LEARNING-BASED HUMAN ACTIVITY RECOGNITION

by

HAO HU

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy
in Computer Science and Engineering

December 2012, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

胡昊

HAO HU

LEARNING-BASED HUMAN ACTIVITY RECOGNITION

by

HAO HU

This is to certify that I have examined the above Ph.D. thesis

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by

the thesis examination committee have been made.

Qiang Yang

PROF. QIANG YANG, THESIS SUPERVISOR

PROF. MOUNIR HAMDI, HEAD OF DEPARTMENT

Department of Computer Science and Engineering

6 December 2012

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere thanks to my supervisor Prof. Qiang Yang. You have influenced and changed me throughout my Ph.D. career. I'm deeply impressed by your vision and ability to approach compelling research problems, by your high academic standards, self-discipline as well as hard work, by your motivation and interest to conduct excellent research. Working with you has been a real pleasure in the past five years.

I would also like to thank Prof. Yu-Hsing Wang, Prof. Raymond Wong, Prof. Dit-Yan Yeung, Prof. Ke Yi, Prof. Zheng Rong and Dr. Xing Xie for serving in my thesis defense committee and provide useful discussions and insightful comments for me to revise my thesis work.

I'm happy to be able to get to work and befriend with so many excellent and nice students in HKUST. I owe a lot to you and you've really helped me out in countless situations. Please allow me to say thank you to all my friends at HKUST: Bin Cao, Weizhu Chen, Yu-Feng Li, Nathan Nan Liu, Zhongqi Lu, Xin Miao, Kaixiang Mo, Sinno Jialin Pan, Si Shen, Ben Tan, Lu Wang, Naiyan Wang, Yige Wang, Bin Wu, Evan Wei Xiang, Xiao Xiao, Qian Xu, Chao Yang, Lili Zhao, Yi Zhen, Vincent Wenchen Zheng, Erheng Zhong, Wenliang Zhong, Yin Zhu, and so on.

I would like to thank Tong Zhu for your friendship, selfless help and encouragement over the past seven years. Thank you for sharing these years and all of these amazing experience with me. It's my honor to have you as my friend and have a friend like you. I would also like to thank my girlfriend, April Hua Liu for her love and encouragement. Thank you with all my heart!

And I am also very thankful to my mentors in Microsoft Research Asia and Microsoft Research, Dr. Jian-Tao Sun and Dr. Chao Liu. I spent an interesting and colorful nine months in Microsoft. Thank you for the fun and the encouraging discussions in Microsoft.

Finally, I would like to thank my parents for supporting me spiritually throughout my life.

TABLE OF CONTENTS

LEARNING-BASED HUMAN ACTIVITY RECOGNITION

by

HAO HU

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

ABSTRACT

Recognizing human activities has been an extensive and interesting research topic since early 1980s. However, when deploying human activity recognition solutions to the real world, the solutions we provide must satisfy a series of requirements. We would expect our solution to be able to learn a reasonable model from as limited training data as possible. We also hope our solution would be able to deal with the complex relationships which exist in human activities. As is the case for almost all machine learning solutions, we would hope that our solution is scalable and efficient. In this thesis, we start by surveying related work and then study the solution to some specific challenges which are important to deploy these activity recognition systems in the real world.

Specifically, We first analyze how to recognize multiple activities in the physical world environment, especially when such activities have concurrent and interleaving relationships. Next, we extend such a framework to the problem of Web query classification, by exploiting the relatedness of search queries to activities with interleaving relationships and propose a context-aware query classification algorithm.

Secondly, we study the problem of abnormal activity recognition. These abnormal activities are rare to happen and it is difficult to collect enough training data about them. We design an algorithm based on the Hierarchical Dirichlet Process and the one-class Support Vector Machine to recognize abnormal activities when the training data is scarce. Finally, when we need to deploy the activity recognition systems in the real-world, it is impractical for us to collect enough training data for different activity recognition scenarios, especially when we need to collect training data for different persons and even for different actions. To solve this problem, we've developed an activity recognition framework based on transfer learning which borrows useful information from previously collected and learned activity recognition domains and then re-use such information into the new target activity recognition domain. Furthermore, we've conducted extensive experiments to demonstrate the effectiveness of our proposed approaches on real-world datasets collected from smart homes or sensor environments. We've also shown that our context-aware query classification algorithm could outperform state-of-the-art query classification approaches on real-world query engine search logs. At the end of this thesis, we discuss some possible directions and problems for future work and extensions.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In this chapter, we would introduce what we are aiming to discuss in this proposal - human activity recognition. In particular, we would provide an overview of this proposal and loosely define what activity recognition is. Next, we would show some applications researchers have developed with the aid of activity recognition techniques. Finally, we would discuss some challenges researchers have faced when they try to tackle activity recognition problems in the real-world.

1.1 Background

Understanding the activities of humans in the physical world has been an extensive research topic since early 1980s. Due to the development of microelectronics and computer systems, sensors and mobile devices with high computation power, small size and low cost become to emerge and now play a major role in our daily lives.

With the advent of such sensors and mobile devices, a natural question that arises is whether one could use the knowledge and information attained from such devices to infer what the subject has been doing or is about to do. Such a research question falls to the dimension of activity recognition, intention recognition or goal recognition. Loosely speaking, activity recognition aims to recognize the actions and goals of one or more agents from a series of observations on the agents' actions and the environmental conditions.

To understand activity recognition better, consider the following scenario. An elderly man wakes up at dawn in his small studio apartment, where he stays alone. He lights the stove to make a pot of tea, switches on the toaster oven, and takes some bread and jelly from the cupboard. After taking his morning medication, a computer-generated voice gently reminds him to turn off the toaster. Later that day, his daughter accesses a secure website where she scans a check-list, which was created by a sensor network in her father's apartment. She finds that her father is eating normally, taking his medicine on schedule, and continuing to manage his daily life on his own. That information puts her mind at ease [1]

Many different applications have been studied by researchers in activity recognition; examples include assisting the sick and disabled. For example, Pollack et al. [124] show that by automatically monitoring human activities, home-based rehabilitation can be provided for people suffering from traumatic brain injuries. One can find applications ranging from security-related applications and logistics support to location-based services. Due to its many-faceted nature, different fields may refer to activity recognition as plan recognition, goal recognition, intent recognition, behavior recognition, location estimation and location-based services.

1.2 Specific Problems and Challenges

Researchers might consider activity recognition as a simple application of machine learning, where sensor readings are directed to a machine learning algorithm as input and then we would expect the human activities we recognize are provided as output. However, despite the abundant amount of work that exist in this research area, there is still a significant gap between current research and successful deployment of activity recognition systems. In this thesis, we will discuss some of these problems as well as our proposed solutions to them.

- The relationships between human activities are quite complex. Human activities can form a hierarchy or other structures. One may pursue several goals at a time or carry out a number of different primitive activities serving a high-level purpose. Models without taking these characteristics into consideration will easily fail to recognize these multiple activities which are taking place. We will propose our solution to this challenge in Chapter 3.

- The efficiency of human activity recognition algorithms must be taken into

[1] http://en.wikipedia.org/wiki/Activity_recognition

consideration. Nowadays, graphical models like HMM or CRF become popular amongst human activity recognition approaches. Such graphical models often have high complexity in nature and cannot scale up easily, which forbids these solutions to be extended to large scale, like human activity recognition in urban settings. We will discuss our algorithm running time in each experiment sections of our proposed solutions.

- The amount of training data required by the activity recognition approaches should not be too much. Many activities (like abnormal activities) won't be able to have a large amount of training data collected before we build their corresponding classifiers. Furthermore, if we collect training data of a specific subject in one activity recognition domain, it won't be able to directly apply such training data onto another activity recognition domain. Such a problem calls for other machine learning techniques like transfer learning for us to borrow and re-use useful knowledge in the source domain and apply them in the target domain. We will propose our solution to this challenge in Chapter 4 and 5.

1.3 Main Contribution

In this thesis, we focus on human activity recognition and propose solutions to the abovementioned challenges of human activity recognition problems. The main contribution in this thesis are as follows:

- **Concurrent and Interleaving Activity Recognition**: We first point out the necessity of considering concurrent and interleaving activities in solving the activity recognition problem and then propose a two-step approach to solve this problem. Firstly, we construct a skip-chain Conditional Random Field model for each activity we are going to recognize. In other words, we are reducing a multi-class classification problem into a series of binary classification problems. Next, we construct a goal correlation graph of the multiple activities we consider and use such a goal correlation graph to regularize the initial probabilities we've recognized from the graphical model. We'll show in our experiments that such a model is *accurate, scalable* and can successfully recognize both concurrent and interleaving activities in real-world sensor datasets.

- **Abnormal Activity Recognition**: We propose a three-phase approach for recognizing abnormal activities, which are rare to happen and difficult to collect enough training data. We first use real-world experiments to demonstrate that traditional state-based models like HMM use a trial-and-error approach to find the suitable number of hidden states, which is impossible to carry out in real-world situations. Hence, we've applied a nonparametric Bayesian approach and used Hierarchical Dirichlet Process Hidden Markov Model, which supports an infinite number of states to solve this problem. Next, we've used Fisher Kernel and One Class Support Vector Machine to estimate a model for the normal activities. Finally, an adaptation method is adopted to automatically recognize whether new types of abnormal activities have been found.

- **Transfer Learning for Activity Recognition**: We first introduce one of our earlier method to solve this problem by using Web knowledge as a bridge and create pseudotraining data in the target domain. Next, we introduce our feature mapping method by trying to align different sensors in two domains and create a "dictionary" for one activity in the source domain to be interpreted as another activity in the target domain. We also discuss how to perform label transfer on state-based models in two domains.

1.4 Organization

The thesis is organized as follows:

- **Chapter 2**: In this chapter, we present a survey of related work on a series of problems in the area of activity recognition. We start by introducing various kinds of sensors now used in activity recognition, followed by the descriptions of some state-of-the-art learning methods that are used in activity recognition.

Finally, we discuss some other important topics that are relevant to our work and important to activity recognition.

- **Chapter 3**: In this chapter, we study the case of activity recognition where the relationship of activities can be complex. Concurrent and interleaving activities are common cases of real-world activity relationships and it is necessary to take these relationships into consideration when a real-world activity recognition system is being deployed. We design an algorithm to recognize activities where concurrent and interleaving relationships are considered and such an algorithm is naturally extended to the problem of context-aware query classification, where we can also borrow the knowledge of interleaving queries to help predict the categories of Web queries [51, 23, 50].

- **Chapter 4**: In this chapter, we study the problem of abnormal activity recognition. Anomaly detection is an important research problem in the field of data mining and machine learning. But abnormal activity recognition has some special features which forbids us from directly using other anomaly detection methods. We design an algorithm based on the Hierarchical Dirichlet Process Hidden Markov Model (HDP-HMM) and use an adaptation approach to further reduce the false negative rate obtained from the recognizer [169, 53].

- **Chapter 5**: In this chapter, we study how we could apply the idea of transfer learning into activity recognition. The necessity of applying such a learning framework has been mentioned above since we need to re-use the training data and useful information we had collected and learned from other activity recognition domains as much as possible. In this chapter, we first describe a basic approach of creating pseudo training instances based on transferring information from open Web knowledge and then extend such a framework to feature-level transfer [171, 52, 55].

- **Chapter 6**: Finally in this chapter, we conclude this thesis, summarize our major contributions and findings and point out some directions and problems for future study.

CHAPTER 2

RELATED WORK

2.1 Sensing and Modalities

In this section, we would provide a survey on how different “sensing ability” is provided via different kinds of sensors in previous activity recognition research work. Before we start, we use Figure 2.1 ([123]) to illustrate the range of sensor technologies that are being investigated for activity recognition. Researchers are exploring both environmental sensors as well as biosensors. The former class include motion detectors and RFID readers that can determine a person’s location, contact switches on cabinets and refrigerator doors that indicate whether they have been opened, pressure sensors indicating whether a person is sitting in a bed or chair, and thermometers indicating whether a stove has been turned on. Biosensors are generally worn by a person to measure vital signs such as heart rate and body temperature. This range of sensors can be used to determine where a person is and what household objects he or she has used, as well as to get a general sense of her activity level.

In the following, we will categorize the sensing technologies used in activity recognition into mainly three categories: using complex sensors like cameras that are versatile but sensitive, wearable sensors, corresponding to the biosensors noted in Figure 2.1 and object-based sensors that are similar to the role environmental sensors play in an activity recognition system. Finally, we would also briefly discuss a new sensing approach called Infrastructure Mediated Sensing that has recently been developed.

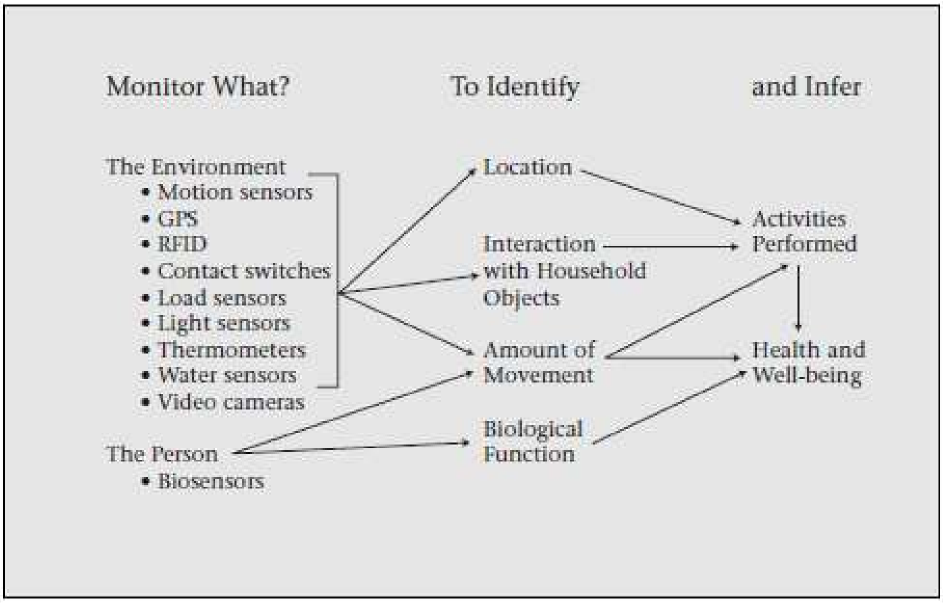


Figure 2.1: Sensors for activity monitoring [123]

2.1.1 Complex Sensors

One approach to activity recognition is to use a small number of versatile sensors such as cameras. Systems such as those proposed by the EasyLiving project at Microsoft made extensive use of vision to infer context about the environment [19]. Other projects have explored using vision for recognizing activities, tracking people in the environment [61], and recognizing user gestures. The advantage of using cameras is that cameras can be extremely versatile. However, the disadvantage is that the vision-based activity recognition algorithm would be sensitive to lighting changes, cluttering of objects as well as shadows and occlusion.

Therefore, such a difficulty of using vision-based complex sensors has led to an increasing interest in activity recognition using wearable sensors and object based sensors as we shall discuss in the next two subsections.

2.1.2 Wearable Sensors

Sensors worn on the body such as heart rate monitors are suggested as a type of health monitoring as well as health feedback to be used in the homes of the future. Skin conductance monitors have been used in attempts to infer the user's emotional state in [138]. However, although different kinds of wearable sensors are used and proposed for different kinds of activity or intention recognition, probably the most widely used kind of wearable sensors for activity recognition is the accelerometer.

So what is an accelerometer? From Wikipedia, an accelerometer "measures the acceleration it experiences relative to freefall". Single- and multi-axis models are available to detect magnitude and direction of the acceleration as a vector quantity, and can be used to sense orientation, vibration and shock. Accelerometers are increasingly present in portable electronic devices and video game controllers.

The usage of accelerometers in the field of activity recognition can be found in a number of different applications. Accelerometers have been suggested for gesture recognition [40, 18], detecting activities [5, 87], and to support other applications such as physical fitness. One particular example is the "Nike + iPod" system which uses a variant of an accelerometer to sync music entertainment with exercise [1].

However, one problem with these sensors is that it is easy to confuse between activities with similar body movements. Consider the case where we infer the activity "eating dinner" by using an accelerometer wearing around your wrist. Therefore, such an inference procedure is very likely to confuse between activities like "eating dinner" or "eating snacks while you are watching TV". To solve this problem, other sensors related to direct object usage are used as well.

2.1.3 Object-Based Sensors

Object-based sensors are sensors attached to household objects of interest. Since object-based sensors are attached to objects rather than to persons, a large number of sensors which are less versatile must be used. These approaches include using reed switches [147, 162], RFID tags [119], accelerometers [6], and other movement-detection services [149].

RFID (Radio-Frequency Identification) Tags are particular useful in the state-of-the-art activity recognition systems since it can provide relatively accurate feedback about whether an object is being used. RFID is the use of an object applied to

[1] http://www.apple.com/ipod/nike/

or incorporated into a product, animal, or person for the purpose of identification and tracking using radio waves. Some tags can be read from several meters away beyond the line of sight of the reader. Most RFID tags contain at least two parts, one is an integrated circuit for storing and processing information, modulating and demodulating a radio-frequency (RF) signal and other specialized function while the other is an antenna for receiving and transmitting signal.

The basic idea of using object-based sensors for activity recognition is to provide a sequence of object usage events and use such a sequence to perform activity recognition. In real-world situations, many activities have key objects, for example, when we make a coffee, we must use coffee beans, coffee pots or sugar. When we observe such a sequence or set of object usage in the activity recognition system, it is possible that the person we are monitoring is making coffee.

Deploying many object-based sensors in a laboratory setting to study the behavior of humans is one of the state-of-the-art approaches for performing activity recognition. A number of institutions are collecting data from such laboratory settings, the notable of which is the MIT PlaceLab, with different kinds of sensors attached to household objects of interest and volunteers perform different kinds of household activities for varied periods of time [148, 96, 63, 62].

2.1.4 Combining Information from Different Sensors

It is natural to ask the question about whether it's possible to improve the activity recognition performance by combining the sensory input from different kinds of sensors. There are also a number of previous research work on this topic.

Depending on the type of activity, recognition performance can be improved by using the same type of sensor at multiple body locations (e.g. multiple accelerometers as used by [80, 5, 57], employing networks of heterogeneous sensors (e.g. [69, 75]) or integrating a variety of sensors on a single device (e.g. [32]).

Combining two or more complementary types of sensor data can also help in recognizing activities, e.g. by combining motion and audio data (e.g. [98, 75, 33]), motion and proximity data [136], motion and location data [140], or motion-data

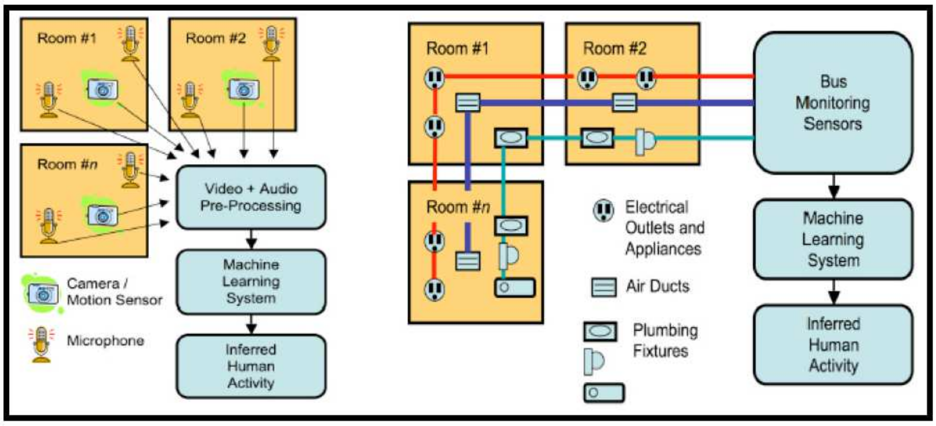


Figure 2.2: The distributed direct sensing (DDS) approach for activity detection and classification (left). The infrastructure mediated sensing approach for activity detection and classification (right) [114]

and readings from wearable RFID tag readers [160]. The latter is an example of combining wearable sensors with an instrumented environment (in this case RFID-tagged objects). A similar approach is taken by [137], who combine data from wearable accelerometers and environmental infra-read sensors.

2.1.5 Infrastructure Mediated Sensing

In most recent years, a new trend of sensing approaches is drawing increasing interest by researchers especially in the pervasive computing community. Such a new sensing approach is called IMS (Infrastructure Mediated Sensing) [114, 115, 139, 77, 116, 117].

As we discuss in the previous sections, a wide variety of sensors must be deployed in a living laboratory environment to observe a wide variety of variables which are then classified and used as proxies for ordinary human activities. Such an approach, which is dominant for over ten years in the field of activity recognition, is called *DDS* (Distributed Direct Sensing) in contrast to IMS. In general, because large numbers of sensors are distributed throughout the environment, special networking infrastructure (either wired or wireless) is installed in the living laboratory to collect sensor data and transport it from the sensor location to special computation resources.

However, it is easy for us to think about several drawbacks the traditional DDS approach would have. The first drawback is that DDS approach can be extremely expensive, since a large number of sensors must be deployed and sensors can be expensive. The second drawback is that DDS approach can be extremely obtrusive. When humans are required to wear on-body sensors to perform activity recognition, it would make them feel uneasy since they "look different from ordinary people". If a large number of object-based sensors are attached to the household objects, it may destroy the original appearance of the objects or make them inconvenient to be used. When wired networks are installed in the house to support transition of data streams from sensors to computation resources, such a network installed would affect the aesthetic feature of the room. In sum, DDS approaches are extremely important and valuable for developing activity recognition in a controlled setting. However, such an approach is not easy for putting into real-world or commercial activity recognition systems. They may either be too costly or too complex to permit widespread deployment.

Figure 2.2 shows the important distinguishing features between DDS and IMS. DDS involves the installation of a new sensing infrastructure into the home, which directly senses the presence, motion or activities of its residents through sensors that are physically located in each space where activity is occurring. However, IMS leverages existing home structure, such as the plumbing or electrical systems, to mediate the transduction of events. Thus, one of the aims of IMS is to reduce the installation and maintenance barriers to adoption by reducing the cost and complexity of deploying and maintaining the activity sensing infrastructure.

Some preliminary applications have been developed using IMS approaches. In [115], an approach for whole-house gross movement and room transition detection through sensing at only one point in the home is proposed. Disruptions in airflow, caused by human inter-room movement result in static pressure changes in the HVAC air handler unit, which is particularly apparent for room-to-room transitions and door open/close events involving full or partial blockage of doorways and thresholds. Such a method requires the installation of only a single sensing unit (*i.e.* an instrumented air filter) connected to an embedded or personal computer to perform the classification function. In [117], an indoor localization system that uses the residential powerline is proposed. Powerline positioning is an inexpensive technique that uses fingerprinting of multiple tones transmitted along the powerline to achieve sub-room-level localization. Furthermore, in [116], a single plug-in sensor is used to detect a variety of electrical events throughout the home. The sensor detects the electrical noise on residential power lines created by the abrupt switching of electrical devices and the noise created by certain devices while in operation. Most recently, in [139], a wideband approach to powerline positioning (WPLP) that injects up to 44 different frequencies into the powerline is proposed. This WPLP approach improves upon overall positioning accuracy, demonstrates greatly improved temporal stability and has the added advantage of working in commercial indoor spaces.

2.2 Learning and Inference

In this section, we would briefly review some important works mainly focusing on the *learning and inference component* of activity recognition system. We would discuss both the logic-based and learning-based approaches used to solve the problem, whereas in deterministic approaches, activity recognition is more commonly known as the “plan recognition” problem. In learning approaches, we would discuss the three major learning mechanism in machine learning, namely supervised learning, semi-supervised learning and unsupervised learning approaches. Finally, we would also briefly discuss some feature selection methods in activity recognition systems.

2.2.1 Logic-based Approaches

When solving the problem of activity recognition via logic-based approaches, the problem of activity recognition is more usually referred to as *plan recognition*. In the literature, the term, *activity recognition* is often used to describe the problem of segmenting and classifying low-level data gathered by cameras or wireless sensors into a description of the activity performed and plan recognition is generally used to describe the mapping of sequences of atomic actions to high-level plans.

Much of the early work on plan recognition relied on logical methods by either viewing plan recognition as a specialized type of hypothetical [29] or unsound reasoning. However, Kautzs event hierarchy framework [73] combined deductive reasoning with a specific set of assumptions. Two important assumptions that he introduced are exhaustiveness assumption and minimum cardinality. The exhaustiveness assumption specifies that the world is limited to the known types of events (plans), hence it is possible to determine that a particular event has taken place by eliminating all other possibilities. The minimum cardinality assumption follows the principle of parsimony, only assuming the minimum number of events that explain the observations. These two assumptions are either explicitly or implicitly made by most current plan recognition frameworks.

Specialized search techniques have been developed to reduce the time required for plan recognition; for instance, RESC (Real-Time Situated Commitments) is a real-time approach to tracking the operator hierarchies of a Soar agent [146]. RESC uses information from the current world state to determine the validity of the operator hierarchies, commits to a a single interpretation, and backtracks in case of mistaken interpretation; this approach has the advantage that it can be used in real-time opponent modeling and handles reactive behaviors well [146]. Rather than committing to a single interpretation, the RESL algorithm marks every plan whose observations matches expectations and maintains it as a possible hypothesis [71]. Avrahami-Zilberbrand and Kaminka later developed a single-agent plan recognition algorithm [3] which improves on RESL in several ways: (1) more efficient observation matching through the use of feature decision trees; (2) the use of temporal structure to rule out inconsistent hypotheses for current state queries.

2.2.2 Learning-based Approaches

Supervised Learning Approaches

Traditional Classifiers Although it is normal for us to use temporal classifiers to perform activity recognition, sometimes non-temporal classification methods, such as support vector machines or decision trees, are also used to classify sensor data, *e.g.* in [5, 130]. In particular, researchers who study the feasibility of using accelerometer data to perform activity recognition often resort to such non-temporal models to perform simple yet still effective activity recognition. Non-temporal approaches often use features such as means, variances and Fourier transforms computed over windows of the data to perform ad-hoc temporal smoothing.

Bayesian Networks A Bayesian network is a directed ayclic graph whose nodes represent random variables and whose edges indicate direct influence between nodes. Bayesian networks provide a compact way to represent the joint distributions of the variables by capturing the conditional independence among variables. Since Bayesian networks have been successfully applied in many areas for inference under uncertainty, it is natural to choose Bayesian networks for human behavior modeling. Bayesian networks have many strengths: they are expressive, flexible, and many off-the-shelf learning and inference algorithms have been developed.

Charniak and Goldman's model [28] for plan recognition is one of the earliest work using Bayesian Networks. They manually translated activity knowledge into an associative network. Then they use a number of rules to automatically convert an associative network to the corresponding Bayesian Network. Their approach is bottom-up by considering the activity hypothesis compatible with the observations and tries to keep the network as small as possible. Therefore they construct the Bayesian Network dynamically to incorporate the latest evidence. One specific drawback of their model is that they rely on general Bayesian Network inference models and therefore specific relationships among the variables cannot be utilized. Another drawback is that temporal constraints are unable to be represented in their model.

Huber presented a top-down approach in [56] where the Bayesian networks are constructed from the plan library before receiving any observations. The plan language they employed, called Procedural Reasoning System (PRS), is expressive enough to handle many types of relations such as explicit sequencing, exclusive branches, iteration and context influence. Similarly, they also design rules that convert PRS into Bayesian networks automatically. In their model, using generic inference algorithms is unable to scale well in large domains.

Stochastic Context Free Grammars Stochastic Context Free Grammars (SCFGs) are a powerful tool to model fairly simple and relatively predictable human activities. This framework was first used for the problem of activity recognition in [64], where it was tested in a parking lot scenario using tracking information of the various vehicles in the scene. More recent work along this line was done in [105], which leverages high-level expectations of different events in an SCFG framework for the purposes of activity recognition.

One of the main shortcomings of SCFGs is that being an extension of a fundamentally grammar-driven framework of Context Free Grammars, they must be explicitly modeled. This makes their applicability limited to relatively simple activities. Moreover, while SCFGs have been augmented to become stochastic, they still can only make hard decisions about choosing the next production rule. Only when we have the next production rule can the notion of probability come into play. Therefore, all potential subsequent tokens need to be indicated explicitly by production rules. This is in contrast with DBNs, in which by default, the subsequent state space is any combination of the hidden variables and does not need to be pointed out explicitly. Because of this reason, when input is noisy, insertion and deletion errors can become a substantial problem for SCFGs.

