SANDIA REPORT

SAND97-2046 • UC-605

Unlimited Release

Printed August 1997

Final Report for LDRD Project "Learning Efficient Hypermedia Navigation"

RECEIVED

OCT 06 1997

OSTI

Pang Chen, Glenn Laguna

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia
Corporation, a Lockheed Martin Company, for the United States
Department of Energy under Contract DE-AC04-94AL85000.

Approved for public release; distribution is unlimited.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

SAND97-2046
Unlimited Release
Printed August 1997

Distribution
Category UC-605

Final Report for LDRD project “Learning Efficient Hypermedia Navigation”

Pang Chen
Algorithms and Discrete Math Department

Glenn Laguna
Intelligent Systems Principles Department

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1110

Abstract

This report documents the work performed under the Laboratory-Directed Research and Development (LDRD) grant “Learning Efficient Hypermedia Navigation.” The bulk of the work is contained in the software developed for the WWW and a copy of the software demonstrating its use has been submitted to the LDRD office.

1 Introduction

This project addresses the need for tools that help the user navigate through large hypermedia systems. Hypermedia is an approach to linking and traversing many forms of related computer-based information, such as text, video, audio, and computer graphics, in a non-linear fashion [3]. Unfortunately, as the body of information contained in hypermedia systems grows, it becomes increasingly difficult for users to find an efficient path through this information. This work illustrates the development of tools to help the user navigate efficiently through the large body of information contained in such systems.

Part of this work has been the development of new learning algorithms that use the actions of past users of the system to suggest hypermedia paths to the current user. By using learning techniques, the system will be able to adapt to new users throughout the life of the system. Ultimately, the system will use this same learning to physically alter the underlying structure of the hypermedia itself, allowing more efficient data access as well [1].

To test the algorithm, we have implemented a World Wide Web server that dynamically creates a path for a user to follow through the hypermedia database. The server gathers information about how previous users have navigated the database, and creates a path for each new user accordingly. We have tested this software on a moderately large hypermedia database, consisting of a 90 page brochure that contains text, pictures, and references to publications.

2 Learning algorithm

The new machine learning algorithm links keywords to user attributes to create probable initial search paths. More generally, the algorithm can be used for document classification. Given a sequence of examples [a set of document features (e.g., a particular word containment) and an associated action (e.g., read with great interest)], the learning algorithm produces a compact rule base composed of patterns (generalized document features) and (recommended) actions.

The algorithm learns the hidden pattern-action rule base incrementally by taking a bottom-up, pattern compression approach. Upon viewing a document, it first predicts the recommended action based on the current rule base with its own assessment of confidence, obtained through user feedback and reinforcement. If confidence is sufficiently high, it would execute the action without user intervention. Otherwise, it would request user approval and check the desired action against the predicted. Learning occurs when the two actions are different. In this case, the example is added into the list of surprises, and the rule base is modified by extracting from this list through the following compression steps: First, combine pairs of patterns (conjunctive presence of a set of document features) with the same action repeatedly through set intersection until no further generalization is possible without causing action conflicts (an example matching a pattern associated with a

DISCLAIMER

**Portions of this document may be illegible
in electronic image products. Images are
produced from the best available original
document.**

different action). Then, remove the pattern having the most successful matches recorded, and append the output rule base this pattern with its associated action.

This algorithm is better than the current state-of-the-art algorithms (ID3) [2] that are top-down, in that standard features (each standard feature corresponds to a set of document features) cannot be enumerated efficiently by standard algorithms, but are not required in advance by ours.

3 Algorithmic Framework

The basic concept of this project is to aid users in navigating through a web of information by suggesting relevant documents based on the declared attributes of the current user and the preferential knowledge learned from previous users. There are many approaches to represent and apply the knowledge of previous users. In our approach, the knowledge base consists of attributes-filter pairs

$$(a_1, a_2, \dots, a_l) \longleftrightarrow ((f_1, f_2, \dots, f_k), T),$$

with each filter component having a relevance count $(r_1, r_2, \dots, r_k)$ in decreasing order and total count T . Each attribute is a text string describing search context. Each filter component is a text string describing document feature.

There are two modes of information navigation: training and usage. During training, user is required to supply relevance scores to information accessed. These scores are then used to modify the corresponding filter and subsequent search during usage.

