# Building Personal Maps from GPS Data

**Lin Liao** and **Donald J. Patterson** and **Dieter Fox** and **Henry Kautz**
Department of Computer Science & Engineering
University of Washington
Seattle, WA 98195

## Abstract

In this paper we discuss a system that can learn personal maps customized for each user and infer his daily activities and movements from raw GPS data. The system uses discriminative and generative models for different parts of this task. A discriminative relational Markov network is used to extract significant places and label them; a generative dynamic Bayesian network is used to learn transportation routines, and infer goals and potential user errors at real time. In this paper we focus on the basic structures of the models and briefly discuss the inference and learning techniques. Experiments show that our system is able to accurately extract and label places, predict the goals of a person, and recognize situations in which the user makes mistakes (*e.g.*, taking a wrong bus).

## 1 Introduction

A typical map consists of significant places and road networks within a geographic region. In this paper, we present the concept of a *personal map*, which is customized based on an individual's behavior. A personal map includes personally significant places, such as home, a workplace, shopping centers, and meeting places and personally significant routes (*i.e.*, the paths and transportation modes such as foot, car, or bus, that the person usually uses to travel from place to place). In contrast with general maps, a personal map is customized and primarily useful for a given person. Because of the customization, it is well-suited for recognizing an individual's behavior and offering detailed personalized help. For example, in this paper we use a personal map to

- Discriminate a user's activities (is she dining at a restaurant or visiting a friend?);
- Predict a user's future movements and transportation modes, both in the short term (will she turn left at the next street corner? will she get off the bus at the next bus stop?) 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.

We describe a system that builds personal maps automatically from raw location data collected by wearable GPS units. Many potential applications can be built upon the system. A motivating application for this work is the development of personal guidance systems that helps cognitively-impaired individuals move safely and independently throughout their community [Patterson *et al.*, 2004]. Other potential applications include *customized* "just in time" information services (for example, providing the user with current bus schedule information when she is likely to need it or real time traffic conditions on her future trajectories), intelligent user interface (instructing a cell phone not to ring when in a restaurant or at a meeting), and so on.

This paper is focused on the fundamental techniques of learning and inference. We develop probabilistic models that bridge low level sensor measurements (*i.e.*, GPS data) with high level information in the personal maps. Given raw GPS data from a user, our system first finds a user's set of significant places, then a Relational Markov Network (RMN) is constructed to recognize the activities in those places (*e.g.*, working, visiting, and dining out); as discriminative models, RMNs often outperform their corresponding generative models (*e.g.*, HMMs) for classification tasks [Taskar *et al.*, 2002]. The system then uses a dynamic Bayesian network (DBN) model [Murphy, 2002] for learning and inferring transportation routines between the significant places; such a generative model is well-suited for online tracking and real-time user error detection.

The paper is organized as follows. In the next section, we discuss related work. In Section 3, we present the discriminative model for place extraction and labeling, followed by the generative DBN model in Section 4. We show experimental evaluations before concluding.

## 2 Related Work

Over the last years, estimating a person's activities has gained increased interest in the AI, robotics, and ubiquitous computing communities. [Ashbrook and Starner, 2003; Hariharan and Toyama, 2004] 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 Markov models. Their approach then predicts the next goal based on the current and the previous goals. Our system goes beyond their work in many aspects. First, our system not only extracts places, but also recognizes activities associated with those places. Second, their models are 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 contrast, our hierarchical generative model is able to learn more specific motion patterns of transpiration routines, which also enables us to detect user errors.

