Web Personalisation with the Cover Coefficient Algorithm

Matthew Anderson, Irfan Altas, and Geoff Fellows

School of Information Studies
Charles Sturt University
Wagga Wagga, NSW, 2678
Australia
ialtas@csu.edu.au

Abstract. In this paper we discuss how to personalise web pages dynamically, based upon customer profiles generated from a click stream dataset using the cover coefficient algorithm. The personalisation model can be applied in an environment where there is a need to know the habits of customers that is beneficial to both the organisation and the web server administrator.

1 Introduction

Casselman [1] states that getting up close and personal with customers is also the goal of progressive companies operating in the online world. An online company attempts to imitate the closeness of a storeowner to a customer in the bricks and mortar world. An example of this is when you enter a store and you are recognised, the storeowner is aware that you have been there before. The storeowner knows your preferences and is then able to recommend products and a service based upon previous purchases and enquires.

Larger companies such as Amazon [2] and Yahoo [3] try to model this traditional retailer environment in an attempt to 'personalise the online experience'. The expectation is to have competitive advantages flowing from increased customer loyalty and retention rates by tailoring web site contents to suit their customers' desires.

They record click stream data from the hyper-text transfer protocol (HTTP) requests made to the web server. The web server log files store Internet Protocol (IP) address, cookies and the files requested [4,5]. The purpose of this exercise is to recognise patterns and models of navigational habits of users that are found in companies' click stream dataset. This information can be used to personalise web advertisements and web content and to promote new services that may be of interest to their customers. However, only few online companies are really effectively implementing personalisation today, mainly because of complexity and cost issues.

There are some common click stream models such as *click fact models* and *session fact models* [6]. These models are used to define profiles of web users' naviga-

P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2659, pp. 422–431, 2003.

tional habits, and these profiles are used to personalise the web documents. Although modelling the click stream data using the above models is possible, there is a need to try and gain an understanding of the customer's navigational habits in an automated way. This can be achieved by using data mining techniques on the click stream dataset.

Data mining systems have many techniques that help answer vague questions that would not normally be able to be answered by someone looking at the raw dataset. These techniques are broken up into studies that allow an implementation of a model to search the data in a specific way and output results. One technique includes a classification study that is a form of *supervised learning* and a clustering study that is *unsupervised learning* [7].

A classification technique classifies an item within a dataset into a predefined class [8]. The number of defined classes can be infinite and do not have to be in sequence. On the other hand a clustering technique processes rows of information that have similar trends and patterns in common and groups them together. These groups do not always contain unique values; it is often found that values can overlap across many clusters [9]. In this work, our aim is to implement a clustering model based on the cover coefficient technique (CC) [10]. The clusters obtained from this process can be used to recommend pages to a new user of a web site.

Since the early days of e-commerce, one of important goals is the development of recommendation based systems to automate the process of recommending products or services to web users. A successful recommendation system in an interactive environment is collaborative filtering that works in a comparative manner. It compares the similarity between users and their habits to make recommendations and predictions. Being able to create accurate predictions to be made from recommendation systems enables web personalisation to be more effective. One of such recommendation model is the top-N recommendation system [11]. In the top-N algorithm the main purpose is to identify a set of N items that will be of interest to a certain user. A top-N algorithm puts information into a binary matrix. From this matrix it can predicted whether a user will be interested in a particular item [12]. The benefits of using the top-N recommendation algorithm is that it shows that both cosine and probability-based schemes have a higher than average accurate recommendations compared with traditional collaborative filtering techniques [11]. Measuring similarities amongst documents are also extensively studied in the information retrieval context and referred to as resemblance coefficients [13]. Resemblance coefficients can be classified into four classes: distance, probabilistic, correlation and association coefficients.

The (probabilistic) CC technique can be used to measure similarities between documents in the information retrieval context. In this work, we implement the CC technique to measure similarities amongst web users by employing the click stream dataset collected from a web site. As a result of these similarity measurements it can be predicted whether a user will be interested in a particular web document and, hence, it can be used as a top-N recommendation model to personalise web documents. The CC technique can store input data from a click stream dataset in either a binary or numerical form. We discuss collecting click stream data in Section 2. The CC technique works from one particular item to calculate a similarity with all other items in the entire item collection where the entire collection includes the item itself.

