Principles of Lifelong Learning for Predictive User Modeling

Ashish Kapoor and Eric Horvitz

Microsoft Research,
Redmond WA 98052, USA
{akapoor, horvitz}@microsoft.com

Abstract. Predictive user models often require a phase of effortful supervised training where cases are tagged with labels that represent the status of unobservable variables. We formulate and study principles of *lifelong learning* where training is ongoing over a prolonged period. In lifelong learning, decisions about extending a case library are made continuously by balancing the cost of acquiring values of hidden states with the long-term benefits of acquiring new labels. We highlight key principles by extending BusyBody, an application that learns to predict the cost of interrupting a user. We transform the prior BusyBody system into a lifelong learner and then review experiments that highlight the promise of the methods.

1 Introduction

Probabilistic user models have been generated via a process of applying a statistical machine-learning procedure to a library of training cases. The process typically relies on supervised learning to acquire labels for variables that are not directly observed in the collection of activity or sensor data. Supervised training often requires an effortful phase of labeling hidden user states such as a user's current or future intention, affective state, or interruptability.

Some user modeling applications bypass manual supervised learning by performing *in-stream supervision*, where tagging occurs in the course of normal activity. For example, in the mixed-initiative Lookout system for calendaring and scheduling [3], a probabilistic user model is used in real-time to infer a user's intention to perform scheduling, based on the content of email messages at the user's focus of attention. To build a case library, the system watches users working with email and assigns labels of a scheduling intention by noticing if calendaring actions occur within some time horizon of the reading of email at the focus of the user's attention. The Priorities system [6], which uses machine learning to assign incoming email messages a measure of urgency, makes available an in-stream supervision capability. A set of policies, communicated to users, is used to label messages with urgency values and made available for review as draft case libraries. For example, messages that are deleted without being read are labeled as non-urgent.

Unfortunately many applications may not be amenable to in-stream supervision as labels for hidden states are not available. In such cases, the construction of predictive user models depends on either manual training sessions or the use of an *experience-sampling* methodology, where users are periodically asked for feedback that is used to label a situation or state of interest. To date, experience-sampling probes have been guided by random probe policies and such heuristics as seeking labels for states that a system is most uncertain about.

We formulate and study a decision-theoretic approach to guide experience sampling, centering on taking a value-of-information perspective. The methods address the challenge of building user-modeling systems that have the ability to perform lifelong learning, by continuing to use the current user model to make decisions about if and when to probe users for feedback, and considering the long-term value associated with such feedback. Beyond the use of the methods for learning predictive models in an efficient manner, the techniques have value for the ongoing updating of a user model given potential changes in users, tasks, and challenges.

We first introduce the legacy BusyBody system [5] as a motivating example. BusyBody employs experience sampling to learn a user model that provides inferences about the cost of interrupting users. We discuss the core challenges of extending BusyBody with machinery that can guide its experience sampling. After laying out core concepts of lifelong learning, we discuss the specialization of the concepts for an alert mediation application. Finally, we discuss experiments with an implementation.

2 Motivating Application: Context-Sensitive Mediation of Alerts

Fig. 1. BusyBody probe for user feedback, running in a binary modality.

Interest has blossomed in the construction of models that can predict the cost of interrupting computer users. To our knowledge, methods and opportunities with the use of probabilistic models to predict the cost of interrupting users, based on the ongoing sensing of a stream of activity, were first described in [6]. The work explored a cost-benefit analysis to controlling the flow of alerts to users, where the inferred urgency of incoming messages is balanced with the inferred cost of interruption, as computed by a Bayesian model. Several studies in the spirit [6] have explored the learning of predictive models for interruptability based on observations of user activity [4, 1]. Efforts in this realm include methods for seeking training from users in an ongoing manner. The BusyBody system employs experience sampling to construct personalized models for real-time predictions of the expected cost of interruption [5]. When BusyBody is in a training mode, the system intermittently probes users with a pop-up query requesting an assessment of their current or recent interruptability. The initial version of the system probed users at random times, constrained to an overall rate set by users. Figure 1 shows a request by BusyBody for input, used when the system is running in a binary hypothesis modality. In other modalities, the system inquires about finer-grained states of the cost of interruption. BusyBody contains an event infrastructure that logs desktop activities including such activities as typing, mouse movements, windows in focus, recent sequences of applications and window titles, and high-level statistics about the rates of switching among applications and windows. The system also considers several kinds of contextual variables, including the time of day and day of week, the name of the

