Bachelor of Science

In

Computing and Artificial Intelligence

Final Year Project

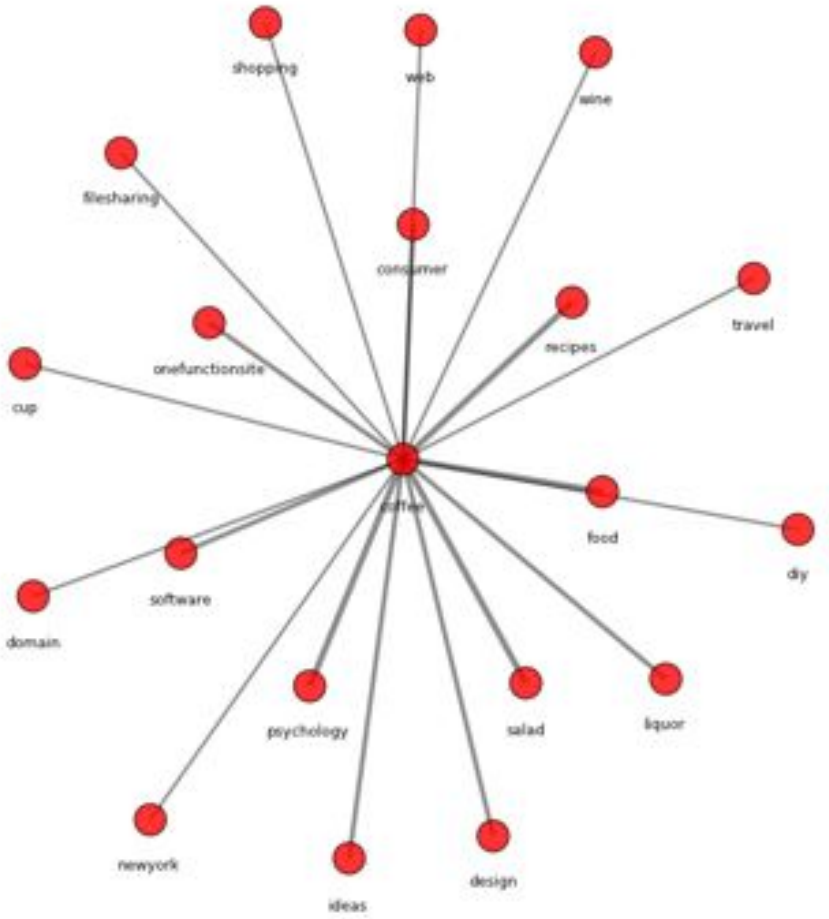
REPRESENTING INTERESTS AS TAG GROUPS

A User Modelling Tool for Recommendation Systems

Author:
Risto M J Lyra
Candidate Number: 31045

Supervisor:
Dr. Luc Berthouze

Wednesday May 4^{th} 2011



Statement of Originality

This report is submitted as part requirement for the degree of Bachelor of Science in Computing and Artificial Intelligence at the University of Sussex. It is the product of my own labour except where indicated in the text. The report may be freely copied and distributed provided the source is acknowledged.

Matti Lyra
Wednesday May 4^{th} 2011

Acknowledgements

While the project itself is my own work the motivation for the project came from working for 15gifts Ltd. during the Spring of 2010. During the Summer of 2010 while participating in the Junior Research Associate (JRA) bursary program I discussed and perfected the project plans with my JRA and Final Year Project supervisor Dr. Luc Berthouze. I would like to thank both Tom from 15gifts Ltd. for giving me the opportunity to work on their recommendation engine and Dr. Luc Berthouze for his invaluable input and support before and during the project.

I would also like to thank my girlfriend Sara for her support and understanding during the project.

Summary

This project was motivated by working for an online startup company designing a recommendation engine during the Spring of 2010. The lack of causal representation in existing recommendation systems became a difficulty during that project. For instance popular recommendation systems such as that of Amazon.com lack the ability to represent whether a product has been purchased as a gift or for personal use. The user profiles are also typically represented as a subset of the product catalog offered by the service. This along with discussions with Dr. Luc Berthouze led to the conclusion that a kind of semantical representation system could be built using tag information in social tagging systems. This project's aim was to research the theoretical possibilities of using tag information to build generalised interest descriptions and representing users as a set of keywords describing their interests. Potential practical application are identified in the discussion.

Social tagging systems have been researched previously [24, 23, 25, 30] and have been shown to be able to produce good recommendations and to offer more flexibility for how recommendation engines are used. The user profiles though still have been represented as a subset of the product catalog or tag set leading to a cold start problem where new users can not be offered any recommendations because of a lack of usage history with a service. The project addresses this issue by building a generalised interest space independent of any one service and mapping users and products onto that interest space to produce recommendations. This allows the users to be described by their interests.

The proposed method is validated against a simulated data set and initial results are presented also for real world data collected from delicious [1]. It is shown that the method can recover a set of hidden interests assigned for users in the simulation. On the real world data set the method is shown to produce semantically coherent groups of tags that can be used to represent interests. Methods for comparing the similarity between different tag groups and users or resources and tag groups are presented.

Initial results from the method show that using a generalised tag based description of user interests to represent the user information is a viable option and offers more flexibility for recommendation engines than representing the user as a subset of the product catalog. Further research possibilities are identified for researching how the similarities between tag groups could be improved, what pre and post processing could be done to the data to clean it and to apply the method for practical applications.

Contents

1. Introduction

1.1. Aims

This project is concerned with user modelling methods in the context of online recommendation systems. The aim is to explore a way of representing user interests through tag information in an abstract model that is independent from any one recommendation engine. As the amount of data in online services has grown is has become necessary to create automatic recommendation systems that through modelling user actions produce recommendations for new products to buy, websites to view or books to read, depending on the service. The size of the product catalogs and the specifics of how the users are modelled has created several problems however. Modern recommendation engines commonly use item-to-item similarity to represent users. This method does not allow representing causal relationships between product purchase and a user. For example a product bought for personal use and one bought as a gift.

The approach taken in this report uses keyword information users themselves have assigned for some resources to build a generic model of interests and how they relate to each other. Real world data collected from an online social bookmarking service, delicious [1], is used alongside simulated data. Previous research on tag based recommendations has focused on modelling users or resources as a set of tags, whereas this project is concerned with modelling the tags themselves according to their usage patterns. This project takes a novel approach to the user modelling problem and aims to provide a framework and objectives for further research. The primary objectives of this project are

- Survey the related research to gain an understanding of the research field and to guide the implementation of the methods.
- Test the hypothesised model on a data simulation to verify that the predicted behaviour is produced.
- Collect real world test data from the social bookmarking service delicious.
- Analyse the tag information contained in the delicious dataset to determine whether the tags contain enough information for the method to work.
- Explore methods for producing recommendations and analyse the feasibility of the approach.

The report is structured as follows: the rest of the introduction focuses on the relevant history of recommendation systems (Chapter 1.2) and the related concepts of social tagging (Chapter 1.3) and the statistics of tag distributions (Chapter 1.4). These are provided to give the reader an understanding of how the project fits in with the research and application field of recommendation systems.

Chapter 2 addresses the professional considerations relevant to this project and references the British Computer Society's Codes of Conduct[8] and Good Practice[7] where appropriate.

Chapter 3 covers the requirements analysis and gives an analytical view of the related tag based recommendation system research.

Chapter 4 is the main body of the report containing the system design and is further divided into subsections. Chapter 4.1 describes in detail the proposed model and outlines how the data in delicious is to be used to achieve the goals. The following two chapters (4.2 and 4.3) describe how the data collection from delicious was done and how the HTML parser works. The database design is described in Chapter 4.4. Chapters 4.5 and 4.6 describe the clustering algorithm used in the project and the simulation built to provide validation data. Finally Chapter 4.7 describes how the model can be used

to produce recommendations and gives a comparison between the proposed model and a collaborative filtering approach.

1.2. History and Motivation

Choice is an important aspect of modern life, and is commonly regarded as a defining characteristic of capitalism. The advent of online services has exponentially increased the amount of choice by allowing retailers to break free from the physical constraints of high street stores. As no shelf space for books or movie titles is needed more products can be included in the offered catalog. Ever increasing the catalog size is not enough to increase sales as customers will also need to know about the increased diversity in the offered product range. Finding products is a non trivial task given a suite comprised of tens of millions of individual items. To ease finding relevant products automatic recommender systems have been developed. These systems rely on statistical information extracted from user behaviour and purchase history. Online stores are able to gather more user data and to aggregate that data across whole populations in ways not possible before. Compared to a brick and mortar supermarket or high street store it is much easier to monitor the order in which products are viewed and consequently bought, collect information about how satisfied people were with their purchase and combine the shopping and browsing habits of tens of millions of users.

Goldberg et al. [12] pioneered the field in 1992 with a system called Tapestry. The aim was to help people cope with an increasing amount of emails arriving daily. At the time the best solution alleviating overflowing inboxes were electronic distribution lists focused around certain specific interests. Distribution lists, or newsgroups, have a human moderator who selects emails that get sent to subscribers. Goldberg's idea was to allow users to implicitly self moderate the lists instead, an approach he named collaborative filtering. Put simply each user would annotate or rate emails and other users would be able to see these annotations and filter their own view of the mailing list accordingly.

Tapestry was very much a proof of concept design for a collaborative filtering system in general. It was followed by similar systems applied to other problem domains from recommending web resources based on UseNet news [31] or browser histories [28] to making sense of social networks within an organisation in order to find the right people within it [16].

Applying collaborative filtering systems to different problems and to ever bigger data sets some problems started to arise. Sarwar [29] mentions data sparsity and scalability. Collaborative filtering builds user profiles by recording the ratings users give to products they view. Recommendations are given based on the overlap of liked items among different users. With a product catalog of two million products — not uncommon for modern e-commerce websites — a user would have to go through 20,000 products to cover 1% of the catalog. With different users only covering a tiny portion of the whole catalog, no overlap between user profiles can be computed because users are looking at different products. Furthermore new users have no rating history at all so they are overlooked by the whole system.

The scalability issue has to do with a growing number of users as well as products. The user neighbourhood computation is often done with a k-nearest neighbours algorithm [29] where the products liked by `k` number of most similar users are used as the recommendations. To get the list of the `k` most similar users the similarity to all users must be computed. The computational costs are far too high for a large enough user base. Instead of relying on the user neighbourhood Sarwar suggested calculating the item neighbourhood. This changes less frequently and would therefore not need to be updated as often. The algorithm would take the items a user has rated as a description of that user's likes. The predicted

rating of an unrated item for user `u` is governed by the ratings other users have given that item, weighted by the similarity of the shared products between the users. The problem with this approach is that using the already liked items as a template for new items creates a very strong bias to a specific type of item, a problem highlighted already in 1997 by Balabanovic [5]. If a user has bought a bicycle she most likely will not want to buy another and might be interested in non cycling related products as well.

Despite problems recommendation systems were applied in the hope they would increase sales by helping users discover new content more suited to their individual needs and desires. Service providers hoped to capture the long tail of the user distribution by increasing variability across the whole user space and personalising the services to individual user needs instead of offering the most common denominator between all users. This was a significant shift away from the business models developed during the previous decades. Anderson [9] argued that the 21^{st} century would mean "selling less of more" and that more and more consumers would discover themselves in niches instead of the mainstream. He based his claim on the observation that product awareness and purchase patterns followed a power law. A power law is a distribution with a long and fat tail (for more details see Figure 1 and Section 1.4). What it means is that the less likely events, or the niches, are more common than would be expected given a normal distribution (see Figure 1). For economics of scale it means that the aggregate sales achieved from the niches (the tail of the distribution) can outweigh the sales achieved from targeting the hits. With shelf space no longer a restriction targeting the niches became a viable business model.

Recent studies [11] have shown however that the commonly used models for recommendation systems suffer from a concentration bias towards the already popular. Fleder [11] ran probabilistic simulations of purchase behaviour based on well known economical user models and found that the most widely used recommendation system (collaborative filtering) suffers from this exact problem, even if the most popular item is discarded as an obvious recommendation. The more something is bought the more it gets recommended. A never ending positive feedback loop is created from the dynamics of user behaviour and the inner workings of the recommendation algorithm. As Fleder demonstrates the problem is not only to do with the algorithm but the way users are modelled in the system. Instead of being an entity separate from the service she is treated as a subset of the product catalog. The only way out of the positive feedback loop is if users themselves explore new areas of the product catalog — the very thing recommendation systems were developed to do. A further complication is that the user descriptions will always depend on the product catalog of the service. This can create the false belief that the service is offering all the variety users want.

