In *Proceedings of the National Conference on Artificial Intelligence (AAAI), 2004.*

Learning and Inferring Transportation Routines

Lin Liao Dieter Fox Henry Kautz
Department of Computer Science & Engineering
University of Washington
Seattle, WA 98195

Abstract

This paper introduces a hierarchical Markov model that can learn and infer a user's daily movements through the community. The model uses multiple levels of abstraction in order to bridge the gap between raw GPS sensor measurements and high level information such as a user's mode of transportation or her goal. We apply Rao-Blackwellised particle filters for efficient inference both at the low level and at the higher levels of the hierarchy. Significant locations such as goals or locations where the user frequently changes mode of transportation are learned from GPS data logs without requiring any manual labeling. We show how to detect abnormal behaviors (*e.g.* taking a wrong bus) by concurrently tracking his activities with a trained and a prior model. Experiments show that our model is able to accurately predict the goals of a person and to recognize situations in which the user performs unknown activities.

Introduction

The advent of low-cost GPS (global positioning system) technology has led to great interest in developing commercial applications that take advantage of information about a user's current location — for example, 911 service. But localization based on immediate sensor data is only one small part of inferring a user's spatial context. In this paper we describe a system that creates a *probabilistic model* of a user's daily movements using unsupervised learning from raw GPS data. The model allows one to:

- Infer the locations of usual goals, such as home or workplace;
- Infer a user's mode of transportation, such as foot, car, or bus, and predict when and where she will change modes;
- Predict her future movements, both in the short term (will the user turn left at the next street corner?) and in terms of distant goals (is she going to her workplace?);
- Infer when a user has *broken his ordinary routine* in a way that may indicate that he has made an error, such as failing to get off his bus at his usual stop on the way home;
- Robustly track and predict locations even in the presence of total loss of GPS signals and other sources of noise.

A motivating application for this work is the development of personal guidance systems that help cognitively-impaired individuals move safely and independently throughout their community. Other potential applications include *customized* "just in time" information services (for example, provide the user with current bus schedule information when she is likely to need it or real time traffic conditions on her future trajectories) and self-configuring appointment calendars.

Our approach is based on an abstract hierarchical Markov model (Bui, Venkatesh, & West 2002) of a user from data collected by a small wearable GPS unit. The model is compactly represented by a dynamic Bayesian network, and inference is efficiently performed using Rao-Blackwellised particle filtering both for the low level sensor integration and for the higher levels of the hierarchical model.

The main research contribution in this paper is a method for learning hierarchical predictive models of user location and transportation mode in an unsupervised manner. While previous authors described inference in hierarchical models (Bui, Venkatesh, & West 2002) and learning flat transportation models (Patterson *et al.* 2003), our work is the first to combine the techniques. A second research contribution are initial results on inferring *user errors and deviations from routine* by model selection. We demonstrate the effectiveness of this approach with an example of the system recognizing when the user has missed his bus stop.

This paper is organized as follows. In the next section, we discuss related work. Then, we provide an overview of the activity model, followed by a description of inference and learning mechanisms. Before concluding, we present experimental results that show the capabilities of our approach.

Related work

Over the last years, estimating a person's activities has gained increased interest in the AI, robotics, and ubiquitous computing communities. (Ashbrook & Starner 2003) learn significant locations from logs of GPS measurements by determining the time a person spends at a certain location. For these locations, they use frequency counting to estimate the transition parameters of a second-order Markov model. Their approach then predicts the next goal based on the current and the previous goals. In contrast to our approach, their model is not able to refine the goal estimates using GPS information observed when moving from one significant location to another. Furthermore, such a coarse representation does not allow the detection of potential user errors. In our previous work (Patterson *et al.* 2003), we estimate a person's location and mode of transportation from GPS measurements using a "flat" model. Since the model has no notion of significant locations, it is not able to predict the high-level goal of a person. By conditioning on goals and segments of a trip, our hierarchical model is able to learn more specific motion patterns of a person, which also enables us to detect user errors.

