Predicting Human Interruptibility with Sensors: A Wizard of Oz Feasibility Study

Scott E. Hudson[†], **James Fogarty**[†], **Christopher G. Atkeson**[†*], **Daniel Avrahami**[†],
Jodi Forlizzi[†+], **Sara Kiesler**[†], **Johnny C. Lee**[†], **and Jie Yang**[†]
HCI Institute[†], Robotics Institute*, and School of Design[+]
Carnegie Mellon University, Pittsburgh, PA 15213, USA
{scott.hudson, jfogarty, cga, nx6, forlizzi, kiesler, johnny, jie_yang}@cs.cmu.edu

ABSTRACT

A person seeking someone else's attention is normally able to quickly assess how interruptible they are. This assessment allows for behavior we perceive as natural, socially appropriate, or simply *polite*. On the other hand, today's computer systems are almost entirely oblivious to the human world they operate in, and typically have no way to take into account the interruptibility of the user. This paper presents a Wizard of Oz study exploring whether, and how, robust sensor-based predictions of interruptibility might be constructed, which sensors might be most useful to such predictions, and how simple such sensors might be.

The study simulates a range of possible sensors through human coding of audio and video recordings. Experience sampling is used to simultaneously collect randomly distributed self-reports of interruptibility. Based on these simulated sensors, we construct statistical models predicting human interruptibility and compare their predictions with the collected self-report data. The results of these models, although covering a demographically limited sample, are very promising, with the overall accuracy of several models reaching about 78%. Additionally, a model tuned to avoiding unwanted interruptions does so for 90% of its predictions, while retaining 75% overall accuracy.

Keywords

Situationally appropriate interaction, context-aware computing, sensor-based interfaces, machine learning.

INTRODUCTION

As a part of our early socialization, human beings normally learn when it is appropriate to interrupt someone. As adults, we can typically assess someone's interruptibility very quickly and with a minimum of effort. For example, in the time it takes to walk past someone's open office door, we can often tell that we should not intrude on the person. This assessment allows us to balance the benefits of an interruption with its cost. Maintaining such a balance usually results in what is recognized as socially appropriate (or informally: *civil* or *polite*) behavior.

Unfortunately, computer and communications systems cannot currently act in a similar fashion – they are almost entirely oblivious to the human context in which they operate and cannot assess whether "now is a bad time." As a result, they operate the same way in essentially all situations, and do not act in ways that remain appropriate to the situation. If left unchecked, current systems can easily disturb or annoy – consuming the valuable resource of human attention in a haphazard and inefficient fashion. As a result, we often avoid building proactive systems – forcing our interfaces to be silent and passive until called upon.

If we could develop relatively robust estimators of interruptibility, we might enhance human-computer interaction and computer mediated communications in a number of ways – making people more efficient (and possibly even more relaxed). For example, we might build a "smart answering machine" which stopped our phone from ringing and diverted our other messaging traffic when we should not be interrupted. We might also be able to build information displays that could balance an estimation of the importance of a piece of information against the attentional costs of delivering it.

This paper describes work exploring the feasibility of creating such an estimator by using sensors to drive models that predict human interruptibility. This should be theoretically possible, since the equivalent assessments made by people are based on directly observable phenomena (rather than, for example, invisible internal cognitive state). However, replicating this kind of rich human judgment in practice may be very challenging and might even be currently impossible. The study described here seeks to assess the feasibility of this kind of automatic prediction based on sensor data. Specifically it seeks answers to at least these five questions:

- Can a practical sensor-driven model reliably predict human interruptibility?
- How can such a model be constructed?
- How accurate can we make such a model?
- Which sensors are most useful for such a model?
- What are the simplest sensors that will produce an accurate prediction?

CHI 2003, April 5–10, 2003, Ft. Lauderdale, Florida, USA.

We might proceed by creating and deploying sensors, then testing their effectiveness with various forms of models. However, given the large uncertainty surrounding these questions, it is almost inevitable that we would spend considerable effort to build sensors which in the end turned out to be ill-suited or sub-optimal for the task. Instead, we have chosen a Wizard of Oz approach that allows us to simulate a wide range of plausible sensors, build multiple models based on data from these simulated sensors, and then test the effectiveness of different models and different combinations of sensors.

Specifically, this study is based on a long-term digital audio and video recording of the working environment of a subject. These recordings were made during full working hours for 14-22 working days for each subject. The recordings were then viewed by a person who coded for actions and situations that could plausibly be sensed. For example, as detailed below, we recorded the number of people present, who was speaking, what task objects were being manipulated, whether the phone was off-hook, and other similar facts about the environment. Overall, we recorded 602 hours of audio and video from the office environments of four subjects with similar job functions.

