Automatically Labeling Web Pages Based on Normal User Actions

Jeremy Goecks[1]
Computer Sciences Dept.
University of Wisconsin
1210 W. Dayton Street
Madison, WI 53706-1685
U. S. A.

Jude Shavlik[2]
Computer Science Dept.
University of Wisconsin
1210 W. Dayton Street
Madison, WI 53706-1685
U. S. A.

Abstract

For agents attempting to learn a user's interests, the cost of obtaining labeled training instances is prohibitive because the user must directly label each training instance, and few users are willing to do so. We present an approach that circumvents the need for human-labeled pages. Instead, we learn 'surrogate' tasks where the desired output is easily measured, such as the number of hyperlinks clicked on a page or the amount of scrolling performed. Our assumption is that these outputs will highly correlate with the user's interests. In other words, by unobtrusively 'observing' the user's behavior we are able to learn functions of value. For example, an agent could silently observe the user's browser behavior during the day, then use these training examples to learn such functions and gather, during the middle of the night, pages that are likely to be of interest to the user. Previous work has focused on learning a user profile by passively observing the hyperlinks clicked on and those passed over. We extend this approach by measuring user mouse and scrolling activity in addition to user browsing activity. We present empirical results that demonstrate our agent can accurately predict some easily measured aspects of one's use of his or her browser.

1 Introduction

Much research has been devoted to the task of developing intelligent agents that can learn a user's interests (a 'profile') and find information in the World Wide Web (WWW) based on such profiles [e.g., Joachims et al, 1997; Pazzani et al, 1996]. Given a particular web page or hyperlink and a particular user, the agent's task is to predict the user's interest level in that page or hyperlink. If the agent predicts that the user will be very interested in the page, such an agent can retrieve this page for later viewing by the user.

A central question in the topic of learning user profiles concerns how the learning algorithm obtains training examples. Expecting the user to manually rate many pages is not desirable. Previous work [Liberman, 1995; Mladenic, 1996] has investigated employing passively recorded user navigation behavior (i.e. analyzing hyperlinks clicked, hyperlinks passed over, and patterns of user behavior) as *surrogate* measures for predicting user interest in a page. We extend this work by unobtrusively recording user navigation behavior *and additional user actions* (such as scrolling). We utilize a combination of these measurement as a surrogate for user interest in a page. We hypothesize (but have not yet tested) that these measurements correlate well with the user's true interests. Since we can collect these automatically generated training examples during the user's normal use of their web browser, our approach allows for the collection of large training sets without burdening the user. We demonstrate in this paper that it is feasible to learn some promising surrogate measurements.

To illustrate how an agent might generate labeled instances from normal user actions, let us consider an example. Imagine an agent watching a user who is interested in financial information about Internet bookstores. When the user navigates to a news article about the stock prices and future prospects of 'Amazon.com' and 'Barnes and Noble,' he or she is likely to scroll down the page reading the article and click on related links after finishing the article. Observing that the user performed a large number of actions on the page, the agent would label the page as a positive instance of the user's interests. Conversely, upon navigating to an article that compares the book selection and the book reviews at 'Amazon.com' and 'Barnes and Noble,' the user will likely not read the article, nor do any scrolling, nor click on any related links on the

[1] Jeremy Goecks is an undergraduate computer science major at the University of Wisconsin.
[2] Jude Shavlik is partially supported by NSF Grant IRI-9502990.

page. Upon observing the relatively few number of actions performed by the user, the agent would label this web page as a negative example of the user's interests. Thus, without any direct input from the user, the agent can generate labeled instances. Using normal user actions to obtain labeled training examples is much less costly to the user, and, as we will discuss below, quite effective in learning a profile of the user.

In Section 2, we discuss how the agent obtains training examples and generalizes from these examples. Sections 3 and 4 discuss the methodology and the results of a cross-validation study for the agent, while Sections 5 and 6, respectively, discuss extensions to this research and its relation to previous research. Finally, we present some conclusions.

2 Agent Architecture

In this section we discuss our agent's architecture. The browser used by our agent is Microsoft's Internet Explorer® 4.0; we made the choice to use Internet Explorer® (IE) as opposed to another browser because the Internet Client SDK and Microsoft's Common Object Model (COM®) provide 'hooks' that allow a program to observe, record, and measure a variety of user actions.

