Cold-Starting Collaborative Filtering Content Recommenders for Mobile Phones

Indika WG. Piyasena, and H. Anthony Chan
Department of Electrical Engineering, University of Cape Town, Rondebosch, South Africa
Email: indika@ieee.org; h.a.chan@ieee.org

***Abstract*— Mobile phones coupled with 3G technology can provide technologically impoverished communities access to web services. Such access, however, is impractical because mobile phone interfaces are cumbersome to use. Hierarchical menus and search engines pose an interaction barrier to the unfamiliar. In order to address both these issues, we design a content recommender system to recommend relevant content to the user. Collaborative filtering is a technique that passively gathers user preferences, identifies users with similar interests, and then makes predictions on unobserved items. A multitude of the algorithms have been developed and have been evaluated for predictive accuracy. Yet, they have not been evaluated in the cold-start scenarios with respect to how quickly these algorithms can identify user preferences. Two user-based collaborative filtering algorithms were evaluated empirically on the MovieLens dataset; Pearson correlation and vector similarity. Vector similarity convincingly outperforms Pearson correlation by identifying preferred items ten times faster. This conclusion is to be confirmed on other datasets. The success of the collaborative filtering algorithms is attributed to a large density of user ratings. Further research is to be required on how to ensure reliable recommendations without the availability of such data.**

***Index Terms*—Collaborative Filtering, Interface design Machine Learning, Mobile phones, Recommender Systems, User centric**

I. INTRODUCTION

THE proliferation of mobile phones throughout Africa can act as a bridge between the currently widening digital divide. These devices, coupled with GPRS or 3G technology, enable technology impoverished communities to access the Internet. While such a combination enables Internet access, access however is not pragmatic.

The interfaces on mobile phones make browsing the web challenging. Small screen sizes, cumbersome text input and the lack of a pointing device deter the exploration of the web [1]. In addition, web pages are filled with paraphernalia to make them more attractive. Large images, Adobe Flash, Ajax and JavaScript etc not only increase the size of web pages, thus increasing the cost of mobile data access, but are also not readily and quickly processed by the light-weight web browsers found on mobile phones.

These limitations have been addressed by the mobile web. The mobile web is a subset of the web that consists of text-based pages with minimal usage of graphics. Two interfaces are presented to the mobile phone user to access these pages:

Aggregators: Mobile web content is collected by content providers and is arranged within a hierarchical menu framework.

Mobile search engines: Content is retrieved from the mobile web from specified keywords.

Interactions with hierarchical interfaces require that the user stores in his or her short term memory a map of the hierarchical structure under which the content is organised [1]. While the addition of this cognitive overhead in easily adopted by users accustomed to file systems found on operating systems, it is met with difficulty to the inexperienced [2].

Small screen impacts, cumbersome text input and slow mobile data access heavily deter searching for content via the mobile device, often leaving the user feeling frustrated [1].

More fundamentally, both the hierarchical and the search based interfaces rely on the user knowing what one is looking for. A user experienced with the Internet is conscious of the content present on the web, and can speculate what may exist on the mobile web. Based on this knowledge, the user may have the motivation to endure these cumbersome interfaces. Those who are unaware of the content may lack such a motivation.

Ignorance of web content may be attributed to the lack of exposure and to the lack of relevance of such content. Web content originates from a socio-economic class of people who have access to fixed Internet connections. Such content may be irrelevant to technologically impoverished communities.

In summary, technology is a necessary, but insufficient criterion to provide Internet services. Both the interaction barrier and the lack of web relevance need addressing.

Manuscript received 15 April 2007. I. Piyasena (indika@ieee.org) et al are members of the Department of Electrical Engineering at the University of Cape Town. This work was supported in part by Telkom, Nokia Siemens Networks, TeleSciences and the National Research Foundation of South Africa, under the Broadband Center of Excellence program.

II. A Community-Centric Design

Syndication is a mechanism whereby a user publishes content to numerous media platforms simultaneously. Protocols such as Really Simple Syndication (RSS) that enable this mechanism have recently grown in popularity on the Internet. Extending the ability to syndicate via Short Message Service (SMS) or via mobile web page would encourage the generation of a web that is relevant to communities of people who only have access to these devices. Figure 1 depicts this mechanism.

