<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Towards a Domain Analysis Methodology for Collaborative Filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authors(s)</strong></td>
<td>Rafter, Rachael; Smyth, Barry</td>
</tr>
<tr>
<td><strong>Publication date</strong></td>
<td>2001-04-04</td>
</tr>
<tr>
<td><strong>Conference details</strong></td>
<td>23rd BCS-IRSG European Colloquium on Information Retrieval Research (ECIR 01), Darmstadt, Germany, 4-6 April, 2001</td>
</tr>
<tr>
<td><strong>Item record/more information</strong></td>
<td><a href="http://hdl.handle.net/10197/4635">http://hdl.handle.net/10197/4635</a></td>
</tr>
</tbody>
</table>
Towards a Domain Analysis Methodology for Collaborative Filtering

Rachael Rafter
Smart Media Institute, Dept. Computer Science, University College Dublin,
Dublin 4, Ireland

Barry Smyth
Smart Media Institute, Dept. Computer Science, University College Dublin,
Dublin 4, Ireland
{rachael.rafter, barry.smyth}@ucd.ie

Abstract
Collaborative filtering has the ability to make personalised information recommendations in the absence of rich content meta-data, relying instead on collations between the preferences of similar users. However, it depends largely on there being sufficient overlap between the profiles of similar users, and its accuracy is compromised in sparse domains with little profile overlap. We describe an extensive analysis that investigates key domain characteristics that are vital to collaborative filtering. We then explain how knowledge of these characteristics has helped to drive the design of a collaborative recommender system for the JobFinder online recruitment service.

1 Introduction
The information overload problem is one of the single most important barriers to the ongoing and future success of the internet and its associated information sources. As the information revolution continues at a pace it is becoming more and more difficult for users to locate the right information at the right time in the right way. For example, current search engine technologies go only part of the way to helping users in their search for relevant information. Recently, researchers have begun to explore new types of information filtering systems, called recommender systems [2, 5, 12], which combine ideas from information retrieval, artificial intelligence, and user profiling, in an attempt to deliver more proactive and personalised information services. One example is the CASPER project, which is designed to provide a personalised online recruitment service [3, 8, 9, 10]. Specifically, CASPER utilises collaborative filtering technology as one of its core personalised information filtering techniques. Collaborative filtering is a powerful and popular filtering technology for recommender systems as it has the ability to make personalised information recommendations in the absence of rich content meta-data, relying instead on collations between the preferences of communities of similar users.

Collaborative filtering technologies in general rely fundamentally on a large database of user profiles, where each user’s preferences are represented as a list of graded content items [10]. The ability of a collaborative filtering system to make accurate recommendations to a target user depends largely on there being sufficient overlap between the profiles of related users. In particular, the accuracy of collaborative filtering is compromised in domains where there is sparse profile coverage and little or no profile overlap. Given this state of affairs, it is perhaps somewhat surprising that previous research on this topic has tended to ignore the domain factors that can influence the success or failure of the collaborative filtering system. For example, previous work rarely documents the coverage and overlap characteristics of a given profiling domain.

In this paper we describe research carried out as part of the CASPER project into the development of a collaborative recommender system for the JobFinder online recruitment service (http://www.jobfinder.ie) 1. The main contribution

1JobFinder has now been bought by StepStone - http://www.jobfinder.ie now points to http://www.stepstone.ie
Towards a Domain Analysis Methodology for Collaborative Filtering

of this research is a better understanding of the previous results we reported from applying collaborative filtering to the JobFinder system, [8, 9]. We describe part of an extensive domain analysis that has been carried out in CASPER in an attempt to fine-tune its recommendation technologies. Specifically we will describe certain key domain characteristics that are vital to any collaborative filtering endeavour and explain how knowledge of these characteristics has helped to drive CASPER’s collaborative filtering component. Our concern here is not so much the end solution we arrived at for CASPER, but rather the path that lead to it. We are interested in the identification of domain features, for example the size of the profiles in the domain and the nature of the overlap between them, that influence the collaborative filtering mechanism, and the understanding of the impact that these features can have. Although we believe that such domain analysis must be performed within related research endeavours, it has not to our knowledge, been reported.

The remainder of this paper is organised as follows. In Section 2, we provide some background information for our research. In Section 3, we describe a comprehensive analysis of the domain in CASPER, focussing on those features that are particularly important for the success of collaborative recommendation techniques. In Section 4, we reason about the finding of the domain analysis and illustrate how these results have driven the design of the collaborative filtering mechanism in CASPER. In Section 5 we give some directions for further research, and finally, in Section 6 we present some conclusions.