During the time that recordings were being made we employed *experience sampling* techniques [5] (sometimes called a *beeper study*) to elicit in-situ self-reports of their interruptibility. Finally, a variety of machine learning techniques were used to create predictive statistical models that could use the simulated sensor data to predict the collected self-reports.

While currently limited to a fairly narrow demographic group, the results produced have been quite promising. When predicting overall interruptibility or non-interruptibility of subjects (as a binary decision), several models produced very similar results with a cross-subject accuracy of approximately 78% (compared to a base accuracy of 68%). We were also surprised to discover that much of that predictive power of several of these models could be obtained using a single, relatively easy to build, sensor that indicates whether anyone in the space is talking.

A primary use for an interruptibility estimator will likely be to structure information delivery to avoid situations when the user does not wish to be interrupted. In that setting it may be more important to avoid "incorrect" interruptions than it is to make sure not to miss opportunities for "correct" interruptions. We have constructed a model tuned to avoiding unwanted interruptions that does so for 90% of its predictions (while retaining 75% overall accuracy).

RELATED WORK

The study of interruptions began with classic experiments in the 1920s showing that tasks that were interrupted were more likely to be recalled after a delay than tasks that were not interrupted [17]. Much of the psychological literature on interruptions has been devoted to examining this effect [1, 7, 8], although recent research in HCI has sought to find what technological interventions might be best to negotiate multiple and sometimes complex tasks [9, 10, 11].

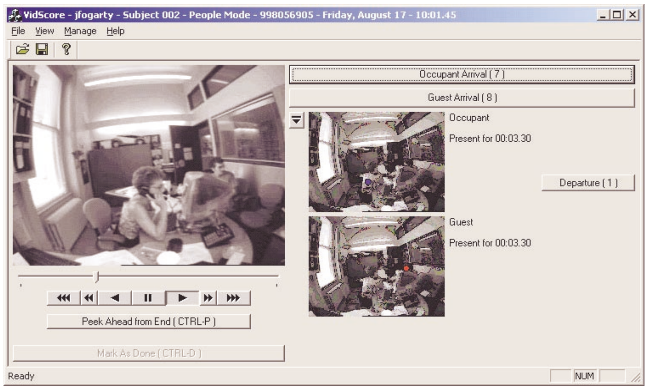


Figure 1. Custom Coding Interface Showing a Typical View of a Subject's Office

HCI researchers have only begun to provide a more analytic and precise approach to understanding interruption, so as to design better context-aware systems. Some researchers believe that understanding the context of interruptions cannot be handled successfully by machines, but instead machines must defer to users in an accessible and useful fashion [2]. Others, such as Horvitz [9], are optimistic that machine learning techniques can handle many of the predictions needed to present information appropriately.

Other researchers have worked towards design guidelines for coordinating interruption in HCI. McFarlane tested four known methods for deciding when to interrupt people [11]. Although the results have implications for structuring appropriate interruption in HCI, no one method emerged as best in doing so. O'Conaill completed an ethnographic study on the nature of interruptions in the workplace with implications for how to better design communication technology [13]. One major finding was that the recipients of interruptions often derive personal benefit from the interruption, often at the expense of the initiator. This suggests that a blanket approach of suspending all interruptions may eliminate the benefit that recipients receive from being interrupted, and that an intelligent filtering approach, such as done by human assistants, would be useful. Bellotti defines the components of context, design guidelines and human-salient details for realizing them [2]. Hudson poses the challenge of making interruptions more effective, since many view interruptions as a valuable part of getting work done [10].

STUDY DETAILS

In order to increase uniformity for this first experiment, we chose four subjects who are similar in terms of working environment and the types of tasks they perform. Each serves in a high level staff position in our university, and has significant responsibilities for day to day administration of a large university department and/or graduate program. Each subject has a private office with a closable door. Their jobs require them to interact with quite a few different people during the day and generally they do not have full control over their own time. They typically respond to a significant number of "walk in" requests, and overall are frequently interrupted. Each of these subjects almost always works with their door open, making them accessible to others most of the time. (Overall their doors were closed at some point in the five minutes before a sample only 3% of the time, and closed the entire five minutes only 0.3% of the time. Note that this mostly eliminates one of the explicit cues people use to indicate non-interruptibility, and hence likely makes predictions for these subjects more difficult.)

