Activity Recognition through Goal-Based Segmentation

Jie Yin and **Dou Shen** and **Qiang Yang**
Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong, China
{yinjie, dshen, qyang}@cs.ust.hk

Ze-Nian Li
School of Computing Science
Simon Fraser University
Burnaby, BC Canada V5A 1S6
li@cs.sfu.ca

Abstract

A major issue in activity recognition in a sensor network is how to automatically segment the low-level signal sequences in order to optimize the probabilistic recognition models for goals and activities. Past efforts have relied on segmenting the signal sequences by hand, which is both time-consuming and error-prone. In our view, segments should correspond to atomic human activities that enable a goal-recognizer to operate optimally; the two are intimately related. In this paper, we present a novel method for building probabilistic activity models at the same time as we segment signal sequences into motion patterns. We model each motion pattern as a linear dynamic model and the transitions between motion patterns as a Markov process conditioned on goals. Our EM learning algorithm simultaneously learns the motion-pattern boundaries and probabilistic models for goals and activities, which in turn can be used to accurately recognize activities in an online phase. A major advantage of our algorithm is that it can reduce the human effort in segmenting and labeling signal sequences. We demonstrate the effectiveness of our algorithm using the data collected in a real wireless environment.

Introduction

With recent advances in pervasive computing technology, it is now possible to track a moving object's context information as streams of signal data. From these data, a moving object's activities can be recognized using various probabilistic techniques. Being able to accomplish activity recognition is critical to many advanced applications. An important example is to help people with cognitive disorders live safely in the community. In recent years, probabilistic models, such as hidden Markov models (H. H. Bui & West 2002) and dynamic Bayesian networks (Liao, Fox, & Kautz 2004; Yin, Chai, & Yang 2004), have been proposed for activity recognition. An overriding theme of these research works has been to infer high-level goals from streams of low-level signals gathered in an uncertain environment where a network of beacons (e.g. satellites and WLAN access points) and sensors are available.

All these approaches employ a hierarchical probabilistic framework, by which the gap between high-level goals and low-level signals is bridged through the inference of locations or actions. At the lowest level, a sensor model is needed to transform low-level signals to high-level actions with a action-level sensor model, or to locations with a location-level sensor model. These models must be trained in order to enable subsequent higher-level inferences to be made for plan and goals. However, calibrating signals with location or action labels is a difficult problem. First, building the training data itself is labor intensive, as a large number of samples must be calibrated at each location or for each action. Second, the labeling of subsequences of signals may be infeasible to do by hand, because it is impossible to find the precise signal-segment boundaries that delineate different actions. Third, even if location or action labels can be obtained and the training signal sequences can be labeled, such data may need repeated updates due to the dynamically changing environment. What would be ideal is to allow an activity recognition system to optimally recognize goals without tediously learning a sensor model at the location or action levels. This would permit a mobile agent to automatically collect the signal traces that correspond to one or more goals, and allow an activity-recognition system to figure out signal segments and their correspondence with the high-level goals.

Our observation is that high-level goals can be directly inferred from low-level signal segments, where these segments should in turn be determined by goals themselves. This seemingly circular argument can be operationalized through an EM algorithm, which is what we do in this paper. The result is a goal-level recognition model that by-passes the location and action levels and directly bridges between signals and goals. This resulting model can be automatically trained with less human effort. Consider the following input to an activity recognition system. The training data consist of a set of user traces along with goal labels associated with each trace; this can be relatively cheaply done as we only require the user to provide goal labels for an entire trace. A by-product of such a goal-level recognition model is that each new trace can be automatically partitioned where each segment naturally defines a user behavior-pattern. Moreover, goals can be recognized from sequences of discovered motion patterns.

