Extracting Places and Activities from GPS Traces

Lin Liao, Dieter Fox, and Henry Kautz

University of Washington, Dept. of Computer Science & Engineering, Seattle, WA

Abstract. Learning patterns of human behavior from sensor data is extremely important for high-level activity inference. We show how to extract and label a person’s activities and significant places from traces of GPS data. In contrast to existing techniques, our approach simultaneously detect and classify the significant locations of a person and takes the high-level context into account. Our system uses relational Markov networks to represent the hierarchical activity model that encodes the complex relations among GPS readings, activities and significant places. We apply FFT-based message passing to perform efficient summation over large numbers of nodes in the networks. We present experiments that show significant improvements over existing techniques.

1 Introduction

The problem of learning patterns of human behavior from sensor data arises in many areas and applications of computer science, including intelligent environments, surveillance, and assistive technology for the disabled. A focus of recent interest is the use of data from wearable sensors, and in particular, GPS (global positioning system) location data, to learn to recognize the high-level activities in which a person is engaged over a period of many weeks, and to further determine the relationship between activities and locations that are important to the user [1,8,10,4]. The goal is to segment the user’s day into everyday activities (such as “working”, “visiting”, “travel”) and to recognize and label significant locations that are associated with one or more activity (such as “workplace”, “friend’s house”, “user’s bus stop”). Such activity logs can be used as an automated personal diary, or collected from a group of users for large-scale studies of human behavior across time and space for disciplines such as urban studies and sociology [3]. However, previous approaches to automated activity and place labeling suffer from design decisions that restrict their accuracy and flexibility:

First, previous work decoupled the subproblem of determining whether or not a geographic location is significant and should be assigned a label, from that of labeling places and activities. The first problem was handled by simply assuming that a location is significant if and only if the user spends at least \( N \) minutes there, for some fixed threshold \( N \) [1,8,10,4]. Some way of restricting the enormous set of all locations recorded for the user to a meaningful subset is clearly necessary. However, in practice, any fixed threshold leads to many errors. Some significant locations, for example, the place where the user drops off his children at school, may be visited only briefly, and so would be excluded by a high threshold. A lower threshold, however, would include too many insignificant locations, for example, a place where the user briefly waited at a traffic light. The inevitable errors cannot be
resolved because information cannot flow from the label assignment process back to the one that determines the domain to be labeled.

Second, concerns for computational efficiency prevented previous approaches from tackling the problem of activity and place labeling in full generality. [1] does not distinguish between places and activities; although [10] does, the implementation limited places to a single activity. Neither approaches model or label the user’s activities when moving between places. [8] and [4] learn transportation patterns, but not place labels.

The third problem is one of the underlying causes of the other limitations. The representations and algorithms used in previous work make it difficult to learn and reason with the kinds of non-local features that are useful in disambiguating human activity. For a simple example, if a system could learn that a person rarely went to a restaurant more than once a day, then it could correctly give a low probability to an interpretation of a day’s data under which the user went to three restaurants. Our previous work [10] used clique templates in relational Markov networks for concisely expressing global features, but the MCMC inference algorithm we used made it costly to reason with aggregate features, such as statistics on the number of times a given activity occurs. The ability to efficiently leverage global features of the data stream could enhance the scope and accuracy of activity recognition.

This paper presents a unified approach to automated activity and place labeling which overcomes these limitations. Key features of our system are:

- It simultaneously solves the tasks of identifying significant locations and labeling both places and activities from raw GPS data, all in a conditionally trained relational Markov network. No arbitrary thresholds regarding the time spent at a location or the number of significant places are employed.
- It creates a complete interpretation of the log of a user’s data, including transportation activities as well as activities performed at particular places. It allows different kinds of activities to be performed at the same location, and vice-versa.
- Efficient inference is performed using belief propagation with Fast Fourier Transform for aggregations, and parameter learning is done efficiently using pseudo-likelihood.

This paper is organized as follows. We begin with a discussion of relational Markov networks (RMN). Then we explain how to apply RMNs to the problem of location-based activity recognition. Finally, we present experimental results on real-world data that demonstrate significant improvement in coverage and accuracy over previous work.

