Evaluating User Feedback Systems

by
Kevin Menard

A Thesis
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the
Degree of Master of Science
in
Computer Science

By

____

May 2006

APPROVED:

____
Prof. Mark Claypool, Thesis Advisor

____
Prof. David Brown, Thesis Advisor

____
Prof. Gary Pollice, Thesis Reader

____
Prof. Michael Gennert, Head of Department

Abstract

The increasing reliance of people on computers for daily tasks has resulted in a vast number of digital documents. Search engines were once luxury tools for quickly scanning a set of documents but are now quickly becoming the only practical way to navigate through this sea of information. Traditionally, search engine results are based upon a mathematical formula of document relevance to a search phrase. Often, however, what a user deems to be relevant and what a search engine computes as relevant are not the same. User feedback regarding the utility of a search result can be collected in order to refine query results. Additionally, user feedback can be used to identify queries that lack high quality search results. A content author can then further develop existing content or create new content to improve those search results.

The most straightforward way of collecting user feedback is to add a graphical user interface component to the search interface that asks the user how much he or she liked the search result. However, if the feedback mechanism requires the user to provide feedback before he or she can progress further with his or her search, the user may become annoyed and provide incorrect feedback values out of spite. Conversely, if the feedback mechanism does not require the user to provide feedback at all then the overall amount of collected feedback will be diminished as many users will not expend the effort required to give feedback. This research focused on the collection of explicit user feedback in both mandatory (a user must give feedback) and voluntary (a user may give feedback) scenarios. The collected data was used to train a set of decision tree classifiers that provided user satisfaction values as a function of implicit user behavior and a set of search terms. The results of our study indicate that a more accurate classifier can be built from explicit data collected in a voluntary scenario. Given a limited search domain, the classification accuracy can be further improved.

Acknowledgements

As it turns out, writing a thesis is a lot more work than first appearances would indicate. This document, while comprehensive in detailing my work, does not really do the whole process justice. It is impossible to see how this report evolved from its humble beginnings into its final form. For this reason, I would to call special attention to the contributions made by others that may not be immediately evident in reading this text.

I would like to thank Professor David Brown and Professor Mark Claypool for co-advising this project. It was an interesting process as one went on sabbatical and the other became the director of the newly formed Interactive Media and Game Development major. That they were able to allocate the amount of time they did for me was amazing. I would also like to thank Professor Gary Pollice for being a third party reader of this thesis. His insightful comments helped to ensure what I hope is a high level of quality in the thesis writing.

Many thanks must also be extended to the Microsoft Corporation, which funded my first year of research on this project as a Research Assistant. In particular, Steve Fox was a great resource that helped in the experimental design. Additionally, I would like to thank the students and faculty at WPI that helped provide us with valuable data. The members of the Campus Computing Center were very gracious in allowing us to install data collection software in the public access computing laboratories.

Aside from actual contributions to the research and thesis, there are the intangibles. My parents have always been supportive of my career decisions and for that I am grateful. Likewise, six years is a long time to spend at a particular school. I would like to express my deepest gratitude to Mel, who stuck with me through those six years and helped me persevere.

Table of Contents

List of Figures

List of Tables

1 Introduction

Collecting user feedback is an important practice that product developers can employ in order to increase the likelihood of success of their products. Product developers that do not solicit feedback from their users are more likely to provide a sub-optimal experience simply because there is a natural divide between what developers think users want and what users actually want [Kvavik et al. 1994]. Ideally, during system development, a rigorous set of user studies are performed in order to account for user expectations. These user studies are costly in terms of both time and money. To reduce this overhead, many producers continuously request feedback from users over the life of a deployed system. When feedback is to be collected from a running system, the user interface (UI) must be augmented with a user feedback mechanism.

The UI designers can either force a user to give feedback via a mandatory feedback mechanism or they can rely on the goodwill of a user by simply requesting feedback via a voluntary feedback mechanism. A mandatory feedback mechanism limits what actions a user can perform until the user provides a feedback value. A voluntary feedback mechanism, while resident in the UI, does not constrain the actions a user can perform and can be ignored by the user. This thesis investigates the differences between mandatory and voluntary feedback mechanisms and how they affect two major properties of user feedback data: quantity and quality. Here, the quantity attribute is defined as the total number of user feedback responses collected by the feedback mechanism over a given time period. The quality attribute is defined as how accurately a user feedback response represents the user's actual impression of the product.

1.1 Problem Description

User feedback about a system can provide information that allows the designers to fix a plethora of problems, ranging from basic usability to improving the actual content being served. For this thesis, we consider the case of improving World Wide Web search engine results. Web search engine results are typically returned by an algorithm that accepts a user search query as input, which is then used to scan the contents of a database of Web pages. The search engine results are a ranked and returned sorted based upon how well a particular Web page matches the search query and by other attributes of that page, such as the number of incoming and outgoing hypertext links, that give an indication its overall relevance [López-Ortiz 2005]. While current search engine technology can be effective, the reliance on such algorithms allow for malicious sites to "game" the system, exploiting properties of the algorithm to artificially increase the ranking of a page [Dalvi et al. 2004]. If user satisfaction with a search result could be taken into consideration, a search engine could validate its search results in order to improve the overall search session [Hijikata 2004]. For example, if a search engine ranks a particular Web page high in its search result listings, but users indicate they are not very satisfied with the page, then the search engine can adjust its listings to rank the page lower.

The most straightforward way of collecting user satisfaction values for search results is to add a feedback collection feature to the UI. Collecting feedback via such an explicit method can be problematic. The UI designer must choose either a mandatory or a voluntary feedback mechanism, considering the benefits and disadvantages of each approach. The constraints imposed on a user by each method have a different effect on the amount of data collected and the quality of that data. If a user satisfaction value could be determined without explicitly asking the user, however, the best of both approaches could be achieved. Every user of the search engine would implicitly provide a feedback value without being hindered by feedback constructs added to the UI.

Previous work has found correlations between user behaviors collected during a user's interaction with a Web browser and a user's level of satisfaction with the Web page [Claypool et al. 2001a; Claypool et al. 2001b; Cen et al. 2002]. In addition to user behaviors, certain environmental attributes, such as the number of images embedded in a Web page, have shown to be related to user satisfaction [Ivory et al. 2001]. The combination of these user behaviors and environmental attributes are termed *implicit indicators* in the literature. A classifier can be built that predicts user satisfaction as a function of implicit indicators. Search engines could then use the classifier to determine user satisfaction with search results without explicitly asking users for feedback.

The discovery of a set of implicit indicators required the retrieval of explicit feedback values from users in order to determine the appropriate correlations between a particular behavior and a satisfaction level. The user satisfaction values used in the implicit indicators research were collected via a mandatory feedback mechanism. Unfortunately, forcing a user to provide feedback for each search result is not practical; users will quickly become annoyed with the search system and seek out alternative tools [Adamczyk & Bailey 2004]. A search system using a voluntary feedback mechanism may be much more tolerable to users and thus more likely to be used in "real world" settings. However, to the best of our knowledge there has been no work performed that shows a correlation between data collected via a voluntary feedback mechanism and implicit user behaviors that indicate user satisfaction values.

The research question is therefore:

> *Can voluntary data can be used to train a classifier that is as effective as a classifier trained with mandatory data.*

1.2 Hypotheses

Due to the different natures of mandatory and voluntary feedback mechanisms, we expect that they will yield a different amount of feedback as well as a different quality of feedback. Both high quantity and high quality are important properties for constructing accurate classifiers. Quantity is a straightforward attribute to measure, namely the number or feedback values given per user. Quality, on the other hand, is not easily quantifiable and thus difficult to gauge. We relied on the previous work on implicit indicators in order to evaluate the degree of quality of our collected feedback data. High quality data will exhibit little variation between collected feedback values and the corresponding expected feedback values that are calculated using implicit indicators.

The first hypothesis (**H1**) is that a mandatory feedback method will collect higher quantities of data than a voluntary feedback method. If users are not required to give feedback, in most cases they will not, and thus it follows that a mandatory feedback mechanism would yield more data than a voluntary feedback mechanism.

The second hypothesis (**H2**) is that a voluntary feedback method will collect higher quality data than a mandatory method. The hypothesis is based upon typical user response to elements that they deem to be intrusive. For example, if the only way to remove a mandatory feedback mechanism is to click a button that is tied to a particular feedback value, many users will simply click the nearest button in order to make the UI element disappear without regard to the actual feedback given. However, if the user goes out of the way to give feedback, one can assume that the feedback rating is of high quality rather than simply a means to an end.

1.3 Outline of Thesis

We investigated the differences in the quality and quantity of data collected by mandatory and voluntary feedback mechanisms. We collected feedback from users as they performed actions in *controlled* and *uncontrolled* scenarios. In the controlled scenario, users performed Microsoft® Excel tasks and their search domain was limited to the Microsoft Office help system. In the uncontrolled scenario, users were allowed to leisurely search the Web using the Google™ Web search engine. We chose two different scenarios in order to determine to what degree the search domain affects user feedback. By factoring out the search domain, we can focus more on the differences between mandatory and voluntary feedback mechanisms.

We experimented with several different feedback UI implementations in the context of a Web browser and observed how those differences relate to the quality and quantity of the data collected. We enhanced the Microsoft Internet Explorer Web browser with our UI modifications in a system we dubbed the *Mandorvol Browser*[1]. The Mandorvol Browser used a pop-up window for mandatory feedback collection and a passive side panel for voluntary feedback collection. The Mandorvol Browser

[1] The name *Mandorvol* was derived from the two types of feedback mechanisms being studied: MANDatory OR VOLuntary.

Microsoft is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

Google is a trademark of Google Inc.

was programmed to randomly choose from one of four experiment types upon initialization and its behavior adjusted accordingly.

The four experiment types were *mandatory controlled*, *mandatory uncontrolled*, *voluntary controlled*, and *voluntary uncontrolled*. In the controlled experiments, the feedback mechanism was only displayed when the user viewed a search result from a limited search domain (here, the Microsoft Office help system), whereas in the uncontrolled experiments, the feedback mechanism was displayed when the user viewed a search result from an unbounded search domain (here, the Google WWW search engine). The Mandorvol Browser presented the user with a pop-up window for the mandatory set of experiments and the passive side panel for the voluntary set of experiments.

We conducted the experiments on the Worcester Polytechnic Institute campus in its three primary public access computer labs. The experimentation period lasted 38 days and consisted of 161 participants. The collected data was analyzed using traditional statistical methods and was also used to train a classifier that predicted a user satisfaction value for a search result as a function of implicit behavior values. The explicit feedback values collected were used to both train and test the classifier's predictions. As with the work completed by Fox et al. [2005], the classification accuracy of our classifier provided insight into the quality of the collected data. Having thoroughly investigated the issue of data quality, we were able to address the primary research question.

We found that a mandatory feedback mechanism will indeed collect more data than a voluntary feedback mechanism will. In the uncontrolled scenario the mandatory feedback mechanism collected 27% more responses (normalized per user) than the corresponding voluntary feedback mechanism, while in the controlled scenario it collected 32% more. We also found that a voluntary feedback mechanism will collect higher quality data than a mandatory feedback mechanism will. Both of these results support our original hypotheses. Through a detailed data analysis, we found that a classifier used to predict user satisfaction values can be trained with data collected via a voluntary feedback mechanism. Moreover, we observed that such a classifier will perform at least as good as, if not better than, one trained with data collected via a mandatory feedback mechanism. Additionally, we noted a threshold at which the increased amount of data collected by a mandatory feedback mechanism ceases to be a contributing factor to a classifier's accuracy.

The thesis is structured as follows: Chapter 2 provides a survey of previous work performed in relation to user feedback systems; Chapter 3 offers supplemental material about machine learning techniques, focusing on decision trees, which we use in our comprehensive data analysis; Chapter 4 details the design and execution of the two pilot studies performed in preparation of the experiment; Chapter 5 describes the experimental methodology and high-level results; Chapter 6 presents a detailed analysis of the collected data and how it relates to the original problem; and Chapter 7 provides a conclusion with suggestions for future work.

2 Background

This chapter provides background information on several key concepts necessary for complete understanding of the thesis, including classifiers and decision trees.

2.1 Classifiers

It is often the case that system designers embed a data collection component into their software systems. The collected data can be used to generate a variety of statistics that the designers can use to further enhance the system. For example, by tracking the most popular paths through a Web site, the Web site's designers can dedicate resources to improving those paths. While such historical data is useful for reasoning about the current state of the system, the system designers may want to reason about data that have yet to be seen. Continuing the example, the Web site designers may want to be able to *classify* a Web page as either popular or not popular before the page is even published.

A *classification* is a value, drawn from a predefined class, that is assigned to a datum for a particular attribute of interest. In this case, the attribute of interest is Web page popularity. Classification values are drawn from a discrete set of values, allowing the full data set to be partitioned by each datum's value. In the example, the possible attributes values are {`popular`, `not popular`} and the set of Web pages can be partitioned into two disjoint sets by their given classification value. For data that has already been observed, all classification values will be known. For unobserved data, however, the classification values will not be known, even while other attributes have known values. As an example, while the Web page's popularity may not be known, its content and layout are known values.

Classification learning, a subset of machine learning, is the process by which a discrete-valued function – a *classifier* – can be developed that will predict classification values based on historical classifications of currently observed attribute values [Baralis & Chiusano 2004].

Machine learning is a discipline of artificial intelligence that attempts to endow computers with knowledge through exposure to data. The underlying assumption is that data trends do not change significantly and thus what was observed in the past is likely to be similar to what will occur in the future. The knowledge gained in machine learning is experiential, with an optional initial knowledge base provided as a basis for learning. This process contrasts with other techniques, such as expert systems, in which the computer is given all the knowledge it needs to perform its tasks up front [Robinson & Domingos 2003]. The benefit of a machine learning approach is that full knowledge is not required prior to system deployment; the system is capable of learning new concepts as it is exposed to them, making it far more adaptable to changes in its environment.

There are many machine learning algorithms that can be used for classification, each with a unique structure for essentially solving the same class of problems. Artificial neural networks attempt to mimic the basic structures in human brains that store and retrieve information [Lane & Neidinger 1995]. Bayesian networks use localized conditional probability tables in a graph that represents causal relationships between the nodes [Pearl 2000]. Nearest neighbor instance-based learning uses Euclidean distance between encoded data instances for problem solving [Yianilos 1993]. Decision trees use tree structures to represent decision points based upon different attributes of the data [Moret 1982]. Each of these constructs has a common set of base operations, while being vastly different forms of treating data. We chose decision trees for our study since they can encode a human-readable set of rules and because previous work has also used them (Fox et al. 2005), allowing us to further validate our work.

The classification learning process is divided into a training and a testing phase. In the training phase, new data is entered into the system and the system's knowledge base is updated due to any learning that occurs. In the testing phase, the algorithm is asked to answer questions with solutions known only to the experimenter. The algorithm's *classification accuracy* is the percentage of correct responses it produces.

Consequently, the machine learning approach to classifier construction requires splitting the data set into two different subsets for the training and testing stages. The split is necessary to prevent a phenomenon known as *overfitting* in which the classifier is so specialized that it can only correctly answer questions from the data with which it was trained. By partitioning the data set and removing the testing set from the training process, it is possible to adequately test how well the classifier will perform over data it has not yet observed.

Dividing the data set can be a complicated matter. The person constructing the classifier wants as much data as possible to train the classifier but at the same time, have a large enough test set to ensure the correctness of the classifier. Furthermore, the training and testing sets must have the same distribution of data values in order to be representative of the same problem domain. If the distribution of values is different, then the classifier will perform poorly over the testing set.

There are many strategies for performing the data set decomposition. Unfortunately, choosing the best method for a given classifier and data set may rely heavily on the skills of the experimenter. For example, in some contexts it may be appropriate to use the same test set for all experimental runs. This approach is easy, but inflexible as new data is added. Furthermore, it causes the classifier to be susceptible to overfitting to the test set, as the test set is the single, constant point of validation. In other cases, it may be more appropriate to randomly choose the test set, using a stratified random sampling of the full data set. While flexible, the non-deterministic nature of the test set selection may make it difficult for the experimenter to validate results with this approach. The experimenter will have to exercise judgment in selecting a proper technique.

Although much time can be expended on determining how to divide the data set, over the past thirty years of machine learning, some methods have shown to generally work better than others. For the purpose of this thesis, we will only concern ourselves with *n-fold cross-validation*. N-fold cross-validation is a technique for training classifiers without a dedicated test set. The data set is split into *n* disjoint, equally sized bins of equal data distribution and one bin is reserved for testing while the other *n* - 1 are used for training. This process is repeated for each bin and the resulting classification accuracy averaged over the course of the run. The rationale for this method is that all data has the opportunity to be used for training and testing in the different folds. This approach limits variation associated with an “unlucky” data split in which either the training or the testing set are not truly representative of the full data distribution.

There is no guiding principle that leads to an appropriate selection for *n*, but 10 has shown to work very well in practice [Witten & Frank 2000]. In this case, each datum is used for training in nine cases and used for testing in one. For our data analysis, we performed 10 runs of 10-fold cross-validation for each experiment in order to further limit the effects of "unlucky" data splits.

2.2 Decision Trees

A decision tree is a classifier that represents decision points used for classification in a tree structure. In a decision tree, each internal node represents one of a set of attributes on which a decision is to be made. The edges leaving that node represent the conditions for the decision. Leaf nodes in the tree correspond to the classification value.

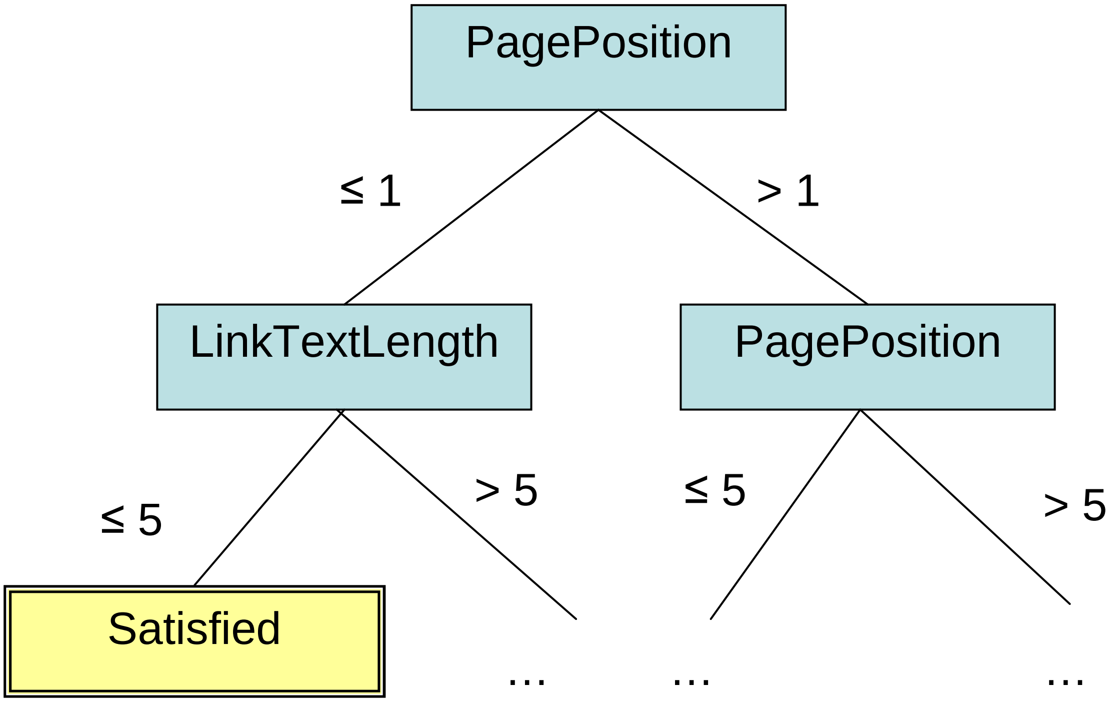


Figure 2-1 Example decision tree shown as a binary tree. The internal nodes (shown in blue, single-bordered boxes) represent decision points while the outgoing edges represent the decision value. The leaf nodes (shown in yellow, double-bordered boxes) represent the classification value.

Figure 2-1 shows an example decision tree fragment that can be used for predicting user satisfaction with a Web page. In this example, the decision nodes are `PagePosition` and `LinkTextLength`, shown in the blue, single-bordered boxes. The one leaf node is the classification value `Satisfied`, shown in the yellow, double-bordered box.

Each datum, such as {`PagePosition` = 1, `LinkTextLength` = 4}, is entered into the tree and progresses in a path from the root to a single leaf. The path the datum takes is determined by its attribute values and the decision points in the tree. Since datum evaluation proceeds from the root of the tree to a leaf node, decision nodes higher in the tree should be able to group more data instances together than nodes lower in the tree; this illustrates how information gain affects the topology of the decision tree. In the above example, the value of the datum's `PagePosition` attribute is considered first. If the value is less than or equal to 1, then the `LinkTextLength` value is evaluated next. If the `PagePosition` value was greater than 1, then the `PagePosition` is considered again for further refinement.

The ultimate goal of a classifier is to be able to produce valid classifications based upon data previously seen. Extending beyond classification accuracy, each classifier also has innate properties that make it fit for a particular class of problems. Decision trees, for example, produce a set of human readable rules that allow the experimenter to easily understand how the classifier is deriving its classifications. Artificial neural networks, on the other hand, must be treated as a closed entity and yield few insights as to their decision making process. The experimenter must determine the goals for a project and choose an algorithm that will address them. For this thesis, we chose to use decision trees precisely because we wanted to have a rich set of generated rules.

Each path through the tree encodes a set of rules that can be used for classification. Since the nodes are attributes of the data and the edges are based upon the domain of their corresponding nodes, the rules are easily understandable by an experimenter familiar with the structure of the data. The partial set of rules representing the tree in Figure 2-1 is:

$$(\mathtt{PagePosition} \le 1) \wedge (\mathtt{LinkTextLength} \le 5) \Rightarrow \mathtt{Satisfied}$$
$$(\mathtt{PagePosition} \le 1) \wedge (\mathtt{LinkTextLength} > 5) \wedge \ldots \Rightarrow \ldots$$
$$(1 < \mathtt{PagePosition} \le 5) \wedge \ldots \Rightarrow \ldots$$
$$(\mathtt{PagePosition} > 5) \wedge \ldots \Rightarrow \ldots$$

Although some of the rules are only partial rules due to the incompleteness of the tree, all of them have the same basic structure. The rules are conjunctions of predicates that entail a classification value. In this example, the predicted value is the user’s level of satisfaction. Despite being a little terse, to an experimenter familiar with the data, generating a natural language representation of these rules is a trivial matter. For example, the first rule states: “If the result page appears as either the first or second link (i.e., the page position is 0 or 1) on a search result page and the HTTP link consists of 5 or fewer characters, then the user will be satisfied with the search result.”

We chose to use decision trees for our data analysis primarily due to the fact that it can generate a set of human readable rules. By having human readable rules, we were able to both validate and reason about our results. For example, early in the experiments we used a time-based attribute in the classifier. Coincidentally, the experimentation times were nearly unique for all subjects. As a result, the time-based attribute could not be used for general rules and was causing an overfitting of the data. By monitoring the decision trees throughout the analysis, we were able to detect this flaw and correct it. Validation of our results was made possible by observing generated rules that were consistent with the previous work on implicit indicators.

2.2.1 Decision Tree Construction

Decision trees are actually a family of classifiers with common attributes. The choice of the decision nodes and edge values can vary substantially between different decision tree construction algorithms. Despite the lack of a definitive decision tree algorithm, the pioneering work of Quinlan on his ID3 and subsequently, C4.5, algorithms has become the de facto standard for how to build a decision tree.

ID3 was Quinlan's first decision tree construction algorithm. Quinlan [1986] proposed the notion of *information gain* for choosing among attributes of the data set for each level in the tree. Information gain is derived from the concept of *entropy* from the field of information theory. Entropy is simply a measure of variation in a given sample and information gain is used reduce the "disorder" of a sample by segmenting it into different subsets, each of which has less variation than the whole sample. Classification accuracy is increased as the entropy at the leaf nodes is decreased, since a lower entropy value means a classification will correctly apply to more instances in that subset. As each attribute in the dataset is considered as a candidate for a decision node, its information gain is calculated as its expected reduction of entropy. The attribute that reduces the total entropy value the most, or provides the maximum information gain, is then chosen for the decision node, since it will partition the data into subsets with the least variation. Once an attribute is used, it cannot be reused in the same path. The selection process continues until either all attributes have been used or all the data in a given subset have the same classification value.

The description of the attribute selection process is a bit of a simplification. In actuality, an attribute can appear more than once in a path, but all such appearances must be sequential (see Figure 2-1 as an example). In this sense, the attribute is being used to make decision using a conjunction of predicates. In fact, the path could be normalized such that the attribute is used only once, so the expressive power is equivalent. An experimenter may choose to use attributes in this manner if they would like the decision tree to have special properties. For example, the experimenter may wish to only generate binary trees in order to take advantage of various algorithms that work well over binary trees.

2.2.2 Decision Tree Pruning

Since new paths are added to the decision tree only as they are needed, the classifier will naturally attempt to create the smallest trees that it can. Despite this preference for small trees, it is still possible for the classifier to generate decision trees that overfit the data. Overfitting in this case refers to paths in the tree that only exist due to "bad" instances in the training set that do not accurately represent the true data set distribution being sampled. A testing set will help identify overfitting during the evaluation of the classifier, but it will not prevent overfitting from occurring.