Step 1. When the user visits a web page, record (a) the HTML contents of the page; and (b) normal actions performed by the user on the page.

Specifically, the HTML text of the page is the input of the instance, and the normal user actions are the output of the training example. Recording the HTML text of the page is a straightforward operation. We next consider the normal user actions recorded by the agent.

Various user actions that appear likely to correlate with the user's interest in a page include the number of hyperlinks clicked on by the user, the amount of scrolling the user performed, and whether the user bookmarked the page. In fact, our agent does record the number of hyperlinks clicked by the user for the page. However, our agent does not directly utilize the latter two actions, but for different reasons. Technology limitations currently prevent the agent from obtaining an accurate measure of the amount of scrolling by a user, and although bookmarking a page is likely the action most highly correlated with user interest in a page, it is too rare an event for it to be of much use. However, measuring only one action is unlikely to be sufficient for an agent to reliably label a web page as either interesting or uninteresting. Thus, in addition to recording the number of links clicked on by the user, our agent records the level of activity for two other normal actions, which give the agent additional insight into the level of user interest in a page.

The first action serves as a surrogate measure of the amount of scrolling performed by the user on a page, and, similarly, the second action serves as a surrogate measure for mouse activity on the page.

The surrogate measure for the amount of scrolling performed by the user is the number of *command-state changes* on the page. A command-state change is an internal event in the IE event model that occurs when the user resizes the IE window, utilizes the Edit menu, or scrolls. The vast majority of the command-state changes are the result of user scrolling, and thus command-state changes serve as a reliable surrogate for the amount of user-scrolling activity.

Similarly, *status-bar-text* changes serve as the surrogate for mouse activity. Status-bar-text changes occur when the user moves the mouse over a hyperlink (the URL of the hyperlink is displayed in the status bar) or navigates through a menu (a description of the command pointed to by the mouse is displayed in the status bar). Hence, status bar text changes correlate with mouse activity.

For simplicity, we will refer to these surrogates by the action that each measures; hence, we will refer to the agent's measure of *user scrolling* and *mouse activity*.

Thus, when the user navigates to a new page, the agents records (a) the text of the HTML text; (b) the number of hyperlinks the user clicked on; (c) the amount of user scrolling activity; and (d) the amount of user mouse activity. Of these, (a) creates the input and (b)-(d) constitute the outputs of our training examples.

When the user returns to a page he or she has previously visited recently, it is logical that the agent should *add* the user actions recorded during this return visit to the user actions recorded during the previous visit. This makes sense because often a user navigates back-and-forth to many pages from a single 'start' page, and the user is likely very interested in this start page. Creating a new training example for the page each time and recording only the user actions for that particular visit does not capture the user's true interest in the page in this situation. Hence, the agent *sums* the actions of the user on each page visited over a finite period of time; the *active time* for an instance is this finite period of time during which the user's actions on a page are summed together.

Our agent currently uses an active time of twenty (20) minutes; future work will address means to determine the optimal active time for a user. If the user returns to the page after the active time, a new instance for the page is created and the actions are again during that instance's active time. By employing the concept of an instance active time, the agent attempts to capture the user's true interest in a page, whether the user performs many actions on a page in one visit or the user revisits a page many times and performs just a few actions on the page during each visit.

Step 2. Build labeled training instances from the information recorded in Step 1 and train on these instances.

The representation of the information collected by agent in Step 1 and the method of learning employed by the agent are highly interdependent. Hence, we discuss both the agent's input vector representation and its learning algorithm together in this step.

Our agent's fundamental goal is, given the HTML text of a web page, to predict the amount of normal user actions on the page. In other words, using the learning algorithm, the agent is to successfully predict the number of hyperlinks the user will click on, the amount of scrolling action the user will perform, and the amount of mouse action the user will do. Our agent employs a fully connected, three-layer neural network as its learning algorithm. Our choice to initially use a neural network was made largely out of convenience. We are currently evaluating other learning algorithms for this task.