More specifically, taking the training data as input, we apply an EM algorithm to obtain a probabilistic segmentation model and a goal-recognizer simultaneously. The input of our algorithm consists of a collection of user traces that record sensor readings. Each trace is a multivariate time series associated with a goal label. The EM algorithm aims to find a set of motion patterns from the training data along with their transition probabilities, so that goals can be inferred from sequences of motion patterns for a newly arrived user trace. Therefore, the output of the algorithm is a segmentation model for new traces along with an activity-recognition model for goals. With the two models learned from the training data, a new trace can then be partitioned as it is received by a wireless device using the segmentation model. The corresponding goals for the new trace can be recognized using the activity-recognition model.

The rest of the paper is organized as follows. We begin by formulating our activity recognition problem. We discuss related work and present our algorithm in the following two sections. Then we present the results of experiments conducted in a real wireless environment. Finally, we conclude the paper and discuss directions for future work.

Problem Formulation

Let Y be an observed signal sequence on a user's activities in the environment. The sequence Y is a multivariate time series which consists of T samples $Y_1, Y_2, \cdots, Y_T$. Each sample $Y_i,\ 1 \leq i \leq T$, is a signal vector received at a time instant t_i, where m is the dimension of the vector space. Each sequence Y is associated with one goal label aimed at by the user. Consider a collection D of labeled sequences Y, our objective is to obtain two models, $\mathcal{M}_\mathcal{S}$ and $\mathcal{M}_\mathcal{A}$, where $\mathcal{M}_\mathcal{S}$ is a sequence segmentation model and $\mathcal{M}_\mathcal{A}$ is an activity recognition model. The output models $\mathcal{M}_\mathcal{S}$ and $\mathcal{M}_\mathcal{A}$ can be precisely described. Given a collection of training sequences D that consist of signal sequences D_i along with their associated goal G_i, we wish to find the two models such that the predicted goal $\arg\max_{G_k} P(G_k|D_i, \mathcal{M}_\mathcal{S}, \mathcal{M}_\mathcal{A})$ coincides with the ground truth goal G_i. This definition can be extended to multiple goals.

In the online phase, given a new signal sequence Y with unknown goals, we would like to compute a predicted goal G^* for the sequence Y, such that

$$\begin{aligned} L^* &= \arg\max_{L} P(L|Y, \mathcal{M}_\mathcal{S}), \\ G^* &= \arg\max_{G_k} P(G_k|L^*, \mathcal{M}_\mathcal{A}), \end{aligned}$$

where L^* is a sequence of segments used to partition the sequence Y, and G^* is a predicted goal for Y. The segmentation and goal recognition are performed simultaneously.

Related Work

Activity recognition has been a major focus in the area of artificial intelligence. However, many traditional approaches assume that high-level action sequences are provided in the input and do not concern themselves with the low-level sensory data (Kautz & Allen 1986; Goldman, Geib, & Miller 1999). In recent years, there has been an increasing interest in inferring a user's activities through integrating both high-level behavior and low-level sensor modeling. The work of (Liao, Fox, & Kautz 2004) applied a dynamic Bayesian network to estimate a person's locations and transportation modes from logs of GPS data with relatively precise location information. The work of (H. H. Bui & West 2002) introduced an abstract hidden Markov model to infer a person's goal from camera data in an indoor environment, but it is not clear from the article how action sequences are obtained from camera data. In our previous work (Yin, Chai, & Yang 2004), we proposed a two-level architecture of Bayesian models to infer a user's goals using signal-strength measurements in a complex wireless LAN environment. This work explicitly relied on training a location-based sensor model to infer locations from signals; the locations are part of the input that can serve as labels in the training data.

Several works in behavior recognition are related to our effort. The work of (Czielniak, Bennewitz, & Burgard 2003) learned motion patterns from collections of trajectories using the technique of clustering, but the trajectory-segments need to be constructed by hand. The work of (Li, Wang, & Shum 2002) applied a linear dynamic model to learn motion *texton*s from a human-dance sequence, which can then be used to generate new animation sequences. However, this work followed an unsupervised framework, which is not aimed at recognizing high-level goals. The work of (Peursum *et al.* 2004) employed a hidden Markov model to segment individual actions from a stream of human motion, but it requires human-supplied action labels as part of the input during the learning process. The work of (Bregler 1997) in computer vision explored how to extract significant features from video frames to enable high-level activity recognition, but this work relied on the visual features of camera images that are not available in a general sensor network.