Addressing overfitting in a decision tree requires the removal of nodes from the tree that are detected as being unnecessary. There are many ways that this can be accomplished, but the most common approach is for the classifier to further split the training set into a validation set and training set. The validation set is used similarly to the testing set, with the exception that it is internal to the training procedure and can thus be used to alter the training process. In *reduced-error pruning* [Quinlan 1987], each subtree rooted at a decision node is iteratively replaced with the child node that matches the most training instances. The resulting tree is evaluated with the validation set and if the classification accuracy is greater than or equal to the classification accuracy of the same tree with the node present, then the decision node is considered unnecessary and pruned away.

2.2.3 ID3 Versus C4.5

ID3 introduced some important concepts to the field of decision trees. It did have several shortcomings however, which Quinlan later addressed [1993] with his C4.5 algorithm. For example, C4.5 introduced a cost component into the attribute selection process so the classifier could balance between attributes that provided the largest information gain and the cost of collecting a value for that attribute. A particularly important improvement made with C4.5, however, was the ability to handle real-valued attributes and data with missing attribute values. The data we collected consisted of both types of values, making ID3 an inappropriate classifier for our analysis.

C4.5 addresses the problem of real-valued attributes by dynamically creating subintervals of the data range for the decision points. An edge in the tree thus represents a membership test for a given value in a particular subinterval. Fayyad presented a method that selected a bisection point, splitting the data into two subintervals, that maximized information gain in [Fayyad 1991]. Fayyad and Irani [1993] extended this work to work for an arbitrary number of subintervals.

The matter of missing data values requires a more complicated solution than the handling of real-valued attributes. C4.5 handles such cases by considering each value that can be assigned to an attribute and associating a probability with it. Assuming homogeneity of the data, the probabilities can be inferred by the distribution of values from other data instances that do not have a missing value. A datum can thus follow multiple paths through a tree, one corresponding to each different probability. The path that has the highest probability, calculated by the probability at each node, is the one that will yield the classification value for that datum.

Another important enhancement made in C4.5 is the *rule post-pruning algorithm* used to prune trees. The reduced-error pruning algorithm of ID3, while effective, has one major drawback. The removal of a decision node causes the tree to change in ways that may be problematic in certain cases. For example, if a decision node has four children, it may be the case that in three of the four cases the node is unnecessary but in the fourth case is needed. Nevertheless, the removal of the subtree rooted at that decision node will affect each of its children equally. Rule post-pruning circumvents this issue by considering each path through the tree individually.

In rule post-pruning, each path through the tree is converted to a rule identical to those shown in section 2.2. Each rule is then considered in isolation and each predicate in the rule is temporarily removed. Using the validation set, the classification accuracies of the rule before and after the change are compared. If the reduced rule performs at least as well as the longer rule, the predicate is permanently removed. In this way, the same decision node can be handled differently for each path through the tree.

For our data analysis, detailed in Chapter 5, we used the C4.5 algorithm. Our data consisted of both real-valued attributes and attributes with missing data instances. C4.5 handled the data appropriately and produced a rich set of rules that we used for further analysis.

3 Pilot Study

Prior to the commencement of the actual experiment, two pilot studies were conducted in order to determine how the final experiment should be performed. The pilot studies were of the type *voluntary controlled* and explored the efficacy of various voluntary feedback mechanisms as well as the end-user tasks to be performed in controlled situations.

3.1 Rationale

The purpose of the pilot studies was to gather information about several experimental designs in order to determine the best set of parameters for the actual experiment. The pilot studies focused on two key parameters: the voluntary feedback mechanism and the set of Excel tasks to be performed during the controlled experiments.

A clear design goal of the voluntary feedback mechanism was that it should be as unobtrusive as possible; i.e., it should not detract from normal computer use. Our first approach to such a feedback mechanism was to embed the feedback form directly in a Web page. We believed that this method would provide the most natural "feel" to a user by tightly coupling the feedback mechanism with the content. Unfortunately, we encountered a large number of technical hurdles, mostly due to security constraints in Microsoft Internet Explorer (IE), while modifying the HTML DOM to insert our feedback mechanism. As a result, we were forced to investigate other ways of implementing the feedback mechanism. We discovered that an explorer band, while not having as tight an integration with the actual Web page content, was an attractive alternative.

An explorer band is a type of side panel positioned either horizontally along the bottom or vertically along the left-hand-side of an IE window. Explorer bands are commonly used for enhancing IE with such features as history viewing and search engine interfacing. Due to their standard use as an IE enhancement, explorer bands seemed to be the natural choice for the Mandorvol Browser.

In addition to explorer bands, some of the experiments were run using pop-up windows that allowed a voluntary response. The pop-ups were used as a baseline of how much feedback we could reasonably expect to collect. Thus, the pilot tests were used to find a balance between the degree of voluntariness and the utility of the feedback collection method (measured in the amount of feedback obtained). A highly voluntary method that collects no feedback is effectively useless, whereas a method that is highly invasive but collects a lot of feedback may have data that is not predictive of user satisfaction. In order to measure both values, the Mandorvol Browser collected user data such as feedback responses given throughout the study and upon study completion the participant was asked, via a short questionnaire, to provide feedback about how invasive the feedback band was.

The pilot studies revolved around a set of user tasks to be performed in Microsoft Excel, i.e., a controlled situation. The intention was to find a set of tasks that were easy to complete but uncommon enough to require a search over Microsoft Office help assets. Additionally, the choice of these tests was guided by a goal of keeping all experiments to 15 minutes in length. The time to complete each study was recorded and associated with the user feedback. The aforementioned questionnaire also had questions about both the user's prior Excel experience and their experience with the class of Excel tasks used in the study.

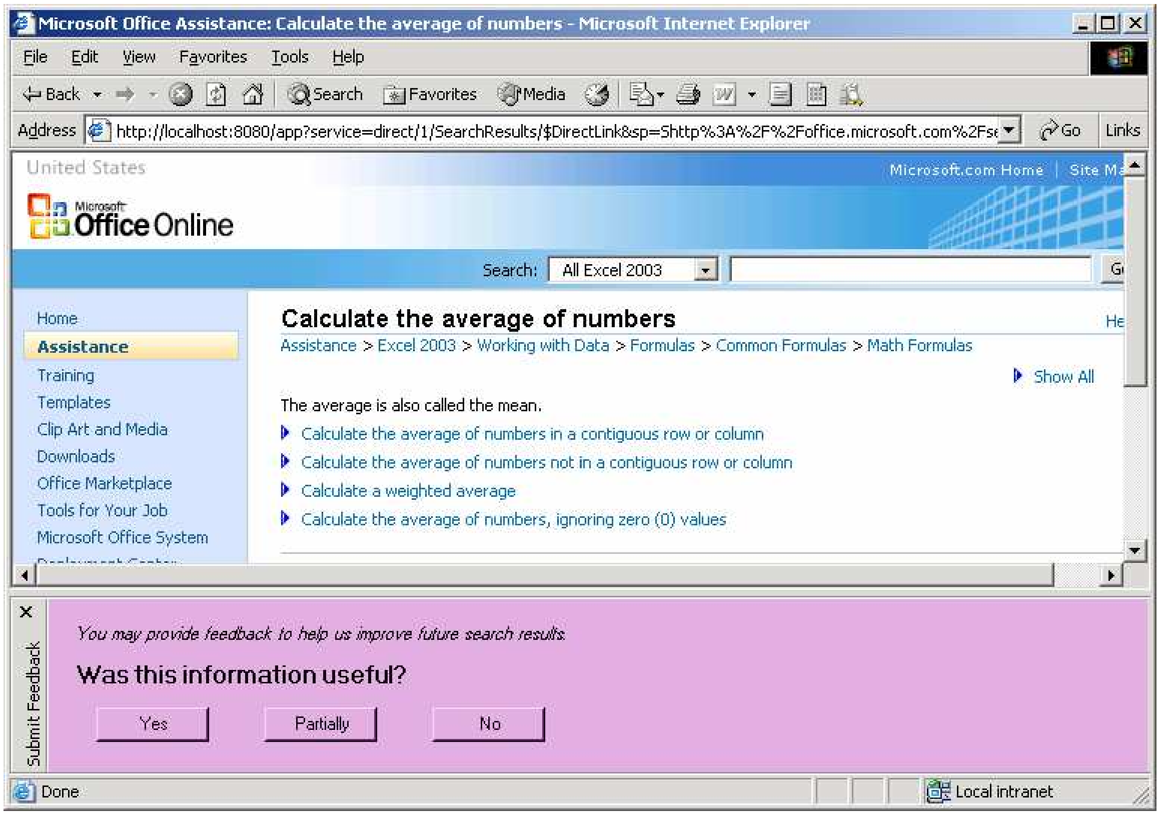
3.2 Methodology

The pilot study runs were all performed on the same PC, one user at a time. The study population consisted primarily of graduate students from the Computer Science department at WPI and was chosen mostly as a matter of convenience. Each participant was able to complete the pilot study with no time pressure and, with the exception of start time, the study environment was consistent from person to person.

Initially, study participants were given written instructions via a Web page about how to complete the study. However, it became clear that participants were not reading the directions completely and in order to remedy the problem the procedure was modified so that the proctor iterated over the directions with the participant prior to the start of the study. Once the study began, the proctor left the participant to provide privacy, but was in the vicinity as to be able to answer any questions the participant may have had.

As mentioned previously, the pilot studies used experiments of the type *voluntary controlled*, meaning that the user was asked to complete a set of Excel tasks but was not required to give any feedback on any of the results from an Office help search. The user was unaware of the purpose of the study and simply worked at completing Excel tasks.

The first pilot study used two different voluntary feedback mechanisms: a horizontal explorer band that spanned the bottom of an IE window and a pop-up window that could be closed without actually rating a search result item. The explorer band was originally colored gray, but after the first participant completely ignored it because he thought it was part of IE, it was changed to a distinctive pink color. A screenshot of the explorer band can be seen in Figure 3-1.



http://localhost:8080/app?service=direct/1/SearchResults/$DirectLink&sp=Shttp%3A%2F%2Foffice.microsoft.com%2Fs

Calculate the average of numbers

Assistance > Excel 2003 > Working with Data > Formulas > Common Formulas > Math Formulas

The average is also called the mean.

- Calculate the average of numbers in a contiguous row or column
- Calculate the average of numbers not in a contiguous row or column
- Calculate a weighted average
- Calculate the average of numbers, ignoring zero (0) values

You may provide feedback to help us improve future search results.

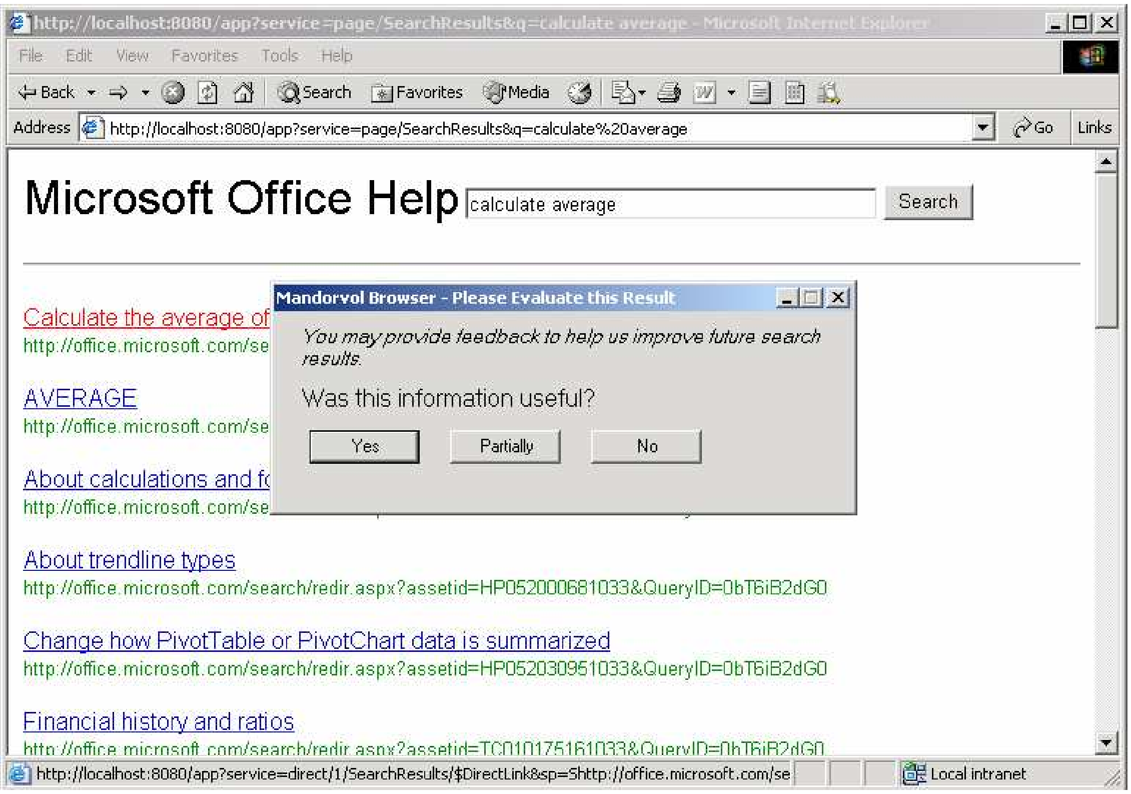
Was this information useful?

Figure 3-1 Horizontal explorer band (first pilot study).

The feedback mechanism used is a modification of the feedback pop-up used by Microsoft in its version of the Curious Browser [Fox et al. 2005]. Although we changed the wording of the prompts and switched from radio buttons to push buttons, the same basic features were retained. The user is made aware that feedback could be given if so desired, then is asked a question with a set of possible (independent) responses. Since this feedback mechanism is voluntary, feedback need not be given in order to make further progress with the Web session; the user can completely ignore the feedback panel or even close it via the "X" button on the left, if so desired.

The Mandorvol Browser detects when the feedback band should be shown and when it should be hidden. Thus, feedback can only be given for search results and can only be given once per search result.

Figure 3-2 shows the voluntary pop-up window used in the first pilot study with the Mandorvol Browser. As can be seen, it is identical in structure to the feedback pane. We did not change the color of the pop-up window from the original gray since there was no motivation to do so. Whereas the feedback pane was vying for the user's attention, the pop-up window had the user's focus by its very nature. Once again, the Mandorvol Browser displayed and hid the pop-up as appropriate.



http://localhost:8080/app?service=page/SearchResults&q=calculate average - Microsoft Internet Explorer

Address http://localhost:8080/app?service=page/SearchResults&q=calculate%20average

Microsoft Office Help calculate average

Calculate the average of
http://office.microsoft.com/se

AVERAGE
http://office.microsoft.com/se

About calculations and fo
http://office.microsoft.com/se

About trendline types
http://office.microsoft.com/search/redir.aspx?assetid=HP052000681033&QueryID=0bT6iB2dG0

Change how PivotTable or PivotChart data is summarized
http://office.microsoft.com/search/redir.aspx?assetid=HP052030951033&QueryID=0bT6iB2dG0

Financial history and ratios
http://office.microsoft.com/search/redir.aspx?assetid=TC010175161033&QueryID=0bT6iB2dG0

Mandorvol Browser - Please Evaluate this Result

You may provide feedback to help us improve future search results.

Was this information useful?

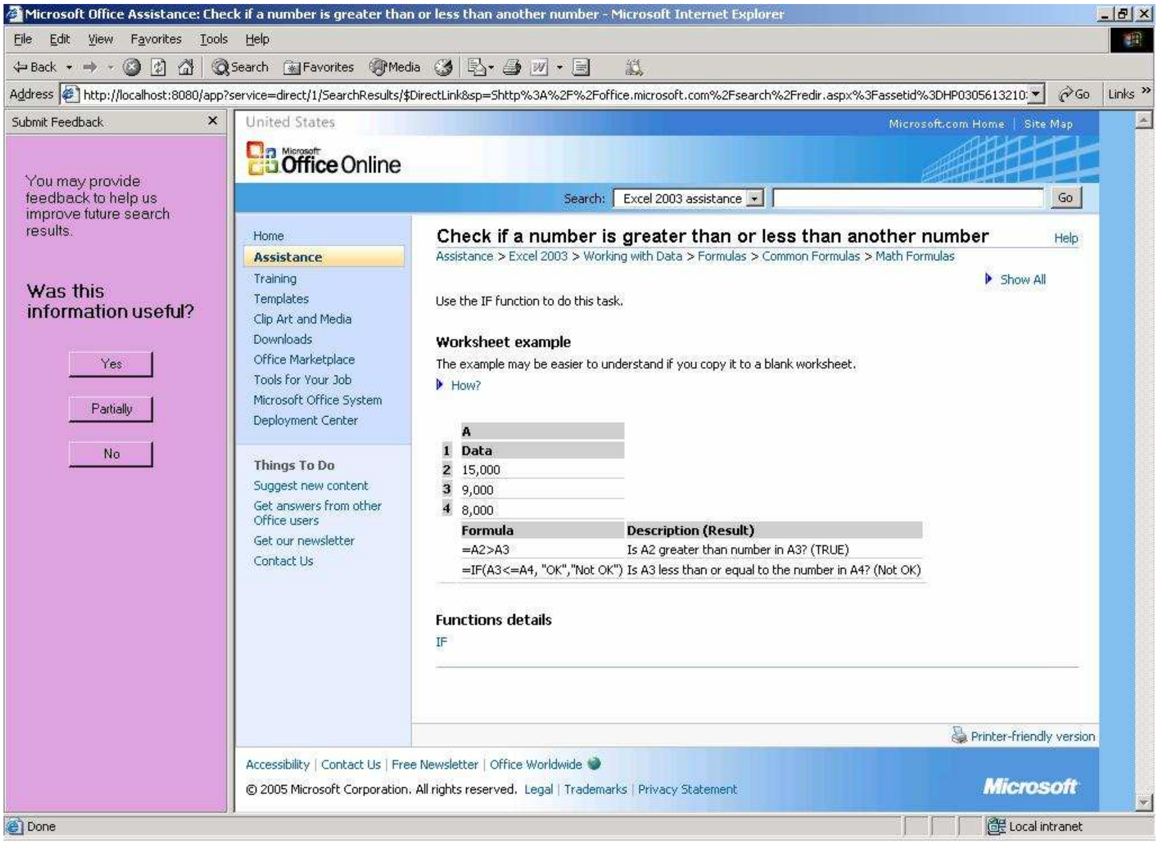
http://localhost:8080/app?service=direct/1/SearchResults/$DirectLink&sp=Shttp://office.microsoft.com/se

Figure 3-2 Voluntary pop-up window (first pilot study).

For the second pilot study, the horizontal explorer band was made vertical and moved to the left-hand-side of the IE window (see Figure 3-3). The decision to move the explorer band was motivated by the low amounts of feedback obtained with the horizontal band. As can be seen in Table 3-1, the feedback ratio (described more precisely in Section 3.3) was 0.04, meaning users only gave feedback after viewing a search result 4% of the time. While we had initially anticipated a rather low feedback ratio, 0.04 would not yield very much data for analysis and thus we attempted to find a voluntary feedback mechanism that would yield more feedback.

Another motivation for moving the explorer band was user evaluations which indicated that a band at the bottom of the screen consumed too much screen space. Furthermore, we found that unless a user completely read a Web page, they often would not look at the bottom of the IE window and could not even see the explorer band, whereas with the pane resident on the left, the user encountered it during Western left-to-right reading. As with the horizontal explorer band, the Mandorvol Browser controlled when the vertical explorer band should be shown or hidden.

While the first pilot study investigated both the horizontal explorer band and the pop-up window, the second study only measured the effectiveness of the vertical explorer band. The decision to only measure the new mechanism was based upon the desire to equally test all three feedback mechanisms for comparison.


Microsoft Office Assistance: Check if a number is greater than or less than another number - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go
Links
Submit Feedback
Yes
Partially
No
United States
Microsoft.com Home | Site Map
Office Online
Search: Excel 2003 assistance
Go
Home
Assistance
Training
Templates
Clip Art and Media
Downloads
Office Marketplace
Tools for Your Job
Microsoft Office System
Deployment Center
Things To Do
Suggest new content
Get answers from other Office users
Get our newsletter
Contact Us
Help
Show All
How?
Printer-friendly version
Accessibility | Contact Us | Free Newsletter | Office Worldwide
Legal | Trademarks | Privacy Statement
Microsoft
Done
Local intranet

http://localhost:8080/app?service=direct/1/SearchResults/$DirectLink&sp=Shttp%3A%2F%2Foffice.microsoft.com%2Fsearch%2Fredir.aspx%3Fassetid%3DHP0305613210:

You may provide feedback to help us improve future search results.

Was this information useful?

Check if a number is greater than or less than another number

Assistance > Excel 2003 > Working with Data > Formulas > Common Formulas > Math Formulas

Use the IF function to do this task.

Worksheet example

The example may be easier to understand if you copy it to a blank worksheet.

	A
1	**Data**
2	15,000
3	9,000
4	8,000

Formula	Description (Result)
=A2>A3	Is A2 greater than number in A3? (TRUE)
=IF(A3<=A4, "OK","Not OK")	Is A3 less than or equal to the number in A4? (Not OK)

Functions details

IF

© 2005 Microsoft Corporation. All rights reserved.

Figure 3-3 Vertical explorer band (second pilot study).

3.3 Results

The first pilot study had nine participants, four of which used the pop-up while the remainder used the horizontal explorer band. Although feedback statistics were collected (which will be discussed shortly), much of the real value of these experiments came from the responses to the open-ended questionnaire (see Appendix A). As mentioned previously, these responses helped shape the second pilot study. In particular, many of the users expressed that they did not like having the feedback band at the bottom because it occupied too much of the screen. They would rather have had the feedback band on the side so that they could view Web pages at full-height. Since monitors and windowed applications tend to be wider than they are tall, this seemed like a reasonable suggestion.

The participants also had varied opinions about several other attributes of the Mandorvol Browser. Although we changed the color of the explorer band from gray to pink based upon initial user feedback, one user thought the pink color was too bright. Some users would have preferred to have the feedback mechanism embedded in the Web page – unfortunately, we found this infeasible to implement. One user also would have liked to have a scale of values to choose from rather than push buttons.

Additionally, the participants generally thought that the Excel tasks were a good sample to use. The tasks that users felt most frustrated with were noted and modified or removed for the second pilot study. Despite warm user response, the Excel tasks were taking too long (more than 15 minutes) to complete and had to be revised for the second study.

The second pilot study had five participants, all of who used the vertical explorer band. For this study, the questionnaire was modified slightly in an attempt to correlate prior Excel knowledge to various other aspects of the study (see Appendix B). Every participant commented independently on the feedback pane at the end of the study. They generally did not like the pink color or how it was separated from the content since they thought it looked like a Web site banner advertisement. However, this time everyone noticed the explorer band, which had been a problematic issue with the first set of pilot tests. Moving it to the left also yielded a higher amount of feedback, although this number is skewed by a single user. The feedback ration of 0.15 for the left-hand-side explorer band is nearly five times greater than that observed with the bottom explorer band (see Table 3-1).

The participants in the second pilot test also believed the Excel tasks they completed were a good sample. The average time to complete the study reduced from 31.8 minutes to 22.2 minutes. Some additional modifications would need to be done in order to achieve our 15 minute mark, but the second pilot test confirmed we were moving in the correct direction.

A summary of the quantifiable results of both studies is presented in Table 3-1. There are three rows in the table, each corresponding to a different feedback mechanism. The rows detail the data collected for each experiment and provide a format for easy comparison.

The table is also split into three partitions vertically. The first partition indicates how much feedback was given (one of the buttons clicked) for each of the different possible options. These values are then summed and stored in the “Total” column.

The second partition relates how much feedback was given per page view. “Result Items” is the count of search result items that were explored while “Pages” is the combination of “Result Items” and any pages the user may have navigated to from a result item. The “Feedback Ratio” is the total amount of feedback given proportional to the total number of result items viewed and is thus constrained from [0, 1], since feedback can only be given once per result item. In the ideal case, the user always gives feedback and the resulting feedback ratio is 1.

The pages count is given here solely to show the number of opportunities a user actually had to give feedback. Each page navigated to via a search result item is assumed to be related to the result item and thus feedback can be collected from it. The ratio between the amount of feedback collected and the total number of pages viewed should be noted, since this would indicate the amount of feedback given per actual opportunity to give feedback. However, such a ratio would not be normalized over [0, 1] because feedback can be given at most once between a set of related pages (i.e., pages all navigated from a common result item). Thus the pages count here is only used as an insight into user browser behavior. For example, in the fourth row of the “Left” partition, it can be seen that the user had one result item but 21 page views. In all likelihood, the user stopped using our custom Office help search interface and navigated to the official

					Result		Feedback	Time		Excel	Help	Help Not	Prompts	Excel
	Yes:	Partially:	No:	Total:	Items:	Pages:	Ratio:	(min.)	Intrusive	Diff.	Useful:	Needed:	Clear:	Expertise:
Pop-up:	-	-	-	-	-	-	-	28	-	3	4	3	-	-
	4	0	0	4	5	5	0.8	11	2	3.5	4	1	y	-
	3	2	1	6	6	6	1	24	4	3	4	4	y	-
	2	5	5	12	13	17	0.92	53	4	3	3	3	y	-
Total:	9	7	6	22										
Avg.:	0.41	0.32	0.3				0.91	29	3.33	3.13	3.75	2.75		-
Bottom:	0	0	0	0	7	7	0	34	1	3	3	2	-	-
	0	0	1	1	24	24	0.04	31	2	3	4	1	y	-
	0	0	0	0	8	9	0	32	1	4	4	2	y	-
	2	0	0	2	15	17	0.13	47	2	3	5	0	y	-
	0	0	0	0	6	6	0	29	4	4	5	2	y	-
Total:	2	0	1	3										
Avg.:	0.67	0	0.3				0.04	34.6	2	3.4	4.2	1.4		-
Left:	3	3	0	6	8	25	0.75	26	3	4	5	2	y	1
	0	0	0	0	4	4	0	12	1	2	5	3	y	3
	0	0	0	0	3	3	0	40	1	4	4	3	y	3
	0	0	0	0	1	21	0	15	4	3	4	2	y	2
	0	0	0	0	3	3	0	18	1	2	4	3	y	3
Total:	3	3	0	6										
Avg.:	0.5	0.5	0				0.15	22.2	2	3	4.4	2.6		2.4

Table 3-1 Summary of pilot study results.

