Partially Hidden Markov Models for Privacy-preserving Modeling of Indoor Trajectories

Aditya Jitta[a], Arto Klami[a]

[a] *Helsinki Institute for Information Technology HIIT,*
Department of Computer Science, University of Helsinki

Abstract

Markov models are natural tools for modeling trajectories, following the principle that recent location history is predictive of near-future directions. In this work we study Markov models for describing and predicting human movement in indoor spaces, with the goal of modeling the movement on a coarse scale to protect the privacy of the individuals. Modern positioning devices, however, provide location information on a much more finer scale. To utilize this additional information we develop a novel family of partially hidden Markov models that couple each observed state with an auxiliary side information vector characterizing the movement within the coarse grid cell. We implement the model as a nonparametric Bayesian model and demonstrate it on real-world trajectory data collected in a hypermarket.

Keywords: Hierarchical Dirichlet process, Markov models, Movement trajectories, Nonparametric Bayesian inference, Privacy

---

1. Introduction

Human movement in indoor spaces can be reliably tracked with various localization techniques, such as wireless network signal strength [1, 2], dead reckoning [3], or locally deployed high-accuracy positioning systems often based on Bluetooth smart [4]. The state-of-the-art solutions can localize individuals with sufficient accuracy to know where exactly they are in some complex indoor space, such as a museum, hypermarket, or other public place. The measurement error of the most accurate systems is measured in the range of 0.1-1 meters, and is hence comparable or smaller than the personal space of an individual. For many practical purposes the localization can therefore be considered error-free.

Given access to location data, a natural question to consider is modeling of movement trajectories, either to describe movement patterns within the building or to attempt predicting future locations based on already observed locations of an individual. In recent years models for indoor trajectories have been presented based on various alternative modeling approaches: For example, Nielsen et al. [5] used hidden Markov models for movement trajectories to improve localization accuracy, Yoo et al. [6] used Gaussian process models for trajectories to learn a map of the building, and Nianyin et al. [7] used particle swarm optimization for planning robot trajectories.

Our work falls into the same general category: We model indoor movement trajectories based on high-accuracy positioning data, building models for both descriptive and predictive analysis. Our goal is to design justified and automated Bayesian tools for this task, without requiring or revealing too detailed information about the individuals. Even though the measurement devices can provide near-perfect positioning accuracy, the typical use cases for the premise owners, such as targeted advertising or collecting statistics on movement patterns, do not require knowing the exact positions. In most cases it is enough to know that the client is, for example, browsing the dairy section of a market, whereas the knowledge that they are currently handling a specific product might be considered intrusive. To preserve the privacy of the customers, it hence makes sense to consider models that do not reveal or even require storing the exact locations.

The easiest privacy-preserving solution is to discretize the locations on a sufficiently coarse scale, effectively mimicking the kind of data a less accurate positioning tool would provide. This is naturally not optimal since it completely ignores the improved positioning accuracy. In this work we build models that are fundamentally based on discretization, but that complement the discretized coarse locations (called cells) with aggregate summary statistics based on the high-accuracy positioning data. These summaries intend to capture the nature of the movement within the cell, without retaining the actual detailed coordinates. For learning the model we then only need to store the information on which cell the person is in, coupled with these summary statistics, without providing further access to the raw high-accuracy coordinates. Ideally the summary statistics would be computed already at the level of the positioning system itself so that the raw coordinates could be discarded in real time, to best guarantee that the privacy of the users is not compromised.

Markov Models (MM) that assume the next state depends solely on the current state are computationally tractable tools for modeling trajectories over such discrete observed states, and have been used as a crude approximation for human mobility trajectories as well [8]. For improved accuracy, we should typically model also higher-order transitions, conditioning the expected movements not only on the latest state but on a sequence of the recent states, for example using variable-order MMs [9, 10]. While Markov models are indeed good tools for predicting future movements, they completely ignore the detailed movement within the cells.

In this work we extend MMs to support also the auxiliary statistics, via a latent state formulation. We provide for each cell a collection of latent states that generate the auxiliary statistics, and further condition the transition probabilities to the next cell not only on the observed state history but also on the latent state. Even though the abstract formulation reminds the concept of hidden Markov models (HMM) [11] and our inference borrows some key elements from HMM literature, it differs fundamentally in two respects: The latent states are conditional on the observed states (and not vice versa as in HMMs), and the model efficiently supports higher-order transitions.

We call the model partially hidden Markov model (PHMM), since the dynamics operate on the combination of the observed discrete states and the latent states conditional on those. We implement the model within the nonparametric Bayesian framework, using a three-level extension of hierarchical Dirichlet process (HDP) [12, 13] for determining the local state cardinalities and beam sampling [14] for the latent state inference. We evaluate the model on artificial data, and then proceed to demonstrate its use in modeling actual high-precision indoor data collected in a hypermarket during a period of one month. We illustrate how the model naturally provides interpretable summaries of movement patterns within the space via the latent states, and that it can predict future movements of the individuals.

The main contributions of this work can be summarized as:

- Introduction of the novel PHMM model that extends Markov models by complementing them with latent states that influence both the transition probabilities and emission probabilities for feature vectors associated with the observed discrete states, improving both interpretability and prediction accuracy.
- Non-parametric Bayesian implementation of PHMM to automatically determine the required number of latent states, and generalization of the PHMM model for higher-order histories.
- Use of PHMM for modeling indoor movement trajectories in a privacy-preserving manner; instead of modeling raw coordinates we model movement along coarse grid that does not reveal unnecessarily detailed information about the user, while using the detailed coordinates only to create a feature vector characterizing the type of movement within each grid cell.

2. Background

Before describing the proposed model, we briefly cover the necessary background. We introduce first the most closely related models to enable understanding how the partially hidden Markov model is related to regular and hidden Markov models, as well as dynamic Bayesian networks in general, and then give a quick overview to the mathematical tools used when building the non-parametric PHMM in Section 3.