Different approaches have been put forward to address the user modelling problem. Balabanovic [5] suggested a hybrid approach, Sarwar [29] an item to item similarity measure which Amazon.com uses [19] and Aggarwal [4] paved the way for graph theoretic approaches picked up later by Mika [24], Hotho [15], Michlmayr [23], Yeung [20] and Martinez [21] in the context of tag based recommendation systems. These tag based systems rely on a new kind of web based service known as social tagging. The dynamics of social tagging services have been analysed by Wetzker in [34] and Mika [24]. For a discussion on the research see section 3.1 on page 12 and section 1.3 for an explanation on social tagging.

1.3. The Wisdom of Crowds and Social Tagging

Social tagging services are online services that allow users to post resources for central storage and for other people to see. Popular ones include photo sharing service Flickr [2] and social bookmarking service del.icio.us [1]. The key observation about these services is that users can annotate the resources with

keywords called tags they themselves find useful for describing the content they have posted. The users are free to use any tags they want and find helpful. Collectively the tags used to describe a resource form a distributed description of that resource dubbed a folksonomy by Mathes [22]. He argues that folkosonomies are a form of categorisation as opposed to classification. Where classification tries to construct a *formal* hierarchical taxonomy for a given domain the point of a folksonomy is to be *informal* and to allow the semantics and hierarchy to arise from the distribution of words used, ie. the semantical structure (if there is one, see 1.4) emerges from the interactions of users' annotations. Furthermore because folksonomies are not an attempt to formalise a domain the categories that do arise often overlap and are not mutually exclusive. Folksonomies are not created — solely — by industry experts. The barrier of entry to using one is therefore lower than for formal taxonomies. The user does not need to learn the intricacies of class differences and is encouraged to mould the folksonomy to suit her own needs. Intuitively this means that folksonomies do not capture the deeper knowledge structures of a domain but instead are broad definitions. They do however capture emerging trends and unforeseeable phenomena in society precisely because of their openness and bottom up structure. As an example of this Mathes mentions the tag 'sometaithurts' (so meta it hurts) found in Flickr which is used to describe self referential images such as a screenshot of Flickr posted on Flickr. It is hard to see how that would fit into a knowledge engineered taxonomy.

The user specific tag sets, called personomies, over time become a freeform description of the user that is not directly dependent on any one service's product catalog and is formed completely bottom up by the user herself. It is this aspect of social tagging that the previous research has focused on (see section 3.1).

1.4. Statistics of Tag Distributions

The previous section described social tagging services. This section will focus on the analysis of and information captured by folksonomies. The utility of a folksonomy can be seen from two complementary viewpoints. The users' aim is to divide the world (or at least the resources they use) into some structure comprehendible to them privately, the folksonomy itself on the other hand needs to maintain a level of descriptiveness and differentiation to be useful at all. A tag such as 'website' has little informational content as it can be used to describe any web based resource. In [13] Halpin looked at how the tag distributions in del.icio.us folksonomies develop over time and whether they stabilise at some point to form a coherent and mutually agreed upon description of a resource. Mathes [22] had hypothesised earlier that the tag distribution in folksonomies would follow a power law. Halpin was able to confirm this by analysing datasets that covered up to 3 years of tagging behaviour for individual resources. He discovered that after an initial period of high variation in tag usage per resource the distribution would invariably start to settle down and eventually stabilise. During the lifetime of this process there were spouts of destabilisation where users introduced new tags, but despite this momentary destabilisation the final distributions followed power law statistics.

As explained by Newman in [27] if a distribution follows a power it means that the likelihood of observing a value for a given variable varies as the inverse of some power of that variable. When plotted on a graph the distribution is biased towards the right hand side. This is illustrated in figure 1. This is important because Halpin's study showed that the distribution of tags in a folksonomy tends towards a power law distribution as the folksonomy grows older. It means that the most common tags used to describe a resource are used many times more often than the less common ones. It also means that

the variance among the less commonly used tags is large. The folksonomy self organises into a shared knowledge base about a particular resource with a few very distinctive tags and a whole bunch of less distinctive but relevant tags that some users considered to be important.

Power law distributions once established are scale free, meaning that as more and more tags are added the ratio between the most common and the least common remains roughly the same. Note however that scale free doesn't mean that the folksonomy cannot break out of a power law distribution. It was shown to do so in Halpin's study [13] during short periods where new tags were added, but despite of this breakout the distribution returned to follow a power law. This observation is important because it shows that over time the folksonomies stop being random collections of words and start to have a structure. Extracting this structure and using it to construct a model of user interests is the aim of my study.

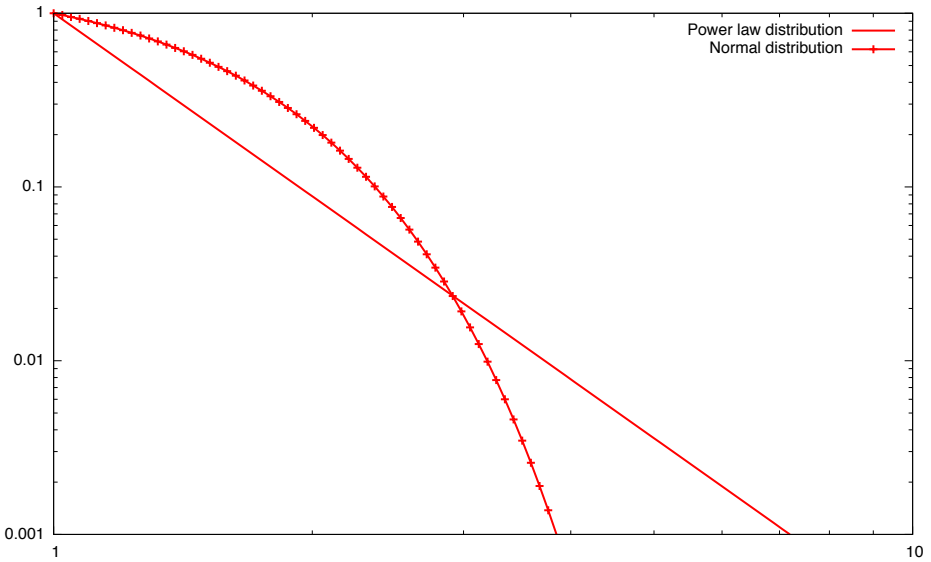


Figure 1: A power law distribution (solid line) and a normal distribution (ticks) on a log log scale. Note that even though the normal distribution is initially higher than the power law it quickly converges to 0 while the power law declines much slower.

2. Professional Considerations

Ethical considerations including those set forth by the British Computer Society in Code of Conduct for BCS Members [8] and Code of Good Practice [7] have been taken into account during this project. The following is a detailed list of relevant issues.

- Data collection and software provided by third parties is needed for the completion of this project. In accordance to Code of Conduct [8] sections 2 and 3 the legitimate rights of third parties and intellectual property laws both foreign and local will be respected.

  Data collection from a publicly available website (http://www.delicious.com) is needed for the completion of this project. The nature of the data is information about what keywords people use to describe specific web resources. While the keywords are linked to specific user profiles the profiles themselves are not meant to be identifiable to specific people. Many usernames however are the person's real first and last name. In accordance to Data Protection and Privacy legislation all user names will be obfuscated and all reported data will be anonymised so as to hide user profiles completely. No data will be shared with third parties, with the exception of the research results presented in the project report.

  The data will need to be collected with a crawler script. In the scope of this project the crawler script will be limited to only working within the specified web domain (http://www.delicious.com). It is however easily modified so that other web sources, possibly ones containing personally identifiable information, could be collected as well. While I cannot prevent people from misusing publicly available software, I am aware that the data collected and code produced in the scope of this project could, with modifications, be used to extract personally identifiable information and discover people's identities. It is not the aim of this project to do so.

- The collected raw data and the information acquired through manipulation of that data could potentially be of interest for advertising, marketing and other user profiling purposes. As required by section 5 of Code of Conduct [8] the data or other information obtained during the project will not be sold or given away to third parties in exchange for any commodities or services with the exception of the research results presented in this project.

- As stated in section 6 of Code of Conduct [8] the requirements of the University of Sussex will be adhered to, including but not limited to the use of computing resources provided by the University of Sussex. Most notably the computing resources offered by the University of Sussex are used in a manner that does not prevent others using the same systems.

- A dissertation project is a challenging project to complete and the University of Sussex has set strict time limits for the completion of each stage of the project. Section 7 of Code of Conduct [8] requires work to be completed on time and on budget. Care will be taken to submit all work in time and as specified.

- Care will be taken that the evaluation of research results will be unbiased and fair and that all information needed to evaluate those results is provided in accordance to section 9 of Code of Conduct [8].

- Even though the dissertation is to be completed alone, consultation with members of staff of the University of Sussex and other professionals will be conducted. Whenever these interactions take

place they will be conducted with integrity as stated in section 11 of Code of Conduct [8].

- Machine learning, artificial intelligence and online social networks are a young and rapidly developing area. Proper care will be taken to maintain awareness of technological developments in the field during the project. Code of Conduct section 14 [8].

- As stated in section 15 of Code of Conduct [8] misrepresentation of professional competence will not be given. I have worked as a professional web developer before starting the degree and have therefore knowledge of web services, databases and database design, HTML parsers and the HTTP protocol. I am familiar with the Python programming language and have completed projects of varying scale before this utilising the language. I have enough knowledge of machine learning techniques and natural language processing from my degree to complete the project.

3. Requirements Analysis

The proposed project is a research oriented project in an area of rising interest. The aim of the project is to produce a theoretical contribution to recommender systems used in online environments and to scope out further research possibilities. The design of the system is guided by previous related work in the area, my own needs for completing the project and considerations for continuing the research after this project has finished.

This section first gives a critical analysis of the state of the art in recommender systems and outlines the proposed solution put forward in this paper. The functional requirements of a reference implementation are given in 3.2.

3.1. State of the Art and Related Research

Recommender systems have been developed to help users find content and have been successfully applied commercially and non commercially to various domains including music, movies, research papers and online retail. The big promise of recommender systems has been to offer better personalised experiences to users and to shift sales focus towards the niches away from blockbuster hits. The most widely implemented recommender system models are collaborative filtering [12] [18] and content based filtering [5] [19]. The methods are both based on the assumption that users who liked items they have previously viewed will have similar correlations for new items. In collaborative filtering the similarity metric between users is the history of liked items between the users and for content based filtering the metric is the similarity of liked items in user profiles. Both methods therefore are based on comparing the usage histories of people and finding similarities in those.

Herlocker [14] conducted a survey of collaborative filtering algorithms in 2004 and identified a number of metrics and user tasks that should be considered when evaluating a particular approach. The tasks include `annotation in context` and `find good items`. These can be summed up as annotating search results with an expected value or recommending a certain set of products. Notably however such tasks as `recommend sequence` for recommending, for instance, an ordered list of music or academic papers to read and `just browsing` for a pass time activity or casual discovery of new items were found to be poorly addressed by the systems present in 2004. Furthermore the metrics used to evaluate an algorithm were found to be focused on accuracy of the recommendations. Novelty and serendipity were not part of the evaluation. The premise of my work is that many recommendation systems would greatly benefit from these. Kawamae [17] and Zhou [35] both also make the case for this. Recommending the obvious achieves high accuracy but fails to provide any useful information for the user. Herlocker's example of this is a grocery list recommender recommending bananas. Bananas are hardly going to be a novel recommendation for any modern shopper and focus should be given to novel items instead.

In 2009 Fleder et. al [11] conducted a theoretical study of various collaborative filtering algorithms and showed that they increased sales concentration. This is opposite to the promise of recommendation systems — namely to capture more of the niche users and products — and bad for the online retailers as most of the product catalog is underutilised.

In this paper we argue that these problems stem from the user model of collaborative filtering. The user is treated as a subset of the product catalog and a usage history with that catalog is required to produce any recommendations. This is known as the cold start problem where nothing can be offered to new users and new products cannot be recommended. Collaborative filtering has a strong bias towards the already known and popular [11] and is unable to produce variance in the offered recommendations.