Hidden Markov Models A Hidden Markov Model is a statistical model in which the system being modeled is assumed to be a Markov process with unobserved state. It can also be considered as the simplest dynamic Bayesian Network. Hidden Markov models [10, 126] are the canonical models for sequential or time series classification. Hidden Markov models have been applied extensively in the domain of activity recognition. Han and Veloso used hidden Markov models, built using hand selected observations, to classify behaviors in the Small Size League of RoboCup [48]. In a less closely related multi-agent military domain, Sukthankar and Sycara used a hidden Markov model to classify team behaviors. In particular, they used spatial relationships between the agents on teach team as well as the spatial relationships between agents and landmarks in the environment as the input features to the HMM [142].

However, generative models, such as HMMs, cannot robustly incorporate complex, non-independent features of their inputs. The complex spatial relationships that we use for activity recognition are non-independent, which is one of the major motivations for using conditional models as we would show later in this section. Lester et al. used a hybrid model to avoid this issue and obtain the benefits of discriminative training as well as the temporal smoothing that comes from using an HMM [87]. They used boosting to discriminatively train ensembles for human activity recognition from data collected by a wearable sensor board. They used ensembles of boosted decision stumps to compute marginal probabilities over the activity states and passed these marginal probabilities to a hidden Markov model for temporal smoothing.

Augmented Markov Models Augmented Markov models use the behavior or activity labels of agents as a representational substrate to model the environment [45, 46]. On a practical level, augmented Markov models are similar to HMMs with noise-free observations. Each state emits a single symbol, but the same symbol may be emitted by multiple states. Augmented Markov models maintain auxilliary statistics about state duration and the frequency of link traverals on a per state or per transition basis. They provide a first-order Markov representation that allows for reasoning about higher order Markovian behavior. Augmented Markov models use the joint role of a team of humans as input to classify states of the world. They abstract away the noise and uncertainty of real sensors by operating at a higher level of abstraction.

Dynamic Bayesian Networks A dynamic Bayesian network [108] is a Bayesian network that represents sequences of variables. These sequences are often time-series or sequences of symbols. They extend Bayes nets [122] to temporal domains. They can be viewed as a superset of hidden Markov models that allow for a much richer set of relationships among random variables. For example, they allow for factored state, e.g. factorial hidden Markov models [44], and can represent complex relationships, e.g. hierarchy, among model variables.

DBNs are popular models for activity recognition from sensor data. For example, Patterson et al. used a DBN to track human subjects on a map using noisy GPS traces [120]. The representational power of DBNs allowed them to explicitly model factors such as bus stop and parking lot locations while simultaneously estimating the position and mode of transportation for the subject. In follow-on work, Liao et al. increased the depth of the hierarchy to include factors such as the end destination of the user. They used a hierarchical hidden Markov model, based on the abstract hidden Markov model of Bui et al. [22], to model the users position and to detect deviation from his or her normal routines [95, 92]. In general, adding hierarchy improves classification accuracy, e.g. [41, 110], although it results in models where exact inference is intractable. In particular, methods such loopy belief propagation [109] or Rao-Blackwellized particle filters [37] are required to perform approximate inference. Raj et al. describe inference via a Rao-Blackwellized partical filter for activity recognition using a hierarchical DBN in detail [128].

Conditional Random Fields A Conditional Random Field is a discriminative graphical model, which focuses on modeling the conditional distribution over the unobserved states given the observations [81].

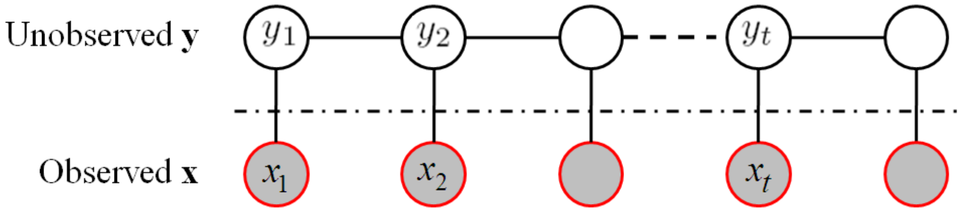


Figure 2.3: Linear-chain Conditional Random Field.

As shown in Figure 3.4, a linear-chain CRF defines the probability of a label sequence $\mathbf{y}$ given an input sequence $\mathbf{x}$ is:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_t \psi_t(y_t, y_{t-1}, \mathbf{x}), \tag{2.1}$$

where $Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_t \psi_t(y_t, y_{t-1}, \mathbf{x})$ is a normalization factor. Potential functions ψ_t describe the linear-chain transitions, and are defined as:

$$\psi_t\left(y_t, y_{t-1}, \mathbf{x}\right) = \exp\left(\sum_k \lambda_k f_k\left(y_t, y_{t-1}, \mathbf{x}, t\right)\right), \tag{2.2}$$

where f_k's are the feature functions and λ_k's are the parameters of the linear-chain template. Given training data (from one domain) $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{T}$, the objective of training a linear-chain CRF is to find a set of parameters $\Lambda = \{\lambda_k\}$ that maximize the conditional log-likelihood:

$$L(\Lambda) = \sum_{i=1}^{T} \log p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}). \tag{2.3}$$

By deriving the parameter Λ, one can infer the labels $\mathbf{y}^*$ for the test data $\mathbf{x}$ as $\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}, \Lambda)$.

As newer models, conditional random fields have received less attention than HMMs or DBNs, but they are used for activity recognition as well as in related domains that require rich, continuous features such as image segmentation [79], object recognition [125], and gesture recognition [161]. In activity recognition domains, Truyen et al. found that discriminative models such as CRFs and maximum entropy Markov models [104] significantly outperform generative models like HMMs for recognizing human activities from images in videos [153].

In closely related work, Liao et al. used a hierarchical conditional random field to recognition human activities from GPS traces [94]. The key contribution of their work is that they show how to build hierarchical CRFs for activity recognition. They perform a first pass of classification over the data in order to predict activity labels for the sequence. They use the predicted activities from the first pass to identify significant places, such as home, work, and friends' houses, for the person whose activities are being labeled. They then perform a second round of prediction in the hierarchical model, which now includes the significant places as unobserved random variables. They show that the label sequence from the hierarchical model is more accurate than the label predictions from the first pass in the flat model, even though the flat model was used to determine the structure of the hierarchical model. When training their models, they used pseudo-likelihood for parameter estimation. For inference in the complete model, where exact inference is not tractable due to cyclic structure, they used loopy belief propagation to predict labels over test sequences.

Vail et al. [155] analyzed the differences in performance between the discriminatively trained CRF and the generative HMM by using data from a simulated robot tag domain, since it is multi-agent and produces complex interactions between observations. They found that discriminatively trained CRF performs as well as or better than an HMM even when the model features do not violate the independence assumptions of the HMM. Hu et al. [51] used a skip-chain conditional random field model [143] to help recognize the interleaving activities in a real-world activity recognition scenario. They also modeled the concurrent activities as a quadratic programming problem to learn a relation matrix between different activities.

Relational Models Relational Markov networks are an extension of conditional random fields that use a relational language to specify the cliques in the graph [150]. Like CRFs, RMNs are undirected graphical models and therefore their clique potentials are the functions of the variables that comprise the cliques in the graph. Rather than using a fixed functional form for their cliques, as in CRFs, which specify their clique potentials as $\phi_t(y_{t-1}, y_t, X)$; RMNs define their cliques using a collection of schema or templates. Schema are defined in terms of entities and attributes. For example, in natural language domains, e.g. [2], individual documents might be the entities and a possible attribute is refers to. For a given data set, a Markov random field is unrolled or instantiated according to the rules embodied in the schema and the usual inference techniques are applied to the unrolled graph. Typically, this graph is densely interconnected and exact inference is intractable [150].

Liao et al. used relational Markov networks for human activity recognition [93]. In their earlier work with CRFs [95], they dynamically created a model structure based on a preliminary classification using a flat model. In this work, they formally defined the clique structures to include notions such as significant places using relational Markov networks.

Non-parametric Models Non-parametric methods avoid the need to define features by using the data itself as the model representation. Lenser and Veloso used non-parametric statistics to detect discrete environmental states from streams of robot sensor data [86]. They used an auto-regressive HMM with a rich, non-parametric observation model $p(x_t|x_{t-1}, y_t)$, where x_{t-1} and x_t represent the current and previous observations (or model emissions) and y_t represents the label at the current time t.They used raw sensor data from robots to detect and adapt to properties of the environment such as the type of carpet under a legged robot and to changes in lighting conditions. Kernel conditional random fields are a non-parametric CRF representation [82] and non-parametric methods are applicable to activity recognition with CRFs.

Semi-supervised Learning Approaches

In real-world applications, unlabeled sensor traces are relatively easy to obtain whereas labeled examples are expensive and tedious to collect. Therefore, semi-supervised activity recognition is an important topic these days. In [99], a semi-supervised training method for parameter estimation and feature selection in conditional random fields (CRFs) is proposed. The semi-supervised algorithm is an extension to the Virtual Evidence Boosting (VEB) algorithm [90] for the feature selection and parameter learning. The objective function of sVEB (semi-supervised Virtual Evidence Boosting) algorithm combines the unlabeled conditional entropy with labeled conditional pseudo-likelihood. The objective function L_{sVEB} is as follows:

$$L_{sVEB} = \sum_{i=1}^{N} \log p(y_i|\mathbf{ve_i}) + \alpha \sum_{i=N+1}^{M} \sum_{y_i'} p(y_i'|\mathbf{ve_i}) \log p(y_i'|\mathbf{ve_i})$$

Here $(i = 1 \ldots N)$ are labeled data and $(i = N + 1 \ldots M)$ are unlabeled data. The sVEB objective maximizes the conditional soft pseudo-likelihood of the labeled data and in addition also minimizes the conditional entropy over unlabeled data.

Recently, Stikic *et al.* also studied the performance of exploiting semi-supervised and active learning approaches in activity recognition. They analyzed the performance of self-training [27] as well as co-training [14] for data from motion sensors and they also showed it is possible to apply co-training for recognition of activities when using two independent sources of information, namely on-body accelerometers and infra-red motion sensors.

Unsupervised Learning Approaches

An unsupervised activity recognition framework usually transforms the unlabeled sensory input into features and then model the data using some kind of density estimation or clustering algorithm. During clustering, each data point is assigned to one or more groups of points that are relatively close to each other with respect to a specific distance measure.

[94, 121] uses unsupervised learning schemes based on graphical models while the focus is on inferring transportation modes (bus, car, walking) and destination goals of the user. Specifically, in [121], a hierarchical activity model representing a person's outdoor activities is proposed. The parameters in the model are estimated in an unsupervised manner, where two rounds of EM (Expectation-Maximization) procedure were used to estimate the value of each parameter. In [106], the problem of "activity discovery" is proposed, which means the unsupervised identification and modeling of human actions embedded in a larger sensor stream. They attempt to discover motifs by combining discrete string matching techniques with continuous HMM classifiers in acceleration data.

More recently, [58] proposes a new method to recognize daily routines as a probabilistic combination of activity patterns. The automatic discovery of such activity patterns is enabled by the use of topic models, more specifically, by the use of Latent Dirichlet Allocation (LDA) [11]. A stream of sensor data is converted into a series of documents consisting of sets of discrete activity labels. These sets are then mined for common topics.

Transfer Learning Approaches

Transfer Learning [112] in activity recognition is a quite new topic that has not been well studied, even at a preliminary stage. Previous machine learning paradigms like supervised or semi-supervised learning work well under the assumption that the training data and the test data are drawn from the same distribution and the same feature space. In many real-world applications, it is expensive the recollect the needed training data and retrain the models when the distribution or the feature space changes. Such a story remains the same for activity recognition, as it is extremely costly and tedious to collect labeled data for activity recognition. The study of transfer learning is motivated by the fact that people can always apply knowledge learnt previously to learn solve new problems faster, for example, learning to recognize apples might help learning to recognize pears, etc.

How does transfer learning apply to activity recognition? Different people have intrinsic differences even when performing the same activities. Therefore, when we have trained an activity recognition model to recognize activities for one person, are we able to use the knowledge from such a model to recognize the same set of activities for another person? Such a scenario corresponds to the case in transfer learning where the “distribution” changes.

Another possible scenario for transfer learning is that, when one has built an activity recognizer for recognizing a set of activities A, is it possible to adapt the knowledge for building activity recognition systems to recognize another set of activities B? Such a scenario corresponds to the case in transfer learning where the label space changes.

Yet another possible scenario for transfer learning in activity recognition is that, consider two different living laboratories with different kinds of sensors. Can we use the training data from one lab and transfer useful knowledge and use them to help predict activities conducted in the other laboratory? Notice that here, since we have different kinds of sensors, such a scenario actually corresponds to the case in transfer learning where the “feature space” changes.

With more and more research work available in the area of transfer learning, there also have been some papers which aim to solve some activity recognition related problems that also fall into the category of transfer learning. In this subsection, we provide a brief overview of these papers and also provide a coarse categorization of these papers.

- Transferring Across Devices: One important component of activity recognition is localization, where locations are used as key factors which could imply the subject's activities. In [172], the authors propose a multi-task learning algorithm to solve the multi-device indoor localization problem. One major drawback of previous localization systems lies in the assumption that the collected signal distribution remains the same across different devices. However, empirical studies in [172] show that this is often not the case and that the localization accuracy would be greatly affected. In [172], the multi-device indoor localization problem was formulated as an optimization problem and an alternating optimization approach was employed to solve the problem.

- Transferring Across Time: Another important dimension by which transfer learning can be carried out is the time dimension. Similarly, signal data distributions also vary from time to time and if we directly apply a model learned in the previous month to try to perform recognition tasks in the current month, variations in signal distribution would degrade the algorithm performance. To solve this problem, [173] proposed a semi-supervised Hidden Markov Model to transfer the learned model from one time period to another.

- Transferring Across Space: Space is another dimension where transfer learning can be used. Since most localization models are supervised learning approaches and would require us to collect labeled data across an entire building if we want to do indoor localization in that building. Therefore, transfer learning would greatly alleviate the calibration efforts one needs to perform before a learning model is trained. In [111], the authors propose a novel approach to learn localization models across space, where a mapping function between the signal space and the location space is learned by solving an optimization problem based on manifold learning techniques.

- Transferring Across Sensor Networks: In activity recognition, one possible scenario when one tries to apply transfer learning is to transfer knowledge between different sensor networks where the layout and the functionality of the sensors may not necessarily be the same. Earlier research work [159] on this topic attempted to transfer the knowledge between two houses which have different sensor network layouts. However, the correspondence between different sensors was defined manually, which greatly reduced the learning difficulty but limited the applicability of such systems. In [157], the authors tried to solve the activity recognition also via a transfer learning approach. In this paper, the authors can perform activity recognition across different sensor networks. However, their algorithm is based on the usage of a *meta-feature space*, which are features that describe the properties of the actual features. Each sensor is described by one or more meta features, for example, a sensor on the microwave might have one meta feature describing the sensor is located in the kitchen, and another that the sensor is attached to a heating device. The limitations of the approach described in [157] is that the meta-feature space needs to be manually constructed, and it has to be the same kind of sensors in order to have the common meta-feature space. Such limitations also avoid the transfer between different kinds of sensors or in very different room layouts.

- Transferring Across Activity Types: This category refers to our work in this thesis since we are aiming to transfer knowledge between different activity label spaces. Based on the above categorization, we can see that our work sets a different dimension from all previous research work on activity recognition that exploit some transfer learning ideas.

2.2.3 Feature Selection in Activity Recognition

Feature selection is also an important step in activity recognition and also an important research topic in machine learning. In a survey related to feature selection, one of the most important advice is to build domain-specific features whenever possible [13, 47, 76].

Here in the problem of activity recognition, spatial features which capture the relationships between agents or important agents in the environment can be especially useful [142]. Huynh and Schiele used features such as fast Fourier transform coefficients computed over windows of accelerometer data to recognize human activities such as walking or riding the bus [59]. They found that while they could pick out a general class of features, e.g. FFT coefficients, as important for the classification, individual activities depended on different scales of those features.

In [154], Vail *et al.* explored feature selection in conditional random fields for activity recognition to choose which features should be included in the final model. They compared two feature selection methods, grafting, a greedy forward-selection strategy and l_1 regularization, which simultaneously smoothes the model and selects a subset of the features.

About feature selection in conditional random fields, one important algorithm is the VEB algorithm (Virtual Evidence Boosting) [90], which is an extension to McCallum's feature induction algorithm [103] that flexibly addresses continuous observations and provides additional smoothing during training. In VEB, a CRF is cut into individual patches, as done in maximum pseudo-likelihood and these patches are used as training instances for boosting. They extend the standard boosting algorithm to handle input features that are either virtual evidences in the form of likelihood values or deterministic quantities. The key difference between VEB and MPL is that in VEB the neighbor labels are not treated as observed, but as *virtual evidences* or *beliefs*. Feature selection in VEB is naturally to find the "best" weak learner in each step, since a weak learner is a certain kind of combination of features.

2.3 Other Important Topics

In this section, we would discuss some important topics related to activity recognition which we feel would be difficult to be categorized into either progress in sensing component or progress in learning / inference component. In particular, we would discuss developments in multi-agent activity recognition, abnormal activity recognition, understanding the substructure of activities, and all that.

2.3.1 Multi-Agent Activity Recognition

In previous sections, we mainly focused on activity recognition based on observations or sensor readings of only one agent. However, understanding the activities of a team is also a very important research topic. Such a research problem is also called “team behavior recognition”.

Previous work on team behavior recognition has been primarily evaluated within athletic domains, including American football [60], basketball [9, 68], and Robocup soccer simulations [131]. Recognition for military air-combat scenarios has been examined in the context of event tracking [145] and teammate monitoring [71], A general framework for multi-agent activity recognition (Hierarchical Multiagent Markov Processes) [134] has been demonstrated for a single pair of humans moving around a laboratory.

To recognize athletic behaviors, researchers have exploited simple region-based [60] or distance-based [131] heuristics to build accurate, but domain-specific classifiers. For instance, based on the premise that all behaviors always occur on the same playing field with a known number of entities, it is often possible to divide the playing field into grids or typed regions (e.g., goal, scrimmage line) that can be used to classify player actions. The first part of Intille and Bobicks work [60] on football game annotation addresses the low-level problem of extracting accurate player trajectories from video sequences that include substantial camera movement (panning and zooming). Using knowledge of the football field geometry and markings, they warp the original distorted image sequence into a rectified one with the appearance of a stationary camera and background. Then, by exploiting closed-world knowledge of the objects in the scene, they create context-specific feature templates for tracking individual players. Play recognition [60] is performed on player trajectories using belief networks both to recognize agent actions from visual evidence (e.g., catching a pass) and to determine the temporal relations between actions (e.g. before, after, around). Jug et al. [68] used a similar framework for basketball analysis. These frameworks do not address the problem of dynamic team composition; all agents are committed to the execution of the single plan for the duration of the play.

Unlike Intille and Bobick, Riley does not attempt to produce annotated traces with sequences of Robocup team behaviors. Instead he extracts specific information, such as home areas, opponent positions during set-plays, and adversarial models [131], from logs of Robocup simulation league games to be used by a coach agent advising a team. For instance, information about opponent agent home areas are used as triggers for coaching advice and for doing formation-based marking, in which different team members are assigned to track members of the opposing team. Formations are generated from past logs in two phases: (1) fitting a minimal rectangle that explains the majority of each agents past positions; (2) modeling pair-wise distance correlations between agents by shifting the minimal rectangles.

Commentating from post-game log analysis has been demonstrated in the Robocup domain by the ISAAC system [127] using a combination of data mining and inductive learning; unsupervised data mining and knowledge discovery approaches have also been applied to the problem of recognizing strategic patterns from NBA basketball data [9]. However, the emphasis of the ISAAC system was to produce a high-level summary of game action, suitable for natural language summaries or debugging general team problems, rather than detailed play recognition of each teams actions.

In the 2005 Robocup coach competition, Kuhlmann et al. [78] demonstrated a game analysis approach in which the teams movement patterns were fitted to a parametric model of agent behavior. Patterns were scored according to their similarity to models learned from the pre-game logs. Recently, Beetz et al. [7] developed a system for matching soccer ball motions to different action models using decision trees. There has also been work on extending single-agent plan recognition frameworks [20, 146], both to create symbolic [145] and probabilistic [134] multi-agent plan recognition frameworks. These efforts have focused on the use of temporal behavior models and do not extensively utilize spatial information; such models have also been employed to detect teamwork failures [71] and agent-coordination termination [134].

Tambe [145] observed that the use of team models for multi-agent plan recognition can aid the process of tracking team activity by constraining the tracking search and eliminating the execution of large numbers of agent models. Unlike our algorithm, his system RESCteam does not maintain multiple hypotheses of team behavior nor does it precompute temporal and inter-agent constraints to prune the plan library. RESCteam executes a team model and invokes a minimal cost repair mechanism if the operator hierarchy selection or the role assignment causes match failure. RESCteam assumes that the subteams are either known in advance or detected solely based on agents proximity to each other; although this assumption is often valid, it is violated when a subteam that shares a common purpose becomes spatially separated (e.g., perimeter guarding in the MOUT domain). Saria and Mahadevan [134] demonstrate how a single-agent probabilistic plan recognition framework (the Abstract Hidden Markov Model) can be extended to reason about the joint policies of multiple cooperating agents. Their model, the Hierarchical Multiagent Markov Process, only represents two possible outcomes of agent coordination: (1) joint policies terminate when all of the participating agents complete their individual policies; (2) joint policies terminate when any of the agents completes its policy. This limited subset of execution outcomes is insufficient to model the common situation in which an agent abandons one plan in favor of a new plan, while the other agents continue executing the initial plan.

Unlike other groups, Kaminka and Tambes work [71] on plan recognition for socially attentive monitoring does not treat teams as automatically sharing a single plan. Due to message failure and sensor error, team members plans might diverge, in spite of sharing the same top level goal. The focus of their research on socially-attentive monitoring is the use of plan recognition by teammates to determine when other team members have inconsistent beliefs. Kaminka [70] developed the concept of team coherence, the ratio of total agents to the number of active plans, to represent the possibility of team coordination failures; he demonstrates that plan recognition can be used as part of scalable disagreement-detection system to detect the existence of incoherent team plans. In this thesis, we represent these teamwork failures as plan abandonment; if the agents reconcile their differences and resume coordination, it is detected as a new plan instance, rather than a continuation of a previous team plan.

2.3.2 Abnormal Activity Recognition

Consider the problem of detecting *abnormal activities*, where we follow the definition used in [167] and define “abnormal activities” as “activities that occur rarely and have not been expected in advance”. Such a problem may first appear to be very similar with the original activity recognition problem in principle. However, the problem of abnormal activity recognition is much harder than the original problem since such abnormal activities, by definition, rarely occur. This difficulty might become more significant during training phase since we lack such labeled sequences of abnormal activities. Up to now, most activity recognition algorithms [87] are systems based on state space-based machine learning models, which require a significant amount of training data in order to perform accurate and successful parameter estimation. Nevertheless, in abnormal activity recognition, such requirements often cannot be satisfied. Most previous research tried to tackle the abnormal activity recognition problem by also using state-space models [167], like HMM or DBN.

Previous approaches on abnormality detection problem range from the computer vision area [38, 169] to data mining areas of outlier detection [85]. In [167], they also detect a user’s abnormal activities from body-worn sensors. They propose a two-phase abnormality detection algorithm where a One-Class SVM is built on normal activities which help filter out most of the normal activities. The suspicious traces are then passed on to a collection of abnormal activity models adapted via KNLR (Kernel Nonlinear Logistic Regression) for further detection. However, before training the One-Class SVM, they need to transform the training traces that are of variable lengths into a set of fixed-length feature vectors. To accomplish this task, they trained M HMMs where M is the number of normal activities, and then the likelihood between each sensor reading and normal activity is used as the feature vector. One major drawback of this model is we need to specify the state number of HMMs when training, and such a number will affect the overall algorithm performance a lot as we will show in our experiment section. Thus, their algorithm may not be easy to use in real-world situations since it is hard for users to tune this parameter easily.

Our most recent work [54] also tries to tackle the abnormal activity recognition problem. We aim to solve the *abnormal activity recognition problem* via a three-phased approach. We first apply the Hierarchical Dirichlet Process Hidden Markov Model (HDP-HMM), which has an infinite number of states and can automatically decide the optimal number of states. Then, we incorporate a Fisher kernel into our model and apply a One-Class Support Vector Machine (OCSVM) to filter out the normal activities. Finally, we derive our abnormal activity model in an unsupervised manner. In [54], two additional contributions, besides providing an effective and efficient algorithm for solving the *abnormal activity recognition* problem, are: (1) we provide an approach to automatically decide the optimal number of states in state-based methods; (2) combine the power of generative model (HDP-HMM) and discriminative power (OCSVM with Fisher Kernel).

2.3.3 Understanding the substructure of activities

Another important research topic in activity recognition is not only focus on how to recognize “activities”, but how to understand the substructures and relationships between different activities.

A person’s activities in each day may very likely be varied. A person might very likely interleave different activities in any given time period. Recent studies in [49] have shown that in daily lives, people often engage in multiple, interleaving activities. Thus, it is important to design techniques that can detect interleaving activities.

[51] attempts to recognize both concurrent and interleaving activities by applying a skip-chain conditional random field model in a two-level probabilistic framework. The skip-chains added between non-adjacent nodes could enhance the linkage between interleaving activities. To further consider the correlation between goals, a correlation graph is designed to represent the correlation between differ-

ent goals, which can be learned at the upper level of the system architecture. The goal graph is learned from the training data, consisting of sequences of sensor readings and activity labels, to allow the inference of goals in a collective-classification manner.

In [107], an Interleaved Hidden Markov Model is proposed for recognizing multitasked activities, which uses a beam search procedure to define a likelihood update equation over a beam in the search space. Moreover, in [83], a similar model is presented which has the assumption that the observed data stems from multiple hidden processes.

Another relationship that indicates the substructure between activities is that subactivities are often permuted and partially ordered. Therefore, in [21], a Hidden Permutation Model (HPM) that can learn the partial ordering constraints in permuted state sequences is proposed. The HPM is parameterized as an exponential family distribution and is therefore flexible to encode constraints via different feature functions. Inference is done through a chain-flipping Metropolis-Hastings MCMC to overcome the $O(n!)$ complexity.

2.3.4 Social Activity Recognition

In recent years, researchers also start to dig the problem of "social activity recognition". For example, in [176, 174, 175], the authors proposed a mobile activity recommendation system to answer several questions: where should we go in a large city and what should we do if we visit a place. The authors modeled the users' location and activity histories as a user-location-activity rating tensor, which is essentially very sparse. Several factorization models based on collaborative filtering are presented, which also shed light on our proposed solutions in this chapter.

In [30], the authors proposed and evaluated a probabilistic framework for estimating a Twitter user's city-level location based purely on the content of the user's Tweets, even in the absence of any other geospatial cues. Their main assumption is that Twitter users in different cities have a different word distribution. Besides, tweets in some cities have local words. Such a word distribution information as well as the local words mentioned can be used to infer geo-location information at city level. The major difference between their work and our proposed solution is that the number of Tweets published by each user is significantly larger than the number of check-ins one perform in Foursquare. Besides, Tweets have rich content information and thereby it can be anticipated that Tweets should reveal much more information compared to social check-in activities alone. However, our experiments reveal that such social activities with barely no content information would also reveal a lot of information from the users.

One important line of work in social activity recognition lies in location-based reasoning, which tried to explore harnessing data collected on regular smart phones for modeling human behavior. For example, in [39], Eagle *et al.* show that predicting if a person is at home, at work or some place else can be achieved with more than 90% accuracy based on inferring data from mobile phones. However, analyzing social network data is more scalable, practical, compared to mobile phone data since not only can you know information about the subject itself, you can also accumulate possible and useful information from his friends and his community. However, it is noteworthy that the published information online is more noisier and less regular than traditional mobile phone-based data or Bluetooth reading information. In the following paragraphs, we'll describe some recent location-based reasoning approaches from social network data.

In [4], the authors predicted the home address of Facebook users based on provided addresses of one's friends. The authors extracted an empirical relationship between geographical distance and the probability of friendship between pairs of users. Such a relationship is then used to find a maximum likelihood assignment of addresses to hidden users. The authors showed that such a method is generally much more accurate than using IP addresses to determine one's whereabouts.

In [31], the authors modeled user location in social networks as a dynamic Gaussian mixture and assumes that each check-in is induced from the vicinity of either a person's home, work, or is a result of social influence of one's friends. They've found that humans experience a combination of periodic movement that is geographically limited and seemingly random jumps correlated with their social networks. Short-ranged travel is periodic both spatially and temporally and not effected by the social network structure, while long-distance travel is more influenced by social network ties. They've also found that social relationships can explain about 10% to 30% of all human movement, while periodic behavior explains 50% to 70%.