Our aim in this work is to find out whether the CC concept can satisfactorily be implemented in web personalisation area. We discuss the CC technique in Section 3. Conclusions are presented in section 4.

2 Collecting Click Stream Data

We implemented the CC algorithm on a subset of web documents found on the Association of Societies for Growing Australian Plants (ASGAP) web site [14]. The web site contains information about the cultivation, propagation, conservation and appreciation of Australia's native flora. The structure of a subset of ASGAP web site is illustrated in Fig. 1. Each box in the Fig. 1 represents a web document while each arrow represents a link to the connected web document. Any page can be the entry point to the web site rather than just the homepage.

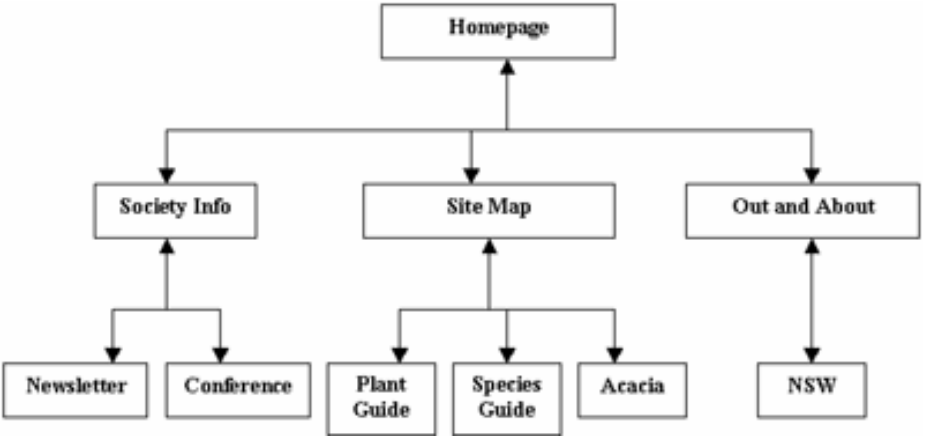

Fig. 1. A subset of the ASGAP web site [14]

The movement of a particular web user has been recorded in an access log file on the web server. An illustration from this log file for two users presented in the Fig. 2.

```
192.168.1.164.3000 GET /outAndAbout.html HTTP/1.0 [07Feb/2002:13:55:01 +100]
192.168.1.164.4000 GET /siteMap.html HTTP/1.0 [07/Feb/2002:13:40:08 +1100]
```

Fig. 2. Two user records from the click stream dataset

The meaning of the fields for the first record of Fig. 2 is as follows. The first field gives the host IP address rather than hostname to cut down the workload of the Domain Name Server (DNS) system. The second one contains the identification number of the web user that is stored in the cookie. The third field contains the HTTP request type made which includes the relative uniform resource locater (URL) where the file is located on the web site and the HTTP version. The last field is the date stamp for the file request event.

An important issue that can affect the click stream data is that some ISPs use proxy servers. Requests made to any resource on the World Wide Web are made via the proxy server. As a result all IP addresses logged in the web server access log file will be that of the proxy server rather than the individual web user, making multiple web users be viewed as one. This will decrease the value of the collected data. For example in Fig. 2 IP numbers are the same that may be assumed as a proxy server IP number.

To overcome the proxy problem above, cookies can be used. They store unique identifying number on the web user's hard disk drive, so upon return to a web site the web user's request will include this identification number. Figure 2 demonstrates this further. Even though both requests were made from the same IP address, 192.168.1.164, the cookie values, 3000 and 4000, identify individual web user.

3 The Cover Coefficient Technique and Web Personalisation

The CC identifies relationships between documents and users of a web site by use of a matrix. The CC technique works from one particular web user to calculate a probabilistic similarity measure with an entire web user collection including that particular user. The measure is the probability of randomly selecting a web document from one particular user, and from all the users containing that web document, the probability of randomly selecting a second particular user [10,3].

The CC technique creates a mxn matrix, say D, by using the click stream dataset. Its columns represent the web documents, (d_1, d_2, … d_n), and its rows represent users, (u_1, u_2, … u_m), who requested web documents. The entries of the matrix, D, simply indicate whether a user visited a particular document (1) or not (0) (see Fig. 3). It is possible to assign different meanings to the entries of the matrix such as the time spent on a particular web document by a particular user or how many times a user visited a particular web document. However, we work only binary values in this paper.

