A Statistical Reasoning System for Medication Prompting

Sengul Vurgun Matthai Philipose Misha Pavel*

Intel Corporation *Oregon Health and Science University

Abstract. We describe our experience building and using a reasoning system for providing context-based prompts to elders to take their medication. We describe the process of specification, design, implementation and use of our system. We chose a simple Dynamic Bayesian Network as our representation. We analyze the design space for the model in some detail. A key challenge in using the model was the overhead of labeling the data. We analyze the impact of a variety of options to ease labeling, and highlight in particular the utility of simple clustering before labeling. A key choice in the design of such reasoning systems is that between statistical and deterministic rule-based approaches. We evaluate a simple rule-based system on our data and discuss some of its pros and cons when compared to the statistical (Bayesian) approach in a practical setting. We discuss challenges to reasoning arising from failures of data collection procedures and calibration drift. The system was deployed among 6 subjects over a period of 12 weeks, and resulted in adherence improving from 56% on average with no prompting to 63% with state of the art context-unaware prompts to 74% with our context-aware prompts.

1 Introduction

A context-based prompt is a message delivered to a person because their physical context satisfies some pre-determined criterion. Such prompts have long been considered a service that could be provided by ubiquitous computing systems. A key part of any context-based prompting system is the reasoning module, which infers high-level user context based on sensor data and determines when to issue prompts. Much has been written on how to infer relevant context and how to integrate it into a reminder system, but little empirical work has tested these ideas over long periods of time on non-researchers to solve particular problems. In this paper, we describe the design, implementation and use of a reasoning engine that prompted 6 elders in their home to take their medication over a deployment of 12 weeks, based on two carefully chosen kinds of context. Although the reasoning system was deliberatly simple in design, we believe the pragmatics of developing and using it to (successfully) complete its mission should be of direct interest to the Ubicomp community.

A real-world deployment of a reasoning-system may be valuable in many ways. First, although many techniques have been proposed for context-awareness, there is not much evidence whether they yield sufficient performance for practical

applications. Applying such systems directly puts their utility to test. Second, techniques proposed have varied widely in their degree of sophistication and infrastructure use. A realistic deployment allows us to evaluate empirically the design space of solutions and determine whether various technical capabilities are worthwhile. In particular, real-world data often contains peculiarities that could serve either as a challenge or a justification for advanced techniques. Third, pragmatic difficulties in using techniques are often underplayed unless they are used at scale. A deployment should reveal major challenges of this kind. Finally, such deployments may reveal fresh challenges that either motivate new techniques or demand ad-hoc solutions of potential interest to practitioners.

We use our deployment experiences to make the following contributions:

1. We show that that simple reasoning techniques, when operating on data from fairly conventional wireless sensors, can indeed produce a useful end-result in an important application. In particular, average adherence rates across our subjects increased by 32% relative to no prompting at all, and 17% relative to state-of-the art time-based prompting.
2. Starting with a conventional statistical representation (the Dynamic Bayesian Network (DBN)) for processing time series data we present a detailed quantitative exploration of the design space for the structure of the DBN. As an extension of the exploration, we show that temporal reasoning does contribute crucially to the performance of our system.
3. We identify the overhead of labeling data as by far our biggest impediment to using the DBN. We explore ways to mitigate labeling overhead, including the impact of labeling different fractions of the training data available to us and using a simple semi-automatic labeling system.
4. We present a comparison between a reasoning system based on simple Bayesian reasoning and that based on simple rule-based reasoning. To our knowledge a direct comparison of these two approaches is rare, and perhaps unsurprisingly our insights support claims from supporters of both approaches. We reflect on the pros and cons of the two approaches in the context of our real-world deployment setting.
5. We identify unexpected challenges including miscalibration of sensors over time and faulty data collection procedures, and describe how we countered them.

The reasoning system described was part of a larger project called Context Aware Medication Prompting (CAMP). The CAMP project was not intended to showcase advanced reasoning techniques. The engineering goal in building CAMP was to provide conservatively designed sensing, communication, data management, reasoning, interaction and logistic support to validate medication adherence hypotheses on a tight schedule. In particular, at every stage of design of the reasoning system, we took pains to simplify requirements, design and implementation to maximize chances of success and minimize resource (time and staffing) requirements while providing performance adequate to the task. In some cases, these pragmatics make for a reasoning system that is less intricate

than one designed to illustrate novel reasoning capabilities: the reasoning problem itself is deliberately simple, and the tools we used are deliberately over- or under-provisioned. The focus of this paper is therefore on presenting conservative engineering that proved effective in practice rather than presenting novel or intricate design.

2 Related Work

There are few examples of longitudinally deployed ubiquitous computing applications that reason about user context. One outstanding exception is the Independent LifeStyle Assistant (ILSA) from Honeywell [6, 5], which deployed a variety of sensors in 11 elder homes over 6 months. ILSA delivered alerts to both elders and caregivers in an attempt to improve elders' medication adherence and mobility. It is unclear whether ILSA succeeded in this primary goal. No detailed description or quantitative analyses have been presented on the design space or efficacy of various parts of the ILSA reasoning system. On the other hand, ILSA involved sophisticated AI machinery, including agents, plan trackers and a variety of machine learning schemes. One of the primary post-mortem recommendations was to avoid most of these complexities. Our work, which presents a simple design that yielded a successful outcome, is a beneficiary of some of these insights. Further, we believe that the detailed quantitative evaluation we present should be of substantial additional value to the practitioner.