On the surface, segments can be obtained by applying time-series analysis algorithms for our problem. However, on a closer examination, this is not the case. In the data mining area, many previous works (Zaki 2001; Oates 2002) focused on finding frequent patterns based on the idea of finding frequently recurring segments in a time series. In our problem, however, the target segments may not correspond to frequent patterns; thus frequency is not a target metric. Our problem is also different from general time-series segmentation which aims at partitioning data sequences into internally homogeneous segments (Keogh *et al.* 2001). This work followed an unsupervised framework, which relied on weak measures of quality that are based on information theory. In contrast, our objective is more specific; it is to segment data sequences such that goals can be accurately recognized. Thus, the segments that we discover are highly dependent on goals and activities, and may be completely different from the ones that resulted from an unsupervised method.

Algorithm Description

In this section, we first introduce our segmentation model used for activity recognition. Then we give a detailed description of our inference and learning algorithms.

Segmentation Model

Given a multivariate time series Y which consists of T samples $Y_1, Y_2, \cdots, Y_T$ and a goal label G, we propose a probabilistic model to represent the observation sequence on a user's activity. In our model, there are N_m motion patterns $P = \{P_1, P_2, \ldots, P_{N_m}\}$ that are generated by hidden goals. The motion patterns are represented by their respective model parameters $\Theta = \{\theta_1, \theta_2, \ldots, \theta_{N_m}\}$. Our objective is to partition the observation sequence Y into N_s segments, such that each segment can be represented by one of the N_m motion patterns.

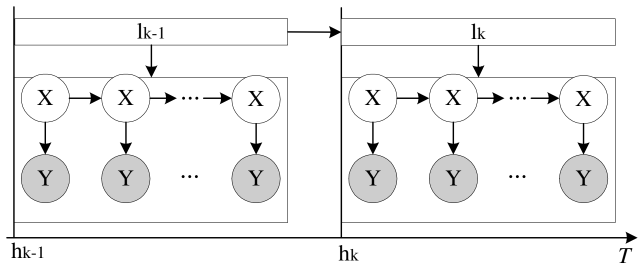

Figure 1: The Probabilistic Segmentation Model

We define segment labels as $L = \{l_1, l_2, \ldots, l_{N_s}\}$ and segmentation points as $H = \{h_1, h_2, \ldots, h_{N_s}\}$. As shown in Figure 1, each segment is labeled as l_k, where $1 \leq k \leq N_s$. The k^{th} segment starts from h_k and has a minimum length constraint $h_k - h_{k-1} \geq T_{min}$. The length of each segment may be different. We have $N_m \leq N_s$ because all the N_m motion patterns are learned from the entire sequence of N_s segments. Consequently, multiple segments can be represented by the same motion pattern. In addition, the relationship between two adjacent motion patterns is represented by the probability of transiting from one motion pattern to another. In general, our segmentation model first uses a Linear Dynamic System (LDS) model to capture the local linear dynamics involved in each segment, and then use a transition matrix to model the global non-linear dynamics in the stochastic process.

Motion Patterns

Intuitively, each motion pattern exhibits similar characteristics in terms of the magnitude and trends in the signal space. Accordingly, goals can be identified through sequences of consecutive motion patterns. We define a motion pattern as an LDS with the following state-space model:

$$\begin{cases} X_{t+1} = A_t X_t + W_t \\ Y_t = C_t X_t + B_t \end{cases} \tag{1}$$

where X_t is the hidden variable, and Y_t is the observed signal measurement at a time instant t. W_t and B_t are independent Gaussian noise with covariance matrices Q and R, respectively. A_t and C_t represent the state transition matrix and observation matrix, respectively. Therefore, the parameters of an LDS can be represented by $\theta = \{A, C, Q, R\}$. The length of a motion pattern should be at least T_{min} so that local dynamics can be captured; later, we will explore the effect of varying the lengths of T_{min}.

