ELECTRONIC WORKSHOPS IN COMPUTING

Series edited by Professor C.J. van Rijsbergen

Jonathan Furner, School of Information and Media Studies, and David Harper, School of Computer and Mathematical Studies, The Robert Gordon University, Aberdeen, Scotland. (Eds)

Information Retrieval Research

Proceedings of the 19th Annual BCS-IRSG Colloquium on IR Research,
Aberdeen, Scotland, 8-9 April 1997

Paper:

User Profiling Techniques: A Critical Review

S. Stewart and J. Davies

Published in collaboration with the
British Computer Society

User Profiling Techniques : A Critical Review

Abstract

The key to effective information filtering is user profiling - the ability to represent and reason about the interests or preferences of a user. This paper briefly surveys techniques for filtering the vast amount of information that is now available on the Internet and other electronic sources, in a variety of media. The non-explicit social information filtering approach is then examined in more detail and a new algorithm proposed which compares favourably with earlier approaches.

Scott Stewart and John Davies

Information Access Research

MLB 1/14,

BT Laboratories

Ipswich IP5 7RE

UK

scott.stewart@bt-sys.bt.co.uk

john.davies@bt-sys.bt.co.uk

1. Introduction

There is a large amount of information available on the Internet, and users have access to all of this information. The need has arisen for an automated method for selecting the information that is passed on. This is normally achieved by the use of an agent which filters the material that it presents to its user. The information presented to the user is typically selected using a profile of each user, which reflects their particular interests. This paper examines the three main categories of technique employed for user profiling. Those categories are :

- Statistical Term-based approaches - well established techniques from the information retrieval field.
- Artificial Intelligence, (AI), and Neural Networks, (NN) - learning information agents that adapt the user profile as they are used.
- Social Filtering - this approach enables a user to filter the information that they receive based upon the ratings given by other users in the system, rather than an explicit analysis of the content of the information.

We look at each category in turn, with particular emphasis on novel social filtering techniques. We present a new social filtering algorithm and compare it's efficiency and effectiveness to other approaches.

2. Statistical Term Based Approaches

2.1. Term Matching

Term Weighting Techniques

Term matching is one of the simplest ways of determining whether some information is relevant to a user's interests. A common method for implementing a term matching scheme is to use a Vector Space Model. Such a model constructs a vector of dimension m for each document, where m is the total number of terms used to identify the content of the documents in the system. Three main approaches exist for deriving the vector representation of a particular document, the first two of which are well known techniques. The approaches are Boolean, Term Frequency and TF-IDF, (which is described in more detail below). For the latter two methods, normalisation of the resulting vector is common to prevent bias towards longer documents.

The TF-IDF Technique

The TF-IDF technique [1] is one of the most successful in use, in which the weighting given to a term depends on the Term Frequency, (TF), and the Inverse Document Frequency, (IDF). The TF is self explanatory and the IDF can be thought of as a measure of how often the term appears in *all* of the documents in the retrieval system. Essentially common terms have a low IDF and terms that appear in only a few documents will have a high IDF. The weighting for any particular term in the document is then given by the product of the TF and the IDF.

Similarity Calculation

The similarity between any two documents may be calculated as the cosine product of the associated vectors. Alternatively, a user's existing term profile can be mapped into the vector space and then the similarity evaluated. In this way the most relevant documents for a user can be determined and retrieved.

Evaluation

This approach is relatively straightforward and is a well established technique, backed up by extensive information retrieval research.

However, there are limitations. Issues of synonymy, (having many ways to express a given concept), and polysemy, (terms having multiple meanings), can lead to a failure in detecting *relevant* documents and to the incorrect selection of *irrelevant* documents. Fortunately, the problem of synonymy can be reduced with the aid of profile enhancement [8] and the effects of polysemy can be limited with the application of linguistic techniques. Several studies have investigated the use of word sense disambiguation to address the polysemy problem but results have been mixed, (see, for example, [9, 10]).

The term matching approach assumes that all terms are orthogonal in vector space, (i.e. unrelated). This is not a correct assumption, due to the problem of synonymy. In addition, related *concepts* are also not orthogonal. Another significant disadvantage, which applies to all of the techniques examined in this section, is the restriction to the filtering of text documents, (or, at least, items with assigned text attributes).

