

Learning Time-Based Presence Probabilities

John Krumm and A.J. Bernheim Brush

Microsoft Research
One Microsoft Way
Redmond, WA USA 98052
{jckrumm | ajbrush}@microsoft.com

Abstract. Many potential pervasive computing applications could use predictions of when a person will be at a certain place. Using a survey and GPS data from 34 participants in 11 households, we develop and test algorithms for predicting when a person will be at home or away. We show that our participants' self-reported home/away schedules are not very accurate, and we introduce a probabilistic home/away schedule computed from observed GPS data. The computation includes smoothing and a soft schedule template. We show how the probabilistic schedule outperforms both the self-reported schedule and an algorithm based on driving time. We also show how to combine our algorithm with the best part of the drive time algorithm for a slight boost in performance.

Keywords: Location prediction, presence prediction, away prediction, energy efficiency, human routines.

1 Introduction

Predicting when a person will be at a particular location could be useful in many pervasive computing scenarios. For example, a person initiating a spoken or typed conversation may want to wait until the other party is at home or in their office if the conversation will be sensitive or long. In other situations, someone may want an impromptu, face to face meeting. Here, predicted presence would be useful to find the best time to drop in, *e.g.* "She's nearly always in her office from 8 a.m. to 9 a.m.". Another application is energy savings. Gupta *et al.* of MIT show that households could save up to 7% on their heating bill with a thermostat that knows how far the occupants are from home [1]. For electric vehicles, cooling or preheating their batteries helps their performance [2], which would be aided by a prediction of when the driver will leave his or her current location. Predicted presence can also be used to detect anomalous behavior such as when a person is predicted to be somewhere but is not. Such behavior could be indicative of cognitive decline or an emergency.

This paper presents a technique for learning the probabilities, as a function of time, that a person will be at a particular place based on observations of their presence there. We concentrate on presence at home, but the technique is equally applicable to any place where a person's binary presence (*i.e.* there *vs.* not there) can be measured. In particular, we demonstrate inferences of an occupant's home/away schedule based on GPS logs of their whereabouts over time. We create a probability distribution

giving their probability of being away from home as a function of the time of day and the day of the week. In addition, we look at the occupant's current location as measured by GPS. We use this to override our probabilistic prediction if we discover the occupant is too far away to drive home within the prediction interval.

There is other work aimed at making general predictions about where people will be. For instance, Ashbrook and Starner look at GPS traces to find a person's significant locations along with a Markov model to predict which one will be visited next [3]. Patterson *et al.* use GPS to sense activities, including making short term predictions about a person's next destination [4]. Similarly, Krumm and Horvitz look at GPS traces to predict a driver's destination based on their previous habits and general driving behaviors [5]. These efforts concentrate on predicting specific locations in the future, not the arrival or departure times that we emphasize in this paper. In particular, algorithms like this that predict destinations and routes do not predict when the trip will start. The results of this paper, instead, can be used to predict when occupancy states will change.

Previous work on time-based presence prediction is normally aimed at thermostat control. An early attempt to solve the problem of occupancy prediction for home heating was that of Mozer *et al.* in 1997 [6]. Mozer's Neural Network House was outfitted with sensors - including motion sensors to detect occupancy - and actuators - including one to control a central hot air furnace. They trained a neural network to predict when the home would be occupied as a function of recent occupancy observations. Gao and Whitehouse, of the University of Virginia, present a "self-programming" thermostat that is sensitive to the home/away schedule of the occupants measured, by, for instance, occupancy sensors in the home [7]. Their algorithm finds a thermostat schedule to minimize heating and cooling times given the occupant's tolerance for "miss time", which is the amount of time the house is not heated or cooled when it should be. Gupta *et al.*'s GPS controlled thermostat uses a driving time heuristic to conservatively predict that an occupant will be home in a given amount of time if it is possible to drive home in that amount of time [1].

One innovation in our approach is that our predictions are probabilistic, meaning that algorithms that use the predictions can tailor their behavior to the inherent uncertainty in people's future behavior. Our predictions are based on a novel way of smoothing and biasing occupancy observations. We combine our learned probabilities with the driving time heuristic of Gupta *et al.* [1] and show how it improves our accuracy slightly. We also show how using our algorithm significantly improves prediction over users' own ideas of their home/away schedules. While the previous work cited above used data from one (Mozer *et al.* [6]), two (Gao and Whitehouse [7]), and eight (Gupta *et al.* [1]) individuals, our results are based on surveys and GPS data from 34 individuals spread among 11 different households. The next section describes our survey and the data we gathered.

2 Household GPS Survey

In late 2009, we recruited 12 volunteer households in our area in order to gather data for our study for a period of approximately eight weeks each. These households were