In the context of probabilistic plan recognition, (Bui, Venkatesh, & West 2002) introduced the abstract hidden Markov model, which uses hierarchical representations to efficiently infer a person's goal in an indoor environment from camera information. (Bui 2003) extended this model to include memory nodes, which enables the transfer of context information over multiple time steps. Bui and colleagues introduced efficient inference algorithms for their models using Rao-Blackwellised particle filters. Since our model has a similar structure to theirs, we apply the inference mechanisms developed in (Bui 2003). Our work goes beyond the work of Bui *et al.* in that we show how to learn the structure and the parameters of the hierarchical activity model from data. Furthermore, our low level estimation problem is more challenging than their indoor tracking problem. In the context of mobile robotics, (Cielniak, Bennewitz, & Burgard 2003) apply a two level model to track and predict the location of people using a mobile robot equipped with a laser range-finder. Their model learns a person's trajectories using a mixtures of Gaussians approach. Due to this representation, they are only able to track a person along paths the robot has observed during training. Thus, the technique is not able to track and detect novel behaviors.

The task of detecting abnormal events in time series data (called *novelty detection*) has been studied extensively in the data-mining community (Guralnik & Srivastava 1999), but remains an open and challenging research problem. We present the first results on abnormality detection in location and transportation prediction using a simple and effective model selection approach based on comparing the likelihood of a learned hierarchical model against that of a prior model.

Hierarchical Activity Model

We estimate a person's activities using the three level dynamic Bayesian network model shown in Fig. 1. The individual nodes in such a temporal graphical model represent different parts of the state space and the arcs indicate dependencies between the nodes (Murphy 2002). Temporal dependencies are represented by arcs connecting the two time slices $k-1$ and k. The highest level of the model, denoted goal level, represents the person's next goal, *e.g.*, her work place. The trip segment level represents the mode of transportation and the locations at which the person transfers from one mode to another. The person's location and motion velocity are estimated from the GPS sensor measurements at the lowest level of the model.

Locations and transportation modes We denote by $x_k = \langle l_k, v_k, c_k \rangle$ the location and motion velocity of the person, and the location of the person's car [1] (subscripts k indicate discrete time). As we will describe in the next section, locations are estimated on a graph structure representing a street map. GPS sensor measurements, z_k, are generated by the person carrying a GPS sensor. Since measurements are given in continuous xy-coordinates, they have to be "snapped" to an edge in the graph structure. The edge to which a specific measurement is "snapped" is estimated by the association variable θ_k. The location of the person at time k depends on his previous location, l_{k-1}, the motion velocity, v_k, and the vertex transition, τ_k. Vertex transitions τ model the decision a person makes when moving over a vertex in the graph, for example, to turn right when crossing a street intersection.

[1] We include the car location because it strongly affects whether the person can switch to the car mode.

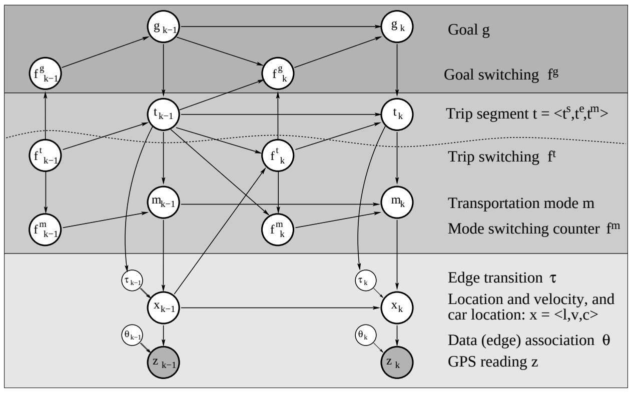
Goal g
Goal switching f^g
Trip segment t = <t^s,t^e,t^m>
Trip switching f^t
Transportation mode m
Mode switching counter f^m
Edge transition τ
Location and velocity, and car location: x = <l,v,c>
Data (edge) association θ
GPS reading z

Figure 1: Hierarchical activity model representing a person's outdoor movements during everyday activities. The upper level estimates the current goal, the middle layer represents segments of a trip and mode of transportation, and the lowest layer estimates the person's location. The dashed line indicates the flat model.