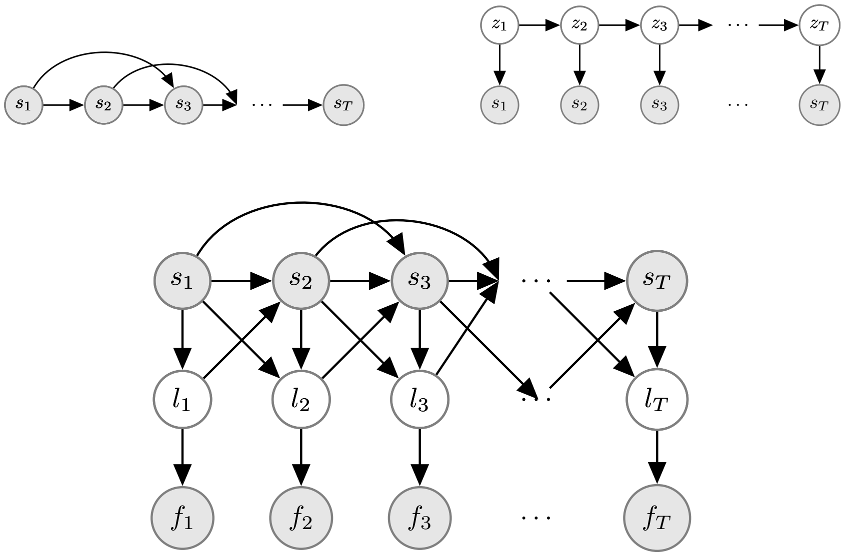


Figure 1: A Markov model (top left) operates solely on the observed states s_t and easily supports higher-order transition histories, whereas a hidden Markov model (top right) introduces latent states z_t that govern the transitions but the inference is only efficient for first-order transitions. The partially-hidden Markov model (bottom) introduced in this work combines advantages of both: It supports higher-order transitions but can still take advantage of latent states, though in a different manner than a HMM. In a PHMM the latent states are conditional on the observed state s_t, and generate an auxiliary representation f_t instead of the main observed state s_t.

2.1. Markov Models

Markov models (Figure 1; top left) are tools for modeling fully observed sequences of discrete states. Given a sequence $s_1, ..., s_T$, the goal is to learn the underlying dynamics in form of the transition probabilities $p(s_t|s_{t-1})$. This is computationally easy, since the maximum likelihood estimates are obtained by merely counting the observed transitions into a $S \times S$ matrix, where S is the number of different states. Bayesian inference is effectively as easy, using Dirichlet priors for the transition probabilities.

For many real-word sequences the pure Markovian assumption of the next state being conditioned only on the previous one does not hold. Higher-order

Markov models relax the assumption by modeling transitions $p(s_t|s_{t-1}, ..., s_{t-O})$ up to some order O. Even though the transitions probabilities now become tensors of order $O + 1$ and hence require more memory, the inference algorithm remains the same. For better accuracy with finite data sets the models are typically implemented with some sort of backing-off, allowing the model to borrow statistical strength from lower-order history for the rare sequences, which increases the complexity of the algorithms but reduces the memory consumption. The backing-off can be implemented for example by variable order Markov models; for nonparametric Bayesian examples, see [15] and [10].

2.2. Hidden Markov Models

A hidden Markov model (HMM) (Figure 1; top right) generalizes Markov models by coupling the observed states s_t with latent states z_t. The Markovian dynamics are assumed for the latent states, so that z_t depends on z_{t-1}, and the observed states are emitted by the latent states:

$$\begin{aligned} z_t \mid z_{t-1} &\sim \pi_{z_t}, \\ s_t \mid z_t, \theta &\sim F(\theta_{z_t}). \end{aligned}$$

Here π_z is a vector of probabilities for the K possible latent states, and $F(\theta_z)$ is some density over the space of the observed states, parameterized by θ_z.

HMMs are more expressive than MMs, but it comes with a notable increase in computational cost: The latent state sequence $z_1, ..., z_T$ needs to be inferred in addition to the transition probabilities, and these two tasks are coupled in a manner that typically requires alternating algorithms.[1]

Ideally the whole latent state sequence is inferred at once, using dynamic programming [11]. For sampling-based Bayesian inference the algorithm is called Forward-Filtering Backward-Sampling (FFBS). Importantly, this part of the inference algorithm has computational complexity $\mathcal{O}(TK^2)$, where T is the length

---

[1] See [16] for a closed-formed solution that requires access to highly accurate estimates of multivariate densities.

of the sequence and K the number of the latent states. In practice HMMs are only applicable for first-order histories because of this term; for completing the forward-backward inference for an Oth order model the complexity would be $\mathcal{O}(TK^{O+1})$, which quickly becomes infeasible, often already for the 2nd order.

Both MMs and HMMs, as well as the PHMM introduced in this work, are special cases of dynamic Bayesian networks [17]. Even though all of them can be presented in this general framework, the special cases typically result in considerably more efficient inference and hence dedicated solutions are important. In our case, the notable advantage compared to generic formulation is that the conditional independence assumptions made on the latent states enable efficient nonparametric treatment where the cardinality of the latent states is learned automatically. To our knowledge, no generic inference solutions for nonparametric dynamic Bayesian networks have been presented.

2.3. Hierarchical Dirichlet Process

HMMs, as well as PHMMs, assume the data is generated by some unknown number of latent states K. While the number K could be manually set by the analyst, an interesting alternative is to infer the number directly based on the data using nonparametric Bayesian modeling techniques. Here we briefly review the nonparametric model of hierarchical Dirichlet process (HDP) [12], which has been used to implement nonparametric HMMs [18, 19, 14]. Later we will use tools belonging to the same family to create a nonparametric version of the proposed model.

Dirichlet process (DP) is a stochastic process that provides densities of the form $f(\theta) = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}(\theta)$, where δ is the delta measure and π_k sum upto one. A draw from such a process is denoted by $G \sim DP(\alpha \mathbf{H})$, where $\mathbf{H}$ is a density from which the θ_k are drawn, and α is a concentration parameter that controls the decay of the weights π_k. In other words, DP gives a countably infinite collection of atoms, weighted points in some space, and can hence be used for example for creating mixture models: Instead of assuming a fixed mixture of K components, we can use a DP to generate infinitely many of them, of which only a finite set is still needed for modeling any finite data collection. This enables efficient inference [12].