Zhou [35] et. al have proposed an approach based on heat dissipation models to address this issue. Their results show that improving diversity of the recommendations can also improve the accuracy of the recommendations but requires fine tuning the parameters of the models.

More recently tag based recommendation systems have been studied [24, 23, 20, 34, 21, 25, 30]. These systems are based on triplets of (`user,tag,resource`) and use the distributed semantic information contained in folksonomies to produce recommendations. Folkosonomies [22] are created in social tagging services where users share resources such as web pages or images with each other. The shared resources are annotated with keywords called tags to help retrieve those resources at a later stage. So far the research has mostly focused on improving recommendations without changing the underlying user model of collaborative filtering.

Mika [24] conducted an investigation of the hypergraph formed from the (`user,tag,resource`) triplets and showed that by splitting it into three bipartite graphs — describing the relationships of users and tags, resources and tags and users and resources — a wealth of semantic information can be extracted about the relationships between the users, the resources and the tags. The data however was still only considered as bipartite relationships between users and tags and resources and tags, where as our claim is that a tag is defined *both* by the user and the resource the tag is given to. Song [30] applied Gaussian Processes and Poisson Mixture Models to improve the performance of document centric systems for recommending tags. He also criticised user based models as too cumbersome and nondynamic.

Yeung et. al [20] and Michlmayr [23] have specifically looked at integrating a diverse set of user interests into one profile. Yeung et. al studied the tag sets of specific users called personomies and used a community discovery algorithm detailed by Newman in [27] to extract the different user interests. The study showed their system to be technically feasible. Michlmayr gave three different approaches, all based on tag usage frequencies and reports favourable results from a small user study of their system. Even though the user model was changed to one based on user interests expressed through tags, both systems suffer from the same history problem as collaborative filtering. Without a usage history in the social tagging service the user interest profile cannot be extracted.

To address the issue of users having to accumulate a usage history while maintaining the ability to represent a diverse set of user interests, this paper proposes a system that constructs an interest space from folksonomies using similar methods to those described by Mika [24] and Yeung et. al [20]. The hypothesis is that by looking at the tagging information of all users, instead of one specific user, and computing the dependencies between tag clusters it will be possible to create a model of user interests, instead of one of user purchase or viewing history. New users can be mapped onto this interest space by asking them to provide a short keyword list of their interests. Notably the keyword list provided by the users will be of their own choosing and not limited by the interest space modelled by the recommendation system. For a thorough discussion of the model see Chapter 4.1.

3.2. Functional Requirements

The model put forward in this paper is an attempt to get around the problem of modelling user interests without requiring an extensive usage history. The main research goal is to test if the distributed semantic information contained in tag clouds is a viable model for user interests. As the information would allow building an abstract interest model it would allow user profiles to be portable between different online services and could serve to standardise user modelling between different services. To this end a collection of annotation triplets is required.

The research goal was tested with two datasets. A simulated dataset was created in order to test how well the proposed method recovers the latent interests of users and under what conditions. The other dataset was real world data collected from a popular social bookmarking service delicious. Delicious was selected as the data source because of its popularity both among users and other researchers. Delicious provides data in an easily accessible format and has a relatively diverse set of resources supporting the aim of fitting together diverse user interests.

In the context of this paper user interest refers to an unknown variable that governs the tag usage in annotating resources. The assumption is that users save resources in the first place because those resources match their interests and that tags are assigned to the resources according to what the interest or interests towards that resource is. For the simulated data set however the definition of an interest was simplified to only govern tag assignments. All resources were equally likely to be viewed regardless of user interest.

Collecting the data required developing a web crawler to harvest HTML resources from delicious. To allow a large dataset to be collected, the web crawler had to be highly automated and able to run for days without intervention. The HTML resources were then parsed and the relevant information stored in an easily accessible format.

The data processing and model validation has the following requirements:

- A similarity measure for tags should be defined.
- A weighted graph of tags and connections between the tags should be computed from the data collected from delicious.
- The graph should be clustered to discover groups of related tags.
- The model for the graph creation and clustering should be validated on an independent data simulation whose parameters can be adjusted to assess the models weaknesses.

The clustering algorithm for this project has the following requirements

- The algorithm must work on weighted undirected graphs.
- The algorithm implementation has to be able to handle graphs of 100000 nodes and above on a modern laptop computer.
- The algorithm must produce clusters without having to define a target number of clusters to produce. This is especially important as the community structure must be allowed to arise from the data instead of being forced upon it.

Additionally it would be desirable that the algorithm has a reference implementation in Java or Python to save development time.

3.3. Evaluation Metrics

The purpose of the proposed method is not to increase the accuracy of previous recommendation methods but instead to introduce a new user modelling tool. The evaluation methods for this project are then by necessity different to those discussed in the literature. As already mentioned in the previous section a simulation was built in order to have an objective truth against which the proposed method can be evaluated.

The simulation had a number of preset interest groups defined as two dimensional probability distributions over a tag matrix where each of the two dimensions define one letter of the two letter tags assigned to resources (Section 4.6). The distributions were stochastically sampled by a number of test users. The evaluation criteria was the level to which the proposed method captures the original preset interest groups. This method though could only be used for the simulated data as the real interest distributions are not known for the data collected from delicious.

A further evaluation criteria is to measure the quality of the match between the users, the resources and the recovered clusters. This method can be used both on the simulated data and the data collected from del.icio.us.

A third metric is using subjective analysis gathered from test users. This requires building a graphical user interface to allow the test users to interact with the system and to evaluate the results. The third metric was not considered however due to time restrictions.

4. System Design

4.1. Model Overview

In the introduction and requirements analysis chapters the current trend of recommendation systems were introduced. Problems with the current trend were also highlighted, including the problem of building recommendation systems based only on usage history and recommendation accuracy. Figure 2 illustrates a situation where this kind of approach can fail to produce recommendations that are useful for the end user. Users 1 and 2 share 66% of their browsing history but have different interests. `User 2` has purchased a gardening related product as a gift for a family member without having any other personal interest to the product or to what the product is related to, while `user 1` purchased the same product for personal use over the weekends. The interest they share in the product is clearly different. Given this scenario the tags those users have assigned to the product could be (`mom, birthday, gift_idea, gardening`) and (`gardening, tools, essentials`) respectively. The tags therefore act as an indicator for what the interest towards the resource. For the purposes of recommending new resources relevant to the users the reason for purchasing the product is therefore more important than the act itself.

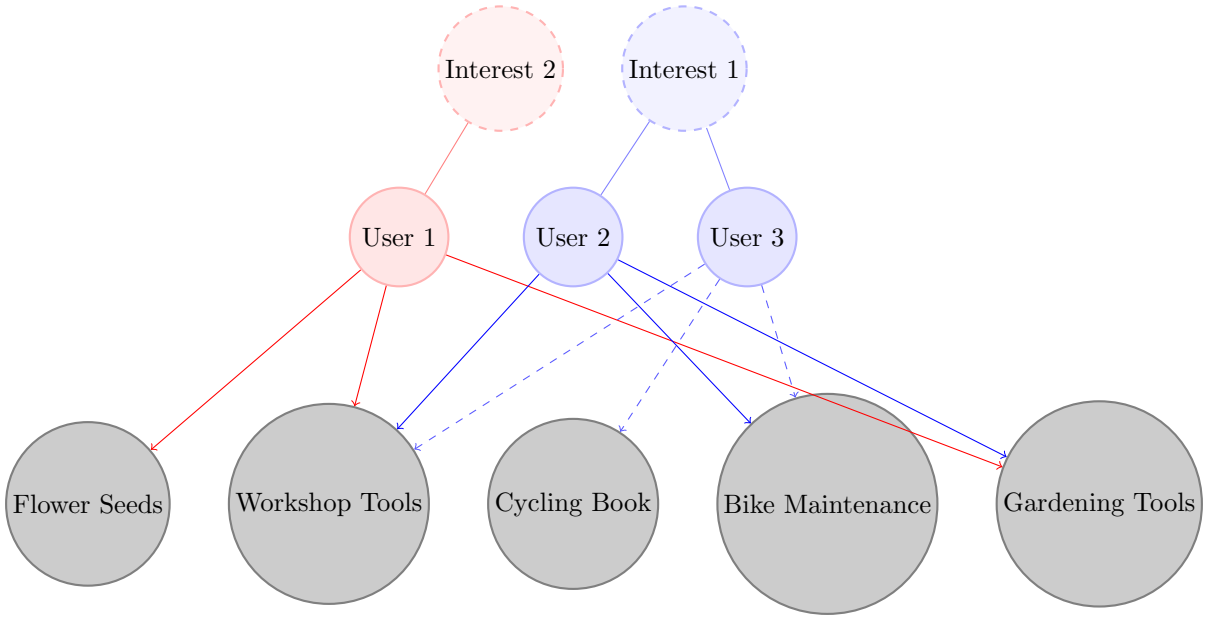


Figure 2: Groups 1 and 2 share 66% of the resources they have looked at, but the interests towards those resources are different.

Aggregating the tag information of many users to many resources allows building a model of what users overall tend to think about a resource, in other words what interest or interests is a particular resource related to. Similarly looking at the tag collections of single users allows modelling what the user is interested in. The connection between both is thus the tags, which leads to the idea of modelling the users and resources indirectly through the tags. The descriptor of a tag `t` is the list of (`user,resource,t`) tuples. Any particular tag cannot be defined by any one user or any one resource, but is always defined by a combination of the two.

From this tag information an undirected weighted graph is built where the nodes are the single tags and the edge weights are the similarities between the tags (the number of (`user,resource`) pairs shared

by the tags). A clustering algorithm is applied to the graph to discover groups of tags, which are treated as hidden interest groups. During clustering the user and resource information is temporarily discarded and the clustering is done purely on the semantics of the tags. After clustering the similarity of each user and each resource to each interest group can be computed.

To facilitate this model the social bookmarking service delicious was selected as a data source. Sections 1.3 and 1.4 gave an overview of how social tagging services work and how a coherent vocabulary describing some set of resources is formed over time by the interactions of many users. Figure 3 shows a part of a resource page in delicious and Section 4.2 describes the data collection process. In short delicious allows users of the service to save a URL reference along with a list of tags. The system is completely open in terms of what tags are used and what web resources are tagged. The list of all tags a user has used in the context of any resource forms a description of that users interests. Similarly the list of all tags used of any resource by all users form a description of that particular resource. Both these descriptions can be weighted by the frequency of the tags used in the description as defined in Equation 2.

A simple measure for the similarity of users to clusters is the overlap between the tags the user has used and the tags present in the interest group (Equation 1).

$$sim_1(u, c) = \frac{|u^t \cap c^t|}{|c^t|} \tag{1}$$

where $|A|$ is the cardinality of set A, u^t is a multiset of all tags used by user u and c^t is the set of tags in cluster c. Each user and resource then receives a distribution of values defining which interest groups that user is closest to. A slightly better metric for measuring the similarity between a user and a cluster takes into account the importance of each of the user's tags (Equation 2).

$$sim_2(u, c) = \frac{1}{|c^t| + |u^t|} \sum_{i=1}^{N} \mathbf{1}_{\mathbf{c^t}}(u_i^t) \tag{2}$$

where N is the number of items (note not number of distinct items) in u^t and $\mathbf{1}_{\mathbf{c^t}}(u_i^t)$ is the multiset indicator function $\mathbf{1}_{\mathbf{c^t}} : u_i^t \to \mathbb{Z}^{\geq}$, which tells the number of times tag u_i^t appears in the multiset u^t if and only if that tag is also present in c^t.

It is important to note that the users of delicious have the option of making the resources they have annotated public or private, and only the resources made public are available for data collection. The data collection therefore does not violate user privacy as the users have explicitly published the resources.

In Figure 3 underneath the history heading it is shown that the resource has been saved in total 387 times. Each line below that contains an annotation made by a specific user with the tag words assigned to the resource by that user at the end of the line in the grey boxes. The right hand side column displays the most popular tags across all the 387 annotations made to this resource. This shows that the two most popular tags are `education` and `ted`.