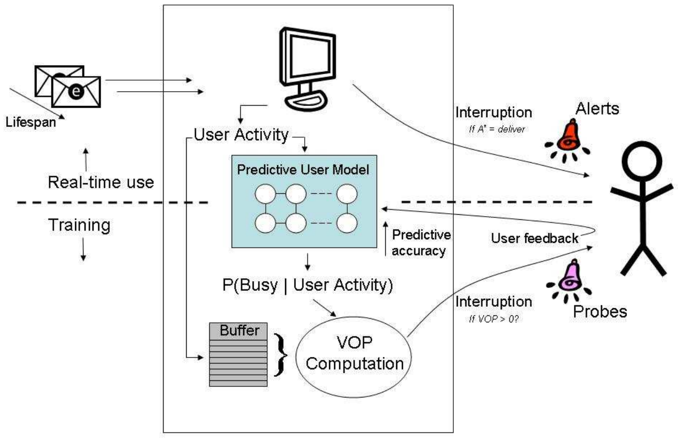


Fig. 2. Lifelong learning framework for training an alert mediation system.

computer being used, the presence and properties meetings drawn from an electronic calendar, and wireless signals. The system employs a conversation-detection system, using a module that detects signals in the human-voice range of the audio spectrum. Responses to probes about interruptability are stored, along with the sensed evidence. Bayesian structure search is employed to build predictive models that are then used in real-time to provide predictions about the cost of interruption from the stream of sensed data.

BusyBody and the models it constructs are typically deployed in larger systems that reason about whether to relay incoming alerts and provides the current cost of interruption to these information triaging systems, which continue to balance the cost of interruption with the inferred urgency of incoming messages [4].

3 Lifelong Learning for User Modeling

We now revisit the experience-sampling challenge in BusyBody to highlight key aspects of a lifelong learning methodology. Assume that BusyBody is used, per its design to continually provide the current cost of interruption within an alert mediation system, based on sensed events and states. The model can become better with additional cases, obtained via experience sampling, where "better" is defined in terms of the performance of the mediation system.

The lifelong-learning challenge is to use the current predictive model within a value-of-information framework to control probes for new cases in an ideal manner, and to incorporate the cost of probing in different contexts into the overall long-term optimization of the use of the system. Figure 2 highlights the lifelong learning framework in a schematic manner. At the core of the framework is the predictive user model that plays a critical role in determining how to handle the incoming alerts. The predictive user model needs to adapt and to learn continuously from the user, and this is done with requests to the user. As shown in the figure, we divide the approach into two interrelated components of analysis: the real-time usage component and the learning component. These components can run simultaneously, each relying on the other.

3.1 Real-time Usage

Over its lifespan, the alert mediation system encounters many incoming messages and the aim is to take appropriate actions when they arrive. Alerting a user about an incoming message that may be urgent comes at the cost of an interruption, which in turn is a function of the user state. Upon receiving a message, the system can either instantly relay it to the user, defer its delivery, or store it for later review, and each of the different actions is associated with a utility. The system aims to maximize the expected value (or equivalently minimize the expected cost) of the handling of messages.

We shall use $U(A, m)$ to refer to the utility of taking a message alerting action A given the arrival of message m. We use $C(A, s)$ to refer to the cost of interruption when the system takes action A given that the user is in state of interruptability s. Upon seeing a message, the optimal action, A^*, is the action associated with maximum expected utility. Assuming decomposability of costs and benefits, A^* is computed as:

$$A^* = \arg\max_A U(A, m) - \int_s C(A, s)p(s|E) \quad (1)$$

We cannot directly observe the user's state of interruptability s. We only have access to the evidence E about the user's context and activity from BusyBody's event system. The user model constructed with available data is used to predict the probability distribution $p(s|E)$ over states of interruptability. The fidelity of the computation of the best action for the system during usage depends upon the accuracy of the user model.