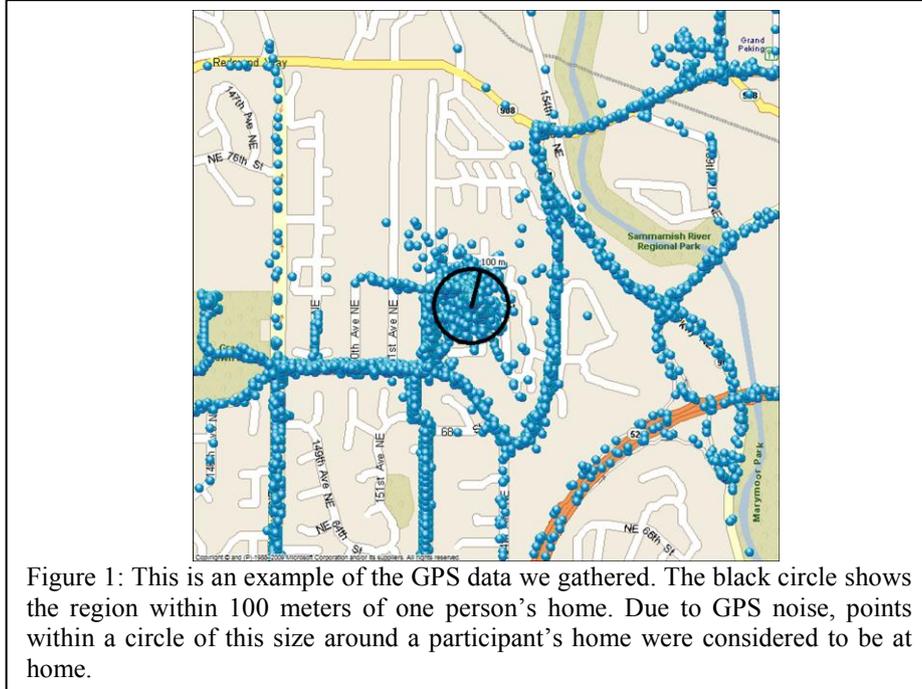
		Day of Week						
		Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Time of Day (hour)	0	sleeping	sleeping	sleeping	sleeping	sleeping	sleeping	awake home
	1	sleeping	sleeping	sleeping	sleeping	sleeping	sleeping	sleeping
	2	sleeping	sleeping	sleeping	sleeping	sleeping	sleeping	sleeping
	3	sleeping	sleeping	sleeping	sleeping	sleeping	sleeping	sleeping
	4	sleeping	sleeping	sleeping	sleeping	sleeping	sleeping	sleeping
	5	sleeping	sleeping	sleeping	sleeping	sleeping	awake home	sleeping
	6	awake home	awake home	sleeping	sleeping	sleeping	awake home	sleeping
	7	away	awake home	awake home	awake home	awake home	away	sleeping
	8	away	awake home	awake home	awake home	awake home	away	sleeping
	9	away	away	away	awake home	awake home	away	awake home
	10	away	away	away	away	away	away	awake home
	11	away	away	away	away	away	away	awake home
	12	away	away	away	away	away	away	awake home
	13	away	away	away	away	away	away	awake home
	14	away	away	away	away	away	away	away
	15	away	away	away	away	away	away	away
	16	awake home	awake home	away	awake home	away	away	away
	17	awake home	awake home	awake home	awake home	away	awake home	away
	18	awake home	awake home	awake home	awake home	away	awake home	away
	19	awake home	awake home	awake home	awake home	awake home	awake home	awake home
	20	awake home	awake home	awake home	awake home	awake home	awake home	awake home
	21	awake home	awake home	awake home	awake home	awake home	awake home	awake home
	22	awake home	sleeping	sleeping	sleeping	awake home	awake home	awake home
	23	sleeping	sleeping	sleeping	sleeping	sleeping	awake home	awake home

Table 1: Each of our participants filled out a time grid representing their typical week. In each one-hour cell, the participant could indicate sleeping, awake at home, or away from home. This is the data provided by one of our participants.

on a list of user study volunteers maintained by our institution, but not employed or otherwise associated with our institution. All the households had either three or four participants each, although one participant dropped out at the beginning of the study, leaving two participants remaining in one household. Also, one household of three did not properly comply with the GPS portion of the survey (explained below), so we dropped them, leaving 11 households with a total of 34 participants. One household had two child participants, and three households had one child participant. The participants were evenly split across genders, and their ages ranged between 21 and 59, with a median age of 27. Six of the households were families with children living at home, and one was a couple without children. In return for participating in our survey, each household was offered four products of their choice from our institution (maximum value US\$ 600 per product) and each participant was offered US\$ 0.50 for each day of at least two hours of GPS log data.

We asked each participant to do two main tasks. One of the tasks was to fill in a time grid predicting their status among “awake at home”, “sleeping at home”, and “away from home” for each hour of each day of a typical week. Data from a grid for one participant is shown in Table 1. This is analogous to programming a thermostat, where a person might pick different temperatures for each of these three states. We used these participant time grids to compare against other algorithms for predicting when a person would be home or away.

The other major task of our survey participants was to carry a GPS logger with them during their waking hours. As part of our initial visit to each household, we loaned each participant a RoyalTek RBT-2300 GPS logger, equipped with an optional 1700 milliamp-hour, rechargeable battery, plus a recharger. These loggers fit conveniently in a pocket or bag, and we set them to record a time-stamped latitude/longitude every five seconds. The larger, optional battery was enough for about 18 hours of operation on one charge. We instructed the participants to carry the logger with them wherever they went and have it turned off and recharging while they were sleeping. We also asked the participants to mail their loggers to us every two weeks, switching to a second set of loggers we left with them. When we received the



loggers, we uploaded and inspected the data to make sure the participants were properly complying. We then mailed back the empty loggers to serve as the replacement set after the next two-week switch, *etc.*, until the end of the survey. An example of the type of GPS data we collected is shown in Figure 1.

An analysis of the data shows that the average, minimum, and maximum number of days we observed the 34 participants were 58, 13, and 95, respectively. The participants did not have their GPS loggers on all the time, *e.g.* normally turned off overnight, and sometimes forgotten in the morning. The average, minimum, and maximum fraction of time we obtained GPS data from the participants were 38%, 18%, and 76%. Some of the lower percentages were due to loggers that failed to upload their data after two weeks of logging.

We used this GPS data to devise an algorithm for predicting when our participants would be home or away. First, however, we used their survey responses to assess how well they could predict their own home/away behavior, described in the next section.

3 Self-Reported Home/Away Schedules

It may be that people are quite good at predicting their own home/away behavior. If so, there would not necessarily be a strong need to make these predictions automatically. Part of our survey asked each participant to fill out a schedule of when they are sleeping, at home, or away from home. An example schedule from one of our participants is shown in Table 1. For the purposes of this study, we designated sleeping times as being at home.

The participants' GPS data, along with knowledge of their home locations, gave us a simple way to measure their actual home/away behavior. We designated any GPS point within 100 meters of the participant's home to be at home, and designated the remaining points as away. We chose the 100 meter radius based on the observed spread of the GPS data as shown in maps such as in Figure 1. While a circle of this size could easily include many neighbors, we felt compelled to keep the circle this large to account for the occasional drift of our GPS logger.

With the GPS home/away data as ground truth, we can assess how well our participants anticipated their own home/away behavior. We note that the quality of predictions based on a schedule like this do not vary with the look-ahead time, since each participant's predicted schedule is static. For other predictions we make below, the look-ahead time is a factor.

Table 2 shows the confusion matrix averaged over all our participants. We computed this by considering every GPS point as a ground truth point, assigning it a label of "home" or "away" depending on its location. We used the GPS point's time stamp to look up the participant's anticipated home/away state in their self-reported home/away schedule. The confusion matrix shows that when a participant was actually away (as measured by GPS), they predicted they would be home about 68% of the time. We conclude that our participants were not good at anticipating their home/away schedules, and we next consider algorithms to automatically infer home/away in hopes of improvement.