The mode of transportation can take on four different values $m_k \in \{BUS, FOOT, CAR, BUILDING\}$. Similar to (Patterson *et al.* 2003), these modes influence the motion velocity, which is picked from a Gaussian mixture model. For example, the walking mode draws velocities only from the Gaussian representing slow motion. $BUILDING$ is a special mode that occurs only when the GPS signal is lost for significantly long time. Finally, the location of the car only changes when the person is in the CAR mode, in which the car location is set to the person's location.

Trip segments A trip segment is defined by its start location, t_k^s, end location, t_k^e, and the mode of transportation, t_k^m, the person uses during the segment. For example, a trip segment models information such as "she gets on the bus at location t_k^s and takes the bus up to location t_k^e, where she gets off the bus". In addition to transportation mode, a trip segment predicts the route on which the person gets from t_k^s to t_k^e. This route is not specified through a deterministic sequence of edges on the graph but rather through transition probabilities on the graph. These probabilities determine the prediction of the person's motion direction when crossing a vertex in the graph, as indicated by the arc from t_k to τ_k.

The transfer between modes and trip segments is handled by the switching nodes f_k^m and f_k^t, respectively. More specifically, the binary trip switching node is set to true whenever the person reaches the end location t_k^e of the current trip segment. In this case, the trip segment is allowed to switch with the constraint that the start location of the next segment is identical to the end location of the current segment. The next trip segment is chosen according to the segment transition of the current goal g_k. Once the next trip segment is active, the person still has to change mode of transportation. This does not happen instantaneously, since, for example, a person has to wait for the bus even though he already reached the bus stop (and thus entered the bus trip segment). This semi-Markov property of delayed mode switching is modeled by the node f_k^m, which is a counter that measures the time steps until the next transportation mode is entered. The counter is initialized by the next trip segment, then decremented until it reaches a value of zero, which triggers the mode switch.

Goals A goal represents the current target location of the person. Goals include locations such as the person's home, work place, the grocery store, and locations of friends. These goals are also contained in the trip segment level. Thus, the goal of the person can only change when the person reaches the end of a trip segment. The goal switching node f_k^g is true only when the trip switching node f_k^t is true and the end of the current trip segment t_k is identical to the goal g_k. If the goal switches, the next goal is chosen according to a learned goal transition model.

Inference, Learning, and Error Detection

Inference

The independence structure of our hierarchical activity model allows us to use efficient inference developed for abstract hidden Markov memory models (Bui 2003). This technique relies on Rao-Blackwellised particle filters, where the states at the lowest level are estimated using particle filters, and higher levels are solved analytically conditioned on the low level particles. For brevity, we focus on the task of estimating a person's location and mode of transportation using GPS measurements. Inference at higher levels will only be outlined, since it is very similar to (Bui 2003).

GPS-based tracking on street maps Our previous work (Patterson *et al.* 2003) uses a "flat" model for location and transportation mode estimation. Such a model is obtained by removing the nodes g, f^g, and t from the activity model shown in Fig. 1, as indicated by the dotted line. In that model, we estimate a person's location on a street map represented by a graph-structure $S = (V, E)$, where V is the set of vertices, v_i, and E is the set of edges, e_j. Typically, vertices are intersections and the length of edges corresponds to city blocks. The person can switch mode of transportation whenever she is near her car or a bus stop. While our previous work uses particle filters for inference, we now rely on a more efficient Rao-Blackwellised solution to the problem (Doucet *et al.* 2000), which is based on the following factorization of the posterior:

$$p(x_k, m_k, f_k^t, f_k^m, \theta_k, \tau_k \mid z_{1:k}) \; = \; p(x_k | m_k, f_k^t, f_k^m, \theta_k, \tau_k, z_{1:k}) \, p(m_k, f_k^t, f_k^m, \theta_k, \tau_k | z_{1:k}) \quad (1)$$

The posterior at time k is conditioned on $z_{1:k}$, the sequence of GPS measurements observed so far. The factorization (1) separates the state space of our estimation problem into its continuous and discrete parts. The continuous part represents the location and motion velocity of the person, x_k, and the discrete part represents the remaining quantities including transportation mode m_k, edge association θ_k, edge transition τ_k, and switching nodes f_k^m and f_k^t.