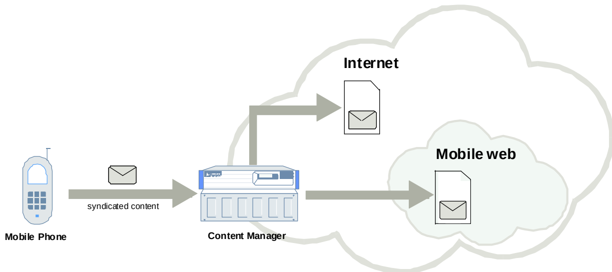

Fig. 1. Content can be syndicated via a mobile phone simultaneously onto the web and the mobile web, thus enabling the decentralisation of web content generation.

Likewise, only the content that is relevant to the mobile phone user is filtered, thereby reducing the sheer volume of content available. In addition, the relevant content can be recommended to the user, thus minimising the cumbersome interaction barrier. The software architecture proposed to realise this solution is presented in figure 2.

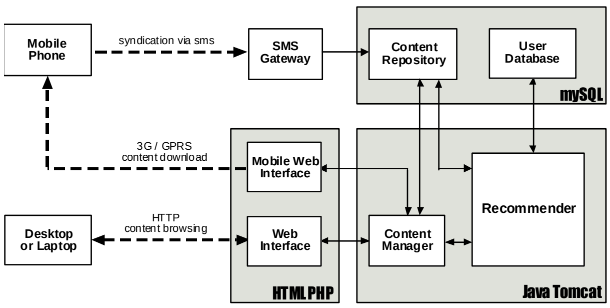

Fig. 2. The proposed software architecture enables syndication via SMS, mobile web content download via 3G / GPRS and access to the syndicated content via the standard web.

All the components in the above architecture are deterministic in their design, with exception in the recommender. This component is responsible for identifying the user's preferences and retrieving the relevant content from the content repository.

III. Related Work

The problem of identifying content on the mobile web that a particular user would prefer, lies in the domain of information filtering.

A. Content-based filtering

Viewing the problem from an idealistic perspective, if the user concerned had explicitly stated the mobile web content that he or she likes, the recommender would simply have to retrieve those resources for the user. Suppose that less information was provided, such as a list of keywords representing the user's interests, then content-based filtering [3] techniques such as Latent Semantic Indexing [4] can be applied. This is a search based method, and requires that the user explicitly states what he or she is looking for. Inputting keywords requires interactions on these devices, which one would like to eliminate, thus is not a suitable approach.

B. Collaborative filtering

The other approach to filter information is through collaborative filtering [5]. With this technique similarities in different users, based on similarities in their preferences, are identified. A simple, but highly visible example of this notion is: users who are interested in content X, are also likely to be interested in content Y. This method has two definite advantages over content-based filtering:

Information is implicitly collected: Interactions of the user with the web content can be observed passively, such as counting page visits. The user's preference to those resources can be inferred from the page count. Such a means of obtaining preference is non-intrusive, thus minimises interactions with the mobile device.

Recommendations made on human judgments: The relationship between similar users and similar resources are based purely on the human interactions between them, and are not subject to algorithms that attempt to understand what the resources represent.

1) User-based collaborative filtering

User-based collaborative filtering developed by GroupLens [6] directly computes a pair-wise comparison of the preferences between users, thus identifying similar users. A nearest neighbourhood [5] [7] around each individual is constructed. The individuals around the primary user's neighbourhood are like-minded individuals. Content that is preferred by those individuals in the neighbourhood are recommended back to the primary user. Despite being the most accurate algorithm to date [7], it requires computations that grow both with the number of users and the number of resources; hence it not highly scalable.

2) Model-based collaborative filtering

Model-based approaches [5] are the other approach to collaborative filtering. Instead of computing pair wise comparisons between user's preferences, these preferences are used to create a web of similarity between the resources. If the primary user has a strong preference towards a particular resource, very similar resources will be recommended to the user. Bayesian clustering and Bayesian network models provide the mathematical basis for such implementations [5]. An advantage these techniques have over user-based collaborative filtering is that a static model is constructed, and therefore response times to users requests are not only low, but are deterministic.

IV. Significance of Problem

Recommender systems have been designed since the 90's beginning with Tapestry [8]. They have been used successfully by e-commerce websites to recommend products to customers, such as www.amazon.com [9].