3.2 Training and Probing

Several statistical machine-learning procedures can be employed to construct a user model that computes $p(s|E)$. These methods associate patterns of evidence with states of the user. Candidate learning procedures include Bayesian structure search, support vector machines (SVMs), decision trees etc. As the posterior probability $p(s|E)$ plays a key role in the lifelong learning methodology, we seek to use a probabilistic methodology such as Bayesian structure search or Gaussian Process (GP) classification.

The goal of the training cycle is to learn and to refine the user model by seeking labeled cases from the user. Increasing the number and representativeness of cases may increase the accuracy of the user model on future cases. Unfortunately requesting feedback from the user in experience sampling results in an interruption; hence, a context-dependent cost of probing must be considered. We shall now review the computation of the *value of probing* (VOP) for a label, which is the expected gain in the long-term utility of a system given a probe.

3.3 Computing the Value of Probing

The computation of the value of probing at any moment is based on (1) the available labeled training set, (2) the current set of observations, (3) a characterization of the instances facing the system over time, and (4) a specified period of time of system usage being considered. The latter duration of usage can range from a specific period of time to the expected lifetime of the system.

Let us assume that the system already has n training cases $\mathcal{E}_L = \{E_1, .., E_n\}$, with labels $\mathcal{S}_L = \{s_1, .., s_n\}$. Each E_i denotes evidence capturing desktop activities and context and s_i again denotes the state of the user. Most learning methods focus on minimizing such metrics as classification accuracy. However, a more comprehensive aim is to construct a lifelong learning process that is sensitive to both the predictive accuracy as well as the cost of interrupting the user with probes.

Consider the decision about whether to seek information from users about their state given E_{new}, a vector of observed evidence with relevance to the hidden state. The decision about whether to proceed with a probe is determined according to a maximization of the expected value of information (VOI) [7]. To embark on the computation of the VOP, we first consider a default situation where no triaging system is available to handle incoming messages. In the absence of a mediation system, the user would be alerted by all messages, ($A = A_{deliver}$). The mediation system is introduced to increase the expected utility of messaging to the user. For each message m, the utility of messaging in the absence of the alert mediation system is:

$$V^0(m, s) = U(A_{deliver}, m) - C(A_{deliver}, s) \tag{2}$$

Let A^* be the action selected according to the policy described in (1). Then, for a user state $\hat{s}$ predicted by the current user model, the utility achieved by the system is:

$$V^*(m, \hat{s}) = U(A^*, m) - C(A^*, \hat{s}) \tag{3}$$

The value that the system provides for an incoming message m is the marginal increase in utility over the default situation:

$$V^*(m, \hat{s}) - V^0(m, s) \tag{4}$$

We need to compute the expected gain in utility for future alerts. Assuming stationarity, we approximate this quantity using the mean utility gained over the labeled $\mathcal{E}_L$ and the unlabeled $\mathcal{E}_U$ cases. We note that a user's pattern of activity may not be stationary over time; as time progresses, a user might acquire new behaviors. A system should have the ability to adapt to these potential dynamics. Nonstationarity in users is addressed by using a moving buffer $\mathcal{E}_U$ that summarizes recent user activity and provides a means for modeling the current underlying distribution of a user's behavior. Given the labeled data points $\mathcal{E}_L$ and the buffer of unlabeled data points $\mathcal{E}_U = \{E_{n+1}, .., E_{n+m}\}$ that represents the recent distribution of data points, we can compute the total gain in utility with the use of the system as:

$$J_{all} = \sum_{E_i \in \mathcal{E}_L \cup \mathcal{E}_U} \int_{m_i} \int_{s} (V^*(m_i, \hat{s}) - V^0(m_i, s)) p(s|E_i) p(m_i) \tag{5}$$