A hierarchical DP (HDP) extends DPs into a hierarchy, generating parallel DPs that share the same atoms but that can have different weights for them. In the simplest form the hierarchy is stated as

$$G \sim DP(\alpha \mathbf{H}), \qquad G_j \sim DP(\beta G).$$

The lower level DPs use G as their prior, which necessarily implies the θ drawn from them are part of the discrete set provided by G, and hence the different G_j share the same atoms, where G_j is the j^{th} random probability measure that shares atoms with the base measure G. Since a HMM can be re-formulated as a collection of mixture models where the mixture weights depend on the previous mixture allocation, the HDP construction can be used to implement a nonparametric HMM [18, 19, 14].

3. Partially Hidden Markov Model

In this work we propose a novel family of partially hidden Markov models for modeling discrete sequences with associated feature vectors, illustrated in Figure 1 (bottom). The input is given as sequences of observations denoted by $x_1, x_2, \dots x_T$, where each time instance is determined by a tuple $x_t = (s_t, f_t)$. The first element s_t is a discrete state, whereas the second element f_t is a D-dimensional feature vector providing (typically real-valued) auxiliary side information for that state. In our application the former corresponds to the grid cell the user is in, and the latter to summary statistics of the movement pattern within the cell. Both s_t and f_t are observed.

A regular Markov model would model such data by ignoring f_t completely, simply modeling s_t conditional on some Oth order history of previous locations. This is naturally sub-optimal, since it completely ignores the features. Another classical alternative for modeling such sequences would be a HMM, which would have a set of K latent states that would emit the whole tuples x_t. In a straightforward application of HMM, each latent state could hence generate several cell locations, and hence the latent states could not be directly interpreted as location information. Furthermore, inference for HMMs is only feasible for very low-order transition histories, typically the first order.

To combine the advantages of both classical alternatives, we model the sequences with a model we call partially hidden Markov model (PHMM). As regular MMs, it supports efficient inference for higher order transitions and directly models the observed state sequence s_t. At the same time, it inherits from HMMs the capability of modeling also the associated feature vectors f_t with a collection of latent states $l_t^{s_t}$. An important difference to HMMs is that the latent states are conditional on the observed state s_t.

The basic formulation of the model is given by

$$\begin{aligned} s_t &\sim p(s_t | l_{t-1}^{s_{t-1}}, s_{t-1}, s_{t-2}, ..., s_{t-O}), \\ l_t^{s_t} &\sim p(l_t^{s_t} | l_{t-1}^{s_{t-1}}, s_t, s_{t-1}), \\ f_t &\sim p(f_t | l_t^{s_t}), \end{aligned} \tag{1}$$

with additional special cases for the first time points, not written out here for brevity. In verbal terms, the feature vector itself depends only on the latent state, the latent state depends on the previous latent state and the previous cell (and naturally also on the current cell, since each cell has its own set of latent states), and finally the next cell depends on the previous latent state and the Oth order history of the cell locations. The use of only first-order history for the latent states themselves is crucial for efficient inference of the latent states, yet the whole model exhibits higher-order transitions efficiently because of the transitions for s_t.

The full model, developed in the next sections, instantiates a nonparametric Bayesian version of this basic pattern by coupling the transition probabilities with suitable prior distributions, inferring the number of local states for each cell nonparametrically, and by hierarchically sharing the latent states of different locations.

3.1. Nonparametric PHMM

In the following, we will provide the details on how to implement PHMM using nonparametric Bayesian tools for automatically inferring the number of latent states required for each of the observed states s_t. Since the latent states are conditional on the observed ones, a parametric model would require setting S different complexity parameters manually. Unless assuming the same cardinality for each observed state, this would render tools like cross-validation completely infeasible. Consequently, nonparametric inference is particularly important for this model class.

A simple mixture density for $f_t|s_t$ could be implemented with a DP prior. Sharing the clusters across the observed states would require a HDP instead [12], as would taking time dependencies into account as in HDP-HMM [18]. Since our model combines both elements, we will need yet another hierarchical layer, for which we adopt the tree-HDP construction [13]. In the following we present the details of these constructions only to the extent it is necessary for deriving the eventual sampling equations for the proposed model; for formal treatment of the random processes the reader should consult the original sources.

The full model, illustrated in Figure 2, is a single tree-HDP with three layers. At the highest level stand a collection of global latent states with associated weights, drawn from the top-level DP. At the next level are S collections of local latent states (l), one for each geographic cell. These use the global collection as their prior, which means they share identities but have different weights. Typically, each local collection uses only a subset of the global states.

Finally, the conditional transformations are tied to each other so that all incoming transitions to a cell s use the local latent state collection as their prior. This means that only the latent states present in that cell can be reached, and that the weights of incoming transitions (π) are regularized towards each other. In the end each local state generates a feature vector (f). The generating distribution can be arbitrary, but in our work we use multivariate Gaussian emissions.

The formal notation for the model can be constructed as a special case of

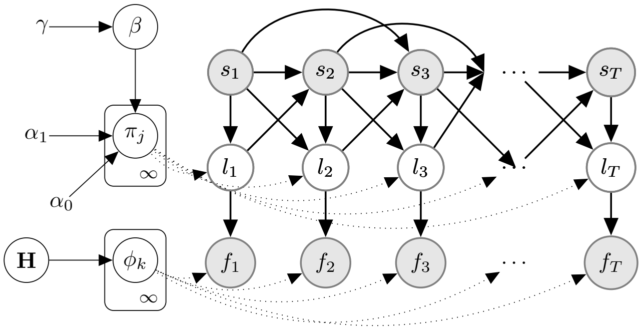


Figure 2: Plate diagram of the nonparametric partially hidden Markov model. The nodes at the left indicate the three levels of the tree-structured HDP construction, l_t are latent states, and the shaded nodes correspond to the observed coarse locations s_t and the associated feature vectors f_t.

the tree-HDP model [13]. Tree-HDP extends HDPs into general tree structures, whereas the model used here is a specific simple tree with three layers, defined as

$$G_0 \sim DP(\gamma \mathbf{H}),$$
$$G_s \sim DP(\alpha_0 G_0),$$
$$G_{is} \sim DP(\alpha_1 G_s).$$

Here H is a base measure over the feature vector space, G_0 is the root-level DP that provides the global latent states, and G_s correspond to the collections of latent states for each cell. Given a grid of S cells, there are S of these collections. Finally, G_{is} refers to one particular incoming transition from a neighboring cell to the sth cell. Here i is an implicit index that runs over the possible states from where one can reach the sth cell; each i corresponds to an Oth order sequence of cells combined with the latent state of the previous time index.