For each subject we placed a PC with a large disk and an A/V capture and compression card connected to a small camera and a microphone in their office. Each machine also had speakers for producing audio prompts, and a keyboard which allowed the user to temporarily disable recording if they felt their conversations were too sensitive to be recorded (to ensure privacy, subjects could also retroactively request the recordings from any time period be destroyed prior to viewing). The PC did not include a visual display.

As illustrated in Figure 1, cameras with wide angle lenses were carefully positioned (using a portable mounting pole) so that the primary work area(s) as well as the door were visible. Data was captured in grayscale with a resolution of 320x240 pixels at about 6 frames per second, and 8-bit audio was recorded at 11 Khz. Recording was performed from 7am to 6pm on workdays for 14-22 days for each subject. We estimate that 300 hours (27 days) of compressed recording could be placed on the 80 Gb disks we used. This recording setup worked well except in one case where a week's data was lost due to an undetected improper compression setting that caused the disk to fill up prematurely. For this subject we collected an additional 10 days of data at a later date. Overall we recorded 602 hours from the subject's offices.

Subjects were given an audio prompt to provide a self-report of interruptiblity at random but controlled intervals, averaging two prompts per hour. In order to try to minimize the disturbance caused by our prompts, we chose to ask only one question, and used a five point scale so that the subject could respond in a minimally disruptive way – holding up some number of fingers on one hand (although almost all responses ended up being verbal). Specifically, subjects were asked to "rate your current interruptibility on a scale from one to five, with one being most interruptible." We collected data for a total of 672 prompts when the subject was present.

While willingness to be interrupted is clearly dependent not only on the state of the person, but also on the nature of the interruption, we made this study more tractable by choosing to only look at the state of the interruptee. We presume that in use of our models, an assessment will also be made of the importance of any given interruption, and that this will be balanced against the interruptibility estimate as well as factors such as the recent and total frequency of interruptions and/or specific strategies for minimizing the impact of interruptions [11].

To facilitate processing of the recorded audio and video, we constructed specialized software for playback and coding of the data. Both the overall operation of the software and the items coded for were iterated based on coding and analysis of the first subject (which was subsequently completely re-coded using the final procedures). Multiple coders – students hired on an hourly basis – were employed, and began their work being trained for consistency with the other coders before performing coding that was retained. For cross-validation between coders we performed duplicate coding for a randomly selected 5% of the data and found 93.4% agreement at the granularity of 15 second intervals. To minimize coding time, we have initially only coded the five minutes prior to each sample point for a total of 56 hours of coded data.

The final coder's interface, shown in Figure 1, presented recordings in 15 second sequences. A series of buttons (all with keyboard shortcuts) were provided to indicate the occurrence of events within that segment. At the coders' option, a sequence could be played at normal or double speed. After each segment, the coder pressed a key to go to the next segment, or could back up and see the segment again.

Overall, we found this setup a very good compromise which allowed coders to operate at speeds near or even better than real-time in the most common cases where very little activity was apparent, but also allowed them to control pacing so that they did not fall behind or lose data when complex actions were occurring. To speed up processing, multiple passes over the recordings were made, starting with whether the occupant was present during each prompt. This information was then used to optimize subsequent passes. For example, after coding how many people were present, no sequences of an empty room were shown, and passes which coded information about the activities of guests automatically skipped all sequences when only the occupant was present. With these optimizations, we are now able to code data at a rate of between three and four minutes of coding time per minute of processed recording.

THE DATA

In 54 of the 672 samples (8.0%) the subject was present but did not respond to the self-report prompt. We examined these cases individually and determined that in the vast majority of them, the subject was either on the phone or engaged in conversation with a guest. Based on empirical results from the literature, we expected these activities to be highly correlated with non-interruptibility (and this is borne out in our data). Further, in testing we found that removing these samples from the data had very little effect on the accuracy of the final predictions. As a result, to make analysis and model building simpler we placed these samples in the "least interruptible" category for purposes of model building.

	Most				Least
Subj 1	6.6%	10.2%	29.2%	13.1%	40.9%
Subj 2	10.2%	12.7%	34.9%	16.3%	25.9%
Subj 3	31.5%	15.8%	12.1%	6.1%	34.5%
Subj 4	6.9%	12.3%	22.1%	29.9%	28.9%
All	13.7%	12.8%	24.3%	17.3%	32.0%

Table 1. Distribution of Self-Reports.

