Learning and Recognizing The Hierarchical and Sequential Structure of Human Activities

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Heng-Tze Cheng

B.S., Electrical Engineering, National Taiwan University
M.S., Electrical and Computer Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

December, 2013

Keywords: Activity recognition, machine learning, zero-shot learning, active learning, semantic attributes, feature extraction, probabilistic graphical models, context-aware systems, ubiquitous computing, mobile computing, user behavior modeling, sensor data analysis, artificial intelligence.

For my parents and family

Abstract

The mission of the research presented in this thesis is to give computers the power to sense and react to human activities. Without the ability to sense the surroundings and understand what humans are doing, computers will not be able to provide active, timely, appropriate, and considerate services to the humans. To accomplish this mission, the work stands on the shoulders of two giants: Machine learning and ubiquitous computing. Because of the ubiquity of sensor-enabled mobile and wearable devices, there has been an emerging opportunity to sense, learn, and infer human activities from the sensor data by leveraging state-of-the-art machine learning algorithms.

While having shown promising results in human activity recognition, most existing approaches using supervised or semi-supervised learning have two fundamental problems. Firstly, most existing approaches require a large set of labeled sensor data for every target class, which requires a costly effort from human annotators. Secondly, an unseen new activity cannot be recognized if no training samples of that activity are available in the dataset. In light of these problems, a new approach in this area is proposed in our research.

This thesis presents our novel approach to address the problem of human activity recognition when few or no training samples of the target activities are available. The main hypothesis is that the problem can be solved by the proposed *NuActiv* activity recognition framework, which consists of modeling the *hierarchical* and *sequential* structure of human activities, as well as bringing humans in the loop of model training. By injecting human knowledge about the hierarchical nature of human activities, a semantic attribute representation and a two-layer attribute-based learning approach are designed. To model the sequential struc-

ture, a probabilistic graphical model is further proposed to take into account the temporal dependency of activities and attributes. Finally, an active learning algorithm is developed to reinforce the recognition accuracy using minimal user feedback.

The hypothesis and approaches presented in this thesis are validated by two case studies and real-world experiments on exercise activities and daily life activities. Experimental results show that the NuActiv framework can effectively recognize unseen new activities even without any training data, with up to 70-80% precision and recall rate. It also outperforms supervised learning with limited labeled data for the new classes. The results significantly advance the state of the art in human activity recognition, and represent a promising step towards bridging the gap between computers and humans.

Acknowledgments

I was truly amazed when I first heard the quote: "*Any sufficiently advanced technology is indistinguishable from magic.*" I am lucky and excited to be one of the crew to advance computer technology and create the world that has never been. While this thesis is just the first milestone towards a great magic show, I would like to show my gratitude to many people who had helped me while I am working my way to become a real magician.

First of all, I would like to thank Professor Martin Griss for all the invaluable advice and inspiration along the way of my study in Carnegie Mellon. While I am finishing up this thesis, I can still remember the day when we were first discussing and feeling excited about the idea and the future of context-aware systems in Martin's office. I enjoyed and cherished every moment working with him throughout these years: From hypotheses to experiments, from ideas to code, and from proposals to papers. Martin has always been a supportive, encouraging, and caring mentor to me. I am extremely lucky to have him as my advisor.

Next, I would like to thank Professor Ying Zhang, Professor Ole Mengshoel, and Dr. Paul Davis for their help on my research, and for being the committee members of my thesis. I very much enjoyed Professor Zhang and Mengshoel's statistical learning class, which stimulated my passion for the theory and applications of machine learning. In addition, I would like to thank Dr. Paul Davis for mentoring me and giving me advice on research for the past two years, including a summer internship. I learned a lot of things from him not only on the research and technical side, but also from his way of leading a team, giving insightful directions, and encouraging people. I am very grateful for the advice and help from Professor Dan Siewiorek, Professor Asim Smailagic, Professor Richard

Stern, Professor Pei Zhang, Professor Patrick Tague, Professor Ian Lane, Professor Stuart Evans, and Wendy Fong. I would also like to thank Professor Tom Mitchell for giving us the best machine learning lectures in the world. Finally, I would like to acknowledge the financial support for our research from Carnegie Mellon University, Nokia, and Motorola Mobility.

Besides Carnegie Mellon, I have gained invaluable experiences from several great summer internships in the industry. I am grateful for all the help from my manager at Qualcomm, Dr. An Chen, who I think has a perfect sense of how to set the right goal and lead the team to achieve it. I wish one day I can be such a great leader as she is. I also appreciate the help from my mentors and teammates, Ashu Razdan, Elliot Buller, and more. For the time at Motorola Mobility, in addition to Paul, I would like to thank my manager, Dr. Georg Treu, who is a successful entrepreneur, an inspiring mentor, and a great friend. I would also like to thank my teammates and co-authors, Jianguo Li, Di You, and more. Finally, I would like to thank my awesome mentors and teammates at Google—Evan Ettinger, Jian Chen, Halit Erdogan, Poonam Suryanarayan, and Chris Harris. Their mentorship and guidance had made me a better researcher and software engineer. Thanks Google for indexing the fruits of our research and making them universally accessible and useful.

Many thanks to my best bros, Lucas Sun and Zheng Sun. Those countless insightful discussions and tasty meals we had together have enriched my research and life. Thanks to the help and fun from all my labmates and friends at Carnegie Mellon: Senaka Buthpitiya, Faisal Luqman, Aveek Purohit, David Huang, Lu Zheng, Le Nguyen, Rahul Rajan, Bruce DeBruhl, Xiao Wang, Yuseung Kim, Eric Chen, Arjun Athreya, Frank Mokaya, Avneesh Saluja, David Cohen, Jiang Zhu, Yuan Tian, Shijia Pan, Dongzhen Piao, Brian Ricks, William Chan, Pang Wu,

Song Luan, John Yu, Pumbaa Peng, Da-Cheng Juan, JR Lin, Cheng-Yuan Wen, Kai-Chiang Wu, Ryan Yang, Chih-Hung Lin, Der-Nien Lee, and more. Special thanks to my roommates and friends in Pittsburgh—Li-Heng Liang, Shun-Ping Chiu, Jean Lu, Wei Jeng, Hsu-Chun Hsiao, Chieh-Li Chen—for all the fun times together. Though the weather outside was frightful, the fun we had was so delightful.

In addition to the colleagues and friends in the U.S., I would like to thank my undergraduate research advisors, Professor Lin-shan Lee and Professor Homer Chen at National Taiwan University, for being so generous with their time and expertise. They enlightened me with the essential research methodology and opened my eyes to the beauty of machine learning and its applications to audiovisual signals. I also want to thank Dr. Yi-Hsuan Yang and Yu-Ching Lin for mentoring me and guiding me through the fundamentals of research. It was them who made me discover how exciting research can be.

Thanks to the help from all of my classmates in NTUEE, who now all become outstanding researchers and engineers in Taiwan and all over the world. Thanks to Michael Tang, Chang-Hong Hsu, Meng-Che Chuang, Christy Chu, Mark Hsiao, Deeann Chen, Ying-Yu Chen, Po-Sen Huang, and many more for those meaningful/meaningless online chat sessions and gatherings. Thanks to those who have been in the bay area—Chris Liao, Jyou, Hsu-Chieh Lee, Tsung-Kai Lin, Clifton Lin, Yao-Hsun Li, Hung-Ming Lin, Yi-Chun Chou and more—for enriching my life with endless cuisine tasting and trash talking. Thanks to the inspiration from NTUEE Googlers, Hsin-Yu Chao, Cheng-Yi Chiang, Tai-Hsu Lin, Ben-Yue Chang, and more. Special thanks to Chih-Chun Chia, for being the best classmate again at Google, and for all the career and want-repreneurship sessions we had. Finally, a big thank you goes to my mentor, Fang-Pei Tracy

Lien, and all my friends in NTUChorus—Yeun-Jong Young, Ting-Chieh Pai, Yu-Ning Chang, H.w. Lu, Shan-Ching Yu, Cheng-Yuan Lin, Miles Lee, Chi Lee, Biglight, Ching-Yi Chen, Chieh-Chun Lee and more—for the cozy gatherings we have every year, and for all the songs we sang together. So I say thank you for the music, for giving those great memories to me.

Thanks to several coffee shops—the Starbucks on Centre Avenue in Pittsburgh, the two Starbucks in Sunnyvale, the Philz Coffee in Palo Alto, the Chromatic Coffee in Santa Clara, and most importantly, the Red Rock Coffee in Mountain View—for their mochas, bagels, and music. Those are my favorite workplaces and my sanctuaries when I am thousands of miles away from my hometown. Countless ideas and work were born and finished there.

With deepest love, I want to say thank you to my best friend and my soulmate, Yi-Hui Lin, for our beautiful adventure together. She has been standing by my side through the hardest times and the best. Through the years we shared our tears and joy and explored the world together. It takes a tenor and a soprano to sing the duet of life—and the melody would never have been so delightful if not for her.

As people say, there's no place like home. I would like to thank my grandparents, Chen-Huai Lee, Kuei-Ying Liao, Cheuk-Leung Cheng, and Fung-Kwan Chan, for their unconditional love and support since my childhood. I would also like to thank my uncles, my aunts, and my cousins—Chien-Jung Lee, Jienken, Chien-Hsing Lee, Vinci Chang, Po-Hsia Cheng, Sean Lee, Sabrina Lee, Coco Lee, and Tang Lee—for all the lovely family times.

Finally and most importantly, I want to thank my mom and dad, Fu-Mei Lee and Tak-Ming Cheng. The best teachers at school turned out to be the best parents at home as well. They have nurtured me with love, educated me

with patience, and supported me with full of their hearts. The reason why I have never felt lost in my life is that I always feel loved whenever I think of my parents. Without them I would never have the ability, confidence, and courage to fulfill my dream. This thesis is dedicated to them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"You see, but you do not observe. The distinction is clear."

— Sherlock Holmes, in *The Adventures of Sherlock Holmes* (1892)
Story by Sir Arthur Conan Doyle

The research presented in this thesis started with one mission: *To give computers the power to sense and react*. Over the years, computers have been redesigned and built to help humans better than ever before. They have been excellent tools that take humans' commands and execute them with ever-increasing speed and precision. The problem is that computers have been good at *passive* services only: They usually do not know what to do when there are no commands given. Without the ability to sense the surroundings and understand what humans are doing, a computer will not be able to truly provide active, timely, appropriate, and considerate services to the humans.

Computer scientists have been trying to give computers the ability to see and listen like humans do through research in computer vision and audition. This work is dedicated to empower computers with a new capability—to "sense" what a person is doing without necessarily seeing it. This is made possible by the ubiquity of sensors on mobile and wearable devices. After living on our desks for decades, computers now shrink, multiply, and fade into

the background of our lives—the phones we use, the watches we wear, and the houses we live in. Computers are increasingly invisible yet more present than ever at the same time. As envisioned by Mark Weiser more than two decades ago [93],

> *"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it."*
>
> *"Specialized elements of hardware and software, connected by wires, radio waves and infrared, will be so ubiquitous that no one will notice their presence."*

This vision of *ubiquitous computing* has now become a reality. It is becoming easier to acquire a large amount of sensor data from the ubiquitous mobile phones and wearable devices. However, just as the quote goes in the beginning of this chapter, computers do "see" a myriad of sensor data but they do not "observe" the meanings, insights, or implications behind those data. To help computers go from "seeing" to "observing," the next big question is: How to infer human activities from these sensor data?

Inferring human activities is challenging because human activities are known to be complex and highly diverse [38]. There is inherently a large semantic gap between the low-level sensor data and the high-level human activities. Most previous work tried to find a direct mapping from the low-level sensor data to the high-level activities. However, there are at least two problems: First, a large set of sensor data and labels is needed for every single class. Second, the existing approach for activity recognition does not generalize to unseen new activities. The goal of our research is to tackle these problems by studying the inherent hierarchical and sequential structure of human activities, and by injecting human knowledge into the learning process to bridge the gap between sensor data and high-level semantics.

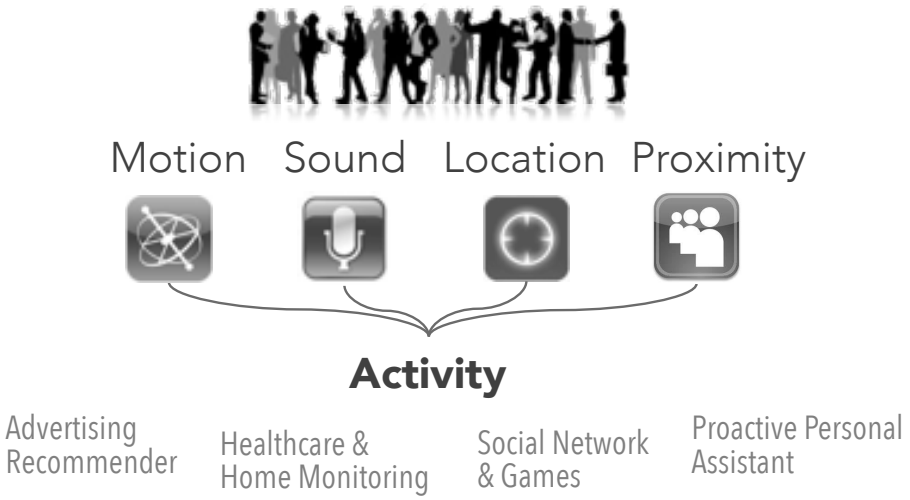


Figure 1.1: Motivating applications of human activity recognition.

1.1 Motivation and Research Problems

The understanding of context and human activities is a core component that supports and enables all kinds of context-aware, user-centric mobile applications [2, 16, 43, 44, 66, 87]. As illustrated in Figure 1.1, examples of application areas include user behavior modeling for marketing and advertising, health care and home monitoring, context-based personal assistants, social networks, and context-enabled games [43].

As sensor-enabled mobile phones and wearable devices become ubiquitous, there has been an emerging research opportunity to learn human activities and the surroundings from low-level sensor data. There has been extensive research on activity recognition using various sensors and various machine learning algorithms [4, 15, 17, 44, 51, 54, 60, 92, 94]. To recognize the activity of a user, most existing approaches require two steps: (1) Collect and label a set of training data for every activity class that the system aims to detect, and (2) classify the current sensor readings into one of the pre-defined classes. However, labeled examples are often very time consuming and expensive to obtain, as they require a lot of effort from test subjects, human annotators, or domain experts. Therefore, it has been reported that a fully

supervised learning method, where labeled examples from different context are provided to the system, would not be practical [59, 83]. More importantly, existing approaches to activity recognition cannot recognize a previously unseen new activity if there are no training samples of that activity in the dataset. According to the activity lexicon in the American Time Use Survey by U.S. Bureau of Labor Statistics [89], there are at least 462 different activities that people do in their daily lives. Considering the diversity of people and cultures that were not covered by the study, the actual number of activities is likely even larger. However, the fundamental problems in existing activity recognition approaches prevent the systems from recognizing any previously unseen activity and from extending the approach to tens or hundreds of different human activity classes.

In light of these existing problems and limitations, this thesis aims to answer two major research questions:

Question 1: Given a sequence of sensor data, how to recognize a human activity class, even when few or no training data for that activity are available?

Question 2: How does an activity recognition system reinforce its recognition accuracy with a minimal number of requests for ground-truth labels?

In this work, the *NuActiv* framework is designed to recognize human activity even when there are no training data for a particular activity class. NuActiv can generalize previously learned knowledge and extend its capability to recognize new activity classes. The proposed approach is inspired by the following observations:

- *Many human activities and context types share the same underlying semantic attributes*: For example, the attributes "`Sitting`" and "`HandsOnTable`" can be observed in both the "having lunch" and "working at desk" activities. Therefore, the statistical model of an attribute can potentially be transferred from one activity to the another.
- *The limits of supervised learning can be overcome by incorporating human knowledge*: Rather than collecting sensor data and labels for every context, using nameable at-

Table 1.1: Different learning problems in human activity recognition.

Problem Type	**Seen** (Instances of target class in training set)	**Unseen** (Instances of target class not in training set)
Known (Target class is known in training)	Supervised Learning (This Thesis)	Zero-Shot Learning (This Thesis)
Unknown (Target class is unknown in training)	Unsupervised Learning	Anomaly Detection, Rare Class Discovery, Open Set Recognition

tributes allows humans to describe a context type without the process of sensor data collection. For example, one can easily associate the activity "office working" with the motion-related attributes such as "`Sitting`," "`HandsOnTable`," and sound-related attributes such as "`PrinterSound`," "`KeyboardSound`," and "`Conversations`."

Based on these observations, we developed the NuActiv framework to tackle the two research questions. Question 1 is often referred to as the *zero-shot learning* problem, where the goal is to learn a classifier that can recognize *unseen* new classes that have never appeared in the training dataset [68]. While having been shown successful in the recent computer vision literature [42], zero-shot learning has not been studied before in the area of human activity recognition from sensor data. Table 1.1 defines the difference between learning problems with seen/unseen and known/unknown classes, and shows how this thesis is positioned in the problem space.

There are several challenges when applying zero-shot learning to activity recognition. Firstly, while there exist some well-established attributes in the field of computer vision (such as shapes and colors), it has not been shown what kinds of representations or attributes are useful for recognizing human activities from sensor data. Secondly, most previous work on zero-shot learning focused on static image data, which is inherently different from sequential sensor data in activity recognition.

To address these challenges, we studied the hierarchical structure of human activities

Table 1.2: Semantic hierarchy of human activities and terminology used in this thesis.

Term	**Description**	**Examples**
Activity Class	A high-level label of a human activity.	Having lunch, playing soccer, cooking meals
Attribute	A mid-level semantic representation that describes a high-level activity by human readable words. The construction of an attribute set is derived from human knowledge about the activity classes. The presence of an attribute is detected by computing a function over the features.	`ArmDown`, `Standing`, `CyclicMotion`, `IsTeamSport`, `MealRelated`
Feature	A low-level representation derived or computed from the raw signals. The degenerate case is the raw signals themselves. Though not required, a feature is often a compact summarization of the raw signal characteristics, and thus the feature space is of lower dimension than the signals.	Mean, standard deviation, pair-wise correlation, zero-crossing rate.
Sensor Data, Signals	The raw signals obtained from sensors.	$\{0.36, 0.15, -1.4, 2.1, 3.7, 0.9 \ldots\}$

and designed a new representation by decomposing high-level activities into combinations of *semantic attributes*, where each attribute is a human readable term that describes a basic element or an intrinsic characteristic of an activity. The definition of the term *attribute* and its relation to other terms used in this thesis is presented in Table 1.2. The semantic attributes are detected based on the low-level features, which capture the temporal dynamics in the sequential sensor data. Using this representation, a two-layer attribute-based learning algorithm is developed for activity recognition. To further model the sequential structure, a probabilistic graphical model is proposed to take into account the temporal dependency of activities and attributes.

For Question 2, to reinforce the activity recognition accuracy by leveraging user feedback, we extended the previous work in active learning [81] by designing an outlier-aware active learning algorithm and a hybrid stream/pool-based sampling scheme, which is suitable for

the scenario of activity recognition using mobile or wearable devices. By integrating active learning into the framework of zero-shot learning for activity recognition, the system is able to not only recognize unseen activities but also actively request labels from users when possible.

This thesis makes contributions to human activity recognition in the field of mobile and ubiquitous computing, as well as attribute-based zero-shot learning in the field of machine learning. The main contributions include:

- The design, development, and implementation of a new framework for human activity recognition even when there are no training data for a particular activity class.
- The novel representation of human activities using bag-of-attributes models and attribute sequence models.
- The design and study of the semantic attribute sequence model, a new and general zero-shot learning model that is suitable for sequential data.
- The design of an active learning algorithm for activity recognition, which efficiently reinforce the recognition accuracy using minimal user feedback.
- The evaluation of the proposed framework on real-world experiments in two activity domains.
- The first results on zero-shot learning for human activity recognition.

1.2 Thesis Statement

In this thesis, a new machine learning framework for human activity recognition is proposed. The framework is designed based on the hierarchical and sequential structure of human activities. Using the proposed framework along with the injection of human knowledge, an activity recognition system can (1) effectively recognize a new activity class even when

few or no training examples of that activity are available, and (2) reinforce its recognition accuracy with a minimal number of requests for ground-truth labels. The thesis presents a thorough study and evaluation of the proposed framework on real-world datasets in two activity domains, including daily life activities and exercise activities.

1.3 Thesis Roadmap

The roadmap of the thesis is shown in Figure 1.2. The thesis is organized as follows: In Chapter 2, the background and related work of activity recognition are presented. The overview of the proposed NuActiv activity recognition framework, including the feature extraction, is presented in Chapter 3. In Chapters 4 to 6, we will go through the design considerations and the details of the proposed approaches. Chapter 4 presents the core idea of semantic attributes for modeling the hierarchical structure of human activities. Chapter 5 extends Chapter 4 by shifting the focus to another dimension: The sequential structure of human activities. Chapter 6 brings humans into the loop of reinforcing an activity recognition system through active learning. We present the dataset collection, evaluation methodology, experimental results, and discussions in Chapter 7. The conclusion and future work are discussed in Chapter 8.

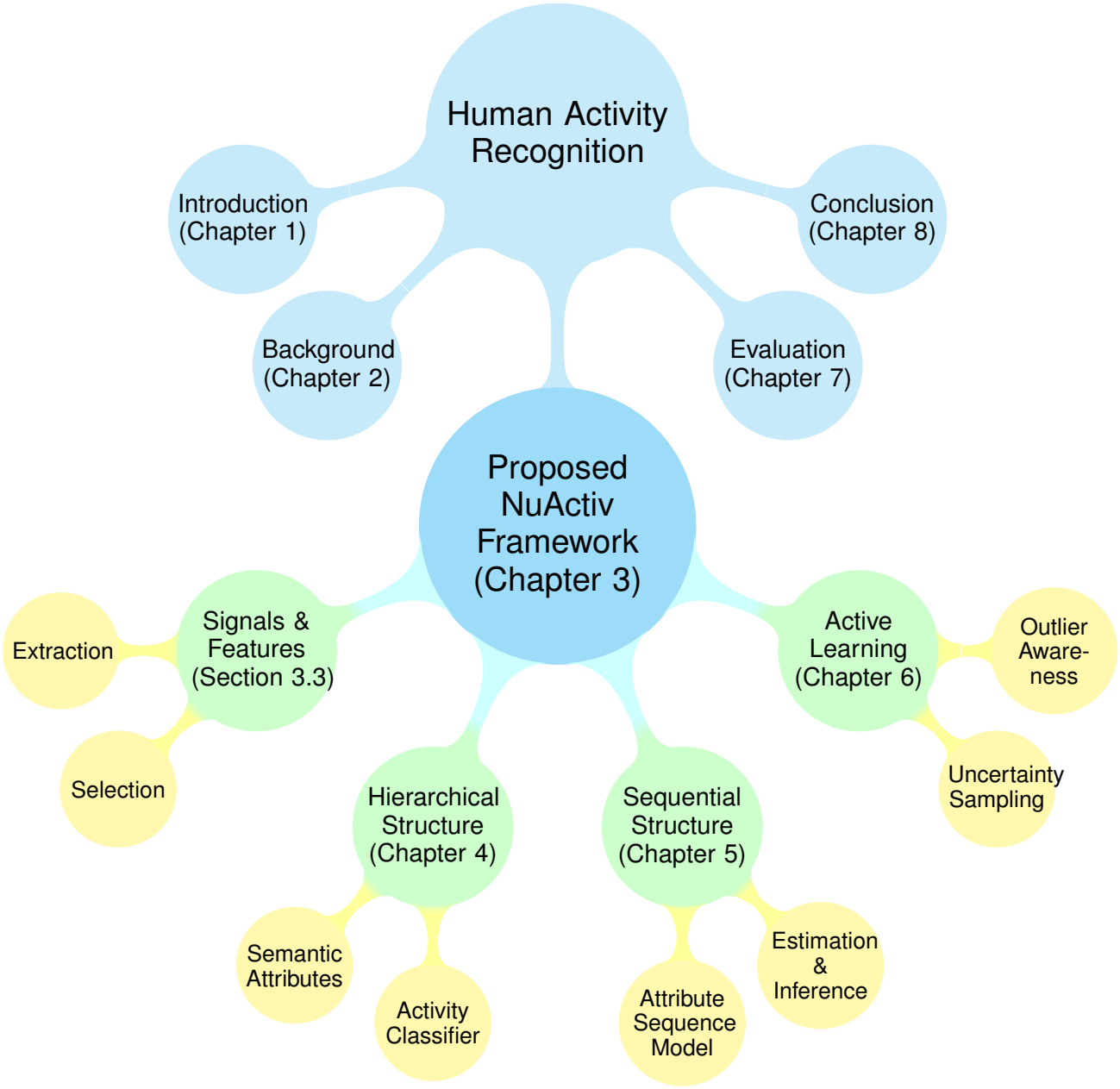


Figure 1.2: Thesis roadmap.

Chapter 2

Background and Related Work

This thesis lies in the intersection of two areas. One is human activity recognition in the field of mobile and ubiquitous computing, and the other is attribute-based zero-shot learning in the field of machine learning. This chapter will review the background and related work in the two areas.

2.1 Human Activity Recognition

The general definition of human activity recognition is to output a label of human activity Y given a set of input observations X. The high-level conceptual block diagram of human activity recognition is shown in Figure 2.1. The two main parts are *sensing* and *inference*.

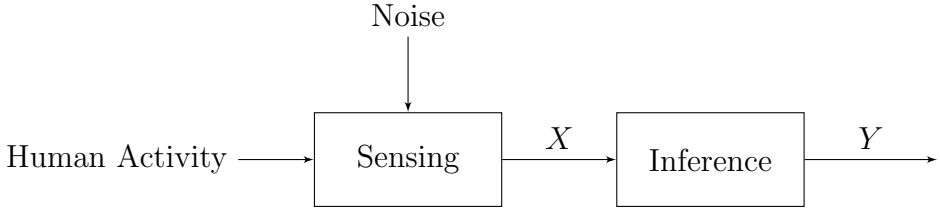


Figure 2.1: Conceptual block diagram of human activity recognition. X is the observed sensor data. Y is the recognized human activity.

The sensing part measures physical quantities that are correlated with the human activity and converts them into signals, through a combination of hardware and software components. The inference part takes the signals as input and outputs an estimation of the human activity that generated those signals. Since noise exists in the physical world and in the sensing process, successfully inferring the human activity from the signals is challenging. In recent years, extensive research has been done to make advances in both of these two parts. These advances can be viewed from various aspects, including sensing modality, sensor placement, user engagement, and learning paradigms. We will review these aspects in the following sections.

2.1.1 Taxonomy

The related work in human activity recognition can be categorized in terms of several aspects. This section will go through each of these aspects and describe where this thesis is positioned in the problem space and design space.

Vision-Based vs. Non-Vision Sensor-Based Activity Recognition

Activity recognition systems can be broadly categorized into vision-based systems [2, 88] and sensor-based systems [17, 44]. In this categorization, sensor-based approaches usually mean non-visual sensors.

There has been extensive research on vision-based activity recognition because of the advances in computer vision and the rich information contained in video content [88]. One of the most important applications of vision-based activity recognition is video surveillance. Another application is human-computer interaction for entertainment, such as the Kinect system [101]. While having shown success in the aforementioned applications, vision-based systems do have several main issues and limitations found by researchers [44]. The first issue is *privacy*, because many users do not want to be monitored or videotaped by cameras. In

other words, the rich information contained in video raises the concern for privacy at the same time. The second issue is *pervasiveness*, since it is difficult to install cameras to target the users in order to obtain images of their entire body during their daily living activities [44]. Also, the users need to stay within the view angle and a certain range of the camera, depending on the capability of the camera. The third issue is *computational complexity*, as video processing and understanding are more computationally expensive than processing audio alone or other non-vision signals. This can be a disadvantage in real-time or mobile applications with certain resource or time constraints.

Non-vision sensor-based approaches have different strengths and weaknesses compared to vision-based approaches, and are suitable for different types of applications. The early work on sensor-based activity recognition dates back to the 1990s [20, 34, 44]. Although non-vision sensor data generally do not contain as rich an information content as video, the sensors are usually easier to attach to human bodies or install in many other places in the surroundings. The privacy concern is usually less severe because it is harder to reconstruct the scene or the user identity from the sensor data. The computational complexity is also lower than vision-based approaches.

As a result, non-vision sensor-based approaches have received a lot of attention in recent years, with promising applications in context-aware services, recommenders, advertising, healthcare, and fitness. The focus of this thesis is on making advances in non-vision sensor-based activity recognition, with applications in context-aware, mobile, and ubiquitous computing.

External vs. On-Body Sensing

In terms of the relative position between the measuring sensors and the subject being measured, there are two types of approaches: One that uses *external sensors*, and the other that uses *on-body sensors*. According the the definition [44], in the former, the sensing devices

are usually fixed at predetermined points of interest, so the inference of activities depends on the interaction of the users with the sensors. In the latter, the devices are attached to the user, and the activity recognition systems are usually running in the background to detect the user's activities using opportunistic sensing [43].