The application of a recommender in the proposed mobile environment differs from web-based recommender services in three ways:

Recommendations are the primary aspect of the service

The applications of recommenders in web-based services have been a secondary function of these environments. If a poor recommendation is made, the user would simply ignore it.

In the mobile phone environment, the recommender takes a primary, active role in the application taking full responsibly of the usability of the application. Thus, a robust recommender is to be designed that has little tolerance to poor recommendations.

Preferences are implicitly collected

Web services often require that the user explicitly rate items of preference before recommendations are made. Such a process can act as a deterrent. Hence, preference information is collected purely implicitly.

All preferred items are recommended

The motivation behind the design of recommenders has been from the e-commerce industry to recommend items for sale. Thus, only undiscovered items are presented to the user. The design of the recommender for the mobile environment requires that both discovered and un-discovered items are presented to the user.

A multitude of user-based and model-based collaborative filtering algorithms have been developed. An empirical analysis of some these algorithms with respect to their accuracy has been conducted [5]. Results have shown that their predictive accuracy averages at 60% with a variance of 4%. These tests have been conducted on datasets with an excess of 10,000 user preferences.

A 60% accuracy rate is adequate for such an application. It is uncertain how quickly such an accuracy rate can be obtained. Hence, an investigation is conducted of the collaborative filtering algorithms during cold-start.

V. Construction of Model

Let community C be a group of n individuals, each denoted by P_i. Associated with this community, is a pool of m resources, each denoted by R_j. A particular individual is isolated from the community and is referred to as the active user and is denoted by P_a. Figure 3 depicts these constructions.

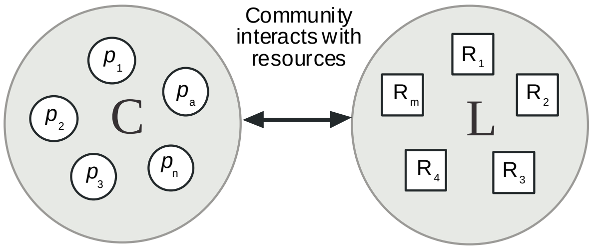

Fig. 3. A community of individuals is associated with a pool of resources. One of the individuals from the community is defined as the active user for whom resources will be recommended to.

The active user is assumed to engage with the application for the lifespan of s sessions. A session refers to the process of the user logging into the application, receiving recommended content, making observations of the content and finally logging off from the application.

For the active user, two $m \times s$ matrices are created, $\mathbf{R}$ and $\mathbf{O}$. These respectively store the recommended items and the observed items at each session s.

Individual P_i's preference towards resource R_j is encoded by a scalar $v_{i,j}$ where $0 \leq v_{i,j} \leq 1$. This preference $v_{i,j}$ is determined by taking the ratio of the number of times the resources were observed by the user, given the number of times they were recommended. Specifically,

$$v_{i,j} = \sum_{k=1}^{s} \frac{\mathbf{O}_i(j,k)}{\mathbf{R}_i(j,k)} \quad \exists \quad \mathbf{R}_i(j,k) \neq 0$$

A set of resources are picked randomly and are presented to the user. The user observes all, a selection, or none of these items. Based on these observations, it is the task of the collaborative filtering process to identify what other items the active user is likely to observe.

Scalability is not an issue during cold-start, hence user-based collaborative filtering [5] is considered for evaluation. In this method, preferences of unobserved items are predicted from the preferences of similar users. Specifically,

$$p_{a,j} = \overline{v}_a + \kappa \sum_{i=1}^{n} w(a,i)(v_{i,j} - \overline{v}_i)$$

where $p_{a,j}$ is the preference the active user has towards some item j. κ is the normalising factor such that the absolute values of the weights sum to unity. The mean is necessary to be factored in due to different individuals engaging with the application more than others. It is specifically,

$$\overline{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j}$$

The weights $w(a,i)$ represent similarity between users, based on their preferences, and can be calculated using two methods. The first method evaluated is the Pearson correlation. It is amongst the most popular and accurate memory-based schemes [6]. The correlation indicates the strength of the linear relationship between the preferences between each user. It is calculated as follows:

$$w(a,i) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - \bar{v}_i)^2}}$$

The second method of obtaining the similarity between users is with vector similarity. This method has been used in the comparison of documents for the purpose of information retrieval. Here the cosine of the angle between the user-item preference vectors indicates similarity.

$$w(a,i) = \sum_j \frac{v_{a,j}}{\sqrt{\sum_{k \in I_a} v_{a,k}^2}} \frac{v_{i,j}}{\sqrt{\sum_{k \in I_i} v_{i,k}^2}}$$

The preference $p_{a,j}$ is calculated for all j items in the pool. Finally, these calculated preferences of items are ranked, and the top r items are recommended back to the active user.