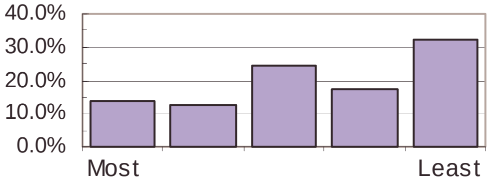


Figure 2. Overall Distribution of Interruptibility Self-Reports

The overall distribution of self-report responses is shown in Table 1 with the aggregate distribution illustrated in Figure 2. Although there are clear differences between the subjects, we can see that a substantial portion of the reports (32.0%) indicated the least-interruptible condition.

In coding from recorded data we logged the following 23 events or situations to act as simulated sensors:

Occupant related:

- Occupant presence.
- Speaking, writing, sitting, standing, or on the phone.
- Touch of, or interaction with: desk (primary work surface), table (large flat surface, not the primary work surface), file cabinet, food, drink, keyboard, mouse, (gaze at) monitor, and papers (any manipulation, including books, newspapers, papers)

Guest related:

- Number of guests present.
- For each guest: sitting, standing, talking or touching (any physical contact or very close proximity with occupant, including handing occupant an object).

Environment:

- Door open or closed.
- Day of the week, and time of day (hour only).

These simulated sensors (which we will also refer to as *features*) were chosen because we a priori believed they might be correlated with interruptibility, a sensor could plausibly be built to detect them, and they could be readily observed in our recordings. (We believe that information about what is happening on the computer screen, such as what application(s) are running, could also be useful, but we could not directly observe that in our recordings.)

Based on the directly recorded information, we also derived a number of variant sensors that captured recency and density effects. For each binary feature – one which either occurred or did not occur in each 15 second recorded segment – we produced variants of the following forms (the names in parenthesis will be used to refer to them later):

- Event occurred in the 15 second interval immediately around the self-report sample (***Imm***).
- Event occurred in every 15 second interval for 1 minute prior to the sample (***All-1***).
- Event occurred in at least one interval in the 1 minute prior to the sample (***Any-1***).
- Event occurred in every interval in the five minutes prior to the sample (***All-5***).
- Event occurred in at least one interval in the 5 minutes prior to the sample (***Any-5***).
- The number of intervals in which the event occurred in the five minutes prior to the sample (***Count-5***).

For guests present, sitting, standing, talking, or touching, we also counted:

- The number of such guests at the sampling point (***Imm***).
- The number of unique guests acting in this way at any time during the prior one and five minutes (***Any-1*** & ***Any-5***).
- The number of guests acting in this way during the entire one or five minute prior period (***All-1*** & ***All-5***).

Overall, we obtained observation values corresponding to a set of 128 direct or derived simulated sensors. Of these, 30 were occurrence counts and 98 were based on binary events. Of the binary event sensors, 8 never occurred in our data, 14 occurred with fewer than 1% of samples (fewer than 7 times), and 20 occurred with fewer than 2% of samples (fewer than 14 times). None of the occurrence count sensors had all counted values occurring less 2% of the time.

PREDICTIVE POWER OF INDIVIDUAL SENSORS

Based on the literature in this area, we would expect that the strongest indicators of non-interruptibility would be those related to social and task engagement (see for example [14]). In particular, interruptions are undesirable when someone is speaking (which can be seen informally by noting that it is almost always rude to directly interrupt someone who is talking). Further, speaking on the telephone is particularly unfavorable for interruption. This may be because the subtle negotiation of an interruption that often occurs in person via eye contact, or other non-verbal cues, cannot include the remote party to the conversation.

While we knew in advance the general type of activities that needed to be detected to produce a good prediction, it was still unclear exactly which specific sensors would be most useful (and which easy to build sensors might work just as well as more complex ones). To try to understand this, we analyzed the predictive power of individual simulated sensors using an *information gain* metric [12].

1	Talk (Imm)	11	N Guests Talk (Imm)
2	Talk (Any-1)	12	Stand (Any-1)
3	Telephone (Imm)	13	Keyboard (Imm)
4	Talk (Count-5)	14	Talk (Any-5)
5	Telephone (All-1)	15	Monitor (All-1)
6	Talk (All-1)	16	Monitor (Imm)
7	Telephone (Count-5)	17	Monitor (Count-5)
8	Sit (All-1)	18	Sit (Imm)
9	Telephone (Any-1)	19	Num Guests (All-1)
10	Stand (Imm)	20	Keyboard (Any-1)

Table 2. Features with Top 20 Information Gain Scores

In simple terms, an information gain metric works by sorting a set of observations with respect to the values of a single feature within the observation. This effectively removes the entropy associated with variations in that feature. The entropy of the resulting ordered data set is then estimated. The entropy estimates from different sortings can then be compared to determine the relative amount of entropy removed, hence the relative information content of each feature. Note that the absolute value of an information gain metric is not of particular interest, only the relative values between features. Also, information gain indicates potential usefulness in prediction, but does not show directly whether the feature indicates interruptibility or non-interruptibility.