Let us now discuss the topology of the neural network. We base our representation of the web page on the bag-of-words representation [Salton, 1991]; this representation is compatible with neural networks and empirically has proven quite successful in information retrieval [Lang, 1995; Pazzani et al., 1996; Salton, 1991]. The bag-of-words representation simply encodes the frequency of 'keywords' on a page of HTML and ignores word order. (We explain below how we obtain a set of key words and phrases). This 'frequency array' serves as the input to the network. Often these frequencies are normalized with respect to their expected values [e.g., *TFIDF*; Salton, 1991]; however, we are not yet performing such normalization in our agent.

Our agent uses an enhanced version of the basic bag-of-words representation. We equip our agent with the ability to handle 'key phrases,' which are simply phrases the agent is to look for as opposed to single words. Our agent can handle phrases containing up to three words. The agent simply treats a key phrase as a long word, and thus it handles keywords and key phrases in the same manner. Therefore, the number of input units for the network is the number of key words and phrases that are used by the agent.

We also make use of the majority of HTML markup tags to further distinguish the usage and context of the word. For example, a keyword found in the title of the document is given more weight than a keyword found in the body of the document. Similarly, keywords that are displayed in smaller fonts are given slightly less weight than keywords displayed in larger fonts.

We then employ two types of discounting to enhance the bag-of-words representation. The first type of discounting derives from the law of decreasing returns. For our purposes, this law says that the more often a keyword is seen, the less valuable it is. Thus, the first time a keyword is found on a page, its weight is W. However, the second time the keyword is found on the page, its weight is (0.7*W), the third time the word is found its weight is (0.5*W), and so on. We refer to this type of discounting as *occurrence discounting*. Occurrence discounting continues for as many times as the word is found on the page. Lastly, we use another discounting method in which we discount words according to their position on the page relative to the top of this page. We will refer to this discounting as *position discounting*. We hypothesize that a keyword is more valuable if it is near the top of the page versus if the keyword is near the bottom of the page. Enhancing the bag-of-words representation by using key phrases, HTML tags, occurrence discounting, and position discounting increases the richness of our input representation for a web page considerably.

Finally, we address the issue of acquiring keywords for a particular user. Numerous means come to mind, such as analyzing the user's homepage or examining the text of web pages visited by the user and using a heuristic function (e.g., *information gain* [Pazzani et al., 1996]) to choose the keywords from those pages' text. The focus of this research is independent of the means used to choose keywords, and thus for simplicity in our experiments below the agent is simply provided with a list of keywords. A production-quality agent would require a more sophisticated method of choosing keywords.

Each of the network output units corresponds to a particular user action; thus, there are three output units for the network. The output unit for *hyperlinks clicked* represents the fraction of the page's hyperlinks that are clicked. E.g., a value of 0.1 means that the user clicked 10% of the current page's hyperlinks. The outputs units for *scrolling activity* and *mouse* activity represent counts of the above-described events, scaled by 100.

The number of hidden units, N_h, is defined by the equation:

$$N_h = (|\text{Input Units}| + |\text{Output Units}|) / 4.$$

Finally, we train the network using the standard backpropagation algorithm. Once trained, the agent can now predict, based on the text of the page, the number of hyperlinks on the page the user will click on, the amount of scrolling and the amount of mouse activity the user will do.

Thus, we have described how our agent obtains training instances for its neural network by silently observing the user browsing the WWW. We have also discussed how the agent utilizes these instances to learn a profile of the user.

3 Experimental Methodology

In this section we present a cross-validation experiment that measures our agent's accuracy in predicting our three chosen user actions. To determine how accurately our agent can predict these normal user actions, we performed a 10-fold cross-validation experiment. For each fold we use 'early stopping' when training; 10% of each fold's training data is used as a tuning set. After training is complete, the agent restores the network that performed best on the tuning set. We then record the agent's accuracy on the fold's test set. We use the root-mean-square method to measure the accuracy of the agent's network.

We now turn to the data used in this study. The WWW is a particularly diverse environment, and thus obtaining a representative sample of web pages from the WWW is not trivial. Therefore, a short discussion of the type of data used in this experiment is relevant. As stated before, we provide

a list of keywords for the agent. The keywords for this study were related to machine learning and included such words and phrases as *machine learning*, *neural network*, *C4.5*, *Q-learning*, and *Quinlan*; in total, we used 107 key words and phrases. To collect the data, one of us (JG) browsed the WWW while our agent observed and collected data. The browsing attempted to simulate an individual interested in reading about various topics and research projects in machine learning; however, browsing was not performed with any particular questions or issues in mind. A total of 200 pages were visited; 150 of these pages were related to machine learning, and the remaining 50 were not. On each page about one fifth of the hyperlinks were clicked, there were about 40 scrolling-related events, and there were about 70 mouse-related events.

