UNIVERSITY OF SOUTHAMPTON
Faculty of Engineering, Science and Mathematics
School of Electronics and Computer Science

A mini-thesis submitted for transfer from MPhil to PhD

Supervisors: Dr. Srinandan Dasmahapatra and Prof. Paul H. Lewis
Examiner: Dr. Terry R. Payne

Unlocking the Potential of Recommender Systems: A Framework to Achieve Multiple Domain Recommendations.

by Antonis Loizou

September 9, 2007

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

A mini-thesis submitted for transfer from MPhil to PhD

by Antonis Loizou

This upgrade report aims to motivate the consideration of an alternative paradigm for Recommender Systems. Traditional systems have been designed under the assumption that an exhaustive index of possible recommendations is available, and that users can be adequately characterised solely through their interactions with resources in this index. Instead, we argue that by compiling a semantic log of all user activities, a much more complete profile can be obtained. Items for recommendation are introduced to the system through importing preference data from external communities and social networks, enabling multiple domain recommendations. A translation phase is required in order to compare user profiles with members of such communities, in order to assess resources for recommendation. This is achieved through exploiting a *'Universal vocabulary'*, in which unique descriptors may be obtained for the concepts discovered in the user's semantic log, as well as for the resources considered for recommendation. Since recommendations may stem from a number of domains, a domain selection process driven by the user's current contextual setting is carried out in order to identify the most appropriate one(s). Moreover, the generality of the approach described herein, unlocks new possibilities for the deployment of RS technologies to problems such as guiding autonomous coordination in Multi–Agent (or peer–to–peer) Systems, discovering new applications of existing information, or new scenaria for the use of computational facilities.

Contents

List of Figures

Acknowledgements

I would like to thank my supervisors, Srinandan Dasmahapatra and Paul Lewis, for their invaluable support, guidance, and patience. I'd also like to express my gratitude to Mischa Tuffield, for his efforts in implementing various elements of the framework, and for always lending a helping hand in general.

This work is supported under the OpenKnowledge EU Framework 6 STREP project, Grant number IST-FP6-027253.

Chapter 1

Introduction

The abundance of available digital information, electronic resources, and on–line services has led to the issue often referred to as the *Information Overload* problem. Given an enormous corpus, how can one identify credible and interesting resources? Under the assumption that people are always aware of what it is they are interested in, the aforementioned problem becomes a (non–trivial) search and retrieval task. This, however, is not always the case. It is human nature to constantly discover new interests, new challenges, and new forms of entertainment as facilitated by the social environment. This observation outlines a need for systems that are able to support this process by predicting how interesting unseen and unsought resources will appear to a given person.

There has recently been a rapid increase in the commercial use of Recommender Systems (RS) technologies, primarily by online retailers. Such systems appear attractive to this audience, since they can be used to identify any, and all products from the retailers catalogue that can be expected to appear interesting to a particular customer. This emphasis was confirmed by Jim Bennet, of NetFlix[1], at his invited talk in Recommenders06: *"... Most of the time we don't actually have to sell them our movie, because they already know it, and just have to recognise it. So as long as we can pull those out, we are actually doing a pretty good job. ... Even though all the disks look the same, they don't cost the same for us to acquire."*, [Bennet (2006)] This view conflicts with the classical notion of recommending in two ways. First, people seek recommendations for things they have limited knowledge of, while using clever search and indexing techniques to remember things they were previously interested in. Secondly, the merit of a recommendation should be measured solely on how well it fulfills the creating need, rather than by taking into account the relative profit of the recommending agent.

It is not the purpose of this report to argue the applicability of the term 'Recommender System' to the few existing commercial implementations. It will, however, be argued

[1]NetFlix (*http://www.netflix.com*) is an online DVD rental company that utilises a RS, typically dealing with billions of ratings and predictions. It is considered one of the most successful commercial RS deployments.

that assuming an explicit, exhaustive catalogue and a single user community at the outset limits the predictive capabilities of the systems that may be developed. Furthermore, this report will outline a framework, under which new recommendation domains and user bases can be added dynamically to a recommendation scheme, and correlated with existing user profiles to improve both on the quality and the suitability of recommendations. Moreover, the generality of the approach described herein, unlocks new possibilities for the deployment of RS technologies to problems such as guiding autonomous coordination in Multi–Agent (or peer–to–peer) Systems, discovering new applications of existing information, or new scenaria for the use of computational facilities.

The next chapter provides an overview of the background literature that was drawn upon to inform the development of this approach. Following that, chapter 3 presents the motivation for the main design decisions taken, listing in detail the problems they address. Chapter 4, Methodology, outlines the framework and describes its essential components. Chapter 5 describes a prototype implementation of an entire system, while in chapter 6, the various algorithms considered to implement the recommendation process are presented along with experiments designed to assess their suitability. The final two chapters outline directions for future work and present the conclusions that may be drawn so far.

Chapter 2

Background Literature

This chapter sets out to pinpoint the origins of the ideas presented in this report in the relevant literature. For clarity of presentation it is divided into four sections, each reporting on literature from different areas. Some focus on reporting the *state of the art* in the field, while others serve as a justification for the choices made and the techniques used in developing this approach, with respect to the scope of this project.

2.1 Recommender Systems

This section provides an overview of the main niches of Recommender Systems architectures and the intuition behind their inner workings. However, it is not intended to be a systematic analysis of the algorithms used, rigorously pinpointing the points of failure; for this, the interested reader is pointed to [Adomavicius and Tuzhilin (2005)]. This is the case since the scope of this project is not to invent the perfect recommendation algorithm, but instead to identify a process where any (internet accessible) resource may be recommended, provided that user preference information is available.

2.1.1 Rule filters

Rule filters were used in the first RS, Tapestry [Goldberg et al. (1992)], and either require a user to explicitly formulate rules to filter out bad recommendations or try to infer these rules based on the user's history. A number of drawbacks are apparent in this architecture. Users find defining rules in a formal language awkward and cannot be expected to formulate good quality rules. On the other hand,the automatic approach can be very complicated and can produce rules that do not reflect user preferences but happen produce good results in the training phase by chance. This is a problem because as the user accesses more items such rules can become conflicting and need

to be reassessed. Moreover, to be able to define rules to constrain which items can be recommended, users are required to be aware of exactly what they'd like to be recommended to them which detracts from the true notion of recommending.

2.1.2 Content based systems

Content based systems record descriptive features of items and try to identify the most similar ones in the catalogue, under the assumption that users will like similar items to the ones they liked before [Balabanovic and Shoham (1997); Pazzani and Billsus (1997)]. This niche of architectures is greatly influenced by progress in data mining, classification and machine learning in general, since the recommendation problem is reduced to the question "Is this item sufficiently similar to those in the training set?". A vector is constructed for each item containing values for the descriptive features of each item and is considered a point in a multidimensional space. Inter–item similarity is assessed by evaluating the distance between such points. This reduction however is not always correct, since the most similar items to those already seen by a user can often be of little interest, if the original recommendation need has already been fulfilled.

2.1.3 Collaborative filtering

Systems that apply Collaborative Filtering (CF) assume that users will be interested in items that users similar to them have rated highly. The large number of deployed RS implementations that use this strategy [Resnick et al. (1994); Linden et al. (2003); Billsus and Pazzani (1998); Pennock et al. (2000); Yang et al. (2004); Linden et al. (2003); Bennet (2006); Last.fm Ltd. (2006)] indicates that the assumption made generalises well. The active user is first matched to the group of most similar users using a similarity threshold and items seen by the group but not by the active user are identified. The predicted rating for each unseen item is then computed by aggregating the group's ratings. This is typically weighted by the number of group members that have accessed a particular item and the variance between ratings, effectively biasing the process towards items that more people in the group have unanimously rated highly. Finally, the items with the highest predicted rating are recommended.

The point of failure in this architecture lies in choosing the correct cluster for each individual user. Due to the level of sparsity in the datasets RS typically deal with, users can appear equally similar to any other user if the similarity metrics used are not sensitive enough. As such, the user is merely provided with recommendations for the items most popular with a group of 'randomly' selected users with a potentially high degree of disagreement among members of this group. The reverse effect is also present: items can only be recommended after being rated by a sufficient amount of users. These problems are commonly referred to as the "cold-start" issues [Schein et al.

(2002)]. Dimensionality reduction techniques such as singular valued decomposition are often applied to reduce the effect of data sparsity, but are computationally expensive and do not resolve the issue in all cases.

