

國立臺灣大學電機資訊學院資訊工程研究所

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

以多異質感測器進行日常生活行為
連續辨識之研究

**Continuous Recognition of Daily Activities
from Multiple Heterogeneous Sensors**

The seal of National Taiwan University is a circular emblem. It features a central bell (the 'University Bell') flanked by two traditional Chinese lanterns. The seal is surrounded by the university's name in Chinese characters: '國立臺灣大學' at the top and '勵學敦人' at the bottom.

吳祖佑

Tsu-yu Wu

指導教授：許永真 博士

Advisor: Jane Yung-jen Hsu, Ph.D.

中華民國九十七年六月

June, 2008

Acknowledgments

這份論文的完成，是我兩年的知識累積與努力的成果。我很感激所有給予我知識以及給予我前進力量的每一個人。

感謝我的指導教授許永真老師，一路上給我鼓勵，導正我一些消極的想法，給我足夠的自信心。也很感謝老師提供一個自由的研究環境，讓我能夠朝著自己的興趣發展，老師總是能適時的在學習上給我一些有用的建言。感謝我的父母，讓我能自由的飛翔，讓我在學習上沒有後顧之憂。感謝各位口試委員在論文上給我的指教。

感謝Brooky，若不是你費盡心思蒐集的資料，我就無法完成這份論文。你是一個非常認真的學長，你的態度一直是我效法的對象。

感謝黃老、小陸兩位學長，你們的報告總是非常的精闢，讓我能在專業知識上有所啟發，在討論中也總是讓我受益良多。感謝SY學長、宗翰學長、益庭學長、婉容學姊，在研究的過程中給了我許多寶貴的知識與意見。

感謝實驗室的每一位同學，學長姊及學弟妹，跟大家在一起相處是我能夠前進的原動力。感謝文芝，妳是這兩年來最重要的存在，妳讓我成長了許多。感謝家峻，你的生活方式一直是我欽佩的，你是一位很好的戰友，跟你討論總是能激發一些新的想法。感謝育仲跟啓嘉，你們是很棒的討論對象，不管是分

享生活或是研究。感謝麗徽、嘉涓、怡靜跟Yuhana，研究生活有了你們多了很多很多的歡笑。

感謝在這段過程中一起合作過的伙伴，感謝有家峻、文芝、育仲在AI Project中的努力。感謝家峻跟我齊心完成Paper。感謝好圓、小嫻、武治中、栗子，在期末百忙之中還是願意一起幫忙iGhost Project。感謝好圓跟薇蓉在Wireless Sensor Network課程中與我為每一個實驗奮鬥，我永遠忘不了最後讓車子跑的那個不眠夜晚。感謝有翰文、栗子、啓嘉和我一起面對可惡的Robot Studio、考卷以及作業，特別感謝翰文總是一肩挑起助教大部分的責任。感謝每一位參與行為辨識實驗的受測者。

感謝我的室友BlueCat，小豬及TB，讓我每天離開了實驗室後還是能擁有溫暖，讓我不用費心於住宿或搬家的麻煩問題。感謝育誠，一位很好的戰友，總是可以和你分享一些研究心得。也感謝你幫忙行為辨識的實驗。感謝dregs，一直以來都是很好的聊天打球的對象。感謝鶴凌，DSP課沒有你的幫忙我不可能順利通過。感謝禹安，每次跟你聊天都很愉快，讓人不知不覺就忘了時間。感謝camel，try，大頭，豆叔，德源，butz等大學同學們，你們一直都是我重要的支柱。

感謝我自己的身體，在這樣混亂的作息下還能夠正常的工作，在我把壓力沮喪以及各種負面情緒倒進來的時候，還能夠掙扎的走出來。

太多的人值得感謝，無法一一盡數，感謝命運之神，讓我能夠有這樣的運氣，遇見你們這樣好的人。



Abstract

Recognition of daily activities is an enabling technology for active service providing and automatic in-home monitoring. In this thesis, we aim to recognize activities in a long sensor stream without knowing the boundary of activities. We formulate this continuous recognition problem as a sequence labeling problem. The activity is labeled every a fixed interval given the sensor readings.

Fusing multiple heterogeneous sensors helps disambiguate different activities. However, these sensors are very diverse in readings. To evaluate the capability of models in dealing with such diverse sensors, we compare several state-of-the-art sequence labeling algorithms including hidden Markov model (HMM), linear-chain conditional random field (LCRF) and SVM^{hmm}. The results show that the two discriminative models, LCRF and SVM^{hmm}, significantly outperform HMM. SVM^{hmm} show robustness in dealing with all sensors we used. By incorporating proper overlapping features, the accuracy can be further improved. In additions, CRF and SVM^{hmm} perform comparably with these overlapping features.

For active service providing, we evaluate various inference strategies for the on-line recognition problem. On-line Viterbi algorithm achieves highest frame accuracy but suffers from high insertion errors that may cause unexpected services. We propose smooth on-line Viterbi algorithm to solve this problem.

Keywords: Activity Recognition, Heterogeneous Sensors, Continuous Recognition, Hidden Markov Model, Conditional Random Field, Structural Support Vector Machine



摘要

日常生活行為辨識是用來達到主動服務以及自動監控的一項關鍵科技。我們希望能從一段包含未知行為的感測器資訊中，連續辨識出發生的行為與時間。在這個論文中，我們透過感測器序列的資訊，不斷的去判斷出每一個時間點發生的行為來達到連續辨識的目的。

透過混合多種異質的感測器有助於我們分辨出各種行為，然而，異質感測器在資訊呈現形式上往往有很大的差別。我們希望能發掘不同模型在整合這些資訊時的特性，在我們的研究中比較了不同的模型在這個問題上的適用度，包括隱藏馬可夫模型(Hidden Markov Model)，條件隨機場(Conditional Random Field)，以及結構式支持向量機(Structural Support Vector Machine)。

實驗結果說明，鑑別式模型如條件隨機場，以及結構式支持向量機對於整合感測器較為有效，其準確度明顯高於隱藏馬可夫模型。其中結構式支持向量機對於各種不同形式的感測器都能擁有相當好的結果。除此之外，我們引入了數種重疊特徵提取的方法，使用這些特徵值能夠進一步的改善準確度，在使用的這些特徵後，條件隨機場跟結構式支持向量機得到了相當接近的準確度。

為了提供主動的服務，我們比較了數種不同的即時辨識方法。在我們所比較的方法中，On-line Viterbi得到了最佳的單位時間準確度，然而卻會產生相當

多不必要的服務。我們提出了Smooth On-line Viterbi方法來改善這種情形。

關鍵詞：行為辨識、異質感測器、連續辨識、隱藏馬可夫模型、條件隨機場、結構式支持向量機



Contents

Acknowledgments	ii
Abstract	v
List of Figures	xiii
List of Tables	xiv
Chapter 1 Introduction	1
1.1 Motivation	1
1.2 Research Objective	3
1.3 Thesis Organization	5
Chapter 2 Related Work	7
2.1 Sensor Setting	7
2.1.1 Sensor Selection	7
2.1.2 Multiple Heterogeneous Sensors	12



2.1.3	Sensor Placement	13
2.2	Classification Algorithms	13
2.2.1	Feature Extraction	14
2.2.2	Classifiers	14
2.2.3	Generative Modeling	16
2.2.4	Sequence Segmentation	17
2.3	Sequence Models	19
2.3.1	Hidden Markov Model	19
2.3.2	Dynamic Bayesian Network	20
2.3.3	Maximum Entropy Markov Model and Conditional Random Field	21
2.3.4	Structural SVM	22
Chapter 3	Off-line Recognition for Monitoring	25
3.1	Problem Definition	25
3.2	E-Home Dataset	26
3.3	Activity Modeling	27
3.3.1	HMM	27
3.3.2	Linear Chain CRF	28
3.3.3	SVM ^{hmm}	30
3.3.4	Other Approaches	31
3.4	Performance Measures	31

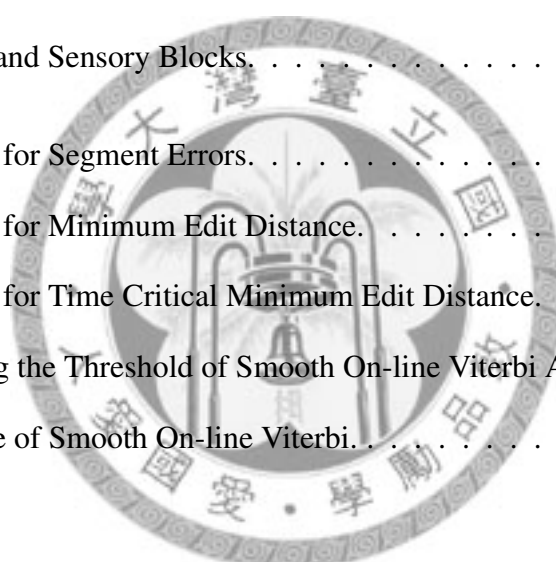
3.5	Raw Features	32
3.5.1	Results	33
3.6	Overlapping Features	35
3.6.1	Generative Audio Probabilities	35
3.6.2	Region and Region Transitions	36
3.6.3	NextRFID and LastRFID	37
3.6.4	Results	38
Chapter 4 On-line Recognition for Active Services		41
4.1	Problem Definition	41
4.2	Dynamic Programming Algorithms	42
4.2.1	On-line Viterbi Algorithm	42
4.2.2	Bayes Filtering	43
4.2.3	Token Passing Algorithm	43
4.3	Evaluation	44
Chapter 5 Segment Analysis		45
5.1	Segment Error	46
5.1.1	Minimum Edit Distance	46
5.1.2	Time Critical Minimum Edit Distance	48
5.2	Evaluation	50
5.2.1	Off-line Recognition	50
5.2.2	On-line Recognition	50

5.3	Smooth on-line Viterbi	51
5.3.1	Evaluation	52
Chapter 6	Conclusion	57
Bibliography		60



List of Figures

3.1	Generative Audio Probabilities.	36
3.2	Regions and Sensory Blocks.	37
5.1	Example for Segment Errors.	46
5.2	Example for Minimum Edit Distance.	47
5.3	Example for Time Critical Minimum Edit Distance.	49
5.4	Searching the Threshold of Smooth On-line Viterbi Algorithm.	52
5.5	PR Curve of Smooth On-line Viterbi.	53



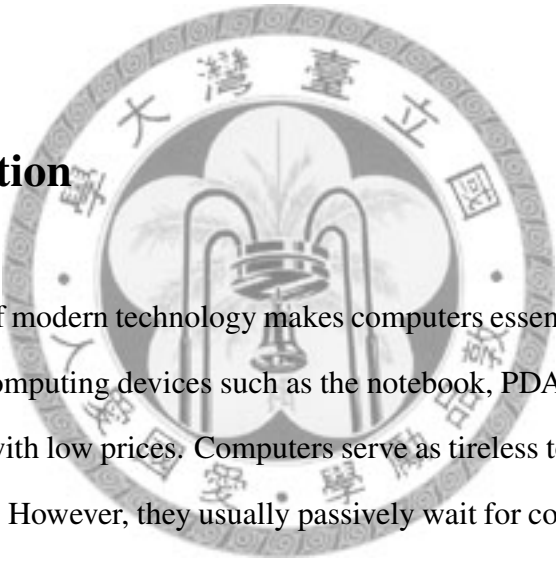
List of Tables

3.1	Performance Comparison of HMM, LCRF and SVM ^{hmm} using Raw Features.	33
3.2	Performance Comparison of Frame-Based Classification and Sequence Models.	34
3.3	Performance Comparison of LCRF and SVM ^{hmm} Using Raw Features and Overlapping Features.	38
4.1	Performance Comparison of Viterbi, On-line Viterbi, Bayes Filtering and Token Passing Algorithms.	44
5.1	Segment Error of LCRF and SVM ^{hmm}	50
5.2	Segment Error of On-line Algorithms.	51

Chapter 1

Introduction

1.1 Motivation



The development of modern technology makes computers essential in our living world. Various forms of computing devices such as the notebook, PDA, cell phone, and desktop are accessible with low prices. Computers serve as tireless tools and are ubiquitous in the environment. However, they usually passively wait for commands. Can computers become more intelligent to know our need and provide services actively?

If the computer understands what we are doing, it is able to provide helpful services automatically. For example, we may have the experience that we fall into an unconscious sleep after the hard working of a day. A considerate computer system should understand this situation and turn off lights. Sensor technology grants computers the ability to sense the world. However, sensing is far from understanding. Sensor data are usually noisy and unstructured. Activity recognition is key to bridge the ambiguous

and noisy sensor data to meaningful activities that we care.

The aim of the activity recognition is widespread.