The other important thing to notice in the picture is the interconnections between the tags. Namely which tags are used with which other tags. From this screenshot it is clear that `technology` and `teaching` are used very frequently together. The tags `education` and `learning` are also used frequently with the previous two. These interactions between the tags reveal something about what users think of the resource. The resource has been saved for a specific reason and has been tagged with these specific words out of all possible words that the users could have chosen. There must be some importance therefore in these tags related to this resource and to each other. If these tags describe why people felt important to save this resource — their interest towards the resource in other words — then it should be possible to

gather the same kind of tag data across many resources and users and use that to build a model of what users of this service are interested in and to produce recommendations for those users not according to the resources they have already saved but according to their interests.

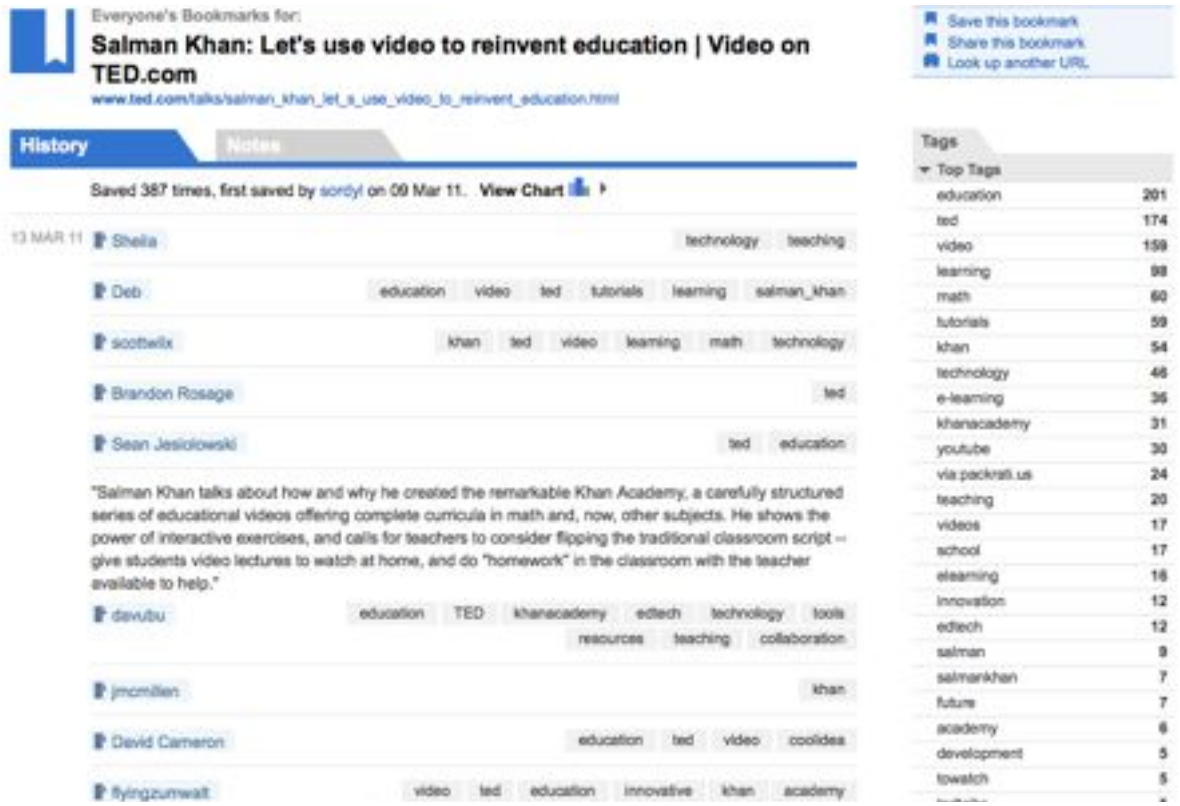


Figure 3: A part of a resource page in delicious.

Figure 4: Page navigation for a resource.

4.2. Web Crawler

The web crawler was designed to work on the resource and user pages and to automatically collect all necessary information for this project. Figure 5 shows a schematic of the crawler's functionality. To achieve this a persistent request fetch queue was designed around a MySQL database. This would allow the crawler to be restarted from where it left off in case of a failure. The failures could be due either because of HTTP errors regarding the requests, or MySQL errors such as duplicate private keys.

The crawler was built using Python's `urllib2` module and handled three specific kinds of HTTP errors, HTTP 404, HTTP 500 and HTTP 999. These are `page not found`, `internal server error` and `unable to process request` respectively. All other errors (including the MySQL errors) were logged and the crawler would continue running as normal. The `HTTP 404` would additionally cause that request URL to be removed from the database but not cause the crawler to stop. The `HTTP 999` error was found to happen when requests were made too often to the delicious servers. Essentially this means that the service provider was throttling the connection from the IP address where the requests were made. This is a way for the server administrators to protect the server from denial of service attacks where a web server is subjected to so great a load that it crashes. The `HTTP 999` error would cause the web crawler to stop processing the rest of the request queue and shut down immediately to comply with the server administrators judgement regarding the acceptable number of requests per second. By experimentation a timeout period of 45 seconds between consecutive requests was found to be adequate for the server not to start throttling the connection and thus to allow the crawler to keep running for days uninterrupted.

Upon a successful HTTP request the crawler parses the HTML extracting the necessary link information and then saves the HTML file to disk. In Figure 3 the blue link at the beginning of each line is a link to that user's personal page containing all the resources that user has saved and made public. The links on the resource page can be followed to find more users, and conversely the links on the users' pages can be followed to find more resources. Each user and resource can span several pages, signified by a page navigation at the bottom of each page (Figure 4). In order to collect as much information about each resource the page navigation at the bottom of the page was parsed and each page of the resource added to the beginning of the request fetch queue. This ensured that the opinions of all users concerning a particular resource were retrieved before moving on to the next resource. Because the clustering is done according to what users have said about particular resources, it was important to have as many opinions about a resource as possible. This necessarily affects the amount of information available about a single user. This tradeoff however was necessary to have enough data about the resources for the clustering method to be effective at all. Figure 5 shows how the web crawler works.

The logging behaviour was built using Python's built-in `logging` module. The module supports email based logs which allowed the crawler to send an email alert in case of a problem. This allowed leaving the crawler running for prolonged periods of time safe in the knowledge that any problems would be reported as they occur.

The web crawler was run a for a total of 18 days. During this time $\sim$ 2.5 Gb of HTML pages were collected. The dataset contained $\sim$ 9.5k distinct bookmarks, $\sim$ 400k distinct users and a vocabulary of $\sim$ 155k tags. Figure 6 shows the tag use counts plotted on a log-log scale, the mean use count of the tags was 35.7 for the sample data. For the resources the mean annotation count (number of users who had saved the resource) was 78.83 and the mean tag count was 31.83. Figure 7 shows the popularity of resources and the tag use counts of the corresponding resources. This shows some correspondence between the vocabulary size used to describe a resource and the number of users participating in the creation of that resource. It should be noted that no synonym detection was done to the data, so Figure 7 includes all tags that are syntactically different — for instance `e-commerce` and `ecommerce` — as separate tags. This was left out to limit the complexity of the project and because it was not a necessity. Indeed from the results in 4.5.2 the method seems to have resolved this issue without preprocessing.

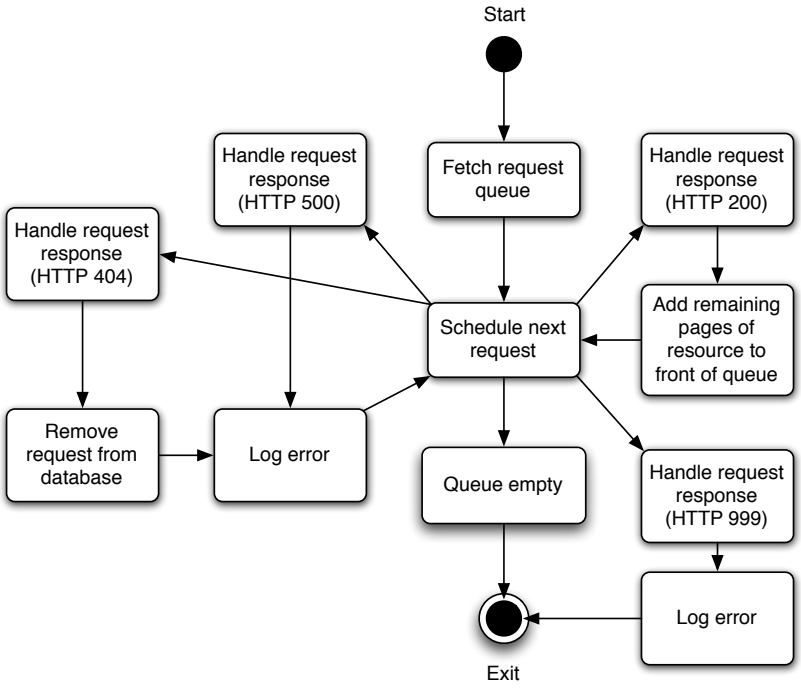


Figure 5: Crawler states and transitions between them.

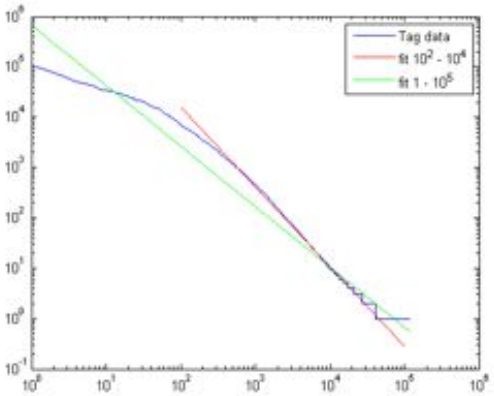


Figure 6: Tag use counts on a log log scale. Green line shows the slope of the power law fitted to the entire data set, red line shows the slope when fitted to data between 10^2 – 10^4. The powers are ~ -1.19 and ~ -1.5 respectively.

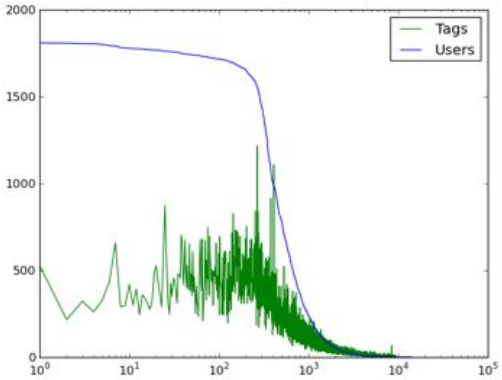


Figure 7: Resource popularity and tag usage. X axis on log scale to emphasise the distribution of resource popularity.

4.3. HTML Parser

The HTML parser was designed to work with the locally saved HTML files. The parser worked independently of the web crawler and was ran after the whole dataset was collected. Because only a limited amount of information was required from each HTML file and because the files had a consistent

and very distinctive structure the parser was designed around regular expressions instead of traversing the whole Document Object Model (DOM) of the HTML pages. This simplification was done mainly to save time during the development of the parser. The end result is not as robust and general as a DOM based parser would be but was sufficient for the purposes of this project.

The parser worked by going through each file in the local file store and extracting information one annotation at a time. Each annotation was split into a number of tuples of the form (`resourceID, userID, tag`). An annotation containing three tags would thus become three triples. These triples were then saved to a MySQL database described in 4.4.

4.4. Database Design

A MySQL database was designed to provide a persistent datastore for the web crawler and for the delicious dataset. Figure 8 shows the database schema. The Resources table was the main table of the database and contained all resources and all users; the `resource_type` parameter specified the type of the resource. The UserData and ResouceData tables contained meta information about the resources and users, such as the number of tags assigned to a resource or the number of tags used by a user. The Annotations table contained the parsed (`bookmarkID,userID,tagID`) tuples. This design preserves the contextual information about each tag and allowed computing the tag similarities during the clustering phase.

The database was first stored on the University of Sussex MySQL server, but this was found to be unreliable and slow and the entire database was later moved to a locally running MySQL instance.

The database was accessed both using a graphical user interface and with the GPL licensed MySQLdb Python module [10].

