Learning Patterns of Pick-ups and Drop-offs to Support Busy Family Coordination

Scott Davidoff, Brian D. Ziebart, John Zimmerman, Anind K. Dey
Carnegie Mellon University
{ scott.davidoff | bziebart | johnz | anind } @ cs.cmu.edu

ABSTRACT

Part of being a parent is taking responsibility for arranging and supplying transportation of children between various events. Dual-income parents frequently develop routines to help manage transportation with a minimal amount of attention. On days when families deviate from their routines, effective logistics can often depend on knowledge of the routine location, availability and intentions of other family members. Since most families rarely document their routine activities, making that needed information unavailable, coordination breakdowns are much more likely to occur. To address this problem we demonstrate the feasibility of learning family routines using mobile phone GPS. We describe how we (1) detect pick-ups and drop-offs; (2) predict which parent will perform a future pick-up or drop-off; and (3) infer if a child will be left at an activity. We discuss how these routine models give digital calendars, reminder and location systems new capabilities to help prevent breakdowns, and improve family life.

Author Keywords
Sensing, machine learning, data mining, statistical models, graphical models, GPS, mobile phones, routines.

ACM Classification Keywords
H.m Information Systems: Miscellaneous.

General Terms
Human Factors, Algorithms.

INTRODUCTION

Parents are often responsible for planning, coordinating, and executing the transportation of their children to and from their many activities. One coping strategy dual-income parents employ is to develop routines. As parents repeatedly perform similar sequences of actions around each pick-up and drop-off, a routine emerges, significantly reducing the attention required to complete the task.

When tasks unfold in a routine fashion, coordination requires minimal attention to detail. However, when families must deviate from their routines – *e.g.* when one parent must travel for work, schedule an orthodontist appointment, plan a new carpool, or remain home with a sick child – the likelihood that some part of their plan will break down significantly increases.

One these days, effective logistics can often depend on knowledge of the routine location, availability and intentions of other family members. Interestingly, families rarely document routine events on their home calendars [13]. Even when they do, descriptions are incomplete, missing key information, like which parent will drive. Without a resource to provide needed information on routines, family members must recall details of other members' routines (or their own), and here errors can take place. Plans for new events or reactions to unanticipated situations can result in double-booked schedules, missed events, and even children being left at events [10, 11], and, or course, stress and anxiety for parents [19, 37].

This paper explores how sensing and modeling can provide computational access to family transportation routines, and how these learned models function as an *enabling technology*. Specifically, we demonstrate how location data from standard mobile phones can be used to:

1. Detect if a pick-up or drop-off has occurred
2. Predict which parent has responsibility for a future pick-up or drop-off
3. Infer if a child will forgotten at an event

We also discuss how these learned models can create new resources that enable end-user applications, for example:

1. Awareness systems can know when pick-ups occur
2. Calendars can display implicit routine data like where and when pick-ups and drop-offs will occur
3. Location systems can show what pick-up and drop-off responsibilities family members have
4. Reminder systems can alert parents about children left at activities without explicit reminder creation

In this paper, we describe the design and evaluation of our learned models of family routine. Our goal is to develop a proof-of-concept while providing insights on how to improve technical performance. We provide an overview of our previous research and situate our contribution within the field; we describe the design and performance of our learning systems; and we discuss how learned models can enable end-user applications, and ultimately, families.

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.
Copyright 2011 ACM 978-1-4503-0267-8/11/05....$10.00.

PREVIOUS RESEARCH

Our focus on routine models emerges from three years of fieldwork on the needs and values of dual-income families.

While many domestic research projects envision technology to provide people more control over home devices, our first yearlong ethnography of dual-income families showed that *more than control of their devices, families desire more control of their lives* [11]. Next, a year Speed Dating design concepts highlighted that parents' often feel seriously out of control as they decide how to transport their children to and from their activities [12]. A third yearlong field study to understand how families manage transportation illustrated how and why coordination breakdowns occur [13].

These breakdowns cut across families, parents, coordination styles and economic class. Parents would commit their spouses to tasks they could not do given the spouse's current routines. Even when known in advance, we saw parents go to a routine place to pick-up a child when the child was at another location. We family members neglect to inform one another when their routines changed. Finally, on multiple occasions, we witnessed well adjusted, responsible parents forget to pick-up their children, leaving them at their activity locations for as long as 40 minutes.

Routines are at once central to many coordination problems, and at the same time not documented (on calendars or elsewhere), so unavailable to family members and coordination support applications.

Since family members currently do not record their routine information, we developed a vision where learned models of routine could create new resources to help families with:

1. **Planning** – Using our models, calendaring systems like LINC [27] and DateLens [29] could display implicit routine events, event times, pick-ups and drop-offs that parents don't document but are critical to making effective plans for and that affect others. Family members could see an entire day's plan, helping make unresolved and conflicting responsibilities salient.
2. **Coordinating** – Using our models, location systems like Motion Presence [6] and the Whereabouts Clock [8] could display where people are with implicit routine information like future pick-ups and drop-offs, helping remind family members what they need to do, and alerting others as days happen as planned.
3. **Improvising** – Using our models, applications could update plans in real-time, helping parents make more reliable decisions. As pick-ups happen, reminder systems like comMotion [25] and PlaceMail [24] could suppress potentially annoying reminders.
4. **Protecting** – Using our models, reminder systems could infer that a required pick-up is not happening, and remind parents without their explicit creation of a reminder, creating a new kind of safety net to guard families against this uncommon but stressful outcome.

BACKGROUND

Existing research has highlighted the value of routines in support of everyday living, while observing limits to that support when people depart from their routines.

Many successful routines serve important organizational and social purposes. A form of habituated memory, routines free up people's attention, creating a tacit knowledge that constrains and directs activity [26]. Successful routines help to scaffold and streamline the decision-making process [38]. However, as their performance becomes more automatic, people cannot recall the details of what they do, making routines difficult to transfer between people [34]. Also, since people often default to routines when faced with limited information, deviations from routine (planned and unanticipated) can cause plans to break down [32].

