

Navigate Like a Cabbie: Probabilistic Reasoning from Observed Context-Aware Behavior

Brian D. Ziebart, Andrew L. Maas, Anind K. Dey, and J. Andrew Bagnell
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

bziebart@cs.cmu.edu, amaas@andrew.cmu.edu, anind@cs.cmu.edu, dbagnell@ri.cmu.edu

ABSTRACT

We present *PROCAB*, an efficient method for Probabilistically Reasoning from Observed Context-Aware Behavior. It models the context-dependent utilities and underlying reasons that people take different actions. The model generalizes to unseen situations and scales to incorporate rich contextual information. We train our model using the route preferences of 25 taxi drivers demonstrated in over 100,000 miles of collected data, and demonstrate the performance of our model by inferring: (1) decision at next intersection, (2) route to known destination, and (3) destination given partially traveled route.

Author Keywords

Decision modeling, vehicle navigation, route prediction

ACM Classification Keywords

I.5.1 Pattern Recognition: Statistical; I.2.6 Artificial Intelligence: Parameter Learning; G.3 Probability and Statistics: Markov Processes

INTRODUCTION

We envision future navigation devices that learn drivers' preferences and habits, and provide valuable services by *reasoning* about driver behavior. These devices will incorporate real-time contextual information, like accident reports, and a detailed knowledge of the road network to help mitigate the many unknowns drivers face every day. They will provide context-sensitive route recommendations [19] that match a driver's abilities and safety requirements, driving style, and fuel efficiency trade-offs. They will also alert drivers of unanticipated hazards well in advance [25] – even when the driver does not specify the intended destination, and better optimize vehicle energy consumption using short-term turn prediction [10]. Realizing such a vision requires new models of non-myopic behavior capable of incorporating rich contextual information.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UbiComp '08, September 21–24, 2008, Seoul, Korea.

Copyright 2008 ACM 978-1-60558-136-1/08/09...\$5.00.

In this paper, we present *PROCAB*, a method for Probabilistically Reasoning from Observed Context-Aware Behavior. *PROCAB* enables reasoning about context-sensitive user actions, preferences, and goals – a requirement for realizing ubiquitous computing systems that provide the right information and services to users at the appropriate moment. Unlike other methods, which directly model action sequences, *PROCAB* models the negative *utility* or *cost* of each action as a function of contextual variables associated with that action. This allows it to model the *reasons* for actions rather than the actions themselves. These reasons *generalize* to new situations and differing goals. In many settings, reason-based modeling provides a *compact* model for human behavior with fewer relationships between variables, which can be learned more precisely. *PROCAB* probabilistically models a distribution over all behaviors (*i.e.*, sequences of actions) [27] using the principle of maximum entropy [9] within the framework of inverse reinforcement learning [18].

In vehicle navigation, roads in the road network differ by type (*e.g.*, interstate vs. alleyway), number of lanes, and speed limit. In the *PROCAB* model, a driver's utility for different combinations of these road features is learned, rather than the driver's affinity for particular roads. This allows generalization to locations where a driver has never previously visited.

Learning good context-dependent driving routes can be difficult for a driver, and mistakes can cause undesirable travel delays and frustration [7]. Our approach is to learn driving routes from a group of experts – 25 taxi cab drivers – who have a good knowledge of alternative routes and contextual influences on driving. We model context-dependent taxi driving behavior from over 100,000 miles of collected GPS data and then recommend routes that an efficient taxi driver would be likely to take.

Route recommendation quality it is difficult to assess. Instead, we evaluate using three prediction tasks:

- **Turn Prediction:** What is the probability distribution over actions at the next intersection?
- **Route Prediction:** What is the most likely route to a specified destination?
- **Destination Prediction:** What is the most probable destination given a partially traveled route?

We validate our model by comparing its prediction accuracy on our taxi driver dataset with that of existing approaches. We believe average drivers who will use our envisioned smart navigation devices choose less efficient routes, but visit fewer destinations, making route prediction somewhat harder and destination prediction significantly easier.

In the remainder of the paper, we first discuss existing probabilistic models of behavior to provide points of comparison to PROCAB. We then present evidence that preference and context are important in route preference modeling. Next, we describe our PROCAB model for probabilistically modeling context-sensitive behavior compactly, which allows us to obtain a more accurate model with smaller amounts of training data. We then describe the data we collected, our experimental formulation, and evaluation results that illustrate the PROCAB model’s ability to reason about user behavior. Finally, we conclude and discuss extensions of our work to other domains.

BACKGROUND

In this section, we describe existing approaches for modeling human actions, behavior, and activities in context-rich domains. We then provide some illustration of the contextual sensitivity of behavior in our domain of focus, modeling vehicle route preferences.

Context-Aware Behavior Modeling

An important research theme in ubiquitous computing is the recognition of a user’s current activities and behaviors from sensor data. This has been explored in a large number of domains including classifying a user’s posture [17], tracking and classifying user exercises [5], classifying a user’s interruptibility [6] and detecting a variety of daily activities within the home [26, 24]. Very little work, in comparison, looks at how to predict what users want to do: their goals and intentions. Most context-aware systems attempt to model context as a proxy for user intention. For example, a system might infer that a user wants more information about a museum exhibit because it observes that she is standing near it.

More sophisticated work in context-awareness has focused on the problem of predicting where someone is going or going to be. Bhattacharya and Das used Markov predictors to develop a probability distribution of transitions from one GSM cell to another [3]. Ashbrook and Starner take a similar approach to develop a distribution of transitions between learned significant locations from GPS data [2]. Patterson *et al.* used a Bayesian model of a mobile user conditioned on the mode of transportation to predict the users future location [20]. Mayrhofer extends this existing work in location prediction to that of context prediction, or predicting what situation a user will be in, by developing a supporting general architecture that uses a neural gas approach [15]. In the Neural Network House, a smart home observed the behaviors and paths of an occupant and learned to anticipate his needs with respect to temperature and lighting control. The occupant was tracked by motion detectors and a neural network was used to predict the next room the person was

going to enter along with the time at which he would enter and leave the home [16].

