In Proceedings of the Twenty-first International Joint Conference
on Artificial Intelligence (IJCAI-09), Pasadena CA, USA, July 2009. Pages 20-25.

Activity Recognition: Linking Low-level Sensors to High-level Intelligence

Qiang Yang
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Kowloon, Hong Kong, China
qyang@cse.ust.hk

Abstract

Sensors provide computer systems with a window to the outside world. Activity recognition "sees" what is in the window to predict the locations, trajectories, actions, goals and plans of humans and objects. Building an activity recognition system requires a full range of interaction from statistical inference on lower level sensor data to symbolic AI at higher levels, where prediction results and acquired knowledge are passed up each level to form a knowledge food chain. In this article, I will give an overview of some of the current activity recognition research works and explore a life-cycle of learning and inference that allows the lowest-level radio-frequency signals to be transformed into symbolic logical representations for AI planning, which in turn controls the robots or guides human users through a sensor network, thus completing a full life cycle of knowledge.

1 Introduction

One of the most important technological innovations in today's world is the arrival of cheap and easy-to-use electronic sensors. With the growing maturity of sensor and sensor-network technologies, advanced applications are gaining speed in areas such as pervasive computing, medical assistive technologies, security and environmental monitoring, gaming, sensor-based farming and coal-mine-safety technologies and many others. Like many previous technological innovations, sensor technology also helps usher a new era for artificial intelligence research, with far-reaching implications. With the help of accurate activity recognition, researchers are now capable of providing various personalized support for many real-world applications. For example, [Pollack *et al.*, 2003] used activity recognition to help the elders recognize and deal with cognitive decline associated with sickness and aging by sending personalized activity reminders. [Liao *et al.*, 2004; Patterson *et al.*, 2004; Zheng *et al.*, 2008c] employed activity recognition to predict transportation modes. [Yin *et al.*, 2007] showed how to detect abnormal human activity for security monitoring.

With the arrival of sensor technology, we now have an unprecedented opportunity to advance the science of artificial intelligence (AI), by linking sensors at the low levels of abstraction with high-level knowledge representation, reasoning, learning and inference. In a way, achieving this linkage is an ultimate goal of AI, which is to provide a closed-loop feedback-control system to embody all aspects of intelligence. Through sensor-based activity recognition systems, we can envision a healthy life-cycle in which a positive feedback loop is in place to allow experience to be continuously acquired and fed into a high-level machinery for creating knowledge bases for AI. A major impetus of this computational machinery will be the ever-widening range of applications, which help provide the much needed raw data for the science of AI to go forward.

In this article, I will summarize recent works on sensor-based activity recognition, including those of my research group at Hong Kong University of Science and Technology. I will argue for activity recognition to be a bridge that links low-level sensors and high-level intelligence.

2 Overview

What is activity recognition? In a *Forbes* article, [Huang, 2003] gave this description:

> Eric Dishman is making a cup of tea and his kitchen knows it. At Intel's Proactive Health Research lab in Hillsboro, OR, tiny sensors monitor the researcher's every move. Radio frequency identification tags and magnetic sensors discreetly affixed to mugs, a tea jar, and a kettle, plus switches that tell when cabinet doors are open or closed, track each tea-making step. A nearby computer makes sense of these signals; if Dishman pauses for too long, video clips on a television prompt him with what to do next.

This vivid description highlights several key aspects of activity recognition, which aims to interpret the sensor readings of humans or moving things (such as a truck) and tell us in high-level, human understandable terms what is going on. First, activity recognition requires sensors, which can generate and receive signals to be read by a computer program. Second, there should be a software program that can interpret the sensor readings. Most of these sensor readings are also uncertain data. Finally, activity recognition involves looking at the past, so that the computer program should be able to learn from experience. At the same time, it helps project into the future through inferences.

Today, we have a variety of sensors and sensory devices that are available to us at a low cost. Some of them are listed here:

RFID Radio-frequency identification (RFID) uses an integrated circuit for storing and sending radio-frequency (RF) signal. A RFID reader device can both send and receive signals. Passive RFID tags can read generate signals when triggered by RF waves sent by the readers, whereas battery-powered active tags can generate and receive signals by themselves. The range of RFID tags is within several meters. RFID has been widely used in product tracking and identification, especially in logistics operations. In Hong Kong, RFID technology is daily used by millions of people in a subway and debit card known as the Octopus Card.