Office Web page to perform his searches. Without the page count, such a conclusion could not be drawn.

While the first two partitions provide summaries of data collected by the Mandorvol Browser, the third partition summarizes the responses to the quantifiable responses from the post-study surveys. As we suspected, users found the pop-up window to be very intrusive and the explorer bands to be slightly intrusive. Not surprisingly, the pop-ups had a much higher feedback ratio than the explorer band implementations.

The remaining attributes are directly related to the Excel tasks. Across both pilot studies, users found the Excel tasks to be moderately difficult. That the difficulty ratings were so similar in both pilot tests is notable since the second pilot test did have a smaller number of tasks to complete. This consistency, however, was not observed with the values for “Help Not Needed”, which indicate the number of Excel tasks each user was able to complete without using the Office help system. It was our goal to minimize this number and as such, the results of the second pilot test showed that further refinement of the Excel tasks would be necessary for the actual study.

Overall, users found the Office help system to be useful, which helped validate our decision to use Excel as the basis for our controlled experiments. The second pilot test attempted to determine any correlation between user Excel expertise and the number of questions requiring help. Unfortunately, with such a small sample size, none was detected.

3.4 Analysis

The pilot study results were instrumental in designing the final experiment. In particular, the pilot studies helped determine the best type of voluntary feedback mechanism to use, how that mechanism should be presented (e.g., color), and how the Excel tasks should be altered to meet our design goals.

Perhaps more importantly, the pilot studies highlighted a few behavioral obstacles that would be necessary to overcome before the actual study could be performed. One of the biggest problems is that users have become conditioned to filter out non-core content when browsing the Web. As a result, UI considerations for the Mandorvol Browser must be made in order to catch the user's attention without the voluntary feedback mechanism (side panel) looking like an annoying advertisement.

Another observed behavior was that since users believed they were simply completing Excel tasks, that is what they focused on. That is to say, their typical action cycle was to search for help, read a help item, try it in Excel, and if it worked, move on to the next task. This sequence of actions is similar to the goal-based approach to seeking help for a task described by Ramachandran & Young [2005]. The problem is that feedback on the utility of a result item cannot be given until the result item's contents are first tried. However, once a user tries the help offered and completes the task, their next natural action is to try to complete the next task, not to consider the previous result item and give it a rating.

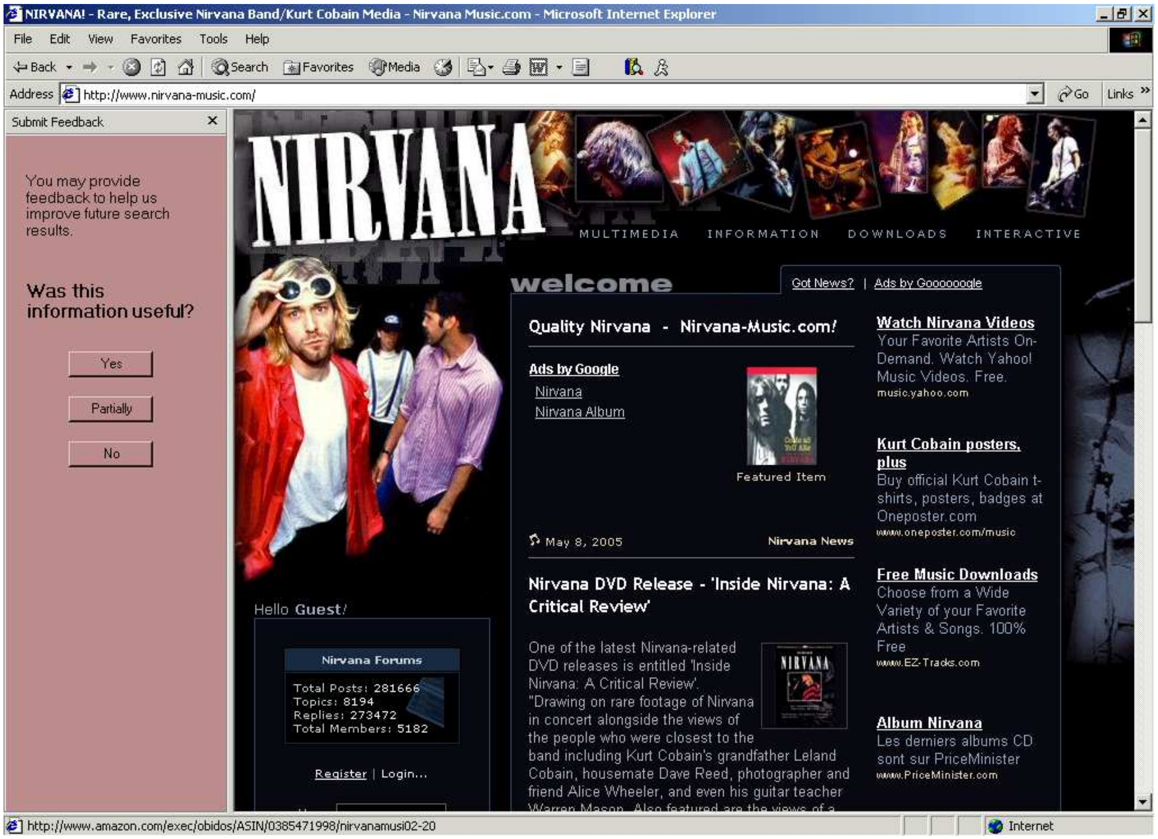
It was observed that users that were unable to find appropriate help immediately, and thus refined their search queries several times, were more likely to give feedback. This can be attributed to the fact that they were not moving on to the next task and as such, still evaluating the current result item. This behavior, while artificial in a sense, does accurately represent real world scenarios where users are typically task-oriented. The implications of these findings would serve to further refine the Excel tasks for the actual experiment.

4 Experiment

This chapter details the structure of the experiments and the motivation behind the chosen structure. The types of data collected by the Mandorvol Browser and their storage format are also discussed.

4.1 Mandorvol Browser

The user feedback mechanisms were implemented as an add-on for the Microsoft Internet Explorer (IE) Web browser. The voluntary feedback mechanism was a noticeable, but non-intrusive pane that spanned the left-hand-side of the IE window (Figure 4-1).


NIRVANA! - Rare, Exclusive Nirvana Band/Kurt Cobain Media - Nirvana Music.com - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go
Links
Submit Feedback
Yes
Partially
No
NIRVANA
MULTIMEDIA
INFORMATION
DOWNLOADS
INTERACTIVE
welcome
Got News? | Ads by Goooooogle
Featured Item
Nirvana News
Internet

Address: http://www.nirvana-music.com/

You may provide feedback to help us improve future search results.

Was this information useful?

Quality Nirvana - Nirvana-Music.com!

Ads by Google
Nirvana
Nirvana Album

May 8, 2005

Nirvana DVD Release - 'Inside Nirvana: A Critical Review'

One of the latest Nirvana-related DVD releases is entitled 'Inside Nirvana: A Critical Review'. "Drawing on rare footage of Nirvana in concert alongside the views of the people who were closest to the band including Kurt Cobain's grandfather Leland Cobain, housemate Dave Reed, photographer and friend Alice Wheeler, and even his guitar teacher Warren Mason. Also featured are the views of a

Watch Nirvana Videos
Your Favorite Artists On-Demand. Watch Yahoo! Music Videos. Free.
music.yahoo.com

Kurt Cobain posters, plus
Buy official Kurt Cobain t-shirts, posters, badges at Oneposter.com
www.oneposter.com/music

Free Music Downloads
Choose from a Wide Variety of your Favorite Artists & Songs. 100% Free
www.EZ-Tracks.com

Album Nirvana
Les derniers albums CD sont sur PriceMinister
www.PriceMinister.com

Hello Guest!

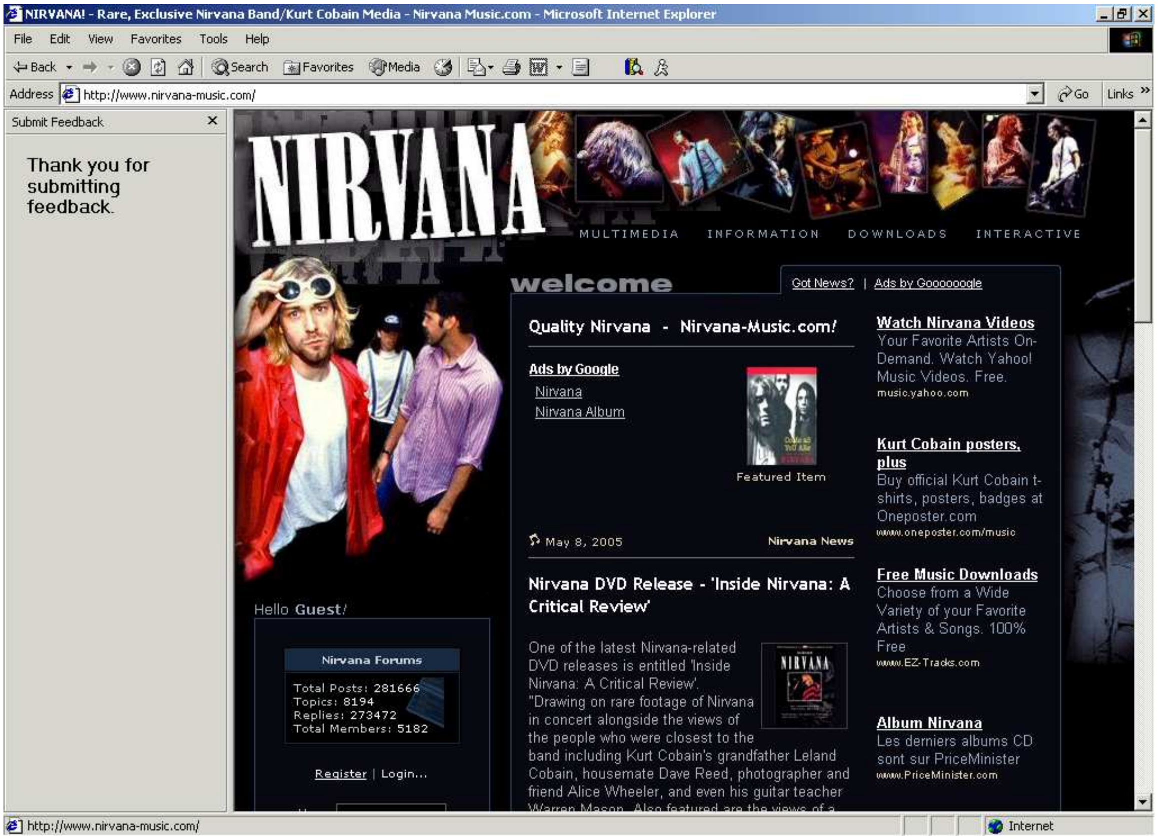
Nirvana Forums
Total Posts: 281666
Topics: 8194
Replies: 273472
Total Members: 5182

Register | Login...

http://www.amazon.com/exec/obidos/ASIN/0385471998/nirvanamusi02-20

Figure 4-1 The Mandorvol Browser voluntary feedback mechanism.

In order to prevent people from submitting feedback more than once for a given search result item, the feedback band transitioned from a feedback prompt to a "thank you" message once a feedback value is chosen (Figure 4-2). Ideally, we would have simply made the feedback band disappear, but we were not able to do so due to technical limitations in the explorer band API.



Address http://www.nirvana-music.com/

Submit Feedback

Thank you for submitting feedback.

Quality Nirvana - Nirvana-Music.com!

Ads by Google
Nirvana
Nirvana Album

May 8, 2005 Nirvana News

Nirvana DVD Release - 'Inside Nirvana: A Critical Review'

One of the latest Nirvana-related DVD releases is entitled 'Inside Nirvana: A Critical Review'. "Drawing on rare footage of Nirvana in concert alongside the views of the people who were closest to the band including Kurt Cobain's grandfather Leland Cobain, housemate Dave Reed, photographer and friend Alice Wheeler, and even his guitar teacher Warren Mason. Also featured are the views of a

Watch Nirvana Videos
Your Favorite Artists On-Demand. Watch Yahoo! Music Videos. Free.
music.yahoo.com

Kurt Cobain posters, plus
Buy official Kurt Cobain t-shirts, posters, badges at Oneposter.com
www.oneposter.com/music

Free Music Downloads
Choose from a Wide Variety of your Favorite Artists & Songs. 100% Free
www.EZ-Tracks.com

Album Nirvana
Les derniers albums CD sont sur PriceMinister
www.PriceMinister.com

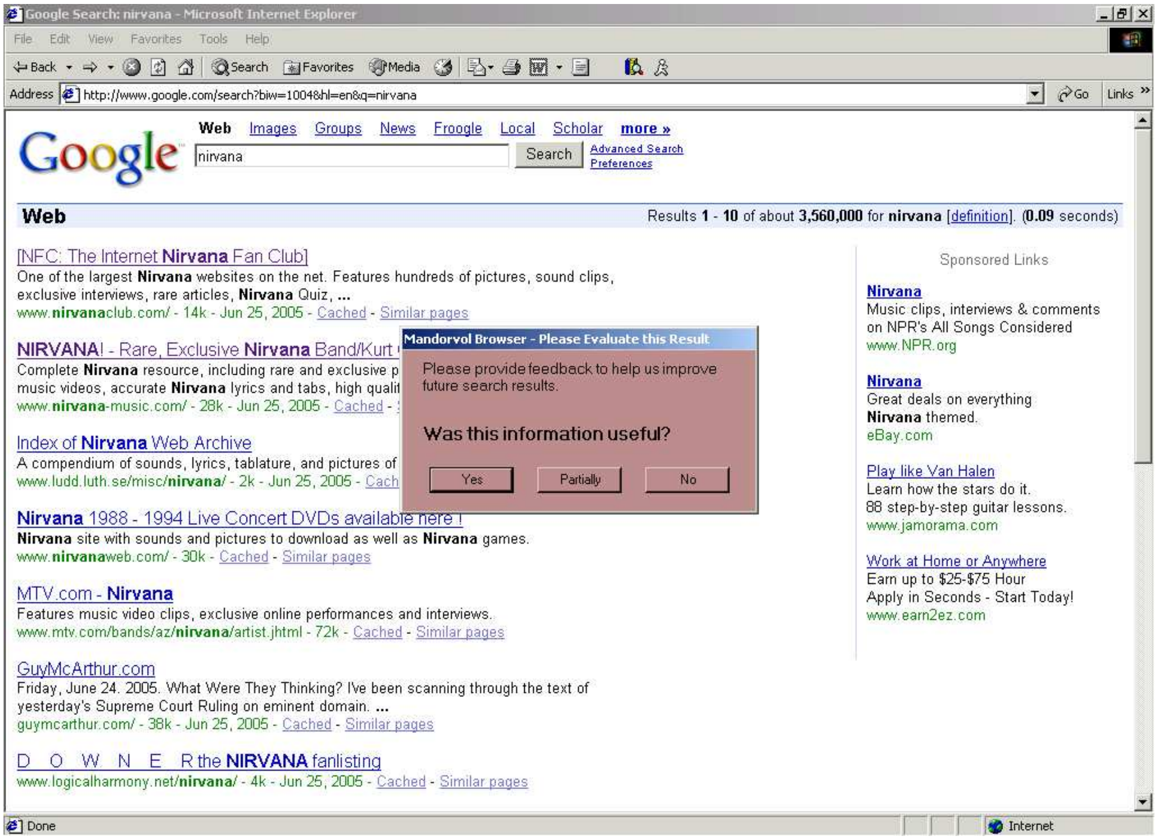
Hello Guest!

Total Posts: 281666
Topics: 8194
Replies: 273472
Total Members: 5182

http://www.nirvana-music.com/

Figure 4-2 The voluntary feedback mechanism after user provides feedback value.

The mandatory feedback mechanism was a pop-up window that, unlike the vertical pane, could not be ignored, requiring the user to provide feedback in order to continue with the search session (Figure 4-3). It simply disappeared once a feedback value was given, preventing the user from providing more than one feedback value.


Google Search: nirvana - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go Links »
Web Images Groups News Froogle Local Scholar more »
Search
Advanced Search
Preferences
Yes
Partially
No
Done
Internet

Address: http://www.google.com/search?biw=1004&hl=en&q=nirvana

Search: nirvana

Web Results **1** - **10** of about **3,560,000** for **nirvana** [definition]. (**0.09** seconds)

[NFC: The Internet **Nirvana** Fan Club]
One of the largest **Nirvana** websites on the net. Features hundreds of pictures, sound clips, exclusive interviews, rare articles, **Nirvana** Quiz, **...**
www.**nirvana**club.com/ - 14k - Jun 25, 2005 - Cached - Similar pages

NIRVANA! - Rare, Exclusive **Nirvana** Band/Kurt
Complete **Nirvana** resource, including rare and exclusive p
music videos, accurate **Nirvana** lyrics and tabs, high qualit
www.**nirvana**-music.com/ - 28k - Jun 25, 2005 - Cached -

Index of **Nirvana** Web Archive
A compendium of sounds, lyrics, tablature, and pictures of
www.ludd.luth.se/misc/**nirvana**/ - 2k - Jun 25, 2005 - Cach

Nirvana 1988 - 1994 Live Concert DVDs available here !
Nirvana site with sounds and pictures to download as well as **Nirvana** games.
www.**nirvana**web.com/ - 30k - Cached - Similar pages

MTV.com - **Nirvana**
Features music video clips, exclusive online performances and interviews.
www.mtv.com/bands/az/**nirvana**/artist.jhtml - 72k - Cached - Similar pages

GuyMcArthur.com
Friday, June 24. 2005. What Were They Thinking? I've been scanning through the text of yesterday's Supreme Court Ruling on eminent domain. **...**
guymcarthur.com/ - 38k - Jun 25, 2005 - Cached - Similar pages

D O W N E R the **NIRVANA** fanlisting
www.logicalharmony.net/**nirvana**/ - 4k - Jun 25, 2005 - Cached - Similar pages

Sponsored Links

Nirvana
Music clips, interviews & comments on NPR's All Songs Considered
www.NPR.org

Nirvana
Great deals on everything **Nirvana** themed.
eBay.com

Play like Van Halen
Learn how the stars do it.
88 step-by-step guitar lessons.
www.jamorama.com

Work at Home or Anywhere
Earn up to $25-$75 Hour
Apply in Seconds - Start Today!
www.earn2ez.com

Mandorvol Browser - Please Evaluate this Result

Please provide feedback to help us improve future search results.

Was this information useful?

Figure 4-3 The Mandorvol Browser mandatory feedback mechanism.

4.1.1 Modes of Operation

The Mandorvol Browser operated in four modes of operation in order to allow experiments to test the effects of both the feedback mechanism and the search scenario on the quantity and quality of user feedback, as shown in Table 4-1.

		Scenario	
		Controlled	**Uncontrolled**
Feedback Type	**Mandatory**	Mandatory Controlled	Mandatory Uncontrolled
	Voluntary	Voluntary Controlled	Voluntary Uncontrolled

Table 4-1 The four different modes of operation for the Mandorvol Browser.

The values *uncontrolled* and *controlled* described the scenario under which the experiment is run whereas the values *voluntary* and *mandatory* indicate the type of feedback mechanism shown to the user during the course of the experiment. While the values were considered in pairs, they can be best described individually.

4.1.1.1 Uncontrolled

The purpose of the experiments was to collect feedback data from users as they issued queries against a search engine. Search engines can interface to a wide variety of data sources, the most commonly used is the Internet. The scope of the search activities provided by the Internet can be considered to be unbounded and thus corresponds to our uncontrolled scenario.

The uncontrolled experiments were designed to model user behavior in a general search environment. The users were allowed to search for anything using the Google Web search engine for a period of 15 minutes. As search result items were presented, the Mandorvol Browser presented a feedback mechanism to collect a user's rating of the content. The type of the feedback mechanism used was dependent upon the corresponding mandatory or voluntary value.

As the idea of the uncontrolled experiments was to model actual user behavior, it made the most sense to allow users to use their preferred search engine. Unfortunately, calibrating the Mandorvol Browser for a search engine is a massive undertaking and our resources were limited. As a result, we were forced to make a compromise and support a single search engine. Google was the search engine of choice since it is currently the most popular Web search engine on the WPI campus.

4.1.1.2 Controlled

In order to limit the search domain, a set of controlled experiments was designed to collect data from searches over Microsoft Excel help assets. For these experiments, the users were asked to complete a series of Microsoft Excel tasks that were chosen to be easy to complete with the correct information, but not commonly known, so that a user would be likely to need to search for help. Microsoft Excel was used for the controlled experiments because it is a software package many people are familiar with while having many features most users will not know how to use without consulting its extensive help system. Thus, we were able to control the search domain and direct the search queries, allowing us to shape the environment for supplying feedback. An additional design goal prompting the use of Microsoft Excel was to show how such a feedback system could be applied to the Microsoft Office product suite.

In order to limit the variation between the controlled and uncontrolled experiments, a Java Web application was written as an interface to the Excel help assets. The interface was designed to look and feel very much like the interface for the Google Web search engine. Similar to the uncontrolled experiments, as search results were rendered, the Mandorvol Browser presented a feedback mechanism to collect a user's feedback rating for the content. Indeed, user interaction with the Mandorvol Browser was as identical as possible across both the controlled and uncontrolled experiments.

The tasks were designed so that the most obvious search terms would not yield immediately useful results. We observed during the pilot studies that when users must re-evaluate their search queries in order to complete a task, they are much more likely to provide feedback. In keeping with our time goal of 15 minutes to complete all tasks, we provided each subject with a pre-filled Excel worksheet and used the following three tasks for the experiment:

1. **Calculate the average of all the values in column A that are greater than 25.**
2. **Determine the rank of the number in cell A4.**
3. **Have the text in column A displayed in Red if the value is greater than 10.**

The two primary sources of ambiguity in potential search terms are the use of conditionals (e.g., "greater than") and the use of arrays of values. The provided dataset stored 50 values in column A as a means of deterring users from completing the tasks by inspection. At the end of the study, each subject was asked to upload the modified Excel file to our Web site so we could validate that the users in fact completed the study. This latter point was important in correlating the amount of feedback given with the set of tasks – if a user did not complete all the tasks, then the feedback given by that user was discarded.

4.1.1.3 Voluntary

The voluntary feedback mechanism was non-intrusive and did not force the user to provide feedback. As indicated in Section 3.2, the voluntary feedback UI component was implemented as a vertical explorer band for IE. An explorer band was chosen because it is a standard way of enhancing IE and would thus be familiar to users. The actual design of the UI component, i.e., the colors, location, etc., was driven by user comments during the pilot studies (see Chapter 3).

Experiments using a voluntary feedback mechanism did not require users to provide feedback in order to complete the study. The study directions made each user aware of the feedback mechanism prior to the start of the experiment so that they did not confuse it with banner advertisements, which are typically displayed as vertical bars in a Web page. However, once the study began, the user was not again coerced into looking at the feedback mechanism. As search results were rendered, the Mandorvol Browser displayed the explorer band, which prompted users for feedback, but the explorer band was separated from the content of the search result and thus was a passive device.

4.1.1.4 Mandatory

The mandatory feedback mechanism was a pop-up window that could not be closed unless the user provided a feedback value for a search result. These experiments were thus mandatory in the sense that all subjects were required to provide feedback for search results in order to complete the study.

The pop-up window was designed to look identical to the voluntary feedback mechanism, differing only in size, location, and a means of closing the UI component (the voluntary UI component had a close button whereas the pop-up windows did not). By conveying the same message in a consistent format, we were able to effectively measure the key point of variation: whether or not the user was forced to provide feedback.

4.1.2 Data Recorded

The format of the data collected was very similar to that collected by the previous Microsoft Curious Browser project [Fox et al. 2005] – a byproduct of using Microsoft's Curious Browser code as the basis for the Mandorvol Browser. The data was stored in a Microsoft SQL Server 2000 database and the database schema is an augmented version of the schema used by the Curious Browser (see [Fox et al. 2003] for more details on the Curious Browser database schema). The Mandorvol Browser makes use of an `ExperimentType` table that holds static representations of each of the experiment types and a `MandorvolBrowserUser` table that uniquely identifies each user of the Mandorvol Browser and associates them with a particular experiment type. No further modifications were made to the database schema, allowing for maximum code reuse.

The collected data can be classified as either explicit data, which is actively provided by the user, or implicit data, which is collected by the Mandorvol Browser based upon the user's search behavior. The explicit data consists of all search queries and their corresponding search result lists. The explicit data also consists of any feedback provided by the user. The feedback values are classified as *Satisfied*, *Partially Satisfied*, and *Dissatisfied* and correspond to the *Yes*, *Partially*, and *No* buttons in the feedback mechanism, respectively.

The implicit data is precisely that which is collected by the Curious Browser without any indication of "end of search session". That is to say, the Mandorvol Browser treats all queries as new search sessions and does not attempt to determine whether a query is a refinement of a previous query or a new search. The decision to remove this functionality was driven by the nature of the voluntary feedback mechanism. In the Curious Browser, everything was pop-up-window based and when it detected what it believed to be an end of search session, the user was presented with a pop-up that prompted for a feedback value for the overall search. It would be unnatural to do something similar with an explorer band, as used for the voluntary feedback mechanism in the Mandorvol Browser, and thus, after discussions with Microsoft, it was decided that the feature was not very necessary and could be removed.