Physical activities such as exercises and sports are important for keeping us healthy. In additions, exercises need to be done carefully. Wrong postures in a fierce exercise may induce severe injury. Monitoring physical activities is helpful for a coach or a doctor to keep their customers in correct progress. Examples for recognizing exercises involve tracking free-weight exercise [5] and rehabilitative exercise [28].

Accidents ranks five in the ten leading causes of death in 2005 in Taiwan. Detecting abnormal activities enables the system to be aware of the emergency. For example, with the video surveillance system, we are able to identify car accidents automatically. Accidents in the home living should also be carefully monitored. When we rolls on the floor or slipping in a bathroom, it would be dangerous if we are alone and nobody is aware. The detection of abnormal activities is thus of great help for saving lives.

Social network researchers may be favored by an activity recognition system that recognizes the conversations and social events between people. For example, knowing the conversations between participants in a conference can be utilized to construct a social network [8].

Daily activity is one of the most important activities we want to recognize. *Medical professionals believe that one of the best way to detect an emerging medical condition before it becomes critical is to look for changes in the activities of daily living (ADLs) [13], instrumental ADLs (IADLs) [17], and enhanced ADLs (EADLs) [33]. ADLs are “the things we normally do in daily living including any daily activity we perform for self-care such as feeding ourselves, bathing, dressing, grooming, work, homemaking,*

and leisure”. “The ability or inability to perform ADLs can be used as a very practical measure of ability and disability in many disorders”.¹ Collecting information about ADLs is thus important for the health assessment. However, self-reporting of ADLs may not be reliable due to the dishonesty or inability of the subjects. With a computational system that keeps tracking daily activities of human, it would be easier for family members, care-givers or physicians to identify the potential health problem at home.

Recognizing daily activities also enables the computer to be an active service provider. An intelligent housekeeper can control the lighting, heating, air conditioning, and noisy level to adapt the environment to us. A personal assistant can remind us the schedule or medication time. A consultant can show us what to notice when we are doing the housework or cooking. With the recognition of daily activities, our computer is able to play the role of an intelligent housekeeper, a personal assistant, and a consultant to provide us prompt help at the right time and at the right place.

1.2 Research Objective

Our goal is to recognize when and which activities occur given a sensor trace. Daily activities are performed sequentially. As a result, no explicit cue is available in the trace for us to know the boundary of the activities. In additions, durations of activities can be as long as one hour and as short as twenty seconds. The task is thus challenging.

Different daily activities involve similar patterns in some aspects. For example,

¹Definition in MedicineNet.com

preparing meals, having meals and washing dishes involve touches of the dishes. Watching TV has similar audio pattern with listening to music since the TV program plays music as well. The bed is usually a place for sleeping but can also be a place for reading. We can sit on the same chair for both reading and playing computers. It seems impossible to use single kind of sensor to distinguish daily activities well due to the ambiguity. As a result, we favor utilizing multiple heterogeneous sensors for the recognition system. However, heterogeneous sensors are diverse in readings. We need to properly integrate the information of all sensors.

Different people have different ways of executing activities. For example, someone may take a bath for an hour while another completes it in three minutes. The preferences is also different. One may favor drinking the juice while another may favor the coffee. The sequential orders of activities or sub-activities are also subject to individuals. In cooking dishes, one may put condiments first while another may put ingredients first. A recognition system would be more practical when it is able to handle the individual difference.

We want to build a robust model that is able to properly deal with different forms of sensor readings as well as fuse them. It should also be robust to learn the variation of different individuals.

In this thesis, we utilize state-of-the-art sequence models to recognize daily activities from multiple heterogeneous sensors including the microphone, the location system and the RFID system. We discuss several issues including the model comparison, overlapping features, on-line recognition and segment errors.

1.3 Thesis Organization

The thesis is organized as follows: In chapter 2, we survey existing approaches for both sensor usages and recognition algorithms. In chapter 3, we compare and discuss various models in solving the off-line recognition problem. We also propose strategies for extracting useful overlapping features. In chapter 4, we evaluate several inference algorithms to handle the on-line recognition problem. In chapter 5, we analyze the segment error using the edit distance and propose a smooth algorithm to reduce the high insertion errors of the on-line algorithm. The conclusion and future research direction are presented in chapter 6.

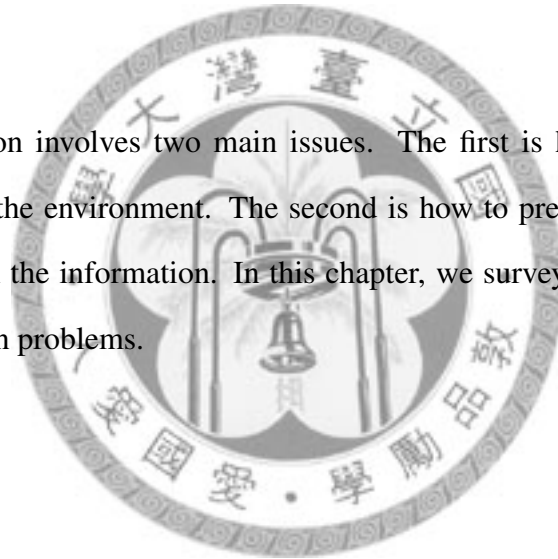




Chapter 2

Related Work

Activity recognition involves two main issues. The first is how to collect relevant information from the environment. The second is how to predict what occurs in the environment given the information. In this chapter, we survey various approaches to activity recognition problems.



2.1 Sensor Setting

2.1.1 Sensor Selection

It is important to identify what information is crucial for the recognition of target activities. In this way, we can select proper sensors for acquiring information.

Vision

Vision is one of the major sensory component for human. We easily recognize what one is doing in a complex scene with our eyes. Therefore, we believe visual information contains most clues about the environment. With the computer vision, although it is not as powerful as our eye in the current development, the activities can be recognized automatically with cameras. Wada et al. [42] conduct an experiment for detecting entering and exiting the door using multiple cameras in different views. Video clips are widely used for the recognition of motions. Sminchisescu et al. [35] recognize different motions such as walking, running, bending, picking, and dancing with the silhouettes.

However, the privacy issue is a major concern. In Wilson's report [45], the camera is less acceptable for the user. In his report, 11% of participants never accept the camera in the home while 31% of them hesitate. None of them definitely accepts the usage of the camera.

In additions, the visual information is quite sensitive to the variation of the environment such as the change of lighting conditions. The processing of 2-D images or 3-D videos is usually computational costly. The storage requirement of the visual data is also high.

Audio

Hearing is another important sensory component for human. We use speech as our main communication median. Doing activities sometimes generates distinguishable

sounds for the recognition. For example, Chen et al. [6] use a microphone to recognize bathroom activities such as showering, urination, flushing, washing hands, and sighing. The operation of machines generates unusual sounds. Ward [44] recognizes activities in the wood workshop with the microphone worn on the right hand. The results show that the audio information is of great help for the detection of the usage of the hammer, drill, grind, and drawer. Daily activities can be recognized via audio. Lopes et al. [24] recognize different sounds generated by the car, door, train, clapping, stepping, and talking.

The privacy issue is also a concern for the microphone. In Wilson's report [45], 17% of participants never accept the microphone while 43% hesitate. None of them definitely accept the usage of the microphone as well.

Human Object Interaction

Activities involve the interaction with objects. For example, we drink with a cup and have meals with dishes. We cook with a stove and wash clothes with a washing machine. Sensors attached on objects inform the recognition system how people are interacting with these objects. Current sensors can be used to detect the usage of electrical devices such as the microwave and the refrigerator. Flow sensor can be used to detect the usage of water faucet. Switch sensors can be used to detect opening and closing the door and window. To detect the usage of mobile objects such as cups, attaching them accelerometers or tilt sensors to sensor the movement is a possible choice.

Radio-frequency identification (RFID) is another choice for detecting the human object interaction. An RFID system consists of readers and tags. When a reader is

close to a tag, the reader senses which tag is nearby. In this way, it informs the system the interaction between the reader and the tag. Passive RFID tags can be used without the internal power supply. In additions, they can be small enough to be attached on most of the objects. Patterson et al. [29] use an RFID glove to detect activities which involve the interaction with objects such as cleaning the table, eating breakfast, taking out trash and making tea. Work similar RFID usages is done by Lin [21], Wyatt et al. [49], and Pentney et al. [31].

Location

Location can be used to recognize activities since some activity is usually done in some specific place. For example, we work in the office and sleep at home. Global positioning system (GPS) is a popular tool for the outdoor localization. Although the weather can affect the accuracy, it is still quite accurate. Researchers in Washington university contribute much effort to the activity recognition using GPS. Patterson et al. [30] track transportation modes such as foot, bus, and car using the GPS readings. Liao et al. [18] recognize the high level activity such as at home, at work, shopping, dining out, visiting by the inferring different locations. For the indoor localization, existing approaches include using ultrasound, infrared, and radio-frequency signal.

Wilson [46] suggests a formulation that recognizes the location and activity simultaneously. We know that the activity and location may constrain each other. If we know someone is cooking, it is highly probable that he is in the kitchen. On the other hands, if we know someone is in the bedroom, it is more likely he is sleeping. His work utilizes motion detectors, contact switches, pressure mats, and break beams to

track people. Unlike GPS and other indoor location systems, this kind of sensors is anonymous such that you are not able to directly know the association of the sensor trigger and the person. The association problem is difficult when there are many people in the same places.

Physical and Physiological Information

The physical status is useful for the activity recognition, particularly for the recognition of exercises. Exercising involves frequent and repetitive movement. The information can be acquired with accelerometers, pedometers or force sensors. As we mention previously, Pan [28] recognizes rehabilitative exercises such as shoulder rolling, Pectoralis stretch and front raise with accelerometers attached on the elbow, wrist and shoulder. Chang et al. [5] recognize free-weight exercises with accelerometers attached on the back of the hand and on the belt.

Actions can also be recognized by the accelerometers since they also involve frequent and repetitive movement. Ravi et al. [32] recognize activities such as standing, walking, running with one accelerometer attached near the pelvic region. Maurer et al. [25] recognize running, sitting, standing, and walking effectively with accelerometers attached on 6 positions.

The other high-level activities can be recognized by identifying the posture and action. For example, we usually lie when sleeping and sit when playing a computer. Bao et al. attach 5 accelerometers on the hip, wrist, arm, ankle and thigh to detect activities such as working on a computer, folding laundry, bicycling and reading.

When the motion is slow or in constant speed, acceleration may be dominated by

noise. It is not easy to track the real trajectory of our limbs using accelerometers only. As an alternative, the ultrasound location system tracks the targets accurately in a 3D space. Stiefmeier et al. [37] utilize both accelerometers and the ultrasound location system to recognize activities of repairing the bicycle.

The change of activities may affect the physiological status. Our breath may become more frequent when we are nervous in a game. Physiological data modeling contest (PDMC) [1] is a contest for the activity recognition. The dataset are collected using both physical sensors such as the pedometer and accelerometer and physiological signals such as galvanic skin reflex (GSR), skin temperature and heat flux. In the contest, participants are asked to predict two different activities, watching TV and sleeping. The results show that they are highly recognizable with the physical and physiological sensors.

2.1.2 Multiple Heterogeneous Sensors

Since the sensor information is directly related to what activity we care, by combining multiple heterogeneous sensors, we are able to gather information to disambiguate different activities.

Yen [50] shows the accuracy of the activity recognition can be improved by fusing multiple heterogeneous sensors. However, he models this problem in a simplified scenario that is to recognize activities in a set of manual segmented observation sequence. This work is an extension by relaxing the need of manual segmentation that is usually not feasible in real applications.

2.1.3 Sensor Placement

The placement of sensors is an important issue. Sensors can be worn on human body or ubiquitous in the environment. Wearable sensors excel at the low equipping cost and the scalability. However, it is not natural for us to wear something like the wrist support or the badge all day long.

On the contrary, ubiquitous sensors can be attached on objects and invisible to the user. House.n [11] is such a realization that they attach objects current sensors, flow sensors, light sensors, switch sensors and accelerometers. The disadvantage is that we need to spend much effort arranging sensors and keep them alive. If sensors lost their power supply or suddenly crash, how can we detect it and recover the sensors in an acceptable cost becomes a big problem. In additions, the range of the recognition is limited by the scale of the sensors.

Some sensors are flexible to be used in different settings. For example, the microphone can be either placed in the room or on one's hand. RFID can be used in multiple schemes. We can wear an RFID reader and tag objects that we are interested in. We can put tags and readers both in the environment and detect the displacement of the objects. We can put the reader in the environment and wear a tag as well.

