IMPORTANT PAGE PREDICTING FOR INTERNET RECOMMENDATION : A MACHINE LEARNING WAY

Tingshao Zhu, Russ Greiner

Dept. of Computing Science
University of Alberta
Canada T6G 2E1

Gerald Häubl

School of Business
University of Alberta
Canada T6G 2R6

ABSTRACT

While the World Wide Web contains a vast quantity of information, it is often difficult for web users to find the information they need. If we can figure out what the user wants, we could save the user much time and effort by recommending the "important pages", which contain the information that she must examine to accomplish her task. While there are a number of current recommendation systems, most such systems use only the correlation among the pages visited to predict specific URLs as recommendation, but it is hard to say that these recommendation can really help user. This paper presents our preliminary research on identifying important pages, that is, based on current click stream, predicting whether the given URL is important or not. Classifiers are trained on annotated web data from our Travel Planning experiment, and some preliminary results are also presented.

1. INTRODUCTION

While the World Wide Web contains a vast quantity of information, it is often difficult for web users to find the information they need. This has lead to the development of a number of *recommendation systems*, which typically watch a user as she navigates through a sequence of pages, and suggests pages that (they hope) will provide the relevant information ([6], [9], etc).

These systems are often based on correlations amongst the pages. But unfortunately, there is no reason to believe that these pages will contain useful information — indeed, they may correspond simply to the paths that others have taken towards their goals, or worse, simply to standard dead-ends that everyone seems to hit.

In this paper, we take seriously the task of helping the user reach the pages that contains the information she really wants. In particular, we say a page is "important" to a user, in a specific context (of her earlier click-stream, her (implicit) task, and other information; see below), if that page contains the information that she must examine to accomplish her task. To do so, at first we conduct an experiment to collect labeled web log data, that is, the subject should label which page is her important page while browsing. After the data collection, we train classifiers to predict important pages for web user.

In Section 2, we will give brief introduction of the related research. Section 3 describes the whole process of our research. Section 4 introduces the tool that we developed for the data collection, the experiment process, and some general statistics of the log data. In Section 5, we use this collected information to train classifiers for important pages. Section 6 shows the preliminary results of several learning tools on this task.

2. RELATED WORK

There are lots of research have been done on the recommendation generation for web user, , and this section will summerize serveral related approaches, then discuss how they differ from ours.

Collaborative Filtering [12] is the first attempt using AI technology for personalization [8], but it is unrealistic to ask the user to rank all the pages that explored, and it is very difficult to get enough manually labeled web pages in real world. In our research, we try to learn patterns for important pages identification, and if we focus on a specific topic, we can build our learning system on a much smaller dataset.

Mobasher et al. [6] report their personalization based on ARHP[7]. In [5], they generated recommendations from URL clusters to build an adaptive web site.

In IJCAI-97, Mike Perkowitz and Oren Etzioni challenged the AI community to create adaptive web sites: web sites that automatically improve their organization and presentation by learning from user access patterns [9]. And in [10], they propose PageGather for adaptive web site construction, but since PageGather is based from frequency and co-occurence, it doesnot work in our case.

Association Rule and Sequential pattern are two methods that widely used in recommendation system. An *association rule* is a rule of the form "$U_1,\ U_2,\ U_3\ \rightarrow\ V_1$" where

the U_is and V_js are each URLs, with the intended meaning that a user who has visited the URLs on the left side (here, U_1, U_2 and U_3), will typically also visit the URLs on the right side (here V_1). There are many systems that attempt to learning such rules, such as Apriori [1]. While almost all of these reports claim their algorithms can find the *appropriate* association rules, none have demonstrated that the rules produced are truly useful in web applications.

[2] supplies a standard definition of *sequential pattern discovery*: Given a database D of customer transactions, find the maximal length subsequences among all sequences that occur at least some (user-specified) minimum number of items.

ARHP, PageGather, AR and SP are widely used methods to predict specific URLs for recommendation, but they are not capable in our case due to the special characteristics of our dataset. In Section 4.2 we can see that most of the pages have been visited only once across the whole dataset, so according to these above methods, very few or even no clusters, rules or sequentials patterns can be obtained, thus the recommendation system will keep in silence almost all the time.

All the above methods just try to predict specific URLs based on frequency and co-occurence, but nothing to do with important or not, and the recommended URLs must be in the training data, so it is very possible that the recommended pages are meaningless, or even worse, misleading user. In our research, we try to infer some general patterns to identify important pages, which can be used not only on the training data. Our first step is to collect labeled web data, and we have developed tool and conducted experiment to do so.