It is also important to note that many of our simulated sensors are, by design, inherently overlapping. For example, Talk (Any-1) will always be true when Talk (Imm) is true. In addition, there are some less obvious overlaps such as the fact that the Telephone (Any-*) sensors will almost always imply the corresponding Talk (Any-*) sensors, and that Guest Talk (Any-*) will be quite correlated with Talk (Any-*), since long monologs by either the occupant or guest would be expected to be fairly rare. Information gain statistics allow us to consider multiple overlapping sensors and provide a way to estimate which will be most predictive if we need to choose between them.

Table 2 presents the information gain ordering for the top 20 features. Here we can see that the occupant talk and telephone sensors clearly rise to the top in predictive power – holding eight of the top nine positions.

After talk and telephone, occupant movement sensors Sit (All-1) and Stand (Imm) show the next highest predictive power. These are followed a little further down by Sit (Imm) and Stand (Any-1). Taken together, the sit and stand sensors might be interpreted as being positive and negative indicators, respectively, of engagement with office tasks that are typically done in a seated position.

The next highest indicator is the number of guests talking (Imm), which clearly indicates social engagement. This is followed by several indicators of computer use – Keyboard (Imm) and Monitor (All-1, Imm & Count-5).

In addition to the power of particular sensors, we can also see that the shorter term binary sensors (Imm, Any-1, and All-1) generally tend to be more predictive than the binary sensors working over a five minute period. However, the five minute density sensors (Count-5) seem to have roughly the same power as the short term sensors. (We had initially eliminated the Count-1 sensors to make the sensor set smaller. Based on this result, future work will include reintroduction of the Count-1 sensors and the analysis of their effects.)

Most of the sensors having very low information gain scores occurred too infrequently to provide any predictive power. The only potentially surprising sensor among the bottom 30 scores is the Desk (Imm) sensor at rank 101 out of 128. (In contrast desk sensors for Any-1, Any-5, Count-5, All-1, and All-5 appear at ranks 41, 42, 51, 88, and 99 respectively).

Information gain statistics only consider the predictive power of features in isolation and do not take into account the overlapping nature of our sensors. In the next section we will also consider an approach to analysis of predictive power based on constructing models with more and more sensors and measuring the results of adding sensors on the accuracy of the model.

CONSTRUCTING PREDICTIVE MODELS

In order to explore the question of whether predictive models can be constructed at all, as well as how predictive they might be made, we employed a number of well known machine learning algorithms to construct several different forms of predictive model. To make this work simpler and less ambiguous we first considered the binary decision problem of predicting whether or not the user would indicate "least interruptible". We will call these two states "interruptible" and "not-interruptible." This split was motivated in part by the expected uses of the predictor in avoiding the most harmful interruptions. In addition, anecdotal evidence suggests people often have strong feelings about particular times being "obviously not-interruptible," but often have more ambivalent attitudes towards "partially interruptible" times. This seems to be at least hinted at in the bimodal distribution of self-report values, and would also argue for such a split. After considering this binary problem we also took the most promising modeling approach and explored other variations.

For the binary decision problem we constructed models using decision trees [15], naïve Bayesian predictors, support vector machines [3], and AdaBoost with decision stumps [6]. (We will not attempt to fully describe each of these techniques here. Interested readers can consult the original references above, or a machine learning text such as [12].) All of these models were constructed using widely available, open source software packages (specifically the C4.5 decision tree package [15] and the Weka 3 machine learning software package [16]).

For this data set there is a base accuracy rate of 68.0% (which would be obtained by always indicating "interruptible").

For the model evaluations shown in Table 3, we used a standard cross-validation approach involving multiple trials of model construction. In each trial we randomly selected 90% of the data for training, and used the resulting model to predict the remaining 10%. The numbers reported here are sums from 10 such trials.

Decision trees are perhaps the simplest of the techniques. The decision trees we used are constructed by first selecting the binary test (such as "Num Guests (Imm) > 0") which most usefully splits the data into two parts. Decision trees are then recursively constructed for those subsets. Leaves of the resulting tree are then assigned predicted values. One drawback of decision trees is that, after many subdivisions, each leaf may represent only a small number of samples and hence may be susceptible to noise in the data. As a result, one does not normally build decision trees as deeply as possible, but instead applies certain stopping criteria. In our case we used the C4.5 decision tree package [15] with 10 trials and a minimum branch size of 15.

