The Hidden Permutation Model and Location-Based Activity Recognition

Hung H. Bui
Artificial Intelligence Center
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025, USA

Dinh Phung, Svetha Venkatesh, Hai Phan
Department of Computing
Curtin University of Technology
GPO Box U1987, Perth, WA 6845
Australia

Abstract

Permutation modeling is challenging because of the combinatorial nature of the problem. However, such modeling is often required in many real-world applications, including activity recognition where subactivities are often permuted and partially ordered. This paper introduces a novel Hidden Permutation Model (HPM) that can learn the partial ordering constraints in permuted state sequences. The HPM is parameterized as an exponential family distribution and is flexible so that it can encode constraints via different feature functions. A chain-flipping Metropolis-Hastings Markov chain Monte Carlo (MCMC) is employed for inference to overcome the $O(n!)$ complexity. Gradient-based maximum likelihood parameter learning is presented for two cases when the permutation is known and when it is hidden. The HPM is evaluated using both simulated and real data from a location-based activity recognition domain. Experimental results indicate that the HPM performs far better than other baseline models, including the naive Bayes classifier, the HMM classifier, and Kirshner's multinomial permutation model. Our presented HPM is generic and can potentially be utilized in any problem where the modeling of permuted states from noisy data is needed.

Introduction

The problem of dealing with permutations of objects arises naturally in many real-world problems, e.g., in modeling astronomical image data, solving correspondence and data association in computer vision, and information extraction (Kirshner, Parise, & Smyth 2003). In the realm of activity recognition, many activity routines consist of a number of steps, each of which occurs only once; in other words, they consist of a permutation of some subactivities. Furthermore, these substeps often follow certain partial ordering constraints: some are strongly ordered (e.g., if step A enables a condition required by another step B, or if both need to follow a strict timetable), some are weakly ordered (e.g., step A is performed before step B out of habit), while others might not follow any specific order at all (e.g., their relative ordering is purely by chance). In a physically grounded activity recognition system, the actual permutation of steps remains hidden to the system in most cases; available in the data stream are noisy observations that occur when a step is executed by a person. Furthermore, the strong and weak ordering constraints might not be known a priori. The main challenge is to learn these ordering constraints from noisy sensory data.

We therefore would like to investigate techniques that are effective for modeling permuted states and capture aspects of strong and weak ordering constraints exhibited in the permutation. Such techniques would need to overcome the typical $O(n!)$ complexity in dealing with a large number of permutations. In this paper, we focus mostly on the activity recognition domain; however, our approach developed here can potentially be useful in any problem where there is the need for careful modeling of permuted states from noisy data.

Existing activity modeling tools in activity recognition typically employ Markov models, e.g., hidden Markov model (HMM) and Dynamic Bayesian Network (DBN). In cases where the decomposition structures of the activities are available, more sophisticated extensions of Markov models such as stochastic grammars (Moore & Essa 2002; Minnen, Essa, & Starner 2003), Abstract HMM (AHMM) (Bui, Venkatesh, & West 2002), and Propagation Network (Shi *et al.* 2004) can be used to better capture the dynamics of the activities. Note that even for these powerful Markov models, modeling a set of partially ordered activities remains somewhat awkward. For example, in the Propagation Network, the model needs to remember all the activities that have occurred before a certain time; the size of the state space is thus exponential in the number of activities. Often, this awkwardness is the result of attempts to model permutation as a Markov state-space model. Because of the extra added complexity, these models typically require that the ordering constraints are given as a priori knowledge, in the form of either transition rules or activity transition networks.

In contrast, our work here focuses on directly learning permutation patterns corresponding to partial ordering constraints that might exhibit in the data, but are not known in advance. Closest to our approach is the work of (Kirshner, Parise, & Smyth 2003) whose aim is also to model permutation of data features arising from unknown permuted configurations in astronomical image data. Their approach, although directly attempting to model a set of permuted objects, deals with a very small number of permutations (of three objects or less), and is not designed to model subtle patterns in the permutations (such as the partial ordering constraints among the objects). Thus, two important issues in permutation modeling have not been addressed: how to

capture the partial ordering constraints, and how to scale up to larger permutation sets.