2.1.4 Knowledge based systems

Knowledge based systems make use of an underlying ontology to describe both the users of the system and items to be recommended. Recommendations are provided via assessing the similarity between instances associated with the user and all other instances in the system's knowledge base, by applying graph-based edge expansion heuristics [Alani et al. (2003)]. Such an approach also enables the user to visualise and amend their profile in order to reflect their preferences more accurately, since it is represented as graph. One such system, Foxtrot, [Middleton et al. (2001)] has empirically been shown to outperform other strategies, however, as with any knowledge based system, knowledge acquisition poses a problem which is typically dealt with by employing (usually labour intensive) bootstrapping techniques.

2.1.5 Context dependent systems

The architectures mentioned above (with the exception of knowledge based methods) all use a flat matrix representation of the problem domain where rows correspond to users and columns to items. As mentioned before, these tend to be vast and very sparse, thus adding to the problems of defining efficient similarity metrics and computational complexity. Adomavicius et al. (2005) propose that contextual dimensions (such as time) should be added to this representation so that at the time of recommendation a relatively dense *'slice'* of this space is used, selected based on the current context. They show that such an approach is beneficial in most cases and have defined a metric to evaluate a priori whether the multidimensional approach is likely to outperform the traditional flat representation. Where it is not their system falls back to a conventional recommendation scheme.

2.1.6 Hybrid systems

Hybrid systems combine two or more recommendation techniques in order to improve the recommendation quality, [Balabanovic and Shoham (1997); Pazzani (1999); Adomavicius et al. (2005); Berenzweig et al. (2003)]. A comprehensive analysis of such systems can be found in [Burke (2002)]. In summary, there are three main types of hybrids:

- Weighted hybrid RS assign weights to each strategy used, and aggregate the results from each one to compute the predicted rating for a recommendation.

- Mixed hybrids will provide any recommendations above a confidence threshold from each scheme used.
- Hybrids where a single strategy is chosen each time, based on a heuristic to identify the one likely to perform best.

It should be noted that the first two design paradigms for Hybrid Systems introduce large increases in the computational requirements of the developed systems, since a number of distinct recommendation techniques is carried out. Conversely, accurately predicting the performance of the recommendation algorithm is no simple task. Various heuristics (such as the sparsity index of the entire space, the number of available ratings for the active user, etc.) are commonly used to predict the algorithm's performance, incurring minimal, but observable losses in terms of predictive accuracy.

2.2 Semantic Web

The Semantic Web vision promotes the adoption of a common standard for the representation of knowledge[1], allowing for its seamless integration and enabling automated reasoning over it. The knowledge is structured using ontologies, representing concepts, instances and relationships between them as nodes and edges in a graph. Inheritance and transitive relationships are supported adding to the inferential power of the representation[Gruber (1993)].

The Friend Of A Friend[2] project(FOAF), allows people to publish any demographic (or other) information about themselves and indicate who their 'friends' are. The adoption of such an infrastructure in a RS setting rids users of the burden of describing themselves, and also enables the easy detection of social networks. These can be exploited for recommendation purposes. Furthermore, the availability of semantic resource descriptors at variable levels of granularity permits the better assessment of the relevance of their features in the recommendation process.

2.3 Information Retrieval and Natural Language Processing

.

The majority of the information held at a private machine is expected to be in textual form, such as e–mails, calendar entries, articles, web–pages, etc. However, this information is of limited use if it is kept in its' original form, as contiguous sentences. The

[1] Resource Description Framework, http://www.w3.org/RDF/
[2] FOAF, http://www.foaf-project.org/

field of Natural Language Processing deals with the issue of extracting semantics from such fragments, by performing tasks such as *named entity recognition* and *part–of–speech tagging*. GATE [Cunningham (2002)], the General Architecture for Text Engineering, is comprised of an architecture, a free open source framework and a graphical development environment to support such task. In the context of this work, GATE is to be used to identify concepts and named entities in text, that can then be used as profiling features. Traditional Information retrieval techniques such as the *Term frequency – Inverse document frequency* metric can subsequently be used to filter out common concepts.

We also intend to extract semantic information from multimedia data, captured by metadata associated with each format. The Exchangeable Image file Format for digital still cameras, [EXIF (2002)], automatically captures camera parameters at the time that the photograph was taken. These parameters include: aperture setting; focal length of the lens; exposure time; time of photo; flash information; camera orientation (portrait/landscape); and focal distance. Other information may also be derived from this metadata, such as the average scene brightness of an image. In the case of audio, the ID3 tagging standard[3] together with the Musicbrainz [Swartz (2002)] knowledge base can be used to obtain ample information. These have been selected based on their widespread use, since descriptions in more detailed representations, such as the MPEG – 7 [Salembier and Smith (2001)] standard, are rare.

2.4 Summary

The issues that hinder the performance of current RS architectures are well understood in the literature. As was mentioned in the previous chapter, most Recommender Systems are preoccupied with 'pushing' the most appropriate products from a business catalogue to customers. This has led to a narrow view of the field in which the users are represented solely through their interactions with the system, and as such may only be correlated with other users of the same system.

While various bootstrapping techniques are commonly applied to lessen the effects of *cold–start* problems, they are costly and hardly a solution. The practices and techniques used in the Semantic Web community can facilitate the seamless integration of prior information about users, items and the relationships between them. The integrated knowledge can then be used to overcome such problems. In parallel, the emergence of a plethora of social networking sites that record and expose the opinions of their users on resources in some domain, opens up new possibilities for identifying resources to recommend and eliminates the need for an exhaustive product catalogue.

Of course, the deployment of any knowledge–based system requires the completion of a knowledge acquisition phase. This can be tackled using tools developed in the fields of

[3] http://www.id3.org

Information Retrieval and Natural Language Processing to extract semantics from the raw data already held at the user's node.

Hybrid systems have been developed to improve on the accuracy of predicted ratings by combining a number of distinct approaches. In their majority they amount to the sequential application of filters, thus increasing the computational requirements of the system. In addition, the datasets commonly used in commercial situations are both incredibly large and sparse, hindering the performance and efficient of learning algorithms.

Chapter 3

Motivation

A number of problems with conventional RS implementations have been identified in the relevant literature. These have served as the primary motivation for the approach described in this report, shaping the methodology used to develop it. An overview of these characteristic problems is provided in this chapter, indicating the directions taken to rectify them.

1. *The 'cold–start' problems*

   In CF–based recommending, the similarity between user profiles is assessed to predict ratings for unseen items. The shortcoming of such a method rests in its assumption that active users will respond positively to unseen items rated highly by similar users [Pennock et al. (2000)]. As most users are not inclined to rate previously seen items, only a few items will receive ratings. This limited data – the 'cold start' problem – renders similarity metrics not sensitive enough to distinguish between users, particularly new ones introduced to the system[Schein et al. (2002)]. Hence, the most highly rated items from anyone are recommended.

   This is viewed as a direct consequence of assuming an explicit and exhaustive catalogue to draw recommendations from, and also to describe users. Since the focus in RS applications has been to enable organisations to suggest appropriate items from their catalogue to customers, not much effort has been put into learning user preferences based on the items they already have in their possession, regardless of their origin. However, a good sales assistant in a clothing shop will first look at what the customer is wearing before making suggestions.

   To remedy this, the user profiling components of this approach have been designed to surreptitiously integrate as much of the information present in the user's local machine as possible, while at the same time maintaining privacy and control over the exposed information. In addition, the user is supported in extracting added

value from this profiling scheme (see section 5.1 for details), providing new incentives to export personal information to the system.

The reverse effect is also present, i.e. a newly imported resource can not be recommended until it has received a sufficient number of ratings. There are two possible reasons to introduce a new resource to a catalogue; either a new product has been developed, or an existing resource has now become available to the recommending agent. Little can be done to improve performance in the former case. If no–one other than the manufacturer has empirically evaluated the resource, recommendations cannot be made. If, however, the items to be recommended are to be obtained by minimising the preferences expressed by external (to the system) user communities, the latter case will never occur.

2. *The most similar items are not always good recommendations*

   Content Based (CB) approaches index the items of possible interest in terms of a set of automatically derived descriptive features, and unseen items with similar attributes to those rated highly by the user are recommended. A drawback of the CB method is that it recommends items interchangeable with those that have previously received high ratings, by virtue of its focus on items' features, ignoring potential user requirements.

   As shown in [Ziegler et al. (2005)], diversity is a desirable property of lists of recommendations. By considering more than one domains, however, the probability of recommending identical resources increases, since preferences for a resource may have been expressed in more than one 'expert community' considered by the system. This can be avoided by exploiting a 'universal vocabulary', a corpus of terms and relationships between them. Permutations of these can form description vectors and in turn be used to uniquely identify any resource in the world. A comparison of such unique identifiers can then indicate whether two resources can be considered equivalent.