However, all of these approaches are limited in their ability to learn or predict from only small variations in user behavior. In contrast, the PROCAB approach is specifically designed to learn and predict from small amounts of observed user data, much of which is expected to contain context-dependent and preference-dependent variations. In the next section, we discuss the importance of these variations in understanding route desirability.

Understanding Route Preference

Previous research on predicting the route preferences of drivers found that only 35% of the 2,517 routes taken by 102 different drivers were the “fastest” route, as defined by a popular commercial route planning application [13]. Disagreements between route planning software and empirical routes were attributed to contextual factors, like the time of day, and differences in personal preferences between drivers.

We conducted a survey of 21 college students who drive regularly in our city to help understand the variability of route preference as well as the personal and contextual factors that influence route choice. We presented each participant with 4 different maps labeled with a familiar start and destination point. In each map we labeled a number of different potentially preferable routes. Participants selected their preferred route from the set of provided routes under 6 different contextual situations for each endpoint pair. The contextual situations we considered were: early weekday morning, morning rush hour, noon on Saturday, evening rush hour, immediately after snow fall, and at night.

Context	Route						
	A	B	C	D	E	F	G
Early morning	6	6	4	1	2	2	0
Morning rush hour	8	4	5	0	1	2	1
Saturday noon	7	5	4	0	1	2	2
Evening rush hour	8	4	5	0	0	3	1
After snow	7	4	4	3	2	1	0
Midnight	6	4	4	2	1	4	0

Table 1. Context-dependent route preference survey results for one particular pair of endpoints

The routing problem most familiar to our participants had the most variance of preference. The number of participants who most preferred each route under each of the contextual situations for this particular routing problem is shown in Table 1. Of the 7 available routes to choose from (A-G) under 6 different contexts, the route with highest agreement was only preferred by 8 people (38%). In addition to route choice being dependent on personal preferences, route choice was often context-dependent. Of the 21 participants, only 6 had route choices that were context-invariant. 11 participants used two different routes depending on the context, and 4 participants employed three different context-dependent routes.

Our participants were not the only ones in disagreement over the best route for this particular endpoint pair. We generated route recommendations from four major commercial mapping and direction services for the same endpoints to compare against. The resulting route recommendations also demonstrated significant variation, though all services are context- and preference-independent. Google Maps and Microsoft’s MapPoint both chose route E, while MapQuest generated route D, and Yahoo! Maps provided route A.

Participants additionally provided their preference towards different driving situations on a five-point Likert scale (see Table 2). The scale was defined as: (1) very strong dislike, avoid at all costs; (2) dislike, sometimes avoid; (3) don’t care, doesn’t affect route choice; (4) like, sometimes prefer; (5) very strong like, always prefer.

Situation	Preference				
	1	2	3	4	5
Interstate/highway	0	0	3	14	4
Excess of speed limit	1	4	5	10	1
Exploring unfamiliar area	1	8	4	7	1
Stuck behind slow driver	8	10	3	0	0
Longer routes with no stops	0	4	3	13	1

Table 2. Situational preference survey results

While some situations, like driving on the interstate are disliked by no one and being stuck behind a slow driver are preferred by no one, other situations have a wide range of preference with only a small number of people expressing indifference. For example, the number of drivers who prefer routes that explore unfamiliar areas is roughly the same as the number of drivers with the opposite preference, and while the majority prefer to drive in excess of the speed limit and take longer routes with no stops, there were a number of others with differing preferences. We expect that a population covering all ranges of age and driving ability will possess an even larger preference variance than the more homogeneous participants in our surveys.

The results of our formative research strongly suggest that drivers’ choices of routes vary greatly and are highly dependent on personal preferences and contextual factors.

PROCAB: CONTEXT-AWARE BEHAVIOR MODELING

The variability of route preference from person to person and from situation to situation makes perfectly predicting every route choice for a group of drivers extremely difficult. We adopt the more modest goal of developing a *probabilistic model* that assigns as much probability as possible to the routes the drivers prefer. Some of the variability from personal preference and situation is explained by incorporating contextual variables within our probabilistic model. The remaining variability in the probabilistic model stems from influences on route choices that are unobserved by our model.

Many different assumptions and frameworks can be employed to probabilistically model the relationships between contextual variables and sequences of actions. The PROCAB approach is based on three principled techniques:

- Representing behavior as sequential actions in a *Markov Decision Process* (MDP) with parametric cost values
- Using *Inverse Reinforcement Learning* to recover cost weights for the MDP that explain observed behavior
- Employing the *principle of maximum entropy* to find cost weights that have the least commitment

The resulting probabilistic model of behavior is context-dependent, compact, and efficient to learn and reason about. In this section, we describe the comprising techniques and the benefits each provides. We then explain how the resulting model is employed to efficiently reason about context-dependent behavior. Our previous work [27] provides a more rigorous and general theoretical derivation and evaluation.

Markov Decision Process Representation

Markov Decision Processes (MDPs) [21] provide a natural framework for representing sequential decision making, such as route planning. The agent takes a sequence of *actions* ($a \in A$), which transition the agent between *states* ($s \in S$) and incur an action-based *cost*¹ ($c(a) \in \mathbb{R}$). A simple deterministic MDP with 8 states and 20 actions is shown in Figure 1.

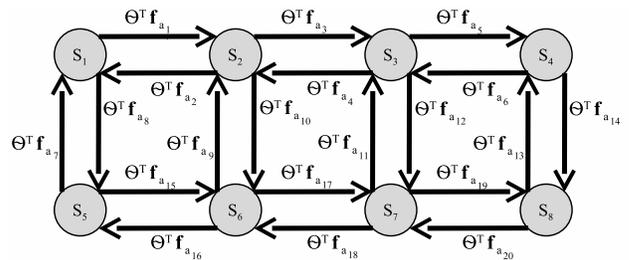


Figure 1. A simple Markov Decision Process with action costs

The agent is trying to minimize the sum of costs while reaching some destination. We call the sequence of actions a *path*, ζ . For MDPs with parametric costs, a set of *features* ($\mathbf{f}_a \in \mathbb{R}^k$) characterize each action, and the cost of the action is a linear function of these features parameterized by a *cost weight vector* ($\theta \in \mathbb{R}^k$). Path features, \mathbf{f}_ζ , are the sum of the features of actions in the path: $\sum_{a \in \zeta} \mathbf{f}_a$. The path cost is the sum of action costs (Figure 1), or, equivalently, the cost weight applied to the path features.

$$\text{cost}(\zeta|\theta) = \sum_{a \in \zeta} \theta^\top \mathbf{f}_a = \theta^\top \mathbf{f}_\zeta$$

The MDP formulation provides a natural framework for a number of behavior modeling tasks relevant to ubiquitous computing. For example, in a shopping assistant application, shopping strategies within an MDP are generated that optimally offset the time cost of visiting more stores with the benefits of purchasing needed items [4].