In the machine learning community, a variety of *relational probabilistic models* were introduced to overcome limitations of propositional probabilistic models. Relational models combine first-order logical languages with probabilistic graphical models. Intuitively, a relational probabilistic model is a *template* for propositional models such as Bayesian networks or MRFs (similar to how first-order logic formulas can be instantiated to propositional logic). Templates are defined over object classes through logical languages such as Horn clauses, frame systems, SQL, and full first-order logic. Given data, these templates are then *instantiated* to generate propositional models (typically Bayesian networks or MRFs), on which inference and learning is performed. Relational probabilistic models use high level languages for describing systems involving complex relations and uncertainties. Since the structures and parameters are defined at the level of classes, they are shared by the instantiated networks. Parameter sharing is particularly essential for learning from sparse training data and for knowledge transfer between different people. As a popular relational probabilistic model, Relational Markov Networks (RMN) define the templates using SQL, a widely used query language for database systems, and the templates are instantiated into (conditional) Markov networks, which are *undirected* models that do not suffer the cyclicity problem and are thereby more flexible and convenient. Since their introduction, RMNs have been used successfully in a number of domains, including web page classification [Taskar *et al.*, 2002], link prediction [Taskar *et al.*, 2003], and information extraction [Bunescu and Mooney, 2004].

In the context of probabilistic plan recognition, [Bui *et al.*, 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 parameters of the hierarchical activity model, and their domains, from data. Furthermore, our low level estimation problem is more challenging than their indoor tracking problem.

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

# 3 Extracting and Labeling Places

In this section, we briefly discuss place extraction and activity labeling. For full technical details of the activity labeling refer to [Liao *et al.*, 2005].

## 3.1 Place Extraction

Similar to [Ashbrook and Starner, 2003; Hariharan and Toyama, 2004], our current system considers significant places to be those locations where a person typically spends extended periods of time. From the GPS data, it first looks for locations where the person stays for a given amount of time (*e.g.*, 10 minutes), and then these locations are clustered to merge spatially similar points. An extension of the approach that takes into account more complex features is discussed in future work (Section 6).

## 3.2 Activity Labeling

We build our activity model based on the Relational Markov Network (RMN) framework [Taskar *et al.*, 2002]. RMNs describe specific relations between objects using clique templates specified by SQL queries: each query $C$ selects the relevant objects and their attributes, and specifies a *potential* function, or clique potential, $\phi_C$, on the possible values of these attributes. Intuitively, the clique potentials measure the "compatibility" between values of the attributes. Clique potentials are usually defined as a log-linear combinations of *feature* functions, *i.e.*, $\phi_C(\mathbf{v}_C) = \exp\{\mathbf{w}_C^T \cdot \mathbf{f}_C(\mathbf{v}_C)\}$, where $\mathbf{v}_C$ are the attributes selected in the query, $\mathbf{f}_C()$ is a feature vector for $C$, and $\mathbf{w}_C^T$ is the transpose of the corresponding weight vector. For instance, a feature could be the number of different homes defined using aggregations.

To perform inference, an RMN is *unrolled* into a Markov network, in which the nodes correspond to the attributes of objects. The connections among the nodes are built by applying the SQL templates to the data; each template $C$ can result in several cliques, which share the same feature weights. Standard inference algorithms, such as belief propagation and MCMC, can be used to estimate the conditional distribution of hidden variables given all the observations.

**Relational activity model**

Because behavior patterns can be highly variable, a reliable discrimination between activities must take several sources of evidence into account. More specifically, our model defines the following templates:

1. *Temporal* patterns: Different activities often have different temporal patterns, such as their duration or the time of day. Such local patterns are modeled by clique templates that connect each attribute with the activity label.
2. *Geographic* evidence: Information about the types of businesses close to a location can be extremely useful to determine a user's activity. Such information can be extracted from geographic databases like

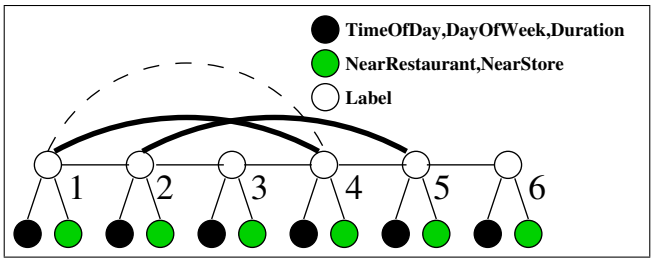


Figure 1: An example of unrolled Markov network with six activities. Solid straight lines indicate the cliques generated by the templates of temporal, geographic, and transition features; bold solid curves represent spatial constraints (activity 1 and 4 are associated with the same place and so are 2 and 5); dashed curves stand for global features, which are label-specific cliques (activity 1 and 4 are both labeled as 'AtHome' or 'AtWork' at this moment).