To tackle these issues, we propose a novel model for permutation modeling called the *Hidden Permutation Model* (HPM). The HPM consists of a distribution of permutations expressed as an exponential family with suitably designed features. Random permutation is drawn from the distribution and then serves as ordering indices to generate the observations. Our work can be viewed as a generalization of (Kirshner, Parise, & Smyth 2003) in which we replace their naive multinomial permutation distribution with a more suitable set of features and parametric form. To overcome the $O(n!)$ complexity, we employ the Metropolis-Hastings Markov chain Monte Carlo (MCMC) with the chain-flipping proposal scheme detailed in (Dellaert *et al.* 2003) for inference. We then provide algorithms for maximum-likelihood parameter estimation for the HPM, both when the permutation is observed (supervised) and when the permutation remains hidden (unsupervised).

We evaluate the model using both simulated data and real data from a location-based activity recognition application domain. In the latter, we use the HPM in combination with density-based clustering (Ester *et al.* 1996) to detect the occurrences of high-level activity routines in a long sequence of GPS signals. We show that the proposed HPM model outperforms other baseline models, including the naive Bayes classifier, HMM classifier, and Kirshner's multinomial permutation model.

The rest of the paper is organized as follows. We discuss the related work next, followed by a detailed description of the HPM and its learning algorithms. We then present the setting of location-based activity recognition and the experimental results in evaluating the HPM for this domain. Finally, we conclude and present our future directions.

Related Work

Recognizing high-order activities is the central theme in activity recognition research. Numerous approaches have been proposed in literature, but to our best knowledge, none has addressed the problem of learning partial ordering constraints from permuted activity sequences. Related background is vast and we do not attempt to cover it all here. Rather we focus on what is more closely related to the work in this paper. Briefly, these approaches include stochastic grammars (Moore & Essa 2002; Minnen, Essa, & Starner 2003), probabilistic propagation networks (Shi *et al.* 2004), DBN-based approaches (e.g., (Bui, Venkatesh, & West 2002; Yin, Chai, & Yang 2004; Wilson & Atkeson 2005)), conditional random fields and relational Markov networks (Liao, Fox, & Kautz 2005) and the naive Bayes classifier (Tapia, Intille, & Larson 2004).

In stochastic grammar-based approaches (Moore & Essa 2002; Minnen, Essa, & Starner 2003), a high-level activity is assumed to be generated from a set of atomic activities via a grammar defined by a set of production rules. These rules are often predefined manually, and the atomic activities are observed during training and testing. High-level activity recognition is performed by parsing a sequence of atomic activities using parsing techniques in NLP with the predefined grammar.

In the probabilistic propagation network approach (Shi *et al.* 2004), the high-level activity is modeled as a probabilistic network of atomic activities in which nodes represent atomic activities and links represent partial ordering constraints among them. Joint conditional probability distributions are used to enforce the temporal and logical constraints among parent nodes and their children. High-level activity recognition is performed by belief propagation on the probabilistic network. However, in order to use this approach, the structure of the high-level activity (ordering constraints) needs to be known a priori. In our proposed HPM, these constraints are modeled and learned directly via the feature functions specified in the exponential family parameterization.

In DBN-based approaches, e.g., (Bui, Venkatesh, & West 2002; Yin, Chai, & Yang 2004; Wilson & Atkeson 2005), high-level activity is often constructed hierarchically on top of the atomic activities layer. These activities are usually treated as hidden variables, and standard inference and learning techniques in DBN can be applied to learn and recognize activities. However, because of the strict first-order assumption, it is hard for DBN-based approaches to capture long-range ordering and permutation constraints among the atomic activities. The DBN-based approach has been applied to location-based activity recognition where high-level activities are inferred from low-level location information (Liao, Fox, & Kautz 2004). In this paper, we focus however on activities that form out of a permuted sequence of subactivities and on the problem of learning the ordering constraints among them.

The HPM is based on a general exponential family formulation; thus, it is related to the conditional random fields and relational Markov networks (Liao, Fox, & Kautz 2005). However, the support of the HPM permutation distribution, by definition, is restricted to the set of permutations. In addition, the features used in the HPM are designed to capture partial-ordering relations. Furthermore, the HPM is a generative model which can be trained in an unsupervised way.