2.2. Latent Semantic Indexing, (LSI)

Singular Value Decomposition

The basic term matching approach to information retrieval described above relies upon the assumption that the terms are orthogonal in vector space. While this is satisfactory for an initial assumption it does not hold in reality. LSI seeks to resolve this problem by examining the 'latent' structure of a document and the terms within it. A technique known as Singular Value Decomposition, (SVD), takes a large 'word by document' matrix and reduces it to k orthogonal dimensions, where k is typically between 100 and 300. (The matrix is compiled from the individual document vectors in the system). The mathematics behind this reduction is rather complex, but standard routines are available to perform the calculations. It is sufficient to say that SVD is very similar to eigenvector decomposition [6]. A good example of the benefits LSI can be found at [11].

Evaluation

LSI suffers from the limitation to text documents, as with term matching. However, unlike term matching, the technique of latent semantic indexing attempts to retrieve relevant documents, even if they have no words in common with the profile. This is achieved with the use of SVD and is an enhancement of the term matching approach.

3. Artificial Intelligence and Neural Networks

3.1. Introduction

Various attempts have been made to incorporate Artificial Intelligence, (AI), and Neural Network, (NN), techniques into user profiling. The following sections, (3.2 and 3.3), briefly present two typical examples of the use of such techniques.

3.2. Learning Information Agents

Outline

These agents learn the preferences of their users by using feedback to update the user profiles. The example to be considered here is that of an agent attempting to support the browsing of the World Wide Web[2]. This involves searching the web for pages that might be of interest, (perhaps based on the user's profile), applying a filter to the results, and then presenting only the top p pages to the user. The user then gives feedback on the relevance or interest of each page presented and this feedback is used to update the user profile.

Document Vectors and User Profiling

The normalised TF-IDF weighting system is used, (see section 1.2), to create a vector, V_i , for the new document. User profiles are also maintained and take the form of a vector, M , (of the same type as V_i), where the weight of each term corresponds to the user's preferences. A document is then scored by simply taking the dot product $V_i \cdot M$.

Learning Algorithm

The information supplied to the learning part of the system consists of a selection of documents and the associated evaluations given by the user. The learning algorithm uses this information to update the weights in the user's profile. A decay function may also be applied

so that terms that have not appeared in recent documents deemed relevant by the user will start to lose their weighting. This may be useful for matching changes in the user's tastes over time.

A good way to initialise the user profile is to ask a new user to decide whether each of a selection of existing documents is relevant or not. If a relevance threshold is used to limit the range documents that are presented to the user then it is more effective to start with a relatively low threshold, and increase it as the system learns more about that user. This ensures that the user gets to view pages that may normally be classified as irrelevant and has the chance to indicate their interest if appropriate.

The weights can be updated by a simple addition process. Assume the user gives an evaluation e_i to page i . Then the new user vector, (i.e. profile), $\mathbf{M}^+$, becomes :

$$\mathbf{M}^+ = \mathbf{M} + \sum_{i=1}^p (e_i \times \mathbf{V}_i)$$

(Where p is the number of pages evaluated.) This update is applied for every document that is rated by the user.

3.3. Neural Networks

User Profiling

Neural Network agents also learn the interests of the user with continued use of the system. A user profile consists of a neural network which represents the concepts which are important to that user. The network is based on terms which occur in the documents that are accepted and those that are rejected by the user. The network also models the associations between words, so that related words are related in the network. For example, if the user has read many documents related to “telecommunications” and a new document uses the terms “telephone communications” then there is a high probability that the association will be made by the network [3]. This occurs because the terms “telecommunications”, “telephone” and “communications” will have relatively higher interconnection weights, hence indicating the association. These high interconnection weights are the result of the terms appearing together in several of the documents processed by the network.

fig..1 - A section of an example neural network

Learning

When a user accepts a document as relevant the significant terms are extracted, (e.g. by using TF-IDF), and can be added to the network for a more comprehensive profile, (or suggested to the user for approval of the addition). This technique can also be used to initialise the network. Alternatively, the user may be prompted to manually enter a selection of terms that represent their interest.

Scoring of Documents

When a document is presented to the neural network, the key terms are extracted and the outcome evaluated using standard neural network techniques. This outcome, in general, is taken as the degree of association between the article and the user's profile of interests.