Rao-Blackwellised particle filters (RBPF) estimate this factorized posterior by sampling the discrete states using a particle filter and then estimating the person's location and motion velocity using Kalman filters conditioned on the samples. More specifically, RBPFs represent posteriors by sets of weighted samples, or particles:

$$S_k = \{s_k^{(i)}, w_k^{(i)} \mid 1 \le i \le N\}$$

Each $s_k^{(i)} = \langle \langle {\mu_k}^{(i)}, {\Sigma_k}^{(i)} \rangle, {m_k}^{(i)}, f_k^{m(i)}, f_k^{t(i)}, {\theta_k}^{(i)}, {\tau_k}^{(i)} \rangle$, where the person's location and velocity are represented by $\langle {\mu_k}^{(i)}, {\Sigma_k}^{(i)} \rangle$, the mean and covariance of the Kalman filter, which represents posteriors by Gaussian approximations (Bar-Shalom, Li, & Kirubarajan 2001). The other components of the particle are instances of the discrete parts of the state space. At each time step, RBPFs first sample the discrete components from the posterior at the previous time $k-1$. This can be done stepwise by simulating (1) from right to left, using the independence represented in the activity model (see Fig. 1). Then, the continuous part is updated analytically using Kalman filters.

The discrete variables are sampled as follows. The value of the trip switching node $f_k^{t(i)}$ is sampled conditioned on the previous location of the person x_{k-1}. For example, whenever the person approaches a bus stop, $f_k^{t(i)}$ is set to true with a small probability. If $f_k^{t(i)} = \mathrm{T}$, then the value of the mode switching counter $f_k^{m(i)}$ is initialized to the waiting time. Otherwise, the value of $f_k^{m(i)}$ is decremented, and when it reaches zero, the new transportation mode ${m_k}^{(i)}$ is sampled according to the mode transition probability; otherwise, ${m_k}^{(i)} = {m_{k-1}}^{(i)}$. The value of the edge transition variable τ_k determines, for example, whether the person moves straight or turns right at the next intersection. ${\tau_k}^{(i)}$ is sampled based on the previous position of the person and a learned transition model. Finally, the edge association variable θ_k "snaps" the GPS reading to a street in the map. This step is crucial for the Kalman filter update described below. To sample $\theta_k^{(i)}$, we first determine the distance between the measurement, z_k, and the different streets in the vicinity. The probability of "snapping" z_k to one of these streets is then computed from this distance.

At this step of the algorithm, we can assume that all discrete values of a sample are already generated, that is, $s_k^{(i)} = \langle \langle \cdot \rangle, {m_k}^{(i)}, f_k^{m(i)}, f_k^{t(i)}, {\theta_k}^{(i)}, {\tau_k}^{(i)} \rangle$. The RBPF now generates the missing values $\langle {\mu_k}^{(i)}, {\Sigma_k}^{(i)} \rangle$ by updating the Kalman filter conditioned on the already sampled values. To see, let us rewrite the left term on the right hand side of (1):

$$p(x_k \mid m_k^{(i)}, f_k^{m(i)}, f_k^{t(i)}, \theta_k^{(i)}, \tau_k^{(i)}, z_{1:k}) \;\propto\; p(z_k | x_k, \theta_k^{(i)}) \int p(x_k | m_k^{(i)}, \tau_k^{(i)}, x_{k-1}^{(i)}) \; p(x_{k-1}^{(i)} | z_{1:k-1}) \; \mathrm{d}x_{k-1}^{(i)} \quad (2)$$

(2) follows by applying Bayes rule and the independences in our estimation problem. It represents the standard recursive Bayes filter update rule; see (Bar-Shalom, Li, & Kirubarajan 2001) for details. The prior probability is given by the Gaussian of the previous Kalman filter estimate: $p(x_{k-1}^{(i)} | z_{1:k-1}) = \mathcal{N}(x_{k-1}^{(i)}; \mu_{k-1}^{(i)}, \Sigma_{k-1}^{(i)})$. The Kalman filter implements the update rule (2) by two steps: a *prediction step* followed by a *correction step*.