In [132], the authors tried to predict the locations of an individual user in a social network by using the help of known GPS positions of his friends. They've tried to use such GPS information as features and found that the predicted information of an individual users has a very strong correlation with the location of his friends. Following a similar theme, in [133], the authors considered the task of fine-grained prediction of the health of specific people from noisy and incomplete social network data. A probabilistic model is constructed which can predict if and when an individual will fall ill on the basis of his social ties and co-locations with other people, as revealed by their Twitter posts.

2.3.5 Scaling up Graphical Model Methods

As we've shown in the previous sections, currently state-based models are popular for human activity recognition approaches. However, methods like Hidden Markov Model (HMM) or Conditional Random Field (CRF) have a high time complexity, especially when we perform training on a general graph CRF. Although current human activity recognition datasets do not require scaling up of the graphical model methods. In the future, when the number of subjects, number of activities and the amount of training data we collected begin to increase, we will soon find scaling up such graphical model methods a necessity. In this section, we will briefly discuss the possibility of scaling up HMMs and CRFs.

The most typical application on a trained HMM is decoding, which can be performed fairly efficient on a single machine. Therefore, in a MapReduce framework, a trainned HMM can be efficiently used within the Mappers or Reducers. The major problem and the most expensive operation on HMM is the parameter training step. So our main objective is to parallelize the training step of HMMs [10].

One typical method of training the HMM is the Baum-Welch algorithm, which makes two passes of the data using the forward-backward algorithm and compute the expected number of transitions from state i to state j. Both in the forward and the backward algorithm, each task is merely a summation of an array of quantiles. Therefore, each task can be carried out in a parallel reduction manner [102].

The parallel training of CRF is still an active research area. To the best of our knowledge, little has been done over the topic of CRF structure learning, not to mention parallelizing the learning process of CRF structures. Luckily, in the scenario of human activity recognition, the structures of CRFs we use are usually linear or add some skip chains on a linear CRF. Therefore, most of the current research focus on parallelizing the parameter learning step of CRF, as similar to the case of HMM. Most of the optimization methods parallelize over samples [84, 100]. However, in the case of human activity recognition, there are usually more features we extract than the number of samples we have, so parallelizing over samples may be of limited utility. In [17], the authors developed a parallelized version of coordinate descent which aims to parallelize on the number of features and solves this problem. Nevertheless, parallelizing on the full learning process of CRF is still an ongoing research area which remains to be exploited.

CHAPTER 3

RECOGNIZING ACTIVITIES WITH COMPLEX RELATIONSHIPS

In this chapter, we discuss the first problem we've mentioned in Chapter 1, which is to recognize activities where complex relationships exist. Traditionally, many of the activity recognition approaches [118, 155] assume that users achieve one goal at a time, and that goals are achieved through a consecutive sequence of actions (See Figure 3.1, top). However, in many real-world situations, users may accomplish multiple goals within a single sequence of actions where goals are achieved concurrently and the actions that achieve them are interleaving. We call this problem the *multiple-goal recognition* problem. Previous approaches will have problems in this situation. At the end of this section, we show that we can easily apply this idea into recognizing query intention in the virtual world.

3.1 Motivation and Background

Two real-world examples help explain the necessities and difficulties of modeling concurrent and interleaving goals in the multiple-goal recognition problem, respectively. Consider a professor who is leaving his office to achieve the goal of "printing some research papers", he then goes to the seminar room for achieving the goal of "presentation". If the printing room is on his way to the seminar room, then the professor can be considered as pursuing two goals through an observed activity sequence, i.e. the goals of printing and presentation, *concurrently*. In another example, an individual gets up early in the morning and boils water on the kettle. The kettle boils while he is having his breakfast. To attend to the boiling water, he has to pause the process of having breakfast to finish the "water-boiling" goal, by turning off the stove and pouring the hot water. Then he can resume his goal of

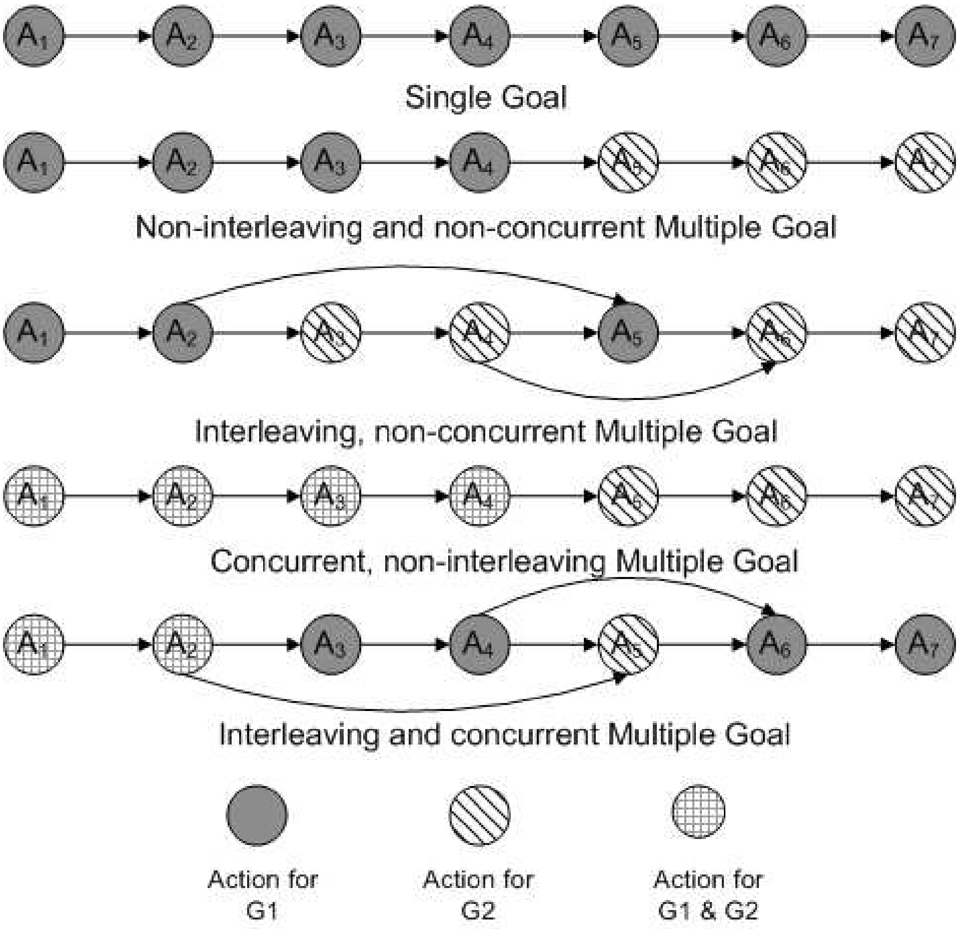

A1
A2
A3
A4
A5
A6
A7
Single Goal
Non-interleaving and non-concurrent Multiple Goal
Interleaving, non-concurrent Multiple Goal
Concurrent, non-interleaving Multiple Goal
Interleaving and concurrent Multiple Goal
Action for G1
Action for G2
Action for G1 & G2

Figure 3.1: Goal composition types in activity sequences

"having-breakfast". In this example, the user is pursuing two goals in an *interleaving* way, where one goal is paused and then resumed after executing some activities for pursuing a different goal.

Generally speaking, in real-world scenarios, there are five basic goal composition types in activity sequences, which are illustrated in Figure 3.1.

MG-Recognizer in [25] tries to tackle the multiple-goal recognition problem, by creating finite state machines to model transitions between states of various goals in a deterministic way. Thus, the approach has trouble handling uncertainty, which is a major drawback. Another drawback is that the *MG-Recognizer* system did not explicitly consider the correlations between different goals. In real-world situations, when we know that a user is pursuing one goal that has strong correlation with some other goals, there is high probability that he is pursuing these correlated goals at the same time. Hence, exploiting correlations between goals can help improve the accuracy of recognizing multiple goals. However, the *MG-Recognizer* system, as well as many previous approaches, did not handle this case either.

In this chapter, we propose a novel two-level probabilistic and goal-correlation framework that deals with both concurrent and interleaving goals from observed activity sequences. Both single-goal recognition and multiple-goal recognition are supported by our solution. In order to reason about goals that can pause and continue through activities in the course of observations, we exploit skip-chain conditional random fields (SCCRF) [143] at the lower level to estimate the probabilities of whether each goal is being pursued given a newly observed activity. To further consider the correlation between goals, a graph that represents the correlation between different goals is learned at the upper level. This goal graph allows us to infer goals in a "collective classification" manner. The probability inferred from the lower level is adjusted by minimizing a loss function via quadratic programming (QP) to derive a more accurate probability of all goals, taking the correlation graph into consideration. We show experimental results using several real-world data sets to demonstrate that our recognition algorithm is effective and accurate than several state-of-the-art methods.

3.2 Our Proposed Method

We formally define our multiple-goal recognition problem. We assume that, as training data, a set S of observed activity sequences is given, without loss of generality, each sequence consists of T observed actions in the form of $\{A_1, A_2, \ldots, A_T\}$. We also assume that there are m goals which are used to label the activity sequences in all. Our objective is to train a model that can decide which subset of the m goals are being pursued given newly observed actions.

3.2.1 Modeling interleaving goals via SCCRF

Our model for interleaving goals is illustrated by the following example. Consider a professor who goes to the general office to get the projector for the "seminar" goal, he then goes to the printing room to pick up the printing material out for the "printing" goal. Finally, the professor may go down a corridor towards the seminar room. Through this example, we can observe that the "get projector" and "go towards seminar room" activities may have *long-distance dependencies* because the "seminar" goal is paused when the professor goes to the printing room. Generally, a goal G may be paused after an action a in a time slice t_i, and then resumed at a later time slice t_j with action b, where actions b and a are separated by several other actions for other goals.

We choose SCCRF proposed in [143] to model the interleaving goal issue for the following reasons. Firstly, SCCRF has deep roots in Natural Language Processing (NLP). In NLP, the problem of Named Entity Recognition (NER) has similarities with the multiple-goal recognition problem, which needs to model the correlation between non-consecutive identical words in the text. Secondly, being a probabilistic graphical model, SCCRF has its advantage in modeling uncertainty in a natural and convenient way. Thirdly, the key issue in SCCRF is how to add skip edges. We use the posterior probabilities from the training data to add the skip edges. Based on the above reasons, we believe that SCCRF would be a model appropriate for handling the interleaving property of multiple goals.

At the lower level of our two-level framework, we consider recognizing each goal separately. Each SCCRF will infer whether an individual goal is active or not at each newly observed activity (see Figure 3.2). To model long-distance dependen-

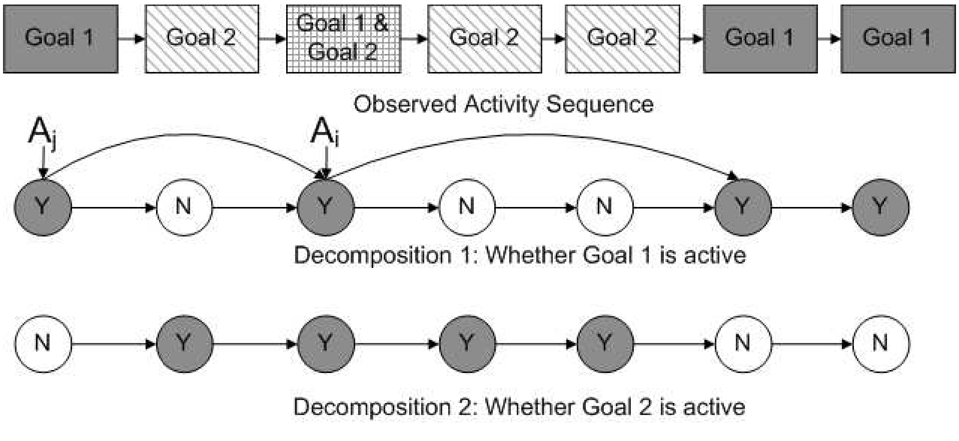

Goal 1
Goal 2
Goal 1 & Goal 2
Goal 2
Goal 2
Goal 1
Goal 1
Observed Activity Sequence
A_j
A_i
Y
N
Y
N
N
Y
Y
Decomposition 1: Whether Goal 1 is active
N
Y
Y
Y
Y
N
N
Decomposition 2: Whether Goal 2 is active

Figure 3.2: Decomposition into goal sequences

cies, for each goal G_k, we first infer the action-transition probability $P(A_i|A_j, G_k)$ (G_k is shown as Goal 1 in Figure 3.2), which stands for the probability of the following situation: given that the goal being pursued is G_k, the last action in the process of pursuing goal G_k is A_j and the next activity being A_i. This probability can be learned by the standard statistical method of Maximum Likelihood Estimation (MLE) or maximum a posteriori (MAP) estimation, where a prior distribution is known. To simplify this preprocessing step, we assume the prior distribution is uniform and then employ the MAP approach in a standard way.

The main characteristic of a SCCRF model over the commonly used linear-chain CRF models is that the SCCRF model added a second type of potential, which was represented using long-distance edges, to the linear-chain model. For each of the m goals under consideration, we build a corresponding SCCRF model, with the i^{th} SCCRF being used to infer whether goal G_i is active given the set of observed activity sequences as training data.

Formally, for the k^{th} SCCRF model which is used to infer the probability of G_k being active, let y_t be a random variable whose value represents whether goal G_k is active or not given activity A_t, which occurs at time slice t. Let x_t be the observed activity at time slice t. For the factor graph $\mathcal{G} = \langle V, E \rangle$, it is essentially a linear-chain CRF with additional long-distance edges between activities A_i and A_j such that $P(A_i|A_j, G_k) > \theta$ (Refer to Figure 3.3 for an illustration). θ is a parameter that can be tuned to adjust the confidence of such long-distance dependencies. We will experimentally verify that small modifications of θ will not affect accuracy greatly.

For an observation sequence $\mathbf{x}$, let $\mathcal{I}$ be the set of all pairs of activities for which there are skip edges connected with each other. Then the probability of a label sequence $\mathbf{y}$ given an observation activity sequence $\mathbf{x}$ is:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(x)} \prod_{t=1}^{n} \Psi_t(y_t, y_{t-1}, \mathbf{x}) \prod_{(u,v)\in I} \Psi_{uv}(y_u, y_v, \mathbf{x}). \tag{3.1}$$

In Equation 3.1, Ψ_t are the factors for linear-chain edges and Ψ_{uv} are the factors over the skip edges. (Also refer to Figure 3.3 for illustration) $Z(x)$ is the normalization factor. We define the potential functions Ψ_t and Ψ_{uv} in Equation 3.2 and Equation

3.3 as:

$$\Psi_t\left(y_t, y_{t-1}, \mathbf{x}\right) = \exp\left(\sum_k \lambda_{1k} f_{1k}\left(y_t, y_{t-1}, \mathbf{x}, t\right)\right) \tag{3.2}$$

$$\Psi_{uv}\left(y_u, y_v, \mathbf{x}\right) = \exp\left(\sum_k \lambda_{2k} f_{2k}\left(y_u, y_v, \mathbf{x}, u, v\right)\right) \tag{3.3}$$

λ_{1k} are the parameters of the linear-chain template and λ_{2k} are the parameters of the skip-chain template. Each of them factorize according to a set of features f_{1k} or f_{2k}.

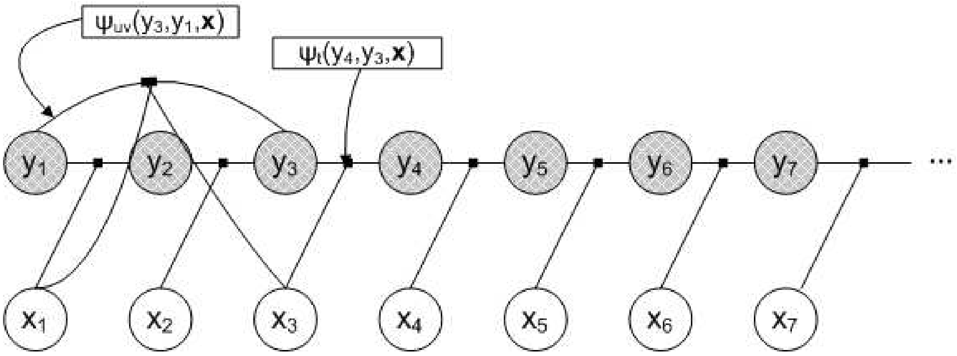


Figure 3.3: Illustration of the SCCRF model

Exact inference in CRFs maybe intractable as the time complexity is exponential in the size of the largest clique in the junction tree of the graph, and that there may be long and overlapping loops in the model. Loopy Belief Propagation (LBP) is used widely for performing approximate inference in CRFs and experiments show that LBP has been effective. Therefore, we set a maximum number of iterations, after which we can calculate the marginal probability of nodes.

Learning the weights λ_{1k} and λ_{2k} for the SCCRF model can be achieved by maximizing the log-likelihood of the training data, which requires calculating the partial derivative and optimization techniques. We omit the details of inference and parameter estimation. Interested readers can consult [81, 144] for technical details.

3.2.2 Modeling concurrent goals via correlation graph

In order to model correlations between concurrent goals, we need to know how similar and correlated two goals are. Similarly, correlation can also tell when one goal is being pursued (e.g. academic-related goals), other goals may be unlikely to be pursued at the same time (e.g. sports-related goals). Therefore, we wish to use the training data to build a correlation graph of goals, where two goals are related by an edge with a large positive weight in $[0, 1]$ if they have strong positive correlations. We omit the considerations of negative correlations here, which we leave for future work.

Note that a full-fledged Bayesian network can be built to model more complex correlations between goals in the form of conditional dependencies which are dependent on multiple random variables, such as $P(G_i|G_j, G_k)$. Furthermore, it is also possible to model concurrency and interleaving together in a CRF framework. However, in real-world situations, such kinds of complex dependencies between goals usually may not occur frequently, resulting the training data acquired to be too sparse to model such a probability and that the learned probability may be highly biased. Another reason is that usually the correlation between goals will not be known as prior knowledge, and such unknown structure adds expensive cost to training. In particular, combining the model of interleaving goals and concurrent goals via a CRF framework will make the training time intolerable, for which we will explain at the end of this section.

Therefore, we only model the probability $P(G_i|G_j)$ using our goal graph explicitly. We show in the experimental section that a factorial conditional random field (FCRF) [163], which represents fully-connected goals through a Bayesian network structure and where goals are modeled in the CRF model, often does not perform as well as our correlation graph-based inference.

From the training data, we can infer the posterior probability $P(G_i|G_j)$ and use it as the initial similarity matrix. The reason why we do not take the currently observed activity into consideration and calculate posterior probability $P(G_i|G_j, A_t)$ is that in real-world situations, the activity sequence usually is not explicitly given as prior knowledge and should be inferred from sensor readings. Therefore, the activity inferred may have noise or bias, which may hinder the inference of probability of goal correlations. Another reason is that we want to model the correlations between goals under a more general environment and assumption.

After calculating the posterior probability of each pair of goals, we take this value and define an $m \times m$ initial similarity matrix S as $S[i,j] = P(G_i|G_j)$. Since the training data may be sparse, the posterior probability we get from the training data may not be so reliable. [12] proposed a method for computing the similarity matrix between vertices of different graphs. We adapted their method for modeling the similarity between vertices of the same graph. We build a directed graph $\mathcal{G} = \langle V, E \rangle$, where the vertices V indicate different goals and $e = \langle G_a, G_b \rangle$ indicates that a goal G_a and a goal G_b have some kind of connection, so that when G_a appears, G_b is also likely to appear. The similarity matrix is updated through iterations of $S_{k+1} = AS_kA^T + A^TS_kA$, where A denotes the adjacency matrix of the similarity graph, where $A[i,j] = 0$ if $P(G_i|G_j) = 0$, otherwise $A[i,j] = 1$. Here S_0 is the initial similarity matrix as defined above. [12] proved the convergence property of this update function. When the iteration procedure converges, some of the edge weights will become zero.

Given m goals, we infer m initial posterior probabilities $P_i^{'}, i = 1, 2, \ldots, m$ from the SCCRF model, which means the probability of goal i being active given a particular observed activity. Then we create the similarity matrix using the probability and the update function mentioned above. After creating the similarity matrix S, we can model concurrent goals by minimizing the differences between strong correlated goals (i.e, $P(G_i|G_j)$ is rather large), to ensure that they will appear together, and minimizing the differences between adjusted posterior probability of a goal P_i and its initial posterior probability $P_i^{'}$ from its individual SCCRF, since this probability carries the observed evidence.

As a result, our top level inference consists of minimizing the following objective function with similarity matrix S and initially inferred probabilities $\mathbf{P}^{'} =$

$\{P_1', P_2', \ldots, P_m'\}$ and our desired output $\mathbf{P} = \{P_1, P_2, \ldots, P_m\}$.

$$\min \sum_{i,j \in \{1,\ldots,m\}} (P_i - P_j)^2 S_{ij} + \mu(P_i - P_i')^2 \tag{3.4}$$

The new probabilities P_i are then used as our predictions. Considering the similarity matrix **S**, as $S[i,j]$ increases towards 1, the difference between P_i and P_j needs to decrease in order to minimize the objective function. The parameter μ can be tuned to reflect the importance of the initial posterior probability learnt from the SCCRF model.

Next, we show that the optimization problem mentioned above can be formulated as a quadratic programming (QP) problem and solved using standard techniques in QP.

We define vector $\mathbf{P} = [P_1, P_2, \ldots, P_m]$ and vector $\mathbf{P}' = [P_1', P_2', \ldots, P_m']$. Then the problem can be expressed as as an optimization problem:

$$\begin{aligned} \min_{P} \quad & P^T(L_S + \mu I)P - 2\mu(P')^T P \\ s.t. \quad & 0 \leq P_i \leq 1, \quad i \in \{1, 2, \ldots, m\} \end{aligned} \tag{3.5}$$

L_S is the Laplacian of S, and is defined as $L_S = D - S$. D is a diagonal matrix where $D[i,i] = \sum_{j=1}^{m} S_{i,j}$. Equation 3.4 can be shown to lead to Equation 3.5. Also, it is evident to show that $(L_S + \mu I)$ is positive definite. Thus the above QP formulation is convex and always has only one global optimum. Furthermore, many state-of-the-art methods can solve QP problems efficiently.

Putting the above together, our main *CIGAR* algorithm is shown in Algorithm 1. We analyze the time complexity of our algorithm and compare with the referenced FCRF method [163]. Assume that training a CRF with T nodes requires time $O(V)$. Therefore, training m CRFs with T nodes only require a complexity of $O(mV)$. However, worst case analysis shows that training a FCRF with mT nodes require a complexity of $O(V^m)$, where m is the number goals and T is the number of activities. Therefore, another advantage of our algorithm over the FCRF method is that our algorithm is more scalable than the FCRF method. The reason also applies

Algorithm 1 Multiple Goal Recognition: *CIGAR*

Input: T is the length of an observed activity sequence $\mathbf{A} = \{A_1, A_2, \ldots, A_T\}$ and m goals $G_1, G_2, \ldots, G_m$.

Output: P_j^i is the probability of goal G_j to be active given observed activity A_i.

```
for $i$ = 1 to m do
    Learn the posterior probability $P(A_j|A_k, G_i)$ for every pair of actions $A_j$ and $A_k$.
    Add a skip edge between $y_j$ and $y_k$ in the $i^{th}$ SCCRF if $P(A_j|A_k, G_i) > \theta$.
    Train the corresponding $i^{th}$ SCCRF model.
end for
for $i$ = 1 to T do
    for $j$ = 1 to m do
        Infer probability $P_j^{i'}$, which represents whether goal $G_j$ is active at time slice $i$.
    end for
    Adjust the initial inferred probability and get the adjusted inferred probability $P_j^i$ with QP.
end for
```

to why we did not model concurrent goals explicitly in the CRF model. In the future, we plan to use other methods for training CRF, like the Virtual Evidence Boosting [91], hoping that we can achieve better accuracy as well as improved training time with the new method.

3.3 Extension: Context-Aware Query Classification

In this section, we discuss how we can easily extend the above intuition into recognizing the intention of human actions in the virtual world. More specifically, we discuss the problem of *query classification*. *Query classification* (or *query categorization*), denoted as QC, has been studied for this purpose by classifying user queries into a ranked list of predefined target categories. Such category information can be used to trigger the most appropriate vertical searches corresponding to a query, improve Web page ranking [72], and help find the relevant online advertisements.

Query classification is dramatically different from traditional text classification because of two issues. First, Web queries are usually very short. As reported in [8], most queries contain only 2-3 terms. Second, many queries are ambiguous [35], and it is common that a query belongs to multiple categories. For example, [135] manually labels 800 randomly sampled queries from the public data set from ACM KDD Cup'05[1], and 682 queries have multiple category labels.

To address the above challenges, a variety of query classification approaches have been proposed in the literature. In general, these approaches can be divided into three categories. The first category tries to augment the queries with extra data, including the search results returned for a certain query, the information from an existing corpus, or an intermediate taxonomy [135]. The second category leverages unlabeled data to help improve the accuracy of supervised learning. Finally, the third category of approaches expands the training data by automatically labeling some queries in some click-through data via a self-training-like approach [88]. Although the existing methods may be successful in some cases, most of them are not context-aware; that is, they treat each query individually without considering the user behavior history.

A MOTIVATING EXAMPLE. Suppose that a user issues a query "*Michael Jordan*". It is not clear whether the user is interested in the famous basketball player or the machine learning researcher at UC Berkeley. Without understanding the user's search intent, many existing methods may classify the query into both categories "Sports" and "Computer Science". However, if we find that the user has issued a query "*NBA*" before "*Michael Jordan*", it is likely that the user is interested in the category of "Sports". Conversely, if the user issues some queries related to machine learning before the query "*Michael Jordan*", it may suggest the user is interested in the topics related to "Computer Science".

Intuitively, using search context information, such as the adjacent queries in the same session as well as the clicked URLs of these queries, can help better understand users' search intent and thus improve the classification accuracy. As shown in previous studies (e.g., [42]), adjacent queries raised by the same user are usu-

[1] ACM KDD Cup'05 is an open contest conducted in conjunction with the ACM KDD'05 conference, which gives a QC task on 800,000 randomly selected Web queries.

ally semantically related. Moreover, compared with search queries, which are often short and ambiguous, the URLs that are selectively clicked by a user after issuing the queries may better reveal the search intent of the user.

3.3.1 Modeling Search Context by CRF

The *Conditional Random Field* (CRF) model is a discriminative graphical model, which focuses on modeling the conditional distribution of unobserved state sequences given an observation sequence [81]. The strength of processing sequential data and incorporating rich features makes CRF model particularly suitable for context-aware query classification.

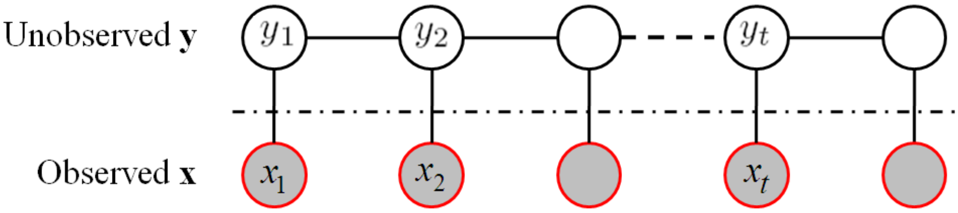


Figure 3.4: Modeling search context by a Linear-chain CRF.

As shown in Figure 3.4, in our problem, a Linear-chain CRF defines the conditional probability of a category label sequence $\mathbf{c} = c_1...c_{T-1}c_T$ given an observation sequence $\mathbf{o} = o_1...o_{T-1}q_T$ as:

$$p(\mathbf{c}|\mathbf{o}) = \frac{1}{Z(\mathbf{o})}\prod_{t=1}^{T}\psi(c_{t-1}, c_t, \mathbf{o}), \tag{3.6}$$

where $Z(\mathbf{o}) = \sum_{\mathbf{c}}\prod_{t=1}^{T}\psi(c_{t-1}, c_t, \mathbf{o})$ is a normalization factor and c_0 is an empty category label which is added for simplicity of defining the model. Potential functions ψ describe the Linear-chain transitions, and are defined as:

$$\psi\left(c_{t-1}, c_t, \mathbf{o}\right) = \exp\left(\sum_k \lambda_k f_k\left(c_{t-1}, c_t, \mathbf{o}\right)\right), \tag{3.7}$$

where f_k is a feature function and λ_k is the weight of f_k. Given training data $\mathcal{D} = \{\mathbf{o}^{(n)}, \mathbf{c}^{(n)}\}_{n=1}^{N}$, the objective of training a Linear-chain CRF is to find a set of parameters $\Lambda = \{\lambda_k\}$ that maximize the conditional log-likelihood:

$$L(\Lambda) = \sum_{n=1}^{N} \log p(\mathbf{c}^{(n)}|\mathbf{o}^{(n)}). \tag{3.8}$$

Once the parameters Λ have been learned using a training data set, we can infer the category label c_T^* for the test query q_T as $c_T^* = \arg\max_{\mathbf{c_T}} p(c_T|\, \mathbf{o}, \Lambda)$.