As with the Curious Browser, the Mandorvol Browser collects data related to page navigation and user behavior on each page visited. Of particular interest is that the Mandorvol Browser detects when the user visits a search result list and when a search result is visited. Using this data, it can be determined how many search results the user needed to look at before finding the required information. It can also be seen whether the user navigated away from a search result, which is useful in determining page correlation, i.e., whether links from a given search result are useful to the user. Additionally, by comparing the timestamps between successive page views, the amount of time the user spent looking at a search result, the dwell time, can be calculated. The dwell time has previously shown in [Fox et al. 2005] to be a very useful implicit indicator for training a classifier.

4.2 Methodology

This section describes how the Mandorvol Browser was deployed across the Worcester Polytechnic Institute campus and how we attracted users to the study. Instructions on how to use the Mandorvol Browser are also detailed.

4.2.1 Mandorvol Browser Installation

The Mandorvol Browser was installed in several public computer access labs on the WPI campus. The installation was performed by the Windows administration group in the Campus Computing Center (CCC) at WPI. We provided the CCC with a Microsoft Windows Installer (MSI) file that installed the necessary files and registry entries. They created a group policy that ensured that the Mandorvol Browser was installed in all the computers in the three primary public access computer labs on campus. The installation procedure was thus automated as much as possible, and more importantly, easy to update. In fact, initial deployment of the Mandorvol Browser uncovered an “off-by-one” issue not detected during testing. Unfortunately, the issue affected the choice of mode of operation, so the four types were not evenly distributed, but we were able to quickly fix it and update the group policy with the new installer, which reinstalled the Mandorvol Browser upon each computer’s daily reboot cycle.

Due to the group policy however, great care had to be taken to guarantee that the MSI file would install the Mandorvol Browser correctly for users with unprivileged computer access. Furthermore, it was necessary to make all registry and file accesses local to the computer rather than to the user's roaming profile. Meeting these two goals was time-intensive to build and test, but the end result was an installation file that could easily be deployed and redeployed.

4.2.2 Encouragement

Students at the Worcester Polytechnic Institute were solicited via email announcements and flyers that were placed near computers in the public access computer labs. An example message that was broadcast to the student population is in Appendix H. As can be noted in the message, there was a set of prizes that were raffled off to participants in order to attract as many users as possible.

In addition to the raffled prizes, we suggested that the Computer Science Department faculty reward students with academic credit, e.g., extra points on an exam, for their participation in the study. Three of the faculty members obliged and offered credit to their students. Two of the classes for which credit was offered were undergraduate computer science courses while the other was a graduate computer science course.

We believed that providing encouragement for completing the study would increase the quantity of data collected while not adversely affecting quality. The users were rewarded solely for participation and as such the incentives should not have affected the actual feedback values given. Additionally, the encouragement factors were just that, encouragement. There was no requirement for any student to complete the study: all student participation was done on a voluntary basis.

4.2.3 Mandorvol Browser Usage

Every user of the Mandorvol Browser began the study by reading the directions shown in Appendix C, which they were directed to via the aforementioned email announcements and flyers. Once the users read through the instructions and activated the Mandorvol Browser IE add-on, they were redirected to a further set of directions tailored for each of the four different experiment types. The experiment types were chosen randomly and were logged in the database along with a way to uniquely identify each individual. This unique identifier was also used to ensure that each participant could only participate in the study once.

The experiment-specific instructions were structured to minimize variation as much as possible between the different experiment types. Thus, the instructions regarding the controlled tasks and the uncontrolled tasks, with both types of feedback mechanisms, are very similar. Likewise, the text explaining the feedback mechanism is very similar for the voluntary tasks and the mandatory tasks, across both uncontrolled and controlled scenarios. These specialized pages can be seen in Appendices D - G for *mandatory controlled*, *mandatory uncontrolled*, *voluntary controlled*, and *voluntary uncontrolled*, respectively.

Throughout the duration of each experiment, implicit user behavior and all explicit feedback values were transparently logged to the database server. Since no batching of data transmission was performed in the data storage, the effect of users performing unplanned actions that could cause potential data loss was minimized. In fact, it allowed for a nicer user experience because once the user was done with the study, they simply needed to close the IE window, which is the most natural workflow action for a user when done with a browsing task. Once the IE window was closed, a shutdown procedure was invoked that disabled the Mandorvol Browser so that any future users of the computer would be required to explicitly re-enable the Mandorvol Browser before it would begin collecting data again.

4.3 Results

This section presents statistics about the data collected throughout the course of the study. The core foci of the data are the amount of feedback collected and the distribution of the feedback values. In some cases, explanations of the distribution of the data are briefly presented. A full analysis of the data however is deferred until Section 4.4.

4.3.1 Demographics

There were 161 participants in the study and the population was fairly evenly distributed among the four experiment types. The *mandatory controlled* experiments had a smaller population size due to a software defect that was found and corrected early in the experiment. The other three experiments had populations that were approximately the same size. The population size for each experiment type is summarized in

Table 4-2, leading to the overall population distributions represented in Table 4-3.

	Controlled	Uncontrolled
Mandatory	28	45
Voluntary	48	40

Table 4-2 Experiment type distribution.

	Controlled	Uncontrolled
Mandatory	17.39%	27.95%
Voluntary	29.81%	24.84%

Table 4-3 Experiment type distribution.

In addition to the experiment type distributions, we have approximate values for the class and major of each participant. By investigating the subject demographic, we can better understand how the demographic may have affected our results. The demographic attributes were retrieved without user interaction via a campus-wide directory. Unfortunately, the directory did not have up-to-date information for all students, but the data we were able to extract was valuable nonetheless.

Class	Number of Participants	Distribution
2005	20	12.42%
2006	32	19.87%
2007	29	18.01%
2008	30	18.63%
Graduate	29	18.01%
Unknown	22	13.66%

Table 4-4 Study population class distribution.

Table 4-4 shows the decomposition of the study population by school class. Outlying values and values for students missing from the campus directory are lumped together in the miscellaneous category.

The study population was fairly evenly distributed across all classes. Since the Mandorvol Browser could only be run in public access computer labs on campus, there were initial concerns that older students that live off campus would not be likely to complete the study. This table indicates that such a restriction had no more impact on students that live off campus than those that live on campus. That is not to say that we would not have had more user participation if we had allowed people to install the Mandorvol Browser on their own computers, simply that it affected all students equally.

Major	Number of Participants	Distribution
Biology/Biotechnology	8	4.96%
Computer Science	69	42.85%
Electrical/Computer Eng.	17	10.55%
Management	4	2.48%
Math	3	1.86%
Mechanical Eng.	10	6.21%
Unknown	50	31.05%

Table 4-5 Study population major distribution.

The major distribution values for the study population however, are far more skewed than the class data. Computer science students and closely related electrical/computer engineering students represented the majority of the population. Such a sharp divide in the major distribution may be attributed to the grade encouragement offered to computer science students, the close affiliation between the computer science and electrical/computer engineering departments, and the fact that such students tend to be in the public access computer labs more often than students in other disciplines.

4.3.2 Feedback Ratio

One metric to look at is the *feedback ratio*, which we have defined to be the ratio between the total number of times feedback was given to the total number of search results. Since feedback can only be given once per search result, the feedback ratio takes the range [0, 1]. The feedback ratio is an indication of how well each feedback mechanism solicits feedback from a user. The feedback ratios for each of the four experiments can be seen in Table 4-6.

	Controlled	Uncontrolled
Mandatory	0.95	0.98
Voluntary	0.75	0.92

Table 4-6 Feedback ratios.

Note that we would expect the feedback ratio values for a mandatory feedback mechanism to be 1. In actuality, we observed numbers slightly under this value. The smaller numbers can be attributed to users closing the IE window when done with their experiments. The mandatory feedback pop-up only appears when the user leaves a search result item so as to limit interruption of the user's workflow [Bailey et al. 2000]. When the user is viewing a search result item however, and chooses to close the IE application, the internal state machine did not detect this as leaving a search result item, and thus the user was not prompted to give feedback. However, the number of search results viewed was incremented as soon as the user clicked on the search result link. The net result is a small, but noticeable skew in the feedback ratio values.

Closely related to the feedback ratio is the *feedback to opportunity ratio*, which we have defined to be the ratio between the total number of times feedback was given to the total number of pages viewed. The difference between this and the feedback ratio is subtle, but important. While viewing a search result, a user may navigate away from the search result to other pages linked from the search result. Each page navigated to increments the total number of pages viewed count, and on each such page the user is given the opportunity to provide feedback. However, feedback can only be given once for a search result and all the pages navigated to from that search result. There is no defined range for the feedback to opportunity ratio, but it is bounded by 0 on the low end and 1 on the high end. The value of knowing the feedback to opportunity ratio is that it can be seen how much feedback is given over a complete search session. The feedback to opportunity ratios for each of the four experiments can be seen in Table 4-7.

	Controlled	Uncontrolled
Mandatory	0.63	0.57
Voluntary	0.41	0.61

Table 4-7 Feedback to opportunity ratios.

4.3.3 Feedback Values Distribution

The collected feedback is grouped by value for each experiment type and is presented in Table 4-8. These feedback values are normalized in that each value is a percentage of the total amount of feedback for a given experiment – *No Feedback* values are omitted. The normalized values indicate the feedback value distributions when feedback is given.

	Satisfied	Partially Satisfied	Dissatisfied
Mandatory Controlled	29.66%	23.57%	46.77%
Mandatory Uncontrolled	46.85%	22.28%	30.87%
Voluntary Controlled	50.76%	16.67%	32.58%
Voluntary Uncontrolled	49.42%	21.71%	28.88%

Table 4-8 Normalized feedback distributions.

Table 4-9 shows the feedback value distributions for each experiment type when *No Feedback* values are considered. These values are the feedback type distributions for all queries, regardless of whether or not feedback is provided. As can be seen, the percentage values drop, in some cases considerably, from the values in Table 4-8.

	Satisfied	Partially Satisfied	Dissatisfied	No Feedback
Mandatory Controlled	28.06%	22.30%	44.24%	5.40%
Mandatory Uncontrolled	45.80%	21.78%	30.18%	2.23%
Voluntary Controlled	37.85%	12.43%	24.29%	25.42%
Voluntary Uncontrolled	45.37%	19.93%	26.51%	8.19%

Table 4-9 Feedback ratios with *No Feedback* values.

4.4 Analysis

The data collected during the study has yielded some insight into user behavior with regards to providing feedback. Certainly, with such a small population it is not possible to make broad conclusions from the study. However, the population is sizable enough to indicate trends that may be worthy of further consideration.

The sample population was composed of people that are generally quite proficient with modern computing technology. This was further compounded by the large percentage of students in computer-related academic programs. It is not entirely clear what the consequence of this is, since in the general sense, the sample population was mostly homogenous. It would be reasonable, however, to expect to see different results with users that are not as familiar with computing technology. For example, most students on the WPI campus are quite familiar with the Microsoft applications used during the study, the Google search engine, and even responding to pop-up windows and explorer bands. Users without this background, however, may have had considerable difficulty with study. In many ways, the Mandorvol Study was tailored to the WPI student body.

Another factor that may have potentially affected the outcome of the study is student bias against Microsoft. Once again, there is no real way to measure this, but especially in the computer-related fields of study, students tend to view Microsoft in an unfavorable light. In anticipation of this, we were careful not to mention that Microsoft had funded the study when we invited students to participate. However, some students prefer not to use Microsoft products, and that may have affected their attitude towards the study, and ultimately the quality of the collected data.

In the end, it did not appear as though student feelings about Microsoft had a large impact on the overall study, as we observed very large feedback ratios for the experiments that collected voluntary feedback. If such bias did play a large role, we would expect to have seen low feedback ratios with the voluntary feedback mechanism since it is easier not to provide feedback than it is to provide it.

During the pilot studies we saw feedback ratios of 0.035 and 0.15 in the controlled scenario (see Table 3-1). While the Excel tasks went through several modifications, as did the explorer band, in the full study we saw feedback ratios with a 5-22 times increase in the controlled case. In the uncontrolled case, the feedback ratio was even larger, but since the pilot studies did not examine the uncontrolled case, there is no baseline to compare against.

The choice of feedback mechanism clearly had an effect on the amount of feedback collected. However, we collected far more data with a voluntary feedback mechanism than we had anticipated. Indeed, the feedback ratio values for the mandatory and voluntary feedback mechanisms in the uncontrolled scenario experiments are quite close to one another. Nevertheless, the mandatory feedback mechanism did collect more feedback than the voluntary mechanism, supporting our **H1** hypothesis.

We believe that the choice of feedback mechanism will also have an effect on the data quality, although we were not able to directly correlate the two. The thought is that when presented with pop-up windows, users will take the path of least resistance and click the feedback value button that is closest to the mouse cursor. Furthermore, we believed that if users became frustrated with the pop-up windows, it is likely that they may start providing blatantly incorrect feedback values as a way of “punishing” the entity collecting feedback values.

In informal conversation with study participants, some had told us that they had clicked feedback buttons that did not accurately represent their true feeling about the utility of the search result in order to make the pop-up window disappear as quickly as possible. With that in mind, we had initially proposed a voluntary pop-up window as a third feedback mechanism to use for the experiments. The pop-up window would look just like the one used as the mandatory feedback mechanism, except that it would have an “X” button to close the window without explicitly providing a feedback value. Since it would appear that users were not providing improper feedback values as a way to subvert the study, but rather as a way to proceed with their workflow, we do not believe that a voluntary pop-up would yield higher data quality than a mandatory pop-up. If this is true, the user is not annoyed by the fact that he must give feedback, but rather that the pop-up is present until a button is clicked, and as such, the user will still click the closest target.

We took great care not to influence people to give feedback, but in order to make sure users did not confuse our explorer band with a banner advertisement, they were made aware of the feedback band via prompts as shown in Appendix F & Appendix G. As a result, it is not certain how the color and location of the voluntary feedback mechanism affect the amount of feedback collected. Additionally, as noted in Section 4.1.1.2, the tasks in the controlled scenarios were chosen in a manner that would lead users to re-evaluate a page. While the intention was to help ensure users were able to utilize the information given by the Office help system before providing feedback, there also appears to be a correlation between page re-evaluation and feedback response rate. This design decision may have biased the participant to give more feedback in the *voluntary controlled* case than they would have in a general help system search.

Our data seems to indicate that a distinctive voluntary feedback mechanism yields a higher quantity of feedback responses than a mechanism that simply blends in with the user application. However, it is possible that the users were simply driven to give more feedback upon reading the text alerting them to the feedback mechanism. It cannot be determined if this is a result of the user somehow feeling obligated to give data, perhaps to improve the study results, or if users are genuinely more apt to give feedback if they are made aware of the ability to do so.

The data in Table 4-6 and Table 4-7 show that users tend to provide more feedback when performing Web searches at their leisure. This is evidenced by the feedback ratio values, which are typically higher for the uncontrolled scenarios than they are for the controlled scenarios. We believe the difference in the amount of feedback provided is correlated to a task-oriented versus leisurely mindset for the user. The controlled experiments consisted of a set of tasks to perform in Excel that the user wanted to complete as quickly as possible. As soon as the task was completed, the user moved from one search query to a completely unrelated one. The user was not interested in going back to the previous search result to provide feedback; this corresponds to the *voluntary controlled* scenario. However, when leisurely searching the web, the user is often interested in general topics rather than specific answers, and such, the user will spend more time evaluating search results.

The data in Table 4-8 and Table 4-9 show that when users are leisurely browsing the Web, the feedback they provide tends to be positive. This finding is consistent with previous work [Saito & Ohmura 1998] that showed that when users have a mental model of what it is they are searching for, they tend to be more satisfied with their search results. The uncontrolled scenario experiments had significantly higher *Satisfied* values than the controlled scenario experiments regardless of the type of feedback mechanism used. We believe this, too, is related to the task-oriented versus “at leisure” mentality.

When leisurely browsing the Web, users tend to search for items that they are already familiar with, and thus the users are already familiar with the search results. Users know what they are looking for in this scenario and are better able to gauge the search results. In a task-oriented scenario, the user is attempting to complete a task using an unknown process. The user is not sure what to search for and has no a priori expectations about the search results. Furthermore, there is no personal connection to the results: either they help complete the task at hand or they do not. In such scenarios, we expect to see more diversified feedback values which relate directly to the specific utility of the search result.

5 Classifier Construction

Having collected all the data from the experiment, there was a need to analyze the data to uncover any relationships between attributes. Ultimately, we wanted to be able to predict user satisfaction with a given search result using that data. The nature of the problem lent itself naturally to machine learning techniques.

While Chapter 4 detailed the field experiment and descriptive statistics about our data, this chapter provides an in-depth analysis of the collected data. Using machine learning techniques and tools, the data is processed in such a way as to address our hypotheses and answer the research question.

5.1 Decision Tree Construction

Decision tree classifiers are a common family of algorithms employed in the field of machine learning for predicting classifications. The trees constructed by such classifiers represent a set of rules, with each path through the tree ultimately leading to a leaf that represents a classification value. More information on the mechanics of decision trees can be found in Section 2.2. The Weka[2] machine learning program was used to construct the decision trees. Weka is an open-source data mining tool written in the Java programming language and licensed under the Gnu's Not Unix (GNU) General Public License (GPL). The software was developed at the University of Waikato, New Zealand and complements Witten and Frank's book [2000] on data mining techniques.

5.1.1 Motivation

The choice of using a decision tree classifier over other machine learning approaches was largely driven by the desire to have a set of human readable rules that describe what leads to a user's satisfaction. Since each path through a decision tree encodes such a rule, as described in Chapter 2, decision trees are a good data representation for what we wanted to achieve. Additionally, we knew there were causal relationships between implicit behavior indicators and user satisfaction values

[2] Weka 3.5.2 was used for the experiments. It is available from http://www.cs.waikato.ac.nz/ml/weka/.

due to previous work in the field [Claypool et al. 2001a; Claypool et al. 2001b; Cen et al. 2002], so the use of a classifier that can account for these relationships, such as decision trees, was a natural choice. Finally, Microsoft had used decision trees in its Curious Browser project [Fox et al. 2005], so our use of decision trees provides a fairly straightforward way of comparing results with existing work.

5.1.2 Data Preparation

Weka uses a custom file format called Attribute-Relation File Format (ARFF) that is very similar to a file of simple comma-separated values. During the experiment, all of the collected data was stored in a series of tables in a relational database. Using SQL queries that joined the tables based upon the unique user identifiers and other foreign keys in the database, the necessary data was extracted from the database. A custom Python script was then used to process the data accordingly to create the ARFF files suitable for use with Weka.

During the experimental design, we did not know precisely which data attributes would be good predictors of user satisfaction. As a result, the Mandorvol Browser was programmed to collect as much data about the user's interaction with the Web browser and the general computing environment as it could, subject to the limitations imposed by the executing environment. Before we could proceed with our data analysis, we were tasked with choosing a subset of the total set of attributes available to use for training a classifier.

In an attempt to not bias results, we initially considered all attributes as candidates for our classifier. We employed a hold-one-out strategy for determining whether a particular attribute positively contributed to classification accuracy. The basic idea was to build a classifier both with and without a particular attribute being present and then comparing the classification accuracies. If the decision tree without the attribute had a classification accuracy that was at least as good as the decision tree with the attribute present, then the attribute was deemed superfluous and removed. This approach is very similar to Quinlan's reduced-error pruning [Quinlan 1987], which is used for removing unnecessary nodes from decision trees. In fact, we had initially expected the decision tree construction algorithm to prune away all unnecessary attributes, but found in practice the automatic pruning method still required some human involvement in the form of attribute pruning. The decision node pruning process, however, worked without intervention.

Once we discovered an attribute that did not positively contribute to classification accuracy, we investigated the rule representation of the tree to understand why that was the case. In nearly all cases, we removed attributes because they were intimately tied to an individual subject, leading to overfitting of the decision tree. For example, we found that time-based attributes were correlated to an individual subject because it was hardly ever the case that more than one person was participating in the experiment simultaneously. Likewise, terms used in search queries were tightly coupled to the individual. These discoveries were not necessarily intuitive to us at first, but upon inspection of the data and the rules generated from the decision tree, we were able to establish that these correlations did exist.

When the attribute reduction process was completed, a total of fourteen different attributes remained. These attributes and their associated values constitute a single datum for the data that is used to build a decision tree in our data analysis.

Table 5-1 summarizes these attributes in alphabetical order:

Attribute Name	Attribute Type	Description
AbsolutePosition	Real-valued	The search result's position in all the search pages.
BehaviorType	Discrete-valued	Indicates whether the user has visited a search result or browsed away from it.
BehaviorUrlLength	Real-valued	The length of the URL on which the user performed some action (not necessarily the search result).
DescriptionLength	Real-valued	The length of the search result's description (specified as a meta tag in HTML).
DurationSeconds	Real-valued	How long the user spent on a page before performing an action.
ExitType	Discrete-valued	How the user left the search result.
FeedbackOption	Discrete-valued	The user's satisfaction with the search result.
FileSize	Real-valued	The length in bytes of the search result page.
ImageCount	Real-valued	The number of images linked into a search result.
LinkTextLength	Real-valued	The length of search result's title (specified in HTML).
Page	Real-valued	The search result page number.
PagePosition	Real-valued	The search result's position relative to the top of a search page.
ScriptLength	Real-valued	The length in bytes of all linked JavaScript files.
SearchResultUrlLength	Real-valued	The length of the search result URL.

Table 5-1 Data set attributes used for building classifiers.

Nearly all of the data was complete, meaning there were few instances with missing data. Complete data is a desirable property for training a classifier, because otherwise the classifier construction algorithm will have to infer the missing values. Some of the data entries did have missing values, however, and thus had to be treated before use in Weka. For example, in the event that the user closes the browser, the browser exit type value is unknown, but since this is the only case in which the value is unknown, by deduction it is known. These missing values are replaced with a token representing "closed browser". The remaining attributes that were missing data were environmental and corresponded to the length of any linked JavaScript files, the length of the HTML document, or the number of images embedded in the page. They were handled by the J48 (C4.5) algorithm as detailed in Section 2.2.3.

Some of the values appeared real-valued but were in fact discrete by nature. These attributes related to the user's behavior type and the user's submitted feedback value. In order to prevent the classifier from discretizing these data itself, we discretized the data during the ARFF file creation. Had the classifier discretized the data, it would have used a binning strategy that split the range of values into sub-intervals. The desired effect was to actually treat each integral value in the range as a value independent of any other in the range. As an example, our discretization process converts [1, 6] into {1, 2, 3, 4, 5, 6}, which is a set of discrete elements. Left on its own, Weka may convert that range to the bins{-∞ – 1.4, 1.5 – 2.9, 3.0 – 4.4, 4.5 – 5.9, 6.0 – ∞}.

5.1.3 Method

Weka ships with implementations of two of the most common decision tree construction algorithms: ID3 and C4.5 (although Weka calls its version J48), which are described in more detail in Chapter 2. For these experiments, we opted to use the J48 method because it performs better than ID3 in nearly all circumstances [Quinlan 1993]. Weka also allows configuration of certain properties of these algorithms that will affect the tree construction process. Using previous experience in building decision trees to guide our selection process, we performed experiments with various configurations in order to determine how the differences would affect classification accuracy. The results of these experiments can be seen in Figure 5-1:

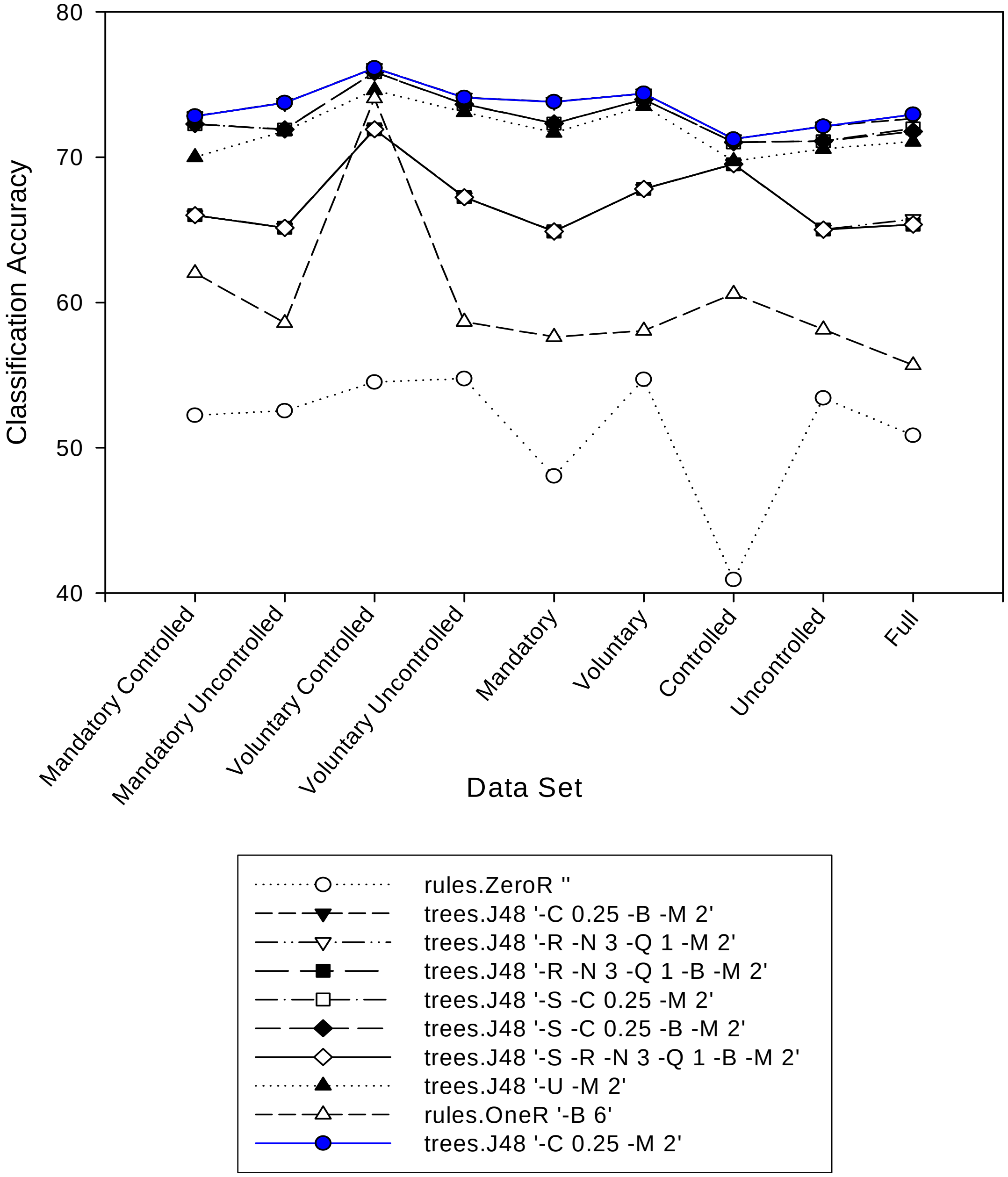


Figure 5-1 Comparison of classifier accuracies.