3. *Shifts and temporal cycles of user interests*

   Most conventional RS architectures do not model for shifts of the user's interest over time, since all ratings provided by a user have an equal bearing on the recommendation selection. To accommodate this requirement of preference time dependence, conventional architectures recompute their user clusters periodically, effectively choosing a different training set every time. This can aggravate problems caused by data sparsity, and important modeling decisions about transitions between user needs have to be addressed.

   Furthermore, while this strategy may be somewhat effective in capturing permanent shifts of interest, it cannot be expected to cope in situations where preferences are highly dependent on temporal and contextual attributes. For example, a user

may find articles interesting during work hours, but would not even consider reading the same articles in his free time, as he would rather watch a movie and relax. In order to be able to accommodate such processes, a system would need a way to differentiate between the various types of resources a user may be interested in at given times. This can be achieved, provided that expert communities are available for these implicit 'interest domains', by carrying out a domain selection process guided by contextual and temporal attributes.

In summary, the approach is novel in that:

- The user is given incentives to expose (a part of) a log of their digitally visible life to the system, instead of selecting *a priori* the attributes that will characterise users. This rich information is then processed to drive the recommendation engine, while providing immediate added value.
- The recommendation process is viewed from the user's perspective. The goal here is to identify the most interesting resources for the user at a particular time, rather than trying to 'push' as many products from a catalogue to the customer as possible while ensuring their satisfaction.
- Because only resources from a pre–compiled catalogue could be recommended, this is typically used as the single means to index both users, and resources. Since this requirement is not applicable here, the user can be correlated directly with 'domain experts', members of communities that specialise in the domains from which recommendations are drawn.
- New domains can be added dynamically to the recommendation scheme with minimal effort.
- By not imposing restrictions on either the user characteristics or the origin of the resources for recommendation, Recommender System technology can be applied to a variety of new situations, such as coordinating the behaviour of agents (or peers) in complex systems.
- Finally, the use of a universal vocabulary to automatically and uniquely describe any resource, allows predictions to be made even when the user has not previously expressed interest for any resources in a particular domain.

Chapter 4

Methodology

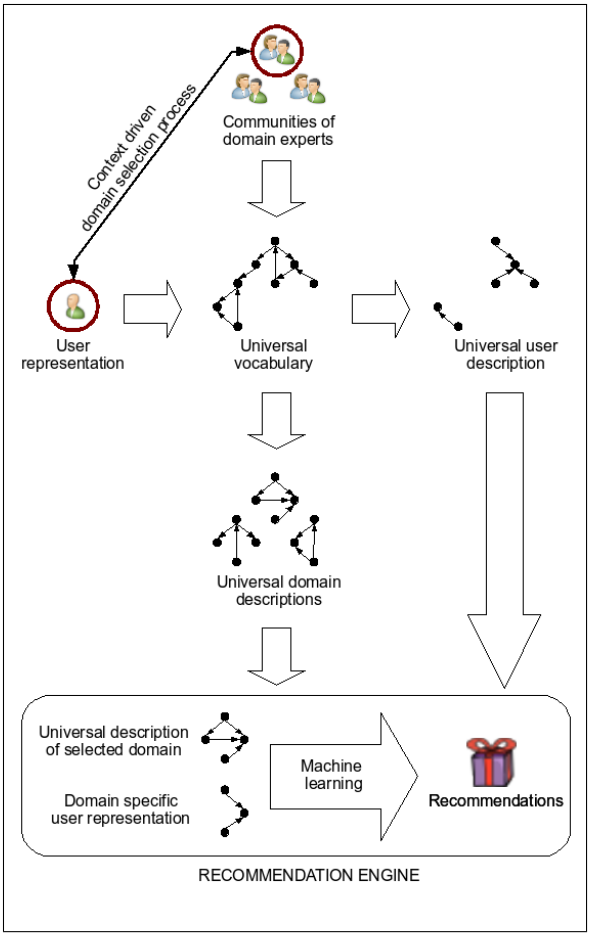

FIGURE 4.1: Overview of the framework.

In the previous chapter, a number of directions were identified in order to develop a recommendation framework able to overcome well known problems with conventional architectures. The specifics of the approach are presented here.

The recommendation process can be seen as consisting of four, non–sequential stages. First, a complete (as much as possible) representation of the user is required. In addition,

the contextual setting at the time of the recommendation is evaluated, and used to identify a domain from which to draw resources. Then, the preferences of an external community (or social network) pertaining to resources in the domain, are obtained and processed. The user will be compared to the members of this community (the *'experts'*) and the closest matches used to produce recommendations. There are two essential components required to carry out this comparison: a vocabulary expressive enough to describe both the user and the *'experts'* in arbitrary detail, and a machine learning component able to carry out the matching efficiently. The following sections elaborate on this process, each introducing the details of a stage.

4.1 User representation

In order to be able to draw recommendations from multiple domains, the user needs to be represented in as much detail as possible. This requirement stems from the assumption that for distinct domains, different subsets of these characteristic features will be appropriate for assessing correlations.

Furthermore, the acquisition process should be carried out with minimal user involvement, since the unwillingness of users to explicitly provide information is well documented. To achieve this, a variety of local components able to process and extract semantics from the various forms of data in the user's possession is required. Semantic Web technologies are apt for representing such knowledge, and can also easily facilitate its integration.

Ideally, all digital information created by a user would be encapsulated in the system. This highlights privacy and security issues that have to be carefully dealt with. As such the user should have complete control over the information that will be processed. This can be ensured by providing each user with a distinct knowledge base to which only themselves, and the automated algorithms have access to.

It is expected that even the minimal burden of registering and configuring the system, coupled with reasonable privacy concerns can still be off–putting to some users. Joining the system can be made to appear more attractive, by supporting the users in extracting immediate added value from the information they have exported. As an effect of integrating this information into a single knowledge base, much more complex queries may be resolved (e.g. how many hyper–links have I received in email correspondence and not yet visited?). In addition, by carrying out inference over this knowledge base, events that have occurred at the same place or time (or any other attribute) can be identified, allowing the information to be indexed based on such attributes.

4.2 A universal vocabulary

The external communities that the users are to be compared to cannot be expected to expose information in the same format as that used to represent users within the system. Rather than attempting to obtain a mapping for each combination of users and domains, the notion of a universal vocabulary is introduced. This would consist of a large number of terms and associations between them, such that any distinct resource in the world can be uniquely identified through a sequence of such terms. Under this assumption, the comparisons between users and 'experts' can be made using their respective descriptions in the universal vocabulary. Such an expressive vocabulary can not be expected to be available, and Section **??** deals with the subject of using Wikipedia as an adequate replacement for the universal vocabulary.

4.2.1 Context and semantics

Since recommendations may stem from a number of distinct domains, each with their own expert community, the more appropriate domains need to be selected each time a recommendation is to be made. Provided that there exists a taxonomy of the universal domain, the available historical data about users can be used to identify the contextual properties that have a bearing on how interesting certain classes in the taxonomy will appear. The candidate recommendation domains are also identified with such classes, and the domain selection is based on the intersection between the two.

The presence of a taxonomy (or even richer constructs) over the universal vocabulary also enables the system to present information about the semantics of a recommendation being made. Furthermore, once the classes of interest for a particular user have been identified, the user can be prompted to sort them by degree of interest, introducing weights to the elements comprising their description in the system.

4.3 External, 'expert' communities

This section serves as a justification for using the preferences of external (to the system) communities to produce recommendations. First and foremost, the need to bootstrap the system is eliminated. Most RS suffer greatly in terms of predictive accuracy when they are first deployed, since there are not enough users and ratings to produce meaningful clusters. Hence, a number of ratings is typically injected into the system manually, to 'kick start' the process.

However, there are other advantages associated with this approach. The vector space spanned by a single domain is relatively much smaller and more dense than that spanned by the terms in the universal vocabulary. Also, since members of such communities are

considered to be enthusiasts in the field, they are more prone to provide larger quantities of high quality ratings. Furthermore, they are expected to react faster in assessing the quality of new resources as they become available.

4.4 Recommendation engine

A number of challenges become apparent when attempting to apply machine learning algorithms to cluster user preferences regarding the resources in a domain. The very high number of dimensions in the vector space (one for each possible item) coupled with the fact that approximately 99% of the values are missing are the main causes of the problem.

As such, the use of complex, computationally intensive algorithms on the raw data is prohibited, since this would render the system unusable in terms of responsiveness. For similar reasons, so are traditional dimensionality reduction techniques such as Singular Valued Decomposition (more in chapter 6). Furthermore, such techniques are typically applied to reduce the number of attributes that need to be recorded for each datapoint, while retaining the intrinsic structure of the vector space. In the recommendation case, such techniques would eliminate resources that do not seem to be indicative of significant differences between experts. It is trivial to observe that the effectiveness of such methods reduces as sparsity in the dataset increases.