GPS The Global Positioning System (GPS) is a global system based on between 24 and 32 satellites which send RF signals. Receivers can determine their current locations based on the time-of-flight information carried by the RF signals. In an outdoor environment, based on the location sequences, high-level inference can be done to ascertain an agent's transportation modes, goals and intentions, as done in [Liao *et al.*, 2004; Patterson *et al.*, 2004; Zheng *et al.*, 2008c].

WiFi Most notebook computers, PDAs and some mobile phones today are equipped with WiFi device that can communicate through an IEEE 802.11b/g wireless network in the 2.4GHz frequency bandwidth. In an indoor or even outdoor area, one or more access points (APs) can send and receive RF signals from other APs or notebooks. WiFi devices are especially useful for locating a user and tracking his/her movement in an indoor environment where GPS is often no longer available.

Mobile Phones The mobile phones nowadays have advanced sensors for measuring some predefined activity of the mobile user, such as turning directions. They are known as the Inertial Navigation System (INS), which are motion sensors such as gyroscope, accelerometer and compass.

A common feature of these sensors is that they are very widely available in our everyday lives, as opposed to some specialized and expensive sensors.

Taking the sensor-reading data, as well as some labels attached to them through calibration, from the above sensor devices as input, our goal is to build a model in the form of a computer program. The input to the model is a sequence of sensor reading vectors and the output is a set of meaningful activity terms that reflect the observed actions of humans or moving objects, or the final intentions (i.e., goals). For example, a user holding a mobile phone that is equipped with WLAN cards for reading WiFi signals can perform various actions, such as Walk-in-HW2, Enter-Office and Make Photocopies, in an office WiFi environment. The mobile device periodically (e.g. per second) records signal-strength measurements sent by various access points (APs). For example, an observation may be $o =< 48; 83; 57 >$, consisting of radio signal strength (RSS) values from three available APs. A user's behavior can be understood as a sequence of actions taken to achieve high-level goals such as Seminar-in-Room2 and Print-in-Room3. A user's signal trace is often represented as a sequence of radio-signal strength (RSS) measurements such as $< o_1; o_2; \ldots ; o_t >$, where each o_i is a signal vector at some time.

Early systems for activity recognition treat the inputs as a sequence of high-level symbolic observations. They typically output symbolic goal descriptions at a higher level of abstraction [Kautz and Allen, 1986; Lesh and Etzioni, 1995]. The fact that the input actions are mostly defined at the symbolic level is partly due to the unavailability of the low-level sensor data several decades ago. These systems are mostly deterministic in nature. Their input consists of a plan library that describes the logical models of actions and their relationship in the form of an action taxonomy and associated logical axioms. Given a sequence of symbolic action descriptions as input, the task of plan recognition can be accomplished by searching in a space of possible goal hypotheses for candidate plans and goals that are consistent with the observed action sequences. Many of these methods relied on set covering[Kautz and Allen, 1986], inductive logic programming and natural language parsing.

3 From Sensors to Locations

Once we set up a sensor network and can collect signal values, our first task is to infer a user's location. Given the current RSS values $o =< 25, 98, \ldots , 40 >$, where is the user located?

One approach is multilateration, which consists two main steps. It first transforms the sensor readings into a distance measure. It then recovers the coordinates in terms of relative distance to the *beacon* nodes. This approach relies on an ideal signal propagation model and extensive hardware support. However, it suffers from low accuracy because RSS signals do not follow ideal propagation patterns. Specialized methods such as [Bahl *et al.*, 2000] have been developed to accurately track the *mobile* nodes using ultrasonic signals. However, these methods require special hardware devices such as ultrasonic transceivers.

A complementary localization method is through machine learning. Many localization systems operate in two phases: an *offline* or *training phase* and an *online localization phase*.

If we model the location-estimation problem as a classification problem in machine learning, the area of interest can be modeled as a finite location space $\mathbb{L} = \{l_1, \ldots , l_n\}$. If we consider the location coordinates as continuous values, then we can use a regression model. An advantage of the machine-learning methods is that the locations of APs are not necessarily known.