2.2 Classification Algorithms

Machine learning approach is key to the activity recognition problem. There is a semantic gap between the target activity and the sensor readings so that it is not easy to encode the mapping by a simple rule-based system. Machine learning serves as a tool

that automatically identify the relationship between the sensor readings and the activities by learning from the training data. In additions, sensors usually involve noise. The modern models can take the noise into account and provide an accurate and robust prediction.

2.2.1 Feature Extraction

We need preprocessing the raw sensor data for filtering out the noise and aggregating information. Features are extracted materials from the raw sensor readings. Although the feature extraction is domain dependent, ideas in many other fields can be applied to the activity recognition problems. For example, the mean, variance, correlation, and entropy in statistics and information theory are often used as the features for signal-based sensors such as accelerometers. Mel frequency cepstral coefficients (MFCCs) for speech recognition can also be used as the features in the activity recognition problem. For the discrete sensor event such as the RFID reading, the order of the sensor events can be considered using specialized temporal features. For example, in Tapia's [40] work, the order of the object usage is encoded as the 'Before Feature' which denotes one sensor is triggered before another sensor. With proper features, we can make the recognition better.

2.2.2 Classifiers

A classifier is the bridge between noisy and unstructured features and target classes. Given an unlabeled instance, a classifier maps the features to a predicted class. Due

to the popularity of the machine learning, we now have an amount of classifiers and widely available tools. Thus, we can easily formulate the activity recognition task as a classification problem and solve it with existing classifiers. We introduce some popular classifiers in the following paragraph.

Decision tree (DT) is a tree-structured model. The classification process goes through the tree and stop at a leaf node. In each internal node, we check feature values with a certain condition and branch to a child according to the condition. In the leaf node, we determine a class that is associated with the leaf node. The decision tree can be converted to a set of rules and easily understood by human.

Statistical approach is popular in dealing with noisy data. For example, Naïve Bayes (NB) classifier models the joint probability of the features and the class label. Conditional independence is assumed for every feature given the class label. As a result, the joint probability is factorized to multiplication of simple distributions. The classification process selects the class with the maximum likelihood given the features. The distributions can be easily on-line updated when new training data come. DT and NB are already widely used in activity recognition problems [3] [40] [25] [44] [23] [39].

The classification process in k-Nearest Neighborhood (kNN) simply computes the distances in the feature space between the testing instance and the training instances. The predicted class is determined by the labels of k nearest training instances. No training process is needed but high computational and storage cost is therefore unavoidable, particularly when the training data are large. Lopes et al. [24] use kNN as their implementation for the audio recognition problem.

Support vector machine (SVM) learns a decision hyperplane in the feature space. It is originally designed for the binary classification, but the extension to multiple classes is available. One may directly train a multi-class SVM or a set of binary SVMs for the multi-class classification problem. Huynh et al. [10] make a comparison of utilizing kNN and SVM for the recognition of both low-level and high-level activities. In their work, SVM outperforms kNN.

To avoid overfitting, ensemble of multiple weak classifiers is a possible choice. For example, different combinations of base classifiers such as DT, NB, kNN and SVM and ensemble strategies such as boosting, bagging, plurality voting and stacking are compared in Ravi's work [32]. They show plurality voting and boosted SVM achieve highest accuracy.

2.2.3 Generative Modeling

A generative model captures the joint probability of the observed random variables and the hidden random variables. Since the joint probability is modeled, we can infer any joint and conditional probabilities of any random variables. For example, if we model the joint probability of the humidity, temperature and raining, it is possible to ask how possible the temperature is high if we know it is raining or how possible it is raining if the temperature and the humidity are high.

Bayesian network (BN) is a generative model that allows flexible factorization of the joint probability of different random variables. Naïve Bayes classifier is one simple instantiation of BN. To model the real value random variable, Gaussian mixture model (GMM) is a possible choice. In a GMM, the distribution is assumed to be a mixture of

Gaussian distributions. For example, Patterson et al. [30] model the speed of moving using 4 mixtures of Gaussian distributions since simple Gaussian distribution may not be able to describe the moving speed of 3 different transportation modes. However, it is usually difficult to observe which mixture each training instance belongs to. Fortunately, we can automatically learn it by maximizing the likelihood of the training data using EM algorithm.

Sometimes the input is a series of features that comes sequentially. Hidden Markov model (HMM) is a generative model for capturing the temporal relationship in the sequence. For each frame of the feature sequence, it introduces a single latent random variable with multiple states to represent the underlying multiple stages of the sequence. The latent random variables are assumed to be conditionally independent to early frames given the previous frame. This is known as Markov assumption. Given the latent random variable for a frame, the features in the same frame is assumed conditionally independent to any other random variables in other frames. These two assumptions result in a very good factorization of the joint probability. Thus, there exists an efficient way for inferring the generative probability of the observation. Similar to GMM, we can automatically learn an HMM by maximizing the likelihood of the training data.

2.2.4 Sequence Segmentation

Activities occur continuously without explicit cues. In this way, we need to segment a small sequence from the long observation for the classification algorithm.

Sliding Window

Windowing is a commonly used techniques in the signal processing. Since we do not know the activity boundary for a continuous sensor trace, we assume the useful information reveals in a finite length of window. We slide the window and segment a short sequence for the classification. Thus, we can easily reduce the complex continuous recognition problem into a classification problem. This approach is widely used in previous work [3] [40] [23] [10]. Tapia et al. [40] propose the window length should be variant according to the duration of each activity.

However, windowing results in unavoidable noise at the activity boundary. In additions, there is no guarantee what window length involves sufficient information for classification. Longer window involves more information but suffers from more noise at the boundary. In Huynh's [10] work, they select best window length by search. But the search requires much more computation.

Dynamic Programming Search

Dynamically determining the activity boundary is probable by defining a search criteria over the whole observation sequence. In this way, the problem of the boundary noise and insufficient information in sliding window approach is not presented. Dynamic programming search is widely used in large-vocabulary continuous speech recognition. The searching criteria can be determined according to the probability estimations of generative models and the transition probabilities by the language model. Although the search space is exponentially large, we can use the dynamically programming tech-

nique to find optimal or sub-optimal solutions efficiently. For example, In HTK [47] which refers to HMM toolkit, token passing algorithm is used to recursively search the best next state in multiple HMMs where each HMM models the generative probability of the phone or the word in speech. The best sequence is thus determined for a given audio sequence.

2.3 Sequence Models

Similar to the classification problem, sequence labeling model seeks a mapping from the input to the output. However, sequence labeling problem differs from the classification problem in that the input and output are sequences. Therefore, the input space and output space are exponentially large. Consider a noun-phrase chunk problem that is to label each word an IOB tag in a natural language sentence. With only 3 classes and 20 words, there will be 3^{20} possible sequences for the output space. This kind of problem exists in many popular fields including natural language processing, computer vision and bioinformatics. Researchers propose different models that represent the input and output space with a structure. Here we introduce some of the state-of-the-art sequence labeling models.

2.3.1 Hidden Markov Model

Although HMM is a generative model, it is widely used in such a discrimination problem. By defining the output sequence as the hidden random variable sequence, the states of the random variable are corresponding to labels of the output sequence. The

state sequence that has highest joint probability with the observation is determined as the output sequence. Wyatt et al. [49] use automatically mined common sense to build an HMM as their model for inferring activities from an RFID sequence. Patterson et al. [29] compare three different formulations using HMM.

2.3.2 Dynamic Bayesian Network

Dynamic Bayesian network (DBN) is a more general model that extends BN to represent the temporal relationship between time slices. HMM and many of its extensions are specialization of DBN. Rather than a single latent random variable every frame in HMM, DBN allows complex dependency structure of different random variables in the time frame and between time frames. Patterson et al. [30] and Liao et al. [18] propose different DBNs that model the complex dependency of the transportation mode, speed, location and GPS readings. Consider modeling this kind of relationship in an HMM, there can be exponentially large states and too many conditional probabilities to be learned. Some filtering algorithm such as Rao-Blackwellized particle filter (RBPF) favors this kind of factored structured since it allows different filtering strategies for different parts. Wilson [46] propose an RBPF that estimates the association using the particle filter but estimates the distribution of the location and activity using Bayes filter.

HMM implicitly models the duration of each class using the geometric distribution which may not be appropriate. Hidden semi-Markov Model (HSMM) is a kind of DBN that add a duration node or an end node which models the distribution of staying in the same state and transiting to other states. Another limit in HMM is that the transi-

tion probabilities remains the same for every time frame. Hierarchical hidden Markov model (HHMM) is another kind of DBN that represents the transition probabilities using a hierarchy of HMMs such that different transition probabilities are modeled in different time scales. Duong et al. [9] introduce the Switching Hidden Semi-Markov Model (S-HSMM) that use a two-layer hierarchy of HSMMs to model the complex relationship of high level activity that is the composite of atomic actions.

Although DBN is powerful for modeling complex relationship of multiple random variables, it usually pay for more computational effort. Exact inference algorithm such as junction tree algorithm [36] needs exponential inference time if the structure is too complex in DBN. Approximation techniques such as Gibbs sampling and loopy belief propagation try to solve this problem.

2.3.3 Maximum Entropy Markov Model and Conditional Random Field

Generative models select the sequence with maximum likelihood of observation. However, sequence labeling is a discrimination problem that is to predict the sequence given the observation. Directly modeling the conditional probabilities of the label sequence given the observation seems more natural for this problem. For example, maximum entropy Markov model (MEMM) [26] is a conditional model that uses similar structure of HMM except the relationship of the observation is inverted. In this way, we can model dependent observation such as overlapping features and long-term observation without making improper independence assumption. However, MEMM suffers from

the label bias problem due to the per-state normalization.

Conditional random field [16] (CRF) achieves great success in this field by solving the label bias problem in MEMM using the global normalization. CRF has also been applied in activity recognition. Chieu et al. [7] and Sminchisescu et al. [35] utilize a linear-chain CRF (LCRF) for the activity recognition problem and show the superiority over HMM. Liao et al. propose using a hierarchical CRF and iteratively inferring activities and important locations simultaneously [19]. Shimosaka et al. [34] and we [48] propose using a factored structure of CRF for solving the multi-tasking activity recognition. Benson et al. [20] propose a CRF-Filter that is adapted from the particle filter to solve the on-line recognition problem in localization.

2.3.4 Structural SVM

SVM is an effective approach for the classification problem. Altun et al. [2] propose an extension of SVM, hidden markov support vector machine, that handles the sequence labeling problem. A general model for arbitrary output space, structural SVM is proposed by Tsochantaridis et al. [41]. In an experiment conducted by Nguyen and Guo [27], structural SVM outperforms 5 other state-of-the-art models including HMM, CRF, Averaged perceptron (AP), Maximum margin Markov networks(M^3N), and an integration of search and learning algorithm (SEARN) in two sequence labeling tasks, part-of-speech (POS) tagging and optical character recognition (OCR). The experiments show the superiority of structural SVM in two problems. But in a later technical report [14], CRF is shown to be comparable with structural SVM when appropriate features are used. In this thesis, we also address this issue with the activity

recognition problem.

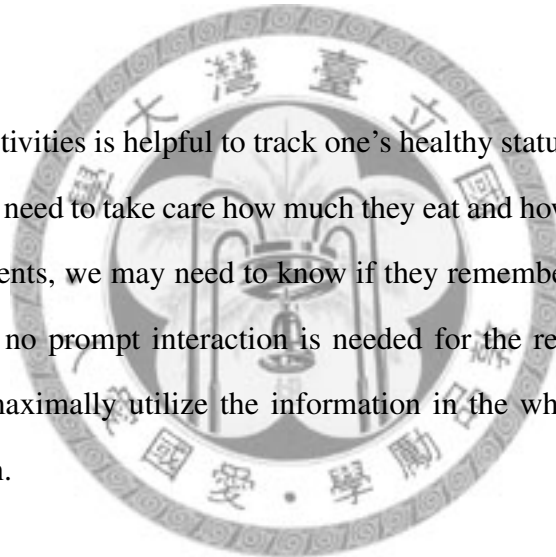




Chapter 3

Off-line Recognition for Monitoring

Analyzing daily activities is helpful to track one's healthy status and living ability. For the elders, we may need to take care how much they eat and how often they sleep every week. For the patients, we may need to know if they remember to take medicines. In such applications, no prompt interaction is needed for the recognition system. The main issue is to maximally utilize the information in the whole observation for the accurate prediction.



3.1 Problem Definition

To monitor daily activities automatically, we collect a long trace of sensor data from the subject. In the trace, we do not know how many activities occur as well as the duration of each activity. The problem is difficult since there are a large number of possible hypotheses. We formulate this problem as a sequence labeling problem.