Table 3a gives the results from our decision tree model. Here, correct predictions appear on the diagonal, and incorrect predictions appear off the diagonal (shaded in gray). Incorrect predictions come in two forms, which we will call "incorrect interruptions" (bottom left) where "interruptible" is incorrectly predicted (and a typical application would improperly interrupt), and "incorrect delays" (top right) where "non-interruptible" is incorrectly predicted (and a typical application would unnecessarily delay delivering information).

We would expect decision trees to work well for this problem because there is a strong and unambiguous feature (talking) that provides a very good initial split. In fact as shown in the rest of Table 3, the 78.1% accuracy provided by decision trees is the best result across the modeling techniques. This prediction is significantly better than chance ($\chi^2(1, 1344) = 17.5, p < .001$).

In addition to decision trees we also tried creating models based on naïve Bayesian predictors, support vector machines, and AdaBoost with decision stumps. The results from these four techniques are presented in Table 3b-d. These results are all similar and there is no statistically significant difference between the largest and smallest. Since results from a variety of unrelated approaches produce very similar results, we feel this clearly shows that predictive models can be constructed, and we are quite hopeful that robust models with results in the 75-80% accuracy range can be driven from real sensors.

MODEL VARIATIONS

Since decision trees represent, in some sense, the simplest of the models and also produced the best results, we used them to explore several additional variations.

All the results reported thus far have been for predictions across all subjects. This is a preferable approach because it offers the hope that general models could be constructed without an extensive individual training period. However, it might be possible to produce better predictions by tailoring models to one specific person. To explore this, we constructed four decision tree models isolated to the data from each individual. While we would expect these models to perform better, in fact they did not in most cases. The resulting accuracy for individualized models for the four subjects was 69.1%, 81.9%, 74.6%, and 76.0%. This lack of improvement is likely due to the effects of having substantially less training data. As a result, it is difficult to draw conclusions about how well personalized models might work with more extensive individual training data.

We next revisited the decision to produce predictions of "least interruptive" vs. something else. (Recall that subjects gave an interruptibility rating from one for most interruptible, to five for least interruptible.) To do this, we considered whether a threshold value of three rather than four might produce better results. However, this instead reduced accuracy to 69.6% (with a base accuracy for this decision problem of 50.7%).

Finally, we looked at whether better results might be obtained via a five-way decision problem rather than a binary decision – in other words, whether there was an advantage to attempting to directly predict the one-to-five interruptibility value. For a multi-way decision problem we were able to use a more sophisticated technique: decision trees with error correction codes [4]. Table 4 presents the results of this model. Since this multi-way problem is substantially harder than the binary problem, overall accuracy (the sum of the main diagonal) is substantially lower than for the binary problems.

	Predict Inter.	Predict Not
Actually Inter.	60.6% (407)	7.4% (50)
Actually Not	14.4% (97)	17.6% (118)
	Accuracy: 78.1%	

a) Decision Trees

Predict Inter.	Predict Not
54.4% (366)	13.5% (91)
11.5% (77)	20.5% (138)
Accuracy: 75.0%	

b) Naïve Bayesian

Predict Inter.	Predict Not
60.6% (407)	7.4% (50)
14.7% (99)	17.3% (116)
Accuracy: 77.8%	

c) Support Vector

Predict Inter.	Predict Not
61.5% (413)	6.5% (44)
16.5% (111)	15.5% (104)
Accuracy: 76.9%	

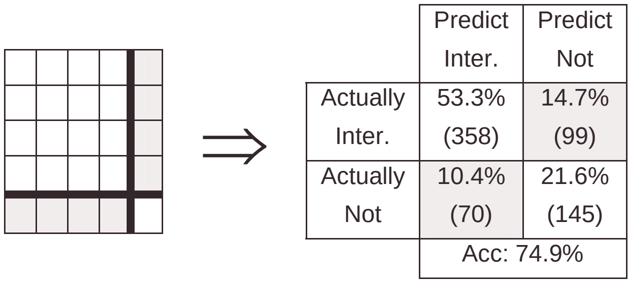
d) AdaBoost Stumps

Table 3. Results from Various Model Types

	Predict 1	Predict 2	Predict 3	Predict 4	Predict 5
Actual 1	5.21% (35)	2.38% (16)	2.83% (19)	1.19% (8)	2.08% (14)
Actual 2	2.23% (15)	1.34% (9)	5.36% (36)	1.49% (10)	2.38% (16)
Actual 3	1.49% (10)	1.79% (12)	11.76% (79)	3.72% (25)	5.51% (37)
Actual 4	1.49% (10)	1.19% (8)	4.61% (31)	5.21% (35)	4.76% (32)
Actual 5	1.93% (13)	1.04% (7)	4.76% (32)	2.68% (18)	21.58% (145)
	Accuracy: 45.1%; within 1: 72.6%				