In addition, the datasets RS are expected to deal with typically contain much more points than variables – more users than resources. Thus, a reduction in the number of points considered through the use of fast summarisation techniques is more beneficial in terms of speed than reducing the number of features available for each user.

Chapter 5

Implementation

As *proof of concept*, a prototype implementation of the framework has been developed. While it would be nice to produce a fully functional, scalable system, the goal for the prototype is mainly to provide evaluation opportunities for the ideas presented in this report.

It should be noted that although much emphasis has been put into reducing the computational requirements of the system, the sheer size of the datasets involved requires an extremely large amount of both memory and computing power. As such, it has been intended from its conception that the architecture should be implemented in a distributed fashion. The user profiling components would run locally on the user's node, and specialised nodes for performing recommendations, preprocessing domain preferences, and obtaining universal descriptors would be in place to support the process.

5.1 The semantic logger

The Semantic Logger[1] [Tuffield et al. (2006)] is an auto-biographical metadata acquisition system that can be seen as a means to populate the Semantic Web with personal metadata. However it has been developed mainly as a platform for the deployment of additional context aware applications, such as the work presented in this report.

The Semantic Squirrel Special Interest Group (SSSIG) [2] is a group of researchers based at the University of Southampton who aim to automate the process of logging available raw data, (or *'nuts'*), that can describe aspects of one's personal experience. A number of squirrels have been developed in this process, and an ethos of the group is to preserve this raw data in order to retain any unforeseen potentials for exploitation and transcend

[1] Semantic Logger, http://akt.ecs.soton.ac.uk:8080/
[2] http://www.semantic-squirrel.org/

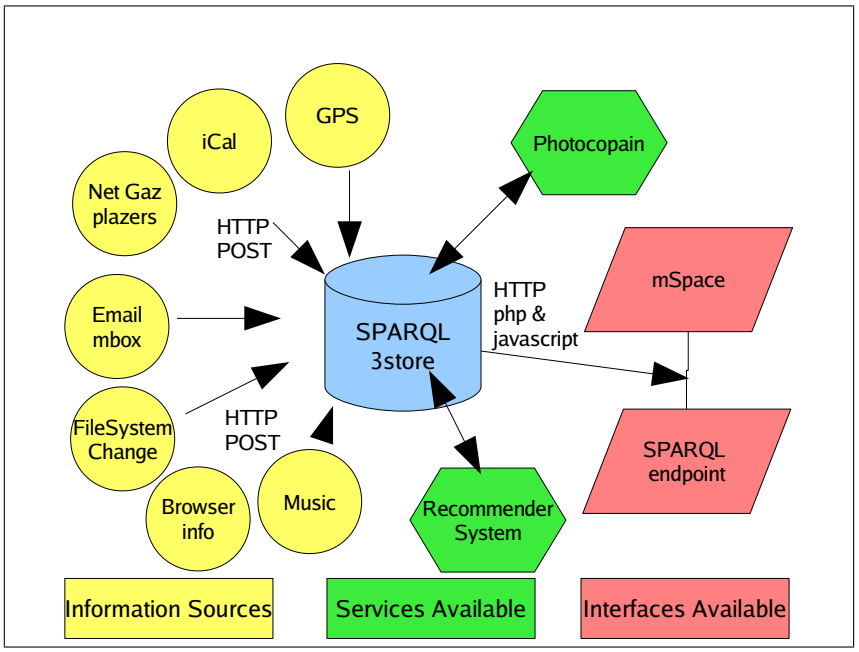

FIGURE 5.1: Overview of the Semantic Logger architecture, taken from [Tuffield et al. (2006)]

issues pertaining to platform and application restrictions. The SSSIG is also focusing on identifying novel systems using the collected data.

This raw data forms the basis of the knowledge acquisition phase for the Semantic Logger and is parsed into RDF representations. Effort has been put in selecting appropriate representations: they have been taken from proposed standards at the W3C or other standard making bodies, or have been selected due to current uptake on the web. Where such standards have not been available, we constructed local ontologies which describe the given phenomenon[3], while simplicity and generality maintenance have been paramount. The intent is to use raw data about people in order to build the context of a particular event at a particular time. By virtue of the fact that each event logged by the system is time-stamped and related to a FOAF URI, we are able to choose variable levels of granularity to describe its context.

Upon registration of a Semantic Log, a user is presented with tools that allow for the surreptitious recording of personal information. The set of information sources presented in figure 5.1 is far from an exhaustive one, and is not intended to limit the functionality of the system. The Semantic Logger has been designed in a manner to allow information, in various forms of RDF to be posted to the knowledge base (KB).

The system has a service-based architecture, and has been designed so that new services may join on an ad-hoc basis. The interactions between components have been implemented using HTTP requests, while the interactions with the central RDF triplestore make use of the SPARQL RDF query language [World Wide Web Consortium (2005)].

[3] http://akt.ecs.soton.ac.uk:8080/downloads.php

The system uses a Universal Resource Identifier (URI) to point to a user's FOAF file and each user's FOAF file is linked to their RDF data. This URI is subsequently used to log the provenance of all the information asserted in the Semantic Logger.

At the heart of the system is the AKT Project's [4] SPARQL-compliant RDF triplestore 3store [Harris (2005)]. The key role of the triplestore is to act as a persistent store for the system, and to mediate the interactions between the other system components. The main requirements in selecting an appropriate RDF Knowledge Base implementation were efficiency and consistency. 3store is a system benchmarked against other RDF storage and query engines such as Jena [McBride (2001)], Sesame [Broekstra et al. (2002)] and Parka [Stoffel et al. (1997)] and shown to outperform in terms of both efficiency and scalability [Streatfield (2005), Lee (2004)].

The interactions between Web Services have been implemented using HTTP requests, while the interactions with the central RDF triplestore make use of the SPARQL[5] RDF query language. The sources of information we have identified and integrated are rationalised by the nature of the services currently provided by the system, and are merely presented as inspiration for future development.

The sensitive nature of this metadata–chronology implied that the Semantic Logger had to allow users to decide whether any information logged was to be posted for public consumption or not. This guided the design such that each user has access to two distinct knowledge bases: one publicly accessible, the other private. Furthermore, users are allowed to select which aspects of their personal experience they would like the system to monitor.

5.1.1 Added value

Despite our intentions to develop a solid platform for evaluating present and future work, it can be argued that immediate added value emerges from the use of this system. Firstly, support for this argument arises from enabling the application of SPARQL queries on the available information, to answer questions that would be unfeasible under representations of singular domains, and also the added inferential capabilities that are enabled. For example, named entity recognition can be applied to email correspondence to identify closely related groups while co-authorship and co-reference between scholarly articles can be analysed as shown in [Alani et al. (2003)]. Co-location at various events can be inferred from geo-data and calendar entries, while the latter, in combination with the analysis of locally stored multimedia files (e.g. music and video files) can aid in identifying common interests.

[4] Advanced Knowledge Technologies, http://www.aktors.org/
[5] SPARQL, http://www.w3.org/TR/2004/WD-rdf-sparql-query-20041012/

In addition since information is represented in an RDF graph, by virtue of the representation there exist multiple dimensions in which the data may be indexed and viewed. The mSpace interface[6] has the ability to organise such data, in multi–pane browsers. Furthermore, the edges of the graph are allowed to be reordered using dimensional sorting, independent of the hierarchical nature of the representation, allowing for a number of such trees to be visualised and browsed. mSpace requires the definition of a *default column* and a *target column* along with the path, through ontological relationships (edges in the graph), between them to create a multi-columned re-arrangeable browser.

While in the past these had to be made explicit by the system engineer, the algorithm has been extended to enable the automatic deployment of the interface for arbitrary RDF fragments, eliminating the need for engineered visualisation models. As such users can dynamically explore their personal information space, without requiring any effort on their part.

5.2 Wikipedia

Rosetta stone

IR vs AI

Wikipedia [The Wikimedia Foundation Inc. (2006)] was identified as a rich external source of information, due to its wide coverage of subjects, maintained by their respective interest groups, [Giles (2005)]. This information may be deemed expert knowledge, and the web–graph spanned by it's articles and the links between them can be used as the universal vocabulary in our approach. In addition, the fact that articles are organised in categories provides opportunities for extracting some semantics for the recommendations made.

Since the whole graph would be too cumbersome to work with, the part relevant to the recommendation domain is extracted each time. This is done by identifying exactly one node with every resource in the domain and including them in the extracted graph. All nodes that link to, or are linked from these nodes are also included. The same process is carried out to obtain Wikipedia based representations for users, where the concepts identified by the Semantic Logger are used instead of domain resources. The rich, highly interconnected structure of wikis is considered ideal for this purpose.