Dual-income families, in particular, rely heavily on routines to support their dynamic schedules [10, 37], providing them with a feeling of control (*cf.* Bandura [3]) over the environment [38]. But because dual-income families also experience many deviations from routine – *e.g.* rained out games, holidays, scheduled school closings and half-days, doctor and dental checkups, forgotten items and sick children – their plans are often susceptible to breakdowns [5, 10], leading to increased stress [37]. Even the thought that a non-routine day might occur, such as a child showing signs of getting sick, can cause parents stress [10, 37].

During non-routine days, information about the routines of others would often help improvise appropriate responses. Despite this value, routines themselves rarely are documented – they do not appear on calendars [14]. Our work explores the use of machine learning and data mining to learn routines in order to make this information available to families and to family support systems.

The routine is emerging as a useful computational abstraction across a variety of domains. It has been used, for example, to improve the classification of domestic activities [21, 35] and to create opportunities for workplace communication [5]. GPS has provided a gateway to many of those capabilities. Researchers have used GPS data to model geographic mobility [20], social networks [15], individual one-step destination [22, 39], and navigation for the cognitively impaired [23].

Our work builds upon these findings, extending them into the complex and highly nuanced context of family coordination. We model a family as a group of collaborating dependents, not independent individuals (*e.g.* [20, 22, 23, 25]), With multiple people, we can introduce models of collaborative family goals, like when parents pick up and drop off children, and whether a pick-up might be forgotten. Currently, the only extant notification mechanism we know for forgotten children arrives as phone calls from impatient day care managers [19], embarrassed friends [10], or irritated spouses [11]. Notification of this event in any form would present a significant contribution to family life, and create a new kind of safety net.

APPROACH

The goal of this paper is to provide an initial demonstration of the feasibility of our underlying technical approach in a realistic setting. It is also important to note that we in no way claim that our approach is optimal. Instead, the reader should consider the approach in this paper a demonstration of what is possible, and food for thought about other capabilities that knowledge of routines can offer.

First, to operate with a shared vocabulary, we offer a set of definitions. Since we are relying on GPS as our primary sensor for learning routines, we developed a place-centric view of coordination. A person's day can be described as an ordered list of the places they go. We call the transition between each place a *ride*. In our families, each ride has a *driver* (the parent) and possibly a *passenger* (the child). A *drop-off* is a ride given to a place, and a *pick-up* is a ride given from a place. A *plan* is an ordered list of pick-ups and drop-offs, each with an *ideal time*, when the parent intends it to occur, and an *actual time*, when it actually does occur. As a plan unfolds, the family *coordinates*, acknowledging completed rides, and reflecting on the need to modify their current plan for future rides. Families *improvise*; they dynamically modify and even generate new plans based on unanticipated situations that challenge the current plan. Collectively, we refer to planning, coordinating and improvising as family *logistics* (also known as coordination in other literature). Children participate in *activities* like violin and swimming lessons Parents give children rides to *events*, which are instances of the activities.

With this shared vocabulary, the remainder of this section describes the construction of three models of routine, their relationships, and how they can help families plan, coordinate and improvise; creating a new kind of protection against logistical breakdowns. Figure 1 depicts our modeling approach graphically, with arrows indicating the flow of data and inference. Models are constructed from two data sources: interviews, and GPS (Figure 1, top). Our GPS data set follows the movements of 23 people in 5 dual-income families across approximately six months, and is labeled with data collected from nearly 1,000 surveys, over 500 phone interviews, and over 100 home visits (collection reported in [13]). Leveraging previous work on automated place discovery [1, 25], we assume that all models include a set of known locations (taken from our interviews). Interviews provide a ground truth about which activities occur each day, and the ideal pick-up and drop-off times. We combined this information with GPS, and designed a series of three models, each recombining and building on the output of the previous model, and each supporting family logistics in different ways.

First, we apply a temporal logic to our GPS data streams, and compare the location and driving state of separate individuals to *recognize when rides occur and who participated in the rides*. Participation allows us infer which parent drives which kid to which activity. In the most basic sense, this is part of the routine that families almost never record on their calendar. The output from this model can begin to fill in the empty calendar spaces for the things people do every day.

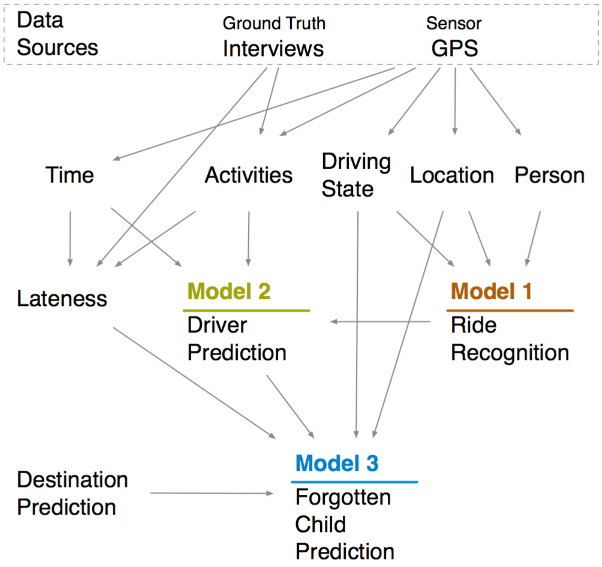

Figure 1. Interviews provide each activity's location and each event's intended pick-up time. All other knowledge is inferred from low-level GPS, creating unsupervised models that (1) sense rides; (2) predict the driver for the next event; and (3) predict if parents will arrive late for pick-ups. End-user coordination applications can use any model's output as a new data source to help families plan, coordinate and improvise. Model 3 can work with reminder svstems to create a new kind of safetv net.