The Hidden Permutation Model

Let $\{a_1, \ldots, a_n\}$ be the set of n objects (e.g., subactivities in our case), each a_i can generate an observation w with the emission probability $P(w \mid a_i)$ where $w = 1, \ldots, m$. Let $Per(n)$ be the set of all possible permutations over $\{1, 2, \ldots, n\}$. We define the generative process of an n-order HPM as follows. First a permutation $X = \{x_1, \ldots, x_n\} \in Per(n)$ is generated according to a distribution $P(X)$. Then each x_t is used as an index to generate the observation $o_t = w$ from the emission probability $P(w \mid a_{x_t})$. For our convenience, we will use the notation x_j^{-1} to denote the position i such that $x_i = j$.

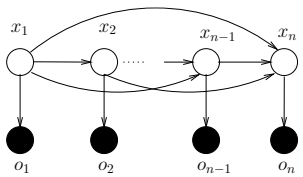

Figure 1: Graphical model for HPM

Figure 1 illustrates the HPM as a graphical model. Note that the structure of the model is somewhat similar to an

HMM, except that the hidden state sequence is now a random permutation, and thus Markov properties do not hold; in fact, $\{x_i\}$ form a fully connected Bayesian network. However, at the observation layer, each observation o_t depends only on x_t. With a slight abuse of notation we write $P(o_t \mid x_t)$ to mean $P(o_t \mid a_{x_t})$. The joint distribution of all variables in the model is given as

$$\begin{aligned} P(X, O \mid \theta) &= P(X \mid \lambda) P(O \mid X, \eta) \\ &= P(X \mid \lambda) \prod_{t=1}^{n} P(o_t \mid x_t, \eta) \end{aligned}$$

Here $\theta = \{\lambda, \eta\}$ is the parameter of the model: λ parameterizes the permutation probability $P(X \mid \lambda)$ and η parameterizes the observation model $P(O \mid X, \eta)$.

There are $n!$ elements in $Per(n)$, so it would take $n!$ parameters if $P(X)$ is modeled as a multinomial as in (Kirshner, Parise, & Smyth 2003). This approach is clearly not scalable in terms of computational complexity and, more important, sample complexity.

The key idea of our contribution is a general parameterization of the permutation distribution based on the exponential family. Let $f : Per(n) \rightarrow \mathbb{R}^d$ be a d-dimension feature function, and $\lambda \in \mathbb{R}^d$ be the parameter in our model. We consider an exponential family of the form

$$P(X \mid \lambda) = \exp\left\{\langle f(X), \lambda \rangle - A(\lambda)\right\}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, and $A(\lambda) = \ln \sum_{X \in Per(n)} \exp\{\langle f(X), \lambda \rangle\}$ is the log normalizer (a.k.a. log-partition function).

The different choice of feature functions gives us great flexibility in modeling the characteristics that we care about in the permutation. One way to capture the long-range ordering constraints[1] among subactivities is to define the features as $f(X) = \{f_{ij}(X),\ 1 \leq i, j \leq n\}$ where $f_{ij}(X) = \mathbb{I}\{x_i^{-1} < x_j^{-1}\}$, e.g., 1 if i appears before j in the permutation X and 0 otherwise. The inner-product $\langle f(X), \lambda \rangle$ becomes

$$\sum_{i,j} \lambda_{ij} \mathbb{I}\{x_i^{-1} < x_j^{-1}\} = \sum_{l,k} \lambda_{x_l x_k} \mathbb{I}\{l < k\} = \sum_{1 \leq l < k \leq n} \lambda_{x_l x_k}$$

We thus arrive at the following model:

$$P(X \mid \lambda) \quad = \quad \exp\left\{ \sum_{1 \leq l < k \leq n} \lambda_{x_l x_k} - A(\lambda) \right\} \quad (1)$$

In contrast to the multinomial model, this model has only $n(n-1)$ parameters.