Sequence labeling problem is to assign a single label to each element in an observation sequence. In our problem, the activity is labeled every a fixed interval given the sensor readings.

We define the problem as follow.

Given an observation sequence $O = (O_1, O_2, \dots, O_T)$ where O_t is the feature vector of readings from multiple heterogeneous sensors at time t , the goal is to predict an activity sequence $A = (A_1, A_2, \dots, A_T)$ where A_t is the activity occurs at time t . A_t belongs to a set of N target activities $T = \{t1, t2, \dots, tN\}$.

3.2 E-Home Dataset

We use E-Home dataset as the evaluation dataset. E-Home dataset is collected in a home-like environment for the research of the activity recognition by Yen [50]. The dataset involves 13 subjects and each performs 12 activities including "listening to music", "watching TV", "reading", "telephoning", "preparing meals", "having meals", "drinking", "resting", "taking medicines", "mopping", "taking out of the garbage" and "using the computer". The order of activities is set as random and the parts of reading experimental instructions in the trace is manually eliminated. The dataset consists of totally 27818 seconds and the activity is annotated every second. In the dataset, three primary kinds of sensors including a microphone in the corner of the room, a wearable RFID reader and 40 load sensory blocks on the floor are used.

For the audio stream from the microphone, 24-dimensional real value feature vector including the mean of volume, variance of volume, low short time energy rate

(LSTER), mean of zero-crossing rate, variance of zero-crossing rate, range of zero-crossing rate, high zero-crossing rate ratio (HZCRR), mean of the spectrum flux (SF), mean of band energy, variance of band energy, mean of band energy ratio, and variance of band energy ratio, are extracted. A location system estimates the most active block of the 40 load sensory blocks by filtering out the deformation noise. The RFID reader returns null or one of the 24 tagged objects including "book1", "book2", "CD1", "CD2", "CD player", "computer keyboard", "cup1", "cup2", "dish", "garbage bag", "garbage bag2", "juice", "juice2", "microwave oven", "mop", "plastic spoon", "refrigerator", "TV remote control", "teabag box", "teakettle", "telephone", "trash can", "vitamin and water boiler". The features are extracted every second. The detail of the experiment and feature extraction can be found in Yen's thesis [50].

3.3 Activity Modeling

Since we model the recognition problem as a sequence labeling problem, existing sequence models can be used to recognize activities. We compare state-of-the-art models including HMM, linear chain CRF and SVM^{hmm} in this problem.

3.3.1 HMM

Each activity is modeled as one state of the hidden node in HMM. Thus, HMM models the joint probability $P(A, O)$ of the activity sequence A and the observation sequence O with the initial probabilities of activities, the conditional probabilities of the transitions between activities and the conditional probabilities between activities and the

sensor observation. Each dimension of the observation is assumed conditionally independent of each other given the activity. The conditional probability of each discrete feature given the activity is modeled as a multinomial distribution; the conditional probability of each real value feature given the activity is modeled as a mixture of 5 Gaussian distributions.

We use the Graphical Modeling Toolkit (GMTK) [4] for the implementation of HMM. We smooth each multinomial distribution using a weighted sum of the learned distribution and a uniform distribution. The weight w of the uniform distribution is used to control the smoothness of parameters in HMM.

3.3.2 Linear Chain CRF

We use a linear chain CRF (LCRF) [16] for the recognition problem. LCRF is an instantiation of CRF that uses similar graphical structure with HMM. Tutorials can be found in the introduction by Wallach [43] and Sutton [38].

CRF models the conditional probability $P(A|O)$ by a set of feature functions $F = \{f_1(A, O, t), f_2(A, O, t), \dots, f_J(A, O, t)\}$ and a weight vector $W = \{w_1, w_2, \dots, w_J\}$. The conditional probability $P(A|O)$ is defined as

$$P(A|O) = \frac{\exp(\sum_j \sum_t w_j f_j(A, O, t))}{Z(O)}$$

where $Z(O)$ is the normalization constant.

Here we use unigram feature functions between the activity label and a feature in the same frame. For the discrete feature, a set of binary feature functions is defined for every combination of the activity and the feature value. For example, we define a

binary feature function for the activity "reading" and the RFID reading "book1" as

$$f_{reading,book1}(A, O, t) = \begin{cases} 1, & \text{if } A_t = \text{"reading"} \text{ and } O_t^{RFID} = \text{"book1"}. \\ 0, & \text{otherwise.} \end{cases}$$

where O_t^{RFID} is the RFID reading at time t .

For the real value feature, a set of real value feature functions is defined for every activity. The value of the feature functions is defined as the feature value. For example, we define a feature function for the activity "reading" and the mean of volume as

$$f_{reading,meanVolume}(A, O, t) = \begin{cases} x, & \text{if } A_t = \text{"reading"} \text{ and } O_t^{meanVolume} = x \\ 0, & \text{otherwise.} \end{cases}$$

where $O_t^{meanVolume}$ is the mean of volume at time t .

For modeling the temporal relationship, we use bigram feature functions between adjacent activities. A set of binary feature functions is defined for every combination of the activities. For example, we define a binary feature function for the consecutive "reading" activities as

$$f_{reading,reading}(A, O, t) = \begin{cases} 1, & \text{if } A_{t-1} = \text{"reading"} \text{ and } A_t = \text{"reading"}. \\ 0, & \text{otherwise.} \end{cases}$$

We choose CRF++ [15] for the implementation of LCRF. CRF++ is an open source package which allows flexible definition of the feature functions. CRF++ is originally designed for discrete features. We extend CRF++ to handle the real value features. CRF++ uses forward/backward algorithms for computing the marginal probabilities and the normalization constant. We predict the activity sequence with the maximum

conditional probability given the observation sequence using Viterbi algorithm. Given the training data $D = (D_1, D_2, \dots, D_N)$ where $D_i = (A_i, O_i)$, the learning criteria is to find a weight vector W that maximizes the log-likelihood of the training data. A zero mean Gaussian prior is assumed to avoid overfitting. A single variance σ^2 is used to control the degree of penalization for each weight w_i . Higher σ^2 makes the model tend to fit the training data. CRF++ uses L-BFGS [22] for the optimization that is shown to be effective in previous papers.

3.3.3 SVM^{hmm}

SVM^{hmm} [12] is a sequence labeling instantiation of structural SVM. Similar to LCRF, structural defines a linear discriminant function $D(A, O)$ by a set of feature functions $F = \{f_1(A, O, t), f_2(A, O, t), \dots, f_J(A, O, t)\}$ and a weight vector $W = \{w_1, w_2, \dots, w_J\}$. The linear discriminant function $D(A, O)$ is defined as

$$D(A, O) = \sum_j \sum_t w_j f_j(A, O, t).$$

Here we use the same feature functions in LCRF and SVM^{hmm}.

Unlike the maximum likelihood estimation in CRF and HMM, structural SVM does not model the probabilities but discriminate between different label sequences. The learning criteria is similar to conventional SVM that maximizes the margin. The loss function is the misclassified labels in a sequence. A cost factor c is used to control the trade off between the margin and loss. Higher c makes the model tend to fit the training data.

3.3.4 Other Approaches

Frame-Based Classification

To evaluate how we benefit from the modeling of the temporal relationship, we use three classifiers, NBC, maximum entropy classifier (MEC), and SVM for comparison. In this formulation, each time frame is viewed as an instance and the activity is independently classified with the observation in the frame. NBC, MEC and SVM can be viewed as a specialization of HMM, CRF and SVM^{hmm}. Here we use the same implementation in HMM, CRF and SVM^{hmm}.

3.4 Performance Measures

The output $P = (P_1, P_2, \dots, P_T)$ of the recognition is a string of T predictions. Each prediction P_t belongs to one of the 12 activities. The ground truth $G = (G_1, G_2, \dots, G_T)$ is the annotated activity sequence. To evaluate how our recognition algorithm performs, the frame accuracy and the average class accuracy are used for the following comparisons.

Frame Accuracy

The frame accuracy (FA) is the rate of matching frames between the prediction and the ground truth. Thus, it is defined as

$$FA(P, G) = \frac{\sum_{t=1}^T \delta(P_t, G_t)}{T} \text{ where } \delta(P_t, G_t) = \begin{cases} 1, & \text{if } P_t = G_t. \\ 0, & \text{otherwise.} \end{cases}$$

Average Class Accuracy

The frame accuracy is easily affected by the long activity. For example, in E-Home dataset, a system that always predicts the activity as "watching TV" can be as accurate as 17% that is much higher than random guess due to the high coverage of watching TV.

The average class accuracy (ACA) is the normalized frame accuracy by each activity.

The measure is defined as

$$Recall(P, G, a) = \frac{\sum_{t=1}^T \Delta(P_i, G_i, a)}{\sum_{t=1}^T \delta(G_i, a)} \text{ where } \Delta(P_i, G_i, a) = \begin{cases} 1, & \text{if } P_i = G_i = a. \\ 0, & \text{otherwise.} \end{cases}$$

$$ACA(P, G) = \frac{\sum_{a=1}^N Recall(P, G, a)}{N}.$$

3.5 Raw Features

We first use the raw features for the evaluation. Raw features include a 24-dimensional sequence of real value vector for the audio sensor, a discrete sequence for the RFID system and a discrete sequence for the location system. The features of these three sensors diverge in form. Audio features are real value vectors while RFID features and location features are discrete values. In additions, RFID features differ from location features in sparsity. In E-home dataset, RFID returns null in 90% of time.

For evaluation, we use leave-one-subject-out cross-validation. The sequence of each subject is tested once using the model trained with sequences of the rest 12 subjects.

3.5.1 Results

To compare the characteristic of each sequence model for dealing with heterogeneous sensors, we evaluate the frame accuracy and the average class accuracy for all seven combinations of the three sensors using HMM, LCRF and SVM^{hmm}.

The smooth parameter w in HMM is tested from 0.90 to 0.99 and 0.01 as a step. The variance σ^2 for LCRF and the cost factor c for SVM^{hmm} are tested from 2^{-3} to 2^4 and multiplied by 2 as a step. The parameters that achieve highest frame accuracy are used for each model. The results are summarized in Table 3.1.

FA/ACA(%)	HMM	LCRF	SVM ^{hmm}
Audio	23.5/24.8	37.6/27.3	45.0/36.4
RFID	44.9/44.1	51.4/44.4	59.5/50.9
Location	31.5/37.4	43.3/37.1	40.2/37.5
Audio+RFID	31.9/33.5	62.3/54.9	69.7/63.5
Audio+Location	39.5/41.9	56.3/48.6	61.0/56.0
Location+RFID	39.6/45.0	63.8/56.8	65.2/60.3
All	44.3/46.8	68.8/64.6	72.0/67.8

Table 3.1: Performance Comparison of HMM, LCRF and SVM^{hmm} using Raw Features.

In HMM, the result of fusing all sensors is even worse than using RFID only. The situation informs us that it can be dangerous to fuse sensors in a single HMM. Assuming independence for the 24-dimensional audio features in HMM is dangerous since these features are extracted from the same audio signal.

With discriminative models such as LCRF and SVM^{hmm}, fusing more sensors generally performs better. The accuracy of LCRF and SVM^{hmm} using all sensors is much

better than HMM.

In our experiment, LCRF performs slightly worse than SVM^{hmm} . We can see that SVM^{hmm} show robustness in dealing with varieties of sensors while LCRF is relatively weak in dealing with RFID and audio features. It seems inappropriate to assume simple distribution between the activity and the audio features. Fitting parameters to a wrong distribution can result in severe bias. For RFID, since the event is sparse, there may not be enough counts for CRF to overcome the prior.

To show the usefulness of temporal relationship, we evaluate how sequence models improve over the frame-based classification by considering the temporal relationship. The results of different models are shown in Table 3.2.

FA/ACA(%)	NBC/HMM	MEC/LCRF	SVM/SVM ^{hmm}
Classification	35.5/34.2	42.7/39.7	43.7/40.9
Sequence Models	40.9/43.1	68.8/64.6	72.0/67.8

Table 3.2: Performance Comparison of Frame-Based Classification and Sequence Models.

The results show that all sequence models outperform corresponding classifiers in both performance measures. The improvement of the frame accuracy by considering the temporal relationship can be up to 28.3% in SVM^{hmm} . The improvement of the average class accuracy can be up to 27% in LCRF. The significant difference shows us frame-based classification is not adequate in this problem.

3.6 Overlapping Features

Observation can be view in multiple ways. For example, in natural language processing, the word "White" can be viewed as the word itself or a capitalized word. In recognizing the name entity, the capitalization feature may be very informative. Discriminative models such as LCRF and SVM^{hmm} are shown to be able to utilize this kind of overlapping features. We describe three different strategies to extract features in our problem.

