook. Human activity recognition and pattern discovery. *Pervasive Computing, IEEE*, 9(1):48–53, 2010.

[70] V. Könönen, J. Mäntyjärvi, H. Similä, J. Pärkkä, and M. Ermes. Automatic feature selection for context recognition in mobile devices. *Pervasive and Mobile Computing*, 6(2):181–197, 2010.

[71] J. Korpela, R. Miyaji, T. Maekawa, K. Nozaki, and H. Tamagawa. Evaluating tooth brushing performance with smartphone sound data. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 109–120. ACM, 2015.

[72] N. Krishnan and D. Cook. Activity recognition on streaming sensor data. *Pervasive and mobile computing*, 10(Pt B):138, 2014.

[73] J. Krumm and D. Rouhana. Placer: semantic place labels from diary data. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 163–172. ACM, 2013.

[74] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[75] H. Lakkaraju, E. Aguiar, C. Shan, D. Miller, N. Bhanpuri, R. Ghani, and K. L. Addison. A machine learning framework to identify students at risk of adverse academic outcomes. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1909–1918. ACM, 2015.

[76] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell. A survey of mobile phone sensing. *Communications Magazine, IEEE*, 48(9):140–150, 2010.

[77] N. D. Lane, Y. Xu, H. Lu, S. Hu, T. Choudhury, A. T. Campbell, and F. Zhao. Enabling large-scale human activity inference on smartphones using community similarity networks (csn). In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 355–364. ACM, 2011.

[78] Ó. D. Lara, A. J. Pérez, M. A. Labrador, and J. D. Posada. Centinela: A human activity recognition system based on acceleration and vital sign data. *Pervasive and mobile computing*, 8(5):717–729, 2012.

[79] Y. Lee, Y. Ju, C. Min, S. Kang, I. Hwang, and J. Song. Comon: cooperative ambience monitoring platform with continuity and benefit awareness. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 43–56. ACM, 2012.

[80] Y.-S. Lee and S.-B. Cho. Activity recognition using hierarchical hidden markov models on a smartphone with 3d accelerometer. In *Hybrid Artificial Intelligent Systems*, pages 460–467. Springer, 2011.

[81] J. Lester, T. Choudhury, and G. Borriello. A practical approach to recognizing physical activities. In *Pervasive Computing*, pages 1–16. Springer, 2006.

[82] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

[83] L. Liao, T. Choudhury, D. Fox, and H. A. Kautz. Training conditional random fields using virtual evidence boosting. In *IJCAI*, volume 7, pages 2530–2535, 2007.

[84] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao. Energy-accuracy trade-off for continuous mobile device location. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 285–298. ACM, 2010.

[85] B. Liu, D. Kong, L. Cen, N. Z. Gong, H. Jin, and H. Xiong. Personalized mobile app recommendation: Reconciling app functionality and user privacy preference. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 315–324. ACM, 2015.

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

[87] B. Longstaff, S. Reddy, and D. Estrin. Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on-NO PERMISSIONS*, pages 1–7. IEEE, 2010.

[88] K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh. Mercury: a wearable sensor network platform for high-fidelity motion analysis. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 183–196. ACM, 2009.

[89] H. Lu, D. Frauendorfer, M. Rabbi, M. S. Mast, G. T. Chittaranjan, A. T. Campbell, D. Gatica-Perez, and T. Choudhury. Stresssense: Detecting stress in unconstrained acoustic environments using smartphones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 351–360. ACM, 2012.

[90] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell. Soundsense: scalable sound sensing for people-centric applications on mobile phones. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 165–178. ACM, 2009.

[91] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM conference on embedded networked sensor systems*, pages 71–84. ACM, 2010.

[92] G. Luis and O. Amft. mining relations and physical grouping building-embedded sensors and actuator. In *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on*. IEEE, 2014.

[93] C. Luo and M. C. Chan. Socialweaver: collaborative inference of human conversation networks using smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 20. ACM, 2013.

[94] T. Maekawa, Y. Kishino, Y. Sakurai, and T. Suyama. Activity recognition with hand-worn magnetic sensors. *Personal and ubiquitous computing*, 17(6):1085–1094, 2013.

[95] T. Maekawa and S. Watanabe. Unsupervised activity recognition with user's physical characteristics data. In *Wearable Computers (ISWC), 2011 15th Annual International Symposium on*, pages 89–96. IEEE, 2011.