In an offline phase, signal-strength measurements are collected at each location l_i. After the data are calibrated, a histogram of observations is built for each AP or base-station b_k at each location l_i. This is done by constructing a conditional probability $P(s_k|b_k, l_i)$, which is the probability that b_k has the signal-strength value s_k at the location l_i. By making an independence assumption among signals from different transmitters, we multiply all these probabilities to obtain the conditional probability of receiving a particular observation o_j at the location l_i as $P(o_j|l_i) = \prod_{k=1}^{p} P(s_k|b_k, l_i)$, which looks up the content stored in the radio map $P(s_k|b_k, l_i)$. In the online phase, when a real-time signal-strength vector o' is observed, a posterior distribution over all the locations is computed using Bayes' rule:

$$P(l_i|o') = \frac{P(o'|l_i)P(l_i)}{\sum_{i=1}^{n} P(o'|l_i)P(l_i)}, \quad (1)$$

where $P(l_i)$ encodes prior knowledge about where a user might be. Based on this equation, the estimated location is the one with a maximum posterior probability $l^* = \arg\max_{l_i} P(l_i|o')$. The advantage of the above machine-learning based method is that it captures the noise in signal propagation through conditional probabilities. Therefore, it can preserve information carried by the signals for localization. However, there are several limitations as we pointed out above. One issue is that the above method assumes that the location labels are available, which is often not the case. To get the labels often requires expensive human effort. Several recent approaches have been proposed for reducing the calibration effort of learning localization models offline. [Ferris *et al.*, 2007] solved a WiFi-SLAM (simultaneous localization and mapping) by applying Gaussian-Process-Latent-Variable models (GP-LVMs) to construct RSS map under an unsupervised learning framework. In this model, an appropriate motion dynamics model needs to be given. [Pan *et al.*, 2006] applied a semi-supervised learning framework in WiFi-based location estimation.

Another major assumption is that the signal space does not change, which is often wrong due to the dynamic characteristics of signal propagation and the environment. Signal distribution can be vastly different when we move across the floors of a building, and when we switch between different sensor devices when one device is used to collect the training data and another device is used for location estimation. On this issue, previous solutions have been proposed. The LEASE system [Krishnan *et al.*, 2004] utilizes different hardware systems to solve this problem. LEASE employs a number of stationary emitters and sniffers to obtain up-to-date RSS values for updating the maps. The localization accuracy can only be guaranteed when these additional hardware systems are deployed. Yin et al. [Yin *et al.*, 2005] placed several reference points in an office environment to help provide up to date signal and location information, which provides the current labeled data to help calibrate a past model.

Recent research works have considered the dynamic-data problem as a *transfer learning problem*. Transfer learning is a machine learning framework that adapts learned models in target domains by making use of the knowledge and data in source domains [Caruana, 1997]. When the user trace information is available online, the parameters of a hidden Markov model can be transferred by adapting the parameters of model from one time period to another [Zheng *et al.*, 2008b]. A manifold co-regularization based solution is proposed when the trace information is not available online [Pan *et al.*, 2007].

Similarly, the problem of adapting models across space can be considered as a transfer learning problem for spatial transfer. [Pan *et al.*, 2008] presented a solution by exploiting the data collected in one area and propagate them to the rest of the environment. Domain knowledge of an indoor environment is first extracted from the labeled data collected in one area. Then, the extracted domain knowledge is adapted in a model to propagate the label information to unlabeled data collected in the rest of the environment. The learning problem was formulated as a quadratically constrained quadratic program optimization problem to discover an underlying semantic manifold of the WiFi signal data. This semantic manifold acts as a bridge that propagates the common knowledge across different areas.

[Zheng *et al.*, 2008a] considered transfer learning across sensor devices for a two-dimensional WiFi-based indoor-localization problem. In this approach, a multi-device localization problem can be formulated as a multi-task learning problem by exploiting an often-satisfied assumption that the models learned in a *latent* feature space from the multiple devices are often similar. In this latent space, a new device can benefit from learning from the data collected by other devices to train a localization model.

A collection of location estimation benchmark data is available at http://www.cse.ust.hk/∼qyang/ICDMDMC07, and an IEEE ICDM competition based on the data is described in [Yang *et al.*, 2008].