3. IMPORTANT PAGE PREDICTION AS A CLASSIFICATION TASK

To avoid simply guessing which pages may qualify as important, or assuming that simple correlation somehow implies relevance, in our data collection phase, we conducted an experiment, in which 144 different users use a tool, – Annotation Internet Explorer(AIE), to explicitly indicate which pages were important for their specified task. This produced a data sample of "annotated web-logs", where each page in each web-log (over 15,000 pages) is labeled with a user-reported measure of whether she considered the page to be important or not. More details about this experiment will be introduced in Section 4.

In our experiment, participants have clear tasks before they use the Internet, and they do not need to print out any information by using our tool – AIE, so it is expected to easy the participants, so they would not skip some information to avoid the abundant print task.

In our research, we take important page prediction as a classification task, that is, give the observed page sequence and some properties of page P, predict whether P is important page or not. If such classifier can be obtained, combining with web crawler, we can infer which page will be important starting from the last page of the current click stream. Section 5 will give more detail for the learning process, and in Section 6 we will show some preliminary result of the classifier that we build from our web log data.

4. DATA COLLECTION

Recall our initial goal is to determine which pages were important — that is, which pages contain information required to complete a task. To do this, we collected a set of *annotated web-logs*, a sequence of webpages that a user visits, where each page is labeled with a bit that indicates whether this page is "important" — i.e. essential to achieving the user's specific task. We enlisted the service of a number of students (from School of Business at the University of Alberta) to provide these annotated web-logs.

Each participant was asked to perform a specific task:

1. Identify 3 novel vacation destinations (i.e. places you have never visited)

2. Plan a *detailed* vacation to each destination specifying specific travel dates, flight numbers, accomodation (hotels, campsite, . . .), activities, etc.

They were given about 45 minutes, and given access to our augmented browsing tool (AIE; see Section 4.1), which recorded their specific web-logs, and required them to provide the "importance" annotation. The participants also had to produce a short report summarizing the vacation plans, and citing the specific important webpages that were involved in these decisions; here AIE made it easy to remember and insert these citations. To help motivate the participants to take this exercise seriously, we told them that two (randomly selected) participants would win money to help pay for the specific vacation they had planned.

There are sevral reasons that we chose this specific task: (1). It represents a fairly standard way of using the web; (2) It was goal directed (in contrast to simply asking the participants to "meander about the web"); (3) The contents of the travel web sites do not change frequently; (4) A diverse set of pages may be relevant — plane schedules, travel brochures, recent news (terrorist attacks), . . . ; (5) It is easy to motivate students to do this task, as they will get a chance to go on this trip; (6) The task is fairly well-defined and delimited; the references in the report help identify which pages qualify and which do not.

4.1. AIE: Annotation Internet Explorer

To help us collect the relevant information, we built an enhanced version of INTERNET EXPLORER, call AIE which we installed in all of the computers of the lab we used for our study.

As with all browsers, AIE user can see the current web page. This tool incorporates several relevant extensions. First, the user can declare the current page to be "important", by clicking the *Important* button on the top bar.

The *History* button on the toolbar brings up the side-panel, which shows the user the set of all pages seen so far, with a flag showing which pages the user tagged as important.

The *Report* button will switch the browse view to the report editor, which she can use to enter her report. Here, each subject has access to the pages she labeled as important during her browsing, which she can use in producing her report.

After completing her report, the user can then submit her entire session using the *Submit* button. This sends over the entire sequence of web-sites visited, together with the user's "important page" annotations, as well as other information, such as time-stamp for each page, etc.

4.2. Some Aspects of the Web Log Data

A total of 144 undergraduate business students participated in the experimet, and study sessions were administered in a supervised computer laboratory in groups of approximately 25 subjects. Due to technical problems, usable data were obtained from 129 participants. The number of the page requests is 15105, and actually there are only 5995 distinct URLs, so each URL is requested 2.519 times on average. The number of Important pages is 1887, each subject labels 14.627907 pages as important.

In Table 1, we list the number of requests Vs. the percentage of distinct URLs, each row shows how much percent of the distinct URLs has been requested how many times, for exmaple, 58.93% of distinct URLs have been requested only once. From Table 1, 82.39% of the URLs have been visited one or two times, and each URL has been requested 2.519 times. Very few URLs can get strong support in the dataset.