2 Background

In this section we provide some background for the research in this paper. We describe the JobFinder application and why it is a suitable test-bed for the CASPER project. We then provide a brief description of collaborative filtering.

2.1 JobFinder

Online recruitment services have rapidly become one of the most popular types of application on the World Wide Web, (WWW). The award-winning JobFinder site is a good example. Agencies and employers can submit job advertisements which are indexed and stored in the JobFinder database, and this in turn is browsed by users who can search for current jobs in 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 leave the burden of actively searching for relevant information and updates on the user. For example, if a user is interested in Java programming jobs in Galway she must search through the job database for the relevant jobs, and she must perform this task regularly to ensure she sees all relevant job updates, as the set of jobs available is always changing.

Information content is impersonal in JobFinder; each user is presented with the same information regardless of her particular preferences. For example, two users may both submit queries for contract, technical support jobs with two years experience, and both will receive the same results. However these users may have different implicit job preferences, for example one user is only interested in contract work in Dublin, while the other user is willing to travel abroad if the job specification suits her. Therefore, the two users should receive different job information that is personalised to suit their individual preferences. This represents a common scenario where different users have different preferences even though they are not evident from the queries submitted. It is often the case that users only have a rough idea of what they want before they search through what is on offer, and it is common that certain preferences are not represented in the query submitted.

These shortcomings makes JobFinder an ideal candidate to benefit from techniques like collaborative filtering that take advantage of the user’s preference information. The application of collaborative filtering technologies to JobFinder is also interesting because JobFinder is an existing application and the collaborative filtering mechanism must fit into the current system. This is in contrast to many existing recommender systems which have been built specifically for the purpose of recommendation, for example the PTV system [13, 14, 15], the Ringo system [12] or NewsDude [2].
2.2 Collaborative Filtering

The core of CASPER’s recommender system is a (automated) collaborative filtering engine. Essentially, collaborative filtering is a recommendation strategy that bases its recommendations on similarities between users. Thus collaborative recommendation is content-free, since the process relies on information about users, rather than the actual information items at hand. Collaborative filtering recommends new items to a user based on what similar users have liked in the past. In contrast, methods like case-based reasoning for example, are content-based recommendation strategies - a user is recommended similar items to those she liked in the past, [3, 4, 14]. Collaborative recommendation therefore relies on similarity assessments between users rather than between the content of items. The CASPER project investigates the use of both types of recommendation strategies in JobFinder. However, the full details of these strategies in the context of CASPER is beyond the scope of this paper; the interested reader should refer to [9] where we describe each of these strategies and how the similarity assessments are made in CASPER in detail.

Basically, collaborative filtering can be seen as a three-step process. A set of user profiles is created and maintained as a knowledge base where each profile contains a given user’s preferences for items (see [10] for a full description of the user profiling component in CASPER). At recommendation time a group of users that are related (similar) to the target user are identified. Finally, the profile items from these related users, that are not already present in the target profile, are ranked for recommendation to the target user. The collaborative filtering process in CASPER is described in more detail in [8].

3 Domain Analysis

In this section we focus our attention on the domain in CASPER and analyse features of it that are important for the success of collaborative filtering. The core of successful collaborative filtering is rooted in its ability to predict or recommend new items to a user based on her similarity with other users, that is, based on whether she shares the same taste in those items with other users. This is usually accomplished by comparing the user’s history (contained in her user profile) of preferring certain (previously seen) items over others. It is therefore essential that there exist across the user community a certain degree of overlapping items between different user profiles on which to base the similarity calculations. A lack of any significant overlap between users implies a lack of similarity relationships between users and renders the collaborative filtering technology virtually useless.

Briefly, user profiles in CASPER are implicitly compiled by mining the relevant information from the original raw JobFinder server logs and therefore do not require the user to actively engage in the process. The details of the profiling techniques used in CASPER is beyond the scope of this paper, but the interested reader is referred to [10] where a full description is given.

For collaborative filtering to work effectively as a recommendation strategy in a given domain, certain domain features must be present:

- Many rich user profiles (Section 3.2.1).
- Significant overlap between profiles (Section 3.2.2).
- Significant coverage of items by profiles (Section 3.3.1).
- Significant distribution of items across the profile space (Section 3.3.2).
- Items that discriminate well between profiles (Section 3.3.2).