External sensors usually include video cameras, microphones, and signal transceivers (infrared, ultrasound, and radio-frequency signals). One typical example is smart home or smart office systems that monitor the activities of users using the ambient sensors installed in the building.

On-body sensors usually include but are not limited to: inertial sensors (accelerometers and gyroscopes) and location sensors (e.g. GPS, cellular network, and Wi-Fi transceivers). Typical devices used for on-body sensing include the mobile phones that users carry with them, sensor-enabled wristwatches, wristbands, glasses, shoes, and other wearable devices. The motivating scenarios and system designs in this thesis are based primarily on the on-body sensing scheme, though the general ideas of attribute-based learning and active learning approaches can be applied to external sensing schemes as well.

Participatory and Opportunistic Sensing

Another aspect in the design space of activity recognition and mobile sensing is the level of user engagement, which can be categorized into *participatory sensing* and *opportunistic sensing*. As defined in [43], in participatory sensing, a user actively engages in the data collection activity. That is, the user manually determines how, when, what, and where to sample, and may also provide ground-truth labels to facilitate model training. In opportunistic sensing, the data collection stage is automated and running in the background with little to no user involvement. In this thesis, we focus on the scenario of opportunistic sensing, where the system aims to sense and infer the user's activities in the background using mobile and wearable devices, with minimal user engagement in the sensing process.

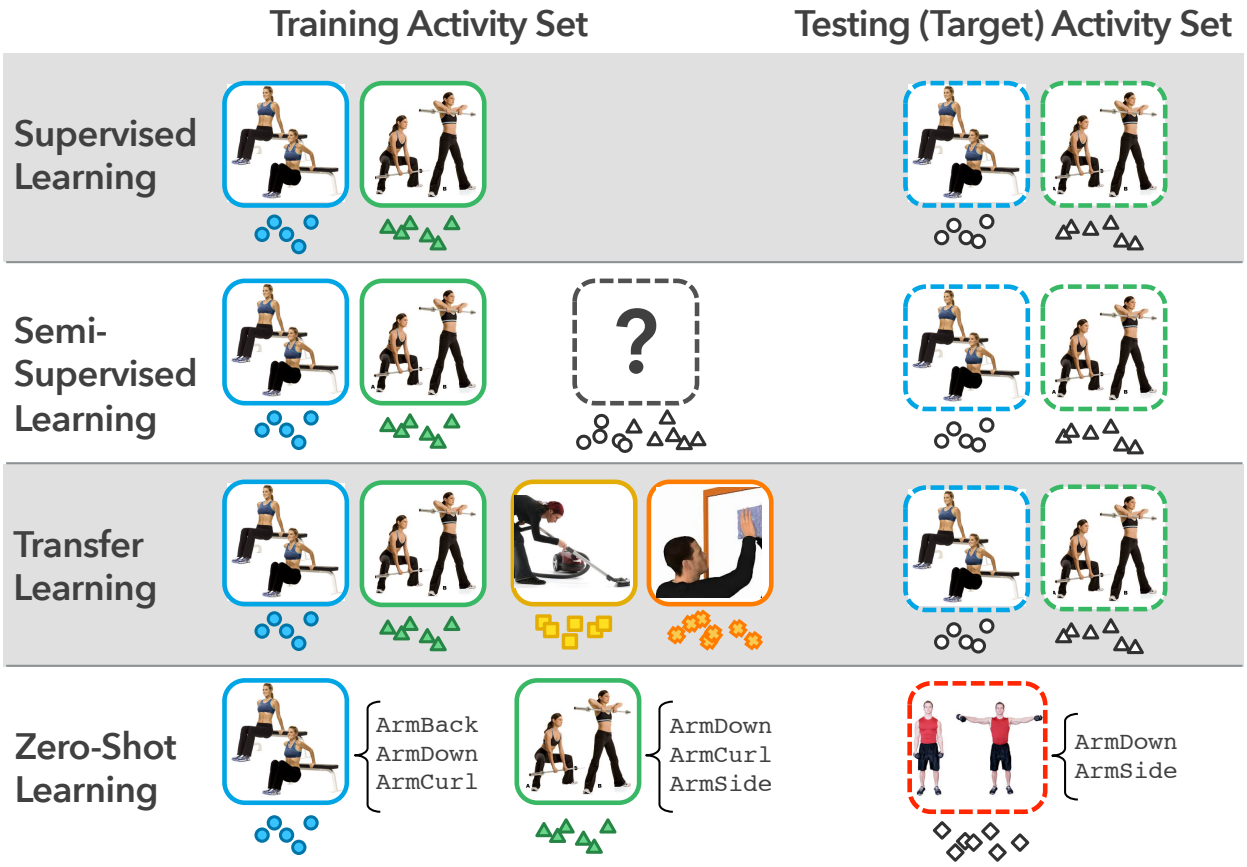


Figure 2.2: Comparisons of different problem settings and learning methods for activity recognition. Different shapes represent samples of different classes. Solid and hollow shapes represent labeled and unlabeled samples, respectively.

Learning Paradigm

Learning paradigm is about how the model for activity inference is acquired. Most of the approaches are machine learning-based, which include supervised learning, semi-supervised learning, unsupervised learning, transfer learning, and active learning. One of the main contributions of this thesis is introducing the attribute-based zero-shot learning to the field of human activity recognition. The comparison of different problem settings and learning methods for activity recognition is shown in Figure 2.2. As illustrated in the figure, supervised learning only learns from labeled samples (solid shapes) of the target activities.

Semi-supervised learning leverages additional unlabeled samples (hollow shapes). Transfer learning improves recognition using data in a different but related task. The proposed zero-shot learning method generalizes learned mid-level attributes to recognize unseen new activities. The next few subsections will go into a deeper look at each of the existing learning paradigms, and explain how this study extends or differ from the previous work.

Table 2.1: Summary and comparison of related work in human activity recognition.

Ref.	Learning Paradigm	Models/ Algorithms	Activity Domain/Classes	Input Sensor Data/Devices	Results
[4]	Supervised Learning	k-NN, C4.5 decision tree, naive Bayes	20 daily life activities	Accelerometer on upper arm, wrist, thigh, leg, hip	50-80% accuracy
[56]	Supervised Learning	AdaBoost + hidden Markov model (HMM), C4.5 decision tree + HMM	15 daily life activities	Camera, microphone, accelerometer, compass on wristband	70-80% accuracy
[64]	Supervised Learning	k-nearest neighbor (k-NN), SVM, Decision Trees	10 upper body exercise activities	Accelerometer and gyroscope on upper arm band	94% accuracy
[13]	Supervised Learning	Naive Bayes, hidden Markov models	9 free-weight exercise activities	Accelerometers on glove and belt	90% accuracy
[54]	Supervised Learning	Decision tree, Gaussian mixture model (GMM), SVM, naive Bayes	5 motion classes, 7 sound event classes	Accelerometer, microphone, GPS on mobile phone	Accuracy: 94% (motion), 84% (sound)
[96]	Supervised Learning	2-tier framework + classifiers (Decision tree, AdaBoost, SVM, naive Bayes, Bayesian network)	12 home and office activities	Accelerometers in pockets (front shirt, front pants, back pants)	77% accuracy

Ref.	Learning Paradigm	Models/ Algorithms	Activity Domain/Classes	Input Sensor Data/Devices	Results
[83]	Semi-Supervised Learning	Multi-Instance support vector machine (SVM), graph-based label propagation	PlaceLab dataset [51], 34 daily life activities[33]	Accelerometers on wrist, hip, thigh	Up to 70-80% accuracy
[84]	Semi-Supervised Learning	Self-training, co-training + pool-based active learning	9 activities in PlaceLab dataset [51]	Accelerometers on wrist, hip, thigh	40-60% accuracy
[57]	Semi-Supervised Learning	Semi-supervised virtual evidence boosting (sVEB), CRF	8 basic physical activities (sitting, walking, running, etc.)	Wearable sensors (audio, light, acceleration, pressure, temperature, humidity)	70-85% accuracy
[8]	Transfer Learning	Layered conditional random field (CRF)	Bookshelf building, mirror building	Accelerometer and gyroscope on upper arms, forearms, top back	Outperforms 1-layer model by 10% EER [79]
[33]	Unsupervised Learning	k-means clustering, latent Dirichlet allocation (LDA)	34 daily life activities	Accelerometers on wrist and hip	77% precision, 66% recall
[61]	Unsupervised Learning	Motif discovery + HMM	6 dumbbell exercise activities	Accelerometer on wrist	88% precision, 96% recall
[95]	Unsupervised Learning	HMM	26 daily life activities	RFID reader bracelet, RFID tags on home objects	52% accuracy
[102]	Unsupervised Learning	Hierarchical Bayesian network	Reality Mining dataset [18]	Call logs, Bluetooth, cell tower IDs, application usage, phone status	$\sim$70% area under curve (AUC) [10]

Ref.	Learning Paradigm	Models/ Algorithms	Activity Domain/Classes	Input Sensor Data/Devices	Results
[32]	Unsupervised Learning (Nonpara-metric)	Hierarchical Dirichlet process HMM, one-class SVM	PlaceLab dataset, abnormal activity dataset [98]	Sensors (light, temperature, microphone, accelerometer) on shoulder, waist, thigh	0.83-0.86 AUC

2.1.2 Supervised Learning

In the field of mobile, wearable, and pervasive computing, extensive research has been done to recognize human activities (e.g. sitting, walking, running) [4, 8, 43, 54, 60, 72, 83, 84, 92]. In terms of the learning method, the majority of the research in this field used supervised learning approaches, including discriminative classifiers (e.g. Decision Trees, SVM) and generative models (e.g. Naive Bayes, Hidden Markov Model), where a classifier is trained on a large set of labeled examples of every target activity [4, 6, 46, 51, 54, 60, 92]. There has also been prior study of representing high-level activities as a composite of simple actions, using a supervised layered dynamic Bayesian network [91]. While many promising results have been reported, a widely acknowledged problem is that labeled examples are often time consuming and expensive to obtain, requiring a lot of effort from test subjects, human annotators, or domain experts [83, 84].

2.1.3 Semi-Supervised and Transfer Learning

To lessen the reliance on labeled training data and to exploit the benefits of abundant unlabeled data, previous work has incorporated semi-supervised learning into activity or context recognition systems [52, 57, 59, 83, 84]. Semi-supervised learning approaches can

improve the recognition accuracy by refining the decision boundary based on the distribution of the unlabeled data, or by assigning highly-confident estimated labels to the unlabeled data. Recently, transfer learning has also been explored so that a model learned for one target class can be transferred to improve the recognition accuracy of another target class [8, 103]. As a result, the amount of training data required for new applications can be reduced. While many promising results have been reported, most of the existing approaches can only recognize activity classes that were included in the training data. Inspired by previous study, our work presents an early attempt to recognize unseen human activities with no training data using attribute-based zero-shot learning.

2.1.4 Active Learning

The idea of active learning algorithms is that a machine learning algorithm can perform better with less training data if it is allowed to choose the data from which it learns [80]. Active learning has been used to improve the accuracy of human activity recognition [50, 52, 84] or to model the interruptibility of a mobile phone user [74]. We extend the previous work by incorporating active learning in the framework of zero-shot learning for activity recognition, so that the system is able to not only recognize unseen activities but also actively request labels from users when possible.

2.1.5 Unsupervised Learning

Another related research direction is unsupervised learning. Unsupervised learning focuses on clustering or pattern discovery rather than classification [33, 61]. In [38], human activity understanding is divided into activity recognition and activity pattern discovery. The first category focuses on accurate detection of human activities based on a pre-defined or pre-trained activity model, while the second category focuses on finding unknown patterns

directly from low-level sensor data.

One major area using unsupervised learning is routine discovery, which aims to extract temporal regularities in people's daily lives. A routine can be seen as a combination of multiple low-level activities, with different proportions from one routine to another. For example, the "Grocery Shopping" routine may involve more "standing" and "walking" activities than the "Office Work" routine.

Most existing approaches for unsupervised routine discovery are based on parametric topic models such as probabilistic latent semantic analysis (pLSA) [29, 30] or latent Dirichlet allocation (LDA) [9]. In [33], the authors discover daily routines from wearable sensor data by first building an activity vocabulary using k-means clustering, and then using LDA to learn topic proportions for each window of sensor data, which is analogous to a document containing a collection of words.

Farrahi et al. [19] also apply LDA on labeled cell tower data to automatically discover routines, including "being at work" or "going home from work". Zheng et al. [102] propose a probabilistic generative model for learning users' latent behavior patterns based on unlabeled cell tower data. One limitation of these methods is that they all require a heuristic process for parameter selection, such as the size of vocabulary, the number of topics, and the number of typical states of a user.

More recently, nonparametric methods have been proposed and applied in the field of activity recognition and mobile computing. The major benefit of nonparametric approaches is that the models can be applied without having to find the ideal parameters (e.g. number of topics or clusters) beforehand. For example, Hu et al. [32] tackles the abnormal activity recognition problem by using hierarchical Dirichlet process hidden Markov model (HDP-HMM) to automatically decide the right number of states for HMM. Similarly, Zhu et al. [104] segment activity sensor reading sequence and group the segments into meaningful categories by leveraging Sticky HDP-HMM.

Although labels are not required for unsupervised learning approaches, the output of these approaches is a set of unnamed clusters which cannot be used for classification or recognition purposes. To perform recognition, labels are still needed to connect the discovered patterns to the actual classes.

2.1.6 Rule-Based Approach

There are also some rule-based approaches to activity recognition. In [85], Storf et al. proposed a multi-agent-based framework using rules and manual configurations written in the Extensible Markup Language (XML) format. For example, the detection of "a person staying in front of the kitchen counter" is formulated as a rule of whether the pressure mat detects a person standing on it for longer than some specified time threshold (e.g., ten seconds). The authors also use fuzzy reasoning. For example, the detection of the activity "preparing meal" involves a set of cases and rules, including the combination of `usage_stove`, `usage_fridge`, `stay_at_kitchen_counter`, etc. with different weights for each case. While simple and suitable for certain condition-action pairs, rule-based approaches may be hard to apply without much domain knowledge, or when the rules are not straightforward and thus have to be learned from data.

2.1.7 Human Activity Domain

In terms of the activity domain of interest, some previous work in the area of human activity recognition focused on daily life activities [33, 51, 83] and some focused on sports and exercise activities [13, 61, 64]. In this thesis, the proposed approach is evaluated in both activity domains to validate its effectiveness for general unseen activity recognition.

2.2 Zero-Shot Learning

The idea of zero-shot learning has recently been explored and has been shown to be useful for recognizing unseen new classes [21, 27, 36, 45, 67, 68]. Palatucci et al. presented one early study on the problem of zero-shot learning [68], where the goal is to learn a classifier that can predict new classes that were omitted from the training dataset. A theoretical analysis was done to study the conditions under which a classifier can predict novel classes. In a case study, the authors studied the problem of decoding the word that a human is thinking of using functional magnetic resonance images (fMRI). The authors used multiple output linear regression to learn the mapping from raw image data to semantic attributes (referred to as the semantic output code classifier [68]). That is, a weight matrix $\mathbf{W}$ is learned by the matrix operation over the training data:

$$\mathbf{W} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{A} \tag{2.1}$$

where $\mathbf{I}$ is the identity matrix and λ is a regularization parameter which avoids overfitting. $\mathbf{X} \in \mathcal{R}^{n\times d}$ is a training set of n fMRI examples where each row is the image for a particular class and d is the number of dimensions of the fMRI image. $\mathbf{A} \in \mathcal{R}^{n\times p}$ is a matrix of semantic attributes for those classes, where p is the number of semantic attributes. After training, given a novel fMRI image $\mathbf{x}$, we can obtain a prediction $\hat{\mathbf{a}}$ of the semantic attributes for this image by multiplying the image $\mathbf{x}$ by the weights $\mathbf{W}$:

$$\hat{\mathbf{a}} = \mathbf{x}\mathbf{W} \tag{2.2}$$

Inspired by their work, our work extends the zero-shot learning framework to handle sequential data by modeling the sequence and structure of the attributes. Furthermore, while most previous work used only mid-level attributes for classification, we explore a hybrid attribute/feature-based activity recognition algorithm to improve the classification

accuracy on both seen and unseen classes.

2.3 Semantic Attributes

The idea of mid-level semantic attributes has shown promise in the field of computer vision, including object and scene recognition [22, 42, 71, 75], neural activity recognition [68], and human action recognition from video frames [22, 49, 97]. For example, in computer vision, the general idea is to recognize a visual object through a set of human-specified semantic descriptions (such as shape, color, and texture) of the target objects instead of directly from the raw images [42]. Compared to our work, these problem domains are inherently different from activity recognition because image data are static rather than sequential data. Furthermore, the image attributes and description for visual objects cannot be directly applied to the inertial/motion sensor data in activity recognition.

Inspired by these previous studies, we designed and implemented a new activity recognition system using the concept of attribute-based zero-shot learning. In addition, we investigate the use of attribute sequence models compared to the bag-of-attributes models used in the related work. The next chapter will introduce *NuActiv*, the new human activity recognition framework proposed in this research.

Chapter 3

Human Activity Recognition Framework

In the previous chapter, we went through the background of human activity recognition and the related work in the field of ubiquitous computing and machine learning. In this chapter, we will first present the scenarios and design considerations that motivate our study and proposed approach. Then, we will give an overview of the proposed *NuActiv* human activity recognition framework.

3.1 Scenarios and Design Considerations

The goal of this work is to study and design a general human activity recognition framework in the field of mobile, wearable, and pervasive computing. The learning and recognition framework is independent of sensor data types or device types, so the source of sensor data is not limited to mobile phones but can also be wearable devices. Wearable mobile devices are becoming increasingly available in the commercial market [3]. Phones and devices can be worn as wrist watches (e.g., MotoACTV [63]), glasses (e.g., Google Glass [25]), and more.

Advances in nano-technology are further driving this trend by introducing flexible materials. These new wearable devices enable a wide range of context sensing, inference, and pervasive computing applications [43]. With these considerations in mind, in this work we choose phones and wristwatches with inertial sensors as examples to demonstrate two scenarios of activity domain: *daily life activities* and *exercise activities*.

The first scenario is daily life activity monitoring [33, 51]. Suppose we have the training data for two activities "`ReadingAtHome`" and "`Driving`". What if we want to detect if the user is "`ReadingOnTrain`"? Instead of hiring subjects to collect and label new sensor data, our goal is to directly recognize the new activity class "`ReadingOnTrain`" by reusing the semantic attributes within the model that we already trained for "`ReadingAtHome`" and "`Driving`". For example, the pattern of wrist motion when reading a book may be consistent regardless of the user location. In this case, "`ReadingAtHome`" and "`ReadingOnTrain`" may share the same wrist motion attribute. Furthermore, the velocity or acceleration experienced by a person on a vehicle, though different depending on the vehicle type, may be easily distinguishable from walking, running, or biking. Therefore, the attribute "moving at vehicle speed" may be found in both the "`Driving`" and "`ReadingOnTrain`" activities.

The second scenario is exercise activity detection. Detecting physical exercises and sports activities is useful for health and fitness monitoring applications [13, 64]. Through experiments on real-world sensor data, we will show that our semantic attribute-based learning applies well to this activity domain because many exercise activities are built up by the same underlying attributes, as illustrated in Figure 2.2.

Daily life activities are of greater interest because they comprise the major part of people's lives. On the other hand, daily life activities are also arguably of much larger variation because different people do the same things differently, and thus are harder to recognize. Even the same person can do one activity differently at different times. In this research, we started by testing our system and algorithms for the exercise activity scenario because

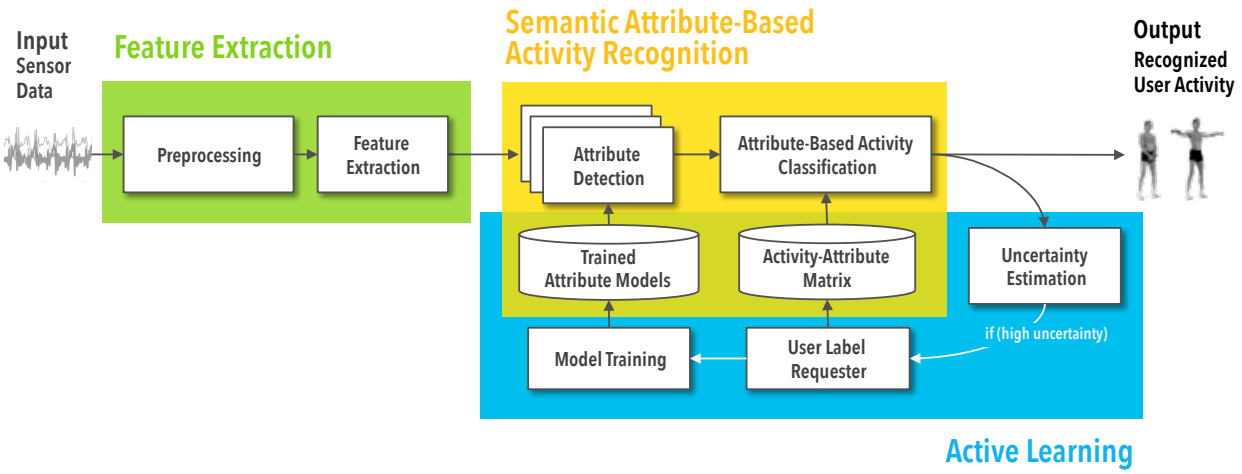


Figure 3.1: The *NuActiv* activity recognition framework.

the activities are well-defined, repeatable, and typically of lower variation among different people. After observing the effectiveness of our approach on exercise activities, we further generalized the approach to daily life activities.

3.2 The *NuActiv* Framework

Based on the aforementioned design considerations, a new activity recognition framework, *NuActiv*, is proposed in this thesis. The NuActiv framework, as shown in Figure 3.1, consists of three main components:

(1) Feature Extraction: This component preprocesses the raw sensor data (or signals) from various sensor inputs, and extracts low-level features from the processed sensor data. (Section 3.3).

(2) Semantic Attribute-Based Activity Recognition: This component can be further divided into two parts. The first part is *Attribute Detection*, which transforms low-level features into a vector of human-readable semantic attributes. The second part is *Attribute-Based Activity Classification*, which classifies the detected attribute vector as one of the

activity classes given the *activity-attribute matrix*, even if no training data exist for some of the target activity classes. (Chapter 4 and Chapter 5).

(3) Active Learning: Given the output recognized user activity class, the active learning component estimates the uncertainty of the recognition result. When the result is estimated to be uncertain using a predefined metric, the *user label requester* prompts the user for feedback or ground-truth labels. The labels are then used for re-training and updating models for attribute detection and activity classification. The function of this component is to reinforce activity recognition accuracy using minimal user feedback (Chapter 6).

3.3 From Signals to Features: Feature Extraction

The activity recognition framework described in this thesis is agnostic to input data type. Any kind of sensor data (or signals) can be fed into the system for learning an activity recognition model. We select inertial sensor data, including accelerometer and gyroscope, from two activity domains—exercise activities and daily life activities—as examples to demonstrate the effectiveness of the proposed framework.

3.3.1 Sensors

Accelerometer

An accelerometer is a sensor that measures the acceleration experienced by an object relative to a free-falling frame of reference. A triaxial accelerometer installed on a mobile or wearable device returns a real-valued estimate of acceleration along the x, y, and z axes (as shown in Figure 3.2) in units of meter per second squared (m/s^2) [72].

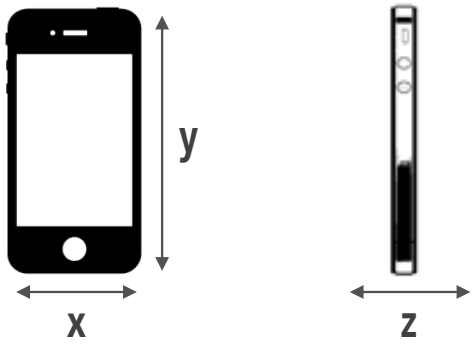


Figure 3.2: Illustration of the coordinate system of an accelerometer on a mobile phone.

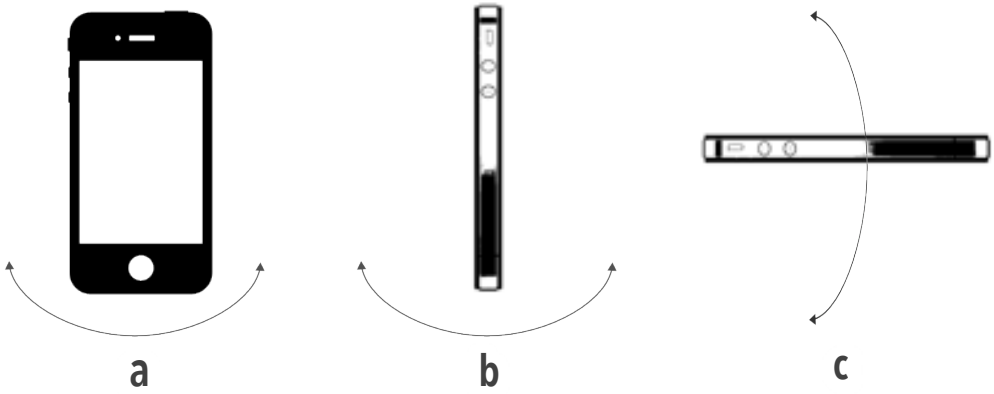


Figure 3.3: Illustration of the coordinate system of a gyroscope on a mobile phone.

Gyroscope

A gyroscope is a device that can be used to either measure, or maintain, the orientation of an object [5, 47]. Also known as angular rate sensors or angular velocity sensors, gyroscopes can sense the angular velocity along the a, b, and c axes of a mobile or wearable device (as shown in Figure 3.3), in units of radian per second (rad/s).

3.3.2 Motion Features

For an input stream of sensor data, it is assumed that the samples within a window (i.e. a segment of the input sensor data stream) are a sequence of samples drawn from some random

variable X. Using a sliding window of length n, features are computed from a sequence of n samples x_t starting at time $t = t_0$, namely $\{x_{t_0}, x_{t_0+1}, \ldots, x_{t_0+n-1}\}$, where t is the discrete index in the time domain. For each dimension in the coordinate system of accelerometer and gyroscope, we compute a set of motion features that are widely used and shown useful in the related work. The feature set includes:

- The mean of the sensor data. Let the sequence of any dimension of the sensor data in a window be $(x_t)_{t=t_0}^{t_0+n-1}$, which are regarded as samples drawn from a random variable X. The mean of the random variable X is computed as:

$$\overline{X} = E(X) = \frac{1}{n} \sum_{t=t_0}^{t_0+n-1} x_t \tag{3.1}$$

- The standard deviation. Using the same notation above, it is calculated as:

$$S_X = \sqrt{\mathrm{Var}(X)} = \sqrt{\frac{1}{n-1} \sum_{t=t_0}^{t_0+n-1} (x_t - \overline{X})^2} \tag{3.2}$$

- The pairwise correlation between each pair of dimensions. Let the sequences of any two dimensions of the sensor data be $(x_{i,t})_{t=t_0}^{t_0+n-1}$ and $(x_{j,t})_{t=t_0}^{t_0+n-1}$.

$$\begin{aligned} \rho_{X_i,X_j} &= \frac{\mathrm{Cov}(X_i, X_j)}{S_{X_i} S_{X_j}} = \frac{E[(X_i - \overline{X_i})(X_j - \overline{X_j})]}{S_{X_i} S_{X_j}} \\ &= \frac{\sum_{t=t_0}^{t_0+n-1} (x_{i,t} - \overline{X_i})(x_{j,t} - \overline{X_j})}{\sqrt{\sum_{t=t_0}^{t_0+n-1} (x_{i,t} - \overline{X_i})^2 \sum_{t=t_0}^{t_0+n-1} (x_{j,t} - \overline{X_j})^2}} \end{aligned} \tag{3.3}$$

- The local slope of sensor data using 1st-order linear regression. That is, find parameters $\mathbf{w} = \{w_0, w_1\}$ that define a line $X = w_0 + w_1 t$ which best fits the sensor data in the window. w_1 is then used as the local slope feature.

$$w_1 = \frac{\sum_{t=t_0}^{t_0+n-1} (t - \overline{t})(x_t - \overline{X})}{\sum_{t=t_0}^{t_0+n-1} (t - \overline{t})^2} \tag{3.4}$$

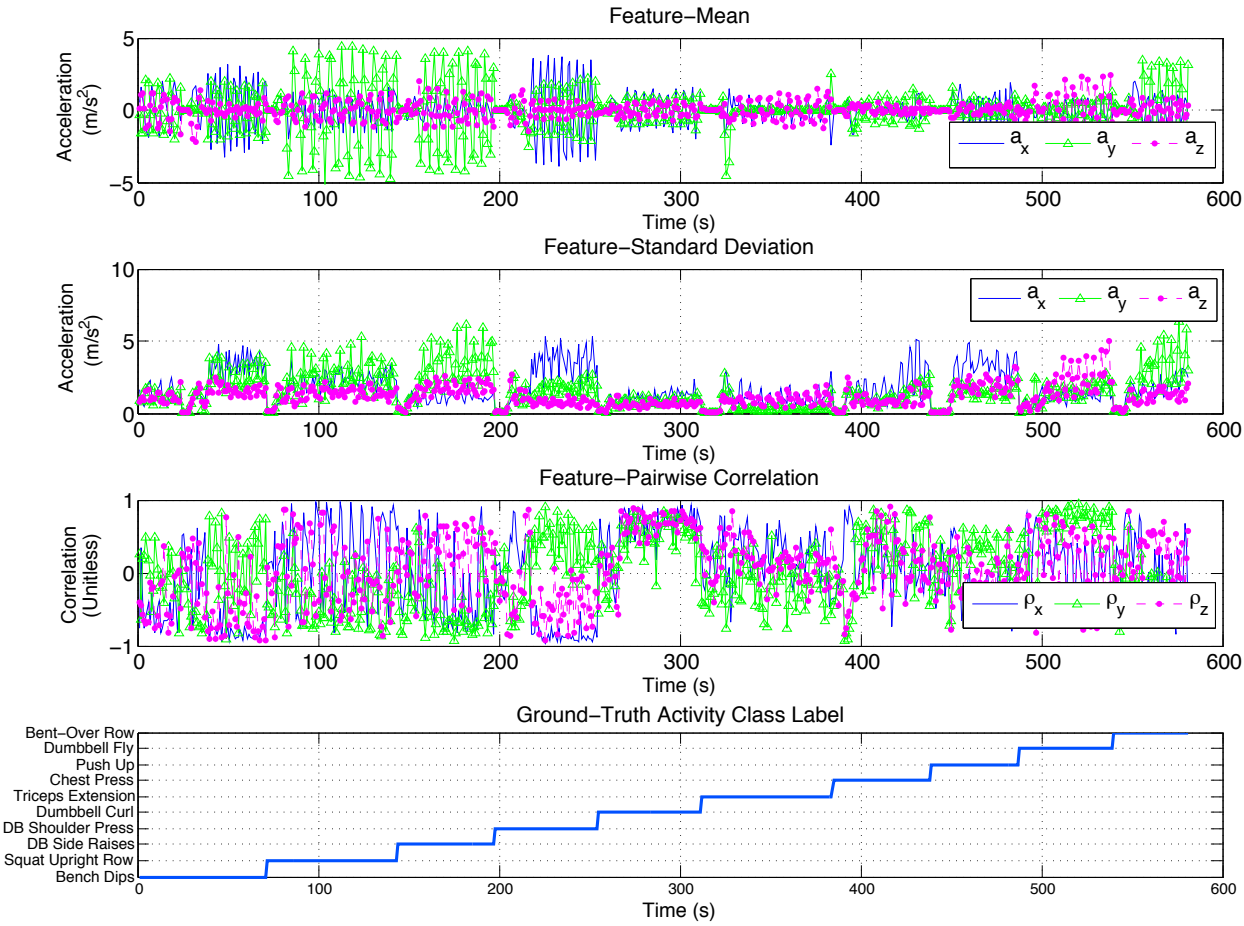


Figure 3.4: Examples of features extracted from acceleration data for each exercise activity.