3.1 Usage Mode

In the system usage mode, the user enters his attributes $(a_1, \dots, a_l)$ and a keyword search K . In response, the system first searches the information base with K and highlights those documents with a match. To provide more focus onto the relevant documents, the system uses the user attributes and retrieve those associated document filters stored in the knowledge base. Using these filters, the system performs a secondary search on the highlighted documents, and mark those filtered documents having sufficiently high relevance counts (relative to the total count) with a different color, thereby indicating the system's recommendation. In other words, if D is a document that contains the keyword K , and if it also contain a feature f_i with r_i/T sufficiently large, then it will be deemed relevant and highlighted accordingly. In actual implementation, the searches are done using Unix `grep` to perform the single word matches. Of course, more sophisticated implementation can use other customized document filters to search based on semantics instead of pure syntactic patterns.

3.2 Training Mode

In the system training mode, the user is requested to score each document visited as either relevant or irrelevant. We use this discriminating information to either construct a filter to be associated with the user attributes of the trainer, or modify the associated filter if it already exists. In our simplified approach, we take those words in each relevant document and remove all words appeared in the irrelevant document, and obtain a feature vector $(d_1, \dots, d_m)$. We absorb this feature vector into the (possibly nonexistent) filter $(f_1, \dots, f_k)$ associated with the trainer attributes as follows:

1. For each d_i , if it is not in the filter, append it at the end of the filter if allowable (a priori size k not reached).
2. If it is in the filter as f_j , increment r_j and move f_j up in the sequence to maintain the decreasing relevance count order.
3. Increment total count T .

Thus, we are attempting to discriminate the relevant documents from the irrelevant ones using only single words of the documents as potential features. Of course, there are situations when this simple approach may not suffice. In these cases, a more complicated approach such as using two or three word phrases as potential document features may be required to yield a discriminating feature.

4 WWW Server — How the software works

The new World Wide Web server software dynamically creates a path for a user to follow through a hypertext document. It is based on the WWW server software from the University of Illinois, with extensions for determining the probable path that a user will take in navigating the data.

When a user enters the top level of the hypertext document, he is presented with a number of choices as to how he would classify himself (e.g. programmer, manager, secretary). Based on this selection, certain URL's in the document are highlighted. These correspond to the URL's that the user would most probably want to look at, based on his classification. He is not excluded from viewing other URL's in the document, but these will have been shown statistically to be less interesting to people in his classification. The highlighting is propagated throughout the document. If the user makes different selections other than the highlighted ones, the server records these choices, and the database is updated.

5 Demo — Example of Hypertext Navigation

We have tested the algorithm on a moderately large, multi-level hypertext document called "Dual- Use Intelligent Systems and Technologies." This is a hypertext version of a brochure

(SAND91-2525) that was published in 1991 describing the ISRC. The brochure is over 90 pages long, and includes text, graphics, and publication references. The accompanying tape includes a working demo of this system for an SGI Indigo computer.

To install the demo, do the following:

1. This demo is based on version 1.3 of the UIUC webserver software. It requires that PERL 5.001, or greater, be installed on your computer.
2. For the learning algorithm to work correctly, the webserver needs to be started by `inetd`, rather than as a stand-alone daemon.
3. The software is set up to install in the directory `/usr/local/etc/httpd`. In the server documentation, this is referred to as "`$basedir`". If you want to install it someplace else, all references to this directory need to be changed, however we have written an Install script to take care of this.
4. These are the steps to install:
 - (a) Untar the distribution file in your basedir.
 - (b) Run the Install script. It will change all of the references to `basedir` and also to your host.
 - (c) Re-make `httpd` by changing to the source dir and typing "`make sgi5`".
 - (d) Copy the `httpd` executable to wherever you want it.
 - (e) Edit `/etc/inetd.conf` and `/etc/services` to reflect the location and port number of `httpd`.
 - (f) Send a `SIGHUP` to `inetd`.
 - (g) Make a symbolic link needs to be made between `$basedir/user` and `/tmp/user`, i.e. `ln -s /tmp/user $basedir/user`.
 - (h) Try it out.

References

- [1] Chen, P., "Learning to Improve Path Planning Performance," *AI and Automation*, World Scientific, 1995.
- [2] Thornton, C.J., *Techniques in Computational Learning*, Chapman and Hall, 1992.
- [3] Woodhead, N., *Hypertext and Hypermedia*, Addison-Wesley, 1991.

DISTRIBUTION:

2	MS 1436	4523	C. E. Meyers
2	MS 1110	9223	P. Chen
2	MS1008	9621	G. Laguna
5	MS 0899	0899	Technical Library (5)
1	MS 9018	8940-2	Central Technical Files
2	MS 0619	12690	Review & Approval Desk (2) for DOE/OSTI