For a specific goal G, we assume that the transition probability among motion patterns satisfies a discrete first-order Markov process, which is represented by a transition matrix

$$M_{ij}(G) = P(l_k = j | l_{k-1} = i, G). \tag{2}$$

Such a transition matrix can model the global non-linear dynamics involved in the sequences with different goals, corresponding to different sequences of motion patterns that give rise to goals. Below, we will omit the parameter G when discussing M_{ij}.

Goal-Based Segmentation Algorithm

Given a sequence of observed measurements $Y_{1:T} = \{Y_1, Y_2, \ldots, Y_T\}$, the model parameters $\{\Theta, M\}$ can be learned using a maximum likelihood (ML) method

$$\{\hat{\Theta}, \hat{M}\} = \arg\max_{\{\Theta, M\}} P(Y_{1:T} | \Theta, M) \tag{3}$$

By introducing segment labels L and segmentation points H and applying the first-order Markov property, the above equation can be rewritten as:

$$\begin{aligned} P(Y_{1:T}|\Theta, M) &= \sum_{L,H} P(Y_{1:T}, L, H | \Theta, M) \quad (4) \\ &= \sum_{L,H} \Big[\prod_{j=1}^{N_s} P(Y_{h_j : h_{j+1}-1}) M_{l_j l_{j+1}} \Big] \end{aligned}$$

where $M_{l_{N_s} l_{N_s+1}} = 1$. In the above equation, $P(Y_{h_j:h_{j+1}-1})$ is the likelihood of observations given a LDS model, and $M_{l_j l_{j+1}}$ represents the transition probability between two adjacent LDS models.

Since the variables L and H are hidden, we can use an EM algorithm (Dempster, Laird, & B.Rubin 1977) to solve the above maximum likelihood problem. The algorithm iterates through two steps until it converges to a local optimum:

- E-step: An inference algorithm is used to find the optimal segment labels L and segmentation points H given the current model parameters Θ and M such that the likelihood $P(Y_{1:T}|\Theta, M)$ in equation (4) is maximized. The detailed algorithm, which is based on dynamic programming and similar to the one used in (Li, Wang, & Shum 2002), will be discussed later.
- M-step: Model parameters Θ are updated by fitting an LDS model to each segment. The transition matrix M is calculated as $M_{ij} = \sum_{k=2}^{N_s} \delta(l_{k-1} = i)\delta(l_k = j)$ by counting the labels of segments, where $\delta(C) = 1$ if and only if C is true. Then the matrix M is normalized such that $\sum_{j=1}^{N_m} M_{ij} = 1$.

Model Initialization The initialization of the EM algorithm is done by using a moving-window-based greedy approach. First, we use a subsequence of the length T_{min} starting from t_1 to fit an LDS model and learn the model parameters. We gradually increase the length of this subsequence and update the model parameters until the likelihood of the model drops dramatically. This implies the end of this segment and thus we label this segment with its corresponding LDS model. Then we restart the same process on a new subsequence of T_{min} except that we need to test all the existing

LDS models learnt from the preceding segments. If this subsequence can be fit well by one of these LDS models, it will be labeled as the one with the highest likelihood. If none of those LDS models fit the subsequence well, we introduce a new LDS model and repeat the above precess until the entire sequence is processed. In this way, we can provide our algorithm with initial model parameters, and let the EM algorithm fine-tune these parameters through an iterative procedure.

E-step: Inference Algorithm Inference is used as an important subroutine in the EM algorithm to find the optimal segmentation based on the current model parameters. We partition an observation sequence into a sequence of concatenated segments and label each segment as a motion pattern. We compute globally optimal segmentation points $H = \{h_2, \ldots, h_{N_s}\}$ and segment labels $L = \{l_1, \ldots, l_{N_s}\}$ by a dynamic programming algorithm shown below.