An extensive literature exists on sensors and reasoning techniques for inferring user context including location [7, 17], activities [13, 18, 12, 20], interruptibility [8, 4] and affect [11]. These efforts focus on developing (often sophisticated) techniques to handle limitations in existing systems. Common themes include the use of machine learning techniques to learn models and the use of a few representations such as Bayesian Networks, Support Vector Machines and boosted ensembles. To our knowledge, none of these techniques were deployed as part of longitudinal applications. Our work may be regarded as an early application of simple versions of these techniques in a realistic setting. We focus on how to produce adequate models using these techniques, and how to minimize the overhead of using them.

Labeling has long been recognized as a bottleneck to scaling machine learning. Our work provides empirical support for the importance of reducing the overhead of labeling; it is in fact not practical for us to label sufficient data by hand. Proposed solutions include semi-supervised learning [21] (which utilizes unlabeled data in addition to hopefully small quantities of labeled data), active learning (where users are queried for profitable labels) [2], the use of prior information [16] and clustering data automatically before labeling aggregate clusters instead of individual data points. We adapt the latter idea because of its simplicity: we present a simple interactive approach to labeling that groups similar data before presenting it for labeling.

The question of how and when to prompt subjects most effectively has been examined extensively both in the attention sensitivity [9, 4] and the interaction

planning [8, 3, 19] communities. One focus of the former work is identifying when users are most receptive to prompts and how to identify this with sensors. The latter considers how to jointly perform state estimation and identify optimal sequences of actions under uncertainty of observation and effect. In our work, we focus on identifying (using sensors) simple cues that subjects are receptive. However, based on ethnographic work we discuss below, we restrict ourselves to providing point (i.e., not multi step) reminders to users without explicit cost/benefit reasoning.

3 Context-Aware Prompting Requirements

Our reasoning system was built to support a project called Context Aware Medication Prompting (CAMP). One of the two hypotheses that CAMP was designed to test is that *automated contextual prompting can significantly improve medication adherence* (compared to state-of-the art techniques). The state of the art in medication prompting are medication dispensers that beep loudly at fixed times, dispense medication and in some cases, verbally prompt the elder to take medication. Although these devices do improve adherence significantly, there is still a residual lack of adherence. Based on extensive formative work, CAMP ethnographers and domain experts noted a variety of reasons limiting these devices. Based on an iterative process between ethnographers and engineers on the team, two particular failure modes were chosen to be addressed using context aware techniques. When time-based prompts go off, the elder:

1. May not be at home. Prompting the elder to take their medication before leaving the home (if appropriate) could be useful.
2. May be sleeping, on the phone, or engaged in activity away from the medication dispenser. It could be useful to deliver the prompt when the elder is close to the dispenser and neither sleeping or on the phone.

Our reasoning system is therefore designed to track two pieces of context about the user: whether they are about to leave the house, and which room of the house they are in. Use of phone, whether the elder is sleeping and whether the elder took their medication was inferred deterministically with custom sensors and we will not discuss these much further. Below, we go into these requirements in more detail.

3.1 Rules for Prompting

CAMP researchers distilled the functionality of the prompting system into a set of rules. These rules were executed within the pill taking window, in our case, 90 minutes before and after the recommended time to take the medication.

1. Never prompt outside the window.
2. Don't prompt if pill is already taken within the current window.
3. Don't prompt if the participant is not home. Prompting will resume if the participant returns home before the window expires.

4. Don't prompt if participant is in bed.
5. Don't prompt if participant is on the phone.
6. Prompt at level 2 if participant is leaving (this is the only time we prompt before the usual pill taking time).
7. Wait till the time the user *usually* takes the pill. If it is earlier than the recommended pill taking time, start checking for level 1 prompting opportunities at the usual pill time.
8. If only less than 20 minutes left till the window expires, start prompting at level 1 disregarding all other rules (except 1-3).

The system supported two kinds of prompting:

- Level 1: Prompt using the nearest device every minute. The chime is played 10 seconds each time and lights stay on till location changes. Stop if pill is taken. Escalate to level 2 after 10 minutes.
- Level 2: Prompt using all prompting devices in the house every minute. Lights on devices stay on and chime is played for 10 seconds every minute.

The team briefly considered a planning-based approach to the reasoning and prompting engine, where relevant states of the world (elder, home and prompting system), possible actions and their costs, and the likely results of actions would be encoded in a representation like a (Partially Observable) Markov Decision Process (POMDP)[10]. However, we decided on a deterministic rule-based implementation of the prompter for two reasons:

- It was much simpler for engineers and ethnographers to agree on the rules than on costs, and to implement the tens of lines of dispatch code. This was especially so because we decided against sophisticated sequences of prompts.
- Although we initially thought that minimizing user annoyance would be a crucial and subtle aspect of prompting (and therefore worthwhile for a sophisticated technique to maximize the value of actions), formative work found that elders' tolerance threshold to medication reminders was surprisingly high. In fact, with time-based devices, they were comfortable with, and to some extent preferred, loud and persistent reminders.