One approach for understanding the fairly abstract formulation above is to think in terms of the analogous finite model. Then we would simply have K global latent states with emission distributions drawn from the prior H and weights following a Dirichlet distribution. For each cell s we would then have

Dirichlet-distributed weights using the global weights as their prior, and finally for all incoming transitions to that location the probabilities would again be Dirichlets, this time using the local weights as their prior. In practice the nonparametric construction can also be implemented in a similar fashion, using the stick-breaking construction [20]; the finite Dirichlets are replaced with infinite ones parameterized via sticks drawn from beta-distributions. We denote the global sticks by $V_k \sim \mathrm{Beta}(1, \gamma)$ and the associated weights by $\beta_k = V_k \prod_i^{k-1}(1 - V_i)$, the cell-level sticks by $v_{jk} \sim \mathrm{Beta}(\alpha_0 \beta_k, \alpha_1(1 - \sum_{l<k} \beta_l))$ and the corresponding weights by p_{jk}, and finally the sticks and weights corresponding to the transitions by $a_{ijk} \sim \mathrm{Beta}(\alpha_1 p_{jk}, \alpha_1(1 - \sum_{l<k} p_{jl}))$ and $\pi_{ijk} = a_{ijk} \prod_{l<k}(1 - a_{ijl})$. For derivations of these exact forms, see [12, 13].

3.2. Inference

Given a collection of observed sequences, we infer the model parameters by Gibbs sampling. The whole inference process is split into two separate parts: Inference of the latent state sequences given the rest of the parameters, and inference of the parameters given the state sequences.

Given the latent state sequences the inference details follow from [13], since the model is a special case of their tree-HDP model. Despite the somewhat complicated machinery required for correctly handling the nonparametric nature of the model, the updates for the model parameters still depend only on aggregate count statistics as they would for a parametric model.

We denote by N_j the total number of incoming transitions into the jth grid cell, and by n_{ijk} the number of those coming from the ith history and using the global latent state k. Furthermore, we denote by m_{jk} the total number of latent states at the grid cell level G_j that are assigned to the global latent state k. The quantities n_{ijk} and m_{jk} are not fully observed, but instead need to be sampled as explained by [13]. Finally, the transition counts t_{gjkl} are the number of transitions from the gth grid cell using the latent state l to the jth grid cell using the latent state k. Given the above aggregate statistics, the model parameters can be sampled as follows:

1. Break more sticks at the global level to support creation of new states at lower levels:
$$\beta \mid \gamma \sim \text{Stick}(\gamma)$$
2. Update the global weights:
$$P(m_{jk} = m) \propto s(m_{jk}, k).(\alpha_0 \beta_k^{old})^m$$
$$v_k = \sum_j m_{jk}$$
$$\beta_k \sim \text{Dirichlet}(v_1, \ldots, v_k, \gamma)$$
3. Update the cell weights:
$$P(n_{ijk} = n) \propto s(n_{ijk}, k).(\alpha_1 \beta_{jk}^{old})^n$$
$$l_{jk} = \sum_i^{N_j} n_{ijk}$$
$$\beta_{jk} \sim \text{Dirichlet}(l_{j1}, \ldots, l_{jk}, \alpha_0 \beta_k^{old})$$
4. Update the transition probabilities:
$$\pi_{gjl} \sim \text{Dirichlet}(t_{gjl1}, \ldots, t_{gjlk}, \alpha_1 \beta_{jk}^{old})$$

Here $s(n, k)$ denotes the Stirling numbers of the first kind; see [13] for further explanation.

Given the current values for the transition probabilities we then sample the full state sequence of T elements at once. Even though the model is not a HMM, we can perform this stage using an analogous forward-backward sampling procedure since the transitions depend only on the previous latent state and not longer history of those. As we recall from Section 2, the complexity of this depends on the number of latent states, which here is unbounded. The first HDP-HMM models circumvented this by not sampling the whole sequence at one go, but the beam sampler by [14] showed how we can not only perform forward-backward sampling for HDP-HMM but in fact can often do it with less computational demand compared to a regular HMM with similar state cardinality; only transitions with sufficiently high probability need to be considered, which often means considerably less computation in total.

We extend the beam sampler of [14] for our model as follows. We denote by s_t the grid cell at time t, and by l_t the corresponding latent state, and for brevity denote (s_t, l_t) by h_t.

Following the basic idea of beam sampling (and slice samplers in general), we draw an auxiliary slice variable $u_t \sim \text{Uniform}(0, \pi_{h_{t-1},h_t})$ for each time point to represent the lowest transition probability that need to be considered, adaptively truncating the model to a finite one for the purpose of this step alone. The forward filtering step can then be written as

$$\begin{aligned}
&P(h_t \mid x_{1:t}, u_{1:t}) \propto P(h_t, u_t, x_t \mid x_{1:t-1}, u_{1:t-1}) \\
&= P(x_t \mid h_t) \sum_{h_{t-1}} \mathbb{1}(u_t < \pi_{h_{t-1},h_t}) P(h_{t-1} \mid x_{1:t-1}, u_{1:t-1}) \\
&= P(x_t \mid h_t) \sum_{h_{t-1}: u_t < \pi_{h_{t-1},h_t}} P(h_{t-1} \mid x_{1:t-1}, u_{1:t-1}),
\end{aligned}$$

and the backwards sampling is performed by

$$h_t \sim P(h_t \mid x_{1:t}, u_{1:t}) P(h_{t+1} \mid h_t, u_{t+1}).$$

Note that this adaptive truncation is not a heuristic strategy, but the slice sampling technique indeed draws samples from the correct posterior. Furthermore, we typically need to consider only a small subset of the states for each summation above; see [14] for details.

4. Illustration

To illustrate the basic behavior of the model, we apply it on four different artificial data sets generated from a parametric version of the PHMM model, assuming a set of $S = 6$ discrete states (cells). The four data sets showcase increasingly more complex dynamics, the easiest corresponding to regular Markovian assumption and the last one corresponding to second-order Markovian transitions further conditioned on $K = 5$ latent states associated with each cell. The generative process is exactly as described in (1), where we draw the transition probabilities for both s and l from Dirichlet distributions with the prior

Table 1: Predictive accuracy (in percentages) of the proposed partially observed Markov model (PHMM) and the baseline of regular Markov model (MM) on four artificial data sets. For the data sets with only one latent state (K=1) the models are equivalent, as they should, but for cases with more latent states (K=5) PHMM outperforms MM that ignores the auxiliary features. The table also shows the models restricted to only first order histories are not as accurate if the data exhibits second order transitions (O=2), motivating the support for higher-order dynamics. The boldface font indicates for each data set the best methods for which the performance is indistinguishable; the small deviations are because of random fluctuation.