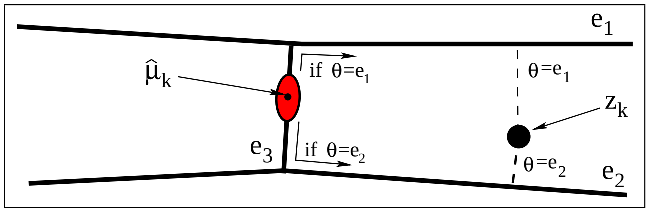

Figure 2: Kalman filter update and data association: The person is located on edge e_3. The continuous coordinates of the GPS measurement, z_k, are between edges e_1 and e_2. Depending on the value of the edge association, θ, the correction step moves the estimate up-wards or down-wards.

In the prediction step, the distance traveled since the last filter update is predicted using the previous velocity estimate. The prediction, $\langle \hat{\mu}_k^{(i)}, \hat{\Sigma}_k^{(i)} \rangle$, results then from shifting and convolving the previous estimate by the predicted motion, thereby implementing the integration in (2). This prediction step is straightforward if the person stays on the same edge of the graph. If she transits over a vertex of the graph, then the edge is given by the already sampled edge transition ${\tau_k}^{(i)}$.

In the correction step, the predicted estimate $\langle \hat{\mu}_k^{(i)}, \hat{\Sigma}_k^{(i)} \rangle$ is corrected based on the most recent GPS measurement, z_k. Intuitively, this correction compares the predicted mean $\hat{\mu}_k^{(i)}$ with the location of z_k and shifts the mean toward the measurement (under consideration of the uncertainties). The correction step is illustrated in Fig. 2. The predicted location is on edge e_3, and the GPS sensor reports a measurement, z_k, between edges e_1 and e_2. Depending on whether z_k originates from e_1 or e_2, the predicted estimate is corrected either up-wards or down-wards.[2] The already sampled value of the edge association variable ${\theta_k}^{(i)}$ uniquely determines to which edge the reading is "snapped" (see previous paragraph).

After all components of each particle are generated, the importance weights of the particles are updated. This is done by computing the likelihood of the GPS measurement z_k, which is provided by the update innovations of the Kalman filters (Doucet *et al.* 2000).

Goal and trip segment estimation So far, we concentrated on the estimation in a flat model. To further estimate a person's goal and trip segment, we apply the inference algorithm used for the abstract hidden Markov memory models (Bui 2003). More specifically, we use a Rao-Blackwellised particle filter both at the low level and at the higher levels. Each sample of the resulting particle filter contains the discrete and continuous states described in the previous section, and a joint distribution over the goals and trip segments. These additional distributions are updated using exact inference. To summarize, at each time step, the filter is updated as follows (see (Bui 2003)):

1. For each particle, sample the discrete states, $m_k^{(i)}, f_k^{m(i)}, f_k^{t(i)}, f_k^{g(i)}, \theta_k^{(i)}, \tau_k^{(i)}$, update the continuous state $\langle \mu_k^{(i)}, \Sigma_k^{(i)} \rangle$ by performing one-step Kalman filtering, and compute importance weight $w_k^{(i)}$.
2. Do re-sampling according to the importance weights.
3. For each particle, perform one-step exact inference to update the distribution of goals $g_k^{(i)}$ and trip segments $t_k^{(i)}$.

[2] Alternatively, one could compute the innovation in xy-space and project it onto the graph. However, such an approach can result in "stuck situations", for example, in dead ends on the graph.

The first step is extremely similar to the flat model described in the previous section. The main difference lies in the fact that the transportation mode $m_k^{(i)}$, the trip switching $f_k^{t(i)}$, and the edge transitions $\tau_k^{(i)}$ are sampled *conditioned* on the estimates of the high level goal and trip segment. Thereby, the sampling is adjusted to the current high level information.

Learning

Learning of the hierarchical model includes two procedures: structural learning and parameter learning, both are completely unsupervised. Structural learning searches for the significant locations, *i.e.*, usual goals and mode transfer locations, from GPS logs collected over an extended period of time. To do that, we apply expectation maximization (EM) using the "flat" activity model described above (called flat EM). When it finishes, the structure of the model is determined. EM is then used to estimate the transition probabilities in the hierarchical model (called hierarchical EM).