Second, the ride recognition model provides seed data to learn the probability distribution that each parent will be driving a child to an activity (Figure 1, Model 2). We can use this distribution to *predict future drivers*, which, when compared with observations in real time, can offer an indicator of when events are happening in non-routine ways [31]. Because non-routine events are those most likely to lead to logistical breakdowns, early detection of non-routine rides can help propagate that information to coordinating family members, and, when appropriate, allow them to respond earlier, and to make plans in response to these situations with a more complete overall picture.

Third, the driver prediction model, along with real-time location and driving state, a learned distribution on lateness, and a driver destination model, feeds into a higher-level inference machine to detect when *parents forget a child* at an activity (Figure 1, Model 3).

We demonstrate that our approach is practical by conducting this work under legitimate sensor and modeling constraints. Since families literally rely on their phones, battery-heavy GPS sampling is limited to once per minute instead of the more common rates upwards of once per second. Models are then trained using an online approach, meaning we use only the data that would be available at any given point in time, and not the entire data set.

MODEL 1: RECOGNIZING RIDES

The routine rides family members take every day are one of the basic units of undocumented family logistics. The automatic capture of rides introduces the possibility of computational support while requiring minimal behavioral changes. Each sensed ride includes a driver, a passenger, a place and a time, providing information that can be immediately shared with calendaring and awareness systems, as well as creating a source of labeled data with which computing systems can perform further reasoning. In this section, we describe a method for recognizing rides and examine its performance.

How are rides recognized?

To recognize when rides occur, we apply a simple temporal logic to the synchronized, discretized GPS data. We define three states for a person (see Eq. 1). A person is always either at a location, L_n, or traveling, T, which we define as the unique location occupying the space between all known places. Two family members are said to be co-traveling, *CoT*, if during the same time, t_i they are both traveling and are located at a Euclidian distance of less than 500 meters apart. Any remaining states (*e.g.*, no sample, outliers) are collectively labeled *else*.

$$States = \left\{ L_n, T \mid CoT, else \right\} \qquad (1)$$

We examine each parent-child combination separately, and refer to parent as *P*, and child as *C* (see Eq. 2)

$$People = \left\{ P, C \right\} \qquad (2)$$

We define a pick-up as the conjunction of states of a parent, *P*, and a child, *C, over time* (See Eq. 3). At time t_1, the child is at location L_n. At time t_2 both parent and child are at location L_n. Lastly, at time t_3, both parent and child are co-traveling, *CoT*. This definition is broad enough to cover two cases of pick-ups. In the majority of cases, a parent comes from a different location to the child's location and they drive off together. We also consider a pick-up in the case where a parent is already at the child's location, and they depart together. Drop-offs are the inverse of the sequence for pick-ups (see Eq. 4).

$$\left.\begin{array}{ll} \text{a.} & (t_1, P, \neg CoT) \wedge (t_1, C, L_n) \wedge \\ \text{b.} & (t_2, P, L_n) \wedge (t_2, C, L_n) \wedge \\ \text{c.} & (t_3, P, CoT) \wedge (t_3, C, CoT) \end{array}\right\} \textit{Pick-up} \qquad (3)$$

$$\left.\begin{array}{ll} \text{a.} & (t_1, P, CoT) \wedge (t_1, C, CoT) \wedge \\ \text{b.} & (t_2, P, L_n) \wedge (t_2, C, L_n) \wedge \\ \text{c.} & (t_3, P, \neg CoT) \wedge (t_1, C, L_n) \end{array}\right\} \textit{Drop-off} \qquad (4)$$

Our method to detect rides is based directly on the definitions. We parse the GPS streams from all members of a family and predict that a ride is occurring for each example that meets all three requirements of our definition.

Accuracy of ride-sensing

To evaluate the performance of our ride-sensing method, we compare its predictions with the ground truth taken from our interviews.

Aggregate interviews report 3283 rides, or roughly one ride per day per family. Real-world constraints on phone use (*e.g.*, forgotten phones, dead batteries, school mandates) place 1721 beyond the reach of sensor data collection. We report on the 1562 rides accessible to sensing. Across all families, for both pick-ups and drop-offs, our simple temporal model performs well. 90.1% of the events the model identified were in fact rides (see Table 1, Precision), and 95.5% of all rides were sensed (see Table 1, Recall).

Family	***Precision***		***Recall***	
	How many of the sensed rides are right?		*How many of the total rides are sensed?*	
	Pick-up	*Drop-off*	*Pick-up*	*Drop-off*
A	.991	.987	.912	.910
B	.966	.962	.979	.981
C	.913	.824	.971	.921
D	.878	.873	.980	.944
E	.931	.684	.959	.985
Average	.936	.866	.960	.950
	.901		.955	

Table 1. Evaluation of our temporal logic model for sensing when children are picked up and dropped off.

Precision and recall measures between families are also consistent, with a variance of 0.072. This variance measure is dominated by the precision value of 0.684 for family E drop-offs, the only value below .80. Three factors appear responsible for this low drop-off precision value. First, for family E, false positive drop-offs, where a drop-off was detected but did not occur, are five times as likely to occur at home than at any other location. Second, false positive errors at home occur almost exclusively in the late afternoon. Third, false positive errors in the afternoon at home occur almost entirely between their 10-year-old daughter (D10) and Mom.

Narratives from our interviews help explain the meaning behind these numbers. In family E, D10 walks to and from school. Family E's Mom, a doctor, arrives home from some hospital shifts at around the same time as D10 arrives home from school. Our temporal logic correctly identifies that parent and child are at the same location (Eq. 4b), and not co-traveling (Eq. 4c). Our temporal logic, however, mistakes arrival at home within the same minute for co-travel (Eq. 4a). Both Mom and D10 are correctly labeled as traveling, and are within 500 meters of one another, but are *not* traveling together, causing the false positive errors.