3.2 Subjects and Infrastructure

To test CAMP hypotheses, we recruited elders who were known to be at risk for medication non-adherence from a prior study from two apartment complexes. Twelve subjects over the age of 70, 10 women and 2 men agreed to participate in the study. No subjects were currently receiving formal long-term care, so they are not completely (or mostly) devoid of cognitive and physical abilities. All subjects lived on their own. Figure 1 shows the typical layout of an apartment, along with the sensor deployment.

Sensors installed were mostly stock devices. They included 4 to 6 wireless infra-red motion sensors (roughly one per room or major area in the home), a pressure mat on the bed, contact sensors on apartment doors and refrigerator

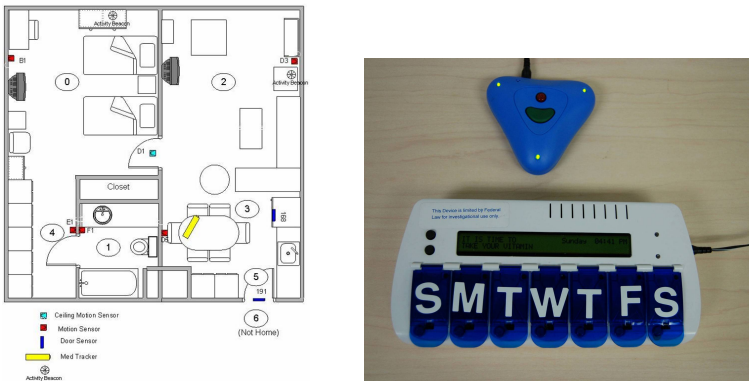

Fig. 1. Typical Floorplan (left); Activity Beacon (top) and MedTracker (bottom)

doors, and sensor for reporting phone use. Figure 1 shows two devices built specifically for the study.

The MedTracker is a pill box that allows pills to be allocated into individual compartments for a whole week. Sensors on the lid for each day of the week can detect if the lid is opened and wirelessly communicate this information to the computing device in the house. In our study, we assumed that the MedTracker provided direct evidence on whether a pill is taken: if the lid for the current day is opened during a period when a medication is supposed to be taken, we assume that the subject successfully took the pill. This indirect notion of adherence is in line with existing practice in such studies. Although there are many documented cases of subjects misplacing pills taken out of a bottle, an informal check revealed that this was rare in our case, perhaps because we ensured that the subjects had reasonable cognitive abilities. The MedTracker is also capable of beeping, flashing and delivering a text message on an LED.

The activity beacon is a wireless, battery backed-up device the size of a saucer that can be placed easily at points around the space being monitored. It is capable of flashing a light, beeping and delivering an audio message. Both the MedTracker and the activity beacon serve as prompting devices.

3.3 The Experiment

Subjects were required to take a vitamin C pill three times a day, morning, mid-day and evening at a fixed time with a 90 minute window allowed on either side of the prescribed time.

We installed sensors, reasoning system and actuators in the apartments for a period of 28 weeks on average. Our original intention was to have a 6-week baseline period where infrastructure would be installed but no prompts would be delivered, followed by two 4-week stretches where subjects would get prompts either from a time-based prompting system or from the context-aware system.

The baseline period would be used to evaluate adherence level with no intervention as well as to construct an appropriate model for the user, which could be used during the subsequent context-based prompting period. In practice, because of initial problems with the infrastructure, we spent 7-16 weeks in baseline followed by 12-15 weeks of intervention.

The original group of 12 subjects dwindled to 6 during the baseline period, so that we were able to perform prompting only on the latter smaller group. We will refer to these subjects by the labels HP05, HP52, M26, M32, M44 and M45. Most of the drop-offs were due to personal reasons (e.g. sickness, marriage).

3.4 Modeling Choices

Our final inference tasks (inferring location and whether leaving home) were carefully selected so that they were likely to provide useful reminders to users while still being fairly directly inferable from our sensors. For instance, we expected that motion sensor readings would tell us subject location most of the time, and that a location next to the front door of a home coupled with the opening of the door would indicate whether the user is leaving home. However, we also expected a number complications:

- Motion sensors readings are often only indirect indicators of subject location. In particular sensor lockout periods, thresholds for activity levels before triggering and detection of events in adjacent areas through open doors all result in sensors firing or failing to fire unexpectedly (relative to subject motion). Techniques that reason about uncertainty and noise have therefore proved valuable in inferring location from motion sensors [14].
- Contact sensors, such as those for the front door and refrigerator are prone to missing events especially when installed imperfectly or when misaligned due to common use. It is important therefore to make inferences with incomplete information.
- Given noise in sensor data, it is possible to get contradictory information from multiple sensors. For instance, the bed sensor may indicate that the subject is on the bed while the kitchen sensor fires in the same time slice or the refrigerator door sensor triggers (due to vibrations). It is important therefore to weigh evidence from different sensors when inferring the final result.
- In some cases, the duration of stay in a particular state is important. For instance, a subject in the passage way next to the door may be much more likely to leave if they spend more than a few seconds there.
- In all cases, we expect considerable variability in layout of homes and behavior of subjects. We therefore expected some level of customization to each subject to be important.