Obviously the exact nature of what constitutes ‘significant’ overlap etc. will change from system to system. In any given system there will be a threshold that denotes the smallest level of ‘useful’ overlap for example; that is, the minimum reliable overlap measure. For example, consider a domain where the average overlap between any random pair of profiles is .05 i.e. the profiles overlap by 5% of their full profiles. It is unlikely that this level of overlap is enough to guarantee that calculations of user similarity based on it will be reliable. By its very nature collaborative filtering relies on there being a strong level of overlap between profiles so that there is enough information on which to base similarity assessments.
Towards a Domain Analysis Methodology for Collaborative Filtering

<table>
<thead>
<tr>
<th>no. of profiles</th>
<th>5132</th>
</tr>
</thead>
<tbody>
<tr>
<td>no. of unique jobs</td>
<td>8235</td>
</tr>
<tr>
<td>no. of job accesses</td>
<td>112,824</td>
</tr>
</tbody>
</table>

Figure 1: Summary Statistics

We believe that this analysis is an important part of our research, which previous studies have failed to provide. Our intention is to provide a sound basis for further research by statistically examining CASPER’s domain. We feel this analysis is especially important for the application of collaborative filtering where domain features like those examined in this section lie at the heart of its success.

3.1 Summary Analysis

Our analysis begins by looking at the CASPER profile space as a whole and examining the number of users and jobs within it. If the ratio of jobs to profiles is too large, the likelihood of finding overlapping jobs between profiles decreases (so-called sparsity problem). However, if the ratio of jobs to profiles is too small, there will be fewer items actually left to recommend, (that is, items that the user has not already seen). Furthermore, there will be insufficient deviation between the likes or dislikes of individual users (which is based on the jobs they have looked at) to differentiate between individual tastes by examining their history. For example, in a movie recommendation task there would be no value in noting that all or most of the users, liked the film "Star Wars", because this does not help to individuate users.

As a test-bed for the CASPER project a section of the JobFinder server log-file dating from 2/6/98 to 22/9/98 was taken and used to create a user population for CASPER. This log section contained some 233,011 lines - each corresponding to a single user interaction with the JobFinder site, for example, a user clicking on a job title to read its description (job access), and 112,824 of these provided the basis of the population, (some lines were discarded because they were missing necessary information). In total 5132 user profiles were produced, and these spanned 8235 unique jobs, see Figure 1.

In CASPER’s domain, the proportion of jobs to profiles is very low, a ratio of 1.6:1 items (jobs) to profiles, in comparison to other systems; for example, the EachMovie collaborative filtering data set which has been used by others for experimentation (for example, [1]) has a ratio of items (movies) to profiles of 9:2, over 2,811,983 accesses. Although this suggests that there should be plenty of overlap between profiles in CASPER, we find that CASPER actually suffers from a very sparse profile space because the mean profile size is very low - 14.6 compared to 38.5 in EachMovie, rather than because of the ratio of jobs to profiles. Later, we will explain how this sparsity is linked to the perishable nature of jobs in CASPER.

3.2 Profile Analysis

We begin our analysis by examining the profiles in the CASPER domain. In this analysis we look at the size and the nature of overlap between profiles. For the following experiments, six groups of profiles \( G_i \) are produced for \( i = \{100, 250, 500, 1000, 2000, 5000\} \), each containing the largest \( i \) profiles from the original 5132 profiles.

3.2.1 Profile Size Experiment

We begin by looking at the size of profiles across the different profile groups \( G_i \), that is, the number of different jobs in the user’s history. It has been well documented that profiles need to be of an adequate size for successful collaborative filtering - if they are too small there is not enough information to learn anything about the user. The extreme of this is commonly referred to as the cold-start problem in collaborative filtering where the technique cannot function without sufficient user information, [11]. In this case, when new users start to use the system they have no history so their tastes cannot be predicted and subsequently no recommendations can be made. This experiment

\[ \text{publicly available and obtainable from Compaq Systems Research Center [6]} \]
examines the size of profiles in the different sized groups of the largest profiles ($G_i$), to provide an indication of the range of profile sizes available.

**Results:** The results are shown in Figures 2 and 3. Figure 2 tabulates the size values for each $G_i$, and and Figure 3 shows a graph of average profile size vs. group size.