All experiments were performed using 10 fold cross-validation and the results averaged over 10 runs. The legend indicates the classifier used followed by the corresponding parameters suitable for use in Weka. For example, “trees.J48 ‘-C 0.25 -M 2’ means a J48 decision tree that uses a confidence factor of 0.25 for pruning and requires a minimum of 2 data instances per classification before creating a leaf node to represent that classification. The set of J48 parameters, as defined within Weka, can be viewed in Table 5-2:

Short Name	**Long Name**	**Description**
-B	binarySplits	Whether to use binary splits on nominal attributes when building the trees. *Its presence indicates binary splits are to be used.*
-C	confidenceFactor	The confidence factor used for pruning (smaller values incur more pruning).
-M	minNumObj	The minimum number of instances per leaf.
-U	unpruned	Whether pruning is performed. *Its presence indicates the decision tree is not to be pruned.*
-S	subtreeRaising	Whether to consider the subtree raising operation when pruning. *Its presence indicates subtree raising is not to occur.*
-R	reducedErrorPruning	Whether reduced-error pruning is used instead of C.4.5 pruning. *Its presence indicates that reduced-error pruning is to be used.*
-N	numFolds	Determines the amount of data used for reduced-error pruning. One fold is used for pruning, the rest for growing the tree.
-Q	Seed	The seed used for randomizing the data when reduced-error pruning is used.

Table 5-2 Weka parameters for J48 decision tree classifier. The names and descriptions come directly from the Weka in-program help system, with personal annotations appearing in italics.

The “rules.ZeroR” line represents the baseline operation for all classifiers. In ZeroR, the majority classification value observed during training is the only response the classifier ever predicts; there is no reasoning involved at all, it is simply a matter of target value distribution in the training set. For example, if during training it is observed that users are dissatisfied 60% of the time, then ZeroR will always predict *Dissatisfied* and be correct approximately 60% of the time. OneR is similar to ZeroR, but does consider a single attribute used to create a single rule prior to predicting. In fact, the one rule from OneR will be the same as the root of any decision tree, since decision trees use more general rules near the root and become more specific near the leaves. The remaining lines represent varying configurations of J48 decision trees. The blue line with the filled circle represents the Weka default for J48.

Across all datasets, the default Weka J48 classifier performs quite well. It clearly has higher classification accuracy than several other decision trees, but is marginally less than others on certain datasets. Using a corrected paired *t* test, it was found that the default J48 does not perform significantly worse than any other decision tree at the 0.05 confidence level. As such, and in order to not add unnecessary complication, the Weka default options are used for all decision trees hereafter.

There are factors to consider beyond classification accuracy, however. As an example, presuming that the data is consistent (meaning two different classifications cannot be derived from the same sequence of attribute values), a tree can be built that describes each datum with a single rule. The classification accuracy of this tree would be 100%, but the tree would be so specialized as to be useless in the general case. Normally it is said such a tree is *overfit* to the data. There would be little insight gained as to what features have a large impact on user satisfaction by looking at such a tree. Generally speaking, Occam’s razor[3] rules in such cases and simpler trees are probably more accurate predictors of the underlying phenomena [Russell & Norvig 2003]. Thus, after having chosen our set of parameters for the J48 algorithm, we investigated how we could prune branches in an attempt to maximize classification accuracy while minimizing the overall tree size. Weka’s J48 implementation allows the experimenter to control pruning by setting a confidence factor from [0, 1] for the classifier. A lower confidence value leads to more aggressive pruning of tree nodes.

In order to determine the best confidence factor to use, we conducted a series of experiments that compared J48 classifiers built with different confidence factors. We chose six different values ranging from 0.05 to 0.30, at 0.05 intervals, and observed the differences between them in terms of classification accuracy, number of generated rules (total number of leaves), and total tree size (defined as total number of internal nodes + total number of leaves). Statistical significance in differences between them was tested using a corrected paired *t* test for each data set, with the Weka default J48 classifier (C = 0.25) as the point of comparison.

[3] Occam's razor is a philosophy that dictates the simplest solution is the most correct. It is employed in nearly all scientific fields (it is common to derive simple models to build on top of) and Mitchell [1997] argues that it is also applicable to machine learning.

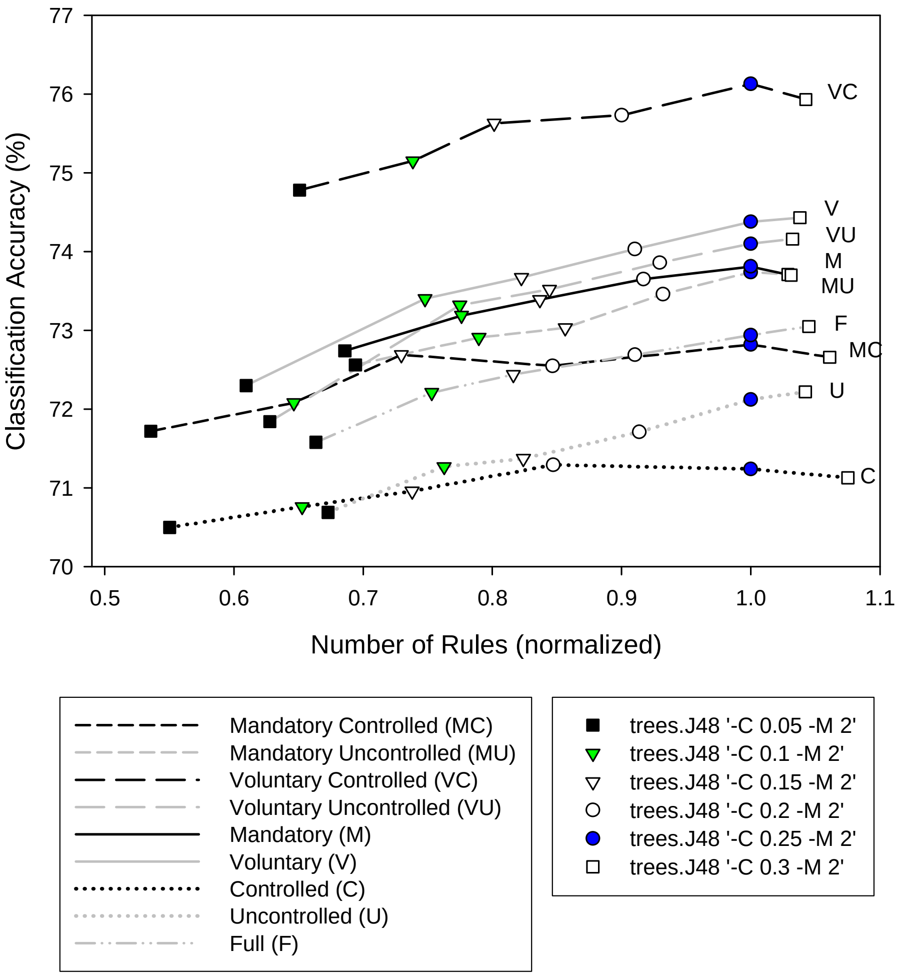


Figure 5-2 Classification accuracy versus number of rules for various J48 confidence factors.

The results of these confidence factor experiments can be seen in Figure 5-2. The graph depicts the classification accuracy and normalized number of rules for each of the six different confidence factors over the nine data sets. The number of rules is normalized by the Weka default J48 classifier. Thus, the Weka default classifier has a normalized rule count of 1.0 on the graph and all other classifier values are relative to this baseline. Using normalized values rather than raw count allows for easy comparison of classifiers in terms of rule reduction across all data sets.

The goal of these confidence factor experiments was to find a point on the graph that significantly reduces the number of rules while not significantly reducing the classification accuracy. The solid, blue circle corresponds to the Weka default J48 classifier. As the confidence factor decreases, the number of rules also decreases, as expected. Likewise, increasing the confidence factor increases the number of rules, as evidenced by the open square points on the graph, corresponding to C = 0.30.

As can be seen in the graph, the confidence factor values generally have a linear relationship with the classification accuracy and number of generated rules. Several of the lines, however, have an initial period of steep ascent and then grow slowly. Up until the change in slope, there is a significant difference between classification accuracies in the classifiers represented by the steeply rising line segment and the Weka default. More importantly, at these junctions, there is no longer a significant difference in classification accuracy between the Weka default classifier and the other classifiers plotted in the connecting line segment. In general, the solid, green triangles mark these junction points and correspond to a confidence factor of 0.10. Most notably, the *mandatory controlled* and *voluntary controlled* classifiers do not conform to this general trend and a confidence factor of 0.15 may have been a better choice for their junction points. We chose the confidence factor of 0.10 to be our best confidence value since it minimizes the number of generated rules while not significantly decreasing classification accuracy over the majority of data sets.

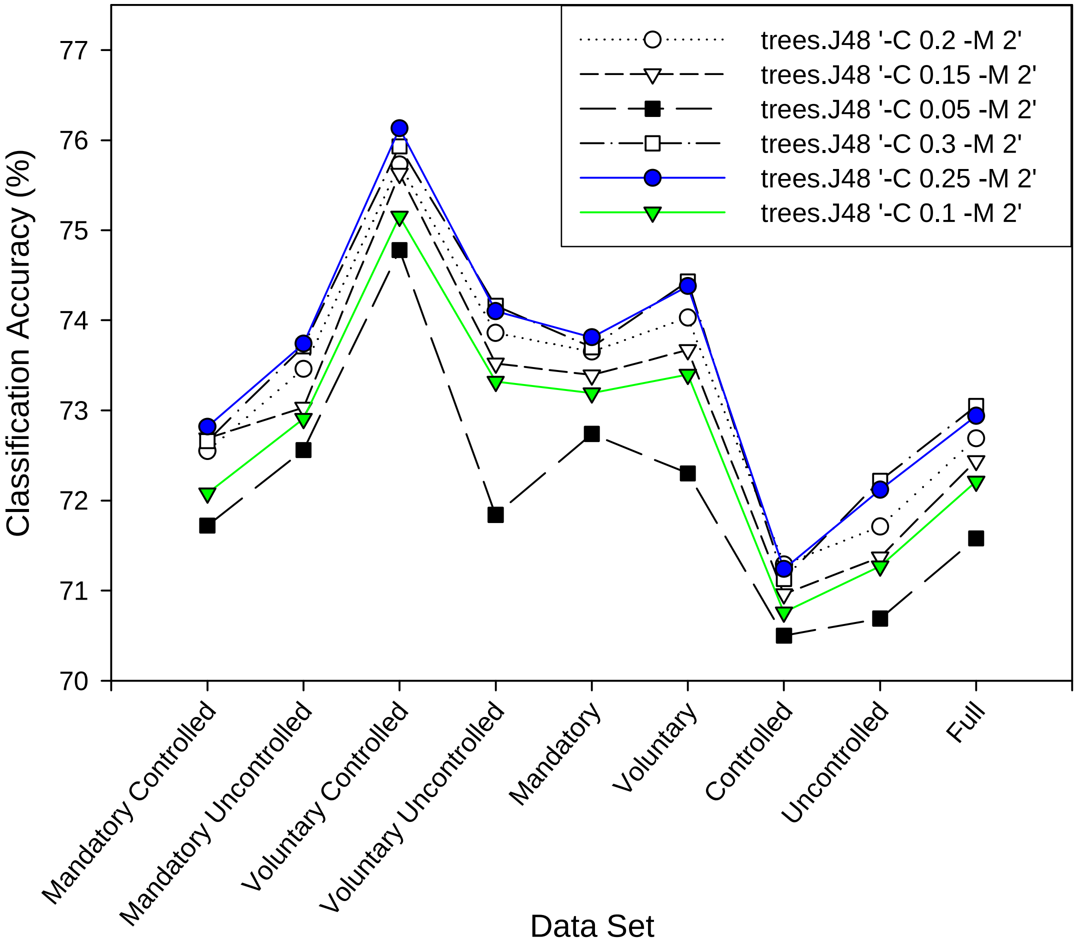


Figure 5-3 Comparison of accuracies over different pruning levels.

Figure 5-3 is an alternative representation of information from Figure 5-2, highlighting the differences in the classifiers in terms of classification accuracy over different subsets of our data. In this figure and those that follow, the solid, blue line with the filled circle indicates the baseline as the Weka default. The solid, green line with the filled triangle represents the confidence factor that maximizes the balance between classification accuracy and number of rules (C = 0.10). It should be noted that of the six different confidence factors tested, only one was significantly different from the Weka default; this classifier is represented on the graph by the dashed line with the filled, black square (C = 0.05).

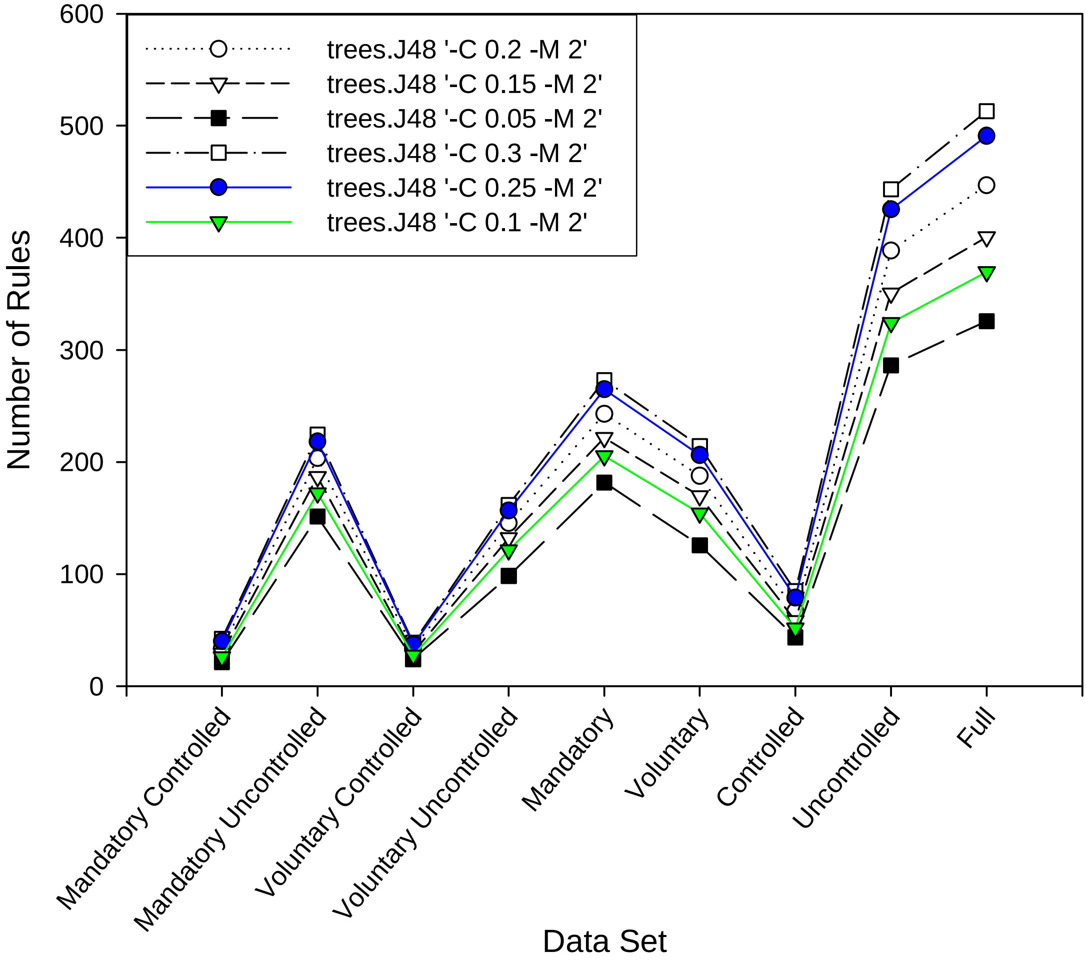


Figure 5-4 Comparison of number of generated rules over different pruning levels.

Figure 5-2 shows that the trees generated with C = 0.10 are between 21% and 35% smaller than those generated with the Weka default of C = 0.25, while reducing classification accuracy by only 1 – 1.5 percentage points. Figure 5-4 shows how these percentages translate into raw values. Of the confidence factors tested, only one (C = 0.05) produced a fewer number of rules than our chosen optimal classifier but was disregarded due to having a significantly worse classification accuracy. As mentioned previously, a smaller number of rules is usually more applicable to the general problem domain. Pragmatically speaking, a rule reduction also makes it easier for humans to comprehend.

Smaller tree sizes equate to more efficient training in terms of both time and space. Figure 5-5 illustrates how the various confidence factors affect overall tree size. This information is not directly represented in Figure 5-2, but as the tree size is defined in terms of the number of internal nodes combined with the number of leaf nodes (the number of generated rules), it can be inferred. As with the number of generated rules, our chosen classifier reduced the tree size by 21% - 36% relative to the Weka default, which can significantly reduce training time over a large number of data instances. The remaining classification experiments, thus, use the Weka default for the tree generation algorithm with a pruning confidence value of 0.10.

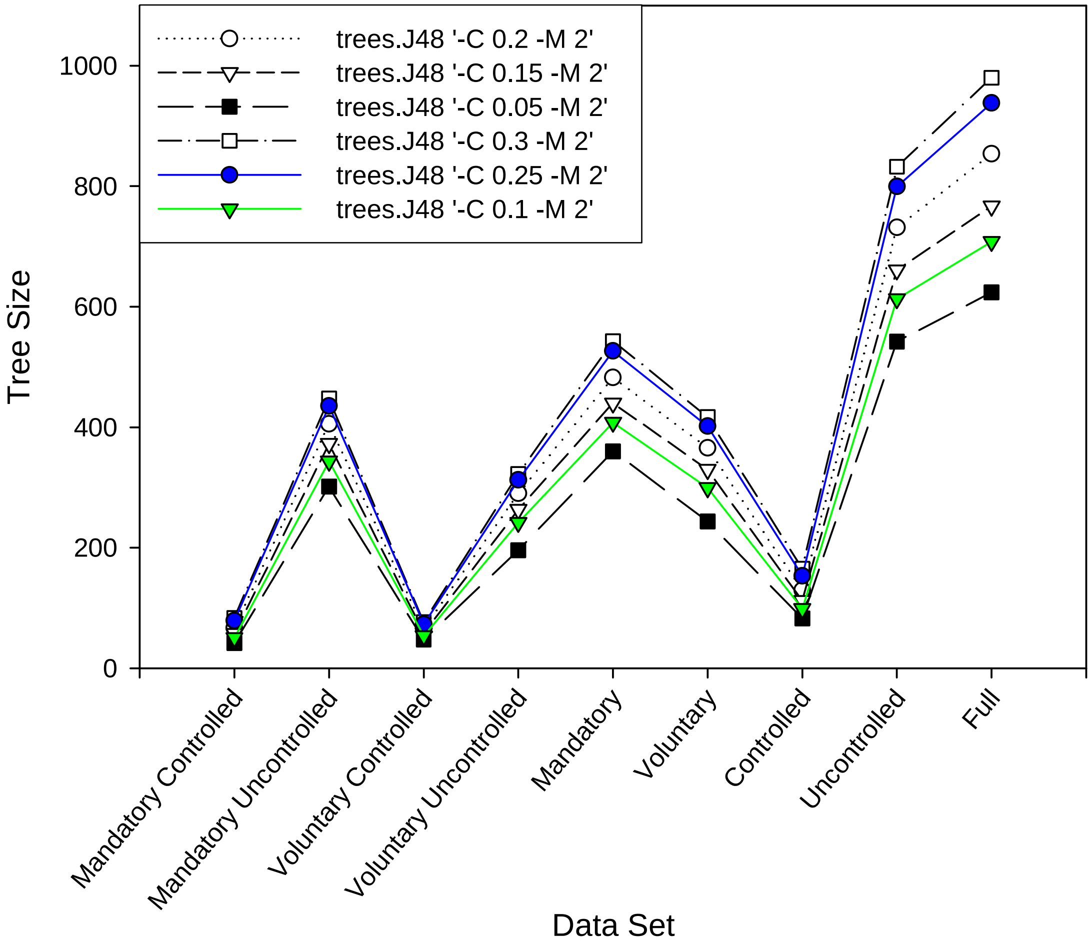


Figure 5-5 Comparison of tree sizes over different pruning levels.

5.2 Results

The decision trees for the four Mandorvol study experiment types, built using the J48 classifier with a confidence factor of 0.10 in Section 5.1.3 are summarized in Table 5-3:

Mandatory Controlled		**Mandatory Uncontrolled**	
Data Collected:	362 (20 users)	Data Collected:	2050 (37 users)
# of Rules:	28	# of Rules:	168
Tree Size:	55	Tree Size:	329
Accuracy (%):	67	Accuracy (%):	67
Std. Dev. (%):	8.2	Std. Dev. (%):	3.5
Voluntary Controlled		**Voluntary Uncontrolled**	
Data Collected:	398 (29 users)	Data Collected:	1348 (31 users)
# of Rules:	32	# of Rules:	114
Tree Size:	61	Tree Size:	221
Accuracy (%):	74	Accuracy (%):	70
Std. Dev. (%):	6.9	Std. Dev. (%):	4.1

Table 5-3 Classifier properties by experiment type.

Note that the number of users that contributed data to the classifiers and the number of users reported to have completed the study (Section 4.3.1) are not the same. While the figures in Section 4.3.1 do accurately represent the number of users that participated in the study, it was not discovered until detailed inspection of the data that some users did not in fact complete the study. In most of these cases, the Mandorvol Browser was turned off after the start of the experiment, either voluntarily or inadvertently. Additionally, in the cases of the voluntary experiments, some users simply opted not to provide feedback. In these cases, the subject did complete the study but did not contribute any data that could be used for classifier training. Henceforth, only participants that completed the experiment are considered in our analysis.

Figure 5-6 shows the total number of study participants by day. The large slope beginning at day 7 is the result of our first marketing effort. After approximately five days, the rate of user participation slowed down. On day 17, the number of new participants spiked up briefly again. From that point to the end of the experiment, the number of new users participating each day was mostly constant, at approximately one or two users per day.

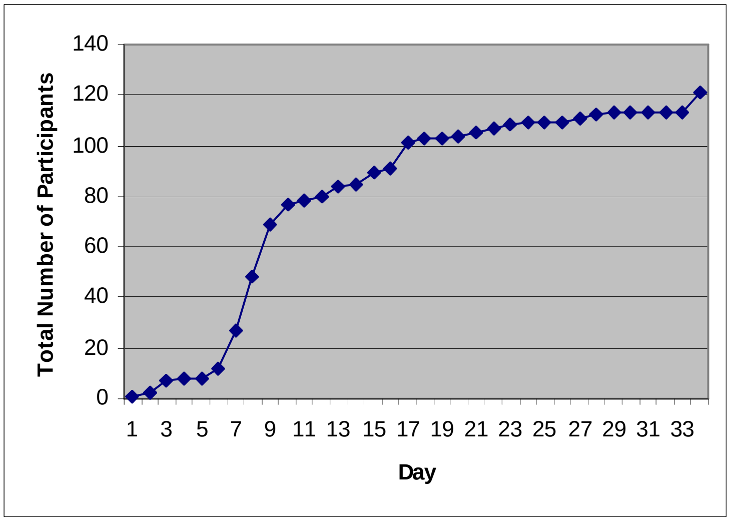


Figure 5-6 Total number of participants by day.

As the data was collected over the course of 38 days, we decided to investigate how well the classifier performed as new data was added (see Figures 5-7 to 5-10). The choice of a "day" as a line of demarcation is arbitrary, since the collected data was not evenly distributed over all days. However, it was still used because it provided a natural boundary and is simple to reason about. The data used for each of the daily classifiers is accumulative. For example, on day three, the data collected on days one, two, and three are used. Note that the graphs only show data through day 33 because no useful information was collected during the last five days of the experiment. We averaged 3.5 new subjects per day that contributed 126 data instances per day for those 33 days.