¹The negation of costs, *rewards*, are more common in the MDP literature, but less intuitive for our application.

The advantage of the MDP approach is that the cost weight is a compact set of variables representing the reasons for preferring different behavior, and if it is assumed that the agent acts sensibly with regard to incurred costs, the approach generalizes to previously unseen situations.

Inverse Reinforcement Learning

Much research on MDPs focuses on efficiently finding the optimal behavior for an agent given its cost weight [21]. In our work, we focus on the inverse problem, that of finding an agent’s cost weight given demonstrated behavior (*i.e.*, traversed driving routes). Recent research in *Inverse Reinforcement Learning* [18, 1] focuses on exactly this problem. Abbeel and Ng showed that if a model of behavior matches feature counts with demonstrated feature counts, \mathbf{f} (Equation 1), then the model’s expected cost matches the agent’s incurred costs[1].

$$\sum_{\text{Path } \zeta_i} P(\zeta_i) \mathbf{f}_{\zeta_i} = \tilde{\mathbf{f}} \quad (1)$$

This constraint has an intuitive appeal. If a driver uses 136.3 miles of interstate and crosses 12 bridges in a month’s worth of trips, the model should also use 136.3 miles of interstate and 12 bridges in expectation for those same start-destination pairs, along with matching other features of significance. By incorporating many relevant features to match, the model will begin to behave similarly to the agent. However, many distributions over paths can match feature counts, and some will be very different from observed behavior. In our simple example, the model could produce plans that avoid the interstate and bridges for all routes except one, which drives in circles on the interstate for 136 miles and crosses 12 bridges.

Maximum Entropy Principle

If we only care about matching feature counts, and many different distributions can accomplish that, it is not sensible to choose a distribution that shows preference towards other factors that we do not care about. We employ the mathematical formulation of this intuition, the principle of maximum entropy [9], to select a distribution over behaviors. The probability distribution over paths that maximizes Shannon’s information entropy while satisfying constraints (Equation 1) has the least *commitment* possible to other non-influential factors. For our problem, this means the model will show no more preference for routes containing specific streets than is required to match feature counts. In our previous example, avoiding all 136 miles of interstate and 12 bridge crossings except for one single trip is a very low entropy distribution over behavior. The maximum entropy principle much more evenly assigns the probability of interstate driving and bridge crossing to many different trips from the set. The resulting distribution is shown in Equation 2.

$$P(\zeta|\theta) = \frac{e^{-\text{cost}(\zeta|\theta)}}{\sum_{\text{path } \zeta'} e^{-\text{cost}(\zeta'|\theta)}} \quad (2)$$

Low-cost paths are strongly preferred by the model, and paths with equal cost have equal probability.

Compactness Advantage

Many different probabilistic models can represent the same dependencies between variables that pertain to behavior – context, actions, preference, and goal. The main advantage of the PROCAB distribution is that of compactness. In a more compact model, fewer parameters need to be learned so a more accurate model can be obtained from smaller amounts of training data.

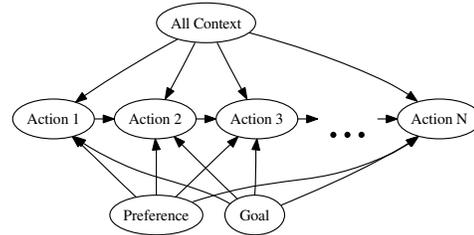


Figure 2. Directed graphical model of sequential context-sensitive, goal-oriented, preference-dependent actions

Directly modeling the probability of each action given all other relevant information is difficult when incorporating context, because a non-myopic action choice depends not only on the context directly associated with that action (*e.g.*, whether the next road is congested), but context associated with all future actions as well (*e.g.*, whether the roads that a road leads to are congested). This approach corresponds to a directed graphical model (Figure 2) [22, 14] where the probability of each action is dependent on all contextual information, the intended destination, and personal preferences. If there are a large number of possible destinations and a rich set of contextual variables, these conditional action probability distributions will require an inordinate amount of observed data to accurately model.

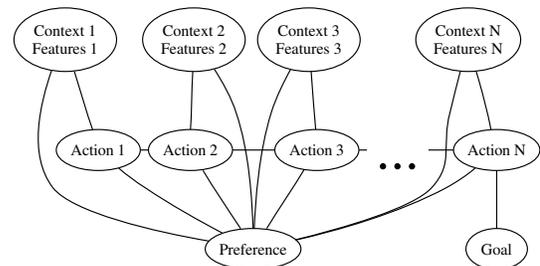


Figure 3. Undirected graphical model of sequential context-sensitive, goal-oriented, personalized actions

The PROCAB approach assumes drivers imperfectly minimize the cost of their route to some destination (Equation 2). The cost of each road depends only on the contextual information directly associated with that road. This model corresponds to an undirected graphical model (Figure 3), where cliques (*e.g.*, “Action 1,” “Context 1,” and “Preference”) define road costs, and the model forms a distribution over paths based on those costs. If a road becomes congested, its cost will increase and other alternative routes will become more probable in the model without requiring any other road costs to change. This independence is why the PROCAB model

is more compact and can be more accurately learned using more reasonable amounts of data.