4 From Locations to Activities

In the next level up, we will infer activities and goals from location sequences. Here the concept of a location is understood in a general sense, where it can either be a physical 3-D location, or it can be a virtual location in a multi-dimensional space spanned by all available sensors (such as RFID sensors attached to a pen or a door knob).

I now highlight a recent location-based activity-recognition model (LAR) [Yin *et al.*, 2004]. This model transforms sequences of sensor readings and inferred locations from the last step (see last section) to user activities and goals. The LAR model relies on a sensor model for location estimation at the lowest level (see Figure 1), which shows two time slices that are numbered t and $t-1$, respectively. In the figure, the shaded nodes SS represent the RSS variables of signals received from the sensor beacons (e.g., APs in a Wireless LAN), which can be directly observed. All other variables are hidden, including the physical location L of the user, the action A and the goal G.

Based on the sensor model, the LAR model can learn a dynamic Bayesian network (DBN) model from a collection of training traces $\mathcal{D}$. The model parameters are estimated using an expectation maximization (EM) algorithm. After learning the DBN model, we can infer the most probable action sequence $A_1, A_2, \ldots, A_t$ from the sensor readings and inferred location sequences. We can then infer goals from the actions. Given an inferred sequence of actions obtained so far $A_1, A_2, \ldots, A_t$, we can find the most likely goal set G^* as follows:

$$G^* = \arg\max_{G_k} P(G_k|A_1, A_2, \ldots, A_t) \quad (2)$$

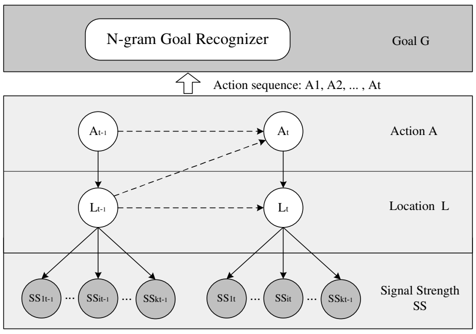

Figure 1: Location-based Activity Recognition Model (LAR) from [Yin et al. 2004]

By applying the Bayesian Rule, we then have

$$G^* = \arg\max_{G_k} \frac{P(A_{1:t}|G_k)P(G_k)}{P(A_{1:t})} \tag{3}$$

Actions and goals do not have to be inferred from locations, instead they can be predicted from trends of sensor signals in a period of time. In [Yin *et al.*, 2008], an alternative probabilistic approach for activity recognition, which is referred to as *Segmentation-based Activity Recognition*(SAR), was proposed. Intuitively, a large class of a person's activities exists where each activity is more or less *rough-grained*, in that the precise location information is not needed, or impossible to obtain. Instead, a rough idea of the general *trends* of a user's movements is sufficient for activity recognition. These trends correspond to some segments of sensor readings along the time dimension, which can be obtained through image-segmentation algorithms.An advantage of this view is that we can treat the patterns of an activity as a whole rather than each individual values. For example, on seeing that a certain sensor's signal reading is gradually increasing while another, far away sensor's reading is gradually decreasing, we can recognize that an agent is moving towards the first sensor, without having to know the precise location of the agent at each moment.

When GPS data are available, location sequences are relatively easier to obtain, and a major research focus is to infer user activities in terms of their transportation modes. For example, [Liao *et al.*, 2004; Patterson *et al.*, 2004; Zheng *et al.*, 2008c] employed probabilistic techniques to infer whether a user is taking a bus or walking based on the GPS readings.

One of the key features of real-world human activities is that multiple goals are often achieved together and in a sophisticated way. [Hu *et al.*, 2008] analyzed the MIT PlaceLab House_n PLIA1("PlaceLab Intensive Activity Test Dataset 1") [Intille *et al.*, 2006] in detail for illustrating the taxonomic nature of multiple goals. This dataset was recorded on Friday March 4, 2005 from 9 AM to 12 noon with a volunteer familiar with Placelab. [Hu *et al.*, 2008] manually constructed a goal hierarchy from this dataset. The lowest level, where the activities are extracted from the original data, includes activities such as "sweeping", "washing-ingredients", etc. Relevant activities are combined into more general activities that form the medium level, with activities such as "preparing ingredients", "Dealing-with-clothes", etc. These activities are grouped into 9 categories, comprising the highest level to include: cleaning indoor, yard-work, laundry, dishwashing, meal-preparation, hygiene, grooming, personal and information/leisure. The higher-level goals are more coarse grained, whereas the lower level ones are detailed. Through this taxonomy and the collected real-world activity sequences, it is observed that interleaving goals, where one goal may pause for a period of time while the human agent pursues another, often occurs. Furthermore, the likelihood increases as we move up the taxonomy. Similarly, concurrent goals, which are goals being pursued together, are more often observed as one moves down the goal taxonomy.

