<table>
<thead>
<tr>
<th>Title</th>
<th>Passive Profiling and Collaborative Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authors(s)</td>
<td>Rafter, Rachael; Bradley, Keith; Smyth, Barry</td>
</tr>
<tr>
<td>Publication date</td>
<td>1999-09</td>
</tr>
<tr>
<td>Conference details</td>
<td>The 10th Irish Conference on Artificial Intelligence and Cognitive Science (AICS 99), Cork, Ireland, September, 1999</td>
</tr>
<tr>
<td>Item record/more information</td>
<td><a href="http://hdl.handle.net/10197/4637">http://hdl.handle.net/10197/4637</a></td>
</tr>
</tbody>
</table>
Passive Profiling and Collaborative Recommendation

Rachael Rafter, Keith Bradley, Barry Smyth,
Department of Computer Science
University College Dublin
Belfield, Dublin 4, Ireland

1 Introduction

Online recruitment services have rapidly become one of the most popular types of application on the World-Wide Web. The award-winning JobFinder site (www.jobfinder.ie) is a good example. It allows users to browse and search a large database of current jobs from a variety of employment sectors. However, like many similar Internet applications JobFinder has a number of shortcomings, due mainly to its reliance on traditional database technology and client-pull information access models, which place the burden of navigation and search on the user. To address these shortcomings the CASPER (Case-based Agency: Skill Profiling & Electronic Recruitment) project seeks to investigate the role of Artificial Intelligence techniques such as Automated Collaborative Filtering (ACF) and Case-Based Reasoning (CBR) as a means of providing a more proactive, personalised and intelligent model of information access.

In this paper, we will examine the role of ACF within the CASPER project. Briefly, collaborative filtering is a feature-less recommendation technique that selects information items to recommend for a particular user on the grounds that other similar users have previously expressed an interest in these items [1,2]. In this sense ACF implements a computational model of a familiar social recommendation policy. However, the success of ACF depends critically on the availability of high-quality user profiles, which contain graded lists of information items, and the technique is shown to work well when there is a high expected overlap between related profiles. However, JobFinder’s large dynamic database of jobs, each with a limited life-span, means that the expected overlap between similar users can be vanishingly small [see also 2]. In this paper we will discuss how this characteristic presents a serious problem to one common form of collaborative filtering and we describe the solution being investigated in CASPER.

2 User Profiling in CASPER

User profiles are the primary source of knowledge in an ACF system. In CASPER, user profiles are automatically and passively constructed by mining JobFinder’s server logs for the relevant information. One of the central objectives is the construction of user profiles without interfering with the way that users interact with JobFinder. Hence, unlike many related systems CASPER does not require explicit graded feedback from users to construct its user profiles.
Figure 1 shows a small part of a JobFinder server-log. Each line records a single job access by a user and encodes details such as the time of the access and the job and user ids. The basic form of a user profile is simply a list of jobs that a given user has looked at over the course of their entire interaction with the JobFinder system. However, a simple list of job ids will not provide a very accurate picture of a user’s preferences. In general, a given user is likely to have different levels of interest in different jobs and this implicit relevancy information can be added to the profile representation in a number of ways. CASPER collects three types of relevancy information.

1) **Revisit Data:** The number of times that a user clicks on a job is thought to be an indicator of their interest in that job, in the sense that users will often return for a second or third read of an interesting job description while they are unlikely to revisit uninteresting jobs after an initial read. For this reason CASPER logs the number of times that each user clicks on a job – a thresholding technique is used to eliminated so-called “irritation clicks” where a user repeatedly clicks on a job while waiting for a description to download.

2) **Read Time Data:** Coarse-grained revisit data can be augmented with a more fine-grained measure of relevancy obtained from read-time data [3]. The time a user spends reading a job description should correlate with that user’s degree of interest. For this reason, CASPER also calculates read-time information from the server-logs by noting the time difference between successive requests by the same user. Again, a suitable thresholding technique is used to eliminate spurious read-times due to a user logging off or leaving her terminal.

3) **Activity Data:** The final and perhaps most reliable way to judge user interest in a particular job is to make use of JobFinder’s online application or email facility. Briefly, a JobFinder user can either email a job description to themselves, for later consideration, or apply for the job directly. Both actions indicate serious interest and both are noted in the CASPER profiles.