Probabilistic Inference

We would like to predict future behavior using our PRO-CAB model, but first we must train the model by finding the parameter values that best explain previously demonstrated behavior. Both prediction and training require probabilistic inference within the model. We focus on the inference of computing the expected number of times a certain action will be taken given known origin, goal state, and cost weights². A simple approach is to enumerate all paths and probabilistically count the number of paths and times in each path the particular state is visited.

Algorithm 1 Expected Action Frequency Calculation

Inputs: cost weight θ , initial state s_o , and goal state s_g

Output: expected action visitation frequencies $D_{a_{i,j}}$

Backward pass

1. Set $Z_{s_i} = 1$ for valid goal states, 0 otherwise
2. Recursively compute for T iterations

$$Z_{a_{i,j}} = e^{-\text{cost}(a_{i,j}|\theta)} Z_{s:a_{i,j}}$$

$$Z_{s_i} = \sum_{\text{actions } a_{i,j} \text{ of } s_i} Z_{a_{i,j}}$$

Forward pass

3. Set $Z'_{s_i} = 1$ for initial state, 0 otherwise
4. Recursively compute for T iterations

$$Z'_{a_{i,j}} = Z'_{s_i} e^{-\text{cost}(a_{i,j}|\theta)}$$

$$Z'_{s_i} = \sum_{\text{actions } a_{j,i} \text{ to } s_i} Z'_{a_{j,i}}$$

Summing frequencies

5. $D_{a_{i,j}} = \frac{Z'_{s_i} e^{-\text{cost}(a_{i,j}|\theta)} Z_{s_j}}{Z_{s_{\text{initial}}}}$

Algorithm 1 employs a more tractable approach by finding the probabilistic weight of all paths from the origin (o) to a specific action (a), $Z'_a = \sum_{\zeta_{o \rightarrow a}} e^{-\text{cost}(\zeta)}$, all paths from the action to the goal (g)³, $Z_a = \sum_{\zeta_{a \rightarrow g}} e^{-\text{cost}(\zeta)}$ and all paths from the origin to the goal, $Z_o = Z'_g = \sum_{\zeta_{o \rightarrow g}} e^{-\text{cost}(\zeta)}$. Expected action frequencies are obtained by combining these results (Equation 3).

$$D_a = \frac{Z_a Z'_a e^{-\text{cost}(a)}}{Z_o} \quad (3)$$

Using dynamic programming to compute the required Z values is exponentially more efficient than path enumeration.

²Other inferences, like probabilities of sequence of actions or one specific action are obtained using the same approach.

³In practice, we use a finite T , which considers a set of paths of limited length.

Cost Weight Learning

We train our model by finding the parameters that maximize the [log] probability of (and best explain) previously observed behavior.

$$\theta^* = \underset{\theta}{\text{argmax}} \log \prod_i \frac{e^{-\text{cost}(\tilde{\zeta}_i|\theta)}}{\sum_{\text{path } \zeta_j} e^{-\text{cost}(\zeta_j|\theta)}} \quad (4)$$

Algorithm 2 Learn Cost Weights from Data

Stochastic Exponentiated Gradient Ascent

Initialize random $\theta \in \mathfrak{R}^k, \gamma > 0$

For $t = 1$ to T :

For random example, compute D_a for all actions a
(using Algorithm 1)

Compute gradient, ∇F from D_a (Equation 5)

$\theta \leftarrow \theta e^{\frac{\gamma}{t} \nabla F}$

We employ a gradient-based method (Algorithm 2) for this convex optimization. The gradient is simply the difference between the demonstrated behavior feature counts and the model's expected feature counts, so at the optima these feature counts match. We use the action frequency expectations, D_a , to more efficiently compute the gradient (Equation 5).

$$\tilde{\mathbf{f}} - \sum_{\text{Path } \zeta_i} P(\zeta_i) \mathbf{f}_{\zeta_i} = \tilde{\mathbf{f}} - \sum_a D_a \mathbf{f}_a \quad (5)$$

The algorithm raises the cost of features that the model is over-predicting so that they will be avoided more and lowers the cost of features that the model is under-predicting until convergence near the optima.

TAXI DRIVER ROUTE PREFERENCE DATA

Now that we have described our model for probabilistically reasoning from observed context-aware behavior, we will describe the data we collected to evaluate this model. We recruited 25 Yellow Cab taxi drivers to collect data from. Their experience as taxi drivers in the city ranged from 1 month to 40 years. The average and median were 12.9 years and 9 years respectively. All participants reported living in the area for at least 15 years.

Collected Position Data

We collected location traces from our study participants over a 3 month period using global positioning system (GPS) devices that log locations over time. Each participant was provided one of these devices⁴, which records a reading roughly every 6 to 10 seconds while in motion. The data collection yielded a dataset of over 100,000 miles of travel collected from over 3,000 hours of driving. It covers a large area surrounding our city (Figure 4). Note that no map is being overlaid in this figure. Repeated travel over the same roads leaves the appearance of the road network itself.

⁴In a few cases where two participants who shared a taxi also shared a GPS logging device

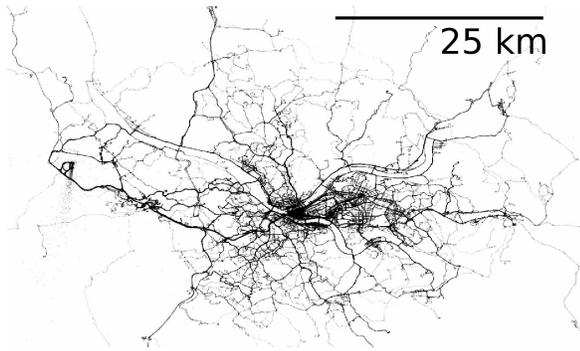


Figure 4. The collected GPS datapoints

Road Network Representation

The deterministic action-state representation of the corresponding road network contains over 300,000 states (*i.e.*, road segments) and over 900,000 actions (*i.e.*, available transitions between road segments). There are characteristics describing the speed categorization, functionality categorization, and lane categorization of roads. Additionally we can use geographic information to obtain road lengths and turn types at intersection (*e.g.*, straight, right, left).