Note, that we do not know the state of the user s for all $E_i \in \mathcal{E}_U$; thus, we need to marginalize over s by considering the conditional posterior $p(s|E_i)$. We must rely on our current predictive user model to provide us a good estimate of $p(s|E_i)$. We also need to learn a model of the future stream of messages m_i associated with each situation E_i. Such a model provides the likelihood of different messages, $p(m_i)$, allowing us to marginalize over m_i. We can simply use probability distributions compiled via observation of incoming messages m_i as approximations of future streams of messages. We can alternately model $p(m_i)$ via over time via updating of Beta or Dirichlet distributions. Let us consider the use of a Beta distribution for the case where there are only two kinds of messages, $m = 0$ and $m = 1$. Specifically, if $P(m = 1) = q$, the system models the distribution of future messages as:

$$P(q) = \text{Beta}(\alpha, \beta) = \frac{1}{\mathbf{B}(\alpha, \beta)} q^{\alpha-1}(1-q)^{\beta-1} \tag{6}$$

Here, $q \in [0, 1]$, $\mathbf{B}(\cdot)$ is the Beta function with α and β as parameters. Intuitively, α and β correspond to the number of messages encountered so far where $m = 1$ and $m = 0$ respectively. At the start, we have no information about the proportions of messages, so we have $\alpha = 0$ and $\beta = 0$. Note, that these values of α and β lead $P(q)$ to be a uniform distribution, representing an uninformative prior. As the system encounters more messages, it updates α and β, thus, maintaining an up-to-date belief about the proportions of urgent messages that the system might encounter.

Given the gains in utility computed by considering the labeled points and the unlabeled points, we can compute the *expected value of a system* (EVS) associated with each incoming message as the average gain per message:

$$EVS = \frac{J_{all}}{|\mathcal{E}_L| + |\mathcal{E}_U|} \tag{7}$$

The EVS per incoming message can be converted into an EVS per second, representing the rate at which value is being delivered by the system, given the expected rate of incoming messages.

Following a user response to a probe for a label, we update the predictive user model and may see a gain in the expected value that the system would be delivering per message. However, we must consider the cost of the probe. The difference in the gain and the cost guides the selection of cases to label. Let C^{probe}_{new} be the cost that will be incurred when the user is interrupted by a probe. For simplicity, we shall assume that the cost of interruption for the probe, like the cost of interruption for incoming messages, only depends upon the user state.

We introduce an optimization horizon, k that defines the duration of system usage considered in the learning optimization. k refers to the number of future alerts that will be handled. This value is selected according to the time frame that the user wishes to optimize over. For example, a user may wish to have the system probe so as to optimize the value of the system over two weeks. k determines the tradeoff between the acute cost of a probe and the long-term benefits associated with the expected improvements of system performance by refining the model using the additional case. A large k will tend to push the system to probe the user a great deal early on, while a small k would make the system reluctant to ask for supervision. Formally, we define the value of probing (VOP_k) for the new point E_{new} as the gain in the total expected value that the system is expected to deliver for the k alerts subtracted by the cost of probing:

$$VOP_k(E_{new}) = k \cdot (EVS_{new} - EVS) - C^{probe}_{new} \tag{8}$$

Here, EVS_{new} denotes the total expected value of the system delivered per alert should a label for E_{new} be acquired from the user. The VOP_k quantifies the gain in utility that can be obtained by interrupting the user. Thus, our strategy is to probe the user when $VOP_k \geq 0$. This approach differs from the earlier methods in active learning where the focus has been to minimize the classification error. Note, that this formulation of VOP_k assumes stationarity in the distribution of cases and associated patterns of evidences.

We need to compute VOP_k *before* we know the label for E_{new}. Note that J^{new}_{all} and EVS_{new} cannot be computed before we know the actual label s_{new}. Similarly, C^{probe}_{new} cannot be computed as the costs of labels are different for different classes. Thus, we must approximate J^{new}_{all} with an expectation of the empirical gain:

$$J^{new}_{all} \approx \int_s J^{new,s}_{all} p(s|E_{new}) \tag{9}$$

Here $J_{all}^{new,s}$ is the gain in utility when E_{new} is considered labeled as s and to calculate $J_{all}^{new,s}$, we retrain the predictive model by considering E_{new} labeled as s in the training set. Similarly, we can use the expectation of C_{new}^{probe} as the costs of labeling vary with the user state. Thus, given the VOP_k for the new point E_{new}, our strategy is to interrupt the user if $VOP_k \geq 0$. This strategy ensures that the system learns continuously while working to minimize interruptions to the user.