Table 4. Results for Decision Tree with Error Correction Codes

However, we can compare this model more directly if its results are mapped onto the same decision problem as the previous predictors. This is done by considering all 1-4 predictions to match any 1-4 actuals (i.e., a threshold of 4), as illustrated in Figure 3. In this case, the overall accuracy is 74.9%. While this is not an improvement, it is important to note that this model has two potential advantages. First, this model allows the decision problem to be changed by the user at run-time without changing the model. In particular, the user may set how conservative they would like the system to be in choosing to interrupt by selecting a threshold value between 0 and 5 (e.g., with 0 meaning never interrupt, 3 meaning interrupt when predicted ≤ 3, and 5 meaning always interrupt). The second major advantage concerns the distribution of incorrect results. In this model, the percentage of incorrect interruptions as defined in the binary problem is only 10.4% of the total predictions. If we assume that incorrect delays are preferable to incorrect interruptions in our final application, this could be a substantial advantage and worth the loss of 3% overall accuracy. Figure 4 illustrates the tradeoffs between errors of each type (and resulting overall accuracy) at each possible threshold setting. Note that while it is possible to reduce inappropriate interruptions to as low as 2% of total predictions without stopping all interruptions, this causes overall accuracy to drop to 40% which may be unacceptably low.



	Predict Inter.	Predict Not
Actually Inter.	53.3% (358)	14.7% (99)
Actually Not	10.4% (70)	21.6% (145)
	Acc: 74.9%	

Figure 3. Mapping Multi-Way to Binary Decision Using the "Interrupt ≤ 4" Decision Rule

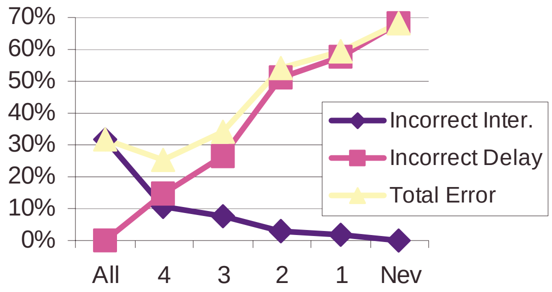


Figure 4. Threshold vs. Error Distribution Tradeoff

PREDICTIVE POWER AND SELECTING REAL SENSORS

In addition to the information gain metric described above, we can also examine predictive power of sensors by looking at the effect of sensors on the accuracy of the models themselves. To do this, we constructed a series of decision tree models constrained by the number of different sensors they could employ and measured the accuracy of each model.

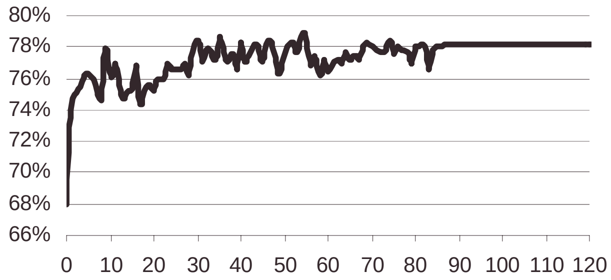


Figure 5. Accuracy (%) as Features are Added to Decision Tree Models

Figure 5 presents a graph showing this effect. The important thing to note here is that the first few sensors (most notably the first one) have a very large impact which accounts for most of the prediction, then the remaining sensors provide mixed results, eventually adding only a few percentage points to the overall accuracy. Note that the sensors added here are chosen by the decision tree algorithm. This is done on the basis of an information gain metric. However, the analysis performed is more sophisticated than the independent information gain scores presented earlier, in that it accounts for the overlapping effects of previously chosen sensors. Each point in this graph represents the average of several decision trees, and so does not necessarily represent a specific sensor being added. However, some of the early sensors added are: Telephone (Imm), Talk (Imm), Num Guests (Imm), Sit (Imm), Write (Any-5), Table (Count-5), and Keyboard (Count-5).

This indicates that a relatively small number of sensors can be used to attain most of the predictive result. In order to further test this, we constructed a final model using sensors chosen based on their ease of implementation. These included a new combined "anyone talking sensor" (since it is easier to not to have to segregate guests from the occupant), as well as telephone (*), keyboard (*), mouse (*), and time of day (*). As indicated in Table 5 the accuracy of this model falls within the range of results found in main models in Table 3. Thus we are lead to believe that robust results should be attainable from practical, relatively easy to implement sensors.