<table>
<thead>
<tr>
<th>Group</th>
<th>Max Size</th>
<th>Min Size</th>
<th>Average Size</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_{100}$</td>
<td>415</td>
<td>79</td>
<td>118</td>
<td>49.23</td>
</tr>
<tr>
<td>$G_{250}$</td>
<td>415</td>
<td>53</td>
<td>86</td>
<td>41.32</td>
</tr>
<tr>
<td>$G_{500}$</td>
<td>415</td>
<td>36</td>
<td>64</td>
<td>36.36</td>
</tr>
<tr>
<td>$G_{1000}$</td>
<td>415</td>
<td>21</td>
<td>45</td>
<td>32.02</td>
</tr>
<tr>
<td>$G_{2000}$</td>
<td>415</td>
<td>11</td>
<td>30</td>
<td>27.44</td>
</tr>
<tr>
<td>$G_{5000}$</td>
<td>415</td>
<td>1</td>
<td>14</td>
<td>21.66</td>
</tr>
</tbody>
</table>

**Discussion:** These results show that the mean size of profiles decreases with increasing profile group size, (as expected given the experimental setup). We also see that while there is a substantial difference between the largest profiles and the smallest profiles, on average the size of profiles tends to be small, and the maximum size results are due to a small number of large outliers. In the group of the 100 largest users ($G_{100}$) each user has only looked at 1.4% of the total 8235 jobs on average, and this decreases to a mere 0.17% when we consider 5000 users ($G_{5000}$). This has considerable implications for the collaborative filtering process as it suggests that there could be a low expected overlap between profiles, because most users have only accessed a small amount of the available jobs, and the probability of other users also accessing these jobs will be low. This is the essence of the *sparsity problem* [5, 11] in collaborative filtering where there is a general lack of users’ ratings for items (in CASPER this corresponds to users accessing jobs).

**3.2.2 Profile Overlap Experiment**
From our preliminary investigations into the size of profiles, we hypothesise that there will not be a strong degree of overlap on average between profiles because the profiles are generally small in comparison with the total number.
Towards a Domain Analysis Methodology for Collaborative Filtering

If this is the case, it will reduce the prospect of the collaborative filtering technique working well because collaborative filtering relies on the similarity between any two users, which can only be computed over items that are common to both profiles. We now turn to study some actual overlap results.

When we make recommendations or predictions for a given target user in collaborative filtering, it is common for the algorithm to select a set of best users from which to make its recommendations. These best users or nearest-neighbours (NNs) are the set of users that have been selected by the collaborative filtering algorithm as the most similar users to the target user. It is from the profiles of these users that the recommendations for the target user are compiled.

The following experiment tests the NN-overlap in CASPER and again considers how this will be affected as increasingly smaller profiles are allowed into the calculations, (that is, the group size increases). We define NN-overlap as the average overlap for a given user and her $k$ nearest-neighbours (users with highest overlap), where $k = \{1, 5, 10, 20\}$. In each profile group we take 100 target users at random, compute their average NN-overlap and use this as a representative of the group as a whole.

**Results:** The results are shown in Figures 4, 5, 6 and 7. Figures 4, 5 and 6 tabulate the minimum, maximum and average NN-overlap results for $k$ for each group, and Figure 7 shows a graph of average overlap vs. group size.

<table>
<thead>
<tr>
<th>Group Size</th>
<th>$k = 1$</th>
<th>$k = 5$</th>
<th>$k = 10$</th>
<th>$k = 20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_{100}$</td>
<td>7</td>
<td>5.4</td>
<td>4.6</td>
<td>3.55</td>
</tr>
<tr>
<td>$G_{250}$</td>
<td>6.67</td>
<td>5.73</td>
<td>5.07</td>
<td>3.87</td>
</tr>
<tr>
<td>$G_{500}$</td>
<td>5</td>
<td>3.6</td>
<td>3.23</td>
<td>2.65</td>
</tr>
<tr>
<td>$G_{1000}$</td>
<td>4.33</td>
<td>3.4</td>
<td>2.93</td>
<td>2.37</td>
</tr>
<tr>
<td>$G_{2000}$</td>
<td>2.33</td>
<td>1.87</td>
<td>1.67</td>
<td>1.3</td>
</tr>
<tr>
<td>$G_{5000}$</td>
<td>1</td>
<td>1</td>
<td>0.8</td>
<td>0.4</td>
</tr>
</tbody>
</table>

**Figure 4:** Minimum NN Overlap

<table>
<thead>
<tr>
<th>Group Size</th>
<th>$k = 1$</th>
<th>$k = 5$</th>
<th>$k = 10$</th>
<th>$k = 20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_{100}$</td>
<td>85</td>
<td>56</td>
<td>44.4</td>
<td>30.6</td>
</tr>
<tr>
<td>$G_{250}$</td>
<td>68.67</td>
<td>43.4</td>
<td>36.2</td>
<td>29.13</td>
</tr>
<tr>
<td>$G_{500}$</td>
<td>63.67</td>
<td>45.13</td>
<td>37</td>
<td>29.2</td>
</tr>
<tr>
<td>$G_{1000}$</td>
<td>70.33</td>
<td>40.87</td>
<td>32.87</td>
<td>26.93</td>
</tr>
<tr>
<td>$G_{2000}$</td>
<td>44.67</td>
<td>36.73</td>
<td>32.8</td>
<td>28.08</td>
</tr>
<tr>
<td>$G_{5000}$</td>
<td>37</td>
<td>20.5</td>
<td>15.35</td>
<td>12.326</td>
</tr>
</tbody>
</table>