To complete the model, we employ multinomial distribution for the emission probability $P(O \mid X, \eta)$. The straightforward choice is to define $P(o_t = w \mid a_i) = \eta_{iw}$. This is known as the mean-parameterization for the multinomial. However, mean-parameterization requires two constraints: $\sum_{v=1}^{m} \eta_{iv} = 1,\ \forall i$ and $0 \leq \eta_{iv} \leq 1$. These constraints make maximum likelihood estimation a constrained optimization problem, which is inconvenient to solve. We therefore prefer to work with a minimal canonical parameterization as follows

$$P(w \mid a_i, \eta_i) = \exp\left\{ \sum_{v=1}^{m-1} \eta_{iv} \mathbb{I}\{v = w\} - B(\eta_i) \right\} \quad (2)$$

where $B(\eta_i) = \ln(1 + \sum_{v=1}^{m-1} e^{\eta_{iv}})$ is the log-partition function. With canonical parameterization, there are no constraints on the parameter η_{iv}.

In summary, the HPM is parameterized by $\theta = (\lambda, \eta)$ where $\lambda = \{\lambda_{ij} \in \mathbb{R}, 1 \leq i \neq j \leq n\}$ are canonical parameters for $P(X \mid \lambda)$, and $\eta = \{\eta_{iv} \in \mathbb{R}, 1 \leq i \leq n, 1 \leq v \leq m-1\}$ are multinomial canonical parameters for the observation model.

MCMC for Approximate Inference

Given a sequence of observations $O = \{o_1, \ldots, o_n\}$, the inference task is to compute the conditional probability

$$P(X \mid O, \theta) = \frac{P(X \mid \lambda) \prod_{i=1}^{n} P(o_i | a_{x_i}, \eta)}{\sum_{X \in Per(n)} P(X \mid \lambda) \prod_{i=1}^{n} P(o_i \mid a_{x_i}, \eta)}$$

We are also interested in calculating the likelihood of the model θ with respect to the observation sequence O , i.e., the probability

$$P(O| \theta) = \sum_{X \in Per(n)} P(X \mid \lambda) \prod_{i} P(o_i \mid x_i, \eta)$$

Note that this is also the denominator that appears in the expression for $P(X \mid O,\ \theta)$.

The inference problem requires summation over all possible permutations in $Per(n)$, which takes $n!$ operations and is clearly not scalable. In this paper, we propose to use a Metropolis-Hastings MCMC approximation at this step:

$$\sum_{X} P(X \mid \lambda) \prod_{i} P(o_i | a_{x_i}, \eta) \approx \frac{\sum_{r=1}^{R} \prod_j P(o_j | a_{x_j^r}, \eta)}{R}$$

where $X^r \sim P(X|\lambda)$ and R is the number of samples. To simulate from $P(X|\lambda)$ via MCMC, we use the flipping proposal distribution in (Dellaert *et al.* 2003). From a current permutation X, we generate the next permutation X' by swapping two random positions in X. This proposal distribution is symmetric, and thus the acceptance ratio becomes $\min\left\{\frac{P(X'|\lambda)}{P(X|\lambda)} = \exp\{\sum_{l<k}(\lambda_{x'_l x'_k} - \lambda_{x_l x_k})\}, 1\right\}$.

Parameter Estimation

We present maximum likelihood (ML) parameter estimation for the model, first in the case where all variables $\{X, O\}$ are observed and then for the case where X is hidden.

Supervised learning with observed permutation Given the fully labeled dataset D consisting of K i.i.d. training sequences $\left\{X^k, O^k, k = 1, \ldots, K\right\}$, ML estimation is the solution

$$\theta_{ML} = \underset{\theta}{\text{argmax}}\ P\left(D \mid \theta\right) = \underset{\theta}{\text{argmax}}\ \mathcal{L}(D; \theta)$$

[1] Another useful set of features can be constructed to focus on neighboring positions: $f_{ij}(X) = 1$ if i appears *immediately* before j in the permutation and 0 otherwise. Yet another useful set of features can focus on value at a given position: $f_{i,l}(X) = 1$ if i appears at position l and 0 otherwise. In this paper, we use only features that lead to the form in Eq. (1).