At the second stage of the CC algorithm, the D matrix is mapped into an m x m matrix, say C. The C matrix indicates the relationship between web users. Web user u_i contains n web documents (d_1, d_2 … d_n) and the probability of randomly selecting any one of these web documents, from all the web documents in the set, is s_i. Web document d_n is contained in m web users (u_1, u_2,…u_m) and the probability of randomly selecting any one of these web users, from all the web users in the set, is s'_n. The probability of selecting the same web documents from web users u_m and u_i, which is to say the extend to which web user u_i is covered by web user u_m, is therefore s_n multiplied

by s'_n [13]. The entries of the C matrix in Fig. 4 can be calculated by using this definition as is illustrated for c_{12}

$$c_{12} = \sum_{k-1}^{10} s_k s_k^{'} = (\frac{1}{5}\frac{1}{6}) + (\frac{1}{5}\frac{1}{4}) + (0\frac{1}{2}) + (0\,0) + ... + (\frac{1}{5}0) + (0\,0) \tag{1}$$

User ID	home	Society Info	News	Conf.	Site Map	Plant Guide	Species Guide	Acacia	Out & About	NSW	Total
1000	1	1	0	0	1	1	0	0	1	0	5
2000	1	1	1	0	1	0	1	0	0	0	5
3000	1	0	0	0	0	0	0	0	1	1	3
4000	0	0	0	0	1	1	0	1	0	0	3
5000	0	1	0	0	1	0	0	0	1	0	3
6000	1	0	0	0	0	0	0	0	1	0	2
7000	1	0	0	0	0	0	0	0	0	0	1
8000	1	1	0	1	1	1	0	0	0	0	5
Total	6	4	1	1	5	3	1	1	4	1	

Fig. 3. D matrix associating web users to web documents

Each entry in the C matrix indicates how well web user u_i covers web user u_j, including when web user u_i and web user u_j are the same web user. The diagonal terms of the matrix C are known as the decoupling coefficients. The diagonal term, c_{ii}, is the dissimilarity of u_i to all other users found in the collection. The coupling coefficient, calculated as (1- c_{ii}), is the indication to how similar the user u_i is to all other web users in the collection.

The decoupling coefficient is used to estimate the number of clusters to be created. The number of clusters can be calculated as

$$n_c = \sum_{i=1}^{n} c_{ii} \tag{2}$$

where m denotes the number of users in the collection. Thus, for this example the number of clusters is

$$n_c = 0.240 + 0.523 + 0.472 + 0.511 + 0.233 + 0.208 + 0.167 + 0.390 = 2.744 \approx 3$$

U. ID	1000	2000	3000	4000	5000	6000	7000	8000	Total
1000	0.24	0.123	0.083	0.107	0.140	0.083	0.083	0.19	1.00
2000	0.123	0.523	0.033	0.04	0.09	0.033	0.033	0.123	1.00
3000	0.139	0.055	0.472	0.0	0.083	0.139	0.56	0.056	1.00
4000	0.178	0.067	0.00	0.511	0.067	0.0	0.0	0.178	1.0
5000	0.233	0.150	0.083	0.066	0.233	0.083	0.0	0.15	1.0
6000	0.208	0.083	0.208	0.0	0.125	0.208	0.083	0.083	1.0
7000	0.167	0.167	0.167	0.0	0.0	0.167	0.167	0.167	1.0
8000	0.19	0.123	0.033	0.107	0.090	0.033	0.033	0.390	1.0

Fig. 4. C Matrix representing similarities of web users

Once the number of clusters is calculated the next step is to select a single user that will represent a single cluster. Such a representative user is called a seed. Calculating the seed powers using the following formula identifies the seeds of the clusters

$$p_i = c_{ii}(1 - c_{ii}) \sum_{j=1}^{n} d_{ij} \tag{3}$$

Where d_{ij} is the (i,j) entry of the matrix D. Thus, for the example the seed powers are

$$1000: \qquad 0.240 \text{ x } 0.760 \text{ x } 5 = 0.912$$

Similarly for 2000: 1.247; for 3000: 0.748; for 4000: 0.750; for 5000: 0.536; for 6000: 0.329; for 7000: 0.139; and for 8000: 1.190.