Finding goals We consider goal locations to be those locations where a person typically spends extended periods of time. (Ashbrook & Starner 2003) extract significant locations by detecting places where the GPS signal is lost. The disadvantage of such an approach is that it can only detect indoor goals. To overcome this problem, we store for each edge on the graph how long the person stays on this edge, estimated during the flat EM. Since we model loss of GPS signal by transiting into a "BUILDING" mode, our model can thus detect both indoor and outdoor goals. Once significant edges are detected, they are clustered by combining edges that are connected or very close.

Finding mode transfer locations The mode transition probabilities for each street are estimated during the flat EM. Even before learning, knowledge about the bus stops and the fact that the car is either parked or moves with the person, already provides important constraints on mode transitions. In the E-step, both a forward and a backward filtering pass are performed and the transition counts of the two passes are combined. Then in the M-step, the parameters are updated based on the counts. The mode transfer locations for a user, *i.e.*, usual bus stops and parking lots, are then those locations at which the mode switching exceeds a certain threshold.

Estimating transition matrices Once goals and trip segments are determined, we can extend the flat model by inserting these significant locations into the higher levels of the activity model. Then, we can re-use the GPS data in the hierarchical EM to estimate the transition matrices between the goals, between the trip segments given the goal, and between the adjacent streets given the trip segment. Hierarchical EM is similar to flat EM. During the E-steps, smoothing is performed by tracking the states both forward and backward in time. The M-steps update the model parameters using the frequency counts generated in the E-step. All transition parameters are smoothed using Dirichlet priors.

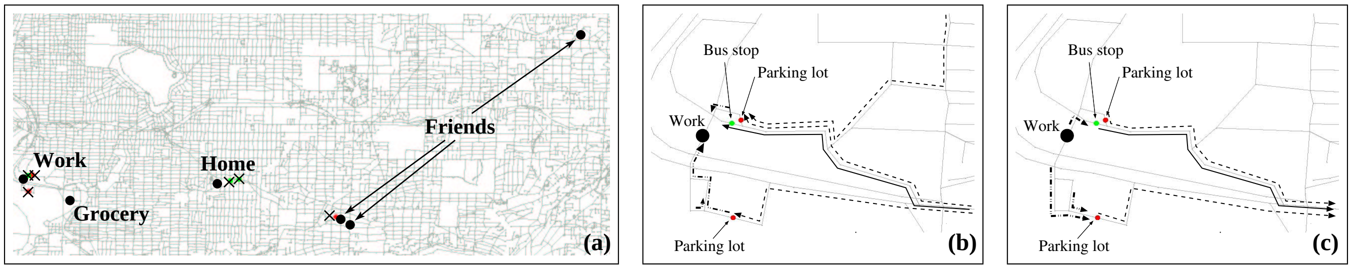

Figure 3: (a) Street map along with goals (dots) learned from 30 days of data. Learned trip switching locations are indicated by cross marks. From home, the person either walks to one of the two bus stops or takes the car, which is not learned as a trip switching location since the car is parked inside the house. (b) Zoom into the area around the work place. Shown are very likely transitions (probability above 0.75), given that the goal is the work place (dashed lines indicate car mode, solid lines bus, and dashed-dotted lines foot). (c) Learned transitions in the same area conditioned on the home being the goal.

Detection of User Errors

If the person always repeats her past activities, activity tracking can be done with only a small number of particles in the learned model. This is mainly because the model has low uncertainty in where the person switches modes and goals. In reality, however, people often perform novel activities or commit some errors. The most straightforward way to model abnormalities is to add an unknown goal and an unknown mode transfer location to the learned model, and estimate the probability of the unknowns. However, this means the person can change mode and goal *everywhere*, because any place could be an unknown goal or transfer location. This would require a huge number of particles to track correctly.

Instead, we use two different trackers simultaneously and perform model monitoring by computing the Bayes factors between the two models (West & Harrison 1997). The first tracker uses hierarchical inference on the learned model that models the person's ordinary routine. The second uses a flat model with the *apriori* parameter settings; these account for general physical constraints but are not adjusted to the individual's ordinary routines. The trackers are run in parallel, and the probability of each model is calculated from the observation likelihoods of the two models. When the user is following her ordinary routine the hierarchical model has higher likelihoods, but when the user does something unexpected the general flat model becomes more likely.