2 Relational Markov Networks for Activity Recognition

2.1 Relational Markov Networks

Our goal is to develop a probabilistic model that can be used to classify sequences of activities from GPS readings. One possible approach is to use Hidden Markov Models (HMM). However, discriminative models, especially Conditional Random Fields (CRF), have been shown to outperform generative approaches such as HMMs.
and Markov random fields in areas such as natural language processing [7] and computer vision [6]. CRFs are undirected graphical models that were developed for labeling sequence data [7]. Unlike HMMs, CRFs directly represent the conditional distribution \( p(y|x) \) over labels \( y \) (activities) given the observed attributes \( x \) (GPS readings). Therefore, they do not assume the attributes are independent given the labels, and thereby are especially suitable for classification tasks with complex and overlapped attributes.

Relational Markov networks (RMN) [13] extend CRFs by providing a relational language for describing clique structures and enforcing parameter sharing at the template level. Thereby RMNs provide an extremely flexible and concise framework for defining features that can be used in the activity recognition context. A key concept of RMNs are relational clique templates, which specify the structure of a CRFs in a concise way. In a nutshell, a clique template \( C \in C \) is similar to a database query (e.g., SQL) in that it selects tuples of nodes from a CRF and connects them into cliques. Each clique template \( C \) is additionally associated with a potential function \( \phi_C(v_C) \) that maps values of variables to a non-negative real number. Using a log-linear combination of feature functions, we use the following clique potentials:

\[
\phi_C(v_C) = \exp\{w^T_C \cdot f_C(v_C)\},
\]

where \( f_C() \) defines a feature vector for clique \( C \) and \( w^T_C \) is the transpose of the corresponding weight vector.

To perform inference and estimate parameters, an RMN is unrolled into a CRF, in which the nodes corresponds to the attributes and labels. The connections among the nodes are built by applying the clique templates to the data; each template \( C \) can result in several cliques which have identical structure and share the same weights \( w_C \). The resulting CRF factorizes the conditional distribution as

\[
p(y \mid x) = \frac{1}{Z(x)} \prod_{C \in C} \prod_{v_C \in C} \exp\{w^T_C \cdot f_C(v_C)\},
\]

where \( Z(x) = \sum_{y'} \prod_{C \in C} \prod_{v_C \in C} \phi_C(v'_C) \) is the normalizing partition function. In essence, (2) multiplies the potentials of all cliques generated by the clique templates \( C \). The partition function requires summation over all possible labels, which makes exact inference intractable for all but the most simple networks. Before we describe how to perform approximate inference and learning in RMNs, we will first describe how RMNs can be used to model location-based activities.

### 2.2 Hierarchical Activity Model

There are three types of basic objects in our activity model: GPS reading, activity, and significant place. Their relations are explained in Figure 1.

**GPS traces** indicate a person’s location at each time. To bridge the gap between the raw measurements and the high level activities, we first associated all GPS readings to the street map, shown at the lowest level of the hierarchical activity model. To perform this association, we discretize the edges of the street map graph
Fig. 1. An example of the hierarchical model for location-based activity recognition. Per day, the lowest level typically consists of several thousand GPS nodes, the next level contains around one thousand activity (grid cell) nodes, and the place level contains around five places.

into grid cells of 10m length. In order to reason about motion direction, grid cells have two different orientations corresponding to moving up or down a street.

Activities are estimated at the granularity of grid cells in the street map. Hence, our model labels a person’s activity whenever she passes through or stays at a grid cell. There are two main groups of activities, navigation activities and location activities. Activities related to navigation are walking, driving car, riding bus, getting on/off a bus, or getting in/out of a car. Location activities are typically performed while staying at a location, such as work, leisure, sleep, visit, drop off / pickup, or other. To perform this reasoning, we combine consecutive GPS readings that are associated to the same grid cell into an activity object, as shown in Figure 1 (the second level from the bottom). Note that the same grid location in the street map can be associated with multiple activities occurring at different times (typical combinations are work and drop off or riding bus and driving car).

Significant places are those locations that play a significant role in the activities of a person. Such places include a person’s home and work place, the bus stops and parking lots the person typically uses, the homes of friends, and stores the person frequently shops in. Since different activities can occur at the same grid cell in the map, we also allow different activities to occur at the same significant place. Furthermore, due to signal loss and noise in the GPS readings, the same significant place can comprise multiple, different cells in the street map.