Fitting to the Road Network and Segmenting

To address noise in the GPS data, we fit it to the road network using a particle filter. A particle filter simulates a large number of vehicles traversing over the road network, focusing its attention on particles that best match the GPS readings. A motion model is employed to simulate the movement of the vehicle and an observation model is employed to express the relationship between the true location of the vehicle and the GPS reading of the vehicle. We use a motion model based on the empirical distribution of changes in speed and a Laplace distribution for our observation model.

Once fitted to the road network, we segmented our GPS traces into distinct trips. Our segmentation is based on time-thresholds. Vehicles with a small velocity for a period of time are considered to be at the end of one trip and the beginning of a new trip. We note that this problem is particularly difficult for taxi driver data, because these drivers may often stop only long enough to let out a passenger and this can be difficult to distinguish from stopping at a long stoplight. We discard trips that are too short, too noisy, and too cyclic.

MODELING ROUTE PREFERENCES

In this section, we describe the features that we employed to model the utilities of different road segments in our model, and machine learning techniques employed to learn good cost weights for the model.

Feature Sets and Context-Awareness

As mentioned, we have characteristics of the road network that describe speed, functionality, lanes, and turns. These combine to form path-level features that describe a path as numbers of different turn type along the path and road mileage at different speed categories, functionality categories, and numbers of lanes. To form these path features,

we combine the comprising road segments' features for each of their characteristics weighted by the road segment length or intersection transition count.

Feature	Value	Feature	Value
Highway	3.3 miles	Hard left turn	1
Major Streets	2.0 miles	Soft left turn	3
Local Streets	0.3 miles	Soft right turn	5
Above 55mph	4.0 miles	Hard right turn	0
35-54mph	1.1 miles	No turn	25
25-34 mph	0.5 miles	U-turn	0
Below 24mph	0 miles		
3+ Lanes	0.5 miles		
2 Lanes	3.3 miles		
1 Lane	1.8 miles		

Table 3. Example feature counts for a driver's demonstrated route(s)

The PROCAB model finds the cost weight for different features so that the model's feature counts will match (in expectation) those demonstrated by a driver (*e.g.*, as shown in Table 3). when planning for the same starting point and destination. Additionally, unobserved features may make certain road segments more or less desirable. To model these unobservable features, we add unique features associated with each road segment to our model. This allows the cost of each road segment to vary independently.

We conducted a survey of our taxi drivers to help identify the main contextual factors that impact their route choices. The perceived influences (with average response) on a 5-point Likert Scale ranging from "no influence" (1) to "strong influence" (5) are: *Accidents* (4.50), *Construction* (4.42), *Time of Day* (4.31), *Sporting Events* (4.27), and *Weather* (3.62). We model some of these influences by adding real-time features to our road segments for *Accident*, *Congestion*, and *Closures* (Road and Lane) according to traffic data collected every 15 minutes from Yahoo's traffic service.

We incorporate sensitivity to time of day and day of week by adding duplicate features that are only active during certain times of day and days of week. We use *morning rush hour*, *day time*, *evening rush hour*, *evening*, and *night* as our time of day categories, and *weekday* and *weekend* as day of week categories. Using these features, the model tries to not only match the total number of *e.g.*, interstate miles, but it also tries to get the right amount of interstate miles under each time of day category. For example, if our taxi drivers try to avoid the interstate during rush hour, the PROCAB model assigns a higher weight to the joint interstate and rush hour feature. It is then less likely to prefer routes on the interstate during rush hour given that higher weight.

Matching all possible contextual feature counts highly constrains the model's predictions. In fact, given enough contextual features, the model may exactly predict the actual demonstrated behavior and *overfit* to the training data. We avoid this problem using *regularization*, a technique that relaxes the feature matching constraint by introducing a penalty term ($-\sum_i \lambda_i \theta_i^2$) to the optimization. This penalty

prevents cost weights corresponding to highly specialized features from becoming large enough to force the model to perfectly fit observed behavior.

Learned Cost Weights

We learn the cost weights that best explain a set of demonstrated routes using Algorithm 2. Using this approach, we can obtain cost weights for each driver or a collective cost weight for a group of drivers. In this paper, we group the routes gathered from all 25 taxi drivers together and learn a single cost weight using a training set of 80% of those routes.

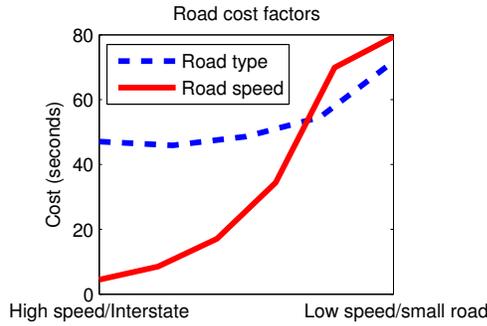


Figure 5. Speed categorization and road type cost factors normalized to seconds assuming 65mph driving on fastest and largest roads

Figure 5 shows how road type and speed categorization influence the road’s cost in our learned model.

NAVIGATION APPLICATIONS AND EVALUATION

We now focus on the applications that our route preference model enables. We evaluate our model on a number of prediction tasks needed for those applications. We compare the PROCAB model’s performance with other state of the art methods on these tasks using the remaining 20% of the taxi route dataset to evaluate.

Turn Prediction

We first consider the problem of predicting the action at the next intersection given a known final destination. This problem is important for applications such as automatic turn signaling and fuel consumption optimization when the destination is either specified by the driver or can be inferred from his or her normal driving habits. We compare PROCAB’s ability to predict 55,725 decisions at intersections with multiple options⁵ to approaches based on Markov Models.

We measure the accuracy of each model’s predictions (*i.e.*, percentage of predictions that are correct) and the average log likelihood, $\frac{1}{\#decisions} \sum_{decision\ d} \log P_{model}(d)$ of the actual actions in the model. This latter metric evaluates the model’s ability to probabilistically model the distribution of decisions. Values closer to 0 are closer to perfectly modeling the data. As a baseline, guessing uniformly at random (without U-turns) yields an accuracy of 46.4% and a log likelihood of -0.781 for our dataset.