`http://www.microsoft.com/mappoint`. Since location information in such databases is not accurate enough, we consider such information by checking whether, for example, a restaurant is within a certain range from the location.

3. *Transition* relations: First-order transitions between activities can also be informative. For example, going from home to work is very common while dining out twice in a row is rare.
4. *Spatial* constraints: Activities at the same place are often similar. In other words, the number of different types of activities in a place is often limited.
5. *Global* features: These are soft constraints on the activities of a person. The number of different home locations is an example of the global constraints. Such a constraint is modeled by a clique template that selects all places labeled as home and returns how many of them are different.

**Inference**

In our application, the task of inference is to estimate the labels of activities in the *unrolled* Markov networks (see Fig. 1 for an example). Inference in our relational activity models is complicated by the fact that the structure of the unrolled Markov network can change during inference. This is due to the fact that, in the templates of global features, the label of an object determines to which cliques it belongs. We call such cliques *label-specific cliques*. Because the label values are hidden during inference, such cliques potentially involve all the labels, which makes exact inference intractable.

We perform approximate inference using Markov Chain Monte Carlo (MCMC) [Gilks *et al.*, 1996]. We first implemented MCMC using basic Gibbs sampling. Unfortunately, this technique performs poorly in our model because of the strong dependencies among labels. To make MCMC mix faster, we developed a mixture of two transition kernels: the first is a block Gibbs sampler and the second is a Metropolis sampler (see [Liao *et al.*, 2005] for details). The numbers of different homes and workplaces are stored in the chains as global variables. This allows us to compute the *global* features *locally* in both kernels. In order to determine which kernel to use at each step, we sample a random number $u$ between 0 and 1 uniformly, and compare $u$ with the given threshold $\gamma$ ($\gamma = 0.5$ in our experiments).

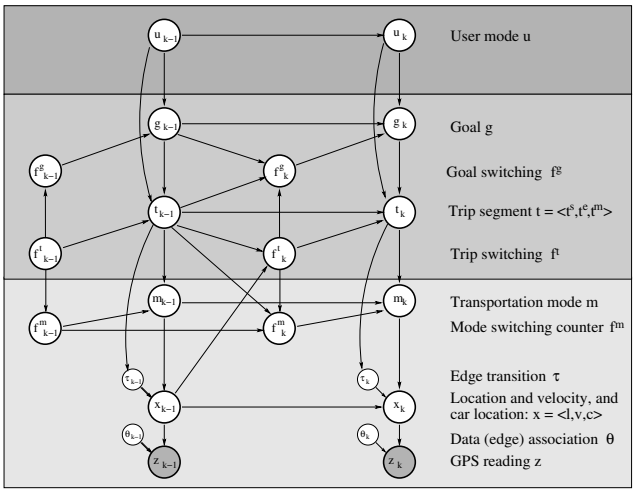


Figure 2: Hierarchical activity model representing a person's outdoor movements during everyday activities. The upper level estimates the user mode, the middle layer represents goals and trip segments, and the lowest layer is the flat model, estimating the person's location, velocity, and transportation mode.

**Learning**

The parameters to be learned are the weights $\mathbf{w}$ of the features that define the clique potentials. To avoid overfitting, we perform maximum a posterior (MAP) parameter estimation and impose an independent Gaussian prior with constant variance for each component of $\mathbf{w}$. Since the objective function for MAP estimation is convex, the global optimum can be found using standard numerical optimization algorithms [Taskar *et al.*, 2002]. We apply the quasi-Newton methods to find the optimal weights because they have been found to be very efficient for CRFs [Sha and Pereira, 2003]. Each iteration of this technique requires the value and gradient of the objective function computed at the weights returned in the previous iteration. In [Liao *et al.*, 2005], we presented an algorithm that *simultaneously* estimates at each iteration the value and its gradient using MCMC.

Although parameter learning in RMNs requires manually labeled training data, parameter sharing makes it easy to transfer knowledge. For example, in our system, we can learn a *generic* model from people who have manually labeled data, and then apply the model to people who have no labeled data. Generic models in our system can perform reasonably well, as we will show in the experiments.