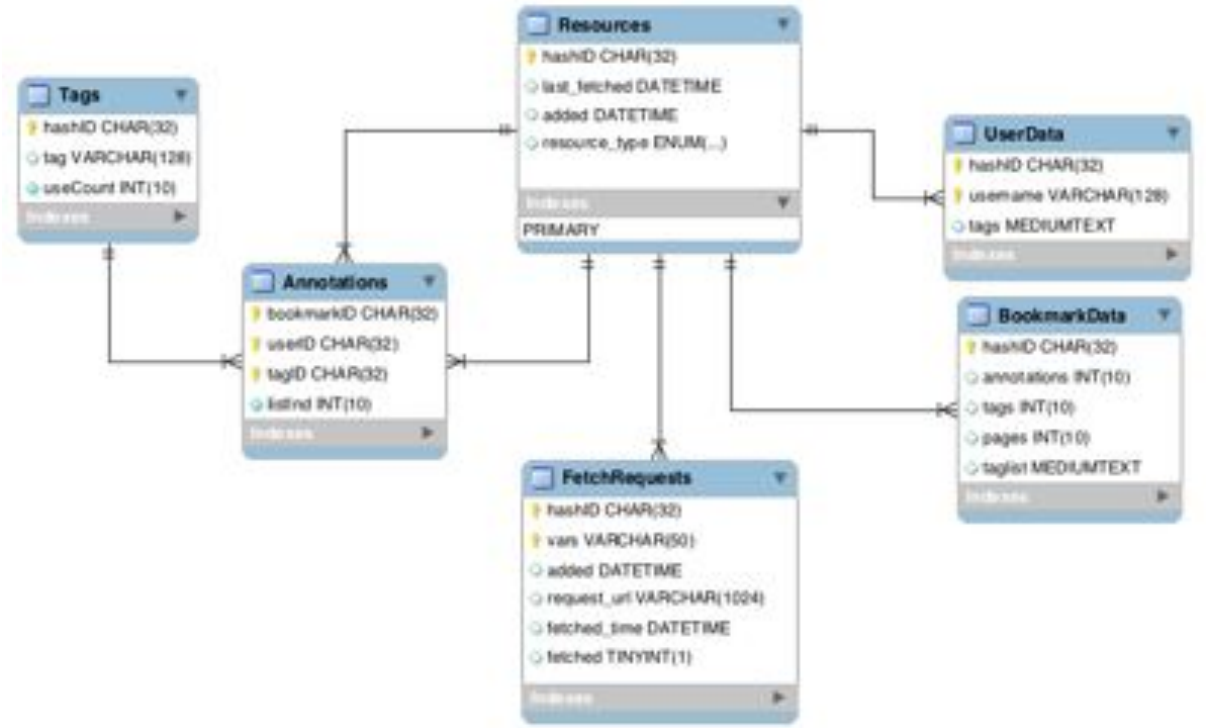


Figure 8: The database design.

4.5. Chinese Whispers

The clustering algorithm requirements were listed in 3.2. A number of clustering algorithms were researched for their suitability to the project and an algorithm called Chinese Whispers was eventually selected. Newman [26] gives a good overview of graph partitioning or clustering methods and also highlights some problems with traditional computer science approaches, one being that the number of clusters must be specified before running the clustering algorithm. If the problem involves allocating a number of processes to processors this is not a problem but for detecting an unknown number of communities in a social graph it is. This is true for well known and fairly simple algorithms such as k-meana or k-nearest neighbours as well as those listed by Newman. To address these problems Newman et. al propose their own solution to the community detection problem. No reference implementation for this algorithm however was found and Chinese Whispers was used instead for this research.

Chinese Whispers is a fast stochastic graph clustering algorithm that works on undirected weighted graphs. It has been developed by Biemann [6] for the purposes of natural language processing tasks. Chinese Whispers works by initialising every node in a graph in its own class. At every iteration of the algorithm all the nodes in the graph are iterated over in a random order and their class is changed to that of the strongest class in their neighbourhood. The strongest class is defined as the one whose sum of edge weights to node i is highest. There are several parameters that can be defined for the algorithm, including the number of iterations the algorithm should run, and the similarity measure to use for finding the strongest class in a node's neighbourhood. The algorithm has a reference Java implementation provided by the authors that was used in this project. The reference implementation has the following options for measuring the strongest class:

- 'top' – sum of edge weights to a class.
- 'dist log' / 'nolog' – downgrades the influence of a node by the degree of the node.
- 'vote' – same as top but similarities are normalised and vote takes an extra parameter defining a minimum proportion a class must cover of a node's neighbourhood before a class change.

4.5.1. Clustering Process

The aim for the clustering phase was to find a set of parameters that would distribute the tags as evenly as possible to all the clusters. This means few or preferably no clusters with only one tag in them and few very large clusters. The reason for avoiding very large clusters is to keep the semantic coherence of the clusters high. The larger a cluster becomes to more the semantic meaning of that cluster is generalised. To a certain extent this cannot be avoided due to the power law nature of the tag distribution, but finding good parameters for the algorithm should minimise this. On the other hand very small clusters are also problematic as comparing the similarities of two small clusters is difficult because there is no overlap between the two clusters.

Experimentation showed the 'dist nolog' measure to produce the best results for the delicious dataset. Figure 9 shows the behaviour of the different distance measures. This shows that `top` and `dist log` each produce a single huge cluster with more 10000 tags in it while the rest of the tags are dispersed in very small clusters of under 10 tags each. `Vote` depending on the threshold parameter either produces the same results as the previous two, or a very large number of single tag clusters. `Dist nolog` produces a good dispersion of cluster sizes.

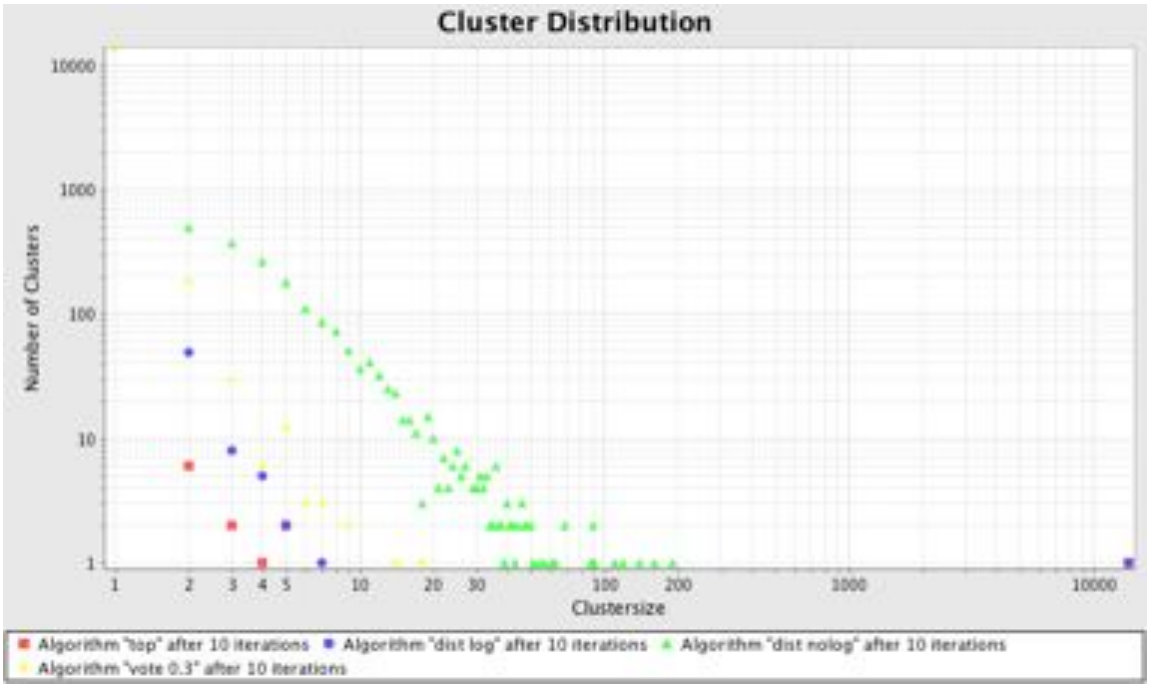


Figure 9: Cluster distributions produced by different class similarity measures.

4.5.2. Analysis of the 'Sundance' Cluster

During several runs of the clustering algorithm many of the clusters were found to be invariant. The clustering algorithm itself being stochastic this suggests that the data has a definite structure. Every time the algorithm is run the clusters are formed in a random order and the clusters formed earlier during the execution have a higher likelihood of attracting more tags to them as the total weight of that cluster grows the more tags the cluster contains. The invariant clusters therefore suggest that the tags in those clusters have a much stronger correlation with each other than any of the tags in the other clusters.

One such invariant cluster was the 'sundance' cluster related to the Sundance Independent Film Festival. This section provides an analysis of it and it's neighbours to illustrate the clustering results and to justify that the interest groups recovered by the clustering are semantically coherent. The section also suggests methods that could be use to clean up the dataset from noise. These methods however are outside of the scope of this project and are only provided as reference for possible future work.

The cluster typically had five nodes in it with the labels (`festival, filmfestival, utah, festivals, sundance`). The cluster is connected to 73 other clusters covering a wide range of topic areas. The connectedness to other clusters is measured by the combined edge weight of all edges between the nodes of the 'sundance' cluster and the nodes in a neighbouring cluster. Table 1 shows the contents of selected clusters. This table shows several important aspects about the whole delicious dataset and the clustering method. A quick glance at the clusters shows that synonyms and the singular and plural versions of words are often in the same cluster but as different nodes. These duplicate forms of tags could be dealt with by performing stemming on the tags and only keeping the root form of the tags in the dataset. This however is not a trivial task especially given the fact that the tag corpus contains several languages. Ideally then the tags would all be translated into a common language and then stemmed. While word stemming in english is fairly well understood machine translation is a field of active research.

This was therefore left outside the scope of this project.

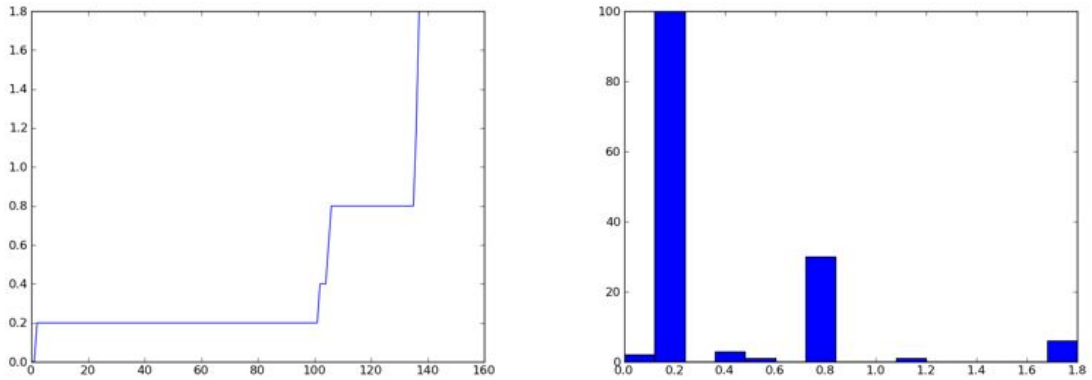
Semantically the clusters appear to be internally coherent and capture some distinct aspects of the whole dataset. For instance cluster ❶ has captured data regarding interface and usability design. The same cluster also shows a problem with the syntactic form of the tags. Jakob Nielsen's name — a well known web usability expert [3] — has eight different spellings. Named entity recognition could help in unifying these different spellings into one context.

Table 1 also shows that there are tags in many different languages. Most tags are in english, but spanish especially is a popular second language with tags such as `seguimento` and `monitorizacion` (cluster ❻), although the latter might be a misspelled english word. There are also tags at least in swedish `webbutveckling` and german `wienerrunde`.

Lastly Table 1 shows the ambiguities and noise present in the data. In cluster ❷ the tag `tiger` probably refers to the nickname of the Mac OS X operating system version 10.4 instead of the feline mammal. This problem could also be addressed with entity resolution within a context. One solution could be to use other web services such as wikipedia.org or freebase.com, which is an open data repository containing machine readable contextual information. The search term `tiger` returns a long list of results from freebase.com among them a musical artist, Tiger Woods, Siberian Tiger and Bengal Tiger.