Artificial Data	MM-1	MM-2	PHMM-1	PHMM-2
O = 1, K = 1	**92.3**	**92.2**	**91.9**	**92.5**
O = 1, K = 5	34.5	44.0	**65.7**	**64.5**
O = 2, K = 1	33.8	**63.7**	34.1	**62.3**
O = 2, K = 5	28.4	40.3	46.3	**65.4**

parameter 0.1, in order to create distributions that deviate notably from uniform density. For the cases with latent states the emissions $p(f_t|l_t)$ are normal distributions with means $(-5, -2.5, 0, 2.5, 5)$ and shared standard deviation 1.

We applied four alternative models on each of the data sets: MM-1, MM-2, PHMM-1, and PHMM-2, where the number after the dash denotes the order of the model. These correspond exactly to the requirements of the four data sets; all four methods should solve the first data sets, whereas only the last one is flexible enough to model the most complex data set. We train the models using a sequence of $5,000$ samples drawn from the model and evaluate them using the predictive accuracy on a separate test sequence of $5,000$ samples, for the task of identifying the next cell. Table 1 shows the methods work as expected; PHMM-2 is superior for the data that requires 2nd order dynamics and latent states. The simpler data sets can be modeled correctly by some of the alternatives as well, but notably PHMM-2 is always on par with the best ones. We also confirmed that the nonparametric PHMM models correctly learnt the number of the latent states, and that the best methods reach the optimal accuracy obtained when predicting with the true generating model.

Finally, we also tried higher-order MMs for the last data set to study whether the lack of latent states can be compensated by considering even longer histories on the observed states. The accuracy of MM indeed goes up for higher orders, reaching 50.8% for MM-3 and 48.0% for MM-4, but PHMM-2 still outperforms these clearly while requiring smaller transition tensors than MM-4. The same effect is visible already in the $O = 1, K = 5$ case where MM-2 outperforms MM-1.

To summarize the results, PHMM reached the best accuracy in all experiments; for the more complex data sets it outperformed the simpler alternatives, but even for the simpler generative processes it reached the same accuracy and hence the only drawback is in additional computational cost. Importantly, MMs with even higher order transitions were not as accurate as PHMMs.

5. Modeling Indoor Movement

The main application motivation for this work is in modeling indoor movement while preserving the privacy of the clients. Typical end-use scenarios for such models are in understanding how people behave in public spaces such as shopping centers, museums, or office buildings. For all these cases the owners of the premise are interested in understanding the flows and making predictions to support location-based services and to dynamically allocate resources, for example by opening more counters based on the predicted movement patterns. At the same time, these are all examples where the owner has no need to know the exact locations of the individuals, and it is reasonable to actively prevent them from being able to spy on them by never storing the detailed location data.

5.1. Data and feature representation

We apply the model in a retail environment, modeling location data collected in a hypermarket by tracking shopping carts and baskets with a commercial high-accuracy positioning system (HAIP) provided by Quuppa. The system tracks small Bluetooth Smart chips integrated in the carts and baskets, providing accurate (error less than 1 meter) position with 10Hz frequency. We model the data at granularity of 20x20 meter coarse locations, to get a rough overview at a departmental level, using a total of 1,839 sequences collected during a period of 30 days.

At the core of the PHMM model are the aggregate summaries collected based on the more detailed location data, stored to accompany each time point of the coarse trajectory. In this work we present a few simple alternative representations, primarily to demonstrate that the more accurately the aggregate characteristics capture the nature of the movement within the coarse location, the better the overall model will be.

These representations are not specifically tuned for our evaluation, since the idea is that the features would be extracted already before handing the data for someone that learns the actual model and hence they should be generally applicable for various kinds of modeling tasks.

We compute a set of eight basic features (Table 2). These features extract natural elements about the movement, covering aspects like the amount of time spent (Δt), how often the person stopped (for example to pick items from the shelves; Pauses), and characterizations of their general movement direction ($\Delta X_+, \Delta X_-, \Delta Y_+, \Delta Y_-$). All of these features are privacy-preserving in the sense that they do not reveal the precise location of the person at any point. For illustrating the effect of the quality of the local representation, we then construct alternative feature sets as subsets of these eight basic features. The simplest set includes just the time spent (corresponding to a semi-Markov model), whereas the best coverage is obtained by using all of them. Besides these extremes, we also ran experiments with two intermediate collections.

5.2. Experiments

The PHMM model has two core elements that control its expressive power: The maximum order of transition history with respect to s, and the accuracy of the feature vector f in characterizing the local movement behavior. To illustrate how the model behaves with respect to these two elements, we first conduct

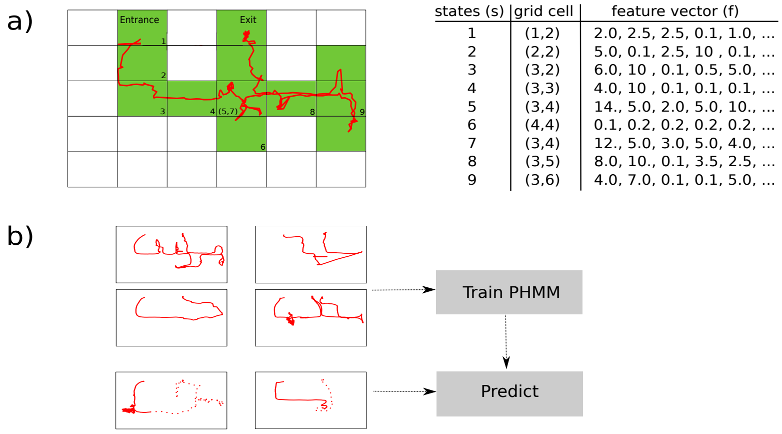
Table 2: Description of the features used for characterizing the local movement patterns. Each feature describes the movement within the cell, and hence for example the mean coordinates are with respect to the origin of the grid cell. ΔX_+ and ΔX_- correspond to the distances moved in positive X direction and negative X direction respectively. Similarly, we have ΔY_+ and ΔY_-.