MODEL 2: PREDICTING DRIVERS

With a capable ride recognition model providing reliable information on the driver, passenger, place and time of each ride, we can combine this information gleaned from sensed rides with low-level GPS data, derive a set of features, and use this to train a model of driver prediction.

Modeling which parent drives

To model the online probability distribution of which parent drives, at each point in time, we create a labeled set of rides using the ride detection model's output. For each row, or prediction example, we build a feature vector, *f*, containing:

Name	*Meaning*	*Values*
L_n	Location of pick-up or drop-off	Place ID
RType	Ride type	Pick-up, Drop-off
DoW	Day of week	0,1,2,3,4,5,6
ToD	Discretized time of day (15 min)	1,2,3…96
$driver_{t-j}$	Driver for the last 5 rides to L_n	Mom, Dad
ϕ	Driver distribution model	[0,1]

Table 2. Feature vector for predicted rides.

We define the driver distribution for rides of type *RType*, on day *DoW*, at location L_n, to be the following:

$$\phi = \frac{\sum_{L_n, Rtype, DoW} driver = P_1}{\sum_{L_n, Rtype, DoW} driver = P_1 + \sum_{L_n, Rtype, DoW} driver = P_2} \qquad (5)$$

For each example, we assign the label, *y*, as -1 if Dad was detected as the driver, and +1 if Mom was detected. Our goal is to create a classifier to determine the probability that each parent is driving. Because the shape of the decision boundary is unknown, we use a decision tree classifier, which performs well with both linear and non-linear boundaries. Because previous research on family routines observes frequent deviations from the intended family schedule [10, 37], before classification we chose to use local weighting to reduce the impact of these non-routine outliers [2]. For each labeled example *q, we calculate a new label d(y) as a function of the old label y and the features f:*

$$d(y)_{LW}^{q} = \frac{\sum_{i \in labeled\ data, i \neq q} y_i e^{-\sqrt{\sum_j^i \left(x_i^j - q^j\right)^2}}}{\sum_{i=q} e^{-\sqrt{\sum_j^i \left(x_i^j - q^j\right)^2}}} \qquad (6)$$

This transformation pushes the labels for deviation examples closer to 0 and non-deviation examples closer to the original -1 or 1. The closer the label to 0, the less impact it has on the accuracy of the classifier. The overall classification process combines local weighting with a binary decision tree, creating a locally weighted decision tree (LWDT) to predict which parent drives for each ride.

Driver prediction accuracy

The ride model's treatment of the output class (Mom or Dad) as a nominal variable leaves only two cases for evaluation: (1) a prediction of Mom when Mom drives or Dad when Dad drives is correct; and (2) a prediction of Mom when Dad drives or Dad when Mom drives is incorrect. With no case that can lead to false positive or false negative, we cannot evaluate using precision or recall. Instead, we look at accuracy: how many times out of the total did the classifier predict mom *or* dad correctly.

We then train the model using a sliding window, and test on the week immediately following the training period, a common technique used with time-series data [30]. Searching for an optimal window size, we vary the size of the sliding window from one to 24 weeks, and analyze the results. Across all families, when the model is trained using only one week of training data (Figure 2, top left), the model correctly predicts the driver 72.1% of the time, showing that a deployed system might be able to provide useable data after just a single week in the field. The optimal window size is four weeks, which predicts the driver correctly 87.7% of the time.

At test weeks twelve (June 1) and eighteen (July 13), nearly every graph in Figure 2 shows a steep drop in performance, correctly predicting the driver only marginally better than chance. These dips correspond to significant changes in family routines at test week eleven (May 25), when all our families transitioned from school to a new summer schedule, and at test week seventeen (July 6), when camps ended (families D, E), families vacationed (Families A, C), and summer sports ended (family B). Poor performance during these weeks is consistent with previous findings that routines are least stable during transition times [13, 37].

Changes in routine that occurred in weeks eleven and seventeen would not appear in the training data until weeks twelve and eighteen, which exactly overlap the drops in performance. Looking more closely, the changes in routine at weeks eleven and seventeen also influence the outcome of our driver model optimization.

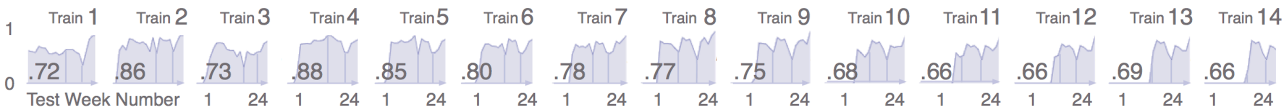

Figure 2. Accuracy (vertical axis) of our driver prediction model across all families, plotted by test week number (horizontal axis), varying the amount of training data provided to the model from one (left) to fourteen (right) weeks. No matter how much training data is used, accuracy drops at weeks 12 and 18 (vertical lines).

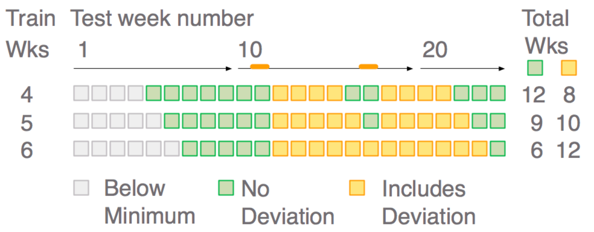

Figure 3. Performance drops of the driver prediction model at weeks 12 and 18 occur as transitions from the routines of school to into summer (week 11), and from the routines of camp to back-to-school prep (week 18) first appear in training data. Because these changepoints are separated by five weeks, models using five weeks or more of training data will include one of these aberrant weeks.