3.4. Evaluation of AI and NN Techniques

The learning information agent and neural network approaches to user profiling can both cope automatically with changes in the user's interests, although this is at the cost of the user having to rate pages that they have read. The neural network approach has an advantage over the learning information agent in this respect since the user does not have to decide on a score for the page - they simply accept or reject each document that is presented to them. This is quicker and more convenient.

The two techniques examined in this section share two main disadvantages. Primarily, neither approach can deal with items that are non-text based. Secondly, extra processing time is required to maintain the user profiles. This is probably more significant in the neural network model, where the updating calculations are likely to be more complex. The option exists to update the user profiles off-line, providing that the delay in profile update is acceptable.

4. Social Filtering

4.1. General Approach

Introduction

The filtering approaches examined so far in this paper have all been based on the use of keywords or terms found in documents and user profiles, (or the use of textual indexes for non-text items)t. These approaches can be described as content-based filtering. There are two main disadvantages associated with all content-based filtering techniques. Firstly the items that are being filtered must be text documents or some other form that can be parsed by a computer. Hence items such as sounds, media, art, and so on cannot be easily filtered with these systems. Secondly, content-based filtering has no inherent method for presenting the user with items that they did not know that they liked, since the filtering only selects items that fit the defined user profile.

‘Social filtering’ is an attempt to overcome these problems. It enables a user to filter the information that they receive based upon the ratings given by other users in the system. This often relies upon collaborative agents where the user agents work together to produce recommendations. Generally, social filtering automates the process of ‘word-of-mouth’ recommendations but uses the ratings of many users to make these recommendations more accurate to a particular user’s tastes.

Social Filtering can also lead to the recommendation of items in a category that the user has not previously declared as interesting or relevant. This can arise when another user, (or users), with similar tastes (in other categories) gives a high rating an item in this new category.

User Profiling

A user's profile is simply the set of ratings that they have given to a selection of items from the system database. As can be expected, the greater the number of ratings in a user's profile, the better match the system can make to their tastes and hence the more accurate the recommendations will be for that user.

Application

Filtering of an item is then based upon feedback from other users who have already rated that item, either implicitly or explicitly. Implicit rating may be as simple as sending a reply to a news article. Explicit feedback involves the user actively giving an item a score based on their opinion of it.

The use of the feedback then depends on its type. For *implicit* Boolean feedback, it could be, "Show me all news articles to which Fred Smith made a reply." Alternatively, when the feedback involves *explicit* ratings, an ordered list of recommendations with predicted scores may be generated.

Actual applications of social filtering include :

- Ringo [4]
- MORSE [5]
- Tapestry [4]
- videos@bellcore.com. [4]

Several other applications can be found in the literature [4, 5].

4.2. Algorithms

Outline

Several algorithms have been applied to social filtering systems: this paper will examine four of them. We have implemented and compared the following social filtering algorithms :

- Average Score
- Mean Squared Difference
- Pearson r Algorithm
- Personal Recommendation Agent, **ProRate**, (an extension of MORSE)

The purpose of the exercise was to determine the method which produced the best results for the application in hand. A brief description of each method is given below, followed by a quantitative comparison of the accuracy and time performances.

Test Data

In order to evaluate the performance of each method, a realistic test data set was required. This was extracted from the MORSE movie recommendation system and consisted of the scores, (between 0 and 10), that each user had given to the films they had rated. At the time of testing there were 589 films in the database, (each rated at least once), with 626 registered users.

This testing was carried out by predicting ratings for films that the user had already rated, and then calculating the root mean square, (rms), error between the predicted score and the score given by the user. In addition to calculating the rms error for each user, the overall error was determined for the entire user set.

Average Score

The average score algorithm sets the predicted rating for a particular film to the average rating given by all users to that film. This is a very simple and fast technique, but is an ideal benchmark, against which the other algorithms can be tested.

Mean Squared Difference

This technique calculates the mean squared difference, MSD, between the current user's profile and the profiles of all other users in the system. Any users with an MSD below a certain threshold, L , are deemed to be within a neighbourhood of the current user. Each user, j , is then given a weighting, w_j , which is proportional to their similarity with the current user's profile. This could be calculated as follows :

$$w_j = \frac{L - MSD_j}{L}$$

The predicted score for any film is then the weighted average of the scores that the neighbouring users gave to that film.