---

1. Initialize

$$J_1(t) = \max_{1\le i\le N_m} P(Y_{1:t}|\theta_i),$$
$$E_1(t) = \arg\max_i P(Y_{1:t}|\theta_i), \quad T_{min} \le t \le T$$

2. Iterate while $2 \le n \le T/T_{min}$

$$J_n(t) = \max_{\substack{1\le i\le N_m \\ (n-1)T_{\min}<b\le(t-T_{\min})}} [J_{n-1}(b-1)P(Y_{b:t}|\theta_i)M_{li}]$$
$$E_n(t), F_n(t) = \arg\max_{i,b}[J_{n-1}(b-1)P(Y_{b:t}|\theta_i)M_{li}]$$
$$\text{where } l = E_{n-1}(b-1).$$

3. Compute the final solution

$$J(T) = \max_{1\le n\le T/T_{min}} J_n(T)$$
$$N_s = \arg\max_n J_n(T)$$

4. Backtrack the segment points and labels

$$h_{N_s+1} = T+1, \quad l_{N_s} = E_{N_s(T)}, \quad h_1 = 1$$
$$h_n = F_n(h_{n+1}-1), l_{n-1} = E_{n-1}(h_n - 1), 1 < n \le N_s$$

In this algorithm, $J_n(t)$ represents the maximum likelihood calculated from partitioning the sequence ending at time t into a sequence of n segments. $E_n(t)$ and $F_n(t)$ denote the segment label and the beginning point of the last segment, respectively, in the sequence to achieve $J_n(t)$. To find the optimal segmentation point b for the n^{th} segment, the algorithm aims to maximize $J_n(t)$, the likelihood of partitioning $Y_{1:t}$ into n segments. In fact, $J_n(t)$ can be calculated based on three components: (1) the optimal solution to segmenting $Y_{1:(b-1)}$ into $n-1$ segments, with the last segment being recognized as the l^{th} motion pattern; (2) the likelihood of fitting $Y_{b:t}$ with current model parameters of motion patterns; (3) the transition probability from l to a motion pattern used to fit $Y_{b:t}$. Consequently, $Y_{b:t}$ is labeled as the i^{th} motion pattern such that $J_n(t)$ is maximized. The complexity of this algorithm is $O(N_s T^2)$, but it can be approximately optimized to be linear with the length of sequence T using the greedy algorithms of (Himberg *et al.* 2001).

M-step: Fitting an LDS Given a segmentation of an observation sequence, we can learn the model parameters of an LDS used to fit each segment. Since the observation sequence is a multivariate time series, different dimensions may correlate with each other in terms of their signal values. Therefore, we apply Singular Value Decomposition (SVD) on the observation sequence as follows:

$$Y = USV^T \tag{5}$$

where U is a column-orthonormal matrix representing the principal component directions. S is a diagonal matrix with nonnegative singular values in descending order along the diagonal. These singular values represent the importance of their corresponding principal components. The matrix V encodes the coefficients used to expand Y in terms of U. Since the top k principal components, $k \le m$, can capture a significant amount of information of the original data, it is possible to approximate each dimension by the linear combinations of the k most significant principal components. Therefore, after performing SVD, we can transform the data Y in the original space into X of low dimension

$$X = SV^T, \tag{6}$$

where $C = U$. At the same time, the effect of noise involved in Y can be dramatically reduced through SVD. Then the parameters A and Q of a first-order LDS can be estimated using a maximum likelihood estimation approach. The two parameters are both diagonal matrices, with each diagonal element $(1 \le i \le k)$ corresponding to one dimension of X, which are given by:

$$a_j = \frac{\sum_{i=1}^T x_t x_{t-1}}{\sum_{i=1}^T x_{t-1}^2}, \quad q_j = \frac{1}{T-1}\sum_{i=1}^T (x_t - a_i x_{t-1})^2 \tag{7}$$

After the learning process, each goal is associated with a set of motion patterns and their transition probabilities, which can be further used to perform goal recognition.