## 4 Learning and Inferring Transportation Routines

We estimate a person's activities using the three level dynamic Bayesian network model shown in Fig. 2. 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 indicates the user

mode, which could be typical behavior, user error, or deliberate novel behavior. The middle level represents the person's goal (*i.e.*, next significant place) and trip segment ( defined below). The lowest level is the *flat* model, which estimates the person's transportation mode, location and motion velocity from the GPS sensor measurements. In this section, we explain the model from bottom up; refer to [Liao *et al.*, 2004] for more details.

### 4.1 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). In our DBN model, 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 $\theta_k$. 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, $\tau_k$. Vertex transitions $\tau$ model the decision a person makes when moving over a vertex in the graph, for example, to turn right when crossing a street intersection.

The mode of transportation can take on four different values $m_k \in \{\text{BUS, FOOT, CAR, BUILDING}\}$. 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.

An efficient algorithm based on Rao-Blackwellised particle filters (RBPFs) [Doucet *et al.*, 2000] has been developed to perform online inference for the flat model. In a nutshell, the RBPF samples transportation mode ${m_k}^{(i)}$, transportation mode switch $f_k^{m(i)}$, data association ${\theta_k}^{(i)}$, edge transition ${\tau_k}^{(i)}$, and velocity ${v_k}^{(i)}$, then it updates the Gaussian distribution of location ${l_k}^{(i)}$ using a one-dimensional Kalman filter. 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].

We apply expectation maximization (EM) to learn the model parameters. Before learning, the model has no preference for when a person switches mode of transportation, or which edge a person transits to when crossing a vertex on the graph. However, information about bus routes, and the fact that the car is either parked or moves with the person, already provide important constraints on mode transitions. At each iteration of EM, the location, velocity, and mode of transportation are estimated using the Rao-Blackwellised particle filter of the flat model. In the E-step, transition counts of a forward and a backward filtering pass through the data log are combined, based on which we update the model parameters in the M-step. In our Rao-Blackwellised model, edge transitions are counted whenever the *mean* of a Kalman filter transits the edge. The learned flat model encodes information about typical motion patterns and significant locations by edge and mode transition probabilities.

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

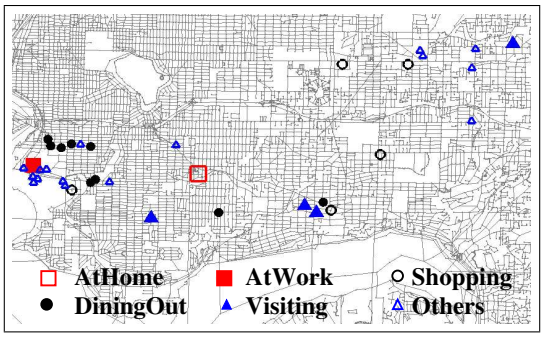

Figure 3: Part of the locations contained in the data set of a single person, collected over a period of four months ($x$-axis is 8 miles long).

After we estimate the mode transition probabilities for each edge, we find *mode transfer locations*, *i.e.*, usual bus stops and parking lots, by looking for those locations at which the mode switching exceeds a certain threshold.

### 4.2 Goals and 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 $\tau_k$.

A goal represents the current target location of the person. Goals include the significant locations extracted using our discriminative model. The transfer between trip segments and goals is handled by the boolean switching nodes $f_k^t$ and $f_k^g$, respectively.

To 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.

Because we have learned the set of goals using the discriminative model and the set of trip segments using the flat model, we only need to estimate the transition matrices at all levels: between the goals, between the trip segments given the goal, and between the adjacent streets given the trip segment.