3.3.2 Features of the CRF model

When we use the CRF to model a search context, one of the most important parts is to choose the effective feature functions. In this section, we introduce the features used for building a CRF model of the search context for QC. In general, the features can be divided into two categories. The features that do not rely on the context information are called *local features*, and those that are dependent on context information are called *contextual features*.

Local features To leverage the local information of individual queries, we consider three types of features that associate queries with the corresponding category label, namely, query terms, pseudo feedback, and implicit feedback.

Query terms Given a query q_t $(1 \leq t \leq T)$ and its category label c_t, the elementary features that reflect the association between q_t and c_t are the terms of q_t. Suppose q_t consists of a set of terms $\{t_{q_t}\}$, each t_{q_t} can be considered as a feature to support the category label c_t. The weights of these features can be learned in the training process of the CRF model.

The problem of this type of features is that query terms are usually sparse. Consequently, the available training data are usually with limited size and could not cover a sufficient set of query terms that are useful for reflecting the association between queries and category labels. Therefore, given a new query whose part of, or all terms do not occur in the training data, this kind of features will not work.

The above problem is difficult to solve because it is hard to label a large number of sessions with a complex taxonomy for a sufficiently large set of terms for all categories. For this reason, we also consider some other features that represent the association between queries and category labels by leveraging some external Web knowledge.

Pseudo feedback This type of features exploits the top M results returned by an external Web directory. Given a query q_t ($1 \leq t \leq T$) and its category label c_t, we first submit q_t to an external Web directory, such as the Google Directory or Yahoo Directory, and get the top M search results. In the second step, for each of the top-M results, we follow the method in [135] and map its category label from a category in the Web directory's taxonomy to a category in the target taxonomy. Finally, we calculate a *general label confidence score*:

$$GConf(c_t, q_t) = \frac{M_{c_t,q_t}}{M},$$

where M_{c_t,q_t} means the number of returned related search results of q_t whose category labels are c_t after mapping. Intuitively, the $GConf$ score reflects the confidence that q_t is labeled as c_t gained from pseudo feedback; the larger the score, the higher the confidence.

Implicit feedback The third type of local features considers the click information as the implicit feedback from users. Similar to the type of features from pseudo feedback, we also exploit an external Web directory. However, we use the clicked URLs by users instead of the top-M results returned by the Web directory to enrich queries. To be more specific, given a query q_t ($1 \leq t \leq T-1$), let the set of clicked URLs of q_t be $U_t = \{u_t\}$, the *click-based label confidence score* of c_t given q_t is defined as:

$$CConf(c_t, q_t, U_t) = \frac{\sum_{u_t} CConf(c_t, u_t)}{|U_t|},$$

where $CConf(c_t, u_t)$ means the confidence that c_t is the most appropriate category label of u_t.

We calculate $CConf(c_t, u_t)$ in three steps. The first two steps are similar to those in calculating the general label confidence score. That is, we first submit q_t to a Web directory and then map the category of each top-M result to a corresponding category in the target taxonomy. After these two steps, we obtain a document collection for each possible category of q_t in the target taxonomy, which will be used to calculate $CConf(c_t, u_t)$. In the third step, we build a *Vector Space Model* (VSM) for each category from its document collection and make the cosine similarity between the term vector of c_t and the term vector of u_t as $CConf(c_t, u_t)$. The snippets of the web pages are used for generating the term vectors.

It is a special case that the top-M search results returned by the Web directory contain the clicked URL u_t. In this case, u_t is associated with a Web directory label $\widetilde{c_t}$. Denoting the mapped category label of $\widetilde{c_t}$ as $\widehat{c_t}$, we define $CConf(\widehat{c_t}, u_t) = 1$ and $\forall_{c \neq \widehat{c_t}} CConf(c, u_t) = 0$.

Note that the $CConf$ score is only applicable when the click information of q_t is available. If a user does not click on any URL for q_t, or q_t is the current query to be classified, this score cannot be calculated.

Contextual features

To use the context information, we consider some features that can reflect the association between adjacent category labels.

Direct association between adjacent labels Occurrence of a pair of adjacent labels $\langle c_{t-1}, c_t \rangle$ $(1 < t \leq T)$ is an obvious feature of the association between adjacent labels, where c_{t-1} and c_t are leaf categories in the target taxonomy. The higher the weight $\langle c_{t-1}, c_t \rangle$, the larger the probability c_{t-1} transits into c_t. The weights of these features are learned from the training data during the training process of the CRF model.

Taxonomy-based association between adjacent labels

Limited by the size of the training data, some transition between categories may not occur in the training data. Moreover, the number of observed transitions may not be reflect the distribution in real world applications. Consequently, the CRF model may not be able to capture the direct association between categories properly.

To reduce the bias of training data, besides considering the feature of direct association between adjacent labels, we also consider the structure of the taxonomy. Intuitively, the association between two sibling categories is stronger than that of two non-sibling categories. For example, the category "Computer$\backslash$Software" is more relevant to "Computer$\backslash$Hard-ware" than to "Live$\backslash$Career& Jobs".

To be more specific, given a pair of adjacent labels $\langle c_{t-1}$,$c_t\rangle$, where c_{t-1} and c_t are both leaf categories at level n, we consider $n-1$ features of taxonomy-based association between c_{t-1} and c_t as $\{\langle\alpha^i_{c_{t-1}}, \alpha^i_{c_t}\rangle\}$ $(1 \leq i \leq n-1)$. The weights of these features are learned from the training data. This idea is similar to smoothing, where, if there are no training data for the feature when $\langle c_{t-1}$,$c_t\rangle$ occurs, there may still be some training data for the features at higher-level transitions $\langle\alpha^i_{c_{t-1}}, \alpha^i_{c_t}\rangle$ in the training data. Let $\beta^i_{c_{t-1}}$ and $\beta^i_{c_t}$ be the level-i siblings of c_{t-1} and c_t, respectively. It is easy to see that $\langle\alpha^n_{c_{t-1}}, \alpha^n_{c_t}\rangle$ occurs if $\exists\beta^n_{c_{t-1}}, \beta^n_{c_t}$ such that $\langle\beta^n_{c_{t-1}}, \beta^n_{c_t}\rangle$ occurs.

3.4 Activity Recognition Experiments

In previous sections, we have described our *CIGAR* approach for recognizing multiple goals in an activity sequence to allow concurrent and interleaving goals. In this section, we will present experimental results of our model to demonstrate that it is both accurate and effective. We compare our algorithm *CIGAR* to the following competing methods. (1) *SCCRF*: interleaving but not concurrent goal recognizer, which applies SCCRF model without correlation graph; (2) *MG-Recognizer* : multiple goal-recognition algorithm presented in [25], with several finite state machines which have different states indicating whether a goal is evolving, suspending or terminating; (3) *FCRF*: which builds a factorial conditional random field (FCRF) over the observed activity sequence, as presented in [163]. We show that our *CIGAR* algorithm can outperform these baseline algorithms. We use three datasets in a cross validation setting to get the recognition accuracy against the baseline methods. Recognition accuracy is defined as the percentage of correctly recognized goals over all goals across all time slices for all the activity sequences.

The first domain is from [25] where the observations are obtained directly from sensor data and the activities correspond to that of a professor walking in a university office area. In this data set, nine goals of a professor's activities are recorded, 850 single-goal traces, 750 two-goal traces and 300 three-goal traces are collected so that the dataset can evaluate both multiple-goal recognition and single-goal recognition. We used three-fold cross validation for training and testing. Table 3.1 shows the comparison in recognition accuracy for both single and multiple-goal recognition tasks. We also tested the performance of our algorithm with different parameter settings. As we can see, *CIGAR* achieves the best performance among all baseline methods, also, small modifications of the parameters θ and μ won't change the recognition accuracy much. Note that FCRF performs much worse in the multiple goal dataset. This is because FCRF did not model interleaving goals.

Algorithm	Single	Multi
MG-Recognizer	94.6%(3.3)	91.4%(4.7)
FCRF	93.6%(5.7)	74.4%(3.8)
SCCRF $(\theta = 0.7)$	94.0%(2.5)	93.5%(2.6)
SCCRF $(\theta = 0.8)$	**94.9%(2.8)**	93.1%(3.9)
SCCRF $(\theta = 1)$	94.8%(2.9)	91.6%(2.9)
CIGAR $(\theta = 0.7, \mu = 0.4)$	94.0%(2.7)	**95.3%(3.4)**
CIGAR $(\theta = 0.7, \mu = 0.5)$	94.8%(2.7)	94.5%(3.7)
CIGAR $(\theta = 0.7, \mu = 0.6)$	94.2%(2.7)	94.4%(3.2)

Table 3.1: Comparison in office dataset

We also used the dataset collected in [118] to further test the accuracy of our algorithm. In this dataset, routine morning activities which used common objects interleavingly are detected through sensors and recorded as sensor data. In this domain, there are a lot of interleaving activities, but there are no concurrent activities. Ten-fold cross-validation is used for testing on this dataset. Table 3.2 shows the comparison in recognition accuracy for this dataset. As we can see, SCCRF and *CIGAR* performs the best amongst all other methods. Note that there are no concurrent goals in this domain, QP actually does no adjustment of the inferred probabilities from the SCCRF.

Algorithm	Accuracy (Variance)
MG-Recognizer	85%(4.6)
FCRF	83%(3.3)
SCCRF $(\theta = 0.7)$	**92%(5.4)**
SCCRF $(\theta = 0.8)$	91%(6.2)
SCCRF $(\theta = 1)$	91%(5.9)
CIGAR $(\theta = 0.7, \mu = 0.4)$	92%(5.2)
CIGAR $(\theta = 0.7, \mu = 0.5)$	92%(5.4)
CIGAR $(\theta = 0.7, \mu = 0.6)$	**92%(5.0)**

Table 3.2: Comparison in [118] dataset

The last dataset we are using is the MIT PlaceLab dataset from [63] and also used for the activity recognition experiment in [163]. We used the *PLIA1* dataset, which was recorded on Friday March 4, 2005 from 9AM to 1PM with a volunteer in the MIT PlaceLab. Note that in this dataset, we are using the location information to predict what activity the user is currently pursuing. Since the original dataset may not contain many concurrent activities, we follow the method in [163] to cluster the 89 activities into six categories where each category corresponds to a new goal. In this way, both interleaving and concurrent activities can be modeled. Table 3.3 shows the comparison in recognition accuracy for the MIT PlaceLab dataset. In this dataset, *CIGAR* performs much better than the baseline methods.

Hence, from the above experiments, we show that *CIGAR* can perform significantly better than baseline methods, and that *CIGAR* can better model concurrent and interleaving goals in real-world situations.

3.5 Query Classification Experiments

In this section, we validate our proposed methods through a systematic empirical comparisons with two baselines over a real data set.

Algorithm	Accuracy(Variance)
MG-Recognizer	68% (4.1)
FCRF	73% (3.8)
SCCRF $(\theta = 0.7)$	80%(3.1)
SCCRF $(\theta = 0.8)$	80%(3.3)
SCCRF $(\theta = 1)$	79%(4.5)
CIGAR $(\theta = 0.7, \mu = 0.4)$	84%(4.3)
CIGAR $(\theta = 0.7, \mu = 0.5)$	**86%(3.0)**
CIGAR $(\theta = 0.7, \mu = 0.6)$	85%(3.3)

Table 3.3: Comparison in MIT PlaceLab dataset

3.5.1 Experimental Set Up and Data Sets

We use the target taxonomy of ACM KDD Cup'05 as our target taxonomy, which is widely used in the literature for QC. This taxonomy is a two-level taxonomy and has seven level-1 categories and 67 level-2 categories.

We randomly extract 10,000 sessions from one day's search log of a major commercial search engine. In this paper, all the extracted sessions contain at least two queries so that we can exploit the impact of contextual information for query classification. The proportion of sessions with more than one query is usually not small. Our search log shows that there are more than 45% such sessions among all sessions. It implies that our approach can help in many cases. Moreover, the queries in single query sessions are mostly “easy queries” that have clear meanings and are easy to be classified. In the extracted sessions, there are 23,091 unique queries and 32,410 unique clicked URLs in total.

Figure 3.5 (a) and Figure 3.5 (b) show the session length distribution and the query frequency distribution of the data set, respectively. From these two figures we can see that in this data set, both the distribution of session lengths and the distribution of query frequencies roughly follow the power law. This phenomenon is consistent with some previous analysis on large scale search logs [24].

We invited three human labelers to label the queries of each session with the 67 level-2 category labels. For each query, a labeler gives a most appropriate category label by considering not only the query itself, but also the search context and the

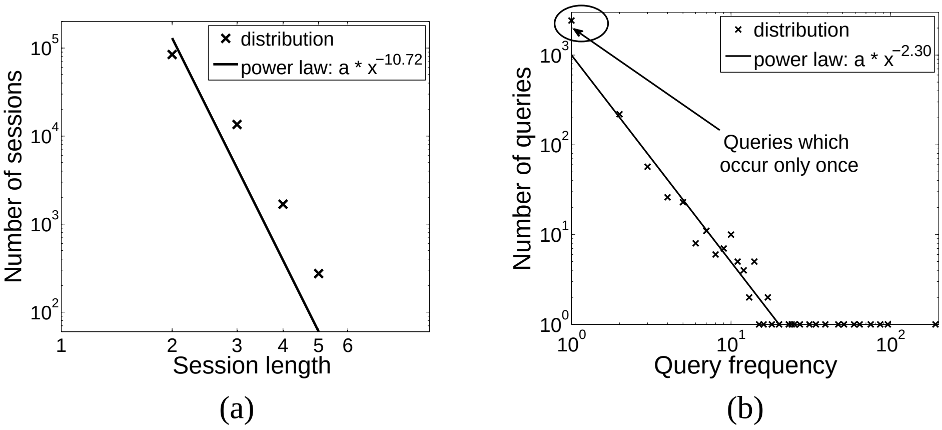


Figure 3.5: Distributions of (a) session lengths and (b) query frequencies of the training data.

clicked URLs of the query. A query's final label is voted by the three labelers. Since each query is associated with context information (except for the beginning queries of sessions) and real user clicks which can help determine the meaning or intent of the query, the consistency among the labelers is quite high. For more than 90% queries, the three labelers give the same labels. This is very different from the general query classification problem [135].

Figure 3.6 shows the category distribution of the labeled queries. From this figure, we can see the category labels of the queries in our data set cover all seven level-1 categories.

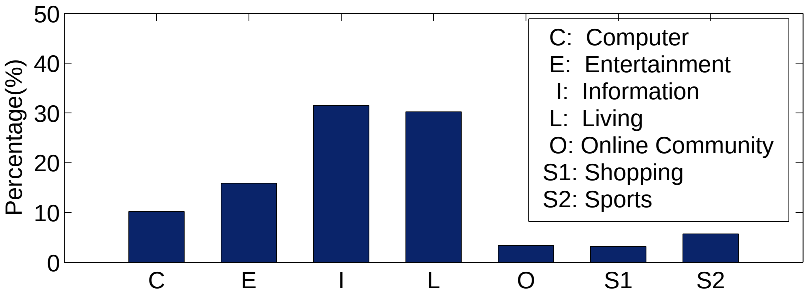


Figure 3.6: Distribution of different category labels in the training data.

3.5.2 Baselines

In this paper, we adopt two baselines to evaluate the performance of our approach:

Bridging classifier(BC): We implement the *bridging classifier* introduced by Shen et al in [135]. The idea of this approach is training a classifier on a immediate taxonomy and then bridging the queries and the target taxonomy in the online step of QC. Experiments in [135] show this approach outperforms the wining approach in KDD Cup'05.

Collaborating classifier(CC): Since there is no existing approach for query classification that takes into account the context information, we design a naive context-aware approach as the second baseline to further evaluate the modeling power of CRF in this problem. The idea of this approach is as follow: given a test query q_T and the previous query q_{T-1} in the same session, 1) firstly we use the bridging classifier to obtain all possible categories of q_T as $C_{q_T} = \{c_{q_T}\}$ with scores $Score(q_T, c_{q_T})$ and all possible categories of q_{T-1} as $C_{q_{T-1}} = \{c_{q_{T-1}}\}$ with scores $Score(q_T, c_{q_{T-1}})$; 2) After that, for each c_{q_T}, we let:

$$\begin{gathered} Score(c_{q_T}) = Score(q_T, c_{q_T}) + \sum_{c_{q_{T-1}}} Score(q_{T-1}, c_{q_{T-1}}) \\ \times AConf(c_{q_{T-1}}, c_{q_T}), \end{gathered}$$

where $AConf(c_{q_{T-1}}, c_{q_T})$ means the *association confidence* [1]of the adjacent label pair $\langle c_{q_{T-1}}, c_{q_T} \rangle$. The association confidence which is calculated as:

$$AConf(c_{q_{T-1}}, c_{q_T}) = \frac{freq(c_{q_{T-1}}, c_{q_T})}{\sum_c freq(c_{q_{T-1}}, c)},$$

where $freq(c1, c2)$ means the frequency of the adjacent label pair $\langle c1, c2 \rangle$ in the training data. Finally the category label ranked list of C_T is generated by ranking $Score(c_{q_T})$.

3.5.3 Evaluation Metrics

Given a test session $q_1q_2...q_T$, we take the last query q_T as the test query and take the queries $q_1q_2...q_{T-1}$ and their corresponding clicked URL sets $U_1U_2...U_{T-1}$ as the search context. In order to evaluate the performance of our approach and the two baselines on the task of query classification with search context, we use three metrics, namely, overall precision, overall recall and overall F_1 score. For a test query q_T with the true category label c_T, given the classification results $C_{T,K}$ where $C_{T,K}$ is a set of the top K predicted category labels from a tested approach, the precision(P) for q_T is represented as $\frac{\delta(c_T \in C_{T,K})}{|K|}$, where $\delta(*)$ is a boolean function of indicating whether $*$ is true(=1) or false(=0). The recall(R) for q_T is represented as $\delta(c_T \in C_{T,K})$ and the F_1 score for q_T is represented as $\frac{2 \times P \times R}{P+R}$. The overall precision is calculated as $\frac{\sum_{n=1}^{N} P_n}{N}$, where N means the number of all test cases and P_n means the precision for the nth test query. The overall recall and overall F_1 score are both calculated in similar ways.

To reduce the uncertainty of splitting the data into training data and test data, we adopt a ten-fold cross validation as follow: 1) Firstly we randomly partition the labeled sessions into ten folds; 2) Then we take each of the ten folds as test data and the remaining nine folds as training data; 3) Finally, we report the average performance of the ten runs.

3.5.4 Overall Results and Analysis

In order to study the contribution of context information, we compare three CRF models with different features: CRF-B (*CRF with Basic features*[2]), CRF-B-C (*CRF with Basic features + Click-based label confidence*) and CRF-B-C-T (*CRF with Basic features + Click-based label confidence + Taxonomy-based association*), respectively. In our experiments, we choose Google Directory as our external Web directory for calculating general label confidence and click-based label confidence. We set M, i.e., the number of used search results of a Web directory, to be 10, which

[2] $Basic\ features$ mean Query terms, General label confidence and Direct association between adjacent labels

equals the number of search results in a search page.

In this section, we evaluate the overall precision, overall recall and overall F_1 score with different K for each tested approach. We set the maximum K to be 5.

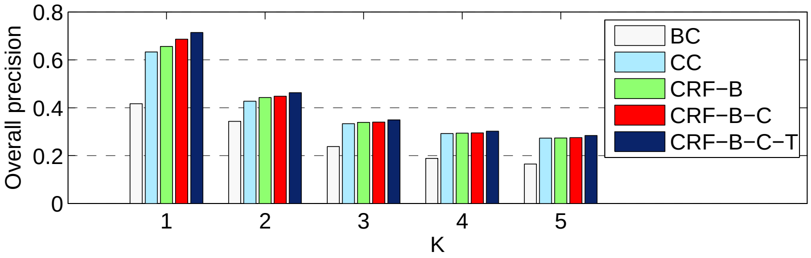


Figure 3.7: The average overall precision of CRF-B, CRF-B-C, CRF-B-C-T and two baselines with different K.

Figure 3.7 compares the average overall precision of CRF-B, CRF-B-C, CRF-B-C-T to the two baselines with different K values. From this figure we can see that all tested approaches' average overall precision numbers drop when we increase K. Compared with the non-context-aware baseline BC, the average overall precision of CRF-B, CRF-B-C and CRF-B-C-T is improved across different K by 50%, 52% and 57% , respectively. Compared with the naive context-aware baseline CC, average overall precision of CRF-B, CRF-B-C and CRF-B-C-T is also improved by 2%, 3% and 7%, respectively.

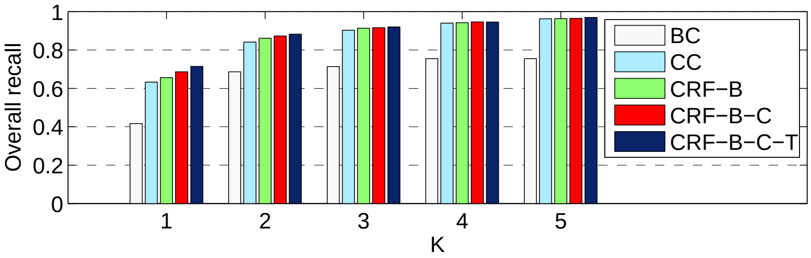


Figure 3.8: The average overall recall of CRF-B, CRF-B-C, CRF-B-C-T and two baselines with different K.

Similarly, Figure 3.8 compares the average overall recall of CRF-B, CRF-B-C, CRF-B-C-T and the two baselines with different K. From this figure we can see that all tested approaches' average overall recall values increase when we increase

K. It is reasonable because the probability that the ground truth label is covered by the predicted results will increase with more predicted category labels. Compared with the non-context-aware baseline BC, the average overall recall of CRF-B, CRF-B-C and CRF-B-C-T is improved across different K by 33%, 35% and 37% , respectively. Compared with the naive context-aware baseline CC, the average overall precision of CRF-B, CRF-B-C and CRF-B-C-T is also improved by 2%, 3% and 4%, respectively.

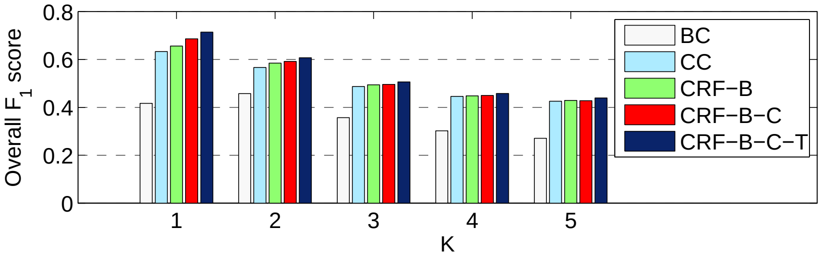


Figure 3.9: The average overall F_1 scores of CRF-B, CRF-B-C, CRF-B-C-T and two baselines with different K.

Figure 3.9 compares the average overall F_1 scores of CRF-B, CRF-B-C, CRF-B-C-T and the two baselines with different K. From this figure, we can see the CRF-B, CRF-B-C and CRF-B-C-T can improve the average F_1 scores by 46%, 48% and 52%, respectively, when compared to the non-context-aware baseline BC. Compared with the naive context-aware baseline CC, the average overall F_1 scores of CRF-B, CRF-B-C and CRF-B-C-T are also improved by 2%, 3% and 6%, respectively.

We conduct a series of paired T-tests of 0.95 confidence level which show that the improvements of our approaches on overall precision, overall recall and overall F_1 are all statistically significant. We also study the variances of overall precision, overall recall and overall F_1 scores of all tested approaches in the ten-fold cross validation. Table 3.4 shows the mean deviations of these values of each tested approach in the ten-fold cross validation with $K = 1$ and $K = 2$, respectively. Notice that when $K = 1$, the overall precision, overall recall and over all F_1 scores are same for each tested approach. From this table we can see that the variances of all three CRFs' performance are consistently smaller than the collaborating classifier. It implies that there is indeed a major advantage of using CRFs for extracting context information, as compared to the collaborating classifier based on a naive context-aware strategy.

K	Approach	Overall P	Overall R	Overall F_1
1	BC	6.77×10^{-3}	-	-
	CC	1.97×10^{-2}	-	-
	CRF-B	7.87×10^{-3}	-	-
	CRF-B-T	1.03×10^{-2}	-	-
	CRF-B-C-T	1.8×10^{-2}	-	-
2	BC	6.48×10^{-4}	1.30×10^{-3}	8.64×10^{-4}
	CC	1.58×10^{-2}	2.43×10^{-2}	1.94×10^{-2}
	CRF-B	2.31×10^{-3}	6.34×10^{-3}	3.47×10^{-3}
	CRF-B-C	5.57×10^{-3}	1.17×10^{-2}	7.47×10^{-3}
	CRF-B-C-T	6.81×10^{-3}	9.88×10^{-3}	8.20×10^{-3}

Table 3.4: Mean deviations of overall precision, overall recall and overall F_1 scores of each tested approach in the ten-fold cross validation.

We also compare the performance of our proposed approaches and the two baseline methods on user-session data with different lengths, where the shortest length is two. From the experiments, we find that the performance of all tested approaches on length-two sessions is a little better than sessions with more queries. This is because it is often the case that the shorter the sessions are, the more likely the queries are common queries that are easy to be classified. Moreover, for sessions with more than two queries, we compare the performance of CRFs by considering different lengths of search context. We find that considering longer search context does not significantly improve the performance as compared to considering only one previous query and its corresponding clicked URLs.

From the above experiments, we can come to the following conclusions: 1) Firstly, all three CRF models and collaborating classifier consistently outperform the bridging classifier on the task of query classification given search context, which implies the effectiveness of context information; 2) Secondly, all three CRF models consistently outperform the collaborating classifier, which is a naive context-aware baseline. It implies that it's an effective approach of modeling context information by CRFs; 3) Thirdly, CRF-B-C outperforms CRF-B, which shows that click information is a good source of context information for query classification; 4) Finally, CRF-B-C-T outperforms CRF-B-C, which indicates that the taxonomy-based association between adjacent labels is useful for the query classification problem with search context.

3.5.5 Case Study

In addition to the study on the overall performance of CRF-B, CRF-B-C, CRF-B-C-T and the two baselines, we also study the cases in which our approach outperforms the baselines.

Context info: travel guide → www.worldtravelguide.net
Query: santa fe new mexico
Snippet of the clicked URL: Santa Fe Travel Information and Travel Guide - USA - Lonely Planet
Ground truth: Living\Travel & Vacation

Category Labels	
Bridging classifier	Information\Local & Regional **Living\Travel & Vacation**
Collaborating classifier	**Living\Travel & Vacation** Information\Local & Regional
CRF-B-C-T	**Living\Travel & Vacation** Information\Local & Regional

Table 3.5: An example of query classification with a search context.

Table 3.5 shows an example of query classification with a search context. In this example, the test query is "*santa fe new mexico*". Without considering the context, this query may have multiple possible search intents. One possible intent is that the user wants to know some general information of the city of Santa Fe, such as the area, the population of this city, etc. In this case, the query should be classified into the "Information\Local & Regional" category. Another possible intent is that the user wants to go on a vocation in the city of Santa Fe and need some travel information about this city, such as hotels and tourist attractions. In this case, the query should be classified into the "Living\Travel & Vacation" category. However, given the context with the query "*travel guide*" in which the user visits a web site related to travel, the appropriate category of this query should be narrowed down to "Living\Travel & Vacation". From Table 3.5, we can see that both CRF-B-C-T and the collaborating classifier give the correct category label in the first position because they considering contextual information, while the bridging classifier's first label is not appropriate. This case exemplifies the effectiveness of considering context information.

Context info: FIFA → fifa08.ea.com
Query: FIFA news
Snippet of the clicked URL: FIFA 08 News, Videos
Ground truth: Entertainment\Games & Toys

Category Labels	
Bridging classifier	Sports\Soccer **Entertainment\Games & Toys**
Collaborating classifier	Sports\Soccer **Entertainment\Games & Toys**
CRF-B-C-T	**Entertainment\Games & Toys** Sports\Soccer

Table 3.6: Another example of query classification with a search context.

Table 3.6 shows another example of query classification given the search context. In this example, the test query is "*FIFA news*". Without considering the context, this query may have two possible meanings: news of the International Federation of Association Football, or news on a soccer video game named "*FIFA*". And the corresponding categories are "Sports\Soccer" and "Entertainment\Games & Toys", respectively. However, given the context that the user has issued a query "*FIFA*" and clicked a URL which is related to the video game "FIFA" , the appropriate category is most likely "Entertainment\Games & Toys". From Table 3.6 we can see that CRF-B-C-T gives the correct category label in the first position, while the collaborating classifier and bridging classifier's first labels are not appropriate. This case exemplifies that CRF-B-C-T leverage search context better than the collaborating classifier.