Both trackers can be run very efficiently. The hierarchical tracker has "expensive" particles, each containing much state information, but requires few particles for accurate tracking. The flat, untrained tracker needs more particles to maintain tracking, but each particle is cheaper to compute. Furthermore, calculating the likelihood of a model introduces no extra expense, because the value already needs to be computed as part of importance weighting.

Experimental Results

We collected a log of 60 days of GPS data from one person using a wearable GPS unit. We use the first 30 days for learning and the other 30 days for the empirical comparison.

Activity model learning The learning was done completely unsupervised without any manual labeling. The structural learning precisely identifies the subject's six most common transportation goals and all frequently used bus stops and parking lots, as shown in Fig. 3 (a). After recognizing the goals and transfer locations, parameter learning estimates the transition matrices at all levels of the model. Fig. 3 (b) and (c) show the learned street transitions given high-level information. The model successfully discovered the most frequent trajectories for traveling from home to the workplace and vice-versa, as well as other common trips, such as to the homes of friends.

Empirical comparison to other models The hierarchical model is very expressive and able to answer many useful queries. For example, many applications need to query the probability of a given goal. Here we compare the performance of our hierarchical model to the goal prediction of a flat model (Patterson *et al.* 2003) and a second-order Markov model between goals (2MM for simplicity) (Ashbrook & Starner 2003).

A flat model only keeps a first-order Markov model over the street blocks. Thus, in order to calculate the probability of a goal, one must calculate the sum over all possible paths to the goal, which is intractable if the goal is far away. A reasonable approximation is to compute the probability of the most likely path to the goal. Fig. 4 (a) compares the result of such a query on the probability of the goal being the work place during an episode of traveling from home to work. As one can see, quite early on the hierarchical model assigns a high probability to the true goal, while the estimate from the flat model is meaningless until the user is near the goal.

Model	Avg. accuracy at given time			
	beginning	25%	50%	75%
2MM	0.69	0.69	0.69	0.69
Hierarchical model	0.66	0.75	0.82	0.98

Table 1: Goal predictions using 2MM and hierarchical model

The 2MM models the goal transitions explicitly, but it cannot refine the prediction using the observations collected during transit. To show the difference, we labeled the 30 days of test data with the true goals and computed the prediction accuracy using the 2MM and our hierarchical model, which are learned using the same training data. The average prediction accuracies at the beginning of the trips and after 25%, 50%, 75% of the trips are listed in Table 1. At the beginning, our model predicts the next goal using first-order transition matrices; it performs a little worse than the 2MM. But by integrating real time measurements, our predictions become more accurate while 2MM's estimates remain the same.

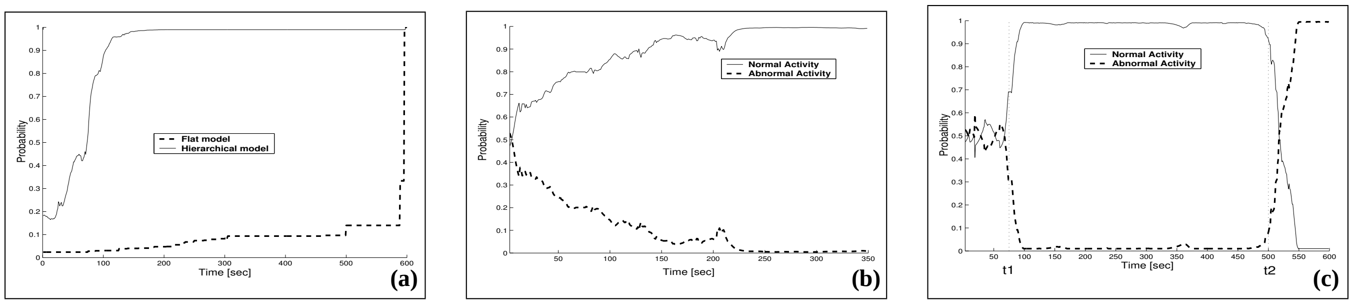

Figure 4: (a) Probability of the true goal (work place) during an episode from home to work, estimated using the flat and the hierarchical model. (b,c) Probability of an activity being normal or abnormal, estimated by the concurrent trackers. (b) Normal activity: driving from home to work. (c) Abnormal activity: missing to get off the bus at time t2.