2.3 Relational Markov Network for Activity Recognition

Key components of the RMN activity model are the templates generating the cliques of the Markov network and the feature functions defining the clique potentials.

Associating GPS traces to the street map: The low level inference task is to associate the raw GPS traces to grid cells in the street map. The main difficulty of this task is due to inconsistencies between GPS traces and the street locations in the map, as shown in Figure 2(a); simply projecting each reading to the nearest street will not perform well. As illustrated in Figure 1, the RMN contains a so-called GPS node for each GPS reading. The observed values of each such node are the long/lat coordinate of the GPS reading, and the hidden state of the node is the index
of the grid cell it corresponds to. To determine this index, the model takes three different spatial features into account.

- GPS noise and map uncertainty are considered by a feature that measures the distance between the GPS measurement and the center of the grid cell it is associated with.
- To ensure consistency between the GPS sequence and the corresponding sequence of grid cells in the street map, each GPS node is connected to its immediate predecessor and successor. For each connected pair of GPS nodes, the feature $f_{\text{seq}}$ measures how consistent the distances between the GPS readings and the associated grid cells are:

$$f_{\text{seq}}(n_i, n_j) = ||n_i^{\text{gps}} - n_j^{\text{gps}}|| - ||n_i^{\text{cell}} - n_j^{\text{cell}}||,$$

where $n_i$ and $n_j$ are neighbors in the GPS trace, and $n_i^{\text{gps}}$ and $n_i^{\text{cell}}$ are a node’s GPS measurement and the location of the grid cell it is associated to, respectively.
- Additional features measure whether the grid cells of connected GPS nodes are on the same street and points in the same direction.

During inference, the RMN computes (2) using these features to estimate distributions over associations between GPS readings and grid cells. Figure 2(a) illustrates how this layer of the model corrects for GPS error.

**Activities:** Once the association between GPS readings and grid cells is performed, the GPS trace can be segmented by grouping together consecutive GPS readings that are associated to the same grid cell. Each group generates a new node in the activity model (second level in Figure 1). Our model then estimates the activity at each node on this segmented trace. Each activity node has various observed features, summarizing information about the stay at this location. These features include:

- Temporal information such as time of day, day of week, and duration on the grid cell, similar to [10].
- Average speed through a grid cell, which is important for discriminating different transportation modes.
- Information extracted from geographic databases, such as whether a grid cell is on a bus route and whether it is close to a bus stop.
- Additionally, each node is connected to its neighbors. The features at these clique potentials measure compatibility between types of activities at neighboring nodes in the trace. For instance, it is extremely unlikely to get on the bus at one location and drive a car at the neighboring location right afterwards.

**Significant places:** Our model aims at determining those places that play a significant role in the activities of a person. To do so, we proceed as follows. First, the activities inferred at the nodes of the second level in our model are classified into whether or not they belong to a significant place. For instance, while walking,
driving car, or riding bus are not associated to significant places, working or getting on or off the bus indicate a significant place. Then, during inference, all locations at which a significant activity is more likely than a non-significant one generate a place (third level in Figure 1). All grid cells generated by nearby activities are merged into the same place. The type of a place can be workplace, home, friend’s home, shop, bus stop, parking lot, or other. Once a place is determined, our RMN model generates a corresponding node. In order to infer the type of a place, the model considers the frequency of the different activities occurring at that place. This is done by connecting each place node to all grid cells associated to it. Then, the model counts the frequency of the different activities occurring at that place. This frequency takes into account the uncertainty in the activities occurring at the individual places.

**Soft constraints:** At the highest level of the model, two additional summation cliques count the number of different homes and work places, given the current labeling of the significant places. These counts provide soft constraints that bias the system to generate interpretations that result in reasonable numbers of different homes and work places.

**2.4 Inference**

In our application, the task of inference is to estimate the associations between GPS readings and the street map, the activities occurring at each grid cell, and the types of significant places indicated by the activities.