Pearson r Coefficient

The Pearson r correlation coefficient can be taken as a measure of similarity between two user profiles. If the rating given by user i to item k is s_{ik} , and the mean rating given by i is $\bar{s}_i$, then the correlation between users i and j is :

$$r_{ij} = \frac{\sum_k (s_{ik} - \bar{s}_i)(s_{jk} - \bar{s}_j)}{\sqrt{\sum_k (s_{ik} - \bar{s}_i)^2 \times \sum_k (s_{jk} - \bar{s}_j)^2}}$$

The value of r is between -1 and +1 and a negative value of r indicates a negative correlation, whereas a positive value indicates a positive correlation, i.e. a similarity. The greater the magnitude of r , the greater the similarity, (if positive), or the dissimilarity, (if negative), of the two profiles. As with the MSD algorithm, a weighted average of ratings is then taken to produce a predicted score.

Personal Recommendation Agent

The final method to be considered here is an extension of the algorithm used in the MORSE movie recommendation system [5], known as ProRate. The algorithm has been adjusted to increase the speed of recommendations, with a negligible reduction in accuracy. In most cases there is no difference between the MORSE recommendations and those from ProRate.

To estimate the rating that user i would give film k :

Calculate the correlation between k and every other film, l . The correlation between two films is calculated using the Pearson formula, by comparing the ratings given by all the users to each of the two films, (disregarding the cases where a user has only seen one of the two films). The formula is used in the form :

$$\frac{F \sum s_{jk} s_{jl} - \sum s_{jk} \sum s_{jl}}{\sqrt{\left(F \sum s_{jk}^2 - \left(\sum s_{jk} \right)^2 \right) \left(F \sum s_{jl}^2 - \left(\sum s_{jl} \right)^2 \right)}}$$

Where :

F is the total number of films in the system.

s_{jk} is the score given by user j to film k

s_{jl} is the score given by user j to film l

Then :

For every other user j do :

Determine the ratings given by j to the N films most closely correlated with k .

Determine the correlation between the scores given by i and j .

(using the Pearson formula)

End For

Extrapolate estimated rating for film k by user i .

N was varied to find the best results and finally set to 10.

This algorithm generates a hypothetical user with exactly the same tastes as the current user and can hence give a rating for all films in the database. However, if an insufficient number of ratings have been given for a film then it is not recommended at all, since the results are not statistically accurate. That threshold was set to 10 ratings per film.

Accuracy

The overall rms error for the results from each method was calculated, (as described at the beginning of this section), and these errors are shown in the following table.

Average Score	2.016
Mean Squared Difference	1.594
Pearson r Coefficient	1.559
ProRate	1.404

As expected, the average score method produced the least accurate results, but it's purpose was to act as a benchmark, against which to compare the other results. The mean squared difference and Pearson r Coefficient algorithms produced very similar rms error levels, but it is clear that ProRate offers the lowest error of all of the methods considered.

Time Performance

The following graph shows the time taken to recommend the top 10 films for 3 users. User A had rated 215 films, user B had rated 21 films and user C had yet to rate some films. Note that under normal operation with ProRate, when a user requests a recommendation, but has not rated any items in the database, the average score method is used as a substitute. None of the social filtering algorithms produce meaningful results if a user has not rated any items. Hence the times for user C are hypothetical.

Clearly the mean squared difference approach is the fastest overall, (excluding the average score algorithm), but it does not produce results with the same accuracy as the ProRate algorithm.

Evaluation

In terms of the effectiveness of the different social filtering algorithms, the MORSE algorithm gave the most accurate predictions for the test set. However, the ProRate method produced results which were virtually identical to the MORSE recommendations in a fraction of the time, as can be seen in the graph above. Whilst the Pearson and Mean Squared Difference algorithms maintained reasonable overall RMS errors in very good time, the ProRate system had a lower overall error and produced recommendations in a similar time scale to these two and hence is the most effective.

As mentioned above, social filtering offers several advantages over the content-based filtering methods that have been described in this paper. Clearly, it is not limited to text documents, and can be easily applied to any user rateable item. The accuracy of the predictions is also increased as the user profile is extended. This profile extension can arise from the user explicitly adding more ratings or by feedback from the recommendations presented to them. Another key advantage of social filtering over traditional methods is that it may generate recommendations for items that a user did not know they wanted.