There were 143 users connected to the sundance cluster as defined by Equation 1. The mean resource-to-resource similarity between the users in this group 21.5% and the weighted tag-to-tag similarity was 27.4%. The corresponding figures for the entire dataset were estimated at $\sim$ 0.3% and $\sim$ 2.1% respectively. The estimates were computed from a subset of 100000 users. Due to time and computational constraints it was not feasible to get a better estimate of these values. This however shows that even though the users within the sundance cluster only shared a fifth of their resources and a third of their tags, the clustering had significantly focused the user group. As mentioned in Section 4.2 not all of the resources of each user were contained in the dataset. This is bound to have had an effect on the accuracy measurements.

Figures 10(a) and 10(b) show how well each of the 143 users match to the sundance cluster. The measurement is done according to the weighted tag similarity in Equation 1.



(a) User similarities

(b) Histogram of user similarities

Figure 10: User similarities to the sundance cluster measured according to Equation 1.

❶

Node	Weight
...	
10	0
accessibility	0
bestpractice	0
bestpractices	0
contentmanagement	0
contentstrategy	0
corporate	2
corporateblogging	0
gui	0
guideline	0
guidelines	0
hci	0
interaction	0
interaction_design	0
interactiondesign	0
interface	0
interface_design	0
interfaces	0
jacob	0
jacobnielsen	0
jakob	0
jakob_nielsen	0
jakobnielsen	0
misconceptions	0
mitos	0
myth	0
myths	0
neilsen	0
nielsen	0
nielson	0
ui	0
uidesign	0
usabilidad	0
usabilidade	0
usability	0
usabilty	0
useit	0
user	0
user_interface	0
userexperience	0
userinterface	0
users	0
usertesting	0
...	

❷

Node	Weight
...	
bindings	0
cocoa	1
coredata	0
debug	0
debugger	0
debugging	0
devel	0
objc	0
objectivec	0
tiger	0
xcode	0
...	

❸

Node	Weight
...	
arts	0
foundation	0
foundationcenter	0
foundations	0
fundraising	0
giving	0
grant	2
grant_writing	0
grants	1
grantwriting	0
kinoeyepicks	0
opportunities	0
...	

❹

Node	Weight
poem	0
poetry	1
wasteland	0

❺

Node	Weight
cinematography	0
filmmaking	20

❻

Node	Weight
...	
adidas	0
brands	1
casestudies	0
casestudy	0
cocacola	0
coke	0
cola	0
monitorizacion	0
pringles	0
red	0
redbull	0
seguimiento	0
...	

❼

Node	Weight
...	
art	16
design	29
designresource	0
designs	0
eclectic	0
graphic	0
graphic_design	0
graphicdesign	0
inspiration	42
webbutveckling	0
wienerrunde	0
...	

❽

Node	Weight
...	
cinema	24
entertainment	14
film	130
films	3
media	13
movie	3
movies	65
towatch	0
tv	0
tv_online	0
tvlinks	0
tvshow	0
...	

Table 1: Clusters connected to the 'sundance' cluster. The ellipsis signifies that only part of the clusters content is shown. The weight is the sum of all edge weights from any tag in the sundance cluster to the specified tag in the neighbouring cluster. Please refer to the electronic submission for more information about the sundance cluster.

4.6. Simulation

The simulation was built in order to verify that the proposed method can recover the hidden interests of the users. It is assumed that the tags users assign to resources are governed by the interest the user has towards the resource — plus possibly some amount of noise. It should therefore be possible to take the aggregate of all the tags assigned to all resources and group the tags according to their use context. Since tag usage is always governed by the interest, given no noise in the data and a tag similarity measure based on the use context this grouping should produce groups of tags that resemble the original hidden interests. The item to item collaborative filtering approach relies on the overlap of items in users' viewing or purchase history. This method does not take into account the reason why those items are in the history (see Figure 2 on page 16) and can fail to produce worthwhile recommendations when users share a common resource but for different reasons. This resource overlap but diverging interests is the motivation for building the simulation.

The simulation defines a number of two dimensional normal distributions over a tag matrix. Each distribution represents an interest. Figure 11 shows the distributions that were used for the simulation.

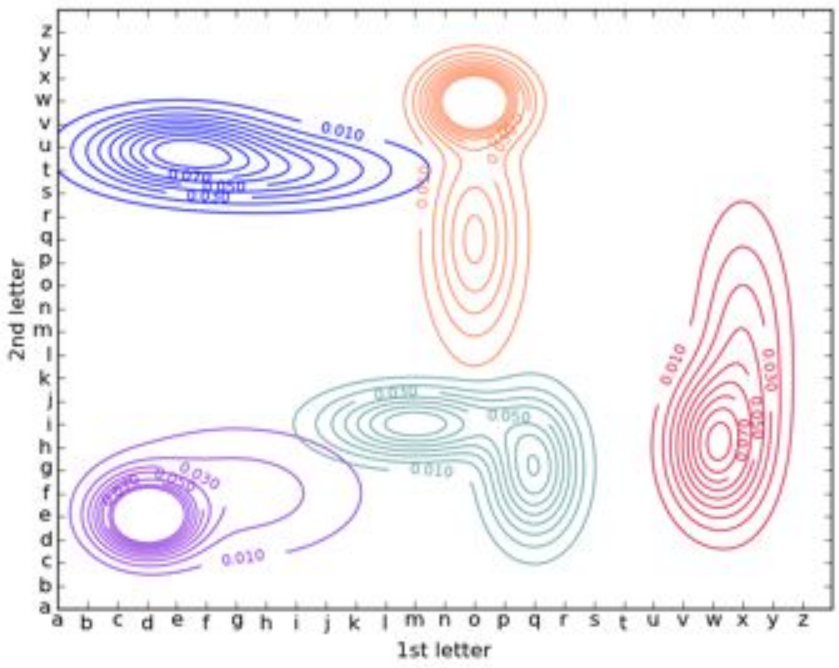


Figure 11: Interest distributions used in the simulation. A tag is formed by combining the letters picked from the x and y axis.

Annotations are produced by assigning each user one of the interest distributions defined above. This distribution defines which tags the user will assign to resources. Later the number of interests was increased to two to explore the effect of per user overlapping interests. A varying number of resources (between 5 and 25) is picked for each user to annotate. The resources are picked from a uniform distribution of dummy resource variables. The user will then annotate the resource by stochastically selecting tags from the interest distributions assigned to her. Users were set to assign 10 tags per

annotation. Depending on the number of users the tags used in the annotations would cover about half of the available vocabulary for 150 users to the entire 676 tag vocabulary for 3000 users.

After the annotation process a tag graph is computed from the annotations as defined in 4.1. The graph is ran through the clustering algorithm and the tag clusters are then compared to the original interest distributions. The model was also tested against noisy inputs by adding uniform noise to the annotations before the tag graph was built. Each annotation was iterated over and a proportion of tags within the annotation were changed to random tags picked uniformly from the entire tag matrix.

The simulation was run several times varying the number of users from 150 to 3000. In the case of each user having only one interest the clustering was found to reliably reproduce the original tag distributions as the tag clusters. The similarity between a tag cluster and the original interest distribution was defined as the Kullback-Leibler divergence measure between the tag cluster and a discrete sample taken from the original distribution at each point in the tag matrix. For discrete random variables this is defined as

$$Div_{KL}(P||Q) = \sum_i P(i) log \frac{P(i)}{Q(i)} \tag{3}$$

where $P(i)$ is the frequency of the i^{th} tag in the cluster and $Q(i)$ the probability of the tag. This measure approaches 0 when the tag cluster matches the original interest distribution. Figure 12 shows a typical result for the cluster match as defined by equation 3. It is clear that each cluster captures one and only one interest distribution. Figure 13 shows the clusters themselves and how they match the original interest distributions.

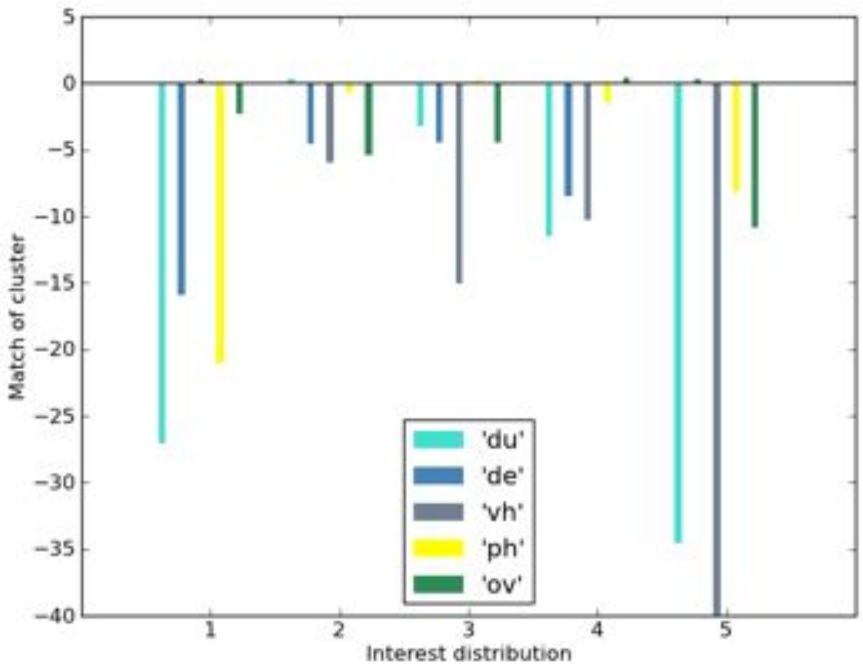
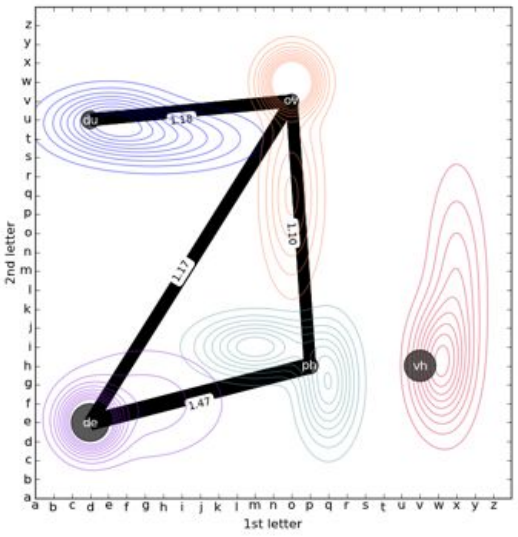
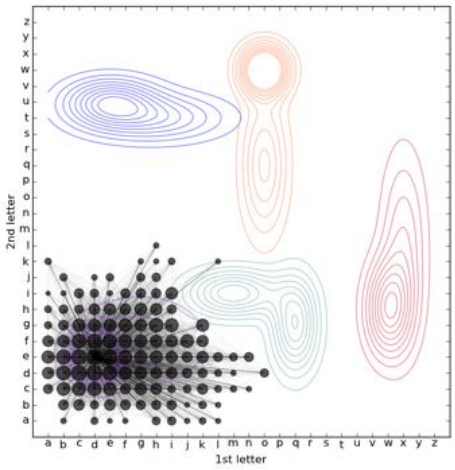


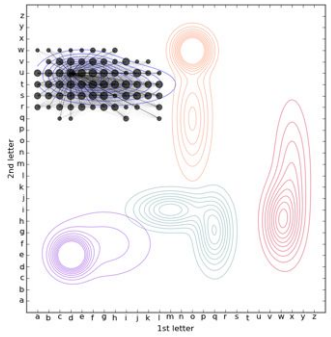
Figure 12: How the different clusters match the original interest distributions. Legend shows the label of the central node in each cluster.