3.5.6 Efficiency of Our Approach

Our approach consists of an offline part and an online part. In the offline part, the time cost of our approach comes from the training cost for the CRF model. Figure 3.10 (a), (b) and (c) show the convergence curves of CRF-B, CRF-B-C and

CRF-B-C-T, respectively. From these figures we can see the objective function value of CRF-B-C converges to a better optima as compared to CRF-B and the objective function value of CRF-B-C-T converges to a better optima point than CRF-B-C. This implies that considering click information and taxonomy-based association between adjacent category labels can help build a stronger CRF model. The training algorithms are implemented on an Intel Core2 2×2.0G, 4G main memory machine. Each iteration of these algorithms takes about 300 milliseconds. Therefore, the time cost of training a CRF is acceptable as an off line process.

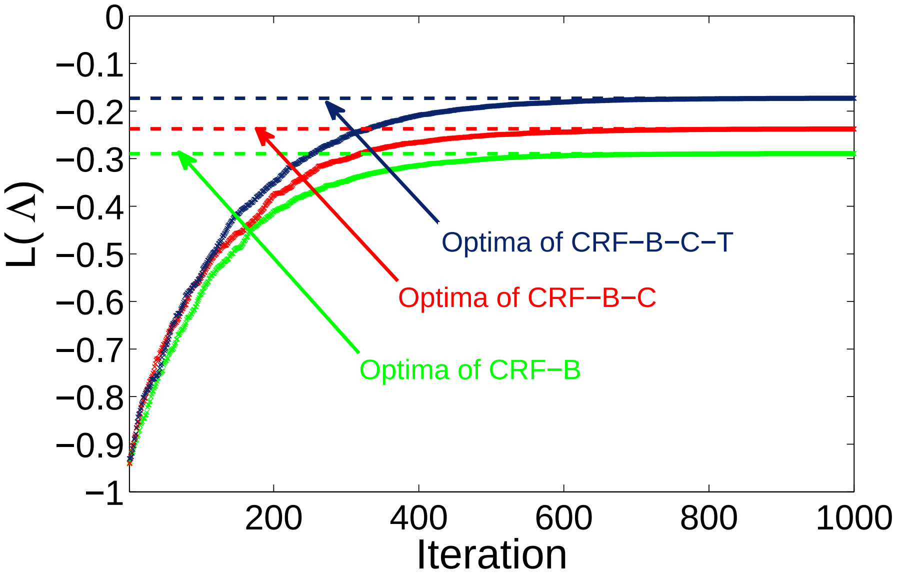


Figure 3.10: Objective function values per iteration of training CRF-B, CRF-B-C and CRF-B-C-T.

As well known, Web users often have strict requirements on the response time of online applications. Thus, the efficiency of an online application is an important problem. In the online part, the time cost of our approach comes from calculating features and inference. In the stage of calculating features, the main cost comes from the process of calculating label confidence. This process can be very fast for a commercial search engine since most modern search engines have their own Web directories locally. Moreover, if we calculate these features offline in advance and store them in local servers, the process will be even faster. Besides, the stage of reference is very fast (less than 0.1 millisecond). This is because usually the length of search context is short and the number of possible categories for a query is small as well. For improving the efficiency of inference further, we can consider only one previous query and its corresponding clicked URLs as search context, since our experiments show that such context information is effective enough for improving the quality of QC significantly.

3.6 Summary

In this chapter, we proposed a two-level framework for inferring the user's high-level goals from activity sequences, meeting the real-world requirement that goals are often interleaving and concurrent. We improve previous algorithms with probabilistic transitions and considered the advantage of exploiting correlations between different goals. Experimental results show that our algorithm achieves better accuracy than baseline methods.

Our work can be extended in several directions. One is that our models can be adapted into an online inference algorithm such that the real-world requirement is better modeled. Also, the effect of negative or more complex correlations between goals may be considered. Another is that we could try to use some other CRF training methods for better accuracy and training complexity.

Web query classification is an important problem with wide applications. However, although many existing works have studied this problem, none of them considered the search context together with query classification. In this chapter, we propose a novel approach for leveraging context information to classify queries by modeling search context though CRFs. Experiments on a real data set extracted from a commercial search engine log clearly show that our approach consistently outperforms a non-context-aware baseline and a naive context-aware baseline.

Our current approach cannot handle the first-query problem well, which is the problem of not being able to find a search context if the query is located at the beginning of a search session. However, if we can capture some events that occurred a little earlier at the beginning of the session, such as events of Web page browsing, we can solve the first query problem well. In our future research, we plan to study this problem in detail.

CHAPTER 4

DETECTING ABNORMAL HUMAN ACTIVITIES

In the real world, activity recognition can be used in a variety of applications, including security monitoring to detect acts of terrorism [66], where terrorist activities are defined as abnormal activities, and helping patients with cognitive disabilities [124].

4.1 Overview

In this chapter, instead of considering how to perform accurate activity recognition, we consider the problem of detecting *abnormal activities*, where we follow the definition used in [167] and define “abnormal activities” as “activities that occur rarely and have not been expected in advance”. Such a problem may first appear to be very similar with the original activity recognition problem in principle. However, the problem of abnormal activity recognition is much harder than the original problem since such abnormal activities, by definition, rarely occur. This difficulty might become more significant during training phase since we lack such labeled sequences of abnormal activities. Up to now, most activity recognition algorithms [87] are systems based on state space-based machine learning models, which require a significant amount of training data in order to perform accurate and successful parameter estimation. Nevertheless, in abnormal activity recognition, such requirements often cannot be satisfied. Most previous research tried to tackle the abnormal activity recognition problem by also using state-space models [167], like Hidden Markov Models (HMMs) or Dynamic Bayesian Networks (DBNs).

There exists one serious problem with these state-space based models, especially HMMs, where one needs to define an appropriate number of states. Usually such a number is determined through a trial-and-error process. In practice, such a number is usually difficult to be known beforehand and the recognition accuracy is usually sensitive to the number of states chosen. However, in real-world applications, it is impossible to undergo this trial-and-error process when the recognizer is attached to humans already and when there is not enough data to validate the accuracy of the model under a particular number of states. Therefore, this drawback can become a major hurdle in real-world activity recognition systems.

In this chapter, we aim to solve the *abnormal activity recognition problem* via a three-phased approach. We first apply the Hierarchical Dirichlet Process Hidden Markov Model (HDP-HMM), which has an infinite number of states and can automatically decide the optimal number of states. Then, we incorporate a Fisher kernel into our model and apply a One-Class Support Vector Machine (OCSVM) to filter out the normal activities. Finally, we derive our abnormal activity model in an unsupervised manner. In this paper, two additional contributions, besides providing an effective and efficient algorithm for solving the *abnormal activity recognition* problem, are: (1) we provide an approach to automatically decide the optimal number of states in state-based methods; (2) combine the power of generative model (HDP-HMM) and discriminative power (OCSVM with Fisher Kernel). We demonstrated the effectiveness of our algorithm through extensive experiments.

One of our previous works [169] also aims to detect the abnormal events from video sequences using Hierarchical Dirichlet Processes. Our work differs from this previous work in several aspects. Firstly, our previous work, which detects abnormal events in video sequences, relies heavily upon feature vectors that we extract from video sequences. Such features normally contain more “representative knowledge” compared to sensor-based activity recognition, where sensor-readings can have both continuous and discrete attributes and understanding the role different sensor readings play in the feature vector is not direct. Secondly, instead of using an ensemble learning algorithm to extract the candidate abnormal events, which might be more heuristic and difficult to explain in principle, in this paper, we incorporate the Fisher Kernel into a One-Class Support Vector Machine model to bring benefit from both generative learning and discriminative learning. Thirdly, in this paper we perform more extensive experiments of our algorithm with different parameters and compare it to different baselines to show each of our components to be useful in our final abnormal activity recognition system.

4.2 Related Work

There is much important previous research work done in trying to tackle the problem of abnormal activity recognition. Due to space constraints, we only review a few related papers which are most relevant to our approach.

With the recent development of sensor networks, activity recognition from sensor data becomes more and more attractive. Many real-world applications require accurate recognition results [124, 43]. So far, among the state-of-the-art learning-based activity recognition algorithms, state-space based models are quite representative. State-space based models usually treat the activities and goals as hidden states, and try to infer such hidden states from the low-level sensor readings by statistical learning. For example, [20] employed an Abstract Hidden Markov Memory Model to represent the probabilistic plans, and used an approximate inference method to uncover the plans. [156] and [94] focused on using Conditional Random Fields and its variants to model the activity recognition problem. However, these algorithms are centered on the recognition of a set of predefined *normal activities*.

Previous approaches on abnormality detection problem range from the computer vision area [38, 169] to data mining areas of outlier detection [85]. Besides our previous work [169], the most relevant work to our approach is [167], which also aims to detect a user's abnormal activities from body-worn sensors. We will describe their algorithm detail in brief since we will be using their algorithm as a baseline. They propose a two-phase abnormality detection algorithm where a One-Class SVM is built on normal activities which help filter out most of the normal activities. The suspicious traces are then passed on to a collection of abnormal activity models adapted via KNLR (Kernel Nonlinear Logistic Regression) for further detection. However, before training the One-Class SVM, they need to transform the training traces that are of variable lengths into a set of fixed-length feature vectors.

To accomplish this task, they trained M HMMs where M is the number of normal activities, and then the likelihood between each sensor reading and normal activity is used as the feature vector. One major drawback of this model is we need to specify the state number of HMMs when training, and such a number will affect the overall algorithm performance a lot as we will show in our experiment section. Thus, their algorithm may not be easy to use in real-world situations since it is hard for users to tune this parameter easily.

4.3 Proposed Solution

We first present an overview of our three-phase approach for abnormal activity recognition from sensor readings. In the first step, we extract the significant features from normal traces, where these features are then used to train an HDP-HMM-based classifier in a sequential manner. The classifier can then be used to decide on a suitable model for every feature automatically. In the second step, we learn a decision boundary around the normal data in the feature space and then use the boundary to classify activities as normal or abnormal via One-Class SVMs. We intentionally train the One-Class SVMs so that they can identify normal activities with a higher likelihood, under the assumption that everything else is abnormal with a lower likelihood. When choosing a threshold value for the general model, we tend to reduce the false positive rate. In the third phase, we perform model adaption to adapt the abnormal activity model to a new model, which gives each abnormal activity a "second chance" to be classified as normal activities[168].

In the remainder of this section, we first briefly review HDP and its Gibbs Sampling methods. We then describe how we incorporate HDP-HMM with OCSVM model. Finally, we describe how we build suitable model adaptation techniques.

4.3.1 HDP-HMM

Hierarchical Dirichlet Process hidden Markov Model

Consider groups of data, denoted as $\{\{y_{ij}\}_{i=1}^{n_j}\}_{j=1}^{J}$, where n_j denotes the number of data in group j, J denotes the total number of groups thought to be produced by related, yet unique, generative processes. Each group of data is modeled via a mixture model. A Dirichlet Process (DP) representation may be used separately for each of the data group. In an HDP, the base distribution of each of the DPs are drawn from a DP, which is discrete with probability 1, so each of the DPs can share the statistical strength, for instance, this encourages appropriate sharing of information between the data sets. An HDP formulation can decide the right number of states for the Hidden Markov Model (HMM) from its posterior density function on the appropriate number of mixture components, to some extent, the number of states in HMM can go to infinite if necessary. Besides, it learns the appropriate degree of sharing of data across data sets through the sharing of mixture components.

The HDP can be built as follows (Due to space constraint, we will omit the detailed explanation of HDP in this paper, interested readers please refer to [152] for technical details.):

$$G_0(\theta) = \sum_{k=1}^{\infty} \beta_k \delta(\theta - \theta_k)$$

$$\beta \sim GEM(\gamma) \qquad \theta_k \sim H(\lambda) \qquad k = 1, 2, \ldots$$

$$G_j(\theta) = \sum_{t=1}^{\infty} \bar{\pi}_{jt} \delta(\theta - \tilde{\theta}_{jt})$$

$$\tilde{\pi}_j \sim GEM(\alpha) \qquad j = 1, \ldots, J \tilde{\theta}_{jt} \sim G_0 \qquad t = 1, 2, \ldots$$

$$\bar{\theta}_{ji} \sim G_j \qquad y_{ji} \sim F(\bar{\theta}_{ji})$$

$$j = 1, \ldots, J, i = 1, \ldots, N_j.$$

where $GEM(\cdot)$ stands for the stick-breaking process as follows:

$$\beta'_k \sim Beta(1, \gamma)$$

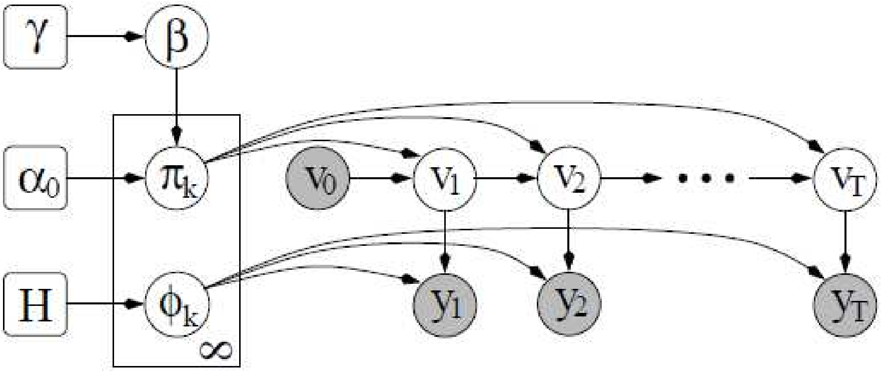


Figure 4.1: A graphical representation of the HDP-HMM Model. [Teh et al. 2006]

$$\beta_k = \beta'_k \prod_{l=1}^{k-1} (1 - \beta'_l), \qquad k = 1, 2, \ldots$$

HMM can be viewed as a doubly stochastic Markov chain and is essentially a dynamic variant of a finite mixture model. Therefore, by replacing the finite mixture with a Dirichlet process, we can complete the design of HDP-HMM (See Figure 4.1 for a graphical representation.)

To better illustrate the construction of HDP-HMM, we introduce another equivalent representation of the generative model using indicator random variables:

$$\beta \sim GEM(\gamma) \qquad \pi_j \sim DP(\alpha, \beta)$$

$$z_{ji} \sim Mult(\pi_j) \qquad \theta_k \sim H(\lambda) \qquad y_{ji} \sim F(\theta z_{ji})$$

Identifying each $G(k)$ as describing both the transition probabilities $\pi_{kk'}$ from state k to k' and the emission distributions parameterized by $\phi_{k'}$, we can now formally define the HDP-HMM as follows:

$$\beta \sim GEM(\gamma), \qquad \pi_k \sim DP(\alpha, \beta), \qquad \phi_k \sim H, \tag{4.1}$$

$$s_t \sim Mult(\pi_{s_{t-1}}), \qquad y_t \sim F(\phi_{s_t}) \tag{4.2}$$

The Gibbs Sampler

The Gibbs sampler was the first MCMC algorithm for the HDP-HMM that converges to the true posterior. [152] proposed three sampling schemes, one of them that is heuristic to HDP-HMM builds on the direct assignment sampling scheme for the HDP, by marginalizing out the hidden variables $\boldsymbol{\pi}$, ϕ from Equations 4.1 and 4.2 and ignoring the ordering of states implicit in $\boldsymbol{\beta}$. Thus we only need to sample the hidden trajectory $\boldsymbol{s}$, the base DP parameters $\boldsymbol{\beta}$ and the hyperparameters α, γ, for this sampler, a set of auxiliary variables m_{jk} is needed, we denote m_{jk} as the number of transitions from state i to state j, and $m_{j\cdot}, m_{\cdot j}$ denote the transitions out and in of state j, the sampling schemes are listed below:

Sampling $\boldsymbol{\beta}$: According to [152], the desired posterior distribution of $\boldsymbol{\beta}$ is:

$$p((\beta_1, \ldots, \beta_K, \beta_{\bar{k}})\boldsymbol{t}, \boldsymbol{k}, y_{1:T}, \gamma) \propto Dir(m_{.1}, ..., m_{.K}, \gamma).$$

Sampling s_t: We now determine the posterior distribution of s_t:

$$p(s_t = k \mid s_{\setminus t}, y_{1:T}, \beta, \alpha, \lambda) \propto$$

$$p(s_t = k \mid s_{\setminus t}, \beta, \alpha) p(y_t \mid y_{\setminus t}, s_t = k, s_{\setminus t}, \lambda)$$

According to the property of Dirichlet processes, we have

$$p(s_t = k \mid s_{\setminus t}, \beta, \alpha) \propto$$

$$\begin{cases} (\alpha\beta_k + m^{-t}_{s_{t-1}k})(\frac{\alpha\beta_k + n^{-t}_{s_{t-1}k} + \delta(s_{t-1}, k)\delta(k, s_{t+1})}{\alpha + n^{-t}_{k.} + \delta(s_{t-1}, k)}) & k \in 1, \ldots, K \\ \alpha\beta_{\bar{k}}\beta_{s_{t+1}} & k = \tilde{k}. \end{cases}$$

The conditional distribution of the observation y_t given an assignment $s_t = k$ and given all other observations y_τ, having marginalized out θ_k, is derived as follows:

$$p(y_t \mid y_{\setminus t}, s_t = k, s_{\setminus t}, \lambda) \propto$$

$$\int_{\theta_k} p(y_t \mid \theta_k) p(\theta_k \mid \{y_\tau \mid s_\tau = k, \tau \neq t\}, \lambda) d\theta_k$$

Sampling m_{jk}:

$$p(m_{jk} = m \mid n_{jk}, \beta, \alpha) = \frac{\Gamma(\alpha\beta_k)}{\Gamma(\alpha\beta_k + n_{jk})} s(n_{jk}, m)(\alpha\beta_k)^m$$

where $s(n, m)$ are unsigned Stirling numbers of the first kind.

4.3.2 Building One-Class SVM with Fisher kernel

Similar to [167], we applied the One-Class SVMs to learn a decision boundary around the normal data in the feature space and then use the boundary to classify activities as normal or abnormal. However, in [167] the author used Gaussian Radial Basis Function (RBF) kernel for the One-Class SVM, but we choose the Fisher Kernel to more effectively combine the strength from both the generative model (HDP-HMM) and the discriminative model (One-Class SVM). Such a combination is usually expected to obtain a robust classifier which has the strengths of each approach.

Fisher kernel

Fisher kernel is introduced in [65]. A kernel that is capable of mapping variable length sequences to fixed length vectors enables the use of discriminative classifiers for variable length examples. Fisher Kernel combines the advantages of generative statistical models (in our framework HDP-HMM) and those of discriminative methods (in our framework One-Class SVMs), where HDP-HMM can process data of variable length and automatically select the suitable model, while One-Class SVMs can have flexible criteria and yield better results. The gradient space of the generative model is used for this purpose since the gradient of the log likelihood with respect to a parameter of the model describes how that parameter contributes to the process of generating a particular example.

The Fisher Score is defined as the gradient of the log likelihood with respect to the parameters of the model:

$$U_X = \nabla_\theta \log P(X|\theta)$$

The Fisher kernel is defined as:

$$K(X_i, X_j) = U_{X_i}^T I^{-1} U_{X_j}$$

where I is the Fisher information matrix [65]and U_X is the Fisher score. In [65], the Fisher information matrix is proposed for normalization, while we also can use other measures to accomplish this task.

One-Class SVM Training

According to [167], we first need to convert the training traces with variable lengths into a set of fixed length feature vectors, here we adopt a set of HDP-HMMs as described in the above section to model the normal traces, one for each type of M features, using beam sampling methods. And the feature vectors in our framework are just the log-likelihood value for each of the N normal traces computed as follows:

$$L_j(Y_i) = \log P_j(Y_i), 1 \leq i \leq N, 1 \leq j \leq M$$

where $\log P_j(Y_i)$ is the log-likelihood of the i^{th} activities trained from the HDP-HMMs based on the j^{th} feature. In this way, for each training trace Y_i, we can obtain an M-dimensional feature vector $X_i = \{L_1(Y_i), \cdots, L_M(Y_i)\}$ for One-Class SVM:

$$\max \sum_{i=1}^{n} \alpha_i K(X_i, X_i) - \sum_{i,j=1}^{n} \alpha_i \alpha_j K(X_i, X_j).$$

where $K(X_i, X_j)$ is the Fisher kernel described above.

As described in [167], a major limitation of using a One-Class SVM for abnormality detection is the difficulty in selecting a sensitivity level that is sufficiently high to yield a low false negative rate and a low false positive rate. To deal with this problem, we also fit our One-Class SVM by selecting parameters so that it is biased toward a low false negative rate. That is, our One-Class SVM can identify, with high confidence, that a portion of data is normal. The rest of the data that are deemed suspicious are passed on to the third phase for further detection. Thus, our One-Class SVM acts as a filter to a classifier by singling out the normal data without creating a model for abnormal characteristics.

4.3.3 Model adaptation

In [167], the abnormal events are derived from a general normal model in an unsupervised manner. The benefit of such an unsupervised manner is that this framework can address the unbalanced label problem due to the scarcity of training data and the difficulty in pre-defining abnormal activities. More specifically, after the second step we may get a high false negative rate, i.e., we may have many normal activities be incorrectly classified as abnormal activities, so it's necessary for us to apply a third phase, that is, to adapt models for the abnormal events, and use these abnormal classifiers to reduce the false negative rate. Besides, due to the lack of negative training data, we cannot directly build models for abnormal events. However, we can use adaptation techniques to get them during the test time or even in future use, that is, we can dynamically build the model for the abnormal event after the training phase. Here we briefly introduce the algorithm's framework first. The steps are listed as below:

Prerequisites: A well defined general HDP-HMM with Gaussian observation density trained by all normal training sequences.

Step 0 : Use the first outlier detected from the former phase - which is considered to be able to represent a particular type of abnormal activities - to train an abnormal event model by adaptation using beam sampler.

Step 1 : Slice the test sequence into fixed length segments, calculate the likelihood of these segments by the existing normal activity models, if the maximum likelihood is given by the general model, we predict this trace to be of a normal activity, then goto Step 4. Else goto Step 2;

Step 2 : If the maximum likelihood is larger than the threshold, we consider this trace to belong to an existing abnormal model; then we predict this trace to be possible abnormal events, go to Step 4, else go to Step 3;

Step 3 : Use adaptation methods to adapt the general model to a new abnormal activity model, then add this adapted abnormal model to the set of models and go to Step 4, here this outlier is regarded to represent one kind of the certain events.

Step 4 : Go to Step 1 if new outlier comes.

In this procedure, we provide the outlier with a second chance to be recognized as a normal event, so that normal events that tend to be unexpected or scarce in the training data are not misclassified. Thanks to the effectiveness of beam sampler again, we can do the adaptation effectively without other special design. Suppose that we have the new parameters for the HDP-HMM λ, here we update the HDP parameters $\beta, \alpha_0, \gamma, K$ and HMM parameters π, μ. Notice that in Step 1, an abnormal activity sequence may be predicted as normal activities again, thereby decreases the false negative rate in this Step. And in Step 2, we classify such an abnormal activity sequence to one abnormal activity in the current activity set we are now holding. There may still be cases where we have not seen this abnormal activity before, and we perform Step 3 so that we can create a new abnormal activity set, and humans can be involved to analyze what this abnormal activity sequence actually means in real life. Such a framework is useful for real-world deployment of our abnormal activity recognition algorithm.

4.4 Empirical Evaluation

4.4.1 Datasets, Metrics and Baselines

We use two real-world activity recognition datasets. The first is the MIT PLIA 1 dataset [63], which was recorded on Friday March 4, 2005 from 9AM to 1PM with a volunteer in the MIT PlaceLab. The dataset contains 89 different activities and was manually classified into several categories including: Cleaning, Yardwork, Laundry, Dishwashing, Meal Preparation, Hygiene, Grooming, Personal and Information/Leisure. Due to the fact that "abnormal activities" are usually hard to define and previous work including [167] and [168] often manually defined some activities with low probabilities as abnormal activities, we manually selected three activities with low probabilities and consider such activities as abnormal activities we aim to detect from sensor readings. And the second dataset we are using, referred to as *Yin* in Table 5.2 is from [167], where a number of traces of a user's normal daily activities in an indoor environment are recorded. In this dataset, the user was asked to simulate the effect of carrying out several abnormal activities.

The evaluation metric that we are using in this paper is the AUC (Area Under Curve) measurement [16], since a good abnormal activity recognition system should have both high detection rate (defined as the ratio of the number of correctly detected abnormal activities to the total number of abnormal activities) and low false alarm rate (defined as the ratio of the number of normal activities that are incorrectly detected as normal activities to the total number of normal activities). The ROC curve plots the detection rate against the false alarm rate and therefore becomes our choice in such a problem.

The algorithms we plan to analyze in this paper are as follows: **HMM + RBF + KNLR**, which is the algorithm discussed in [167]'s paper and implemented by ourselves, **HDP + Fisher + Adaptation**, which is our proposed method by using HDP and Support Vector Machine with Fisher Kernel, together with the model adaptation method we proposed, **HDP + RBF + KNLR**, which is exactly the original baselines except that we use a HDP-HMM in the first phase to automatically determine the optimal number of states in HMM. **HDP + RBF + Adaptation**, same as our algorithm but we use a traditional RBF kernel to train the OCSVM model. We design these baseline methods to demonstrate the effectiveness of our framework, and also show that our two main contributions, (1) using HDP-HMM to optimally decide the optimal number of states and (2) incorporating Fisher Kernel into the OCSVM model, are both effective in this problem.

4.4.2 Experimental Results

In this section we present our experimental results in Table 5.1. The AUC score of each algorithm is calculated and the training set is drawn at random ten times to calculate a variance of the AUC score. For the baseline methods, since the number of states in the HMM model Q needs to be manually defined, we tested the algorithm performance with varying numbers of Q from 2 to 8.

From Table 5.1 and Table 5.2, we can see that our framework *HDP + Fisher + Adaption* outperforms the baseline algorithm and some other baselines that we have set. When we set Q from 2 to 8, we can see that the AUC score varies between 0.683 and 0.793 in PLIA1 dataset, and the AUC score varies between 0.713 and 0.785 in [167]'s dataset, which clearly indicates the difficulty of choosing an appro-

Algorithm	PLIA1 AUC (Variance)
HMM + RBF + KNLR (Q = 2)	0.683(0.025)
HMM + RBF + KNLR (Q = 3)	0.764(0.027)
HMM + RBF + KNLR (Q = 4)	0.793(0.025)
HMM + RBF + KNLR (Q = 5)	0.721(0.018)
HMM + RBF + KNLR (Q = 6)	0.657(0.030)
HMM + RBF + KNLR (Q = 7)	0.642(0.019)
HMM + RBF + KNLR (Q = 8)	0.631(0.016)
HDP + RBF + KNLR	0.811(0.032)
HDP + RBF + Adaptation	0.835(0.017)
HDP + Fisher + Adaptation	**0.857(0.028)**

Table 4.1: Performance Comparison of our algorithm and the baseline methods on the MIT PLIA Dataset

Algorithm	Yin's AUC (Variance)
HMM + RBF + KNLR (Q = 2)	0.713 (0.028)
HMM + RBF + KNLR (Q = 3)	0.725 (0.021)
HMM + RBF + KNLR (Q = 4)	0.748 (0.010)
HMM + RBF + KNLR (Q = 5)	0.785 (0.015)
HMM + RBF + KNLR (Q = 6)	0.732 (0.017)
HMM + RBF + KNLR (Q = 7)	0.718 (0.013)
HMM + RBF + KNLR (Q = 8)	0.707 (0.019)
HDP + RBF + KNLR	0.792 (0.018)
HDP + RBF + Adaptation	0.813 (0.021)
HDP + Fisher + Adaptation	**0.834((0.029)**

Table 4.2: Performance Comparison of our algorithm and the baseline methods on the dataset from [Yin *et al.*,2008]

priate number of states and the impact of a non-optimal state on the final recognition accuracy cannot be neglected. When using *HDBP + RBF + KNLR*, we notice that its performance already outperforms that of HMM-based models. Therefore, adopting HDP-HMM in our model can automatically determine the appropriate number of states and algorithm performance will not be affected since we avoid a step of trial-and-error process. We can also see that using *HDP + RBF + Adaptation* is not as good as our proposed method which uses Fisher kernels on the two datasets we've tested, which suggests that our proposed approach for incorporating Fisher kernel into this framework will have stronger predictive strengths compared to incorporating the commonly-used RBF Kernels.