⁵Previous Markov Model evaluations include “intersections” with only one available decision, which comprise 95% [22] and 28% [11] of decisions depending on the road network representation.

Markov Models for Turn Prediction

We implemented two Markov Models for turn prediction. A Markov Model predicts the next action (or road segment) given some observed conditioning variables. Its predictions are obtained by considering a set of previously observed decisions at an intersection that match the conditioning variables. The most common action from that set can be predicted or the probability of each action can be set in proportion to its frequency in the set⁶. For example, Liao *et al.* [14] employ the destination and mode of transport for modeling multi-modal transportation routines. We evaluate the Markov Models employed by Krumm [11], which conditions on the previous K traversed road segments, and Simmons *et al.* [22], which conditions on the route destination.

History	Non-reducing		Reducing	
	Accu.	Likel.	Accu.	Likel.
1 edge	85.8%	-0.322	85.8%	-0.322
2 edge	85.6%	-0.321	86.0%	-0.319
5 edge	84.8%	-0.330	86.2%	-0.321
10 edge	83.4%	-0.347	86.1%	-0.328
20 edge	81.5%	-0.367	85.7%	-0.337
100 edge	79.3%	-0.399	85.1%	-0.356

Table 4. K -order Markov Model Performance

Evaluation results of a K -order Markov Model based on road segment histories are shown in Table 4. We consider two variants of the model. For the *Non-reducing* variant, if there are no previously observed decisions that match the K -sized history, a random guess is made. In the *Reducing* variant, instead of guessing randomly when no matching histories are found, the model tries to match a smaller history size until some matching examples are found. If no matching histories exist for $K = 1$ (*i.e.*, no previous experience with this decision), a random guess is then made.

In the non-reducing model we notice a performance degradation as the history size, K , increases. The reduction strategy for history matching helps to greatly diminish the degradation of having a larger history, but still we still find a small performance degradation as the history size increases beyond 5 edges.

Destinations	Accuracy	Likelihood
1x1 grid	85.8%	-0.322
2x2 grid	85.1%	-0.319
5x5 grid	84.1%	-0.327
10x10 grid	83.5%	-0.327
40x40 grid	78.6%	-0.365
100x100 grid	73.1%	-0.416
2000x2000 grid	59.3%	-0.551

Table 5. Destination Markov Model Performance

We present the evaluation of the Markov Model conditioned on a known destination in Table 5. We approximate each unique destination with a grid of destination cells, starting from a single cell (1x1) covering the entire map, all the way

⁶Some chance of selecting previously untaken actions is added by *smoothing* the distribution.

up to 4 million cells (2000x2000). As the grid dimensions grow to infinity, the treats each destination as unique. We find empirically that using finer grid cells actually degrades accuracy, and knowing the destination provides no advantage over the history-based Markov Model for this task.

Both of these results show the inherent problem of *data sparsity* for directed graphical models in these types of domains. With an infinite amount of previously observed data available to construct a model, having more information (*i.e.*, a larger history or a finer grid resolution) can never degrade the model’s performance. However, with finite amounts of data there will often be few or no examples with matching conditional variables, providing a poor estimate of the true conditional distribution, which leads to lower performance in applications of the model.

PROCAB Turn Prediction

The PROCAB model predicts turns by reasoning about paths to the destination. Each path has some probability within the model and many different paths share the same first actions. An action’s probability is obtained by summing up all path probabilities that start with that action. The PROCAB model provides a compact representation over destination, context, and action sequences, so the exact destination and rich contextual information can be incorporated without leading to data sparsity degradations like the Markov Model.

	Accuracy	Likelihood
Random guess	46.4%	-0.781
Best History MM	86.2%	-0.319
Best Destination MM	85.8%	-0.319
PROCAB (no context)	91.0%	-0.240
PROCAB (context)	93.2%	-0.201

Table 6. Baseline and PROCAB turn prediction performance

We summarize the best comparison models’ results for turn prediction and present the PROCAB turn prediction performance in Table 6. The PROCAB approach provides a large improvement over the other models in both prediction accuracy and log likelihood. Additionally, incorporating time of day, day of week, and traffic report contextual information provides improved performance. We include this additional contextual information in the remainder of our experiments.

Route Prediction

We now focus on route prediction, where the origin and destination of a route are known, but the route between the two is not and must be predicted. Two important applications of this problem are route recommendation, where a driver requests a desirable route connecting two specified points, and unanticipated hazard warning, where an application that can predict the driver will encounter some hazard he is unaware of and warn him beforehand.

We evaluate the prediction quality based on the amount of matching distance between the predicted route and the actual route, and the percentage of predictions that match, where we consider all routes that share 90% of distance as

Model	Dist. Match	90% Match
Markov (1x1)	62.4%	30.1%
Markov (3x3)	62.5%	30.1%
Markov (5x5)	62.5%	29.9%
Markov (10x10)	62.4%	29.6%
Markov (30x30)	62.2%	29.4%
Travel Time	72.5%	44.0%
PROCAB	82.6%	61.0%

Table 7. Evaluation results for Markov Model with various grid sizes, time-based model, and PROCAB model

matches. This final measure ignores minor route differences, like those caused by noise in GPS data. We evaluate a previously described Markov Model, a model based on estimated travel time, and our PROCAB model.

Markov Model Route Planning

We employ route planning using the previously described destination-conditioned Markov Model [22]. The model recommends the most probable route satisfying origin and destination constraints. The results (Table 7, *Markov*) are fairly uniform regardless of the number of grid cells employed, though there is a subtle degradation with more grid cells.

Travel Time-Based Planning

A number of approaches for vehicle route recommendation are based on estimating the travel time of each route and recommending the fastest route. Commercial route recommendation systems, for example, try to provide the fastest route. The Cartel Project [8] works to better estimate the travel times of different road segments in near real-time using fleets of instrumented vehicles. One approach to route prediction is to assume the driver will also try to take this most expedient route.