- The zero-crossing rate.

$$ZCR(X) = \frac{1}{2(n-1)} \sum_{t=t_0+1}^{t_0+n-1} |\text{sgn}(x_t) - \text{sgn}(x_{t-1})| \tag{3.5}$$

Some examples of extracted features are shown in Figure 3.4. The sensor data and settings used in each dataset are described in Section 7.2. Similar features have been used in the related work [72, 83, 100]. To capture the temporal dynamics of the features, the n^{th}-order temporal features are further included. Specifically, the feature vector at time t is concatenated with those at time $t-1, t-2, \ldots, t-n$ (n is empirically set to 2 using a 10-fold cross validation on the validation set in our experiments).

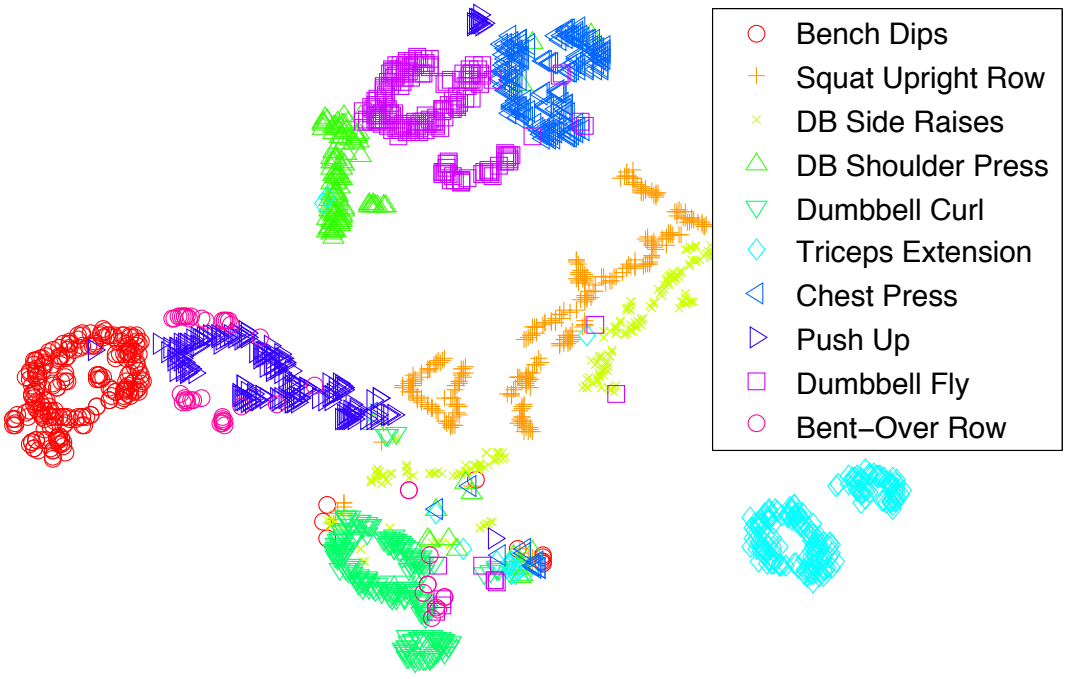


Figure 3.5: Visualization of exercise activity samples in the feature space.

To visualize the effectiveness of the features, we plot the dataset in the feature space in Figure 3.5. The dimensionality is reduced to 2 for visualization using t-Distributed Stochastic Neighbor Embedding (t-SNE) [55]. t-SNE visualizes high-dimensional data by giving each datapoint a location in a two or three-dimensional map. The technique is a variation of Stochastic Neighbor Embedding (SNE) [28] that is easier to optimize, and generates better visualizations by reducing the likelihood of crowding points together in the center of the resulting map. It can be observed from the figure that the samples form natural clusters with minor overlaps, showing that these features do discriminate.

3.3.3 Time Features

Time is an important factor in human activities. The activities that people engage in often exhibit patterns that are correlated with the time of day and day of week [33]. For example,

many people regularly have lunch around 12 p.m. and dinner around 6 p.m. Most people commute at a certain time of day during the weekdays because of work or school, and engage in some entertainment activities during the weekends. This is all very useful information that one can leverage to infer a person's activities or to rule out the unlikely ones.

The "raw data" of time on most computer systems is a Unix timestamp, defined as the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, January 1st, 1970. Widely used time features extracted from the raw data include time of day, day of week, weekday/weekend, or whether the day is a holiday or not. In our experiments, we use time of day as the time feature for the daily life activity dataset [33], as it carries important information about the daily life routines of a user.

3.3.4 Feature Selection

Selecting the right set of features is important for improving the recognition accuracy [100]. One approach to feature selection is based on the mutual information between the features and the class labels. Based on the co-occurrence of feature values and class labels, we compute the score of each feature based on the normalized mutual information between the class label and the feature.

Entropy, in information theory, is a measure of uncertainty in a random variable. In a feature selection problem, the entropy of the class random variable Y, $H(Y)$, is defined as:

$$H(Y) = E(I(Y)) = \sum_{y \in \mathbb{Y}} I(Y) P(Y = y) = \sum_{y \in \mathbb{Y}} \log \left(\frac{1}{P(Y = y)} \right) P(Y = y) \qquad (3.6)$$

where $\mathbb{Y}$ is the set of all possible values that Y can take on, and $I(Y)$ is the self-information of Y. $H(Y)$ is the intrinsic uncertainty of Y. When we observe some set of features X, the remaining uncertainty of Y when X is known is $H(Y|X)$, which is defined as:

$$H(Y|X) = \sum_{x \in \mathbb{X}} P(X = x) H(Y|X = x) \qquad (3.7)$$

where

$$H(Y|X=x) = \sum_{y \in \mathbb{Y}} \log\left(\frac{1}{P(Y=y|X=x)}\right) P(Y=y|X=x) \tag{3.8}$$

If the uncertainty of Y is reduced, it's easier to predict Y. Therefore, a reasonable way to measure the effectiveness of an feature is to use the Information Gain, $H(Y) - H(Y|X)$. However, one issue of using information gain is that it gives an unfair advantage over X with many keys (i.e. an X that can take on a large number of possible values). We tackle this problem by using the Normalized Mutual Information, which is defined as:

$$\hat{I}(Y;X) = \frac{I(Y;X)}{H(X)+H(Y)} = \frac{H(X)+H(Y)-H(X,Y)}{H(X)+H(Y)} = \frac{H(Y)-H(Y|X)}{H(X)+H(Y)} \tag{3.9}$$

Since $H(Y)$ is a constant value that does not depend on X, the feature X that has the lowest $H(Y|X)$ will have the most information gain $H(Y) - H(Y|X)$. This information gain will then be normalized by the denominator $(H(X)+H(Y))$. Once we have the score defined by the normalized information gain, we can rank the features by their scores.

While intuitively reasonable and computationally inexpensive, the aforementioned approach does not necessarily arrive at an optimal subset of features. Given a total number of D features, the optimal feature subset can only be obtained by doing a full search over all 2^D possible subsets (because each feature can either be included or excluded from the subset), which can quickly become computationally intractable as D increases.

To solve the computational intractability, two widely used heuristic approaches that approximate the full search are the *forward search* and the *backward search* algorithms [39]. As the name suggests, the forward search algorithm starts with an empty feature subset, and iteratively adds features one at a time. At each iteration, the feature to be added is the one that leads to the lowest classification error on the validation set. Conversely, the backward search algorithm begins with a subset containing all of the features, and iteratively removes one feature at a time. In a similar manner, the feature to be removed at each iteration is

the one leading to the lowest classification error.

3.4 Modeling and Recognizing Activities

After the low-level features are extracted from the raw sensor data stream (the "signals"), the next step is to infer the high-level activity (the "semantics") from these features. This is achieved by building a statistical model that effectively describes the relationship between the signals and the semantics. As mentioned in Chapter 1, our goal is to build a activity model that is applicable to both supervised/semi-supervised learning as well as zero-shot learning problems, making it possible to recognize an activity with or without training samples in the dataset. In the next three chapters, I will present the proposed approach to model and recognize human activities with this goal in mind. Chapter 4 presents the main idea of using mid-level semantic attributes to model the hierarchical structure of human activities. In Chapter 5, the sequential structure of human activities is further taken into consideration. Finally, in Chapter 6 humans are brought back into the loop to efficiently improve the performance of an activity recognition system through active learning.

Chapter 4

Hierarchical Structure: Semantic Attribute-Based Activity Recognition

In the previous chapter, the *NuActiv* activity recognition framework and the methodology for feature extraction has been presented. In this chapter, we will introduce the core idea of mid-level semantic attribute-based activity recognition. The idea stems from the study on the hierarchical nature of human activities.

Human activities exhibit inherent hierarchical structures [8]. The activities that a person does often begin with one or multiple high-level goals or intents, which can be recursively broken down into a set of lower-level sub-activities or atomic actions [11, 105]. For example, a person may first have an intention to do some sports activity. The intention drives the person to perform sports activities, which may include playing baseball and playing golf. In the course of doing these activities, several sub-activities are performed. These sub-activities include swinging, pitching, walking, and running. Each sub-activity can then be further broken down into fine-grained motions of limbs, joints, and muscles.

Most existing approaches in activity recognition are based on training a k-way classifier (either directly or by combining multiple binary classifiers) to classify a low-level feature

vector as one of the k different high-level activity classes. Scaling such a classifier beyond a limited number of discrete categories remains an unsolved problem [21]. Furthermore, the problem is exacerbated by the fact that k-way discrete classifiers treat all labels as disconnected and unrelated [21].

These observations motivate the study of the hierarchical structure of human activities, as presented in this chapter. During the modeling phase, a high-level activity is decomposed into lower-level building blocks (i.e. semantic attributes), each of which in turn generates motion features according to certain probability distributions. The relationship between attributes and high-level activities is obtained by injecting human knowledge about the definitions and descriptions of the activities. During the recognition phase, we take the reverse route: starting from the low-level signal features, we try to infer the higher-level semantics of an activity one level at a time. To bridge the semantic gap between low-level sensor data and high-level human activities, we introduce the notion of mid-level *semantic attributes* that allows a smooth transition in the low-to-high-level semantic continuum.

4.1 Problem Definition

The problem of activity recognition can be formalized as follows. Let Y be the class label, a random variable that can be one of the k classes in the activity class space $\mathbb{Y} = \{y_1, y_2, \ldots, y_k\}$. $\mathbf{x}$ is a d-dimensional vector containing d input features in the feature space $\mathbb{X}$. We want to learn a classifier function $f : \mathbb{X} \rightarrow \mathbb{Y}$ where the function outputs an estimate or prediction of the class y given an input feature vector $\mathbf{x}$. Most of the existing approaches in activity recognition train the classifier f using a training dataset $D_{train} = \{(\mathbf{x}_i, y_i)|i = 1, 2, ..., N\}$, which contains N pairs of input features and ground-truth output class labels. If we have training instances for every class in $\mathbb{Y}$, we are able to train a classifier f. However, if there are no training data for a subset of classes $\mathbb{Y}$, we are not able

to predict those classes.

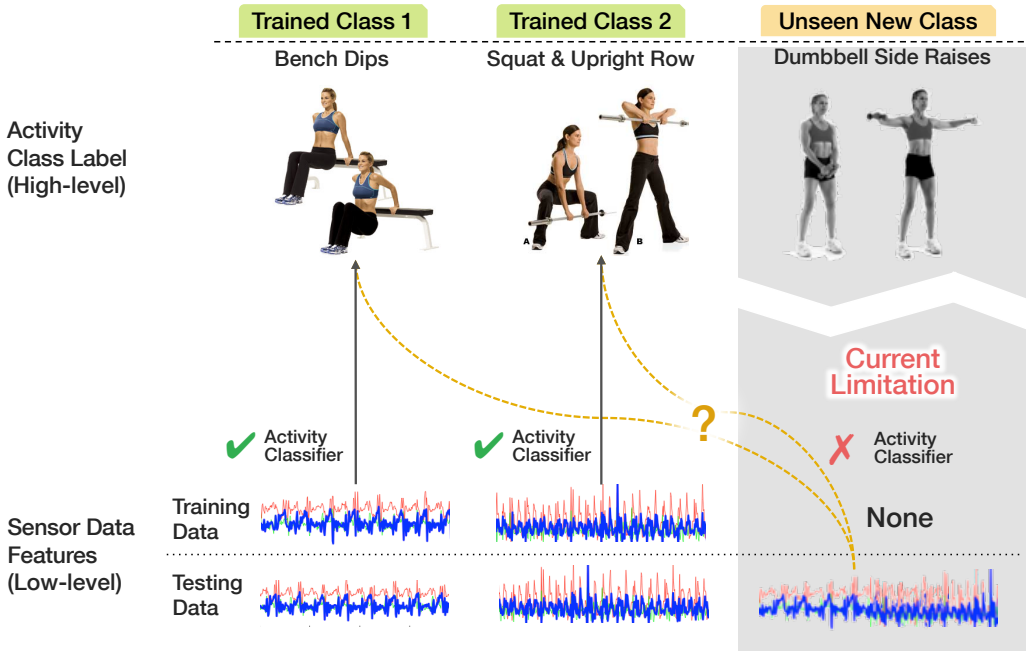
This research aims to solve the problem of recognizing previously unseen activity classes. Suppose $\mathbb{Y} = \{y_1, y_2, \ldots, y_s, y_{s+1}, \ldots, y_k\} = \mathbb{Y}_S \cup \mathbb{Y}_U$. $\mathbb{Y}_S$ is the set of *seen classes*, where there exists some training data for every class in $\mathbb{Y}_S$. $\mathbb{Y}_U$ is the *unseen classes* set where there are no training data for any class in $\mathbb{Y}_U$. The problem is: How to recognize an unseen class $y \in \mathbb{Y}_U$?

The idea is to first transform low-level features $\mathbf{x}$ into a vector of mid-level semantic attributes $\mathbf{a}$ in the attribute space $\mathbb{A}$. Each attribute corresponds to a human readable atomic physical motion or a specific characteristic of a complex activity. If every high-level activity in $\mathbb{Y}$ can be mapped to a point in the attribute space $\mathbb{A}$, then it is possible for us to recognize every activity class $y \in \mathbb{Y}$ given an accurately detected attribute vector $\mathbf{a}$. Since every semantic attribute in $\mathbf{a}$ is a human readable term, the mapping between y and $\mathbf{a}$ can be defined based on human knowledge without training data. Without this attribute layer, the direct mapping between y and $\mathbf{x}$ can only be trained with labeled sensor data, because x's are low-level signal features that are hard for humans to interpret directly.

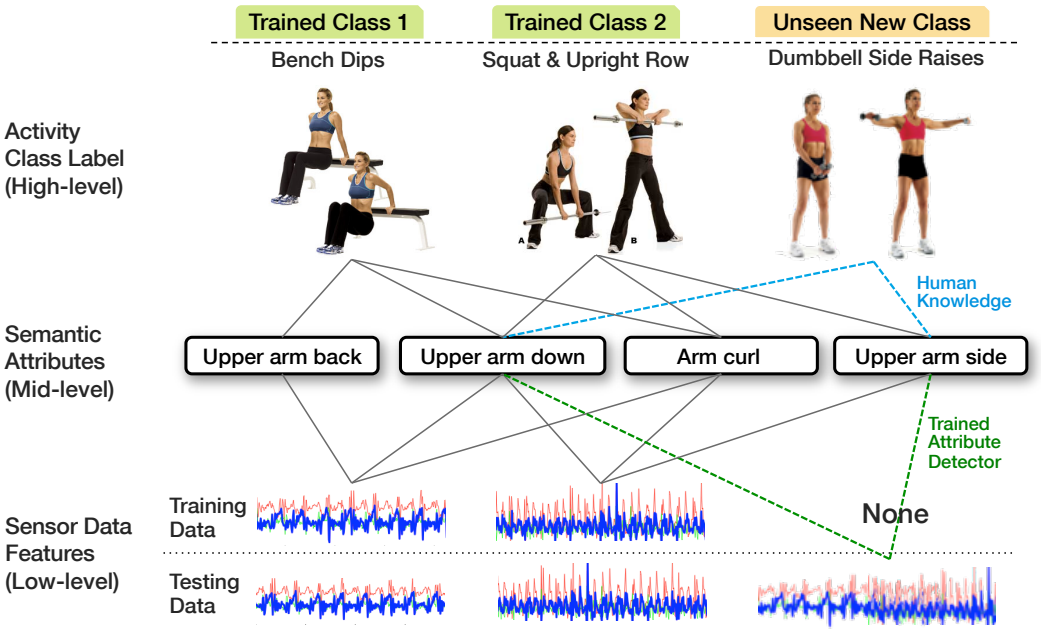
Figure 4.1 illustrates the difference between existing supervised-learning-based approach and our proposed semantic attribute-based learning approach, using exercise activities as examples. This fundamental idea of semantic attribute-based learning for activity recognition can be abstracted as illustrated in Figure 4.2, where each link represents $P(A|X)$, the probability of the attribute A given a feature X, or $P(Y|A)$, the probability of the class label Y given an attribute A.

4.2 Semantic Attributes

As explained in Table 1.2, a semantic attribute is a mid-level representation that describes a high-level activity by human readable words. Broadly speaking, there are two types of



(a) Existing supervised-learning approaches.



(b) Proposed semantic attribute-based learning.

Figure 4.1: Comparison between the existing supervised learning approach to activity recognition and the proposed semantic attribute-based learning approach. See Table 1.2 for terminology and definitions.

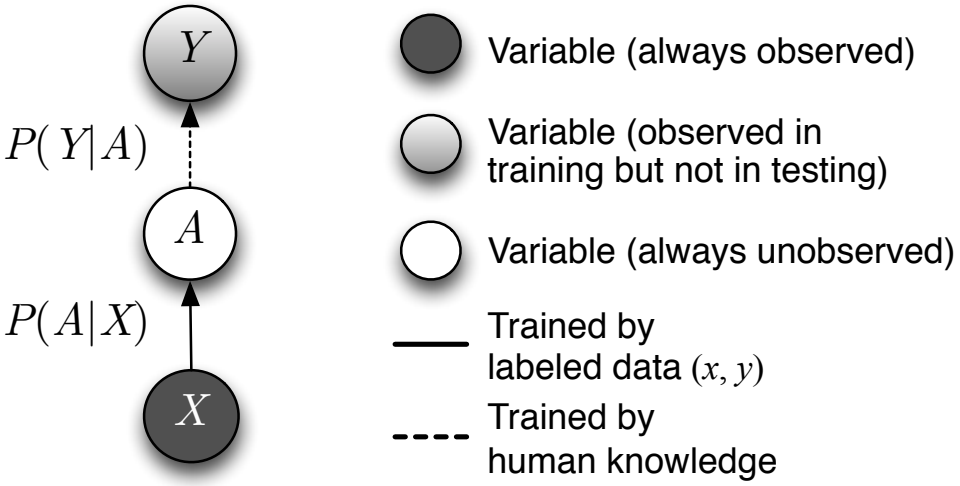


Figure 4.2: Graphical representation of semantic attribute-based activity recognition.

semantic attributes: *Compositional* and *non-compositional* attributes. The descriptions and examples of them are shown in Table 4.1. We will present the definition, construction, and detection of semantic attributes in the following sections.

4.2.1 Representation of Human Activities

There are various kinds of candidate representations of human activities, which are described as follows:

Body Model-Derived Primitives

One approach is to create a representation based on human anatomy and kinesiology (also known as human kinetics). That is, one can represent a physical activity using body model-derived primitives, including the models for torso, limbs, and joints [105]. The body model provides natural constraints on the relationship between attributes. For example, one cannot raise one's hand without lifting one's forearm. While it benefits from the domain knowledge of human kinetics, the limitation of this approach is that not all attributes are directly

Table 4.1: Categorization of semantic attributes: Compositional vs. non-compositional.

Attribute Type	**Description**	**Examples**
Compositional Attribute	An attribute that is an atomic action, a sub-activity, or a building block that can be combined to constitute a high-level composite activity. The composition can be a set or a sequence of attributes, as described in Section 4.2.1. This type of attributes is analogous to *verbs* in natural language grammars.	`ArmDown`, `ArmCurl`, `Running`, `Kicking`
Non-Compositional Attribute	An attribute that is a description or a characteristic of a high-level activity. Expressed in natural language, the attribute is a logical predicate, which can be true or false for an activity, but is not considered a sub-activity or a part that constitutes the activity. This type of attributes is analogous to *adjectives* in natural language grammars.	`IsCyclicMotion`, `IsTeamSport`, `IsMealRelated`, `IsWorkRelated`

associated with the motion of a part of the human body. For example, if we want to use some descriptive attributes such as "`IsWorkRelated`," "`IsMealRelated`," or "`IsTimeEvening`".

Bag of attributes

In this thesis, we propose the *bag-of-attributes* representation for human activities. The bag-of-attributes representation expresses a high-level activity in terms of a set of descriptive words in natural language. For example, we can represent a set of high-level sports activities {Soccer, Golf, Baseball} using a set of semantic attributes as follows:

$$\mathbf{a}_{\text{Soccer}} = [\texttt{Running} = 1, \texttt{Kicking} = 1, \texttt{Swinging} = 0, \texttt{IsTeamSport} = 1]$$

$$\mathbf{a}_{\text{Golf}} = [\texttt{Running} = 0, \texttt{Kicking} = 0, \texttt{Swinging} = 1, \texttt{IsTeamSport} = 0]$$

$$\mathbf{a}_{\text{Baseball}} = [\texttt{Running} = 1, \texttt{Kicking} = 0, \texttt{Swinging} = 1, \texttt{IsTeamSport} = 1]$$

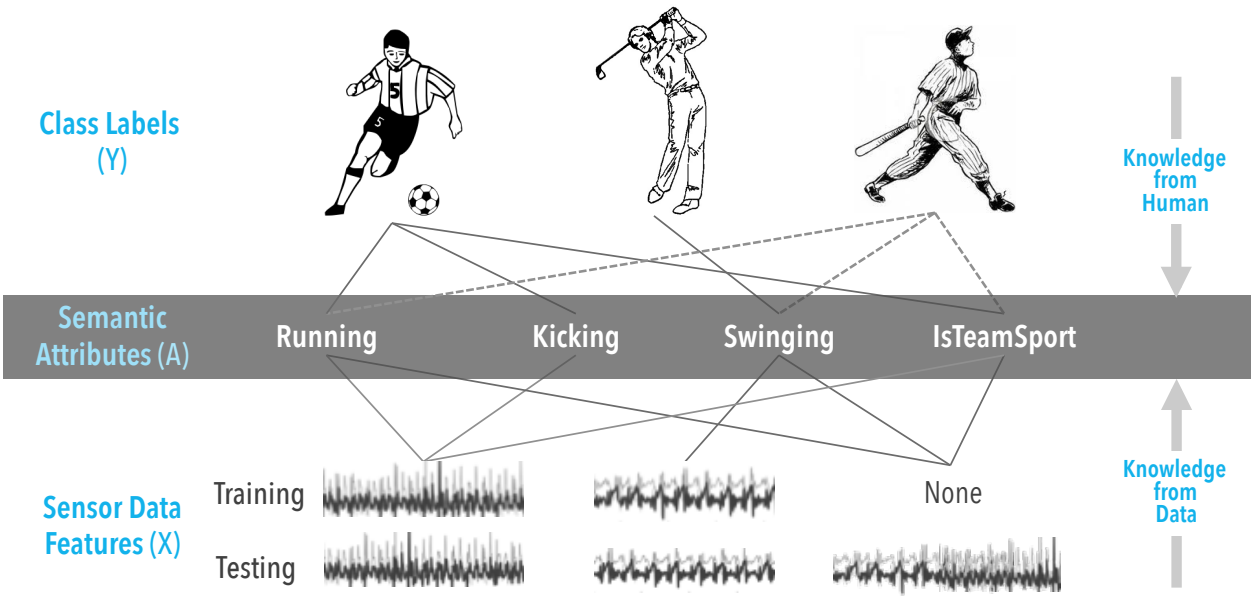


Figure 4.3: Illustration of using the bag-of-attributes representation for zero-shot activity recognition.

Figure 4.3 illustrates the idea of expressing the activities using bag of attributes in order to achieve zero-shot learning. When an attribute takes on a value of 1, it means that the attribute is related to or can be associated with that activity class. By the definition in this thesis or in the related work [42, 68], it does not necessarily mean that the attribute is always present. For example, a person playing baseball does not always exhibit the `Swinging` attribute, but `Swinging` is certainly a part of the baseball activity.

The bag-of-attributes representation also does not specify the order in which the attributes occur. For instance, we know that playing soccer involves `Running` and `Kicking`, but the order or the sequence of their occurrence is not specified for simplicity and generality.

As explained in Table 4.1, it is to be noted that an attribute used in the bag-of-attributes representation is not necessarily a compositional attribute, which is a sub-activity or a part of a high-level activity. Rather, it can be a non-compositional attribute, which is a description or a characteristic of a high-level activity. For example, the attribute `IsTeamSport`

is not a sub-activity that constitute the soccer or baseball activity, but it does describe a characteristic of those activities. Imagine the case where the players are equipped with some wearable, miniature radio transceivers that can detect each other in proximity. Then, this `IsTeamSport` attribute can then be used directly to boost the confidence in recognizing the correct sports activity.

Sequence of attributes

In cases where the ordering of actions is important, a possible approach is to use the *sequence-of-attributes* representation, which preserves the information about the order of the attributes. For example, an instance of the "Having Meal" activity [38] can be represented by a sequence a_k of K elements such as

$$(a_k)_{k=0}^{K-1} = (a_0, a_1, \ldots, a_{K-1}) = (\texttt{PickFood}, \texttt{HaveSoup}, \texttt{CutSteak}, \texttt{Drink})$$

More generally, taking into account the possibility of multiple concurrent actions or attributes, we can even use a directed acyclic graph (DAG) to represent an ordered relationship between the attributes in an activity. From this point of view, the sequence-of-attributes representation can be regarded as a degenerate case of the DAG-based representation. While being more descriptive, in many cases it is difficult or impractical to describe an activity in terms of a sequence of attributes. This is due to several reasons. Firstly, many attributes can occur simultaneously. Secondly, many non-compositional attributes (such as `IsWorkRelated`) are general descriptions or characteristics of an activity, rather than an atomic activity unit that can be put in order. Finally, the inherent variation of an activity can make it hard to define a specific sequence in which the attributes will be observed.