We note that the participants were likely not quite as poor at predicting their home/away status as the confusion matrix implies. We assessed their home/away prediction only when we had GPS data for ground truth, which did not include overnights, because we asked participants to turn off their GPS overnight for recharging. Thus nighttime data, when the participants were most likely home and when they likely correctly predicted they would be home, was not included in the calculations. So, we conclude that during waking hours, our participants were not good at predicting their home/away pattern. In the following sections, we use the same GPS ground truth data to assess other algorithms, so we can directly compare performance, despite the lack of nighttime data.

		Inferred	
		home	away
Actual from GPS	home	76%	24%
	away	68%	32%

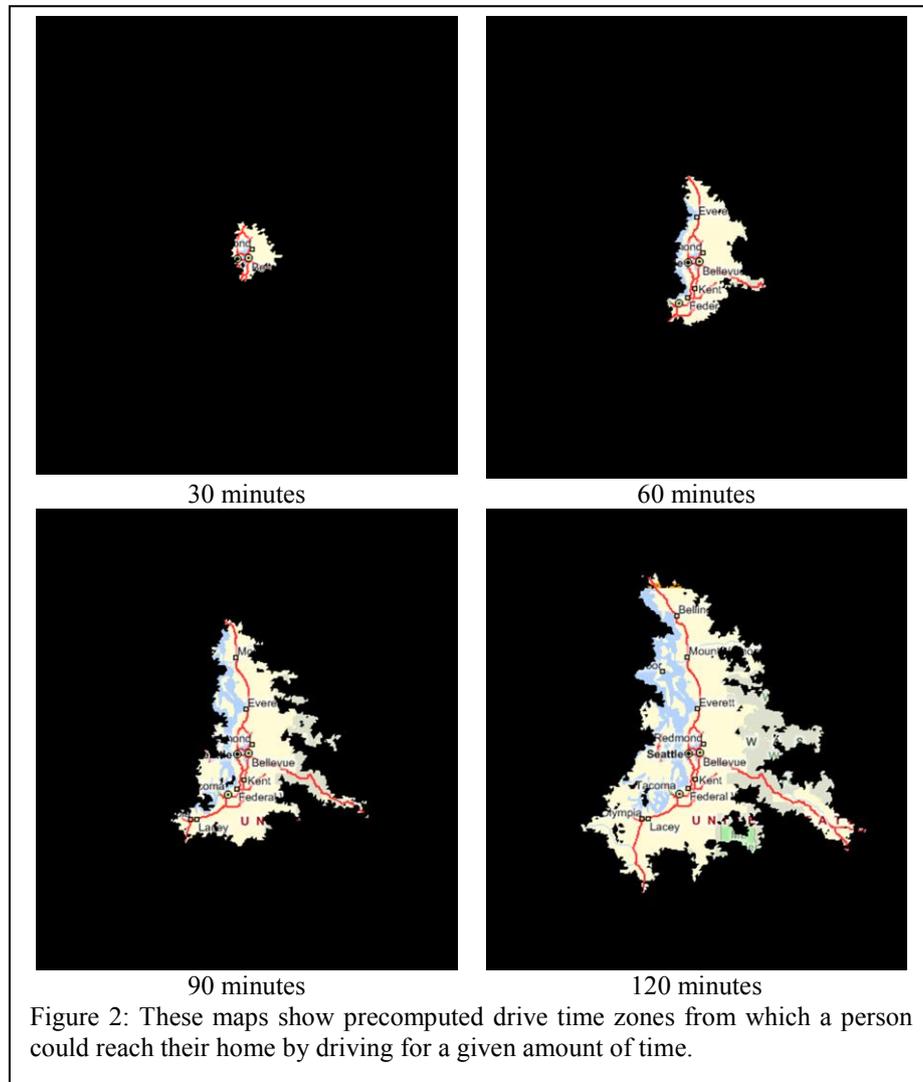
Table 2: Participant Self-Report Confusion Matrix.

The confusion matrix shows that our participants were not good at anticipating when they would be home or away, based on ground truth from GPS. They predicted they would be home much more often than they actually were.

3 Drive Time Prediction

The work in [1] introduces thermostat control based on the location of the home's occupants. They recommend, in the absence of a programmable thermostat, to keep the house warm if the time to heat the home is more than the time it would take an occupant to drive home. Thus, this algorithm conservatively predicts that a person will always be home in the amount of time it would take him or her to drive home. We will refer to this algorithm as the "drive time" algorithm, and we will use it to measure the relative accuracy of our own presence prediction algorithm and to augment our algorithm for more accuracy.

With our GPS data, we were able to assess the drive time algorithm in terms of the



same type of confusion matrix presented in the last section. However, before we present the results, we describe one modification we made to the drive time algorithm for efficiency. While [1] used MapQuest to predict driving times from each GPS point, we instead precomputed driving times from points sampled on a map. In particular, we tessellated the map in our study region with triangles from the Hierarchical Triangular Mesh (HTM) [8]. From the available mesh resolutions, we used level 12 triangles, whose size in our study region was about 5.1 square kilometers (area) and 3.4 kilometers (length of each side).

For each triangle, we computed the driving time from the triangle’s center to the participant’s home and stored the result. Then, given an arbitrary latitude/longitude, we found which triangle contained it and returned that triangle’s driving time as an approximation of the driving time from that point. This modification of the algorithm in [1] was not designed to increase the accuracy of the algorithm, but rather to increase the computational efficiency. Instead of computing a driving time for each query, we simply have to look up the precomputed driving time from the relevant triangle. Since the triangles are small, the loss in driving time accuracy caused by discretization is small. Thresholding the drive times in the triangles is a convenient way to show a map of the region over which a participant’s home is reachable in a given amount of time, as shown in Figure 2 for an arbitrary home location.

From Figure 2 it is easy to understand the drive time algorithm. For example, if the look-ahead time for the prediction is 90 minutes, the occupant would be predicted to arrive at home in at most 90 minutes from anywhere within the 90-minute drive time region. We call this region the drive time zone.