Δt	Logarithm of the total time spent
$\Delta X_+, \Delta X_-, \Delta Y_+, \Delta Y_-$	Total distance moved rightwards, leftwards, upwards, downwards
M_x, M_y	Mean of the x and y
Pauses	Total number of stops or pauses

separate experiments for each of them in isolation. After demonstrating that improving either element indeed results in better predictive accuracy, we present the results for a model that uses the best choices for both elements.

For all models we measure the accuracy using a setup where $1,471$ trajectories are used for training and 368 trajectories are used for testing. We train the model using the training trajectories, running the Gibbs sampler for $2,000$ iterations and discarding the first $1,500$ samples as burn-in. For each test trajectory we randomly sample a time of prediction, meaning that we assume we have recorded the trajectory up to that point and then need to predict the next few locations. For the observed part we infer the latent trajectory as we do for the training samples, and we then predict the future points using simple forward sampling: We instantiate 100 particles for each test sequence, propagate them forward in time using the transition probabilities, and finally compute the accuracy by averaging over the predictions of these particles. The accuracy score is defined as the ratio of these particles that fall to the exact correct grid cell.

Figure 3 illustrates the complete modeling pipeline, showing both how individual trajectories are represented using grid cells and feature vectors, as well depicting the training and test procedure described above.



states (s)	grid cell	feature vector (f)
1	(1,2)	2.0, 2.5, 2.5, 0.1, 1.0, ...
2	(2,2)	5.0, 0.1, 2.5, 10 , 0.1, ...
3	(3,2)	6.0, 10 , 0.1, 0.5, 5.0, ...
4	(3,3)	4.0, 10 , 0.1, 0.1, 0.1, ...
5	(3,4)	14., 5.0, 2.0, 5.0, 10., ...
6	(4,4)	0.1, 0.2, 0.2, 0.2, 0.2, ...
7	(3,4)	12., 5.0, 3.0, 5.0, 4.0, ...
8	(3,5)	8.0, 10., 0.1, 3.5, 2.5, ...
9	(3,6)	4.0, 7.0, 0.1, 0.1, 5.0, ...

Figure 3: Illustration of the way the PHMM model is applied for modeling indoor trajectories in a privacy-preserving manner. **a)** Every path is pre-processed by extracting the grid cell identifiers for all distinct visits to individual cells, and for each visit we compute the feature vector representation characterizing the nature of the movement within the cell. **b)** We use 75% of the path sequences for training, utilizing the full trajectories. The empirical performance of the model is then evaluated in a prediction task: For the remaining 25% of sequences we observe the sequence only upto a randomly determined time point, and attempt to predict the remaining steps along the sequence, indicated here by the dotted lines.

5.2.1. Higher-order History

We start the experiments by looking at a special case of PHMM with no local states, which corresponds to a regular MM. This experiment is conducted to verify that higher-order transitions indeed are useful for this kind of data. Figure 4 (left) shows that 2nd order MMs are considerably more accurate than 1st order, but there is no notable difference between 2nd order models and the ones with even higher order on this data. Based on this observation, we will use 2nd order history for the final PHMM model, as the lowest complexity choice of the well-performing ones.

5.2.2. Local Pattern Models

The more interesting element of the PHMM model is the local feature description and the associated local states. Here we experiment with increasingly more complex feature descriptions, keeping the order of the model fixed to one,

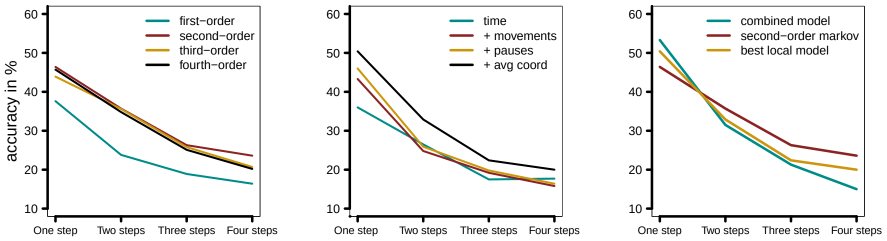


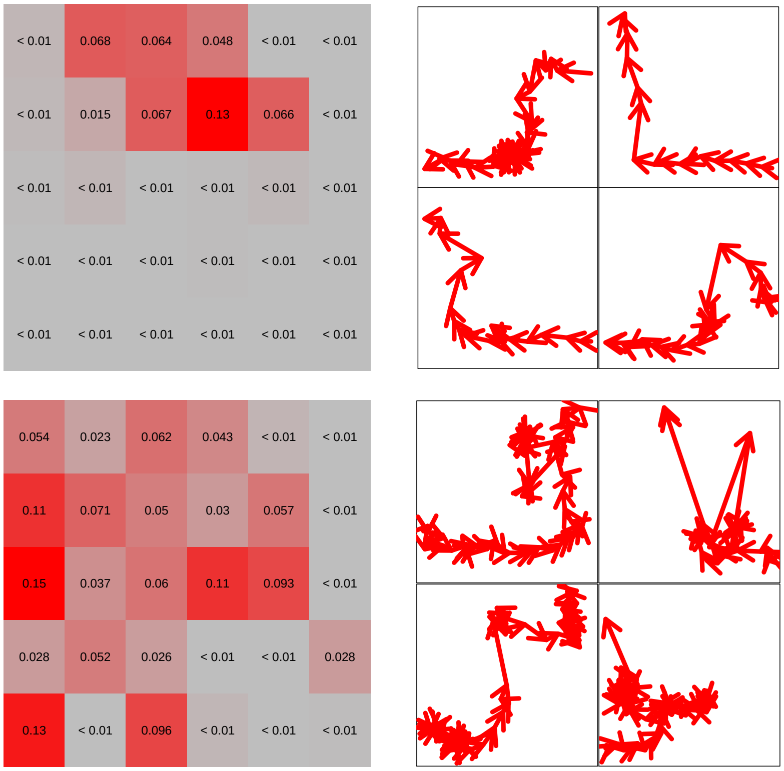
Figure 4: **Left**: 2nd order Markov models are clearly more accurate than 1st order models, but on this data the higher order ones do not help. **Middle**: More descriptive local feature representations improve the accuracy of a 1st order PHMM. Merely knowing the time the user spent in the grid location does not help much compared to no local states (see the left subfigure), but all additional features improve the accuracy. **Right**: Combining the 2nd order history with the best local model further improves the accuracy for short-term predictions, but not for long-term ones. In all three sub-plots the x-axis denotes how many time-points in future we are predicting and the y-axis indicates the accuracy of making exactly the right prediction.

to show that knowing more about the fine movements within the grid cell helps creating more expressive models.