[6] mSpace, http://mspace.fm/

5.3 Recommendation domains

The system produces recommendations by injecting the user into an external user community (or domain) and comparing them with its' members. In order to do this we obtain a projection of the user in the the target domain, simply by evaluating the intersection between the graph representing the domain, and that representing the user. This section describes the two domains used to evaluate the approach.

5.3.1 Last.fm

Last.fm is a commercial system that offers personalised music broadcasts to its users. It is the extension of the Audioscrobbler system, that was designed to record and expose the music its' users listen to. Preferences are represented by the number of times each track is played.

Data was collected for 5 964 UK users of Last.fm and 17 882 artists, by implementing components to interact with the Audioscrobbler Webservice API. 4 495 of these Artists were identified in Wikipedia, extracting a graph containing 158 273 articles in total, in the manner described above.

5.3.2 NetFlix

NetFlixBennet (2006) is an online DVD rental company, that allows users to rate the titles they watch (on a 1–5 scale) to obtain personalised recommendations. Their three million users produce over a billion ratings per year. In October 2006 company announced a challenge [7] to grant $1000000 to anyone who develops an algorithm that achieves a 10% improvement on their current predictive accuracy. They have released 100 462 465 ratings provided by 2 649 429 users about 17 770 movies. An additional 2 817 131 ratings were removed from this set, and were held as the test set.

This development has spurred great interest amongst the research community in the field which resulted in holding a NetFlix workshop for the participants in the challenge, at the ACM International Conference on Knowledge Discovery and Data Mining 2007. The interest expressed in this dataset provides an extraordinary opportunity for comparing this approach to other *state–of–the–art* recommendation schemes.

[7] http://www.netflixprize.com

Chapter 6

Algorithm evaluation

This chapter focuses on the various evaluation experiments carried out to ensure that the expected results were obtained at each stage of developing the framework. The following sections are arranged chronologically, since each experiment provided the motivation for further development and experimentation.

6.1 Singular valued decomposition and nearest neighbours

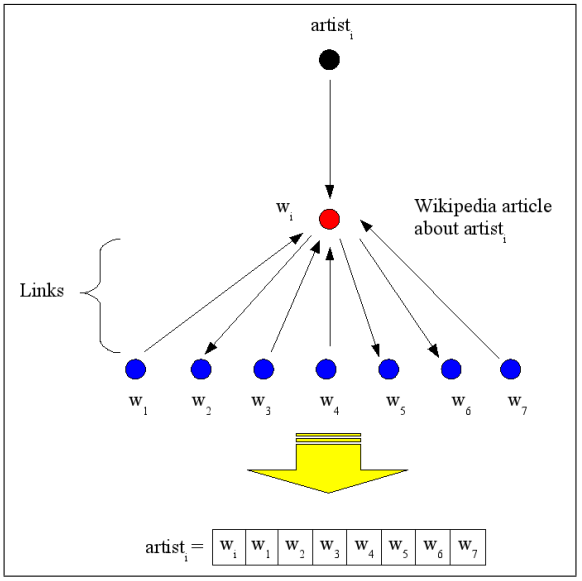

FIGURE 6.1: An Artist's feature vector based on the corresponding Wikipedia graph

First, we set out to assess whether the similarity between Last.fm *Artist*s, computed based on Wikipedia is an approximation of that obtained through recording playcount statistics. Such a finding can be used to support the argument for Wikipedia being an adequate approximation for a universal vocabulary.

Artists	**17882**
Artists with no Last.fm cluster	1120
Artists not found in Last.fm	528
Artists with no links to Wikipedia pages	13309
Working set	**4495**

k	***100***	***50***
Mean precision	**0.0655**	**0.0890**
Mean recall	**0.1025**	**0.0710**
Mean hits	6.5497	4.4523
Mean Kendall's τ	0.0868	0.0665
Corresponding Z-score	**1.9643***	**1.0760**
Critical Z value at the 10% level (two-sided) or 5% level(right-sided): ***1.64***		
Artists with > 0 *hits*	**3180**	**2937**
% of working set	70.75	65.34
Mean precision	**0.0926**	**0.1363**
Mean recall	**0.1449**	**0.1087**
Mean Kendall's τ	0.1226	0.1017
Corresponding Z-score	**2.7765***	**1.6468***
Random recommender		
Expected mean precision	0.000056	
Expected mean recall	0.000056	0.000028
Expected mean Kendall's τ	0	

FIGURE 6.2: Precision/Recall and Kendall's correlation coefficient analysis of the results achieved by applying 100-NN and 50-NN on feature vectors reduced via SVD.

A feature vector has been compiled for each artist, as shown in figure 6.1. Resources, in this case artists, are matched to wikipedia articles. The matching article, along with the articles that are connected to it via hyperlinks compose the feature vector. Singular valued decomposition was applied to the resulting matrix, reducing its dimensionality to 200 columns. The Euclidean distance between any pair of artists was then computed, to produce lists of the most similar artists. These were then compared to the equivalent lists produced by Last.fm. Figure 6.2 presents the results of this analysis.

While precision and recall are typically used to assess the quality of query results using labeled test data, their use in this context is not as well defined. Since the number of all hits is unknown, we are forced to assume that the list provided by Last.fm is exhaustive, which is clearly untrue. In addition, our choice of clustering algorithm explicitly defines the number of results that will be retrieved for any artist.

Kendall's correlation coefficient, τ, is a widely used statistic to compare two different rankings of the same variables and thus it was used to measure whether the 'hits' produced by k-NN are ranked in a similar manner as in Last.fm's lists of artists commonly played together.

We observe that the obtained precision and recall values are not large enough to suggest that the features collected are sufficient to reproduce the same clusterings that emerge through recording real users' listening behaviour. However, two observations provide the motivation for further evaluating the utility of the contextual features gathered. The order of improvement over the *'random recommender'*, and the fact that reducing the number of neighbours causes recall to reduce, while increasing precision and vice versa, as expected. In addition, it can be shown that both precision and recall monotonically increase as functions of the number of features available for each artist.

The values obtained for *tau*, however, showed statistically significant evidence at the 5% level (right-sided) of correlation between the lists of 100-NN and those provided by Last.fm, and also for 50-NN when artists with 0 hits were excluded from the analysis, reinforcing our beliefs about the quality of the collected features.

It should also be noted that both algorithms used in this experiment (SVD and k–NN) are not suitable for use when producing recommendations. Since the number of users (points) is typically much larger than that of resources (variables), the computational requirements of both algorithms explode. As such, the results of this experiment serve only as a justification for the continuation of this work.

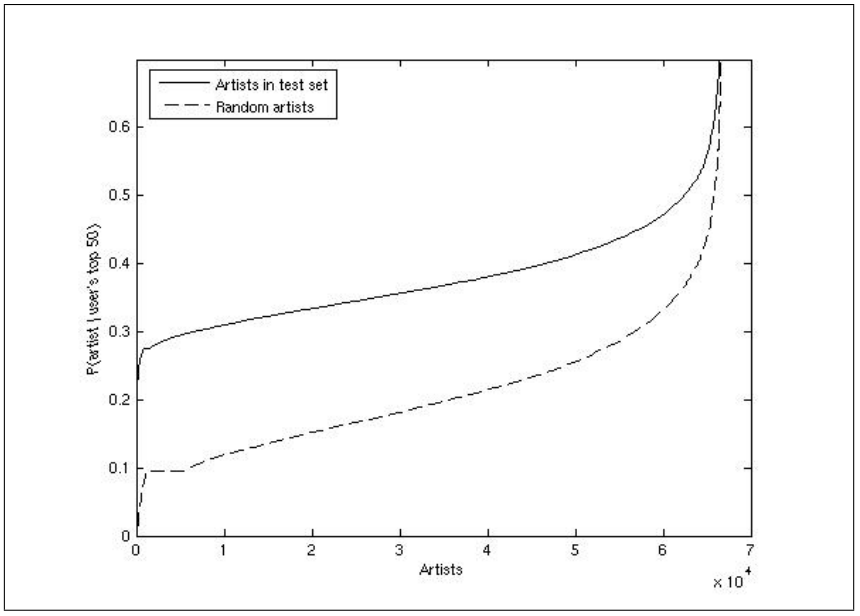

FIGURE 6.3: $P(artist_i|top50_j)$ for test and random sets. The data points are sorted in ascending order to better illustrate the difference in the posterior probability between *Artist*s in the test and training sets.