Figure 3 shows the relationship between the size of the training data sets, and the number of times they include an aberrant week. Since five weeks separate the routine changes, training data sets that include five weeks of data or more will include one of these aberrant weeks more often than not, providing them with noisier training data, and lowering their accuracy. In fact, four weeks of training data marks the turning point, when training data sets do not include aberrant weeks more often than they do, explaining why it is the optimal number of weeks of training data to use in the driver prediction model.

Lastly, despite poor performance at test weeks twelve and eighteen across the variations in size of the sliding window, performance makes a sharp turnaround in weeks thirteen and nineteen. This suggests that the model is able to make correct predictions with only one week's data from the new routine, adding support to the belief that a deployed model could recover quickly and again provide useful data even after facing changes in routine.

FORGETTING CHILDREN

In this section, we explore the feasibility of a system that can predict when a parent will forget to pick a child up from an activity at an agreed-upon time, using only GPS trace information for family members. The fear of being late and forgetting a child is a constant source of stress and anxiety for dual-income families [4, 11, 18]. Such a prediction system could create a new kind of safety net, reducing anxiety and increasing feelings of safety.

Modeling forgotten children

Forgetting is an ambiguous term. It can mean that a parent never goes to get a child, or that a parent failed to remember at the appropriate time and thus began the pick-up task later than expected. Relying on GPS for all our information, we develop a time- and place-centric definition of forgetting. We say that a parent wants to arrive at an ideal time, t_{ideal}. We say a parent forgets a pick-up when their actual arrival time, t_0, is more than ten minutes after t_{ideal} (we defer the explanation for the choice of the 10 minute threshold to the discussion). From within the set of 1562 sensed rides, we apply this rule to the 813 pick-ups, and identify 83 instances of parents forgetting pick-ups.

Building upon our earlier driver prediction model, and relying on our GPS data set, we designate the following features to help us detect these incidents of forgetting:

Name	***Meaning***	***Values***
R	Whether the parent remembers	True, False
J	Driver prediction model	Mom, Dad
T	If the parent is traveling	True, False
λ	*Empirical cumulative distribution(ecdf)* of on-time arrivals to L_{child} at time $T_{now}T_{ideal}$	[0,1]
L_{child}	Location of the child	Place ID
L_{start}	Starting location of a parent	Place ID
L_{curr}	Ending location of a parent	Place ID
D	Destination of a parent	Place ID

We assemble these features into a Bayesian Network (see Figure 4). The Bayesian network captures the complex dependencies shared by coordinating parents. Here, we use the term *dependency* as used in probability theory to mean the outcome of an event makes its dependent event more or less likely. It can be read as *influences*. Each side of the symmetrical model represents the state of one parent (Mom on the left, Dad on the right). Mutual dependencies are represented at the crossover nodes in the graph's center. For example, the location of each parent depends on (is influenced by) the location of the child, L_{child}, and whose job the pick-up is, J.

Other dependencies are mirrored for each parent. For example, according to the model, whether or not a parent is traveling, T, depends on their location, L_{start}, the location of the child, L_{child}, how often they have been late in the past, λ, and if they remember the pick-up, R. The model also shows that a parent's destination, D, depends on the location of their child, L_{child}, their current location, $L_{current}$, and if they remember the pick-up, R. The driver prediction model appears in the network as node J, predicting which parent has the job to make the pick-up. We can see that according to the network, if a parent remembers the pick-up, R depends on whether or not the pick-up is their job, J.

Starting 30 minutes before every late pick-up, we ask the network to make *two* predictions. First, we see if the non-driver *is not* going to be late. Second, we look to see if the driver *is* going to be late. We repeat these calculations at each minute until the actual time of the late pick-up, and see how early we can make the correct predictions.

We use the properties of the network structure to derive the formulas that will provide us the prediction values. To see if Dad remembers, we calculate the posterior probability R_{dad}, and then repeat the calculation for Mom.

$$P(R_d \mid L^d_{start}, L^m_{start}, L^d_{curr}, L^m_{curr}, L_{child}, T_d, T_m, \phi) \qquad (7)$$

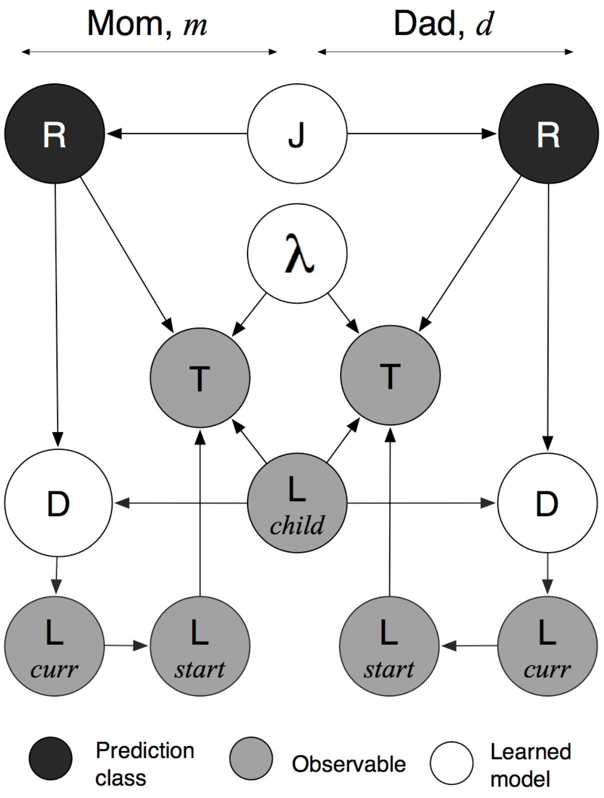

Figure 4. A Bayesian Network used to predict if a parent remembers to make a pick-up. Each side of the symmetrical model represents the state of one parent. At each point in time the model makes two predictions: (1) does the parent who drives remember the pick-up; and (2) does the other parent not remember the pick-up.