The choice of reasoning technique needed to be made months before actual data from elderly subjects was available in order to allow for implementation, testing and integration with CAMP infrastructure. The above concerns about

noise and variability in the sensor data led us to select a statistical (Bayesian) approach as opposed to a deterministic rule-based one. The decision came with a risk: much of the design exploration work and all implementation work for the CAMP reasoning system was to be done with an engineer with little prior experience with statistical reasoning. The engineer worked with two experienced users of Bayesian techniques as occasional advisers, based on a two-week crash course in Bayesian Networks. The learning curve for a "heavyweight" technique such as Bayesian network was a serious concern. We were pleasantly surprised to find that as long as we limited ourselves to simple structures, the Bayesian approach corresponded closely to intuitive rules.

Before the deployment, and based partially on data from a trial with a friendly subject, we defined the structures for a family of models for the elders' behavior. During the baseline period, we trained these models on roughly 100 hours of data per subject spread over a week using leave-one-week-out cross validation (with 5 folds) and picked the best performing one for each subject. The model that performed best for each subject at baseline was used during their intervention period. Training originally involved substantial labeling overhead, of the order of 1 day for each day labeled. Section 4 below details the process of finding good models, and section 5 describes how we addressed the cost of labeling.

4 Selecting Models

Table 1 lists the inputs and outputs for our context model. The outputs (termed *hidden variables*) are the location of the user and a boolean variable indicating whether they are about to leave their home. The inputs, or*observed variables*, correspond to information pooled from differently sized time windows preceding the current moment. The time windows and the information represented by the observables were selected based on experience. We track the last motion sensor fired because there are runs of time slices with no motion sensor information. In these cases, we found that the last motion sensor fired is a good indicator of current location. In our initial design, we instead tracked the motion sensor that fires in the current time slice (and allowed a NoSensor value). The two other MS variables track the "level of activity" in the home because we believed that high levels of activity may correlate with intent to leave the home.

4.1 A Dynamic Bayesian Model

Model 1 of Figure 2 shows our basic model, a Dynamic Bayesian Network (DBN) [15]. Nodes in the graph correspond to hidden and observation variables in a 5-second time slice. We choose to infer at this granularity because we have a narrow window of 10 or more seconds when a subject is leaving the home. Each node n has a *conditional probability table* (CPT, not shown), which represents the probability distribution $\Pr(n|\mathrm{Pa}(n))$, where $\mathrm{Pa}(n)$ are the parent nodes of n. The dotted line separates values of variables in two adjacent time slices, with the

Table 1. Model Variables and Their Possible Values

Variable	Values	Comment
Location	NotHome, Bedroom, Kitchen, Livingroom, Bathroom, Frontdoor	Hidden variable
Leaving	True, False	Hidden variable
LastMSFiring	MS1, MS2, ..., MSN	Which motion sensor (MS) got fired last?
MSFiringsFreq	None, L(1-2), M(3-5), H (more than 5)	Num. MS firings in last 1 minute
MSTransitions	None, L(1-5), M(6-10), High(more than 10)	Num. MS transitions in last 5 minutes
Bed	In, Out	
DoorEvent	OpenEvent, CloseEvent, NoEvent	
Refrigerator	OpenEvent, CloseEvent, NoEvent	
Time	EM (6-9am), MM (9-11am), Noon (11-2pm), A (2-5pm), E (5-9pm), Night (9pm-6am)	

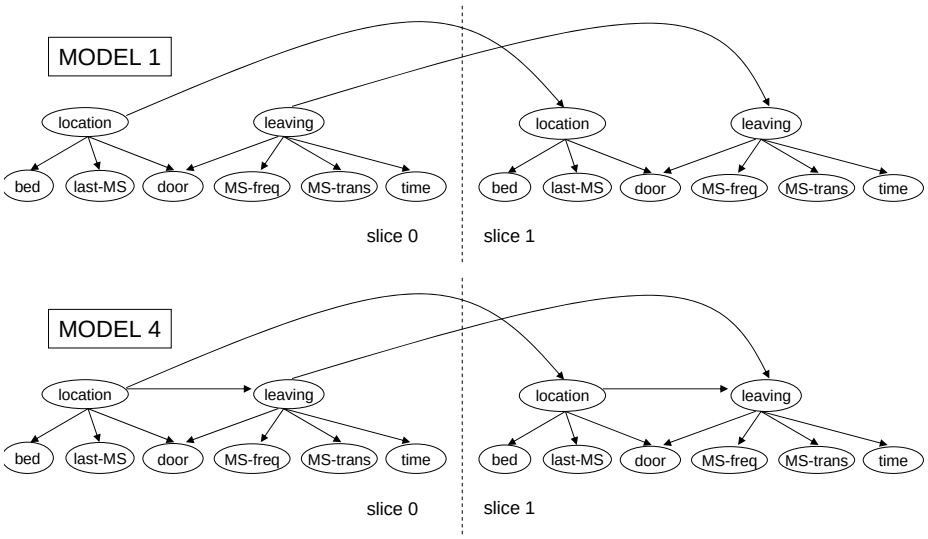

Fig. 2. The Baseline (Model1) and Best (Model4) DBNs

left side representing the current time slice and the right representing the next time slice. Arrows across the boxes represent temporal conditional dependences between variables.