6.2 (Very) Naive Bayes

The lists of 50 most played artists for each Last.fm user were collected, and randomly sampled to obtain test sets containing 15 artists. The remaining 35 observations were used as training sets for a naive Bayes classifier. Feature vectors for artists were obtained in the same way as before, while users are represented as the concatenation of the vectors corresponding to the artists in the user's training set. This was carried out as follows:

$$P(artist_i|user_j) = \frac{P(user_j|artist_i)P(artist_i)}{P(user_j)}$$

$P(user_j|artist_i)$ expresses the probability of selecting the active user, given that $artist_i$ has been observed. This is defined as the inverse of the number of users with $artist_i$ in their training set.

Assuming conditional independence between the constituents of user and artist feature vectors, we use a product form to combine their probabilities and obtain $P(user_j)$ and $P(artist_i)$. The function ϕ maps a resource to it's feature vector.

$$P(artist_i) \propto \prod_{w_k \in \phi(artist_i)} (P(w_k)).$$

$$P(user_j) \propto \prod_{w_k \in \phi(user_j)} P(w_k)$$

The probability of observing an arbitrary Wikipedia page, w_k, was calculated by dividing the number of times it appears within our dataset with the total number of non–zero entries in user feature vectors.

As shown in figure 6.3, on average $P(artist_i|top50_j)$ is consistently higher for artists in the test set. In particular, recommending the 15 artists with the largest $P(artist_i|top50_j)$ gives $Precision = 0.4841$ and $Recall = 0.7333$ with respect to the test sets, and averaged over all users. These values are comparable with those reported in evaluating systems found in the literature.

6.3 Probabilistic Latent Semantic Indexing (PLSI)

The result presented in the previous section provide motivation for further investigating probability models, and dropping the rather general independence assumptions made. In particular, PLSI was identified by virtue of its solid formal grounding.

Probabilistic Latent Semantic Indexing, [Hofmann (1999)] is a probabilistic extension to standard Latent Semantic Analysis, [Deerwester et al. (1990)], and relies on decomposing the dataset using an aspect model rather than applying SVD, [Saul and Pereira (1997)].

The probabilistic nature of PLSA provides a solid statistical background and defines a generative data model.

An aspect model is a statistical latent variable model [Saul and Pereira (1997)] to associate an unobserved class, to each observation pair. In terms of a joint probability model it can be expressed as :

$$P(d, w) = P(d)P(w|d) \tag{6.1}$$

$$P(w|d) = \sum_{z \in Z} P(w|z)P(z|d) \tag{6.2}$$

where z represents the latent classes, while d and w encode for the observations in terms of documents and words respectively. To generate an observed pair (d, w) under this model, a document is first selected with probability $P(d)$, the latent class z with the highest $P(z|d)$ is identified and a word w is generated with probability $P(d|w)$ [McLachlan and Basford (1988)]. In terms of the approach described in this report, documents are identified with users, while words are represented by the connected nodes in the Wikipedia graph.

Two independence assumptions are made here:

- Observations (d, w) are generated independently. This is equivalent to the *'document–as–bag–of–words'* approach commonly used in information retrieval.
- Conditional independence is assumed, stating that given the latent class z the observed variables d and w are independent.

The estimated values of $P(d)$, $P(z|d)$ and $P(w|d)$ are approximated by following the likelihood principle and maximising the log-likelihood function:

$$L = \sum_{d \in D} \sum_{w \in W} n(d, w) * logP(d, w) \tag{6.3}$$

where $n(d, w)$ is the number of occurrences of word w in document d.

Expectation Maximisation (EM) is the standard procedure with which the maximum likelihood is estimated [Dempster et al. (1977)] for the complete data, including the unobserved variables. This is a two-step iterative process where the posterior probabilities, $P(z|d, w)$, for the latent classes, z, are computed in the expectation step. The model parameters are then re–estimated for the posteriors previously computed and available statistics from the training set in the Maximisation step.

Assuming that observations (d, w) are conditionally independent, the Expectation (E) step is a consistent definition of $P(z|d, w)$. Given the posterior probabilities for all latent

classes and the term frequency vectors for all documents, the Maximisation (M) step is defined by finding the maximum of the Likelihood function. This is achieved by partially differentiating 6.3 with respect to each variable and equating the result to 0.

The computational requirements of this process can be drastically reduced through the use of summarisation techniques, such as mrkd–trees, as shown in [Moore (1999); Verbeek et al. (2005)]. Even with the improvements such techniques provide, the requirement of recomputing the likelihood function at every time step coupled with the large number of latent classes required to reflect the scale of our datasets, make using PLSI unfeasible.

6.4 Multi–resolution kd–trees (mrkd–trees) as a summarisation technique

Although EM–based algorithms were judged inappropriate for our problem domain, the summarisation technique used to accelerate such algorithms was considered a valuable tool for reducing both the size and the sparsity of the datasets.

A mrkd-tree, introduced in [Deng and Moore (1995)] is a binary tree where each node contains a subset of the datapoints. All points belong to the root node and each node has two children that split the parent's datapoints between them. Each node also records sufficient statistics for its members:

- Number of points owned.
- Centroid of points owned.
- Covariance matrix of points owned.
- The hyper-rectangle bounding the points owned.

The tree is then built top-down, by identifying the widest dimension of the bounding hyper-rectangle at the current node and splitting in two using the middle value. This is the dimension where the dataset varies the most, at the current node. Any point strictly lower than this value are assigned to the left child node and the rest to the right one. There are four possible stopping criteria for building such trees:

- The tree is built exhaustively until we obtain a leaf node for each datapoint.
- The number of splits to be made is explicitly predefined.
- The maximum number of points contained in a leaf node is explicitly predefined. Any node with less point than the threshold becomes a leaf.

- A minimum threshold on the size of the widest dimension of the bounding box is explicitly defined and any nodes that fall below this are not split and become leaf nodes. A suggested value for this threshold is approximately 1% of the range of datapoints.

As a first experiment, we used the NetFlix dataset to build a 10 level deep mrkd–tree. This was evaluated by subsequently using it to assign the users in the test set to nodes in the tree, and the centroid of the node was used to obtain predicted ratings. The resulting accuracy closely approximated what would be achieved if the global average rating was used as the prediction for each movie. This indicates that the majority of the data was still contained in a single leaf node. The process of building a tree with leaf nodes containing strictly less than 100 000 nodes is currently underway, and the results will be compared.

6.5 Web graphs and Bayesian networks

Inspired by Kleinberg's Hubs and Authorities algorithm [Kleinberg (1999)], a decision was made to investigate any potential benefits in preserving and exploiting the structure of the graph obtained by considering a recommendation domain. Figure 6.4 gives a visual example of such graphs.

Members of the expert community are summarised in the leaf nodes of the constructed mrkd–tree; the multiplicity of these nodes provides probability estimates for selecting them. Having selected one such node, the probabilities of obtaining resources are estimated based on the rating assigned to them by the experts. For example, on a five point scale (as in the NetFlix case), we consider traversing the graph from an expert node to a resource rated 5 twice as likely as traversing to one rated 4. In the first instance, the assumption that edges linking experts to items that were rated 3 or lower should never be traversed was made, assigning 0 probability to such edges. Once a resource is reached, the graph may be traversed to any of the nodes connected to it, with equal probability.

In this representation, the recommendation problem can be seen as identifying the node in the graph that will most likely be reached, given the nodes that represent the user as a starting point. This is denoted $P(r_i|u_j, W)$, where r_i stands for the resource assessed for recommendation, u_j the active user, and W is the extracted graph. Bayes rule gives:

$$P(r_i|u_j, W) = \frac{P(u_j|r_i, W) * P(r_i|W)}{P(u_j|W)}$$

The graph is encoded in a probabilistic version of an adjacency matrix. This is a square matrix, with rows and columns for all nodes in the graph, where each cell encodes for $P(row|column)$. A high–level interpretation of this matrix is that it contains the