**Figure 5:** Maximum NN Overlap

**Discussion:** The results confirm our hypothesis that the average overlap between profiles in CASPER is low. In the group of the 100 largest profiles ($G_{100}$), the average size of the profiles is 118 (Figure 2), and the average best overlap (where $k = 1$) for any user in the group is 27.81, (Figure 6), that is, a 20% average profile overlap. We see that the average overlap between profiles degrades significantly as the number of profiles in the group increases, since new profiles are decreasing in size from group to group. These results indicate that it may only be profitable to base the collaborative filtering on a small subset of the users - those with larger sizes.
Towards a Domain Analysis Methodology for Collaborative Filtering

<table>
<thead>
<tr>
<th>Group Size</th>
<th>(k = 1)</th>
<th>(k = 5)</th>
<th>(k = 10)</th>
<th>(k = 20)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(G_{100})</td>
<td>27.81</td>
<td>19.64</td>
<td>16.02</td>
<td>12.37</td>
</tr>
<tr>
<td>(G_{250})</td>
<td>23.04</td>
<td>16.95</td>
<td>13.99</td>
<td>11.01</td>
</tr>
<tr>
<td>(G_{500})</td>
<td>18.64</td>
<td>14.3</td>
<td>11.97</td>
<td>9.49</td>
</tr>
<tr>
<td>(G_{1000})</td>
<td>14.81</td>
<td>11.33</td>
<td>9.58</td>
<td>7.72</td>
</tr>
<tr>
<td>(G_{2000})</td>
<td>11.41</td>
<td>8.97</td>
<td>7.63</td>
<td>6.26</td>
</tr>
<tr>
<td>(G_{5000})</td>
<td>6.06</td>
<td>4.62</td>
<td>3.92</td>
<td>3.2</td>
</tr>
</tbody>
</table>

Figure 6: Average NN Overlap

![Average NN-Overlap for Groups](image)

Figure 7: Average NN Overlap vs. Group Size

3.3 Job Analysis

Having looked at the properties of the profiles in the CASPER domain and examined the quality of the overlap that exists between the profiles we would also like to analyse the jobs in the domain. Specifically, we would like to examine the coverage of the jobs within the domain, and the nature of the distribution of jobs across the profiles.

3.3.1 Job Coverage Experiment

Our initial job analysis concerns the coverage of jobs within the user population. We are interested in examining how many profiles are necessary to cover all the jobs in the domain. We would like to investigate whether a subset of the largest profiles will cover the total set of jobs, or whether these jobs are spread out over all the profiles. The following experiment examines the job coverage within the different groups \(G_i\). We define the job coverage of a group to be the number of unique jobs that occur across the profile members of the group. For each group we measure the total group job coverage.

**Results:** The results are shown in Figure 8 which graphs percentage of job coverage vs. group size.

**Discussion:** From the above results we find that as group size increases, the number of different jobs covered by the profiles in the group also increases. However, the rate of increase is not constant, and after a while most of the available jobs are already covered by the group, and increasing its size adds little to the overall group coverage. This is a very positive result. It shows for example that even relatively small groups of the user population provide adequate
coverage of the job space. For example, the largest 2000 profiles ($G_{2000}$), provides a job coverage of over 97% of the job space. Thus, there is little coverage lost in focussing on a profile subset of 2000 profiles.

### 3.3.2 Job Distribution Experiment

We would also like to examine the distribution of jobs within groups. Specifically we are interested in examining whether the high job coverage results in the previous experiment (Section 3.3.1) are due to a few very large profiles covering most of the jobs, or whether the coverage is distributed more evenly across the group. In this experiment we analyse the job distribution across the different groups of profiles ($G_{i}$). We define the distribution of a job within a group to be the number of users in the group whose profile contains that job.