Regardless of the values in the CPT, this model encodes the following assumptions:

1. The Bed, LastMSFiring and Refrigerator variables depend just on Location. Once the Location is known, the subject's Leaving status has no effect on these. Similarly for MSFiringFreq, MSTransitions and Time with respect to Leaving.
2. DoorEvent depends in the same sense on both Location and Leaving, since if you are located close to the door and not leaving, you will likely not open the door, and if you are leaving but your current location is not next to the door, you will again not open the door.
3. Location and Leaving in a given time slice are independent of all other variables in the previous time slices given their values in the previous time slice.

This model is one of the simplest dynamical models using our variables: it is very close to one Naive Bayesian model for each hidden variable with a temporal constraint on hidden variables thrown in to smooth their values over time.

Table 2. Best Classifiers for Leaving (% Correct Averaged Over Folds)

House	Model	Leaving=true	Leaving=false	Loc=AtHome	Loc=NotHome
HP05	Model3	0	95.66	96.87	18.73
HP52	Model4	86.89	92.69	98.06	95.67
M26	Model4	92.00	88.33	90.96	77.72
M32	Model4	77.29	97.60	96.99	98.42
M45	Model4	90.73	90.76	97.86	95.01

We experimented with a few variants on this basic structure encoding slightly different sets of assumptions. Model 4, also shown in Figure 2, was the best performing of all our models when applied to subject data during the test period. It encodes an additional dependency between Leaving and Location. This dependency was crucial in the detection of leaving, because in its absence (e.g., in model 1), leaving has no access to either the hidden location or its determining sensors. Since leaving is a combination of being located near the door followed by opening the door, it is essentially impossible to determine without location information. For one of our subjects, a slightly different model (named Model 3) was the best performing. Exploring the design space of these DBNs by adding dependence arcs between random variables proved to be surprisingly powerful. However, we should note that we stopped reasoning about every conditional

independence encoded in the DBN (in particular verifying whether various V-structures were sensible) early in our explorations. We simply drew an arrow between a hidden node and an observed node when the latter depended on the former. We expect that more sophisticated DBNs where encoding correct conditional independence structure is key would be much more resource-intensive to develop.

Table 3. Results from Non-Temporal Classifiers (% Correct)

House	Model	Leaving=true	Leaving=false	Loc=AtHome	Loc=NotHome
HP05	Model3 (Fold0)	0	95.66	96.87	18.73
HP05 NT	Model3 (Fold0)	0	95.26	99.53	0
HP52	Model4 (Fold1)	96.97	93.51	99.64	99.23
HP52 NT	Model4 (Fold1)	0	92.77	99.82	99.29
M26	Model2 (Fold3)	100.00	89.86	93.74	80.28
M26 NT	Model2 (Fold3)	0	87.37	94.64	79.86
M32	Model4 (Fold1)	100.00	98.67	98.00	98.58
M32 NT	Model4 (Fold1)	0	98.46	97.78	98.66
M44	Model4 (Fold3)	100.00	93.32	99.55	98.14
M44 NT	Model4 (Fold3)	0	83.44	96.97	98.82
M45	Model4 (Fold2)	100.00	92.31	98.99	99.70
M45 NT	Model4 (Fold2)	0	91.36	98.99	91.37

We used the above analysis to select, for each subject, the appropriate model (of five possibilities) for use intervention phase of CAMP. Table 2 shows the true positive and true negative rates of the model that performed best on classifying Leaving (all models did quite well on Location) for each subject. Although for compactness we show Location results as an AtHome/NotHome classifier, we actually performed an N-way classification over the rooms in the house, and the numbers reported ar the results of the N-way classification. HP05 turns out to be an anomalous case: it had very few (4) leaving examples and less data overall than the others because the subject spent much her time at her friend's home.

4.2 Dropping Temporal Information

Although our intuition was that temporal reasoning (i.e., incorporating reasoning from past time slices in the present) would contribute strongly to performance, we tested a model that omitted the temporal arrows in the DBN so that we

had a conventional Bayesian Network to classify each time slice. Table 3 shows the results of applying this model on a single folds (we did not validate over all folds due to time; we picked the fold with the best Leaving result for the temporal model). For instance HP52 data is analyzed with Model 4 on Fold1 of the data; the "NT" line gives results without temporal dependences. Location is classified quite well even without temporal dependences. This primarily because LastMSFired is an excellent indicator of current location. Although it may seem surprising that the Bayes Net uniformly resulted in zero detection of labeling (for instance, one would expect at least an occasional guess for leaving when door opens), an examination of the learned networks reveals that that this was because of the prior bias towards not leaving the house; leaving is a rare event.