Therefore, in this section, by reporting the performance of our algorithm on two activity recognition datasets and by comparing the performance of our algorithm with the baseline algorithms, we have demonstrated empirically that our framework is useful at each step, and that introducing HDP and Fisher Kernel can improve the overall performance.

4.5 Summary

In this chapter, we have presented a novel framework for tackling the problem of abnormal activity recognition. Our method does not suffer the problem of hard to determine an optimal number of states as previous state-based approaches do. We applied an HDP-HMM model that can automatically select the suitable model with the optimal number of states. We analyzed the efficiency and effectiveness of introducing beam sampling in the HDP-HMM model. We also combined the powers of both generative models and discriminative models by using the Fisher Kernel in the One-Class SVM model in the second step. Finally, we described a model adaptation approach so that we can detect unseen abnormal activities. In the future, we wish to explore some effective online inference algorithms for us to tackle the abnormal activity recognition problem in a more natural way to meet the need of real-world applications.

CHAPTER 5

ACTIVITY RECOGNITION ACROSS DIFFERENT DOMAINS

5.1 Overview

A common problem with supervised machine learning is the potentially expensive manual effort needed to label the training data. This problem is very pronounced in the field of activity recognition. One assumption required by most supervised learning methods is that the training and test data should be in the same feature space and have the same underlying distribution and the same label space. However, when the distributions and features are different between training and future data, the model performance often drops.

In the context of activity recognition, the above assumption manifests itself as: 1) The same feature space requirement means that training and testing data should use the same set of sensors; 2) The same underlying distribution requirement means that the preferences or the habit of the subjects should be similar in both training and testing data and 3) The same label space requirement means the activity set recognized in the training and testing data are the same.

Consider an example in Figure 5.1. The taxonomy is an activity taxonomy extracted from the MIT PLIA1 dataset [63, 49], representing the common daily activities. Suppose that some user wants to set up an activity recognition system at his/her home to recognize the activities in the taxonomy. However, the user only wants to spend very little amount of time and effort to label the sensor data, where labels correspond to activity names such as "washing dishes". A user may not be able to label the sensor data associated with all activities described in the taxonomy. For example, the user may only label the sensor data from the activities in the "Cleaning Indoor" category, and leave unlabeled the sensor data from the other

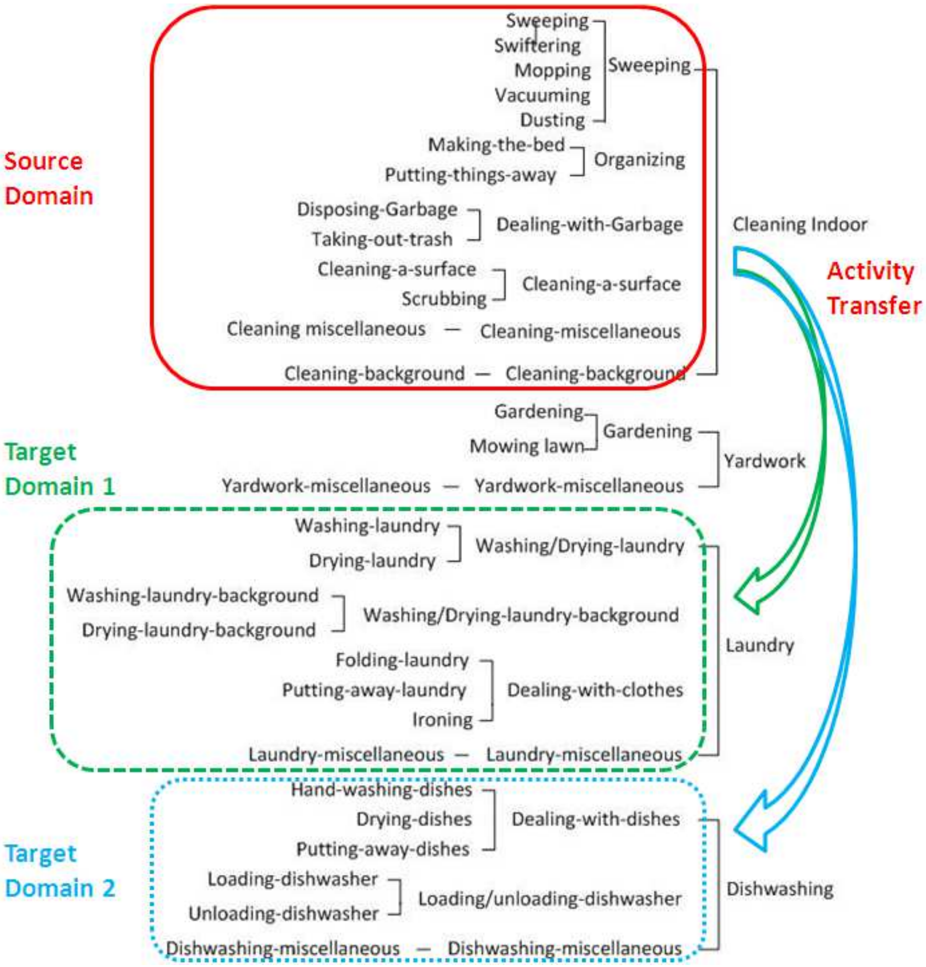


Figure 5.1: An example of cross-domain activity recognition.

categories' activities (*e.g.* in "Laundry", "Dishwashing"). So in our problem, we have a *source domain* of activities that has the labeled sensor data from activities in the "Cleaning Indoor" category. We also have some *target domain* that has the unlabeled sensor data from the activities in some other category such as "Laundry" (denoted as "Target Domain 1") or "Dishwashing" (denoted as "Target Domain 2"). Then, we ask the following fundamental questions:

1. *Is it possible for us to use the labeled data in the source domain to help train an activity recognizer in the target domain?* For example, can we use the sensor data from the "Cleaning Indoor" category in training an activity recognizer for the "Laundry" category?
2. *Under what conditions can domain transfer work for activity recognition?*

To relax the assumption of same feature and label space as well as underlying distributions, many transfer learning algorithms have been developed to reduce labeling effort while still maintaining a reasonable accuracy . In transfer learning, useful knowledge from the source domain are being transferred to the target domain where labeled data is usually insufficient to build a reliable classifier on its own [113, 151]. Recently, researchers have tried to bring the idea of transfer learning into activity recognition, but but most of their approaches have certain associated limitations, which we will discuss in Section 2.

In this chapter, we propose a transfer learning framework under which one can transfer the knowledge between different activity recognition tasks, relaxing the assumption of same feature space, same label space as well as same underlying distribution by automatically learning a mapping between different sensors. To build a mapping between the two domains, we use Web knowledge as a bridge to help link the different label spaces.

5.2 Related Work

Activity recognition aims to infer user behaviors from observations such as low-level sensor readings. However, most of the proposed activity recognition algorithms focused on sensor readings from only one domain, and usually require lots of annotated data to train the activity recognition model.

Transfer learning is motivated by the fact that humans can intelligently apply knowledge learned previously to solve new problems faster. Transfer learning has already been demonstrated successful in many scenarios [113] . More specifically, there have been several works that tries to link transfer learning with activity recognition [157, 171]. In [171], the authors relax the same label space assumption by using Web knowledge as a bridge to transfer knowledge between different label spaces. The limitation in this paper is that the requirement of same feature space still applies, *i.e.*, the sensors in the source domain and the target domain should be the same. Such a limitation forbids many possible transferring scenarios in activity recognition. In [157], the authors studied activity recognition across different sensors. However, their algorithm is based on the usage of a *meta-feature space*, which are features that describe the properties of the actual features, *e.g.*, both sensors installed on microwaves and stoves have a meta feature as “kitchen heating”. Each sensor is described by one or more meta features. The limitations of the approach described in [157] is that the meta-feature space needs to be manually constructed. Besides, different room layouts or different kinds of sensors would lead to huge difference in the meta features of the two rooms and hence the applicability of the algorithm is limited. Rashidi and Cook [129] studied transferring activity recognition knowledge from multiple source domains. However, their work does not allow the feature space of different domains to be different.

In our activity recognition problem setting, we need to transfer knowledge between different feature spaces, underlying distribution as well as different label spaces. In the transfer learning literature, transferring between different feature spaces has been studied extensively, *e.g.*, [36, 166]. However, few research works have dealt with the knowledge transfer problem that involve changes in all of feature, distribution and label spaces.

5.3 Motivating Approach: Transfer Between Same Feature Space

In this section, we first describe a motivating approach where we attempt to transfer knowledge from one domain to the other under the same feature representation. by using information from Web knowledge. More specifically, we try to analyze the "correlation" between different activities and then attempt to create pseudo training instances based on such correlation learned from Web knowledge. Next, we build a classifier based on these pseudo-instances. Such an instance-transfer method, motivated by data instance importance sampling, will also play an important part in the approach we describe in the next section.

5.3.1 Algorithm Overview

Our work belongs to the instance-transfer category in transfer learning framework. In general, the instance-transfer algorithms are motivated by data instance importance sampling [141]. That is, the training data from a source domain are weighted to train a model for the target domain, and the weights can be generally seen as the similarities between the source domain's data and the target domain's data. The more similar some source domain's data are to the target domain's data, the higher weights the source domain's data will be assigned in the learning procedure. Different from the previous work on instance-transfer which measures the similarities from the data (features), we show that in our cross-domain activity recognition problem, we need to measure the similarities from the label information.

We first present an overview of our cross-domain activity recognition (referred to as CDAR below) algorithm to provide the readers with a high-level overview of our algorithm. Our CDAR algorithm can be generalized into three steps.

In the first step, we aim to learn a similarity function between different activities by mining knowledge from the Web. In particular, we will use Web search to extract related Web pages for the activities, and then apply information retrieval techniques to further process the extracted Web pages. After that, we use some similarity measure, such as Maximum Mean Discrepancy in Equation (5.2), to calculate the similarities between any pair of activities from the source domain and target domain. Such similarities will be used later to calculate confidence for pseudo training data for domain transfer.

In the second step, given the assumption that we only have labeled training data in the source domain but no labeled training data in the target domain, it is impossible to follow supervised learning methods to train a recognizer for the target domain's activities. Therefore, by using the similarity values we have learned in the first step, we aim to generate some pseudo training data for the target domain with some confidence values. Here "pseudo training data" are the training data with the same feature values as in the source domain, but relabeled with the activity labels in the target domain. Such data relabelings will be assigned with some confidences, whose values equal to the similarities we calculated in the first step; and these confidences will measure how "strong" a particular training data instance in the source domain can be explained as the data instances in the target domain.

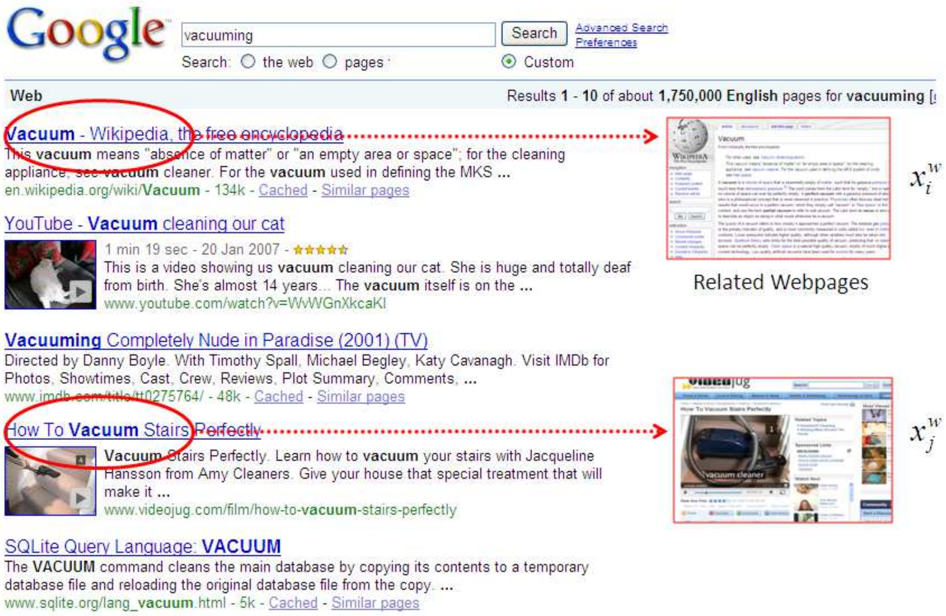
Following the second step, if we have N activities in the source domain and M activities in the target domain, we will create pseudo training data in the target domain with a label space with size NM. Each pseudo training instance is supported with a confidence value. Therefore, by using the pseudo training data, we can apply a weighted Support Vector Machine method to train a classifier, so that we can use it to recognize the activities in the target domain.

5.3.2 Learning the Similarity Function

In this section, we will show how to learn a similarity function for any pair of activities from the source domain and the target domain. To achieve this, we will novelly exploit the Web data.

Calculate Similarity from Web Data

With the proliferation of the Web services, there are emerging Web pages that describe the daily activities. For example, we can easily find many web pages introducing how to make coffee. Such web pages encode the human understanding of the activity semantics, such as what kind of activity it is, what kind of objects it uses, *etc*. Such semantics can greatly help measure the similarities between the activities.



Results **1** - **10** of about **1,750,000 English** pages for **vacuuming**

Vacuum - Wikipedia, the free encyclopedia
This **vacuum** means "absence of matter" or "an empty area or space"; for the cleaning appliance, see **vacuum** cleaner. For the **vacuum** used in defining the MKS ...
en.wikipedia.org/wiki/**Vacuum** - 134k - Cached - Similar pages

YouTube - **Vacuum** cleaning our cat
1 min 19 sec - 20 Jan 2007 - ★★★★★
This is a video showing us **vacuum** cleaning our cat. She is huge and totally deaf from birth. She's almost 14 years... The **vacuum** itself is on the ...
www.youtube.com/watch?v=WvWGnXkcaKl

Vacuuming Completely Nude in Paradise (2001) (TV)
Directed by Danny Boyle. With Timothy Spall, Michael Begley, Katy Cavanagh. Visit IMDb for Photos, Showtimes, Cast, Crew, Reviews, Plot Summary, Comments, ...
www.imdb.com/title/tt0275764/ - 48k - Cached - Similar pages

How To **Vacuum** Stairs Perfectly
Vacuum Stairs Perfectly. Learn how to **vacuum** your stairs with Jacqueline Hansson from Amy Cleaners. Give your house that special treatment that will make it ...
www.videojug.com/film/how-to-**vacuum**-stairs-perfectly

SQLite Query Language: **VACUUM**
The **VACUUM** command cleans the main database by copying its contents to a temporary database file and reloading the original database file from the copy. ...
www.sqlite.org/lang_**vacuum**.html - 5k - Cached - Similar pages

Figure 5.2: Extract Web data for activities.

In practice, as the activity names are known, we can employ Web search to extract Web pages related to the activities. For example, for an activity "Vacuuming" defined in the taxonomy of Figure 5.1, we can search on Google with the query "Vacuuming" as shown in Figure 5.2. Then we can get a list of search results on the page. By clicking all search results, we can get a set of Web pages. Although the Web pages contain a lot of information, only a small amount of such information is related to the semantics of the queried activity. So we apply classic information retrieval techniques to retrieve the useful information for each Web page.

In particular, for each Web page, we first extract all the words in it, and then treat the extracted words as a document d_i with a bag of words. Such a document d_i can be further processed as a vector x_i^w, each dimension of which is the term frequency-inverse document frequency value (tf-idf) [101] of a word t:

$$tf\text{-}idf_{i,t} = \frac{n_{i,t}}{\sum_l n_{i,l}} \cdot \log \frac{|\{d_i\}|}{|\{d_i : t \in d_i\}|},$$

where $n_{i,t}$ is the number of occurrences of word t in document d_i. Besides, $|\{d_i\}|$ is the total number of collected documents, and $|\{d_i : t \in d_i\}|$ is number of documents where word t appears. The terms in the tf-idf equation are explained as follows:

- The first term $\frac{n_{i,t}}{\sum_l n_{i,l}}$ denotes the frequency of the word t that appears in the document d_i. If the word t appears more frequently in the document d_i, then $|\{d_i : t \in d_i\}|$ is larger, and thus the whole term is larger. For example, in the returned Web page of "Vacuuming", the word "clean" may appear many times, so its term frequency is high. It means that such a word encodes some semantics of "Vacuuming", and it has higher weights in the Web data's feature vector.

- The second term $\log \frac{|\{d_i\}|}{|\{d_i : t \in d_i\}|}$ denotes the inverse document frequency for the word t. If the word t appears in more documents of the corpus, then $|\{d_i : t \in d_i\}|$ is larger, and thus the whole term is smaller. For example, for the word "the", it is used in almost all the documents, but it is a stop word without any meaning. Hence, its inverse document frequency will vanish to zero, thus the whole tf-idf value of the word is zero. It means that such a word does not encode any semantics of the searched activity, so it can be removed from the Web data's feature vector.

Therefore, for an activity u (*e.g.* "Vacuuming"), we can get a set of documents $\mathfrak{D}_u^w = \{x_i^w | i = 1, ..., m_u\}$, with each x_i^w as a tf-idf vector. Similarly, for another activity v (*e.g.* "Washing-laundry"), we can also Google it and get another set of documents $\mathfrak{D}_v^w = \{z_i^w | i = 1, ..., m_v\}$, with each z_i^w as a tf-idf vector.

After having the extracted Web data $\mathfrak{D}_u^w$ and $\mathfrak{D}_v^w$, now we will show how to measure the similarity between the activity u and the activity v. Note that a possible choice to calculate the similarity between two (Web) data distributions is using the Kullback-Leibler (KL) divergence as [164] did. However, generally the Web text data are high-dimensional and it is hard to model the distributions over the two different data sets. Hence, we propose to use the Maximum Mean Discrepancy (MMD) [15], which can directly measure the distribution distance without density estimation, to calculate the similarity. Considering the universal reproducing kernel Hilbert spaces (RKHS), we can interpret the function f as the feature mapping function $\phi(\cdot)$ of a Gaussian kernel [15].

Given the Web data $\mathfrak{D}_u^w = \{x_i^w | i = 1, ..., m_u\}$ for activity u and the Web data $\mathfrak{D}_v^w = \{z_i^w | i = 1, ..., m_v\}$ for activity v, we can finally have the similarity between u and v as

$$sim(u, v) = MMD^2[\mathfrak{D}_u^w, \mathfrak{D}_v^w], \tag{5.1}$$

where $MMD^2[\mathfrak{D}_u^w, \mathfrak{D}_v^w]$ is the maximum mean discrepancy defined as:

$$\begin{aligned} MMD^2[\mathfrak{D}_u^w, \mathfrak{D}_v^w] &= \left\| \frac{1}{m_u} \sum_{i=1}^{m_u} \phi(x_i^w) - \frac{1}{m_v} \sum_{i=1}^{m_n} \phi(z_i^w) \right\|_{\mathcal{H}}^2 \\ &= \frac{1}{m_u^2} \left\| K_{uu}^w \right\|_1 - \frac{2}{m_u m_v} \left\| K_{uv}^w \right\|_1 + \frac{1}{m_v^2} \left\| K_{vv}^w \right\|_1 , \end{aligned} \tag{5.2}$$

where K_{uv}^w is the Gaussian kernel defined over the data $\mathfrak{D}_u^w$ and $\mathfrak{D}_v^w$. Specifically, K_{uv}^w is a $m_u \times m_v$ matrix, with its entry at row i and column j defined as

$$K_{uv}^w(x_i^w, z_j^w) = \exp(-\frac{\left\| x_i^w - z_j^w \right\|^2}{2\sigma^2}),$$

where σ is the kernel width for the Gaussian kernel function. In Equation (5.2), $\|\cdot\|_1$ is an entry-wise norm which sums up all the entries in the matrix.

5.3.3 Generating Pseudo Training Data

Now we have the similarity value $\mathrm{sim}(u, v)$ for each pair of activities $u \in \mathcal{A}_{src}$ and $v \in \mathcal{A}_{tar}$. How can we generate a new training data set defined over the label space of the target domain? Recall that, in the source domain, we have the training labeled data $\mathcal{D}_{src}^{trn} = \{(x_{src}^{(i)}, y_{src}^{(i)})\}_{i=1}^{T_1}$, where $y_{src}^{(i)} \in \mathcal{A}_{src}$. For each training instance $(x_{src}^{(i)}, y_{src}^{(i)})$ with $y_{src}^{(i)} = u$ where $u \in \mathcal{A}_{src}$, we will relabel it to get a set of pseudo training data as $\{(x_{src}^{(i)}, v_j, sim(u, v_j))|v_j \in \mathcal{A}_{tar}\}_{j=1}^{|\mathcal{A}_{tar}|}$. Here, the similarity $sim(u, v_j)$ between activity u and v_j is used as the confidence of such a relabeling. In other words, we duplicate each training instance $|\mathcal{A}_{tar}|$ times; and each duplication will be relabeled using one activity category in the target domain with some confidence. Finally, these relabeled training data duplications, which we call "pseudo" training data, are then used for training classifiers to classify activities in the target domain.

5.3.4 Weighted SVM Method

Now we have a pseudo training data set on the target domain where each data instances in the dataset contains not only a category label but also a confidence value. The confidence value is defined as the similarity value we calculate between two activities. Therefore, the larger the value is, the more similar the two activities are, and the more confident we are when interpreting such a training data instance to this activity in the target domain.

However, training support vector machines with confidence values attached to training instances is a non-trivial task and we apply the method proposed in [26] to accomplish our goal. Interested readers can follow the original paper for technical details. Here we will briefly introduce the weighted SVM model for multi-class classification.

In [26], a "one-against-one" approach is employed for multi-class classification. Given the N classes (in our case, each activity in the target domain is a class, so $N = |\mathcal{A}_{tar}|$), this approach constructs $N(N-1)/2$ classifiers, each of which trains the data from two different classes. For training data from the i^{th} and the j^{th} classes, the weighted SVM model solves the following two-class classification problem:

$$\begin{aligned}\min_{\mathbf{w}^{ij}, b^{ij}, \xi^{ij}} \ & \frac{1}{2}(\mathbf{w}^{ij})^T\mathbf{w}^{ij} + C_t^i \sum_{y_t=i} (\xi^{ij})_t + C_t^j \sum_{y_t=j} (\xi^{ij})_t \\ s.t. \quad & (\mathbf{w}^{ij})^T\phi(x_t) + b^{ij} \geqslant 1 - \xi_t^{ij}, \quad if \;\; y_t = i, \\ & (\mathbf{w}^{ij})^T\phi(x_t) + b^{ij} \leqslant -1 + \xi_t^{ij}, \quad if \;\; y_t = j, \\ & \xi_t^{ij} \geqslant 0.\end{aligned} \tag{5.3}$$

Here, x_t is the t^{th} data instance, y_t is its class label. $\phi(x_t)$ is a feature mapping to x_t. $\mathbf{w}^{ij}$ is the model parameter, b^{ij} is the bias term, and ξ^{ij} is the slack variable denoting the classification error. C_t^i and C_t^j are the weights for the t^{th} instance of i^{th} and j^{th} classes respectively. C_i^t and C_t^j are derived using the similarity function learned from the previous step; and they reflect the confidence values of the data instances x_t interpreted as being from the i^{th} class (*i.e.* activity) in the target domain. In other words, the pseudo training data point x_t is from the i^{th} class with confidence value of C_t^i. Therefore, in Eq. (5.3), the first term makes sure that in training the support vector machine the margin is maximized; the second and third terms controls the weighted classification errors for both classes. Intuitively, if the weight C_t^i is higher, the pseudo training data instance x_t from the i^{th} class are more trusted in training the SVM model.

After the optimization in Eq. (5.3) is solved, [26] uses a voting strategy for multi-class classification. In particular, each binary classification for the i^{th} and the j^{th} classes is considered to be a vote. Then, the votes can be cast for all data instances x_t, and in the end each x_t is designated to be in a class with maximum number of votes. In case that two classes have identical votes, the one with the smallest class index is simply chosen.

5.3.5 Cross-Domain Activity Recognition (CDAR) Algorithm

Finally, we summarize our CDAR method in Algorithm 2. As shown in Algorithm 2, at the first 3 steps, we extract the Web pages for each activity from both domains, and apply the information retrieval technique to transform the Web pages into tf-idf vectors. At step 4, after having the Web data (*i.e.* a set of tf-idf vectors) for each activity, we compute a similarity matrix for each pair of activities between the source domain and the target domain. At step 5, based on the learned similarities, we generate the pseudo training data by relabeling each training instance with the activity labels from the target domain. Each relabeled training instance is assigned with some confidence (weight), which equals to the similarity between its original activity label (from the source domain) and the newly given activity label (from the target domain). At step 6, we train the CDAR model using a weighted

Support Vector Machine. At step 7, we use the trained weighted SVM classifier to performing testing on the target domain's data.

Algorithm 2 Algorithm for CDAR

```
Input: Source domain has $T_1$ labeled training data $\mathcal{D}_{src} = \{(x_{src}^{(i)}, y_{src}^{(i)})\}_{i=1}^{T_1}$, where $y_{src}^{(i)} \in \mathcal{A}_{src}$. Target domain does not have any labeled training data; instead it has $T_2$ test data $\mathcal{D}_{tar} = \{(x_{tar}^{(j)}, y_{tar}^{(j)})\}_{j=1}^{T_2}$, where $y_{tar}^{(j)} \in \mathcal{A}_{tar}$ are the ground truth labels for testing only.
Output: Predicted labels on the test data in target domain.
begin
  1: For each activity $u \in \mathcal{A}_{src}$, extract a list of Web pages from some search engine (such as Google);
  2: For each $u$'s Web pages, apply information retrieval technique and transform each Web page to a tf-idf vector, and form a Web data set $\mathfrak{D}_u^w$;
  3: Similar to the above two steps, extract a Web data set $\mathfrak{D}_v^w$ for each activity $v \in \mathcal{A}_{tar}$;
  4: For any two activities $u \in \mathcal{A}_{src}$ and $v \in \mathcal{A}_{tar}$, calculate the similarity $sim(u, v) = MMD^2(\mathfrak{D}_u^w, \mathfrak{D}_v^w)$ using the maximum mean discrepancy in Eq.(5.2);
  5: Generate pseudo training data as follows: each training instance $(x_{src}^{(i)}, y_{src}^{(i)})$ with $y_{src}^{(i)} = u$ where $u \in \mathcal{A}_{src}$, is relabeled to get a set of pseudo training data as $\{(x_{src}^{(i)}, v_j, sim(u, v_j))|v_j \in \mathcal{A}_{tar}\}_{j=1}^{|\mathcal{A}_{tar}|}$ with $sim(u, v_j)$ as the confidences.
  6: Train the model CDAR with a weighted SVM [26] on the generated pseudo training data.
  7: Testing by the trained weighted-SVM classifier.
end
```

5.4 Proposed Approach: Transfer Between Different Feature Spaces

We first define our *transfer learning for activity recognition* problem setting. We study two domains that have different sets of sensors and different activity labels. Specifically, we have a source domain where the labeled sensor readings are in the form of $\{(\mathbf{x_s}, \mathbf{y_s})\}$, and a target domain where we assume that we have only the unlabeled data the form of $\{(\mathbf{x_t}\}$. The source domain label space is defined as $\mathcal{L}_s$ and the target domain label space is defined as $\mathcal{L}_t$. We make the assumption that $\mathcal{L}_s$ and $\mathcal{L}_t$ are different, but are related through a probability function $p(y_s, y_t)$ where y_s and y_t are source and target-domain activity labels, respectively. This probability function between the label spaces can be learned by labeling some of the target domain instances, or through the Web (as we do in this paper).

Our final goal is to estimate $p(\mathbf{y_t}|\mathbf{x_t})$. We know that:

$$p(\mathbf{y_t}|\mathbf{x_t}) = \sum_{\mathbf{c}^{(i)} \in \mathcal{L}_s} p(\mathbf{c}|\mathbf{x_t}) \cdot p(\mathbf{y_t}|\mathbf{c})$$

Since the activity-label spaces $\mathcal{L}_s, \mathcal{L}_t$ may be large, for simplicity, in this paper, we approximate the value of $p(\mathbf{y_t}|\mathbf{x_t})$ by the *mode* (the most frequent label) of $p(\mathbf{c}|\mathbf{x_t})$, where $\mathbf{c}$ is an activity label, and denote the mode as $\hat{\mathbf{c}}$. $\hat{\mathbf{c}}$ is labeled using labels from the source domain $\mathcal{L}_s$. In other words,

$$p(\mathbf{y_t}|\mathbf{x_t}) \approx p(\hat{\mathbf{c}}|\mathbf{x_t}) \cdot p(\mathbf{y_t}|\hat{\mathbf{c}}) \quad (\hat{\mathbf{c}} = \arg\max_{\mathbf{c} \in \mathcal{L}_s} p(\mathbf{c}|\mathbf{x_t}))$$

In this paper, since we assume the two label spaces to be different but related, the joint distribution $p(\mathbf{y_s}, \mathbf{y_t})$ should have high mutual information in general. Therefore, $p(\mathbf{y_t}|\hat{\mathbf{c}})$ should also be high.