We use the distance and speed categorization of each road segment to estimate travel times, and then provide route predictions using the fastest route between origin and destination. The results (Table 7, *Travel Time*) show a large improvement over the Markov Model. We believe this is due to data sparsity in the Markov Model and the Travel time model’s ability to generalize to previously unseen situations (*e.g.*, new origin-destination pairs).

PROCAB Route Prediction

Our view of route prediction and recommendation is fundamentally different than those based solely on travel time estimates. Earlier research [13] and our own study have shown that there is a great deal of variability in route preference between drivers. Rather than assume drivers are trying to optimize one particular metric and more accurately estimate that metric in the road network, we implicitly learn the metric that the driver is actually trying to optimize in practice. This allows other factors on route choice, such as fuel efficiency, safety, reduced stress, and familiarity, to be modeled. While the model does not explicitly understand that one route is more stressful than another, it does learn to imitate a driver’s avoidance of more stressful routes and features associated with those routes. The PROCAB model provides

Model	Percentage of trip observed					
	10%	20%	40%	60%	80%	90%
MM 5x5	9.61	9.24	8.75	8.65	8.34	8.17
MM 10x10	9.31	8.50	7.91	7.58	7.09	6.74
MM 20x20	9.69	8.99	8.23	7.66	6.94	6.40
MM 30x30	10.5	9.94	9.14	8.66	7.95	7.59
Pre 40x40 MAP	7.98	7.74	6.10	5.42	4.59	4.24
Pre 40x40 Mean	7.74	7.53	5.77	4.83	4.12	3.79
Pre 80x80 MAP	11.02	7.26	5.17	4.86	4.21	3.88
Pre 80x80 Mean	8.69	7.27	5.28	4.46	3.95	3.69
PRO MAP	11.18	8.63	6.63	5.44	3.99	3.13
PRO Mean	6.70	5.81	5.01	4.70	3.98	3.32

Table 8. Prediction error of Markov, Predestination, and PROCAB models in kilometers

increased performance over both the Markov Model and the model based on travel time estimates, as shown in Table 7. This is because it is able to better estimate the utilities of the road segments than the feature-based travel time estimates.

Destination Prediction

Finally, we evaluate models for destination prediction. In situations where a driver has not entered her destination into an in-car navigation system, accurately predicting her destination would be useful for proactively providing an alternative route. Given the difficulty that users have in entering destinations [23], destination prediction can be particularly useful. It can also be used in conjunction with our model’s route prediction and turn prediction to deal with settings where the destination is unknown.

In this setting, a partial route of the driver is observed and the final destination of the driver is predicted. This application is especially difficult given our set of drivers, who visit a much wider range of destinations than typical drivers. We compare PROCAB’s ability to predict destination to two other models, in settings where the set of possible destinations is not fully known beforehand.

We evaluate our models using 1881 withheld routes and allow our model to observe various amounts of the route from 10% to 90%. The model is provided no knowledge of how much of the trip has been completed. Each model provides a single point estimate for the location of the intended destination and we evaluate the distance between the true destination and this predicted destination in kilometers.

Bayes’ Rule

Using Bayes’ Rule (Equation 6), route preference models that predict route (B) given destination (A) can be employed along with a prior on destinations, $P(A)$, to obtain a probabilistic distribution over destinations, $P(A|B)$ ⁷.

$$P(A|B) = \frac{P(B|A)P(A)}{\sum_{A'} P(B|A')P(A')} \propto P(B|A)P(A) \quad (6)$$

⁷Since the denominator is constant with respect to A, the probability is often expressed as being proportionate to the numerator.

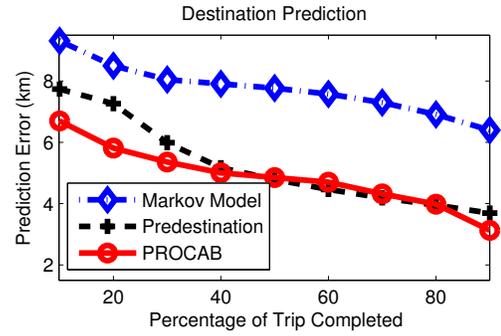


Figure 6. The best Markov Model, Predestination, and PROCAB prediction errors

Markov Model Destination Prediction

We first evaluate a destination-conditioned Markov Model for predicting destination region in a grid. The model employs Bayes’ Rule to obtain a distribution over cells for the destination based on the observed partial route. The prior distribution over grid cells is obtained from the empirical distribution of destinations in the training dataset. The center point of the most probable grid cell is used as a destination location estimate.

Evaluation results for various grid cell sizes are shown in Table 8 (MM). We find that as more of the driver’s route is observed, the destination is more accurately predicted. As with the other applications of this model, we note that having the most grid cells does not provide the best model due to data sparsity issues.

Predestination

The Predestination system [12] grids the world into a number of cells and uses the observation that the partially traveled route is usually an *efficient* path through the grid to the final destination. Using Bayes’ rule, destinations that are opposite of the direction of travel have much lower probability than destinations for which the partially traveled route is efficient. Our implementation employs a prior distribution over destination grid cells conditioned on the starting location rather than the more detailed original Predestination prior. We consider two variants of prediction with Predestination. One predicts the center of the most probable cell (*i.e.*, the *Maximum a Posteriori* or *MAP* estimate). The other, *Mean*, predicts the probabilistic average over cell center beliefs. We find that Predestination shows significant improvement (Table 8, Pre) over the Markov Model’s performance.

PROCAB Destination Prediction

In the PROCAB model, destination prediction is also an application of Bayes’ rule. Consider a *partial path*, $\zeta_{A \rightarrow B}$ from point A to point B. The destination probability is then:

$$P(\text{dest}|\zeta_{A \rightarrow B}, \theta) \propto P(\zeta_{A \rightarrow B}|\text{dest}, \theta)P(\text{dest}) \propto \frac{\sum_{\zeta_{B \rightarrow \text{dest}}} e^{-\text{cost}(\zeta|\theta)}}{\sum_{\zeta_{A \rightarrow \text{dest}}} e^{-\text{cost}(\zeta|\theta)}} P(\text{dest}) \quad (7)$$

We use the same prior that depends on the starting point (A) that we employed in our Predestination implementation. The

posterior probabilities are efficiently computed by taking the sums over paths from points A and B to each possible destination (Equation 7) using the forward pass of Algorithm 1.