5 Using the Model

5.1 Implementation

We implemented our model in C++ using the Probabilistic Network Library (PNL) [1] toolkit for graphical modeling. We perform inference by filtering with a junction tree algorithm and stick to fully supervised parameter learning. The models were easy to implement and reason with, involving under a hundred lines of code. However, the toolkit itself was not mature and required debugging.

5.2 Labeling

The biggest bottleneck in using our models was to label data so that parameters of the DBNs could be learned on the basis of observed data. Purely manual labeling of all 5 folds of data in each case was unsustainable because labeling an hour of data often took roughly an hour. In what follows, we examine a simple semi-automated labeling technique, the impact of labeling (and learning with) less data. We also considered (but do not report here) the potential for transferring models across homes.

Interactive Labeling After labeling manually for a few days, we noticed that the labels remained unchanged for long stretches. In particular, in the absence of observations or if observation values were unchanged, labels did not change. Alternately, segmenting the time series data by missing or identical observations resulted in segments with unique labels. We therefore decided to segment the data before presenting to the user for labeling.

Algorithm 1 specifies the rules for labeling location. We assume that location remains fixed over time unless an observation is detected from a sensor in a different room. If such an observation is detected, we give the user to provide a new label. Note that in some cases because of noise in the sensors, it is incorrect to simply label with the location of the sensor that generated the new observation. We have a similar scheme for labeling Leaving.

Algorithm 1 InteractiveLabel(s)

Require: A list s of sensor events.

```
set l to unknown
for all events e_i in s do
    if room in which sensor for e_i is located is l then
        label e_i with l
    else
        display e_i ... e_{i+10}
        if user labels e_j with location l' ≠ l then
            label e_i ... e_{j-1} with l
            set l to l'
            continue loop at event e_j
        else
            label e_i with l
        end if
    end if
end for
```

Table 4 shows the degree to which the tool can cut labeling overhead. For each house, the table lists the number N of events to be labeled, the number M of events for which the tool requests manual help, and ratio of M to N. The tool reduced labeling requirements by 1 to 2 orders of magnitude, and in practical terms made the difference between being able to train our DBNs and not.

Table 4. Reduction in Manual Labeling Using Labeling Tool

House	#events labeled (N)	#hand(M)	$\frac{M}{2*N}$(%)
HP05	45105	1871	2.07
HP52	34333	839	1.22
M26	66237	2365	1.79
M44	63979	941	0.74
M45	54909	6031	5.49

The reduction brings up the question of whether labeling could have been done away with completely using further heuristics. Note however, that the success of the above segmentation algorithm depends wholly on having the correct label at the points where a new label is introduced. The key question therefore is whether the "challenging" events that were manually labeled by the human can be automatically labeled using (simple) rules. To understand this better, we implemented a simple set of rules that sets the location of a time slice to the location of the last bed, refrigerator or motion sensor (tried in that order) fired in that time slice. We declare that the subject is leaving if their location is

Frontdoor and we see an OpenEvent. If no sensor readings are seen for $n = 30$ seconds, then the user's location is set to their last computed location; if the last location was Frontdoor, then we set the location to NotHome. Table 5 shows the results.

Overall, the rule-based system does quite well; in fact it often has higher true negative and true positive rates for Leaving and Location = AtHome than the Bayesian system does. However, it has a few failure modes, which result in significantly lower true positives and true negatives on Leaving and Location respectively. Note that missing instances of Leaving is especially debilitating because it results in missed opportunities to prompt the user. The failures occur for the following reasons, all having to do with sensor noise. First, because of anomalous motion sensor firings away from the front door while the door is being opened (e.g., in M26 the kitchen sensor near the front door fired after the front door OpenEvent) the rule-based system concludes that the subject is not leaving after all. This results in missing Leaving = true cases. Second, after the user actually leaves in this case, since the last observed sensor is not Frontdoor, the location is set to the last sensor seen (e.g., Kitchen for M26) as opposed to NotHome. This results in missed cases of Location=NotHome. Finally, the DoorOpen sensor message is occasionally missed (e.g., in M44); the rules therefore do not detect Leaving = true; interestingly the Bayesian Network was able to infer just from the fact that the Location=FrontDoor that Leaving=True was likely for the user.

Table 5. Results of Stochastic vs. Rule-Based (RB) Systems (% Correct)

House	Model	Leaving=true	Leaving=false	Loc=AtHome	Loc=NotHome
HP05	Model3	0	95.66	96.87	18.73
HP05 RB	Model3	50	99.54	99.59	25.81
HP52	Model4	96.97	93.51	99.64	99.23
HP52 RB	Model4	78.79	98.66	99.9	99.7
M26	Model2	100.00	89.86	93.74	80.28
M26 RB	Model2	71.43	98.78	99.6	9.18
M32	Model4	100.00	98.67	98.00	98.58
M32 RB	Model4	20	98.93	99.8	13.64
M44	Model4	100.00	93.32	99.55	98.14
M44 RB	Model4	77.78	98.8	99.88	26.08
M45	Model4	100.00	92.31	98.99	99.70
M45 RB	Model4	12.5	99.14	99.17	31.95

This brief analysis shows that the requirement of dealing with noise can complicate rule-based systems. We do not make any claims about the superiority of the statistical approach to the deterministic one, since it is possible that a few simple extensions to the existing rules may suffice to substantially improve performance. However, we also note that the overhead of using the simple Naive-Bayes type Bayesian network was low enough (after the initial 2-week crash course and with the interactive labeling tool) that we think it unlikely that a good set of rules would be substantially easier to develop.