VI. Evaluation Method

The purpose of evaluation is to determine the feasibility of employing user-based collaborative filtering for the proposed recommender system. The reason for recommending content is to address the usability issues presented by the mobile phone interface. While the gratification of the recommended content cannot be measured directly, it can however be determined indirectly by measuring the content's usage. This is simply the ratio of the number of items observed with the number of items recommended. Specifically, for session s, a utility metric U_s is defined as:

$$U_s = \sum_{k=1}^{m} \frac{\mathbf{O}_i(k,s)}{\mathbf{R}_i(k,s)}$$

Due to this project being in its design phase, no usage data is available to the author. Hence, the performance of the algorithms is measured empirically in Matlab.

Human behaviour is modelled using a dataset from the GroupLens project [6] called MovieLens. This dataset consists of ratings individuals have made on movies; in particular, 80,000 ratings are made by 943 users on 1682 movies.

The ratings of movies are used to represent individual preferences towards resources. These ratings per individual are translated into a probability mass function. The ratings govern how likely the individual will, or will not observe a recommended resource. Such a probabilistic function is necessary to model the nature of an individual that although has a stronger preference towards some item A, may sometimes observe item B instead.

The evaluation method is depicted in figure 4.

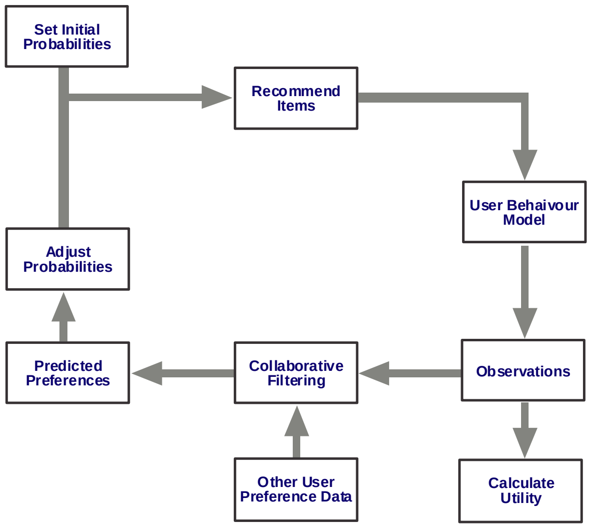

Fig. 4. Evaluation method. Each cycle represents a session: where a user logs in, views recommended content and logs off.

A user is removed from the dataset and is assigned as the active user. The initial probabilities that this user will observe a particular resource are each assigned to m^{-1} (where m is the number of resources). A set of resources is chosen from the resource pool and recommended to this user. Based on the constructed behaviour model (the PMF generated from all the votes this user has made), a set of observations from the recommended items are noted. These observations mark the end of the session the user has had with the application, and the utility of the session is calculated. The observations are fed into the collaborative filtering algorithm, along with observations from the other users in the dataset. The outcome of the collaborative filtering process is a set of predicted preferences towards unobserved resources. These preferences are used to adjust the user's preference probabilities. In the session to follow, the resources associated with higher preference probabilities are more likely to be recommended. This entire process is repeated 50 times to observe how the utility of the session improves given more user behaviour information.

Both the Pearson correlation and the vector similarity methods are evaluated. In addition, a simple preference learning method is implemented. This method simply increases the probability that an observed resource will be recommended, and makes no use of observations from other users. This method is implemented to benchmark the collaborative filtering algorithms.

The evaluation method is probabilistic in nature, thus the results of the evaluation are subject to a high variance. To obtain a better approximation of the behaviour of the algorithms, the experiment for each individual is repeated ten times and the average utility per session is extracted.

VII. Empirical Results

An experiment was performed where the 943 users were interacting with 1682 resources. Twenty-five items were recommended during each session. Figure 5 represents the results of this experiment.