3.6.1 Generative Audio Probabilities

The audio in a single second may not contain sufficient information. In additions, performing activities may generate different sounds in different stages. For example, in preparing meals, the sound of chewing can be very different from using microwave. We use an HMM to model the relationship between a specific activity and a small segment of audio features.

Given the training data, at each time c , we segment a small length w of audio features $S_c^{LR} = (O_{c-w}^{Audio}, O_{c-w+1}^{Audio}, \dots, O_c^{Audio})$ and associate the segment S_c^{LR} with the activity label A_c . For each activity i , we independently train an HMM parameter λ_i^{LR} using the segments with activity label i . By reversing the time index, we segment a set of audio features S_c^{RL} and train an HMM parameter λ_i^{RL} for each activity i . As a result, we have 24 HMM parameters.

For a testing sequence, we use a backward sliding window as well as a forward sliding window to segment two audio sequence of w frames with 50% overlapping.

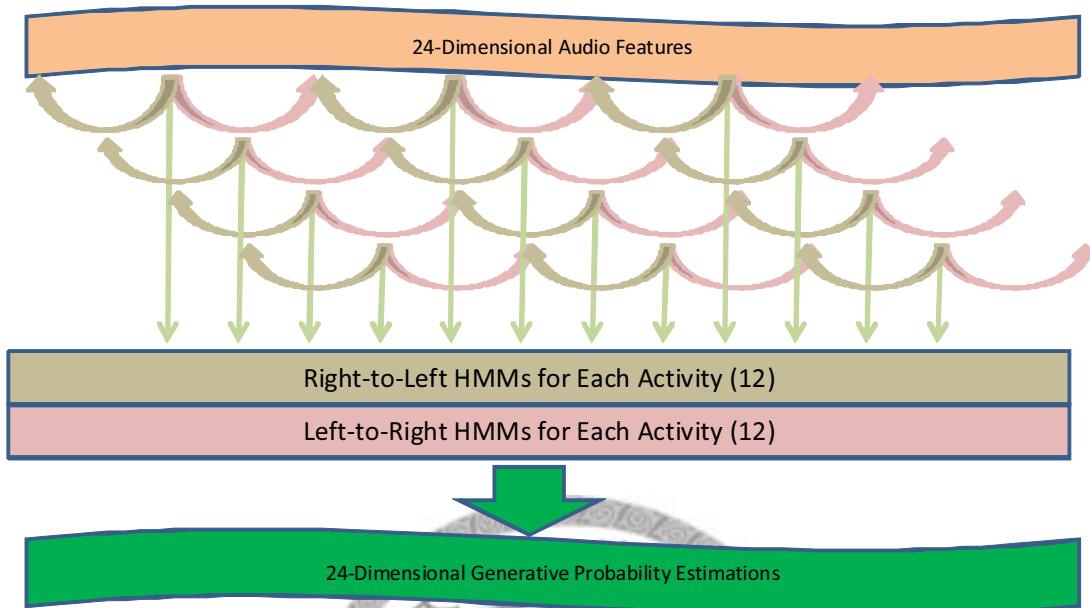


Figure 3.1: Generative Audio Probabilities.

We then use the 24 HMM parameters to estimate the generative probabilities of the two segments. The probability estimations are logged and scaled to values ranging from 0 to 1. These 24 probability estimations are used as an additional feature vector. Figure 3.1 show the process of creating these features.

3.6.2 Region and Region Transitions

We group the 40 load sensory blocks into 3 places including the living room, the dining room and the workspace. The location feature is the index of the 40 blocks while the region feature is the corresponding place. In additions, we consider all transitions between these 3 places as additional features. Figure 3.2 shows the relationship of the

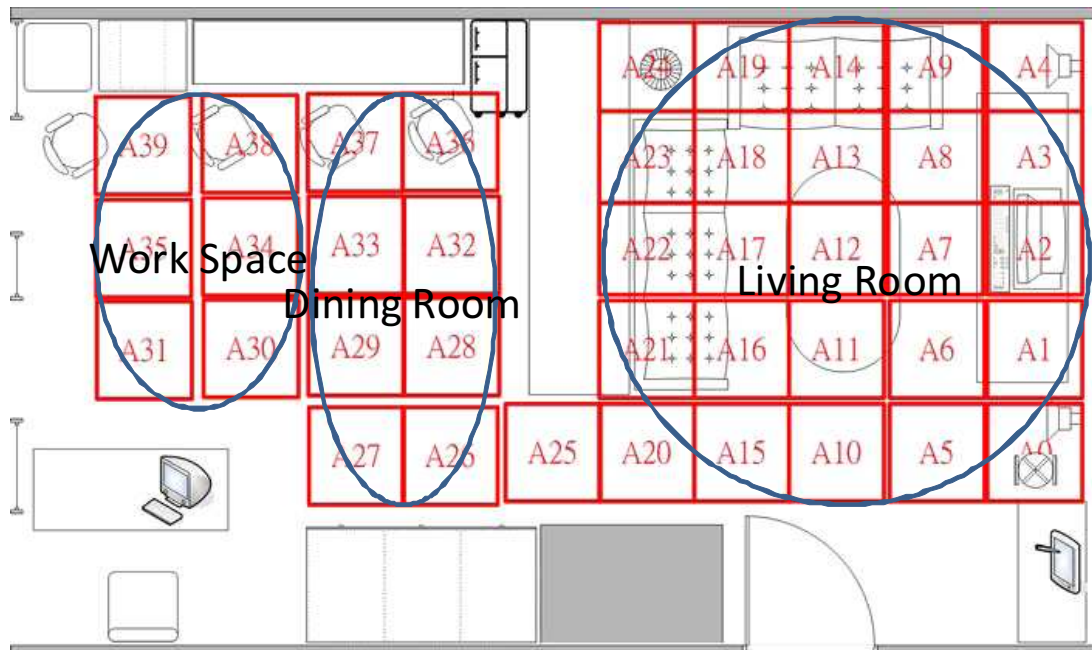


Figure 3.2: Regions and Sensory Blocks.

regions and location sensory blocks.

3.6.3 NextRFID and LastRFID

When a non-null object is read by the RFID reader, it is usually very informative for disambiguating activities. Referring to the recent non-null object reading is helpful. For example, if we hold the TV remote control at last 3 seconds, it is very possible that we are watching TV currently. We define two features, NextRFID and LastRFID for expanding the raw RFID features.

For non-null object i at the time frame c , if the nearest RFID reading of object i is at the time frame c' where c' is larger than c , NextRFID distance is defined as $(c' - c)$.

To prevent referring to the object that is irrelevant in time frames of other activities, the maximum distance is limited. In additions, the forward referencing process ended when it encounters the location change or another object. LastRFID distance is defined in the same way of the NextRFID except that the order of time frames is reversed. The resulting distances for objects are scaled to values ranging from 0 to 1. As a result, we have a new 48-dimensional features for RFID. These features are used as a replacement of the raw RFID readings.

3.6.4 Results

To show the effect of incorporating these overlapping features, we evaluate the performance of LCRF and SVM^{hmm} with these overlapping features. The results are summarized in Table 3.3.

FA/ACA(%)	CRF		SVM ^{hmm}	
	Raw	Overlapping	Raw	Overlapping
Audio	37.6/27.3	40.1/29.8	45.0/36.4	45.9/37.4
RFID	51.4/44.4	63.6/56.8	59.5/50.9	63.0/55.4
Location	43.3/37.1	45.9/39.2	40.2/37.5	44.1/40.3
All	68.8/64.6	74.8/70.0	72.0/67.8	73.0/71.1

Table 3.3: Performance Comparison of LCRF and SVM^{hmm} Using Raw Features and Overlapping Features.

By combining these overlapping features, we improve the accuracy of LCRF and SVM^{hmm} in all sensor settings. Note that the frame accuracy of LCRF is improved from 51.4% to 63.6% with the NextRFID and LastRFID features because the two features solve the sparsity of the raw RFID readings. With these features, the performance

of LCRF and SVM^{hmm} is close.

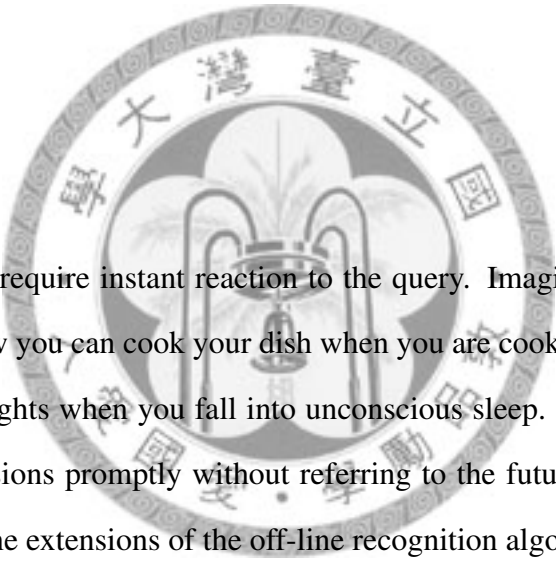




Chapter 4

On-line Recognition for Active

Services



Some applications require instant reaction to the query. Imagine an advising system which suggests how you can cook your dish when you are cooking or a care-giver that turns off TV and lights when you fall into unconscious sleep. In such a problem, we have to make decisions promptly without referring to the future. In this chapter, we introduce the on-line extensions of the off-line recognition algorithms.

4.1 Problem Definition

We define the recognition problem as follow. For each time c from 1 to T , given the partial observation sequence $O_{1:c} = (O_1, O_2, \dots, O_c)$ where O_t is the sensor observation at time t , we predict an activity label A_c where A_c is the activity at time c . The

definition of the observation sequence is the same as what we used in the previous chapter.

4.2 Dynamic Programming Algorithms

Viterbi algorithm is a general inference algorithm for HMM, LCRF and SVM^{hmm}. Given an observation sequence O , Viterbi algorithm finds an output sequence A that maximizes measures such as the generative probability $P(A, O)$ in HMM, the conditional probability $P(A|O)$ in LCRF, and the linear discriminant function $D(A, O)$ in SVM^{hmm}. Here we generalize these measures as a score function $S(A, O)$.

4.2.1 On-line Viterbi Algorithm

With simple modification, Viterbi algorithm can be used to handle the on-line recognition problem. Assume Viterbi algorithm returns an output sequence $A_{1:c}^* = (A_1^*, A_2^*, \dots, A_c^*)$ given the partial observation sequence $O_{1:c}$ such that

$$A_{1:c}^* = \arg \max_{A_{1:c}} S(A_{1:c}, O_{1:c}).$$

On-line Viterbi algorithm is simply to return A_c^* at time c . The output can be efficiently determined with the well-know max-product algorithm. As a result, the inference time for every time slice is $O(n^2)$ where n is the number of activities.

4.2.2 Bayes Filtering

Bayes filtering is originally used to estimate the marginal probability $P(A_c|O_{1:c})$ in HMM. By following the framework, we use Bayes filtering to solve the on-line recognition problem. We return the activity A_c^* at time c such that

$$A_c^* = \arg \max_{A_c} \sum_{A_{1:c-1}} S(A_{1:c-1} : A_c, O_{1:c})$$

where $A_{1:c-1} : A_c = (A_1, A_2, \dots, A_{c-1}, A_c)$. The output can be determined with the well-known sum-product algorithm. As a result, the inference time for every time slice is also $O(n^2)$.

4.2.3 Token Passing Algorithm

Token passing algorithm can be viewed as a greedy Viterbi algorithm. Once the activity label is predicted at time c , we determine it as the true label and find the next best label from the current label. Assume token passing algorithm returns A_t^{tp} at time t , token passing algorithm returns A_c^{tp} at time c such that

$$A_c^{tp} = \arg \max_{A_c} S(A_{1:c-1}^{tp} : A_c, O_{1:c})$$

where $A_{1:c-1}^{tp} : A_c = (A_1^{tp}, A_2^{tp}, \dots, A_{c-1}^{tp}, A_c)$. Since it is a greedy algorithm, the inference time is only $O(n)$ for every time slice.

We can see that the three algorithms are similar except they return labels according to different measures. Bayes filtering returns the optimal label given the partial observation while on-line Viterbi algorithm returns the label in the optimal sequence.

We suppose both algorithms should be accurate. However, when two activities are ambiguous, the output labels may frequently alter.

On the contrary, token passing algorithm returns sub-optimal labels. Due to the greedy fashion, the error propagates when the early result is wrong. However, the labeling result is consistent in the whole process. In additions, token passing algorithm is much efficient. When the number of activities is large, token passing algorithm is favored since $O(n^2)$ in other algorithms can be too slow.