**Results:** The results are shown in Figures 9 and 10. Figure 9 depicts a graph of average distribution vs. group size, and Figure 10 shows a graph of percentage average distribution (i.e. the percentage of the population that a job is distributed across) vs. group size.
Towards a Domain Analysis Methodology for Collaborative Filtering

Figure 10: Average Job Distribution Percentage vs. Group Size

Discussion: Figure 9 shows how the number of profiles that a job is distributed among increases as the size of the group increases. This implies that as groups get larger, the expected number of profiles that any target profile will overlap with (by at least one job) increases. For collaborative filtering this means that a larger user group size should provide a better base for finding similar users. However, Figure 10 demonstrates that although the number of profiles that a job is distributed among increases, the actual percentage of the profiles in a group that a job is distributed among decreases. This indicates that individual jobs are better at discriminating between users in larger groups, because they are contained in a small relative subset of the user group.

4 Recommender System Design

In this section we reason about our findings from the domain analysis. Furthermore, we describe how the domain analysis has been an important driving factor in the design of the recommender system in CASPER.

4.1 Lessons Learned

In our domain analysis we have identified and examined the features of a domain that are crucial to the success of collaborative filtering. In our profile analysis we have looked at the important overlap properties between users in the different user groups. This allows us to begin to understand some of the important factors that will impact on the success of any collaborative filtering technique in the JobFinder domain. The main contribution of these experiments is the identification of the sparsity in CASPER’s domain. Furthermore these experiments allow us to draw some general conclusions regarding the setting of parameters in collaborative filtering for CASPER. For example, we notice that in the 5000 user group ($G_{5000}$), the 10 nearest-neighbour profiles to a target can include profiles that do not share any jobs with the target, (Figure 4). Therefore, we suggest concentrating on a reduced subset of the profile space in order to guarantee that there will be a strong relationship between profiles. For example, we suggest focussing on the 2000 user group ($G_{2000}$) at the $k = 10$ setting, since this guarantees an average NN-overlap of approximately 8 jobs (Figure 6), and a minimum NN-overlap of at least 1 job (Figure 4).

Of course, if we base our collaborative filtering strategy on a subset of profiles, will this compromise the recommendation coverage? That is, will this compromise the number and range of different jobs that are contained in the set of profiles and therefore that can ultimately be recommended? If coverage is compromised, some jobs will be excluded from the system and will not be recommended, thus the number of jobs available for recommendation in general will be reduced.

Our analysis of the jobs in the CASPER domain allows to draw some more important conclusions about the domain
as a platform for applying collaborative filtering. It also allows us to examine the full implications of concentrating on the smaller subset of users that the profile analysis suggested we should. Ultimately, we need to have enough users so that lots of jobs are covered by their profiles. This allows us to have good recommendation coverage. In Figure 8, we find that job coverage increases asymptotically with increasing numbers of users. If we select the largest 2000 users ($G_{2000}$), 97% of the jobs are covered, and even if we only select the largest 1000 users ($G_{1000}$), we still attain a coverage of 92%. Therefore, we conclude that the job coverage in the domain is good.

In addition to good job coverage, we also need a good distribution of the jobs across the profiles, to allow us to find similar users within the population, that is, users who share the same jobs. In Figure 9 we notice that as the group size increased, so too does the job distribution; that is, on average a job appears in more profiles as group size increases.

However, although we would like to have a good distribution of the jobs across profiles so we can identify similar users, we still need the jobs to be predictive. In other words, we would like them to discriminate well between users. For example, there is no benefit in having a job that is distributed among 20 users, if there are only 25 users in the population, since this only indicates that this job is a very popular job. Conversely, if a job is only found in 3 profiles, out of a population of 100 profiles, it indicates that there might be a relationship between these 3 users because they are the only users that liked this relatively unpopular job. Essentially this means that, while we want the distribution of the jobs to increase, we want the percentage of the population that the jobs are distributed across to decrease so they maintain some predictive value. This is what we find in Figure 9 and Figure 10. Indeed, if we combine Figure 9 and Figure 10 to plot the percentage average distribution of a job vs. the actual number of users the job is distributed across, for different values of group size we can see this more clearly, (Figure 11).

![Figure 11: Average Job Distribution Percentage in Groups](image)

As a job finds its way into more profiles, for increasing group sizes, these profile sets actually represent smaller and smaller fractions of the full population. We notice that in the 2000 user group ($G_{2000}$), a job is distributed among approximately 0.4% of the population on average, and in $G_{5000}$ a job is distributed among approximately 0.2% of the population on average. Therefore, considering the entire population of users (5132) as opposed to a smaller subset of 2000 users adds very little to the predictive quality of the jobs. These results therefore, support those in the profile experiments, (Section 3.2) that indicate that it may be more useful to concentrate on a subset of the population for recommendation purposes. Once again, the ideal population size seems to be about 2000.