Goal Recognition

We can now infer the most likely goal that corresponds to a sequence of observations. We define this task of *goal recognition* as follows: given a sequence of segments $l_1, l_2, \ldots, l_{N_s}$ generated by the inference algorithm, infer the most likely goal G^*:

$$\begin{aligned} G^* &= \arg\max_{G_k} P(G_k|l_1, l_2, \ldots, l_{N_s}) \\ &= \arg\max_{G_k} P(G_k|l_{1:N_s}). \end{aligned}$$

By applying the Bayes' Rule, the above equation becomes:

$$\begin{aligned} G^* &= \arg\max_{G_k} \frac{P(l_{1:N_s}|G_k)P(G_k)}{P(l_{1:N_s})} \\ &= \arg\max_{G_k} P(l_{1:N_s}|G_k)P(G_k) \end{aligned}$$

where the term $P(l_{1:N_s})$ is a constant and can be dropped when performing the comparison. In addition, $P(l_{1:N_s}|G_k)$ can be calculated through the model parameters of motion patterns learned from the training data corresponding to G_k. In fact, we perform the segmentation and goal recognition task at the same time.

Experimental Results

We conducted experiments to evaluate our proposed algorithm in the office area shown in Figure 2. This environment is equipped with an IEEE 802.11b wireless network. A user's activities are carried out in the three main areas (Office, Room1 and Room2) and four hallways. Room1 and Room2 provide facilities for printing services and holding seminars. In this environment, there are eight access points that can be detected, of which four access points (AP's) are marked with solid circles in the figure. Six goals of a professor's activities, such as "*Seminar-in-Room1*" and "*Exit-through-Entrance1*", are modeled. We collected 180 traces using the device driver and API we have developed. Each trace is represented as a sequence of signal-strength measurements received from the AP's. We labeled each trace by its intended goal by hand for evaluation. In addition to our goal-based segmentation algorithm (which we refer to as *GBS-based model*), we implemented another sensor-model-based recognition algorithm proposed in our previous work (Yin, Chai, & Yang 2004) (which we refer to as *SM-based model*) for comparison.

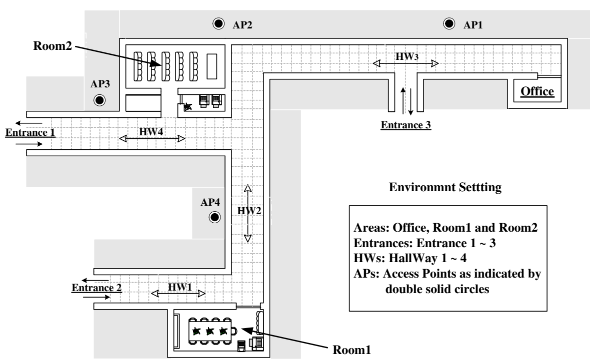

Figure 2: The Layout of Office Area

Sensor-Model-Based Recognition Algorithm

SM-based model is a two-level Bayesian model used to infer a user's goal from traces of signal-strength measurements. The lowest level is a location-level sensor model. In this model, we relate each location in a finite location space $\mathbb{S} = \{s_1, \ldots, s_n\}$ with observations in an observation space $\mathbb{Y} = \{Y_1, \ldots, Y_m\}$. The *location-level sensor model* is defined as a predictive model using the conditional probabilities $Pr(Y_j|s_i)$, the likelihood of observing sensor measurement $Y_j \in \mathbb{Y}$ at location $s_i \in \mathbb{S}$.

Based on the location-level sensor model, actions can be next inferred in this framework by computing $P(A_\tau|Y_1, Y_2, \ldots, Y_t)$. At the next level up, we can further infer goals from actions. In particular, given an inferred temporal sequence of actions obtained so far $A_1, A_2, \ldots, A_t$, find the corresponding most likely goal G^*:

$$\begin{aligned} G^* &= \arg\max_{G_k} P(G_k|A_1, A_2, \ldots, A_t) \\ &= \arg\max_{G_k} P(G_k|A_{1,t}). \end{aligned}$$

By applying the Bayes' Rule, the above formula becomes:

$$\begin{aligned} G^* &= \arg\max_{G_k} \frac{P(A_{1,t}|G_k)P(G_k)}{P(A_{1,t})} \\ &= \arg\max_{G_k} P(A_{1,t}|G_k)P(G_k) \end{aligned}$$

In our implementation of this model, much calibration effort was incurred for training the location-level sensor model. In our experiments, the environment was modeled as a space of 99 locations, each representing a 1.5-meter grid cell. To build the sensor model, we collected 100 samples at each location, one per second. This calibration work requires that we label each sample by its corresponding location. The whole process took several days to finish.