4.3 Evaluation

We use E-home dataset as our evaluation dataset. We implement on-line Viterbi algorithm, Bayes filtering and token passing algorithm in SVM^{hmm}. We evaluate the results with all raw features. The results are summarized in Table 4.1.

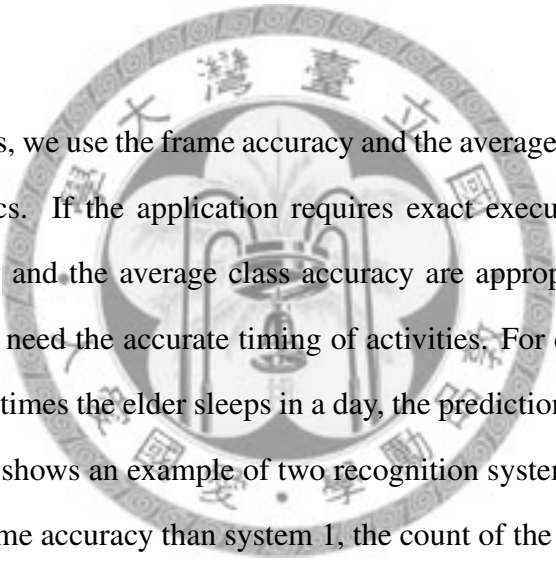
FA/ACA(%)	Viterbi	On-line Viterbi	Bayes Filtering	Token Passing
Accuracy	72.0/67.8	65.0/62.0	64.8/61.9	52.1/47.3

Table 4.1: Performance Comparison of Viterbi, On-line Viterbi, Bayes Filtering and Token Passing Algorithms.

Due to the limitation of instant response, all three on-line recognition algorithms are worse than the standard Viterbi algorithm for the off-line recognition. On-line Viterbi and Bayes filtering perform comparably in this setting. The low accuracy of token passing is due to the error propagation. Comparing to the frame-based classification algorithms, the frame accuracy of these algorithms are still better. Historical temporal information is still very helpful in this problem.

Chapter 5

Segment Analysis



In previous chapters, we use the frame accuracy and the average class accuracy as main performance metrics. If the application requires exact executive time of activities, the frame accuracy and the average class accuracy are appropriate. However, some applications do not need the accurate timing of activities. For example, if we want to monitor how many times the elder sleeps in a day, the prediction of timing is not really needed. Figure 5.1 shows an example of two recognition systems. Although system 2 achieves higher frame accuracy than system 1, the count of the sleeping is wrong.

Assuming the service is provided when an activity transition occurs, we can find similar problem in such an application. Consider a context-aware service that turns off the light when the user sleeps and turn on the light when the user wakes up. In the same example in Figure 5.1, although system 2 achieves higher frame accuracy than system 1, it unexpectedly turns on the light when the user is sleeping.

To evaluate how the recognition system performs in such applications, we define

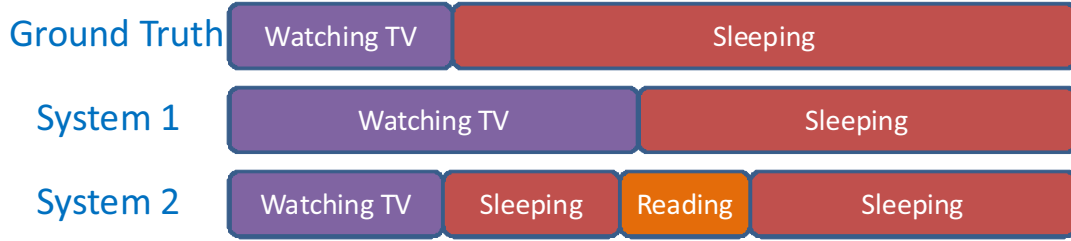


Figure 5.1: Example for Segment Errors.

the performance measure in a different view. For a predicted activity sequence $P = (P_1, P_2, \dots, P_T)$, we transform it to a segment sequence $PS = (PS_1, PS_2, \dots, PS_P)$. Each segment consists of three elements such that $PS_i = (PS_i^A, PS_i^{ST}, PS_i^{ET})$ where PS_i^A is the activity label, PS_i^{ST} is the starting time and PS_i^{ET} is the ending time. Algorithm 1 shows how the transformation is done. For the ground sequence $G = (G_1, G_2, \dots, G_T)$, the ground segment $GS = (GS_1, GS_2, \dots, GS_G)$ are defined in the same way where $GS_i = (GS_i^A, GS_i^{ST}, GS_i^{ET})$.

5.1 Segment Error

5.1.1 Minimum Edit Distance

We evaluate the segment error using minimum edit distance which is a well-known measure. Edit distance or Levenshtein distance is a metric for measuring the difference between two strings. For example, the minimum edit distance between the string “abba” and “ababb” is 2 because we can modify the “abba” to be “ababb” with 2 edits.

An edit can be a substitution that substitutes a character, an insertion that inserts a

Algorithm 1 SEGMENT(P)

```

1:  $PS_1^A \leftarrow P_1$ 
2:  $PS_1^{ST} \leftarrow 1$ 
3:  $i \leftarrow 1$ 
4:  $last \leftarrow P_1$ 
5: for  $t \leftarrow 2$  to  $T$  do
6:   if  $P_t \neq last$  then
7:      $PS_i^{ET} \leftarrow t - 1$ 
8:      $i \leftarrow i + 1$ 
9:      $PS_i^A \leftarrow P_t$ 
10:     $PS_i^{ST} \leftarrow t$ 
11:     $last \leftarrow P_t$ 
12:   end if
13: end for
14:  $PS_i^{ET} \leftarrow T$ 
15: return  $PS$ 

```

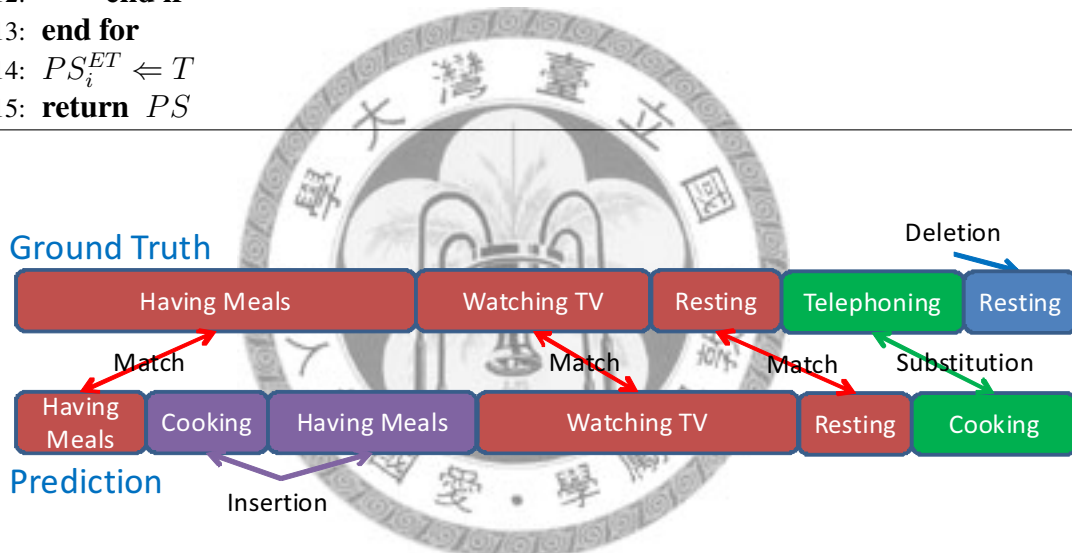


Figure 5.2: Example for Minimum Edit Distance.

character and a deletion that deletes a character. In the example above, we add an “a” before the third character and substitute the last “a” with a “b” in the string “abba” such that the two strings become equal. Figure 5.2 shows an example of the minimum edit distance. Algorithm 2 shows how to evaluate the number of matches, substitutions, insertions, deletions between two strings.

The edit distance over the prediction segments PS and the ground truth segments GS is thus directly related to the counting problem we mentioned. An insertion error means one over count of an activity while a deletion error means one less count. In addition to the counting problem, other applications such as mining patterns of activity transitions may also favor this measure.

In addition to the edit distance, word error rate (WER) is a widely used measurement in the speech recognition. WER is defined based on the edit distance. The definition is as follow.

$$WER = \frac{\textit{insertion} + \textit{deletion} + \textit{substitution}}{G}$$

5.1.2 Time Critical Minimum Edit Distance

Some on-line applications are time critical. If we make a prediction after the activity is finished, the service is also provided in vain. We define a time critical minimum edit distance for these applications.

The algorithm for computing the time critical minimum edit distance is similar the minimum edit distance except we do not allow two segments to match when there is no overlapping. We match the first segment that overlaps with the ground segment. For an insertion segment that has the same activity label with the ground truth, the segment is specialized as a fragment segment. A substitution is replaced with an insertion and a deletion. Figure 5.3 shows an example of the time critical minimum edit distance. We can see that an insertion results in an unexpected service while a deletion results in a

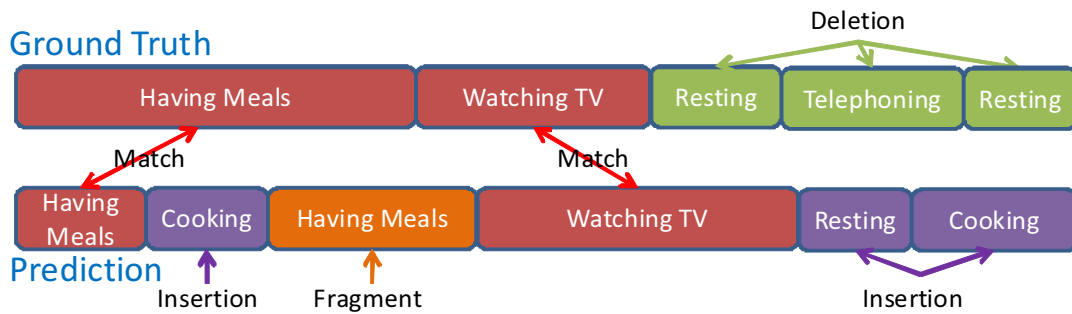


Figure 5.3: Example for Time Critical Minimum Edit Distance.

missing service. A fragment results in a duplicate service.

Algorithm 3 is the process for evaluating the time critical minimum edit distance.

We define the word error rate (WER) to measure the quality of services. We also define the precision and recall such that a recognition system with low precision tends to provide more unexpected services while a system with low recall tends to miss the chance to provide the service. The definitions are as follow.

$$WER = \frac{insertion + deletion}{G}$$

$$Precision = \frac{match}{(match + insertion)}$$

$$Recall = \frac{match}{G}$$

5.2 Evaluation

5.2.1 Off-line Recognition

We evaluate LCRF and SVM^{hmm} with raw features using the edit distance. The results are summarized in Table 5.1.

Results(%)	Match	Substitution	Insertion	Deletion	WER
LCRF	113	27	24	16	42.9
SVM ^{hmm}	127	25	65	4	60.3

Table 5.1: Segment Error of LCRF and SVM^{hmm}.

Although the frame accuracy is very similar in LCRF and SVM^{hmm}. Their behavior of prediction in segment level is quite different. LCRF makes fewer predictions while SVM^{hmm} tends to predict aggressively. As a result, the insertion errors of SVM^{hmm} are higher and the deletion and substitution errors of LCRF are higher. This may guide us in choosing models when applications vary.

5.2.2 On-line Recognition

We evaluate inference algorithms in the previous chapter using the time critical minimum edit distance. The results are summarized in table 5.2.

Although on-line Viterbi algorithm and Bayes filtering are superior to token passing algorithm in the frame accuracy, they are very bad at word error rate. Due to the ambiguous situation when two activities are highly possible, the prediction may jump back and forth in these two activities. As a result, the insertion errors become very

Results(%)	Match	Insertion	Deletion	Fragment	WER	Precision	Recall
Viterbi	129	86	27	11	67.9	60.0	82.7
O. V.	139	376	17	99	251.9	27.0	89.1
B. F.	140	379	16	102	253.2	31.0	89.7
T. P.	97	122	59	15	116	44.3	62.2

Table 5.2: Segment Error of On-line Algorithms.

high. We can see that wrong services are more than twice of correct services.

5.3 Smooth on-line Viterbi

In some services, a wrong prediction may be critical. For example, the light control system should not turn on the light when the user is sleeping. To reduce the high insertion errors in on-line Viterbi algorithm, we propose smooth on-line Viterbi algorithm.