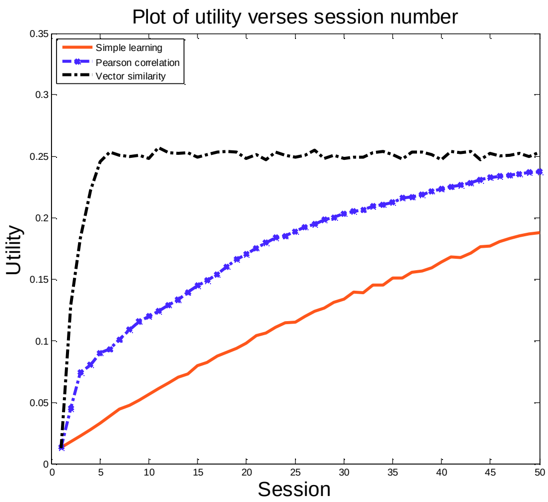

Fig. 5. Three preference learning algorithms are evaluated against each other in terms of the rate at which they identify user's preferences. These are in ascending order of performance: Simple learning, Pearson correlation and Vector similarity.

The simple learning function appears to grow linearly, approximately learning one item per session. The Pearson correlation method improves the rate at which preferred items are identified, which clearly shows that the use of collaborative filtering is advantageous. Vector similarity remarkably out performs the Pearson correlation method in terms of the rate at which user preferences are identified – in less than five sessions, the user's preferences are identified.

The maximum average utility obtained was 0.25. This value is well below one because the experiment was averaged over a wide range of users – some of which had rated less than five items.

VIII. Conclusions and Future Work

Applying collaborative filtering clearly shows an improvement in the rate at which preferred content is identified.

Vector similarity remarkably outperformed the Pearson correlation technique with respect to the rate at which user preferences were identified. This conclusion is to be reconfirmed on different user-preference datasets.

The accuracy reached is attributed to the large amount of ratings present for the collaborative filtering algorithms to operate on. In a pure cold-start situation, such data is unavailable to the designer. A possible means to ensure that a certain accuracy rate is guaranteed is by constraining the diversity of the available resources. As more preference data becomes available, the constraints can then be relaxed according some relationship.

Alternatively, users can be stereotyped and their preferences towards particular items can be predisposed according to these stereotypes. This would ensure a certain accuracy rate to begin with. Their particular preferences can then be identified using the standard user-based collaborative filtering.

References

[1] M. Jones and G. Marsden, *Mobile Interaction Design*. West Sussex, England: John Wiley and Sons, 2006.

[2] K. Sullivan, "The Windows 95 user interface: a case study in usability engineering," in CHI '96: *Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM Press, 1996, pp. 473–480.

[3] R. J. Mooney and L. Roy, "Content-based book recommending using learning for text categorization," in *Proceedings of DL-00, 5th ACM Conference on Digital Libraries. San Antonio*, US: ACM Press, New York, US, 2000, pp. 195–204.

[4] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society of Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[5] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. Of UAI*, 1998, pp. 43–52.

[6] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," in *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*. New York, NY, USA: ACM Press, 1994, pp. 175–186.

[7] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in SIGIR '99: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM Press, 1999, pp. 230–237.

[8] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "*Using collaborative filtering to weave an information tapestry*," Commun. ACM, vol. 35, no. 12, pp. 61–70, 1992.

[9] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

Indika Piyasena received his B.Eng. from The University of Cape Town (UCT), South Africa in 2005, and then spent time in industry designing radio direction finding systems. He then began his M.Sc. research at UCT focusing on

machine learning algorithms and mobile phone interface design.

H. Anthony Chan (M'94–SM'95) received his PhD in physics at University of Maryland, College Park in 1982 and then continued post-doctorate research there in basic science. After joining the former AT&T Bell Labs in 1986, his work moved to industry-oriented research in areas of interconnection, electronic packaging, reliability, and assembly in manufacturing, and then moved again to network management, network architecture and standards for both wireless and wireline networks. He designed the Wireless section of the year 2000 state-of-the-art Network Operation Center in AT&T. He was the AT&T delegate in several standards work groups under 3rd generation partnership program (3GPP). During 2001-2003, he was visiting Endowed Pinson Chair Professor in Networking at San Jose State University. In 2004, he joined University of Cape Town as professor in the Department of Electrical Engineering. Prof. Chan is Administrative Vice President of the IEEE CPMT Society and has chaired or served numerous technical committees and conferences. He is a distinguished speaker of the IEEE CPMT Society and has been in the speaker list of the IEEE Reliability Society since 1997.