Multiple, concurrent and interleaving activities and goals (which corresponds to a whole sequence of activities) are difficult to recognize due to their inherent complexity. By exploiting a Conditional Random Fields (CRF) model for these activities, [Hu and Yang, 2008] applied CRF in a two-level probabilistic framework that deals with both concurrent and interleaving goals from observed sensor-reading sequences. CRF has been previously used by several other researchers as well [Vail *et al.*, 2007; Liao *et al.*, 2007]. To further consider the correlation between goals, a correlation graph is designed to represent the correlation between different 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. Experimental results using several real wireless sensor network data sets demonstrate that the recognition algorithm, known as CIGAR, is both efficient and accurate.

5 From Activities to Action Models

Above I have described some recent works on how to generate sequences of actions from observed sensor readings. In this section, I will describe how to generate logical, generative models of actions that allow autonomous planning to function, once sequences of user activities and some domain conditions are known. I will only give an overview in this section, and leave some of the details in [Yang *et al.*, 2007].

Automatic planning systems today take as input the formal definitions of actions, an initial state and a goal state description in logical forms, and produce symbolic plans, which are sequences of activity terms, for execution. To achieve goals, automatic planning systems produce sequences of actions from the given action models that are provided as input [Ghallab *et al.*, 2004]. A typical way to describe action models is to use action languages such as the **P**lanning **D**omain **D**efinition **L**anguage (PDDL) [Fox and Long, 2003; Ghallab *et al.*, 2004]. In the past, the task of building action models has been accomplished manually, which can be time consuming and error-prone. In a way, the lack of real-world data has greatly hampered the progress of AI planning in its practical applications, as action models have become a bottleneck in this important field. Thus, it is desirable to be able to *automatically learn* action models from sensory observations, so that these actions can be taken as inputs to planning systems. This is a crucial step in the whole knowledge lifecycle that I describe in this article.

In this section, I describe an algorithm known as ARMS (*Action-Relation Modelling System* [Yang *et al.*, 2007]) for automatically acquiring action models. The input to the ARMS system is a collection of observed activity traces that are recognized through an activity recognition system such as LAR or SAR described earlier. It first applies a frequent itemset-mining algorithm to these traces to find a collection of frequent action-sets. It then encodes these sets sets as constraints on the candidate action models. These constraints then become the input to another modeling system for solving the weighted MAX-SAT [Kautz and Selman, 1996], whose solution corresponds to the learned action models in terms of their pre and post conditions. The output of ARMS is a set of relational actions that can be further edited by human editors to generate plans, thus reducing the burden of humans in creating the planning domains. Because the preliminary forms of the actions have been encoded in logical forms, they can also be accepted directly by autonomous planning systems to produce plans.

Consider an example input and output of our algorithm in a typical problem domain from an AI planning competition [Fox and Long, 2003]. The actions to be learned are listed in the form of activity names along with their likely parameters which are objects that are often associated with the actions. For example, an action in a logistics domain might be: drive(?x:truck ?y:place ?z:place) where truck and place are the types of objects given as input. ?x and ?y are variable parameters. Relations in the domain should also be given in current version of our system, such as (at ?x:locatable ?y:place), but they can also be learned from sensor readings. As part of the input, we need activity traces, which are sequences of activities. As an example, an activity sequence in the depot domain is: $< I_1$; lift(h1 c0 p1 ds0);load(h1 c0 t0 ds0); $\ldots$; drop (h0 c0 p0 dp0); goal=((on c0 p0)$>$, where $I_1 = \{$(at p0 dp0), (clear p0), $\ldots$, (clear c0), (on c0 p1), (available h1), (at h1 ds0) $\}$ is an initial state description. The initial and goal descriptions can be obtained by converting sensor readings to propositional literals, or provided by human editors.