As we mention in the previous chapter, a score function $S(A, O)$ is used for each model. Given the partial observation sequence $O_{1:c}$ and an activity label a , we define a function $ViterbiScore(O_{1:c}, a)$ that returns a score s such that

$$s = \max_{A_{1:c-1}} S(A_{1:c-1} : a, O_{1:c})$$

where $A_{1:c-1} : a = (A_1, A_2, \dots, A_{c-1}, a)$. $ViterbiScore(O_{1:c}, a)$ function is implicitly implemented in Viterbi algorithm.

To prevent prediction when there is ambiguity, we define a measure for estimating whether it is ambiguous. As we know that the score degrades with time. Absolute score is not a good measure. We define the discriminability d_c at time c as the ratio of highest score versus the second highest score.

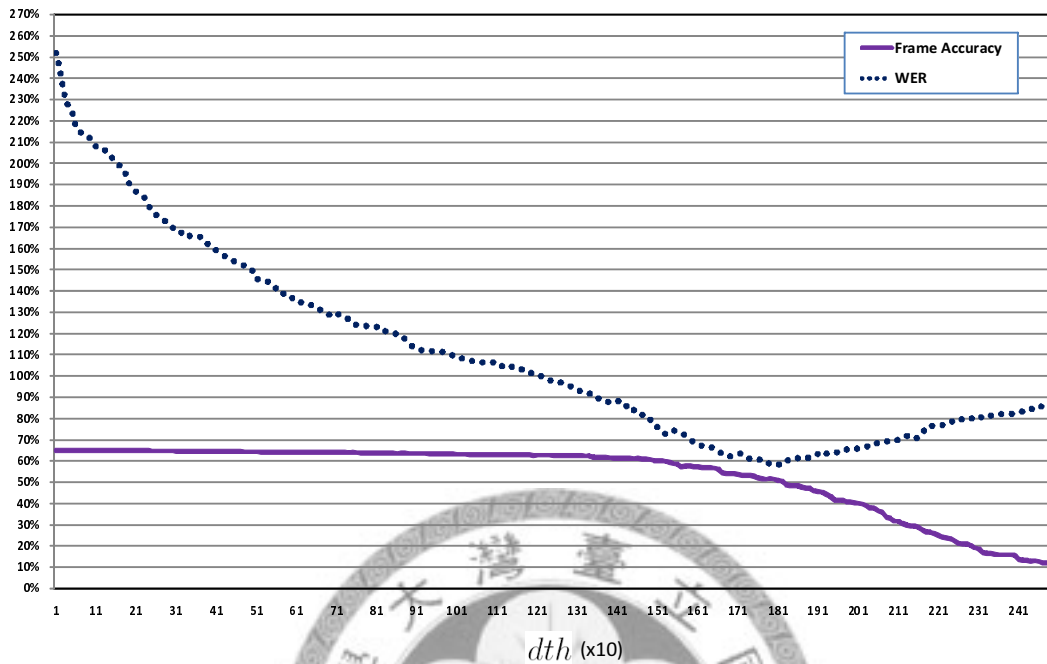


Figure 5.4: Searching the Threshold of Smooth On-line Viterbi Algorithm.

We define a threshold dth such that if d_c is larger than dth , we predict the class with highest score and use the last prediction otherwise.

Algorithm 4 show the smooth on-line Viterbi algorithm.

5.3.1 Evaluation

The evaluation is as follow, Figure 5.4 is the WER and the frame accuracy by increasing dth . The frame accuracy degrades slowly when dth is small. This means that the frames we reject are usually wrong originally. As a result, WER decreases quickly with dth . When dth is 8.9, we achieve similar WER with token passing algorithm

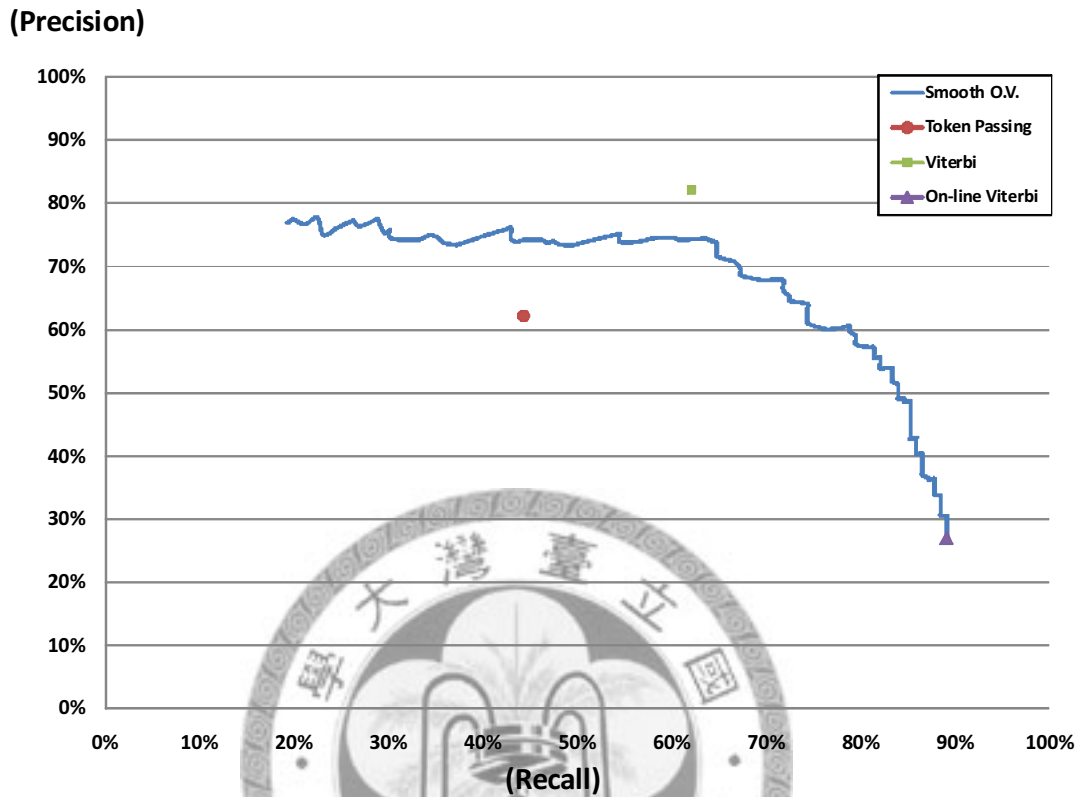


Figure 5.5: PR Curve of Smooth On-line Viterbi.

but remain the frame accuracy as 63.6% which is higher than 52.1% in token passing algorithm. When dth is 17.5, we achieve similar frame accuracy with token passing algorithm but result a lower WER 60.8%. As a result, with a proper threshold, smooth on-line Viterbi algorithm is better than token passing algorithm.

Figure 5.5 is another way to evaluate this algorithm. We show the PR curve by adjusting the dth . The original on-line Viterbi algorithm achieves very low precision. By increasing dth , the precision increase with a price of lower recall. In this way, we can choose different quality of services by adjusting dth for specific applications.

Algorithm 2 MED(GS, PS)

```

1: for  $i \leftarrow 0$  to  $G$  do
2:    $SCORE_{i,0} \leftarrow i$ 
3: end for
4: for  $j \leftarrow 1$  to  $P$  do
5:    $SCORE_{0,j} \leftarrow j$ 
6: end for
7: for  $i \leftarrow 1$  to  $G$  do
8:   for  $j \leftarrow 1$  to  $P$  do
9:      $SCORE_{i,j} \leftarrow \infty$ 
10:    if  $GS_i^A = PS_j^A$  then
11:       $SCORE_{i,j} \leftarrow SCORE_{i-1,j-1}$ 
12:       $BACKTRACK_{i,j} \leftarrow \text{"match"}$ 
13:    else
14:       $SCORE_{i,j} \leftarrow SCORE_{i-1,j-1} + 1$ 
15:       $BACKTRACK_{i,j} \leftarrow \text{"substitution"}$ 
16:    end if
17:    if  $SCORE_{i,j-1} + 1 < SCORE_{i,j}$  then
18:       $SCORE_{i,j} \leftarrow SCORE_{i,j-1} + 1$ 
19:       $BACKTRACK_{i,j} \leftarrow \text{"insert"}$ 
20:    end if
21:    if  $SCORE_{i-1,j} + 1 < SCORE_{i,j}$  then
22:       $SCORE_{i,j} \leftarrow SCORE_{i-1,j} + 1$ 
23:       $BACKTRACK_{i,j} \leftarrow \text{"delete"}$ 
24:    end if
25:  end for
26: end for
27:  $(i, j) \leftarrow (G, P)$ 
28:  $(match, insertion, deletion, substitution) \leftarrow (0, 0, 0, 0)$ 
29: while  $i \neq 0$  or  $j \neq 0$  do
30:   if  $BACKTRACK_{i,j} = \text{"match"}$  then
31:      $(match, i, j) \leftarrow (match + 1, i - 1, j - 1)$ 
32:   else if  $BACKTRACK_{i,j} = \text{"substitution"}$  then
33:      $(substitution, i, j) \leftarrow (substitution + 1, i - 1, j - 1)$ 
34:   else if  $BACKTRACK_{i,j} = \text{"insert"}$  then
35:      $(insertion, j) \leftarrow (insertion + 1, j - 1)$ 
36:   else
37:      $(deletion, i) \leftarrow (deletion + 1, i - 1)$ 
38:   end if
39: end while
40: return  $(match, insertion, deletion, substitution)$ 

```

Algorithm 3 TCMED(GS, PS, G, P)

```

1: for  $i \leftarrow 0$  to  $G$  do
2:    $SCORE_{i,0} \leftarrow i$ 
3: end for
4: for  $j \leftarrow 1$  to  $P$  do
5:   if  $G_{PS_j^{ST}} = P_{PS_j^{ST}}$  then
6:      $SCORE_{0,j} \leftarrow SCORE_{0,j-1} + f_{penal}$   $\{f_{penal} \ll 1\}$ 
7:   else
8:      $SCORE_{0,j} \leftarrow SCORE_{0,j-1} + 1$ 
9:   end if
10: end for
11: for  $i \leftarrow 1$  to  $G$  do
12:   for  $j \leftarrow 1$  to  $P$  do
13:      $SCORE_{i,j} \leftarrow \infty$ 
14:     if  $GS_i^A = PS_j^A$  and  $GS_i^{ST} \leq PS_j^{ET}$  and  $PS_j^{ST} \leq GS_i^{ET}$  then
15:        $latescore \leftarrow l_{penal} \times \min(0, PS_j^{ST} - GS_i^{ST})$   $\{l_{penal} \ll f_{penal}\}$ 
16:        $SCORE_{i,j} \leftarrow SCORE_{i-1,j-1} + latescore$ 
17:        $BACKTRACK_{i,j} \leftarrow "match"$ 
18:     end if
19:     if  $G_{PS_j^{ST}} = P_{PS_j^{ST}}$  and  $SCORE_{i,j-1} + f_{penal} < SCORE_{i,j}$  then
20:        $SCORE_{i,j} \leftarrow SCORE_{i,j-1} + f_{penal}$ 
21:        $BACKTRACK_{i,j} \leftarrow "fragment"$ 
22:     else if  $SCORE_{i,j-1} + 1 < SCORE_{i,j}$  then
23:        $SCORE_{i,j} \leftarrow SCORE_{i,j-1} + 1$ 
24:        $BACKTRACK_{i,j} \leftarrow "insert"$ 
25:     end if
26:     if  $SCORE_{i-1,j} + 1 < SCORE_{i,j}$  then
27:        $SCORE_{i,j} \leftarrow SCORE_{i-1,j} + 1$ 
28:        $BACKTRACK_{i,j} \leftarrow "delete"$ 
29:     end if
30:   end for
31: end for
32:  $(i, j) \leftarrow (G, P)$ 
33:  $(match, insertion, deletion, fragment) \leftarrow (0, 0, 0, 0)$ 
34: while  $i \neq 0$  or  $j \neq 0$  do
35:   if  $BACKTRACK_{i,j} = "match"$  then
36:      $(match, i, j) \leftarrow (match + 1, i - 1, j - 1)$ 
37:   else if  $BACKTRACK_{i,j} = "insert"$  then
38:      $(insertion, j) \leftarrow (insertion + 1, j - 1)$ 
39:   else if  $BACKTRACK_{i,j} = "fragment"$  then
40:      $(fragment, j) \leftarrow (fragment + 1, j - 1)$ 
41:   else
42:      $(deletion, i) \leftarrow (deletion + 1, i - 1)$ 
43:   end if
44: end while
45: return  $(match, insertion, deletion, fragment)$ 