Table 8 (PRO) and Figure 6 show the accuracy of PROCAB destination prediction compared with other models. Using averaging is much more beneficial for the PROCAB model, likely because it is more sensitive to particular road segments than models based on grid cells. We find that the PROCAB model performs comparably to the Predestination model given a large percentage of observed trip, and better than the Predestination model when less of the trip is observed. PROCAB's abilities to learn non-travel time desirability metrics, reason about road segments rather than grid cells, and incorporate additional contextual information may each contribute to this improvement.

CONCLUSIONS AND FUTURE WORK

We presented PROCAB, a novel approach for modeling observed behavior that learns context-sensitive action utilities rather than directly learning action sequences. We applied this approach to vehicle route preference modeling and showed significant performance improvements over existing models for turn and route prediction and showed similar performance to the state-of-the-art destination prediction model. Our model has a more compact representation than those which directly model action sequences. We attribute our model's improvements to its compact representation, because of an increased ability to generalize to new situations.

Though we focused on vehicle route preference modeling and smart navigation applications, PROCAB is a general approach for modeling context-sensitive behavior that we believe it is well-suited for a number of application domains. In addition to improving our route preference models by incorporating additional contextual data, we plan to model elder drivers and other interesting populations and we plan to model behavior in other context-rich domains, including modeling the movements of people throughout dynamic work environments and the location-dependent activities of families. We hope to demonstrate PROCAB as a domain-general framework for modeling context-dependent user behavior.

ACKNOWLEDGMENTS

The authors thank Eric Oatneal and Jerry Campolongo of Yellow Cab Pittsburgh for their assistance, Ellie Lin Ratliff for helping to conduct the study of driving habits, and John Krumm for his help in improving this paper and sharing GPS hardware hacks. Finally, we thank the Quality of Life Technology Center/National Science Foundation for supporting this research under Grant No. EEEEC-0540865.

REFERENCES

1. P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proc. ICML*, pages 1–8, 2004.
2. D. Ashbrook and T. Starner. Using gps to learn significant locations and prediction movement. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.
3. A. Bhattacharya and S. K. Das. Lezi-update: An information-theoretic framework for personal mobility tracking in pcs networks. *Wireless Networks*, 8:121–135, 2002.
4. T. Bohnenberger, A. Jameson, A. Krüger, and A. Butz. Location-aware shopping assistance: Evaluation of a decision-theoretic approach. In *Proc. Mobile HCI*, pages 155–169, 2002.
5. K. Chang, M. Y. Chen, and J. Canny. Tracking free-weight exercises. In *Proc. Ubicomp*, pages 19–37, 2007.
6. J. Fogarty, S. E. Hudson, C. G. Atkeson, D. Avrahami, J. Forlizzi, S. B. Kiesler, J. C. Lee, and J. Yang. Predicting human interruptibility with sensors. *ACM Transactions on Computer-Human Interaction*, 12(1):119–146, 2005.
7. D. A. Hennesy and D. L. Wiesenhal. Traffic congestion, driver stress, and driver aggression. *Aggressive Behavior*, 25:409–423, 1999.
8. B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. K. Miu, E. Shih, H. Balakrishnan, and S. Madden. CarTel: A Distributed Mobile Sensor Computing System. In *ACM SenSys*, pages 125–138, 2006.
9. E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106:620–630, 1957.
10. L. Johannesson, M. Asbogard, and B. Egardt. Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming. *IEEE Transactions on Intelligent Transportation Systems*, 8(1):71–83, March 2007.
11. J. Krumm. A markov model for driver route prediction. *Society of Automotive Engineers (SAE) World Congress*, 2007.
12. J. Krumm and E. Horvitz. Predestination: Inferring destinations from partial trajectories. In *Proc. Ubicomp*, pages 243–260, 2006.
13. J. Letchner, J. Krumm, and E. Horvitz. Trip router with individualized preferences (trip): Incorporating personalization into route planning. In *Proc. IAAI*, pages 1795–1800, 2006.
14. L. Liao, D. J. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5-6):311–331, 2007.
15. R. M. Mayrhofer. *An architecture for context prediction*. PhD thesis, Johannes Kepler University of Linz, Austria, 2004.
16. M. C. Mozer. The neural network house: An environment that adapts to its inhabitants. In *AAAI Spring Symposium*, pages 110–114, 1998.
17. B. Mutlu, A. Krause, J. Forlizzi, C. Guestrin, and J. Hodgins. Robust, low-cost, non-intrusive sensing and recognition of seated postures. In *Proc. UIST*, pages 149–158, 2007.
18. A. Y. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *Proc. ICML*, pages 663–670, 2000.
19. K. Patel, M. Y. Chen, I. Smith, and J. A. Landay. Personalizing routes. In *Proc. UIST*, pages 187–190, 2006.
20. D. J. Patterson, L. Liao, D. Fox, and H. A. Kautz. Inferring high-level behavior from low-level sensors. In *Proc. Ubicomp*, pages 73–89, 2003.
21. M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 1994.
22. R. Simmons, B. Browning, Y. Zhang, and V. Sadekar. Learning to predict driver route and destination intent. *Proc. Intelligent Transportation Systems Conference*, pages 127–132, 2006.
23. A. Steinfeld, D. Manes, P. Green, and D. Hunter. Destination entry and retrieval with the ali-scout navigation system. *The University of Michigan Transportation Research Institute No. UMTRI-96-30*, 1996.
24. E. M. Tapia, S. S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. In *Proc. Pervasive*, pages 158–175, 2004.
25. K. Torkkola, K. Zhang, H. Li, H. Zhang, C. Schreiner, and M. Gardner. Traffic advisories based on route prediction. In *Workshop on Mobile Interaction with the Real World*, 2007.
26. D. Wyatt, M. Philipose, and T. Choudhury. Unsupervised activity recognition using automatically mined common sense. In *Proc. AAI*, pages 21–27, 2005.
27. B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *Proc. AAI*, 2008.