From these input, we wish to learn the preconditions, add and delete lists of all actions, in a STRIPS action representation, or more sophisticated forms. ARMS learns an action model for every action in a problem domain in order to "explain" all training examples successfully. An example output from our learning algorithms for the load(?x ?y ?z ?p) action signature is:

action	load(?x:hoist ?y:crate ?z:truck ?p:place)
pre:	(at ?x ?p), (at ?z ?p), (*lifting* ?x ?y)
del:	(lifting ?x ?y)
add:	(at ?y ?p), (in ?y ?z), (available ?x), (clear ?y)

ARMS was shown in [Yang *et al.*, 2007] to perform similar inference and learning as a Markov Logic Network [Richardson and Domingos, 2006]. It works even when partially observed states are available. Further extensions have been made to allow ARMS to generate more expressive action models that include conditional effects and first-order logic formulas in preconditions and postconditions of action models. Another extension was recently made to allow hierarchical task network (HTN) task models to be learned from action sequences and partial state observations [Zhuo *et al.*, 2009].

6 Closing the Loop

We have generated some action models for the logistics domains using ARMS based on action sequences generated by an activity recognition module. These generated action models are given to a planning system to generate new plans. We are currently testing a robotic system that can take these plans and execute them. Preliminary tests have shown that, because the action models generated by ARMS are imperfect, some plans cannot be executed successfully. In such cases, we can generate feedback to ARMS or to activity recognition modules for them to learn the action models better. Some plans are indeed successful, in which case they can be passed on to a robotic system for execution. Alternatively, we can send the generated plans as guides for human users. For example, researchers have used planning modules to generate reminders for people in their daily lives [Pollack *et al.*, 2003; Patterson *et al.*, 2004]. Agents equipped with these reminders and plans can further help generate more data in a sensor network, creating an opportunity for activity recognition engines to adapt and evolve.

Several decades have passed since the first inception of artificial intelligence (AI). As a science and an engineering endeavor, AI has achieved much over the years, but it is also largely fragmented. A lesson that we can learn from other fields, such as physics, is to develop an empirical subfield of AI, and to integrate this subfield with the more theoretical fronts of the discipline. To do this, we have to learn to get our hands dirty. In building activity recognition systems into a knowledge food chain, we will hopefully close the loop in a positive feedback-loop for AI.

Acknowledgement

This work is supported by funding from Hong Kong CERG Project 621307 and NEC China Lab. I sincerely thank my students for collaboration over the years: Jie Yin, Dou Shen, Junfeng Pan, Sinno Jialin Pan, Xiaoyong Chai, Rong Pan, Vincent Wenchen Zheng, Derek Hao Hu, Kangheng Wu and Hankz Hankui Zhuo.

References

[Bahl *et al.*, 2000] P. Bahl, A. Balachandran, and V. Padmanabhan. Enhancements to the RADAR user location and tracking system. Technical report, Microsoft Research, February 2000.

[Caruana, 1997] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[Ferris *et al.*, 2007] Brian Ferris, Dieter Fox, and Neil Lawrence. WiFi-SLAM Using Gaussian Process Latent Variable Models. In *IJCAI 2007*, January 2007.

[Fox and Long, 2003] Maria Fox and Derek Long. Pddl2.1: An extension to pddl for expressing temporal planning domains. *J. Artif. Intell. Res. (JAIR)*, 20:61–124, 2003.

[Ghallab *et al.*, 2004] Mark Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann Publishers, 2004.

[Hu and Yang, 2008] Derek Hao Hu and Qiang Yang. CIGAR: Concurrent and interleaving goal and activity recognition. In *AAAI*, pages 1363–1368, 2008.

[Hu *et al.*, 2008] Derek Hao Hu, Sinno Jialin Pan, Vincent Wenchen Zheng, Nathan Nan Liu, and Qiang Yang. Real world activity recognition with multiple goals. In *UbiComp '08*, pages 30–39, 2008. ACM.