From the above equation, our transfer learning framework takes two steps. In the first step, we will estimate $p(\hat{\mathbf{c}}|\mathbf{x_t})$ where $\hat{\mathbf{c}}$ is labeled using the source domain label space $\mathcal{L}_s$. Briefly speaking, we aim to use the source domain label space to explain the target domain sequences $\mathbf{x_t}$ first. Since the two domains have different feature spaces, in our first step we need to transfer across different feature spaces.

Next, we estimate $p(\mathbf{y_t}|\hat{\mathbf{c}})$ where $\mathbf{y_t}$ is defined on the target domain label space $\mathcal{L}_t$ and $\hat{\mathbf{c}}$ is defined on the source domain label space $\mathcal{L}_s$; *i.e.*, in our second step, we need to transfer across different label spaces.

5.4.1 Transfer Across Feature Spaces

Based on the above discussions, in this section we first need to transfer knowledge between different feature spaces and estimate $p(\hat{\mathbf{c}}|\mathbf{x_t})$. For each sensor reading $\mathbf{x_s}$ in the source domain $\mathcal{S}$, $\mathbf{x_s}$ is represented by features $f_\mathcal{S}$. Similarly, for each sensor reading $\mathbf{x_t}$ in the target domain $\mathcal{T}$, denote the features composing $\mathbf{x_t}$ as $f_\mathcal{T}$. For example, $f_\mathcal{S}$ can be an on-body 3D accelerometer attached to the wrist and $f_\mathcal{T}$ can be a the Wifi signals from a mobile phone. In this section, we build a bridge between $f_\mathcal{S}$ and $f_\mathcal{T}$.

We use a framework similar to translated learning [36]. When transferring the knowledge across different feature spaces, an important step is to find a *translator*

$\phi(f_t, f_s) \propto p(f_t|f_s)$ (Here f_s and f_t are features of the data in $\mathcal{S}$ and $\mathcal{T}$, respectively.) between the source and target domains. Since f_t and f_s are conditionally independent given x_s, we have:

$$\begin{aligned} p(f_t, f_s) &= \int_{\mathcal{X}_s} p(f_t|x_s)p(f_s|x_s)p(x_s)dx_s \\ &= \int_{\mathcal{X}_s} p(f_t, x_s)p(f_s|x_s)dx_s \end{aligned}$$

In order to measure the joint distribution $p(f_t, f_s)$, we need to measure $p(f_t, x_s)$, or more precisely, the joint distribution between each feature in $\mathcal{T}$ with the source domain sensor readings $\mathbf{x_s}$. In order to measure this joint distribution, depending on whether we compute based on the difference on distributions or difference on signal data, we can use two basic tools to approximate $p(f_t, x_s)$: Jeffrey's J-divergence [67] (the symmetric version of the KL-divergence) and Dynamic Time Warping [74].

We can extract two kinds of information from sensor readings. The first is that, given a sequence of sensor reading, we can try to estimate the generative distribution from which such a sensor reading is generated. Since we only care about the relative distance between two distributions of sensor readings instead of describing these distributions in high accuracy, we simply plot the frequency of each sensor value (discretize the sensor value if it is continuous), and then smooth the discretized probability distribution. Since we have quite different feature spaces, we first normalize all our sensor readings into the range of [0,1].

In particular, suppose that we have a training set in the source domain $\{x_i, y_i\}$, where x_i are sensor readings and y_i are target labels. For each activity y_i, we can select all sequences of sensor readings x that have y_i as its label. Next, we would count the occurrences of sensor values x_{ij}, and then estimate the probability distribution for each of the sensor in the sensor reading sequence x_i. An intuitive explanation of the above-mentioned method is that we try to link each generative distribution of different sensors to a target activity. We could imagine that we are trying to compose a dictionary where words in this dictionary are in fact distributions of sensor readings, and we attempt to tell the readers that "if you encounter such a distribution in your sensor readings, then it is possible that the sensor readings correspond to such an activity".

Following a similar approach, we can also estimate the probability distribution for each sensor reading sequence in the target domain. Now that for each sensor reading sequence, we have an estimated distribution Q and we wish to find a close distribution P in the source domain. Since KL divergence is asymmetric, *i.e.* $D_{KL}(P \parallel Q) \neq D_{KL}(Q \parallel P)$. Therefore, instead of calculating $D_{KL}(P \parallel Q)$, we use $D_{KL}(P \parallel Q) + D_{KL}(Q \parallel P)$, which is undoubtedly a symmetric measurement, to measure the distance between two distributions generating sensor readings.

Two issues need addressed for the selection of candidate labels based on relative entropy measurements alone. The first is that, although $D_{KL}(P \parallel Q) + D_{KL}(Q \parallel P)$ equals to zero if and only if the two distributions P and Q are identical, the fact that sensors have a very large value does not necessarily mean the two distributions are highly uncorrelated. Consider two accelerometers where the directions of accelerations are different. In this case, whenever the first accelerometer senses a high value, the second accelerometer will sense a low value. Therefore, we need to consider distribution pairs at both high divergence and low divergence values.

The second issue we consider is the different sampling rates of different sensors when plotting their signal values versus time. Different kinds of sensors have very different sampling rates and the accuracy of distributions estimated can vary a lot. When calculating the correlation between different sensors, another important step is to use a distance metric that can take different sampling rates into account. Now given two series of sensor readings of only one dimension: Q and C of length n and m, we wish to align two sequences use dynamic time warping (DTW) [74]. The idea of DTW is simple. We could construct an n-by-m matrix where the element at (i, j) contains the distance $d(q_i, c_j)$ between the two points q_i and c_j, which is measured as the absolute value of difference of q_i and c_j: $|q_i - c_j|$. Since each element (i, j) corresponds to the alignment between q_i and c_j. Our objective is to find a warping path W which is a contiguous set of matrix elements that defines mapping between Q and C. Thus, the element at position K of the warping path W is defined as $w_k = (i, j)_k$. This warping path can be found using dynamic programming under a quadratic time complexity.

Algorithm 1 shows the step for projecting the labels in the source domain to the unlabeled sensor readings in the target domain. Notice that in this algorithm, we had introduced a parameter K, which is used to control the number of candidate label sequences in the source domain. In our experiments, we would test how variations of this parameter K would affect the overall algorithm performance.

Algorithm 3 Projecting the labels in the source domain to the unlabeled sensor readings in the target domain

```
Input: Source domain activities $\S_s$, source domain data $\mathcal{D}_s = \{(\mathbf{x}_s, \mathbf{y}_s)\} = \{(x_i, y_i) | y_i \in \mathcal{L}_s\}$, target domain data $\mathcal{D}_t = \{(\mathbf{x}_t)\}$
Output: Pseudo-labeled target domain data: $\mathcal{D}'_t = \{(\mathbf{x}_s, \mathbf{y}'_s\}$
begin
  Normalize each sensor reading sequence both in $\mathcal{S}$ and $\mathcal{T}$.
  For each pair of sensor reading and activity in $(\mathbf{x}_s, \mathbf{y}_s) \in \mathcal{S}$, estimate its probability distribution $p(f_s|y_s)$.
  For each unlabeled sequence in the target domain $\mathbf{x}_t$, estimate the distribution of its feature values: $P(f_t)$.
  Calculate the relative entropy between distributions in $\mathcal{T}$ and all the distributions in $\mathcal{S}$. Take the top-$K$ similar and the bottom-$K$ similar distributions out and record their labels as candidates.
  Calculate the DTW score between this sensor reading sequence $\mathbf{x}_t$ and all the labeled sensor reading sequences $(\mathbf{x}_s, \mathbf{y}_s)$ in the source domain. Take the top-$K$ highest and the bottom-$K$ lowest similar sensor readings out and record their labels as candidates.
  Label this unlabeled sequence $\mathbf{x}_t$ with the label that appeared maximum times in the candidate label set.
end
```

5.4.2 Transfer Across Label Spaces

In our previous subsection, we had already estimated the value for $\arg\max_{\mathbf{c}} p(\hat{\mathbf{c}}|\mathbf{x_t})$. In this subsection, we aim to estimate $p(\mathbf{y_t}|\hat{\mathbf{c}})$, we have:

$$p(\mathbf{y_t}|\mathbf{c}) = p(\mathbf{y_t}, \mathbf{c})/p(\mathbf{c})$$

If we assume that there is no distinction between the prior distribution $p(\mathbf{c})$, then we have $p(\mathbf{y_t}|\mathbf{c}) \propto p(\mathbf{y_t}, \mathbf{c})$.

Based on the Markov assumption, we have:

$$\begin{aligned} p(\mathbf{y_t}, \mathbf{c}) &= p(y_t^0) \prod_i p(y_t^i | y_t^{i-1}) \prod_i p(c^i | y_t^i) \\ &\propto \prod_i p(y_t^i | y_t^{i-1}) \prod_i p(c^i | y_t^i) \\ \log p(\mathbf{y_t}, \mathbf{c}) &\propto \sum_i \log p(y_t^i | y_t^{i-1}) + \sum_i \log p(c^i | y_t^i) \end{aligned}$$

From the above formulation, we can see that such a problem can be reduced to estimating $p(l_s|l_t)$, where $l_s \in \mathcal{L}_s, l_t \in \mathcal{L}_t$ and $p(l_t^1|l_t^2)$, where $l_t^1, l_t^2 \in \mathcal{L}_t$. Since the number of labeled training data in the target domain is not sufficient, we need extra knowledge sources to estimate such probabilities.

For example, in [135], the authors used Web pages from Open Directory Project (ODP) as a bridge to estimate the probabilities. In [171], the authors tried to calculate the cosine similarity of two word vectors, which are composed by the words of the Web search results when two activity names are used as queries and issued as input. In practice, such algorithms based on words from Web pages could be extremely slow. Instead of measuring the conditional probabilities directly, we choose to optimize a similar measurement that intrinsically can be optimized similarly as $p(\mathbf{y_t}, \mathbf{c}$, stated below.

We define $R(i, j)$ as the expected loss of assigning $j \in \mathcal{L}_t$ to y_t^i. $Q(l_1, l_2)$ as the "information distance" between l_1 and l_2, which are activity labels from the source and target domains, respectively. Then $R(i, j)$ is defined recursively as:

$$R(i, j) = \min_{k \in \mathcal{L}_t} \{R(i-1, k) + Q((\hat{\mathbf{c}})^i, j) + Q(k, j)\}$$

We briefly explain the nature of this recurrence relation. In order to minimize the loss up to time slice i, we need to first consider the minimum loss up to time slice $i - 1$. We need to enumerate all possible $R(i - 1, k)$, where $k \in \mathcal{L}_t$ is the label we assigned to time slice $i - 1$. Next, we need to minimize the distance between the original "pseudo-label" $\hat{\mathbf{c}}^i$ and this new label $j \in \mathcal{L}_t$. Furthermore, $Q(k, j)$ is also considered in the loss function to minimize the distance between successive slices ${\mathbf{y_t}}^i$ and ${\mathbf{y_t}}^{i-1}$. It can be seen that the above recurrence relation could be solved using dynamic programming. In this paper, we use the Google Similarity Distance [34] as Q to approximate the information distance between two entities.

The definition of Google similarity difference is as follows:

$$NGD(x, y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log N - \min\{\log f(x), \log f(y)\}}$$

where $f(x)$ denotes the number of pages containing x, and $f(x, y)$ denotes the number of pages containing both x and y, as reported by Google. N is just a normalized factor that does not affect whether x is closer to y or z. Therefore, what we need to know is just a count of the search results. A detailed explanation of the *Google distance* is beyond the scope of this paper and we encourage readers to read [34] for technical details.

By using the normalized Google distance, our loss function becomes:

$$R(i, j) = \min_{k \in \mathcal{L}_t} \{R(i-1, k) + NGD((\hat{\mathbf{c}})^i, j) + NGD(k, j)\}$$

Algorithm 4 Projecting target domain sequences with source domain labels to target domain sequences with target domain labels

```
Input: Pseudo-labeled target domain data $\mathcal{D}'_t = \{(\mathbf{x}_t, \hat{\mathbf{c}})\}$
Output: Labeled target domain data: $\mathcal{D}^*_t = \{(\mathbf{x}_t, \mathbf{y}_t)\}$
begin
  1: For each pseudo-labeled target domain instance $d'_t$, calculate its minimum loss value $R(i,j)$ based on the recurrence relation $R(i,j) = \min_{k \in \mathcal{L}_t}\{R(i-1,k) + NGD((\hat{\mathbf{c}})^i, j) + NGD(k,j)\}$, where $NGD$ is the Google similarity distance metric.
  2: Relabel this $d'_t$ using the labels in the target domain label space, thereby creating a new sequence $d^*_t$.
end
```

Algorithm 4 shows our transferring procedure. After these two steps, we now have the labels $y^i_t \in L_t$ for each unlabeled sensor reading in the target domain, and we can apply *any* machine learning algorithms used for activity recognition such as hidden Markov models (HMM) [118] or conditional random fields (CRF) [156], to train activity recognition classifiers in the target domain.

5.5 Experimental Results

In this section, we investigate how our algorithm performs in several real-world activity recognition domains. Specifically, we test the recognition accuracy of our algorithm when transferring across different feature and label spaces.

5.5.1 Datasets and Evaluation Criteria

In this paper, we use three real-world activity recognition datasets to validate our algorithm. Our first dataset (UvA in short) [1] is from [158] where a dataset is recorded in the house of a 26-year-old man, living alone in a three-room apartment where 14 state-change sensors are installed. The second dataset we use is the MIT PLIA1 dataset [2] [63], which was recorded on March 4, 2005 from 9AM to 1PM in the MIT PlaceLab. The third dataset is from [118] (Intel in short), which aims to recognize 12 routine morning activities based on RFID sensors.

[1] http://staff.science.uva.nl/∼tlmkaste/research/software.php

[2] http://architecture.mit.edu/house_n/data/PlaceLab/PLIA1.htm

5.5.2 Baseline

To allow a better comparison of our algorithm performance against state-of-the-art research in activity recognition, we compared against an unsupervised activity recognition algorithm described in [165]. Briefly, in [165] describes an unsupervised activity recognition algorithm that can infer the activities being performed based on object names involved in the activities. Notice that, since algorithms described in [171] and [157] have different problem settings compared to our paper, (the former assumes different label space but same feature space and the latter assumes the meta-feature space is constructed manually), we cannot use their algorithms as baselines for comparison.

5.5.3 Different Features and Same Labels

In our first experiment, we aim to validate the effectiveness of our algorithm when transferring knowledge between different feature spaces. More precisely, in all of the three datasets we used, we divide the feature space into two. Half of the sensor readings are used as data in the source domain and the remaining half are used as data in the target domain. Since this split of source and target domains are done manually, the label space is still the same.

K	UvA Acc(Var)	Intel Acc(Var)
K = 5	55.8% (5.1%)	52.1% (4.7%)
K = 10	58.2% (4.3%)	53.4% (4.5%)
K = 15	67.3% (4.1%)	55.3% (3.8%)
K = 20	**68.2% (4.0%)**	**57.2% (4.2%)**
Unsupervised	47.3%(4.1%)	42.8% (3.8%)

Table 5.1: Algorithm Performance on UvA and Intel Dataset

Table 5.1 shows our algorithm performance on the UvA dataset and Intel dataset. We have repeated the splitting process for ten times and both the average accuracy and the variance are reported. We also report our algorithm performance by varying the parameter K. Recall that K is the parameter we use to control our candidate "label set size". More precisely, we select both the top-K similar sensor reading distributions and bottom-K dissimilar sensor reading distributions, as well as the top-K minimum DTW score sensor reading sequences, a total of $3K$ sensor readings in the source domain and their corresponding labels in the source domain, and put these labels in the candidate set.

Our result in Table 5.1 shows that our algorithm could consistently outperform the unsupervised activity recognition approach. We also observe that that with the increase of K, the accuracy also increases whereas the variance is also consistently decreasing. This is due to the fact more candidate labels are taken into account and therefore we could expect to consider more "probable" labels and our assignment of labels could be more precise. However, when K is larger than 20, performance starts to converge and also drops slightly. Therefore, we end by reporting our best result, which is achieved at $K = 20$.

89 activities are included in the MIT PLIA1 dataset and a taxonomy could be built to describe these activities [63]. In MIT PLIA1 dataset, we analyze how the performance will be when we use the activities under the same category as both the sensor readings in the source domain and in the target domain. The same splitting process that was applied to the UvA and the Intel datasets is also applied to the sensor readings under each category to split the source domain and the target domain. The MIT PLIA1 dataset can be categorized into 9 subcategories, including cleaning indoor, yardwork, laundry, dishwashing, meal preparation, hygiene, grooming, personal and leisure. We report the accuracy and variance our algorithm had achieved in each subcategory in Table 5.2.

Category	K = 5 Acc (Var)	K = 10 Acc (Var)	K = 15 Acc (Var)	Unsupervised Acc (Var)
Cleaning Indoor	**68.5%(2.4%)**	62.5%(2.9%)	61.8%(2.8%)	50.7%(2.6%)
Yardwork	52.3%(2.9%)	**69.1%(3.9%)**	55.8%(2.8%)	51.8%(2.1%)
Laundry	**69.3%(3.3%)**	60.6%(2.3%)	50.5%(2.9%)	56.8%(3.2%)
Dishwashing	51.9%(2.6%)	**69.7%(3.6%)**	57.8%(3.5%)	68.2%(2.9%)
Meal Preparation	53.3%(3.1%)	63.7%(3.0%)	64.4%(2.9%)	**68.6%(2.9%)**
Hygiene	56.3%(2.5%)	**62.4%(3.7%)**	52.6%(2.9%)	54.2%(3.6%)
Grooming	59.6%(3.9%)	65.1%(3.0%)	57.2%(2.7%)	**69.9%(2.7%)**
Personal	**68.6%(3.2%)**	59.2%(3.4%)	57.8%(2.3%)	68.2%(3.1%)
Leisure	59.3%(3.9%)	57.2%(3.2%)	65.3%(2.6%)	**65.9%(3.2%)**

Table 5.2: Algorithm Performance on MIT PLIA1 Dataset

From Table 5.2, we can also see that our transfer learning activity recognition algorithm outperforms the unsupervised baseline in most cases and achieves comparable performance with the unsupervised baseline in other subcategories. For the choice of parameter K, we could see it exhibits a very different behavior as in Table 5.1. Generally speaking, the best performance is usually achieved when K is small. One possible explanation for this phenomenon is that in MIT PLIA1 dataset, since the dataset size is relatively large, the probability distribution estimated is relatively more accurate than the UvA or the Intel dataset, and therefore it is possible to achieve a much better performance with a smaller K. However, when K is larger, more noisy sensor readings are induced.

5.5.4 Different Features and Labels

In this experiment, we use the full MIT PLIA1 dataset as the source domain and then try to transfer to both the UvA dataset and the Intel dataset. Such two domains have their data reduced to the same subfeature representations (similar sensors) before transferring. We use such a way to validate our algorithm since we believe the direction of "transfer" is especially important since the size of UvA or Intel dataset will not contain enough knowledge from which we could transfer to the MIT dataset. Since the dimension of the feature space (number of sensors) and the dimension of the label space (number of activities) in MIT PLIA1 dataset are both significantly larger than those of UvA and Intel datasets, we choose to transfer from PLIA1 to UvA and Intel.

K	MIT → UvA Acc(Var)
K = 5	**59.8% (4.2%)**
K = 10	57.5% (4.1%)
K = 15	51.0% (4.8%)
K = 20	41.0% (4.1%)
Unsupervised	47.3%(4.1%)

Table 5.3: Algorithm performance of transferring knowledge from MIT PLIA1 to UvA dataset

K	MIT → Intel Acc(Var)
K = 5	60.5% (4.2%)
K = 10	**61.2% (3.8%)**
K = 15	53.2% (4.1%)
K = 20	42.0% (2.5%)
Unsupervised	42.8%(3.8%)

Table 5.4: Algorithm performance of transferring knowledge from MIT PLIA1 to Intel dataset

The results in Table 5.3 and 5.4 have validated that our approach of transferring knowledge across feature space and label space is effective.

Furthermore, since our algorithm relies on the Hidden Markov Model as the main model for inference, the inference steps usually cost less than one second and the training phase usually cost less than two minutes in all of our experiments conducted, no matter whether we've used the Jeffrey's J-divergence or the Dynamic Time Warping algorithm in the feature transfer phase. (But typically the DTW algorithm has a more significant time cost compared to calculating Jeffrey's J-divergence). Therefore, we can also confirm that our transfer learning algorithm for activity recognition also fits the need for real-world deployment.

5.6 Summary and Discussion

In this chapter, we have proposed an approach to solve the activity recognition problem under the transfer learning setting. By comparing our method with many previous solutions which also attempt to solve the activity recognition problem under a transfer learning setting, we can see that our method does not have many of the limitations which have been encountered in previous papers. The fundamental assumption of our paper by transferring the knowledge across different feature spaces is that although the kinds of sensors we may encounter are highly different, the distributions may be similar and we can exploit such knowledge for building a bridge across domains. Furthermore, when handling the case of different label space across two domains, we can alleviate this problem to estimating the conditional distribution of two label spaces and then use Web knowledge as a tool to help estimate such a value. We have validated our approach in several real-world sensor-based activity recognition tasks and have demonstrated the effectiveness of our algorithm compared to unsupervised activity recognition methods.

We plan to extend our work in the following directions. Firstly, we wish to study the detailed constraints under which our algorithm would work. Stating the correlation between source domain and target domain as "different but related" is difficult to judge in reality. Setting an accurate distance metric or constraint would be nicer for end users to judge whether the two domains can be used for transfer. Secondly, when we perform the transfer step, we have used $\hat{\mathbf{c}}$ as an approximation of all possible labeling sequences from the source domain label space. Different from other machine learning methods which use modes to approximate integrals or summations, the approximation ratio of such a method is not satisfactory. In our future study, we plan to choose a candidate pseudolabeling set, instead of a $\hat{\mathbf{c}}$ alone, to study the effect of transfer.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, we study the problem of recognizing human activity, which is an important research topic in ubiquitous computing, artificial intelligence, web mining and social network analysis. In recognizing human activities in the physical world, we study the problem of how to recognize human activities when the relationships between activities are complex; how to detect abnormal physical activities when the training data is sparse and how to transfer useful knowledge from one activity recognition domain to another. We also show that one simple extension of recognizing human activities with concurrent and interleaving goals can be applied to the problem of context-aware query classification. For each problem we've discussed, we review the background knowledge and related work. Next we analyze the challenges of these specific problems and we propose our own solutions to solve these challenging problems.

In particular, for recognizing activities with concurrent and interleaving goals, we demonstrate that previous activity recognition methods did not take such relationships between activities into consideration when developing their methods and hence will fail in complex real-world scenarios. Next, we propose a collective inference model based on the Conditional Random Field to recognize interleaving activities. We estimate the correlation between goals by constructing a goal correlation graph automatically.

For detecting abnormal human activities, we point out that one of the most critical challenges in detecting abnormal activities is the impossibility of predefining a number of states for a possible human activity recognition scenario. We develop a HDP-HMM model which can specifically tackle this challenge. Such a hybrid model is followed by one-class SVM to automatically learn useful information from the limited sensor information we gain from the abnormal activities.

Next, for transferring activity recognition model from one domain to the other, we exemplify the necessity of applying transfer learning framework into the activity recognition problem when we try to deploy an activity recognition problem to the real-world since one cannot expect we can have the same sensor configuration or the activity description as what we have in the training data when building the classifier. One of our preliminary work is to develop a cross-domain activity recognition algorithm to automatically bridge the gap between recognizing different activities using Web data. In this thesis, we've developed an algorithm that can further build an automatic correlation between different types of sensors and transfer across different sensing modalities. For all the abovementioned three problems in recognizing human activities in the physical worlds, we've evaluated our proposed algorithms on several real-world sensor-based activity recognition datasets, and show we are able to obtain reasonable recognition performances.

The context-aware query classification problem we've discussed share the same merit with recognizing human activities with concurrent and interleaving goals in the physical world. We've demonstrated that in practice, when we take the contextual information from the query sessions, we'll be able to obtain much more satisfying recognition accuracy to understand the "true needs" of Web users. The algorithm we propose is also based on the Conditional Random Field model but we've developed new contextual features as well as taxonomical features to fit the specific characteristics we have in the Web domain. We've evaluated our algorithm on query logs from a large-scale commercial search engine and have demonstrated its practical effectiveness.

6.2 Future Work

In addition to the possible future work directions we have mentioned in each chapter, we will continue to explore some other possible research opportunities along the following directions:

- *Hybrid recognition framework across two worlds.* In our thesis, we've discussed several possible approaches for recognizing human activities from sensor information. It seems natural for us to use such information to help aid recognizing physical activities and provide useful recommendation advice. For example, if we can learn that a user is a student based on his online check in history. It seems not necessary for us to provide recommendations of luxury goods to him / her. Zheng et al. [176, 175, 170] have pursued work along this direction of "collaborative recommendation". However, their work is based on physical trajectories received from GPS information. A hybrid recognition framework that encompasses rich information from both physical and social worlds would be quite appealing.

- *Energy-aware Activity Recognition.* Although we've enumerated some challenging problems we are facing when trying to deploy an activity recognition framework to the real world, one problem we have not yet discussed is the energy problem. It is a natural expectation for us to deploy our activity recognition application or systems on smartphones due to the ubiquitous usage of smartphones nowadays and their convenience. However, continuous sensing would be a heavy burden for current smartphone batteries. Developing algorithms that can take energy factors into consideration is also quite important and challenging. Recently, there has been some works in this area. For example, Li et al. [89] try to infer the status of high-energy-consuming sensors according to the outputs of software-based sensors and the physical sensors that are necessary to work all the time for supporting the basic functions of mobile devices. Lu et al. [97] developed a speaker identification prototype that uses a heterogeneous multi-processor hardware architecture that splits computation between a low power processor and the phone's application processor to enable continuous background sensing with minimal power requirements. Nevertheless, most of the research in this area are still confining their experimental settings to a very preliminary stage where there are only a limited number of actions and high granularity. Therefore, in our opinion, such a direction is also important for future research and also for putting activity recognition into real-world practice.

- *Activity Prediction.* Most of the approaches as well as the solutions we've discussed and proposed fall into the category of *activity recognition*. In practice, we are also interested into learning what a person will be doing given some sensor readings. Will the user be going to the gym after work or going to a restaurant for dinner today? Or will she be going shopping given that it's now Black Friday? Such questions enable the possibility of providing context-aware advertisements to the specific users and would be extremely useful for understanding human intentions, which is one level higher than recognizing human activities, in the real-world.

REFERENCES

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *SIGMOD Conference*, pages 207–216. ACM Press, 1993.

[2] Corin R. Anderson, Pedro Domingos, and Daniel S. Weld. Relational markov models and their application to adaptive web navigation. In *KDD*, pages 143–152. ACM, 2002.

[3] Dorit Avrahami-Zilberbrand and Gal A. Kaminka. Fast and complete symbolic plan recognition. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI*, pages 653–658. Professional Book Center, 2005.

[4] Lars Backstrom, Eric Sun, and Cameron Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *WWW*, pages 61–70. ACM, 2010.

[5] Ling Bao and Stephen S. Intille. Activity recognition from user-annotated acceleration data. In Alois Ferscha and Friedemann Mattern, editors, *Pervasive*, volume 3001 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2004.

[6] Jennifer Beaudin, Stephen S. Intille, and Emmanuel Munguia Tapia. Lessons learned using ubiquitous sensors for data collection in real homes. In Elizabeth Dykstra-Erickson and Manfred Tscheligi, editors, *CHI Extended Abstracts*, pages 1359–1362. ACM, 2004.

[7] Michael Beetz, Nico von Hoyningen-Huene, Jan Bandouch, Bernhard Kirchlechner, Suat Gedikli, and Alexis Maldonado. Camera-based observation of football games for analyzing multi-agent activities. In *AAMAS*, pages 42–49, 2006.

[8] Steven M. Beitzel, Eric C. Jensen, Abdur Chowdhury, David A. Grossman, and Ophir Frieder. Hourly analysis of a very large topically categorized web query log. In Mark Sanderson, Kalervo Järvelin, James Allan, and Peter Bruza, editors, *SIGIR*, pages 321–328. ACM, 2004.

[9] Inderpal S. Bhandari, Edward Colet, Jennifer Parker, Zachary Pines, Rajiv Pratap, and Krishnakumar Ramanujam. Advanced scout: Data mining and knowledge discovery in nba data. *Data Min. Knowl. Discov.*, 1(1):121–125, 1997.