**Segmenting GPS traces into grid cells:** In the first step, our RMN generates the sequence of GPS readings (observations) and their corresponding grid cell indices (hidden nodes). To make the inference tractable, we only consider the grid cells within a certain range from each reading (say 30 meters). The clique potentials are computed as in (1) using the features described in Section 2.3. To infer the most likely sequence of grid cells, we use max-product belief propagation, similar to the Viterbi algorithm in Hidden Markov Models.

**Simultaneous extraction and labeling of activities and places:** The most likely sequence of grid cells from the previous step is used to generate the activity nodes (one activity node per grid cell) and their feature values. We then run sum-product belief propagation to infer the posterior distribution of activity labels. Based on the activities estimated at each grid cell, significant places at the third level of the model are generated: only activities that are more likely to be significant can generate significant places. These nodes are connected to all grid cells in their vicinity. Finally, the two nodes estimating the number of homes and workplaces are generated at the top level of the model. Once all these nodes, their connections, and the features are determined, a final round of belief propagation is performed, now re-estimating distributions over activities at the grid cells and over types of the significant places (note that the lowest level GPS associations are not re-estimated).
Standard belief propagation is very slow in this model because big cliques are generated when we connect each place with its corresponding activities and when we estimate the number of homes and workplaces. Therefore we use Fast Fourier Transform (FFT) so as to efficiently perform the summations required by this estimation (see [9] for details). Note that the high level place labels will affect the labels of lower level activities by propagating the place type information downward. This downward information passing can be done efficiently using the bidirectional summation cliques. As a result, activities at the same place will be labeled in a more consistent way. And more interestingly, new significant activities can be discovered because of the information passed from places.

2.5 Parameter Learning

The goal of parameter learning is to learn the weights of the features used in (2). To do so, we manually label GPS traces with activities and significant locations. Typically, the weights \( w \) are learned discriminatively by maximizing the log-likelihood of the labeled training data [13,10]:

\[
L(w) \equiv \log p(y \mid x, w) - \frac{w^T w}{2\sigma^2}.
\]  

(3)

The rightmost term in (3) is added to avoid overfitting. It imposes a zero-mean, Gaussian shrinkage prior with variance \( \sigma^2 \) on each component of the weight vector [13]. Since (3) can be shown to be concave, the global optimum of \( L \) can be found using modern numerical optimization algorithms, such as conjugate gradient or quasi-Newton techniques [12]. However, maximizing the conditional likelihood requires running the inference procedure at each iteration of the optimization, which can be very expensive. An alternative is to maximize the pseudo-likelihood of the training data [2]:

\[
L(w) \approx \sum_{i=1}^{n} \log p(y_i \mid MB(y_i), w) - \frac{w^T w}{2\sigma^2}
\]  

(4)

where \( MB(y_i) \) is the Markov Blanket of variable \( y_i \). This approximate technique can be evaluated extremely efficiently and has been shown to perform well in several domains [6,11]. In the context of place labeling, we showed how to use non-zero mean priors in order to transfer weights learned for one person to another person [10]. In our experiments, learning the weights using pseudo-log-likelihood is very efficient and performs well in our tests.

3 Experimental Results

In our experiments, we want to answer the following three questions: 1) How well can our system extract significant places from raw data? 2) How accurate can our system label the places and activities, and especially how much do aggregation features help? 3) How efficient is inference in our framework? To do so, we collected GPS data traces from a user over four weeks: two weeks for training and the other
two weeks for testing. We manually labeled all activities for supervised training and evaluation. Learning from two weeks of training data took less than three minutes using pseudolikelihood. Inference took approximately ten minutes per two weeks of data (> 50,000 GPS readings).

3.1 Extracting significant places

Our test data contained 19 true significant places. We compare our model with the widely-used approach that uses a single time threshold to determine whether or not a location is significant [1,5,8,10,4]. We use four different thresholds from 1 minute to 10 minutes, and we measure the false positive and false negative locations extracted from the GPS traces. As shown in Fig. 2(b), any fixed threshold is not satisfactory: low thresholds have many false negatives, and high thresholds result in many false positives. In contrast, our model performs much better: it only generates 2 false positives and 1 false negative. This experiment shows that using high-level context information drastically improves the extraction of significant places.