Table 1: Number of Requests vs Percentage of the URLs

Number of Request(s)	Percentage of the URLs
1	58.93%
2	23.46%
3	7.63%
4	4.08%
5	1.85%
6	1.16%
7	0.88%
8	0.40%
9	0.40%
10	0.18%
...	...

5. LEARNING TO CLASSIFY IMPORTANT PAGES

Our overall goal is to help users obtain the information they need, by directing them to specific pages that are "important", wrt their current information needs, etc.

Our first and simple step is to classify whether the current page is important, based on the available information. That is, assume at time t the user has visited the "annotated pages" $U_1, \pm_1, U_2, \pm_2, U_3, \pm_3, \ldots, U_{t-1}, \pm_{t-1}, U_t$, where each $\pm_i$ is "$+$" if this page was deemed important and is "$-$" otherwise. The challenge is to use this information (augmented with other data, such as length of time at each page, etc; see below), to determine whether U_t is important — i.e. the value of $\pm_t$.

Here, we try to learn this "important page classifier": Given a number of such annotated weblogs, learn a classifier that can take annoted page sequence as input, and determine whether the final page is important or not. (While this task does share some superficial similaritis with some of the standard web usage mining systems, we explain in Section 2 how our task is different.)

5.1. Source of Participants

A total of 144 undergraduate business students participated in the study for partial course credit and a lottery incentive.Study sessions were administered in a supervised computer laboratory in groups of approximately 25 subjects. Due to technical problems, usable data were obtained from 129 participants. These 129 annoted web-logs contained over 15,000 page requests; about 10% of these pages were labeled as important.

5.2. Imbalanced Dataset

As only 10% of the pages are important, there is a trivial way to obtain 90% accuracy: just return "not important" to each instance. Of course, this will not serve our needs — it is important to know which pages are important. To address this problem of "imbalanced data" [4], we generate testing and training data, from our set of 15,000 instances, as follows.

Testing Data Randomly select 100 important pages and 100 unimportant pages — so the total number of testing instances is 200.

Training Data From the remaining instances, the number of unimportant pages is denoted as `num_class`, then randomly draw, with replacement, `num_class` important pages. Notice if `num_class` is large, some important pages may appear several times in a single training sample.

5.3. Attributes Used

We propose serveral attributes from the click stream, and after simple feature selection, by removing some abundent features, we keep the following attributes in our training data. Note a site-session is the click stream within a single web domain — i.e. if the subject enters a new web site, a new site-session begins. The site-session is only a click stream segmentation method in our research, it doesnot mean that one site-session concentrate on one specific task.

1. URL Properties

   URL Type: wrong (e.g. "404"), search (e.g. `GOOGLE`), dynamic (e.g. produced by CGI script), static, (e.g. typical *.html page) misc (e.g. pointer to jpg, mpg, or mov file, or whatever)

   DomainType: wrong (404), edu, com, net, org, gov, misc

   Depth: Number of "/"s in the URL

2. User Click Stream

   FollowSearchEngine: Does this page follow immediately from some search engine (e.g. `GOOGLE`)?

   isLastEntry: whether this page is the last one in the site-session.

   inTotalNumberofPage: the number of pages that have been visited within this site-session, until now

   inTotalNumberofImportantPage: the number of pages, within this site-session, that have been labeled as important

   inLastImportant: the number of pages that have been visited since last important page within this site-session. If no previous important pages, just use the number of pages visited in this site-session until now

   TotalNumberOfPages: Until now, how many pages have been visited.

   TotalNumberOfImportantPages: Until now, the number of important pages.

   LastImportant: the number of pages have been visited since last important page. If no such previous important page, just use the number of pages visited until now.

   PercentageDomain: Until now, the percentage of the pages that have the same domain as this page.

   PercentageImportant: Until now, percentage of the important pages that have the same domain as this page.

   PercentageSameDomainImportant: Until now, for the same domain entries, the percentage of the important pages.

Note that we wanted to include "time" information — i.e. how long the user spent on each page. While we did record that information, we were unable to use it, as we found that many users switched modes (to "Report mode", in Section 4.1) on finding an important page. This means much of the time between requesting an important page, and request the next page, is in recording information, which skews the statistics. Before we ran the experiment, we assumed the subjects would browse the Internet first, marking important pages before switching to report writing. But in the lab, we found that most subjects did not follow this: after identifying an important page, they would switch to write report for this page, then switch back to find other relevant page, and so forth. Hence, the time spent on an important pages are not purely the reading time, but also the time for reporting. In our current system, we did not record the time spent in reporting mode, so the time information can not be used to predict important pages.