The more interesting element of the PHMM model is the local feature description and the associated local states. Here we experiment with increasingly more complex feature descriptions, keeping the order of the model fixed to one, to show that knowing more about the fine movements within the grid cell helps creating more expressive models.

Figure 4 (middle) shows prediction accuracies for four model variants. The first variant has very simple feature representation (only the time spent in the cell), whereas all others progressively add more features. The best accuracy is obtained with the model that has the most features, which confirms the intuitive expectation.

To further understand the difference between the different local models, we can inspect the number of global latent states the nonparametric formulation learns for each of them. These also go up when the feature description contains more information, from 6 to 24 (averaged over the posterior samples; note that none of the cells actually use all of these) when going from the simplest repre-



< 0.01	0.068	0.064	0.048	< 0.01	< 0.01
< 0.01	0.015	0.067	0.13	0.066	< 0.01
< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01

0.054	0.023	0.062	0.043	< 0.01	< 0.01
0.11	0.071	0.05	0.03	0.057	< 0.01
0.15	0.037	0.06	0.11	0.093	< 0.01
0.028	0.052	0.026	< 0.01	< 0.01	0.028
0.13	< 0.01	0.096	< 0.01	< 0.01	< 0.01

Figure 5: Illlustration of prevalences of movement patterns across the hypermarket. The pattern on the top row corresponds to primarily leftwards movement carried out without notable pauses, illustrated for examples on the right. The heatmap on the left shows the pattern is primarily used in the top-most areas of the store, near the counters, and it corresponds to customers walking in front of the counters towards one with a short queue. The other example shown on the bottom row corresponds to more complex movement pattern within a grid cell, revealing multiple stops or reversals. This pattern is frequent in the left side of the store, which contains clothes and other items the customers often browse for a longer time. The heatmaps show the cell-level probabilities acting as the prior for all incoming transitions, roughly corresponding to the ratio of partial trajectories belonging to this particular latent state; both of these patterns explain 10-15% of the local movement patterns in the most common cells.

sentation to the most complex. Intuitively, there is no need for multiple local states when the feature descriptions are not expressive, whereas more states can be used to differentiate between different movement patterns when the feature

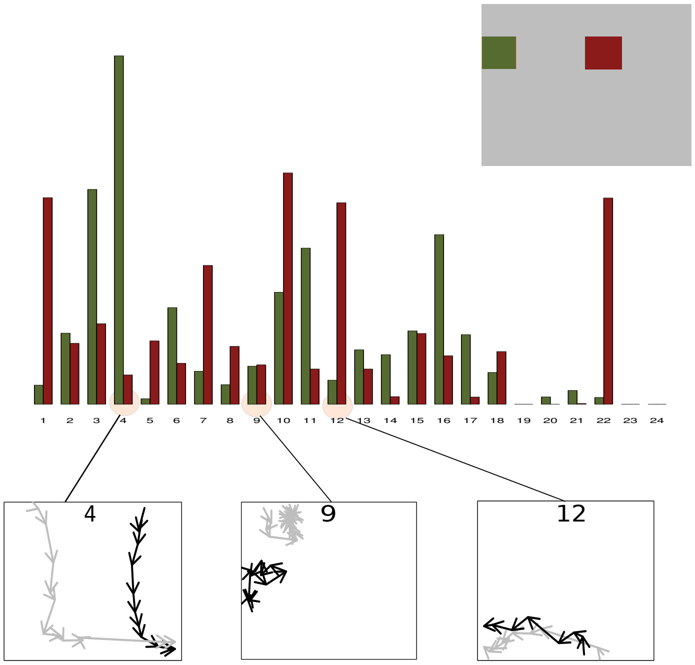


Figure 6: Illustration of PHMM in analysis of indoor movement. The grid on the top-right corner shows the discretized store layout, and we have chosen two cells (green and red) for illustrating the model. This posterior sample contains 24 global latent states shared by all cells (ordered in decreasing global probability β_k), and the bar plot in the middle shows the local probabilities p_{jk} of these two cells. We see that infrequently used global states (states from 19 to 24) have on average lower weights for these cells and some rare states can also have high probability in specific cells (like state 22 for the red cell). To further illustrate the results we show three latent states in more detail, presenting two example path snippets falling into each of these. State 4 has high probability in the green cell and corresponds to downwards movement that eventually turns left; it is almost absent in the red cell that has no corridors like this. State 9 corresponds to a pattern where the customer stops to inspect something, and it is present in both cells. Finally, state 12 corresponds to leftwards movement and naturally has low probability in the green cell since the store ends on its left side. Even though we here illustrated the actual path snippets for visualization purposes, it is good to remember the model itself only knew about the summary statistics.

vector is rich.

5.2.3. Combination

Given the above results indicating that higher-order history and accurate local feature representations are beneficial, we run the final experiments with a PHMM model that uses 2nd order history and the most complex representation for the local movements.

Figure 4 (right) collects the predictive accuracies of the earlier special cases together with this final model, showing that for predicting the immediate next grid cell the combination provides the best accuracy. For longer-term predictions it is only on par with the PHMM using 1st order history, and in fact less accurate than a standard 2nd order MM. However, it is important to note that predictive accuracy is here merely a proxy for roughly evaluating the models, and not the end goal of our work as such. Even though the PHMM model is only comparable to the alternatives in pure predictive accuracy, its primary use is in describing the movement patterns, which we will illustrate next.

The only way to interpret a Markov model is to inspect the transition probabilities, which becomes cumbersome especially when looking at higher order transitions, whereas HMMs would not directly associate the latent states with the grids. PHMM, in turn, provides latent states for each grid cell and the states share identities across different cells. Each latent state is also coupled with a prototypical feature vector that describes a movement pattern within the cell. By inspecting the probabilities of the different local states across the space we can easily identify areas where people exhibit certain type of behavior. We present an brief example of such analysis in Figure 6, showing example latent states for two distinct locations of the store. We also show a spatial distribution of the usage of a particular latent state, and examples of real trajectory pieces mapped to these latent states in Figure 5 More detailed analysis of the movement patterns within the undisclosed market studied here is outside the scope of this publication.