3.2 Labeling places and activities

A key feature of our system is that it explicitly models the relationships between places and activities. The labels of activities generate instances of places, which then help to better estimate the activities occurring in their spatial area. The test data contained a total of 70 activities at 19 different places. We measure the performance of activity detection with and without the use of summation cliques (that is, without the upper two levels of the model in Figure 1). The results are summarized in Table 1. The first two columns measure the quality of detecting activities (that is, detecting that an activity occurs at a specific place). The last two columns measure how often the detected activities are labeled correctly. From the numbers we can see that even without the upper two levels, our system performs reasonably well. However, adding summation cliques boosts the accuracy significantly. It is interesting to note that adding places increases the number of detected activities at the lower level of the model (less false negatives). For example, in our experiment, we observed that
the low level inference can detect “get off car” reliably, but sometimes misses “get into car” at the same place; however, by inferring that the place is a parking area, the system is able to catch some of the missed “get into car” activities. In addition to improved accuracy on activity detection, our system correctly classified 18 out of 19 significant places.

<table>
<thead>
<tr>
<th>Model</th>
<th>false negative</th>
<th>false positive</th>
<th>Number of activities correctly extracted correctly labeled</th>
<th>mis-labeled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without summ. cliques</td>
<td>13</td>
<td>3</td>
<td>44</td>
<td>13</td>
</tr>
<tr>
<td>With summ. cliques</td>
<td>9</td>
<td>2</td>
<td>53</td>
<td>8</td>
</tr>
</tbody>
</table>

**Table 1.** Performance of activity labeling

The inference performed by our system can also be used to automatically generate a log of a person’s daily life. For example, Table 2 provides a textual summary of the activities inferred by our system on a typical day (this summary is extracted from the most likely state sequence).

<table>
<thead>
<tr>
<th>Time</th>
<th>Activity and transportation</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:15am - 8:34am</td>
<td>Drive from home 1 to parking lot 2, walk to workplace 1;</td>
</tr>
<tr>
<td>8:34am - 5:44pm</td>
<td>Work at workplace 1;</td>
</tr>
<tr>
<td>5:44pm - 6:54pm</td>
<td>Walk from workplace 1 to parking lot 2, drive to friend’s place 3;</td>
</tr>
<tr>
<td>6:54pm - 6:56pm</td>
<td>Pick up/drop off at friend 3’s place;</td>
</tr>
<tr>
<td>6:56pm - 7:15pm</td>
<td>Drive from friend 3’s place to other place 5;</td>
</tr>
<tr>
<td>9:01pm - 9:20pm</td>
<td>Drive from other place 5 to friend 3’s place;</td>
</tr>
<tr>
<td>9:20pm - 9:21pm</td>
<td>Pick up/drop off at friend 3’s place;</td>
</tr>
<tr>
<td>9:21pm - 9:50pm</td>
<td>Drive from friend 3’s place to home 1;</td>
</tr>
<tr>
<td>9:50pm - 8:22am</td>
<td>Sleep at home 1.</td>
</tr>
</tbody>
</table>

**Table 2.** Summary of a typical day based on the inference results.

4 Conclusions

We provided a novel approach to extract activities and places from GPS traces. In contrast to existing techniques, our approach uses one consistent framework for both low-level inference and the extraction of a person’s significant places. Thereby, our model is able to take high-level context into account in order to detect the significant locations of a person. Furthermore, once these locations are determined, they help to better detect low-level activities occurring in their vicinity. Summation cliques are extremely important to introduce long-term, soft constraints into activity recognition. We show how to efficiently insert such cliques into belief propagation using bidirectional FFT computations.

Our experiments based on traces of GPS data show that our system significantly outperforms existing approaches. In addition to being able to learn a person’s
significant locations, it can infer low level activities. Initial experiments show that the output of our system can be used to generate textual summaries of a person’s daily activities. In future work, we will add more sensor data to our technique, including accelerometers, audio signals, and barometric pressure. Using the additional information provided by these sensors, we will be able to perform extremely fine-grained activity recognition.

In this paper, we introduced FFT as an efficient computational tool for summation aggregates. The clique templates of RMNs are well suited to specify such clique specific inference mechanisms and we are currently developing additional techniques, including clique-specific MCMC and local dynamic programming.

References