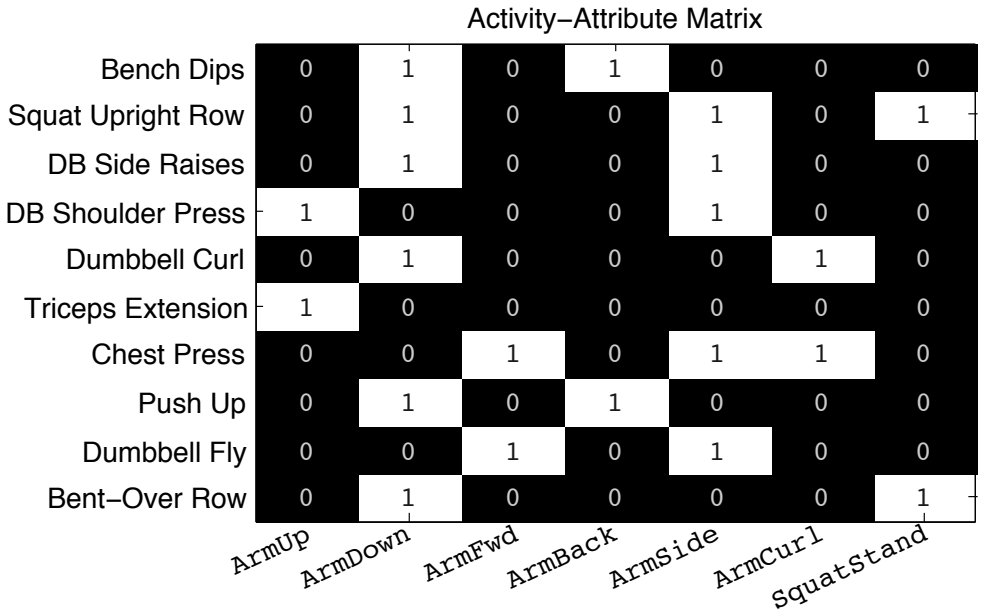


Figure 4.4: Activity-attribute matrix for exercise activities. The rows are the activities and the columns are the attributes.

4.2.2 Activity-Attribute Matrix

After taking into account the pros and cons of each representation, we define an *Activity-Attribute Matrix* over the bag of attributes. The Activity-Attribute Matrix encodes the human knowledge on the relationship between an activity and a set of semantic attributes that are associated with the activity. For M activities and N attributes, the activity-attribute matrix is an $M \times N$ matrix where the value of each element a_{ij} represents the level of association between activity i and attribute j. For simplicity, in this work we define each element as a binary value, indicating whether such an association exist ($a_{ij} = 1$) or not ($a_{ij} = 0$). It is straightforward to generalize a_{ij} to be real-valued ($0 \leq a_{ij} \leq 1$), indicating the level or confidence of the association. An example of a binary activity-attribute matrix we manually defined for the exercise activity domain in our experiments is shown in Figure 4.4. The matrix is defined based on the definitions and descriptions of the exercise activities in the literature [13, 35, 64].

4.2.3 Attribute Set Construction

Domain Knowledge

In general, an activity-attribute matrix can be manually defined by common-sense knowledge or domain knowledge [2]. The activity-attribute matrix used in our experiments is an extension of related work in attribute-based object similarity and classification [37, 42, 49, 97]. A user can also manually define a custom new activity by describing it using the attributes, which is equivalent to inserting a row in the matrix.

Text Mining and Deep Learning

Instead of human annotation, some recent work in zero-shot learning and computer vision has tried to automate the construction of attribute sets and the mappings between attributes and classes using text mining [21, 69, 82]. The motivation is that modern visual recognition systems are often limited in their ability to scale to a large number of object categories. Socher et al. [82] presented a model for zero-shot learning where a deep neural network was first trained in an unsupervised way from a large collection of images to obtain an image representation in the hidden layer. In parallel, a neural network language model was trained to obtain an embedding representation for thousands of common terms. That is, a class label can be represented in a d_w-dimensional word space, where the word vectors are learned from Wikipedia text in an unsupervised way using the deep neural network. The authors then trained a linear mapping $\boldsymbol{\theta}$ between the image representations and the word embeddings representing a few classes for which they had labeled images:

$$J(\boldsymbol{\theta}) = \sum_{y \in Y_s} \sum_{\mathbf{x}^{(i)} \in X_y} ||\mathbf{w}_y - \boldsymbol{\theta}\mathbf{x}^{(i)}||^2 \tag{4.1}$$

where Y_s is the set of seen classes and y is a certain class label. $\mathbf{x}^{(i)}$ is the image feature vector of the i-th training instance, and X_y is the set of all instances that belong to class y. $\mathbf{w}_y$ is the

vector representation of class y in the embedding "semantic attribute" space. By minimizing this objective function, $\boldsymbol{\theta}$ maps the image representation space onto the embedding semantic attribute space.

Crowd-Sourcing

An alternative approach is to leverage crowdsourcing platforms such as Amazon Mechanical Turk [68, 75] to acquire a mapping between classes and attributes. A set of high-level class labels are presented to the test subjects, who are instructed to associate certain semantic attributes with each class label. In [69], the system actively choose a set of image samples that are likely to be nameable based on a nameability model, and a user can either provide attribute names or an "unnameable" tag. Still, it is more challenging to apply the approach described in [69] to non-vision-based human activity recognition, because non-vision sensor data are harder to visualize in a way that users can comprehend.

4.3 Attribute Detection

Given an activity-attribute matrix, the next step is to train a set of attribute detectors so that we are able to infer the presence/absence of an attribute from the sensor data features. However, collecting a separate training dataset for every attribute is not practical for several reasons. First of all, not all of the attributes are sub-activities themselves. Many attributes are descriptions, characteristics, or consequences of an activity rather than standalone sub-activities. Therefore, it may not be possible to collect data for an attribute "alone" without other interference or confounding factors. Furthermore, there can be a large number of possible attributes. If there were a need to collect many separate training datasets, the benefit of attribute-based learning would diminish significantly.

Since the goal is only to infer if an attribute is present or not given the feature vector

(i.e. $P(A|X)$ in Figure 4.2), what we need is one set of positive samples and another set of negative samples. Therefore, to learn an attribute detector, we reuse the existing training data by merging the labeled data of all activity classes that are associated with the attribute as the positive set. Similarly, the negative set consists of the data of all activity classes that are not associated with the attribute.

After the training sets are constructed, a binary classifier is trained for each attribute. In general, any type of classifier can be used. We evaluated various classifiers and selected the Support Vector Machine (SVM) classifier [7] as the optimal implementation. The classifiers we evaluated are described in the following subsections. The comparison of their performances is presented in Chapter 7.

4.3.1 Support Vector Machine Classifier

SVM finds the hyperplane $\mathbf{w}^T\mathbf{x}_i + b = 0$ that maximizes the margin between the data points of different classes by optimizing the following Quadratic Programming problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & a_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \ \forall i \end{aligned} \tag{4.2}$$

where $\mathbf{x}_i$ and a_i are the feature vector and the attribute value for the i-th training sample, respectively. Parameters $\mathbf{w}$ and b control the orientation and the offset of the hyperplane. The parameter C is a regularization term which controls overfitting and the tolerance on the degree of false classification ξ_i for each sample. After the training phase, we have a trained attribute detector for each attribute specified in the activity-attribute matrix.

In some cases, we might only have positive or negative examples for an attribute. For example, this can happen when all of the seen classes in the training data exhibit a certain

attribute. In such cases, we train the attribute detector using one-class SVM [58], which classifies a sample as one of the two classes given only training data of one class (positive or negative).

4.3.2 k-Nearest Neighbor Classifier

In the k-nearest neighbor (k-NN) classifier, a new sample is classified as the majority class membership of the k closest training data points (ties can be broken at random). In our experiments, the L_2 distance (i.e. Euclidean distance) metric is used. The resulting decision boundaries are Voronoi cells, which are composed of hyperplanes that form perpendicular bisectors of pairs of points from different classes.

k-NN can be susceptible to noises or outliers because its decision is simply based on the neighbors, rather than looking at the big picture and finding a general decision boundary. Another limitation is that classification requires access to the whole training dataset. In cases where the training data is large or the resource (e.g. storage or transmission bandwidth) on the device is limited, k-NN may not be the ideal classifier to use.

4.3.3 Decision Tree Classifier

As the name suggests, a decision tree classifier [62] is a tree structure where each intermediate node (including the root) is a logical predicate L on the input X

$$L : X \longrightarrow \{0, 1\} \tag{4.3}$$

in which 0 means false and 1 means true, and each leaf node is associated with one class label. In other words, a decision tree classifier classifies an instance through a sequence of questions and yes-or-no decisions.

During the training process, the training instances are classified starting from the root by

picking one predicate at a time, and then recursively classified until all the samples at a leaf node belong to the same class. Since the hypothesis space is exponential in the number of features, a brute-force full search for the optimal tree is usually computationally intractable. Most decision tree training algorithms thus use a greedy heuristic to find the approximately optimal tree, which can be suboptimal in the entire hypothesis space. The greedy heuristic to find the optimal logical predicate for each node is to look at the information gain achieved after passing the training instances through the predicate. Specifically, the optimal logical predicate L^* for a node is found using the following equation:

$$L^* = \underset{L}{\operatorname{argmax}}\, I(Y, L(X)) = H(Y) - H(Y|L(X)) \tag{4.4}$$

where $L(X)$ is a logical predicate operating on the input X. $H(Y)$ and $H(Y|L(X))$ is the entropy of the class labels of the remaining training instances before and after being classified by L, respectively, where

$$H(Y) = \sum_{y \in \mathbb{Y}} \log\left(\frac{1}{P(Y=y)}\right) P(Y=y) \tag{4.5}$$

$$H(Y|L(X)) = \sum_{l \in \{0,1\}} P(L(X)=l) H(Y|L(X)=l) \tag{4.6}$$

One advantage of using a decision tree for human activity recognition is that the classification is computationally inexpensive, which is attractive to real-time recognition running on mobile or wearable devices. A potential drawback is due to the greedy nature in the training process, which may lead to suboptimal recognition accuracy or overfitting issues if tree pruning is not handled well [62].

4.3.4 Naive Bayes Classifier

The naive Bayes classifier [62], as shown in Figure 4.5, belongs to the family of Bayesian networks [86]. The most important assumption of the naive Bayes classifier is that the

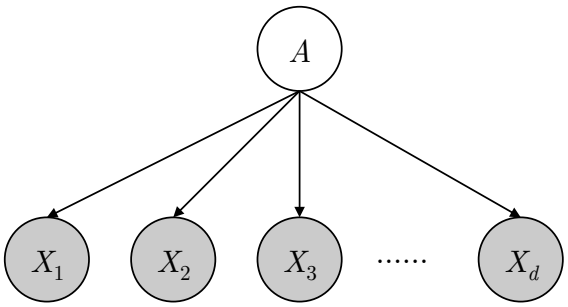


Figure 4.5: The graphical model representation of the Naive Bayes classifier.

features are conditionally independent given the class label [62], which significantly reduces the number of parameters to train in the model from exponential to linear in the number of features. Based on this assumption, the conditional probability of observing features $X_1, X_2, \ldots, X_d$ given the class random variable Y is formulated as:

$$P(X_1, X_2, \ldots, X_d|Y) = \prod_{k=1}^{d} P(X_k|Y) \tag{4.7}$$

The model parameters $P(X_k|Y)$ and $P(Y)$ can be trained through maximum likelihood estimation (MLE) or maximum a posteriori (MAP) estimation. After training, the classification is done by:

$$y^* = \underset{y}{\operatorname{argmax}}\, P(Y = y|X_1, X_2, \ldots, X_d) = \underset{y}{\operatorname{argmax}}\, \frac{P(Y = y)\prod_{k=1}^{d} P(X_k|Y)}{P(X_1, X_2, \ldots, X_d)} \tag{4.8}$$

For continuous-valued data, it is usually assumed that the generative distribution $P(X|Y)$ follows a Gaussian distribution. Therefore, the accuracy of Naive Bayes classifier can be sub-optimal when the assumptions of Gaussian-distributed and pair-wise independent features given a certain class do not hold for the input sensor data features.

4.4 Attribute-Based Activity Classification

After the attributes are detected, in the attribute space, a nearest-neighbor classifier is used to recognize the high-level activity given an attribute vector generated from attribute detectors. Specifically, the activity recognizer takes an attribute vector $\mathbf{a} = [a_1, a_2, \ldots, a_m]$ as input and returns the closest high-level activity y^* represented in the attribute space $\mathbb{A}^m$ according to the activity-attribute matrix. That is,

$$y^* = \underset{y}{\operatorname{argmax}}\, d(\mathbf{a}, \mathbf{a}_y) \tag{4.9}$$

where $\mathbf{a}_y$ is the attribute vector template for class y as specified in the activity-attribute matrix, and $d(\cdot, \cdot)$ is the distance between two vectors in the attribute space. In our experiments, Euclidean distance (i.e. L_2 distance) is used as the distance metric. More generally, the decision can be formulated as:

$$y^* = \underset{y}{\operatorname{argmax}}\, d(P(\mathbf{a}|\mathbf{x}), \mathbf{a}_y) \tag{4.10}$$

where $P(\mathbf{a}|\mathbf{x})$ is the posterior probability of the attributes being $\mathbf{a}$ given an observed feature vector $\mathbf{x}$. This can be thought as using a *soft* decision rather than a hard, binary decision on the attributes.

4.5 Hybrid Activity Recognition

While attributes are human readable and can be used to recognize previously unseen new classes, there is a certain amount of information in the low-level feature space that we do not want to discard. Transforming low-level features to mid-level attributes has the benefit for unseen class recognition, but there is an information loss associated with it, just as there is in the process of feature extraction from raw sensor data.

Inspired by this thought, our idea is to keep the advantages of both feature-based and

Algorithm 1 Hybrid Feature/Attribute-Based Activity Recognition Algorithm

Input: low-level feature vector x
Output: estimated activity class y
$isUnseenClass \leftarrow$ unseenClassDetection(x);
if $isUnseenClass = true$ **then**
    Keep only unseen classes in the attribute space;
    $a \leftarrow$ attributeDetector(x);
    $y \leftarrow$ attributeBasedActivityClassifier(a);
else
    Keep only seen classes in the feature space;
    $y \leftarrow$ featureBasedActivityClassifier(x);
end if
return y;

attribute-based activity recognition. Specifically, if we know that a sample belongs to a seen class where we had training data in the dataset, we can directly apply a feature-based classifier to directly recognize the activity. On the other hand, if we think that a sample belongs to a new class that we have not had any training data associated with, we have to apply attribute-based activity recognition so that we can learn by reusing the known attributes.

Now the question is: How do we know if a sample belongs to a seen class or an unseen class? We draw an analogy between this problem and the problem of anomaly detection. A sample from a seen class is like a typical sample, which is similar to the other samples we had in the training data. In comparison, a sample from an unseen class is like an "anomaly" because it does not look like anything that the system has seen before. To approach this problem, we first train an unseen class detector using the one-class SVM classifier [12], where only the positive samples (all samples that belong to the seen classes) are given to the classifier. After using the unseen class detector, we then do a hybrid feature/attribute-based activity recognition using the algorithm described in Algorithm 1.

4.6 Discussion

In the current version of our semantic attribute-based learning approach, it is assumed that there exists a one-to-one mapping between an activity class label and a point in the attribute space, and that the associations between activities and attributes are fixed. This implies a fundamental limitation that the lower bound on the minimum number of attributes is:

$$n_A \geq \log_2 n_Y \tag{4.11}$$

for n_Y activity classes and n_A different attributes, assuming binary-valued attributes are used. From the equation, we can see that in the best case one does not need many attributes to cover a large number of classes. For example, ten attributes can cover up to $2^{10} = 1024$ classes in the best case where no two classes are identical in the attribute space. On the other hand, in the worst case where many attributes are highly correlated or the classes are different in a dimension that is orthogonal to the space spanned by the attribute vectors, the number of classes covered can be much lower than 2^{n_A}. One way to prevent these worst cases from happening is to select attributes based on their discriminability and detectability, as discussed in Section 7.4.4. Another possible research direction is to relax the limitation by incorporating continuous-valued attributes, relative attributes [70], or sequence of attributes. Furthermore, while we present an initial attempt to evaluate the attribute-based learning approach on two datasets, it would be beneficial to expand the study to more activity domains with a larger number of activities, attributes, and users in the future.

In the current implementation, the attributes are manually defined using common-sense knowledge and domain knowledge as an initial attempt towards zero-shot learning for activity recognition. To further reduce the effort of one-time manual definition per class, a potential future direction and part of our ongoing work is to automate the process using web text mining [69] or crowdsourcing [75] as explored in the zero-shot learning literature.

As we will see in Chapter 7, our results suggest that the performance of the zero-shot learning model varies depending on the selected semantic attributes. Therefore, another future direction is to develop a systematic way for semantic attribute selection based on the discriminability and detectability of the attributes. Further, to truly exploit the advantages of both low-level features and mid-level attributes, future work and experiments are to be done to explore and compare various kinds of algorithms for hybrid feature/attribute-based activity recognition.

Chapter 5

Sequential Structure: The Semantic Attribute Sequence Model

In the previous chapter, semantic attribute-based activity recognition is presented to model the hierarchical structure of human activities and to reuse semantic attributes as a bridge to recognize unseen new activity classes. However, so far we have been classifying each window of sensor data independently and have not taken into account the temporal dependency between neighboring windows or the temporal arrangement of attributes.

Human activities, similar to most natural phenomena and animal behaviors, are continuous in nature [38]. As a result, a sensor data stream, captured by the sensor-enabled devices that a person is carrying or encountering, is a time-series with strong temporal dependency. It has been shown that probabilistic graphical models are useful for modeling various kinds of time-series data [40, 86]. In this chapter, a probabilistic graphical model—the *semantic attribute sequence* model—and its application to recognizing human activities are presented.

5.1 Problem Definition

In this chapter, we expand the problem definition in Section 4.1. Consider a set of target human activity classes, $\mathbb{Y}$, that we aim to recognize. $\mathbb{Y} = \mathbb{Y}_S \cup \mathbb{Y}_U = \{\{y_1, y_2, \ldots, y_s\}, \{y_{s+1}, \ldots, y_{s+u}\}\}$. $\mathbb{Y}_S$ is the set of *seen activity classes*, where there exists some training data for every class. $\mathbb{Y}_U$ is the *unseen activity classes* set where there are no training data for any class. The problem is: How to train a model to recognize an unseen activity class $y \in \mathbb{Y}_U$ given a set of N training instances $\{(\boldsymbol{X}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$, each including a sequence of feature vectors $\boldsymbol{X}^{(i)} = \{\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_T^{(i)}\}$ and the corresponding ground truth class labels $\mathbf{y}^{(i)} = \{y_1^{(i)}, \ldots, y_T^{(i)}\}$, where $y_t^{(i)} \in \mathbb{Y}_S \; \forall \; t, i$?

5.2 Probabilistic Graphical Models

Two types of probabilistic graphical models are considered for the problem because they have shown promise in activity recognition [11, 57] and other related fields where the input time series exhibit a strong temporal dependency, such as speech recognition and natural language processing [86]. The first type is generative directed model, or often referred to as the dynamic Bayesian network (DBN). A generative model describes the underlying process of how the observable data are stochastically generated, typically given some parameters. For any classification or regression problem, fitting a model to data is essentially approximating a function $f : \mathcal{X} \longrightarrow \mathcal{Y}$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output (target) space. The way that generative models approach this is to look at how likely an observed value x of the input random variable X is generated if the value of the target random variable Y were y. In terms of probability distribution, generative models generally estimate the probability $P(Y|X)$ using Bayes' rule:

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)} \tag{5.1}$$

where $P(X|Y)$ is the probability distribution of "generating" X given Y, and $P(Y)$ is the prior distribution of the target random variable.

The second type is the conditional random field (CRF), which is a family of undirected, discriminative graphical model. Compared to generative models, discriminative models make weaker assumptions. Discriminative models directly approximate the function $f : \mathcal{X} \longrightarrow \mathcal{Y}$. Probabilistic discriminative models are usually directly in the form of $P(Y|X)$, which is the posterior probability distribution of the target random variable Y given the input random variable X.

We will go through the definition, analysis, and comparison of these two types of models, and how they relate to the expanded problem definition in Section 5.1.

5.2.1 Dynamic Bayesian Networks

Let G be a directed graphical model over n random variables $\{X_1, X_2, \ldots, X_n\}$. G is defined as a *Bayesian Network* (BN) if the probability distribution over G can be factorized as:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|\pi(X_i)) \tag{5.2}$$

where $\pi(X_i)$ is the set of *parents* of X_i (i.e. those nodes pointing directly to the X_i node via a single directed edge). The equation is also referred to as the *chain rule* for Bayesian networks [40].

A *Dynamic Bayesian Network* (DBN) is a Bayesian Network in which the random variables are related to each other over adjacent time steps [65]. One simple and widely-used kind of dynamic Bayesian network is the hidden Markov model (HMM) shown in Figure 5.1, which can be viewed as the sequential version of the naive Bayes model [86]. Another way to view it is that dynamic Bayesian networks generalize hidden Markov models by allowing the state space to be represented in factored form, instead of as a single random variable [65]. Hidden Markov models are suitable for problems where there is a sequence of hidden states

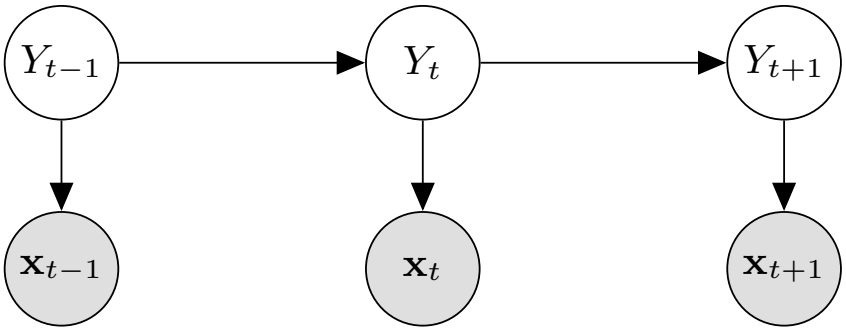


Figure 5.1: A hidden Markov model.

(or class labels) exhibiting a temporal dependency, and generating a set of observations at each time step. As shown in Figure 5.1, Y_t is the hidden state at time t, and $\mathbf{x}_t$ is the observed feature vector at time t.

Previous work has applied hidden Markov models to human activity recognition, where Y corresponds the activity class label and $\mathbf{x}$ corresponds to the feature vector extracted from sensor data. However, as explained in Chapter 4, this direct mapping between Y and $\mathbf{x}$ does not generalize well to unseen new classes in a zero-shot learning setting. Therefore, one intuitive generalization is to introduce the random variable A representing the mid-level semantic attributes. The resulting dynamic Bayesian network is shown in Figure 5.2, which can be viewed as a sequential and generative version of the model discussed in Chapter 4.

5.2.2 Relation to The n-gram Model

A sequence of semantic attributes in human activities is similar to a sentence of words in natural languages. In natural language processing, a statistical language model assigns a probability to a sequence of m words $P(W_1, \ldots, W_m)$ using a probability distribution. One of the most widely used language models is the n-gram model. In an n-gram model, the

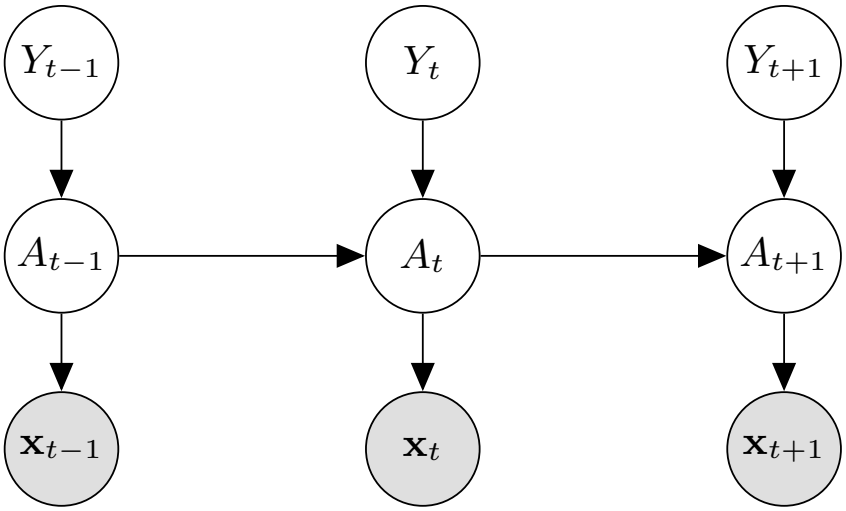


Figure 5.2: Generative probabilistic graphical model: dynamic Bayesian network.

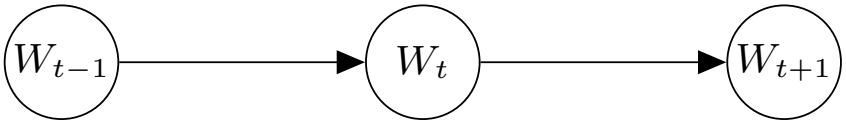


Figure 5.3: Graphical model representation of a bi-gram model.

probability $P(W_1, \ldots, W_m)$ of observing a sentence $W_1, \ldots, W_m$ is modeled as:

$$P(W_1, \ldots, W_m) = \prod_{i=1}^{m} P(W_i | W_1, \ldots, W_{i-1}) \simeq \prod_{i=1}^{m} P(W_i | W_{i-n+1}, \ldots, W_{i-1}) \tag{5.3}$$

That is, it is assumed that the probability of observing the i-th word W_i in the context history of the preceding $i-1$ words can be approximated by the probability of observing it in the shortened context history of the preceding $n-1$ words (n-th order Markov property). For example, in a *bi-gram model*, the context is the immediately preceding word. That is:

$$P(W_1, \ldots, W_m) = \prod_{i=1}^{m} P(W_i | W_{i-1}) \tag{5.4}$$

Comparing Figure 5.1 to 5.3 and Equation 5.2 to Equation 5.4, we can see that a bi-gram model can be viewed as a special case of hidden Markov model. In other words, using a

dynamic Bayesian network with first-order temporal links can be viewed as a generalization of the bi-gram model.

5.2.3 Conditional Random Fields

Another way to model the relationship between the sequence of activities, attributes, and sensor data features is through a family of undirected probabilistic graphical models—conditional random fields. By assuming a different form of probability distribution over an undirected graph, conditional random fields focus on the conditional, discriminative probability distribution of the hidden states or class variables, rather than the generative distribution of the observed variables. We first go through the background and definition of the conditional random field, and then in the next few sections present how it can be adapted and applied to the problem of human activity recognition.

A distribution P_Φ over a undirected graph G of n random variables $\{X_1, X_2, \ldots, X_n\}$ is a *Gibbs distribution* parameterized by a set of factors $\Phi = \{\phi_1(\boldsymbol{D}_1), \phi_2(\boldsymbol{D}_2), \ldots, \phi_k(\boldsymbol{D}_k)\}$ if it is defined as follows:

$$P_\Phi(X_1, X_2, \ldots, X_n) = \frac{1}{Z(\boldsymbol{X})} \tilde{P}_\Phi(X_1, X_2, \ldots, X_n) \tag{5.5}$$

where

$$\tilde{P}_\Phi(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{k} \phi_k(\boldsymbol{D}_k) \tag{5.6}$$

$$Z(\boldsymbol{X}) = \sum_{X_1, \ldots, X_n} \tilde{P}_\Phi(X_1, X_2, \ldots, X_n) \tag{5.7}$$

$\tilde{P}_\Phi(X_1, X_2, \ldots, X_n)$ is an unnormalized measure of the probability, and $Z(\boldsymbol{X})$ is a normalizing constant called the *partition function*, which ensures a normalized probability value between 0 and 1. Each $\boldsymbol{D}_i$ is a set of random variables in G. If every $\boldsymbol{D}_i$ is a clique (i.e. a complete subgraph) of G, then the graph is called a *Markov network* or a *Markov random field* (MRF).

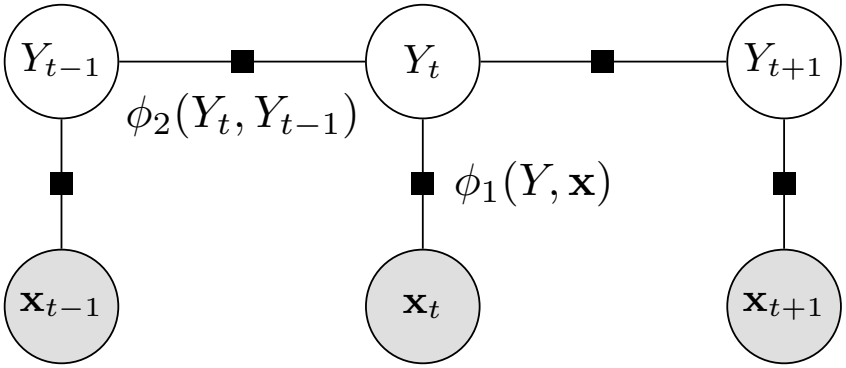


Figure 5.4: A linear-chain conditional random field model.