4 Implementation and Experiments

We now describe experiments with a sample instantiation of the lifelong learning methodology for an alert mediation system. Let us assume that there are two kinds of incoming messages: urgent ($m = 1$) and non-urgent ($m = 0$). Next, we assume that there are two kinds of actions the system can take: either deliver the message ($A = 1$) or postpone the delivery ($A = 0$). We shall consider the utility of outcomes in terms of the cost of delayed review of messages [6]. For simplicity, we shall assume that a fixed cost C_u is incurred if an urgent message is not delivered immediately and that this cost is greater than the cost of deferring delivery of a non-urgent message, $C_{\neg u}$. Note, this requires that we know if the message received by the system is urgent or not. Prior work has applied machine learning to infer the urgency of the messages [6]. We are interested in building a predictive user model that detects whether the user is busy or not, and have $s \in \{1, 2\}$, where $s = 1$ ($s = 2$) correspond to the state that the user is busy (not busy).

Next, we define the cost of interruption $C(A, s)$ by taking an action A. When we hold back ($A = 0$), there is no interruption so ($C(A = 0, m) = 0$). However, the cost of interruption is different when we relay the message to the user in different states:

$$C(A = 1, s) = \begin{bmatrix} C_b & \text{if the user is busy} \\ C_{\neg b} & \text{if the user is not busy} \end{bmatrix} \tag{10}$$

In cases where $C_u \geq C_b \geq C_{\neg u} \geq C_{\neg b}$, the optimal policy is to withhold delivery of the alert if the user is busy, unless the alert is urgent. We shall assume this policy.

We shall use a binary classifier as the predictive user model to detect the state of busy ($s = 1$) and not busy ($s = 2$). We use the GP classification to generate the probability distribution, $p(s|E)$. Details of the GP classification and its implementation can be found in [7] and [12].

If an incoming message is non-urgent and the system correctly detects that the user is busy, then per the policy described above, the message will not be sent to the user and the user will incur the cost of delayed review of non-urgent information ($C_{\neg u}$). However, in absence of the alert mediation system, the non-urgent message would be sent to the user who would incur the cost of interruption should they be busy (C_b). Thus, the net gain of the system is $G_{11}^{\neg u} = C_b - C_{\neg u}$. Here, $G_{ij}^{\neg u}$ denote the reduction in cost when classifying the user state belonging to class i as j while handling a non-urgent message. Similarly, consider the scenario when a non-urgent message is received and the system misclassifies the user state as busy. The system will not deliver the message immediately; consequently, we have $G_{21}^{\neg u} = C_{\neg b} - C_{\neg u}$. Note that the cost of interruption when the user is not busy is low; thus, $C_{\neg b} \leq C_{\neg u}$ suggesting that $G_{21}^{\neg u} \leq 0$. Further, the system relays all messages when the user is not busy and relays all the urgent messages regardless of the user state; consequently, there is no net gain in utilities for the rest of the cases. Note, that the system provides gain in utilities only via suppressing the delivery of non-urgent messages. The system maintains the Beta distribution over the set of urgent and non-urgent messages. Thus, Equation 5 reduces to:

$$J_{all} = \frac{\beta}{\alpha+\beta} \cdot [\sum_{i \in L_1} G_{11}^{\neg u} p_i + G_{21}^{\neg u}(1-p_i)] \tag{11}$$

Here $p_i = p(s_i = 1|E_i)$, the probability that the user is busy, given the evidence E_i and L_1 is the indices of points labeled by the current predictive user model as class 1 (busy). The term $\frac{\beta}{\alpha+\beta}$ appears in the equation as gains only occur for the non-urgent alerts; consequently, the term enables us to consider the likelihood of receiving a non-urgent alert while computing the total gain J_{all}.