The most serious disadvantage is that the algorithms need a relatively large number of ratings from several users before they can be used effectively. This requires a community of users and a certain amount of time from each user to initialise their profile, (by giving the ratings). In addition, the user may have to rate *new* items that they have ‘seen’ in order to maintain an accurate profile. However this is a small price to pay for the benefits that each user would reap from the system.

5. Conclusion

5.1. Current Technology

This paper has considered three types of information filtering and user profiling : statistical term based methods, artificial intelligence and neural network term based techniques and non-explicit social filtering approaches.

Currently, social filtering is the only method available when dealing with items that cannot be parsed by a computer, such as films, books and so on. This, however, does not mean that the approach is limited to such items. There is no reason why text documents cannot be rated by users. In fact, a system known as Tapestry allows users to filter e-mail or news articles based on the users who replied to each document [7].

The statistical term based approaches are probably the most common, since they are relatively straightforward to implement. There has been a lot of research into methods for improving the performance of the simple term matching techniques and very efficient algorithms have been developed. The other two methods, (particularly Social Filtering), are relatively new and hence have not been subject to the same amount of research. The statistical approaches can be used to filter text based documents simply and effectively, without paying the price of large processing time requirements.

The remaining category of learning agents is really an extension of the statistical methods. The applications for techniques in this category are the same as those for normal term methods, although the leaning element inherent in these agents leads to a more accurate and up to date user profile being maintain. This in turn supports more accurate filtering. However, the learning process takes processing time, (the magnitude of which varies with

technique), and the approaches examined in this paper do not lend themselves to certain efficiency improving techniques, (such as Profile Indexing).

In summary, social filtering is particularly appropriate where non-textual media are under consideration and where there are a sufficient number of participating users to make the results statistically significant. Statistical approaches are based on well established research from the Information Retrieval community and are probably the best method for text based media, (or media that can be tagged with text in some way). In addition we have examined a number of longer term research areas, including the use of AI and NN techniques. As these techniques mature, they will offer new, smarter ways to model, learn and filter a users interests.

5.2. Future Work

In terms of the continuation of this work, the main emphasis for the authors will be on social filtering. There are some additional aspects to be taken into consideration, in particular how to model the *changes* in a user's interests. When relying upon social opinion as a basis for the generation and maintenance of the user profile, certain factors such as fashions and fads become relevant. It is in this area that future research and investigations are likely to be made.

6. Bibliography

1. Donna Harman

Ranking Algorithms

National Institute of Standards and Technology

Paper in “Information Retrieval : Data Structures and Algorithms”

Editors : William B.Frakes and Ricardo Baeza-Yates,

Prentice-Hall, 1992.

2. Marko Balabanović and Yoav Shoham

Learning Information Retrieval Agents : Experiments with Automated Web Browsing

Department of Computer Science, Stanford University, CA 94305

3. Andrew Jennings and Hideyuki Higuchi

A Personal News Service based on a User Model Neural Network

Kansai Advanced Research Centre, Communications Research Laboratory,

Owaoka, Kobe, Japan

4. Upendra Shardanand

Social Information Filtering for Music Recommendation, (section 2), 1994

M.I.T. Media Laboratory, Learning and Common Sense Group,

Technical Report 94-04

5. Donald Fisk

Recommending Films Using Social Filtering, (section 4.3), 1995

6. Peter W. Foltz

Using Latent Semantic Indexing for Information Filtering, (section 1.3), 1990

<http://www-psych.nmsu.edu/~pfoltz/cois/filtering-cois.html>

7. David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry

Using Collaborative Filtering to Weave an Information Tapestry

Communications of the ACM / December 1992 / Vol.35 No.12

8. John Davies, Richard Weeks, Mike Revett and Andrew McGrath

Using Clustering in a WWW Information Agent, March 1996

Proceedings of the 18th British Computer Society Information Retrieval Specialist

Group Annual Colloquium on Information Retrieval Research

<http://www.labs.bt.com/jasper/html/jasclus2.htm>

9. Sanderson M.

Word Sense Disambiguation and Information Retrieval

Proceedings of the 17th ACM-SIGIR Conference, 1994

10. Sussna, M.

Word Sense Disambiguation for Free-text Indexing Using a Massive Semantic Network

Proceedings of the Second International Conference on Information and

Knowledge Base Management, 1993

11. EOSDIS Core System (ECS) Technology Assessment

Latent Semantic Indexing

http://newsroom.gsfc.nasa.gov/eval/lsi_sb.html