Goal-Based Segmentation Algorithm Performance

Figure 3 gives an example of conducting our proposed algorithm on an observation trace collected from four AP's. This sequence records the movements of a user from "Entrance2" to his office. As we can see from the figure, this sequence is partitioned into five segments and each segment represents a typical motion pattern. For example, the segments *seg3* and *seg4* indicate the user walks through HW3 because the signals from AP1 significantly increase and then decrease, while the signals from the other AP's gradually decrease; *seg5* indicates the user stays in his office because signals are relatively stable. This shows that our algorithm can segment an observation trace into meaningful motion patterns. In our experiments, such motion patterns can be usually found within four iterations using the EM algorithm.

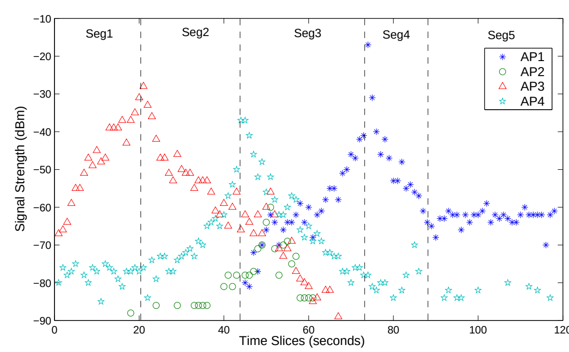

Figure 3: An Observation Trace from Four AP's

Figure 4 shows the change of log-likelihood with respect to different minimum length constraints T_{min}. T_{min} affects the parameters of LDS models fitted on a signal sequence. Therefore, we performed experiments to determine an optimal T_{min} for each goal. We can see from the figure that the optimal T_{min} for $G1$ and $G2$ are 35 and 25, respectively. At these two points, the maximum log-likelihood of signal sequences for the two goals can be reached. We can also observe that when T_{min} is either too large or too small, the performance of segmentation and goal recognition degrades. This is because when T_{min} is too large, the segments cannot be modeled well by an LDS model. When T_{min} is too small, the search space becomes large, and the greedy search algorithm can easily get trapped in a local minimum.