We can apply the drive time algorithm to home/away prediction by predicting that an occupant will be home in some amount of time if they are within the drive time zone associated with that time. Otherwise we predict they will be away. We note that

		Inferred	
		home	away
Actual from GPS	home	100%	0%
	away	90%	10%

30-minute drive time prediction

		Inferred	
		home	away
Actual from GPS	home	100%	0%
	away	93%	7%

60-minute drive time prediction

		Inferred	
		home	away
Actual from GPS	home	100%	0%
	away	94%	6%

90-minute drive time prediction

Table 3: **Drive Time Algorithm.** These confusion matrices show the performance of an algorithm that predicts the user will be home in X minutes whenever he or she is within X minutes of driving time from their home. Since most people in our study spent most of their time close to home, this algorithm almost always predicts they will be home within the prediction interval.

the schedule-based algorithm in the previous section is insensitive to the look-ahead time, because its predictions are completely determined by the time of day and day of week. The drive time, algorithm, in contrast, depends on the look-ahead time.

Applying the drive time algorithm to our participants' data, we get the confusion matrices shown in Table 3 for look-ahead times of 30, 60, and 90 minutes. The charts in Table 3, Figure 4, and Figure 5 show how the drive time algorithm compared to others we tested. The defining aspect of the drive time confusion matrices is that the algorithm almost always predicts "home" regardless of the data. This is because our participants spent the vast majority of their time near their homes. This is likely true of the general U.S. population, whose average commute time from work is about 23 minutes, and 81% of whom work within 45 minutes of home [9].

While this algorithm did not perform well on our participants' data, we show later how to combine it with a more accurate algorithm for a slight improvement in the other algorithm's accuracy.

4 Probabilistic Home/Away Schedules

Despite the fact that our participants were not good at anticipating their own home/away schedules, we suspect there is much to be gained by looking at their regular habits. This section describes how, using their GPS data, we computed the probability of them being away from home as a function of the time of day and day of week, as shown in Table 4. (We note that if the probability of being away from home is p_{away} , then $p_{home} = 1 - p_{away}$.) In this table the time slots are 30 minutes long. This is an arbitrary choice, but we found that 30 minutes worked well for our purposes.

The advantages of using a probabilistic table such as this are:

- It is based on users' actual home/away behavior, and thus is a more accurate reflection of their schedule than a self-reported one.
- The probabilities capture the fact that people are not completely predictable.
- Using probabilities means that algorithms using these predictions can explicitly account for the inherent uncertainty.
- The probabilities can be used as a prior for a more sophisticated Bayesian approach to home/away prediction.

As we did previously, we say that a participant was home when their GPS data indicated they were within 100 meters of their home latitude/longitude.

One way to build a probabilistic home/away schedule would be to create a simple histogram of normalized frequencies. For each time/day slot in the schedule, we could simply count the number of times the user was away from home, based on GPS readings, and divide by the total number of GPS readings in that slot. However, this leads to problems when there is no sample data for a slot, and it also neglects the opportunity to impose prior assumptions on the schedule.

Below we describe our procedure for building a probabilistic home/away schedule which fills in missing values, smoothes the data, and allows a soft bias in the regularity of the schedule.

Imposing a Schedule Template

We formulate the problem of finding a p_{away} schedule as a linear matrix problem, where the unknowns are the p_{away} probabilities in the time slots. Specifically, the unknowns form a vector \mathbf{p}_{week} , where each element is p_{away} for a particular time slot on a particular day of the week, *i.e.*

$$\mathbf{p}_{week} = (p_1 p_2 p_3 \dots p_i \dots p_{336})^T \quad (1)$$

This vector is 336 elements long, which is the number of 30-minute periods in 7 days. The elements are organized in day-major order, so p_1 corresponds to the first 30 minutes of Sunday after midnight, and p_{336} corresponds to the last 30 minutes of Saturday before midnight.

We suspect that people have a somewhat unvarying home/away schedule on weekdays, with more variations on weekends. Therefore, we introduce another vector of away probabilities that correspond to a generic weekday, Monday - Friday. This vector is $\mathbf{p}_{generic\ weekday}$, and there is one element for each 30-minute slot of a single weekday, *i.e.*

$$\mathbf{p}_{generic\ weekday} = (p'_1 p'_2 p'_3 \dots p'_j \dots p'_{48})^T \quad (2)$$

where 48 is the number of 30-minute periods in one 24-hour, generic weekday. After solving for \mathbf{p}_{week} and $\mathbf{p}_{generic\ weekday}$, the final probability for a weekday slot is computed as the sum of the relevant element of \mathbf{p}_{week} (corresponding to a time slot on a specific day of the week) and the relevant element of $\mathbf{p}_{generic\ weekday}$ (corresponding to the time slot on a generic weekday). The final probability for a weekend slot comes solely from \mathbf{p}_{week} .

Introducing $\mathbf{p}_{generic\ weekday}$ is a way to impose our bias that people have a somewhat regular schedule on weekdays. $\mathbf{p}_{generic\ weekday}$ represents the unvarying part of a weekday, which is summed with the elements of \mathbf{p}_{week} that represent the variable parts of specific weekdays. There are many such possible decompositions. For instance, it may be that only daytime hours of weekdays are unvarying. We introduced the generic weekday as the intuitively most likely decomposition, but we leave for future work a verification that it improves accuracy. An interesting extension to this technique is to examine different types of probability decompositions to find which one, if any, works best for an individual. As it stands, our generic weekday decomposition is an example of how to impose these types of decompositions mathematically.

The linear matrix equation for computing the probabilities is

$$A \begin{pmatrix} \mathbf{p}_{week} \\ \mathbf{p}_{generic\ weekday} \end{pmatrix} = \mathbf{b} \quad (3)$$

Here A is a matrix representing constraint equations on the probabilities, with the \mathbf{b} vector representing the constraints' constant parts. The unknown vector

$$\mathbf{p} = \begin{pmatrix} \mathbf{p}_{week} \\ \mathbf{p}_{generic\ weekday} \end{pmatrix}$$

contains the probabilities we want to compute. The remainder of this section discusses how we fill the elements of A and \mathbf{b} based on data and other constraints.