```

Algorithm 4 SMOOTHONLINEVITERBI($O_{1:c}, dth$)

```

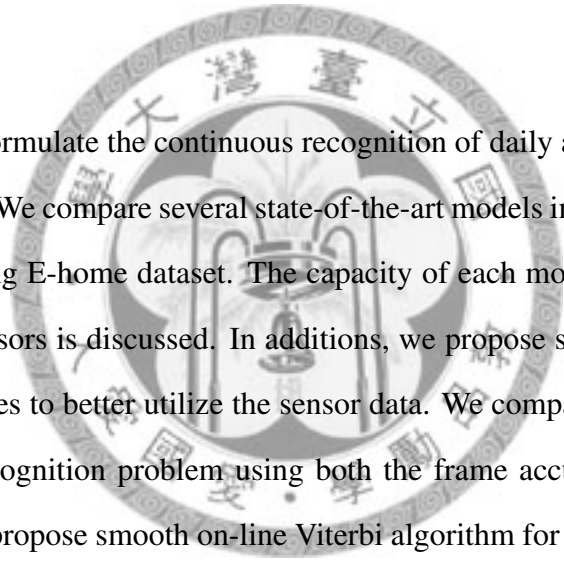
1:  $\hat{a} \leftarrow \arg \max_a ViterbiScore(O_{1:c}, a)$ 
2:  $\hat{a} \leftarrow \arg \max_{a \neq \hat{a}} ViterbiScore(O_{1:c}, a)$ 
3:  $\hat{s} \leftarrow ViterbiScore(O_{1:c}, \hat{a})$ 
4:  $\hat{s} \leftarrow ViterbiScore(O_{1:c}, \hat{a})$ 
5:  $d_t \leftarrow \hat{s} / \hat{s}$ 
6:  $lastact \leftarrow \text{SMOOTHONLINEVITERBI}(O_{1:c-1}, dth)$ 
7: if  $\log(d_c) > dth$  or  $c = 1$  then
8:   return  $\hat{a}$ 
9: else
10:  return  $lastact$ 
11: end if

```



Chapter 6

Conclusion



In this work, we formulate the continuous recognition of daily activities as a sequence labeling problem. We compare several state-of-the-art models including HMM, LCRF, and SVM^{hmm} using E-home dataset. The capacity of each model in dealing with the heterogeneous sensors is discussed. In additions, we propose strategies for extracting overlapping features to better utilize the sensor data. We compare inference strategies for the on-line recognition problem using both the frame accuracy and the segment error. Finally, we propose smooth on-line Viterbi algorithm for the on-line recognition problem.

In our experiment, discriminative models such as LCRF and SVM^{hmm} significantly outperform HMM. SVM^{hmm} is robust in dealing with all three kinds of sensors we used and LCRF is relatively weak in the RFID and audio sensors. By incorporating three overlapping features, the accuracy of both models is improved. We found that the NextRFID and LastRFID features greatly improve the accuracy of LCRF by eliminating

the sparsity of the RFID sensor. As a result, SVM^{hmm} and LCRF perform similarly with these overlapping features.

For the on-line recognition problem, we have shown that the on-line Viterbi algorithm with highest frame accuracy suffers from high insertion errors. Smooth on-line Viterbi algorithm is a way to remove insertion errors without losing too much frame accuracy. In additions, the threshold in smooth on-line Viterbi algorithm can be used to trade the rate of wrong services and missing services.

There is still limitation in this work. The evaluation dataset, E-home dataset is a semi-natural dataset which is collected in an instructed laboratory environment. The activities in a real home can be more complex. For example, we may stop an activity to do another. In additions, we usually carry out multiple activities at the same time. As a result, we should model the multi-tasking problems. An even more complex scenario is to recognize activities in a multi-people environment. Associating the sensor trigger with the subject is not a trivial job.

The performance of current recognition is also not perfect. The highest frame accuracy is 74.8% which is not practical in many real applications. Currently we do not use the physical information of the subjects. Attaching accelerometers are shown to be helpful in recognizing activities. Therefore, we want to incorporate this kind of sensors in the future.

Currently the parameters of models are selected by evaluating a set of predefined values. In addition to the parameters for models, smooth on-line Viterbi algorithm also needs a good threshold. The resolution and range of the searching process greatly affect the accuracy. Selecting suitable parameters can be very time consuming when

encountering a large dataset. As a result, finding an efficient way to choose parameters is a possible research direction.



Bibliography

- [1] Physiological data modeling contest, 2004. Data available at <http://www.cs.utexas.edu/~sherstov/pdmc/>.
- [2] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden markov support vector machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2003.
- [3] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *Proceedings of the International Conference on Pervasive Computing (Pervasive)*, volume 3001 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2004.
- [4] J. Bilmes. *The Graphical Models Toolkit (GMTK)*, May 2007. Software available at <http://ssli.ee.washington.edu/~bilmes/gmtk/>.
- [5] K.-h. Chang, M. Y. Chen, and J. Canny. Tracking free-weight exercises. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp)*, 2007.

- [6] J. Chen, A. H. Kam, J. Zhang, N. Liu, and L. Shue. Bathroom activity monitoring based on sound. In *Proceedings of the International Conference on Pervasive Computing (Pervasive)*, volume 3468 of *Lecture Notes in Computer Science*, pages 47–61. Springer, 2005.
- [7] H. L. Chieu, W. S. Lee, and L. P. Kaelbling. Activity recognition from physiological data using conditional random fields. Technical report, Singapore-MIT Alliance (SMA) Annual Symposium, January 2006.
- [8] T. Choudhury and S. Basu. Modeling conversational dynamics as a mixed memory markov process. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.
- [9] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages I: 838–845, 2005.
- [10] T. Huynh, U. Blanke, and B. Schiele. Scalable recognition of daily activities with wearable sensors. In *Proceedings of the International Symposium on Location- and Context-Awareness (LoCA)*, volume 4718 of *Lecture Notes in Computer Science*, pages 50–67. Springer, 2007.
- [11] S. S. Intille, K. Larson, E. M. Tapia, J. S. Beaudin, P. Kaushik, J. Nawyn, and R. Rockinson. Using a live-in laboratory for ubiquitous computing research. In *Pro-*

ceedings of the International Conference on Pervasive Computing (*Pervasive*), volume 1, pages 349–365, 2006.

- [12] T. Joachims. *SVM^{hmm} : Sequence Tagging with Structural Support Vector Machines*, May 2008. Software available at http://svmlight.joachims.org/svm_struct.html.
- [13] S. Katz, A.B. Ford, R.W. Moskowitz, B.A. Jackson, and M.W. Jaffe. Studies of illness in the aged: The index of ADL: A standardized measure of biological and psychosocial function. *Journal of the American Medical Association*, 185(12):914–9, 1963.
- [14] S. S. Keerthi and S. Sundararajan. CRF versus SVM-struct for sequence labeling. Technical report, Yahoo Research, 2007.
- [15] T. Kudo. *CRF++: Yet Another CRF toolkit*, December 2007. Software available at <http://crfpp.sourceforge.net/>.
- [16] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models. for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML)*, June 2001.
- [17] M.P. Lawton and E.M. Brody. Assessment of older people: self-maintaining and instrumental activities of daily living. *Gerontologist*, 9(3):179–86, 1969.
- [18] L. Liao, D. Fox, and H. A. Kautz. Location-based activity recognition using relational markov networks. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 773–778. Professional Book Center, 2005.

- [19] L. Liao, D. Fox, and H. A. Kautz. Extracting places and activities from GPS traces using hierarchical conditional random fields. *International Journal of Robotics Research*, 26:119–134, 2007.
- [20] B. Limketkai, L. Liao, and D. Fox. CRF-Filters: Discriminative particle filters for sequential state estimation. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, April 2007.
- [21] C.-y. Lin. IPARS: Intelligent portable activity recognition system. Master’s thesis, National Taiwan University, 2006.
- [22] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.
- [23] B. Logan, J. Healey, M. Philipose, E. M. Tapia, and S. S. Intille. A long-term evaluation of sensing modalities for activity recognition. In *Proceedings of the International Conference on Ubiquitous Computing (UbiComp)*, volume 4717 of *Lecture Notes in Computer Science*, pages 483–500. Springer, 2007.
- [24] J. Lopes, C. Lin, and S. Singh. Multi-stage classification for audio based activity recognition. In *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, volume 4224 of *Lecture Notes in Computer Science*, pages 832–840. Springer, 2006.
- [25] U. Maurer, A. Smailagic, D. P. Siewiorek, and Michael D. Activity recognition and monitoring using multiple sensors on different body positions. In *Interna-*

- tional Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, pages 113–116. IEEE Computer Society, 2006.
- [26] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2000.
- [27] N. Nguyen and Y. Guo. Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2007.
- [28] C.-w. Pan. Rehabilitation exercises recognition based on acceleration signals. Master’s thesis, National Taiwan University, 2007.
- [29] D. J. Patterson, D. Fox, H. A. Kautz, and M. Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *Proceedings of the International Symposium on Wearable Computers (ISWC)*, pages 44–51. IEEE Computer Society, 2005.
- [30] D. J. Patterson, L. Liao, D. Fox, and H. A. Kautz. Inferring high-level behavior from low-level sensors. In *Proceedings of the International Conference on Ubiquitous Computing (UbiComp)*, volume 2864 of *Lecture Notes in Computer Science*, pages 73–89. Springer, 2003.
- [31] W. Pentney, A.-M. Popescu, S. Wang, H. A. Kautz, and M. Philipose. Sensor-based understanding of daily life via large-scale use of common sense. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, 2006.

- [32] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman. Activity recognition from accelerometer data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1541–1546. AAAI Press / The MIT Press, 2005.
- [33] W.A. Rogers, B. Meyer, N. Walker, and A.D. Fisk. Functional limitations to daily living tasks in the aged: a focus group analysis. *Human Factors*, 40(1):111–125, 1998.
- [34] M. Shimosaka, T. Mori, and T. Sato. Robust action recognition and segmentation with multi-task conditional random fields. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2007.
- [35] C. Sminchisescu, A. Kanaujia, and D. Metaxas. Conditional models for contextual human motion recognition. *Computer Vision and Image Understanding*, 104(2):210–220, 2006.
- [36] P. Smyth, D. Heckerman, and M. I. Jordan. Probabilistic independence networks for hidden markov probability models. *Neural Computation*, 9(2):227–269, 1997.
- [37] T. Stiefmeier, G. Ogris, H. Junker, P. Lukowicz, and G. Tröster. Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario. In *International Symposium on Wearable Computers (ISWC)*, pages 97–104. IEEE, 2006.
- [38] C. A. Sutton and A. McCallum. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Tasker, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2006.

- [39] E. M. Tapia, S. Intille, and K. Larson. Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In *Proceedings of the International Symposium on Wearable Computers (ISWC)*, 2007.
- [40] E. M. Tapia, S. S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. In *Proceedings of the International Conference on Pervasive Computing (Pervasive)*, volume 3001 of *Lecture Notes in Computer Science*, pages 158–175. Springer, 2004.
- [41] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6:1453–1484, 2005.
- [42] T. Wada and T. Matsuyama. Multiobject behavior recognition by event driven selective attention method. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(8):873–887, 2000.
- [43] H. M. Wallach. Conditional random fields: An introduction. Technical Report MS-CIS-04-21, Department of Computer and Information Science, University of Pennsylvania, April 2004.
- [44] J. A. Ward. *Activity Monitoring: Continuous Recognition and Performance Evaluation*. PhD thesis, ETH Zürich, Switzerland, 2006.
- [45] D. H. Wilson. *Assistive Intelligent Environments for Automatic Health Monitoring*. PhD thesis, Carnegie Mellon University, September 2005.

- [46] D. H. Wilson and C. G. Atkeson. Simultaneous tracking and activity recognition (STAR) using many anonymous, binary sensors. In Hans-Werner Gellersen, Roy Want, and Albrecht Schmidt, editors, *Proceedings of the International Conference on Pervasive Computing (Pervasive)*, volume 3468 of *Lecture Notes in Computer Science*, pages 62–79. Springer, 2005.
- [47] Phil Woodland, Gunnar Evermann, and Mark Gales. *HTK - Hidden Markov Model Toolkit - Speech Recognition toolkit*, December 2006. Software available at <http://htk.eng.cam.ac.uk/>.
- [48] T.-y. Wu, C.-c. Lian, and J. Y.-j. Hsu. Joint recognition of multiple concurrent activities using factorial conditional random fields. Technical Report WS-07-09, AAAI Workshop on Plan, Activity, Intent Recognition (PAIR), 2007.
- [49] D. Wyatt, M. Philipose, and T. Choudhury. Unsupervised activity recognition using automatically mined common sense. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 21–27. AAAI Press / The MIT Press, 2005.
- [50] C.-n. Yen. Utilizing heterogeneous sensors for everyday activity recognition. Master's thesis, National Taiwan University, 2007.