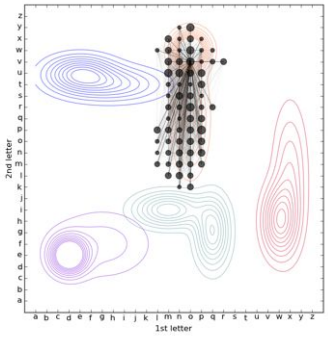
(a) Cluster hubs



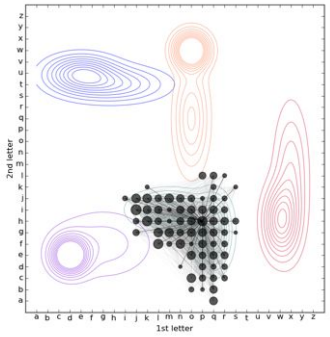
(b) Main cluster 'dd'



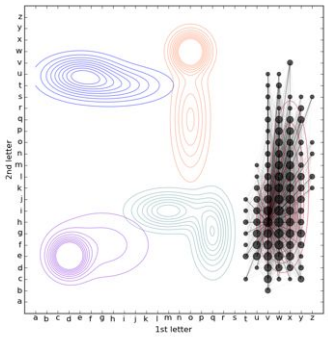
(c) Main cluster 'et'



(d) Main cluster 'mi'



(e) Main cluster 'nw'

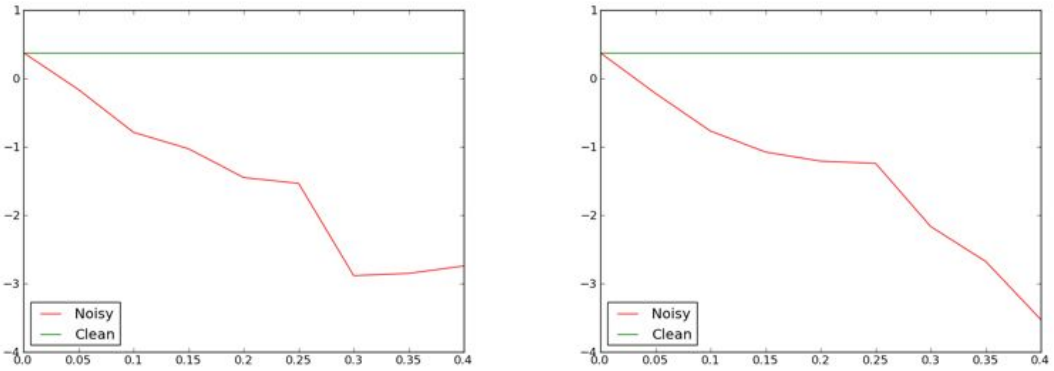


(f) Main cluster 'wg'

Figure 13: The hubs of each cluster and each cluster in detail.

Typically a low number of users would produce clustering results where there were one or two clusters with two tags in them. Increasing the number of users would often get rid of the smaller residual clusters as the connections between tags was increased. The existence of the residual clusters is also dependant on the settings of the clustering algorithm and noise in the annotations. The model seems to have a varying level of robustness against noise depending on the number of users creating the annotations. The model was tested by adding uniform noise into the annotations before clustering the tags. This would usually result in a catastrophic failure where only one cluster is produced when the noise was increased to 25% to 30%. Before that the model would capture the interest distributions well. This was found to be consistent through all sizes of user groups. Figure 14 shows how well the produced tag clusters match the original interest distributions when noise is increased. There is a clear shift in the trend of the fitness decrease around 25% of noise.

The drop in cluster fitness could be addressed however by changing the number of iterations the clustering algorithm was run. Initial tests suggested that a value of 8 iterations produced good results both for the delicious dataset and for the simulated data. 8 iterations were used for the majority of the tests. Varying the iteration count so as to match the number of clusters produced to the number of interest distributions per user was also tested. This was done by running the clustering algorithm consecutively each time increasing the number of iterations the algorithm would perform until five clusters remained. The results were found to be highly dependant on the number of users. As already noted when the number of users was low (150 – 500) the algorithm would never converge to only five clusters. For 1500 users and up however the results showed that even with a noise amount of 40% the method would still be able to recover the original interest clusters (Figure 14).



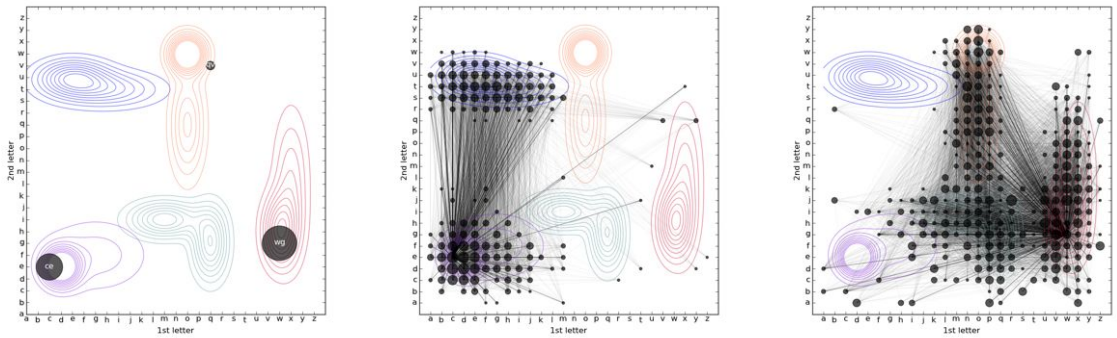
(a) Varying iteration count, force 5 clusters

(b) Fixed 8 Iterations

Figure 14: Cluster fits over different noise levels measured by Kullback-Leibler divergence. Both trials were done with 1500 users and averaged over 5 runs.

Changing the number of interests a user had to two had a significant effect on the clustering performance. A model where each user assigns tags from both interests to all resources was tested. The number of tags assigned to the resource was split evenly among the two interests. Intuitively this means that a user has two interests towards a resource and both interests are equally strong. This model produced clusters that did not capture a single interest but would span two or more interest distributions (Figure 15). Having the users assign tags to a resource from multiple interests creates very strong connections

between the tags and has a very strong influence on the clustering. This can also be observed from the number of clusters produced during the different number of iterations during the clustering. Typically for 1500 users five clusters would be produced after 7 to 8 iterations of the clustering algorithm. Increasing the number of interests reduces this number such that after only 4 to 5 iterations there would be only one cluster left. These settings were only tested to see how the model reacts. For the delicious dataset it is not likely that the user tags would be evenly distributed among several interests given one resource.



(a) Cluster hubs (b) Cluster 'ce' (c) Cluster 'wg'

Figure 15: Final clusters for simulated data when using 2 interests per user. The clusters are produced from clean data with 1500 users.

4.7. Producing Recommendations

To evaluate the feasibility of using the proposed method to produce recommendations a comparison between the item-to-item similarity measure and the proposed method was done. As the purpose of this project was to explore the feasibility of a new user modelling method and not specifically to produce a recommendation model the tested method is unoptimised. There are also several areas of improvement in data handling such as named entity recognition and tag word stemming as already mentioned in the previous sections.

The test users were 10 users from the delicious dataset whose data were not used in the graph computations. The users themselves were unseen to the methods, but some of the resources and tags of the users were not. The item-to-item similarity measurements were done against a subsample of 15000 randomly selected users. The item similarity to each 15000 was computed and all of the resources of the top 10 users were then selected as the recommendations.

For the tag based method the similarity of the test users to each of the clusters was first computed. For every cluster whose similarity to the user was greater than 0.0 a number a resources were selected from that cluster. The resources connected to that cluster ($sim(r, c) > 0.0$ Equation 2) were ordered according to $sim(r^t, c^t) + sim(r^t, u^t)$ and the top 5 resources from each cluster were then selected as the recommendations.

For both methods the accuracy of the recommendation was defined as the number of the user's resources found in the recommendation set. The similarity between the user's tags and those of the recommendations was also computed. Table 2 shows the recommendation results. The tag based clustering method seems to be performing better on average with regards to the accuracy of the recommendations. The

weighted tag similarity on the other hand clearly favours the collaborative filtering approach. These results are preliminary and further analysis of the delicious dataset and the recommendation logic is needed for conclusive evidence. The purpose of this project has been to explore a way of producing recommendations based on the semantics of the resources being recommended. As such the final evaluation on the quality of the recommendations lies on the shoulders of end users.

Accuracy		Weighted Tag Similarity	
tag clusters	collaborative filtering	tag clusters	collaborative filtering
0.25	0.083	0.002	0.008
0.14	0.14	0.004	0.003
0.0	0.07	0.002	0.008
0.07	0.04	0.0	0.003
0.31	0.15	0.006	0.022
0.07	0.02	0.005	0.02
0.25	0.08	0.003	0.01
0.13	0.02	0.006	0.04
0.40	0.07	0.004	0.04
0.04	0.04	0.002	0.015

Table 2: Recommendation accuracy and tag similarity of recommended resources measured according to Equation 2.

5. Discussion

This report has presented a novel way for modelling user interests in the context of recommendation systems. The method is based on the assumption that tag usage is governed by interest towards a resource. A simulated dataset has confirmed that given a set of predefined interests that define what tags are used in resource annotations, the proposed method can recover the original interests as tag clusters even with noisy data.

Applying the method to real world data collected from delicious shows promising results in terms of the produced tag clusters, which were semantically coherent. Further work on preprocessing the data by performing entity resolution and word sense disambiguation could bring considerable improvements to the performance.

For the methods to be put into practical use a lot of further research needs to be done, both on the dynamics of tag systems and the methods presented in this paper. Incorporating new and changed annotation information into the dataset for instance is something that cannot be overlooked for any real world system. This aside tag and keyword based models for recommendation systems show promise. The model is also not constrained to only tag information specified by users and could be extended for instance by keyword or keyphrase extraction mechanisms to apply to systems that do not already contain the tag information used in this paper. Some previous research into these methods have been done at least by Mika in [24]. This would allow the model to be applied to areas other than those relying on social bookmarking. For instance the product reviews found in many online retail websites could be used to build a keyword description about the products.

As it stands however the method serves mainly as a proof of concept for the methodology and guiding future work. The data set collected for the research was relatively small due to restrictions in computational resources and data collection time. Enlarging the dataset would be a first step for further research. Wetzker et. al [34] did a large scale study of delicious and showed delicious bookmarks and annotations to be prone to spam further increasing the importance of preprocessing the data. A small number of automated clients was observed to post large quantities of bookmarks per day or create a one time dump of thousands of bookmarks infrequently. Typically these automated agents had a small vocabulary for the annotations. Their dataset has been released for others to use [33], but at $\sim 950k$ users and ~ 132 million bookmarks was too big to be used for this paper.

There are two big issues that have not been fully explored due to time restrictions. The dataset for this research was not cleaned from noise in any. Wetzker's study [34] showed delicious data to be subject to spam. In order to improve the method preprocessing should therefore be applied to the dataset. The similarity measurements between tags should be researched in more detail. Only a first order similarity metric was used in this research. Even though this was shown to produce semantically coherent clusters it does not provide a full picture of the semantic relationships between tags. Second order similarity — overlap in the tag neighbourhood of two tags — is one such improvement. Furthermore the mechanics of producing the recommendations was only implemented as during this research to get an idea of how the method performs. Producing good recommendations being the goal of an end user system this area provides lots of further research possibilities especially in terms of exploring the different tasks of recommender systems identified by Herlocker [14]. These have only now started to attract the attention of the research community [17].

The tag clustering algorithm used for this research worked reasonably well but at times was found to be cumbersome and unreliable. Many clusters were semantically internally coherent others however

were not. The similarity metric between tags and noise in the data affect this as well as the choice of the clustering algorithm. Changing the parameters of Chinese Whispers [6] often produced unexpected and undesirable results. Implementing the community detection methods described by Newman in [26] would be a good starting point for further research.

Lastly in the introduction it was discussed that a user description not based on the data held by the recommendation system (usually a collection of products, but in this case groups of tags) could be mapped on to the interest space built using the proposed method. For this research the implementation remained the weighted overlap of tags defined in Equation 2. This measurement however could be abstracted much further from the overlap of the tag sets describing the two entities by using thesauri and other semantic information sources to expand the tags. These methods however require considerable computational resources.

5.1. Practical Applications

Even though being purely research focused the work still has relevant practical applications. Similar methods to the ones described in this paper have been used for detecting emerging trends [32] for instance. The methods can also be used for building, expanding and amending formal ontologies, or thesauri with some post processing of the results presented here.

The motivation for this work came from abstracting user profiles from the product catalog of a particular recommendation system. The results show that more work is required however before this is feasible in a reliable way. That said this remains a possibility and a promising step towards better services that rely not only on what users do but also why.

References

[1] delicious.com. http://www.delicious.com/. Accessed: 26/03/2011.