6. Conclusion

In this work we studied Markovian models for privacy-preserving modeling of indoor movement patterns in a fixed physical space. Instead of directly modeling high-accuracy positioning traces, we model the movement patterns at a level of a coarse grid, representing the fine movements with the grid cells via aggregate feature vectors. This way the detailed locations of the user are not revealed for the analyst, yet some information about them enters the model.

To model sequences of observed grid locations and the associated feature vectors, we proposed novel partially-hidden Markov model, which borrows elements both from higher-order Markov models and hidden Markov models. It supports transitions that take into account higher-order dynamics in terms of the observed grid locations, but still can infer the latent states associated with them with a forward-backward step that is as efficient to compute as the corresponding algorithm for HMMs. We instantiated a nonparametric Bayesian version of the general model structure, using tools from tree-structured HDPs [13] and beam sampling [14] for inference.

We demonstrated on real trajectory data collected in a hypermarket that the proposed model is suitable for this kind of data, and that both good-quality auxiliary features and higher-order transitions are needed for interpretable summaries that are also predictive of future movement. While a regular higher-order Markov model provides predictions of comparable accuracy, it does not provide the latent states that describe typical movement patterns in different parts of the space.

The main limitation of the work considers specification of the cell grid underlying the model. Here we used a simple evenly spaced grid that did not take the layout of the store into account. This sometimes results in spurious back and forth movement between two cells when cell borders are located on areas where the users spends time without moving much. Furthermore, the While the model handles these cases correctly, learning states that correspond to short visits followed by returning to the previous cell, it has detrimental effect on the prediction accuracy. One potential remedy would be to discard very short visits as a pre-processing step. Taking the store layout into account when placing the grid, so that cell borders would typically be along walls or shelves, would help as well. Finally, an interesting possibility for future work would be to extend the model to support more flexible cells that would not necessarily fall into a regular grid but would be designed based on the actual layout.

The main focus in this work was in presenting the PHMM model itself and providing the necessary inference details. For practical applications the feature representation used for characterising detailed movement within grid cells would warrant more extensive study; replacing the crude representation used here with more elaborate descriptions would likely result in improved accuracy with no additional computation.

Acknowledgments

We acknowledge support from TEKES (Digile SHOK programme D2I) and Academy of Finland (Finnish Centre of Excellence in Computational Inference Research COIN, grant 251170; grant 266969).

References

[1] H. Liu, H. Darabi, P. Banerjee, J. Liu, Survey of wireless indoor positioning techniques and systems, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 37 (6) (2007) 1067–1080.

[2] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, J. Sievänen, A probabilistic approach to wlan user location estimation, International Journal of Wireless Information Networks 9 (3) (2002) 155–164.

[3] O. Woodman, R. Harle, Pedestrian localisation for indoor environments, in: Proceedings of the 10th international conference on Ubiquitous computing, ACM, 2008, pp. 114–123.

[4] E. Dahlgren, H. Mahmood, Evaluation of indoor positioning based on Bluetooth Smart technology, Master's thesis, Chalmers University of Technology (2014).

[5] J. J. Nielsen, N. Amiot, T. K. Madsen, Directional hidden markov model for indoor tracking of mobile users and realistic case study, in: European Wireless 2013; 19th European Wireless Conference, 2013, pp. 1–6.

[6] J. Yoo, H. J. Kim, K. H. Johansson, Mapless indoor localization by trajectory learning from a crowd, in: 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2016, pp. 1–7.

[7] N. Zeng, H. Zhang, Y. Chen, B. Chen, Y. Liu, Path planning for intelligent robot based on switching local evolutionary pso algorithm, Assembly Automation 36 (2) (2016) 120–126.

[8] A. Asahara, A. Sato, K. Maruyama, K. Seto, Predestrian-movement prediction based on mixed Markov-chain model, in: Proceedings of ACM SIGSPATIAL GIS, 2011, pp. 25–33.

[9] R. Begleiter, R. El-Yaniv, G. Yona, On prediction using variable order Markov models, Journal of Artificial Intelligence Research 22 (2004) 385–421.

[10] F. Wood, C. Archambeau, J. Gasthaus, L. James, Y. W. Teh, A stochastic memoizer for sequence data, in: Proceedings of the 26th International Conference on Machine Learning, 2009.

[11] L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, in: Proceedings of the IEEE, 1989, pp. 257–286.

[12] Y. W. Teh, M. I. Jordan, M. J. Beal, D. M. Blei, Hierarchical Dirichlet processes, Journal of the American Statistical Association 101 (476) (2006) 1566–1581.

[13] K. R. Canini, T. L. Griffiths, A nonparametric Bayesian model of multi-level category learning, in: Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI), 2011.

[14] J. Van Gael, Y. Saatci, Y. W. Teh, Z. Ghahramani, Beam sampling for the infinite hidden Markov model, in: Proceedings of the 25th International Conference on Machine Learning (ICML), 2008.

[15] D. Mochihashi, E. Sumita, The infinite Markov model, in: Advances in neural information processing systems, 2007, pp. 1017–1024.

[16] D. Hsu, S. M. Kakade, T. Zhang, A spectral algorithm for learning hidden markov models, Journal of Computer and System Sciences 78 (5) (2012) 1460–1480.

[17] K. P. Murphy, Dynamic Bayesian networks: Representation, inference and learning, Ph.D. thesis, University of California, Berkeley (2002).

[18] M. J. Beal, Z. Ghahramani, C. E. Rasmussen, The infinite hidden Markov model, in: Advances in Neural Information Processing Systems, 2001, pp. 577–584.

[19] E. B. Fox, E. B. Sudderth, M. I. Jordan, A. S. Willsky, An HDP-HMM for systems with state persistence, in: Proceedings of the 25th International Conference on Machine Learning, ICML '08, 2008, pp. 312–319.

[20] H. Ishwaran, L. F. James, Gibbs sampling methods for stick-breaking priors, Journal of the Americal Statistical Association 96 (453) (2001) 161–173.