Performance of the forgotten child model

To evaluate the forgotten child model, we first learn the optimal distributions for the remaining learned model nodes – λ, and D. Using four weeks of training data for the driver prediction model, *J* (our learned optimum) we vary the amount of training data to λ and D, and compare the output. Ten weeks of data gives optimal values for both λ, and D.

To calculate the output of the entire network, we use the Maximum Entropy Inverse Optimal Control algorithm [39] to compute *D*, destination, and starting thirty minutes before each forgotten pick-up, compute values for R_{mom} and R_{dad} at one-minute intervals, until we arrive at t_{ideal}+10, ten minutes after the ideal pick-up time, when the parent is late (by our definition).

Given the unequal distribution of forgotten pick-ups (83 examples of forgotten children out of 1562 sensed rides – majority of these are parents arriving late but some are instances of parents actually forgetting their children), a model based on no data, but that simply always predicted on-time arrival would be right 0.885 of the time. Evaluation using precision and recall would output high values, but would lack a way to distinguish how much of that number comes from the excellence of the model, and how much simply from the sheer number of negative examples.

As an alternative, we evaluate the model using a technique from signal processing [30] called the receiver operating characteristic (ROC), which is a more conservative measure of performance [16]. Figure 5 shows two ROC plots. For each plot, the y-axis shows correct predictions, and the x-axis shows incorrect predictions. The points inside each square represent different ratios of correct predictions to incorrect predictions. Points along the diagonals mean the ratio is even, and the model is performing no better than a coin flip – for every one correct prediction, there is one incorrect prediction. The upper left corner is perfection, with all correct predictions and no mistakes.

We can use this property of ROC curves to visualize how a notification system would balance the desire of the parent to never miss a pick-up, against the cost of sending wrong information (*e.g.*, an unnecessary reminder). For any given number of correct predictions, an ROC curve indicates how many incorrect predictions the model will also produce. At t_{ideal}-30 (Figure 4, right, dark pink line), for every 6 correct predictions the model makes 5 incorrect predictions. When viewed as percentages, every rate of accurate predictions between 0 and 1 has a corresponding rate of inaccurate predictions between 0 and 1, producing a smooth curve. If we continue to follow this curve up and to the right, we can see that the ratio improves little. At t_{ideal} (light orange line), for every 8 correct predictions the model makes 2 incorrect predictions. At t_{ideal}+10 (dark orange line) for every 8 correct predictions the model makes only 1 incorrect prediction.

We can also measure the area beneath an ROC curve to describe a model's cumulative performance across all confidence values. At thirty minutes before t_{ideal}, the area under the ROC curve is 0.649 (Figure 4, pink area). The performance gradually rises as the pick-up time approaches, and is highest t_{ideal}+10, at 0.826.

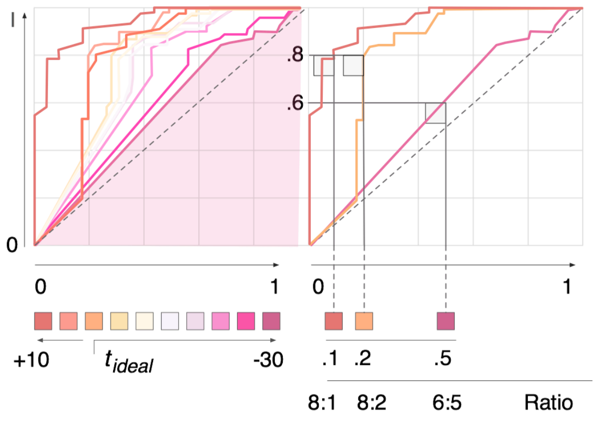

Figure 5. Forgotten child model performance. At left, ROC curves show performance improving from 0.640 at t_{ideal}-30 to 0.826 at t_{ideal}+10. At right, the values demonstrate the relationship between right and wrong guesses.

DISCUSSION

This paper describes a successful initial demonstration of the feasibility of our approach to the sensing and modeling of pick-ups and drop-offs, and hopes to spur investigation of routine as a versatile and enabling abstraction. In this section, we discuss possible directions, and identify ways to continue towards optimal performance. As a broad reaching proof-of-concept, this work required many simplifications. In this section we discuss their impact, and extensions to the work as a whole.

Building on the ride model

Our ride detection model provides ample evidence that the large scale detection of rides is within reach. Still, across all families, the model overlooks about ten percent of rides (false negatives), and makes incorrect detections (false positives) around ten percent of the time. Because our hierarchical approach to modeling means that errors in ride detection propagate to downstream models, causing further errors, improvements in ride detection will pay threefold, helping to improve driver and forgotten child prediction as well.

A variety of ways exist to improve upon our ride detection. Because we sample GPS only once per minute, pick-ups, or examples of co-location that happen on a faster time scale are simply missed. In our fieldwork, we observed busy families carrying out pick-ups and drop-offs below this detection threshold. An intelligent approach to sampling would increase rates during times when data collection would be more valuable (*i.e.*, when a person is moving), and lower the sampling rate during periods of stasis, avoiding battery depletion. This approach would require the integration of other low-level sensors into the model (*e.g.*, accelerometer, noise or light). The addition of Bluetooth could also improve co-location and co-travel accuracy by providing additional proximity cues to the model.

Missed rides are also caused by the model's simple ride representation. We defined travel in terms of one parent, one child and one car. The model has no representation of other modes of transport, causing it to miss common activities like walking, riding bicycles, and taking the bus. For simplicity we limited our observations to the nuclear family, but in reality families plan and coordinate with their extended families, friends and carpoolers, all whom were outside the scope of this investigation. The model also requires that activities occur while people are at a location. Activities like paper routes, however, occur over a wide area that contains home, neighbors, and schools, making it appear to the model as an unusual occurrence of travel.