Closely related to Markov random fields, a *conditional random field* is an undirected graph G whose nodes consist of two sets of nodes, $\boldsymbol{X} \cup \boldsymbol{Y}$. The graph is associated with a set of factors $\Phi = \{\phi_1(\boldsymbol{D}_1), \phi_2(\boldsymbol{D}_2), \ldots, \phi_k(\boldsymbol{D}_k)\}$ such that $\boldsymbol{D}_i \not\subseteq \boldsymbol{X}$ for all i. The conditional distribution defined over the graph is:

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \frac{1}{Z(\boldsymbol{X})}\tilde{P}(\boldsymbol{Y}, \boldsymbol{X}) \tag{5.8}$$

where

$$\tilde{P}(\boldsymbol{Y}, \boldsymbol{X}) = \prod_{i=1}^{k} \phi_k(\boldsymbol{D}_k) \tag{5.9}$$

$$Z(\boldsymbol{X}) = \sum_{\boldsymbol{Y}} \tilde{P}(\boldsymbol{Y}, \boldsymbol{X}) \tag{5.10}$$

One representative example, the linear-chain conditional random field, is shown in Figure 5.4. It can be observed from the form that it is closely related to the hidden Markov model, with a different probability distribution defined over the graph. The difference between them will be further discussed in the next section.

5.3 Generative vs. Discriminative Models

We now discuss the design considerations of choosing between generative and discriminative models. Generative models typically impose a stronger assumption on the data generation process. For example, in most generative models it is assumed that a continuous-valued random variable (e.g., the value of a sensor reading) follows a Gaussian distribution. This assumption comes from the *central limit theorem* in probability theory: Let $X_1, X_2, \ldots, X_n$ be a sequence of i.i.d. random variables (of arbitrary distribution), each with mean μ and variance σ^2, then the distribution of the standardized sum of X_i's

$$Z_n = \frac{\sum_{i=1}^{n} X_i - n\mu}{\sigma\sqrt{n}}$$

converges to a standard normal distribution $N(0, 1)$. That is,

$$\lim_{n\to\infty} P\left(\frac{\sum_{i=1}^{n} X_i - n\mu}{\sigma\sqrt{n}} \leq x\right) = \lim_{n\to\infty} P(Z_n \leq x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-y^2}{2}} dy \tag{5.11}$$

or equivalently,

$$\frac{\overline{X} - \mu}{\sigma/\sqrt{n}} \longrightarrow N(0, 1) \tag{5.12}$$

However, there exist many features that do not follow the assumption of Gaussian distribution, or even any kind of parametric distribution. In this case, even with perfect parameter estimation with infinite training data, there is no way to avoid the *model error*, which stems from the fact that the true model (which generates the data) is not included in the model family we assume.

In comparison, in discriminative models we do not attempt to explain how the observed data are generated or how likely they occur. We simply try to make a prediction or decision based on what we have already observed. One of the major advantages of choosing discriminative models is that they are better suited to including rich, overlapping features [86].

Table 5.1: List of notations used in the semantic attribute sequence model.

Symbol	Description
Y_t	High-level activity class label at time t
A_t	Mid-level semantic attribute at time t
$\mathbf{x}_t$	Low-level D-dimensional feature vector at time t. $\mathbf{x}_t = \{X_{1,t}, X_{2,t}, \ldots, X_{D,t}\}$
$\phi_k(\mathbb{V})$	The k-th potential function in a probabilistic graphical model involving a set of vertices $\mathbb{V}$

When the size of the training data is sufficiently representative of the whole input space, discriminative models usually outperform generative counterparts. On the other hand, when there are few training data available, generative models often lead to better results because they make stronger assumptions and are usually less susceptible to the bias of the training data.

5.4 The Semantic Attribute Sequence Model

Based on the analysis in the previous sections, we propose a probabilistic graphical model—the *Semantic Attribute Sequence* model—to model the sequential structure of human activities and to recognize seen or unseen activities through a layer of semantic attributes [14]. The model is shown in Figure 5.5, and the notations are listed in Table 5.1.

The model is a variation of conditional random field (CRF) [7], which is suitable for activity recognition because it models the temporal dependency in sequential data [38]. It also supports the use of complex features, whose distributions and dependencies may not have a simple parametric form, by imposing weaker assumptions on the dependencies between features compared to hidden Markov models [7, 8]. Given a sequence of observed features $\boldsymbol{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$, the conditional probability distribution of the activity sequence

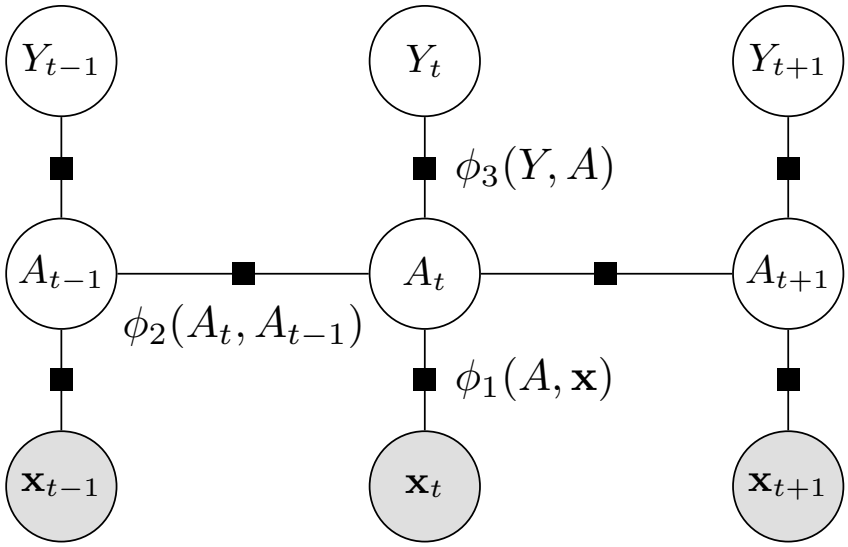


Figure 5.5: Discriminative probabilistic graphical model of a sequence of high-level human activities, mid-level semantic attributes, and observed low-level signal features.

$\boldsymbol{Y} = \{Y_1, \ldots, Y_T\}$, the attribute sequence $\boldsymbol{A} = \{A_1, \ldots, A_T\}$ given $\boldsymbol{X}$ is modeled as:

$$P(\boldsymbol{Y}, \boldsymbol{A}|\boldsymbol{X}) = \frac{1}{Z(\boldsymbol{X})} \prod_{t=1}^{T} \prod_{k=1}^{K} \phi_{k,t}(Y_t, A_t, A_{t-1}, \mathbf{x}_t) \tag{5.13}$$

where

$$Z(\boldsymbol{X}) = \sum_{\boldsymbol{Y}} \sum_{\boldsymbol{A}} \prod_{t=1}^{T} \prod_{k=1}^{K} \phi_{k,t}(Y_t, A_t, A_{t-1}, \mathbf{x}_t) \tag{5.14}$$

is a normalization term that ensures the probability distribution sums up to one. Each $\phi_{k,t} = \exp\left(w_m f_m(Y_t, A_t, A_{t-1}, \mathbf{x}_t)\right)$ is a potential function that consists of a model parameter w_m and a feature function f_m defined over a subset of the random variables Y, A, and $\mathbf{x}$. Our probabilistic graphical model consists of three types of potential functions (thus $K = 3$ in our current design):

- $\phi_{1,t}$ models the probability distribution of an attribute A_t given a feature vector $\mathbf{x}_t$:

$$\phi_{1,t}(A_t, \mathbf{x}_t) = \exp\left(\sum_{a \in \mathbb{A}} \sum_{d=1}^{D} w_{a,d} x_{d,t} \cdot I(A_t = a)\right) \tag{5.15}$$

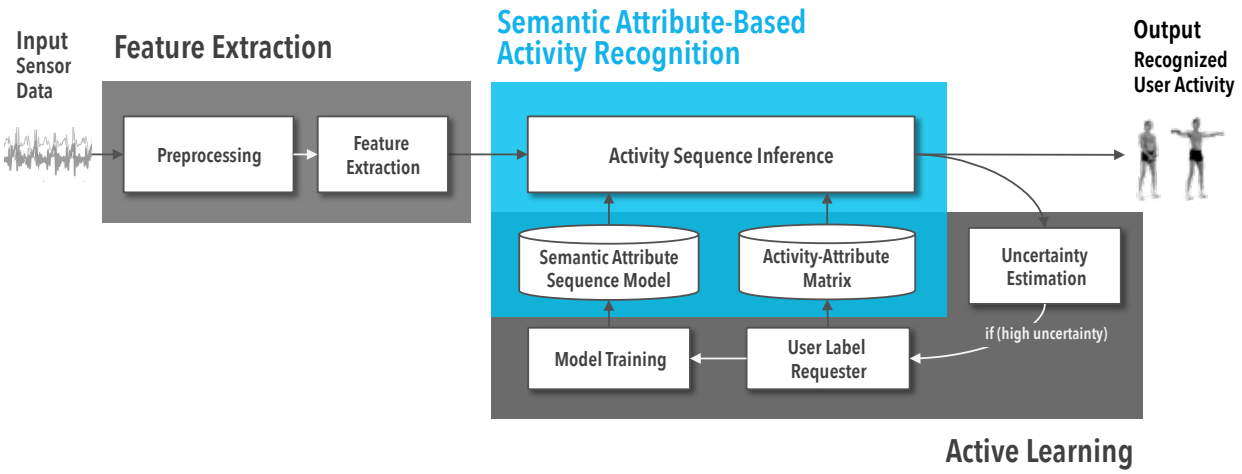


Figure 5.6: The NuActiv activity recognition framework with the semantic attribute sequence model.

where $I(p)$ is the indicator function that takes the value 1 if the statement p is true, and takes the value 0 otherwise.

- $\phi_{2,t}$ models the temporal dependency between neighboring semantic attribute values, which can be regarded as a smoothness term between neighboring nodes:

$$\phi_{2,t}(A_t, A_{t-1}) = \exp\left(\sum_{a\in\mathbb{A}}\sum_{a'\in\mathbb{A}} w_{a,a'} I(A_t = a) I(A_{t-1} = a')\right) \tag{5.16}$$

- $\phi_{3,t}$ models the correlation between the activity class Y and the semantic attribute A:

$$\phi_{3,t}(Y_t, A_t) = \exp\left(\sum_{y\in\mathbb{Y}}\sum_{a\in\mathbb{A}} w_{y,a} I(Y_t = y) I(A_t = a)\right) \tag{5.17}$$

Figure 5.6 shows how the semantic attribute sequence model fits in the NuActiv activity recognition framework. As an extension of the approach presented in Chapter 4, the semantic attribute sequence model is an alternative to the two-layer attribute-based activity recognition. It can be used when temporal dependency is important to the activities, without changing the other parts (the feature extraction part and the active learning part) of the NuActiv framework.

5.5 Parameter Estimation

Given the probability distribution defined in Equation 5.13, the likelihood of a training dataset with N instances is:

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{N} P(\boldsymbol{Y}^{(i)}, \boldsymbol{A}^{(i)} | \boldsymbol{X}^{(i)}) \\ &= \prod_{i=1}^{N} \frac{1}{Z(\boldsymbol{X}^{(i)})} \prod_{t=1}^{T} \prod_{k=1}^{K} \phi_{k,t}(Y_t^{(i)}, A_t^{(i)}, A_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) \end{aligned} \tag{5.18}$$

where $\theta = \{w_m\}$ denotes the set of all model parameters to be estimated. During the offline training phase, the optimal model parameters $\theta^* = \{w_m^*\}$ are learned by maximum likelihood estimation (MLE), which maximizes the log-likelihood of the training data $L(\theta)$:

$$\log L(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{m=1}^{M} w_m f_m(Y_t^{(i)}, A_t^{(i)}, A_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) - \sum_{i=1}^{N} \log Z(\boldsymbol{X}^{(i)}) \tag{5.19}$$

where $Y_t^{(i)}, A_t^{(i)}, \mathbf{x}_t^{(i)}$ are the class label, attributes, and features of the i-th instance in the training data, respectively.

5.6 Regularization

While the maximum likelihood estimation can fit the model to the training data, it can lead to the problem of overfitting. Overfitting occurs when a statistical model describes random error or noise instead of the underlying relationship. Overfitting generally occurs when a model is excessively complex, such as having too many parameters relative to the number of observations. For example, there might be a lot of learned weights w_m's that have large values, which fit the training data well but generalize to future testing data poorly. Observing a decrease in the training error and an increase in the testing error at the same time is an indicator of overfitting. In other words, overfitting occurs when a model begins to memorize training data rather than learning to generalize from trend.

To avoid the overfitting problem, regularization is evaluated and incorporated into the training process. Specifically, we maximize the regularized log-likelihood of the training data:

$$\begin{aligned}\log L_r(\theta) = &\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{m=1}^{M} w_m f_m(Y_t^{(i)}, A_t^{(i)}, A_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) \\ &- \sum_{i=1}^{N} \log Z(\boldsymbol{X}^{(i)}) - \lambda \sum_{m=1}^{M} w_m^2 \end{aligned} \tag{5.20}$$

where the last term is the L^2-regularization term that penalizes large w_m values with a weighting λ to prevent overfitting [7]. Adding the L^2-regularization term in the log-likelihood space is equivalent to multiplying the data likelihood by a Gaussian prior. In other words, instead of maximum likelihood estimation, we use maximum a posteriori (MAP) estimation where the prior probability of the parameters are Gaussian-distributed. In the experiments, λ is empirically set to 50 based on a cross-validation test. The optimization problem $\theta^* = \text{argmax}_\theta\, L_r(\theta)$ is solved using L-BFGS [7], a widely used optimization algorithm.

5.7 Inference

The goal of activity recognition is to infer the sequence of activities $\boldsymbol{Y}$ (may be seen or unseen) given a sequence of observed features $\boldsymbol{X}$, through a layer of attribute sequence $\boldsymbol{A}$. During the online testing phase, the states of the sequence $\boldsymbol{Y}$, $\boldsymbol{A}$ with maximum likelihood are inferred using the Junction Tree algorithm [40]. The Junction Tree algorithm is a message passing algorithm for performing inference by belief propagation in a graphical model. Since all the target classes in $\mathbb{Y}$ are described using the semantic attributes in the attribute space $\mathbb{A}$ in the form of an Activity-Attribute Matrix, we are able to infer the value y_t^* that Y_t takes on even if y_t^* corresponds to an previously unseen new activity, i.e. $y_t^* \in \mathbb{Y}_U$.

5.8 Discussion

The proposed semantic attribute sequence model is only one of the possible models in the design space. There are many alternatives using either generative or discriminative graphical models. For example, in our current graphical model, the nodes are only connected to their immediate neighbors. As a result, long-range dependencies between nodes are not directly taken into account in this model. In cases where long-range dependencies are important, additional links can be added between distant class nodes, attribute nodes, or feature nodes. For example, the skip-chain conditional random field [23, 86], which has shown promise in natural language processing, can be applied to activity recognition as well.

In addition to adding temporal links, another research direction is to expand the graphical model vertically. In other words, the model is not limited to the three-layer structure. Additional layers could be added to represent composite attributes or higher-level complex activities. As in most machine learning problems, the challenge of having a more complex model is to balance the tradeoff between model complexity and generality, and to avoid overfitting the training dataset. Further research effort is also required to find out the best way to acquire human knowledge about the compositionality and relationship of multiple layers of complex activities and attributes.

Chapter 6

Human in The Loop: Active Learning for Activity Recognition

So far we have focused on the scenario where no training data for the target class are available. What if we have the opportunity to acquire some ground-truth labeled data from users? Obviously, if we ask users to label every single sample, we can achieve the best recognition accuracy possible. However, it is impractical to ask a user to label his/her activity every single minute because it would be extremely intrusive. The more frequently we prompt the users for inputs, the more intrusive the system is [81]. This observation motivates us to design a user feedback loop for the NuActiv system using active learning algorithms [81].

Our idea is simple: We ask a user for labels only when we are highly uncertain about our recognition result. The intuition is that asking the user to confirm an uncertain recognition result can help improve the activity recognition system the most. To achieve this, the ideas and techniques of uncertainty sampling in the field of active learning are used.

6.1 Problem Definition

The problem of active learning for activity recognition in this work can be formalized as follows. Let $\mathcal{U}$ be a set of unlabeled instances, $\mathcal{U} = \{x^{(u)}\}_{u=1}^{U}$, and $\mathcal{L}$ be a set of labeled instances, $\mathcal{L} = \{(x^{(l)}, y^{(l)})\}_{l=1}^{L}$. Let $\mathbb{X}$ and $\mathbb{Y}$ be the feature space and the activity class space, respectively. As mentioned in Chapter 4, an activity recognition system is essentially a classifier function $f : \mathbb{X} \rightarrow \mathbb{Y}$. Let $f^*(\mathcal{L})$ be the optimal classifier trained on $\mathcal{L}$. Suppose there is a family of *select-and-label* functions h, which operate on a set of unlabeled instances and return an instance selected from the set and its label. That is, $(x^{(v)}, y^{(v)}) = h(\mathcal{U})$. In the real-world scenario, h is a process where the system selects an unlabeled instance and acquires its label from a source, such as a human annotator. Furthermore, let $m(f, \mathcal{T})$ be a certain performance metric function which returns a real value given a classifier f and a test set $\mathcal{T}$, where a larger value of m indicates a better performance. For example, m can be accuracy, precision, recall, or some other measure. Given these definitions, the problem of active learning is to find an optimal select-and-label function h^* such that:

$$h^* = \underset{h}{\operatorname{argmax}} \; m\left(f^*(\mathcal{L} \cup \{h(\mathcal{U})\}), \mathcal{T}\right) \tag{6.1}$$

That is, active learning aims to find an optimal way to select an instance from a set of unlabeled instances and obtain its label, such that the performance of the classifier (trained on the labeled instance set plus the newly labeled instance) is maximized. In the following sections, we will give an overview of active learning and the details of the approach used in this work.

6.2 Overview of Active Learning

The idea of active learning algorithms is that a machine learning algorithm can perform better when presented with less training data if it is allowed to choose the data from which

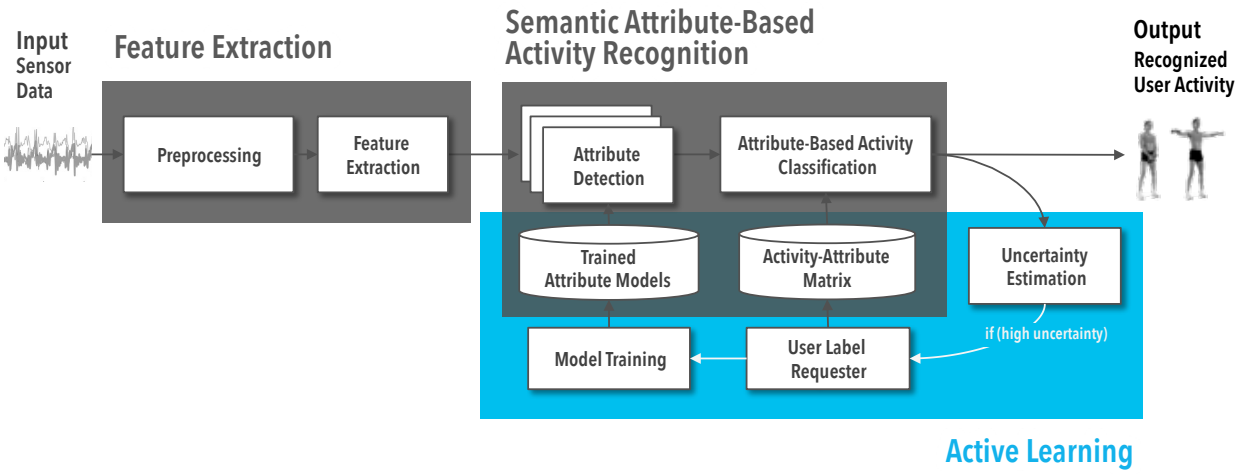


Figure 6.1: The active learning part in the NuActiv activity recognition framework.

it learns [80]. This is different from traditional "passive" learning systems, which generally induce a hypothesis to explain whatever training data happens to be available (e.g., a collection of labeled instances) [81].

One common assumption of active learning is that obtaining an instance is significantly less expensive than obtaining its label. This is true in many real-world applications. For example, downloading a large collection of images from the internet is relatively low-cost compared with asking humans to assign class labels to them.

The goal of active learning is to "get the most out of the least." In other words, the goal is to achieve a certain amount of targeted improvement on the prediction accuracy of a learner using as few new labeling actions as possible. With this goal in mind, we integrate active learning into the NuActiv framework, as highlighted in Figure 6.1. We will describe the approach in more detail in the following sections.

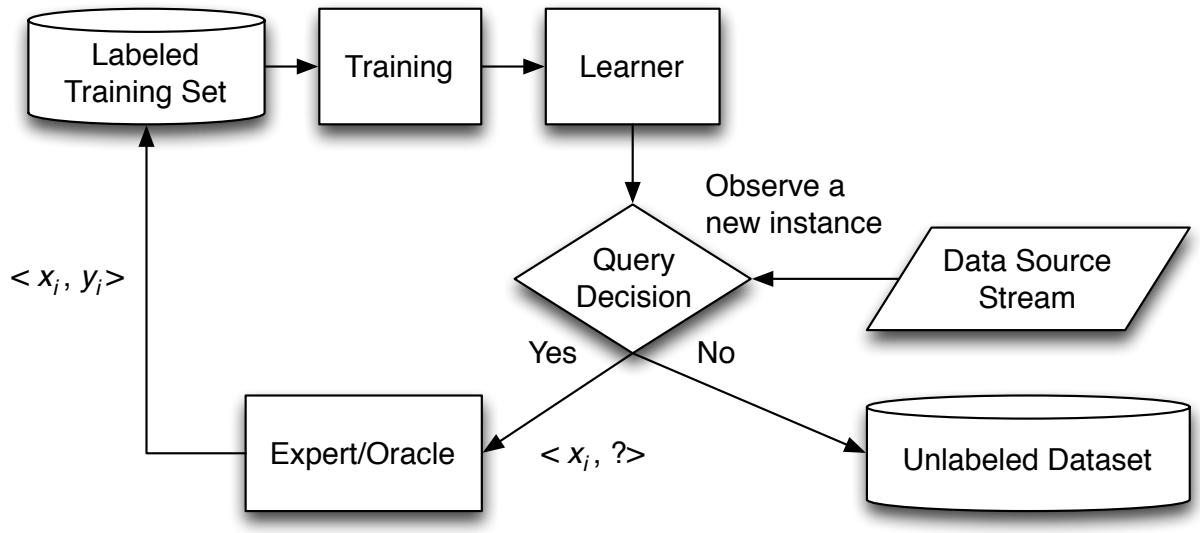


Figure 6.2: Illustration of stream-based active learning.

6.3 Sampling Scheme

There are two types of selective sampling schemes. The first one is *Stream-Based Sampling*(Figure 6.2), where unlabeled instances are typically drawn one at a time from the input source, and the system must decide whether to query or discard each instance. The second scheme is *Pool-Based Sampling* (Figure 6.3), where a large pool of unlabeled data is available. Having observed all the unlabeled instances, the system can ask for the label of one instance at a time according to certain decision criteria.

Considering the pros and cons of the two schemes, I propose a hybrid stream/pool based sampling scheme that is more suitable for the scenario of human activity recognition using mobile phones or wearable devices. The pool is not so big so that a user forgets what he/she did during the time interval asked by the system, yet large enough for the system to select a good sample to ask the user for a ground-truth label. The detailed settings of the experiments are described in Chapter 7.

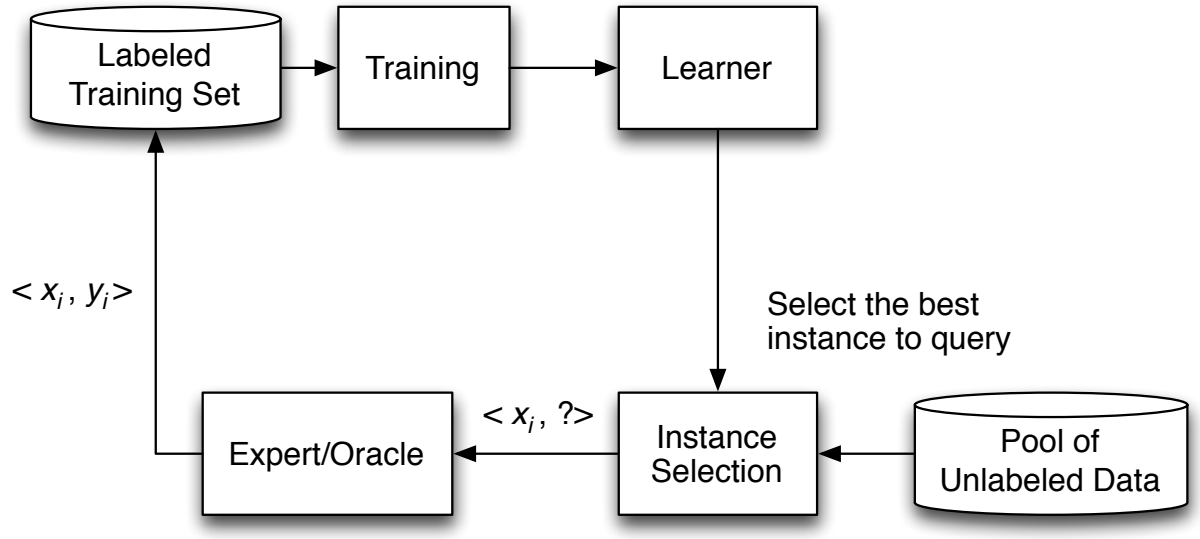


Figure 6.3: Illustration of pool-based active learning.

6.4 Uncertainty Sampling Metrics

In this work, several different metrics are used to measure the uncertainty of a sample x in the unlabeled sample pool $\mathcal{U}$ to the classifier.

6.4.1 Least Confident

Suppose the classifier output is $\hat{y}$, which satisfies the condition that

$$P_\theta(Y = \hat{y}|x) \geq P_\theta(Y = y'|x),\ \forall y' \neq \hat{y} \tag{6.2}$$

Under the least confident sampling metric, the learner will initiate a request for a ground-truth label when the confidence score of the classifier output $\hat{y}$ given the input feature x of a sample is minimum. That is,

$$x^*_{LC} = \underset{x \in \mathcal{U}}{\operatorname{argmin}}\, P_\theta(\hat{y}|x) \tag{6.3}$$

6.4.2 Minimum Margin

The learner initiates a request for a ground-truth label when the difference between the confidence of the first and the second likely classes ($\hat{y}_1$ and $\hat{y}_2$, where where $P_\theta(Y = \hat{y}_1|x) \geq P_\theta(Y = \hat{y}_2|x) \geq P_\theta(Y = y'|x),\ \forall y' \notin \{\hat{y}_1, \hat{y}_2\}$) is small:

$$x_M^* = \underset{x \in \mathcal{U}}{\operatorname{argmin}} \left[P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x) \right] \tag{6.4}$$

6.4.3 Maximum Entropy

Entropy, in information theory, is measure of the uncertainty associated with a random variable. $H_\theta(Y|x) = -\sum_y P_\theta(y|x) \log P_\theta(y|x)$, given a sample x and classifier model θ, measures how uncertain the classifier is about the value of class label Y. Therefore, we can ask the user for a ground-truth label when the entropy over Y given a specific sample x is the largest among all x in consideration:

$$x_H^* = \underset{x \in \mathcal{U}}{\operatorname{argmax}} \left(-\sum_y P_\theta(y|x) \log P_\theta(y|x) \right) \tag{6.5}$$

6.5 Outlier-Aware Uncertainty Sampling

Use of uncertainty sampling, however, can run the risk of choosing outliers as samples to query [81]. The reason is that outliers are far away from the other samples of the same class in the feature space; therefore, for most uncertainty metrics we use, outliers are likely to receive higher uncertainty scores than other samples. Unfortunately, knowing the label of outliers does not help in training a classifier because outliers are exceptions rather than representative examples that a classifier should learn from. As a result, actively choosing outliers for training can even "mislead" the classifier and end up degrading the accuracy.