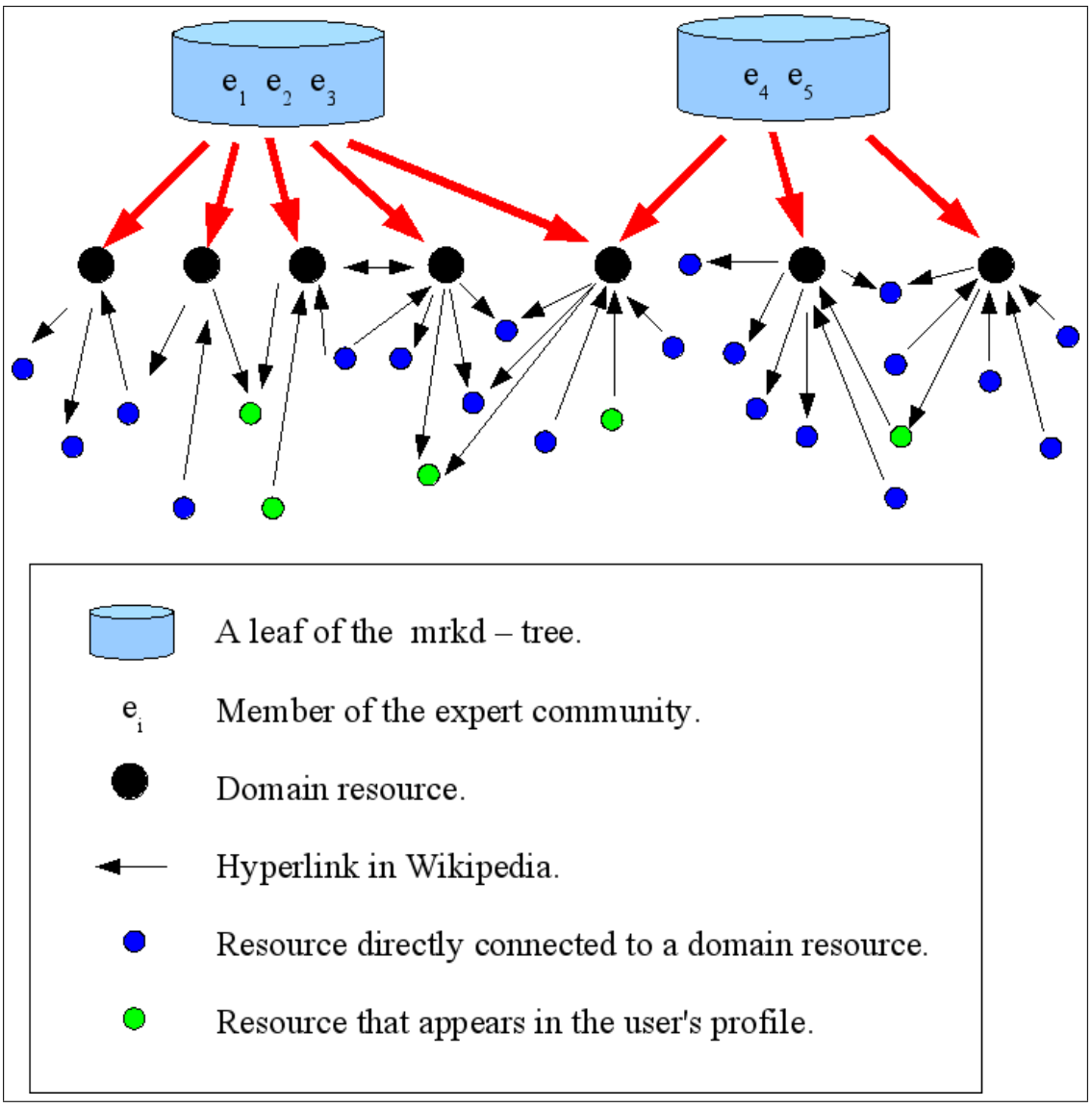

FIGURE 6.4: An example of the graph structures used.

probabilities for all possible traversals in the graph, in one 'hop'. The overall structure of this matrix is illustrated in figure 6.5.

Since we intend to use a Bayesian approach to assess recommendations, the need to compute the prior ($P(r_i|W)$) and marginal likelihood ($P(u_j|W)$) probabilities arises. These are interpreted as the probability of reaching r_i, or u_j independently of starting point and number of hops. The leading eigenvector is sufficient to provide estimates for such probabilities. This can be shown by taking the spectral decomposition of the matrix:

$$W = \lambda_1(v_1 * v_1^T) + \lambda_2(v_2 * v_2^T) + \cdots + \lambda_N(v_N * v_N^T)$$

This is then raised to an arbitrary power L, and $\lim_{L\to\infty} W^L$ is evaluated. It can be seen that since eigenvectors are orthogonal the intermediate terms become 0, while everything

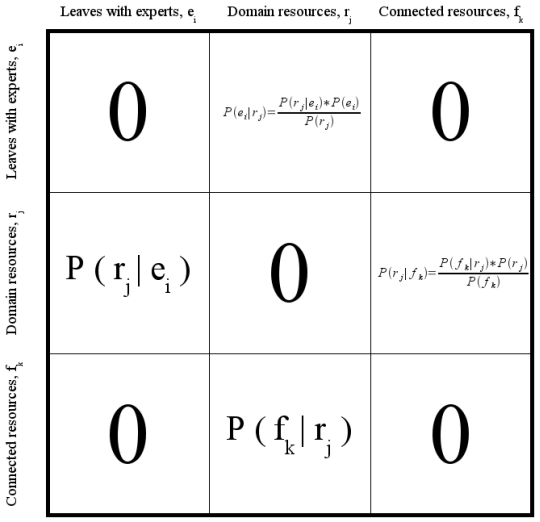

FIGURE 6.5: The overall structure of the probabilistic adjacency matrix.

other than the leading term is negligible in magnitude.

However, the way to compute the likelihood, $P(u_j|r_i, W)$ is unclear. As such we enforce the assumption that $P(u_j|r_i, W) = P(u_j|r_i)$, i.e. that r_i *d–separates* the resources in u_j's profile from the rest of the graph. It was found that the graph was not connected enough to support this assumption, and most posterior probabilities became 0.

6.5.1 A 'random surfer' model

Similar findings were reported in the work of Brin and Page[Brin and Page (1998)] in applying a similar approach to indexing the WWW. The inconsistent structure of the Web was identified as the cause of the counter–intuitive behaviour exhibited by the pure weight–balancing model. This was rectified by giving every page a small but positive connection strength to every other page, a concept often referred to as the 'random surfer' model.

A similar smoothing operation can be applied in the recommendation case, by allowing, with a marginal probability, the transition from a node representing a domain resource to any other Wikipedia node.

Chapter 7

Future work

In this chapter, directions for further development are outlined, and appropriate milestones are set forward for the completion of this project.

7.1 Further evaluation

The evaluation carried out to this point suggests that the approach presented in this report can identify resources expected to appear interesting based on a user profile. However, other aspects of the system remain untested and in need of experimental evaluation:

- A heuristic indicating the stopping criterion to be used when building the mrkd–tree should be identified. Such a heuristic would encode the trade off between accuracy and summarisation efficiency in the resulting tree.
- Using an appropriately fine grained tree as the baseline, the predictions made by applying the graph–based algorithm will be evaluated in terms of relative improvement.
- The NetFlix dataset will be sampled to produce a toy example, which will in turn be used to analyse in detail the mathematical properties of the graph–based algorithm. This analysis is expected to shed some light on the shortcomings of the approach, and help identify directions for improving the algorithm.

7.2 Mapping context to domains

As mentioned earlier, the contextual setting at the time of recommendation is to be evaluated in order to carry out the domain selection process. While PLSI was found to

be too computationally expensive for producing recommendations, it may be applied to this problem since the spaces involved are much smaller.

The task is to identify which domains are more likely to be of interest, given the current place and time. Documents can be identified with $< place, time >$ pairs, while words represent the domains from which the events occurring under this context originate (according to the structure of the universal vocabulary). This conceptualisation allows the identification of latent contextual classes, each defining a probability distribution over all possible recommendation domains. By classifying the current context using such a strategy, the system is able to select the domains most likely to be of interest in order to produce recommendations

7.3 Cross-domain recommendations

The power of this approach lies in the fact that an arbitrary number of domains may be considered, in order to identify the most relevant recommendation. In this way, explicit information about the active user's preferences pertaining to the resources in the domain is not required, and can be replaced by a more general log of the user's digitally visible life. It is hard to provide support for this claim by attempting to reproduce the ratings provided by users whose profiles only include information for one specific domain.

Instead, the Last.fm user profiles can be used to predict interest in NetFlix films and vice versa. Analysis can then be carried out to show whether, for example, films with large box office revenues are recommended to people that tend to listen to music in the charts, or other similar trends. The identification of such trends can provide positive evidence towards the suitability of the recommendations produced.

Data is also available for a small number of Semantic Logger users. In addition, a member of the SSSIG has offered full squirrel data over a two year period, for evaluation purposes. These datasets will be used to produce recommendations which will be evaluated in a user study.

7.4 Tags

The recent uptake in the use of 'free–form tagging' in a plethora of social networking sites poses intriguing questions as to whether these can be utilised as the features of resources for recommendation purposes. A graph can be constructed by assigning a node to each distinct resource and tag, with edges representing the fact that a tag has been using to describe a resource. The semantics associated with each tag provide another opportunity to build the graph using relationships such as synonym, hypernym,

etc. These are readily available from WordNet[Miller (1995)]. The results will then be compared with the ones obtained through the use of Wikipedia.