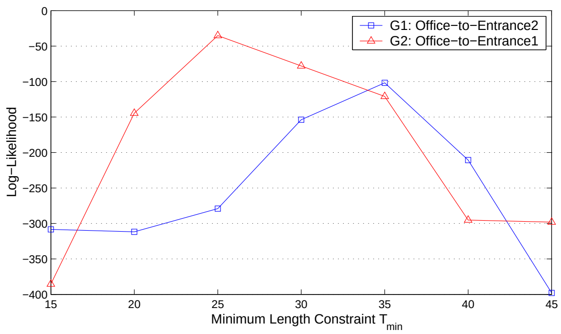

Figure 4: Log-Likelihood vs. Minimum Length Constraint

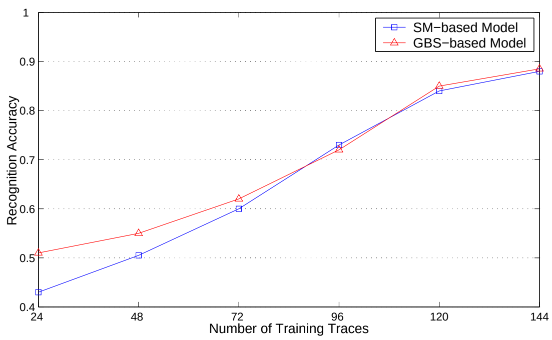

Figure 5: Accuracy vs. Number of Training Traces

Figure 5 compares the average recognition accuracy of *GBS-based model* and *SM-based model* with respect to different numbers of training traces. As we can see from the figure, as the number of training traces increases, the recognition accuracy of the two models increases steadily. This occurs because, the model parameters can be estimated more accurately when more training data are used. As a whole, the performance of *GBS-based model* without building sensor models is comparable to that of *SM-based model*. In particular, when the training data are sparse, *GBS-based model* can even outperform *SM-based model*. This is because *SM-based model* employs a hierarchical model to infer a user's locations, actions and goals based on the sensor model. In this model, a large number of parameters need to be estimated and, as a result, an accurate estimation can not be obtained without a lot of training data. *GBS-based model*, in contrast, relies on the characteristics of signals themselves to generate motion patterns, which reduce the number of parameters that need to be estimated. Therefore, less training data are required in the learning process.

Conclusions and Future Work

We have presented a novel approach for activity recognition through segmenting low-level signal sequences with a goal-based model. In our approach, we applied a probabilistic model in which each segment of signals is represented as an LDS model and the transitions between segments as a Markov process conditioned on goals. Our EM learning algorithm simultaneously learns the motion-pattern boundaries and probability models for goals, which in turn can be used to accurately recognize activities in an online phase. We have conducted experiments in a real wireless environment, in which we showed that our proposed model can accurately recognize a user's goals with less calibration effort.

In the future, we wish to continue in this direction in reusing the motion patterns that are obtained in this analysis. One application is to use them for the task of planning, and another is to recognize abnormal activities performed by unknown agents for security monitoring applications.

Acknowledgments

We would like to thank Hong Kong RGC for supporting this research under grant HKUST6187/04E.

References

Bregler, C. 1997. Learning and recognizing human dynamics in video sequences. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 568–574.

Czielniak, G.; Bennewitz, M.; and Burgard, W. 2003. Where is ...? learning and utilizing motion patterns of persons with mobile robots. In *Proceedings of International Joint Conference on Artificial Intelligence*, 909–914.

Dempster, A. P.; Laird, N. M.; and B.Rubin, D. 1977. Maximum likelihood from incomplete data via EM algorithm. *Journal of the Royal Statistical Society Series B* 39:1–38.

Goldman, R.; Geib, C.; and Miller, C. 1999. A new model of plan recognition. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 245–254.

H. H. Bui, S. V., and West, G. 2002. Policy recognition in the abstract hidden markov model. *Journal of Artificial Intelligence Research* 17:451–499.

Himberg, J.; Korpiaho, K.; Mannila, H.; Tikanmaki, J.; and Toivonen, H. 2001. Time series segmentation for context recognition in mobile devices. In *Proceedings of the International Conference on Data Mining*, 203–210.

Kautz, H., and Allen, J. 1986. Generalized plan recognition. In *Proceedings of the National Conference on Artificial Intelligence*, 32–37.

Keogh, E. J.; Chu, S.; Hart, D.; and Pazzani, M. J. 2001. An online algorithm for segmenting time series. In *Proceedings of the International Conference on Data Mining*, 289–296.

Li, Y.; Wang, T.; and Shum, H.-Y. 2002. Motion texture: a two-level statistical model for character motion synthesis. In *Proceedings of the International Conference on Computer Graphics and Interactive Techniques*, 465–472.

Liao, L.; Fox, D.; and Kautz, H. 2004. Learning and inferring transportation routines. In *Proceedings of the National Conference in Artificial Intelligence*, 348–353.

Oates, T. 2002. PERUSE: An unsupervised algorithm for finding recurring patterns in time series. In *Proceedings of the International Conference on Data Mining*, 330–337.

Peursum, P.; Bui, H. H.; Venkatesh, S.; and West, G. 2004. Human action segmentation via controlled use of missing data in hmms. In *Proceedings of International Conference on Pattern Recoginition*, 440–445.

Yin, J.; Chai, X. Y.; and Yang, Q. 2004. High-level goal recognition in a wireless lan. In *Proceedings of the National Conference in Artificial Intelligence*, 578–584.

Zaki, M. J. 2001. SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning* 42:31–60.