Home/Away Frequencies

The main influence on the away probabilities is the home/away data itself. We create one constraint equation for each 30-minute period of collected GPS data. In these periods, we compute the proportion of GPS points outside the 100-meter radius of the home compared to the total number of GPS points measured in the time period. If one row of matrix A is represented by the row vector \mathbf{a} , and one element of vector \mathbf{b} is represented by b , then the form of this constraint for one observed 30-minute time slot is

$$\mathbf{a} \cdot \mathbf{p} = b$$

$$(0 \ 0 \ \dots \ 1 \ \dots \ 0 \ 0 \mid 0 \ 0 \ \dots \ 1 \ \dots \ 0 \ 0) \cdot \mathbf{p} = \frac{n_{away}}{n_{away} + n_{home}} \quad (4)$$

Here the two 1's in \mathbf{a} are positioned to pick up the time of day and day of week slot in \mathbf{p}_{week} and $\mathbf{p}_{generic \ weekday}$ that correspond the time slot in the data. The vertical divider in \mathbf{a} corresponds to the division between the two parts of \mathbf{p} : \mathbf{p}_{week} and $\mathbf{p}_{generic \ weekday}$. If the time slot is on a weekend, the second 1 in \mathbf{a} is replaced with a 0, because there is no generic time slot for weekends. The integers n_{home} and n_{away} are the counts of GPS points inside and outside the 100-meter circle in the data's time slot.

There is one (\mathbf{a}, b) pair, and thus one row of matrix A , for every 30-minute time slot in the observed data. We keep appending (\mathbf{a}, b) pairs to $A\mathbf{p} = \mathbf{b}$ until we exhaust all the participant's GPS data. With approximately eight weeks of data from each participant, there are many more 30-minute data slots than unknowns in \mathbf{p} , making the matrix equation over-constrained. We eventually use a least squares approach to find a solution.

Generic Weekday Influence

We want to adjust the magnitude of the probabilities for a generic weekday, $\mathbf{p}_{generic \ weekday}$, to allow for more or less variation on weekdays. To do this, we introduce a regularization factor, λ_{wd} , to potentially reduce the generic weekday probabilities. In terms of the growing $A\mathbf{p} = \mathbf{b}$ equation, we add rows to A and \mathbf{b} that look like the following:

$$\lambda_{wd} \begin{bmatrix} 0 & \dots & \dots & 0 & \mid & 1 & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots & \mid & 0 & 1 & 0 & \vdots \\ \vdots & & \ddots & \vdots & \mid & \vdots & & \ddots & 0 \\ 0 & \dots & \dots & 0 & \mid & 0 & \dots & 0 & 1 \end{bmatrix} \mathbf{p} = \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \quad (5)$$

$$[0_{48 \times 336} \mid \lambda_{wd} I_{48 \times 48}] \mathbf{p} = 0_{48 \times 1}$$

This has the effect driving all the elements of $\mathbf{p}_{generic\ weekday}$ to zero. This effect is moderated by λ_{wd} . We used $\lambda_{wd} = 0.0001$, and we describe subsequently how we chose this value.

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Gnrc Wkdy
12:00 AM	0.050	0.000	0.000	0.000	0.000	0.000	0.453	0.000
12:30 AM	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1:00 AM	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1:30 AM	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000
2:00 AM	0.000	0.000	0.012	0.000	0.000	0.000	0.000	0.000
2:30 AM	0.000	0.000	0.035	0.000	0.000	0.000	0.000	0.000
3:00 AM	0.000	0.000	0.075	0.000	0.000	0.000	0.000	0.000
3:30 AM	0.000	0.000	0.133	0.000	0.000	0.007	0.000	0.000
4:00 AM	0.000	0.000	0.209	0.000	0.000	0.032	0.000	0.000
4:30 AM	0.000	0.000	0.300	0.000	0.000	0.084	0.000	0.000
5:00 AM	0.000	0.000	0.404	0.000	0.000	0.171	0.000	0.000
5:30 AM	0.000	0.000	0.515	0.000	0.000	0.298	0.000	0.000
6:00 AM	0.000	0.001	0.625	0.000	0.000	0.471	0.000	0.000
6:30 AM	0.000	0.807	0.728	0.371	0.001	0.692	0.000	0.001
7:00 AM	0.000	1.000	0.812	0.427	0.170	0.962	0.000	0.002
7:30 AM	0.000	1.000	0.934	0.583	0.461	0.964	0.000	0.073
8:00 AM	0.000	1.000	0.999	0.649	0.565	0.875	0.000	0.132
8:30 AM	0.000	0.833	0.294	0.797	0.587	0.875	0.000	0.061
9:00 AM	0.000	0.857	0.091	0.560	0.379	0.810	0.182	0.002
9:30 AM	0.000	0.857	0.200	0.546	0.090	0.714	0.200	0.000
10:00 AM	0.149	0.993	0.443	0.429	0.000	0.514	0.200	0.000
10:30 AM	0.376	1.000	0.833	0.637	0.341	0.571	0.011	0.219
11:00 AM	0.600	1.000	0.833	0.804	0.571	0.571	0.101	0.322
11:30 AM	0.567	1.000	0.714	0.625	0.400	0.574	0.189	0.252
12:00 PM	0.383	1.000	0.714	0.581	0.400	0.541	0.368	0.255
12:30 PM	0.400	1.000	0.714	0.714	0.703	0.400	0.375	0.325
1:00 PM	0.388	1.000	0.714	0.714	0.750	0.500	0.348	0.357
1:30 PM	0.376	1.000	0.714	0.714	0.750	0.352	0.287	0.324
2:00 PM	0.400	0.985	0.714	0.667	0.750	0.310	0.345	0.294
2:30 PM	0.721	1.000	0.714	0.667	0.714	0.315	0.143	0.283
3:00 PM	0.750	0.897	0.667	0.729	0.714	0.250	0.208	0.250
3:30 PM	0.600	0.500	0.650	0.712	0.559	0.328	0.427	0.160
4:00 PM	0.600	0.600	0.571	0.440	0.498	0.250	0.375	0.099
4:30 PM	0.600	0.368	0.709	0.336	0.429	0.151	0.148	0.043
5:00 PM	0.600	0.200	0.612	0.251	0.519	0.142	0.125	0.000
5:30 PM	0.595	0.314	0.429	0.375	0.506	0.125	0.143	0.007
6:00 PM	0.333	0.500	0.510	0.599	0.571	0.125	0.000	0.125
6:30 PM	0.333	0.429	0.532	0.429	0.460	0.125	0.143	0.085
7:00 PM	0.305	0.429	0.418	0.371	0.429	0.125	0.053	0.080
7:30 PM	0.167	0.302	0.250	0.384	0.313	0.290	0.000	0.073
8:00 PM	0.167	0.286	0.220	0.286	0.250	0.351	0.094	0.081
8:30 PM	0.167	0.172	0.125	0.286	0.290	0.375	0.143	0.083
9:00 PM	0.167	0.143	0.125	0.206	0.343	0.375	0.143	0.095
9:30 PM	0.108	0.143	0.053	0.143	0.202	0.375	0.143	0.053
10:00 PM	0.000	0.000	0.000	0.143	0.143	0.333	0.143	0.000
10:30 PM	0.000	0.000	0.000	0.200	0.143	0.400	0.143	0.000
11:00 PM	0.000	0.000	0.000	0.250	0.000	0.667	0.143	0.000
11:30 PM	0.000	0.000	0.000	0.000	0.000	0.667	0.143	0.000

Table 4: This table gives the probability of someone being away from their home as a function of the time of day and day of week. In this case, there is a high probability of being away during most normal working hours on Monday – Thursday. Also, this person appears to be often away from home on Friday nights until the first 30 minutes of Saturday. The generic weekday in the last column shows a bulge during normal work hours as expected.