[96] T. Maekawa, Y. Yanagisawa, Y. Kishino, K. Ishiguro, K. Kamei, Y. Sakurai, and T. Okadome. Object-based activity recognition with heterogeneous sensors on wrist. In *Pervasive Computing*, pages 246–264. Springer, 2010.

[97] M. Mahdaviani and T. Choudhury. Fast and scalable training of semi-supervised crfs with application to activity recognition. In *Advances in Neural Information Processing Systems*, pages 977–984, 2008.

[98] A. Mannini and A. M. Sabatini. Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, 10(2):1154–1175, 2010.

[99] J. Meng, H. Li, and Z. Han. Sparse event detection in wireless sensor networks using compressive sensing. In *Information Sciences and Systems, 2009. CISS 2009. 43rd Annual Conference on*, pages 181–185. IEEE, 2009.

[100] E. Miluzzo, C. T. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu, and A. T. Campbell. Darwin phones: the evolution of sensing and inference on mobile phones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 5–20. ACM, 2010.

[101] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 337–350. ACM, 2008.

[102] M. Mühlenbrock, O. Brdiczka, D. Snowdon, and J.-L. Meunier. Learning to detect user activity and availability from a variety of sensor data. In *Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04)*, page 13. IEEE Computer Society, 2004.

[103] S. Nath. Ace: exploiting correlation for energy-efficient and continuous context sensing. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 29–42. ACM, 2012.

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

[105] L. T. Nguyen, M. Zeng, P. Tague, and J. Zhang. I did not smoke 100 cigarettes today!: avoiding false positives in real-world activity recognition. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1053–1063. ACM, 2015.

[106] L. T. Nguyen, M. Zeng, P. Tague, and J. Zhang. Recognizing new activities with limited training data. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pages 67–74. ACM, 2015.

[107] T. Nguyen, D. Phung, S. Gupta, and S. Venkatesh. Extraction of latent patterns and contexts from social honest signals using hierarchical dirichlet processes. In *Pervasive Computing and Communications (PerCom), 2013 IEEE International Conference on*, pages 47–55. IEEE, 2013.

[108] M. Niepert, J. Noessner, and H. Stuckenschmidt. Log-linear description logics. In *IJCAI*, pages 2153–2158. AAAI Press, 2011.

[109] S. Nirjon, R. F. Dickerson, P. Asare, Q. Li, D. Hong, J. A. Stankovic, P. Hu, G. Shen, and X. Jiang. Auditeur: a mobile-cloud service platform for acoustic event detection on smartphones. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 403–416. ACM, 2013.

[110] T. Oberlies. Understanding virtual evidence boosting: A new derivation and application to recognizing spatial context with wearable cameras. *Graphical Models*, 2(5), 2007.

[111] G. Okeyo, L. Chen, H. Wang, and R. Sterritt. Dynamic sensor data segmentation for real-time knowledge-driven activity recognition. *Pervasive and Mobile Computing*, 2012.

[112] A. Parate, M.-C. Chiu, D. Ganesan, and B. M. Marlin. Leveraging graphical models to improve accuracy and reduce privacy risks of mobile sensing. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 83–96. ACM, 2013.

[113] T. Park, J. Lee, I. Hwang, C. Yoo, L. Nachman, and J. Song. E-gesture: a collaborative architecture for energy-efficient gesture recognition with hand-worn sensor and mobile devices. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pages 260–273. ACM, 2011.

[114] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *ISWC*, pages 44–51. IEEE, 2005.

[115] D. J. Patterson, L. Liao, D. Fox, and H. Kautz. Inferring high-level behavior from low-level sensors. In *UbiComp 2003: Ubiquitous Computing*, pages 73–89. Springer, 2003.

[116] M. Perkowitz, M. Philipose, K. Fishkin, and D. J. Patterson. Mining models of human activities from the web. In *Proceedings of the 13th international conference on World Wide Web*, pages 573–582. ACM, 2004.

[117] M.-R. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govindan. Odessa: enabling interactive perception applications on mobile devices. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 43–56. ACM, 2011.

[118] K. K. Rachuri, M. Musolesi, C. Mascolo, P. J. Rentfrow, C. Longworth, and A. Aucinas. Emotionsense: a mobile phones based adaptive platform for experimental social psychology research. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 281–290. ACM, 2010.