To mitigate the negative affect of outliers, we used Outlier-Aware Uncertainty Sampling in tandem with the uncertainty sampling metrics. The idea is to select samples that are

uncertain but not outliers, i.e., samples that are representative of the underlying distribution (e.g. in a dense region of the feature space). To determine whether a sample is representative of the underlying distribution, we calculate the mean similarity between this sample and all the other samples. If a sample is close to many other samples in the feature space, its mean similarity with all the other samples will be high; on the other hand, for an outlier that is far from most samples, the mean similarity will be low. Incorporating this constraint into the uncertainty sampling metric, the new objective function is:

$$x^*_{OA} = \underset{x \in \mathcal{U}}{\operatorname{argmax}} \left(\phi(x) \cdot \frac{1}{N_{\mathcal{U}}} \sum_{x' \in \mathcal{U}} S(x, x') \right) \tag{6.6}$$

The first term $\phi(x)$ refers to one of the uncertainty metrics we described in Section 6.4. To be consistent with the argmax objective, for Least Confident uncertainty metric, $\phi(x)$ is defined as $\exp(-P_\theta(\hat{y}|x))$. Similarly, $\phi(x) = \exp(-(P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x)))$ for minimum margin metric, and $\phi(x) = H_\theta(Y|x)$ for the maximum entropy metric. The second term $\frac{1}{N_{\mathcal{U}}} \sum_{x' \in \mathcal{U}} S(x, x')$ measures the mean similarity, $S(x, x')$, between a sample x and all other samples x' in the unlabeled sample pool $\mathcal{U}$, where $N_{\mathcal{U}}$ is the total number of samples in $\mathcal{U}$. The complete algorithm is shown in Algorithm 2.

6.6 Discussion

The detailed protocol and the results of the active learning experiment is presented in Section 7.7. In our active learning experiment, it is assumed that the users are willing to provide labels and all the labels provided by the user are correct. Related study or future work on usability and interruptibility [74] can be further leveraged to adjust the frequency of requesting labels from users based on their preferences, and to improve the effectiveness of active learning in real practice. It would also be beneficial to study the ideal way (e.g. haptic, gestural, or audio-based interfaces) to engage users to provide labeled data for activity

Algorithm 2 Outlier-Aware Uncertainty-Sampling Active Learning Algorithm for Activity Recognition

Input: A sequence of initial unlabeled instances $\mathcal{U} = \{x_i | i = 1, ..., N_{\mathcal{U}}\}$; A set of initial labeled instances $\mathcal{L} = \{(x_i, y_i) | i = 1, ..., N_{\mathcal{L}}\}$; An initial classifier model θ; A pool window length L_{pwin}
Output: Updated activity classifier model θ
/* $N_{\mathcal{U}}$: the number of unlabeled samples available in the pool window */
while Activity Recognition Service is running **do**
    while $N_{\mathcal{U}} < L_{pwin}$ **do**
        $d \leftarrow$ getCurrentSensorData();
        $x \leftarrow$ extractFeatures(d);
        insert x into $\mathcal{U}$;
        $N_{\mathcal{U}} \leftarrow N_{\mathcal{U}} + 1$;
    end while
    $maxScore \leftarrow -\infty$; $x^* \leftarrow x_1$;
    for i **from** 1 **to** L_{pwin} **do**
        $score \leftarrow$ getOutlierAwareUncertainty(x_i);
        if $score > maxScore$ **then**
            $maxScore \leftarrow score$;
            $x^* \leftarrow x_i$
        end if
    end for
    $y^* \leftarrow$ queryForLabel(x);
    insert (x^*, y^*) to $\mathcal{L}$;
    $\theta \leftarrow$ trainClassifier($\mathcal{L}$);
    Remove all samples in pool $\mathcal{U}$; $N_{\mathcal{U}} \leftarrow 0$;
end while
return θ;

recognition using wearable and mobile devices.

In the current algorithm, once the best sample to query is selected from a pool, the rest of the samples in the pool are discarded and the system begins to collect a new pool of samples. One possible way to improve this is to keep the top K samples, which are rolled over to the next pool for consideration. Further user study may be required to evaluate the impact of asking a user to recall an activity that was performed a while ago, and to determine the optimal way to select a timeout threshold for unlabeled samples.

Chapter 7

Evaluation

In the previous chapters, we have covered the major research problems and challenges in human activity recognition, and have presented the details of the proposed activity recognition framework.

In this chapter, the effectiveness of the proposed approaches will be evaluated through system implementation, dataset collection, and real-world experiments. The main questions that are investigated and answered in this chapter are summarized as follows:

- What is the overall precision and recall rate of unseen activity recognition using NuActiv? How does the performance vary among classes? (Section 7.4.1)
- How does the recognition accuracy change with the number of unseen classes? (Section 7.4.2)
- How does the performance vary with the use of different classification algorithms for attribute detectors? (Section 7.4.3)
- How to select attributes based on their importance to unseen activity recognition? (Section 7.4.4)
- What is the cross-user performance, i.e. when the users in the training set are different from those in the testing set? Is the system able to generalize from one or a few users

to many new users? (Section 7.4.5)

- How does the attribute detection accuracy vary with the position and combination of the devices and sensors? (Section 7.4.6)
- How does NuActiv perform on recognizing unseen daily life activities? (Section 7.5.2)
- What is the effect of taking into account the sequential structure of human activities in NuActiv (Section 7.6)?
- How efficiently can the system reinforce its performance using active learning? How does the performance vary with different active learning algorithms? (Section 7.7.1)
- What is the effect of outlier-aware uncertainty sampling on active learning algorithms? (Section 7.7.2)

7.1 System Implementation

We have implemented and tested the system on Nexus S 4G phones and MotoACTV wristwatches [63]. A picture of our system running on these two types of devices is shown in Figure 7.1. The Nexus S 4G phone has a three-dimensional accelerometer and a gyroscope. The MotoACTV has a three-dimensional accelerometer. Both have Wi-Fi, Bluetooth radio, and GPS.

For the software part, we have implemented the code for feature extraction and the classification algorithm in the Java programming language. The code runs on the Android Operating System installed on the Nexus S 4G phones and the MotoACTV. For the classification algorithm, the Support Vector Machine classifier is implemented using the LibSVM library [12]. The probabilistic graphical model is implemented based on the Undirected Graphical Model (UGM) toolbox [78]. The screenshots of several operations of the application on the phone are shown in Figure 7.2.

All of the system components can run on a mobile phone or sensor-enabled wristwatch in

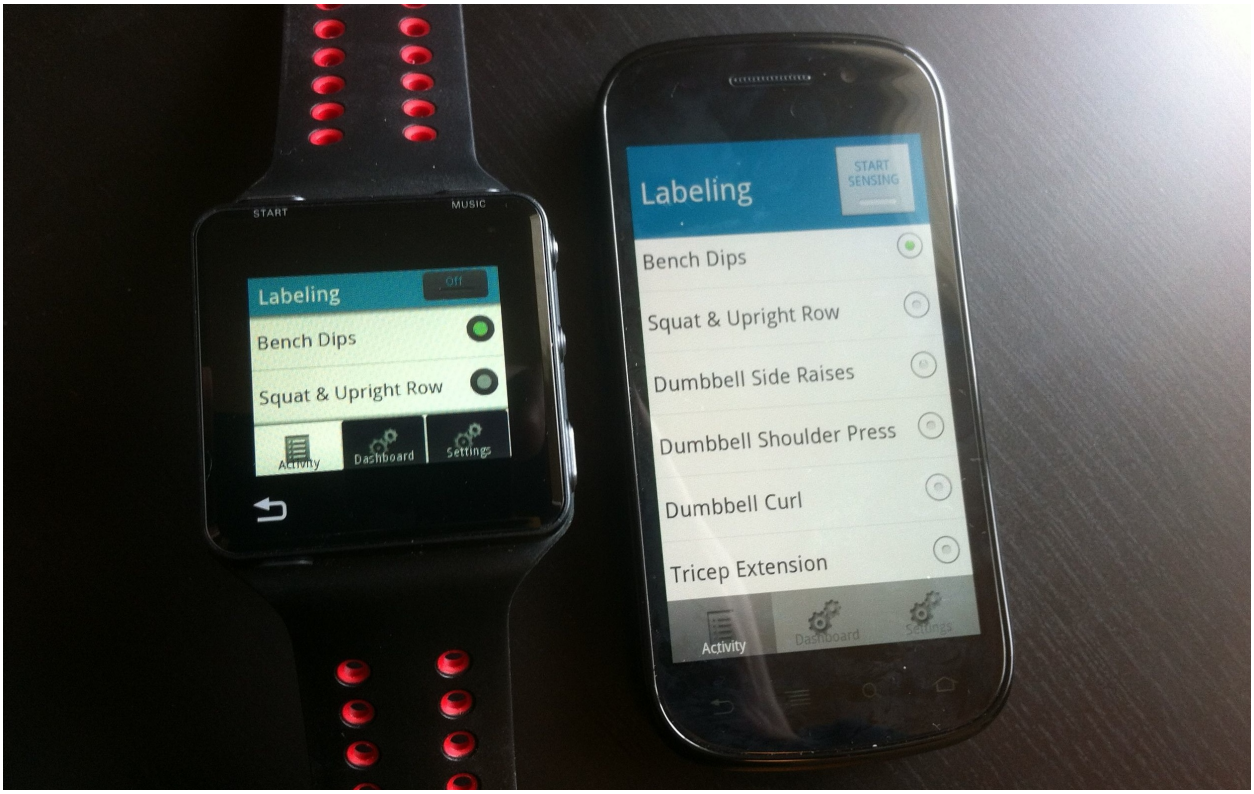


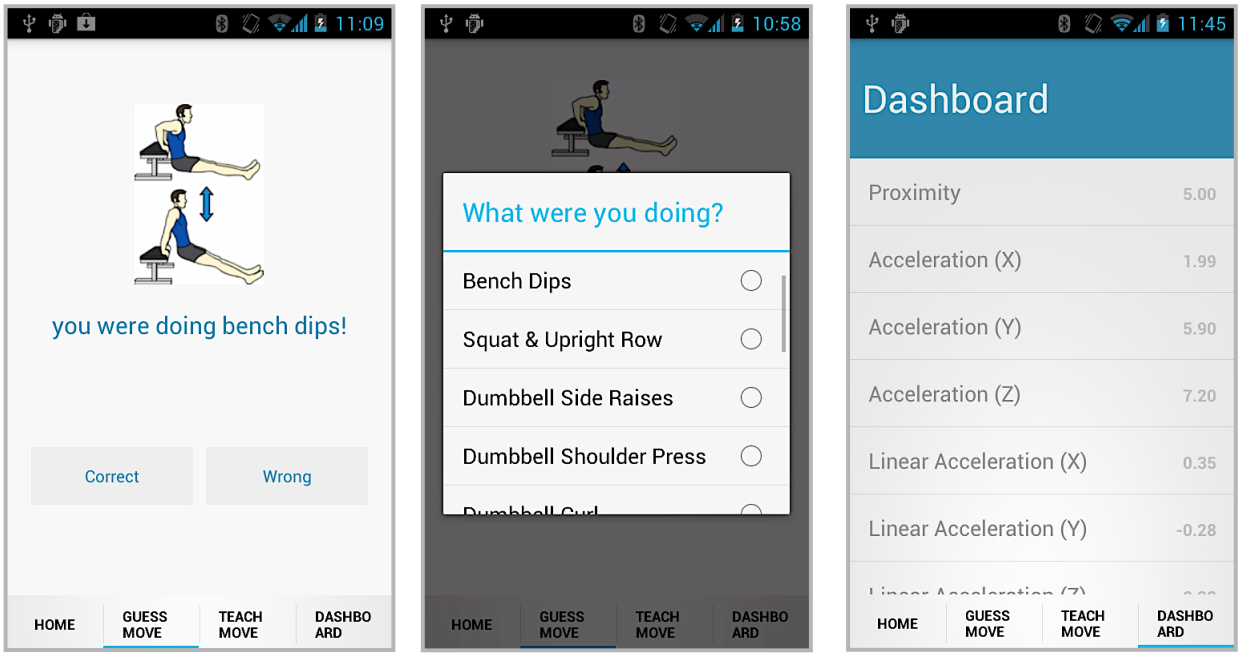
Figure 7.1: NuActiv running on MotoACTV wristwatch (left) and Nexus S 4G phone (right).

our implementation. In cases where offline model training is needed, the attribute detection models can be pre-trained on a server and then be downloaded to a mobile device.

7.2 Datasets

7.2.1 Exercise Activity Dataset

We conducted an experiment involving exercise activities on 20 subjects. Each subject is asked to perform a set of 10 exercise activities as listed in Figure 4.4 with 10 iterations. Before the experiments, the subjects are given instructions on how to perform each of the 10 exercise activities. More information about these activities can be found in the literature [13, 64]. During the experiment, each subject is equipped with three sensor-enabled devices:



Activity Recognition Output **Requesting User for Labels** **Sensor Data & Feature Dashboard**

Figure 7.2: The screenshots of our mobile app running NuActiv activity recognition system.

A Nexus S 4G phones attached to the arm using an armband, a MotoACTV wristwatch, and a second MotoACTV unit fixed at the hip position using a clip. A pair of three-pound dumbbells is also provided to the subject to perform some of the free-weight exercises (e.g. Dumbbell Side Raises, Dumbbell Curl, etc.).

For sensor data collection, we collected accelerometer and gyroscope data using our mobile application with a sampling rate of 30 Hz. The accelerometer and gyroscope data are acquired through the Sensor API [24] in the Android operating system [73]. The coordinate system is shown in Figure 7.3. For feature extraction, the sliding window size is empirically set to 1 second with 50% overlap, based on a 10-fold cross-validation test on the validation dataset to find the optimal parameter.

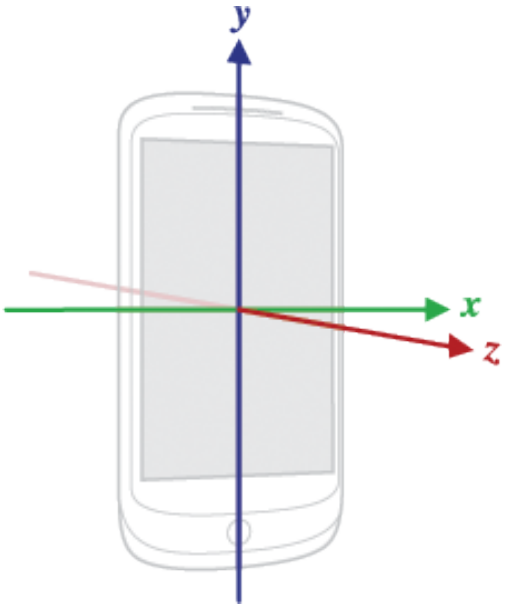


Figure 7.3: Coordinate system relative to a mobile device used by the Sensor API in the Android operating system. The coordinate system is the same as the one discussed in Figure 3.2 of Section 3.3.1.

7.2.2 Daily-Life Activity Dataset

For the scenario of recognizing daily-life activities, we use a published dataset collected by Technische Universitat Darmstadt (TU Darmstadt) [33, 83]. The dataset includes 34 daily life activity classes (including the *unlabeled* class) collected from one subject for seven days. The sensor data were collected using a wearable sensor platform with a three-axis accelerometer (ADXL330) worn on the wrist and the hip of the subject. The sampling rate is 100Hz, and the features are computed from a sliding window of 30 seconds with 50% overlap.

To apply NuActiv to the TU Darmstadt daily-life activity dataset, we defined a list of 17 attributes (as shown in Table 7.1) and an activity-attribute matrix[1] based on the 34 daily life activities in the dataset. It is to be noted that the list is not mutually exclusive or collectively exhaustive. We show that these semantic attributes defined by human knowledge can enable

[1]The activity-attribute matrix can be downloaded from the supplemental materials at `http://www.ece.cmu.edu/~hengtzec/data/DLActivityAttributeMatrix.pdf`

Table 7.1: Attribute list for daily life activities.

Type	**Attribute Name**
Basic	`Sitting`, `Standing`, `Walking`
Posture	`PostureUpright`, `PostureKneeling`
Hand/ Arm	`HandsOnTable`, `HandAboveChest`, `WristMovement`, `ArmPendulumSwing`
Motion Type	`IsTranslationMotion`, `IsCyclicMotion`, `IsIntenseMotion`
Relation	`IsWashingRelated`, `IsMealRelated`
Time	`IsTimeMorning`, `IsTimeNoon`, `IsTimeEvening`

unseen activity recognition using NuActiv in Section 7.5.

7.3 Evaluation Methodology

We used leave-two-class-out cross validation, the most widely used validation method used in the literature of zero-shot/zero-data learning [45, 68]. The validation scheme is used for recognizing unseen classes that do not have any sample in the training set. The traditional 10-fold cross validation is not applicable to unseen class recognition because it does not leave out all samples of certain "unseen" classes in the training step, so that every class will have some samples in the training set.

The leave-two-class-out cross validation works as follows. Suppose there are a total of N classes. Each time we first train our system on $(N-2)$ classes, and then test the discriminative capability of the classifier on the remaining 2 classes that were "unseen" by the system during the training process. We repeat this test for all $\binom{N}{2}$ unseen/seen class combinations. Finally, the average performance over all tests is reported.

The results are reported in precision, recall, F_1-score, and accuracy. These metrics show different aspects of the performance of an activity recognition system. Specifically, the

metrics are defined as follows:

$$precision = \frac{TP}{TP + FP} \tag{7.1}$$

$$recall = \frac{TP}{TP + FN} \tag{7.2}$$

$$F_1\text{-}score = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{7.3}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7.4}$$

where TP, FP, TN, and FN denotes true positive, false positive, true negative, and false negative, respectively. Precision indicates the percentage of times that a recognition result made by the system is correct. Recall means the percentage of times that an activity performed by a user is detected by the system. F_1-score is a integrated measure that combines both. For overall performance across all classes, the *accuracy* is computed as the number of correctly recognized samples divided by the number of all samples in the test set.

7.4 Case Study I: Exercise Activities

7.4.1 Unseen Activity Recognition Result

The confusion matrix of recognizing previously unseen exercise activities is shown in Figure 7.4. The rows indicate ground-truth classes, while the columns indicate recognized classes. The correct predictions fall on the diagonal, whereas large off-diagonal values indicate challenges for the system. The average accuracy is 79% over all activities, among which the system achieved a promising recognition accuracy of 80-90% for five activity classes. It is to be noted that in these results, the target activities are recognized under the situation that no training data of any target activity class were given to or seen by the system during the training phase. The results support our hypothesis that unseen new human activities can be recognized with a respectable accuracy using the proposed semantic-attribute-based

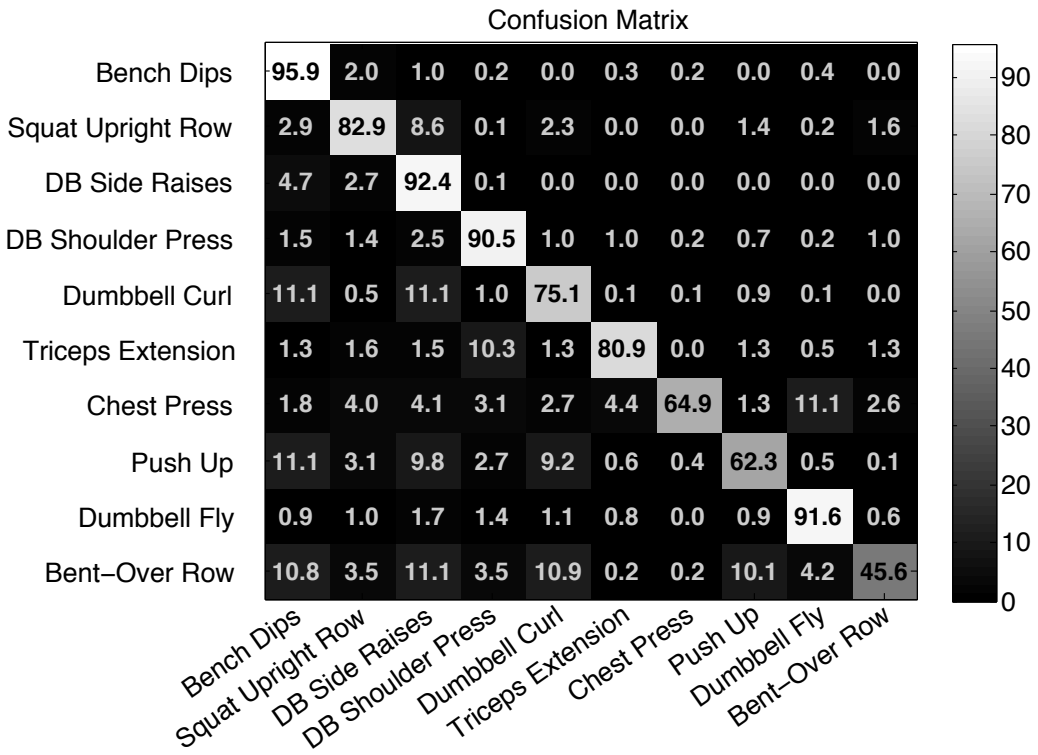


Figure 7.4: Confusion matrix of recognizing unseen exercise activities using the two-layer attribute-based recognition presented in Chapter 4. The numbers are shown in percentages (rows: ground-truth classes; columns: recognized classes).

learning approach in NuActiv.

One observation that we can draw from the experimental result is that misclassification usually happens when two activities only differ in one attribute. In this case, the success of recognition depends heavily on the detection accuracy of the presence of that particular attribute. For example, "`DumbbellFly`" is classified as "`ChestPress`" because these two activities are inherently similar and are only different in the attribute "`ArmCurl`" (see the activity-attribute matrix in 4.4). The problem can potentially be overcome by including additional sensors/modalities so that other discriminative attributes can be used to further distinguish two similar classes.

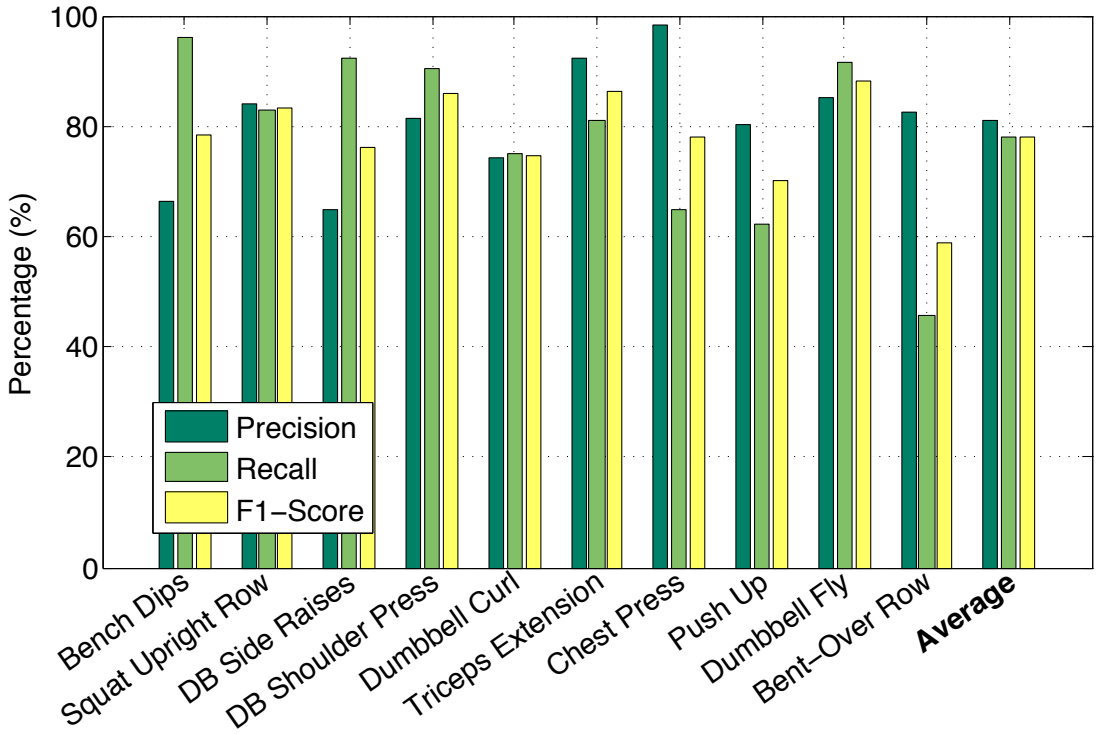


Figure 7.5: Precision and recall rate of recognizing unseen activities using the exercise dataset.

7.4.2 The Impact of The Number of Unseen Classes

The capability to recognize unseen new activity classes is built on the knowledge learned from the seen activity classes. As we can imagine, if all the classes were unseen, the system has nothing to learn from and thus is not able to recognize any activity with reasonable accuracy. To understand the capability and limitation of our approach, we conducted the following experiment: For a total of k classes, we vary the number of unseen classes (n_u) in the testing data from 2 to k, where the corresponding number of seen classes ($n_s = k - n_u$) in the training data varies from $k-2$ to 0. For each number of unseen classes, we repeat the test for all $\binom{k}{n_u}$ combinations and report the average results.

The result is shown in Figure 7.6, where the total number of classes is ten ($k = 10$). We observe that the recognition accuracy gradually degrades as the number of unseen classes in

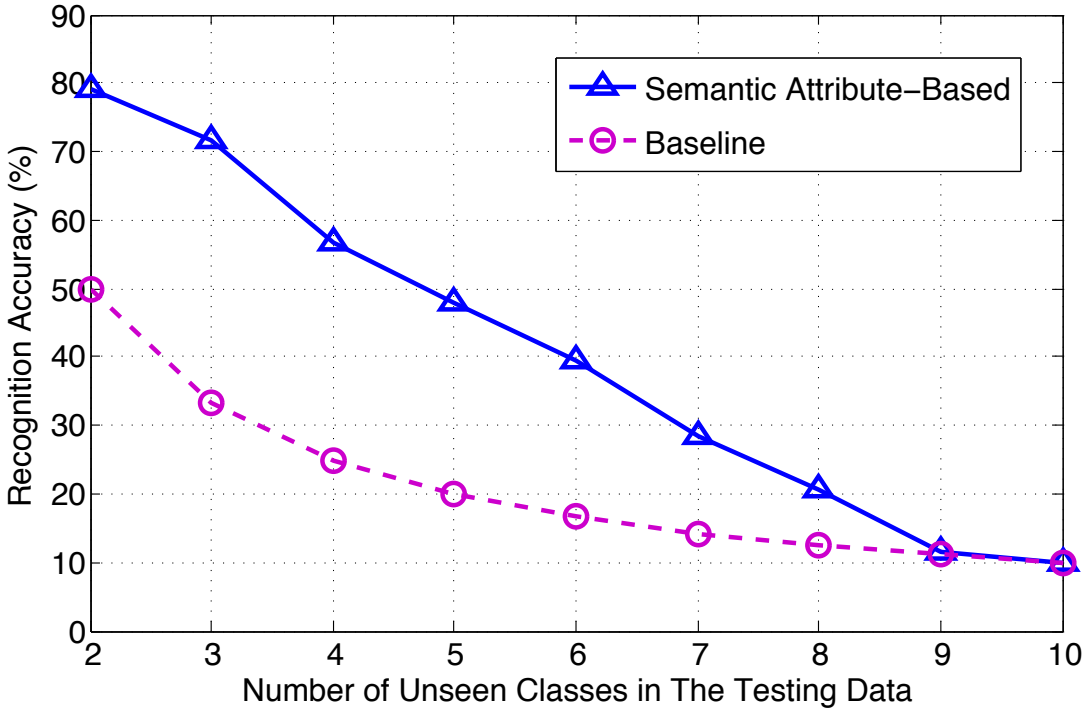


Figure 7.6: Accuracy vs. number of unseen classes in the testing dataset.

the testing data increases (i.e. the number of seen classes in the training data decreases). This is in accordance with the expectation, because it gradually becomes difficult for the system to generalize to a large number of unseen activity classes based on only a few seen classes. Furthermore, a successful unseen activity recognition relies on an accurate attribute detection. To accurately detect an attribute, it is important for the training classes to cover both positive and negative examples of the attribute. Therefore, the larger the seen-to-unseen class ratio is, the more likely that we can recognize unseen activities effectively.