[Huang, 2003] Gregory T. Huang. Monitoring Mom – As population matures, so do assisted-living technologies. *Forbes Magazine, Technology Review*, July 2003.

[Intille *et al.*, 2006] 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 2006)*, pages 349–365, 2006.

[Kautz and Allen, 1986] Henry Kautz and James Allen. Generalized plan recognition. In *AAAI-1986*, pages 32–37, 1986.

[Kautz and Selman, 1996] Henry Kautz and Bart Selman. Pushing the envelope: Planning, propositional logic, and stochastic search. In *AAAI 1996*, pages 1194–1201, Portland, Oregon USA, 1996.

[Krishnan *et al.*, 2004] P. Krishnan, A. S. Krishnakumar, Wen-Hua Ju, Colin Mallows, and Sachin Ganu. A System for LEASE: Location Estimation Assisted by Stationary Emitters for Indoor RF Wireless Networks. In *INFOCOM 2004*, 2004.

[Lesh and Etzioni, 1995] Neal Lesh and Oren Etzioni. A sound and fast goal recognizer. In *IJCAI 1995*, pages 1704–1710, Montreal, Canada, 1995.

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

[Liao *et al.*, 2007] 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.

[Pan *et al.*, 2006] Jeffrey J. Pan, Qiang Yang, Hong Chang, and Dit-Yan Yeung. A Manifold Regularization Approach to Calibration Reduction for Sensor-Network Based Tracking. In *AAAI 2006*, July 2006.

[Pan *et al.*, 2007] Sinno Jialin Pan, James T. Kwok, Qiang Yang, and Jeffrey Junfeng Pan. Adaptive localization in a dynamic WiFi environment through multi-view learning. In *AAAI 2007*, pages 1108–1113, Canada, July 2007.

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

[Patterson *et al.*, 2004] 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 *UbiComp 2004*, pages 433–450, 2004.

[Pollack *et al.*, 2003] 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.

[Richardson and Domingos, 2006] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

[Vail *et al.*, 2007] Douglas L. Vail, Manuela M. Veloso, and John D. Lafferty. Conditional random fields for activity recognition. In *AAMAS 2007*, page 235, 2007.

[Yang *et al.*, 2007] Qiang Yang, Kangheng Wu, and Yunfei Jiang. Learning action models from plan examples using weighted max-sat. *Artif. Intell.*, 171(2-3):107–143, 2007.

[Yang *et al.*, 2008] Qiang Yang, Sinno Jialin Pan, and Vincent Wenchen Zheng. Estimating location using Wi-Fi. *IEEE Intelligent Systems*, 23(1):8–13, 2008.

[Yin *et al.*, 2004] Jie Yin, Xiaoyong Chai, and Qiang Yang. High-level goal recognition in a wireless LAN. In *AAAI 2004*, pages 578–584, San Jose, CA, USA, July 2004.

[Yin *et al.*, 2005] J. Yin, Q. Yang, and L. Ni. Adaptive temporal radio maps for indoor location estimation. In *PerCom 2005*, Kauai Island, Hawaii, USA, March 2005.

[Yin *et al.*, 2007] Jie Yin, Qiang Yang, and Jeffrey Junfeng Pan. Sensor-based abnormal human-activity detection. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 20(8):17–31, 2007.

[Yin *et al.*, 2008] Jie Yin, Qiang Yang, Dou Shen, and Ze-Nian Li. Activity recognition via user-trace segmentation. *ACM Trans. Sen. Netw.*, 4(4):1–34, 2008.

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

[Zheng *et al.*, 2008b] Vincent Wenchen Zheng, Qiang Yang, Wei Xiang, and Dou Shen. Transferring localization models over time. In *AAAI 2008*, pages 1421–1426, Chicago, Illinois, USA, July 2008.

[Zheng *et al.*, 2008c] Yu Zheng, Like Liu, Longhao Wang, and Xing Xie. Learning transportation mode from raw gps data for geographic applications on the web. In *WWW '08*, pages 247–256, New York, NY, USA, 2008. ACM.

[Zhuo *et al.*, 2009] Hankz Hankui Zhuo, Derek Hao Hu, Chad Hogg, Qiang Yang, and Hector Munoz-Avila. Learning htn method preconditions and action models from partial observations. In *IJCAI 2009*.