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

[120] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman. Activity recognition from accelerometer data. In *AAAI*, volume 5, pages 1541–1546, 2005.

[121] J. Rebetez, H. F. Satizábal, and A. Perez-Uribe. Reducing user intervention in incremental activityrecognition for assistive technologies. In *Proceedings of the 2013 International Symposium on Wearable Computers*, pages 29–32. ACM, 2013.

[122] A. Reiss, G. Hendeby, and D. Stricker. Confidence-based multiclass adaboost for physical activity monitoring. In *Proceedings of the 2013 International Symposium on Wearable Computers*, pages 13–20. ACM, 2013.

[123] A. Reiss and D. Stricker. Personalized mobile physical activity recognition. In *Proceedings of the 2013 International Symposium on Wearable Computers*, pages 25–28. ACM, 2013.

[124] D. Riboni and C. Bettini. Cosar: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3):271–289, 2011.

[125] D. Riboni, L. Pareschi, L. Radaelli, and C. Bettini. Is ontology-based activity recognition really effective? In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 427–431. IEEE, 2011.

[126] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, et al. Collecting complex activity datasets in highly rich networked sensor environments. In *Networked Sensing Systems (INSS), 2010 Seventh International Conference on*, pages 233–240. IEEE, 2010.

[127] K. Sankaran, M. Zhu, X. F. Guo, A. L. Ananda, M. C. Chan, and L.-S. Peh. Using mobile phone barometer for low-power transportation context detection. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pages 191–205. ACM, 2014.

[128] J. Seiter, W.-C. Chiu, M. Fritz, O. Amft, and G. Troster. Joint segmentation and activity discovery using semantic and temporal priors. In *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on*, pages 71–78. IEEE, 2015.

[129] G. Seni and J. F. Elder. Ensemble methods in data mining: improving accuracy through combining predictions. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 2(1):1–126, 2010.

[130] M. Shoaib, H. Scholten, and P. J. Havinga. Towards physical activity recognition using smartphone sensors. In *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)*, pages 80–87. IEEE, 2013.

[131] M. Stikic, D. Larlus, and B. Schiele. Multi-graph based semi-supervised learning for activity recognition. In *Wearable Computers, 2009. ISWC'09. International Symposium on*, pages 85–92. IEEE, 2009.

[132] M. Stikic and B. Schiele. Activity recognition from sparsely labeled data using multi-instance learning. In *Location and Context Awareness*, pages 156–173. Springer, 2009.

[133] M. Stikic, K. Van Laerhoven, and B. Schiele. Exploring semi-supervised and active learning for activity recognition. In *Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium on*, pages 81–88. IEEE, 2008.

[134] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 127–140. ACM, 2015.

[135] I. Suarez, A. Jahn, C. Anderson, and K. David. Improved activity recognition by using enriched acceleration data. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1011–1015. ACM, 2015.

[136] F.-T. Sun, Y.-T. Yeh, H.-T. Cheng, C. Kuo, and M. Griss. Nonparametric discovery of human routines from sensor data. In *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*, pages 11–19. IEEE, 2014.

[137] M. Sundholm, J. Cheng, B. Zhou, A. Sethi, and P. Lukowicz. Smart-mat: Recognizing and counting gym exercises with low-cost resistive pressure sensing matrix. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*, pages 373–382. ACM, 2014.

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

[139] E. M. Tapia, T. Choudhury, and M. Philipose. Building reliable activity models using hierarchical shrinkage and mined ontology. In *Pervasive Computing*, pages 17–32. Springer, 2006.

[140] E. Thomaz, I. Essa, and G. D. Abowd. A practical approach for recognizing eating moments with wrist-mounted inertial sensing. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1029–1040. ACM, 2015.

[141] T. Van Kasteren, G. Englebienne, and B. J. Kröse. An activity monitoring system for elderly care using generative and discriminative models. *Personal and ubiquitous computing*, 14(6):489–498, 2010.

[142] T. Van Kasteren, G. Englebienne, and B. J. Kröse. Transferring knowledge of activity recognition across sensor networks. In *Pervasive computing*, pages 283–300. Springer, 2010.

[143] T. van Kasteren and B. Krose. Bayesian activity recognition in residence for elders. In *Intelligent Environments, 2007. IE 07. 3rd IET International Conference on*, pages 209–212. IET, 2007.