Then, three seed clusters would be the web user 2000 (1.247), the web user 8000 (1.190) and the web user 1000 (0.912) that are the first three web users with the largest seed powers. We refer them as the seed users. Using these seed users we can then distribute users to particular clusters. For example, the web user 3000 will be allocated to the seed user 1000 for the following reason. We identify the probabilistic cover coefficient numbers from the fourth row of the C matrix in Fig. 4 for the web user 3000 corresponding to each seed user. In this case, they are 0.139, 0.055 and 0.056 for the seed users 1000, 2000 and 8000, respectively. Amongst those numbers the largest probabilistic cover coefficient is 0.139 that corresponds to the seed user 1000. Therefore, the web user 3000 will be allocated to the cluster identified by the seed user 1000.

In some cases, the probabilistic cover coefficient values corresponding to the seed users may be equal. In that case, we allocate that user to the cluster with the least number of web users. For example, the user 7000 has the probabilistic cover coefficient value 0.167 corresponding to every seed user. Therefore, it is allocated to the cluster identified with the seed user 2000.

The three clusters for the example are given in Fig. 5. In implementations there is another cluster referred to as red bag cluster. Web users who request web documents that have not been requested by any other web users are allocated to the rag bag cluster due to unknown comparisons to any seed.

Cluster 1	Cluster 2	Cluster 3
1000	8000	2000
3000	4000	7000
5000		
6000		

Fig. 5. Cluster generation for the example

Once the clusters are generated they can be used for recommendations to the new web users entering to the web site. After a new user requests a few web documents they can be allocated to one of the clusters that have been previously created. Then, the system can dynamically present some recommended web documents (top-N recommendation) to the user according to the cluster, which the user was allocated to during their initial navigation of the web site.

The standard technique implemented on all recommendations is that the top-N most requested web documents by the users of that particular cluster be recommended to a new user allocated to that cluster. For example, if a new user is allocated to the cluster 1, then the new user may be recommended with the four most commonly requested web documents that were previously identified for this cluster. From Fig. 6 those will be “out & about”, “home”, “Society Info” and “Site Map”.

User ID	home	Society Info	News	Conf.	Site Map	Plant Guide	Species Guide	Acacia	Out & About	NSW
1000	1	1	0	0	1	1	0	0	1	0
3000	1	0	0	0	0	0	0	0	1	1
5000	0	1	0	0	1	0	0	0	1	0
6000	1	0	0	0	0	0	0	0	1	0
Total	3	2	0	0	2	1	0	0	4	1

Fig. 6. Frequencies of the web document requests for the cluster 1

The system initially creates a set of clusters from existing click stream data. It uses these clusters to assign new web users to a particular cluster until the number of new users reaches for a predefined number. The system re-clusters all web users as a

background job when the number of new users reaches to this predefined level. As soon as a new clustering of web users is available the system works with these new clusters. This is mainly to the time consuming nature of the clustering process. In our implementations we found little difference between real-time and delayed clustering approaches.