where $\mathcal{L}(D;\theta) = \ln P(D \mid \theta)$ is the complete log-likelihood:

$$\mathcal{L}(D;\theta) = \sum_{k=1}^{K} \left[\ln P(X^k \mid \lambda) + \ln P(O^k \mid X^k, \eta)\right]$$

Clearly, the parameters λ and η are decoupled and can be optimized separately. Note that with the exponential family parameterization, $\ln P(X \mid \lambda)$ is a concave function of λ and so there exists a global maximum point for the above maximum likelihood problem. In this work, gradient ascent[2] is used to find the optimal λ. The derivative[3] is given as

$$\bigtriangledown_{\lambda_{ij}}(\mathcal{L}(D;\theta)) = \sum_{k=1}^{K} f_{ij}(X^k) - K\sum_{X} f_{ij}(X)P(X \mid \lambda)$$

Optimizing η_i however has a simple closed-form solution in which the mean parameter $P(o = w|a_i)$ is simply set to the normalized frequency count in the data.

Unsupervised learning with hidden permutation When X is hidden, our dataset D reduces to K i.i.d. sequences of observations $\{O^k, k = 1, \ldots, K\}$. The (incomplete) log-likelihood is

$$l(D;\theta) = \sum_{k=1}^{K} \log\left\{\sum_{X} P(O^k, X \mid \theta)\right\}$$

This objective function is no longer concave, and the two parameters λ and η are now coupled and need to be optimized jointly. Nevertheless, the derivatives can be evaluated and given below:

$$\begin{aligned}\bigtriangledown_{\lambda_{ij}}(l(D;\theta)) &= \sum_{k=1}^{K}\sum_{X} f_{ij}(X)P(X \mid O^k, \theta) \\ &\quad - K\sum_{X} f_{ij}(X)P(X \mid \lambda) \\ \bigtriangledown_{\eta_{iv}}(l(D;\theta)) &= \sum_{k=1}^{K}\sum_{X} \mathbb{I}\{x_i^{-1} \in O^k[v]\}P(X \mid O^k, \theta) \\ &\quad - KP(w = v|a_i, \eta_i)\end{aligned}$$

where $O^k[v] = \{j \text{ s.t. } o_j^k = v\}$.

We have experimented with two different optimization strategies: (1) simple gradient ascent, and (2) coordinate ascent that alternatively optimizes λ via a gradient step (while η is fixed) and optimizes η via an expectation maximization (EM) step (while λ is fixed). However, we find that the simple gradient ascent works much better, and this is used in our reported experiments. For a small number of permutations, the gradients can be computed exactly by summing over all permutations. When the number of permutations is large, the MCMC approximation described in previously can be used.

[2]We expect a quasi-Newton method such as conjugate gradient to give better performance.

[3]Full derivation of the derivatives in this section is omitted for brevity, but can be supplied if requested.

Activity Recognition on Campus

A student's activities on campus are rarely random. Some are spontaneous and are not bound to any strict schedule such as having coffee with friends, reading in the library, going banking, and so on. But often they are driven by a timetable such as when to take a class, where to meet for assigments, when to attend a seminar, and so on. Daily routines of a student can thus be viewed as a sequence of partially ordered 'atomic' activities.

We are interested in identifying and extracting such typical routine segments from location information contained in long daily GPS traces. The problem is non-trivial since several atomic activities might take place in the same known location. Furthermore, because of the proximity between campus buildings, locations of interest might not be detected or distinguishable from one another via GPS signals alone. We use a two-stage approach to this problem. In the first stage, raw GPS signals are clustered to find locations of significance where activities might take place. In the second stage, sequences of raw GPS signals are converted to sequences of cluster labels that are used to train the HPM models. The learned HPM models are then used to identify and extract corresponding routine segments in future GPS traces. We describe the process used to cluster GPS data below and present experimental results for activity routine extraction using the HPM in the next section.