We also compare our results with a baseline approach. The baseline approach is the random-guess prediction given the number of unseen classes, which is the best that a supervised learning-based activity recognition system can do, assuming each class is equally probable and there are no training samples for the target classes. From Figure 7.6, we can see that our semantic-attribute-based approach is 20-30% better than the baseline for most

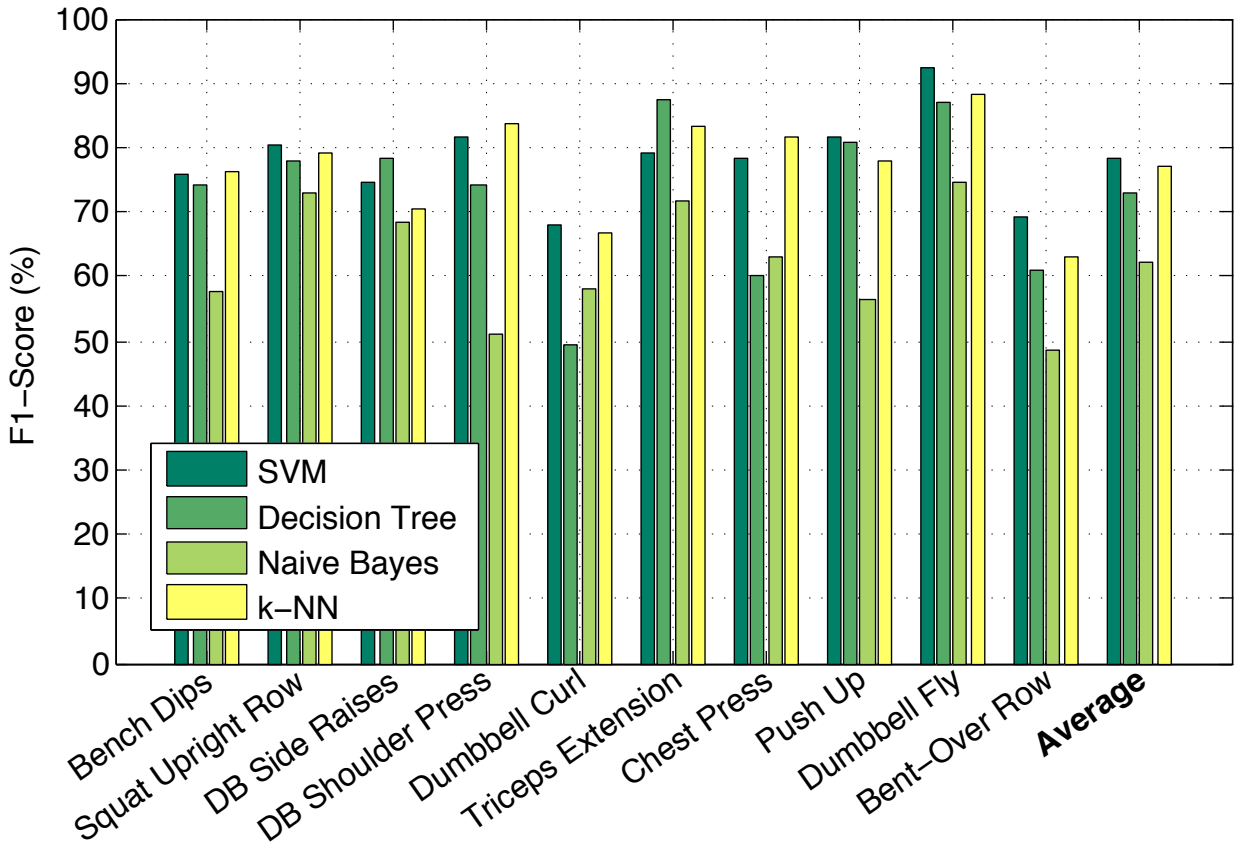


Figure 7.7: F1-score of unseen activity recognition vs. different classifiers for attribute detectors.

cases, except for the cases where almost all the classes were unseen (when the system has seen zero to two classes in the training data). The results suggest that NuActiv is a viable approach to unseen activity recognition.

7.4.3 Comparison of Different Attribute Detectors

We also compare the SVM classifier with other classifiers that are widely used in the related work, including the Decision Tree classifier, Naive Bayes classifier, and k-Nearest Neighbor (k-NN) classifier [4, 51, 53]. For k-NN, the optimal result with $k = 3$ is reported.

The results are shown in Figure 7.7. SVM outperforms the Decision Tree and Naive Bayes

classifier on average and for most of the classes if we break down the results by activity class. Overall, the accuracy using k-NN is comparable to the result using SVM. However, k-NN classification requires storage of and access to all the training data. Thus, k-NN is less scalable for a large dataset and less practical to run on mobile devices given their limited storage. Therefore, we used SVM for our experiments and implementation on the mobile devices.

7.4.4 Evaluation of Attribute Selection

We now investigate the importance of each semantic attribute and gain insights into attribute selection. The selection of attributes is important in two aspects. The first one is *discriminability*, meaning how well can an attribute discriminate between different high-level classes. The second one is *detectability*, namely how accurately can we detect the presence or absence of an attribute.

To test the *discriminability*, we conducted the following experiment: First, we run the unseen activity recognition test using all n_A semantic attributes ($n_A = 7$ for exercise activities). Then, we run n_A tests where each time we exclude one of the attributes and observe the change of the performance. If the performance drops significantly when an attribute is excluded, then being able to detect this attribute is important to accurately recognize the high-level activities. On the other hand, if the performance does not change much without an attribute, then the attribute is less likely to be important. To test the *detectability*, we compute the detection accuracy of each attribute. The accuracy is computed as the number of times an attribute is correctly detected divided by the number of times an attribute appears in the samples of the testing data.

The results of the discriminability and detectability test are shown in Figure 7.8 and 7.9, respectively. From the average F1-score (the rightmost bar group) in Figure 7.8, we can see that the attributes `ArmUp`, `ArmDown`, and `ArmFwd` have a higher impact on the activity

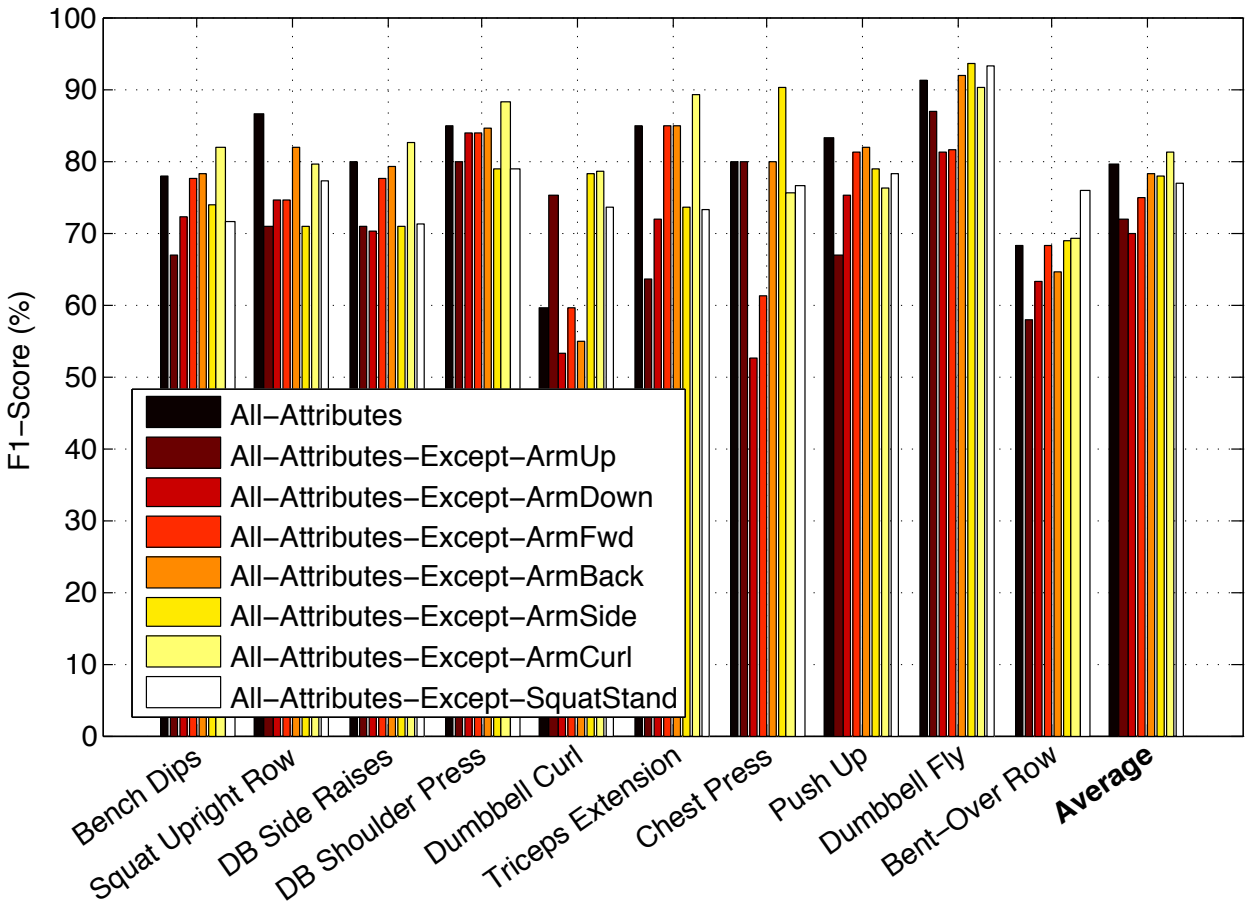


Figure 7.8: Discriminability: F1-score vs selected attributes. Each bar in a group represents an attribute that was unselected.

recognition performance than other attributes. However, from the results broken down by activity class, we can see that an attribute may be important to some activity classes but not for other classes. Therefore, the selection of attributes also depends on the characteristics of the targeted classes. One reason for these phenomena is the inherent attribute-composition of an activity. Another possible reason is that some attributes are easier to detect than others. As shown in Figure 7.9, the system generally achieves higher accuracy detecting the first four attributes (`ArmUp`, `ArmDown`, `ArmFwd`, and `ArmBack`). These differences in detectability can be caused by the positions and types of the sensors used, the type of classifier used for detection, and the consistency of the presence of an attribute given an activity is performed.

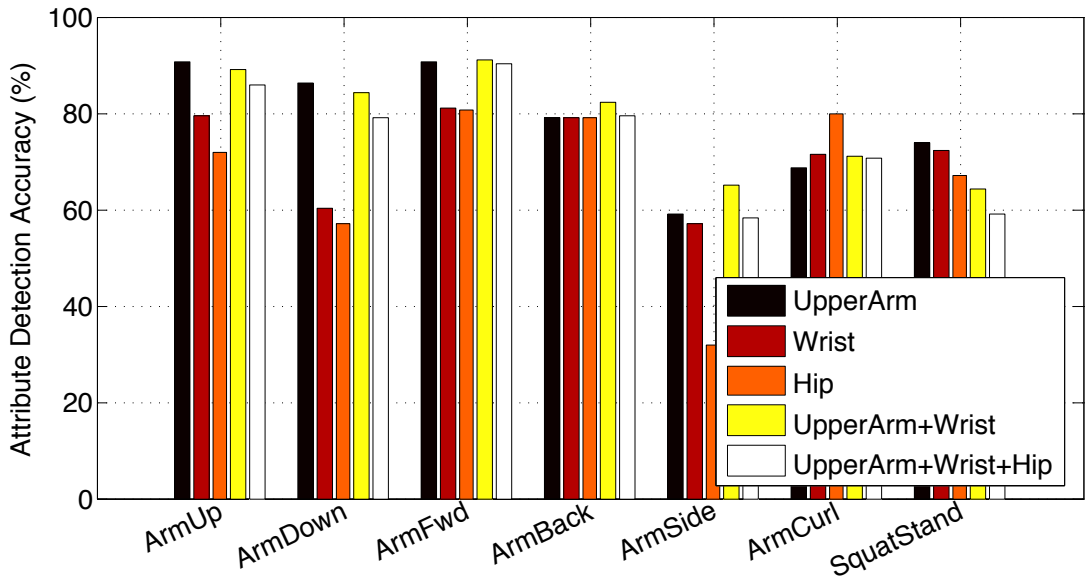


Figure 7.9: Detectability: Attribute detection accuracy vs. device/sensor positions for each attribute.

7.4.5 Cross-User Activity Recognition Results

It is important for an activity recognition system to not only be able to recognize the activities of the users it has seen, but also be able to generalize to the activities of new users it has never seen before. Our main hypothesis is: The association between an activity and the semantic attributes, as specified in the activity-attribute matrix, is the same across users. When different users perform the same activity, although the low-level sensor data vary, the mid-level attributes hold across users and thus the trained attribute model can be trained on one person and applied to another.

To evaluate the generalizability and the limitation of our approach, we randomly divide the 20 users into two equal sets. While fixing the test set to be the 10 users in the first held-out set, we iterate the number of training users from 1 to 10, randomly chosen from the second set. For each test, we repeat the random choice 100 times and report the average performance.

The results are shown in Figure 7.10. As we can see, the performance stays approximately

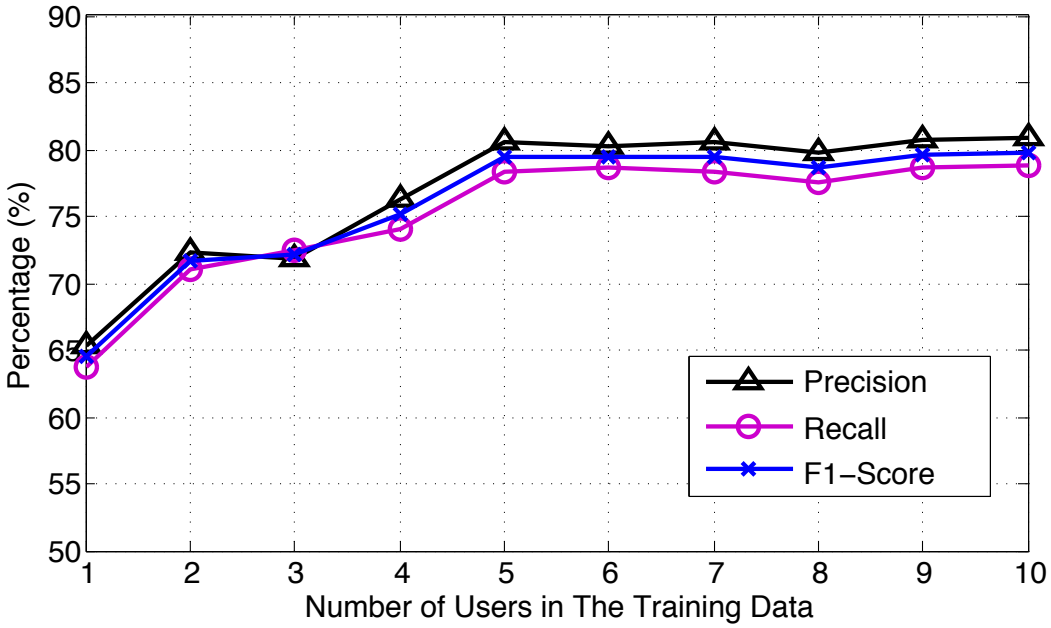


Figure 7.10: Cross-user recognition accuracy vs. number of seen users in the training data. The testing set includes 10 users that are different from those in the training data.

constant when the number of seen users in the training data is equal to or greater than five. Furthermore, the precision, recall, and F1-score are almost the same as the case where the data of all the users exist in both the training set and the testing set, as shown previously in Figure 7.5. The F1-score decreases slightly when number of seen users in the training data falls below four, yet the system can maintain an F1-score of over 70% when having seen 2–4 users in the training data. The edge case happens when the system has only seen one user in the training set, where the F1-score is 60%. Overall, the system is able to achieve 70-80% accuracy after training on two or more users in the training set.

7.4.6 Impact of Device Position on Attribute Detection Accuracy

An attribute is often inherently associated with a characteristic of a human activity or a motion of a specific part of the human body. Therefore, we conducted experiments to understand how the position or the set of positions at which the sensors are placed affects

the attribute detection accuracy, which in turns affects the final activity recognition accuracy. When collecting the exercise activity dataset, we have placed sensor-enabled phones/devices on three different body positions of the users (as described in Section 7.2.1).

The experimental results using sensor data from different body positions is shown in Figure 7.9. It is observed that while using the upper arm sensors (phone in an armband) usually achieves better and more stable accuracy than using the wrist sensors (wristwatch), combining these two data sources leads to improvement in accuracy in some cases. One possible reason is that while the movement of the wrist is of larger variation and less predictable, it complements the limitation of what can be observed by the upper arm sensors. Adding the hip sensor did not improve the accuracy, possibly because most attributes defined in our case study do not involve changes in lower-body postures. It is to be noted that the results do depend on the activity domains and sensor types.

7.5 Case Study II: Daily-Life Activities

7.5.1 Reproduction of Results in Previous Work

For the daily life activity recognition experiments, we used the public dataset of 34 daily life activities provided by TU Darmstadt [33], as described in Section 7.2.2. We first implement the supervised learning method closely following the experiment protocols in the paper of the dataset provider [33]. Given that all the classes were seen in the training set, our implementation achieved 71.2% accuracy, which is very close to 72.7% as reported in the previous work [33]. This reproduction of previous results confirms that our use and understanding of the dataset and features are valid.

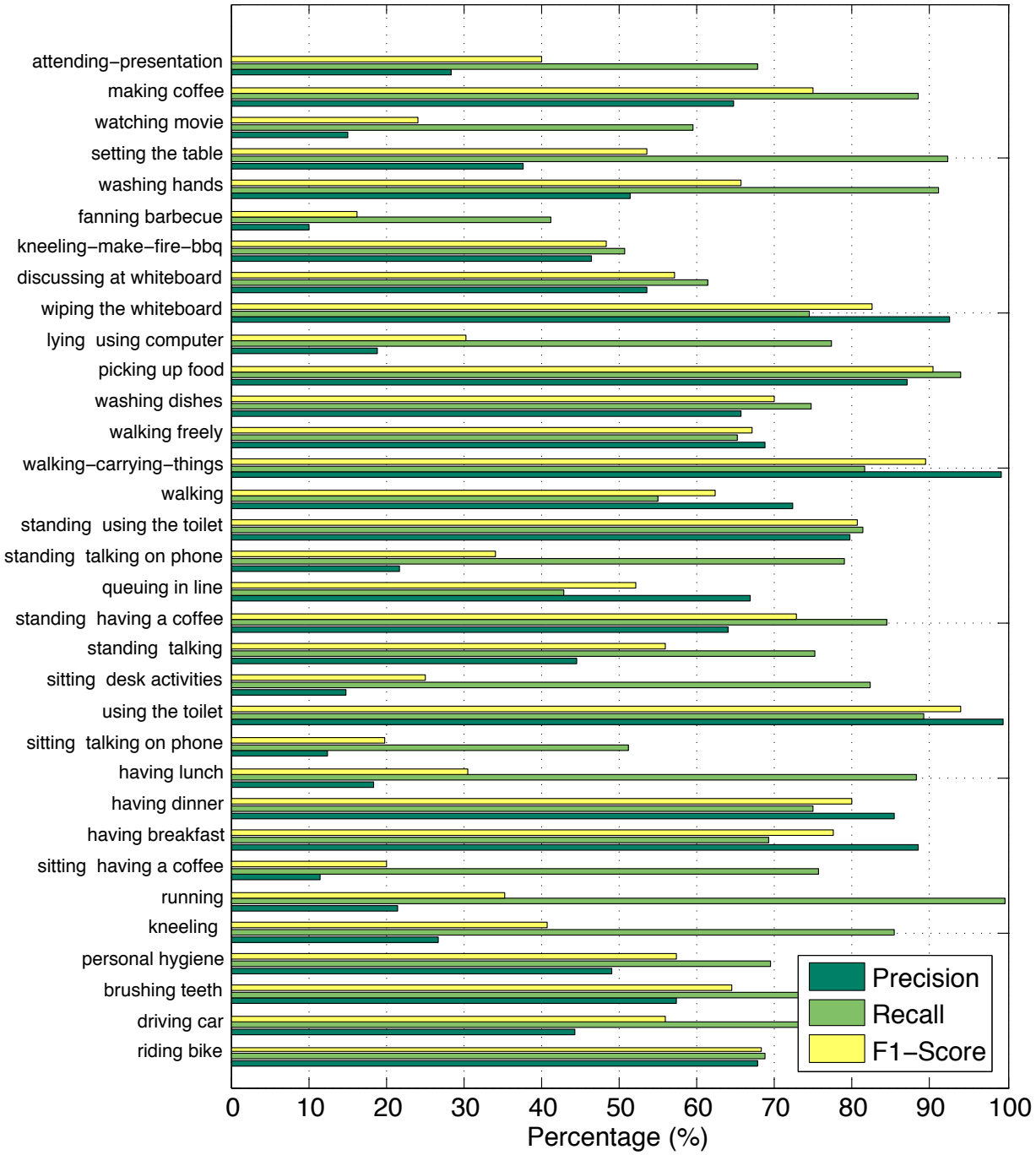


Figure 7.11: Precision, recall, and F1-score of recognizing unseen daily life activities in the TU Darmstadt dataset using NuActiv.

7.5.2 New Task: Recognizing Previously Unseen New Daily Life Activity

After successfully reproducing the results in the previous work, we proceeded to a new problem—recognizing unseen daily life activities—which has not been addressed before in the activity recognition literature. We applied the NuActiv system and algorithms to the 34-daily-life dataset [33], and evaluated the performance using the same evaluation methodology (leave-two-class-out cross validation) as described in Section 7.3. The results are shown in Figure 7.11. We can see that for some classes the system can achieve high recall and lower precision, and vice versa for other classes. Overall, the system achieves 60-70% precision and recall rate for most classes. The mean precision and recall rate is 52.3% and 73.4%, respectively, averaged over all classes. Some classes, such as "sitting-desk-activities" or "sitting-talking-on-phone", do not have a clear difference in attributes since we only have inertial sensor data available in the dataset. Therefore, the system tends to have a low precision rate on these classes. This problem can be alleviated by incorporating extra sensing modalities such as ambient sound. While there is clearly room for improvement, the fact that our system is able to recognize an unseen daily life activity class with no training data with a reasonable accuracy is a new result in the field of activity recognition.

7.6 Evaluation of Semantic Attribute Sequence Model

In the previous sections, we have discussed the results using the NuActiv framework with the attribute-based learning algorithm presented in Chapter 4. In this section, we will discuss the experimental results where the sequential structure is added to the framework using the semantic attribute sequence model presented in Chapter 5.

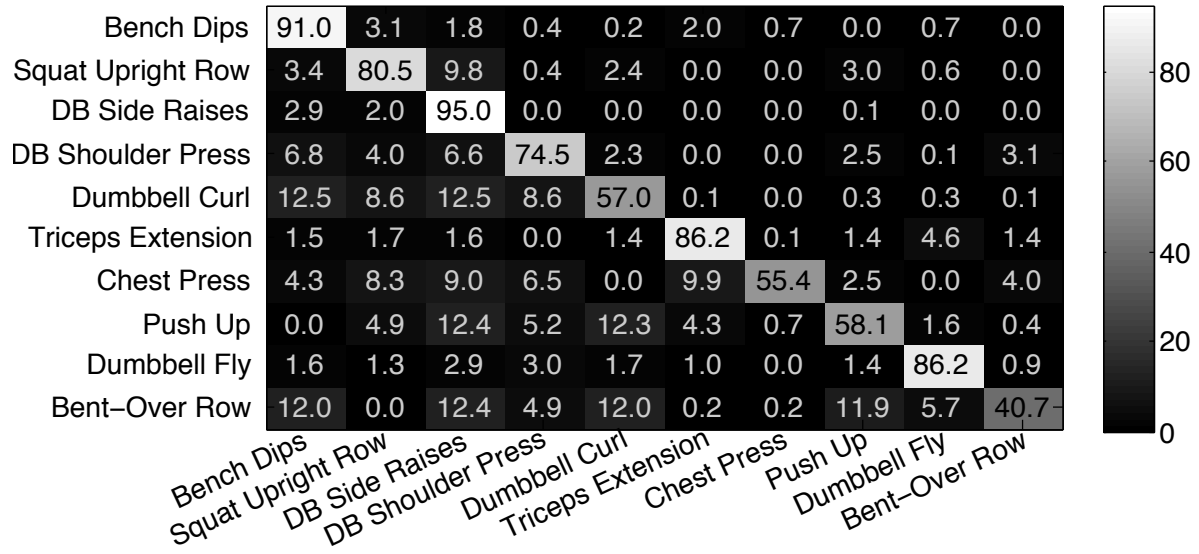
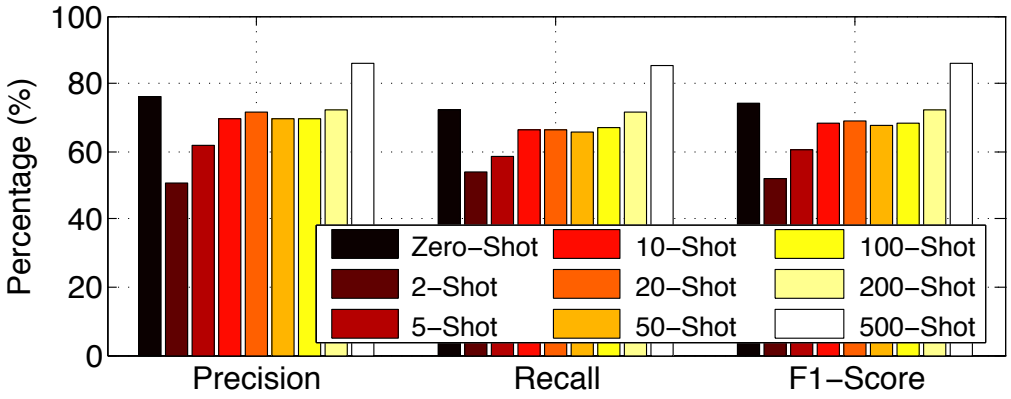


Figure 7.12: Confusion matrix of unseen exercise activity recognition using the *semantic attribute sequence* model presented in Chapter 5. The numbers are shown in percentages (rows: ground-truth classes; columns: recognized classes).

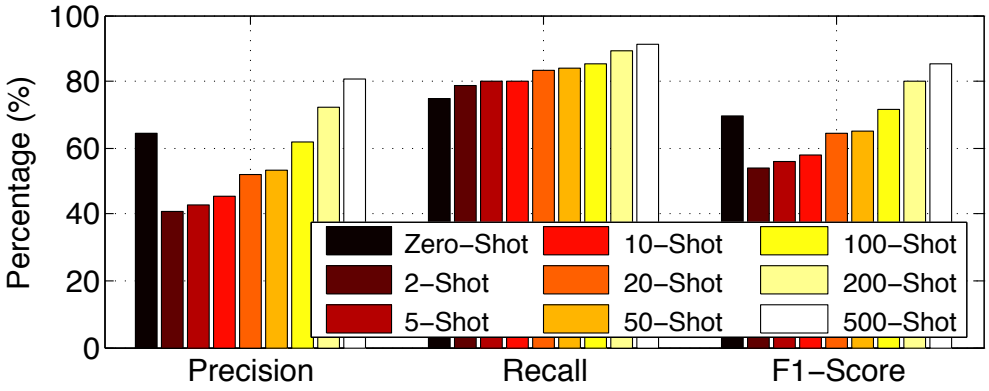
7.6.1 Case Study I: Unseen Exercise Activity Recognition

As shown in the confusion matrix in Figure 7.12, our approach achieved 76% precision and 72% recall averaged over all activities. The standard deviation of precision and recall across different users is 13% and 16%, respectively. The results show that even without training samples, unseen new activities can be recognized with a promising accuracy. The results are also comparable to those reported in Section 7.4.1. On the other hand, the limitation of the approach is observed when two activities only differ in one or two attributes (e.g. Push Up and Dumbbell Side Raises), or when an attribute is not consistent for every person (e.g. `ArmCurl`).

We compare our approach with existing supervised learning approach using a linear-chain CRF [38], which belongs to the same model family except without the semantic attribute layer. Since supervised learning cannot recognize unseen activities without training samples, we compare with the cases where it is possible to obtain n samples of each unseen activity performed and labeled by the users (denoted by n-shot learning). In contrast, zero-shot



(a) Exercise activities.



(b) Daily life activities.

Figure 7.13: Comparison between proposed zero-shot learning and n-shot supervised learning (n labeled training samples for each target activity).

learning can be thought as having users provide a one-time description of an unseen activity using the semantic attributes. As shown in Figure 7.13(a), zero-shot learning outperforms supervised learning with up to 200 labeled samples. This shows that zero-shot learning is effective for bootstrapping an activity recognition system when sufficient labeled samples for every activity are not available. On the other hand, in cases where obtaining a large amount of labeled data is inexpensive for every activity, supervised learning tends to achieve higher accuracy.

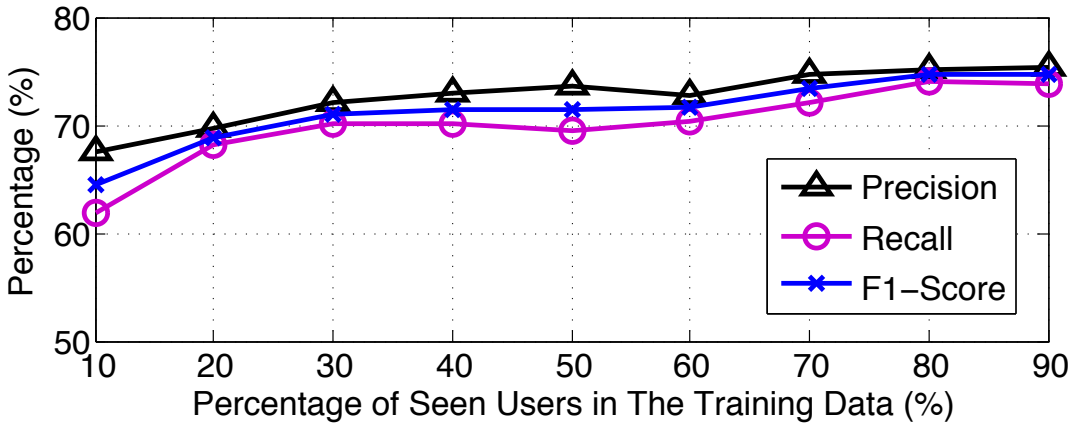


Figure 7.14: Cross-user recognition accuracy using the semantic attribute sequence model. The testing set includes 10% of the users, which are different from those in the training data.