Smoothing

We also allow for a degree of temporal smoothing of the away probabilities to account for vagaries in the limited observation time. Smoothing is also critical for filling in missing data, because sometimes we have no GPS data for certain nighttime time slots. For an away probability p_i from \mathbf{p}_{week} , we smooth with the probabilities of the previous and next time slots, *i.e.* we want

$$\begin{aligned} \frac{s}{2}p_{i-1} + (1-s)p_i + \frac{s}{2}p_{i+1} &= p_i \\ \frac{s}{2}p_{i-1} - sp_i + \frac{s}{2}p_{i+1} &= 0 \end{aligned} \quad (6)$$

where $0 \leq s \leq 0.5$ controls the amount of smoothing and p_{i-1} , p_i , and p_{i+1} are three, temporally adjacent away probabilities. This smoothing constraint is moderated by a smoothing regularization factor λ_s . For smoothing, we add rows to A and \mathbf{b} that look like the following:

$$\lambda_s \begin{bmatrix} -s & s/2 & 0 & \cdots & 0 & s/2 \\ s/2 & -s & s/2 & 0 & \vdots & 0 \\ 0 & s/2 & -s & s/2 & 0 & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & s/2 & -s & s/2 \\ s/2 & 0 & \cdots & 0 & s/2 & -s \end{bmatrix} \mathbf{p} = \mathbf{0} \quad (7)$$

$$[\lambda_s S_{336 \times 336} | \mathbf{0}_{336 \times 48}] \mathbf{p} = \mathbf{0}_{336 \times 1}$$

Smaller values of λ_s tend to reduce the effect of smoothing on the final probabilities. Likewise, a smaller value of s means less smoothing between temporally adjacent probabilities.

Solving $A\mathbf{p} = \mathbf{b}$ and Choosing Parameters

The equation $A\mathbf{p} = \mathbf{b}$ is built from three parts: away frequencies from GPS data, moderating the effect of the generic weekday with λ_{wd} , and smoothing with s and λ_s . The equation is over-constrained, so we solve with least squares. We also require the resulting probabilities to be between zero and one, so we use a constrained solver.

To choose the parameters λ_{wd} , s , and λ_s , we used two-way cross validation on eight weeks of GPS data taken from a participant outside our study. We made a rough sweep through possible values of the parameters. For each set of parameter values, we compared the computed probabilities from half the GPS data to the ground truth computed from the other half of the GPS data. The best values of the parameters were

$$\begin{aligned} \lambda_{wd} &= 0.0001 \\ s &= 0.4 \\ \lambda_s &= 0.1 \end{aligned}$$

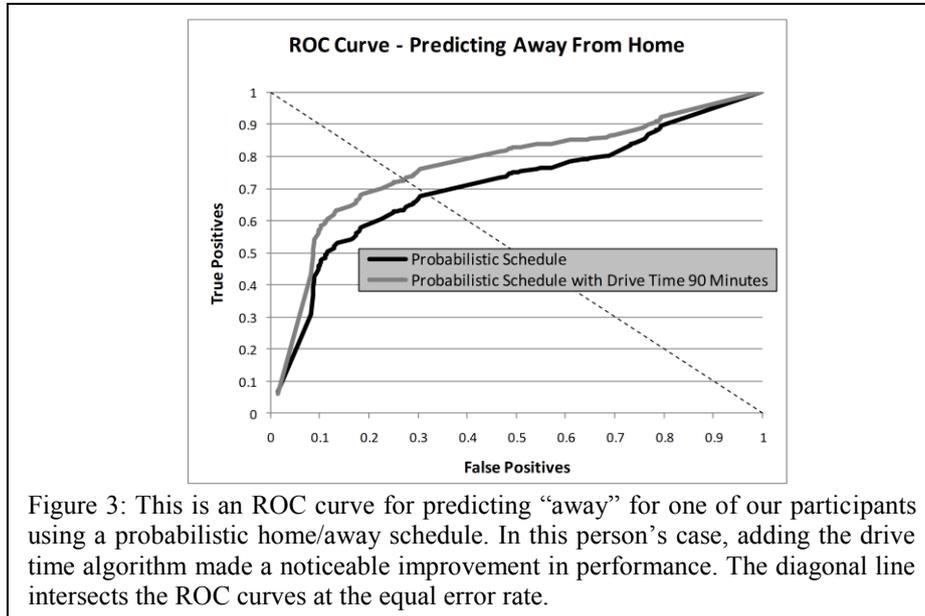


Figure 3: This is an ROC curve for predicting “away” for one of our participants using a probabilistic home/away schedule. In this person’s case, adding the drive time algorithm made a noticeable improvement in performance. The diagonal line intersects the ROC curves at the equal error rate.

We used these parameters to compute away probabilities for each participant. An example result for one of our participants is shown in Table 4.

Evaluation of Probabilistic Schedule

The computed away probabilities introduce a convenient parameter into prediction for presence. For presenting estimates to other people, such as the probability of a person being in their office, a system could simply present the computed presence probability and let the other person decide what action to take. For automatic behaviors, such as controlling a home’s temperature, we can set a threshold on the away probability to decide when to trigger an action. The probability can be combined with perceived costs of incorrect predictions, giving a decision-theoretic result. For instance, a low threshold on p_{away} translates into a high threshold on p_{home} , and it means that the system would have to be more confident of an impending arrival in order to take any action. As an example, for home heating, this threshold translates into a user-adjustable tradeoff between comfort and energy savings. If comfort is more important, the user would set the threshold such that the home would be heated even if there was only a relatively small chance of arriving at home at the cost of sometimes heating an empty house. To save more energy, the user would adjust the threshold to reduce the chance of heating the home unnecessarily at the cost of sometimes arriving home to a cold house. With a probabilistic schedule like the one we produce, this tradeoff becomes possible. It is similar in spirit to the tradeoff introduced in [7] in which users set the “miss time” to control for how long the home’s temperature is miscontrolled. The drive time algorithm and the self-reported schedule have no such adjustment available.