In our research so far we have examined each of these factors as a measure of relevance. However, to date we have not found any significant improvements over the simple ungraded profile
representation. It seems that unlike other domains [4], simple measures of clicks or readtimes do not act as accurate relevancy indicators and we believe that a more complex measure that combines clicks, readtime and activity data is needed – however, this is beyond the scope of this paper.

3 ACF and the Problem with Relationships

ACF is a two-step process. First, a set of users that are related to the target user are identified, and second, profile items from the related users, that are not present in the target profile, are ranked for recommendation to the target user. Thus, the critical idea behind ACF is that users develop implicit relationships as they interact with a system over time, and by recognising these relationships it is possible to make targeted recommendations. In the JobFinder system users look at job descriptions – similar users look at many of the same jobs – thus the nature of user relationships in JobFinder is the degree of overlap between two user profiles. Obviously, the ability to recognise and exploit relationships between users is the key to successful ACF. In general, there are two basic ACF approaches, one based on the measurement of direct user relationships, and the other based on indirect relationships.

3.1 Direct Relationships & Nearest Neighbours

In the simplest form of ACF, users are related on the basis of a direct similarity between their profiles, for example, by measuring the degree of overlap between their profile items, or by measuring the correlation coefficient between their gradings lists [2, 6]. This leads to a lazy (in the machine learning sense) form of ACF whereby the target user is used to select the k nearest profiles. Currently CASPER uses a simple overlap metric (Def. 1) to determine profile similarity.

**Def. 1:** \( \text{Overlap}(t, p) = \frac{|\text{Items}(t) \cap \text{Items}(p)|}{|\text{Items}(t) \cup \text{Items}(p)|} \)

**Def. 2:** \( \text{Quality}(j, t, P) = \sum_{\forall p_i : j \in P} \text{Overlap}(t, p_i) \)

{where t and p are profiles, t being the target profile, and j is a job id}

Once the k nearest users (base profiles) have been identified their recommendable items (jobs that are not present in the target profile) are ranked according to the metric shown in Def. 2. This measure biases jobs in two ways. First, jobs that occur more frequently in the base profiles are preferred over jobs that occur less frequently, and second, a job that occurs in a profile that is very similar to the target is preferred over a job that occurs in a less similar base profile.

3.2 Indirect Relationships & Virtual Communities

The above lazy form of ACF suffers from a reliance on direct user relationships. In many sparse-profile domains, such as CASPER’s, the expected overlap (similarity) between two user profiles is likely to be vanishingly small and thus the number of base users with significant overlap is also likely to be very low [2, 5]. As a result, target user recommendations may be based on a very small
number of base profiles with very low measures of target similarity. Fortunately there is an alternative ACF method that relies on the detection of indirect user relationships. For example, consider users A, B, and C, all of whom have similar job tastes. User A has looked at jobs j1, j2, j3, and j4, user B has looked at j3, j4, j5, and j6, and user C has looked at j5, j6, j7 and j8. There is clearly a direct (overlap) relationship between users A and B, and also between users B and C, but no direct relationship between users A and C (they share no jobs). Using lazy ACF user B could be considered as a base profile for user A. However, this form of ACF ignores the clear, albeit indirect, relationship between users A and C. If this type of indirect relationship can be exploited then the potential for ACF in sparse profile spaces is greatly enhanced.

The solution is to make use of (eager) profile clustering techniques to group users prior to recommendation – profiles are clustered into virtual communities such that all of the users in a given community are related. In order to specifically exploit the above form of indirect relationship the single-link clustering technique can be used with a thresholded version of the similarity metric from Def 1; essentially each community is a maximal set of users such that every user has a similarity value greater than the threshold with at least one other community member. Recommendations can then proceed in a way that is analogous to the nearest-neighbour approach, except that instead of selecting k neighbours for the target profile, we select the members of the target profile’s community. Since it’s no longer possible to judge the direct similarity between all pairs of profiles in a community (there may be no direct relationship between member pairs) we must grade items for recommendation by their frequency in the community only (Def. 3).

\[
\text{Def. 3 } \text{Quality}(j,P) = \frac{|\{p \in P : p \text{ contains } j\}|}{|P|} \quad \text{\{where } j \text{ is a job id and } P \text{ is a community of profiles}\}
\]

4 Experimental Analysis

At this point, we have described two basic collaborative recommendation strategies for use in CASPER. In this section, we describe a preliminary evaluation to test the quality of CASPER’s ACF component. Specifically we evaluate the quality of the job recommendations produced by each flavour of ACF, the lazy, k nearest neighbour version and the eager, cluster-based approach.