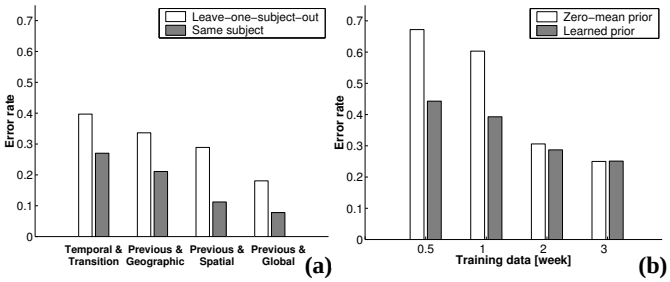


Figure 4: (a) Error rates of cross-validation of the generic models and customized models using different feature sets; (b) Zero-mean prior vs. learned model as prior mean (shown the error rates over the new places only).

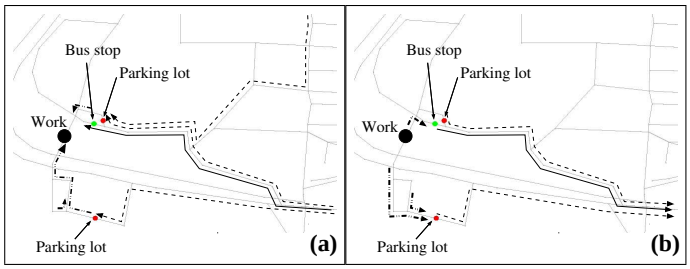


Figure 5: Learned model zoomed into the area around the work place and the very likely transitions (probability above 0.75). Dashed lines indicate car mode, solid lines bus, and dashed-dotted lines foot. (a) Given that the goal is the work place. (b) Given that the goal is home.

Again, we use EM in the hierarchical model, which is similar to that in the flat model. 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.

### 4.3 User Modes

To detect user errors or novel behavior, we add the variable $u_k$ to the highest level, which indicates the user's behavior mode $\in \{Normal, Novel, Erroneous\}$. Different values of $u_k$ instantiate different parameters for the lower part of the model. When user mode is typical behavior, the model is instantiated using the parameters learned from training data. When a user's behavior is $Erroneous$, the goal remains the same, but the trip segment is set to a distinguished value "unknown" and as a consequence the parameters of the flat model (*i.e.*, transportation mode transitions and edge transitions) are switched to their a priori values: An "unknown" trip segment cannot provide any information for the low level parameters. When a user's behavior is $Novel$, the goal is set to "unknown," the trip segment is set to "unknown," and the parameters of the flat model are again set to their a priori values.

To infer the distribution of $u_k$, we run two trackers simultaneously and at each time their relative likelihood is used to update the distribution. The first tracker uses the hierarchical model with learned parameters and second tracker uses the flat model with a priori parameters. When a user is following her ordinary routine, the first tracker has higher likelihoods, but when the user makes error or does something novel, the second tracker becomes more likely. Unless the true goal is observed, the system cannot distinguish errors from novel behavior, so the precise ratio between the two is determined by hand selected prior probabilities. In some situations, however, the system knows where the user is going, *e.g.*, if the user asks for directions to a destination, or if a caregiver indicates the "correct" destination, and thus the goal is fixed, treated as an observed, and therefore *clamped*. After we have clamped the goal, the probability of novel behavior becomes zero and the second tracker just determines the probabilities of an error.

## 5 Experiments

To evaluate our system, we collected two sets of location data using wearable GPS units. The first data set contains location traces from a single person over a time period of four months (see Fig. 3). It includes about 400 activities at 50 different places. The second data set consists of one-week of data from five different people. Each person has 25 to 30 activities and 10 to 15 different significant places. We extracted from the logs each instance of a subject spending more than 10 minutes at one place. Each instance corresponds to an activity. We then clustered the nearby activity locations into places.

### 5.1 Evaluating the RMN Model

For training and evaluation, the subjects manually labeled the data with their activities from the following set: {**AtHome, AtWork, Shopping, DiningOut, Visiting, Other**}. Then, we constructed the unrolled Markov networks using the templates described above, trained the models, and tested their accuracy. Accuracy was determined by the activities for which the most likely labeling was correct.