Labeling Data Partially Another option to reduce labeling overhead is to label only as much data as is useful. Excess labeled data can lead to over-fitting. Table 6 shows the result of learning models using only a fraction of the data from each fold. Due to time constraints, these numbers are from a single fold. We trained model 4 on first 10, 35, 60 and 100% of the data from the fold. It seems that we could have gotten away with labeling roughly half of the data we did label. The savings are, however, small relative to interactive labeling. It is possible, that if we had used the unlabeled data for learning (using unsupervised learning techniques), we could have gotten acceptable performance with below 35% of the labels. An order of magnitude reduction seems unlikely, though.

Table 6. Inference Results for M32 and M45 With Partial Labeling (% Correct)

% Training Data	Leaving=true	Leaving=false	Loc=athome	Loc=nothome
10	0	79.72	77.67	98.42
35	0	99.26	98.36	97.95
60	100	98.89	98.06	98.66
100	100	98.67	98.00	98.58

% Training Data	Leaving=true	Leaving=false	Loc=athome	Loc=nothome
10	0	73.23	69.34	98.48
35	100	91.66	99.07	99.74
60	100	92.73	99.12	99.81
100	100	92.31	98.99	99.70

5.3 Other challenges

The deployment posed a variety of unexpected challenges beyond the expected ones of model selection and labeling. Two particularly worth mentioning are drift in sensors and anomalous data due to infrastructure errors. Figure 3 shows data from one of our bed sensors. Note that the value of the sensor when no one

is on the bed (e.g., between 0800 and 1400 hours) drifts downwards substantially even during the course of one day. This was a common occurrence with our bed sensors. Since we convert the bed sensor into a binary sensor (In, Out, as per Table 1) by thresholding, it is important for us to recompute thresholds if the baseline drifts too far downwards. We opted to take low-tech approach to the problem: an engineer monitored the baseline signal relative to threshold for each house every day and reset the threshold manually if needed. We had to perform this operation just once over all houses during the deployment. Of course, engineers performing manual thresholding does not scale and some unsupervised thresholding scheme is in order here.

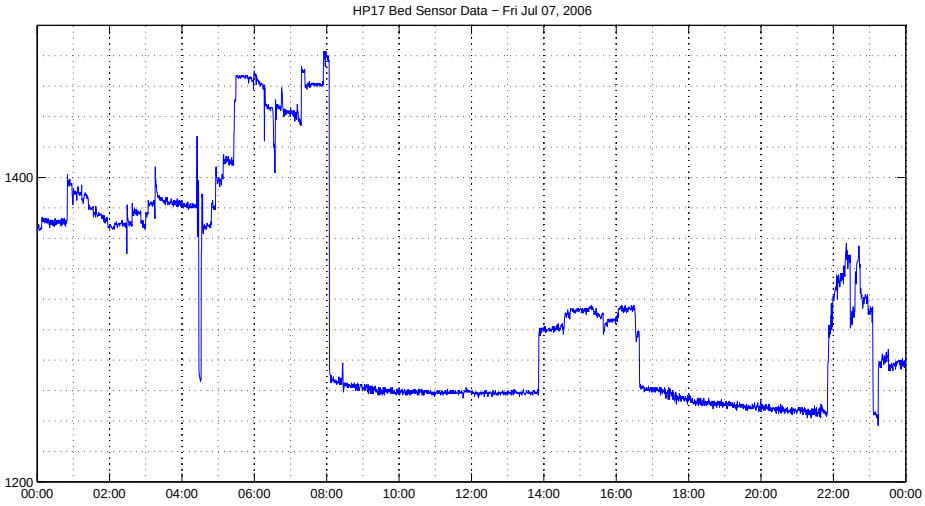

Fig. 3. Drift in Bed Sensor Calibration

A second challenge that recurred was the occasional corruption of data due to sensor and connectivity problems, and also because in some cases our maintenance staff entered homes without logging that they did so. We handled this problem manually by scanning through visualizations of the data looking for telltale signs such as an excess of reset messages and evidence of multiple people in the house. Reliable computer readable documentation of these sources of anomaly would have noticeably reduced the burden of training.

6 End-to-End Results

The end goal of the reasoning system was to produce context-aware prompts that enhanced the subjects' medication adherence. We counted a subject as having taken their pill if they opened the appropriate compartment of the MedTracker

Table 7. Change in Adherence Rates

Participant	Baseline%	Time-Based%	Context-Aware%
HP05	33.3	69.1	54.2
HP52	75.8	70.2	84.9
M26	65.8	71.3	81.6
M32	47.7	77.0	93.1
M44	N/A	45.7	48.0
M45	58.3	46.1	81.8
avg.	56.2	63.2	73.9

pillbox during the 3-hour period. We measured adherence in this manner during the baseline, conventional (time-based) prompting period and the context-based prompting periods. Table 7 shows the results. In every case except HP05, context-based prompting improved over no prompting and time-based prompting, often substantially. It is not surprising that HP05 decreased in adherence, since she took to spending long periods outside her home (caring for a friend) after the baseline period. Baseline data for M44 is not available because we discovered at the end of the baseline that the MedTracker had been malfunctioning.