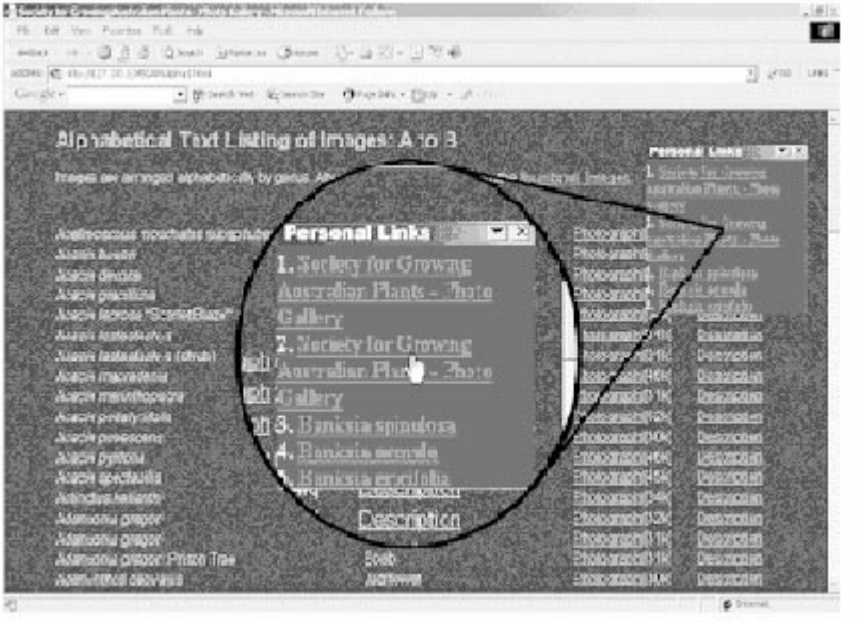

Fig. 7. Personalised web documents

This prototype model is implemented on the click stream data obtained from ASGAP web site. It is implemented on a Java Virtual Machine running on a Microsoft Windows 2000 platform. The web server used to handle the HTTP requests is Jakarta Tomcat version 4.1.12. The system runs as an application using Java to parse the web server log file. Web users and web documents are represented in memory as Java objects. The implementation details of the system can be found in [15].

Figure 7 is a screen snapshot from the system. It represents how the web document would be altered and presented to the web user. The windows application opens the document and parses it looking for the <TITLE> tag as this should be a very short description of the information within the web page. The title tags from the recommended web documents are then presented back to the user as links in a separate window as the top-5 recommendations.

4 Conclusions

In this work we developed a top-N recommendation system to dynamically introduce web documents to a web user of a site. The introduced web documents are predicted to be in the interest area of the user. The model uses the cover coefficient technique as the engine to measure the similarities amongst users of a web site by using the click stream dataset. According to those similarity measures the users of the web site are clustered. When a new web user is navigating the web site, the system assigns the user to one of the existing clusters according to the web documents that the user has already accessed. Then, the top-N most requested web documents by the users of that particular cluster will be recommended to this new user. The system is satisfactorily tested over a controlled dataset for which the number of clusters and member of clusters were manually created. Then, as a prototype it was implemented over a click stream dataset obtained from the ASGAP web site.

References

1. Casselman, G. 2001, *Web Personalization* (online). http://www.casselman.net/artlist/webpersonalization.htm [Accessed 22 Oct. 2002].
2. Amazon.com. 2002, Amazon.com Privacy Notice (online). http://www.amazon.com/exec/obidos/tg/browse/-/468496/104-2622634-4059946 [Accessed 16 May 20002].
3. Yahoo Inc. 2002, Yahoo! Privacy Policy (online). http://privacy.yahoo.com/ [Accessed 15 May 2002].
4. Keen, P. 1987, Information systems education: recommendations and implementation. Cambridge University Press, New York, USA.
5. Fielding, R., Gettys, J., Modul, J., Frystyk, H., Masinter, L., Leach, P. and Berner_Lee, T. 1999, Hypertext Transfer Protocol – HTTP/1.1 (online). http://www.w3.org/Protocols/rfc2616/rfc2616.txt [Accessed 24 June 2002].
6. Anderson, D. 2000, Personalizing: Port 80: Docks: Communities (online). http://riccistreet.net/port80/docks/personalizing.htm [Accessed 19 Sept. 2002].
7. Groth, R. 1998, Data Mining A Hands-on Approach for Business Professionals. Prentice Hall, New Jersey, USA.
8. Kurzeme, I. 1996, VICNET's users: a longitudinal market survey of the users of Victoria's network, VICNET. RMIT.
9. Hamilton, J., Gurak, E., Findlater, L. & Olive W. 2000, *The Virtuous Cycle of Data Mining* (online). http://www.cs.uregina.ca/~dbd/cs831/notes/virtuous_cycle/virtuous_cycle.html [Accessed 5 Apr. 2002].
10. Can, F. and Ozkarahan, E. 1990, Computation of Term/Document Discrimination Values by Use of the Cover Coefficient Concept, Journal of American Society for Information Science, 38(3), pp. 171–183.
11. Karypis, G. 2000, Analysis of Recommendation Algorithms for E-Commerce, in Proceedings of the 2nd ACM conference, ACM Press, pp. 158–167.
12. Yin, K. 1994, Case Study Research: Design and methods. 2nd edn. Sage, USA.
13. Lindley, D. 1997, Interactive Classification of Dynamic Document Collections. Phd Thesis, University of New South Wales, Australia.

14. ASGAP. 2002, The Society for Growing Australian Plants (online). http://farrer.riv.csu.edu.au/ASGAP [Accessed 30 June 2002].
15. Anderson, M. 2002, Effectiveness of Web Personalisation Using the Cover Coefficient Algorithm, Honours Thesis, School of Information Studies Charles Sturt University, Australia