In practice, it is of great value to learn a *generic* activity model that can be immediately applied to new users without additional training. In the first experiment, we used the data set of multiple users and performed leave-one-subject-out cross-validation: we trained using data from four subjects, and tested on the remaining one. The average error rates are indicated by the white bars in Fig. 4(a). By using all the features, the generic model achieved an error rate of $20\%$. Note that the global features and the spatial constraints are very useful. To gauge the impact of different habits on the results, we also performed the same evaluation using the data set of single subject. In this case, we used one-month data for training and the other three-month data for test, and we repeated the validation process for each month. The results are shown by the gray bars in Fig. 4(a). In this case, the model achieved an error rate of only $7\%$. This experiment shows that it is possible to learn good activity models from groups of people. It also shows that if the model is learned from more "similar" people, then higher accuracy can be achieved. This indicates that models can be improved by grouping people based on their activity patterns.

When estimating the weights of RMNs, a prior is imposed in order to avoid overfitting. Without additional information, a zero mean Gaussian is typically used as the prior [Taskar *et al.*, 2002]. Here we show that performance can also be improved by estimating the *hyper-parameters* for the means of the weights using data collected from other people. Similar to the first experiment, we want to learn a customized model

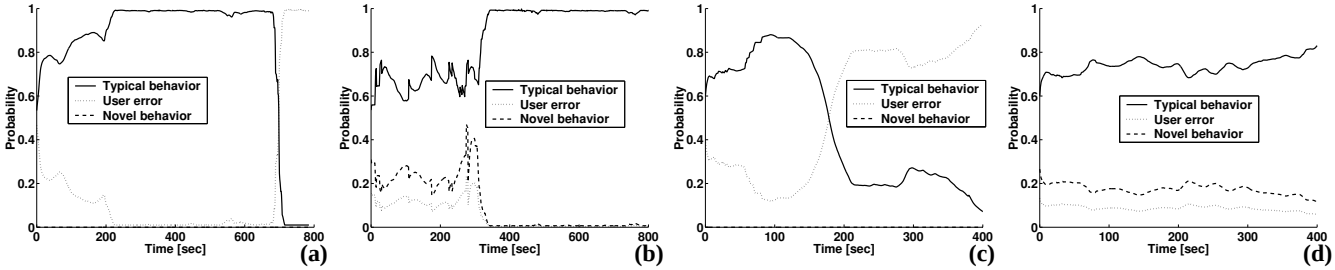


Figure 6: The probabilities of user mode in two experiments (when goal is unclamped, the prior ratio of typical behavior, user error and novel behavior is 3:1:2; when goal is clamped, the probabilities of novel behavior are always zero): (a) Bus experiment with goal clamped; (b) Bus experiment with goal unclamped; (c) Foot experiment with goal clamped; (d) Foot experiment with goal unclamped.

for a person **A**, but this time we also have labeled data from others. We could simply ignore the others' data and use the labeled data from **A** with a zero-mean prior. Or we can first learn the weights from the other people and use that as the mean of the Gaussian prior for **A**. We evaluate the performance of the two approaches for different amounts of training data from person **A**. The results are shown in Fig. 4(b). We can see that using data from others to generate a prior boosts the accuracy significantly, especially when only small amounts of training data are available.

Using the Bayesian prior smoothly shifts from generic to customized models: on one end, when no data from the given subject is available, the approach returns the generic (prior) model; on the other end, as more labeled data become available, the model adjusts more and more to the specific patterns of the user and we get a customized model.

### 5.2 Evaluating the DBN Model

The learning of the generative model was done completely unsupervised without any manual labeling. Fig. 5 show the learned trip segments and street transitions zoomed into the workplace. 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.

As we described, an important feature of our model is the capability to capture user errors and novel behavior using a parallel tracking approach. To demonstrate the performance of this technique, we did the following two experiments:

In the first experiment, a user took the wrong bus home. For the first 700 seconds, the wrong bus route coincided with the correct one and the system believed that the user was in $\{u_k = Normal\}$ mode. But when the bus took a turn that the user had never taken to get home, the probability of errors in the clamped model dramatically jumped (see Fig. 6(a)). In contrast, the unclamped model cannot determine a user error because the user, while on the wrong bus route, was on a bus route consistent with other previous goals (see Fig. 6(b)).