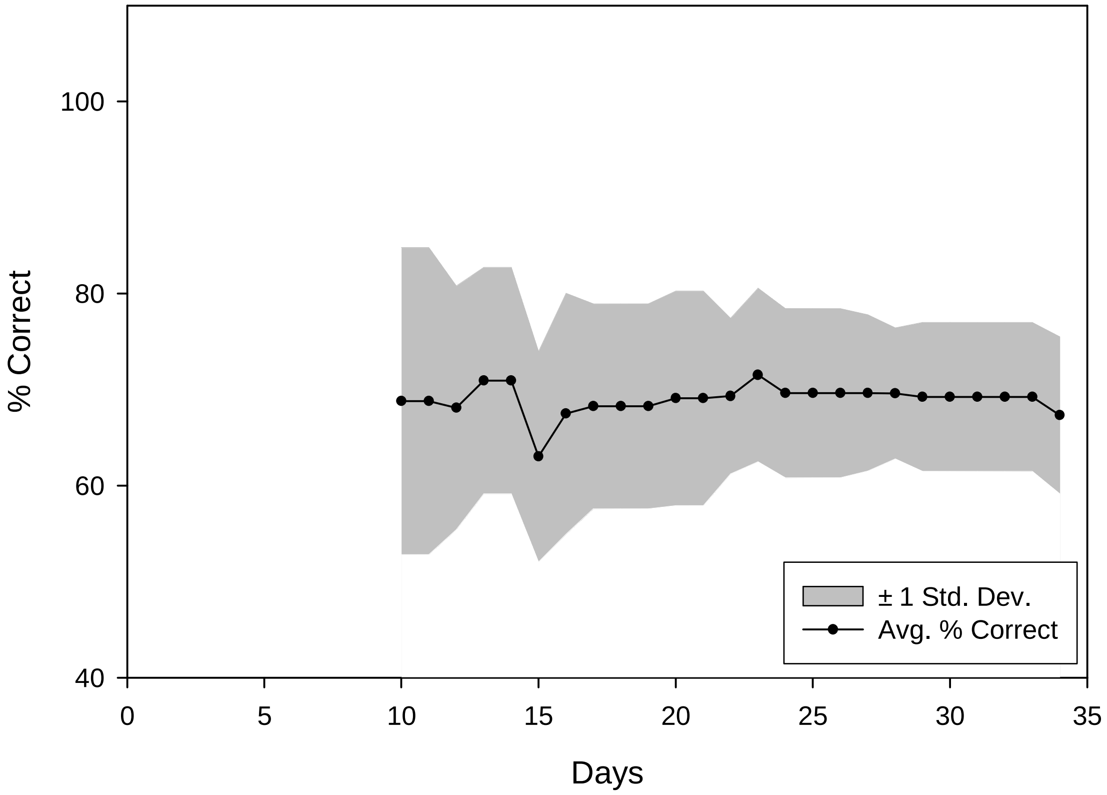


Figure 5-7 Daily classification accuracy for *mandatory controlled* experiment.

Figure 5-7 shows the daily performance of the classifiers for the *mandatory controlled* experiment. Recall that all experiments were run 10 times with 10 fold cross-validation. In order for a 10 fold cross-validation to work, there must be at least ten data entries, so that all the folds have some data. As can be seen in Figure 5-7, the *mandatory controlled* experiments did not garner a sufficient amount of data until the tenth day of the study. This was due to the software defect that affected the experiment, described in Section 4.3.1.

Approximately 24 days into the experiment, the classification results began to stabilize. The most turbulent areas in the graph correspond directly to the large growth in population shown in Figure 5-6. While the classification accuracy does not change very much, the standard deviation in accuracy between different classifiers is both the 10 fold cross-validation and across the 10 experiment runs continues to decrease.

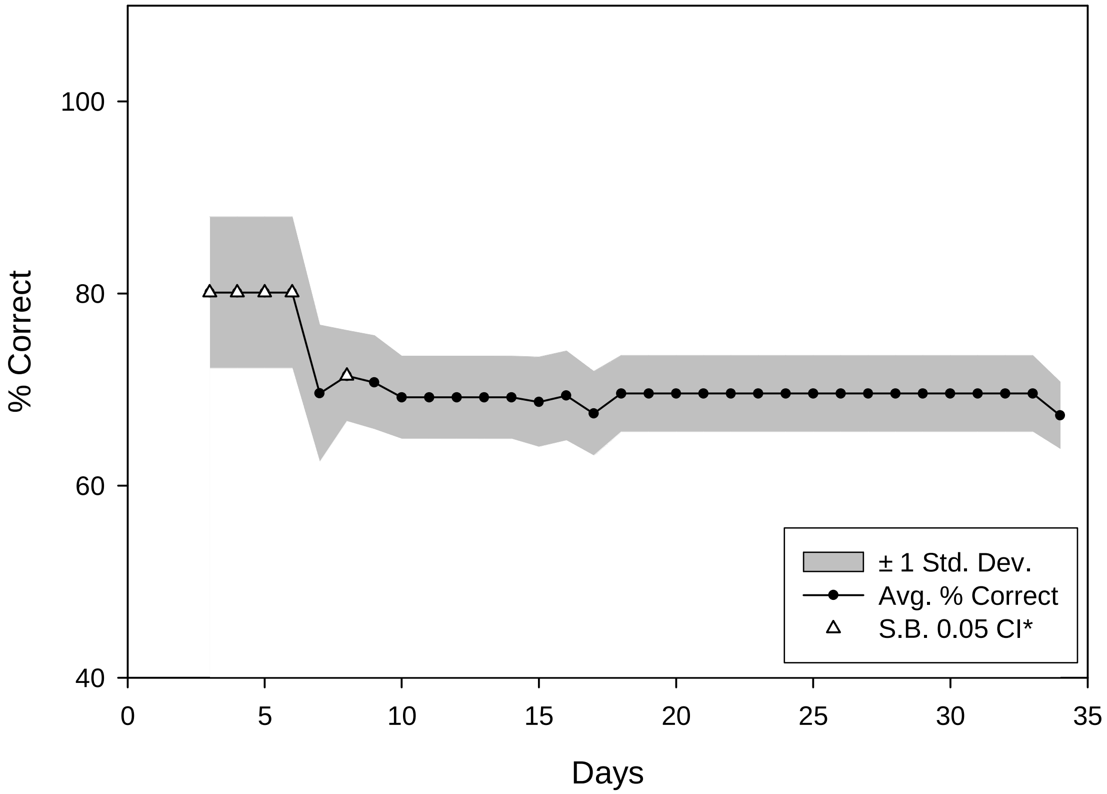


Figure 5-8 Daily classification accuracy for *mandatory uncontrolled* experiment.

The triangular points in Figure 5-8 indicate classification accuracies that are significantly better than the baseline, which was the last day of the experiments, at the 0.05 confidence interval. The last day of the experiment, day 33, was chosen at the baseline because at that point, all collected data would be used in the decision tree construction. The underlying heuristic is that the more data the decision tree has, the better the classifier. Each daily classifier was compared to the last day using a corrected paired *t* test.

While the triangular points have significantly better classification accuracies, the standard deviations at those points are considerably larger than the baseline. It is not sufficient to merely pick the subset of data that yields the highest accuracy. The decision tree construction must be reliable and high standard deviation indicates that there was substantial variation in the generated trees for each fold and each experiment. As can be seen in Figure 5-8, the classification results did not begin to stabilize until approximately day ten, with all of the significantly better classification accuracies occurring before that point.

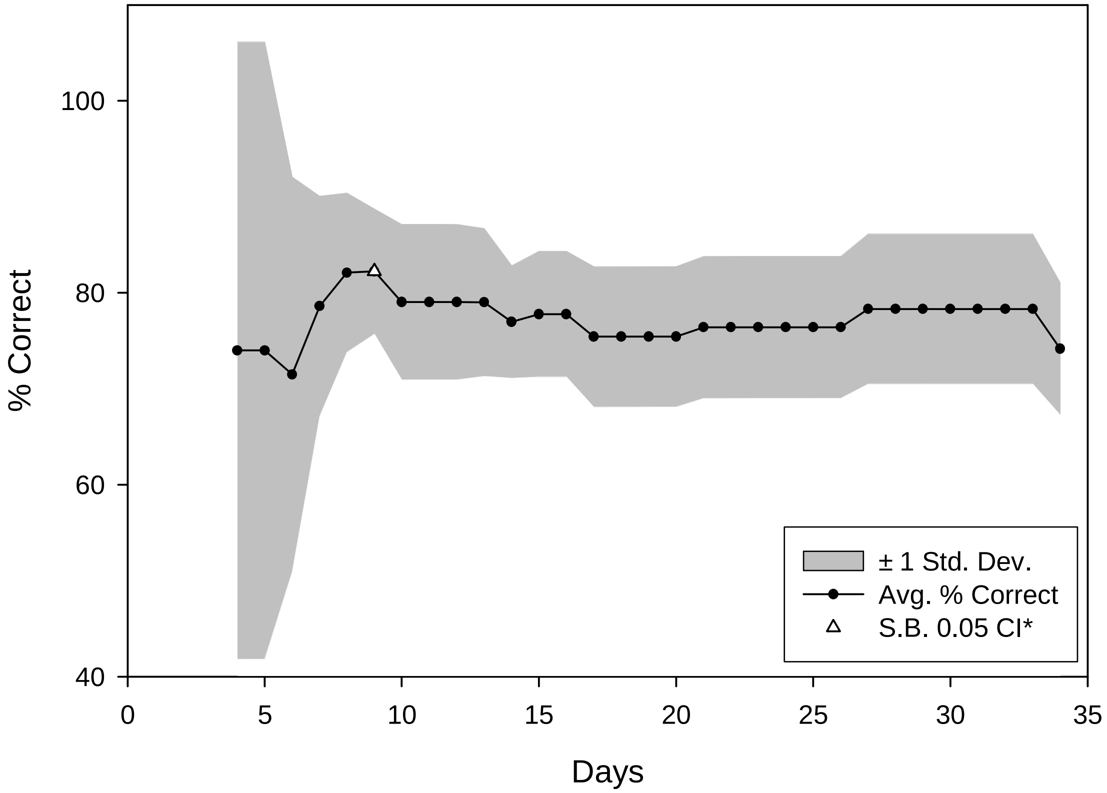


Figure 5-9 Daily classification accuracy for *voluntary controlled* experiment.

As more diverse data is added, the J48 algorithm is able to detect relationships between attributes in the data set. Using attribute values in a datum, rather than just the classification value, allows the decision tree to reason about a classification prediction rather than simply reporting the training set's classification distribution. In doing so, the folds are not as dissimilar and as such the overall variance is reduced.

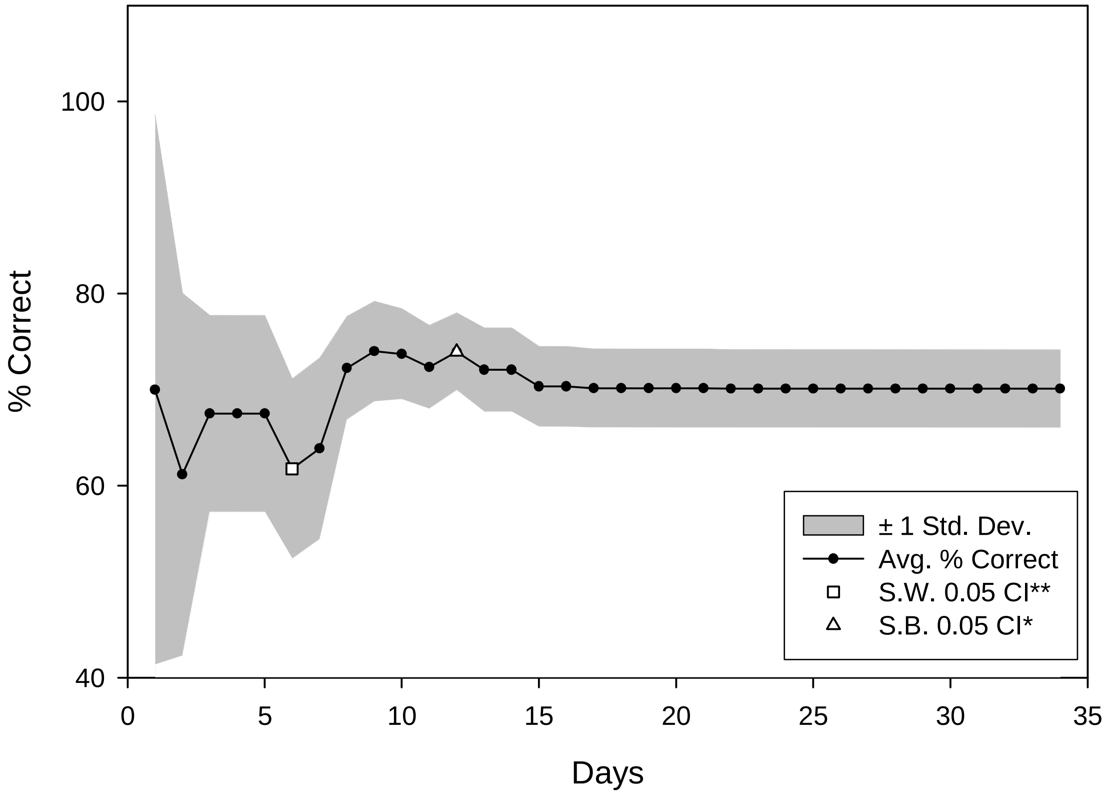


Figure 5-10 Daily classification accuracy for *voluntary uncontrolled* experiments.

Figure 5-10 contains a point, indicated by a square, that is significantly worse than the baseline classifier. Like points that are significantly better, this point is deemed significantly worse at the 0.05 confidence interval using a corrected paired *t* test. Fortunately, this point occurs very early in the experiment and as more data is collected, no significantly bad classifiers are created. In fact, this is true of classifiers generated for all experiment types.

5.2.1 Results Summary

The initial standard deviation for all of the data sets is high. Early in the experiments, little data were available and as such, the constructed trees were based more on probable data distribution than discovered relationships. For example, in the *voluntary controlled* set of experiments (Figure 5-9), users tended to give feedback when they were satisfied with a search result (see Table 4-8). If early on in the experiments the majority classification value was *Satisfied*, the trees would resemble

ZeroR in that they would simply predict *Satisfied* in all cases. If some of the data contained other classifications values, however, the results of an n-fold cross-validation would fluctuate between 0% and 100%. The average of these runs would be the actual distribution of *Satisfied* values.

At approximately day 15, all classifiers begin to stabilize in their classification accuracies. This suggests that prolonged data collection will not significantly improve the results of any constructed decision tree. However, the standard deviations do continue to decrease as more data is added, suggesting a convergence to a single tree that will be created by all folds in all experiments.

The standard deviations are lower for the uncontrolled experiments than for the controlled using both mandatory and voluntary feedback mechanisms, as can be seen by comparing Figure 5-7 with Figure 5-8 and Figure 5-9 with Figure 5-10. The differences in standard deviations correlate to the difference in the number of data instances and, as such, were expected. Table 5-3 shows an average standard deviation of 3.8% for the uncontrolled experiments and an average standard deviation of 7.6% for the controlled experiments. A natural consequence of these differences is that we have much greater certainty in any conclusions we derive regarding the uncontrolled experiments rather than the controlled ones.

5.3 Analysis

Using the user satisfaction values distribution from Table 4-8 and the J48 decision tree properties shown in Table 5-3, it can be seen how well our trained classifiers predicted user satisfaction compared to the baseline operation. Table 5-4 reports both the baseline classification accuracy and the J48 classification accuracies for each of the four Mandorvol study experiment types. The table also shows the increase in accuracy obtained by the decision tree over the ZeroR method.

	ZeroR (baseline) Accuracy (%)	Decision Tree Accuracy (%)	Difference (%)
Mandatory Controlled	47	67	+44
Mandatory Uncontrolled	47	67	+44
Voluntary Controlled	51	74	+46
Voluntary Uncontrolled	49	70	+42

Table 5-4 Comparison of baseline and decision tree classification accuracies.

The decision tree classifiers were able to predict user satisfaction values much more accurately than did ZeroR. As can be seen in Table 5-4, in all cases, the decision tree classification accuracy is at least 20% higher than that of ZeroR, corresponding to an increase of 40% or more in classification accuracy. Furthermore, the classification accuracies are high enough to be useful. As an example, in a *voluntary controlled* scenario the decision tree is able to correctly predict how satisfied a user is with a search result three out of four times.

5.3.1 Mandatory Versus Voluntary

The mandatory dimension is the pivot point of this experiment. Previous work [Fox et al. 2005] has shown that classifiers for predicting user satisfaction can be built using data collected from a mandatory feedback mechanism. The rationale behind the choice of using a mandatory feedback mechanism is to maximize the amount of collected data. Table 5-3 clearly shows that the mandatory experiments collected a greater amount of feedback data than the voluntary experiments, supporting our hypothesis **H1** that a mandatory feedback mechanism would collect more data than a voluntary one. Classifiers such as decision trees typically become increasingly accurate as more data is supplied for training. Thus, it is noteworthy that the voluntary dimension has higher classification accuracy in both the controlled and uncontrolled scenarios, as seen in Table 5-4.

Hypothesis **H2** stated that data collected via a voluntary feedback mechanism would be of higher quality than that collected via a mandatory feedback mechanism. Here, quality is defined as most accurately representing a user’s true level of satisfaction with a search result. The underlying idea is that if a user gives feedback of their own free will, then they are likely to give correct data. If, however, users are forced to give feedback values, they may give incorrect results either as simply a means of removing the pop-up or as a form of retribution for being annoyed. While it is not clear that the voluntary dimension is of higher quality due to these factors, it is nonetheless better than the mandatory dimension in terms of classification accuracy.

Due to the stabilization of classifier accuracies shown in the daily classifier graphs in Figure 5-9 & Figure 5-10, it is clear that the lower amount of data collected via a voluntary feedback mechanism does not impact the results of constructed decision trees. Thus, choosing a mandatory feedback mechanism simply as a means of collecting more data is not a justified decision, unless it is believed that not enough data can be acquired before reaching the critical point (approximately day 15 – 1,200 instances – in this experiment) at which the addition of more data will not significantly affect classification accuracy. A classifier built using a voluntary feedback mechanism can perform just as well, if not better, than a classifier built using a mandatory feedback mechanism without adversely affecting a user's search session.

As mentioned previously, there are a large number of similarities between the work completed by Fox et al. [2005] and our own study. Sections 3.2 & 4.1.2 describe how our experimental design was derived from their work, but does not discuss analogues between their results and ours. By comparing the results of the Fox et al. study with our own, we further enhance both their findings and our own.

The Fox et al. [2005] experiment was conducted over a six week period with 146 participants, yielding approximately 3,700 different data instances. These values are very similar to our own results, as highlighted in Table 5-3. Using their collected data, Fox et al. were able to construct classifiers that were able to correctly predict user satisfaction with a search engine result 57% of the time. By removing problematic leaf nodes in their decision tree, they were able to improve the accuracy to 66%. The improved classification accuracy is very similar to those observed in our *mandatory controlled* and *mandatory uncontrolled* experiments. The Fox et al. experiment most nearly correlates to our *mandatory uncontrolled* experiment type.

If it is to be believed that the results of our mandatory set of experiments are analogous to those of Fox et al., then there is further evidence that data collected via a voluntary feedback mechanism is able to yield better classifiers than data collected via a mandatory feedback mechanism. Unfortunately, despite the similarities between the two studies, it is hard to provide a direct comparison between our work and that of Fox et al.. In particular, their feedback distributions were different than those we observed, affecting their baseline. Additionally, they used a time-based method for splitting their data set into training and test sets, which may impact the classification results.

5.3.2 Controlled Versus Uncontrolled

The differences in classifiers over controlled and uncontrolled scenarios are not nearly as pronounced as the differences between the mandatory and voluntary dimensions. In fact, looking at the mandatory dimension, the classification accuracies between controlled and uncontrolled are virtually identical. In the voluntary dimension, however, the difference is quite large. It is not clear why this is the case, and determining the correlation between voluntary feedback systems and the scope of a search domain extended beyond the bounds of this project. An in-depth study of this relationship may yield additional insights into improving the constructed decision trees.

5.3.3 Implicit Indicators

Treating our decision trees as a set of rules shows that certain attributes in the data set consistently have higher information gain. The `PagePosition` and `DurationSeconds` attributes appeared in all four experiment types. A notable observation is that for the two controlled experiments, the `LinkTextLength` attribute had high information gain, perhaps indicating that users searching for help prefer links with short, descriptive titles. No such trend was observed in the uncontrolled experiments, although the `Page` and `SearchResultUrlLength` were used much more frequently in this set of experiments. Fox et al. observed that `DurationSeconds` and `ExitType` were highly predictive attribute in their classifiers.

Clearly, the user dwell time on a page (`DurationSeconds`) is an implicit indicator highly correlated with user satisfaction. There is a difference between the utility of other implicit indicators we recorded and those that Fox et al. noted. For example, `ExitType` did not have high information gain in our classifiers, while it was highly predictive for Fox et al. It is not immediately clear why this is the case. It could simply be due to different user populations or changing trends in how users use Web browsers (there is a four year time gap between when the Fox et al. study was completed and when our study was completed).

6 Conclusions

This chapter provides a summary of the results of this research. Here we discuss what can be concluded from the research in this thesis and the primary contributions. Additionally, we offer ideas for future research that may enhance our findings.

6.1 Research Question Revisited

Previous work has found correlations between user behaviors collected during a user's interaction with a Web browser and a user's level of satisfaction with the Web page [Claypool et al. 2001a; Claypool et al. 2001b; Cen et al. 2002]. As these behaviors are indications of user satisfaction, they have been termed *implicit indicators*. Fox et al. [2005] constructed a classifier using feedback values collected with a mandatory feedback mechanism that was able to predict user satisfaction with a search engine result as a function of implicit indicator values. Using the results of Fox et al., search engines can improve their results by incorporating an implicit human rating of document relevance into their ranking process.

While the results of Fox et al. are promising, we believed that their choice of using a mandatory feedback mechanism may have caused bias in their results. Furthermore, we believe that in a deployed system, a voluntary feedback mechanism will be much more user-friendly than a mandatory feedback mechanism. Thus, we set out to investigate the following research question:

> *Can voluntary data can be used to train a classifier that is as effective as a classifier trained with mandatory data.*

In order to answer the research question, we developed two hypotheses. Hypothesis **H1** is that a mandatory feedback method will collect higher quantities of data than a voluntary feedback method. Hypothesis **H2** is that a voluntary feedback method will collect higher quality data than a mandatory method. Each of these hypotheses addresses an important property of data used in the construction of classifiers. Principally, both high quantity and high quality positively contribute to the classification accuracy of a classifier.

Having derived our hypotheses, we began designing our study. We decided to test voluntary and mandatory feedback mechanisms in controlled and uncontrolled scenarios. The voluntary feedback mechanism was a vertical explorer band while the mandatory feedback mechanism was a modal pop-up window that could not be closed without providing feedback. The controlled scenario required a subject to complete a set of tasks in Microsoft Excel while searching for help only from the Microsoft Office help system. The uncontrolled scenario allowed subjects to search the Web using the Google Web search engine. We conducted two pilot tests in order to refine our choice of a voluntary feedback mechanism as well as the Excel tasks.

During the pilot studies, we found that a distinctive feedback mechanism will yield more feedback than one that blends in with the rest of the containing application (in this case, the Internet Explorer Web browser). Furthermore, we found that a vertical explorer band placed on the left-hand-side of a Web browser will elicit more feedback than a horizontal one placed at the bottom of a Web browser due to Western reading direction. If a user does not read a Web page completely, they may never see a feedback component placed at the bottom of the Web browser. To our dismay, we also discovered that the choice of a Web browser as the feedback tool introduced challenges related to pervasive Web content. In particular, we were required to refine our voluntary feedback mechanism so that it would not be confused with banner advertisements on Web pages, lest users would ignore it.

Integrating the results of the pilot studies into our experimental design, we commenced a two-month long study consisting of 161 users divided into four experimental groups. We analyzed the data we collected through the construction of decision tree classifiers using the open-source Weka machine learning tool. During our analysis, we were able to address our two hypotheses and showed evidence that supports both of them.

Using the data collected during the study, we began constructing classifiers to address the research question. We processed the data into 14 key attributes that were used to train a set of decision tree classifiers using the open-source Weka machine learning tool. We performed a series of experiments with different classifier configurations in order to find an optimal parameter set that balanced decision tree accuracy against the number of rules generated by the tree. Having found an optimal set of parameters, we constructed and analyzed decision trees for each of the four experimental groups: *mandatory controlled, mandatory uncontrolled*, *voluntary controlled*, and *voluntary uncontrolled*.

We found that in the controlled scenario, users tended not to provide feedback since they were task-oriented. In particular, we found that in order to give feedback, users would be required to evaluate a page twice. The first evaluation would be an attempt to apply the Web page's contents to the task at hand. If the page was helpful, typically users clicked the Web browser's "back" button in order to move onto the next task. In order to properly give feedback, users would have to evaluate the page again. Providing feedback about the Web page was not part of the users' workflow, and thus was often not completed. In the uncontrolled scenario, however, we found users were much more relaxed and did tend to provide feedback.

There was no significant difference between classifiers constructed with data from the *mandatory controlled* and *mandatory uncontrolled* experiments. There was a significant difference observed between classifiers built with the *voluntary controlled* and *voluntary uncontrolled* data. We were not able to deduce exactly what caused this difference, unfortunately. Determining the correlation between voluntary feedback mechanisms and the scope of the search domain extended beyond the bounds of our work, but may be worthwhile for future research.

Based on the results of the analysis, we have shown that not only can a classifier be built with data collected via a voluntary feedback mechanism, but such a classifier performs as well as, if not better than, one created with data collected via a mandatory feedback mechanism. Such a classifier can accurately predict user satisfaction approximately 70% of the time in an uncontrolled scenario and approximately 75% of the time in a controlled scenario. Additionally, through daily analysis of the data, we found that the increased quantity of data collected with a mandatory feedback mechanism does not eclipse the higher quality data of a voluntary feedback mechanism. We found that after about 15 days of data collection, providing more data to the decision tree construction algorithm did not affect the results of the constructed trees.

Our findings indicate that a search engine provider could integrate predicted user satisfaction values, based on a classifier trained using data collected via a voluntary feedback mechanism, into its search result ranking process. Using a voluntary feedback mechanism is more practical than a mandatory mechanism due to the user annoyance factor. Forcing users to provide feedback while reviewing a search engine result is likely to cause them to cease using the system. A passive, voluntary feedback mechanism will be more acceptable to users and will yield a more accurate classifier.