[2] Flickr.com. http://www.flickr.com. Accessed: 26/03/2011.

[3] www.useit.com. http://www.useit.com/. Accessed: 13/04/2011.

[4] Aggarwal, C. C., Wolf, J. L., Wu, K.-L., and Yu, P. S. Horting hatches an egg: a new graph-theoretic approach to collaborative filtering. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 1999), ACM, pp. 201–212.

[5] Balabanović, M., and Shoham, Y. Fab: content-based, collaborative recommendation. *Commun. ACM 40*, 3 (1997), 66–72.

[6] Biemann, C. Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing* (Stroudsburg, PA, USA, 2006), TextGraphs-1, Association for Computational Linguistics, pp. 73–80.

[7] British Computer Society. Code of Good Practice, 2004.

[8] British Computer Society. Code of Conduct for BCS Members, 2006.

[9] Chris, A. The long tail. *The Wired Magazine 12*, 10 (10 2004), 170–177.

[10] Dustman, A. Python MySQLdb module. http://sourceforge.net/projects/mysql-python/. Accessed 04.05.2011.

[11] Fleder, D., and Hosanagar, K. Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity. *Manage. Sci. 55*, 5 (2009), 697–712.

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

[13] Halpin, H., Robu, V., and Shepherd, H. The complex dynamics of collaborative tagging. In *WWW '07: Proceedings of the 16th international conference on World Wide Web* (New York, NY, USA, 2007), ACM, pp. 211–220.

[14] Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst. 22*, 1 (2004), 5–53.

[15] Hotho, A., Jäschke, R., Schmitz, C., and Stumme, G. Information retrieval in folksonomies: Search and ranking. In *The Semantic Web: Research and Applications*, Y. Sure and J. Domingue, Eds., vol. 4011 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2006, pp. 411–426. 10.1007/11762256_31.

[16] Kautz, H., Selman, B., and Shah, M. Referral Web: combining social networks and collaborative filtering. *Commun. ACM 40*, 3 (1997), 63–65.

[17] Kawamae, N. Serendipitous recommendations via innovators. In *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2010), ACM, pp. 218–225.

[18] Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. GroupLens: applying collaborative filtering to Usenet news. *Commun. ACM 40*, 3 (1997), 77–87.

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

[20] man Au Yeung, C., Gibbins, N., and Shadbolt, N. Discovering and modelling multiple interests of users in collaborative tagging systems. In *Web Intelligence and Intelligent Agent Technology* (dec. 2008), vol. 3, pp. 115 –118.

[21] Martinez, A. B. B., Lopez, M. R., Montenegro, E. C., Fonte, F. A. M., Burguillo, J. C., and Peleteiro, A. Exploitation of social tagging for outperforming traditional recommender systems techniques. *IEEE Internet Computing 99*, PrePrints (2010).

[22] Mathes, A. Folksonomies – cooperative classification and communication through shared metadata. *Computer Mediated Communication - LIS590CMC* (December 2004).

[23] Michlmayr, E. Learning user profiles from tagging data and leveraging them for personal(ized) information access. In *In Proceedings of the Workshop on Tagging and Metadata for Social Information Organization, 16th International World Wide Web Conference (WWW2007* (2007).

[24] Mika, P. Ontologies are us: A unified model of social networks and semantics. In *The Semantic Web – ISWC 2005*, Y. Gil, E. Motta, V. Benjamins, and M. Musen, Eds., vol. 3729 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2005, pp. 522–536.

[25] Milicevic, A. K., Nanopoulos, A., and Ivanovic, M. Social tagging in recommender systems: a survey of the state-of-the-art and possible extensions. *Artif. Intell. Rev. 33*, 3 (2010), 187–209.

[26] Newman, M. Detecting community structure in networks. *The European Physical Journal B - Condensed Matter and Complex Systems 38* (2004), 321–330. 10.1140/epjb/e2004-00124-y.

[27] Newman, M. Power laws, Pareto distributions and Zipf's law. *Contemporary physics 46*, 5 (2005), 323–351.

[28] Rucker, J., and Polanco, M. J. Siteseer: personalized navigation for the Web. *Commun. ACM 40*, 3 (1997), 73–76.

[29] Sarwar, B., Karypis, G., Konstan, J., and Reidl, J. Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web* (New York, NY, USA, 2001), ACM, pp. 285–295.

[30] Song, Y., Zhang, L., and Giles, C. L. Automatic tag recommendation algorithms for social recommender systems. *ACM Trans. Web 5* (February 2011), 4:1–4:31.

[31] Terveen, L., Hill, W., Amento, B., McDonald, D., and Creter, J. PHOAKS: a system for sharing recommendations. *Commun. ACM 40*, 3 (1997), 59–62.

[32] Wetzker, R., Plumbaum, T., Korth, A., Bauckhage, C., Alpcan, T., and Metze, F. Detecting trends in social bookmarking systems using a probabilistic generative model and smoothing. In *Proceedings of the International Conference on Pattern Recognition (ICPR)* (2008).

[33] Wetzker, R., and Zimmermann, C. Analyzing social bookmarking systems: A del.icio.us cookbook (dataset). http://www.dai-labor.de/en/irml/datasets/delicious/. Accessed 30.04.2011.

[34] Wetzker, R., and Zimmermann, C. Analyzing social bookmarking systems: A del.icio.us cookbook. In *European Conference on Artifial Intelligence* (7 2008), pp. 26–30.

[35] Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J. R., and Zhang, Y.-C. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences 107*, 10 (2010), 4511–4515.

Appendices

A. Work Log

Autumn Term

Fri Oct 8th

Read *Item-Based Collaborative Filtering Recommendation Algorithms* (Sarwar2001)

Read *On the recommending of citations for research* (McNee2002)

Mon Oct 11th

Read *Combining social networks and collaborative filtering* (Kautz1997)

Read *Discovering shared interests using graph analysis* (Schwartz1993)

Tue Oct 12th

Read *Mediation of user models for enhanced personalization in recommender systems* (Berkovsky2009)

Wed Oct 13th

Read *Discovering and Modelling Multiple Interests of Users in Collaborative Tagging Systems* (Yeung2008)

Read *A recommender model for social bookmarking sites* (Ilhan2009)

Fri Oct 15th

Writing research proposal

Mon Oct 18th

Finished research proposal

Met with Dr. Luc Berthouzeand discussed the work done the previous week. Research question accepted.

Tue Oct 19th

Wrote project proposal final version

Organised work log and reading from previous week.

Wed Oct 20th

Started exploring delicious tags and how to scrape them.

Read *Exploitation of Social Tagging for Outperforming Traditional Recommender Systems Techniques* (Martinez2010)

Read *Folksonomies - Cooperative Classification and Communication Through Shared Metadata* (Mathes2004)

Read *Information Retrieval in Folksonomies: Search and Ranking* (Hotho2006)

Fri Oct 22nd

Mining test data from delicious

Mon Oct 25th

Read Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity (Fleder2009).

Tue Oct 26th

Read *Straightening out heavy tails* (Savage2010).

Started writing interim report.

Wed Oct 27th

Read *Content based, collaborative recommendation* (Balabanovic1997).

Met with Dr. Luc Berthouzeto discuss progress. Progress is in the right direction, agreed to deliver a draft of the interim report on week 5.

Wrote interim report introduction (unfinished).

Thu Oct 28th

Read *The complex dynamics of social tagging systems* (Halpin2007).

Wrote interim report introduction (unfinished).

Read *Power laws, Pareto distributions and Zipf's law* (Newman2005).

Wrote professional considerations (finished).

Mon Nov 1st
Read *Social tagging in recommender systems: a survey of the state-of-the-art and possible extensions* (Milisevic 2010).
Read *Exploiting Additional Context for Graph-Based Tag Recommendations in Folksonomy Systems* (Abel2008).

Tue Nov 2nd
First draft of interim report finished and sent to Dr. Luc Berthouze.

Wed Nov 3rd
Met with Dr. Luc Berthouzeto discuss interim report draft.
Draft is going in a good direction. More focus needs to be given to the specific research question.

Thu Nov 4th
Started re writing requirements analysis
Read *Learning user profiles from tagging data and leveraging them for personal(ized) information access* Michlmayr2007.

Fri Nov 5th
Wrote 2nd draft of interim report.

Mon Nov 8th
Read *Detecting community structure in networks* (Newman2004).

Tue Nov 9th
Writing final version of interim report.

Wed Nov 10th
Writing final version of interim report.

Thu Nov 11th
Writing final version of interim report.

Spring Term

Mon Jan 10th
Meeting with Dr. Luc Berthouzeonly had time to go through the collected data quickly before meeting and the data seems to be ok.

Tue Jan 11th to Wed Jan 12th
Analysed collected data, cleaned up database and ran HTML parser to get tags into database. Data insertion taking longer than expected.

Mon Jan 17th
Meeting with Dr. Luc Berthouze. Discussed problems with HTML parser and database being (very) slow and unpredictable.

Tue Jan 18th
Started testing Chinese Whispers on dummy data provided by Biemann. Chinese Whispers has a graph rendering tool, so I won't need to implement one myself.

Tue Jan 19th
Tried to get Chinese Whispers running on real data set. Cannot - program crashes.

Mon Jan 24th
Meeting with Dr. Luc Berthouze. Discussed problems with Chinese Whispers, Luc suggested increasing the max heap memory of JVM. Unsurprisingly this worked.

Tue Jan 25th
Tried Chinese Whispers on real data set, parameters seem to be a little fiddly. Manages to get some good results.

Wed Jan 26th
Played around some more with Chinese Whispers.

Mon Jan 31st
Meeting with Dr. Luc Berthouze. Running a bit ahead of schedule since I don't need to implement a clustering algorithm myself.

Tue Feb 1st
Started implementing similarity measure for tag

comparisons, discovered a bug in the HTML parser. Have to throw the entire dataset away and reinsert the dataset into the database.

Wed Feb 2nd
Fixed bug in HTML parser, rest of week will be spent running the data insertion again.

Mon Feb 7th
Meeting with Dr. Luc Berthouze. Discussed HTML parser bug issue, still a little ahead of schedule though.

Tue Feb 8th
Finished implementing tag similarity measure. Running similarity computations - database being very slow at times.

Wed Feb 9th
Running similarity computations - database being very slow, spent an hour trying to get a connection to it.

Sat Feb 12th
Got the similarity computations done. Database being a real pain, will need to transfer database onto local machine.

Mon Feb 14th
Meeting with Dr. Luc Berthouze. Discussed problems with database. Discussed the need to create a data simulation to be able to validate what I'm doing and evaluate the quality of the system.

Tue Feb 15th
Started looking into creating the data simulation. Transferred database onto local machine, working much better now.

Wed Feb 16th
Designing data simulation.

Mon Feb 21st
Meeting with Dr. Luc Berthouze. Discussed data simulation design and how best to validate what I'm doing, and also what it is that I'm doing the first place. Project plan has now changed quite drastically, on top of doing the analysis for the delicious data I need to make the simulation. Staring to be a bit short on time, luckily I saved some time not having to implement the clustering algorithm.

Tue Feb 22nd to Fri Feb 25th
Implementing data simulation.

Mon Feb 28th
Meeting with Dr. Luc Berthouze. Went over data simulation plans and implementation. All is well.

Tue Mar 1st
Running data simulator. Results seem to be good.

Mon Mar 7th
Meeting with Dr. Luc Berthouze. Had nothing to report, have had no time to do anything for the project.

Mon Mar 14th
Meeting with Dr. Luc Berthouze. Had nothing to report, have had no time to do anything for the project.

Mon Mar 21st to Fri Mar 25th
Started writing draft report. Running simulation and doing tests on the original delicious dataset.

Sat Apr 2nd to Thursday Apr 7th
Writing draft report.

Fri Apr 8th
Handed in draft report to Dr. Luc Berthouze. Should get feedback in about a week.

Wed Apr 13th
Got feedback from Dr. Luc Berthouze. Meeting him on Monday to over it.

Mon Apr 18th
Meeting with Dr. Luc Berthouze. Discussed draft report, need to do a bit more writing but otherwise things seem to be in order.

Mon Apr 25th to Fri Apr 29th
Writing final report.

Tue May 3rd
Writing final report.

Wed May 4th
Final report hand in.