7.5 Time management

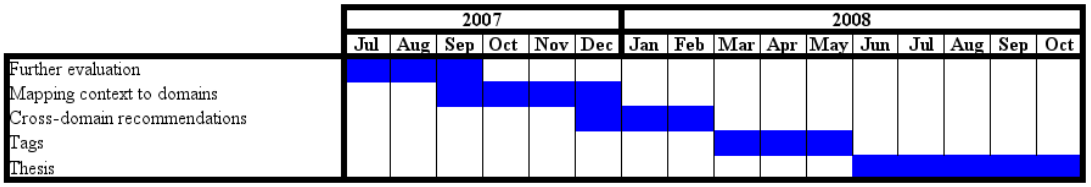

FIGURE 7.1: Gantt chart indicating the milestones associated with the activities identified in this chapter and their expected time of completion.

The Gantt chart in figure 7.1 shows the projected dates of completion for the tasks put forward previously in this chapter. The next three months will be spent finishing the evaluation for the graph-based algorithm. Once results are obtained, the aim is to achieve a journal publication.

Following from there, work needs to be carried out to integrate the Semantic Logger with the recommendation algorithm. The main issue lies with automatically identifying the user's domains of interest based on the data we have available. This is expected to introduce requirements such identifying and importing relevant expert communities to the system. Because of this, the duration of the task is set to four months.

Of course, evaluation needs to be continuously carried out to assess the various configurations of the system as new domains are imported. In addition, the ways in which the presence (or absence) of particular types of information in user profiles affects the performance of the system are expected to be of interest. These are the types of activities that will be carried out during the three months of the *"Cross–domain recommendations"* task.

The Spring of 2008 will be spent looking at ways of using tags, instead of having to resort to Wikipedia. This is preferable, since it is much easier to support the argument *"There is an English word for anything in the world"* than *"There is a Wikipedia article . . . "*. However, a hyper–link between two articles could carry much more latent information than a *synonym* relationship.

The final five months are set aside for writing up my thesis, as well as any (probable) delays in completing the other tasks.

Chapter 8

Conclusions

This upgrade report aims to motivate the consideration of an alternative paradigm for Recommender Systems. Traditional systems have been designed under the assumption that an exhaustive index of possible recommendations is available, and that users can be adequately characterised solely through their interactions with resources in this index. Instead, we argue that by compiling a semantic log of all user activities, a much more complete profile can be obtained. Items for recommendation are introduced to the system through importing preference data from external communities and social networks, enabling multiple domain recommendations. A translation phase is required in order to compare user profiles with members of such communities, in order to assess resources for recommendation. This is achieved through exploiting a *'Universal vocabulary'*, in which unique descriptors may be obtained for the concepts discovered in the user's semantic log, as well as for the resources considered for recommendation. Wikipedia was identified as one such domain, and experiments have been carried out to show that the descriptors obtained are sufficient to evaluate similarity between domain resources. Since recommendations may stem from a number of domains, a domain selection process is carried out in order to identify the most appropriate one(s).

By design, this approach circumvents a number of issues that hinder the performance of conventional RS, such as the *'cold–start'* problems, and the production of uninteresting, homogeneous recommendation lists. Furthermore, a unique opportunity for assessing the utility of potential recommendations based on the user's current contextual setting arises from the fact that events in the Semantic Logger can be cross–referenced, based on temporal and spatial information. Such events can be clustered together to identify different classes of context, and used to drive the domain selection process.

Further experimentation needs to be carried out, as put forward in the previous chapter, in order to identify potential pit–falls in the framework and directions for further improvements. The utility of the system as whole can then be comprehensively evaluated and analysed.

Bibliography

G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *j-TOIS*, 23(1):103–145, January 2005. ISSN 1046-8188.

G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.

H. Alani, S. Dasmahapatra, K. O'Hara, and N. R. Shadbolt. Identifying communities of practice through ontology network analysis. *IEEE Intelligent Systems*, 18(2):18–25, 2003.

M. Balabanovic and Y. Shoham. Fab: content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, 1997. ISSN 0001-0782.

J. Bennet. The case of netflix. pages 1–21, Bilbao, 2006.

A. Berenzweig, B. Logan, D. Ellis, and B. Whitman. A large-scale evaluation of acoustic and subjective music similarity measures. In *4th International Symposium on Music Information Retrieval*, 2003.

D. Billsus and M. J. Pazzani. Learning collaborative information filters. In *15th International Conference on Machine Learning*, pages 46–54. Morgan Kaufmann, San Francisco, CA, 1998.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

J. Broekstra, A. Kampman, and F. van Harmelen. Sesame: A generic architecture for storing and querying RDF and RDF schema. 2002.

R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002. ISSN 0924-1868.

H. Cunningham. GATE, a General Architecture for Text Engineering. *Computers and the Humanities*, 36:223–254, 2002.

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*, 41(6):391–407, 1990.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1): 1–38, 1977.

K. Deng and A. Moore. Multiresolution instance-based learning. In *Proceedings of the Twelfth International Joint Conference on Artificial Intellingence*, pages 1233–1239, San Francisco, 1995. Morgan Kaufmann.

EXIF. Exchangebale image file format for digital still cameras: EXIF version 2.2. Technical report, Japan Electronics and Information Technology Industries Association, 2002.

J. Giles. Internet encyclopaedias go head to head. *Nature*, 438(7070):900–901, December 2005. ISSN 0028-0836.

D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992. ISSN 0001-0782.

T. R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands, 1993. Kluwer Academic Publishers.

Stephen Harris. SPARQL query processing with conventional relational database systems. In *WISE Workshops*, pages 235–244, 2005.

T. Hofmann. Probabilistic latent semantic indexing. In *22nd Annual ACM Conference on Research and Development in Information Retrieval*, pages 50–57, Berkeley, California, 1999.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

Last.fm Ltd. Last.fm. *http://www.last.fm*, 2006.

Ryan Lee. Scalability report on triple store applications. Technical report, MIT, 2004.

G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

B. McBride. Jena: Implementing the RDF model and syntax specification. In *SemWeb*, 2001.

G. McLachlan and K. E. Basford. Mixture models. *Marcel Dekker, INC, New York Basel*, 1988.

S. E. Middleton, D. C. De Roure, and N. R. Shadbolt. Capturing knowledge of user preferences: ontologies in recommender systems. In *1st international conference on Knowledge Capture*, pages 100–107, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-380-4.

George A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11): 39–41, 1995. ISSN 0001-0782.

A. Moore. Very fast EM-based mixture model clustering using multiresolution kd-trees. In M. Kearns and D. Cohn, editors, *Advances in Neural Information Processing Systems*, pages 543–549, 340 Pine Street, 6th Fl., San Francisco, CA 94104, April 1999. Morgan Kaufman.

M. J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999.

M. J. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313–331, 1997.

D. M. Pennock, E. Horvitz, and C. L. Giles. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In *AAAI/IAAI*, pages 729–734, 2000.

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*, pages 175–186, New York, NY, USA, 1994. ACM Press. ISBN 0-89791-689-1.

P. Salembier and J. Smith. Mpeg-7 multimedia description schemes, 2001.

L. Saul and F. Pereira. Aggregate and mixed-order Markov models for statistical language processing. In Claire Cardie and Ralph Weischedel, editors, *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 81–89. Association for Computational Linguistics, Somerset, New Jersey, 1997.

A. I. Schein, A Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *25th International ACM Conference on Research and Development in Information Retreival*, 2002.

Kilian Stoffel, Merwyn G. Taylor, and James A. Hendler. Efficient management of very large ontologies. In *AAAI/IAAI*, pages 442–447, 1997.

Michael Streatfield. Report on Summer Internship Work For the AKT Project: Benchmarking RDF Triplestores. Technical report, University of Southampton, 2005.

A. Swartz. Musicbrainz: A semantic web service. *IEEE Intelligent Systems*, 17(1):76–77, 2002.

The Wikimedia Foundation Inc. Wikipedia: The free encyclopedia. *http://wikipedia.org*, 2006.

M. M. Tuffield, A. Loizou, D. Dupplaw, S. Dasmahapatra, P. H. Lewis, D. E. Millard, and N. R. Shadbolt. The semantic logger: Supporting service building from personal context. In *Proceedings of the 3rd ACM Workshop on Capture and Archival of Personal Experience, ACM Multimedia*, 2006.

J. J. Verbeek, J. R. J. Nunnink, and N. Vlassis. Accelerated EM-based clustering of large data sets. The Netherlands, 2005. Kluwer Academic Publishers.

World Wide Web Consortium. SPARQL query language for RDF, working draft. Technical report, World Wide Web Consortium, 2005.

W. Yang, Z. Wang, and M. You. An improved collaborative filtering method for recommendations' generation. In *IEEE International Conference on Systems, Man & Cybernetics*, pages 4135–4139, 2004.

C. N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *World Wibe Web*, pages 22–32, 2005.