6.2 Suggestions for Future Work

In this thesis, we considered the utility of a voluntary feedback mechanism versus a mandatory feedback mechanism. Our choice of a side panel as our feedback component, while based on pilot studies, was almost arbitrary. Given a Voluntary – Mandatory continuum, one can imagine other types of feedback mechanisms that lay at different points on the scale. For example, a pop-up window that was non-modal and that could be closed without giving feedback would appear somewhere near the middle. Such a feedback mechanism may collect more feedback than our explorer band did at the cost of lowering data quality due to user annoyance. It would be interesting to see how different feedback mechanisms with varying degrees of voluntariness perform against each other. Such work can be viewed as a refinement of our research.

Furthermore, the decision to use search engines as our problem domain and a Web browser as our experimentation tool impacted the design and execution of our experiment. As discussed in Section 3.4, people have become adept at filtering out non-core content in a Web browser due to the pervasive nature of online advertising. We believe that had the experiment been performed within the context of a domain-specific application (e.g., a spreadsheet application) help system, users would have been more responsive to prompts for feedback. In such applications, there is more flexibility in the design of the feedback mechanism, since it will likely not be confused with a banner advertisement. Additionally, it may be the case that the quantity and quality of data collected will differ considerably from that observed in the Web search space. In these cases, our work cannot be directly applied, but rather can serve as a framework for similar studies.

Future work that focuses on the study population may be able to discover new relationships between the user and the quality and quantity of data collected. While we attempted to attract as large and as diversified a population as we could, our sample population consisted mostly of undergraduate computer science and electrical/computer engineering students. These individuals have higher than average computer skills and thus may have skewed our results. Unfortunately, this was a limiting factor in the work of Fox et al. [2005] as well. A future study that can test either a less technically-inclined population or simply a more diversified one may yield different results.

Finally, future work into the discovery of implicit indicators can also serve to enhance our findings. In particular, implicit indicators that highly correlate with user satisfaction may be able to improve the various decision trees we constructed. Such implicit indicators would thus lead to better predictions of user satisfaction, which may be used to further refine search engine results.

7 References

Adamczyk, P. D. and Bailey, B. P. (2004). If not now, when?: the effects of interruption at different moments within task execution. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vienna, Austria, April 24 - 29, 2004). CHI '04. ACM Press, New York, NY, 271-278.

Bailey, B. P., Konstan, J. A. & Carlis, J. V. (2000). The effect of interruptions on task performance in the user interface. In *IEEE International Conference on Systems, Man, and Cybernetics* (Nashville, TN, USA, Oct. 8 - 11, 2000) 2, 757-762.

Baralis, E. & Chiusano, S. (2004). Essential classification rule sets. *ACM Trans. Database Syst.* 29, 4 (Dec. 2004), 635-674.

Cen, M., Goodwin, B., & Law, S. T. (2002). Curious browser. WPI MQP:DCB-0106.

Claypool, M., Brown, D., Le, P. & Waseda, M. (2001a). Inferring user interest. *IEEE Internet Computing* (November/December 2001).

Claypool, M., Le, P., Waseda, M., & Brown, D. (2001b). Implicit interest indicators. In *Proceedings of ACM Intelligent User Interfaces Conference (IUI)* (Santa Fe, New Mexico, January 2001).

Dalvi, N., Domingos, P., Mausam, Sanghai, S., & Verma, D. (2004). Adversarial classification. In Proceedings of the Tenth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining (Seattle, WA, USA, August 22 - 25, 2004). KDD '04. ACM Press, New York, NY, 99-108.

Fayyad, U. M. (1991). *On the Induction of Decision Trees for Multiple Concept Learning*, (Ph.D. dissertation). EECS Department: University of Michigan.

Fayyad, U. M., & Irani, K. B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, 1022-1027. Amherst: Morgan Kaufmann.

Fox, S., Finger, J., & Taylor, T. (2003). *Curious Browser Data Acquisition for Content Watson*. Internal MS Document.

Fox, S., Karnawat, K., Mydland, M., Dumais, S., & White, T. (2005). Evaluating implicit measures to improve web search. ACM Trans. Inf. Syst. 23, 2 (Apr. 2005), 147-168.

Hijikata, Y. (2004). Implicit user profiling for on demand relevance feedback. In Proceedings of the 9th international Conference on intelligent User interface (Funchal, Madeira, Portugal, January 13 - 16, 2004). IUI '04. ACM Press, New York, NY, 198-205.

Ivory, M. Y., Sinha, R. R., & Hearst, M. A. (2001). Empirically validated web page design metrics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, United States). CHI '01. ACM Press, New York, NY, 53-60.

Kvavik, K. H., Karimi, S., Cypher, A., & Mayhew, D. J. (1994). User-centered processes and evaluation in product development. *interactions* 1, 3 (Jul. 1994), 65-71.

Lane, K. M. and Neidinger, R. D. (1995). Neural networks from idea to implementation. *SIGAPL APL Quote Quad* 25, 3 (Mar. 1995), 27-37.

López-Ortiz, A. (2005). Algorithmic foundations of the internet. SIGACT News 36, 2 (Jun. 2005), 45-62.

Mitchell, T. (1997). *Machine Learning*. New York: McGraw-Hill.

Moret, B. M. (1982). Decision Trees and Diagrams. *ACM Comput. Surv.* 14, 4 (Dec. 1982), 593-623.

Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*. New York: Cambridge University Press.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81-106.

Quinlan, J. R. (1987). Rule induction with statistical data – a comparison with multiple regression. *Journal of the Operational Research Society*, 38, 347 - 352.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Mateo: Morgan Kaufmann.

Ramachandran, A. and Young, R. M. (2005). Providing intelligent help across applications in dynamic user and environment contexts. In Proceedings of the 10th international Conference on intelligent User interfaces (San Diego, California, USA, January 10 - 13, 2005). IUI '05. ACM Press, New York, NY, 269-271.

Richardson, M. & Domingos, P. (2003). Building large knowledge bases by mass collaboration. In *Proceedings of the 2nd international Conference on Knowledge Capture* (Sanibel Island, FL, USA, October 23 - 25, 2003). K-CAP '03. ACM Press, New York, NY, 129-137.

Russell, S., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach (2nd Edition)*. Upper Saddle River: Pearson Education.

Saito, M. & Ohmura, K. (1998). A cognitive model for searching for III-defined targets on the Web: the relationship between search strategies and user satisfaction. In *Proceedings of the 21st Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (Melbourne, Australia, August 24 - 28, 1998). SIGIR '98. ACM Press, New York, NY, 155-163.

Winston, P. (1992). *Artificial Intelligence (3rd Edition)*. Reading: Addison-Wesley Publishing Company.

Witten, I., & Frank, E. (2000). *Data Mining: Practical Machine Learning Tools with Java Implementations*. San Francisco: Morgan Kaufmann.

Yianilos, P. N. (1993). Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms* (Austin, Texas, United States, January 25 - 27, 1993). Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, Philadelphia, PA, 311-321.

Appendix A - Pilot Study 1 Questionnaire

Post Study Questionnaire

1. How intrusive did you find the feedback window? (circle one)

1 2 3 4 5
Not Very ←--→ Very

2. How difficult were the Excel tasks? (circle one)

1 2 3 4 5
Not Very ←--→ Very

3. How useful were the help items returned by the Microsoft Office Help Search? (circle one)

1 2 3 4 5
Not Very ←--→ Very

4. How many Excel tasks were you able to complete without using the Microsoft Office Help Search? Please enter a number 0 - 7: ____

5. Were the prompts and options in the feedback window clear? (y / n)

6. What would you change about the feedback window?

7. What would you change about the Excel tasks?

8. Please provide any additional comments or suggestions about the quality of the feedback system.

Appendix B - Pilot Study 2 Questionnaire

Post Study Questionnaire

1. How intrusive did you find the feedback window? (circle one)

1 2 3 4 5
Not Very ←---→ Very

2. How difficult were the Excel tasks? (circle one)

1 2 3 4 5
Not Very ←---→ Very

3. How useful were the help items returned by the Microsoft Office Help Search? (circle one)

1 2 3 4 5
Not Very ←---→ Very

4. What is your level of expertise with Excel? (circle one)

1 2 3 4 5
Low ←---→ High

5. How many Excel tasks were you able to complete without using the Microsoft Office Help Search? Please enter a number 0 - 7: ____

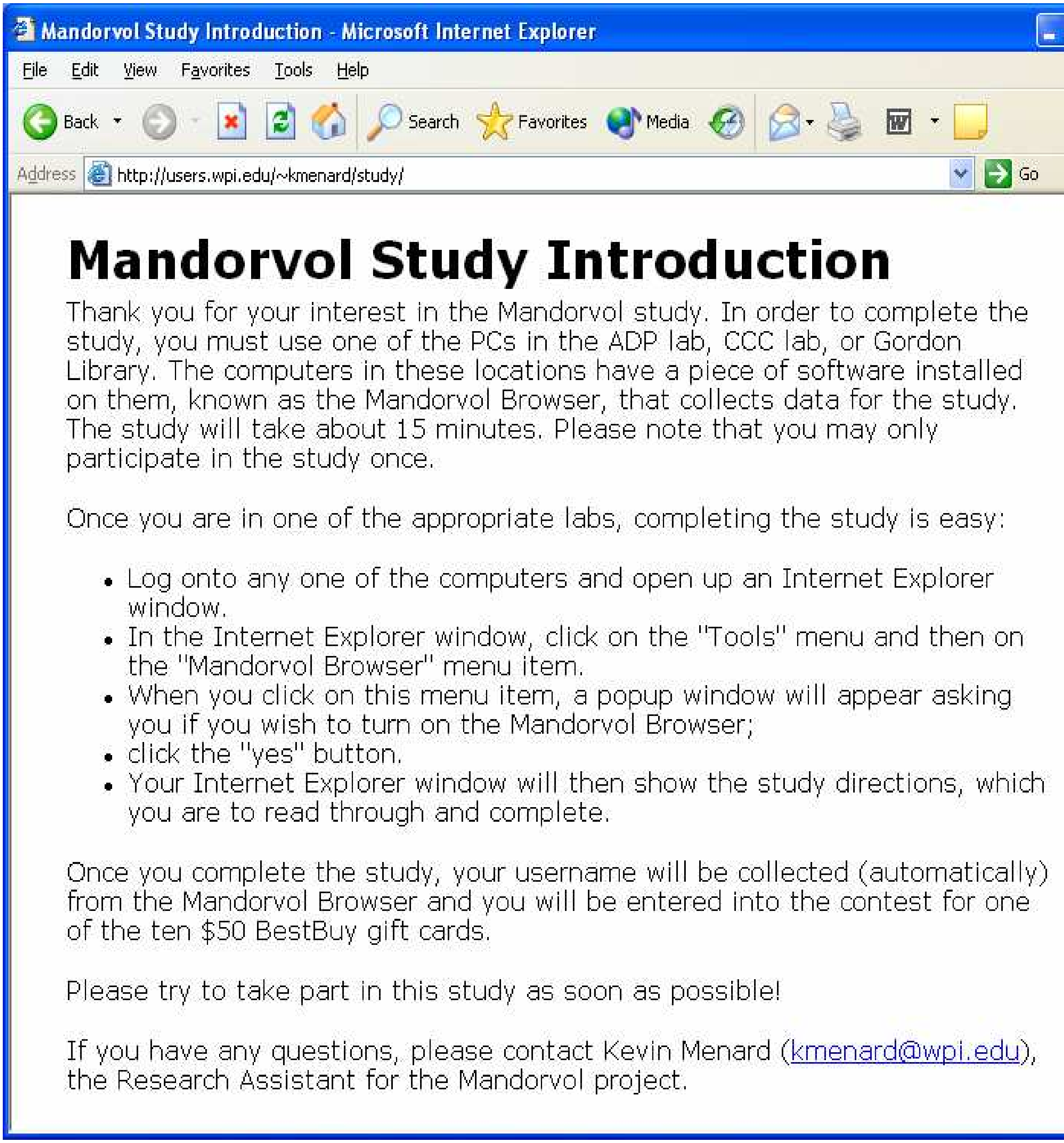
6. Were the prompts and options in the feedback window clear? (y / n)

7. What would you change about the feedback window?

8. What would you change about the Excel tasks?

9. Please provide any additional comments or suggestions about the quality of the feedback system.

Appendix C - Mandorvol Study Introduction



Mandorvol Study Introduction - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://users.wpi.edu/~kmenard/study/
Go

Mandorvol Study Introduction

Thank you for your interest in the Mandorvol study. In order to complete the study, you must use one of the PCs in the ADP lab, CCC lab, or Gordon Library. The computers in these locations have a piece of software installed on them, known as the Mandorvol Browser, that collects data for the study. The study will take about 15 minutes. Please note that you may only participate in the study once.

Once you are in one of the appropriate labs, completing the study is easy:

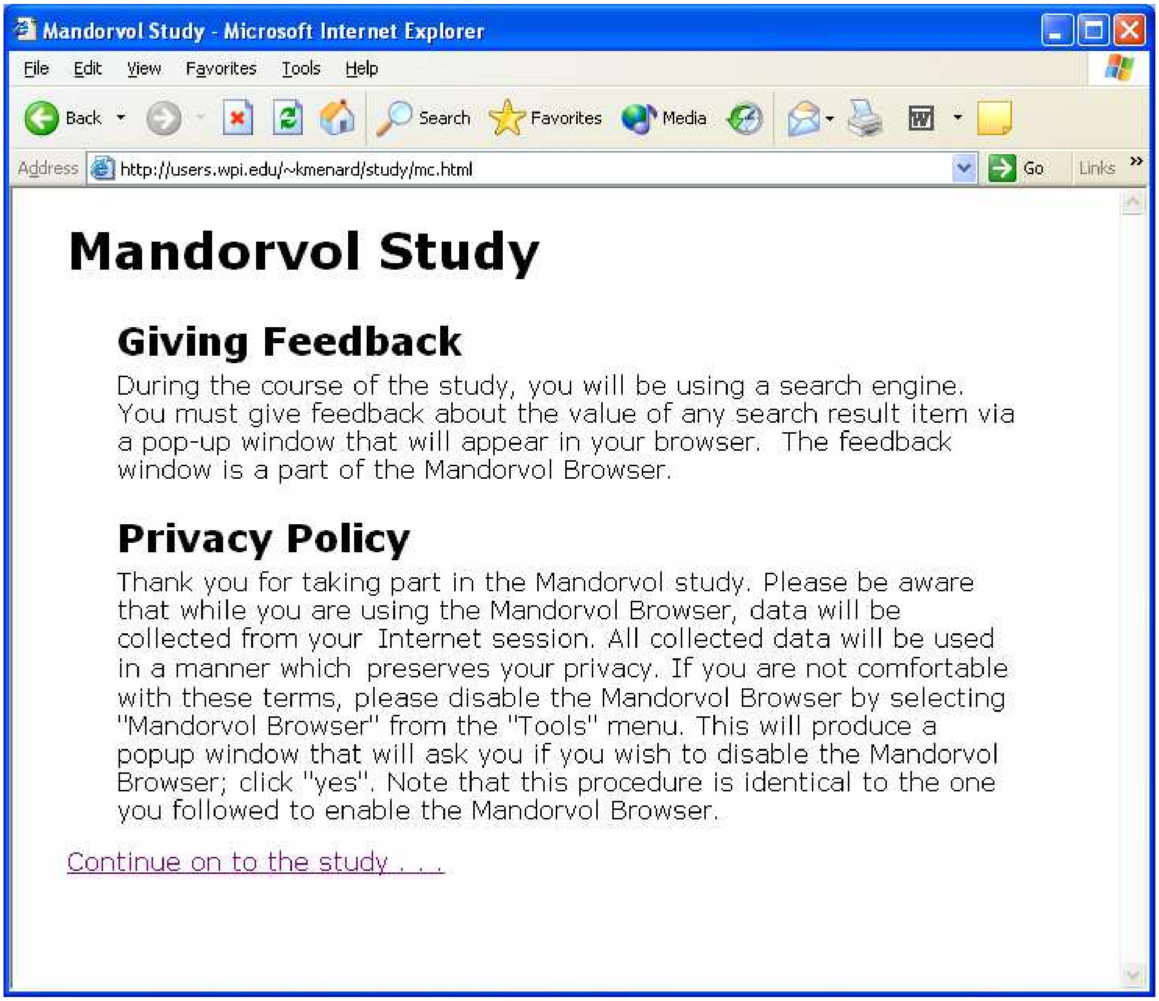
- Log onto any one of the computers and open up an Internet Explorer window.
- In the Internet Explorer window, click on the "Tools" menu and then on the "Mandorvol Browser" menu item.
- When you click on this menu item, a popup window will appear asking you if you wish to turn on the Mandorvol Browser;
- click the "yes" button.
- Your Internet Explorer window will then show the study directions, which you are to read through and complete.

Once you complete the study, your username will be collected (automatically) from the Mandorvol Browser and you will be entered into the contest for one of the ten $50 BestBuy gift cards.

Please try to take part in this study as soon as possible!

If you have any questions, please contact Kevin Menard (kmenard@wpi.edu), the Research Assistant for the Mandorvol project.

Appendix D - Mandatory Controlled Experiment Directions


Mandorvol Study - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://users.wpi.edu/~kmenard/study/mc.html
Go
Links

Mandorvol Study

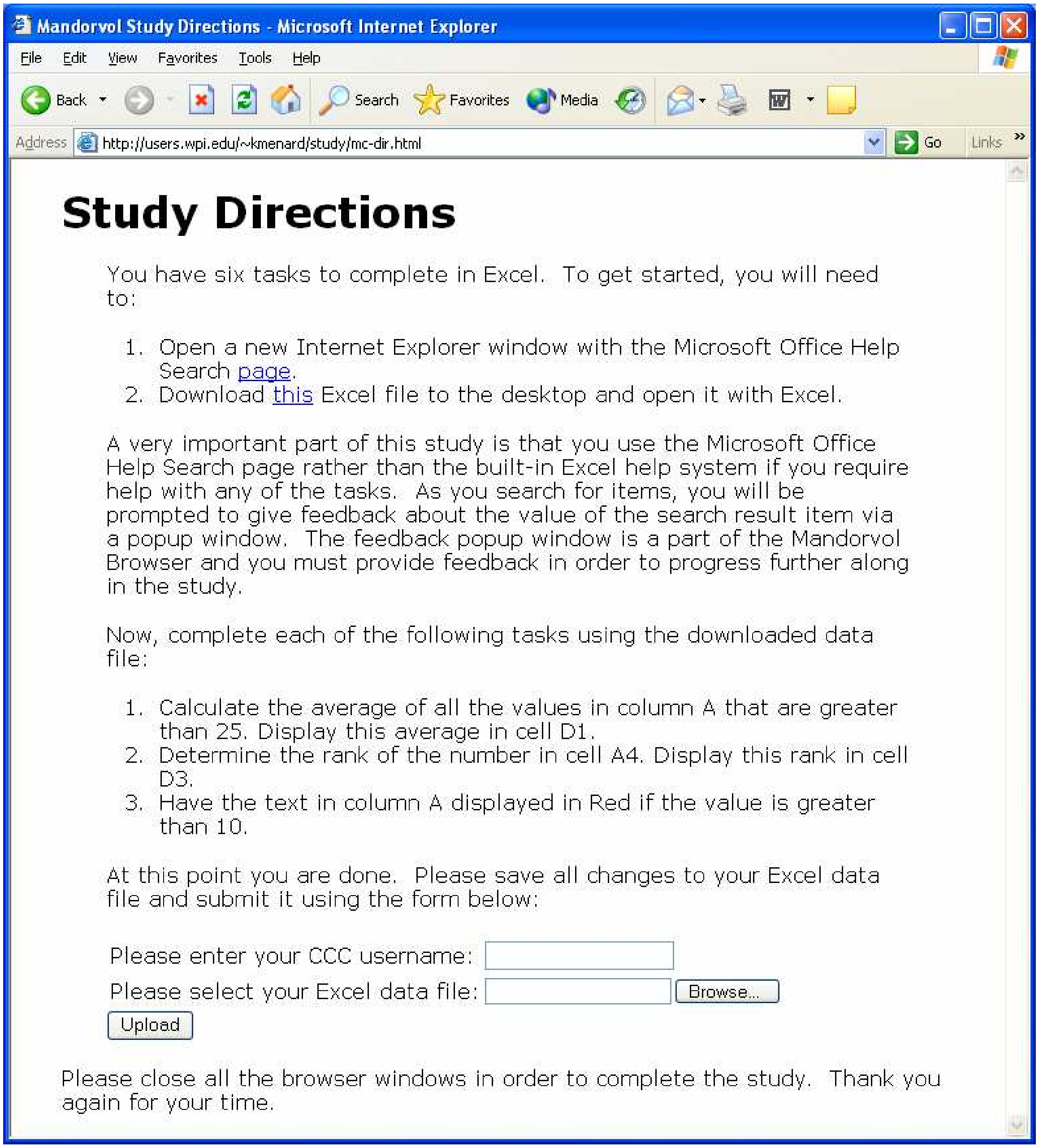
Giving Feedback

During the course of the study, you will be using a search engine. You must give feedback about the value of any search result item via a pop-up window that will appear in your browser. The feedback window is a part of the Mandorvol Browser.

Privacy Policy

Thank you for taking part in the Mandorvol study. Please be aware that while you are using the Mandorvol Browser, data will be collected from your Internet session. All collected data will be used in a manner which preserves your privacy. If you are not comfortable with these terms, please disable the Mandorvol Browser by selecting "Mandorvol Browser" from the "Tools" menu. This will produce a popup window that will ask you if you wish to disable the Mandorvol Browser; click "yes". Note that this procedure is identical to the one you followed to enable the Mandorvol Browser.

Continue on to the study . . .



Mandorvol Study Directions - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address http://users.wpi.edu/~kmenard/study/mc-dir.html
Go
Links

Study Directions

You have six tasks to complete in Excel. To get started, you will need to:

1. Open a new Internet Explorer window with the Microsoft Office Help Search page.
2. Download this Excel file to the desktop and open it with Excel.

A very important part of this study is that you use the Microsoft Office Help Search page rather than the built-in Excel help system if you require help with any of the tasks. As you search for items, you will be prompted to give feedback about the value of the search result item via a popup window. The feedback popup window is a part of the Mandorvol Browser and you must provide feedback in order to progress further along in the study.

Now, complete each of the following tasks using the downloaded data file:

1. Calculate the average of all the values in column A that are greater than 25. Display this average in cell D1.
2. Determine the rank of the number in cell A4. Display this rank in cell D3.
3. Have the text in column A displayed in Red if the value is greater than 10.

At this point you are done. Please save all changes to your Excel data file and submit it using the form below:

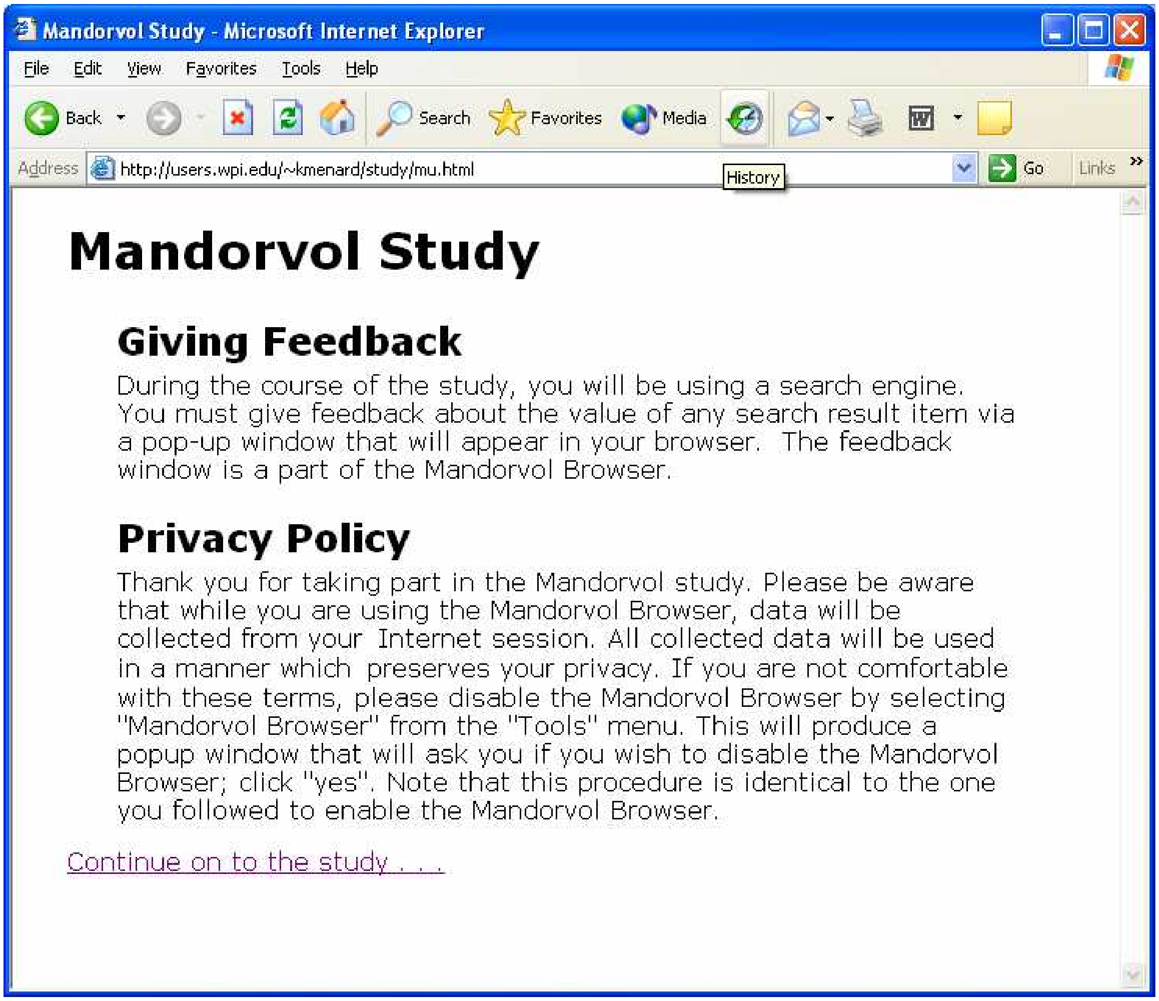
Please enter your CCC username:

Please select your Excel data file: Browse...