The lifelong learning policy guides the BusyBody probe for assessments. Let us consider the cost C_{new}^{probe} incurred when the user is interrupted to label the current instance E_{new}. We assume that the cost of probing depends upon the user state, that is:

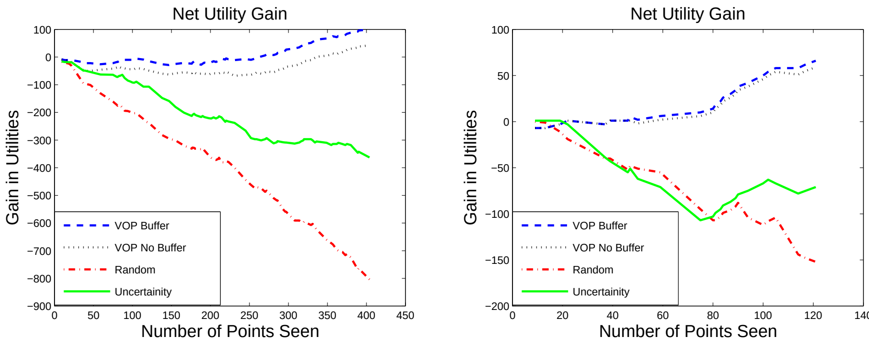
$$C_{new}^{probe} = \begin{bmatrix} C_b^{probe} & \text{if the user is busy} \\ C_{\neg b}^{probe} & \text{if the user is not busy} \end{bmatrix} \tag{12}$$

We employ the concepts in Section 3.3 to guide requests for labels based on a computation of the value of probing.

We studied the value of the methods with simulations on data collected previously by the BusyBody system for two subjects. The first user is a program manager and the other a developer at our organization. The data for each contains two weeks of desktop activity as well as the busy/not-busy tags collected by the legacy BusyBody system, using a random probe policy. We only consider data points in the sequence for which the label for the user state was available, rather than all labeled and unlabeled cases. Thus, the results described can be considered as providing lower-bounds on performance. We expect the value to be greater in usage settings where the system monitors users continuously and can make decisions about all cases. We performed hold-out cross validation, randomly holding out 20% of the data for testing. For evaluation, the system employs the predictive model trained using the data seen *up to* the point being tested. Thus, we can observe and characterize the performance of the system as it is evolving.

In the experiments, we assigned utilities of different outcomes as follows: $C_u = 16$, $C_{\neg u} = 4$, $C_b = 8$, $C_{\neg b} = 1$, $C_b^{probe} = 8$, $C_{\neg b}^{probe} = 1$. We assumed that all of the incoming alerts are non-urgent, *i.e.*, $\frac{\beta}{\alpha+\beta} = 1.0$. Also, we chose k to be the length of the whole sequence. We employed a GP classifier using a polynomial kernel of degree 2 as the core machine-learning methodology for constructing the predictive model. We compare the lifelong learning scheme, both with and without a case buffer, with two alternate policies. First, we consider the policy of randomly selecting cases with a probability of 0.5 to query the user. The other scheme selects cases on which the predictive user model is most uncertain. Specifically, the system probes for labels if $0.3 \leq p(s_{new}|E_{new}) \leq 0.7$.

Table 1 shows the recognition accuracy on the test points and net gain in utilities over the hold-out set. The net gain in utilities includes the gain associated with the system usage and the cost of interruptions from the probes themselves. The lifelong learning method (VOP) outperformed the heuristic policies in accuracy as well as gain in utilities. We found that the buffer helps to improve the performance of the system as it enables the system to exploit the additional available data in computing the expected gain in utility. The lifelong learning scheme with the use of a buffer resulted in overall



(a) (b)

Fig. 3. The net gain in utilities by the system on the test points as it encounters data instances from the case libraries of the (a) program manager and (b) developer.

accuracies of 67.90% for the program manager and 92.31% for developer. The program manager was queried 6 times and the developer was queried 3 times. We compared the accuracy achieved for the same number of labels with the random probe policy used in the legacy BusyBody system. Drawing the same numbers of cases for each subject randomly led to models with accuracies of 59.26% and 50%, respectively, a significant drop in accuracy for both. Figure 3 shows the gain in utilities over the hold-out set as the system sees progressively more labels. The graph highlights the ability of the lifelong learning methodology to provide an efficient means of learning predictive user models continuously over time.