[10] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[11] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[12] Vincent D. Blondel, Anahí Gajardo, Maureen Heymans, Pierre Senellart, and Paul Van Dooren. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM Review*, 46:647–666, 2004.

[13] Avrim Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artif. Intell.*, 97(1-2):245–271, 1997.

[14] Avrim Blum and Tom M. Mitchell. Combining labeled and unlabeled sata with co-training. In *COLT*, pages 92–100, 1998.

[15] K.M. Borgwardt, A. Gretton, M.J. Rasch, H.P. Kriegel, B. Schölkopf, and A. Smola. Integrating stuctured biological data by kernel maximum mean discrepancy. In *Proceedings of the 14th International Conference on Intelligent Systems for Molecular Biology*, pages 49–57, 2006.

[16] Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[17] Joseph K. Bradley, Aapo Kyrola, Danny Bickson, Carlos Guestrin, and Carlos Guestrin. Parallel coordinate descent for l1-regularized loss minimization. *CoRR*, 2011.

[18] Helene Brashear, Thad Starner, Paul Lukowicz, and Holger Junker. Using multiple sensors for mobile sign language recognition. In *ISWC*, pages 45–52, 2003.

[19] Barry Brumitt, Brian Meyers, John Krumm, Amanda Kern, and Steven A. Shafer. Easyliving: Technologies for intelligent environments. In *HUC*, pages 12–29, 2000.

[20] Hung Hai Bui. A general model for online probabilistic plan recognition. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 1309–1318, 2003.

[21] Hung Hai Bui, Dinh Q. Phung, Svetha Venkatesh, and Hai Phan. The hidden permutation model and location-based activity recognition. In *AAAI*, pages 1345–1350, 2008.

[22] Hung Hai Bui, Svetha Venkatesh, and Geoff A. W. West. Policy recognition in the abstract hidden markov model. *J. Artif. Intell. Res. (JAIR)*, 17:451–499, 2002.

[23] Huanhuan Cao, Derek Hao Hu, Dou Shen, Daxin Jiang, Jian-Tao Sun, Enhong Chen, and Qiang Yang. Context-aware query classification. In James Allan, Javed A. Aslam, Mark Sanderson, ChengXiang Zhai, and Justin Zobel, editors, *SIGIR*, pages 3–10. ACM, 2009.

[24] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. Context-aware query suggestion by mining click-through and session data. In *KDD*, pages 875–883, 2008.

[25] Xiaoyong Chai and Qiang Yang. Multiple-goal recognition from low-level signals. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI 2005)*, pages 3–8, 2005.

[26] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001.

[27] Oliver Chapelle, Bernhard Scholkopf, and Alexander Zien. *Semi-Supervised Learning*. MIT Press, 2006.

[28] Eugene Charniak and Robert P. Goldman. A bayesian model of plan recognition. *Artif. Intell.*, 64(1):53–79, 1993.

[29] Eugene Charniak and Drew McDermott. *Introduction to Artificial Intelligence*. Addison-Wesley, 1985.

[30] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *CIKM*, pages 759–768, 2010.

[31] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *KDD*, pages 1082–1090, 2011.

[32] Tanzeem Choudhury, Gaetano Borriello, Sunny Consolvo, Dirk Hähnel, Beverly L. Harrison, Bruce Hemingway, Jeffrey Hightower, Predrag V. Klasnja, Karl Koscher, Anthony LaMarca, James A. Landay, Louis LeGrand, Jonathan Lester, Ali Rahimi, Adam Rea, and Danny Wyatt. The mobile sensing platform: An embedded activity recognition system. *IEEE Pervasive Computing*, 7(2):32–41, 2008.

[33] Tanzeem Choudhury and Alex Pentland. Sensing and modeling human networks using the sociometer. In *ISWC*, pages 216–222, 2003.

[34] Rudi Cilibrasi and Paul M. B. Vitányi. The google similarity distance. *IEEE Trans. Knowl. Data Eng.*, 19(3):370–383, 2007.

[35] Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. Probabilistic query expansion using query logs. In *WWW*, pages 325–332, 2002.

[36] Wenyuan Dai, Yuqiang Chen, Gui-Rong Xue, Qiang Yang, and Yong Yu. Translated learning: Transfer learning across different feature spaces. In *NIPS*, pages 353–360, 2008.

[37] Arnaud Doucet, Nando de Freitas, Kevin P. Murphy, and Stuart J. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *UAI*, pages 176–183, 2000.

[38] Thi V. Duong, Hung Hai Bui, Dinh Q. Phung, and Svetha Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *CVPR (1)*, pages 838–845, 2005.

[39] Nathan Eagle and Alex Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

[40] Assaf Feldman, Emmanuel Munguia Tapia, Sajid Sadi, Pattie Maes, and Chris Schmandt. Reachmedia: On-the-move interaction with everyday objects. In *ISWC*, pages 52–59, 2005.

[41] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, 32(1):41–62, 1998.

[42] Bruno M. Fonseca, Paulo Braz Golgher, Bruno Pôssas, Berthier A. Ribeiro-Neto, and Nivio Ziviani. Concept-based interactive query expansion. In *CIKM*, pages 696–703, 2005.

[43] Christopher W. Geib, John Maraist, and Robert P. Goldman. A new probabilistic plan recognition algorithm based on string rewriting. In *ICAPS*, pages 91–98, 2008.

[44] Zoubin Ghahramani and Michael I. Jordan. Factorial hidden markov models. *Machine Learning*, 29(2-3):245–273, 1997.

[45] Dani Goldberg and Maja J. Mataric. Augmented markov models. Technical report, University of Southern California, 1999.

[46] Dani Goldberg and Maja J. Mataric. Coordinating mobile robot group behavior using a model of interaction dynamics. In *Agents*, 1999.

[47] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[48] Kwun Han and Manuela Veloso. Automated robot behavior recognition applied to robotic soccer. In *IJCAI-99 Workshop on Team Behaviors and Plan Recognition*, 1999.

[49] Derek Hao Hu, Sinno Jialin Pan, Vincent Wenchen Zheng, Nathan Nan Liu, and Qiang Yang. Real world activity recognition with multiple goals. In Hee Yong Youn and We-Duke Cho, editors, *UbiComp*, volume 344 of *ACM International Conference Proceeding Series*, pages 30–39. ACM, 2008.

[50] Derek Hao Hu, Dou Shen, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Context-aware online commercial intention detection. In Zhi-Hua Zhou and Takashi Washio, editors, *ACML*, volume 5828 of *Lecture Notes in Computer Science*, pages 135–149. Springer, 2009.

[51] Derek Hao Hu and Qiang Yang. Cigar: Concurrent and interleaving goal and activity recognition. In Dieter Fox and Carla P. Gomes, editors, *AAAI*, pages 1363–1368. AAAI Press, 2008.

[52] Derek Hao Hu and Qiang Yang. Transfer learning for activity recognition via sensor mapping. In Toby Walsh, editor, *IJCAI*, pages 1962–1967. IJCAI/AAAI, 2011.

[53] Derek Hao Hu, Xian-Xing Zhang, Jie Yin, Vincent Wenchen Zheng, and Qiang Yang. Abnormal activity recognition based on hdp-hmm models. In Craig Boutilier, editor, *IJCAI*, pages 1715–1720, 2009.

[54] Derek Hao Hu, Xian-Xing Zhang, Jie Yin, Vincent Wenchen Zheng, and Qiang Yang. Abnormal activity recognition based on hdp-hmm models. In *IJCAI*, page To appear, 2009.

[55] Derek Hao Hu, Vincent Wenchen Zheng, and Qiang Yang. Cross-domain activity recognition via transfer learning. *Pervasive and Mobile Computing*, 7(3):344–358, 2011.

[56] Marcus J. Huber, Edmund H. Durfee, and Michael P. Wellman. The automated mapping of plans for plan recognition. In *AAAI*, page 1460, 1994.

[57] Tâm Huynh, Ulf Blanke, and Bernt Schiele. Scalable recognition of daily activities with wearable sensors. In *LoCA*, pages 50–67, 2007.

[58] Tâm Huynh, Mario Fritz, and Bernt Schiele. Discovery of activity patterns using topic models. In *UbiComp*, pages 10–19, 2008.

[59] Tâm Huynh and Bernt Schiele. Analyzing features for activity recognition. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*, 2005.

[60] Stephen S. Intille and Aaron F. Bobick. A framework for recognizing multi-agent action from visual evidence. In *AAAI/IAAI*, pages 518–525, 1999.

[61] Stephen S. Intille and Aaron F. Bobick. Recognizing planned, multiperson action. *Computer Vision and Image Understanding*, 81(3):414–445, 2001.

[62] Stephen S. Intille, Kent Larson, Jennifer Beaudin, Jason Nawyn, Emmanuel Munguia Tapia, and Pallavi Kaushik. A living laboratory for the design and evaluation of ubiquitous computing technologies. In *CHI Extended Abstracts*, pages 1941–1944, 2005.

[63] Stephen S. Intille, Kent Larson, Emmanuel Munguia Tapia, Jennifer Beaudin, Pallavi Kaushik, Jason Nawyn, and Randy Rockinson. Using a live-in laboratory for ubiquitous computing research. In *Pervasive*, pages 349–365, 2006.

[64] Yuri A. Ivanov and Aaron F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):852–872, 2000.

[65] Tommi Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. In *NIPS*, pages 487–493, 1998.

[66] Peter Jarvis, Teresa F. Lunt, and Karen L. Myers. Identifying terrorist activity with ai plan recognition technology. In *AAAI*, pages 858–863, 2004.

[67] Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 1946.

[68] Marko Jug, Janez Pers, Branko Dezman, and Stanislav Kovacic. Trajectory based assessment of coordinated human activity. In *ICVS*, pages 534–543, 2003.

[69] Holger Junker, Paul Lukowicz, and Gerhard Tröster. Padnet: Wearable physical activity detection network. In *ISWC*, pages 244–245, 2003.

[70] Gal A. Kaminka and Michael H. Bowling. Towards robust teams with many agents. In *AAMAS*, pages 729–736, 2002.

[71] Gal A. Kaminka and Milind Tambe. Robust agent teams via socially-attentive monitoring. *J. Artif. Intell. Res. (JAIR)*, 12:105–147, 2000.

[72] In-Ho Kang and Gil-Chang Kim. Query type classification for web document retrieval. In *SIGIR*, pages 64–71, 2003.

[73] Henry Kautz. *A Formal Theory of Plan Recognition*. PhD thesis, University of Rochester, 1987.

[74] Eamonn J. Keogh and Michael J. Pazzani. Scaling up dynamic time warping for datamining applications. In *KDD*, pages 285–289, 2000.

[75] Nicky Kern, Stavros Antifakos, Bernt Schiele, and Adrian Schwaninger. A model for human interruptability: Experimental evaluation and automatic estimation from wearable sensors. In *ISWC*, pages 158–165, 2004.

[76] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1-2):273–324, 1997.

[77] Moritz Köhler, Shwetak N. Patel, Jay Summet, Erich P. Stuntebeck, and Gregory D. Abowd. Tracksense: Infrastructure free precise indoor positioning using projected patterns. In *Pervasive*, pages 334–350, 2007.

[78] Gregory Kuhlmann, William B. Knox, and Peter Stone. Know thine enemy: A champion robocup coach agent. In *AAAI*, 2006.

[79] Sanjiv Kumar and Martial Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *ICCV*, pages 1150–1159, 2003.

[80] Kristof Van Laerhoven and Hans-Werner Gellersen. Spine versus porcupine: A study in distributed wearable activity recognition. In *ISWC*, pages 142–149, 2004.

[81] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001.

[82] John D. Lafferty, Xiaojin Zhu, and Yan Liu. Kernel conditional random fields: representation and clique selection. In *ICML*, 2004.

[83] Niels Landwehr. Modeling interleaved hidden processes. In *ICML*, pages 520–527, 2008.

[84] John Langford, Lihong Li, and Tong Zhang. Sparse online learning via truncated gradient. In *NIPS*, pages 905–912, 2008.

[85] Aleksandar Lazarevic, Levent Ertöz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. In *SDM*, 2003.

[86] Scott Lenser and Manuela M. Veloso. Non-parametric time series classification. In *ICRA*, pages 3918–3923, 2005.

[87] Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *IJCAI*, pages 766–772, 2005.

[88] Xiao Li, Ye-Yi Wang, and Alex Acero. Learning query intent from regularized click graphs. In *SIGIR*, pages 339–346, 2008.

[89] Xueying Li, Huanhuan Cao, Enhong Chen, and Jilei Tian. Learning to infer the status of heavy-duty sensors for energy-efficient context-sensing. *ACM TIST*, 3(2):35, 2012.

[90] Lin Liao, Tanzeem Choudhury, Dieter Fox, and Henry A. Kautz. Training conditional random fields using virtual evidence boosting. In *IJCAI*, pages 2530–2535, 2007.

[91] Lin Liao, Tanzeem Choudhury, Dieter Fox, and Henry A. Kautz. Training conditional random fields using virtual evidence boosting. In *IJCAI*, pages 2530–2535, 2007.

[92] Lin Liao, Dieter Fox, and Henry A. Kautz. Learning and inferring transportation routines. In *AAAI*, pages 348–353, 2004.

[93] Lin Liao, Dieter Fox, and Henry A. Kautz. Location-based activity recognition using relational markov networks. In *IJCAI*, pages 773–778, 2005.

[94] Lin Liao, Dieter Fox, and Henry A. Kautz. Extracting places and activities from gps traces using hierarchical conditional random fields. *International Journal of Robotics Research (IJRR)*, 26(1):119–134, 2007.

[95] Lin Liao, Donald J. Patterson, Dieter Fox, and Henry A. Kautz. Learning and inferring transportation routines. *Artif. Intell.*, 171(5-6):311–331, 2007.

[96] Beth Logan, Jennifer Healey, Matthai Philipose, Emmanuel Munguia Tapia, and Stephen S. Intille. A long-term evaluation of sensing modalities for activity recognition. In *Ubicomp*, pages 483–500, 2007.

[97] Hong Lu, A. J. Bernheim Brush, Bodhi Priyantha, Amy K. Karlson, and Jie Liu. Speakersense: Energy efficient unobtrusive speaker identification on mobile phones. In Kent Lyons, Jeffrey Hightower, and Elaine M. Huang, editors, *Pervasive*, volume 6696 of *Lecture Notes in Computer Science*, pages 188–205. Springer, 2011.

[98] Paul Lukowicz, Jamie A. Ward, Holger Junker, Mathias Stäger, Gerhard Tröster, Amin Atrash, and Thad Starner. Recognizing workshop activity using body worn microphones and accelerometers. In *Pervasive*, pages 18–32, 2004.

[99] Maryam Mahdaviani and Tanzeem Choudhury. Fast and scalable training of semi-supervised crfs with application to activity recognition. In *NIPS*, 2007.

[100] Gideon Mann, Ryan McDonald, Mehryar Mohri, Nathan Silberman, and Dan Walker. Efficient large-scale distributed training of conditional maximum entropy models. In *NIPS*, pages 1231–1239, 2009.

[101] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, July 2008.

[102] Pedro J. Martn, Luis F. Ayuso, Roberto Torres, and Antonio Gavilanes. Algorithmic strategies for optimizing the parallel reduction primitive in cuda. In Waleed W. Smari and Vesna Zeljkovic, editors, *HPCS*, pages 511–519, 2012.

[103] Andrew McCallum. Efficiently inducing features of conditional random fields. In *UAI*, pages 403–410, 2003.

[104] Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. Maximum entropy markov models for information extraction and segmentation. In *ICML*, pages 591–598, 2000.

[105] David Minnen, Irfan A. Essa, and Thad Starner. Expectation grammars: Leveraging high-level expectations for activity recognition. In *CVPR (2)*, pages 626–632, 2003.

[106] David Minnen, Thad Starner, Irfan A. Essa, and Charles Lee Isbell Jr. Discovering characteristic actions from on-body sensor data. In *ISWC*, pages 11–18, 2006.

[107] Joseph Modayil, Tongxin Bai, and Henry A. Kautz. Improving the recognition of interleaved activities. In *UbiComp*, pages 40–43, 2008.

[108] Kevin Murphy. *Dynamic Bayesian Networks: Representation, Inference, and Learning*. PhD thesis, University of California, Berkeley, 2002.

[109] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI*, pages 467–475, 1999.

[110] Nam Thanh Nguyen, Dinh Q. Phung, Svetha Venkatesh, and Hung Hai Bui. Learning and detecting activities from movement trajectories using the hierarchical hidden markov models. In *CVPR (2)*, pages 955–960, 2005.

[111] Sinno Jialin Pan, Dou Shen, Qiang Yang, and James T. Kwok. Transferring localization models across space. In *AAAI*, pages 1383–1388, 2008.

[112] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. Technical report, Hong Kong University of Science and Technology, 2008.

[113] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.

[114] Shwetak N. Patel. *Infrastructure Mediated Sensing*. PhD thesis, Georgia Institute of Technology, 2008.

[115] Shwetak N. Patel, Matthew S. Reynolds, and Gregory D. Abowd. Detecting human movement by differential air pressure sensing in hvac system ductwork: An exploration in infrastructure mediated sensing. In *Pervasive*, pages 1–18, 2008.

[116] Shwetak N. Patel, Thomas Robertson, Julie A. Kientz, Matthew S. Reynolds, and Gregory D. Abowd. At the flick of a switch: Detecting and classifying unique electrical events on the residential power line (nominated for the best paper award). In *Ubicomp*, pages 271–288, 2007.

[117] Shwetak N. Patel, Khai N. Truong, and Gregory D. Abowd. Powerline positioning: A practical sub-room-level indoor location system for domestic use. In *Ubicomp*, pages 441–458, 2006.

[118] Donald J. Patterson, Dieter Fox, Henry A. Kautz, and Matthai Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *ISWC*, pages 44–51, 2005.

[119] Donald J. Patterson, Dieter Fox, Henry A. Kautz, and Matthai Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *ISWC*, pages 44–51, 2005.

[120] Donald J. Patterson, Lin Liao, Dieter Fox, and Henry A. Kautz. Inferring high-level behavior from low-level sensors. In *Proceedings of the Fifth International Conference on Ubiquitous Computing (UbiComp 2003)*, pages 73–89, 2003.

[121] Donald J. Patterson, Lin Liao, Krzysztof Gajos, Michael Collier, Nik Livic, Katherine Olson, Shiaokai Wang, Dieter Fox, and Henry A. Kautz. Opportunity knocks: A system to provide cognitive assistance with transportation

services. In *Proceedings of the Sixth International Conference on Ubiquitous Computing (UbiComp 2004)*, pages 433–450, 2004.

[122] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[123] Martha E. Pollack. Intelligent technology for an aging population: The use of ai to assist elders with cognitive impairment. *AI Magazine*, 26(2):9–24, 2005.

[124] Martha E. Pollack, Laura E. Brown, Dirk Colbry, Colleen E. McCarthy, Cheryl Orosz, Bart Peintner, Sailesh Ramakrishnan, and Ioannis Tsamardinos. Autominder: an intelligent cognitive orthotic system for people with memory impairment. *Robotics and Autonomous Systems (RAS)*, 44(3-4):273–282, 2003.

[125] Ariadna Quattoni, Michael Collins, and Trevor Darrell. Conditional random fields for object recognition. In *NIPS*, 2004.

[126] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[127] Taylor Raines, Milind Tambe, and Stacy Marsella. Automated assistants to aid humans in understanding team behaviors. In *Agents*, pages 419–426, 2000.

[128] Alvin Raj, Amarnag Subramanya, Dieter Fox, and Jeff Bilmes. Rao-blackwellized particle filters for recognizing activities and spatial context from wearable sensors. In *Experimental Robotics: The 10th International Symposium, Springer Tracts in Advanced Robotics*, 2006.

[129] Parisha Rashidi and Diane Cook. Activity knowledge transfer in smart environments. *Pervasive and Mobile Computing*, 2011.

[130] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L. Littman. Activity recognition from accelerometer data. In *AAAI*, pages 1541–1546, 2005.

[131] Patrick Riley and Manuela M. Veloso. Recognizing probabilistic opponent movement models. In *RoboCup*, pages 453–458, 2001.

[132] Adam Sadilek, Henry A. Kautz, and Jeffrey P. Bigham. Finding your friends and following them to where you are. In *WSDM*, pages 723–732, 2012.

[133] Adam Sadilek, Henry A. Kautz, and Vincent Silenzio. Predicting disease transmission from geo-tagged micro-blog data. In *AAAI*, 2012.

[134] Suchi Saria and Sridhar Mahadevan. Probabilistic plan recognition in multi-agent systems. In *ICAPS*, pages 287–296, 2004.

[135] Dou Shen, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Building bridges for web query classification. In *SIGIR*, pages 131–138, 2006.

[136] Thomas Stiefmeier, Georg Ogris, Holger Junker, Paul Lukowicz, and Gerhard Tröster. Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario. In *ISWC*, pages 97–104, 2006.

[137] Maja Stikic, Kristof Van Laerhoven, and Bernt Schiele. Exploring semi-supervised and active learning for activity recognition. In *ISWC*, 2008.

[138] Marc Strauss. *HandWave: Design and Manufacture of a Wearable Wireless Skin Conductance Sensor and Housing*. PhD thesis, Massachusetts Institute of Technology, 2005.

[139] Erich P. Stuntebeck, Shwetak N. Patel, Thomas Robertson, Matthew S. Reynolds, and Gregory D. Abowd. Wideband powerline positioning for indoor localization. In *UbiComp*, pages 94–103, 2008.

[140] Amar Subramanya, Alvin Raj, Jeff A. Bilmes, and Dieter Fox. Recognizing activities and spatial context using wearable sensors. In *UAI*, 2006.

[141] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *NIPS*. Curran Associates, Inc., 2007.

[142] Gita Sukthankar and Katia P. Sycara. Robust recognition of physical team behaviors using spatio-temporal models. In *AAMAS*, pages 638–645, 2006.

[143] Charles Sutton and Andrew McCallum. Collective segmentation and labeling of distant entities in information extraction. Technical Report TR 04-49, University of Massachusetts Amherst, 2004.

[144] Charles A. Sutton, Khashayar Rohanimanesh, and Andrew McCallum. Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data. In *ICML*, 2004.

[145] Milind Tambe. Tracking dynamic team activity. In *AAAI/IAAI, Vol. 1*, pages 80–87, 1996.

[146] Milind Tambe and Paul S. Rosenbloom. Resc: An approach for real-time, dynamic agent tracking. In *IJCAI*, pages 103–111, 1995.

[147] Emmanuel Munguia Tapia, Stephen S. Intille, and Kent Larson. Activity recognition in the home using simple and ubiquitous sensors. In *Pervasive*, pages 158–175, 2004.

[148] Emmanuel Munguia Tapia, Stephen S. Intille, and Kent Larson. Portable wireless sensors for object usage sensing in the home: Challenges and practicalities. In *AmI*, pages 19–37, 2007.

[149] Emmanuel Munguia Tapia, Stephen S. Intille, Louis Lopez, and Kent Larson. The design of a portable kit of wireless sensors for naturalistic data collection. In *Pervasive*, pages 117–134, 2006.

[150] Benjamin Taskar, Pieter Abbeel, and Daphne Koller. Discriminative probabilistic models for relational data. In *UAI*, pages 485–492, 2002.

[151] Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10:1633–1685, 2009.

[152] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical dirichlet processes. *Journal of The American Statistical Association*, 101(476):1566–1581, 2006.

[153] Tran The Truyen, Hung Hai Bui, and Svetha Venkatesh. Human activity learning and segmentation using partially hidden discriminative models. In *Workshop on Human Activity Recognition and Modelling HAREM 2005*, pages 87–95, 2005.

[154] Douglas L. Vail and Manuela M. Veloso. Feature selection for activity recognition in multi-robot domains. In *AAAI*, pages 1415–1420, 2008.

[155] Douglas L. Vail, Manuela M. Veloso, and John D. Lafferty. Conditional random fields for activity recognition. In *AAMAS*, pages 1331–1338, 2007.

[156] Douglas L. Vail, Manuela M. Veloso, and John D. Lafferty. Conditional random fields for activity recognition. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007)*, page 235, 2007.

[157] Tim van Kasteren, Gwenn Englebienne, and Ben J. A. Kröse. Transferring knowledge of activity recognition across sensor networks. In Patrik Floréen, Antonio Krüger, and Mirjana Spasojevic, editors, *Pervasive*, volume 6030 of *Lecture Notes in Computer Science*, pages 283–300. Springer, 2010.

[158] Tim van Kasteren, Athanasios K. Noulas, Gwenn Englebienne, and Ben J. A. Kröse. Accurate activity recognition in a home setting. In *UbiComp*, pages 1–9, 2008.

[159] T.L.M. van Kasteren, G. Englebienne, and B.J.A. Kröse. Recognizing activities in multiple contexts using transfer learning. In *AAAI Fall 2008 Symposium: AI in Eldercare*, Washington DC, USA, 2008.

[160] Shiaokai Wang, William Pentney, Ana-Maria Popescu, Tanzeem Choudhury, and Matthai Philipose. Common sense based joint training of human activity recognizers. In *IJCAI*, pages 2237–2242, 2007.

[161] Sy Bor Wang, Ariadna Quattoni, Louis-Philippe Morency, David Demirdjian, and Trevor Darrell. Hidden conditional random fields for gesture recognition. In *CVPR (2)*, pages 1521–1527, 2006.

[162] D. H. Wilson and Christopher G. Atkeson. Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors. In *Pervasive*, pages 62–79, 2005.

[163] Tsuyu Wu, Chiachun Lian, and Jane Yungjen Hsu. Joint recognition of multiple concurrent activities using factorial conditional random fields. In *AAAI Workshop on Plan, Activity, and Intent Recognition*, 2007.

[164] Danny Wyatt, Matthai Philipose, and Tanzeem Choudhury. Unsupervised activity recognition using automatically mined common sense. In *AAAI*, pages 21–27, 2005.

[165] Danny Wyatt, Matthai Philipose, and Tanzeem Choudhury. Unsupervised activity recognition using automatically mined common sense. In Manuela M. Veloso and Subbarao Kambhampati, editors, *AAAI*, pages 21–27. AAAI Press / The MIT Press, 2005.

[166] Qiang Yang, Yuqiang Chen, Gui-Rong Xue, Wenyuan Dai, and Yong Yu. Heterogeneous transfer learning for image clustering via the socialweb. In *ACL/AFNLP*, pages 1–9, 2009.

[167] Jie Yin, Qiang Yang, and Jeffrey Junfeng Pan. Sensor-based abnormal human-activity detection. *IEEE Trans. Knowl. Data Eng.*, 20(8):1082–1090, 2008.

[168] Dong Zhang, Daniel Gatica-Perez, Samy Bengio, and Iain McCowan. Semi-supervised adapted hmms for unusual event detection. In *CVPR (1)*, pages 611–618, 2005.

[169] Xian-Xing Zhang, Hua Liu, Yang Gao, and Derek Hao Hu. Detecting abnormal events via hierarchical dirichlet processes. In Thanaruk Theeramunkong, Boonserm Kijsirikul, Nick Cercone, and Tu Bao Ho, editors, *PAKDD*, volume 5476 of *Lecture Notes in Computer Science*, pages 278–289. Springer, 2009.

[170] Vincent Wenchen Zheng, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative filtering meets mobile recommendation: A user-centered approach. In Maria Fox and David Poole, editors, *AAAI*. AAAI Press, 2010.

[171] Vincent Wenchen Zheng, Derek Hao Hu, and Qiang Yang. Cross-domain activity recognition. In Sumi Helal, Hans Gellersen, and Sunny Consolvo, editors, *UbiComp*, ACM International Conference Proceeding Series, pages 61–70. ACM, 2009.

[172] Vincent Wenchen Zheng, Sinno Jialin Pan, Qiang Yang, and Jeffrey Junfeng Pan. Transferring multi-device localization models using latent multi-task learning. In *AAAI*, pages 1427–1432, 2008.

[173] Vincent Wenchen Zheng, Evan Wei Xiang, Qiang Yang, and Dou Shen. Transferring localization models over time. In *AAAI*, pages 1421–1426, 2008.

[174] Vincent Wenchen Zheng and Qiang Yang. User-dependent aspect model for collaborative activity recognition. In *IJCAI*, pages 2085–2090, 2011.

[175] Vincent Wenchen Zheng, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative location and activity recommendations with gps history data. In *WWW*, pages 1029–1038, 2010.

[176] Vincent Wenchen Zheng, Yu Zheng, Xing Xie, and Qiang Yang. Towards mobile intelligence: Learning from gps history data for collaborative recommendation. *Artif. Intell.*, 184-185:17–37, 2012.