Upload

Please close all the browser windows in order to complete the study. Thank you again for your time.

Appendix E - Mandatory Uncontrolled Experiment Directions


Mandorvol Study - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
History
Address http://users.wpi.edu/~kmenard/study/mu.html
Go
Links

Mandorvol Study

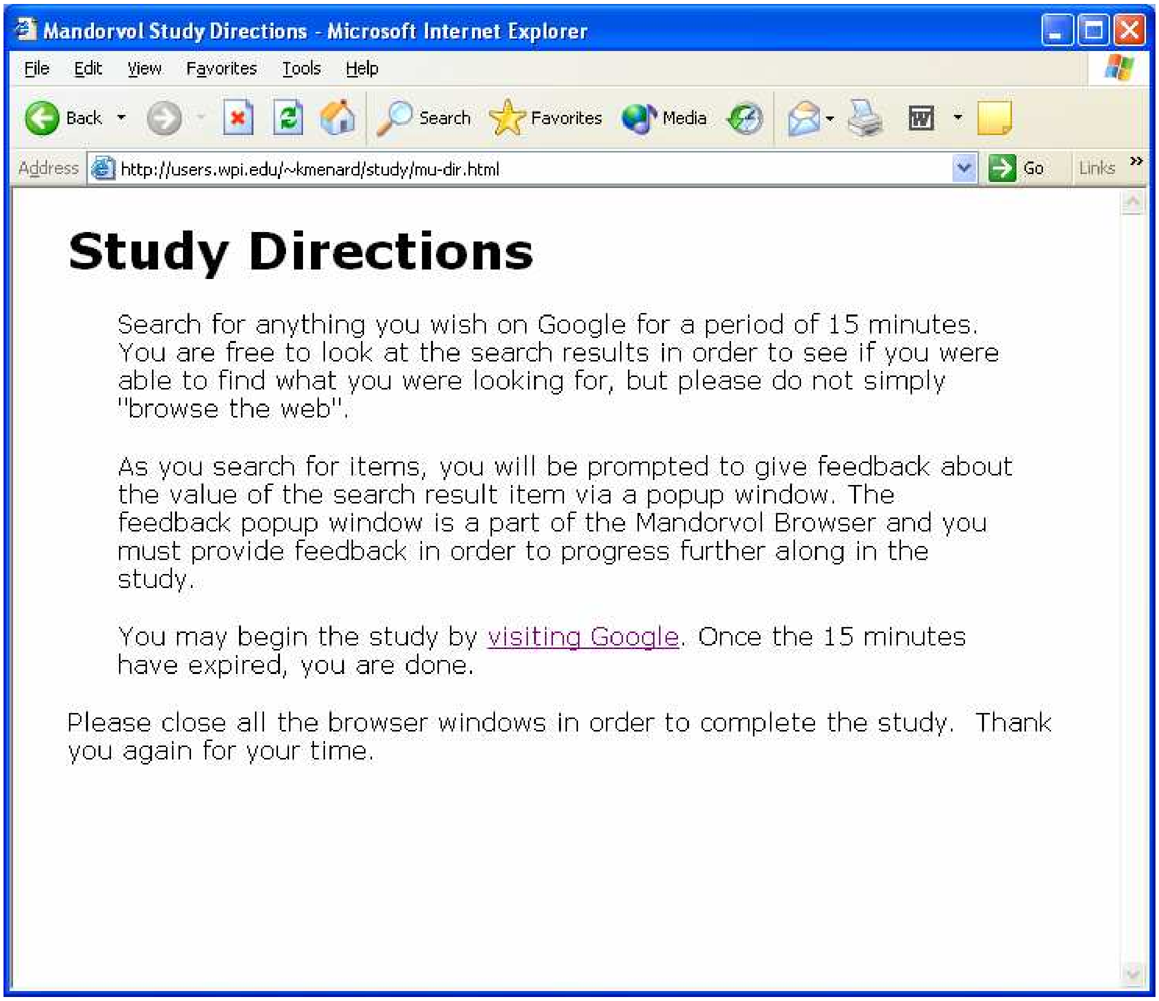
Giving Feedback

During the course of the study, you will be using a search engine. You must give feedback about the value of any search result item via a pop-up window that will appear in your browser. The feedback window is a part of the Mandorvol Browser.

Privacy Policy

Thank you for taking part in the Mandorvol study. Please be aware that while you are using the Mandorvol Browser, data will be collected from your Internet session. All collected data will be used in a manner which preserves your privacy. If you are not comfortable with these terms, please disable the Mandorvol Browser by selecting "Mandorvol Browser" from the "Tools" menu. This will produce a popup window that will ask you if you wish to disable the Mandorvol Browser; click "yes". Note that this procedure is identical to the one you followed to enable the Mandorvol Browser.

Continue on to the study . . .


Mandorvol Study Directions - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://users.wpi.edu/~kmenard/study/mu-dir.html
Go
Links

Study Directions

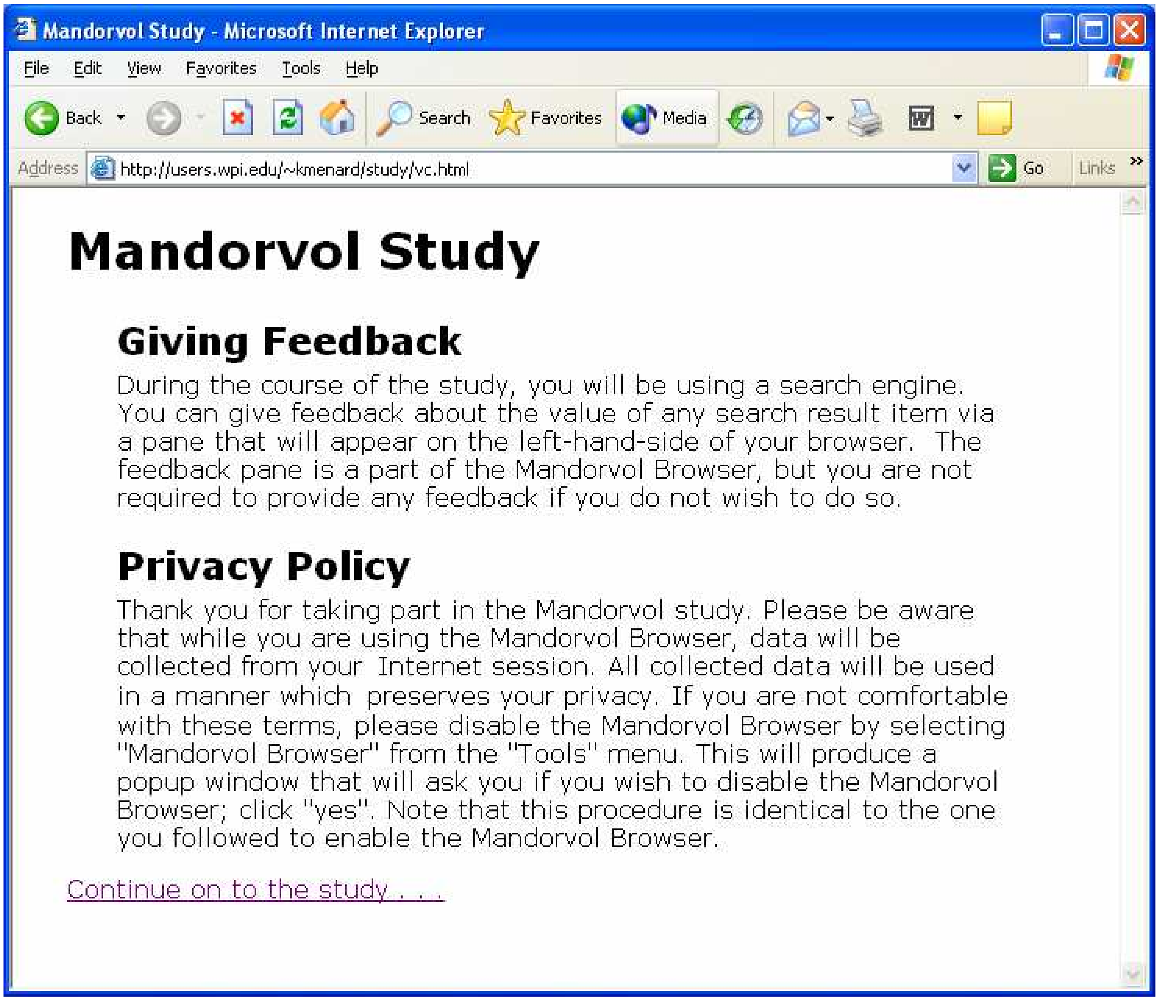
Search for anything you wish on Google for a period of 15 minutes. You are free to look at the search results in order to see if you were able to find what you were looking for, but please do not simply "browse the web".

As you search for items, you will be prompted to give feedback about the value of the search result item via a popup window. The feedback popup window is a part of the Mandorvol Browser and you must provide feedback in order to progress further along in the study.

You may begin the study by visiting Google. Once the 15 minutes have expired, you are done.

Please close all the browser windows in order to complete the study. Thank you again for your time.

Appendix F - Voluntary Controlled Experiment Directions


Mandorvol Study - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://users.wpi.edu/~kmenard/study/vc.html
Go
Links

Mandorvol Study

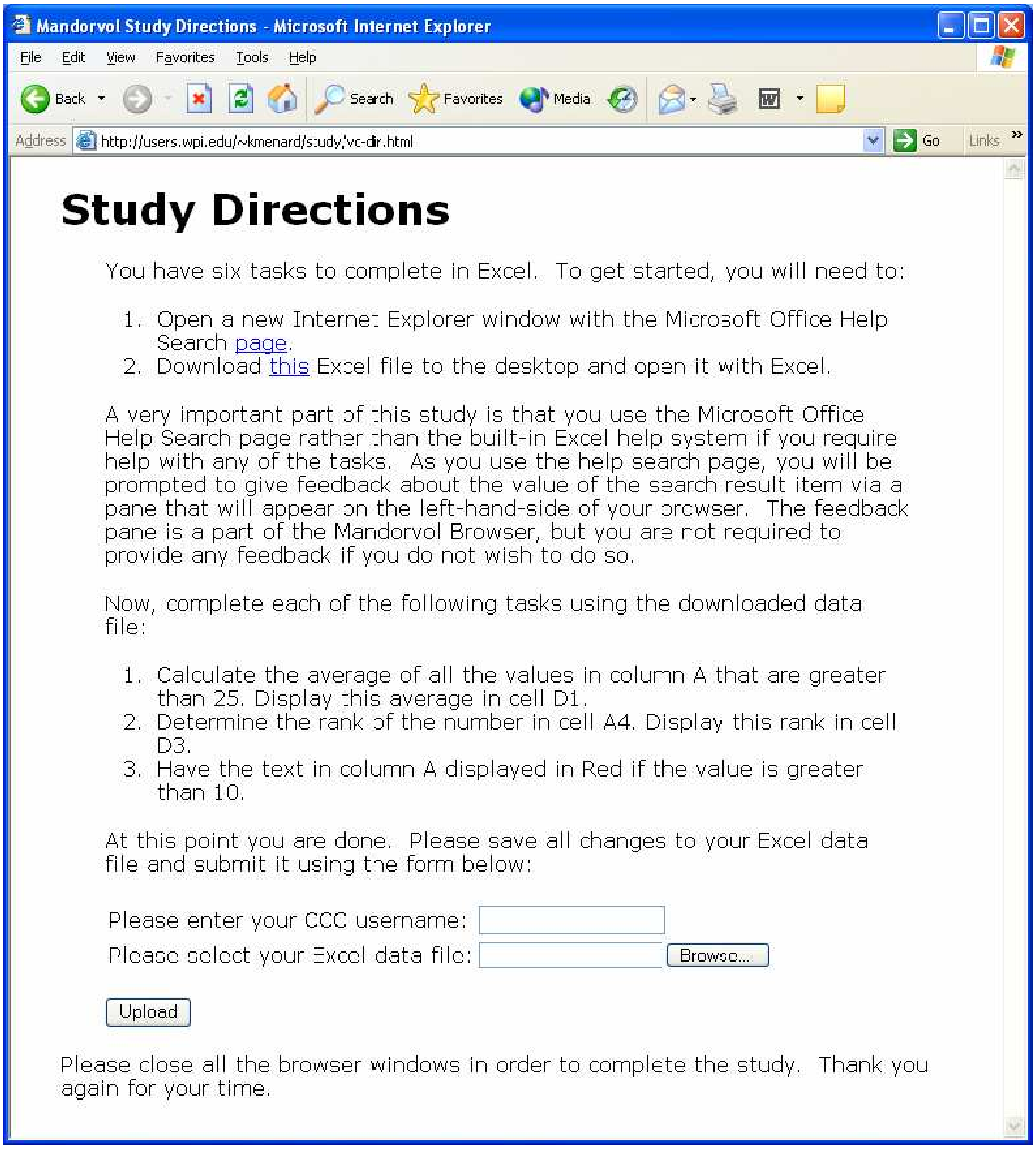
Giving Feedback

During the course of the study, you will be using a search engine. You can give feedback about the value of any search result item via a pane that will appear on the left-hand-side of your browser. The feedback pane is a part of the Mandorvol Browser, but you are not required to provide any feedback if you do not wish to do so.

Privacy Policy

Thank you for taking part in the Mandorvol study. Please be aware that while you are using the Mandorvol Browser, data will be collected from your Internet session. All collected data will be used in a manner which preserves your privacy. If you are not comfortable with these terms, please disable the Mandorvol Browser by selecting "Mandorvol Browser" from the "Tools" menu. This will produce a popup window that will ask you if you wish to disable the Mandorvol Browser; click "yes". Note that this procedure is identical to the one you followed to enable the Mandorvol Browser.

Continue on to the study . . .


Mandorvol Study Directions - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://users.wpi.edu/~kmenard/study/vc-dir.html
Go
Links

Study Directions

You have six tasks to complete in Excel. To get started, you will need to:

1. Open a new Internet Explorer window with the Microsoft Office Help Search page.
2. Download this Excel file to the desktop and open it with Excel.

A very important part of this study is that you use the Microsoft Office Help Search page rather than the built-in Excel help system if you require help with any of the tasks. As you use the help search page, you will be prompted to give feedback about the value of the search result item via a pane that will appear on the left-hand-side of your browser. The feedback pane is a part of the Mandorvol Browser, but you are not required to provide any feedback if you do not wish to do so.

Now, complete each of the following tasks using the downloaded data file:

1. Calculate the average of all the values in column A that are greater than 25. Display this average in cell D1.
2. Determine the rank of the number in cell A4. Display this rank in cell D3.
3. Have the text in column A displayed in Red if the value is greater than 10.

At this point you are done. Please save all changes to your Excel data file and submit it using the form below:

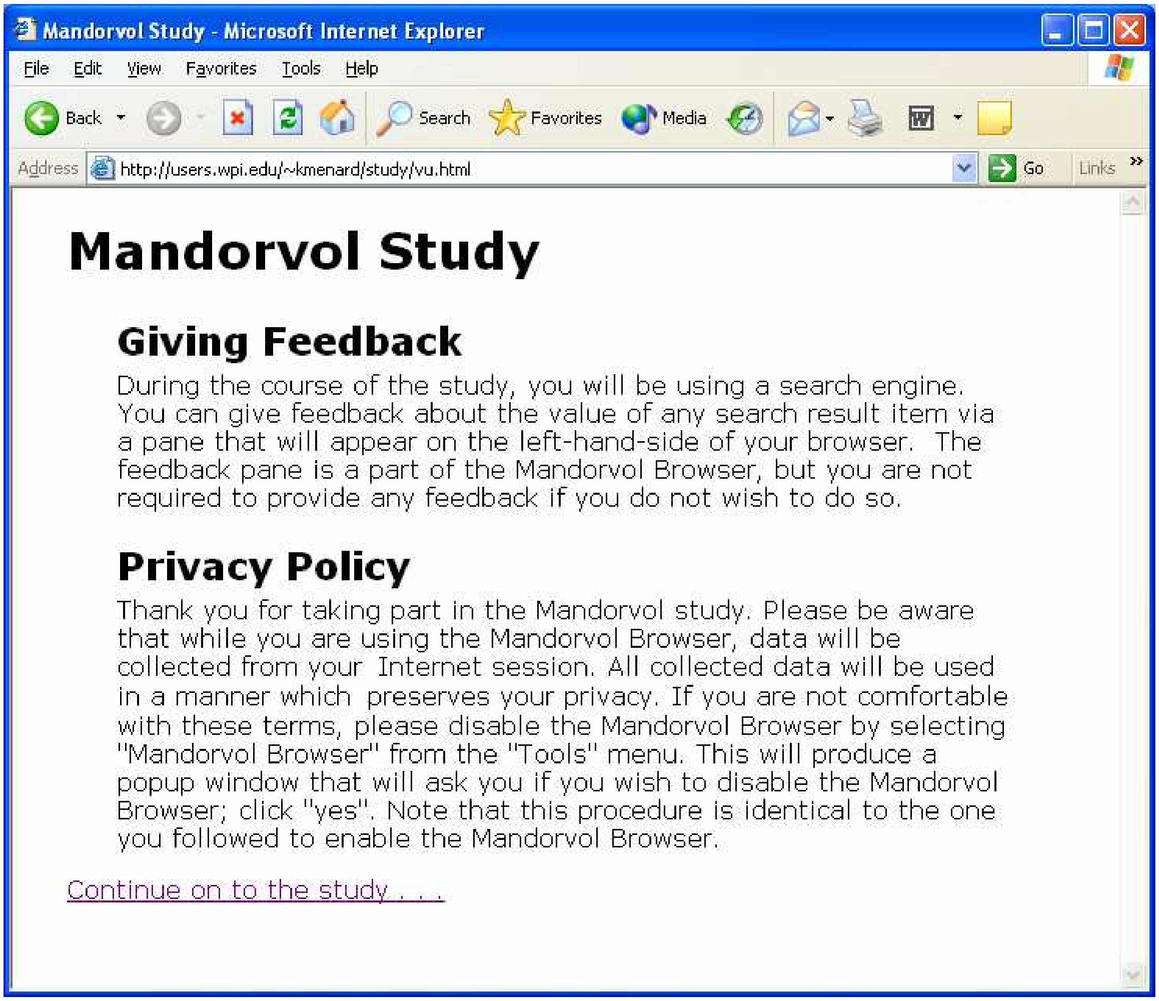
Please enter your CCC username:

Please select your Excel data file: Browse...

Upload

Please close all the browser windows in order to complete the study. Thank you again for your time.

Appendix G - Voluntary Uncontrolled Experiment Directions



Mandorvol Study - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://users.wpi.edu/~kmenard/study/vu.html
Go
Links

Mandorvol Study

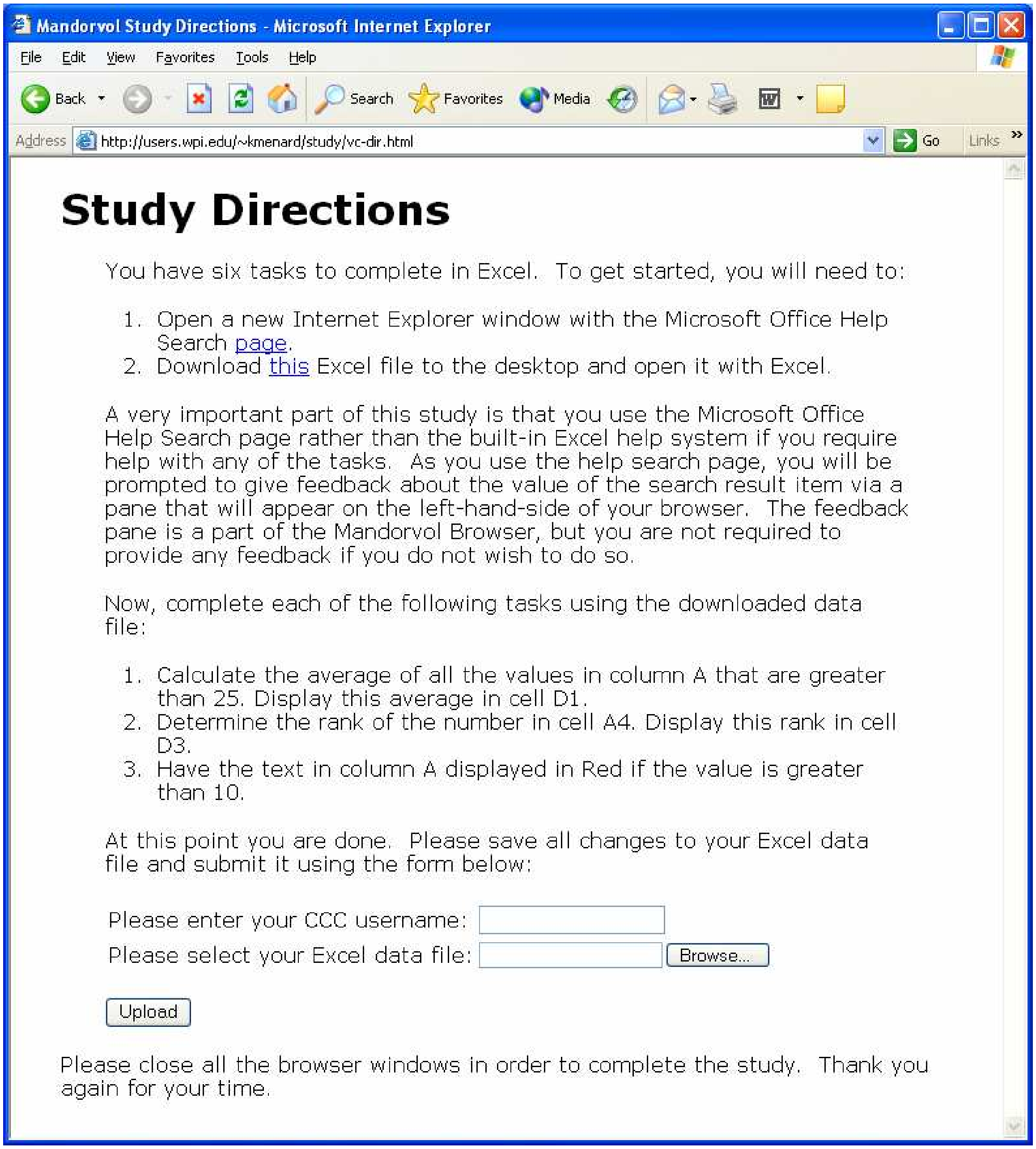
Giving Feedback

During the course of the study, you will be using a search engine. You can give feedback about the value of any search result item via a pane that will appear on the left-hand-side of your browser. The feedback pane is a part of the Mandorvol Browser, but you are not required to provide any feedback if you do not wish to do so.

Privacy Policy

Thank you for taking part in the Mandorvol study. Please be aware that while you are using the Mandorvol Browser, data will be collected from your Internet session. All collected data will be used in a manner which preserves your privacy. If you are not comfortable with these terms, please disable the Mandorvol Browser by selecting "Mandorvol Browser" from the "Tools" menu. This will produce a popup window that will ask you if you wish to disable the Mandorvol Browser; click "yes". Note that this procedure is identical to the one you followed to enable the Mandorvol Browser.

Continue on to the study . . .



Mandorvol Study Directions - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://users.wpi.edu/~kmenard/study/vc-dir.html
Go
Links

Study Directions

You have six tasks to complete in Excel. To get started, you will need to:

1. Open a new Internet Explorer window with the Microsoft Office Help Search page.
2. Download this Excel file to the desktop and open it with Excel.

A very important part of this study is that you use the Microsoft Office Help Search page rather than the built-in Excel help system if you require help with any of the tasks. As you use the help search page, you will be prompted to give feedback about the value of the search result item via a pane that will appear on the left-hand-side of your browser. The feedback pane is a part of the Mandorvol Browser, but you are not required to provide any feedback if you do not wish to do so.

Now, complete each of the following tasks using the downloaded data file:

1. Calculate the average of all the values in column A that are greater than 25. Display this average in cell D1.
2. Determine the rank of the number in cell A4. Display this rank in cell D3.
3. Have the text in column A displayed in Red if the value is greater than 10.

At this point you are done. Please save all changes to your Excel data file and submit it using the form below:

Please enter your CCC username:

Please select your Excel data file: Browse...

Upload

Please close all the browser windows in order to complete the study. Thank you again for your time.

Appendix H - User Participation Encouragement

Subject: Help with CS Research! Win a prize!

Hi,

We need your help!

Members of the Computer Science Department are doing some important funded research about improving the results of search engines and we need participants in our experimental study. The results should help many people so this makes it very exciting to take part.

Every person who completes the study will be entered into a drawing to win one of ten $50 BestBuy gift cards.

The study takes approximately 15 minutes to complete and can be done at your leisure in the ADP lab, CCC lab, or Gordon Library.

Please visit

http://www.wpi.edu/~kmenard/study/

for directions on how to get started. To be entered into the drawing, you must complete the study by April 4.

Thank you,

Kevin J. Menard, Jr.

Research Assistant, Mandorvol Project
WPI Computer Science Department