5 Related Work

Most research on statistical models considers training and usage phases separately. Training data is used to generate predictive models and these models are analyzed. Exceptions include the paradigm of active and online learning where the model is continuously updated as the system collects data from the environment. In active learning, the aim is to probe a human/oracle about the label of the points as they arrive. Numerous heuristics and schemes have been proposed for choosing unlabeled points for tagging. For example, Freund *et al.* [2] propose as a criterion for active learning the disagreement among a committee of classifiers. Tong and Koller [11] propose to choose unlabeled points to query that minimize the version space for SVMs. Within the Gaussian Process framework, the method of choice has been to look at the expected informativeness of unlabeled data points [8, 9]. All of these methods inherently focus on minimizing the misclassification rate. Key aspects of the work presented here build upon our earlier work on *selective supervision* [7], employing decision-theoretic principles.

6 Conclusion

We reviewed principles of lifelong learning where the costs and benefits of acquiring and learning from additional cases are considered over the lifetime of a system. We focused on the use of lifelong learning to guide supervision in experience sampling. The method harnesses the value of information to make decisions about probing users for

Table 1. Performance on the test set. Left: program manager data. Right: developer data.

Strategy	Accuracy	# of Probes	Utility Gain
VOP (Buffer)	67.90%	6	100
VOP (No Buffer)	62.96%	12	42
Most Uncertain	66.67%	88	-371
Random (p = 0.5)	59.26%	169	-812

Strategy	Accuracy	# of Probes	Utility Gain
VOP (Buffer)	92.31%	3	66
VOP (No Buffer)	84.62%	3	59
Most Uncertain	80.77%	24	-79
Random (p = 0.5)	69.23%	36	-200

states that are not available to the system. Concepts were illustrated in the context of the BusyBody system, applied on the challenge of balancing the costs and benefits of alerting users to potentially urgent messages. We reviewed the use of a comprehensive measure of the expected value of a system that incorporates both the cost of acquiring additional cases for learning and the net gains associated with real-world use of refined predictive models. In ongoing work, we are pursuing the use of principles of lifelong learning in multiple applications as well as working to extend the methods. Our current research includes investigating the modeling of non-stationary distributions and methods for caching, forgetting, and reusing cases.

References

1. Fogarty, J., Hudson, S.E., Lai, J.: Examining the Robustness of Sensor-Based Statistical Models of Human Interruptability. Proceedings of CHI (2004).
2. Freund, Y., Seung, H. S., Shamir, E., Tishby, N.: Selective Sampling Using the Query by Committee Algorithm. Machine Learning Volume 28 (1997).
3. Horvitz, E.: Principles of Mixed-Initiative User Interfaces. ACM SIGCHI Conference on Human Factors in Computing Systems (1999).
4. Horvitz, E., Apacible, J.: Learning and Reasoning about Interruption, Intl. Conference on Multimodal Interfaces (2003).
5. Horvitz, E., Apacible, J., Koch, P. BusyBody: Creating and Fielding Personalized Models of the Cost of Interruption. Conference on Computer Supported Cooperative Work (2004).
6. Horvitz, E., Jacobs, A., Hovel, D.: Attention-Sensitive Alerting. Uncertainty in Artificial Intelligence (1999).
7. Kapoor, A., Horvitz, E., Basu S.: Selective Supervision: Guiding Supervised Learning with Decision-Theoretic Active Learning. Intl. Joint Conference on Artificial Intelligence (2007).
8. Lawrence, N., Seeger, M., Herbrich, R.: Fast Sparse Gaussian Process Method: Informative Vector Machines. Neural Information Processing Systems Volume 15 (2002).
9. MacKay, D.: Information-Based Objective Functions for Active Data Selection. Neural Computation Volume 4(4) (1992).
10. Platt, J.: Probabilities for support vector machines. Adv. in Large Margin Classifiers (2000).
11. Tong, S., Koller, D.: Support Vector Machine Active Learning with Applications to Text Classification. Intl. Conference on Machine Learning (2000).
12. Williams, C. K. I., Barber, D.: Bayesian Classification with Gaussian Processes. IEEE Transactions on Pattern Analysis and Machine Intelligence (1998).