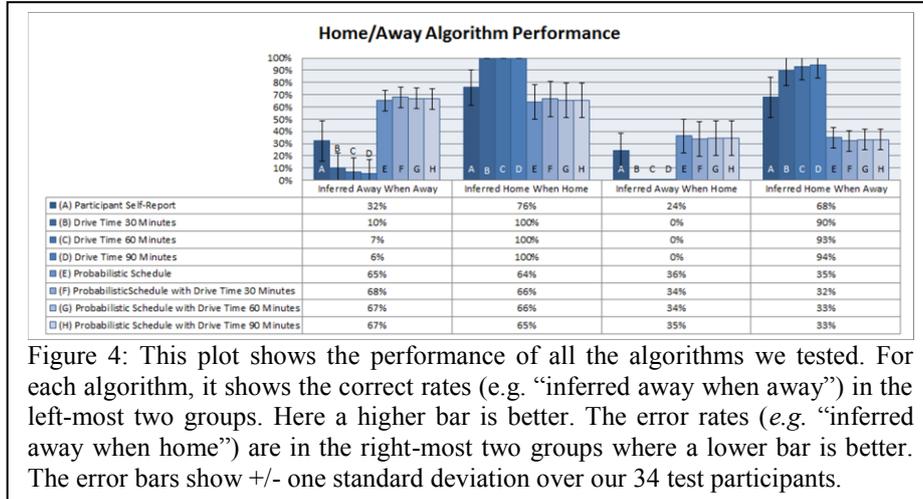


Figure 4: This plot shows the performance of all the algorithms we tested. For each algorithm, it shows the correct rates (e.g. “inferred away when away”) in the left-most two groups. Here a higher bar is better. The error rates (e.g. “inferred away when home”) are in the right-most two groups where a lower bar is better. The error bars show +/- one standard deviation over our 34 test participants.

We evaluated our probabilistic schedules with 5-fold cross validation. For each participant, we split their GPS data into five equal-length parts in temporal order. For each of the five validation runs, we tested on one part and trained on the other four parts, picking a new test part for each run.

The probabilistic schedule predictor does not use a specific look-ahead time for prediction. Since it assumes that the probabilistic schedule is forever unvarying, it can be used to predict ahead any amount of time. This is manifest in our results, because we show no look-ahead time for this algorithm, unlike the drive time algorithm which considers a specific amount of time for its predictions.

In evaluating the accuracy of the probabilistic schedule, we account for the adjustable probability threshold by creating an ROC curve that demonstrates the performance tradeoff at different settings of the probability threshold. An example of an ROC curve for one of our participants is shown in Figure 3. This shows the performance of predicting if the person will be away from home at different settings of the threshold on p_{away} . At high settings, the system must be very confident of an upcoming departure before it will predict an away state. This corresponds to the lower left part of the plot where the chance of a false positive is low, but where the high threshold also reduces the chance of a true positive. At the other end of the plot, the threshold is low, where the chances of a false positive and true positive are both high. Ideally there would be a threshold that gives 100% true positives and no false positives, which is the upper left corner of the plot.

One advantage of our algorithm is that it allows this adjustment, which gives higher level algorithms the flexibility to trade off one type of error for another.

To reduce the ROC curve to a confusion matrix for comparison with the other algorithms, we look at the equal error rate, which in Figure 3 is where the diagonal line intersects the ROC curve. Using the equal error rate point, the confusion matrix associated with home/away prediction using probabilistic schedules from all our participants is shown in Table 5. Figure 4 shows how this algorithm’s confusion matrix numbers compare with the others. The probabilistic schedule algorithm gives a much better balance for predicting home and away compared to participant’s self-

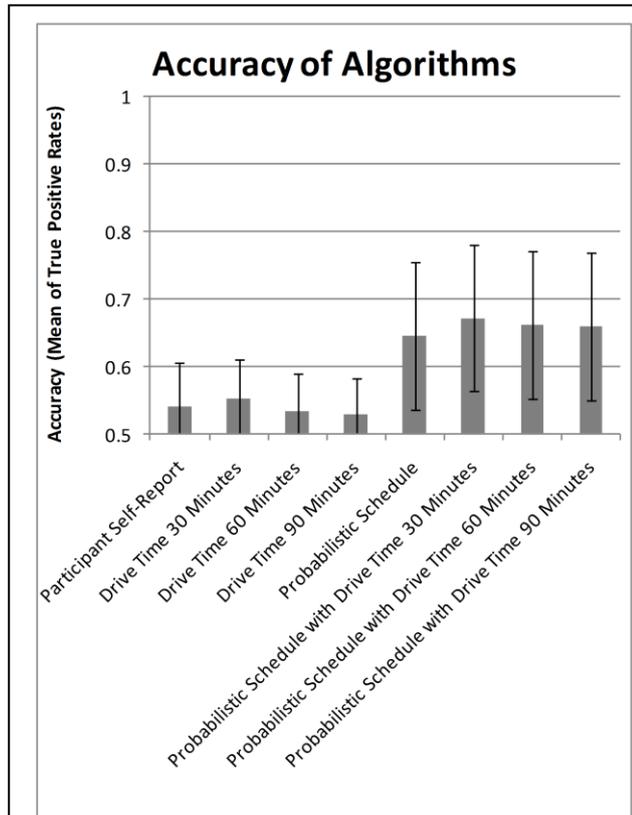


Figure 5: This chart shows the accuracy of each algorithm, which in our case is the average of the true positive rates in the confusion matrices. The error bars show +/- one standard deviation across our 34 participants. The minimum accuracy is $\frac{1}{2}$, because that is achievable by simply guessing “home” or “away” 100% of the time. The maximum possible accuracy is 1.0.

$\frac{1}{2}$, since this is trivially achievable by guessing “home” or “away” 100% of the time.

reported schedules and the drive time algorithm, both of which significantly overestimate predictions that the participants will be home.