The second experiment was a walking experiment in which the user left his office and proceeded to walk away from his normal parking spot. When the destination was not specified, the tracker had a steady level of confidence in the user's path (see Fig. 6(d), there are lots of previously observed paths from his office), but when the goal was specified, the system initially saw behavior consistent with walking toward the parking spot, and then as the user turned away at time 125, the tracker's confidence in the user's success dropped (see Fig. 6(c)).

## 6 Conclusions and Future Work

In this paper we have described a system that can build personal maps automatically from GPS sensors. More specifically, the system is able to: recognize significant locations of a user and activities associated with those places, infer transportation modes and goals, and detect user errors or novel behavior. The system uses a Relational Markov Network for place classification and a hierarchical Dynamic Bayesian Network for online tracking and error detection. This technique has been used as the basis for both experimentation and for real context-aware applications including an automated transportation routing system that ensures the efficiency, safety, and independence of individuals with mild cognitive disabilities (see [Patterson *et al.*, 2004]).

In our future work we plan to improve the place extraction. The current approach only relies on measuring the time periods a person stays at each place and uses a fixed threshold to distinguish significant places from insignificant ones. However, it is hard to find a fixed threshold that works for all significant places. If we set the threshold too big (say 10 minutes, as in our experiments), some places could be missed (*e.g.*, places a user stops by to get coffee or pick up his kids); if we set the value too small (*e.g.*, 1 minute), some trivial places (such as traffic lights) may be considered significant. Therefore, to extract more places accurately, we will take into account more features besides stay duration. For example, transportation mode is a very useful indicator: if a user switches to *foot* at some place during a *car* trip, that place is likely to be significant. Since transportation mode itself has to be inferred, we must design a model that considers all these uncertainties comprehensively. In order to do that, we plan to extend the existing relational probabilistic languages so that we can model complex relations and still perform efficient inference and learning.

## Acknowledgments

This research is supported in part by NSF under grant number IIS-0433637 and SRI International subcontract 03-000225 under DARPA's CALO project. We also thank anonymous reviewers for their helpful comments.

## References


[Ashbrook and Starner, 2003] D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. In *Personal and Ubiquitous Computing*, 2003.

[Bui *et al.*, 2002] Hung H. Bui, Svetha Venkatesh, and Geoff West. Policy recognition in the abstract hidden markov models. *Journal of Artificial Intelligence Research*, 2002.

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

[Bunescu and Mooney, 2004] R. Bunescu and R. J. Mooney. Collective information extraction with relational markov networks. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004.

[Doucet *et al.*, 2000] Arnaud Doucet, Nando de Freitas, Kevin Murphy, and Stuart Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2000.

[Gilks *et al.*, 1996] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, 1996.

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

[Hariharan and Toyama, 2004] R. Hariharan and K. Toyama. Project Lachesis: parsing and modeling location histories. In *Geographic Information Science*, 2004.

[Liao *et al.*, 2004] L. Liao, D. Fox, and H. Kautz. Learning and inferring transportation routines. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2004.

[Liao *et al.*, 2005] Lin Liao, Dieter Fox, and Henry Kautz. Location-based activity recognition using relational Markov networks. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

[Murphy, 2002] Kevin Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, 2002.

[Patterson *et al.*, 2003] Donald J. Patterson, Lin Liao, Dieter Fox, and Henry Kautz. Inferring high-level behavior from low-level sensors. In *International Conference on Ubiquitous Computing (UbiComp)*, 2003.

[Patterson *et al.*, 2004] Donald J. Patterson, Lin Liao, Krzysztof Gajos, Michael Collier, Nik Livic, Katherine Olson, Shiaokai Wang, Dieter Fox, and Henry Kautz. Opportunity Knocks: a System to Provide Cognitive Assistance with Transportation Services. In Itiro Siio Nigel Davies, Elizabeth Mynatt, editor, *Proceedings of UBICOMP 2004: The Sixth International Conference on Ubiquitous Computing*, volume LNCS 3205, pages 433–450. Springer-Verlag, October 2004.

[Sha and Pereira, 2003] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of Human Language Technology-NAACL*, 2003.

[Taskar *et al.*, 2002] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2002.

[Taskar *et al.*, 2003] B. Taskar, M. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.