Building on the driver prediction model

The driver prediction model learns parental responsibilities at more than 70% accuracy with only one week of training data, and about 85% accuracy with four weeks of training data. These are satisfactory numbers when seen as a technical problem. A deployed system, however, will face unknown scrutiny from coordinating parents during stressful times. Even small errors may deter adoption, encouraging examination of ways to achieve better results with less training.

While we explored a general-purpose algorithm for use across all families, we know that there is great variation in the ways families plan, coordinate and improvise. Even within families, as children grow, parents age, interests, even entire geographies, evolve. Exploration remains to find algorithms both for family style, and for attending to and integrating to the inevitable changes.

The driver prediction model also showed a remarkable potential for detecting large-scale transitions in routine. Because families experience the most stress during times of large-scale routine change, an algorithm tuned to this class of events could effectively function as a seasonal boundary detector, offering an index of routineness to other models providing services to families, helping the driver prediction model identify seasonal boundaries in training data and achieve better learning faster.

Building on the forgotten child model

Because end-user evaluation metrics will vary with application context (e.g. missed pickups for a reminder application will be less tolerable than incorrect event times in a calendar application), a precise assessment of its performance requires a field deployment. Instead, we presented an ROC analysis, which quantifies performance tradeoffs as confidence values. End-user application designers can define the behavior of systems that implement the model by calculating a future expected value based on model confidence, and the costs and benefits as defined by their domain.

Unfortunately, without a field deployment, we can only estimate costs and benefits. The ROC analysis, for example, showed that the model has about sixty percent accuracy at thirty minutes before a parent forgets a child. With about twenty actual incidents of forgetting children per family, this would equate to forty alerts over six months, or about 1.5 per week. Given the stress caused by forgetting a child, would this number of alerts constitute an annoyance, or prompt a parent to take a moment and double-check their plan and their assumptions?

This question is further complicated by the question of information delivery. It is unknown how early a forgotten pick-up needs to be detected in order to be of use. A message that comes too late to enable a parent to arrive on time might be marginally helpful, but the real goal is to make this detection early enough to ensure prevention of the event.

Improved models will also need to address the temporal complexity of coordinating parents. In some families, ten minutes late might be considered on time. In others, it might constitute abandonment. A range might more faithfully represent an ideal time than any particular minute. Also, if a parent plans to be late, then their late arrival time is their ideal time, and not the one the model assumes.

Applications of models of routine

The haphazard introduction of learning systems into the home has been a subject of concern to researchers. Their caution reflects an understanding of the technical challenges that idiosyncratic human behavior could present to modeling [32, 34], a sensitivity to the unintended consequences of technology in the home [9], a concern over the appropriate role of domestic technology [33], all which threaten to take away autonomy from families.

Ultimately, we distinguish a belief that aspects of family life can be *modeled* from the belief that family life should be *automated*. We advocate a measured approach where intelligence is applied in such a way that it does not take on the role of parent and does not conflict with the social structure within the family.

Our work does not focus on optimizing the work of families in order to allow an ever-increasing number of activities to be performed. Instead, our intention is to help family members more elegantly and effortlessly perform their role within the family by providing the resources they need to better understand what they do and what they plan to do.

Our technical models of routine support this decisively human need by acting through existing research family support systems, allowing them to better address the situations they were originally designed for. Calendaring systems like LINC [27] and DateLens [29] could display implicit routine events, event times, pick-ups and drop-offs that parents don't document but are critical to making effective plans for and that affect others. Location systems like Motion Presence [6] and the Whereabouts Clock [8] could display where people are with implicit routine information like future pick-ups and drop-offs, helping remind family members what they need to do, and alerting others as days happen as planned. Applications could update plans in real-time, helping parents make more reliable decisions. As pick-ups happen, reminder systems like comMotion [25] and PlaceMail [24] could suppress potentially annoying reminders. Reminder systems could infer that a required pick-up is not happening, and remind parents without their explicit creation of a reminder, creating a new kind of safety net to guard families against this stressful outcome.

CONCLUSION

In this paper, we demonstrated that dual-income family transportation routines could be sensed and modeled without any supervision using the GPS available on commodity mobile phones. Towards that end, we gathered a large data set of family location, and of pick-ups and drop-offs. Using simple heuristics and statistical models, we demonstrated that pick-ups and drop-offs can be sensed, and their drivers predicted. We also demonstrated a model that can predict when parents are going to forget to pick-up their children at activities before these damaging events happen.

We provided examples of how learned models of family routines can function as an *enabling* technology. We described how they could be added to digital calendars to help families make better plans, to reminder systems and to location systems to help families coordinate on routine days, and to function as a safety net, helping observe and prevent incidents where children are forgotten at their activities.

Future work

We intend this work to stimulate discussion around the broad applicability of routine as an enabling abstraction. Considering our choice of exploration over optimization, a significant amount of the modeling space remains open to investigation. Additionally, the use of routine in various application domains brings with it a series of questions. Identifying appropriate ways to visually represent implicit information, or entire learned routines, presents challenges to the visualization community. And ways to repair and amend incorrect implicit information presents challenges to the interaction techniques community.

Because routine as a computational abstraction is able to capture and model events that are meaningful but ephemeral, the algorithms developed in this work could be applied across other domains. In the workplace, for example, best practices could be described as a form of routine. Their modeling, visualization, and communication would represent another significant research opportunity.

ACKNOWLEDGMENTS

The authors would like to thank Google for their financial support. This work was funded by NSF grant IIS-1017429.