The most important insight from our domain analysis is the identification of the sparsity problem that exists in the JobFinder domain. This problem is a noted challenge for collaborative filtering. Our analysis indicates that the one of the reasons that such sparsity exists in the domain is that the size of profiles is generally small, which means there is a low expected overlap between users. One of the main contributions of the domain analysis is the realisation that we can concentrate on a subset of the larger profiles in the domain (approximately 2000 in our case). In doing so, we find that the overall expected overlap is improved, and importantly this improvement does not result in a major compromise of other important domain features like job coverage and distribution. In the next section we consider the
reasons behind the sparsity in the JobFinder domain more fully.

4.2 JobFinder’s Dynamic Domain

At the heart of the sparsity problem in JobFinder we find the dynamic nature of the domain. The user population is continuously changing due to the nature of job seeking - users use the JobFinder system for a limited period of time, until they find a job. The small profile size that contributes to the sparsity is a direct result of this feature. Unfortunately, this is a feature of the domain that we cannot change. However, it is still interesting to look at such a domain, and analyse whether collaborative filtering can be applied when this is the case.

If there is a low expected overlap between the profiles, it suggests that users in general rarely share the same tastes in jobs. Intuitively, though, many users will look for similar types of jobs and therefore share the same taste for jobs. This leads us to consider another important feature of the domain in JobFinder. Unlike many related systems, the set of jobs in the JobFinder domain is also dynamic - in the sense that jobs are perishable - each with a limited life-span as jobs are constantly filled and advertised. This means that jobs do not have the potential of being seen by all the users of the system, and probably only by a small subset of users who happen to be using the system at the time the job is available. In related systems (for example, amazon.com, PTV, Ringo), the information items at hand are long-lived static items such as music, movies, television programmes, books etc., the difference being that books or movies don’t (usually) go off the market and are therefore available to any user at any time. Therefore, in such systems each user has the potential to see all the given information items. For collaborative filtering, this means that a user profile represents the user’s preferences for particular items over all the available items (in the sense that a user has the opportunity to purchase any book for example in amazon.com, rather than the fact that the user has read all the books).

However in CASPER the set of jobs available to any user at a given point in time may be completely different to the set of jobs available to another user at a different point in time. Therefore a user profile in CASPER only includes information about the user’s job preferences over the set of jobs that were available at the time she was searching for a job.

Essentially, this means that there can exist users in CASPER who share the same taste for particular types of jobs (i.e. their profiles show that they look at the same types of jobs) but have no overlapping items between their profiles because they have used the JobFinder system at different times. These indirect relationships between users also increase the sparsity in the profile space as similar users cannot be identified by measuring the overlap between their profiles. The traditional memory-based method of collaborative filtering relies on this measurement and judges similarities between users based on a direct overlap of items between their profiles, [8]. If there is a high degree of sparsity and therefore little overlap between profiles the traditional memory-based algorithm is rendered virtually useless.

Our solution to this has been to adopt a clustering approach, [8]. Other research (for example [5, 7]) has also used clustering to overcome sparsity. For our purposes the single-link clustering algorithm is especially useful as it identifies chains of similarity and therefore allows for the identification of indirect relationships between users. Consider the example in Figure 12. The diagram shows three profiles for users A (target user), B and C (recommending users), with each job title shown beside the job ID (the job title information is not contained in the profile as recommendation is content-free, but is displayed for this example). Jobs that overlap between profiles are highlighted, and the profile that each job overlaps with is indicated (circle on right of job). Underneath each profile, the dates of the first and last access by the user are displayed, and define the time frame during which the user was job-searching in the JobFinder site. The diagram shows that user A has used JobFinder during July ’98, and user C has used JobFinder during September ’98, and that both users are similar, that is, they both search for similar types of jobs. In this example, both users tend to look at technical support type jobs.

Due to the dynamic nature of the job market, the jobs that are available to A in July, and the jobs that are available to C in September will be different, and therefore their profiles will contain different jobs, that is, there will be little or no overlap between their profiles. However, consider that user B, who also looks at technical support jobs has used JobFinder from June through September. Therefore B has seen the jobs that were available to both A and C, and shares some jobs in her profile with A, for example berlitzjobs*128, and some with C, for example esmjobs*190. Then, it is possible to identify the transitive indirect relationship between users A and C through user B. If relationships are measured based purely on direct relationships the indirect similarity between A and C will be lost. However if this
Towards a Domain Analysis Methodology for Collaborative Filtering

Figure 12: Indirect Relationship

type of indirect relationship can be exploited, then the potential for collaborative filtering in sparse profile spaces like CASPER’s is greatly enhanced.