Detection of user errors Another important feature of our model is the capability to capture user errors using the parallel tracking approach. To demonstrate the performance of parallel tracking, we did two experiments, with a subject who sometimes drives and sometimes takes the bus from work to home. In the first experiment, the subject drove from home to work as usual. In the second experiment, the subject took his usual bus toward home but failed to get off the bus at the usual stop. In each experiment, we determine the probability of each tracker over time, as shown in Fig. 4 (b) and (c). In the first experiment, because the trajectory matches the learned model well, the normal tracker quickly becomes dominant. The second experiment starts when the subject is waiting at the bus stop. At time t1, the person gets on the bus. Since the route is already well-learned, the belief that the subject is going home becomes high. At time t2, however, the person misses the usual bus stop. At this point the probability of the learned model quickly drops, while that of the flat "abnormal" model quickly rises. We performed additional experiments in which the goal and transportation mode were instantiated explicitly in the model. In such a setting, the model is able to quickly determine when the user deviates from the desired behavior; an ability that can be very useful to monitor and guide cognitively-impaired individuals.

Conclusions and Future Work

We have described the foundations and experimental validation of a hierarchical model that can learn and infer a user's daily movements and use of different modes of transportation. The model can be learned using unlabeled data, and online inference can be efficiently performed. Our results show that the approach can provide predictions of movements to distant goals, and support a simple and effective strategy for detecting novel events that may indicate user errors.

Our future work builds upon the foundation laid in this paper in several directions. One obvious extension is to incorporate information about time of day and the day of the week into the model, which we expect to greatly enhance predictive power. We furthermore plan to use probabilistic relational models (Getoor *et al.* 2001) to better represent and learn different types of locations. Using relational model learning, specific features of location types can be learned from data sets collected by several people. The same data can also be used to create loosely-coupled models of several individuals, so that one can predict joint activities, such as when two people will meet. Finally, the approach described here will be incorporated into a safety monitoring and guidance system that we are constructing for cognitively-impaired individuals who often become lost and have difficulty in using public transportation safely.

Acknowledgments

We thank Donald J. Patterson for providing useful software, data, and discussions in the course of this work. This research is based on work that has been supported by the NSF under grant numbers IIS-0225774 and IIS-0093406, and by DARPA's SDR Programme (grant number NBCHC020073).

References

Ashbrook, D., and Starner, T. 2003. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing* 7(5).

Bar-Shalom, Y.; Li, X.-R.; and Kirubarajan, T. 2001. *Estimation with Applications to Tracking and Navigation*. John Wiley.

Bui, H.; Venkatesh, S.; and West, G. 2002. Policy recognition in the abstract hidden markov model. *Journal of Artificial Intelligence Research* 17.

Bui, H. 2003. A general model for online probabilistic plan recognition. In *Proc. of the International Joint Conference on Artificial Intelligence*.

Cielniak, G.; Bennewitz, M.; and Burgard, W. 2003. Where is ...? learning and utilizing motion patterns of persons with mobile robots. In *Proc. of the International Joint Conference on Artificial Intelligence*.

Doucet, A.; de Freitas, J.; Murphy, K.; and Russell, S. 2000. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proc. of the Conference on Uncertainty in Artificial Intelligence*.

Getoor, L.; Friedman, N.; Koller, D.; and Pfeffer, A. 2001. Learning probabilistic relational models. In Dzeroski, S., and Lavrac, N., eds., *Relational Data Mining*. Springer-Verlag.

Guralnik, V., and Srivastava, J. 1999. Event detection from time series data. In *5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Murphy, K. 2002. *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph.D. Dissertation, UC Berkeley, Computer Science Division.

Patterson, D.; Liao, L.; Fox, D.; and Kautz, H. 2003. Inferring high-level behavior from low-level sensors. In *International Conference on Ubiquitous Computing*.

West, M., and Harrison, P. 1997. *Bayesian Forecasting and Dynamic Models*. New York: Springer-Verlag, 2nd edition.