7 Conclusions

We have described the specification, design, implementation and use of a reasoning system for prompting subjects to take their medication in a context-sensitive manner. The system was deployed longitudinally with 6 elderly subjects and resulted in significant increase in adherence rates among most of these subjects. We provide a detailed account of the pragmatics of using conventional statistical reasoning techniques in the real world, starting with utilizing domain constraints to simplify the problem as far as possible, using sensors that are strongly correlated with hidden variables, performing an exploration of the space of possible models, using simple but effective techniques to minimize labeling and handling a variety of other problems related to real-world deployment. Although the description of a system sufficient for producing significant results in an important application is itself of potential interest to Ubicomp practitioners, our detailed analysis of design choices may be of especially strong interest.

8 Acknowledgements

This paper would not have been possible without the work of the CAMP team: Stephen Agritelley, Kofi Cobbinah, Terry Dishongh, Farzin Guilak, Tamara Hayes, Jeffrey Kaye, Janna Kimel, Michael Labhard, Jay Lundell, Brad Needham, Kevin Rhodes and Umut Ozertem.

References

1. Open Source Probabilistic Networks Library. *https://sourceforge.net/projects/openpnl/*.
2. B. Anderson and A. Moore. Active learning for hidden markov models: objective functions and algorithms. In *ICML*, pages 9–16, 2005.
3. J. Boger, J. Hoey, P. Poupart, C. Boutilier, G. Fernie, and A. Mihailidis. A planning system based on markov decision processes to guide people with dementia through activities of daily living. *IEEE Transactions on Information Technology in Biomedicine*, 10(2):323–333, 2006.
4. J. Fogarty, S. E. Hudson, C. G. Atkeson, D. Avrahami, J. Forlizzi, S. B. Kiesler, J. C. Lee, and J. Yang. Predicting human interruptibility with sensors. *ACM Trans. Comput.-Hum. Interact.*, 12(1):119–146, 2005.
5. K. Z. Haigh, L. M. Kiff, and G. Ho. The Independent LifeStyle Assistant$^{\mathrm{TM}}$ (I.L.S.A.): Lessons Learned. *Assistive Technology*, 2006.
6. K. Z. Haigh, L. M. Kiff, J. Myers, V. Guralnik, C. W. Geib, J. Phelps, and T. Wagner. The Independent LifeStyle Assistant$^{\mathrm{TM}}$ (I.L.S.A.): AI Lessons Learned. In *AAAI*, pages 852–857, 2004.
7. Jeffrey Hightower and Gaetano Borriello. Particle filters for location estimation in ubiquitous computing: A case study. In *Ubicomp*, pages 88–106, 2004.
8. E. Horvitz, A. Jacobs, and D. Hovel. Attention-sensitive alerting. In *UAI*, pages 305–313, 1999.
9. E.ric Horvitz and J. Apacible. Learning and reasoning about interruption. In *ICMI*, pages 20–27, 2003.
10. L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.
11. A. Kapoor and R. W. Picard. Multimodal affect recognition in learning environments. In *ACM Multimedia*, pages 677–682, 2005.
12. J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *IJCAI*, pages 766–772, 2005.
13. Lin Liao, Dieter Fox, and Henry Kautz. Location-based activity recognition using relational markov networks. In *IJCAI*, 2005.
14. A. G. Adami H. B. Jimison J. Kaye M. Pavel, T. L. Hayes. Unobtrusive assessment of mobility. In *EMBS*, 2006.
15. Kevin P. Murphy. Dynamic Bayesian Networks: Representation, Inference and Learning. 2002.
16. Nuria Oliver, Barbara Rosario, and Alex Pentland. Graphical models for recognizing human interactions. In *NIPS*, pages 924–930, 1998.
17. Veljo Otsason, Alex Varshavsky, Anthony LaMarca, and Eyal de Lara. Accurate gsm indoor localization. In *Ubicomp*, pages 141–158, 2005.
18. M. Philipose, K.P. Fishkin, M. Perkowitz, D.J. Patterson, H. Kautz, and D. Hahnel. Inferring activities from interactions with objects. *IEEE Pervasive Computing Magazine*, 3(4):50–57, 2004.
19. M. E. Pollack, L. E. Brown, D. Colbry, C. E. McCarthy, C. Orosz, B. Peintner, S. Ramakrishnan, and I. Tsamardinos. Autominder: an intelligent cognitive orthotic system for people with memory impairment. *Robotics and Autonomous Systems*, 44(3-4):273–282, 2003.
20. E. M. Tapia, S. S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. In *Pervasive*, pages 158–175, 2004.
21. X. Zhu. Semi-supervised learning literature survey. *Computer Sciences TR 1530, University of Wisconsin, Madison*, 2005.