Clustering Raw GPS Signals

Clustering of GPS signals to automatically discover places the student visits on campus presents some challenges: we do not know the number of places (clusters) in advance, the GPS signals can be noisy or lost when inside a building, and the clusters could have arbitrary, non-ellipsoidal shapes. These represent difficulties for standard parametric methods such as Gaussian mixture models. We thus employ a density-based clustering algorithm (DBSCAN) (Ester *et al.* 1996; Adams, Phung, & Venkatesh 2006) to find significant places where 'significance' implies that the student has spent more than a threshold time period at that location (5 minutes in our case). Similar to (Adams, Phung, & Venkatesh 2006), lost signals are interpolated for the interval where the signal disappears and reappears again within a spatial proximity. Density-based clustering is particularly desirable in our case because it can discover arbitrary cluster shapes, exclude noise and outliers (e.g., GPS readings recorded when the subject is moving between two places), and automatically discover the number of clusters.

In our typical setup, a student carries a PDA equipped with GPS logging functionality. Raw GPS signals are then preprocessed (interpolated) and clustered using DBSCAN. Each cluster is then consulted with the campus map to find possible matched locations. Logs of GPS data can thus be transformed into sequences of cluster labels that serve as the input to our HPM model. We collected approximately 40 hours of data from a student walking around the campus with a GPS-enabled device. Figure 2 plots seven clusters representing seven discovered significant places and their 11 corresponding matched locations (see also Table 1). Many location pairs such as Angazi/Bank, Library/Bookmark, Jones/CBS, and Watson/Psych are grouped as one, as they are indeed close in spatial proximity. For example, the

Bookmark cafe is attached to the Library, and the Bank is adjacent to the Angazi cafe.

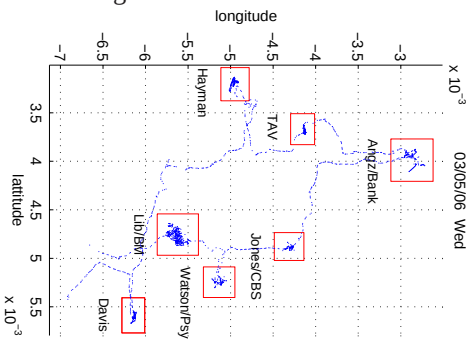

Figure 2: Learned clusters and corresponding locations

Experiment

Activity Description

We experiment with two activity routines, one consisting of five atomic activities and the other consisting of six atomic activities. Figure 3 shows the partial ordering structure among the atomic activities (each directed arrow represents an ordering constraint). For example, in the first routine, 'Group meeting' must occur after 'Lecture 1' but could occur before or after 'Lecture 2'.

Table 1 shows the names of the physical locations mapped to the atomic activities. We observe that some atomic activities (e.g., group meetings or having lunch) might occur at a number of different locations, and similarly a physical location can be associated with more than one atomic activity.

Results with Simulated Data

This experiment is designed to test performance of the HPM with large data synthesized artificially. The objective is to detect the occurrence of an activity routine in a long sequence of significant places (clusters). For each routine, 100 runs were performed. In each run a training data set consisting of 30 pairs of atomic activity/observation sequences was randomly generated. The random routine generator has a transition matrix among the atomic activities for sequence generation. Random transitions that do not satisfy the partial order constraints or the permutation constraints are discarded. In addition, a long test sequence of 1100 random observations was generated, and 20 instances of the given routine were generated and inserted randomly into the test sequence. For HPM models, MCMC is used for approximate inference. The training time required for unsupervised learning in each run takes roughly one minute on a 2GHz Pentium machine. For Kirshner's model, we perform exact summation over the entire permutation set, and thus training time is roughly doubled.

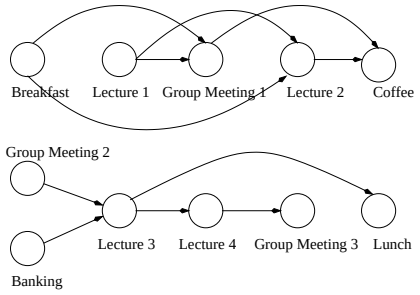

Figure 3: Structure of atomic activities in two high-level routines and their partial ordering constraints. An arrow implies an ordering constraint.