7.6.2 Case Study II: Unseen Daily Life Activity Recognition

We applied the same approach to the daily life activity dataset. The average precision and recall is 69% and 75%, respectively, which outperformed the results (52.3% precision and 73.4% recall) reported in [16]. The results suggest that the semantic attribute sequence model can better capture the temporal dependency in the daily life activities, in comparison to applying an activity classifier to each frame independently [16]. As shown in Figure 7.13(b), the F_1-score of zero-shot learning outperforms supervised learning approach with less than 100 labeled samples. Possible reasons for a lower precision than the exercise activity recognition results include a larger number of different activity classes, and a larger feature variation in daily life activities because they are less well-defined than exercise activities.

7.6.3 Cross-User Unseen Activity Recognition Results

As discussed in Section 7.4.5, the ability to apply the model learned from some users to new users is important for an activity recognition system. Figure 7.14 shows the results on unseen exercise activity recognition using the semantic attribute sequence model when the users in the training set and testing set are different. Our approach achieves a stable

accuracy of 70-75% when 20% or more users were seen in the training set. The general trend of the results is similar to those reported in Section 7.4.5 using the semantic attribute-based activity recognition. The results show the learned attribute sequence model can be generalized to new users and new activities, and the performance degrades gracefully with the decrease of seen users. In comparison, in the experiments where the training data and testing data are drawn from the same user, the average precision and recall increases to 78% and 76%, respectively.

7.7 Evaluation of Active Learning for Activity Recognition

7.7.1 Comparison of Active Learning Algorithms

In addition to unseen activity recognition, we further evaluate how the system can improve itself using minimal user feedback. The active learning algorithms we used for the experiment are explained in Chapter 6.

Following the experiment protocols in the active learning literature [81], our experiment setting is described as follows. The data set we used is the 34-daily-life dataset [33]. Each sample is a feature vector extracted from a window of 30 seconds of sensor data. An initial labeled set $\mathcal{L}$ of 50 samples from Day 1 of the dataset is provided to the system. Then, an unlabeled set $\mathcal{U}$ of 11087 samples from the rest of Day 1 to Day 5 is sequentially provided to the system in time-order. Each time the system improves itself using the newly acquired labeled data, we evaluate its performance on a separate test set $\mathcal{T}$ of 5951 samples from Day 6 and 7. The active learning is performed as described in Algorithm 2, with $L_{pwin} = 100$ and SVM classifiers.

The results are shown in Figure 7.15. Using active learning, the classifier generally

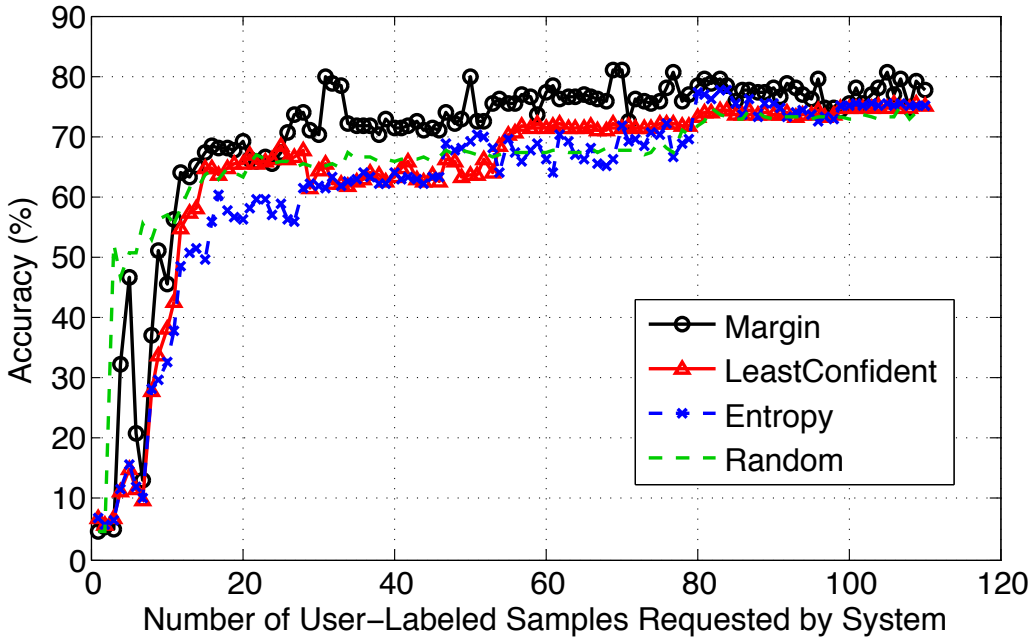
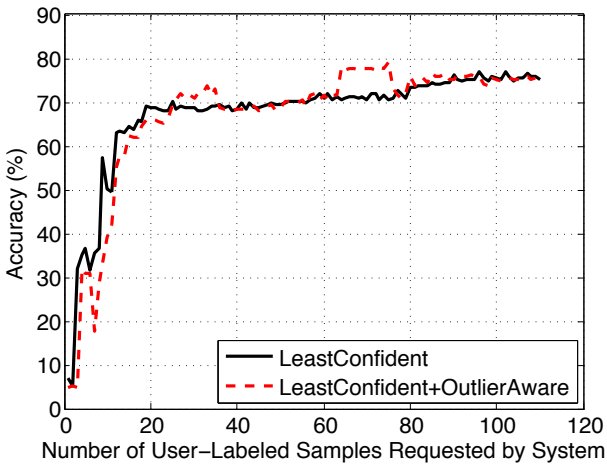


Figure 7.15: Recognition accuracy vs. user labels requested in the active learning experiment.

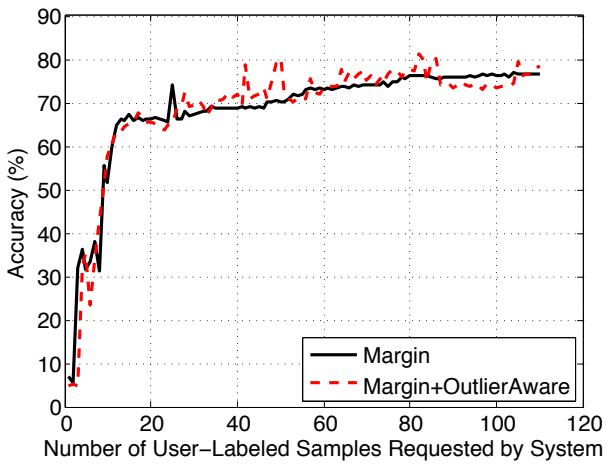
improves faster (using less labeled samples from the user) than the *random* baseline (i.e., randomly choosing samples from the unlabeled dataset and asking the user for labels without using active learning approaches). The margin-based uncertainty metric achieved 70% accuracy using only 30 labeled samples from the user and converged faster than other approaches. The entropy and least-confident metrics yielded comparable results.

7.7.2 Outlier-Aware Uncertainty Sampling Results

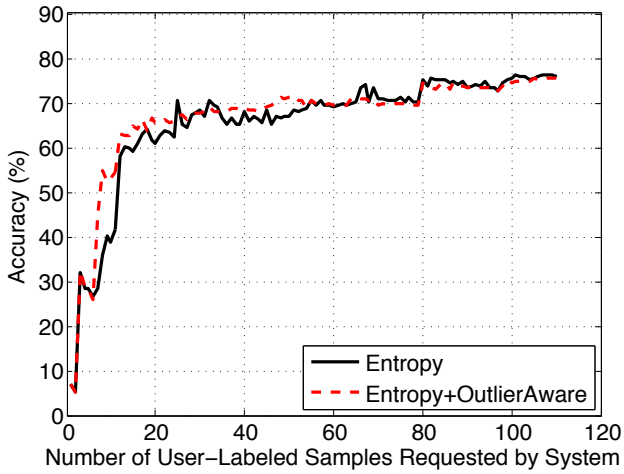
We further incorporate the outlier-aware uncertainty sampling as described in Section 6.5, and compare the results with those not using outlier-aware uncertainty sampling. The results are shown in Figure 7.16. It is observed that given the same number of user-labeled samples requested by the system, using outlier-aware uncertainty sampling leads to comparable or better accuracy in some cases when compared to active learning algorithms without outlier-awareness. In general, though, we do not observe a large improvement on the dataset. The



(a) Least-confidence-based sampling.



(b) Margin-based sampling.



(c) Entropy-based sampling.

Figure 7.16: Comparison between the learning curve of active learning algorithms with/without outlier-aware uncertainty sampling.

actual amount of improvement depends on the relative weighting between the uncertainty term and the outlier-awareness term in Equation 6.6, and on whether there are a large number of outlier samples in the input data.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

This thesis presents a study of the hierarchical and sequential structure of human activities, and proposes a new activity recognition framework. Most existing machine-learning-based approaches can only classify sensor data into one of the pre-trained classes in the training data, and thus cannot recognize any previous unseen class without training data. To overcome the existing challenges and limitations, the new NuActiv framework uses semantic attribute-based zero-shot learning along with a probabilistic graphical model to recognize unseen new activity classes. This zero-shot recognition is achieved by injecting human knowledge about the relationship between high-level activities and mid-level semantic attributes, and by reusing and generalizing the attribute model learned for other seen activities. Furthermore, the active learning approach efficiently improves the recognition accuracy of the system using minimal user feedback.

Experimental results show that the proposed approach achieves 70-80% precision and recall in recognizing previously unseen new activities, and outperforms supervised learning where hundreds of labeled samples for the unseen activities are provided to the supervised

learner. This thesis reports the first results on attribute-based zero-shot learning for human activity recognition. The results extend and advance the state of the art in human activity recognition, and represent an important step towards bridging the gap between computers and humans.

8.2 Future Work

Building on the previous work in ubiquitous computing and machine learning, this thesis also opens doors for many future research opportunities in the related fields. Some possible future research directions and areas for improvement are summarized in the following subsections.

8.2.1 From One to Many: Group Activity Recognition

This thesis considers and studies the activities that are performed by a single person, rather than a group of people. One direction of future work is to extend the study from individual activities to group activities. In general, the notion of a "group" can be a small group of several people [48], a medium-sized community on a campus, or even a larger population in an urban environment [1]. There have been a few research projects on group activity recognition [48, 76, 99], but most of them are vision-based systems. By extending existing sensor-based activity recognition systems to group activities, we can obtain insights into the interaction among individuals, family members, coworkers, and more. This allows computers to have a more holistic understanding about high-level human activities.

8.2.2 From Known to Unknown: Open Set Recognition

As stated in Table 1.1 in Chapter 1, the problems studied in this thesis are learning problems with *known classes*, where the names of the target classes to recognize are known at the time of training and testing. This is also referred to as *closed set recognition* [77]. In some cases,

however, we do not have knowledge of the entire set of possible classes. As an example in human activity recognition, a user may be performing a new job function or engaging in a new sports activity. It is possible for a system to ask the user for a name for the new activity, if the pattern is seen repeatedly. Another example is that a patient may perform some abnormal activities that are not in the set of common daily life activities. There has been little research on this kind of *open set recognition* problem, where incomplete knowledge of the class space is present at training time, and *unknown classes* can be added to an algorithm during testing time [31, 77]. Therefore, the ideal approach to detecting unknown/rare classes [31], distinguishing different unknown classes, and incorporating human knowledge remains an open challenge.

8.2.3 From Absolute to Relative: Alternative Representations

The approach and experiments presented in this work are based on an assumption that each target activity can be represented as a vector of binary attribute values (i.e., a bag-of-attributes representation). However, there are certain attributes that are similar in nature, except different in intensity, frequency, or other aspects. For example, in our current representation, {`Walking`, `Jogging`, `Running`, `Sprinting`} are treated as four different binary attributes, each of which can be true (present) or false (not present). One alternative representation is to combine these related attributes into a single attribute such as `HumanLocomotion`, which can take on multiple values representing the difference in the speed of movement.

Another research direction is to explore the use of *relative attributes* [70, 71], such that it is possible to inject human knowledge in the form of "Activity A has more `AttributeX` than Activity B". For example, in our current representation, we can say that both *playing soccer* and *playing baseball* involves the attribute `Running`, but we are not able to describe to which extent that `Running` is involved in each activity. As an improvement, one can extend the framework so that an activity recognition system is able to understand the statement

"*Playing soccer* has more `Running` than *playing baseball*" and use it for training the classifiers. These alternative representation of attributes will allow a system to better understand and model the relationship between related high-level activities.

8.2.4 From Single to Parallel: Concurrent Activities

A person can be engaged in several activities concurrently [26, 38]. For example, one can be watching television while talking to friends and having dinner. These behaviors need to be recognized using a different approach from that for sequential activity. Another related notion is *interleaved activities* [26]. As an example, if a friend calls you while you are cooking, you might talk to your friend on the phone for a minute, while you continue or suspend the cooking activity [38]. As an analogy with operating systems, human activities are "multi-threaded" rather than "single-threaded." However, the limitation of most existing approaches is that only single-threaded activities are supported. Generalizing existing approaches from recognizing single to concurrent activities can make activity recognition systems more applicable to real world situations.

8.2.5 From Now to Ever After: Long-Term Behavior Modeling

Most of the activities studied in this thesis are short-term activities, whose duration ranges from seconds to tens or hundreds of minutes. There are many higher-level human activities and routines that last for a longer time—e.g. "studying for an exam", "participating in a weight watcher program", "shopping for holiday season"—which are not covered in our study. These semantics are more complex to understand and are composed by a set or a sequence of high-level activities. Some previous work focused on the sensing platform design or energy efficiency issues of long-term activity monitoring [41, 51, 54, 90]; however, less research has been conducted on clustering or inferring the long-term behavior. Future research in this

direction can push the understanding of human activities to the order of days or months.

8.3 Final Thoughts

This is an exciting time to work on human activity recognition. From the sensing point of view, the source of sensor data not only includes the omnipresent mobile phones, but all the emerging wearable devices and ambient sensors. From the inference point of view, the recent advance in machine learning has made it possible to learn high-level concepts about human activities by leveraging these heterogeneous sensor data, either in a supervised or unsupervised fashion. I look forward to a day when a computer, with the intelligence to understand what humans are doing, is not only fast, but also thoughtful and considerate.

Bibliography

[1] S. Abdullah, N. D. Lane, and T. Choudhury. Towards population scale activity recognition: A framework for handling data diversity. In *Proceedings of The 26th AAAI Conference on Artificial Intelligence*, AAAI'12, 2012.

[2] J. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428 – 440, 1999.

[3] M. Azizyan, I. Constandache, and R. Roy Choudhury. SurroundSense: Mobile phone localization via ambience fingerprinting. In *Proceedings of The International Conference on Mobile Computing and Networking*, pages 261–272, 2009.

[4] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *Proceedings of The International Conference on Pervasive Computing*, pages 1–17. Springer, 2004.

[5] C. Barthold, K. Subbu, and R. Dantu. Evaluation of gyroscope-embedded mobile phones. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1632–1638, 2011.

[6] E. Berlin and K. Van Laerhoven. Detecting leisure activities with dense motif discovery. In *Proceedings of The 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 250–259, New York, NY, USA, 2012. ACM.

[7] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag Inc., 2006.

[8] U. Blanke and B. Schiele. Remember and transfer what you have learned - recognizing composite activities based on activity spotting. In *International Symposium on Wearable Computers*, 2010.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, Mar. 2003.

[10] A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recogn.*, 30(7):1145–1159, July 1997.

[11] X. Chai and Q. Yang. Multiple-goal recognition from low-level signals. In *Proceedings of The 20th National Conference on Artificial Intelligence - Volume 1*, AAAI '05, pages 3–8. AAAI Press, 2005.

[12] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[13] K.-H. Chang, M. Y. Chen, and J. Canny. Tracking free-weight exercises. In *Proceedings of The International Conference on Ubiquitous Computing*, UbiComp '07, 2007.

[14] 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*, UbiComp '13, pages 355–358, New York, NY, USA, 2013. ACM.

[15] H.-T. Cheng, F.-T. Sun, S. Buthpitiya, and M. Griss. SensOrchestra: Collaborative sensing for symbolic location recognition. In *Mobile Computing, Applications, and Services*, volume 76 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 195–210. Springer Berlin Heidelberg, 2012.

[16] 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, MobiSys '13, pages 361–374, New York, NY, USA, 2013. ACM.

[17] T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. Legrand, A. Rahimi, A. Rea, G. Bordello, B. Hemingway, P. Klasnja, K. Koscher, J. Landay, J. Lester, D. Wyatt, and D. Haehnel. The mobile sensing platform: An embedded activity recognition system. *IEEE Pervasive Computing*, 7(2):32–41, 2008.

[18] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal Ubiquitous Computing*, 10(4):255–268, Mar. 2006.

[19] K. Farrahi and D. Gatica-Perez. Discovering routines from large-scale human locations using probabilistic topic models. *ACM Transactions on Intelligent System and Technology*, 2(1):3:1–3:27, Jan. 2011.

[20] F. Foerster, M. Smeja, and J. Fahrenberg. Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring. *Computers in Human Behavior*, 15(5):571 – 583, 1999.

[21] A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *Advances In Neural Information Processing Systems, NIPS*, 2013.

[22] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong. Attribute learning for understanding unstructured social activity. In *Proceedings of The European Conference on Computer Vision*, pages 530–543, 2012.

[23] M. Galley. A skip-chain conditional random field for ranking meeting utterances by importance. In *Proceedings of The 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 364–372, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[24] Google Inc. Android API Guide: Sensors Overview. Retrieved Novem-

ber 30, 2013, http://developer.android.com/guide/topics/sensors/sensors_overview.html.

[25] Google Inc. Google Glass. Retrieved November 27, 2013, from http://www.google.com/glass/.

[26] 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 *Proceedings of The IEEE International Conference on Pervasive Computing and Communications*, PerCom '09, pages 1–9, Washington, DC, USA, 2009. IEEE Computer Society.

[27] B. Hariharan, S. V. Vishwanathan, and M. Varma. Efficient max-margin multi-label classification with applications to zero-shot learning. *Journal of Machine Learning Research*, 88(1-2):127–155, July 2012.

[28] G. E. Hinton and S. T. Roweis. Stochastic neighbor embedding. In *Advances In Neural Information Processing Systems (NIPS)*, pages 833–840, 2002.

[29] T. Hofmann. Probabilistic latent semantic analysis. In *In Proceedings of Uncertainty in Artificial Intelligence*, UAI '99, pages 289–296, 1999.

[30] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of The 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[31] T. Hospedales, S. Gong, and T. Xiang. Finding rare classes: Active learning with generative and discriminative models. *IEEE Transactions on Knowledge and Data Engineering*, 25(2):374–386, 2013.

[32] D. H. Hu, X.-X. Zhang, J. Yin, V. W. Zheng, and Q. Yang. Abnormal activity recognition based on HDP-HMM models. In *Proceedings of The 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 1715–1720, San Francisco, CA, USA, 2009.

[33] T. Huynh, M. Fritz, and B. Schiele. Discovery of activity patterns using topic models. In *Proceedings of The International Conference on Ubiquitous Computing*, UbiComp '08, pages 10–19, 2008.

[34] W. Iba. Learning to classify observed motor behavior. In *Proceedings of The 12th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI '91, pages 732–738, 1991.

[35] D. Johnson-Cane, J. Glickman, and J. Cane. *The Complete Idiot's Guide to Weight Training*. The Complete Idiot's Guide Series. Alpha Books, 2002.

[36] P. Kankuekul, A. Kawewong, S. Tangruamsub, and O. Hasegawa. Online incremental attribute-based zero-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3657–3664, 2012.

[37] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *Proceedings of The 21st AAAI conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 381–388. AAAI Press, 2006.

[38] E. Kim, S. Helal, and D. Cook. Human activity recognition and pattern discovery. *IEEE Pervasive Computing*, 2010.

[39] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, Dec. 1997.

[40] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[41] K. Laerhoven, H.-W. Gellersen, and Y. Malliaris. Long term activity monitoring with a wearable sensor node. In *International Workshop on Wearable and Implantable Body Sensor Networks*, pages 4 pp.–174, 2006.

[42] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *International Conference on Computer Vision and Pattern Recognition*, 2009.

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

[44] O. Lara and M. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys Tutorials*, 15(3):1192–1209, 2013.

[45] H. Larochelle, D. Erhan, and Y. Bengio. Zero-data learning of new tasks. In *Proceedings of The Conference on on Artificial Intelligence*, AAAI'08, pages 646–651, 2008.

[46] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *Proceedings of The 19th International Joint Conference on Artificial Intelligence*, IJCAI '05, pages 766–772, 2005.

[47] Q. Li, J. Stankovic, M. Hanson, A. Barth, J. Lach, and G. Zhou. Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on*, pages 138–143, 2009.

[48] W. Lin, M.-T. Sun, R. Poovendran, and Z. Zhang. Group event detection with a varying number of group members for video surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(8):1057–1067, 2010.

[49] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *Conference on Computer Vision and Pattern Recognition*, pages 3337–3344, 2011.

[50] R. Liu, T. Chen, and L. Huang. Research on human activity recognition based on active learning. In *International Conference on Machine Learning and Cybernetics*

(ICMLC), volume 1, pages 285–290, 2010.

[51] 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 International Conference on Ubiquitous Computing*, UbiComp '07, pages 483–500, 2007.

[52] B. Longstaff, S. Reddy, and D. Estrin. Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In *International Conference on Pervasive Computing Technologies for Healthcare*, PervasiveHealth'10, pages 1–7, 2010.

[53] H. Lu, W. Pan, N. Lane, T. Choudhury, and A. Campbell. SoundSense: Scalable sound sensing for people-centric applications on mobile phones. In *Proceedings of The International Conference on Mobile Systems, Applications, and Services*, pages 165–178, 2009.

[54] 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 ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 71–84, 2010.

[55] L. Maaten and G. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 2008.

[56] 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 *Proceedings of The International Conference on Pervasive Computing*, pages 246–264. Springer, 2010.

[57] 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, Cambridge, MA, 2007. MIT Press.

[58] L. M. Manevitz and M. Yousef. One-class SVMs for document classification. *Journal*

of Machine Learning Research, 2:139–154, Mar. 2002.

[59] 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 International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 5–20, 2010.

[60] 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 ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 337–350, 2008.

[61] D. Minnen, T. Starner, I. Essa, and C. Isbell. Discovering characteristic actions from on-body sensor data. In *In Proceedings of The IEEE International Symposium on Wearable Computing*, ISWC '06, pages 11–18, 2006.

[62] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1st edition, 1997.

[63] Motorola Mobility. MotoACTV. Retrieved November 27, 2013, from `https://motoactv.com/`.

[64] M. Muehlbauer, G. Bahle, and P. Lukowicz. What can an arm holster worn smart phone do for activity recognition? In *Proceedings of The International Symposium on Wearable Computers*, ISWC '11, pages 79–82, 2011.

[65] K. P. Murphy. *Dynamic Bayesian Networks: Representation, inference and learning*. PhD thesis, University of California, Berkeley, 2002.

[66] S. Nath. ACE: exploiting correlation for energy-efficient and continuous context sensing. In *Proceedings of The International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pages 29–42, 2012.

[67] M. Palatucci. *Thought Recognition: Predicting and Decoding Brain Activity Using the Zero-Shot Learning Model*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 2011.

[68] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell. Zero-shot learning with semantic output codes. In *Proceedings of The Neural Information Processing Systems (NIPS)*, 2009.

[69] D. Parikh and K. Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *Proceedings of The International Conference on Computer Vision and Pattern Recognition*, 2011.

[70] D. Parikh and K. Grauman. Relative attributes. In *IEEE International Conference on Computer Vision (ICCV)*, pages 503–510, 2011.

[71] D. Parikh, A. Kovashka, A. Parkash, and K. Grauman. Relative Attributes for Enhanced Human-Machine Communication. *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[72] N. Ravi, N. D, P. Mysore, and M. L. Littman. Activity recognition from accelerometer data. In *In Proceedings of The Seventeenth Conference on Innovative Applications of Artificial Intelligence*, IAAI '05, pages 1541–1546. AAAI Press, 2005.

[73] R. Rogers, J. Lombardo, Z. Mednieks, and B. Meike. *Android Application Development: Programming with the Google SDK*. O'Reilly Media, Inc., 1st edition, 2009.

[74] S. Rosenthal, A. K. Dey, and M. Veloso. Using decision-theoretic experience sampling to build personalized mobile phone interruption models. In *Proceedings of The International Conference on Pervasive Computing*, Pervasive'11, pages 170–187, Berlin, Heidelberg, 2011. Springer-Verlag.

[75] O. Russakovsky and L. Fei-fei. Attribute learning in largescale datasets. In *Proceedings of The ECCV 2010 Workshop on Parts and Attributes*, 2010.

[76] M. S. Ryoo and J. K. Aggarwal. Recognition of high-level group activities based on activities of individual members. In *Proceedings of The 2008 IEEE Workshop on Motion and Video Computing*, WMVC '08, pages 1–8, Washington, DC, USA, 2008. IEEE Computer Society.

[77] W. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, 2013.

[78] M. Schmidt, K. Murphy, G. Fung, and R. Rosales. Structure learning in random fields for heart motion abnormality detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[79] M. Schuckers. Receiver operating characteristic curve andequal error rate. In *Computational Methods in Biometric Authentication*, Information Science and Statistics, pages 155–204. Springer London, 2010.

[80] B. Settles. Active learning literature survey. In *Computer Sciences Technical Report 1648, University of Wisconsin–Madison*, 2010.

[81] B. Settles. *Active Learning*. Morgan & Claypool, 2012.

[82] R. Socher, M. Ganjoo, C. D. Manning, and A. Y. Ng. Zero-shot learning through cross-modal transfer. In *Proceedings of The Neural Information Processing Systems (NIPS)*, 2013.

[83] M. Stikic, D. Larlus, S. Ebert, and B. Schiele. Weakly supervised recognition of daily life activities with wearable sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

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

[85] H. Storf, M. Becker, and M. Riedl. Rule-based activity recognition framework: Challenges, technique and learning. In *3rd International Conference on Pervasive Computing Technologies for Healthcare*, PervasiveHealth '09, pages 1–7, 2009.

[86] C. Sutton and A. Mccallum. An introduction to conditional random fields for relational learning. In *Introduction to Statistical Relational Learning*, 2007.

[87] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, 2008.

[88] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, 2008.

[89] U.S. Bureau of Labor Statistics. American time use survey activity lexicon. *American Time Use Survey*, 2011.

[90] K. Van Laerhoven, D. Kilian, and B. Schiele. Using rhythm awareness in long-term activity recognition. In *Proceedings of The 2008 12th IEEE International Symposium on Wearable Computers*, ISWC '08, pages 63–66, Washington, DC, USA, 2008. IEEE Computer Society.

[91] S. Wang, W. Pentney, A.-M. Popescu, T. Choudhury, and M. Philipose. Common sense based joint training of human activity recognizers. In *Proceedings of The 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2237–2242, San Francisco, CA, USA, 2007.

[92] 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 International Conference on Mobile Systems, Applications, and Services*, MobiSys '09, pages 179–192, 2009.

[93] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):66–75, Sept. 1991.

[94] P. Wu, J. Zhu, and J. Y. Zhang. MobiSens: A versatile mobile sensing platform for real-world applications. *Journal of Mobile Networks and Applications*, 18(1):60–80, Feb. 2013.

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

[96] Z. Yan, D. Chakraborty, A. Misra, H. Jeung, and K. Aberer. SAMMPLE: Detecting semantic indoor activities in practical settings using locomotive signatures. In *Proceedings of The 2012 16th Annual International Symposium on Wearable Computers*, ISWC '12, pages 37–40, Washington, DC, USA, 2012.

[97] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2011.

[98] J. Yin, Q. Yang, and J. Pan. Sensor-based abnormal human-activity detection. *IEEE Transactions on Knowledge and Data Engineering*, 20(8):1082–1090, 2008.

[99] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Modeling individual and group actions in meetings with layered HMMs. *IEEE Transactions on Multimedia*, 8(3):509–520, 2006.

[100] M. Zhang and A. A. Sawchuk. A feature selection-based framework for human activity recognition using wearable multimodal sensors. In *Proceedings of The 6th International Conference on Body Area Networks*, BodyNets '11, pages 92–98, 2011.

[101] Z. Zhang. Microsoft Kinect sensor and its effect. *IEEE Multimedia*, 19(2):4–10, 2012.

[102] J. Zheng and L. M. Ni. An unsupervised framework for sensing individual and cluster behavior patterns from human mobile data. In *Proceedings of The 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 153–162, New York, NY, USA, 2012. ACM.

[103] V. Zheng, H. Hu, and Q. Yang. Cross-domain activity recognition. In *International Conference on Ubiquitous Computing*, UbiComp '09, 2009.

[104] Y. Zhu, Y. Arase, X. Xie, and Q. Yang. Bayesian nonparametric modeling of user activities. In *Proceedings of The 2011 International Workshop on Trajectory Data Mining and Analysis*, TDMA '11, pages 1–4, New York, NY, USA, 2011. ACM.

[105] A. Zinnen, C. Wojek, and B. Schiele. Multi activity recognition based on bodymodel-derived primitives. In *Location and Context Awareness*, volume 5561 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin Heidelberg, 2009.