6. PRELIMINARY RESULTS

After the data preparation, we run several classification algorithms on the data set, producing

- decision tree, using C4.5 (see [11])
- NaiveBayes(NB) — a simple belief net structure which claims that the attributes are independent of one another, conditioned on the class label [3].
- Boosted NaiveBayes(BNB): In general, "boosting" is an approach to improve the result of a learning algorithm A, by using A to learn a set of classifiers over slightly different datasamples (which differ by reweighting the elements in training set); see [13]. Here, we boosted the NaiveBayes learner.

Note that we were able to run them all using the WEKA system, which is a large collection of learning algorithms; see [14], $http://www.cs.waikato.ac.nz/ml/weka$.)

In all cases, we used the default setting, and ran 10-fold cross validation. The "Precision" for Important pages is TruePositive / PredictedAsPositive and the "Recall" for Important pages is TruePositive / AllRealPositive. Of course, TruePositive are those pages that are predicted as positive

Table 2: Empirical Results for Important Prediction

C4.5	0.712 ± 0.063	0.27 ± 0.05
NaiveBayes	0.594 ± 0.035	0.82 ± 0.03
Boosted NaiveBayes	0.669 ± 0.048	0.70 ± 0.04

Table 3: Empirical Results for UnImportant Prediction

C4.5	0.5486 ± 0.02	0.89 ± 0.02
NaiveBayes	0.7075 ± 0.06	0.46 ± 0.12
Boosted NaiveBayes	0.6861 ± 0.041	0.65 ± 0.07

and which are positive, etc. We can similarly define Precision and Recall for Unimportant pages, as TrueNegative / PredictedAsNegative and TrueNegative / AllRealNegative respectively. The results, over all 10 CV folds, appear in Table 2, 3, in the form mean$\pm$standard-deviation.

Notice that Boosted NaiveBayes has the best "worst-case" over these 4 values, averaging around 65%.

7. CONCLUSION AND FUTURE WORK

This preliminary study attempts to determine which pages a user will find important. We first ran an Travel Planning experiment, producing over 15,000 annotated webpages from over 120 participants, and about 12% of URLs are labeled as important. Our subsequent analysis shows that this information is sufficient to learn a fairly accurate classifier, of around 65% precision and recall, for both classes of pages. This is the first step to building a recommendation system that will reliably help users reach the pages they need to satisfy their information needs.

Our final goal is to help users get where they want to go; this paper discusses the first step: just identifying these user-specific, task-specific important pages. And our next step is to find more efficient way to predict important pages for web user, including content and structure mining, trying other imbalance classifying algorithms, and also we want to conduct more experiment to collect more annotated web data for our research.

8. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of the 20th Int'l Conference on Very Large Databases (VLDB'94)*, Santiago, Chile, Sep 1994.

[2] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. of the Int'l Conference on Data Engineering (ICDE)*, Taipei, Taiwan, Mar 1995.

[3] Richard Duda and Peter Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[4] Rob Holte, Nathalie Japkowicz, Charles Ling, and Stan Matwin. *AAAI'2000 Workshop on Learning from Imbalanced Data Sets*. AAAI Press, 2000.

[5] Bamshad Mobasher, R. Cooley, and J.Srivastava. Creating adaptive web sites through usage-based clustering of urls. In *Proceedings of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'99)*, nov 1999.

[6] Bamshad Mobasher, R. Cooley, and J. Srivastava. Automatic personalization through web usage mining. Technical Report TR99-010, Department of Computer Science, Depaul University, 1999.

[7] Bamshad Mobasher, E. Han, G. Karypis, and V. Kumar. Clustering based on association rule hypergraphs. In *Proccedings of SIGMOD'97 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'97)*, May 1997.

[8] Maurice Mulvenna, Sarabjot Anand, and Alex Bchner. Personalization on the net using web mining: introduction. *Communications of ACM*, 43(8):122–125, Aug 2000.

[9] Mike Perkowitz and Oren Etzioni. Adaptive sites: Automatically learning from user access patterns. Technical Report UW-CSE-97-03-01, University of Washington, 1997.

[10] Mike Perkowitz and Oren Etzioni. Towards adaptive web sites: Conceptual framework and case study. *Artificial Intelligence*, 118(1-2), 2000.

[11] Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, 1992.

[12] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.

[13] Robert Schapire. Theoretical views of boosting and applications. In *Tenth International Conference on Algorithmic Learning Theory*, 1999.

[14] Ian Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, Oct 1999.