Due to the current technological limitations, our agent cannot obtain instances for pages with frames and/or server-side scripts. Hence, most data was collected at academic websites, which tend not to use these technologies. An effort was made to visit a broad range of pages and perform a number of different navigation actions, including the use of the forward/back buttons, clicking on 'dead' links and links to download files, and browsing FAQ's and ftp sites. Given this information, we argue that the data collected for this experiment is representative of a 'real-world' data sample that might be obtained from an ordinary user. Therefore, the results presented in Section 4 are likely representative of our agent's performance for a wide class of users.

4 Experimental Results

In this section, we present and discuss the results of the cross-validation study described in Section 3. Table 1 summarizes these results (recall that the average values for these three measurements were about 0.20, 0.40, and 0.70, respectively).

Table 1. Results of cross-validation study described in Section 3. The agent collected a total of two hundred (200) instances and used ten (10) train/test folds. For each fold, the agent set aside 10% of the data as a tuning set for early stopping. The agent trains on the train'(train instances minus tune instances) set for 100 epochs, chooses the network state that performs best on the tune set, then measures accuracy on the test instances.

Action Predicted	Root Mean Square Error on Test Sets
Hyperlinks Clicked	0.07 ± 0.02
Scrolling Activity	0.05 ± 0.02
Mouse Activity	0.13 ± 0.03

The data in table 1 shows that the error for predicting mouse activity was much larger than the error for the other output units. Although these results could indicate that user mouse activity on a page does not correlate with the user's interest in a page, we hypothesize that the agent's means of detecting mouse activity are not yet refined enough. Therefore the error observed in this experiment might well be a data-collection error, rather than indicative that this is a hard property to predict. Research is ongoing in an effort to design more accurate means of detecting mouse activity.

Lastly, it is important to remember that the instances created by the agent are not perfect. Indirect interpretations of the user's actions are most likely less accurate than directly asking the user for input. However, our claim is that this is overcome by the fact that we can collect so many examples (plus the user is much less burdened).

5 Related Work

As discussed earlier, previous work has focused on employing user navigation behavior as a surrogate to measuring user interest in a web page. Personal Web Watcher [Mladenic, 1996] and Letizia [Liberman, 1995] are examples of agents that utilize observed user browsing patterns to recommend hyperlinks to follow (in the case of Personal Web Watcher) or pages to view (in the case of Letizia). We extended this research by recording user mouse actions and user scrolling actions. As the amount of information available on the WWW continues to expand, it will become increasing necessary for users to have an assistance in finding information of interest to them; thus, it is likely that research on agents such as ours and those mentioned above will continue to grow.

An active recent topic in machine learning is reducing the need for labeled examples. For example, several recent papers described techniques that augment labeled training data with unlabeled data [e.g., McCallum & Nigam, 1998; Miller & Uyar, 1997; Nigam et al, 1998]. Reducing the need for human labeling is also the main focus of our project, however our approach is to use surrogate measures that web browsers and operating systems can easily obtain without any burden on the user. This perspective goes back at least to some of the mid-1980's work on *learning apprentices* [Mitchell et al, 1985]. Learning apprentices were typically defined as learning by observing normal human use of software systems, and aimed to avoid the need for explicit training modes.

6 On-Going and Future Work

A number of questions have arisen from our initial work on this agent. We discuss one possible extension of this agent and discuss an important enhancement to the agent.

We have described above an agent that accurately learns a user profile by observing the user while he or she is browsing the WWW. Let us now briefly summarize our

ongoing work on embedding the above neural network in a larger system. A fully autonomous agent would first build a profile of the user's interest, as described above, and then employ this profile in a heuristic search of the WWW to retrieve pages of interest to the user.