	Predict Inter.	Predict Not
Actually Inter.	58.9% (396)	9.1% (61)
Actually Not	14.4% (97)	17.6% (118)
	Accuracy: 76.5%	

Table 5. Results for "Easiest Sensors" Model

CONCLUSIONS

While the study described here only considers a particular category of office worker, and we cannot yet tell how well the results might translate to other demographics, its results are still quite promising. We have demonstrated that sensor-based estimators of human interruptibility are possible, that robust sensors operating in the 75-80% accuracy range might be constructed using several different types of models, that speech detectors are the most promising sensor for this problem, and that overall a relatively simple set of sensors can probably be employed to achieve good results.

FUTURE WORK

There are many areas for future work in this line of research. First it will be important to expand the study done here to different demographic groups (i.e., different job functions and different work settings, perhaps even to different cultures) in order to understand how robust the results might be across the population. We would also like to compare the predictions made by our model with the performance of humans estimating interruptibility. There are also many additional opportunities for analysis of this data. For example, the analysis done thus far has concentrated almost exclusively on questions relating to construction of predictive models. There is another set of interesting questions related to understanding human behavior that we have only partially touched on here. We would also like to do an in depth review of the misclassifications made by our models to see if there are discernible patterns which could be used to improve the models, and to systematically explore the effects of sensor errors on predictions. Finally, based on these promising results in a Wizard of Oz setting, we hope to be able to construct working systems with real sensors, and create new interactive applications that use them.

ACKNOWLEDGMENTS

This work was funded in part by DARPA and by the National Science Foundation under grants IIS-01215603, IIS-0205219, IIS-9980013 and the second author's NSF Graduate Research Fellowship. We would like to acknowledge the employees that recorded simulated sensor values: Ben Davies, Rick Ebert, Rucha Humnabadkar, Becky Kaplan, Matt Mowczko, and Long Pan.

REFERENCES

1. Adams, M.J., Tenney, Y.J., and Pew, R.W. (1995) "Situation Awareness and the Cognitive Management of Complex Systems." *Human Factors*, 37(1), pp. 85-104.
2. Bellotti, V., and Edwards, K. (2001) "Intelligibility and Accountability: Human Considerations in Context-Aware Systems." Journal of Human-Computer Interaction 16, pp. 193-212.
3. Burges, C.J.C. (1998) "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery*, 2(2), pp. 121-167.
4. Dietterich, T.G., and Bakiri, G. (1995) "Solving Multiclass Learning via Error-Correcting Output Codes", *Journal of Artificial Intelligence Research*, 2, pp. 263-286.
5. Feldman-Barrett, L., and Barrett, D.J. (2001). "Computerized experience-sampling: How technology facilitates the study of conscious experience", *Social Science Computer Review*, *19*, pp. 175-185.
6. Freund Y. and Schapire, R. (1997) A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences, 55(1), pp. 119-139.
7. Gillie, T. and Broadbent, D. (1989) "What makes interruptions disruptive? A study of length, similarity, and complexity." Psychological Research (1989)50: pp. 243-250.
8. Hess, S.M., and Detweiler, M. (1994). "Training to Reduce the Disruptive Effects of Interruptions." Proceedings of the HFES 38th Annual Meeting, v2, pp.1173-1177.
9. Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. (1998) "The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users." *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence.*
10. Hudson, J.M., Christensen, J., Kellogg, W.A., and Erikson, T. (2002) ' "I'd Be Overwhelmed, But It's Just One More Thing to Do:" Availability and Interruption in Research Management.' Proceedings of CHI02, pp. 97-104.
11. McFarlane, D.C. (1999) "Coordinating the Interruption of People in Human-Computer Interaction." *Proceedings of INTERACT'99*, pp. 295-303.
12. Mitchell, T.M. (1997) "Machine Learning", *McGraw-Hill*.
13. O'Conaill, B., and Frolich, D. (1995) "Timespace in the Workplace: Dealing with Interruptions." *CHI '95 Conference Companion*, pp. 262-263.
14. Seshadri, S. & Shapira, Z. (2001). Managerial allocation of time and effort: The effects of interruptions. Management Science, 47, pp. 647-662.
15. Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning*, Morgan Kaufmann.
16. Witten, I.H., and Frank E. (1999) *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann. (Open source software available from: http://www.cs.waikato.ac.nz/~ml/weka/).
17. Zeignaric, B. (1927) Das Behalten erledigter und unerledigter Handlungern. *Psychologische Forschung* 9, pp. 1-85.