REFERENCES

1. Ashbrook, D. & Starner, T. (2002). Learning significant locations and predicting user movement with GPS. *Proc. ISWC 2002*, 101-108.
2. Atkeson, C. G., Moore, A. W. & Schaal, S. (1997). Locally Weighted Learning. *Artificial Intelligence Review*, 11:11-73.
3. Bandura, A. (1997). *Self-efficacy: The exercise of control*. New York: Freeman.
4. Beech, S. *et al*. (2004) Lifestyles of working parents: Implications and opportunities for new technologies, HP Tech report HPL-2003-88 (R.1).
5. Begole, J., Tang, J.C. & Hill, R. (2003). Rhythm modeling, visualizations and applications. *Proc. UIST 2003*, 11-20.
6. Bentley, F. & Metcalf, C. (2007). Sharing motion information with close family and friends. *Proc. CHI 2007*, 1361-1370.
7. Brdiczka, O., Makoto Su, N., Begole, J. (2009). Using temporal patterns (t-patterns) to derive stress factors of routine tasks. Proc. *CHI 2009 Ext. Abst.*, 4081-4086.

8. Brown, B. *et al*. (2007). Locating family values: A field trial of the Whereabouts Clock. *Proc. UbiComp 2007*, 354-371.

9. Cowan, R. S. (1985). *More work for mother: The Ironies of household technology from the open hearth to the microwave*. New York: Basic Books

10. Darrah, C.N., Freeman, J.M. & English-Lueck, J.A. (2007). *Busier than ever: Why American families can't slow down*, Palo Alto: Stanford University Press.

11. Davidoff, S., Lee, M.K., Yiu, C.M., Zimmerman, J. & Dey, A.K. (2006) Principles of smart home control. *Proc. UbiComp 2006*, 19-34.

12. Davidoff, S., Lee, M.K., Dey, A.K. & Zimmerman, J. (2007). Rapidly exploring application design with Speed Dating. *Proc. UbiComp 2007*, 429-446.

13. Davidoff, S., Zimmerman, J. & Dey, A.K. (2010). How routine learners can support family coordination, *Proc. CHI 2010*, 2461-2470.

14. Dey, A.K. & Abowd, G.D. (2000). CybreMinder: A Context-Aware System for Supporting Reminders. *Proc. HUC 2000*, 172-186.

15. Eagle, N. & Pentland, A. (2006). Reality mining: Sensing complex social systems, *Personal and Ubiquitous Computing*, 10(4): 255-268.

16. Fogarty, J., Baker, R. S., & Hudson, S. E. (2005). Case studies in the use of ROC curve analysis for sensor-based estimates in human computer interaction. *Proc GI 2005*, 129-136.

17. Fiese, B.H. *et al*. (2002). A review of 50 years of research on naturally occurring family routines and rituals: Cause for celebration? *Journal of Family Psychology*, 16(4): 381-390.

18. Frissen, V.A.J. (2000) ICTs in the rush hour of life, *The Information Society*, 16: 65-75.

19. Gneezy, U. & Rustichini A. (2000). Pay enough or don't pay at all. *Quarterly Journal of Economics,* August 2000: 791-810.

20. González, M.C., Hidalgo, C.A. & Barabási, A.-L. (2008). Understanding individual human mobility patterns, *Nature*, 453, 479-482 (2008).

21. Huynh, T., Fritz, M. & Schiele, B. (2008) Discovery of activity patterns using topic models. *Proc. UbiComp 2008*, 10-19.

22. Krumm, J. & Horvitz, E. (2006). Predestination: Inferring destinations from partial trajectories. *Proc. UbiComp 2006*, 243-260.

23. Liao, L., Patterson, D.J., Fox, D., & Kautz, H.A. (2007). Learning and inferring transportation routines. *Artificial Intelligence*. 171, 311-331.

24. Ludford, P. J. *et. al* (2006). Because I carry my cell phone anyway: Functional location-based reminder applications. *Proc. CHI 2006*, 889-898.

25. Marmasse, N. & Schmandt, C. (2000). Location-aware information delivery with ComMotion. *Proc. Handheld and Ubiquitous Computing*, 64-73.

26. Medved, C.E. (2004) The everyday accomplishment of work and family: Exploring practical actions in daily routines, *Communication studies*, 55(1): 128-145.

27. Neustaedter, C., Brush, A. J. & Greenberg, S. (2009). The calendar is crucial: Coordination and awareness through the family calendar. *ToCHI*, 16(1): 1-48.

28. Pentland, B.T., Haerem, T. & Hillison, D.W. (2009). Using workflow data to explore the structure of an organizational routine, in M.C. Baecker & N. Lazaric (Eds.) *Organizational routines: Advancing empirical research* (pp. 47-67), Cheltenham: Edward Elgar.

29. Plaisant, C et al. (2006). Shared family calendars: Promoting symmetry and accessibility. *ACM ToCHI*. 13(3): 313-346.

30. Roberts, S. W. (1959). Control chart tests based on geometric moving averages. *Technometrics*,1, 239-250.

31. Simpson, R.C., Schreckenghost, D., LoPresti, E.F, Kirsch, N. (2006). Plans and planning in smart homes, J.C. Augusto & C.D. Nugent (Eds.), *Designing Smart Homes: The Role of Artificial Intelligence*, 71-84.

32. Suchman, L. (1983). Office procedures as practical action: Models of work and system design. *ACM ToIS*, 1(4): 320-328.

33. Taylor, A. S. *et. al* (2007). Homes that make us smart. *Personal and Ubiquitous Computing*. 11(5): 383-393

34. Tolmie, P. *et al*. (2002). Unremarkable computing. *Proc. CHI 2002*, 399-406.

35. Van Laerhoven, K., Kilian , D. & Schiele, B. (2008). Using rhythm awareness in long-term activity recognition, *Proc. ISWC 2008*.

36. Wakkary, R. & Maestri, L. (2007). The resourcefulness of everyday design. *Proc. Creativity and Cognition 2007*, 163-172.

37. Wolin, S.J. & Bennett, L.A. (1984). *Family rituals. Family Process*, 23: 401-420.

38. Zerubavel, E. (1981). *Hidden rhythms: schedules and calendars in social life*, Chicago: The University of Chicago Press, 1981.

39. Ziebart, B.D. *et. al* (2008). Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior, *Proc. Ubicomp 2008*, 322-331.