Our agent would greatly benefit from improved methods for detecting user actions. The three user actions currently detected by our agent provide a good 'first-order' approximation to the user's activity on a web page, but one can imagine much more sensitive methods of recording user activity. For instance, we asserted that command-state-change activity is a good approximation of the amount of scrolling a user does on a page; however, we can imagine a much more precise measure of scrolling that records the exact amount of scrolling by the user. Similarly, status-bar-text change is a good approximation to user mouse activity, but one can imagine must more precise ways of measuring mouse activity, as well. However, today's Internet technology, including even client-side page scripts (currently the most sensitive instruments for user activity), is not yet advanced enough to record this information. The most promising approach at the time is to intercept user actions by using 'hooks' into the operating system; it is possible to intercept scrolling activity and mouse activity using the Win32 API on Windows NT 4.0. In the future we would ideally embed the agent inside the web browser application, as opposed to the agent running as a separate application as it is currently. We expect that embedding the agent into the browser would facilitate considerable improvement in the agent's activity detection mechanisms. Fortunately, the trend in browsers (e. g., Microsoft's IE 5) is to provide increasingly more access by programs to the actions users perform.

We plan to perform experiments on our agent using human subjects. For each subject, the agent will collect a number of instances using the methods discussed in this paper. After collecting these instances, the agent will learn a user profile for the subject from the instances and utilize the profile in a heuristic search of the WWW for pages of interest to the subject. The subject will then rate (i.e., the subject will assign a number between 0 and 10) the pages returned by the agent according to his or her interest in them. Thus, in this experiment, the agent employs the learned user profile in a practical task and the agent's results are judged using a concrete standard.

Lastly, consider the quality of the instances generated by our agent. Future research should address our agent's performance with its generated examples relative to its performance with examples constructed from user-labeled pages. Because our approach requires no effort from the user, a ratio of one hundred automatically generated examples to one user-labeled example would be acceptable. That is, assume we train our agent with a N user-labeled pages and it performs at a certain level L consistently; we would be happy if our agent had to collect $100N$ examples in order to reach performance level L.

7 Conclusions

We presented and evaluated the design of an agent that employs a standard neural network to learn a user's interests in the WWW. The key aspect of our system is that it unobtrusively measures normal actions performed by the user on a page. It uses these measurements as a surrogate for the desirable, but too burdensome to collect, measurements of the user's interest level. Hence, rather than explicitly labeling the interest level of WWW pages, users implicitly label web pages by the actions they perform on them. Our cross-validation experiment suggests that the agent can learn to predict, at a high degree of accuracy, the surrogate measurements of user interest that we investigated.

References

[Joachims et al, 1997] T. Joachims, D. Freitag and T. Mitchell. WebWatcher: A Tour Guide for the World Wide Web. *IJCAI-97*, pp. 770-775.

[Lang, 1995] K. Lang. NewsWeeder: Learning to Filter News, *ICML-95*, pp. 331-339.

[Lieberman, 1995] Liberman, H. Letizia: An Agent that Assists Web Browsing. *IJCAI-95, pp. 924-929.*

[McCallum and Nigam, 1998] A. McCallum and K. Nigam. Employing EM and Pool-Based Active Learning for Text Classification. *ICML-98*, pp. 350-358.

[Miller and Uyar, 1997] D. Miller and Uyar. A Mixture of Experts Classifier with Learning Based on Both Labelled and Unlabelled Data. *Advances in Neural Information Processing* (NIPS-9), pp. 571-577.

[Mitchell et al, 1985] T. Mitchell, S. Mahadevan, and L. Steinberg. LEAP: A Learning Apprentice for VLSI Design. *IJCAI-85*., pp 574-580.

[Mladenic, 1996] Mladenic, D. Personal WebWatcher: Implementation and Design, *Technical Report IJS-DP-7472*, Department for Intelligent Systems, J. Stefan Institute, October, 1996.

[Nigam et al, 1998] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Learning to Classify Text from Labeled and Unlabelled Documents, *AAAI-98*, pp. 792-799.

[Pazzani et al, 1996] M. Pazzani, J. Muramatsu, and D. Billsus. Syskill & Webert: Identifying Interesting Web Sites, *AAAI-96*, pp. 54-61.

[Salton, 1991] G. Salton. Developments in Automatic Text Retrieval, *Science* 253:974-97