[144] T. Van Kasteren, A. Noulas, G. Englebienne, and B. Kröse. Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 1–9. ACM, 2008.

[145] V. Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2013.

[146] L. Wang, T. Gu, X. Tao, H. Chen, and J. Lu. Recognizing multi-user activities using wearable sensors in a smart home. *Pervasive and Mobile Computing*, 7(3):287–298, 2011.

[147] S. Wang, W. Pentney, A.-M. Popescu, T. Choudhury, and M. Philipose. Common sense based joint training of human activity recognizers. In *In: Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007.

[148] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh. A framework of energy efficient mobile sensing for automatic user state recognition. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 179–192. ACM, 2009.

[149] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu. E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 617–628. ACM, 2014.

[150] G. M. Weiss and J. W. Lockhart. The impact of personalization on smartphone-based activity recognition. In *AAAI Workshop*, 2012.

[151] J. Wen, S. Loke, J. Indulska, and M. Zhong. Sensor-based activity recognition with dynamically added context. In *proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services on 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*,

pages 1–10. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2015.

[152] J. Wen, M. Zhong, and J. Indulska. Creating general model for activity recognition with minimum labelled data. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pages 87–90. ACM, 2015.

[153] C. Wojek, K. Nickel, and R. Stiefelhagen. Activity recognition and room-level tracking in an office environment. In *Multisensor Fusion and Integration for Intelligent Systems, 2006 IEEE International Conference on*, pages 25–30. IEEE, 2006.

[154] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg. A scalable approach to activity recognition based on object use. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[155] M. Wu, P. H. Pathak, and P. Mohapatra. Monitoring building door events using barometer sensor in smartphones. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 319–323. ACM, 2015.

[156] D. Wyatt, M. Philipose, and T. Choudhury. Unsupervised activity recognition using automatically mined common sense. In *AAAI*, volume 5, pages 21–27, 2005.

[157] C. Xu, S. Li, G. Liu, Y. Zhang, E. Miluzzo, Y.-F. Chen, J. Li, and B. Firner. Crowd++: unsupervised speaker count with smartphones. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 43–52. ACM, 2013.

[158] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, and K. Aberer. Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach. In *Wearable Computers (ISWC), 2012 16th International Symposium on*, pages 17–24. Ieee, 2012.

[159] K. Yatani and K. N. Truong. Bodyscope: a wearable acoustic sensor for activity recognition. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 341–350. ACM, 2012.

[160] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *Information Theory, IEEE Transactions on*, 51(7):2282–2312, 2005.

[161] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Tröster. Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection. In *Wireless sensor networks*, pages 17–33. Springer, 2008.

[162] K. Zhan, S. Faux, and F. Ramos. Multi-scale conditional random fields for first-person activity recognition. In *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*, pages 51–59. IEEE, 2014.

[163] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu. Cross-people mobile-phone based activity recognition. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[164] J. Zheng, S. Liu, and L. M. Ni. Effective mobile context pattern discovery via adapted hierarchical dirichlet processes. In *Mobile Data Management (MDM), 2014 IEEE 15th International Conference on*, volume 1, pages 146–155. IEEE, 2014.

[165] V. W. Zheng, D. H. Hu, and Q. Yang. Cross-domain activity recognition. In *Proceedings of the 11th international conference on Ubiquitous computing*, pages 61–70. ACM, 2009.

[166] V. W. Zheng and Q. Yang. User-dependent aspect model for collaborative activity recognition. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 2085, 2011.

[167] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th international conference on World wide web*, pages 1029–1038. ACM, 2010.

[168] B. Zhou, J. Cheng, M. Sundholm, A. Reiss, W. Huang, O. Amft, and P. Lukowicz. Smart table surface: A novel approach to pervasive dining monitoring. In *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on*, pages 155–162. IEEE, 2015.

[169] P. Zhou, Y. Zheng, Z. Li, M. Li, and G. Shen. Iodetector: a generic service for indoor outdoor detection. In *Proceedings of the 10th acm conference on embedded network sensor systems*, pages 113–126. ACM, 2012.

[170] Z. Zhu, U. Blanke, and G. Tröster. Recognizing composite daily activities from crowd-labelled social media data. *Pervasive and Mobile Computing*, 2015.

[171] Z. Zhuang, K.-H. Kim, and J. P. Singh. Improving energy efficiency of location sensing on smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 315–330. ACM, 2010.