Figure 5 shows how the probabilistic schedule algorithm compares to the previous algorithms in terms of accuracy, where accuracy is in our case simply the mean of the diagonal elements of the confusion matrix. The probabilistic schedule algorithm is significantly more accurate than the previous algorithms, although the accuracy figure hides the fact that the previous algorithms

(participants’ self-reported schedule and drive time) get most of their accuracy from over-predicting when the participant will be home. Note that the minimum accuracy in the plot in Figure 5 is

5 Combining Probabilistic Schedule and Drive Time Algorithms

There is an easy way to combine the drive time algorithm with the probabilistic schedule algorithm. The strength of the drive time algorithm is that it will never predict that a person can arrive at home in less time than it would take to drive home. Unless the person is traveling home faster than normal vehicular traffic, this heuristic will almost always be correct. Thus, we modified our probabilistic schedule algorithm to always predict “away” if the participant was outside the relevant drive time zone, regardless of the probability in the schedule. If the participant was within the drive time zone, we resorted to the probabilistic schedule instead. This combination of the algorithms takes the best part of the drive time algorithm and ignores its rule to predict “home” whenever a participant is within the drive time zone. We found that this addition improved the probabilistic schedule algorithm slightly but noticeably. The confusion matrices are shown in Table 6, and Figure 4 & Figure 5 show how this algorithm compares to the others. Figure 3 shows the improvement in the ROC curve

		Inferred	
		home	away
Actual from GPS	home	64%	36%
	away	35%	65%

Table 5: **Probabilistic Schedule.** The confusion matrix shows the performance of prediction for the probabilistic schedule we derived from participants’ GPS data.

for one of our participants. In all cases, there is a slight improvement.

6 Discussion and Summary

With the goal of predicting a home’s occupancy for energy

efficiency, this paper shows that a probabilistic home/away schedule derived from GPS data works much better than peoples’ self-reported schedules and much better than making predictions based purely on the time it would take to drive home. Our study was based on approximately two months GPS data from each of 34 participants.

We introduced a matrix-based method to compute probabilistic schedules that allows for the application of a soft schedule template on the data. In our case, we used a template that emphasizes a similar schedule on weekdays. Our method also smoothes the data.

We also showed how to increase the performance of our probabilistic schedule algorithm by adding the best part of the drive time algorithm.

Our probabilistic schedule proved much more accurate than our participants’ own impression of their weekly home/away schedules. One possible objection to this result is that our participants filled out a schedule with time discretized to 1-hour pieces, while our probabilistic schedule worked with 30-minute pieces, allowing more accuracy in transition times. However, we found our participants were so poor at predicting home/away, that higher resolution discretization would not help much. For instance, as shown in Table 2, for 68% of the time when participants predicted they would be home, they were actually away. With only a few home arrivals and departures per day, adjusting these times by 30 minutes would not be enough to eliminate an error this large.

In practice, these probabilistic schedules could be kept up-to-date by processing only the most recent location traces of an individual, thus staying more current as weekly schedules inevitably change. It would be interesting to investigate more sophisticated methods for maintaining a probabilistic schedule, perhaps by assembling chunks of previous schedules. Recent work has shown that only about 38% of a family's travel activities are routine, implying that there is an opportunity for improved predictions beyond a derived schedule like ours [10]. Another promising research

question is whether or not a system like ours could use a coarser, more energy efficient location system like WiFi or cell tower positioning instead of GPS.

		Inferred	
		home	away
Actual from GPS	home	66%	34%
	away	32%	68%

30-minute drive time prediction with probabilistic schedule

		Inferred	
		home	away
Actual from GPS	home	66%	34%
	away	33%	67%

60-minute drive time prediction with probabilistic schedule

		Inferred	
		home	away
Actual from GPS	home	65%	35%
	away	33%	67%

90-minute drive time prediction with probabilistic schedule

Table 6: **Probabilistic Schedule + Drive Time.** Adding information about the participants' distance from home slightly improves the performance of the probabilistic schedule algorithm.

Acknowledgments

We thank our paper's shepherd, Alexander Varshavsky, for his careful review and comments.

References

1. Gupta, M., S.S. Intille, and K. Larson. *Adding GPS-Control to Traditional Thermostats: An Exploration of Potential Energy Savings and Design Challenges*. in *Seventh International Conference on Pervasive Computing (Pervasive 2007)*. 2009.
2. Pesaran, A., A. Vlahinos, and T. Stuart, *Cooling and Preheating of Batteries in Hybrid Electric Vehicles*, in *6th ASME-JSME Thermal Engineering Joint Conference*. 2003.

3. Ashbrook, D. and T. Starner, *Using GPS to Learn Significant Locations and Predict Movement across Multiple Users*. Personal and Ubiquitous Computing, 2003. 7(5): p. 275-286.
4. Patterson, D.J., et al., *Inferring High-Level Behavior from Low-Level Sensors*, in *UbiComp 2003: Ubiquitous Computing*. 2003: Seattle, WA USA. p. 73-89.
5. Krumm, J. and E. Horvitz, *Predestination: Inferring Destinations from Partial Trajectories*, in *UbiComp 2006: Ubiquitous Computing*. 2006: Orange County, CA USA. p. 243-260.
6. Mozer, M.C., L. Vidmar, and R.M. Dodier, *The Neurothermostat: Predictive Optimal Control of Residential Heating Systems*, in *Advances in Neural Information Processing Systems 9*. 1997. p. 953-959.
7. Gao, G. and K. Whitehouse, *The Self-Programming Thermostat: Optimizing Setback Schedules based on Home Occupancy Patterns*, in *First ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings*. 2009: Berkeley, CA USA.
8. Szalay, A., et al., *Indexing the Sphere with the Hierarchical Triangular Mesh*. 2005, Microsoft Research, MSR-TR-2005-123.
9. Carroll, J. *Workers' Average Commute Round-Trip Is 46 Minutes in a Typical Day*. 2007 [cited 2010; Available from: <http://www.gallup.com/poll/28504/Workers-Average-Commute-RoundTrip-Minutes-Typical-Day.aspx>].
10. Davidoff, S., J. Zimmerman, and A.K. Dey, *How Routine Learners can Support Family Coordination*, in *28th ACM Conference on Human Factors in Computing Systems (CHI 2010)*. 2010: Atlanta, Georgia, USA.