4.1 Experimental Setup

The experimental study is based on the user profiles generated from server logs between 2/6/98 and 22/9/98. These logs contained a total of 8248 job access from 5132 different users. These profiles spanned a total of 8035 unique jobs with an average profile size of approximately 14 jobs and nearly 3000 profiles containing less than 10 jobs – and indication of CASPER’s extremely sparse profile space.
4.2 Experimental Method

Unfortunately, we had no way of automatically evaluating the recommendations produced by the two ACF versions. Instead the evaluation had to be carried out by hand and for this reason we restricted our evaluation to a small set of users. Ten target users were selected, each from a different virtual community. Furthermore, the communities to which these target users belong were chosen to cover a range of different community sizes (from small to large). For each target user, we produced two recommendation lists containing ten jobs:

1. **ACF-NN**: The list of jobs recommended according to the nearest-neighbour version of ACF. That is, each target user is associated with its k nearest users (k=10 in this experiment) and a ranked recommendation list based on the equation in Def 2 is produced.

2. **ACF-Cluster**: The list of jobs recommended according to the cluster-based ACF approach. That is, each target user is recommended the most frequent jobs in its virtual community.

Both sets of results for each target user are then manually graded as good, satisfactory, or poor (mapped on to a numeric value of 3, 2, or 1 respectively), based on how similar the recommended jobs were to the existing jobs in each target user profile – for these preliminary experiments the grading was performed by the researchers involved in the project, a more elaborate evaluation is planned and will include a range of different graders that are not associated with the CASPER project. Therefore, every target user receives a cumulative grading score across the 10 recommended jobs from each ACF technique.

![Recommendation Quality](image)

Figure 2. Recommendation quality for ACF-NN and ACF-Cluster techniques.

4.3 Results

Each grading score is normalised by dividing by the maximum cumulative grade of 30 and presented in Figure 2 for each target user. Figure 2 also encodes a measure of cluster size so that we can see how the recommendation quality behaves for different cluster sizes using the cluster-based ACF method. It is clear from the results that the cluster-based method is outperforming the nearest-
neighbour version for all target users (except those that hail from the smallest virtual communities). We believe that the reason has to do with the sparseness of the profile space. As explained in Section 3, the expected overlap between users is very small and therefore many of the 10 nearest users for a given target chosen by ACF-NN may exhibit only very low degrees of similarity to that target. In fact, many of these nearest users may overlap with the target profile by just 2 or 3 jobs and therefore may not constitute a reliable recommendation partner for the target user. This means that we can expect part of the recommendations for the ACF-NN method to come from unreliable recommendation sources. In contrast, the ACF-Cluster method is basing its recommendation on potentially more reliable measures of indirect relationships between profiles. In the case of this experiment the similarity threshold used to construct the virtual communities was set at 10 and therefore implicit relationship are based on a transitive overlap of 10 between community members. However, the ACF-Cluster method does show quality degradation for small cluster sizes. Again, this is to be expected as these small clusters do not provide a rich enough recommendation source for their target users.

5 Conclusions

The CASPER collaborative filtering component has been described, showing how personalised recommendations can be formed using two different ACF approaches, one based on a k nearest neighbour technique and another based on an explicit clustering technique. We have argued that because of the sparseness of CASPER’s profile space presents significant problems for the memory-based k nearest neighbour approach to ACF, basically because the expected overlap between profiles is likely to be very low. However, one of the benefits of the cluster-based ACF technique is that it can exploit transitive overlap relationships between profiles that otherwise do not overlap. We have argued that this property makes the cluster based technique especially appropriate in the case of CASPER. This hypothesis has been backed up by some preliminary experimental results and we expect to add a more comprehensive evaluation in the near future.

4 References