Indeed previous experimentation ([8]) supported these findings. We have performed a preliminary experiment to test the advantages of using a (single-link) cluster-based algorithm over the more traditional memory-based method of collaborative filtering for JobFinder. The results support the hypothesis that the cluster-based method is a more suitable strategy for CASPER than the memory-based version. We have argued that the sparseness of CASPER’s profile space presents significant problems for the more traditional memory-based approach to collaborative filtering. This approach relies on the identification of direct relationships between users which is necessarily a function of the overlap between profiles. However, as explained in Section 3, the sparseness of the JobFinder domain is such that the expected overlap between profiles is likely to be very low, making it difficult to identify nearest-neighbours based on direct overlap calculations.

In contrast, the cluster-based method is basing its recommendation on a potentially more reliable measure of indirect relationships between profiles. These indirect relationships can potentially occur frequently in a domain like CASPER’s where the set of information items and/or profiles is dynamic, and users can share the same taste for jobs, despite the fact that they don’t share jobs between their profiles because they have used the system at different times. It could be argued of course, that these indirect relationships may not constitute fruitful recommendation bases as many jobs cannot be recommended across large gaps in time as they are either not available then, or have already been filled. However, we believe that even if this is the case, they can still play an important part in linking other users together, in other words, they are an integral part in the formation of relationship chains within the user population.

23rd BCS European Annual Colloquium on Information Retrieval Research, 2001
5 Future Work

One of the key motivations of the work we describe in this paper is the development of a collaborative filtering methodology that takes crucial domain features into account. In this endeavour so far, we have identified particular domain features such as the overlap between profiles, and the distribution of items across the domain etc. that are important to the success of collaborative filtering (Section 3). Furthermore, we have described how knowledge of these features has helped the construction of a particular collaborative filtering system for the JobFinder online recruitment service (Section 4). We believe this research is one step on the way to establishing a complete domain analysis methodology for collaborative filtering.

Obviously, we would like to determine how useful such a methodology is for collaborative filtering. Based on preliminary experimental evidence we believe that such an analysis is important, however, it would be useful to determine the benefit of using such a methodology over more standard approaches by doing a comparative study of the performances. Moreover, we would like to provide a cross-domain comparison of the methodology to see if our findings in the JobFinder domain also hold across other domains. These studies should help towards evaluating the usefulness of such a domain analysis methodology.

Essentially, our final aim is to develop a methodology that allows us to make predictive statements about the practicality of applying collaborative filtering in a given domain. In other words, if we have thresholds for overlap, coverage etc. can we decide whether it is useful or possible to apply collaborative filtering? Furthermore, we would like to analyse whether such a methodology is useful in predicting the style of filtering that best suits a given domain.

The research presented here outlines how the sparsity in the JobFinder domain was identified using domain analysis, and how knowledge of this helped select a style of collaborative filtering (cluster-based) that outperformed a more traditional (memory-based) approach in preliminary experiments. In the future, we would like to investigate if such predictions are possible from domain analysis in other domains too.

6 Conclusions

In this paper we have presented research towards a domain analysis methodology for collaborative filtering. We have performed an extensive domain analysis of the JobFinder domain, identifying and examining particular features that influence the success of collaborative filtering technologies. This analysis identified the sparsity problem that exists in the JobFinder domain and indicated that concentrating on a small subset (approx. 2000 in our case) of the largest users would improve the sparsity. Furthermore, the analysis showed that concentrating on this subset of users would not compromise other important features such as job coverage, which are also important in successful collaborative filtering.

The domain analysis prompted further consideration as to why this sparsity exists within the JobFinder domain and why the expected overlap between profiles is low. This led us to identify the dynamic nature of the profiles and more importantly the perishable nature of jobs in the domain as key influential factors of the sparsity. We noticed that many (indirect) relationships existed between users despite the fact that this was not evident from direct profile overlap measurements. Indeed previous experimentation indicated that using a clustering approach that allowed for the identification of indirect relationships improved sparsity and gleaned better results than a more traditional approach.

We believe the contribution of this research is not so much the end solution of a cluster-based collaborative filtering approach for JobFinder’s sparse domain, but rather the methodology that was involved in arriving at this solution. We have identified domain features that are crucial to the success of collaborative filtering, and we have performed an extensive analysis of these features which previous related research has failed to provide. We believe that this research provides an important basis for developing a complete domain analysis methodology for collaborative filtering. We advocate the use of domain analysis for collaborative filtering to facilitate a thorough investigation of features critical to its success. Furthermore, we have demonstrated how knowledge of these domain features has been used successfully to select a suitable style of collaborative filtering for the domain at hand.
References