Three baseline models are used as comparisons: the naive Bayesian classifier (NBC), hidden Markov model (HMM), and Kirshner's model (KIR) (Kirshner, Parise, & Smyth 2003). Detection of the routine is performed by submitting the data in each window of length n in the data sequence to the models for classification. Average true positive (TP) and false positive (FP) rates over 100 runs are used as performance metrics; recall is the ratio of TP to the number of true activities (20 in our case) and precision is the ratio TP/(TP + FN). All models under evaluation do not have access to information about the partial ordering constraints in Figure 3.

Atomic activities	Physical locations
Banking	Bank
Lecture 1	Watson theater
Lecture 2	Hayman theater
Lecture 3	Davis theater
Lecture 4	Jones theater
Group meeting 1	Bookmark cafe, Library, CBS
Group meeting 2	Library, CBS, Psychology Bld
Group meeting 3	Angazi cafe, Psychology Bld
Coffee	TAV, Angazi cafe, Bookmark cafe
Breakfast	TAV, Angazi cafe, Bookmark cafe
Lunch	TAV, Bookmark cafe

Table 1: Atomic activities and their corresponding physical locations from campus map

Tables 2 and 3 describe the experimental results for the supervised (permutation given) and unsupervised learning (permutation hidden) cases, respectively.[4] The results reveal that the HPM outperforms both the NBC and HMM. In both supervised and unsupervised learning cases, the HPM produces a higher number of true positives, making the recall rate 4% higher than that of the HMM, while maintaining a low number of false positives, resulting in a precision rate much higher than that produced by HMM (30% higher for activity 1 and nearly 20% higher for activity 2).

		TP	FP	Precision	Recall
Activity 1	HMM	18.2	19.5	48.3%	91.0%
	KIR	18.5	2.0	**90.2**%	92.5%
	HPM	19.1	4.1	82.3%	**95.5**%
Activity 2	HMM	17.9	4.4	80.3%	89.5%
	KIR	18.0	0.7	96.3%	90.5%
	HPM	18.8	0.4	**97.9**%	**94.0**%

Table 2: Results for simulated data (supervised learning)

In comparsion with Kirshner's model, the HPM performs much better, especially in precision, except for one case in a supervised setting. Notably in the unobserved permutation case (unsupervised) the HPM improves from 68.3% to

[4]Only the sequence of cluster labels (not the true permutation) is given to the NBC; thus, NBC results are reported only in the unsupervised setting.

		TP	FP	Precision	Recall
Activity 1	NBC	16.6	11.1	59.9%	80.3%
	HMM	18.3	19.8	48.0%	91.5%
	KIR	18.3	8.5	68.3 %	91.5%
	HPM	19.1	5.1	**78.9**%	**95.5**%
Activity 2	NBC	17.1	11.0	60.9%	85.5%
	HMM	17.7	3.8	82.3%	88.5%
	KIR	18.1	4.7	79.4 %	90.5%
	HPM	18.5	0.5	**97.4**%	**92.5**%

Table 3: Results for simulated data (unsupervised learning)

78.8% and 79.4% to 97.4% for activities 1 and 2, respectively. Besides, we notice that the HPM performs fairly consistently when the number of atomic activities increases from routine 1 to routine 2, while Kirshner's model degrades very quickly. This confirms our belief in the superior performance of the HPM over KIR in handling permutations over larger sets, not to mention the complexity explosion that the KIR model may face.

Results with Real Data

We use the real data collected by a student walking around the campus as described in the previous section. Out of the approximately 40 hours of data collected, converting to cluster labels results in about 350 different cluster visits where routine 1 was noted to be performed 10 times. The best unsupervised permutation model (HPM) is compared against the HMM and the NBC. In unsupervised setting, the results can be sensitive to initialization; thus for the HMM and the HPM, we ran the program 25 times, each with a random initialization, and the mean of the results was recorded. Note that all the models being evaluated do not have access to information about the correspondence between GPS clusters and locations of interest (Figure 2), or the correspondence between atomic activities and locations (Table 1), or the partial ordering constraints (Figure 3).

	TP	FP	Precision	Recall
NBC	6	4	60%	60%
HMM	8.5	5.3	61.6%	85%
HPM	9.8	1.9	83.8%	98%

Table 4: Results for real data (unsupervised learning)

Table 4 reports the results. Performance in this experiment is fairly consistent with the simulated case. The HPM produces a high recall rate (98%), while the recall rates for HMM and NBC are 85% and 60%, respectively. In addition, the number of false positives produced by the HPM is less than half that produced by the HMM and the NBC, making the precision rate of the HPM higher than 80%, which is more than 20% higher than that produced by the two rival models.

Conclusion

In summary, we introduce the Hidden Permutation Model (HPM) for learning and recognizing high-level activity routines that consist of a permutation of partially ordered subactivities. We propose a general way to parameterize a distribution over permutations using the exponential family, design a set of suitable features for the model, discuss approximate inference, and derive updates for both supervised and unsupervised learning of the model. We also present a method based on a combination of the HPM and density-based clustering for learning activity models from GPS data.

The HPM is evaluated against the naive Bayes classifier, the HMM classifier and the multinomial permutation model in several scenarios in which ordering constraints among the subactivities of the high-level routine are not known in advance. Results show that the HPM outperforms all the other baseline models, especially in unsupervised learning, indicating that the HPM is a promising tool for permutation modeling.

The current HPM model can be extended in a number of ways. First, the strict constraint of unique occurrence in the permutation can be relaxed so that a distribution over the set of permutations with repetition can be defined. We expect that features corresponding to the number of repetitions of an object in the permutation will be useful. Second, since the HPM is defined based on an exponential family, it is conceptually straightforward to consider a discriminative counterpart of the HPM. Similar to the way we have compared the HPM to the HMM, comparison of such a discriminative permutation model against standard a discriminative model (e.g., conditional random fields) could provide further proof of the usefulness of the current permutation model.

Note that the HPM proposed in this work is not necessarily restricted to the activity recognition domain. We would like to investigate its applicability to other domains with permuted data such as in target tracking or a wide range of other problems mentioned in (Kirshner, Parise, & Smyth 2003).

Acknowledgment. This material is based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-07-D-0185/0004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA, or the Air Force Research Laboratory (AFRL).

References

Adams, B.; Phung, D.; and Venkatesh, S. 2006. Extraction of social context and application to personal multimedia exploration. In *Proceedings 14th Annual ACM International Conference on Multimedia*, 987–996. ACM Press New York, NY, USA.

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

Dellaert, F.; Seitz, S. M.; Thorpe, E.; and Thrun, S. 2003. EM, MCMC, and chain flipping for structure from motion with unknown correspondence. *Machine Learning* 50(1-2):45–71.

Ester, M.; Kriegel, H.-P.; Sander, J.; and Xu, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings KDD-96*, 226–231.

Kirshner, S.; Parise, S.; and Smyth, P. 2003. Unsupervised learning with permuted data. In *Proceedings ICML 03*, 345–352.

Liao, L.; Fox, D.; and Kautz, H. 2004. Learning and inferring transportation routines. In *Proceedings AAAI'04*.

Liao, L.; Fox, D.; and Kautz, H. 2005. Location-based activity recognition using relational Markov networks. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'05)*.

Minnen, D.; Essa, I.; and Starner, T. 2003. Expectation grammars: Leveraging high-level expectations for activity recognition. In *Proceedings CVPR'03*.

Moore, D., and Essa, I. 2002. Recognizing multitasked activities using stochastic context-free grammar. In *Proceedings AAAI'02*.

Shi, Y.; Huang, Y.; Minen, D.; Bobick, A.; and Essa, I. 2004. Propagation networks for recognizing partially ordered sequential action. In *Proceedings CVPR'04*.

Tapia, E. M.; Intille, S. S.; and Larson, K. 2004. Activity recognition in the home setting using simple and ubiquitous sensors. In *Proceedings PERVASIVE '04, LNCS 3001*, 158–175.

Wilson, D. H., and Atkeson, C. 2005. Simultaneous tracking and activity recognition (STAR) using many anonymous, binary sensors. In *Proceedings PERVASIVE'05*.

Yin, J.; Chai, X.; and Yang, Q. 2004. High-level goal recognition in a wireless lan. In *Proceedings AAAI'04*.